

EViews 6 User's Guide II



Quantitative Micro Software

EViews 6 User's Guide II

Copyright © 1994–2007 Quantitative Micro Software, LLC

All Rights Reserved

Printed in the United States of America

This software product, including program code and manual, is copyrighted, and all rights are reserved by Quantitative Micro Software, LLC. The distribution and sale of this product are intended for the use of the original purchaser only. Except as permitted under the United States Copyright Act of 1976, no part of this product may be reproduced or distributed in any form or by any means, or stored in a database or retrieval system, without the prior written permission of Quantitative Micro Software.

Disclaimer

The authors and Quantitative Micro Software assume no responsibility for any errors that may appear in this manual or the EViews program. The user assumes all responsibility for the selection of the program to achieve intended results, and for the installation, use, and results obtained from the program.

Trademarks

Windows, Windows 95/98/2000/NT/Me/XP, and Microsoft Excel are trademarks of Microsoft Corporation. PostScript is a trademark of Adobe Corporation. X11.2 and X12-ARIMA Version 0.2.7 are seasonal adjustment programs developed by the U. S. Census Bureau. Tramo/Seats is copyright by Agustin Maravall and Victor Gomez. All other product names mentioned in this manual may be trademarks or registered trademarks of their respective companies.

Quantitative Micro Software, LLC

4521 Campus Drive, #336, Irvine CA, 92612-2621

Telephone: (949) 856-3368

Fax: (949) 856-2044

e-mail: sales@eviews.com

web: www.eviews.com

March 9, 2007

Preface

The first volume of the EViews 6 *User's Guide* describes the basics of using EViews. This second volume, *User's Guide II*, offers a description of EViews' interactive statistical and estimation features.

The *User's Guide II* is divided into five parts:

- [Part IV. "Basic Single Equation Analysis" on page 3](#) discusses standard regression analysis: ordinary least squares, weighted least squares, two-stage least squares (TSLS), nonlinear least squares, time series analysis, specification testing and forecasting.
- [Part V. "Advanced Single Equation Analysis," beginning on page 183](#) documents autoregressive conditional heteroskedasticity (ARCH) models, discrete and limited dependent variable models, user-specified likelihood estimation, and quantile regression.
- [Part VI. "Multiple Equation Analysis" on page 305](#) describes estimation and forecasting with systems of equations (least squares, weighted least squares, SUR, system TSLS, 3SLS, FIML, GMM, multivariate ARCH), vector autoregression and error correction models, state space models, and model solution.
- [Part VII. "Panel and Pooled Data" on page 457](#) documents working with and estimating models with time series, cross-section data. The analysis may involve small numbers of cross-sections, with data unstacked series (pooled data) or large numbers systems of cross-sections, with stacked data (panel data).
- [Part VIII. "Other Multivariate Analysis," beginning on page 577](#) describes the estimation of factor analysis models in EViews.

Part IV. Basic Single Equation Analysis

The following chapters describe the EViews features for basic single equation analysis.

- [Chapter 24. “Basic Regression,” beginning on page 5](#) outlines the basics of ordinary least squares estimation in EViews.
- [Chapter 25. “Additional Regression Methods,” on page 23](#) discusses weighted least squares, two-stage least squares and nonlinear least square estimation techniques.
- [Chapter 26. “Time Series Regression,” on page 63](#) describes single equation regression techniques for the analysis of time series data: testing for serial correlation, estimation of ARMAX and ARIMAX models, using polynomial distributed lags, and unit root tests for nonstationary time series.
- [Chapter 27. “Forecasting from an Equation,” beginning on page 113](#) outlines the fundamentals of using EViews to forecast from estimated equations.
- [Chapter 28. “Specification and Diagnostic Tests,” beginning on page 141](#) describes specification testing in EViews.

The chapters describing advanced single equation techniques for autoregressive conditional heteroskedasticity, and discrete and limited dependent variable models are listed in [Part V. “Advanced Single Equation Analysis”](#).

Multiple equation estimation is described in the chapters listed in [Part VI. “Multiple Equation Analysis”](#).

[Chapter 39. “Panel Estimation,” beginning on page 541](#) describes estimation in panel structured workfiles.

Chapter 24. Basic Regression

Single equation regression is one of the most versatile and widely used statistical techniques. Here, we describe the use of basic regression techniques in EViews: specifying and estimating a regression model, performing simple diagnostic analysis, and using your estimation results in further analysis.

Subsequent chapters discuss testing and forecasting, as well as more advanced and specialized techniques such as weighted least squares, two-stage least squares (TSLS), nonlinear least squares, ARIMA/ARIMAX models, generalized method of moments (GMM), GARCH models, and qualitative and limited dependent variable models. These techniques and models all build upon the basic ideas presented in this chapter.

You will probably find it useful to own an econometrics textbook as a reference for the techniques discussed in this and subsequent documentation. Standard textbooks that we have found to be useful are listed below (in generally increasing order of difficulty):

- Pindyck and Rubinfeld (1998), *Econometric Models and Economic Forecasts*, 4th edition.
- Johnston and DiNardo (1997), *Econometric Methods*, 4th Edition.
- Wooldridge (2000), *Introductory Econometrics: A Modern Approach*.
- Greene (1997), *Econometric Analysis*, 3rd Edition.
- Davidson and MacKinnon (1993), *Estimation and Inference in Econometrics*.

Where appropriate, we will also provide you with specialized references for specific topics.

Equation Objects

Single equation regression estimation in EViews is performed using the *equation object*. To create an equation object in EViews: select **Object/New Object.../Equation** or **Quick/Estimate Equation...** from the main menu, or simply type the keyword `equation` in the command window.

Next, you will specify your equation in the **Equation Specification** dialog box that appears, and select an estimation method. Below, we provide details on specifying equations in EViews. EViews will estimate the equation and display results in the equation window.

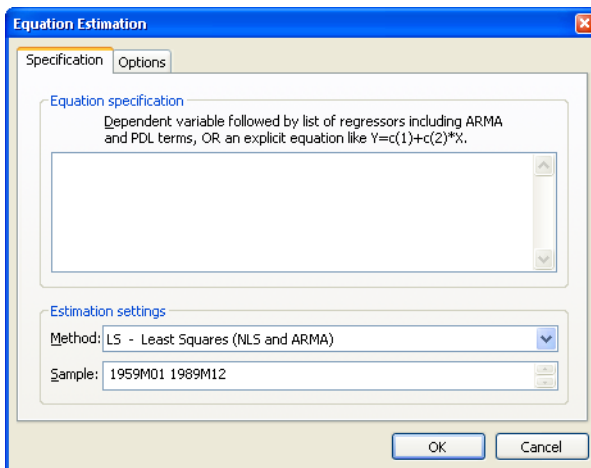
The estimation results are stored as part of the equation object so they can be accessed at any time. Simply open the object to display the summary results, or to access EViews tools for working with results from an equation object. For example, you can retrieve the sum-of-squares from any equation, or you can use the estimated equation as part of a multi-equation model.

Specifying an Equation in EViews

When you create an equation object, a specification dialog box is displayed.

You need to specify three things in this dialog: the equation specification, the estimation method, and the sample to be used in estimation.

In the upper edit box, you can specify the equation: the dependent (left-hand side) and independent (right-hand side) variables and the functional form. There are two basic ways of specifying an equation: “by list” and “by formula” or “by expression”. The list method is easier but may only be used with unrestricted linear specifications; the formula method is more general and must be used to specify nonlinear models or models with parametric restrictions.



Specifying an Equation by List

The simplest way to specify a linear equation is to provide a list of variables that you wish to use in the equation. First, include the name of the dependent variable or expression, followed by a list of explanatory variables. For example, to specify a linear consumption function, CS regressed on a constant and INC, type the following in the upper field of the **Equation Specification** dialog:

```
cs c inc
```

Note the presence of the series name C in the list of regressors. This is a built-in EViews series that is used to specify a constant in a regression. EViews does not automatically include a constant in a regression so you must explicitly list the constant (or its equivalent) as a regressor. The internal series C does not appear in your workfile, and you may not use it outside of specifying an equation. If you need a series of ones, you can generate a new series, or use the number 1 as an auto-series.

You may have noticed that there is a pre-defined object C in your workfile. This is the *default coefficient vector*—when you specify an equation by listing variable names, EViews stores the estimated coefficients in this vector, in the order of appearance in the list. In the example above, the constant will be stored in C(1) and the coefficient on INC will be held in C(2).

Lagged series may be included in statistical operations using the same notation as in generating a new series with a formula—put the lag in parentheses after the name of the series. For example, the specification:

```
cs cs(-1) c inc
```

tells EViews to regress CS on its own lagged value, a constant, and INC. The coefficient for lagged CS will be placed in C(1), the coefficient for the constant is C(2), and the coefficient of INC is C(3).

You can include a consecutive range of lagged series by using the word “to” between the lags. For example:

```
cs c cs(-1 to -4) inc
```

regresses CS on a constant, CS(-1), CS(-2), CS(-3), CS(-4), and INC. If you don't include the first lag, it is taken to be zero. For example:

```
cs c inc(to -2) inc(-4)
```

regresses CS on a constant, INC, INC(-1), INC(-2), and INC(-4).

You may include auto-series in the list of variables. If the auto-series expressions contain spaces, they should be enclosed in parentheses. For example:

```
log(cs) c log(cs(-1)) ((inc+inc(-1)) / 2)
```

specifies a regression of the natural logarithm of CS on a constant, its own lagged value, and a two period moving average of INC.

Typing the list of series may be cumbersome, especially if you are working with many regressors. If you wish, EViews can create the specification list for you. First, highlight the dependent variable in the workfile window by single clicking on the entry. Next, CTRL-click on each of the explanatory variables to highlight them as well. When you are done selecting all of your variables, double click on any of the highlighted series, and select **Open/Equation...**, or right click and select **Open/as Equation...**. The **Equation Specification** dialog box should appear with the names entered in the specification field. The constant C is automatically included in this list; you must delete the C if you do not wish to include the constant.

Specifying an Equation by Formula

You will need to specify your equation using a formula when the list method is not general enough for your specification. Many, but not all, estimation methods allow you to specify your equation using a formula.

An equation formula in EViews is a mathematical expression involving regressors and coefficients. To specify an equation using a formula, simply enter the expression in the dialog in

place of the list of variables. EViews will add an implicit additive disturbance to this equation and will estimate the parameters of the model using least squares.

When you specify an equation by list, EViews converts this into an equivalent equation formula. For example, the list,

```
log(cs) c log(cs(-1)) log(inc)
```

is interpreted by EViews as:

```
log(cs) = c(1) + c(2)*log(cs(-1)) + c(3)*log(inc)
```

Equations do not have to have a dependent variable followed by an equal sign and then an expression. The “=” sign can be anywhere in the formula, as in:

```
log(urate) - c(1)*dmr = c(2)
```

The residuals for this equation are given by:

$$\epsilon = \log(\text{urate}) - c(1)\text{dmr} - c(2). \quad (24.1)$$

EViews will minimize the sum-of-squares of these residuals.

If you wish, you can specify an equation as a simple expression, without a dependent variable and an equal sign. If there is no equal sign, EViews assumes that the entire expression is the disturbance term. For example, if you specify an equation as:

```
c(1)*x + c(2)*y + 4*z
```

EViews will find the coefficient values that minimize the sum of squares of the given expression, in this case $(C(1)*X + C(2)*Y + 4*Z)$. While EViews will estimate an expression of this type, since there is no dependent variable, some regression statistics (e.g. R-squared) are not reported and the equation cannot be used for forecasting. This restriction also holds for any equation that includes coefficients to the left of the equal sign. For example, if you specify:

```
x + c(1)*y = c(2)*z
```

EViews finds the values of $C(1)$ and $C(2)$ that minimize the sum of squares of $(X + C(1)*Y - C(2)*Z)$. The estimated coefficients will be identical to those from an equation specified using:

```
x = -c(1)*y + c(2)*z
```

but some regression statistics are not reported.

The two most common motivations for specifying your equation by formula are to estimate restricted and nonlinear models. For example, suppose that you wish to constrain the coefficients on the lags on the variable X to sum to one. Solving out for the coefficient restriction leads to the following linear model with parameter restrictions:

$$y = c(1) + c(2)*x + c(3)*x(-1) + c(4)*x(-2) + (1 - c(2) - c(3) - c(4))*x(-3)$$

To estimate a nonlinear model, simply enter the nonlinear formula. EViews will automatically detect the nonlinearity and estimate the model using nonlinear least squares. For details, see [“Nonlinear Least Squares” on page 43](#).

One benefit to specifying an equation by formula is that you can elect to use a different coefficient vector. To create a new coefficient vector, choose **Object/New Object...** and select **Matrix-Vector-Coeff** from the main menu, type in a name for the coefficient vector, and click **OK**. In the New Matrix dialog box that appears, select **Coefficient Vector** and specify how many rows there should be in the vector. The object will be listed in the workfile directory with the coefficient vector icon (the little β).

You may then use this coefficient vector in your specification. For example, suppose you created coefficient vectors A and BETA, each with a single row. Then you can specify your equation using the new coefficients in place of C:

$$\log(cs) = a(1) + beta(1) * \log(cs(-1))$$

Estimating an Equation in EViews

Estimation Methods

Having specified your equation, you now need to choose an estimation method. Click on the **Method:** entry in the dialog and you will see a drop-down menu listing estimation methods.

Standard, single-equation regression is performed using least squares. The other methods are described in subsequent chapters.

Equations estimated by ordinary least squares and two-stage least squares, equations with AR terms, GMM, and ARCH equations may be specified with a formula. Expression equations are not allowed with binary, ordered, censored, and count models, or in equations with MA terms.

| | |
|----------|---|
| LS | - Least Squares (NLS and ARMA) |
| TSLS | - Two-Stage Least Squares (TSNLS and ARMA) |
| GMM | - Generalized Method of Moments |
| ARCH | - Autoregressive Conditional Heteroskedasticity |
| BINARY | - Binary Choice (Logit, Probit, Extreme Value) |
| ORDERED | - Ordered Choice |
| CENSORED | - Censored or Truncated Data (including Tobit) |
| COUNT | - Integer Count Data |
| QREG | - Quantile Regression (including LAD) |
| STEPLS | - Stepwise Least Squares |

Estimation Sample

You should also specify the sample to be used in estimation. EViews will fill out the dialog with the current workfile sample, but you can change the sample for purposes of estimation by entering your sample string or object in the edit box (see [“Samples” on page 86](#) of the *User's Guide I* for details). Changing the estimation sample does not affect the current workfile sample.

If any of the series used in estimation contain missing data, EViews will temporarily adjust the estimation sample of observations to exclude those observations (listwise exclusion). EViews notifies you that it has adjusted the sample by reporting the actual sample used in the estimation results:

```
Dependent Variable: Y
Method: Least Squares
Date: 08/19/97   Time: 10:24
Sample(adjusted): 1959:01 1989:12
Included observations: 340
Excluded observations: 32 after adjusting endpoints
```

Here we see the top of an equation output view. EViews reports that it has adjusted the sample. Out of the 372 observations in the period 1959M01–1989M12, EViews uses the 340 observations with observations for all of the relevant variables.

You should be aware that if you include lagged variables in a regression, the degree of sample adjustment will differ depending on whether data for the pre-sample period are available or not. For example, suppose you have nonmissing data for the two series M1 and IP over the period 1959:01–1989:12 and specify the regression as:

```
m1 c ip ip(-1) ip(-2) ip(-3)
```

If you set the estimation sample to the period 1959:01–1989:12, EViews adjusts the sample to:

```
Dependent Variable: M1
Method: Least Squares
Date: 08/19/97   Time: 10:49
Sample: 1960:01 1989:12
Included observations: 360
```

since data for IP(-3) are not available until 1959M04. However, if you set the estimation sample to the period 1960M01–1989M12, EViews will not make any adjustment to the sample since all values of IP(-3) are available during the estimation sample.

Some operations, most notably estimation with MA terms and ARCH, do not allow missing observations in the middle of the sample. When executing these procedures, an error message is displayed and execution is halted if an NA is encountered in the middle of the sample. EViews handles missing data at the very start or the very end of the sample range by adjusting the sample endpoints and proceeding with the estimation procedure.

Estimation Options

EViews provides a number of estimation options. These options allow you to weight the estimating equation, to compute heteroskedasticity and auto-correlation robust covariances, and to control various features of your estimation algorithm. These options are discussed in detail in [“Estimation Options” on page 46](#).

Equation Output

When you click **OK** in the Equation Specification dialog, EViews displays the equation window displaying the estimation output view:

| Dependent Variable: LOG(M1) | | | | |
|-----------------------------|-------------|-----------------------|-------------|--------|
| Method: Least Squares | | | | |
| Date: 08/18/97 Time: 14:02 | | | | |
| Sample: 1959:01 1989:12 | | | | |
| Included observations: 372 | | | | |
| | Coefficient | Std. Error | t-Statistic | Prob. |
| C | -1.699912 | 0.164954 | -10.30539 | 0.0000 |
| LOG(IP) | 1.765866 | 0.043546 | 40.55199 | 0.0000 |
| TB3 | -0.011895 | 0.004628 | -2.570016 | 0.0106 |
| R-squared | 0.886416 | Mean dependent var | 5.663717 | |
| Adjusted R-squared | 0.885800 | S.D. dependent var | 0.553903 | |
| S.E. of regression | 0.187183 | Akaike info criterion | -0.505429 | |
| Sum squared resid | 12.92882 | Schwarz criterion | -0.473825 | |
| Log likelihood | 97.00980 | Hannan-Quinn criter. | 0.008687 | |
| F-statistic | 0.000000 | | | |

Using matrix notation, the standard regression may be written as:

$$y = X\beta + \epsilon \quad (24.2)$$

where y is a T -dimensional vector containing observations on the dependent variable, X is a $T \times k$ matrix of independent variables, β is a k -vector of coefficients, and ϵ is a T -vector of disturbances. T is the number of observations and k is the number of right-hand side regressors.

In the output above, y is log(M1), X consists of three variables C, log(IP), and TB3, where $T = 372$ and $k = 3$.

Coefficient Results

Regression Coefficients

The column labeled “Coefficient” depicts the estimated coefficients. The least squares regression coefficients b are computed by the standard OLS formula:

$$b = (X'X)^{-1}X'y \quad (24.3)$$

If your equation is specified by list, the coefficients will be labeled in the “Variable” column with the name of the corresponding regressor; if your equation is specified by formula, EViews lists the actual coefficients, C(1), C(2), *etc.*

For the simple linear models considered here, the coefficient measures the marginal contribution of the independent variable to the dependent variable, holding all other variables fixed. If present, the coefficient of the C is the constant or intercept in the regression—it is the base level of the prediction when all of the other independent variables are zero. The other coefficients are interpreted as the slope of the relation between the corresponding independent variable and the dependent variable, assuming all other variables do not change.

Standard Errors

The “Std. Error” column reports the estimated standard errors of the coefficient estimates. The standard errors measure the statistical reliability of the coefficient estimates—the larger the standard errors, the more statistical noise in the estimates. If the errors are normally distributed, there are about 2 chances in 3 that the true regression coefficient lies within one standard error of the reported coefficient, and 95 chances out of 100 that it lies within two standard errors.

The covariance matrix of the estimated coefficients is computed as:

$$\text{var}(b) = s^2(X'X)^{-1}; \quad s^2 = \hat{\epsilon}'\hat{\epsilon}/(T-k); \quad \hat{\epsilon} = y - Xb \quad (24.4)$$

where $\hat{\epsilon}$ is the residual. The standard errors of the estimated coefficients are the square roots of the diagonal elements of the coefficient covariance matrix. You can view the whole covariance matrix by choosing **View/Covariance Matrix**.

t-Statistics

The t -statistic, which is computed as the ratio of an estimated coefficient to its standard error, is used to test the hypothesis that a coefficient is equal to zero. To interpret the t -statistic, you should examine the probability of observing the t -statistic given that the coefficient is equal to zero. This probability computation is described below.

In cases where normality can only hold asymptotically, EViews will report a z -statistic instead of a t -statistic.

Probability

The last column of the output shows the probability of drawing a t -statistic (or a z -statistic) as extreme as the one actually observed, under the assumption that the errors are normally distributed, or that the estimated coefficients are asymptotically normally distributed.

This probability is also known as the p -value or the *marginal significance level*. Given a p -value, you can tell at a glance if you reject or accept the hypothesis that the true coefficient is zero against a two-sided alternative that it differs from zero. For example, if you are performing the test at the 5% significance level, a p -value lower than 0.05 is taken as evidence

to reject the null hypothesis of a zero coefficient. If you want to conduct a one-sided test, the appropriate probability is one-half that reported by EViews.

For the above example output, the hypothesis that the coefficient on TB3 is zero is rejected at the 5 % significance level but not at the 1 % level. However, if theory suggests that the coefficient on TB3 cannot be positive, then a one-sided test will reject the zero null hypothesis at the 1 % level.

The p -values are computed from a t -distribution with $T - k$ degrees of freedom.

Summary Statistics

R-squared

The R-squared (R^2) statistic measures the success of the regression in predicting the values of the dependent variable within the sample. In standard settings, R^2 may be interpreted as the fraction of the variance of the dependent variable explained by the independent variables. The statistic will equal one if the regression fits perfectly, and zero if it fits no better than the simple mean of the dependent variable. It can be negative for a number of reasons. For example, if the regression does not have an intercept or constant, if the regression contains coefficient restrictions, or if the estimation method is two-stage least squares or ARCH.

EViews computes the (centered) R^2 as:

$$R^2 = 1 - \frac{\hat{\epsilon}'\hat{\epsilon}}{(y - \bar{y})'(y - \bar{y})}; \quad \bar{y} = \sum_{t=1}^T y_t / T \quad (24.5)$$

where \bar{y} is the mean of the dependent (left-hand) variable.

Adjusted R-squared

One problem with using R^2 as a measure of goodness of fit is that the R^2 will never decrease as you add more regressors. In the extreme case, you can always obtain an R^2 of one if you include as many independent regressors as there are sample observations.

The adjusted R^2 , commonly denoted as \bar{R}^2 , penalizes the R^2 for the addition of regressors which do not contribute to the explanatory power of the model. The adjusted R^2 is computed as:

$$\bar{R}^2 = 1 - (1 - R^2) \frac{T-1}{T-k} \quad (24.6)$$

The \bar{R}^2 is never larger than the R^2 , can decrease as you add regressors, and for poorly fitting models, may be negative.

Standard Error of the Regression (S.E. of regression)

The standard error of the regression is a summary measure based on the estimated variance of the residuals. The standard error of the regression is computed as:

$$s = \sqrt{\frac{\hat{\epsilon}'\hat{\epsilon}}{(T - k)}} \quad (24.7)$$

Sum-of-Squared Residuals

The sum-of-squared residuals can be used in a variety of statistical calculations, and is presented separately for your convenience:

$$\hat{\epsilon}'\hat{\epsilon} = \sum_{t=1}^T (y_t - X_t'b)^2 \quad (24.8)$$

Log Likelihood

EViews reports the value of the log likelihood function (assuming normally distributed errors) evaluated at the estimated values of the coefficients. Likelihood ratio tests may be conducted by looking at the difference between the log likelihood values of the restricted and unrestricted versions of an equation.

The log likelihood is computed as:

$$l = -\frac{T}{2}(1 + \log(2\pi) + \log(\hat{\epsilon}'\hat{\epsilon}/T)) \quad (24.9)$$

When comparing EViews output to that reported from other sources, note that EViews does not ignore constant terms.

Durbin-Watson Statistic

The Durbin-Watson statistic measures the serial correlation in the residuals. The statistic is computed as

$$DW = \sum_{t=2}^T (\hat{\epsilon}_t - \hat{\epsilon}_{t-1})^2 / \sum_{t=1}^T \hat{\epsilon}_t^2 \quad (24.10)$$

See Johnston and DiNardo (1997, Table D.5) for a table of the significance points of the distribution of the Durbin-Watson statistic.

As a rule of thumb, if the DW is less than 2, there is evidence of positive serial correlation. The DW statistic in our output is very close to one, indicating the presence of serial correlation in the residuals. See [“Serial Correlation Theory,” beginning on page 63](#), for a more extensive discussion of the Durbin-Watson statistic and the consequences of serially correlated residuals.

There are better tests for serial correlation. In [“Testing for Serial Correlation” on page 64](#), we discuss the Q -statistic, and the Breusch-Godfrey LM test, both of which provide a more general testing framework than the Durbin-Watson test.

Mean and Standard Deviation (S.D.) of the Dependent Variable

The mean and standard deviation of y are computed using the standard formulae:

$$\bar{y} = \sum_{t=1}^T y_t / T; \quad s_y = \sqrt{\sum_{t=1}^T (y_t - \bar{y})^2 / (T - 1)} \quad (24.11)$$

Akaike Information Criterion

The Akaike Information Criterion (AIC) is computed as:

$$\text{AIC} = -2l / T + 2k / T \quad (24.12)$$

where l is the log likelihood (given by [Equation \(24.9\) on page 14](#)).

The AIC is often used in model selection for non-nested alternatives—smaller values of the AIC are preferred. For example, you can choose the length of a lag distribution by choosing the specification with the lowest value of the AIC. See [Appendix F. “Information Criteria,” on page 645](#), for additional discussion.

Schwarz Criterion

The Schwarz Criterion (SC) is an alternative to the AIC that imposes a larger penalty for additional coefficients:

$$\text{SC} = -2l / T + (k \log T) / T \quad (24.13)$$

F-Statistic

The F -statistic reported in the regression output is from a test of the hypothesis that *all* of the slope coefficients (excluding the constant, or intercept) in a regression are zero. For ordinary least squares models, the F -statistic is computed as:

$$F = \frac{R^2 / (k - 1)}{(1 - R^2) / (T - k)} \quad (24.14)$$

Under the null hypothesis with normally distributed errors, this statistic has an F -distribution with $k - 1$ numerator degrees of freedom and $T - k$ denominator degrees of freedom.

The p -value given just below the F -statistic, denoted **Prob(F-statistic)**, is the marginal significance level of the F -test. If the p -value is less than the significance level you are testing, say 0.05, you reject the null hypothesis that all slope coefficients are equal to zero. For the example above, the p -value is essentially zero, so we reject the null hypothesis that all of the

regression coefficients are zero. Note that the F -test is a joint test so that even if all the t -statistics are insignificant, the F -statistic can be highly significant.

Working With Equation Statistics

The regression statistics reported in the estimation output view are stored with the equation and are accessible through special “@-functions”. You can retrieve any of these statistics for further analysis by using these functions in genr, scalar, or matrix expressions. If a particular statistic is not computed for a given estimation method, the function will return an NA.

There are two kinds of “@-functions”: those that return a scalar value, and those that return matrices or vectors.

Keywords that return scalar values

| | |
|--------------|--|
| @aic | Akaike information criterion |
| @cofcov(i,j) | covariance of coefficient estimates i and j |
| @coefs(i) | i -th coefficient value |
| @dw | Durbin-Watson statistic |
| @f | F -statistic |
| @hq | Hannan-Quinn information criterion |
| @jstat | J -statistic — value of the GMM objective function (for GMM) |
| @logl | value of the log likelihood function |
| @meandep | mean of the dependent variable |
| @ncoef | number of estimated coefficients |
| @r2 | R-squared statistic |
| @rbar2 | adjusted R-squared statistic |
| @regobs | number of observations in regression |
| @schwarz | Schwarz information criterion |
| @sddep | standard deviation of the dependent variable |
| @se | standard error of the regression |
| @ssr | sum of squared residuals |
| @stderrs(i) | standard error for coefficient i |
| @tstats(i) | t -statistic value for coefficient i |
| c(i) | i -th element of default coefficient vector for equation (if applicable) |

Keywords that return vector or matrix objects

| | |
|----------|---|
| @cofcov | matrix containing the coefficient covariance matrix |
| @coefs | vector of coefficient values |
| @stderrs | vector of standard errors for the coefficients |
| @tstats | vector of <i>t</i> -statistic values for coefficients |

See also “Equation” (p. 27) in the *Command Reference*.

Functions that return a vector or matrix object should be assigned to the corresponding object type. For example, you should assign the results from @tstats to a vector:

```
vector tstats = eq1.@tstats
```

and the covariance matrix to a matrix:

```
matrix mycov = eq1.@cov
```

You can also access individual elements of these statistics:

```
scalar pvalue = 1-@cnorm(@abs(eq1.@tstats(4)))
```

```
scalar var1 = eq1.@covariance(1,1)
```

For documentation on using vectors and matrices in EViews, see [Chapter 18. “Matrix Language,” on page 627](#) of the *User’s Guide I*.

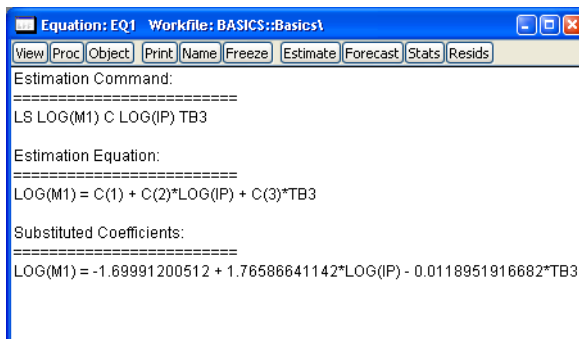
Working with Equations

Views of an Equation

- **Representations.** Displays the equation in three forms: EViews command form, as an algebraic equation with symbolic coefficients, and as an equation with the estimated values of the coefficients.

You can cut-and-paste from the representations view into any application that supports the Windows clipboard.

- **Estimation Output.** Displays the equation output results described above.
- **Actual, Fitted, Residual.** These views display the actual and fitted values of

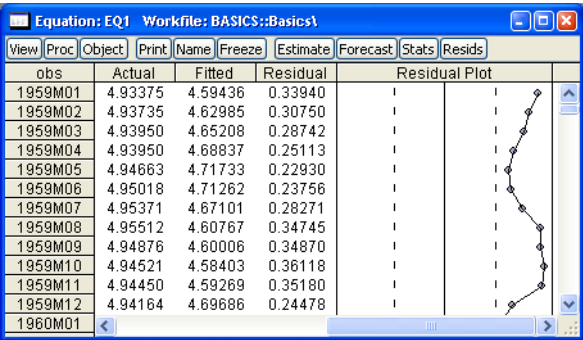


the dependent variable and the residuals from the regression in tabular and graphical form. **Actual, Fitted, Residual Table** displays these values in table form.

Note that the actual value is always the sum of the fitted value and the residual. **Actual, Fitted, Residual Graph** displays a standard EViews graph of the actual values, fitted values, and residuals.

Residual Graph plots only the residuals, while the **Standardized Residual**

Graph plots the residuals divided by the estimated residual standard deviation.



- **ARMA structure....** Provides views which describe the estimated ARMA structure of your residuals. Details on these views are provided in [“ARMA Structure” on page 83](#).
- **Gradients and Derivatives.** Provides views which describe the gradients of the objective function and the information about the computation of any derivatives of the regression function. Details on these views are provided in [Appendix E. “Gradients and Derivatives,” on page 637](#).
- **Covariance Matrix.** Displays the covariance matrix of the coefficient estimates as a spreadsheet view. To save this covariance matrix as a matrix object, use the @cov function.
- **Coefficient Tests, Residual Tests, and Stability Tests.** These are views for specification and diagnostic tests and are described in detail in [Chapter 28. “Specification and Diagnostic Tests,” beginning on page 141](#).

Procedures of an Equation

- **Specify/Estimate....** Brings up the **Equation Specification** dialog box so that you can modify your specification. You can edit the equation specification, or change the estimation method or estimation sample.
- **Forecast....** Forecasts or fits values using the estimated equation. Forecasting using equations is discussed in [Chapter 27. “Forecasting from an Equation,” on page 113](#).
- **Make Residual Series....** Saves the residuals from the regression as a series in the workfile. Depending on the estimation method, you may choose from three types of residuals: ordinary, standardized, and generalized. For ordinary least squares, only the ordinary residuals may be saved.

- **Make Regressor Group.** Creates an untitled group comprised of all the variables used in the equation (with the exception of the constant).
- **Make Gradient Group.** Creates a group containing the gradients of the objective function with respect to the coefficients of the model.
- **Make Derivative Group.** Creates a group containing the derivatives of the regression function with respect to the coefficients in the regression function.
- **Make Model.** Creates an untitled model containing a link to the estimated equation. This model can be solved in the usual manner. See [Chapter 36, “Models,” on page 407](#) for information on how to use models for forecasting and simulations.
- **Update Coefs from Equation.** Places the estimated coefficients of the equation in the coefficient vector. You can use this procedure to initialize starting values for various estimation procedures.

Residuals from an Equation

The residuals from the default equation are stored in a series object called RESID. RESID may be used directly as if it were a regular series, except in estimation.

RESID will be overwritten whenever you estimate an equation and will contain the residuals from the latest estimated equation. To save the residuals from a particular equation for later analysis, you should save them in a different series so they are not overwritten by the next estimation command. For example, you can copy the residuals into a regular EViews series called RES1 by the command:

```
series res1 = resid
```

Even if you have already overwritten the RESID series, you can always create the desired series using EViews' built-in procedures if you still have the equation object. If your equation is named EQ1, open the equation window and select **Proc/Make Residual Series**, or enter:

```
eq1.makeresid res1
```

to create the desired series.

Regression Statistics

You may refer to various regression statistics through the @-functions described above. For example, to generate a new series equal to FIT plus twice the standard error from the last regression, you can use the command:

```
series plus = fit + 2*eq1.@se
```

To get the *t*-statistic for the second coefficient from equation EQ1, you could specify

```
eq1.@tstats(2)
```

To store the coefficient covariance matrix from EQ1 as a named symmetric matrix, you can use the command:

```
sym ccov1 = eq1.@cov
```

See [“Keywords that return scalar values” on page 16](#) for additional details.

Storing and Retrieving an Equation

As with other objects, equations may be stored to disk in data bank or database files. You can also fetch equations from these files.

Equations may also be copied-and-pasted to, or from, workfiles or databases.

EViews even allows you to access equations directly from your databases or another workfile. You can estimate an equation, store it in a database, and then use it to forecast in several workfiles.

See [Chapter 4. “Object Basics,” beginning on page 63](#) and [Chapter 10. “EViews Databases,” beginning on page 257](#) of the *User’s Guide I* for additional information about objects, databases, and object containers.

Using Estimated Coefficients

The coefficients of an equation are listed in the representations view. By default, EViews will use the C coefficient vector when you specify an equation, but you may explicitly use other coefficient vectors in defining your equation.

These stored coefficients may be used as scalars in generating data. While there are easier ways of generating fitted values (see [“Forecasting from an Equation” on page 113](#)), for purposes of illustration, note that we can use the coefficients to form the fitted values from an equation. The command:

```
series cshat = eq1.c(1) + eq1.c(2)*gdp
```

forms the fitted value of CS, CSHAT, from the OLS regression coefficients and the independent variables from the equation object EQ1.

Note that while EViews will accept a series generating equation which does not explicitly refer to a named equation:

```
series cshat = c(1) + c(2)*gdp
```

and will use the existing values in the C coefficient vector, we strongly recommend that you always use named equations to identify the appropriate coefficients. In general, C will contain the correct coefficient values only immediately following estimation or a coefficient update. Using a named equation, or selecting **Proc/Update coefs from equation**, guarantees that you are using the correct coefficient values.

An alternative to referring to the coefficient vector is to reference the `@coefs` elements of your equation (see [page 16](#)). For example, the examples above may be written as:

```
series cshat=eq1.@coefs(1)+eq1.@coefs(2)*gdp
```

EViews assigns an index to each coefficient in the order that it appears in the representations view. Thus, if you estimate the equation:

```
equation eq01.ls y=c(10)+b(5)*y(-1)+a(7)*inc
```

where B and A are also coefficient vectors, then:

- `eq01.@coefs(1)` contains C(10)
- `eq01.@coefs(2)` contains B(5)
- `eq01.@coefs(3)` contains A(7)

This method should prove useful in matching coefficients to standard errors derived from the `@stderrs` elements of the equation (see [“Equation Data Members” on page 29](#) of the *Command Reference*). The `@coefs` elements allow you to refer to both the coefficients and the standard errors using a common index.

If you have used an alternative named coefficient vector in specifying your equation, you can also access the coefficient vector directly. For example, if you have used a coefficient vector named BETA, you can generate the fitted values by issuing the commands:

```
equation eq02.ls cs=beta(1)+beta(2)*gdp
series cshat=beta(1)+beta(2)*gdp
```

where BETA is a coefficient vector. Again, however, we recommend that you use the `@coefs` elements to refer to the coefficients of EQ02. Alternatively, you can update the coefficients in BETA prior to use by selecting **Proc/Update coefs from equation** from the equation window. Note that EViews does not allow you to refer to the named equation coefficients EQ02.BETA(1) and EQ02.BETA(2). You must instead use the expressions, EQ02.@COEFS(1) and EQ02.@COEFS(2).

Estimation Problems

Exact Collinearity

If the regressors are very highly collinear, EViews may encounter difficulty in computing the regression estimates. In such cases, EViews will issue an error message “Near singular matrix.” When you get this error message, you should check to see whether the regressors are *exactly* collinear. The regressors are exactly collinear if one regressor can be written as a linear combination of the other regressors. Under exact collinearity, the regressor matrix X does not have full column rank and the OLS estimator cannot be computed.

You should watch out for exact collinearity when you are using dummy variables in your regression. A set of mutually exclusive dummy variables and the constant term are exactly collinear. For example, suppose you have quarterly data and you try to run a regression with the specification:

$$y = c + x + @seas(1) + @seas(2) + @seas(3) + @seas(4)$$

EViews will return a “Near singular matrix” error message since the constant and the four quarterly dummy variables are exactly collinear through the relation:

$$c = @seas(1) + @seas(2) + @seas(3) + @seas(4)$$

In this case, simply drop either the constant term or one of the dummy variables.

The textbooks listed above provide extensive discussion of the issue of collinearity.

References

- Davidson, Russell and James G. MacKinnon (1993). *Estimation and Inference in Econometrics*, Oxford: Oxford University Press.
- Greene, William H. (1997). *Econometric Analysis*, 3rd Edition, Upper Saddle River, NJ: Prentice Hall.
- Johnston, Jack and John Enrico DiNardo (1997). *Econometric Methods*, 4th Edition, New York: McGraw-Hill.
- Pindyck, Robert S. and Daniel L. Rubinfeld (1998). *Econometric Models and Economic Forecasts*, 4th edition, New York: McGraw-Hill.
- Wooldridge, Jeffrey M. (2000). *Introductory Econometrics: A Modern Approach*, South-Western College Publishing.

Chapter 25. Additional Regression Methods

This first portion of this chapter describes special terms that may be used in estimation to estimate models with Polynomial Distributed Lags (PDLs) or dummy variables.

Next, we describe weighted least squares, heteroskedasticity and autocorrelation consistent covariance estimation, two-stage least squares (TSLS), nonlinear least squares, and generalized method of moments (GMM). Note that most of these methods are also available in systems of equations; see [Chapter 33. “System Estimation,” on page 308](#).

Lastly, we document tools for performing variable selection using stepwise regression.

Parts of this chapter refer to estimation of models which have autoregressive (AR) and moving average (MA) error terms. These concepts are discussed in greater depth in [Chapter 26. “Time Series Regression,” on page 63](#).

Special Equation Terms

EViews provides you with special terms that may be used to specify and estimate equations with PDLs, dummy variables, or ARMA errors. We begin with a discussion of PDLs and dummy variables, and defer the discussion of ARMA estimation to [“Time Series Regression” on page 63](#).

Polynomial Distributed Lags (PDLs)

A distributed lag is a relation of the type:

$$y_t = w_t\delta + \beta_0 x_t + \beta_1 x_{t-1} + \dots + \beta_k x_{t-k} + \epsilon_t \quad (25.1)$$

The coefficients β describe the lag in the effect of x on y . In many cases, the coefficients can be estimated directly using this specification. In other cases, the high collinearity of current and lagged values of x will defeat direct estimation.

You can reduce the number of parameters to be estimated by using polynomial distributed lags (PDLs) to impose a smoothness condition on the lag coefficients. Smoothness is expressed as requiring that the coefficients lie on a polynomial of relatively low degree. A polynomial distributed lag model with order p restricts the β coefficients to lie on a p -th order polynomial of the form,

$$\beta_j = \gamma_1 + \gamma_2(j - \bar{c}) + \gamma_3(j - \bar{c})^2 + \dots + \gamma_{p+1}(j - \bar{c})^p \quad (25.2)$$

for $j = 1, 2, \dots, k$, where \bar{c} is a pre-specified constant given by:

$$\bar{c} = \begin{cases} (k)/2 & \text{if } p \text{ is even} \\ (k-1)/2 & \text{if } p \text{ is odd} \end{cases} \quad (25.3)$$

The PDL is sometimes referred to as an Almon lag. The constant \bar{c} is included only to avoid numerical problems that can arise from collinearity and does not affect the estimates of β .

This specification allows you to estimate a model with k lags of x using only p parameters (if you choose $p > k$, EViews will return a “Near Singular Matrix” error).

If you specify a PDL, EViews substitutes Equation (25.2) into (25.1), yielding,

$$y_t = w_t \delta + \gamma_1 z_1 + \gamma_2 z_2 + \dots + \gamma_{p+1} z_{p+1} + \epsilon_t \quad (25.4)$$

where:

$$\begin{aligned} z_1 &= x_t + x_{t-1} + \dots + x_{t-k} \\ z_2 &= -\bar{c}x_t + (1 - \bar{c})x_{t-1} + \dots + (k - \bar{c})x_{t-k} \\ &\dots \\ z_{p+1} &= (-\bar{c})^p x_t + (1 - \bar{c})^p x_{t-1} + \dots + (k - \bar{c})^p x_{t-k} \end{aligned} \quad (25.5)$$

Once we estimate γ from Equation (25.4), we can recover the parameters of interest β , and their standard errors using the relationship described in Equation (25.2). This procedure is straightforward since β is a linear transformation of γ .

The specification of a polynomial distributed lag has three elements: the length of the lag k , the degree of the polynomial (the highest power in the polynomial) p , and the constraints that you want to apply. A near end constraint restricts the one-period lead effect of x on y to be zero:

$$\beta_{-1} = \gamma_1 + \gamma_2(-1 - \bar{c}) + \dots + \gamma_{p+1}(-1 - \bar{c})^p = 0. \quad (25.6)$$

A far end constraint restricts the effect of x on y to die off beyond the number of specified lags:

$$\beta_{k+1} = \gamma_1 + \gamma_2(k+1 - \bar{c}) + \dots + \gamma_{p+1}(k+1 - \bar{c})^p = 0. \quad (25.7)$$

If you restrict either the near or far end of the lag, the number of γ parameters estimated is reduced by one to account for the restriction; if you restrict both the near and far end of the lag, the number of γ parameters is reduced by two.

By default, EViews does not impose constraints.

How to Estimate Models Containing PDLs

You specify a polynomial distributed lag by the `pdl` term, with the following information in parentheses, each separated by a comma in this order:

- The name of the series.
- The lag length (the number of lagged values of the series to be included).
- The degree of the polynomial.

- A numerical code to constrain the lag polynomial (optional):

| | |
|---|--|
| 1 | constrain the near end of the lag to zero. |
| 2 | constrain the far end. |
| 3 | constrain both ends. |

You may omit the constraint code if you do not want to constrain the lag polynomial. Any number of `pdl` terms may be included in an equation. Each one tells EViews to fit distributed lag coefficients to the series and to constrain the coefficients to lie on a polynomial.

For example, the commands:

```
ls sales c pdl(orders,8,3)
```

fits SALES to a constant, and a distributed lag of current and eight lags of ORDERS, where the lag coefficients of ORDERS lie on a third degree polynomial with no endpoint constraints. Similarly:

```
ls div c pdl(rev,12,4,2)
```

fits DIV to a distributed lag of current and 12 lags of REV, where the coefficients of REV lie on a 4th degree polynomial with a constraint at the far end.

The `pdl` specification may also be used in two-stage least squares. If the series in the `pdl` is exogenous, you should include the PDL of the series in the instruments as well. For this purpose, you may specify `pdl(*)` as an instrument; all `pdl` variables will be used as instruments. For example, if you specify the TSLS equation as,

```
sales c inc pdl(orders(-1),12,4)
```

with instruments:

```
fed fed(-1) pdl(*)
```

the distributed lag of ORDERS will be used as instruments together with FED and FED(-1).

Polynomial distributed lags cannot be used in nonlinear specifications.

Example

We may estimate a distributed lag model of industrial production (IP) on money (M1) by entering the command:

```
ls ip c m1(0 to -12)
```

which yields the following results:

Dependent Variable: IP
 Method: Least Squares
 Date: 08/02/06 Time: 17:27
 Sample (adjusted): 1960M01 1989M12
 Included observations: 360 after adjustments

| | Coefficient | Std. Error | t-Statistic | Prob. |
|--------------------|-------------|-----------------------|-------------|--------|
| C | 40.67568 | 0.823866 | 49.37171 | 0.0000 |
| M1 | 0.129699 | 0.214574 | 0.604449 | 0.5459 |
| M1(-1) | -0.045962 | 0.376907 | -0.121944 | 0.9030 |
| M1(-2) | 0.033183 | 0.397099 | 0.083563 | 0.9335 |
| M1(-3) | 0.010621 | 0.405861 | 0.026169 | 0.9791 |
| M1(-4) | 0.031425 | 0.418805 | 0.075035 | 0.9402 |
| M1(-5) | -0.048847 | 0.431728 | -0.113143 | 0.9100 |
| M1(-6) | 0.053880 | 0.440753 | 0.122245 | 0.9028 |
| M1(-7) | -0.015240 | 0.436123 | -0.034944 | 0.9721 |
| M1(-8) | -0.024902 | 0.423546 | -0.058795 | 0.9531 |
| M1(-9) | -0.028048 | 0.413540 | -0.067825 | 0.9460 |
| M1(-10) | 0.030806 | 0.407523 | 0.075593 | 0.9398 |
| M1(-11) | 0.018509 | 0.389133 | 0.047564 | 0.9621 |
| M1(-12) | -0.057373 | 0.228826 | -0.250728 | 0.8022 |
| R-squared | 0.852398 | Mean dependent var | 71.72679 | |
| Adjusted R-squared | 0.846852 | S.D. dependent var | 19.53063 | |
| S.E. of regression | 7.643137 | Akaike info criterion | 6.943606 | |
| Sum squared resid | 20212.47 | Schwarz criterion | 7.094732 | |
| Log likelihood | -1235.849 | Hannan-Quinn criter. | 7.003697 | |
| F-statistic | 153.7030 | Durbin-Watson stat | 0.008255 | |
| Prob(F-statistic) | 0.000000 | | | |

Taken individually, none of the coefficients on lagged M1 are statistically different from zero. Yet the regression as a whole has a reasonable R^2 with a very significant F -statistic (though with a very low Durbin-Watson statistic). This is a typical symptom of high collinearity among the regressors and suggests fitting a polynomial distributed lag model.

To estimate a fifth-degree polynomial distributed lag model with no constraints, set the sample using the command,

```
smp1 1959m01 1989m12
```

then estimate the equation specification:

```
ip c pdl(m1,12,5)
```

by entering the expression in the **Equation Estimation** dialog and estimating using **Least Squares**.

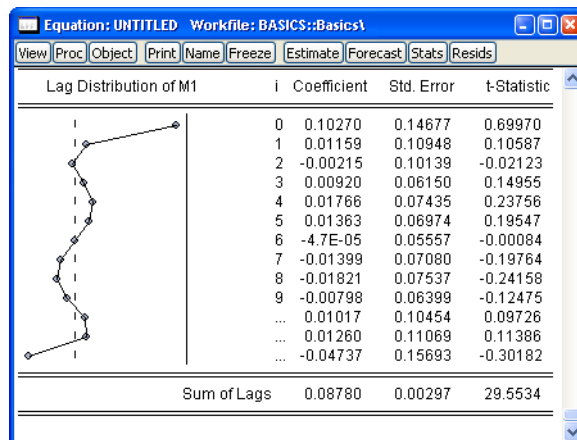
The following result is reported at the top of the equation window:

Dependent Variable: IP
 Method: Least Squares
 Date: 08/02/06 Time: 17:29
 Sample (adjusted): 1960M01 1989M12
 Included observations: 360 after adjustments

| | Coefficient | Std. Error | t-Statistic | Prob. |
|--------------------|-------------|-----------------------|-------------|--------|
| C | 40.67311 | 0.815195 | 49.89374 | 0.0000 |
| PDL01 | -4.66E-05 | 0.055566 | -0.000839 | 0.9993 |
| PDL02 | -0.015625 | 0.062884 | -0.248479 | 0.8039 |
| PDL03 | -0.000160 | 0.013909 | -0.011485 | 0.9908 |
| PDL04 | 0.001862 | 0.007700 | 0.241788 | 0.8091 |
| PDL05 | 2.58E-05 | 0.000408 | 0.063211 | 0.9496 |
| PDL06 | -4.93E-05 | 0.000180 | -0.273611 | 0.7845 |
| R-squared | 0.852371 | Mean dependent var | 71.72679 | |
| Adjusted R-squared | 0.849862 | S.D. dependent var | 19.53063 | |
| S.E. of regression | 7.567664 | Akaike info criterion | 6.904899 | |
| Sum squared resid | 20216.15 | Schwarz criterion | 6.980462 | |
| Log likelihood | -1235.882 | Hannan-Quinn criter. | 6.934944 | |
| F-statistic | 339.6882 | Durbin-Watson stat | 0.008026 | |
| Prob(F-statistic) | 0.000000 | | | |

This portion of the view reports the estimated coefficients γ of the polynomial in Equation (25.2) on page 23. The terms PDL01, PDL02, PDL03, ..., correspond to z_1, z_2, \dots in Equation (25.4).

The implied coefficients of interest β_j in equation (1) are reported at the bottom of the table, together with a plot of the estimated polynomial:



The Sum of Lags reported at the bottom of the table is the sum of the estimated coefficients on the distributed lag and has the interpretation of the long run effect of M1 on IP, assuming stationarity.

Note that selecting **View/Coefficient Tests** for an equation estimated with PDL terms tests the restrictions on γ , not on β . In this example, the coefficients on the fourth- (PDL05) and fifth-order (PDL06) terms are individually insignificant and very close to zero. To test the joint significance of these two terms, click **View/Coefficient Tests/Wald-Coefficient Restrictions...** and enter:

$$c(6)=0, \quad c(7)=0$$

in the Wald Test dialog box (see “[Wald Test \(Coefficient Restrictions\)](#)” on page 145 for an extensive discussion of Wald tests in EViews). EViews displays the result of the joint test:

| Wald Test: Equation: IP_PDL | | | |
|--|-----------|-----------|-------------|
| Test Statistic | Value | df | Probability |
| F-statistic | 0.039852 | (2, 353) | 0.9609 |
| Chi-square | 0.079704 | 2 | 0.9609 |
| Null Hypothesis Summary: | | | |
| Normalized Restriction (= 0) | Value | Std. Err. | |
| C(6) | 2.58E-05 | 2448.827 | |
| C(7) | -4.93E-05 | 5550.537 | |
| Restrictions are linear in coefficients. | | | |

There is no evidence to reject the null hypothesis, suggesting that you could have fit a lower order polynomial to your lag structure.

Automatic Categorical Dummy Variables

EViews equation specifications support expressions of the form:

$$@expand(ser1[, ser2, ser3, ...][, drop_spec])$$

When used in an equation specification, `@expand` creates a set of dummy variables that span the unique integer or string values of the input series.

For example consider the following two variables:

- SEX is a numeric series which takes the values 1 and 0.
- REGION is an alpha series which takes the values “North”, “South”, “East”, and “West”.

The equation list specification

$$income \text{ } age \text{ } @expand(sex)$$

is used to regress INCOME on the regressor AGE, and two dummy variables, one for “SEX = 0” and one for “SEX = 1”.

Similarly, the `@expand` statement in the equation list specification,

```
income @expand(sex, region) age
```

creates 8 dummy variables corresponding to :

```
sex = 0, region = "North"
sex = 0, region = "South"
sex = 0, region = "East"
sex = 0, region = "West"
sex = 1, region = "North"
sex = 1, region = "South"
sex = 1, region = "East"
sex = 1, region = "West"
```

Note that our two example equation specifications did not include an intercept. This is because the default `@expand` statements created a full set of dummy variables that would preclude including an intercept.

You may wish to drop one or more of the dummy variables. `@expand` takes several options for dropping variables.

The option `@dropfirst` specifies that the first category should be dropped so that:

```
@expand(sex, region, @dropfirst)
```

no dummy is created for “SEX = 0, REGION = “North””.

Similarly, `@droplast` specifies that the last category should be dropped. In:

```
@expand(sex, region, @droplast)
```

no dummy is created for “SEX = 1, REGION = “WEST””.

You may specify the dummy variables to be dropped, explicitly, using the syntax `@drop(val1[, val2, val3,...])`, where each argument specified corresponds to a successive category in `@expand`. For example, in the expression:

```
@expand(sex, region, @drop(0, "West"), @drop(1, "North"))
```

no dummy is created for “SEX = 0, REGION = “West”” and “SEX = 1, REGION = “North””.

When you specify drops by explicit value you may use the wild card “*” to indicate all values of a corresponding category. For example:

```
@expand(sex, region, @drop(1, *))
```

specifies that dummy variables for all values of REGION where “SEX = 1” should be dropped.

We caution you to take some care in using `@expand` since it is very easy to generate excessively large numbers of regressors.

`@expand` may also be used as part of a general mathematical expression, for example, in interactions with another variable as in:

```
2*@expand(x)
log(x+y)*@expand(z)
a*@expand(x)/b
```

Also useful is the ability to renormalize the dummies

```
@expand(x) - .5
```

Somewhat less useful (at least its uses may not be obvious) but supported are cases like:

```
log(x+y*@expand(z))
(@expand(x) - @expand(y))
```

As with all expressions included on an estimation or group creation command line, they should be enclosed in parentheses if they contain spaces.

The following expressions are valid,

```
a*@expand(x)
(a * @expand(x))
```

while this last expression is not,

```
a * @expand(x)
```

Example

Following Wooldridge (2000, Example 3.9, p. 106), we regress the log median housing price, LPRICE, on a constant, the log of the amount of pollution (LNOX), and the average number of houses in the community, ROOMS, using data from Harrison and Rubinfeld (1978).

We expand the example to include a dummy variable for each value of the series RADIAL, representing an index for community access to highways. We use `@expand` to create the dummy variables of interest, with a list specification of:

```
lprice lnox rooms @expand(radial)
```

We deliberately omit the constant term C since the `@expand` creates a full set of dummy variables. The top portion of the results is depicted below:

Dependent Variable: LPRICE

Method: Least Squares

Date: 08/07/06 Time: 10:37

Sample: 1 506

Included observations: 506

| | Coefficient | Std. Error | t-Statistic | Prob. |
|-----------|-------------|------------|-------------|--------|
| LNOX | -0.487579 | 0.084998 | -5.736396 | 0.0000 |
| ROOMS | 0.284844 | 0.018790 | 15.15945 | 0.0000 |
| RADIAL=1 | 8.930255 | 0.205986 | 43.35368 | 0.0000 |
| RADIAL=2 | 9.030875 | 0.209225 | 43.16343 | 0.0000 |
| RADIAL=3 | 9.085988 | 0.199781 | 45.47970 | 0.0000 |
| RADIAL=4 | 8.960967 | 0.198646 | 45.11016 | 0.0000 |
| RADIAL=5 | 9.110542 | 0.209759 | 43.43330 | 0.0000 |
| RADIAL=6 | 9.001712 | 0.205166 | 43.87528 | 0.0000 |
| RADIAL=7 | 9.013491 | 0.206797 | 43.58621 | 0.0000 |
| RADIAL=8 | 9.070626 | 0.214776 | 42.23297 | 0.0000 |
| RADIAL=24 | 8.811812 | 0.217787 | 40.46069 | 0.0000 |

Note that EViews has automatically created dummy variable expressions for each distinct value in RADIAL. If we wish to renormalize our dummy variables with respect to a different omitted category, we may include the C in the regression list, and explicitly exclude a value. For example, to exclude the category RADIAL = 24, we use the list:

```
lprice c lnox rooms @expand(radial, @drop(24))
```

Estimation of this specification yields:

Dependent Variable: LPRICE

Method: Least Squares

Date: 08/07/06 Time: 10:39

Sample: 1 506

Included observations: 506

| | Coefficient | Std. Error | t-Statistic | Prob. |
|--------------------|-------------|-----------------------|-------------|--------|
| C | 8.811812 | 0.217787 | 40.46069 | 0.0000 |
| LNOX | -0.487579 | 0.084998 | -5.736396 | 0.0000 |
| ROOMS | 0.284844 | 0.018790 | 15.15945 | 0.0000 |
| RADIAL=1 | 0.118444 | 0.072129 | 1.642117 | 0.1012 |
| RADIAL=2 | 0.219063 | 0.066055 | 3.316398 | 0.0010 |
| RADIAL=3 | 0.274176 | 0.059458 | 4.611253 | 0.0000 |
| RADIAL=4 | 0.149156 | 0.042649 | 3.497285 | 0.0005 |
| RADIAL=5 | 0.298730 | 0.037827 | 7.897337 | 0.0000 |
| RADIAL=6 | 0.189901 | 0.062190 | 3.053568 | 0.0024 |
| RADIAL=7 | 0.201679 | 0.077635 | 2.597794 | 0.0097 |
| RADIAL=8 | 0.258814 | 0.066166 | 3.911591 | 0.0001 |
| R-squared | 0.573871 | Mean dependent var | 9.941057 | |
| Adjusted R-squared | 0.565262 | S.D. dependent var | 0.409255 | |
| S.E. of regression | 0.269841 | Akaike info criterion | 0.239530 | |
| Sum squared resid | 36.04295 | Schwarz criterion | 0.331411 | |
| Log likelihood | -49.60111 | Hannan-Quinn criter. | 0.275566 | |
| F-statistic | 66.66195 | Durbin-Watson stat | 0.671010 | |
| Prob(F-statistic) | 0.000000 | | | |

Weighted Least Squares

Suppose that you have heteroskedasticity of known form, and that there is a series w , whose values are proportional to the *reciprocals* of the error standard deviations. You can use weighted least squares, with weight series w , to correct for the heteroskedasticity.

EViews performs weighted least squares by first dividing the weight series by its mean, then multiplying all of the data for each observation by the scaled weight series. The scaling of the weight series is a normalization that has no effect on the parameter results, but makes the weighted residuals more comparable to the unweighted residuals. The normalization does imply, however, that EViews weighted least squares is not appropriate in situations where the scale of the weight series is relevant, as in frequency weighting.

Estimation is then completed by running a regression using the weighted dependent and independent variables to minimize the sum-of-squared residuals:

$$S(\beta) = \sum_t w_t^2 (y_t - x_t' \beta)^2 \quad (25.8)$$

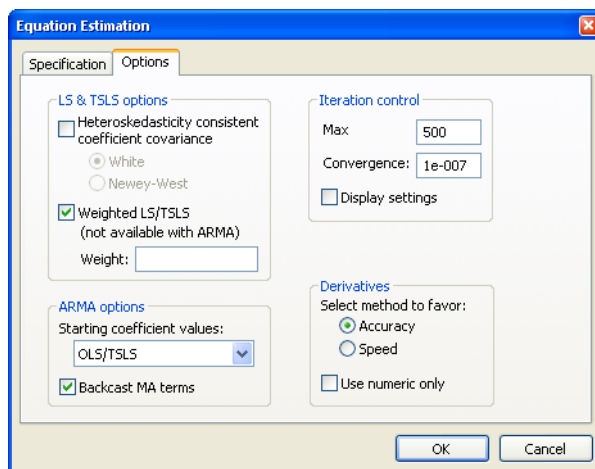
with respect to the k -dimensional vector of parameters β . In matrix notation, let W be a diagonal matrix containing the scaled w along the diagonal and zeroes elsewhere, and let y and X be the usual matrices associated with the left and right-hand side variables. The weighted least squares estimator is,

$$b_{WLS} = (X' W' W X)^{-1} X' W' W y, \quad (25.9)$$

and the estimated covariance matrix is:

$$\hat{\Sigma}_{WLS} = s^2 (X' W' W X)^{-1}. \quad (25.10)$$

To estimate an equation using weighted least squares, first go to the main menu and select **Quick/Estimate Equation...**, then choose **LS—Least Squares (NLS and ARMA)** from the combo box. Enter your equation specification and sample in the **Specification** tab, then select the **Options** tab and click on the **Weighted LS/TSLS** option.



Fill in the blank after **Weight** with the name of the series containing your weights, and click on **OK**. Click on **OK** again to accept the dialog and estimate the equation.

Dependent Variable: LOG(X)
 Method: Least Squares
 Date: 08/07/06 Time: 17:05
 Sample (adjusted): 1891 1983
 Included observations: 93 after adjustments
 Weighting series: POP

| | Coefficient | Std. Error | t-Statistic | Prob. |
|-----------------------|-------------|-----------------------|-------------|--------|
| C | 0.004233 | 0.012745 | 0.332092 | 0.7406 |
| LOG(X(-1)) | 0.099840 | 0.112539 | 0.887163 | 0.3774 |
| LOG(W(-1)) | 0.194219 | 0.421005 | 0.461322 | 0.6457 |
| Weighted Statistics | | | | |
| R-squared | 0.017150 | Mean dependent var | 0.009762 | |
| Adjusted R-squared | -0.004691 | S.D. dependent var | 0.106487 | |
| S.E. of regression | 0.106785 | Akaike info criterion | -1.604274 | |
| Sum squared resid | 1.026272 | Schwarz criterion | -1.522577 | |
| Log likelihood | 77.59873 | Hannan-Quinn criter. | -1.571287 | |
| F-statistic | 0.785202 | Durbin-Watson stat | 1.948087 | |
| Prob(F-statistic) | 0.459126 | | | |
| Unweighted Statistics | | | | |
| R-squared | -0.002922 | Mean dependent var | 0.011093 | |
| Adjusted R-squared | -0.025209 | S.D. dependent var | 0.121357 | |
| S.E. of regression | 0.122877 | Sum squared resid | 1.358893 | |
| Durbin-Watson stat | 2.086669 | | | |

EViews will open an output window displaying the standard coefficient results, and both weighted and unweighted summary statistics. The weighted summary statistics are based on the fitted residuals, computed using the weighted data:

$$\tilde{u}_t = w_t(y_t - x_t' b_{WLS}). \quad (25.11)$$

The unweighted summary results are based on the residuals computed from the original (unweighted) data:

$$u_t = y_t - x_t' b_{WLS}. \quad (25.12)$$

Following estimation, the unweighted residuals are placed in the RESID series.

If the residual variance assumptions are correct, the weighted residuals should show no evidence of heteroskedasticity. If the variance assumptions are correct, the unweighted residuals should be heteroskedastic, with the reciprocal of the standard deviation of the residual at each period t being proportional to w_t .

The weighting option will be ignored in equations containing ARMA specifications. Note also that the weighting option is not available for binary, count, censored and truncated, or ordered discrete choice models.

Heteroskedasticity and Autocorrelation Consistent Covariances

When the form of heteroskedasticity is not known, it may not be possible to obtain efficient estimates of the parameters using weighted least squares. OLS provides consistent parameter estimates in the presence of heteroskedasticity, but the usual OLS standard errors will be incorrect and should not be used for inference.

Before we describe the techniques for HAC covariance estimation, note that:

- Using the White heteroskedasticity consistent or the Newey-West HAC consistent covariance estimates does not change the point estimates of the parameters, only the estimated standard errors.
- There is nothing to keep you from combining various methods of accounting for heteroskedasticity and serial correlation. For example, weighted least squares estimation might be accompanied by White or Newey-West covariance matrix estimates.

Heteroskedasticity Consistent Covariances (White)

White (1980) has derived a heteroskedasticity consistent covariance matrix estimator which provides correct estimates of the coefficient covariances in the presence of heteroskedasticity of unknown form. The White covariance matrix is given by:

$$\hat{\Sigma}_W = \frac{T}{T-k} (X'X)^{-1} \left(\sum_{t=1}^T u_t^2 x_t x_t' \right) (X'X)^{-1}, \quad (25.13)$$

where T is the number of observations, k is the number of regressors, and u_t is the least squares residual.

EViews provides you the option to use the White covariance estimator in place of the standard OLS formula. Open the equation dialog and specify the equation as before, then push the **Options** button. Next, click on the check box labeled **Heteroskedasticity Consistent Covariance** and click on the **White** radio button. Accept the options and click **OK** to estimate the equation.

EViews will estimate your equation and compute the variances using White's covariance estimator. You can always tell when EViews is using White covariances, since the output display will include a line to document this fact:

Dependent Variable: LOG(X)
 Method: Least Squares
 Date: 08/07/06 Time: 17:08
 Sample (adjusted): 1891 1983
 Included observations: 93 after adjustments
 Weighting series: POP
 White Heteroskedasticity-Consistent Standard Errors & Covariance

| | Coefficient | Std. Error | t-Statistic | Prob. |
|------------|-------------|------------|-------------|--------|
| C | 0.004233 | 0.012519 | 0.338088 | 0.7361 |
| LOG(X(-1)) | 0.099840 | 0.137262 | 0.727369 | 0.4689 |
| LOG(W(-1)) | 0.194219 | 0.436644 | 0.444800 | 0.6575 |

HAC Consistent Covariances (Newey-West)

The White covariance matrix described above assumes that the residuals of the estimated equation are serially uncorrelated. Newey and West (1987b) have proposed a more general covariance estimator that is consistent in the presence of both heteroskedasticity and autocorrelation of unknown form. The Newey-West estimator is given by,

$$\hat{\Sigma}_{NW} = \frac{T}{T-k} (X'X)^{-1} \hat{\Omega} (X'X)^{-1}, \quad (25.14)$$

where:

$$\begin{aligned} \hat{\Omega} = & \frac{T}{T-k} \left\{ \sum_{t=1}^T u_t^2 x_t x_t' \right. \\ & \left. + \sum_{v=1}^q \left(\left(1 - \frac{v}{q+1} \right) \sum_{t=v+1}^T (x_t u_t u_{t-v} x_{t-v}' + x_{t-v} u_{t-v} u_t x_t') \right) \right\} \end{aligned} \quad (25.15)$$

and q , the truncation lag, is a parameter representing the number of autocorrelations used in evaluating the dynamics of the OLS residuals u_t . Following the suggestion of Newey and West, EViews sets q using the formula:

$$q = \text{floor}(4(T/100)^{2/9}). \quad (25.16)$$

To use the Newey-West method, select the **Options** tab in the **Equation Estimation**. Check the box labeled **Heteroskedasticity Consistent Covariance** and press the **Newey-West** radio button.

Two-stage Least Squares

A fundamental assumption of regression analysis is that the right-hand side variables are uncorrelated with the disturbance term. If this assumption is violated, both OLS and weighted LS are biased and inconsistent.

There are a number of situations where some of the right-hand side variables are correlated with disturbances. Some classic examples occur when:

- There are endogenously determined variables on the right-hand side of the equation.
- Right-hand side variables are measured with error.

For simplicity, we will refer to variables that are correlated with the residuals as *endogenous*, and variables that are not correlated with the residuals as *exogenous* or *predetermined*.

The standard approach in cases where right-hand side variables are correlated with the residuals is to estimate the equation using *instrumental variables* regression. The idea behind instrumental variables is to find a set of variables, termed *instruments*, that are both (1) correlated with the explanatory variables in the equation, and (2) uncorrelated with the disturbances. These instruments are used to eliminate the correlation between right-hand side variables and the disturbances.

Two-stage least squares (TSLS) is a special case of instrumental variables regression. As the name suggests, there are two distinct stages in two-stage least squares. In the first stage, TSLS finds the portions of the endogenous and exogenous variables that can be attributed to the instruments. This stage involves estimating an OLS regression of each variable in the model on the set of instruments. The second stage is a regression of the original equation, with all of the variables replaced by the fitted values from the first-stage regressions. The coefficients of this regression are the TSLS estimates.

You need not worry about the separate stages of TSLS since EViews will estimate both stages simultaneously using instrumental variables techniques. More formally, let Z be the matrix of instruments, and let y and X be the dependent and explanatory variables. Then the coefficients computed in two-stage least squares are given by,

$$b_{TSLS} = (X'Z(Z'Z)^{-1}Z'X)^{-1}X'Z(Z'Z)^{-1}Z'y, \quad (25.17)$$

and the estimated covariance matrix of these coefficients is given by:

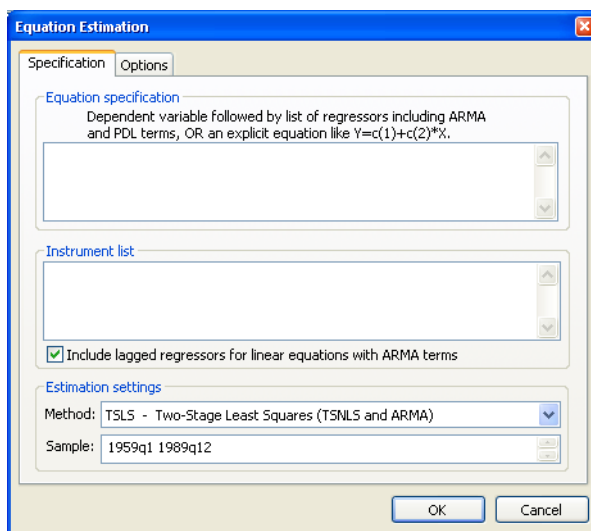
$$\hat{\Sigma}_{TSLS} = s^2(X'Z(Z'Z)^{-1}Z'X)^{-1}, \quad (25.18)$$

where s^2 is the estimated residual variance (square of the standard error of the regression).

Estimating TSLS in EViews

To use two-stage least squares, open the equation specification box by choosing **Object/New Object.../Equation...** or **Quick/Estimate Equation...**. Choose **TSLS** from the **Method**: combo box and the dialog will change to include an edit window where you will list the instruments.

In the **Equation specification** edit box, specify your dependent variable and independent variables, and in the **Instrument list** edit box, provide a list of instruments.



There are a few things to keep in mind as you enter your instruments:

- In order to calculate TSLS estimates, your specification must satisfy the *order condition* for identification, which says that there must be at least as many instruments as there are coefficients in your equation. There is an additional rank condition which must also be satisfied. See Davidson and MacKinnon (1993) and Johnston and DiNardo (1997) for additional discussion.
- For econometric reasons that we will not pursue here, any right-hand side variables that are not correlated with the disturbances should be included as instruments.
- The constant, C, is always a suitable instrument, so EViews will add it to the instrument list if you omit it.

For example, suppose you are interested in estimating a consumption equation relating consumption (CONS) to gross domestic product (GDP), lagged consumption (CONS(-1)), a trend variable (TIME) and a constant (C). GDP is endogenous and therefore correlated with the residuals. You may, however, believe that government expenditures (G), the log of the money supply (LM), lagged consumption, TIME, and C, are exogenous and uncorrelated with the disturbances, so that these variables may be used as instruments. Your equation specification is then,

```
cons c gdp cons(-1) time
```

and the instrument list is:

```
c gov cons(-1) time lm
```

This specification satisfies the order condition for identification, which requires that there are at least as many instruments (five) as there are coefficients (four) in the equation specification.

Furthermore, all of the variables in the consumption equation that are believed to be uncorrelated with the disturbances, (CONS(-1), TIME, and C), appear both in the equation specification and in the instrument list. Note that listing C as an instrument is redundant, since EViews automatically adds it to the instrument list.

Output from TSLS

Below, we present TSLS estimates from a regression of LOG(CS) on a constant and LOG(GDP), with the instrument list “C LOG(CS(-1)) LOG(GDP(-1))”:

| Dependent Variable: LOG(CS) | | | | |
|--|-------------|--------------------|-------------|--------|
| Method: Two-Stage Least Squares | | | | |
| Date: 08/07/06 Time: 13:56 | | | | |
| Sample (adjusted): 1947Q2 1995Q1 | | | | |
| Included observations: 192 after adjustments | | | | |
| Instrument list: C LOG(CS(-1)) LOG(GDP(-1)) | | | | |
| | Coefficient | Std. Error | t-Statistic | Prob. |
| C | -1.209268 | 0.039151 | -30.88699 | 0.0000 |
| LOG(GDP) | 1.094339 | 0.004924 | 222.2597 | 0.0000 |
| R-squared | 0.996168 | Mean dependent var | 7.480286 | |
| Adjusted R-squared | 0.996148 | S.D. dependent var | 0.462990 | |
| S.E. of regression | 0.028735 | Sum squared resid | 0.156888 | |
| F-statistic | 49399.36 | Durbin-Watson stat | 0.102639 | |
| Prob(F-statistic) | 0.000000 | Second-Stage SSR | 0.152396 | |

EViews identifies the estimation procedure, as well as the list of instruments in the header. This information is followed by the usual coefficient, t -statistics, and asymptotic p -values.

The summary statistics reported at the bottom of the table are computed using the formulas outlined in “[Summary Statistics](#)” on page 13. Bear in mind that all reported statistics are only asymptotically valid. For a discussion of the finite sample properties of TSLS, see Johnston and DiNardo (1997, p. 355–358) or Davidson and MacKinnon (1993, p. 221–224).

EViews uses the *structural residuals* $u_t = y_t - x_t' b_{TSLS}$ in calculating all of the summary statistics. For example, the standard error of the regression used in the asymptotic covariance calculation is computed as:

$$s^2 = \sum_t u_t^2 / (T - k). \quad (25.19)$$

These structural residuals should be distinguished from the *second stage residuals* that you would obtain from the second stage regression if you actually computed the two-stage least squares estimates in two separate stages. The second stage residuals are given by $\tilde{u}_t = \hat{y}_t - \hat{x}_t' b_{TSLs}$, where the \hat{y}_t and \hat{x}_t are the fitted values from the first-stage regressions.

We caution you that some of the reported statistics should be interpreted with care. For example, since different equation specifications will have different instrument lists, the reported R^2 for TSLS can be negative even when there is a constant in the equation.

Weighted TSLS

You can combine TSLS with weighted regression. Simply enter your TSLS specification as above, then press the **Options** button, select the **Weighted LS/TSLS** option, and enter the weighting series.

Weighted two-stage least squares is performed by multiplying all of the data, including the instruments, by the weight variable, and estimating TSLS on the transformed model. Equivalently, EViews then estimates the coefficients using the formula,

$$b_{WTSLs} = (X' W' WZ(Z' W' WZ)^{-1} Z' W' WX)^{-1} \cdot X' W' WZ(Z' W' WZ)^{-1} Z' W' Wy \quad (25.20)$$

The estimated covariance matrix is:

$$\hat{\Sigma}_{WTSLs} = s^2 (X' W' WZ(Z' W' WZ)^{-1} Z' W' WX)^{-1}. \quad (25.21)$$

TSLS with AR errors

You can adjust your TSLS estimates to account for serial correlation by adding AR terms to your equation specification. EViews will automatically transform the model to a nonlinear least squares problem, and estimate the model using instrumental variables. Details of this procedure may be found in Fair (1984, p. 210–214). The output from TSLS with an AR(1) specification looks as follows:

Dependent Variable: LOG(CS)
 Method: Two-Stage Least Squares
 Date: 08/07/06 Time: 13:58
 Sample (adjusted): 1947Q2 1995Q1
 Included observations: 192 after adjustments
 Convergence achieved after 4 iterations
 Instrument list: C LOG(CS(-1)) LOG(GDP(-1))
 Lagged dependent
 variable & regressors
 added to instrument list

| | Coefficient | Std. Error | t-Statistic | Prob. |
|--------------------|-------------|--------------------|-------------|--------|
| C | -1.420705 | 0.203266 | -6.989390 | 0.0000 |
| LOG(GDP) | 1.119858 | 0.025116 | 44.58783 | 0.0000 |
| AR(1) | 0.930900 | 0.022267 | 41.80595 | 0.0000 |
| R-squared | 0.999611 | Mean dependent var | 7.480286 | |
| Adjusted R-squared | 0.999607 | S.D. dependent var | 0.462990 | |
| S.E. of regression | 0.009175 | Sum squared resid | 0.015909 | |
| F-statistic | 243139.7 | Durbin-Watson stat | 1.931027 | |
| Prob(F-statistic) | 0.000000 | Second-Stage SSR | 0.010799 | |
| Inverted AR Roots | .93 | | | |

The **Options** button in the estimation box may be used to change the iteration limit and convergence criterion for the nonlinear instrumental variables procedure.

First-order AR errors

Suppose your specification is:

$$\begin{aligned} y_t &= x_t' \beta + w_t' \gamma + u_t \\ u_t &= \rho_1 u_{t-1} + \epsilon_t \end{aligned} \quad (25.22)$$

where x_t is a vector of endogenous variables, and w_t is a vector of predetermined variables, which, in this context, may include lags of the dependent variable. z_t is a vector of instrumental variables not in w_t that is large enough to identify the parameters of the model.

In this setting, there are important technical issues to be raised in connection with the choice of instruments. In a widely quoted result, Fair (1970) shows that if the model is estimated using an iterative Cochrane-Orcutt procedure, all of the lagged left- and right-hand side variables (y_{t-1} , x_{t-1} , w_{t-1}) must be included in the instrument list to obtain consistent estimates. In this case, then the instrument list should include:

$$(w_t, z_t, y_{t-1}, x_{t-1}, w_{t-1}) . \quad (25.23)$$

Despite the fact the EViews estimates the model as a nonlinear regression model, the first stage instruments in TSLS are formed as if running Cochrane-Orcutt. Thus, if you choose to omit the lagged left- and right-hand side terms from the instrument list, EViews will automatically add each of the lagged terms as instruments. This fact is noted in your output.

Higher Order AR errors

The AR(1) result extends naturally to specifications involving higher order serial correlation. For example, if you include a single AR(4) term in your model, the natural instrument list will be:

$$(w_t, z_t, y_{t-4}, x_{t-4}, w_{t-4}) \quad (25.24)$$

If you include AR terms from 1 through 4, one possible instrument list is:

$$(w_t, z_t, y_{t-1}, \dots, y_{t-4}, x_{t-1}, \dots, x_{t-4}, w_{t-1}, \dots, w_{t-4}) \quad (25.25)$$

Note that while theoretically valid, this instrument list has a large number of overidentifying instruments, which may lead to computational difficulties and large finite sample biases (Fair (1984, p. 214), Davidson and MacKinnon (1993, p. 222-224)). In theory, adding instruments should always improve your estimates, but as a practical matter this may not be so in small samples.

Examples

Suppose that you wish to estimate the consumption function by two-stage least squares, allowing for first-order serial correlation. You may then use two-stage least squares with the variable list,

```
cons c gdp ar(1)
```

and instrument list:

```
c gov log(m1) time cons(-1) gdp(-1)
```

Notice that the lags of both the dependent and endogenous variables (CONS(-1) and GDP(-1)), are included in the instrument list.

Similarly, consider the consumption function:

```
cons c cons(-1) gdp ar(1)
```

A valid instrument list is given by:

```
c gov log(m1) time cons(-1) cons(-2) gdp(-1)
```

Here we treat the lagged left and right-hand side variables from the original specification as predetermined and add the lagged values to the instrument list.

Lastly, consider the specification:

```
cons c gdp ar(1) ar(2) ar(3) ar(4)
```


Adding all of the relevant instruments in the list, we have:

```
c gov log(ml) time cons(-1) cons(-2) cons(-3) cons(-4) gdp(-1)
  gdp(-2) gdp(-3) gdp(-4)
```

TSLS with MA errors

You can also estimate two-stage least squares variable problems with MA error terms of various orders. To account for the presence of MA errors, simply add the appropriate terms to your specification prior to estimation.

Illustration

Suppose that you wish to estimate the consumption function by two-stage least squares, accounting for first-order moving average errors. You may then use two-stage least squares with the variable list,

```
cons c gdp ma(1)
```

and instrument list:

```
c gov log(ml) time
```

EViews will add both first and second lags of CONS and GDP to the instrument list.

Technical Details

Most of the technical details are identical to those outlined above for AR errors. EViews transforms the model that is nonlinear in parameters (employing backcasting, if appropriate) and then estimates the model using nonlinear instrumental variables techniques.

Note that EViews augments the instrument list by adding lagged dependent and regressor variables. Note however, that each MA term involves an infinite number of AR terms. Clearly, it is impossible to add an infinite number of lags to the instrument list, so that EViews performs an ad hoc approximation by adding a truncated set of instruments involving the MA order and an additional lag. If for example, you have an MA(5), EViews will add lagged instruments corresponding to lags 5 and 6.

Nonlinear Least Squares

Suppose that we have the regression specification:

$$y_t = f(x_t, \beta) + \epsilon_t, \quad (25.26)$$

where f is a general function of the explanatory variables x_t and the parameters β . Least squares estimation chooses the parameter values that minimize the sum of squared residuals:

$$S(\beta) = \sum_t (y_t - f(x_t, \beta))^2 = (y - f(X, \beta))'(y - f(X, \beta)) \quad (25.27)$$

We say that a model is *linear in parameters* if the derivatives of f with respect to the parameters do not depend upon β ; if the derivatives are functions of β , we say that the model is *nonlinear in parameters*.

For example, consider the model given by:

$$y_t = \beta_1 + \beta_2 \log L_t + \beta_3 \log K_t + \epsilon_t. \quad (25.28)$$

It is easy to see that this model is linear in its parameters, implying that it can be estimated using ordinary least squares.

In contrast, the equation specification:

$$y_t = \beta_1 L_t^{\beta_2} K_t^{\beta_3} + \epsilon_t \quad (25.29)$$

has derivatives that depend upon the elements of β . There is no way to rearrange the terms in this model so that ordinary least squares can be used to minimize the sum-of-squared residuals. We must use nonlinear least squares techniques to estimate the parameters of the model.

Nonlinear least squares minimizes the sum-of-squared residuals with respect to the choice of parameters β . While there is no closed form solution for the parameter estimates, the estimates satisfy the first-order conditions:

$$(G(\beta))'(y - f(X, \beta)) = 0, \quad (25.30)$$

where $G(\beta)$ is the matrix of first derivatives of $f(X, \beta)$ with respect to β (to simplify notation we suppress the dependence of G upon X). The estimated covariance matrix is given by:

$$\hat{\Sigma}_{NLLS} = s^2 (G(b_{NLLS})' G(b_{NLLS}))^{-1}. \quad (25.31)$$

where b_{NLLS} are the estimated parameters. For additional discussion of nonlinear estimation, see Pindyck and Rubinfeld (1998, p. 265-273) or Davidson and MacKinnon (1993).

Estimating NLS Models in EViews

It is easy to tell EViews that you wish to estimate the parameters of a model using nonlinear least squares. EViews automatically applies nonlinear least squares to any regression equation that is nonlinear in its coefficients. Simply select **Object/New Object.../Equation**, enter the equation in the equation specification dialog box, and click **OK**. EViews will do all of the work of estimating your model using an iterative algorithm.

A full technical discussion of iterative estimation procedures is provided in [Appendix D. "Estimation and Solution Options," beginning on page 625.](#)

Specifying Nonlinear Least Squares

For nonlinear regression models, you will have to enter your specification in equation form using EViews expressions that contain direct references to coefficients. You may use elements of the default coefficient vector C (e.g. $C(1)$, $C(2)$, $C(34)$, $C(87)$), or you can define and use other coefficient vectors. For example:

$$y = c(1) + c(2) * (k^{c(3)} + l^{c(4)})$$

is a nonlinear specification that uses the first through the fourth elements of the default coefficient vector, C .

To create a new coefficient vector, select **Object/New Object.../Matrix-Vector-Coef** in the main menu and provide a name. You may now use this coefficient vector in your specification. For example, if you create a coefficient vector named CF , you can rewrite the specification above as:

$$y = cf(11) + cf(12) * (k^{cf(13)} + l^{cf(14)})$$

which uses the eleventh through the fourteenth elements of CF .

You can also use multiple coefficient vectors in your specification:

$$y = c(11) + c(12) * (k^{cf(1)} + l^{cf(2)})$$

which uses both C and CF in the specification.

It is worth noting that EViews implicitly adds an additive disturbance to your specification. For example, the input

$$y = (c(1) * x + c(2) * z + 4)^2$$

is interpreted as $y_t = (c(1)x_t + c(2)z_t + 4)^2 + \epsilon_t$, and EViews will minimize:

$$S(c(1), c(2)) = \sum_t (y_t - (c(1)x_t + c(2)z_t + 4)^2)^2 \quad (25.32)$$

If you wish, the equation specification may be given by a simple expression that does not include a dependent variable. For example, the input,

$$(c(1) * x + c(2) * z + 4)^2$$

is interpreted by EViews as $-(c(1)x_t + c(2)z_t + 4)^2 = \epsilon_t$, and EViews will minimize:

$$S(c(1), c(2)) = \sum_t (-(c(1)x_t + c(2)z_t + 4)^2)^2 \quad (25.33)$$

While EViews will estimate the parameters of this last specification, the equation cannot be used for forecasting and cannot be included in a model. This restriction also holds for any equation that includes coefficients to the left of the equal sign. For example, if you specify,

$$x + c(1) * y = z^{c(2)}$$

EViews will find the values of $C(1)$ and $C(2)$ that minimize the sum of squares of the implicit equation:

$$x_t + c(1)y_t - z_t^{c(2)} = \epsilon_t \quad (25.34)$$

The estimated equation cannot be used in forecasting or included in a model, since there is no dependent variable.

Estimation Options

Starting Values. Iterative estimation procedures require starting values for the coefficients of the model. There are no general rules for selecting starting values for parameters. The closer to the true values the better, so if you have reasonable guesses for parameter values, these can be useful. In some cases, you can obtain good starting values by estimating a restricted version of the model using least squares. In general, however, you will have to experiment in order to find starting values.

EViews uses the values in the coefficient vector at the time you begin the estimation procedure as starting values for the iterative procedure. It is easy to examine and change these coefficient starting values.

To see the starting values, double click on the coefficient vector in the workfile directory. If the values appear to be reasonable, you can close the window and proceed with estimating your model.

If you wish to change the starting values, first make certain that the spreadsheet view of your coefficients is in edit mode, then enter the coefficient values. When you are finished setting the initial values, close the coefficient vector window and estimate your model.

You may also set starting coefficient values from the command window using the `PARAM` command. Simply enter the `PARAM` keyword, following by each coefficient and desired value:

```
param c(1) 153 c(2) .68 c(3) .15
```

sets $C(1) = 153$, $C(2) = .68$, and $C(3) = .15$.

See [Appendix D, “Estimation and Solution Options” on page 625](#), for further details.

Derivative Methods. Estimation in EViews requires computation of the derivatives of the regression function with respect to the parameters. EViews provides you with the option of computing analytic expressions for these derivatives (if possible), or computing finite difference numeric derivatives in cases where the derivative is not constant. Furthermore, if numeric derivatives are computed, you can choose whether to favor speed of computation (fewer function evaluations) or whether to favor accuracy (more function evaluations). Additional issues associated with ARIMA models are discussed in [“Estimation Options” on page 79](#).

Iteration and Convergence Options. You can control the iterative process by specifying convergence criterion and the maximum number of iterations. Press the **Options** button in the equation dialog box and enter the desired values.

EViews will report that the estimation procedure has converged if the convergence test value is below your convergence tolerance. See [“Iteration and Convergence Options” on page 627](#) for details.

In most cases, you will not need to change the maximum number of iterations. However, for some difficult to estimate models, the iterative procedure will not converge within the maximum number of iterations. If your model does not converge within the allotted number of iterations, simply click on the **Estimate** button, and, if desired, increase the maximum number of iterations. Click on **OK** to accept the options, and click on **OK** to begin estimation. EViews will start estimation using the last set of parameter values as starting values.

These options may also be set from the global options dialog. See [Appendix B, “Estimation Defaults” on page 767](#).

Output from NLS

Once your model has been estimated, EViews displays an equation output screen showing the results of the nonlinear least squares procedure. Below is the output from a regression of LOG(CS) on C, and the Box-Cox transform of GDP:

| Dependent Variable: LOG(CS) | | | | |
|--|-------------|-----------------------|-------------|--------|
| Method: Least Squares | | | | |
| Date: 08/07/06 Time: 14:39 | | | | |
| Sample: 1947Q1 1995Q1 | | | | |
| Included observations: 193 | | | | |
| Convergence achieved after 52 iterations | | | | |
| LOG(CS) = C(1)+ C(2)*(GDP^C(3)-1)/C(3) | | | | |
| Variable | Coefficient | Std. Error | t-Statistic | Prob. |
| C(1) | 2.769341 | 0.286679 | 9.660072 | 0.0000 |
| C(2) | 0.269884 | 0.043126 | 6.258035 | 0.0000 |
| C(3) | 0.177070 | 0.020194 | 8.768404 | 0.0000 |
| R-squared | 0.997253 | Mean dependent var | 7.476058 | |
| Adjusted R-squared | 0.997224 | S.D. dependent var | 0.465503 | |
| S.E. of regression | 0.024527 | Akaike info criterion | -4.562688 | |
| Sum squared resid | 0.114296 | Schwarz criterion | -4.511973 | |
| Log likelihood | 443.2994 | Hannan-Quinn criter. | -4.542150 | |
| F-statistic | 34486.03 | Durbin-Watson stat | 0.134844 | |
| Prob(F-statistic) | 0.000000 | | | |

If the estimation procedure has converged, EViews will report this fact, along with the number of iterations that were required. If the iterative procedure did not converge, EViews will report “Convergence not achieved after” followed by the number of iterations attempted.

Below the line describing convergence, EViews will repeat the nonlinear specification so that you can easily interpret the estimated coefficients of your model.

EViews provides you with all of the usual summary statistics for regression models. Provided that your model has converged, the standard statistical results and tests are *asymptotically* valid.

Weighted NLS

Weights can be used in nonlinear estimation in a manner analogous to weighted linear least squares. To estimate an equation using weighted nonlinear least squares, enter your specification, press the **Options** button and click on the **Weighted LS/TLS** option. Fill in the blank after **Weight:** with the name of the weight series and then estimate the equation.

EViews minimizes the sum of the weighted squared residuals:

$$S(\beta) = \sum_t w_t^2 (y_t - f(x_t, \beta))^2 = (y - f(X, \beta))' W' W (y - f(X, \beta)) \quad (25.35)$$

with respect to the parameters β , where w_t are the values of the weight series and W is the matrix of weights. The first-order conditions are given by,

$$(G(\beta))' W' W (y - f(X, \beta)) = 0 \quad (25.36)$$

and the covariance estimate is computed as:

$$\hat{\Sigma}_{WNLLS} = s^2 (G(b_{WNLLS})' W' W G(b_{WNLLS}))^{-1}. \quad (25.37)$$

NLS with AR errors

EViews will estimate nonlinear regression models with autoregressive error terms. Simply select **Object/New Object.../Equation...** or **Quick/Estimate Equation...** and specify your model using EViews expressions, followed by an additive term describing the AR correction enclosed in square brackets. The AR term should consist of a coefficient assignment for each AR term, separated by commas. For example, if you wish to estimate,

$$\begin{aligned} CS_t &= c_1 + GDP_t^{c_2} + u_t \\ u_t &= c_3 u_{t-1} + c_4 u_{t-2} + \epsilon_t \end{aligned} \quad (25.38)$$

you should enter the specification:

$$cs = c(1) + gdp^{c(2)} + [ar(1)=c(3), ar(2)=c(4)]$$

See “How EViews Estimates AR Models,” on page 70 for additional details. EViews does not currently estimate nonlinear models with MA errors, nor does it estimate weighted models

with AR terms—if you add AR terms to a weighted nonlinear model, the weighting series will be ignored.

Nonlinear TSLS

Nonlinear two-stage least squares refers to an instrumental variables procedure for estimating nonlinear regression models involving functions of endogenous and exogenous variables and parameters. Suppose we have the usual nonlinear regression model:

$$y_t = f(x_t, \beta) + \epsilon_t, \quad (25.39)$$

where β is a k -dimensional vector of parameters, and x_t contains both exogenous and endogenous variables. In matrix form, if we have $m \geq k$ instruments z_t , nonlinear two-stage least squares minimizes:

$$S(\beta) = (y - f(X, \beta))' Z (Z' Z)^{-1} Z' (y - f(X, \beta)) \quad (25.40)$$

with respect to the choice of β .

While there is no closed form solution for the parameter estimates, the parameter estimates satisfy the first-order conditions:

$$G(\beta)' Z (Z' Z)^{-1} Z' (y - f(X, \beta)) = 0 \quad (25.41)$$

with estimated covariance given by:

$$\hat{\Sigma}_{TSNLLS} = s^2 (G(b_{TSNLLS})' Z (Z' Z)^{-1} Z' G(b_{TSNLLS}))^{-1}. \quad (25.42)$$

How to Estimate Nonlinear TSLS in EViews

EViews performs the estimation procedure in a single step so that you don't have to perform the separate stages yourself. Simply select **Object/New Object.../Equation...** or **Quick/Estimate Equation...** Choose **TSLS** from the **Method:** combo box, enter your nonlinear specification and the list of instruments. Click **OK**.

With nonlinear two-stage least squares estimation, you have a great deal of flexibility with your choice of instruments. Intuitively, you want instruments that are correlated with $G(\beta)$. Since G is nonlinear, you may begin to think about using more than just the exogenous and predetermined variables as instruments. Various nonlinear functions of these variables, for example, cross-products and powers, may also be valid instruments. One should be aware, however, of the possible finite sample biases resulting from using too many instruments.

Weighted Nonlinear Two-stage Least Squares

Weights can be used in nonlinear two-stage least squares estimation. Simply add weighting to your nonlinear TSLS specification above by pressing the **Options** button, selecting **Weighted LS/TSLS** option, and entering the name of the weight series.

The objective function for weighted TSLS is,

$$S(\beta) = (y - f(X, \beta))' W' WZ(Z' W' WZ)^{-1} Z' W' W(y - f(X, \beta)). \quad (25.43)$$

The reported standard errors are based on the covariance matrix estimate given by:

$$\hat{\Sigma}_{WTSNLLS} = s^2(G(b)' W' WZ(Z' W' WZ)^{-1} Z' W' WG(b))^{-1} \quad (25.44)$$

where $b \equiv b_{WTSNLLS}$. Note that if you add AR or MA terms to a weighted specification, the weighting series will be ignored.

Nonlinear Two-stage Least Squares with AR errors

While we will not go into much detail here, note that EViews can estimate non-linear TSLS models where there are autoregressive error terms. EViews does not currently estimate non-linear models with MA errors.

To estimate your model, simply open your equation specification window, and enter your nonlinear specification, including all AR terms, and provide your instrument list. For example, you could enter the regression specification:

```
cs = exp(c(1) + gdp^c(2)) + [ar(1)=c(3)]
```

with the instrument list:

```
c gov
```

EViews will transform the nonlinear regression model as described in [“Estimating AR Models” on page 67](#), and then estimate nonlinear TSLS on the transformed specification using the instruments C and GOV. For nonlinear models with AR errors, EViews uses a Gauss-Newton algorithm. See [“Optimization Algorithms” on page 629](#) for further details.

Solving Estimation Problems

EViews may not be able to estimate your nonlinear equation on the first attempt. Sometimes, the nonlinear least squares procedure will stop immediately. Other times, EViews may stop estimation after several iterations without achieving convergence. EViews might even report that it cannot improve the sums-of-squares. While there are no specific rules on how to proceed if you encounter these estimation problems, there are a few general areas you might want to examine.

Starting Values

If you experience problems with the very first iteration of a nonlinear procedure, the problem is almost certainly related to starting values. See the discussion above for how to examine and change your starting values.

Model Identification

If EViews goes through a number of iterations and then reports that it encounters a “Near Singular Matrix”, you should check to make certain that your model is identified. Models are said to be non-identified if there are multiple sets of coefficients which identically yield the minimized sum-of-squares value. If this condition holds, it is impossible to choose between the coefficients on the basis of the minimum sum-of-squares criterion.

For example, the nonlinear specification:

$$y_t = \beta_1\beta_2 + \beta_2^2x_t + \epsilon_t \quad (25.45)$$

is not identified, since any coefficient pair (β_1, β_2) is indistinguishable from the pair $(-\beta_1, -\beta_2)$ in terms of the sum-of-squared residuals.

For a thorough discussion of identification of nonlinear least squares models, see Davidson and MacKinnon (1993, Sections 2.3, 5.2 and 6.3).

Convergence Criterion

EViews may report that it is unable to improve the sums-of-squares. This result may be evidence of non-identification or model misspecification. Alternatively, it may be the result of setting your convergence criterion too low, which can occur if your nonlinear specification is particularly complex.

If you wish to change the convergence criterion, enter the new value in the **Options** tab. Be aware that increasing this value increases the possibility that you will stop at a local minimum, and may hide misspecification or non-identification of your model.

See [“Setting Estimation Options” on page 625](#), for related discussion.

Generalized Method of Moments (GMM)

The starting point of GMM estimation is a theoretical relation that the parameters should satisfy. The idea is to choose the parameter estimates so that the theoretical relation is satisfied as “closely” as possible. The theoretical relation is replaced by its sample counterpart and the estimates are chosen to minimize the weighted distance between the theoretical and actual values. GMM is a robust estimator in that, unlike maximum likelihood estimation, it does not require information of the exact distribution of the disturbances. In fact, many common estimators in econometrics can be considered as special cases of GMM.

The theoretical relation that the parameters should satisfy are usually *orthogonality conditions* between some (possibly nonlinear) function of the parameters $f(\theta)$ and a set of instrumental variables z_t :

$$E(f(\theta)'Z) = 0, \quad (25.46)$$

where θ are the parameters to be estimated. The GMM estimator selects parameter estimates so that the sample correlations between the instruments and the function f are as close to zero as possible, as defined by the criterion function:

$$J(\theta) = (m(\theta))' A m(\theta), \quad (25.47)$$

where $m(\theta) = f(\theta)'Z$ and A is a weighting matrix. Any symmetric positive definite matrix A will yield a consistent estimate of q . However, it can be shown that a necessary (but not sufficient) condition to obtain an (asymptotically) efficient estimate of q is to set A equal to the inverse of the covariance matrix of the sample moments m . It is worth noting here that in standard single equation GMM, EViews estimates a model with the objective function (25.47) divided by the number of observations. For equations with panel data (“GMM Estimation,” beginning on page 545) the objective function is *not* similarly scaled.

Many standard estimators, including all of the system estimators provided in EViews, can be set up as special cases of GMM. For example, the ordinary least squares estimator can be viewed as a GMM estimator, based upon the conditions that each of the right-hand variables is uncorrelated with the residual.

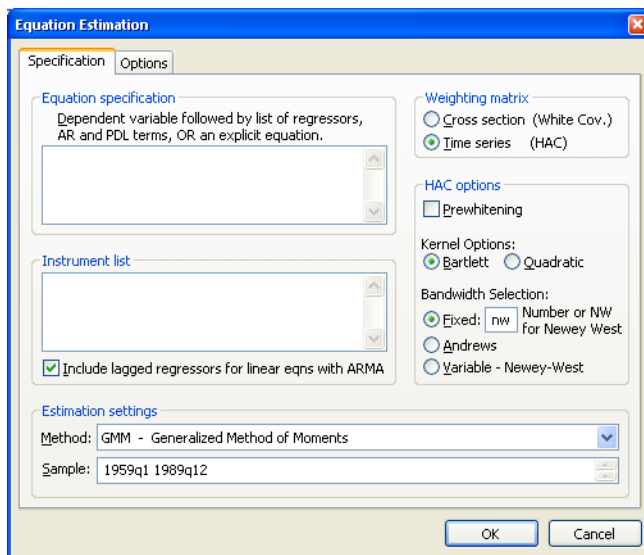
Estimation by GMM in EViews

To estimate an equation by GMM, either create a new equation object by selecting **Object/New Object.../Equation**, or press the **Estimate** button in the toolbar of an existing equation. From the **Equation Specification** dialog choose **Estimation Method: GMM**. The estimation specification dialog will change as depicted below.

To obtain GMM estimates in EViews, you need to write the moment condition as an orthogonality condition between an expression including the parameters and a set of instrumental variables. There are two ways you can write the orthogonality condition: with and without a dependent variable.

If you specify the equation either by listing variable names or by an expression with an equal sign,

EViews will interpret the moment condition as an orthogonality condition between the



instruments and the residuals defined by the equation. If you specify the equation by an expression without an equal sign, EViews will orthogonalize that expression to the set of instruments.

You must also list the names of the instruments in the **Instrument list** field box of the Equation Specification dialog box. For the GMM estimator to be identified, there must be at least as many instrumental variables as there are parameters to estimate. EViews will always include the constant in the list of instruments.

For example, if you type,

Equation spec: $y \ c \ x$

Instrument list: $c \ z \ w$

the orthogonality conditions given by:

$$\begin{aligned}\sum (y_t - c(1) - c(2)x_t) &= 0 \\ \sum (y_t - c(1) - c(2)x_t)z_t &= 0 \\ \sum (y_t - c(1) - c(2)x_t)w_t &= 0\end{aligned}\tag{25.48}$$

If you enter the specification,

Equation spec: $c(1) * \log(y) + x^{c(2)}$

Instrument list: $c \ z \ z(-1)$

the orthogonality conditions are:

$$\begin{aligned}\sum (c(1)\log y_t + x_t^{c(2)}) &= 0 \\ \sum (c(1)\log y_t + x_t^{c(2)})z_t &= 0 \\ \sum (c(1)\log y_t + x_t^{c(2)})z_{t-1} &= 0\end{aligned}\tag{25.49}$$

On the right part of the Equation Specification dialog are the options for selecting the weighting matrix A in the objective function. If you select **Weighting Matrix: Cross section (White Cov)**, the GMM estimates will be robust to heteroskedasticity of unknown form.

If you select **Weighting Matrix: Time series (HAC)**, the GMM estimates will be robust to heteroskedasticity and autocorrelation of unknown form. For the HAC option, you have to specify the kernel type and bandwidth.

- The **Kernel Options** determine the functional form of the kernel used to weight the autocovariances in computing the weighting matrix.
- The **Bandwidth Selection** option determines how the weights given by the kernel change with the lags of the autocovariances in the computation of the weighting matrix. If you select **Fixed** bandwidth, you may either enter a number for the bandwidth or type `nw` to use Newey and West's fixed bandwidth selection criterion.

- The **Prewhitening** option runs a preliminary VAR(1) prior to estimation to “soak up” the correlation in the moment conditions.

The technical notes in [“Generalized Method of Moments \(GMM\)” on page 337](#) describe these options in greater detail.

Example

Tauchen (1986) considers the problem of estimating the taste parameters β , γ from the Euler equation:

$$(\beta R_{t+1} w_{t+1}^{-\gamma} - 1)' z_t = 0 \quad (25.50)$$

where we use instruments $z_t = (1, w_t, w_{t-1}, r_t, r_{t-1})'$. To estimate the parameters β , γ by GMM, fill in the Equation Specification dialog as:

Equation Specification: `c (1) * r (+1) * w (+1) ^ (-c (2)) -1`

Instrument list: `c w w (-1) r r (-1)`

The estimation result using the default HAC **Weighting Matrix** option looks as follows:

| Dependent Variable: Implicit Equation | | | | |
|--|-------------|--------------------|-------------|----------|
| Method: Generalized Method of Moments | | | | |
| Date: 08/07/06 Time: 15:15 | | | | |
| Sample (adjusted): 1891 1982 | | | | |
| Included observations: 92 after adjustments | | | | |
| Kernel: Bartlett, Bandwidth: Fixed (3), No prewhitening | | | | |
| Simultaneous weighting matrix & coefficient iteration | | | | |
| Convergence achieved after: 8 weight matrices, 9 total coef iterations | | | | |
| C(1)*R(+1)*W(+1)^(-C(2))-1 | | | | |
| Instrument list: C W W(-1) R R(-1) | | | | |
| Variable | Coefficient | Std. Error | t-Statistic | Prob. |
| C(1) | 0.934098 | 0.018751 | 49.81496 | 0.0000 |
| C(2) | 1.366557 | 0.741771 | 1.842288 | 0.0687 |
| Mean dependent var | 0.000000 | S.D. dependent var | | 0.000000 |
| S.E. of regression | 0.154082 | Sum squared resid | | 2.136723 |
| Durbin-Watson stat | 1.903829 | J-statistic | | 0.054524 |

Note that when you specify an equation without a dependent variable, EViews does not report some of the regression statistics such as the R-squared. The J -statistic reported at the bottom of the table is the minimized value of the objective function, where we report [\(25.47\)](#) divided by the number of observations (see [“Generalized Method of Moments \(GMM\),” beginning on page 51](#), for additional discussion). This J -statistic may be used to carry out hypothesis tests from GMM estimation; see Newey and West (1987a). A simple

application of the J -statistic is to test the validity of overidentifying restrictions. Under the null hypothesis that the overidentifying restrictions are satisfied, the J -statistic times the number of regression observations is asymptotically χ^2 with degrees of freedom equal to the number of overidentifying restrictions. You can compute the test statistic as a named scalar in EViews using the commands:

```
scalar overid=eq_gmm.@regobs*eq_gmm.@jstat  
scalar overid_p=1-@cchisq(overid,3)
```

where EQ_GMM is the name of the equation containing the GMM estimates. The second command computes the p -value of the test statistic as a named scalar OVERID_P. To view the value of OVERID_P, double click on its name; the value will be displayed in the status line at the bottom of the EViews window.

Stepwise Least Squares Regression

EViews allows you to perform automatic variable selection using stepwise regression. Stepwise regression allows some or all of the variables in a standard linear multivariate regression to be chosen automatically, using various statistical criteria, from a set of variables.

There is a fairly large literature describing the benefits and the pitfalls of stepwise regression. Without making any recommendations ourselves, we refer the user to Derksen and Keselman (1992), Roecker (1991), Hurvich and Tsai (1990).

Stepwise Least Squares Estimation in EViews

To perform a Stepwise selection procedure (STEPLS) in EViews select **Object/New Object/Equation**, or press **Estimate** from the toolbar of an existing equation. From the **Equation Specification** dialog choose **Method: STEPLS - Stepwise Least Squares**. EViews will display the following dialog:

The **Specification** page allows you to provide the basic STEPLS regression specification. In the upper edit field you should first specify the dependent variable followed by the always included variables you wish to use in the final regression. Note that the STEPLS equation must be specified by list.

You should enter a list of variables to be used as the set of potentially included variables in the second edit field.

The screenshot shows the 'Equation Estimation' dialog box with the 'Specification' tab selected. The 'Equation specification' section has two empty text boxes for 'Dependent variable followed by list of always included regressors' and 'List of search regressors'. The 'Estimation settings' section shows 'Method' set to 'STEPLS - Stepwise Least Squares' and 'Sample' set to '1 100'. 'OK' and 'Cancel' buttons are at the bottom right.

Next, you may use the **Options** tab to control the stepwise estimation method.

The **Selection Method** portion of the **Options** page is used to specify the STEPLS method.

By default, EViews will estimate the stepwise specification using the Stepwise-Forwards method. To change the basic method, change the **Selection Method** combo box; the combo allows you to choose between: **Uni-directional**, **Stepwise**, **Swapwise**, and **Combinatorial**.

The other items on this dialog tab will change depending upon which method you

The screenshot shows the 'Equation Estimation' dialog box with the 'Options' tab selected. The 'Selection Method' dropdown is set to 'Stepwise'. Under 'Stopping Criteria', 'p-value' is selected, with 'p-value forwards' and 'p-value backwards' both set to 0.5. The 'Use number of regressors' checkbox is checked, with 'Number of regressors to select' set to 3. Under 'Weights', 'Use weight series' is unchecked. Under 'Maximum steps', 'Forwards' is 1000, 'Backwards' is 1000, and 'Total' is 2000. 'OK' and 'Cancel' buttons are at the bottom right.

choose. For the **Uni-directional** and **Stepwise** methods you may specify the direction of the method using the **Forwards** and **Backwards** radio buttons. These two methods allow you to provide a **Stopping Criteria** using either a p -value or t -statistic tolerance for adding or removing variables. You may also choose to stop the procedures once they have added or

removed a specified number of regressors by selecting the **Use number of regressors** option and providing a number of the corresponding edit field.

You may also set the maximum number of steps taken by the procedure. To set the maximum number of additions to the model, change the **Forwards** steps, and to set the maximum number of removals, change the **Backwards** steps. You may also set the total number of additions and removals. In general it is best to leave these numbers at a high value. Note, however, that the **Stepwise** routines have the potential to repetitively add and remove the same variables, and by setting the maximum number of steps you can mitigate this behavior.

The **Swapwise** method lets you choose whether you wish to use **Max R-squared** or **Min R-squared**, and choose the number of additional variables to be selected. The **Combinatorial** method simply prompts you to provide the number of additional variables. By default both of these procedures have the number of additional variables set to one. In both cases this merely chooses the single variable that will lead to the largest increase in R-squared.

For additional discussion, see [“Selection Methods,” beginning on page 59](#).

Lastly, each of the methods lets you choose a **Weight series** to perform weighted least squares estimation. Simply check the **Use weight series** option, then enter the name of the weight series in the edit field. See [“Weighted Least Squares” on page 32](#) for details.

Example

As an example we use the following code to generate a workfile with 40 independent variables (X1–X40), and a dependent variable, Y, which is a linear combination of a constant, variables X11–X15, and a normally distributed random error term.

```
create u 100
rndseed 1
group xs
for !i=1 to 40
    series x!i=nrnd
    %name="x"+@str(!i)
    xs.add {%name}
next
series y = nrnd + 3
for !i=11 to 15
    y = y + !i*x{!i}
next
```

The 40 independent variables are contained in the group XS.

Given this data we can use a forwards stepwise routine to choose the “best” 5 regressors, after the constant, from the group of 40 in XS. We do this by entering “Y C” in the first **Specification** box of the estimation dialog, and “XS” in the second **Specification** box. In the **Stopping Criteria** section of the **Options** tab we check **Use Number of Regressors**, and enter “5” as the number of regressors. Estimating this specification yields the results:

| Dependent Variable: Y | | | | |
|--|-------------|-----------------------|-------------|--------|
| Method: Stepwise Regression | | | | |
| Date: 10/09/06 Time: 16:12 | | | | |
| Sample: 1 100 | | | | |
| Included observations: 100 | | | | |
| Number of always included regressors: 1 | | | | |
| Number of search regressors: 40 | | | | |
| Selection method: Stepwise forwards | | | | |
| Stopping criterion: p-value forwards/backwards = 0.5/0.5 | | | | |
| Stopping criterion: Number of search regressors = 5 | | | | |
| | Coefficient | Std. Error | t-Statistic | Prob.* |
| C | 2.973731 | 0.102755 | 28.93992 | 0.0000 |
| X15 | 14.98849 | 0.091087 | 164.5517 | 0.0000 |
| X14 | 14.01298 | 0.091173 | 153.6967 | 0.0000 |
| X12 | 11.85221 | 0.101569 | 116.6914 | 0.0000 |
| X13 | 12.88029 | 0.102182 | 126.0526 | 0.0000 |
| X11 | 11.02252 | 0.102758 | 107.2664 | 0.0000 |
| R-squared | 0.999211 | Mean dependent var | -0.992126 | |
| Adjusted R-squared | 0.999169 | S.D. dependent var | 33.58749 | |
| S.E. of regression | 0.968339 | Akaike info criterion | 2.831656 | |
| Sum squared resid | 88.14197 | Schwarz criterion | 2.987966 | |
| Log likelihood | -135.5828 | Hannan-Quinn criter. | 2.894917 | |
| F-statistic | 23802.50 | Durbin-Watson stat | 1.921653 | |
| Prob(F-statistic) | 0.000000 | | | |
| Selection Summary | | | | |
| Added X15 | | | | |
| Added X14 | | | | |
| Added X12 | | | | |
| Added X13 | | | | |
| Added X11 | | | | |

*Note: p-values and subsequent tests do not account for stepwise selection.

The top portion of the output shows the equation specification and information about the stepwise method. The next section shows the final estimated specification along with coefficient estimates, standard errors and *t*-statistics, and *p*-values. Note that the stepwise routine chose the “correct” five regressors, X11–X15. The bottom portion of the output shows a sum-

mary of the steps taken by the selection method. Specifications with a large number of steps may show only a brief summary.

Selection Methods

EViews allows you to specify variables to be included as regressors along with a set of variables from which the selection procedure will choose additional regressors. The first set of variables are termed the “always included” variables, and the latter are the set of potential “added variables”. EViews supports several procedures for selecting the added variables.

Uni-directional-Forwards

The Uni-directional-Forwards method uses either a lowest p -value or largest t -statistic criterion for adding variables.

The method begins with no added regressors. If using the p -value criterion, we select the variable that would have the lowest p -value were it added to the regression. If the p -value is lower than the specified stopping criteria, the variable is added. The selection continues by selecting the variable with the next lowest p -value, given the inclusion of the first variable. The procedure stops when the lowest p -value of the variables not yet included is greater than the specified forwards stopping criterion, or the number of forward steps or number of added regressors reach the optional user specified limits.

If using the largest t -statistic criterion, the same variables are selected, but the stopping criterion is specified in terms of the statistic value instead of the p -value.

Uni-directional-Backwards

The Uni-directional-Backwards method is analogous to the Uni-directional-Forwards method, but begins with all possible added variables included, and then removes the variable with the highest p -value. The procedure continues by removing the variable with the next highest p -value, given that the first variable has already been removed. This process continues until the highest p -value is less than the specified backwards stopping criteria, or the number of backward steps or number of added regressors reach the optional user specified limits.

The largest t -statistic may be used in place of the lowest p -value as a selection criterion.

Stepwise-Forwards

The Stepwise-Forwards method is a combination of the Uni-directional-Forwards and Backwards methods. Stepwise-Forwards begins with no additional regressors in the regression, then adds the variable with the lowest p -value. The variable with the next lowest p -value given that the first variable has already been chosen, is then added. Next both of the added variables are checked against the backwards p -value criterion. Any variable whose p -value is higher than the criterion is removed.

Once the removal step has been performed, the next variable is added. At this, and each successive addition to the model, all the previously added variables are checked against the backwards criterion and possibly removed. The Stepwise-Forwards routine ends when the lowest p -value of the variables not yet included is greater than the specified forwards stopping criteria (or the number of forwards and backwards steps or the number of added regressors has reached the corresponding optional user specified limit).

You may elect to use the largest t -statistic in place of the lowest p -value as the selection criterion.

Stepwise-Backwards

The Stepwise-Backwards procedure reverses the Stepwise-Forwards method. All possible added variables are first included in the model. The variable with the highest p -value is first removed. The variable with the next highest p -value, given the removal of the first variable, is also removed. Next both of the removed variables are checked against the forwards p -value criterion. Any variable whose p -value is lower than the criterion is added back in to the model.

Once the addition step has been performed, the next variable is removed. This process continues where at each successive removal from the model, all the previously removed variables are checked against the forwards criterion and potentially re-added. The Stepwise-Backwards routine ends when the largest p -value of the variables inside the model is less than the specified backwards stopping criterion, or the number of forwards and backwards steps or number of regressors reaches the corresponding optional user specified limit.

The largest t -statistic may be used in place of the lowest p -value as a selection criterion.

Swapwise-Max R-Squared Increment

The Swapwise method starts with no additional regressors in the model. The procedure starts by adding the variable which maximizes the resulting regression R-squared. The variable that leads to the largest increase in R-squared is then added. Next each of the two variables that have been added as regressors are compared individually with all variables not included in the model, calculating whether the R-squared could be improved by swapping the “inside” with an “outside” variable. If such an improvement exists then the “inside” variable is replaced by the “outside” variable. If there exists more than one swap that would improve the R-squared, the swap that yields the largest increase is made.

Once a swap has been made the comparison process starts again. Once all comparisons and possible swaps are made, a third variable is added, with the variable chosen to produce the largest increase in R-squared. The three variables inside the model are then compared with all the variables outside the model and any R-squared increasing swaps are made. This process continues until the number of variables added to the model reaches the user-specified limit.

Swapwise-Min R-Squared Increment

The Min R-squared Swapwise method is very similar to the Max R-squared method. The difference lies in the swapping procedure. Whereas the Max R-squared swaps the variables that would lead to the largest increase in R-squared, the Min R-squared method makes a swap based on the smallest increase. This can lead to a more lengthy selection process, with a larger number of combinations of variables compared.

Combinatorial

For a given number of added variables, the Combinatorial method evaluates every possible combination of added variables, and selects the combination that leads to the largest R-squared in a regression using the added and always included variables as regressors. This method is more thorough than the previous methods, since those methods do not compare every possible combination of variables, and obviously requires additional computation. With large numbers of potential added variables, the Combinatorial approach can take a very long time to complete.

Issues with Stepwise Estimation

The set of search variables may contain variables that are linear combinations of other variables in the regression (either in the always included list, or in the search set). EViews will drop those variables from the search set. In a case where two or more of the search variables are collinear, EViews will select the variable listed first in the list of search variables.

Following the Stepwise selection process, EViews reports the results of the final regression, *i.e.* the regression of the always-included and the selected variables on the dependent variable. In some cases the sample used in this equation may not coincide with the regression that was used during the selection process. This will occur if some of the omitted search variables have missing values for some observations that do not have missing values in the final regression. In such cases EViews will print a warning in the regression output.

The p -values listed in the final regression output and all subsequent testing procedures do not account for the regressions that were run during the selection process. One should take care to interpret results accordingly.

Invalid inference is but one of the reasons that stepwise regression and other variable selection methods have a large number of critics amongst statisticians. Other problems include an upwardly biased final R-squared, possibly upwardly biased coefficient estimates, and narrow confidence intervals. It is also often pointed out that the selection methods themselves use statistics that do not account for the selection process.

References

- Davidson, Russell and James G. MacKinnon (1993). *Estimation and Inference in Econometrics*, Oxford: Oxford University Press.
- Derksen, S. and H. J. Keselman (1992). “Backward, Forward and Stepwise Automated Subset Selection Algorithms: Frequency of Obtaining Authentic and Noise Variables,” *British Journal of Mathematical and Statistical Psychology*, 45, 265–282.
- Fair, Ray C. (1970). “The Estimation of Simultaneous Equation Models With Lagged Endogenous Variables and First Order Serially Correlated Errors,” *Econometrica*, 38, 507–516.
- Fair, Ray C. (1984). *Specification, Estimation, and Analysis of Macroeconometric Models*, Cambridge, MA: Harvard University Press.
- Harrison, D. and D. L. Rubinfeld (1978). “Hedonic Housing Prices and the Demand for Clean Air,” *Journal of Environmental Economics and Management*, 5, 81-102.
- Hurvich, C. M. and C. L. Tsai (1990). “The Impact of Model Selection on Inference in Linear Regression,” *American Statistician*, 44, 214–217.
- Johnston, Jack and John Enrico DiNardo (1997). *Econometric Methods*, 4th Edition, New York: McGraw-Hill.
- Newey, Whitney and Kenneth West (1987a). “Hypothesis Testing with Efficient Method of Moments Estimation,” *International Economic Review*, 28, 777–787.
- Newey, Whitney and Kenneth West (1987b). “A Simple Positive Semi-Definite, Heteroskedasticity and Autocorrelation Consistent Covariance Matrix,” *Econometrica*, 55, 703–708.
- Pindyck, Robert S. and Daniel L. Rubinfeld (1998). *Econometric Models and Economic Forecasts*, 4th edition, New York: McGraw-Hill.
- Roecker, E. B. (1991). “Prediction Error and its Estimation for Subset-Selection Models,” *Technometrics*, 33, 459–469.
- Tauchen, George (1986). “Statistical Properties of Generalized Method-of-Moments Estimators of Structural Parameters Obtained From Financial Market Data,” *Journal of Business & Economic Statistics*, 4, 397–416.
- White, Halbert (1980). “A Heteroskedasticity-Consistent Covariance Matrix and a Direct Test for Heteroskedasticity,” *Econometrica*, 48, 817–838.
- Wooldridge, Jeffrey M. (2000). *Introductory Econometrics: A Modern Approach*, South-Western College Publishing.

Chapter 26. Time Series Regression

In this section, we discuss single equation regression techniques that are important for the analysis of time series data: testing for serial correlation, estimation of ARMA models, using polynomial distributed lags, and testing for unit roots in potentially nonstationary time series.

The chapter focuses on the specification and estimation of time series models. A number of related topics are discussed elsewhere. For example, standard multiple regression techniques are discussed in [Chapter 24. “Basic Regression,” on page 5](#) and [Chapter 25. “Additional Regression Methods,” on page 23](#), while forecasting and inference are discussed extensively in [Chapter 27. “Forecasting from an Equation,” on page 113](#).

Serial Correlation Theory

A common finding in time series regressions is that the residuals are correlated with their own lagged values. This serial correlation violates the standard assumption of regression theory that disturbances are not correlated with other disturbances. The primary problems associated with serial correlation are:

- OLS is no longer efficient among linear estimators. Furthermore, since prior residuals help to predict current residuals, we can take advantage of this information to form a better prediction of the dependent variable.
- Standard errors computed using the textbook OLS formula are not correct, and are generally understated.
- If there are lagged dependent variables on the right-hand side, OLS estimates are biased and inconsistent.

EViews provides tools for detecting serial correlation and estimation methods that take account of its presence.

In general, we will be concerned with specifications of the form:

$$\begin{aligned}y_t &= x_t' \beta + u_t \\ u_t &= z_{t-1}' \gamma + \epsilon_t\end{aligned}\tag{26.1}$$

where x_t is a vector of explanatory variables observed at time t , z_{t-1} is a vector of variables known in the previous period, β and γ are vectors of parameters, u_t is a disturbance term, and ϵ_t is the innovation in the disturbance. The vector z_{t-1} may contain lagged values of u , lagged values of ϵ , or both.

The disturbance u_t is termed the *unconditional residual*. It is the residual based on the structural component ($x_t \beta$) but not using the information contained in z_{t-1} . The innova-

tion ϵ_t is also known as the *one-period ahead forecast error* or the *prediction error*. It is the difference between the actual value of the dependent variable and a forecast made on the basis of the independent variables and the past forecast errors.

The First-Order Autoregressive Model

The simplest and most widely used model of serial correlation is the first-order autoregressive, or AR(1), model. The AR(1) model is specified as:

$$\begin{aligned} y_t &= x_t' \beta + u_t \\ u_t &= \rho u_{t-1} + \epsilon_t \end{aligned} \tag{26.2}$$

The parameter ρ is the first-order serial correlation coefficient. In effect, the AR(1) model incorporates the residual from the past observation into the regression model for the current observation.

Higher-Order Autoregressive Models

More generally, a regression with an autoregressive process of order p , AR(p) error is given by:

$$\begin{aligned} y_t &= x_t' \beta + u_t \\ u_t &= \rho_1 u_{t-1} + \rho_2 u_{t-2} + \dots + \rho_p u_{t-p} + \epsilon_t \end{aligned} \tag{26.3}$$

The autocorrelations of a stationary AR(p) process gradually die out to zero, while the partial autocorrelations for lags larger than p are zero.

Testing for Serial Correlation

Before you use an estimated equation for statistical inference (*e.g.* hypothesis tests and forecasting), you should generally examine the residuals for evidence of serial correlation.

EViews provides several methods of testing a specification for the presence of serial correlation.

The Durbin-Watson Statistic

EViews reports the Durbin-Watson (DW) statistic as a part of the standard regression output. The Durbin-Watson statistic is a test for first-order serial correlation. More formally, the DW statistic measures the linear association between adjacent residuals from a regression model. The Durbin-Watson is a test of the hypothesis $\rho = 0$ in the specification:

$$u_t = \rho u_{t-1} + \epsilon_t. \tag{26.4}$$

If there is no serial correlation, the DW statistic will be around 2. The DW statistic will fall below 2 if there is positive serial correlation (in the worst case, it will be near zero). If there is negative correlation, the statistic will lie somewhere between 2 and 4.

Positive serial correlation is the most commonly observed form of dependence. As a rule of thumb, with 50 or more observations and only a few independent variables, a DW statistic below about 1.5 is a strong indication of positive first order serial correlation. See Johnston and DiNardo (1997, Chapter 6.6.1) for a thorough discussion on the Durbin-Watson test and a table of the significance points of the statistic.

There are three main limitations of the DW test as a test for serial correlation. First, the distribution of the DW statistic under the null hypothesis depends on the data matrix x . The usual approach to handling this problem is to place bounds on the critical region, creating a region where the test results are inconclusive. Second, if there are lagged dependent variables on the right-hand side of the regression, the DW test is no longer valid. Lastly, you may only test the null hypothesis of no serial correlation against the alternative hypothesis of first-order serial correlation.

Two other tests of serial correlation—the Q -statistic and the Breusch-Godfrey LM test—overcome these limitations, and are preferred in most applications.

Correlograms and Q -statistics

If you select **View/Residual Tests/Correlogram- Q -statistics** on the equation toolbar, EViews will display the autocorrelation and partial autocorrelation functions of the residuals, together with the Ljung-Box Q -statistics for high-order serial correlation. If there is no serial correlation in the residuals, the autocorrelations and partial autocorrelations at all lags should be nearly zero, and all Q -statistics should be insignificant with large p -values.

Note that the p -values of the Q -statistics will be computed with the degrees of freedom adjusted for the inclusion of ARMA terms in your regression. There is evidence that some care should be taken in interpreting the results of a Ljung-Box test applied to the residuals from an ARMAX specification (see Dezhbaksh, 1990, for simulation evidence on the finite sample performance of the test in this setting).

Details on the computation of correlograms and Q -statistics are provided in greater detail in [Chapter 11. “Series,” on page 326](#) of the *User’s Guide I*.

Serial Correlation LM Test

Selecting **View/Residual Tests/Serial Correlation LM Test...** carries out the Breusch-Godfrey Lagrange multiplier test for general, high-order, ARMA errors. In the Lag Specification dialog box, you should enter the highest order of serial correlation to be tested.

The null hypothesis of the test is that there is no serial correlation in the residuals up to the specified order. EViews reports a statistic labeled “ F -statistic” and an “Obs*R-squared” (NR^2 —the number of observations times the R-square) statistic. The NR^2 statistic has an asymptotic χ^2 distribution under the null hypothesis. The distribution of the F -statistic is not known, but is often used to conduct an informal test of the null.

See “[Serial Correlation LM Test](#)” on page 155 for further discussion of the serial correlation LM test.

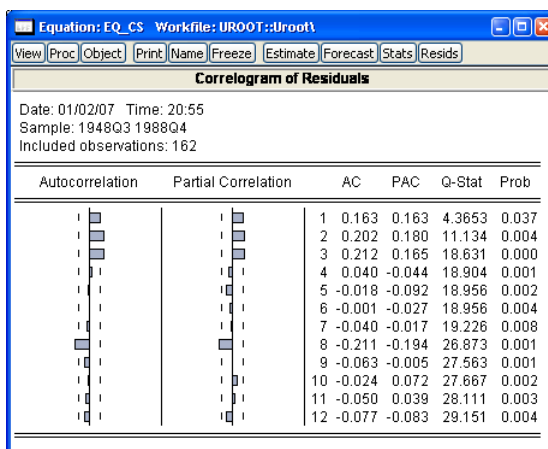
Example

As an example of the application of these testing procedures, consider the following results from estimating a simple consumption function by ordinary least squares:

| Dependent Variable: CS | | | | |
|----------------------------|-------------|-----------------------|-------------|----------|
| Method: Least Squares | | | | |
| Date: 08/07/06 Time: 15:44 | | | | |
| Sample: 1948Q3 1988Q4 | | | | |
| Included observations: 162 | | | | |
| | Coefficient | Std. Error | t-Statistic | Prob. |
| C | -9.227624 | 5.898177 | -1.564487 | 0.1197 |
| GDP | 0.038732 | 0.017205 | 2.251193 | 0.0257 |
| CS(-1) | 0.952049 | 0.024484 | 38.88516 | 0.0000 |
| R-squared | 0.999625 | Mean dependent var | | 1781.675 |
| Adjusted R-squared | 0.999621 | S.D. dependent var | | 694.5419 |
| S.E. of regression | 13.53003 | Akaike info criterion | | 8.066046 |
| Sum squared resid | 29106.82 | Schwarz criterion | | 8.123223 |
| Log likelihood | -650.3497 | Hannan-Quinn criter. | | 8.089261 |
| F-statistic | 212047.1 | Durbin-Watson stat | | 1.672255 |
| Prob(F-statistic) | 0.000000 | | | |

A quick glance at the results reveals that the coefficients are statistically significant and the fit is very tight. However, if the error term is serially correlated, the estimated OLS standard errors are invalid and the estimated coefficients will be biased and inconsistent due to the presence of a lagged dependent variable on the right-hand side. The Durbin-Watson statistic is not appropriate as a test for serial correlation in this case, since there is a lagged dependent variable on the right-hand side of the equation.

Selecting **View/Residual Tests/Correlogram-Q-statistics** for the first 12 lags from this equation produces the following view:



The correlogram has spikes at lags up to three and at lag eight. The Q -statistics are significant at all lags, indicating significant serial correlation in the residuals.

Selecting **View/Residual Tests/Serial Correlation LM Test...** and entering a lag of 4 yields the following result:

Breusch-Godfrey Serial Correlation LM Test:

| | | | |
|---------------|----------|---------------------|--------|
| F-statistic | 3.654696 | Prob. F(4,155) | 0.0071 |
| Obs*R-squared | 13.96215 | Prob. Chi-Square(4) | 0.0074 |

The test rejects the hypothesis of no serial correlation up to order four. The Q -statistic and the LM test both indicate that the residuals are serially correlated and the equation should be re-specified before using it for hypothesis tests and forecasting.

Estimating AR Models

Before you use the tools described in this section, you may first wish to examine your model for other signs of misspecification. Serial correlation in the errors may be evidence of serious problems with your specification. In particular, you should be on guard for an excessively restrictive specification that you arrived at by experimenting with ordinary least squares. Sometimes, adding improperly excluded variables to your regression will eliminate the serial correlation.

For a discussion of the efficiency gains from the serial correlation correction and some Monte-Carlo evidence, see Rao and Griliches (1969).

First-Order Serial Correlation

To estimate an AR(1) model in EViews, open an equation by selecting **Quick/Estimate Equation...** and enter your specification as usual, adding the special expression “AR(1)” to the end of your list. For example, to estimate a simple consumption function with AR(1) errors,

$$\begin{aligned}CS_t &= c_1 + c_2 GDP_t + u_t \\ u_t &= \rho u_{t-1} + \epsilon_t\end{aligned}\tag{26.5}$$

you should specify your equation as:

```
cs c gdp ar(1)
```

EViews automatically adjusts your sample to account for the lagged data used in estimation, estimates the model, and reports the adjusted sample along with the remainder of the estimation output.

Higher-Order Serial Correlation

Estimating higher order AR models is only slightly more complicated. To estimate an AR(k), you should enter your specification, followed by expressions for each AR term you wish to include. If you wish to estimate a model with autocorrelations from one to five:

$$\begin{aligned}CS_t &= c_1 + c_2 GDP_t + u_t \\ u_t &= \rho_1 u_{t-1} + \rho_2 u_{t-2} + \dots + \rho_5 u_{t-5} + \epsilon_t\end{aligned}\tag{26.6}$$

you should enter:

```
cs c gdp ar(1) ar(2) ar(3) ar(4) ar(5)
```

By requiring that you enter all of the autocorrelations you wish to include in your model, EViews allows you great flexibility in restricting lower order correlations to be zero. For example, if you have quarterly data and want to include a single term to account for seasonal autocorrelation, you could enter

```
cs c gdp ar(4)
```

Nonlinear Models with Serial Correlation

EViews can estimate nonlinear regression models with additive AR errors. For example, suppose you wish to estimate the following nonlinear specification with an AR(2) error:

$$\begin{aligned}CS_t &= c_1 + GDP_t^{c_2} + u_t \\ u_t &= c_3 u_{t-1} + c_4 u_{t-2} + \epsilon_t\end{aligned}\tag{26.7}$$

Simply specify your model using EViews expressions, followed by an additive term describing the AR correction enclosed in square brackets. The AR term should contain a coefficient assignment for each AR lag, separated by commas:

$$cs = c(1) + gdp^{\wedge}c(2) + [ar(1)=c(3), ar(2)=c(4)]$$

EViews transforms this nonlinear model by differencing, and estimates the transformed nonlinear specification using a Gauss-Newton iterative procedure (see [“How EViews Estimates AR Models” on page 70](#)).

Two-Stage Regression Models with Serial Correlation

By combining two-stage least squares or two-stage nonlinear least squares with AR terms, you can estimate models where there is correlation between regressors and the innovations as well as serial correlation in the residuals.

If the original regression model is linear, EViews uses the Marquardt algorithm to estimate the parameters of the transformed specification. If the original model is nonlinear, EViews uses Gauss-Newton to estimate the AR corrected specification.

For further details on the algorithms and related issues associated with the choice of instruments, see the discussion in [“TSLS with AR errors,” beginning on page 40](#).

Output from AR Estimation

When estimating an AR model, some care must be taken in interpreting your results. While the estimated coefficients, coefficient standard errors, and t -statistics may be interpreted in the usual manner, results involving residuals differ from those computed in OLS settings.

To understand these differences, keep in mind that there are two different residuals associated with an AR model. The first are the estimated *unconditional residuals*:

$$\hat{u}_t = y_t - x_t' b, \quad (26.8)$$

which are computed using the original variables, and the estimated coefficients, b . These residuals are the errors that you would observe if you made a prediction of the value of y_t using contemporaneous information, but ignoring the information contained in the lagged residual.

Normally, there is no strong reason to examine these residuals, and EViews does not automatically compute them following estimation.

The second set of residuals are the estimated *one-period ahead forecast errors*, $\hat{\epsilon}$. As the name suggests, these residuals represent the forecast errors you would make if you computed forecasts using a prediction of the residuals based upon past values of your data, in addition to the contemporaneous information. In essence, you improve upon the unconditional forecasts and residuals by taking advantage of the predictive power of the lagged residuals.

For AR models, the residual-based regression statistics—such as the R^2 , the standard error of regression, and the Durbin-Watson statistic—reported by EViews are based on the one-period ahead forecast errors, $\hat{\epsilon}$.

A set of statistics that is unique to AR models is the estimated AR parameters, $\hat{\rho}_i$. For the simple AR(1) model, the estimated parameter $\hat{\rho}$ is the serial correlation coefficient of the unconditional residuals. For a stationary AR(1) model, the true ρ lies between -1 (extreme negative serial correlation) and $+1$ (extreme positive serial correlation). The stationarity condition for general AR(p) processes is that the inverted roots of the lag polynomial lie inside the unit circle. EViews reports these roots as **Inverted AR Roots** at the bottom of the regression output. There is no particular problem if the roots are imaginary, but a stationary AR model should have all roots with modulus less than one.

How EViews Estimates AR Models

Textbooks often describe techniques for estimating AR models. The most widely discussed approaches, the Cochrane-Orcutt, Prais-Winsten, Hatanaka, and Hildreth-Lu procedures, are multi-step approaches designed so that estimation can be performed using standard *linear* regression. All of these approaches suffer from important drawbacks which occur when working with models containing lagged dependent variables as regressors, or models using higher-order AR specifications; see Davidson and MacKinnon (1993, p. 329–341), Greene (1997, p. 600–607).

EViews estimates AR models using nonlinear regression techniques. This approach has the advantage of being easy to understand, generally applicable, and easily extended to nonlinear specifications and models that contain endogenous right-hand side variables. Note that the nonlinear least squares estimates are asymptotically equivalent to maximum likelihood estimates and are asymptotically efficient.

To estimate an AR(1) model, EViews transforms the linear model,

$$\begin{aligned} y_t &= x_t' \beta + u_t \\ u_t &= \rho u_{t-1} + \epsilon_t \end{aligned} \tag{26.9}$$

into the nonlinear model:

$$y_t = \rho y_{t-1} + (x_t - \rho x_{t-1})' \beta + \epsilon_t, \tag{26.10}$$

by substituting the second equation into the first, and rearranging terms. The coefficients ρ and β are estimated simultaneously by applying a Marquardt nonlinear least squares algorithm to the transformed equation. See [Appendix D. “Estimation and Solution Options,” on page 625](#) for details on nonlinear estimation.

For a nonlinear AR(1) specification, EViews transforms the nonlinear model,

$$\begin{aligned}y_t &= f(x_t, \beta) + u_t \\u_t &= \rho u_{t-1} + \epsilon_t\end{aligned}\tag{26.11}$$

into the alternative nonlinear specification:

$$y_t = \rho y_{t-1} + f(x_t, \beta) - \rho f(x_{t-1}, \beta) + \epsilon_t\tag{26.12}$$

and estimates the coefficients using a Marquardt nonlinear least squares algorithm.

Higher order AR specifications are handled analogously. For example, a nonlinear AR(3) is estimated using nonlinear least squares on the equation:

$$\begin{aligned}y_t &= (\rho_1 y_{t-1} + \rho_2 y_{t-2} + \rho_3 y_{t-3}) + f(x_t, \beta) - \rho_1 f(x_{t-1}, \beta) \\&\quad - \rho_2 f(x_{t-2}, \beta) - \rho_3 f(x_{t-3}, \beta) + \epsilon_t\end{aligned}\tag{26.13}$$

For details, see Fair (1984, p. 210–214), and Davidson and MacKinnon (1993, p. 331–341).

ARIMA Theory

ARIMA (autoregressive integrated moving average) models are generalizations of the simple AR model that use three tools for modeling the serial correlation in the disturbance:

- The first tool is the autoregressive, or AR, term. The AR(1) model introduced above uses only the first-order term, but in general, you may use additional, higher-order AR terms. Each AR term corresponds to the use of a lagged value of the residual in the forecasting equation for the unconditional residual. An autoregressive model of order p , AR(p) has the form

$$u_t = \rho_1 u_{t-1} + \rho_2 u_{t-2} + \dots + \rho_p u_{t-p} + \epsilon_t.\tag{26.14}$$

- The second tool is the integration order term. Each integration order corresponds to differencing the series being forecast. A first-order integrated component means that the forecasting model is designed for the first difference of the original series. A second-order component corresponds to using second differences, and so on.
- The third tool is the MA, or moving average term. A moving average forecasting model uses lagged values of the forecast error to improve the current forecast. A first-order moving average term uses the most recent forecast error, a second-order term uses the forecast error from the two most recent periods, and so on. An MA(q) has the form:

$$u_t = \epsilon_t + \theta_1 \epsilon_{t-1} + \theta_2 \epsilon_{t-2} + \dots + \theta_q \epsilon_{t-q}.\tag{26.15}$$

Please be aware that some authors and software packages use the opposite sign convention for the θ coefficients so that the signs of the MA coefficients may be reversed.

The autoregressive and moving average specifications can be combined to form an ARMA(p, q) specification:

$$u_t = \rho_1 u_{t-1} + \rho_2 u_{t-2} + \dots + \rho_p u_{t-p} + \epsilon_t + \theta_1 \epsilon_{t-1} + \theta_2 \epsilon_{t-2} + \dots + \theta_q \epsilon_{t-q} \quad (26.16)$$

Although econometricians typically use ARIMA models applied to the residuals from a regression model, the specification can also be applied directly to a series. This latter approach provides a univariate model, specifying the conditional mean of the series as a constant, and measuring the residuals as differences of the series from its mean.

Principles of ARIMA Modeling (Box-Jenkins 1976)

In ARIMA forecasting, you assemble a complete forecasting model by using combinations of the three building blocks described above. The first step in forming an ARIMA model for a series of residuals is to look at its autocorrelation properties. You can use the correlogram view of a series for this purpose, as outlined in [“Correlogram” on page 324](#) of the *User’s Guide I*.

This phase of the ARIMA modeling procedure is called *identification* (not to be confused with the same term used in the simultaneous equations literature). The nature of the correlation between current values of residuals and their past values provides guidance in selecting an ARIMA specification.

The autocorrelations are easy to interpret—each one is the correlation coefficient of the current value of the series with the series lagged a certain number of periods. The partial autocorrelations are a bit more complicated; they measure the correlation of the current and lagged series after taking into account the predictive power of all the values of the series with smaller lags. The partial autocorrelation for lag 6, for example, measures the added predictive power of u_{t-6} when u_1, \dots, u_{t-5} are already in the prediction model. In fact, the partial autocorrelation is precisely the regression coefficient of u_{t-6} in a regression where the earlier lags are also used as predictors of u_t .

If you suspect that there is a distributed lag relationship between your dependent (left-hand) variable and some other predictor, you may want to look at their cross correlations before carrying out estimation.

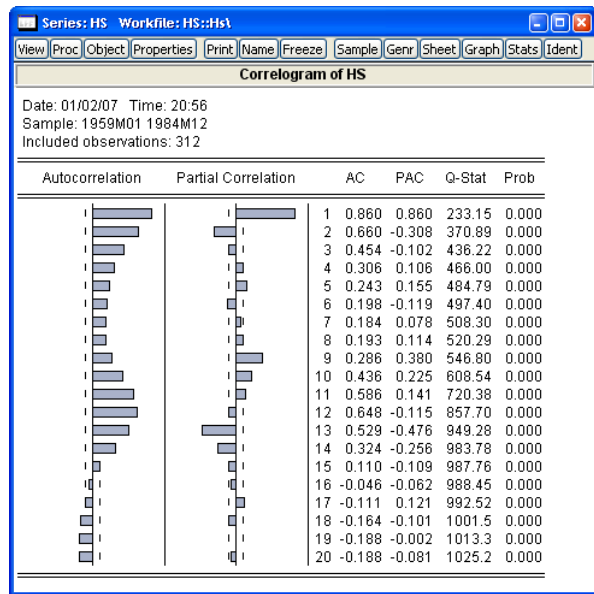
The next step is to decide what kind of ARIMA model to use. If the autocorrelation function dies off smoothly at a geometric rate, and the partial autocorrelations were zero after one lag, then a first-order autoregressive model is appropriate. Alternatively, if the autocorrelations were zero after one lag and the partial autocorrelations declined geometrically, a first-order moving average process would seem appropriate. If the autocorrelations appear to have a seasonal pattern, this would suggest the presence of a seasonal ARMA structure (see [“Seasonal ARMA Terms” on page 75](#)).

For example, we can examine the correlogram of the DRI Basics housing series in the HS.WF1 workfile by selecting **View/Correlogram...** from the HS series toolbar:

The “wavy” cyclical correlogram with a seasonal frequency suggests fitting a seasonal ARMA model to HS.

The goal of ARIMA analysis is a parsimonious representation of the process governing the residual. You should use only enough AR and MA terms to fit the properties of the residuals. The Akaike information criterion and Schwarz criterion provided with each set of estimates may also be used as a guide for the appropriate lag order selection.

After fitting a candidate ARIMA specification, you should verify that there are no remaining autocorrelations that your model has not accounted for. Examine the autocorrelations and the partial autocorrelations of the innovations (the residuals from the ARIMA model) to see if any important forecasting power has been overlooked. EViews provides views for diagnostic checks after estimation.



Estimating ARIMA Models

EViews estimates general ARIMA specifications that allow for right-hand side explanatory variables. Despite the fact that these models are sometimes termed ARIMAX specifications, we will refer to this general class of models as ARIMA.

To specify your ARIMA model, you will:

- Difference your dependent variable, if necessary, to account for the order of integration.
- Describe your structural regression model (dependent variables and regressors) and add any AR or MA terms, as described above.

Differencing

The d operator can be used to specify differences of series. To specify first differencing, simply include the series name in parentheses after d . For example, $d(gdp)$ specifies the first difference of GDP, or $GDP - GDP(-1)$.

More complicated forms of differencing may be specified with two optional parameters, n and s . `d(x, n)` specifies the n -th order difference of the series X :

$$d(x, n) = (1 - L)^n x, \quad (26.17)$$

where L is the lag operator. For example, `d(gdp, 2)` specifies the second order difference of GDP:

$$d(\text{gdp}, 2) = \text{gdp} - 2 * \text{gdp}(-1) + \text{gdp}(-2)$$

`d(x, n, s)` specifies n -th order ordinary differencing of X with a seasonal difference at lag s :

$$d(x, n, s) = (1 - L)^n (1 - L^s) x. \quad (26.18)$$

For example, `d(gdp, 0, 4)` specifies zero ordinary differencing with a seasonal difference at lag 4, or $\text{GDP} - \text{GDP}(-4)$.

If you need to work in logs, you can also use the `dlog` operator, which returns differences in the log values. For example, `dlog(gdp)` specifies the first difference of $\log(\text{GDP})$ or $\log(\text{GDP}) - \log(\text{GDP}(-1))$. You may also specify the n and s options as described for the simple `d` operator, `dlog(x, n, s)`.

There are two ways to estimate integrated models in EViews. First, you may generate a new series containing the differenced data, and then estimate an ARMA model using the new data. For example, to estimate a Box-Jenkins ARIMA(1, 1, 1) model for M1, you can enter:

```
series dm1 = d(m1)
equation eq1.ls dm1 c ar(1) ma(1)
```

Alternatively, you may include the difference operator `d` directly in the estimation specification. For example, the same ARIMA(1,1,1) model can be estimated using the command:

```
equation eq1.ls d(m1) c ar(1) ma(1)
```

The latter method should generally be preferred for an important reason. If you define a new variable, such as DM1 above, and use it in your estimation procedure, then when you forecast from the estimated model, EViews will make forecasts of the dependent variable DM1. That is, you will get a forecast of the differenced series. If you are really interested in forecasts of the level variable, in this case M1, you will have to manually transform the forecasted value and adjust the computed standard errors accordingly. Moreover, if any other transformation or lags of M1 are included as regressors, EViews will not know that they are related to DM1. If, however, you specify the model using the difference operator expression for the dependent variable, `d(m1)`, the forecasting procedure will provide you with the option of forecasting the level variable, in this case M1.

The difference operator may also be used in specifying exogenous variables and can be used in equations without ARMA terms. Simply include them in the list of regressors in addition to the endogenous variables. For example:

```
d(cs,2) c d(gdp,2) d(gdp(-1),2) d(gdp(-2),2) time
```

is a valid specification that employs the difference operator on both the left-hand and right-hand sides of the equation.

ARMA Terms

The AR and MA parts of your model will be specified using the keywords `ar` and `ma` as part of the equation. We have already seen examples of this approach in our specification of the AR terms above, and the concepts carry over directly to MA terms.

For example, to estimate a second-order autoregressive and first-order moving average error process ARMA(2,1), you would include expressions for the AR(1), AR(2), and MA(1) terms along with your other regressors:

```
c gov ar(1) ar(2) ma(1)
```

Once again, you need not use the AR and MA terms consecutively. For example, if you want to fit a fourth-order autoregressive model to take account of seasonal movements, you could use AR(4) by itself:

```
c gov ar(4)
```

You may also specify a pure moving average model by using only MA terms. Thus:

```
c gov ma(1) ma(2)
```

indicates an MA(2) model for the residuals.

The traditional Box-Jenkins or ARMA models do not have any right-hand side variables except for the constant. In this case, your list of regressors would just contain a C in addition to the AR and MA terms. For example:

```
c ar(1) ar(2) ma(1) ma(2)
```

is a standard Box-Jenkins ARMA (2,2).

Seasonal ARMA Terms

Box and Jenkins (1976) recommend the use of seasonal autoregressive (SAR) and seasonal moving average (SMA) terms for monthly or quarterly data with systematic seasonal movements. A SAR(p) term can be included in your equation specification for a seasonal autoregressive term with lag p . The lag polynomial used in estimation is the product of the one specified by the AR terms and the one specified by the SAR terms. The purpose of the SAR is to allow you to form the product of lag polynomials.

Similarly, $\text{SMA}(q)$ can be included in your specification to specify a seasonal moving average term with lag q . The lag polynomial used in estimation is the product of the one defined by the MA terms and the one specified by the SMA terms. As with the SAR, the SMA term allows you to build up a polynomial that is the product of underlying lag polynomials.

For example, a second-order AR process without seasonality is given by,

$$u_t = \rho_1 u_{t-1} + \rho_2 u_{t-2} + \epsilon_t, \quad (26.19)$$

which can be represented using the lag operator L , $L^n x_t = x_{t-n}$ as:

$$(1 - \rho_1 L - \rho_2 L^2)u_t = \epsilon_t. \quad (26.20)$$

You can estimate this process by including `ar(1)` and `ar(2)` terms in the list of regressors. With quarterly data, you might want to add a `sar(4)` expression to take account of seasonality. If you specify the equation as,

`sales c inc ar(1) ar(2) sar(4)`

then the estimated error structure would be:

$$(1 - \rho_1 L - \rho_2 L^2)(1 - \theta L^4)u_t = \epsilon_t. \quad (26.21)$$

The error process is equivalent to:

$$u_t = \rho_1 u_{t-1} + \rho_2 u_{t-2} + \theta u_{t-4} - \theta \rho_1 u_{t-5} - \theta \rho_2 u_{t-6} + \epsilon_t. \quad (26.22)$$

The parameter θ is associated with the seasonal part of the process. Note that this is an AR(6) process with nonlinear restrictions on the coefficients.

As another example, a second-order MA process without seasonality may be written,

$$u_t = \epsilon_t + \theta_1 \epsilon_{t-1} + \theta_2 \epsilon_{t-2}, \quad (26.23)$$

or using lag operators:

$$u_t = (1 + \theta_1 L + \theta_2 L^2)\epsilon_t. \quad (26.24)$$

You may estimate this second-order process by including both the MA(1) and MA(2) terms in your equation specification.

With quarterly data, you might want to add `sma(4)` to take account of seasonality. If you specify the equation as,

`cs c ad ma(1) ma(2) sma(4)`

then the estimated model is:

$$\begin{aligned} CS_t &= \beta_1 + \beta_2 A D_t + u_t \\ u_t &= (1 + \theta_1 L + \theta_2 L^2)(1 + \omega L^4)\epsilon_t \end{aligned} \quad (26.25)$$

The error process is equivalent to:

$$u_t = \epsilon_t + \theta_1 \epsilon_{t-1} + \theta_2 \epsilon_{t-2} + \omega \epsilon_{t-4} + \omega \theta_1 \epsilon_{t-5} + \omega \theta_2 \epsilon_{t-6}. \quad (26.26)$$

The parameter w is associated with the seasonal part of the process. This is just an MA(6) process with nonlinear restrictions on the coefficients. You can also include both SAR and SMA terms.

Output from ARIMA Estimation

The output from estimation with AR or MA specifications is the same as for ordinary least squares, with the addition of a lower block that shows the reciprocal roots of the AR and MA polynomials. If we write the general ARMA model using the lag polynomial $\rho(L)$ and $\theta(L)$ as,

$$\rho(L)u_t = \theta(L)\epsilon_t, \quad (26.27)$$

then the reported roots are the roots of the polynomials:

$$\rho(x^{-1}) = 0 \quad \text{and} \quad \theta(x^{-1}) = 0. \quad (26.28)$$

The roots, which may be imaginary, should have modulus no greater than one. The output will display a warning message if any of the roots violate this condition.

If ρ has a real root whose absolute value exceeds one or a pair of complex reciprocal roots outside the unit circle (that is, with modulus greater than one), it means that the autoregressive process is explosive.

If θ has reciprocal roots outside the unit circle, we say that the MA process is *noninvertible*, which makes interpreting and using the MA results difficult. However, noninvertibility poses no substantive problem, since as Hamilton (1994a, p. 65) notes, there is always an equivalent representation for the MA model where the reciprocal roots lie inside the unit circle. Accordingly, you should re-estimate your model with different starting values until you get a moving average process that satisfies invertibility. Alternatively, you may wish to turn off MA backcasting (see [“Backcasting MA terms” on page 80](#)).

If the estimated MA process has roots with modulus close to one, it is a sign that you may have over-differenced the data. The process will be difficult to estimate and even more difficult to forecast. If possible, you should re-estimate with one less round of differencing.

Consider the following example output from ARMA estimation:

Dependent Variable: R
Method: Least Squares
Date: 08/07/06 Time: 16:08
Sample (adjusted): 1954M06 1993M07
Included observations: 470 after adjustments
Convergence achieved after 15 iterations
MA Backcast: 1953M08 1953M12

| | Coefficient | Std. Error | t-Statistic | Prob. |
|--------------------|-------------|-----------------------|-------------|-----------|
| C | 8.638339 | 1.200586 | 7.195100 | 0.0000 |
| AR(1) | 0.982693 | 0.011018 | 89.18928 | 0.0000 |
| SAR(4) | 0.965494 | 0.017062 | 56.58647 | 0.0000 |
| MA(1) | 0.510996 | 0.040194 | 12.71322 | 0.0000 |
| SMA(4) | -0.979696 | 0.009688 | -101.1235 | 0.0000 |
| R-squared | 0.991576 | Mean dependent var | 6.978830 | |
| Adjusted R-squared | 0.991504 | S.D. dependent var | 2.919607 | |
| S.E. of regression | 0.269115 | Akaike info criterion | 0.223228 | |
| Sum squared resid | 33.67674 | Schwarz criterion | 0.267406 | |
| Log likelihood | -47.45869 | Hannan-Quinn criter. | 0.240609 | |
| F-statistic | 13683.92 | Durbin-Watson stat | 2.100664 | |
| Prob(F-statistic) | 0.000000 | | | |
| Inverted AR Roots | .99 | .98 | -.00+.99i | -.00-.99i |
| | -.99 | | | |
| Inverted MA Roots | .99 | -.00+.99i | -.00-.99i | -.51 |
| | -.99 | | | |

This estimation result corresponds to the following specification,

$$y_t = 8.64 + u_t$$

$$(1 - 0.98L)(1 - 0.97L^4)u_t = (1 + 0.51L)(1 - 0.98L^4)\epsilon_t \quad (26.29)$$

or equivalently, to:

$$y_t = 0.0052 + 0.98y_{t-1} + 0.97y_{t-4} - 0.95y_{t-5} + \epsilon_t$$

$$+ 0.51\epsilon_{t-1} - 0.98\epsilon_{t-4} - 0.50\epsilon_{t-5} \quad (26.30)$$

Note that the signs of the MA terms may be reversed from those in textbooks. Note also that the inverted roots have moduli very close to one, which is typical for many macro time series models.

Estimation Options

ARMA estimation employs the same nonlinear estimation techniques described earlier for AR estimation. These nonlinear estimation techniques are discussed further in [Chapter 25](#). “[Additional Regression Methods](#),” on page 44.

You may need to use the **Estimation Options** dialog box to control the iterative process. EViews provides a number of options that allow you to control the iterative procedure of the estimation algorithm. In general, you can rely on the EViews choices, but on occasion you may wish to override the default settings.

Iteration Limits and Convergence Criterion

Controlling the maximum number of iterations and convergence criterion are described in detail in “[Iteration and Convergence Options](#)” on page 627.

Derivative Methods

EViews always computes the derivatives of AR coefficients analytically and the derivatives of the MA coefficients using finite difference numeric derivative methods. For other coefficients in the model, EViews provides you with the option of computing analytic expressions for derivatives of the regression equation (if possible) or computing finite difference numeric derivatives in cases where the derivative is not constant. Furthermore, you can choose whether to favor speed of computation (fewer function evaluations) or whether to favor accuracy (more function evaluations) in the numeric derivative computation.

Starting Values for ARMA Estimation

As discussed above, models with AR or MA terms are estimated by nonlinear least squares. Nonlinear estimation techniques require starting values for all coefficient estimates. Normally, EViews determines its own starting values and for the most part this is an issue that you need not be concerned about. However, there are a few times when you may want to override the default starting values.

First, estimation will sometimes halt when the maximum number of iterations is reached, despite the fact that convergence is not achieved. Resuming the estimation with starting values from the previous step causes estimation to pick up where it left off instead of starting over. You may also want to try different starting values to ensure that the estimates are a global rather than a local minimum of the squared errors. You might also want to supply starting values if you have a good idea of what the answers should be, and want to speed up the estimation process.

To control the starting values for ARMA estimation, click on the **Options** button in the Equation Specification dialog. Among the options which EViews provides are several alternatives for setting starting values that you can see by accessing the drop-down menu labeled **Starting Coefficient Values for ARMA**.

EViews' default approach is **OLS/TSLS**, which runs a preliminary estimation without the ARMA terms and then starts nonlinear estimation from those values. An alternative is to use fractions of the OLS or TSLS coefficients as starting values. You can choose **.8**, **.5**, **.3**, or you can start with all coefficient values set equal to zero.

The final starting value option is **User Supplied**. Under this option, EViews uses the coefficient values that are in the coefficient vector. To set the starting values, open a window for the coefficient vector C by double clicking on the icon, and editing the values.

To properly set starting values, you will need a little more information about how EViews assigns coefficients for the ARMA terms. As with other estimation methods, when you specify your equation as a list of variables, EViews uses the built-in C coefficient vector. It assigns coefficient numbers to the variables in the following order:

- First are the coefficients of the variables, in order of entry.
- Next come the AR terms in the order you typed them.
- The SAR, MA, and SMA coefficients follow, in that order.

Thus the following two specifications will have their coefficients in the same order:

```
y c x ma(2) ma(1) sma(4) ar(1)
y sma(4) c ar(1) ma(2) x ma(1)
```

You may also assign values in the C vector using the `param` command:

```
param c(1) 50 c(2) .8 c(3) .2 c(4) .6 c(5) .1 c(6) .5
```

The starting values will be 50 for the constant, 0.8 for X, 0.2 for AR(1), 0.6 for MA(2), 0.1 for MA(1) and 0.5 for SMA(4). Following estimation, you can always see the assignment of coefficients by looking at the **Representations** view of your equation.

You can also fill the C vector from any estimated equation (without typing the numbers) by choosing **Proc/Update Coefs from Equation** in the equation toolbar.

Backcasting MA terms

Consider an MA(q) regression model of the form:

$$\begin{aligned} y_t &= X_t' \beta + u_t \\ u_t &= \epsilon_t + \theta_1 \epsilon_{t-1} + \theta_2 \epsilon_{t-2} + \dots + \theta_q \epsilon_{t-q} \end{aligned} \quad (26.31)$$

for $t = 1, 2, \dots, T$. Estimation of this model using conditional least squares requires computation of the innovations ϵ_t for each period in the estimation sample.

Computing the innovations is a straightforward process. Suppose we have an initial estimate of the coefficients, $(\hat{\beta}, \hat{\theta})$, and estimates of the pre-estimation sample values of ϵ :

$$\{\hat{\epsilon}_{-(q-1)}, \hat{\epsilon}_{-(q-2)}, \dots, \hat{\epsilon}_0\} \quad (26.32)$$

Then, after first computing the unconditional residuals $\hat{u}_t = y_t - X_t' \hat{\beta}$, we may use forward recursion to solve for the remaining values of the innovations:

$$\hat{\epsilon}_t = \hat{u}_t - \hat{\theta}_1 \hat{\epsilon}_{t-1} - \dots - \hat{\theta}_q \hat{\epsilon}_{t-q} \quad (26.33)$$

for $t = 1, 2, \dots, T$.

All that remains is to specify a method of obtaining estimates of the pre-sample values of ϵ :

$$\{\hat{\epsilon}_{-(q-1)}, \hat{\epsilon}_{-(q-2)}, \dots, \hat{\epsilon}_0\} \quad (26.34)$$

By default, EViews performs backcasting to obtain the pre-sample innovations (Box and Jenkins, 1976). As the name suggests, backcasting uses a backward recursion method to obtain estimates of ϵ for this period.

To start the recursion, the q values for the innovations *beyond* the estimation sample are set to zero:

$$\tilde{\epsilon}_{T+1} = \tilde{\epsilon}_{T+2} = \dots = \tilde{\epsilon}_{T+q} = 0 \quad (26.35)$$

EViews then uses the actual results to perform the backward recursion:

$$\tilde{\epsilon}_t = \hat{u}_t - \hat{\theta}_1 \tilde{\epsilon}_{t+1} - \dots - \hat{\theta}_q \tilde{\epsilon}_{t+q} \quad (26.36)$$

for $t = T, \dots, 0, \dots, -(q-1)$. The final q values, $\{\tilde{\epsilon}_0, \dots, \tilde{\epsilon}_{-(q-2)}, \tilde{\epsilon}_{-(q-1)}\}$, which we use as our estimates, may be termed the backcast estimates of the pre-sample innovations. (Note that if your model also includes AR terms, EViews will ρ -difference the \hat{u}_t to eliminate the serial correlation prior to performing the backcast.)

If backcasting is turned off, the values of the pre-sample ϵ are simply set to zero:

$$\hat{\epsilon}_{-(q-1)} = \dots = \hat{\epsilon}_0 = 0, \quad (26.37)$$

The sum of squared residuals (SSR) is formed as a function of the β and θ , using the fitted values of the lagged innovations:

$$\text{SSR}(\beta, \theta) = \sum_{t=q+1}^T (y_t - X_t' \beta - \theta_1 \hat{\epsilon}_{t-1} - \dots - \theta_q \hat{\epsilon}_{t-q})^2. \quad (26.38)$$

This expression is minimized with respect to β and θ .

The backcast step, forward recursion, and minimization procedures are repeated until the estimates of β and θ converge.

Dealing with Estimation Problems

Since EViews uses nonlinear least squares algorithms to estimate ARMA models, all of the discussion in [Chapter 25, “Solving Estimation Problems” on page 50](#), is applicable, especially the advice to try alternative starting values.

There are a few other issues to consider that are specific to estimation of ARMA models.

First, MA models are notoriously difficult to estimate. In particular, you should avoid high order MA terms unless absolutely required for your model as they are likely to cause estimation difficulties. For example, a single large spike at lag 57 in the correlogram does not necessarily require you to include an MA(57) term in your model unless you know there is something special happening every 57 periods. It is more likely that the spike in the correlogram is simply the product of one or more outliers in the series. By including many MA terms in your model, you lose degrees of freedom, and may sacrifice stability and reliability of your estimates.

If the underlying roots of the MA process have modulus close to one, you may encounter estimation difficulties, with EViews reporting that it cannot improve the sum-of-squares or that it failed to converge in the maximum number of iterations. This behavior may be a sign that you have over-differenced the data. You should check the correlogram of the series to determine whether you can re-estimate with one less round of differencing.

Lastly, if you continue to have problems, you may wish to turn off MA backcasting.

TSLS with ARIMA errors

Two-stage least squares or instrumental variable estimation with ARIMA poses no particular difficulties.

For a detailed discussion of how to estimate TSLS specifications with ARMA errors, see [“Two-stage Least Squares” on page 37](#).

Nonlinear Models with ARMA errors

EViews will estimate nonlinear ordinary and two-stage least squares models with autoregressive error terms. For details, see the extended discussion in [“Nonlinear Least Squares,” beginning on page 43](#).

EViews does not currently estimate nonlinear models with MA errors. You can, however, use the state space object to specify and estimate these models (see [“ARMAX\(2, 3\) with a Random Coefficient” on page 392](#)).

Weighted Models with ARMA errors

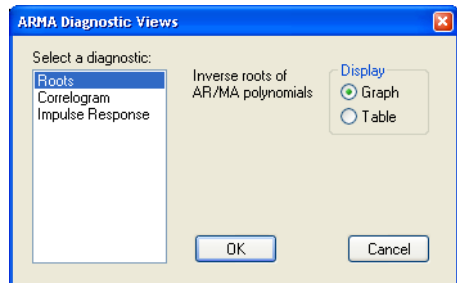
EViews does not have procedures to automatically estimate weighted models with ARMA error terms—if you add AR terms to a weighted model, the weighting series will be ignored. You can, of course, always construct the weighted series and then perform estimation using the weighted data and ARMA terms.

ARMA Equation Diagnostics

ARMA Structure

This set of views provides access to several diagnostic views that help you assess the structure of the ARMA portion of the estimated equation. The view is currently available only for models specified by list that includes at least one AR or MA term and estimated by least squares. There are three views available: roots, correlogram, and impulse response.

To display the ARMA structure, select **View/ARMA Structure...** from the menu of an estimated equation. If the equation type supports this view and there are no ARMA components in the specification, EViews will open the **ARMA Diagnostic Views** dialog:



On the left-hand side of the dialog, you will select one of the three types of diagnostics.

When you click on one of the types, the right-hand side of the dialog will change to show you the options for each type.

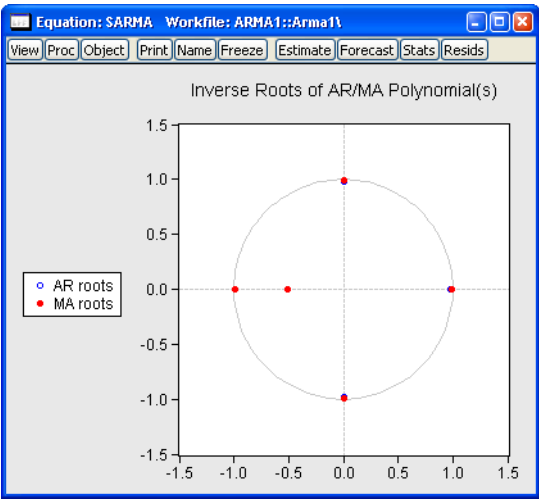
Roots

The roots view displays the inverse roots of the AR and/or MA characteristic polynomial. The roots may be displayed as a graph or as a table by selecting the appropriate radio button.

The graph view plots the roots in the complex plane where the horizontal axis is the real part and the vertical axis is the imaginary part of each root.

If the estimated ARMA process is (covariance) stationary, then all AR roots should lie *inside* the unit circle. If the estimated ARMA process is invertible, then all MA roots should lie *inside* the unit circle. The table view displays all roots in order of decreasing modulus (square root of the sum of squares of the real and imaginary parts).

For imaginary roots (which come in conjugate pairs), we also display the cycle corresponding to that root. The cycle is computed as $2\pi/a$, where $a = \text{atan}(i/r)$, and i and r are the imaginary and real parts of the root, respectively. The cycle for a real root is infinite and is not reported.



Inverse Roots of AR/MA Polynomial(s)
Specification: R C AR(1) SAR(4) MA(1) SMA(4)
Date: 08/07/06 Time: 16:17
Sample: 1954M01 1994M12
Included observations: 470

| AR Root(s) | Modulus | Cycle |
|----------------------|----------|----------|
| 0.991260 | 0.991260 | |
| 1.36e-20 ± 0.991260i | 0.991260 | 4.000000 |
| -0.991260 | 0.991260 | |
| 0.982693 | 0.982693 | |

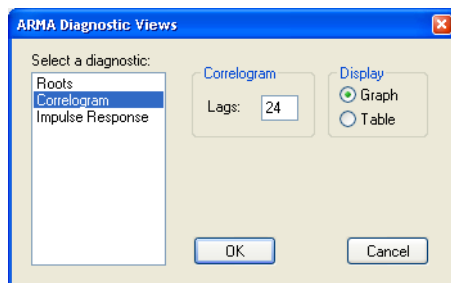
No root lies outside the unit circle.
ARMA model is stationary.

| MA Root(s) | Modulus | Cycle |
|-----------------------|----------|----------|
| -8.67e-18 ± 0.994885i | 0.994885 | 4.000000 |
| 0.994885 | 0.994885 | |
| -0.994885 | 0.994885 | |
| -0.510996 | 0.510996 | |

No root lies outside the unit circle.
ARMA model is invertible.

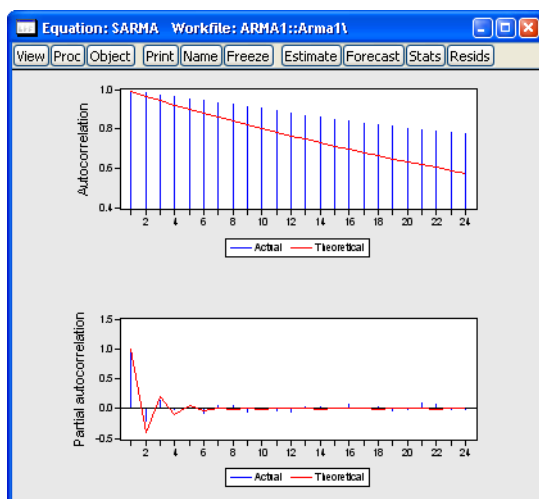
Correlogram

The correlogram view compares the autocorrelation pattern of the structural residuals and that of the estimated model for a specified number of periods (recall that the structural residuals are the residuals after removing the effect of the fitted exogenous regressors but *not* the ARMA terms). For a properly specified model, the residual and theoretical (estimated) autocorrelations and partial autocorrelations should be “close”.



To perform the comparison, simply select the **Correlogram** diagnostic, specify a number of lags to be evaluated, and a display format (**Graph** or **Table**).

Here, we have specified a graphical comparison over 24 periods/lags. The graph view plots the autocorrelations and partial autocorrelations of the sample structural residuals and those that are implied from the estimated ARMA parameters. If the estimated ARMA model is not stationary, only the sample second moments from the structural residuals are plotted.



The table view displays the numerical values for each of the second moments and the difference between from the estimated theoretical. If the estimated ARMA model is not stationary, the theoretical second moments implied from the estimated ARMA parameters will be filled with NAs.

Note that the table view starts from lag zero, while the graph view starts from lag one.

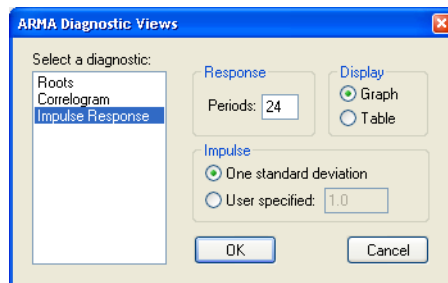
Impulse Response

The ARMA impulse response view traces the response of the ARMA part of the estimated equation to shocks in the innovation.

An impulse response function traces the response to a one-time shock in the innovation. The accumulated response is the accumulated sum of the impulse responses. It can be inter-

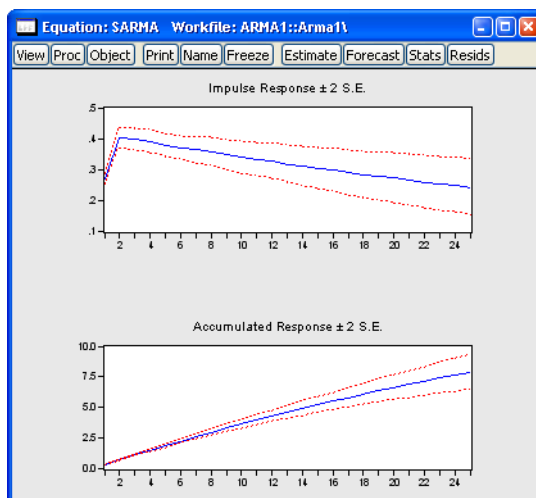
puted as the response to step impulse where the same shock occurs in every period from the first.

To compute the impulse response (and accumulated responses), select the **Impulse Response** diagnostic, enter the number of periods, and display type, and define the shock. For the latter, you have the choice of using a one standard deviation shock (using the standard error of the regression for the estimated equation), or providing a user specified value. Note that if you select a one standard deviation shock, EViews will take account of innovation uncertainty when estimating the standard errors of the responses.



If the estimated ARMA model is stationary, the impulse responses will asymptote to zero, while the accumulated responses will asymptote to its long-run value. These asymptotic values will be shown as dotted horizontal lines in the graph view.

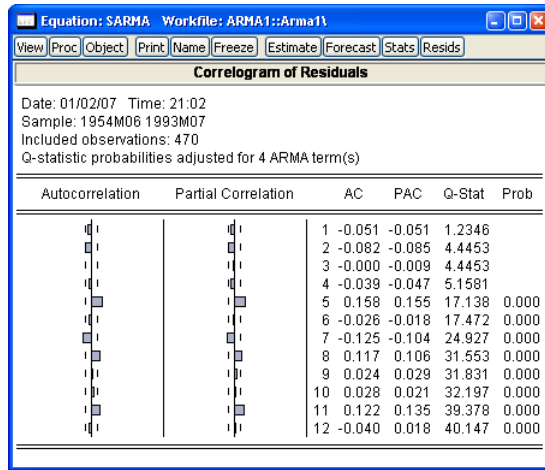
For a highly persistent near unit root but stationary process, the asymptotes may not be drawn in the graph for a short horizon. For a table view, the asymptotic values (together with its standard errors) will be shown at the bottom of the table. If the estimated ARMA process is not stationary, the asymptotic values will not be displayed since they do not exist.



Q-statistics

If your ARMA model is correctly specified, the residuals from the model should be nearly white noise. This means that there should be no serial correlation left in the residuals. The Durbin-Watson statistic reported in the regression output is a test for AR(1) in the absence of lagged dependent variables on the right-hand side. As discussed in [“Correlograms and Q-statistics” on page 65](#), more general tests for serial correlation in the residuals may be carried out with **View/Residual Tests/Correlogram-Q-statistic** and **View/Residual Tests/Serial Correlation LM Test....**

For the example seasonal ARMA model, the 12-period residual correlogram looks as follows:



The correlogram has a significant spike at lag 5, and all subsequent Q -statistics are highly significant. This result clearly indicates the need for respecification of the model.

Nonstationary Time Series

The theory behind ARMA estimation is based on stationary time series. A series is said to be (weakly or covariance) *stationary* if the mean and autocovariances of the series do not depend on time. Any series that is not stationary is said to be *nonstationary*.

A common example of a nonstationary series is the *random walk*:

$$y_t = y_{t-1} + \epsilon_t, \quad (26.39)$$

where ϵ is a stationary random disturbance term. The series y has a constant forecast value, conditional on t , and the variance is increasing over time. The random walk is a difference stationary series since the first difference of y is stationary:

$$y_t - y_{t-1} = (1 - L)y_t = \epsilon_t. \quad (26.40)$$

A difference stationary series is said to be *integrated* and is denoted as $I(d)$ where d is the order of integration. The order of integration is the number of unit roots contained in the series, or the number of differencing operations it takes to make the series stationary. For the random walk above, there is one unit root, so it is an $I(1)$ series. Similarly, a stationary series is $I(0)$.

Standard inference procedures do not apply to regressions which contain an integrated dependent variable or integrated regressors. Therefore, it is important to check whether a

series is stationary or not before using it in a regression. The formal method to test the stationarity of a series is the unit root test.

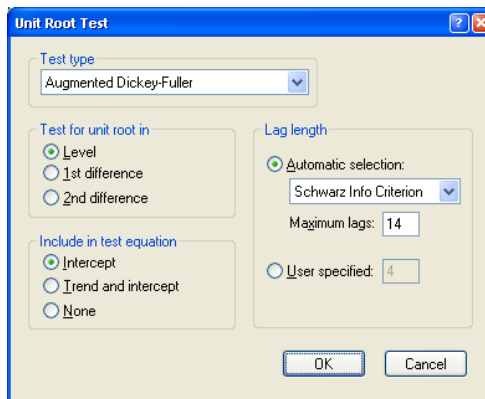
Unit Root Tests

EViews provides you with a variety of powerful tools for testing a series (or the first or second difference of the series) for the presence of a unit root. In addition to the existing Augmented Dickey-Fuller (1979) and Phillips-Perron (1988) tests, EViews now allows you to compute the GLS-detrended Dickey-Fuller (Elliot, Rothenberg, and Stock, 1996), Kwiatkowski, Phillips, Schmidt, and Shin (KPSS, 1992), Elliott, Rothenberg, and Stock Point Optimal (ERS, 1996), and Ng and Perron (NP, 2001) unit root tests. All of these tests are available as a view of a series.

Performing Unit Root Tests in EViews

The following discussion assumes that you are familiar with the basic forms of the unit root tests and the associated options. We provide theoretical background for these tests in [“Basic Unit Root Theory,” beginning on page 92](#), and document the settings used when performing these tests.

To begin, double click on the series name to open the series window, and choose **View/Unit Root Test...**



You must specify four sets of options to carry out a unit root test. The first three settings (on the left-hand side of the dialog) determine the basic form of the unit root test. The fourth set of options (on the right-hand side of the dialog) consist of test-specific advanced settings. You only need concern yourself with these settings if you wish to customize the calculation of your unit root test.

First, you should use the topmost combo box to select the type of unit root test that you wish to perform. You may choose one of six tests: ADF, DFGLS, PP, KPSS, ERS, and NP.

Next, specify whether you wish to test for a unit root in the level, first difference, or second difference of the series.

Lastly, choose your exogenous regressors. You can choose to include a constant, a constant and linear trend, or neither (there are limitations on these choices for some of the tests).

You can click on **OK** to compute the test using the specified settings, or you can customize your test using the advanced settings portion of the dialog.

The advanced settings for both the ADF and DFGLS tests allow you to specify how lagged difference terms p are to be included in the ADF test equation. You may choose to let EViews automatically select p , or you may specify a fixed positive integer value (if you choose automatic selection, you are given the additional option of selecting both the information criterion and maximum number of lags to be used in the selection procedure).

In this case, we have chosen to estimate an ADF test that includes a constant in the test regression and employs automatic lag length selection using a Schwarz Information Criterion (BIC) and a maximum lag length of 14. Applying these settings to data on the U. S. one-month Treasury bill rate for the period from March 1953 to July 1971, we can replicate Example 9.2 of Hayashi (2000, p. 596). The results are described below.

The first part of the unit root output provides information about the form of the test (the type of test, the exogenous variables, and lag length used), and contains the test output, associated critical values, and in this case, the p -value:

| Null Hypothesis: TBILL has a unit root | | |
|---|-------------|--------|
| Exogenous: Constant | | |
| Lag Length: 1 (Automatic based on SIC, MAXLAG=14) | | |
| | t-Statistic | Prob.* |
| Augmented Dickey-Fuller test statistic | -1.417410 | 0.5734 |
| Test critical values: 1% level | -3.459898 | |
| 5% level | -2.874435 | |
| 10% level | -2.573719 | |

*MacKinnon (1996) one-sided p -values.

The ADF statistic value is -1.417 and the associated one-sided p -value (for a test with 221 observations) is .573. In addition, EViews reports the critical values at the 1%, 5% and 10% levels. Notice here that the statistic t_α value is greater than the critical values so that we do not reject the null at conventional test sizes.

The second part of the output shows the intermediate test equation that EViews used to calculate the ADF statistic:

Augmented Dickey-Fuller Test Equation

Dependent Variable: D(TBILL)

Method: Least Squares

Date: 08/08/06 Time: 13:55

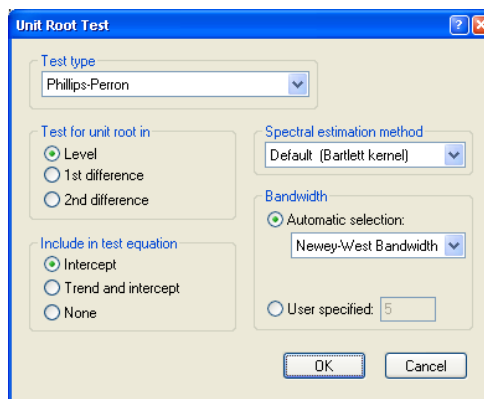
Sample: 1953M03 1971M07

Included observations: 221

| | Coefficient | Std. Error | t-Statistic | Prob. |
|--------------------|-------------|-----------------------|-------------|--------|
| TBILL(-1) | -0.022951 | 0.016192 | -1.417410 | 0.1578 |
| D(TBILL(-1)) | -0.203330 | 0.067007 | -3.034470 | 0.0027 |
| C | 0.088398 | 0.056934 | 1.552626 | 0.1220 |
| R-squared | 0.053856 | Mean dependent var | 0.013826 | |
| Adjusted R-squared | 0.045175 | S.D. dependent var | 0.379758 | |
| S.E. of regression | 0.371081 | Akaike info criterion | 0.868688 | |
| Sum squared resid | 30.01882 | Schwarz criterion | 0.914817 | |
| Log likelihood | -92.99005 | Hannan-Quinn criter. | 0.887314 | |
| F-statistic | 6.204410 | Durbin-Watson stat | 1.976361 | |
| Prob(F-statistic) | 0.002395 | | | |

If you had chosen to perform any of the other unit root tests (PP, KPSS, ERS, NP), the right side of the dialog would show the different options associated with the specified test. The options are associated with the method used to estimate the zero frequency spectrum term, f_0 , that is used in constructing the particular test statistic. As before, you only need pay attention to these settings if you wish to change from the EViews defaults.

Here, we have selected the PP test in the combo box. Note that the right-hand side of the dialog has changed, and now features a combo box for selecting the spectral estimation method. You may use this combo box to choose between various kernel or AR regression based estimators for f_0 . The entry labeled “Default” will show you the default estimator for the specific unit root test—in this example, we see that the PP default uses a kernel sum-of-covariances estimator with Bartlett weights. Alternately, if you had selected a NP test, the default entry would be “AR spectral-GLS”.



Lastly, you can control the lag length or bandwidth used for your spectral estimator. If you select one of the kernel estimation methods (Bartlett, Parzen, Quadratic Spectral), the dialog

will give you a choice between using Newey-West or Andrews automatic bandwidth selection methods, or providing a user specified bandwidth. If you choose one of the AR spectral density estimation methods (AR Spectral - OLS, AR Spectral - OLS detrended, AR Spectral - GLS detrended), the dialog will prompt you to choose from various automatic lag length selection methods (using information criteria) or to provide a user-specified lag length. See [“Automatic Bandwidth and Lag Length Selection” on page 99](#).

Once you have chosen the appropriate settings for your test, click on the **OK** button. EViews reports the test statistic along with output from the corresponding test regression. For these tests, EViews reports the uncorrected estimate of the residual variance and the estimate of the frequency zero spectrum f_0 (labeled as the “HAC corrected variance”) in addition to the basic output. Running a PP test using the TBILL series yields:

| | | |
|---|-------------|----------|
| Null Hypothesis: TBILL has a unit root | | |
| Exogenous: Constant | | |
| Bandwidth: 3.82 (Andrews using Bartlett kernel) | | |
| | Adj. t-Stat | Prob.* |
| Phillips-Perron test statistic | -1.519035 | 0.5223 |
| Test critical values: 1% level | -3.459898 | |
| 5% level | -2.874435 | |
| 10% level | -2.573719 | |
| *MacKinnon (1996) one-sided p-values. | | |
| Residual variance (no correction) | | 0.141569 |
| HAC corrected variance (Bartlett kernel) | | 0.107615 |

As with the ADF test, we fail to reject the null hypothesis of a unit root in the TBILL series at conventional significance levels.

