EViews® 12

User's Guide II

IHS Markit

EViews 12 User's Guide II

IHS Markit™

EViews 12 User's Guide II

Copyright © 1994–2020 IHS Global Inc.
All Rights Reserved

ISBN: 978-1-880411-64-3

This software product, including program code and manual, is copyrighted, and all rights are reserved by IHS Global Inc. 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 IHS Global Inc.

Disclaimer

The authors and IHS Global Inc. 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

EViews® is a registered trademark of IHS Global Inc. Windows, Excel, PowerPoint, and Access are registered trademarks of Microsoft Corporation. PostScript is a trademark of Adobe Corporation. Bloomberg is a trademark of Bloomberg Finance L.P. All other product names mentioned in this manual may be trademarks or registered trademarks of their respective companies.

Third Party Licenses

This section contains third party notices or additional terms and conditions applicable to certain software technologies which may be used in one or more EViews products and/or services. Please be sure to consult the individual product files, about box, and/or install or manual documentation for specific copyright notices and author attributions. Notices on this page are current for EViews products released on or after October 1, 2017.

- diff template Library - Copyright © 2015 Tatsuhiko Kubo cubicdaiya@gmail.com. All rights reserved.
- FFmpeg is a trademark of Fabrice Bellard, originator of the FFmpeg project.
- GZipHelper - Copyright © 1995-2002 Gao Dasheng dsgao@hotmail.com.
- jsonCPP Library - Copyright © 2007-2010 Baptiste Lepilleur and The JsonCPP Authors.
- openssl Library - Copyright © 1998-2016 The OpenSSL Project. All rights reserved.
- libcurl Library - Copyright © 1996-2013, Daniel Stenberg daniel@haxx.se.
- libharu Library - Copyright © 2000-2006 Takeshi Kanno, Copyright © 2007-2009 Antony Dovgal et all.

- libssh2 Library - Copyright © 2004-2007 Sara Golemon sarag@libssh2.org, Copyright © 2005, 2006 Mikhail Gusarov dottedmag@dottedmag.net, Copyright © 2006-2007 The Written Word, Inc., Copyright © 2007 Eli Fant elifantu@mail.ru, Copyright © 2009 Daniel Stenberg, Copyright © 2008, 2009 Simon Josefsson. All rights reserved.
- ssleay License - Copyright © 1995-1998 Eric Young (eay@cryptsoft.com) All rights reserved.
- Tableau Data Extract API - Copyright © 2003-2017 Tableau and its licensors. All rights reserved.
- Tramo/Seats - Copyright (c) 1996 Agustin Maravall and Victor Gomez. Windows version developed by G. Caporello and A. Maravall (Bank of Spain)
- X11.2 and X12-ARIMA version 0.2.7 and X-13ARIMA-SEATS - Copyright (c) U.S. Census Bureau.
- zlib Data Compression Library - Copyright © 1995-2017 Jean-loup Gailly and Mark Adler.

Notices, terms and conditions pertaining to third party software are located at http://www.eviews.com/thirdparty and incorporated by reference herein.Notices, terms and conditions pertaining to third party software are located at http://www.eviews.com/thirdparty and incorporated by reference herein.

IHS Markit - EViews
3030 Old Ranch Parkway, Suite 260
Seal Beach, CA, 90740
Telephone: (949) 856-3368
Fax: (949) 856-2044
e-mail: sales@eviews.com
web: www.eviews.com

November 10, 2020

Table of Contents

EVIEWS 12 USER'S GUIDE I

PREFACE 1

PART I. EVIEWS FUNDAMENTALS 3

CHAPTER 1. INTRODUCTION 5

- What is EViews? 5
- The EViews Window 5
- Custom Edit Fields in EViews 10
- Breaking or Canceling in EViews 14
- Closing EViews 14
- Where to Go For Help 14
- EViews Updates 15

CHAPTER 2. A DEMONSTRATION 17

- Getting Data into EViews 17
- Examining the Data 20
- Estimating a Regression Model 27
- Specification and Hypothesis Tests 30
- Modifying the Equation 31
- Forecasting from an Estimated Equation 34
- Additional Testing 39

CHAPTER 3. WORKFILE BASICS 41

- What is a Workfile? 41
- Creating a Workfile 42
- The Workfile Window 60
- Saving a Workfile 75
- Loading a Workfile 81
- Multi-page Workfiles 82
- Comparing Workfiles 92
- Addendum 94

CHAPTER 4. OBJECT BASICS 101

- What is an Object? 101

Basic Object Operations 105
The Object Window 112
Working with Objects 115

CHAPTER 5. BASIC DATA HANDLING 123
Data Objects 123
Samples 136
Sample Objects 144
Importing Data 146
Exporting Data 161
Reading EViews Data using Other Applications 164
Frequency Conversion 170
Value-Based Text and Fill Coloring 178
References 186

CHAPTER 6. WORKING WITH DATA 187
Numeric Expressions 187
Series 197
Auto-series 201
Groups 206
Scalars 210

CHAPTER 7. WORKING WITH DATA (ADVANCED) 211
Auto-Updating Series 211
Alpha Series 216
Date Series 224
Value Maps 227

CHAPTER 8. SERIES LINKS 241
Basic Link Concepts 241
Creating a Link 255
Working with Links 265

CHAPTER 9. ADVANCED WORKFILES 271
Structuring a Workfile 271
Resizing a Workfile 296
Appending to a Workfile 299
Contracting a Workfile 302
Copying from a Workfile 302

Reshaping a Workfile 306
Sorting a Workfile 323
Exporting from a Workfile 323
References 324

CHAPTER 10. EVIEWS DATABASES 325

Database Overview 325
Database Basics 326
Working with Objects in Databases 330
Database Auto-Series 337
The Database Registry 339
Querying the Database 341
Object Aliases and Illegal Names 349
Maintaining the Database 351
Foreign Format Databases 353
Working with DRIPro Links 430

PART II. BASIC DATA ANALYSIS 437

CHAPTER 11. SERIES 439

Series Views Overview 439
Spreadsheet 440
Graph 440
Descriptive Statistics & Tests 440
One-Way Tabulation 458
Duplicates Analysis 458
Correlogram 462
Long-run Variance 465
Unit Root Test 465
Breakpoint Unit Root Test 465
Variance Ratio Test 465
BDS Independence Test 466
Forecast Evaluation 466
Wavelet Analysis 473
Label 474
Properties 474
Series Adjust 475
Series Procs Overview 477

Generate by Equation 477
Generate by Classification 477
Resample 482
Interpolate 484
Seasonal Adjustment 487
Automatic ARIMA Forecasting 538
Forecast Averaging 547
Exponential Smoothing 553
Hodrick-Prescott Filter 580
Frequency (Band-Pass) Filter 581
Whiten Data 585
Distribution Plot Data 585
Wavelet Objects 586
References 588

CHAPTER 12. GROUPS 591

Group Views Overview 591
Group Members 591
Spreadsheet 592
Dated Data Table 598
Graph 616
Descriptive Statistics 616
Covariance Analysis 616
N-Way Tabulation 629
Tests of Equality 633
Principal Components 634
Correlograms 654
Cross Correlations and Correlograms 655
Long-run Covariance 655
Unit Root Tests 661
Cointegration Test 661
Granger Causality 661
Label 663
Group Procedures Overview 663
References 667

CHAPTER 13. GEOMAPS 669

CHAPTER 14. GRAPHING DATA 679

Quick Start 680
Graphing a Series 683
Graphing Multiple Series (Groups) 692
The Graph Sample 703
Graph Pan and Zoom 706
Multi-graph Slideshow 708
Animated Graphs 710
Basic Customization 724
Graph Types 745
References 790

CHAPTER 15. CATEGORICAL GRAPHS 793

Illustrative Examples 793
Specifying Factors 810

PART III. CUSTOMIZING OUTPUT 825

CHAPTER 16. GRAPH OBJECTS 827

Creating Graph Objects 827
Combining Graphs 832
Customizing Graphs 832
Modifying Multiple Graphs 859
Printing Graphs 862
Saving Graphs to a File 863
Graph Commands 864

CHAPTER 17. TABLE AND TEXT OBJECTS 865

Table Objects 865
Text Objects 876

CHAPTER 18. SPOOL OBJECTS 877

Creating a Spool 877
Managing the Spool 878
Customizing the Spool 889
Editing Objects in a Spool 892
Printing a Spool 895

Saving a Spool 896

PART IV. EXTENDING EVIEWS 899

Create Your Own Estimator 899

Add Menu Items and Object Classes 899

Connect with External Applications 900

CHAPTER 19. OBJECT LINKING AND EMBEDDING (OLE) 903

Embedding vs. Linking 904

Using OLE 904

Opening Documents with Linked Objects 931

APPENDIX A. GLOBAL OPTIONS 935

The Options Menu 935

Print Setup 956

APPENDIX B. ENHANCED SPREADSHEET EDITING 961

Array Expressions 961

EVIEWS 12 USER'S GUIDE II

PREFACE 1

PART V. BASIC SINGLE EQUATION ANALYSIS 3

CHAPTER 20. BASIC REGRESSION ANALYSIS 5

Equation Objects 5

Specifying an Equation in EViews 6

Estimating an Equation in EViews 9

Equation Output 11

Working with Equations 18

Estimation Problems 22

References 22

CHAPTER 21. ADDITIONAL REGRESSION TOOLS 23

Special Equation Expressions 23

Robust Standard Errors 32

Weighted Least Squares 47

Nonlinear Least Squares 51

Indicator Saturation 60

References 63

CHAPTER 22. REGRESSION VARIABLE SELECTION 65

Background 65
Variable Selection Estimation in EViews 70
Working with an Estimated Equation 76
Example 77
References 87

CHAPTER 23. INSTRUMENTAL VARIABLES AND GMM 89

Background 89
Two-stage Least Squares 89
Nonlinear Two-stage Least Squares 96
Limited Information Maximum Likelihood and K-Class Estimation 97
Generalized Method of Moments 101
IV Diagnostics and Tests 112
References 117

CHAPTER 24. TIME SERIES REGRESSION 119

Background 119
Testing for Serial Correlation 127
Estimating ARIMA and ARFIMA Models in EViews 130
Estimation Output 144
Equation Diagnostics 147
Examples 153
Additional Topics 158
Estimation Method Details 160
References 165

CHAPTER 25. FORECASTING FROM AN EQUATION 167

Forecasting from Equations in EViews 167
An Illustration 170
Forecast Basics 174
Forecasts with Lagged Dependent Variables 180
Forecasting with ARMA Errors 182
Forecasting from Equations with Expressions 187
Forecasting with Nonlinear and PDL Specifications 193
References 194

CHAPTER 26. SPECIFICATION AND DIAGNOSTIC TESTS 195
Background 195
Coefficient Diagnostics 196
Residual Diagnostics 213
Stability Diagnostics 225
Applications 254
References 259

PART VI. ADVANCED SINGLE EQUATION ANALYSIS 261

CHAPTER 27. ARCH AND GARCH ESTIMATION 263
Basic ARCH Specifications 263
Estimating ARCH Models in EViews 266
Working with ARCH Models 273
Additional ARCH Models 276
Examples 286
References 290

CHAPTER 28. COINTEGRATING REGRESSION 293
Background 293
Estimating a Cointegrating Regression 295
Testing for Cointegration 308
Working with an Equation 317
References 319

CHAPTER 29. AUTOREGRESSIVE DISTRIBUTED LAG (ARDL) MODELS 321
Background 321
Estimating ARDL Models in EViews 327
An Example 331
References 338

CHAPTER 30. MIDAS REGRESSION 341
Background 341
MIDAS Estimation in EViews 346
An Example 353
References 360

CHAPTER 31. DISCRETE AND LIMITED DEPENDENT VARIABLE MODELS 361
Binary Dependent Variable Models 361

Ordered Dependent Variable Models . . . 380
Censored Regression Models . . . 387
Truncated Regression Models . . . 397
Heckman Selection Model . . . 401
Count Models . . . 407
Technical Notes . . . 417
References . . . 419

CHAPTER 32. GENERALIZED LINEAR MODELS . . . 421
Overview . . . 421
How to Estimate a GLM in EViews . . . 423
Examples . . . 427
Working with a GLM Equation . . . 435
Technical Details . . . 439
References . . . 450

CHAPTER 33. ROBUST LEAST SQUARES . . . 451
Background . . . 451
Estimating Robust Regression in EViews . . . 459
An Illustration . . . 464
References . . . 470

CHAPTER 34. LEAST SQUARES WITH BREAKPOINTS . . . 471
Background . . . 471
Estimating Least Squares with Breakpoints in EViews . . . 473
Working with Breakpoint Equations . . . 476
Example . . . 483
References . . . 489

CHAPTER 35. DISCRETE THRESHOLD REGRESSION . . . 491
Background . . . 491
Threshold Estimation in EViews . . . 493
Working with Threshold Equations . . . 495
References . . . 505

CHAPTER 36. SMOOTH TRANSITION REGRESSION . . . 507
Background . . . 507
Estimating a Smooth Transition Regression in EViews . . . 510
Working with Smooth Threshold Equations . . . 513

Example 526
References 533

CHAPTER 37. ELASTIC NET AND LASSO 535
Background 535
Estimating an Elastic Net Regression in EViews 538
Working with Elastic Net Equations 541
Example 552
References 556

CHAPTER 38. FUNCTIONAL COEFFICIENT REGRESSION 557
Background 557
Estimating Functional Coefficients Models in EViews 562
Post-Estimation Views and Procs 566
References 581

CHAPTER 39. SWITCHING REGRESSION 583
Background 583
Estimating Switching Regressions in EViews 591
Estimation Output 597
Switching Views 599
Switching VAR Procs 602
Examples 603
References 617

CHAPTER 40. QUANTILE REGRESSION 619
Estimating Quantile Regression in EViews 619
Views and Procedures 624
Background 631
References 641

CHAPTER 41. THE LOG LIKELIHOOD (LOGL) OBJECT 643
Overview 643
Specification 645
Estimation 650
LogL Views 653
LogL Procs 653
Troubleshooting 656
Limitations 657

Examples 658
References 664

PART VII. ADVANCED UNIVARIATE ANALYSIS 665

CHAPTER 42. UNIVARIATE TIME SERIES ANALYSIS 667

Unit Root Testing 667
Unit Root Tests with a Breakpoint 679
Seasonal Unit Root Testing 695
Cross-sectionally Independent Panel Unit Root Testing 705
Cross-sectionally Dependent Panel Unit Root Tests 716
Variance Ratio Test 739
BDS Independence Test 749
Wavelet Analysis 753
References 783