Note that your test output will differ somewhat for alternative test specifications. For example, the KPSS output only provides the asymptotic critical values tabulated by KPSS:

| | | |
|--|-----------|----------|
| Null Hypothesis: TBILL is stationary | | |
| Exogenous: Constant | | |
| Bandwidth: 11 (Newey-West Fixed using Bartlett kernel) | | |
| | | LM-Stat. |
| Kwiatkowski-Phillips-Schmidt-Shin test statistic | | 1.537310 |
| Asymptotic critical values*: | | |
| | 1% level | 0.739000 |
| | 5% level | 0.463000 |
| | 10% level | 0.347000 |
| *Kwiatkowski-Phillips-Schmidt-Shin (1992, Table 1) | | |
| Residual variance (no correction) | | 2.415060 |
| HAC corrected variance (Bartlett kernel) | | 26.11028 |

Similarly, the NP test output will contain results for all four test statistics, along with the NP tabulated critical values.

A word of caution. You should note that the critical values reported by EViews are valid only for unit root tests of a data series, and will be invalid if the series is based on estimated values. For example, Engle and Granger (1987) proposed a two-step method of testing for cointegration which looks for a unit root in the residuals of a first-stage regression. Since these residuals are estimates of the disturbance term, the asymptotic distribution of the test statistic differs from the one for ordinary series. The correct critical values for a subset of the tests may be found in Davidson and MacKinnon (1993, Table 20.2).

Basic Unit Root Theory

The following discussion outlines the basic features of unit root tests. By necessity, the discussion will be brief. Users who require detail should consult the original sources and standard references (see, for example, Davidson and MacKinnon, 1993, Chapter 20, Hamilton, 1994a, Chapter 17, and Hayashi, 2000, Chapter 9).

Consider a simple AR(1) process:

$$y_t = \rho y_{t-1} + x_t' \delta + \epsilon_t, \quad (26.41)$$

where x_t are optional exogenous regressors which may consist of constant, or a constant and trend, ρ and δ are parameters to be estimated, and the ϵ_t are assumed to be white noise. If $|\rho| \geq 1$, y is a nonstationary series and the variance of y increases with time and approaches infinity. If $|\rho| < 1$, y is a (trend-)stationary series. Thus, the hypothesis of (trend-)stationarity can be evaluated by testing whether the absolute value of ρ is strictly less than one.

The unit root tests that EViews provides generally test the null hypothesis $H_0: \rho = 1$ against the one-sided alternative $H_1: \rho < 1$. In some cases, the null is tested against a point alternative. In contrast, the KPSS Lagrange Multiplier test evaluates the null of $H_0: \rho < 1$ against the alternative $H_1: \rho = 1$.

The Augmented Dickey-Fuller (ADF) Test

The standard DF test is carried out by estimating [Equation \(26.41\)](#) after subtracting y_{t-1} from both sides of the equation:

$$\Delta y_t = \alpha y_{t-1} + x_t' \delta + \epsilon_t, \quad (26.42)$$

where $\alpha = \rho - 1$. The null and alternative hypotheses may be written as,

$$\begin{aligned} H_0: \alpha &= 0 \\ H_1: \alpha &< 0 \end{aligned} \quad (26.43)$$

and evaluated using the conventional t -ratio for α :

$$t_\alpha = \hat{\alpha} / (se(\hat{\alpha})) \quad (26.44)$$

where $\hat{\alpha}$ is the estimate of α , and $se(\hat{\alpha})$ is the coefficient standard error.

Dickey and Fuller (1979) show that under the null hypothesis of a unit root, this statistic does not follow the conventional Student's t -distribution, and they derive asymptotic results and simulate critical values for various test and sample sizes. More recently, MacKinnon (1991, 1996) implements a much larger set of simulations than those tabulated by Dickey and Fuller. In addition, MacKinnon estimates response surfaces for the simulation results, permitting the calculation of Dickey-Fuller critical values and p -values for arbitrary sample sizes. The more recent MacKinnon critical value calculations are used by EViews in constructing test output.

The simple Dickey-Fuller unit root test described above is valid only if the series is an AR(1) process. If the series is correlated at higher order lags, the assumption of white noise disturbances ϵ_t is violated. The Augmented Dickey-Fuller (ADF) test constructs a parametric correction for higher-order correlation by assuming that the y series follows an AR(p) process and adding p lagged difference terms of the dependent variable y to the right-hand side of the test regression:

$$\Delta y_t = \alpha y_{t-1} + x_t' \delta + \beta_1 \Delta y_{t-1} + \beta_2 \Delta y_{t-2} + \dots + \beta_p \Delta y_{t-p} + v_t. \quad (26.45)$$

This augmented specification is then used to test (26.43) using the t -ratio (26.44). An important result obtained by Fuller is that the asymptotic distribution of the t -ratio for α is independent of the number of lagged first differences included in the ADF regression. Moreover, while the assumption that y follows an autoregressive (AR) process may seem restrictive, Said and Dickey (1984) demonstrate that the ADF test is asymptotically valid in the presence of a moving average (MA) component, provided that sufficient lagged difference terms are included in the test regression.

You will face two practical issues in performing an ADF test. First, you must choose whether to include exogenous variables in the test regression. You have the choice of including a constant, a constant and a linear time trend, or neither in the test regression. One approach would be to run the test with both a constant and a linear trend since the other two cases are just special cases of this more general specification. However, including irrelevant regressors in the regression will reduce the power of the test to reject the null of a unit root. The standard recommendation is to choose a specification that is a plausible description of the data under both the null and alternative hypotheses. See Hamilton (1994a, p. 501) for discussion.

Second, you will have to specify the number of lagged difference terms (which we will term the “lag length”) to be added to the test regression (0 yields the standard DF test; integers greater than 0 correspond to ADF tests). The usual (though not particularly useful) advice is to include a number of lags sufficient to remove serial correlation in the residuals. EViews provides both automatic and manual lag length selection options. For details, see [“Automatic Bandwidth and Lag Length Selection,” beginning on page 99](#).

Dickey-Fuller Test with GLS Detrending (DFGLS)

As noted above, you may elect to include a constant, or a constant and a linear time trend, in your ADF test regression. For these two cases, ERS (1996) propose a simple modification of the ADF tests in which the data are detrended so that explanatory variables are “taken out” of the data prior to running the test regression.

ERS define a quasi-difference of y_t that depends on the value a representing the specific point alternative against which we wish to test the null:

$$d(y_t|a) = \begin{cases} y_t & \text{if } t = 1 \\ y_t - ay_{t-1} & \text{if } t > 1 \end{cases} \quad (26.46)$$

Next, consider an OLS regression of the quasi-differenced data $d(y_t|a)$ on the quasi-differenced $d(x_t|a)$:

$$d(y_t|a) = d(x_t|a)' \hat{\delta}(a) + \eta_t \quad (26.47)$$

where x_t contains either a constant, or a constant and trend, and let $\hat{\delta}(a)$ be the OLS estimates from this regression.

All that we need now is a value for a . ERS recommend the use of $a = \bar{a}$, where:

$$\bar{a} = \begin{cases} 1 - 7/T & \text{if } x_t = \{1\} \\ 1 - 13.5/T & \text{if } x_t = \{1, t\} \end{cases} \quad (26.48)$$

We now define the GLS detrended data, y_t^d using the estimates associated with the \bar{a} :

$$y_t^d \equiv y_t - x_t' \hat{\delta}(\bar{a}) \quad (26.49)$$

Then the DFGLS test involves estimating the standard ADF test equation, (26.45), after substituting the GLS detrended y_t^d for the original y_t :

$$\Delta y_t^d = \alpha y_{t-1}^d + \beta_1 \Delta y_{t-1}^d + \dots + \beta_p y_{t-p}^d + v_t \quad (26.50)$$

Note that since the y_t^d are detrended, we do not include the x_t in the DFGLS test equation. As with the ADF test, we consider the t -ratio for $\hat{\alpha}$ from this test equation.

While the DFGLS t -ratio follows a Dickey-Fuller distribution in the constant only case, the asymptotic distribution differs when you include both a constant and trend. ERS (1996, Table 1, p. 825) simulate the critical values of the test statistic in this latter setting for $T = \{50, 100, 200, \infty\}$. Thus, the EViews lower tail critical values use the MacKinnon simulations for the no constant case, but are interpolated from the ERS simulated values for the constant and trend case. The null hypothesis is rejected for values that fall below these critical values.

The Phillips-Perron (PP) Test

Phillips and Perron (1988) propose an alternative (nonparametric) method of controlling for serial correlation when testing for a unit root. The PP method estimates the non-augmented DF test equation (26.42), and modifies the t -ratio of the α coefficient so that serial correlation does not affect the asymptotic distribution of the test statistic. The PP test is based on the statistic:

$$\tilde{t}_\alpha = t_\alpha \left(\frac{\gamma_0}{f_0} \right)^{1/2} - \frac{T(f_0 - \gamma_0)(se(\hat{\alpha}))}{2f_0^{1/2}s} \quad (26.51)$$

where $\hat{\alpha}$ is the estimate, and t_α the t -ratio of α , $se(\hat{\alpha})$ is coefficient standard error, and s is the standard error of the test regression. In addition, γ_0 is a consistent estimate of the error variance in (26.42) (calculated as $(T - k)s^2 / T$, where k is the number of regressors). The remaining term, f_0 , is an estimator of the residual spectrum at frequency zero.

There are two choices you will have to make when performing the PP test. First, you must choose whether to include a constant, a constant and a linear time trend, or neither, in the test regression. Second, you will have to choose a method for estimating f_0 . EViews supports estimators for f_0 based on kernel-based sum-of-covariances, or on autoregressive spectral density estimation. See “Frequency Zero Spectrum Estimation,” beginning on page 97 for details.

The asymptotic distribution of the PP modified t -ratio is the same as that of the ADF statistic. EViews reports MacKinnon lower-tail critical and p -values for this test.

The Kwiatkowski, Phillips, Schmidt, and Shin (KPSS) Test

The KPSS (1992) test differs from the other unit root tests described here in that the series y_t is assumed to be (trend-) stationary under the null. The KPSS statistic is based on the residuals from the OLS regression of y_t on the exogenous variables x_t :

$$y_t = x_t' \delta + u_t \quad (26.52)$$

The LM statistic is defined as:

$$LM = \sum_t S(t)^2 / (T^2 f_0) \quad (26.53)$$

where f_0 is an estimator of the residual spectrum at frequency zero and where $S(t)$ is a cumulative residual function:

$$S(t) = \sum_{r=1}^t \hat{u}_r \quad (26.54)$$

based on the residuals $\hat{u}_t = y_t - x_t' \hat{\delta}(0)$. We point out that the estimator of δ used in this calculation differs from the estimator for δ used by GLS detrending since it is based on a regression involving the original data and not on the quasi-differenced data.

To specify the KPSS test, you must specify the set of exogenous regressors x_t and a method for estimating f_0 . See “Frequency Zero Spectrum Estimation” on page 97 for discussion.

The reported critical values for the LM test statistic are based upon the asymptotic results presented in KPSS (Table 1, p. 166).

Elliot, Rothenberg, and Stock Point Optimal (ERS) Test

The ERS Point Optimal test is based on the quasi-differencing regression defined in Equations (26.47). Define the residuals from (26.47) as $\hat{\eta}_t(a) = d(y_t|a) - d(x_t|a)' \hat{\delta}(a)$, and let $SSR(a) = \sum \hat{\eta}_t^2(a)$ be the sum-of-squared residuals function. The ERS (feasible) point optimal test statistic of the null that $\alpha = 1$ against the alternative that $\alpha = \bar{a}$, is then defined as:

$$P_T = (SSR(\bar{a}) - \bar{a}SSR(1))/f_0 \quad (26.55)$$

where f_0 , is an estimator of the residual spectrum at frequency zero.

To compute the ERS test, you must specify the set of exogenous regressors x_t and a method for estimating f_0 (see “Frequency Zero Spectrum Estimation” on page 97).

Critical values for the ERS test statistic are computed by interpolating the simulation results provided by ERS (1996, Table 1, p. 825) for $T = \{50, 100, 200, \infty\}$.

Ng and Perron (NP) Tests

Ng and Perron (2001) construct four test statistics that are based upon the GLS detrended data y_t^d . These test statistics are modified forms of Phillips and Perron Z_α and Z_t statistics, the Bhargava (1986) R_1 statistic, and the ERS Point Optimal statistic. First, define the term:

$$\kappa = \sum_{t=2}^T (y_{t-1}^d)^2 / T^2 \quad (26.56)$$

The modified statistics may then be written as,

$$\begin{aligned}
MZ_{\alpha}^d &= (T^{-1}(y_T^d)^2 - f_0)/(2\kappa) \\
MZ_t^d &= MZ_{\alpha} \times MSB \\
MSB^d &= (\kappa/f_0)^{1/2}
\end{aligned} \tag{26.57}$$

$$MP_T^d = \begin{cases} (\bar{c}^2 \kappa - \bar{c} T^{-1}(y_T^d)^2)/f_0 & \text{if } x_t = \{1\} \\ (\bar{c}^2 \kappa + (1 - \bar{c}) T^{-1}(y_T^d)^2)/f_0 & \text{if } x_t = \{1, t\} \end{cases}$$

where:

$$\bar{c} = \begin{cases} -7 & \text{if } x_t = \{1\} \\ -13.5 & \text{if } x_t = \{1, t\} \end{cases} \tag{26.58}$$

The NP tests require a specification for x_t and a choice of method for estimating f_0 (see [“Frequency Zero Spectrum Estimation” on page 97](#)).

Frequency Zero Spectrum Estimation

Many of the unit root tests described above require a consistent estimate of the residual spectrum at frequency zero. EViews supports two classes of estimators for f_0 : kernel-based sum-of-covariances estimators, and autoregressive spectral density estimators.

Kernel Sum-of-Covariances Estimation

The kernel-based estimator of the frequency zero spectrum is based on a weighted sum of the autocovariances, with the weights are defined by a kernel function. The estimator takes the form,

$$\hat{f}_0 = \sum_{j=-(T-1)}^{T-1} \hat{\gamma}(j) \cdot K(j/l) \tag{26.59}$$

where l is a bandwidth parameter (which acts as a truncation lag in the covariance weighting), K is a kernel function, and where $\hat{\gamma}(j)$, the j -th sample autocovariance of the residuals \tilde{u}_t , is defined as:

$$\hat{\gamma}(j) = \sum_{t=j+1}^T (\tilde{u}_t \tilde{u}_{t-j})/T \tag{26.60}$$

Note that the residuals \tilde{u}_t that EViews uses in estimating the autocovariance functions in [\(26.60\)](#) will differ depending on the specified unit root test:

| Unit root test | Source of \tilde{u}_t residuals for kernel estimator |
|---------------------------|--|
| ADF, DFGLS | <i>not applicable.</i> |
| PP, ERS Point Optimal, NP | residuals from the Dickey-Fuller test equation, (26.42). |
| KPSS | residuals from the OLS test equation, (26.52). |

EViews supports the following kernel functions:

| | |
|--------------------|---|
| Bartlett: | $K(x) = \begin{cases} 1 - x & \text{if } x \leq 1 \\ 0 & \text{otherwise} \end{cases}$ |
| Parzen: | $K(x) = \begin{cases} 1 - 6x^2 + 6 x ^3 & \text{if } 0 \leq x \leq (1/2) \\ 2(1 - x)^3 & \text{if } (1/2) < x \leq 1 \\ 0 & \text{otherwise} \end{cases}$ |
| Quadratic Spectral | $K(x) = \frac{25}{12\pi^2 x^2} \left(\frac{\sin(6\pi x/5)}{6\pi x/5} - \cos(6\pi x/5) \right)$ |

The properties of these kernels are described in Andrews (1991).

As with most kernel estimators, the choice of the bandwidth parameter l is of considerable importance. EViews allows you to specify a fixed parameter or to have EViews select one using a data-dependent method. Automatic bandwidth parameter selection is discussed in “Automatic Bandwidth and Lag Length Selection,” beginning on page 99.

Autoregressive Spectral Density Estimator

The autoregressive spectral density estimator at frequency zero is based upon the residual variance and estimated coefficients from the auxiliary regression:

$$\Delta \tilde{y}_t = \alpha \tilde{y}_{t-1} + \varphi \cdot \tilde{x}_t' \delta + \beta_1 \Delta \tilde{y}_{t-1} + \dots + \beta_p \Delta \tilde{y}_{t-p} + u_t \quad (26.61)$$

EViews provides three autoregressive spectral methods: OLS, OLS detrending, and GLS detrending, corresponding to difference choices for the data \tilde{y}_t . The following table summarizes the auxiliary equation estimated by the various AR spectral density estimators:

| AR spectral method | Auxiliary AR regression specification |
|--------------------|--|
| OLS | $\tilde{y}_t = y_t$, and $\varphi = 1$, $\tilde{x}_t = x_t$. |
| OLS detrended | $\tilde{y}_t = y_t - x_t' \hat{\delta}(0)$, and $\varphi = 0$. |
| GLS detrended | $\tilde{y}_t = y_t - x_t' \hat{\delta}(\bar{a}) = y_t^d$, and $\varphi = 0$. |

where $\hat{\delta}(a)$ are the coefficient estimates from the regression defined in (26.47).

The AR spectral estimator of the frequency zero spectrum is defined as:

$$\hat{f}_0 = \hat{\sigma}_u^2 / (1 - \hat{\beta}_1 - \hat{\beta}_2 - \dots - \hat{\beta}_p) \quad (26.62)$$

where $\hat{\sigma}_u^2 = \sum \tilde{u}_t^2 / T$ is the residual variance, and $\hat{\beta}$ are the estimates from (26.61). We note here that EViews uses the non-degree of freedom estimator of the residual variance. As a result, spectral estimates computed in EViews may differ slightly from those obtained from other sources.

Not surprisingly, the spectrum estimator is sensitive to the number of lagged difference terms in the auxiliary equation. You may either specify a fixed parameter or have EViews automatically select one based on an information criterion. Automatic lag length selection is examined in “Automatic Bandwidth and Lag Length Selection” on page 99.

Default Settings

By default, EViews will choose the estimator of f_0 used by the authors of a given test specification. You may, of course, override the default settings and choose from either family of estimation methods. The default settings are listed below:

| Unit root test | Frequency zero spectrum default method |
|-------------------|--|
| ADF, DFGLS | <i>not applicable</i> |
| PP, KPSS | Kernel (Bartlett) sum-of-covariances |
| ERS Point Optimal | AR spectral regression (OLS) |
| NP | AR spectral regression (GLS-detrended) |

Automatic Bandwidth and Lag Length Selection

There are three distinct situations in which EViews can automatically compute a bandwidth or a lag length parameter.

The first situation occurs when you are selecting the bandwidth parameter l for the kernel-based estimators of f_0 . For the kernel estimators, EViews provides you with the option of using the Newey-West (1994) or the Andrews (1991) data-based automatic bandwidth parameter methods. See the original sources for details. For those familiar with the Newey-West procedure, we note that EViews uses the lag selection parameter formulae given in the

corresponding first lines of Table II-C. The Andrews method is based on an AR(1) specification.

The latter two situations occur when the unit root test requires estimation of a regression with a parametric correction for serial correlation as in the ADF and DFGLS test equation regressions, and in the AR spectral estimator for f_0 . In all of these cases, p lagged difference terms are added to a regression equation. The automatic selection methods choose p (less than the specified maximum) to minimize one of the following criteria:

| Information criterion | Definition |
|-----------------------------|--|
| Akaike (AIC) | $-2(l/T) + 2k/T$ |
| Schwarz (SIC) | $-2(l/T) + k\log(T)/T$ |
| Hannan-Quinn (HQ) | $-2(l/T) + 2k\log(\log(T))/T$ |
| Modified AIC (MAIC) | $-2(l/T) + 2(k + \tau)/T$ |
| Modified SIC (MSIC) | $-2(l/T) + (k + \tau)\log(T)/T$ |
| Modified Hannan-Quinn (MHQ) | $-2(l/T) + 2(k + \tau)\log(\log(T))/T$ |

where the modification factor τ is computed as:

$$\tau = \alpha^2 \sum_t \tilde{y}_{t-1}^2 / \hat{\sigma}_u^2 \quad (26.63)$$

for $\tilde{y}_t = y_t$, when computing the ADF test equation, and for \tilde{y}_t as defined in “Autoregressive Spectral Density Estimator” on page 98, when estimating f_0 . NP (2001) propose and examine the modified criteria, concluding with a recommendation of the MAIC.

For the information criterion selection methods, you must also specify an upper bound to the lag length. By default, EViews chooses a maximum lag of:

$$k_{\max} = \text{int}(12(T/100)^{1/4}) \quad (26.64)$$

See Hayashi (2000, p. 594) for a discussion of the selection of this upper bound.

Panel Unit Root Tests

Recent literature suggests that panel-based unit root tests have higher power than unit root tests based on individual time series. EViews will compute one of the following five types of panel unit root tests: Levin, Lin and Chu (2002), Breitung (2000), Im, Pesaran and Shin (2003), Fisher-type tests using ADF and PP tests (Maddala and Wu (1999) and Choi (2001)), and Hadri (2000).

While these tests are commonly termed “panel unit root” tests, theoretically, they are simply multiple-series unit root tests that have been applied to panel data structures (where the

presence of cross-sections generates “multiple series” out of a single series). Accordingly, EViews supports these tests in settings involving multiple series: as a series view (if the workfile is panel structured), as a group view, or as a pool view.

Performing Panel Unit Root Tests in EViews

The following discussion assumes that you are familiar with the basics of both unit root tests and panel unit root tests.

To begin, select **View/Unit Root Test...** from the menu of an EViews group or pool object, or from the menu of an individual series in a panel structured workfile. Here we show the dialog for a Group unit root test—the other dialogs differ slightly (for the pool object, there is an additional field in the upper-left hand portion of the dialog where you must indicate the name of the pool series on which you wish to conduct your test; for the series object in a panel workfile, the “balanced sample” option is not present).

If you wish to accept the default settings, simply click on **OK**.

EViews will use the default **Summary** setting, and will compute a full suite of unit root tests on the levels of the series, along with a summary of the results.

To customize the unit root calculations, you will choose from a variety of options. The options on the left-hand side of the dialog determine the basic structure of the test or tests, while the options on the right-hand side of the dialog control advanced computational details such as bandwidth or lag selection methods, or kernel methods.

The combo box at the top of the dialog is where you will choose the type of test to perform. There are six settings: “**Summary**”, “**Common root - Levin, Lin, Chu**”, “**Common root - Breitung**”, “**Individual root - Im, Pesaran, Shin**”, “**Individual root - Fisher - ADF**”, “**Individual root - Fisher - PP**”, and “**Common root - Hadri**”, corresponding to one or more of the tests listed above. The combo box labels include a brief description of the assumptions under which the tests are computed. “Common root” indicates that the tests are estimated assuming a common AR structure for all of the series; “Individual root” is used for tests which allow for different AR coefficients in each series.

We have already pointed out that the **Summary** default instructs EViews to estimate the first five of the tests, where applicable, and to provide a brief summary of the results. Selecting

an individual test type allows you better control over the computational method and provides additional detail on the test results.

The next two sets of radio buttons allow you to control the specification of your test equation. First, you may choose to conduct the unit root on the **Level**, **1st difference**, or **2nd difference** of your series. Next, you may choose between sets of exogenous regressors to be included. You can select **Individual intercept** if you wish to include individual fixed effects, **Individual intercepts and individual trends** to include both fixed effects and trends, or **None** for no regressors.

The **Use balanced sample** option is present only if you are estimating a Pool or a Group unit root test. If you select this option, EViews will adjust your sample so that only observations where all series values are not missing will be included in the test equations.

Depending on the form of the test or tests to be computed, you will be presented with various advanced options on the right side of the dialog. For tests that involve regressions on lagged difference terms (Levin, Lin, and Chu, Breitung, Im, Pesaran, and Shin, Fisher - ADF) these options relate to the choice of the number of lags to be included. For the tests involving kernel weighting (Levin, Lin, and Chu, Fisher - PP, Hadri), the options relate to the choice of bandwidth and kernel type.

For a group or pool unit root test, the EViews default is to use automatic selection methods: information matrix criterion based for the number of lag difference terms (with automatic selection of the maximum lag to evaluate), and the Andrews or Newey-West method for bandwidth selection. For unit root tests on a series in a panel workfile, the default behavior uses user-specified options.

If you wish to override these settings, simply enter the appropriate information. You may, for example, select a fixed, user-specified number of lags by entering a number in the **User specified** field. Alternatively, you may customize the settings for automatic lag selection method. Alternative criteria for evaluating the optimal lag length may be selected via the combo box (Akaike, Schwarz, Hannan-Quinn, Modified Akaike, Modified Schwarz, Modified Hannan-Quinn), and you may limit the number of lags to try in automatic selection by entering a number in the **Maximum lags** box. For the kernel based methods, you may select a kernel type from the combo box (**Bartlett**, **Parzen**, **Quadratic spectral**), and you may specify either an automatic bandwidth selection method (**Andrews**, **Newey-West**) or user-specified fixed bandwidth.

As an illustration, we compute the summary panel unit root test, using individual fixed effects as regressors, and automatic lag difference term and bandwidth selection (using the Schwarz criterion for the lag differences, and the Newey-West method and the Bartlett kernel for the bandwidth). The results for the panel unit root test are presented below:

Panel unit root test: Summary
 Date: 08/08/06 Time: 16:02
 Sample: 1935 1954
 Exogenous variables: Individual effects
 Automatic selection of maximum lags
 Automatic selection of lags based on SIC: 0 to 3
 Newey-West bandwidth selection using Bartlett kernel

| Method | Statistic | Prob.** | Cross- sections | Obs |
|--|-----------|---------|--------------------|-----|
| Null: Unit root (assumes common unit root process) | | | | |
| Levin, Lin & Chu t* | 2.39544 | 0.9917 | 10 | 184 |
| Null: Unit root (assumes individual unit root process) | | | | |
| Im, Pesaran and Shin W-stat | 2.80541 | 0.9975 | 10 | 184 |
| ADF - Fisher Chi-square | 12.0000 | 0.9161 | 10 | 184 |
| PP - Fisher Chi-square | 12.9243 | 0.8806 | 10 | 190 |

** Probabilities for Fisher tests are computed using an asymptotic Chi-square distribution. All other tests assume asymptotic normality.

The top of the output indicates the type of test, exogenous variables and test equation options. If we were instead estimating a Pool or Group test, a list of the series used in the test would also be depicted. The lower part of the summary output gives the main test results, organized both by null hypothesis as well as the maintained hypothesis concerning the type of unit root process.

All of the results indicate the presence of a unit root, as the LLC, IPS, and both Fisher tests fail to reject the null of a unit root.

If you only wish to compute a single unit root test type, or if you wish to examine the tests results in greater detail, you may simply repeat the unit root test after selecting the desired test in **Test type** combo box. Here, we show the bottom portion of the LLC test specific output for the same data:

Intermediate results on I

| Cross section | 2nd Stage Coefficient | Variance of Reg | HAC of Dep. | Lag | Max Lag | Bandwidth | Obs |
|---------------|-----------------------|-----------------|-------------|--------|---------|-----------|-----|
| 1 | 0.22672 | 113.14 | 18734. | 0 | 4 | 1.0 | 19 |
| 2 | -0.55912 | 7838.8 | 1851.4 | 1 | 4 | 11.0 | 18 |
| 3 | -0.10233 | 408.12 | 179.68 | 3 | 4 | 5.0 | 16 |
| 4 | -0.05375 | 444.60 | 236.40 | 0 | 4 | 7.0 | 19 |
| 5 | -0.35898 | 147.58 | 11.767 | 1 | 4 | 18.0 | 18 |
| 6 | 0.12362 | 62.429 | 82.716 | 0 | 4 | 1.0 | 19 |
| 7 | -0.13862 | 129.04 | 22.173 | 0 | 4 | 17.0 | 19 |
| 8 | -0.44416 | 113.56 | 43.504 | 1 | 4 | 6.0 | 18 |
| 9 | -0.26332 | 90.040 | 89.960 | 0 | 4 | 2.0 | 19 |
| 10 | -0.11741 | 0.8153 | 0.5243 | 0 | 4 | 5.0 | 19 |
| | Coefficient | t-Stat | SE Reg | mu* | sig* | | Obs |
| Pooled | -0.01940 | -0.464 | 1.079 | -0.554 | 0.919 | | 184 |

For each cross-section, the autoregression coefficient, variance of the regression, HAC of the dependent variable, the selected lag order, maximum lag, bandwidth truncation parameter, and the number of observations used are displayed.

Panel Unit Root Details

Panel unit root tests are similar, but not identical, to unit root tests carried out on a single series. Here, we briefly describe the five panel unit root tests currently supported in EViews; for additional detail, we encourage you to consult the original literature. The discussion assumes that you have a basic knowledge of unit root theory.

We begin by classifying our unit root tests on the basis of whether there are restrictions on the autoregressive process across cross-sections or series. Consider a following AR(1) process for panel data:

$$y_{it} = \rho_i y_{it-1} + X_{it} \delta_i + \epsilon_{it} \quad (26.65)$$

where $i = 1, 2, \dots, N$ cross-section units or series, that are observed over periods $t = 1, 2, \dots, T_i$.

The X_{it} represent the exogenous variables in the model, including any fixed effects or individual trends, ρ_i are the autoregressive coefficients, and the errors ϵ_{it} are assumed to be mutually independent idiosyncratic disturbance. If $|\rho_i| < 1$, y_i is said to be weakly (trend-) stationary. On the other hand, if $|\rho_i| = 1$ then y_i contains a unit root.

For purposes of testing, there are two natural assumptions that we can make about the ρ_i . First, one can assume that the persistence parameters are common across cross-sections so that $\rho_i = \rho$ for all i . The Levin, Lin, and Chu (LLC), Breitung, and Hadri tests all employ

this assumption. Alternatively, one can allow ρ_i to vary freely across cross-sections. The Im, Pesaran, and Shin (IPS), and Fisher-ADF and Fisher-PP tests are of this form.

Tests with Common Unit Root Process

Levin, Lin, and Chu (LLC), Breitung, and Hadri tests all assume that there is a common unit root process so that ρ_i is identical across cross-sections. The first two tests employ a null hypothesis of a unit root while the Hadri test uses a null of no unit root.

LLC and Breitung both consider the following basic ADF specification:

$$\Delta y_{it} = \alpha y_{it-1} + \sum_{j=1}^{p_i} \beta_{ij} \Delta y_{it-j} + X'_{it} \delta + \epsilon_{it} \quad (26.66)$$

where we assume a common $\alpha = \rho - 1$, but allow the lag order for the difference terms, p_i , to vary across cross-sections. The null and alternative hypotheses for the tests may be written as:

$$H_0: \alpha = 0 \quad (26.67)$$

$$H_1: \alpha < 0 \quad (26.68)$$

Under the null hypothesis, there is a unit root, while under the alternative, there is no unit root.

Levin, Lin, and Chu

The method described in LLC derives estimates of α from proxies for Δy_{it} and y_{it} that are standardized and free of autocorrelations and deterministic components.

For a given set of lag orders, we begin by estimating two additional sets of equations, regressing both Δy_{it} , and y_{it-1} on the lag terms Δy_{it-j} (for $j = 1, \dots, p_i$) and the exogenous variables X_{it} . The estimated coefficients from these two regressions will be denoted $(\hat{\beta}, \hat{\delta})$ and $(\hat{\beta}, \hat{\delta})$, respectively.

We define $\Delta \bar{y}_{it}$ by taking Δy_{it} and removing the autocorrelations and deterministic components using the first set of auxiliary estimates:

$$\Delta \bar{y}_{it} = \Delta y_{it} - \sum_{j=1}^{p_i} \hat{\beta}_{ij} \Delta y_{it-j} - X'_{it} \hat{\delta} \quad (26.69)$$

Likewise, we may define the analogous \bar{y}_{it-1} using the second set of coefficients:

$$\bar{y}_{it-1} = y_{it-1} - \sum_{j=1}^{p_i} \hat{\beta}_{ij} \Delta y_{it-j} - X'_{it} \hat{\delta} \quad (26.70)$$

Next, we obtain our proxies by standardizing both $\Delta \bar{y}_{it}$ and \bar{y}_{it-1} , dividing by the regression standard error:

$$\begin{aligned}\Delta \tilde{y}_{it} &= (\Delta \bar{y}_{it} / s_i) \\ \tilde{y}_{it-1} &= (\bar{y}_{it-1} / s_i)\end{aligned}\tag{26.71}$$

where s_i are the estimated standard errors from estimating each ADF in Equation (26.66).

Lastly, an estimate of the coefficient α may be obtained from the pooled proxy equation:

$$\Delta \tilde{y}_{it} = \alpha \tilde{y}_{it-1} + \eta_{it}\tag{26.72}$$

LLC show that under the null, a modified t -statistic for the resulting $\hat{\alpha}$ is asymptotically normally distributed

$$t_{\alpha}^* = \frac{t_{\alpha} - (N\tilde{T})S_N\hat{\sigma}^{-2}se(\hat{\alpha})\mu_{m\tilde{T}}^*}{\sigma_{m\tilde{T}}^*} \rightarrow N(0, 1)\tag{26.73}$$

where t_{α} is the standard t -statistic for $\hat{\alpha} = 0$, $\hat{\sigma}^2$ is the estimated variance of the error term η , $se(\hat{\alpha})$ is the standard error of $\hat{\alpha}$, and:

$$\tilde{T} = T - \left(\sum_i p_i / N \right) - 1\tag{26.74}$$

The remaining terms, which involve complicated moment calculations, are described in greater detail in LLC. The average standard deviation ratio, S_N , is defined as the mean of the ratios of the long-run standard deviation to the innovation standard deviation for each individual. Its estimate is derived using kernel-based techniques. The remaining two terms, $\mu_{m\tilde{T}}^*$ and $\sigma_{m\tilde{T}}^*$ are adjustment terms for the mean and standard deviation.

The LLC method requires a specification of the number of lags used in each cross-section ADF regression, p_i , as well as kernel choices used in the computation of S_N . In addition, you must specify the exogenous variables used in the test equations. You may elect to include no exogenous regressors, or to include individual constant terms (fixed effects), or to employ individual constants and trends.

Breitung

The Breitung method differs from LLC in two distinct ways. First, only the autoregressive portion (and not the exogenous components) is removed when constructing the standardized proxies:

$$\begin{aligned}\Delta \tilde{y}_{it} &= \left(\Delta y_{it} - \sum_{j=1}^{p_i} \hat{\beta}_{ij} \Delta y_{it-j} \right) / s_i \\ \tilde{y}_{it-1} &= \left(y_{it-1} - \sum_{j=1}^{p_i} \hat{\beta}_{ij} y_{it-j} \right) / s_i\end{aligned}\tag{26.75}$$

where $\hat{\beta}$, $\hat{\beta}$, and s_i are as defined for LLC.

Second, the proxies are transformed and detrended,

$$\begin{aligned}\Delta y_{it}^* &= \sqrt{\frac{(T-t)}{(T-t+1)}} \left(\Delta \tilde{y}_{it} - \frac{\Delta \tilde{y}_{it+1} + \dots + \Delta \tilde{y}_{iT}}{T-t} \right) \\ y_{it}^* &= \tilde{y}_{it} - \tilde{y}_{i1} - \frac{t-1}{T-1} (\tilde{y}_{iT} - \tilde{y}_{i1})\end{aligned}\quad (26.76)$$

The persistence parameter α is estimated from the pooled proxy equation:

$$\Delta y_{it}^* = \alpha y_{it-1}^* + \nu_{it} \quad (26.77)$$

Breitung shows that under the null, the resulting estimator α^* is asymptotically distributed as a standard normal.

The Breitung method requires only a specification of the number of lags used in each cross-section ADF regression, p_i , and the exogenous regressors. Note that in contrast with LLC, no kernel computations are required.

Hadri

The Hadri panel unit root test is similar to the KPSS unit root test, and has a null hypothesis of no unit root in any of the series in the panel. Like the KPSS test, the Hadri test is based on the residuals from the individual OLS regressions of y_{it} on a constant, or on a constant and a trend. For example, if we include both the constant and a trend, we derive estimates from:

$$y_{it} = \delta_i + \eta_i t + \epsilon_{it} \quad (26.78)$$

Given the residuals $\hat{\epsilon}$ from the individual regressions, we form the LM statistic:

$$LM_1 = \frac{1}{N} \left(\sum_{i=1}^N \left(\sum_t S_i(t)^2 / T^2 \right) / \tilde{f}_0 \right) \quad (26.79)$$

where $S_i(t)$ are the cumulative sums of the residuals,

$$S_i(t) = \sum_{s=1}^t \hat{\epsilon}_{is} \quad (26.80)$$

and \tilde{f}_0 is the average of the individual estimators of the residual spectrum at frequency zero:

$$\tilde{f}_0 = \sum_{i=1}^N f_{i0} / N \quad (26.81)$$

EViews provides several methods for estimating the f_{i0} . See [“Unit Root Tests” on page 88](#) for additional details.

An alternative form of the LM statistic allows for heteroskedasticity across i :

$$LM_2 = \frac{1}{N} \left(\sum_{i=1}^N \left(\sum_t S_i(t)^2 / T^2 \right) / f_{i0} \right) \quad (26.82)$$

Hadri shows that under mild assumptions,

$$Z = \frac{\sqrt{N}(LM - \xi)}{\zeta} \rightarrow N(0, 1) \quad (26.83)$$

where $\xi = 1/6$ and $\zeta = 1/45$, if the model only includes constants (η_i is set to 0 for all i), and $\xi = 1/15$ and $\zeta = 11/6300$, otherwise.

The Hadri panel unit root tests require only the specification of the form of the OLS regressions: whether to include only individual specific constant terms, or whether to include both constant and trend terms. EViews reports two Z -statistic values, one based on LM_1 with the associated homoskedasticity assumption, and the other using LM_2 that is heteroskedasticity consistent.

It is worth noting that simulation evidence suggests that in various settings (for example, small T), Hadri's panel unit root test experiences significant size distortion in the presence of autocorrelation when there is no unit root. In particular, the Hadri test appears to over-reject the null of stationarity, and *may yield results that directly contradict those obtained using alternative test statistics* (see Hlouskova and Wagner (2006) for discussion and details).

Tests with Individual Unit Root Processes

The Im, Pesaran, and Shin, and the Fisher-ADF and PP tests all allow for individual unit root processes so that ρ_i may vary across cross-sections. The tests are all characterized by the combining of individual unit root tests to derive a panel-specific result.

Im, Pesaran, and Shin

Im, Pesaran, and Shin begin by specifying a separate ADF regression for each cross section:

$$\Delta y_{it} = \alpha y_{it-1} + \sum_{j=1}^{p_i} \beta_{ij} \Delta y_{it-j} + X'_{it} \delta + \epsilon_{it} \quad (26.84)$$

The null hypothesis may be written as,

$$H_0: \alpha_i = 0, \text{ for all } i \quad (26.85)$$

while the alternative hypothesis is given by:

$$H_1: \begin{cases} \alpha_i = 0 & \text{for } i = 1, 2, \dots, N_1 \\ \alpha_i < 0 & \text{for } i = N+1, N+2, \dots, N \end{cases} \quad (26.86)$$

(where the i may be reordered as necessary) which may be interpreted as a non-zero fraction of the individual processes is stationary.

After estimating the separate ADF regressions, the average of the t -statistics for α_i from the individual ADF regressions, $\bar{t}_{iT_i}(p_i)$:

$$\bar{t}_{NT} = \left(\sum_{i=1}^N \bar{t}_{iT_i}(p_i) \right) / N \quad (26.87)$$

is then adjusted to arrive at the desired test statistics.

In the case where the lag order is always zero ($p_i = 0$ for all i), simulated critical values for \bar{t}_{NT} are provided in the IPS paper for different numbers of cross sections N , series lengths T , and for test equations containing either intercepts, or intercepts and linear trends. EViews uses these values, or linearly interpolated values, in evaluating the significance of the test statistics.

In the general case where the lag order in Equation (26.84) may be non-zero for some cross-sections, IPS show that a properly standardized \bar{t}_{NT} has an asymptotic standard normal distribution:

$$W_{\bar{t}_{NT}} = \frac{\sqrt{N} \left(\bar{t}_{NT} - N^{-1} \sum_{i=1}^N E(\bar{t}_{iT}(p_i)) \right)}{\sqrt{N^{-1} \sum_{i=1}^N Var(\bar{t}_{iT}(p_i))}} \rightarrow N(0, 1) \quad (26.88)$$

The expressions for the expected mean and variance of the ADF regression t -statistics, $E(\bar{t}_{iT}(p_i))$ and $Var(\bar{t}_{iT}(p_i))$, are provided by IPS for various values of T and p and differing test equation assumptions, and are not provided here.

The IPS test statistic requires specification of the number of lags and the specification of the deterministic component for each cross-section ADF equation. You may choose to include individual constants, or to include individual constant and trend terms.

Fisher-ADF and Fisher-PP

An alternative approach to panel unit root tests uses Fisher's (1932) results to derive tests that combine the p -values from individual unit root tests. This idea has been proposed by Maddala and Wu, and by Choi.

If we define π_i as the p -value from any individual unit root test for cross-section i , then under the null of unit root for all N cross-sections, we have the asymptotic result that

$$-2 \sum_{i=1}^N \log(\pi_i) \rightarrow \chi_{2N}^2 \quad (26.89)$$

In addition, Choi demonstrates that:

$$Z = \frac{1}{\sqrt{N}} \sum_{i=1}^N \Phi^{-1}(\pi_i) \rightarrow N(0, 1) \quad (26.90)$$

where Φ^{-1} is the inverse of the standard normal cumulative distribution function.

EViews reports both the asymptotic χ^2 and standard normal statistics using ADF and Phillips-Perron individual unit root tests. The null and alternative hypotheses are the same as for the as IPS.

For both Fisher tests, you must specify the exogenous variables for the test equations. You may elect to include no exogenous regressors, to include individual constants (effects), or include individual constant and trend terms.

Additionally, when the Fisher tests are based on ADF test statistics, you must specify the number of lags used in each cross-section ADF regression. For the PP form of the test, you must instead specify a method for estimating f_0 . EViews supports estimators for f_0 based on kernel-based sum-of-covariances. See [“Frequency Zero Spectrum Estimation,” beginning on page 97](#) for details.

Summary of Available Panel Unit Root Tests

The following table summarizes the basic characteristics of the panel unit root tests available in EViews:

| Test | Null | Alternative | Possible Deterministic Component | Autocorrelation Correction Method |
|--------------------|--------------|--------------------------------|----------------------------------|-----------------------------------|
| Levin, Lin and Chu | Unit root | No Unit Root | None, F, T | Lags |
| Breitung | Unit root | No Unit Root | None, F, T | Lags |
| IPS | Unit Root | Some cross-sections without UR | F, T | Lags |
| Fisher-ADF | Unit Root | Some cross-sections without UR | None, F, T | Lags |
| Fisher-PP | Unit Root | Some cross-sections without UR | None, F, T | Kernel |
| Hadri | No Unit Root | Unit Root | F, T | Kernel |

None - no exogenous variables; F - fixed effect; and T - individual effect and individual trend.

References

- Andrews, Donald W. K. (1991). "Heteroskedasticity and Autocorrelation Consistent Covariance Matrix Estimation," *Econometrica*, 59, 817–858.
- Bhargava, A. (1986). "On the Theory of Testing for Unit Roots in Observed Time Series," *Review of Economic Studies*, 53, 369–384.
- Box, George E. P. and Gwilym M. Jenkins (1976). *Time Series Analysis: Forecasting and Control*, Revised Edition, Oakland, CA: Holden-Day.
- Breitung, Jörg (2000). "The Local Power of Some Unit Root Tests for Panel Data," in B. Baltagi (ed.), *Advances in Econometrics, Vol. 15: Nonstationary Panels, Panel Cointegration, and Dynamic Panels*, Amsterdam: JAI Press, p. 161–178.
- Choi, I. (2001). "Unit Root Tests for Panel Data," *Journal of International Money and Finance*, 20: 249–272.
- Davidson, Russell and James G. MacKinnon (1993). *Estimation and Inference in Econometrics*, Oxford: Oxford University Press.
- Dezhbaksh, Hashem (1990). "The Inappropriate Use of Serial Correlation Tests in Dynamic Linear Models," *Review of Economics and Statistics*, 72, 126–132.
- Dickey, D.A. and W.A. Fuller (1979). "Distribution of the Estimators for Autoregressive Time Series with a Unit Root," *Journal of the American Statistical Association*, 74, 427–431.
- Elliott, Graham, Thomas J. Rothenberg and James H. Stock (1996). "Efficient Tests for an Autoregressive Unit Root," *Econometrica* 64, 813–836.
- Engle, Robert F. and C. W. J. Granger (1987). "Co-integration and Error Correction: Representation, Estimation, and Testing," *Econometrica*, 55, 251–276.
- Fair, Ray C. (1984). *Specification, Estimation, and Analysis of Macroeconometric Models*, Cambridge, MA: Harvard University Press.
- Fisher, R. A. (1932). *Statistical Methods for Research Workers*, 4th Edition, Edinburgh: Oliver & Boyd.
- Greene, William H. (1997). *Econometric Analysis*, 3rd Edition, Upper Saddle River, NJ: Prentice Hall.
- Hadri, Kaddour (2000). "Testing for Stationarity in Heterogeneous Panel Data," *Econometric Journal*, 3, 148–161.
- Hamilton, James D. (1994a). *Time Series Analysis*, Princeton University Press.
- Hayashi, Fumio. (2000). *Econometrics*, Princeton, NJ: Princeton University Press.
- Hlouskova, Jaroslava and M. Wagner (2006). "The Performance of Panel Unit Root and Stationarity Tests: Results from a Large Scale Simulation Study," *Econometric Reviews*, 25, 85–116.
- Im, K. S., M. H. Pesaran, and Y. Shin (2003). "Testing for Unit Roots in Heterogeneous Panels," *Journal of Econometrics*, 115, 53–74.
- Johnston, Jack and John Enrico DiNardo (1997). *Econometric Methods*, 4th Edition, New York: McGraw-Hill.
- Kwiatkowski, Denis, Peter C. B. Phillips, Peter Schmidt & Yongcheol Shin (1992). "Testing the Null Hypothesis of Stationary against the Alternative of a Unit Root," *Journal of Econometrics*, 54, 159–178.
- Levin, A., C. F. Lin, and C. Chu (2002). "Unit Root Tests in Panel Data: Asymptotic and Finite-Sample Properties," *Journal of Econometrics*, 108, 1–24.

- MacKinnon, James G. (1991). "Critical Values for Cointegration Tests," Chapter 13 in R. F. Engle and C. W. J. Granger (eds.), *Long-run Economic Relationships: Readings in Cointegration*, Oxford: Oxford University Press.
- MacKinnon, James G. (1996). "Numerical Distribution Functions for Unit Root and Cointegration Tests," *Journal of Applied Econometrics*, 11, 601-618.
- Maddala, G. S. and S. Wu (1999). "A Comparative Study of Unit Root Tests with Panel Data and A New Simple Test," *Oxford Bulletin of Economics and Statistics*, 61, 631-52.
- Newey, Whitney and Kenneth West (1994). "Automatic Lag Selection in Covariance Matrix Estimation," *Review of Economic Studies*, 61, 631-653.
- Ng, Serena and Pierre Perron (2001). "Lag Length Selection and the Construction of Unit Root Tests with Good Size and Power," *Econometrica*, 69(6), 1519-1554.
- Phillips, P.C.B. and P. Perron (1988). "Testing for a Unit Root in Time Series Regression," *Biometrika*, 75, 335-346.
- Rao, P. and Z. Griliches (1969). "Small Sample Properties of Several Two-Stage Regression Methods in the Context of Auto-Correlated Errors," *Journal of the American Statistical Association*, 64, 253-272.
- Said, Said E. and David A. Dickey (1984). "Testing for Unit Roots in Autoregressive Moving Average Models of Unknown Order," *Biometrika*, 71, 599-607.

Chapter 27. Forecasting from an Equation

This chapter describes procedures for forecasting and computing fitted values from a single equation. The techniques described here are for forecasting with equation objects estimated using regression methods. Forecasts from equations estimated by specialized techniques, such as ARCH, binary, ordered, tobit, and count methods, are discussed in the corresponding chapters. Forecasting from a series using exponential smoothing methods is explained in [“Exponential Smoothing” on page 354](#) of the *User’s Guide I*, and forecasting using multiple equations and models is described in [Chapter 36. “Models,” on page 407](#).

Forecasting from Equations in EViews

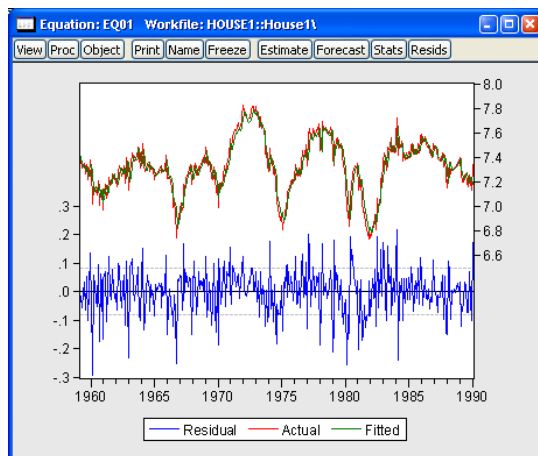
To illustrate the process of forecasting from an estimated equation, we begin with a simple example. Suppose we have data on the logarithm of monthly housing starts (HS) and the logarithm of the S&P index (SP) over the period 1959M01–1996M01. The data are contained in a workfile with range 1959M01–1998M12.

We estimate a regression of HS on a constant, SP, and the lag of HS, with an AR(1) to correct for residual serial correlation, using data for the period 1959M01–1990M01, and then use the model to forecast housing starts under a variety of settings. Following estimation, the equation results are held in the equation object EQ01:

| Dependent Variable: HS | | | | |
|--|-------------|-----------------------|-------------|--------|
| Method: Least Squares | | | | |
| Date: 08/08/06 Time: 16:23 | | | | |
| Sample (adjusted): 1959M03 1990M01 | | | | |
| Included observations: 371 after adjustments | | | | |
| Convergence achieved after 4 iterations | | | | |
| | Coefficient | Std. Error | t-Statistic | Prob. |
| C | 0.321924 | 0.117278 | 2.744975 | 0.0063 |
| HS(-1) | 0.952653 | 0.016218 | 58.74157 | 0.0000 |
| SP | 0.005222 | 0.007588 | 0.688249 | 0.4917 |
| AR(1) | -0.271254 | 0.052114 | -5.205027 | 0.0000 |
| R-squared | 0.861373 | Mean dependent var | 7.324051 | |
| Adjusted R-squared | 0.860240 | S.D. dependent var | 0.220996 | |
| S.E. of regression | 0.082618 | Akaike info criterion | -2.138453 | |
| Sum squared resid | 2.505050 | Schwarz criterion | -2.096230 | |
| Log likelihood | 400.6830 | Hannan-Quinn criter. | -2.121683 | |
| F-statistic | 760.1338 | Durbin-Watson stat | 2.013460 | |
| Prob(F-statistic) | 0.000000 | | | |
| Inverted AR Roots | - .27 | | | |

Note that the estimation sample is adjusted by two observations to account for the first difference of the lagged endogenous variable used in deriving AR(1) estimates for this model.

To get a feel for the fit of the model, select **View/Actual, Fitted, Residual...**, then choose **Actual, Fitted, Residual Graph**:



The actual and fitted values depicted on the upper portion of the graph are virtually indistinguishable. This view provides little control over the process of producing fitted values, and does not allow you to save your fitted values. These limitations are overcome by using EViews built-in forecasting procedures to compute fitted values for the dependent variable.

How to Perform a Forecast

To forecast HS from this equation, push the **Forecast** button on the equation toolbar, or select **Proc/Forecast...**.

At the top of the **Forecast** dialog, EViews displays information about the forecast. Here, we show a basic version of the dialog showing that we are forecasting values for the dependent series HS using the estimated EQ01. More complex settings are described in [“Forecasting from Equations with Expressions” on page 132](#).

You should provide the following information:

- **Forecast name.** Fill in the edit box with the series name to be given to your forecast. EViews suggests a name, but you can change it to any valid series name. The name should be different from the name of the dependent variable, since the forecast procedure will overwrite data in the specified series.
- **S.E. (optional).** If desired, you may provide a name for the series to be filled with the forecast standard errors. If you do not provide a name, no forecast errors will be saved.
- **GARCH (optional).** For models estimated by ARCH, you will be given a further option of saving forecasts of the conditional variances (GARCH terms). See [Chapter 29](#), “ARCH and GARCH Estimation,” on page 185 for a discussion of GARCH estimation.
- **Forecasting method.** You have a choice between **Dynamic** and **Static** forecast methods. **Dynamic** calculates *dynamic, multi-step forecasts* starting from the first period in the forecast sample. In dynamic forecasting, previously forecasted values for the lagged dependent variables are used in forming forecasts of the current value (see “[Forecasts with Lagged Dependent Variables](#)” on page 125 and “[Forecasting with ARMA Errors](#)” on page 127). This choice will only be available when the estimated equation contains dynamic components, *e.g.*, lagged dependent variables or ARMA terms. **Static** calculates a sequence of *one-step ahead forecasts*, using the actual, rather than forecasted values for lagged dependent variables, if available.

You may elect to always ignore coefficient uncertainty in computing forecast standard errors (when relevant) by unselecting the **Coef uncertainty in S.E. calc** box.

In addition, in specifications that contain ARMA terms, you can set the **Structural** option, instructing EViews to ignore any ARMA terms in the equation when forecasting. By default, when your equation has ARMA terms, both dynamic and static solution methods form forecasts of the residuals. If you select **Structural**, all forecasts will ignore the forecasted residuals and will form predictions using only the structural part of the ARMA specification.

- **Sample range.** You must specify the sample to be used for the forecast. By default, EViews sets this sample to be the workfile sample. By specifying a sample outside the sample used in estimating your equation (the *estimation sample*), you can instruct EViews to produce out-of-sample forecasts.

Note that you are responsible for supplying the values for the independent variables in the out-of-sample forecasting period. For static forecasts, you must also supply the values for any lagged dependent variables.

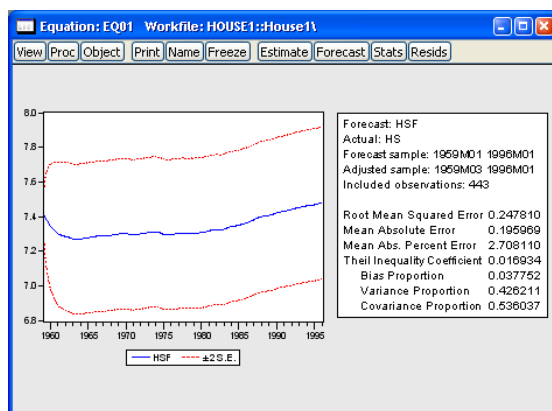
- **Output.** You can choose to see the forecast output as a graph or a numerical forecast evaluation, or both. Forecast evaluation is only available if the forecast sample includes observations for which the dependent variable is observed.

- **Insert actuals for out-of-sample observations.** By default, EViews will fill the forecast series with the values of the actual dependent variable for observations not in the forecast sample. This feature is convenient if you wish to show the divergence of the forecast from the actual values; for observations prior to the beginning of the forecast sample, the two series will contain the same values, then they will diverge as the forecast differs from the actuals. In some contexts, however, you may wish to have forecasted values only for the observations in the forecast sample. If you uncheck this option, EViews will fill the out-of-sample observations with missing values.

Note that when performing forecasts from equations specified using expressions or auto-updating series, you may encounter a version of the **Forecast** dialog that differs from the basic dialog depicted above. See [“Forecasting from Equations with Expressions” on page 132](#) for details.

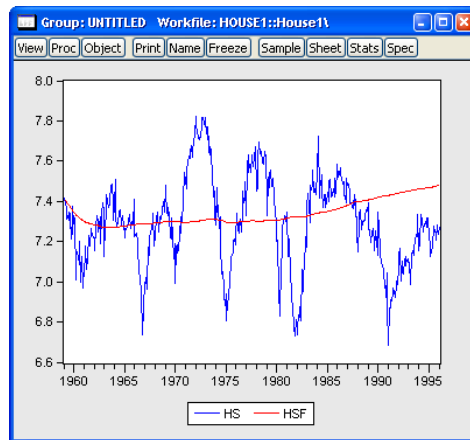
An Illustration

Suppose we produce a dynamic forecast using EQ01 over the sample 1959M01 to 1996M01. The forecast values will be placed in the series HSF, and EViews will display a graph of the forecasts and the plus and minus two standard error bands, as well as a forecast evaluation:



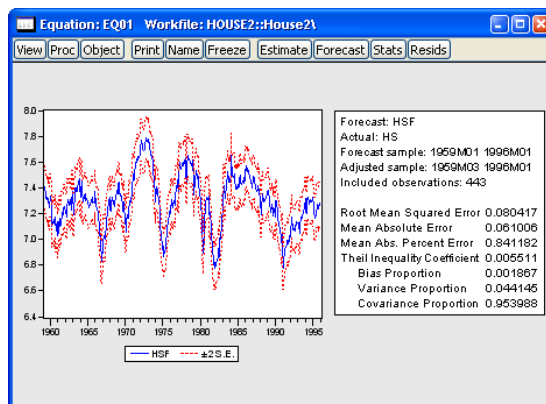
This is a dynamic forecast for the period from 1959M01 through 1996M01. For every period, the previously forecasted values for $HS(-1)$ are used in forming a forecast of the subsequent value of HS . As noted in the output, the forecast values are saved in the series HSF. Since HSF is a standard EViews series, you may examine your forecasts using all of the standard tools for working with series objects.

For example, we may examine the actual versus fitted values by creating a group containing HS and HSF , and plotting the two series. Click on **Quick/Show...** and enter “ HS ” and “ HSF ”. Then select **View/Graph...** and select **Line & Symbol** on the left hand side of the dialog to display the two series:

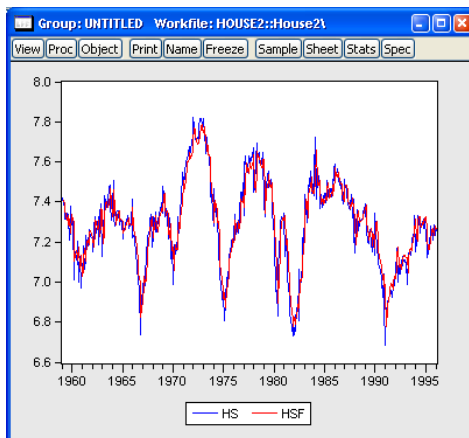


Note the considerable difference between this actual and fitted graph and the **Actual, Fitted, Residual Graph** depicted above.

To perform a series of one-step ahead forecasts, click on **Forecast** on the equation toolbar, and select **Static** forecast. EViews will display the forecast results:

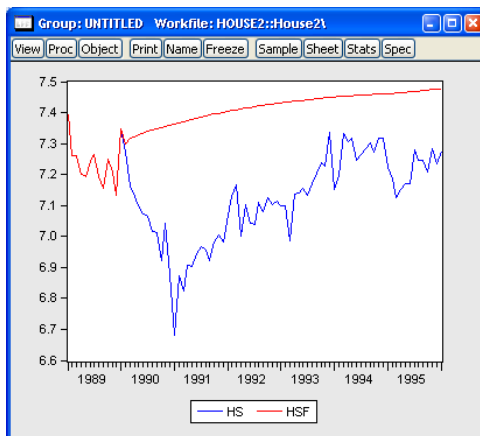


We may also compare the actual and fitted values from the static forecast by examining a line graph of a group containing HS and the new HSF.



The one-step ahead static forecasts are more accurate than the dynamic forecasts since, for each period, the actual value of $HS(-1)$ is used in forming the forecast of HS . These one-step ahead static forecasts are the same forecasts used in the **Actual, Fitted, Residual Graph** displayed above.

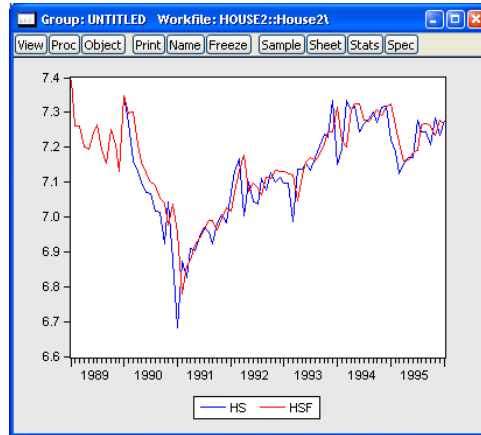
Lastly, we construct a dynamic forecast beginning in 1990M02 (the first period following the estimation sample) and ending in 1996M01. Keep in mind that data are available for SP for this entire period. The plot of the actual and the forecast values for 1989M01 to 1996M01 is given by:



Since we use the default settings for out-of-forecast sample values, EViews backfills the forecast series prior to the forecast sample (up through 1990M01), then dynamically forecasts HS for each subsequent period through 1996M01. This is the forecast that you would have

constructed if, in 1990M01, you predicted values of HS from 1990M02 through 1996M01, given knowledge about the entire path of SP over that period.

The corresponding static forecast is displayed below:



Again, EViews backfills the values of the forecast series, HSF1, through 1990M01. This forecast is the one you would have constructed if, in 1990M01, you used all available data to estimate a model, and then used this estimated model to perform one-step ahead forecasts every month for the next six years.

The remainder of this chapter focuses on the details associated with the construction of these forecasts, the corresponding forecast evaluations, and forecasting in more complex settings involving equations with expressions or auto-updating series.

Forecast Basics

EViews stores the forecast results in the series specified in the **Forecast name** field. We will refer to this series as the *forecast series*.

The *forecast sample* specifies the observations for which EViews will try to compute fitted or forecasted values. If the forecast is not computable, a missing value will be returned. In some cases, EViews will carry out automatic adjustment of the sample to prevent a forecast consisting entirely of missing values (see [“Adjustment for Missing Values”](#) on page 120, below). Note that the forecast sample may or may not overlap with the sample of observations used to estimate the equation.

For values not included in the forecast sample, there are two options. By default, EViews fills in the actual values of the dependent variable. If you turn off the **Insert actuals for out-of-sample** option, out-of-forecast-sample values will be filled with NAs.

As a consequence of these rules, *all data in the forecast series will be overwritten during the forecast procedure*. Existing values in the forecast series will be lost.

Computing Point Forecasts

For each observation in the forecast sample, EViews computes the fitted value of the dependent variable using the estimated parameters, the right-hand side exogenous variables, and either the actual or estimated values for lagged endogenous variables and residuals. The method of constructing these forecasted values depends upon the estimated model and user-specified settings.

To illustrate the forecasting procedure, we begin with a simple linear regression model with no lagged endogenous right-hand side variables, and no ARMA terms. Suppose that you have estimated the following equation specification:

$$y \text{ c } x \text{ z}$$

Now click on **Forecast**, specify a forecast period, and click **OK**.

For every observation in the forecast period, EViews will compute the fitted value of Y using the estimated parameters and the corresponding values of the regressors, X and Z :

$$\hat{y}_t = \hat{c}(1) + \hat{c}(2)x_t + \hat{c}(3)z_t. \quad (27.1)$$

You should make certain that you have valid values for the exogenous right-hand side variables for all observations in the forecast period. If any data are missing in the forecast sample, the corresponding forecast observation will be an NA.

Adjustment for Missing Values

There are two cases when a missing value will be returned for the forecast value. First, if any of the regressors have a missing value, and second, if any of the regressors are out of the range of the workfile. This includes the implicit error terms in AR models.

In the case of forecasts with no dynamic components in the specification (*i.e.* with no lagged endogenous or ARMA error terms), a missing value in the forecast series will not affect subsequent forecasted values. In the case where there are dynamic components, however, a single missing value in the forecasted series will propagate throughout all future values of the series.

As a convenience feature, EViews will move the starting point of the sample forward where necessary until a valid forecast value is obtained. Without these adjustments, the user would have to figure out the appropriate number of presample values to skip, otherwise the forecast would consist entirely of missing values. For example, suppose you wanted to forecast dynamically from the following equation specification:

$$y \text{ c } y(-1) \text{ ar}(1)$$

If you specified the beginning of the forecast sample to the beginning of the workfile range, EViews will adjust forward the forecast sample by 2 observations, and will use the pre-forecast-sample values of the lagged variables (the loss of 2 observations occurs because the residual loses one observation due to the lagged endogenous variable so that the forecast for the error term can begin only from the third observation.)

Forecast Errors and Variances

Suppose the “true” model is given by:

$$y_t = x_t' \beta + \epsilon_t, \quad (27.2)$$

where ϵ_t is an independent, and identically distributed, mean zero random disturbance, and β is a vector of unknown parameters. Below, we relax the restriction that the ϵ 's be independent.

The true model generating y is not known, but we obtain estimates b of the unknown parameters β . Then, setting the error term equal to its mean value of zero, the (point) forecasts of y are obtained as:

$$\hat{y}_t = x_t' b. \quad (27.3)$$

Forecasts are made with error, where the error is simply the difference between the actual and forecasted value $e_t = y_t - x_t' b$. Assuming that the model is correctly specified, there are two sources of forecast error: residual uncertainty and coefficient uncertainty.

Residual Uncertainty

The first source of error, termed *residual* or *innovation uncertainty*, arises because the innovations ϵ in the equation are unknown for the forecast period and are replaced with their expectations. While the residuals are zero in expected value, the individual values are non-zero; the larger the variation in the individual residuals, the greater the overall error in the forecasts.

The standard measure of this variation is the standard error of the regression (labeled “S.E. of regression” in the equation output). Residual uncertainty is usually the largest source of forecast error.

In dynamic forecasts, innovation uncertainty is compounded by the fact that lagged dependent variables and ARMA terms depend on lagged innovations. EViews also sets these equal to their expected values, which differ randomly from realized values. This additional source of forecast uncertainty tends to rise over the forecast horizon, leading to a pattern of increasing forecast errors. Forecasting with lagged dependent variables and ARMA terms is discussed in more detail below.

Coefficient Uncertainty

The second source of forecast error is *coefficient uncertainty*. The estimated coefficients b of the equation deviate from the true coefficients β in a random fashion. The standard error of the estimated coefficient, given in the regression output, is a measure of the precision with which the estimated coefficients measure the true coefficients.

The effect of coefficient uncertainty depends upon the exogenous variables. Since the estimated coefficients are multiplied by the exogenous variables x in the computation of forecasts, the more the exogenous variables deviate from their mean values, the greater is the forecast uncertainty.

Forecast Variability

The variability of forecasts is measured by the forecast standard errors. For a single equation without lagged dependent variables or ARMA terms, the forecast standard errors are computed as:

$$\text{forecast se} = s\sqrt{1 + x_t'(X'X)^{-1}x_t} \quad (27.4)$$

where s is the standard error of regression. These standard errors account for both innovation (the first term) and coefficient uncertainty (the second term). Point forecasts made from linear regression models estimated by least squares are optimal in the sense that they have the smallest forecast variance among forecasts made by linear unbiased estimators. Moreover, if the innovations are normally distributed, the forecast errors have a t -distribution and forecast intervals can be readily formed.

If you supply a name for the forecast standard errors, EViews computes and saves a series of forecast standard errors in your workfile. You can use these standard errors to form forecast intervals. If you choose the **Do graph** option for output, EViews will plot the forecasts with plus and minus two standard error bands. These two standard error bands provide an approximate 95% forecast interval; if you (hypothetically) make many forecasts, the actual value of the dependent variable will fall inside these bounds 95 percent of the time.

Additional Details

EViews accounts for the additional forecast uncertainty generated when lagged dependent variables are used as explanatory variables (see [“Forecasts with Lagged Dependent Variables” on page 125](#)).

There are cases where coefficient uncertainty is ignored in forming the forecast standard error. For example, coefficient uncertainty is always ignored in equations specified by expression, for example, nonlinear least squares, and equations that include PDL (polynomial distributed lag) terms ([“Forecasting with Nonlinear and PDL Specifications” on page 138](#)).

In addition, forecast standard errors do not account for GLS weights in estimated panel equations.

Forecast Evaluation

Suppose we construct a dynamic forecast for HS over the period 1990M02 to 1996M01 using our estimated housing equation. If the **Forecast evaluation** option is checked, and there are actual data for the forecasted variable for the forecast sample, EViews reports a table of statistical results evaluating the forecast:

| | |
|--------------------------------|----------|
| Forecast: HSF | |
| Actual: HS | |
| Sample: 1990M02 1996M01 | |
| Include observations: 72 | |
| Root Mean Squared Error | 0.318700 |
| Mean Absolute Error | 0.297261 |
| Mean Absolute Percentage Error | 4.205889 |
| Theil Inequality Coefficient | 0.021917 |
| Bias Proportion | 0.869982 |
| Variance Proportion | 0.082804 |
| Covariance Proportion | 0.047214 |

Note that EViews cannot compute a forecast evaluation if there are no data for the dependent variable for the forecast sample.

The forecast evaluation is saved in one of two formats. If you turn on the **Do graph** option, the forecasts are included along with a graph of the forecasts. If you wish to display the evaluations in their own table, you should turn off the **Do graph** option in the Forecast dialog box.

Suppose the forecast sample is $j = T + 1, T + 2, \dots, T + h$, and denote the actual and forecasted value in period t as y_t and \hat{y}_t , respectively. The reported forecast error statistics are computed as follows:

| | |
|-------------------------|---|
| Root Mean Squared Error | $\sqrt{\frac{1}{h} \sum_{t=T+1}^{T+h} (\hat{y}_t - y_t)^2}$ |
| Mean Absolute Error | $\frac{1}{h} \sum_{t=T+1}^{T+h} \hat{y}_t - y_t $ |

| | |
|--------------------------------|---|
| Mean Absolute Percentage Error | $100 \sum_{t=T+1}^{T+h} \left \frac{\hat{y}_t - y_t}{y_t} \right / h$ |
| Theil Inequality Coefficient | $\frac{\sqrt{\sum_{t=T+1}^{T+h} (\hat{y}_t - y_t)^2 / h}}{\sqrt{\sum_{t=T+1}^{T+h} \hat{y}_t^2 / h} + \sqrt{\sum_{t=T+1}^{T+h} y_t^2 / h}}$ |

The first two forecast error statistics depend on the scale of the dependent variable. These should be used as relative measures to compare forecasts for the same series across different models; the smaller the error, the better the forecasting ability of that model according to that criterion. The remaining two statistics are scale invariant. The Theil inequality coefficient always lies between zero and one, where zero indicates a perfect fit.

The mean squared forecast error can be decomposed as:

$$\sum (\hat{y}_t - y_t)^2 / h = ((\sum \hat{y}_t / h) - \bar{y})^2 + (s_{\hat{y}} - s_y)^2 + 2(1 - r)s_{\hat{y}}s_y \quad (27.5)$$

where $\sum \hat{y}_t / h$, \bar{y} , $s_{\hat{y}}$, s_y are the means and (biased) standard deviations of \hat{y}_t and y , and r is the correlation between \hat{y} and y . The proportions are defined as:

| | |
|-----------------------|---|
| Bias Proportion | $\frac{((\sum \hat{y}_t / h) - \bar{y})^2}{\sum (\hat{y}_t - y_t)^2 / h}$ |
| Variance Proportion | $\frac{(s_{\hat{y}} - s_y)^2}{\sum (\hat{y}_t - y_t)^2 / h}$ |
| Covariance Proportion | $\frac{2(1 - r)s_{\hat{y}}s_y}{\sum (\hat{y}_t - y_t)^2 / h}$ |

- The bias proportion tells us how far the mean of the forecast is from the mean of the actual series.
- The variance proportion tells us how far the variation of the forecast is from the variation of the actual series.
- The covariance proportion measures the remaining unsystematic forecasting errors.

Note that the bias, variance, and covariance proportions add up to one.

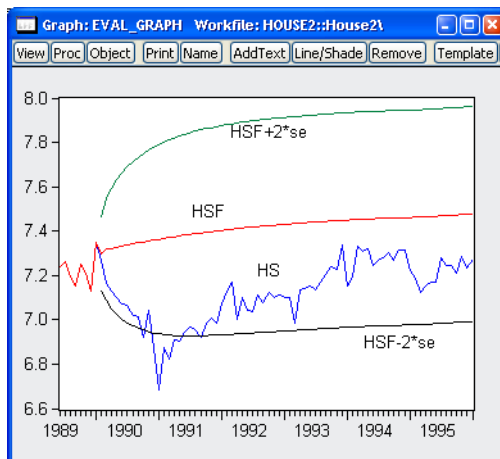
If your forecast is “good”, the bias and variance proportions should be small so that most of the bias should be concentrated on the covariance proportions. For additional discussion of forecast evaluation, see Pindyck and Rubinfeld (1998, p. 210-214).

For the example output, the bias proportion is large, indicating that the mean of the forecasts does a poor job of tracking the mean of the dependent variable. To check this, we will plot the forecasted series together with the actual series in the forecast sample with the two standard error bounds. Suppose we saved the forecasts and their standard errors as HSF and HSFSE, respectively. Then the plus and minus two standard error series can be generated by the commands:

```
smp1 1990m02 1996m01

series hsf_high = hsf + 2*hsfse
series hsf_low = hsf - 2*hsfse
```

Create a group containing the four series. You can highlight the four series HS, HSF, HSF_HIGH, and HSF_LOW, double click on the selected area, and select **Open Group**, or you can select **Quick/Show...** and enter the four series names. Once you have the group open, select **View/Graph...** and select **Line & Symbol** from the left side of the dialog.



The forecasts completely miss the downturn at the start of the 1990's, but, subsequent to the recovery, track the trend reasonably well from 1992 to 1996.

Forecasts with Lagged Dependent Variables

Forecasting is complicated by the presence of lagged dependent variables on the right-hand side of the equation. For example, we can augment the earlier specification to include the first lag of Y :

$$y = c + xz + y(-1)$$

and click on the **Forecast** button and fill out the series names in the dialog as above. There is some question, however, as to how we should evaluate the lagged value of Y that appears on the right-hand side of the equation. There are two possibilities: dynamic forecasting and static forecasting.

Dynamic Forecasting

If you select dynamic forecasting, EViews will perform a multi-step forecast of Y , beginning at the start of the forecast sample. For our single lag specification above:

- The initial observation in the forecast sample will use the actual value of lagged Y . Thus, if S is the first observation in the forecast sample, EViews will compute:

$$\hat{y}_S = \hat{c}(1) + \hat{c}(2)x_S + \hat{c}(3)z_S + \hat{c}(4)y_{S-1}, \quad (27.6)$$

where y_{S-1} is the value of the lagged endogenous variable in the period prior to the start of the forecast sample. This is the one-step ahead forecast.

- Forecasts for subsequent observations will use the previously *forecasted* values of Y :

$$\hat{y}_{S+k} = \hat{c}(1) + \hat{c}(2)x_{S+k} + \hat{c}(3)z_{S+k} + \hat{c}(4)\hat{y}_{S+k-1}. \quad (27.7)$$

- These forecasts may differ significantly from the one-step ahead forecasts.

If there are additional lags of Y in the estimating equation, the above algorithm is modified to account for the non-availability of lagged forecasted values in the additional period. For example, if there are three lags of Y in the equation:

- The first observation (S) uses the actual values for all three lags, y_{S-3} , y_{S-2} , and y_{S-1} .
- The second observation ($S+1$) uses actual values for y_{S-2} and, y_{S-1} and the forecasted value \hat{y}_S of the first lag of y_{S+1} .
- The third observation ($S+2$) will use the actual values for y_{S-1} , and forecasted values \hat{y}_{S+1} and \hat{y}_S for the first and second lags of y_{S+2} .
- All subsequent observations will use the forecasted values for all three lags.

The selection of the start of the forecast sample is very important for dynamic forecasting. The dynamic forecasts are true multi-step forecasts (from the start of the forecast sample), since they use the recursively computed forecast of the lagged value of the dependent variable. These forecasts may be interpreted as the forecasts for subsequent periods that would be computed using information available at the start of the forecast sample.

Dynamic forecasting requires that data for the exogenous variables be available for every observation in the forecast sample, and that values for any lagged dependent variables be observed at the start of the forecast sample (in our example, y_{S-1} , but more generally, any lags of y). If necessary, the forecast sample will be adjusted.

Any missing values for the explanatory variables will generate an NA for that observation and in all subsequent observations, via the dynamic forecasts of the lagged dependent variable.

Static Forecasting

Static forecasting performs a series of one-step ahead forecasts of the dependent variable:

- For each observation in the forecast sample, EViews computes:

$$\hat{y}_{S+k} = \hat{c}(1) + \hat{c}(2)x_{S+k} + \hat{c}(3)z_{S+k} + \hat{c}(4)y_{S+k-1} \quad (27.8)$$

always using the actual value of the lagged endogenous variable.

Static forecasting requires that data for both the exogenous and any lagged endogenous variables be observed for every observation in the forecast sample. As above, EViews will, if necessary, adjust the forecast sample to account for pre-sample lagged variables. If the data are not available for any period, the forecasted value for that observation will be an NA. The presence of a forecasted value of NA does not have any impact on forecasts for subsequent observations.

A Comparison of Dynamic and Static Forecasting

Both methods will always yield identical results in the first period of a multi-period forecast. Thus, two forecast series, one dynamic and the other static, should be identical for the first observation in the forecast sample.

The two methods will differ for subsequent periods only if there are lagged dependent variables or ARMA terms.

Forecasting with ARMA Errors

Forecasting from equations with ARMA components involves some additional complexities. When you use the AR or MA specifications, you will need to be aware of how EViews handles the forecasts of the lagged residuals which are used in forecasting.

Structural Forecasts

By default, EViews will forecast values for the residuals using the estimated ARMA structure, as described below.

For some types of work, you may wish to assume that the ARMA errors are always zero. If you select the structural forecast option by checking **Structural (ignore ARMA)**, EViews computes the forecasts assuming that the errors are always zero. If the equation is estimated without ARMA terms, this option has no effect on the forecasts.

Forecasting with AR Errors

For equations with AR errors, EViews adds forecasts of the residuals from the equation to the forecast of the structural model that is based on the right-hand side variables.

In order to compute an estimate of the residual, EViews requires estimates or actual values of the lagged residuals. For the first observation in the forecast sample, EViews will use pre-sample data to compute the lagged residuals. If the pre-sample data needed to compute the lagged residuals are not available, EViews will adjust the forecast sample, and backfill the forecast series with actual values (see the discussion of “[Adjustment for Missing Values](#)” on [page 120](#)).

If you choose the **Dynamic** option, both the lagged dependent variable and the lagged residuals will be forecasted dynamically. If you select **Static**, both will be set to the actual lagged values. For example, consider the following AR(2) model:

$$\begin{aligned} y_t &= x_t' \beta + u_t \\ u_t &= \rho_1 u_{t-1} + \rho_2 u_{t-2} + \epsilon_t \end{aligned} \quad (27.9)$$

Denote the fitted residuals as $e_t = y_t - x_t' b$, and suppose the model was estimated using data up to $t = S - 1$. Then, provided that the x_t values are available, the static and dynamic forecasts for $t = S, S + 1, \dots$, are given by:

| | Static | Dynamic |
|-----------------|--|--|
| \hat{y}_S | $x_S' b + \hat{\rho}_1 e_{S-1} + \hat{\rho}_2 e_{S-2}$ | $x_S' b + \hat{\rho}_1 e_{S-1} + \hat{\rho}_2 e_{S-2}$ |
| \hat{y}_{S+1} | $x_{S+1}' b + \hat{\rho}_1 e_S + \hat{\rho}_2 e_{S-1}$ | $x_{S+1}' b + \hat{\rho}_1 \hat{u}_S + \hat{\rho}_2 e_{S-1}$ |
| \hat{y}_{S+2} | $x_{S+2}' b + \hat{\rho}_1 e_{S+1} + \hat{\rho}_2 e_S$ | $x_{S+2}' b + \hat{\rho}_1 \hat{u}_{S+1} + \hat{\rho}_2 \hat{u}_S$ |

where the residuals $\hat{u}_t = \hat{y}_t - x_t' b$ are formed using the forecasted values of y_t . For subsequent observations, the dynamic forecast will always use the residuals based upon the multi-step forecasts, while the static forecast will use the one-step ahead forecast residuals.

Forecasting with MA Errors

In general, you need not concern yourselves with the details of MA forecasting, since EViews will do all of the work for you. However, for those of you who are interested in the details of dynamic forecasting, the following discussion should aid you in relating EViews results with those obtained from other sources.

We begin by noting that the key step in computing forecasts using MA terms is to obtain fitted values for the innovations in the pre-forecast sample period. For example, if you are performing dynamic forecasting of the values of y , beginning in period S , with a simple $MA(q)$ process:

$$\hat{y}_S = \hat{\phi}_1 \epsilon_{S-1} + \dots + \hat{\phi}_q \epsilon_{S-q}, \quad (27.10)$$

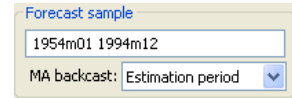
you will need values for the pre-forecast sample innovations, $\epsilon_{S-1}, \epsilon_{S-2}, \dots, \epsilon_{S-q}$. Similarly, constructing a static forecast for a given period will require estimates of the q lagged innovations at every period in the forecast sample.

If your equation is estimated with backcasting turned on, EViews will perform backcasting to obtain these values. If your equation is estimated with backcasting turned off, or if the forecast sample precedes the estimation sample, the initial values will be set to zero.

Backcast Sample

The first step in obtaining pre-forecast innovations is obtaining estimates of the pre-*estimation* sample innovations: $\epsilon_0, \epsilon_{-1}, \epsilon_{-2}, \dots, \epsilon_{-q}$. (For notational convenience, we normalize the start and end of the estimation sample to $t = 1$ and $t = T$, respectively.)

EViews offers two different approaches for obtaining estimates—you may use the **MA backcast** combo box to choose between the default **Estimation period** and the **Forecast available (v5)** methods.



The **Estimation period** method uses data for the estimation sample to compute backcast estimates. Then as in estimation ([“Backcasting MA terms” on page 80](#)), the q values for the innovations *beyond* the estimation sample are set to zero:

$$\tilde{\epsilon}_{T+1} = \tilde{\epsilon}_{T+2} = \dots = \tilde{\epsilon}_{T+q} = 0 \quad (27.11)$$

EViews then uses the unconditional residuals to perform the backward recursion:

$$\tilde{\epsilon}_t = \hat{u}_t - \hat{\theta}_1 \tilde{\epsilon}_{t+1} - \dots - \hat{\theta}_q \tilde{\epsilon}_{t+q} \quad (27.12)$$

for $t = T, \dots, 0, \dots, -(q-1)$ to obtain the pre-estimation sample residuals. Note that absent changes in the data, using **Estimation period** produces pre-forecast sample innovations that match those employed in estimation (where applicable).

The **Forecast available (v5)** method offers different approaches for dynamic and static forecasting:

- For dynamic forecasting, EViews applies the backcasting procedure using data from the beginning of the estimation sample to either the beginning of the forecast period, or the end of the estimation sample, whichever comes first.
- For static forecasting, the backcasting procedure uses data from the beginning of the estimation sample to the end of the forecast period.

For both dynamic and static forecasts, the post-backcast sample innovations are initialized to zero and the backward recursion is employed to obtain estimates of the pre-estimation

sample innovations. Note that **Forecast available (v5)** does not guarantee that the pre-sample forecast innovations match those employed in estimation.

Pre-Forecast Innovations

Given the backcast estimates of the *pre-estimation* sample residuals, forward recursion is used to obtain values for the *pre-forecast* sample innovations.

For dynamic forecasting, one need only obtain innovation values for the q periods prior to the start of the forecast sample; all subsequent innovations are set to zero. EViews obtains estimates of the pre-sample $\epsilon_{S-1}, \epsilon_{S-2}, \dots, \epsilon_{S-q}$ using the recursion:

$$\hat{\epsilon}_t = \hat{u}_t - \hat{\theta}_1 \hat{\epsilon}_{t-1} - \dots - \hat{\theta}_q \hat{\epsilon}_{t-q} \quad (27.13)$$

for $t = 1, \dots, S-1$, where S is the beginning of the forecast period

Static forecasts perform the forward recursion through the *end* of the forecast sample so that innovations are estimated through the last forecast period. Computation of the static forecast for each period uses the q lagged estimated innovations. Extending the recursion produces a series of one-step ahead forecasts of both the structural model and the innovations.

Additional Notes

Note that EViews computes the residuals used in backcast and forward recursion from the observed data and estimated coefficients. If EViews is unable to compute values for the unconditional residuals u_t for a given period, the sequence of innovations and forecasts will be filled with NAs. In particular, static forecasts must have valid data for both the dependent and explanatory variables for all periods from the beginning of estimation sample to the end of the forecast sample, otherwise the backcast values of the innovations, and hence the forecasts will contain NAs. Likewise, dynamic forecasts must have valid data from the beginning of the estimation period through the start of the forecast period.

Example

As an example of forecasting from ARMA models, consider forecasting the monthly new housing starts (HS) series. The estimation period is 1959M01–1984M12 and we forecast for the period 1985M01–1991M12. We estimated the following simple multiplicative seasonal autoregressive model,

```
hs c ar(1) sar(12)
```

yielding:

Dependent Variable: HS
 Method: Least Squares
 Date: 08/08/06 Time: 17:42
 Sample (adjusted): 1960M02 1984M12
 Included observations: 299 after adjustments
 Convergence achieved after 5 iterations

| | Coefficient | Std. Error | t-Statistic | Prob. |
|--------------------|-------------|-----------------------|-------------|-----------|
| C | 7.317283 | 0.071371 | 102.5243 | 0.0000 |
| AR(1) | 0.935392 | 0.021028 | 44.48403 | 0.0000 |
| SAR(12) | -0.113868 | 0.060510 | -1.881798 | 0.0608 |
| R-squared | 0.862967 | Mean dependent var | 7.313496 | |
| Adjusted R-squared | 0.862041 | S.D. dependent var | 0.239053 | |
| S.E. of regression | 0.088791 | Akaike info criterion | -1.995080 | |
| Sum squared resid | 2.333617 | Schwarz criterion | -1.957952 | |
| Log likelihood | 301.2645 | Hannan-Quinn criter. | -1.980220 | |
| F-statistic | 932.0312 | Durbin-Watson stat | 2.452568 | |
| Prob(F-statistic) | 0.000000 | | | |
| Inverted AR Roots | .94 | .81-.22i | .81+.22i | .59-.59i |
| | .59+.59i | .22+.81i | .22-.81i | -.22+.81i |
| | -.22-.81i | -.59+.59i | -.59-.59i | -.81-.22i |
| | -.81+.22i | | | |

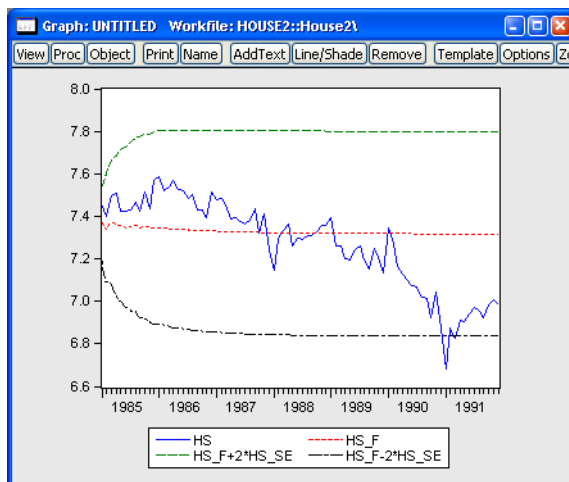
To perform a dynamic forecast from this estimated model, click **Forecast** on the equation toolbar, enter “1985m01 1991m12” in the **Forecast sample** field, then select **Forecast evaluation** and unselect **Forecast graph**. The forecast evaluation statistics for the model are shown below:

| | |
|----------------------------------|----------|
| Forecast: HSF | |
| Actual: HS | |
| Forecast sample: 1985M01 1991M12 | |
| Included observations: 84 | |
| Root Mean Squared Error | 0.210180 |
| Mean Absolute Error | 0.162446 |
| Mean Absolute Percentage Error | 2.276588 |
| Theil Inequality Coefficient | 0.014396 |
| Bias Proportion | 0.076041 |
| Variance Proportion | 0.883149 |
| Covariance Proportion | 0.040809 |

The large variance proportion indicates that the forecasts are not tracking the variation in the actual HS series. To plot the actual and forecasted series together with the two standard error bands, you can type:

```
smp1 1985m01 1991m12
plot hs hs_f hs_f+2*hs_se hs_f-2*hs_se
```

where HS_F and HS_SE are the forecasts and standard errors of HS.



As indicated by the large variance proportion, the forecasts track the seasonal movements in HS only at the beginning of the forecast sample and quickly flatten out to the mean forecast value.

Forecasting from Equations with Expressions

One of the most useful EViews innovations is the ability to estimate and forecast from equations that are specified using expressions or auto-updating series. You may, for example, specify your dependent variable as $\text{LOG}(X)$, or use an auto-updating regressor series EXPZ that is defined using the expression $\text{EXP}(Z)$. Using expressions or auto-updating series in equations creates no added complexity for estimation since EViews simply evaluates the implicit series prior to computing the equation estimator.

The use of expressions in equations does raise issues when computing forecasts from equations. While not particularly complex or difficult to address, the situation does require a basic understanding of the issues involved, and some care must be taken when specifying your forecast.

In discussing the relevant issues, we distinguish between specifications that contain only auto-series expressions such as $\text{LOG}(X)$, and those that contain auto-updating series such as EXPZ .

Forecasting using Auto-series Expressions

When forecasting from an equation that contains only ordinary series or auto-series expressions such as $\text{LOG}(X)$, issues arise only when the dependent variable is specified using an expression.

Point Forecasts

EViews always provides you with the option to forecast the dependent variable expression. If the expression can be normalized (solved for the first series in the expression), EViews also provides you with the option to forecast the normalized series.

For example, suppose you estimated an equation with the specification:

$$(\log(\text{hs}) + \text{sp}) \text{ c hs}(-1)$$

If you press the **Forecast** button, EViews will open a dialog prompting you for your forecast specification.

The resulting **Forecast** dialog is a slightly more complex version of the basic dialog, providing you with a new section allowing you to choose between two series to forecast: the normalized series, HS , or the equation dependent variable, $\text{LOG}(\text{HS}) + \text{SP}$.

Simply select the radio button for the desired forecast series. Note that you are not provided with the opportunity to forecast SP directly since HS , the first series that appears on the left-hand side of the estimation equation, is offered as the choice of normalized series.

It is important to note that the **Dynamic forecast** method is available since EViews is able to determine that the forecast equation has dynamic elements, with HS appearing on the left-hand side of the equation (either directly as HS or in the expression $\text{LOG}(\text{HS}) + \text{SP}$) and on the right-hand side of the equation in lagged form. If you select dynamic forecasting, previ-

ously forecasted values for $HS(-1)$ will be used in forming forecasts of either HS or $\text{LOG}(HS) + SP$.

If the formula can be normalized, EViews will compute the forecasts of the transformed dependent variable by first forecasting the normalized series and then transforming the forecasts of the normalized series. This methodology has important consequences when the formula includes lagged series. For example, consider the following two models:

```
series dhs = d(hs)
equation eq1.ls d(hs) c sp
equation eq2.ls dhs c sp
```

The dynamic forecasts of the first difference $D(HS)$ from the first equation will be numerically identical to those for DHS from the second equation. However, the static forecasts for $D(HS)$ from the two equations will not be identical. In the first equation, EViews knows that the dependent variable is a transformation of HS , so it will use the actual lagged value of HS in computing the static forecast of the first difference $D(HS)$. In the second equation, EViews simply views DY as an ordinary series, so that only the estimated constant and SP are used to compute the static forecast.

One additional word of caution—when you have dependent variables that use lagged values of a series, you should avoid referring to the lagged series before the current series in a dependent variable expression. For example, consider the two equation specifications:

```
d(hs) c sp
(-hs(-1)+hs) c sp
```

Both specifications have the first difference of HS as the dependent variable and the estimation results are identical for the two models. However, if you forecast HS from the second model, EViews will try to calculate the forecasts of HS using leads of the actual series HS . These forecasts of HS will differ from those produced by the first model, which may not be what you expected.

In some cases, EViews will not be able to normalize the dependent variable expression. In this case, the **Forecast** dialog will only offer you the option of forecasting the entire expression. If, for example, you specify your equation as:

```
log(hs)+1/log(hs) = c(1) + c(2)*hs(-1)
```

EViews will not be able to normalize the dependent variable for forecasting. The corresponding **Forecast** dialog will reflect this fact.

This version of the dialog only allows you to forecast the dependent variable expression, since EViews is unable to normalize and solve for HS. Note also that only static forecasts are available for this case since EViews is unable to solve for lagged values of HS on the right hand-side.

Plotted Standard Errors

When you select **Do graph** in the forecast dialog, EViews will plot the forecasts, along with plus and minus two standard error bands. When you estimate an equation with an expression for the left-hand side, EViews will plot the standard error bands for either the normalized or the unnormalized expression, depending upon which term you elect to forecast.

If you elect to predict the normalized dependent variable, EViews will automatically account for any nonlinearity in the standard error transformation. The next section provides additional details on the procedure used to normalize the upper and lower error bounds.

Saved Forecast Standard Errors

If you provide a name in this edit box, EViews will store the standard errors of the underlying series or expression that you chose to forecast.

When the dependent variable of the equation is a simple series or an expression involving only *linear* transformations, the saved standard errors will be exact (except where the forecasts do not account for coefficient uncertainty, as described below). If the dependent variable involves nonlinear transformations, the saved forecast standard errors will be exact if you choose to forecast the entire formula. If you choose to forecast the underlying endogenous series, the forecast uncertainty cannot be computed exactly, and EViews will provide a linear (first-order) approximation to the forecast standard errors.

Consider the following equations involving a formula dependent variable:

$$d(hs) \quad c \quad sp$$

$$\log(hs) \quad c \quad sp$$

For the first equation, you may choose to forecast either HS or D(HS). In both cases, the forecast standard errors will be exact, since the expression involves only linear transformations. The two standard errors will, however, differ in dynamic forecasts since the forecast standard errors for HS take into account the forecast uncertainty from the lagged value of

HS. In the second example, the forecast standard errors for LOG(HS) will be exact. If, however, you request a forecast for HS itself, the standard errors saved in the series will be the approximate (linearized) forecast standard errors for HS.

Note that when EViews displays a graph view of the forecasts together with standard error bands, the standard error bands are always exact. Thus, in forecasting the underlying dependent variable in a nonlinear expression, the standard error bands will not be the same as those you would obtain by constructing series using the linearized standard errors saved in the workfile.

Suppose in our second example above that you store the forecast of HS and its standard errors in the workfile as the series HSHAT and SE_HSHAT. Then the *approximate* two standard error bounds can be generated manually as:

```
series hshat_high1 = hshat + 2*se_hshat
series hshat_low1 = hshat - 2*se_hshat
```

These forecast error bounds will be symmetric about the point forecasts HSHAT.

On the other hand, when EViews plots the forecast error bounds of HS, it proceeds in two steps. It first obtains the forecast of LOG(HS) and its standard errors (named, say, LHSHAT and SE_LHSHAT) and forms the forecast error bounds on LOG(HS):

```
lhshat + 2*se_lhshat
lhshat - 2*se_lhshat
```

It then normalizes (inverts the transformation) of the two standard error bounds to obtain the prediction interval for HS:

```
series hshat_high2 = exp(hshat + 2*se_hshat)
series hshat_low2 = exp(hshat - 2*se_hshat)
```

Because this transformation is a non-linear transformation, these bands will not be symmetric around the forecast.

To take a more complicated example, suppose that you generate the series DLHS and LHS, and then estimate three equivalent models:

```
series dlhs = dlog(hs)
series lhs = log(hs)
equation eq1.ls dlog(hs) c sp
equation eq2.ls d(lhs) c sp
equation eq3.ls dlhs c sp
```

The estimated equations from the three models are numerically identical. If you choose to forecast the underlying dependent (normalized) series from each model, EQ1 will forecast HS, EQ2 will forecast LHS (the log of HS), and EQ3 will forecast DLHS (the first difference of

the logs of HS, $\text{LOG}(\text{HS}) - \text{LOG}(\text{HS}(-1))$. The forecast standard errors saved from EQ1 will be linearized approximations to the forecast standard error of HS, while those from the latter two will be exact for the forecast standard error of $\text{LOG}(\text{HS})$ and the first difference of the logs of HS.

Static forecasts from all three models are identical because the forecasts from previous periods are not used in calculating this period's forecast when performing static forecasts. For dynamic forecasts, the log of the forecasts from EQ1 will be identical to those from EQ2 and the log first difference of the forecasts from EQ1 will be identical to the first difference of the forecasts from EQ2 and to the forecasts from EQ3. For static forecasts, the log first difference of the forecasts from EQ1 will be identical to the first difference of the forecasts from EQ2. However, these forecasts differ from those obtained from EQ3 because EViews does not know that the generated series DLY is actually a difference term so that it does not use the dynamic relation in the forecasts.

Forecasting with Auto-updating series

When forecasting from an equation that contains auto-updating series defined by formulae, the central question is whether EViews interprets the series as ordinary series, or whether it treats the auto-updating series as expressions.

Suppose for example, that we have defined auto-updating series LOGHS and LOGHSLAG, for the log of HAS and the log of $\text{HS}(-1)$, respectively,

```
frml loghs = log(hs)
frml loghslag = log(hs(-1))
```

and that we employ these auto-updating series in estimating an equation specification:

```
loghs c loghslag
```

It is worth pointing out this specification yields results that are identical to those obtained from estimating an equation using the expressions directly using $\text{LOG}(\text{HS})$ and $\text{LOG}(\text{HS}(-1))$:

```
log(hs) c log(hs(-1))
```

The **Forecast** dialog for the first equation specification (using LOGHS and LOGHSLAG) contains an additional combo box allowing you to specify whether to interpret the auto-updating series as ordinary series, or whether to look inside LOGHS and LOGHSLAG to use their expressions.

By default, the combo box is set to **Ignore formulae within series**, so that LOGHS and LOGHSLAG are viewed as ordinary series. Note that since EViews ignores the expressions underlying the auto-updating series, you may only forecast the dependent series LOGHS, and there are no dynamics implied by the equation.

Alternatively, you may instruct EViews to use the expressions in place of all auto-updating series by changing the combo box setting to **Substitute formulae within series**.

If you elect to substitute the formulae, the **Forecast** dialog will change to reflect the use of the underlying expressions as you may now choose between forecasting HS or LOG(HS). We also see that when you use the substituted expressions you are able to perform either dynamic or static forecasting.

It is worth noting that substituting expressions yields a **Forecast** dialog that offers the same options as if you were to forecast from the second equation specification above—using LOG(HS) as the dependent series expression, and LOG(HS(-1)) as an independent series expression.

The screenshot shows the 'Forecast' dialog box in EViews. The 'Forecast equation' is 'UNTITLED'. The 'Series to forecast' is 'LOGHS'. The 'Series names' section has 'Forecast name' set to 'hsf', and 'S.E. (optional)' and 'GARCH(optional)' are empty. The 'Method' section has 'Static forecast (no dynamics in equation)' selected, with 'Structural (ignore ARMA)' unchecked and 'Coef uncertainty in S.E. calc' checked. The 'Forecast sample' is '1989m06 1996m01'. The 'Output' section has 'Forecast graph' and 'Forecast evaluation' checked. The 'Insert actuals for out-of-sample observations' checkbox is checked. The 'OK' and 'Cancel' buttons are at the bottom.

The screenshot shows the 'Forecast' dialog box in EViews with the 'Substitute formulae within series' option selected in the 'Forecast equation' dropdown. The 'Series to forecast' section now has two radio buttons: 'HS' (selected) and 'LOG(HS)'. The 'Series names' section is the same as in the first screenshot. The 'Method' section now has 'Dynamic forecast' selected, with 'Static forecast' unchecked, 'Structural (ignore ARMA)' unchecked, and 'Coef uncertainty in S.E. calc' checked. The 'Forecast sample' is '1989m06 1996m01'. The 'Output' section has 'Forecast graph' and 'Forecast evaluation' checked. The 'Insert actuals for out-of-sample observations' checkbox is checked. The 'OK' and 'Cancel' buttons are at the bottom.

Forecasting with Nonlinear and PDL Specifications

As explained above, forecast errors can arise from two sources: coefficient uncertainty and innovation uncertainty. For linear regression models, the forecast standard errors account for both coefficient and innovation uncertainty. However, if the model is specified by expression (or if it contains a PDL specification), then the standard errors ignore coefficient uncertainty. EViews will display a message in the status line at the bottom of the EViews window when forecast standard errors only account for innovation uncertainty.

For example, consider the three specifications:

$$\log(y) = c + x$$

$$y = c(1) + c(2) * x$$

$$y = \exp(c(1) * x)$$

$$y = c + x \cdot \text{pdl}(z, 4, 2)$$

Forecast standard errors from the first model account for both coefficient and innovation uncertainty since the model is specified by list, and does not contain a PDL specification. The remaining specifications have forecast standard errors that account only for residual uncertainty.

References

Pindyck, Robert S. and Daniel L. Rubinfeld (1998). *Econometric Models and Economic Forecasts*, 4th edition, New York: McGraw-Hill.

Chapter 28. Specification and Diagnostic Tests

Empirical research is usually an interactive process. The process begins with a specification of the relationship to be estimated. Selecting a specification usually involves several choices: the variables to be included, the functional form connecting these variables, and if the data are time series, the dynamic structure of the relationship between the variables.

Inevitably, there is uncertainty regarding the appropriateness of this initial specification. Once you estimate your equation, EViews provides tools for evaluating the quality of your specification along a number of dimensions. In turn, the results of these tests influence the chosen specification, and the process is repeated.

This chapter describes the extensive menu of specification test statistics that are available as views or procedures of an equation object. While we attempt to provide you with sufficient statistical background to conduct the tests, practical considerations ensure that many of the descriptions are incomplete. We refer you to standard statistical and econometric references for further details.

Background

Each test procedure described below involves the specification of a null hypothesis, which is the hypothesis under test. Output from a test command consists of the sample values of one or more test statistics and their associated probability numbers (p -values). The latter indicate the probability of obtaining a test statistic whose absolute value is greater than or equal to that of the sample statistic if the null hypothesis is true. Thus, low p -values lead to the rejection of the null hypothesis. For example, if a p -value lies between 0.05 and 0.01, the null hypothesis is rejected at the 5 percent but not at the 1 percent level.

Bear in mind that there are different assumptions and distributional results associated with each test. For example, some of the test statistics have exact, finite sample distributions (usually t or F -distributions). Others are large sample test statistics with asymptotic χ^2 distributions. Details vary from one test to another and are given below in the description of each test.

The **View** button on the equation toolbar gives you a choice among three categories of tests to check the specification of the equation.

Additional tests are discussed elsewhere in the *User's Guide*.

These tests include unit root tests ([“Performing Unit Root Tests in EViews” on page 88](#)), the Granger causality test ([“Granger Causality” on page 410](#) of the *User's Guide I*), tests specific to binary, order, censored, and count models ([Chapter 30. “Discrete and Limited Dependent Variable](#)

| | |
|-------------------|---|
| Coefficient Tests | ▶ |
| Residual Tests | ▶ |
| Stability Tests | ▶ |

Models,” on page 209), and the Johansen test for cointegration (“How to Perform a Cointegration Test” on page 364).

Coefficient Tests

These tests evaluate restrictions on the estimated coefficients, including the special case of tests for omitted and redundant variables.

Confidence Ellipses

The confidence ellipse view plots the joint confidence region of any two functions of estimated parameters from an EViews estimation object. Along with the ellipses, you can choose to display the individual confidence intervals.

Confidence Ellipse...

Wald - Coefficient Restrictions...

Omitted Variables - Likelihood Ratio...

Redundant Variables - Likelihood Ratio...

Factor Breakpoint Test...

We motivate our discussion of this view by pointing out that the Wald test view (**View/Coefficient Tests/Wald - Coefficient Restrictions...**) allows you to test restrictions on the estimated coefficients from an estimation object. When you perform a Wald test, EViews provides a table of output showing the numeric values associated with the test.

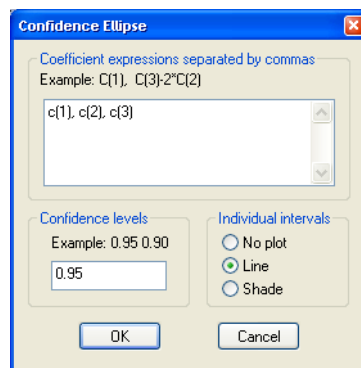
An alternative approach to displaying the results of a Wald test is to display a confidence interval. For a given test size, say 5%, we may display the one-dimensional interval within which the test statistic must lie for us not to reject the null hypothesis. Comparing the realization of the test statistic to the interval corresponds to performing the Wald test.

The one-dimensional confidence interval may be generalized to the case involving two restrictions, where we form a joint confidence region, or confidence ellipse. The confidence ellipse may be interpreted as the region in which the realization of two test statistics must lie for us not to reject the null.

To display confidence ellipses in EViews, simply select **View/Coefficient Tests/Confidence Ellipse...** from the estimation object toolbar. EViews will display a dialog prompting you to specify the coefficient restrictions and test size, and to select display options.

The first part of the dialog is identical to that found in the Wald test view—here, you will enter your coefficient restrictions into the edit box, with multiple restrictions separated by commas. The computation of the confidence ellipse requires a minimum of two restrictions. If you provide more than two restrictions, EViews will display all unique pairs of confidence ellipses.

In this simple example depicted here, we provide a (comma separated) list of coefficients from the estimated equation. This description of the restrictions takes advantage of the fact that EViews interprets any expression without an explicit equal sign as being equal to zero (so that “C(1)” and “C(1) = 0” are equivalent). You may, of course, enter an explicit restriction involving an equal sign (for example, “C(1) + C(2) = C(3)/2”).



Next, select a size or sizes for the confidence ellipses. Here, we instruct EViews to construct a 95% confidence ellipse. Under the null hypothesis, the test statistic values will fall outside of the corresponding confidence ellipse 5% of the time.

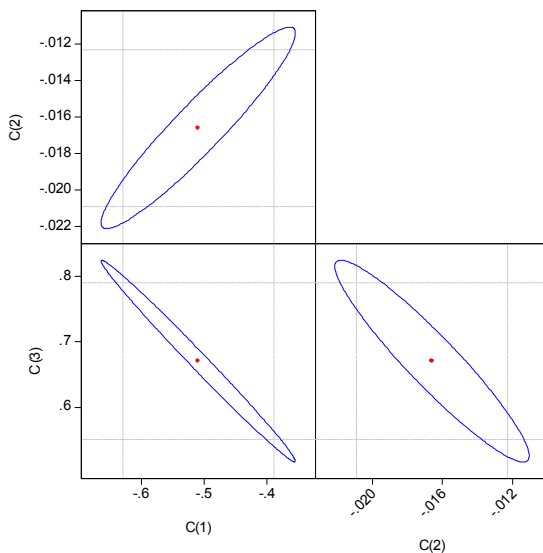
Lastly, we choose a display option for the individual confidence intervals. If you select **Line** or **Shade**, EViews will mark the confidence interval for each restriction, allowing you to see, at a glance, the individual results. **Line** will display the individual confidence intervals as dotted lines; **Shade** will display the confidence intervals as a shaded region. If you select **None**, EViews will not display the individual intervals.

The output depicts three confidence ellipses that result from pairwise tests implied by the three restrictions (“C(1) = 0”, “C(2) = 0”, and “C(3) = 0”).

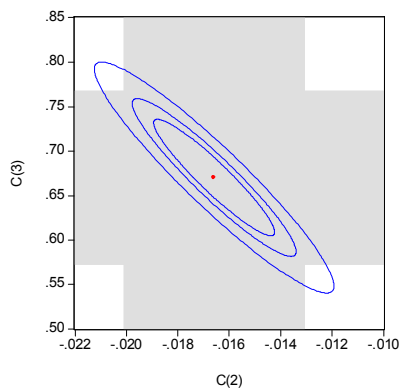
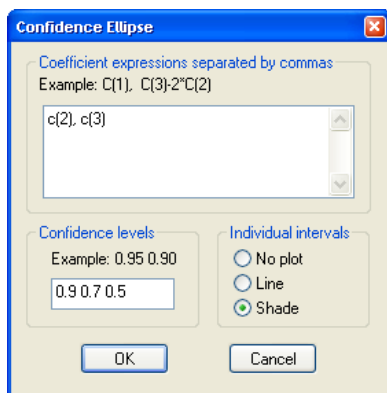
Notice first the presence of the dotted lines showing the corresponding confidence intervals for the individual coefficients.

The next thing that jumps out from this example is that the coefficient estimates are highly correlated—if the estimates were independent, the ellipses would be exact circles.

You can easily see the importance of this correlation. For example, focusing on the ellipse for $C(1)$ and $C(3)$ depicted in the lower left-hand corner, an estimated $C(1)$ of $-.65$ is sufficient to reject the hypothesis that $C(1) = 0$ (since it falls below the end of the univariate confidence interval). If $C(3) = .8$, we cannot reject the joint null that $C(1) = 0$, and $C(3) = 0$ (since $C(1) = -.65$, $C(3) = .8$ falls within the confidence ellipse).



EViews allows you to display more than one size for your confidence ellipses. This feature allows you to draw confidence contours so that you may see how the rejection region changes at different probability values. To do so, simply enter a space delimited list of confidence levels. Note that while the coefficient restriction expressions must be separated by commas, the contour levels must be separated by spaces.



Here, the individual confidence intervals are depicted with shading. The individual intervals are based on the largest size confidence level (which has the widest interval), in this case, 0.9.

Computational Details

Consider two functions of the parameters $f_1(\beta)$ and $f_2(\beta)$, and define the bivariate function $f(\beta) = (f_1(\beta), f_2(\beta))$.

The size α joint confidence ellipse is defined as the set of points b such that:

$$(b - f(\hat{\beta}))'(V(\hat{\beta}))^{-1}(b - f(\hat{\beta})) = c_\alpha \quad (28.1)$$

where $\hat{\beta}$ are the parameter estimates, $V(\hat{\beta})$ is the covariance matrix of $\hat{\beta}$, and c_α is the size α critical value for the related distribution. If the parameter estimates are least-squares based, the $F(2, n - 2)$ distribution is used; if the parameter estimates are likelihood based, the $\chi^2(2)$ distribution will be employed.

The individual intervals are two-sided intervals based on either the t -distribution (in the cases where c_α is computed using the F -distribution), or the normal distribution (where c_α is taken from the χ^2 distribution).

Wald Test (Coefficient Restrictions)

The Wald test computes a test statistic based on the unrestricted regression. The Wald statistic measures how close the unrestricted estimates come to satisfying the restrictions under the null hypothesis. If the restrictions are in fact true, then the unrestricted estimates should come close to satisfying the restrictions.

How to Perform Wald Coefficient Tests

To demonstrate the calculation of Wald tests in EViews, we consider simple examples. Suppose a Cobb-Douglas production function has been estimated in the form:

$$\log Q = A + \alpha \log L + \beta \log K + \epsilon, \quad (28.2)$$

where Q , K and L denote value-added output and the inputs of capital and labor respectively. The hypothesis of constant returns to scale is then tested by the restriction: $\alpha + \beta = 1$.

Estimation of the Cobb-Douglas production function using annual data from 1947 to 1971 provided the following result:

Dependent Variable: LOG(Q)
 Method: Least Squares
 Date: 08/11/97 Time: 16:56
 Sample: 1947 1971
 Included observations: 25

| Variable | Coefficient | Std. Error | t-Statistic | Prob. |
|--------------------|-------------|-----------------------|-------------|-----------|
| C | -2.327939 | 0.410601 | -5.669595 | 0.0000 |
| LOG(L) | 1.591175 | 0.167740 | 9.485970 | 0.0000 |
| LOG(K) | 0.239604 | 0.105390 | 2.273498 | 0.0331 |
| R-squared | 0.983672 | Mean dependent var | | 4.767586 |
| Adjusted R-squared | 0.982187 | S.D. dependent var | | 0.326086 |
| S.E. of regression | 0.043521 | Akaike info criterion | | -3.318997 |
| Sum squared resid | 0.041669 | Schwarz criterion | | -3.172732 |
| Log likelihood | 44.48746 | F-statistic | | 662.6819 |
| Durbin-Watson stat | 0.637300 | Prob(F-statistic) | | 0.000000 |

The sum of the coefficients on LOG(L) and LOG(K) appears to be in excess of one, but to determine whether the difference is statistically relevant, we will conduct the hypothesis test of constant returns.

To carry out a Wald test, choose **View/Coefficient Tests/Wald-Coefficient Restrictions...** from the equation toolbar. Enter the restrictions into the edit box, with multiple coefficient restrictions separated by commas. The restrictions should be expressed as equations involving the estimated coefficients and constants. The coefficients should be referred to as C(1), C(2), and so on, unless you have used a different coefficient vector in estimation.

If you enter a restriction that involves a series name, EViews will prompt you to enter an observation at which the test statistic will be evaluated. The value of the series will at that period will be treated as a constant for purposes of constructing the test statistic.

To test the hypothesis of constant returns to scale, type the following restriction in the dialog box:

$$c(2) + c(3) = 1$$

and click **OK**. EViews reports the following result of the Wald test:

| Wald Test: | | | |
|----------------|----------|---------|-------------|
| Equation: EQ1 | | | |
| Test Statistic | Value | df | Probability |
| Chi-square | 120.0177 | 1 | 0.0000 |
| F-statistic | 120.0177 | (1, 22) | 0.0000 |

| Null Hypothesis Summary: | | |
|------------------------------|----------|-----------|
| Normalized Restriction (= 0) | Value | Std. Err. |
| -1 + C(2) + C(3) | 0.830779 | 0.075834 |

Restrictions are linear in coefficients.

EViews reports an F -statistic and a Chi-square statistic with associated p -values. See “[Wald Test Details](#)” on page 148 for a discussion of these statistics. In addition, EViews reports the value of the normalized (homogeneous) restriction and an associated standard error. In this example, we have a single linear restriction so the two test statistics are identical, with the p -value indicating that we can decisively reject the null hypothesis of constant returns to scale.

To test more than one restriction, separate the restrictions by commas. For example, to test the hypothesis that the elasticity of output with respect to labor is $2/3$ and the elasticity with respect to capital is $1/3$, enter the restrictions as,

$$c(2)=2/3, \quad c(3)=1/3$$

and EViews reports:

| | | | |
|--|-----------|-----------|-------------|
| Wald Test: Equation: EQ1 | | | |
| Test Statistic | Value | df | Probability |
| Chi-square | 53.99105 | 2 | 0.0000 |
| F-statistic | 26.99553 | (2, 22) | 0.0000 |
| Null Hypothesis Summary: | | | |
| Normalized Restriction (= 0) | Value | Std. Err. | |
| -2/3 + C(2) | 0.924508 | 0.167740 | |
| -1/3 + C(1) | -2.661272 | 0.410601 | |
| Restrictions are linear in coefficients. | | | |

Note that in addition to the test statistic summary, we report the values of both of the normalized restrictions, along with their standard errors (the square roots of the diagonal elements of the restriction covariance matrix).

As an example of a nonlinear model with a nonlinear restriction, we estimate a production function of the form:

$$\log Q = \beta_1 + \beta_2 \log(\beta_3 K^{\beta_4} + (1 - \beta_3)L^{\beta_4}) + \epsilon \quad (28.3)$$

and test the constant elasticity of substitution (CES) production function restriction $\beta_2 = 1/\beta_4$. This is an example of a nonlinear restriction. To estimate the (unrestricted) nonlinear model, you should select **Quick/Estimate Equation...** and then enter the following specification:

$$\log(q) = c(1) + c(2) * \log(c(3) * k^{c(4)} + (1 - c(3)) * l^{c(4)})$$

To test the nonlinear restriction, choose **View/Coefficient Tests/Wald-Coefficient Restrictions...** from the equation toolbar and type the following restriction in the Wald Test dialog box:

$$c(2)=1/c(4)$$

The results are presented below:

Wald Test:

Equation: EQ2

| Test Statistic | Value | df | Probability |
|----------------|----------|---------|-------------|
| Chi-square | 0.028508 | 1 | 0.8659 |
| F-statistic | 0.028508 | (1, 21) | 0.8675 |

Null Hypothesis Summary:

| Normalized Restriction (= 0) | Value | Std. Err. |
|------------------------------|----------|-----------|
| C(2) - 1/C(4) | 1.292163 | 7.653088 |

Delta method computed using analytic derivatives.

Since this is a nonlinear equation, we focus on the Chi-square statistic which fails to reject the null hypothesis. Note that EViews reports that it used the delta method (with analytic derivatives) to compute the Wald restriction variance for the nonlinear restriction.

It is well-known that nonlinear Wald tests are not invariant to the way that you specify the nonlinear restrictions. In this example, the nonlinear restriction $\beta_2 = 1/\beta_4$ may equivalently be written as $\beta_2\beta_4 = 1$ or $\beta_4 = 1/\beta_2$ (for nonzero β_2 and β_4). For example, entering the restriction as,

$$C(2) * C(4) = 1$$

yields:

Wald Test:

Equation: EQ2

| Test Statistic | Value | df | Probability |
|----------------|----------|---------|-------------|
| Chi-square | 104.5599 | 1 | 0.0000 |
| F-statistic | 104.5599 | (1, 21) | 0.0000 |

Null Hypothesis Summary:

| Normalized Restriction (= 0) | Value | Std. Err. |
|------------------------------|----------|-----------|
| -1 + C(2)*C(4) | 0.835330 | 0.081691 |

Delta method computed using analytic derivatives.

so that the test now decisively rejects the null hypothesis. We hasten to add that type of inconsistency is not unique to EViews, but is a more general property of the Wald test. Unfortunately, there does not seem to be a general solution to this problem (see Davidson and MacKinnon, 1993, Chapter 13).

Wald Test Details

Consider a general nonlinear regression model:

$$y = f(\beta) + \epsilon \quad (28.4)$$

where y and ϵ are T -vectors and β is a k -vector of parameters to be estimated. Any restrictions on the parameters can be written as:

$$H_0: g(\beta) = 0, \quad (28.5)$$

where g is a smooth function, $g: R^k \rightarrow R^q$, imposing q restrictions on β . The Wald statistic is then computed as:

$$W = g(\beta)' \left(\frac{\partial g(\beta)}{\partial \beta} \hat{V}(b) \frac{\partial g(\beta)}{\partial \beta'} \right) g(\beta) \Big|_{\beta=b} \quad (28.6)$$

where T is the number of observations and b is the vector of unrestricted parameter estimates, and where \hat{V} is an estimate of the b covariance. In the standard regression case, \hat{V} is given by:

$$\hat{V}(b) = s^2 \left(\frac{\partial f(\beta)}{\partial \beta} \frac{\partial f(\beta)}{\partial \beta'} \right)^{-1} \Big|_{\beta=b} \quad (28.7)$$

where u is the vector of unrestricted residuals, and s^2 is the usual estimator of the unrestricted residual variance, $s^2 = (u'u)/(N-k)$, but the estimator of V may differ. For example, \hat{V} may be a robust variance matrix estimator computing using White or Newey-West techniques.

More formally, under the null hypothesis H_0 , the Wald statistic has an asymptotic $\chi^2(q)$ distribution, where q is the number of restrictions under H_0 .

For the textbook case of a linear regression model,

$$y = X\beta + \epsilon \quad (28.8)$$

and linear restrictions:

$$H_0: R\beta - r = 0, \quad (28.9)$$

where R is a known $q \times k$ matrix, and r is a q -vector, respectively. The Wald statistic in Equation (28.6) reduces to:

$$W = (Rb - r)' (Rs^2(X'X)^{-1}R')^{-1} (Rb - r), \quad (28.10)$$

which is asymptotically distributed as $\chi^2(q)$ under H_0 .

If we further assume that the errors ϵ are independent and identically normally distributed, we have an exact, finite sample F -statistic:

$$F = \frac{W}{q} = \frac{(\tilde{u}'\tilde{u} - u'u)/q}{(u'u)/(T-k)}, \quad (28.11)$$

where \tilde{u} is the vector of residuals from the restricted regression. In this case, the F -statistic compares the residual sum of squares computed with and without the restrictions imposed.

We remind you that the expression for the finite sample F -statistic in (28.11) is for standard linear regression, and is not valid for more general cases (nonlinear models, ARMA specifications, or equations where the variances are estimated using other methods such as Newey-West or White). In non-standard settings, the reported F -statistic (which EViews always computes as W/q), does not possess the desired finite-sample properties. In these cases, while asymptotically valid, the F -statistic results should be viewed as illustrative and for comparison purposes only.

Omitted Variables

This test enables you to add a set of variables to an existing equation and to ask whether the set makes a significant contribution to explaining the variation in the dependent variable. The null hypothesis H_0 is that the additional set of regressors are not jointly significant.

The output from the test is an F -statistic and a likelihood ratio (LR) statistic with associated p -values, together with the estimation results of the unrestricted model under the alternative. The F -statistic is based on the difference between the residual sums of squares of the restricted and unrestricted regressions and is only valid in linear regression based settings. The LR statistic is computed as:

$$LR = -2(l_r - l_u) \quad (28.12)$$

where l_r and l_u are the maximized values of the (Gaussian) log likelihood function of the unrestricted and restricted regressions, respectively. Under H_0 , the LR statistic has an asymptotic χ^2 distribution with degrees of freedom equal to the number of restrictions (the number of added variables).

Bear in mind that:

- The omitted variables test requires that the same number of observations exist in the original and test equations. If any of the series to be added contain missing observations over the sample of the original equation (which will often be the case when you add lagged variables), the test statistics cannot be constructed.
- The omitted variables test can be applied to equations estimated with linear LS, TSLS, ARCH (mean equation only), binary, ordered, censored, truncated, and count models. The test is available only if you specify the equation by listing the regressors, not by a formula.

To perform an LR test in these settings, you can estimate a separate equation for the unrestricted and restricted models over a common sample, and evaluate the LR statistic and p -value using scalars and the `@cchisq` function, as described above.

How to Perform an Omitted Variables Test

To test for omitted variables, select **View/Coefficient Tests/Omitted Variables-Likelihood Ratio...** In the dialog that opens, list the names of the test variables, each separated by at least one space. Suppose, for example, that the initial regression is:

```
ls log(q) c log(l) log(k)
```

If you enter the list:

```
log(m) log(e)
```

in the dialog, then EViews reports the results of the unrestricted regression containing the two additional explanatory variables, and displays statistics testing the hypothesis that the coefficients on the new variables are jointly zero. The top part of the output depicts the test results:

| Omitted Variables: LOG(M) LOG(E) | | | |
|----------------------------------|----------|-------------|----------|
| F-statistic | 4.267478 | Probability | 0.028611 |
| Log likelihood ratio | 8.884940 | Probability | 0.011767 |

The F -statistic has an exact finite sample F -distribution under H_0 for linear models if the errors are independent and identically distributed normal random variables. The numerator degrees of freedom is the number of additional regressors and the denominator degrees of freedom is the number of observations less the total number of regressors. The log likelihood ratio statistic is the LR test statistic and is asymptotically distributed as a χ^2 with degrees of freedom equal to the number of added regressors.

In our example, the tests reject the null hypothesis that the two series do not belong to the equation at a 5% significance level, but cannot reject the hypothesis at a 1% significance level.

Redundant Variables

The redundant variables test allows you to test for the statistical significance of a subset of your included variables. More formally, the test is for whether a subset of variables in an equation all have zero coefficients and might thus be deleted from the equation. The redundant variables test can be applied to equations estimated by linear LS, TSLS, ARCH (mean equation only), binary, ordered, censored, truncated, and count methods. The test is available only if you specify the equation by listing the regressors, not by a formula.

How to Perform a Redundant Variables Test

To test for redundant variables, select **View/Coefficient Tests/Redundant Variables-Likelihood Ratio...** In the dialog that appears, list the names of each of the test variables, separated by at least one space. Suppose, for example, that the initial regression is:

```
ls log(q) c log(l) log(k) log(m) log(e)
```

If you type the list:

```
log(m) log(e)
```

in the dialog, then EViews reports the results of the restricted regression dropping the two regressors, followed by the statistics associated with the test of the hypothesis that the coefficients on the two variables are jointly zero.

The test statistics are the F -statistic and the Log likelihood ratio. The F -statistic has an exact finite sample F -distribution under H_0 if the errors are independent and identically distributed normal random variables and the model is linear. The numerator degrees of freedom are given by the number of coefficient restrictions in the null hypothesis. The denominator degrees of freedom are given by the total regression degrees of freedom. The LR test is an asymptotic test, distributed as a χ^2 with degrees of freedom equal to the number of excluded variables under H_0 . In this case, there are two degrees of freedom.

Factor Breakpoint Test

The Factor Breakpoint test splits an estimated equation's sample into a number of subsamples classified by one or more variables and examines whether there are significant differences in equations estimated in each of those subsamples. A significant difference indicates a structural change in the relationship. For example, you can use this test to examine whether the demand function for energy differs between the different states of the USA. The test may be used with least squares and two-stage least squares regressions.

By default the Factor Breakpoint test tests whether there is a structural change in all of the equation parameters. However if the equation is linear EViews allows you to test whether there has been a structural change in a subset of the parameters.

To carry out the test, we partition the data by splitting the estimation sample into subsamples of each unique value of the classification variable. Each subsample must contain more observations than the number of coefficients in the equation so that the equation can be estimated. The Factor Breakpoint test compares the sum of squared residuals obtained by fitting a single equation to the entire sample with the sum of squared residuals obtained when separate equations are fit to each subsample of the data.

EViews reports three test statistics for the Factor Breakpoint test. The F -statistic is based on the comparison of the restricted and unrestricted sum of squared residuals and in the simplest case involving two subsamples, is computed as:

$$F = \frac{(\tilde{u}'\tilde{u} - (u_1'u_1 + u_2'u_2))/k}{(u_1'u_1 + u_2'u_2)/(T - 2k)} \quad (28.13)$$

where $\tilde{u}'\tilde{u}$ is the restricted sum of squared residuals, $u_i'u_i$ is the sum of squared residuals from subsample i , T is the total number of observations, and k is the number of param-

ters in the equation. This formula can be generalized naturally to more than two subsamples. The F -statistic has an exact finite sample F -distribution if the errors are independent and identically distributed normal random variables.

The log likelihood ratio statistic is based on the comparison of the restricted and unrestricted maximum of the (Gaussian) log likelihood function. The LR test statistic has an asymptotic χ^2 distribution with degrees of freedom equal to $(m-1)k$ under the null hypothesis of no structural change, where m is the number of subsamples.

The Wald statistic is computed from a standard Wald test of the restriction that the coefficients on the equation parameters are the same in all subsamples. As with the log likelihood ratio statistic, the Wald statistic has an asymptotic χ^2 distribution with $(m-1)k$ degrees of freedom, where m is the number of subsamples.

To apply the breakpoint test, push **View/Coefficient Tests/Factor Breakpoint Test...** on the equation toolbar. In the dialog that appears, list the series that will be used to classify the equation into subsamples. For example, if you have a series, UNION, that contains values representing either union or non-union, entering “union” in the dialog specifies two subsamples. If you have a variable on marriage status, MARRIED, containing values for married and single, entering “union married” will specify 4 subsamples: non-union/married, non-union/single, union/married, and union/single.

For example, suppose we have estimated an equation of

$$\text{lwage} = c(1) + c(2) * \text{grade} + c(3) * \text{age} + c(4) * \text{high}$$

from the CPS88.WF1 dataset. From this equation we can investigate whether the coefficient estimates on the wage equation differ by union membership and marriage status by using the UNION and MARRIED variables in a factor breakpoint test. This test yields the following result:

| | | | |
|---|------------------------|----------------------|--------|
| Factor Breakpoint Test: UNION MARRIED | | | |
| Null Hypothesis: No breaks at specified breakpoints | | | |
| Varying regressors: All equation variables | | | |
| Equation Sample: 1 1000 | | | |
| F-statistic | 6.227078 | Prob. F(12,984) | 0.0000 |
| Log likelihood ratio | 73.19468 | Prob. Chi-Square(12) | 0.0000 |
| Wald Statistic | 74.72494 | Prob. Chi-Square(12) | 0.0000 |
| Factor values: | UNION = 0, MARRIED = 0 | | |
| | UNION = 0, MARRIED = 1 | | |
| | UNION = 1, MARRIED = 0 | | |
| | UNION = 1, MARRIED = 1 | | |

Note all three statistics decisively reject the null hypothesis.

Residual Tests

EViews provides tests for serial correlation, normality, heteroskedasticity, and autoregressive conditional heteroskedasticity in the residuals from your estimated equation. Not all of these tests are available for every specification.

Correlograms and Q-statistics

This view displays the autocorrelations and partial autocorrelations of the equation residuals up to the specified number of lags. Further details on these statistics and the Ljung-Box Q -statistics that are also computed are provided in “Q-Statistics” on page 326 in the *User’s Guide I*.

| |
|-------------------------------|
| Correlogram - Q-statistics |
| Correlogram Squared Residuals |
| Histogram - Normality Test |
| Serial Correlation LM Test... |
| Heteroskedasticity Tests... |

This view is available for the residuals from least squares, two-stage least squares, nonlinear least squares and binary, ordered, censored, and count models. In calculating the probability values for the Q -statistics, the degrees of freedom are adjusted to account for estimated ARMA terms.

To display the correlograms and Q -statistics, push **View/Residual Tests/Correlogram-Q-statistics** on the equation toolbar. In the **Lag Specification** dialog box, specify the number of lags you wish to use in computing the correlogram.

Correlograms of Squared Residuals

This view displays the autocorrelations and partial autocorrelations of the squared residuals up to any specified number of lags and computes the Ljung-Box Q -statistics for the corresponding lags. The correlograms of the squared residuals can be used to check autoregressive conditional heteroskedasticity (ARCH) in the residuals; see also “ARCH LM Test” on page 158, below.

If there is no ARCH in the residuals, the autocorrelations and partial autocorrelations should be zero at all lags and the Q -statistics should not be significant; see Chapter 11, page 324 of the *User’s Guide I*, for a discussion of the correlograms and Q -statistics.

This view is available for equations estimated by least squares, two-stage least squares, and nonlinear least squares estimation. In calculating the probability for Q -statistics, the degrees of freedom are adjusted for the inclusion of ARMA terms.

To display the correlograms and Q -statistics of the squared residuals, push **View/Residual Tests/Correlogram Squared Residuals** on the equation toolbar. In the **Lag Specification** dialog box that opens, specify the number of lags over which to compute the correlograms.

Histogram and Normality Test

This view displays a histogram and descriptive statistics of the residuals, including the Jarque-Bera statistic for testing normality. If the residuals are normally distributed, the histogram should be bell-shaped and the Jarque-Bera statistic should not be significant; see [Chapter 11, page 308](#) of the *User's Guide I*, for a discussion of the Jarque-Bera test. This view is available for residuals from least squares, two-stage least squares, nonlinear least squares, and binary, ordered, censored, and count models.

To display the histogram and Jarque-Bera statistic, select **View/Residual Tests/Histogram-Normality**. The Jarque-Bera statistic has a χ^2 distribution with two degrees of freedom under the null hypothesis of normally distributed errors.

Serial Correlation LM Test

This test is an alternative to the Q -statistics for testing serial correlation. The test belongs to the class of asymptotic (large sample) tests known as Lagrange multiplier (LM) tests.

Unlike the Durbin-Watson statistic for AR(1) errors, the LM test may be used to test for higher order ARMA errors and is applicable whether or not there are lagged dependent variables. Therefore, we recommend its use (in preference to the DW statistic) whenever you are concerned with the possibility that your errors exhibit autocorrelation.

The null hypothesis of the LM test is that there is no serial correlation up to lag order p , where p is a pre-specified integer. The local alternative is ARMA(r, q) errors, where the number of lag terms $p = \max(r, q)$. Note that this alternative includes both AR(p) and MA(p) error processes, so that the test may have power against a variety of alternative autocorrelation structures. See Godfrey (1988), for further discussion.

The test statistic is computed by an auxiliary regression as follows. First, suppose you have estimated the regression;

$$y_t = X_t\beta + \epsilon_t \quad (28.14)$$

where b are the estimated coefficients and ϵ are the errors. The test statistic for lag order p is based on the auxiliary regression for the residuals $e = y - X\beta$:

$$e_t = X_t\gamma + \left(\sum_{s=1}^p \alpha_s e_{t-s} \right) + v_t. \quad (28.15)$$

Following the suggestion by Davidson and MacKinnon (1993), EViews sets any presample values of the residuals to 0. This approach does not affect the asymptotic distribution of the statistic, and Davidson and MacKinnon argue that doing so provides a test statistic which has better finite sample properties than an approach which drops the initial observations.

This is a regression of the residuals on the original regressors X and lagged residuals up to order p . EViews reports two test statistics from this test regression. The F -statistic is an omitted variable test for the joint significance of all lagged residuals. Because the omitted variables are residuals and not independent variables, the exact finite sample distribution of the F -statistic under H_0 is still not known, but we present the F -statistic for comparison purposes.

The Obs*R-squared statistic is the Breusch-Godfrey LM test statistic. This LM statistic is computed as the number of observations, times the (uncentered) R^2 from the test regression. Under quite general conditions, the LM test statistic is asymptotically distributed as a $\chi^2(p)$.

The serial correlation LM test is available for residuals from either least squares or two-stage least squares estimation. The original regression may include AR and MA terms, in which case the test regression will be modified to take account of the ARMA terms. Testing in 2SLS settings involves additional complications, see Wooldridge (1990) for details.

To carry out the test, push **View/Residual Tests/Serial Correlation LM Test...** on the equation toolbar and specify the highest order of the AR or MA process that might describe the serial correlation. If the test indicates serial correlation in the residuals, LS standard errors are invalid and should not be used for inference.

Heteroskedasticity Tests

This set of tests allows you to test for a range of specifications of heteroskedasticity in the residuals of your equation. Ordinary least squares estimates are consistent in the presence of heteroskedasticity, but the conventional computed standard errors are no longer valid. If you find evidence of heteroskedasticity, you should either choose the robust standard errors option to correct the standard errors (see [“Heteroskedasticity Consistent Covariances \(White\)” on page 35](#)) or you should model the heteroskedasticity to obtain more efficient estimates using weighted least squares.

EViews lets you employ a number of different heteroskedasticity tests, or to use our custom test wizard to test for departures from heteroskedasticity using a combination of methods. Each of these tests involve performing an auxiliary regression using the residuals from the original equation. These tests are available for equations estimated by least squares, two-stage least squares, and nonlinear least squares. The individual tests are outlined below.

Breusch-Pagan-Godfrey (BPG)

The Breusch-Pagan-Godfrey test (see Breusch-Pagan, 1979, and Godfrey, 1978) is a Lagrange multiplier test of the null hypothesis of no heteroskedasticity against heteroskedasticity of the form $\sigma_t^2 = \sigma^2 h(z_t' \alpha)$, where z_t is a vector of independent variables. Usually this vector contains the regressors from the original least squares regression, but it is not necessary.

The test is performed by completing an auxiliary regression of the log of the original equation's squared residuals on $(1, z_t)$. The explained sum of squares from this auxiliary regression is then divided by $2\hat{\sigma}^4$ to give an LM statistic, which follows a χ^2 -distribution with degrees of freedom equal to the number of variables in z under the null hypothesis of no heteroskedasticity. Koenker (1981) suggested that a more easily computed statistic of Obs*R-squared (where R^2 is from the auxiliary regression) be used. Koenker's statistic is also distributed as a χ^2 with degrees of freedom equal to the number of variables in z . Along with these two statistics, EViews also quotes an F -statistic for a redundant variable test for the joint significance of the variables in z in the auxiliary regression.

As an example of a BPG test suppose we had an original equation of

$$\log(m1) = c(1) + c(2)*\log(ip) + c(3)*tb3$$

and we believed that there was heteroskedasticity in the residuals that depended on a function of LOG(IP) and TB3, then the following auxiliary regression could be performed

$$\text{resid}^2 = c(1) + c(2)*\log(ip) + c(3)*tb3$$

Note that both the ARCH and White tests outlined below can be seen as Breusch-Pagan-Godfrey type tests, since both are auxiliary regressions of the log of squared residuals on a set of regressors and a constant.

Harvey

The Harvey (1976) test for heteroskedasticity is similar to the Breusch-Pagan-Godfrey test. However Harvey tests a null hypothesis of no heteroskedasticity against heteroskedasticity of the form of $\sigma_t^2 = \exp(z_t'\alpha)$, where, again, z_t is a vector of independent variables.

To test for this form of heteroskedasticity, an auxiliary regression of the log of the original equation's squared residuals on $(1, z_t)$ is performed. The LM statistic is then the explained sum of squares from the auxiliary regression divided by $\psi'(0.5)$, the derivative of the log gamma function evaluated at 0.5. This statistic is distributed as a χ^2 with degrees of freedom equal to the number of variables in z . EViews also quotes the Obs*R-squared statistic, and the redundant variable F -statistic.

Glejser

The Glejser (1969) test is also similar to the Breusch-Pagan-Godfrey test. This test tests against an alternative hypothesis of heteroskedasticity of the form $\sigma_t^2 = (\sigma^2 + z_t'a)^m$ with $m = 1, 2$. The auxiliary regression that Glejser proposes regresses the absolute value of the residuals from the original equation upon $(1, z_t)$. An LM statistic can be formed by dividing the explained sum of squares from this auxiliary regression by $((1 - 2/\pi)\hat{\sigma}^2)$. As with the previous tests, this statistic is distributed from a chi-squared distribution with degrees of freedom equal to the number of variables in z . EViews also quotes the Obs*R-squared statistic, and the redundant variable F -statistic.

ARCH LM Test

The ARCH test is a Lagrange multiplier (LM) test for autoregressive conditional heteroskedasticity (ARCH) in the residuals (Engle 1982). This particular heteroskedasticity specification was motivated by the observation that in many financial time series, the magnitude of residuals appeared to be related to the magnitude of recent residuals. ARCH in itself does not invalidate standard LS inference. However, ignoring ARCH effects may result in loss of efficiency; see [Chapter 29. “ARCH and GARCH Estimation,” on page 185](#) for a discussion of estimation of ARCH models in EViews.

The ARCH LM test statistic is computed from an auxiliary test regression. To test the null hypothesis that there is no ARCH up to order q in the residuals, we run the regression:

$$e_t^2 = \beta_0 + \left(\sum_{s=1}^q \beta_s e_{t-s}^2 \right) + v_t, \quad (28.16)$$

where e is the residual. This is a regression of the squared residuals on a constant and lagged squared residuals up to order q . EViews reports two test statistics from this test regression. The F -statistic is an omitted variable test for the joint significance of all lagged squared residuals. The Obs*R-squared statistic is Engle's LM test statistic, computed as the number of observations times the R^2 from the test regression. The exact finite sample distribution of the F -statistic under H_0 is not known, but the LM test statistic is asymptotically distributed as a $\chi^2(q)$ under quite general conditions.

White's Heteroskedasticity Test

White's (1980) test is a test of the null hypothesis of no heteroskedasticity against heteroskedasticity of unknown, general form. The test statistic is computed by an auxiliary regression, where we regress the squared residuals on all possible (nonredundant) cross products of the regressors. For example, suppose we estimated the following regression:

$$y_t = b_1 + b_2 x_t + b_3 z_t + e_t \quad (28.17)$$

where the b are the estimated parameters and e the residual. The test statistic is then based on the auxiliary regression:

$$e_t^2 = \alpha_0 + \alpha_1 x_t + \alpha_2 z_t + \alpha_3 x_t^2 + \alpha_4 z_t^2 + \alpha_5 x_t z_t + v_t. \quad (28.18)$$

Note editions of EViews previous to version 6 included the level values of the regressors (*i.e.* the cross product of the regressors and a constant) whether or not the original regression included a constant term. As of EViews 6.0, this is no longer the case. Level values are only included if the original regression included a constant.

EViews reports three test statistics from the test regression. The F -statistic is a redundant variable test for the joint significance of all cross products, excluding the constant. It is presented for comparison purposes.

The Obs*R-squared statistic is White's test statistic, computed as the number of observations times the centered R^2 from the test regression. The exact finite sample distribution of the F -statistic under H_0 is not known, but White's test statistic is asymptotically distributed as a χ^2 with degrees of freedom equal to the number of slope coefficients (excluding the constant) in the test regression.

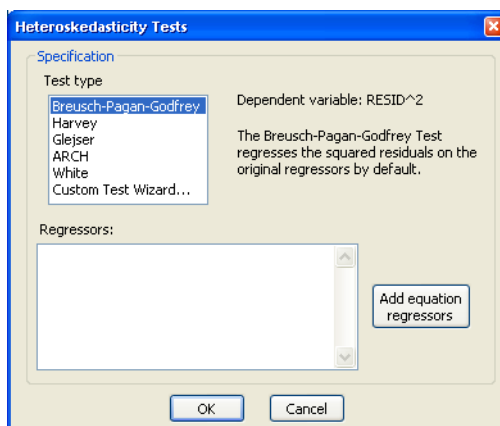
The third statistic, an LM statistic, is the explained sum of squares from the auxiliary regression divided by $2\hat{\sigma}^4$. This, too, is distributed as chi-squared distribution with degrees of freedom equal to the number of slope coefficients (minus the constant) in the auxiliary regression.

White also describes this approach as a general test for model misspecification, since the null hypothesis underlying the test assumes that the errors are both homoskedastic and independent of the regressors, and that the linear specification of the model is correct. Failure of any one of these conditions could lead to a significant test statistic. Conversely, a non-significant test statistic implies that none of the three conditions is violated.

When there are redundant cross-products, EViews automatically drops them from the test regression. For example, the square of a dummy variable is the dummy variable itself, so EViews drops the squared term to avoid perfect collinearity.

Performing a test for Heteroskedasticity in EViews

To carry out any of the heteroskedasticity tests, select **View/Residual Tests/Heteroskedasticity Tests**. This will bring you to the following dialog:



You may choose which type of test to perform by clicking on the name in the **Test type** box. The remainder of the dialog will change, allowing you to specify various options for the selected test.

The BPG, Harvey and Glejser tests allow you to specify which variables to use in the auxiliary regression. Note that you may choose to add all of the variables used in the original equation by pressing the **Add equation regressors** button. If the original equation was non-linear this button will add the coefficient gradients from that equation. Individual gradients can be added by using the @grad keyword to add the *i*-th gradient (e.g., “@grad(2)”).

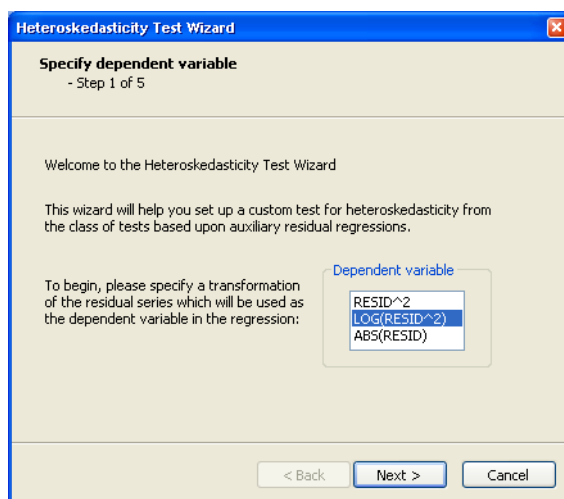
The ARCH test simply lets you specify the number of lags to include for the ARCH specification.

The White test lets you choose whether to include cross terms or no cross terms using the **Include cross terms** checkbox. The cross terms version of the test is the original version of White's test that includes all of the cross product terms. However, the number of cross-product terms increases with the square of the number of right-hand side variables in the regression; with large numbers of regressors, it may not be practical to include all of these terms. The no cross terms specification runs the test regression using only squares of the regressors.

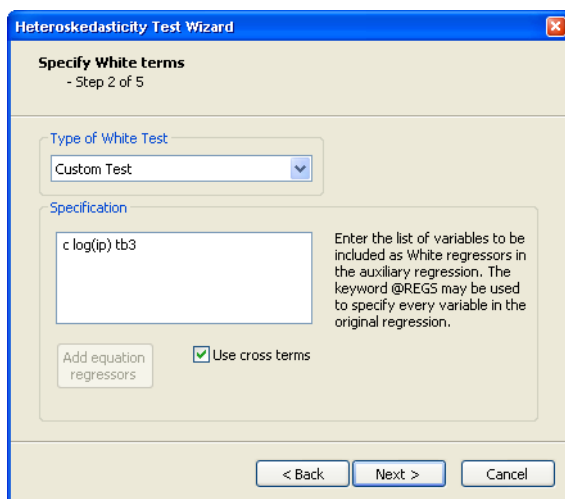
The **Custom Test Wizard** lets you combine or specify in greater detail the various tests. The following example, using EQ1 from the BASICS.WF1 workfile, shows how to use the Custom Wizard. The equation has the following specification:

$$\log(m1) = c(1) + c(2) * \log(ip) + c(3) * tb3$$

The first page of the wizard allows you to choose which transformation of the residuals you want to use as the dependent variable in the auxiliary regression. Note this is really a choice between doing a Breusch-Pagan-Godfrey, a Harvey, or a Glejser type test. In our example we choose to use the LOG of the squared residuals:



Once you have chosen a dependent variable, click on **Next**. Step two of the wizard lets you decide whether to include a White specification. If you check the **Include White specification** checkbox and click on Next, EViews will display the **White Specification** page which lets you specify options for the test. If you do not elect to include a White specification and click on **Next**, EViews will skip the **White Specification** page, and continue on to the next section of the wizard.



There are two parts to the dialog. In the upper section you may use the **Type of White Test** combo box to select the basic test.

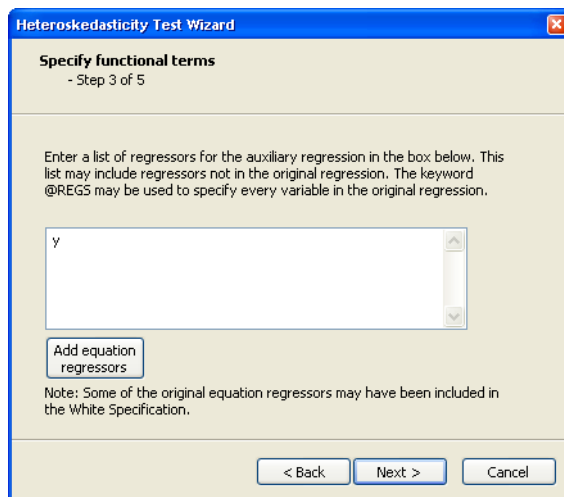
You may choose to include cross terms or not, whether to run an EViews 5 compatible test (as noted above, the auxiliary regression run by EViews differs slightly in Version 6 when there is no constant in the original equation), or, by choosing

Custom, whether to include a set of variables not identical to those used in the original equation. The custom test allows you to perform a test where you include the squares and cross products of an arbitrary set of regressors. Note if you when you provide a set of variables that differs from those in the original equation, the test is no longer a White test, but could still be a valid test for heteroskedasticity. For our example we choose to include C and LOG(IP) as regressors, and choose to use cross terms.

| |
|-------------------------------------|
| Standard Test (cross terms) |
| Standard Test (no cross terms) |
| V5 Compatible Test (cross terms) |
| V5 Compatible Test (no cross terms) |
| Custom Test |

Click on **Next** to continue to the next section of the wizard. EViews prompts you for whether you wish to add any other variables as part of a BPG/Harvey/Glejser specification. If you elect to do so, EViews will display a dialog prompting you to add additional regressors. Note that if you have already included a White specification and your original equation had a constant term, your auxiliary regression will already include level values of the original equation regressors (since the cross-product of the constant term and those regressors is

their level values). In our example we choose to add the variable Y to the auxiliary regression:



Heteroskedasticity Test Wizard
- Step 3 of 5

Enter a list of regressors for the auxiliary regression in the box below. This list may include regressors not in the original regression. The keyword @REGS may be used to specify every variable in the original regression.

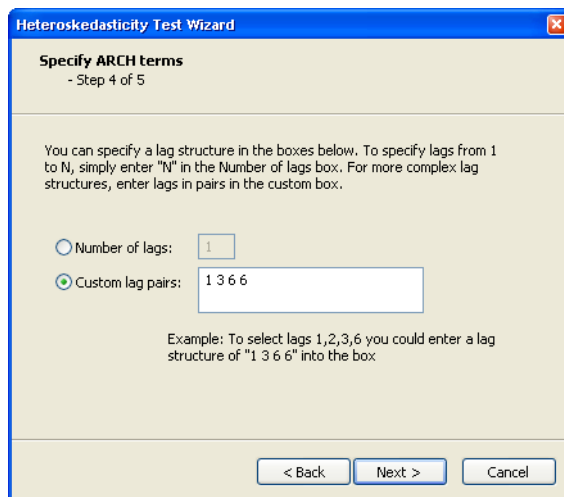
y

Add equation regressors

Note: Some of the original equation regressors may have been included in the White Specification.

< Back Next > Cancel

Next we can add ARCH terms to the auxiliary regression. The ARCH specification lets you specify a lag structure. You can either specify a number of lags, so that the auxiliary regression will include lagged values of the squared residuals up to the number you choose, or you may provide a custom lag structure. Custom structures are entered in pairs of lags. In our example we choose to include lags of 1,2, 3 and 6:



Heteroskedasticity Test Wizard
- Step 4 of 5

You can specify a lag structure in the boxes below. To specify lags from 1 to N, simply enter "N" in the Number of lags box. For more complex lag structures, enter lags in pairs in the custom box.

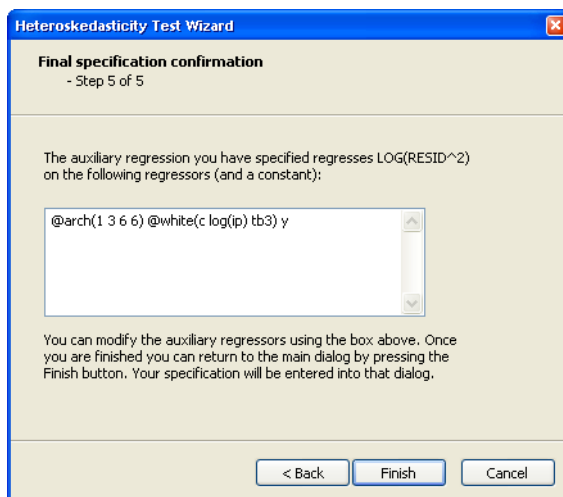
☐ Number of lags: 1

☒ Custom lag pairs: 1 3 6 6

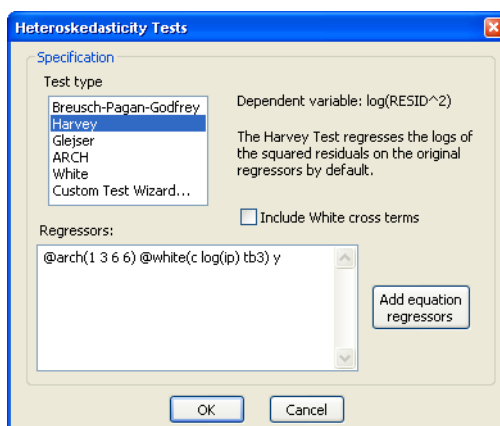
Example: To select lags 1,2,3,6 you could enter a lag structure of "1 3 6 6" into the box

< Back Next > Cancel

The final step of the wizard is to view the final specification of the auxiliary regression, with all the options you have previously chosen, and make any modifications. For our choices, the final specification looks like this:



Our ARCH specification with lags of 1,2,3,6 is shown first, followed by the White specification, and then the additional term, Y, we added. Upon clicking **Finish** the main **Heteroskedasticity Tests** dialog has been filled out with our specification:



Note, rather than go through the wizard, we could have typed this specification directly into the dialog.

This test results in the following output:

Heteroskedasticity Test: Harvey

| | | | |
|---------------------|----------|----------------------|--------|
| F-statistic | 203.6910 | Prob. F(10,324) | 0.0000 |
| Obs*R-squared | 289.0262 | Prob. Chi-Square(10) | 0.0000 |
| Scaled explained SS | 160.8560 | Prob. Chi-Square(10) | 0.0000 |

Test Equation:

Dependent Variable: LRESID2

Method: Least Squares

Date: 08/30/06 Time: 14:49

Sample (adjusted): 1959M07 1989M12

Included observations: 335 after adjustments

| | Coefficient | Std. Error | t-Statistic | Prob. |
|--------------------|-------------|-----------------------|-------------|--------|
| C | 2.320248 | 10.82443 | 0.214353 | 0.8304 |
| LRESID2(-1) | 0.875599 | 0.055882 | 15.66873 | 0.0000 |
| LRESID2(-2) | 0.061016 | 0.074610 | 0.817805 | 0.4141 |
| LRESID2(-3) | -0.035013 | 0.061022 | -0.573768 | 0.5665 |
| LRESID2(-6) | 0.024621 | 0.036220 | 0.679761 | 0.4971 |
| LOG(IP) | -1.622303 | 5.792786 | -0.280056 | 0.7796 |
| (LOG(IP))^2 | 0.255666 | 0.764826 | 0.334280 | 0.7384 |
| (LOG(IP))*TB3 | -0.040560 | 0.154475 | -0.262566 | 0.7931 |
| TB3 | 0.097993 | 0.631189 | 0.155252 | 0.8767 |
| TB3^2 | 0.002845 | 0.005380 | 0.528851 | 0.5973 |
| Y | -0.023621 | 0.039166 | -0.603101 | 0.5469 |
| R-squared | 0.862765 | Mean dependent var | -4.046849 | |
| Adjusted R-squared | 0.858529 | S.D. dependent var | 1.659717 | |
| S.E. of regression | 0.624263 | Akaike info criterion | 1.927794 | |
| Sum squared resid | 126.2642 | Schwarz criterion | 2.053035 | |
| Log likelihood | -311.9056 | Hannan-Quinn criter. | 1.977724 | |
| F-statistic | 203.6910 | Durbin-Watson stat | 2.130511 | |
| Prob(F-statistic) | 0.000000 | | | |

This output contains both the set of test statistics, and the results of the auxiliary regression on which they are based. All three statistics reject the null hypothesis of homoskedasticity.

Specification and Stability Tests

EViews provides several test statistic views that examine whether the parameters of your model are stable across various subsamples of your data.

One common approach is to split the T observations in your data set of observations into T_1 observations to be used for estimation, and $T_2 = T - T_1$ observations to be used for testing and evaluation. In time series work, you will usually take the first T_1 observations

for estimation and the last T_2 for testing. With cross-section data, you may wish to order the data by some variable, such as household income, sales of a firm, or other indicator variables and use a subset for testing.

Note that the alternative of using all available sample observations for estimation promotes a search for a specification that best fits that specific data set, but does not allow for testing predictions of the model against data that have not been used in estimating the model. Nor does it allow one to test for parameter constancy, stability and robustness of the estimated relationship.

There are no hard and fast rules for determining the relative sizes of T_1 and T_2 . In some cases there may be obvious points at which a break in structure might have taken place—a war, a piece of legislation, a switch from fixed to floating exchange rates, or an oil shock. Where there is no reason *a priori* to expect a structural break, a commonly used rule-of-thumb is to use 85 to 90 percent of the observations for estimation and the remainder for testing.

EViews provides built-in procedures which facilitate variations on this type of analysis.

| |
|------------------------------------|
| Chow Breakpoint Test... |
| Quandt-Andrews Breakpoint Test... |
| Chow Forecast Test... |
| Ramsey RESET Test... |
| Recursive Estimates (OLS only) ... |

Chow's Breakpoint Test

The idea of the breakpoint Chow test is to fit the equation separately for each subsample and to see whether there are significant differences in the estimated equations. A significant difference indicates a structural change in the relationship. For example, you can use this test to examine whether the demand function for energy was the same before and after the oil shock. The test may be used with least squares and two-stage least squares regressions.

By default the Chow breakpoint test tests whether there is a structural change in all of the equation parameters. However if the equation is linear EViews allows you to test whether there has been a structural change in a subset of the parameters.

To carry out the test, we partition the data into two or more subsamples. Each subsample must contain more observations than the number of coefficients in the equation so that the equation can be estimated. The Chow breakpoint test compares the sum of squared residuals obtained by fitting a single equation to the entire sample with the sum of squared residuals obtained when separate equations are fit to each subsample of the data.

EViews reports three test statistics for the Chow breakpoint test. The F -statistic is based on the comparison of the restricted and unrestricted sum of squared residuals and in the simplest case involving a single breakpoint, is computed as:

$$F = \frac{(\tilde{u}'\tilde{u} - (u_1'u_1 + u_2'u_2))/k}{(u_1'u_1 + u_2'u_2)/(T - 2k)}, \quad (28.19)$$

where $\tilde{u}'\tilde{u}$ is the restricted sum of squared residuals, $u_i'u_i$ is the sum of squared residuals from subsample i , T is the total number of observations, and k is the number of parameters in the equation. This formula can be generalized naturally to more than one breakpoint. The F -statistic has an exact finite sample F -distribution if the errors are independent and identically distributed normal random variables.

The log likelihood ratio statistic is based on the comparison of the restricted and unrestricted maximum of the (Gaussian) log likelihood function. The LR test statistic has an asymptotic χ^2 distribution with degrees of freedom equal to $(m-1)k$ under the null hypothesis of no structural change, where m is the number of subsamples.

The Wald statistic is computed from a standard Wald test of the restriction that the coefficients on the equation parameters are the same in all subsamples. As with the log likelihood ratio statistic, the Wald statistic has an asymptotic χ^2 distribution with $(m-1)k$ degrees of freedom, where m is the number of subsamples.

One major drawback of the breakpoint test is that each subsample requires at least as many observations as the number of estimated parameters. This may be a problem if, for example, you want to test for structural change between wartime and peacetime where there are only a few observations in the wartime sample. The Chow forecast test, discussed below, should be used in such cases.

To apply the Chow breakpoint test, push **View/Stability Tests/Chow Breakpoint Test...** on the equation toolbar. In the dialog that appears, list the dates or observation numbers for the breakpoints. For example, if your original equation was estimated from 1950 to 1994, entering:

1960

in the dialog specifies two subsamples, one from 1950 to 1959 and one from 1960 to 1994. Typing:

1960 1970

specifies three subsamples, 1950 to 1959, 1960 to 1969, and 1970 to 1994.

Quandt-Andrews Breakpoint Test

The Quandt-Andrews Breakpoint Test tests for one or more unknown structural breakpoints in the sample for a specified equation. The idea behind the Quandt-Andrews test is that a single Chow Breakpoint Test is performed at every observation between two dates, or observations, τ_1 and τ_2 . The k test statistics from those Chow tests are then summarized into one test statistic for a test against the null hypothesis of no breakpoints between τ_1 and τ_2 .

By default the test tests whether there is a structural change in all of the original equation parameters. For linear specifications, EViews also allows you to test whether there has been a structural change in a subset of the parameters.

From each individual Chow Breakpoint Test two statistics are retained, the Likelihood Ratio F -statistic and the Wald F -statistic. The Likelihood Ratio F -statistic is based on the comparison of the restricted and unrestricted sums of squared residuals. The Wald F -statistic is computed from a standard Wald test of the restriction that the coefficients on the equation parameters are the same in all subsamples. Note that in linear equations these two statistics will be identical. For more details on these statistics, see [“Chow's Breakpoint Test” on page 165](#).

The individual test statistics can be summarized into three different statistics; the Sup or Maximum statistic, the Exp Statistic, and the Ave statistic (see Andrews, 1993 and Andrews and Ploberger, 1994). The Maximum statistic is simply the maximum of the individual Chow F -statistics:

$$\text{MaxF} = \max_{\tau_1 \leq \tau \leq \tau_2} (F(\tau)) \quad (28.20)$$

The Exp statistic takes the form:

$$\text{ExpF} = \ln \left(\frac{1}{k} \sum_{\tau = \tau_1}^{\tau_2} \exp \left(\frac{1}{2} F(\tau) \right) \right) \quad (28.21)$$

The Ave statistic is the simple average of the individual F -statistics:

$$\text{AveF} = \frac{1}{k} \sum_{\tau = \tau_1}^{\tau_2} F(\tau) \quad (28.22)$$

The distribution of these test statistics is non-standard. Andrews (1993) developed their true distribution, and Hansen (1997) provided approximate asymptotic p -values. EViews reports the Hansen p -values. The distribution of these statistics becomes degenerate as τ_1 approaches the beginning of the equation sample, or τ_2 approaches the end of the equation sample. To compensate for this behavior, it is generally suggested that the ends of the equation sample not be included in the testing procedure. A standard level for this “trimming” is 15%, where we exclude the first and last 7.5% of the observations. EViews sets trimming at 15% by default, but also allows the user to choose other levels. Note EViews only allows symmetric trimming, *i.e.* the same number of observations are removed from the beginning of the estimation sample as from the end.

The Quandt-Andrews Breakpoint Test can be evaluated for an equation by selecting **View/ Stability Tests/Quandt-Andrews Breakpoint Test...** from the equation toolbar. The resulting dialog allows you to choose the level of symmetric observation trimming for the test, and, if your original equation was linear, which variables you wish to test for the unknown break point. You may also choose to save the individual Chow Breakpoint test statistics into new series within your workfile by entering a name for the new series.

As an example we estimate a consumption function, EQ1 in the workfile COEF_TEST.WF1, using annual data from 1947 to 1971. To test for an unknown structural break point amongst all the original regressors we run the Quandt-Andrews test with 15% trimming. This test gives the following results:

Quandt-Andrews unknown breakpoint test
 Null Hypothesis: No breakpoints within trimmed data
 Varying regressors: All equation variables
 Equation Sample: 1947 1971
 Test Sample: 1951 1967
 Number of breaks compared: 17

| Statistic | Value | Prob. |
|---------------------------------|----------|--------|
| Maximum LR F-statistic (1962) | 7.783485 | 0.4241 |
| Maximum Wald F-statistic (1962) | 7.783485 | 0.4241 |
| Exp LR F-statistic | 2.392496 | 0.2748 |
| Exp Wald F-statistic | 2.392496 | 0.2748 |
| Ave LR F-statistic | 3.808887 | 0.2435 |
| Ave Wald F-statistic | 3.808887 | 0.2435 |

Note: probabilities calculated using Hansen's (1997) method

Note all three of the summary statistic measures fail to reject the null hypothesis of no structural breaks within the 17 possible dates tested. The maximum statistic was in 1962, and that is the most likely breakpoint location. Also, since the original equation was linear, note that the LR F -statistic is identical to the Wald F -statistic.

Chow's Forecast Test

The Chow forecast test estimates two models—one using the full set of data T , and the other using a long subperiod T_1 . Differences between the results for the two estimated models casts doubt on the stability of the estimated relation over the sample period. The Chow forecast test can be used with least squares and two-stage least squares regressions.

EViews reports two test statistics for the Chow forecast test. The F -statistic is computed as

$$F = \frac{(\tilde{u}'\tilde{u} - u'u)/T_2}{u'u/(T_1 - k)}, \quad (28.23)$$

where $\tilde{u}'\tilde{u}$ is the residual sum of squares when the equation is fitted to all T sample observations, $u'u$ is the residual sum of squares when the equation is fitted to T_1 observations, and k is the number of estimated coefficients. This F -statistic follows an exact finite sample F -distribution if the errors are independent, and identically, normally distributed.

The log likelihood ratio statistic is based on the comparison of the restricted and unrestricted maximum of the (Gaussian) log likelihood function. Both the restricted and unrestricted log likelihood are obtained by estimating the regression using the whole sample. The restricted regression uses the original set of regressors, while the unrestricted regression adds a dummy variable for each forecast point. The LR test statistic has an asymptotic χ^2 distribution with degrees of freedom equal to the number of forecast points T_2 under the null hypothesis of no structural change.

To apply Chow's forecast test, push **View/Stability Tests/Chow Forecast Test...** on the equation toolbar and specify the date or observation number for the beginning of the forecasting sample. The date should be within the current sample of observations.

As an example, using the COEF_TEST2.WF1 workfile, suppose we estimate a consumption function, EQ1, using quarterly data from 1947q1 to 1994q4 and specify 1973q1 as the first observation in the forecast period. The test reestimates the equation for the period 1947q1 to 1972q4, and uses the result to compute the prediction errors for the remaining quarters, and the top portion of the table shows the following results:

| Chow Forecast Test: Forecast from 1973Q1 to 1994Q4 | | | |
|--|----------|----------------------|--------|
| F-statistic | 0.708348 | Prob. F(88,102) | 0.9511 |
| Log likelihood ratio | 91.57088 | Prob. Chi-Square(88) | 0.3761 |

Neither of the forecast test statistics reject the null hypothesis of no structural change in the consumption function before and after 1973q1.

If we test the same hypothesis using the Chow breakpoint test, the result is:

| Chow Breakpoint Test: 1973Q1 | | | |
|---|----------|---------------------|--------|
| Null Hypothesis: No breaks at specified breakpoints | | | |
| Varying regressors: All equation variables | | | |
| Equation Sample: 1947Q1 1994Q4 | | | |
| F-statistic | 38.39198 | Prob. F(2,188) | 0.0000 |
| Log likelihood ratio | 65.75466 | Prob. Chi-Square(2) | 0.0000 |
| Wald Statistic | 76.78396 | Prob. Chi-Square(2) | 0.0000 |

Note that the breakpoint test statistics decisively reject the hypothesis from above. This example illustrates the possibility that the two Chow tests may yield conflicting results.

Ramsey's RESET Test

RESET stands for *Regression Specification Error Test* and was proposed by Ramsey (1969). The classical normal linear regression model is specified as:

$$y = X\beta + \epsilon, \quad (28.24)$$

where the disturbance vector ϵ is presumed to follow the multivariate normal distribution $N(0, \sigma^2 I)$. Specification error is an omnibus term which covers any departure from the assumptions of the maintained model. Serial correlation, heteroskedasticity, or non-normality of all violate the assumption that the disturbances are distributed $N(0, \sigma^2 I)$. Tests for these specification errors have been described above. In contrast, RESET is a general test for the following types of specification errors:

- Omitted variables; X does not include all relevant variables.
- Incorrect functional form; some or all of the variables in y and X should be transformed to logs, powers, reciprocals, or in some other way.
- Correlation between X and ϵ , which may be caused, among other things, by measurement error in X , simultaneity, or the presence of lagged y values and serially correlated disturbances.

Under such specification errors, LS estimators will be biased and inconsistent, and conventional inference procedures will be invalidated. Ramsey (1969) showed that any or all of these specification errors produce a non-zero mean vector for ϵ . Therefore, the null and alternative hypotheses of the RESET test are:

$$\begin{aligned} H_0: \epsilon &\sim N(0, \sigma^2 I) \\ H_1: \epsilon &\sim N(\mu, \sigma^2 I) \quad \mu \neq 0 \end{aligned} \quad (28.25)$$

The test is based on an augmented regression:

$$y = X\beta + Z\gamma + \epsilon. \quad (28.26)$$

The test of specification error evaluates the restriction $\gamma = 0$. The crucial question in constructing the test is to determine what variables should enter the Z matrix. Note that the Z matrix may, for example, be comprised of variables that are not in the original specification, so that the test of $\gamma = 0$ is simply the omitted variables test described above.

In testing for incorrect functional form, the nonlinear part of the regression model may be some function of the regressors included in X . For example, if a linear relation,

$$y = \beta_0 + \beta_1 X + \epsilon, \quad (28.27)$$

is specified instead of the true relation:

$$y = \beta_0 + \beta_1 X + \beta_2 X^2 + \epsilon \quad (28.28)$$

the augmented model has $Z = X^2$ and we are back to the omitted variable case. A more general example might be the specification of an additive relation,

$$y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \epsilon \quad (28.29)$$

instead of the (true) multiplicative relation:

$$y = \beta_0 X_1^{\beta_1} X_2^{\beta_2} + \epsilon. \quad (28.30)$$

A Taylor series approximation of the multiplicative relation would yield an expression involving powers and cross-products of the explanatory variables. Ramsey's suggestion is to include powers of the predicted values of the dependent variable (which are, of course, linear combinations of powers and cross-product terms of the explanatory variables) in Z :

$$Z = [\hat{y}^2, \hat{y}^3, \hat{y}^4, \dots] \quad (28.31)$$

where \hat{y} is the vector of fitted values from the regression of y on X . The superscripts indicate the powers to which these predictions are raised. The first power is not included since it is perfectly collinear with the X matrix.

Output from the test reports the test regression and the F -statistic and log likelihood ratio for testing the hypothesis that the coefficients on the powers of fitted values are all zero. A study by Ramsey and Alexander (1984) showed that the RESET test could detect specification error in an equation which was known *a priori* to be misspecified but which nonetheless gave satisfactory values for all the more traditional test criteria—goodness of fit, test for first order serial correlation, high t -ratios.

To apply the test, select **View/Stability Tests/Ramsey RESET Test...** and specify the number of fitted terms to include in the test regression. The fitted terms are the powers of the fitted values from the original regression, starting with the square or second power. For example, if you specify 1, then the test will add \hat{y}^2 in the regression, and if you specify 2, then the test will add \hat{y}^2 and \hat{y}^3 in the regression, and so on. If you specify a large number of fitted terms, EViews may report a near singular matrix error message since the powers of the fitted values are likely to be highly collinear. The Ramsey RESET test is applicable only to an equation estimated by least squares.

Recursive Least Squares

In recursive least squares the equation is estimated repeatedly, using ever larger subsets of the sample data. If there are k coefficients to be estimated in the b vector, then the first k observations are used to form the first estimate of b . The next observation is then added to the data set and $k + 1$ observations are used to compute the second estimate of b . This process is repeated until all the T sample points have been used, yielding $T - k + 1$ estimates of the b vector. At each step the last estimate of b can be used to predict the next value of the dependent variable. The one-step ahead forecast error resulting from this prediction, suitably scaled, is defined to be a *recursive residual*.

More formally, let X_{t-1} denote the $(t-1) \times k$ matrix of the regressors from period 1 to period $t-1$, and y_{t-1} the corresponding vector of observations on the dependent variable. These data up to period $t-1$ give an estimated coefficient vector, denoted by b_{t-1} . This coefficient vector gives you a forecast of the dependent variable in period t . The forecast is $x_t' b$, where x_t' is the row vector of observations on the regressors in period t . The forecast error is $y_t - x_t' b$, and the forecast variance is given by:

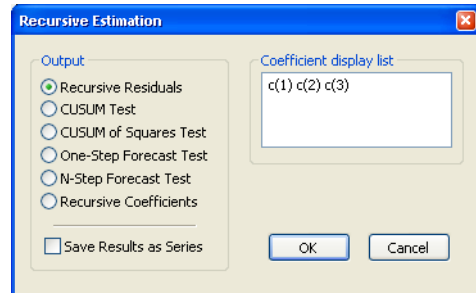
$$\sigma^2(1 + x_t'(X_t'X_t)^{-1}x_t). \quad (28.32)$$

The recursive residual w_t is defined in EViews as:

$$w_t = \frac{(y_t - x_t' b)}{(1 + x_t'(X_t'X_t)^{-1}x_t)^{1/2}}. \quad (28.33)$$

These residuals can be computed for $t = k+1, \dots, T$. If the maintained model is valid, the recursive residuals will be independently and normally distributed with zero mean and constant variance σ^2 .

To calculate the recursive residuals, press **View/Stability Tests/Recursive Estimates (OLS only)...** on the equation toolbar. There are six options available for the recursive estimates view. The recursive estimates view is only available for equations estimated by ordinary least squares without AR and MA terms. The **Save Results as Series** option allows you to save the recursive residuals and recursive coefficients as named series in the workfile; see “[Save Results as Series](#)” on page 175.



Recursive Residuals

This option shows a plot of the recursive residuals about the zero line. Plus and minus two standard errors are also shown at each point. Residuals outside the standard error bands suggest instability in the parameters of the equation.

CUSUM Test

The CUSUM test (Brown, Durbin, and Evans, 1975) is based on the cumulative sum of the recursive residuals. This option plots the cumulative sum together with the 5% critical lines. The test finds parameter instability if the cumulative sum goes outside the area between the two critical lines.

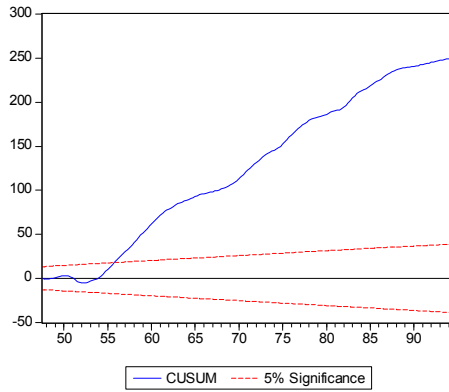
The CUSUM test is based on the statistic:

$$W_t = \sum_{r=k+1}^t w_r / s, \quad (28.34)$$

for $t = k+1, \dots, T$, where w is the recursive residual defined above, and s is the standard error of the regression fitted to all T sample points. If the β vector remains constant from period to period, $E(W_t) = 0$, but if β changes, W_t will tend to diverge from the zero mean value line. The significance of any departure from the zero line is assessed by reference to a pair of 5% significance lines, the distance between which increases with t . The 5% significance lines are found by connecting the points:

$$[k, \pm 0.948(T-k)^{1/2}] \quad \text{and} \quad [T, \pm 3 \times 0.948(T-k)^{1/2}]. \quad (28.35)$$

Movement of W_t outside the critical lines is suggestive of coefficient instability. A sample CUSUM test is given below:



The test clearly indicates instability in the equation during the sample period.

CUSUM of Squares Test

The CUSUM of squares test (Brown, Durbin, and Evans, 1975) is based on the test statistic:

$$W_t = \left(\sum_{r=k+1}^t w_r^2 \right) / \left(\sum_{r=k+1}^T w_r^2 \right). \quad (28.36)$$

The expected value of S under the hypothesis of parameter constancy is:

$$E(S_t) = (t-k)/(T-k) \quad (28.37)$$

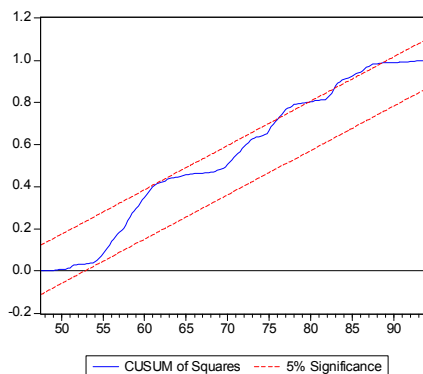
which goes from zero at $t = k$ to unity at $t = T$. The significance of the departure of S from its expected value is assessed by reference to a pair of parallel straight lines around the expected value. See Brown, Durbin, and Evans (1975) or Johnston and DiNardo (1997, Table D.8) for a table of significance lines for the CUSUM of squares test.

The CUSUM of squares test provides a plot of S_t against t and the pair of 5 percent critical lines. As with the CUSUM test, movement outside the critical lines is suggestive of parameter or variance instability.

The cumulative sum of squares is generally within the 5% significance lines, suggesting that the residual variance is somewhat stable.

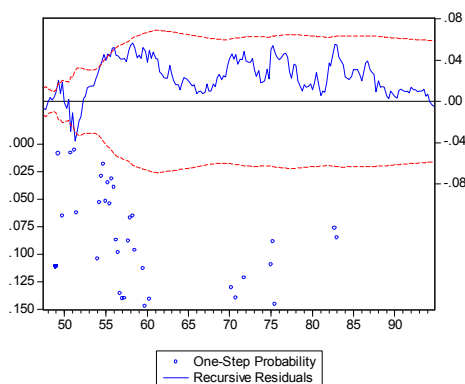
One-Step Forecast Test

If you look back at the definition of the recursive residuals given above, you will see that each recursive residual is the error in a one-step ahead forecast. To test whether the value of the dependent variable at time t might have come from the model fitted to all the data up to that point, each error can be compared with its standard deviation from the full sample.



The **One-Step Forecast Test** option produces a plot of the recursive residuals and standard errors and the sample points whose probability value is at or below 15 percent. The plot can help you spot the periods when your equation is least successful. For example, the one-step ahead forecast test might look like this:

The upper portion of the plot (right vertical axis) repeats the recursive residuals and standard errors displayed by the **Recursive Residuals** option. The lower portion of the plot (left vertical axis) shows the probability values for those sample points where the hypothesis of parameter constancy would be rejected at the 5, 10, or 15 percent levels. The points with p -values less the 0.05 correspond to those points where the recursive residuals go outside the two standard error bounds.



For the test equation, there is evidence of instability early in the sample period.

N-Step Forecast Test

This test uses the recursive calculations to carry out a sequence of Chow Forecast tests. In contrast to the single Chow Forecast test described earlier, this test does not require the

specification of a forecast period— it automatically computes all feasible cases, starting with the smallest possible sample size for estimating the forecasting equation and then adding one observation at a time. The plot from this test shows the recursive residuals at the top and significant probabilities (based on the F -statistic) in the lower portion of the diagram.

Recursive Coefficient Estimates

This view enables you to trace the evolution of estimates for any coefficient as more and more of the sample data are used in the estimation. The view will provide a plot of selected coefficients in the equation for all feasible recursive estimations. Also shown are the two standard error bands around the estimated coefficients.

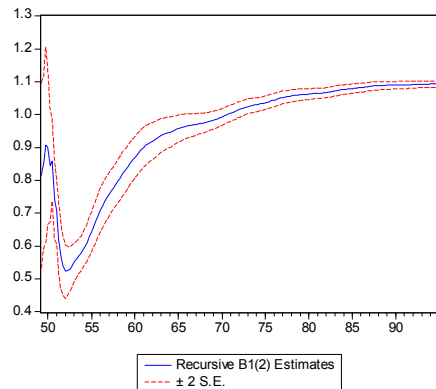
If the coefficient displays significant variation as more data is added to the estimating equation, it is a strong indication of instability. Coefficient plots will sometimes show dramatic jumps as the postulated equation tries to digest a structural break.

To view the recursive coefficient estimates, click the **Recursive Coefficients** option and list the coefficients you want to plot in the **Coefficient Display List** field of the dialog box. The recursive estimates of the marginal propensity to consume (coefficient $C(2)$), from the sample consumption function are provided below:

The estimated propensity to consume rises steadily as we add more data over the sample period, approaching a value of one.

Save Results as Series

The **Save Results as Series** checkbox will do different things depending on the plot you have asked to be displayed. When paired with the **Recursive Coefficients** option, **Save Results as Series** will instruct EViews to save all recursive coefficients and their standard errors in the workfile as named series. EViews will name the coefficients using the next available name of the form, R_C1 , R_C2 , ..., and the corresponding standard errors as R_C1SE , R_C2SE , and so on.



If you check the **Save Results as Series** box with any of the other options, EViews saves the recursive residuals and the recursive standard errors as named series in the workfile. EViews will name the residual and standard errors as R_RES and R_RESSE , respectively.

Note that you can use the recursive residuals to reconstruct the CUSUM and CUSUM of squares series.

Applications

For illustrative purposes, we provide a demonstration of how to carry out some other specification tests in EViews. For brevity, the discussion is based on commands, but most of these procedures can also be carried out using the menu system.

A Wald test of structural change with unequal variance

The F -statistics reported in the Chow tests have an F -distribution only if the errors are independent and identically normally distributed. This restriction implies that the residual variance in the two subsamples must be equal.

Suppose now that we wish to compute a Wald statistic for structural change with unequal subsample variances. Denote the parameter estimates and their covariance matrix in subsample i as b_i and V_i for $i = 1, 2$. Under the assumption that b_1 and b_2 are independent normal random variables, the difference $b_1 - b_2$ has mean zero and variance $V_1 + V_2$. Therefore, a Wald statistic for the null hypothesis of no structural change and independent samples can be constructed as:

$$W = (b_1 - b_2)'(V_1 + V_2)^{-1}(b_1 - b_2), \quad (28.38)$$

which has an asymptotic χ^2 distribution with degrees of freedom equal to the number of estimated parameters in the b vector.

To carry out this test in EViews, we estimate the model in each subsample and save the estimated coefficients and their covariance matrix. For example, suppose we have a quarterly sample of 1947:1–1994:4 and wish to test whether there was a structural change in the consumption function in 1973:1. First, estimate the model in the first sample and save the results by the commands:

```
coef(2) b1
smpl 1947:1 1972:4
equation eq_1.ls log(cs)=b1(1)+b1(2)*log(gdp)
sym v1=eq_1.@cov
```

The first line declares the coefficient vector, B1, into which we will place the coefficient estimates in the first sample. Note that the equation specification in the third line explicitly refers to elements of this coefficient vector. The last line saves the coefficient covariance matrix as a symmetric matrix named V1. Similarly, estimate the model in the second sample and save the results by the commands:

```
coef(2) b2
smpl 1973:1 1994:4
equation eq_2.ls log(cs)=b2(1)+b2(2)*log(gdp)
sym v2=eq_2.@cov
```

To compute the Wald statistic, use the command:

```
matrix wald=@transpose(b1-b2)*@inverse(v1+v2)*(b1-b2)
```

The Wald statistic is saved in the 1×1 matrix named WALD. To see the value, either double click on WALD or type “show wald”. You can compare this value with the critical values from the χ^2 distribution with 2 degrees of freedom. Alternatively, you can compute the p -value in EViews using the command:

```
scalar wald_p=1-@cchisq(wald(1,1),2)
```

The p -value is saved as a scalar named WALD_P. To see the p -value, double click on WALD_P or type “show wald_p”. The p -value will be displayed in the status line at the bottom of the EViews window.

The Hausman test

A widely used class of tests in econometrics is the Hausman test. The underlying idea of the Hausman test is to compare two sets of estimates, one of which is consistent under both the null and the alternative and another which is consistent only under the null hypothesis. A large difference between the two sets of estimates is taken as evidence in favor of the alternative hypothesis.

Hausman (1978) originally proposed a test statistic for endogeneity based upon a direct comparison of coefficient values. Here, we illustrate the version of the Hausman test proposed by Davidson and MacKinnon (1989, 1993), which carries out the test by running an auxiliary regression.