PART VIII. MULTIPLE EQUATION ANALYSIS 789

CHAPTER 43. SYSTEM ESTIMATION 791

Background 791
System Estimation Methods 792
How to Create and Specify a System 794
Working With Systems 807
Technical Discussion 820
References 831

CHAPTER 44. VECTOR AUTOREGRESSION AND ERROR CORRECTION MODELS 833

Vector Autoregressions (VARs) 833
VARs With Linear Constraints 839
Views and Procs of a VAR 846
Structural (Identified) VARs 869
Vector Error Correction (VEC) Models 881
Technical Discussion 887
References 892

CHAPTER 45. BAYESIAN VAR MODELS 895

Estimating a Bayesian VAR in EViews 895
Post-Estimation Procedures 898
Examples 902

Technical Background 912
References 924

CHAPTER 46. MIXED FREQUENCY VAR 927

Estimating a Mixed Frequency VAR in EViews 927
An Example 930
Technical Background 934
References 937

CHAPTER 47. SWITCHING VAR 939

Background 939
Specifying a Switching VAR in EViews 947
Post-Estimation Results 952
Examples 956
References 967

CHAPTER 48. STATE SPACE MODELS AND THE KALMAN FILTER 969

Background 969
Specifying a State Space Model in EViews 974
Working with the State Space 987
Converting from Version 3 Sspace 992
Technical Discussion 993
References 993

CHAPTER 49. MODELS 995

Overview 995
An Example Model 998
Building a Model 1013
Working with the Model Structure 1015
Specifying Scenarios 1024
Using Add Factors 1028
Locking the Model 1030
Solving the Model 1030
Working with the Model Data 1051
Comparing Solution Data 1055
References 1057

PART IX. PANEL AND POOLED DATA 1059

CHAPTER 50. POOLED TIME SERIES, CROSS-SECTION DATA 1061

The Pool Workfile 1061
The Pool Object 1062
Pooled Data 1065
Setting up a Pool Workfile 1067
Working with Pooled Data 1074
Pooled Estimation 1082
References 1108

CHAPTER 51. WORKING WITH PANEL DATA 1111

Structuring a Panel Workfile 1111
Panel Workfile Display 1114
Panel Workfile Information 1115
Working with Panel Data 1119
Panel Data Analysis 1133
References 1133

CHAPTER 52. PANEL ESTIMATION 1135

Estimating a Panel Equation 1135
Panel Estimation Examples 1146
Panel Equation Testing 1162
Estimation Background 1184
References 1189

CHAPTER 53. PANEL COINTEGRATION ESTIMATION 1191

Background 1191
Estimating Panel Cointegration in EViews 1192
Working With a Panel Cointegrating Equation 1198
Examples 1199
Technical Details 1205
References 1210

CHAPTER 54. PANEL STATISTICS 1213

By-Statistics 1216
Panel Covariances 1219
Panel Principal Components 1224

Panel Causality Testing 1230
Panel Long-run Variances 1232
Panel Unit Root Testing 1234
Panel Cointegration Testing 1238
Panel Cross-section Dependence Test 1239
Panel Resampling 1240
Panel Stacked Analysis 1240
References 1241

PART X. ADVANCED MULTIVARIATE ANALYSIS 1243

CHAPTER 55. COINTEGRATION TESTING 1245
Johansen Cointegration Test 1245
Single-Equation Cointegration Tests 1254
Panel Cointegration Testing 1258
References 1263

CHAPTER 56. FACTOR ANALYSIS 1265
Creating a Factor Object 1266
Rotating Factors 1273
Estimating Scores 1274
Factor Views 1277
Factor Procedures 1281
Factor Data Members 1282
An Example 1282
Background 1298
References 1311

APPENDIX C. ESTIMATION AND SOLUTION OPTIONS 1313
Setting Estimation Options 1313
Optimization Algorithms 1319
Nonlinear Equation Solution Methods 1322
References 1324

APPENDIX D. GRADIENTS AND DERIVATIVES 1327
Gradients 1327
Derivatives 1330
References 1334

APPENDIX E. INFORMATION CRITERIA 1335
Definitions 1335
Using Information Criteria as a Guide to Model Selection 1337
References 1337
APPENDIX F. LONG-RUN COVARIANCE ESTIMATION 1339
Technical Discussion 1339
Kernel Function Properties 1349
References 1350
INDEX 1351

Preface

The first volume of the EViews *User's Guide* describes the basics of using EViews and describes a number of tools for basic statistical analysis using series and group objects.

The second volume of the EViews *User's Guide*, offers a description of EViews' interactive tools for advanced statistical and econometric analysis. The material in *User's Guide II* may be divided into several parts:

- Part V. "Basic Single Equation Analysis" on page 3 discusses the use of the equation object to perform standard regression analysis, ordinary least squares, weighted least squares, nonlinear least squares, basic time series regression, specification testing and forecasting.
- Part VI. "Advanced Single Equation Analysis," beginning on page 261 documents two-stage least squares (TSLS) and generalized method of moments (GMM), autoregressive conditional heteroskedasticity (ARCH) models, single-equation cointegration equation specifications, discrete and limited dependent variable models, generalized linear models (GLM), robust least squares, least squares regression with breakpoints, threshold regression, switching regression, quantile regression, and user-specified likelihood estimation.
- Part VII. "Advanced Univariate Analysis," on page 665 describes advanced tools for univariate time series analysis, including unit root tests in both conventional and panel data settings, variance ratio tests, and the BDS test for independence.
- Part VIII. "Multiple Equation Analysis" on page 789 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 (VARs and VECs), state space models and model solution.
- Part IX. "Panel and Pooled Data" on page 1059 documents working with and estimating models with time series, cross-sectional data. The analysis may involve small numbers of cross-sections, with series for each cross-section variable (pooled data) or large numbers systems of cross-sections, with stacked data (panel data).
- Part X. "Advanced Multivariate Analysis," beginning on page 1243 describes tools for testing for cointegration and for performing Factor Analysis.

Part V. Basic Single Equation Analysis

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

- Chapter 20. "Basic Regression Analysis," beginning on page 5 outlines the basics of ordinary least squares estimation in EViews.
- Chapter 21. "Additional Regression Tools," on page 23 discusses special equation terms such as PDLs and automatically generated dummy variables, robust standard errors, weighted least squares, nonlinear least square, and the analysis of outliers and breaks indicator selection techniques.
- Chapter 22. "Regression Variable Selection," on page 65 discusses tools that may be used to automatically determine the variables used as regressors in a least squares regression.
- Chapter 23. "Instrumental Variables and GMM," on page 89 describes estimation of single equation Two-stage Least Squares (TSLS), Limited Information Maximum Likelihood (LIML) and K-Class Estimation, and Generalized Method of Moments (GMM) models.
- Chapter 24. "Time Series Regression," on page 119 describes a number of basic tools for analyzing and working with time series regression models: testing for serial correlation, estimation of ARMAX and ARIMAX models, and diagnostics for equations estimated using ARMA terms.
- Chapter 25. "Forecasting from an Equation," beginning on page 167 outlines the fundamentals of using EViews to forecast from estimated equations.
- Chapter 26. "Specification and Diagnostic Tests," beginning on page 195 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 VI. "Advanced Single Equation Analysis".

Multiple equation estimation is described in the chapters listed in Part VIII. "Multiple Equation Analysis".

Part IX. "Panel and Pooled Data" on page 1059 describes estimation in pooled data settings and panel structured workfiles.

Chapter 20. Basic Regression Analysis

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 advanced and specialized techniques such as weighted least squares, nonlinear least squares, ARIMA/ARIMAX models, two-stage least squares (TSLS), 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 (2013), *Introductory Econometrics: A Modern Approach, 5th Edition*.
- Greene (2008), *Econometric Analysis*, 6th Edition.
- Davidson and MacKinnon (1993), *Estimation and Inference in Econometric*s.

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:

$$\epsilon = \log(\text{urate}) - c(1)\text{dmr} - c(2)\,. \tag{20.1}$$

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 51.

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-Coef** 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 cointegrating regression, GLM or stepwise, or equations including MA terms, may only be specified by list and may not be specified by expression. All other types of equations (among others, ordinary least squares and two-stage least squares, equations with AR terms, GMM, and ARCH equations) may be specified either by list or expression. Note that some equations, such as quantile regression may be specified by expression, but only linear specifications are permitted.

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 136 of *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: 08/08/09   Time: 14:44
Sample (adjusted): 1959M01 1989M12
Included observations: 340 after adjustments
```

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 valid data 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 1959M01–1989M12 and specify the regression as:

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

If you set the estimation sample to the period 1959M01–1989M12, EViews adjusts the sample to:

```
Dependent Variable: M1
Method: Least Squares
Date: 08/08/09   Time: 14:45
Sample: 1960M01 1989M12
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 54.

Equation Output

When you click **OK** in the **Equation Specification** dialog, EViews displays the equation window displaying the estimation output view (the examples in this chapter are obtained using the workfile "Basics.WF1"):

Dependent Variable: LOG(M1)
Method: Least Squares
Date: 08/08/09 Time: 14:51
Sample: 1959M01 1989M12
Included observations: 372

Variable	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	97.00979	Hannan-Quinn criter.	-0.492878
F-statistic	1439.848	Durbin-Watson stat	0.008687
Prob(F-statistic)	0.000000		

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

$$y = X\beta + \epsilon \tag{20.2}$$

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:

$$b = (X'X)^{-1}X'y \tag{20.3}$$

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 you have included "C" in your list of regressors, the corresponding coefficient 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:

$$\mathrm{var}(b) = s^2(X'X)^{-1}; \qquad s^2 = \hat{\epsilon}'\hat{\epsilon}/(T-k); \qquad \hat{\epsilon} = y - Xb \tag{20.4}$$

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 often 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 for t-statistics are computed from a t-distribution with $T-k$ degrees of freedom. The p-value for z-statistics are computed using the standard normal distribution.

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:

$$R^2 = 1 - \frac{\hat{\epsilon}'\hat{\epsilon}}{(y-\bar{y})'(y-\bar{y})}; \qquad \bar{y} = \sum_{t=1}^{T} y_t / T \tag{20.5}$$

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:

$$\bar{R}^2 = 1 - (1 - R^2)\frac{T-1}{T-k} \tag{20.6}$$

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:

$$s = \sqrt{\frac{\hat{\epsilon}'\hat{\epsilon}}{(T-k)}} \tag{20.7}$$

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:

$$\hat{\epsilon}'\hat{\epsilon} = \sum_{t=1}^{T} (y_i - X_i'b)^2 \tag{20.8}$$

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:

$$l = -\frac{T}{2}(1 + \log(2\pi) + \log(\hat{\epsilon}'\hat{\epsilon}/T)) \tag{20.9}$$

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

Durbin-Watson Statistic

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

$$\mathrm{DW} = \sum_{t=2}^{T} (\hat{\epsilon}_t - \hat{\epsilon}_{t-1})^2 / \sum_{t=1}^{T} \hat{\epsilon}_t^2 \tag{20.10}$$

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 "Background," beginning on page 119, 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 127, 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:

$$\bar{y} = \sum_{t=1}^{T} y_t / T; \qquad s_y = \sqrt{\sum_{t=1}^{T} (y_t - \bar{y})^2 / (T-1)} \tag{20.11}$$

Akaike Information Criterion

The Akaike Information Criterion (AIC) is computed as:

$$\text{AIC} = -2l/T + 2k/T \tag{20.12}$$

where l is the log likelihood (given by Equation (20.9) 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 E. "Information Criteria," on page 1335, for additional discussion.

Schwarz Criterion

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

$$\text{SC} = -2l/T + (k \log T)/T \tag{20.13}$$

Hannan-Quinn Criterion

The Hannan-Quinn Criterion (HQ) employs yet another penalty function:

$$\text{HQ} = -2(l/T) + 2k \log(\log(T))/T \tag{20.14}$$

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:

$$F = \frac{R^2/(k-1)}{(1-R^2)/(T-k)} \tag{20.15}$$

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.

Note that since the F-statistic depends only on the sums-of-squared residuals of the estimated equation, it is not robust to heterogeneity or serial correlation. The use of robust estimators of the coefficient covariances ("Robust Standard Errors" on page 32) will have no effect on the F-statistic. If you do choose to employ robust covariance estimators, EViews will also report a robust Wald test statistic and p-value for the hypothesis that all non-intercept coefficients are equal to zero.

Working With Equation Statistics

The regression statistics reported in the estimation output view are stored with the equation. These equation data members 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 three kinds of "@-functions": those that return a scalar value, those that return matrices or vectors, and those that return strings.

Selected Keywords that Return Scalar Values

@aic	Akaike information criterion
@coefcov(i,j)	covariance of coefficient estimates i and j
@coefs(i)	i-th coefficient value
@dw	Durbin-Watson statistic
@f	F-statistic
@fprob	F-statistic probability.
@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

@rlogl	retricted (constant only) log-likelihood.
@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)

Selected Keywords that Return Vector or Matrix Objects

@coefcov	matrix containing the coefficient covariance matrix
@coefs	vector of coefficient values
@stderrs	vector of standard errors for the coefficients
@tstats	vector of t-statistic values for coefficients
@pvals	vector of p-values for coefficients

Selected Keywords that Return Strings

@command	full command line form of the estimation command
@smpl	description of the sample used for estimation
@updatetime	string representation of the time and date at which the equation was estimated

See also "Equation" (p. 35) in the *Object Reference* for a complete list.

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 11. "Matrix Language," on page 261 of the *Command and Programming Reference*.

Working with Equations

The follow description outlines common views and procedures that are available for an estimated equation. Specialized estimators may support only a subset of these views and procs, while perhaps offering others.

Views of an Equation

- **Representations**. Displays the equation in three basic forms: EViews command form showing the command associated with the equation, as an algebraic equation with symbolic coefficients, and as an equation with a text representation of 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, along with dotted lines showing at plus and minus one estimated standard error. **Residual Graph** plots only the residuals, while the **Standardized Residual Graph** plots the residuals divided by the estimated residual standard deviation.

obs	Actual	Fitted	Residual
1959M01	4.93375	4.59436	0.33940
1959M02	4.93735	4.62985	0.30750
1959M03	4.93950	4.65208	0.28742
1959M04	4.93950	4.68837	0.25113
1959M05	4.94663	4.71733	0.22930
1959M06	4.95018	4.71262	0.23756
1959M07	4.95371	4.67101	0.28271
1959M08	4.95512	4.60767	0.34745
1959M09	4.94876	4.60006	0.34870
1959M10	4.94521	4.58403	0.36118
1959M11	4.94450	4.59269	0.35180
1959M12	4.94164	4.69686	0.24478
1960M01			

- **ARMA structure...**. Provides views which describe the estimated ARMA structure of your residuals. Details on these views are provided in "ARMA Structure" on page 148.
- **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 D. "Gradients and Derivatives," on page 1327.
- **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 `@coefcov` member of the equation, as in

  ```
  sym mycov = eq1.@coefcov
  ```

- **Coefficient Diagnostics, Residual Diagnostics, and Stability Diagnostics**. These are views for specification and diagnostic tests and are described in detail in Chapter 26. "Specification and Diagnostic Tests," beginning on page 195.

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 25. "Forecasting from an Equation," on page 167.
- **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 if a named equation or the substituted coefficients representation of an untitled equation. This model can be solved in the usual manner. See Chapter 49. "Models," on page 995 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 using the command:

```
series res1 = resid
```

There is an even better approach to saving the residuals. 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.

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 101 and Chapter 10. "EViews Databases," beginning on page 325, both in *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 167), 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 "Selected Keywords that Return Scalar Values" on 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 40 of the *Object 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. (2008). *Econometric Analysis*, 6th 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. (2013). *Introductory Econometrics: A Modern Approach*. Mason, OH: South-Western – Cengage Learning.

Chapter 21. Additional Regression Tools

This chapter describes additional tools that may be used to augment the techniques described in Chapter 20. "Basic Regression Analysis," beginning on page 5.

- This first portion of this chapter describes special EViews expressions that may be used in specifying and estimating models with Polynomial Distributed Lags (PDLs) or dummy variables ("Special Equation Expressions," on page 23).
- In the second section, we describe methods for heteroskedasticity consistent, heteroskedasticity and autocorrelation consistent, and cluster robust covariance estimation ("Robust Standard Errors," on page 32).
- Next we describe tools for weighted least squares ("Weighted Least Squares," on page 47) and nonlinear least squares ("Nonlinear Least Squares," on page 51) estimation.
- Lastly, we describe the indicator saturation approach which extends basic regression analysis by testing for outliers and structural breaks in the regression specification ("Indicator Saturation" on page 60).

Note that 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 24. "Time Series Regression," on page 119.

Special Equation Expressions

EViews provides you with special expressions that may be used to specify and estimate equations with PDLs, dummy variables, or ARMA errors. We consider here terms for incorporating PDLs and dummy variables into your equation, and defer the discussion of ARMA estimation to "Time Series Regression" on page 119.

Polynomial Distributed Lags (PDLs)

A distributed lag is a relation of the type:

$$y_t = w_t\delta + \beta_0 x_t + \beta_1 x_{t-1} + \ldots + \beta_k x_{t-k} + \epsilon_t \tag{21.1}$$

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,

$$\beta_j = \gamma_1 + \gamma_2(j - \bar{c}) + \gamma_3(j - \bar{c})^2 + \ldots + \gamma_{p+1}(j - \bar{c})^p \tag{21.2}$$

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

$$\bar{c} = \begin{cases} (k)/2 & \text{if } k \text{ is even} \\ (k-1)/2 & \text{if } k \text{ is odd} \end{cases} \tag{21.3}$$

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 (21.2) into (21.1), yielding,

$$y_t = w_t\delta + \gamma_1 z_1 + \gamma_2 z_2 + \ldots + \gamma_{p+1} z_{p+1} + \epsilon_t \tag{21.4}$$

where:

$$\begin{aligned} z_1 &= x_t + x_{t-1} + \ldots + x_{t-k} \\ z_2 &= -\bar{c}x_t + (1-\bar{c})x_{t-1} + \ldots + (k-\bar{c})x_{t-k} \\ &\ldots \\ z_{p+1} &= (-\bar{c})^p x_t + (1-\bar{c})^p x_{t-1} + \ldots + (k-\bar{c})^p x_{t-k} \end{aligned} \tag{21.5}$$

Once we estimate γ from Equation (21.4), we can recover the parameters of interest β, and their standard errors using the relationship described in Equation (21.2). 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:

$$\beta_{-1} = \gamma_1 + \gamma_2(-1 - \bar{c}) + \ldots + \gamma_{p+1}(-1 - \bar{c})^p = 0. \tag{21.6}$$

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

$$\beta_{k+1} = \gamma_1 + \gamma_2(k + 1 - \bar{c}) + \ldots + \gamma_{p+1}(k + 1 - \bar{c})^p = 0. \tag{21.7}$$

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) in the workfile "Basics.WF1" by entering the command:

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

which yields the following results:

Dependent Variable: IP
Method: Least Squares
Date: 08/08/09 Time: 15:27
Sample (adjusted): 1960M01 1989M12
Included observations: 360 after adjustments

Variable	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,

```
smpl 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: 08/08/09 Time: 15:35
Sample (adjusted): 1960M01 1989M12
Included observations: 360 after adjustments

Variable	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 (21.2) on page 24. The terms PDL01, PDL02, PDL03, …, correspond to z_1, z_2, … in Equation (21.4).

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:

Lag Distribution of...	i	Coefficient	Std. Error	t-Statistic
	0	0.10270	0.14677	0.69970
	1	0.01159	0.10948	0.10587
	2	-0.00215	0.10139	-0.02123
	3	0.00920	0.06150	0.14955
	4	0.01766	0.07435	0.23756
	5	0.01363	0.06974	0.19547
	6	-4.7E-05	0.05557	-0.00084
	7	-0.01399	0.07080	-0.19764
	8	-0.01821	0.07537	-0.24158
	9	-0.00798	0.06399	-0.12475
	10	0.01017	0.10454	0.09726
	11	0.01260	0.11069	0.11386
	12	-0.04737	0.15693	-0.30182
	Sum of Lags	0.08780	0.00297	29.5534

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 Diagnostics** 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 Diagnostics/Wald Test-Coefficient Restrictions…** and enter:

```
c(6)=0, c(7)=0
```

in the Wald Test dialog box (see "Wald Test (Coefficient Restrictions)" on page 202 for an extensive discussion of Wald tests in EViews). EViews displays the result of the joint test:

Wald Test:
Equation: Untitled
Null Hypothesis: C(6)=0, C(7)=0

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	0.000408
C(7)	-4.93E-05	0.000180

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 age @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. Thus, 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). The data are available in the workfile "Hprice2.WF1".

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: 08/08/09 Time: 22:11
Sample: 1 506
Included observations: 506

Variable	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: 08/08/09 Time: 22:15
Sample: 1 506
Included observations: 506

Variable	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		

Robust Standard Errors

In the linear least squares regression model, the variance-covariance matrix of the estimated coefficients may be written as:

$$V(\hat{\beta}) = E(\hat{\beta} - \beta)(\hat{\beta} - \beta)' = (X'X)^{-1} X'\Omega X(X'X)^{-1} \tag{21.8}$$

where $\Omega \equiv E(\epsilon\epsilon')$.

If the error terms, ϵ, are homoskedastic and uncorrelated so that $E(\epsilon\epsilon') = \sigma^2 I$, the covariance matrix simplifies to the familiar expression

$$V(\hat{\beta}) = (X'X)^{-1}\sigma^2(X'X)(X'X)^{-1} = \sigma^2(X'X)^{-1}. \tag{21.9}$$

By default, EViews estimates the coefficient covariance matrix under the assumptions underlying Equation (21.9), so that

$$\hat{\Omega} = \left[\frac{\sum_{t=1}^{T}\hat{\epsilon}_t^2}{T-k}\right] I = s^2 I \tag{21.10}$$

and

$$\hat{V}(\hat{\beta}) = (X'X)^{-1}(X's^2 IX)(X'X)^{-1} = s^2(X'X)^{-1} \tag{21.11}$$

where s^2 is the standard degree-of-freedom corrected estimator of the residual variance.

We may instead employ robust estimators of the coefficient variance $V(\hat{\beta})$ which relax the assumptions of heteroskedasticity and/or zero correlation. Broadly speaking, EViews offers three classes of robust variance estimators that are:

- Robust in the presence of heteroskedasticity. Estimators in this first class are termed Heteroskedasticity Consistent (HC) Covariance estimators.
- Robust in the presence of correlation between observations in different groups or clusters. This second class consists of the family of Cluster Robust (CR) variance estimators.
- Robust in the presence of heteroskedasticity *and* serial correlation. Estimators in the third class are referred to as Heteroskedasticity and Autocorrelation Consistent Covariance (HAC) estimators.

All of these estimators are special cases of *sandwich estimators* of the coefficient covariances. The name follows from the structure of the estimators in which different estimates of Ω are sandwiched between two instances of an outer moment matrix.

It is worth emphasizing all three of these approaches alter the estimates of the coefficient standard errors of an equation but not the point estimates themselves.

Lastly, our discussion here focuses on the linear model. The extension to nonlinear regression is described in "Nonlinear Least Squares" on page 51.

Heteroskedasticity Consistent Covariances

First, we consider coefficient covariance estimators that are robust to the presence of heteroskedasticity.

We divide our discussion of HC covariance estimation into two groups: basic estimators, consisting of White (1980) and degree-of-freedom corrected White (Davidson and MacKinnon 1985), and more general estimators that account for finite samples by adjusting the weights given to residuals on the basis of leverage (Long and Ervin, 2000; Cribari-Neto and da Silva, 2011).

Basic HC Estimators

White (1980) derives a heteroskedasticity consistent covariance matrix estimator which provides consistent estimates of the coefficient covariances in the presence of (conditional) heteroskedasticity of unknown form, where

$$\Omega = E(\epsilon\epsilon') = \mathrm{diag}(\sigma_1^2, \sigma_2^2, \ldots, \sigma_T^2) \tag{21.12}$$

Recall that the coefficient variance in question is

$$V(\hat{\beta}) = E(\hat{\beta} - \beta)(\hat{\beta} - \beta)' = (X'X)^{-1} X'\Omega X(X'X)^{-1}$$

Under the basic White approach. we estimate the central matrix $\Phi = X'\Omega X$ using either the d.f. corrected form

$$\hat{\Phi}_{W1} = \frac{T}{T-K} \sum_{t=1}^{T} \hat{\epsilon}_t^2 X_t X_t' \tag{21.13}$$

or the uncorrected form

$$\hat{\Phi}_{W0} = \sum_{t=1}^{T} \hat{\epsilon}_t^2 X_t X_t' \tag{21.14}$$

where $\hat{\epsilon}_t$ are the estimated residuals, T is the number of observations, k is the number of regressors, and $T/(T-k)$ is the conventional degree-of-freedom correction.

The estimator of $\hat{\Phi}$ is then used to form the heteroskedasticity consistent coefficient covariance estimator. For example, the degree-of-freedom White heteroskedasticity consistent covariance matrix estimator is given by

$$\hat{\Sigma}_{HC1} = (X'X)^{-1} \left(\frac{T}{T-k} \sum_{t=1}^{T} \hat{\epsilon}_t^2 X_t X_t' \right) (X'X)^{-1} \tag{21.15}$$

Estimates using this approach are typically referred to as *White* or *Huber-White* or (for the d.f. corrected case) *White-Hinkley* covariances and standard errors.

Example

To illustrate the computation of White covariance estimates in EViews, we employ an example from Wooldridge (2000, p. 251) of an estimate of a wage equation for college professors. The equation uses dummy variables to examine wage differences between four groups of individuals: married men (MARRMALE), married women (MARRFEM), single women (SINGLEFEM), and the base group of single men. The explanatory variables include levels of education (EDUC), experience (EXPER) and tenure (TENURE). The data are in the workfile "Wooldridge.WF1".

To select the White covariance estimator, specify the equation as before, then select the **Options** tab and select **Huber-White** in the **Covariance method** drop-down. You may, if desired, use the checkbox to remove the default **d.f. Adjustment**, but in this example, we will use the default setting. (Note that the **Information matrix** combo setting is not important in linear specifications).

The output for the robust covariances for this regression are shown below:

Dependent Variable: LOG(WAGE)
Method: Least Squares
Date: 31/05/17 Time: 12:15
Sample: 1 526
Included observations: 526
White-Hinkley (HC1) heteroskedasticity consistent standard errors and covariance

Variable	Coefficient	Std. Error	t-Statistic	Prob.
C	0.321378	0.109469	2.935791	0.0035
MARRMALE	0.212676	0.057142	3.721886	0.0002
MARRFEM	-0.198268	0.058770	-3.373619	0.0008
SINGFEM	-0.110350	0.057116	-1.932028	0.0539
EDUC	0.078910	0.007415	10.64246	0.0000
EXPER	0.026801	0.005139	5.215010	0.0000
EXPER^2	-0.000535	0.000106	-5.033361	0.0000
TENURE	0.029088	0.006941	4.190731	0.0000
TENURE^2	-0.000533	0.000244	-2.187835	0.0291

R-squared	0.460877	Mean dependent var	1.623268
Adjusted R-squared	0.452535	S.D. dependent var	0.531538
S.E. of regression	0.393290	Akaike info criterion	0.988423
Sum squared resid	79.96799	Schwarz criterion	1.061403
Log likelihood	-250.9552	Hannan-Quinn criter.	1.016998
F-statistic	55.24559	Durbin-Watson stat	1.784785
Prob(F-statistic)	0.000000	Wald F-statistic	51.69553
Prob(Wald F-statistic)	0.000000		

As Wooldridge notes, the heteroskedasticity robust standard errors for this specification are not very different from the non-robust forms, and the test statistics for statistical significance of coefficients are generally unchanged. While robust standard errors are often larger than their usual counterparts, this is not necessarily the case, and indeed in this example, there are some robust standard errors that are smaller than their conventional counterparts.

Notice that EViews reports both the conventional residual-based F-statistic and associated probability and the robust Wald test statistic and p-value for the hypothesis that all non-intercept coefficients are equal to zero.

Recall that the familiar residual F-statistic for testing the null hypothesis depends only on the coefficient point estimates, and not their standard error estimates, and is valid only under the maintained hypotheses of no heteroskedasticity or serial correlation. For ordinary least squares with conventionally estimated standard errors, this statistic is numerically identical to the Wald statistic. When robust standard errors are employed, the numerical equivalence between the two breaks down, so EViews reports both the non-robust conventional residual and the robust Wald F-statistics.

EViews reports the robust F-statistic as the **Wald F-statistic** in equation output, and the corresponding p-value as **Prob(Wald F-statistic)**. In this example, both the non-robust F-statis-

tic and the robust Wald show that the non-intercept coefficients are jointly statistically significant.

Alternative HC Estimators

The two familiar White covariance estimators described in "Basic HC Estimators" on page 33 are two elements of a wider class of HC methods (Long and Ervin, 2000; Cribari-Neto and da Silva, 2011).

This general class of heteroskedasticity consistent sandwich covariance estimators may be written as:

$$\hat{\Sigma}_{HC} = (X'X)^{-1}\left(\sum_{t=1}^{T}(d_t\hat{\epsilon}_t)^2 X_t X_t'\right)(X'X)^{-1} \tag{21.16}$$

where d_t are observation-specific weights that are chosen to improve finite sample performance.

The various members of the class are obtained through different choices for the weights. For example, the standard White and d.f. corrected White estimators are obtained by setting $d_t = 1$ and $d_t = T/(T-k)$ for all t, respectively.

EViews allows you to estimate your covariances using several choices for d_t. In addition to the standard White covariance estimators from above, EViews supports the bias-correcting HC2, pseudo-jackknife HC3 (MacKinnon and White, 1985), and the leverage weighting HC4, HC4m, and HC5 (Cribari-Neto, 2004; Cribaro-Neto and da Silva, 2011; Cribari-Neto, Souza, and Vasconcellos, 2007 and 2008).

The weighting functions for the various HC estimators supported by EViews are provided below:

Method	d_t
HC0 – White	1
HC1 – White with d.f. correction	$\sqrt{T/(T-k)}$
HC2 – bias corrected	$(1-h_t)^{-1/2}$
HC3 – pseudo-jackknife	$(1-h_t)^{-1}$
HC4 – relative leverage	$(1-h_t)^{-\delta_t/2}$

HC4m	$(1-h_t)^{-\delta_t/2}$
HC5	$(1-h_t)^{-\delta_t/4}$
User – user-specified	arbitrary

where $h_t = X_t'(X'X)^{-1}X_t$ are the diagonal elements of the familiar "hat matrix" $H = X(X'X)^{-1}X'$.

Note that the HC0 and HC1 methods correspond to the basic White estimators outlined earlier.

Note that HC4, HC4m, and HC5 all depend on an exponential discounting factor δ_t that differs across methods:

- For HC4,

  $$\delta_t = \min(Th_i/k, 4)$$

  is a truncated function of the ratio between h_t and the mean $\bar{h}$ (Cribari-Neto, 2004; p. 230).

- For HC4m,

  $$\delta_t = \min(Th_t/k, \kappa_1) + \min(Th_t/k, \kappa_2)$$

  where κ_1 and κ_2 are pre-specified parameters (Cribari-Neto and da Silva, 2011). Following Cribari-Neto and da Silva,

  EViews chooses default values of $\kappa_1 = 1.0$ and $\kappa_2 = 1.5$.

- For HC5,

  $$\delta_t = \min(Th_t/k, \max(4, \kappa Th_{\max}/k))$$

  is similar to the HC4 version of δ_t, but with observation specific truncation that depends on the maximal leverage and a pre-specified parameter κ (Cribari-Neto, Souza, and Vasconcellos, 2007 and 2008).

  EViews employs a default value of $\kappa = 0.7$.

Lastly, to allow for maximum flexibility, EViews allows you to provide user-specified d_t in the form of a series containing those values.

Each of the weight choices modifies the effects of high leverage observations on the calculation of the covariance. See Cribari-Neto (2004), Cribari-Neto and da Silva (2011), and Cribari-Neto, Souza, and Vasconcellos (2007, 2008) for discussion of the effects of these choices.

Note that the full set of HC estimators is only available for linear regression models. For non-linear regression models, the leverage based methods are not available and only user-specified may be computed.

Example

To illustrate the use of the alternative HC estimators, we continue with the Wooldridge example ("Example" on page 34) considered above. We specify the equation variables as before, then select the **Options** tab and click on the **Covariance method** drop-down and select **HC (Various)**:

The dialog will change to show you additional options for selecting the **HC method**:

Note that the **HC0 (ordinary)** and **HC1 (d.f. adjusted)** items replicate the **Huber-White** option from the original **Covariance method** dropdown and are included in this list for completeness. If desired, change the method from the default **HC2 (bias adjusted)**, and if necessary, specify values for the parameters. For example, if you select User-specified, you will be prompted to provide the name of a series in the workfile containing the values of the weights d_t.

Continuing with our example, we use the **Covariance method** dropdown to select the **HC5** method, and retain the default value $\kappa = 0.7$. Click on **OK** to estimate the model with these settings producing the following results:

Dependent Variable: LOG(WAGE)
Method: Least Squares
Date: 31/05/17 Time: 12:11
Sample: 1 526
Included observations: 526
Cribari-Neto et al. (H5) heteroskedasticity-consistent standard errors
& covariance (k=.7)

Variable	Coefficient	Std. Error	t-Statistic	Prob.
C	0.321378	0.110175	2.916988	0.0037
MARRMALE	0.212676	0.057869	3.675129	0.0003
MARRFEM	-0.198268	0.059220	-3.347974	0.0009
SINGFEM	-0.110350	0.057236	-1.927970	0.0544
EDUC	0.078910	0.007502	10.51918	0.0000
EXPER	0.026801	0.005147	5.206623	0.0000
EXPER^2	-0.000535	0.000107	-4.995281	0.0000
TENURE	0.029088	0.009312	3.123691	0.0019
TENURE^2	-0.000533	0.000387	-1.378837	0.1685

R-squared	0.460877	Mean dependent var	1.623268
Adjusted R-squared	0.452535	S.D. dependent var	0.531538
S.E. of regression	0.393290	Akaike info criterion	0.988423
Sum squared resid	79.96799	Schwarz criterion	1.061403
Log likelihood	-250.9552	Hannan-Quinn criter.	1.016998
F-statistic	55.24559	Durbin-Watson stat	1.784785
Prob(F-statistic)	0.000000	Wald F-statistic	51.54430
Prob(Wald F-statistic)	0.000000		

The effects on statistical inference resulting from a different HC estimator are minor, though the quadratic effect of TENURE is no longer significant at conventional test levels.

Cluster-Robust Covariances

In many settings, observations may be grouped into different groups or "clusters" where errors are correlated for observations in the same cluster and uncorrelated for observations in different clusters. EViews offers support for consistent estimation of coefficient covariances that are robust to either one and two-way clustering.

We begin with a single clustering classifier and assume that

$$\begin{aligned} E(\epsilon_i \epsilon_j) &\neq 0 \\ E(\epsilon_i \epsilon_h) &= 0 \end{aligned} \tag{21.17}$$

for all i and j in the same cluster, and all i and h that are in different clusters. If we assume that the number of clusters G goes to infinity, we may compute a cluster-robust (CR) covariance estimate that is robust to both heteroskedasticity and to within-cluster correlation (Liang and Zeger, 1986; Wooldridge, 2003; Cameron and Miller, 2015).

As with the HC and HAC estimators, the cluster-robust estimator is based upon a sandwich form with an estimator the central matrix $\Phi = X'\Omega X$:

$$\hat{\Phi}_{CR} = \sum_{g=1}^{G} X_g' D_g \hat{\epsilon}_g \hat{\epsilon}_g' D_g' X_g \tag{21.18}$$

where X_g is the $T_g \times k$ matrix of regressors for the T_g observations in the g-th cluster, ϵ_g is a T_g vector of errors, and D_g is a $T_g \times T_g$ diagonal matrix of weights for the observations in the cluster. The resulting family of CR variance estimators is given by:

$$\hat{\Sigma}_{CR} = (X'X)^{-1}\left(\sum_{g=1}^{G} X_g' D_g \hat{\epsilon}_g \hat{\epsilon}_g' D_g' X_g\right)(X'X)^{-1} \tag{21.19}$$

The EViews supported weighting functions for the various CR estimators are analogues to those available for HC estimation:

Method	d_t
CR0 – Ordinary	1
CR1 – finite sample corrected (default)	$\sqrt{\frac{G}{(G-1)} \cdot \frac{(T-1)}{(T-k)}}$
CR2 – bias corrected	$(1-h_t)^{-1/2}$
CR3 – pseudo-jackknife	$(1-h_t)^{-1}$
CR4 – relative leverage	$(1-h_t)^{-\delta_t/2}$
CR4m	$(1-h_t)^{-\delta_t/2}$
CR5	$(1-h_t)^{-\delta_t/4}$
User – user-specified	arbitrary

where $h_t = X_t'(X'X)^{-1}X_t$ are the diagonal elements of $H = X(X'X)^{-1}X'$. For further discussion and detailed definitions of the discounting factor δ_t in the various methods, see "Alternative HC Estimators" on page 36. See Cameron and Miller (CM 2015, p. 342) for discussion of bias adjustments in the context of cluster robust estimation.