The following equation was estimated by OLS:

| Dependent Variable: LOG(M1) | | | | |
|--|-------------|-----------------------|-------------|--------|
| Method: Least Squares | | | | |
| Date: 08/13/97 Time: 14:12 | | | | |
| Sample(adjusted): 1959:02 1995:04 | | | | |
| Included observations: 435 after adjusting endpoints | | | | |
| Variable | Coefficient | Std. Error | t-Statistic | Prob. |
| C | -0.022699 | 0.004443 | -5.108528 | 0.0000 |
| LOG(IP) | 0.011630 | 0.002585 | 4.499708 | 0.0000 |
| DLOG(PPI) | -0.024886 | 0.042754 | -0.582071 | 0.5608 |
| TB3 | -0.000366 | 9.91E-05 | -3.692675 | 0.0003 |
| LOG(M1(-1)) | 0.996578 | 0.001210 | 823.4440 | 0.0000 |
| R-squared | 0.999953 | Mean dependent var | 5.844581 | |
| Adjusted R-squared | 0.999953 | S.D. dependent var | 0.670596 | |
| S.E. of regression | 0.004601 | Akaike info criterion | -7.913714 | |
| Sum squared resid | 0.009102 | Schwarz criterion | -7.866871 | |
| Log likelihood | 1726.233 | F-statistic | 2304897. | |
| Durbin-Watson stat | 1.265920 | Prob(F-statistic) | 0.000000 | |

Suppose we are concerned that industrial production (IP) is endogenously determined with money (M1) through the money supply function. If endogeneity is present, then OLS estimates will be biased and inconsistent. To test this hypothesis, we need to find a set of instrumental variables that are correlated with the “suspect” variable IP but not with the error term of the money demand equation. The choice of the appropriate instrument is a crucial step. Here, we take the unemployment rate (URATE) and Moody’s AAA corporate bond yield (AAA) as instruments.

To carry out the Hausman test by artificial regression, we run two OLS regressions. In the first regression, we regress the suspect variable (log) IP on all exogenous variables and instruments and retrieve the residuals:

```
ls log(ip) c dlog(ppi) tb3 log(m1(-1)) urate aaa
series res_ip=resid
```

Then in the second regression, we re-estimate the money demand function including the residuals from the first regression as additional regressors. The result is:

| Dependent Variable: LOG(M1) | | | | |
|--|-------------|-----------------------|-------------|-----------|
| Method: Least Squares | | | | |
| Date: 08/13/97 Time: 15:28 | | | | |
| Sample(adjusted): 1959:02 1995:04 | | | | |
| Included observations: 435 after adjusting endpoints | | | | |
| Variable | Coefficient | Std. Error | t-Statistic | Prob. |
| C | -0.007145 | 0.007473 | -0.956158 | 0.3395 |
| LOG(IP) | 0.001560 | 0.004672 | 0.333832 | 0.7387 |
| DLOG(PPI) | 0.020233 | 0.045935 | 0.440465 | 0.6598 |
| TB3 | -0.000185 | 0.000121 | -1.527775 | 0.1273 |
| LOG(M1(-1)) | 1.001093 | 0.002123 | 471.4894 | 0.0000 |
| RES_IP | 0.014428 | 0.005593 | 2.579826 | 0.0102 |
| R-squared | 0.999954 | Mean dependent var | | 5.844581 |
| Adjusted R-squared | 0.999954 | S.D. dependent var | | 0.670596 |
| S.E. of regression | 0.004571 | Akaike info criterion | | -7.924512 |
| Sum squared resid | 0.008963 | Schwarz criterion | | -7.868300 |
| Log likelihood | 1729.581 | F-statistic | | 1868171. |
| Durbin-Watson stat | 1.307838 | Prob(F-statistic) | | 0.000000 |

If the OLS estimates are consistent, then the coefficient on the first stage residuals should not be significantly different from zero. In this example, the test (marginally) rejects the hypothesis of consistent OLS estimates (to be more precise, this is an asymptotic test and you should compare the t -statistic with the critical values from the standard normal).

Non-nested Tests

Most of the tests discussed in this chapter are nested tests in which the null hypothesis is obtained as a special case of the alternative hypothesis. Now consider the problem of choosing between the following two specifications of a consumption function:

$$\begin{aligned}
H_1: \quad CS_t &= \alpha_1 + \alpha_2 GDP_t + \alpha_3 GDP_{t-1} + \epsilon_t \\
H_2: \quad CS_t &= \beta_1 + \beta_2 GDP_t + \beta_3 CS_{t-1} + \epsilon_t
\end{aligned}
\tag{28.39}$$

These are examples of non-nested models since neither model may be expressed as a restricted version of the other.

The J -test proposed by Davidson and MacKinnon (1993) provides one method of choosing between two non-nested models. The idea is that if one model is the correct model, then the fitted values from the other model should not have explanatory power when estimating that model. For example, to test model H_1 against model H_2 , we first estimate model H_2 and retrieve the fitted values:

```
equation eq_cs2.ls cs c gdp cs(-1)
eq_cs2.fit cs2
```

The second line saves the fitted values as a series named CS2. Then estimate model H_1 including the fitted values from model H_2 . The result is:

| Dependent Variable: CS | | | | |
|--|-------------|-----------------------|-------------|--------|
| Method: Least Squares | | | | |
| Date: 8/13/97 Time: 00:49 | | | | |
| Sample(adjusted): 1947:2 1994:4 | | | | |
| Included observations: 191 after adjusting endpoints | | | | |
| Variable | Coefficient | Std. Error | t-Statistic | Prob. |
| C | 7.313232 | 4.391305 | 1.665389 | 0.0975 |
| GDP | 0.278749 | 0.029278 | 9.520694 | 0.0000 |
| GDP(-1) | -0.314540 | 0.029287 | -10.73978 | 0.0000 |
| CS2 | 1.048470 | 0.019684 | 53.26506 | 0.0000 |
| R-squared | 0.999833 | Mean dependent var | 1953.966 | |
| Adjusted R-squared | 0.999830 | S.D. dependent var | 848.4387 | |
| S.E. of regression | 11.05357 | Akaike info criterion | 7.664104 | |
| Sum squared resid | 22847.93 | Schwarz criterion | 7.732215 | |
| Log likelihood | -727.9219 | F-statistic | 373074.4 | |
| Durbin-Watson stat | 2.253186 | Prob(F-statistic) | 0.000000 | |

The fitted values from model H_2 enter significantly in model H_1 and we reject model H_1 .

We must also test model H_2 against model H_1 . Estimate model H_1 , retrieve the fitted values, and estimate model H_2 including the fitted values from model H_1 . The results of this “reverse” test are given by:

Dependent Variable: CS
Method: Least Squares
Date: 08/13/97 Time: 16:58
Sample(adjusted): 1947:2 1994:4
Included observations: 191 after adjusting endpoints

| Variable | Coefficient | Std. Error | t-Statistic | Prob. |
|--------------------|-------------|-----------------------|-------------|----------|
| C | -1427.716 | 132.0349 | -10.81318 | 0.0000 |
| GDP | 5.170543 | 0.476803 | 10.84419 | 0.0000 |
| CS(-1) | 0.977296 | 0.018348 | 53.26506 | 0.0000 |
| CS1 | -7.292771 | 0.679043 | -10.73978 | 0.0000 |
| R-squared | 0.999833 | Mean dependent var | | 1953.966 |
| Adjusted R-squared | 0.999830 | S.D. dependent var | | 848.4387 |
| S.E. of regression | 11.05357 | Akaike info criterion | | 7.664104 |
| Sum squared resid | 22847.93 | Schwarz criterion | | 7.732215 |
| Log likelihood | -727.9219 | F-statistic | | 373074.4 |
| Durbin-Watson stat | 2.253186 | Prob(F-statistic) | | 0.000000 |

The fitted values are again statistically significant and we reject model H_2 .

In this example, we reject both specifications, against the alternatives, suggesting that another model for the data is needed. It is also possible that we fail to reject both models, in which case the data do not provide enough information to discriminate between the two models.

References

- Andrews, Donald W. K. (1993). "Tests for Parameter Instability and Structural Change With Unknown Change Point," *Econometrica*, 61(4), 821–856.
- Andrews, Donald W. K. and W. Ploberger (1994). "Optimal Tests When a Nuisance Parameter is Present Only Under the Alternative," *Econometrica*, 62(6), 1383–1414.
- Breusch, T. S., and A. R. Pagan (1979). "A Simple Test for Heteroskedasticity and Random Coefficient Variation," *Econometrica*, 48, 1287–1294.
- Brown, R. L., J. Durbin, and J. M. Evans (1975). "Techniques for Testing the Constancy of Regression Relationships Over Time," *Journal of the Royal Statistical Society, Series B*, 37, 149–192.
- Davidson, Russell and James G. MacKinnon (1989). "Testing for Consistency using Artificial Regressions," *Econometric Theory*, 5, 363–384.
- Davidson, Russell and James G. MacKinnon (1993). *Estimation and Inference in Econometrics*, Oxford: Oxford University Press.
- Engle, Robert F. (1982). "Autoregressive Conditional Heteroskedasticity with Estimates of the Variance of U.K. Inflation," *Econometrica*, 50, 987–1008.
- Glejser, H. (1969). "A New Test For Heteroscedasticity," *Journal of the American Statistical Association*, 64, 316–323.
- Godfrey, L. G. (1978). "Testing for Multiplicative Heteroscedasticity," *Journal of Econometrics*, 8, 227–236.
- Godfrey, L. G. (1988). *Specification Tests in Econometrics*, Cambridge: Cambridge University Press.

- Hansen, B. E. (1997). "Approximate Asymptotic P Values for Structural-Change Tests," *Journal of Business and Economic Statistics*, 15(1), 60–67.
- Harvey, Andrew C. (1976). "Estimating Regression Models with Multiplicative Heteroscedasticity," *Econometrica*, 44, 461–465.
- Hausman, Jerry A. (1978). "Specification Tests in Econometrics," *Econometrica*, 46, 1251–1272.
- Johnston, Jack and John Enrico DiNardo (1997). *Econometric Methods*, 4th Edition, New York: McGraw-Hill.
- Koenker, R. (1981). "A Note on Studentizing a Test for Heteroskedasticity," *Journal of Econometrics*, 17, 107–112.
- Ramsey, J. B. (1969). "Tests for Specification Errors in Classical Linear Least Squares Regression Analysis," *Journal of the Royal Statistical Society, Series B*, 31, 350–371.
- Ramsey, J. B. and A. Alexander (1984). "The Econometric Approach to Business-Cycle Analysis Reconsidered," *Journal of Macroeconomics*, 6, 347–356.
- White, Halbert (1980). "A Heteroskedasticity-Consistent Covariance Matrix and a Direct Test for Heteroskedasticity," *Econometrica*, 48, 817–838.
- Wooldridge, Jeffrey M. (1990). "Quasi-Likelihood Methods for Count Data," Chapter 8 in M. Hashem Pesaran and P. Schmidt (eds.) *Handbook of Applied Econometrics, Volume 2*, Malden, MA: Blackwell, 352–406.

Part V. Advanced Single Equation Analysis

The following sections describe EViews tools for the estimation and analysis of advanced single equation models.

- [Chapter 29. “ARCH and GARCH Estimation,” beginning on page 185](#), outlines the EViews tools for ARCH and GARCH modeling of the conditional variance, or volatility, of a variable.
- [Chapter 30. “Discrete and Limited Dependent Variable Models,” on page 209](#) documents EViews tools for estimating qualitative and limited dependent variable models. EViews provides estimation routines for binary or ordered (probit, logit, gompit), censored or truncated (tobit, etc.), and integer valued (count data).
- [Chapter 31. “Quantile Regression,” beginning on page 259](#) describes the estimation of quantile regression and least absolute deviations estimation in EViews.
- [Chapter 32. “The Log Likelihood \(LogL\) Object,” beginning on page 283](#) describes techniques for using EViews to estimate the parameters of maximum likelihood models where you may specify the form of the likelihood.

Multiple equation models and forecasting are described in [Part VI. “Multiple Equation Analysis,” beginning on page 305](#).

[Chapter 39. “Panel Estimation,” beginning on page 541](#) describes estimation in panel structured workfiles.

Chapter 29. ARCH and GARCH Estimation

Most of the statistical tools in EViews are designed to model the conditional mean of a random variable. The tools described in this chapter differ by modeling the conditional variance, or volatility, of a variable.

There are several reasons that you may wish to model and forecast volatility. First, you may need to analyze the risk of holding an asset or the value of an option. Second, forecast confidence intervals may be time-varying, so that more accurate intervals can be obtained by modeling the variance of the errors. Third, more efficient estimators can be obtained if heteroskedasticity in the errors is handled properly.

Autoregressive Conditional Heteroskedasticity (ARCH) models are specifically designed to model and forecast conditional variances. The variance of the dependent variable is modeled as a function of past values of the dependent variable and independent, or exogenous variables.

ARCH models were introduced by Engle (1982) and generalized as GARCH (Generalized ARCH) by Bollerslev (1986) and Taylor (1986). These models are widely used in various branches of econometrics, especially in financial time series analysis. See Bollerslev, Chou, and Kroner (1992) and Bollerslev, Engle, and Nelson (1994) for recent surveys.

In the next section, the basic ARCH model will be described in detail. In subsequent sections, we consider the wide range of specifications available in EViews for modeling volatility. For brevity of discussion, we will use ARCH to refer to both ARCH and GARCH models, except where there is the possibility of confusion.

Basic ARCH Specifications

In developing an ARCH model, you will have to provide three distinct specifications—one for the conditional mean equation, one for the conditional variance, and one for the conditional error distribution. We begin by describing some basic specifications for these terms. The discussion of more complicated models is taken up in [“Additional ARCH Models” on page 197](#).

The GARCH(1, 1) Model

We begin with the simplest GARCH(1,1) specification:

$$Y_t = X_t' \theta + \epsilon_t \quad (29.1)$$

$$\sigma_t^2 = \omega + \alpha \epsilon_{t-1}^2 + \beta \sigma_{t-1}^2 \quad (29.2)$$

in which the mean equation given in (29.1) is written as a function of exogenous variables with an error term. Since σ_t^2 is the one-period ahead forecast variance based on past infor-

mation, it is called the *conditional variance*. The conditional variance equation specified in (29.2) is a function of three terms:

- A constant term: ω .
- News about volatility from the previous period, measured as the lag of the squared residual from the mean equation: ϵ_{t-1}^2 (the ARCH term).
- Last period's forecast variance: σ_{t-1}^2 (the GARCH term).

The (1, 1) in GARCH(1, 1) refers to the presence of a first-order autoregressive GARCH term (the first term in parentheses) and a first-order moving average ARCH term (the second term in parentheses). An ordinary ARCH model is a special case of a GARCH specification in which there are no lagged forecast variances in the conditional variance equation—*i.e.*, a GARCH(0, 1).

This specification is often interpreted in a financial context, where an agent or trader predicts this period's variance by forming a weighted average of a long term average (the constant), the forecasted variance from last period (the GARCH term), and information about volatility observed in the previous period (the ARCH term). If the asset return was unexpectedly large in either the upward or the downward direction, then the trader will increase the estimate of the variance for the next period. This model is also consistent with the volatility clustering often seen in financial returns data, where large changes in returns are likely to be followed by further large changes.

There are two equivalent representations of the variance equation that may aid you in interpreting the model:

- If we recursively substitute for the lagged variance on the right-hand side of Equation (29.2), we can express the conditional variance as a weighted average of all of the lagged squared residuals:

$$\sigma_t^2 = \frac{\omega}{(1-\beta)} + \alpha \sum_{j=1}^{\infty} \beta^{j-1} \epsilon_{t-j}^2. \quad (29.3)$$

We see that the GARCH(1,1) variance specification is analogous to the sample variance, but that it down-weights more distant lagged squared errors.

- The error in the squared returns is given by $v_t = \epsilon_t^2 - \sigma_t^2$. Substituting for the variances in the variance equation and rearranging terms we can write our model in terms of the errors:

$$\epsilon_t^2 = \omega + (\alpha + \beta)\epsilon_{t-1}^2 + v_t - \beta v_{t-1}. \quad (29.4)$$

Thus, the squared errors follow a heteroskedastic ARMA(1,1) process. The autoregressive root which governs the persistence of volatility shocks is the sum of α plus β . In many applied settings, this root is very close to unity so that shocks die out rather slowly.

The GARCH(q, p) Model

Higher order GARCH models, denoted GARCH(q, p), can be estimated by choosing either q or p greater than 1 where q is the order of the autoregressive GARCH terms and p is the order of the moving average ARCH terms.

The representation of the GARCH(q, p) variance is:

$$\sigma_t^2 = \omega + \sum_{j=1}^q \beta_j \sigma_{t-j}^2 + \sum_{i=1}^p \alpha_i \epsilon_{t-i}^2 \quad (29.5)$$

The GARCH-M Model

The X_t in equation (29.2) represent exogenous or predetermined variables that are included in the mean equation. If we introduce the conditional variance or standard deviation into the mean equation, we get the GARCH-in-Mean (GARCH-M) model (Engle, Lilien and Robins, 1987):

$$Y_t = X_t' \theta + \lambda \sigma_t^2 + \epsilon_t. \quad (29.6)$$

The ARCH-M model is often used in financial applications where the expected return on an asset is related to the expected asset risk. The estimated coefficient on the expected risk is a measure of the risk-return tradeoff.

Two variants of this ARCH-M specification use the conditional standard deviation or the log of the conditional variance in place of the variance in Equation (29.6).

$$Y_t = X_t' \theta + \lambda \sigma_t + \epsilon_t. \quad (29.7)$$

$$Y_t = X_t' \theta + \lambda \log(\sigma_t^2) + \epsilon_t \quad (29.8)$$

Regressors in the Variance Equation

Equation (29.5) may be extended to allow for the inclusion of exogenous or predetermined regressors, z , in the variance equation:

$$\sigma_t^2 = \omega + \sum_{j=1}^q \beta_j \sigma_{t-j}^2 + \sum_{i=1}^p \alpha_i \epsilon_{t-i}^2 + Z_t' \pi. \quad (29.9)$$

Note that the forecasted variances from this model are not guaranteed to be positive. You may wish to introduce regressors in a form where they are always positive to minimize the possibility that a single, large negative value generates a negative forecasted value.

Distributional Assumptions

To complete the basic ARCH specification, we require an assumption about the conditional distribution of the error term ϵ . There are three assumptions commonly employed when working with ARCH models: normal (Gaussian) distribution, Student's t -distribution, and

the Generalized Error Distribution (GED). Given a distributional assumption, ARCH models are typically estimated by the method of maximum likelihood.

For example, for the GARCH(1, 1) model with conditionally normal errors, the contribution to the log-likelihood for observation t is:

$$l_t = -\frac{1}{2}\log(2\pi) - \frac{1}{2}\log\sigma_t^2 - \frac{1}{2}(y_t - X_t'\theta)^2 / \sigma_t^2, \quad (29.10)$$

where σ_t^2 is specified in one of the forms above.

For the Student's t -distribution, the log-likelihood contributions are of the form:

$$l_t = -\frac{1}{2}\log\left(\frac{\pi(\nu-2)\Gamma(\nu/2)^2}{\Gamma((\nu+1)/2)^2}\right) - \frac{1}{2}\log\sigma_t^2 - \frac{(\nu+1)}{2}\log\left(1 + \frac{(y_t - X_t'\theta)^2}{\sigma_t^2(\nu-2)}\right) \quad (29.11)$$

where the degree of freedom $\nu > 2$ controls the tail behavior. The t -distribution approaches the normal as $\nu \rightarrow \infty$.

For the GED, we have:

$$l_t = -\frac{1}{2}\log\left(\frac{\Gamma(1/r)^3}{\Gamma(3/r)(r/2)^2}\right) - \frac{1}{2}\log\sigma_t^2 - \left(\frac{\Gamma(3/r)(y_t - X_t'\theta)^2}{\sigma_t^2\Gamma(1/r)}\right)^{r/2} \quad (29.12)$$

where the tail parameter $r > 0$. The GED is a normal distribution if $r = 2$, and fat-tailed if $r < 2$.

By default, ARCH models in EViews are estimated by the method of maximum likelihood under the assumption that the errors are conditionally normally distributed.

Estimating ARCH Models in EViews

To estimate an ARCH or GARCH model, open the equation specification dialog by selecting **Quick/Estimate Equation...** or by selecting **Object/New Object.../Equation...**. Select **ARCH** from the method combo box at the bottom of the dialog.

The dialog will change to show you the ARCH specification dialog. You will need to specify both the mean and the variance specifications, the error distribution and the estimation sample.

The Mean Equation

In the dependent variable edit box, you should enter the specification of the mean equation. You can enter the specification in list form by listing the dependent variable followed by the regressors. You should add the C to your specification if you wish to include a constant. If you have a more complex mean specification, you can enter your mean equation using an explicit expression.

If your specification includes an ARCH-M term, you should select the appropriate item of the combo box in the upper right-hand side of the dialog.

The Variance Equation

Your next step is to specify your variance equation.

Class of models

To estimate one of the standard GARCH models as described above, select the **GARCH/TARCH** entry from the **Model** combo box. The other entries (**EGARCH**, **PARCH**, and **Component ARCH(1, 1)**) correspond to more complicated variants of the GARCH specification. We discuss each of these models in [“Additional ARCH Models” on page 197](#).

Under the **Options** label, you should choose the number of ARCH and GARCH terms. The default, which includes one ARCH and one GARCH term is by far the most popular specification.

If you wish to estimate an asymmetric model, you should enter the number of asymmetry terms in the **Threshold order** edit field. The default settings estimate a symmetric model.

Variance regressors

In the **Variance Regressors** edit box, you may optionally list variables you wish to include in the variance specification. Note that, with the exception of IGARCH models, EViews will

always include a constant as a variance regressor so that you do not need to add C to the list.

The distinction between the permanent and transitory regressors is discussed in [“The Component GARCH \(CGARCH\) Model” on page 200](#).

Restrictions

If you choose the GARCH/TARCH model, you may restrict the parameters of the GARCH model in two ways. One option is **IGARCH**, which restricts the persistent parameters to sum up to one. Another is **Variance Target**, which restricts the constant term to a function of the GARCH parameters and the unconditional variance:

$$\omega = \hat{\sigma}^2 \left(1 - \sum_{j=1}^q \beta_j - \sum_{i=1}^p \alpha_i \right) \quad (29.13)$$

where $\hat{\sigma}^2$ is the unconditional variance of the residuals.

The Error Distribution

To specify the form of the conditional distribution for your errors, you should select an entry from the combo box labeled **Error Distribution**. You may choose between the default **Normal (Gaussian)**, the **Student’s t**, the **Generalized Error (GED)**, the **Student’s t with fixed d.f.**, or the **GED with fixed parameter**. In the latter two cases, you will be prompted to enter a value for the fixed parameter. See [“Distributional Assumptions” on page 187](#) for details on the supported distributions.

Estimation Options

EViews provides you with access to a number of optional estimation settings. Simply click on the **Options** tab and fill out the dialog as desired.

Backcasting

By default, both the innovations used in initializing MA estimation and the initial variance required for the GARCH terms are computed using backcasting methods. Details on the MA backcasting procedure are provided in “Backcasting MA terms” on page 80.

When computing backcast initial variances for GARCH, EViews first uses the coefficient values to compute the residuals of the mean equation, and then computes an exponential smoothing estimator of the initial values,

$$\sigma_0^2 = \epsilon_0^2 = \lambda^T \hat{\sigma}^2 + (1 - \lambda) \sum_{j=0}^T \lambda^{T-j-1} (\hat{\epsilon}_{T-j}^2), \quad (29.14)$$

where $\hat{\epsilon}$ are the residuals from the mean equation, $\hat{\sigma}^2$ is the unconditional variance estimate:

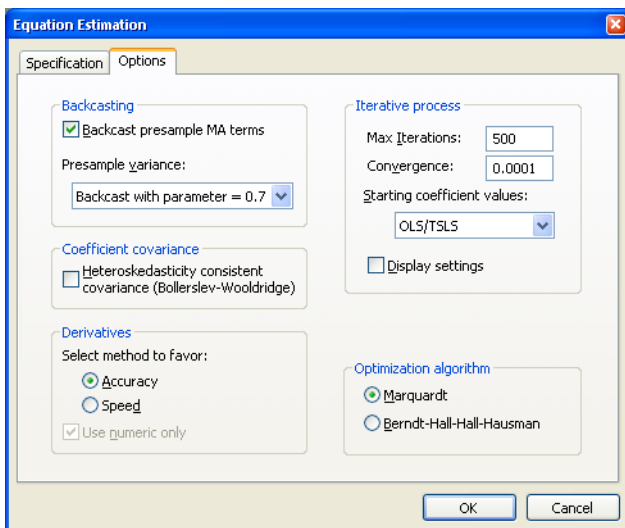
$$\hat{\sigma}^2 = \sum_{t=1}^T \hat{\epsilon}_t^2 / T \quad (29.15)$$

and the smoothing parameter $\lambda = 0.7$. However, you have the option to choose from a number of weights from 0.1 to 1, in increments of 0.1, by using the **Presample variance** drop-down list. Notice that if the parameter is set to 1, then the initial value is simply the unconditional variance, e.g. backcasting is not calculated:

$$\sigma_0^2 = \hat{\sigma}^2. \quad (29.16)$$

Using the unconditional variance provides another common way to set the presample variance.

Our experience has been that GARCH models initialized using backcast exponential smoothing often outperform models initialized using the unconditional variance.



Heteroskedasticity Consistent Covariances

Click on the check box labeled **Heteroskedasticity Consistent Covariance** to compute the quasi-maximum likelihood (QML) covariances and standard errors using the methods described by Bollerslev and Wooldridge (1992). This option is only available if you choose the conditional normal as the error distribution.

You should use this option if you suspect that the residuals are not conditionally normally distributed. When the assumption of conditional normality does not hold, the ARCH parameter estimates will still be consistent, provided the mean and variance functions are correctly specified. The estimates of the covariance matrix will not be consistent unless this option is specified, resulting in incorrect standard errors.

Note that the parameter estimates will be unchanged if you select this option; only the estimated covariance matrix will be altered.

Derivative Methods

EViews uses both numeric and analytic derivatives in estimating ARCH models. Fully analytic derivatives are available for GARCH(p,q) models with simple mean specifications assuming normal or unrestricted *t*-distribution errors.

Analytic derivatives are not available for models with ARCH in mean specifications, complex variance equation specifications (*e.g.* threshold terms, exogenous variance regressors, or integrated or target variance restrictions), models with certain error assumptions (*e.g.* errors following the GED or fixed parameter *t*-distributions), and all non-GARCH(p,q) models (*e.g.* EGARCH, PARCH, component GARCH).

Some specifications offer analytic derivatives for a subset of coefficients. For example, simple GARCH models with non-constant regressors allow for analytic derivatives for the variance coefficients but use numeric derivatives for any non-constant regressor coefficients.

You may control the method used in computing numeric derivatives to favor speed (fewer function evaluations) or to favor accuracy (more function evaluations).

Iterative Estimation Control

The likelihood functions of ARCH models are not always well-behaved so that convergence may not be achieved with the default estimation settings. You can use the options dialog to select the iterative algorithm (Marquardt, BHHH/Gauss-Newton), change starting values, increase the maximum number of iterations, or adjust the convergence criterion.

Starting Values

As with other iterative procedures, starting coefficient values are required. EViews will supply its own starting values for ARCH procedures using OLS regression for the mean equation. Using the **Options** dialog, you can also set starting values to various fractions of the

OLS starting values, or you can specify the values yourself by choosing the **User Specified** option, and placing the desired coefficients in the default coefficient vector.

GARCH(1,1) examples

To estimate a standard GARCH(1,1) model with no regressors in the mean and variance equations:

$$\begin{aligned} R_t &= c + \epsilon_t \\ \sigma_t^2 &= \omega + \alpha \epsilon_{t-1}^2 + \beta \sigma_{t-1}^2 \end{aligned} \quad (29.17)$$

you should enter the various parts of your specification:

- Fill in the **Mean Equation Specification** edit box as
r c
- Enter 1 for the number of ARCH terms, and 1 for the number of GARCH terms, and select **GARCH/TARCH**.
- Select **None** for the **ARCH-M term**.
- Leave blank the **Variance Regressors** edit box.

To estimate the ARCH(4)-M model:

$$\begin{aligned} R_t &= \gamma_0 + \gamma_1 DUM_t + \gamma_2 \sigma_t + \epsilon_t \\ \sigma_t^2 &= \omega + \alpha_1 \epsilon_{t-1}^2 + \alpha_2 \epsilon_{t-2}^2 + \alpha_3 \epsilon_{t-3}^2 + \alpha_4 \epsilon_{t-4}^2 + \gamma_3 DUM_t \end{aligned} \quad (29.18)$$

you should fill out the dialog in the following fashion:

- Enter the mean equation specification “R C DUM”.
- Enter “4” for the ARCH term and “0” for the GARCH term, and select **GARCH (sym-metric)**.
- Select **Std. Dev.** for the **ARCH-M term**.
- Enter DUM in the **Variance Regressors** edit box.

Once you have filled in the **Equation Specification** dialog, click **OK** to estimate the model. ARCH models are estimated by the method of maximum likelihood, under the assumption that the errors are conditionally normally distributed. Because the variance appears in a non-linear way in the likelihood function, the likelihood function must be estimated using iterative algorithms. In the status line, you can watch the value of the likelihood as it changes with each iteration. When estimates converge, the parameter estimates and conventional regression statistics are presented in the ARCH object window.

As an example, we fit a GARCH(1,1) model to the first difference of log daily S&P 500 (DLOG(SPX)) using backcast values for the initial variances and Bollerslev-Wooldridge standard errors. The output is presented below:

| Dependent Variable: DLOG(SPX) | | | | |
|---|-------------|-----------------------|-------------|--------|
| Method: ARCH - Normal distribution (Marquardt) | | | | |
| Date: 09/16/03 Time: 13:52 | | | | |
| Sample: 1/02/1990 12/31/1999 | | | | |
| Included observations: 2528 | | | | |
| Convergence achieved after 12 iterations | | | | |
| Bollerslev-Wooldridge robust standard errors & covariance | | | | |
| Variance backcast: ON | | | | |
| GARCH = C(2) + C(3)*RESID(-1)^2 + C(4)*GARCH(-1) | | | | |
| | Coefficient | Std. Error | z-Statistic | Prob. |
| C | 0.000598 | 0.000143 | 4.176757 | 0.0000 |
| Variance Equation | | | | |
| C | 5.82E-07 | 1.93E-07 | 3.018858 | 0.0025 |
| RESID(-1)^2 | 0.053337 | 0.011688 | 4.563462 | 0.0000 |
| GARCH(-1) | 0.939945 | 0.011201 | 83.91679 | 0.0000 |
| R-squared | -0.000015 | Mean dependent var | 0.000564 | |
| Adjusted R-squared | -0.001204 | S.D. dependent var | 0.008888 | |
| S.E. of regression | 0.008894 | Akaike info criterion | -6.807476 | |
| Sum squared resid | 0.199649 | Schwarz criterion | -6.798243 | |
| Log likelihood | 8608.650 | Durbin-Watson stat | 1.964028 | |

By default, the estimation output header describes the estimation sample, and the methods used for computing the coefficient standard errors, the initial variance terms, and the variance equation.

The main output from ARCH estimation is divided into two sections—the upper part provides the standard output for the mean equation, while the lower part, labeled “Variance Equation”, contains the coefficients, standard errors, z -statistics and p -values for the coefficients of the variance equation.

The ARCH parameters correspond to α and the GARCH parameters to β in Equation (29.2) on page 185. The bottom panel of the output presents the standard set of regression statistics using the residuals from the mean equation. Note that measures such as R^2 may not be meaningful if there are no regressors in the mean equation. Here, for example, the R^2 is negative.

In this example, the sum of the ARCH and GARCH coefficients ($\alpha + \beta$) is very close to one, indicating that volatility shocks are quite persistent. This result is often observed in high frequency financial data.

Working with ARCH Models

Once your model has been estimated, EViews provides a variety of views and procedures for inference and diagnostic checking.

Views of ARCH Models

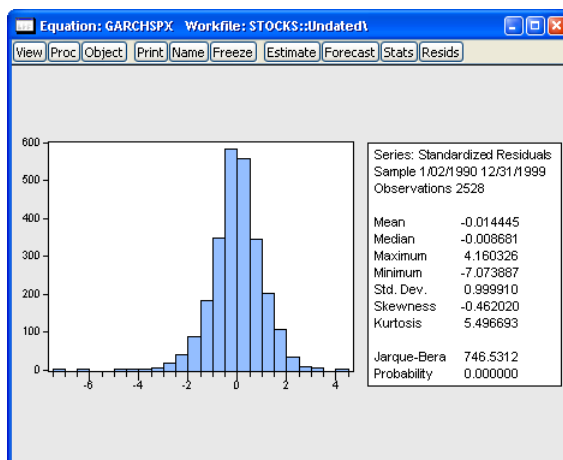
- **Actual, Fitted, Residual** view displays the residuals in various forms, such as table, graphs, and standardized residuals. You can save the residuals as a named series in your workfile using a procedure (see below).
- **GARCH Graph/Conditional Standard Deviation** or **GARCH Graph/Conditional Variance** plots the one-step ahead standard deviation σ_t or variance σ_t^2 for each observation in the sample. The observation at period t is the forecast for t made using information available in $t - 1$. You can save the conditional standard deviations or variances as named series in your workfile using a procedure (see below). If the specification is for a component model, EViews will also display the permanent and transitory components.
- **Covariance Matrix** displays the estimated coefficient covariance matrix. Most ARCH models (except ARCH-M models) are block diagonal so that the covariance between the mean coefficients and the variance coefficients is very close to zero. If you include a constant in the mean equation, there will be two C's in the covariance matrix; the first C is the constant of the mean equation, and the second C is the constant of the variance equation.
- **Coefficient Tests** carries out standard hypothesis tests on the estimated coefficients. See [“Coefficient Tests” on page 142](#) for details. Note that the likelihood ratio tests are not appropriate under a quasi-maximum likelihood interpretation of your results.
- **Residual Tests/Correlogram–Q-statistics** displays the correlogram (autocorrelations and partial autocorrelations) of the standardized residuals. This view can be used to test for remaining serial correlation in the mean equation and to check the specification of the mean equation. If the mean equation is correctly specified, all Q -statistics should not be significant. See [“Correlogram” on page 324](#) of the *User's Guide I* for an explanation of correlograms and Q -statistics.
- **Residual Tests/Correlogram Squared Residuals** displays the correlogram (autocorrelations and partial autocorrelations) of the squared standardized residuals. This view can be used to test for remaining ARCH in the variance equation and to check the specification of the variance equation. If the variance equation is correctly specified, all Q -statistics should not be significant. See [“Correlogram” on page 324](#) of the *User's*

Guide I for an explanation of correlograms and Q -statistics. See also **Residual Tests/ARCH LM Test**.

- **Residual Tests/Histogram–Normality Test** displays descriptive statistics and a histogram of the standardized residuals. You can use the Jarque-Bera statistic to test the null of whether the standardized residuals are normally distributed. If the standardized residuals are normally distributed, the Jarque-Bera statistic should not be significant. See “[Descriptive Statistics & Tests](#),” beginning on page 306 of the *User’s Guide I* for an explanation of the Jarque-Bera test. For example, the histogram of the standardized residuals from the GARCH(1,1) model fit to the daily stock return looks as follows:

The standardized residuals are leptokurtic and the Jarque-Bera statistic strongly rejects the hypothesis of normal distribution.

- **Residual Tests/ARCH LM Test** carries out Lagrange multiplier tests to test whether the standardized residuals exhibit additional ARCH. If the variance equation is correctly specified, there should be no ARCH left in the standardized residuals.



See “[ARCH LM Test](#)” on page 158 for a discussion of testing. See also **Residual Tests/Correlogram Squared Residuals**.

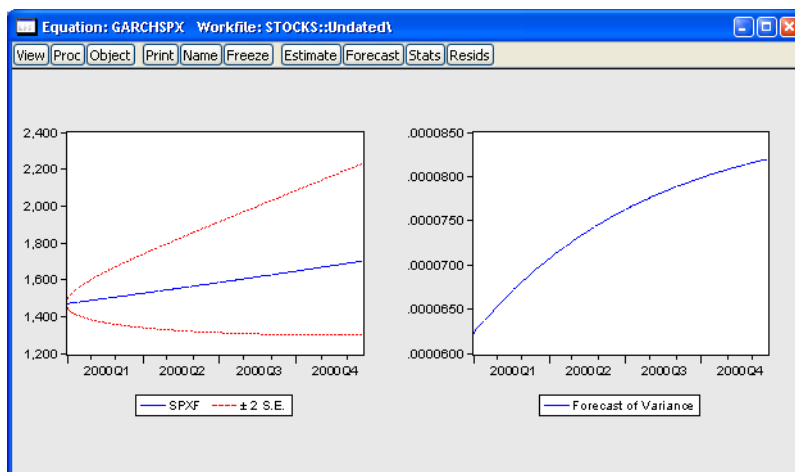
ARCH Model Procedures

- **Make Residual Series** saves the residuals as named series in your workfile. You have the option to save the ordinary residuals, ϵ_t , or the standardized residuals, ϵ_t/σ_t . The residuals will be named RESID1, RESID2, and so on; you can rename the series with the **name** button in the series window.
- **Make GARCH Variance Series...** saves the conditional variances σ_t^2 as named series in your workfile. You should provide a name for the target conditional variance series and, if relevant, you may provide a name for the permanent component series. You may take the square root of the conditional variance series to get the conditional standard deviations as displayed by the **View/GARCH Graph/Conditional Standard Deviation**.

- **Forecast** uses the estimated ARCH model to compute static and dynamic forecasts of the mean, its forecast standard error, and the conditional variance. To save any of these forecasts in your workfile, type a name in the corresponding dialog box. If you choose the **Do graph** option, EViews displays the graphs of the forecasts and two standard deviation bands for the mean forecast.

Note that the squared residuals ϵ_t^2 may not be available for presample values or when computing dynamic forecasts. In such cases, EViews will replace the term by its expected value. In the simple GARCH(p, q) case, for example, the expected value of the squared residual is the fitted variance, *e.g.*, $E(\epsilon_t^2) = \sigma_t^2$. In other models, the expected value of the residual term will differ depending on the distribution and, in some cases, the estimated parameters of the model.

For example, to construct dynamic forecasts of SPX using the previously estimated model, click on **Forecast** and fill in the **Forecast** dialog, setting the sample after the estimation period. If you choose **Do graph**, the equation view changes to display the forecast results. Here, we compute the forecasts from Jan. 1, 2000 to Jan. 1, 2001, and display them side-by-side.



The first graph is the forecast of SPX (SPXF) from the mean equation with two standard deviation bands. The second graph is the forecast of the conditional variance σ_t^2 .

Additional ARCH Models

In addition to the standard GARCH specification, EViews has the flexibility to estimate several other variance models. These include IGARCH, TARCH, EGARCH, PARCH, and component GARCH. For each of these models, the user has the ability to choose the order, if any, of asymmetry.

The Integrated GARCH (IGARCH) Model

If one restricts the parameters of the GARCH model to sum to one and drop the constant term

$$\sigma_t^2 = \sum_{j=1}^q \beta_j \sigma_{t-j}^2 + \sum_{i=1}^p \alpha_i \epsilon_{t-i}^2 \quad (29.19)$$

such that

$$\sum_{j=1}^q \beta_j + \sum_{i=1}^p \alpha_i = 1 \quad (29.20)$$

then we have an integrated GARCH. This model was originally described in Engle and Bollerslev (1986). To estimate this model, select **IGARCH** in the **Restrictions** drop-down menu for the GARCH/TARCH model.

The Threshold GARCH (TARCH) Model

TARCH or Threshold ARCH and Threshold GARCH were introduced independently by Zakoian (1994) and Glosten, Jaganathan, and Runkle (1993). The generalized specification for the conditional variance is given by:

$$\sigma_t^2 = \omega + \sum_{j=1}^q \beta_j \sigma_{t-j}^2 + \sum_{i=1}^p \alpha_i \epsilon_{t-i}^2 + \sum_{k=1}^r \gamma_k \epsilon_{t-k}^2 I_{t-k}^- \quad (29.21)$$

where $I_t^- = 1$ if $\epsilon_t < 0$ and 0 otherwise.

In this model, good news, $\epsilon_{t-i} > 0$, and bad news, $\epsilon_{t-i} < 0$, have differential effects on the conditional variance; good news has an impact of α_i , while bad news has an impact of $\alpha_i + \gamma_i$. If $\gamma_i > 0$, bad news increases volatility, and we say that there is a *leverage effect* for the i -th order. If $\gamma_i \neq 0$, the news impact is asymmetric.

Note that GARCH is a special case of the TARCH model where the threshold term is set to zero. To estimate a TARCH model, specify your GARCH model with ARCH and GARCH order and then change the **Threshold order** to the desired value.

The Exponential GARCH (EGARCH) Model

The EGARCH or Exponential GARCH model was proposed by Nelson (1991). The specification for the conditional variance is:

$$\log(\sigma_t^2) = \omega + \sum_{j=1}^q \beta_j \log(\sigma_{t-j}^2) + \sum_{i=1}^p \alpha_i \left| \frac{\epsilon_{t-i}}{\sigma_{t-i}} \right| + \sum_{k=1}^r \gamma_k \frac{\epsilon_{t-k}}{\sigma_{t-k}}. \quad (29.22)$$

Note that the left-hand side is the *log* of the conditional variance. This implies that the leverage effect is exponential, rather than quadratic, and that forecasts of the conditional vari-

ance are guaranteed to be nonnegative. The presence of leverage effects can be tested by the hypothesis that $\gamma_i < 0$. The impact is asymmetric if $\gamma_i \neq 0$.

There are two differences between the EViews specification of the EGARCH model and the original Nelson model. First, Nelson assumes that the ϵ_t follows a Generalized Error Distribution (GED), while EViews gives you a choice of normal, Student's t -distribution, or GED. Second, Nelson's specification for the log conditional variance is a restricted version of:

$$\log(\sigma_t^2) = \omega + \sum_{j=1}^q \beta_j \log(\sigma_{t-j}^2) + \sum_{i=1}^p \alpha_i \left| \frac{\epsilon_{t-i}}{\sigma_{t-i}} - E\left(\frac{\epsilon_{t-i}}{\sigma_{t-i}}\right) \right| + \sum_{k=1}^r \gamma_k \frac{\epsilon_{t-k}}{\sigma_{t-k}}$$

which differs slightly from the specification above. Estimating this model will yield identical estimates to those reported by EViews except for the intercept term w , which will differ in a manner that depends upon the distributional assumption and the order p . For example, in a $p = 1$ model with a normal distribution, the difference will be $\alpha_1 \sqrt{2/\pi}$.

To estimate an EGARCH model, simply select the **EGARCH** in the model specification combo box and enter the orders for the **ARCH**, **GARCH** and the **Asymmetry order**.

The screenshot shows the 'Equation Estimation' dialog box in EViews. The 'Specification' tab is active. Under 'Mean equation', the dependent variable is 'dlog(ibm)' and the equation is 'dlog(ibm)=c(1)+c(2)*dlog(spx)'. The 'Variance and distribution specification' section shows the 'Model' set to 'EGARCH'. The 'Order' section has 'ARCH' set to 1, 'Asymmetric' checked, and 'GARCH' set to 1. The 'Error distribution' is set to 'Generalized Error (GED)'. The 'Estimation settings' section shows the 'Method' set to 'ARCH - Autoregressive Conditional Heteroskedasticity' and the 'Sample' set to '1/01/1990 12/31/1999'.

The Power ARCH (PARCH) Model

Taylor (1986) and Schwert (1989) introduced the standard deviation GARCH model, where the standard deviation is modeled rather than the variance. This model, along with several other models, is generalized in Ding *et al.* (1993) with the Power ARCH specification. In the Power ARCH model, the power parameter δ of the standard deviation can be estimated rather than imposed, and the optional γ parameters are added to capture asymmetry of up to order r :

$$\sigma_t^\delta = \omega + \sum_{j=1}^q \beta_j \sigma_{t-j}^\delta + \sum_{i=1}^p \alpha_i (|\epsilon_{t-i}| - \gamma_i \epsilon_{t-i})^\delta \quad (29.23)$$

where $\delta > 0$, $|\gamma_i| \leq 1$ for $i = 1, \dots, r$, $\gamma_i = 0$ for all $i > r$, and $r \leq p$.

The symmetric model sets $\gamma_i = 0$ for all i . Note that if $\delta = 2$ and $\gamma_i = 0$ for all i , the PARCH model is simply a standard GARCH specification. As in the previous models, the asymmetric effects are present if $\gamma \neq 0$.

To estimate this model, simply select the PARCH in the model specification combo box and input the orders for the **ARCH**, **GARCH** and **Asymmetric** terms. EVIEWS provides you with the option of either estimating or fixing a value for δ . To estimate the Taylor-Schwert's model, for example, you will to set the order of the asymmetric terms to zero and will set δ to 1.

The Component GARCH (CGARCH) Model

The conditional variance in the GARCH(1, 1) model:

$$\sigma_t^2 = \bar{\omega} + \alpha(\epsilon_{t-1}^2 - \bar{\omega}) + \beta(\sigma_{t-1}^2 - \bar{\omega}). \quad (29.24)$$

shows mean reversion to $\bar{\omega}$, which is a constant for all time. By contrast, the component model allows mean reversion to a varying level m_t , modeled as:

$$\begin{aligned} \sigma_t^2 - m_t &= \alpha(\epsilon_{t-1}^2 - m_{t-1}) + \beta(\sigma_{t-1}^2 - m_{t-1}) \\ m_t &= \omega + \rho(m_{t-1} - \omega) + \phi(\epsilon_{t-1}^2 - \sigma_{t-1}^2). \end{aligned} \quad (29.25)$$

Here σ_t^2 is still the volatility, while m_t takes the place of ω and is the time varying long-run volatility. The first equation describes the transitory component, $\sigma_t^2 - m_t$, which converges to zero with powers of $(\alpha + \beta)$. The second equation describes the long run component m_t , which converges to ω with powers of ρ . ρ is typically between 0.99 and 1 so that m_t approaches ω very slowly. We can combine the transitory and permanent equations and write:

$$\begin{aligned}\sigma_t^2 = & (1 - \alpha - \beta)(1 - \rho)\omega + (\alpha + \phi)\epsilon_{t-1}^2 - (\alpha\rho + (\alpha + \beta)\phi)\epsilon_{t-2}^2 \\ & + (\beta - \phi)\sigma_{t-1}^2 - (\beta\rho - (\alpha + \beta)\phi)\sigma_{t-2}^2\end{aligned}\quad (29.26)$$

which shows that the component model is a (nonlinear) restricted GARCH(2, 2) model.

To select the Component ARCH model, simply choose **Component ARCH(1,1)** in the **Model** combo box. You can include exogenous variables in the conditional variance equation of component models, either in the permanent or transitory equation (or both). The variables in the transitory equation will have an impact on the short run movements in volatility, while the variables in the permanent equation will affect the long run levels of volatility.

An asymmetric Component ARCH model may be estimated by checking the **Include threshold term** checkbox. This option combines the component model with the asymmetric TARARCH model, introducing asymmetric effects in the transitory equation and estimates models of the form:

$$\begin{aligned}y_t &= x_t' \pi + \epsilon_t \\ m_t &= \omega + \rho(m_{t-1} - \omega) + \phi(\epsilon_{t-1}^2 - \sigma_{t-1}^2) + \theta_1 z_{1t} \\ \sigma_t^2 - m_t &= \alpha(\epsilon_{t-1}^2 - m_{t-1}) + \gamma(\epsilon_{t-1}^2 - m_{t-1})d_{t-1} + \beta(\sigma_{t-1}^2 - m_{t-1}) + \theta_2 z_{2t}\end{aligned}\quad (29.27)$$

where z are the exogenous variables and d is the dummy variable indicating negative shocks. $\gamma > 0$ indicates the presence of transitory leverage effects in the conditional variance.

Equation Estimation

Specification Options

Mean equation
Dependent followed by regressors and ARMA terms OR explicit
dlog(ibm)=c(1)+c(2)*dlog(spx) ARCH-M: Std. Dev.

Variance and distribution specification
Model: Component ARCH(1,1)
Order: 1
Include threshold term: ☐
Variance regressors: (enter components as "permanent @ transitory")
@ d(tbond)
Error distribution: Generalized Error (GED)

Estimation settings
Method: ARCH - Autoregressive Conditional Heteroskedasticity
Sample: 1/01/1990 12/31/1999

OK Cancel

User Specified Models

In some cases, you might wish to estimate an ARCH model not mentioned above, for example a special variant of PARCH. Many other ARCH models can be estimated using the `logl` object. For example, [Chapter 32. “The Log Likelihood \(LogL\) Object,” beginning on page 283](#) contains examples of using `logl` objects for simple bivariate GARCH models.

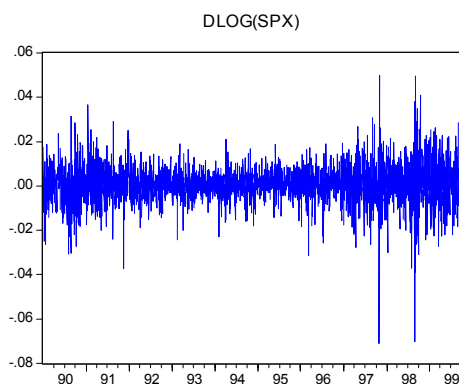
Examples

As an illustration of ARCH modeling in EViews, we estimate a model for the daily S&P 500 stock index from 1990 to 1999. The dependent variable is the daily continuously compounding return, $\log(s_t/s_{t-1})$, where s_t is the daily close of the index. A graph of the return series clearly shows volatility clustering.

We will specify our mean equation with a simple constant:

$$\log(s_t/s_{t-1}) = c_1 + \epsilon_t \quad (29.28)$$

For the variance specification, we employ an EGARCH(1, 1) model:



$$\log(\sigma_t^2) = \omega + \beta \log(\sigma_{t-1}^2) + \alpha \left| \frac{\epsilon_{t-1}}{\sigma_{t-1}} \right| + \gamma \frac{\epsilon_{t-1}}{\sigma_{t-1}} \quad (29.29)$$

When we previously estimated a GARCH(1,1) model with the data, the standardized residual showed evidence of excess kurtosis. To model the thick tail in the residuals, we will assume that the errors follow a Student's t -distribution.

To estimate this model, open the GARCH estimation dialog, enter the mean specification:

```
dlog(spx) c
```

select the **EGARCH** method, enter 1 for the **ARCH** and **GARCH** orders and the **Asymmetric order**, and select **Student's t** for the **Error distribution**. Click on **OK** to continue.

EViews displays the results of the estimation procedure. The top portion contains a description of the estimation specification, including the estimation sample, error distribution assumption, and backcast assumption.

Below the header information are the results for the mean and the variance equations, followed by the results for any distributional parameters. Here, we see that the relatively small degrees of freedom parameter for the t -distribution suggests that the distribution of the standardized errors departs significantly from normality.

Method: ARCH - Student's t distribution (Marquardt)

Date: 09/16/03 Time: 13:52

Sample: 1/02/1990 12/31/1999

Included observations: 2528

Convergence achieved after 18 iterations

Variance backcast: ON

LOG(GARCH) = C(2) + C(3)*ABS(RESID(-1)/@SQRT(GARCH(-1))) +

C(4)*RESID(-1)/@SQRT(GARCH(-1)) + C(5)*LOG(GARCH(-1))

| | Coefficient | Std. Error | z-Statistic | Prob. |
|--------------------|-------------|-----------------------|-------------|--------|
| C | 0.000513 | 0.000135 | 3.810316 | 0.0001 |
| Variance Equation | | | | |
| C(2) | -0.196806 | 0.039216 | -5.018500 | 0.0000 |
| C(3) | 0.113679 | 0.017573 | 6.468881 | 0.0000 |
| C(4) | -0.064136 | 0.011584 | -5.536811 | 0.0000 |
| C(5) | 0.988574 | 0.003366 | 293.7065 | 0.0000 |
| T-DIST. DOF | 6.703937 | 0.844864 | 7.934935 | 0.0000 |
| R-squared | -0.000032 | Mean dependent var | 0.000564 | |
| Adjusted R-squared | -0.002015 | S.D. dependent var | 0.008888 | |
| S.E. of regression | 0.008897 | Akaike info criterion | -6.871798 | |
| Sum squared resid | 0.199653 | Schwarz criterion | -6.857949 | |
| Log likelihood | 8691.953 | Durbin-Watson stat | 1.963994 | |

To test whether there any remaining ARCH effects in the residuals, select **View/Residual Tests/ARCH LM Test...** and specify the order to test. The top portion of the output from testing up-to an ARCH(7) is given by:

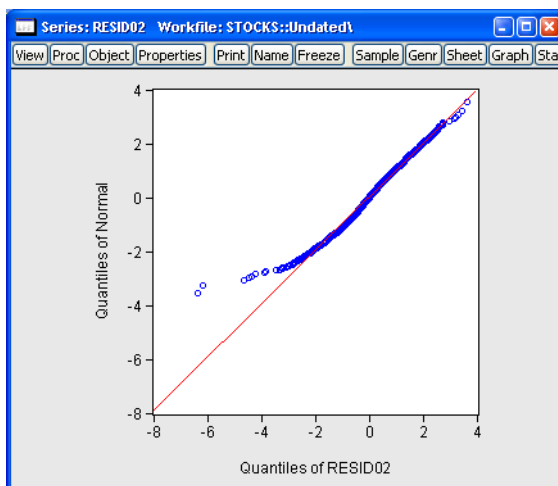
ARCH Test:

| | | | |
|---------------|----------|-------------|----------|
| F-statistic | 0.398638 | Probability | 0.903581 |
| Obs*R-squared | 2.796245 | Probability | 0.903190 |

so there is little evidence of remaining ARCH effects.

One way of further examining the distribution of the residuals is to plot the quantiles. First, save the standardized residuals by clicking on **Proc/Make Residual Series...**, select the **Standardized** option, and specify a name for the resulting series. EViews will create a series containing the desired residuals; in this example, we create a series named RESID02. Then open the residual series window and select **View/Graph...** and **Quantile-Quantile** from the list of graph types on the left-hand side of the dialog.

If the residuals are normally distributed, the points in the QQ-plots should lie alongside a straight line; see “[Quantile-Quantile \(Theoretical\)](#)” on page 475 of the *User’s Guide I* for details on QQ-plots. The plot indicates that it is primarily large negative shocks that are driving the departure from normality. Note that we have modified the QQ-plot slightly by setting identical axes to facilitate comparison with the diagonal line.



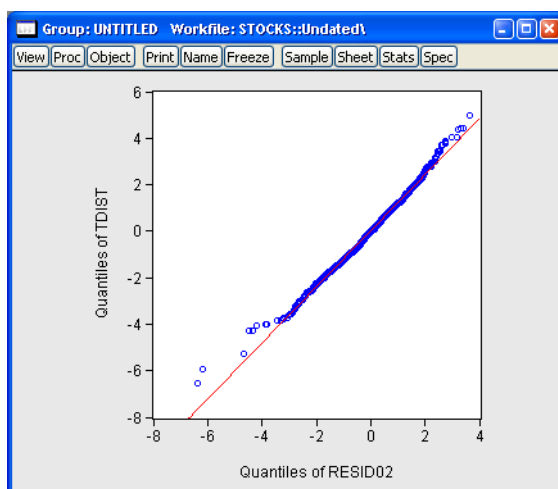
We can also plot the residuals against the quantiles of the t -distribution. Instead of using the built-in QQ-plot for the t -distribution, you could instead simulate a draw from a t -distribution and examine whether the quantiles of the simulated observations match the quantiles of the residuals (this technique is useful for distributions not supported by EViews). The command:

```
series tdist = @qtdist(rnd, 6.7)
```

simulates a random draw from the t -distribution with 6.7 degrees of freedom. Then, create a group containing the series RESID02 and TDIST. Select **View/Graph...** and choose **Quantile-Quantile** from the left-hand side of the dialog and **Empirical** from the **Q-Q graph** dropdown on the right-hand side.

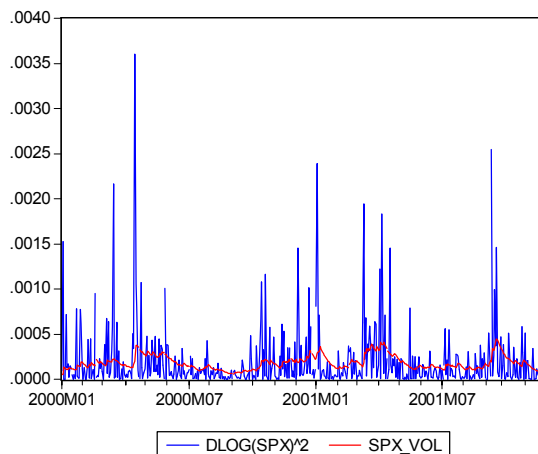
The large negative residuals more closely follow a straight line. On the other hand, one can see a slight deviation from t -distribution for large positive shocks. This is not unexpected, as the previous QQ-plot suggested that, with the exception of the large negative shocks, the residuals were close to normally distributed.

To see how the model might fit real data, we examine static forecasts for out-of-sample data. Click on **Forecast**, type in SPX_VOL in the GARCH field to save the forecasted



conditional variance, change the sample to the post-estimation sample period “1/1/2000 1/1/2002” and click on **Static** to select a static forecast.

Since the actual volatility is unobserved, we will use the squared return series ($\text{DLOG}(\text{SPX})^2$) as a proxy for the realized volatility. A plot of the proxy against the forecasted volatility provides an indication of the model’s ability to track variations in market volatility.



References

- Bollerslev, Tim (1986). “Generalized Autoregressive Conditional Heteroskedasticity,” *Journal of Econometrics*, 31, 307–327.
- Bollerslev, Tim, Ray Y. Chou, and Kenneth F. Kroner (1992). “ARCH Modeling in Finance: A Review of the Theory and Empirical Evidence,” *Journal of Econometrics*, 52, 5–59.
- Bollerslev, Tim, Robert F. Engle and Daniel B. Nelson (1994). “ARCH Models,” Chapter 49 in Robert F. Engle and Daniel L. McFadden (eds.), *Handbook of Econometrics, Volume 4*, Amsterdam: Elsevier Science B.V.
- Bollerslev, Tim and Jeffrey M. Wooldridge (1992). “Quasi-Maximum Likelihood Estimation and Inference in Dynamic Models with Time Varying Covariances,” *Econometric Reviews*, 11, 143–172.
- Ding, Zhuangxin, C. W. J. Granger, and R. F. Engle (1993). “A Long Memory Property of Stock Market Returns and a New Model,” *Journal of Empirical Finance*, 1, 83–106.
- Engle, Robert F. (1982). “Autoregressive Conditional Heteroskedasticity with Estimates of the Variance of U.K. Inflation,” *Econometrica*, 50, 987–1008.
- Engle, Robert F., and Bollerslev, Tim (1986). “Modeling the Persistence of Conditional Variances,” *Econometric Reviews*, 5, 1–50.
- Engle, Robert F., David M. Lilien, and Russell P. Robins (1987). “Estimating Time Varying Risk Premia in the Term Structure: The ARCH-M Model,” *Econometrica*, 55, 391–407.

- Glosten, L. R., R. Jaganathan, and D. Runkle (1993). "On the Relation between the Expected Value and the Volatility of the Normal Excess Return on Stocks," *Journal of Finance*, 48, 1779–1801.
- Nelson, Daniel B. (1991). "Conditional Heteroskedasticity in Asset Returns: A New Approach," *Econometrica*, 59, 347–370.
- Schwert, W. (1989). "Stock Volatility and Crash of '87," *Review of Financial Studies*, 3, 77–102.
- Taylor, S. (1986). *Modeling Financial Time Series*, New York: John Wiley & Sons.
- Zakoïan, J. M. (1994). "Threshold Heteroskedastic Models," *Journal of Economic Dynamics and Control*, 18, 931–944.

Chapter 30. Discrete and Limited Dependent Variable Models

The regression methods described in [Chapter 24. “Basic Regression”](#) and [Chapter 25. “Additional Regression Methods”](#) require that the dependent variable be observed on a continuous and unrestricted scale. It is quite common, however, for this condition to be violated, resulting in a non-continuous, or a limited dependent variable. We will distinguish between three types of these variables:

- qualitative (observed on a discrete or ordinal scale)
- censored or truncated
- integer valued

In this chapter, we discuss estimation methods for several qualitative and limited dependent variable models. EViews provides estimation routines for binary or ordered (probit, logit, gompit), censored or truncated (tobit, etc.), and integer valued (count data) models.

Standard introductory discussion for the models presented in this chapter may be found in Greene (1997), Johnston and DiNardo (1997), and Maddala (1983). Wooldridge (1996) provides an excellent reference for quasi-likelihood methods and count models.

Binary Dependent Variable Models

In this class of models, the dependent variable, y may take on only two values— y might be a dummy variable representing the occurrence of an event, or a choice between two alternatives. For example, you may be interested in modeling the employment status of each individual in your sample (whether employed or not). The individuals differ in age, educational attainment, race, marital status, and other observable characteristics, which we denote as x . The goal is to quantify the relationship between the individual characteristics and the probability of being employed.

Background

Suppose that a binary dependent variable, y , takes on values of zero and one. A simple linear regression of y on x is not appropriate, since among other things, the implied model of the conditional mean places inappropriate restrictions on the residuals of the model. Furthermore, the fitted value of y from a simple linear regression is not restricted to lie between zero and one.

Instead, we adopt a specification that is designed to handle the specific requirements of binary dependent variables. Suppose that we model the probability of observing a value of one as:

$$\Pr(y_i = 1 | x_i, \beta) = 1 - F(-x_i' \beta), \quad (30.1)$$

where F is a continuous, strictly increasing function that takes a real value and returns a value ranging from zero to one. In this, and the remaining discussion in this chapter follows we adopt the standard simplifying convention of assuming that the index specification is linear in the parameters so that it takes the form $x_i' \beta$. Note, however, that EViews allows you to estimate models with nonlinear index specifications.

The choice of the function F determines the type of binary model. It follows that:

$$\Pr(y_i = 0 | x_i, \beta) = F(-x_i' \beta). \quad (30.2)$$

Given such a specification, we can estimate the parameters of this model using the method of maximum likelihood. The likelihood function is given by:

$$l(\beta) = \sum_{i=0}^n y_i \log(1 - F(-x_i' \beta)) + (1 - y_i) \log(F(-x_i' \beta)). \quad (30.3)$$

The first order conditions for this likelihood are nonlinear so that obtaining parameter estimates requires an iterative solution. By default, EViews uses a second derivative method for iteration and computation of the covariance matrix of the parameter estimates. As discussed below, EViews allows you to override these defaults using the Options dialog (see [“Second Derivative Methods” on page 630](#) for additional details on the estimation methods).

There are two alternative interpretations of this specification that are of interest. First, the binary model is often motivated as a latent variables specification. Suppose that there is an unobserved latent variable y_i^* that is linearly related to x :

$$y_i^* = x_i' \beta + u_i \quad (30.4)$$

where u_i is a random disturbance. Then the observed dependent variable is determined by whether y_i^* exceeds a threshold value:

$$y_i = \begin{cases} 1 & \text{if } y_i^* > 0 \\ 0 & \text{if } y_i^* \leq 0. \end{cases} \quad (30.5)$$

In this case, the threshold is set to zero, but the choice of a threshold value is irrelevant, so long as a constant term is included in x_i . Then:

$$\Pr(y_i = 1 | x_i, \beta) = \Pr(y_i^* > 0) = \Pr(x_i' \beta + u_i > 0) = 1 - F_u(-x_i' \beta) \quad (30.6)$$

where F_u is the cumulative distribution function of u . Common models include probit (standard normal), logit (logistic), and gompit (extreme value) specifications for the F function.

In principle, the coding of the two numerical values of y is not critical since each of the binary responses only represents an event. Nevertheless, EViews requires that you code y as a zero-one variable. This restriction yields a number of advantages. For one, coding the variable in this fashion implies that expected value of y is simply the probability that $y = 1$:

$$\begin{aligned} E(y_i | x_i, \beta) &= 1 \cdot \Pr(y_i = 1 | x_i, \beta) + 0 \cdot \Pr(y_i = 0 | x_i, \beta) \\ &= \Pr(y_i = 1 | x_i, \beta). \end{aligned} \quad (30.7)$$

This convention provides us with a second interpretation of the binary specification: as a conditional mean specification. It follows that we can write the binary model as a regression model:

$$y_i = (1 - F(-x_i' \beta)) + \epsilon_i, \quad (30.8)$$

where ϵ_i is a residual representing the deviation of the binary y_i from its conditional mean. Then:

$$\begin{aligned} E(\epsilon_i | x_i, \beta) &= 0 \\ \text{var}(\epsilon_i | x_i, \beta) &= F(-x_i' \beta)(1 - F(-x_i' \beta)). \end{aligned} \quad (30.9)$$

We will use the conditional mean interpretation in our discussion of binary model residuals (see [“Make Residual Series” on page 222](#)).

Estimating Binary Models in EViews

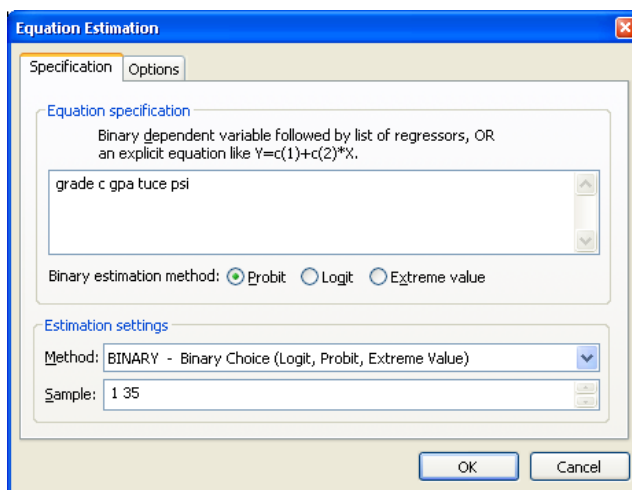
To estimate a binary dependent variable model, choose **Object/New Object...** from the main menu and select the **Equation** object from the main menu. From the **Equation Specification** dialog, select the **BINARY** estimation method. The dialog will change to reflect your choice.

There are two parts to the binary model specification. First, in the **Equation Specification** field, you may type the name of the binary dependent variable followed by a list of regressors or you may enter an explicit expression for the index. Next, select from among the three distributions for your error term:

| | |
|--------|--|
| Probit | $\Pr(y_i = 1 x_i, \beta) = 1 - \Phi(-x_i' \beta) = \Phi(x_i' \beta)$ <p>where Φ is the cumulative distribution function of the standard normal distribution.</p> |
|--------|--|

| | |
|------------------------|--|
| Logit | $\Pr(y_i = 1 x_i, \beta) = 1 - (e^{-x_i' \beta} / (1 + e^{-x_i' \beta}))$ $= e^{x_i' \beta} / (1 + e^{x_i' \beta})$ <p>which is based upon the cumulative distribution function for the logistic distribution.</p> |
| Extreme value (Gompit) | $\Pr(y_i = 1 x_i, \beta) = 1 - (1 - \exp(-e^{-x_i' \beta}))$ $= \exp(-e^{-x_i' \beta})$ <p>which is based upon the CDF for the Type-I extreme value distribution. Note that this distribution is skewed.</p> |

For example, consider the probit specification example described in Greene (1997, p. 876) where we analyze the effectiveness of teaching methods on grades. The variable GRADE represents improvement on grades following exposure to the new teaching method PSI. Also controlling for alternative measures of knowledge (GPA and TUCE), we have the specification:



Once you have specified the model, click **OK**. EViews estimates the parameters of the model using iterative procedures, and will display information in the status line. EViews requires that the dependent variable be coded with the values zero-one with all other observations dropped from the estimation.

Following estimation, EViews displays results in the equation window. The top part of the estimation output is given by:

Dependent Variable: GRADE

Method: ML - Binary Probit

Date: 07/31/00 Time: 15:57

Sample: 1 32

Included observations: 32

Convergence achieved after 5 iterations

Covariance matrix computed using second derivatives

| Variable | Coefficient | Std. Error | z-Statistic | Prob. |
|----------|-------------|------------|-------------|--------|
| C | -7.452320 | 2.542472 | -2.931131 | 0.0034 |
| GPA | 1.625810 | 0.693882 | 2.343063 | 0.0191 |
| TUCE | 0.051729 | 0.083890 | 0.616626 | 0.5375 |
| PSI | 1.426332 | 0.595038 | 2.397045 | 0.0165 |

The header contains basic information regarding the estimation technique (ML for maximum likelihood) and the sample used in estimation, as well as information on the number of iterations required for convergence, and on the method used to compute the coefficient covariance matrix.

Displayed next are the coefficient estimates, asymptotic standard errors, z -statistics and corresponding p -values.

Interpretation of the coefficient values is complicated by the fact that estimated coefficients from a binary model cannot be interpreted as the marginal effect on the dependent variable. The marginal effect of x_j on the conditional probability is given by:

$$\frac{\partial E(y_i | x_i, \beta)}{\partial x_{ij}} = f(-x_i' \beta) \beta_j, \quad (30.10)$$

where $f(x) = dF(x)/dx$ is the density function corresponding to F . Note that β_j is weighted by a factor f that depends on the values of *all* of the regressors in x . The *direction* of the effect of a change in x_j depends only on the sign of the β_j coefficient. Positive values of β_j imply that increasing x_j will increase the probability of the response; negative values imply the opposite.

While marginal effects calculation is not provided as a built-in view or procedure, in “[Forecast](#)” on page 221, we show you how to use EViews to compute the marginal effects.

An alternative interpretation of the coefficients results from noting that the ratios of coefficients provide a measure of the relative changes in the probabilities:

$$\frac{\beta_j}{\beta_k} = \frac{\partial E(y_i | x_i, \beta) / \partial x_{ij}}{\partial E(y_i | x_i, \beta) / \partial x_{ik}}. \quad (30.11)$$

In addition to the summary statistics of the dependent variable, EViews also presents the following summary statistics:

| | | | |
|-----------------------|-----------|-----------------------|-----------|
| Mean dependent var | 0.343750 | S.D. dependent var | 0.482559 |
| S.E. of regression | 0.386128 | Akaike info criterion | 1.051175 |
| Sum squared resid | 4.174660 | Schwarz criterion | 1.234392 |
| Log likelihood | -12.81880 | Hannan-Quinn criter. | 1.111907 |
| Restr. log likelihood | -20.59173 | Avg. log likelihood | -0.400588 |
| LR statistic (3 df) | 15.54585 | McFadden R-squared | 0.377478 |
| Probability(LR stat) | 0.001405 | | |
| Obs with Dep=0 | 21 | Total obs | 32 |
| Obs with Dep=1 | 11 | | |

First, there are several familiar summary descriptive statistics: the mean and standard deviation of the dependent variable, standard error of the regression, and the sum of the squared residuals. The latter two measures are computed in the usual fashion using the residuals:

$$e_i = y_i - E(y_i | x_i, \hat{\beta}) = y_i - (1 - F(-x_i' \hat{\beta})). \quad (30.12)$$

Additionally, there are several likelihood based statistics:

- **Log likelihood** is the maximized value of the log likelihood function $l(\hat{\beta})$.
- **Avg. log likelihood** is the log likelihood $l(\hat{\beta})$ divided by the number of observations n .
- **Restr. log likelihood** is the maximized log likelihood value, when all slope coefficients are restricted to zero, $l(\tilde{\beta})$. Since the constant term is included, this specification is equivalent to estimating the unconditional mean probability of “success”.
- The **LR statistic** tests the joint null hypothesis that all slope coefficients except the constant are zero and is computed as $-2(l(\tilde{\beta}) - l(\hat{\beta}))$. This statistic, which is only reported when you include a constant in your specification, is used to test the overall significance of the model. The number in parentheses is the degrees of freedom, which is the number of restrictions under test.
- **Probability(LR stat)** is the p -value of the LR test statistic. Under the null hypothesis, the LR test statistic is asymptotically distributed as a χ^2 variable, with degrees of freedom equal to the number of restrictions under test.
- **McFadden R-squared** is the likelihood ratio index computed as $1 - l(\tilde{\beta})/l(\hat{\beta})$, where $l(\tilde{\beta})$ is the restricted log likelihood. As the name suggests, this is an analog to the R^2 reported in linear regression models. It has the property that it always lies between zero and one.
- The various information criteria are detailed in [Appendix F. “Information Criteria,” beginning on page 645](#). For additional discussion, see Grasa (1989).

Estimation Options

The iteration limit and convergence criterion may be set in the usual fashion by clicking on the **Options** tab in the **Equation Estimation** dialog. In addition, there are options that are specific to binary models. These options are described below.

Robust Covariances

For binary dependent variable models, EViews allows you to estimate the standard errors using quasi-maximum likelihood (**Huber/White**) or generalized linear model (**GLM**) methods. See [“Technical Notes” on page 255](#) for a discussion of these two methods.

Click on the **Options** tab to bring up the settings, check the **Robust Covariance** box and select one of the two methods. When you estimate the binary model using this option, the header in the equation output will indicate the method used to compute the coefficient covariance matrix.

Starting Values

As with other estimation procedures, EViews allows you to specify starting values. In the options menu, select one of the items from the combo box. You can use the default EViews values, or you can choose a fraction of those values, zero coefficients, or user supplied values. To employ the latter, enter the coefficients in the C coefficient vector, and select **User Supplied** in the combo box.

The EViews default values are selected using a sophisticated algorithm that is specialized for each type of binary model. Unless there is a good reason to choose otherwise, we recommend that you use the default values.

Estimation Algorithm

By default, EViews uses quadratic hill-climbing to obtain parameter estimates. This algorithm uses the matrix of analytic second derivatives of the log likelihood in forming iteration updates and in computing the estimated covariance matrix of the coefficients.

If you wish, you can employ a different estimation algorithm: Newton-Raphson also employs second derivatives (without the diagonal weighting); BHHH uses first derivatives to determine both iteration updates and the covariance matrix estimates (see [Appendix D. “Estimation and Solution Options,” on page 625](#)). To employ one of these latter methods, click on **Options** in the Equation specification dialog box, and select the desired method.

Estimation Problems

In general, estimation of binary models is quite straightforward, and you should experience little difficulty in obtaining parameter estimates. There are a few situations, however, where you may experience problems.

First, you may get the error message “Dependent variable has no variance.” This error means that there is no variation in the dependent variable (the variable is always one or zero for all valid observations). This error most often occurs when EViews excludes the entire sample of observations for which y takes values other than zero or one, leaving too few observations for estimation.

You should make certain to recode your data so that the binary indicators take the values zero and one. This requirement is not as restrictive as it may first seem, since the recoding may easily be done using auto-series. Suppose, for example, that you have data where y takes the values 1000 and 2000. You could then use the boolean auto-series, “ $y=1000$ ”, or perhaps, “ $y<1500$ ”, as your dependent variable.

Second, you may receive an error message of the form “[xxxx] perfectly predicts binary response [success/failure]”, where xxxx is a sample condition. This error occurs when one of the regressors contains a separating value for which all of the observations with values below the threshold are associated with a single binary response, and all of the values above the threshold are associated with the alternative response. In this circumstance, the method of maximum likelihood breaks down.

For example, if all values of the explanatory variable $x > 0$ are associated with $y = 1$, then x is a perfect predictor of the dependent variable, and EViews will issue an error message and stop the estimation procedure.

The only solution to this problem is to remove the offending variable from your specification. Usually, the variable has been incorrectly entered in the model, as when a researcher includes a dummy variable that is identical to the dependent variable (for discussion, see Greene, 1997).

Thirdly, you may experience the error, “Non-positive likelihood value observed for observation [xxxx].” This error most commonly arises when the starting values for estimation are poor. The default EViews starting values should be adequate for most uses. You may wish to check the Options dialog to make certain that you are not using user specified starting values, or you may experiment with alternative user-specified values.

Lastly, the error message “Near-singular matrix” indicates that EViews was unable to invert the matrix required for iterative estimation. This will occur if the model is not identified. It may also occur if the current parameters are far from the true values. If you believe the latter to be the case, you may wish to experiment with starting values or the estimation algorithm. The BHHH and quadratic hill-climbing algorithms are less sensitive to this particular problem than is Newton-Raphson.

Views of Binary Equations

EViews provides a number of standard views and procedures for binary models. For example, you can easily perform Wald or likelihood ratio tests by selecting **View/Coefficient**

Tests, and then choosing the appropriate test. In addition, EViews allows you to examine and perform tests using the residuals from your model. The ordinary residuals used in most calculations are described above—additional residual types are defined below. Note that some care should be taken in interpreting test statistics that use these residuals since some of the underlying test assumptions may not be valid in the current setting.

There are a number of additional specialized views and procedures which allow you to examine the properties and performance of your estimated binary model.

Categorical Regressor Stats

This view displays descriptive statistics (mean and standard deviation) for each regressor. The descriptive statistics are computed for the whole sample, as well as the sample broken down by the value of the dependent variable y .

Expectation-Prediction (Classification) Table

This view displays 2×2 tables of correct and incorrect classification based on a user specified prediction rule, and on expected value calculations. Click on **View/Expectation-Prediction Table**. EViews opens a dialog prompting you to specify a prediction cutoff value, p , lying between zero and one. Each observation will be classified as having a predicted probability that lies above or below this cutoff.

After you enter the cutoff value and click on **OK**, EViews will display four (bordered) 2×2 tables in the equation window. Each table corresponds to a contingency table of the predicted response classified against the observed dependent variable. The top two tables and associated statistics depict the classification results based upon the specified cutoff value:

| Dependent Variable: GRADE | | | | | | |
|--|--------------------|-------|-------|----------------------|--------|-------|
| Method: ML - Binary Probit | | | | | | |
| Date: 07/31/00 Time: 15:57 | | | | | | |
| Sample: 1 32 | | | | | | |
| Included observations: 32 | | | | | | |
| Prediction Evaluation (success cutoff C = 0.5) | | | | | | |
| | Estimated Equation | | | Constant Probability | | |
| | Dep=0 | Dep=1 | Total | Dep=0 | Dep=1 | Total |
| P(Dep=1)≤C | 18 | 3 | 21 | 21 | 11 | 32 |
| P(Dep=1)>C | 3 | 8 | 11 | 0 | 0 | 0 |
| Total | 21 | 11 | 32 | 21 | 11 | 32 |
| Correct | 18 | 8 | 26 | 21 | 0 | 21 |
| % Correct | 85.71 | 72.73 | 81.25 | 100.00 | 0.00 | 65.62 |
| % Incorrect | 14.29 | 27.27 | 18.75 | 0.00 | 100.00 | 34.38 |
| Total Gain* | -14.29 | 72.73 | 15.62 | | | |
| Percent Gain** | NA | 72.73 | 45.45 | | | |

In the left-hand table, we classify observations as having predicted probabilities $\hat{p}_i = 1 - F(-x_i'\beta)$ that are above or below the specified cutoff value (here set to the default of 0.5). In the upper right-hand table, we classify observations using \bar{p} , the sample

proportion of $y = 1$ observations. This probability, which is constant across individuals, is the value computed from estimating a model that includes only the intercept term, C .

“Correct” classifications are obtained when the predicted probability is less than or equal to the cutoff and the observed $y = 0$, or when the predicted probability is greater than the cutoff and the observed $y = 1$. In the example above, 18 of the $\text{Dep} = 0$ observations and 8 of the $\text{Dep} = 1$ observations are correctly classified by the estimated model.

It is worth noting that in the statistics literature, what we term the expectation-prediction table is sometimes referred to as the *classification table*. The fraction of $y = 1$ observations that are correctly predicted is termed the *sensitivity*, while the fraction of $y = 0$ observations that are correctly predicted is known as *specificity*. In EViews, these two values, expressed in percentage terms, are labeled “% Correct”. Overall, the estimated model correctly predicts 81.25% of the observations (85.71% of the $\text{Dep} = 0$ and 72.73% of the $\text{Dep} = 1$ observations).

The gain in the number of correct predictions obtained in moving from the right table to the left table provides a measure of the predictive ability of your model. The gain measures are reported in both absolute percentage increases (**Total Gain**), and as a percentage of the incorrect classifications in the constant probability model (**Percent Gain**). In the example above, the restricted model predicts that all 21 individuals will have $\text{Dep} = 0$. This prediction is correct for the 21 $y = 0$ observations, but is incorrect for the 11 $y = 1$ observations.

The estimated model improves on the $\text{Dep} = 1$ predictions by 72.73 percentage points, but does more poorly on the $\text{Dep} = 0$ predictions (-14.29 percentage points). Overall, the estimated equation is 15.62 percentage points better at predicting responses than the constant probability model. This change represents a 45.45 percent improvement over the 65.62 percent correct prediction of the default model.

The bottom portion of the equation window contains analogous prediction results based upon expected value calculations:

| | Estimated Equation | | | Constant Probability | | |
|----------------|--------------------|-------|-------|----------------------|-------|-------|
| | Dep=0 | Dep=1 | Total | Dep=0 | Dep=1 | Total |
| E(# of Dep=0) | 16.89 | 4.14 | 21.03 | 13.78 | 7.22 | 21.00 |
| E(# of Dep=1) | 4.11 | 6.86 | 10.97 | 7.22 | 3.78 | 11.00 |
| Total | 21.00 | 11.00 | 32.00 | 21.00 | 11.00 | 32.00 |
| Correct | 16.89 | 6.86 | 23.74 | 13.78 | 3.78 | 17.56 |
| % Correct | 80.42 | 62.32 | 74.20 | 65.62 | 34.38 | 54.88 |
| % Incorrect | 19.58 | 37.68 | 25.80 | 34.38 | 65.62 | 45.12 |
| Total Gain* | 14.80 | 27.95 | 19.32 | | | |
| Percent Gain** | 43.05 | 42.59 | 42.82 | | | |

In the left-hand table, we compute the expected number of $y = 0$ and $y = 1$ observations in the sample. For example, **E(# of Dep = 0)** is computed as:

$$\sum_i \Pr(y_i = 0 | x_i, \beta) = \sum_i F(-x_i' \hat{\beta}), \quad (30.13)$$

where the cumulative distribution function F is for the normal, logistic, or extreme value distribution.

In the lower right-hand table, we compute the expected number of $y = 0$ and $y = 1$ observations for a model estimated with only a constant. For this restricted model, $\mathbf{E}(\# \text{ of Dep} = 0)$ is computed as $n(1 - \bar{p})$, where \bar{p} is the sample proportion of $y = 1$ observations. EViews also reports summary measures of the total gain and the percent (of the incorrect expectation) gain.

Among the 21 individuals with $y = 0$, the expected number of $y = 0$ observations in the estimated model is 16.89. Among the 11 observations with $y = 1$, the expected number of $y = 1$ observations is 6.86. These numbers represent roughly a 19.32 percentage point (42.82 percent) improvement over the constant probability model.

Goodness-of-Fit Tests

This view allows you to perform Pearson χ^2 -type tests of goodness-of-fit. EViews carries out two goodness-of-fit tests: Hosmer-Lemeshow (1989) and Andrews (1988a, 1988b). The idea underlying these tests is to compare the fitted expected values to the actual values *by group*. If these differences are “large”, we reject the model as providing an insufficient fit to the data.

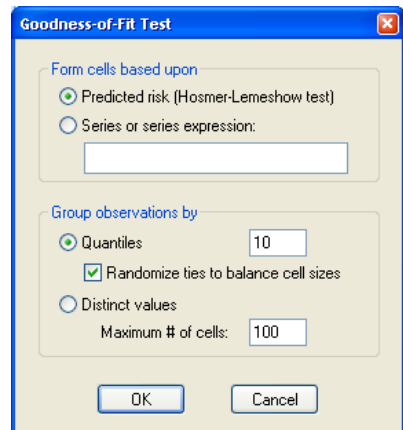
Details on the two tests are described in the “[Technical Notes](#)” on page 255. Briefly, the tests differ in how the observations are grouped and in the asymptotic distribution of the test statistic. The Hosmer-Lemeshow test groups observations on the basis of the predicted probability that $y = 1$. The Andrews test is a more general test that groups observations on the basis of any series or series expression.

To carry out the test, select **View/Goodness-of-Fit Test...**

You must first decide on the grouping variable.

You can select Hosmer-Lemeshow (predicted probability) grouping by clicking on the corresponding radio button, or you can select series grouping, and provide a series to be used in forming the groups.

Next, you need to specify the grouping rule. EViews allows you to group on the basis of either distinct values or quantiles of the grouping variable.



If your grouping variable takes relatively few distinct values, you should choose the **Distinct values** grouping. EViews will form a separate group for each distinct value of the grouping variable. For example, if your grouping variable is TUCE, EViews will create a group for each distinct TUCE value and compare the expected and actual numbers of $y = 1$ observations in each group. By default, EViews limits you to 100 distinct values. If the distinct values in your grouping series exceeds this value, EViews will return an error message. If you wish to evaluate the test for more than 100 values, you must explicitly increase the maximum number of distinct values.

If your grouping variable takes on a large number of distinct values, you should select **Quantiles**, and enter the number of desired bins in the edit field. If you select this method, EViews will group your observations into the number of specified bins, on the basis of the ordered values of the grouping series. For example, if you choose to group by TUCE, select **Quantiles**, and enter 10, EViews will form groups on the basis of TUCE deciles.

If you choose to group by quantiles and there are ties in the grouping variable, EViews may not be able to form the exact number of groups you specify unless tied values are assigned to different groups. Furthermore, the number of observations in each group may be very unbalanced. Selecting the **randomize ties** option randomly assigns ties to adjacent groups in order to balance the number of observations in each group.

Since the properties of the test statistics require that the number of observations in each group is “large”, some care needs to be taken in selecting a rule so that you do not end up with a large number of cells, each containing small numbers of observations.

By default, EViews will perform the test using Hosmer-Lemeshow grouping. The default grouping method is to form deciles. The test result using the default specification is given by:

Dependent Variable: GRADE
 Method: ML - Binary Probit
 Date: 07/31/00 Time: 15:57
 Sample: 1 32
 Included observations: 32
 Andrews and Hosmer-Lemeshow Goodness-of-Fit Tests
 Grouping based upon predicted risk (randomize ties)

| | Quantile of Risk | | Dep=0 | | Dep=1 | | Total Obs | H-L Value |
|--------------------|------------------|--------|---------|---------|------------------|---------|--------------|--------------|
| | Low | High | Actual | Expect | Actual | Expect | | |
| 1 | 0.0161 | 0.0185 | 3 | 2.94722 | 0 | 0.05278 | 3 | 0.05372 |
| 2 | 0.0186 | 0.0272 | 3 | 2.93223 | 0 | 0.06777 | 3 | 0.06934 |
| 3 | 0.0309 | 0.0457 | 3 | 2.87888 | 0 | 0.12112 | 3 | 0.12621 |
| 4 | 0.0531 | 0.1088 | 3 | 2.77618 | 0 | 0.22382 | 3 | 0.24186 |
| 5 | 0.1235 | 0.1952 | 2 | 3.29779 | 2 | 0.70221 | 4 | 2.90924 |
| 6 | 0.2732 | 0.3287 | 3 | 2.07481 | 0 | 0.92519 | 3 | 1.33775 |
| 7 | 0.3563 | 0.5400 | 2 | 1.61497 | 1 | 1.38503 | 3 | 0.19883 |
| 8 | 0.5546 | 0.6424 | 1 | 1.20962 | 2 | 1.79038 | 3 | 0.06087 |
| 9 | 0.6572 | 0.8342 | 0 | 0.84550 | 3 | 2.15450 | 3 | 1.17730 |
| 10 | 0.8400 | 0.9522 | 1 | 0.45575 | 3 | 3.54425 | 4 | 0.73351 |
| Total | | | 21 | 21.0330 | 11 | 10.9670 | 32 | 6.90863 |
| H-L Statistic: | | | 6.9086 | | Prob. Chi-Sq(8) | | 0.5465 | |
| Andrews Statistic: | | | 20.6045 | | Prob. Chi-Sq(10) | | 0.0240 | |

The columns labeled “Quantiles of Risk” depict the high and low value of the predicted probability for each decile. Also depicted are the actual and expected number of observations in each group, as well as the contribution of each group to the overall Hosmer-Lemeshow (H-L) statistic—large values indicate large differences between the actual and predicted values for that decile.

The χ^2 statistics are reported at the bottom of the table. Since grouping on the basis of the fitted values falls within the structure of an Andrews test, we report results for both the H-L and the Andrews test statistic. The p -value for the HL test is large while the value for the Andrews test statistic is small, providing mixed evidence of problems. Furthermore, the relatively small sample sizes suggest that caution is in order in interpreting the results.

Procedures for Binary Equations

In addition to the usual procedures for equations, EViews allows you to forecast the dependent variable and linear index, or to compute a variety of residuals associated with the binary model.

Forecast

EViews allows you to compute either the fitted probability, $\hat{p}_i = 1 - F(-x_i'\hat{\beta})$, or the fitted values of the index $x_i'\hat{\beta}$. From the equation toolbar select **Proc/Forecast (Fitted Probability/Index)...**, and then click on the desired entry.

As with other estimators, you can select a forecast sample, and display a graph of the forecast. If your explanatory variables, x_t , include lagged values of the binary dependent variable y_t , forecasting with the **Dynamic** option instructs EViews to use the fitted values

\hat{p}_{t-1} , to derive the forecasts, in contrast with the **Static** option, which uses the actual (lagged) y_{t-1} .

Neither forecast evaluations nor automatic calculation of standard errors of the forecast are currently available for this estimation method. The latter can be computed using the variance matrix of the coefficients displayed by **View/Covariance Matrix**, or using the `@covariance` function.

You can use the fitted index in a variety of ways, for example, to compute the marginal effects of the explanatory variables. Simply forecast the fitted index and save the results in a series, say XB. Then the auto-series `@dnorm(-xb)`, `@dlogistic(-xb)`, or `@dextreme(-xb)` may be multiplied by the coefficients of interest to provide an estimate of the derivatives of the expected value of y_i with respect to the j -th variable in x_i :

$$\frac{\partial E(y_i | x_i, \beta)}{\partial x_{ij}} = f(-x_i' \beta) \beta_j. \quad (30.14)$$

Make Residual Series

Proc/Make Residual Series gives you the option of generating one of the following three types of residuals:

| | |
|--------------|---|
| Ordinary | $e_{oi} = y_i - \hat{p}_i$ |
| Standardized | $e_{si} = \frac{y_i - \hat{p}_i}{\sqrt{\hat{p}_i(1 - \hat{p}_i)}}$ |
| Generalized | $e_{gi} = \frac{(y_i - \hat{p}_i)f(-x_i' \hat{\beta})}{\hat{p}_i(1 - \hat{p}_i)}$ |

where $\hat{p}_i = 1 - F(-x_i' \hat{\beta})$ is the fitted probability, and the distribution and density functions F and f , depend on the specified distribution.

The ordinary residuals have been described above. The standardized residuals are simply the ordinary residuals divided by an estimate of the theoretical standard deviation. The generalized residuals are derived from the first order conditions that define the ML estimates. The first order conditions may be regarded as an orthogonality condition between the generalized residuals and the regressors x .

$$\frac{\partial l(\beta)}{\partial \beta} = \sum_{i=1}^N \frac{(y_i - (1 - F(-x_i' \beta)))f(-x_i' \beta)}{F(-x_i' \beta)(1 - F(-x_i' \beta))} \cdot x_i = \sum_{i=1}^N e_{gi} \cdot x_i. \quad (30.15)$$

This property is analogous to the orthogonality condition between the (ordinary) residuals and the regressors in linear regression models.

The usefulness of the generalized residuals derives from the fact that you can easily obtain the score vectors by multiplying the generalized residuals by each of the regressors in x . These scores can be used in a variety of LM specification tests (see Chesher, Lancaster and Irish (1985), and Gourieroux, Monfort, Renault, and Trognon (1987)). We provide an example below.

Demonstrations

You can easily use the results of a binary model in additional analysis. Here, we provide demonstrations of using EViews to plot a probability response curve and to test for heteroskedasticity in the residuals.

Plotting Probability Response Curves

You can use the estimated coefficients from a binary model to examine how the predicted probabilities vary with an independent variable. To do so, we will use the EViews built-in modeling features.

For the probit example above, suppose we are interested in the effect of teaching method (PSI) on educational improvement (GRADE). We wish to plot the fitted probabilities of GRADE improvement as a function of GPA for the two values of PSI, fixing the values of other variables at their sample means.

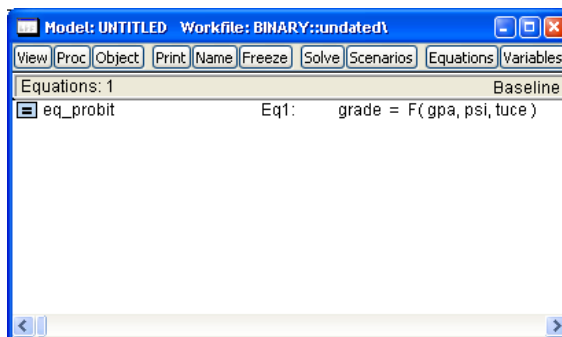
We will perform the analysis using a grid of values for GPA from 2 to 4. First, we will create a series containing the values of GPA for which we wish to examine the fitted probabilities for GRADE. The easiest way to do this is to use the `@trend` function to generate a new series:

```
series gpa_plot=2+(4-2)*@trend/(@obs(@trend)-1)
```

`@trend` creates a series that begins at 0 in the first observation of the sample, and increases by 1 for each subsequent observation, up through `@obs-1`.

Next, we will use a model object to define and perform the desired computations. The following discussion skims over many of the useful features of EViews models. Those wishing greater detail should consult [Chapter 36. “Models,” beginning on page 407](#).

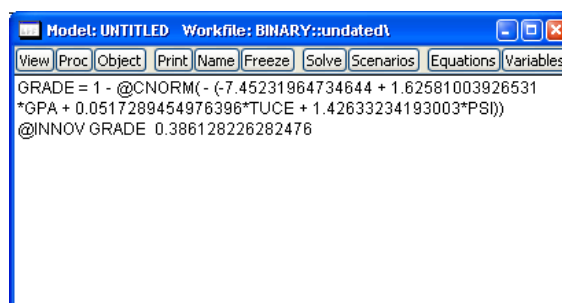
First, we create a model out of the estimated equation by selecting **Proc/Make Model** from the equation toolbar. EViews will create an untitled model object containing a link to the estimated equation and will open the model window.



Next we want to edit this model specification so that calculations are performed using our simulation values. To do so we must first break the link between the original equation and the model specification by selecting **Proc/Links/Break All Links**. Next, click on the Text button or select **View/Source Text** to display the text editing screen.

We wish to create two separate equations: one with the value of PSI set to 0 and one with the value of PSI set to 1 (you can, of course, use copy-and-paste to aid in creating the additional equation). We will also edit the specification so that references to GPA are replaced with the series of simulation values GPA_PLOT, and references to TUCE are replaced with the calculated mean, “@MEAN(TUCE)”. The GRADE_0 equation sets PSI to 0, while the GRADE_1 contains an additional expression, 1.426332342, which is the coefficient on the PSI variable.

Once you have edited your model, click on **Solve** and set the “Active” solution scenario to “Actuals”. This tells EViews that you wish to place the solutions in the series “GRADE_0” and “GRADE_1” as specified in the equation definitions. You can safely ignore the remaining solution settings and simply click on **OK**. EViews will report that your model has solved successfully.



You are now ready to plot results. Select **Object/New Object.../Group**, and enter:

```
gpa_plot grade_0 grade_1
```

EViews will open an untitled group window containing these two series. Select **View/Graph/XY line** to display the probability of GRADE improvement plotted against GPA for those with and without PSI (and with the TUCE evaluated at means).

EViews will open a group window containing these series. Select **View/Graph/XY line** from the group toolbar to display a graph of your results.

We have annotated the graph slightly so that you can better judge the effect of the new teaching methods (PSI) on the GPA—Grade Improvement relationship.

Testing for Heteroskedasticity

As an example of specification tests for binary dependent variable models, we carry out the LM test for heteroskedasticity using the artificial regression method described by Davidson and MacKinnon (1993, section 15.4). We test the null hypothesis of homoskedasticity against the alternative of heteroskedasticity of the form:

$$\text{var}(u_i) = \exp(2z_i'\gamma), \quad (30.16)$$

where γ is an unknown parameter. In this example, we take PSI as the only variable in z . The test statistic is the explained sum of squares from the regression:

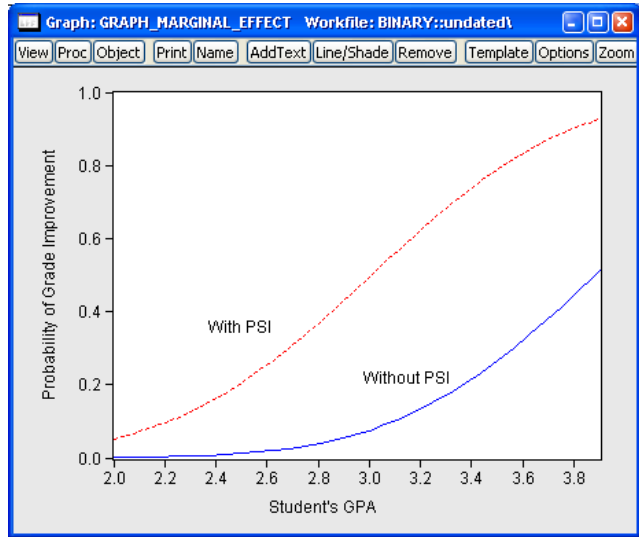
$$\frac{(y_i - \hat{p}_i)}{\sqrt{\hat{p}_i(1 - \hat{p}_i)}} = \frac{f(-x_i'\hat{\beta})}{\sqrt{\hat{p}_i(1 - \hat{p}_i)}} x_i' b_1 + \frac{f(-x_i'\hat{\beta})(-x_i'\hat{\beta})}{\sqrt{\hat{p}_i(1 - \hat{p}_i)}} z_i' b_2 + v_i, \quad (30.17)$$

which is asymptotically distributed as a χ^2 with degrees of freedom equal to the number of variables in z (in this case 1).

To carry out the test, we first retrieve the fitted probabilities \hat{p}_i and fitted index $x_i'\hat{\beta}$. Click on the **Forecast** button and first save the fitted probabilities as P_HAT and then the index as XB (you will have to click **Forecast** twice to save the two series).

Next, the dependent variable in the test regression may be obtained as the standardized residual. Select **Proc/Make Residual Series...** and select **Standardized Residual**. We will save the series as BRMR_Y.

Lastly, we will use the built-in EViews functions for evaluating the normal density and cumulative distribution function to create a group object containing the independent variables:



```
series fac=@dnorm(-xb)/@sqrt(p_hat*(1-p_hat))
group brmr_x fac (gpa*fac) (tuce*fac) (psi*fac)
```

Then run the artificial regression by clicking on **Quick/Estimate Equation...**, selecting **Least Squares**, and entering:

```
brmr_y brmr_x (psi*(-xb)*fac)
```

You can obtain the fitted values by clicking on the **Forecast** button in the equation toolbar of this artificial regression. The LM test statistic is the sum of squares of these fitted values. If the fitted values from the artificial regression are saved in BRMR_YF, the test statistic can be saved as a scalar named LM_TEST:

```
scalar lm_test=@sumsq(brmr_yf)
```

which contains the value 1.5408. You can compare the value of this test statistic with the critical values from the chi-square table with one degree of freedom. To save the p -value as a scalar, enter the command:

```
scalar p_val=1-@cchisq(lm_test,1)
```

To examine the value of LM_TEST or P_VAL, double click on the name in the workfile window; the value will be displayed in the status line at the bottom of the EVIEWS window. The p -value in this example is roughly 0.21, so we have little evidence against the null hypothesis of homoskedasticity.

Ordered Dependent Variable Models

EVIEWS estimates the ordered-response model of Aitchison and Silvey (1957) under a variety of assumptions about the latent error distribution. In ordered dependent variable models, the observed y denotes outcomes representing ordered or ranked categories. For example, we may observe individuals who choose between one of four educational outcomes: less than high school, high school, college, advanced degree. Or we may observe individuals who are employed, partially retired, or fully retired.

As in the binary dependent variable model, we can model the observed response by considering a latent variable y_i^* that depends linearly on the explanatory variables x_i :

$$y_i^* = x_i' \beta + \epsilon_i \quad (30.18)$$

where ϵ_i are independent and identically distributed random variables. The observed y_i is determined from y_i^* using the rule:

$$y_i = \begin{cases} 0 & \text{if } y_i^* \leq \gamma_1 \\ 1 & \text{if } \gamma_1 < y_i^* \leq \gamma_2 \\ 2 & \text{if } \gamma_2 < y_i^* \leq \gamma_3 \\ \vdots & \vdots \\ M & \text{if } \gamma_M < y_i^* \end{cases} \quad (30.19)$$

It is worth noting that the actual values chosen to represent the categories in y are completely arbitrary. All the ordered specification requires is for ordering to be preserved so that $y_i^* < y_j^*$ implies that $y_i < y_j$.

It follows that the probabilities of observing each value of y are given by

$$\begin{aligned} \Pr(y_i = 0 | x_i, \beta, \gamma) &= F(\gamma_1 - x_i' \beta) \\ \Pr(y_i = 1 | x_i, \beta, \gamma) &= F(\gamma_2 - x_i' \beta) - F(\gamma_1 - x_i' \beta) \\ \Pr(y_i = 2 | x_i, \beta, \gamma) &= F(\gamma_3 - x_i' \beta) - F(\gamma_2 - x_i' \beta) \\ &\dots \\ \Pr(y_i = M | x_i, \beta, \gamma) &= 1 - F(\gamma_M - x_i' \beta) \end{aligned} \quad (30.20)$$

where F is the cumulative distribution function of ϵ .

The threshold values γ are estimated along with the β coefficients by maximizing the log likelihood function:

$$l(\beta, \gamma) = \sum_{i=1}^N \sum_{j=0}^M \log(\Pr(y_i = j | x_i, \beta, \gamma)) \cdot 1(y_i = j) \quad (30.21)$$

where $1(\cdot)$ is an indicator function which takes the value 1 if the argument is true, and 0 if the argument is false. By default, EViews uses analytic second derivative methods to obtain parameter and variance matrix of the estimated coefficient estimates (see [“Quadratic hill-climbing \(Goldfeld-Quandt\)” on page 631](#)).

Estimating Ordered Models in EViews

Suppose that the dependent variable DANGER is an index ordered from 1 (least dangerous animal) to 5 (most dangerous animal). We wish to model this ordered dependent variable as a function of the explanatory variables, BODY, BRAIN and SLEEP. Note that the values that we have assigned to the dependent variable are not relevant, only the ordering implied by those values. EViews will estimate an identical model if the dependent variable is recorded to take the values 1, 2, 3, 4, 5 or 10, 234, 3243, 54321, 123456.

To estimate this model, select **Quick/Estimate Equation...** from the main menu. From the **Equation Estimation** dialog, select estimation method **ORDERED**. The standard estimation dialog will change to match this specification.

There are three parts to specifying an ordered variable model: the equation specification, the error specification, and the sample specification. First, in the **Equation specification** field, you should type the name of the ordered dependent variable followed by the list of your regressors, or you may enter an explicit expression for the index. In our example, you will enter:

```
danger body brain sleep
```

Also keep in mind that:

- A separate constant term is not separately identified from the limit points γ , so EViews will ignore any constant term in your specification. Thus, the model:

```
danger c body brain sleep
```

is equivalent to the specification above.

- EViews requires the dependent variable to be integer valued, otherwise you will see an error message, and estimation will stop. This is not, however, a serious restriction, since you can easily convert the series into an integer using `@round`, `@floor` or `@ceil` in an auto-series expression.

Next, select between the ordered logit, ordered probit, and the ordered extreme value models by choosing one of the three distributions for the latent error term.

Lastly, specify the estimation sample.

Now click on **OK**, EViews will estimate the parameters of the model using iterative procedures.

Once the estimation procedure converges, EViews will display the estimation results in the equation window. The first part of the table contains the usual header information, including the assumed error distribution, estimation sample, iteration and convergence information, number of distinct values for y , and the method of computing the coefficient covariance matrix.

Dependent Variable: DANGER

Method: ML - Ordered Probit

Date: 09/13/97 Time: 10:00

Sample(adjusted): 1 61

Included observations: 58

Excluded observations: 3 after adjusting endpoints

Number of ordered indicator values: 5

Convergence achieved after 5 iterations

Covariance matrix computed using second derivatives

| | Coefficient | Std. Error | z-Statistic | Prob. |
|-------|-------------|------------|-------------|--------|
| BODY | 0.006346 | 0.003262 | 1.945385 | 0.0517 |
| BRAIN | -0.003506 | 0.001822 | -1.924244 | 0.0543 |
| SLEEP | -0.158596 | 0.040440 | -3.921741 | 0.0001 |

Below the header information are the coefficient estimates and asymptotic standard errors, and the corresponding z -statistics and significance levels. The estimated coefficients of the ordered model must be interpreted with care (see Greene (1997, section 19.8) or Johnston and DiNardo (1997, section 13.9)).

The *sign* of $\hat{\beta}_j$ shows the direction of the change in the probability of falling in the endpoint rankings ($y = 0$ or $y = 1$) when x_{ij} changes. $\Pr(y = 0)$ changes in the *opposite* direction of the sign of $\hat{\beta}_j$ and $\Pr(y = M)$ changes in the *same* direction as the sign of $\hat{\beta}_j$. The effects on the probability of falling in any of the middle rankings are given by:

$$\frac{\partial \Pr(y = k)}{\partial \beta_j} = \frac{\partial F(\gamma_{k+1} - x_i' \beta)}{\partial \beta_j} - \frac{\partial F(\gamma_k - x_i' \beta)}{\partial \beta_j} \quad (30.22)$$

for $k = 1, 2, \dots, M - 1$. It is impossible to determine the signs of these terms, *a priori*.

The lower part of the estimation output, labeled “Limit Points”, presents the estimates of the γ coefficients and the associated standard errors and probability values:

| LimitPoints | | | | |
|-----------------------|-----------|--------------------|-----------|--------|
| Limit_2:C(4) | -2.382697 | 0.512993 | -4.644695 | 0.0000 |
| Limit_3:C(5) | -1.598777 | 0.484884 | -3.297237 | 0.0010 |
| Limit_4:C(6) | -1.028655 | 0.465433 | -2.210104 | 0.0271 |
| Limit_5:C(7) | -0.241152 | 0.445500 | -0.541307 | 0.5883 |
| Akaike info criterion | 11.74480 | Schwarz criterion | 11.99348 | |
| Log likelihood | -333.5993 | Hannan-Quinn crit. | 11.84167 | |
| Avg. log likelihood | -5.751712 | | | |

Note that the coefficients are labeled both with the identity of the limit point, and the coefficient number. Just below the limit points are the summary statistics for the equation.

Estimation Problems

Most of the previous discussion of estimation problems for binary models ([“Estimation Problems” on page 215](#)) also holds for ordered models. In general, these models are well-behaved and will require little intervention.

There are cases, however, where problems will arise. First, EViews currently has a limit of 750 total coefficients in an ordered dependent variable model. Thus, if you have 25 right-hand side variables, and a dependent variable with 726 distinct values, you will be unable to estimate your model using EViews.

Second, you may run into identification problems and estimation difficulties if you have some groups where there are very few observations. If necessary, you may choose to combine adjacent groups and re-estimate the model.

EViews may stop estimation with the message “Parameter estimates for limit points are non-ascending”, most likely on the first iteration. This error indicates that parameter values for the limit points were invalid, and that EViews was unable to adjust these values to make them valid. Make certain that if you are using user defined parameters, the limit points are strictly increasing. Better yet, we recommend that you employ the EViews starting values since they are based on a consistent first-stage estimation procedure, and should therefore be quite well-behaved.

Views of Ordered Equations

EViews provides you with several views of an ordered equation. As with other equations, you can examine the specification and estimated covariance matrix as well as perform Wald and likelihood ratio tests on coefficients of the model. In addition, there are several views that are specialized for the ordered model:

- **Dependent Variable Frequencies** — computes a one-way frequency table for the ordered dependent variable for the observations in the estimation sample. EViews presents both the frequency table and the cumulative frequency table in levels and percentages.
- **Expectation-Prediction Table** — classifies observations on the basis of the predicted response. EViews performs the classification on the basis of maximum predicted probability as well as the expected probability.

Dependent Variable: DANGER
 Method: ML - Ordered Probit
 Date: 09/13/97 Time: 10:00
 Sample(adjusted): 1 61
 Included observations: 58
 Excluded observations: 3 after adjusting endpoints
 Prediction table for ordered dependent variable

| Value | Count | Count of obs with Max Prob | Error | Sum of all Probabilities | Error |
|-------|-------|-------------------------------|-------|-----------------------------|--------|
| 1 | 18 | 27 | -9 | 18.571 | -0.571 |
| 2 | 14 | 16 | -2 | 13.417 | 0.583 |
| 3 | 10 | 0 | 10 | 9.163 | 0.837 |
| 4 | 9 | 8 | 1 | 8.940 | 0.060 |
| 5 | 7 | 7 | 0 | 7.909 | -0.909 |

There are two columns labeled “Error”. The first measures the difference between the observed count and the number of observations where the probability of that response is highest. For example, 18 individuals reported a value of 1 for DANGER, while 27 individuals had predicted probabilities that were highest for this value. The actual count minus the predicted is -9 . The second error column measures the difference between the actual number of individuals reporting the value, and the sum of all of the individual probabilities for that value.

Procedures for Ordered Equations

Make Ordered Limit Vector/Matrix

The full set of coefficients and the covariance matrix may be obtained from the estimated equation in the usual fashion (see [“Working With Equation Statistics” on page 16](#)). In some circumstances, however, you may wish to perform inference using only the estimates of the γ coefficients and the associated covariances.

The **Make Ordered Limit Vector** and **Make Ordered Limit Covariance Matrix** procedures provide a shortcut method of obtaining the estimates associated with the γ coefficients. The first procedure creates a vector (using the next unused name of the form LIMITS01, LIMITS02, etc.) containing the estimated γ coefficients. The latter procedure creates a symmetric matrix containing the estimated covariance matrix of the γ . The matrix will be given an unused name of the form VLIMITS01, VLIMITS02, etc., where the “V” is used to indicate that these are the variances of the estimated limit points.

Forecasting using Models

You cannot forecast directly from an estimated ordered model since the dependent variable represents categorical or rank data. EViews does, however, allow you to forecast the probability associated with each category. To forecast these probabilities, you must first create a model. Choose **Proc/Make Model** and EViews will open an untitled model window contain-

ing a system of equations, with a separate equation for the probability of each ordered response value.

To forecast from this model, simply click the Solve button in the model window toolbar. If you select Scenario 1 as your solution scenario, the default settings will save your results in a set of named series with “_1” appended to the end of the each underlying name. See [Chapter 36. “Models,” beginning on page 407](#) for additional detail on modifying and solving models.

For this example, the series I_DANGER_1 will contain the fitted linear index $x_i'\hat{\beta}$. The fitted probability of falling in category 1 will be stored as a series named DANGER_1_1, the fitted probability of falling in category 2 will be stored as a series named DANGER_2_1, and so on. Note that for each observation, the fitted probability of falling in each of the categories sums up to one.

Make Residual Series

The generalized residuals of the ordered model are the derivatives of the log likelihood with respect to a hypothetical unit- x variable. These residuals are defined to be uncorrelated with the explanatory variables of the model (see Chesher and Irish (1987), and Gourieroux, Monfort, Renault and Trognon (1987) for details), and thus may be used in a variety of specification tests.

To create a series containing the generalized residuals, select **View/Make Residual Series...**, enter a name or accept the default name, and click **OK**. The generalized residuals for an ordered model are given by:

$$e_{gi} = \frac{f(\gamma_{y_{i+1}} - x_i'\hat{\beta}) - f(\gamma_{y_i} - x_i'\hat{\beta})}{F(\gamma_{y_{i+1}} - x_i'\hat{\beta}) - F(\gamma_{y_i} - x_i'\hat{\beta})}, \quad (30.23)$$

where $\gamma_0 = -\infty$, and $\gamma_{M+1} = \infty$.

Censored Regression Models

In some settings, the dependent variable is only partially observed. For example, in survey data, data on incomes above a specified level are often top-coded to protect confidentiality. Similarly desired consumption on durable goods may be censored at a small positive or zero value. EViews provides tools to perform maximum likelihood estimation of these models and to use the results for further analysis.

Background

Consider the following latent variable regression model:

$$y_i^* = x_i'\beta + \sigma\epsilon_i, \quad (30.24)$$

where σ is a scale parameter. The scale parameter σ is identified in censored and truncated regression models, and will be estimated along with the β .

In the canonical *censored regression model*, known as the *tobit* (when there are normally distributed errors), the observed data y are given by:

$$y_i = \begin{cases} 0 & \text{if } y_i^* \leq 0 \\ y_i^* & \text{if } y_i^* > 0 \end{cases} \quad (30.25)$$

In other words, all negative values of y_i^* are coded as 0. We say that these data are *left censored* at 0. Note that this situation differs from a *truncated regression model* where negative values of y_i^* are dropped from the sample. More generally, EViews allows for both left and right censoring at arbitrary limit points so that:

$$y_i = \begin{cases} \underline{c}_i & \text{if } y_i^* \leq \underline{c}_i \\ y_i^* & \text{if } \underline{c}_i < y_i^* \leq \bar{c}_i \\ \bar{c}_i & \text{if } \bar{c}_i < y_i^* \end{cases} \quad (30.26)$$

where \underline{c}_i , \bar{c}_i are fixed numbers representing the censoring points. If there is no left censoring, then we can set $\underline{c}_i = -\infty$. If there is no right censoring, then $\bar{c}_i = \infty$. The canonical tobit model is a special case with $\underline{c}_i = 0$ and $\bar{c}_i = \infty$.

The parameters β , σ are estimated by maximizing the log likelihood function:

$$\begin{aligned} l(\beta, \sigma) = & \sum_{i=1}^N \log f((y_i - x_i' \beta) / \sigma) \cdot 1(\underline{c}_i < y_i < \bar{c}_i) \\ & + \sum_{i=1}^N \log(F((\underline{c}_i - x_i' \beta) / \sigma)) \cdot 1(y_i = \underline{c}_i) \\ & + \sum_{i=1}^N \log(1 - F((\bar{c}_i - x_i' \beta) / \sigma)) \cdot 1(y_i = \bar{c}_i) \end{aligned} \quad (30.27)$$

where f , F are the density and cumulative distribution functions of ϵ , respectively.

Estimating Censored Models in EViews

Consider the model:

$$\text{HRS}_i = \beta_1 + \beta_2 \text{AGE}_i + \beta_3 \text{EDU}_i + \beta_4 \text{KID}1_i + \epsilon_i, \quad (30.28)$$

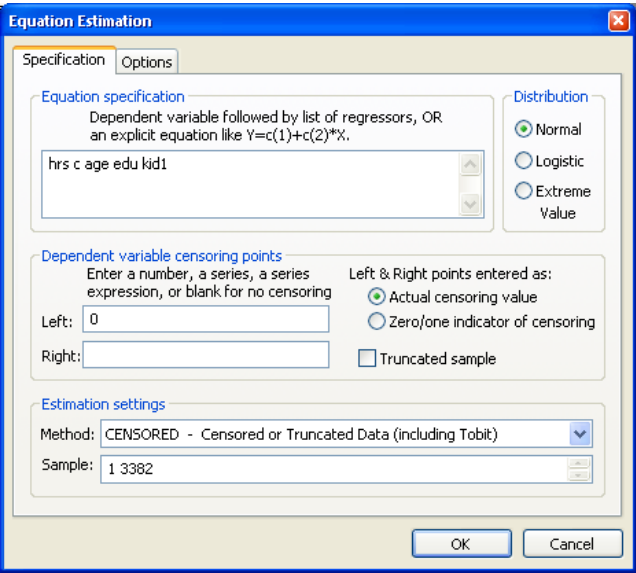
where hours worked (HRS) is left censored at zero. To estimate this model, select **Quick/Estimate Equation...** from the main menu. Then from the **Equation Estimation** dialog,

select the **CENSORED** estimation method. The dialog will change to provide a number of different input options.

Specifying the Regression Equation

In the **Equation specification** field, enter the name of the censored dependent variable followed by a list of regressors or an explicit expression for the equation. In our example, you will enter:

```
hrs c age edu kid1
```



Next, select one of the three distributions for the error term. EViews allows you three possible choices for the distribution of ϵ :

| | |
|------------------------|---|
| Standard normal | $E(\epsilon) = 0, \text{var}(\epsilon) = 1$ |
| Logistic | $E(\epsilon) = 0, \text{var}(\epsilon) = \pi^2/3$ |
| Extreme value (Type I) | $E(\epsilon) \approx -0.5772$ (Euler's constant), $\text{var}(\epsilon) = \pi^2/6$ |

Specifying the Censoring Points

You must also provide information about the censoring points of the dependent variable. There are two cases to consider: (1) where the limit points are known for all individuals,

and (2) where the censoring is by indicator and the limit points are known only for individuals with censored observations.

Limit Points Known

You should enter expressions for the left and right censoring points in the edit fields as required. Note that if you leave an edit field blank, EViews will assume that there is no censoring of observations of that type.

For example, in the canonical tobit model the data are censored on the left at zero, and are uncensored on the right. This case may be specified as:

Left edit field: 0

Right edit field: *[blank]*

Similarly, top-coded censored data may be specified as,

Left edit field: *[blank]*

Right edit field: 20000

while the more general case of left and right censoring is given by:

Left edit field: 10000

Right edit field: 20000

EViews also allows more general specifications where the censoring points are known to differ across observations. Simply enter the name of the series or auto-series containing the censoring points in the appropriate edit field. For example:

Left edit field: lowinc

Right edit field: vcens1+10

specifies a model with LOWINC censoring on the left-hand side, and right censoring at the value of VCENS1 + 10.

Limit Points Not Known

In some cases, the hypothetical censoring point is unknown for some individuals (c_i and \bar{c}_i are not observed for all observations). This situation often occurs with data where censoring is indicated with a zero-one dummy variable, but no additional information is provided about potential censoring points.

EViews provides you an alternative method of describing data censoring that matches this format. Simply select the **Field is zero/one indicator of censoring** option in the estimation dialog, and enter the series expression for the censoring indicator(s) in the appropriate edit field(s). Observations with a censoring indicator of one are assumed to be censored while those with a value of zero are assumed to be actual responses.

For example, suppose that we have observations on the length of time that an individual has been unemployed (U), but that some of these observations represent ongoing unemploy-

ment at the time the sample is taken. These latter observations may be treated as right censored at the reported value. If the variable RCENS is a dummy variable representing censoring, you can click on the **Field is zero/one indicator of censoring** setting and enter:

Left edit field: `[blank]`

Right edit field: `rcens`

in the edit fields. If the data are censored on both the left and the right, use separate binary indicators for each form of censoring:

Left edit field: `lcens`

Right edit field: `rcens`

where LCENS is also a binary indicator.

Once you have specified the model, click **OK**. EViews will estimate the parameters of the model using appropriate iterative techniques.

A Comparison of Censoring Methods

An alternative to specifying index censoring is to enter a very large positive or negative value for the censoring limit for non-censored observations. For example, you could enter “1e-100” and “1e100” as the censoring limits for an observation on a completed unemployment spell. In fact, any limit point that is “outside” the observed data will suffice.

While this latter approach will yield the same likelihood function and therefore the same parameter values and coefficient covariance matrix, there is a drawback to the artificial limit approach. The presence of a censoring value implies that it is possible to evaluate the conditional mean of the observed dependent variable, as well as the ordinary and standardized residuals. All of the calculations that use residuals will, however, be based upon the arbitrary artificial data and will be invalid.

If you specify your censoring by index, you are informing EViews that you do not have information about the censoring for those observations that are not censored. Similarly, if an observation is left censored, you may not have information about the right censoring limit. In these circumstances, you should specify your censoring by index so that EViews will prevent you from computing the conditional mean of the dependent variable and the associated residuals.

Interpreting the Output

If your model converges, EViews will display the estimation results in the equation window. The first part of the table presents the usual header information, including information about the assumed error distribution, estimation sample, estimation algorithms, and number of iterations required for convergence.

EViews also provides information about the specification for the censoring. If the estimated model is the canonical tobit with left-censoring at zero, EViews will label the method as a

TOBIT. For all other censoring methods, EViews will display detailed information about form of the left and/or right censoring.

Here, we have the header output from a left censored model where the censoring is specified by value:

```
Dependent Variable: Y_PT
Method: ML - Censored Normal (TOBIT)
Date: 09/14/97   Time: 08:27
Sample: 1 601
Included observations: 601
Convergence achieved after 8 iterations
Covariance matrix computed using second
derivatives
```

Below the header are the usual results for the coefficients, including the asymptotic standard errors, z -statistics, and significance levels. As in other limited dependent variable models, the estimated coefficients do not have a direct interpretation as the marginal effect of the associated regressor j for individual i , x_{ij} . In censored regression models, a change in x_{ij} has two effects: an effect on the mean of y , given that it is observed, and an effect on the probability of y being observed (see McDonald and Moffitt, 1980).

In addition to results for the regression coefficients, EViews reports an additional coefficient named SCALE, which is the estimated scale factor σ . This scale factor may be used to estimate the standard deviation of the residual, using the known variance of the assumed distribution. For example, if the estimated SCALE has a value of 0.446 for a model with extreme value errors, the implied standard error of the error term is $0.5977 = 0.466\pi/\sqrt{6}$.

Most of the other output is self-explanatory. As in the binary and ordered models above, EViews reports summary statistics for the dependent variable and likelihood based statistics. The regression statistics at the bottom of the table are computed in the usual fashion, using the residuals $\hat{\epsilon}_i = y_i - E(y_i|x_i, \hat{\beta}, \hat{\sigma})$ from the observed y .

Views of Censored Equations

Most of the views that are available for a censored regression are familiar from other settings. The residuals used in the calculations are defined below.

The one new view is the **Categorical Regressor Stats** view, which presents means and standard deviations for the dependent and independent variables for the estimation sample. EViews provides statistics computed over the entire sample, as well as for the left censored, right censored and non-censored individuals.

Procedures for Censored Equations

EViews provides several procedures which provide access to information derived from your censored equation estimates.

Make Residual Series

Select **Proc/Make Residual Series**, and select from among the three types of residuals. The three types of residuals for censored models are defined as:

| | |
|--------------|---|
| Ordinary | $e_{oi} = y_i - E(y_i x_i, \hat{\beta}, \hat{\sigma})$ |
| Standardized | $e_{si} = \frac{y_i - E(y_i x_i, \hat{\beta}, \hat{\sigma})}{\sqrt{\text{var}(y_i x_i, \hat{\beta}, \hat{\sigma})}}$ |
| Generalized | $e_{gi} = -\frac{f((\underline{c}_i - x_i' \hat{\beta}) / \hat{\sigma})}{\hat{\sigma} F((\underline{c}_i - x_i' \hat{\beta}) / \hat{\sigma})} \cdot 1(y_i \leq \underline{c}_i)$ $-\frac{f'((y_i - x_i' \hat{\beta}) / \hat{\sigma})}{\hat{\sigma} f((y_i - x_i' \hat{\beta}) / \hat{\sigma})} \cdot 1(\underline{c}_i < y_i < \bar{c}_i)$ $+ \frac{f((\bar{c}_i - x_i' \hat{\beta}) / \hat{\sigma})}{\hat{\sigma} (1 - F((\bar{c}_i - x_i' \hat{\beta}) / \hat{\sigma}))} \cdot 1(y_i \geq \bar{c}_i)$ |

where f , F are the density and distribution functions, and where 1 is an indicator function which takes the value 1 if the condition in parentheses is true, and 0 if it is false. All of the above terms will be evaluated at the estimated β and σ . See the discussion of forecasting for details on the computation of $E(y_i | x_i, \beta, \sigma)$.

The generalized residuals may be used as the basis of a number of LM tests, including LM tests of normality (see Lancaster, Chesher and Irish (1985), Chesher and Irish (1987), and Gourioux, Monfort, Renault and Trognon (1987); Greene (1997), provides a brief discussion and additional references).

Forecasting

EViews provides you with the option of forecasting the expected dependent variable, $E(y_i | x_i, \beta, \sigma)$, or the expected latent variable, $E(y_i^* | x_i, \beta, \sigma)$. Select **Forecast** from the equation toolbar to open the forecast dialog.

To forecast the expected *latent variable*, click on **Index - Expected latent variable**, and enter a name for the series to hold the output. The forecasts of the expected latent variable $E(y_i^* | x_i, \beta, \sigma)$ may be derived from the latent model using the relationship:

$$\hat{y}_i^* = E(y_i^* | x_i, \hat{\beta}, \hat{\sigma}) = x_i' \hat{\beta} - \hat{\sigma} \gamma. \quad (30.29)$$

where γ is the Euler-Mascheroni constant ($\gamma \approx 0.5772156649$).

To forecast the expected *observed dependent variable*, you should select **Expected dependent variable**, and enter a series name. These forecasts are computed using the relationship:

$$\begin{aligned}\hat{y}_i = & E(y_i | x_i, \hat{\beta}, \hat{\sigma}) = \underline{c}_i \cdot \Pr(y_i = \underline{c}_i | x_i, \hat{\beta}, \hat{\sigma}) \\ & + E(y_i^* | \underline{c}_i < y_i^* < \bar{c}_i; x_i, \hat{\beta}, \hat{\sigma}) \cdot \Pr(\underline{c}_i < y_i^* < \bar{c}_i | x_i, \hat{\beta}, \hat{\sigma}) \\ & + \bar{c}_i \cdot \Pr(y_i = \bar{c}_i | x_i, \hat{\beta}, \hat{\sigma})\end{aligned}\quad (30.30)$$

Note that these forecasts always satisfy $\underline{c}_i \leq \hat{y}_i \leq \bar{c}_i$. The probabilities associated with being in the various classifications are computed by evaluating the cumulative distribution function of the specified distribution. For example, the probability of being at the lower limit is given by:

$$\Pr(y_i = \underline{c}_i | x_i, \hat{\beta}, \hat{\sigma}) = \Pr(y_i^* \leq \underline{c}_i | x_i, \hat{\beta}, \hat{\sigma}) = F((\underline{c}_i - x_i' \hat{\beta}) / \hat{\sigma}). \quad (30.31)$$

Censored Model Example

As an example, we replicate Fair's (1978) tobit model that estimates the incidence of extra-marital affairs. The dependent variable, number of extramarital affairs (Y_PT), is left censored at zero and the errors are assumed to be normally distributed. The bottom portion of the output is presented below:

| | Coefficient | Std. Error | z-Statistic | Prob. |
|---------------------|-------------|-----------------------|-------------|--------|
| C | 7.608487 | 3.905837 | 1.947979 | 0.0514 |
| Z1 | 0.945787 | 1.062824 | 0.889881 | 0.3735 |
| Z2 | -0.192698 | 0.080965 | -2.380015 | 0.0173 |
| Z3 | 0.533190 | 0.146602 | 3.636997 | 0.0003 |
| Z4 | 1.019182 | 1.279524 | 0.796532 | 0.4257 |
| Z5 | -1.699000 | 0.405467 | -4.190231 | 0.0000 |
| Z6 | 0.025361 | 0.227658 | 0.111399 | 0.9113 |
| Z7 | 0.212983 | 0.321145 | 0.663198 | 0.5072 |
| Z8 | -2.273284 | 0.415389 | -5.472657 | 0.0000 |
| | | | | |
| | Error | Distribution | | |
| SCALE:C(10) | 8.258432 | 0.554534 | 14.89256 | 0.0000 |
| R-squared | 0.151569 | Mean dependent var | 1.455907 | |
| Adjusted R-squared | 0.138649 | S.D. dependent var | 3.298758 | |
| S.E. of regression | 3.061544 | Akaike info criterion | 2.378473 | |
| Sum squared resid | 5539.472 | Schwarz criterion | 2.451661 | |
| Log likelihood | -704.7311 | Hannan-Quinn criter. | 2.406961 | |
| Avg. log likelihood | -1.172597 | | | |
| | | | | |
| Left censored obs | 451 | Right censored obs | 0 | |
| Uncensored obs | 150 | Total obs | 601 | |

Tests of Significance

EViews does not, by default, provide you with the usual likelihood ratio test of the overall significance for the tobit and other censored regression models. There are several ways to perform this test (or an asymptotically equivalent test).

First, you can use the built-in coefficient testing procedures to test the exclusion of all of the explanatory variables. Select the redundant variables test and enter the names of all of the explanatory variables you wish to exclude. EViews will compute the appropriate likelihood ratio test statistic and the p -value associated with the statistic.

To take an example, suppose we wish to test whether the variables in the Fair tobit, above, contribute to the fit of the model. Select **View/Coefficient Tests/Redundant Variables - Likelihood Ratio...** and enter all of the explanatory variables:

```
z1 z2 z3 z4 z5 z6 z7 z8
```

EViews will estimate the restricted model for you and compute the LR statistic and p -value. In this case, the value of the test statistic is 80.01, which for eight degrees of freedom, yields a p -value of less than 0.000001.

Alternatively, you could test the restriction using the Wald test by selecting **View/Coefficient Tests/Wald - Coefficient Restrictions...**, and entering the restriction that:

```
c (2) = c (3) = c (4) = c (5) = c (6) = c (7) = c (8) = c (9) = 0
```

The reported statistic is 68.14, with a p -value of less than 0.000001.

Lastly, we demonstrate the direct computation of the LR test. Suppose the Fair tobit model estimated above is saved in the named equation EQ_TOBIT. Then you could estimate an equation containing only a constant, say EQ_RESTR, and place the likelihood ratio statistic in a scalar:

```
scalar lrstat=-2*(eq_restr.@logl-eq_tobit.@logl)
```

Next, evaluate the chi-square probability associated with this statistic:

```
scalar lrprob=1-@cchisq(lrstat, 8)
```

with degrees of freedom given by the number of coefficient restrictions in the constant only model. You can double click on the LRSTAT icon or the LRPROB icon in the workfile window to display the results in the status line.

A Specification Test for the Tobit

As a rough diagnostic check, Pagan and Vella (1989) suggest plotting Powell's (1986) symmetrically trimmed residuals. If the error terms have a symmetric distribution centered at zero (as assumed by the normal distribution), so should the trimmed residuals. To construct the trimmed residuals, first save the forecasts of the index (expected latent variable): click **Forecast**, choose **Index-Expected latent variable**, and provide a name for the fitted index, say XB. The trimmed residuals are obtained by dropping observations for which $x_i'\hat{\beta} < 0$, and replacing y_i with $2(x_i'\hat{\beta})$ for all observations where $y_i < 2(x_i'\hat{\beta})$. The trimmed residuals RES_T can be obtained by using the commands:

```
series res_t=(y_pt<=2*xb)*(y_pt-xb) +(y_pt>2*xb)*xb
```

```

smpl if xb<0
series res_t=na
smpl @all

```

The histogram of the trimmed residual is depicted below.

This example illustrates the possibility that the number of observations that are lost by trimming can be quite large; out of the 601 observations in the sample, only 47 observations are left after trimming.

The tobit model imposes the restriction that the coefficients that determine the probability of being censored are the same as those that determine the con-

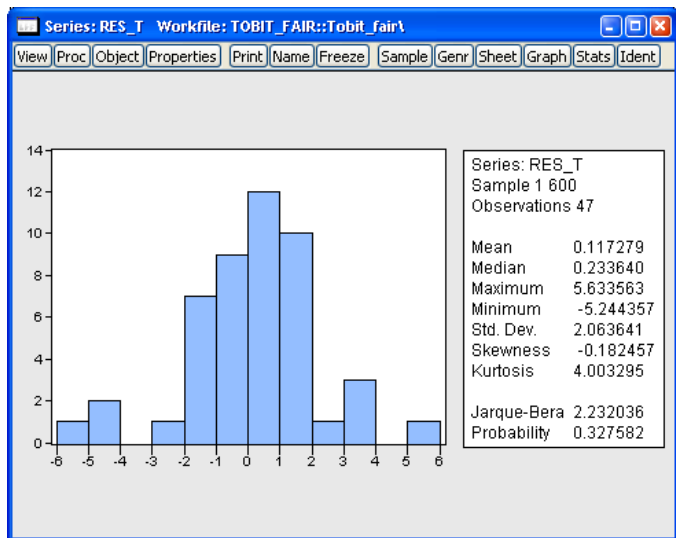
ditional mean of the uncensored observations. To test this restriction, we carry out the LR test by comparing the (restricted) tobit to the unrestricted log likelihood that is the sum of a probit and a truncated regression (we discuss truncated regression in detail in the following section). Save the tobit equation in the workfile by pressing the **Name** button, and enter a name, say EQ_TOBIT.

To estimate the probit, first create a dummy variable indicating uncensored observations by the command:

```
series y_c = (y_pt>0)
```

Then estimate a probit by replacing the dependent variable Y_PT by Y_C. A simple way to do this is to press **Object/Copy Object...** from the tobit equation toolbar. From the new untitled equation window that appears, press **Estimate**, replace the dependent variable with Y_C and choose **Method: BINARY** and click **OK**. Save the probit equation by pressing the **Name** button, say as EQ_BIN.

To estimate the truncated model, press **Object/Copy Object...** from the tobit equation toolbar again. From the new untitled equation window that appears, press **Estimate**, mark the **Truncated sample** option, and click **OK**. Save the truncated regression by pressing the **Name** button, say as EQ_TR.



Then the LR test statistic and its p -value can be saved as a scalar by the commands:

```
scalar lr_test=2*(eq_bin.@logl+eq_tr.@logl-eq_tobit.@logl)
scalar lr_pval=1-@cchisq(lr_test,eq_tobit.@ncoef)
```

Double click on the scalar name to display the value in the status line at the bottom of the EViews window. For the example data set, the p -value is 0.066, which rejects the tobit model at the 10% level, but not at the 5% level.

For other specification tests for the tobit, see Greene (1997, 20.3.4) or Pagan and Vella (1989).

Truncated Regression Models

A close relative of the censored regression model is the truncated regression model. Suppose that an observation is not observed whenever the dependent variable falls below one threshold, or exceeds a second threshold. This sampling rule occurs, for example, in earnings function studies for low-income families that exclude observations with incomes above a threshold, and in studies of durables demand among individuals who purchase durables.

The general two-limit truncated regression model may be written as:

$$y_i^* = x_i' \beta + \sigma \epsilon_i \quad (30.32)$$

where $y_i = y_i^*$ is only observed if:

$$\underline{c}_i < y_i^* < \bar{c}_i. \quad (30.33)$$

If there is no lower truncation, then we can set $\underline{c}_i = -\infty$. If there is no upper truncation, then we set $\bar{c}_i = \infty$.

The log likelihood function associated with these data is given by:

$$\begin{aligned} l(\beta, \sigma) = & \sum_{i=1}^N \log f((y_i - x_i' \beta) / \sigma) \cdot 1(\underline{c}_i < y_i < \bar{c}_i) \\ & - \sum_{i=1}^N \log (F((\bar{c}_i - x_i' \beta) / \sigma) - F((\underline{c}_i - x_i' \beta) / \sigma)). \end{aligned} \quad (30.34)$$

The likelihood function is maximized with respect to β and σ , using standard iterative methods.

Estimating a Truncated Model in EViews

Estimation of a truncated regression model follows the same steps as estimating a censored regression:

- Select **Quick/Estimate Equation...** from the main menu, and in the Equation Specification dialog, select the **CENSORED** estimation method. The censored and truncated regression dialog will appear.
- Enter the name of the truncated dependent variable and the list of the regressors or provide explicit expression for the equation in the **Equation Specification** field, and select one of the three distributions for the error term.
- Indicate that you wish to estimate the truncated model by checking the **Truncated sample** option.
- Specify the truncation points of the dependent variable by entering the appropriate expressions in the two edit fields. If you leave an edit field blank, EViews will assume that there is no truncation along that dimension.

You should keep a few points in mind. First, truncated estimation is only available for models where the truncation points are known, since the likelihood function is not otherwise defined. If you attempt to specify your truncation points by index, EViews will issue an error message indicating that this selection is not available.

Second, EViews will issue an error message if any values of the dependent variable are outside the truncation points. Furthermore, EViews will automatically exclude any observations that are exactly equal to a truncation point. Thus, if you specify zero as the lower truncation limit, EViews will issue an error message if any observations are less than zero, and will exclude any observations where the dependent variable exactly equals zero.

The cumulative distribution function and density of the assumed distribution will be used to form the likelihood function, as described above.

Procedures for Truncated Equations

EViews provides the same procedures for truncated equations as for censored equations. The residual and forecast calculations differ to reflect the truncated dependent variable and the different likelihood function.

Make Residual Series

Select **Proc/Make Residual Series**, and select from among the three types of residuals. The three types of residuals for censored models are defined as:

| | |
|----------|---|
| Ordinary | $e_{oi} = y_i - E(y_i^* \underline{c}_i < y_i^* < \bar{c}_i; x_i, \hat{\beta}, \hat{\sigma})$ |
|----------|---|

| | |
|--------------|---|
| Standardized | $e_{si} = \frac{y_i - E(y_i^* \mathcal{L}_i < y_i^* < \bar{c}_i; x_i, \hat{\beta}, \hat{\sigma})}{\sqrt{\text{var}(y_i^* \mathcal{L}_i < y_i^* < \bar{c}_i; x_i, \hat{\beta}, \hat{\sigma})}}$ |
| Generalized | $e_{gi} = - \frac{f'((y_i - x_i' \hat{\beta}) / \hat{\sigma})}{\sigma f((y_i - x_i' \hat{\beta}) / \hat{\sigma})} - \frac{f((\bar{c}_i - x_i' \hat{\beta}) / \hat{\sigma}) - f((\mathcal{L}_i - x_i' \hat{\beta}) / \hat{\sigma})}{\sigma (F((\bar{c}_i - x_i' \hat{\beta}) / \hat{\sigma}) - F((\mathcal{L}_i - x_i' \hat{\beta}) / \hat{\sigma}))}$ |

where f , F , are the density and distribution functions. Details on the computation of $E(y_i | \mathcal{L}_i < y_i < \bar{c}_i; x_i, \hat{\beta}, \hat{\sigma})$ are provided below.

The generalized residuals may be used as the basis of a number of LM tests, including LM tests of normality (see Chesher and Irish (1984, 1987), and Gouriéroux, Monfort and Trognon (1987); Greene (1997) provides a brief discussion and additional references).

Forecasting

EViews provides you with the option of forecasting the expected observed dependent variable, $E(y_i | x_i, \hat{\beta}, \hat{\sigma})$, or the expected latent variable, $E(y_i^* | x_i, \hat{\beta}, \hat{\sigma})$.

To forecast the expected latent variable, select **Forecast** from the equation toolbar to open the forecast dialog, click on **Index - Expected latent variable**, and enter a name for the series to hold the output. The forecasts of the expected latent variable $E(y_i^* | x_i, \hat{\beta}, \hat{\sigma})$ are computed using:

$$\hat{y}_i^* = E(y_i^* | x_i, \hat{\beta}, \hat{\sigma}) = x_i' \hat{\beta} - \hat{\sigma} \gamma. \quad (30.35)$$

where γ is the Euler-Mascheroni constant ($\gamma \approx 0.5772156649$).

To forecast the expected observed dependent variable for the truncated model, you should select **Expected dependent variable**, and enter a series name. These forecasts are computed using:

$$\hat{y}_i = E(y_i^* | \mathcal{L}_i < y_i^* < \bar{c}_i; x_i, \hat{\beta}, \hat{\sigma}) \quad (30.36)$$

so that the expectations for the latent variable are taken with respect to the conditional (on being observed) distribution of the y_i^* . Note that these forecasts always satisfy the inequality $\mathcal{L}_i < \hat{y}_i < \bar{c}_i$.

It is instructive to compare this latter expected value with the expected value derived for the censored model in [Equation \(30.30\)](#) above (repeated here for convenience):

$$\begin{aligned}\hat{y}_i &= E(y_i | x_i, \hat{\beta}, \hat{\sigma}) = \underline{\varepsilon}_i \cdot \Pr(y_i = \underline{\varepsilon}_i | x_i, \hat{\beta}, \hat{\sigma}) \\ &+ E(y_i^* | \underline{\varepsilon}_i < y_i^* < \bar{c}_i; x_i, \hat{\beta}, \hat{\sigma}) \cdot \Pr(\underline{\varepsilon}_i < y_i^* < \bar{c}_i | x_i, \hat{\beta}, \hat{\sigma}) \\ &+ \bar{c}_i \cdot \Pr(y_i = \bar{c}_i | x_i, \hat{\beta}, \hat{\sigma}).\end{aligned}\quad (30.37)$$

The expected value of the dependent variable for the truncated model is the first part of the middle term of the censored expected value. The differences between the two expected values (the probability weight and the first and third terms) reflect the different treatment of latent observations that do not lie between $\underline{\varepsilon}_i$ and \bar{c}_i . In the censored case, those observations are included in the sample and are accounted for in the expected value. In the truncated case, data outside the interval are not observed and are not used in the expected value computation.

An Illustration

As an example, we reestimate the Fair tobit model from above, truncating the data so that observations at or below zero are removed from the sample. The output from truncated estimation of the Fair model is presented below:

Dependent Variable: Y_PT
Method: ML - Censored Normal (TOBIT)
Date: 10/13/97 Time: 22:45
Sample(adjusted): 452 601
Included observations: 150 after adjusting endpoints
Truncated sample
Left censoring (value) at zero
Convergence achieved after 8 iterations
Covariance matrix computed using second derivatives

| | Coefficient | Std. Error | z-Statistic | Prob. |
|----|-------------|------------|-------------|--------|
| C | 12.37288 | 5.178306 | 2.389368 | 0.0169 |
| Z1 | -1.336872 | 1.453133 | -0.919993 | 0.3576 |
| Z2 | -0.044792 | 0.116141 | -0.385670 | 0.6997 |
| Z3 | 0.544182 | 0.220119 | 2.472218 | 0.0134 |
| Z4 | -2.142896 | 1.787720 | -1.198675 | 0.2307 |
| Z5 | -1.423128 | 0.600472 | -2.370014 | 0.0178 |
| Z6 | -0.316721 | 0.322327 | -0.982609 | 0.3258 |
| Z7 | 0.621428 | 0.478827 | 1.297813 | 0.1944 |
| Z8 | -1.210037 | 0.552131 | -2.191578 | 0.0284 |

| Error Distribution | | | | |
|---------------------|-----------|-----------------------|----------|--------|
| SCALE:C(10) | 5.379557 | 0.688875 | 7.809196 | 0.0000 |
| R-squared | 0.654664 | Mean dependent var | 1.455907 | |
| Adjusted R-squared | 0.649405 | S.D. dependent var | 3.298758 | |
| S.E. of regression | 1.953229 | Akaike info criterion | 1.333891 | |
| Sum squared resid | 2254.726 | Schwarz criterion | 1.407079 | |
| Log likelihood | -390.8342 | Hannan-Quinn criter. | 1.362379 | |
| Avg. log likelihood | -0.650306 | | | |
| Left censored obs | 0 | Right censored obs | 0 | |
| Uncensored obs | 150 | Total obs | 150 | |

Note that the header information indicates that the model is a truncated specification, and that the sample information at the bottom of the screen shows that there are no left and right censored observations.

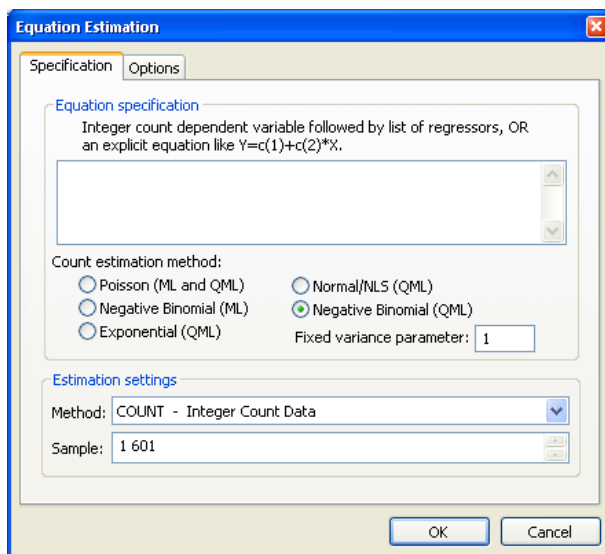
Count Models

Count models are employed when y takes integer values that represent the number of events that occur—examples of count data include the number of patents filed by a company, and the number of spells of unemployment experienced over a fixed time interval.

EViews provides support for the estimation of several models of count data. In addition to the standard poisson and negative binomial maximum likelihood (ML) specifications, EViews provides a number of quasi-maximum likelihood (QML) estimators for count data.

Estimating Count Models in EViews

To estimate a count data model, select **Quick/Estimate Equation...** from the main menu, and select **COUNT** as the estimation method. EViews displays the count estimation dialog into which you will enter the dependent and explanatory variable regressors, select a type of count model, and if desired, set estimation options.



There are three parts to the specification of the count model:

- In the upper edit field, you should list the dependent variable and the independent variables or you should provide an explicit expression for the index. The list of explanatory variables specifies a model for the conditional mean of the dependent variable:

$$m(x_i, \beta) = E(y_i | x_i, \beta) = \exp(x_i' \beta). \quad (30.38)$$

- Next, click on **Options** and, if desired, change the default estimation algorithm, convergence criterion, starting values, and method of computing the coefficient covariance.
- Lastly, select one of the entries listed under count estimation method, and if appropriate, specify a value for the variance parameter. Details for each method are provided in the following discussion.

Poisson Model

For the Poisson model, the conditional density of y_i given x_i is:

$$f(y_i | x_i, \beta) = e^{-m(x_i, \beta)} m(x_i, \beta)^{y_i} / y_i! \quad (30.39)$$

where y_i is a non-negative integer valued random variable. The maximum likelihood estimator (MLE) of the parameter β is obtained by maximizing the log likelihood function:

$$l(\beta) = \sum_{i=1}^N y_i \log m(x_i, \beta) - m(x_i, \beta) - \log(y_i!). \quad (30.40)$$

Provided the conditional mean function is correctly specified and the conditional distribution of y is Poisson, the MLE $\hat{\beta}$ is consistent, efficient, and asymptotically normally distributed, with variance matrix consistently estimated by:

$$V = \text{var}(\hat{\beta}) = \left(\sum_{i=1}^N \left(\frac{\partial \hat{m}_i}{\partial \beta} \frac{\partial \hat{m}_i}{\partial \beta'} \right) / \hat{m}_i \right)^{-1} \quad (30.41)$$

where $\hat{m}_i = m(x_i, \hat{\beta})$.

The Poisson assumption imposes restrictions that are often violated in empirical applications. The most important restriction is the equality of the (conditional) mean and variance:

$$v(x_i, \beta) = \text{var}(y_i | x_i, \beta) = E(y_i | x_i, \beta) = m(x_i, \beta). \quad (30.42)$$

If the mean-variance equality does not hold, the model is misspecified. EViews provides a number of other estimators for count data which relax this restriction.

We note here that the Poisson estimator may also be interpreted as a quasi-maximum likelihood estimator. The implications of this result are discussed below.

Negative Binomial (ML)

One common alternative to the Poisson model is to estimate the parameters of the model using maximum likelihood of a negative binomial specification. The log likelihood for the negative binomial distribution is given by:

$$\begin{aligned}
l(\beta, \eta) = & \sum_{i=1}^N y_i \log(\eta^2 m(x_i, \beta)) - (y_i + 1/\eta^2) \log(1 + \eta^2 m(x_i, \beta)) \\
& + \log \Gamma(y_i + 1/\eta^2) - \log(y_i!) - \log \Gamma(1/\eta^2)
\end{aligned} \tag{30.43}$$

where η^2 is a variance parameter to be jointly estimated with the conditional mean parameters β . EViews estimates the log of η^2 , and labels this parameter as the “SHAPE” parameter in the output. Standard errors are computed using the inverse of the information matrix.

The negative binomial distribution is often used when there is *overdispersion* in the data, so that $v(x_i, \beta) > m(x_i, \beta)$, since the following moment conditions hold:

$$\begin{aligned}
E(y_i | x_i, \beta) &= m(x_i, \beta) \\
\text{var}(y_i | x_i, \beta) &= m(x_i, \beta)(1 + \eta^2 m(x_i, \beta))
\end{aligned} \tag{30.44}$$

η^2 is therefore a measure of the extent to which the conditional variance exceeds the conditional mean.

Consistency and efficiency of the negative binomial ML requires that the conditional distribution of y be negative binomial.

Quasi-maximum Likelihood (QML)

We can perform maximum likelihood estimation under a number of alternative distributional assumptions. These *quasi-maximum likelihood (QML) estimators* are robust in the sense that they produce consistent estimates of the parameters of a correctly specified conditional mean, even if the distribution is incorrectly specified.

This robustness result is exactly analogous to the situation in ordinary regression, where the normal ML estimator (least squares) is consistent, even if the underlying error distribution is not normally distributed. In ordinary least squares, all that is required for consistency is a correct specification of the conditional mean $m(x_i, \beta) = x_i' \beta$. For QML count models, all that is required for consistency is a correct specification of the conditional mean $m(x_i, \beta)$.

The estimated standard errors computed using the inverse of the information matrix will not be consistent unless the conditional distribution of y is correctly specified. However, it is possible to estimate the standard errors in a robust fashion so that we can conduct valid inference, even if the distribution is incorrectly specified.

EViews provides options to compute two types of robust standard errors. Click **Options** in the Equation Specification dialog box and mark the **Robust Covariance** option. The **Huber/White** option computes QML standard errors, while the **GLM** option computes standard errors corrected for overdispersion. See “[Technical Notes](#)” on page 255 for details on these options.

Further details on QML estimation are provided by Gourioux, Monfort, and Trognon (1994a, 1994b). Wooldridge (1996) provides an excellent summary of the use of QML techniques in estimating parameters of count models. See also the extensive related literature on Generalized Linear Models (McCullagh and Nelder, 1989).

Poisson

The Poisson MLE is also a QMLE for data from alternative distributions. Provided that the conditional mean is correctly specified, it will yield consistent estimates of the parameters β of the mean function. By default, EViews reports the ML standard errors. If you wish to compute the QML standard errors, you should click on **Options**, select **Robust Covariances**, and select the desired covariance matrix estimator.

Exponential

The log likelihood for the exponential distribution is given by:

$$l(\beta) = \sum_{i=1}^N -\log m(x_i, \beta) - y_i / (m(x_i, \beta)). \quad (30.45)$$

As with the other QML estimators, the exponential QMLE is consistent even if the conditional distribution of y_i is not exponential, provided that m_i is correctly specified. By default, EViews reports the robust QML standard errors.

Normal

The log likelihood for the normal distribution is:

$$l(\beta) = \sum_{i=1}^N -\frac{1}{2} \left(\frac{y_i - m(x_i, \beta)}{\sigma} \right)^2 - \frac{1}{2} \log(\sigma^2) - \frac{1}{2} \log(2\pi). \quad (30.46)$$

For *fixed* σ^2 and correctly specified m_i , maximizing the normal log likelihood function provides consistent estimates even if the distribution is not normal. Note that maximizing the normal log likelihood for a fixed σ^2 is equivalent to minimizing the sum of squares for the nonlinear regression model:

$$y_i = m(x_i, \beta) + \epsilon_i. \quad (30.47)$$

EViews sets $\sigma^2 = 1$ by default. You may specify any other (positive) value for σ^2 by changing the number in the **Fixed variance parameter** field box. By default, EViews reports the robust QML standard errors when estimating this specification.

Negative Binomial

If we maximize the negative binomial log likelihood, given above, for *fixed* η^2 , we obtain the QMLE of the conditional mean parameters β . This QML estimator is consistent even if the conditional distribution of y is not negative binomial, provided that m_i is correctly specified.

EViews sets $\eta^2 = 1$ by default, which is a special case known as the geometric distribution. You may specify any other (positive) value by changing the number in the **Fixed variance parameter** field box. For the negative binomial QMLE, EViews by default reports the robust QMLE standard errors.

Views of Count Models

EViews provides a full complement of views of count models. You can examine the estimation output, compute frequencies for the dependent variable, view the covariance matrix, or perform coefficient tests. Additionally, you can select **View/Actual, Fitted, Residual...** and pick from a number of views describing the ordinary residuals $e_{oi} = y_i - m(x_i, \hat{\beta})$, or you can examine the correlogram and histogram of these residuals. For the most part, all of these views are self-explanatory.

Note, however, that the LR test statistics presented in the summary statistics at the bottom of the equation output, or as computed under the **View/Coefficient Tests/Redundant Variables - Likelihood Ratio...** have a known asymptotic distribution only if the conditional distribution is correctly specified. Under the weaker GLM assumption that the true variance is proportional to the nominal variance, we can form a quasi-likelihood ratio, $QLR = LR/\hat{\sigma}^2$, where $\hat{\sigma}^2$ is the estimated proportional variance factor. This QLR statistic has an asymptotic χ^2 distribution under the assumption that the mean is correctly specified and that the variances follow the GLM structure. EViews does not compute the QLR statistic, but it can be estimated by computing an estimate of $\hat{\sigma}^2$ based upon the standardized residuals. We provide an example of the use of the QLR test statistic below.

If the GLM assumption does not hold, then there is no usable QLR test statistic with a known distribution; see Wooldridge (1996).

Procedures for Count Models

Most of the procedures are self-explanatory. Some details are required for the forecasting and residual creation procedures.

- **Forecast...** provides you the option to forecast the dependent variable y_i or the predicted linear index $x_i\beta$. Note that for all of these models the forecasts of y_i are given by $\hat{y}_i = m(x_i, \hat{\beta})$ where $m(x_i, \beta) = \exp(x_i\beta)$.
- **Make Residual Series...** provides the following three types of residuals for count models:

| | |
|----------|--------------------------------------|
| Ordinary | $e_{oi} = y_i - m(x_i, \hat{\beta})$ |
|----------|--------------------------------------|

| | |
|------------------------|---|
| Standardized (Pearson) | $e_{si} = \frac{y_i - m(x_i, \hat{\beta})}{\sqrt{v(x_i, \hat{\beta}, \hat{\gamma})}}$ |
| Generalized | $e_g = (\text{varies})$ |

where the γ represents any additional parameters in the variance specification. Note that the specification of the variances may vary significantly between specifications. For example, the Poisson model has $v(x_i, \hat{\beta}) = m(x_i, \hat{\beta})$, while the exponential has $v(x_i, \hat{\beta}) = m(x_i, \hat{\beta})^2$.

The generalized residuals can be used to obtain the score vector by multiplying the generalized residuals by each variable in x . These scores can be used in a variety of LM or conditional moment tests for specification testing; see Wooldridge (1996).

Demonstrations

A Specification Test for Overdispersion

Consider the model:

$$\text{NUMB}_i = \beta_1 + \beta_2 \text{IP}_i + \beta_3 \text{FEB}_i + \epsilon_i, \quad (30.48)$$

where the dependent variable NUMB is the number of strikes, IP is a measure of industrial production, and FEB is a February dummy variable, as reported in Kennan (1985, Table 1).

The results from Poisson estimation of this model are presented below:

Dependent Variable: NUMB
 Method: ML/QML - Poisson Count (Quadratic hill climbing)
 Date: 09/14/97 Time: 10:58
 Sample: 1 103
 Included observations: 103
 Convergence achieved after 4 iterations
 Covariance matrix computed using second derivatives

| Variable | Coefficient | Std. Error | z-Statistic | Prob. |
|-----------------------|-------------|-----------------------|-------------|--------|
| C | 1.725630 | 0.043656 | 39.52764 | 0.0000 |
| IP | 2.775334 | 0.819104 | 3.388254 | 0.0007 |
| FEB | -0.377407 | 0.174520 | -2.162539 | 0.0306 |
| R-squared | 0.064502 | Mean dependent var | 5.495146 | |
| Adjusted R-squared | 0.045792 | S.D. dependent var | 3.653829 | |
| S.E. of regression | 3.569190 | Akaike info criterion | 5.583421 | |
| Sum squared resid | 1273.912 | Schwarz criterion | 5.660160 | |
| Log likelihood | -284.5462 | Hannan-Quinn criter. | 5.614503 | |
| Restr. log likelihood | -292.9694 | Avg. log likelihood | -2.762584 | |
| LR statistic (2 df) | 16.84645 | LR index (Pseudo-R2) | 0.028751 | |
| Probability(LR stat) | 0.000220 | | | |

Cameron and Trivedi (1990) propose a regression based test of the Poisson restriction $v(x_i, \beta) = m(x_i, \beta)$. To carry out the test, first estimate the Poisson model and obtain the fitted values of the dependent variable. Click **Forecast** and provide a name for the forecasted dependent variable, say NUMB_F. The test is based on an auxiliary regression of $e_{oi}^2 - y_i$ on \hat{y}_i^2 and testing the significance of the regression coefficient. For this example, the test regression can be estimated by the command:

```
ls (numb-numb_f)^2-numb numb_f^2
```

yielding the following results:

Dependent Variable: (NUMB-NUMB_F)^2-NUMB

Method: Least Squares

Date: 01/02/07 Time: 16:28

Sample: 1 103

Included observations: 103

| Variable | Coefficient | Std. Error | t-Statistic | Prob. |
|--------------------|-------------|-----------------------|-------------|----------|
| NUMB_F^2 | 0.238874 | 0.052115 | 4.583571 | 0.0000 |
| R-squared | 0.043930 | Mean dependent var | | 6.872929 |
| Adjusted R-squared | 0.043930 | S.D. dependent var | | 17.65726 |
| S.E. of regression | 17.26506 | Akaike info criterion | | 8.544908 |
| Sum squared resid | 30404.41 | Schwarz criterion | | 8.570488 |
| Log likelihood | -439.0628 | Durbin-Watson stat | | 1.711805 |

The t -statistic of the coefficient is highly significant, leading us to reject the Poisson restriction. Moreover, the estimated coefficient is significantly positive, indicating overdispersion in the residuals.

An alternative approach, suggested by Wooldridge (1996), is to regress $e_{si} - 1$, on \hat{y}_i . To perform this test, select **Proc/Make Residual Series...** and select **Standardized**. Save the results in a series, say SRESID. Then estimating the regression specification:

```
sresid^2-1 numbf
```

yields the results:

Dependent Variable: SRESID^2-1

Method: Least Squares

Date: 01/02/07 Time: 16:29

Sample: 1 103

Included observations: 103

| Variable | Coefficient | Std. Error | t-Statistic | Prob. |
|--------------------|-------------|-----------------------|-------------|----------|
| NUMBF | 0.221238 | 0.055002 | 4.022326 | 0.0001 |
| R-squared | 0.017556 | Mean dependent var | | 1.161573 |
| Adjusted R-squared | 0.017556 | S.D. dependent var | | 3.138974 |
| S.E. of regression | 3.111299 | Akaike info criterion | | 5.117619 |
| Sum squared resid | 987.3786 | Schwarz criterion | | 5.143199 |
| Log likelihood | -262.5574 | Durbin-Watson stat | | 1.764537 |

Both tests suggest the presence of overdispersion, with the variance approximated by roughly $v = m(1 + 0.23m)$.

Given the evidence of overdispersion and the rejection of the Poisson restriction, we will re-estimate the model, allowing for mean-variance inequality. Our approach will be to estimate the two-step negative binomial QMLE specification (termed the *quasi-generalized pseudo-maximum likelihood estimator* by Gourieroux, Monfort, and Trognon (1984a, b)) using the estimate of $\hat{\eta}^2$ from the Wooldridge test derived above. To compute this estimator, simply select **Negative Binomial (QML)** and enter 0.22124 in the edit field for **Fixed variance parameter**.

We will use the GLM variance calculations, so you should click on **Option** in the Equation Specification dialog and mark the **Robust Covariance** and **GLM** options. The estimation results are shown below:

| Dependent Variable: NUMB | | | | |
|---|-------------|-----------------------|-------------|-----------|
| Method: QML - Negative Binomial Count (Quadratic hill climbing) | | | | |
| Date: 01/02/07 Time: 16:32 | | | | |
| Sample: 1 103 | | | | |
| Included observations: 103 | | | | |
| QML parameter used in estimation: 0.22124 | | | | |
| Convergence achieved after 3 iterations | | | | |
| GLM Robust Standard Errors & Covariance | | | | |
| Variance factor estimate = 0.9899965097 | | | | |
| Covariance matrix computed using second derivatives | | | | |
| Variable | Coefficient | Std. Error | z-Statistic | Prob. |
| C | 1.724906 | 0.064976 | 26.54671 | 0.0000 |
| IP | 2.833103 | 1.216260 | 2.329357 | 0.0198 |
| FEB | -0.369558 | 0.239125 | -1.545463 | 0.1222 |
| R-squared | 0.064374 | Mean dependent var | | 5.495146 |
| Adjusted R-squared | 0.045661 | S.D. dependent var | | 3.653829 |
| S.E. of regression | 3.569435 | Akaike info criterion | | 5.174385 |
| Sum squared resid | 1274.087 | Schwarz criterion | | 5.251125 |
| Log likelihood | -263.4808 | Hannan-Quinn criter. | | 5.205468 |
| Restr. log likelihood | -522.9973 | Avg. log likelihood | | -2.558066 |
| LR statistic (2 df) | 519.0330 | LR index (Pseudo-R2) | | 0.496210 |
| Probability(LR stat) | 0.000000 | | | |

The negative binomial QML should be consistent, and under the GLM assumption, the standard errors should be consistently estimated. It is worth noting that the coefficient on FEB, which was strongly statistically significant in the Poisson specification, is no longer significantly different from zero at conventional significance levels.

Quasi-likelihood Ratio Statistic

As described by Wooldridge (1996), specification testing using likelihood ratio statistics requires some care when based upon QML models. We illustrate here the differences between a standard LR test for significant coefficients and the corresponding QLR statistic.

From the results above, we know that the overall likelihood ratio statistic for the Poisson model is 16.85, with a corresponding p -value of 0.0002. This statistic is valid under the assumption that $m(x_i, \beta)$ is specified correctly and that the mean-variance equality holds.

We can decisively reject the latter hypothesis, suggesting that we should derive the QML estimator with consistently estimated covariance matrix under the GLM variance assumption. While EViews does not automatically adjust the LR statistic to reflect the QML assumption, it is easy enough to compute the adjustment by hand. Following Wooldridge, we construct the QLR statistic by dividing the original LR statistic by the estimated GLM variance factor.

Suppose that the estimated QML equation is named EQ1. Then you can use EViews to compute p -value associated with this statistic, placing the results in scalars using the following commands:

```
scalar qlr = eq1.@logl/2.226420477
scalar qpval = 1-@cchisq(qlr, 2)
```

You can examine the results by clicking on the scalar objects in the workfile window and viewing the results in the status line. The QLR statistic is 7.5666, and the p -value is 0.023. The statistic and p -value are valid under the weaker conditions that the conditional mean is correctly specified, and that the conditional variance is proportional (but not necessarily equal) to the conditional mean.

Technical Notes

Huber/White (QML) Standard Errors

The Huber/White options for robust standard errors computes the quasi-maximum likelihood (or pseudo-ML) standard errors:

$$\text{var}_{QML}(\hat{\beta}) = \hat{H}^{-1} \hat{g} \hat{g}' \hat{H}^{-1}, \quad (30.49)$$

where \hat{g} and \hat{H} are the gradient (or score) and Hessian of the log likelihood evaluated at the ML estimates.

Note that these standard errors are *not* robust to heteroskedasticity in binary dependent variable models. They are robust to certain misspecifications of the underlying distribution of y .

GLM Standard Errors

Many of the discrete and limited dependent variable models described in this chapter belong to a class of models known as *generalized linear models* (GLM). The assumption of GLM is that the distribution of the dependent variable y_i belongs to the exponential family and that the conditional mean of y_i is a (smooth) nonlinear transformation of the linear part $x_i'\beta$:

$$E(y_i|x_i, \beta) = h(x_i'\beta). \quad (30.50)$$

Even though the QML covariance is robust to general misspecification of the conditional distribution of y_i , it does not possess any efficiency properties. An alternative consistent estimate of the covariance is obtained if we impose the GLM condition that the (true) variance of y_i is proportional to the variance of the distribution used to specify the log likelihood:

$$\text{var}(y_i|x_i, \beta) = \sigma^2 \text{var}_{ML}(y_i|x_i, \beta). \quad (30.51)$$

In other words, the ratio of the (conditional) variance to the mean is some constant σ^2 that is independent of x . The most empirically relevant case is $\sigma^2 > 1$, which is known as *overdispersion*. If this proportional variance condition holds, a consistent estimate of the GLM covariance is given by:

$$\text{var}_{GLM}(\hat{\beta}) = \hat{\sigma}^2 \text{var}_{ML}(\hat{\beta}), \quad (30.52)$$

where

$$\hat{\sigma}^2 = \frac{1}{N-K} \cdot \sum_{i=1}^N \frac{(y_i - \hat{y}_i)^2}{\sqrt{v(x_i, \hat{\beta}, \hat{\gamma})}} = \frac{1}{N-K} \cdot \sum_{i=1}^N \frac{\hat{u}_i^2}{\sqrt{(v(x_i, \hat{\beta}, \hat{\gamma}))}}. \quad (30.53)$$

If you select GLM standard errors, the estimated proportionality term $\hat{\sigma}^2$ is reported as the variance factor estimate in EViews.

For more discussion on GLM and the phenomenon of overdispersion, see McCullough and Nelder (1989).

The Hosmer-Lemeshow Test

Let the data be grouped into $j = 1, 2, \dots, J$ groups, and let n_j be the number of observations in group j . Define the number of $y_i = 1$ observations and the average of predicted values in group j as:

$$\begin{aligned} y(j) &= \sum_{i \in j} y_i \\ \bar{p}(j) &= \sum_{i \in j} \hat{p}_i / n_j = \sum_{i \in j} (1 - F(-x_i'\hat{\beta})) / n_j \end{aligned} \quad (30.54)$$

The Hosmer-Lemeshow test statistic is computed as:

$$HL = \sum_{j=1}^J \frac{(y(j) - n_j \bar{p}(j))^2}{n_j \bar{p}(j)(1 - \bar{p}(j))}. \quad (30.55)$$

The distribution of the HL statistic is not known; however, Hosmer and Lemeshow (1989, p.141) report evidence from extensive simulation indicating that when the model is correctly specified, the distribution of the statistic is well approximated by a χ^2 distribution with $J - 2$ degrees of freedom. Note that these findings are based on a simulation where J is close to n .

The Andrews Test

Let the data be grouped into $j = 1, 2, \dots, J$ groups. Since y is binary, there are $2J$ cells into which any observation can fall. Andrews (1988a, 1988b) compares the $2J$ vector of the actual number of observations in each cell to those predicted from the model, forms a quadratic form, and shows that the quadratic form has an asymptotic χ^2 distribution if the model is specified correctly.

Andrews suggests three tests depending on the choice of the weighting matrix in the quadratic form. EVIEWS uses the test that can be computed by an auxiliary regression as described in Andrews (1988a, 3.18) or Andrews (1988b, 17).

Briefly, let \tilde{A} be an $n \times J$ matrix with element $\tilde{a}_{ij} = 1(i \in j) - \hat{p}_i$, where the indicator function $1(i \in j)$ takes the value one if observation i belongs to group j with $y_i = 1$, and zero otherwise (we drop the columns for the groups with $y = 0$ to avoid singularity). Let B be the $n \times K$ matrix of the contributions to the score $\partial l(\beta)/\partial \beta'$. The Andrews test statistic is n times the R^2 from regressing a constant (one) on each column of \tilde{A} and B . Under the null hypothesis that the model is correctly specified, nR^2 is asymptotically distributed χ^2 with J degrees of freedom.

References

- Aitchison, J. and S.D. Silvey (1957). "The Generalization of Probit Analysis to the Case of Multiple Responses," *Biometrika*, 44, 131–140.
- Agresti, Alan (1996). *An Introduction to Categorical Data Analysis*, New York: John Wiley & Sons.
- Andrews, Donald W. K. (1988a). "Chi-Square Diagnostic Tests for Econometric Models: Theory," *Econometrica*, 56, 1419–1453.
- Andrews, Donald W. K. (1988b). "Chi-Square Diagnostic Tests for Econometric Models: Introduction and Applications," *Journal of Econometrics*, 37, 135–156.
- Cameron, A. Colin and Pravin K. Trivedi (1990). "Regression-based Tests for Overdispersion in the Poisson Model," *Journal of Econometrics*, 46, 347–364.
- Chesher, A. and M. Irish (1987). "Residual Analysis in the Grouped Data and Censored Normal Linear Model," *Journal of Econometrics*, 34, 33–62.

- Chesher, A., T. Lancaster, and M. Irish (1985). "On Detecting the Failure of Distributional Assumptions," *Annales de L'Insee*, 59/60, 7–44.
- Davidson, Russell and James G. MacKinnon (1993). *Estimation and Inference in Econometrics*, Oxford: Oxford University Press.
- Gourieroux, C., A. Monfort, E. Renault, and A. Trognon (1987). "Generalized Residuals," *Journal of Econometrics*, 34, 5–32.
- Gourieroux, C., A. Monfort, and C. Trognon (1984a). "Pseudo-Maximum Likelihood Methods: Theory," *Econometrica*, 52, 681–700.
- Gourieroux, C., A. Monfort, and C. Trognon (1984b). "Pseudo-Maximum Likelihood Methods: Applications to Poisson Models," *Econometrica*, 52, 701–720.
- Greene, William H. (1997). *Econometric Analysis*, 3rd Edition, Upper Saddle River, NJ: Prentice Hall.
- Harvey, Andrew C. (1987). "Applications of the Kalman Filter in Econometrics," Chapter 8 in Truman F. Bewley (ed.), *Advances in Econometrics—Fifth World Congress*, Volume 1, Cambridge: Cambridge University Press.
- Harvey, Andrew C. (1989). *Forecasting, Structural Time Series Models and the Kalman Filter*, Cambridge: Cambridge University Press.
- Hosmer, David W. Jr. and Stanley Lemeshow (1989). *Applied Logistic Regression*, New York: John Wiley & Sons.
- Johnston, Jack and John Enrico DiNardo (1997). *Econometric Methods*, 4th Edition, New York: McGraw-Hill.
- Kennan, John (1985). "The Duration of Contract Strikes in U.S. Manufacturing," *Journal of Econometrics*, 28, 5–28.
- Maddala, G. S. (1983). *Limited-Dependent and Qualitative Variables in Econometrics*, Cambridge: Cambridge University Press.
- McCullagh, P. and J. A. Nelder (1989). *Generalized Linear Models*, 2nd Edition, London: Chapman & Hall.
- McDonald, J. and R. Moffitt (1980). "The Uses of Tobit Analysis," *Review of Economic and Statistics*, 62, 318–321.
- Pagan, A. and F. Vella (1989). "Diagnostic Tests for Models Based on Individual Data: A Survey," *Journal of Applied Econometrics*, 4, S29–S59.
- Pindyck, Robert S. and Daniel L. Rubinfeld (1998). *Econometric Models and Economic Forecasts*, 4th edition, New York: McGraw-Hill.
- Powell, J. L. (1986). "Symmetrically Trimmed Least Squares Estimation for Tobit Models," *Econometrica*, 54, 1435–1460.
- Wooldridge, Jeffrey M. (1990). "Quasi-Likelihood Methods for Count Data," Chapter 8 in M. Hashem Pesaran and P. Schmidt (eds.) *Handbook of Applied Econometrics*, Volume 2, Malden, MA: Blackwell, 352–406.

Chapter 31. Quantile Regression

While the great majority of regression models are concerned with analyzing the conditional mean of a dependent variable, there is increasing interest in methods of modeling other aspects of the conditional distribution. One increasingly popular approach, *quantile regression*, models the quantiles of the dependent variable given a set of conditioning variables.

As originally proposed by Koenker and Bassett (1978), quantile regression provides estimates of the linear relationship between regressors X and a specified quantile of the dependent variable Y . One important special case of quantile regression is the least absolute deviations (LAD) estimator, which corresponds to fitting the conditional median of the response variable.

Quantile regression permits a more complete description of the conditional distribution than conditional mean analysis alone, allowing us, for example, to describe how the median, or perhaps the 10th or 95th percentile of the response variable, are affected by regressor variables. Moreover, since the quantile regression approach does not require strong distributional assumptions, it offers a distributionally robust method of modeling these relationships.

The remainder of this chapter describes the basics of performing quantile regression in EViews. We begin with a walkthrough showing how to estimate a quantile regression specification and describe the output from the procedure. Next we examine the various views and procedures that one may perform using an estimated quantile regression equation. Lastly, we provide background information on the quantile regression model.

Estimating Quantile Regression in EViews

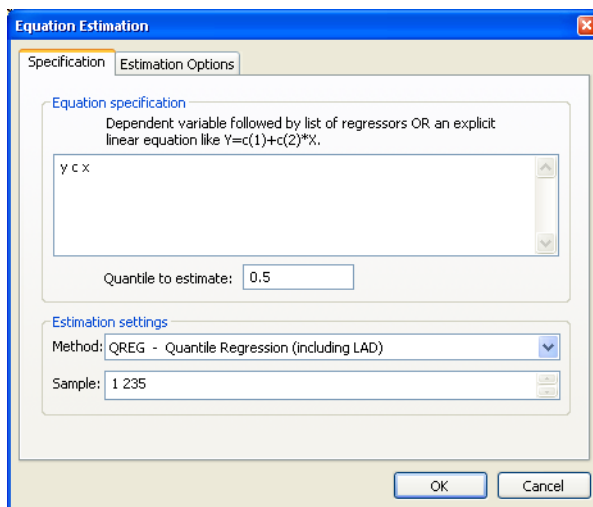
To estimate a quantile regression specification in EViews you may select **Object/New Object.../Equation** or **Quick/Estimate Equation...** from the main menu, or simply type the keyword `equation` in the command window. From the main estimation dialog you should select **QREG - Quantile Regression (including LAD)**. Alternately, you may type `qreg` in the command window.

EViews will open the quantile regression form of the **Equation Estimation** dialog.

Specification

The dialog has two pages. The first page, depicted here, is used to specify the variables in the conditional quantile function, the quantile to estimate, and the sample of observations to use.

You may enter the **Equation specification** using a list of the dependent and regressor variables, as depicted here, or you may enter an explicit expression. Note that if you enter an explicit expression it must be linear in the coefficients.



The **Quantile to estimate** edit field is where you will enter your desired quantile. By default, EViews estimates the median regression as depicted here, but you may enter any value between 0 and 1 (though values very close to 0 and 1 may cause estimation difficulties).

Here we specify a conditional median function for Y that depends on a constant term and the series X. EViews will estimate the LAD estimator for the entire sample of 235 observations.

Estimation Options

Most of the quantile regression settings are set using this page. The options on the left-hand side of the page control the method for computing the coefficient covariances, allow you to specify a weight series for weighted estimation, and specify the method for computing scalar sparsity estimates.

Quantile Regression Options

The combo box labeled **Coefficient Covariance** is where you will choose your method of computing covariances: computing **Ordinary (IID)** covariances, using a **Huber Sandwich** method, or using **Bootstrap** resampling. By default, EViews uses the **Huber Sandwich** calculations which are valid under independent but non-identical sampling.

Just below the combo box is an edit field labeled **Weight**, where you may enter a weight series or series expression.

The remaining settings in this section control the estimation of the scalar sparsity value. Different options are available for different **Coefficient Covariance** settings. For ordinary or bootstrap covariances you may choose either **Siddiqui (mean fitted)**, **Kernel (residual)**, or **Siddiqui (residual)** as your sparsity estimation method, while if the covariance method is set to **Huber Sandwich**, only the **Siddiqui (mean fitted)** and **Kernel (residual)** methods are available.

There are additional options for the bandwidth method (and associated size parameter if relevant), the method for computing empirical quantiles (used to estimate the sparsity or the kernel bandwidth), and the choice of kernel function. Most of these settings should be self-explanatory; if necessary, see the discussion in [“Sparsity Estimation,” beginning on page 273](#) for details.

It is worth mentioning that the sparsity estimation options are always relevant, since EViews always computes and reports a scalar sparsity estimate, even if it is not used in computing the covariance matrix. In particular, a sparsity value is estimated even when you compute the asymptotic covariance using a Huber Sandwich method. The sparsity estimate will be used in non-robust quasi-likelihood ratio tests statistics as necessary.

Iteration Control

The iteration control section offers the standard edit field for changing the maximum number of iterations, a combo box for specifying starting values, and a check box for displaying the estimation settings in the output. Note that the default starting value for quantile regression is 0, but you may choose a fraction of the OLS estimates, or provide a set of user specified values.

Bootstrap Settings

When you select **Bootstrap** in the **Coefficient Covariance** combo, the right side of the dialog changes to offer a set of bootstrap options.

Quantile regression options

Coefficient Covariance: Huber Sandwich

Weight:

Sparsity Estimation

Method: Kernel (residual)

Bandwidth Method: Hall-Sheather

Size Param: 0.05

Quantile Method: Rankit (Cleveland)

Kernel: Epanechnikov

You may use the **Method** combo box to choose from one of four bootstrap methods: **Residual**, **XY-pair**, **MCMB**, **MCMB-A**. See “[Bootstrapping](#),” beginning on page 276 for a discussion of the various methods. The default method is **XY-pair**.

Just below the combo box are two edit fields labeled **Replications** and **No. of obs.** By default, EViews will perform 100 bootstrap replications, but you may override this by entering your desired value. The **No. of obs.** edit field controls the size of the bootstrap sample. If the edit field is left blank, EViews will draw samples of the same size as the original data. There is some evidence that specifying a bootstrap sample size m smaller than n may produce more accurate results, especially for very large sample sizes; Koenker (2005, p. 108) provides a brief summary.

The screenshot shows a dialog box titled "Bootstrap settings". It contains the following fields and controls:

- Method:** A dropdown menu with "XY-pair" selected.
- Replications:** A text input field containing the value "100".
- No. of obs:** A text input field that is currently empty.
- Output:** A text input field that is currently empty.
- Random generator:** A dropdown menu with "Knuth" selected.
- Seed:** A text input field that is currently empty, followed by a "Clear" button.

To save the results of your bootstrap replications in a matrix object, enter the name in the **Output** edit field.

The last two items control the generation of random numbers. The **Random generator** combo should be self-explanatory. Simply use the combo to choose your desired generator. EViews will initialize the combo using the default settings for the choice of generator.

The random **Seed** field requires some discussion. By default, the first time that you perform a bootstrap for a given equation, the **Seed** edit field will be blank; you may provide your own integer value, if desired. If an initial seed is not provided, EViews will randomly select a seed value. The value of this initial seed will be saved with the equation so that by default, subsequent estimation will employ the same seed, allowing you to replicate results when re-estimating the equation, and when performing tests. If you wish to use a different seed, simply enter a value in the **Seed** edit field or press the **Clear** button to have EViews draw a new random seed value.

Estimation Output

Once you have provided your quantile regression specification and specified your options, you may click on **OK** to estimate your equation. Unless you are performing bootstrapping with a very large number of observations, the estimation results should be displayed shortly.

Our example uses the Engel dataset containing food expenditure and household income considered by Koenker (2005, p. 78-79, 297-307). The default model estimates the median of food expenditure Y as a function of a constant term and household income X .

Dependent Variable: Y
 Method: Quantile Regression (Median)
 Date: 11/06/06 Time: 09:59
 Sample: 1 235
 Included observations: 235
 Huber Sandwich Standard Errors & Covariance
 Sparsity method: Kernel (Epanechnikov) using residuals
 Bandwidth method: Hall-Sheather, bw=0.15744
 Estimation successfully identifies unique optimal solution

| | Coefficient | Std. Error | t-Statistic | Prob. |
|------------------------|-------------|-------------------------|-------------|----------|
| C | 81.48225 | 24.03494 | 3.390158 | 0.0008 |
| X | 0.560181 | 0.031370 | 17.85707 | 0.0000 |
| Pseudo R-squared | 0.620556 | Mean dependent var | | 624.1501 |
| Adjusted R-squared | 0.618927 | S.D. dependent var | | 276.4570 |
| S.E. of regression | 120.8447 | Objective | | 8779.966 |
| Quantile dependent var | 582.5413 | Objective (const. only) | | 23139.03 |
| Sparsity | 209.3504 | Quasi-LR statistic | | 548.7091 |
| Prob(Quasi-LR stat) | 0.000000 | | | |

The top portion of the output displays the estimation settings. Here we see that our estimates use the Huber sandwich method for computing the covariance matrix, with individual sparsity estimates obtained using kernel methods. The bandwidth uses the Hall and Sheather formula, yielding a value of 0.15744.

Below the header information are the coefficients, along with standard errors, t -statistics and associated p -values. We see that both coefficients are statistically significantly different from zero and conventional levels.

The bottom portion of the output reports the Koenker and Machado (1999) goodness-of-fit measure (pseudo R-squared), and adjusted version of the statistic, as well as the scalar estimate of the sparsity using the kernel method. Note that this scalar estimate is not used in the computation of the standard errors in this case since we are employing the Huber sandwich method.

Also reported are the minimized value of the objective function (“Objective”), the minimized constant-only version of the objective (“Objective (const. only)”), the constant-only coefficient estimate (“Quantile dependent var”), and the corresponding $L_n(\tau)$ form of the Quasi-LR statistic and associated probability for the difference between the two specifications (Koenker and Machado, 1999). Note that despite the fact that the coefficient covariances are computed using the robust Huber Sandwich, the QLR statistic assumes *i.i.d.* errors and uses the estimated value of the sparsity.

The reported S.E. of the regression is based on the usual d.f. adjusted sample variance of the residuals. This measure of scale is used in forming standardized residuals and forecast standard errors. It is replaced by the Koenker and Machado (1999) scale estimator in the computation of the $\Lambda_n(\tau)$ form of the QLR statistics (see “[Standard Views and Procedures](#)” on page 265 and “[Quasi-Likelihood Ratio Tests](#)” on page 278).