Note that the EViews CR3 differs slightly from the CR3 described by CM in not including the $\sqrt{G/(G-1)}$ factor, and that we have defined the CR4, CR4m and CR5 estimators which employ weights that are analogues to those defined for HC covariance estimation.

We may easily extend the robust variance calculation to two-way clustering to handle cases where observations are correlated within two different clustering dimensions (Petersen 2009, Thompson 2011, Cameron, Gelbach, and Miller 2015).

It is easily shown that the variance estimator for clustering by A and B may be written as:

$$\hat{\Sigma}_{CR} = \hat{\Sigma}_{CR}^{A} + \hat{\Sigma}_{CR}^{B} - \hat{\Sigma}_{CR}^{A \cap B} \tag{21.20}$$

where the $\hat{\Sigma}_{CR}^{j}$ is used to indicate the estimator obtained assuming clustering along the given dimension. Thus, the estimator is formed by adding the covariances obtained by clustering along each of the two dimensions individually, and subtracting off the covariance obtained by defining clusters for the intersection of the two dimensions.

Note that EViews does not perform the eigenvalue adjustment in cases where the resulting estimate is not positive semidefinite.

If you elect to compute cluster-robust covariance estimates, EViews will adjust the *t*-statistic probabilities in the main estimation output to account for the clustering and will note this adjustment in the output comments. Following Cameron and Miller (CM, 2015), the probabilities are computed using the *t*-distribution with $G - 1$ degrees-of-freedom in the one-way cluster case, and by $\min(G_A, G_B) - 1$ degrees-of-freedom under two-way clustering. Bear in mind that CM note that even with these adjustments, the tests tend to overreject the null.

Furthermore, when cluster-robust covariances are computed, EViews will not display the residual-based F-statistic for the test of significance of the non-intercept regressors. The robust Wald-based F-statistic will be displayed.

Example

We illustrate the computation of cluster-robust covariance estimation in EViews using the test data provided by Petersen via his website:

http://www.kellogg.northwestern.edu/faculty/petersen/htm/papers/se/test_data.htm

The data are provided as an EViews workfile "Petersen_cluster.WF1". There are 5000 observations on four variables in the workfile, the dependent variable Y, independent variable X, and two cluster variables, a firm identifier (FIRMID), and time identifier (YEAR). There are 500 firms, and 10 periods in the balanced design.

First, create the equation object in EViews by selecting **Object/New Object…/Equation** or **Quick/Estimate Equation…** from the main menu, or simply type the keyword `equation` in the command window. Enter, the regression specification "Y C X" in the **Specification** edit field, and click on **OK** to estimate the equation using standard covariance settings.

The results of this estimation are given below:

Dependent Variable: Y
Method: Least Squares
Date: 30/05/17 Time: 15:44
Sample: 1 5000
Included observations: 5000

Variable	Coefficient	Std. Error	t-Statistic	Prob.
C	0.029680	0.028359	1.046560	0.2954
X	1.034833	0.028583	36.20414	0.0000

R-squared	0.207766	Mean dependent var	0.035238
Adjusted R-squared	0.207607	S.D. dependent var	2.252704
S.E. of regression	2.005277	Akaike info criterion	4.229841
Sum squared resid	20097.64	Schwarz criterion	4.232448
Log likelihood	-10572.60	Hannan-Quinn criter.	4.230755
F-statistic	1310.740	Durbin-Watson stat	1.096121
Prob(F-statistic)	0.000000		

Next, to estimate the equation with FIRMID cluster-robust covariances, click on the **Estimate** button on the equation toolbar to display the estimation dialog, and then click on the **Options** tab to show the **Coefficient covariance** options.

Select **Cluster robust** in the **Covariance method** dropdown, enter "FIRMID" in the **Cluster series** edit field, and select a **CR method**. Here, we choose the **CR1 (finite sample)** method which employs a simple d.f. style adjustment to the basic cluster covariance estimate. Click on **OK** to estimate the equation using these settings.

The results are displayed below:

Dependent Variable: Y
Method: Least Squares
Date: 30/05/17 Time: 15:50
Sample: 1 5000
Included observations: 5000
CR1 (finite sample adjusted) cluster-robust standard errors & covariance
Cluster series: FIRMID (500 clusters)
Standard errors and t-statistic probabilities adjusted for clustering

Variable	Coefficient	Std. Error	t-Statistic	Prob.
C	0.029680	0.067013	0.442897	0.6580
X	1.034833	0.050596	20.45298	0.0000

R-squared	0.207766	Mean dependent var	0.035238
Adjusted R-squared	0.207607	S.D. dependent var	2.252704
S.E. of regression	2.005277	Akaike info criterion	4.229841
Sum squared resid	20097.64	Schwarz criterion	4.232448
Log likelihood	-10572.60	Hannan-Quinn criter.	4.230755
Durbin-Watson stat	1.096121	Wald F-statistic	418.3244
Prob(Wald F-statistic)	0.000000		

The top portion of the equation output describes both the cluster method (CR1) and the cluster series (FIRMID), along with the number of clusters (500) observed in the estimation sample. In addition, EViews indicates that the reported coefficient standard errors, and t-statistic probabilities have been adjusted for the clustering. As noted earlier, the probabilities are computed using the t-distribution with $G - 1 = 499$ degrees-of-freedom.

Note also that EViews no longer displays the ordinary F-statistic and associated probability, but instead shows the robust Wald F-statistic and probability.

Dependent Variable: Y
Method: Least Squares
Date: 30/05/17 Time: 15:50
Sample: 1 5000
Included observations: 5000
CR1 (finite sample adjusted) cluster-robust standard errors & covariance
Cluster series: FIRMID (500 clusters), YEAR (10 clusters)
Standard errors and t-statistic probabilities adjusted for clustering

Variable	Coefficient	Std. Error	t-Statistic	Prob.
C	0.029680	0.065064	0.456163	0.6591
X	1.034833	0.053558	19.32173	0.0000

R-squared	0.207766	Mean dependent var	0.035238
Adjusted R-squared	0.207607	S.D. dependent var	2.252704
S.E. of regression	2.005277	Akaike info criterion	4.229841
Sum squared resid	20097.64	Schwarz criterion	4.232448
Log likelihood	-10572.60	Hannan-Quinn criter.	4.230755
Durbin-Watson stat	1.096121	Wald F-statistic	373.3291
Prob(Wald F-statistic)	0.000000		

For two-way clustering, we create an equation with the same regression specification, click on the **Options** tab, and enter the two cluster series identifiers "FIRMID YEAR" in the **Cluster series** edit field. Leaving the remaining options at their current settings, click on **OK** to compute and display the estimation results:

Dependent Variable: Y
Method: Least Squares
Date: 30/05/17 Time: 15:23
Sample: 1 5000
Included observations: 5000
CR1 (finite sample adjusted) cluster-robust standard errors & covariance
Cluster series: FIRMID (500 clusters), YEAR (10 clusters)
Standard errors and t-statistic probabilities adjusted for clustering

Variable	Coefficient	Std. Error	t-Statistic	Prob.
X	1.034833	0.053558	19.32173	0.0000
C	0.029680	0.065064	0.456163	0.6591

R-squared	0.207766	Mean dependent var	0.035238
Adjusted R-squared	0.207607	S.D. dependent var	2.252704
S.E. of regression	2.005277	Akaike info criterion	4.229841
Sum squared resid	20097.64	Schwarz criterion	4.232448
Log likelihood	-10572.60	Hannan-Quinn criter.	4.230755
Durbin-Watson stat	1.096121	Wald F-statistic	373.3291
Prob(Wald F-statistic)	0.000000		

The output shows that the cluster-robust covariance computation is now based on two-way clustering using the 500 clusters in FIRMID and the 10 clusters in YEAR. The t-statistic probabilities are now based on the t-distribution with $\min(500, 10) - 1 = 9$ degrees-of-freedom.

Notice in all of these examples that the coefficient point estimates and basic fit measures are unchanged. The only changes in results are in the estimated standard errors, and the associated t-statistics, probabilities, and the F-statistics and probabilities.

Lastly, we note that the standard errors and corresponding statistics in the EViews two-way results differ slightly from those reported on the Petersen website. These differences appear to be the result of slightly different finite sample adjustments in the computation of the three individual matrices used to compute the two-way covariance. When you select the CR1 method, EViews adjusts each of the three matrices using the CR1 finite sample adjustment; Petersen's example appears to apply CR1 to the one-way cluster covariances, while the joint two-way cluster results are computing using CR0.

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) propose a covariance estimator that is consistent in the presence of both heteroskedasticity and autocorrelation (HAC) of unknown form, under the assumption that the autocorrelations between distant observations die out. NW advocate using kernel methods to form an estimate of the long-run variance, $E(X'\epsilon\epsilon'X/T)$.

EViews incorporates and extends the Newey-West approach by allowing you to estimate the HAC consistent coefficient covariance estimator given by:

$$\hat{\Sigma}_{NW} = (X'X)^{-1} T\hat{\Lambda}(X'X)^{-1} \tag{21.21}$$

where $\hat{\Lambda}$ is any of the LRCOV estimators described in Appendix F. "Long-run Covariance Estimation," on page 1339.

To use the Newey-West HAC method, select the **Options** tab and select **HAC (Newey-West)** in the **Coefficient covariance matrix** drop-down. As before, you may use the checkbox to remove the default **d.f. Adjustment**.

Press the **HAC options** button to change the options for the LRCOV estimate.

We illustrate the computation of HAC covariances using an example from Stock and Watson (2007, p. 620). In this example, the percentage change of the price of orange juice is regressed upon a constant and the number of days the temperature in Florida reached zero for the current and previous 18 months, using monthly data from 1950 to 2000 The data are in the workfile "Stock_wat.WF1".

Stock and Watson report Newey-West standard errors computed using a non pre-whitened Bartlett Kernel with a user-specified bandwidth of 8 (note that the bandwidth is equal to *one plus* what Stock and Watson term the "truncation parameter" m).

The results of this estimation are shown below:

Dependent Variable: 100*D(LOG(POJ))
Method: Least Squares
Date: 02/10/13 Time: 06:44
Sample: 1950M01 2000M12
Included observations: 612
HAC standard errors & covariance (Bartlett kernel, User bandwidth = 8.0000)

Variable	Coefficient	Std. Error	t-Statistic	Prob.
FDD	0.503798	0.139563	3.609818	0.0003
FDD(-1)	0.169918	0.088943	1.910407	0.0566
FDD(-2)	0.067014	0.060693	1.104158	0.2700
FDD(-3)	0.071087	0.044894	1.583444	0.1139
FDD(-4)	0.024776	0.031656	0.782679	0.4341
FDD(-5)	0.031935	0.030763	1.038086	0.2997
FDD(-6)	0.032560	0.047602	0.684014	0.4942
FDD(-7)	0.014913	0.015743	0.947323	0.3439
FDD(-8)	-0.042196	0.034885	-1.209594	0.2269
FDD(-9)	-0.010300	0.051452	-0.200181	0.8414
FDD(-10)	-0.116300	0.070656	-1.646013	0.1003
FDD(-11)	-0.066283	0.053014	-1.250288	0.2117
FDD(-12)	-0.142268	0.077424	-1.837518	0.0666
FDD(-13)	-0.081575	0.042992	-1.897435	0.0583
FDD(-14)	-0.056372	0.035300	-1.596959	0.1108
FDD(-15)	-0.031875	0.028018	-1.137658	0.2557
FDD(-16)	-0.006777	0.055701	-0.121670	0.9032
FDD(-17)	0.001394	0.018445	0.075584	0.9398
FDD(-18)	0.001824	0.016973	0.107450	0.9145
C	-0.340237	0.273659	-1.243289	0.2143

R-squared	0.128503	Mean dependent var	-0.115821
Adjusted R-squared	0.100532	S.D. dependent var	5.065300
S.E. of regression	4.803944	Akaike info criterion	6.008886
Sum squared resid	13662.11	Schwarz criterion	6.153223
Log likelihood	-1818.719	Hannan-Quinn criter.	6.065023
F-statistic	4.594247	Durbin-Watson stat	1.821196
Prob(F-statistic)	0.000000	Wald F-statistic	2.257876
Prob(Wald F-statistic)	0.001769		

Note in particular that the top of the equation output documents the use of HAC covariance estimates along with relevant information about the settings used to compute the long-run covariance matrix.

The HAC robust Wald p-value is slightly higher than the corresponding non-robust F-statistic p-value, but are significant at conventional test levels.

Weighted Least Squares

Suppose that you have heteroskedasticity of known form, where the conditional error variances are given by σ_t^2. The presence of heteroskedasticity does not alter the bias or consis-

tency properties of ordinary least squares estimates, but OLS is no longer efficient and conventional estimates of the coefficient standard errors are not valid.

If the variances σ_t^2 are known up to a positive scale factor, you may use weighted least squares (WLS) to obtain efficient estimates that support valid inference. Specifically, if

$$\begin{aligned} y_t &= x_t'\beta + \epsilon_t \\ E(\epsilon_t | X_t) &= 0 \\ Var(\epsilon_t | X_t) &= \sigma_t^2 \end{aligned} \tag{21.22}$$

and we observe $h_t = a\sigma_t^2$, the WLS estimator for β minimizes the weighted sum-of-squared residuals:

$$\begin{aligned} S(\beta) &= \sum_t \frac{1}{h_t}(y_t - x_t'\beta)^2 \\ &= \sum_t w_t(y_t - x_t'\beta)^2 \end{aligned} \tag{21.23}$$

with respect to the k-dimensional vector of parameters β, where the weights $w_t = 1/h_t$ are proportional to the inverse conditional variances. Equivalently, you may estimate the regression of the square-root weighted transformed data $y_t^* = \sqrt{w_t} \cdot y_t$ on the transformed $x_t^* = \sqrt{w_t} \cdot x_t$.

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 matrices associated with y_t and x_t. The WLS estimator may be written,

$$\hat{\beta}_{WLS} = (X'WX)^{-1}X'Wy \tag{21.24}$$

and the default estimated coefficient covariance matrix is:

$$\hat{\Sigma}_{WLS} = s^2(X'WX)^{-1} \tag{21.25}$$

where

$$s^2 = \frac{1}{T-k}(y - X\hat{\beta}_{WLS})'W(y - X\hat{\beta}_{WLS}) \tag{21.26}$$

is a d.f. corrected estimator of the weighted residual variance.

To perform WLS in EViews, open the equation estimation dialog and select a method that supports WLS such as **LS—Least Squares (NLS and ARMA)**, then click on the **Options** tab. (You should note that weighted estimation is not offered in equations containing ARMA specifications, nor is it available for some equation methods, such as those estimated with ARCH, binary, count, censored and truncated, or ordered discrete choice techniques.)

You will use the three parts of the **Weights** section of the **Options** tab to specify your weights.

The **Type** dropdown is used to specify the form in which the weight data are provided. If, for example, your weight series VARWGT contains values proportional to the conditional variance, you should select **Variance**. Alternately, if your series INVARWGT contains the values proportional to the inverse of the standard deviation of the residuals you should choose **Inverse std. dev.**

Next, you should enter an expression for your weight series in the **Weight series** edit field.

Lastly, you should choose a scaling method for the weights. There are three choices: **Average**, **None**, and (in some cases) **EViews default**. If you select **Average**, EViews will, prior to use, scale the weights so that the w_i sum to T. The **EViews default** specification scales the weights so the square roots of the w_i sum to T. (The latter square root scaling, which offers backward compatibility to EViews 6 and earlier, was originally introduced in an effort to make the weighted residuals $\sqrt{w_t} \cdot (y_t - x_t'\beta)$ comparable to the unweighted residuals.) Note that the EViews default method is only available if you select **Inverse std. dev.** as weighting **Type**.

> *Unless there is good reason to do so, we recommend that you employ **Inverse std. dev. weights** with **EViews default** scaling, even if it means you must transform your weight series. The other weight types and scaling methods were introduced in EViews 7, so equations estimated using the alternate settings may not be read by prior versions of EViews.*

We emphasize the fact that b_{WLS} and $\hat{\Sigma}_{WLS}$ are almost always invariant to the scaling of weights. One important exception to this invariance occurs in the special case where some of the weight series values are non-positive since observations with non-positive weights will be excluded from the analysis unless you have selected **EViews default** scaling, in which case only observations with zero weights are excluded.

As an illustration, we consider a simple example taken from Gujarati (2003, Example 11.7, p. 416) which examines the relationship between compensation (Y) and index for employment size (X) for nine nondurable manufacturing industries. The data, which are in the workfile "Gujarati_wls.WF1", also contain a series SIGMA believed to be proportional to the standard deviation of each error.

To estimate WLS for this specification, open an equation dialog and enter

```
y c x
```

as the equation specification.

Click on the **Options** tab, and fill out the **Weights** section as depicted here. We select **Inverse std. dev.** as our **Type**, and specify "1/SIGMA" for our **Weight series**. Lastly, we select **EViews default** as our **Scaling** method.

Click on **OK** to estimate the specified equation. The results are given by:

Dependent Variable: Y
Method: Least Squares
Date: 06/17/09 Time: 10:01
Sample: 1 9
Included observations: 9
Weighting series: 1/SIGMA
Weight type: Inverse standard deviation (EViews default scaling)

Variable	Coefficient	Std. Error	t-Statistic	Prob.
C	3406.640	80.98322	42.06600	0.0000
X	154.1526	16.95929	9.089565	0.0000

Weighted Statistics

R-squared	0.921893	Mean dependent var	4098.417
Adjusted R-squared	0.910734	S.D. dependent var	629.1767
S.E. of regression	126.6652	Akaike info criterion	12.71410
Sum squared resid	112308.5	Schwarz criterion	12.75793
Log likelihood	-55.21346	Hannan-Quinn criter.	12.61952
F-statistic	82.62018	Durbin-Watson stat	1.183941
Prob(F-statistic)	0.000040	Weighted mean dep.	4039.404

Unweighted Statistics

R-squared	0.935499	Mean dependent var	4161.667
Adjusted R-squared	0.926285	S.D. dependent var	420.5954
S.E. of regression	114.1939	Sum squared resid	91281.79
Durbin-Watson stat	1.141034		

The top portion of the output displays the estimation settings which show both the specified weighting series and the type of weighting employed in estimation. The middle section shows the estimated coefficient values and corresponding standard errors, *t*-statistics and probabilities.

The bottom portion of the output displays two sets of statistics. The **Weighted Statistics** show statistics corresponding to the actual estimated equation. For purposes of discussion, there are two types of summary statistics: those that are (generally) invariant to the scaling of the weights, and those that vary with the weight scale.

The "R-squared", "Adjusted R-squared", "F-statistic" and "Prob(F-stat)", and the "Durbin-Watson stat", are all invariant to your choice of scale. Notice that these are all fit measures or test statistics which involve ratios of terms that remove the scaling.

One additional invariant statistic of note is the "Weighted mean dep." which is the weighted mean of the dependent variable, computed as:

$$\bar{y}_w = \frac{\sum w_t y_t}{\sum w_t}, \tag{21.27}$$

The weighted mean is the value of the estimated intercept in the restricted model, and is used in forming the reported F-test.

The remaining statistics such as the "Mean dependent var.", "Sum squared resid", and the "Log likelihood" all depend on the choice of scale. They may be thought of as the statistics computed using the weighted data, $y_t^* = \sqrt{w_t} \cdot y_t$ and $x_t^* = \sqrt{w_t} \cdot x_t$. For example, the mean of the dependent variable is computed as $(\sum y_t^*)/T$, and the sum-of-squared residuals is given by $\sum w_t (y_t^* - x_t^{*\prime}\beta)^2$. These values should not be compared across equations estimated using different weight scaling.

Lastly, EViews reports a set of **Unweighted Statistics**. As the name suggests, these are statistics computed using the unweighted data and the WLS coefficients.

Nonlinear Least Squares

Suppose that we have the regression specification:

$$y_t = f(x_t, \beta) + \epsilon_t, \tag{21.28}$$

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:

$$S(\beta) = \sum_t (y_t - f(x_t, \beta))^2 = (y - f(X, \beta))'(y - f(X, \beta)) \tag{21.29}$$

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:

$$y_t = \beta_1 + \beta_2 \log L_t + \beta_3 \log K_t + \epsilon_t. \tag{21.30}$$

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:

$$y_t = \beta_1 L_t^{\beta_2} K_t^{\beta_3} + \epsilon_t \tag{21.31}$$

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 parameters, estimates my be obtained from iterative methods as described in "Optimization Algorithms," beginning on page 1319.

Estimates of the coefficient covariance take the general form:

$$\hat{\Sigma}_{NLLS} = cA^{-1}BA^{-1} \tag{21.32}$$

where A is an estimate of the information, B is the variance of the residual weighted gradients, and c is a scale parameter.

For the ordinary covariance estimator, we assume that $A = B$. Then we have

$$\hat{\Sigma}_{NLLS} = cA^{-1} \tag{21.33}$$

where c is an estimator of the residual variance (with or without degree-of-freedom correction).

As in Amemiya (1983), we may estimate A using the outer-product of the gradients (OPG) so we have

$$\hat{\Sigma}_{NLLS} = c \cdot \left(\frac{\partial f(\beta)'}{\partial \beta} \frac{\partial f(\beta)}{\partial \beta} \right)^{-1} \tag{21.34}$$

where the derivatives are evaluated at b_{NLLS}.

Similarly, we may set A to the one-half of the Hessian matrix of second derivatives of the sum-of-squares function:

$$\hat{\Sigma}_{NLLS} = c \cdot \left(\frac{1}{2} \cdot \frac{\partial^2 S(\beta)}{\partial \beta \partial \beta'} \right)^{-1} \tag{21.35}$$

evaluated at b_{NLLS}.

Alternately, we may assume distinct A and B and employ a White or HAC sandwich estimator for the coefficient covariance as in "Robust Standard Errors," beginning on page 32. In this case, A is estimated using the OPG or Hessian, and the B is a robust estimate of the variance of the gradient weighted residuals. In this case, c is a scalar representing the degree-of-freedom correction, if employed.

For additional discussion of nonlinear estimation, see Pindyck and Rubinfeld (1998, p. 265-273), Davidson and MacKinnon (1993), or Amemiya(1983).

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 C. "Estimation and Solution Options," beginning on page 1313.

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:

$$S(c(1), c(2)) = \sum_t \{ y_t - (c(1)x_t + c(2)z_t + 4)^2 \}^2 \tag{21.36}$$

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:

$$S(c(1), c(2)) = \sum_t \{-(c(1)x_t + c(2)z_t + 4)^2\}^2 \tag{21.37}$$

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:

$$x_t + c(1)y_t - z_t^{c(2)} = \epsilon_t \tag{21.38}$$

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

Estimation Options

Clicking on the **Options** tab displays the nonlinear least squares estimation options:

Coefficient Covariance

EViews allows you to compute ordinary coefficient covariances using the inverse of either the OPG of the mean function or the observed Hessian of the objective function, or to com-

pute robust sandwich estimators for the covariance matrix using White or HAC (Newey-West) estimators.

- The topmost **Covariance method** dropdown menu should be used to choose between the default **Ordinary** or the robust **Huber-White** or **HAC (Newey-West)** methods.
- In the **Information matrix** menu you should choose between the **OPG** and the **Hessian - observed** estimators for the information.
- If you select **HAC (Newey-West)**, you will be presented with a **HAC options** button that, if pressed, brings up a dialog to allow you to control the long-run variance computation.

See "Robust Standard Errors," beginning on page 32 for a discussion of White and HAC standard errors.

You may use the **d.f. Adjustment** checkbox to enable or disable the degree-of-freedom correction for the coefficient covariance. For the **Ordinary** method, this setting amounts to determining whether the residual variance estimator is or is not degree-of-freedom corrected. For the sandwich estimators, the degree-of-freedom correction is applied to the entire matrix.

Optimization

You may control the iterative process by specifying the optimization method, convergence criterion, and maximum number of iterations.

The **Optimization method** dropdown menu lets you choose between the default **Gauss-Newton** and **BFGS**, **Newton-Raphson**, and **EViews legacy** methods.

In general, the differences between the estimates should be small for well-behaved nonlinear specifications, but if you are experiencing trouble, you may wish to experiment with methods. Note that EViews legacy is a particular implementation of Gauss-Newton with Marquardt or line search steps, and is provided for backward estimation compatibility.

The **Step method** menu allows you to select the approach for candidate iterative steps. The default method is **Marquardt**, but you may instead select **Dogleg** or **Line Search**.

See "Optimization Method" on page 1314, and "Optimization Algorithms" on page 1319 for related discussion.

EViews will report that the estimation procedure has converged if the convergence test value is below your convergence tolerance. While there is no best choice of convergence tolerance, and the choice is somewhat individual, as a guideline note that we generally set ours something on the order of 1e-8 or so and then adjust it upward if necessary for models with difficult to compute numeric derivatives.

See "Iteration and Convergence" on page 1314 for additional discussion.

In most cases, you need not change the maximum number of iterations. However, for some difficult to estimate models, the iterative procedure may not converge within the maximum number of iterations. If your model does not converge within the allotted number of iterations, one solution is to click the **Estimate** button and increase the maximum number of iterations. Click **OK** to accept the options, and **OK** again 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 A, "Estimation Defaults" on page 947 for details.

Derivative Methods

Estimation in EViews requires computation of the derivatives of the regression function with respect to the parameters.

In most cases, you need not worry about the settings for the derivative computation. The EViews estimation engine will employ analytic expressions for the derivatives, if possible, or will compute high numeric derivatives, switching between lower precision computation early in the iterative procedure and higher precision computation for later iterations and final computation. You may elect to use only numeric derivatives.

See "Derivative Computation" on page 1317 for additional discussion.

Starting Values

Iterative estimation procedures require starting values for the coefficients of the model. 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 may need to experiment in order to find starting values.

There are no general rules for selecting starting values for parameters so there are no settings in this page for choosing 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 current 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, followed by each coefficient and desired value. For example, if your default coefficient vector is C, the statement:

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

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

See Appendix C, “Estimation and Solution Options” on page 1313, for further details.

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 using the data in the workfile “Chow_-var.WF1”:

Dependent Variable: LOG(CS)
Method: Least Squares (Gauss-Newton / Marquardt steps)
Date: 03/09/15 Time: 11:25
Sample: 1947Q1 1994Q4
Included observations: 192
Convergence achieved after 68 iterations
Coefficient covariance computed using outer product of gradients
LOG(CS)=C(1)+C(2)*(GDP^C(3)-1)/C(3)

	Coefficient	Std. Error	t-Statistic	Prob.
C(1)	2.839332	0.281733	10.07810	0.0000
C(2)	0.259119	0.041680	6.216837	0.0000
C(3)	0.182315	0.020335	8.965475	0.0000

R-squared	0.997260	Mean dependent var	7.472280
Adjusted R-squared	0.997231	S.D. dependent var	0.463744
S.E. of regression	0.024403	Akaike info criterion	-4.572707
Sum squared resid	0.112552	Schwarz criterion	-4.521808
Log likelihood	441.9798	Hannan-Quinn criter.	-4.552093
F-statistic	34393.45	Durbin-Watson stat	0.136871
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, and a description of the method employed in computing the coefficient covariances, 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.

NLS with ARMA 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,

$$\begin{aligned} CS_t &= c_1 + GDP_t^{c_2} + u_t \\ u_t &= c_3 u_{t-1} + c_4 u_{t-2} + \epsilon_t \end{aligned} \tag{21.39}$$

you should enter the specification:

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

See "Initializing the AR Errors," on page 162 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.

Weighted NLS

Weights can be used in nonlinear estimation in a manner analogous to weighted linear least squares in equations without ARMA terms. To estimate an equation using weighted nonlinear least squares, enter your specification, press the **Options** button and fill in the weight specification.

EViews minimizes the sum of the weighted squared residuals:

$$S(\beta) = \sum_t w_t(y_t - f(x_t, \beta))^2 = (y - f(X, \beta))' W(y - f(X, \beta)) \tag{21.40}$$

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

$$\frac{\partial f(\beta)}{\partial \beta}' W(y - f(X, \beta)) = 0 \tag{21.41}$$

and the default OPG d.f. corrected covariance estimate is computed as:

$$\hat{\Sigma}_{WNLLS} = s^2\left(\frac{\partial f(\beta)}{\partial \beta}' W \frac{\partial f(\beta)}{\partial \beta}\right)^{-1} \tag{21.42}$$

and the corresponding Hessian estimate is

$$\hat{\Sigma}_{WNLLS} = s^2\left(\frac{1}{2} \cdot \frac{\partial^2 S(\beta)}{\partial \beta \partial \beta'}\right)^{-1} \tag{21.43}$$

for the weighted objective function given in Equation (21.41).

One may elect, of course, to compute a White or HAC sandwich estimator for the coefficient covariance as in "Robust Standard Errors," beginning on page 32.

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 in "Starting Values" on page 56 for details on 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:

$$y_t = \beta_1\beta_2 + \beta_2^2 x_t + \epsilon_t \tag{21.44}$$

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).

Optimization Algorithm

In general, the choice of optimization algorithm should have little effect on the computation of estimates. That said, if you are experiencing trouble, you may wish to experiment with different methods. In addition, you may wish to experiment with different optimizers to ensure that your estimates are robust to the choice of optimization method.

Note that EViews legacy is a particular implementation of Gauss-Newton with Marquardt or line search steps, and is provided for backward estimation compatibility.

See "Optimization" on page 55 for discussion.

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 1313, for related discussion.

Indicator Saturation

Traditionally, the analysis of outliers and structural breaks in regression analysis was conducted by testing the statistical significance of a parsimonious, predetermined set of related indicator variables. More recently, advances in computing power and the development of general-to-specific (GETS) modeling permit testing of indicators at each observation in the estimation sample.

The *indicator saturation* approach is an extension of least squares regression for testing for outliers and structural breaks in a regression specification. The indicator saturation approach works by including indicator variables for outliers or structural breaks at every observation in the regression, and then employing the GETS algorithms to select which of the included variables should be retained in a final regression model.

Note that inclusion of indicators at each observation would ordinarily be prohibited due to the singularity of the regressor matrix, but the introduction of block estimation in GETS modeling "Auto-Search / GETS" on page 68, allows testing of indicators throughout the estimation sample.

In EViews, you may perform indicator saturation on simple linear equations (which do not include ARMA terms) estimated using least squares. EViews offers testing for three different indicator types at each period t:

- **Impulse Indicators (IIS)**: a dummy variable equal to zero everywhere other than a single value of one at period t. This indicator is equivalent to the `@isperiod` function used at the date corresponding to t.
- **Step Indicators (SIS)**: a step function variable equal to zero until t and one thereafter. This indicator is equivalent to the `@after` function used at the date corresponding to t.
- **Trend Indicators (TIS):** a trend-break variable that is equal to zero until period t and then a follows a trend afterward. This indicator is equivalent to the `@trendbr` function used at the date corresponding to t.

(Note that an SIS indicator at t and $t+1$ is numerically the same as an IIS indicator at t. Similarly, a TIS indicator at t and $t+1$ is numerically identical to an SIS indicator at t. For this reason, when testing for these combinations of indicators, EViews will automatically drop the numerically equivalent variables.)

To instruct EViews to detect indicators in your least squares regression, open the equation estimation dialog, enter your least squares specification in the **Equation specification** edit field, and select **LS - Least Squares (NLS and ARMA)** in the **Method** dialog. Next, click on the **Options** tab to display the dialog:

Select the **Auto-detect** check box in the **Outlier/indicator saturation** area on the right-hand side of the dialog, and then press the **Options** tab to bring up the **Indicator Options** dialog:

There are a number of settings which control the indicator saturation analysis:

- The **Indicator types** section allows you to specify which type of indicator you would like to detect. You may individually choose to test for **Impulse (IIS)**, **Step-shift (SIS)**, and **Trend (TIS)** indicators by selecting the corresponding checkbox.
- In the **Model selection** area, the **Criterion** dropdown specifies the information criteria used to select the final model from the candidate models. The **Include empty model** check box specifies whether to the empty model (with no indicators) as possible a possible candidate model.
- The **Sample** edit field in the **Variable creation ar**ea allows you to specify a different set of dates over which to create the indicators. By default, the estimation sample is used, but you may enter a different sample if desired.
- The **Blocks** edit field allows you to specify the number of blocks the indicators will be split into. Since the number of indicators will exceed the number of observations, indicators will be added in blocks. EViews will automatically determine the optimal number of blocks, but you may enter your own choice in this field to override the EViews default.

  The **Chronological blocking** and **Alternating blocking** radio buttons determine whether the indicators are split into blocks in chronologically (the first group of dates in the first block, followed by the next group in the second block and so on), or alternating (the first date in the first block, second date in the second block and so on).

- The **Diagnostics** set of options allows selection of which indicator saturation diagnostic tests to include along with their p-values. You may perform **AR LM** (autoregressive LM), **ARCH LM**, **Normality**, and the **PET** (Parsimonious Encompassing Test) testing. For the **AR LM** test and **ARCH LM** test, you should specify the number of lags to include.

Once an equation has been estimated with indicator saturation variables, a new post-estimation diagnostic is available from the **View** menu. Select **View/Indicator Summary** to display a summary of the indicators that were included, and, if any were detected, produces a chart of the trajectory of the indicators through time.

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.

Cameron, A. Colin, Gelbach, Jonah B., and Douglas L. Miller (2011). "Robust Inference with Multiway Clustering," *Journal of Business & Economic Statistics*, 29, 238–249.

Cameron, A. Colin, and Douglas L. Miller (2015). "A Practitioner's Guide to Cluster-Robust Inference," *Journal of Human Resources*, 50, 317–337.

Castle, Jennifer L., Doornik, Jurgen A., Hendry, David F., and Felix Pretis (2015). "Detecting location shifts during model selection by step-indicator saturation." *Econometrics*, 3, 240–264.

Cribari-Neto, Francisco (2004). "Asymptotic inference under heteroskedasticity of unknown form," *Computational Statistics & Data Analysis*, 45, 215–233

Cribari-Neto, Francisco, and Wilton Bernardino da Silva (2011). "A new heteroskedasticity-consistent covariance matrix estimator for the linear regression model," *AStA Advances in Statistical Analysis*, 95, 129–148.

Cribari-Neto, Francisco, Souza, Tatiene C., and Klaus L. P. Vasconcellos (2007), "Inference Under Heteroskedasticity and Leveraged Data", *Communications in Statistics - Theory and Methods*, 36, 1877–1888.

Cribari-Neto, Francisco, Souza, Tatiene C., and Klaus L. P. Vasconcellos (2008), "Errata: Inference Under Heteroskedasticity and Leveraged Data", *Communications in Statistics - Theory and Methods*, 37, 3329–3330.

Davidson, Russell and James G. MacKinnon (1993). *Estimation and Inference in Econometric*s, Oxford: Oxford University Press.

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.

Johansen, Soren, and Bent Nielsen (2016). "Asymptotic theory of outlier detection algorithms for linear time series regression models," *Scandinavian Journal of Statistics,* 43, 321–348.

Johnston, Jack, and John Enrico DiNardo (1997). *Econometric Methods*, 4th Edition, New York: McGraw-Hill.

Liang, Kung-Yee, and Scott L. Zeger (1986). "Longitudinal Data Analysis Using Generalized Linear Models," Biometrika, 73, 13–22.

Long, J. Scott, and Laurie Ervin (2000). "Using Heteroscedasticity Consistent Standard Errors in the Linear Regression Model," *The American Statistician*, 54, 217–224.