We may elect instead to perform bootstrapping to obtain the covariance matrix. Click on the **Estimate** button to bring up the dialog, then on **Estimation Options** to show the options tab. Select Bootstrap as the **Coefficient Covariance**, then choose **MCMB-A** as the bootstrap method. Next, we increase the number of replications to 500. Lastly, to see the effect of using a different estimator of the sparsity, we change the scalar sparsity estimation method to **Siddiqui (mean fitted)**. Click on **OK** to estimate the specification.

| Dependent Variable: Y | | | | |
|--|-------------|-------------------------|-------------|----------|
| Method: Quantile Regression (Median) | | | | |
| Date: 11/06/06 Time: 11:00 | | | | |
| Sample: 1 235 | | | | |
| Included observations: 235 | | | | |
| Bootstrap Standard Errors & Covariance | | | | |
| Bootstrap method: MCMB-A, reps=500, mg=kn, seed=47500547 | | | | |
| Sparsity method: Siddiqui using fitted quantiles | | | | |
| Bandwidth method: Hall-Sheather, bw=0.15744 | | | | |
| Estimation successfully identifies unique optimal solution | | | | |
| | Coefficient | Std. Error | t-Statistic | Prob. |
| C | 81.48225 | 22.01534 | 3.701158 | 0.0003 |
| X | 0.560181 | 0.023804 | 23.53350 | 0.0000 |
| Pseudo R-squared | 0.620556 | Mean dependent var | | 624.1501 |
| Adjusted R-squared | 0.618927 | S.D. dependent var | | 276.4570 |
| S.E. of regression | 120.8447 | Objective | | 8779.966 |
| Quantile dependent var | 582.5413 | Objective (const. only) | | 23139.03 |
| Sparsity | 267.8284 | Quasi-LR statistic | | 428.9034 |
| Prob(Quasi-LR stat) | 0.000000 | | | |

For the most part the results are quite similar. The header information shows the different method of computing coefficient covariances and sparsity estimates. The Huber Sandwich and bootstrap standard errors are reasonably close (24.03 versus 22.02, and 0.031 versus 0.024). There are moderate differences between the two sparsity estimates, with the Siddiqui estimator of the sparsity roughly 25 % higher (267.83 versus 209.35), but this difference has no appreciable impact on the QLR statistic.

Views and Procedures

We turn now to a brief description of the views and procedures that are available for equations estimated using quantile regression. Most of the available views and procedures for the quantile regression equation are identical to those for an ordinary least squares regression, but a few require additional discussion.

Standard Views and Procedures

With the exception of the views listed under **Quantile Process**, the quantile regression views and procedures should be familiar from the discussion in ordinary least squares regression (see [“Working with Equations” on page 17](#)).

A few of the familiar views and procedures do require a brief comment or two:

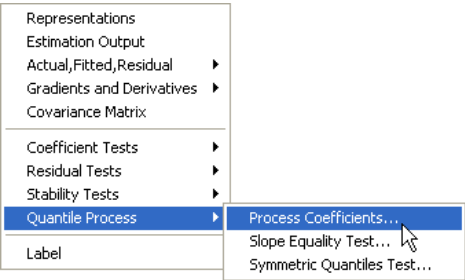
- Residuals are computed using the estimated parameters for the specified quantile: $\hat{\epsilon}_i(\tau) = Y_i - X_i' \beta(\tau)$. Standardized residuals are the ratios of the residuals to the the degree-of-freedom corrected sample standard deviation of the residuals.

Note that an alternative approach to standardizing residuals that is not employed here would follow Koenker and Machado (1999) in estimating the scale parameter using the average value of the minimized objective function $\hat{\sigma}(\tau) = n^{-1} \hat{V}(\tau)$. This latter estimator is used in forming quasi-likelihood ratio (QLR) tests ([“Quasi-Likelihood Ratio Tests” on page 278](#)).

- Wald tests and confidence ellipses are constructed in the usual fashion using the possibly robust estimator for the coefficient covariance matrix specified during estimation.
- The omitted and redundant variables tests and the Ramsey RESET test all perform QLR tests of the specified restrictions (Koenker and Machado, 1999). These tests require the *i.i.d.* assumption for the sparsity estimator to be valid.
- Forecasts and models will be for the estimated conditional quantile specification, using the estimated $\hat{\beta}(\tau)$. We remind you that by default, EViews forecasts will insert the actual values for out-of-forecast-sample observations, which may not be the desired approach. You may switch the insertion off by unselecting the **Insert actuals for out-of-sample observations** checkbox in the **Forecast** dialog.

Quantile Process Views

The **Quantile Process** view submenu lists three specialized views that rely on quantile process estimates. Before describing the three views, we note that since each requires estimation of quantile regression specifications for various τ , they may be time-consuming, especially for specifications where the coefficient covariance is estimated via bootstrapping.

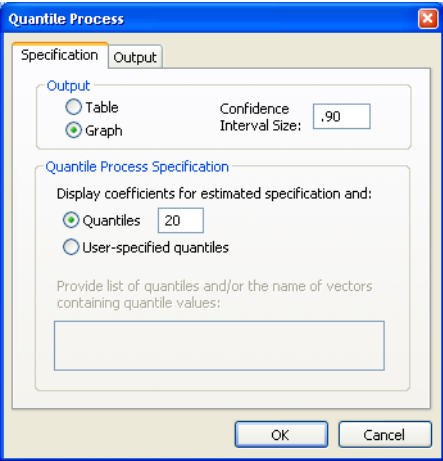


Process Coefficients

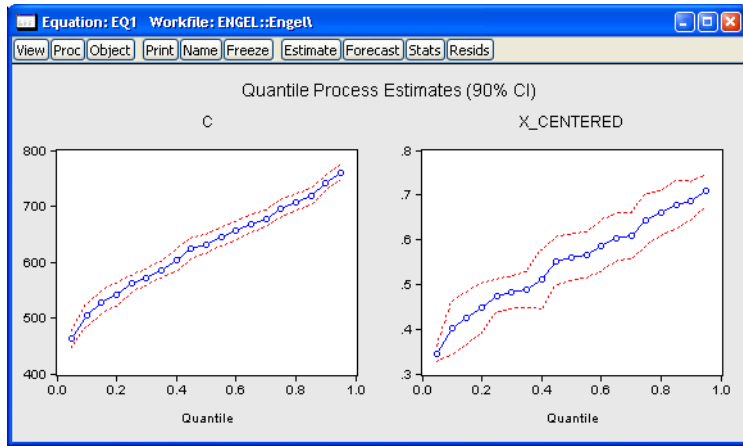
You may select **View/Quantile Process/Process Coefficients** to examine the process coefficients estimated at various quantiles.

The **Output** section of the **Specification** page is used to control how the process results are displayed. By default, EViews displays the results as a table of coefficient estimates, standard errors, t -statistics, and p -values. You may instead click on the **Graph** radio button and enter the size of the confidence interval in the edit field that appears. The default is to display a 95% confidence interval.

The **Quantile Process Specification** section of the page determines the quantiles at which the process will be estimated. By default, EViews will estimate models for each of the deciles (10 quantiles, $\tau = \{0.1, 0.2, \dots, 0.9\}$). You may specify a different number of quantiles using the edit field, or you may select **User-specified quantiles** and then enter a list of quantiles or one or more vectors containing quantile values.



Here, we follow Koenker (2005), in displaying a process graph for a modified version of the earlier equation; a median regression using the Engel data, where we fit the Y data to the centered X series and a constant. We display the results for 20 quantiles, along with 90% confidence intervals.



In both cases, the coefficient estimates show a clear positive relationship between the quantile value and the estimated coefficients; the positive relationship between X_CENTERED is clear evidence that the conditional quantiles are not *i.i.d.* We test the strength of this relationship formally below.

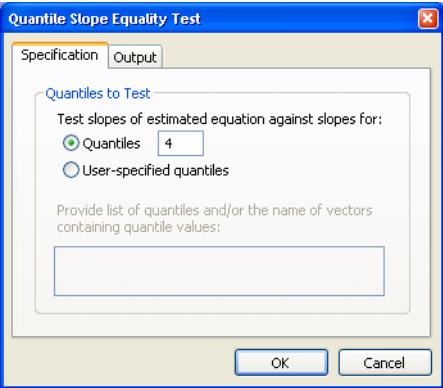
The **Output** page of the dialog allows you to save the results of the quantile process estimation. You may provide a name for the vector of quantiles, the matrix of process coefficients, and the covariance matrix of the coefficients. For the k sorted quantile estimates, each row of the $k \times p$ coefficient matrix contains estimates for a given quantile. The covariance matrix is the covariance of the vec of the coefficient matrix.

Slope Equality Test

To perform the Koenker and Bassett (1982a) test for the equality of the slope coefficients across quantiles, select **View/Quantile Process/Slope Equality Test...** and fill out the dialog.

The image shows the 'Quantile Process' dialog box with the 'Output' tab selected. Under the heading 'Save Quantile Estimates', there are three input fields: 'Quantiles Vector:', 'Coefficient Matrix:', and 'Covariance Matrix:'. At the bottom of the dialog are 'OK' and 'Cancel' buttons.

The dialog has two pages. The **Specification** page is used to determine the quantiles at which the process will be compared. EViews will compare with slope (non-intercept) coefficients of the estimated tau, with the taus specified in the dialog. By default, the comparison taus will be the three quartile limits ($\tau = \{0.25, 0.5, 0.75\}$), but you may select **User-specified quantiles** and provide your own values.



The **Output** page allows you to save the results from the supplementary process estimation. As in “[Process Coefficients](#)” on page 266, you may provide a name for the vector of quantiles, the matrix of process coefficients, and the covariance matrix of the coefficients.

The results for the slope equality test for a median regression of our earlier equation relating food expenditure and household income in the Engel data set. We compare the slope coefficient for the median against those estimated at the upper and lower quartile.

Quantile Slope Equality Test
Equation: UNTITLED
Specification: Y C X

| Test Summary | Chi-Sq. Statistic | Chi-Sq. d.f. | Prob. |
|--------------|-------------------|--------------|--------|
| Wald Test | 25.22366 | 2 | 0.0000 |

Restriction Detail: b(tau_h) - b(tau_k) = 0

| Quantiles | Variable | Restr. Value | Std. Error | Prob. |
|-----------|----------|--------------|------------|--------|
| 0.25, 0.5 | X | -0.086077 | 0.025923 | 0.0009 |
| 0.5, 0.75 | | -0.083834 | 0.030529 | 0.0060 |

The top portion of the output shows the equation specification, and the Wald test summary. Not surprisingly (given the graph of the coefficients above), we see that the χ^2 -statistic value of 25.22 is statistically significant at conventional test levels. We conclude that coefficients differ across quantile values and that the conditional quantiles are not identical.

Symmetric Quantiles Test

The symmetric quantiles test performs the Newey and Powell (1987) of conditional symmetry. Conditional symmetry implies that the average value of two sets of coefficients for symmetric quantiles around the median will equal the value of the coefficients at the median:

$$\frac{\beta(\tau) + \beta(1 - \tau)}{2} = \beta(1/2) \quad (31.1)$$

By default, EViews will test for symmetry using the estimated quantile and the quantile limits specified in the dialog. Thus, if the estimated model fits the median, there will be a single set of restrictions:

$$(\beta(0.25) + \beta(0.75))/2 = \beta(0.5).$$

If the estimated model fits the 0.6 quantile, there will be an additional set of restrictions:

$$(\beta(0.4) + \beta(0.6))/2 = \beta(0.5).$$

As with the other process routines, you may select **User-specified quantiles** and provide your own values. EViews will estimate a model for both the specified quantile, τ , and its complement $1 - \tau$, and will compare the results to the median estimates.

If your original model is for a quantile other than the median, you will be offered a third choice of performing the test using only the estimated quantile. For example, if the model is fit to the 0.6, an additional radio button will appear: **Estimation quantile only (0.6)**.

Choosing this form of the test, there will be a single set of restrictions:

$$(\beta(0.4) + \beta(0.6))/2 = \beta(0.5).$$

Also, if it is known *a priori* that the errors are *i.i.d.*, but possibly not symmetrically distributed, one can restrict the null to examine only the restriction associated with the intercept. To perform this restricted version of the test, simply click on **Intercept only** in the **Test Specification** portion of the page.

Lastly, you may use the **Output** page to save the results from the supplementary process estimation. You may provide a name for the vector of quantiles, the matrix of process coefficients, and the covariance matrix of the coefficients.

The default test of symmetry for the basic median Engel curve specification is given below:

Symmetric Quantiles Test
Equation: UNTITLED
Specification: Y C X
Test statistic compares all coefficients

| Test Summary | Chi-Sq. Statistic | Chi-Sq. d.f. | Prob. |
|--------------|----------------------|--------------|--------|
| Wald Test | 0.530024 | 2 | 0.7672 |

Restriction Detail: $b(\text{tau}) + b(1-\text{tau}) - 2*b(.5) = 0$

| Quantiles | Variable | Restr. Value | Std. Error | Prob. |
|------------|----------|--------------|------------|--------|
| 0.25, 0.75 | C | -5.084370 | 34.59898 | 0.8832 |
| | X | -0.002244 | 0.045012 | 0.9602 |

We see that the test compares estimates at the first and third quartile with the median specification. While earlier we saw strong evidence that the slope coefficients are not constant across quantiles, we now see that there is little evidence of departures from symmetry. The overall p -value for the test is around 0.75, and the individual coefficient restriction test values show even less evidence of asymmetry.

Background

We present here a brief discussion of quantile regression. As always, the discussion is necessarily brief and omits considerable detail. For a book-length treatment of quantile regression see Koenker (2005).

The Model

Suppose that we have a random variable Y with probability distribution function

$$F(y) = \text{Prob}(Y \leq y)$$

(31.2)

so that for $0 < \tau < 1$, the τ -th quantile of Y may be defined as the smallest y satisfying $F(y) \geq \tau$:

$$Q(\tau) = \inf\{y: F(y) \geq \tau\}$$

(31.3)

Given a set of n observations on Y , the traditional empirical distribution function is given by:

$$F_n(y) = \sum_k 1(Y_i \leq y)$$

(31.4)

where $1(z)$ is an indicator function that takes the value 1 if the argument z is true and 0 otherwise. The associated empirical quantile is given by,

$$Q_n(\tau) = \inf\{y: F_n(y) \geq \tau\} \quad (31.5)$$

or equivalently, in the form of a simple optimization problem:

$$\begin{aligned} Q_n(\tau) &= \operatorname{argmin}_{\xi} \left\{ \sum_{i: Y_i \geq \xi} \tau |Y_i - \xi| + \sum_{i: Y_i < \xi} (1 - \tau) |Y_i - \xi| \right\} \\ &= \operatorname{argmin}_{\xi} \left\{ \sum_i \rho_{\tau}(Y_i - \xi) \right\} \end{aligned} \quad (31.6)$$

where $\rho_{\tau}(u) = u(\tau - 1(u < 0))$ is the so-called *check function* which weights positive and negative values asymmetrically.

Quantile regression extends this simple formulation to allow for regressors X . We assume a linear specification for the conditional quantile of the response variable Y given values for the p -vector of explanatory variables X :

$$Q(\tau|X_i, \beta(\tau)) = X_i' \beta(\tau) \quad (31.7)$$

where $\beta(\tau)$ is the vector of coefficients associated with the τ -th quantile.

Then the analog to the unconditional quantile minimization above is the conditional quantile regression estimator:

$$\hat{\beta}_n(\tau) = \operatorname{argmin}_{\beta(\tau)} \left\{ \sum_i \rho_{\tau}(Y_i - X_i' \beta(\tau)) \right\} \quad (31.8)$$

Estimation

The quantile regression estimator can be obtained as the solution to a linear programming problem. Several algorithms for obtaining a solution to this problem have been proposed in the literature. EViews uses a modified version of the Koenker and D'Orey (1987) version of the Barrodale and Roberts (1973) simplex algorithm.

The Barrodale and Roberts (BR) algorithm has received more than its fair share of criticism for being computationally inefficient, with dire theoretical results for worst-case scenarios in problems involving large numbers of observations. Simulations showing poor relative performance of the BR algorithm as compared with alternatives such as interior point methods appear to bear this out, with estimation times that are roughly quadratic in the number of observations (Koenker and Hallock, 2001; Portnoy and Koenker, 1997).

Our experience with a suitably optimized version of the BR algorithm is that its performance is certainly better than commonly portrayed. Using various subsets of the low-birthweight

data described in Koenker and Hallock (2001), we find that while certainly not as fast as Cholesky-based linear regression (and most likely not as fast as interior point methods), the estimation times for the modified BR algorithm are quite reasonable.

For example, estimating a 16 explanatory variable model for the median using the first 20,000 observations of the data set takes a bit more than 1.2 seconds on a 3.2GHz Pentium 4, with 1.0Gb of RAM; this time includes both estimation and computation of a kernel based estimator of the coefficient covariance matrix. The same specification using the full sample of 198,377 observations takes under 7.5 seconds.

Overall, our experience is that estimation times for the modified BR are roughly linear in the number of observations through a broad range of sample sizes. While our results are not definitive, we see no real impediment to using this algorithm for virtually all practical problems.

Asymptotic Distributions

Under mild regularity conditions, quantile regression coefficients may be shown to be asymptotically normally distributed (Koenker, 2005) with different forms of the asymptotic covariance matrix depending on the model assumptions.

Computation of the coefficient covariance matrices occupies an important place in quantile regression analysis. In large part, this importance stems from the fact that the covariance matrix of the estimates depends on one or more nuisance quantities which must be estimated. Accordingly, a large literature has developed to consider the relative merits of various approaches to estimating the asymptotic variances (see Koenker (2005), for an overview).

We may divide the estimators into three distinct classes: (1) direct methods for estimating the covariance matrix in *i.i.d.* settings; (2) direct methods for estimating the covariance matrix for independent but not-identical distribution; (3) bootstrap resampling methods for both *i.i.d.* and *i.n.i.d.* settings.

Independent and Identical

Koenker and Bassett (1978) derive asymptotic normality results for the quantile regression estimator in the *i.i.d.* setting, showing that under mild regularity conditions,

$$\sqrt{n}(\hat{\beta}(\tau) - \beta(\tau)) \sim N(0, \tau(1 - \tau)s(\tau)^2 J^{-1}) \quad (31.9)$$

where:

$$J = \lim_{n \rightarrow \infty} \left(\sum_i X_i X_i' / n \right) = \lim_{n \rightarrow \infty} (X' X / n) \quad (31.10)$$

$$s(\tau) = F^{-1}(\tau) = 1/f(F^{-1}(\tau))$$

and $s(\tau)$, which is termed the *sparsity function* or the *quantile density function*, may be interpreted either as the derivative of the quantile function or the inverse of the density function evaluated at the τ -th quantile (see, for example, Welsh, 1988). Note that the *i.i.d.* error assumption implies that $s(\tau)$ does not depend on X so that the quantile functions depend on X only in location, hence all conditional quantile planes are parallel.

Given the value of the sparsity at a given quantile, direct estimation of the coefficient covariance matrix is straightforward. In fact, the expression for the asymptotic covariance in Equation (31.9) is analogous to the ordinary least squares covariance in the *i.i.d.* setting, with $\tau(1 - \tau)s(\tau)^2$ standing in for the error variance in the usual formula.

Sparsity Estimation

We have seen the importance of the sparsity function in the formula for the asymptotic covariance matrix of the quantile regression estimates for *i.i.d.* data. Unfortunately, the sparsity is a function of the unknown distribution F , and therefore is a nuisance quantity which must be estimated.

EViews provides three methods for estimating the scalar sparsity $s(\tau)$: two Siddiqui (1960) difference quotient methods (Koenker, 1994; Bassett and Koenker (1982) and one kernel density estimator (Powell, 1986; Jones, 1992; Buchinsky 1995).

Siddiqui Difference Quotient

The first two methods are variants of a procedure originally proposed by Siddiqui (1960; see Koenker, 1994), where we compute a simple difference quotient of the empirical quantile function:

$$\hat{s}(\tau) = [\hat{F}^{-1}(\tau + h_n) - \hat{F}^{-1}(\tau - h_n)] / (2h_n) \quad (31.11)$$

for some bandwidth h_n tending to zero as the sample size $n \rightarrow \infty$. $\hat{s}(\tau)$ is in essence computed using a simply two-sided numeric derivative of the quantile function. To make this procedure operational we need to determine: (1) how to obtain estimates of the empirical quantile function $F^{-1}(\tau)$ at the two evaluation points, and (2) what bandwidth to employ.

The first approach to evaluating the quantile functions, which EViews terms **Siddiqui (mean fitted)**, is due to Bassett and Koenker (1982). The approach involves estimating two additional quantile regression models for $\tau - h_n$ and $\tau + h_n$, and using the estimated coefficients to compute fitted quantiles. Substituting the fitted quantiles into the numeric derivative expression yields:

$$\hat{s}(\tau) = X^* (\hat{\beta}(\tau + h_n) - \hat{\beta}(\tau - h_n)) / (2h_n) \quad (31.12)$$

for an arbitrary X^* . While the *i.i.d.* assumption implies that X^* may be set to any value, Bassett and Koenker propose using the mean value of X , noting out that the mean possesses two very desirable properties: the precision of the estimate is maximized at that

point, and the empirical quantile function is monotone in τ when evaluated at $X^* = \bar{X}$, so that $\hat{s}(\tau)$ will always yield a positive value for suitable h_n .

A second, less computationally intensive approach to evaluating the quantile functions computes the $\tau + h$ and $\tau - h$ empirical quantiles of the residuals from the original quantile regression equation, as in Koenker (1994). Following Koencker, we compute quantiles for the residuals excluding the p residuals that are set to zero in estimation, and interpolating values to get a piecewise linear version of the quantile. EVIEWS refers to this method as **Siddiqui (residual)**.

Both Siddiqui methods require specification of a bandwidth h_n . EVIEWS offers the Bofinger (1975), Hall-Sheather (1988), and Chamberlain (1994) bandwidth methods (along with the ability to specify an arbitrary bandwidth).

The Bofinger bandwidth, which is given by:

$$h_n = n^{-1/5} \left(\frac{4.5(\phi(\Phi^{-1}(\tau)))^4}{[2(\Phi^{-1}(\tau))^2 + 1]^2} \right)^{1/5} \quad (31.13)$$

(approximately) minimizes the mean square error (MSE) of the sparsity estimates.

Hall-Sheather proposed an alternative bandwidth that is designed specifically for testing. The Hall-Sheather bandwidth is given by:

$$h_n = n^{-1/3} z_\alpha^{2/3} \left(\frac{1.5(\phi(\Phi^{-1}(\tau)))^2}{2(\Phi^{-1}(\tau))^2 + 1} \right)^{1/3} \quad (31.14)$$

where $z_\alpha = \Phi^{-1}(1 - \alpha/2)$, for α the parameter controlling the size of the desired $1 - \alpha$ confidence intervals.

A similar testing motivation underlies the Chamberlain bandwidth:

$$h_n = z_\alpha \sqrt{\frac{\tau(1 - \tau)}{n}} \quad (31.15)$$

which is derived using the exact and normal asymptotic confidence intervals for the order statistics (Buchinsky, 1995).

Kernel Density

Kernel density estimators of the sparsity offer an important alternative to the Siddiqui approach. Most of the attention has focused on kernel methods for estimating $F^{-1'}(\tau)$ directly (Falk, 1988; Welsh, 1988), but one may also estimate $s(\tau)$ using the inverse of a kernel density function estimator (Powell, 1986; Jones, 1992; Buchinsky 1995). In the present context, we may compute:

$$\hat{s}(\tau) = 1 / \left[(1/n) \sum_{i=1}^n c_n^{-1} K(\hat{u}_i(\tau)/c_n) \right] \quad (31.16)$$

where $\hat{u}(\tau)$ are the residuals from the quantile regression fit. EViews supports the latter density function approach, which is termed the **Kernel (residual)** method, since it is closely related to the more commonly employed Powell (1984, 1989) kernel estimator for the non-*i.i.d.* case described below.

Kernel estimation of the density function requires specification of a bandwidth c_n . We follow Koenker (2005, p. 81) in choosing:

$$c_n = \kappa(\Phi^{-1}(\tau + h_n) - \Phi^{-1}(\tau - h_n)) \quad (31.17)$$

where $\kappa = \min(s, IQR/1.34)$ is the Silverman (1986) robust estimate of scale (where s is the sample standard deviation and IQR the interquartile range) and h_n is the Siddiqui bandwidth.

Independent, Non-Identical

We may relax the assumption that the quantile density function does not depend on X . The asymptotic distribution of $\sqrt{n}(\hat{\beta}(\tau) - \beta(\tau))$ in the *i.n.i.d.* setting takes the Huber sandwich form (see, among others, Hendricks and Koenker, 1992):

$$\sqrt{n}(\hat{\beta}(\tau) - \beta(\tau)) \sim N(0, \tau(1 - \tau)H(\tau)^{-1} JH(\tau)^{-1}) \quad (31.18)$$

where J is as defined earlier,

$$J = \lim_{n \rightarrow \infty} \left(\sum_i X_i X_i' / n \right) \quad (31.19)$$

and:

$$H(\tau) = \lim_{n \rightarrow \infty} \left(\sum_i X_i X_i' f_i(q_i(\tau)) / n \right) \quad (31.20)$$

$f_i(q_i(\tau))$ is the conditional density function of the response, evaluated at the τ -th conditional quantile for individual i . Note that if the conditional density does not depend on the observation, the Huber sandwich form of the variance in [Equation \(31.18\)](#) reduces to the simple scalar sparsity form given in [Equation \(31.9\)](#).

Computation of a sample analogue to J is straightforward so we focus on estimation of $H(\tau)$. EViews offers a choice of two methods for estimating $H(\tau)$: a Siddiqui-type difference method proposed by Hendricks and Koenker (1992), and a Powell (1984, 1989) kernel method based on residuals of the estimated model. EViews labels the first method **Siddiqui (mean fitted)**, and the latter method **Kernel (residual)**:

The Siddiqui-type method proposed by Hendricks and Koenker (1991) is a straightforward generalization of the scalar Siddiqui method (see [“Siddiqui Difference Quotient,” beginning on page 273](#)). As before, two additional quantile regression models are estimated for $\tau - h$ and $\tau + h$, and the estimated coefficients may be used to compute the Siddiqui difference quotient:

$$\begin{aligned}\hat{f}_i(q_i(\tau)) &= 2h_n/(\hat{F}_i^{-1}(q_i(\tau + h)) - \hat{F}_i^{-1}(q_i(\tau - h))) \\ &= 2h_n/(X_i'(\hat{\beta}(\tau + h) - \hat{\beta}(\tau - h)))\end{aligned}\quad (31.21)$$

Note that in the absence of identically distributed data the quantile density function $\hat{f}_i(q_i(\tau))$ must be evaluated for each individual. One minor complication is that the [Equation \(31.21\)](#) is not guaranteed to be positive except at $X_i = \bar{X}$. Accordingly, Hendricks and Koenker modify the expression slightly to use only positive values:

$$\hat{f}_i(q_i(\tau)) = \max(0, 2h_n/(X_i'(\hat{\beta}(\tau + h) - \hat{\beta}(\tau - h)) - \delta)) \quad (31.22)$$

where δ is a small positive number included to prevent division by zero.

The estimated quantile densities $\hat{f}_i(q_i(\tau))$ are then used to form an estimator \hat{H}_n of H :

$$\hat{H}_n = \sum_i \hat{f}_i(q_i(\tau)) X_i X_i' / n \quad (31.23)$$

The Powell (1984, 1989) kernel approach replaces the Siddiqui difference with a kernel density estimator using the residuals of the original fitted model:

$$\hat{H}_n = (1/n) \sum_i c_n^{-1} K(\hat{u}_i(\tau)/c_n) X_i X_i' \quad (31.24)$$

where K is a kernel function that integrates to 1, and c_n is a kernel bandwidth. EViews uses the Koenker (2005) kernel bandwidth as described in [“Kernel Density” on page 274](#) above.

Bootstrapping

The direct methods of estimating the asymptotic covariance matrices of the estimates require the estimation of the sparsity nuisance parameter, either at a single point, or conditionally for each observation. One method of avoiding this cumbersome estimation is to employ bootstrapping techniques for the estimation of the covariance matrix.

EViews supports four different bootstrap methods: the residual bootstrap (**Residual**), the design, or XY-pair, bootstrap (**XY-pair**), and two variants of the Markov Chain Marginal Bootstrap (**MCMC** and **MBMB-A**).

The following discussion provides a brief overview of the various bootstrap methods. For additional detail, see Buchinsky (1995, He and Hu (2002) and Kocherginsky, He, and Mu (2005).

Residual Bootstrap

The *residual bootstrap*, is constructed by resampling (with replacement) separately from the residuals $\hat{u}_i(\tau)$ and from the X_i .

Let u^* be an m -vector of resampled residuals, and let X^* be a $m \times p$ matrix of independently resampled X . (Note that m need not be equal to the original sample size n .) We form the dependent variable using the resampled residuals, resampled data, and estimated coefficients, $Y^* = X^* \hat{\beta}(\tau) + u^*$, and then construct a bootstrap estimate of $\beta(\tau)$ using Y^* and X^* .

This procedure is repeated for M bootstrap replications, and the estimator of the asymptotic covariance matrix is formed from:

$$\hat{V}(\hat{\beta}) = n \left(\frac{m}{n} \right) \frac{1}{B} \sum_{j=1}^B (\hat{\beta}_j(\tau) - \overline{\hat{\beta}(\tau)}) (\hat{\beta}_j(\tau) - \overline{\hat{\beta}(\tau)})' \quad (31.25)$$

where $\overline{\hat{\beta}(\tau)}$ is the mean of the bootstrap elements. The bootstrap covariance matrix $\hat{V}(\hat{\beta})$ is simply a (scaled) estimate of the sample variance of the bootstrap estimates of $\beta(\tau)$.

Note that the validity of using separate draws from $\hat{u}_i(\tau)$ and X_i requires independence of the u and the X .

XY-pair (Design) Bootstrap

The *XY-pair* bootstrap is the most natural form of bootstrap resampling, and is valid in settings where u and X are not independent. For the XY-pair bootstrap, we simply form B randomly drawn (with replacement) subsamples of size m from the original data, then compute estimates of $\beta(\tau)$ using the (y^*, X^*) for each subsample. The asymptotic covariance matrix is then estimated from sample variance of the bootstrap results using [Equation \(31.25\)](#).

Markov Chain Marginal Bootstrap

The primary disadvantage to the residual and design bootstrapping methods is that they are computationally intensive, requiring estimation of a relatively difficult p -dimensional linear programming problem for each bootstrap replication.

He and Hu (2002) proposed a new method for constructing bootstrap replications that reduces each p -dimensional bootstrap optimization to a sequence of p easily solved one-dimensional problems. The sequence of one-dimensional solutions forms a Markov chain whose sample variance, computed using [Equation \(31.25\)](#), consistently approximates the true covariance for large n and M .

One problem with the MCMB is that high autocorrelations in the MCMB sequence for specific coefficients will result in a poor estimates for the asymptotic covariance for given chain

length M , and may result in non-convergence of the covariance estimates for any chain of practical length.

Kocherginsky, He, and Mu (KHM, 2005) propose a modification to MCMB, which alleviates autocorrelation problems by transforming the parameter space prior to performing the MCMB algorithm, and then transforming the result back to the original space. Note that the resulting MCMB-A algorithm requires the *i.i.d.* assumption, though the authors suggest that the method is robust against heteroskedasticity.

Practical recommendations for the MCMB-A are provided in KHM. Summarizing, they recommend that the methods be applied to problems where $n \cdot \min(\tau, 1 - \tau) > 5p$ with M between 100 and 200 for relatively small problems ($n \leq 1000$, $p \leq 10$). For moderately large problems with np between 10,000 and 2,000,000, they recommend M between 50 and 200 depending on one's level of patience.

Model Evaluation and Testing

Evaluation of the quality of a quantile regression model may be conducted using goodness-of-fit criteria, as well as formal testing using quasi-likelihood ratio and Wald tests.

Goodness-of-Fit

Koenker and Machado (1999) define a goodness-of-fit statistics for quantile regression that is analogous to the R^2 from conventional regression analysis. We begin by recalling our linear quantile specification, $Q(\tau|X_i, \beta(\tau)) = X_i' \beta(\tau)$ and assume that we may partition the data and coefficient vector as $X_i = (1, X_{i1}')'$ and $\beta(\tau) = (\beta_0(\tau), \beta_1(\tau)')'$, so that

$$Q(\tau|X_i, \beta(\tau)) = \beta_0(\tau) + X_{i1}' \beta_1(\tau) \quad (31.26)$$

We may then define:

$$\begin{aligned} \hat{V}(\tau) &= \min_{\beta(\tau)} \sum_i \rho_\tau(Y_i - \beta_0(\tau) - X_{i1}' \beta_1(\tau)) \\ \tilde{V}(\tau) &= \min_{\beta_0(\tau)} \sum_i \rho_\tau(Y_i - \beta_0(\tau)) \end{aligned} \quad (31.27)$$

the minimized unrestricted and intercept-only objective functions. The Koenker and Machado goodness-of-fit criterion is given by:

$$R^1(\tau) = 1 - \hat{V}(\tau) / \tilde{V}(\tau) \quad (31.28)$$

This statistic is an obvious analogue of the conventional R^2 . $R^1(\tau)$ lies between 0 and 1, and measures the relative success of the model in fitting the data for the τ -th quantile.

Quasi-Likelihood Ratio Tests

Koenker and Machado (1999) describe quasi-likelihood ratio tests based on the change in the optimized value of the objective function after relaxation of the restrictions imposed by

the null hypothesis. They offer two test statistics which they term *quantile- ρ* tests, though as Koenker (2005) points out, they may also be thought of as quasi-likelihood ratio tests.

We define the test statistics:

$$\begin{aligned} L_n(\tau) &= \frac{2(\tilde{V}(\tau) - \hat{V}(\tau))}{\tau(1-\tau)s(\tau)} \\ \Lambda_n(\tau) &= \frac{2\hat{V}(\tau)}{\tau(1-\tau)s(\tau)} \log(\tilde{V}(\tau)/\hat{V}(\tau)) \end{aligned} \quad (31.29)$$

which are both asymptotically χ_q^2 where q is the number of restrictions imposed by the null hypothesis.

You should note the presence of the sparsity term $s(\tau)$ in the denominator of both expressions. Any of the sparsity estimators outlined in “[Sparsity Estimation](#),” on page 273 may be employed for either the null or alternative specifications; EViews uses the sparsity estimated under the alternative. The presence of $s(\tau)$ should be a tipoff that these test statistics require that the quantile density function does not depend on X , as in the pure location-shift model.

Note that EViews will always compute an estimate of the scalar sparsity, even when you specify a Huber sandwich covariance method. This value of the sparsity will be used to compute QLR test statistics which may be less robust than the corresponding Wald counterparts.

Coefficient Tests

Given estimates of the asymptotic covariance matrix for the quantile regression estimates, you may construct Wald-type tests of hypotheses and construct coefficient confidence ellipses as in “[Coefficient Tests](#),” beginning on page 142.

Quantile Process Testing

The focus of our analysis thus far has been on the quantile regression model for a single quantile, τ . In a number of cases, we may instead be interested in forming joint hypotheses using coefficients for more than one quantile. We may, for example, be interested in evaluating whether the location-shift model is appropriate by testing for equality of slopes across quantile values. Consideration of more than one quantile regression at the same time comes under the general category of *quantile process* analysis.

While the EViews equation object is set up to consider only one quantile at a time, specialized tools allow you to perform the most commonly performed quantile process analyses.

Before proceeding to the hypothesis tests of interest, we must first outline the required distributional theory. Define the process coefficient vector:

$$\beta = (\beta(\tau_1)', \beta(\tau_2)', \dots, \beta(\tau_K'))' \quad (31.30)$$

Then

$$\sqrt{n}(\hat{\beta} - \beta) \sim N(0, \Omega) \quad (31.31)$$

where Ω has blocks of the form:

$$\Omega_{ij} = [\min(\tau_i, \tau_j) - \tau_i \tau_j] H^{-1}(\tau_i) J H^{-1}(\tau_j) \quad (31.32)$$

In the *i.i.d.* setting, Ω simplifies to,

$$\Omega = \Omega_0 \otimes J \quad (31.33)$$

where Ω_0 has representative element:

$$\omega_{ij} = \frac{\min(\tau_i, \tau_j) - \tau_i \tau_j}{f(F^{-1}(\tau_i))(f(F^{-1}(\tau_j)))} \quad (31.34)$$

Estimation of Ω may be performed directly using (31.32), (31.33) and (31.34), or using one of the bootstrap variants.

Slope Equality Testing

Koenker and Bassett (1982a) propose testing for slope equality across quantiles as a robust test of heteroskedasticity. The null hypothesis is given by:

$$H_0: \beta_1(\tau_1) = \beta_1(\tau_2) = \dots = \beta_1(\tau_K) \quad (31.35)$$

which imposes $(p-1)(K-1)$ restrictions on the coefficients. We may form the corresponding Wald statistic, which is distributed as a $\chi^2_{(p-1)(K-1)}$.

Symmetry Testing

Newey and Powell (1987) construct a test of the less restrictive hypothesis of symmetry, for asymmetric least squares estimators, but the approach may easily be applied to the quantile regression case.

The premise of the Newey and Powell test is that if the distribution of Y given X is symmetric, then:

$$\frac{\beta(\tau) + \beta(1-\tau)}{2} = \beta(1/2) \quad (31.36)$$

We may evaluate this restriction using Wald tests on the quantile process. Suppose that there are an odd number, K , of sets of estimated coefficients ordered by τ_k . The middle value $\tau_{(K+1)/2}$ is assumed to be equal to 0.5, and the remaining τ are symmetric around 0.5, with $\tau_j = 1 - \tau_{K-j+1}$, for $j = 1, \dots, (K-1)/2$. Then the Newey and Powell test null is the joint hypothesis that:

$$H_0: \frac{\beta(\tau_j) + \beta(\tau_{K-j+1})}{2} = \beta(1/2) \quad (31.37)$$

for $j = 1, \dots, (K - 1)/2$.

The Wald test formed for this null is zero under the null hypothesis of symmetry. The null has $p(K - 1)/2$ restrictions, so the Wald statistic is distributed as a $\chi^2_{p(K-1)/2}$. Newey and Powell point out that if it is known *a priori* that the errors are *i.i.d.*, but possibly asymmetric, one can restrict the null to only examine the restriction for the intercept. This restricted null imposes only $(K - 1)/2$ restrictions on the process coefficients.

References

- Barrodale I. and F. D. K. Roberts (1974). "Solution of an Overdetermined System of Equations in the l_1 Norm," *Communications of the ACM*, 17(6), 319-320.
- Bassett, Gilbert Jr. and Roger Koenker (1982). "An Empirical Quantile Function for Linear Models with *i.i.d.* Errors," *Journal of the American Statistical Association*, 77(378), 407-415.
- Bofinger, E. (1975). "Estimation of a Density Function Using Order Statistics," *Australian Journal of Statistics*, 17, 1-7.
- Buchinsky, M. (1995). "Estimating the Asymptotic Covariance Matrix for Quantile Regression Models: A Monte Carlo Study," *Journal of Econometrics*, 68, 303-338.
- Chamberlain, Gary (1994). "Quantile Regression, Censoring and the Structure of Wages," in *Advances in Econometrics*, Christopher Sims, ed., New York: Elsevier, 171-209.
- Falk, Michael (1986). "On the Estimation of the Quantile Density Function," *Statistics & Probability Letters*, 4, 69-73.
- Hall, Peter and Simon J. Sheather, "On the Distribution of the Studentized Quantile," *Journal of the Royal Statistical Society, Series B*, 50(3), 381-391.
- He, Xuming and Feifang Hu (2002). "Markov Chain Marginal Bootstrap," *Journal of the American Statistical Association*, 97(459), 783-795.
- Hendricks, Wallace and Roger Koenker (1992). "Hierarchical Spline Models for Conditional Quantiles and the Demand for Electricity," *Journal of the American Statistical Association*, 87(417), 58-68.
- Jones, M. C. (1992). "Estimating Densities, Quantiles, Quantile Densities and Density Quantiles," *Annals of the Institute of Statistical Mathematics*, 44(4), 721-727.
- Kocherginsky, Masha, Xuming He, and Yunming Mu (2005). "Practical Confidence Intervals for Regression Quantiles," *Journal of Computational and Graphical Statistics*, 14(1), 41-55.
- Koenker, Roger (1994), "Confidence Intervals for Regression Quantiles," in *Asymptotic Statistics*, P. Mandl and M. Huskova, eds., New York: Springer-Verlag, 349-359.
- Koenker, Roger (2005). *Quantile Regression*. New York: Cambridge University Press.
- Koenker, Roger and Gilbert Bassett, Jr. (1978). "Regression Quantiles," *Econometrica*, 46(1), 33-50.
- Koenker, Roger and Gilbert Bassett, Jr. (1982a). "Robust Tests for Heteroskedasticity Based on Regression Quantiles," *Econometrica*, 50(1), 43-62.
- Koenker, Roger and Gilbert Bassett, Jr. (1982b). "Tests of Linear Hypotheses and l_1 Estimation," *Econometrica*, 50(6), 1577-1584.
- Koenker, Roger W. and Vasco D'Orey (1987). "Algorithm AS 229: Computing Regression Quantiles," *Applied Statistics*, 36(3), 383-393.

- Koenker, Roger and Kevin F. Hallock (2001). "Quantile Regression," *Journal of Economic Perspectives*, 15(4), 143-156.
- Koenker, Roger and Jose A. F. Machado (1999). "Goodness of Fit and Related Inference Processes for Quantile Regression," *Journal of the American Statistical Association*, 94(448), 1296-1310.
- Newey, Whitney K., and James L. Powell (1987). "Asymmetric Least Squares Estimation," *Econometrica*, 55(4), 819-847.
- Portnoy, Stephen and Roger Koenker (1997), "The Gaussian Hare and the Laplacian Tortoise: Computability of Squared-Error versus Absolute-Error Estimators," *Statistical Science*, 12(4), 279-300.
- Powell, J. (1984). "Least Absolute Deviations Estimation for the Censored Regression Model," *Journal of Econometrics*, 25, 303-325.
- Powell, J. (1986). "Censored Regression Quantiles," *Journal of Econometrics*, 32, 143-155.
- Powell, J. (1989). "Estimation of Monotonic Regression Models Under Quantile Restrictions," in *Non-parametric and Semiparametric Methods in Econometrics*, W. Barnett, J. Powell, and G. Tauchen, eds., Cambridge: Cambridge University Press.
- Siddiqui, M. M. (1960). "Distribution of Quantiles in Samples from a Bivariate Population," *Journal of Research of the National Bureau of Standards-B*, 64(3), 145-150.
- Silverman, B. W. (1986). *Density Estimation for Statistics and Data Analysis*, London: Chapman & Hall.
- Welsh, A. H. (1988). "Asymptotically Efficient Estimation of the Sparsity Function at a Point," *Statistics & Probability Letters*, 6, 427-432.

Chapter 32. The Log Likelihood (LogL) Object

EViews contains customized procedures which help solve the majority of the estimation problems that you might encounter. On occasion, however, you may come across an estimation specification which is not included among these specialized routines. This specification may be an extension of an existing procedure, or it could be an entirely new class of problem.

Fortunately, EViews provides you with tools to estimate a wide variety of specifications through the *log likelihood* (*logl*) object. The *logl* object provides you with a general, open-ended tool for estimating a broad class of specifications by maximizing a likelihood function with respect to parameters.

When working with a log likelihood object, you will use EViews' series generation capabilities to describe the log likelihood contribution of each observation in your sample as a function of unknown parameters. You may supply analytical derivatives of the likelihood for one or more parameters, or you can simply let EViews calculate numeric derivatives automatically. EViews will search for the parameter values that maximize the specified likelihood function, and will provide estimated standard errors for these parameter estimates.

In this chapter, we provide an overview and describe the general features of the *logl* object. We also give examples of specifications which may be estimated using the object. The examples include: multinomial logit, unconditional maximum likelihood AR(1) estimation, Box-Cox regression, disequilibrium switching models, least squares with multiplicative heteroskedasticity, probit specifications with heteroskedasticity, probit with grouped data, nested logit, zero-altered Poisson models, Heckman sample selection models, Weibull hazard models, GARCH(1,1) with *t*-distributed errors, GARCH with coefficient restrictions, EGARCH with a generalized error distribution, and multivariate GARCH.

Overview

Most of the work in estimating a model using the *logl* object is in creating the text specification which will be used to evaluate the likelihood function.

If you are familiar with the process of generating series in EViews, you should find it easy to work with the *logl* specification, since the likelihood specification is merely a list of series assignment statements which are evaluated iteratively during the course of the maximization procedure. All you need to do is write down a set of statements which, when evaluated, will describe a series containing the contributions of each observation to the log likelihood function.

To take a simple example, suppose you believe that your data are generated by the conditional heteroskedasticity regression model:

$$\begin{aligned}
 y_t &= \beta_1 + \beta_2 x_t + \beta_3 z_t + \epsilon_t \\
 \epsilon_t &\sim N(0, \sigma^2 z_t^\alpha)
 \end{aligned}
 \tag{32.1}$$

where x , y , and z are the observed series (data) and $\beta_1, \beta_2, \beta_3, \sigma, \alpha$ are the parameters of the model. The log likelihood function (the log of the density of the observed data) for a sample of T observations can be written as:

$$\begin{aligned}
 l(\beta, \alpha, \sigma) &= -\frac{T}{2}(\log(2\pi) + \log \sigma^2) - \frac{\alpha}{2} \sum_{t=1}^T \log(z_t) - \sum_{t=1}^T \frac{(y_t - \beta_1 - \beta_2 x_t - \beta_3 z_t)^2}{\sigma^2 z_t^\alpha} \\
 &= \sum_{t=1}^T \left\{ \log \phi \left(\frac{y_t - \beta_1 - \beta_2 x_t - \beta_3 z_t}{\sigma z_t^{\alpha/2}} \right) - \frac{1}{2} \log(\sigma^2 z_t^\alpha) \right\}
 \end{aligned}
 \tag{32.2}$$

where ϕ is the standard normal density function.

Note that we can write the log likelihood function as a sum of the log likelihood contributions for each observation t :

$$l(\beta, \alpha, \sigma) = \sum_{t=1}^T l_t(\beta, \alpha, \sigma) \tag{32.3}$$

where the individual contributions are given by:

$$l_t(\beta, \alpha, \sigma) = \log \phi \left(\frac{y_t - \beta_1 - \beta_2 x_t - \beta_3 z_t}{\sigma z_t^{\alpha/2}} \right) - \frac{1}{2} \log(\sigma^2 z_t^\alpha) \tag{32.4}$$

Suppose that you know the true parameter values of the model, and you wish to generate a series in EViews which contains the contributions for each observation. To do this, you could assign the known values of the parameters to the elements C(1) to C(5) of the coefficient vector, and then execute the following list of assignment statements as commands or in an EViews program:

```

series res = y - c(1) - c(2)*x - c(3)*z
series var = c(4) * z^c(5)
series logl1 = log(@dnorm(res/@sqrt(var))) - log(var)/2

```

The first two statements describe series which will contain intermediate results used in the calculations. The first statement creates the residual series, RES, and the second statement creates the variance series, VAR. The series LOGL1 contains the set of log likelihood contributions for each observation.

Now suppose instead that you do not know the true parameter values of the model, and would like to estimate them from the data. The maximum likelihood estimates of the param-

eters are defined as the set of parameter values which produce the largest value of the likelihood function evaluated across all the observations in the sample.

The logl object makes finding these maximum likelihood estimates easy. Simply create a new log likelihood object, input the assignment statements above into the logl specification view, then ask EViews to estimate the specification.

In entering the assignment statements, you need only make two minor changes to the text above. First, the `series` keyword must be removed from the beginning of each line (since the likelihood specification implicitly assumes it is present). Second, an extra line must be added to the specification which identifies the name of the series in which the likelihood contributions will be contained. Thus, you should enter the following into your log likelihood object:

```
@logl logl1
res = y - c(1) - c(2)*x - c(3)*z
var = c(4) * z^c(5)
logl1 = log(@dnorm(res/@sqrt(var))) - log(var)/2
```

The first line in the log likelihood specification, `@logl logl1`, tells EViews that the series LOGL1 should be used to store the likelihood contributions. The remaining lines describe the computation of the intermediate results, and the actual likelihood contributions.

When you tell EViews to estimate the parameters of this model, it will execute the assignment statements in the specification repeatedly for different parameter values, using an iterative algorithm to search for the set of values that maximize the sum of the log likelihood contributions. When EViews can no longer improve the overall likelihood, it will stop iterating and will report final parameter values and estimated standard errors in the estimation output.

The remainder of this chapter discusses the rules for specification, estimation and testing using the likelihood object in greater detail.

Specification

To create a likelihood object, choose **Object/New Object.../LogL** or type “logl” in the command window. The likelihood window will open with a blank specification view. The specification view is a text window into which you enter a list of statements which describe your statistical model, and in which you set options which control various aspects of the estimation procedure.

Specifying the Likelihood

As described in the overview above, the core of the likelihood specification is a set of assignment statements which, when evaluated, generate a series containing the log likelihood con-

tribution of each observation in the sample. There can be as many or as few of these assignment statements as you wish.

Each likelihood specification must contain a control statement which provides the name of the series which is used to contain the likelihood contributions. The format of this statement is:

```
@logl series_name
```

where `series_name` is the name of the series which will contain the contributions. This control statement may appear anywhere in the logl specification.

Whenever the specification is evaluated, whether for estimation or for carrying out a View or Proc, each assignment statement will be evaluated at the current parameter values, and the results stored in a series with the specified name. If the series does not exist, it will be created automatically. If the series already exists, EViews will use the existing series for storage, and will overwrite the data contained in the series.

If you would like to remove one or more of the series used in the specification after evaluation, you can use the `@temp` statement, as in:

```
@temp series_name1 series_name2
```

This statement tells EViews to delete any series in the list after evaluation of the specification is completed. Deleting these series may be useful if your logl creates a lot of intermediate results, and you do not want the series containing these results to clutter your workfile.

Parameter Names

In the example above, we used the coefficients `C(1)` to `C(5)` as names for our unknown parameters. More generally, any element of a named coefficient vector which appears in the specification will be treated as a parameter to be estimated.

In the conditional heteroskedasticity example, you might choose to use coefficients from three different coefficient vectors: one vector for the mean equation, one for the variance equation, and one for the variance parameters. You would first create three named coefficient vectors by the commands:

```
coef(3) beta
coef(1) scale
coef(1) alpha
```

You could then write the likelihood specification as:

```
@logl logl1
res = y - beta(1) - beta(2)*x - beta(3)*z
var = scale(1)*z^alpha(1)
logl1 = log(@dnorm(res/@sqrt(var))) - log(var)/2
```

Since all elements of named coefficient vectors in the specification will be treated as parameters, you should make certain that all coefficients really do affect the value of one or more of the likelihood contributions. If a parameter has no effect upon the likelihood, you will experience a singularity error when you attempt to estimate the parameters.

Note that all objects other than coefficient elements will be considered fixed and will not be updated during estimation. For example, suppose that SIGMA is a named scalar in your workfile. Then if you redefine the subexpression for VAR as:

```
var = sigma*z^alpha(1)
```

EViews will not estimate SIGMA. The value of SIGMA will remain fixed at its value at the start of estimation.

Order of Evaluation

The logl specification contains one or more assignment statements which generate the series containing the likelihood contributions. EViews always evaluates from top to bottom when executing these assignment statements, so expressions which are used in subsequent calculations should always be placed first.

EViews must also iterate through the observations in the sample. Since EViews iterates through both the equations in the specification and the observations in the sample, you will need to specify the order in which the evaluation of observations and equations occurs.

By default, EViews evaluates the specification *by observation* so that *all of the assignment statements* are evaluated for the *first observation*, then for the second observation, and so on across all the observations in the estimation sample. This is the correct order for recursive models where the likelihood of an observation depends on previously observed (lagged) values, as in AR or ARCH models.

You can change the order of evaluation so EViews evaluates the specification *by equation*, so the *first assignment statement* is evaluated for *all the observations*, then the second assignment statement is evaluated for all the observations, and so on for each of the assignment statements in the specification. This is the correct order for models where aggregate statistics from intermediate series are used as input to subsequent calculations.

You can explicitly select which method of evaluation you would like by adding a statement to the likelihood specification. To force evaluation by equation, simply add a line containing the keyword “@byeqn”. To explicitly state that you require evaluation by observation, the “@byobs” keyword can be used. If no keyword is provided, @byobs is assumed.

In the conditional heteroskedasticity example above, it does not matter whether the assignment statements are evaluated by equation (line by line) or by observation, since the results do not depend upon the order of evaluation.

However, if the specification has a recursive structure, or if the specification requires the calculation of aggregate statistics based on intermediate series, you must select the appropriate evaluation order if the calculations are to be carried out correctly.

As an example of the `@byeqn` statement, consider the following specification:

```
@logl robust1
@byeqn
res1 = y-c(1)-c(2)*x
delta = @abs(res1)/6/@median(@abs(res1))
weight = (delta<1)*(1-delta^2)^2
robust1 = -(weight*res1^2)
```

This specification performs robust regression by downweighting outlier residuals at each iteration. The assignment statement for DELTA computes the median of the absolute value of the residuals in each iteration, and this is used as a reference point for forming a weighting function for outliers. The `@byeqn` statement instructs EViews to compute all residuals RES1 at a given iteration before computing the median of those residuals when calculating the DELTA series.

Analytic Derivatives

By default, when maximizing the likelihood and forming estimates of the standard errors, EViews computes numeric derivatives of the likelihood function with respect to the parameters. If you would like to specify an analytic expression for one or more of the derivatives, you may use the `@deriv` statement. The `@deriv` statement has the form:

```
@deriv pname1 sname1 pname2 sname2 ...
```

where `pname` is a parameter in the model and `sname` is the name of the corresponding derivative series generated by the specification.

For example, consider the following likelihood object that specifies a multinomial logit model:

```
' multinomial logit with 3 outcomes
@logl logl1
xb2 = b2(1)+b2(2)*x1+b2(3)*x2
xb3 = b3(1)+b3(2)*x1+b3(3)*x2
denom = 1+exp(xb2)+exp(xb3)
' derivatives wrt the 2nd outcome params
@deriv b2(1) grad21 b2(2) grad22 b2(3) grad23
grad21 = d2-exp(xb2)/denom
```

```

grad22 = grad21*x1
grad23 = grad21*x2
' derivatives wrt the 3rd outcome params
@deriv b3(1) grad31 b3(2) grad32 b3(3) grad33
grad31 = d3-exp(xb3)/denom
grad32 = grad31*x1
grad33 = grad31*x2
' specify log likelihood
logl1 = d2*xb2+d3*xb3-log(1+exp(xb2)+exp(xb3))

```

See Greene (1997), Chapter 19.7.1 for a discussion of multinomial logit models. There are three possible outcomes, and the parameters of the three regressors (X1, X2 and the constant) are normalized relative to the first outcome. The analytic derivatives are particularly simple for the multinomial logit model and the two `@deriv` statements in the specification instruct EViews to use the expressions for GRAD21, GRAD22, GRAD23, GRAD31, GRAD32, and GRAD33, instead of computing numeric derivatives.

When working with analytic derivatives, you may wish to check the validity of your expressions for the derivatives by comparing them with numerically computed derivatives. EViews provides you with tools which will perform this comparison at the current values of parameters or at the specified starting values. See the discussion of the **Check Derivatives** view of the likelihood object in [“Check Derivatives” on page 295](#).

Derivative Step Sizes

If analytic derivatives are not specified for any of your parameters, EViews numerically evaluates the derivatives of the likelihood function for those parameters. The step sizes used in computing the derivatives are controlled by two parameters: r (relative step size) and m (minimum step size). Let $\theta^{(i)}$ denote the value of the parameter θ at iteration i . Then the step size at iteration $i + 1$ is determined by:

$$s^{(i+1)} = \max(r\theta^{(i)}, m) \quad (32.5)$$

The two-sided numeric derivative is evaluated as:

$$\frac{f(\theta^{(i)} + s^{(i+1)}) - f(\theta^{(i)} - s^{(i+1)})}{2s^{(i+1)}} \quad (32.6)$$

The one-sided numeric derivative is evaluated as:

$$\frac{f(\theta^{(i)} + s^{(i+1)}) - f(\theta^{(i)})}{s^{(i+1)}} \quad (32.7)$$

where f is the likelihood function. Two-sided derivatives are more accurate, but require roughly twice as many evaluations of the likelihood function and so take about twice as long to evaluate.

The `@derivstep` statement can be used to control the step size and method used to evaluate the derivative at each iteration. The `@derivstep` keyword should be followed by sets of three arguments: the name of the parameter to be set (or the keyword `@all`), the relative step size, and the minimum step size.

The default setting is (approximately):

```
@derivstep(1) @all 1.49e-8 1e-10
```

where “1” in the parentheses indicates that one-sided numeric derivatives should be used and `@all` indicates that the following setting applies to all of the parameters. The first number following `@all` is the relative step size and the second number is the minimum step size. The default relative step size is set to the square root of machine epsilon (1.49×10^{-8}) and the minimum step size is set to $m = 10^{-10}$.

The step size can be set separately for each parameter in a single or in multiple `@derivstep` statements. The evaluation method option specified in parentheses is a global option; it cannot be specified separately for each parameter.

For example, if you include the line:

```
@derivstep(2) c(2) 1e-7 1e-10
```

the relative step size for coefficient C(2) will be increased to $m = 10^{-7}$ and a two-sided derivative will be used to evaluate the derivative. In a more complex example,

```
@derivstep(2) @all 1.49e-8 1e-10 c(2) 1e-7 1e-10 c(3) 1e-5 1e-8
```

computes two-sided derivatives using the default step sizes for all coefficients except C(2) and C(3). The values for these latter coefficients are specified directly.

Estimation

Once you have specified the `logl` object, you can ask EViews to find the parameter values which maximize the likelihood parameters. Simply click the **Estimate** button in the likelihood window toolbar to open the **Estimation Options** dialog.

There are a number of options which allow you to control various aspects of the estimation procedure. See [“Setting Estimation Options” on page 625](#) for a discussion of these options. The default settings, however, should provide a good start for most problems. When you click on **OK**, EViews will begin estimation using the current settings.

Starting Values

Since EViews uses an iterative algorithm to find the maximum likelihood estimates, the choice of starting values is important. For problems in which the likelihood function is globally concave, it will influence how many iterations are taken for estimation to converge. For problems where the likelihood function is not concave, it may determine which of several local maxima is found. In some cases, estimation will fail unless reasonable starting values are provided.

By default, EViews uses the values stored in the coefficient vector or vectors prior to estimation. If a `@param` statement is included in the specification, the values specified in the statement will be used instead.

In our conditional heteroskedasticity regression example, one choice for starting values for the coefficients of the mean equation coefficients are the simple OLS estimates, since OLS provides consistent point estimates even in the presence of (bounded) heteroskedasticity. To use the OLS estimates as starting values, first estimate the OLS equation by the command:

```
equation eq1.ls y c x z
```

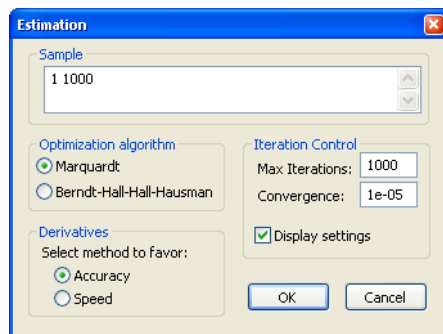
After estimating this equation, the elements `C(1)`, `C(2)`, `C(3)` of the `C` coefficient vector will contain the OLS estimates. To set the variance scale parameter `C(4)` to the estimated OLS residual variance, you can type the assignment statement in the command window:

```
c(4) = eq1.@se^2
```

For the final heteroskedasticity parameter `C(5)`, you can use the residuals from the original OLS regression to carry out a second OLS regression, and set the value of `C(5)` to the appropriate coefficient. Alternatively, you can arbitrarily set the parameter value using a simple assignment statement:

```
c(5) = 1
```

Now, if you estimate the logit specification immediately after carrying out the OLS estimation and subsequent commands, it will use the values that you have placed in the `C` vector as starting values.



As noted above, an alternative method of initializing the parameters to known values is to include a `@param` statement in the likelihood specification. For example, if you include the line:

```
@param c(1) 0.1 c(2) 0.1 c(3) 0.1 c(4) 1 c(5) 1
```

in the specification of the `logl`, EViews will always set the starting values to $C(1) = C(2) = C(3) = 0.1$, $C(4) = C(5) = 1$.

See also the discussion of starting values in [“Starting Coefficient Values” on page 625](#).

Estimation Sample

EViews uses the sample of observations specified in the Estimation Options dialog when estimating the parameters of the log likelihood. EViews evaluates each expression in the `logl` for every observation in the sample at current parameter values, using the *by observation* or *by equation* ordering. All of these evaluations follow the standard EViews rules for evaluating series expressions.

If there are missing values in the log likelihood series at the initial parameter values, EViews will issue an error message and the estimation procedure will stop. In contrast to the behavior of other EViews built-in procedures, `logl` estimation performs no endpoint adjustments or dropping of observations with missing values when estimating the parameters of the model.

LogL Views

- **Likelihood Specification:** displays the window where you specify and edit the likelihood specification.
- **Estimation Output:** displays the estimation results obtained from maximizing the likelihood function.
- **Covariance Matrix:** displays the estimated covariance matrix of the parameter estimates. These are computed from the inverse of the sum of the outer product of the first derivatives evaluated at the optimum parameter values. To save this covariance matrix as a (SYM) MATRIX, you may use the `@cov` function.
- **Wald Coefficient Tests...:** performs the Wald coefficient restriction test. See [“Wald Test \(Coefficient Restrictions\)” on page 145](#), for a discussion of Wald tests.
- **Gradients:** displays view of the gradients (first derivatives) of the log likelihood at the current parameter values (if the model has not yet been estimated), or at the converged parameter values (if the model has been estimated). These views may prove to be useful diagnostic tools if you are experiencing problems with convergence.
- **Check Derivatives:** displays the values of the numeric derivatives and analytic derivatives (if available) at the starting values (if a `@param` statement is included), or at current parameter values (if there is no `@param` statement).

LogL Procs

- **Estimate...:** brings up a dialog to set estimation options, and to estimate the parameters of the log likelihood.
- **Make Model:** creates an untitled model object out of the estimated likelihood specification.
- **Make Gradient Group:** creates an untitled group of the gradients (first derivatives) of the log likelihood at the estimated parameter values. These gradients are often used in constructing Lagrange multiplier tests.
- **Update Coefs from LogL:** updates the coefficient vector(s) with the estimates from the likelihood object. This procedure allows you to export the maximum likelihood estimates for use as starting values in other estimation problems.

Most of these procedures should be familiar to you from other EViews estimation objects. We describe below the features that are specific to the logl object.

Estimation Output

In addition to the coefficient and standard error estimates, the standard output for the logl object describes the method of estimation, sample used in estimation, date and time that the logl was estimated, evaluation order, and information about the convergence of the estimation procedure.

LogL: MLOGIT
Method: Maximum Likelihood (Marquardt)
Date: 01/16/04 Time: 08:36
Sample: 1 1000
Included observations: 1000
Evaluation order: By observation
Estimation settings: tol= 1.0e-09, derivs=analytic
Initial Values: B2(1)=-1.08356, B2(2)=0.90467, B2(3)=-0.06786, B3(1)=-0.69842, B3(2)=-0.33212, B3(3)=0.32981
Convergence achieved after 7 iterations

| | Coefficient | Std. Error | z-Statistic | Prob. |
|---------------------|-------------|-----------------------|-------------|--------|
| B2(1) | -0.521793 | 0.205568 | -2.538302 | 0.0111 |
| B2(2) | 0.994358 | 0.267963 | 3.710798 | 0.0002 |
| B2(3) | 0.134983 | 0.265655 | 0.508115 | 0.6114 |
| B3(1) | -0.262307 | 0.207174 | -1.266122 | 0.2055 |
| B3(2) | 0.176770 | 0.274756 | 0.643371 | 0.5200 |
| B3(3) | 0.399166 | 0.274056 | 1.456511 | 0.1453 |
| Log likelihood | -1089.415 | Akaike info criterion | 2.190830 | |
| Avg. log likelihood | -1.089415 | Schwarz criterion | 2.220277 | |
| Number of Coefs. | 6 | Hannan-Quinn criter. | 2.202022 | |

EViews also provides the log likelihood value, average log likelihood value, number of coefficients, and three Information Criteria. By default, the starting values are not displayed.

Here, we have used the **Estimation Options** dialog to instruct EViews to display the estimation starting values in the output.

Gradients

The gradient summary table and gradient summary graph view allow you to examine the gradients of the likelihood. These gradients are computed at the current parameter values (if the model has not yet been estimated), or at the converged parameter values (if the model has been estimated). See [Appendix E. “Gradients and Derivatives,” on page 637](#) for additional details.

You may find this view to be a useful diagnostic tool when experiencing problems with convergence or singularity. One common problem leading to singular matrices is a zero derivative for a parameter due to an incorrectly specified likelihood, poor starting values, or a lack of model identification. See the discussion below for further details.

| obs | B2(1) | B2(2) | B2(3) | B3(1) | B3 |
|-----|-----------|-----------|-----------|-----------|------|
| 1 | 0.617633 | 0.446035 | 0.449061 | -0.332790 | -0.2 |
| 2 | 0.660916 | 0.310714 | 0.074241 | -0.308290 | -0.1 |
| 3 | 0.665234 | 0.330282 | 0.518881 | -0.355333 | -0.1 |
| 4 | -0.420696 | -0.360283 | -0.076988 | -0.284167 | -0.2 |
| 5 | 0.576206 | 0.516712 | 0.306708 | -0.303744 | -0.2 |
| 6 | -0.394766 | -0.296582 | -0.147463 | 0.694417 | 0.5 |
| 7 | -0.386750 | -0.269690 | -0.047604 | -0.292850 | -0.2 |
| 8 | 0.582613 | 0.509899 | 0.362163 | -0.311741 | -0.2 |
| 9 | -0.288308 | -0.076096 | -0.252996 | -0.379763 | -0.1 |
| 10 | 0.553928 | 0.549244 | 0.256273 | -0.290475 | -0.2 |
| 11 | -0.301615 | -0.086085 | -0.056264 | 0.674758 | 0.1 |
| 12 | -0.259970 | -0.029439 | -0.251003 | -0.396446 | -0.0 |
| 13 | 0.689356 | 0.244054 | 0.363363 | -0.346518 | -0.1 |
| 14 | 0.735086 | 0.057760 | 0.005284 | -0.322644 | -0.0 |
| 15 | | | | | |

Check Derivatives

You can use the **Check Derivatives** view to examine your numeric derivatives or to check the validity of your expressions for the analytic derivatives. If the logl specification contains a @param statement, the derivatives will be evaluated at the specified values, otherwise, the derivatives will be computed at the current coefficient values.

Consider the derivative view for coefficients estimated using two binary logit specifications. The first part of this view displays the names of the user supplied derivatives, step size parameters, and the coefficient values at which the derivatives are evaluated. The relative and minimum step sizes shown in this example are the default settings.

The second part of the view computes the sum (over all individuals in the sample) of the numeric and, if applicable, the analytic derivatives for each coefficient. If appropriate, EViews will also compute the largest individual difference between the analytic and the numeric derivatives in both absolute, and percentage terms.

| Coefficient | User | Rel. Step | Min. Step | Coef. Value |
|-------------|--------|-----------|-----------|-------------|
| B2(1) | GRAD21 | 1.49E-08 | 1.00E-10 | -1.083560 |
| B2(2) | GRAD22 | 1.49E-08 | 1.00E-10 | 0.904669 |
| B2(3) | GRAD23 | 1.49E-08 | 1.00E-10 | -0.067858 |
| B3(1) | GRAD31 | 1.49E-08 | 1.00E-10 | -0.698420 |
| B3(2) | GRAD32 | 1.49E-08 | 1.00E-10 | -0.332125 |
| B3(3) | GRAD33 | 1.49E-08 | 1.00E-10 | 0.329809 |

| Coefficient | Sum Over Observations | | Maximum Difference | |
|-------------|-----------------------|----------|--------------------|----------|
| | Numeric | User | Absolute | Percent |
| B2(1) | 82.78805 | 82.78805 | -4.85E-08 | 1.92E-05 |
| B2(2) | 43.00270 | 43.00270 | 5.04E-08 | 0.001841 |
| B2(3) | 43.10014 | 43.10014 | 6.09E-07 | 0.057752 |
| B3(1) | 85.35492 | 85.35492 | 6.50E-08 | 2.39E-05 |
| B3(2) | 47.27110 | 47.27110 | -1.53E-07 | 0.005156 |
| B3(3) | 43.55109 | 43.55108 | -1.30E-07 | 0.025601 |

Troubleshooting

Because the logl object provides a great deal of flexibility, you are more likely to experience problems with estimation using the logl object than with EViews' built-in estimators.

If you are experiencing difficulties with estimation the following suggestions may help you in solving your problem:

- **Check your likelihood specification.** A simple error involving a wrong sign can easily stop the estimation process from working. You should also verify that the parameters of the model are really identified (in some specifications you may have to impose a normalization across the parameters). Also, every parameter which appears in the model must feed directly or indirectly into the likelihood contributions. The **Check Derivatives** view is particularly useful in helping you spot the latter problem.
- **Choose your starting values.** If any of the likelihood contributions in your sample cannot be evaluated due to missing values or because of domain errors in mathematical operations (logs and square roots of negative numbers, division by zero, etc.) the estimation will stop immediately with the message: "Cannot compute @logl due to missing values". In other cases, a bad choice of starting values may lead you into regions where the likelihood function is poorly behaved. You should always try to initialize your parameters to sensible numerical values. If you have a simpler estimation technique available which approximates the problem, you may wish to use estimates from this method as starting values for the maximum likelihood specification.
- **Make sure lagged values are initialized correctly.** In contrast to most other estimation routines in EViews, the logl estimation procedure will not automatically drop observations with NAs or lags from the sample when estimating a log likelihood model. If your likelihood specification involves lags, you will either have to drop observations from the beginning of your estimation sample, or you will have to carefully code the specification so that missing values from before the sample do not cause NAs to propagate through the entire sample (see the AR(1) and GARCH examples for a demonstration).

Since the series used to evaluate the likelihood are contained in your workfile (unless you use the `@temp` statement to delete them), you can examine the values in the log likelihood and intermediate series to find problems involving lags and missing values.

- **Verify your derivatives.** If you are using analytic derivatives, use the **Check Derivatives** view to make sure you have coded the derivatives correctly. If you are using numerical derivatives, consider specifying analytic derivatives or adjusting the options for derivative method or step size.
- **Reparametrize your model.** If you are having problems with parameter values causing mathematical errors, you may wish to consider reparameterizing the model to restrict the parameter within its valid domain. See the discussion below for examples.

Most of the error messages you are likely to see during estimation are self-explanatory. The error message “near singular matrix” may be less obvious. This error message occurs when EViews is unable to invert the matrix of the sum of the outer product of the derivatives so that it is impossible to determine the direction of the next step of the optimization. This error may indicate a wide variety of problems, including bad starting values, but will almost always occur if the model is not identified, either theoretically, or in terms of the available data.

Limitations

The likelihood object can be used to estimate parameters that maximize (or minimize) a variety of objective functions. Although the main use of the likelihood object will be to specify a log likelihood, you can specify least squares and minimum distance estimation problems with the likelihood object as long as the objective function is additive over the sample.

You should be aware that the algorithm used in estimating the parameters of the log likelihood is not well suited to solving arbitrary maximization or minimization problems. The algorithm forms an approximation to the Hessian of the log likelihood, based on the sum of the outer product of the derivatives of the likelihood contributions. This approximation relies on both the functional form and statistical properties of maximum likelihood objective functions, and may not be a good approximation in general settings. Consequently, you may or may not be able to obtain results with other functional forms. Furthermore, the standard error estimates of the parameter values will only have meaning if the series describing the log likelihood contributions are (up to an additive constant) the individual contributions to a correctly specified, well-defined theoretical log likelihood.

Currently, the expressions used to describe the likelihood contribution must follow the rules of EViews series expressions. This restriction implies that we do not allow matrix operations in the likelihood specification. In order to specify likelihood functions for multiple equation models, you may have to write out the expression for the determinants and quadratic forms. Although possible, this may become tedious for models with more than two or three equations. See the multivariate GARCH sample programs for examples of this approach.

Additionally, the `logl` object does not directly handle optimization subject to general inequality constraints. There are, however, a variety of well-established techniques for imposing simple inequality constraints. We provide examples below. The underlying idea is to apply a monotonic transformation to the coefficient so that the new coefficient term takes on values only in the desired range. The commonly used transformations are the `@exp` for one-sided restrictions and the `@logit` and `@arctan` for two-sided restrictions.

You should be aware of the limitations of the transformation approach. First, the approach only works for relatively simple inequality constraints. If you have several cross-coefficient inequality restrictions, the solution will quickly become intractable. Second, in order to per-

form hypothesis tests on the untransformed coefficient, you will have to obtain an estimate of the standard errors of the associated expressions. Since the transformations are generally nonlinear, you will have to compute linear approximations to the variances yourself (using the delta method). Lastly, inference will be poor near the boundary values of the inequality restrictions.

Simple One-Sided Restrictions

Suppose you would like to restrict the estimate of the coefficient of X to be no larger than 1. One way you could do this is to specify the corresponding subexpression as follows:

```
' restrict coef on x to not exceed 1
res1 = y - c(1) - (1-exp(c(2)))*x
```

Note that EViews will report the point estimate and the standard error for the parameter $C(2)$, not the coefficient of X . To find the standard error of the expression $1 - \exp(c(2))$, you will have to use the delta method; see for example Greene (1997), Theorems 4.15 and 4.16.

Simple Two-Sided Restrictions

Suppose instead that you want to restrict the coefficient for X to be between -1 and 1. Then you can specify the expression as:

```
' restrict coef on x to be between -1 and 1
res1 = y - c(1) - (2*@logit(c(2))-1)*x
```

Again, EViews will report the point estimate and standard error for the parameter $C(2)$. You will have to use the delta method to compute the standard error of the transformation expression $2 * @logit(c(2)) - 1$.

More generally, if you want to restrict the parameter to lie between L and H , you can use the transformation:

```
(H-L)*@logit(c(1)) + L
```

where $C(1)$ is the parameter to be estimated. In the above example, $L = -1$ and $H = 1$.

Examples

In this section, we provide extended examples of working with the logl object to estimate a multinomial logit and a maximum likelihood AR(1) specification. Example programs for these and several other specifications are provided in your default EViews data directory. If you set your default directory to point to the EViews data directory, you should be able to issue a RUN command for each of these programs to create the logl object and to estimate the unknown parameters.

Multinomial Logit (mlogit1.prg)

In this example, we demonstrate how to specify and estimate a simple multinomial logit model using the `logl` object. Suppose the dependent variable Y can take one of three categories 1, 2, and 3. Further suppose that there are data on two regressors, X_1 and X_2 that vary across observations (individuals). Standard examples include variables such as age and level of education. Then the multinomial logit model assumes that the probability of observing each category in Y is given by:

$$\Pr(y_i = j) = \frac{\exp(\beta_{0j} + \beta_{1j}x_{1i} + \beta_{2j}x_{2i})}{\sum_{k=1}^3 \exp(\beta_{0k} + \beta_{1k}x_{1i} + \beta_{2k}x_{2i})} = P_{ij} \quad (32.8)$$

for $j = 1, 2, 3$. Note that the parameters β are specific to each category so there are $3 \times 3 = 9$ parameters in this specification. The parameters are not all identified unless we impose a normalization (see for example Greene, 1997, Chapter 19.7), so we normalize the parameters of the first choice category $j = 1$ to be all zero: $\beta_{0,1} = \beta_{1,1} = \beta_{2,1} = 0$.

The log likelihood function for the multinomial logit can be written as:

$$l = \sum_{i=1}^N \sum_{j=1}^3 d_{ij} \log(P_{ij}) \quad (32.9)$$

where d_{ij} is a dummy variable that takes the value 1 if observation i has chosen alternative j and 0 otherwise. The first-order conditions are:

$$\frac{\partial l}{\partial \beta_{kj}} = \sum_{i=1}^N (d_{ij} - P_{ij}) x_{ki} \quad (32.10)$$

for $k = 0, 1, 2$ and $j = 1, 2, 3$.

We have provided, in the Example Files subdirectory of your default EViews directory, a workfile `MLOGIT.WK1` containing artificial multinomial data. The program begins by loading this workfile:

```
' load artificial data
%evworkfile = @evpath + "\example files\logl\mlogit"
load "{%evworkfile}"
```

from the EViews example directory.

Next, we declare the coefficient vectors that will contain the estimated parameters for each choice alternative:

```
' declare parameter vector
coef(3) b2
```

```
coef(3) b3
```

As an alternative, we could have used the default coefficient vector `C`.

We then set up the likelihood function by issuing a series of `append` statements:

```
mlogit.append xb2 = b2(1)+b2(2)*x1+b2(3)*x2
mlogit.append xb3 = b3(1)+b3(2)*x1+b3(3)*x2
' define prob for each choice
mlogit.append denom = 1+exp(xb2)+exp(xb3)
mlogit.append pr1 = 1/denom
mlogit.append pr2 = exp(xb2)/denom
mlogit.append pr3 = exp(xb3)/denom
' specify likelihood
mlogit.append logll = (1-dd2-dd3)*log(pr1)
                    +dd2*log(pr2)+dd3*log(pr3)
```

Since the analytic derivatives for the multinomial logit are particularly simple, we also specify the expressions for the analytic derivatives to be used during estimation and the appropriate `@deriv` statements:

```
' specify analytic derivatives
for!i = 2 to 3
mlogit.append @deriv b{!i}(1) grad{!i}1 b{!i}(2) grad{!i}2
                  b{!i}(3) grad{!i}3
mlogit.append grad{!i}1 = dd{!i}-pr{!i}
mlogit.append grad{!i}2 = grad{!i}1*x1
mlogit.append grad{!i}3 = grad{!i}1*x2
next
```

Note that if you were to specify this likelihood interactively, you would simply type the expression that follows each `append` statement directly into the `MLOGIT` object.

This concludes the actual specification of the likelihood object. Before estimating the model, we get the starting values by estimating a series of binary logit models:

```
' get starting values from binomial logit
equation eq2.binary(d=1) dd2 c x1 x2
b2 = eq2.@coefs
equation eq3.binary(d=1) dd3 c x1 x2
b3 = eq3.@coefs
```

To check whether you have specified the analytic derivatives correctly, choose **View/Check Derivatives** or use the command:

```
show mlogit.checkderiv
```

If you have correctly specified the analytic derivatives, they should be fairly close to the numeric derivatives.

We are now ready to estimate the model. Either click the **Estimate** button or use the command:

```
' do MLE
mlogit.ml(showopts, m=1000, c=1e-5)
show mlogit.output
```

Note that you can examine the derivatives for this model using the **Gradient Table** view, or you can examine the series in the workfile containing the gradients. You can also look at the intermediate results and log likelihood values. For example, to look at the likelihood contributions for each individual, simply double click on the LOGL1 series.

AR(1) Model (ar1.prg)

In this example, we demonstrate how to obtain full maximum likelihood estimates of an AR(1). The maximum likelihood procedure uses the first observation in the sample, in contrast to the built-in AR(1) procedure in EViews which treats the first observation as fixed and maximizes the conditional likelihood for the remaining observations by nonlinear least squares.

As an illustration, we first generate data that follows an AR(1) process:

```
' make up data
create m 80 89
rndseed 123
series y=0
smpl @first+1 @last
y = 1+0.85*y(-1) + nrnd
```

The exact Gaussian likelihood function for an AR(1) model is given by:

$$f(y, \theta) = \begin{cases} \frac{1}{\sigma\sqrt{2\pi(1-\rho^2)}} \exp\left\{-\frac{(y_t - c/(1-\rho^2))^2}{2(\sigma^2/(1-\rho^2))}\right\} & t = 1 \\ \frac{1}{\sigma\sqrt{2\pi}} \exp\left\{-\frac{(y_t - c - \rho y_{t-1})^2}{2(\sigma^2)}\right\} & t > 0 \end{cases} \quad (32.11)$$

where c is the constant term, ρ is the AR(1) coefficient, and σ^2 is the error variance, all to be estimated (see for example Hamilton, 1994a, Chapter 5.2).

Since the likelihood function evaluation differs for the first observation in our sample, we create a dummy variable indicator for the first observation:

```
' create dummy variable for first obs
series d1 = 0
smpl @first @first
d1 = 1
smpl @all
```

Next, we declare the coefficient vectors to store the parameter estimates and initialize them with the least squares estimates:

```
' set starting values to LS (drops first obs)
equation eq1.ls y c ar(1)
coef(1) rho = c(2)
coef(1) s2 = eq1.@se^2
```

We then specify the likelihood function. We make use of the `@recode` function to differentiate the evaluation of the likelihood for the first observation from the remaining observations. Note: the `@recode` function used here uses the updated syntax for this function—please double-check the current documentation for details.

```
' set up likelihood
logl ar1
ar1.append @logl logl1
ar1.append var = @recode(d1=1,s2(1)/(1-rho(1)^2),s2(1))
ar1.append res = @recode(d1=1,y-c(1)/(1-rho(1)),y-c(1)-rho(1)*y(-1))
ar1.append sres = res/@sqrt(var)
ar1.append logl1 = log(@dnorm(sres))-log(var)/2
```

The likelihood specification uses the built-in function `@dnorm` for the standard normal density. The second term is the Jacobian term that arises from transforming the standard normal variable to one with non-unit variance. (You could, of course, write out the likelihood for the normal distribution without using the `@dnorm` function.)

The program displays the MLE together with the least squares estimates:

```
' do MLE
ar1.ml(showopts, m=1000, c=1e-5)
show ar1.output
```

```
' compare with EViews AR(1) which ignores first obs
show eq1.output
```

Additional Examples

The following additional example programs can be found in the “Example Files” subdirectory of your default EViews directory.

- **Conditional logit** (clogit1.prg): estimates a conditional logit with 3 outcomes and both individual specific and choice specific regressors. The program also displays the prediction table and carries out a Hausman test for independence of irrelevant alternatives (IIA). See Greene (1997, Chapter 19.7) for a discussion of multinomial logit models.
- **Box-Cox transformation** (boxcox1.prg): estimates a simple bivariate regression with an estimated Box-Cox transformation on both the dependent and independent variables. Box-Cox transformation models are notoriously difficult to estimate and the results are very sensitive to starting values.
- **Disequilibrium switching model** (diseq1.prg): estimates the switching model in exercise 15.14–15.15 of Judge et al. (1985, pages 644–646). Note that there are some typos in Judge et al. (1985, pages 639–640). The program uses the likelihood specification in Quandt (1988, page 32, equations 2.3.16–2.3.17).
- **Multiplicative heteroskedasticity** (hetero1.prg): estimates a linear regression model with multiplicative heteroskedasticity. Replicates the results in Greene (1997, example 12.14).
- **Probit with heteroskedasticity** (hprobit1.prg): estimates a probit specification with multiplicative heteroskedasticity. See Greene (1997, example 19.7).
- **Probit with grouped data** (gprobit1.prg): estimates a probit with grouped data (proportions data). Estimates the model in Greene (1997, exercise 19.6).
- **Nested logit** (nlogit1.prg): estimates a nested logit model with 2 branches. Tests the IIA assumption by a Wald test. See Greene (1997, Chapter 19.7.4) for a discussion of nested logit models.
- **Zero-altered Poisson model** (zpoiss1.prg): estimates the zero-altered Poisson model. Also carries out the non-nested LR test of Vuong (1989). See Greene (1997, Chapter 19.9.6) for a discussion of zero-altered Poisson models and Vuong’s non-nested likelihood ratio test.
- **Heckman sample selection model** (heckman1.prg): estimates Heckman’s two equation sample selection model by MLE using the two-step estimates as starting values.

- **Weibull hazard model** (`weibull1.prg`): estimates the uncensored Weibull hazard model described in Greene (1997, example 20.18). The program also carries out one of the conditional moment tests in Greene (1997, example 20.19).
- **GARCH(1,1) with t-distributed errors** (`arch_t1.prg`): estimates a GARCH(1,1) model with t -distribution. The log likelihood function for this model can be found in Hamilton (1994a, equation 21.1.24, page 662). Note that this model may more easily be estimated using the standard ARCH estimation tools provided in EViews ([Chapter 29. “ARCH and GARCH Estimation,” on page 185](#)).
- **GARCH with coefficient restrictions** (`garch1.prg`): estimates an MA(1)-GARCH(1,1) model with coefficient restrictions in the conditional variance equation. This model is estimated by Bollerslev, Engle, and Nelson (1994, equation 9.1, page 3015) for different data.
- **EGARCH with generalized error distributed errors** (`egarch1.prg`): estimates Nelson’s (1991) exponential GARCH with generalized error distribution. The specification and likelihood are described in Hamilton (1994a, pages 668–669). Note that this model may more easily be estimated using the standard ARCH estimation tools provided in EViews ([Chapter 29. “ARCH and GARCH Estimation,” on page 185](#)).
- **Multivariate GARCH** (`bv_garch.prg` and `tv_garch.prg`): estimates the bi- or the tri-variate version of the BEKK GARCH specification (Engle and Kroner, 1995). Note that this specification may be estimated using the built-in procedures available in the system object ([“System Estimation,” on page 307](#)).

References

- Bollerslev, Tim, Robert F. Engle and Daniel B. Nelson (1994). “ARCH Models,” Chapter 49 in Robert F. Engle and Daniel L. McFadden (*eds.*), *Handbook of Econometrics, Volume 4*, Amsterdam: Elsevier Science B.V.
- Engle, Robert F. and K. F. Kroner (1995). “Multivariate Simultaneous Generalized ARCH,” *Econometric Theory*, 11, 122-150.
- Greene, William H. (1997). *Econometric Analysis*, 3rd Edition, Upper Saddle River, NJ: Prentice Hall.
- Hamilton, James D. (1994a). *Time Series Analysis*, Princeton University Press.
- Judge, George G., W. E. Griffiths, R. Carter Hill, Helmut Lütkepohl, and Tsoung-Chao Lee (1985). *The Theory and Practice of Econometrics, 2nd edition*, New York: John Wiley & Sons.
- Nelson, Daniel B. (1991). “Conditional Heteroskedasticity in Asset Returns: A New Approach,” *Econometrica*, 59, 347–370.
- Quandt, Richard E. (1988). *The Econometrics of Disequilibrium*, Oxford: Blackwell Publishing Co.
- Vuong, Q. H. (1989). “Likelihood Ratio Tests for Model Selection and Non-Nested Hypotheses,” *Econometrica*, 57(2), 307–333.

Part VI. Multiple Equation Analysis

In this section, we document EViews tools for multiple equation estimation, forecasting and data analysis.

- The first two chapters describe estimation techniques for systems of equations ([Chapter 33. “System Estimation,” on page 307](#)), and VARs and VECs ([Chapter 34. “Vector Autoregression and Error Correction Models,” on page 345](#)).
- [Chapter 35. “State Space Models and the Kalman Filter,” on page 383](#) describes the use of EViews’ state space and Kalman filter tools for modeling structural time series models.
- [Chapter 36. “Models,” beginning on page 407](#) describes the use of model objects to forecast from multiple equation estimates, or to perform multivariate simulation.

Chapter 33. System Estimation

This chapter describes methods of estimating the parameters of systems of equations. We describe least squares, weighted least squares, seemingly unrelated regression (SUR), weighted two-stage least squares, three-stage least squares, full-information maximum likelihood (FIML), generalized method of moments (GMM), and autoregressive conditional heteroskedasticity (ARCH) estimation techniques.

Once you have estimated the parameters of your system of equations, you may wish to forecast future values or perform simulations for different values of the explanatory variables. [Chapter 36. “Models,” on page 407](#) describes the use of models to forecast from an estimated system of equations or to perform single and multivariate simulation.

Background

A *system* is a group of equations containing unknown parameters. Systems can be estimated using a number of multivariate techniques that take into account the interdependencies among the equations in the system.

The general form of a system is:

$$f(y_t, x_t, \beta) = \epsilon_t, \quad (33.1)$$

where y_t is a vector of endogenous variables, x_t is a vector of exogenous variables, and ϵ_t is a vector of possibly serially correlated disturbances. The task of estimation is to find estimates of the vector of parameters β .

EViews provides you with a number of methods of estimating the parameters of the system. One approach is to estimate each equation in the system separately, using one of the single equation methods described earlier in this manual. A second approach is to estimate, simultaneously, the complete set of parameters of the equations in the system. The simultaneous approach allows you to place constraints on coefficients across equations and to employ techniques that account for correlation in the residuals across equations.

While there are important advantages to using a system to estimate your parameters, they do not come without cost. Most importantly, if you misspecify one of the equations in the system and estimate your parameters using single equation methods, only the misspecified equation will be poorly estimated. If you employ system estimation techniques, the poor estimates for the misspecification equation may “contaminate” estimates for other equations.

At this point, we take care to distinguish between systems of equations and models. A *model* is a group of known equations describing endogenous variables. Models are used to solve for values of the endogenous variables, given information on other variables in the model.

Systems and models often work together quite closely. You might estimate the parameters of a system of equations, and then create a model in order to forecast or simulate values of the endogenous variables in the system. We discuss this process in greater detail in [Chapter 36. “Models,” on page 407.](#)

System Estimation Methods

EViews will estimate the parameters of a system of equations using:

- Ordinary least squares.
- Equation weighted regression.
- Seemingly unrelated regression (SUR).
- System two-state least squares.
- Weighted two-stage least squares.
- Three-stage least squares.
- Full information maximum likelihood (FIML).
- Generalized method of moments (GMM).
- Autoregressive Conditional Heteroskedasticity (ARCH).

The equations in the system may be linear or nonlinear, and may contain autoregressive error terms.

In the remainder of this section, we describe each technique at a general level. Users who are interested in the technical details are referred to the [“Technical Discussion” on page 333.](#)

Ordinary Least Squares

This technique minimizes the sum-of-squared residuals for each equation, accounting for any cross-equation restrictions on the parameters of the system. If there are no such restrictions, this method is identical to estimating each equation using single-equation ordinary least squares.

Cross-Equation Weighting

This method accounts for cross-equation heteroskedasticity by minimizing the weighted sum-of-squared residuals. The equation weights are the inverses of the estimated equation variances, and are derived from unweighted estimation of the parameters of the system. This method yields identical results to unweighted single-equation least squares if there are no cross-equation restrictions.

Seemingly Unrelated Regression

The seemingly unrelated regression (SUR) method, also known as the multivariate regression, or Zellner's method, estimates the parameters of the system, accounting for heteroskedasticity and contemporaneous correlation in the errors across equations. The estimates of the cross-equation covariance matrix are based upon parameter estimates of the unweighted system.

Note that EViews estimates a more general form of SUR than is typically described in the literature, since it allows for cross-equation restrictions on parameters.

Two-Stage Least Squares

The system two-stage least squares (STLS) estimator is the system version of the single equation two-stage least squares estimator described above. STLS is an appropriate technique when some of the right-hand side variables are correlated with the error terms, and there is neither heteroskedasticity, nor contemporaneous correlation in the residuals. EViews estimates STLS by applying TSLS equation by equation to the unweighted system, enforcing any cross-equation parameter restrictions. If there are no cross-equation restrictions, the results will be identical to unweighted single-equation TSLS.

Weighted Two-Stage Least Squares

The weighted two-stage least squares (WTLS) estimator is the two-stage version of the weighted least squares estimator. WTLS is an appropriate technique when some of the right-hand side variables are correlated with the error terms, and there is heteroskedasticity, but no contemporaneous correlation in the residuals.

EViews first applies STLS to the unweighted system. The results from this estimation are used to form the equation weights, based upon the estimated equation variances. If there are no cross-equation restrictions, these first-stage results will be identical to unweighted single-equation TSLS.

Three-Stage Least Squares

Three-stage least squares (3SLS) is the two-stage least squares version of the SUR method. It is an appropriate technique when right-hand side variables are correlated with the error terms, and there is both heteroskedasticity, and contemporaneous correlation in the residuals.

EViews applies TSLS to the unweighted system, enforcing any cross-equation parameter restrictions. These estimates are used to form an estimate of the full cross-equation covariance matrix which, in turn, is used to transform the equations to eliminate the cross-equation correlation. TSLS is applied to the transformed model.

Full Information Maximum Likelihood (FIML)

Full Information Maximum Likelihood (FIML) estimates the likelihood function under the assumption that the contemporaneous errors have a joint normal distribution. Provided that the likelihood function is correctly specified, FIML is fully efficient.

Generalized Method of Moments (GMM)

The GMM estimator belongs to a class of estimators known as M-estimators that are defined by minimizing some criterion function. GMM is a robust estimator in that it does not require information of the exact distribution of the disturbances.

GMM estimation is based upon the assumption that the disturbances in the equations are uncorrelated with a set of instrumental variables. The GMM estimator selects parameter estimates so that the correlations between the instruments and disturbances are as close to zero as possible, as defined by a criterion function. By choosing the weighting matrix in the criterion function appropriately, GMM can be made robust to heteroskedasticity and/or autocorrelation of unknown form.

Many standard estimators, including all of the system estimators provided in EViews, can be set up as special cases of GMM. For example, the ordinary least squares estimator can be viewed as a GMM estimator, based upon the conditions that each of the right-hand side variables is uncorrelated with the residual.

Autoregressive Conditional Heteroskedasticity (ARCH)

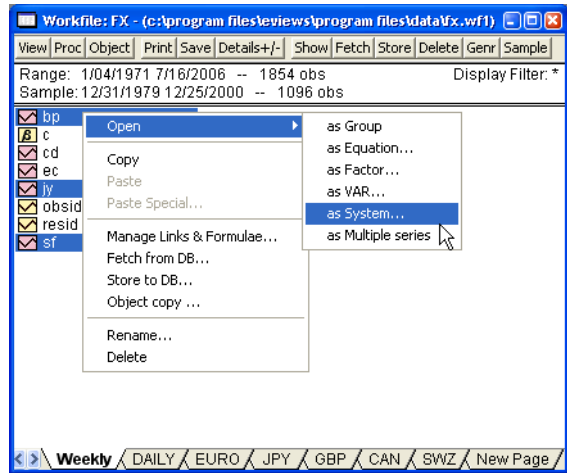
The System ARCH estimator is the multivariate version of ARCH estimator. System ARCH is an appropriate technique when one wants to model the variance and covariance of the error terms, generally in an autoregressive form. System ARCH allows you to choose from the most popular multivariate ARCH specifications: Conditional Constant Correlation, the Diagonal VECH, and (indirectly) the Diagonal BEKK.

How to Create and Specify a System

To estimate the parameters of your system of equations, you should first create a system object and specify the system of equations. Click on **Object/New Object.../System** or type `system` in the command window. The system object window should appear. When you first create the system, the window will be blank. You will fill the system specification window with text describing the equations, and potentially, lines describing the instruments and the parameter starting values.

From a list of variables, EViews can also automatically generate linear equations in a system. To use this procedure, first highlight the dependent variables that will be in the system. Next, double click on any of the highlighted series, and select **Open/Open System...**, or right click and select **Open/as System....** The **Make System** dialog box should appear with the variable names entered in the **Dependent variables** field. You can augment the specification by adding regressors or AR()

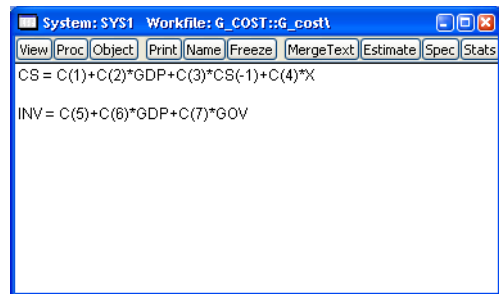
terms, either estimated with common or equation specific coefficients. See “[System Procs](#)” on [page 323](#) for additional details on this dialog.



Equations

Enter your equations, by formula, using standard EViews expressions. The equations in your system should be behavioral equations with unknown coefficients and an implicit error term.

Consider the specification of a simple two equation system. You can use the default EViews coefficients, C(1), C(2), and so on, or you can use other coefficient vectors, in which case you should first declare them by clicking **Object/New Object.../Matrix-Vector-Coef/Coefficient Vector** in the main menu.



There are some general rules for specifying your equations:

- Equations can be nonlinear in their variables, coefficients, or both. Cross equation coefficient restrictions may be imposed by using the same coefficients in different equations. For example:

$$y = c(1) + c(2) * x$$

$$z = c(3) + c(2) * z + (1 - c(2)) * x$$

- You may also impose adding up constraints. Suppose for the equation:

$$y = c(1) * x_1 + c(2) * x_2 + c(3) * x_3$$

you wish to impose $C(1) + C(2) + C(3) = 1$. You can impose this restriction by specifying the equation as:

$$y = c(1)*x1 + c(2)*x2 + (1-c(1)-c(2))*x3$$

- The equations in a system may contain autoregressive (AR) error specifications, but not MA, SAR, or SMA error specifications. You must associate coefficients with each AR specification. Enclose the entire AR specification in square brackets and follow each AR with an “=”-sign and a coefficient. For example:

$$cs = c(1) + c(2)*gdp + [ar(1)=c(3), ar(2)=c(4)]$$

You can constrain all of the equations in a system to have the same AR coefficient by giving all equations the same AR coefficient number, or you can estimate separate AR processes, by assigning each equation its own coefficient.

- Equations in a system need not have a dependent variable followed by an equal sign and then an expression. The “=”-sign can be anywhere in the formula, as in:

$$\log(unemp/(1-unemp)) = c(1) + c(2)*dmr$$

You can also write the equation as a simple expression without a dependent variable, as in:

$$(c(1)*x + c(2)*y + 4)^2$$

When encountering an expression that does not contain an equal sign, EViews sets the entire expression equal to the implicit error term.

If an equation should not have a disturbance, it is an identity, and should not be included in a system. If necessary, you should solve out for any identities to obtain the behavioral equations.

You should make certain that there is no identity linking all of the disturbances in your system. For example, if each of your equations describes a fraction of a total, the sum of the equations will always equal one, and the sum of the disturbances will identically equal zero. You will need to drop one of these equations to avoid numerical problems.

Instruments

If you plan to estimate your system using two-stage least squares, three-stage least squares, or GMM, you must specify the instrumental variables to be used in estimation. There are several ways to specify your instruments, with the appropriate form depending on whether you wish to have identical instruments in each equation, and whether you wish to compute the projections on an equation-by-equation basis, or whether you wish to compute a restricted projection using the stacked system.

In the simplest (default) case, EViews will form your instrumental variable projections on an equation-by-equation basis. If you prefer to think of this process as a two-step (2SLS) proce-

ture, the first-stage regression of the variables in your model on the instruments will be run separately for each equation.

In this setting, there are two ways to specify your instruments. If you would like to use identical instruments in every equation, you should include a line beginning with the keyword “@INST” or “INST”, followed by a list of all the exogenous variables to be used as instruments. For example, the line:

```
@inst gdp(-1 to -4) x gov
```

instructs EViews to use these six variables as instruments for all of the equations in the system. System estimation will involve a separate projection for each equation in your system.

You may also specify different instruments for each equation by appending an “@”-sign at the end of the equation, followed by a list of instruments for that equation. For example:

```
cs = c(1)+c(2)*gdp+c(3)*cs(-1) @ cs(-1) inv(-1) gov
inv = c(4)+c(5)*gdp+c(6)*gov @ gdp(-1) gov
```

The first equation uses CS(-1), INV(-1), GOV, and a constant as instruments, while the second equation uses GDP(-1), GOV, and a constant as instruments.

Lastly, you can mix the two methods. Any equation without individually specified instruments will use the instruments specified by the @inst statement. The system:

```
@inst gdp(-1 to -4) x gov
cs = c(1)+c(2)*gdp+c(3)*cs(-1)
inv = c(4)+c(5)*gdp+c(6)*gov @ gdp(-1) gov
```

will use the instruments GDP(-1), GDP(-2), GDP(-3), GDP(-4), X, GOV, and C, for the CS equation, but only GDP(-1), GOV, and C, for the INV equation.

As noted above, the EViews default behavior is to perform the instrumental variables projection on an equation-by-equation basis. You may, however, wish to perform the projections on the stacked system. Notably, where the number of instruments is large, relative to the number of observations, stacking the equations and instruments prior to performing the projection may be the only feasible way to compute 2SLS estimates.

To designate instruments for a stacked projection, you should use the @stackinst statement (note: this statement is only available for systems estimated by 2SLS or 3SLS; it is not available for systems estimated using GMM).

In a @stackinst statement, the “@STACKINST” keyword should be followed by a list of stacked instrument specifications. Each specification is a comma delimited list of series enclosed in parentheses (one per equation), describing the instruments to be constrained in a stacked specification.

For example, the following `@stackinst` specification creates two instruments in a three equation model:

```
@stackinst (z1,z2,z3) (m1,m1,m1)
```

This statement instructs EViews to form two stacked instruments, one by stacking the separate series Z1, Z2, and Z3, and the other formed by stacking M1 three times. The first-stage instrumental variables projection is then of the variables in the stacked system on the stacked instruments.

When working with systems that have a large number of equations, the above syntax may be unwieldy. For these cases, EViews provides a couple of shortcuts. First, for instruments that are identical in all equations, you may use an “*” after the comma to instruct EViews to repeat the specified series. Thus, the above statement is equivalent to:

```
@stackinst (z1,z2,z3) (m1,*)
```

Second, for non-identical instruments, you may specify a set of stacked instruments using an EViews group object, so long as the number of variables in the group is equal to the number of equations in the system. Thus, if you create a group Z with,

```
group z z1 z2 z3
```

the above statement can be simplified to:

```
@stackinst z (m1,*)
```

You can, of course, combine ordinary instrument and stacked instrument specifications. This situation is equivalent to having common and equation specific coefficients for variables in your system. Simply think of the stacked instruments as representing common (coefficient) instruments, and ordinary instruments as representing equation specific (coefficient) instruments. For example, consider the system given by,

```
@stackinst (z1,z2,z3) (m1,*)
@inst ia
y1 = c(1)*x1
y2 = c(1)*x2
y3 = c(1)*x3 @ ic
```

The stacked instruments for this specification may be represented as:

$$\begin{bmatrix} Z1 & M1 & IA & C & 0 & 0 & 0 & 0 & 0 \\ Z2 & M1 & 0 & 0 & IA & C & 0 & 0 & 0 \\ Z3 & M1 & 0 & 0 & 0 & 0 & IA & C & IC \end{bmatrix} \quad (33.2)$$

so it is easy to see that this specification is equivalent to the following stacked specification,

```
@stackinst (z1, z2, z3) (m1, *) (ia, 0, 0) (0, ia, 0) (0, 0, ia)
(0, 0, ic)
```


since the common instrument specification,

```
@inst ia
```

is equivalent to:

```
@stackinst (ia, 0, 0) (0, ia, 0) (0, 0, ia)
```

Note that the constant instruments are added implicitly.

Additional Comments

- If you include a “C” in the stacked instrument list, it will not be included in the individual equations. If you do not include the “C” as a stacked instrument, it will be included as an instrument in every equation, whether specified explicitly or not.
- You should list all exogenous right-hand side variables as instruments for a given equation.
- Identification requires that there should be at least as many instruments (including the constant) in each equation as there are right-hand side variables in that equation.
- The `@stackinst` statement is only available for estimation by 2SLS and 3SLS. It is not currently supported for GMM.
- If you estimate your system using a method that does not use instruments, all instrument specification lines will be ignored.

Starting Values

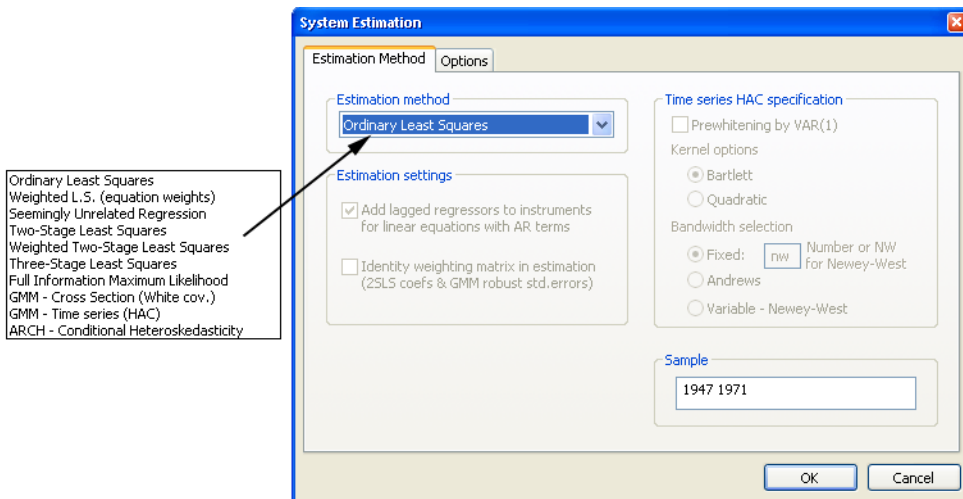
For systems that contain nonlinear equations, you can include a line that begins with `param` to provide starting values for some or all of the parameters. List pairs of parameters and values. For example:

```
param c(1) .15 b(3) .5
```

sets the initial values of C(1) and B(3). If you do not provide starting values, EViews uses the values in the current coefficient vector. In ARCH estimation, by default, EViews does provide a set of starting coefficients. Users are able to provide their own set of starting values by selecting **User Supplied** in the **Starting coefficient value** field located in the **Options** tab.

How to Estimate a System

Once you have created and specified your system, you may push the **Estimate** button on the toolbar to bring up the **System Estimation** dialog.



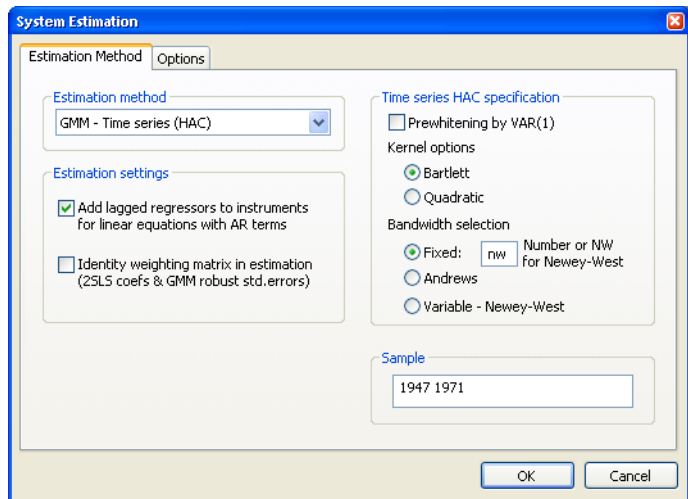
The drop-down menu marked **Estimation Method** provides you with several options for the estimation method. You may choose from one of a number of methods for estimating the parameters of your specification.

The estimation dialog may change to reflect your choice, providing you with additional options. If you select an estimator which uses instrumental variables, a checkbox will appear, prompting you to choose whether to **Add lagged regressors to instruments for linear equations with AR terms**. As the checkbox label suggests, if selected, EViews will add lagged values of the dependent and independent variable to the instrument list when estimating AR models. The lag order for these instruments will match the AR order of the specification. This automatic lag inclusion reflects the fact that EViews transforms the linear specification to a nonlinear specification when estimating AR models, and that the lagged values are ideal instruments for the transformed specification. If you wish to maintain precise control over the instruments added to your model, you should unselect this option.

Additional options appear if you are estimating a GMM specification. Note that the **GMM-Cross section** option uses a weighting matrix that is robust to heteroskedasticity and contemporaneous correlation of unknown form, while the **GMM-Time series (HAC)** option extends this robustness to autocorrelation of unknown form.

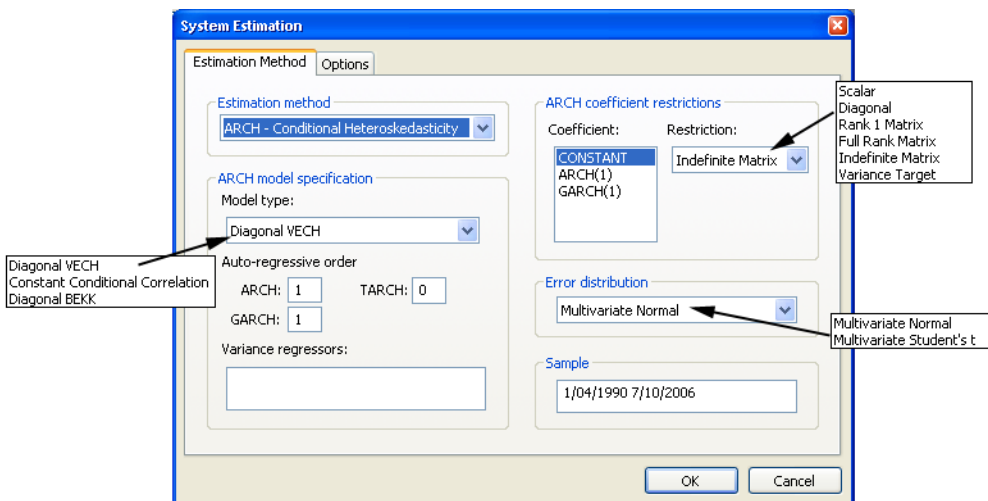
If you select either GMM method, EViews will display a checkbox labeled **Identity weighting matrix in estimation**. If selected, EViews will estimate the model using identity weights, and will use the estimated coefficients and GMM specification you provide to compute a coefficient covariance matrix that is robust to cross-section heteroskedasticity (White) or heteroskedasticity and autocorrelation (Newey-West). If this option is not selected, EViews will use the GMM weights both in estimation, and in computing the coefficient covariances.

When you select the **GMM-Time series (HAC)** option, the dialog displays additional options for specifying the weighting matrix. The new options will appear on the right side of the dialog. These options control the computation of the heteroskedasticity and autocorrelation robust (HAC) weighting matrix. See “[Technical Discussion](#)” on [page 333](#) for a more detailed discussion of these options.



The **Kernel Options** determines the functional form of the kernel used to weight the autocovariances to compute the weighting matrix. The **Bandwidth Selection** option determines how the weights given by the kernel change with the lags of the autocovariances in the computation of the weighting matrix. If you select **Fixed** bandwidth, you may enter a number for the bandwidth or type *nw* to use Newey and West’s fixed bandwidth selection criterion.

The **Prewhitening** option runs a preliminary VAR(1) prior to estimation to “soak up” the correlation in the moment conditions.



If the **ARCH - Conditional Heteroskedasticity** method is selected, the dialog displays the options appropriate for ARCH models. **Model type** allows you to select among three different multivariate ARCH models: **Diagonal VECH**, **Constant Conditional Correlation (CCC)**, and **Diagonal BEKK**. **Auto-regressive order** indicates the number of autoregressive terms included in the model. You may use the **Variance Regressors** edit field to specify any regressors in the variance equation.

The coefficient specifications for the auto-regressive terms and regressors in the variance equation may be fine-tuned using the controls in the **ARCH coefficient restrictions** section of the dialog page. Each auto-regression or regressor term is displayed in the **Coefficient** list. You should select a term to modify it, and in the **Restriction** field select a type coefficient specification for that term. For the Diagonal VECH model, each of the coefficient matrices may be restricted to be **Scalar**, **Diagonal**, **Rank One**, **Full Rank**, **Indefinite Matrix** or (in the case of the constant coefficient) **Variance Target**. The options for the BEKK model behave the same except that the ARCH, GARCH, and TARCH term is restricted to be **Diagonal**. For the CCC model, **Scalar** is the only option for ARCH, TARCH and GARCH terms, **Scalar** and **Variance Target** are allowed for the constant term. For exogenous variables you may choose between **Individual** and **Common**, indicating whether the parameters are restricted to be the same for all variance equations (common) or are unrestricted.

By default, the conditional distribution of the error terms is assumed to be **Multivariate Normal**. You have the option of instead using **Multivariate Student's t** by selecting it in the **Error distribution** dropdown list.

Options

For weighted least squares, SUR, weighted TSLS, 3SLS, GMM, and nonlinear systems of equations, there are additional issues involving the procedure for computing the GLS weighting matrix and the coefficient vector and for ARCH system, the coefficient vector used in estimation, as well as backcasting and robust standard error options.

To specify the method used in iteration, click on the **Options** tab.

The estimation option controls the method of iterating over coefficients, over the weighting matrices, or both:

- **Update weights once, then—Iterate coefs to convergence** is the default method.

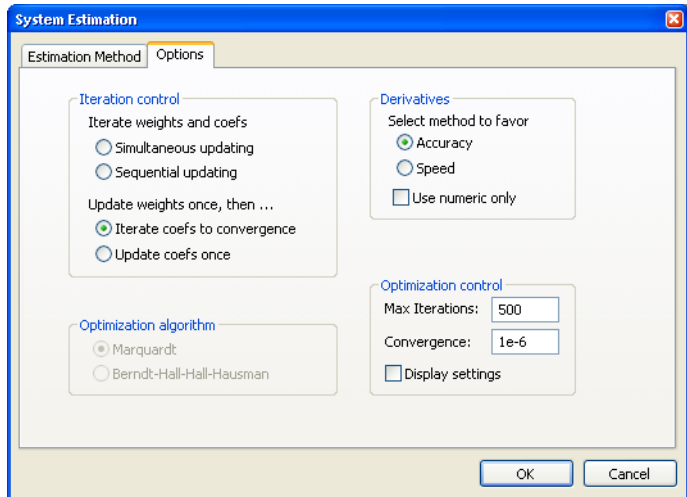
By default, EViews carries out a first-stage estimation of the coefficients using

no weighting matrix (the identity matrix). Using starting values obtained from OLS (or TSLS, if there are instruments), EViews iterates the first-stage estimates until the coefficients converge. If the specification is linear, this procedure involves a single OLS or TSLS regression.

The residuals from this first-stage iteration are used to form a consistent estimate of the weighting matrix.

In the second stage of the procedure, EViews uses the estimated weighting matrix in forming new estimates of the coefficients. If the model is nonlinear, EViews iterates the coefficient estimates until convergence.

- **Update weights once, then—Update coefs once** performs the first-stage estimation of the coefficients, and constructs an estimate of the weighting matrix. In the second stage, EViews does not iterate the coefficients to convergence, instead performing a single coefficient iteration step. Since the first stage coefficients are consistent, this one-step update is asymptotically efficient, but unless the specification is linear, does not produce results that are identical to the first method.
- **Iterate Weights and Coefs—Simultaneous updating** updates both the coefficients and the weighting matrix at each iteration. These steps are then repeated until both the coefficients and weighting matrix converge. This is the iteration method employed in EViews prior to version 4.
- **Iterate Weights and Coefs—Sequential updating** repeats the default method of updating weights and then iterating coefficients to convergence until both the coefficients and the weighting matrix converge.



Note that all four of the estimation techniques yield results that are asymptotically efficient. For linear models, the two **Iterate Weights and Coefs** options are equivalent, and the two **One-Step Weighting Matrix** options are equivalent, since obtaining coefficient estimates does not require iteration.

When ARCH is the estimation method a set of ARCH options appears:

- **Starting coefficient value** indicates what starting values EViews should use to start the iteration process. By default **EViews Supplied** is set. You can also select **User Supplied** which allows you to set your own starting coefficient via the C coefficient vector or another of your choice.
- **Coefficient name** specifies the name of the coefficient to be used in the variance equation. This can be different from the mean equation.
- **Starting (presample) covariance** indicates the method by which presample conditional variance and expected innovation should be calculated. Initial variance for the conditional variance are set using backcasting of the innovations,

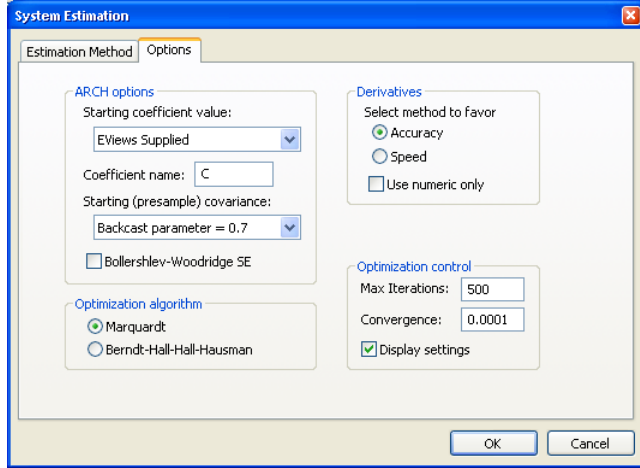
$$H_0 = \epsilon_0 \epsilon'_0 = \lambda^T \hat{H} + (1 - \lambda) \sum_{j=0}^T \lambda^{T-j-1} \epsilon_{T-j} \epsilon'_{T-j} \quad (33.3)$$

where:

$$\hat{H} = \sum_{t=1}^T (\epsilon_t \epsilon'_t) / T \quad (33.4)$$

is the unconditional variance of the residuals. By default, the smoothing parameter, λ is set to 0.7. However, you have the option to choose from a number of weights from 0.1 to 1, in increments of 0.1. Notice that if the parameter is set to 1 the initial value is simply the unconditional variance, *i.e.* backcasting is not performed.

- EViews will report the robust standard errors when the **Bollerslev-Wooldridge SE** box is checked.



For basic specifications, ARCH analytic derivatives are available, and are employed by default. For a more complex model, either in the means or conditional variance, numerical or a combination of numerical and analytics are used. Analytic derivatives are generally, but not always, faster than numeric.

In addition, the **Options** tab allows you to set a number of options for estimation, including convergence criterion, maximum number of iterations, and derivative calculation settings. See [“Setting Estimation Options” on page 625](#) for related discussion.

Estimation Output

The system estimation output contains parameter estimates, standard errors, and t -statistics (or z -statistics for maximum likelihood estimations), for each of the coefficients in the system. Additionally, EViews reports the determinant of the residual covariance matrix, and, for ARCH and FIML estimates, the maximized likelihood values, Akaike and Schwarz criteria. For ARCH estimations, the mean equation coefficients are separated from the variance coefficient section.

In addition, EViews reports a set of summary statistics for each equation. The R^2 statistic, Durbin-Watson statistic, standard error of the regression, sum-of-squared residuals, etc., are computed for each equation using the standard definitions, based on the residuals from the system estimation procedure.

In ARCH estimations, the raw coefficients of the variance equation do not necessarily give a clear understanding of the variance equations in many specifications. An extended coefficient view is supplied at the end of the output table to provide an enhanced view of the coefficient values involved.

You may access most of these results using regression statistics functions. See [Chapter 24, page 16](#) for a discussion of the use of these functions, and [Chapter 1. “Object View and Procedure Reference,” on page 3](#) of the *Command Reference* for a full listing of the available functions for systems.

Working With Systems

After obtaining estimates, the system object provides a number of tools for examining the equation results, and performing inference and specification testing.

System Views

- The **System Specification** view displays the specification window for the system. The specification window may also be displayed by pressing **Spec** on the toolbar.
- **Representations** provides you with the estimation command, the estimated equations and the substituted coefficient counterpart. For ARCH estimation this view also

includes additional variance and covariance specification in matrix formation as well as single equation with and without substituted coefficients.

- The **Estimation Output** view displays the coefficient estimates and summary statistics for the system. You may also access this view by pressing **Stats** on the system toolbar.
- **Residuals/Graphs** displays a separate graph of the residuals from each equation in the system.
- **Residuals/Correlation Matrix** computes the contemporaneous correlation matrix for the residuals of each equation.
- **Residuals/Covariance Matrix** computes the contemporaneous covariance matrix for the residuals. See also the function `@residcov` in [“System” on page 475](#) of the *Command Reference*.
- **Gradients and Derivatives** provides views which describe the gradients of the objective function and the information about the computation of any derivatives of the regression functions. Details on these views are provided in [Appendix E. “Gradients and Derivatives,” on page 637](#).
- **Conditional Covariance...** gives you the option to generate conditional covariances, variances, correlations or standard deviations for systems estimated using ARCH methods.
- **Coefficient Covariance Matrix** allows you to examine the estimated covariance matrix.
- **Coefficient Tests** allows you to display confidence ellipses or to perform hypothesis tests for restrictions on the coefficients. These views are discussed in greater depth in [“Confidence Ellipses” on page 142](#) and [“Wald Test \(Coefficient Restrictions\)” on page 145](#).
- A number of **Residual Tests** are supported, including **Correlograms**, **Portmanteau Autocorrelation Test**, and **Normality Test**. For most estimation methods, the Correlogram and Portmanteau views employ raw residuals, while Normality tests are based on standardized residuals. For ARCH estimation, the user has the added option of using a number of standardized residuals to calculate Correlogram and Portmanteau tests. The available standardization methods include Cholesky, Inverse Square Root of Residual Correlation, or Inverse Square Root of Residual Covariance. See [“Residual Tests” on page 349](#) for details on these tests and factorization methods.
- **Endogenous Table** presents a spreadsheet view of the endogenous variables in the system.
- **Endogenous Graph** displays graphs of each of the endogenous variables.

System Procs

One notable difference between systems and single equation objects is that there is no forecast procedure for systems. To forecast or perform simulation using an estimated system, you must use a model object.

EViews provides you with a simple method of incorporating the results of a system into a model. If you select **Proc/Make Model**, EViews will open an untitled model object containing the estimated system. This model can be used for forecasting and simulation. An alternative approach, creating the model and including the system object by name, is described in [“Building a Model” on page 424](#).

There are other procedures for working with the system:

- **Define System...** provides an easy way to define a system without having to type in every equation.

Dependent variables

allows you to list the dependent variables in the system. You have the option to transform these variables by selecting from the **Dependent variable transformation** list in the **Option** section. **Regressors and AR() terms** that share the same coefficient

across equations can be listed in **Common coefficients**, while those that do not can be placed in **Equation specific coefficients**. Command instruments can be listed in the **Common** field in the **Instrument list** section.

- **Estimate...** opens the dialog for estimating the system of equations. It may also be accessed by pressing **Estimate** on the system toolbar.
- **Make Residuals** creates a number of series containing the residuals for each equation in the system. The residuals will be given the next unused name of the form RESID01, RESID02, etc., in the order that the equations are specified in the system.
- **Make Endogenous Group** creates an untitled group object containing the endogenous variables.
- **Make Loglikelihoods** (for system ARCH) creates a series containing the log likelihood contribution.

- **Make Conditional Covariance** (for system ARCH) allows you to generate estimates of the conditional variances, covariances, or correlations for the specified set of dependent variables. (EViews automatically places all of the dependent variables in the **Variable** field. You have the option to modify this field to include only the variable of interest.)

If you select **Group** under **Format**, EViews will save the data in series. The **Base name** edit box indicates the base name to be used when generating series data. For the conditional variance series, the naming convention will be the specified base name plus terms of the form “_01”, “_02”. For covariances or correlations, the naming convention will use the base name plus “_01_02”, “_01_03”, *etc.*, where the additional text indicates the covariance/correlation between member 1 and 2, member 1 and 3, *etc.*

If **Matrix** is selected then whatever is in the **Matrix name** field will be generated for what is in the **Date** (or **Presample** if it is checked) edit field.

Example

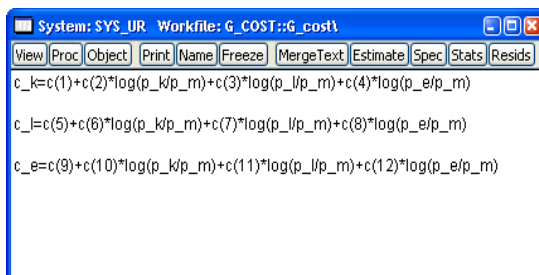
As an illustration of the process of estimating a system of equations in EViews, we estimate a translog cost function using data from Berndt and Wood (1975) as tabulated in Greene (1997). The translog cost function has four factors with three equations of the form:

$$\begin{aligned}
 c_K &= \beta_K + \delta_{KK} \log\left(\frac{p_K}{p_M}\right) + \delta_{KL} \log\left(\frac{p_L}{p_M}\right) + \delta_{KE} \log\left(\frac{p_E}{p_M}\right) + \epsilon_K \\
 c_L &= \beta_L + \delta_{LK} \log\left(\frac{p_K}{p_M}\right) + \delta_{LL} \log\left(\frac{p_L}{p_M}\right) + \delta_{LE} \log\left(\frac{p_E}{p_M}\right) + \epsilon_L \\
 c_E &= \beta_E + \delta_{EK} \log\left(\frac{p_K}{p_M}\right) + \delta_{EL} \log\left(\frac{p_L}{p_M}\right) + \delta_{EE} \log\left(\frac{p_E}{p_M}\right) + \epsilon_E
 \end{aligned} \tag{33.5}$$

where c_i and p_i are the cost share and price of factor i , respectively. β and δ are the parameters to be estimated. Note that there are cross equation coefficient restrictions that ensure symmetry of the cross partial derivatives.

We first estimate this system without imposing the cross equation restrictions and test whether the symmetry restrictions hold. Create a system by clicking **Object/New Object.../System** in the main toolbar or type `system` in the command window. Press the **Name** button and type in the name “SYS_UR” to name the system.

Next, type in the system window and specify the system as:



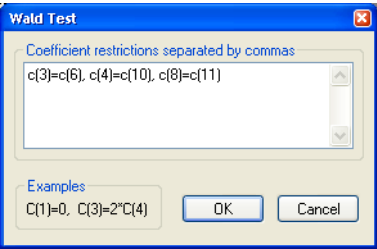
We estimate this model by full information maximum likelihood (FIML). FIML is invariant to the equation that is dropped. Press the **Estimate** button and choose **Full Information Maximum Likelihood**. EViews presents the estimated coefficients and regression statistics for each equation. The top portion of the output describes the coefficient estimates:

System: SYS_UR
 Estimation Method: Full Information Maximum Likelihood (Marquardt)
 Date: 08/15/06 Time: 16:19
 Sample: 1947 1971
 Included observations: 25
 Total system (balanced) observations 75
 Convergence achieved after 128 iterations

| | Coefficient | Std. Error | z-Statistic | Prob. |
|---------------------------------|-------------|------------|-------------|--------|
| C(1) | 0.054983 | 0.009353 | 5.878830 | 0.0000 |
| C(2) | 0.035130 | 0.035677 | 0.984676 | 0.3248 |
| C(3) | 0.004136 | 0.025616 | 0.161445 | 0.8717 |
| C(4) | 0.023633 | 0.084444 | 0.279867 | 0.7796 |
| C(5) | 0.250180 | 0.012019 | 20.81592 | 0.0000 |
| C(6) | 0.014758 | 0.024771 | 0.595766 | 0.5513 |
| C(7) | 0.083909 | 0.032188 | 2.606811 | 0.0091 |
| C(8) | 0.056411 | 0.096020 | 0.587493 | 0.5569 |
| C(9) | 0.043257 | 0.007981 | 5.420095 | 0.0000 |
| C(10) | -0.007707 | 0.012518 | -0.615722 | 0.5381 |
| C(11) | -0.002183 | 0.020123 | -0.108489 | 0.9136 |
| C(12) | 0.035624 | 0.061802 | 0.576422 | 0.5643 |
| <hr/> | | | | |
| Log Likelihood | 349.0326 | | | |
| Akaike info criterion | -26.96261 | | | |
| Schwarz criterion | -26.37755 | | | |
| Hannan-Quinn criterion | -26.80034 | | | |
| Determinant residual covariance | 1.50E-16 | | | |

while the bottom describes equation specific statistics.

To test the symmetry restrictions, select **View/Coefficient Tests/Wald Coefficient Tests...**, fill in the dialog:



and click **OK**. The result:

Wald Test:
System: SYS_UR

| Test Statistic | Value | df | Probability |
|----------------|----------|----|-------------|
| Chi-square | 0.418853 | 3 | 0.9363 |

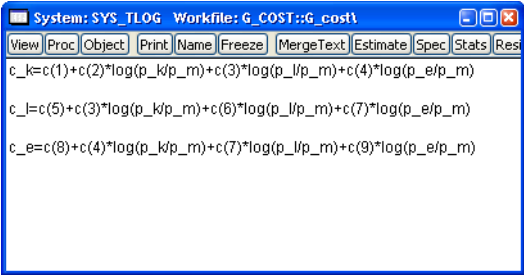
Null Hypothesis Summary:

| Normalized Restriction (= 0) | Value | Std. Err. |
|------------------------------|-----------|-----------|
| C(3) - C(6) | -0.010623 | 0.039839 |
| C(4) - C(10) | 0.031338 | 0.077778 |
| C(8) - C(11) | 0.058593 | 0.090751 |

Restrictions are linear in coefficients.

fails to reject the symmetry restrictions. To estimate the system imposing the symmetry restrictions, copy the object using **Object/Copy Object**, click **View/System Specification** and modify the system.

We have named the system SYS_TLOG. Note that to impose symmetry in the translog specification, we have restricted the coefficients on the cross-price terms to be the same (we have also renumbered the 9 remaining coefficients so that they are consecutive). The restrictions are imposed by using the same coefficients in each equation. For example, the coefficient on the



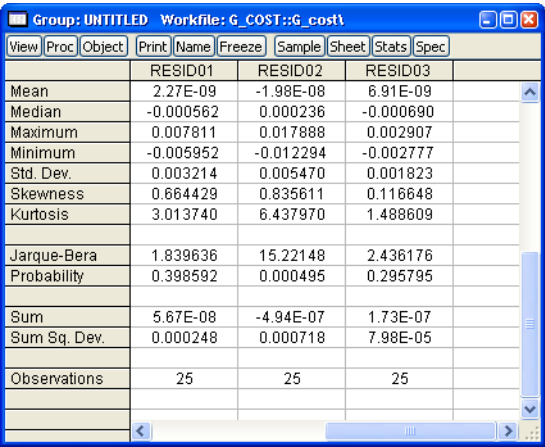
$\log(P_L/P_M)$ term in the C_K equation, $C(3)$, is the same as the coefficient on the $\log(P_K/P_M)$ term in the C_L equation.

To estimate this model using FIML, click **Estimate** and choose **Full Information Maximum Likelihood**. The top part of the equation describes the estimation specification, and provides coefficient and standard error estimates, t -statistics, p -values, and summary statistics:

| System: SYS_TLOG | | | | |
|--|-------------|------------|-------------|--------|
| Estimation Method: Full Information Maximum Likelihood (Marquardt) | | | | |
| Date: 08/15/06 Time: 16:31 | | | | |
| Sample: 1947 1971 | | | | |
| Included observations: 25 | | | | |
| Total system (balanced) observations 75 | | | | |
| Convergence achieved after 62 iterations | | | | |
| | Coefficient | Std. Error | z-Statistic | Prob. |
| C(1) | 0.057022 | 0.003306 | 17.24930 | 0.0000 |
| C(2) | 0.029742 | 0.012583 | 2.363708 | 0.0181 |
| C(3) | -0.000369 | 0.011205 | -0.032975 | 0.9737 |
| C(4) | -0.010228 | 0.006027 | -1.697186 | 0.0897 |
| C(5) | 0.253398 | 0.005050 | 50.17748 | 0.0000 |
| C(6) | 0.075427 | 0.015483 | 4.871651 | 0.0000 |
| C(7) | -0.004414 | 0.009141 | -0.482910 | 0.6292 |
| C(8) | 0.044286 | 0.003349 | 13.22352 | 0.0000 |
| C(9) | 0.018767 | 0.014894 | 1.260015 | 0.2077 |
| Log Likelihood | | 344.5916 | | |
| Akaike info criterion | | -26.84733 | | |
| Schwarz criterion | | -26.40853 | | |
| Hannan-Quinn criterion | | -26.72563 | | |
| Determinant residual covariance | | 2.14E-16 | | |

The log likelihood value reported at the bottom of the first part of the table may be used to construct likelihood ratio tests.

Since maximum likelihood assumes the errors are multivariate normal, we may wish to test whether the residuals are normally distributed. Click **Proc/Make Residuals** and EViews opens an untitled group window containing the residuals of each equation in the system. Then to compute descriptive statistics for each residual in the group, select **View/Descriptive Stats/Common Sample** from the group window toolbar.



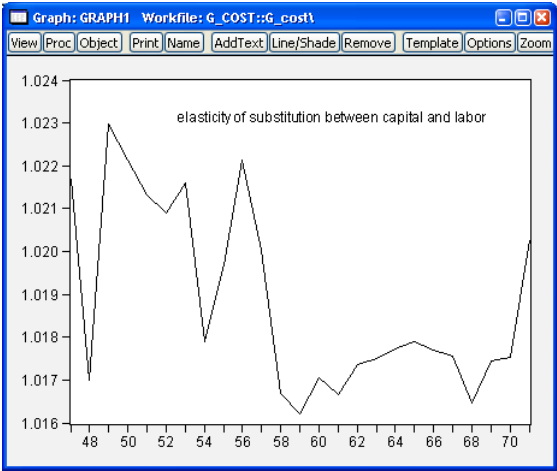
| | RESID01 | RESID02 | RESID03 |
|--------------|-----------|-----------|-----------|
| Mean | 2.27E-09 | -1.98E-08 | 6.91E-09 |
| Median | -0.000562 | 0.000236 | -0.000690 |
| Maximum | 0.007811 | 0.017888 | 0.002907 |
| Minimum | -0.005952 | -0.012294 | -0.002777 |
| Std. Dev. | 0.003214 | 0.005470 | 0.001823 |
| Skewness | 0.664429 | 0.835611 | 0.116648 |
| Kurtosis | 3.013740 | 6.437970 | 1.488609 |
| Jarque-Bera | 1.839636 | 15.22148 | 2.436176 |
| Probability | 0.398592 | 0.000495 | 0.295795 |
| Sum | 5.67E-08 | -4.94E-07 | 1.73E-07 |
| Sum Sq. Dev. | 0.000248 | 0.000718 | 7.98E-05 |
| Observations | 25 | 25 | 25 |

The Jarque-Bera statistic rejects the hypothesis of normal distribution for the second equation but not for the other equations.

The estimated coefficients of the translog cost function may be used to construct estimates of the elasticity of substitution between factors of production. For example, the elasticity of substitution between capital and labor is given by $1 + c(3)/(C_K * C_L)$. Note that the elasticity of substitution is not a constant, and depends on the values of C_K and C_L . To create a series containing the elasticities computed for each observation, select **Quick/Generate Series...**, and enter:

`es_kl = 1 + sys1.c(3)/(c_k*c_l)`

To plot the series of elasticity of substitution between capital and labor for each observation, double click on the series name ES_KL in the workfile and select **View/Line Graph**:



While it varies over the sample, the elasticity of substitution is generally close to one, which is consistent with the assumption of a Cobb-Douglas cost function.

System ARCH Example

In this section we provide an example for system arch estimation. We will model the weekly returns of Japanese Yen (jy_t), Swiss Franc (sf_t) and British Pound (bp_t). The data, which are located in the “Weekly” page of the workfile “FX.WF1”, which may be located in the Example File folder, contain the weekly exchange rates of these currencies against the U.S. dollar. The mean equations for the continuously compounding returns is regressed against a constant:

$$\begin{aligned}\log(jy_t/jy_{t-1}) &= c_1 + \epsilon_{1t} \\ \log(sf_t/sf_{t-1}) &= c_2 + \epsilon_{2t} \\ \log(bp_t/bp_{t-1}) &= c_3 + \epsilon_{3t}\end{aligned}\tag{33.6}$$

where $\epsilon_t = [\epsilon_{1t}, \epsilon_{2t}, \epsilon_{3t}]'$ is assumed to be distributed normally with mean zero and covariance H_t . The conditional covariance is modeled with a basic Diagonal VEC model:

$$H_t = \Omega + A \otimes \epsilon_{t-1} \epsilon_{t-1}' + B \otimes H_{t-1}\tag{33.7}$$

To estimate this model, create a system with the following specification:

```
dlog(jy) = c(1)
dlog(sf) = c(2)
dlog(bp) = c(3)
```

We estimate this model by selecting **ARCH - Conditional Heteroskedasticity** as the estimation method in the estimation dialog. Since the model we want to estimate is the default Diagonal VEC model we leave the rest of the options as they are. In the sample field, we change the sample to “1980 2000” to use only a portion of the data. Click on **OK** to estimate the system.

EViews displays the results of the estimation, which are similar to other system estimation output with a few differences. The ARCH results contain the coefficients statistics section (which includes both the mean and raw variance coefficients), model and equation specific statistics, and an extended section describing the variance coefficients.

The coefficient section at the top is separated into two parts, one contains the estimated coefficient for the mean equation and the other contains the estimated raw coefficients for the variance equation. The parameters estimates of the mean equation, C(1), C(2) and C(3), are listed in the upper portion of the coefficient list.

System: SYS01

Estimation Method: ARCH Maximum Likelihood (Marquardt)

Covariance Specification: Diagonal VEC

Date: 10/06/06 Time: 15:33

Sample: 12/31/1979 12/25/2000

Included observations: 1096

Total system (balanced) observations 3288

Presample covariance backcast parameter: 0.7

Convergence achieved after 129 iterations

| | Coefficient | Std. Error | z-Statistic | Prob. |
|--------------------------------|-------------|------------|-------------|--------|
| C(1) | -0.000865 | 0.000446 | -1.936740 | 0.0528 |
| C(2) | 5.43E-05 | 0.000454 | 0.119511 | 0.9049 |
| C(3) | -3.49E-05 | 0.000378 | -0.092283 | 0.9265 |
| Variance Equation Coefficients | | | | |
| C(4) | 6.49E-06 | 1.10E-06 | 5.919904 | 0.0000 |
| C(5) | 3.64E-06 | 9.67E-07 | 3.759946 | 0.0002 |
| C(6) | -2.64E-06 | 7.39E-07 | -3.575568 | 0.0003 |
| C(7) | 1.04E-05 | 2.28E-06 | 4.550942 | 0.0000 |
| C(8) | -8.03E-06 | 1.62E-06 | -4.972744 | 0.0000 |
| C(9) | 1.39E-05 | 2.49E-06 | 5.590125 | 0.0000 |
| C(10) | 0.059566 | 0.007893 | 7.546441 | 0.0000 |
| C(11) | 0.052100 | 0.007282 | 7.154665 | 0.0000 |
| C(12) | 0.046822 | 0.008259 | 5.669004 | 0.0000 |
| C(13) | 0.058630 | 0.007199 | 8.144180 | 0.0000 |
| C(14) | 0.067051 | 0.007508 | 8.931140 | 0.0000 |
| C(15) | 0.112734 | 0.008091 | 13.93396 | 0.0000 |
| C(16) | 0.917973 | 0.010867 | 84.47655 | 0.0000 |
| C(17) | 0.928844 | 0.009860 | 94.20362 | 0.0000 |
| C(18) | 0.924802 | 0.010562 | 87.55917 | 0.0000 |
| C(19) | 0.908492 | 0.011498 | 79.01313 | 0.0000 |
| C(20) | 0.886249 | 0.011892 | 74.52722 | 0.0000 |
| C(21) | 0.829154 | 0.012741 | 65.07760 | 0.0000 |

The variance coefficients are displayed in their own section. Coefficients C(4) to C(9) are the coefficients for the constant matrix, C(10) to C(15) are the coefficients for the ARCH term, and C(16) through C(21) are the coefficients for the GARCH term.

Note that the number of variance coefficients in an ARCH model can be very large. Even in this small 3-variable system, 18 parameters are estimated, making interpretation somewhat difficult; EViews provides a variance specification section that re-labels or transforms coefficients to aid in their interpretation.

Covariance Specification: Diagonal VEC

GARCH = M + A1.*RESID(-1)*RESID(-1)' + B1.*GARCH(-1)

M is an indefinite matrix

A1 is an indefinite matrix

B1 is an indefinite matrix

| Transformed Variance Coefficients | | | | |
|-----------------------------------|-------------|------------|-------------|--------|
| | Coefficient | Std. Error | z-Statistic | Prob. |
| M(1,1) | 6.49E-06 | 1.10E-06 | 5.919904 | 0.0000 |
| M(1,2) | 3.64E-06 | 9.67E-07 | 3.759946 | 0.0002 |
| M(1,3) | -2.64E-06 | 7.39E-07 | -3.575568 | 0.0003 |
| M(2,2) | 1.04E-05 | 2.28E-06 | 4.550942 | 0.0000 |
| M(2,3) | -8.03E-06 | 1.62E-06 | -4.972744 | 0.0000 |
| M(3,3) | 1.39E-05 | 2.49E-06 | 5.590125 | 0.0000 |
| A1(1,1) | 0.059566 | 0.007893 | 7.546441 | 0.0000 |
| A1(1,2) | 0.052100 | 0.007282 | 7.154665 | 0.0000 |
| A1(1,3) | 0.046822 | 0.008259 | 5.669004 | 0.0000 |
| A1(2,2) | 0.058630 | 0.007199 | 8.144180 | 0.0000 |
| A1(2,3) | 0.067051 | 0.007508 | 8.931140 | 0.0000 |
| A1(3,3) | 0.112734 | 0.008091 | 13.93396 | 0.0000 |
| B1(1,1) | 0.917973 | 0.010867 | 84.47655 | 0.0000 |
| B1(1,2) | 0.928844 | 0.009860 | 94.20362 | 0.0000 |
| B1(1,3) | 0.924802 | 0.010562 | 87.55917 | 0.0000 |
| B1(2,2) | 0.908492 | 0.011498 | 79.01313 | 0.0000 |
| B1(2,3) | 0.886249 | 0.011892 | 74.52722 | 0.0000 |
| B1(3,3) | 0.829154 | 0.012741 | 65.07760 | 0.0000 |

The first line of this section states the covariance model used in estimation, in this case Diagonal VEC. The next line of the header describes the model that we have estimated in abbreviated text form. In this case, “GARCH” is the conditional variance matrix, “M” is the constant matrix coefficient, A1 is the coefficient matrix for the ARCH term and B1 is the coefficient matrix for the GARCH term. M, A1, and B1 are all specified as indefinite matrices.

Next, the estimated values of the matrix elements as well as other statistics are displayed. Since the variance matrices are indefinite, the values are identical to those reported for the raw variance coefficients. For example, M(1,1), the (1,1) element in matrix M, corresponds to raw coefficient C(4), M(1,2) corresponds to C(5), A1(1,1) to C(10), etc.

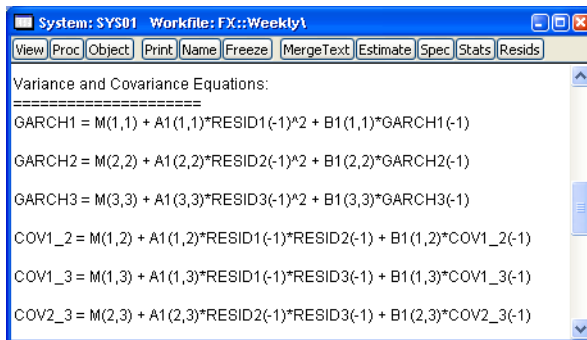
For matrix coefficients that are rank 1 or full rank, the values reported in this section are a transformation of the raw estimated coefficients, *i.e.* they are a function of one or more of the raw coefficients. Thus, the reported values do not have a one-to-one correspondence with the raw parameters.

A single equation representation of the variance-covariance equation may be viewed by clicking on **View/Representations** and examining the **Variance and Covariance Equations** section.

The GARCH equations are the conditional variance equations while the COV equations are the conditional covariance equations. For example GARCH1 is the conditional variance of Japanese yen. COV1_2 is the conditional covariance between the Japanese Yen and the Swiss Franc.

Before proceeding we name the system SYS01 by clicking on the **Name** button and accepting the default name.

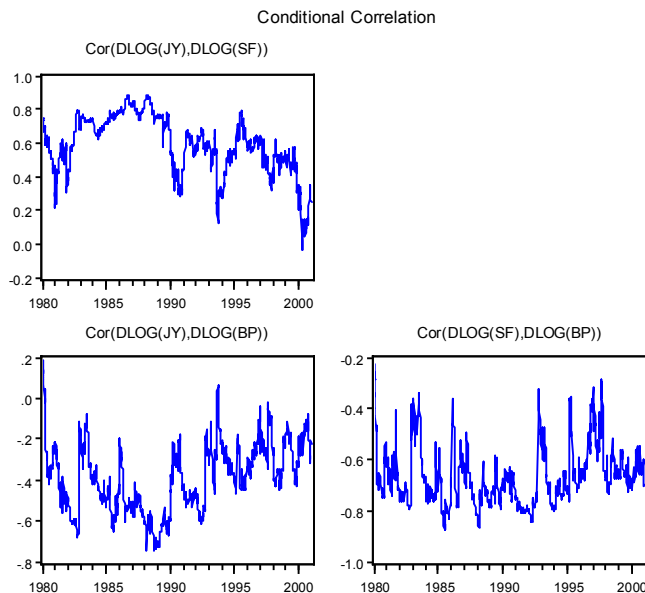
A graph of the conditional variance can be generated using **View/Conditional Covariance....** An extensive list of options is available, including **Covariance**, **Correlation**, **Variance**, and **Standard Deviation**. Data may also be displayed in graph, matrix or time series list format. Here is the correlation view:



```

System: SYS01  Workfile: F:\Weekly1
View|Proc|Object|Print|Name|Freeze|MergeText|Estimate|Spec|Stats|Resids
Variance and Covariance Equations:
=====
GARCH1 = M(1,1) + A1(1,1)*RESID1(-1)^2 + B1(1,1)*GARCH1(-1)
GARCH2 = M(2,2) + A1(2,2)*RESID2(-1)^2 + B1(2,2)*GARCH2(-1)
GARCH3 = M(3,3) + A1(3,3)*RESID3(-1)^2 + B1(3,3)*GARCH3(-1)
COV1_2 = M(1,2) + A1(1,2)*RESID1(-1)*RESID2(-1) + B1(1,2)*COV1_2(-1)
COV1_3 = M(1,3) + A1(1,3)*RESID1(-1)*RESID3(-1) + B1(1,3)*COV1_3(-1)
COV2_3 = M(2,3) + A1(2,3)*RESID2(-1)*RESID3(-1) + B1(2,3)*COV2_3(-1)

```



The correlation looks to be time varying, which is a general characteristic of this model.

Another possibility is to model the covariance matrix using the CCC specification, which imposes a constant correlation over time. We proceed by creating a new system just as above. This time we'll select **Constant Conditional Correlation** as the **Model type** for estimation.

The results show that this model has only 15 free parameters in the variance equation, as compared with 18 in the previous model. The extended variance section represents the variance equation as,

$$\text{GARCH}(i) = M(i) + A1(i) * \text{RESID}(i) (-1)^2 + B1(i) * \text{GARCH}(i) (-1)$$

while the model for the covariance equation is:

$$\text{COV}(i, j) = R(i, j) * \text{SQRT}(\text{GARCH}(i) * \text{GARCH}(j))$$

The output shows that the correlations, $R(1,2)$, $R(1,3)$, and $R(2,3)$ are 0.571322, -0.403232, and -0.677349, respectively.

Let's name this system SYS02.

Is this model better than the previous model? While the log likelihood value is lower, it also has fewer coefficients. We may compare the two system by looking at model selection criteria. The Akaike, Schwarz and Hannan-Quinn all show lower information criteria values for the VECH model than the CCC specification, suggesting that the time-varying Diagonal VECH specification may be preferred.

Technical Discussion

While the discussion to follow is expressed in terms of a balanced system of linear equations, the analysis carries forward in a straightforward way to unbalanced systems containing nonlinear equations.

Denote a system of m equations in stacked form as:

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_M \end{bmatrix} = \begin{bmatrix} X_1 & 0 & \dots & 0 \\ 0 & X_2 & & \vdots \\ & & \ddots & 0 \\ 0 & \dots & 0 & X_M \end{bmatrix} \begin{bmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_M \end{bmatrix} + \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_M \end{bmatrix} \quad (33.8)$$

where y_m is T vector, X_m is a $T \times k_m$ matrix, and β_m is a k_m vector of coefficients. The error terms ϵ have an $MT \times MT$ covariance matrix V . The system may be written in compact form as:

$$y = X\beta + \epsilon. \quad (33.9)$$

Under the standard assumptions, the residual variance matrix from this stacked system is given by:

$$V = E(\epsilon\epsilon') = \sigma^2(I_M \otimes I_T). \quad (33.10)$$

Other residual structures are of interest. First, the errors may be heteroskedastic across the m equations. Second, they may be heteroskedastic and contemporaneously correlated. We can characterize both of these cases by defining the $M \times M$ matrix of contemporaneous correlations, Σ , where the (i,j) -th element of Σ is given by $\sigma_{ij} = E(\epsilon_{it}\epsilon_{jt})$ for all t . If the errors are contemporaneously uncorrelated, then, $\sigma_{ij} = 0$ for $i \neq j$, and we can write:

$$V = \text{diag}(\sigma_1^2, \sigma_2^2, \dots, \sigma_M^2) \otimes I_T \quad (33.11)$$

More generally, if the errors are heteroskedastic and contemporaneously correlated:

$$V = \Sigma \otimes I_T. \quad (33.12)$$

Lastly, at the most general level, there may be heteroskedasticity, contemporaneous correlation, and autocorrelation of the residuals. The general variance matrix of the residuals may be written:

$$V = \begin{pmatrix} \sigma_{11}\Sigma_{11} & \sigma_{12}\Sigma_{12} & \cdots & \sigma_{1M}\Sigma_{1M} \\ \sigma_{21}\Sigma_{21} & \sigma_{22}\Sigma_{22} & & \vdots \\ & & \ddots & \\ \sigma_{M1}\Sigma_{M1} & \cdots & & \sigma_{MM}\Sigma_{MM} \end{pmatrix} \quad (33.13)$$

where Σ_{ij} is an autocorrelation matrix for the i -th and j -th equations.

Ordinary Least Squares

The OLS estimator of the estimated variance matrix of the parameters is valid under the assumption that $V = \Sigma \otimes I_T$. The estimator for β is given by,

$$b_{LS} = (X'X)^{-1}X'y \quad (33.14)$$

and the variance estimator is given by:

$$\text{var}(b_{LS}) = s^2(X'X)^{-1} \quad (33.15)$$

where s^2 is the residual variance estimate for the stacked system.

Weighted Least Squares

The weighted least squares estimator is given by:

$$b_{WLS} = (X'\hat{V}^{-1}X)^{-1}X'\hat{V}^{-1}y \quad (33.16)$$

where $\hat{V} = \text{diag}(s_{11}, s_{22}, \dots, s_{MM}) \otimes I_T$ is a consistent estimator of V , and s_{ii} is the residual variance estimator:

$$s_{ij} = ((y_i - X_i b_{LS})'(y_j - X_j b_{LS}))/\max(T_i, T_j) \quad (33.17)$$

where the inner product is taken over the non-missing common elements of i and j . The max function in Equation (33.17) is designed to handle the case of unbalanced data by down-weighting the covariance terms. Provided the missing values are asymptotically negligible, this yields a consistent estimator of the variance elements. Note also that there is no adjustment for degrees of freedom.

When specifying your estimation specification, you are given a choice of which coefficients to use in computing the s_{ij} . If you choose not to iterate the weights, the OLS coefficient estimates will be used to estimate the variances. If you choose to iterate the weights, the current parameter estimates (which may be based on the previously computed weights) are used in computing the s_{ij} . This latter procedure may be iterated until the weights and coefficients converge.

The estimator for the coefficient variance matrix is:

$$\text{var}(b_{WLS}) = (X' \hat{V}^{-1} X)^{-1}. \quad (33.18)$$

The weighted least squares estimator is efficient, and the variance estimator consistent, under the assumption that there is heteroskedasticity, but no serial or contemporaneous correlation in the residuals.

It is worth pointing out that if there are no cross-equation restrictions on the parameters of the model, weighted LS on the entire system yields estimates that are identical to those obtained by equation-by-equation LS. Consider the following simple model:

$$\begin{aligned} y_1 &= X_1 \beta_1 + \epsilon_1 \\ y_2 &= X_2 \beta_2 + \epsilon_2 \end{aligned} \quad (33.19)$$

If β_1 and β_2 are unrestricted, the WLS estimator given in Equation (33.18) yields:

$$b_{WLS} = \begin{bmatrix} ((X_1' X_1)/s_{11})^{-1} ((X_1' y_1)/s_{11}) \\ ((X_2' X_2)/s_{22})^{-1} ((X_2' y_2)/s_{22}) \end{bmatrix} = \begin{bmatrix} (X_1' X_1)^{-1} X_1' y_1 \\ (X_2' X_2)^{-1} X_2' y_2 \end{bmatrix}. \quad (33.20)$$

The expression on the right is equivalent to equation-by-equation OLS. Note, however, that even without cross-equation restrictions, the standard errors are not the same in the two cases.

Seemingly Unrelated Regression (SUR)

SUR is appropriate when all the right-hand side regressors X are assumed to be exogenous, and the errors are heteroskedastic and contemporaneously correlated so that the error variance matrix is given by $V = \Sigma \otimes I_T$. Zellner's SUR estimator of β takes the form:

$$b_{SUR} = (X'(\hat{\Sigma} \otimes I_T)^{-1} X)^{-1} X'(\hat{\Sigma} \otimes I_T)^{-1} y, \quad (33.21)$$

where $\hat{\Sigma}$ is a consistent estimate of Σ with typical element s_{ij} , for all i and j .

If you include AR terms in equation j , EViews transforms the model (see “[Estimating AR Models](#)” on page 67) and estimates the following equation:

$$y_{jt} = X_{jt}\beta_j + \left(\sum_{r=1}^{p_j} \rho_{jr}(y_{j(t-r)} - X_{j(t-r)}) \right) + \epsilon_{jt} \quad (33.22)$$

where ϵ_j is assumed to be serially independent, but possibly correlated contemporaneously across equations. At the beginning of the first iteration, we estimate the equation by nonlinear LS and use the estimates to compute the residuals $\hat{\epsilon}$. We then construct an estimate of Σ using $s_{ij} = (\hat{\epsilon}_i' \hat{\epsilon}_j) / \max(T_i, T_j)$ and perform nonlinear GLS to complete one iteration of the estimation procedure. These iterations may be repeated until the coefficients and weights converge.

Two-Stage Least Squares (TSLS) and Weighted TSLS

TSLS is a single equation estimation method that is appropriate when some of the variables in X are endogenous. Write the j -th equation of the system as,

$$Y\Gamma_j + XB_j + \epsilon_j = 0 \quad (33.23)$$

or, alternatively:

$$y_j = Y_j\gamma_j + X_j\beta_j + \epsilon_j = Z_j\delta_j + \epsilon_j \quad (33.24)$$

where $\Gamma_j' = (-1, \gamma_j', 0)$, $B_j' = (\beta_j', 0)$, $Z_j' = (Y_j', X_j')$ and $\delta_j' = (\gamma_j', \beta_j')$. Y is the matrix of endogenous variables and X is the matrix of exogenous variables.

In the first stage, we regress the right-hand side endogenous variables y_j on all exogenous variables X and get the fitted values:

$$\hat{Y}_j = X(X'X)^{-1}X'Y_j. \quad (33.25)$$

In the second stage, we regress y_j on \hat{Y}_j and X_j to get:

$$\hat{\delta}_{2SLs} = (\hat{Z}_j' \hat{Z}_j)^{-1} \hat{Z}_j' y_j. \quad (33.26)$$

where $\hat{Z}_j = (\hat{Y}_j, X_j)$.

Weighted TSLS applies the weights in the second stage so that:

$$\hat{\delta}_{W2SLs} = (\hat{Z}_j' \hat{V}^{-1} \hat{Z}_j)^{-1} \hat{Z}_j' \hat{V}^{-1} y_j \quad (33.27)$$

where the elements of the variance matrix are estimated in the usual fashion using the residuals from unweighted TSLS.

If you choose to iterate the weights, X is estimated at each step using the current values of the coefficients and residuals.

Three-Stage Least Squares (3SLS)

Since TSLS is a single equation estimator that does not take account of the covariances between residuals, it is not, in general, fully efficient. 3SLS is a system method that estimates all of the coefficients of the model, then forms weights and reestimates the model using the estimated weighting matrix. It should be viewed as the endogenous variable analogue to the SUR estimator described above.

The first two stages of 3SLS are the same as in TSLS. In the third stage, we apply feasible generalized least squares (FGLS) to the equations in the system in a manner analogous to the SUR estimator.

SUR uses the OLS residuals to obtain a consistent estimate of the cross-equation covariance matrix Σ . This covariance estimator is not, however, consistent if any of the right-hand side variables are endogenous. 3SLS uses the 2SLS residuals to obtain a consistent estimate of Σ .

In the balanced case, we may write the equation as,

$$\hat{\delta}_{3SLS} = (Z(\hat{\Sigma}^{-1} \otimes X(X'X)^{-1}X')Z)^{-1}Z(\hat{\Sigma}^{-1} \otimes X(X'X)^{-1}X')y \quad (33.28)$$

where $\hat{\Sigma}$ has typical element:

$$s_{ij} = ((y_i - Z_i\hat{\gamma}_{2SLS})'(y_j - Z_j\hat{\gamma}_{2SLS}))/\max(T_i, T_j) \quad (33.29)$$

If you choose to iterate the weights, the current coefficients and residuals will be used to estimate $\hat{\Sigma}$.

Generalized Method of Moments (GMM)

The basic idea underlying GMM is simple and intuitive. We have a set of theoretical moment conditions that the parameters of interest θ should satisfy. We denote these moment conditions as:

$$E(m(y, \theta)) = 0. \quad (33.30)$$

The method of moments estimator is defined by replacing the moment condition (33.30) by its sample analog:

$$\left(\sum_t m(y_t, \theta) \right) / T = 0. \quad (33.31)$$

However, condition (33.31) will not be satisfied for any θ when there are more restrictions m than there are parameters θ . To allow for such overidentification, the GMM estimator is defined by minimizing the following criterion function:

$$\sum_t m(y_t, \theta)' A(y_t, \theta) m(y_t, \theta) \quad (33.32)$$

which measures the “distance” between m and zero. A is a weighting matrix that weights each moment condition. Any symmetric positive definite matrix A will yield a consistent estimate of θ . However, it can be shown that a necessary (but not sufficient) condition to obtain an (asymptotically) efficient estimate of θ is to set A equal to the inverse of the covariance matrix Ω of the sample moments m . This follows intuitively, since we want to put less weight on the conditions that are more imprecise.

To obtain GMM estimates in EViews, you must be able to write the moment conditions in Equation (33.30) as an orthogonality condition between the residuals of a regression equation, $u(y, \theta, X)$, and a set of instrumental variables, Z , so that:

$$m(\theta, y, X, Z) = Z' u(\theta, y, X) \quad (33.33)$$

For example, the OLS estimator is obtained as a GMM estimator with the orthogonality conditions:

$$X'(y - X\beta) = 0. \quad (33.34)$$

For the GMM estimator to be identified, there must be at least as many instrumental variables Z as there are parameters θ . See the section on “[Generalized Method of Moments \(GMM\)](#),” beginning on page 51 for additional examples of GMM orthogonality conditions.

An important aspect of specifying a GMM problem is the choice of the weighting matrix A . EViews uses the optimal $A = \Omega^{-1}$, where Ω is the estimated covariance matrix of the sample moments m . EViews uses the consistent TSLS estimates for the initial estimate of θ in forming the estimate of Ω .

White’s Heteroskedasticity Consistent Covariance Matrix

If you choose the **GMM-Cross section** option, EViews estimates Ω using White’s heteroskedasticity consistent covariance matrix:

$$\hat{\Omega}_W = \hat{\Gamma}(0) = \frac{1}{T-k} \left(\sum_{t=1}^T Z_t' u_t u_t' Z_t \right) \quad (33.35)$$

where u is the vector of residuals, and Z_t is a $k \times p$ matrix such that the p moment conditions at t may be written as $m(\theta, y_t, X_t, Z_t) = Z_t' u(\theta, y_t, X_t)$.

Heteroskedasticity and Autocorrelation Consistent (HAC) Covariance Matrix

If you choose the **GMM-Time series** option, EViews estimates Ω by,

$$\hat{\Omega}_{HAC} = \hat{\Gamma}(0) + \left(\sum_{j=1}^{T-1} k(j, q) (\hat{\Gamma}(j) + \hat{\Gamma}'(j)) \right) \quad (33.36)$$

where:

$$\hat{\Gamma}(j) = \frac{1}{T-k} \left(\sum_{t=j+1}^T Z_{t-j}' u_t u_{t-j}' Z_t \right). \quad (33.37)$$

You also need to specify the *kernel* κ and the *bandwidth* q .

Kernel Options

The kernel κ is used to weight the covariances so that $\hat{\Omega}$ is ensured to be positive semi-definite. EViews provides two choices for the kernel, Bartlett and quadratic spectral (QS). The Bartlett kernel is given by:

$$\kappa(j, q) = \begin{cases} 1 - (j/q) & 0 \leq j \leq q \\ 0 & \text{otherwise} \end{cases} \quad (33.38)$$

while the quadratic spectral (QS) kernel is given by:

$$k(j/q) = \frac{25}{12(\pi x)^2} \left(\frac{\sin(6\pi x/5)}{6\pi x/5} - \cos(6\pi x/5) \right) \quad (33.39)$$

where $x = j/q$. The QS has a faster rate of convergence than the Bartlett and is smooth and not truncated (Andrews 1991). Note that even though the QS kernel is not truncated, it still depends on the bandwidth q (which need not be an integer).

Bandwidth Selection

The bandwidth q determines how the weights given by the kernel change with the lags in the estimation of Ω . Newey-West fixed bandwidth is based solely on the number of observations in the sample and is given by:

$$q = \text{int}(4(T/100)^{2/9}) \quad (33.40)$$

where $\text{int}(\cdot)$ denotes the integer part of the argument.

EViews also provides two “automatic”, or data dependent bandwidth selection methods that are based on the autocorrelations in the data. Both methods select the bandwidth according to:

$$q = \begin{cases} \text{int}(1.1447(\hat{\alpha}(1)T)^{1/3}) & \text{for the Bartlett kernel} \\ 1.3221(\hat{\alpha}(2)T)^{1/5} & \text{for the QS kernel} \end{cases} \quad (33.41)$$

The two methods, Andrews and Variable-Newey-West, differ in how they estimate $\hat{\alpha}(1)$ and $\hat{\alpha}(2)$.

Andrews (1991) is a parametric method that assumes the sample moments follow an AR(1) process. We first fit an AR(1) to each sample moment (33.33) and estimate the autocorrelation coefficients $\hat{\rho}_i$ and the residual variances $\hat{\sigma}_i^2$ for $i = 1, 2, \dots, zn$, where z is the

number of instrumental variables and n is the number of equations in the system. Then $\hat{\alpha}(1)$ and $\hat{\alpha}(2)$ are estimated by:

$$\begin{aligned}\hat{\alpha}(1) &= \left(\sum_{i=1}^{zn} \frac{4\hat{\rho}_i^2 \hat{\sigma}_i^4}{(1-\hat{\rho}_i)^6 (1+\hat{\rho}_i)^2} \right) / \left(\sum_{i=1}^{zn} \frac{\hat{\sigma}_i^4}{(1-\hat{\rho}_i)^4} \right) \\ \hat{\alpha}(2) &= \left(\sum_{i=1}^{zn} \frac{4\hat{\rho}_i^2 \hat{\sigma}_i^4}{(1-\hat{\rho}_i)^8} \right) / \left(\sum_{i=1}^{zn} \frac{\hat{\sigma}_i^4}{(1-\hat{\rho}_i)^4} \right)\end{aligned}\quad (33.42)$$

Note that we weight all moments equally, including the moment corresponding to the constant.

Newey-West (1994) is a nonparametric method based on a truncated weighted sum of the estimated cross-moments $\hat{\Gamma}(j)$. $\hat{\alpha}(1)$ and $\hat{\alpha}(2)$ are estimated by,

$$\hat{\alpha}(p) = \left(\frac{l' F(p) l}{l' F(0) l} \right) \quad (33.43)$$

where l is a vector of ones and:

$$F(p) = \Gamma(0) + \sum_{i=1}^L i^p (\hat{\Gamma}(i) + \hat{\Gamma}'(i)), \quad (33.44)$$

for $p = 1, 2$.

One practical problem with the Newey-West method is that we have to choose a lag selection parameter L . The choice of L is arbitrary, subject to the condition that it grow at a certain rate. EViews sets the lag parameter to:

$$L = \begin{cases} \text{int}(4(T/100)^{2/9}) & \text{for the Bartlett kernel} \\ T & \text{for the QS kernel} \end{cases} \quad (33.45)$$

Prewhitening

You can also choose to prewhiten the sample moments m to “soak up” the correlations in m prior to GMM estimation. We first fit a VAR(1) to the sample moments:

$$m_t = A m_{t-1} + v_t. \quad (33.46)$$

Then the variance $\hat{\Omega}$ of m is estimated by $\hat{\Omega} = (I - A)^{-1} \hat{\Omega}^* (I - A)^{-1}$ where $\hat{\Omega}^*$ is the variance of the residuals v_t and is computed using any of the above methods. The GMM estimator is then found by minimizing the criterion function:

$$u' Z \hat{\Omega}^{-1} Z' u \quad (33.47)$$

Note that while Andrews and Monahan (1992) adjust the VAR estimates to avoid singularity when the moments are near unit root processes, EViews does not perform this eigenvalue adjustment.

Multivariate ARCH

ARCH estimation uses maximum likelihood to jointly estimate the parameters of the mean and the variance equations.

Assuming multivariate normality, the log likelihood contributions for GARCH models are given by:

$$l_t = -\frac{1}{2} m \log(2\pi) - \frac{1}{2} \log(|H_t|) - \frac{1}{2} \epsilon_t' H_t^{-1} \epsilon_t \quad (33.48)$$

where m is the number of mean equations, and ϵ_t is the m vector of mean equation residuals. For Student's t -distribution, the contributions are of the form:

$$l_t = \log \left\{ \frac{\Gamma\left(\frac{v+m}{2}\right) v^{m/2}}{(v\pi)^{m/2} \Gamma\left(\frac{v}{2}\right) (v-2)^{m/2}} \right\} - \frac{1}{2} \log(|H_t|) - \frac{1}{2} (v+m) \log \left[1 + \frac{\epsilon_t' H_t^{-1} \epsilon_t}{v-2} \right] \quad (33.49)$$

where v is the estimated degree of freedom.

Given a specification for the mean equation and a distributional assumption, all that we require is a specification for the conditional covariance matrix. We consider, in turn, each of the three basic specifications: Diagonal VEC, Conditional Constant Correlation (CCC), and Diagonal BEKK.

Diagonal VEC

Bollerslev, *et. al* (1988) introduce a restricted version of the general multivariate VEC model of the conditional covariance with the following formulation:

$$H_t = \Omega + A \bullet \epsilon_{t-1} \epsilon_{t-1}' + B \bullet H_{t-1} \quad (33.50)$$

where the coefficient matrices A , B , and Ω are $N \times N$ symmetric matrices, and the operator “ \bullet ” is the element by element (Hadamard) product. The coefficient matrices may be parametrized in several ways. The most general way is to allow the parameters in the matrices to vary without any restrictions, *i.e.* parameterize them as indefinite matrices. In that case the model may be written in single equation format as:

$$(H_t)_{ij} = (\Omega)_{ij} + (A)_{ij} \epsilon_{jt-1} \epsilon_{it-1} + (B)_{ij} (H_{t-1})_{ij} \quad (33.51)$$

where, for instance, $(H_t)_{ij}$ is the i -th row and j -th column of matrix H_t .

Each matrix contains $N(N+1)/2$ parameters. This model is the most unrestricted version of a Diagonal VECH model. At the same time, it does not ensure that the conditional covariance matrix is positive semidefinite (PSD). As summarized in Ding and Engle (2001), there are several approaches for specifying coefficient matrices that restrict H to be PSD, possibly by reducing the number of parameters. One example is:

$$H_t = \tilde{\Omega}\Omega' + \tilde{A}A' \bullet \epsilon_{t-1}\epsilon_{t-1}' + \tilde{B}B' \otimes H_{t-1} \quad (33.52)$$

where raw matrices \tilde{A} , \tilde{B} , and $\tilde{\Omega}$ are any matrix up to rank N . For example, one may use the rank N Cholesky factorized matrix of the coefficient matrix. This method is labeled the **Full Rank Matrix** in the coefficient **Restriction** selection of the system ARCH dialog. While this method contains the same number of parameters as the indefinite version, it does ensure that the conditional covariance is PSD.

A second method, which we term **Rank One**, reduces the number of parameter estimated to N and guarantees that the conditional covariance is PSD. In this case, the estimated raw matrix is restricted, with all but the first column of coefficients equal to zero.

In both of these specifications, the reported raw variance coefficients are elements of \tilde{A} , \tilde{B} , and $\tilde{\Omega}$. These coefficients must be transformed to obtain the matrix of interest: $A = \tilde{A}A'$, $B = \tilde{B}B'$, and $\Omega = \tilde{\Omega}\Omega'$. These transformed coefficients are reported in the extended variance coefficient section at the end of the system estimation results.

There are two other covariance specifications that you may employ. First, the values in the $N \times N$ matrix may be a constant, so that:

$$B = b \cdot ii' \quad (33.53)$$

where b is a scalar and i is an $N \times 1$ vector of ones. This **Scalar** specification implies that for a particular term, the parameters of the variance and covariance equations are restricted to be the same. Alternately, the matrix coefficients may be parameterized as **Diagonal** so that all off diagonal elements are restricted to be zero. In both of these parameterizations, the coefficients are not restricted to be positive, so that H is not guaranteed to be PSD.

Lastly, for the constant matrix Ω , we may also impose a **Variance Target** on the coefficients which restricts the values of the coefficient matrix so that:

$$\Omega = \Omega_0 \bullet (ii' - A - B) \quad (33.54)$$

where Ω_0 is the unconditional sample variance of the residuals. When using this option, the constant matrix is not estimated, reducing the number of estimated parameters.

You may specify a different type of coefficient matrix for each term. For example, if one estimates a multivariate GARCH(1,1) model with indefinite matrix coefficient for the constant while specifying the coefficients of the ARCH and GARCH term to be rank one matrices, then the number of parameters will be $N((N+1)/2) + 2N$, instead of $3N((N+1)/2)$.

Conditional Constant Correlation (CCC)

Bollerslev (1990) specifies the elements of the conditional covariance matrix as follows:

$$\begin{aligned} h_{iit} &= c_i + a_i \epsilon_{it-1}^2 + d_i I_{it-1}^- \epsilon_{it-1}^2 + b_i h_{iit-1} \\ h_{ijt} &= \rho_{ij} \sqrt{h_{iit} h_{jjt}} \end{aligned} \quad (33.55)$$

Restrictions may be imposed on the constant term using variance targeting so that:

$$c_i = \sigma_0^2 (1 - a_i - b_i) \quad (33.56)$$

where σ_0^2 is the unconditional variance.

When exogenous variables are included in the variance specification, the user may choose between *individual* coefficients and *common* coefficients. For common coefficients, exogenous variables are assumed to have the same slope, g , for every equation. Individual coefficients allow each exogenous variable effect e_i to differ across equations.

$$h_{iit} = c_i + a_i \epsilon_{it-1}^2 + d_i I_{it-1}^- \epsilon_{it-1}^2 + b_i h_{iit-1} + e_i x_{1t} + g x_{2t} \quad (33.57)$$

Diagonal BEKK

BEKK (Engle and Kroner, 1995) is defined as:

$$H_t = \Omega \Omega' + A \epsilon_{t-1} \epsilon_{t-1}' A' + B H_{t-1} B' \quad (33.58)$$

EViews does not estimate the general form of BEKK in which A and B are unrestricted. However, a common and popular form, diagonal BEKK, may be specified that restricts A and B to be diagonals. This Diagonal BEKK model is identical to the Diagonal VEC model where the coefficient matrices are rank one matrices. For convenience, EViews provides an option to estimate the Diagonal VEC model, but display the result in Diagonal BEKK form.

References

- Andrews, Donald W. K. (1991). "Heteroskedasticity and Autocorrelation Consistent Covariance Matrix Estimation," *Econometrica*, 59, 817–858.
- Andrews, Donald W. K. and J. Christopher Monahan (1992). "An Improved Heteroskedasticity and Autocorrelation Consistent Covariance Matrix Estimator," *Econometrica*, 60, 953–966.
- Berndt, Ernst R. and David O. Wood (1975). "Technology, Prices and the Derived Demand for Energy," *Review of Economics and Statistics*, 57(3), 259–268.
- Bollerslev, Tim (1990). "Modelling the Coherence in Short-run Nominal Exchange Rates: A Multivariate Generalized ARCH Model," *The Review of Economics and Statistics*, 72, 498–505.
- Bollerslev, Tim, Robert F. Engle and Jeffrey M. Wooldridge (1988). "A Capital-Asset Pricing Model with Time-varying Covariances," *Journal of Political Economy*, 96(1), 116–131.
- Ding, Zhuangxin and R. F. Engle (2001). "Large Scale Conditional Covariance Matrix Modeling, Estimation and Testing," *Academia Economic Paper*, 29, 157–184.

Engle, Robert F. and K. F. Kroner (1995). "Multivariate Simultaneous Generalized ARCH," *Econometric Theory*, 11, 122-150.

Greene, William H. (1997). *Econometric Analysis*, 3rd Edition, Upper Saddle River, NJ: Prentice Hall.

Newey, Whitney and Kenneth West (1994). "Automatic Lag Selection in Covariance Matrix Estimation," *Review of Economic Studies*, 61, 631-653.

Chapter 34. Vector Autoregression and Error Correction Models

The structural approach to time series modeling uses economic theory to model the relationship among the variables of interest. Unfortunately, economic theory is often not rich enough to provide a dynamic specification that identifies all of these relationships. Furthermore, estimation and inference are complicated by the fact that endogenous variables may appear on both the left and right sides of equations.

These problems lead to alternative, non-structural approaches to modeling the relationship among several variables. This chapter describes the estimation and analysis of vector autoregression (VAR) and the vector error correction (VEC) models. We also describe tools for testing the presence of cointegrating relationships among several non-stationary variables.

Vector Autoregressions (VARs)

The vector autoregression (VAR) is commonly used for forecasting systems of interrelated time series and for analyzing the dynamic impact of random disturbances on the system of variables. The VAR approach sidesteps the need for structural modeling by treating every endogenous variable in the system as a function of the lagged values of all of the endogenous variables in the system.

The mathematical representation of a VAR is:

$$y_t = A_1 y_{t-1} + \dots + A_p y_{t-p} + B x_t + \epsilon_t \quad (34.1)$$

where y_t is a k vector of endogenous variables, x_t is a d vector of exogenous variables, A_1, \dots, A_p and B are matrices of coefficients to be estimated, and ϵ_t is a vector of innovations that may be contemporaneously correlated but are uncorrelated with their own lagged values and uncorrelated with all of the right-hand side variables.

Since only lagged values of the endogenous variables appear on the right-hand side of the equations, simultaneity is not an issue and OLS yields consistent estimates. Moreover, even though the innovations ϵ_t may be contemporaneously correlated, OLS is efficient and equivalent to GLS since all equations have identical regressors.

As an example, suppose that industrial production (IP) and money supply (M1) are jointly determined by a VAR and let a constant be the only exogenous variable. Assuming that the VAR contains two lagged values of the endogenous variables, it may be written as:

$$\begin{aligned} IP_t &= a_{11} IP_{t-1} + a_{12} M1_{t-1} + b_{11} IP_{t-2} + b_{12} M1_{t-2} + c_1 + \epsilon_{1t} \\ M1_t &= a_{21} IP_{t-1} + a_{22} M1_{t-1} + b_{21} IP_{t-2} + b_{22} M1_{t-2} + c_2 + \epsilon_{2t} \end{aligned} \quad (34.2)$$

where a_{ij} , b_{ij} , c_i are the parameters to be estimated.

Estimating a VAR in EViews

To specify a VAR in EViews, you must first create a var object. Select **Quick/Estimate VAR...** or type `var` in the command window. The **Basics** tab of the VAR Specification dialog will prompt you to define the structure of your VAR.

You should fill out the dialog with the appropriate information:

- Select the VAR type: **Unrestricted VAR** or **Vector Error Correction (VEC)**. What we have been calling a VAR is actually an *unrestricted* VAR. VECs are explained below.
- Set the estimation sample.
- Enter the lag specification in the appropriate edit box. This information is entered in pairs: each pair of numbers defines a *range* of lags. For example, the lag pair shown above:

1 4

tells EViews to use the first *through* fourth lags of all the endogenous variables in the system as right-hand side variables.

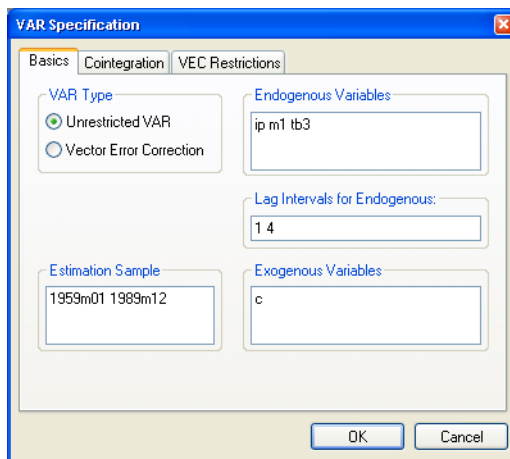
You can add any number of lag intervals, all entered in pairs. The lag specification:

2 4 6 9 12 12

uses lags 2–4, 6–9, and 12.

- Enter the names of endogenous and exogenous series in the appropriate edit boxes. Here we have listed M1, IP, and TB3 as endogenous series, and have used the special series `c` as the constant exogenous term. If either list of series were longer, we could have created a named group object containing the list and then entered the group name.

The remaining dialog tabs (**Cointegration** and **Restrictions**) are relevant only for VEC models and are explained below.



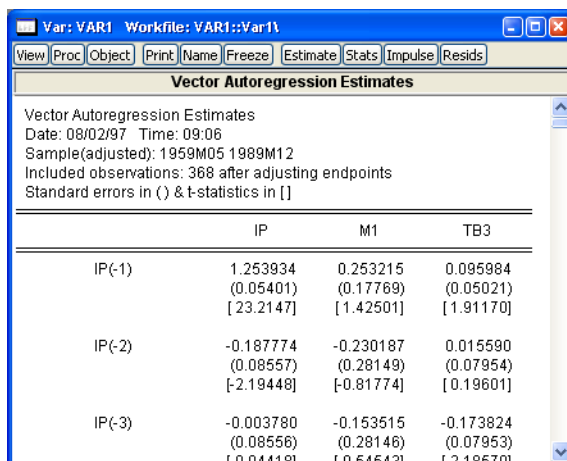
VAR Estimation Output

Once you have specified the VAR, click **OK**. EViews will display the estimation results in the VAR window.

Each column in the table corresponds to an equation in the VAR. For each right-hand side variable, EViews reports the estimated coefficient, its standard error, and the t -statistic. For example, the coefficient for IP(-1) in the TB3 equation is 0.095984.

EViews displays additional information below the coefficient summary. The first part of the additional output presents standard OLS regression statistics for each equation. The results are

computed separately for each equation using the appropriate residuals and are displayed in the corresponding column. The numbers at the very bottom of the table are the summary statistics for the VAR system as a whole.



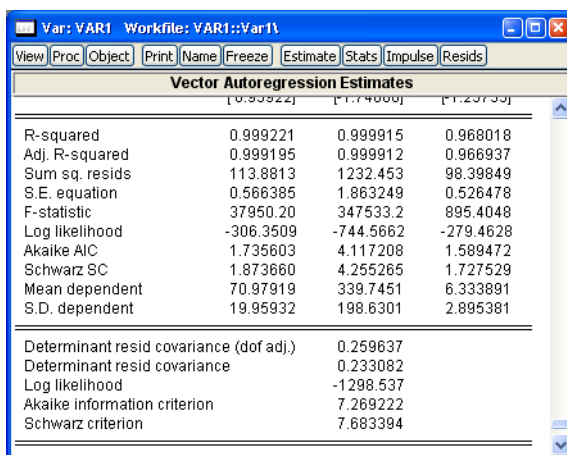
Var: VAR1 Workfile: VAR1::Var1

View Proc Object Print Name Freeze Estimate Stats Impulse Resids

Vector Autoregression Estimates

Vector Autoregression Estimates
Date: 08/02/97 Time: 09:06
Sample(adjusted): 1959M05 1989M12
Included observations: 368 after adjusting endpoints
Standard errors in () & t-statistics in []

| | IP | M1 | TB3 |
|--------|--------------------------------------|--------------------------------------|--------------------------------------|
| IP(-1) | 1.253934 (0.05401) [23.2147] | 0.253215 (0.17769) [1.42501] | 0.095984 (0.05021) [1.91170] |
| IP(-2) | -0.187774 (0.08557) [-2.19448] | -0.230187 (0.28149) [-0.81774] | 0.015590 (0.07954) [0.19601] |
| IP(-3) | -0.003780 (0.08556) [-0.04418] | -0.153515 (0.28146) [-0.54543] | -0.173824 (0.07953) [-2.18570] |



Var: VAR1 Workfile: VAR1::Var1

View Proc Object Print Name Freeze Estimate Stats Impulse Resids

Vector Autoregression Estimates

[0.999221] [0.999115] [0.968018]

| | | | |
|----------------|-----------|-----------|-----------|
| R-squared | 0.999221 | 0.999115 | 0.968018 |
| Adj. R-squared | 0.999195 | 0.999112 | 0.966937 |
| Sum sq. resids | 113.8813 | 1232.453 | 98.39849 |
| S.E. equation | 0.566385 | 1.863249 | 0.526478 |
| F-statistic | 37950.20 | 347533.2 | 895.4048 |
| Log likelihood | -306.3509 | -744.5662 | -279.4628 |
| Akaike AIC | 1.735603 | 4.117208 | 1.589472 |
| Schwarz SC | 1.873660 | 4.255265 | 1.727529 |
| Mean dependent | 70.97919 | 339.7451 | 6.333891 |
| S.D. dependent | 19.95932 | 198.6301 | 2.895381 |

| | |
|---|-----------|
| Determinant resid covariance (dof adj.) | 0.259637 |
| Determinant resid covariance | 0.233082 |
| Log likelihood | -1298.537 |
| Akaike information criterion | 7.269222 |
| Schwarz criterion | 7.683394 |

The determinant of the residual covariance (degree of freedom adjusted) is computed as:

$$|\hat{\Omega}| = \det\left(\frac{1}{T-p} \sum_t \hat{\epsilon}_t \hat{\epsilon}_t'\right) \quad (34.3)$$

where p is the number of parameters per equation in the VAR. The unadjusted calculation ignores the p . The log likelihood value is computed assuming a multivariate normal (Gaussian) distribution as:

$$l = -\frac{T}{2} \{ k(1 + \log 2\pi) + \log |\hat{\Omega}| \} \quad (34.4)$$

The two information criteria are computed as:

$$\begin{aligned} AIC &= -2l/T + 2n/T \\ SC &= -2l/T + n \log T/T \end{aligned} \quad (34.5)$$

where $n = k(d + pk)$ is the total number of estimated parameters in the VAR. These information criteria can be used for model selection such as determining the lag length of the VAR, with smaller values of the information criterion being preferred. It is worth noting that some reference sources may define the AIC/SC differently, either omitting the “inessential” constant terms from the likelihood, or not dividing by T (see also [Appendix F. “Information Criteria,”](#) on page 645 for additional discussion of information criteria).

Views and Procs of a VAR

Once you have estimated a VAR, EViews provides various views to work with the estimated VAR. In this section, we discuss views that are specific to VARs. For other views and procedures, see the general discussion of system views in [Chapter 33. “System Estimation,”](#) beginning on page 307.

Diagnostic Views

A set of diagnostic views are provided under the menus **View/Lag Structure** and **View/Residual Tests** in the VAR window. These views should help you check the appropriateness of the estimated VAR.

Lag Structure

AR Roots Table/Graph

Reports the *inverse* roots of the characteristic AR polynomial; see Lütkepohl (1991). The estimated VAR is stable (stationary) if all roots have modulus less than one and lie inside the unit circle. If the VAR is not stable, certain results (such as impulse response standard errors) are not valid. There will be kp roots, where k is the number of endogenous variables and p is the largest lag. If you estimated a VEC with r cointegrating relations, $k - r$ roots should be equal to unity.

Pairwise Granger Causality Tests

Carries out pairwise Granger causality tests and tests whether an endogenous variable can be treated as exogenous. For each equation in the VAR, the output displays χ^2 (Wald) statistics for the joint significance of each of the other lagged endogenous variables in that equation. The statistic in the last row (**All**) is the χ^2 statistic for joint significance of all other lagged endogenous variables in the equation.

Warning: if you have estimated a VEC, the lagged variables that are tested for exclusion are only those that are first differenced. The lagged level terms in the cointegrating equations (the error correction terms) are not tested.

Lag Exclusion Tests

Carries out lag exclusion tests for each lag in the VAR. For each lag, the χ^2 (Wald) statistic for the joint significance of all endogenous variables at that lag is reported for each equation separately and jointly (last column).

Lag Length Criteria

Computes various criteria to select the lag order of an unrestricted VAR. You will be prompted to specify the maximum lag to “test” for. The table displays various information criteria for all lags up to the specified maximum. (If there are no exogenous variables in the VAR, the lag starts at 1; otherwise the lag starts at 0.) The table indicates the selected lag from each column criterion by an asterisk “*”. For columns 4–7, these are the lags with the smallest value of the criterion.

All the criteria are discussed in Lütkepohl (1991, Section 4.3). The sequential modified likelihood ratio (LR) test is carried out as follows. Starting from the maximum lag, test the hypothesis that the coefficients on lag l are jointly zero using the χ^2 statistics:

$$LR = (T - m) \{ \log |\Omega_{\ell-1}| - \log |\Omega_{\ell}| \} \sim \chi^2(k^2) \quad (34.6)$$

where m is the number of parameters per equation under the alternative. Note that we employ Sims’ (1980) small sample modification which uses $(T - m)$ rather than T . We compare the modified LR statistics to the 5% critical values starting from the maximum lag, and decreasing the lag one at a time until we first get a rejection. The alternative lag order from the first rejected test is marked with an asterisk (if no test rejects, the minimum lag will be marked with an asterisk). It is worth emphasizing that even though the individual tests have size 0.05, the overall size of the test will not be 5%; see the discussion in Lütkepohl (1991, p. 125–126).

Residual Tests

Correlograms

Displays the pairwise cross-correlograms (sample autocorrelations) for the estimated residuals in the VAR for the specified number of lags. The cross-correlograms can be displayed in three different formats. There are two tabular forms, one ordered by variables (**Tabulate by Variable**) and one ordered by lags (**Tabulate by Lag**). The **Graph** form displays a matrix of pairwise cross-correlograms. The dotted line in the graphs represent plus or minus two times the asymptotic standard errors of the lagged correlations (computed as $1/\sqrt{T}$).

Portmanteau Autocorrelation Test

Computes the multivariate Box-Pierce/Ljung-Box Q -statistics for residual serial correlation up to the specified order (see Lütkepohl, 1991, 4.4.21 & 4.4.23 for details). We report both the Q -statistics and the adjusted Q -statistics (with a small sample correction). Under the null hypothesis of no serial correlation up to lag h , both statistics are *approximately* distributed χ^2 with degrees of freedom $k^2(h-p)$ where p is the VAR lag order. The asymptotic distribution is approximate in the sense that it requires the MA coefficients to be zero for lags $i > h-p$. Therefore, this approximation will be poor if the roots of the AR polynomial are close to one and h is small. In fact, the degrees of freedom becomes negative for $h < p$.

Autocorrelation LM Test

Reports the multivariate LM test statistics for residual serial correlation up to the specified order. The test statistic for lag order h is computed by running an auxiliary regression of the residuals u_t on the original right-hand regressors and the lagged residual u_{t-h} , where the missing first h values of u_{t-h} are filled with zeros. See Johansen (1995, p. 22) for the formula of the LM statistic. Under the null hypothesis of no serial correlation of order h , the LM statistic is asymptotically distributed χ^2 with k^2 degrees of freedom.

Normality Test

Reports the multivariate extensions of the Jarque-Bera residual normality test, which compares the third and fourth moments of the residuals to those from the normal distribution. For the multivariate test, you must choose a factorization of the k residuals that are orthogonal to each other (see “[Impulse Responses](#)” on page 353 for additional discussion of the need for orthogonalization).

Let P be a $k \times k$ factorization matrix such that:

$$v_t = Pu_t \sim N(0, I_k) \quad (34.7)$$

where u_t is the demeaned residuals. Define the third and fourth moment vectors

$m_3 = \sum_t v_t^3 / T$ and $m_4 = \sum_t v_t^4 / T$. Then:

$$\sqrt{T} \begin{bmatrix} m_3 \\ m_4 - 3 \end{bmatrix} \rightarrow N \left(0, \begin{bmatrix} 6I_k & 0 \\ 0 & 24I_k \end{bmatrix} \right) \quad (34.8)$$

under the null hypothesis of normal distribution. Since each component is independent of each other, we can form a χ^2 statistic by summing squares of any of these third and fourth moments.

EViews provides you with choices for the factorization matrix P :

- **Cholesky** (Lütkepohl 1991, p. 155-158): P is the inverse of the lower triangular Cholesky factor of the residual covariance matrix. The resulting test statistics depend on the ordering of the variables in the VAR.

- **Inverse Square Root of Residual Correlation Matrix** (Doornik and Hansen 1994): $P = H\Lambda^{-1/2}H'V$ where Λ is a diagonal matrix containing the eigenvalues of the residual correlation matrix on the diagonal, H is a matrix whose columns are the corresponding eigenvectors, and V is a diagonal matrix containing the inverse square root of the residual variances on the diagonal. This P is essentially the inverse square root of the residual correlation matrix. The test is invariant to the ordering and to the scale of the variables in the VAR. As suggested by Doornik and Hansen (1994), we perform a small sample correction to the transformed residuals v_t before computing the statistics.
- **Inverse Square Root of Residual Covariance Matrix** (Urzua 1997): $P = GD^{-1/2}G'$ where D is the diagonal matrix containing the eigenvalues of the residual covariance matrix on the diagonal and G is a matrix whose columns are the corresponding eigenvectors. This test has a specific alternative, which is the quartic exponential distribution. According to Urzua, this is the “most likely” alternative to the multivariate normal with finite fourth moments since it can approximate the multivariate Pearson family “as close as needed.” As recommended by Urzua, we make a small sample correction to the transformed residuals v_t before computing the statistics. This small sample correction differs from the one used by Doornik and Hansen (1994); see Urzua (1997, Section D).
- **Factorization from Identified (Structural) VAR**: $P = B^{-1}A$ where A , B are estimated from the structural VAR model. This option is available only if you have estimated the factorization matrices A and B using the structural VAR (see [page 357](#), below).

EViews reports test statistics for each orthogonal component (labeled RESID1, RESID2, and so on) and for the joint test. For individual components, the estimated skewness m_3 and kurtosis m_4 are reported in the first two columns together with the p -values from the $\chi^2(1)$ distribution (in square brackets). The Jarque-Bera column reports:

$$T \left\{ \frac{m_3^2}{6} + \frac{(m_4 - 3)^2}{24} \right\} \quad (34.9)$$

with p -values from the $\chi^2(2)$ distribution. *Note: in contrast to the Jarque-Bera statistic computed in the series view, this statistic is not computed using a degrees of freedom correction.*

For the joint tests, we will generally report:

$$\begin{aligned} \lambda_3 &= Tm_3'm_3/6 \rightarrow \chi^2(k) \\ \lambda_4 &= T(m_4 - 3)'(m_4 - 3)/24 \rightarrow \chi^2(k) \\ \lambda &= \lambda_3 + \lambda_4 \rightarrow \chi^2(2k). \end{aligned} \quad (34.10)$$

If, however, you choose Urzua's (1997) test, λ will not only use the sum of squares of the "pure" third and fourth moments but will also include the sum of squares of all *cross* third and fourth moments. In this case, λ is asymptotically distributed as a χ^2 with $k(k+1)(k+2)(k+7)/24$ degrees of freedom.

White Heteroskedasticity Test

These tests are the extension of White's (1980) test to systems of equations as discussed by Kelejian (1982) and Doornik (1995). The test regression is run by regressing each cross product of the residuals on the cross products of the regressors and testing the joint significance of the regression. The **No Cross Terms** option uses only the levels and squares of the original regressors, while the **With Cross Terms** option includes all non-redundant cross-products of the original regressors in the test equation. The test regression always includes a constant term as a regressor.

The first part of the output displays the joint significance of the regressors excluding the constant term for each test regression. You may think of each test regression as testing the constancy of each element in the residual covariance matrix separately. Under the null of no heteroskedasticity or (no misspecification), the non-constant regressors should not be jointly significant.

The last line of the output table shows the LM chi-square statistics for the joint significance of all regressors in the system of test equations (see Doornik, 1995, for details). The system LM statistic is distributed as a χ^2 with degrees of freedom mn , where $m = k(k+1)/2$ is the number of cross-products of the residuals in the system and n is the number of the common set of right-hand side variables in the test regression.

Notes on Comparability

Many of the diagnostic tests given above may be computed "manually" by estimating the VAR using a system object and selecting **View/Wald Coefficient Tests...** We caution you that the results from the system will not match those from the VAR diagnostic views for various reasons:

- The system object will, in general, use the maximum possible observations for each equation in the system. By contrast, VAR objects force a balanced sample in case there are missing values.
- The estimates of the weighting matrix used in system estimation do not contain a degrees of freedom correction (the residual sums-of-squares are divided by T rather than by $T - k$), while the VAR estimates do perform this adjustment. Even though estimated using comparable specifications and yielding identifiable coefficients, the test statistics from system SUR and the VARs will show small (asymptotically insignificant) differences.

Impulse Responses

A shock to the i -th variable not only directly affects the i -th variable but is also transmitted to all of the other endogenous variables through the dynamic (lag) structure of the VAR. An impulse response function traces the effect of a one-time shock to one of the innovations on current and future values of the endogenous variables.

If the innovations ϵ_t are contemporaneously uncorrelated, interpretation of the impulse response is straightforward. The i -th innovation $\epsilon_{i,t}$ is simply a shock to the i -th endogenous variable $y_{i,t}$. Innovations, however, are usually correlated, and may be viewed as having a common component which cannot be associated with a specific variable. In order to interpret the impulses, it is common to apply a transformation P to the innovations so that they become uncorrelated:

$$v_t = P\epsilon_t \sim (0, D) \quad (34.11)$$

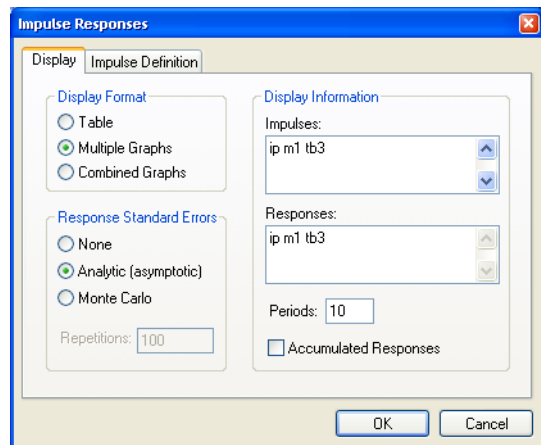
where D is a *diagonal* covariance matrix. As explained below, EViews provides several options for the choice of P .

To obtain the impulse response functions, first estimate a VAR. Then select **View/Impulse Response...** from the VAR toolbar. You will see a dialog box with two tabs: **Display** and **Impulse Definition**.

The **Display** tab provides the following options:

- Display Format:** displays results as a table or graph. Keep in mind that if you choose the **Combined Graphs** option, the **Response Standard Errors** option will be ignored and the standard errors will not be displayed. Note also that the output table format is ordered by response variables, not by impulse variables.
- Display Information:** you should enter the variables for which you wish to generate innovations (**Impulses**) and the variables for which you wish to observe the responses (**Responses**). You may either enter the name of the endogenous variables or the numbers corresponding to the ordering of the variables. For example, if you specified the VAR as GDP, M1, CPI, then you may either type,

GDP CPI M1



or,

1 3 2

The order in which you enter these variables only affects the display of results.

You should also specify a positive integer for the number of periods to trace the response function. To display the accumulated responses, check the **Accumulate Response** box. For stationary VARs, the impulse responses should die out to zero and the accumulated responses should asymptote to some (non-zero) constant.

- **Response Standard Errors:** provides options for computing the response standard errors. Note that analytic and/or Monte Carlo standard errors are currently not available for certain **Impulse** options and for vector error correction (VEC) models. If you choose **Monte Carlo** standard errors, you should also specify the number of repetitions to use in the appropriate edit box.

If you choose the table format, the estimated standard errors will be reported in parentheses below the responses. If you choose to display the results in multiple graphs, the graph will contain the plus/minus two standard error bands about the impulse responses. The standard error bands are not displayed in combined graphs.

The **Impulse** tab provides the following options for transforming the impulses:

- **Residual—One Unit** sets the impulses to one unit of the residuals. This option ignores the units of measurement and the correlations in the VAR residuals so that no transformation is performed. The responses from this option are the MA coefficients of the infinite MA order Wold representation of the VAR.
- **Residual—One Std. Dev.** sets the impulses to one standard deviation of the residuals. This option ignores the correlations in the VAR residuals.
- **Cholesky** uses the inverse of the Cholesky factor of the residual covariance matrix to orthogonalize the impulses. This option imposes an ordering of the variables in the VAR and attributes all of the effect of any common component to the variable that comes first in the VAR system. Note that responses can change dramatically if you change the ordering of the variables. You may specify a different VAR ordering by reordering the variables in the **Cholesky Ordering** edit box.

The **(d.f. adjustment)** option makes a small sample degrees of freedom correction when estimating the residual covariance matrix used to derive the Cholesky factor. The (i,j) -th element of the residual covariance matrix with degrees of freedom correction is computed as $\sum_t e_{i,t} e_{j,t} / (T - p)$ where p is the number of parameters per equation in the VAR. The **(no d.f. adjustment)** option estimates the (i,j) -th element of the residual covariance matrix as $\sum_t e_{i,t} e_{j,t} / T$. *Note: early versions of EViews computed the impulses using the Cholesky factor from the residual covariance matrix with no degrees of freedom adjustment.*

- **Generalized Impulses** as described by Pesaran and Shin (1998) constructs an orthogonal set of innovations that does not depend on the VAR ordering. The generalized impulse responses from an innovation to the j -th variable are derived by applying a variable specific Cholesky factor computed with the j -th variable at the top of the Cholesky ordering.
- **Structural Decomposition** uses the orthogonal transformation estimated from the structural factorization matrices. This approach is not available unless you have estimated the structural factorization matrices as explained in [“Structural \(Identified\) VARs” on page 357](#).
- **User Specified** allows you to specify your own impulses. Create a matrix (or vector) that contains the impulses and type the name of that matrix in the edit box. If the VAR has k endogenous variables, the impulse matrix must have k rows and 1 or k columns, where each column is a impulse vector.

For example, say you have a $k = 3$ variable VAR and wish to apply simultaneously a positive one unit shock to the first variable and a negative one unit shock to the second variable. Then you will create a 3×1 impulse matrix containing the values 1, -1, and 0. Using commands, you can enter:

```
matrix(3,1) shock
shock.fill(by=c) 1,-1,0
```

and type the name of the matrix SHOCK in the edit box.

Variance Decomposition

While impulse response functions trace the effects of a shock to one endogenous variable on to the other variables in the VAR, *variance decomposition* separates the variation in an endogenous variable into the component shocks to the VAR. Thus, the variance decomposition provides information about the relative importance of each random innovation in affecting the variables in the VAR.

To obtain the variance decomposition, select **View/Variance Decomposition...** from the var object toolbar. You should provide the same information as for impulse responses above. Note that since non-orthogonal factorization will yield decompositions that do not satisfy an adding up property, your choice of factorization is limited to orthogonal factorizations.

The table format displays a separate variance decomposition for each endogenous variable. The second column, labeled “S.E.”, contains the forecast error of the variable at the given forecast horizon. The source of this forecast error is the variation in the current and future values of the innovations to each endogenous variable in the VAR. The remaining columns give the percentage of the forecast variance due to each innovation, with each row adding up to 100.

As with the impulse responses, the variance decomposition based on the Cholesky factor can change dramatically if you alter the ordering of the variables in the VAR. For example, the first period decomposition for the first variable in the VAR ordering is completely due to its own innovation.

Factorization based on structural orthogonalization is available only if you have estimated the structural factorization matrices as explained in “[Structural \(Identified\) VARs](#)” on [page 357](#). Note that the forecast standard errors should be identical to those from the Cholesky factorization *if the structural VAR is just identified*. For over-identified structural VARs, the forecast standard errors may differ in order to maintain the adding up property.

Procs of a VAR

Most of the procedures available for a VAR are common to those available for a system object (see “[System Procs](#)” on [page 323](#)). Here, we discuss only those procedures that are unique to the VAR object.

Make System

This proc creates a system object that contains an equivalent VAR specification. If you want to estimate a non-standard VAR, you may use this proc as a quick way to specify a VAR in a system object which you can then modify to meet your needs. For example, while the VAR object requires each equation to have the same lag structure, you may want to relax this restriction. To estimate a VAR with unbalanced lag structure, use the **Proc/Make System** procedure to create a VAR system with a balanced lag structure and edit the system specification to meet the desired lag specification.

The **By Variable** option creates a system whose specification (and coefficient number) is ordered by variables. Use this option if you want to edit the specification to exclude lags of a specific variable from some of the equations. The **By Lag** option creates a system whose specification (and coefficient number) is ordered by lags. Use this option if you want to edit the specification to exclude certain lags from some of the equations.

For vector error correction (VEC) models, treating the coefficients of the cointegrating vector as additional unknown coefficients will make the resulting system unidentified. In this case, EViews will create a system object where the coefficients for the cointegrating vectors are fixed at the estimated values from the VEC. If you want to estimate the coefficients of the cointegrating vector in the system, you may edit the specification, but you should make certain that the resulting system is identified.

You should also note that while the standard VAR can be estimated efficiently by equation-by-equation OLS, this is generally not the case for the modified specification. You may wish to use one of the system-wide estimation methods (e.g. SUR) when estimating non-standard VARs using the system object.

Estimate Structural Factorization

This procedure is used to estimate the factorization matrices for a structural (or identified) VAR. The full details of this procedure is given in [“Structural \(Identified\) VARs” on page 357](#). You must first estimate the structural factorization matrices using this proc in order to use the structural options in impulse responses and variance decompositions.

Structural (Identified) VARs

The main purpose of structural VAR (SVAR) estimation is to obtain non-recursive orthogonalization of the error terms for impulse response analysis. This alternative to the recursive Cholesky orthogonalization requires the user to impose enough restrictions to identify the orthogonal (structural) components of the error terms.

Let y_t be a k -element vector of the endogenous variables and let $\Sigma = E[e_t e_t']$ be the residual covariance matrix. Following Amisano and Giannini (1997), the class of SVAR models that EViews estimates may be written as:

$$A e_t = B u_t \quad (34.12)$$

where e_t and u_t are vectors of length k . e_t is the observed (or reduced form) residuals, while u_t is the unobserved structural innovations. A and B are $k \times k$ matrices to be estimated. The structural innovations u_t are assumed to be orthonormal, *i.e.* its covariance matrix is an identity matrix $E[u_t u_t'] = I$. The assumption of orthonormal innovations u_t imposes the following identifying restrictions on A and B :

$$A \Sigma A' = B B' . \quad (34.13)$$

Noting that the expressions on either side of (34.13) are symmetric, this imposes $k(k+1)/2$ restrictions on the $2k^2$ unknown elements in A and B . Therefore, in order to identify A and B , you need to supply at least $2k^2 - k(k+1)/2 = k(3k-1)/2$ additional restrictions.

Specifying the Identifying Restrictions

As explained above, in order to estimate the orthogonal factorization matrices A and B , you need to provide additional identifying restrictions. We distinguish two types of identifying restrictions: *short-run* and *long-run*. For either type, the identifying restrictions can be specified either in text form or by pattern matrices.

Short-run Restrictions by Pattern Matrices

For many problems, the identifying restrictions on the A and B matrices are simple zero exclusion restrictions. In this case, you can specify the restrictions by creating a named “pattern” matrix for A and B . Any elements of the matrix that you want to be estimated

should be assigned a missing value “NA”. All non-missing values in the pattern matrix will be held fixed at the specified values.

For example, suppose you want to restrict A to be a lower triangular matrix with ones on the main diagonal and B to be a diagonal matrix. Then the pattern matrices (for a $k = 3$ variable VAR) would be:

$$A = \begin{pmatrix} 1 & 0 & 0 \\ \text{NA} & 1 & 0 \\ \text{NA} & \text{NA} & 1 \end{pmatrix}, \quad B = \begin{pmatrix} \text{NA} & 0 & 0 \\ 0 & \text{NA} & 0 \\ 0 & 0 & \text{NA} \end{pmatrix}. \quad (34.14)$$

You can create these matrices interactively. Simply use **Object/New Object...** to create two new 3×3 matrices, A and B , and then use the spreadsheet view to edit the values. Alternatively, you can issue the following commands:

```
matrix(3,3) pata
' fill matrix in row major order
pata.fill(by=r) 1,0,0, na,1,0, na,na,1
matrix(3,3) patb = 0
patb(1,1) = na
patb(2,2) = na
patb(3,3) = na
```

Once you have created the pattern matrices, select **Proc/Estimate Structural Factorization...** from the VAR window menu. In the **SVAR Options** dialog, click the **Matrix** button and the **Short-Run Pattern** button and type in the name of the pattern matrices in the relevant edit boxes.

Short-run Restrictions in Text Form

For more general restrictions, you can specify the identifying restrictions in text form. In text form, you will write out the relation $Ae_t = Bu_t$ as a set of equations, identifying each element of the e_t and u_t vectors with special symbols. Elements of the A and B matrices to be estimated must be specified as elements of a coefficient vector.

To take an example, suppose again that you have a $k = 3$ variable VAR where you want to restrict A to be a lower triangular matrix with ones on the main diagonal and B to be a diagonal matrix. Under these restrictions, the relation $Ae_t = Bu_t$ can be written as:

$$\begin{aligned} e_1 &= b_{11}u_1 \\ e_2 &= -a_{21}e_1 + b_{22}u_2 \\ e_3 &= -a_{31}e_1 - a_{32}e_2 + b_{33}u_3 \end{aligned} \quad (34.15)$$

To specify these restrictions in text form, select **Proc/Estimate Structural Factorization...** from the VAR window and click the **Text** button. In the edit window, you should type the following:

```
@e1 = c(1)*@u1
@e2 = -c(2)*@e1 + c(3)*@u2
@e3 = -c(4)*@e1 - c(5)*@e2 + c(6)*@u3
```

The special key symbols “@e1,” “@e2,” “@e3,” represent the first, second, and third elements of the e_t vector, while “@u1,” “@u2,” “@u3” represent the first, second, and third elements of the u_t vector. In this example, all unknown elements of the A and B matrices are represented by elements of the c coefficient vector.

Long-run Restrictions

The identifying restrictions embodied in the relation $Ae = Bu$ are commonly referred to as short-run restrictions. Blanchard and Quah (1989) proposed an alternative identification method based on restrictions on the long-run properties of the impulse responses. The (accumulated) long-run response C to structural innovations takes the form:

$$C = \hat{\Psi}_{\infty} A^{-1} B \quad (34.16)$$

where $\hat{\Psi}_{\infty} = (I - \hat{A}_1 - \dots - \hat{A}_p)^{-1}$ is the estimated accumulated responses to the reduced form (observed) shocks. Long-run identifying restrictions are specified in terms of the elements of this C matrix, typically in the form of zero restrictions. The restriction $C_{i,j} = 0$ means that the (accumulated) response of the i -th variable to the j -th structural shock is zero in the long-run.

It is important to note that the expression for the long-run response (34.16) involves the inverse of A . Since EViews currently requires all restrictions to be linear in the elements of A and B , *if you specify a long-run restriction, the A matrix must be the identity matrix.*

To specify long-run restrictions by a pattern matrix, create a named matrix that contains the pattern for the long-run response matrix C . Unrestricted elements in the C matrix should be assigned a missing value “NA”. For example, suppose you have a $k = 2$ variable VAR where you want to restrict the long-run response of the second endogenous variable to the first structural shock to be zero $C_{2,1} = 0$. Then the long-run response matrix will have the following pattern:

$$C = \begin{pmatrix} \text{NA} & \text{NA} \\ 0 & \text{NA} \end{pmatrix} \quad (34.17)$$

You can create this matrix with the following commands:

```
matrix(2,2) patc = na
patc(2,1) = 0
```

Once you have created the pattern matrix, select **Proc/Estimate Structural Factorization...** from the VAR window menu. In the **SVAR Options** dialog, click the **Matrix** button and the **Long-Run Pattern** button and type in the name of the pattern matrix in the relevant edit box.

To specify the same long-run restriction in text form, select **Proc/Estimate Structural Factorization...** from the VAR window and click the **Text** button. In the edit window, you would type the following:

```
@lr2(@u1)=0 ' zero LR response of 2nd variable to 1st shock
```

where everything on the line after the apostrophe is a comment. This restriction begins with the special keyword “@LR#”, with the “#” representing the response variable to restrict. Inside the parentheses, you must specify the impulse keyword “@U” and the innovation number, followed by an equal sign and the value of the response (typically 0). We caution you that while you can list multiple long-run restrictions, *you cannot mix short-run and long-run restrictions*.

Note that it is possible to specify long-run restrictions as short-run restrictions (by obtaining the infinite MA order representation). While the estimated A and B matrices should be the same, the impulse response standard errors from the short-run representation would be incorrect (since it does not take into account the uncertainty in the estimated infinite MA order coefficients).

Some Important Notes

Currently we have the following limitations for the specification of identifying restrictions:

- The A and B matrices must be square and non-singular. In text form, there must be exactly as many equations as there are endogenous variables in the VAR. For short-run restrictions in pattern form, you must provide the pattern matrices for both A and B matrices.
- The restrictions must be linear in the elements of A and B . Moreover, the restrictions on A and B must be independent (no restrictions across elements of A and B).
- You cannot impose both short-run and long-run restrictions.
- Structural decompositions are currently not available for VEC models.
- The identifying restriction assumes that the structural innovations u_t have unit variances. Therefore, you will almost always want to estimate the diagonal elements of the B matrix so that you obtain estimates of the standard deviations of the structural shocks.
- It is common in the literature to assume that the structural innovations have a diagonal covariance matrix rather than an identity matrix. To compare your results to those

from these studies, you will have to divide each column of the B matrix with the diagonal element in that column (so that the resulting B matrix has ones on the main diagonal). To illustrate this transformation, consider a simple $k = 2$ variable model with $A = 1$:

$$\begin{aligned} e_{1,t} &= b_{11}u_{1,t} + b_{12}u_{2,t} \\ e_{2,t} &= b_{21}u_{1,t} + b_{22}u_{2,t} \end{aligned} \quad (34.18)$$

where $u_{1,t}$ and $u_{2,t}$ are independent structural shocks with unit variances as assumed in the EViews specification. To rewrite this specification with a B matrix containing ones on the main diagonal, define a new set of structural shocks by the transformations $v_{1,t} = b_{11}u_{1,t}$ and $v_{2,t} = b_{22}u_{2,t}$. Then the structural relation can be rewritten as,

$$\begin{aligned} e_{1,t} &= v_{1,t} + (b_{12}/b_{22})v_{2,t} \\ e_{2,t} &= (b_{21}/b_{11})v_{1,t} + v_{2,t} \end{aligned} \quad (34.19)$$

where now:

$$B = \begin{pmatrix} 1 & b_{12}/b_{22} \\ b_{21}/b_{11} & 1 \end{pmatrix}, \quad v_t = \begin{bmatrix} v_{1,t} \\ v_{2,t} \end{bmatrix} \sim \left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} b_{11}^2 & 0 \\ 0 & b_{22}^2 \end{bmatrix} \right) \quad (34.20)$$

Note that the transformation involves only rescaling elements of the B matrix and not on the A matrix. For the case where B is a diagonal matrix, the elements in the main diagonal are simply the estimated standard deviations of the structural shocks.

Identification Conditions

As stated above, the assumption of orthonormal structural innovations imposes $k(k+1)/2$ restrictions on the $2k^2$ unknown elements in A and B , where k is the number of endogenous variables in the VAR. In order to identify A and B , you need to provide at least $k(k+1)/2 - 2k^2 = k(3k-1)/2$ additional identifying restrictions. This is a necessary order condition for identification and is checked by counting the number of restrictions provided.

As discussed in Amisano and Giannini (1997), a sufficient condition for local identification can be checked by the invertibility of the “augmented” information matrix (see Amisano and Giannini, 1997). This local identification condition is evaluated numerically at the starting values. If EViews returns a singularity error message for different starting values, you should make certain that your restrictions identify the A and B matrices.

We also require the A and B matrices to be square and non-singular. The non-singularity condition is checked numerically at the starting values. If the A and B matrix is non-singular,

lar at the starting values, an error message will ask you to provide a different set of starting values.

Sign Indeterminacy

For some restrictions, the signs of the A and B matrices are not identified; see Christiano, Eichenbaum, and Evans (1999) for a discussion of this issue. When the sign is indeterminate, we choose a normalization so that the diagonal elements of the factorization matrix $A^{-1}B$ are all positive. This normalization ensures that all structural impulses have positive signs (as does the Cholesky factorization). The default is to always apply this normalization rules whenever applicable. If you do not want to switch the signs, deselect the **Normalize Sign** option from the **Optimization Control** tab of the **SVAR Options** dialog.

Estimation of A and B Matrices

Once you provide the identifying restrictions in any of the forms described above, you are ready to estimate the A and B matrices. Simply click the **OK** button in the **SVAR Options** dialog. You must first estimate these matrices in order to use the structural option in impulse responses and variance decompositions.

A and B are estimated by maximum likelihood, assuming the innovations are multivariate normal. We evaluate the likelihood in terms of unconstrained parameters by substituting out the constraints. The log likelihood is maximized by the method of scoring (with a Marquardt-type diagonal correction—See “Marquardt,” on page 632), where the gradient and expected information matrix are evaluated analytically. See Amisano and Giannini (1997) for the analytic expression of these derivatives.

Optimization Control

Options for controlling the optimization process are provided in the **Optimization Control** tab of the **SVAR Options** dialog. You have the option to specify the starting values, maximum number of iterations, and the convergence criterion.

The starting values are those for the unconstrained parameters after substituting out the constraints. **Fixed** sets all free parameters to the value specified in the edit box. **User Specified** uses the values in the coefficient vector as specified in text form as starting values. For restrictions specified in pattern form, user specified starting values are taken from the first m elements of the default c coefficient vector, where m is the number of free parameters. **Draw from...** options randomly draw the starting values for the free parameters from the specified distributions.

Estimation Output

Once convergence is achieved, EViews displays the estimation output in the VAR window. The point estimates, standard errors, and z -statistics of the estimated free parameters are reported together with the maximized value of the log likelihood. The estimated standard

errors are based on the inverse of the estimated information matrix (negative expected value of the Hessian) evaluated at the final estimates.

For overidentified models, we also report the LR test for over-identification. The LR test statistic is computed as:

$$LR = 2(l_u - l_r) = T(\text{tr}(P) - \log|P| - k) \quad (34.21)$$

where $P = A'B^{-T}B^{-1}A\Sigma$. Under the null hypothesis that the restrictions are valid, the LR statistic is asymptotically distributed $\chi^2(q - k)$ where q is the number of identifying restrictions.

If you switch the view of the VAR window, you can come back to the previous results (without reestimating) by selecting **View/Estimation Output** from the VAR window. In addition, some of the SVAR estimation results can be retrieved as data members of the VAR; see “[Var Data Members](#)” on page 544 of the *Command Reference* for a list of available VAR data members.

Cointegration Testing

The finding that many macro time series may contain a unit root has spurred the development of the theory of non-stationary time series analysis. Engle and Granger (1987) pointed out that a linear combination of two or more non-stationary series may be stationary. If such a stationary linear combination exists, the non-stationary time series are said to be *cointegrated*. The stationary linear combination is called the *cointegrating equation* and may be interpreted as a long-run equilibrium relationship among the variables.

The purpose of the cointegration test is to determine whether a group of non-stationary series are cointegrated or not. As explained below, the presence of a cointegrating relation forms the basis of the VEC specification. EViews implements VAR-based cointegration tests using the methodology developed in Johansen (1991, 1995).

Consider a VAR of order p :

$$y_t = A_1 y_{t-1} + \dots + A_p y_{t-p} + Bx_t + \epsilon_t \quad (34.22)$$

where y_t is a k -vector of non-stationary I(1) variables, x_t is a d -vector of deterministic variables, and ϵ_t is a vector of innovations. We may rewrite this VAR as,

$$\Delta y_t = \Pi y_{t-1} + \sum_{i=1}^{p-1} \Gamma_i \Delta y_{t-i} + Bx_t + \epsilon_t \quad (34.23)$$

where:

$$\Pi = \sum_{i=1}^p A_i - I, \quad \Gamma_i = - \sum_{j=i+1}^p A_j \quad (34.24)$$

Granger's representation theorem asserts that if the coefficient matrix Π has reduced rank $r < k$, then there exist $k \times r$ matrices α and β each with rank r such that $\Pi = \alpha\beta'$ and $\beta'y_t$ is $I(0)$. r is the number of cointegrating relations (the *cointegrating rank*) and each column of β is the cointegrating vector. As explained below, the elements of α are known as the adjustment parameters in the VEC model. Johansen's method is to estimate the Π matrix from an unrestricted VAR and to test whether we can reject the restrictions implied by the reduced rank of Π .

How to Perform a Cointegration Test

To carry out the Johansen cointegration test, select **View/Cointegration Test...** from the group or VAR window toolbar. The **Cointegration Test Specification** page prompts you for information about the test.

Note that since this is a test for cointegration, this test is only valid when you are working with series that are known to be nonstationary. You may wish first to apply unit root tests to each series in the VAR. See [“Unit Root Test” on page 327](#) of the *User's Guide I* for details on carrying out unit root tests in EViews.

Deterministic Trend Specification

Your series may have nonzero means and deterministic trends as well as stochastic trends. Similarly, the cointegrating equations may have intercepts and deterministic trends. The asymptotic distribution of the LR test statistic for cointegration does not have the usual χ^2 distribution and depends on the assumptions made with respect to deterministic trends. Therefore, in order to carry out the test, you need to make an assumption regarding the trend underlying your data.

For each row case in the dialog, the COINTEQ column lists the deterministic variables that appear inside the cointegrating relations (error correction term), while the OUTSIDE column lists the deterministic variables that appear in the VEC equation outside the cointegrating relations. Cases 2 and 4 do not have the same set of deterministic terms in the two columns. For these two cases, some of the deterministic term is restricted to belong only in the cointegrating relation. For cases 3 and 5, the deterministic terms are common in the two columns and the decomposition of the deterministic effects inside and outside the cointegrating space is not uniquely identified; see the technical discussion below.

Johansen Cointegration Test

Cointegration Test Specification

Deterministic trend assumption of test

Assume no deterministic trend in data:

- ☐ 1) No intercept or trend in CE or test VAR
- ☐ 2) Intercept (no trend) in CE - no intercept in VAR

Allow for linear deterministic trend in data:

- ☒ 3) Intercept (no trend) in CE and test VAR
- ☐ 4) Intercept and trend in CE - no trend in VAR

Allow for quadratic deterministic trend in data:

- ☐ 5) Intercept and trend in CE - linear trend in VAR

Summary:

- ☐ 6) Summarize all 5 sets of assumptions

* Critical values may not be valid with exogenous variables; do not include C or Trend.

Exog variables*

Lag intervals

1 4

Lag spec for differenced endogenous

Critical Values

- ☒ MHM
- ☐ Osterwald-Lenum

Size 0.05

OK Cancel

In practice, cases 1 and 5 are rarely used. You should use case 1 only if you know that all series have zero mean. Case 5 may provide a good fit in-sample but will produce implausible forecasts out-of-sample. As a rough guide, use case 2 if none of the series appear to have a trend. For trending series, use case 3 if you believe all trends are stochastic; if you believe some of the series are trend stationary, use case 4.

If you are not certain which trend assumption to use, you may choose the **Summary of all 5 trend assumptions** option (case 6) to help you determine the choice of the trend assumption. This option indicates the number of cointegrating relations under each of the 5 trend assumptions, and you will be able to assess the sensitivity of the results to the trend assumption.

Technical Discussion

EViews considers the following five deterministic trend cases considered by Johansen (1995, p. 80–84):

1. The level data y_t have no deterministic trends and the cointegrating equations do not have intercepts:

$$H_2(r): \Pi y_{t-1} + Bx_t = \alpha\beta' y_{t-1}$$

2. The level data y_t have no deterministic trends and the cointegrating equations have intercepts:

$$H_1^*(r): \Pi y_{t-1} + Bx_t = \alpha(\beta' y_{t-1} + \rho_0)$$

3. The level data y_t have linear trends but the cointegrating equations have only intercepts:

$$H_1(r): \Pi y_{t-1} + Bx_t = \alpha(\beta' y_{t-1} + \rho_0) + \alpha_{\perp} \gamma_0$$

4. The level data y_t and the cointegrating equations have linear trends:

$$H^*(r): \Pi y_{t-1} + Bx_t = \alpha(\beta' y_{t-1} + \rho_0 + \rho_1 t) + \alpha_{\perp} \gamma_0$$

5. The level data y_t have quadratic trends and the cointegrating equations have linear trends:

$$H(r): \Pi y_{t-1} + Bx_t = \alpha(\beta' y_{t-1} + \rho_0 + \rho_1 t) + \alpha_{\perp}(\gamma_0 + \gamma_1 t)$$

The terms associated with α_{\perp} are the deterministic terms “outside” the cointegrating relations. When a deterministic term appears both inside and outside the cointegrating relation, the decomposition is not uniquely identified. Johansen (1995) identifies the part that belongs inside the error correction term by orthogonally projecting the exogenous terms onto the α space so that α_{\perp} is the null space of α such that $\alpha' \alpha_{\perp} = 0$. EViews uses a different identification method so that the error correction term has a sample mean of zero.

More specifically, we identify the part inside the error correction term by regressing the cointegrating relations $\beta' y_t$ on a constant (and linear trend).

Exogenous Variables

The test dialog allows you to specify additional exogenous variables x_t to include in the test VAR. *The constant and linear trend should not be listed in the edit box* since they are specified using the five **Trend Specification** options. If you choose to include exogenous variables, be aware that the critical values reported by EViews *do not account* for these variables.

The most commonly added deterministic terms are seasonal dummy variables. Note, however, that if you include standard 0–1 seasonal dummy variables in the test VAR, this will affect both the mean and the trend of the level series y_t . To handle this problem, Johansen (1995, page 84) suggests using centered (orthogonalized) seasonal dummy variables, which shift the mean without contributing to the trend. Centered seasonal dummy variables for quarterly and monthly series can be generated by the commands:

```
series d_q = @seas(q) - 1/4
series d_m = @seas(m) - 1/12
```

for quarter q and month m , respectively.

Lag Intervals

You should specify the lags of the test VAR as pairs of intervals. Note that the lags are specified as *lags of the first differenced terms* used in the auxiliary regression, not in terms of the levels. For example, if you type “1 2” in the edit field, the test VAR regresses Δy_t on Δy_{t-1} , Δy_{t-2} , and any other exogenous variables that you have specified. Note that in terms of the level series y_t the largest lag is 3. To run a cointegration test with one lag in the level series, type “0 0” in the edit field.

Critical Values

By default, EViews will compute the critical values for the test using MacKinnon-Haug-Michelis (1999) p -values. You may elect instead to report the Osterwald-Lenum (1992) at the 5% and 1% levels by changing the radio button selection from **MHM** to **Osterwald-Lenum**.

Interpreting Results of a Cointegration Test

As an example, the first part of the cointegration test output for the four-variable system used by Johansen and Juselius (1990) for the Danish data is shown below.

Date: 08/16/06 Time: 11:29
 Sample (adjusted): 1974Q3 1987Q3
 Included observations: 53 after adjustments
 Trend assumption: No deterministic trend (restricted constant)
 Series: LRM LRY IBO IDE
 Lags interval (in first differences): 1 to 1

Unrestricted Cointegration Rank Test (Trace)

| Hypothesized No. of CE(s) | Eigenvalue | Trace Statistic | 0.05 Critical Value | Prob.** |
|------------------------------|------------|--------------------|------------------------|---------|
| None | 0.469677 | 52.71087 | 54.07904 | 0.0659 |
| At most 1 | 0.174241 | 19.09464 | 35.19275 | 0.7814 |
| At most 2 | 0.118083 | 8.947661 | 20.26184 | 0.7411 |
| At most 3 | 0.042249 | 2.287849 | 9.164546 | 0.7200 |

Trace test indicates no cointegration at the 0.05 level

* denotes rejection of the hypothesis at the 0.05 level

**MacKinnon-Haug-Michelis (1999) p-values

As indicated in the header of the output, the test assumes no trend in the series with a restricted intercept in the cointegration relation (second trend specification in the dialog), includes three orthogonalized seasonal dummy variables D1–D3, and uses one lag in differences (two lags in levels) which is specified as “1 1” in the edit field.

Number of Cointegrating Relations

The first part of the table reports results for testing the number of cointegrating relations. Two types of test statistics are reported. The first block reports the so-called *trace* statistics and the second block (not shown above) reports the *maximum eigenvalue* statistics. For each block, the first column is the number of cointegrating relations under the null hypothesis, the second column is the ordered eigenvalues of the Π matrix in (34.24), the third column is the test statistic, and the last two columns are the 5% and 1% critical values. The (nonstandard) critical values are taken from MacKinnon-Haug-Michelis (1999), and differ slightly from those reported in Johansen and Juselius (1990).

To determine the number of cointegrating relations *r conditional on the assumptions made about the trend*, we can proceed sequentially from $r = 0$ to $r = k - 1$ until we fail to reject. The result of this sequential testing procedure is reported at the bottom of each table block.

The trace statistic reported in the first block tests the null hypothesis of r cointegrating relations against the alternative of k cointegrating relations, where k is the number of endogenous variables, for $r = 0, 1, \dots, k - 1$. The alternative of k cointegrating relations corresponds to the case where none of the series has a unit root and a stationary VAR may

be specified in terms of the levels of all of the series. The trace statistic for the null hypothesis of r cointegrating relations is computed as:

$$LR_{\text{tr}}(r|k) = -T \sum_{i=r+1}^k \log(1 - \lambda_i) \quad (34.25)$$

where λ_i is the i -th largest eigenvalue of the Π matrix in (34.24) which is reported in the second column of the output table.

The second block of the output reports the maximum eigenvalue statistic which tests the null hypothesis of r cointegrating relations against the alternative of $r+1$ cointegrating relations. This test statistic is computed as:

$$\begin{aligned} LR_{\text{max}}(r|r+1) &= -T \log(1 - \lambda_{r+1}) \\ &= LR_{\text{tr}}(r|k) - LR_{\text{tr}}(r+1|k) \end{aligned} \quad (34.26)$$

for $r = 0, 1, \dots, k-1$.

There are a few other details to keep in mind:

- Critical values are available for up to $k = 10$ series. Also note that the critical values depend on the trend assumptions and may not be appropriate for models that contain other deterministic regressors. For example, a shift dummy variable in the test VAR implies a broken linear trend in the level series y_t .
- The trace statistic and the maximum eigenvalue statistic may yield conflicting results. For such cases, we recommend that you examine the estimated cointegrating vector and base your choice on the interpretability of the cointegrating relations; see Johansen and Juselius (1990) for an example.
- In some cases, the individual unit root tests will show that some of the series are integrated, but the cointegration test will indicate that the Π matrix has full rank ($r = k$). This apparent contradiction may be the result of low power of the cointegration tests, stemming perhaps from a small sample size or serving as an indication of specification error.

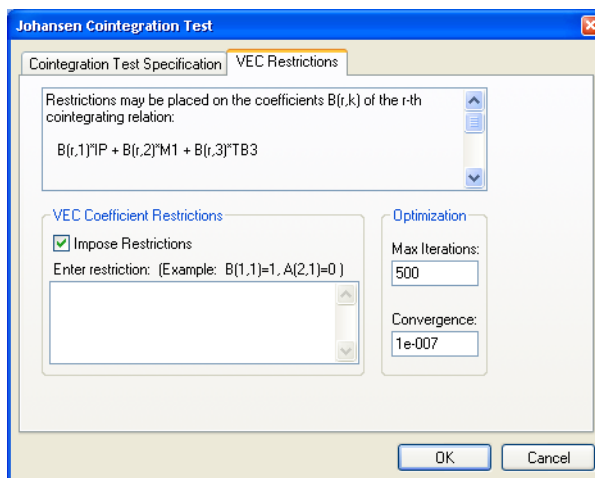
Cointegrating relations

The second part of the output provides estimates of the cointegrating relations β and the adjustment parameters α . As is well known, the cointegrating vector β is not identified unless we impose some arbitrary normalization. The first block reports estimates of β and α based on the normalization $\beta' S_{11} \beta = I$, where S_{11} is defined in Johansen (1995). Note that the *transpose* of β is reported under **Unrestricted Cointegrating Coefficients** so that the first *row* is the first cointegrating vector, the second *row* is the second cointegrating vector, and so on.

The remaining blocks report estimates from a different normalization for each possible number of cointegrating relations $r = 0, 1, \dots, k-1$. This alternative normalization expresses the first r variables as functions of the remaining $k-r$ variables in the system. Asymptotic standard errors are reported in parentheses for the parameters that are identified.

Imposing Restrictions

Since the cointegrating vector β is not identified, you may wish to impose your own identifying restrictions. Restrictions can be imposed on the cointegrating vector (elements of the β matrix) and/or on the adjustment coefficients (elements of the α matrix). To impose restrictions in a cointegration test, select **View/Cointegration Test...** and specify the options in the **Cointegration Test Specification** tab as explained above. Then bring up the **VEC Restrictions** tab. You will enter your restrictions in the edit box that appears when you check the **Impose Restrictions** box:



Restrictions on the Cointegrating Vector

To impose restrictions on the cointegrating vector β , you must refer to the (i,j) -th element of the *transpose* of the β matrix by $B(i,j)$. The i -th cointegrating relation has the representation:

$$B(i,1)*y_1 + B(i,2)*y_2 + \dots + B(i,k)*y_k$$

where y_1, y_2, \dots are the (lagged) endogenous variable. Then, if you want to impose the restriction that the coefficient on y_1 for the second cointegrating equation is 1, you would type the following in the edit box:

$$B(2,1) = 1$$

You can impose multiple restrictions by separating each restriction with a comma on the same line or typing each restriction on a separate line. For example, if you want to impose

the restriction that the coefficients on y_1 for the first and second cointegrating equations are 1, you would type:

$$B(1,1) = 1$$

$$B(2,1) = 1$$

Currently *all restrictions must be linear (or more precisely affine) in the elements of the β matrix*. So for example

$$B(1,1) * B(2,1) = 1$$

will return a syntax error.

Restrictions on the Adjustment Coefficients

To impose restrictions on the adjustment coefficients, you must refer to the (i,j) -th elements of the α matrix by $A(i,j)$. The error correction terms in the i -th VEC equation will have the representation:

$$A(i,1)*CointEq1 + A(i,2)*CointEq2 + \dots + A(i,r)*CointEq_r$$

Restrictions on the adjustment coefficients are currently limited to linear homogeneous restrictions so that you must be able to write your restriction as $R\text{vec}(\alpha) = 0$, where R is a known $qk \times r$ matrix. This condition implies, for example, that the restriction,

$$A(1,1) = A(2,1)$$

is valid but:

$$A(1,1) = 1$$

will return a restriction syntax error.

One restriction of particular interest is whether the i -th row of the α matrix is all zero. If this is the case, then the i -th endogenous variable is said to be *weakly exogenous with respect to the β parameters*. See Johansen (1995) for the definition and implications of weak exogeneity. For example, if we assume that there is only one cointegrating relation in the VEC, to test whether the second endogenous variable is weakly exogenous with respect to β you would enter:

$$A(2,1) = 0$$

To impose multiple restrictions, you may either separate each restriction with a comma on the same line or type each restriction on a separate line. For example, to test whether the second endogenous variable is weakly exogenous with respect to β in a VEC with two cointegrating relations, you can type:

$$A(2,1) = 0$$

$$A(2,2) = 0$$

You may also impose restrictions on both β and α . However, the restrictions on β and α must be *independent*. So for example,

$$A(1, 1) = 0$$

$$B(1, 1) = 1$$

is a valid restriction but:

$$A(1, 1) = B(1, 1)$$

will return a restriction syntax error.

Identifying Restrictions and Binding Restrictions

EViews will check to see whether the restrictions you provided identify all cointegrating vectors for each possible rank. The identification condition is checked numerically by the rank of the appropriate Jacobian matrix; see Boswijk (1995) for the technical details. Asymptotic standard errors for the estimated cointegrating parameters will be reported only if the restrictions identify the cointegrating vectors.

If the restrictions are binding, EViews will report the LR statistic to test the binding restrictions. The LR statistic is reported if the degrees of freedom of the asymptotic χ^2 distribution is positive. Note that the restrictions can be binding even if they are not identifying, (e.g. when you impose restrictions on the adjustment coefficients but not on the cointegrating vector).

Options for Restricted Estimation

Estimation of the restricted cointegrating vectors β and adjustment coefficients α generally involves an iterative process. The **VEC Restrictions** tab provides iteration control for the maximum number of iterations and the convergence criterion. EViews estimates the restricted β and α using the switching algorithm as described in Boswijk (1995). Each step of the algorithm is guaranteed to increase the likelihood and the algorithm should eventually converge (though convergence may be to a local rather than a global optimum). You may need to increase the number of iterations in case you are having difficulty achieving convergence at the default settings.

Results of Restricted Cointegration Test

If you impose restrictions in the **Cointegration Test** view, the output will first display the test results without the restrictions as described above. The second part of the output begins by displaying the results of the LR test for binding restrictions.

Restrictions:

$a(3,1)=0$

Tests of cointegration restrictions:

| Hypothesized No. of CE(s) | Restricted Log-likelihood | LR Statistic | Degrees of Freedom | Probability |
|------------------------------|------------------------------|-----------------|-----------------------|-------------|
| 1 | 668.6698 | 0.891088 | 1 | 0.345183 |
| 2 | 674.2964 | NA | NA | NA |
| 3 | 677.4677 | NA | NA | NA |

NA indicates restriction not binding.

If the restrictions are not binding for a particular rank, the corresponding rows will be filled with NAs. If the restrictions are binding but the algorithm did not converge, the corresponding row will be filled with an asterisk “*”. (You should redo the test by increasing the number of iterations or relaxing the convergence criterion.) For the example output displayed above, we see that the single restriction $\alpha_{31} = 0$ is binding only under the assumption that there is one cointegrating relation. *Conditional on there being only one cointegrating relation*, the LR test does not reject the imposed restriction at conventional levels.

The output also reports the estimated β and α imposing the restrictions. Since the cointegration test does not specify the number of cointegrating relations, results for all ranks that are consistent with the specified restrictions will be displayed. For example, suppose the restriction is:

$$B(2,1) = 1$$

Since this is a restriction on the second cointegrating vector, EViews will display results for ranks $r = 2, 3, \dots, k-1$ (if the VAR has only $k = 2$ variables, EViews will return an error message pointing out that the “implied rank from restrictions must be of reduced order”).

For each rank, the output reports whether convergence was achieved and the number of iterations. The output also reports whether the restrictions identify all cointegrating parameters under the assumed rank. If the cointegrating vectors are identified, asymptotic standard errors will be reported together with the parameters β .

Panel Cointegration Testing

The extensive interest in and the availability of panel data has led to an emphasis on extending various statistical tests to panel data. Recent literature has focused on tests of cointegration in a panel setting. EViews will compute one of the following types of panel

cointegration tests: Pedroni (1999), Pedroni (2004), Kao (1999) and a Fisher-type test using an underlying Johansen methodology (Maddala and Wu (1999)).

Performing Panel Cointegration Tests in EViews

You may perform a cointegration test using either a pool object or a group in a panel workfile setting. We focus here on the panel setting as this is likely to be the most commonly employed; conducting a cointegration test using a pool involves only minor differences in specification (see “Performing Cointegration Tests,” beginning on page 475).

To perform the panel cointegration test you should first make certain you are in a panel structured workfile ([Chapter 38. “Working with Panel Data,” on page 509](#)).

Next, open an EViews group containing the series of interest, and select **Views/Cointegration Test...** to display the cointegration dialog.

The combo box at the top of the dialog box allow you to choose between three types of tests: **Pedroni (Engle-Granger based)**, **Kao (Engle-Granger based)**, **Fisher (combined Johansen)**. As you select different test types, the remainder of the dialog will change to present you with different options. Here, we see the options associated with the Pedroni test.

(Note, the Pedroni test will only be available for groups containing seven or fewer series).

The customizable options associated with Pedroni and Kao tests are very similar to the options found in panel unit root testing (“[Panel Unit Root Tests](#)” on page 100).

The **Deterministic trend specification** portion of the dialog specifies the exogenous regressors to be included in the second-stage regression. You should select **Individual intercept** if you wish to include individual fixed effects, **Individual intercept and individual trend** if you wish to include both individual fixed effects and trends, or **No intercept or trend** to include no regressors. The Kao test only allows for **Individual intercept**.

The **Lag length** section is used to determine the number of lags to be included in the second stage regression. If you select **Automatic selection**, EViews will determine the optimum lag

using the information criterion specified in the combo box (**Akaike, Schwarz, Hannan-Quinn**). In addition you may provide a **Maximum lag** to be used in automatic selection. An empty field will instruct EViews to calculate the maximum lag for each cross-section based on the number of observations. The default maximum lag length for cross-section i is computed as:

$$\text{floor}[\min(12, T_i/3)(T_i/100)^{0.25}]$$

where T_i is the length of the cross-section i . Alternatively, you may provide your own value by selecting **User specified**, and entering a value in the edit field.

Both tests employ kernel estimation of the long run variance. The **Spectral estimation** portion of the dialog allows you to specify settings for this estimation. You may select from a number of kernel types (**Bartlett, Parzen, Quadratic spectral**) and specify how the bandwidth is to be selected (**Newey-West automatic** or **User specified**).

Here, we see the options for the Fisher test selection. These options are similar to the options available in the Johansen cointegration test ([“Cointegration Testing,” beginning on page 363](#)).

The **Deterministic trend specification** section determines the type of exogenous trend to be used.

The **Lag intervals** section specifies the lag-pair to be used in estimation.

The screenshot shows the 'Panel Cointegration Test' dialog box. The 'Test type' dropdown is set to 'Fisher (combined Johansen)'. The 'Deterministic trend specification' section has five radio button options: 'No trend in data', '1) No intercept or trend in CE or VAR', '2) Intercept (no trend) in CE - no intercept in VAR', '3) Intercept (no trend) in CE and VAR' (which is selected), '4) Intercept and trend in CE - no trend in VAR', and '5) Intercept and trend in CE - linear trend in VAR'. The 'Lag intervals' section has a text box containing '11' and a label 'Lag spec for differenced endogenous'. At the bottom are 'OK' and 'Cancel' buttons.

Panel Cointegration Details

Here we provide a brief description of the cointegration tests supported by EViews. The Pedroni and Kao tests are based on Engle-Granger (1987) two-step (residual-based) cointegration tests. The Fisher test is a combined Johansen test.

Pedroni (Engle-Granger based) Cointegration Tests

The Engle-Granger (1987) cointegration test is based on an examination of the residuals of a spurious regression performed using $I(1)$ variables. If the variables are cointegrated then the residuals should be $I(0)$. On the other hand if the variables are not cointegrated then the

residuals will be I(1). Pedroni (1999, 2004) and Kao (1999) extend the Engle-Granger framework to tests involving panel data.

Pedroni proposes several tests for cointegration that allow for heterogeneous intercepts and trend coefficients across cross-sections. Consider the following regression

$$y_{it} = \alpha_i + \delta_i t + \beta_{1i} x_{1i,t} + \beta_{2i} x_{2i,t} + \dots + \beta_{Mi} x_{Mi,t} + e_{i,t} \quad (34.27)$$

for $t = 1, \dots, T$; $i = 1, \dots, N$; $m = 1, \dots, M$; where y and x are assumed to be integrated of order one, *e.g.* I(1). The parameters α_i and δ_i are individual and trend effects which may be set to zero if desired.

Under the null hypothesis of no cointegration, the residuals $e_{i,t}$ will be I(1). The general approach is to obtain residuals from Equation (34.27) and then to test whether residuals are I(1) by running the auxiliary regression,

$$e_{it} = \rho_i e_{it-1} + u_{it} \quad (34.28)$$

or

$$e_{it} = \rho_i e_{it-1} + \sum_{j=1}^{p_i} \psi_{ij} \Delta e_{it-j} + v_{it} \quad (34.29)$$

for each cross-section. Pedroni describes various methods of constructing statistics for testing for null hypothesis of no cointegration ($\rho_i = 1$). There are two alternative hypotheses: the homogenous alternative, ($\rho_i = \rho$) < 1 for all i (which Pedroni terms the within-dimension test or panel statistics test), and the heterogeneous alternative, $\rho_i < 1$ for all i (also referred to as the between-dimension or group statistics test).

The Pedroni panel cointegration statistic $\mathfrak{N}_{N,T}$ is constructed from the residuals from either Equation (34.28) or Equation (34.29). A total of eleven statistics with varying degree of properties (size and power for different N and T) are generated.

Pedroni shows that the standardized statistic is asymptotically normally distributed,

$$\frac{\mathfrak{N}_{N,T} - \mu \sqrt{N}}{\sqrt{v}} \Rightarrow N(0, 1) \quad (34.30)$$

where μ and v are Monte Carlo generated adjustment terms.

Details for these calculations are provided in the original papers.

Kao (Engle-Granger based) Cointegration Tests

The Kao test follows the same basic approach as the Pedroni tests, but specifies cross-section specific intercepts and homogeneous coefficients on the first-stage regressors.

In the bivariate case described in Kao (1999), we have

$$y_{it} = \alpha_i + \beta x_{it} + e_{it} \quad (34.31)$$

for

$$y_{it} = y_{it-1} + u_{i,t} \quad (34.32)$$

$$x_{it} = x_{it-1} + \epsilon_{i,t} \quad (34.33)$$

for $t = 1, \dots, T$; $i = 1, \dots, N$. More generally, we may consider running the first stage regression Equation (34.27), requiring the α_i to be heterogeneous, β_i to be homogeneous across cross-sections, and setting all of the trend coefficients γ_i to zero.

Kao then runs either the pooled auxiliary regression,

$$e_{it} = \rho e_{it-1} + v_{it} \quad (34.34)$$

or the augmented version of the pooled specification,

$$e_{it} = \tilde{\rho} e_{it-1} + \sum_{j=1}^p \psi_j \Delta e_{it-j} + v_{it} \quad (34.35)$$

Under the null of no cointegration, Kao shows that following the statistics,

$$DF_{\rho} = \frac{T\sqrt{N}(\hat{\rho} - 1) + 3\sqrt{N}}{\sqrt{10.2}} \quad (34.36)$$

$$DF_t = \sqrt{1.25} t_{\rho} + \sqrt{1.875 N} \quad (34.37)$$

$$DF_{\rho}^* = \frac{\sqrt{N}T(\hat{\rho} - 1) + 3\sqrt{N}\hat{\sigma}_v^2/\hat{\sigma}_{0v}^2}{\sqrt{3 + 36\hat{\sigma}_v^4/(5\hat{\sigma}_{0v}^4)}} \quad (34.38)$$

$$DF_t^* = \frac{t_{\rho} + \sqrt{6N}\hat{\sigma}_v/(2\hat{\sigma}_{0v})}{\sqrt{\hat{\sigma}_{0v}^2/(2\hat{\sigma}_v^2) + 3\hat{\sigma}_v^2/(10\hat{\sigma}_{0v}^2)}} \quad (34.39)$$

and for $p > 0$ (i.e. the augmented version),

$$ADF = \frac{t_{\rho} + \sqrt{6N}\hat{\sigma}_v/(2\hat{\sigma}_{0u})}{\sqrt{\hat{\sigma}_{0v}^2/(2\hat{\sigma}_v^2) + 3\hat{\sigma}_v^2/(10\hat{\sigma}_{0v}^2)}} \quad (34.40)$$

converge to $N(0, 1)$ asymptotically, where the estimated variance is $\hat{\sigma}_v^2 = \hat{\sigma}_u^2 - \hat{\sigma}_{u\epsilon}^2 \sigma_{\epsilon}^{-2}$ with estimated long run variance $\hat{\sigma}_{0v}^2 = \hat{\sigma}_{0u}^2 - \hat{\sigma}_{0u\epsilon}^2 \sigma_{0\epsilon}^{-2}$.

The covariance of

$$w_{it} = \begin{bmatrix} u_{it} \\ \epsilon_{it} \end{bmatrix} \quad (34.41)$$

is estimated as

$$\hat{\Sigma} = \begin{bmatrix} \hat{\sigma}_u^2 & \hat{\sigma}_{u\epsilon} \\ \hat{\sigma}_{u\epsilon} & \hat{\sigma}_\epsilon^2 \end{bmatrix} = \frac{1}{NT} \sum_{i=1}^N \sum_{t=1}^T \hat{w}_{it} \hat{w}'_{it} \quad (34.42)$$

and the long run covariance is estimated as

$$\hat{\Omega} = \begin{bmatrix} \hat{\sigma}_{0u}^2 & \hat{\sigma}_{0u\epsilon} \\ \hat{\sigma}_{0u\epsilon} & \hat{\sigma}_{0\epsilon}^2 \end{bmatrix} = \frac{1}{N} \sum_{i=1}^N \left[\frac{1}{T} \sum_{t=1}^T \hat{w}_{it} \hat{w}'_{it} + \kappa(\hat{w}_i) \right] \quad (34.43)$$

where κ is any kernel function.

Combined Individual Tests (Fisher/Johansen)

Fisher (1932) derives a combined test that uses the results of the individual independent tests. Maddala and Wu (1999) use Fisher's result to propose an alternative approach to testing for cointegration in panel data by combining tests from individual cross-sections to obtain a test statistic for the full panel.

If π_i is the p -value from an individual cointegration test for cross-section i , then under the null hypothesis for the panel,

$$-2 \sum_{i=1}^N \log(\pi_i) \rightarrow \chi^2_{2N} \quad (34.44)$$

By default, EViews reports the χ^2 value based on MacKinnon-Haug-Michelis (1999) p -values for Johansen's cointegration trace test and maximum eigenvalue test.

Vector Error Correction (VEC) Models

A vector error correction (VEC) model is a restricted VAR designed for use with nonstationary series that are known to be cointegrated. The VEC has cointegration relations built into the specification so that it restricts the long-run behavior of the endogenous variables to converge to their cointegrating relationships while allowing for short-run adjustment dynamics. The cointegration term is known as the *error correction* term since the deviation from long-run equilibrium is corrected gradually through a series of partial short-run adjustments.

To take the simplest possible example, consider a two variable system with one cointegrating equation and no lagged difference terms. The cointegrating equation is:

$$y_{2,t} = \beta y_{1,t} \quad (34.45)$$

The corresponding VEC model is:

$$\begin{aligned}\Delta y_{1,t} &= \alpha_1(y_{2,t-1} - \beta y_{1,t-1}) + \epsilon_{1,t} \\ \Delta y_{2,t} &= \alpha_2(y_{2,t-1} - \beta y_{1,t-1}) + \epsilon_{2,t}\end{aligned}\tag{34.46}$$

In this simple model, the only right-hand side variable is the error correction term. In long run equilibrium, this term is zero. However, if y_1 and y_2 deviate from the long run equilibrium, the error correction term will be nonzero and each variable adjusts to partially restore the equilibrium relation. The coefficient α_i measures the speed of adjustment of the i -th endogenous variable towards the equilibrium.

How to Estimate a VEC

As the VEC specification only applies to cointegrated series, you should first run the Johansen cointegration test as described above and determine the number of cointegrating relations. You will need to provide this information as part of the VEC specification.

To set up a VEC, click the **Estimate** button in the VAR toolbar and choose the **Vector Error Correction** specification from the **VAR/VEC Specification** tab. In the **VAR/VEC Specification** tab, you should provide the same information as for an unrestricted VAR, except that:

- The constant or linear trend term should *not* be included in the **Exogenous Series** edit box. The constant and trend specification for VECs should be specified in the **Cointegration** tab (see below).
- The lag interval specification refers to *lags of the first difference terms* in the VEC. For example, the lag specification “1 1” will include lagged first difference terms on the right-hand side of the VEC. Rewritten in levels, this VEC is a restricted VAR with two lags. To estimate a VEC with no lagged first difference terms, specify the lag as “0 0”.
- The constant and trend specification for VECs should be specified in the **Cointegration** tab. You must choose from one of the five trend specifications as explained in [“Deterministic Trend Specification” on page 364](#). You must also specify the number of cointegrating relations in the appropriate edit field. This number should be a positive integer less than the number of endogenous variables in the VEC.
- If you want to impose restrictions on the cointegrating relations and/or the adjustment coefficients, use the **Restrictions** tab. [“Restrictions on the Cointegrating Vector” on page 369](#) describes this restriction in greater detail. Note that this tab is grayed out unless you have clicked the **Vector Error Correction** specification in the **VAR/VEC Specification** tab.

Once you have filled the dialog, simply click **OK** to estimate the VEC. Estimation of a VEC model is carried out in two steps. In the first step, we estimate the cointegrating relations from the Johansen procedure as used in the cointegration test. We then construct the error correction terms from the estimated cointegrating relations and estimate a VAR in first differences including the error correction terms as regressors.

VEC Estimation Output

The VEC estimation output consists of two parts. The first part reports the results from the first step Johansen procedure. If you did not impose restrictions, EViews will use a default normalization that identifies all cointegrating relations. This default normalization expresses the first r variables in the VEC as functions of the remaining $k - r$ variables, where r is the number of cointegrating relations and k is the number of endogenous variables. Asymptotic standard errors (corrected for degrees of freedom) are reported for parameters that are identified under the restrictions. If you provided your own restrictions, standard errors will not be reported unless the restrictions identify all cointegrating vectors.

The second part of the output reports results from the second step VAR in first differences, including the error correction terms estimated from the first step. The error correction terms are denoted `CointEq1`, `CointEq2`, and so on in the output. This part of the output has the same format as the output from unrestricted VARs as explained in [“VAR Estimation Output” on page 346](#), with one difference. At the bottom of the VEC output table, you will see two log likelihood values reported for the system. The first value, labeled **Log Likelihood (d.f. adjusted)**, is computed using the determinant of the residual covariance matrix (reported as **Determinant Residual Covariance**), using small sample degrees of freedom correction as in [\(34.3\)](#). This is the log likelihood value reported for unrestricted VARs. The **Log Likelihood** value is computed using the residual covariance matrix without correcting for degrees of freedom. This log likelihood value is comparable to the one reported in the cointegration test output.

Views and Procs of a VEC

Views and procs available for VECs are mostly the same as those available for VARs as explained above. Here, we only mention those that are specific to VECs.

Cointegrating Relations

View/Cointegration Graph displays a graph of the estimated cointegrating relations as used in the VEC. To store these estimated cointegrating relations as named series in the workfile, use **Proc/Make Cointegration Group**. This proc will create and display an untitled group object containing the estimated cointegrating relations as named series. These series are named `COINTEQ01`, `COINTEQ02` and so on.

Forecasting

Currently forecasts from a VAR or VEC are not available from the VAR object. Forecasts can be obtained by solving a model created from the estimated VAR/VEC. Click on **Proc/Make Model** from the VAR window toolbar to create a model object from the estimated VAR/VEC. You may then make any changes to the model specification, including modifying the `ASSIGN` statement before solving the model to obtain the forecasts. See [Chapter 36. “Models,” on page 407](#), for further discussion on how to forecast from model objects in EViews.

Data Members

Various results from the estimated VAR/VEC can be retrieved through the command line data members. “[Var Data Members](#)” on page 544 of the *Command Reference* provides a complete list of data members that are available for a VAR object. Here, we focus on retrieving the estimated coefficients of a VAR/VEC.

Obtaining Coefficients of a VAR

Coefficients of (unrestricted) VARs can be accessed by referring to elements of a *two dimensional array* C. The first dimension of C refers to the equation number of the VAR, while the second dimension refers to the variable number in each equation. For example, C(2,3) is the coefficient of the third regressor in the second equation of the VAR. The C(2,3) coefficient of a VAR named VAR01 can then be accessed by the command

```
var01.c(2,3)
```

To examine the correspondence between each element of C and the estimated coefficients, select **View/Representations** from the VAR toolbar.

Obtaining Coefficients of a VEC

For VEC models, the estimated coefficients are stored in three different two dimensional arrays: A, B, and C. A contains the adjustment parameters α , B contains the cointegrating vectors β' , and C holds the short-run parameters (the coefficients on the lagged first difference terms).

- The first index of A is the equation number of the VEC, while the second index is the number of the cointegrating equation. For example, A(2,1) is the adjustment coefficient of the first cointegrating equation in the second equation of the VEC.
- The first index of B is the number of the cointegrating equation, while the second index is the variable number in the cointegrating equation. For example, B(2,1) is the coefficient of the first variable in the second cointegrating equation. Note that this indexing scheme corresponds to the *transpose* of β .
- The first index of C is the equation number of the VEC, while the second index is the variable number of the first differenced regressor of the VEC. For example, C(2, 1) is the coefficient of the first differenced regressor in the second equation of the VEC.

You can access each element of these coefficients by referring to the name of the VEC followed by a dot and coefficient element:

```
var01.a(2,1)
var01.b(2,1)
var01.c(2,1)
```

To see the correspondence between each element of A, B, and C and the estimated coefficients, select **View/Representations** from the VAR toolbar.

A Note on Version Compatibility

The following changes made in Version 4 may yield VAR results that do not match those reported from previous versions of EViews:

- The estimated residual covariance matrix is now computed using the finite sample adjustment so the sum-of-squares is divided by $T - p$ where p is the number of estimated coefficients in each VAR equation. Previous versions of EViews divided the sum-of-squares by T .
- The standard errors for the cointegrating vector are now computed using the more general formula in Boswijk (1995), which also covers the restricted case.

References

- Amisano, Gianni and Carlo Giannini (1997). *Topics in Structural VAR Econometrics*, 2nd ed, Berlin: Springer-Verlag.
- Blanchard, Olivier and Danny Quah (1989). “The Dynamic Effects of Aggregate Demand and Aggregate Supply Disturbances,” *American Economic Review*, 79, 655-673.
- Boswijk, H. Peter (1995). “Identifiability of Cointegrated Systems,” Technical Report, Tinbergen Institute.
- Christiano, L. J., M. Eichenbaum, C. L. Evans (1999). “Monetary Policy Shocks: What Have We Learned and to What End?” Chapter 2 in J. B. Taylor and M. Woodford, (eds.), *Handbook of Macroeconomics, Volume 1A*, Amsterdam: Elsevier Science Publishers B.V.
- Dickey, D.A. and W.A. Fuller (1979). “Distribution of the Estimators for Autoregressive Time Series with a Unit Root,” *Journal of the American Statistical Association*, 74, 427-431.
- Doornik, Jurgen A. (1995). “Testing General Restrictions on the Cointegrating Space,” manuscript.
- Doornik, Jurgen A. and Henrik Hansen (1994). “An Omnibus Test for Univariate and Multivariate Normality,” manuscript.
- Engle, Robert F. and C. W. J. Granger (1987). “Co-integration and Error Correction: Representation, Estimation, and Testing,” *Econometrica*, 55, 251-276.
- Fisher, R. A. (1932). *Statistical Methods for Research Workers*, 4th Edition, Edinburgh: Oliver & Boyd.
- Johansen, Søren (1991). “Estimation and Hypothesis Testing of Cointegration Vectors in Gaussian Vector Autoregressive Models,” *Econometrica*, 59, 1551-1580.
- Johansen, Søren (1995). *Likelihood-based Inference in Cointegrated Vector Autoregressive Models*, Oxford: Oxford University Press.
- Johansen, Søren and Katarina Juselius (1990). “Maximum Likelihood Estimation and Inferences on Cointegration—with applications to the demand for money,” *Oxford Bulletin of Economics and Statistics*, 52, 169-210.
- Kao, C. (1999). “Spurious Regression and Residual-Based Tests for Cointegration in Panel Data,” *Journal of Econometrics*, 90, 1-44.

- Kelejian, H. H. (1982). "An Extension of a Standard Test for Heteroskedasticity to a Systems Framework," *Journal of Econometrics*, 20, 325-333.
- Lütkepohl, Helmut (1991). *Introduction to Multiple Time Series Analysis*, New York: Springer-Verlag.
- Maddala, G. S. and S. Wu (1999). "A Comparative Study of Unit Root Tests with Panel Data and A New Simple Test," *Oxford Bulletin of Economics and Statistics*, 61, 631-52.
- MacKinnon, James G., Alfred A. Haug, and Leo Michelis (1999), "Numerical Distribution Functions of Likelihood Ratio Tests for Cointegration," *Journal of Applied Econometrics*, 14, 563-577.
- Newey, Whitney and Kenneth West (1994). "Automatic Lag Selection in Covariance Matrix Estimation," *Review of Economic Studies*, 61, 631-653.
- Osterwald-Lenum, Michael (1992). "A Note with Quantiles of the Asymptotic Distribution of the Maximum Likelihood Cointegration Rank Test Statistics," *Oxford Bulletin of Economics and Statistics*, 54, 461-472.
- Pedroni, P. (1999). "Critical Values for Cointegration Tests in Heterogeneous Panels with Multiple Regressors," *Oxford Bulletin of Economics and Statistics*, 61, 653-70.
- Pedroni, P. (2004). "Panel Cointegration; Asymptotic and Finite Sample Properties of Pooled Time Series Tests with an Application to the PPP Hypothesis," *Econometric Theory*, 20, 597-625.
- Pesaran, M. Hashem and Yongcheol Shin (1998). "Impulse Response Analysis in Linear Multivariate Models," *Economics Letters*, 58, 17-29.
- Phillips, P.C.B. and P. Perron (1988). "Testing for a Unit Root in Time Series Regression," *Biometrika*, 75, 335-346.
- Said, Said E. and David A. Dickey (1984). "Testing for Unit Roots in Autoregressive Moving Average Models of Unknown Order," *Biometrika*, 71, 599-607.
- Sims, Chris (1980). "Macroeconomics and Reality," *Econometrica*, 48, 1-48.
- Urzua, Carlos M. (1997). "Omnibus Tests for Multivariate Normality Based on a Class of Maximum Entropy Distributions," in *Advances in Econometrics*, Volume 12, Greenwich, Conn.: JAI Press, 341-358.
- White, Halbert (1980). "A Heteroskedasticity-Consistent Covariance Matrix and a Direct Test for Heteroskedasticity," *Econometrica*, 48, 817-838.

Chapter 35. State Space Models and the Kalman Filter

The EViews `sspace` (state space) object provides a straightforward, easy-to-use interface for specifying, estimating, and working with the results of your single or multiple equation dynamic system. EViews provides a wide range of specification, filtering, smoothing, and other forecasting tools which aid you in working with dynamic systems specified in state space form.

A wide range of time series models, including the classical linear regression model and ARIMA models, can be written and estimated as special cases of a state space specification. State space models have been applied in the econometrics literature to model unobserved variables: (rational) expectations, measurement errors, missing observations, permanent income, unobserved components (cycles and trends), and the non-accelerating rate of unemployment. Extensive surveys of applications of state space models in econometrics can be found in Hamilton (1994a, Chapter 13; 1994b) and Harvey (1989, Chapters 3, 4).

There are two main benefits to representing a dynamic system in state space form. First, the state space allows unobserved variables (known as the state variables) to be incorporated into, and estimated along with, the observable model. Second, state space models can be analyzed using a powerful recursive algorithm known as the Kalman (Bucy) filter. The Kalman filter algorithm has been used, among other things, to compute exact, finite sample forecasts for Gaussian ARMA models, multivariate (vector) ARMA models, MIMIC (multiple indicators and multiple causes), Markov switching models, and time varying (random) coefficient models.

Those of you who have used early versions of the `sspace` object will note that much was changed with the EViews 4 release. We strongly recommend that you read [“Converting from Version 3 Sspace” on page 404](#) before loading existing workfiles and before beginning to work with the new state space routines.

Background

We present here a very brief discussion of the specification and estimation of a linear state space model. Those desiring greater detail are directed to Harvey (1989), Hamilton (1994a, Chapter 13; 1994b), and especially the excellent treatment of Koopman, Shephard and Doornik (1999).

Specification

A linear state space representation of the dynamics of the $n \times 1$ vector y_t is given by the system of equations:

$$y_t = c_t + Z_t \alpha_t + \epsilon_t \quad (35.1)$$

$$\alpha_{t+1} = d_t + T_t \alpha_t + v_t \quad (35.2)$$

where α_t is an $m \times 1$ vector of possibly unobserved state variables, where c_t , Z_t , d_t and T_t are conformable vectors and matrices, and where ϵ_t and v_t are vectors of mean zero, Gaussian disturbances. Note that the unobserved state vector is assumed to move over time as a first-order vector autoregression.

We will refer to the first set of equations as the “signal” or “observation” equations and the second set as the “state” or “transition” equations. The disturbance vectors ϵ_t and v_t are assumed to be serially independent, with contemporaneous variance structure:

$$\Omega_t = \text{var} \begin{bmatrix} \epsilon_t \\ v_t \end{bmatrix} = \begin{bmatrix} H_t & G_t \\ G_t' & Q_t \end{bmatrix} \quad (35.3)$$

where H_t is an $n \times n$ symmetric variance matrix, Q_t is an $m \times m$ symmetric variance matrix, and G_t is an $n \times m$ matrix of covariances.

In the discussion that follows, we will generalize the specification given in (35.1)—(35.3) by allowing the system matrices and vectors $\Xi_t \equiv \{c_t, d_t, Z_t, T_t, H_t, Q_t, G_t\}$ to depend upon observable explanatory variables X_t and unobservable parameters θ . Estimation of the parameters θ is discussed in “Estimation,” beginning on page 387.

Filtering

Consider the conditional distribution of the state vector α_t given information available at time s . We can define the mean and variance matrix of the conditional distribution as:

$$a_{t|s} \equiv E_s(\alpha_t) \quad (35.4)$$

$$P_{t|s} \equiv E_s[(\alpha_t - a_{t|s})(\alpha_t - a_{t|s})'] \quad (35.5)$$

where the subscript below the expectation operator indicates that expectations are taken using the conditional distribution for that period.

One important conditional distribution is obtained by setting $s = t - 1$, so that we obtain the *one-step ahead mean* $a_{t|t-1}$ and *one-step ahead variance* $P_{t|t-1}$ of the states α_t . Under the Gaussian error assumption, $a_{t|t-1}$ is also the minimum mean square error estimator of α_t and $P_{t|t-1}$ is the mean square error (MSE) of $a_{t|t-1}$. If the normality assumption is dropped, $a_{t|t-1}$ is still the minimum mean square *linear* estimator of α_t .

Given the one-step ahead state conditional mean, we can also form the (linear) minimum MSE *one-step ahead estimate* of y_t :

$$\tilde{y}_t = y_{t|t-1} \equiv E_{t-1}(y_t) = E(y_t | a_{t|t-1}) = c_t + Z_t a_{t|t-1} \quad (35.6)$$

The *one-step ahead prediction error* is given by,

$$\tilde{\epsilon}_t = \epsilon_{t|t-1} \equiv y_t - \tilde{y}_{t|t-1} \quad (35.7)$$

and the *prediction error variance* is defined as:

$$\tilde{F}_t = F_{t|t-1} \equiv \text{var}(\epsilon_{t|t-1}) = Z_t P_{t|t-1} Z_t' + H_t \quad (35.8)$$

The Kalman (Bucy) filter is a recursive algorithm for sequentially updating the one-step ahead estimate of the state mean and variance given new information. Details on the recursion are provided in the references above. For our purposes, it is sufficient to note that given initial values for the state mean and covariance, values for the system matrices \mathbf{Z}_t , and observations on y_t , the Kalman filter may be used to compute one-step ahead estimates of the state and the associated mean square error matrix, $\{a_{t|t-1}, P_{t|t-1}\}$, the contemporaneous or *filtered* state mean and variance, $\{a_t, P_t\}$, and the one-step ahead prediction, prediction error, and prediction error variance, $\{y_{t|t-1}, \epsilon_{t|t-1}, F_{t|t-1}\}$. Note that we may also obtain the standardized prediction residual, $e_{t|t-1}$, by dividing $\epsilon_{t|t-1}$ by the square-root of the corresponding diagonal element of $F_{t|t-1}$.

Fixed-Interval Smoothing

Suppose that we observe the sequence of data up to time period T . The process of using this information to form expectations at any time period up to T is known as *fixed-interval smoothing*. Despite the fact that there are a variety of other distinct forms of smoothing (e.g., fixed-point, fixed-lag), we will use the term *smoothing* to refer to fixed-interval smoothing.

Additional details on the smoothing procedure are provided in the references given above. For now, note that smoothing uses all of the information in the sample to provide *smoothed estimates of the states*, $\hat{\alpha}_t \equiv a_{t|T} \equiv E_T(\alpha_t)$, and *smoothed estimates of the state variances*, $V_t \equiv \text{var}_T(\alpha_t)$. The matrix V_t may also be interpreted as the MSE of the smoothed state estimate $\hat{\alpha}_t$.

As with the one-step ahead states and variances above, we may use the smoothed values to form *smoothed estimates of the signal variables*,

$$\hat{y}_t \equiv E(y_t | \hat{\alpha}_t) = c_t + Z_t \hat{\alpha}_t \quad (35.9)$$

and to compute the *variance of the smoothed signal estimates*:

$$S_t \equiv \text{var}(\hat{y}_{t|T}) = Z_t V_t Z_t'. \quad (35.10)$$

Lastly, the smoothing procedure allows us to compute *smoothed disturbance estimates*, $\hat{\epsilon}_t \equiv \epsilon_{t|T} \equiv E_T(\epsilon_t)$ and $\hat{v}_t \equiv v_{t|T} \equiv E_T(v_t)$, and a corresponding *smoothed disturbance variance matrix*:

$$\hat{\Omega}_t = \text{var}_T \left(\begin{bmatrix} \epsilon_t \\ v_t \end{bmatrix} \right) \quad (35.11)$$

Dividing the smoothed disturbance estimates by the square roots of the corresponding diagonal elements of the smoothed variance matrix yields the *standardized smoothed disturbance estimates* \hat{e}_t and \hat{v}_t .

Forecasting

There are a variety of types of forecasting which may be performed with state space models. These methods differ primarily in what and how information is used. We will focus on the three methods that are supported by EViews built-in forecasting routines.

n-Step Ahead Forecasting

Earlier, we examined the notion of one-step ahead prediction. Consider now the notion of multi-step ahead prediction of observations, in which we take a fixed set of information available at a given period, and forecast several periods ahead. Modifying slightly the expressions in (35.4)—(35.8) yields the *n-step ahead state conditional mean and variance*:

$$a_{t+n|t} \equiv E_t(\alpha_{t+n}), \quad (35.12)$$

$$P_{t+n|t} \equiv E_t[(\alpha_{t+n} - a_{t+n|t})(\alpha_{t+n} - a_{t+n|t})'] \quad (35.13)$$

the *n-step ahead forecast*,

$$y_{t+n|t} \equiv E_t(y_{t+n}) = c_t + Z_t a_{t+n|t} \quad (35.14)$$

and the corresponding *n-step ahead forecast MSE matrix*:

$$F_{t+n|t} \equiv \text{MSE}(\tilde{y}_{t+n|t}) = Z_{t+n} P_{t+n|t} Z_{t+n}' + H_t \quad (35.15)$$

for $n = 1, 2, \dots$. As before, $a_{t+n|t}$ may also be interpreted as the minimum MSE estimate of α_{t+n} based on the information set available at time t , and $P_{t+n|t}$ is the MSE of the estimate.

It is worth emphasizing that the definitions given above for the forecast MSE matrices $F_{t+n|t}$ do not account for extra variability introduced in the estimation of any unknown parameters θ . In this setting, the $F_{t+n|t}$ will understate the true variability of the forecast, and should be viewed as being computed conditional on the specific value of the estimated parameters.

It is also worth noting that the *n-step ahead forecasts* may be computed using a slightly modified version of the basic Kalman recursion (Harvey 1989). To forecast at period $s = t + n$, simply initialize a Kalman filter at time $t + 1$ with the values of the predicted states and state covariances using information at time t , and run the filter forward $n - 1$ additional periods using no additional signal information. This procedure is repeated for each observation in the forecast sample, $s = t + 1, \dots, t + n^*$.

Dynamic Forecasting

The concept of *dynamic forecasting* should be familiar to you from other EViews estimation objects. In dynamic forecasting, we start at the beginning of the forecast sample t , and compute a complete set of *n-period ahead forecasts* for each period $n = 1, \dots, n^*$ in the forecast interval. Thus, if we wish to start at period t and forecast dynamically to $t + n^*$, we

would compute a one-step ahead forecast for $t + 1$, a two-step ahead forecast for $t + 2$, and so forth, up to an n^* -step ahead forecast for $t + n^*$. It may be useful to note that as with n -step ahead forecasting, we simply initialize a Kalman filter at time $t + 1$ and run the filter forward additional periods using no additional signal information. For dynamic forecasting, however, only one n -step ahead forecast is required to compute all of the forecast values since the information set is not updated from the beginning of the forecast period.

Smoothed Forecasting

Alternatively, we can compute *smoothed forecasts* which use all available signal data over the forecast sample (for example, $a_{t+n|t+n^*}$). These forward looking forecasts may be computed by initializing the states at the start of the forecast period, and performing a Kalman smooth over the entire forecast period using all relevant signal data. This technique is useful in settings where information on the entire path of the signals is used to interpolate values throughout the forecast sample.

We make one final comment about the forecasting methods described above. For traditional n -step ahead and dynamic forecasting, the states are typically initialized using the one-step ahead forecasts of the states and variances at the start of the forecast window. For smoothed forecasts, one would generally initialize the forecasts using the corresponding smoothed values of states and variances. There may, however, be situations where you wish to choose a different set of initial values for the forecast filter or smoother. The EViews forecasting routines (described in [“State Space Procedures,” beginning on page 401](#)) provide you with considerable control over these initial settings. Be aware, however, that the interpretation of the forecasts in terms of the available information will change if you choose alternative settings.

Estimation

To implement the Kalman filter and the fixed-interval smoother, we must first replace any unknown elements of the system matrices by their estimates. Under the assumption that the ϵ_t and v_t are Gaussian, the sample log likelihood:

$$\log L(\theta) = -\frac{nT}{2} \log 2\pi - \frac{1}{2} \sum_t \log |\tilde{F}_t(\theta)| - \frac{1}{2} \sum_t \tilde{\epsilon}_t'(\theta) \tilde{F}_t(\theta)^{-1} \tilde{\epsilon}_t(\theta) \quad (35.16)$$

may be evaluated using the Kalman filter. Using numeric derivatives, standard iterative techniques may be employed to maximize the likelihood with respect to the unknown parameters θ (see [Appendix D. “Estimation and Solution Options,” on page 629](#)).

Initial Conditions

Evaluation of the Kalman filter, smoother, and forecasting procedures all require that we provide the initial one-step ahead predicted values for the states $\alpha_{1|0}$ and variance matrix $P_{1|0}$. With some stationary models, steady-state conditions allow us to use the system matrices to solve for the values of $\alpha_{1|0}$ and $P_{1|0}$. In other cases, we may have preliminary

estimates of $\alpha_{1|0}$, along with measures of uncertainty about those estimates. But in many cases, we may have no information, or *diffuse priors*, about the initial conditions.

Specifying a State Space Model in EViews

EViews handles a wide range of single and multiple-equation state space models, providing you with detailed control over the specification of your system equations, covariance matrices, and initial conditions.

The first step in specifying and estimating a state space model is to create a state space object. Select **Object/New Object.../Sspace** from the main toolbar or type `sspace` in the command window. EViews will create a state space object and open an empty state space specification window.

There are two ways to specify your state space model. The easiest is to use EViews' special "auto-specification" features to guide you in creating some of the standard forms for these models. Simply select **Proc/Define State Space...** from the `sspace` object menu. Specialized dialogs will open to guide you through the specification process. We will describe this method in greater detail in ["Auto-Specification" on page 395](#).

The more general method of describing your state space model uses keywords and text to describe the signal equations, state equations, error structure, initial conditions, and if desired, parameter starting values for estimation. The next section describes the general syntax for the state space object.

Specification Syntax

State Equations

A state equation contains the "@STATE" keyword followed by a valid state equation specification. Bear in mind that:

- Each equation must have a unique dependent variable name; expressions are not allowed. Since EViews does not automatically create workfile series for the states, you may use the name of an existing (non-series) EViews object.
- State equations may not contain signal equation dependent variables, or leads or lags of these variables.
- Each state equation must be linear in the one-period lag of the states. Nonlinearities in the states, or the presence of contemporaneous, lead, or multi-period lag states will generate an error message. We emphasize the point that the one-period lag restriction on states *is not restrictive* since higher order lags may be written as new state variables. An example of this technique is provided in the example ["ARMAX\(2, 3\) with a Random Coefficient" on page 392](#).

- State equations may contain exogenous variables and unknown coefficients, and may be nonlinear in these elements.

In addition, state equations may contain an optional error or error variance specification. If there is no error or error variance, the state equation is assumed to be deterministic. Specification of the error structure of state space models is described in greater detail in [“Errors and Variances” on page 390](#).

Examples

The following two state equations define an unobserved error with an AR(2) process:

```
@state sv1 = c(2)*sv1(-1) + c(3)*sv2(-1) + [var = exp(c(5))]  
@state sv2 = sv1(-1)
```

The first equation parameterizes the AR(2) for SV1 in terms of an AR(1) coefficient, C(2), and an AR(2) coefficient, C(3). The error variance specification is given in square brackets. Note that the state equation for SV2 defines the lag of SV1 so that SV2(-1) is the two period lag of SV1.

Similarly, the following are valid state equations:

```
@state sv1 = sv1(-1) + [var = exp(c(3))]  
@state sv2 = c(1) + c(2)*sv2(-1) + [var = exp(c(3))]  
@state sv3 = c(1) + exp(c(3)*x/z) + c(2)*sv3(-1) + [var =  
exp(c(3))]
```

describing a random walk, and an AR(1) with drift (without/with exogenous variables).

The following are *not* valid state equations:

```
@state exp(sv1) = sv1(-1) + [var = exp(c(3))]  
@state sv2 = log(sv2(-1)) + [var = exp(c(3))]  
@state sv3 = c(1) + c(2)*sv3(-2) + [var=exp(c(3))]
```

since they violate at least one of the conditions described above (in order: expression for dependent state variable, nonlinear in state, multi-period lag of state variables).

Observation/Signal Equations

By default, if an equation specification is not specifically identified as a state equation using the “@STATE” keyword, it will be treated by EViews as an observation or signal equation. Signal equations may also be identified explicitly by the keyword “@SIGNAL”. There are some aspects of signal equation specification to keep in mind:

- Signal equation dependent variables may involve expressions.
- Signal equations may not contain current values or leads of signal variables. You should be aware that any lagged signals are treated as predetermined for purposes of

multi-step ahead forecasting (for discussion and alternative specifications, see Harvey 1989, p. 367-368).

- Signal equations must be linear in the contemporaneous states. Nonlinearities in the states, or the presence of leads or lags of states will generate an error message. Again, the restriction that there are no state lags is not restrictive since additional deterministic states may be created to represent the lagged values of the states.
- Signal equations may have exogenous variables and unknown coefficients, and may be nonlinear in these elements.

Signal equations may also contain an optional error or error variance specification. If there is no error or error variance, the equation is assumed to be deterministic. Specification of the error structure of state space models is described in greater detail in [“Errors and Variances” on page 390](#).

Examples

The following are valid signal equation specifications:

```
log(passenger) = c(1) + c(3)*x + sv1 + c(4)*sv2
@signal y = sv1 + sv2*x1 + sv3*x2 + sv4*y(-1) + [var=exp(c(1))]
z = sv1 + sv2*x1 + sv3*x2 + c(1) + [var=exp(c(2))]
```

The following are invalid equations:

```
log(passenger) = c(1) + c(3)*x + sv1(-1)
@signal y = sv1*sv2*x1 + [var = exp(c(1))]
z = sv1 + sv2*x1 + z(1) + c(1) + [var = exp(c(2))]
```

since they violate at least one of the conditions described above (in order: lag of state variable, nonlinear in a state variable, lead of signal variable).

Errors and Variances

While EViews always adds an implicit error term to each equation in an equation or system object, the handling of error terms differs in a sspace object. In a sspace object, the equation specifications in a signal or state equation do not contain error terms unless specified explicitly.

The easiest way to add an error to a state space equation is to specify an implied error term using its variance. You can simply add an error variance expression, consisting of the keyword “VAR” followed by an assignment statement (all enclosed in square brackets), to the existing equation:

```
@signal y = c(1) + sv1 + sv2 + [var = 1]
@state sv1 = sv1(-1) + [var = exp(c(2))]
```

```
@state sv2 = c(3) + c(4)*sv2(-1) + [var = exp(c(2)*x)]
```

The specified variance may be a known constant value, or it can be an expression containing unknown parameters to be estimated. You may also build time-variation into the variances using a series expression. Variance expressions may not, however, contain state or signal variables.

While straightforward, this direct variance specification method does not admit correlation between errors in different equations (by default, EViews assumes that the covariance between error terms is 0). If you require a more flexible variance structure, you will need to use the “named error” approach to define named errors with variances and covariances, and then to use these named errors as parts of expressions in the signal and state equations.

The first step of this general approach is to define your named errors. You may declare a named error by including a line with the keyword “@ENAME” followed by the name of the error:

```
@ename e1
@ename e2
```

Once declared, a named error may enter linearly into state and signal equations. In this manner, one can build correlation between the equation errors. For example, the errors in the state and signal equations in the sspace specification:

```
y = c(1) + sv1*x1 + e1
@state sv1 = sv1(-1) + e2 + c(2)*e1
@ename e1
@ename e2
```

are, in general, correlated since the named error E1 appears in both equations.

In the special case where a named error is the only error in a given equation, you can both declare and use the named residual by adding an error expression consisting of the keyword “ENAME” followed by an assignment and a name identifier:

```
y = c(1) + sv1*x1 + [ename = e1]
@state sv1 = sv1(-1) + [ename = e2]
```

The final step in building a general error structure is to define the variances and covariances associated with your named errors. You should include a sspace line comprised of the keyword “@EVAR” followed by an assignment statement for the variance of the error or the covariance between two errors:

```
@evar cov(e1, e2) = c(2)
@evar var(e1) = exp(c(3))
@evar var(e2) = exp(c(4))*x
```

The syntax for the `@EVAR` assignment statements should be self-explanatory. Simply indicate whether the term is a variance or covariance, identify the error(s), and enter the specification for the variance or covariance. There should be a separate line for each named error covariance or variance that you wish to specify. If an error term is named, but there are no corresponding “VAR = ” or `@EVAR` specifications, the missing variance or covariance specifications will remain at the default values of “NA” and “0”, respectively.

As you might expect, in the special case where an equation contains a single error term, you may combine the named error and direct variance assignment statements:

```
@state sv1 = sv1(-1) + [ename = e1, var = exp(c(3))]  
@state sv2 = sv2(-1) + [ename = e2, var = exp(c(4))]  
@evar cov(e1, e2) = c(5)
```

Specification Examples

ARMAX(2, 3) with a Random Coefficient

We can use the syntax described above to define an ARMAX(2,3) with a random coefficient for the regression variable X:

```
y = c(1) + sv5*x + sv1 + c(4)*sv2 + c(5)*sv3 + c(6)*sv4  
@state sv1 = c(2)*sv1(-1) + c(3)*sv2(-1) + [var=exp(c(7))]  
@state sv2 = sv1(-1)  
@state sv3 = sv2(-1)  
@state sv4 = sv3(-1)  
@state sv5 = sv5(-1) + [var=3]
```

The AR coefficients are parameterized in terms of C(2) and C(3), while the MA coefficients are given by C(4), C(5) and C(6). The variance of the innovation is restricted to be a positive function of C(7). SV5 is the random coefficient on X, with variance restricted to be 3.

Recursive and Random Coefficients

The following example describes a model with one random coefficient (SV1), one recursive coefficient (SV2), and possible correlation between the errors for SV1 and Y:

```
y = c(1) + sv1*x1 + sv2*x2 + [ename = e1, var = exp(c(2))]  
@state sv1 = sv1(-1) + [ename = e2, var = exp(c(3)*x)]  
@state sv2 = sv2(-1)  
@evar cov(e1,e2) = c(4)
```

The variances and covariances in the model are parameterized in terms of the coefficients C(2), C(3) and C(4), with the variances of the observed Y and the unobserved state SV1 restricted to be non-negative functions of the parameters.

Parameter Starting Values

Unless otherwise instructed, EViews will initialize all parameters to the current values in the corresponding coefficient vector or vectors. As in the system object, you may override this default behavior by specifying explicitly the desired values of the parameters using a `PARAM` or `@PARAM` statement. For additional details, see [“Starting Values” on page 315](#).

Specifying Initial Conditions

By default, EViews will handle the initial conditions for you. For some stationary models, steady-state conditions allow us to solve for the values of α_0 and P_0 . For cases where it is not possible to solve for the initial conditions, EViews will treat the initial values as diffuse, setting $\alpha_{1|0} = 0$, and $P_{1|0}$ to an arbitrarily high number to reflect our uncertainty about the values (see [“Technical Discussion” on page 405](#)).

You may, however have prior information about the values of $\alpha_{1|0}$ and $P_{1|0}$. In this case, you can create a vector or matrix that contains the appropriate values, and use the `@MPRIOR` or `@VPRIOR` keywords to perform the assignment.

To set the initial states, enter `@MPRIOR` followed by the name of a vector object. The length of the vector object must match the state dimension. The order of elements should follow the order in which the states were introduced in the specification screen.

```
@mprior v1
@vprior m1
```

To set the initial state variance matrix, enter `@VPRIOR` followed by the name of a sym object (note that it must be a sym object, and not an ordinary matrix object). The dimensions of the sym must match the state dimension, with the ordering following the order in which the states appear in the specification. If you wish to set a specific element to be diffuse, simply assign the element the “NA” missing value. EViews will reset all of the corresponding variances and covariances to be diffuse.

For example, suppose you have a two equation state space object named `SS1` and you want to set the initial values of the state vector and the state variance matrix as:

$$\begin{bmatrix} SV1 \\ SV2 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad \text{var} \begin{bmatrix} SV1 \\ SV2 \end{bmatrix} = \begin{bmatrix} 1 & 0.5 \\ 0.5 & 2 \end{bmatrix} \quad (35.17)$$

First, create a named vector object, say `SVEC0`, to hold the initial values. Click **Object/New Object**, choose **Matrix-Vector-Coef** and enter the name `SVEC0`. Click **OK**, and then choose the type **Vector** and specify the size of the vector (in this case 2 rows). When you click **OK**, EViews will display the spreadsheet view of the vector `SVEC0`. Click the **Edit** +/- button to toggle on edit mode and type in the desired values. Then create a named symmetric matrix object, say `SVAR0`, in an analogous fashion.

Alternatively, you may find it easier to create and initialize the vector and matrix using commands. You can enter the following commands in the command window:

```
vector(2) svec0
svec0.fill 1, 0
sym(2) svar0
svar0.fill 1, 0.5, 2
```

Then, simply add the lines:

```
@mprior svec0
@vprior svar0
```

to your sspace object by editing the specification window. Alternatively, you can type the following commands in the command window:

```
ssl.append @mprior svec0
ssl.append @vprior svar0
```

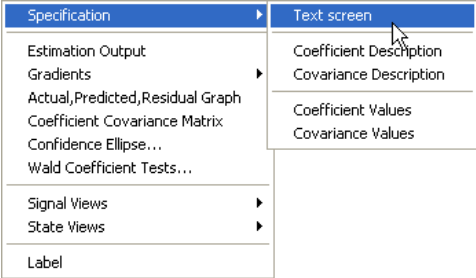
For more details on matrix objects and the `fill` and `append` commands, see [Chapter 18](#), “Matrix Language,” on page 627 of the *User’s Guide I*.

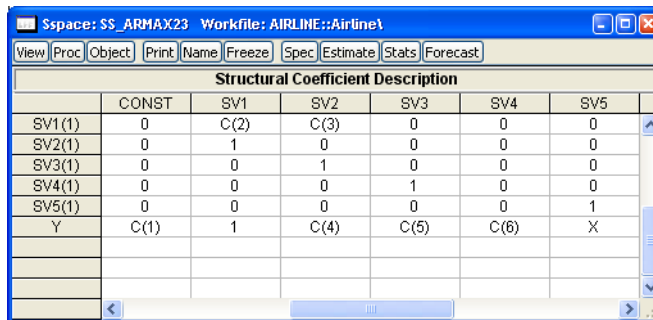
Specification Views

State space models may be very complex. To aid you in examining your specification, EViews provides views which allow you to view the text specification in a more compact form, and to examine the numerical values of your system matrices evaluated at current parameter values.

Click on the **View** menu and select **Specification...** The following Specification views are always available, regardless of whether the sspace has previously been estimated:

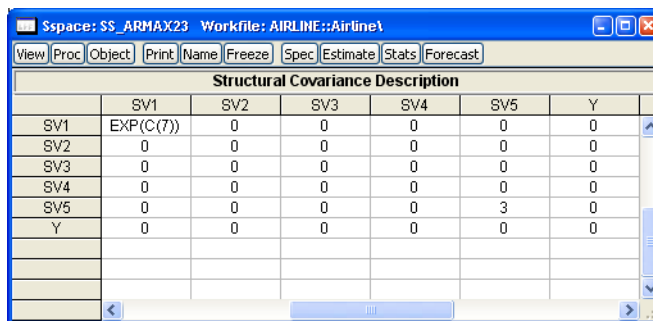
- **Text Screen.** This is the familiar text view of the specification. You should use this view when you create or edit the state space specification. This view may also be accessed by clicking on the **Spec** button on the sspace toolbar.
- **Coefficient Description.** Text description of the structure of your state space specification. The variables on the left-hand side, representing α_{t+1} and y_t , are expressed as linear functions of the state variables α_t , and a remainder term CONST. The elements of the matrix are the corresponding coefficients. For example, the ARMAX example has the following Coefficient Description view:





| | CONST | SV1 | SV2 | SV3 | SV4 | SV5 | |
|--------|-------|------|------|------|------|-----|--|
| SV1(1) | 0 | C(2) | C(3) | 0 | 0 | 0 | |
| SV2(1) | 0 | 1 | 0 | 0 | 0 | 0 | |
| SV3(1) | 0 | 0 | 1 | 0 | 0 | 0 | |
| SV4(1) | 0 | 0 | 0 | 1 | 0 | 0 | |
| SV5(1) | 0 | 0 | 0 | 0 | 0 | 1 | |
| Y | C(1) | 1 | C(4) | C(5) | C(6) | X | |

- **Covariance Description.** Text description of the covariance matrix of the state space specification. For example, the ARMAX example has the following Covariance Description view:



| | SV1 | SV2 | SV3 | SV4 | SV5 | Y | |
|-----|-----------|-----|-----|-----|-----|---|--|
| SV1 | EXP(C(7)) | 0 | 0 | 0 | 0 | 0 | |
| SV2 | 0 | 0 | 0 | 0 | 0 | 0 | |
| SV3 | 0 | 0 | 0 | 0 | 0 | 0 | |
| SV4 | 0 | 0 | 0 | 0 | 0 | 0 | |
| SV5 | 0 | 0 | 0 | 0 | 3 | 0 | |
| Y | 0 | 0 | 0 | 0 | 0 | 0 | |

- **Coefficient Values.** Numeric description of the structure of the signal and the state equations evaluated at current parameter values. If the system coefficient matrix is time-varying, EViews will prompt you for a date/observation at which to evaluate the matrix.
- **Covariance Values.** Numeric description of the structure of the state space specification evaluated at current parameter values. If the system covariance matrix is time-varying, EViews will prompt you for a date/observation at which to evaluate the matrix.

Auto-Specification

To aid you in creating a state space specification, EViews provides you with “auto-specification” tools which will create the text representation of a model that you specify using dialogs. This tool may be very useful if your model is a standard regression with fixed, recursive, and various random coefficient specifications, and/or your errors have a general ARMA structure.

When you select **Proc/Define State Space...** from the menu, EViews opens a three tab dialog. The first tab is used to describe the basic regression portion of your specification. Enter the dependent variable, and any regressors which have fixed or recursive coefficients. You can choose which COEF object EViews uses for indicating unknowns when setting up the specification. At the bottom, you can specify an ARMA structure for your errors. Here, we have specified a simple ARMA(2,1) specification for LOG(PASSENGER).

The screenshot shows the 'Auto-Specification' dialog box with the 'Basic Regression' tab selected. The 'Dependent variables' field contains 'log(passenger)'. The 'Coefficient basename' field contains 'c'. The 'Regressors with fixed coefficients' field is empty. The 'Regressors with recursive coefficients' field is empty. The 'ARMA specification' section shows 'AR order: 2' and 'MA order: 1'. A warning message on the right states: 'Warning! Clicking the OK button will delete existing Sspace results.' The 'OK' and 'Cancel' buttons are at the bottom right.

The second tab of the dialog is used to add any regressors which have random coefficients. Simply enter the appropriate regressors in each of the four edit fields. EViews allows you to define regressors with any combination of constant mean, AR(1), random walk, or random walk (with drift) coefficients.

Lastly, the **Auto-Specification** dialog allows you to choose between basic variance structures for your state space model. Click on the **Variance Specification** tab, and choose between an identity matrix, common diagonal (diagonal with common variances), diagonal, or general (unrestricted) variance matrix for the signals and for the states. The dialog also allows you to allow the signal equation(s) and state equations(s) to have non-zero error covariances.

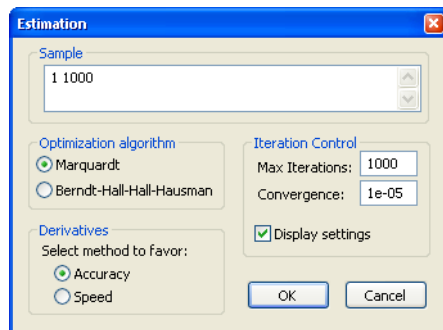
The screenshot shows the 'Auto-Specification' dialog box with the 'Stochastic Regressors' tab selected. The 'Regressors with stochastic coefficients' section contains four empty text boxes for: 'Constant mean (plus noise) coefficients', 'AR(1) coefficients', 'Random walk coefficients', and 'Random walk (with drift) coefficients'. A warning message on the right states: 'Warning! Clicking the OK button will delete existing Sspace results.' The 'OK' and 'Cancel' buttons are at the bottom right.

We emphasize the fact that your sspace object is not restricted to the choices provided in this dialog. If you find that the set of specifications supported by **Auto-Specification** is too restrictive, you may use it the dialogs as a tool to build a basic specification, and then edit the specification to describe your model.

Estimating a State Space Model

Once you have specified a state space model and verified that your specification is correct, you are ready to estimate the model. To open the estimation dialog, simply click on the **Estimate** button on the toolbar or select **Proc/Estimate...**

As with other estimation objects, EViews allows you to set the estimation sample, the maximum number of iterations, convergence tolerance, the estimation algorithm, derivative settings and whether to display the starting values. The default settings should provide a good start for most problems; if you choose to change the settings, see [“Setting Estimation Options” on page 625](#) for related discussion of estimation options. When you click on **OK**, EViews will begin estimation using the specified settings.



There are two additional things to keep in mind when estimating your model:

- Although the EViews Kalman filter routines will automatically handle any missing values in your sample, EViews does require that your estimation sample be contiguous, with no gaps between successive observations.
- If there are no unknown coefficients in your specification, you will still have to “estimate” your sspace to run the Kalman filter and initialize elements that EViews needs in order to perform further analysis.

Interpreting the estimation results

After you choose the variance options and click **OK**, EViews presents the estimation results in the state space window. For example, if we specify an ARMA(2,1) for the log of the monthly international airline passenger totals from January 1949 to December 1960 (from Box and Jenkins, 1976, series G, p. 531):

```
log(passenger) = c(1) + sv1 + c(4)*sv2
@state sv1 = c(2)*sv1(-1) + c(3)*sv2(-1) + [var=exp(c(5))]
@state sv2 = sv1(-1)
```

and estimate the model, EViews will open the estimation output view:

Sspace: SS_ARMA21
 Method: Maximum likelihood (Marquardt)
 Date: 08/17/06 Time: 14:20
 Sample: 1949M01 1960M12
 Included observations: 144
 Convergence achieved after 40 iterations

| | Coefficient | Std. Error | z-Statistic | Prob. |
|----------------|-------------|-----------------------|-------------|--------|
| C(1) | 5.499766 | 0.257517 | 21.35693 | 0.0000 |
| C(2) | 0.409012 | 0.167201 | 2.446235 | 0.0144 |
| C(3) | 0.547166 | 0.164607 | 3.324066 | 0.0009 |
| C(4) | 0.841481 | 0.100167 | 8.400812 | 0.0000 |
| C(5) | -4.589400 | 0.172696 | -26.57500 | 0.0000 |
| | Final State | Root MSE | z-Statistic | Prob. |
| SV1 | 0.267126 | 0.100792 | 2.650278 | 0.0080 |
| SV2 | 0.425489 | 1.32E-09 | 3.23E+08 | 0.0000 |
| Log likelihood | 124.3366 | Akaike info criterion | -1.657452 | |
| Parameters | 5 | Schwarz criterion | -1.554334 | |
| Diffuse priors | 0 | Hannan-Quinn criter. | -1.615551 | |

The bulk of the output view should be familiar from other EViews estimation objects. The information at the top describes the basics of the estimation: the name of the sspace object, estimation method, the date and time of estimation, sample and number of objects in the sample, convergence information, and the coefficient estimates. The bottom part of the view reports the maximized log likelihood value, the number of estimated parameters, and the associated information criteria.

Some parts of the output, however, are new and may require discussion. The bottom section provides additional information about the handling of missing values in estimation. “Likelihood observations” reports the actual number of observations that are used in forming the likelihood. This number (which is the one used in computing the information criteria) will differ from the “Included observations” reported at the top of the view when EViews drops an observation from the likelihood calculation because all of the signal equations have missing values. The number of omitted observations is reported in “Missing observations”. “Partial observations” reports the number of observations that are included in the likelihood, but for which some equations have been dropped. “Diffuse priors” indicates the number of initial state covariances for which EViews is unable to solve and for which there is no user initialization. EViews’ handling of initial states and covariances is described in greater detail in [“Initial Conditions” on page 405](#).

EViews also displays the final one-step ahead values of the state vector, $\alpha_{T+1|T}$, and the corresponding RMSE values (square roots of the diagonal elements of $P_{T+1|T}$). For settings where you may care about the entire path of the state vector and covariance matrix, EViews provides you with a variety of views and procedures for examining the state results in greater detail.

Working with the State Space

EViews provides a variety of specialized tools for specifying and examining your state space specification. As with other estimation objects, the `sspace` object provides additional views and procedures for examining the estimation results, performing inference and specification testing, and extracting results into other EViews objects.

State Space Views

Many of the state space views should be familiar from previous discussion:

- We have already discussed the **Specification...** views in our analysis of “[Specification Views](#)” on page 394.
- The **Estimation Output** view displays the coefficient estimates and summary statistics as described above in “[Interpreting the estimation results](#)” on page 397. You may also access this view by pressing **Stats** on the `sspace` toolbar.
- The **Gradients and Derivatives...** views should be familiar from other estimation objects. If the `sspace` contains parameters to be estimated, this view provides summary and visual information about the gradients of the log likelihood at estimated parameters (if the `sspace` is estimated) or at current parameter values.
- **Actual, Predicted, Residual Graph** displays, in graphical form, the actual and one-step ahead fitted values of the signal dependent variable(s), $y_{t|t-1}$, and the one-step ahead standardized residuals, $e_{t|t-1}$.
- Select **Coefficient Covariance Matrix** to view the estimated coefficient covariance.
- **Wald Coefficient Tests...** allows you to perform hypothesis tests on the estimated coefficients. For details, see “[Wald Test \(Coefficient Restrictions\)](#)” on page 145.
- **Label** allows you to annotate your object. See “[Labeling Objects](#)” on page 72 of the *User’s Guide I*.

| | |
|-----------------------------------|---|
| Specification | ► |
| Estimation Output | |
| Gradients | ► |
| Actual, Predicted, Residual Graph | |
| Coefficient Covariance Matrix | |
| Confidence Ellipse... | |
| Wald Coefficient Tests... | |
| Signal Views | ► |
| State Views | ► |
| Label | |

Note that with the exception of the **Label** and **Specification...** views, these views are available only following successful estimation of your state space model.

Signal Views

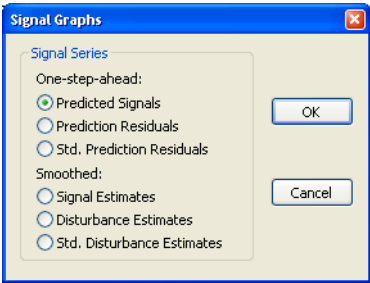
When you click on **View/Signal Views**, EViews displays a sub-menu containing additional view selections. Two of these selections are always available, even if the state space model has not yet been estimated:

| |
|----------------------------------|
| Graph Signal Series... |
| Actual Signal Table |
| Actual Signal Graph |
| Std. Residual Correlation Matrix |
| Std. Residual Covariance Matrix |

- **Actual Signal Table** and **Actual Signal Graph** display the dependent signal variables in spreadsheet and graphical forms, respectively. If there are multiple signal equations, EViews will display a each series with its own axes.

The remaining views are only available following estimation.

- **Graph Signal Series...** opens a dialog with choices for the results to be displayed. The dialog allows you to choose between the one-step ahead predicted signals, $y_{t|t-1}$, the corresponding one-step residuals, $e_{t|t-1}$, or standardized one-step residuals, $e_{t|t-1}$, the smoothed signals, \hat{y}_t , smoothed signal disturbances, \hat{e}_t , or the standardized smoothed signal disturbances, $\hat{e}_t \cdot \pm 2$ (root mean square) standard error bands are plotted where appropriate.



- **Std. Residual Correlation Matrix** and **Std. Residual Covariance Matrix** display the correlation and covariance matrix of the standardized one-step ahead signal residual, $e_{t|t-1}$.

State Views

To examine the unobserved state components, click on **View/State Views** to display the state submenu. EViews allows you to examine the initial or final values of the state components, or to graph the full time-path of various filtered or smoothed state data.

| |
|---------------------------------|
| Graph State Series... |
| Initial State Vector |
| Initial State Covariance Matrix |
| Final State Vector |
| Final State Covariance Matrix |

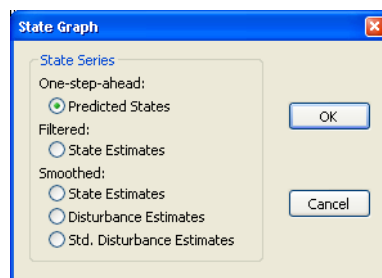
Two of the views are available either before or after estimation:

- **Initial State Vector** and **Initial State Covariance Matrix** display the values of the initial state vector, α_0 , and covariance matrix, P_0 . If the unknown parameters have previously been estimated, EViews will evaluate the initial conditions using the estimated values. If the sspace has not been estimated, the current coefficient values will be used in evaluating the initial conditions.

This information is especially relevant in models where EViews is using the current values of the system matrices to solve for the initial conditions. In cases where you are having difficulty starting your estimation, you may wish to examine the values of the initial conditions at the starting parameter values for any sign of problems.

The remainder of the views are only available following successful estimation:

- **Final State Vector** and **Final State Covariance Matrix** display the values of the final state vector, α_T , and covariance matrix, P_T , evaluated at the estimated parameters.
- Select **Graph State Series...** to display a dialog containing several choices for the state information. You can graph the one-step ahead predicted states, $a_{t|t-1}$, the filtered (contemporaneous) states, a_t , the smoothed state estimates, α_t , smoothed state disturbance estimates, v_t , or the standardized smoothed state disturbances, $\hat{\eta}_t$. In each case, the data are displayed along with corresponding ± 2 standard error bands.



State Space Procedures

You can use the EViews procedures to create, estimate, forecast, and generate data from your state space specification. Select **Proc** in the sspace toolbar to display the available procedures:

- **Define State Space...** calls up the **Auto-Specification** dialog (see “[Auto-Specification](#)” on page 395). This feature provides a method of specifying a variety of common state space specifications using interactive menus.
- Select **Estimate...** to estimate the parameters of the specification (see “[Estimating a State Space Model](#)” on page 397).



These above items are available both before and after estimation. The automatic specification tool will replace the existing state space specification and will clear any results.

Once you have estimated your sspace, EViews provides additional tools for generating data:

- The **Forecast...** dialog allows you to generate forecasts of the states, signals, and the associated standard errors using alternative methods and initialization approaches.

First, select the forecast method. You can select between dynamic, smoothed, and n -period ahead forecasting, as described in “[Forecasting](#)” on page 386. Note that any

lagged endogenous variables on the right-hand side of your signal equations will be treated as predetermined for purposes of forecasting.

EViews allows you to save various types of forecast output in series in your workfile. Simply check any of the output boxes, and specify the names for the series in the corresponding edit field.

You may specify the names either as a list or using a wildcard expression. If you choose to list the names, the number of identifiers must match the number of signals in your specification. You should be aware that if an output series with a specified name already exists in the workfile, EViews will overwrite the entire contents of the series.

The screenshot shows the 'Forecast' dialog box in EViews. It is divided into several sections:

- Forecast method:** Three radio buttons are present: 'Dynamic' (selected), 'Smoothed', and 'N-period ahead' (with a value of 1).
- Forecast sample:** A text box containing '1949m01 1960m12'.
- Forecast output:** Four checkboxes are present: 'Signals' (checked), 'Signal SEs', 'States', and 'State SEs'. Each checked item has a corresponding text box for naming the output series, all containing the wildcard '*f'.
- Initialize states with:** Three radio buttons are present: 'One-step states from estimation' (selected), 'Smoothed states from estimation', and 'EViews computed'. Below these are two text boxes for 'Mean' and 'Variance'.
- Buttons:** 'OK' and 'Cancel' buttons are at the bottom right.

If you use a wildcard expression, EViews will substitute the name of each signal in the appropriate position in the wildcard expression. For example, if you have a model with signals Y1 and Y2, and elect to save the one-step predictions in “PRED*”, EViews will use the series PREDY1 and PREDY2 for output. There are two limitations to this feature: (1) you may not use the wildcard expression “*” to save signal results since this will overwrite the original signal data, and (2) you may not use a wildcard when any signal dependent variables are specified by expression, or when there are multiple equations for a signal variable. In both cases, EViews will be unable to create the new series and will generate an error message.

Keep in mind that if your signal dependent variable is an expression, EViews will only provide forecasts of the expression. Thus, if your signal variable is LOG(Y), EViews will forecast the logarithm of Y.

Now enter a sample and specify the treatment of the initial states, and then click **OK**. EViews will compute the forecast and will place the results in the specified series. No output window will open.

There are several options available for setting the initial conditions. If you wish, you can instruct the sspace object to use the **One-step ahead** or **Smoothed** estimates of the state and state covariance as initial values for the forecast period. The two initialization methods differ in the amount of information used from the estimation sample; one-step ahead uses information up to the beginning of the forecast period, while smoothed uses the entire estimation period.

Alternatively, you may use **EViews computed** initial conditions. As in estimation, if possible, EViews will solve the Algebraic Riccati equations to obtain values for the initial state and state covariance at the start of each forecast interval. If solution of these conditions is not possible, EViews will use diffuse priors for the initial values.

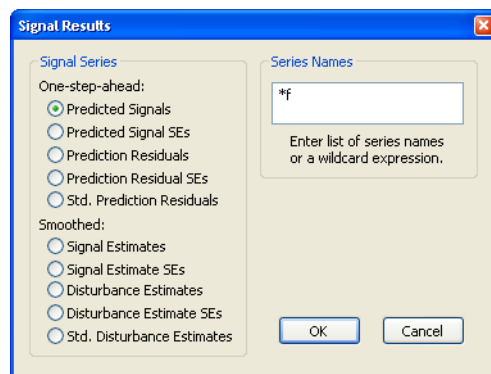
Lastly, you may choose to provide a vector and sym object which contain the values for the forecast initialization. Simply select **User Specified** and enter the name of valid EViews objects in the appropriate edit fields.

Note that when performing either dynamic or smoothed forecasting, EViews requires that one-step ahead and smoothed initial conditions be computed from the estimation sample. If you choose one of these two forecasting methods and your forecast period begins either before or after the estimation sample, EViews will issue an error and instruct you to select a different initialization method.

When computing n -step ahead forecasting, EViews will adjust the start of the forecast period so that it is possible to obtain initial conditions for each period using the specified method. For the one-step ahead and smoothed methods, this means that at the earliest, the forecast period will begin $n - 1$ observations into the estimation sample, with earlier forecasted values set to NA. For the other initialization methods, forecast sample endpoint adjustment is not required.

- **Make Signal Series...** allows you to create series containing various signal results computed over the estimation sample. Simply click on the menu entry to display the results dialog.

You may select the one-step ahead predicted signals, $\tilde{y}_{t|t-1}$, one-step prediction residuals, $\epsilon_{t|t-1}$, smoothed signal, \hat{y}_t , or signal disturbance estimates, $\hat{\epsilon}_t$. EViews also allows you to save the corresponding standard errors for each of these components (square roots of the diagonal elements of $F_{t|t-1}$, S_t , and $\hat{\Omega}_t$), or the standardized values of the one-step residuals and smoothed disturbances, $e_{t|t-1}$ or \hat{e}_t .

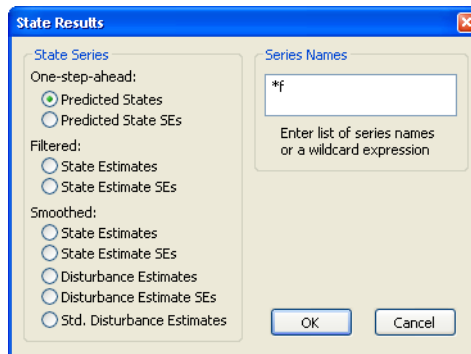


Next, specify the names of your series in the edit field using a list or wildcards as described above. Click **OK** to generate a group containing the desired signal series.

As above, if your signal dependent variable is an expression, EViews will only export results based upon the entire expression.

- **Make State Series...** opens a dialog allowing you to create series containing results for the state variables computed over the estimation sample. You can choose to save either the one-step ahead state estimate, $a_{t|t-1}$, the filtered state mean, a_t , the smoothed states, $\hat{\alpha}_t$, state disturbances, \hat{v}_t , standardized state disturbances, $\hat{\eta}_t$, or the corresponding standard error series (square roots of the diagonal elements of $P_{t|t-1}$, P_t , V_t and $\hat{\Omega}_t$).

Simply select one of the output types, and enter the names of the output series in the edit field. The rules for specifying the output names are the same as for the **Forecast...** procedure described above. Note that the wildcard expression “*” is permitted when saving state results. EViews will simply use the state names defined in your specification.



We again caution you that if an output series exists in the workfile, EViews will overwrite the entire contents of the series.

- Click on **Make Endogenous Group** to create a group object containing the signal dependent variable series.
- **Make Gradient Group** creates a group object with series containing the gradients of the log likelihood. These series are named “GRAD##” where ## is a unique number in the workfile.
- **Make Kalman Filter** creates a new state space object containing the current specification, but with all parameters replaced by their estimated values. In this way you can “freeze” the current state space for additional analysis. This procedure is similar to the **Make Model** procedure found in other estimation objects.
- **Make Model** creates a model object containing the state space equations.
- **Update Coefs from Sspace** will place the estimated parameters in the appropriate coefficient vectors.

Converting from Version 3 Sspace

Those of you who have worked with the EViews Version 3 sspace object will undoubtedly be struck by the large number of changes and additional features in Version 4 and later. In addition to new estimation options, views and procedures, we have changed the underlying specification syntax to provide you with considerable additional flexibility. A wide variety of

specifications that were not supported in earlier versions may be estimated with the current sspace object.

The cost of these additional features and added flexibility is that Version 3 sspace objects are not fully compatible with those in the current version. This has two important practical effects:

- If you load in a workfile which contains a Version 3 sspace object, *all previous estimation results will be cleared* and the text of the specification will be translated to the current syntax. The original text will be retained as comments at the bottom of your sspace specification.
- If you take a workfile which contains a new sspace object created with EViews 4 or later and attempt to read it into an earlier version of EViews, the object will not be read, and EViews will warn you that a partial load of the workfile was performed. If you subsequently save the workfile, *the original sspace object will not be saved with the workfile*.

Technical Discussion

Initial Conditions

If there are no @MPRIOR or @VPRIOR statements in the specification, EViews will either: (1) solve for the initial state mean and variance, or (2) initialize the states and variances using diffuse priors.

Solving for the initial conditions is only possible if the state transition matrices T , and variance matrices P and Q are non time-varying and satisfy certain stability conditions (see Harvey, 1989, p. 121). If possible, EViews will solve for the conditions $P_{1|0}$ using the familiar relationship: $(I - T \otimes T) \times \text{vec}(P) = \text{vec}(Q)$. If this is not possible, the states will be treated as diffuse unless otherwise specified.

When using diffuse priors, EViews follows the method adopted by Koopman, Shephard and Doornik (1999) in setting $\alpha_{1|0} = 0$, and $P_{1|0} = \kappa I_M$, where the κ is an arbitrarily chosen large number. EViews uses the authors' recommendation that one first set $\kappa = 10^6$ and then adjust it for scale by multiplying by the largest diagonal element of the residual covariances.

References

- Box, George E. P. and Gwilym M. Jenkins (1976). *Time Series Analysis: Forecasting and Control*, Revised Edition, Oakland, CA: Holden-Day.
- Hamilton, James D. (1994a). *Time Series Analysis*, Princeton University Press.
- Hamilton, James D. (1994b). "State Space Models," Chapter 50 in Robert F. Engle and Daniel L. McFadden (eds.), *Handbook of Econometrics, Volume 4*, Amsterdam: Elsevier Science B.V.

Harvey, Andrew C. (1989). *Forecasting, Structural Time Series Models and the Kalman Filter*, Cambridge: Cambridge University Press.

Koopman, Siem Jan, Neil Shephard, and Jurgen A. Doornik (1999). "Statistical Algorithms for Models in State Space using SsfPack 2.2," *Econometrics Journal*, 2(1), 107-160.

Chapter 36. Models

A model in EViews is a set of one or more equations that jointly describe the relationship between a set of variables. The model equations can come from many sources: they can be simple identities, they can be the result of estimation of single equations, or they can be the result of estimation using any one of EViews' multiple equation estimators.

EViews models allow you to combine equations from all these sources inside a single object, which may be used to create a deterministic or stochastic joint forecast or simulation for all of the variables in the model. In a deterministic setting, the inputs to the model are fixed at known values, and a single path is calculated for the output variables. In a stochastic environment, uncertainty is incorporated into the model by adding a random element to the coefficients, the equation residuals or the exogenous variables.

Models also allow you to examine simulation results under different assumptions concerning the variables that are determined outside the model. In EViews, we refer to these sets of assumptions as *scenarios*, and provide a variety of tools for working with multiple model scenarios.

Even if you are working with only a single equation, you may find that it is worth creating a model from that equation so that you may use the features provided by the EViews Model object.

Overview

The following section provides a brief introduction to the purpose and structure of the EViews model object, and introduces terminology that will be used throughout the rest of the chapter.

A model consists of a set of *equations* that describe the relationships between a set of *variables*.

The variables in a model can be divided into two categories: those determined inside the model, which we refer to as the *endogenous variables*, and those determined outside the model, which we refer to as the *exogenous variables*. A third category of variables, the *add factors*, are a special case of exogenous variables.

In its most general form, a model can be written in mathematical notation as:

$$F(y, x) = 0 \tag{36.1}$$

where y is the vector of endogenous variables, x is the vector of exogenous variables, and F is a vector of real-valued functions $f_i(y, x)$. For the model to have a unique solution, there should typically be as many equations as there are endogenous variables.

In EViews, each equation in the model must have a unique endogenous variable assigned to it. That is, each equation in the model must be able to be written in the form:

$$y_i = f_i(y, x) \quad (36.2)$$

where y_i is the endogenous variable assigned to equation i . EViews has the ability to *normalize* equations involving simple transformations of the endogenous variable, rewriting them automatically into explicit form when necessary. Any variable that is not assigned as the endogenous variable for any equation is considered exogenous to the model.

Equations in an EViews model can either be *inline* or *linked*. An inline equation contains the specification for the equation as text within the model. A linked equation is one that brings its specification into the model from an external EViews object such as a single or multiple equation estimation object, or even another model. Linking allows you to couple a model more closely with the estimation procedure underlying the equations, or with another model on which it depends. For example, a model for industry supply and demand might link to another model and to estimated equations:

| Industry Supply And Demand Model | |
|----------------------------------|---|
| ← | link to macro model object for forecasts of total consumption |
| ← | link to equation object containing industry supply equation |
| ← | link to equation object containing industry demand equation |
| ← | inline identity: supply = demand |

Equations can also be divided into *stochastic equations* and *identities*. Roughly speaking, an identity is an equation that we would expect to hold exactly when applied to real world data, while a stochastic equation is one that we would expect to hold only with random error. Stochastic equations typically result from statistical estimation procedures while identities are drawn from accounting relationships between the variables.

The most important operation performed on a model is to *solve* the model. By solving the model, we mean that for a given set of values of the exogenous variables, X , we will try to find a set of values for the endogenous variables, Y , so that the equations in the model are satisfied within some numerical tolerance. Often, we will be interested in solving the model over a sequence of periods, in which case, for a simple model, we will iterate through the periods one by one. If the equations of the model contain future endogenous variables, we

may require a more complicated procedure to solve for the entire set of periods simultaneously.

In EViews, when solving a model, we must first associate data with each variable in the model by *binding* each of the model variables to a series in the workfile. We then solve the model for each observation in the selected sample and place the results in the corresponding series.

When binding the variables of the model to specific series in the workfile, EViews will often modify the name of the variable to generate the name of the series. Typically, this will involve adding an extension of a few characters to the end of the name. For example, an endogenous variable in the model may be called “Y”, but when EViews solves the model, it may assign the result into an observation of a series in the workfile called “Y_0”. We refer to this mapping of names as *aliasing*. Aliasing is an important feature of an EViews model, as it allows the variables in the model to be mapped into different sets of workfile series, without having to alter the equations of the model.

When a model is solved, aliasing is typically applied to the endogenous variables so that historical data is not overwritten. Furthermore, for models which contain lagged endogenous variables, aliasing allows us to bind the lagged variables to either the actual historical data, which we refer to as a *static forecast*, or to the values solved for in previous periods, which we refer to as a *dynamic forecast*. In both cases, the lagged endogenous variables are effectively treated as exogenous variables in the model when solving the model for a single period.

Aliasing is also frequently applied to exogenous variables when using *model scenarios*. Model scenarios allow you to investigate how the predictions of your model vary under different assumptions concerning the path of exogenous variables or add factors. In a scenario, you can change the path of an exogenous variable by overriding the variable. When a variable is *overridden*, the values for that variable will be fetched from a workfile series specific to that scenario. The name of the series is formed by adding a suffix associated with the scenario to the variable name. This same suffix is also used when storing the solutions of the model for the scenario. By using scenarios it is easy to compare the outcomes predicted by your model under a variety of different assumptions without having to edit the structure of your model.

The following table gives a typical example of how model aliasing might map variable names in a model into series names in the workfile:

| Model Variable | | Workfile Series |
|----------------|-------|-------------------|
| endogenous Y | → Y | historical data |
| | → Y_0 | baseline solution |

| | | | |
|-------------|---|-----|---|
| | → | Y_1 | scenario 1 |
| exogenous X | → | X | historical data followed by baseline forecast |
| | → | X_1 | overridden forecast for scenario 1 |

Earlier, we mentioned a third category of variables called *add factors*. An add factor is a special type of exogenous variable that is used to shift the results of a stochastic equation to provide a better fit to historical data or to fine-tune the forecasting results of the model. While there is nothing that you can do with an add factor that could not be done using exogenous variables, EViews provides a separate interface for add factors to facilitate a number of common tasks.

An Example Model

In this section, we demonstrate how we can use the EViews model object to implement a simple macroeconomic model of the U.S. economy. The specification of the model is taken from Pindyck and Rubinfeld (1998, p. 390). We have provided the data and other objects relating to the model in the sample workfile MACROMOD.WF1. You may find it useful to follow along with the steps in the example, and you can use the workfile to experiment further with the model object.

The macro model contains three stochastic equations and one identity. In EViews notation, these can be written:

$$\begin{aligned}
 cn &= c(1) + c(2)*y + c(3)*cn(-1) \\
 i &= c(4) + c(5)*(y(-1)-y(-2)) + c(6)*y + c(7)*r(-4) \\
 r &= c(8) + c(9)*y + c(10)*(y-y(-1)) + c(11)*(m-m(-1)) + c(12)*(r(-1)+r(-2)) \\
 y &= cn + i + g
 \end{aligned}$$

where:

- CN is real personal consumption
- I is real private investment
- G is real government expenditure
- Y is real GDP less net exports
- R is the interest rate on three-month treasury bills
- M is the real money supply, narrowly defined (M1)

and the C(i) are the unknown coefficients.

The model follows the structure of a simple textbook ISLM macroeconomic model, with expenditure equations relating consumption and investment to GDP and interest rates, and a money market equation relating interest rates to GDP and the money supply. The fourth equation is the national accounts expenditure identity which ensures that the components of GDP add to total GDP. The model differs from a typical textbook model in its more dynamic structure, with many of the variables appearing in lagged or differenced form.

To begin, we must first estimate the unknown coefficients in the stochastic equations. For simplicity, we estimate the coefficients by simple single equation OLS. Note that this approach is not strictly valid, since Y appears on the right-hand side of several of the equations as an independent variable but is endogenous to the system as a whole. Because of this, we would expect Y to be correlated with the residuals of the equations, which violates the assumptions of OLS estimation. To adjust for this, we would need to use some form of instrumental variables or system estimation (for details, see the discussion of single equation “Two-stage Least Squares,” beginning on page 37 and system “Two-Stage Least Squares” and related sections beginning on page 309).

To estimate the equations in EViews, we create three new equation objects in the workfile (using **Object/New Object.../Equation**), and then enter the appropriate specifications. Since all three equations are linear, we can specify them using list form. To minimize confusion, we will name the three equations according to their endogenous variables. The resulting names and specifications are:

| | |
|----------------|-------------------------------------|
| Equation EQCN: | cn c y cn(-1) |
| Equation EQI: | i c y(-1)-y(-2) y r(-4) |
| Equation EQR: | r c y y-y(-1) m-m(-1) r(-1) + r(-2) |

The three equations estimate satisfactorily and provide a reasonably close fit to the data, although much of the fit probably comes from the lagged endogenous variables. The consumption and investment equations show signs of heteroskedasticity, possibly indicating that we should be modeling the relationships in log form. All three equations show signs of serial correlation. We will ignore these problems for the purpose of this example, although you may like to experiment with alternative specifications and compare their performance.

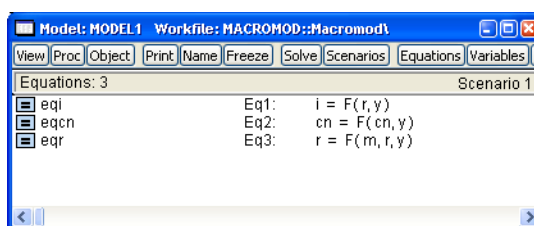
Now that we have estimated the three equations, we can proceed to the model itself. To create the model, we simply select **Object/New Object.../Model** from the menus. To keep the model permanently in the workfile, we name the model by clicking on the **Name** button, enter the name MODEL1, and click on **OK**.

When first created, the model object defaults to *equation view*. Equation view allows us to browse through the specifications and properties of the equations contained in the model. Since we have not yet added any equations to the model, this window will appear empty.

To add our estimated stochastic equations to the model, we can simply copy-and-paste them across from the workfile window. To copy-and-paste, first select the objects in the workfile window, and then use **Edit/Copy** or the right mouse button menu to copy the objects to the clipboard. Click anywhere in the model object window, and use **Edit/Paste** or the right mouse button menu to paste the objects into the model object window.

Alternatively, we could have combined the two steps by first highlighting the three equations, right-mouse clicking, and selecting **Open as Model**. EViews will create a new unnamed model containing the three equations. Press on the **Name** button to name the model object.

The three estimated equations should now appear in the equation window. Each equation appears on a line with an icon showing the type of object, its name, its equation number, and a symbolic representation of the equation in terms of the variables that it contains. Double clicking on any equation will bring up a dialog of properties of that equation. For the moment, we do not need to alter any of these properties.

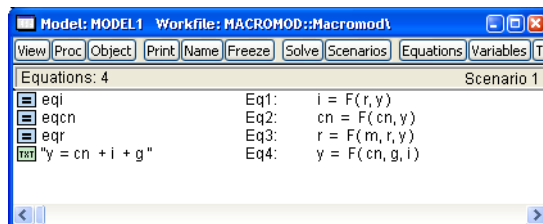


We have added our three equations as linked equations. This means if we go back and reestimate one or more of the equations, we can automatically update the equations in the model to the new estimates by using the procedure **Proc/Links/Update All Links**.

To complete the model, we must add our final equation, the national accounts expenditure identity. There is no estimation involved in this equation, so instead of including the equation via a link to an external object, we merely add the equation as inline text.

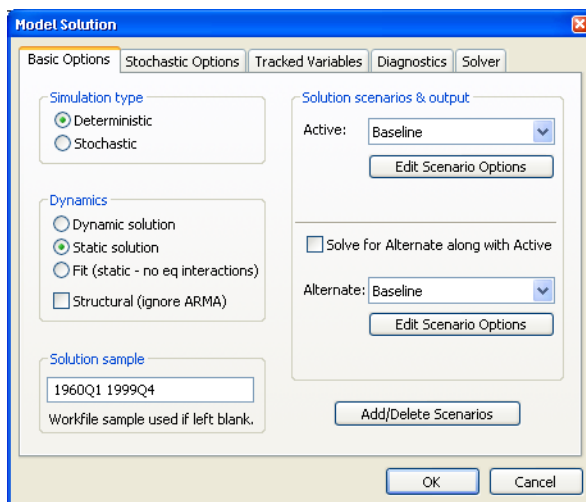
To add the identity, we click with the right mouse button anywhere in the equation window, and select **Insert....** A dialog box will appear titled **Model Source Edit** which contains a text box with the heading **Enter one or more lines**. Simply type the identity, " $Y = CN + I + G$ ", into the text box, then click on **OK** to add it to the model.

The equation should now appear in the model window. The appearance differs slightly from the other equations, which is an indicator that the new equation is an inline text equation rather than a link.



Our model specification is now complete. At this point, we can proceed straight to solving the model. To solve the model, simply click on the **Solve** button in the model window button bar.

There are many options available from the dialog, but for the moment we will consider only the basic settings. As our first exercise in assessing our model, we would like to examine the ability of our model to provide one-period ahead forecasts of our endogenous variables. To do this, we can look at the predictions of our model against our historical data, using actual values for both the exogenous and the lagged endogenous variables of the model. In EViews, we refer to this as a *static* simulation. We may easily perform this type of simulation by choosing **Static solution** in the **Dynamics** box of the dialog.

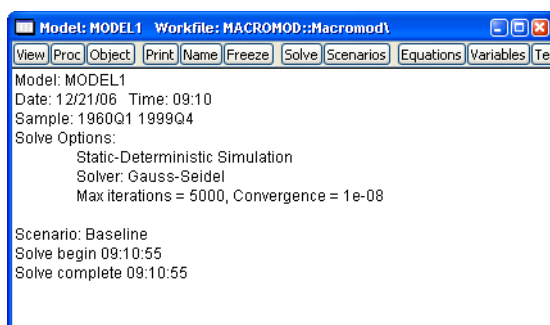


We must also adjust the sample over which to solve the model, so as to avoid initializing our solution with missing values from our data. Most of our series are defined over the range of 1947Q1 to 1999Q4, but our money supply series is available only from 1959Q1. Because of this, we set the sample to 1960Q1 to 1999Q4, allowing a few extra periods prior to the sample for any lagged variables.

We are now ready to solve the model. Simply click on **OK** to start the calculations. The model window will switch to the **Solution Messages** view.

The output should be fairly self-explanatory. In this case, the solution took less than a second and there were no errors while performing the calculations.

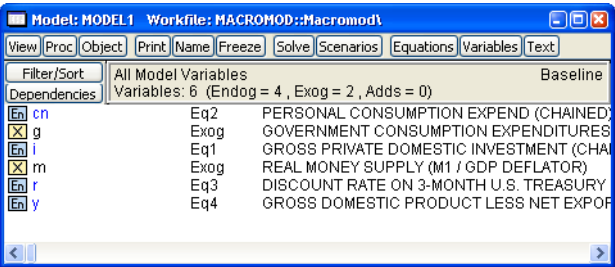
Now that we have solved the model, we would like to look at the results. When we solved the model, the results for the endogenous variables were placed into series in the work-



file with names determined by the name aliasing rules of the model. Since these series are ordinary EViews objects, we could use the workfile window to open the series and examine them directly. However, the model object provides a much more convenient way to work with the series through a view called the Variable View.

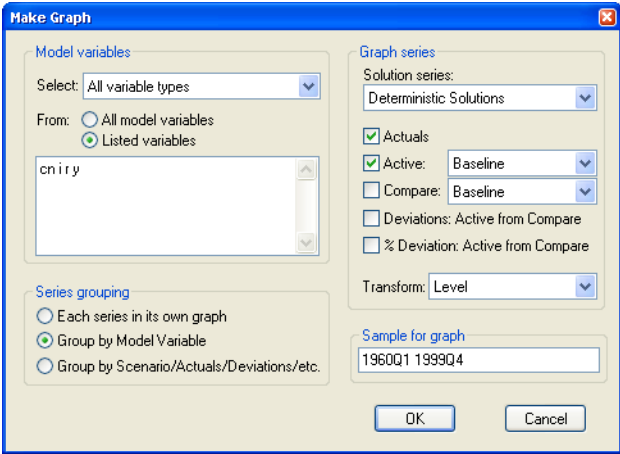
The easiest way to switch to the variable view is to click on the button labeled **Variables** on the model window button bar.

In the variable view, each line in the window is used to represent a variable. The line contains an icon indicating the variable type (endogenous, exogenous or add factor), the name of the variable, the equation with which the variable is associated (if any), and the description field from the label of the underlying series (if available). The name of the variable may be colored according to its status, indicating whether it is being traced (blue) or whether it has been overridden (red). In our model, we can see from the variable view that CN, I, R and Y are endogenous variables in the model, while G and M are exogenous.



Much of the convenience of the variable view comes from the fact that it allows you to work directly with the names of the variables in the model, rather than the names of series in the workfile. This is useful because when working with a model, there will often be many different series associated with each variable. For endogenous variables, there will be the actual historical values and one or more series of solution values. For exogenous variables, there may be several alternative scenarios for the variable. The variable view and its associated procedures help you move between these different sets of series without having to worry about the many different names involved.

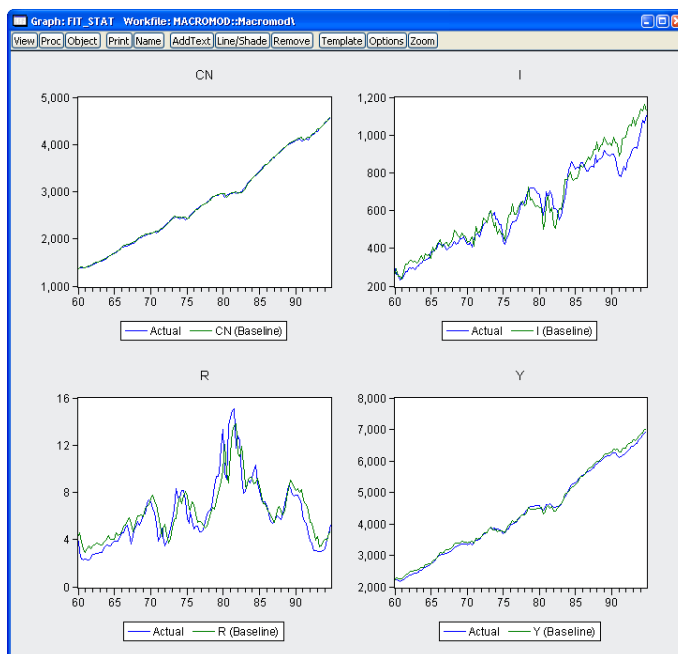
For example, to look at graphs containing the actual and fitted values for the endogenous variables in our model, we simply select the four variables (by holding down the control key and clicking on the variable names), then use **Proc/Make Graph...** to enter the dialog.



Again, the dialog has many options, but for our current purposes, we can leave most settings at their default values. Simply make sure that the **Actuals** and **Active** checkboxes are checked, set the sample for the graph to 1960Q1 to 1999Q4, then click on **OK**.

The graphs show that as a one-step ahead predictor, the model performs quite well, although the ability of the model to predict investment deteriorates during the second half of the sample.

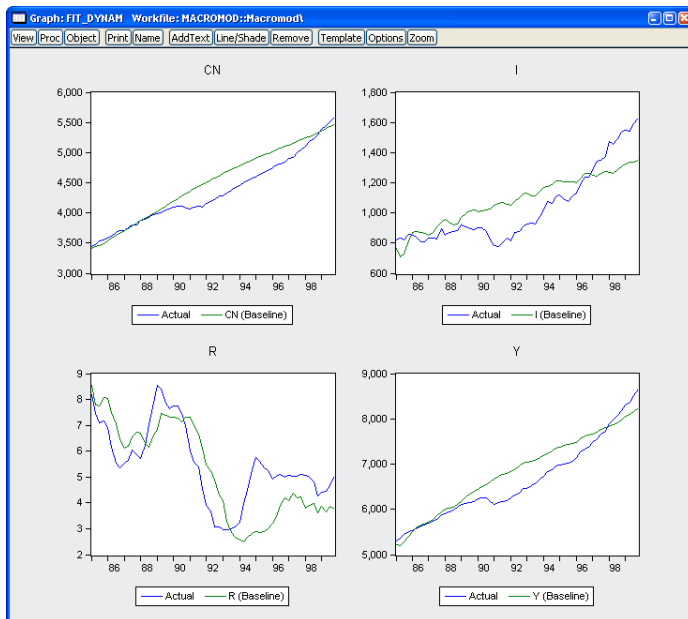
An alternative way of evaluating the model is to examine how the model performs when used to forecast many periods into the future. To do this, we must use our forecasts from previous periods, not actual historical data, when assigning values to the lagged endogenous terms in our model. In EViews, we refer to such a forecast as a *dynamic* forecast.



To perform a dynamic forecast, we resolve the model with a slightly different set of options. Return to the model window and again click on the **Solve** button. In the model solution dialog, choose **Dynamic solution** in the **Dynamics** section of the dialog, and set the solution sample to 1985Q1 to 1999Q4.

Click on **OK** to solve the model. To examine the results, return to the variable view, select the endogenous series again, and use **Proc/Make Graph...** exactly as above. Make sure the sample is set to 1985Q1 to 1999Q4 then click on **OK**. The results illustrate how our model would have performed if we had used it back in 1985 to make a forecast for the economy over the next fifteen years, assuming that we had used the correct paths for the exogenous variables (in reality, we would not have known these values at the time the forecasts were generated). Not surprisingly, the results show substantial deviations from the actual outcomes, although they do seem to follow the general trends in the data.

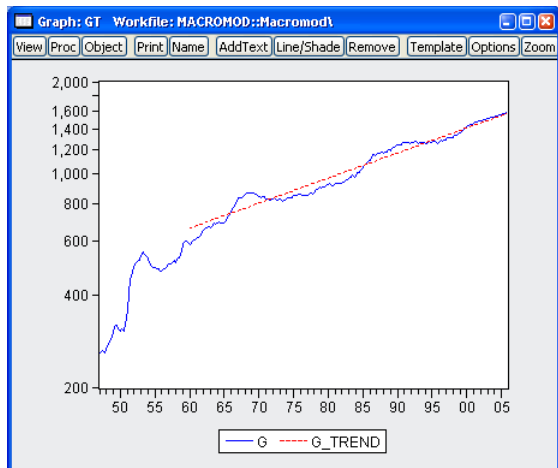
Once we are satisfied with the performance of our model against historical data, we can use the model to forecast future values of our endogenous variables. The first step in producing such a forecast is to decide on values for our exogenous variables during the forecast period. These may be based on our best guess as to what will actually happen, or they may be simply one particular possibility that we are interested in considering. Often we will be interested in constructing several different paths and then comparing the results.



In our model, we must provide future values for our two exogenous variables: government expenditure (G), and the real money supply (M). For our example, we will try to construct a set of paths that broadly follow the trends of the historical data.

A quick look at our historical series for G suggests that the growth rate of G has been fairly constant since 1960, so that the log of G roughly follows a linear trend. Where G deviates from the trend, the deviations seem to follow a cyclical pattern.

As a simple model of this behavior, we can regress the log of G against a constant and a time trend, using an AR(4) error structure to model



the cyclical deviations. This gives the following equation, which we save in the workfile as EQG:

$$\log(g) = 6.252335363 + 0.004716422189 * @trend +$$

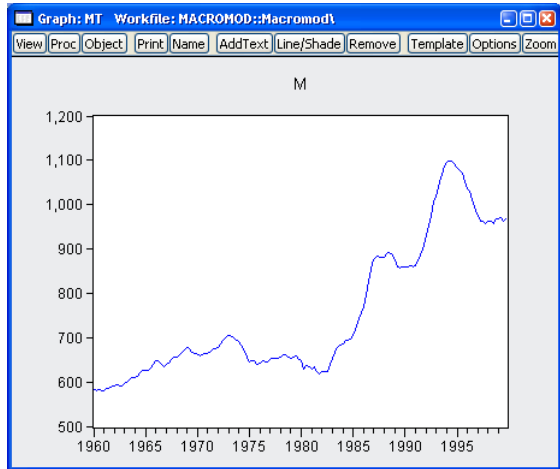
$$[ar(1)=1.169491542, ar(2)=0.1986105964, ar(3)=0.239913126, ar(4)=-0.2453607091]$$

To produce a set of future values for G, we can use this equation to perform a dynamic forecast from 2000Q1 to 2005Q4, saving the results back into G itself (see [page 427](#) for details).

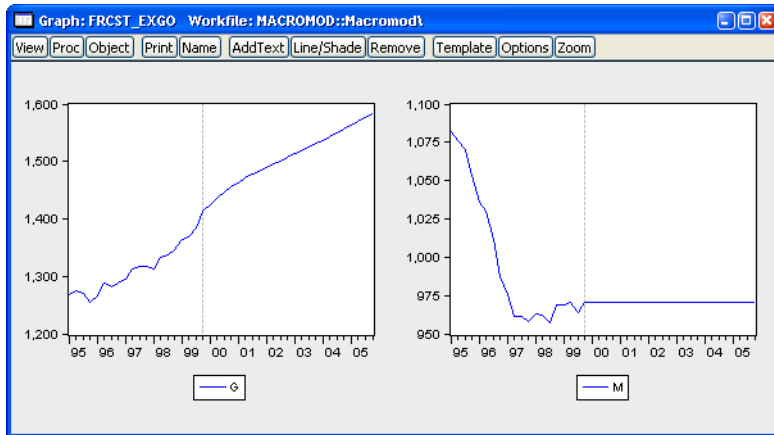
The historical path of the real M1 money supply, M, is quite different from G, showing spurts of growth followed by periods of stability. For now, we will assume that the real money supply simply remains at its last observed historical value over the entire forecast period.

We can use an EViews series statement to fill in this path. The following lines will fill the series M from 2000Q1 to the last observation in the sample with the last observed historical value for M:

```
smpl 2000q1 @last
series m = m(-1)
smpl @all
```



We now have a set of possible values for our exogenous variables over the forecast period. After plotting, we have:



To produce forecasts for our *endogenous* variables, we return to the model window, click on **Solve**, choose **Dynamic Solution**, set the forecast sample for 2000Q1 to 2005Q4, and then click on **OK**. The Solution Messages screen should appear, indicating that the model was successfully solved.

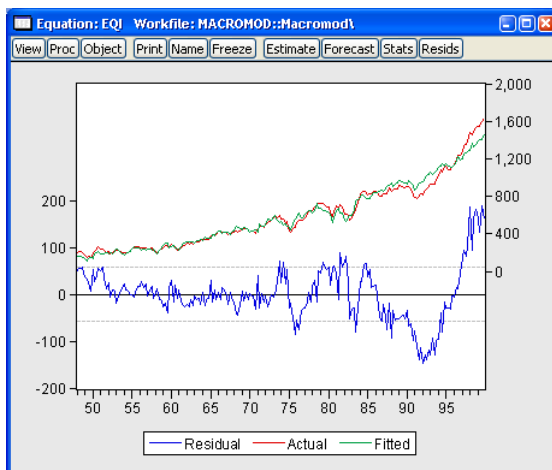
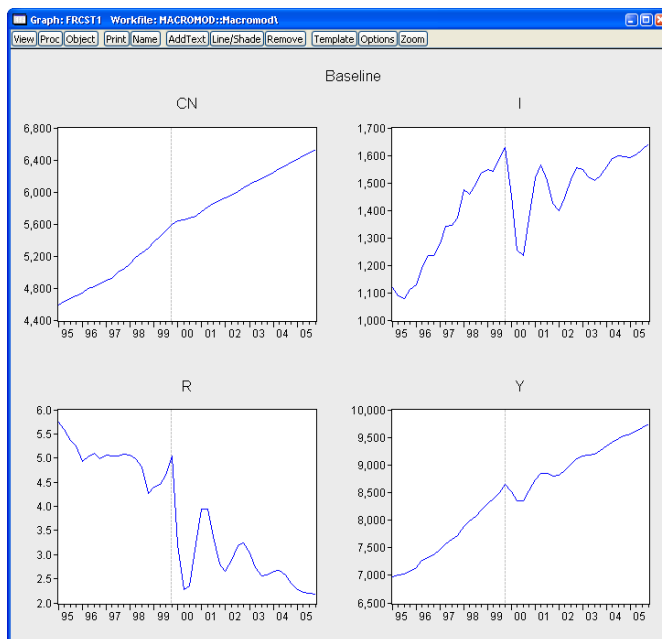
To examine the results in a graph, we again use **Proc/Make Graph...** from the variables view, set the sample to 1995Q1 to 2005Q4 (so that we include five years of historical data), then click on **OK**. After adding a line in 1999Q4 to separate historical and actual results, we get a graph showing the results:

We observe strange behavior in the results. At the beginning of the forecast period, we see a heavy dip in investment, GDP, and interest rates. This is followed by a series of oscillations in these series with a period of about a year, which die out slowly during the forecast period. This is not a particularly convincing forecast.

There is little in the paths of our exogenous variables or the history of our endogenous variables that would lead to this sharp dip, suggesting

that the problem may lie with the residuals of our equations. Our investment equation is the most likely candidate, as it has a large, persistent positive residual near the end of the historical data (see figure below). This residual will be set to zero over the forecast period when solving the model, which might be the cause of the sudden drop in investment at the beginning of the forecast.

One way of dealing with this problem would be to change the specification of the investment equation. The simplest modification would be to add an autoregressive component to the equation, which would help reduce the persistence of the error. A better alternative would be to try to modify the variables in the equation so that the equation can provide some explanation for the sharp rise in investment during the 1990s.



An alternative approach to the problem is to leave the equation as it is, but to include an add factor in the equation so that we can model the path of the residual by hand. To include the add factor, we switch to the equation view of the model, double click on the investment equation, EQI, select the **Add factors** tab. Under **Factor type**, choose **Equation intercept (residual shift)**. A prompt will appear asking if we would like to create the add factor series. Click on **Yes** to create the series. When you return to the variable view, you should see that a new variable, I_A, has been added to the list of variables in the model.

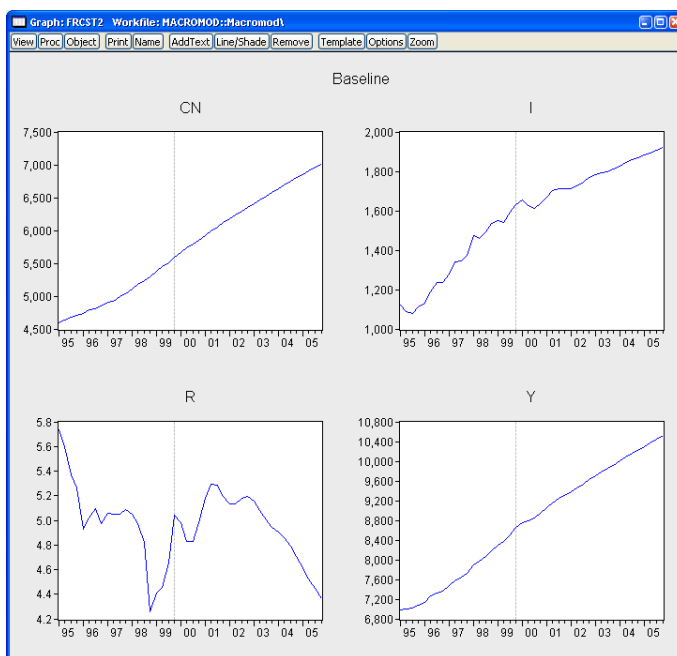
Using the add factor, we can specify any path we choose for the residual of the investment equation during the forecast period. By examining the **Actual/Fitted/Residual Graph** view from the equation object, we see that near the end of the historical data, the residual appears to be hovering around a value of about 160. We will assume that this value holds throughout the forecast period. We can set the add factor using a few simple EViews commands:

```
smpl 2000q1 @last
i_a = 160
smpl @all
```

With the add factor in place, we can follow exactly the same procedure that we followed above to produce a new set of solutions for the model and a new graph for the results.

Including the add factor in the model has made the results far more appealing. The sudden dip in the first period of the forecast that we saw above has been removed. The oscillations are still apparent, but are much less pronounced.

So far, we have been working under the assumption that our stochastic equations hold exactly over the forecast period. In reality, we would expect to see the same sort of errors occurring in the future as we have seen in history. We have also been ignoring the fact that some of the coef-



ficients in our equations are estimated, rather than fixed at known values. We may like to reflect this uncertainty about our coefficients in some way in the results from our model.

We can incorporate these features into our EViews model using stochastic simulation.

Up until now, we have thought of our model as forecasting a single point for each of our endogenous variables at each observation. As soon as we add uncertainty to the model, we should think instead of our model as predicting a whole distribution of outcomes for each variable at each observation. Our goal is to summarize these distributions using appropriate statistics.

If the model is linear (as in our example) and the errors are normal, then the endogenous variables will follow a normal distribution, and the mean and standard deviation of each distribution should be sufficient to describe the distribution completely. In this case, the mean will actually be equal to the deterministic solution to the model. If the model is not linear, then the distributions of the endogenous variables need not be normal. In this case, the quantiles of the distribution may be more informative than the first two moments, since the distributions may have tails which are very different from the normal case. In a non-linear model, the mean of the distribution need not match up to the deterministic solution of the model.

EViews makes it easy to calculate statistics to describe the distributions of your endogenous variables in an uncertain environment. To simulate the distributions, the model object uses a Monte Carlo approach, where the model is solved many times with pseudo-random numbers substituted for the unknown errors at each repetition. This method provides only approximate results. However, as the number of repetitions is increased, we would expect the results to approach their true values.

To return to our simple macroeconomic model, we can use a stochastic simulation to provide some measure of the uncertainty in our results by adding error bounds to our predictions. From the model window, click on the **Solve** button. When the model solution dialog appears, choose **Stochastic** for the simulation type. In the **Solution scenarios & output** box, make sure that the **Std. Dev.** checkbox in the **Active** section is checked. Click on **OK** to begin the simulation.

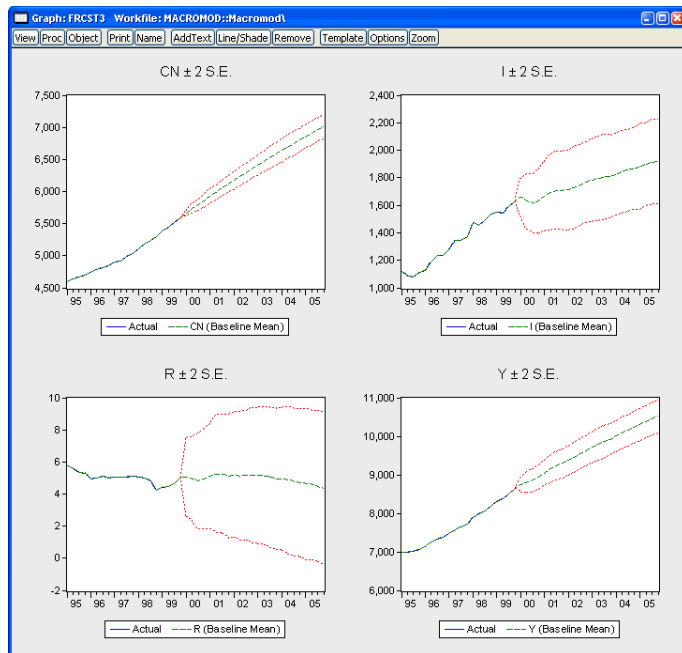
The simulation should take about half a minute. Status messages will appear to indicate progress through the repetitions. When the simulation is complete, you may return to the variable view, use the mouse to select the variables as discussed above, and then select **Proc/Make Graph...** When the **Make Graph** dialog appears, select the option **Mean + - 2 standard deviations** in the **Solution Series** list box in the **Graph Series** area on the right of the dialog. Set the sample to 1995Q1 to 2005Q4 and click on **OK**.

The error bounds in the resulting output graph show that we should be reluctant to place too much weight on the point forecasts of our model, since the bounds are quite wide on several of the variables. Much of the uncertainty is probably due to the large residual in the investment equation, which is creating a lot of variation in investment and interest rates in the stochastic simulation.

Another exercise we might like to consider when working with our model is to examine how the model behaves under alternative assumptions with respect to the exogenous variables. One approach to this would be to directly edit the exogenous series so that they contain the new values, and then resolve the model, overwriting any existing results. The problem with this approach is that it makes it awkward to manage the data and to compare the different sets of outcomes.

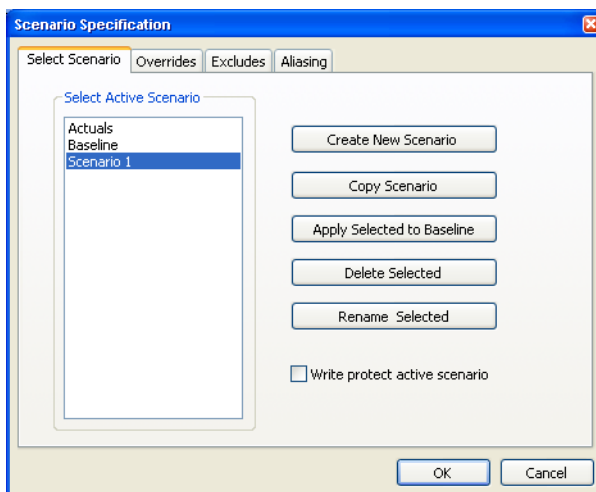
EViews provides a better way of carrying out exercises such as this through the use of model scenarios. Using a model scenario, you can override a subset of the exogenous variables in a model to give them new values, while using the values stored in the actual series for the remainder of the variables. When you solve for a scenario, the values of the endogenous variables are assigned into workfile series with an extension specific to that scenario, making it easy to keep multiple solutions for the model within a single workfile.

To create a scenario, we begin by selecting **View/Scenarios...** from the model object menus. The scenario specification dialog will appear with a list of the scenarios currently included in the model. There are two special scenarios that are always present in the model: **Actuals** and **Baseline**. These two scenarios are special in that they cannot contain any overridden variables.



They differ in that the actuals scenario writes its solution values directly into the workfile series with the same names as the endogenous variables, while the baseline scenario writes its solution values back into workfile series with the extension “_0”.

To add a new scenario to the model, simply click on the button labeled **Create New Scenario**. A new scenario will be created immediately. You can use this dialog to select which scenario is currently active, or to rename and delete scenarios.



Once we have created the scenario, we can modify the scenario from the baseline case by overriding one of our exogenous variables. To do this, return to the variable window of the model, click on the variable *M*, use the right mouse button to call up the **Properties** dialog for the variable, and then in the **Scenario** box, click on the checkbox for **Use override series in scenario**. A message will appear asking if you would like to create the new series. Click on **Yes** to create the series, then **OK** to return to the variable window.

In the variable window, the variable name “*M*” should now appear in red, indicating that it has been overridden in the active scenario. This means that the variable *M* will now be bound to the series *M_1* instead of the series *M* when solving the model. In our previous forecast for *M*, we assumed that the real money supply would be kept at a constant level during the forecast period. For our alternative scenario, we are going to assume that the real money supply is contracted sharply at the beginning of the forecast period, and held at this lower value throughout the forecast. We can set the new values using a few simple commands:

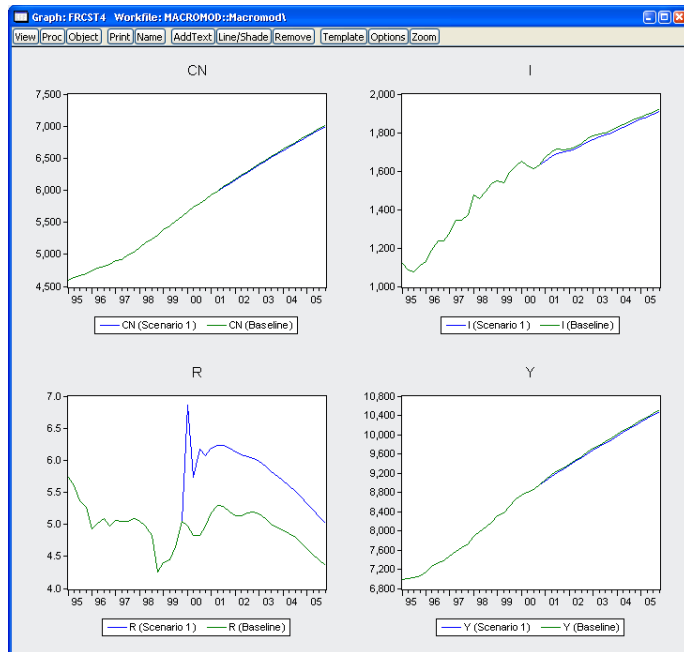
```
smpl 2000q1 2005q4
series m_1 = 900
smpl @all
```

As before, we can solve the model by clicking on the **Solve** button. Restore the **Simulation type** to deterministic, make sure that Scenario 1 is the active scenario, then click on **OK**. Once the solution is complete, we can use **Proc/Make Graph...** to display the results following the same procedure as above. Restore the **Solution series** list box to the setting **Deterministic solutions**, then check both the **Active** and **Compare** solution checkboxes below,

making sure that the active scenario is set to Scenario 1, and the comparison scenario is set to Baseline. Again set the sample to 1995Q1 to 2005Q4. The following graph should be displayed:

The simulation results suggest that the cut in the money supply causes a substantial increase in interest rates, which creates a small reduction in investment and a relatively minor drop in income and consumption. Overall, the predicted effects of changes in the money supply on the real economy are relatively minor in this model.

This concludes the discussion of our example model. The remainder of this chapter provides detailed information about working with particular features of the EViews model object.



Building a Model

Creating a Model

The first step in working with a model is to create the model object itself. There are several different ways of creating a model:

- You can create an empty model by using **Object/New Object...** and then choosing **Model**, or by performing the same operation using the right mouse button menu from inside the workfile window.
- You can select a list of estimation objects in the workfile window (equations, VARs, systems), and then select **Open as Model** from the right mouse button menu. This item will create a model which contains the equations from the selected objects as links.

- You can use the **Make model** procedure from an estimation object to create a model containing the equation or equations in that object.

Adding Equations to the Model

The equations in a model can be classified into two types: linked equations and inline equations. Linked equations are equations that import their specification from other objects in the workfile. Inline equations are contained inside the model as text.

There are a number of ways to add equations to your model:

- To add a linked equation: from the workfile window, select the object which contains the equation or equations you would like to add to the model, then copy-and-paste the object into the model equation view window.
- To add an equation using text: select **Insert...** from the right mouse button menu. In the text box titled: **Enter one or more lines...**, type in one or more equations in standard EViews format. You can also add linked equations from this dialog by typing a colon followed by the name of the object you would like to link to, for example “:EQ1”, because this is the text form of a linked object.



In an EViews model, the first variable that appears in an equation will be considered the endogenous variable for that equation. Since each endogenous variable can be associated with only one equation, you may need to rewrite your equations to ensure that each equation begins with a different variable. For example, say we have an equation in the model:

$$x / y = z$$

EViews will associate the equation with the variable X. If we would like the equation to be associated with the variable Y, we would have to rewrite the equation:

$$1 / y * x = z$$

Note that EViews has the ability to handle simple expressions involving the endogenous variable. You may use functions like LOG, D, and DLOG on the left-hand side of your equation. EViews will normalize the equation into explicit form if the Gauss-Seidel method is selected for solving the model.

Removing equations from the model

To remove equations from the model, simply select the equations using the mouse in Equation view, then use **Delete** from the right mouse button menu to remove the equations.

Both adding and removing equations from the model will change which variables are considered endogenous to the model.

Updating Links in the Model

If a model contains linked equations, changes to the specification of the equations made outside the model can cause the equations contained in the model to become out of date. You can incorporate these changes in the model by using **Proc/Link/Update All Links**. Alternatively, you can update just a single equation using the **Proc/Link/Update Link** item from the right mouse button menu. Links are also updated when a workfile is reloaded from disk.

Sometimes, you may want to sever equations in the model from their linked objects. For example, you may wish to see the entire model in text form, with all equations written in place. To do this, you can use the **Proc/Link/Break All Links** procedure to convert all linked equations in the model into inline text. You can convert just a single equation by selecting the equation, then using **Break Link** from the right mouse button menu.

When a link is broken, the equation is written in text form with the unknown coefficients replaced by their point estimates. Any information relating to uncertainty of the coefficients will be lost. This will have no effect on deterministic solutions to the model, but may alter the results of stochastic simulations if the **Include coefficient uncertainty** option has been selected.

Working with the Model Structure

As with other objects in EViews, we can look at the information contained in the model object in several ways. Since a model is a set of equations that describe the relationship between a set of variables, the two primary views of a model are the equation view and the variable view. EViews also provides two additional views of the structure of the model: the block view and the text view.

Equation View

The equation view is used for displaying, selecting, and modifying the equations contained in the model. An example of the equation view can be seen on [page 412](#).

Each line of the window is used to represent either a linked object or an inline text equation. Linked objects will appear similarly to how they do in the workfile, with an icon representing their type, followed by the name of the object. Even if the linked object contains many equations, it will use only one line in the view. Inline equations will appear with a “TXT” icon, followed by the beginning of the equation text in quotation marks.

The remainder of the line contains the equation number, followed by a symbolic representation of the equation, indicating which variables appear in the equation.

Any errors in the model will appear as red lines containing an error message describing the cause of the problem.

You can open any linked objects directly from the equation view. Simply select the line representing the object using the mouse, then choose **Open Link** from the right mouse button menu.

The contents of a line can be examined in more detail using the equation properties dialog. Simply select the line with the mouse, then choose **Properties...** from the right mouse button menu. Alternatively, simply double click on the object to call up the dialog.

For a link to a single equation, the dialog shows the functional form of the equation, the values of any estimated coefficients, and the standard error of the equation residual from the estimation. If the link is to an object containing many equations, you can move between the different equations imported from the object using the **Endogenous** list box at the top of the dialog. For an inline equation, the dialog simply shows the text of the equation.

Properties

Equation Endogenous Add Factors

Equation 1

Endogenous: I Link: EQI

Equation: EQI estimated on 01/19/04 - 21:52

$$i = @coef(1) + @coef(2) * (y(-1) - y(-2)) + @coef(3) * y + @coef(4) * r(-4)$$

@coef(1) = -110.46584
 @coef(2) = 0.5782885
 @coef(3) = 0.1778304
 @coef(4) = -8.5029116

Estimated S.E. = 57.404917

Edit Equation or Link Specification

Equation type

☒ Stochastic with S.D.: 57.404917
☐ Identity

OK Cancel

The **Edit Equation or Link Specification** button allows you to edit the text of an inline equation or to modify a link to point to an object with a different name. A link is represented in text form as a colon followed by the name of the object. Note that you cannot modify the specification of a linked object from within the model object, you must work directly with the linked object itself.

In the bottom right of the dialog, there are a set of fields that allow you to set the stochastic properties of the residual of the equation. If you are only performing deterministic simulations, then these settings will not affect your results in any way. If you are performing stochastic simulations, then these settings are used in conjunction with the solution options to determine the size of the random innovations applied to this equation.

The **Stochastic with S.D.** option for **Equation type** lets you set a standard deviation for any random innovations applied to the equation. If the standard deviation field is blank or is set to “NA”, then the standard deviation will be estimated from the historical data. The **Identity** option specifies that the selected equation is an identity, and should hold without error even in a stochastic simulation. See [“Stochastic Options” on page 442](#) below for further details.

The equation properties dialog also gives you access to the property dialogs for the endogenous variable and add factor associated with the equation. Simply click on the appropriate tab. These will be discussed in greater detail below.

Variable View

The variable view is used for adjusting options related to variables and for displaying and editing the series associated with the model (see the discussion in [“An Example Model”](#) (p. 414)). The variable view lists all the variables contained in the model, with each line representing one variable. Each line begins with an icon classifying the variable as endogenous, exogenous or an add factor. This is followed by the name of the variable, the equation number associated with the variable, and the description of the variable. The description is read from the associated series in the workfile.

Note that the names and types of the variables in the model are determined fully by the equations of the model. The only way to add a variable or to change the type of a variable in the model is to modify the model equations.

You can adjust what is displayed in the variable view in a number of ways. By clicking on the **Filter/Sort** button just above the variable list, you can choose to display only variables that match a certain name pattern, or to display the variables in a particular order. For example, sorting by type of variable makes the division into endogenous and exogenous variables clearer, while sorting by override highlights which variables have been overridden in the currently active scenario.

The variable view also allows you to browse through the dependencies between variables in the model by clicking on the **Dependencies** button. Each equation in the model can be thought of as a set of links that connect other variables in the model to the endogenous variable of the equation. Starting from any variable, we can travel up the links, showing all the endogenous variables that this variable directly feeds into, or we can travel down the links, showing all the variables upon which this variable directly depends. This may sometimes be useful when trying to find the cause of unexpected behavior. Note, however, that in a simultaneous model, every endogenous variable is indirectly connected to every other variable in the same block, so that it may be hard to understand the model as a whole by looking at any particular part.

You can quickly view or edit one or more of the series associated with a variable by double clicking on the variable. For several variables, simply select each of them with the mouse then double click inside the selected area.

Block Structure View

The block structure view of the model analyzes and displays any block structure in the dependencies of the model.

Block structure refers to whether the model can be split into a number of smaller parts, each of which can be solved for in sequence. For example, consider the system:

| | |
|---------|-----------------|
| block 1 | $x = y + 4$ |
| | $y = 2 * x - 3$ |
| block 2 | $z = x + y$ |

Because the variable Z does not appear in either of the first two equations, we can split this equation system into two blocks: a block containing the first two equations, and a block containing the third equation. We can use the first block to solve for the variables X and Y, then use the second block to solve for the variable Z. By using the block structure of the system, we can reduce the number of variables we must solve for at any one time. This typically improves performance when calculating solutions.

Blocks can be classified further into *recursive* and *simultaneous* blocks. A recursive block is one which can be written so that each equation contains only variables whose values have already been determined. A recursive block can be solved by a single evaluation of all the equations in the block. A simultaneous block cannot be written in a way that removes feedback between the variables, so it must be solved as a simultaneous system. In our example above, the first block is simultaneous, since X and Y must be solved for jointly, while the second block is recursive, since Z depends only on X and Y, which have already been determined in solving the first block.

The block structure view displays the structure of the model, labeling each of the blocks as recursive or simultaneous. EViews uses this block structure whenever the model is solved. The block structure of a model may also be interesting in its own right, since reducing the system to a set of smaller blocks can make the dependencies in the system easier to understand.

Text View

The text view of a model allows you to see the entire structure of the model in a single screen of text. This provides a quick way to input small models, or a way to edit larger models using copy-and-paste.

The text view consists of a series of lines. In a simple model, each line simply contains the text of one of the inline equations of the model. More complicated models may contain one of more of the following:

- A line beginning with a colon “:” represents a link to an external object. The colon must be followed by the name of an object in the workfile. Equations contained in the external object will be imported into the model whenever the model is opened, or when links are updated.

- A line beginning with “@ADD” specifies an add factor. The add factor command has the form:

```
@add(v) endogenous_name add_name
```

where `endogenous_name` is the name of the endogenous variable of the equation to which the add factor will be applied, and `add_name` is the name of the series. The option `(v)` is used to specify that the add factor should be applied to the endogenous variable. The default is to apply the add factor to the residual of the equation. See [“Using Add Factors” on page 432](#) for details.

- A line beginning with “@INNOV” specifies an innovation variance. The innovation variance has two forms. When applied to an endogenous variable it has the form:

```
@innov endogenous_name number
```

where `endogenous name` is the name of the endogenous variable and `number` is the standard deviation of the innovation to be applied during stochastic simulation. When applied to an exogenous variable, it has the form:

```
@innov exogenous_name number_or_series
```

where `exogenous name` is the name of the exogenous variable and `number_or_series` is either a number or the name of the series that contains the standard deviation to be applied to the variable during stochastic simulation. Note that when an equation in a model is linked to an external estimation object, the variance from the estimated equation will be brought into the model automatically and does not require an `@innov` specification unless you would like to modify its value.

- The keyword “@TRACE”, followed by the names of the endogenous variables that you wish to trace, may be used to request model solution diagnostics. See [“Diagnostics” on page 446](#).

Users of earlier versions of EViews should note that two commands that were previously available, `@assign` and `@exclude`, are no longer part of the text form of the model. These commands have been removed because they now address options that apply only to specific model scenarios rather than to the model as a whole. When loading in models created by earlier versions of EViews, these commands will be converted automatically into scenario options in the new model object.

Specifying Scenarios

When working with a model, you will often want to compare model predictions under a variety of different assumptions regarding the paths of your exogenous variables, or with one or more of your equations excluded from the model. Model scenarios allow you to do this without overwriting previous data or changing the structure of your model.

The most important function of a scenario is to specify which series will be used to hold the data associated with a particular solution of the model. To distinguish the data associated with different scenarios, each scenario modifies the names of the model variables according to an aliasing rule. Typically, aliasing will involve adding an underline followed by a number, such as “_0” or “_1” to the variable names of the model. The data for each scenario will be contained in series in the workfile with the aliased names.

Model scenarios support the analysis of different assumptions for exogenous variables by allowing you to override a set of variables you would like to alter. Exogenous variables which are overridden will draw their values from series with names aliased for that scenario, while exogenous variables which are not overridden will draw their values from series with the same name as the variable.

Scenarios also allow you to exclude one or more endogenous variables from the model. When an endogenous variable is excluded, the equation associated with that variable is dropped from the model and the value of the variable is taken directly from the workfile series with the same name. Excluding an endogenous variable effectively treats the variable as an exogenous variable for the purposes of solving the model.

When excluding an endogenous variable, you can specify a sample range over which the variable should be excluded. One use of this is to handle the case where more recent historical data is available for some of your endogenous variables than others. By excluding the variables for which you have data, your forecast can use actual data where possible, and results from the model where data are not yet available.

Each model can contain many scenarios. You can view the scenarios associated with the current model by choosing **View/Scenario Specification...** as shown above on [page 423](#).

There are two special scenarios associated with every model: actuals and baseline. These two scenarios have in common the special property that they cannot contain any overrides or excludes. They differ in that the actuals scenario writes the values for endogenous variables back into the series with the same name as the variables in the model, while the baseline scenario modifies the names. When solving the model using actuals as your active scenario, you should be careful not to accidentally overwrite your historical data.

The baseline scenario gets its name from the fact that it provides the base case from which other scenarios are constructed. Scenarios differ from the baseline by having one or more variables overridden or excluded. By comparing the results from another scenario against those of the baseline case, we can separate out the movements in the endogenous variables that are due to the changes made in that particular scenario from movements which are present in the baseline itself.

The **Select Scenario** page of the dialog allows you to select, create, copy, delete and rename the scenarios associated with the model. You may also apply the selected scenario to the baseline data, which involves copying the series associated with any overridden variables in

the selected scenario on top of the baseline values. Applying a scenario to the baseline is a way of committing to the edited values of the selected scenario making them a permanent part of the baseline case.

The **Scenario overrides** page provides a summary of variables which have been overridden in the selected scenario and equations which have been excluded. This is a useful way of seeing a complete list of all the changes which have been made to the scenario from the baseline case.

The **Aliasing** page allows you to examine the name aliasing rules associated with any scenario. The page displays the complete set of aliases that will be applied to the different types of variables in the model.

Although the scenario dialog lets you see all the settings for a scenario in one place, you will probably alter most scenario settings directly from the variable view instead. For both exogenous variables and add factors, you can select the variable from the variable view window, then use the right mouse button menu to call up the properties page for the variable. The override status of the variable can be adjusted using the **Use override** checkbox. Once a variable has been overridden, it will appear in red in the variable view.

Using Add Factors

Normally, when a model is solved deterministically, the equations of the model are solved so that each of the equations of the model is exactly satisfied. When a model is solved stochastically, random errors are added to each equation, but the random errors are still chosen so that their average value is zero.

If we have no information as to the errors in our stochastic equations that are likely to occur during the forecast period, then this behavior is appropriate. If, however, we have additional information as to the sort of errors that are likely during our forecast period, then we may incorporate that information into the model using add factors.

The most common use for add factors is to provide a smoother transition from historical data into the forecast period. Typically, add factors will be used to compensate for a poor fit of one or more equations of the model near the end of the historical data, when we suspect this will persist into the forecast period. Add factors provide an ad hoc way of trying to adjust the results of the model without respecifying or reestimating the equations of the model.

In reality, an add factor is just an extra exogenous variable which is included in the selected equation in a particular way. EViews allows an add factor to take one of two forms. If our equation has the form:

$$f(y_i) = f_i(y, x) \quad (36.3)$$

then we can provide an add factor for the equation intercept or residual by simply including the add factor at the end of the equation:

$$f(y_i) = f_i(y, x) + a \quad (36.4)$$

Alternatively, we may provide an add factor for the endogenous variable of the model by using the add factor as an offset:

$$f(y_i - a) = f_i(y, x) \quad (36.5)$$

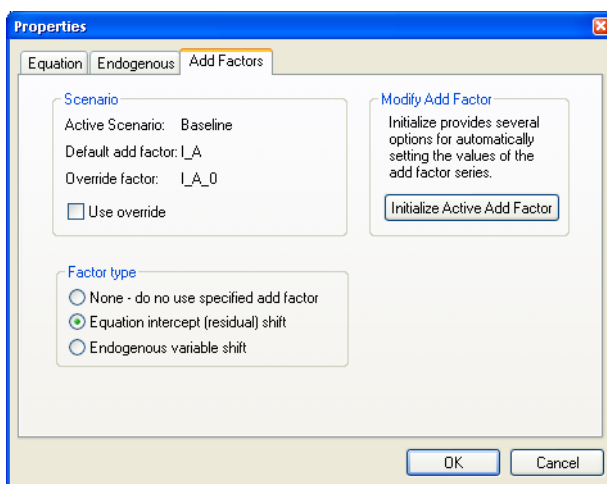
where the sign of the add factor is reversed so that it acts in the same direction as for the previous case.

If the endogenous variable appears by itself on the left hand side of the equal sign, then the two types of add factor are equivalent. If the endogenous variable is contained in an expression, for example, a log transformation, then this is no longer the case. Although the two add factors will have a similar effect, they will be expressed in different units with the former in the units of the residual of the equation, and the latter in the units of the endogenous variable of the equation.

There are two ways to include add factors. The easiest way is to go to the equation view of the model, then double click on the equation in which you would like to include an add factor.

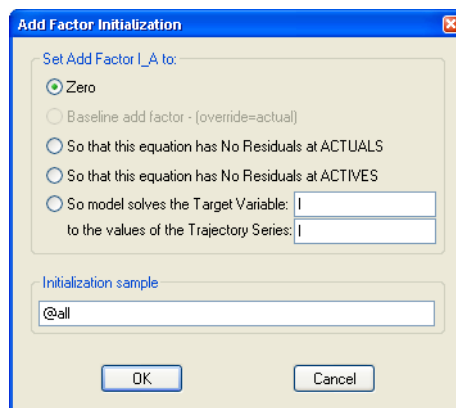
When the equation properties dialog appears, switch to the **Add Factors** tab. In the **Factor type** box, select whether you would like an intercept or an endogenous variable shift add factor. A message box will prompt for whether you would like to create a series in the workfile to hold the add factor values. Click on **Yes** to create the series.

The series will initially be filled with NAs. You can initialize the add factor using one of several methods by clicking on the **Initialize Add Factor** button.



A dialog box will come up offering the following options:

- **Zero:** set the add factor to zero for every period.
- **So that this equation has no residuals at actuals:** set the values of the add factor so that the equation is exactly satisfied without error when the variables of the model are set to the values contained in the actual series (typically the historical data).
- **So that this equation has no residuals at actives:** set the values of the add factor so that the equation is exactly satisfied without error when the variables of the model are set to the values contained in the endogenous and exogenous series associated with the active scenario.
- **So model solves the target variable to the values of the trajectory series:** set the values of the add factor so that an endogenous variable of the model follows a particular target path when the model is solved.



You can also change the sample over which you would like the add factor to be initialized by modifying the **Initialization sample** box. Click on OK to accept the settings.

Once an add factor has been added to an equation, it will appear in the variable view of the model as an additional variable. If an add factor is present in any scenario, then it must be present in every scenario, although the values of the add factor can be overridden for a particular scenario in the same way as for an exogenous variable.

The second way to handle add factors is to assign, initialize or override them for all the equations in the model at the same time using the **Proc/Add Factors** menu from the model window. For example, to create a complete set of add factors that make the model solve to actual values over history, we can use **Add Factors/Equation Assignment...** to create add factors for every equation, then use **Add Factors/Set Values...** to set the add factors so that all the equations have no residuals at the actual values.

When solving a model with an add factor, any missing values in the add factor will be treated as zeros.

Solving the Model

Once the model specification is complete, you can solve the model. EViews can perform both deterministic and stochastic simulations.

A deterministic simulation consists of the following steps:

- The block structure of the model is analyzed.
- The variables in the model are bound to series in the workfile, according to the override settings and name aliasing rules of the scenario that is being solved. If an endogenous variable is being tracked and a series does not already exist in the workfile, a new series will be created. If an endogenous variable is not being tracked, a temporary series will be created to hold the results.
- The equations of the model are solved for each observation in the solution sample, using an iterative algorithm to compute values for the endogenous variables.
- Any temporary series which were created are deleted.
- The results are rounded to their final values.

A stochastic simulation follows a similar sequence, with the following differences:

- When binding the variables, a temporary series is created for every endogenous variable in the model. Additional series in the workfile are used to hold the statistics for the tracked endogenous variables. If bounds are being calculated, extra memory is allocated as working space for intermediate results.
- The model is solved repeatedly for different draws of the stochastic components of the model. If coefficient uncertainty is included in the model, then a new set of coefficients is drawn before each repetition (note that coefficient uncertainty is ignored in nonlinear equations, or linear equations specified with PDL terms). During the repetition, errors are generated for each observation in accordance with the residual uncertainty and the exogenous variable uncertainty in the model. At the end of each repetition, the statistics for the tracked endogenous variables are updated to reflect the additional results.
- If a comparison is being performed with an alternate scenario, then the same set of random residuals and exogenous variable shocks are applied to both scenarios during each repetition. This is done so that the deviation between the two is based only on differences in the exogenous and excluded variables, not on differences in random errors.

Models Containing Future Values

So far, we have assumed that the structure of the model allows us to solve each period of the model in sequence. This will not be true in the case where the equations of the model contain future (as well as past) values of the endogenous variables.

Consider a model where the equations have the form:

$$F(y(-\text{maxlag}), \dots, y(-1), y, y(1), \dots, y(\text{maxlead}), x) = 0 \quad (36.6)$$

where F is the complete set of equations of the model, y is a vector of all the endogenous variables, x is a vector of all the exogenous variables, and the parentheses follow the usual EViews syntax to indicate leads and lags.

Since solving the model for any particular period requires both past and future values of the endogenous variables, it is not possible to solve the model recursively in one pass. Instead, the equations from all the periods across which the model will be solved must be treated as a simultaneous system, and we will require terminal as well as initial conditions. For example, in the case with a single lead and a single lag and a sample that runs from s to t , we must effectively solve the entire stacked system:

$$\begin{aligned}
 F(y_{s-1}, y_s, y_{s+1}, x) &= 0 \\
 F(y_s, y_{s+1}, y_{s+2}, x) &= 0 \\
 F(y_{s+1}, y_{s+2}, y_{s+3}, x) &= 0 \\
 &\dots \\
 F(y_{t-2}, y_{t-1}, y_t, x) &= 0 \\
 F(y_{t-1}, y_t, y_{t+1}, x) &= 0
 \end{aligned} \tag{36.7}$$

where the unknowns are y_s, y_{s+1}, \dots, y_t the initial conditions are given by y_{s-1} and the terminal conditions are used to determine y_{t+1} . Note that if the leads or lags extend more than one period, we will require multiple periods of initial or terminal conditions.

To solve models such as these, EViews applies a Gauss-Seidel iterative scheme across all the observations of the sample. Roughly speaking, this involves looping repeatedly through every observation in the forecast sample, at each observation solving the model while treating the past and future values as fixed, where the loop is repeated until changes in the values of the endogenous variables between successive iterations become less than a specified tolerance.

This method is often referred to as the Fair-Taylor method, although the Fair-Taylor algorithm includes a particular handling of terminal conditions (the extended path method) that is slightly different from the options provided by EViews. When solving the model, EViews allows the user to specify fixed end conditions by providing values for the endogenous variables beyond the end of the forecast sample, or to determine the terminal conditions endogenously by adding extra equations for the terminal periods which impose either a constant level, a linear trend, or a constant growth rate on the endogenous variables for values beyond the end of the forecast period.

Although this method is not guaranteed to converge, failure to converge is often a sign of the instability which results when the influence of the past or the future on the present does not die out as the length of time considered is increased. Such instability is often undesirable for other reasons and may indicate a poorly specified model.

Model Consistent Expectations

One source of models in which future values of endogenous variables may appear in equations are models of economic behavior in which expectations of future periods influence the decisions made in the current period. For example, when negotiating long term wage contracts, employers and employees must consider expected changes in prices over the duration of the contract. Similarly, when choosing to hold a security denominated in foreign currency, an individual must consider how the exchange rate is expected to change over the time that they hold the security.

Although the way that individuals form expectations is obviously complex, if the model being considered accurately captures the structure of the problem, we might expect the expectations of individuals to be broadly consistent with the outcomes predicted by the model. In the absence of any other information, we may choose to make this relationship hold exactly. Expectations of this form are often referred to as *model consistent expectations*.

If we assume that there is no uncertainty in the model, imposing model consistent expectations simply involves replacing any expectations that appear in the model with the future values predicted by the model. In EViews, we can simply write out the expectation terms that appear in equations using the lead operator. A deterministic simulation of the model can then be run using EViews ability to solve models with equations which contain future values of the endogenous variables.

When we add uncertainty to the model, the situation becomes more complex. In this case, instead of the expectations of agents being set equal to the single deterministic outcome predicted by the model, the expectations of agents should be calculated based on the entire distribution of stochastic outcomes predicted by the model. To run a stochastic simulation of a model involving expectations would require a procedure like the following:

1. Take an initial guess as to a path for expectations over the forecast period (for example, by calculating a solution for the expectations in the deterministic case)
2. Run a large number of stochastic repetitions of the model holding these expectations constant, calculating the mean paths of the endogenous variables over the entire set of outcomes.
3. Test if the mean paths of the endogenous variables are equal to the current guess of expectations within some tolerance. If not, replace the current guess of expectations with the mean of the endogenous variables obtained in step 2, and return to step 2.

At present, EViews does not have built in functionality for automatically carrying out this procedure. Because of this, EViews will not perform stochastic simulations if your model contains equations involving future values of endogenous variables. We hope to add this functionality to future revisions of EViews.

Models Containing MA Terms

Solving models with equations that contain MA terms requires that we first obtain fitted values for the equation innovations in the pre-forecast sample period. For example, to perform dynamic forecasting of the values of y , beginning in period S using a simple MA(q):

$$\hat{y}_S = \hat{\phi}_1 \epsilon_{S-1} + \dots + \hat{\phi}_q \epsilon_{S-q}, \quad (36.8)$$

you require values for the pre-forecast sample innovations, $\epsilon_{S-1}, \epsilon_{S-2}, \dots, \epsilon_{S-q}$. Similarly, constructing a static forecast for a given period will require estimates of the q lagged innovations at every period in the forecast sample.

Initialization Methods

If your equation was estimated with backcasting turned on, EViews will, by default, perform backcasting to obtain initial values for model solution. If your equation is estimated with backcasting turned off, or if the forecast sample precedes the estimation sample, the initial values will be set to zero.

You may examine the equation specification in the model to determine whether backcasting was employed in estimation. The specification will include either the expression “BACKCAST = ”, or “INITMA = ” followed by an observation identifier for the first period of the estimation sample. As one might guess, “BACKCAST = ” is used to indicate the use of backcasting in estimation; alternately, “INITMA = ” indicates that the pre-sample values were initialized with zeros.

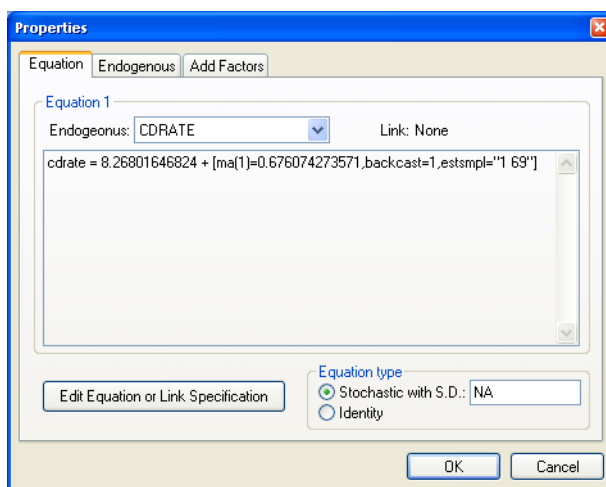
If neither “BACKCAST = ” nor “INITMA = ” is specified, the model will error when solved since EViews will be unable to obtain initial values for the forecast.

Here we see that the MA(1) equation for CDRATE in our model was estimated using “1 69” as the backcast estimation sample.

Backcast Methods

EViews offers alternate approaches for obtaining backcast estimates of the innovations when “BACKCAST = ” is specified.

The *estimation period* method uses data for the estimation sample to compute backcast estimates. The post-backcast sample innovations are initialized to zero and backward recursion



is employed to obtain estimates of the pre-estimation sample innovations. A forward recursion is then run to the end of the estimation sample and the resulting values are used as estimates of the innovations.

The alternative *forecast available* method offers different approaches for dynamic and static forecasting:

- For dynamic forecasting, EViews applies the backcasting procedure using data from the beginning of the estimation sample to either the beginning of the forecast period, or the end of the estimation sample, whichever comes first.
- For static forecasting, the backcasting procedure uses data from the beginning of the estimation sample to the end of the forecast period.

As before, the post-backcast sample innovations are set to zero and backward recursion is used to obtain estimates of the pre-estimation sample innovations, and forward recursion is used to obtain innovation estimates. Note that the forecast available method does not guarantee that the pre-sample forecast innovations match those employed in estimation.

See [“Forecasting with MA Errors” on page 128](#) for additional discussion.

The backcast initialization method employed by EViews for an equation in model solution depends on a variety of factors:

- For equations estimated using EViews 6 and later, the initialization method is determined from the equation specification. If the equation was estimated using estimation sample backcasting, its specification will contain “BACKCAST = ” and “ESTSMPL = ” statements instructing EViews to backcast using the specified sample.

The example dialog above shows an equation estimated using the estimation sample backcasting method.

- For equations estimated prior to EViews 6, the model will only contain the “BACKCAST = ” statement so that by default, the equation will be initialized using forecast available.
- In both cases, you may override the default settings by changing the specification of the equation in the model. To ensure that the equation backcasts using the forecast available method, simply delete the “ESTSMPL = ” portion of the equation specification. To force the estimation sample method for model solution, you may add an “ESTSMPL = ” statement to the equation specification.

Note that models containing EViews 6 equations solved in previous versions of EViews will always backcast using the forecast available method.

Basic Options

To begin solving a model, you can use **Proc/Solve Model...** or you can simply click on the **Solve** button on the model toolbar. EViews will display a tabbed dialog containing the solution options.

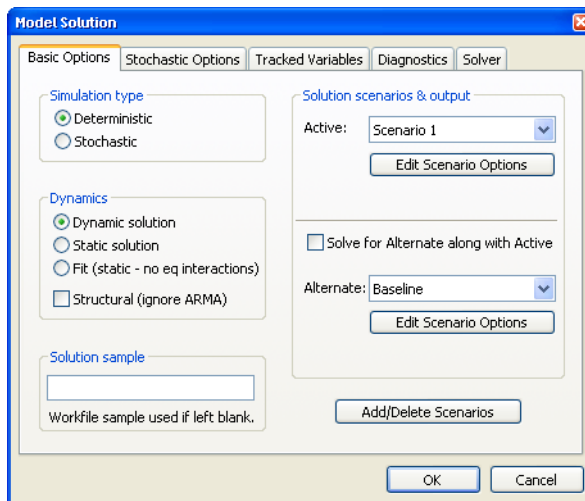
The basic options page contains the most important options for the simulation. While the options on other pages can often be left at their default values, the options on this page will need to be set appropriately for the task you are trying to perform.

At the top left, the **Simulation type** box allows you to determine whether the model should be simulated deterministically or stochastically. In a deterministic simulation, all equations in the model are solved so that they hold without error during the simulation period, all coefficients are held fixed at their point estimates, and all exogenous variables are held constant. This results in a single path for the endogenous variables which can be evaluated by solving the model once.

In a stochastic simulation, the equations of the model are solved so that they have residuals which match to randomly drawn errors, and, optionally, the coefficients and exogenous variables of the model are also varied randomly (see “[Stochastic Options](#)” on page 442 for details). For stochastic simulation, the model solution generates a distribution of outcomes for the endogenous variables in every period. We approximate the distribution by solving the model many times using different draws for the random components in the model then calculating statistics over all the different outcomes.

Typically, you will first analyze a model using deterministic simulation, and then later proceed to stochastic simulation to get an idea of the sensitivity of the results to various sorts of error. You should generally make sure that the model can be solved deterministically and is behaving as expected before trying a stochastic simulation, since stochastic simulation can be very time consuming.

The next option is the **Dynamics** box. This option determines how EViews uses historical data for the endogenous variables when solving the model:



- When **Dynamic solution** is chosen, only values of the endogenous variables from before the solution sample are used when forming the forecast. Lagged endogenous variables and ARMA terms in the model are calculated using the solutions calculated in previous periods, not from actual historical values. A dynamic solution is typically the correct method to use when forecasting values several periods into the future (a multi-step forecast), or evaluating how a multi-step forecast would have performed historically.
- When **Static solution** is chosen, values of the endogenous variables up to the previous period are used each time the model is solved. Lagged endogenous variables and ARMA terms in the model are based on actual values of the endogenous variables. A static solution is typically used to produce a set of one-step ahead forecasts over the historical data so as to examine the historical fit of the model. A static solution cannot be used to predict more than one observation into the future.
- When the **Fit** option is selected, values of the endogenous variables for the current period are used when the model is solved. All endogenous variables except the one variable for the equation being evaluated are replaced by their actual values. The fit option can be used to examine the fit of each of the equations in the model when considered separately, ignoring their interdependence in the model. The fit option can only be used for periods when historical values are available for all the endogenous variables.

In addition to these options, the **Structural** checkbox gives you the option of ignoring any ARMA specifications that appear in the equations of the model.

At the bottom left of the dialog is a box for the solution sample. The solution sample is the set of observations over which the model will be solved. Unlike in some other EViews procedures, the solution sample will not be contracted automatically to exclude missing data. For the solution to produce results, data must be available for all exogenous variables over the course of the solution sample. If you are carrying out a static solution or a fit, data must also be available for all endogenous variables during the solution sample. If you are performing a dynamic solution, only pre-sample values are needed to initialize any lagged endogenous or ARMA terms in the model.

On the right-hand side of the dialog are controls for selecting which scenarios we would like to solve. By clicking on one of the **Edit Scenario Options** buttons, you can quickly examine the settings of the selected scenario. The option **Solve for Alternate along with Active** should be used mainly in a stochastic setting, where the two scenarios must be solved together to ensure that the same set of random shocks is used in both cases. Whenever two models are solved together stochastically, a set of series will also be created containing the deviations between the scenarios (this is necessary because in a non-linear model, the difference of the means need not equal the mean of the differences).

When stochastic simulation has been selected, additional checkboxes are available for selecting which statistics you would like to calculate for your tracked endogenous variables. A series for the mean will always be calculated. You can also optionally collect series for the standard deviation or quantile bounds. Quantile bounds require considerable working memory, but are useful if you suspect that your endogenous variables may have skewed distributions or fat tails. If standard deviations or quantile bounds are chosen for either the active or alternate scenario, they will also be calculated for the deviations series.

Stochastic Options

The stochastic options page contains settings used during stochastic simulation. In many cases, you can leave these options at their default settings.

The **Repetitions** box, in the top left corner of the dialog, allows you to set the number of repetitions that will be performed during the stochastic simulation. A higher number of repetitions will reduce the sampling variation in the statistics being calculated, but will take more time. The default value of one thousand repetitions is generally adequate to get a good idea of the underlying values, although there may still be some random variation visible between adjacent observations.

The screenshot shows the 'Model Solution' dialog box with the 'Stochastic Options' tab selected. The 'Repetitions' section includes a 'Successes' field set to 1000, a '% Failed reps before halting' field set to 2, and a text area for 'Enter a name to save successful repetitions in a new WF page:'. Below this is a 'Page name:' field and a checkbox for 'Delete any existing page with same name'. The 'Confidence interval' section has two radio buttons: 'Calc from entire sample' (selected) and 'Reduced memory approx.', with an 'Interval size (2-sided):' field set to 0.95. The 'Innovation generation' section includes a 'Method:' dropdown set to 'Normal random numbers', a 'Covariance estimation sample:' field, a checkbox for 'Diagonal covariance matrix - no cross equation correlation', a checked checkbox for 'Scale variances to match equation specified innovation std. deviations', and a 'Multiply covariance matrix by:' field set to 1. There is also an unchecked checkbox for 'Include coefficient uncertainty'. 'OK' and 'Cancel' buttons are at the bottom right.

Also in the repetitions box is a field labeled **% Failed reps before halting**. Failed repetitions typically result from random errors driving the model into a region in which it is not defined, for example where the model is forced to take the log or square root of a negative number. When a repetition fails, EViews will discard any partial results from that repetition, then check whether the total number of failures exceeds the threshold set in the **% Failed reps before halting** box. The simulation continues until either this threshold is exceeded, or the target number of successful repetitions is met.

Note, however, that even one failed repetition indicates that care should be taken when interpreting the simulation results, since it indicates that the model is ill-defined for some possible draws of the random components. Simply discarding these extreme values may create misleading results, particularly when the tails of the distribution are used to measure the error bounds of the system.

The repetitions box also contains a field with the heading: **Enter a name to save successful repetitions in a new WF page**. If a name is provided, the values of the tracked endogenous variables for each successful repetition of the stochastic simulation will be copied into a new workfile page with the specified name. The new page is created with a panel structure where the values of the endogenous variables for individual repetitions are stacked on top of each other as cross sections within the panel. If the checkbox **Delete any existing page with the same name** is checked, any existing page with the specified page name will be deleted. If the checkbox is not checked, a number will be appended to the name of the new page so that it does not conflict with any existing page names.

The **Confidence interval** box sets options for how confidence intervals should be calculated, assuming they have been selected. The **Calc from entire sample** option uses the sample quantile as an estimate of the quantile of the underlying distribution. This involves storing complete tails for the observed outcomes. This can be very memory intensive since the memory used increases linearly in the number of repetitions. The **Reduced memory approx** option uses an updating algorithm due to Jain and Chlamtac (1985). This requires much less memory overall, and the amount used is independent of the number of repetitions. The updating algorithm should provide a reasonable estimate of the tails of the underlying distribution as long as the number of repetitions is not too small.

The **Interval size (2 sided)** box lets you select the size of the confidence interval given by the upper and lower bounds. The default size of 0.95 provides a 95 % confidence interval with a weight of 2.5 % in each tail. If, instead, you would like to calculate the interquartile range for the simulation results, you should input 0.5 to obtain a confidence interval with bounds at the 25 % and 75 % quantiles.

The **Innovation generation** box on the right side of the dialog determines how the innovations to stochastic equations will be generated. There are two basic methods available for generating the innovations. If **Method** is set to **Normal Random Numbers** the innovations will be generated by drawing a set of random numbers from the standard normal distribution. If **Method** is set to **Bootstrap** the innovations will be generated by drawing randomly (with replacement) from the set of actual innovations observed within a specified sample. Using bootstrapped innovations may be more appropriate than normal random numbers in cases where the equation innovations do not seem to follow a normal distribution, for example if the innovations appear asymmetric or appear to contain more outlying values than a normal distribution would suggest. Note, however, that a set of bootstrapped innovations

drawn from a small sample may provide only a rough approximation to the true underlying distribution of the innovations.

When normal random numbers are used, a set of independent random numbers are drawn from the standard normal distribution at each time period, then these numbers are scaled to match the desired variance-covariance structure of the system. In the general case, this involves multiplying the vector of random numbers by the Cholesky factor of the covariance matrix. If the matrix is diagonal, this reduces to multiplying each random number by its desired standard deviation.

Innovation generation

Method: Normal random numbers

Covariance estimation sample:

☐ Diagonal covariance matrix - no cross equation correlation

☒ Scale variances to match equation specified innovation std. deviations

Multiply covariance matrix by: 1

The **Scale variances to match equation specified standard deviations** box lets you determine how the variances of the residuals in the equations are determined. If the box is not checked, the variances are calculated from the model equation residuals. If the box is checked, then any equation that contains a specified standard deviation will use that number instead (see [page 427](#) for details on how to specify a standard deviation from the equation properties page). Note that the sample used for estimation in a linked equation may differ from the sample used when estimating the variances of the model residuals.

The **Diagonal covariance matrix** box lets you determine how the off diagonal elements of the covariance matrix are determined. If the box is checked, the off diagonal elements are set to zero. If the box is not checked, the off diagonal elements are set so that the correlation of the random draws matches the correlation of the observed equation residuals. If the variances are being scaled, this will involve rescaling the estimated covariances so that the correlations are maintained.

The **Estimation sample** box allows you to specify the set of observations that will be used when estimating the variance-covariance matrix of the model residuals. By default, EViews will use the default workfile sample.

The **Multiply covariance matrix** field allows you to set an overall scale factor to be applied to the entire covariance matrix. This can be useful for seeing how the stochastic behavior of the model changes as levels of random variation are applied which are different from those that were observed historically, or as a means of trouble-shooting the model by reducing the overall level of random variation if the model behaves badly.

When bootstrapped innovations are used, the dialog changes to show options available for bootstrapping. Similar options are available to those provided when using normal random numbers, although the meanings of the options are slightly different.

The field **Bootstrap residual draw sample** may be used to specify a sample period from which to draw the residuals used in the bootstrap procedure. If no sample is provided, the bootstrap sample will be set to include the set of observations from the start of the workfile to the last observation before the start of the solution sample. Note that if the bootstrap sample is different from the estimation sample for an equation, then the variance of the bootstrapped innovations need not match the variance of the innovations as estimated by the equation.

Innovation generation

Method: Bootstrap

Bootstrap residual draw sample:

☐ Diagonal covariance matrix - draw resid independently for each eqn

☒ Scale variances to match equation specified innovation std. deviations

Multiply bootstrap resids by: 1

The **Diagonal covariance matrix - draw resid independently for each equation** checkbox specifies whether each equation draws independently from a separate observation of the bootstrap sample, or whether a single observation is drawn from the bootstrap sample for all the equations in the model. If the innovation is drawn independently for each equation, there will be no correlation between the innovations used in the different equations in the model. If the same observation is used for all residuals, then the covariance of the innovations in the forecast period will match the covariance of the observed innovations within the bootstrap sample.

The **Multiply bootstrap resids by** option can be used to rescale all bootstrapped innovations by the specified factor before applying them to the equations. This can be useful for providing a broad adjustment to the overall level of uncertainty to be applied to the model, which can be useful for trouble-shooting if the model is producing errors during stochastic simulation. Note that multiplying the innovation by the specified factor causes the variance of the innovation to increase by the square of the factor, so this option has a slightly different meaning in the bootstrap case than when using normally distributed errors.

As noted above, stochastic simulation may include both coefficient uncertainty and exogenous variable uncertainty. There are very different ways methods of specifying these two types of uncertainty.

The **Include coefficient uncertainty** field at the bottom right of the **Stochastic Options** dialog specifies whether estimated coefficients in linked equations should be varied randomly during a stochastic simulation. When this option is selected, coefficients are randomly redrawn at the beginning of each repetition, using the coefficient variability in the estimated equation, if possible. This technique provides a method of incorporating uncertainty surrounding the true values of the coefficients into variation in our forecast results. Note that

☐ Include coefficient uncertainty

coefficient uncertainty is ignored in nonlinear equations and in linear equations estimated with PDL terms.

We emphasize that the dynamic behavior of a model may be altered considerably when the coefficients in the model are varied randomly. A model which is stable may become unstable, or a model which converges exponentially may develop cyclical oscillations. One consequence is that the standard errors from a stochastic simulation of a single equation may vary from the standard errors obtained when the same equation is forecast using the EViews equation object. This result arises since the equation object uses an analytic approach to calculating standard errors based on a local linear approximation that effectively imposes stationarity on the original equation.

To specify exogenous variable uncertainty, you must provide information about the variability of each relevant exogenous variable. First, display the model in *variable view* by selecting **View/Variables** or clicking on the **Variables** button in the toolbar. Next, select the exogenous variable in question, and right mouse click, select **Properties...**, and enter the exogenous variable variance in the resulting dialog. If you supply a positive value, EViews will incorporate exogenous variable uncertainty in the simulation; if the variance is not a valid value (negative or NA), the exogenous variable will be treated as deterministic.

Tracked Variables

The Tracked Variables page of the dialog lets you examine and modify which endogenous variables are being tracked by the model. When a variable is tracked, the results for that variable are saved in a series in the workfile after the simulation is complete. No results are saved for variables that are not tracked.

Tracking is most useful when working with large models, where keeping the results for every endogenous variable in the model would clutter the workfile and use up too much memory.

By default, all variables are tracked. You can switch on selective tracking using the radio button at the top of the dialog. Once selective tracking is selected, you can type in variable names in the dialog below, or use the properties dialog for the endogenous variable to switch tracking on and off.

You can also see which variables are currently being tracked using the variable view, since the names of tracked variables appear in blue.

Diagnostics

The **Diagnostics** dialog page lets you set options to control the display of intermediate output. This can be useful if you are having problems getting your model to solve.

When the **Display detailed messages** box is checked, extra output will be produced in the solution messages window as the model is solved.

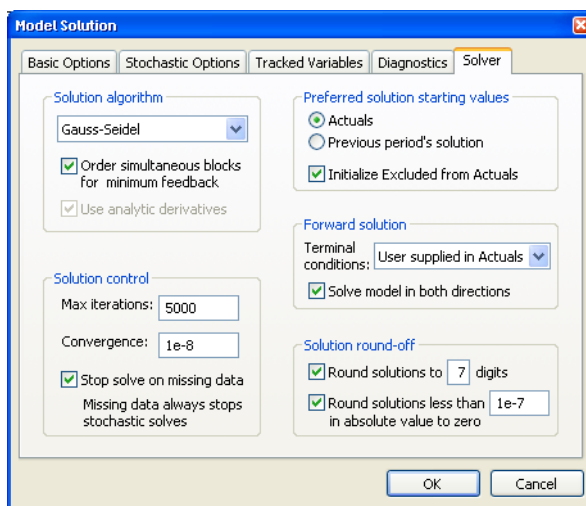
The traced variables list lets you specify a list of variables for which intermediate values will be stored during the iterations of the solution process. These results can be examined by switching to the **Trace Output** view after the model is complete. Tracing intermediate values may give you some idea of where to look for problems when a model is generating errors or failing to converge.

Solver

The **Solver** dialog page sets options relating to the non-linear equation solver which is applied to the model.

The **Solution algorithm** box lets you select the algorithm that will be used to solve the model for a single period. The following choices are available:

- Gauss-Seidel:** the Gauss-Seidel algorithm is an iterative algorithm, where at each iteration we solve each equation in the model for the value of its associated endogenous variable, treating all other endogenous variables as fixed. This algorithm requires little working memory and has fairly low computational costs, but requires the equation system to have certain stability properties for it to converge. Although it is easy to construct models that do not satisfy these properties, in practice, the algorithm generally performs well on most econometric models. If you are having difficulties with the algorithm, you might like to try reordering the equations, or rewriting the equations to change the assignment of endogenous variables, since these changes can affect the stability of the Gauss-Seidel iterations. (See “[Gauss-Seidel](#),” on page 633.)
- Newton:** Newton's method is also an iterative method, where at each iteration we take a linear approximation to the model, then solve the linear system to find a root of the model. This algorithm can handle a wider class of problems than Gauss-Seidel, but requires considerably more working memory and has a much greater computational cost when applied to large models. Newton's method is invariant to equation reordering or rewriting. (See “[Newton's Method](#),” on page 634.)
- Broyden:** Broyden's method is a modification of Newton's method (often referred to as a quasi-Newton or secant method) where an approximation to the Jacobian is used



when linearizing the model rather than the true Jacobian which is used in Newton's method. This approximation is updated at each iteration by comparing the equation residuals obtained at the new trial values of the endogenous variables with the equation residuals predicted by the linear model based on the current Jacobian approximation. Because each iteration in Broyden's method is based on less information than in Newton's method, Broyden's method typically requires more iterations to converge to a solution. Since each iteration will generally be cheaper to calculate, however, the total time required for solving a model by Broyden's method will often be less than that required to solve the model by Newton's method. Note that Broyden's method retains many of the desirable properties of Newton's method, such as being invariant to equation reordering or rewriting. (See [“Broyden's Method,”](#) on page 634.)

Note that even if Newton or Broyden's method is selected for solving within each period of the model, a Gauss-Seidel type method is used between all the periods if the model requires iterative forward solution. See [“Models Containing Future Values”](#) on page 435.

The **Excluded variables/Initialize from Actuals** checkbox controls where EViews takes values for excluded variables. By default, this box is checked and all excluded observations for solved endogenous variables (both in the solution sample and pre-solution observations) are initialized to the actual values of the endogenous variables prior to the start of a model solution. If this box is unchecked, EViews will initialize the excluded variables with values from the solution series (aliased series), so that you may set the values manually without editing the original series.

The **Order simultaneous blocks for minimum feedback** checkbox tells the solver to reorder the equations/variables within each simultaneous block in a way that will typically reduce the time required to solve the model. You should generally leave this box checked unless your model fails to converge, in which case you may want to see whether the same behavior occurs when the option is switched off.

The goal of the reordering is to separate a subset of the equations/variables of the simultaneous block into a subsystem which is recursive conditional on the values of the variables not included in the recursive subsystem. In mathematical notation, if F are the equations of the simultaneous block and y are the endogenous variables:

$$F(y, x) = 0 \quad (36.9)$$

the reordering is chosen to partition the system into two parts:

$$\begin{aligned} F_1(y_1, y_2, x) &= 0 \\ F_2(y_1, y_2, x) &= 0 \end{aligned} \quad (36.10)$$

where F has been partitioned into F_1 and F_2 and y has been partitioned into y_1 and y_2 .

The equations in F_1 are chosen so that they form a recursive system in the variables in the first partition, y_1 , conditional on the values or the variables in the second partition, y_2 . By a recursive system we mean that the first equation in F_1 may contain only the first element of y_1 , the second equation in F_1 may contain only the first and second elements of y_1 , and so on.

The reordering is chosen to make the first (recursive) partition as large as possible, or, equivalently, to make the second (feedback) partition as small as possible. Finding the best possible reordering is a time consuming problem for a large system, so EViews uses an algorithm proposed by Levy and Low (1988) to obtain a reordering which will generally be close to optimal, although it may not be the best of all possible reorderings. Note that in models containing hundreds of equations the recursive partition will often contain 90% or more of the equations/variables of the simultaneous block, with only 10% or less of the equations/variables placed in the feedback partition.

The reordering is used by the solution algorithms in a variety of ways.

- If the Gauss-Seidel algorithm is used, the basic operations performed by the algorithm are unchanged, but the equations are evaluated in the minimum feedback order instead of the order that they appear in the model. While for any particular model, either order could require less iterations to converge, in practice many models seem to converge faster when the equations are evaluated using the minimum feedback ordering.
- If the Newton solution algorithm is used, the reordering implies that the Jacobian matrix used in the Newton step has a bordered lower triangular structure (it has an upper left corner that is lower triangular). This structure is used inside the Newton solver to reduce the number of calculations required to find the solution to the linearized set of equations used by the Newton step.
- If the Broyden solution algorithm is used, the reordering is used to reduce the size of the equation system presented to the Broyden solver by using the equations of the recursive partition to 'substitute out' the variables of the recursive partition, producing a system which has only the feedback variables as unknowns. This more compact system of equations can generally be solved more quickly than the complete set of equations of the simultaneous block.

The **Use Analytic Derivatives** checkbox determines whether the solver will take analytic derivatives of the equations with respect to the endogenous variables within each simultaneous block when using solution methods that require the Jacobian matrix. If the box is not checked, derivatives will be obtained numerically. Analytic derivatives will often be faster to evaluate than numeric derivatives, but they will require more memory than numeric derivatives since an additional expression must be stored for each non-zero element of the Jacobian matrix. Analytic derivatives must also be recompiled each time the equations in the model are changed. Note that analytic derivatives will be discarded automatically if the

expression for the derivative is much larger than the expression for the original equation, as in this case the numeric derivative will be both faster to evaluate and require less memory.

The **Preferred solution starting values** section lets you select the values to be used as starting values in the iterative procedure. When **Actuals** is selected, EViews will first try to use values contained in the actuals series as starting values. If these are not available, EViews will try to use the values solved for in the previous period. If these are not available, EViews will default to using arbitrary starting values of 0.1. When **Previous period's solution** is selected, the order is changed so that the previous periods values are tried first, and only if they are not available, are the actuals used.

The **Solution control** section allows you to set termination options for the solver. **Max iterations** sets the maximum number of iterations that the solver will carry out before aborting. **Convergence** sets the threshold for the convergence test. If the largest relative change between iterations of any endogenous variable has an absolute value less than this threshold, then the solution is considered to have converged. **Stop on missing data** means that the solver should stop as soon as one or more exogenous (or lagged endogenous) variables is not available. If this option is not checked, the solver will proceed to subsequent periods, storing NAs for this period's results.

The **Forward solution** section allows you to adjust options that affect how the model is solved when one or more equations in the model contain future (forward) values of the endogenous variables. The **Terminal conditions** section lets you specify how the values of the endogenous variables are determined for leads that extend past the end of the forecast period. If **User supplied in Actuals** is selected, the values contained in the Actuals series after the end of the forecast sample will be used as fixed terminal values. If no values are available, the solver will be unable to proceed. If **Constant level** is selected, the terminal values are determined endogenously by adding the condition to the model that the values of the endogenous variables are constant over the post-forecast period at the same level as the final forecasted values ($y_t = y_{t-1}$ for $t = T, T+1, \dots, T+k-1$), where T is the first observation past the end of the forecast sample, and k is the maximum lead in the model). This option may be a good choice if the model converges to a stationary state. If **Constant difference** is selected, the terminal values are determined endogenously by adding the condition that the values of the endogenous variables follow a linear trend over the post forecast period, with a slope given by the difference between the last two forecasted values:

$$y_t - y_{t-1} = y_{t-1} - y_{t-2} \quad (36.11)$$

for $t = T, T+1, \dots, T+k-1$). This option may be a good choice if the model is in log form and tends to converge to a steady state. If **Constant growth rate** is selected, the terminal values are determined endogenously by adding the condition to the model that the endogenous variables grow exponentially over the post-forecast period, with the growth rate given by the growth between the final two forecasted values:

$$(y_t - y_{t-1})/y_{t-1} = (y_{t-1} - y_{t-2})/y_{t-2} \quad (36.12)$$

for $t = T, T + 1, \dots, T + k - 1$). This latter option may be a good choice if the model tends to produce forecasts for the endogenous variables which converge to constant growth paths.

The **Solve in both directions** option affects how the solver loops over periods when calculating forward solutions. When the box is not checked, the solver always proceeds from the beginning to the end of the forecast period during the Gauss-Seidel iterations. When the box is checked, the solver alternates between moving forwards and moving backwards through the forecast period. The two approaches will generally converge at slightly different rates depending on the level of forward or backward persistence in the model. You should choose whichever setting results in a lower iteration count for your particular model.

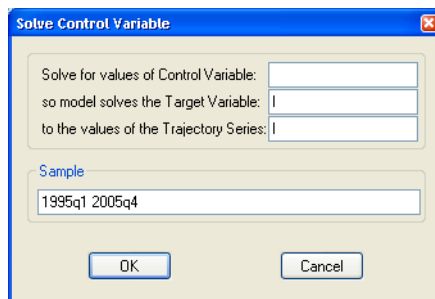
The **Solution round-off** section of the dialog controls how the results are rounded after convergence has been achieved. Because the solution algorithms are iterative and provide only approximate results to a specified tolerance, small variations can occur when comparing solutions from models, even when the results should be identical in theory. Rounding can be used to remove some of this minor variation so that results will be more consistent. The default settings will normally be adequate, but if your model has one or more endogenous variables of very small magnitude, you will need to switch off the rounding to zero or rescale the variables so that their solutions are farther from zero.

Solve Control for Target

Normally, when solving a model, we start with a set of known values for our exogenous variables, then solve for the unknown values of the endogenous variables of the model. If we would like an endogenous variable in our model to follow a particular path, we can solve the model repeatedly for different values of the exogenous variables, changing the values until the path we want for the endogenous variable is produced. For example, in a macroeconomic model, we may be interested in examining what value of the personal tax rate would be needed in each period to produce a balanced budget over the forecast horizon.

The problem with carrying out this procedure by hand is that the interactions between variables in the model make it difficult to guess the correct values for the exogenous variables. It will often require many attempts to find the values that solve the model to give the desired results.

To make this process easier, EViews provides a special procedure for solving a model which automatically searches for the unknown values. Simply create a series in the workfile which contains the values you would like the endogenous variable to achieve, then select **Proc/Solve Control for Target...** from the menus. Enter the name of the exogenous variable you would like to modify in the **Control Variable** box, the name of the endogenous variable which you are targeting in the **Target Variable** box, and the name of the workfile series which contains the target values in the **Trajectory Variable** box. Set the sample to the range for you would like to solve, then click on **OK**.



The procedure may take some time to complete, since it involves repeatedly solving the model to search for the desired solution. It is also possible for the procedure to fail if it cannot find a value of the exogenous variable for which the endogenous variable solves to the target value. If the procedure fails, you may like to try moving the trajectory series closer to values that you are sure the model can achieve.

Working with the Model Data

When working with a model, much of your time will be spent viewing and modifying the data associated with the model. Before solving the model, you will edit the paths of your exogenous variables or add factors during the forecast period. After solving the model, you will use graphs or tables of the endogenous variables to evaluate the results. Because there is a large amount of data associated with a model, you will also spend time simply managing the data.

Since all the data associated with a model is stored inside standard series in the workfile, you can use all of the usual tools in EViews to work with the data of your model. However, it is often more convenient to work directly from the model window.

Although there are some differences in details, working with the model data generally involves following the same basic steps. You will typically first use the variable view to select the set of variables you would like to work with, then use either the right mouse button menu or the model procedure menu to select the operation to perform.

Because there may be several series in the workfile associated with each variable in the model, you will then need to select the types of series with which you wish to work. The following types will generally be available:

- **Actuals:** the workfile series with the same name as the variable name. This will typically hold the historical data for the endogenous variables, and the historical data and baseline forecast for the exogenous variables.
- **Active:** the workfile series that is used when solving the active scenario. For endogenous variables, this will be the series with a name consisting of the variable name followed by the scenario extension. For exogenous variables, the actual series will be used unless it has been overridden. In this case, the exogenous variable will also be the workfile series formed by appending the scenario extension to the variable name.
- **Alternate:** the workfile series that is used when solving the alternate scenario. The rules are the same as for active.

In the following sections, we discuss how different operations can be performed on the model data from within the variable view.

Editing Data

The easiest way to make simple changes to the data associated with a model is to open a series or group spreadsheet window containing the data, then edit the data by hand.

To open a series window from within the model, simply select the variable using the mouse in the variable view, then use the right mouse button menu to choose **Open selected series...**, followed by **Actuals**, **Active Scenario** or **Alternate Scenario**. If you select several series before using the option, an unnamed group object will be created to hold all the series.

To edit the data, click the **Edit +/-** button to make sure the spreadsheet is in edit mode. You can either edit the data directly in levels or use the **Units** button to work with a transformed form of the data, such as the differences or percentage changes.

To create a group which allows you to edit more than one of the series associated with a variable at the same time, you can use the **Make Group/Table** procedure discussed below to create a dated data table, then switch the group to spreadsheet view to edit the data.

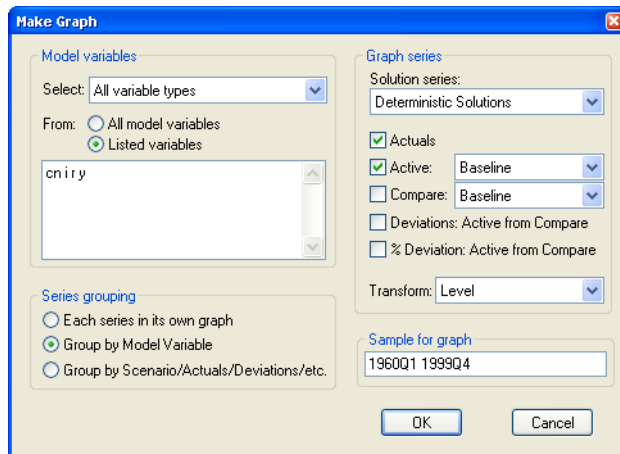
More complicated changes to the data may require using a **genr** command to calculate the series by specifying an expression. Click the **Genr** button from the series window toolbar to call up the dialog, then type in the expression to generate values for the series and set the workfile sample to the range of values you would like to modify.

Displaying Data

The EViews model object provides two main forms in which to display data: as a graph or as a table. Both of these can be generated easily from the model window.

From the variable view, select the variables you wish to display, then use the right mouse button menu or the main menu to select **Proc** and then **Make Group/Table** or **Make Graph**.

The dialogs for the two procs are almost identical. Here we see the **Make Graph** dialog. We saw this dialog earlier in our macro model example. The majority of fields in the dialog control which series



you would like the table or graph to contain. At the top left of the graph is the **Model Variables** box, which is used to select the set of variables to place in the graph. By default, the table or graph will contain the variables that are currently selected in the variable view. You can expand this to include all model variables, or add or remove particular variables from the list of selected variables using the radio buttons and text box labeled **From**. You can also restrict the set of variables chosen according to variable type using the drop down menu next to **Select**. By combining these fields, it is easy to select sets of variables such as all of the endogenous variables of the model, or all of the overridden variables.

Once the set of variables has been determined, it is necessary to map the variable names into the names of series in the workfile. This typically involves adding an extension to each name according to which scenario the data is from and the type of data contained in the series. The options affecting this are contained in the **Graph series** (if you are making a graph) or **Series types** (if you are making a group/table) box at the right of the dialog.

The **Solution series** box lets you choose which solution results you would like to examine when working with endogenous variables. You can choose from a variety of series generated during deterministic or stochastic simulations.

The series of checkboxes below determine which scenarios you would like to display in the graphs, as well as whether you would like to calculate deviations between various scenarios. You can choose to display the actual series, the series from the active scenario, or the series from an alternate scenario (labeled “Compare”). You can also display either the difference between the active and alternate scenario (labeled “Deviations: Active from Compare”), or the ratio between the active and alternate scenario in percentage terms (labeled “% Deviation: Active from Compare”).

The final field in the **Graph series** or **Series types** box is the **Transform** listbox. This lets you apply a transformation to the data similar to the **Transform** button in the series spreadsheet.

While the deviations and units options allow you to present a variety of transformations of your data, in some cases you may be interested in other transformations that are not directly available. Similarly, in a stochastic simulation, you may be interested in examining standard errors or confidence bounds on the transformed series, which will not be available when you apply transformations to the data after the simulation is complete. In either of these cases, it may be worth adding an identity to the model that generates the series you are interested in examining as part of the model solution.

For example, if your model contains a variable GDP, you may like to add a new equation to the model to calculate the percentage change of GDP:

$$pgdp = @pch(gdp)$$

After you have solved the model, you can use the variable PGDP to examine the percentage change in GDP, including examining the error bounds from a stochastic simulation. Note that the cost of adding such identities is relatively low, since EViews will place all such identities in a final recursive block which is evaluated only once after the main endogenous variables have already been solved.

The remaining option, at the bottom left of the dialog, lets you determine how the series will be grouped in the output. The options are slightly different for tables and graphs. For tables, you can choose to either place all series associated with the same model variable together, or to place each series of the same series type together. For graphs, you have the same two choices, and one additional choice, which is to place every series in its own graph.

In the graph dialog, you also have the option of setting a sample for the graph. This is often useful when you are plotting forecast results since it allows you to choose the amount of historical data to display in the graph prior to the forecast results. By default, the sample is set to the workfile sample.

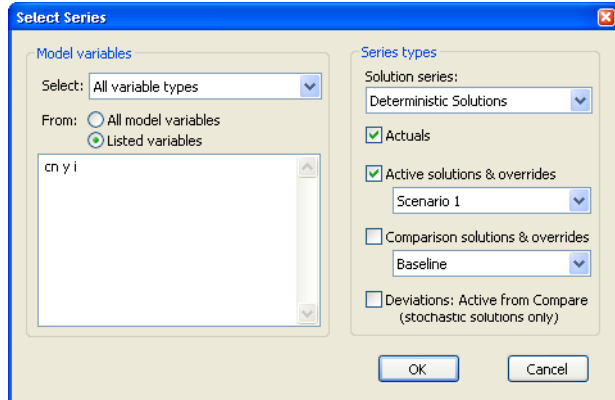
When you have finished setting the options, simply click on **OK** to create the new table or graph. All of EViews usual editing features are available to modify the table or graph for final presentation.

Managing Data

When working with a model, you will often create many series in the workfile for each variable, each containing different types of results or the data from different scenarios. The model object provides a number of tools to help you manage these series, allowing you to perform copy, fetch, store and delete operations directly from within the model.

Because the series names are related to the variable names in a consistent way, management tasks can often also be performed from outside the model by using the pattern matching features available in EViews commands (see [Appendix C. “Wildcards,”](#) on page 775 of the *User’s Guide I*).

The data management operations from within the model window proceed very similarly to the data display operations. First, select the variables you would like to work with from the variable view, then choose **Copy**, **Store series...**, **Fetch series...** or **Delete series...** from the right mouse button menu or the object procedures menu. A dialog will appear, similar to the one used when making a table or graph.



In the same way as for the table and graph dialogs, the left side of the dialog is used to choose which of the model variables to work with, while the right side of the dialog is used to select one or more series associated with each variable. Most of the choices are exactly the same as for graphs and tables. One significant difference is that the checkboxes for active and comparison scenarios include exogenous variables only if they have been overridden in the scenario. Unlike when displaying or editing the data, if an exogenous variable has not been overridden, the actual series will not be included in its place. The only way to store, fetch or delete any actual series is to use the **Actuals** checkbox.

After clicking on **OK**, you will receive the usual prompts for the store, fetch and delete operations. You can proceed as usual.

References

- Dennis, J. E. and R. B. Schnabel (1983). “Secant Methods for Systems of Nonlinear Equations,” *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. Prentice-Hall, London.
- Jain, Raj and Imrich Chlamtac (1985). “The P2 Algorithm for Dynamic Calculation of Quantiles and Histograms Without Storing Observations,” *Communications of the ACM*, 28(10), 1076–1085.
- Levy, Hanoach and David W. Low (1988). “A Contraction Algorithm for Finding Small Cycle Cutsets,” *Journal of Algorithms*, 9, 470–493.
- Pindyck, Robert S. and Daniel L. Rubinfeld (1998). *Econometric Models and Economic Forecasts*, 4th edition, New York: McGraw-Hill.

Part VII. Panel and Pooled Data

Panel and pool data involve observations that possess both cross-section, and within-cross-section identifiers.

Generally speaking, we distinguish between the two by noting that pooled time-series, cross-section data refer to data with relatively few cross-sections, where variables are held in cross-section specific individual series, while panel data correspond to data with large numbers of cross-sections, with variables held in single series in stacked form.

The discussion of these data is divided into parts. Pooled data structures are discussed first:

- [Chapter 37. “Pooled Time Series, Cross-Section Data,” on page 459](#) outlines tools for working with pooled time series, cross-section data, and estimating standard equation specifications which account for the pooled structure of the data.

Stacked panel data are described separately:

- In [Chapter 9. “Advanced Workfiles,” beginning on page 203](#) of the *User’s Guide I*, we describe the basics of structuring a workfile for use with panel data. Once a workfile is structured as a panel workfile, EViews provides you with different tools for working with data in the workfile, and for estimating equation specifications using both the data and the panel structure.
- [Chapter 38. “Working with Panel Data,” beginning on page 509](#), outlines the basics of working with panel workfiles.
- [Chapter 39. “Panel Estimation,” beginning on page 541](#) describes estimation in panel structured workfiles.

Chapter 37. Pooled Time Series, Cross-Section Data

Data often contain information on a relatively small number of cross-sectional units observed over time. For example, you may have time series data on GDP for a number of European nations. Or perhaps you have state level data on unemployment observed over time. We term such data *pooled* time series, cross-section data.

EViews provides a number of specialized tools to help you work with pooled data. EViews will help you manage your data, perform operations in either the time series or the cross-section dimension, and apply estimation methods that account for the pooled structure of your data.

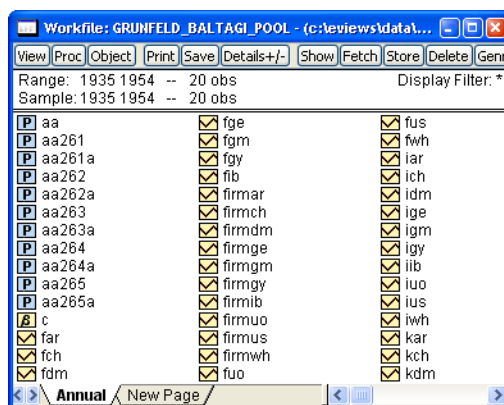
The EViews object that manages time series/cross-section data is called a *pool*. The remainder of this chapter will describe how to set up your data to work with pools, and how to define and work with pool objects.

Note that the data structures described in this chapter should be distinguished from data where there are large numbers of cross-sectional units. This type of data is typically termed *panel* data. Working with panel structured data in EViews is described in [Chapter 38](#). “Working with Panel Data,” on page 509 and [Chapter 39](#). “Panel Estimation,” beginning on page 541.

The Pool Workfile

The first step in working with pooled data is to set up a *pool workfile*. There are several characteristics of an EViews workfile that allow it to be used with pooled time series, cross-section data.

First, a pool workfile is an ordinary EViews workfile structured to match the time series dimension of your data. The range of your workfile should represent the earliest and latest dates or observations you wish to consider for any of the cross-section units. For example, if you want to work with data for some firms from 1932 to 1954, and data for other firms from 1930 to 1950, you should create a workfile ranging from 1930 to 1954.



Second, the pool workfile should contain EViews series that follow a user-defined naming convention. For each cross-section spe-

cific variable, you should have a separate series corresponding to each cross-section/variable combination. For example, if you have time series data for an economic variable like investment that differs for each of 10 firms, you should have 10 separate investment series in the workfile with names that follow the user-defined convention.

Lastly, and most importantly, a pool workfile must contain one or more *pool objects*, each of which contains a (possibly different) description of the pooled structure of your workfile in the form of rules specifying the user-defined naming convention for your series.

There are various approaches that you may use to set up your pool workfile:

- First, you may simply create a new workfile in the usual manner, by describing, the time series structure of your data. Once you have a workfile with the desired structure, you may define a pool object, and use this object as a tool in creating the series of interest and importing data into the series.
- Second, you may create an EViews workfile containing your data in stacked form. Once you have your stacked data, you may use the built-in workfile reshaping tools to create a workfile containing the desired structure and series.

Both of these procedures require a bit more background on the nature of the pool object, and the way that your pooled data are held in the workfile. We begin with a brief description of the basic components of the pool object, and then return to a description of the task of setting up your workfile and data ([“Setting up a Pool Workfile” on page 465](#)).

The Pool Object

Before describing the pooled workfile in greater detail, we must first provide a brief description of the EViews *pool object*.

We begin by noting that the pool object serves two distinct roles. First, the pool contains a set of definitions that describe the structure of the pooled time series, cross-section data in your workfile. In this role, the pool object serves as a tool for managing and working with pooled data, much like the group object serves is used as a tool for working with sets of series. Second, the pool provides procedures for estimating econometric models using pooled data, and examining and working with the results from this estimation. In this role, the pool object is analogous to an equation object that is used to estimate econometric specifications.

In this section, we focus on the definitions that serve as the foundation for the pool object and simple tools for managing your pool object. The tools for working with data are described in [“Working with Pooled Data,” beginning on page 472](#), and the role of the pool object in estimation is the focus of [“Pooled Estimation,” beginning on page 480](#).

Defining a Pool Object

There are two parts to the definitions in a pool object: the cross-section identifiers, and optionally, definitions of groups of identifiers.

Cross-section Identifiers

The central feature of a pool object is a list of cross-section members which provides a naming convention for series in the workfile. The entries in this list are termed *cross-section identifiers*. For example, in a cross-country study, you might use “_USA” to refer to the United States, “_KOR” to identify Korea, “_JPN” for Japan, and “_UK” for the United Kingdom. Since the cross-section identifiers will be used as a base in forming series names, we recommend that they be kept relatively short.

Specifying the list cross-section identifiers in a pool tells EViews about the structure of your data. When using a pool with the four cross-section identifiers given above, you instruct EViews to work with separate time series data for each of the four countries, and that the data may be held in series that contain the identifiers as part of the series names.

The most direct way of creating a pool object is to select **Object/New Object.../Pool**. EViews will open the pool specification view into which you should enter or copy-and-paste a list of identifiers, with individual entries separated by spaces, tabs, or carriage returns. Here, we have entered four identifiers on separate lines.

There are no special restrictions on the labels that you can use for cross-section identifiers, though you must be able to form legal EViews series names containing these identifiers.

Note that we have used the “_” character at the start of each of the identifiers in our list; this is not necessary, but you may find that it makes it easier to spot the identifier when it is used as the end of a series name.



Before moving on, it is important to note that a pool object is simply a description of the underlying structure of your data, so that it does not itself contain series or data. This separation of the object and the data has important consequences.

First, you may use pool objects to define multiple sets of cross-section identifiers. Suppose, for example, that the pool object POOL01 contains the definitions given above. You may also have a POOL02 that contains the identifiers “_GER,” “_AUS,” “_SWTZ,” and a POOL03 that contains the identifiers “_JPN” and “_KOR”. Each of these three pool objects defines a differ-

ent set of identifiers, and may be used to work with different sets of series in the workfile. Alternatively, you may have multiple pool objects in a workfile, each of which contain the same list of identifiers. A POOL04 that contains the same identifiers as POOL01 may be used to work with data from the same set of countries.

Second, since pool objects contain only definitions and not series data, deleting a pool will not delete underlying series data. You may, however, *use* a pool object to delete, create, and manipulate underlying series data.

Group Definitions

In addition to the main list of cross-section identifiers, you may define groups made up of subsets of your identifiers. To define a group of identifiers, you should enter the keyword “@GROUP” followed by a name for the group, and the subset of the pool identifiers that are to be used in the group. EViews will define a group using the specified name and any identifiers provided.

We may, for example, define the ASIA group containing the “_JPN” and “_KOR” identifiers, or the NORTHAMERICA group containing the “_USA” identifier by adding:

```
@group asia _jpn _kor
@group northamerica _usa
```

to the pool definition.

These subsets of cross-section identifiers may be used to define virtual series indicating whether a given observation corresponds to a given subgroup or not. The ASIA group, for example, can be used along with special tools to identify whether a given observation should be viewed as coming from Japan or Korea, or from one of the other countries in the pool. We describe this functionality in greater detail in [“Pool Series” on page 464](#).

Viewing or Editing Definitions

You may, at any time, change the view of an existing pool object to examine the current list of cross-section identifiers and group definitions. Simply push the **Define** button on the toolbar, or select **View/Cross-Section Identifiers**. If desired, you can edit the list of identifiers or group definitions.

Copying a Pool Object

Typically, you will work with more than one pool object. Multiple pools are used to define various subsamples of cross-section identifiers, or to work with different pooled estimation specifications.

To copy a pool object, open the original pool, and select **Object/Copy Object...** Alternatively, you can highlight the name of the pool in the workfile window, select **Object/Copy Selected...** and enter the new name.

Pooled Data

As noted previously, all of your pooled data will be held in ordinary EViews series. These series can be used in all of the usual ways: they may, among other things, be tabulated, graphed, used to generate new series, or used in estimation. You may also use a pool object to work with sets of the individual series.

There are two classes of series in a pooled workfile: *ordinary series* and *cross-section specific series*.

Ordinary Series

An ordinary series is one that has common values across all cross-sections. A single series may be used to hold the data for each variable, and these data may be applied to every cross-section. For example, in a pooled workfile with firm cross-section identifiers, data on overall economic conditions such as GDP or money supply do not vary across firms. You need only create a single series to hold the GDP data, and a single series to hold the money supply variable.

Since ordinary series do not interact with cross-sections, they may be defined without reference to a pool object. Most importantly, there are no naming conventions associated with ordinary series beyond those for ordinary EViews objects.

Cross-section Specific Series

Cross-section specific series are those that have values that differ between cross-sections. A set of these series are required to hold the data for a given variable, with each series corresponding to data for a specific cross-section.

Since cross-section specific series interact with cross-sections, they should be defined in conjunction with the identifiers in pool objects. Suppose, for example, that you have a pool object that contains the identifiers “_USA,” “_KOR,” “_JPN,” and “_UK”, and that you have time series data on GDP for each of the cross-section units. In this setting, you should have a four cross-section specific GDP series in your workfile.

The key to naming your cross-section specific series is to use names that are a combination of a *base name* and a cross-section identifier. The cross-section identifiers may be embedded at an arbitrary location in the series name, so long as this is done consistently across identifiers.

You may elect to place the identifier at the end of the base name, in which case, you should name your series “GDP_USA,” “GDP_KOR,” “GDP_JPN,” and “GDP_UK”. Alternatively, you may choose to put the section identifiers in front of the name, so that you have the names “_USAGDP,” “_KORGDP,” “_JPNGDP,” and “_UKGDP”. The identifiers may also be placed in the middle of series names—for example, using the names “GDP_USAINF,” “GDP_KORIN,” “GDP_JPNIN,” “GDP_UKIN”.

It really doesn't matter whether the identifiers are used at the beginning, middle, or end of your cross-section specific names; you should adopt a naming style that you find easiest to manage. Consistency in the naming of the set of cross-section series is, however, absolutely essential. You should not, for example, name your four GDP series "GDP_USA", "GDP_KOR", "_JPN_GDPIN", "_UKGDP", as this will make it impossible for EViews to refer to the set of series using a pool object.

Pool Series

Once your series names have been chosen to correspond with the identifiers in your pool, the pool object can be used to work with a set of series as though it were a single item. The key to this processing is the concept of a *pool series*.

A pool series is actually a set of series defined by a base name and the entire list of cross-section identifiers in a specified pool. Pool series are specified using the base name, and a "?" character placeholder for the cross-section identifier. If your series are named "GDP_USA", "GDP_KOR", "GDP_JPN", and "GDP_UK", the corresponding pool series may be referred to as "GDP?". If the names of your series are "_USAGDP", "_KORGDP", "_JPN_GDP", and "_UKGDP", the pool series is "?GDP".

When you use a pool series name, EViews understands that you wish to work with all of the series in the workfile that match the pool series specification. EViews loops through the list of cross-section identifiers in the specified pool, and substitutes each identifier in place of the "?". EViews then uses the complete set of cross-section specific series formed in this fashion.

In addition to pool series defined with "?", EViews provides a special function, @INGRP, that you may use to generate a group identity pool series that takes the value 1 if an observation is in the specified group, and 0 otherwise.

Consider, for example, the @GROUP for "ASIA" defined using the identifiers "_KOR" and "_JPN", and suppose that we wish to create a dummy variable series for whether an observation is in the group. One approach to representing these data is to create the following four cross-section specific series:

```
series asia_usa = 0
series asia_kor = 1
series asia_jpn = 1
series asia_uk = 0
```

and to refer to them collectively as the pool series "ASIA?". While not particularly difficult to do, this direct approach becomes more cumbersome the greater the number of cross-section identifiers.

More easily, we may use the special pool series expression:

```
@ingrp(asia)
```

to define a special virtual pool series in which each observation takes a 0 or 1 indicator for whether an observation is in the specified group. This expression is equivalent to creating the four cross-section specific series, and referring to them as “ASIA_?”.

We must emphasize that pool series specifiers using the “?” and the @INGRP function may only be used through a pool object, since they have no meaning without a list of cross-section identifiers. If you attempt to use a pool series outside the context of a pool object, EViews will attempt to interpret the “?” as a wildcard character (see [Appendix C. “Wildcards,” on page 775](#) in the *User’s Guide I*). The result, most often, will be an error message saying that your variable is not defined.

Setting up a Pool Workfile

Your goal in setting up a pool workfile is to obtain a workfile containing individual series for ordinary variables, sets of appropriately named series for the cross-section specific data, and pool objects containing the related sets of identifiers. The workfile should have frequency and range matching the time series dimension of your pooled data.

There are two basic approaches to setting up such a workfile. The direct approach involves first creating an empty workfile with the desired structure, and then importing data into individual series using either standard or pool specific import methods. The indirect approach involves first creating a stacked representation of the data in EViews, and then using EViews built-in reshaping tools to set up a pooled workfile.

Direct Setup

The direct approach to setting up your pool workfile involves three distinct steps: first creating a workfile with the desired time series structure; next, creating one or more pool objects containing the desired cross-section identifiers; and lastly, using pool object tools to import data into individual series in the workfile.

Creating the Workfile and Pool Object

The first step in the direct setup is to create an ordinary EViews workfile structured to match the time series dimension of your data. The range of your workfile should represent the earliest and latest dates or observations you wish to consider for any of the cross-section units.

Simply select **File/New workfile...** to bring up the **Workfile Create** dialog which you will use to describe the structure of your workfile. For additional detail, see [“Creating a Workfile by Describing its Structure” on page 39](#) of the *User’s Guide I*.

For example, to create a pool workfile that has annual data ranging from 1950 to 1992, simply select **Annual** in the **Frequency** combo box, and enter “1950” as the **Start date** and “1992” as the **End date**.

Next, you should create one or more pool objects containing cross-section identifiers and group definitions as described in [“The Pool Object” on page 460](#).

Importing Pooled Data

Lastly, you should use one of the various methods for importing data into series in the workfile. Before considering the various approaches, we require an understanding the various representations of pooled time series, cross-section data that you may encounter.

Bear in mind that in a pooled setting, a given observation on a variable may be indexed along three dimensions: the variable, the cross-section, and the time period. For example, you may be interested in the value of GDP, for the U.K., in 1989.

Despite the fact that there are three dimensions of interest, you will eventually find yourself working with a two-dimensional representation of your pooled data. There is obviously no unique way to organize three-dimensional data in two-dimensions, but several formats are commonly employed.

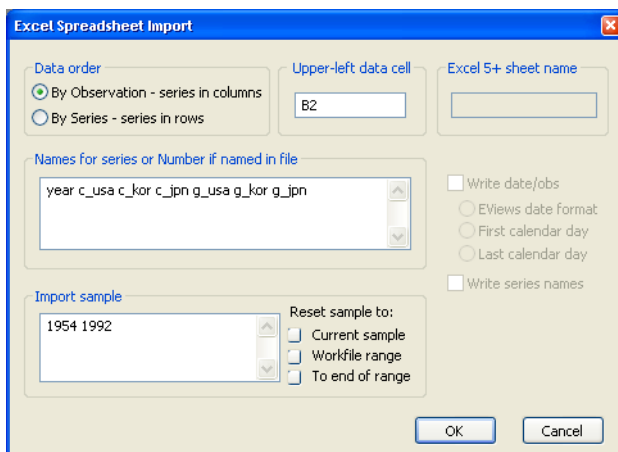
Unstacked Data

In this form, observations on a given variable for a given cross-section are grouped together, but are separated from observations for other variables and other cross sections. For example, suppose the top of our Excel data file contains the following:

| year | c_usa | c_kor | c_jpn | g_usa | g_jpn | g_kor |
|------|-------|-------|-------|-------|-------|-------|
| 1954 | 61.6 | 77.4 | 66 | 17.8 | 18.7 | 17.6 |
| 1955 | 61.1 | 79.2 | 65.7 | 15.8 | 17.1 | 16.9 |
| 1956 | 61.7 | 80.2 | 66.1 | 15.7 | 15.9 | 17.5 |
| 1957 | 62.4 | 78.6 | 65.5 | 16.3 | 14.8 | 16.3 |
| ... | ... | ... | ... | ... | ... | ... |

Here, the base name “C” represents consumption, while “G” represents government expenditure. Each country has its own separately identified column for consumption, and its own column for government expenditure.

EViews pooled workfiles are structured to work naturally with data that are unstacked, since the sets of cross-section specific series in the pool workfile correspond directly to the multiple columns of unstacked source data. You may read unstacked data directly into EViews using the standard import procedures described in [“Frequency Conversion” on page 106](#) of the *User’s Guide I*. Simply read each cross-section specific variable as an individual series, making certain that the names of the resulting series follow the pool naming conventions given in your pool object. Ordinary series may be imported in the usual fashion with no additional complications.



In this example, we use the standard EViews import tools to read separate series for each column. We create the individual series “YEAR”, “C_USA”, “C_KOR”, “C_JPN”, “G_USA”, “G_JPN”, and “G_KOR”.

Stacked Data

Pooled data can also be arranged in stacked form, where all of the data for a variable are grouped together in a single column.

In the most common form, the data for different cross-sections are stacked on top of one another, with all of the sequentially dated observations for a given cross-section grouped together. We may say that these data are *stacked by cross-section*:

| id | year | c | g |
|------|------|------|------|
| _usa | 1954 | 61.6 | 17.8 |
| _usa | ... | ... | ... |
| _usa | ... | ... | ... |
| _usa | 1992 | 68.1 | 13.2 |
| ... | ... | ... | ... |
| _kor | 1954 | 77.4 | 17.6 |
| _kor | ... | ... | ... |
| _kor | 1992 | na | na |

Alternatively, we may have data that are *stacked by date*, with all of the observations of a given period grouped together:

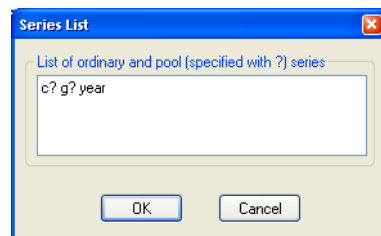
| per | id | c | g |
|------|------|------|------|
| 1954 | _usa | 61.6 | 17.8 |
| 1954 | _uk | 62.4 | 23.8 |
| 1954 | _jpn | 66 | 18.7 |
| 1954 | _kor | 77.4 | 17.6 |
| ... | ... | ... | ... |
| 1992 | _usa | 68.1 | 13.2 |
| 1992 | _uk | 67.9 | 17.3 |
| 1992 | _jpn | 54.2 | 7.6 |
| 1992 | _kor | na | na |

Each column again represents a single variable, but within each column, all of the cross-sections for a given year are grouped together. If data are stacked by year, you should make certain that the ordering of the cross-sectional identifiers within a year is consistent across years.

One straightforward method of importing data into your pool series is by manually entering into, or copying-and-pasting from and into, a stacked representation of your data. First, using the pool object, we will create the stacked representation of the data in EViews:

- First, specify which time series observations will be included in your stacked spreadsheet by setting the workfile sample.
- Next, open the pool, then select **View/Spreadsheet View...** EViews will prompt you for a list of series. You can enter ordinary series names or pool series names. If the series exist, then EViews will display the data in the series. If the series do not exist, then EViews will create the series or group of series, using the cross-section identifiers if you specify a pool series.
- EViews will open the stacked spreadsheet view of the pool series. If desired, click on the **Order** +/- button to toggle between stacking by cross-section and stacking by date.
- Click **Edit** +/- to turn on edit mode in the spreadsheet window, and enter your data, or cut-and-paste from another application.

For example, if we have a pool object that contains the identifiers “_USA”, “_UK”, “_JPN”, and “_KOR”, we can instruct EViews to create the series C_USA, C_UK, C_JPN, C_KOR, and G_USA, G_UK, G_JPN, G_KOR, and YEAR simply by entering the pool series names “C?”, “G?” and the ordinary series name “YEAR”, and pressing **OK**.



EViews will open a stacked spreadsheet view of the series in your list. Here we see the series stacked by cross-section, with the pool or ordinary series names in the column header, and the cross-section/date identifiers labeling each row. Note that since YEAR is an ordinary series, its values are repeated for each cross-section in the stacked spreadsheet.

If desired, click on **Order** +/- to toggle between stacking methods to match the organization of the data to be imported. Click on **Edit** +/- to turn on edit mode, and enter or cut-and-paste into the window.

| obs | C? | G? | YEAR |
|-----------|----|----|------|
| _USA-1950 | NA | NA | NA |
| _USA-1951 | NA | NA | NA |
| _USA-1952 | NA | NA | NA |
| _USA-1953 | NA | NA | NA |
| _USA-1954 | NA | NA | NA |
| _USA-1955 | NA | NA | NA |
| _USA-1956 | NA | NA | NA |
| _USA-1957 | NA | NA | NA |
| _USA-1958 | NA | NA | NA |
| _USA-1959 | NA | NA | NA |
| _USA-1960 | NA | NA | NA |
| _USA-1961 | NA | NA | NA |
| _USA-1962 | NA | NA | NA |

Alternatively, you can import stacked data from a file using import tools built into the pool object. While the data in the file may be stacked either by cross-section or by period, EViews does require that the stacked data are “balanced,” and that the cross-sections ordering in the file matches the cross-sectional identifiers in the pool. By “balanced,” we mean that if the data are stacked by cross-section, each cross-section should contain exactly the same number of periods—if the data are stacked by date, each date should have exactly the same number of cross-sectional observations arranged in the same order.

We emphasize that only the representation of the data in the import file needs to be balanced; *the underlying data need not be balanced*. Notably, if you have missing values for some observations, you should make certain that there are lines in the file representing the missing values. In the two examples above, the underlying data are not balanced, since information is not available for Korea in 1992. The data in the file have been balanced by including an observation for the missing data.

To import stacked pool data from a file, first open the pool object, then select **Proc/Import Pool data (ASCII, .XLS, .WK?)...** It is important that you use the import procedure associated with the pool object, and not the standard file import procedure.

Select your input file in the usual fashion. If you select a spreadsheet file, EViews will open a spreadsheet import dialog prompting you for additional input.

Much of this dialog should be familiar from the discussion in [Chapter 5. “Basic Data Handling,”](#) on page 77 of the *User’s Guide I*.

First, indicate whether the pool series are in rows or in columns, and whether the data are stacked by cross-section, or stacked by date.

Next, in the pool series edit box, enter the names of the series you wish to import.

This list may contain any combination of ordinary series names and pool series names.

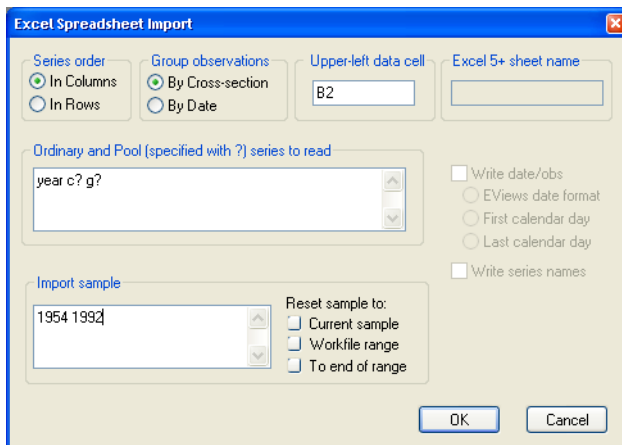
Lastly, fill in the sample information, starting cell location, and optionally, the sheet name.

When you specify your series using pool series names, EViews will, if necessary, create and name the corresponding set of pool series using the list of cross-section identifiers in the pool object. If you list an ordinary series name, EViews will, if needed, create a single series to hold the data.

EViews will read the contents of your file into the specified pool variables using the sample information. When reading into pool series, the first set of observations in the file will be placed in the individual series corresponding to the first cross-section (if reading data that is grouped by cross-section), or the first sample observation of each series in the set of cross-sectional series (if reading data that is grouped by date), and so forth.

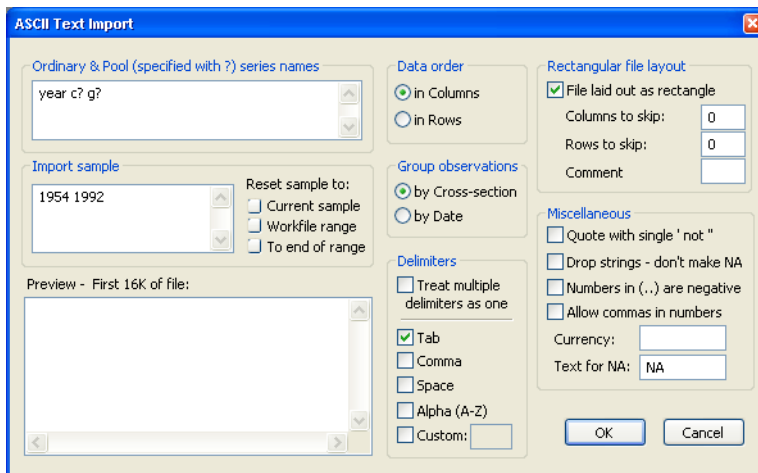
If you read data into an ordinary series, EViews will continually assign values into the corresponding observation of the single series, so that upon completion of the import procedure, the series will contain the last set of values read from the file.

The basic technique for importing stacked data from ASCII text files is analogous, but the corresponding dialog contains many additional options to handle the complexity of text files.



The dialog box is titled "Excel Spreadsheet Import". It contains several sections:

- Series order:** Radio buttons for "In Columns" (selected) and "In Rows".
- Group observations:** Radio buttons for "By Cross-section" (selected) and "By Date".
- Upper-left data cell:** A text box containing "B2".
- Excel 5+ sheet name:** An empty text box.
- Ordinary and Pool (specified with ?) series to read:** A list box containing "year c? g?".
- Import sample:** A list box containing "1954 1992".
- Reset sample to:** Checkboxes for "Current sample", "Workfile range", and "To end of range".
- Options:** Checkboxes for "Write date/obs", "EViews date format", "First calendar day", "Last calendar day", and "Write series names".
- Buttons:** "OK" and "Cancel" at the bottom right.



For a discussion of the text specific settings in the dialog, see [“Importing ASCII Text Files” on page 111](#) of the *User’s Guide I*.

Indirect Setup (Restructuring)

Second, you may create an ordinary EViews workfile containing your data in stacked form, and then use the workfile reshaping tools to create a pool workfile with the desired structure and contents.

The first step in the indirect setup of a pool workfile is to create a workfile containing the contents of your stacked data file. You may manually create the workfile and import the stacked series data, or you may use EViews tools for opening foreign source data directly into a new workfile ([“Creating a Workfile by Reading from a Foreign Data Source” on page 41](#) of the *User’s Guide I*).

Once you have your stacked data in an EViews workfile, you may use the workfile reshaping tools to unstack the data into a pool workfile page. In addition to unstacking the data into multiple series, EViews will create a pool object containing identifiers obtained from patterns in the series names. See [“Reshaping a Workfile,” beginning on page 237](#) of the *User’s Guide I* for a general discussion of reshaping, and [“Unstacking a Workfile” on page 240](#) of the *User’s Guide I* for a more specific discussion of the unstack procedure.

The indirect method is almost always easier to use than the direct approach and has the advantage of not requiring that the stacked data be balanced. It has the disadvantage of using more computer memory since EViews must have two copies of the source data in memory at the same time.

Working with Pooled Data

The underlying series for each cross-section member are ordinary series, so all of the EViews tools for working with the individual cross-section series are available. In addition, EViews provides you with a number of specialized tools which allow you to work with your pool data. Using EViews, you can perform, in a single step, similar operations on all the series corresponding to a particular pooled variable.

Generating Pooled Data

You can generate or modify pool series using the pool series **genr** procedure. Click on **Pool-Genr** on the pool toolbar and enter a formula as you would for a regular **genr**, using pool series names as appropriate. Using our example from above, entering:

```
ratio? = g?/g_usa
```

is equivalent to entering the following four commands:

```
ratio_usa = g_usa/g_usa
```

```
ratio_uk = g_uk/g_usa
```

```
ratio_jpn = g_jpn/g_usa
```

```
ratio_kor = g_kor/g_usa
```

Generation of a pool series applies the formula you supply using an implicit loop across cross-section identifiers, creating or modifying one or more series as appropriate.

You may use pool and ordinary **genr** together to generate new pool variables. For example, to create a dummy variable that is equal to 1 for the US and 0 for all other countries, first select **PoolGenr** and enter:

```
dum? = 0
```

to initialize all four of the dummy variable series to 0. Then, to set the US values to 1, select **Quick/Generate Series...** from the main menu, and enter:

```
dum_usa = 1
```

It is worth pointing out that a superior method of creating this pool series is to use **@GROUP** to define a group called US containing only the “_USA” identifier (see [“Group Definitions” on page 462](#)), then to use the **@INGRP** function:

```
dum? = @ingrp(us)
```

to generate and implicitly refer to the four series (see [“Pool Series” on page 464](#)).

To modify a set of series using a pool, select **PoolGenr**, and enter the new pool series expression:

```
dum? = dum? * (g? > c?)
```

It is worth the reminder that the method used by the pool `genr` is to perform an implicit loop across the cross-section identifiers. This implicit loop may be exploited in various ways, for example, to perform calculations across cross-sectional units in a given period. Suppose, we have an *ordinary* series `SUM` which is initialized to zero. The pool `genr` expression:

```
sum = sum + c?
```

is equivalent to the following four ordinary `genr` statements:

```
sum = sum + c_usa
```

```
sum = sum + c_uk
```

```
sum = sum + c_jpn
```

```
sum = sum + c_kor
```

Bear in mind that this example is provided merely to illustrate the notion of implicit looping, since EViews provides built-in features to compute period-specific statistics.

Examining Your Data

Pool workfiles provide you with the flexibility to examine cross-section specific series as individual time series or as part of a larger set of series.

Examining Unstacked Data

Simply open an individual series and work with it using the standard tools available for examining a series object. Or create a group of series and work with the tools for a group object. One convenient way to create groups of series is to use tools for creating groups out of pool and ordinary series; another is to use wildcards expressions in forming the group.

Examining Stacked Data

As demonstrated in [“Stacked Data,” beginning on page 467](#), you may use your pool object to view your data in stacked spreadsheet form. Select **View/Spreadsheet View...**, and list the series you wish to display. The names can include both ordinary and pool series names. Click on the **Order** \pm button to toggle between stacking your observations by cross-section and by date.

We emphasize that stacking your data only provides an alternative view of the data, and does not change the structure of the individual series in your workfile. Stacking data is not necessary for any of the data management or estimation procedures described below.

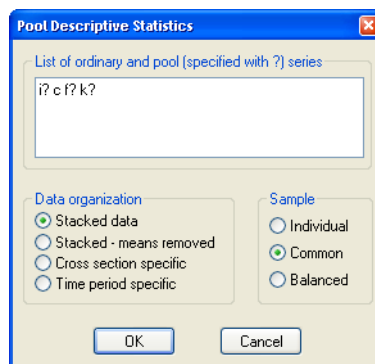
Calculating Descriptive Statistics

EViews provides convenient built-in features for computing various descriptive statistics for pool series using a pool object. To display the **Pool Descriptive Statistics** dialog, select **View/Descriptive Statistics...** from the pool toolbar.

In the edit box, you should list the ordinary and pooled series for which you want to compute the descriptive statistics. EViews will compute the mean, median, minimum, maximum, standard deviation, skewness, kurtosis, and the Jarque-Bera statistic for these series.

First, you should choose between the three sample options on the right of the dialog:

- **Individual:** uses the maximum number of observations available. If an observation on a variable is available for a particular cross-section, it is used in computation.
- **Common:** uses an observation only if data on the variable are available for all cross-sections in the same period. This method is equivalent to performing listwise exclusion by variable, then cross-sectional casewise exclusion within each variable.
- **Balanced:** includes observations when data on all variables in the list are available for all cross-sections in the same period. The balanced option performs casewise exclusion by both variable and cross-section.



Next, you should choose the computational method corresponding to one of the four data structures:

- **Stacked data:** display statistics for each variable in the list, computed over all cross-sections and periods. These are the descriptive statistics that you would get if you ignored the pooled nature of the data, stacked the data, and computed descriptive statistics.
- **Stacked – means removed:** compute statistics for each variable in the list after removing the cross-sectional means, taken over all cross-sections and periods.
- **Cross-section specific:** show the descriptive statistics for each cross-sectional variable, computed across all periods. These are the descriptive statistics derived by computing statistics for the individual series.
- **Time period specific:** compute period-specific statistics. For each period, compute the statistic using data on the variable from all the cross-sectional units in the pool.

Click on **OK**, and EViews will display a pool view containing tabular output with the requested statistics. If you select **Stacked data** or **Stacked - means removed**, the view will show a single column containing the descriptive statistics for each ordinary and pool series in the list, computed from the stacked data. If you select **Cross-section specific**, EViews will show a single column for each ordinary series, and multiple columns for each pool series. If

you select **Time period specific**, the view will show a single column for each ordinary or pool series statistic, with each row of the column corresponding to a period in the workfile. Note that there will be a separate column for each statistic computed for an ordinary or pool series; a column for the mean, a column for the variance, *etc.*

You should be aware that the latter two methods may produce a great deal of output. Cross-section specific computation generates a set of statistics for each pool series/cross-section combination. If you ask for statistics for three pool series and there are 20 cross-sections in your pool, EViews will display 60 columns of descriptive statistics. For time period specific computation, EViews computes a set of statistics for each date/series combination. If you have a sample with 100 periods and you provide a list of three pool series, EViews will compute and display a view with columns corresponding to 3 sets of statistics, each of which contains values for 100 periods.

If you wish to compute period-specific statistics, you may alternatively elect to save the results in series objects. See [“Making Period Stats” on page 477](#).

Computing Unit Root Tests

EViews provides convenient tools for computing multiple-series unit root tests for pooled data using a pool object. You may use the pool to compute one or more of the following types of unit root tests: Levin, Lin and Chu (2002), Breitung (2000), Im, Pesaran and Shin (2003), Fisher-type tests using ADF and PP tests—Maddala and Wu (1999) and Choi (2001), and Hadri (2000).

To compute the unit root test, select **View/Unit root test...** from the menu of a pool object.

Enter the name of an ordinary or pool series in the topmost edit field, then specify the remaining settings in the dialog.

These tests, along with the settings in the dialog, are described in [“Panel Unit Root Tests” on page 100](#).

Performing Cointegration Tests

Panel cointegration tests are available for pooled data using a pool object. EViews allows you to conduct several different

Unit Root Test

Pool series:

Test type:

Test for unit root in:

- ☒ Level
- ☐ 1st difference
- ☐ 2nd difference

Include in test equation:

- ☒ Individual intercept
- ☐ Individual trend and intercept
- ☐ None

Options:

- ☐ Use balanced sample

Lag length:

- ☒ Automatic selection:
 - Schwarz
 - Maximum lags: (* - indicates automatic selection of maximum lags)
- ☐ User specified:

Spectral estimation:

- Kernel method:
- Bandwidth selection:
 - ☒ Automatic:
 - ☐ User specified:

OK Cancel

tests: Pedroni (1999, 2004), Kao (1999) and Fisher-type test using Johansen’s test methodology (Maddala and Wu (1999)).

To compute the panel cointegration test for pooled data, select **Views/Cointegration Test...** from the menu of a pool object. Enter the names of at least two pool series or a combination of at least two pool and ordinary series in the topmost **Variables** field, then specify the rest of the options.

The remaining options are identical to those encountered when performing panel cointegration testing using a group in a panel-structured workfile. For details, see [“Panel Cointegration Testing,” beginning on page 372](#).

Panel Cointegration Test

Variables
IVM? MM? X Enter at least two pool or ordinary series to test

Test type
Pedroni (Engle-Granger based)

Deterministic trend specification
☒ Individual intercept
☐ Individual intercept and individual trend
☐ No intercept or trend

Lag length
☐ Automatic selection: Schwarz
 Maximum lag:
(Space indicates automatic selection of maximum lags)
☒ User specified:

Spectral estimation
 Kernel method: Bartlett
 Bandwidth selection
☒ Newey-West automatic
☐ User specified:

OK Cancel

In this example, specify two pool variables “IVM?” and “MM?” and one ordinary variable “X”, so that EViews tests for cointegration between the pool series IVM? against pool series MM? and the stacked common series X.

Making a Group of Pool Series

If you click on **Proc/Make Group...** and enter the names of ordinary and pool series. EViews will use the pool definitions to create an untitled group object containing the specified series. This procedure is useful when you wish to work with a set of pool series using the tools provided for groups.

Suppose, for example, that you wish to compute the covariance matrix for the C? series. Simply open the **Make Group** dialog, and enter the pool series name C?. EViews will create a group containing the set of cross-section specific series, with names beginning with “C” and ending with a cross-section identifier.

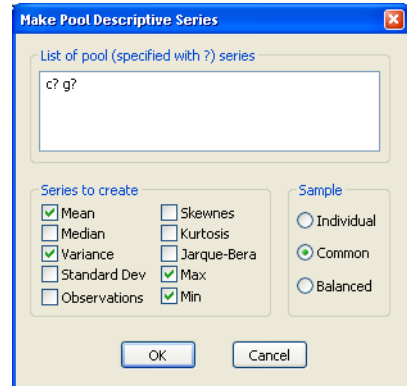
Then, in the new group object, you may select **View/Covariances** and either **Common Sample** or **Pairwise Sample** to describe the handling of missing values. EViews will display a view of the covariance matrix for all of the individual series in the group. Each element represents a covariance between individual members of the set of series in the pool series C?.

Making Period Stats

To save period-specific statistics in series in the workfile, select **Proc/Make Period Stats Series...** from the pool window, and fill out the dialog.

In the edit window, list the series for which you wish to calculate period-statistics. Next, select the particular statistics you wish to compute, and choose a sample option.

EViews will save your statistics in new series and will open an untitled group window to display the results. The series will be named automatically using the base name followed by the name of the statistic (MEAN, MED, VAR, SD, OBS, SKEW, KURT, JARQ, MAX, MIN). In this example, EViews will save the statistics using the names CMEAN, GMEAN, CVAR, GVAR, CMAX, GMAX, CMIN, and GMIN.



Making a System

Suppose that you wish to estimate a complex specification that cannot easily be estimated using the built-in features of the pool object. For example, you may wish to estimate a pooled equation imposing arbitrary coefficient restrictions, or using specialized GMM techniques that are not available in pooled estimation.

In these circumstances, you may use the pool to create a system object using both common and cross-section specific coefficients, AR terms, and instruments. The resulting system object may then be further customized, and estimated using all of the techniques available for system estimation.

Select **Proc/Make System...** and fill out the dialog. You may enter the dependent variable, common and cross-section specific variables, and use the checkbox to allow for cross-sectional fixed effects. You may also enter a list of common and cross-section specific instrumental variables, and instruct EViews to add lagged dependent and independent regressors as instruments in models with AR specifications.

When you click on **OK**, EViews will take your specification and create a new system object containing a single equation for each cross-section, using the specification provided.

Deleting/Storing/Fetching Pool Data

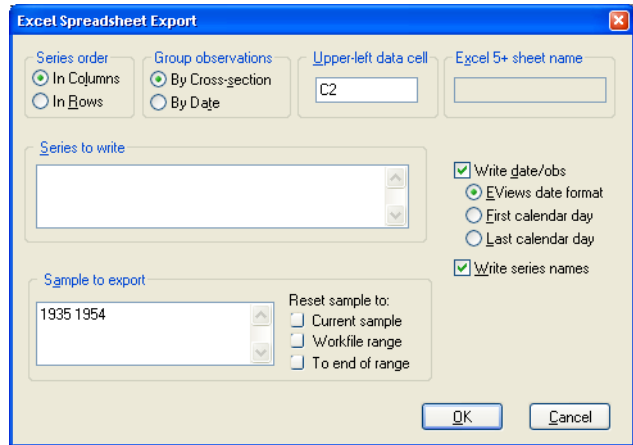
Pools may be used to delete, store, or fetch sets of series. Simply select **Proc/Delete pool series...**, **Proc/Store pool series (DB)...**, or **Proc/Fetch pool series (DB)...** as appropriate, and enter the ordinary and pool series names of interest.

If, for example, you instruct EViews to delete the pool series C?, EViews will loop through all of the cross-section identifiers and delete all series whose names begin with the letter “C” and end with the cross-section identifier.

Exporting Pooled Data

You can export your data into a disk file, or into a new workfile or workfile page, by reversing one of the procedures described above for data input.

To write pooled data in stacked form into an ASCII text, Excel, or Lotus worksheet file, first open the pool object, then from the pool menu, select **Proc/Export Pool data (ASCII, .XLS, .WK?)**.... Note that in order to access the pool specific export tools, you must select this procedure from the pool menu, not from the workfile menu.



EViews will first open a file dialog prompting you to specify a file name and type. If you provide a new name, EViews will create the file; otherwise it will prompt you to overwrite the existing file.

Once you have specified your file, a pool write dialog will be displayed. Here we see the **Excel Spreadsheet Export** dialog. Specify the format of your data, including whether to write series in columns or in rows, and whether to stack by cross-section or by period. Then list the ordinary series, groups, and pool series to be written to the file, the sample of observations to be written, and select any export options. When you click on OK, EViews will write the specified file.

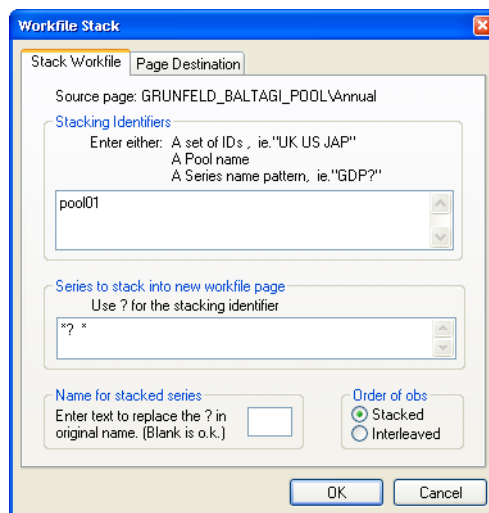
Since EViews allows you to both read and write data that are unstacked, stacked by cross-section, or stacked by date, you may use the pool import and export procedures to restructure your data in accordance with your needs.

Alternatively, you may use the workfile reshaping tools to stack the pooled data in a new workfile page. From the main workfile menu, select **Proc/Reshape Current Page/Stack in New Page...** to open the **Workfile Stack** dialog, and enter the name of a pool object in the top edit field, and the names of the ordinary series, groups, and pool series to be stacked in the second edit field.

The **Order of obs** option allows you to order the data in **Stacked** form (stacking the data by series, which orders by cross-section), or in **Interleaved** format (stacked the data by interleaving series, which orders the data by period or date).

The default naming rule for series in the destination is to use the base name. For example, if you stack the pool series “SALES?” and the individual series GENDER, the corresponding stacked series will, by default, be named “SALES”, and “GENDER”. If use of the default naming convention will create problems in the destination workfile, you should use the **Name for stacked series** field to specify an alternative. If, for example, you enter “_NEW”, the target names will be formed by taking the base name, and appending the additional text, as in “SALES_NEW” and “GENDER_NEW”.

See “[Stacking a Workfile](#)” on page 246 of the *User’s Guide I* for a more detailed discussion of the workfile stacking procedure.



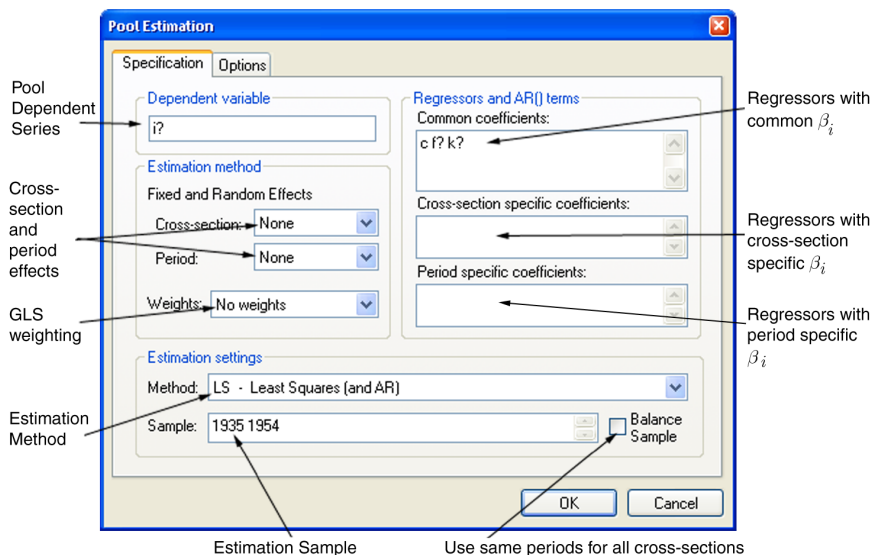
Pooled Estimation

EViews pool objects allow you to estimate your model using least squares or instrumental variables (two-stage least squares), with correction for fixed or random effects in both the cross-section and period dimensions, AR errors, GLS weighting, and robust standard errors, all without rearranging or reordering your data.

We begin our discussion by walking you through the steps that you will take in estimating a pool equation. The wide range of models that EViews supports means that we cannot exhaustively describe all of the settings and specifications. A brief background discussion of the supported techniques is provided in “[Estimation Background](#),” beginning on page 495.

Estimating a Pool Equation

To estimate a pool equation specification, simply press the **Estimate** button on your pool object toolbar or select **Proc/Estimate...** from the pool menu, and the basic pool estimation dialog will open:



First, you should specify the estimation settings in the lower portion of the dialog. Using the **Method** combo box, you may choose between **LS - Least Squares (and AR)**, ordinary least squares regression, **TSLS - Two-Stage Least Squares (and AR)**, two-stage least squares (instrumental variable) regression. If you select the latter, the dialog will differ slightly from this example, with the provision of an additional tab (page) for you to specify your instruments (see “[Instruments](#)” on page 485).

You should also provide an estimation sample in the **Sample** edit box. By default, EViews will use the specified sample string to form use the largest sample possible in each cross-section. An observation will be excluded if any of the explanatory or dependent variables *for that cross-section* are unavailable in that period.

The checkbox for **Balanced Sample** instructs EViews to perform listwise exclusion over all cross-sections. EViews will eliminate an observation if data are unavailable *for any cross-section* in that period. This exclusion ensures that estimates for each cross-section will be based on a common set of dates.

Note that if all of the observations for a cross-section unit are not available, that unit will temporarily be removed from the pool for purposes of estimation. The EViews output will inform you if any cross-section were dropped from the estimation sample.

You may now proceed to fill out the remainder of the dialog.

Dependent Variable

List a pool variable, or an EViews expression containing ordinary and pool variables, in the **Dependent Variable** edit box.

Regressors and AR terms

On the right-hand side of the dialog, you should list your regressors in the appropriate edit boxes:

- **Common coefficients:** — enter variables that have the same coefficient across all cross-section members of the pool. EViews will include a single coefficient for each variable, and will label the output using the original expression.
- **Cross-section specific coefficients:** — list variables with different coefficients for each member of the pool. EViews will include a different coefficient for each cross-sectional unit, and will label the output using a combination of the cross-section identifier and the series name.
- **Period specific coefficients:** — list variables with different coefficients for each observed period. EViews will include a different coefficient for each period unit, and will label the output using a combination of the period identifier and the series name.

For example, if you include the ordinary variable TIME and POP? in the common coefficient list, the output will include estimates for TIME and POP?. If you include these variables in the cross-section specific list, the output will include coefficients labeled “_USA—TIME”, “_UK—TIME”, and “_USA—POP_USA”, “_UK—POP_UK”, etc.

Be aware that estimating your model with cross-section or period specific variables may generate large numbers of coefficients. If there are cross-section specific regressors, the number of these coefficients equals the product of the number of pool identifiers and the number of variables in the list; if there are period specific regressors, the number of corresponding coefficients is the number of periods times the number of variables in the list.

You may include AR terms in either the common or cross-section coefficients lists. If the terms are entered in the common coefficients list, EViews will estimate the model assuming a common AR error. If the AR terms are entered in the cross-section specific list, EViews will estimate separate AR terms for each pool member. See [“Estimating AR Models” on page 67](#) for a description of AR specifications.

Note that EViews only allows specification by list for pool equations. If you wish to estimate a nonlinear specification, you must first create a system object, and then edit the system specification (see [“Making a System” on page 477](#)).

Fixed and Random Effects

You should account for individual and period effects using the **Fixed and Random Effects** combo boxes. By default, EViews assumes that there are no effects so that the combo boxes are both set to **None**. You may change the default settings to allow for either **Fixed** or **Random** effects in either the cross-section or period dimension, or both.

| |
|--------|
| None |
| Fixed |
| Random |

There are some specifications that are not currently supported. You may not, for example, estimate random effects models with cross-section specific coefficients, AR terms, or weighting. Furthermore, while two-way random effects specifications are supported for balanced data, they may not be estimated in unbalanced designs.

Note that when you select a fixed or random effects specification, EViews will automatically add a constant to the common coefficients portion of the specification if necessary, to ensure that the observation weighted sum of the effects is equal to zero.

Weights

By default, all observations are given equal weight in estimation. You may instruct EViews to estimate your specification with estimated GLS weights using the combo box labeled **Weights**.

If you select **Cross section weights**, EViews will estimate a feasible GLS specification assuming the presence of cross-section heteroskedasticity. If you select **Cross-section SUR**, EViews estimates a feasible GLS specification correcting for both cross-section heteroskedasticity and contemporaneous correlation. Similarly, **Period weights** allows for period heteroskedasticity, while **Period SUR** corrects for both period heteroskedasticity and general correlation of observations within a given cross-section. Note that the SUR specifications are each examples of what is sometimes referred to as the Parks estimator.

| |
|-----------------------|
| No weights |
| Cross-section weights |
| Cross-section SUR |
| Period weights |
| Period SUR |

Options

Clicking on the **Options** tab in the dialog brings up a page displaying a variety of estimation options for pool estimation. Settings that are not currently applicable will be grayed out.

Coef Covariance Method

By default, EViews reports conventional estimates of coefficient standard errors and covariances.

You may use the combo box at the top of the page to select from the various robust methods available for computing the coefficient standard errors. Each of the methods is described in greater detail in [“Robust Coefficient Covariances” on page 504](#).

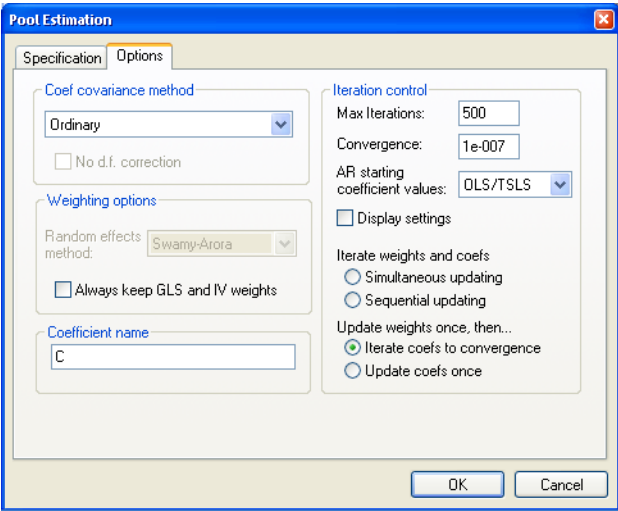
Note that the checkbox **No d.f. correction** permits to you compute robust covariances without the leading degree of freedom correction term. This option may make it easier to match EViews results to those from other sources.

Weighting Options

If you are estimating a specification that includes a random effects specification, EViews will provide you with a **Random effects method** combo box so that you may specify one of the methods for calculating estimates of the component variances. You may choose between the default **Swamy-Arora**, **Wallace-Hussain**, or **Wansbeek-Kapteyn** methods. See [“Random Effects” on page 498](#) for discussion of the differences between the methods. Note that the default Swamy-Arora method should be the most familiar from textbook discussions.

Details on these methods are provided in Baltagi (2001), Baltagi and Chang (1994), Wansbeek and Kapteyn (1989).

The checkbox labeled **Keep GLS weights** may be selected to require EViews to save all estimated GLS weights with the equation, regardless of their size. By default, EViews will not save estimated weights in system (SUR) settings, since the size of the required matrix may be quite large. If the weights are not saved with the equation, there may be some pool views and procedures that are not available.



- Ordinary
- White cross-section
- White period
- White (diagonal)
- Cross-section SUR (PCSE)
- Cross-section weights (PCSE)
- Period SUR (PCSE)
- Period weights (PCSE)

- Swamy-Arora
- Wallace-Hussain
- Wansbeek-Kapteyn

Coefficient Name

By default, EViews uses the default coefficient vector **C** to hold the estimates of the coefficients and effects. If you wish to change the default, simply enter a name in the edit field. If the specified coefficient object exists, it will be used, after resizing if necessary. If the object does not exist, it will be created with the appropriate size. If the object exists but is an incompatible type, EViews will generate an error.

Iteration Control

The familiar **Max Iterations** and **Convergence** criterion edit boxes that allow you to set the convergence test for the coefficients and GLS weights.

If your specification contains AR terms, the **AR starting coefficient values** combo box allows you to specify starting values as a fraction of the OLS (with no AR) coefficients, zero, or user-specified values.

If **Display Settings** is checked, EViews will display additional information about convergence settings and initial coefficient values (where relevant) at the top of the regression output.

The last set of radio buttons is used to determine the iteration settings for coefficients and GLS weighting matrices.

The first two settings, **Simultaneous updating** and **Sequential updating** should be employed when you want to ensure that both coefficients and weighting matrices are iterated to convergence. If you select the first option, EViews will, at every iteration, update both the coefficient vector and the GLS weights; with the second option, the coefficient vector will be iterated to convergence, then the weights will be updated, then the coefficient vector will be iterated, and so forth. Note that the two settings are identical for GLS models without AR terms.

If you select one of the remaining two cases, **Update coefs to convergence** and **Update coefs once**, the GLS weights will only be updated once. In both settings, the coefficients are first iterated to convergence, if necessary, in a model with no weights, and then the weights are computed using these first-stage coefficient estimates. If the first option is selected, EViews will then iterate the coefficients to convergence in a model that uses the first-stage weight estimates. If the second option is selected, the first-stage coefficients will only be iterated once. Note again that the two settings are identical for GLS models without AR terms.

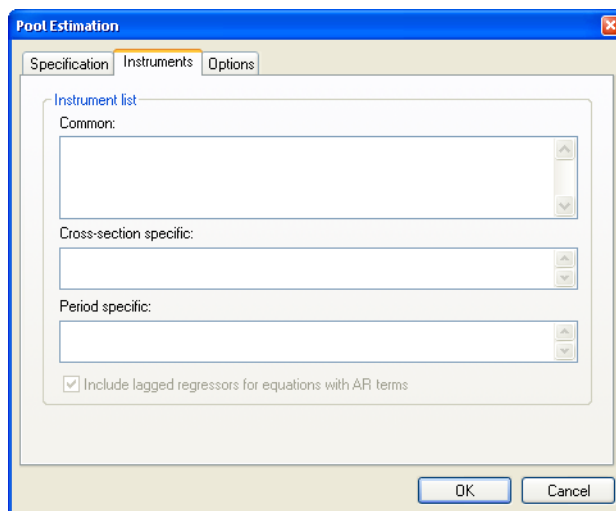
By default, EViews will update GLS weights once, and then will update the coefficients to convergence.

Instruments

To estimate a pool specification using instrumental variables techniques, you should select **TSLS - Two-Stage Least Squares (and AR)** in the **Method** combo box at the bottom of the

main (**Specification**) dialog page. EViews will respond by creating a three-tab dialog in which the middle tab (page) is used to specify your instruments.

As with the regression specification, the instrument list specification is divided into a set of **Common**, **Cross-section specific**, and **Period specific** instruments. The interpretation of these lists is the same as for the regressors; if there are cross-section specific instruments, the number of these instruments equals the product of the number of pool identifiers and the number of variables in the list; if there are period specific instruments, the number of corresponding instruments is the number of periods times the number of variables in the list.



Note that you need not specify constant terms explicitly since EViews will internally add constants to the lists corresponding to the specification in the main page.

Lastly, there is a checkbox labeled **Include lagged regressors for equations with AR terms** that will be displayed if your specification includes AR terms. Recall that when estimating an AR specification, EViews performs nonlinear least squares on an AR differenced specification. By default, EViews will add lagged values of the dependent and independent regressors to the corresponding lists of instrumental variables to account for the modified differenced specification. If, however, you desire greater control over the set of instruments, you may uncheck this setting.

Pool Equation Examples

For illustrative purposes, we employ the balanced firm-level data from Grunfeld (1958) that have been used extensively as an example dataset (*e.g.*, Baltagi, 2001). The panel data consist of annual observations on investment (I_t), firm value (F_t), and capital stock (K_t) for 10 large U.S. manufacturing firms for the 20 years from 1935-54.

The pool identifiers for our data are “AR”, “CH”, “DM”, “GE”, “GM”, “GY”, “IB”, “UO”, “US”, “WH”.

We cannot possibly demonstrate all of the specifications that may be estimated using these data, but we provide a few illustrative examples.

First, we estimate a model regressing $I_i^?$ on the common regressors $F_i^?$ and $K_i^?$, with a cross-section fixed effect. All regression coefficients are restricted to be the same across all cross-sections, so this is equivalent to estimating a model on the stacked data, using the cross-sectional identifiers only for the fixed effect.

The screenshot shows the 'Pool Estimation' dialog box with the following settings:

- Specification** tab is selected.
- Dependent variable:** $i^?$
- Estimation method:** Fixed and Random Effects
 - Cross-section: Fixed
 - Period: None
 - Weights: No weights
- Regressors and AR() terms:**
 - Common coefficients: $i^? k^?$
 - Cross-section specific coefficients: (empty)
 - Period specific coefficients: (empty)
- Estimation settings:**
 - Method: LS - Least Squares (and AR)
 - Sample: 1935 1954
 - ☐ Balance Sample
- Buttons: OK, Cancel

The top portion of the output from this regression, which shows the dependent variable, method, estimation and sample information is given by:

Dependent Variable: I?
 Method: Pooled Least Squares
 Date: 12/03/03 Time: 12:21
 Sample: 1935 1954
 Included observations: 20
 Number of cross-sections used: 10
 Total pool (balanced) observations: 200

| Variable | Coefficient | Std. Error | t-Statistic | Prob. |
|-----------------------|-------------|------------|-------------|--------|
| C | -58.74394 | 12.45369 | -4.716990 | 0.0000 |
| F? | 0.110124 | 0.011857 | 9.287901 | 0.0000 |
| K? | 0.310065 | 0.017355 | 17.86656 | 0.0000 |
| Fixed Effects (Cross) | | | | |
| AR--C | -55.87287 | | | |
| CH--C | 30.93464 | | | |
| DM--C | 52.17610 | | | |
| GE--C | -176.8279 | | | |
| GM--C | -11.55278 | | | |
| GY--C | -28.47833 | | | |
| IB--C | 35.58264 | | | |
| UO--C | -7.809534 | | | |
| US--C | 160.6498 | | | |
| WH--C | 1.198282 | | | |

EViews displays both the estimates of the coefficients and the fixed effects. Note that EViews automatically includes a constant term so that the fixed effects estimates sum to zero and should be interpreted as deviations from an overall mean.

Note also that the estimates of the fixed effects do not have reported standard errors since EViews treats them as nuisance parameters for the purposes of estimation. If you wish to compute standard errors for the cross-section effects, you may estimate a model without a constant and explicitly enter the C in the **Cross-section specific coefficients** edit field.

The bottom portion of the output displays the effects specification and summary statistics for the estimated model.

| Effects Specification | | | |
|---------------------------------------|-----------|-----------------------|----------|
| Cross-section fixed (dummy variables) | | | |
| R-squared | 0.944073 | Mean dependent var | 145.9583 |
| Adjusted R-squared | 0.940800 | S.D. dependent var | 216.8753 |
| S.E. of regression | 52.76797 | Akaike info criterion | 10.82781 |
| Sum squared resid | 523478.1 | Schwarz criterion | 11.02571 |
| Log likelihood | -1070.781 | Hannan-Quinn criter. | 10.90790 |
| F-statistic | 288.4996 | Durbin-Watson stat | 0.716733 |
| Prob(F-statistic) | 0.000000 | | |

A few of these summary statistics require discussion. First, the reported R-squared and F -statistics are based on the difference between the residuals sums of squares from the estimated model, and the sums of squares from a *single* constant-only specification, not from a fixed-effect-only specification. As a result, the interpretation of these statistics is that they describe the explanatory power of the entire specification, including the estimated fixed effects. Second, the reported information criteria use, as the number of parameters, the number of estimated coefficients, including fixed effects. Lastly, the reported Durbin-Watson stat is formed simply by computing the first-order residual correlation on the stacked set of residuals.

We may reestimate this specification using White cross-section standard errors to allow for general contemporaneous correlation between the firm residuals. The “cross-section” designation is used to indicate that covariances are allowed across cross-sections (contemporaneously). Simply click on the **Options** tab and select **White cross-section** as the coefficient covariance matrix, then reestimate the model. The relevant portion of the output is given by:

| White cross-section standard errors & covariance (d.f. corrected) | | | | |
|---|-------------|------------|-------------|--------|
| Variable | Coefficient | Std. Error | t-Statistic | Prob. |
| C | -58.74394 | 19.61460 | -2.994909 | 0.0031 |
| F? | 0.110124 | 0.016932 | 6.504061 | 0.0000 |
| K? | 0.310065 | 0.031541 | 9.830701 | 0.0000 |

The new output shows the method used for computing the standard errors, and the new standard error estimates, t -statistic values, and probabilities reflecting the robust calculation of the coefficient covariances.

Alternatively, we may adopt the Arellano (1987) approach of computing White coefficient covariance estimates that are robust to arbitrary within cross-section residual correlation. Select the **Options** page and choose **White period** as the coefficient covariance method. We

caution, however, that these results assume that the number of cross-sections is large, which is not the case.

White period standard errors & covariance (d.f. corrected)

| Variable | Coefficient | Std. Error | t-Statistic | Prob. |
|----------|-------------|------------|-------------|--------|
| C | -58.74394 | 26.87312 | -2.185974 | 0.0301 |
| F? | 0.110124 | 0.014793 | 7.444423 | 0.0000 |
| K? | 0.310065 | 0.051357 | 6.037432 | 0.0000 |

We may add an AR(1) term to the specification, and compute estimates using Period SUR PCSE methods to compute standard errors that are robust to more general serial correlation. EViews will estimate the transformed model using nonlinear least squares, will form an estimate of the residual covariance matrix, and will use the estimate in forming standard errors. The top portion of the results is given by:

Dependent Variable: I?
 Method: Pooled Least Squares
 Date: 08/21/06 Time: 17:05
 Sample (adjusted): 1936 1954
 Included observations: 19 after adjustments
 Cross-sections included: 10
 Total pool (balanced) observations: 190
 Period SUR (PCSE) standard errors & covariance (d.f. corrected)
 Convergence achieved after 14 iterations

| Variable | Coefficient | Std. Error | t-Statistic | Prob. |
|----------|-------------|------------|-------------|--------|
| C | -63.45169 | 18.10027 | -3.505566 | 0.0006 |
| F? | 0.094744 | 0.010226 | 9.264956 | 0.0000 |
| K? | 0.350205 | 0.044842 | 7.809792 | 0.0000 |
| AR(1) | 0.686108 | 0.073125 | 9.382711 | 0.0000 |

Note in particular the description of the sample adjustment where we show that the estimation drops one observation for each cross-section when performing the AR differencing, as well as the description of the method used to compute coefficient covariances.

Alternatively, we may produce estimates for the two way random effects specification. First, in the **Specification** page, we set both the cross-section and period effects combo boxes to **Random**. Note that the dialog changes to show that weighted estimation is not available with random effects (nor is AR estimation).

Next, in the **Options** page we change the **Random effects method** to use the **Wansbeek-Kapteyn** method of computing the estimates of the random component variances.

Lastly, we click on **OK** to estimate the model.

The top portion of the dialog displays basic information about the specification, including the method used to compute the component variances, as well as the coefficient estimates and associated statistics:

| Dependent Variable: I? | | | | |
|---|-------------|------------|-------------|--------|
| Method: Pooled EGLS (Two-way random effects) | | | | |
| Date: 12/03/03 Time: 14:28 | | | | |
| Sample: 1935 1954 | | | | |
| Included observations: 20 | | | | |
| Number of cross-sections used: 10 | | | | |
| Total pool (balanced) observations: 200 | | | | |
| Wansbeek and Kapteyn estimator of component variances | | | | |
| Variable | Coefficient | Std. Error | t-Statistic | Prob. |
| C | -63.89217 | 30.53284 | -2.092573 | 0.0377 |
| F? | 0.111447 | 0.010963 | 10.16577 | 0.0000 |
| K? | 0.323533 | 0.018767 | 17.23947 | 0.0000 |

The middle portion of the output (not depicted) displays the best-linear unbiased predictor estimates of the random effects themselves.

The next portion of the output describes the estimates of the component variances:

| Effects Specification | | |
|-----------------------|----------|--------|
| | S.D. | Rho |
| Cross-section random | 89.26257 | 0.7315 |
| Period random | 15.77783 | 0.0229 |
| Idiosyncratic random | 51.72452 | 0.2456 |

Here, we see that the estimated cross-section, period, and idiosyncratic error component standard deviations are 89.26, 15.78, and 51.72, respectively. As seen from the values of Rho, these components comprise 0.73, 0.02 and 0.25 of the total variance. Taking the cross-section component, for example, Rho is computed as:

$$0.7315 = 89.26257^2 / (89.26257^2 + 15.77783^2 + 51.72452^2) \quad (37.1)$$

In addition, EViews reports summary statistics for the random effects GLS weighted data used in estimation, and a subset of statistics computed for the unweighted data.

Suppose instead that we elect to estimate a specification with I_t^* as the dependent variable, C and F_t^* as the common regressors, and K_t^* as the cross-section specific regressor, using cross-section weighted least squares. The top portion of the output is given by:

Dependent Variable: I_t^*
Method: Pooled EGLS (Cross-section weights)
Date: 12/18/03 Time: 14:40
Sample: 1935 1954
Included observations: 20
Number of cross-sections used: 10
Total pool (balanced) observations: 200
Linear estimation after one-step weighting matrix

| Variable | Coefficient | Std. Error | t-Statistic | Prob. |
|----------|-------------|------------|-------------|--------|
| C | -4.696363 | 1.103187 | -4.257089 | 0.0000 |
| F_t^* | 0.074084 | 0.004077 | 18.17140 | 0.0000 |
| AR--KAR | 0.092557 | 0.007019 | 13.18710 | 0.0000 |
| CH--KCH | 0.321921 | 0.020352 | 15.81789 | 0.0000 |
| DM--KDM | 0.434331 | 0.151100 | 2.874468 | 0.0045 |
| GE--KGE | -0.028400 | 0.034018 | -0.834854 | 0.4049 |
| GM--KGM | 0.426017 | 0.026380 | 16.14902 | 0.0000 |
| GY--KGY | 0.074208 | 0.007050 | 10.52623 | 0.0000 |
| IB--KIB | 0.273784 | 0.019948 | 13.72498 | 0.0000 |
| UO--KUO | 0.129877 | 0.006307 | 20.59268 | 0.0000 |
| US--KUS | 0.807432 | 0.074870 | 10.78444 | 0.0000 |
| WH--KWH | -0.004321 | 0.031420 | -0.137511 | 0.8908 |

Note that EViews displays results for each of the cross-section specific $K?$ series, labeled using the equation identifier followed by the series name. For example, the coefficient labeled “AR--KAR” is the coefficient of KAR in the cross-section equation for firm AR.

In our last example, we consider the use of the `@INGRP` pool function to estimate an specification containing group dummy variables (see “[Pool Series](#)” on page 464). Suppose we modify our pool definition so that we have defined a group named “MYGROUP” containing the identifiers “GE”, “GM”, and “GY”. We may then estimate a pool specification using the common regressor list:

```
c f? k? @ingrp(mygrp)
```

where the latter pool series expression refers to a set of 10 implicit series containing dummy variables for group membership. The implicit series associated with the identifiers “GE”, “GM”, and “GY” will contain the value 1, and the remaining seven series will contain the value 0.

The results from this estimation are given by:

| Dependent Variable: $I?$ | | | | |
|---|-------------|-----------------------|-------------|--------|
| Method: Pooled Least Squares | | | | |
| Date: 08/22/06 Time: 10:47 | | | | |
| Sample: 1935 1954 | | | | |
| Included observations: 20 | | | | |
| Cross-sections included: 10 | | | | |
| Total pool (balanced) observations: 200 | | | | |
| Variable | Coefficient | Std. Error | t-Statistic | Prob. |
| C | -34.97580 | 8.002410 | -4.370659 | 0.0000 |
| $F?$ | 0.139257 | 0.005515 | 25.25029 | 0.0000 |
| $K?$ | 0.259056 | 0.021536 | 12.02908 | 0.0000 |
| @INGRP(MYGRP) | -137.3389 | 14.86175 | -9.241093 | 0.0000 |
| R-squared | 0.869338 | Mean dependent var | 145.9583 | |
| Adjusted R-squared | 0.867338 | S.D. dependent var | 216.8753 | |
| S.E. of regression | 78.99205 | Akaike info criterion | 11.59637 | |
| Sum squared resid | 1222990. | Schwarz criterion | 11.66234 | |
| Log likelihood | -1155.637 | Hannan-Quinn criter. | 11.62306 | |
| F-statistic | 434.6841 | Durbin-Watson stat | 0.356290 | |
| Prob(F-statistic) | 0.000000 | | | |

We see that the mean value of $I?$ for the three groups is substantially lower than for the remaining groups, and that the difference is statistically significant at conventional levels.

Pool Equation Views and Procedures

Once you have estimated your pool equation, you may examine your output in the usual ways:

Representation

Select **View/Representations** to examine your specification. EViews estimates your pool as a system of equations, one for each cross-section unit.

Estimation Output

View/Estimation Output will change the display to show the results from the pooled estimation.

As with other estimation objects, you can examine the estimates of the coefficient covariance matrix by selecting **View/Coef Covariance Matrix**.

Testing

EViews allows you to perform coefficient tests on the estimated parameters of your pool equation. Select **View/Wald Coefficient Tests...** and enter the restriction to be tested. Additional tests are described in the panel discussion [“Panel Equation Testing” on page 562](#)

Residuals

You can view your residuals in spreadsheet or graphical format by selecting **View/Residuals/Table** or **View/Residuals/Graph**. EViews will display the residuals for each cross-sectional equation. Each residual will be named using the base name RES, followed by the cross-section identifier.

If you wish to save the residuals in series for later use, select **Proc/Make Resids**. This procedure is particularly useful if you wish to form specification or hypothesis tests using the residuals.

Residual Covariance/Correlation

You can examine the estimated residual contemporaneous covariance and correlation matrices. Select **View/Residual** and then either **Correlation Matrix** or **Covariance Matrix** to examine the appropriate matrix.

Forecasting

To perform forecasts using a pool equation you will first make a model. Select **Proc/Make Model** to create an untitled model object that incorporates all of the estimated coefficients. If desired, this model can be edited. Solving the model will generate forecasts for the dependent variable for each of the cross-section units. For further details, see [Chapter 36. “Models,” on page 407](#).

Estimation Background

The basic class of models that can be estimated using a pool object may be written as:

$$Y_{it} = \alpha + X_{it}'\beta_{it} + \delta_i + \gamma_t + \epsilon_{it}, \quad (37.2)$$

where Y_{it} is the dependent variable, and X_{it} is a k -vector of regressors, and ϵ_{it} are the error terms for $i = 1, 2, \dots, M$ cross-sectional units observed for dated periods $t = 1, 2, \dots, T$. The α parameter represents the overall constant in the model, while the δ_i and γ_t represent cross-section or period specific effects (random or fixed). Identification obviously requires that the β coefficients have restrictions placed upon them. They may be divided into sets of common (across cross-section and periods), cross-section specific, and period specific regressor parameters.

While most of our discussion will be in terms of a balanced sample, EViews does not require that your data be balanced; missing values may be used to represent observations that are not available for analysis in a given period. We will detail the unbalanced case only where deemed necessary.

We may view these data as a set of cross-section specific regressions so that we have M cross-sectional equations each with T observations stacked on top of one another:

$$Y_i = \alpha l_T + X_i' \beta_{it} + \delta_i l_T + I_T \gamma + \epsilon_i \quad (37.3)$$

for $i = 1, \dots, M$, where l_T is a T -element unit vector, I_T is the T -element identity matrix, and γ is a vector containing all of the period effects, $\gamma' = (\gamma_1, \gamma_2, \dots, \gamma_T)$.

Analogously, we may write the specification as a set of T period specific equations, each with M observations stacked on top of one another.

$$Y_t = \alpha l_M + X_t' \beta_{it} + I_M \delta + \gamma_t l_M + \epsilon_t \quad (37.4)$$

for $t = 1, \dots, T$, where l_M is a M -element unit vector, I_M is the M -element identity matrix, and δ is a vector containing all of the cross-section effects, $\delta' = (\delta_1, \delta_2, \dots, \delta_M)$.

For purposes of discussion we will employ the stacked representation of these equations. First, for the specification organized as a set of cross-section equations, we have:

$$Y = \alpha l_{MT} + X\beta + (I_M \otimes l_T)\delta + (l_M \otimes I_T)\gamma + \epsilon \quad (37.5)$$

where the matrices β and X are set up to impose any restrictions on the data and parameters between cross-sectional units and periods, and where the general form of the unconditional error covariance matrix is given by:

$$\Omega = E(\epsilon\epsilon') = E \begin{pmatrix} \epsilon_1\epsilon_1' & \epsilon_2\epsilon_1' & \dots & \epsilon_M\epsilon_1' \\ \epsilon_2\epsilon_1' & \epsilon_2\epsilon_2' & & \vdots \\ & & \ddots & \\ \epsilon_M\epsilon_1' & \dots & & \epsilon_M\epsilon_M' \end{pmatrix} \quad (37.6)$$

If instead we treat the specification as a set of period specific equations, the stacked (by period) representation is given by,

$$Y = \alpha l_{MT} + X\beta + (l_M \otimes I_T)\delta + (I_M \otimes l_T)\gamma + \epsilon \quad (37.7)$$

with error covariance,

$$\Omega = E(\epsilon\epsilon') = E \begin{pmatrix} \epsilon_1\epsilon_1' & \epsilon_2\epsilon_1' & \dots & \epsilon_T\epsilon_1' \\ \epsilon_2\epsilon_1' & \epsilon_2\epsilon_2' & & \vdots \\ & & \ddots & \\ \epsilon_T\epsilon_1' & \dots & & \epsilon_T\epsilon_T' \end{pmatrix} \quad (37.8)$$

The remainder of this section describes briefly the various components that you may employ in an EViews pool specification.

Cross-section and Period Specific Regressors

The basic EViews pool specification in [Equation \(37.2\)](#) allows for β slope coefficients that are common to all individuals and periods, as well as coefficients that are either cross-section or period specific. Before turning to the general specification, we consider three extreme cases.

First, if all of the β_{it} are common across cross-sections and periods, we may simplify the expression for [Equation \(37.2\)](#) to:

$$Y_{it} = \alpha + X_{it}'\beta + \delta_i + \gamma_t + \epsilon_{it} \quad (37.9)$$

There are a total of k coefficients in β , each corresponding to an element of x .

Alternately, if all of the β_{it} coefficients are cross-section specific, we have:

$$Y_{it} = \alpha + X_{it}'\beta_i + \delta_i + \gamma_t + \epsilon_{it} \quad (37.10)$$

Note that there are k in each β_i for a total of Mk slope coefficients.

Lastly, if all of the β_{it} coefficients are period specific, the specification may be written as:

$$Y_{it} = \alpha + X_{it}'\beta_t + \delta_i + \gamma_t + \epsilon_{it} \quad (37.11)$$

for a total of Tk slope coefficients.

More generally, splitting X_{it} into the three groups (common regressors X_{0it} , cross-section specific regressors X_{1it}^2 , and period specific regressors X_{2it}), we have:

$$Y_{it} = \alpha + X_{0it}'\beta_0 + X_{1it}'\beta_{1i} + X_{2it}'\beta_{2t} + \delta_i + \gamma_t + \epsilon_{it} \quad (37.12)$$

If there are k_1 common regressors, k_2 cross-section specific regressors, and k_3 period specific regressors, there are a total of $k_0 = k_1 + k_2M + k_3T$ regressors in β .

EViews estimates these models by internally creating interaction variables, M for each regressor in the cross-section regressor list and T for each regressor in the period-specific list, and using them in the regression. Note that estimating models with cross-section or period specific coefficients may lead to the generation of a large number of implicit interaction variables, and may be computationally intensive, or lead to singularities in estimation.

AR Specifications

EViews provides convenient tools for estimating pool specifications that include AR terms. Consider a restricted version of [Equation \(37.2\) on page 495](#) that does not admit period specific regressors or effects,

$$Y_{it} = \alpha + X_{it}'\beta_i + \delta_i + \gamma_t + \epsilon_{it} \quad (37.13)$$

where the cross-section effect δ_i is either not present, or is specified as a fixed effect. We then allow the residuals to follow a general AR process:

$$\epsilon_{it} = \sum_{r=1}^p \rho_{ri} \epsilon_{it-r} + \eta_{it} \quad (37.14)$$

for all i , where the innovations η_{it} are independent and identically distributed, assuming further that there is no unit root. Note that we allow the autocorrelation coefficients ρ to be cross-section, but not period specific.

If, for example, we assume that ϵ_{it} follows an AR(1) process with cross-section specific AR coefficients, EViews will estimate the transformed equation:

$$Y_{it} = \rho_{1i} Y_{it-1} + \alpha(1 - \rho_{1i}) + (X_{it} - \rho_{1i} X_{it-1})' \beta_i + \delta_i(1 - \rho_{1i}) + \eta_{it} \quad (37.15)$$

using iterative techniques to estimate $(\alpha, \beta_i, \rho_i)$ for all i . See [“Estimating AR Models” on page 67](#) for additional discussion.

We emphasize that EViews does place restrictions on the specifications that admit AR errors. AR terms may not be estimated in specifications with period specific regressors or effects. Lastly, AR terms are not allowed in selected GLS specifications (random effects, period specific heteroskedasticity and period SUR). In those GLS specifications where AR terms are allowed, the error covariance assumption is for the innovations not the autoregressive error.

Fixed and Random Effects

The presence of cross-section and period specific effects terms δ and γ may be handled using fixed or random effects methods.

You may, with some restrictions, specify models containing effects in one or both dimension, for example, a fixed effect in the cross-section dimension, a random effect in the period dimension, or a fixed effect in the cross-section and a random effect in the period dimension. Note, in particular, however, that two-way random effects may only be estimated if the data are balanced so that every cross-section has the same set of observations.

Fixed Effects

The fixed effects portions of specifications are handled using orthogonal projections. In the simple one-way fixed effect specifications and the balanced two-way fixed specification, these projections involve the familiar approach of removing cross-section or period specific means from the dependent variable and exogenous regressors, and then performing the specified regression on the demean (see, for example Baltagi, 2001). More generally, we apply the results from Davis (2002) for estimating multi-way error components models with unbalanced data.

Note that if instrumental variables estimation is specified with fixed effects, EViews will automatically add to the instrument list, the constants implied by the fixed effects so that the orthogonal projection is also applied to the instrument list.

Random Effects

The random effects specifications assumes that the corresponding effects δ_i and γ_t are realizations of independent random variables with mean zero and finite variance. Most importantly, the random effects specification assumes that the effect is uncorrelated with the idiosyncratic residual ϵ_{it} .

EViews handles the random effects models using feasible GLS techniques. The first step, estimation of the covariance matrix for the composite error formed by the effects and the residual (e.g., $\nu_{it} = \delta_i + \gamma_t + \epsilon_{it}$ in the two-way random effects specification), uses one of the quadratic unbiased estimators (QUE) from Swamy-Arora, Wallace-Hussain, or Wansbeek-Kapteyn. Briefly, the three QUE methods use the expected values from quadratic forms in one or more sets of first-stage estimated residuals to compute moment estimates of the component variances $(\sigma_\delta^2, \sigma_\gamma^2, \sigma_\epsilon^2)$. The methods differ only in the specifications estimated in evaluating the residuals, and the resulting forms of the moment equations and estimators.

The Swamy-Arora estimator of the component variances, cited most often in textbooks, uses residuals from the within (fixed effect) and between (means) regressions. In contrast, the Wansbeek and Kapteyn estimator uses only residuals from the fixed effect (within) estimator, while the Wallace-Hussain estimator uses only OLS residuals. In general, the three should provide similar answers, especially in large samples. The Swamy-Arora estimator

requires the calculation of an additional model, but has slightly simpler expressions for the component variance estimates. The remaining two may prove easier to estimate in some settings.

Additional details on random effects models are provided in Baltagi (2001), Baltagi and Chang (1994), Wansbeek and Kapteyn (1989). Note that your component estimates may differ slightly from those obtained from other sources since EVIEWS always uses the more complicated *unbiased* estimators involving traces of matrices that depend on the data (see Baltagi (2001) for discussion, especially “Note 2” on p. 27).

Once the component variances have been estimated, we form an estimator of the composite residual covariance, and then GLS transform the dependent and regressor data.

If instrumental variables estimation is specified with random effects, EVIEWS will GLS transform both the data and the instruments prior to estimation. This approach to random effects estimation has been termed generalized two-stage least squares (G2SLS). See Baltagi (2001, p. 111-115) and [“Random Effects and GLS” on page 503](#) for additional discussion.

Generalized Least Squares

You may estimate GLS specifications that account for various patterns of correlation between the residuals. There are four basic variance structures that you may specify: cross-section specific heteroskedasticity, period specific heteroskedasticity, contemporaneous covariances, and between period covariances.

Note that all of the GLS specifications described below may be estimated in one-step form, where we estimate coefficients, compute a GLS weighting transformation, and then reestimate on the weighted data, or in iterative form, where to repeat this process until the coefficients and weights converge.

Cross-section Heteroskedasticity

Cross-section heteroskedasticity allows for a different residual variance for each cross section. Residuals between different cross-sections and different periods are assumed to be 0. Thus, we assume that:

$$\begin{aligned} E(\epsilon_{it}\epsilon_{it}|X_i^*) &= \sigma_i^2 \\ E(\epsilon_{is}\epsilon_{jt}|X_i^*) &= 0 \end{aligned} \tag{37.16}$$

for all i, j, s and t with $i \neq j$ and $s \neq t$, where X_i^* contains X_i and, if estimated by fixed effects, the relevant cross-section or period effects (δ_i, γ) .

Using the cross-section specific residual vectors, we may rewrite the main assumption as:

$$E(\epsilon_i\epsilon_i'|X_i^*) = \sigma_i^2 I_T \tag{37.17}$$

GLS for this specification is straightforward. First, we perform preliminary estimation to obtain cross-section specific residual vectors, then we use these residuals to form estimates of the cross-specific variances. The estimates of the variances are then used in a weighted least squares procedure to form the feasible GLS estimates.

Period Heteroskedasticity

Exactly analogous to the cross-section case, period specific heteroskedasticity allows for a different residual variance for each period. Residuals between different cross-sections and different periods are still assumed to be 0 so that:

$$\begin{aligned} E(\epsilon_{it}\epsilon_{jt} | X_t^*) &= \sigma_t^2 \\ E(\epsilon_{is}\epsilon_{jt} | X_t^*) &= 0 \end{aligned} \quad (37.18)$$

for all i, j, s and t with $s \neq t$, where X_t^* contains X_t and, if estimated by fixed effects, the relevant cross-section or period effects (δ, γ_t) .

Using the period specific residual vectors, we may rewrite the first assumption as:

$$E(\epsilon_t \epsilon_t' | X_t^*) = \sigma_t^2 I_M \quad (37.19)$$

We perform preliminary estimation to obtain period specific residual vectors, then we use these residuals to form estimates of the period variances, reweight the data, and then form the feasible GLS estimates.

Contemporaneous Covariances (Cross-section SUR)

This class of covariance structures allows for conditional correlation between the contemporaneous residuals for cross-section i and j , but restricts residuals in different periods to be uncorrelated. More specifically, we assume that:

$$\begin{aligned} E(\epsilon_{it}\epsilon_{jt} | X_t^*) &= \sigma_{ij} \\ E(\epsilon_{is}\epsilon_{jt} | X_t^*) &= 0 \end{aligned} \quad (37.20)$$

for all i, j, s and t with $s \neq t$. Note that the contemporaneous covariances do not vary over t .

Using the period specific residual vectors, we may rewrite this assumption as,

$$E(\epsilon_t \epsilon_t' | X_t^*) = \Omega_M \quad (37.21)$$

for all t , where,

$$\Omega_M = \begin{pmatrix} \sigma_{11} & \sigma_{12} & \cdots & \sigma_{1M} \\ \sigma_{12} & \sigma_{22} & & \vdots \\ & & \ddots & \\ \sigma_{M1} & \cdots & & \sigma_{MM} \end{pmatrix} \quad (37.22)$$

There does not appear to be a commonly accepted name for this variance structure, so we term it a *Cross-section SUR* specification since it involves covariances across cross-sections as in a seemingly unrelated regressions type framework (where each equation corresponds to a cross-section).

Cross-section SUR weighted least squares on this specification (sometimes referred to as the Parks estimator) is simply the feasible GLS estimator for systems where the residuals are both cross-sectionally heteroskedastic and contemporaneously correlated. We employ residuals from first stage estimates to form an estimate of Ω_M . In the second stage, we perform feasible GLS.

Bear in mind that there are potential pitfalls associated with the SUR/Parks estimation (see Beck and Katz (1995)). For one, EViews may be unable to compute estimates for this model when you the dimension of the relevant covariance matrix is large and there are a small number of observations available from which to obtain covariance estimates. For example, if we have a cross-section SUR specification with large numbers of cross-sections and a small number of time periods, it is quite likely that the estimated residual correlation matrix will be nonsingular so that feasible GLS is not possible.

It is worth noting that an attractive alternative to the SUR methodology estimates the model without a GLS correction, then corrects the coefficient estimate covariances to account for the contemporaneous correlation. See [“Robust Coefficient Covariances” on page 504](#).

Note also that if cross-section SUR is combined with instrumental variables estimation, EViews will employ a Generalized Instrumental Variables estimator in which both the data and the instruments are transformed using the estimated covariances. See Wooldridge (2002) for discussion and comparison with the three-stage least squares approach.

(Period Heteroskedasticity and Serial Correlation) Period SUR

This class of covariance structures allows for arbitrary period serial correlation and period heteroskedasticity between the residuals for a given cross-section, but restricts residuals in different cross-sections to be uncorrelated. Accordingly, we assume that:

$$\begin{aligned} E(\epsilon_{is}\epsilon_{it} | X_i^*) &= \sigma_{st} \\ E(\epsilon_{is}\epsilon_{jt} | X_i^*) &= 0 \end{aligned} \quad (37.23)$$

for all i, j, s and t with $i \neq j$. Note that the heteroskedasticity and serial correlation does not vary across cross-sections i .

Using the cross-section specific residual vectors, we may rewrite this assumption as,

$$E(\epsilon_i \epsilon_i' | X_i^*) = \Omega_T \quad (37.24)$$

for all i , where,

$$\Omega_T = \begin{pmatrix} \sigma_{11} & \sigma_{12} & \cdots & \sigma_{1T} \\ \sigma_{12} & \sigma_{22} & & \vdots \\ & & \ddots & \\ \sigma_{T1} & \cdots & & \sigma_{TT} \end{pmatrix} \quad (37.25)$$

We term this a *Period SUR* specification since it involves covariances across periods within a given cross-section, as in a seemingly unrelated regressions framework with period specific equations. In estimating a specification with Period SUR, we employ residuals obtained from first stage estimates to form an estimate of Ω_T . In the second stage, we perform feasible GLS.

See “[Contemporaneous Covariances \(Cross-section SUR\)](#)” on page 500 for related discussion.

Instrumental Variables

All of the pool specifications may be estimated using instrumental variables techniques. In general, the computation of the instrumental variables estimator is a straightforward extension of the standard OLS estimator. For example, in the simplest model, the OLS estimator may be written as:

$$\hat{\beta}_{OLS} = \left(\sum_i X_i' X_i \right)^{-1} \left(\sum_i X_i' Y_i \right) \quad (37.26)$$

while the corresponding IV estimator is given by:

$$\hat{\beta}_{IV} = \left(\sum_i X_i' P_{Z_i} X_i \right)^{-1} \left(\sum_i X_i' P_{Z_i} Y_i \right) \quad (37.27)$$

where $P_{Z_i} = (Z_i(Z_i' Z_i)^{-1} Z_i')$ is the orthogonal projection matrix onto the Z_i .

There are, however, additional complexities introduced by instruments that require some discussion.

Cross-section and Period Specific Instruments

As with the regressors, we may divide the instruments into three groups (common instruments Z_{0it} , cross-section specific instruments Z_{1it} , and period specific instruments Z_{2it}).

You should make certain that any exogenous variables in the regressor groups are included in the corresponding instrument groups, and be aware that each entry in the latter two groups generates multiple instruments.

Fixed Effects

If instrumental variables estimation is specified with fixed effects, EViews will automatically add to the instrument list any constants implied by the fixed effects so that the orthogonal projection is also applied to the instrument list. Thus, if Q is the fixed effects transformation operator, we have:

$$\begin{aligned}\hat{\beta}_{OLS} &= \left(\sum_i X_i' Q X_i \right)^{-1} \left(\sum_i X_i' Q Y_i \right) \\ \hat{\beta}_{IV} &= \left(\sum_i X_i Q P_{\tilde{Z}_i} Q X_i \right)^{-1} \left(\sum_i X_i' Q P_{\tilde{Z}_i} Q Y_i \right)\end{aligned}\tag{37.28}$$

where $\tilde{Z}_i = Q Z_i$.

Random Effects and GLS

Similarly, for random effects and other GLS estimators, EViews applies the weighting to the instruments as well as the dependent variable and regressors in the model. For example, with data estimated using cross-sectional GLS, we have:

$$\begin{aligned}\hat{\beta}_{GLS} &= \left(\sum_i X_i' \hat{\Omega}_M^{-1} X_i \right)^{-1} \left(\sum_i X_i' \hat{\Omega}_M^{-1} Y_i \right) \\ \hat{\beta}_{GIV} &= \left(\sum_i X_i \hat{\Omega}_M^{-1/2} P_{Z_i^*} \hat{\Omega}_M^{-1/2} X_i \right)^{-1} \left(\sum_i X_i' \hat{\Omega}_M^{-1/2} P_{Z_i^*} \hat{\Omega}_M^{-1/2} Y_i \right)\end{aligned}\tag{37.29}$$

where $Z_i^* = \hat{\Omega}_M^{-1/2} Z_i$.

In the context of random effects specifications, this approach to IV estimation is termed generalized two-stage least squares (G2SLS) method (see Baltagi (2001, p. 111-115) for references and discussion). Note that in implementing the various random effects methods (Swamy-Arora, Wallace-Hussain, Wansbeek-Kapteyn), we have extended the existing results to derive the unbiased variance components estimators in the case of instrumental variables estimation.

More generally, the approach may simply be viewed as a special case of the Generalized Instrumental Variables (GIV) approach in which data and the instruments are both transformed using the estimated covariances. You should be aware that this has approach has the effect of altering the implied orthogonality conditions. See Wooldridge (2002) for discussion and comparison with a three-stage least squares approach in which the instruments are not transformed. See “GMM Details” on page 571 for an alternative approach.

AR Specifications

EViews estimates AR specifications by transforming the data to a nonlinear least squares specification, and jointly estimating the original and the AR coefficients.

This transformation approach raises questions as to what instruments to use in estimation. By default, EViews adds instruments corresponding to the lagged endogenous and lagged exogenous variables introduced into the specification by the transformation.

For example, in an AR(1) specification, we have the original specification,

$$Y_{it} = \alpha + X_{it}'\beta_i + \delta_i + \epsilon_{it} \quad (37.30)$$

and the transformed equation,

$$Y_{it} = \rho_{1i} Y_{it-1} + \alpha(1 - \rho_{1i}) + (X_{it} - \rho_{1i} X_{it-1})'\beta_i + \delta_i(1 - \rho_{1i}) + \eta_{it} \quad (37.31)$$

where Y_{it-1} and X_{it-1} are introduced by the transformation. EViews will, by default, add these to the previously specified list of instruments Z_{it} .

You may, however, instruct EViews not to add these additional instruments. Note, however, that the order condition for the transformed model is different than the order condition for the untransformed specification since we have introduced additional coefficients corresponding to the AR coefficients. If you elect not to add the additional instruments automatically, you should make certain that you have enough instruments to account for the additional terms.

Robust Coefficient Covariances

In this section, we describe the basic features of the various robust estimators, for clarity focusing on the simple cases where we compute robust covariances for models estimated by standard OLS without cross-section or period effects. The extensions to models estimated using instrumental variables, fixed or random effects, and GLS weighted least squares are straightforward.

White Robust Covariances

The White cross-section method is derived by treating the pool regression as a multivariate regression (with an equation for each cross-section), and computing White-type robust standard errors for the system of equations. We may write the coefficient covariance estimator as:

$$\left(\frac{N^*}{N^* - K^*} \right) \left(\sum_t X_t' X_t \right)^{-1} \left(\sum_t X_t' \hat{\epsilon}_t \hat{\epsilon}_t' X_t \right) \left(\sum_t X_t' X_t \right)^{-1} \quad (37.32)$$

where the leading term is a degrees of freedom adjustment depending on the total number of observations in the stacked data, N^* is the total number of stacked observations, and K^* , the total number of estimated parameters.

This estimator is robust to cross-equation (contemporaneous) correlation as well as different error variances in each cross-section. Specifically, the unconditional contemporaneous variance matrix $E(\epsilon_t \epsilon_t') = \Omega_M$ is unrestricted, and the conditional variance matrix $E(\epsilon_t \epsilon_t' | X_t^*)$ can depend on X_t^* in arbitrary, unknown fashion. See Wooldridge (2002, p. 148-153) and Arellano (1987).

Alternatively, the White period method is robust to arbitrary serial correlation and time-varying variances in the disturbances. The coefficient covariances are calculated using:

$$\left(\frac{N^*}{N^* - K^*} \right) \left(\sum_i X_i' X_i \right)^{-1} \left(\sum_i X_i' \hat{\epsilon}_i \hat{\epsilon}_i' X_i \right) \left(\sum_i X_i' X_i \right)^{-1} \quad (37.33)$$

where, in contrast to Equation (37.32), the summations are taken over individuals and individual stacked data instead of periods.

The White period robust coefficient variance estimator is designed to accommodate arbitrary serial correlation and time-varying variances in the disturbances. The corresponding multivariate regression (with an equation for each period) allows the unconditional variance matrix $E(\epsilon_i \epsilon_i') = \Omega_T$ to be unrestricted, and the conditional variance matrix $E(\epsilon_i \epsilon_i' | X_i^*)$ may depend on X_i^* in general fashion.

In contrast, the White (diagonal) method is robust to observation specific heteroskedasticity in the disturbances, but not to correlation between residuals for different observations. The coefficient asymptotic variance is estimated as:

$$\left(\frac{N^*}{N^* - K^*} \right) \left(\sum_{i,t} X_{it}' X_{it} \right)^{-1} \left(\sum_{i,t} \hat{\epsilon}_{it}^2 X_{it}' X_{it} \right) \left(\sum_{i,t} X_{it}' X_{it} \right)^{-1} \quad (37.34)$$

This method allows the unconditional variance matrix $E(\epsilon \epsilon') = \Lambda$ to be an unrestricted diagonal matrix, and the conditional variances $E(\epsilon_{it}^2 | X_{it}^*)$ to depend on X_{it}^* in general fashion. Note that this method is both more general and more restrictive than the previous approaches. It is more general in that observations in the same cross-section or period may have different variances; it is more restrictive in that all off-diagonal variances are restricted to be zero.

EViews also allows you to compute non degree of freedom corrected versions of all of the robust coefficient covariance estimators. In these cases, the leading ratio term in the expressions above is dropped from the calculation. While this has no effect on the asymptotic validity of the estimates, it has the practical effect of lowering all of your standard error estimates.

PCSE Robust Covariances

The remaining methods are variants of the first two White statistics in which residuals are replaced by moment estimators for the unconditional variances. These methods, which are variants of the so-called *Panel Corrected Standard Error* (PCSE) methodology (Beck and

Katz, 1995), are robust to unrestricted unconditional variances Ω_M and Ω_T , but place additional restrictions on the conditional variance matrices. A sufficient (though not necessary) condition is that the conditional and unconditional variances are the same.

For example, the Cross-section SUR (PCSE) method replaces the outer product of the cross-section residuals in Equation (37.32) with an estimate of the cross-section residual (contemporaneous) covariance matrix Ω_M :

$$\left(\frac{N^*}{N^* - K^*}\right) \left(\sum_t X_t' X_t\right)^{-1} \left(\sum_t X_t' \hat{\Omega}_M X_t\right) \left(\sum_t X_t' X_t\right)^{-1} \quad (37.35)$$

Analogously, the Period SUR (PCSE) replaces the outer product of the period residuals in Equation (37.33) with an estimate of the period covariance Ω_T :

$$\left(\frac{N^*}{N^* - K^*}\right) \left(\sum_i X_i' X_i\right)^{-1} \left(\sum_i X_i' \hat{\Omega}_T X_i\right) \left(\sum_i X_i' X_i\right)^{-1} \quad (37.36)$$

The two diagonal forms of these estimators, Cross-section weights (PCSE) _{$\hat{\Omega}_M$} and Period weights (PCSE) _{$\hat{\Omega}_T$} , use only the diagonal elements of the relevant $\hat{\Omega}_M$ and $\hat{\Omega}_T$. These covariance estimators are robust to heteroskedasticity across cross-sections or periods, respectively, but not to general correlation of residuals.

The non degree of freedom corrected versions of these estimators remove the leading term involving the number of observations and number of coefficients.

References

- Arellano, M. (1987). "Computing Robust Standard Errors for Within-groups Estimators," *Oxford Bulletin of Economics and Statistics*, 49, 431-434.
- Baltagi, Badi H. (2001). *Econometric Analysis of Panel Data, Second Edition*, West Sussex, England: John Wiley & Sons.
- Baltagi, Badi H. and Young-Jae Chang (1994). "Incomplete Panels: A Comparative Study of Alternative Estimators for the Unbalanced One-way Error Component Regression Model," *Journal of Econometrics*, 62, 67-89.
- Beck, Nathaniel and Jonathan N. Katz (1995). "What to Do (and Not to Do) With Time-series Cross-section Data," *American Political Science Review*, 89(3), 634-647.
- Breitung, Jörg (2000). "The Local Power of Some Unit Root Tests for Panel Data," in B. Baltagi (ed.), *Advances in Econometrics, Vol. 15: Nonstationary Panels, Panel Cointegration, and Dynamic Panels*, Amsterdam: JAI Press, p. 161-178.
- Choi, I. (2001). "Unit Root Tests for Panel Data," *Journal of International Money and Finance*, 20: 249-272.
- Davis, Peter (2002). "Estimating Multi-way Error Components Models with Unbalanced Data Structures," *Journal of Econometrics*, 106, 67-95.
- Fisher, R. A. (1932). *Statistical Methods for Research Workers, 4th Edition*, Edinburgh: Oliver & Boyd.

- Grunfeld, Yehuda (1958). "The Determinants of Corporate Investment," *Unpublished Ph.D Thesis*, Department of Economics, University of Chicago.
- Hadri, Kaddour (2000). "Testing for Stationarity in Heterogeneous Panel Data," *Econometric Journal*, 3, 148–161.
- Im, K. S., M. H. Pesaran, and Y. Shin (2003). "Testing for Unit Roots in Heterogeneous Panels," *Journal of Econometrics*, 115, 53–74.
- Kao, C. (1999). "Spurious Regression and Residual-Based Tests for Cointegration in Panel Data," *Journal of Econometrics*, 90, 1–44.
- Levin, A., C. F. Lin, and C. Chu (2002). "Unit Root Tests in Panel Data: Asymptotic and Finite-Sample Properties," *Journal of Econometrics*, 108, 1–24.
- Maddala, G. S. and S. Wu (1999). "A Comparative Study of Unit Root Tests with Panel Data and A New Simple Test," *Oxford Bulletin of Economics and Statistics*, 61, 631–52.
- Pedroni, P. (1999). "Critical Values for Cointegration Tests in Heterogeneous Panels with Multiple Regressors," *Oxford Bulletin of Economics and Statistics*, 61, 653–70.
- Pedroni, P. (2004). "Panel Cointegration; Asymptotic and Finite Sample Properties of Pooled Time Series Tests with an Application to the PPP Hypothesis," *Econometric Theory*, 20, 597–625.
- Wansbeek, Tom, and Arie Kapteyn (1989). "Estimation of the Error Components Model with Incomplete Panels," *Journal of Econometrics*, 41, 341–361.
- Wooldridge, Jeffrey M. (2002). *Econometric Analysis of Cross Section and Panel Data*, Cambridge, MA: The MIT Press.

Chapter 38. Working with Panel Data

EViews provides you with specialized tools for working with stacked data that have a panel structure. You may have, for example, data for various individuals or countries that are stacked one on top of another.

The first step in working with stacked panel data is to describe the panel structure of your data: we term this step *structuring the workfile*. Once your workfile is structured as a panel workfile, you may take advantage of the EViews tools for working with panel data, and for estimating equation specifications using the panel structure.

The following discussion assumes that you have an understanding of the basics of panel data. “[Panel Data](#),” beginning on [page 206](#) of the *User’s Guide I* provides background on the characteristics of panel structured data.

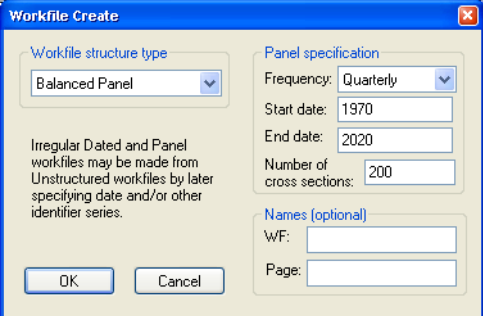
We first review briefly the process of applying a panel structure to a workfile. The remainder of the discussion in this chapter focuses on the basics working with data in a panel workfile. [Chapter 39. “Panel Estimation,” on page 541](#) outlines the features of equation estimation in a panel workfile.

Structuring a Panel Workfile

The first step in panel data analysis is to define the panel structure of your data. By defining a panel structure for your data, you perform the dual tasks of identifying the cross-section associated with each observation in your stacked data, and of defining the way that lags and leads operate in your workfile.

While the procedures for structuring a panel workfile outlined below are described in greater detail elsewhere, an abbreviated review may prove useful (for additional detail, see “[Describing a Balanced Panel Workfile](#)” on [page 41](#), “[Dated Panels](#)” on [page 219](#), and “[Undated Panels](#)” on [page 224](#) of the *User’s Guide I*).

There are two basic ways to create a panel structured workfile. First, you may create a *new* workfile that has a simple balanced panel structure. Simply select **File/New/Workfile...** from the main EViews menu to open the **Workfile Create** dialog. Next, select **Balanced Panel** from the **Workfile structure type** combo box, and fill out the dialog as desired. Here, we create a balanced quarterly panel (ranging from



The screenshot shows the 'Workfile Create' dialog box in EViews. It has a blue title bar with the text 'Workfile Create' and a close button. The dialog is divided into two main sections. The left section, titled 'Workfile structure type', contains a dropdown menu currently set to 'Balanced Panel'. Below this, there is a note: 'Irregular Dated and Panel workfiles may be made from Unstructured workfiles by later specifying date and/or other identifier series.' The right section, titled 'Panel specification', contains several input fields: 'Frequency' is set to 'Quarterly', 'Start date' is '1970', 'End date' is '2020', and 'Number of cross sections' is '200'. Below these are two more input fields labeled 'Names (optional)', 'WF:', and 'Page:'. At the bottom of the dialog are two buttons: 'OK' and 'Cancel'.

1970Q1 to 2020Q4) with 200 cross-sections. We also enter “Quarterly” in the **Page** name edit field.

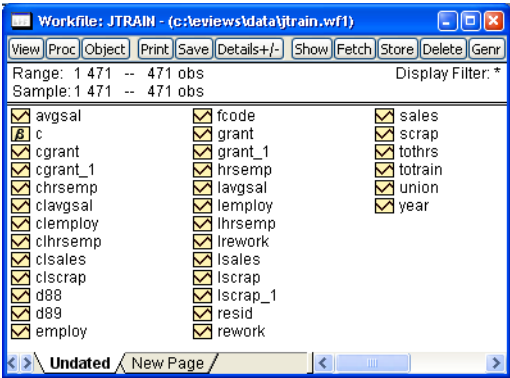
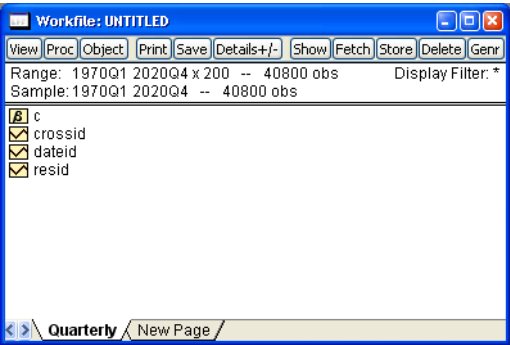
When you click on **OK**, EViews will create an appropriately structured workfile with 40,800 observations (51 years, 4 quarters, 200 cross-sections). You may then enter or import the data into the workfile.

More commonly, you will use the second method of structuring a panel workfile, in which you first read stacked data into an unstructured workfile, and then apply a structure to the workfile. While there are a number of issues involved with this operation, let us consider a simple, illustrative example of the basic method.

Suppose that we have data for the job training example considered by Wooldridge (2002), using data from Holzer, *et al.* (1993).

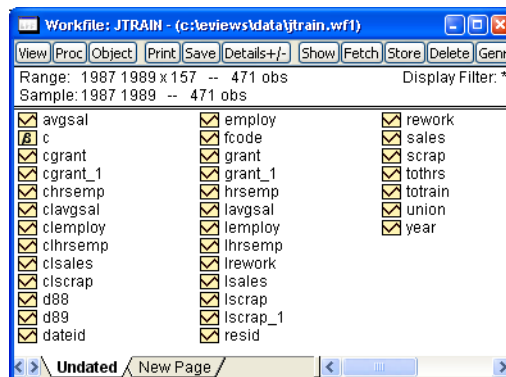
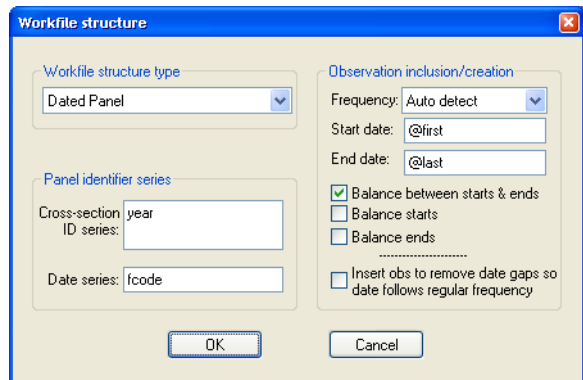
These data form a balanced panel of 3 annual observations on 157 firms. The data are first read into a 471 observation, unstructured EViews workfile. The values of the series **YEAR** and **FCODE** may be used to identify the date and cross-section, respectively, for each observation.

To apply a panel structure to this workfile, simply double click on the “Range:” line at the top of the workfile window, or select **Proc/Structure/Resize Current Page...** to open the **Workfile structure** dialog. Select **Dated Panel** as our **Workfile structure type**.



Next, enter YEAR as the **Date series** and FCODE as the **Cross-section ID series**. Since our data form a simple balanced dated panel, we need not concern ourselves with the remaining settings, so we may simply click on **OK**.

EViews will analyze the data in the specified **Date series** and **Cross-section ID series** to determine the appropriate structure for the workfile. The data in the workfile will be sorted by cross-section ID series, and then by date, and the panel structure will be applied to the workfile.



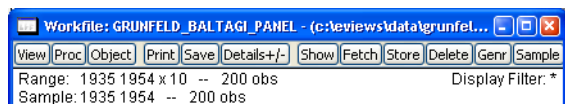
Panel Workfile Display

The two most prominent visual changes in a panel structured workfile are the change in the range and sample information display at the top of the workfile window, and the change in the labels used to identify individual observations.

Range and Sample

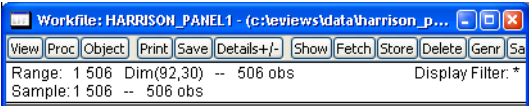
The first visual change in a panel structured workfile is in the **Range** and **Sample** descriptions at the top of workfile window.

For a dated panel workfile, EViews will list both the earliest and latest observed dates, the number of cross-sections, and the total number of



unique observations. Here we see the top portion of an annual workfile with observations from 1935 to 1954 for 10 cross-sections. Note that workfile sample is described using the earliest and latest observed annual frequency dates (“1935 1954”).

In contrast, an undated panel workfile will display an observation range of 1 to the total number of observations. The panel dimension statement will indicate the largest number of observations in a cross-section and the number of cross-sections. Here, we have 92 cross-sections containing up to 30 observations, for a total of 506 observations. Note that the workfile sample is described using the raw observation numbers (“1 506”) since there is no notion of a date pair in undated panels.



You may, at any time, click on the **Range** display line or select **Proc/WF Structure & Range...** to bring up the **Workfile Structure** dialog so that you may modify or remove your panel structure.

Observation Labels

The left-hand side of every workfile contains observation labels that identify each observation. In a simple unstructured workfile, these labels are simply the integers from 1 to the total number of observations in the workfile. For dated, non-panel workfiles, these labels are representations of the unique dates associated with each observation. For example, in an annual workfile ranging from 1935 to 1950, the observation labels are of the form “1935”, “1936”, etc.

In contrast, the observation labels in a panel workfile must reflect the fact that observations possess both cross-section and within-cross-section identifiers. Accordingly, EViews will form observation identifiers using both the cross-section and the cell ID values.

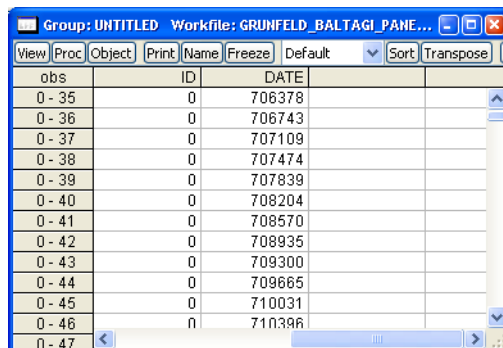
Here, for example, we see the labels for a panel annual workfile structured using ID as the cross-section identifier, and DATE as the cell ID (we have widened the observation column so that the full observation identifier is visible). The labels have been formed by concatenating the ID value, a hyphen, and a two-digit representation of the date.

| obs | ID | DATE |
|---------------|----------|--------|
| 0.000000 - 35 | 0.000000 | 706378 |
| 0.000000 - 36 | 0.000000 | 706743 |
| 0.000000 - 37 | 0.000000 | 707109 |
| 0.000000 - 38 | 0.000000 | 707474 |
| 0.000000 - 39 | 0.000000 | 707839 |
| 0.000000 - 40 | 0.000000 | 708204 |
| 0.000000 - 41 | 0.000000 | 708570 |
| 0.000000 - 42 | 0.000000 | 708935 |
| 0.000000 - 43 | 0.000000 | 709300 |
| 0.000000 - 44 | 0.000000 | 709665 |
| 0.000000 - 45 | 0.000000 | 710031 |
| 0.000000 - 46 | 0.000000 | 710396 |
| 0.000000 - 47 | 0.000000 | 710761 |
| 0.000000 - 48 | 0.000000 | 711126 |

Note that EViews generally uses the display formats for the cross-section and cell ID series when forming the observation label. For example, since ID is displayed using 9 fixed characters, the observation labels include 9

fixed characters for the ID value. A notable exception to this rule is seen in the DATE portion of the observation label. Since we have specified a dated annual panel format, EViews uses this information to shorten the date portion of the observation label by using a 2-digit year identifier.

We may improve the appearance of the observation labels to changing the spreadsheet display settings in the ID series. Simply open the ID series window, click on the **Properties** button and select the **Display** tab, then change the display to show **Significant digits**. Similarly, we may open the DATE series and specify display properties to show the dates by selecting **Period** and then the **Current workfile** format.



| obs | ID | DATE |
|--------|----|--------|
| 0 - 35 | 0 | 706378 |
| 0 - 36 | 0 | 706743 |
| 0 - 37 | 0 | 707109 |
| 0 - 38 | 0 | 707474 |
| 0 - 39 | 0 | 707839 |
| 0 - 40 | 0 | 708204 |
| 0 - 41 | 0 | 708570 |
| 0 - 42 | 0 | 708935 |
| 0 - 43 | 0 | 709300 |
| 0 - 44 | 0 | 709665 |
| 0 - 45 | 0 | 710031 |
| 0 - 46 | 0 | 710396 |
| 0 - 47 | | |

Here, we show the observation labels along with formatted values for ID and DATE. Note again that the observation labels do not use the formatting for DATE, but instead use formatting based on the workfile date structure.

Panel Workfile Information

When working with panel data, it is important to keep the basic structure of your workfile in mind at all times. EViews provides you with tools to access information about the structure of your workfile.

Workfile Structure

First, the workfile statistics view provides a convenient place for you to examine the structure of your panel workfile. Simply click on **View/Statistics** from the main menu to display a summary of the structure and contents of your workfile.

```
Workfile Statistics
Date: 01/13/04   Time: 08:29
Name: GRUNFELD_BALTAGI_PANEL
Number of pages: 3
```

```
Page: panel
```

```
Workfile structure: Panel - Annual
Indices: ID x DATE
Panel dimension: 10 x 20
Range: 1935 1954 x 10  -- 200 obs
```

| Object | Count | Data Points |
|--------------|----------|-------------|
| series | 7 | 1400 |
| alpha | 1 | 200 |
| coef | 1 | 751 |
| equation | 10 | |
| <u>group</u> | <u>2</u> | |
| Total | 21 | 2351 |

The top portion of the display for our first example workfile is depicted above. The statistics view identifies the page as an annual panel workfile that is structured using the identifiers ID and DATE. There are 10 cross-sections with 20 observations each, for years ranging from 1935 to 1954. For unbalanced data, the number of observations per cross-section reported will be the largest number observed across the cross-sections.

To return the display to the original workfile directory, select **View/Workfile Directory** from the main workfile menu.

Identifier Indices

EViews provides series expressions and functions that provide information about the cross-section, cell, and observation IDs associated with each observation in a panel workfile.

Cross-section Index

The series expression `@crossid` provides index identifiers for each observation corresponding to the cross-section to which the observation belongs. If, for example, there are 8 observations with cross-section identifier series values (in order), “B”, “A”, “A”, “A”, “B”, “A”, “A”, and “B”, the command:

```
series cxid = @crossid
```

assigns a group identifier value of 1 or 2 to each observation in the workfile. Since the panel workfile is sorted by the cross-section ID values, observations with the identifier value “A” will be assigned a CXID value of 1, while “B” will be assigned 2.

A one-way tabulation of the CXID series shows the number of observations in each cross-section or group:

Tabulation of CXID
 Date: 02/04/04 Time: 09:08
 Sample: 1 8
 Included observations: 8
 Number of categories: 2

| Value | Count | Percent | Cumulative | Cumulative |
|-------|-------|---------|------------|------------|
| | | | Count | Percent |
| 1 | 5 | 62.50 | 5 | 62.50 |
| 2 | 3 | 37.50 | 8 | 100.00 |
| Total | 8 | 100.00 | 8 | 100.00 |

Cell Index

Similarly, `@cellid` may be used to obtain integers uniquely indexing cell IDs. `@cellid` numbers observations using an index corresponding to the ordered unique values of the cell or date ID values. Note that since the indexing uses all unique values of the cell or date ID series, the observations within a cross-section may be indexed non-sequentially.

Suppose, for example, we have a panel workfile with two cross-sections. There are 5 observations in the cross-section “A” with cell ID values “1991”, “1992”, “1993”, “1994”, and “1999”, and 3 observations in the cross-section “B” with cell ID values “1993”, “1996”, “1998”. There are 7 unique cell ID values (“1991”, “1992”, “1993”, “1994”, “1996”, “1998”, “1999”) in the workfile.

The series assignment

```
series cellid = @cellid
```

will assign to the “A” observations in CELLID the values “1991”, “1992”, “1993”, “1994”, “1997”, and to the “B” observations the values “1993”, “1995”, and “1996”.

A one-way tabulation of the CELLID series provides you with information about the number of observations with each index value:

Tabulation of CELLID
 Date: 02/04/04 Time: 09:11
 Sample: 1 8
 Included observations: 8
 Number of categories: 7

| Value | Count | Percent | Cumulative | |
|-------|-------|---------|------------|---------|
| | | | Count | Percent |
| 1 | 1 | 12.50 | 1 | 12.50 |
| 2 | 1 | 12.50 | 2 | 25.00 |
| 3 | 2 | 25.00 | 4 | 50.00 |
| 4 | 1 | 12.50 | 5 | 62.50 |
| 5 | 1 | 12.50 | 6 | 75.00 |
| 6 | 1 | 12.50 | 7 | 87.50 |
| 7 | 1 | 12.50 | 8 | 100.00 |
| Total | 8 | 100.00 | 8 | 100.00 |

Within Cross-section Observation Index

Alternately, `@obsid` returns an integer uniquely indexing observations within a cross-section. The observations will be numbered sequentially from 1 through the number of observations in the corresponding cross-section. In the example above, with two cross-section groups “A” and “B” containing 5 and 3 observations, respectively, the command:

```
series withinid = @obsid
```

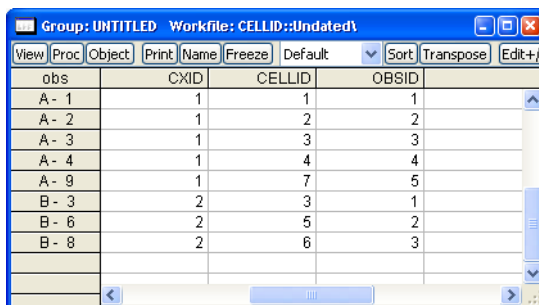
would number the 5 observations in cross-section “A” from 1 through 5, and the 3 observations in group “B” from 1 through 3.

Bear in mind that while `@cellid` uses information about all of the ID values in creating its index, `@obsid` only uses the ordered observations within a cross-section in forming the index. As a result, the only similarity between observations that share an `@obsid` value is their ordering within the cross-section. In contrast, observations that share a `@cellid` value also share values for the underlying cell ID.

It is worth noting that if a panel workfile is balanced so that each cross-section has the same cell ID values, `@obsid` and `@cellid` yield identical results.

Workfile Observation Index

In rare cases, you may wish to enumerate the observations beginning at the first observation in the first cross-section and ending at the last observation in the last cross-section.



The screenshot shows an EViews workfile window titled "Group: UNTITLED Workfile: CELLID::Undated\". The window contains a table with the following data:

| obs | CXID | CELLID | OBSID |
|-------|------|--------|-------|
| A - 1 | 1 | 1 | 1 |
| A - 2 | 1 | 2 | 2 |
| A - 3 | 1 | 3 | 3 |
| A - 4 | 1 | 4 | 4 |
| A - 5 | 1 | 7 | 5 |
| B - 1 | 2 | 3 | 1 |
| B - 2 | 2 | 5 | 2 |
| B - 3 | 2 | 6 | 3 |

```
series _id = @obsnum
```

The `@obsnum` keyword allows you to number the observations in the workfile in sequential order from 1 to the total number of observations.

Working with Panel Data

For the most part, you will find working with data in a panel workfile to be identical to working with data in any other workfile. There are, however, some differences in behavior that require discussion. In addition, we describe useful approaches to working with panel data using standard, non panel-specific tools.

Lags and Leads

For the most part, expressions involving lags and leads should operate as expected (see [“Lags, Leads, and Panel Structured Data” on page 207](#) of the *User’s Guide I* for a full discussion). In particular note that lags and leads do not cross group boundaries so that they will never involve data from a different cross-section (*i.e.*, lags of the first observation in a cross-section are always NAs, as are leads of the last observation in a cross-section).

Since EViews automatically sorts your data by cross-section and cell/date ID, observations in a panel dataset are always stacked by cross-section, with the cell IDs sorted within each cross-section. Accordingly, lags and leads within a cross-section are defined over the sorted values of the cell ID. Lags of an observation are always associated with lower value of the cell ID, and leads always involve a higher value (the first lag observation has the next lowest cell ID value and the first lead has the next highest value).

Lags and leads are specified in the usual fashion, using an offset in parentheses. To assign the sum of the first lag of Y and the second lead of X to the series Z, you may use the command:

```
series z = y(-1) + x(2)
```

Similarly, you may use lags to obtain the name of the previous child in household cross-sections. The command:

```
alpha older = childname(-1)
```

assigns to the alpha series OLDER the name of the preceding observation. Note that since lags never cross over cross-section boundaries, the first value of OLDER in a household will be missing.

Panel Samples

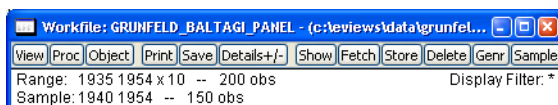
The description of the current workfile sample in the workfile window provides an obvious indication that samples for dated and undated workfiles are specified in different ways.

Dated Panel Samples

For dated workfiles, you may specify panel samples using date pairs to define the earliest and latest dates to be included. For example, in our dated panel example from above, if we issue the sample statement:

```
smp1 1940 1954
```

EViews will exclude all observations that are dated from 1935 through 1939. We see that the new sample has eliminated observations for those dates from each cross-section.



As in non-panel workfiles, you may combine the date specification with additional “if” conditions to exclude additional observations. For example:

```
smp1 1940 1945 1950 1954 if i>50
```

uses any panel observations that are dated from 1940 to 1945 or 1950 to 1954 that have values of the series *I* that are greater than 50.

Additionally, you may use special keywords to refer to the first and last observations for cross-sections. For dated panels, the sample keywords `@first` and `@last` refer to the set of first and last observations for each cross-section. For example, you may specify the sample:

```
smp1 @first 2000
```

to use data from the first observation in each cross-section and observations up through the end of the year 2000. Likewise, the two sample statements:

```
smp1 @first @first+5
```

```
smp1 @last-5 @last
```

use (at most) the first five and the last five observations in each cross-section, respectively.

Note that the included observations for each cross-section may begin at a different date, and that:

```
smp1 @all
```

```
smp1 @first @last
```

are equivalent.

The sample statement keywords `@firstmin` and `@lastmax` are used to refer to the earliest of the start and latest of the end dates observed over all cross-sections, so that the sample:

```
smp1 @firstmin @firstmin+20
```

sets the start date to the earliest observed date, and includes the next 20 observations in each cross-section. The command:

```
smpl @lastmax-20 @lastmax
```

includes the last observed date, and the previous 20 observations in each cross-section.

Similarly, you may use the keywords `@firstmax` and `@lastmin` to refer to the latest of the cross-section start dates, and earliest of the end dates. For example, with *regular* annual data that begin and end at different dates, you may balance the starts and ends of your data using the statement:

```
smpl @firstmax @lastmin
```

which sets the sample to begin at the latest observed start date, and to end at the earliest observed end date.

The special keywords are perhaps most usefully combined with observation offsets. By adding plus and minus terms to the keywords, you may adjust the sample by dropping or adding observations within each cross-section. For example, to drop the first observation from each cross-section, you may use the sample statement:

```
smpl @first+1 @last
```

The following commands generate a series containing cumulative sums of the series `X` for each cross-section:

```
smpl @first @first
series xsum = x
smpl @first+1 @last
xsum = xsum(-1) + x
```

The first two commands initialize the cumulative sum for the first observation in each cross-section. The last two commands accumulate the sum of values of `X` over the remaining observations.

Similarly, if you wish to estimate your equation on a subsample of data and then perform cross-validation on the last 20 observations in each cross-section, you may use the sample defined by,

```
smpl @first @last-20
```

to perform your estimation, and the sample,

```
smpl @last-19 @last
```

to perform your forecast evaluation.

Note that the processing of sample offsets for each cross-section follows the same rules as for non-panel workfiles [“Sample Offsets” on page 90](#) of the *User’s Guide I*.

Undated Panel Samples

For undated workfiles, you must specify the sample range pairs using observation numbers defined over the entire workfile. For example, in our undated 506 observation panel example, you may issue the sample statement:

```
smpl 10 500
```

to drop the first 9 and the last 6 observations in the workfile from the current sample.

One consequence of the use of observation pairs in undated panels is that the keywords `@first`, `@firstmin`, and `@firstmax` all refer to observation 1, and `@last`, `@lastmin`, and `@lastmax`, refer to the last observation in the workfile. Thus, in our example, the command:

```
smpl @first+9 @lastmax-6
```

will also drop the first 9 and the last 6 observations in the workfile from the current sample.

Undated panel sample restrictions of this form are not particularly interesting since they require detailed knowledge of the pattern of observation numbers across those cross-sections. Accordingly, most sample statements in undated workfiles will employ “IF conditions” in place of range pairs.

For example, the sample statement,

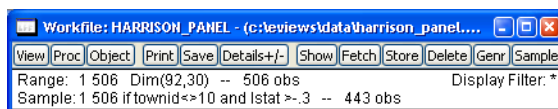
```
smpl if townid<>10 and lstat >-.3
```

is equivalent to either of the commands,

```
smpl @all if townid<>10 and lstat >-.3
```

```
smpl 1 506 if townid<>10 and lstat >-.3
```

and selects all observations with TOWNID values not equal to 10, and LSTAT values greater than -0.3.



You may combine the sample “IF conditions” with the special functions that return information about the observations in the panel. For example, we may use the `@obsid` workfile function to identify each observation in a cross-section, so that:

```
smpl if @obsid>1
```

drops the first observation for each cross-section.

Alternately, to drop the last observation in each cross-section, you may use:

```
smpl if @obsid < @maxsby(townid, townid, "@all")
```

The `@maxsby` function returns the number of non-NA observations for each TOWNID value. Note that we employ the “@ALL” sample to ensure that we compute the `@maxsby` over the entire workfile sample.

Trends

EViews provides several functions that may be used to construct a time trend in your panel structured workfile. A trend in a panel workfile has the property that the values are initialized at the start of a cross-section, increase for successive observations in the specific cross-section, and are reset at the start of the next cross section.

You may use the following to construct your time trend:

- The `@obsid` function may be used to return the simplest notion of a trend in which the values for each cross-section begin at one and increase by one for successive observations in the cross-section.
- The `@trendc` function computes trends in which values for observations with the earliest observed date are normalized to zero, and values for successive observations are incremented based on the calendar associated with the workfile frequency.
- The `@cellid` and `@trend` functions return time trends in which the values increase based on a calendar defined by the observed dates in the workfile.

See also [“Trend Functions” on page 730](#) and [“Panel Trend Functions” on page 731](#) of the *Command Reference* for discussion.

By-Group Statistics

The “by-group” statistical functions ([“By-Group Statistics” on page 749](#) of the *Command Reference*) may be used to compute the value of a statistic for observations in a subgroup, and to assign the computed value to individual observations.

While not strictly panel functions, these tools deserve a place in the current discussion since they are well suited for working with panel data. To use the by-group statistical functions in a panel context, you need only specify the group ID series as the classifier series in the function.

Suppose, for example, that we have the undated panel structured workfile with the group ID series TOWNID, and that you wish to assign to each observation in the workfile the mean value of LSTAT in the corresponding town. You may perform the series assignment using the command,

```
series meanlstat = @meansby(lstat, townid, "@all")
```

or equivalently,

```
series meanlstat = @meansby(lstat, @crossid, "@all")
```

to assign the desired values. EViews will compute the mean value of LSTAT for observations with each TOWNID (or equivalently @crossid, since the workfile is structured using TOWNID) value, and will match merge these values to the corresponding observations.

Likewise, we may use the by-group statistics functions to compute the variance of LSTAT or the number of non-NA values for LSTAT for each subgroup using the assignment statements:

```
series varlstat = @varsby(lstat, townid, "@all")
series nalstat = @nasby(lstat, @crossid, "@all")
```

To compute the statistic over subsamples of the workfile data, simply include a sample string or object as an argument to the by-group statistic, or set the workfile sample prior to issuing the command,

```
smpl @all if zn=0
series meanlstat1 = @meansby(lstat, @cellid)
```

is equivalent to:

```
smpl @all
series meanlstat2 = @meansby(lstat, @cellid, "@all if zn=0")
```

In the former example, the by-group function uses the workfile sample to compute the statistic for each cell ID value, while in the latter, the optional argument explicitly overrides the workfile sample.

One important application of by-group statistics is to compute the “within” deviations for a series by subtracting off panel group means or medians. The following lines:

```
smpl @all
series withinlstat1 = lstat - @meansby(lstat, townid)
series withinlstat2 = lstat - @mediansby(lstat, townid)
```

compute deviations from the TOWNID specific means and medians. In this example, we omit the optional sample argument from the by-group statistics functions since the workfile sample is previously set to use all observations.

Combined with standard EViews tools, the by-group statistics allow you to perform quite complex calculations with little effort. For example, the panel “within” standard deviation for LSTAT may be computed from the single command:

```
scalar within_std = @stdev(lstat - @meansby(lstat, townid, "@all"))
```

while the “between” standard deviation may be calculated from

```
smpl if @obsid = 1
scalar between_std = @stdev(@meansby(lstat, @crossid, "@all"))
```

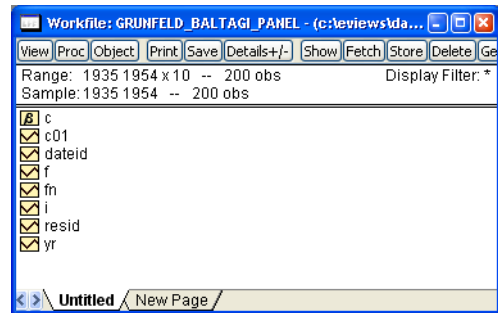

The first line sets the sample to the first observation in each cross-section. The second line calculates the standard deviation of the group means using the single cross-sectional observations. Note that the group means are calculated over the entire sample. An alternative approach to performing this calculation is described in the next section.

Cross-section and Period Summaries

One of the most important tasks in working with panel data is to compute and save summary data, for example, computing means of a series by cross-section or period. In “[By-Group Statistics](#)” on page 521, we outlined tools for computing by-group statistics using the cross-section ID and match merging them back into the original panel workfile page.

Additional tools are available for displaying tables summarizing the by-group statistics or for saving these statistics into new workfile pages.

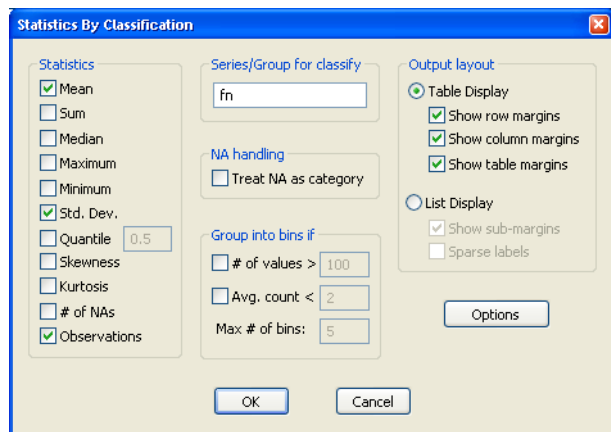
In illustrating these tools, we will work with the familiar Grunfeld data containing data on R&D expenditure and other economic measures for 10 firms for the years 1935 to 1954. These 200 observations form a balanced annual workfile that is structured using the firm number FN as the cross-section ID series, and the date series DATEID to identify the year.



Viewing Summaries

The easiest way to compute by-group statistics is to use the standard by-group statistics view of a series. Simply open the series window and select **View/Descriptive Statistics & Tests/Stats by Classification...** to open the **Statistics by Classification** dialog.

First, you should enter the classifier series in the **Series/Group to classify** edit field. Here, we use FN, so that EViews will compute means, standard deviations, and number of observations for each cross-section in the panel workfile. Note that we have unchecked the **Group into bins** options so that EViews will not combine periods. The result of this computation for the series F is given by:



Descriptive Statistics for F
Categorized by values of FN
Date: 08/22/06 Time: 15:13
Sample: 1935 1954
Included observations: 200

| FN | Mean | Std. Dev. | Obs. |
|-----|----------|-----------|------|
| 1 | 4333.845 | 904.3048 | 20 |
| 2 | 1971.825 | 301.0879 | 20 |
| 3 | 1941.325 | 413.8433 | 20 |
| 4 | 693.2100 | 160.5993 | 20 |
| 5 | 231.4700 | 73.84083 | 20 |
| 6 | 419.8650 | 217.0098 | 20 |
| 7 | 149.7900 | 32.92756 | 20 |
| 8 | 670.9100 | 222.3919 | 20 |
| 9 | 333.6500 | 77.25478 | 20 |
| 10 | 70.92100 | 9.272833 | 20 |
| All | 1081.681 | 1314.470 | 200 |

Alternately, to compute statistics for each period in the panel, you should enter “DATEID” instead of “FN” as the classifier series.

Saving Summaries

Alternately, you may wish to compute the by-group panel statistics and save them in their own workfile page. The standard EViews tools for working with workfiles and creating series links make this task virtually effortless.

Creating Pages for Summaries

Since we will be computing both by-firm and by-period descriptive statistics, the first step is to create workfile pages to hold the results from our two sets of calculations. The firm page will contain observations corresponding to the unique values of the firm identifier found in the panel page; the annual page will contain observations corresponding to the observed years.

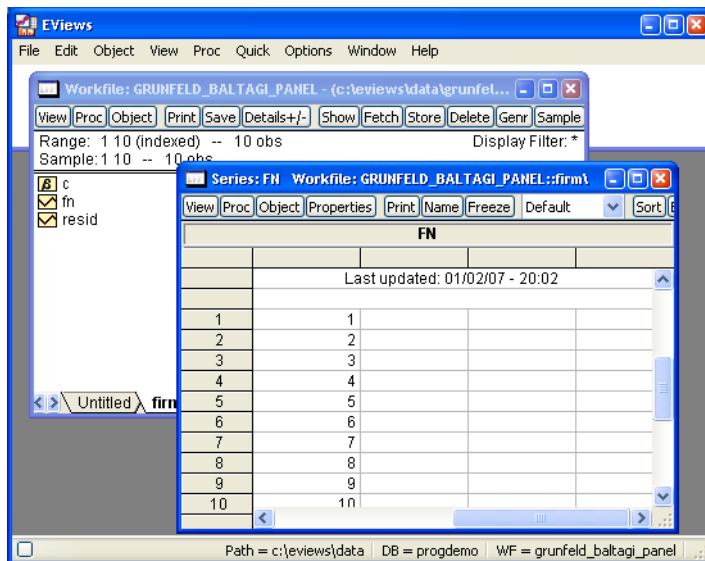
To create a page for the firm data, click on the **New Page** tab in the workfile window, and select **Specify by Identifier series...** EViews opens the **Workfile Page Create by ID** dialog, with the identifiers pre-filled with the series used in the panel workfile structure—the **Date series** field contains the name of the series used to identify dates in the panel, while the **Cross-section ID series** field contains the name of the series used to identify firms.

The default **Method** is set to **Unique values of ID series from one page**, which instructs EViews to simply look at the unique values of the ID series in the specified ID page. Alternatively, you may provide multiple pages and take the union or intersection of IDs (**Union of common ID series from multiple pages** and **Intersection of common ID series from multiple pages**). You may also elect to create observations associated with the crosses of values for multiple series; the different choices permit you to treat date and non-date series asymmetrically when forming these categories (**Cross of two non-date ID series**, **Cross of one date and one non-date ID series**, **Cross of ID series with a date range**). If you select the latter, the dialog will change, prompting you to specify a frequency, start date and end date.

Unique values of ID series from one page
 Union of common ID series from multiple pages
 Intersection of common ID series from multiple pages
 Cross of two non-date ID series
 Cross of one date and one non-date ID series
 Cross of ID series with a date range

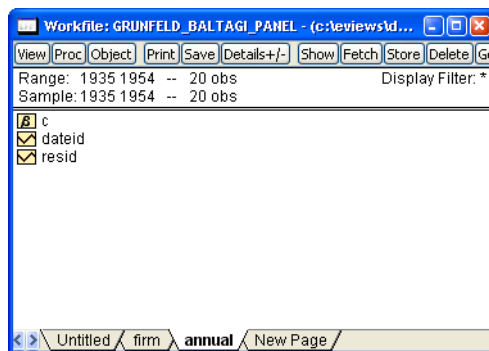
To create a new workfile page using only the values in the FN series, you should delete the **Date series** specification “DATEID” from the dialog. Next, provide a name for the new page by entering “firm” in the **Page** edit field. Now click on **OK**.

EViews will examine the FN series to find its unique values, and will create and structure a workfile page to hold those values.



Here, we see the newly created FIRM page and newly created FN series containing the unique values from FN in the other page. Note that the new page is structured as an **Undated with ID series** page, using the new FN series.

Repeating this process using the DATEID series will create an annual page. First click on the original panel page to make it active, then select **New Page/Specify by Identifier series...** to bring up the previous dialog. Delete the **Cross-section ID series** specification “FN” from the dialog, provide a name for the new page by entering “annual” in the **Page** edit field, and click on **OK**. EViews creates the third page, a regular frequency annual page dated 1935 to 1954.



Computing Summaries using Links

Once the firm and annual pages have been created, it is a simple task to create by-group summaries of the panel data using series links. While links are described elsewhere in greater depth ([Chapter 8. “Series Links,” on page 173 of the User’s Guide I](#)), we provide a brief description of their use in a panel data context.

To create links containing the desired summaries, first click on the original panel page tab to make it active, select one or more series of interest, then right mouse click and select **Copy**. Next, click on either the firm or the annual page, right mouse click, and select **Paste Special....** EViews will open the **Link Dialog**, prompting you to specify a method for summarizing the data.

Suppose, for example, that you select and copy the C01, F, and I series from the panel page and then select **Paste Special...** in the firm page. In this case, EViews analyzes the two pages, and determines that most likely, we wish to match merge the contracted data from the first page into the second page. Accordingly, EViews sets the **Merge by** setting to **General match merge criteria**, and prefills the **Source ID** and **Destination ID** series with two

Paste Special

Paste c01 as
 Pattern: *
 Name: c01

Paste as
☐ Series (by Value)
☒ Link

Merge by
☐ Raw copy
☒ General match merge criteria

Match merge options
 Source ID (one or more series)
 fn
 Destination ID
 fn
☐ Treat NA as ID category
 Contraction method: Mean
 Source sample: @all

OK OK to All Cancel Cancel All

FN cross-section ID series. The default **Contraction method** is set to compute means by ID.

You may provide a different pattern to be used in naming the link series, a contraction method, and a sample over which the contraction should be calculated. Here, we create new series with the same names as the originals, computing means over the entire sample in the panel page. Click on **OK to All** to link all three series into the firm page, yielding:

show c01 fi

Workfile: GRUNFELD_BALTAGI_PANEL - (c:\eviews\d...

Range: 1 10 (indexed) -- 10 obs
 Sample: 1 10 -- 10 obs

Group: UNTITLED Workfile: GRUNFELD_BALT...

| obs | C01 | F | I |
|-----|--------|---------|--------|
| 1 | 648.44 | 4333.85 | 608.02 |
| 2 | 294.86 | 1971.83 | 410.48 |
| 3 | 400.16 | 1941.33 | 102.29 |
| 4 | 121.25 | 693.21 | 86.12 |
| 5 | 486.77 | 231.47 | 61.80 |
| 6 | 104.29 | 419.87 | 55.41 |
| 7 | 314.94 | 149.79 | 47.60 |
| 8 | 85.64 | 670.91 | 42.89 |
| 9 | 297.90 | 333.65 | 41.89 |
| 10 | 5.94 | 70.92 | 3.08 |

Path = c:\eviews\data DB = progdemo WF = grunfeld_baltagi_panel

You may compute other summary statistics by repeating the copy-and-paste-special procedure using alternate contraction methods. For example, selecting the **Standard Deviation** contraction computes the standard deviation for each cross-section and specified series and uses the linking to merge the results into the firm page. Saving them using the pattern “*SD” will create links named “C01SD”, “FSD”, and “ISD”.

Likewise, to compute summary statistics across cross-sections for each year, first create an annual page using **New Page/Specify by Identifier series...**, then paste-special the panel page series as links in the annual page.

Merging Data into the Panel

To merge data into the panel, simply create links from other pages into the panel page. Linking from the annual page into the panel page will repeat observations for each year across firms. Similarly, linking from the cross-section firm page to the panel page will repeat observations for each firm across all years.

In our example, we may link the FSD link from the firm page back into the panel page. Select FSD, switch to the panel page, and paste-special. Click **OK** to accept the defaults in the **Paste Special** dialog.

EViews match merges the data from the firm page to the panel page, matching FN values. Since the merge is from one-to-many, EViews simply repeats the values of FSD in the panel page.

Basic Panel Analysis

EViews provides various degrees of support for the analysis of data in panel structured workfiles.

There is a small number of panel-specific analyses that are provided for data in panel structured workfiles. You may use EViews special tools for graphing dated panel data, perform unit root or cointegration tests, or estimate various panel equation specifications.

Alternately, you may apply EViews standard tools for by-group analysis to the stacked data. These tools do not use the panel structure of the workfile, *per se*, but used appropriately, the by-group tools will allow you to perform various forms of panel analysis.

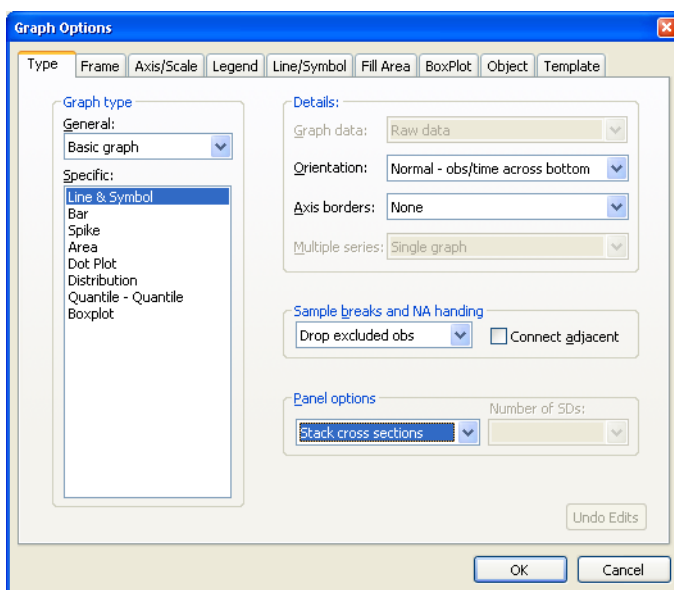
In most other cases, EViews will simply treat panel data as a set of stacked observations. The resulting stacked analysis correctly handles leads and lags in the panel structure, but does not otherwise use the cross-section and cell or period identifiers in the analysis.

Panel-Specific Analysis

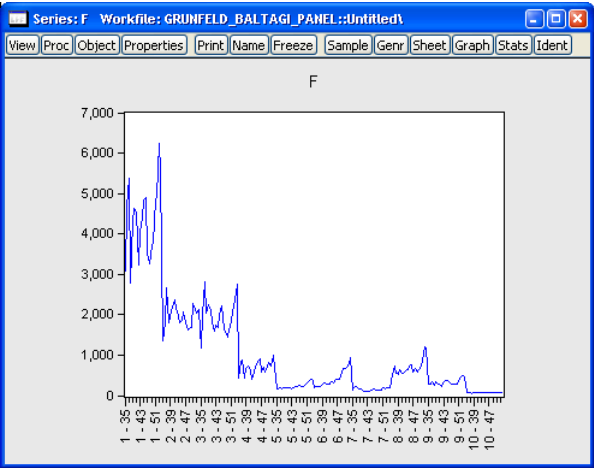
Time Series Graphs

EViews provides tools for displaying time series graphs with panel data. You may use these tools to display a graph of the stacked data, individual or combined graphs for each cross-section, or a time series graph of summary statistics for each period.

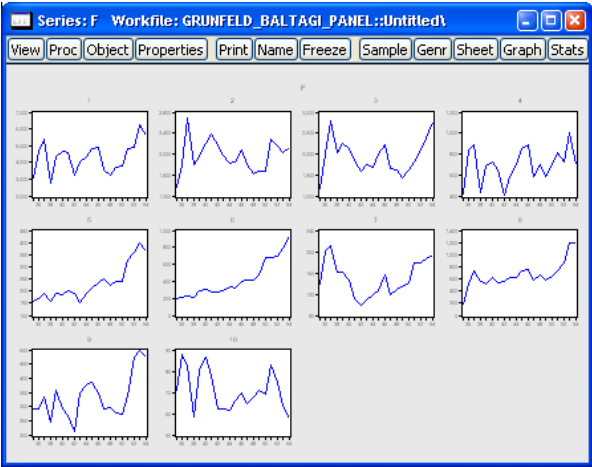
To display panel graphs for a series in a dated workfile, open the series window and click on **View/Graph...** to bring up the **Graph Options** dialog. In the **Panel options** section on the lower right of the dialog, EViews offers you a variety of choices for how you wish to display the data.



If you select **Stack cross sections** and one of the graph types, for example **Line & Symbol**, EViews will display a single graph of the stacked data, labeled with both the cross-section and date.

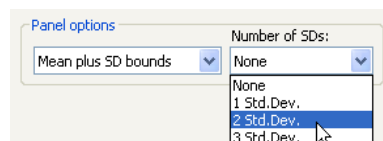


Alternately, selecting **Individual cross sections** displays separate time series graphs for each cross-section, while **Combined cross sections** displays separate lines for each cross-section in a single graph. We caution you that both types of panel graphs may become difficult to read when there are large numbers of cross-sections. For example, the individual graphs for the 10 cross-section panel data depicted here provide information on general trends, but little in the way of detail:

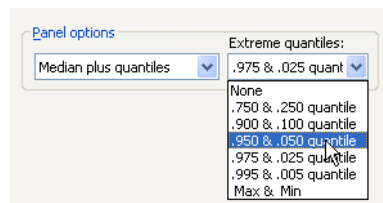


The remaining two options allow you to plot a single graph containing summary statistics for each period.

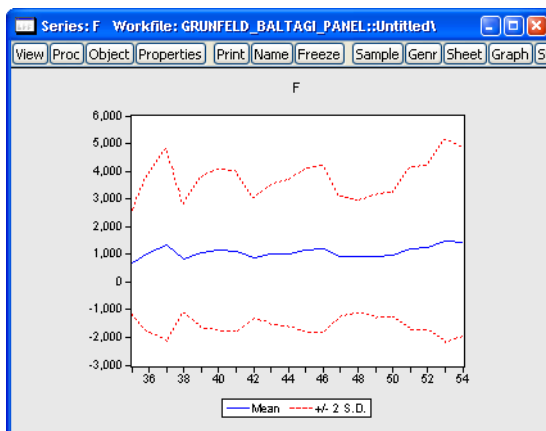
For line graphs, you may select **Mean plus SD bounds**, and then use the drop down menu on the lower right to choose between displaying no bounds, and 1, 2, or 3 standard deviation bounds. For other graph types such as area or spike, you may only display the means of the data by period.



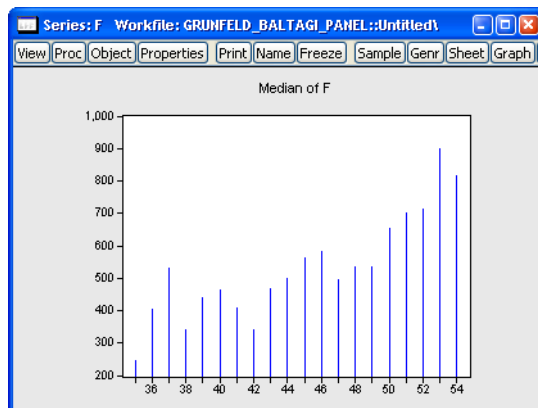
For line graphs you may select **Median plus quantiles**, and then use the drop down menu to choose additional extreme quantiles to be displayed. For other graph types, only the median may be plotted.



Suppose, for example, that we display a line graph containing the mean and 2 standard deviation bounds for the F series. EViews computes, for each period, the mean and standard deviation of F across cross-sections, and displays these in a time series graph:



Similarly, we may display a spike graph of the medians of F for each period:



Displaying graph views of a group object in a panel workfile involves similar choices about the handling of the panel structure.

Panel Unit Root Tests

EViews provides convenient tools for computing panel unit root tests. You may compute one or more of the following tests: Levin, Lin and Chu (2002), Breitung (2000), Im, Pesaran and Shin (2003), Fisher-type tests using ADF and PP tests—Maddala and Wu (1999), Choi (2001), and Hadri (2000).

These tests are described in detail in “Panel Unit Root Tests,” beginning on page 100.

To compute the unit root test on a series, simply select **View/Unit Root Test...** from the menu of a series object.

By default, EViews will compute a **Summary** of all of the first five unit root tests, where applicable, but you may use the combo box in the upper left hand corner to select an individual test statistic.

In addition, you may use the dialog to specify trend and intercept settings, to specify lag length selection, and to provide details on the spectral estimation used in computing the test statistic or statistics.

To begin, we open the F series in our example panel workfile, and accept the defaults to compute the summary of all of the unit root tests on the level of F. The results are given by

Panel unit root test: Summary

Date: 08/22/06 Time: 17:05

Sample: 1935 1954

Exogenous variables: Individual effects

User specified lags at: 1

Newey-West bandwidth selection using Bartlett kernel

Balanced observations for each test

| Method | Statistic | Prob.** | Cross- sections | Obs |
|--|-----------|---------|--------------------|-----|
| Null: Unit root (assumes common unit root process) | | | | |
| Levin, Lin & Chu t* | 1.71727 | 0.9570 | 10 | 180 |
| Null: Unit root (assumes individual unit root process) | | | | |
| Im, Pesaran and Shin W-stat | -0.51923 | 0.3018 | 10 | 180 |
| ADF - Fisher Chi-square | 33.1797 | 0.0322 | 10 | 180 |
| PP - Fisher Chi-square | 41.9742 | 0.0028 | 10 | 190 |

** Probabilities for Fisher tests are computed using an asymptotic Chi-square distribution. All other tests assume asymptotic normality.

Note that there is a fair amount of disagreement in these results as to whether F has a unit root, even within tests that evaluate the same null hypothesis (e.g., Im, Pesaran and Shin vs. the Fisher ADF and PP tests).

To obtain additional information about intermediate results, we may rerun the panel unit root procedure, this time choosing a specific test statistic. Computing the results for the IPS test, for example, displays (in addition to the previous IPS results) ADF test statistic results for each cross-section in the panel:

Intermediate ADF test results

| Cross section | t-Stat | Prob. | E(t) | E(Var) | Lag | Max Lag | Obs |
|------------------|---------|--------|--------|--------|-----|------------|-----|
| 1 | -2.3596 | 0.1659 | -1.511 | 0.953 | 1 | 1 | 18 |
| 2 | -3.6967 | 0.0138 | -1.511 | 0.953 | 1 | 1 | 18 |
| 3 | -2.1030 | 0.2456 | -1.511 | 0.953 | 1 | 1 | 18 |
| 4 | -3.3293 | 0.0287 | -1.511 | 0.953 | 1 | 1 | 18 |
| 5 | 0.0597 | 0.9527 | -1.511 | 0.953 | 1 | 1 | 18 |
| 6 | 1.8743 | 0.9994 | -1.511 | 0.953 | 1 | 1 | 18 |
| 7 | -1.8108 | 0.3636 | -1.511 | 0.953 | 1 | 1 | 18 |
| 8 | -0.5541 | 0.8581 | -1.511 | 0.953 | 1 | 1 | 18 |
| 9 | -1.3223 | 0.5956 | -1.511 | 0.953 | 1 | 1 | 18 |
| 10 | -3.4695 | 0.0218 | -1.511 | 0.953 | 1 | 1 | 18 |
| Average | -1.6711 | | -1.511 | 0.953 | | | |

Panel Cointegration Tests

EViews provides a number of procedures for computing panel cointegration tests. The following tests are available in EViews: Pedroni (1999, 2004), Kao (1999) and Fisher-type test using Johansen's test methodology (Maddala and Wu (1999)). The details of these tests are described in [“Panel Cointegration Testing,” beginning on page 372.](#)

To compute a panel cointegration test, select **Views/Cointegration Test ...** from the menu of an EViews group. You may use various options for specifying the trend specification, lag length selection and spectral estimation methods.

To illustrate, we perform a Pedroni panel cointegration test. The only modification from the default settings that we make is to select **Automatic selection** for lag length. Click on **OK** to accept the settings and perform the test.

Pedroni Residual Cointegration Test
 Series: IVM MM
 Date: 12/13/06 Time: 11:43
 Sample: 1968M01 1995M12
 Included observations: 2688
 Cross-sections included: 8
 Null Hypothesis: No cointegration
 Trend assumption: No deterministic trend
 Lag selection: Automatic SIC with a max lag of 16
 Newey-West bandwidth selection with Bartlett
 kernel

Alternative hypothesis: common AR coefs. (within-dimension)

| | | | Weighted | |
|---------------------|------------------|--------------|------------------|--------------|
| | <u>Statistic</u> | <u>Prob.</u> | <u>Statistic</u> | <u>Prob.</u> |
| Panel v-Statistic | 4.219500 | 0.0001 | 4.119485 | 0.0001 |
| Panel rho-Statistic | -0.400152 | 0.3682 | -2.543473 | 0.0157 |
| Panel PP-Statistic | 0.671083 | 0.3185 | -1.254923 | 0.1815 |
| Panel ADF-Statistic | -0.216806 | 0.3897 | 0.172158 | 0.3931 |

Alternative hypothesis: individual AR coefs. (between-dimension)

| | <u>Statistic</u> | <u>Prob.</u> |
|---------------------|------------------|--------------|
| Group rho-Statistic | -1.776207 | 0.0824 |
| Group PP-Statistic | -0.824320 | 0.2840 |
| Group ADF-Statistic | 0.538943 | 0.3450 |

The top portion of the output indicates the type of test, null hypothesis, exogenous variables, and other test options. The next section provides several Pedroni panel cointegration test statistics which evaluate the null against both the homogeneous and the heterogeneous alternatives. In this case, eight of the eleven statistics do not reject the null hypothesis of no cointegration at the conventional size of 0.05.

The bottom portion of the table reports auxiliary cross-section results showing intermediate calculating used in forming the statistics. For the Pedroni test this section is split into two sections. The first section contains the Phillips-Perron non-parametric results, and the second section presents the Augmented Dickey-Fuller parametric results.

Cross section specific results

Phillips-Peron results (non-parametric)

| Cross ID | AR(1) | Variance | HAC | Bandwidth | Obs |
|----------|-------|----------|----------|-----------|-----|
| AUT | 0.959 | 54057.16 | 46699.67 | 23.00 | 321 |
| BUS | 0.959 | 98387.47 | 98024.05 | 7.00 | 321 |
| CON | 0.966 | 144092.9 | 125609.0 | 4.00 | 321 |
| CST | 0.933 | 579515.0 | 468780.9 | 6.00 | 321 |
| DEP | 0.908 | 896700.4 | 572964.8 | 7.00 | 321 |
| HOA | 0.941 | 146702.7 | 165065.5 | 6.00 | 321 |
| MAE | 0.975 | 2996615. | 2018633. | 3.00 | 321 |
| MIS | 0.991 | 2775962. | 3950850. | 7.00 | 321 |

Augmented Dickey-Fuller results (parametric)

| Cross ID | AR(1) | Variance | Lag | Max lag | Obs |
|----------|-------|----------|-----|---------|-----|
| AUT | 0.983 | 48285.07 | 5 | 16 | 316 |
| BUS | 0.971 | 95843.74 | 1 | 16 | 320 |
| CON | 0.966 | 144092.9 | 0 | 16 | 321 |
| CST | 0.949 | 556149.1 | 1 | 16 | 320 |
| DEP | 0.974 | 647340.5 | 2 | 16 | 319 |
| HOA | 0.941 | 146702.7 | 0 | 16 | 321 |
| MAE | 0.976 | 2459970. | 6 | 16 | 315 |
| MIS | 0.977 | 2605046. | 3 | 16 | 318 |

Estimation

EViews provides sophisticated tools for estimating equations in your panel structured work-file. See [Chapter 39. “Panel Estimation,” beginning on page 541](#) for documentation.

Stacked By-Group Analysis

There are various by-group analysis tools that may be used to perform analysis of panel data. Previously, we considered an example of using by-group tools to examine data in [“Cross-section and Period Summaries” on page 523](#). Standard by-group views may also be to test for equality of means, medians, or variances between groups, or to examine boxplots by cross-section or period.

For example, to compute a test of equality of means for F between firms, simply open the series, then select **View/Descriptive Statistics & Tests/Equality Tests by Classification....** Enter FN in the **Series/Group for Classify** edit field, and select **OK** to continue. EViews will compute and display the results for an ANOVA for F, classifying the data by firm ID. The top portion of the ANOVA results is given by:

Test for Equality of Means of F

Categorized by values of FN

Date: 08/22/06 Time: 17:11

Sample: 1935 1954

Included observations: 200

| Method | df | Value | Probability |
|---------------|--------------|----------|-------------|
| Anova F-test | (9, 190) | 293.4251 | 0.0000 |
| Welch F-test* | (9, 71.2051) | 259.3607 | 0.0000 |

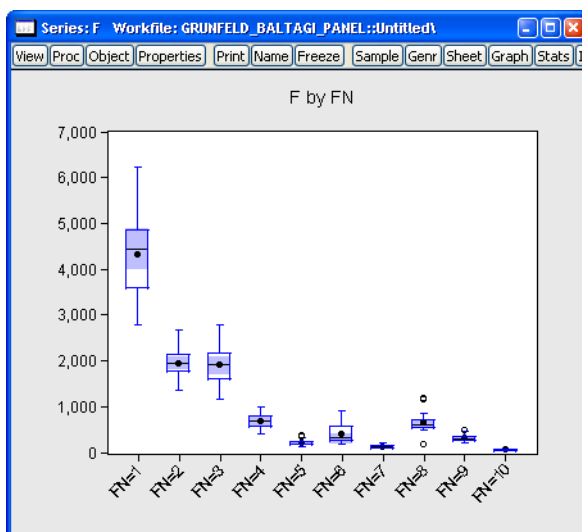
*Test allows for unequal cell variances

Analysis of Variance

| Source of Variation | df | Sum of Sq. | Mean Sq. |
|---------------------|-----|------------|----------|
| Between | 9 | 3.21E+08 | 35640052 |
| Within | 190 | 23077815 | 121462.2 |
| Total | 199 | 3.44E+08 | 1727831. |

Note in this example that we have relatively few cross-sections with moderate numbers of observations in each firm. Data with very large numbers of group identifiers and few observations are not recommended for this type of testing. To test equality of means between periods, call up the dialog and enter either YEAR or DATEID as the series by which you will classify.

A graphical summary of the primary information in the ANOVA may be obtained by displaying boxplots by cross-section or period. For moderate numbers of distinct classifier values, the graphical display may prove informative. **Select View/Graph...** to bring up the **Graph Options** dialog. Select **Categorical graph** from the drop down on the top left, select **Boxplot** from the list of graph types, and enter FN in the **Within graph**



edit field. Click **OK** to display the boxplots using the default settings.

Stacked Analysis

A wide range of analyses are available in panel structured workfiles that have not been specifically redesigned to use the panel structure of your data. These tools allow you to work with and analyze the stacked data, while taking advantage of the support for handling lags and leads in the panel structured workfile.

We may, for example, take our example panel workfile, create a group containing the series C01, F, and the expression $I + I(-1)$, and then select **View/Descriptive Stats/Individual Samples** from the group menu. EViews displays the descriptive statistics for the stacked data.

| | C01 | F | I+I(-1) |
|--------------|----------|----------|----------|
| Mean | 276.0172 | 1081.681 | 289.0423 |
| Median | 205.6000 | 517.9500 | 115.7050 |
| Maximum | 2226.300 | 6241.700 | 2791.100 |
| Minimum | 0.800000 | 58.12000 | 2.110000 |
| Std. Dev. | 301.1039 | 1314.470 | 415.6143 |
| Skewness | 2.766389 | 1.801321 | 2.629995 |
| Kurtosis | 14.71722 | 5.727956 | 11.74612 |
| Jarque-Bera | 1399.206 | 170.1732 | 824.6170 |
| Probability | 0.000000 | 0.000000 | 0.000000 |
| Sum | 55203.43 | 216336.2 | 54918.03 |
| Sum Sq. Dev. | 18042049 | 3.44E+08 | 32646957 |
| Observations | 200 | 200 | 190 |

Note that the calculations are performed over the entire 200 observation stacked data, and that the statistics for $I + I(-1)$ use only 190 observations (200 minus 10 observations corresponding to the lag of the first observation for each firm).

Similarly, suppose you wish to perform a hypothesis testing on a single series. Open the window for the series F, and select **View/Descriptive Statistics & Tests/Simple Hypothesis Tests....** Enter “120” in the edit box for testing the mean value of the stacked series against a null of 120. EViews displays the results of a simple hypothesis test for the mean of the 200 observation stacked data.

| Hypothesis Testing for F | | |
|-------------------------------------|----------|-------------|
| Date: 01/02/07 Time: 20:44 | | |
| Sample: 1935 1954 | | |
| Included observations: 200 | | |
| Test of Hypothesis: Mean = 120.0000 | | |
| Sample Mean = 1081.681 | | |
| Sample Std. Dev. = 1314.470 | | |
| Method | Value | Probability |
| t-statistic | 10.34655 | 0.0000 |

While a wide variety of stacked analyses are supported, various views and procedures are not available in panel structured workfiles. You may not, for example, perform seasonal adjustment or estimate VAR or VEC models with the stacked panel.

References

- Breitung, Jörg (2000). "The Local Power of Some Unit Root Tests for Panel Data," in B. Baltagi (ed.), *Advances in Econometrics, Vol. 15: Nonstationary Panels, Panel Cointegration, and Dynamic Panels*, Amsterdam: JAI Press, p. 161–178.
- Choi, I. (2001). "Unit Root Tests for Panel Data," *Journal of International Money and Finance*, 20: 249–272.
- Fisher, R. A. (1932). *Statistical Methods for Research Workers, 4th Edition*, Edinburgh: Oliver & Boyd.
- Hadri, Kaddour (2000). "Testing for Stationarity in Heterogeneous Panel Data," *Econometric Journal*, 3, 148–161.
- Hlouskova, Jaroslava and M. Wagner (2006). "The Performance of Panel Unit Root and Stationarity Tests: Results from a Large Scale Simulation Study," *Econometric Reviews*, 25, 85–116.
- Holzer, H., R. Block, M. Cheatham, and J. Knott (1993), "Are Training Subsidies Effective? The Michigan Experience," *Industrial and Labor Relations Review*, 46, 625–636.
- Im, K. S., M. H. Pesaran, and Y. Shin (2003). "Testing for Unit Roots in Heterogeneous Panels," *Journal of Econometrics*, 115, 53–74.
- Johansen, Søren (1991). "Estimation and Hypothesis Testing of Cointegration Vectors in Gaussian Vector Autoregressive Models," *Econometrica*, 59, 1551–1580.
- Kao, C. (1999). "Spurious Regression and Residual-Based Tests for Cointegration in Panel Data," *Journal of Econometrics*, 90, 1–44.
- Levin, A., C. F. Lin, and C. Chu (2002). "Unit Root Tests in Panel Data: Asymptotic and Finite-Sample Properties," *Journal of Econometrics*, 108, 1–24.
- Maddala, G. S. and S. Wu (1999). "A Comparative Study of Unit Root Tests with Panel Data and A New Simple Test," *Oxford Bulletin of Economics and Statistics*, 61, 631–52.
- Pedroni, P. (1999). "Critical Values for Cointegration Tests in Heterogeneous Panels with Multiple Regressors," *Oxford Bulletin of Economics and Statistics*, 61, 653–70.
- Pedroni, P. (2004). "Panel Cointegration; Asymptotic and Finite Sample Properties of Pooled Time Series Tests with an Application to the PPP Hypothesis," *Econometric Theory*, 20, 597–625.
- Wooldridge, Jeffrey M. (2002). *Econometric Analysis of Cross Section and Panel Data*, Cambridge, MA: The MIT Press.

Chapter 39. Panel Estimation

EViews allows you to estimate panel equations using linear or nonlinear squares or instrumental variables (two-stage least squares), with correction for fixed or random effects in both the cross-section and period dimensions, AR errors, GLS weighting, and robust standard errors. In addition, GMM tools may be used to estimate most of these specifications with various system-weighting matrices. Specialized forms of GMM also allow you to estimate dynamic panel data specifications. Note that all of the estimators described in this chapter require a panel structured workfile ([“Structuring a Panel Workfile” on page 509](#)).

We begin our discussion by briefly outlining the dialog settings associated with common panel equation specifications. While the wide range of models that EViews supports means that we cannot exhaustively describe all of the settings and specifications, we hope to provide you a roadmap of the steps you must take to estimate your panel equation.

More useful, perhaps, is the discussion that follows, which follows the estimation of some simple panel examples, and describes the use of the wizard for specifying dynamic panel data models.

A background discussion of the supported techniques is provided in [“Estimation Background” in “Pooled Estimation” on page 495](#), and in [“Estimation Background,” beginning on page 570](#).

Estimating a Panel Equation

The first step in estimating a panel equation is to call up an equation dialog by clicking on **Object/New Object.../Equation** or **Quick/Estimate Equation...** from the main menu, or typing the keyword “EQUATION” in the command window. You should make certain that your workfile is structured as a panel workfile. EViews will detect the presence of your panel structure and in place of the standard equation dialog will open the panel **Equation Estimation** dialog.

You should use the **Method** combo box to choose between **LS - Least Squares (and AR)**, ordinary least squares regression, **TOLS - Two-Stage Least Squares (and AR)**, two-stage least squares (instrumental variable) regression, and **GMM / DPD - Generalized Method of Moments / Dynamic Panel Data** techniques. If you select either of the latter two methods, the dialog will be updated to provide you with an additional page for specifying instruments (see [“Instrumental Variables Estimation” on page 544](#)).

Least Squares Estimation

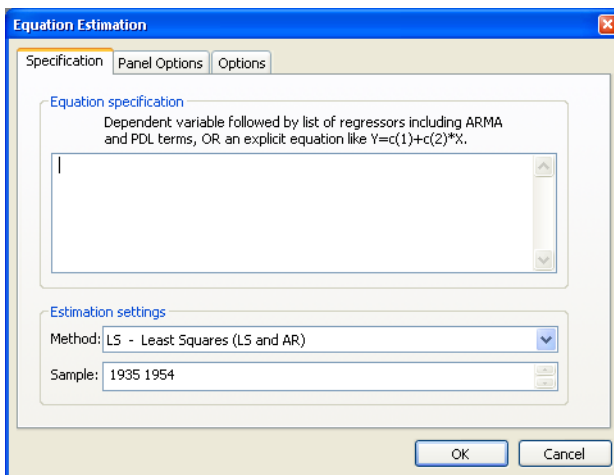
The basic least squares estimation dialog is a multi-page dialog with pages for the basic specification, panel estimation options, and general estimation options.

Least Squares Specification

You should provide an equation specification in the upper **Equation specification** edit box, and an estimation sample in the **Sample** edit box.

The equation may be specified by list or by expression as described in [“Specifying an Equation in EViews”](#) on page 6.

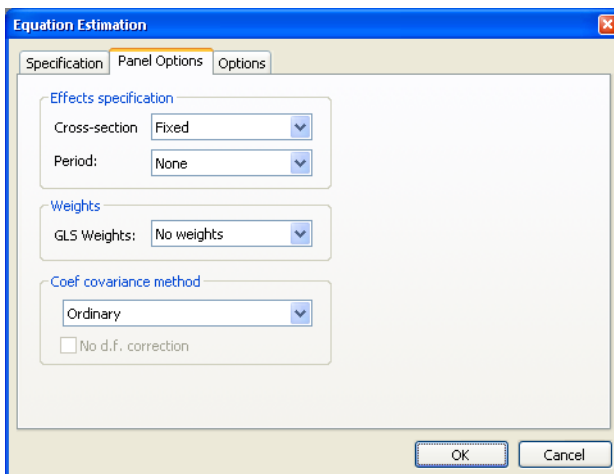
In general, most of the specifications allowed in non-panel equation settings may also be specified here. You may, for example, include AR terms in both linear and nonlinear specifications, and may include PDL terms in equations specified by list. You may not, however, include MA terms in a panel setting.



Least Squares Panel Options

Next, click on the **Panel Options** tab to specify additional panel specific estimation settings.

First, you should account for individual and period effects using the **Effects specification** combo boxes. By default, EViews assumes that there are no effects so that both combo boxes are set to **None**. You may change the default settings to allow for either **Fixed** or **Random** effects in either the cross-section or period dimension, or both. See the pool discussion of [“Fixed and Random Effects”](#) on page 498 for details.



You should be aware that when you select a fixed or random effects specification, EViews will automatically add a constant to the common coefficients portion of the specification if necessary, to ensure that the effects sum to zero.

Next, you should specify settings for **GLS Weights**. You may choose to estimate with no weighting, or with **Cross-section weights**, **Cross-section SUR**, **Period weights**, **Period SUR**. The **Cross-section SUR** setting allows for contemporaneous correlation between cross-sections, while the **Period SUR** allows for general correlation of residuals across periods for a specific cross-section. **Cross-section weights** and **Period weights** allow for heteroskedasticity in the relevant dimension.

| |
|-----------------------|
| No weights |
| Cross-section weights |
| Cross-section SUR |
| Period weights |
| Period SUR |

For example, if you select **Cross section weights**, EViews will estimate a feasible GLS specification assuming the presence of cross-section heteroskedasticity. If you select **Cross-section SUR**, EViews estimates a feasible GLS specification correcting for both cross-section heteroskedasticity and contemporaneous correlation. Similarly, **Period weights** allows for period heteroskedasticity, while **Period SUR** corrects for both period heteroskedasticity and general correlation of observations within a given cross-section. Note that the SUR specifications are both examples of what is sometimes referred to as the Parks estimator. See the pool discussion of [“Generalized Least Squares” on page 499](#) for additional details.

Lastly, you should specify a method for computing coefficient covariances. You may use the combo box labeled **Coef covariance method** to select from the various robust methods available for computing the coefficient standard errors. The covariance calculations may be chosen to be robust under various assumptions, for example, general correlation of observations within a cross-section, or perhaps cross-section heteroskedasticity. Click on the checkbox **No d.f. correction** to perform the calculations without the leading degree of freedom correction term.

| |
|------------------------------|
| Ordinary |
| White cross-section |
| White period |
| White (diagonal) |
| Cross-section SUR (PCSE) |
| Cross-section weights (PCSE) |
| Period SUR (PCSE) |
| Period weights (PCSE) |

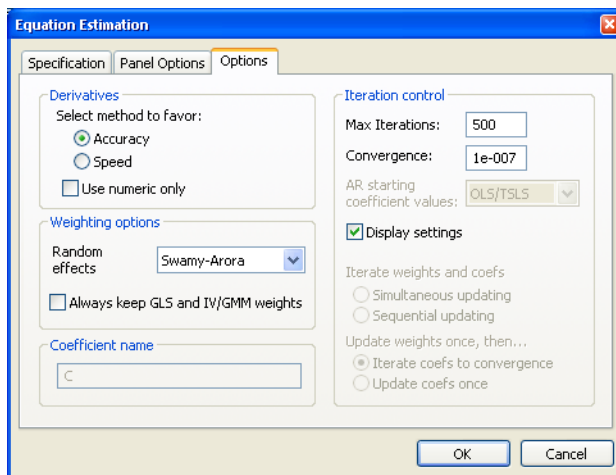
Each of the methods is described in greater detail in [“Robust Coefficient Covariances” on page 504](#) of the pool chapter.

You should note that some combinations of specifications and estimation settings are not currently supported. You may not, for example, estimate random effects models with cross-section specific coefficients, AR terms, or weighting. Furthermore, while two-way random effects specifications are supported for balanced data, they may not be estimated in unbalanced designs.

LS Options

Lastly, clicking on the **Options** tab in the dialog brings up a page displaying computational options for panel estimation. Settings that are not currently applicable will be grayed out.

These options control settings for derivative taking, random effects component variance calculation, coefficient usage, iteration control, and the saving of estimation weights with the equation object.



These options are identical to those found in pool equation estimation, and are described in considerable detail in [“Options” on page 484](#).

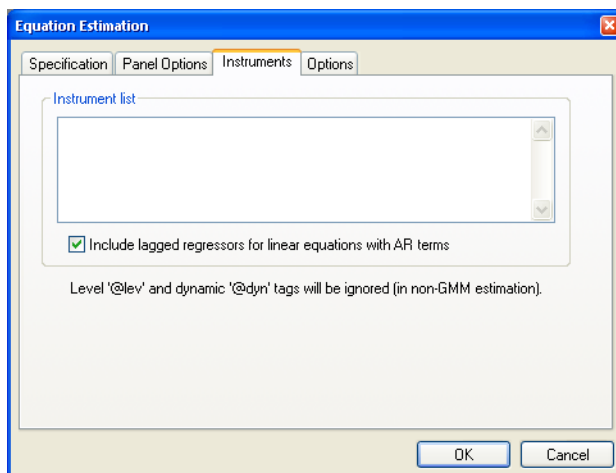
Instrumental Variables Estimation

To estimate a pool specification using instrumental variables techniques, you should select **TSLS - Two-Stage Least Squares (and AR)** in the **Method** combo box at the bottom of the main (**Specification**) dialog page. EViews will respond by creating a four page dialog in which the third page is used to specify your instruments.

While the three original pages are unaffected by this choice of estimation method, note the presence of the new third dialog page labeled **Instruments**, which you will use to specify your instruments. Click on the **Instruments** tab to display the new page.

IV Instrument Specification

There are only two parts to the instrumental variables page. First, in the edit box



labeled **Instrument list**, you will list the names of the series or groups of series you wish to use as instruments.

Next, if your specification contains AR terms, you should use the checkbox to indicate whether EViews should automatically create instruments to be used in estimation from lags of the dependent and regressor variables in the original specification. When estimating an equation specified by list that contains AR terms, EViews transforms the linear model and estimates the nonlinear differenced specification. By default, EViews will add lagged values of the dependent and independent regressors to the corresponding lists of instrumental variables to account for the modified specification, but if you wish, you may uncheck this option.

See the pool chapter discussion [“Instrumental Variables” on page 502](#) for additional detail.

GMM Estimation

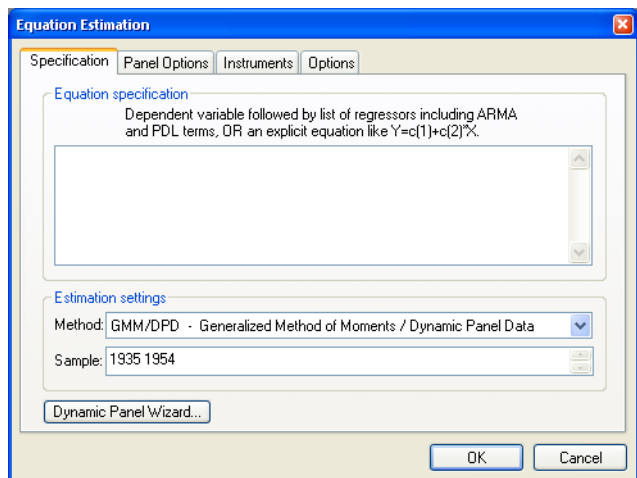
To estimate a panel specification using GMM techniques, you should select **GMM / DPD - Generalized Method of Moments / Dynamic Panel Data** in the **Method** combo box at the bottom of the main (**Specification**) dialog page. Again, you should make certain that your workfile has a panel structure. EViews will respond by displaying a four page dialog that differs significantly from the previous dialogs.

GMM Specification

The specification page is similar to the earlier dialogs. Like the earlier dialogs, you will enter your equation specification in the upper edit box and your sample in the lower edit box.

Note, however, the presence of the **Dynamic Panel Wizard...** button on the bottom of the dialog. Pressing this button opens a wizard that will aid you in filling out the dialog so that you may

employ dynamic panel data techniques such as the Arellano-Bond 1-step estimator for models with lagged endogenous variables and cross-section fixed effects. We will return to this wizard shortly ([“GMM Example” on page 557](#)).



GMM Panel Options

Next, click on the **Panel Options** dialog to specify additional settings for your estimation procedure.

As before, the dialog allows you to indicate the presence of cross-section or period fixed and random effects, to specify GLS weighting, and coefficient covariance calculation methods.

There are, however, notable changes in the available settings.

First, when estimating with GMM, there are two additional choices for handling cross-section fixed effects.

These choices allow you to indicate a transformation method for eliminating the effect from the specification.

You may select **Difference** to indicate that the estimation procedure should use first differenced data (as in Arellano and Bond, 1991), and you may use **Orthogonal Deviations** (Arellano and Bover, 1995) to perform an alternative method of removing the individual effects.

- None
- Fixed
- Random
- Difference
- Orthogonal deviation

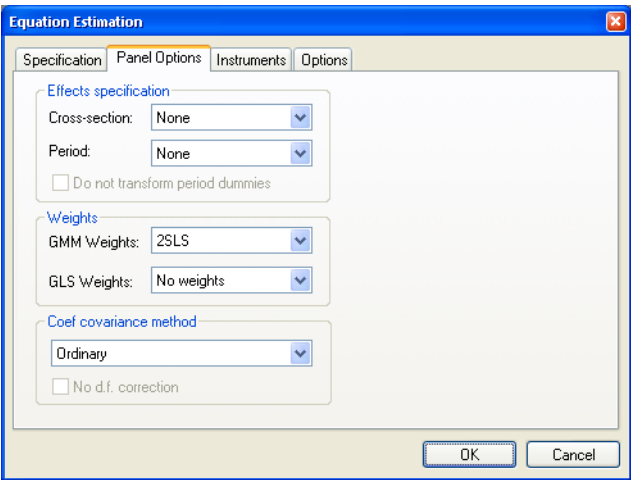
Second, the dialog presents you with a new combo box so that you may specify weighting matrices that may provide for additional efficiency of GMM estimation under appropriate assumptions. Here, the available options depend on other settings in the dialog.

In most cases, you may select a method that computes weights under one of the assumptions associated with the robust covariance calculation methods (see “Least Squares Panel Options” on page 542). If you select **White cross-section**, for example, EVIEWS uses GMM weights that are formed assuming that there is contemporaneous correlation between cross-sections.

- 2SLS
- White cross-section
- White period
- White (diagonal)
- Cross-section SUR
- Cross-section weights
- Period SUR
- Period weights

If, however, you account for cross-section fixed effects by performing first difference estimation, EVIEWS provides you with a modified set of GMM weights choices. In particular, the **Difference (AB 1-step)** weights are those associated with the difference transformation. Selecting these weights allows you to estimate the GMM specification typically referred

- 2SLS
- Difference (AB 1-step)
- White period (AB n-step)
- White (diagonal)
- Cross-section weights
- Period SUR
- Period weights



to as Arellano-Bond 1-step estimation. Similarly, you may choose the **White period (AB 1-step)** weights if you wish to compute Arellano-Bond 2-step or multi-step estimation. Note that the White period weights have been relabeled to indicate that they are typically associated with a specific estimation technique.

Note also that if you estimate your model using difference or orthogonal deviation methods, some GMM weighting methods will no longer be available.

GMM Instruments

Instrument specification in GMM estimation follows the discussion above with a few additional complications.

First, you may enter your instrumental variables as usual by providing the names of series or groups in the edit field. In addition, you may tag instruments as period-specific predetermined instruments, using the `@dyn` keyword, to indicate that the number of implied instruments expands dynamically over time as additional predetermined variables become available.

To specify a set of dynamic instruments associated with the series X , simply enter “`@DYN(X)`” as an instrument in the list. EViews will, by default, use the series $X(-2)$, $X(-3)$, ..., $X(-T)$, as instruments for each period (where available). Note that the default set of instruments grows very quickly as the number of periods increases. With 20 periods, for example, there are 171 implicit instruments associated with a single dynamic instrument. To limit the number of implied instruments, you may use only a subset of the instruments by specifying additional arguments to `@dyn` describing a range of lags to be used.

For example, you may limit the maximum number of lags to be used by specifying both a minimum and maximum number of lags as additional arguments. The instrument specification:

```
@dyn(x, -2, -5)
```

instructs EViews to include lags of X from 2 to 5 as instruments for each period.

If a single argument is provided, EViews will use it as the minimum number of lags to be considered, and will include all higher ordered lags. For example:

```
@dyn(x, -5)
```

includes available lags of X from 5 to the number of periods in the sample.

Second, in specifications estimated using transformations to remove the cross-section fixed effects (first differences or orthogonal deviations), you may use the `@lev` keyword to instruct EViews to use the instrument in untransformed, or level form. Tagging an instrument with “`@LEV`” indicates that the instrument is for the transformed equation. If `@lev` is not provided, EViews will transform the instrument to match the equation transformation.

If, for example, you estimate an equation that uses orthogonal deviations to remove a cross-section fixed effect, EViews will, by default, compute orthogonal deviations of the instruments provided prior to their use. Thus, the instrument list:

```
z1 z2 @lev(z3)
```

will use the transformed Z1 and Z2, and the original Z3 as the instruments for the specification.

Note that in specifications where `@dyn` and `@lev` keywords are not relevant, they will be ignored. If, for example, you first estimate a GMM specification using first differences with both dynamic and level instruments, and then re-estimate the equation using LS, EViews will ignore the keywords, and use the instruments in their original forms.

GMM Options

Lastly, clicking on the **Options** tab in the dialog brings up a page displaying computational options for GMM estimation. These options are virtually identical to those for both LS and IV estimation (see “[LS Options](#)” on page 544). The one difference is in the option for saving estimation weights with the object. In the GMM context, this option applies to both the saving of GLS as well as GMM weights.

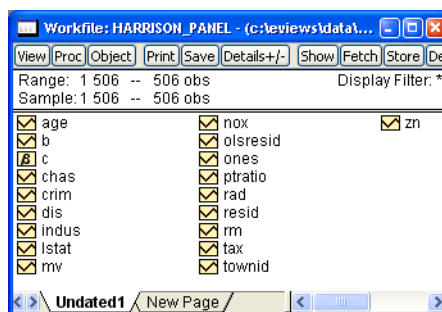
Panel Estimation Examples

Least Squares Examples

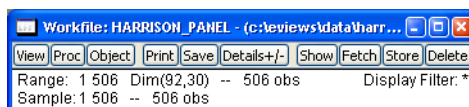
To illustrate the estimation of panel equations in EViews, we first consider an example involving unbalanced panel data from Harrison and Rubinfeld (1978) for the study of hedonic pricing. The data are well known and used as an example dataset in many sources (*e.g.*, Baltagi (2001), p. 166).

The data consist of 506 census tract observations on 92 towns in the Boston area with group sizes ranging from 1 to 30. The dependent variable of interest is the logarithm of the median value of owner occupied houses (MV), and the regressors include various measures of housing desirability.

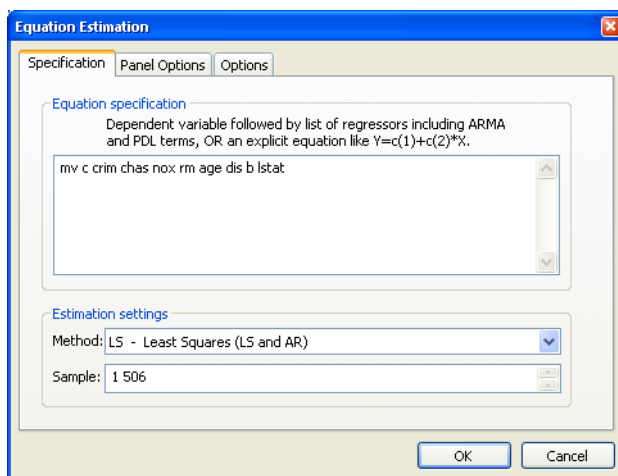
We begin our example by structuring our workfile as an undated panel. Click on the “Range:” description in the workfile window, select **Undated Panel**, and enter “TOWNID” as the **Identifier series**. EViews will prompt you twice to create a CELLID series to uniquely identify observations. Click on **OK** to both questions to accept your settings.



EViews restructures your workfile so that it is an unbalanced panel workfile. The top portion of the workfile window will change to show the undated structure which has 92 cross-sections and a maximum of 30 observations in a cross-section.



Next, we open the equation specification dialog by selecting **Quick/Estimate Equation** from the main EViews menu.



First, following Baltagi and Chang (1994) (also described in Baltagi, 2001), we estimate a fixed effects specification of a hedonic housing equation. The dependent variable in our specification is the median value MV, and the regressors are the crime rate (CRIM), a dummy variable for the property along Charles River (CHAS), air pollution (NOX), average number of rooms (RM), proportion of older units (AGE), distance from employment centers (DIS), proportion of African-Americans in the population (B), and the proportion of lower status individuals (LSTAT). Note that you may include a constant term C in the specification. Since we are estimating a fixed effects specification, EViews will add one if it is not present so that the fixed effects estimates are relative to the constant term and add up to zero.

Click on the **Panel Options** tab and select **Fixed** for the **Cross-section** effects. To match the Baltagi and Chang results, we will leave the remaining settings at their defaults. Click on **OK** to accept the specification.

Dependent Variable: MV
Method: Panel Least Squares
Date: 08/23/06 Time: 14:29
Sample: 1 506
Periods included: 30
Cross-sections included: 92
Total panel (unbalanced) observations: 506

| | Coefficient | Std. Error | t-Statistic | Prob. |
|-------|-------------|------------|-------------|--------|
| C | 8.993272 | 0.134738 | 66.74632 | 0.0000 |
| CRIM | -0.625400 | 0.104012 | -6.012746 | 0.0000 |
| CHAS | -0.452414 | 0.298531 | -1.515467 | 0.1304 |
| NOX | -0.558938 | 0.135011 | -4.139949 | 0.0000 |
| RM | 0.927201 | 0.122470 | 7.570833 | 0.0000 |
| AGE | -1.406955 | 0.486034 | -2.894767 | 0.0040 |
| DIS | 0.801437 | 0.711727 | 1.126045 | 0.2608 |
| B | 0.663405 | 0.103222 | 6.426958 | 0.0000 |
| LSTAT | -2.453027 | 0.255633 | -9.595892 | 0.0000 |

Effects Specification

Cross-section fixed (dummy variables)

| | | | |
|--------------------|----------|-----------------------|-----------|
| R-squared | 0.918370 | Mean dependent var | 9.942268 |
| Adjusted R-squared | 0.898465 | S.D. dependent var | 0.408758 |
| S.E. of regression | 0.130249 | Akaike info criterion | -1.063668 |
| Sum squared resid | 6.887683 | Schwarz criterion | -0.228384 |
| Log likelihood | 369.1080 | Hannan-Quinn criter. | -0.736071 |
| F-statistic | 46.13805 | Durbin-Watson stat | 1.999986 |
| Prob(F-statistic) | 0.000000 | | |

The results for the fixed effects estimation are depicted here. Note that as in pooled estimation, the reported R-squared and F -statistics are based on the difference between the residuals sums of squares from the estimated model, and the sums of squares from a *single* constant-only specification, not from a fixed-effect-only specification. Similarly, the reported information criteria report likelihoods adjusted for the number of estimated coefficients, including fixed effects. Lastly, the reported Durbin-Watson stat is formed simply by computing the first-order residual correlation on the stacked set of residuals.

We may click on the **Estimate** button to modify the specification to match the Wallace-Hussain random effects specification considered by Baltagi and Chang. We modify the specification to include the additional regressors used in estimation, change the cross-section effects to be estimated as a random effect, and use the **Options** page to set the random effects computation method to Wallace-Hussain.

The top portion of the resulting output is given by:

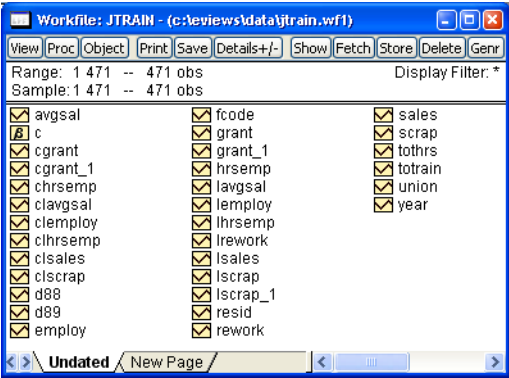
Dependent Variable: MV
 Method: Panel EGLS (Cross-section random effects)
 Date: 08/23/06 Time: 14:34
 Sample: 1 506
 Periods included: 30
 Cross-sections included: 92
 Total panel (unbalanced) observations: 506
 Wallace and Hussain estimator of component variances

| | Coefficient | Std. Error | t-Statistic | Prob. |
|-----------------------|-------------|------------|-------------|--------|
| C | 9.684427 | 0.207691 | 46.62904 | 0.0000 |
| CRIM | -0.737616 | 0.108966 | -6.769233 | 0.0000 |
| ZN | 0.072190 | 0.684633 | 0.105443 | 0.9161 |
| INDUS | 0.164948 | 0.426376 | 0.386860 | 0.6990 |
| CHAS | -0.056459 | 0.304025 | -0.185703 | 0.8528 |
| NOX | -0.584667 | 0.129825 | -4.503496 | 0.0000 |
| RM | 0.908064 | 0.123724 | 7.339410 | 0.0000 |
| AGE | -0.871415 | 0.487161 | -1.788760 | 0.0743 |
| DIS | -1.423611 | 0.462761 | -3.076343 | 0.0022 |
| RAD | 0.961362 | 0.280649 | 3.425493 | 0.0007 |
| TAX | -0.376874 | 0.186695 | -2.018658 | 0.0441 |
| PTRATIO | -2.951420 | 0.958355 | -3.079674 | 0.0022 |
| B | 0.565195 | 0.106121 | 5.325958 | 0.0000 |
| LSTAT | -2.899084 | 0.249300 | -11.62891 | 0.0000 |
| Effects Specification | | | | |
| | | S.D. | Rho | |
| Cross-section random | | 0.126983 | 0.4496 | |
| Idiosyncratic random | | 0.140499 | 0.5504 | |

Note that the estimates of the component standard deviations must be squared to match the component variances reported by Baltagi and Chang (0.016 and 0.020, respectively).

Next, we consider an example of estimation with standard errors that are robust to serial correlation. For this example, we employ data on job training grants used in examples from Wooldridge (2002, p. 276 and 282).

As before, the first step is to structure the workfile as a panel workfile. Click on **Range:** to bring up the dialog, and enter “YEAR” as the date identifier and “FCODE” as the cross-section ID.



EViews will structure the workfile so that it is a panel workfile with 157 cross-sections, and three annual observations. Note that even though there are 471 observations in the workfile, a large number of them contain missing values for variables of interest.



To estimate the fixed effect specification with robust standard errors (Wooldridge example 10.5, p. 276), click on specification **Quick/Estimate Equation** from the main EViews menu. Enter the list specification:

```
lscrap c d88 d89 grant grant_1
```

in the **Equation specification** edit box on the main page and select **Fixed** in the Cross-section effects specification combo box on the **Panel Options** page. Lastly, since we wish to compute standard errors that are robust to serial correlation (Arellano (1987), White (1980)), we choose **White period** as the **Coef covariance method**. To match the reported Wooldridge example, we must select **No d.f. correction** in the covariance calculation.

Equation Estimation

Specification **Panel Options** Options

Effects specification

Cross-section: Fixed

Period: None

Weights

GLS Weights: No weights

Coef covariance method

White period

☒ No d.f. correction

OK Cancel

Click on **OK** to accept the options. EViews displays the results from estimation:

Dependent Variable: LSCRAP

Method: Panel Least Squares

Date: 08/23/06 Time: 14:42

Sample: 1987 1989

Periods included: 3

Cross-sections included: 54

Total panel (balanced) observations: 162

White period standard errors & covariance (no d.f. correction)

| | Coefficient | Std. Error | t-Statistic | Prob. |
|---------|-------------|------------|-------------|--------|
| C | 0.597434 | 0.062489 | 9.560565 | 0.0000 |
| D88 | -0.080216 | 0.095719 | -0.838033 | 0.4039 |
| D89 | -0.247203 | 0.192514 | -1.284075 | 0.2020 |
| GRANT | -0.252315 | 0.140329 | -1.798022 | 0.0751 |
| GRANT_1 | -0.421589 | 0.276335 | -1.525648 | 0.1301 |

Effects Specification

Cross-section fixed (dummy variables)

| | | | |
|--------------------|-----------|-----------------------|----------|
| R-squared | 0.927572 | Mean dependent var | 0.393681 |
| Adjusted R-squared | 0.887876 | S.D. dependent var | 1.486471 |
| S.E. of regression | 0.497744 | Akaike info criterion | 1.715383 |
| Sum squared resid | 25.76593 | Schwarz criterion | 2.820819 |
| Log likelihood | -80.94602 | Hannan-Quinn criter. | 2.164207 |
| F-statistic | 23.36680 | Durbin-Watson stat | 1.996983 |
| Prob(F-statistic) | 0.000000 | | |

Note that EViews automatically adjusts for the missing values in the data. There are only 162 observations on 54 cross-sections used in estimation. The top portion of the output indicates that the results use robust White period standard errors with no d.f. correction.

Alternately, we may estimate a first difference estimator for these data with robust standard errors (Wooldridge example 10.6, p. 282). Open a new equation dialog by clicking on **Quick/Estimate Equation...**, or modify the existing equation by clicking on the **Estimate** button on the equation toolbar. Enter the specification:

```
d(lscrap) c d89 d(grant) d(grant_1)
```

in the **Equation specification** edit box on the main page, select **None** in the Cross-section effects specification combo box, and **White period** without **No d.f. correction** for the coefficient covariance method on the **Panel Options** page. The reported results are given by:

Dependent Variable: D(LSCRAP)

Method: Panel Least Squares

Date: 08/23/06 Time: 14:48

Sample (adjusted): 1988 1989

Periods included: 2

Cross-sections included: 54

Total panel (balanced) observations: 108

White period standard errors & covariance (d.f. corrected)

| | Coefficient | Std. Error | t-Statistic | Prob. |
|--------------------|-------------|-----------------------|-------------|--------|
| C | -0.090607 | 0.089760 | -1.009442 | 0.3151 |
| D89 | -0.096208 | 0.113117 | -0.850519 | 0.3970 |
| D(GRANT) | -0.222781 | 0.131030 | -1.700236 | 0.0921 |
| D(GRANT_1) | -0.351246 | 0.269704 | -1.302339 | 0.1957 |
| R-squared | 0.036518 | Mean dependent var | -0.221132 | |
| Adjusted R-squared | 0.008725 | S.D. dependent var | 0.579248 | |
| S.E. of regression | 0.576716 | Akaike info criterion | 1.773399 | |
| Sum squared resid | 34.59049 | Schwarz criterion | 1.872737 | |
| Log likelihood | -91.76352 | Hannan-Quinn criter. | 1.813677 | |
| F-statistic | 1.313929 | Durbin-Watson stat | 1.498132 | |
| Prob(F-statistic) | 0.273884 | | | |

While current versions of EViews do not provide a full set of specification tests for panel equations, it is a straightforward task to construct some tests using residuals obtained from the panel estimation.

To continue with the Wooldridge example, we may test for AR(1) serial correlation in the first-differenced equation by regressing the residuals from this specification on the lagged residuals using data for the year 1989. First, we save the residual series in the workfile. Click

on **Proc/Make Residual Series...** on the estimated equation toolbar, and save the residuals to the series RESID01.

Next, regress RESID01 on RESID01(-1), yielding:

| Dependent Variable: RESID01 | | | | |
|---|-------------|-----------------------|-------------|----------|
| Method: Panel Least Squares | | | | |
| Date: 08/23/06 Time: 14:50 | | | | |
| Sample (adjusted): 1989 1989 | | | | |
| Periods included: 1 | | | | |
| Cross-sections included: 54 | | | | |
| Total panel (balanced) observations: 54 | | | | |
| | Coefficient | Std. Error | t-Statistic | Prob. |
| RESID01(-1) | 0.236906 | 0.133357 | 1.776481 | 0.0814 |
| R-squared | 0.056199 | Mean dependent var | | 6.17E-18 |
| Adjusted R-squared | 0.056199 | S.D. dependent var | | 0.571061 |
| S.E. of regression | 0.554782 | Akaike info criterion | | 1.677863 |
| Sum squared resid | 16.31252 | Schwarz criterion | | 1.714696 |
| Log likelihood | -44.30230 | Hannan-Quinn criter. | | 1.692068 |
| Durbin-Watson stat | 0.000000 | | | |

Under the null hypothesis that the original idiosyncratic errors are uncorrelated, the residuals from this equation should have an autocorrelation coefficient of -0.5. Here, we obtain an estimate of $\hat{\rho}_1 = 0.237$ which appears to be far from the null value. A formal Wald hypothesis test rejects the null that the original idiosyncratic errors are serially uncorrelated. Perform a Wald test by clicking on **View/Coefficient Tests/Wald-Coefficient Restrictions...** and entering the restriction $C(1) = -0.5$ in the edit box:

| Wald Test: | | | |
|----------------|----------|---------|-------------|
| Equation: EQ01 | | | |
| Test Statistic | Value | df | Probability |
| F-statistic | 30.53460 | (1, 53) | 0.0000 |
| Chi-square | 30.53460 | 1 | 0.0000 |

Null Hypothesis Summary:

| Normalized Restriction (= 0) | Value | Std. Err. |
|------------------------------|----------|-----------|
| 0.5 + C(1) | 0.736906 | 0.133357 |

Restrictions are linear in coefficients.

Instrumental Variables Example

To illustrate the estimation of instrumental variables panel estimators, we consider an example taken from Papke (1994) for enterprise zone data for 22 communities in Indiana that is outlined in Wooldridge (2002, p. 306).

The panel workfile for this example is structured using YEAR as the period identifier, and CITY as the cross-section identifier. The result is a balanced annual panel for dates from 1980 to 1988 for 22 cross-sections.

To estimate the example specification, create a new equation by typing “TSLS” in the command line, or by clicking on **Quick/Estimate Equation...** in the main menu. Selecting **TSLS - Two-Stage Least Squares (and AR)** in the **Method** combo box to display the instrumental variables estimator dialog, if necessary, and enter:

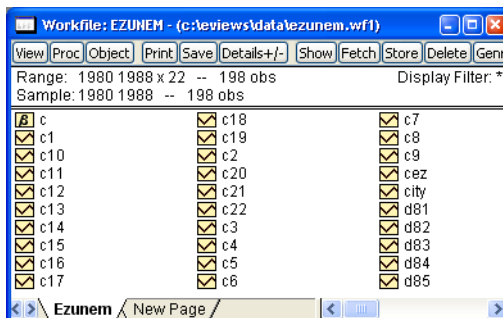
```
d(luclms) c d(luclms(-1)) d(ez)
```

to regress the difference of log unemployment claims (LUCLEMS) on the lag difference, and the difference of enterprise zone designation (EZ). Since the model is estimated with time intercepts, you should click on the **Panel Options** page, and select **Fixed** for the Period effects.

Next, click on the Instruments tab, and add the names:

```
c d(luclms(-2)) d(ez)
```

to the **Instrument list** edit box. Note that adding the constant C to the regressor and instrument boxes is not required since the fixed effects estimator will add it for you. Click on **OK** to accept the dialog settings. EViews displays the output for the IV regression:



Dependent Variable: D(LUCLMS)
 Method: Panel Two-Stage Least Squares
 Date: 08/23/06 Time: 15:52
 Sample (adjusted): 1983 1988
 Periods included: 6
 Cross-sections included: 22
 Total panel (balanced) observations: 132
 Instrument list: C D(LUCLMS(-2)) D(EZ)

| | Coefficient | Std. Error | t-Statistic | Prob. |
|--------------------------------|-------------|--------------------|-------------|--------|
| C | -0.201654 | 0.040473 | -4.982442 | 0.0000 |
| D(LUCLMS(-1)) | 0.164699 | 0.288444 | 0.570992 | 0.5690 |
| D(EZ) | -0.218702 | 0.106141 | -2.060493 | 0.0414 |
| Effects Specification | | | | |
| Period fixed (dummy variables) | | | | |
| R-squared | 0.280533 | Mean dependent var | -0.235098 | |
| Adjusted R-squared | 0.239918 | S.D. dependent var | 0.267204 | |
| S.E. of regression | 0.232956 | Sum squared resid | 6.729300 | |
| F-statistic | 9.223709 | Durbin-Watson stat | 2.857769 | |
| Prob(F-statistic) | 0.000000 | Second-Stage SSR | 6.150596 | |
| Instrument rank | 8.000000 | | | |

Note that the instrument rank in this equation is 8 since the period dummies also serve as instruments, so you have the 3 instruments specified explicitly, plus 5 for the non-collinear period dummy variables.

GMM Example

To illustrate the estimation of dynamic panel data models using GMM, we employ the unbalanced 1031 observation panel of firm level data from Layard and Nickell (1986), previously examined by Arellano and Bond (1991). The analysis fits the log of employment (N) to the log of the real wage (W), log of the capital stock (K), and the log of industry output (YS).

The workfile is structured as a dated annual panel using ID as the cross-section identifier series and YEAR as the date classification series.

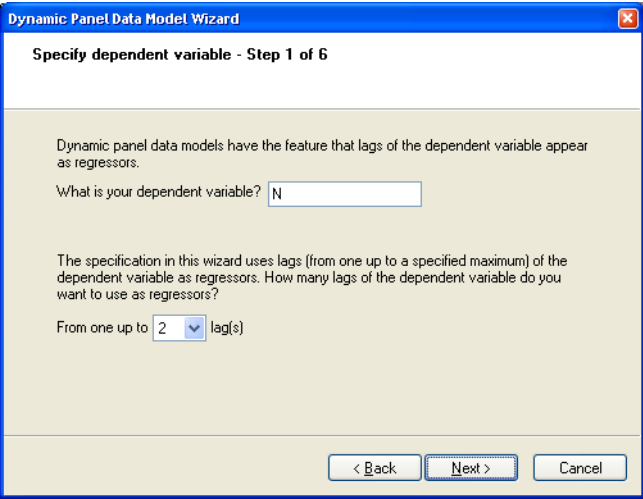
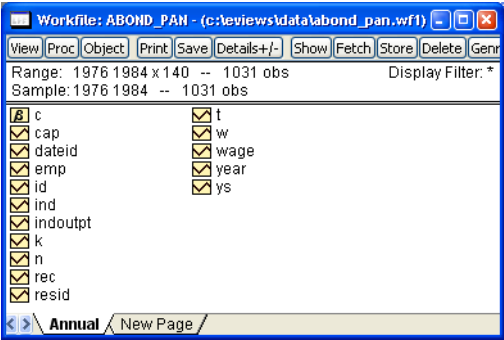
Since the model is assumed to be dynamic, we employ EViews tools for estimating dynamic panel data models. To bring up the GMM dialog, enter “GMM” in the command line, or select **Quick/Estimate Equation...** from the main menu, and choose **GMM/DPD - Generalized Method of Moments /Dynamic Panel Data** in the **Method** combo box to display the IV estimator dialog.

Click on the button labeled **Dynamic Panel Wizard...** to bring up the DPD wizard. The DPD wizard is a tool that will aid you in filling out the general GMM dialog. The first page is an introductory screen describing the basic purpose of the wizard. Click **Next** to continue.

The second page of the wizard prompts you for the dependent variable and the number of its lags to include as explanatory variables. In this example, we wish to estimate an equation with N as the dependent variable and N(-1) and N(-2) as explanatory variables so we enter “N” and select “2” lags in the combo box. Click on **Next** to continue to the next page, where you will specify the remaining explanatory variables.

In the third page, you will complete the specification of your explanatory variables. First, enter the list:

w w(-1) k ys ys(-1)



in the regressor edit box to include these variables. Since the desired specification will include time dummies, make certain that the checkbox for **Include period dummy variables** is selected, then click on **Next** to proceed.

The next page of the wizard is used to specify a transformation to remove the cross-section fixed effect. You may choose to use first **Differences** or **Orthogonal deviations**. In addition, if your specification includes period dummy variables, there is a checkbox asking whether you wish to transform the period dummies, or to enter them in levels. Here we specify the first difference transformation, and choose to include untransformed period dummies in the transformed equation. Click on **Next** to continue.

Dynamic Panel Data Model Wizard

Select transformation method - Step 3 of 6

A transformation is applied to the specification of a dynamic panel model to remove cross-section fixed effects. Select a transformation method.

☒ **Differences**
First difference each variable in the regression. If the innovations are i.i.d., the transformed innovations follow an integrated MA(1) process.

☐ **Orthogonal deviations**
This transformation has the property that if the innovations are i.i.d., the transformed innovations are also i.i.d.

Do you want to apply the transformation to the period dummy variables?
☒ Do not transform period dummy variables

< Back Next > Cancel

The next page is where you will specify your dynamic period-specific (predetermined) instruments. The instruments should be entered with the “@DYN” tag to indicate that they are to be expanded into sets of predetermined instruments, with optional arguments to indicate the lags to be included. If no arguments are provided, the default is to include all valid lags (from -2 to “-infinity”).

Here, we instruct EViews that we wish to use the default lags for N as predetermined instruments.

Dynamic Panel Data Model Wizard

Specify GMM level instruments - Step 4 of 6

Specify Arellano-Bond type dynamic panel instruments with lags that vary by observation.

@dyn(Y,i1,i2) specifies $Y(t-i1), Y(t-i1-1), \dots, Y(t-i2)$ as instruments for observation t . If $i2$ is not specified, all available offsets starting from $i1$ will be used as instruments. If both $i1$ and $i2$ are not specified, $i1$ will be set to -2.

@dyn(N,-2)

Example: @dyn(Y,-2) @dyn(X,-2,-3)

Current specification (dependent followed by regressors):
n n(-1) n(-2) w w(-1) k ys ys(-1)

< Back Next > Cancel

You should now specify the remaining instruments. There are two lists that should be provided. The first list, which is entered in the edit field labeled **Transform**, should contain a list of the strictly exogenous instruments that you wish to transform prior to use in estimating the transformed equation. The second list, which should be entered in the **No transform** edit box should contain a list of instruments that should be used directly without transformation. Enter the remaining instruments:

`w w(-1) k ys ys(-1)`

in the first edit box and click on **Next** to proceed to the final page.

The final page allows you to specify your GMM weighting and coefficient covariance calculation choices. In the first combo box, you will choose a GMM Iteration option. You may select **1-step (for i.i.d. innovations)** to compute the Arellano-Bond 1-step estimator, **2-step (update weights once)**, to compute the Arellano-Bond 2-step estimator, or **n-step (iterate to convergence)**, to iterate the

weight calculations. In the first case, EVIEWS will provide you with choices for computing the standard errors, but here only White period robust standard errors are allowed. Clicking on **Next** takes you to the final page. Click on **Finish** to return to the **Equation Estimation** dialog.

EViews has filled out the **Equation Estimation** dialog with our choices from the DPD wizard. You should take a moment to examine the settings that have been filled out for you since, in the future, you may wish to enter the specification directly into the dialog without using the wizard. You may also, of course, modify the settings in the dialog prior to continuing. For example, click on the **Panel Options** tab and check the **No d.f. correction** setting in the covariance calculation to match the original Arellano-Bond results (Table 4(b), p. 290). Click on **OK** to estimate the specification.

The top portion of the output describes the estimation settings, coefficient estimates, and summary statistics. Note that both the weighting matrix and covariance calculation method used are described in the top portion of the output.

Dependent Variable: N
 Method: Panel Generalized Method of Moments
 Transformation: First Differences
 Date: 08/24/06 Time: 14:21
 Sample (adjusted): 1979 1984
 Periods included: 6
 Cross-sections included: 140
 Total panel (unbalanced) observations: 611
 White period instrument weighting matrix
 White period standard errors & covariance (no d.f. correction)
 Instrument list: @DYN(N, -2) W W(-1) K YS YS(-1)
 @LEV(@SYSPER)

| | Coefficient | Std. Error | t-Statistic | Prob. |
|-------------------------|-------------|------------|-------------|--------|
| N(-1) | 0.474150 | 0.088714 | 5.344699 | 0.0000 |
| N(-2) | -0.052968 | 0.026721 | -1.982222 | 0.0479 |
| W | -0.513205 | 0.057323 | -8.952838 | 0.0000 |
| W(-1) | 0.224640 | 0.080614 | 2.786626 | 0.0055 |
| K | 0.292723 | 0.042243 | 6.929542 | 0.0000 |
| YS | 0.609775 | 0.111029 | 5.492054 | 0.0000 |
| YS(-1) | -0.446371 | 0.125598 | -3.553963 | 0.0004 |
| @LEV(@ISPERIOD("1979")) | 0.010509 | 0.006831 | 1.538482 | 0.1245 |
| @LEV(@ISPERIOD("1980")) | 0.014142 | 0.009924 | 1.425025 | 0.1547 |
| @LEV(@ISPERIOD("1981")) | -0.040453 | 0.012197 | -3.316629 | 0.0010 |
| @LEV(@ISPERIOD("1982")) | -0.021640 | 0.011353 | -1.906127 | 0.0571 |
| @LEV(@ISPERIOD("1983")) | -0.001847 | 0.010807 | -0.170874 | 0.8644 |
| @LEV(@ISPERIOD("1984")) | -0.010221 | 0.010548 | -0.968937 | 0.3330 |

The standard errors that we report here are the standard Arellano-Bond 2-step estimator standard errors. Note that there is evidence in the literature that the standard errors for the two-step estimator may not be reliable.

The bottom portion of the output displays additional information about the specification and summary statistics:

| Effects Specification | | | |
|---|-----------|--------------------|-----------|
| Cross-section fixed (first differences) | | | |
| Period fixed (dummy variables) | | | |
| Mean dependent var | -0.063168 | S.D. dependent var | 0.137637 |
| S.E. of regression | 0.116243 | Sum squared resid | 8.080432 |
| J-statistic | 30.11247 | Instrument rank | 38.000000 |

Note in particular the results labeled “J-statistic” and “Instrument rank”. Since the reported J-statistic is simply the Sargan statistic (value of the GMM objective function at estimated parameters), and the instrument rank of 38 is greater than the number of estimated coefficients (13), we may use it to construct the Sargan test of over-identifying restrictions. It is worth noting here that the J-statistic reported by a panel equation differs from that reported by an ordinary equation by a factor equal to the number of observations. Under the null hypothesis that the over-identifying restrictions are valid, the Sargan statistic is distributed as a $\chi(p - k)$, where k is the number of estimated coefficients and p is the instrument rank. The p -value of 0.22 in this example may be computed using “scalar pval = @chisq(30.11247, 25)”.

Panel Equation Testing

Omitted Variables Test

You may perform an F -test of the joint significance of variables that are presently omitted from a panel or pool equation estimated by list. Select **View/Coefficient Tests/Omitted Variables - Likelihood Ratio...** and in the resulting dialog, enter the names of the variables you wish to add to the default specification. If estimating in a pool setting, you should enter the desired pool or ordinary series in the appropriate edit box (common, cross-section specific, period specific).

When you click on **OK**, EViews will first estimate the unrestricted specification, then form the usual F -test, and will display both the test results as well as the results from the unrestricted specification in the equation or pool window.

Adapting Example 10.6 from Wooldridge (2002, p. 282) slightly, we may first estimate a pooled sample equation for a model of the effect of job training grants on LSCRAP using first differencing. The restricted set of explanatory variables includes a constant and D89. The results from the restricted estimator are given by:

Dependent Variable: D(LSCRAP)
 Method: Panel Least Squares
 Date: 08/24/06 Time: 14:29
 Sample (adjusted): 1988 1989
 Periods included: 2
 Cross-sections included: 54
 Total panel (balanced) observations: 108

| | Coefficient | Std. Error | t-Statistic | Prob. |
|--------------------|-------------|-----------------------|-------------|--------|
| C | -0.168993 | 0.078872 | -2.142622 | 0.0344 |
| D89 | -0.104279 | 0.111542 | -0.934881 | 0.3520 |
| R-squared | 0.008178 | Mean dependent var | -0.221132 | |
| Adjusted R-squared | -0.001179 | S.D. dependent var | 0.579248 | |
| S.E. of regression | 0.579589 | Akaike info criterion | 1.765351 | |
| Sum squared resid | 35.60793 | Schwarz criterion | 1.815020 | |
| Log likelihood | -93.32896 | Hannan-Quinn criter. | 1.785490 | |
| F-statistic | 0.874003 | Durbin-Watson stat | 1.445487 | |
| Prob(F-statistic) | 0.351974 | | | |

We wish to test the significance of the first differences of the omitted job training grant variables GRANT and GRANT_1. Click on **View/Coefficient Tests/Omitted Variables - Likelihood Ratio...** and type “D(GRANT)” and “D(GRANT_1)” to enter the two variables in differences. Click on **OK** to display the omitted variables test results.

The top portion of the results contains a brief description of the test, the test statistic values, and the associated significance levels:

| | | | |
|--|----------|---------------------|--------|
| Omitted Variables: D(GRANT) D(GRANT_1) | | | |
| F-statistic | 1.529525 | Prob. F(2,104) | 0.2215 |
| Log likelihood ratio | 3.130883 | Prob. Chi-Square(2) | 0.2090 |

Here, the test statistics do not reject, at conventional significance levels, the null hypothesis that D(GRANT) and D(GRANT_1) are jointly irrelevant.

The bottom portion of the results shows the test equation which estimates under the unrestricted alternative:

Test Equation:
 Dependent Variable: D(LSCRAP)
 Method: Panel Least Squares
 Date: 08/24/06 Time: 14:31
 Sample: 1988 1989
 Periods included: 2
 Cross-sections included: 54
 Total panel (balanced) observations: 108

| | Coefficient | Std. Error | t-Statistic | Prob. |
|--------------------|-------------|-----------------------|-------------|--------|
| C | -0.090607 | 0.090970 | -0.996017 | 0.3216 |
| D89 | -0.096208 | 0.125447 | -0.766923 | 0.4449 |
| D(GRANT) | -0.222781 | 0.130742 | -1.703970 | 0.0914 |
| D(GRANT_1) | -0.351246 | 0.235085 | -1.494124 | 0.1382 |
| R-squared | 0.036518 | Mean dependent var | -0.221132 | |
| Adjusted R-squared | 0.008725 | S.D. dependent var | 0.579248 | |
| S.E. of regression | 0.576716 | Akaike info criterion | 1.773399 | |
| Sum squared resid | 34.59049 | Schwarz criterion | 1.872737 | |
| Log likelihood | -91.76352 | Hannan-Quinn criter. | 1.813677 | |
| F-statistic | 1.313929 | Durbin-Watson stat | 1.498132 | |
| Prob(F-statistic) | 0.273884 | | | |

Note that if appropriate, the alternative specification will be estimated using the cross-section or period GLS weights obtained from the restricted specification. If these weights were not saved with the restricted specification and are not available, you may first be asked to reestimate the original specification.

Redundant Variables Test

You may perform an F -test of the joint significance of variables that are presently included in a panel or pool equation estimated by list. Select **View/Coefficient Tests/Redundant Variables - Likelihood Ratio...** and in the resulting dialog, enter the names of the variables in the current specification that you wish to remove in the restricted model.

When you click on **OK**, EViews will estimate the restricted specification, form the usual F -test, and will display the test results and restricted estimates. Note that if appropriate, the alternative specification will be estimated using the cross-section or period GLS weights obtained from the unrestricted specification. If these weights were not saved with the specification and are not available, you may first be asked to reestimate the original specification.

To illustrate the redundant variables test, consider Example 10.4 from Wooldridge (2002, p. 262), where we test for the redundancy of GRANT and GRANT_1 in a specification estimated with cross-section random effects. The top portion of the unrestricted specification is given by:

Dependent Variable: LSCRAP
 Method: Panel EGLS (Cross-section random effects)
 Date: 11/24/04 Time: 11:25
 Sample: 1987 1989
 Cross-sections included: 54
 Total panel (balanced) observations: 162
 Swamy and Arora estimator of component variances

| Variable | Coefficient | Std. Error | t-Statistic | Prob. |
|----------|-------------|------------|-------------|--------|
| C | 0.414833 | 0.242965 | 1.707379 | 0.0897 |
| D88 | -0.093452 | 0.108946 | -0.857779 | 0.3923 |
| D89 | -0.269834 | 0.131397 | -2.053577 | 0.0417 |
| UNION | 0.547802 | 0.409837 | 1.336635 | 0.1833 |
| GRANT | -0.214696 | 0.147500 | -1.455565 | 0.1475 |
| GRANT_1 | -0.377070 | 0.204957 | -1.839747 | 0.0677 |

| Effects Specification | | S.D. | Rho |
|-----------------------|--|----------|--------|
| Cross-section random | | 1.390029 | 0.8863 |
| Idiosyncratic random | | 0.497744 | 0.1137 |

Note in particular that our unrestricted model is a random effects specification using Swamy and Arora estimators for the component variances, and that the estimates of the cross-section and idiosyncratic random effects standard deviations are 1.390 and 0.4978, respectively.

If we select the redundant variables test, and perform a joint test on GRANT and GRANT_1, EViews displays the test results in the top of the results window:

| Redundant Variables: GRANT GRANT_1 | | | |
|------------------------------------|----------|----------------|--------|
| F-statistic | 1.832264 | Prob. F(2,156) | 0.1635 |

Here we see that the statistic value of 1.832 does not lead us to reject, at conventional significant levels, the null hypothesis that GRANT and GRANT_1 are redundant in the unrestricted specification.

The restricted test equation results are depicted in the bottom portion of the window. Here we see the top portion of the results for the restricted equation:

Test Equation:
 Dependent Variable: LSCRAP
 Method: Panel EGLS (Cross-section random effects)
 Date: 08/24/06 Time: 15:04
 Sample: 1987 1989
 Periods included: 3
 Cross-sections included: 54
 Total panel (balanced) observations: 162
 Use pre-specified random component estimates
 Swamy and Arora estimator of component variances

| | Coefficient | Std. Error | t-Statistic | Prob. |
|-------|-------------|------------|-------------|--------|
| C | 0.419327 | 0.242949 | 1.725987 | 0.0863 |
| D88 | -0.168993 | 0.095791 | -1.764187 | 0.0796 |
| D89 | -0.442265 | 0.095791 | -4.616981 | 0.0000 |
| UNION | 0.534321 | 0.409752 | 1.304010 | 0.1941 |

| Effects Specification | | | |
|-----------------------|--|----------|--------|
| | | S.D. | Rho |
| Cross-section random | | 1.390029 | 0.8863 |
| Idiosyncratic random | | 0.497744 | 0.1137 |

The first thing to note is that the restricted specification removes the test variables GRANT and GRANT_1. Note further that the output indicates that we are using existing estimates of the random component variances (“Use pre-specified random component estimates”), and that the displayed results for the effects match those for the unrestricted specification.

Fixed Effects Testing

EViews provides built-in tools for testing the joint significance of the fixed effects estimates in least squares specifications. To test the significance of your effects you must first estimate the unrestricted specification that includes the effects of interest. Next, select **View/Fixed/Random Effects Testing/Redundant Fixed Effects – Likelihood Ratio**. EViews will estimate the appropriate restricted specifications, and will display the test output as well as the results for the restricted specifications.

Note that where the unrestricted specification is a two-way fixed effects estimator, EViews will test the joint significance of all of the effects as well as the joint significance of the cross-section effects and the period effects separately.

Let us consider Example 3.6.2 in Baltagi (2001), in which we estimate a two-way fixed effects model. The results for the unrestricted estimated gasoline demand equation are given by:

Dependent Variable: LGASPCAR
 Method: Panel Least Squares
 Date: 08/24/06 Time: 15:32
 Sample: 1960 1978
 Periods included: 19
 Cross-sections included: 18
 Total panel (balanced) observations: 342

| | Coefficient | Std. Error | t-Statistic | Prob. |
|----------|-------------|------------|-------------|--------|
| C | -0.855103 | 0.385169 | -2.220073 | 0.0272 |
| LINCOMEP | 0.051369 | 0.091386 | 0.562103 | 0.5745 |
| LRPMG | -0.192850 | 0.042860 | -4.499545 | 0.0000 |
| LCARPCAP | -0.593448 | 0.027669 | -21.44787 | 0.0000 |

Effects Specification

Cross-section fixed (dummy variables)

Period fixed (dummy variables)

| | | | |
|--------------------|----------|-----------------------|-----------|
| R-squared | 0.980564 | Mean dependent var | 4.296242 |
| Adjusted R-squared | 0.978126 | S.D. dependent var | 0.548907 |
| S.E. of regression | 0.081183 | Akaike info criterion | -2.077237 |
| Sum squared resid | 1.996961 | Schwarz criterion | -1.639934 |
| Log likelihood | 394.2075 | Hannan-Quinn criter. | -1.903027 |
| F-statistic | 402.2697 | Durbin-Watson stat | 0.348394 |
| Prob(F-statistic) | 0.000000 | | |

Note that the specification has both cross-section and period fixed effects. When you select the fixed effect test from the equation menu, EViews estimates three restricted specifications: one with period fixed effects only, one with cross-section fixed effects only, and one with only a common intercept. The test results are displayed at the top of the results window:

Redundant Fixed Effects Tests

Equation: Untitled

Test cross-section and period fixed effects

| Effects Test | Statistic | d.f. | Prob. |
|---------------------------------|------------|----------|--------|
| Cross-section F | 113.351303 | (17,303) | 0.0000 |
| Cross-section Chi-square | 682.635958 | 17 | 0.0000 |
| Period F | 6.233849 | (18,303) | 0.0000 |
| Period Chi-square | 107.747064 | 18 | 0.0000 |
| Cross-Section/Period F | 55.955615 | (35,303) | 0.0000 |
| Cross-Section/Period Chi-square | 687.429282 | 35 | 0.0000 |

Notice that there are three sets of tests. The first set consists of two tests that evaluate the joint significance of the cross-section effects using sums-of-squares (F -test) and the likelihood function (Chi-square test). The corresponding restricted specification is one in which there are period effects only. The two statistic values (113.35 and 682.64) and the associated p -values strongly reject the null that the effects are redundant.

The remaining results evaluate the joint significance of the period effects, and of all of the effects, respectively. All of the results suggest that the corresponding effects are statistically significant.

Below the test statistic results, EViews displays the results for the test equations. In this example, there are three distinct restricted equations so EViews shows three sets of estimates.

Lastly, note that this test statistic is not currently available for instrumental variables and GMM specifications.

Hausman Test for Correlated Random Effects

A central assumption in random effects estimation is the assumption that the random effects are uncorrelated with the explanatory variables. One common method for testing this assumption is to employ a Hausman (1978) test to compare the fixed and random effects estimates of coefficients (for discussion see, for example Wooldridge (2002, p. 288), and Baltagi (2001, p. 65)).

To perform the Hausman test, you must first estimate a model with your random effects specification. Next, select **View/Fixed/Random Effects Testing/Correlated Random Effects - Hausman Test**. EViews will automatically estimate the corresponding fixed effects specifications, compute the test statistics, and display the results and auxiliary equations.

For example, Baltagi (2001) considers an example of Hausman testing (Example 1, p. 69), in which the results for a Swamy-Arora random effects estimator for the Grunfeld data are compared with those obtained from the corresponding fixed effects estimator. To perform this test in EViews 5.1, we first estimate the random effects estimator, obtaining the results:

Dependent Variable: I
 Method: Panel EGLS (Cross-section random effects)
 Date: 08/24/06 Time: 15:44
 Sample: 1935 1954
 Periods included: 20
 Cross-sections included: 10
 Total panel (balanced) observations: 200
 Swamy and Arora estimator of component variances

| | Coefficient | Std. Error | t-Statistic | Prob. |
|---|-------------|------------|-------------|--------|
| C | -57.83441 | 28.88930 | -2.001932 | 0.0467 |
| F | 0.109781 | 0.010489 | 10.46615 | 0.0000 |
| K | 0.308113 | 0.017175 | 17.93989 | 0.0000 |

| Effects Specification | | S.D. | Rho |
|-----------------------|--|----------|--------|
| Cross-section random | | 84.20095 | 0.7180 |
| Idiosyncratic random | | 52.76797 | 0.2820 |

Next we select the Hausman test from the equation menu by clicking on **View/Fixed/Random Effects Testing/Correlated Random Effects - Hausman Test**. EViews estimates the corresponding fixed effects estimator, evaluates the test, and displays the results in the equation window. If the original specification is a two-way random effects model, EViews will test the two sets of effects separately as well as jointly.

There are three parts to the output. The top portion describes the test statistic and provides a summary of the results. Here we have:

Correlated Random Effects - Hausman Test
 Equation: Untitled
 Test cross-section random effects

| Test Summary | Chi-Sq. Statistic | Chi-Sq. d.f. | Prob. |
|----------------------|-------------------|--------------|--------|
| Cross-section random | 2.131366 | 2 | 0.3445 |

The statistic provides little evidence against the null hypothesis that there is no misspecification.

The next portion of output provides additional test detail, showing the coefficient estimates from both the random and fixed effects estimators, along with the variance of the difference and associated p -values for the hypothesis that there is no difference. Note that in some cases, the estimated variances can be negative so that the probabilities cannot be computed.

Cross-section random effects test comparisons:

| Variable | Fixed | Random | Var(Diff.) | Prob. |
|----------|----------|----------|------------|--------|
| F | 0.110124 | 0.109781 | 0.000031 | 0.9506 |
| K | 0.310065 | 0.308113 | 0.000006 | 0.4332 |

The bottom portion of the output contains the results from the corresponding fixed effects estimation:

Cross-section random effects test equation:

Dependent Variable: l

Method: Panel Least Squares

Date: 08/24/06 Time: 15:47

Sample: 1935 1954

Periods included: 20

Cross-sections included: 10

Total panel (balanced) observations: 200

| | Coefficient | Std. Error | t-Statistic | Prob. |
|---|-------------|------------|-------------|--------|
| C | -58.74394 | 12.45369 | -4.716990 | 0.0000 |
| F | 0.110124 | 0.011857 | 9.287901 | 0.0000 |
| K | 0.310065 | 0.017355 | 17.86656 | 0.0000 |

Effects Specification

Cross-section fixed (dummy variables)

In some cases, EViews will automatically drop non-varying variables in order to construct the test statistic. These dropped variables will be indicated in this latter estimation output.

Estimation Background

The basic class of models that can be estimated using panel techniques may be written as:

$$Y_{it} = f(X_{it}, \beta) + \delta_i + \gamma_t + \epsilon_{it} \quad (39.1)$$

The leading case involves a linear conditional mean specification, so that we have:

$$Y_{it} = \alpha + X_{it}'\beta + \delta_i + \gamma_t + \epsilon_{it} \quad (39.2)$$

where Y_{it} is the dependent variable, and X_{it} is a k -vector of regressors, and ϵ_{it} are the error terms for $i = 1, 2, \dots, M$ cross-sectional units observed for dated periods $t = 1, 2, \dots, T$. The α parameter represents the overall constant in the model, while the δ_i and γ_t represent cross-section or period specific effects (random or fixed).

Note that in contrast to the pool specifications described in [Equation \(37.2\) on page 495](#), EViews panel equations allow you to specify equations in general form, allowing for nonlinear coefficients mean equations with additive effects. Panel equations do not automatically allow for β coefficients that vary across cross-sections or periods, but you may, of course, create interaction variables that permit such variation.

Other than these differences, the pool equation discussion of “[Estimation Background](#)” on [page 495](#) applies to the estimation of panel equations. In particular, the calculation of fixed and random effects, GLS weighting, AR estimation, and coefficient covariances for least squares and instrumental variables is equally applicable in the present setting.

Accordingly, the remainder of this discussion will focus on a brief review of the relevant econometric concepts surrounding GMM estimation of panel equations.

GMM Details

The following is a brief review of GMM estimation and dynamic panel estimators. As always, the discussion is merely an overview. For detailed surveys of the literature, see Wooldridge (2002) and Baltagi (2001).

Background

The basic GMM panel estimators are based on moments of the form,

$$g(\beta) = \sum_{i=1}^M g_i(\beta) = \sum_{i=1}^M Z_i' \epsilon_i(\beta) \quad (39.3)$$

where Z_i is a $T_i \times p$ matrix of instruments for cross-section i , and,

$$\epsilon_i(\beta) = (Y_i - f(X_{it}, \beta)) \quad (39.4)$$

In some cases we will work symmetrically with moments where the summation is taken over periods t instead of i .

GMM estimation minimizes the quadratic form:

$$\begin{aligned} S(\beta) &= \left(\sum_{i=1}^M Z_i' \epsilon_i(\beta) \right)' H \left(\sum_{i=1}^M Z_i' \epsilon_i(\beta) \right) \\ &= g(\beta)' H g(\beta) \end{aligned} \quad (39.5)$$

with respect to β for a suitably chosen $p \times p$ weighting matrix H .

Given estimates of the coefficient vector, $\hat{\beta}$, an estimate of the coefficient covariance matrix is computed as,

$$V(\hat{\beta}) = (G' H G)^{-1} (G' H \Lambda H G) (G' H G)^{-1} \quad (39.6)$$

where Λ is an estimator of $E(g_i(\beta)g_i(\beta)') = E(Z_i'\epsilon_i(\beta)\epsilon_i(\beta)'Z_i)$, and G is a $T_i \times k$ derivative matrix given by:

$$G(\beta) = \left(- \sum_{i=1}^M Z_i' \nabla f_i(\beta) \right) \quad (39.7)$$

In the simple linear case where $f(X_{it}, \beta) = X_{it}'\beta$, we may write the coefficient estimator in closed form as,

$$\begin{aligned} \hat{\beta} &= \left(\left(\sum_{i=1}^M Z_i' X_i \right)' H \left(\sum_{i=1}^M Z_i' X_i \right) \right)^{-1} \left(\left(\sum_{i=1}^M Z_i' X_i \right)' H \left(\sum_{i=1}^M Z_i' Y_i \right) \right) \\ &= (M_{ZX}' H M_{ZX})^{-1} (M_{ZX}' H M_{ZY}) \end{aligned} \quad (39.8)$$

with variance estimator,

$$V(\hat{\beta}) = (M_{ZX}' H M_{ZX})^{-1} (M_{ZX}' H \Lambda H M_{ZX}) (M_{ZX}' H M_{ZX})^{-1} \quad (39.9)$$

for M_{AB} of the general form:

$$M_{AB} = M^{-1} \left(\sum_{i=1}^M A_i' B_i \right) \quad (39.10)$$

The basics of GMM estimation involve: (1) specifying the instruments Z , (2) choosing the weighting matrix H , and (3) determining an estimator for Λ .

It is worth pointing out that the summations here are taken over individuals; we may equivalently write the expressions in terms of summations taken over periods. This symmetry will prove useful in describing some of the GMM specifications that EViews supports.

A wide range of specifications may be viewed as specific cases in the GMM framework. For example, the simple 2SLS estimator, using ordinary estimates of the coefficient covariance, specifies:

$$\begin{aligned} H &= (\hat{\sigma}^2 M_{ZZ})^{-1} \\ \Lambda &= \hat{\sigma}^2 M_{ZZ} \end{aligned} \quad (39.11)$$

Substituting, we have the familiar expressions,

$$\begin{aligned} \hat{\beta} &= (M_{ZX}' (\hat{\sigma}^2 M_{ZZ})^{-1} M_{ZX})^{-1} (M_{ZX}' (\hat{\sigma}^2 M_{ZZ})^{-1} M_{ZY}) \\ &= (M_{ZX}' M_{ZZ}^{-1} M_{ZX})^{-1} (M_{ZX}' M_{ZZ}^{-1} M_{ZY}) \end{aligned} \quad (39.12)$$

and,

$$V(\hat{\beta}) = \hat{\sigma}^2 (M_{ZX}' M_{ZZ}^{-1} M_{ZX})^{-1} \quad (39.13)$$

Standard errors that are robust to conditional or unconditional heteroskedasticity and contemporaneous correlation may be computed by substituting a new expression for Λ ,

$$\Lambda = T^{-1} \left(\sum_{t=1}^T Z_t' \hat{\epsilon}_t \hat{\epsilon}_t' Z_t \right) \quad (39.14)$$

so that we have a White cross-section robust coefficient covariance estimator. Additional robust covariance methods are described in detail in [“Robust Coefficient Covariances” on page 504](#).

In addition, EViews supports a variety of weighting matrix choices. All of the choices available for covariance calculation are also available for weight calculations in the standard panel GMM setting: 2SLS, White cross-section, White period, White diagonal, Cross-section SUR (3SLS), Cross-section weights, Period SUR, Period weights. An additional differenced error weighting matrix may be employed when estimating a dynamic panel data specification using GMM.

The formulae for these weights are follow immediately from the choices given in [“Robust Coefficient Covariances” on page 504](#). For example, the Cross-section SUR (3SLS) weighting matrix is computed as:

$$H = \left(T^{-1} \sum_{t=1}^T Z_t' \hat{\Omega}_M Z_t \right)^{-1} \quad (39.15)$$

where $\hat{\Omega}_M$ is an estimator of the contemporaneous covariance matrix. Similarly, the White period weights are given by:

$$H = \left(M^{-1} \sum_{i=1}^M Z_i' \hat{\epsilon}_i \hat{\epsilon}_i' Z_i \right)^{-1} \quad (39.16)$$

These latter GMM weights are associated with specifications that have arbitrary serial correlation and time-varying variances in the disturbances.

GLS Specifications

EViews allows you to estimate a GMM specification on GLS transformed data. Note that the moment conditions are modified to reflect the GLS weighting:

$$g(\beta) = \sum_{i=1}^M g_i(\beta) = \sum_{i=1}^M Z_i' \hat{\Omega}^{-1} \epsilon_i(\beta) \quad (39.17)$$

Dynamic Panel Data

Consider the linear dynamic panel data specification given by:

$$Y_{it} = \sum_{j=1}^p \rho_j Y_{it-j} + X_{it}'\beta + \delta_i + \epsilon_{it} \quad (39.18)$$

First-differencing this specification eliminates the individual effect and produces an equation of the form:

$$\Delta Y_{it} = \sum_{j=1}^p \rho_j \Delta Y_{it-j} + \Delta X_{it}'\beta + \Delta \epsilon_{it} \quad (39.19)$$

which may be estimated using GMM techniques.

Efficient GMM estimation of this equation will typically employ a different number of instruments for each period, with the period-specific instruments corresponding to the different numbers of lagged dependent and predetermined variables available at a given period. Thus, along with any strictly exogenous variables, one may use period-specific sets of instruments corresponding to lagged values of the dependent and other predetermined variables.

Consider, for example, the motivation behind the use of the lagged values of the dependent variable as instruments in [Equation \(39.19\)](#). If the innovations in the original equation are i.i.d., then in $t = 3$, the first period available for analysis of the specification, it is obvious that Y_{i1} is a valid instrument since it is correlated with ΔY_{i2} , but uncorrelated with $\Delta \epsilon_{i3}$. Similarly, in $t = 4$, both Y_{i2} and Y_{i1} are potential instruments. Continuing in this vein, we may form a set of predetermined instruments for individual i using lags of the dependent variable:

$$W_i = \begin{bmatrix} Y_{i1} & 0 & 0 & \dots & \dots & \dots & \dots & 0 \\ 0 & Y_{i1} & Y_{i2} & \dots & \dots & \dots & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & Y_{i1} & Y_{i2} & \dots & Y_{iT_i-2} \end{bmatrix} \quad (39.20)$$

Similar sets of instruments may be formed for each predetermined variables.

Assuming that the ϵ_{it} are not autocorrelated, the optimal GMM weighting matrix for the differenced specification is given by,

$$H^d = \left(M^{-1} \sum_{i=1}^M Z_i' \Xi Z_i \right)^{-1} \quad (39.21)$$

where Ξ is the matrix,

$$\mathbf{E} = \frac{1}{2} \begin{bmatrix} 2 & -1 & 0 & \dots & 0 & 0 \\ -1 & 2 & 0 & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & 2 & -1 \\ 0 & 0 & 0 & \dots & -1 & 2 \end{bmatrix} \sigma^2 \quad (39.22)$$

and where Z_i contains a mixture of strictly exogenous and predetermined instruments. Note that this weighting matrix is the one used in the one-step Arellano-Bond estimator.

Given estimates of the residuals from the one-step estimator, we may replace the H^d weighting matrix with one estimated using computational forms familiar from White period covariance estimation:

$$H = \left(M^{-1} \sum_{i=1}^M Z_i' \Delta \epsilon_i \Delta \epsilon_i' Z_i \right)^{-1} \quad (39.23)$$

This weighting matrix is the one used in the Arellano-Bond two-step estimator.

Lastly, we note that an alternative method of transforming the original equation to eliminate the individual effect involves computing orthogonal deviations (Arellano and Bover, 1995). We will not reproduce the details on here but do note that residuals transformed using orthogonal deviations have the property that the optimal first-stage weighting matrix for the transformed specification is simply the 2SLS weighting matrix:

$$H = \left(M^{-1} \sum_{i=1}^M Z_i' Z_i \right)^{-1} \quad (39.24)$$

References

- Arellano, M. (1987). "Computing Robust Standard Errors for Within-groups Estimators," *Oxford Bulletin of Economics and Statistics*, 49, 431-434.
- Arellano, M., and S. R. Bond (1991). "Some Tests of Specification for Panel Data: Monte Carlo Evidence and an Application to Employment Equations," *Review of Economic Studies*, 58, 277-297.
- Arellano, M., and O. Bover (1995). "Another Look at the Instrumental Variables Estimation of Error-components Models," *Journal of Econometrics*, 68, 29-51.
- Baltagi, Badi H. (2001). *Econometric Analysis of Panel Data, Second Edition*, West Sussex, England: John Wiley & Sons.
- Baltagi, Badi H. and Young-Jae Chang (1994). "Incomplete Panels: A Comparative Study of Alternative Estimators for the Unbalanced One-way Error Component Regression Model," *Journal of Econometrics*, 62, 67-89.
- Harrison, D. and D. L. Rubinfeld (1978). "Hedonic Housing Prices and the Demand for Clean Air," *Journal of Environmental Economics and Management*, 5, 81-102.

- Hausman, Jerry A. (1978). "Specification Tests in Econometrics," *Econometrica*, 46, 1251–1272.
- Layard, R. and S. J. Nickell (1986). "Unemployment in Britain," *Economica*, 53, S121–S169.
- Papke, L. E. (1994). "Tax Policy and Urban Development: Evidence From the Indiana Enterprise Zone Program," *Journal of Public Economics*, 54, 37–49.
- White, Halbert (1980). "A Heteroskedasticity-Consistent Covariance Matrix and a Direct Test for Heteroskedasticity," *Econometrica*, 48, 817–838.
- Wooldridge, Jeffrey M. (2002). *Econometric Analysis of Cross Section and Panel Data*, Cambridge, MA: The MIT Press.

Part VIII. Other Multivariate Analysis

The following chapter describes tools for multivariate analysis using factor analysis:

- [Chapter 40. “Factor Analysis,” on page 579.](#)

Note that several other sections discuss methods of multivariate analysis. Equation based estimation is described in:

- [Chapter 33. “System Estimation,” on page 307.](#)
- [Chapter 34. “Vector Autoregression and Error Correction Models,” on page 345.](#)
- [Chapter 35. “State Space Models and the Kalman Filter,” on page 383.](#)

General tools for multivariate analysis using the group object, including covariance analysis and principal components, are discussed in:

- [Chapter 12. “Groups,” beginning on page 367](#) of the *User’s Guide I*.

Chapter 40. Factor Analysis

Exploratory factor analysis is a method for explaining the covariance relationships amongst a number of *observed* variables in terms of a much smaller number of *unobserved* variables, termed factors.

EViews provides a wide range of tools for performing factor analysis, from computing the covariance matrix from raw data all the way through the construction of factor score estimates.

Factor analysis in EViews is carried out using the factor object. The remainder of this chapter describes the use of the EViews factor object to perform exploratory factor analysis. Using the EViews factor object you may:

- Compute covariances, correlations, or other measures of association.
- Specify the number of factors.
- Obtain initial uniqueness estimates.
- Extract (estimate) factor loadings and uniquenesses.
- Examine diagnostics.
- Perform factor rotation.
- Estimate factor scores.

EViews provides a wide range of choices in each of these areas. You may, for example, select from a menu of automatic methods for choosing the number of factors to be retained, or you may specify an arbitrary number of factors. You may estimate your model using principal factors, iterated principal factors, maximum likelihood, unweighted least squares, generalized least squares, and noniterative partitioned covariance estimation (PACE). Once you obtain initial estimates, rotations may be performed using any of more than 30 orthogonal and oblique methods, and factor scores may be estimated in more than a dozen ways.

We begin with a discussion of the process of creating and specifying a factor object and using the object to estimate the model, perform factor rotation, and estimate factor scores. This section assumes some familiarity with the common factor model and the various issues associated with specification, rotation, and scoring. Those requiring additional detail may wish to consult [“Background,” beginning on page 610](#).

Next, we provide an overview of the views, procedures, and data members provided by the factor object, followed by an extended example highlighting selected features.

The remainder of the chapter provides relevant background information on the common factor model. Our discussion is necessarily limited; the literature on factor analysis is extensive, to say the least, and we cannot possibly attempt a comprehensive overview. For those requiring a detailed treatment, Harman's (1976) book length treatment is a standard reference. Other useful surveys include Gorsuch (1983) and Tucker and MacCallum (1977).

Creating a Factor Object

Factor analysis in EViews is carried out using a factor object. You may create and specify the factor object in a number of ways. The easiest methods are:

- Select **Object/New Object** from the workfile menu, choose **Factor**, and enter the specification in the **Factor Specification** dialog.
- Highlight several series, right-click, select **Open/as Factor...**, and enter the specification in the dialog.
- Open an existing group object, select **Proc/Make Factor...**, and enter the specification in the dialog.

You may also use the commands `factor` or `factest` to create and specify your factor object.

Specifying the Model

There are two distinct parts of a factor object specification. The first part of the specification describes which measure of association or dispersion, typically a correlation or covariance matrix, EViews should fit using the factor model. The second part of the specification defines the properties of the factor model.

The dispersion measure of interest is specified using the **Data** tab of the dialog, and the factor model is defined using the **Estimation** tab. The following sections describe these settings in detail.

Data Specification

The first item in the **Data** tab is the **Type** combo box, which is used to specify whether you wish to compute a **Correlation** or **Covariance** matrix from the series data, or to provide a **User-matrix** containing a previously computed measure of association.

Covariance Specification

Here we see the dialog layout when **Correlation** or **Covariance** is selected.

Most of these fields should be familiar from the Covariance Analysis view of a group. Additional details on all of these settings may be found in [“Covariance Analysis,” beginning on page 380](#).

Method

You may use the **Method** combo to specify the calculation method: ordinary Pearson covariances, uncentered covariances, Spearman rank-order covariances, and Kendall’s tau measures of association.

Note that the computation of factor scores ([“Scoring” on page 620](#)) is not supported for factor models fit to Spearman or Kendall’s tau measures. If you wish to compute scores for measures based on these methods you may, however, estimate a factor model fit to a user-specified matrix.

Variables

You should enter the list of series or groups containing series that you wish to employ for analysis.

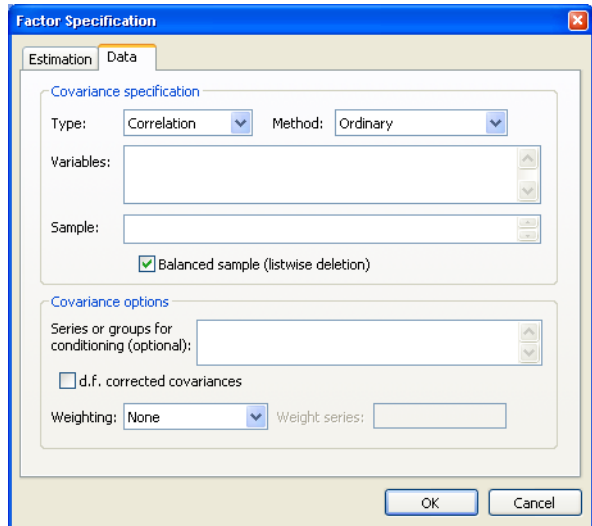
(Note that when you create your factor object from a group object or a set of highlighted series, EVIEWS assumes that you wish to compute a measure of association from the specified series and will initialize the edit field using the series names.)

Sample

You should specify a sample of observations and indicate whether you wish to balance the sample. By default, EVIEWS will perform listwise deletion when it encounters missing values. This option is ignored when performing partial analysis (which may only be computed for balanced samples).

Partialing

Partial covariances or correlations may be computed for each pair of analysis variables by entering a list of conditioning variables in the edit field.



Computation of factor scores is not supported for models fit to partial covariances or correlations. To compute scores for measures in this setting you may, however, estimate a factor model fit to a user-specified matrix.

Weighting

When you specify a weighting method, you will be prompted to enter the name of a weight series. There are five different weight choices: frequency, variance, standard deviation, scaled variance, and scaled standard deviation.

Degrees-of-Freedom Correction

You may choose to compute covariances using the maximum likelihood estimator or the degree-of-freedom corrected formula. By default, EViews computes ML estimates (no d.f. correction) of the covariances. Note that this choice may be relevant even if you will be working with a correlation matrix since standardized data may be used when constructing factor scores.

User-matrix Specification

If you select **User-matrix** in the Type combo, the dialog changes, prompting you for the name of the matrix and optional information for the number of observations, the degrees-of-freedom adjustment, and column names.

- You should specify the name of an EViews matrix object containing the measure of association to be fit. The matrix should be square and symmetric, though it need not be a sym matrix object.
- You may enter a scalar value for the number of observations, or a matrix containing the pairwise numbers of observations. A number of results will not be computed if a number of observations is not provided. If the pairwise number of observations is not constant, EViews will use the minimum number of observations when computing statistics.
- Column names may be provided for labeling results. If not provided, variables will be labeled “V1”, “V2”, *etc.* You need not provide names for all columns; the generic names will be replaced with the specified names in the order they are provided.

The screenshot shows the 'Factor Specification' dialog box with the 'Data' tab selected. Under 'Covariance specification', the 'Type' dropdown is set to 'User-matrix'. Below it are four input fields: 'Matrix:', 'Num. obs. (optional):', 'd.f. adjust: 1', and 'Col. names (optional):'. The 'Matrix:', 'Num. obs. (optional):', and 'Col. names (optional):' fields are currently empty. At the bottom right are 'OK' and 'Cancel' buttons.

Estimation Specification

The main estimation settings are displayed when you click on the **Estimation** tab of the **Factor Specification** dialog. There are four sections in the dialog allowing you to control the method, number of factors, initial communalities, and other options. We describe each in turn.

Method

In the **Method** combo box, you should select your estimation method. EViews supports estimation using **Maximum likelihood**, **Generalized least squares**, **Unweighted least squares**, **Principal factors**, **Iterated principal factors**, and **Partitioned (PACE)** methods.

Depending on the method, different settings may appear in the **Options** section to the right.

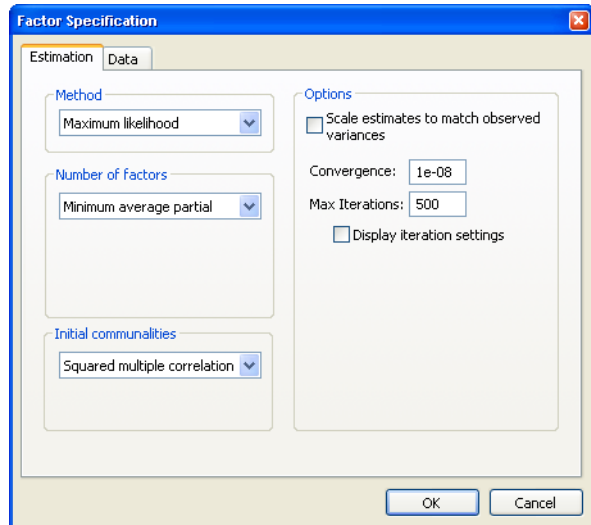
Number of Factors

EViews supports a variety of methods for selecting the number of factors. By default,

EViews uses Velicer's (1976) minimum average partial method (MAP). Simulation evidence suggests that MAP (along with parallel analysis) is more accurate than more commonly used methods such as Kaiser-Guttman (Zwick and Velicer, 1986). See [“Number of Factors,” beginning on page 611](#) for a brief summary of the various methods.

You may change the default by selecting an alternative method from the combo box. The dialog may change to prompt you for additional input:

- The **Minimum eigenvalue** method allows you to employ a modified Kaiser-Guttman rule that uses a different threshold. Simply enter your threshold in the **Cutoff** edit field.
- If you select **Fraction of total variance**, EViews will prompt you to enter the target threshold.
- If you select either **Parallel analysis (mean)** or **Parallel analysis (quantile)** from the combo box, the dialog page will change to provide you with a number of additional options.



Kaiser-Guttman
Minimum eigenvalue
Fraction of total variance
Minimum average partial
Broken stick
Parallel analysis (mean)
Parallel analysis (quantile)
Standard-error scree
User-specified

In the **Number of factor** sections, EViews will prompt you for the number of simulations to run, and, where appropriate, the quantile of the empirical distribution to use for comparison.

By default, EViews compares the eigenvalues of the *reduced* matrix against simulated eigenvalues. This approach is in the spirit of Humphreys and Ilgen (1969), who use the SMC reduced matrix. If you wish to use the eigenvalues of the original (unreduced) matrix, simply check **Use unreduced matrix**.

The **Options** section of the page provides options for the random number generator and the random seed. While the **Random generator** combo should be self-explanatory, the **Seed** field requires some discussion.

By default, the first time that you estimate a given factor model, the **Seed** edit field will be blank; you may provide your own integer value, if desired. If an initial seed is not provided, EViews will randomly select a seed value at estimation time. The value of this initial seed will be saved with the factor object so that by default, subsequent estimation will employ the same seed. If you wish to use a different value, simply enter a new value in the **Seed** edit field or press the **Clear** button to have EViews draw a new random seed value.

- For **User-specified**, you will be prompted to enter the actual number of factors that you wish to employ.

Initial Communalities

Initial estimates of the common variances are required for most estimation methods. For iterative methods like ML and GLS, the initial communalities are simply starting values for the estimation of uniquenesses. For principal factor estimation, the initial communalities are fundamental to the construction of the estimates (see “Principal Factors,” on page 613).

By default, EViews will compute SMC based estimates of the communalities. You may select a different method using the **Initial communalities** combo box. Most of the methods should be self-explanatory; a few require additional comment.

| |
|------------------------------|
| Squared multiple correlation |
| Max absolute correlation |
| Partitioned (PACE) |
| Fraction of diagonals |
| Random diagonal fractions |
| User-specified uniqueness |

- **Partitioned (PACE)** performs a non-iterative PACE estimation of the factor model and uses the fitted estimates of the common variances. The number of factors used is taken from the main estimation settings.
- The **Random diagonal fractions** setting instructs EViews to use a different random fraction of each diagonal element of the original dispersion matrix.
- The **User-specified uniqueness** values will be subtracted from the original variances to form communality estimates. You will specify the name of the vector containing the uniquenesses in the **Vector** edit field. By default, EViews will look at the first elements of the C coefficient vector for uniqueness values.

To facilitate the use of this option, EViews will place the estimated uniqueness values in the coefficient vector C. In addition, you may use the equation data member `@unique` to access the estimated uniqueness from a named factor object.

See “[Communality Estimation](#),” on page 614 for additional discussion.

Estimation Options

We have already seen the iteration control and random number options that are available for various estimation and number of factor methods. The remaining options concern the scaling of results and the handling of Heywood cases.

Scaling

Some estimation methods guarantee that the sums of the uniqueness estimates and the estimated communalities equal the diagonal dispersion matrix elements; for example, principal factors models compute the uniqueness estimates as the residual after accounting for the estimated communalities.

In other cases, the uniqueness and loadings are both estimated directly. In these settings, it is possible for the sum of the components to differ substantively from the original variances.

You can enforce the adding up condition by checking the **Scale estimates to match observed variances** box. If this option is selected, EViews will automatically adjust your uniqueness and loadings estimates so the sum of the unique and common variances matches the diagonals of the dispersion matrix. Note that when scaling has been applied, the reported uniquenesses and loadings will differ from those used to compute fit statistics; the main estimation output will indicate the presence of scaled results.

Heywood Case Handling

In the course of iterating principal factor estimation, one may encounter estimated communalities which implies that at least one unique variance is less than zero; these situations are referred to as *Heywood cases*.

When you encounter a Heywood case in EViews, there are several approaches that you may take. By default, EViews will stop iterating and report the final set of estimates (**Stop and report final**), along with a warning that the results may be inappropriate. Alternately, you may instruct EViews to report the previous iteration's results (**Stop and report last**), to set the results to zero and continue (**Set to zero, continue**), or to ignore the negative unique variance and continue (**Ignore and continue**).

Rotating Factors

You may perform factor rotation on an estimated factor object with two or more retained factors. Simply call up the **Factor Rotation** dialog by clicking on the **Rotate** button or by selecting **Proc/Rotate...** from the factor object menu, and select the desired rotation settings.

The **Type** and **Method** combos may be used to specify the basic rotation method (see “[Types of Rotation](#),” on page 618 for a description of the supported methods). For some methods, you will also be prompted to enter parameter values.

In the depicted example, we specify an oblique Promax rotation with a power parameter of 3.0. The Promax orthogonal pre-rotation step performs Varimax (Orthomax with a parameter of 1).

By default, EViews does not row weight the loadings prior to rotation. To standardize the data, simply change the **Row weight** combo box to **Kaiser** or **Cureton-Mulaik**.

In addition, EViews uses the identity matrix (unrotated loadings) as the default starting value for the rotation iterations. The section labeled **Starting values** allows you to perform different initializations:

- You may instruct EViews to use an initial random rotation by selecting **Random** in the **Starting values** combo. The dialog changes to prompt you to specify the number of

random starting matrices to compare, the random number generator, and the initial seed settings. If you select random, EViews will perform the requested number of rotations, and will use the rotation that minimizes the criterion function.

As with the random number generator used in parallel analysis, the value of this initial seed will be saved with the factor object so that by default, subsequent rotation will employ the same random values. You may override this initialization by entering a value in the **Seed** edit field or press the **Clear** button to have EViews draw a new random seed value.

- You may provide a user-specified initial rotation. Simply select **User-specified** in the **Starting values** combo, then provide the name of a $m \times m$ matrix to be used as the starting T .
- Lastly, if you have previously performed a rotation, you may use the existing results as starting values for a new rotation. You may, for example, perform an oblique Quartimax rotation starting from an orthogonal Varimax solution.

Once you have specified your rotation method you may click on **OK**. EViews will estimate the rotation matrix, and will present a table reporting the rotated loadings, factor correlation, factor rotation matrix, loading rotation matrix, and rotation objective function values. Note that the factor structure matrix is not included in the table output; it may be viewed separately by selecting **View/Structure Matrix** from the factor object menu.

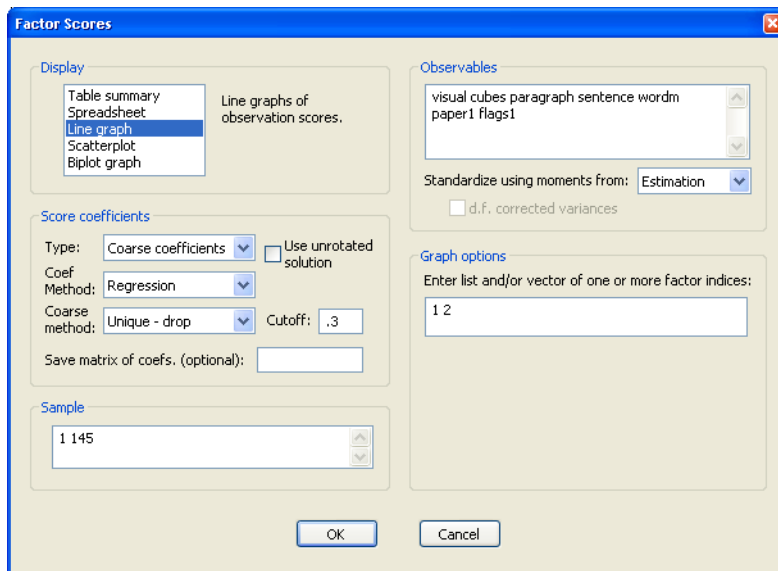
In addition EViews will save the results from the rotation with the factor object. Other routines that rely on estimated loadings such as factor scoring will offer you the option of using the unrotated or the rotated loadings. You may display your rotation results table at any time by selecting **View/Rotation Results** from the factor menu.

Estimating Scores

Factor score estimation may be performed as a factor object view or procedure.

Viewing Scores

To display score coefficients or scores, click on the **Score** button on the factor toolbar, or select **View/Scores...** from the factor menu.



The scores view allows you to display: (1) a table showing the factor score coefficients, indeterminacy and validity indices, and univocality measures; (2) a table of factor score values for a set of observations; (3) a line graph of the scores; (4) scatterplots of scores on pairs of factors; (4) biplots of scores and loadings on pairs of factors.

You should specify the display format by clicking in the list box to choose one of: **Table summary**, **Spreadsheet**, **Line graph**, **Scatterplot**, and **Biplot graph**.

Scores Coefficients

To estimate scores, you must first specify a method for computing the score coefficients. For a brief discussion of methods, see [“Score Estimation” on page 621](#). Details are provided in Gorsuch (1983), Ten Berge et. al (1999), Grice (2001), McDonald (1981), Green (1969).

You must first decide whether to use refined coefficients (**Exact coefficients**), to adjust the refined coefficients (**Coarse coefficients**), or to compute coarse coefficients based on the factor loadings (**Coarse loadings**). By default, EViews will compute scores estimates using exact coefficients.

Next, if rotated factors are available, they will be used as a default. You should check **Use unrotated loadings** to use the original loadings.

Depending on your selections, you will be prompted for additional information:

- If you select **Exact coefficients** or **Coarse coefficients**, EViews will prompt you for a **Coef Method**. You may choose between the following methods: **Regression** (Thurst-

one's regression, **Ideal variables** (Harmon's idealized variables), **Bartlett WLS** (Bartlett weighted least squares), **Anderson-Rubin** (Ten Berge *et al.* generalized Anderson-Rubin-McDonald), and **Green** (Green, MSE minimizing).

- If you select **Coarse coefficients** or **Coarse loadings**, EViews will prompt you for a coarse method and a cutoff value.

Simplified coefficient weights will be computed by recoding elements of the coefficient or loading matrix. In the **Unrestricted** method, values of the matrix that are greater (in absolute value) than some threshold are assigned sign-preserving values of -1 or 1; all other values are recoded at 0.

The two remaining methods restrict the coefficient weights so that each variable loads on a single factor. If you select **Unique - recode**, only the element with the highest absolute value in a row is recoded to a non-zero value; if you select **Unique - drop**, variables with more than loading in excess of the threshold are set to zero.

See Grice (2001) for discussion.

You may instruct EViews to save the matrix of scores coefficients in the workfile by entering a valid EViews object name in the **Save matrix of coeffs** edit field.

Scores Data

You will need to specify a set of observable variables to use in scoring and a sample of observations. The estimated scores will be computed by taking linear combination of the standardized observables over the specified samples.

If available, EViews will fill the **Observables** edit field with the names of the original variables used in computation. You will be prompted for whether to standardize the specified data using the moments obtained from estimation, or whether to standardize the data using the newly computed moments obtained from the data. In the typical case, where we score observations using the same data that we used in estimation, these moments will coincide. When computing scores for observations or variables that differ from estimation, the choice is of considerable importance.

If you have estimated your object from a user-specified matrix, you *must* enter the names of the variables you wish to use as observables. Since moments of the original data are not available in this setting, they will be computed from the specified variables.

Graph Options

When displaying graph views of your results, you will be prompted for which factors to display; by default, EViews will graph all of your factors. Scatterplots and biplots provide additional options for handling multiple graphs, for centering the graph around 0, and for biplot graphs, labeling obs and loading scaling that should be familiar from our discussion of prin-

cipal components (see [“Other Graphs \(Variable Loadings, Component Scores, Biplots\),” beginning on page 401](#)).

Saving Scores

The score procedure allows you to save score values to series in the workfile. When saving scores using the **Proc/Make Scores...**, EViews opens a dialog that differs only slightly from the view dialog. Instead of a **Display** section, EViews provides an **Output specification** section in which you should enter a list of scores to be saved or a list of indices for the scores in the edit field.

To save the first two factors as series AA and BB, you may enter “AA BB” in the edit field. If, instead, you provide the indices “1 2”, EViews will save the first two factors using the default names “F1” and “F2”, unless you have previously named your factors using **Proc/Name Factors...**

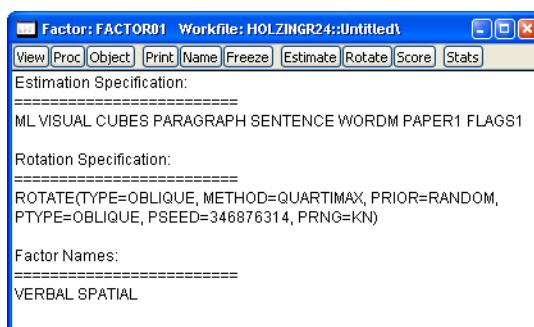
Factor Views

EViews provides a number of factor object views that allow you to examine the properties of your estimated factor model.

Specification

The specification view provides a text representation of the estimation specification, as well as the rotation specifications and assigned factor names (if relevant).

In this example, we see that we have estimated a ML factor model for seven variables, using a convergence criterion of 1e-07. The model was estimated using the default SMCs initial communalities and Velicer’s MAP criterion to select the number of factors.



In addition, the object has a valid rotation method, oblique Quartimax, that was estimated using the default 25 random oblique rotations. If no rotations had been performed, the rotation specification would have read “Factor does not have a valid rotation.”

Lastly, we see that we have provided two factor names, “Verbal”, and “Spatial”, that will be used in place of the default names of the first two factors “F1” and “F2”.

Estimation Output

Select **View/Estimation Output** to display the main estimation output (unrotated loadings, communalities, uniquenesses, variance accounted for by factors, selected goodness-of-fit statistics). Alternately, you may click on the **Stats** toolbar button to display this view.

Rotation Results

Click **View/Rotation Results** to show the output table produced when performing a rotation (rotated loadings, factor correlation, factor rotation matrix, loading rotation matrix, and rotation objective function values).

Goodness-of-fit Summary

Select **View/Goodness-of-fit Summary** to display a table of goodness-of-fit statistics. For models estimated by ML or GLS, EVIEWS computes a large number of absolute and relative fit measures. For details on these measures, see [“Model Evaluation,” beginning on page 615](#).

Matrix Views

You may display spreadsheet views of various matrices of interest. These matrix views are divided into four groups: matrices based on the observed dispersion matrix, matrices based on the reduced matrix, fitted matrices, and residual matrices.

Observed Covariances

You may examine the observed matrices by selecting **View/Observed Covariance Matrix/** and the desired sub-matrix:

- The **Covariance** entry displays the original dispersion matrix, while the **Scaled Covariance matrix** scales the original matrix to have unit diagonals. In the case where the original matrix is a correlation, these two matrices will obviously be the same.
- **Observations** displays a matrix of the number of observations used in each pairwise comparison.
- If you select **Anti-image Covariance**, EVIEWS will display the anti-image covariance of the original matrix. The anti-image covariance is computed by scaling the rows and columns of the inverse (or generalized inverse) of the original matrix by the inverse of its diagonals:

$$A = \text{diag}(S^{-1})^{-1} S^{-1} \text{diag}(S^{-1})^{-1}$$

- **Partial correlations** will display the matrix of partial correlations, where every element represents the partial correlation of the variables conditional on the remaining variables. The partial correlations may be computed by scaling the anti-image covariance to unit diagonals and then performing a sign adjustment.

Reduced Covariance

You may display the initial or final reduced matrices by selecting **View/Reduced Covariance Matrix/** and **Using Initial Uniqueness** or **Using Final Uniqueness**.

Fitted Covariances

To display the fitted covariance matrices, select **View/Fitted Covariance Matrix/** and the desired sub-matrix. **Total Covariance** displays the estimated covariance using both the common and unique variance estimates, while **Common Covariance** displays the estimate of the variance based solely on the common factors.

Residual Covariances

The different residual matrices are based on the total and the common covariance matrix. Select **View/Residual Covariance Matrix/** and the desired matrix, **Using Total Covariance**, or **Using Common Covariance**. The residual matrix computed using the total covariance will generally have numbers close to zero on the main diagonal; the matrix computed using the common covariance will have numbers close to the uniquenesses on the diagonal (see [“Scaling,”](#) on page 585 for caveats).

Factor Structure Matrix

The factor structure matrix reports the correlations between the variables and factors. The correlation is equal to the (possibly rotated) loadings matrix times the factor correlation matrix, $L\Phi$; for orthogonal factors, the structure matrix simplifies so that the correlation is given by the loadings matrix, L .

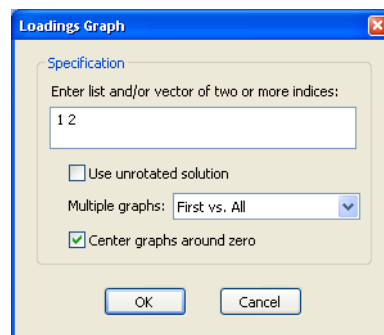
Loadings Views

You may examine your rotated or unrotated loadings in spreadsheet or graphical form.

You may **View/Loadings/Loadings Matrix** to display the current loadings matrix in spreadsheet form. If a rotation has been performed, then this view will show the rotated loadings, otherwise it will display the unrotated loadings. To view the unrotated loadings, you may always select **View/Loadings/Unrotated Loadings Matrix**.

To display the loadings as a graph, select **View/Loadings/Loadings Graph...** The dialog will prompt you for a set of indices for the factors you wish to plot. EViews will produce pairwise plots of the factors, with the loadings displayed as lines from the origin to the points labeled with the variable name.

By default, EViews will use the rotated solution if available; to override this choice, click on the **Use unrotated solution** checkbox.



The other settings allow you to control the handling of multiple graphs, and whether the graphs should be centered around the origin.

Scores

Select **View/Scores...** to compute estimates of factor score coefficients and to compute factor score values for observations. This view and the corresponding procedure are described in detail in [“Estimating Scores,” on page 587](#).

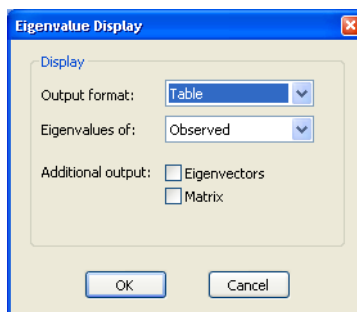
Eigenvalues

One important class of factor model diagnostics is an examination of eigenvalues of the unreduced and the reduced matrices. In addition to being of independent interest, these eigenvalues are central to various methods for selecting the number of factors.

Select **View/Eigenvalues...** to open the **Eigenvalue Display** dialog. By default, EViews will display a table view containing a description of the eigenvalues of the observed dispersion matrix.

The dialog options allow you to control the output format and method of calculation:

- You may change the **Output format** to display a graph of the ordered eigenvalues. By default, EViews will display the resulting Scree plot along with a line representing the mean eigenvalue.
- To base calculations on the scaled observed, initial reduced or final reduced matrix, select the appropriate item in the **Eigenvalues of** combo.
- For table display, you may include the corresponding eigenvectors and dispersion matrix in the output by clicking on the appropriate **Additional output** checkbox.



- For graph display, you may also display the eigenvalue differences, and the cumulative proportion of variance represented by each eigenvalue. The difference graphs also display the mean value of the difference; the cumulative proportion graph shows a reference line with slope equal to the mean eigenvalue.

Additional Views

Additional views allow you to examine:

- The matrix of maximum absolute correlations (**View/Maximum Absolute Correlation**).
- The squared multiple correlations (SMCs) and the related anti-image covariance matrix (**View/Squared Multiple Correlations**).
- The Kaiser-Meyer-Olkin (Kaiser 1970; Kaiser and Rice, 1974; Dziuban and Shirkey, 1974), measure of sampling adequacy (MSA) and corresponding matrix of partial correlations (**View/Kaiser's Measure of Sampling Adequacy**).

The first two views correspond to the calculations used in forming initial communality estimates (see [“Communality Estimation” on page 614](#)). The latter view is an “index of factorial simplicity” that lies between 0 and 1 and indicates the degree to which the data are suitable for common factor analysis. Values for the MSA above 0.90 are deemed “marvelous”; values in the 0.80s are “meritorious”; values in the 0.70s are “middling”; values the 60s are “mediocre”, values in the 0.50s are “miserable”, and all others are “unacceptable” (Kaiser and Rice, 1974).

Factor Procedures

The factor procedures may be accessed either clicking on the **Proc** button on the factor toolbar or by selecting **Proc** from the main factor object menu, and selecting the desired procedure:

- **Specify/Estimate...** is the main procedure for estimating the factor model. When selected, EVIEWS will display the main **Factor Specification** dialog See [“Specifying the Model” on page 580](#).
- **Rotate...** is used to perform factor rotation using the **Factor Rotation** dialog. See [“Rotating Factors” on page 586](#).
- **Make Scores...** is used to save estimated factor scores as series in the workfile. See [“Estimating Scores” on page 587](#).
- **Name Factors...** may be used to provide user-specified labels for the factors. By default, the factors will be labeled “F1” and “F2” or “Factor 1” and “Factor 2”, *etc.* To provide your own names, select **Proc/Name Factors...** and enter a list of factor

names. EViews will use the specified names instead of the generic labels in table and graph output.

To clear a set of previously specified factor names, simply call up the dialog and delete the existing names.

- **Clear Rotation** removes an existing rotation from the object.

Factor Data Members

The factor object provides a number of views for examining the results of factor estimation and rotation. In addition to these views, EViews provides a number of object data members which allow you direct access to results.

For example, if you have an estimated factor object, FACT1, you may save the unique variance estimates in a vector in the workfile using the command:

```
vector unique = fact1.@unique
```

The corresponding loadings may be saved by entering:

```
matrix load = fact1.@loadings
```

The rotated loadings may be accessed by:

```
matrix rload = fact1.@rloadings
```

The fitted and residuals matrices may be obtained by entering:

```
sym fitted = fact1.@fitted
```

```
sym resid = fact1.@resid
```

For a full list of the factor object data members, see [“Factor Data Members” on page 98](#) in the *Command Reference*.

An Example

We illustrate the basic features of the factor object by analyzing a subset of the classic Holzinger and Swineford (1939) data, consisting of measures on 24 psychological tests for 145 Chicago area children attending the Grant-White school (Gorsuch, 1983). A large number of authors have used these data for illustrating various features of factor analysis. The raw data are provided in the EViews workfile “Holzinger24.wf1”. We will work with a subset consisting of seven of the 24 variables: VISUAL (visual perception), CUBES (spatial relations), PARAGRAPH (paragraph comprehension), SENTENCE (sentence completion), WORDM (word meaning), PAPER1 (paper shapes), and FLAGS1 (lozenge shapes).

(As noted by Gorsuch (1983, p. 12), the raw data and the published correlations do not match; for example, the data in “Holzinger24.wf1” produces correlations that differ from those reported in Table 7.4 of Harman (1976). Here, we will assume that the raw data are

correct; later, we will show you how to work directly with the Harman reported correlation matrix.)

Specification and Estimation

Since we have previously created a group object G7 containing the seven series of interest, double click on G7 to open the group and select **Proc/Make Factor...** EViews will open the main factor analysis specification dialog.

When created in this fashion, EViews will pre-define a specification based on the series in the group. You may click on the **Data** tab to see the prefilled settings. Here, we see that EViews has entered in the names of the seven series in G7.

The screenshot shows the 'Factor Specification' dialog box with the 'Data' tab selected. The 'Covariance specification' section has 'Type' set to 'Correlation' and 'Method' set to 'Ordinary'. The 'Variables' list contains 'visual cubes paragraph sentence wordm paper1 flags1'. The 'Sample' is set to '1 145'. The 'Balanced sample (listwise deletion)' checkbox is checked. The 'Covariance options' section has 'Series or groups for conditioning (optional)' empty, 'd.f. corrected covariances' unchecked, 'Weighting' set to 'None', and 'Weight series' empty. 'OK' and 'Cancel' buttons are at the bottom.

The remaining default settings instruct EViews to calculate an ordinary (Pearson) correlation for all of the series in the group using a balanced version of the workfile sample. You may change these as desired, but for now we will use these settings.

Next, click on the **Estimation** tab to see the main factor analysis settings. The settings may be divided into three main categories: **Method** (extraction), **Number of factors**, and **Initial communalities**. In addition, the **Options** section on the right of the dialog may be used to control miscellaneous settings.

By default, EViews will estimate a factor specification using maximum likelihood. The number of factors will be selected using Velicer's minimum average partial (MAP) method, and the

The screenshot shows the 'Factor Specification' dialog box with the 'Estimation' tab selected. The 'Method' section has 'Maximum likelihood' selected. The 'Number of factors' section has 'Minimum average partial' selected. The 'Initial communalities' section has 'Squared multiple correlation' selected. The 'Options' section on the right has 'Scale estimates to match observed variances' unchecked, 'Convergence' set to '1e-08', 'Max Iterations' set to '500', and 'Display iteration settings' unchecked. 'OK' and 'Cancel' buttons are at the bottom.

starting values for the communalities will be taken from the squared multiple correlations (SMCs). We will use the default settings for our example so you may click on **OK** to continue.

EViews estimates the model and displays the results view. Here, we see the top portion of the main results. The heading information provides basic information about the settings used in estimation, and basic status information. We see that the estimation used all 145 observations in the workfile, and converged after five iterations.

Factor Method: Maximum Likelihood
 Date: 09/11/06 Time: 12:00
 Covariance Analysis: Ordinary Correlation
 Sample: 1 145
 Included observations: 145
 Number of factors: Minimum average partial
 Prior communalities: Squared multiple correlation
 Convergence achieved after 5 iterations

| | Unrotated Loadings | | Communality | Uniqueness |
|-----------|--------------------|-----------|-------------|------------|
| | F1 | F2 | | |
| VISUAL | 0.490722 | 0.567542 | 0.562912 | 0.437088 |
| CUBES | 0.295593 | 0.342066 | 0.204384 | 0.795616 |
| PARAGRAPH | 0.855444 | -0.124213 | 0.747214 | 0.252786 |
| SENTENCE | 0.817094 | -0.154615 | 0.691548 | 0.308452 |
| WORDM | 0.810205 | -0.162990 | 0.682998 | 0.317002 |
| PAPER1 | 0.348352 | 0.425868 | 0.302713 | 0.697287 |
| FLAGS1 | 0.462895 | 0.375375 | 0.355179 | 0.644821 |

Below the heading is a section displaying the estimates of the unrotated orthogonal loadings, communalities, and uniqueness estimates obtained from estimation.

We first see that Velicer's MAP method has retained two factors, labeled "F1" and "F2". A brief examination of the unrotated loadings indicates that PARAGRAPH, SENTENCE and WORDM load on the first factor, while VISUAL, CUES, PAPER1, and FLAGS1 load on the second factor. We therefore might reasonably label the first factor as a measure of verbal ability and the second factor as an indicator of spatial ability. We will return to this interpretation shortly.

To the right of the loadings are communality and uniqueness estimates which apportion the diagonals of the correlation matrix into common (explained) and individual (unexplained) components. The communalities are obtained by computing the *row* norms of the loadings matrix, while the uniquenesses are obtained directly from the ML estimation algorithm. We see, for example, that 56% ($0.563 = 0.491^2 + 0.568^2$) of the correlation for the VISUAL variable and 69% ($0.692 = 0.817^2 + (-0.155)^2$) of the SENTENCE correlation are accounted for by the two common factors.

The next section provides summary information on the total variance and proportion of *common* variance accounted for by each of the factors, derived by taking column norms of the loadings matrix. First, we note that the cumulative variance accounted for by the two factors is 6.27, which is close to 90% ($6.267/7.0$) of the total variance. Furthermore, we see that the first factor F1 accounts for 77% ($2.722/6.267$) of the *common* variance and the second factor F2 accounts for the remaining 23% ($0.827/6.267$).

| Factor | Variance | Cumulative | Difference | Proportion | Cumulative |
|--------|----------|------------|------------|------------|------------|
| F1 | 2.719665 | 2.719665 | 1.892381 | 0.766762 | 0.766762 |
| F2 | 0.827284 | 3.546949 | --- | 0.233238 | 1.000000 |
| Total | 3.546949 | 6.266613 | | 1.000000 | |

The bottom portion of the output shows basic goodness-of-fit information for the estimated specification. The first column displays the discrepancy function, number of parameters, and degrees-of-freedom (against the saturated model) for the estimated specification. For this extraction method (ML), EViews also displays the chi-square goodness-of-fit test and Bartlett adjusted version of the test. Both versions of the test have *p*-values of over 0.75, indicating that two factors adequately explain the variation in the data.

| | Model | Independence | Saturated |
|----------------------|----------|--------------|-----------|
| Discrepancy | 0.034836 | 2.411261 | 0.000000 |
| Chi-square statistic | 5.016316 | 347.2215 | --- |
| Chi-square prob. | 0.7558 | 0.0000 | --- |
| Bartlett chi-square | 4.859556 | 339.5859 | --- |
| Bartlett probability | 0.7725 | 0.0000 | --- |
| Parameters | 20 | 7 | 28 |
| Degrees-of-freedom | 8 | 21 | --- |

For purposes of comparison, EViews also presents results for the independence (no factor) model which show that a model with no factors does not adequately model the variances.

Basic Diagnostic Views

Once we have estimated our factor specification we may examine a variety of diagnostics. First, we will examine a variety of goodness-of-fit statistics and indexes by selecting **View/Goodness-of-fit Summary** from the factor menu.

Goodness-of-fit Summary

Factor: FACTOR01

Date: 09/13/06 Time: 15:36

| | Model | Independence | Saturated |
|--------------------|----------|--------------|-----------|
| Parameters | 20 | 7 | 28 |
| Degrees-of-freedom | 8 | 21 | --- |
| Parsimony ratio | 0.380952 | 1.000000 | --- |

Absolute Fit Indices

| | Model | Independence | Saturated |
|----------------------------------|-----------|--------------|-----------|
| Discrepancy | 0.034836 | 2.411261 | 0.000000 |
| Chi-square statistic | 5.016316 | 347.2215 | --- |
| Chi-square probability | 0.7558 | 0.0000 | --- |
| Bartlett chi-square statistic | 4.859556 | 339.5859 | --- |
| Bartlett probability | 0.7725 | 0.0000 | --- |
| Root mean sq. resid. (RMSR) | 0.023188 | 0.385771 | 0.000000 |
| Akaike criterion | -0.075750 | 2.104976 | 0.000000 |
| Schwarz criterion | -0.239983 | 1.673863 | 0.000000 |
| Hannan-Quinn criterion | -0.142483 | 1.929800 | 0.000000 |
| Expected cross-validation (ECVI) | 0.312613 | 2.508483 | 0.388889 |
| Generalized fit index (GFI) | 0.989890 | 0.528286 | 1.000000 |
| Adjusted GFI | 0.964616 | -0.651000 | --- |
| Non-centrality parameter | -2.983684 | 326.2215 | --- |
| Gamma Hat | 1.000000 | 0.306239 | --- |
| McDonald Noncentrality | 1.000000 | 0.322158 | --- |
| Root MSE approximation | 0.000000 | 0.328447 | --- |

Incremental Fit Indices

| | Model |
|--------------------------------|----------|
| Bollen Relative (RFI) | 0.962077 |
| Bentler-Bonnet Normed (NFI) | 0.985553 |
| Tucker-Lewis Non-Normed (NNFI) | 1.024009 |
| Bollen Incremental (IFI) | 1.008796 |
| Bentler Comparative (CFI) | 1.000000 |

As you can see, EViews computes a large number of absolute and relative fit measures. In addition to the discrepancy, chi-square and Bartlett chi-square statistics seen previously, EViews computes scaled information criteria, expected cross-validation indices, generalized fit indices, as well as various measures based on estimates of noncentrality. Also presented are incremental fit indices which compare the fit of the estimated model against the independence model (see [“Model Evaluation,” beginning on page 615](#) for discussion).

In addition, you may examine various matrices associated with the estimation procedure. You may examine the computed correlation matrix, various reduced and fitted matrices, and a variety of residual matrices. For example, you may view the residual variance matrix by selecting **View/Residual Covariance Matrix/Using Total Covariance**.

Factor: FACTOR01 Workfile: HOLZINGR24:Untitled1

ViewProcObjectPrintNameFreezeEstimateRotateScoreStats

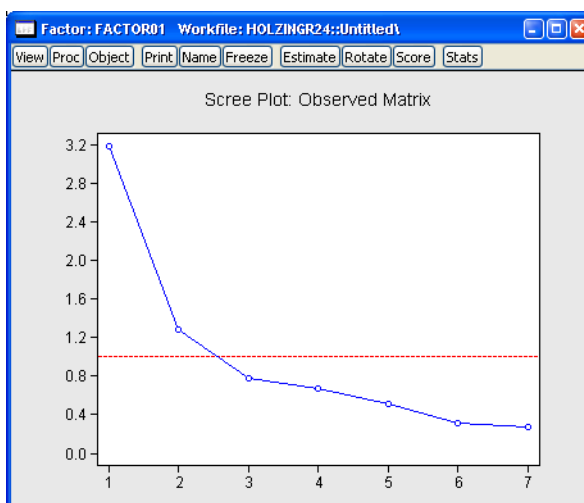
Residual (using Total Covariance)

| | VISUAL | CUBES | PARAGRAPH | SENTENCE | WORDM | PAPER1 | FLAGS1 |
|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|
| VISUAL | -8.56E-11 | -0.013392 | -0.007661 | -0.004118 | 0.012044 | -0.013708 | 0.024899 |
| CUBES | -0.013392 | 3.76E-11 | 0.017623 | -0.029159 | 0.010913 | 0.064152 | -0.038028 |
| PARAGRAPH | -0.007661 | 0.017623 | -1.14E-12 | 0.000427 | 0.001141 | 0.023294 | -0.018990 |
| SENTENCE | -0.004118 | -0.029159 | 0.000427 | 7.48E-13 | -0.001938 | 0.008363 | 0.020595 |
| WORDM | 0.012044 | 0.010913 | 0.001141 | -0.001938 | 1.05E-12 | -0.036569 | 0.002736 |
| PAPER1 | -0.013708 | 0.064152 | 0.023294 | 0.008363 | -0.036569 | 7.72E-11 | -0.022237 |
| FLAGS1 | 0.024899 | -0.038028 | -0.018990 | 0.020595 | 0.002736 | -0.022237 | -4.02E-11 |

Note that the diagonal elements of the residual matrix are zero since we have subtracted off the total fitted covariance (which includes the uniquenesses). To replace the (almost) zero diagonals with the uniqueness estimates, select instead **View/Residual Covariance Matrix/Using Common Covariance**.

You may examine eigenvalues of relevant matrices using the eigenvalue view. EViews allows you to compute eigenvalues for a variety of matrices and display the results in tabular or graphical form, but for the moment we will simply produce a scree plot for the observed correlation matrix. Select **View/Eigenvalues...** and change the **Output format** to **Graph**.

Click on **OK** to accept the settings. EViews will display the scree plot for the data, along with a line indicating the average eigenvalue.



To examine the Kaiser Measure of Sampling Adequacy, select **View/Kaiser's Measure of Sampling Adequacy**. The top portion of the display shows the individual measures and the overall of MSA (0.803) which falls in the category deemed by Kaiser to be “meritorious”.

Kaiser's Measure of Sampling Adequacy

Factor: Untitled

Date: 09/12/06 Time: 10:04

| | MSA |
|--------------|----------|
| VISUAL | 0.800894 |
| CUBES | 0.825519 |
| PARAGRAPH | 0.785366 |
| SENTENCE | 0.802312 |
| WORDM | 0.800434 |
| PAPER1 | 0.800218 |
| FLAGS1 | 0.839796 |
| Kaiser's MSA | 0.803024 |

The bottom portion of the display shows the matrix of partial correlations:

Partial Correlation:

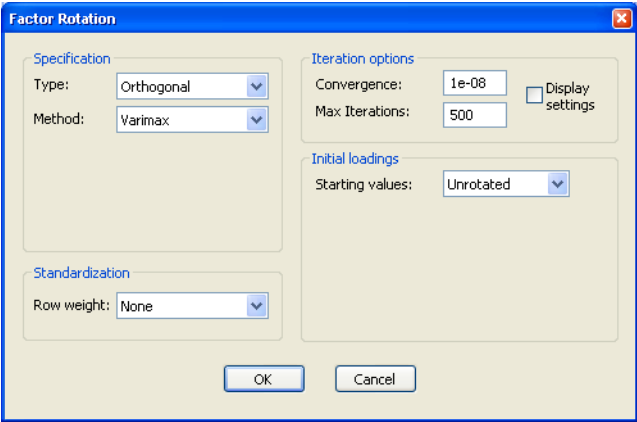
| | VISUAL | CUBES | PARAGRAPH | SENTENCE | WORDM | PAPER1 | FLAGS1 |
|-----------|----------|-----------|-----------|----------|-----------|----------|----------|
| VISUAL | 1.000000 | | | | | | |
| CUBES | 0.169706 | 1.000000 | | | | | |
| PARAGRAPH | 0.051684 | 0.070761 | 1.000000 | | | | |
| SENTENCE | 0.015776 | -0.057423 | 0.424832 | 1.000000 | | | |
| WORDM | 0.070918 | 0.044531 | 0.420902 | 0.342159 | 1.000000 | | |
| PAPER1 | 0.239682 | 0.192417 | 0.102062 | 0.042837 | -0.088688 | 1.000000 | |
| FLAGS1 | 0.321404 | 0.047793 | 0.022723 | 0.105600 | 0.050006 | 0.102442 | 1.000000 |

Each cell of this matrix contains the partial correlation for the two variables, controlling for the remaining variables.

Factor Rotation

Factor rotation may be used to simplify the factor structure and to ease the interpretation of factors. For this example, we will consider one orthogonal and one oblique rotation. To perform a factor rotation, click on the **Rotate** button on the factor toolbar or select **Proc/Rotate...** from the main factor menu.

The factor rotation dialog is used to specify the rotation method, row weighting, iteration control, and choice of initial loadings. We begin by accepting the defaults which rotate the initial loadings using orthogonal Varimax. EViews will perform the rotation and display the results.



The top portion of the displayed output provides information about the rotation and shows the rotated loadings.

| | | |
|---|----------|----------|
| Rotation Method: Orthogonal Varimax | | |
| Factor: Untitled | | |
| Date: 09/12/06 Time: 10:31 | | |
| Initial loadings: Unrotated | | |
| Convergence achieved after 4 iterations | | |
| <hr/> | | |
| Rotated loadings: $L * inv(T)'$ | | |
| | F1 | F2 |
| VISUAL | 0.255573 | 0.705404 |
| CUBES | 0.153876 | 0.425095 |
| PARAGRAPH | 0.843364 | 0.189605 |
| SENTENCE | 0.818407 | 0.147509 |
| WORDM | 0.814965 | 0.137226 |
| PAPER1 | 0.173214 | 0.522217 |
| FLAGS1 | 0.298237 | 0.515978 |

As with the unrotated loadings, the variables PARAGRAPH, SENTENCE, and WORDM load on the first factor while VISUAL, CUBES, PAPER1, and FLAGS1 load on the second factor.

The remaining sections of the output display the rotated factor correlation, initial rotation matrix, the rotation matrices applied to the factors and loadings, and objective functions for the rotations. In this case, The factor correlation and initial rotation matrices are identity matrices since we are performing an orthogonal rotation from the unrotated loadings. The remaining results are presented below:

| | | |
|--------------------------------------|-----------|----------|
| Factor rotation matrix: T | | |
| | F1 | F2 |
| F1 | 0.934003 | 0.357265 |
| F2 | -0.357265 | 0.934003 |
| Loading rotation matrix: inv(T)' | | |
| | F1 | F2 |
| F1 | 0.934003 | 0.357265 |
| F2 | -0.357265 | 0.934003 |
| Initial rotation objective: 1.226715 | | |
| Final rotation objective: 0.909893 | | |

Note that the factor rotation and loading rotation matrices are identical since we are performing an orthogonal rotation.

Perhaps more interesting are the results for an oblique rotation. To replace the Varimax results with an oblique Quartimax/Quartimin rotation, select **Proc/Rotate...** and change the **Type** combo to **Oblique**, and select **Quartimax**. We will make a few other changes in the dialog. We will use random orthogonal rotations as starting values for our rotation, so that

under **Starting values**, you should select **Random**. We also change the convergence tolerance to 1e-06. By default, EViews will perform 25 oblique rotations using random orthogonal rotation matrices as the starting values, and will select the results with the smallest objective function value. Click on **OK** to accept these settings.

The top portion of the results shows information on the rotation method and initial loadings. Just below the header are the rotated loadings. Note that the relative importance of the VISUAL, CUBES, PAPER1, and FLAGS1 loadings on the second factor is somewhat more apparent for the oblique factors.

Rotation Method: Oblique Quartimax
 Factor: Untitled
 Date: 09/12/06 Time: 10:58
 Initial loadings: Oblique Random (reps=25,
 rng=kn, seed=346876314)
 Results obtained from random draw 1 of 25
 Convergence achieved after 23 iterations

| Rotated loadings: $L * \text{inv}(T)'$ | | |
|--|-----------|-----------|
| | F1 | F2 |
| VISUAL | -0.016856 | 0.759022 |
| CUBES | -0.010310 | 0.457438 |
| PARAGRAPH | 0.846439 | 0.033230 |
| SENTENCE | 0.836783 | -0.009926 |
| WORDM | 0.837340 | -0.021054 |
| PAPER1 | -0.030042 | 0.565436 |
| FLAGS1 | 0.109927 | 0.530662 |

The rotated factor correlation is:

| Rotated factor correlation: $T'T$ | | |
|-----------------------------------|----------|----------|
| | F1 | F2 |
| F1 | 1.000000 | |
| F2 | 0.527078 | 1.000000 |

with the large off-diagonal element indicating that the orthogonality factor restriction was very much binding.

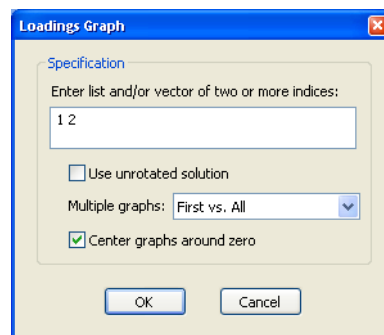
The rotation matrices and objective functions are given by:

| Factor rotation matrix: T | | |
|---|-----------|----------|
| | F1 | F2 |
| F1 | 0.984399 | 0.668380 |
| F2 | -0.175949 | 0.743820 |
| Loading rotation matrix: $\text{inv}(T)'$ | | |
| | F1 | F2 |
| F1 | 0.875271 | 0.207044 |
| F2 | -0.786498 | 1.158366 |
| Initial rotation objective: 41.94657 | | |
| Final rotation objective: 0.010096 | | |

Note that in the absence of orthogonality, the factor and loading rotation matrices differ.

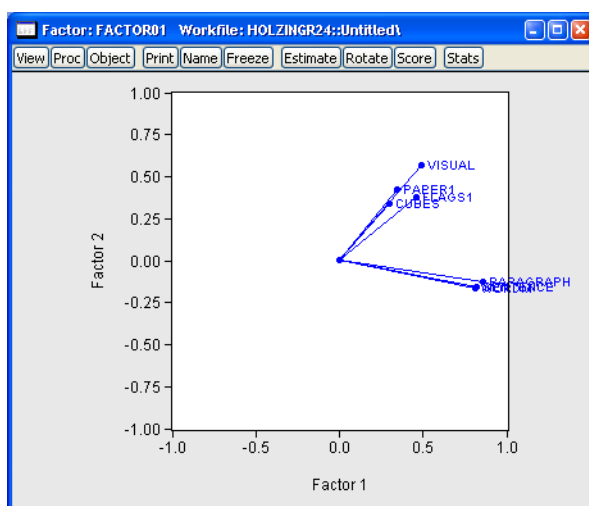
Once a rotation has been performed, the last set of rotated loadings will be available to all routines that use loadings. For example, to visualize the factor loadings, select **View/Loadings/Loadings Graph...** to bring up the loadings graph dialog.

Here you will provide indices for the factor loadings you wish to display. Since there are only two factors, EViews has prefilled the dialog with “1 2” indicating that it will plot the second factor against the first factor.



By default, EViews will use the rotated loadings if available; note the checkbox allowing you to use the unrotated loadings. Check this box and click on **OK** to display the unrotated loadings graph.

As is customary, the loadings are displayed as lines from the origin to the points labeled with the variable name. Here we see visual evidence of our previous interpretation: the variables cluster naturally into two groups (factors), with factor 1 representing verbal ability (PARAGRAPH, SENTENCE, WORDM), and factor 2 representing spatial ability (VISUAL, PAPER1, FLAGS1, CUBES).

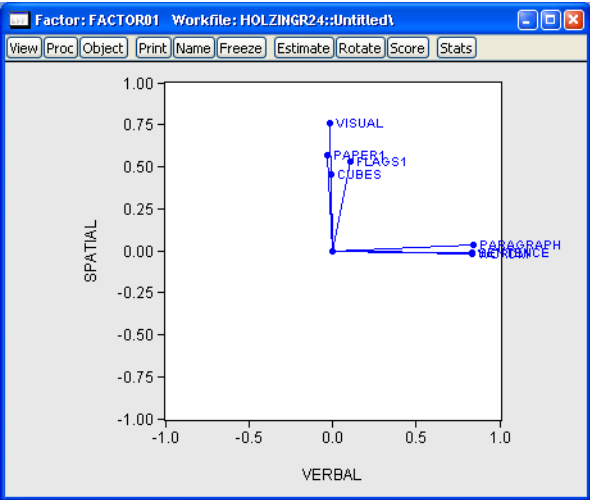


Before displaying the Quartimin rotated loadings, we will apply this labeling to the factors. Select **Proc/Name Factors...** and enter “Verbal” and “Spatial” in the dialog. EViews will subsequently label the factors using the specified names instead of the generic labels “Factor 1” and “Factor 2.”

Now, let us display the graph of the rotated loadings. Click on **View/Loadings Graph...** and simply click on **OK** to accept the defaults. EViews displays the rotated loadings graph. Note the clear separation between the sets of tests.

Factor Scores

The factors used to explain the covariance structure of the observed data are unobserved, but may be estimated from the rotated or unrotated loadings and observable data.



Click on **View/Scores...** to bring up the factor score dialog. As you can see, there are several ways to estimate the factors and several views of the results. For now, we will focus on displaying a summary of the factor score regression estimates, and in producing a biplot of the scores and loadings.

The default method of producing scores is to use exact coefficients from Thurstone's regression method, and to apply these coefficients to the observables data used in factor extraction.

Factor Scores

Display

- Table summary (selected)
- Spreadsheet
- Line graph
- Scatterplot
- Biplot graph

Table summary of scores and coefficients.

Score coefficients

Type: Exact coefficients (selected) ☐ Use unrotated solution

Coef Method: Regression (selected)

Save matrix of coeffs. (optional):

Sample

1 145

Observables

visual cubes paragraph sentence wordn paper1 flags1

Standardize using moments from: Estimation (selected)

☐ d.f. corrected variances

OK Cancel

In our example, EViews will prefill the sample and observables information; all we need to do is to select our **Display output** set-

ting, and the method for computing coefficients. Selecting **Table summary**, EViews produces output describing the score coefficient estimation.

The top portion of the output summarizes the factor score coefficient estimation settings and displays the factor coefficients used in computing scores:

| | | |
|---|----------|----------|
| Factor Score Summary | | |
| Factor: Untitled | | |
| Date: 09/12/06 Time: 11:52 | | |
| Exact scoring coefficients | | |
| Method: Regression (based on rotated loadings) | | |
| Standardize observables using moments from estimation | | |
| Sample: 1 145 | | |
| Included observations: 145 | | |
| <hr/> | | |
| Factor Coefficients: | VERBAL | SPATIAL |
| VISUAL | 0.030492 | 0.454344 |
| CUBES | 0.010073 | 0.150424 |
| PARAGRAPH | 0.391755 | 0.101888 |
| SENTENCE | 0.314600 | 0.046201 |
| WORDM | 0.305612 | 0.035791 |
| PAPER1 | 0.011325 | 0.211658 |
| FLAGS1 | 0.036384 | 0.219118 |

We see that the VERBAL score for an individual is computed as a linear combination of the centered data for VISUAL, CUBES, *etc.*, with weights given by the first column of coefficients (0.03, 0.01, *etc.*).

The next section contains the factor indeterminacy indices:

| | | | |
|------------------------|------------|-----------|---------------|
| Indeterminacy Indices: | Multiple-R | R-squared | Minimum Corr. |
| VERBAL | 0.940103 | 0.883794 | 0.767589 |
| SPATIAL | 0.859020 | 0.737916 | 0.475832 |

The indeterminacy indices show that the correlation between the estimated factors and the variables is high; the multiple correlation for the first factor well over 0.90, while the correlation for the second factor is around 0.85. The minimum correlation indices are also reasonable, suggesting that alternative factor score solutions are highly correlated. At a minimum, the correlation between two different measures of the SPATIAL factors will be nearly 0.50.

The following sections report the validity coefficients, the off-diagonal elements of the univocality matrix, and for comparison purposes, the theoretical factor correlation matrix and estimated scores correlation:

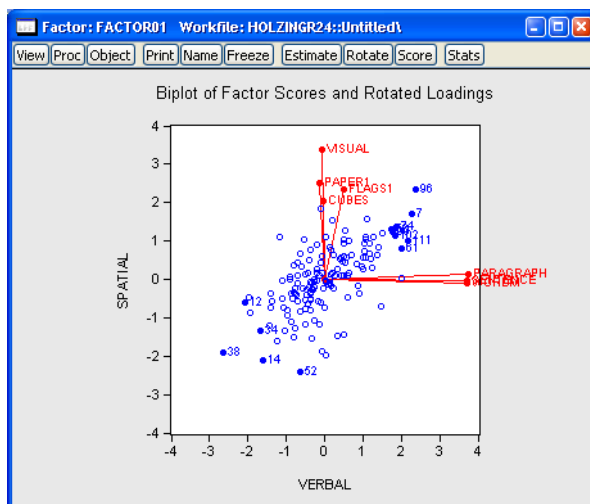
| | | |
|--|----------|----------|
| Validity Coefficients: | | |
| | VERBAL | SPATIAL |
| Validity | 0.940103 | 0.859020 |
| Univocality: (Rows=Factors; Columns=Factor scores) | | |
| | VERBAL | SPATIAL |
| VERBAL | --- | 0.590135 |
| SPATIAL | 0.539237 | --- |
| Estimated Scores Correlation: | | |
| | VERBAL | SPATIAL |
| VERBAL | 1.000000 | |
| SPATIAL | 0.627734 | 1.000000 |
| Factor Correlation: | | |
| | VERBAL | SPATIAL |
| VERBAL | 1.000000 | |
| SPATIAL | 0.527078 | 1.000000 |

The validity coefficients are both in excess of the Gorsuch (1983) recommended 0.80, and close to the stricter target of 0.90 advocated for using the estimated scores as replacements for the original variables.

The univocality matrix reports the correlations between the factors and the factor scores, which should be similar to the corresponding elements of the factor correlation matrix. Comparing results, we see that univocality correlation of 0.539 between the SPATIAL factor and the VERBAL estimated scores is close to the population correlation value of 0.527. The correlation between the VERBAL factor and the SPATIAL estimated score is somewhat higher, 0.590, but still close to the population correlation.

Similarly, the estimated scores correlation matrix should be close to the population factor correlation matrix. The off-diagonal values generally match, though as is often the case, the factor score correlation of 0.627 is a bit higher than the population value of 0.527.

To display a biplot of using these scores, select **View/Scores...** and select **Biplot graph** in the **Display** list box.



The positive correlation between the VERBAL and SPATIAL scores is obvious. The outliers show that individual 96 scores high and individual 38 low on both spatial and verbal ability, while individual 52 scores poorly on spatial relative to verbal ability.

To save scores to the workfile, select **Proc/Make Scores...** and fill out the dialog. The procedure dialog differs from the view dialog only in the **Output specification** section. Here, you should enter a list of scores to be saved or a list of indices for the scores. Since we have previously named our factors, we may specify the indices “1 2” and click on **OK**. EVIEWS will open an untitled group containing the results saved in the series VERBAL and SPATIAL.

Background

We begin with a brief sketch of the basic features of the common factor model. Our notation parallels the discussion in Johnston and Wichtern (1992).

The Model

The factor model assumes that for individual i , the *observable* multivariate p -vector X_i is generated by:

$$X_i - \mu = LF_i + \epsilon_i \quad (40.1)$$

where μ is a $p \times 1$ vector of variable means, L is a $p \times m$ matrix of coefficients, F_i is a $m \times 1$ vector of standardized *unobserved* variables, termed *common factors*, and ϵ_i is a $p \times 1$ vector of errors or *unique factors*.

The model expresses the p observable variables $X_i - \mu$ in terms of m unobservable common factors F_i , and p unobservable unique factors ϵ_i . Note that the number of unobservables exceeds the number of observables.

The *factor loading* or *pattern* matrix L links the unobserved common factors to the observed data. The j -th row of L represents the loadings of the j -th variable on the common factors. Alternately, we may view the row as the coefficients for the common factors for the j -th variable.

To proceed, we must impose additional restrictions on the model. We begin by imposing moment and covariance restrictions so that $E(F_i) = 0$ and $E(\epsilon_i) = 0$, $E(F_i\epsilon_i) = 0$, $E(F_iF_i') = \Phi$, and $E(\epsilon_i\epsilon_i) = \Psi$ where Ψ is a diagonal matrix of unique variances. Given these assumptions, we may derive the fundamental variance relationship of factor analysis by noting that the variance matrix of the observed variables is given by:

$$\begin{aligned} \text{var}(X) &= E[(X_i - \mu)(X_i - \mu)'] \\ &= E[(LF_i + \epsilon_i)(LF_i + \epsilon_i)'] \\ &= L\Phi L' + \Psi \end{aligned} \quad (40.2)$$

The variances of the individual variables may be decomposed into:

$$\sigma_{jj} = h_j^2 + \psi_j \quad (40.3)$$

for each j , where the h_j^2 are taken from the diagonal elements of $L\Phi L'$, and ψ_j is the corresponding diagonal element of Ψ . h_j^2 represents common portion of the variance of the j -th variable, termed the *communality*, while ψ_j is the unique portion of the variance, also referred to as the *uniqueness*.

Furthermore, the *factor structure* matrix containing the correlations between the variables and factors may be obtained from:

$$\begin{aligned}
\text{var}(X, F) &= E[(X_i - \mu) F_i'] \\
&= E[(L F_i + \epsilon_i) F_i'] \\
&= L\Phi
\end{aligned} \tag{40.4}$$

Initially, we make the further assumption that the factors are orthogonal so that $\Phi = I$ (we will relax this assumption shortly). Then:

$$\begin{aligned}
\text{var}(X) &= LL' + \Psi \\
\text{var}(X, F) &= L
\end{aligned} \tag{40.5}$$

Note that with orthogonal factors, the communalities h_j^2 are given by the diagonal elements of LL' (the row-norms of L).

The primary task of factor analysis is to model the $p(p+1)/2$ observed variances and covariances of the X as functions of the pm factor loadings in L , and p specific variances in Ψ . Given estimates of \hat{L} and $\hat{\Psi}$, we may form estimates of the fitted *total variance matrix*, $\hat{\Sigma} = \hat{L}\hat{L}' + \hat{\Psi}$, and the fitted *common variance matrix*, $\hat{\Sigma}_C = \hat{L}\hat{L}'$. If S is the observed dispersion matrix, we may use these estimates to define the *total variance residual matrix* $\hat{E} = S - \hat{\Sigma}$ and the common variance residual $\hat{E}_C = S - \hat{\Sigma}_C$.

Number of Factors

Choosing the number of factors is generally agreed to be one of the most important decisions one makes in factor analysis (Preacher and MacCallum, 2003; Fabrigar, *et al.*, 1999; Jackson, 1993; Zwick and Velicer, 1986). Accordingly, there is a large and varied literature describing methods for determining the number of factors, of which the references listed here are only a small subset.

Kaiser-Guttman, Minimum Eigenvalue

The Kaiser-Guttman rule, commonly termed “eigenvalues greater than 1,” is by far the most commonly used method. In this approach, one computes the eigenvalues of the unreduced dispersion matrix, and retains as many factors as the number eigenvalues that exceed the average (for a correlation matrix, the average eigenvalue is 1, hence the commonly employed description). The criterion has been sharply criticized by many on a number of grounds (*e.g.*, Preacher and MacCallum, 2003), but remains popular.

Fraction of Total Variance

The eigenvalues of the unreduced matrix may be used in a slightly different fashion. You may choose to retain as many factors as required for the sum of the first m eigenvalues to exceed some threshold fraction of the total variance. This method is used more often in principal components analysis where researchers typically include components comprising 95% of the total variance (Jackson, 1993).

Minimum Average Partial

Velicer's (1976) minimum average partial (MAP) method computes the average of the squared partial correlations after m components have been partialled out (for $m = 0, \dots, p - 1$). The number of factors retained is the number that minimizes this average. The intuition here is that the average squared partial correlation is minimized where the residual matrix is closest to being the identity matrix.

Zwick and Velicer (1986) provide evidence that the MAP method outperforms a number of other methods under a variety of conditions.

Broken Stick

We may compare the relative proportions of the total variance that are accounted for by each eigenvalue to the expected proportions obtained by chance (Jackson, 1993). More precisely, the broken stick method compares the proportion of variance given by j -th largest eigenvalue of the unreduced matrix with the corresponding expected value obtained from the broken stick distribution. The number of factors retained is the number of proportions that exceed their expected values.

Standard Error Scree

The Standard Error Scree (Zoski and Jurs, 1996) is an attempt to formalize the visual comparisons of slopes used in the visual scree test. It is based on the standard errors of sets of regression lines fit to later eigenvalues; when the standard error of the regression through the later eigenvalues falls below the specified threshold, the remaining factors are assumed to be negligible.

Parallel Analysis

Parallel analysis (Horn, 1965; Humphreys and Ilgen, 1969; Humphreys and Montanelli, 1975) involves comparing eigenvalues of the (unreduced or reduced) dispersion matrix to results obtained from simulation using uncorrelated data.

The parallel analysis simulation is conducted by generating multiple random data sets of independent random variables with the same variances and number of observations as the original data. The Pearson covariance or correlation matrix of the simulated data is computed and an eigenvalue decomposition performed for each data set. The number of factors retained is then based on the number of eigenvalues that exceed their simulated counterpart. The threshold for comparison is typically chosen to be the mean values of the simulated data as in Horn (1965), or a specific quantile as recommended by Glorfeld (1995).

Estimation Methods

There are several methods for extracting (estimating) the factor loadings and specific variances from an observed dispersion matrix.

EViews supports estimation using maximum likelihood (ML), generalized least squares (GLS), unweighted least squares (ULS), principal factors and iterated principal factors, and partitioned covariance matrix estimation (PACE).

Minimum Discrepancy (ML, GLS, ULS)

One class of extraction methods involves minimizing a discrepancy function with respect to the loadings and unique variances (Jöreskog, 1977). Let S represent the observed dispersion matrix and let the fitted matrix be $\Sigma(L, \Psi) = LL' + \Psi$. Then the discrepancy functions for ML, GLS, and ULS are given by:

$$\begin{aligned} D_{ML}(S, \Sigma) &= \text{tr}[\Sigma^{-1}S] - \ln|\Sigma^{-1}S| - p \\ D_{GLS}(S, \Sigma) &= \text{tr}([I_p - S^{-1}\Sigma]^2)/2 \\ D_{ULS}(S, \Sigma) &= \text{tr}([S - \Sigma]^2)/2 \end{aligned} \tag{40.6}$$

Each estimation method involves minimizing the appropriate discrepancy function with respect to the loadings matrix L and unique variances Ψ . An iterative algorithm for this optimization is detailed in Jöreskog. The functions all achieve an absolute minimum value of 0 when $\Sigma = S$, but in general this minimum will not be achieved.

The ML and GLS methods are scale invariant so that rescaling of the original data matrix or the dispersion matrix does not alter the basic results. The ML and GLS methods do require that the dispersion matrix be positive definite.

ULS does not require a positive definite dispersion matrix. The solution is equivalent to the iterated principal factor solution.

Principal Factors

The principal factor (principal axis) method is derived from the notion that the common factors should explain the common portion of the variance: the off-diagonal elements of the dispersion matrix and the communality portions of the diagonal elements. Accordingly, for some initial estimate of the unique variances Ψ_0 , we may define the reduced dispersion matrix $S_R(\Psi_0) = S - \Psi_0$, and then fit this matrix using common factors (see, for example, Gorsuch, 1993).

The principal factor method fits the reduced matrix using the first m eigenvalues and eigenvectors. Loading estimates, L_1 are obtained from the eigenvectors of the reduced matrix. Given the loading estimates, we may form a common variance residual matrix, $E_1 = S - L_1 L_1'$. Estimates of the uniquenesses are obtained from the diagonal elements of this residual matrix.

Communality Estimation

The construction of the reduced matrix is often described as replacing the diagonal elements of the dispersion matrix with estimates of the communalities. The estimation of these communalities has received considerable attention in the literature. Among the approaches are (Gorsuch, 1993):

- Fraction of the diagonals: use a constant fraction a of the original diagonal elements of S . One important special case is to use $a = 1$; the resulting estimates may be viewed as those from a truncated principal components solution.
- Largest correlation: select the largest absolute correlation of each variable with any other variable in the matrix.
- Squared multiple correlations (SMC): by far the most popular method; uses the squared multiple correlation between a variable and the other variables as an estimate of the communality. SMCs provide a conservative communality estimate since they are a lower bound to the communality in the population. The SMC based communalities are computed as $h_{i0}^2 = 1 - (1/r^{ii})$, where r^{ii} is the i -th diagonal element of the inverse of the observed dispersion matrix. Where the inverse cannot be computed we may employ instead the generalized inverse.

Iteration

Having obtained principal factor estimates based on initial estimates of the communalities, we may repeat the principal factors extraction using the row norms of L_1 as updated estimates of the communalities. This step may be repeated for a fixed number of iterations, or until the results are stable.

While the approach is a popular one, some authors are strongly opposed to iterating principal factors to convergence (*e.g.*, Gorsuch, 1983, p. 107–108). Performing a small number of iterations appears to be less contentious.

Partitioned Covariance (PACE)

Ihara and Kano (1986) provide a closed-form (non-iterative) estimator for the common factor model that is consistent, asymptotically normal, and scale invariant. The method requires a partitioning of the dispersion matrix into sets of variables, leading Cudeck (1991) to term this the partitioned covariance matrix estimator (PACE).

Different partitionings of the variables may lead to different estimates. Cudeck (1991) and Kano (1990) independently propose an efficient method for determining a desirable partitioning.

Since the PACE estimator is non-iterative, it is especially well suited for estimation of large factor models, or for providing initial estimates for iterative estimation methods.

Model Evaluation

One important step in factor analysis is evaluation of the fit of the estimated model. Since a factor analysis model is necessarily an approximation, we would like to examine how well a specified model fits the data, taking account the number of parameters (factors) employed and the sample size.

There are two general classes of indices for model selection and evaluation in factor analytic models. The first class, which may be termed *absolute fit indices*, are evaluated using the results of the estimated specification. Various criteria have been used for measuring absolute fit, including the familiar chi-square test of model adequacy. There is no reference specification against which the model is compared, though there may be a comparison with the observed dispersion of the saturated model.

The second class, which may be termed *relative fit indices*, compare the estimated specification against results for a reference specification, typically the zero common factor (independence model).

Before describing the various indices we first define the chi-square test statistic as a function of the discrepancy function, $T = (N - k)D(S, \Sigma)$, and note that a model with p variables and m factors has $q = p(m + 1) - m(m - 1)/2$ free parameters (pm factor loadings and m uniqueness elements, less $m(m - 1)/2$ implicit zero correlation restrictions on the factors). Since there are $p(p + 1)/2$ distinct elements of the dispersion matrix, there are a total of $df = p(p + 1)/2 - q$ remaining degrees-of-freedom.

One useful measure of the parsimony of a factor model is the parsimony ratio:

$PR = df / df_0$, where df_0 is the degrees of freedom for the independence model.

Note also that the measures described below are not reported for all estimation methods.

Absolute Fit

Most of the absolute fit measures are based on number of observations and conditioning variables, the estimated discrepancy function, D , and the number of degrees-of-freedom.

Discrepancy and Chi-Square Tests

The discrepancy functions for ML, GLS, and ULS are given by [Equation \(40.6\)](#). Principal factor and iterated principal factor discrepancies are computed using the ULS function, but will generally exceed the ULS minimum value of D .

Under the multivariate normal distributional assumptions and a correctly specified factor specification estimated by ML or GLS, the chi-square test statistic T is distributed as an asymptotic χ^2 random variable with df degrees-of-freedom (e.g., Hu and Bentler, 1995). A large value of the statistic relative to the df indicates that the model fits the data poorly (appreciably worse than the saturated model).

It is well known that the performance the T statistic is poor for small samples and non-normal settings. One popular adjustment for small sample size involves applying a Bartlett correction to the test statistic so that the multiplicative factor $N - k$ in the definition of T is replaced by $N - k - (2p + 4m + 5)/6$ (Johnston and Wichern, 1992).

Note that two distinct sets of chi-square tests that are commonly performed. The first set compares the fit of the estimated model against a saturated model; the second set of tests examines the fit of the independence model. The former are sometimes termed tests of model *adequacy* since they evaluate whether the estimated model adequately fits the data. The latter tests are sometimes referred to as test of *sphericity* since they test the assumption that there are no common factors in the data.

Information Criteria

Standard information criteria (IC) such as Akaike (AIC), Schwarz (SC), Hannan-Quinn (HQ) may be adapted for use with ML and GLS factor analysis. These indices are useful measures of fit since they reward parsimony by penalizing based on the number of parameters.

Construction of the EViews factor analysis information criteria measure employ a scaled version of the discrepancy as the log-likelihood, $l = -(N - k)/2 \cdot D$, and begins by forming the standard IC. Following Akaike (1987), we re-center the criteria by subtracting off the value for the saturated model, and following Cudeck and Browne (1983) and EViews convention, we further scale by the number of observations to eliminate the effect of sample size. The resulting factor analysis form of the information criteria are given by:

$$\begin{aligned} AIC &= (N - k)D / N - (2/N)df \\ SC &= (N - k)D / N - (\ln(N)/N)df \\ HQ &= (N - k)D / N - (2 \ln(\ln(N))/N)df \end{aligned} \tag{40.7}$$

You should be aware that these statistics are often quoted in unscaled form, sometimes without adjusting for the saturated model. Most often, if there are discrepancies, multiplying the EViews reported values by N will line up results. Note also that the current definition uses the adjusted number of observations in the numerator of the leading term.

When using information criteria for model selection, bear in mind that the model with the smallest value is considered most desirable.

Other Measures

The root mean square residual (RMSR) is given by the square root of the mean of the unique squared total covariance residuals. The standardized root mean square residual (SRMSR) is a variance standardized version of this RMSR that scales the residuals using the diagonals of the original dispersion matrix, then computes the RMSR of the scaled residuals (Hu and Bentler, 1999).

There are a number of other measures of absolute fit. We refer you to Hu and Bentler (1995, 1999) and Browne and Cudeck (1993), McDonald and Marsh (1990), Marsh, Balla and McDonald (1988) for details on these measures and recommendations on their use. Note that where there are small differences in the various descriptions of the measures due to degree-of-freedom corrections, we have used the formulae provided by Hu and Bentler (1999).

Incremental Fit

Incremental fit indices measure the improvement in fit of the model over a more restricted specification. Typically, the restricted specification is chosen to be the zero factor or independence model.

EVIEWS reports up to five relative fit measures: the generalized Tucker-Lewis Nonnormed Fit Index (NNFI), Bentler and Bonnet's Normed Fit Index (NFI), Bollen's Relative Fit Index (RFI), Bollen's Incremental Fit Index (IFI), and Bentler's Comparative Fit Index (CFI). See Hu and Bentler (1995) for details.

Traditionally, the rule of thumb was for acceptable models to have fit indices that exceed 0.90, but recent evidence suggests that this cutoff criterion may be inadequate. Hu and Bentler (1999) provide some guidelines for evaluating values of the indices; for ML estimation, they recommend use of two indices, with cutoff values close to 0.95 for the NNFI, RFI, IFI, CFI.

Rotation

The estimated loadings and factors are not unique; we may obtain others that fit the observed covariance structure identically. This observation lies behind the notion of *factor rotation*, in which we apply transformation matrices to the original factors and loadings in the hope of obtaining a simpler factor structure.

To elaborate, we begin with the orthogonal factor model from above:

$$X_i - \mu = L F_i + \epsilon_i \quad (40.8)$$

where $E(F_i F_i') = I_m$. Suppose that we pre-multiply our factors by a $m \times m$ rotation matrix T' where $T' T = \Phi$. Then we may re-write the factor model [Equation \(40.1\)](#) as:

$$X_i - \mu = L (T^{-1})' T' F_i + \epsilon_i = \tilde{L} \tilde{F}_i + \epsilon_i \quad (40.9)$$

which is an observationally equivalent common factor model with rotated loadings $\tilde{L} = L (T^{-1})'$ and factors $\tilde{F}_i = T' F_i$, where the correlation of the rotated factors is given by:

$$E(\tilde{F}_i \tilde{F}_i') = T' T = \Phi \quad (40.10)$$

See Browne (2001) and Bernaards and Jennrich (2005) for details.

Types of Rotation

There are two basic types of rotation that involve different restrictions on Φ . In *orthogonal rotation*, we impose $m(m-1)/2$ constraints on the transformation matrix T so that $\Phi = I$, implying that the rotated factors are orthogonal. In *oblique rotation*, we impose only m constraints on T , requiring the diagonal elements of Φ equal 1.

There are a large number of rotation methods. The majority of methods involving minimizing an objective function that measure the complexity of the rotated factor matrix with respect to the choice of T , subject to any constraints on the factor correlation. Jennrich (2001, 2002) describes algorithms for performing orthogonal and oblique rotations by minimizing complexity objective.

For example, suppose we form the $p \times m$ matrix Λ where every element λ_{ij} equals the square of a corresponding factor loading l_{ij} : $\lambda_{ij} = l_{ij}^2$. Intuitively, one or more measures of simplicity of the rotated factor pattern can be expressed as a function of these squared loadings. One such function defines the Crawford-Ferguson family of complexities:

$$f(L) = (1 - \kappa) \sum_{i=1}^p \left(\sum_{j=1}^m \sum_{k \neq j}^m \lambda_{ij} \lambda_{ik} \right) + \kappa \sum_{j=1}^m \left(\sum_{i=1}^p \sum_{p \neq i}^p \lambda_{ij} \lambda_{pj} \right) \quad (40.11)$$

for weighting parameter κ . The Crawford-Ferguson (CF) family is notable since it encompasses a large number of popular rotation methods (including Varimax, Quartimax, Equamax, Parsimax, and Factor Parsimony).

The first summation term in parentheses, which is based on the outer-product of the i -th row of the squared loadings, provides a measure of complexity. Those rows which have few non-zero elements will have low complexity compared to rows with many non-zero elements. Thus, the first term in the function is a measure of the row (variables) complexity of the loadings matrix. Similarly, the second summation term in parentheses is a measure of the complexity of the j -th column of the squared loadings matrix. The second term provides a measure of the column (factor) complexity of the loadings matrix. It follows that higher values for κ assign greater weight to factor complexity and less weight to variable complexity.

Along with the CF family, EViews supports the following rotation methods:

| Method | Orthogonal | Oblique |
|-------------------|------------|---------|
| Biquartimax | • | • |
| Crawford-Ferguson | • | • |
| Entropy | • | |
| Entropy Ratio | • | |
| Equamax | • | • |

| | | |
|-------------------------------|---|---|
| Factor Parsimony | • | • |
| Generalized Crawford-Ferguson | • | • |
| Geomin | • | • |
| Harris-Kaiser (case II) | | • |
| Infomax | • | • |
| Oblimax | | • |
| Oblimin | | • |
| Orthomax | • | • |
| Parsimax | • | • |
| Pattern Simplicity | • | • |
| Promax | | • |
| Quartimax/Quartimin | • | • |
| Simplimax | • | • |
| Tandem I | • | |
| Tandem II | • | |
| Target | • | • |
| Varimax | • | • |

EVIEWS employs the Crawford-Ferguson variants of the Biquartimax, Equamax, Factor Parsimony, Orthomax, Parsimax, Quartimax, and Varimax objective functions. For example, The EVIEWS Orthomax objective for parameter γ is evaluated using the Crawford-Ferguson objective with factor complexity weight $\kappa = \gamma/p$.

These forms of the objective functions yield the same results as the standard versions in the orthogonal case, but are better behaved (*e.g.*, do not permit factor collapse) under direct oblique rotation (see Browne 2001, p. 118-119). Note that oblique Crawford-Ferguson Quartimax is equivalent to Quartimin.

The two orthoblique methods, the Promax and Harris-Kaiser both perform an initial orthogonal rotation, followed by a oblique adjustment. For both of these methods, EVIEWS provides some flexibility in the choice of initial rotation. By default, EVIEWS will perform an initial Orthomax rotation with the default parameter set to 1 (Varimax). To perform initial rotation with Quartimax, you should set the Orthomax parameter to 0. See Gorsuch (1993) and Harris-Kaiser (1964) for details.

Some rotation methods require specification of one or more parameters. A brief description and the default value(s) used by EVIEWS is provided below:

| Method | n | Parameter Description |
|-------------------------------|-----|---|
| Crawford-Ferguson | 1 | Factor complexity weight (<i>default</i> = 0, Quartimax). |
| Generalized Crawford-Ferguson | 4 | Vector of weights for (in order): total squares, variable complexity, factor complexity, diagonal quartics (<i>no default</i>). |
| Geomin | 1 | Epsilon offset (<i>default</i> = 0.01). |
| Harris-Kaiser (case II) | 2 | Power parameter (<i>default</i> = 0, independent cluster solution). |
| Oblimin | 1 | Deviation from orthogonality (<i>default</i> = 0, Quartimin). |
| Orthomax | 1 | Factor complexity weight (<i>default</i> = 1, Varimax). |
| Promax | 1 | Power parameter (<i>default</i> = 3). |
| Simplimax | 1 | Fraction of near-zero loadings (<i>default</i> = 0.75). |
| Target | 1 | $p \times m$ matrix of target loadings. Missing values correspond to unrestricted elements. (<i>No default</i> .) |

Standardization

Weighting the rows of the initial loading matrix prior to rotation can sometimes improve the rotated solution (Browne, 2001). Kaiser standardization weights the rows by the inverse square roots of the communalities. Cureton-Mulaik standardization assigns weights between zero and one to the rows of the loading matrix using a more complicated function of the original matrix.

Both standardization methods may lead to instability in cases with small communalities.

Starting Values

Starting values for the rotation objective minimization procedures are typically taken to be the identity matrix (the unrotated loadings). The presence of local minima is a distinct possibility and it may be prudent to consider random rotations as alternate starting values. Random orthogonal rotations may be used as starting values for orthogonal rotation; random orthogonal or oblique rotations may be used to initialize the oblique rotation objective minimization.

Scoring

The factors used to explain the covariance structure of the observed data are unobserved, but may be estimated from the loadings and observable data. These *factor score estimates* may be used in subsequent diagnostic analysis, or as substitutes for the higher-dimensional observed data.

Score Estimation

We may compute factor score estimates \hat{G}_i as a linear combination of observed data:

$$\hat{G}_i = \hat{W}'(Z_i - \mu_Z) \quad (40.12)$$

where \hat{W} is a $p \times m$ matrix of factor score coefficients derived from the estimates of the factor model. Often, we will construct estimates using the original data so that $Z_i = X_i$ but this is not required; we may for example use coefficients obtained from one set of data to score individuals in a second set of data.

Various methods for estimating the score coefficients W have been proposed. The first class of factor scoring methods computes *exact* or *refined* estimates of the coefficient weights W . Generally speaking, these methods optimize some property of the estimated scores with respect to the choice of W . For example, Thurstone's regression approach maximizes the correlation of the scores with the true factors (Gorsuch, 1983). Other methods minimize a function of the estimated errors \hat{e} with respect to W , subject to constraints on the estimated factor scores. For example, Anderson and Rubin (1956) and McDonald (1981) compute weighted least squares estimators of the factor scores, subject to the condition that the implied correlation structure of the scores $\hat{W}'\Sigma\hat{W}$, equals Φ .

The second set of methods computes *coarse* coefficient weights in which the elements of W are restricted to be (-1, 0, 1) values. These simplified weights are determined by recoding elements of the factor loadings matrix or an exact coefficient weight matrix on the basis of their magnitudes. Values of the matrices that are greater than some threshold (in absolute value) are assigned sign-corresponding values of -1 or 1; all other values are recoded at 0 (Grice, 2001).

Score Evaluation

There are an infinite number of factor score estimates that are consistent with an estimated factor model. This lack of identification, termed *factor indeterminacy*, has received considerable attention in the literature (see for example, Mulaik (1996); Steiger (1979)), and is a primary reason for the multiplicity of estimation methods, and for the development of procedures for evaluating the quality of a given set of scores (Gorsuch, 1983, p. 272).

See Gorsuch (1993) and Grice(2001) for additional discussion of the following measures.

Indeterminacy Indices

There are two distinct types of indeterminacy indices. The first set measures the multiple correlation between each factor and the observed variables, ρ and its square ρ^2 . The squared multiple correlations are obtained from the diagonals of the matrix $P = \Sigma^{-1}\Gamma$ where Σ is the observed dispersion matrix and $\Gamma = L\Phi$ is the factor structure matrix. Both of these indices range from 0 to 1, with high values being desirable.

The second type of indeterminacy index reports the minimum correlation between alternate estimates of the factor scores, $\rho^* = 2\rho^2 - 1$. The minimum correlation measure ranges from -1 to 1. High positive values are desirable since they indicate that differing sets of factor scores will yield similar results.

Grice (2001) suggests that values for ρ that do not exceed 0.707 by a significant degree are problematic since values below this threshold imply that we may generate two sets of factor scores that are orthogonal or negatively correlated (Green, 1976).

Validity, Univocality, Correlational Accuracy

Following Gorsuch (1983), we may define R_{ff} as the population factor correlation matrix, R_{ss} as the factor score correlation matrix, and R_{fs} as the correlation matrix of the known factors with the score estimates. In general, we would like these matrices to be similar.

The diagonal elements of R_{fs} are termed *validity* coefficients. These coefficients range from -1 to 1, with high positive values being desired. Differences between the validities and the multiple correlations are evidence that the computed factor scores have determinacies lower than those computed using the ρ -values. Gorsuch (1983) recommends obtaining validity values of at least 0.80, and notes that values larger than 0.90 may be necessary if we wish to use the score estimates as substitutes for the factors.

The off-diagonal elements of R_{fs} allow us to measure *univocality*, or the degree to which the estimated factor scores have correlations with those of other factors. Off-diagonal values of R_{fs} that differ from those in R_{ff} are evidence of univocality bias.

Lastly, we obviously would like the estimated factor scores to match the correlations among the factors themselves. We may assess the *correlational accuracy* of the scores estimates by comparing the values of the R_{ss} with the values of R_{ff} .

From our earlier discussion, we know that the population correlation $R_{ff} = \hat{W}'\Sigma\hat{W}$. R_{ss} may be obtained from moments of the estimated scores. Computation of R_{fs} is more complicated, but follows the steps outlined in Gorsuch (1983).

References

- Akaike, H. (1987). "Factor Analysis and AIC," *Psychometrika*, 52(3), 317–332.
- Anderson, T. W. and H. Rubin (1956). "Statistical Inference in Factor Analysis," in Neyman, J., editor, *Proceedings of the Third Berkeley Symposium on Mathematical Statistics and Probability, Volume V*, 111-150. Berkeley and Los Angeles: University of California Press.
- Bernaards, C. A., and R. I. Jennrich (2005). "Gradient Projection Algorithms and Software for Arbitrary Rotation Criteria in Factor Analysis", *Educational and Psychological Measurement*, 65(5), 676-696.
- Browne, M. W. (2001). "An Overview of Analytic Rotation in Exploratory Factor Analysis," *Multivariate Behavioral Research*, 36(1), 111–150.
- Browne, M. W. and R. Cudeck (1993). "Alternative ways of Assessing Model Fit," in K. A. Bollen and J. S. Long (eds.), *Testing Structural Equation Models*, Newbury Park, CA: Sage.

- Cudeck, R. and M. W. Browne (1983). "Cross-validation of Covariance Structures," *Multivariate Behavioral Research*, 18, 147–167.
- Dziuban, C. D. and E. C. Shirkey (1974). "When is a Correlation Matrix Appropriate for Factor Analysis," *Psychological Bulletin*, 81(6), 358–361.
- Fabrigar, L. R., D. T. Wegener, R. C. MacCallum, and E. J. Strahan (1999). "Evaluating the Use of Exploratory Factor Analysis in Psychological Research," *Psychological Methods*, 4(3), 272–299.
- Glorfeld, L. W. (1995). "An Improvement on Horn's Parallel Analysis Methodology for Selecting the Correct Number of Factors to Retain," *Educational and Psychological Measurement*, 55(3), 377–393.
- Gorsuch, R. L. (1983). *Factor Analysis*, Hillsdale, New Jersey: Lawrence Erlbaum Associates, Inc.
- Green, B. F., Jr. (1969). "Best Linear Composites with a Specified Structure," *Psychometrika*, 34(3), 301–318.
- Green, B. F., Jr. (1976). "On the Factor Score Controversy," *Psychometrika*, 41(2), 263–266.
- Grice, J. W. (2001). "Computing and Evaluating Factor Scores," *Psychological Methods*, 6(4), 430–450.
- Harman, H. H. (1976). *Modern Factor Analysis, Third Edition Revised*, Chicago: University of Chicago Press.
- Harris, C. W. and H. F. Kaiser (1964). "Oblique Factor Analytic Solutions by Orthogonal Transformations," *Psychometrika*, 29(4), 347–362.
- Hendrickson, A. and P. White (1964). "Promax: A Quick Method for Rotation to Oblique Simple Structure," *The British Journal of Statistical Psychology*, 17(1), 65–70.
- Horn, J. L. (1965). "A Rationale and Test for the Number of Factors in Factor Analysis," *Psychometrika*, 30(2), 179–185.
- Hu, L.-T. and P. M. Bentler (1995). "Evaluating Model Fit," in R. H. Hoyle (Ed.), *Structural Equation Modeling: Concepts, Issues, and Applications*, Thousand Oaks, CA: Sage.
- Hu, L.-T. and P. M. Bentler (1999). "Cut-off Criteria for Fit Indexes in Covariance Structure Analysis: Conventional Criteria Versus New Alternatives," *Structural Equation Modeling*, 6(1), 1–55.
- Humphreys, L. G. and D. R. Ilgen (1969). "Note on a Criterion for the Number of Common Factors," *Educational and Psychological Measurement*, 29, 571–578.
- Humphreys, L. G. and R. G. Montanelli, Jr. (1975). "An Investigation of the Parallel Analysis Criterion for Determining the Number of Common Factors," *Multivariate Behavioral Research*, 10, 193–206.
- Ihara, M. and Y. Kano (1995). "A New Estimator of the Uniqueness in Factor Analysis," *Psychometrika*, 51(4), 563–566.
- Jackson, D. A. (1993). "Stopping Rules in Principal Components Analysis: A Comparison of Heuristical and Statistical Approaches," *Ecology*, 74(8), 2204–2214.
- Jennrich, R. I. (2001). "A Simple General Procedure for Orthogonal Rotation," *Psychometrika*, 66(2), 289–306.
- Jennrich, R. I. (2002). "A Simple General Method for Oblique Rotation," *Psychometrika*, 67(1), 7–20.
- Johnson, R. A., and D. W. Wichern (1992). *Applied Multivariate Statistical Analysis, Third Edition*, Upper Saddle River, New Jersey: Prentice-Hall, Inc.
- Jöreskog, K. G. (1977). "Factor Analysis by Least-Squares and Maximum Likelihood Methods," in *Statistical Methods for Digital Computers*, K. Enslein, A. Ralston, and H. S. Wilf, (eds.), New York: John Wiley & Sons, Inc.
- Kaiser, H. F. (1970). "A Second Generation Little Jiffy," *Psychometrika*, 35(4), 401–415.

- Kaiser, H. F. and J. Rice (1974). "Little Jiffy, Mark IV," *Educational and Psychological Measurement*, 34, 111–117.
- Kano, Y. (1990). "Noniterative estimation and the choice of the number of factors in exploratory factor analysis," *Psychometrika*, 55(2), 277–291.
- Marsh, H. W., J. R. Balla and R. P. McDonald (1988). "Goodness of Fit Indexes in Confirmatory Factor Analysis: The Effect of Sample Size," *Psychological Bulletin*, 103(3), 391–410.
- McDonald, R. P. (1981). "Constrained Least Squares Estimators of Oblique Common Factors," *Psychometrika*, 46(2), 277–291.
- McDonald, R. P. and H. W. Marsh (1990). "Choosing a Multivariate Model: Noncentrality and Goodness of Fit," *Psychological Bulletin*, 107(2), 247–255.
- Preacher, K. J. and R. C. MacCallum (2003). "Repairing Tom Swift's Electric Factor Analysis Machine," *Understanding Statistics*, 2(1), 13–32.
- Ten Berge, J. M. F., W. P. Krijnen, T. Wansbeek, and A. Shapiro (1999). "Some New Results on Correlation Preserving Factor Scores Prediction Methods," *Linear Algebra and Its Applications*, 289, 311–318.
- Tucker, L. R. and R. C. MacCallum (1997). *Exploratory Factor Analysis*, Unpublished manuscript.
- Velicer, W. F. (1976). "Determining the Number of Components from the Matrix of Partial Correlations," *Psychometrika*, 41(3), 321–327.
- Zoski, K. W. and S. Jurs (1996). "An Objective Counterpart to the Visual Scree Test for Factor Analysis: The Standard Error Scree," *Educational and Psychological Measurement*, 56(3), 443–451.
- Zwick, W. R. and W. F. Velicer (1986). "Factors Influencing Five Rules for Determining the Number of Components to Retain," *Psychological Bulletin*, 99(3), 432–442.

Appendix D. Estimation and Solution Options

EViews estimates the parameters of a wide variety of nonlinear models, from nonlinear least squares equations, to maximum likelihood models, to GMM specifications. These types of nonlinear estimation problems do not have closed form solutions and must be estimated using iterative methods. EViews also solves systems of non-linear equations. Again, there are no closed form solutions to these problems, and EViews must use an iterative method to obtain a solution.

Below, we provide details on the algorithms used by EViews in dealing with nonlinear estimation and solution, and the optional settings that we provide to allow you to control estimation.

Our discussion here is necessarily brief. For additional details, we direct you to the quite readable discussions in Press, *et al.* (1992), Quandt (1983), Thisted (1988), and Amemiya (1983).

Setting Estimation Options

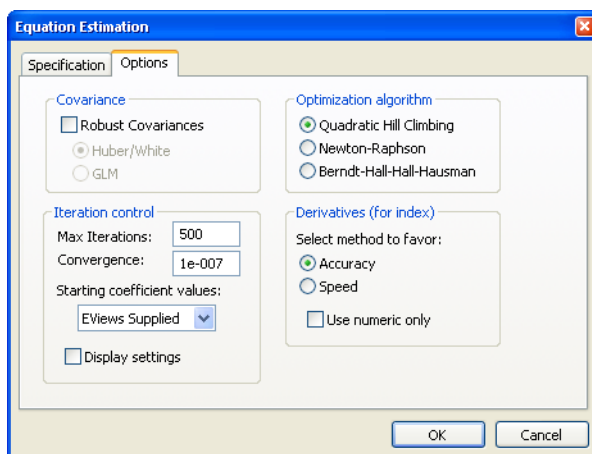
When you estimate an equation in EViews, you enter specification information into the **Specification** tab of the **Equation Estimation** dialog. Clicking on the **Options** tab displays a dialog that allows you to set various options to control the estimation procedure. The contents of the dialog will differ depending upon the options available for a particular estimation procedure.

The default settings for the options will be taken from the global options ([“Estimation Defaults” on page 767](#)), or from the options used previously to estimate the object.

The **Options** tab for binary models is depicted here. For other estimator and estimation techniques (*e.g.* systems) the dialog will differ to reflect the different estimation options that are available.

Starting Coefficient Values

Iterative estimation procedures require starting values for the coefficients of the model. There are no general rules for select-



ing starting values for parameters. Obviously, the closer to the true values, the better, so if you have reasonable guesses for parameter values, these can be useful. In some cases, you can obtain starting values by estimating a restricted version of the model. In general, however, you may have to experiment to find good starting values.

EViews follows three basic rules for selecting starting values:

- For nonlinear least squares type problems, EViews uses the values in the coefficient vector at the time you begin the estimation procedure as starting values.
- For system estimators and ARCH, EViews uses starting values based upon preliminary single equation OLS or TSLS estimation. In the dialogs for these estimators, the drop-down menu for setting starting values will not appear.
- For selected estimation techniques (binary, ordered, count, censored and truncated), EViews has built-in algorithms for determining the starting values using specific information about the objective function. These will be labeled in the **Starting coefficient values** combo box as **EViews supplied**.

In the latter two cases, you may change this default behavior by selecting an item from the **Starting coefficient values** drop down menu. You may choose fractions of the default starting values, zero, or arbitrary **User Supplied**.

If you select **User Supplied**, EViews will use the values stored in the C coefficient vector at the time of estimation as starting values. To see the starting values, double click on the coefficient vector in the workfile directory. If the values appear to be reasonable, you can close the window and proceed with estimating your model.

If you wish to change the starting values, first make certain that the spreadsheet view of the coefficient vector is in edit mode, then enter the coefficient values. When you are finished setting the initial values, close the coefficient vector window and estimate your model.

You may also set starting coefficient values from the command window using the PARAM command. Simply enter the PARAM keyword, followed by pairs of coefficients and their desired values:

```
param c(1) 153 c(2) .68 c(3) .15
```

sets $C(1) = 153$, $C(2) = .68$, and $C(3) = .15$. All of the other elements of the coefficient vector are left unchanged.

Lastly, if you want to use estimated coefficients from another equation, select **Proc/Update Coefs from Equation** from the equation window toolbar.

For nonlinear least squares problems or situations where you specify the starting values, bear in mind that:

- The objective function must be defined at the starting values. For example, if your objective function contains the expression $1/C(1)$, then you cannot set $C(1)$ to zero. Similarly, if the objective function contains $\text{LOG}(C(2))$, then $C(2)$ must be greater than zero.
- A poor choice of starting values may cause the nonlinear least squares algorithm to fail. EViews begins nonlinear estimation by taking derivatives of the objective function with respect to the parameters, evaluated at these values. If these derivatives are not well behaved, the algorithm may be unable to proceed.

If, for example, the starting values are such that the derivatives are all zero, you will immediately see an error message indicating that EViews has encountered a “Near Singular Matrix”, and the estimation procedure will stop.

- Unless the objective function is globally concave, iterative algorithms may stop at a local optimum. There will generally be no evidence of this fact in any of the output from estimation.

If you are concerned with the possibility of local optima, you may wish to select various starting values and see whether the estimates converge to the same values. One common suggestion is to estimate the model and then randomly alter each of the estimated coefficients by some percentage, then use these new coefficients as starting values in estimation.

Iteration and Convergence Options

There are two common iteration stopping rules: based on the change in the objective function, or based on the change in parameters. The convergence rule used in EViews is based upon changes in the parameter values. This rule is generally conservative, since the change in the objective function may be quite small as we approach the optimum (this is how we choose the direction), while the parameters may still be changing.

The exact rule in EViews is based on comparing the norm of the change in the parameters with the norm of the current parameter values. More specifically, the convergence test is:

$$\frac{\|\theta_{(i+1)} - \theta_{(i)}\|_2}{\|\theta_{(i)}\|_2} \leq tol \quad (40.13)$$

where θ is the vector of parameters, $\|x\|_2$ is the 2-norm of x , and tol is the specified tolerance. However, before taking the norms, each parameter is scaled based on the largest observed norm across iterations of the derivative of the least squares residuals with respect to that parameter. This automatic scaling system makes the convergence criteria more robust to changes in the scale of the data, but does mean that restarting the optimization from the final converged values may cause additional iterations to take place, due to slight changes in the automatic scaling value when started from the new parameter values.

The estimation process achieves convergence if the stopping rule is reached using the tolerance specified in the **Convergence** edit box of the Estimation Dialog or the Estimation Options Dialog. By default, the box will be filled with the tolerance value specified in the global estimation options, or if the estimation object has previously been estimated, it will be filled with the convergence value specified for the last set of estimates.

EViews may stop iterating even when convergence is not achieved. This can happen for two reasons. First, the number of iterations may have reached the prespecified upper bound. In this case, you should reset the maximum number of iterations to a larger number and try iterating until convergence is achieved.

Second, EViews may issue an error message indicating a “Failure to improve” after a number of iterations. This means that even though the parameters continue to change, EViews could not find a direction or step size that improves the objective function. This can happen when the objective function is ill-behaved; you should make certain that your model is identified. You might also try other starting values to see if you can approach the optimum from other directions.

Lastly, EViews may converge, but warn you that there is a singularity and that the coefficients are not unique. In this case, EViews will not report standard errors or *t*-statistics for the coefficient estimates.

Derivative Computation Options

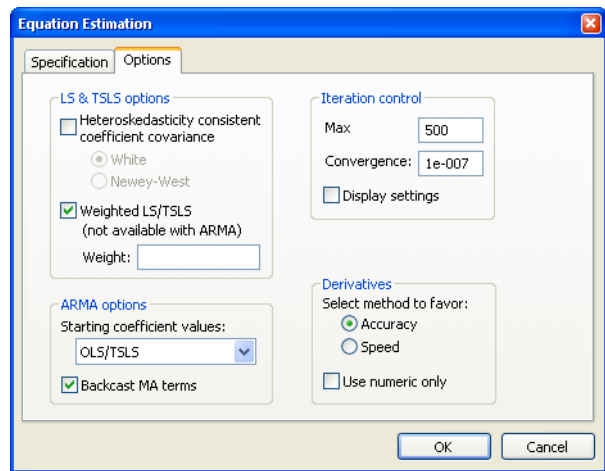
In many EViews estimation procedures, you can specify the form of the function for the mean equation. For example, when estimating a regression model, you may specify an arbitrary nonlinear expression in the coefficients. In these cases, when estimating the model, EViews will compute derivatives of the user-specified function.

EViews uses two techniques for evaluating derivatives: numeric (finite difference) and analytic. The approach that is used depends upon the nature of the optimization problem and any user-defined settings:

- In most cases, EViews offers the user the choice of computing either analytic or numeric derivatives. By default, EViews will fill the options dialog with the global estimation settings. If the **Use numeric only** setting is chosen, EViews will only compute the derivatives using finite difference methods. If this setting is not checked, EViews will attempt to compute analytic derivatives, and will use numeric derivatives only where necessary.
- EViews will ignore the numeric derivative setting and use an analytic derivative whenever a coefficient derivative is a constant value.
- For some procedures where the range of specifications allowed is limited (*e.g.*, VARs, pools), EViews always uses analytic first and/or second derivatives, whatever the values of these settings.

- The derivatives with respect to the AR coefficients in an ARMA specification are always computed analytically while those with respect to the MA coefficients are computed numerically.
- In a limited number of cases, EViews will always use numeric derivatives. For example, selected GARCH (see [“Derivative Methods” on page 192](#)) and state space models always use numeric derivatives. As noted above, MA coefficient derivatives are always computed numerically.
- Logl objects always use numeric derivatives unless you provide the analytic derivatives in the specification.

Where relevant, the estimation options dialog allows you to control the method of taking derivatives. For example, the options dialog for standard regression allows you to override the use of EViews analytic derivatives, and to choose between favoring speed or accuracy in the computation of any numeric derivatives (note that the additional LS and TSLS options are discussed in detail in [Chapter 25. “Additional Regression Methods,” beginning on page 23](#)).



Computing the more accurate numeric derivatives requires additional objective function evaluations. While the algorithms may change in future versions, at present, EViews computes numeric derivatives using either a one-sided finite difference (favor speed), or using a four-point routine using Richardson extrapolation (favor precision). Additional details are provided in Kincaid and Cheney (1996).

Analytic derivatives will often be faster and more accurate than numeric derivatives, especially if the analytic derivatives have been simplified and carefully optimized to remove common subexpressions. Numeric derivatives will sometimes involve fewer floating point operations than analytic, and in these circumstances, may be faster.

Optimization Algorithms

Given the importance of the proper setting of EViews estimation options, it may prove useful to review briefly various basic optimization algorithms used in nonlinear estimation. Recall

that the problem faced in non-linear estimation is to find the values of parameters θ that optimize (maximize or minimize) an objective function $F(\theta)$.

Iterative optimization algorithms work by taking an initial set of values for the parameters, say $\theta_{(0)}$, then performing calculations based on these values to obtain a better set of parameter values, $\theta_{(1)}$. This process is repeated for $\theta_{(2)}$, $\theta_{(3)}$ and so on until the objective function F no longer improves between iterations.

There are three main parts to the optimization process: (1) obtaining the initial parameter values, (2) updating the candidate parameter vector θ at each iteration, and (3) determining when we have reached the optimum.

If the objective function is globally concave so that there is a single maximum, any algorithm which improves the parameter vector at each iteration will eventually find this maximum (assuming that the size of the steps taken does not become negligible). If the objective function is not globally concave, different algorithms may find different local maxima, but all iterative algorithms will suffer from the same problem of being unable to tell apart a local and a global maximum.

The main thing that distinguishes different algorithms is how quickly they find the maximum. Unfortunately, there are no hard and fast rules. For some problems, one method may be faster, for other problems it may not. EViews provides different algorithms, and will often let you choose which method you would like to use.

The following sections outline these methods. The algorithms used in EViews may be broadly classified into three types: *second derivative* methods, *first derivative* methods, and *derivative free* methods. EViews' second derivative methods evaluate current parameter values and the first and second derivatives of the objective function for every observation. First derivative methods use only the first derivatives of the objective function during the iteration process. As the name suggests, derivative free methods do not compute derivatives.

Second Derivative Methods

For binary, ordered, censored, and count models, EViews can estimate the model using Newton-Raphson or *quadratic hill-climbing*.

Newton-Raphson

Candidate values for the parameters $\theta_{(1)}$ may be obtained using the method of Newton-Raphson by linearizing the first order conditions $\partial F / \partial \theta$ at the current parameter values, $\theta_{(i)}$:

$$\begin{aligned} g_{(i)} + H_{(i)}(\theta_{(i+1)} - \theta_{(i)}) &= 0 \\ \theta_{(i+1)} &= \theta_{(i)} - H_{(i)}^{-1} g_{(i)} \end{aligned} \tag{40.14}$$

where g is the gradient vector $\partial F / \partial \theta$, and H is the Hessian matrix $\partial^2 F / \partial \theta^2$.

If the function is quadratic, Newton-Raphson will find the maximum in a single iteration. If the function is not quadratic, the success of the algorithm will depend on how well a local quadratic approximation captures the shape of the function.

Quadratic hill-climbing (Goldfeld-Quandt)

This method, which is a straightforward variation on Newton-Raphson, is sometimes attributed to Goldfeld and Quandt. Quadratic hill-climbing modifies the Newton-Raphson algorithm by adding a correction matrix (or ridge factor) to the Hessian. The quadratic hill-climbing updating algorithm is given by:

$$\theta_{(i+1)} = \theta_{(i)} - \tilde{H}_{(i)}^{-1} g_{(i)} \quad \text{where } -\tilde{H}_{(i)} = -H_{(i)} + \alpha I \quad (40.15)$$

where I is the identity matrix and α is a positive number that is chosen by the algorithm.

The effect of this modification is to push the parameter estimates in the direction of the gradient vector. The idea is that when we are far from the maximum, the local quadratic approximation to the function may be a poor guide to its overall shape, so we may be better off simply following the gradient. The correction may provide better performance at locations far from the optimum, and allows for computation of the direction vector in cases where the Hessian is near singular.

For models which may be estimated using second derivative methods, EViews uses quadratic hill-climbing as its default method. You may elect to use traditional Newton-Raphson, or the first derivative methods described below, by selecting the desired algorithm in the Options menu.

Note that asymptotic standard errors are always computed from the unmodified Hessian once convergence is achieved.

First Derivative Methods

Second derivative methods may be computationally costly since we need to evaluate the $k(k+1)/2$ elements of the second derivative matrix at every iteration. Moreover, second derivatives calculated may be difficult to compute accurately. An alternative is to employ methods which require only the first derivatives of the objective function at the parameter values.

For general nonlinear models (nonlinear least squares, ARCH and GARCH, nonlinear system estimators, GMM, State Space), EViews provides two first derivative methods: Gauss-Newton/BHHH or Marquardt.

Gauss-Newton/BHHH

This algorithm follows Newton-Raphson, but replaces the negative of the Hessian by an approximation formed from the sum of the outer product of the gradient vectors for each observation's contribution to the objective function. For least squares and log likelihood

functions, this approximation is asymptotically equivalent to the actual Hessian when evaluated at the parameter values which maximize the function. When evaluated away from the maximum, this approximation may be quite poor.

The algorithm is referred to as *Gauss-Newton* for general nonlinear least squares problems, and often attributed to Berndt, Hall, Hall and Hausman (*BHHH*) for maximum likelihood problems.

The advantages of approximating the negative Hessian by the outer product of the gradient are that (1) we need to evaluate only the first derivatives, and (2) the outer product is necessarily positive semi-definite. The disadvantage is that, away from the maximum, this approximation may provide a poor guide to the overall shape of the function, so that more iterations may be needed for convergence.

Marquardt

The Marquardt algorithm modifies the Gauss-Newton algorithm in exactly the same manner as quadratic hill climbing modifies the Newton-Raphson method (by adding a correction matrix (or ridge factor) to the Hessian approximation).

The ridge correction handles numerical problems when the outer product is near singular and may improve the convergence rate. As above, the algorithm pushes the updated parameter values in the direction of the gradient.

For models which may be estimated using first derivative methods, EViews uses Marquardt as its default method. You may elect to use traditional Gauss-Newton via the Options menu.

Note that asymptotic standard errors are always computed from the unmodified (Gauss-Newton) Hessian approximation once convergence is achieved.

Choosing the step size

At each iteration, we can search along the given direction for the optimal step size. EViews performs a simple trial-and-error search at each iteration to determine a step size λ that improves the objective function. This procedure is sometimes referred to as *squeezing* or *stretching*.

Note that while EViews will make a crude attempt to find a good step, λ is not actually optimized at each iteration since the computation of the direction vector is often more important than the choice of the step size. It is possible, however, that EViews will be unable to find a step size that improves the objective function. In this case, EViews will issue an error message.

EViews also performs a crude trial-and-error search to determine the scale factor α for Marquardt and quadratic hill-climbing methods.

Derivative free methods

Other optimization routines do not require the computation of derivatives. The *grid search* is a leading example. Grid search simply computes the objective function on a grid of parameter values and chooses the parameters with the highest values. Grid search is computationally costly, especially for multi-parameter models.

EViews uses (a version of) grid search for the exponential smoothing routine.

Nonlinear Equation Solution Methods

When solving a nonlinear equation system, EViews first analyzes the system to determine if the system can be separated into two or more blocks of equations which can be solved sequentially rather than simultaneously. Technically, this is done by using a graph representation of the equation system where each variable is a vertex and each equation provides a set of edges. A well known algorithm from graph theory is then used to find the strongly connected components of the directed graph.

Once the blocks have been determined, each block is solved for in turn. If the block contains no simultaneity, each equation in the block is simply evaluated once to obtain values for each of the variables.

If a block contains simultaneity, the equations in that block are solved by either a Gauss-Seidel or Newton method, depending on how the solver options have been set.

Gauss-Seidel

By default, EViews uses the Gauss-Seidel method when solving systems of nonlinear equations. Suppose the system of equations is given by:

$$\begin{aligned}x_1 &= f_1(x_1, x_2, \dots, x_N, z) \\x_2 &= f_2(x_1, x_2, \dots, x_N, z) \\&\vdots \\x_N &= f_N(x_1, x_2, \dots, x_N, z)\end{aligned}\tag{40.16}$$

where x are the endogenous variables and z are the exogenous variables.

The problem is to find a fixed point such that $x = f(x, z)$. Gauss-Seidel employs an iterative updating rule of the form:

$$x^{(i+1)} = f(x^{(i)}, z).\tag{40.17}$$

to find the solution. At each iteration, EViews solves the equations in the order that they appear in the model. If an endogenous variable that has already been solved for in that iteration appears later in some other equation, EViews uses the value as solved in that iteration. For example, the k -th variable in the i -th iteration is solved by:

$$x_k^{(i)} = f_k(x_1^{(i)}, x_2^{(i)}, \dots, x_{k-1}^{(i)}, x_k^{(i-1)}, x_{k+1}^{(i-1)}, \dots, x_N^{(i-1)}, z). \quad (40.18)$$

The performance of the Gauss-Seidel method can be affected by reordering of the equations. If the Gauss-Seidel method converges slowly or fails to converge, you should try moving the equations with relatively few and unimportant right-hand side endogenous variables so that they appear early in the model.

Newton's Method

Newton's method for solving a system of nonlinear equations consists of repeatedly solving a local linear approximation to the system.

Consider the system of equations written in implicit form:

$$F(x, z) = 0 \quad (40.19)$$

where F is the set of equations, x is the vector of endogenous variables and z is the vector of exogenous variables.

In Newton's method, we take a linear approximation to the system around some values x^* and z^* :

$$F(x, z) = F(x^*, z^*) + \frac{\partial}{\partial x} F(x^*, z^*) \Delta x = 0 \quad (40.20)$$

and then use this approximation to construct an iterative procedure for updating our current guess for x :

$$x_{t+1} = x_t - \left[\frac{\partial}{\partial x} F(x_t, z^*) \right]^{-1} F(x_t, z^*) \quad (40.21)$$

where raising to the power of -1 denotes matrix inversion.

The procedure is repeated until the changes in x between periods are smaller than a specified tolerance.

Note that in contrast to Gauss-Seidel, the ordering of equations under Newton does not affect the rate of convergence of the algorithm.

Broyden's Method

Broyden's Method is a modification of Newton's method which tries to decrease the calculational cost of each iteration by using an approximation to the derivatives of the equation system rather than the true derivatives of the equation system when calculating the Newton step. That is, at each iteration, Broyden's method takes a step:

$$x_{t+1} = x_t - J_t^{-1} F(x_t, z^*) \quad (40.22)$$

where J_t is the current approximation to the matrix of derivatives of the equation system.

As well as updating the value of x at each iteration, Broyden's method also updates the existing Jacobian approximation, J_t , at each iteration based on the difference between the observed change in the residuals of the equation system and the change in the residuals predicted by a linear approximation to the equation system based on the current Jacobian approximation.

In particular, Broyden's method uses the following equation to update J :

$$J_{t+1} = J_t + \frac{(F(x_{t+1}, z^*) - F(x_t, z^*) - J_t \Delta x) \Delta x'}{\Delta x' \Delta x} \quad (40.23)$$

where $\Delta x = x_{t+1} - x_t$. This update has a number of desirable properties (see Chapter 8 of Dennis and Schnabel (1983) for details).

In EViews, the Jacobian approximation is initialized by taking the true derivatives of the equation system at the starting values of x . The updating procedure given above is repeated until changes in x between periods become smaller than a specified tolerance. In some cases the method may stall before reaching a solution, in which case a fresh set of derivatives of the equation system is taken at the current values of x , and the updating is continued using these derivatives as the new Jacobian approximation.

Broyden's method shares many of the properties of Newton's method including the fact that it is not dependent on the ordering of equations in the system and that it will generally converge quickly in the vicinity of a solution. In comparison to Newton's method, Broyden's method will typically take less time to perform each iteration, but may take more iterations to converge to a solution. In most cases Broyden's method will take less overall time to solve a system than Newton's method, but the relative performance will depend on the structure of the derivatives of the equation system.

References

- Amemiya, Takeshi (1983). "Nonlinear Regression Models," Chapter 6 in Z. Griliches and M. D. Intriligator (eds.), *Handbook of Econometrics, Volume 1*, Amsterdam: Elsevier Science Publishers B.V.
- Dennis, J. E. and R. B. Schnabel (1983). "Secant Methods for Systems of Nonlinear Equations," *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. Prentice-Hall, London.
- Kincaid, David, and Ward Cheney (1996). *Numerical Analysis, 2nd edition*, Pacific Grove, CA: Brooks/Cole Publishing Company.
- Press, W. H., S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery (1992). *Numerical Recipes in C, 2nd edition*, Cambridge University Press.
- Quandt, Richard E. (1983). "Computational Problems and Methods," Chapter 12 in Z. Griliches and M. D. Intriligator (eds.), *Handbook of Econometrics, Volume 1*, Amsterdam: Elsevier Science Publishers B.V.
- Thisted, Ronald A. (1988). *Elements of Statistical Computing*, New York: Chapman and Hall.

Appendix E. Gradients and Derivatives

Many EViews estimation objects provide built-in routines for examining the gradients and derivatives of your specifications. You can, for example, use these tools to examine the analytic derivatives of your nonlinear regression specification in numeric or graphical form, or you can save the gradients from your estimation routine for specification tests.

The gradient and derivative views may be accessed from most estimation objects by selecting **View/Gradients and Derivatives** or, in some cases, **View/Gradients**, and then selecting the appropriate view.

If you wish to save the numeric values of your gradients and derivatives, you will need to use the gradient and derivative procedures. These procs may be accessed from the main **Proc** menu.

Note that all views and procs are not available for every estimation object or every estimation technique.

Gradients

EViews provides you with the ability to examine and work with the gradients of the objective function for a variety of estimation objects. Examining these gradients can provide useful information for evaluating the behavior of your nonlinear estimation routine, or can be used as the basis of various tests of specification.

Since EViews provides a variety of estimation methods and techniques, the notion of a gradient is a bit difficult to describe in casual terms. EViews will generally report the values of the first-order conditions used in estimation. To take the simplest example, ordinary least squares minimizes the sum-of-squared residuals:

$$S(\beta) = \sum_t (y_t - X_t' \beta)^2 \quad (40.24)$$

The first-order conditions for this objective function are obtained by differentiating with respect to β , yielding

$$\sum_t -2(y_t - X_t' \beta) X_t \quad (40.25)$$

EViews allows you to examine both the sum and the corresponding average, as well as the value for each of the individual observations. Furthermore, you can save the individual values in series for subsequent analysis.

The individual gradient computations are summarized in the following table:

| | |
|----------------------------------|--|
| Least squares | $g_t = -2(y_t - f_t(X_t, \beta)) \left(\frac{\partial f_t(X_t, \beta)}{\partial \beta} \right)$ |
| Weighted least squares | $g_t = -2(y_t - f_t(X_t, \beta)) w_t^2 \left(\frac{\partial f_t(X_t, \beta)}{\partial \beta} \right)$ |
| Two-stage least squares | $g_t = -2(y_t - f_t(X_t, \beta)) P_t \left(\frac{\partial f_t(X_t, \beta)}{\partial \beta} \right)$ |
| Weighted two-stage least squares | $g_t = -2(y_t - f_t(X_t, \beta)) w_t \tilde{P}_t w_t \left(\frac{\partial f_t(X_t, \beta)}{\partial \beta} \right)$ |
| Maximum likelihood | $g_t = \frac{\partial l_t(X_t, \beta)}{\partial \beta}$ |

where P and \tilde{P} are the projection matrices corresponding to the expressions for the estimators in [Chapter 25. “Additional Regression Methods,” beginning on page 23](#), and where l is the log likelihood contribution function.

Note that the expressions for the regression gradients are adjusted accordingly in the presence of ARMA error terms.

Gradient Summary

To view the summary of the gradients, select **View/Gradients and Derivatives/Gradient Summary**, or **View/Gradients/Summary**. EViews will display a summary table showing the sum, mean, and Newton direction associated with the gradients. Here is an example table from a nonlinear least squares estimation equation:

Gradients of the Objective Function
Gradients evaluated at estimated parameters
Equation: EQ01
Method: Least Squares
Specification: LOG(CS) = C(1) +C(2)*(GDP^C(3)-1)/C(3)
Computed using analytic derivatives

| Coefficient | Sum | Mean | Newton Dir. |
|-------------|----------|----------|-------------|
| C(1) | 5.21E-10 | 2.71E-12 | 1.41E-14 |
| C(2) | 9.53E-09 | 4.96E-11 | -3.11E-18 |
| C(3) | 3.81E-08 | 1.98E-10 | 2.47E-18 |

There are several things to note about this table. The first line of the table indicates that the gradients have been computed at estimated parameters. If you ask for a gradient view for an

estimation object that has not been successfully estimated, EViews will compute the gradients at the current parameter values and will note this in the table. This behavior allows you to diagnose unsuccessful estimation problems using the gradient values.

Second, you will note that EViews informs you that the gradients were computed using analytic derivatives. EViews will also inform you if the specification is linear, if the derivatives were computed numerically, or if EViews used a mixture of analytic and numeric techniques. We remind you that all MA coefficient derivatives are computed numerically.

Lastly, there is a table showing the sum and mean of the gradients as well as a column labeled “Newton Dir.”. The column reports the non-Marquardt adjusted Newton direction used in first-derivative iterative estimation procedures (see [“First Derivative Methods” on page 631](#)).

In the example above, all of the values are “close” to zero. While one might expect these values always to be close to zero when evaluated at the estimated parameters, there are a number of reasons why this will not always be the case. First, note that the sum and mean values are highly scale variant so that changes in the scale of the dependent and independent variables may lead to marked changes in these values. Second, you should bear in mind that while the Newton direction is *related* to the terms used in the optimization procedures, EViews’ test for convergence does not directly use the Newton direction. Third, some of the iteration options for system estimation do not iterate coefficients or weights fully to convergence. Lastly, you should note that the values of these gradients are sensitive to the accuracy of any numeric differentiation.

Gradient Table and Graph

There are a number of situations in which you may wish to examine the individual contributions to the gradient vector. For example, one common source of singularity in nonlinear estimation is the presence of very small combined with very large gradients at a given set of coefficient values.

EViews allows you to examine your gradients in two ways: as a spreadsheet of values, or as line graphs, with each set of coefficient gradients plotted in a separate graph. Using these tools, you can examine your data for observations with outlier values for the gradients.

Gradient Series

You can save the individual gradient values in series using the **Make Gradient Group** procedure. EViews will create a new group containing series with names of the form GRAD## where ## is the next available name.

Note that when you store the gradients, EViews will fill the series for the full workfile range. If you view the series, make sure to set the workfile sample to the sample used in estimation if you want to reproduce the table displayed in the gradient views.

Application to LM Tests

The gradient series are perhaps most useful for carrying out Lagrange multiplier tests for nonlinear models by running what is known as artificial regressions (Davidson and MacKinnon 1993, Chapter 6). A generic artificial regression for hypothesis testing takes the form of regressing:

$$\tilde{u}_t \text{ on } \left(\frac{\partial f_t(X_t, \tilde{\beta})}{\partial \beta} \right) \text{ and } Z_t \quad (40.26)$$

where \tilde{u} are the estimated residuals under the restricted (null) model, and $\tilde{\beta}$ are the estimated coefficients. The Z are a set of “misspecification indicators” which correspond to departures from the null hypothesis.

An example program (“GALLANT2.PRG”) for performing an LM auxiliary regression test is provided in your EViews installation directory.

Gradient Availability

The gradient views are currently available for the equation, logl, sspace and system objects. The views are not, however, currently available for equations estimated by GMM or ARMA equations specified by expression.

Derivatives

EViews employs a variety of rules for computing the derivatives used by iterative estimation procedures. These rules, and the user-defined settings that control derivative taking, are described in detail in [“Derivative Computation Options” on page 628](#).

In addition, EViews provides both object views and object procedures which allow you to examine the effects of those choices, and the results of derivative taking. These views and procedures provide you with quick and easy access to derivatives of your user-specified functions.

It is worth noting that these views and procedures are not available for all estimation techniques. For example, the derivative views are currently not available for binary models since only a limited set of specifications are allowed.

Derivative Description

The **Derivative Description** view provides a quick summary of the derivatives used in estimation.

For example, consider the simple nonlinear regression model:

$$y_t = c(1)(1 - \exp(-c(2)x_t)) + \epsilon_t \quad (40.27)$$

Following estimation of this single equation, we can display the description view by selecting **View/Gradients and Derivatives.../Derivative Description**.

Derivatives of the Equation Specification
 Equation: EQ02
 Method: Least Squares
 Specification: RESID = Y - ((C(1)*(1-EXP(-C(2)*X))))
 Computed using analytic derivatives

| Variable | Derivative of Specification |
|----------|-----------------------------|
| C(1) | -1 + exp(-c(2) * x) |
| C(2) | -c(1) * x * exp(-c(2) * x) |

There are three parts to the output from this view. First, the line labeled “Specification:” describes the equation specification that we are estimating. You will note that we have written the specification in terms of the implied residual from our specification.

The next line describes the method used to compute the derivatives used in estimation. Here, EViews reports that the derivatives were computed analytically.

Lastly, the bottom portion of the table displays the expressions for the derivatives of the regression function with respect to each coefficient. Note that the derivatives are in terms of the implied residual so that the signs of the expressions have been adjusted accordingly.

In this example, all of the derivatives were computed analytically. In some cases, however, EViews will not know how to take analytic derivatives of your expression with respect to one or more of the coefficient. In this situation, EViews will use analytic expressions where possible, and numeric where necessary, and will report which type of derivative was used for each coefficient.

Suppose, for example, that we estimate:

$$y_t = c(1)(1 - \exp(-\phi(c(2)x_t))) + \epsilon_t \quad (40.28)$$

where ϕ is the standard normal density function. The derivative view of this equation is

Derivatives of the Equation Specification
Equation: EQ02
Method: Least Squares
Specification: $RESID = Y - ((C(1)*(1-EXP(-@DNORM(C(2)*X))))$
Computed using analytic derivatives
Use accurate numeric derivatives where necessary

| Variable | Derivative of Specification |
|----------|--------------------------------|
| C(1) | $-1 + \exp(-@dnorm(c(2) * x))$ |
| C(2) | --- accurate numeric --- |

Here, EViews reports that it attempted to use analytic derivatives, but that it was forced to use a numeric derivative for C(2) (since it has not yet been taught the derivative of the @dnorm function).

If we set the estimation option so that we only compute fast numeric derivatives, the view would change to

Derivatives of the Equation Specification
Equation: EQ02
Method: Least Squares
Specification: $RESID = Y - ((C(1)*(1-EXP(-C(2)*X))))$
Computed using fast numeric derivatives

| Variable | Derivative of Specification |
|----------|-----------------------------|
| C(1) | --- fast numeric --- |
| C(2) | --- fast numeric --- |

to reflect the different method of taking derivatives.

If your specification contains autoregressive terms, EViews will only compute the derivatives with respect to the regression part of the equation. The presence of the AR components is, however, noted in the description view.

Derivatives of the Equation Specification

Equation: EQ02

Method: Least Squares

Specification: $[AR(1)=C(3)] = Y - (C(1)-EXP(-C(2)*X))$

Computed using analytic derivatives

| Variable | Derivative of Specification* |
|----------|------------------------------|
| C(1) | -1 |
| C(2) | $-x * \exp(-c(2) * x)$ |

*Note: derivative expressions do not account for ARMA components

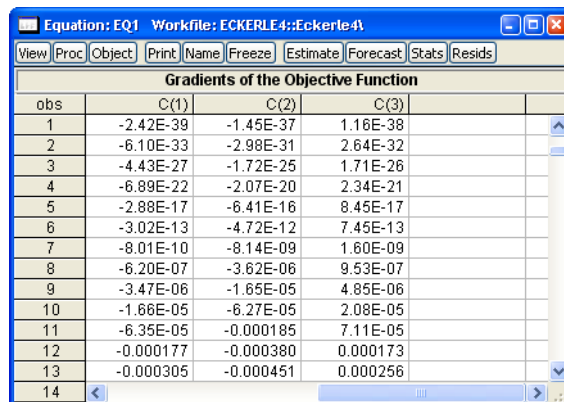
Recall that the derivatives of the objective function with respect to the AR components are always computed analytically using the derivatives of the regression specification, and the lags of these values.

One word of caution about derivative expressions. For many equation specifications, analytic derivative expressions will be quite long. In some cases, the analytic derivatives will be longer than the space allotted to them in the table output. You will be able to identify these cases by the trailing “...” in the expression.

To see the entire expression, you will have to create a table object and then resize the appropriate column. Simply click on the **Freeze** button on the toolbar to create a table object, and then highlight the column of interest. Click on **Width** on the table toolbar and enter in a larger number.

Derivative Table and Graph

Once we obtain estimates of the parameters of our nonlinear regression model, we can examine the values of the derivatives at the estimated parameter values. Simply select **View/Gradients and Derivatives...** to see a spreadsheet view or line graph of the values of the derivatives for each coefficient:



The screenshot shows a window titled 'Equation: EQ1 Workfile: ECKERLE4::Eckerte4'. Below the title bar is a menu bar with 'View', 'Proc', 'Object', 'Print', 'Name', 'Freeze', 'Estimate', 'Forecast', 'Stats', and 'Resids'. The main area displays a table titled 'Gradients of the Objective Function'. The table has columns for 'obs', 'C(1)', 'C(2)', and 'C(3)'. The data rows are numbered 1 through 14, with the last row (14) highlighted. The values are in scientific notation.

| obs | C(1) | C(2) | C(3) |
|-----|-----------|-----------|----------|
| 1 | -2.42E-39 | -1.45E-37 | 1.16E-38 |
| 2 | -6.10E-33 | -2.98E-31 | 2.64E-32 |
| 3 | -4.43E-27 | -1.72E-25 | 1.71E-26 |
| 4 | -6.89E-22 | -2.07E-20 | 2.34E-21 |
| 5 | -2.88E-17 | -6.41E-16 | 8.45E-17 |
| 6 | -3.02E-13 | -4.72E-12 | 7.45E-13 |
| 7 | -8.01E-10 | -8.14E-09 | 1.60E-09 |
| 8 | -6.20E-07 | -3.62E-06 | 9.53E-07 |
| 9 | -3.47E-06 | -1.65E-05 | 4.85E-06 |
| 10 | -1.66E-05 | -6.27E-05 | 2.08E-05 |
| 11 | -6.35E-05 | -0.000185 | 7.11E-05 |
| 12 | -0.000177 | -0.000380 | 0.000173 |
| 13 | -0.000305 | -0.000451 | 0.000256 |
| 14 | | | |

This spreadsheet view displays the value of the derivatives for each observation in the standard spreadsheet form. The graph view, plots the value of each of these derivatives for each coefficient.

Derivative Series

You can save the derivative values in series for later use. Simply select **Proc/Make Derivative Group** and EViews will create an untitled group object containing the new series. The series will be named DERIV##, where ## is a number associated with the next available free name. For example, if you have the objects DERIV01 and DERIV02, but not DERIV03 in the workfile, EViews will save the next derivative in the series DERIV03.

References

Davidson, Russell and James G. MacKinnon (1993). *Estimation and Inference in Econometrics*, Oxford: Oxford University Press.

Appendix F. Information Criteria

As part of the output for most regression procedures, EViews reports various information criteria. The information criteria are often used as a guide in model selection (see for example, Grasa 1989).

The Kullback-Leibler quantity of information contained in a model is the distance from the “true” model and is measured by the log likelihood function. The notion of an information criterion is to provide a measure of information that strikes a balance between this measure of goodness of fit and parsimonious specification of the model. The various information criteria differ in how to strike this balance.

Definitions

The basic information criteria are given by:

| | |
|-----------------------------|-------------------------------|
| Akaike info criterion (AIC) | $-2(l/T) + 2(k/T)$ |
| Schwarz criterion (SC) | $-2(l/T) + k\log(T)/T$ |
| Hannan-Quinn criterion (HQ) | $-2(l/T) + 2k\log(\log(T))/T$ |

Let l be the value of the log of the likelihood function with the k parameters estimated using T observations. The various information criteria are all based on -2 times the average log likelihood function, adjusted by a penalty function.

For factor analysis models, EViews follows convention (Akaike, 1987), re-centering the criteria by subtracting off the value for the saturated model. The resulting factor analysis form of the information criteria are given by:

| | |
|-----------------------------|------------------------------------|
| Akaike info criterion (AIC) | $(T - k)D/T - (2/T)df$ |
| Schwarz criterion (SC) | $(T - k)D/T - (\log(T)/T)df$ |
| Hannan-Quinn criterion (HQ) | $(T - k)D/T - (2\ln(\log(T))/T)df$ |

where D is the discrepancy function, and df is the number of degrees-of-freedom in the estimated dispersion matrix. Note that EViews scales the Akaike form of the statistic by dividing by T .

In addition to the information criteria described above, there are specialized information criteria that are used in by EViews when computing unit root tests:

| | |
|---------------------|-----------------------------|
| Modified AIC (MAIC) | $-2(l/T) + 2((k + \tau)/T)$ |
|---------------------|-----------------------------|

| | |
|-----------------------------|--|
| Modified SIC (MSIC) | $-2(l/T) + (k + \tau)\log(T)/T$ |
| Modified Hannan-Quinn (MHQ) | $-2(l/T) + 2(k + \tau)\log(\log(T))/T$ |

where the modification factor τ is computed as:

$$\tau = \alpha^2 \sum_t \tilde{y}_{t-1}^2 / \sigma^2 \quad (40.29)$$

for $\tilde{y}_t \equiv y_t$ when computing the ADF test equation (26.45), and for \tilde{y}_t as defined in (“Autoregressive Spectral Density Estimator,” beginning on page 98) when estimating the frequency zero spectrum (see Ng and Perron, 2001, for a discussion of the modified information criteria).

Note also that:

- The definitions used by EViews may differ slightly from those used by some authors. For example, Grasa (1989, equation 3.21) does not divide the AIC by T . Other authors omit inessential constants of the Gaussian log likelihood (generally, the terms involving 2π).

While very early versions of EViews reported information criteria that omitted inessential constant terms, the current version of EViews always uses the value of the full likelihood function. All of your equation objects estimated in earlier versions of EViews will automatically be updated to reflect this change. You should, however, keep this fact in mind when comparing results from frozen table objects or printed output from previous versions.

- For systems of equations, where applicable, the information criteria are computed using the full system log likelihood. The log likelihood value is computed assuming a multivariate normal (Gaussian) distribution as:

$$l = -\frac{TM}{2}(1 + \log 2\pi) - \frac{T}{2} \log |\hat{\Omega}| \quad (40.30)$$

where

$$|\hat{\Omega}| = \det \left(\sum_t \hat{\epsilon} \hat{\epsilon}' / T \right) \quad (40.31)$$

M is the number of equations. Note that these expressions are only strictly valid when you there are equal numbers of observations for each equation. When your system is unbalanced, EViews replaces these expressions with the appropriate summations.

- The factor analysis forms of the statistics are often quoted in unscaled form, sometimes without adjusting for the saturated model. Most often, if there are discrepancies, multiplying the EViews reported values by T will line up results.

Using Information Criteria as a Guide to Model Selection

As a user of these information criteria as a model selection guide, you select the model with the smallest information criterion.

The information criterion has been widely used in time series analysis to determine the appropriate length of the distributed lag. Lütkepohl (1991, Chapter 4) presents a number of results regarding consistent lag order selection in VAR models.

You should note, however, that the criteria depend on the unit of measurement of the dependent variable y . For example, you cannot use information criteria to select between a model with dependent variable y and one with $\log(y)$.

References

- Grasa, Antonio Aznar (1989). *Econometric Model Selection: A New Approach*, Dordrecht: Kluwer Academic Publishers.
- Akaike, H. (1987). "Factor Analysis and AIC," *Psychometrika*, 52(3), 317–332.
- Lütkepohl, Helmut (1991). *Introduction to Multiple Time Series Analysis*, New York: Springer-Verlag.
- Ng, Serena and Pierre Perron (2001). "Lag Length Selection and the Construction of Unit Root Tests with Good Size and Power," *Econometrica*, 69(6), 1519-1554.

Index

(Key: *I* = User's Guide I; *II* = User's Guide II)

Symbols

- (dash)
 - negation [I:639](#)
 - subtraction [I:639](#)
- ! (exclamation) control variable [I:598](#)
- % (percent sign)
 - program arguments [I:607](#), [I:608](#)
 - string variable [I:599](#)
- * (asterisk) multiplication [I:639](#)
- + (plus)
 - addition [I:639](#)
- .DB? files [I:262](#)
- .EDB file [I:258](#)
- .RTF file [I:553](#)
- .WF1 file [I:51](#)
- / (slash) division [I:640](#)
- ?
 - pool cross section identifier [II:464](#)
 - wildcard versus pool identifier [I:778](#)
- _ (continuation character) [I:593](#)
- { [I:601](#)
- ~, in backup file name [I:52](#), [I:765](#)

Numerics

- 2sls (two-stage least squares) [II:37](#)
 - with pooled data [II:502](#)
- 3sls (three-stage least squares) [II:309](#), [II:337](#)

A

- Abort key [I:9](#)
- @abs [I:735](#)
- abs [I:735](#)
- Absolute value [I:735](#)
- @acos [I:754](#)
- Across factors [I:511](#)
- Active window [I:6](#), [I:69](#)
- Add factor [II:407](#)
- Add text to graph [I:526](#)
- Adding data [I:230](#)

- Addition operator (+) [I:639](#)
- Addtext [I:668](#)
- ADF
 - See also* Unit root tests.
- Adjusted R-squared
 - for regression [II:13](#)
- Advanced database query [I:275](#)
- AIC [II:645](#)
 - See also* Akaike criterion.
- Akaike criterion [II:15](#), [II:645](#)
 - for equation [II:15](#)
- Alias [II:409](#)
 - database [I:272](#)
 - OBALIAS.INI file [I:282](#)
 - object [I:281](#)
- @all [I:88](#)
- Almon lag [II:24](#)
- Alpha series [I:150](#)
 - additional views [I:155](#)
 - declaring and creating [I:150](#)
 - maximum length [I:151](#), [I:769](#)
 - spreadsheet view [I:155](#)
 - truncation [I:151](#), [I:769](#)
- Alphabetizing [I:697](#)
- Analysis of variance [I:315](#)
 - by ranks [I:318](#)
- Analytical derivatives [II:640](#)
 - logl [II:288](#)
- Analytical graphs [I:462](#)
- and [I:734](#)
- And operator [I:89](#), [I:124](#)
- Anderson-Darling test [I:321](#)
- Andrews test [II:219](#), [II:257](#)
- ANOVA [I:315](#)
 - by ranks [I:318](#)
- Appending data [I:230](#)
- AR Roots (VAR) [II:348](#)
- AR specification
 - estimation [II:67](#)
 - forecast [II:128](#)
 - in 2SLS [II:40](#)
 - in ARIMA models [II:71](#)

- in nonlinear 2SLS [II:50](#)
- in nonlinear least squares [II:48](#)
- in pool [II:482](#)
- in systems [II:312](#)
- AR terms [II:75](#)
- AR(1)
 - coefficient [II:64](#)
 - Durbin-Watson statistic [II:64](#)
 - estimation [II:68](#)
- AR(p) [II:64](#)
 - estimation [II:68](#)
- Arc cosine [I:754](#)
- Arc sine [I:754](#)
- Arc tangent [I:754](#)
- ARCH [II:185](#)
 - See also* GARCH.
 - correlogram test [II:154](#)
 - LM test [II:158](#)
 - multivariate [II:310](#)
 - system [II:310](#)
- ARCH test [II:158](#)
- ARCH-M [II:187](#)
- Area band graph [I:453](#)
- Area graph [I:451](#)
- AREMOS data [I:288](#)
- Arguments
 - in programs [I:608](#)
 - in subroutines [I:620](#)
- ARIMA models [II:71](#)
 - Box-Jenkins approach [II:72](#)
 - diagnostic checking [II:83](#)
 - difference operator [II:73](#)
 - identification [II:72](#)
 - specification [II:73](#)
 - starting values [II:79](#)
 - structure [II:83](#)
- ARMA terms [II:75](#)
 - in models [II:441](#)
 - seasonal [II:75](#)
 - testing [II:155](#)
 - using state spaces models for [II:392](#)
- ARMAX [II:392](#)
- Artificial regression [II:158](#), [II:177](#)
- ASCII file
 - export [I:106](#)
 - import [I:103](#), [I:119](#)
 - open as workfile [I:42](#)
- @asin [I:754](#)
- Assign values to matrix objects [I:628](#)
 - by element [I:628](#)
 - converting series or group [I:635](#)
 - copy [I:632](#)
 - copy submatrix [I:633](#)
 - fill procedure [I:629](#)
- Asymptotic test [II:141](#)
- @atan [I:754](#)
- Augmented Dickey-Fuller test [II:92](#)
 - See also* Unit root tests.
- Augmented regression [II:170](#)
- Autocorrelation [I:325](#), [II:14](#)
- Autocorrelation test
 - VAR [II:350](#)
- Automatic variable selection [II:55](#)
- Automation, *See* Programs.
- Autoregressive spectral density estimator [II:98](#)
- Auto-search
 - database [I:272](#)
- Auto-series [I:135](#)
 - forecasting [II:133](#)
 - generate new series [I:135](#), [I:270](#)
 - in estimation [I:139](#)
 - in groups [I:138](#)
 - in regression [I:270](#)
 - with database [I:269](#)
- Auto-updating series [I:145](#)
 - and databases [I:148](#)
 - converting to ordinary series [I:148](#)
- Autowrap text [I:593](#)
- Auxiliary commands [I:586](#)
- Auxiliary graphs [I:480](#)
- Auxiliary regression [II:155](#), [II:158](#)
- Average log likelihood [II:214](#)
- Average shifted histogram [I:466](#)
- Axis [I:441](#)
 - assignment [I:442](#)
 - characteristics [I:443](#)
 - customization by command [I:660](#)
 - format [I:445](#)
 - labels [I:445](#), [I:446](#)
 - location by command [I:662](#)
 - scale [I:443](#)
 - ticks and lines [I:446](#)

B

Backcast

- in GARCH models [II:191](#)

- MA terms [II:80](#)

Backup files [I:51](#), [I:765](#)Balanced data [II:469](#)Balanced sample [II:481](#)Band-Pass filter [I:361](#)Bandwidth [I:469](#), [I:484](#), [I:486](#)

- Andrews [II:339](#)

- bracketing [I:470](#), [I:485](#), [I:486](#)

- Newey-West (fixed) [II:339](#)

- Newey-West (variable) [II:340](#)

- selection in GMM [II:339](#)

- selection in system GMM [II:317](#)

Bar graph [I:451](#)Bartlett kernel [II:339](#)Bartlett test [I:320](#)Batch mode [I:596](#)

- See also* Program.

BDS test [I:327](#)Berndt-Hall-Hausman (BHHH). *See* Optimization algorithms.

Beta

- distribution [I:755](#)

- integral [I:751](#)

- integral, logarithm of [I:752](#)

@beta [I:751](#)@betainc [I:751](#)@betaincder [I:752](#)@betaincinv [I:752](#)@betalog [I:752](#)Bias proportion [II:124](#)

Bin width

- histograms [I:464](#)

Binary dependent variable [II:209](#)

- categorical regressors stats [II:217](#)

- error messages [II:215](#)

- estimation [II:211](#)

- estimation statistics [II:214](#)

- expectation-prediction table [II:217](#)

- fitted index [II:222](#)

- forecasting [II:221](#)

- goodness-of-fit [II:219](#)

- interpretation of coefficient [II:213](#)

- log likelihood [II:210](#)

- predicted probabilities [II:223](#)

- residuals [II:222](#)

- views [II:216](#)

Binary Estimation

- perfect predictor [II:216](#)

Binary file [I:42](#)Binning [I:470](#), [I:483](#), [I:485](#)

- categorical graphs [I:509](#)

- classifications [I:309](#), [I:392](#)

@binom [I:752](#)

Binomial

- coefficient function [I:752](#)

- coefficients, logarithm of [I:752](#)

- distribution [I:755](#)

Binomial sign test [I:313](#)@binomlog [I:752](#)Black and white [I:658](#)Blom [I:473](#)BMP [I:543](#)Bonferroni [I:386](#)

Boolean

- and [I:734](#)

- or [I:734](#)

Bootstrap [I:337](#)Box-Cox transformation [I:481](#)Box-Jenkins [II:72](#)Boxplot [I:477](#)

- as axis [I:423](#)

Break [I:9](#), [I:596](#)

Breakpoint test

- Chow [II:165](#)

- factor [II:152](#)

- Quandt-Andrews [II:166](#)

- unequal variance [II:176](#)

- unknown [II:166](#)

Breitung [II:106](#)Breusch-Godfrey test [II:65](#), [II:156](#)Breusch-Pagan test [II:156](#)Brown-Forsythe test [I:320](#)Broymen's method [II:447](#), [II:634](#)By-group statistics [I:308](#), [I:749](#), [II:521](#), [II:528](#)**C**

C

- coef vector [II:6](#)

- constant in regression [II:6](#)

- Cache [I:297](#)
- Call subroutine [I:621](#)
- Cancel
 - keystroke [I:9](#)
- Case-sensitivity [I:606](#)
- Categorical graphs [I:491](#)
 - See also* Graphs.
 - analytical [I:500](#)
 - binning [I:509](#)
 - category summaries [I:492](#)
 - descriptive statistics [I:492](#)
 - factor display settings [I:511](#)
 - factor labeling [I:520](#)
 - factor ordering [I:512](#)
 - factors [I:508](#)
 - identifying categories [I:503](#)
 - line [I:497](#)
 - specifying factors [I:508](#)
 - summaries [I:492](#)
- Categorical regressor stats [II:217](#), [II:237](#)
- Categorize [I:751](#)
- Category identifier
 - index [I:751](#)
 - number of [I:751](#)
- Causality
 - Granger's test [I:411](#)
- @ceiling [I:735](#)
- Cell
 - annotation [I:550](#)
 - formatting [I:548](#), [I:680](#)
 - merging [I:551](#)
 - selection [I:545](#)
- @cellid [I:731](#), [II:515](#)
- Censored dependent variable [II:232](#)
 - estimation [II:233](#)
 - expected dependent variable [II:239](#)
 - fitted index [II:238](#)
 - forecasting [II:238](#)
 - goodness-of-fit tests [II:239](#)
 - interpretation of coefficient [II:237](#)
 - log likelihood [II:233](#)
 - residuals [II:238](#)
 - scale factor [II:237](#)
 - specifying the censoring point [II:234](#)
 - views [II:237](#)
- Census X11
 - historical [I:348](#)
 - limitations [I:348](#)
 - using X12 [I:341](#)
- Census X12 [I:340](#)
 - seasonal adjustment options [I:341](#)
- Centered moving average [I:744](#), [I:746](#)
- CGARCH [II:200](#)
- Change default directory [I:61](#)
- Chi-square
 - independence test in tabulation [I:394](#)
 - statistic for Wald test [II:147](#)
 - test for independence in n-way table [I:394](#)
 - test for the median [I:318](#)
- Chi-square distribution [I:755](#)
- Cholesky factor
 - in VAR impulse responses [II:354](#)
 - in VAR normality test [II:350](#)
- Chow test
 - breakpoint [II:165](#)
 - forecast [II:168](#)
 - n-step forecast [II:174](#)
 - one-step forecast [II:174](#)
- Classification
 - from series [I:333](#)
- Classification table
 - binary models [II:217](#)
 - ordered models [II:230](#)
 - sensitivity [II:218](#)
 - specificity [II:218](#)
- @clear [I:148](#)
- Cleveland subsampling [I:488](#)
- Click(ing) [I:6](#)
- Clipboard [I:543](#)
- @cloglog [I:752](#)
- Close
 - EViews [I:13](#)
 - object [I:764](#)
 - window [I:7](#)
- Cochrane-Orcutt [II:41](#), [II:70](#)
- Coef (coefficient vector)
 - default [II:6](#)
 - update from equation [II:19](#)
- Coefficient
 - common (pool) [II:482](#)
 - covariance matrix [II:17](#)
 - covariance matrix of estimates [II:18](#)
 - cross-section specific (pool) [II:482](#)
 - estimated from regression [II:11](#)

- maximum number in default [II:230](#)
- recursive estimates [II:175](#)
- regression [II:11](#)
- restrictions [II:8](#)
- setting initial values [II:46](#), [II:625](#)
- standard error [II:12](#)
- vectors of [II:20](#)
- Coefficient restriction test [II:145](#)
- Coefficient restrictions [II:298](#)
 - cross-equation [II:311](#)
- Coefficient tests [II:142](#)
- Coefficient uncertainty [II:426](#), [II:435](#), [II:445](#)
- Cointegration test [II:363](#)
 - critical values [II:366](#), [II:377](#)
 - panel [II:372](#), [II:534](#)
 - pooled data [II:475](#)
 - restrictions [II:369](#)
 - VAR [II:363](#)
- Collinearity [II:23](#)
- Color
 - graph frame [I:439](#)
 - printing in [I:542](#)
 - tables [I:549](#)
- Column
 - number of columns in matrix [I:748](#)
 - width [I:548](#), [I:678](#)
- @columns [I:748](#)
- Command window [I:11](#)
- Commands
 - auxiliary [I:586](#)
 - batch mode [I:578](#)
 - interactive mode [I:577](#)
 - object assignment [I:580](#)
 - object command [I:582](#)
 - object declaration [I:581](#)
 - save record of [I:578](#)
 - window [I:577](#)
- Comments [I:72](#), [I:691](#)
 - spool [I:560](#)
 - tables [I:550](#)
- Common sample [I:129](#)
- Communalities [II:584](#)
- Comparison operators [I:640](#)
 - with missing values [I:129](#)
- Comparisons [I:124](#)
- Compatibility [I:603](#)
- Complementary log-log function [I:752](#)
- Component GARCH [II:200](#)
- Component plots [I:401](#)
- Concatenation [I:696](#)
- Conditional independence [I:395](#)
- Conditional standard deviation
 - graph [II:195](#)
- Conditional variance [II:183](#), [II:185](#), [II:186](#)
 - forecast [II:197](#)
 - in the mean equation [II:187](#)
 - make series [II:196](#)
- Confidence ellipses [I:489](#), [II:142](#)
- Confidence interval
 - ellipses [II:142](#)
 - for forecast [II:122](#)
 - for stochastic model solution [II:443](#)
- Constant
 - in equation [II:6](#), [II:12](#)
 - in ordered models [II:228](#)
- Container [I:586](#)
 - database [I:587](#)
 - workfile [I:586](#)
- Contingency coefficient [I:395](#)
- Continuation character in programs [I:593](#)
- Contracting data [I:233](#)
- Control variable [I:598](#)
 - as replacement variable [I:602](#)
 - in while statement [I:616](#)
 - save value to scalar [I:599](#)
- Convergence [II:627](#), [II:637](#)
 - logl [II:295](#)
- Convergence criterion [II:627](#)
 - in nonlinear least squares [II:47](#), [II:51](#)
 - in pool estimation [II:485](#)
- Convert
 - matrix objects [I:635](#), [I:648](#)
 - observation number to date [I:704](#)
 - panel to pool [I:240](#)
 - pool to panel [I:246](#)
 - series to matrix/vector [I:636](#)
 - string to scalar [I:600](#), [I:700](#)
- @convert [I:636](#)
- Copy [I:233](#)
 - and paste [I:73](#)
 - by link [I:234](#)
 - by value [I:235](#)
 - command [I:107](#)
 - data [I:104](#)

- data cut-and-paste [I:98](#)
- database [I:283](#)
- graph [I:542](#), [I:543](#)
- matrix [I:632](#)
- objects [I:73](#), [I:588](#)
- pool objects [II:462](#)
- table to clipboard [I:552](#)
- to and from database [I:265](#)
- to spool [I:556](#)
- Copy and paste [I:552](#)
- @cor [I:739](#)
- Correlation [I:739](#)
 - moving [I:745](#), [I:747](#)
- Correlogram [I:324](#), [I:327](#), [II:65](#)
 - autocorrelation function [I:325](#)
 - cross [I:409](#)
 - partial autocorrelation function [I:326](#)
 - Q-statistic [I:326](#)
 - squared residuals [II:154](#), [II:195](#)
 - VAR [II:349](#)
- @cos [I:754](#)
- Cosine [I:754](#)
- @count [I:141](#)
- Count models [II:246](#)
 - estimation [II:246](#)
 - forecasting [II:250](#)
 - negative binomial (ML) [II:247](#)
 - Poisson [II:247](#)
 - QML [II:248](#)
 - residuals [II:250](#)
- @cov [I:739](#)
- Covariance [I:739](#)
 - moving [I:745](#), [I:747](#)
- Covariance analysis [I:380](#)
 - details [I:387](#)
- Covariance matrix
 - HAC (Newey-West) [II:36](#)
 - heteroskedasticity consistent of estimated coefficients [II:35](#)
 - of estimated coefficients [II:17](#)
 - systems [II:322](#)
- Covariance proportion [II:124](#)
- Cramer's V [I:395](#)
- Cramer-von Mises test [I:321](#)
- Create
 - database [I:259](#)
 - dated data table [I:371](#)
 - factor [II:580](#)
 - graph [I:524](#)
 - graph by command [I:651](#)
 - group [I:84](#), [I:139](#)
 - link [I:187](#)
 - objects [I:67](#)
 - page [I:54](#)
 - program [I:593](#)
 - series [I:78](#), [I:131](#)
 - spool [I:554](#)
 - spool by command [I:687](#)
 - table [I:545](#), [I:675](#)
 - text object [I:554](#)
 - workfile [I:38](#)
- Cross correlation [I:409](#)
- Cross correlogram [I:409](#)
- Cross-equation
 - coefficient restriction [II:308](#), [II:311](#)
 - correlation [II:309](#), [II:310](#)
 - weighting [II:308](#)
- @crossid [I:731](#), [II:514](#)
- Cross-section
 - pool identifiers [II:461](#)
 - specific series [II:463](#)
 - summaries [II:523](#)
 - SUR [II:500](#)
- CSV [I:553](#)
- Cubic
 - frequency conversion method [I:108](#)
- @cumbmax [I:743](#)
- @cumbmean [I:742](#)
- @cumbmin [I:743](#)
- @cumbnas [I:743](#)
- @cumbobs [I:743](#)
- @cumbprod [I:742](#)
- @cumbstdev [I:742](#)
- @cumbstdevp [I:742](#)
- @cumbstdevs [I:743](#)
- @cumbsum [I:742](#)
- @cumbsumsq [I:743](#)
- @cumbvar [I:743](#)
- @cumbvarp [I:743](#)
- @cumbvars [I:743](#)
- @cummax [I:742](#)
- @cummean [I:741](#)
- @cummin [I:742](#)

- @cumnas [I:742](#)
- @cumobs [I:742](#)
- @cumprod [I:741](#)
- @cumstdev [I:741](#)
- @cumstdevp [I:741](#)
- @cumstdevs [I:741](#)
- @cumsum [I:741](#)
- @cumsumsq [I:742](#)
- Cumulative distribution [I:473](#)
 - computation [I:473](#)
- Cumulative statistics
 - functions [I:127](#)
 - maximum [I:742](#), [I:743](#)
 - mean [I:741](#), [I:742](#)
 - min [I:742](#)
 - minimum [I:743](#)
 - NAs, number of [I:742](#), [I:743](#)
 - observations, number of [I:742](#), [I:743](#)
 - product [I:741](#), [I:742](#)
 - standard deviation [I:741](#), [I:742](#), [I:743](#)
 - sum [I:741](#), [I:742](#)
 - sum of squares [I:742](#), [I:743](#)
 - variance [I:741](#), [I:742](#), [I:743](#)
- Cumulative statistics functions [I:741](#)
- @cumvar [I:741](#)
- @cumvarp [I:742](#)
- @cumvars [I:742](#)
- Customization
 - graphs [I:438](#)
- CUSUM
 - sum of recursive residuals test [II:172](#)
 - sum of recursive squared residuals test [II:173](#)

D

d [I:736](#)

Data

- appending more [I:230](#)
- cut and paste [I:100](#), [I:104](#)
- enter by keyboard [I:96](#)
- export [I:105](#)
- import [I:100](#)
- irregular [I:203](#)
- keyboard entry [I:98](#)
- pool [II:466](#)
- regular [I:203](#)
- remove [I:233](#)
- structure [I:203](#)

Database

- alias [I:272](#)
- auto-search [I:272](#)
- auto-series [I:271](#)
- cache [I:297](#)
- copy [I:283](#), [I:588](#)
- copy objects [I:265](#)
- create [I:259](#), [I:587](#)
- data storage precision [I:766](#)
- default [I:262](#)
- default in search order [I:273](#)
- delete [I:283](#), [I:588](#)
- delete objects [I:267](#)
- display all objects [I:260](#)
- export [I:265](#)
- fetch objects [I:263](#)
- field [I:275](#)
- foreign formats [I:285](#)
- frequency in query [I:277](#)
- group storage options [I:766](#)
- link [I:264](#), [I:265](#), [I:296](#)
- link options [I:299](#)
- maintenance [I:283](#)
- match operator in query [I:277](#)
- open [I:259](#), [I:588](#)
- packing [I:283](#)
- queries [I:273](#)
- rebuild [I:284](#)
- registry [I:766](#)
- rename [I:283](#), [I:588](#)
- rename object [I:267](#)
- repair [I:284](#)
- sharing violation [I:260](#)
- statistics [I:283](#)
- store objects [I:262](#)
- test integrity [I:284](#)
- using auto-updating series with [I:148](#)
- window [I:259](#)

Database registry [I:271](#)

Datastream [I:295](#)

Date

- @-functions [I:759](#)

@date [I:722](#), [I:728](#)

Date functions [I:728](#)

Date pairs [I:87](#)

Date series [I:156](#)

@dateadd [I:720](#)

Dated data table [I:370](#)

- create [I:371](#)
- formatting options [I:376](#)
- frequency conversion [I:374](#)
- row options [I:377](#)
- table options [I:372](#)
- transformation methods [I:374](#)
- @datediff [I:720](#)
- @datefloor [I:721](#)
- @datepart [I:721](#)
- Dates
 - arithmetic [I:720](#)
 - convert from observation number [I:704](#)
 - converting from numbers [I:716](#)
 - current date and time [I:722](#)
 - current date as a string [I:704](#)
 - date arithmetic [I:720](#), [I:721](#)
 - date associated with observation [I:722](#)
 - date numbers [I:706](#)
 - date operators [I:719](#)
 - date strings [I:705](#)
 - default display format [I:766](#)
 - display format [I:156](#)
 - extract portion of [I:721](#)
 - format strings [I:707](#)
 - formatting [I:707](#)
 - global options [I:766](#)
 - match merging using [I:179](#)
 - rounding [I:721](#)
 - string representations [I:701](#), [I:705](#), [I:715](#)
 - two-digit years [I:712](#), [I:724](#)
 - workfile dates as strings [I:704](#), [I:722](#), [I:729](#)
- @datestr [I:715](#)
- @dateval [I:701](#), [I:715](#)
- @day [I:722](#), [I:729](#)
- Declare
 - matrix [I:627](#)
 - object [I:581](#)
- Default
 - database [I:12](#), [I:262](#)
 - database in search order [I:273](#)
 - directory [I:12](#), [I:764](#)
 - set directory [I:61](#)
 - setting global options [I:763](#)
 - update directory [I:61](#)
- Delete [I:74](#)
 - database [I:283](#)
 - graph element [I:529](#)
 - object [I:590](#)
 - objects from database [I:267](#)
 - observation in series [I:83](#)
 - page [I:60](#)
 - series using pool [II:478](#)
 - spool objects [I:563](#)
- Delimiters [I:113](#)
- Demonstration
 - estimation [I:23](#)
 - examining data [I:18](#)
 - forecasting [I:30](#)
 - getting data into EViews [I:15](#)
 - specification test [I:26](#)
- Dependent variable
 - no variance in binary models [II:216](#)
- Derivatives [II:628](#), [II:640](#)
 - checking [II:295](#)
 - description [II:640](#)
 - in equation [II:18](#)
 - in logl [II:288](#)
 - in system [II:322](#)
 - saving in series [II:644](#)
- Description
 - field in database query [I:279](#)
- Descriptive statistics [I:125](#)
 - balanced sample (pool) [II:474](#)
 - by classification [I:308](#)
 - by group [I:308](#)
 - categorical graphs of [I:492](#)
 - common sample (group) [I:380](#)
 - common sample (pool) [II:474](#)
 - cross-section specific [II:474](#)
 - for a series [I:310](#)
 - @-functions [I:738](#), [I:749](#)
 - graphs of [I:492](#)
 - group [I:379](#)
 - individual samples (group) [I:380](#)
 - individual samples (pool) [II:474](#)
 - pooled [II:473](#)
 - series [I:306](#)
 - stacked data [II:474](#)
 - tests [I:311](#)
 - time period specific [II:474](#)
- Deselect all [I:68](#)
- DFGLS [II:94](#)
- Dialog box [I:9](#)
- Dickey-Fuller test [II:92](#)
 - See also* Unit root tests.

-
- Difference from moving-average [I:354](#)
 - Difference operator [I:125](#), [I:126](#), [I:736](#), [II:73](#)
 - seasonal [I:127](#), [II:74](#)
 - @digamma [I:752](#)
 - Digamma function [I:752](#)
 - Display
 - action [I:582](#)
 - and print [I:583](#)
 - Display filter [I:47](#), [I:203](#)
 - Display format
 - for groups [I:85](#)
 - Display mode
 - spools [I:568](#)
 - Display name
 - field in database query [I:279](#)
 - Distribution
 - empirical distribution function tests [I:321](#)
 - tests [I:321](#)
 - Distribution functions [I:754](#)
 - Division operator (/) [I:640](#)
 - dlog [I:736](#)
 - Doornik and Hansen factorization matrix [II:351](#)
 - Dot plot [I:455](#)
 - Drag(ging) [I:6](#)
 - text in graph [I:527](#)
 - DRI database
 - DRipro [I:296](#)
 - frequency [I:299](#)
 - illegal names [I:299](#)
 - object alias [I:281](#)
 - queries [I:300](#)
 - shadowing of object names [I:282](#)
 - troubleshooting [I:301](#)
 - DRIBase database [I:287](#)
 - DRIPro link [I:286](#)
 - Dummy variables [I:141](#)
 - as binary dependent variable [II:209](#)
 - as censoring point in estimation [II:235](#)
 - automatic creation in estimation [II:28](#)
 - generating pool series dummies [II:472](#)
 - pools [II:472](#)
 - using @GROUP to create pool dummies [II:472](#)
 - Dunn-Sidak [I:386](#)
 - Durbin's h [I:619](#)
 - Durbin-Watson statistic [II:64](#)
 - demonstration [I:316](#)
 - for regression [II:14](#)
 - lagged dependent variable [II:65](#)
 - Dynamic forecasting [II:126](#)
 - Dynamic panel data [II:545](#)
- ## E
- Easy query [I:273](#)
 - Economy.com [I:296](#)
 - EcoWin database [I:290](#)
 - Edit
 - group [I:86](#)
 - series [I:82](#), [I:368](#)
 - table [I:547](#)
 - EGARCH [II:198](#)
 - Eigenvalues
 - factor analysis [II:593](#)
 - plots [I:400](#)
 - @elem [I:124](#), [I:727](#)
 - Element
 - assign in matrix [I:628](#)
 - Elliot, Rothenberg, and Stock point optimal test [II:96](#)
 - See also* Unit root tests.
 - Else clause in if statement [I:611](#)
 - Embedded spools [I:557](#)
 - EMF [I:543](#)
 - Empirical CDF graph [I:473](#)
 - Empirical distribution tests [I:321](#)
 - Empirical quantile graph [I:475](#)
 - Empirical survivor graph [I:474](#)
 - Encrypted program file [I:594](#)
 - End field [I:278](#)
 - @enddate [I:728](#)
 - Endogeneity [II:177](#)
 - Endogenous variable [II:37](#)
 - Enhanced metafile [I:543](#)
 - Enterprise Edition [I:287](#), [I:290](#)
 - Epanechnikov kernel [I:469](#)
 - @eqna [I:700](#), [I:735](#)
 - @eqnq [I:130](#)
 - Equality tests [I:314](#)
 - groups [I:395](#)
 - mean [I:315](#)
 - median [I:317](#)
 - variance [I:319](#)

- Equals comparison [I:640](#)
- Equation [II:5](#)
 - automatic dummy variables in [II:28](#)
 - coefficient covariance matrix [II:17](#)
 - coefficient covariance scalar [II:16](#)
 - coefficient standard error vector [II:16](#), [II:17](#)
 - coefficient t-statistic scalar [II:16](#)
 - coefficient t-statistic vector [II:17](#)
 - coefficient vector [II:17](#), [II:20](#)
 - create [II:5](#)
 - derivatives [II:18](#)
 - gradients [II:18](#)
 - procedures [II:18](#)
 - regression coefficients [II:11](#)
 - regression summary statistics [II:13](#)
 - residuals [II:19](#)
 - results [II:11](#)
 - retrieve previously estimated [II:20](#)
 - r-squared [II:13](#)
 - saved results [II:16](#), [II:19](#)
 - scalar results [II:16](#)
 - specification [II:6](#)
 - specification by list [II:6](#)
 - specify by formula [II:7](#)
 - specify with non-default coefs [II:9](#)
 - specify with restrictions [II:8](#)
 - specify without dependent variable [II:8](#)
 - specifying a constant [II:6](#)
 - store [II:20](#)
 - text representation [II:17](#)
 - t-statistic [II:12](#)
 - vector and matrix results [II:17](#)
 - views [II:17](#)
- @erf [I:752](#)
- @erfc [I:753](#)
- Error
 - handling in programs [I:617](#)
- Error bar graph [I:456](#)
- Error control [I:617](#)
- Error function [I:752](#)
 - complementary [I:753](#)
- Estimation [II:9](#)
 - AR specification [II:67](#)
 - auto-series [I:139](#)
 - behavior [II:637](#)
 - binary dependent variable [II:211](#)
 - censored models [II:233](#)
 - collinearity [II:21](#)
 - convergence [II:637](#)
 - convergence problems [II:628](#)
 - count models [II:246](#)
 - demonstration [I:23](#)
 - derivative computation options [II:628](#)
 - derivatives [II:640](#)
 - failure to improve [II:628](#)
 - for pool [II:480](#)
 - log likelihood [II:290](#)
 - logl [II:290](#)
 - missing data [II:9](#)
 - multi-equation [II:308](#)
 - near singular matrix problems [II:627](#)
 - nonlinear least squares [II:43](#)
 - options [II:625](#)
 - ordered models [II:227](#)
 - output [II:11](#)
 - panel
 - problems in convergence [II:626](#)
 - residuals from equation [II:19](#)
 - sample [II:9](#)
 - sample adjustment [II:10](#)
 - single equation methods [II:9](#)
 - starting values [II:625](#)
 - state space [II:387](#), [II:397](#)
 - systems [II:308](#), [II:315](#)
 - truncated models [II:242](#)
 - two-stage least squares [II:37](#)
 - VAR [II:346](#)
 - VEC [II:378](#)
 - weighted least squares [II:35](#)
- Estimation options [II:625](#)
- Euler's constant [I:753](#)
- Evaluation order [I:122](#)
 - logl [II:287](#)
- EViews Databases [I:257](#)
- EViews Enterprise Edition [I:287](#), [I:288](#), [I:290](#)
- Examining data
 - demonstration [I:18](#)
- Excel file
 - importing into workfile [I:100](#)
 - opening as workfile [I:42](#)
 - opening as workfile demo [I:15](#)
- Execute program [I:595](#)
 - abort [I:596](#)
 - quiet mode [I:595](#)
 - verbose mode [I:595](#)
 - with arguments [I:607](#), [I:608](#)

Exit
 loop [I:618](#)
 Exogenous variable [II:37](#)
 uncertainty [II:435](#), [II:446](#)
 @exp [I:735](#)
 exp [I:735](#)
 @expand [I:141](#), [II:28](#)
 Expectation-prediction table
 binary models [II:217](#)
 ordered models [II:230](#)
 Expectations consistency in models [II:437](#)
 Expected dependent variable
 censored models [II:239](#)
 truncated models [II:244](#)
 Expected latent variable
 censored models [II:238](#)
 truncated models [II:244](#)
 Exponential
 distribution [I:755](#)
 function [I:735](#)
 Exponential GARCH (EGARCH) [II:198](#)
 Exponential smoothing [I:354](#), [I:360](#)
 See also Smoothing.
 double [I:356](#)
 Holt-Winters additive [I:357](#)
 Holt-Winters multiplicative [I:357](#)
 Holt-Winters no-seasonal [I:358](#)
 single [I:355](#)
 Export
 database [I:265](#)
 matrix [I:646](#)
 tables [I:684](#)
 Export data [I:105](#), [I:254](#)
 from pool objects [II:478](#)
 to ASCII files [I:106](#)
 to spreadsheet files [I:105](#)
 Expression [I:121](#)
 for database fields [I:276](#)
 parentheses [I:122](#)
 Extreme value
 binary model [II:212](#)
 censored dependent variable models [II:234](#)
 distribution [I:755](#)

F

@fact [I:735](#)
 @factlog [I:735](#)
 Factor analysis [II:579](#)
 background [II:610](#)
 communalities [II:584](#)
 creation [II:580](#)
 data members [II:595](#)
 details [II:610](#)
 eigenvalues [II:593](#)
 example [II:595](#)
 goodness of fit [II:591](#), [II:615](#)
 graph scores [II:589](#)
 loading views [II:592](#)
 method [II:581](#), [II:583](#)
 method details [II:612](#)
 model evaluation [II:615](#)
 PACE [II:583](#)
 procedures [II:594](#)
 reduced covariance [II:592](#)
 rotation [II:586](#)
 rotation (theory) [II:617](#)
 scaling [II:585](#)
 score estimation [II:587](#)
 specification [II:580](#)
 theory of [II:610](#)
 views [II:590](#)
 Factor and graph layout options [I:515](#)
 Factor breakpoint test [II:152](#)
 Factor display settings [I:511](#)
 Factorial [I:735](#)
 log of [I:735](#)
 Factset [I:295](#)
 Fair-Taylor model solution [II:436](#)
 FAME database [I:287](#)
 F-distribution [I:756](#)
 Fetch [I:75](#)
 from database [I:263](#)
 from pool [II:478](#)
 object [I:591](#)
 Fetch command [I:107](#)
 Fields in database [I:279](#)
 description [I:279](#)
 display_name [I:279](#)
 end [I:278](#)
 expressions [I:276](#)
 freq [I:277](#)
 history [I:279](#)
 last_update [I:279](#)
 last_write [I:279](#)

- name [I:277](#)
- remarks [I:279](#)
- source [I:279](#)
- start [I:278](#)
- type [I:277](#)
- units [I:279](#)
- Files
 - default locations [I:764](#)
- Fill
 - matrix [I:629](#)
 - vector [I:629](#)
- Filter
 - state space models [II:384](#)
- FIML [II:310](#)
- Financial function
 - future value [I:737](#)
 - number of periods [I:737](#)
 - payment amount [I:738](#)
 - present value [I:737](#)
 - rate for annuity [I:738](#)
- Financial functions [I:737](#)
- @first [I:88](#)
- First derivative methods [II:631](#)
- First obs
 - row-wise [I:749](#)
 - row-wise index [I:749](#)
- @firstmax [II:520](#)
- @firstmin [II:520](#)
- Fisher-ADF [II:109](#)
- Fisher-PP [II:109](#)
- Fit lines (graph) [I:437](#)
- Fitted index
 - binary models [II:222](#)
 - censored models [II:238](#)
 - truncated models [II:244](#)
- Fitted probability
 - binary models [II:221](#)
- Fitted values
 - of equation [II:17](#)
- Fixed effects
 - pool [II:483](#)
 - pool description [II:498](#)
- Fixed effects test [II:566](#)
- Fixed variance parameter
 - negative binomial QML [II:250](#)
 - normal QML [II:249](#)
- Flatten
 - spools [I:564](#)
- @floor [I:735](#)
- Fonts
 - defaults [I:763](#)
 - tables [I:549](#)
 - text in graph [I:526](#), [I:536](#)
- For loop [I:612](#)
 - accessing elements of a series [I:613](#)
 - accessing elements of a vector [I:613](#)
 - changing samples within [I:613](#)
 - define using control variables [I:612](#)
 - define using scalars [I:614](#)
 - define using string variables [I:614](#)
 - exit loop [I:618](#)
 - nesting [I:613](#)
- Forecast
 - AR specification [II:128](#)
 - auto-series [II:133](#)
 - backcasting [II:128](#)
 - binary models [II:221](#)
 - by exponential smoothing [I:360](#)
 - censored models [II:238](#)
 - Chow test [II:168](#)
 - conditional variance [II:197](#)
 - count models [II:250](#)
 - demonstration [I:30](#)
 - dynamic [II:126](#), [II:386](#)
 - equations with formula [II:132](#)
 - error [II:121](#)
 - evaluation [II:123](#)
 - example [II:116](#)
 - expressions and auto-updating series [II:132](#)
 - fitted values [II:120](#)
 - from estimated equation [II:113](#)
 - innovation initialization in models [II:438](#)
 - interval [II:122](#)
 - lagged dependent variables [II:125](#)
 - MA specification [II:128](#)
 - missing values [II:120](#)
 - nonlinear models [II:138](#)
 - n-step ahead [II:386](#)
 - n-step test [II:174](#)
 - one-step test [II:174](#)
 - ordered models [II:231](#)
 - out-of-sample [II:119](#)
 - PDLs [II:138](#)
 - smoothed [II:387](#)
 - standard error [II:122](#), [II:135](#)

- state space [II:386](#)
 - static [II:127](#)
 - structural [II:127](#)
 - system [II:323](#)
 - truncated models [II:244](#)
 - VAR/VEC [II:379](#)
 - variance [II:121](#)
 - with AR errors [II:128](#)
 - Foreign data
 - open as workfile [I:15](#)
 - Format
 - number [I:677](#)
 - tables [I:548](#)
 - Formula
 - forecast [II:132](#)
 - implicit assignment [I:133](#)
 - normalize [I:134](#)
 - specify equation by [II:7](#)
 - Forward solution for models [II:435](#)
 - Frame [I:439](#)
 - size [I:441](#)
 - Freedman-Diaconis [I:464](#)
 - Freeze [I:74](#)
 - create graph from view [I:524](#)
 - table [I:675](#)
 - Freq
 - field in database query [I:277](#)
 - Frequency (Band-Pass) filter [I:361](#)
 - Frequency conversion [I:73](#), [I:74](#), [I:106](#), [I:766](#)
 - dated data table [I:374](#)
 - default settings [I:109](#)
 - DRI database [I:299](#)
 - links [I:194](#)
 - methods [I:107](#)
 - panels [I:184](#)
 - propagate NAs [I:107](#)
 - undated series [I:109](#)
 - using links [I:183](#)
 - Frequency zero spectrum estimation [II:97](#)
 - F-statistic [II:147](#), [II:150](#)
 - for regression [II:15](#)
 - F-test
 - for variance equality [I:319](#)
 - Full information maximum likelihood [II:310](#)
 - @-functions
 - by-group statistics [I:749](#)
 - cumulative statistics functions [I:741](#)
 - descriptive statistics [I:738](#)
 - financial functions [I:737](#)
 - group row functions [I:748](#)
 - integrals and other special functions [I:751](#)
 - mathematical functions [I:735](#)
 - moving statistics functions [I:743](#)
 - special p -value functions [I:757](#)
 - statistical distribution functions [I:754](#)
 - time series functions [I:736](#)
 - trigonometric functions [I:754](#)
 - Future value [I:737](#)
 - @fv [I:737](#)
- ## G
- Gamma
 - distribution [I:756](#)
 - @gamma [I:753](#)
 - Gamma function [I:753](#)
 - derivative [I:752](#), [I:753](#)
 - incomplete [I:753](#)
 - incomplete derivative [I:753](#)
 - incomplete inverse [I:753](#)
 - logarithm [I:753](#)
 - second derivative [I:754](#)
 - @gammader [I:753](#)
 - @gammainc [I:753](#)
 - @gammaincder [I:753](#)
 - @gammaincinv [I:753](#)
 - @gammalog [I:753](#)
 - GARCH [II:185](#)
 - ARCH-M model [II:187](#)
 - asymmetric component model [II:201](#)
 - backcasting [II:191](#)
 - component models (CGARCH) [II:200](#)
 - estimation in EViews [II:188](#)
 - examples [II:193](#)
 - exponential GARCH (EGARCH) [II:198](#)
 - GARCH(1,1) model [II:185](#)
 - GARCH(p,q) model [II:187](#)
 - initialization [II:191](#)
 - Integrated GARCH (IGARCH) [II:198](#)
 - mean equation [II:189](#)
 - multivariate [II:304](#)
 - Power ARCH (PARCH) [II:199](#)
 - procedures [II:196](#)
 - robust standard errors [II:192](#)
 - threshold (TARCH) [II:198](#)

- variance equation [II:189](#)
- Gauss file [I:42](#)
- Gaussian distribution [I:756](#)
- Gauss-Newton [II:632](#)
- Gauss-Seidel algorithm [II:447](#), [II:633](#)
- Generalized error distribution [I:756](#), [II:199](#)
- Generalized least squares [II:483](#)
- Generalized linear models [II:256](#)
 - quasi-likelihood ratio test [II:250](#)
 - robust standard errors [II:256](#)
 - variance factor [II:256](#)
- Generalized method of moments, *See* GMM.
- Generalized residual
 - binary models [II:222](#)
 - censored models [II:238](#)
 - count models [II:251](#)
 - ordered models [II:232](#)
 - score vector [II:223](#)
 - truncated models [II:244](#)
- Generate series [I:131](#)
 - by command [I:134](#)
 - dynamic assignment [I:133](#)
 - for pool [II:472](#)
 - implicit assignment [I:133](#)
 - implicit formula [I:133](#)
 - using samples [I:132](#)
- Geometric mean [I:740](#)
- Geometric moving average [I:138](#)
- GiveWin data [I:288](#)
- Glejser heteroskedasticity test [II:157](#)
- GLM (generalized linear model) [II:256](#)
 - standard errors [II:256](#)
- Global
 - subroutine [I:622](#)
 - variable [I:622](#)
- Global optimum [II:627](#)
- GLS
 - detrending [II:94](#)
 - pool estimation details [II:499](#)
 - weights [II:543](#)
- @gmean [I:740](#)
- GMM
 - bandwidth selection [II:317](#)
 - for systems [II:310](#)
 - HAC weighting matrix [II:338](#)
 - J-statistic [II:54](#)
 - kernel options [II:317](#)
 - multi-equation [II:310](#)
 - orthogonality condition [II:51](#)
 - overidentifying restrictions [II:55](#)
 - panels [II:545](#)
 - prewhitening option [II:317](#), [II:340](#)
 - single equation [II:51](#)
 - system [II:337](#)
 - White weighting matrix [II:338](#)
- Godfrey heteroskedasticity test [II:156](#)
- Goldfeld-Quandt [II:631](#)
- Gompit models [II:212](#)
- Goodness-of-fit
 - adjusted R-squared [II:13](#)
 - Andrews test [II:219](#), [II:257](#)
 - factor analysis [II:591](#)
 - forecast [II:123](#)
 - Hosmer-Lemeshow test [II:219](#), [II:256](#)
 - R-squared [II:13](#)
- Gradients [II:637](#)
 - details [II:637](#)
 - in equation [II:18](#), [II:322](#)
 - in logl [II:294](#)
 - saving in series [II:639](#)
 - summary [II:638](#)
- Granger causality [I:410](#)
 - VAR [II:348](#)
- Graph
 - add text [I:668](#)
 - align multiple [I:540](#)
 - analytical graphs [I:462](#)
 - area band [I:453](#)
 - area graph [I:451](#)
 - automating [I:544](#)
 - auxiliary graphs [I:480](#)
 - average shifted histogram [I:466](#)
 - axis borders [I:423](#)
 - axis by command [I:660](#)
 - axis control [I:532](#)
 - axis label format [I:445](#)
 - axis labels [I:533](#)
 - axis location by command [I:662](#)
 - axis *see also* Axis.
 - background color [I:532](#)
 - background printing [I:532](#)
 - bar graph [I:451](#)
 - basic customization [I:438](#)
 - border [I:532](#)
 - boxplot [I:477](#)

categorical *See also* Categorical graphs.
 change types [I:655](#)
 color by command [I:656](#)
 color settings [I:532](#)
 combining [I:525](#)
 confidence ellipse [I:489](#)
 coordinates for positioning elements [I:526](#)
 copying [I:542](#)
 create by command [I:651](#)
 creating [I:524](#)
 custom labels [I:533](#)
 custom obs labels [I:533](#)
 customization [I:525](#)
 customization by command [I:656](#)
 customize axis labels [I:445](#)
 customizing lines and symbols [I:534](#)
 data option [I:421](#)
 dot plot [I:455](#)
 drawing lines and shaded areas [I:527](#)
 empirical CDF [I:473](#)
 empirical log survivor [I:474](#)
 empirical quantile [I:475](#)
 empirical survivor [I:474](#)
 error bar [I:456](#)
 fill areas [I:449](#)
 first vs. all [I:434](#)
 fit lines [I:437](#)
 font [I:526](#)
 font options [I:536](#)
 frame [I:439](#)
 frame color [I:439](#)
 frame customization by command [I:665](#)
 frame fill [I:532](#)
 freeze [I:524](#)
 frequency [I:422](#)
 grid lines [I:532](#)
 groups [I:427](#)
 high-low-open-close [I:456](#)
 histogram [I:463](#)
 histogram edge polygon [I:466](#)
 histogram polygon [I:465](#)
 indentation [I:532](#)
 kernel density [I:467](#)
 kernel regression [I:483](#)
 label by command [I:672](#)
 legend by command [I:667](#)
 legend options [I:534](#)
 legend settings [I:534](#)

line characteristics [I:656](#)
 line formats [I:447](#)
 line graph [I:450](#)
 link frequency [I:422](#)
 location [I:440](#)
 means [I:421](#)
 merge [I:655](#)
 merging multiple [I:68](#)
 mixed frequency data [I:431](#)
 mixed line [I:454](#)
 modifying [I:530](#)
 multiple graph options [I:539](#)
 multiple series option [I:429](#)
 nearest neighbor regression [I:485](#)
 non-consecutive observations [I:532](#)
 observation graphs [I:450](#)
 orientation [I:421](#)
 orthogonal regression [I:488](#)
 pairwise data [I:433](#)
 panel data [II:529](#)
 panel data options [I:424](#)
 patterns by command [I:658](#)
 pie [I:460](#)
 place text in [I:526](#)
 position [I:440](#), [I:540](#)
 print in color [I:542](#)
 printing [I:542](#)
 quantile-quantile [I:475](#), [I:477](#)
 raw data [I:421](#)
 regression line [I:480](#)
 remove element [I:529](#)
 remove elements [I:529](#)
 rotate [I:421](#)
 sample break plotting options [I:532](#)
 saving [I:543](#)
 scale [I:443](#)
 scatter [I:457](#)
 scatterplot matrix [I:435](#)
 scores [II:589](#)
 seasonal [I:461](#)
 series [I:419](#)
 series view [I:306](#)
 settings for multiple graphs [I:539](#)
 shade options [I:535](#)
 shading by command [I:669](#)
 size [I:441](#)
 size by command [I:665](#)
 sorting [I:528](#)

- sorting observations [I:528](#)
- spike [I:454](#)
- stacked [I:434](#)
- symbol graph [I:450](#)
- symbols [I:447](#)
- template by command [I:669](#)
- templates [I:536](#)
- text justification [I:526](#)
- text options [I:535](#)
- theoretical distribution [I:471](#)
- type [I:419](#), [I:428](#), [I:449](#), [I:530](#)
- XY area [I:458](#)
- XY bar [I:459](#)
- XY line [I:458](#)
- XY pairs [I:434](#)
- Greater than comparison [I:640](#)
- Greater than or equal to comparison [I:640](#)
- Grid lines [I:440](#)
 - table [I:548](#)
- Grid search [II:633](#)
- Group [I:139](#), [I:367](#)
 - add member [I:367](#)
 - adding series [I:367](#)
 - auto-series [I:138](#)
 - convert to matrix [I:636](#)
 - create [I:84](#), [I:139](#)
 - creating using wildcards [I:776](#)
 - display format [I:85](#)
 - display type [I:79](#)
 - edit series [I:368](#)
 - editing [I:86](#)
 - element [I:140](#)
 - graph view [I:379](#)
 - graphing [I:427](#)
 - make system of equations [I:412](#)
 - number of series [I:141](#)
 - pool [II:464](#)
 - rearranging series [I:367](#)
 - row functions [I:141](#), [I:748](#)
 - spreadsheet view [I:368](#)
 - spreadsheet view defaults [I:769](#)
 - summaries [I:370](#)
- Group into bins option [I:309](#), [I:392](#)
- @groupid [I:751](#)
- Groupwise heteroskedasticity [I:396](#)
- Gumbel [I:473](#)

H

- Hadri [II:107](#)
- Hannan-Quinn criterion [II:645](#)
- Harvey heteroskedasticity test [II:157](#)
- Hatanaka two step estimator [II:70](#)
- Hausman test [II:177](#), [II:568](#)
- Haver Analytics Database [I:288](#)
- Help [I:13](#)
 - help system [I:13](#)
 - World Wide Web [I:14](#)
- Heteroskedasticity
 - binary models [II:225](#)
 - consistent covariance [II:35](#)
 - cross-sectional details [II:499](#)
 - groupwise [I:396](#)
 - of known form [II:32](#)
 - of unknown form [II:35](#)
 - period details [II:500](#)
 - tests of [II:156](#)
 - White's test [II:158](#)
 - wizard [II:160](#)
- Heteroskedasticity and autocorrelation consistent covariance (HAC) [II:36](#)
- Heteroskedasticity consistent covariance matrix [II:35](#)
- Heywood cases [II:586](#)
- Hide
 - objects in spool [I:560](#)
- High-low-open-close graph [I:456](#)
- Hildreth-Lu [II:70](#)
- Histogram [I:306](#)
 - as axis [I:423](#)
 - average shifted graph [I:466](#)
 - bin width [I:464](#)
 - edge polygon graph [I:466](#)
 - graph [I:463](#)
 - normality test [II:155](#)
 - polygon graph [I:465](#)
 - variable width [I:459](#)
- History
 - field in database query [I:279](#)
- Hodrick-Prescott filter [I:360](#), [I:361](#)
- Holt-Winters
 - additive [I:357](#)
 - multiplicative [I:357](#)
 - no-seasonal [I:358](#)

Hosmer-Lemeshow test [II:219](#), [II:256](#)

HTML [I:553](#)

open page as workfile [I:42](#)

save table as

save table as web page [I:553](#)

Huber/White standard errors [II:255](#)

Hypothesis tests

ARCH [II:158](#)

Bartlett test [I:320](#)

BDS independence [I:313](#)

binomial sign test [I:313](#)

Brown-Forsythe [I:320](#)

chi-square test [I:318](#)

Chow breakpoint [II:165](#)

coefficient based [II:142](#)

coefficient p-value [II:12](#)

CUSUM [II:172](#)

CUSUM of squares [II:173](#)

demonstration [I:26](#)

descriptive statistic tests [I:311](#)

distribution [I:321](#)

F-test [I:319](#)

Hausman test [II:177](#)

heteroskedasticity [II:156](#)

irrelevant or redundant variable [II:151](#)

Kruskal-Wallis test [I:318](#)

Levene test [I:320](#)

mean [I:311](#)

median [I:313](#)

multi-sample equality [I:314](#)

nonnested [II:178](#)

normality [II:155](#)

omitted variables [II:150](#)

Ramsey RESET [II:170](#)

residual based [II:154](#)

See also Tests.

Siegel-Tukey test [I:319](#)

single sample [I:311](#)

stability test [II:164](#)

unit root [I:327](#), [II:88](#)

unknown breakpoint [II:166](#)

Van der Waerden test [I:313](#), [I:318](#)

variance [I:312](#)

Wald coefficient restriction test [II:145](#)

White heteroskedasticity [II:158](#)

Wilcoxon signed ranks test [I:313](#)

I

Icon [I:65](#)

Identification

Box-Jenkins [II:72](#)

nonlinear models [II:51](#)

structural VAR [II:361](#)

Identity

in model [II:408](#)

in system [II:312](#)

If condition in samples [I:88](#)

If statement [I:610](#)

else clause [I:611](#)

@iff [I:735](#)

IGARCH [II:198](#)

Im, Pesaran and Shin [II:108](#)

Import Data [I:95](#)

Import data

for pool objects [II:466](#)

from ASCII [I:103](#), [I:111](#)

from spreadsheet [I:100](#)

matrix [I:646](#)

See also Foreign data.

using a pool object [II:469](#)

Impulse response [II:353](#)

See also VAR.

generalized impulses [II:355](#)

standard errors [II:354](#)

structural decomposition [II:355](#)

transformation of impulses [II:354](#)

user specified impulses [II:355](#)

Include

program file [I:618](#)

Incomplete beta [I:751](#)

derivative [I:752](#)

integral [I:751](#)

inverse [I:752](#)

Incomplete gamma [I:753](#)

derivative [I:753](#)

inverse [I:753](#)

Incorrect functional form [II:159](#), [II:170](#)

Indent

spools [I:563](#)

Independence test [I:327](#)

Index

fitted from binary models [II:222](#)

fitted from censored models [II:238](#)

fitted from truncated models [II:244](#)

Individual sample [I:129](#)

Information criterion

- Akaike [II:15](#), [II:645](#)
- Hannan-Quinn [II:645](#)
- Schwarz [II:15](#), [II:645](#)

@ingrp [II:464](#)

@inner [I:739](#)

Inner product [I:739](#)

- moving [I:746](#), [I:748](#)

Innovation [II:63](#)

Insert

- observation [I:83](#)

@insert [I:702](#)

Insertion point [I:11](#)

- in command line [I:577](#)

Installation [I:5](#)

@instr [I:700](#)

Instrumental variable [II:37](#)

- for 2SLS with AR specification [II:41](#)
- for GMM [II:51](#)
- for nonlinear 2SLS [II:49](#)
- identification [II:315](#)
- in systems [II:312](#)
- order condition [II:38](#)
- using PDL specifications [II:25](#)
- with pooled data [II:502](#)

Integer dependent variable [II:246](#)

Integrated series [II:87](#)

Integrity (database) [I:284](#)

Interactive mode [I:577](#)

Intercept in equation [II:6](#), [II:12](#)

@inv [I:735](#)

Invalid date identifiers [I:226](#)

Inverted AR roots [II:70](#), [II:77](#)

Inverted MA roots [II:77](#)

Irregular data [I:203](#)

Irrelevant variable test [II:151](#)

@isempty [I:700](#)

@isna [I:130](#), [I:735](#)

@isperiod [I:729](#)

Iteration [II:627](#)

- failure to improve message [II:628](#)
- in models [II:450](#)
- in nonlinear least squares [II:47](#)

J

Jarque-Bera statistic [I:308](#), [II:155](#), [II:196](#)

- in VAR [II:350](#)

JPEG [I:543](#)

J-statistic [II:54](#)

- ordinary equation [II:55](#)
- panel equation [II:562](#)

J-test [II:179](#)

K

Kaiser-Guttman [II:611](#)

Kalman filter [II:385](#)

Kao panel cointegration test [II:375](#)

Kendall's tau [I:380](#)

- theory [I:388](#)

Kernel [I:484](#)

- choice in HAC weighting [II:317](#), [II:339](#)

Kernel density graph [I:467](#)

Kernel functions [I:468](#)

Kernel regression [I:483](#)

Keyboard

- data entry [I:96](#)
- focus option [I:764](#)
- shortcuts [I:9](#)

Keyboard focus [I:764](#)

- defaults [I:578](#)

Kolmogorov-Smirnov test [I:321](#)

KPSS test [II:95](#)

Kruskal-Wallis test [I:318](#)

Kullback-Leibler [II:645](#)

@kurt [I:740](#)

Kurtosis [I:308](#), [I:740](#)

- by category [I:750](#)
- moving [I:745](#), [I:747](#)

@kurtsby [I:750](#)

Kwiatkowski, Phillips, Schmidt, and Shin test [II:95](#)

L

Label [I:73](#)

- automatic update option [I:770](#)
- capitalization [I:72](#)
- series [I:331](#)

Lag

- dynamic assignment [I:133](#)

- forecasting [II:125](#)
- panel data [II:517](#)
- series [I:125](#)
- Lag exclusion test [II:349](#)
- Lag length
 - VAR [II:349](#)
- Lag structure
 - VAR [II:348](#)
- Lagged dependent variable
 - and serial correlation [II:63](#)
 - Durbin-Watson statistic [II:65](#)
- Lagged series in equation [II:7](#)
- Lagrange multiplier test [II:65](#)
- Landscape printing [I:584](#)
- Laplace distribution [I:756](#)
- Large sample test [II:141](#)
- @last [I:88](#)
- Last obs
 - row-wise [I:749](#)
 - row-wise index [I:749](#)
- Last_update
 - field in database query [I:279](#)
- Last_write
 - field in database query [I:279](#)
- @lastmax [II:520](#)
- @lastmin [II:520](#)
- Latent variable
 - binary model [II:210](#)
 - censored models [II:232](#)
 - ordered models [II:226](#)
- Lead
 - series [I:125](#)
- Least squares
 - panels [II:541](#)
 - See also* Equation.
 - See also* OLS.
- @left [I:701](#)
- @length [I:699](#)
- Less than comparison [I:640](#)
- Less than or equal to comparison [I:640](#)
- Levene test [I:320](#)
- Levin, Lin and Chu [II:105](#)
- Likelihood [II:14](#)
- Likelihood specification [II:292](#)
- Lilliefors test [I:321](#)
- Limit points [II:229](#)
- censored dependent variables [II:234](#)
- make covariance matrix [II:231](#)
- make vector [II:231](#)
- non-ascending [II:230](#)
- Limited dependent variable [II:209](#)
- Line drawing [I:527](#)
- Line graph [I:450](#)
- Linear
 - frequency conversion method [I:108](#)
- Link [I:173](#)
 - basic concepts [I:173](#)
 - breaking [I:200](#)
 - create by command [I:195](#)
 - create by copy-and-paste [I:73](#)
 - creation [I:187](#)
 - frequency conversion [I:183](#), [I:194](#)
 - match merging [I:174](#)
 - modifying [I:199](#)
 - to databases [I:264](#), [I:265](#)
 - working with [I:197](#)
- Linked equations in models [II:425](#)
- List
 - specifying equation by [II:6](#)
- Ljung-Box Q-statistic [I:326](#)
 - serial correlation test [II:65](#)
- LM test
 - ARCH [II:158](#)
 - auxiliary regression [II:155](#), [II:640](#)
 - serial correlation [II:65](#), [II:155](#)
- Load
 - workfile [I:53](#)
- Loadings [II:592](#)
- Local
 - subroutine [I:624](#)
 - variable [I:622](#)
- Local optimum [II:627](#)
- Local weighting option [I:487](#)
- LOESS [I:486](#), [I:487](#)
- @log [I:735](#)
- Log likelihood
 - average [II:214](#)
 - censored models [II:233](#)
 - exponential [II:249](#)
 - for binary models [II:210](#)
 - for regression (normal errors) [II:14](#)
 - negative binomial [II:247](#)
 - normal [II:249](#)

- ordered models [II:227](#)
- Poisson model [II:247](#)
- restricted [II:214](#)
- truncated models [II:242](#)
- Log likelihood *See* Logl.
- @log10 [I:735](#)
- Logarithm
 - arbitrary base [I:736](#)
 - base-10 [I:735](#)
 - difference [I:736](#)
 - natural [I:735](#)
- Logical expression [I:124](#)
 - in easy query [I:275](#)
- Logistic
 - distribution [I:756](#)
 - logit function [I:753](#)
- @logit [I:753](#)
- Logit models [II:212](#)
- Logl [II:283](#)
 - analytical derivatives [II:288](#)
 - derivatives [II:288](#)
 - errors [II:296](#)
 - estimation [II:290](#)
 - examples [II:298](#)
 - gradients [II:294](#)
 - limitations [II:297](#)
 - order of evaluation [II:287](#)
 - parameters [II:286](#)
 - specification [II:285](#)
 - starting values [II:291](#)
 - step size [II:289](#)
 - troubleshooting [II:296](#)
 - views [II:292](#)
- Log-normal distribution [I:756](#)
- @logx [I:736](#)
- Long name [I:72](#)
 - for series [I:331](#)
- Loop
 - exit loop [I:618](#)
 - for (control variables) [I:612](#)
 - for (scalars) [I:614](#)
 - for (string variables) [I:614](#)
 - nest [I:613](#)
 - over matrix elements [I:613](#), [I:647](#)
 - while [I:612](#), [I:616](#)
- @lower [I:702](#)
- LOWESS [I:486](#), [I:487](#)

- LR statistic [II:150](#), [II:214](#), [II:240](#), [II:242](#), [II:250](#)
 - QLR [II:255](#)
- @ltrim [I:702](#)

M

- MA specification
 - backcasting [II:80](#)
 - forecast [II:128](#)
 - in ARIMA models [II:71](#)
 - in model solution [II:438](#)
 - in two stage least squares [II:43](#)
 - terms [II:75](#)
- @makedate [I:716](#)
- @map [I:169](#)
- Marginal significance level [II:12](#), [II:141](#)
- Marquardt [II:632](#)
- Match merge
 - by date [I:179](#)
 - many-to-many [I:177](#)
 - many-to-one [I:176](#)
 - one-to-many [I:175](#)
 - panels [I:181](#)
 - using links [I:174](#)
- Match operator in database query [I:277](#)
- Mathematical functions [I:735](#)
- Matrix
 - assign values [I:628](#)
 - convert to other matrix objects [I:648](#)
 - convert to series or group [I:635](#)
 - copy [I:632](#)
 - copy submatrix [I:633](#)
 - declare [I:627](#)
 - elementwise operations [I:642](#)
 - export data [I:646](#)
 - fill with values [I:629](#)
 - import data [I:646](#)
 - objects [I:627](#)
 - views [I:645](#)
- Matrix commands and functions
 - column means [I:648](#)
 - column scaling [I:648](#)
 - column summaries [I:643](#)
 - commands [I:644](#)
 - descriptive statistics [I:643](#)
 - difference [I:644](#)
 - element [I:642](#)
 - functions [I:644](#)

- matrix algebra [I:643](#)
- missing values [I:644](#)
- utility [I:641](#)
- Matrix operators
 - addition (+) [I:639](#)
 - and loop operators [I:647](#)
 - comparison operators [I:640](#)
 - division (/) [I:640](#)
 - multiplication (*) [I:639](#)
 - negation (-) [I:639](#)
 - order of evaluation [I:638](#)
 - subtraction (-) [I:639](#)
- @mav [I:746](#)
- @mavc [I:746](#)
- @max [I:739](#)
- Maximize window [I:7](#)
- Maximum [I:739](#)
 - by category [I:750](#)
 - cumulative [I:742](#), [I:743](#)
 - moving [I:745](#), [I:747](#)
 - row-wise [I:749](#)
 - row-wise index [I:749](#)
- Maximum likelihood
 - full information [II:310](#)
 - quasi-generalized pseudo-maximum likelihood [II:254](#)
 - quasi-maximum likelihood [II:248](#)
 - See also* Logl.
 - user specified [II:283](#)
- Maximum number of observations [I:770](#)
- @maxsby [I:750](#)
- McFadden R-squared [II:214](#)
- @mcor [I:747](#)
- @mcov [I:747](#)
- @mconv [I:747](#)
- @mcovs [I:747](#)
- Mean [I:307](#), [I:739](#)
 - by category [I:750](#)
 - cumulative [I:741](#), [I:742](#)
 - equality test [I:315](#)
 - geometric [I:740](#)
 - hypothesis test of [I:311](#)
 - moving [I:744](#), [I:746](#)
 - row-wise [I:748](#)
- Mean absolute error [II:123](#)
- Mean absolute percentage error [II:124](#)
- Mean equation (GARCH) [II:189](#)
- @meansby [I:750](#)
- Measurement equation [II:384](#)
- Measurement error [II:37](#), [II:170](#)
- Median [I:307](#), [I:739](#)
 - by category [I:750](#)
 - equality test [I:317](#)
 - hypothesis test of [I:313](#)
- @median [I:739](#)
- @mediansby [I:750](#)
- Memory allocation [I:770](#)
- Memory, running out of [I:770](#)
- Menu [I:8](#), [I:70](#)
 - main [I:8](#)
 - objects [I:71](#)
- Merge [I:73](#)
 - graphs [I:68](#), [I:655](#)
 - into panel workfiles [II:528](#)
 - store option [I:263](#)
- Messages
 - suppress during program execution [I:595](#)
- Metafile
 - save graph as Windows metafile. [I:543](#)
- Micro TSP
 - opening workfiles [I:254](#)
- @mid [I:701](#)
- @min [I:739](#)
- Minimize window [I:7](#)
- Minimum [I:739](#)
 - by category [I:750](#)
 - cumulative [I:742](#), [I:743](#)
 - moving [I:745](#), [I:747](#)
 - row-wise [I:749](#)
- Minimum discrepancy [II:613](#)
- @minner [I:748](#)
- @minsby [I:750](#)
- Missing values [I:129](#), [I:644](#), [I:735](#)
 - forecasting [II:120](#)
 - handling in estimation [II:9](#)
 - in frequency conversion [I:107](#)
 - in models [II:450](#)
 - in observation graphs [I:423](#)
 - mathematical functions [I:735](#)
 - recoding [I:131](#), [I:736](#)
 - relational comparisons involving [I:129](#)
 - test [I:130](#)
- Mixed frequency graph [I:431](#)

Mixed line graph [I:454](#)

@mkurt [I:747](#)

MLE

See < > [Logl](#)

@mmax [I:747](#)

@mmmin [I:747](#)

@mnas [I:748](#)

@mobs [I:747](#)

@mod [I:735](#)

Mode [I:608](#)

Model consistent expectations [II:437](#)

Models

add factors [II:407](#), [II:432](#)

aliasing [II:409](#), [II:432](#)

binding variables [II:409](#)

block structure [II:428](#)

Broyden solution [II:634](#)

Broyden solver [II:447](#)

coefficient uncertainty [II:426](#), [II:435](#), [II:445](#)

convergence test [II:450](#)

creating [II:424](#)

definition [II:307](#)

derivatives [II:449](#)

diagnostic messages and iteration history

[II:446](#)

dynamic solution [II:441](#)

equation view [II:426](#)

excluding variables [II:431](#)

exogenous variable uncertainty [II:435](#), [II:446](#)

Fair-Taylor solution [II:436](#)

fit option for solution [II:441](#)

future values [II:435](#)

Gauss-Seidel solution [II:633](#)

Gauss-Seidel solver [II:447](#)

handling of ARMA terms [II:441](#)

initialize excluded variables [II:448](#)

inline equations [II:425](#)

intercept shift add factor [II:433](#)

linked equations [II:425](#)

MA error terms [II:438](#)

missing value handling [II:450](#)

Newton solution [II:634](#)

Newton's method [II:447](#)

overriding variables [II:409](#), [II:432](#), [II:435](#)

properties of equations [II:427](#)

roundoff of solution [II:451](#)

scenarios [II:422](#), [II:430](#)

simultaneous and recursive blocks [II:429](#)

solution methods [II:447](#)

solving [II:434](#)

solving to match target [II:451](#)

starting values [II:450](#)

static solution [II:441](#)

stochastic simulation [II:440](#)

stochastic solution [II:442](#)

text description of [II:429](#)

text keywords [II:429](#)

tracking variables [II:446](#)

updating links [II:426](#)

variable dependencies [II:428](#)

variable shift add factor [II:433](#)

variable view [II:428](#)

Modulus [I:735](#)

@month [I:722](#), [I:729](#)

Moody's Economy.com [I:296](#)

Mouse [I:6](#)

@movav [I:744](#)

@movavc [I:744](#)

@movcor [I:745](#)

@movcov [I:745](#)

@movcovp [I:745](#)

@movcovs [I:745](#)

Move window [I:7](#)

Moving statistics [I:743](#)

average [I:744](#)

average, centered [I:746](#)

centered mean [I:744](#)

correlation [I:745](#), [I:747](#)

covariance [I:745](#), [I:747](#)

functions [I:127](#), [I:743](#)

geometric mean [I:138](#)

inner product [I:746](#), [I:748](#)

kurtosis [I:747](#)

maximum [I:745](#), [I:747](#)

mean [I:746](#)

minimum [I:745](#), [I:747](#)

NAs, number of [I:745](#), [I:748](#)

observations, number of [I:745](#), [I:747](#)

skewness [I:745](#), [I:747](#)

standard deviation [I:744](#), [I:746](#)

sum [I:744](#), [I:746](#)

sum of squares [I:745](#), [I:747](#)

variance [I:744](#), [I:746](#)

@movinner [I:746](#)

@movkurt [I:745](#)

@movmax [I:745](#)
 @movmin [I:745](#)
 @movnas [I:745](#)
 @movobs [I:745](#)
 @movskew [I:745](#)
 @movstdev [I:744](#)
 @movstdevp [I:744](#)
 @movstdevs [I:744](#)
 @movsum [I:744](#), [I:746](#)
 @movsumsq [I:745](#)
 @movvar [I:744](#)
 @movvarp [I:744](#)
 @movvars [I:744](#)
 @mskew [I:747](#)
 @mstdev [I:746](#)
 @mstdevp [I:746](#)
 @mstdevs [I:746](#)
 @msumsq [I:747](#)
 Multicollinearity [II:21](#)
 Multiplication [I:740](#)
 Multiplication operator (*) [I:639](#)
 Multivariate ARCH [II:310](#)
 @mvar [I:746](#)
 @mvarp [I:746](#)
 @mvars [I:746](#)

N

NA [I:129](#)
 comparisons [I:606](#)
 forecasting [II:120](#)
 inequality comparison [I:129](#)
 recode [I:736](#)
 test [I:130](#)
 testing [I:606](#)
 NA *See also* Missing data.
 Nadaraya-Watson [I:484](#)
 Name
 object [I:71](#)
 reserved [I:72](#)
 Name field in database query [I:277](#)
 Naming objects
 spool [I:559](#)
 @nan [I:736](#)
 NAs, number of
 by category [I:750](#)

 cumulative [I:742](#), [I:743](#)
 moving [I:745](#), [I:748](#)
 row-wise [I:748](#)
 @nasby [I:750](#)
 Near singular matrix [II:21](#)
 binary models [II:216](#)
 nonlinear models [II:51](#), [II:627](#)
 polynomial distributed lag [II:24](#)
 RESET test [II:171](#)
 Nearest neighbor regression [I:485](#)
 Negation operator (-) [I:639](#)
 Negative binomial
 distribution [I:756](#)
 Negative binomial count model [II:247](#)
 @neqna [I:130](#), [I:700](#), [I:735](#)
 Newey-West [II:36](#)
 HAC covariance [II:36](#)
 truncation lag [II:36](#)
 Newton's method [II:447](#), [II:634](#)
 Newton-Raphson [II:630](#)
 @ngroups [I:751](#)
 Noninvertible MA process [II:77](#), [II:82](#)
 Nonlinear coefficient restriction
 Wald test [II:148](#)
 Nonlinear least squares [II:43](#)
 convergence criterion [II:47](#)
 forecast standard errors [II:122](#)
 iteration option [II:47](#)
 specification [II:45](#)
 starting values [II:46](#)
 two stage [II:49](#)
 two stage with AR specification [II:50](#)
 weighted [II:48](#)
 weighted two stage [II:49](#)
 with AR specification [II:48](#), [II:68](#)
 Nonnested tests [II:178](#)
 Nonstationary time series [II:87](#)
 Non-unique identifiers [I:226](#)
 Normal distribution [I:756](#)
 Normality test [I:308](#), [I:321](#), [II:155](#)
 VAR [II:350](#)
 Normalize formula [I:134](#)
 Not equal to comparison [I:640](#)
 @now [I:722](#)
 @nper [I:737](#)
 N-step forecast test [II:174](#)

Null hypothesis [II:141](#)

Number of periods

annuity [I:737](#)

Numbers

converting from strings [I:699](#)

formatting in tables [I:677](#)

relational comparison [I:124](#)

N-way table [I:395](#)

chi-square tests [I:394](#)

O

Object [I:63](#)

allow multiple untitled [I:764](#)

assignment [I:580](#)

basics [I:64](#)

closing untitled [I:764](#)

command [I:582](#)

containers [I:586](#)

copy [I:73](#), [I:588](#)

create [I:67](#)

data [I:64](#), [I:77](#)

declaration [I:581](#)

delete [I:74](#), [I:590](#)

fetch [I:591](#)

freeze [I:74](#)

icon [I:65](#)

label [I:73](#)

naming [I:72](#)

open [I:68](#)

print [I:75](#)

procedure [I:65](#)

rename [I:590](#)

sample [I:94](#)

save [I:590](#)

show [I:69](#)

store [I:75](#), [I:591](#)

type [I:65](#)

window [I:69](#)

Objects menu [I:71](#)

@obs [I:739](#)

@obsby [I:750](#)

Observation equation [II:384](#), [II:389](#)

Observation graphs [I:423](#), [I:450](#)

missing values [I:423](#)

Observation identifiers [I:243](#)

Observation scale [I:443](#)

Observations

maximum [I:770](#)

Observations, number of [I:739](#)

by category [I:750](#)

cumulative [I:742](#), [I:743](#)

moving [I:745](#), [I:747](#)

row-wise [I:748](#)

@obsid [I:731](#), [II:516](#)

@obsnum [I:727](#)

panel observation numbering [II:516](#)

@obsrange [I:727](#)

@obssmpl [I:728](#)

ODBC [I:42](#)

OLS (ordinary least squares)

adjusted R-squared [II:13](#)

coefficient standard error [II:12](#)

coefficient t-statistic [II:12](#)

coefficients [II:11](#)

standard error of regression [II:14](#)

sum of squared residuals [II:14](#)

system estimation [II:308](#), [II:334](#)

Omitted variables test [II:150](#), [II:170](#)

panel [II:562](#)

One-step forecast test [II:174](#)

One-way tabulation [I:323](#)

Open

database [I:259](#)

multiple objects [I:68](#)

object [I:68](#)

options [I:254](#)

workfile [I:53](#)

Operator [I:121](#), [I:734](#)

arithmetic [I:121](#)

conjunction (and, or) [I:124](#)

difference [I:126](#)

lag [I:125](#)

lead [I:125](#)

parentheses [I:122](#)

relational [I:124](#), [I:696](#)

Optimization algorithms

BHHH [II:632](#)

first derivative methods [II:631](#)

Gauss-Newton [II:632](#)

Goldfeld-Quandt [II:631](#)

grid search [II:633](#)

Marquardt [II:632](#)

Newton-Raphson [II:630](#)

second derivative methods [II:630](#)

- starting values [II:625](#)
- step size [II:632](#)
- Option settings
 - allow only one untitled [I:764](#)
 - backup workfiles [I:767](#)
 - date notation [I:766](#)
 - default fonts [I:763](#)
 - fonts [I:763](#)
 - frequency conversion [I:766](#)
 - keyboard focus [I:764](#)
 - print setup [I:771](#)
 - program execution mode [I:765](#)
 - series auto label [I:770](#)
 - spreadsheet view defaults [I:768](#)
 - warn on close [I:764](#)
- or [I:734](#)
- Or operator [I:89](#), [I:124](#)
- Order condition for identification [II:38](#)
- Order of evaluation [I:638](#)
 - logl [II:287](#)
- Order of stacked data [II:468](#)
- Ordered dependent variable [II:226](#)
 - error messages [II:230](#)
 - estimation [II:227](#)
 - expectation-prediction tables [II:230](#)
 - forecasting [II:231](#)
 - limit points [II:231](#)
 - log likelihood [II:227](#)
 - variable frequencies [II:230](#)
 - views [II:230](#)
- Ordinary residual
 - binary models [II:222](#)
 - censored models [II:238](#)
 - count models [II:250](#)
 - truncated models [II:243](#)
- Orientation [I:421](#)
- Orthogonal regression [I:488](#)
- Orthogonality condition [II:51](#), [II:338](#)
- @otod [I:704](#)
- Output
 - printing [I:583](#)
- Overdispersion [II:248](#), [II:256](#)
 - specification test [II:251](#)
- Overidentifying restrictions [II:55](#)
- P**
- PACE [II:583](#)
- details [II:614](#)
- Pack database [I:283](#)
- Packable space [I:260](#), [I:283](#)
- Page
 - create [I:54](#)
 - delete [I:60](#)
 - rename [I:60](#)
- Page breaks [I:572](#)
- Pairwise graphs [I:433](#)
- Panel cointegration test [II:534](#)
- Panel data
 - analysis [II:528](#)
 - balanced [I:210](#)
 - cell identifier [I:731](#), [II:515](#)
 - cointegration [II:372](#), [II:534](#)
 - convert to pool [I:240](#)
 - create workfile of [I:41](#)
 - cross-section identifiers [II:514](#)
 - cross-section summaries [II:523](#)
 - dated [I:209](#)
 - duplicate identifiers [I:208](#), [I:224](#)
 - dynamic panel data [II:545](#)
 - estimation *see* Panel estimation.
 - fixed effects test [II:566](#)
 - frequency conversion [I:184](#)
 - functions [I:731](#)
 - GMM estimation [II:545](#)
 - graphs [II:529](#)
 - group identifier [I:731](#), [II:514](#)
 - Hausman test [II:568](#)
 - identifiers [I:206](#)
 - instrumental variables estimation [II:544](#)
 - irregular [I:210](#)
 - lags [II:517](#)
 - lags and leads [II:517](#)
 - lags in [I:207](#)
 - least squares estimation [II:541](#)
 - merging data into [II:528](#)
 - nested [I:212](#)
 - period summaries [II:523](#)
 - regular [I:210](#)
 - samples in panel workfiles [II:517](#)
 - statistics [II:521](#)
 - testing [II:562](#)
 - time trend [I:732](#), [II:521](#)
 - trends [I:731](#), [II:521](#)
 - unbalanced [I:210](#)
 - undated [I:209](#)

- unit root tests [II:100](#), [II:532](#)
- within-group identifier [I:731](#), [II:516](#)
- workfile structure [I:206](#)
- Panel estimation [II:541](#)
 - examples [II:548](#)
 - GLS weights [II:543](#)
 - GMM [II:545](#)
 - GMM (example) [II:557](#)
 - GMM details [II:571](#)
 - least squares [II:541](#)
 - TSLs [II:544](#)
- Panel unit root [II:532](#)
- Panel vs. pool [II:459](#)
- Panel workfile
 - create [II:509](#)
 - dated [I:219](#)
 - display [II:511](#)
 - functions [I:731](#)
 - nested [I:212](#)
 - structure [II:509](#), [II:513](#)
 - undated [I:224](#)
 - undated with ID [I:223](#)
- Parallel analysis [II:583](#)
- Param (command) [II:46](#), [II:315](#), [II:626](#)
- Parameters
 - logl [II:286](#)
- PARCH [II:199](#)
- Pareto distribution [I:757](#)
- Parks estimator [II:501](#)
- Partial analysis [I:384](#)
- Partial autocorrelation [I:326](#), [II:72](#)
- Partial covariance analysis [I:384](#)
- Paste [I:73](#)
 - data as new workfile [I:41](#)
 - existing series [I:99](#)
 - new series [I:98](#)
- Payment amount [I:738](#)
- @pc [I:736](#)
- @pca [I:736](#)
- PcGive data [I:288](#)
- @pch [I:736](#)
- @pcha [I:737](#)
- @pchy [I:737](#)
- @pcy [I:737](#)
- PDL (polynomial distributed lag) [II:23](#), [II:122](#)
 - far end restriction [II:24](#)
 - forecast standard errors [II:122](#)
 - instrumental variables [II:25](#)
 - near end restriction [II:24](#)
 - specification [II:24](#)
- Pearson covariance [I:380](#)
- Pedroni panel cointegration test [II:374](#), [II:535](#)
- Percent change
 - one-period [I:736](#)
 - one-period annualized [I:736](#), [I:737](#)
 - one-year [I:737](#)
- Period
 - dummy variable [I:729](#)
 - summaries [II:523](#)
- Phillips-Perron test [II:95](#)
- Pi [I:754](#)
- Pie graph [I:460](#)
- @pmt [I:738](#)
- PNG [I:543](#)
- Poisson
 - distribution [I:757](#)
- Poisson count model [II:247](#)
- Polynomial distributed lags, *See* PDL.
- Pool [II:459](#)
 - ? placeholder [II:464](#)
 - and cross-section specific series [II:463](#)
 - AR specification [II:482](#)
 - balanced data [II:469](#), [II:474](#)
 - balanced sample [II:481](#)
 - base name [II:463](#)
 - coefficient test [II:494](#)
 - cointegration [II:475](#)
 - common coefficients [II:482](#)
 - convergence criterion [II:485](#)
 - convert to panel [I:246](#)
 - copy [II:462](#)
 - creating [II:465](#)
 - cross-section [II:461](#)
 - cross-section specific coefficients [II:482](#)
 - defining [II:461](#)
 - defining groups of identifiers [II:462](#)
 - descriptive statistics [II:473](#)
 - dummy variable [II:472](#)
 - editing definitions [II:462](#)
 - estimation [II:480](#)
 - estimation details [II:495](#)
 - export data [II:478](#)
 - fixed effects [II:483](#), [II:498](#)

-
- generate series [II:472](#)
 - group [II:464](#)
 - identifier comparison with “?” wildcard [I:778](#)
 - import [II:466](#)
 - import data [II:466](#)
 - import stacked data [II:469](#)
 - instrumental variables [II:485](#), [II:502](#)
 - make group [II:476](#)
 - make system [II:477](#)
 - naming series [II:463](#)
 - object [II:460](#)
 - options [II:484](#)
 - order [II:468](#)
 - period-specific coefficients [II:482](#)
 - pool series [II:464](#)
 - procedures [II:494](#)
 - random effects [II:483](#), [II:498](#)
 - residuals [II:494](#)
 - restructure [II:471](#)
 - series [II:464](#)
 - setup [II:465](#)
 - special group identity series [II:464](#)
 - specification [II:461](#)
 - stacked data [II:467](#)
 - tests [II:494](#)
 - unstacked data [II:466](#)
 - workfile [II:459](#)
 - Pool vs. panel [II:459](#)
 - Portmanteau test
 - VAR [II:350](#)
 - Portrait (print orientation) [I:584](#)
 - PostScript [I:543](#)
 - save graph as PostScript file [I:543](#)
 - Power (raise to) [I:734](#)
 - Prais-Winsten [II:70](#)
 - Precedence of evaluation [I:122](#), [I:638](#)
 - Predetermined variable [II:37](#)
 - Prediction table
 - binary models [II:217](#)
 - ordered models [II:230](#)
 - Present value [I:737](#)
 - Prewhitening [II:317](#), [II:340](#)
 - Principal components [I:397](#)
 - Principal factors [II:613](#)
 - Print
 - and display [I:583](#)
 - graphs [I:542](#)
 - landscape [I:584](#)
 - mode [I:572](#)
 - objects [I:75](#)
 - portrait [I:584](#)
 - settings [I:771](#)
 - setup options [I:771](#)
 - spool [I:572](#), [I:772](#)
 - spool by command [I:692](#)
 - table [I:684](#)
 - tables [I:552](#)
 - to a spool [I:555](#)
 - Probability response curve [II:223](#)
 - Probit models [II:211](#)
 - Procedure [I:65](#)
 - @prod [I:740](#)
 - Product [I:740](#)
 - cumulative [I:741](#), [I:742](#)
 - Program [I:593](#)
 - abort [I:596](#)
 - arguments [I:607](#), [I:608](#)
 - backup files [I:765](#)
 - basics [I:593](#)
 - break [I:596](#)
 - call subroutine [I:621](#)
 - create [I:593](#)
 - encrypt [I:594](#)
 - entering text [I:593](#)
 - errors [I:617](#)
 - example [I:596](#)
 - execution [I:595](#)
 - execution option [I:765](#)
 - exit loop [I:618](#)
 - for [I:612](#)
 - if [I:610](#)
 - if statement [I:610](#)
 - include file [I:618](#)
 - line continuation character [I:593](#)
 - modes [I:607](#)
 - multiple programs [I:618](#)
 - open [I:594](#)
 - place subroutine [I:621](#)
 - quiet mode [I:595](#)
 - running [I:595](#)
 - save [I:594](#)
 - stop [I:617](#)
 - strings [I:599](#)
 - syntax coloring [I:765](#)
 - variables [I:598](#)

- verbose mode [I:595](#)
- version 4 compatibility [I:607](#), [I:608](#)
- while [I:616](#)
- Programming *See* Program.
- @psi [I:752](#)
- @pv [I:737](#)
- P-value [II:141](#)
 - for coefficient t-statistic [II:12](#)
- P-value functions [I:757](#)

Q

- QML [II:248](#)
- QQ-plot [I:475](#), [I:477](#)
- Q-statistic
 - Ljung-Box [I:326](#)
 - residual serial correlation test [II:350](#)
 - serial correlation test [II:65](#)
- Quadratic
 - frequency conversion method [I:108](#)
- Quadratic hill-climbing [II:631](#)
- Quadratic spectral kernel [II:339](#)
- Qualitative dependent variable [II:209](#)
- Quandt breakpoint test [II:166](#)
- Quantile
 - by category [I:750](#)
- @quantile [I:740](#)
- Quantile function [I:740](#)
- Quantile method [I:473](#)
- Quantile regression [II:259](#)
- Quantiles
 - from series [I:333](#), [I:334](#)
- @quantilesby [I:750](#)
- @quarter [I:722](#), [I:729](#)
- Quasi-generalized pseudo-maximum likelihood [II:254](#)
- Quasi-likelihood ratio test [II:250](#), [II:255](#)
- Quasi-maximum likelihood [II:248](#)
 - robust standard errors [II:255](#)
- Queries on database [I:273](#)
 - advanced query [I:275](#)
 - DRI [I:300](#)
 - easy query [I:273](#)
 - examples [I:279](#)
 - logical expressions [I:275](#)
 - wildcard characters [I:274](#)
- Quiet mode [I:595](#), [I:607](#), [I:765](#)

R

- Ramsey RESET test [II:170](#)
- Random effects
 - pool [II:483](#)
 - pool descriptions [II:498](#)
 - test [II:568](#)
- Random walk [II:87](#)
- Randomize ties [II:220](#)
- @rank [I:128](#)
- Rank condition for identification [II:38](#)
- Ranking
 - series observations [I:128](#)
- Ranks
 - observations in series or vector [I:740](#)
- @ranks [I:740](#)
- @rate [I:738](#)
- Rate for annuity [I:738](#)
- Ratio to moving-average [I:353](#)
- RATS data
 - 4.x native format [I:289](#)
 - portable format [I:289](#)
- Read [II:466](#)
- Rebuild database [I:284](#)
- Reciprocal [I:735](#)
- @recode [I:736](#)
- Recode values [I:735](#), [I:736](#)
- Recursive coefficient [II:175](#)
 - save as series [II:175](#)
- Recursive estimation
 - least squares [II:171](#)
 - using state space [II:392](#)
- Recursive residual [II:171](#), [II:172](#)
 - CUSUM [II:172](#)
 - CUSUM of squares [II:173](#)
 - n-step forecast test [II:174](#)
 - one-step forecast test [II:174](#)
 - save as series [II:175](#)
- Redirect output to file [I:583](#)
- Reduced covariance [II:592](#)
- Redundant variables test [II:151](#)
 - panel [II:564](#)
- Registry [I:271](#)
- Regression
 - adjusted R-squared [II:13](#)
 - coefficient standard error [II:12](#)

- coefficients [II:11](#)
- collinearity [II:21](#)
- forecast [II:113](#)
- F-statistic [II:15](#)
- log likelihood [II:14](#)
- residuals from [II:19](#)
- See also* Equation.
- standard error of [II:14](#)
- sum of squared residuals [II:14](#)
- t-statistic for coefficient [II:12](#)
- Regression line
 - on graph [I:480](#)
- Regular data [I:203](#)
- Relational operators
 - and missing values [I:129](#)
- Remarks
 - field in database query [I:279](#)
- Removing data [I:233](#)
- Rename [I:72](#)
 - database [I:283](#)
 - object [I:590](#)
 - objects in database [I:267](#)
 - page [I:60](#)
 - workfile page [I:60](#)
- Repair database [I:284](#)
- @replace [I:702](#)
- Replacement variable [I:601](#)
 - naming objects [I:602](#)
- Representation view
 - of equation [II:17](#)
- Resample [I:337](#)
- Reserved names [I:72](#)
- RESET test [II:170](#)
- Reshaping a workfile [I:237](#)
- Residuals
 - binary models [II:222](#)
 - censored dependent variable models [II:238](#)
 - count models [II:250](#)
 - default series RESID [II:19](#)
 - equation [II:19](#)
 - from estimated equation [II:19](#)
 - from two stage least squares [II:39](#)
 - generalized [II:222](#), [II:238](#), [II:244](#), [II:251](#)
 - make series [II:18](#)
 - of equation [II:17](#)
 - ordinary [II:222](#), [II:238](#), [II:243](#), [II:250](#)
 - pool [II:494](#)
 - recursive [II:171](#), [II:172](#)
 - standardized [II:18](#), [II:222](#), [II:238](#), [II:244](#), [II:251](#)
 - sum of squares [II:14](#)
 - symmetrically trimmed [II:240](#)
 - system [II:323](#)
 - tests of [II:154](#)
 - truncated dependent variable [II:243](#)
 - unconditional [II:63](#), [II:69](#)
- Resize
 - spools [I:562](#)
 - table columns and rows [I:548](#)
 - workfile [I:215](#), [I:227](#)
- Resize window [I:7](#)
- Restore window [I:7](#)
- Restricted estimation [II:8](#)
- Restricted log likelihood [II:214](#)
- Restrictions
 - structural VAR [II:357](#)
- Restructuring [II:471](#)
- Results
 - accessing from estimated equation [II:16](#)
- @rfirst [I:749](#)
- @rfirsti [I:749](#)
- Rich Text Format [I:553](#)
 - See* RTF.
- @right [I:701](#)
- @rlast [I:749](#)
- @rlasti [I:749](#)
- @rmax [I:749](#)
- @rmaxi [I:749](#)
- @rmean [I:748](#)
- @rmin [I:749](#)
- @rmini [I:749](#)
- @rnas [I:748](#)
- @robs [I:748](#)
- Robust standard errors
 - Bollerslev-Wooldridge for GARCH [II:192](#)
 - GLM [II:256](#)
 - Huber/White (QML) [II:255](#)
- Robustness iterations [I:482](#), [I:487](#)
- Root mean squared error [II:123](#)
- Rotate
 - graphs [I:421](#)
- Rotation of factors [II:586](#)
 - details [II:617](#)

Round [I:735](#), [I:736](#)

@round [I:736](#)

Row

functions [I:141](#)

height [I:548](#), [I:678](#)

Row statistics

first non-missing obs [I:749](#)

first non-missing obs index [I:749](#)

last non-missing obs [I:749](#)

last non-missing obs index [I:749](#)

maximum [I:749](#)

maximum index [I:749](#)

mean [I:748](#)

minimum [I:749](#)

NAs, number of [I:748](#)

observations matching value, number of [I:748](#)

Observations, number of [I:748](#)

standard deviation [I:748](#), [I:749](#)

sum [I:748](#)

sum of squares [I:749](#)

variance [I:749](#)

R-squared

adjusted [II:13](#)

for regression [II:13](#)

from two stage least squares [II:40](#)

McFadden [II:214](#)

negative [II:194](#)

uncentered [II:156](#), [II:159](#)

with AR specification [II:70](#)

@rstdev [I:748](#)

@rstdevp [I:748](#)

@rstdevs [I:749](#)

@rsum [I:748](#)

@rsumsq [I:749](#)

RTF [I:553](#)

create [I:772](#)

redirecting print to [I:772](#)

@rtrim [I:702](#)

Run program

application in batch mode [I:596](#)

multiple files [I:618](#)

@rvalcount [I:748](#)

@rvar [I:749](#)

@rvarp [I:749](#)

@rvars [I:749](#)

S

Sample

adjustment in estimation [II:10](#)

@all [I:88](#)

all observations [I:88](#)

balanced [II:481](#)

change [I:87](#)

change using for loop [I:613](#)

command [I:89](#)

common [I:129](#)

current [I:47](#)

date pairs [I:87](#)

@first [I:88](#)

first observation [I:88](#)

if condition [I:88](#)

individual [I:129](#)

last observation [I:88](#)

number of observations in [I:728](#)

range pairs [I:87](#)

selection and missing values [I:89](#)

specifying sample object [I:94](#)

specifying samples in panel workfiles [II:517](#)

used in estimation [II:9](#)

using sample objects in expressions [I:94](#)

with expressions [I:90](#)

workfile [I:86](#)

Sample breaks [I:423](#)

SAR specification [II:75](#)

SARMA [II:75](#)

SAS file [I:42](#)

Save

backup workfile [I:51](#)

commands in file [I:578](#)

graphs [I:543](#)

objects to disk [I:590](#)

options [I:254](#)

save as new workfile [I:51](#)

tables [I:553](#), [I:684](#)

workfile [I:50](#)

workfile precision and compression [I:52](#)

Scalar [I:143](#)

matrix equal to [I:630](#)

Scale factor [II:237](#)

Scaling

factor analysis [II:585](#)

Scatterplot [I:457](#)

categorical [I:503](#)

- with confidence ellipse [I:489](#)
- with kernel regression fit [I:483](#)
- with nearest neighbor fit [I:485](#)
- with orthogonal regression line [I:488](#)
- with regression line [I:480](#)
- Scatterplot matrix [I:435](#)
- Scenarios in models [II:422](#)
- Schwarz criterion [II:645](#)
 - for equation [II:15](#)
- Score coefficients [II:588](#)
- Score vector [II:223](#)
- Scores [II:587](#)
- Scroll window [I:6](#)
- @seas [I:729](#)
- Seasonal
 - ARMA [II:75](#)
 - ARMA terms [II:75](#)
 - difference [I:127](#), [II:74](#)
 - dummy variable [I:729](#)
 - graphs [I:461](#)
- Seasonal adjustment [I:339](#)
 - additive [I:354](#)
 - Census X11 (historical) [I:348](#)
 - Census X12 [I:340](#)
 - multiplicative [I:353](#)
 - Tramo/Seats [I:349](#)
- Second derivative methods [II:630](#)
- Seemingly unrelated regression [II:309](#), [II:335](#)
- Select
 - all [I:68](#)
 - multiple items [I:8](#)
 - object [I:67](#)
 - single item [I:8](#)
- Sensitivity of binary prediction [II:218](#)
- Sequential LR tests [I:618](#)
- Serial correlation
 - ARIMA models [II:71](#)
 - Durbin-Watson statistic [II:14](#), [II:64](#)
 - first order [II:64](#)
 - higher order [II:64](#)
 - nonlinear models [II:68](#)
 - tests [II:64](#), [II:155](#)
 - theory [II:63](#)
 - two stage regression [II:69](#)
- Series [I:305](#)
 - auto-series [I:135](#)
 - auto-updating [I:145](#)
 - auto-updating and databases [I:148](#)
 - auto-updating and forecasting [II:132](#)
 - binning [I:333](#)
 - classification [I:333](#)
 - convert to matrix/vector [I:636](#)
 - create [I:78](#), [I:131](#)
 - cross-section specific [II:463](#)
 - delete observation [I:83](#)
 - description of [I:77](#)
 - descriptive statistics [I:306](#)
 - difference [I:126](#)
 - display format [I:79](#)
 - display type [I:79](#)
 - dynamic assignment [I:133](#)
 - edit in differences [I:368](#)
 - editing [I:82](#)
 - extract observation [I:727](#)
 - functions [I:123](#)
 - generate by command [I:134](#)
 - graph [I:306](#), [I:419](#)
 - implicit assignment [I:133](#)
 - in pool objects [II:464](#)
 - insert observation [I:83](#)
 - label [I:331](#)
 - lag [I:125](#)
 - lead [I:125](#)
 - pooled [II:464](#)
 - procs [I:332](#)
 - properties [I:331](#)
 - ranking [I:128](#)
 - resample [I:337](#)
 - setting graph axis [I:442](#)
 - smpl +/- [I:82](#)
 - spreadsheet view [I:306](#)
 - spreadsheet view defaults [I:769](#)
 - using expressions in place of [I:135](#)
- @seriesname [I:141](#)
- Shade region of graph [I:527](#)
 - by command [I:669](#)
- Shadowing of object names [I:282](#)
- Sharing violation [I:260](#)
- Show [I:68](#)
- Siegel-Tukey test [I:319](#)
- Sign test [I:313](#)
- Signal equation [II:389](#)
- Signal views [II:400](#)
- Silverman bandwidth [I:469](#)

Simultaneous equations *See* systems.

@sin [I:754](#)

Sine [I:754](#)

@skew [I:740](#)

Skewness [I:307](#), [I:740](#)

by category [I:750](#)

moving [I:745](#), [I:747](#)

@skewsbby [I:750](#)

SMA specification [II:75](#)

Smoothing

methods [I:354](#)

parameters [I:355](#)

state space [II:385](#)

Smpl command [I:89](#)

Smpl +/- [I:82](#)

Solve

Broyden [II:447](#)

Gauss-Seidel [II:633](#)

Newton-Raphson [II:630](#)

Sort

observations in a graph [I:422](#), [I:528](#)

valmaps [I:163](#)

workfiles [I:254](#)

Source

field in database query [I:279](#)

Sparse label option [I:309](#), [I:393](#)

Spearman rank-order [I:380](#)

theory [I:388](#)

Special functions [I:751](#)

Specification

by formula [II:7](#)

by list [II:6](#)

equation [II:6](#)

Specification test

for binary models [II:225](#)

for overdispersion [II:251](#)

for tobit [II:240](#)

RESET (Ramsey) [II:170](#)

White [II:159](#)

Specificity of binary prediction [II:218](#)

Specify equation [II:6](#), [II:141](#)

in systems [II:311](#)

nonlinear [II:45](#)

Spectrum estimation [II:97](#), [II:98](#)

Spike graph [I:454](#)

Spool [I:554](#)

add objects [I:688](#)

add objects by command [I:688](#)

add to [I:555](#)

appending [I:557](#)

comments [I:560](#)

comments by command [I:691](#)

copying to [I:556](#)

create [I:554](#)

create by command [I:687](#)

customization [I:566](#)

delete objects [I:563](#)

display mode [I:568](#)

display name [I:690](#)

display name by command [I:690](#)

embedding [I:557](#)

extract [I:563](#)

extract by command [I:691](#)

flatten [I:564](#)

hiding objects [I:560](#)

indentation [I:563](#)

management [I:555](#)

naming objects [I:559](#)

naming objects by command [I:690](#)

object comments [I:691](#)

object names [I:690](#)

order [I:563](#)

print [I:772](#)

print by command [I:692](#)

print size [I:573](#)

print to [I:555](#)

printing [I:572](#)

properties [I:566](#)

rearrange [I:563](#)

redirecting print to [I:772](#)

removing objects [I:691](#)

resize [I:562](#)

Spreadsheet

alpha series [I:155](#)

display type [I:79](#)

file export [I:105](#)

file import [I:100](#)

group view [I:368](#)

series [I:306](#)

sort display order [I:370](#)

view option [I:768](#)

SPSS file [I:42](#)

@sqrt [I:736](#)

sqrt [I:736](#)

- Square root [I:736](#)
- SSCP [I:382](#)
- Stability test [II:164](#)
 - Chow breakpoint [II:165](#)
 - Chow forecast [II:168](#)
 - RESET [II:170](#)
 - with unequal variance [II:176](#)
- Stacked data [II:467](#)
 - balanced [II:469](#)
 - descriptive statistics [II:474](#)
 - order [II:468](#)
- Stacking data [I:246](#)
- Standard deviation [I:307](#), [I:740](#), [I:746](#)
 - by category [I:750](#)
 - cumulative [I:741](#), [I:742](#), [I:743](#)
 - moving [I:744](#), [I:746](#)
 - row-wise [I:748](#), [I:749](#)
- Standard error
 - for estimated coefficient [II:12](#)
 - forecast [II:122](#), [II:135](#)
 - of the regression [II:14](#)
 - See also Robust standard errors.
 - VAR [II:354](#)
- Standardized residual [II:18](#)
 - binary models [II:222](#)
 - censored models [II:238](#)
 - count models [II:251](#)
 - truncated models [II:244](#)
- Start
 - field in database query [I:278](#)
- Starting values
 - (G)ARCH models [II:191](#)
 - binary models [II:215](#)
 - for ARMA estimation [II:79](#)
 - for coefficients [II:46](#), [II:625](#)
 - for nonlinear least squares [II:46](#)
 - for systems [II:315](#)
 - logl [II:291](#)
 - param statement [II:46](#), [II:626](#)
 - state space [II:393](#)
 - user supplied [II:80](#)
- Stata file [I:42](#)
- State equation [II:384](#), [II:388](#)
- State space [II:383](#)
 - estimation [II:387](#), [II:397](#)
 - filtering [II:384](#)
 - forecasting [II:386](#)
 - interpreting [II:397](#)
 - @mprior [II:393](#)
 - observation equation [II:384](#)
 - representation [II:383](#)
 - specification [II:383](#), [II:388](#)
 - specification (automatic) [II:395](#)
 - starting values [II:393](#)
 - state equation [II:384](#)
 - views [II:399](#)
 - @vprior [II:393](#)
- State variables [II:383](#)
- State views [II:400](#)
- Static forecast [II:127](#)
- Stationary time series [II:87](#)
- Statistical distribution functions [I:754](#)
- Status line [I:12](#)
- @stdev [I:740](#)
- @stdevpsby [I:750](#)
- @stdevs [I:740](#)
- @stdevsby [I:750](#)
- @stdevssby [I:750](#)
- Step size [II:632](#)
 - logl [II:289](#)
- Stepwise [II:55](#)
 - swapwise [II:60](#)
 - uni-directional [II:59](#)
- Stop program execution [I:617](#)
- Store [I:75](#)
 - as .DB? file [I:263](#)
 - from pool [II:478](#)
 - in database [I:262](#)
 - merge objects [I:263](#)
 - object in database or databank [I:591](#)
- @str [I:702](#)
- @strdate [I:704](#), [I:722](#), [I:729](#)
- String [I:599](#), [I:695](#)
 - as replacement variable [I:601](#)
 - assign to table cell [I:676](#)
 - change case [I:702](#)
 - comparison [I:697](#)
 - concatenation [I:696](#)
 - convert date string to observation [I:704](#)
 - convert date value to string [I:715](#)
 - convert number to string [I:702](#)
 - convert to a scalar [I:600](#)
 - convert to date number [I:701](#), [I:715](#)
 - convert to number [I:700](#)

- convert to numbers [I:699](#)
- current date as a string [I:704](#)
- extract portion [I:701](#)
- find substring [I:700](#)
- for loops [I:614](#)
- functions [I:699](#)
- in for loop [I:614](#)
- insert string into another string [I:702](#)
- length [I:699](#)
- missing value [I:695](#), [I:698](#)
- null string [I:695](#)
- operators [I:696](#)
- ordering [I:697](#)
- program arguments [I:607](#), [I:608](#)
- relational comparison [I:696](#)
- relational comparisons with empty strings [I:698](#)
- replace portion of string with new string [I:702](#)
- test for blank string [I:700](#)
- test two strings for equality [I:700](#)
- test two strings for inequality [I:700](#)
- trim spaces [I:702](#)
- String functions [I:758](#)
- @strnow [I:704](#)
- Structural change [II:164](#)
- Structural forecast [II:127](#)
- Structural solution of models [II:441](#)
- Structural VAR [II:357](#)
 - estimation [II:362](#)
 - factorization matrix [II:351](#)
 - identification [II:361](#)
 - long-run restrictions [II:359](#)
 - short-run restrictions [II:357](#)
- Structuring a workflow [I:203](#)
- Subroutine [I:619](#)
 - arguments [I:620](#)
 - call [I:621](#)
 - define [I:619](#)
 - global [I:622](#)
 - local [I:624](#)
 - placement [I:621](#)
 - return from [I:619](#)
- Subtraction operator (-) [I:639](#)
- Sum [I:740](#)
 - by category [I:750](#)
 - cumulative [I:741](#), [I:742](#)
 - moving [I:744](#), [I:746](#)
 - row-wise [I:748](#)
- @sum [I:740](#)
- Sum of squared residuals
 - for regression [II:14](#)
- Sum of squares [I:740](#)
 - by category [I:750](#)
 - cumulative [I:742](#), [I:743](#)
 - moving [I:745](#), [I:747](#)
 - row-wise [I:749](#)
- Summarizing data [I:370](#)
- Summary statistics
 - for regression variables [II:13](#)
- @sumsby [I:750](#)
- @sumsq [I:740](#)
- @sumsqby [I:750](#)
- SUR [II:309](#), [II:335](#)
- Survivor function [I:474](#)
 - log [I:474](#)
- Swapwise [II:60](#)
- Symbol graph [I:450](#)
- Symmetrically trimmed residuals [II:240](#)
- Syntax coloring [I:765](#)
- System [II:307](#)
 - ARCH [II:310](#)
 - covariance matrix [II:322](#)
 - create [II:310](#)
 - cross-equation weighting [II:308](#)
 - definition [II:307](#)
 - derivatives [II:322](#)
 - estimation [II:308](#), [II:315](#)
 - estimation methods (technical) [II:333](#)
 - forecast [II:323](#)
 - full information maximum likelihood [II:310](#)
 - GMM [II:337](#)
 - gradients [II:322](#)
 - Instruments [II:312](#)
 - make system from group [I:412](#)
 - OLS [II:308](#), [II:334](#)
 - options [II:318](#)
 - residuals [II:323](#)
 - specification [II:311](#)
 - specify from VAR [II:356](#)
 - SUR [II:309](#), [II:335](#)
 - three stage least squares [II:309](#), [II:337](#)
 - two stage least squares [II:309](#), [II:336](#)
 - views [II:321](#)
 - weighted least squares [II:309](#), [II:334](#)
- System options [I:770](#)

T

Table [I:545](#)

- assigning numbers to cells [I:676](#)
- assigning strings to cells [I:676](#)
- assignment with formatting [I:677](#)
- background color [I:682](#)
- borders and lines [I:682](#)
- cell annotation [I:550](#)
- cell format [I:680](#)
- cell formatting [I:548](#)
- cell merging [I:551](#)
- color [I:549](#)
- column resize [I:548](#)
- column width [I:548](#), [I:678](#)
- comments [I:550](#)
- copy [I:552](#)
- copy to other windows programs [I:552](#)
- creating [I:675](#)
- creating by freezing a view [I:675](#)
- customization [I:547](#), [I:685](#)
- decimal format code [I:677](#)
- declare [I:676](#)
- edit [I:547](#)
- editing [I:547](#)
- export [I:684](#)
- font [I:549](#)
- font characteristics [I:682](#)
- fonts [I:549](#)
- formatting [I:548](#)
- gridlines [I:548](#)
- justification and indentation [I:681](#)
- justification code [I:677](#)
- merging [I:551](#)
- paste as unformatted text [I:553](#)
- print [I:552](#), [I:684](#)
- row height [I:678](#)
- row resize [I:548](#)
- save [I:553](#), [I:684](#)
- save to file [I:553](#)
- selecting cells [I:545](#)
- text color [I:682](#)
- title [I:548](#)
- write to file [I:684](#)

Tabulation

- n-way [I:392](#)
- one-way [I:323](#)

@tan [I:754](#)

Tangent [I:754](#)

TARCH [II:198](#)

t-distribution [I:757](#)

Template [I:536](#)

- by command [I:669](#)

Test

- for equality of values [I:735](#)
- for inequality of values [I:735](#)
- for missing value [I:735](#)

Tests

- ARCH [II:158](#)
- breakpoint [II:165](#), [II:166](#)
- coefficient [II:142](#)
- heteroskedasticity [II:156](#)
- pooled [II:494](#)
- RESET [II:170](#)
- residual [II:154](#)
- See also* Hypothesis tests, Specification test and Goodness of fit
- stability tests [II:164](#)
- White [II:158](#)

Text (object) [I:554](#)

Text file

- open as workfile [I:42](#)

Theil inequality coefficient [II:124](#)

Theoretical distribution graph [I:471](#)

Three stage least squares [II:309](#), [II:337](#)

Threshold GARCH (TARCH) [II:198](#)

Time series functions [I:125](#), [I:736](#), [I:737](#)

Title bar [I:10](#), [I:47](#), [I:70](#)

To (lag range) [II:7](#)

Tobit [II:233](#)

Toggling [I:9](#)

Toolbar [I:47](#), [I:70](#)

Tracking model variables [II:446](#)

Tramo/Seats [I:349](#)

Transition equation [II:384](#)

Transpose [I:368](#)

Trend

- panel data [II:521](#)
- panel functions [I:731](#)
- See also* @trend< >.

@trend [I:730](#), [I:732](#)

- in panel [I:732](#)

@trendc [I:730](#), [I:732](#)

@trigamma [I:754](#)

Trigonometric functions [I:754](#)

@trim [I:702](#)

Truncated dependent variable [II:242](#)

estimation [II:242](#)

fitted index [II:244](#)

forecasting [II:244](#)

log likelihood [II:242](#)

residuals [II:243](#)

Truncation point [II:243](#)

TSD data format [I:288](#)

TSP portable data format [I:290](#)

t-statistics

from regression [II:12](#)

Tukey [I:473](#)

Two stage least squares [II:43](#)

in systems [II:309](#)

nonlinear [II:49](#)

nonlinear with AR specification [II:50](#)

order condition [II:38](#)

panels [II:544](#)

rank condition [II:38](#)

residuals [II:39](#)

system estimation [II:336](#)

weighted [II:40](#)

weighted in systems [II:309](#), [II:336](#)

weighted nonlinear [II:49](#)

with AR specification [II:40](#), [II:69](#)

with MA specification [II:43](#)

Type

field in database query [I:277](#)

U

Unconditional residual [II:69](#)

Uni-directional [II:59](#)

Uniform distribution [I:757](#)

Unit root test [I:327](#), [II:88](#)

augmented Dickey-Fuller [II:92](#)

Dickey-Fuller [II:92](#)

Dickey-Fuller GLS detrended [II:94](#)

Elliot, Rothenberg, and Stock [II:96](#)

KPSS [II:95](#)

panel data [II:100](#), [II:532](#)

Phillips-Perron [II:95](#)

pooled data [II:475](#)

trend assumption [II:93](#)

Units

field in database query [I:279](#)

@unmap [I:169](#)

@unmaptxt [I:170](#)

Unstacked data [II:466](#)

Unstacking data [I:240](#)

Unstacking identifiers [I:242](#)

Untitled [I:71](#), [I:72](#)

Untitled objects [I:585](#)

Update

automatic [I:145](#)

coefficient vector [II:19](#), [II:626](#)

from Database [I:76](#)

group [I:368](#)

@upper [I:702](#)

Urzua factorization matrix [II:351](#)

User supplied starting values [II:80](#)

V

@val [I:700](#)

Valmap [I:159](#)

cautions [I:170](#)

find label for value [I:169](#)

find numeric value for label [I:169](#)

find string value for label [I:170](#)

functions [I:169](#), [I:762](#)

properties [I:164](#)

sorting [I:163](#)

Value map *See* Valmap.

Van der Waerden [I:473](#)

Van der Waerden test [I:313](#), [I:318](#)

VAR

AR roots [II:348](#)

autocorrelation LM test [II:350](#)

autocorrelation test [II:350](#)

coefficients [II:380](#)

cointegration [II:363](#)

correlograms [II:349](#)

decomposition [II:355](#)

estimation [II:346](#)

estimation output [II:346](#)

factorization matrix in normality test [II:350](#)

forecasting [II:379](#)

Granger causality test [II:348](#)

impulse response [II:353](#)

Jarque-Bera normality test [II:350](#)

lag exclusion test [II:349](#)

lag length [II:349](#)

lag length choice [II:349](#)

- lag structure [II:348](#)
 - mathematical model [II:345](#)
 - response standard errors [II:354](#)
 - restrictions [II:357](#)
 - See also* Impulse response, Structural VAR.
 - @var [I:740](#)
 - Variable
 - program [I:598](#)
 - Variance [I:740](#)
 - by category [I:750](#)
 - cumulative [I:741](#), [I:742](#), [I:743](#)
 - equality test [I:319](#)
 - hypothesis test of [I:312](#)
 - moving [I:744](#), [I:746](#)
 - row-wise [I:749](#)
 - Variance decomposition [II:355](#)
 - Variance Equation (GARCH) [II:189](#)
 - Variance factor [II:256](#)
 - Variance proportion [II:124](#)
 - @varp [I:740](#)
 - @varpsby [I:750](#)
 - @vars [I:740](#)
 - @varsby [I:750](#)
 - @varssby [I:750](#)
 - VEC [II:377](#)
 - estimating [II:378](#)
 - Vector autoregression *See* VAR.
 - Vector Error Correction *See* VEC.
 - Verbose mode [I:595](#), [I:607](#), [I:765](#)
 - View
 - default [I:68](#)
 - Volatility [II:186](#)
- W**
- Wald test [II:145](#)
 - coefficient restriction [II:145](#)
 - demonstration [I:26](#)
 - formula [II:149](#)
 - F-statistic [II:149](#)
 - joint restriction [II:147](#)
 - nonlinear restriction [II:148](#)
 - structural change with unequal variance [II:176](#)
 - Warning on close option [I:764](#)
 - Watson test [I:321](#)
 - @weekday [I:722](#), [I:729](#)
 - Weibull distribution [I:757](#)
 - Weighted least squares [II:32](#), [II:35](#)
 - cross-equation weighting [II:308](#)
 - nonlinear [II:48](#)
 - nonlinear two stage [II:49](#)
 - pool [II:483](#)
 - system estimation [II:334](#)
 - two stage [II:40](#)
 - two stage in systems [II:309](#), [II:336](#)
 - Weighting matrix
 - for GMM [II:52](#), [II:338](#)
 - heteroskedasticity and autocorrelation consistent (HAC) [II:338](#)
 - kernel options [II:339](#)
 - White heteroskedasticity consistent [II:338](#)
 - While loop [I:616](#)
 - abort [I:616](#)
 - exit loop [I:618](#)
 - White heteroskedasticity consistent covariance matrix [II:35](#), [II:338](#)
 - White heteroskedasticity test [II:158](#)
 - VAR [II:352](#)
 - Width of table column [I:548](#)
 - Wilcoxon signed ranks test [I:313](#), [I:317](#)
 - Wilcoxon test [I:313](#)
 - Wildcard characters [I:48](#), [I:589](#), [I:775](#)
 - in easy query [I:274](#)
 - Window
 - active [I:6](#), [I:69](#)
 - database [I:259](#)
 - EViews main [I:9](#)
 - object [I:71](#)
 - scrolling [I:6](#)
 - size control [I:8](#)
 - Windows metafile [I:543](#)
 - Within deviations [II:521](#), [II:528](#)
 - Within factors [I:511](#)
 - identification [I:515](#)
 - WMF [I:543](#)
 - Work area [I:12](#)
 - Workfile [I:586](#)
 - append to [I:230](#)
 - applying structure to [I:213](#)
 - automatic backup [I:767](#)
 - close [I:587](#)
 - common structure errors [I:225](#)
 - contract [I:233](#)
 - copy from [I:233](#)

creation [I:38](#)
date strings [I:704](#), [I:722](#), [I:729](#)
description of [I:37](#)
directory [I:47](#)
end date of observation interval [I:728](#)
export [I:254](#)
frequency [I:586](#)
@-functions [I:760](#)
functions [I:727](#)
information [I:727](#)
load existing from disk [I:53](#)
multi-page [I:53](#)
number of observations in [I:727](#)
observation date functions [I:722](#), [I:729](#)
observation number [I:727](#)
open existing [I:587](#)
panel [II:509](#)
period indicators [I:729](#)
pool [II:459](#), [II:471](#)
remove structure [I:227](#)
reshape [I:237](#)
resize [I:227](#)
resizing [I:215](#)
sample [I:86](#)
save [I:50](#), [I:587](#)
seasonal indicators [I:729](#)
sorting [I:254](#)
stacking [I:246](#)
start date of observation interval [I:728](#)
statistics [I:50](#)
storage defaults [I:767](#)
storage precision and compression [I:767](#)
structure settings [I:214](#)
structuring [I:203](#)
summary view [I:50](#)
time trend [I:730](#), [I:732](#)
time trend (calendar based) [I:730](#)
undated [I:205](#)
unstacking [I:240](#)
window [I:46](#)

Write [II:478](#)

X

X11 (historical) [I:348](#)
 limitations [I:348](#)
X11 using X12 [I:341](#)
X12 [I:340](#)
XY area graph [I:458](#)

XY bar graph [I:459](#)
XY line graph [I:458](#)

Y

Yates' continuity correction [I:318](#)
@year [I:722](#), [I:729](#)