MacKinnon, James G., and Halbert White (1985). "Some Heteroskedasticity Consistent Covariance Matrix Estimators with Improved Finite Sample Properties," *Journal of Econometrics*, 29, 53–57.

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.

Petersen, M. (2009), "Estimating Standard Errors in Finance Panel Data Sets: Comparing Approaches," *Review of Financial Studies*, 22, 435–480.

Pindyck, Robert S. and Daniel L. Rubinfeld (1998). *Econometric Models and Economic Forecasts*, 4th edition, New York: McGraw-Hill.

Pretis, Felix, Read, J. James, and Genaro Succarat (2018). "Automated General-to-Specific (GETS) regression modeling and indicator saturation for outliers and structural breaks," *Journal of Statistical Software*, 86, 1–44.

Santos, Carlos, Hendry, David, F., and Soren Johansen (2007). "Automatic selection of indicators in a fully saturated regression," *Computational Statistics,* 23, 317–335.

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.

Thompson, Samuel B. (2011)."Simple formulas for standard errors that cluster by both firm and time," *Journal of Financial Economics*, 99, 1–10.

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*. Cincinnati, OH: South-Western College Publishing.

Wooldridge, Jeffrey M. (2003). "Cluster-Sample Methods in Applied Econometrics," The American Economic Review, 93, 133–138.

Chapter 22. Regression Variable Selection

This chapter describes tools that may be used to automatically determine the variables used as regressors in a least squares regression.

The explosion in available data over recent decades, coupled with increases in computing power, has led to growing popularity of methods that allow the data themselves to suggest the most appropriate combination of regressors to use in estimation. These techniques allow the researcher to provide a set of candidate variables for the model, rather than specifying a specific model.

The chapter begins with some general discussion concerning variable selection in EViews, then describes three variable selection techniques available, and concludes with some examples:

- "Variable Selection Methods" on page 66 in EViews provides an introduction to variable selection in EViews.

Each of the supported methods is described in some detail:

- "Uni-Directional" on page 66.
- "Stepwise" on page 67.
- "Swapwise" on page 67.
- "Combinatorial" on page 68.
- "Auto-Search / GETS" on page 68—an improvement of stepwise regression techniques by adding tests of model validity to the selection process.
- "Lasso" on page 69 outlines how the Lasso estimation technique, described in "Chapter 36. Elastic Net and Lasso" be used as a pre-estimation tool to determine the regressors in least squares.

Background

The explosion in available data over recent decades, coupled with increases in computing power, has led to the growing popularity of methods that allow the data themselves to suggest the most appropriate combination of regressors to use in estimation. These techniques allow the researcher to provide a set of candidate variables for the model, rather than specifying a specific model. *Variable selection*, or *feature selection*, is an important component of modern data analysis.

Variable selection techniques are implemented in EViews as a pre-estimation step before performing a standard least squares regression. Before estimation, you must specify a dependent variable together with a list of always-included variable, and a list of selection

variables, from which the selection algorithm will choose the most appropriate. Following the variable 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.

Variable Selection Methods

EViews offers five different variable selection methods: Uni-directional, Stepwise, Swapwise, Combinatorial, Auto-Search/GETS and Lasso Selection.

Uni-Directional

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

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 may be used in place of the lowest p-value as a 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

The *swapwise* methods both begin with no additional regressors in the model.

Swapwise-Max R-Squared Increment

The *Max R-squared Increment* method 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 Increment* 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.

Auto-Search / GETS

The *auto-search/GETS* algorithm follows the steps suggested by AutoSEARCH algorithm of Escribano and Sucarrat (2011), which in turn builds upon the work in Hoover and Perez 1999. The algorithm is similar to the uni-directional-backwards method:

1. The model with all search variables (termed the general unrestricted model, GUM) is estimated, and checked with a set of diagnostic tests.
2. A number of search *paths* are defined, one for each insignificant search variable in the GUM.
3. For each path, the insignificant variable defined in 2) is removed and then a series of further variable removal steps is taken, each time removing the most insignificant variable, and each time checking whether the current model passes the set of diagnos-

tic tests. If the diagnostic tests fail after the removal of a variable, that variable is placed back into the model and prevented from being removed again along this path. Variable removal finishes once there are no more insignificant variables, or it is impossible to removal a variable without failing the diagnostic tests.

4. Once all paths have been calculated the final models produced by the paths are compared using an information criteria selection. The best model is then selected.

For the suite of diagnostic tests performed on both the GUM and in each path, EViews includes:

- AR LM test—a Q-statistic test on the level residuals at the specified significance and number of lags.
- ARCH LM test—a Q-statistic test on the squared residuals at the specified significance and number of lags.
- A Jarque-Bera normality test.
- A parsimonious encompassing test (PET)—for models other than the GUM, an F-test of the overall significance of the removed regressors compared to the GUM.
- Auto-search/GETS allows for cases where there are more search variables than observations in the estimation sample – a situation typically impossible due to singularity of the regressor matrix. In such cases the algorithm is split into blocks, where each block has its own GUM with a subset of the search regressors. The algorithm works through each block individually, and then combines the individually selected variables from each block into a new GUM and runs one more time.

Lasso

The Lasso (Least Absolute Shrinkage and Selection Operator) estimator is the OLS estimator with an L1 penalty term that is designed to guard against overfitting:

$$J = \frac{1}{2m}\sum_{i=1}^{m}\left(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j\right)^2 + \lambda\sum_{j=1}^{p}\left|\beta_j\right| \tag{22.1}$$

with the parameter λ determining the extent of the penalty and the resulting shrinkage toward zero of the parameters. We see here that the standard least squares objective function is augmented with a function that penalizes for non-zero coefficients, pushing coefficients toward zero. This is also called shrinkage.

The Lasso variable selection method implemented in EViews is a sophisticated model selection routine that estimates numerically a Lasso model, possibly performing cross-validation to determine an optimal λ, and optionally transforming regressors.

Variables whose coefficients shrink to zero in the optimal Lasso specification are excluded from the selected model.

(Caution should be used when applying Lasso selection to models with highly collinear variables, as this method will select one variable at random from a group of such variables.)

See “Chapter 37. “Elastic Net and Lasso,” on page 535 for additional discussion.

Variable Selection Estimation in EViews

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

The **Specification** page allows you to provide the basic VARSEL 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 VARSEL 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 **Selection method** dropdown box can be used to specify the type of selection procedure will be used to determine the final model. By default, EViews will estimate the variable selection using the **Stepwise** method. To change the basic method, change the **Selection**

method dropdown menu; the dropdown allows you to choose between: **Uni-directional**, **Stepwise**, **Swapwise**, **Combinatorial**, **Auto-Search/GETS** and **Lasso** selection.

Next, you may use the **Options** tab to more finely tune the selection method. The options available will depend on the selection method chosen, and are described below.

Estimation Options

When you select one of the variable selection methods, the **Options** tab of the dialog will change to display the relevant settings.

Uni-directional and Stepwise Options

For the **Uni-directional** and **Stepwise** methods you may specify the direction of the method using the **Forwards** and **Backwards** radio buttons and 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.

The **Weights** portion of the dialog is the same as that used in other estimators such as least squares. For more information, please see "Weighted Least Squares" on page 47.

Swapwise Options

The Swapwise variable selection method lets you choose whether you wish to use Max R-squared Increment or Min R-squared Increment, and to enter the number of additional variables to be selected.

By default, the number of additional variables is set to one so that if you do not enter a value, EViews will select the single variable that will lead to the largest increase in R-squared.

The **Weights** portion of the dialog is the same as that used in other estimators such as least squares. For more information, please see "Weighted Least Squares" on page 47.

Combinatorial Options

The combinatorial options page simply prompts you to provide the number of additional variables.

By default, the number of additional variables is set to one so that if you do not enter a value, EViews will select the single variable that will lead to the largest increase in R-squared

The **Weights** portion of the dialog is the same as that used in other estimators such as least squares. For more information, please see "Weighted Least Squares" on page 47.

Auto-Search / GETS Options

When **Auto-Search/GETS** is chosen as the **Selection method** on the **Specification** tab, the **Options** tab will change to display options specific to the Auto-Search/GETS algorithm:

The options are divided into several sections.

- The **Model selection** section offers options for the selection criteria used during the variable selection process. The **Criteria** dropdown specifies the information criteria used to select the final model from the candidates remaining in step 4) of the selection algorithm. The **Include GUM** and **Include empty model** checkboxes specify whether to include either the general model (i.e. the model with all possible search variables) or the empty model (the model with zero search variables) as candidates.

  The **Blocks** edit field allows you to overwrite the EViews' choice for the number of blocks into which to split the search variables if the number of search variables is greater than the number of observations.

- The **Diagnostics** section identifies the options for the diagnostic tests used in the diagnostic tests steps 1) and 3) of the algorithm ("Auto-Search / GETS" on page 68). The **Terminal condition p-value** is used to set the p-value against which the significance of the remaining variable is tested when determining whether to stop the selection process along a path in step 3).

The **AR LM test**, **ARCH LM test**, **Normality test** and **PET test** checkboxes are corresponding p-value edit fields are used to determine whether those tests are used when determining the validity of both the GUM and each selection model, and their associated p-value. For both the AR and ARCH LM tests, the number of lags used in the test may also be specified.

The **Weights** portion of the dialog is the same as that used in other estimators such as least squares. For more information, please see "Weighted Least Squares" on page 47.

Lasso Options

For Lasso selection, the options dialog allows you specify the data transformation options for the regressors and, potentially, the properties of the lambda path, optimization settings, and details of the cross-validation:

(All of the options available for elastic net are also available for Lasso selection with the exception of the @VW tag.)

There are several sections in the dialog.

Regularization specification

The **Regularization specification** section prompts you to specify the regressor transformation and settings for lambda.

- For the former, the **Regressor transformation** combo may be used to choose between **None** (the default), **Standardization** by sample and population, **L1** and **L2** normalizations, and **Min-max** transformation.

For multiple *supplied* or default generated values of lambda in the **Penalty** section, the available options will expand to show additional lambda options:

- In the **Min/max lambda ratio** text box, you may supply the ratio of the minimum to the maximum value of lambda desired; the default is 0.0001.
- In the **No. lambdas on path** text box, you may supply the number of values of lambda on the path; the default is 100.

Estimation algorithm

The familiar **Estimation algorithm** section controls the specification of starting values and iterative estimation controls.

Penalty

If the Lasso selection method is chosen, then elastic net estimation is applied with $\alpha = 1$ and variable weights equal to zero for the always included variables.

You should specify information about the Lasso lambda in the **Lambda** edit field.

You are free to fill in the lambda field with a single value, multiple values separated by spaces, or a series object from the workfile. If multiple values of lambda are entered, either directly as a list or through a series object, cross-validation will be performed to determine the value giving the model with the lowest measurement error.

You may also leave the lambda field blank. If you choose to do this EViews will generate its own list of values based on the data.

Cross-validation options

For multiple *supplied* or default generated values of lambda in the **Penalty** section, the available options will expand to include the **Cross-validation options**. Certain options within this section can be chosen depending on the particular method of cross-validation chosen (**Folds** for **K-Fold cross-validation**, for example).

The available **Cross-validation options** include:

- **Shuffle**: After the dataset is divided into training and test sets, the ordering is shuffled. The **Random generator** and **Seed** fields control the details of the shuffling. You may the leave the **Seed** field blank, in which case EViews will use the clock to obtain a seed at the time of estimation, or you may provide an integer from 0 to 2,147,483,647. The **Clear** button may be used to clear the seed used by a previously estimated equation. By changing the value of the **Shuffle reps** field from the default of

1 you may create multiple datasets with different, shuffled ordering of training and test sets.

- **K-Fold**: The dataset is divided into K evenly spaced "folds" and the ordering is shuffled (the details of the shuffling are determined by the **Random generator** and **Seed** fields). One fold is held out as the test set while the remaining K-1 folds are combined into the training set. This process is then repeated, with each fold being held out in turn as the test set. After model estimation the statistics are averaged over all K folds.
- **Leave One Out**: This is the same as K-Fold, but with the number of folds equal to the number of observations.
- **Leave P Out**: This is similar to K-Fold, with the exception that a test set of size P is held out, with the remaining data forming the training set. This process is repeated over all remaining combinations.
- **Rolling Window**: After a window size is chosen for the dataset, the window is divided into training and test sets (the default test set size is 1, and you may also choose the test set size as a fraction), with the test set in each window always coming *after* the training set. The window "rolls" through the dataset until it reaches the end of the dataset. You may also choose how far ahead of the training set you want the test set to be (the horizon), as well as an initial period in the dataset to hold out of the cross-validation (the initial period).
- **Expanding Window**: The dataset is divided into training and test sets (the default test set size is 1, and only integer test set sizes are allowed), with the test set in each window always coming *after* the training set. With each iteration the training set size expands by one while the test set stays the same size. You may choose how far ahead of the training set you want the test set to be (the horizon), as well as an initial period in the dataset to hold out of the cross-validation (the initial period).

Weights

The **Weights** portion of the dialog is the same as that used in other estimators such as least squares. For more information, please see "Weighted Least Squares" on page 47.

Working with an Estimated Equation

Since variable selection is a pre-estimation tool, once estimated, the standard views and procs available for an equation estimated by least-squares will be available.

In addition, if **Auto-Search/GETS** or **Lasso** was selected as the **Selection Method**, a **Model Selection Summary** item and several sub-menu items will be available on the equation **View** menu.

For both methods, there will be three sub-menu entries: **Criteria Graph**, **Criteria Table**, and **Summary Path**:

- The **Criteria Graph** and **Criteria Table** items shows the model selection criteria for the twenty "best" candidate models. The graph form shows just the chosen criteria method (AIC, SIC or HQ). The table form of the view shows the log-likelihood value, the AIC, BIC and HQ values.
- The **Summary Path** view provides a summary of the selection/dropping of variables along each selection path. For each path, the table provides the order in which variables were dropped, along with relevant selection information.

For Lasso, there are two additional sub-menu entries: **Coefficient matrix**, **Lambda Coefficient Graph**:

- The **Coefficient Matrix** view is one of two views that allow you to easily see the behavior of the model over the entire lambda path. The leftmost column lists all the lambdas used in the estimation. The remaining columns display the coefficients at each value of lambda.
- The **Lambda Coefficient Graph** displays values of coefficients ordered by the values of lambda in the lambda path. This is the graph you would see by plotting each column (excluding the constant, if included) of the **View/Coefficient Matrix** table against the leftmost lambda path column.

Example

As an example of using variable selection in EViews, we will estimate a model of hourly electricity spot prices. We will follow, in spirit, the analysis done in Uniejewski, Nowotarski and Weron (2016), estimating a model with a large number of candidate variables to try to perform a one day-ahead forecast of the spot price.

We have a workfile containing hourly data between 2015 and 2018 of the electricity spot price for a region in the United States, along with hourly electricity load data (how much electricity is used) and day-ahead load forecasts for the same region.

As a dependent variable we use the series TPRICE containing the deviation of the log of the spot price from the mean of the log of spot prices for the same hour.

We have a total of 107 search regressors:

- 3 days (72 hours) of lags of TPRICE (72)
- a single one week (168 hour) lag of TPRICE (1)
- the minimum TPRICE of the previous day, two days before and three days before (3)
- the maximum TPRICE of the previous day, two days before and three days before (3)
- the mean TPRICE of the previous day, two days before and three days before (3)

- log of electricity load for current hour, same hour previous day and same hour previous week (3)
- log of forecasted load for same hour next day (1)
- day of week dummy variables (set to zero for weekday holidays) (7)
- Interaction between dummy variables and log of load (y)
- Interaction between dummy variables and TPRICE (7)

These search regressions are stored in the group SEARCHREGS.

The model also has an always-included constant.

For this example we will model TPRICE at noon each day between January 1, 2015 and December 30, 2018, and will use those estimates to forecast TPRICE for December 31, 2018 —a one observation forecast.

Stepwise Example

To begin, we'll use a stepwise regression to select the most appropriate regressors. We click on **Quick/Estimate Equation…** to bring up the **Equation Estimation** dialog, and then change the **Method** dropdown to **VARSEL-Variable selection and Stepwise Least Squares**.

- We enter "TPRICE C" in the first part of the **Equation Specification**, and then "SEARCHREGS" in the **Search Regressors** box.
- We'll leave the **Selection method** as **Stepwise**, and keep the options on the **Options** tab at their default settings.
- Finally we set the sample to be the estimation dates we want, ensuring that only the noon observations are used.

Clicking **OK** performs the stepwise variable selection, and estimates the final model. The top portion of the output describes the selection process:

```
Dependent Variable: TPRICE
Method: Variable Selection
Date: 09/29/20   Time: 16:41
Sample: 1/15/2015 00:00 12/30/2018 23:00 IF @HOUR=12
Included observations: 1435
Number of always included regressors: 1
Number of search regressors: 107
Selection method: Stepwise forwards
Stopping criterion: p-value forwards/backwards = 0.5/0.5
Note: final equation sample is larger than selection sample (rejected
      regressors contain missing values)
```

The final line issues a warning that the final estimation results shown use a different sample than that used during selection. This is because some of the search regressors contain NAs (due to lags in our case), and thus those observations are dropped from the selection process. However some of those variables were not selected, and so those observations can be re-included in the final estimation.

The middle part of the output (partially shown here) displays the selected variables, their coefficients, standard errors, t-statistics and probability values.

Variable	Coefficient	Std. Error	t-Statistic	Prob.*
C	-0.845562	0.218424	-3.871204	0.0001
TPRICE(-1)	1.047528	0.025817	40.57524	0.0000
LOG(ACTLOAD)	0.074737	0.028514	2.621099	0.0089
TPRICE(-5)	-0.036752	0.013575	-2.707206	0.0069
TPRICE(-20)	0.075522	0.019309	3.911224	0.0001
TPRICE(-34)	0.083448	0.026820	3.111367	0.0019
LOG(ACTLOAD(-24))	-0.105708	0.016625	-6.358387	0.0000
TPRICE(-17)	-0.017174	0.011767	-1.459563	0.1446
D2*LOG(ACTLOAD)	0.003542	0.000645	5.491956	0.0000
TPRICE(-2)	-0.079000	0.033118	-2.385411	0.0172
TPRICE(-43)	-0.063304	0.013382	-4.730514	0.0000
LOG(FORCLOAD)	0.110946	0.040737	2.723457	0.0065

In this case a total of 59 of the 107 search regressors were selected.

The bottom of the output (partially shown) describes the selection process the stepwise regression undertook.

Selection Summary
Added TPRICE(-1)
Added LOG(ACTLOAD)
Added TPRICE(-4)
Added TPRICE(-20)
Added TPRICE(-30)
Added LOG(ACTLOAD(-24))
Added TPRICE(-18)
Added D2*LOG(ACTLOAD)
Added TPRICE(-2)
Added TPRICE(-19)

Swapwise Example

For a second estimation of the model, we will use the swapwise algorithm. We again click on **Quick\Estimation Equation** to bring up the **Equation Estimation** dialog, and fill in the variables and sample as before. We change the **Selection method** combo to **Swapwise**.

Switching to the **Options** tab, we switch to **Min R-squared increment** and instruct EViews to find the best 60 regressors (which seems a reasonable number given the stepwise procedure's selection of 59).

The output from this equation is similar to that of the first—the top portion displays a summary of the estimation, the middle section provides the estimation results, and the bottom portion displays the selection process.

Auto-search/GETS Example

Our third estimation will use the Auto-search/GETS algorithm. On the **Specification** page, change the **Selection method** to **Auto-Search/GETS**, and then click on the **Options** tab to alter some of the default settings. We'll change the criteria used to decide between the final candidate models to the Schwarz criterion using the **Criterion** dropdown. We also elect to only use the AR LM test and PET tests as diagnostic tests during the path search.

The output from this estimation is similar to the previous two; the upper portion describes the selection method, the middle displays the final model's coefficient estimates and associates statistics, and the bottom provides some small detail on the selection process. In contrast to the stepwise procedure, Auto-search has only selected 27 regressors.

Lasso Example

In this example we will reproduce an example from *Statistical Learning with Sparsity*, by Trevor Hastie, Robert Tibshirani, and Martin Wainwright.

Section 2.2, Table 2.2 contains the results of a short analysis on a set of crime data. The dataset consists of six variables, with the dependent variable being the total overall reported crime rate per one million residents in fifty US cities. The independent variables are annual police funding in dollars per resident, percent of people 25 years and older with four years of high school, percent of 16- to 19-year olds not in high school and not high school graduates, percent of 18- to 24-year olds in college, and percent of people 25 years and older with at least four years of college.

Table 2.2 *Results from analysis of the crime data. Left panel shows the least-squares estimates, standard errors, and their ratio (Z-score). Middle and right panels show the corresponding results for the lasso, and the least-squares estimates applied to the subset of predictors chosen by the lasso.*

	LS coef	SE	Z	Lasso	SE	Z	LS	SE	Z
`funding`	10.98	3.08	3.6	8.84	3.55	2.5	11.29	2.90	3.9
`hs`	-6.09	6.54	-0.9	-1.41	3.73	-0.4	-4.76	4.53	-1.1
`not-hs`	5.48	10.05	0.5	3.12	5.05	0.6	3.44	7.83	0.4
`college`	0.38	4.42	0.1	0.0	-	-	0.0	-	-
`college4`	5.50	13.75	0.4	0.0	-	-	0.0	-	-

For the first example we will replicate the leftmost side of Table 2.2. This is a simple OLS calculation and using LS in EViews we get an exact match on the coefficients with standard errors. We can see that the "FUNDING" variable has the biggest influence on the overall crime rate:

Equation: LS Workfile: CRIME_DATA::Untitled\

View | Proc | Object | Print | Name | Freeze | Estimate | Forecast | Stats | Resids

Dependent Variable: CRIME_RATE
Method: Least Squares
Date: 09/21/20 Time: 16:34
Sample: 1 50
Included observations: 50

Variable	Coefficient	Std. Error	t-Statistic	Prob.
C	489.6486	472.3659	1.036587	0.3056
FUNDING	10.98067	3.077784	3.567720	0.0009
HIGHSCHOOL	-6.088529	6.543685	-0.930443	0.3572
NOT_HS	5.480304	10.05350	0.545114	0.5884
COLLEGE	0.377044	4.417396	0.085354	0.9324
COLLEGE4	5.500471	13.75391	0.399921	0.6911

R-squared	0.333603	Mean dependent var	717.9600
Adjusted R-squared	0.257876	S.D. dependent var	293.9388
S.E. of regression	253.2184	Akaike info criterion	14.01855
Sum squared resid	2821259.	Schwarz criterion	14.24799
Log likelihood	-344.4637	Hannan-Quinn criter.	14.10592
F-statistic	4.405336	Durbin-Watson stat	1.555652
Prob(F-statistic)	0.002444		

For the second part we will fit a lasso model to the same data. The comparison is with the middle set of coefficients in Table 2.2. In EViews we choose the ENET procedure with the dependent, a constant, and all five independent variables. We select the lasso penalty and leave the lambda field blank to automatically generate a lambda sequence and perform cross-validation. We also standardize the independent variables and use 10-fold cross-validation. These resulting coefficients are not an exact match because of the randomness associated with cross-validation, but relative magnitudes and signs are the same. They also match our intuition about lasso compared to OLS. The coefficients have shrunk toward zero, and in fact two of them are zero at the minimum value of lambda chosen by cross-validation. Again, the FUNDING variable has the greatest influence:

Equation: LASSO Workfile: CRIME_DATA::Untitled\

View | Proc | Object | Print | Name | Freeze | Estimate | Forecast | Stats | Resids

Dependent Variable: CRIME_RATE
Method: Elastic Net Regularization
Date: 09/21/20 Time: 17:42
Sample: 1 50
Included observations: 50
Penalty type: Lasso (alpha = 1)
Lambda at minimum error: 18.26
Regressor transformation: Std Dev (pop)
Cross-validation method: K-Fold (number of folds = 10), rng=kn, seed=1335074198
Selection measure: Mean Squared Error

	(minimum)	(+ 1 SE)	(+ 2 SE)
Lambda	18.26	80.9	155.2

Variable	Coefficients		
C	457.9997	513.0186	717.9600
FUNDING	9.800651	5.427473	0.000000
HIGHSCHOOL	-2.723965	0.000000	0.000000
NOT_HS	3.250446	0.000000	0.000000
COLLEGE	0.000000	0.000000	0.000000
COLLEGE4	0.000000	0.000000	0.000000
d.f.	3	1	0
L1 Norm	473.7747	518.4461	717.9600
R-squared	0.318582	0.207010	0.000000

The last comparison is with the rightmost column in Table 2.2. These values are OLS applied to the variables selected (the nonzero coefficients) from the lasso fit. In EViews we choose the VARSEL procedure with the five independent variables as search variables. The ENET options are the same as those for the lasso analysis earlier. As expected, the two variables COLLEGE and COLLEGE4 that were zero in the lasso model are not included in the subsequent OLS regression. While the lasso shrank, or biased, the coefficients toward zero, the OLS fit expands, or de-biases, them away from zero. This results in a decrease in the variance of the final model, as you can see by comparing the errors in the variable selection model with the first OLS model:

Equation: VARSEL Workfile: CRIME_DATA::Untitled\

View | Proc | Object | Print | Name | Freeze | Estimate | Forecast | Stats | Resids

Dependent Variable: CRIME_RATE
Method: Variable Selection
Date: 09/21/20 Time: 17:46
Sample: 1 50
Included observations: 50
Number of always included regressors: 1
Number of search regressors: 5
Selection method: Lasso
Lambda at minimum AIC: 10.4

Variable	Coefficient	Std. Error	t-Statistic	Prob.*
C	518.7457	324.1003	1.600572	0.1163
FUNDING	11.28598	2.899277	3.892689	0.0003
HIGHSCHOOL	-4.760741	4.529800	-1.050982	0.2988
NOT_HS	3.440721	7.828946	0.439487	0.6624

R-squared	0.327542	Mean dependent var	717.9600
Adjusted R-squared	0.283686	S.D. dependent var	293.9388
S.E. of regression	248.7761	Akaike info criterion	13.94760
Sum squared resid	2846918.	Schwarz criterion	14.10056
Log likelihood	-344.6900	Hannan-Quinn criter.	14.00585
F-statistic	7.468584	Durbin-Watson stat	1.567344
Prob(F-statistic)	0.000356		

Selection Summary

Number of models compared:	100

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

We can examine some of the details behind the model selection process by going to **View/ Model Selection Summary** and choosing **Criteria Graph** and **Criteria Table**. Here we can see the models ranked from best to worst in order of the AIC. Note that the selection process is different from the rankings of the best models in ENET's cross-validation, which is based on error selection measures such as mean squared error.

Equation: VARSEL Workfile: CRIME_DATA::Untitled\

View | Proc | Object | Print | Name | Freeze | Estimate | Forecast | Stats | Resids

Model Selection Criteria Table
Dependent Variable: CRIME_RATE
Date: 09/22/20 Time: 10:41
Sample: 1 50
Included observations: 50

Model	LogL	AIC*	BIC	HQ	Specification (Lambda)
30	-344.798880	0.795392	0.910113	0.839078	10.4
29	-344.821078	0.795992	0.910713	0.839678	11.5
28	-344.847791	0.796715	0.911436	0.840401	12.6
27	-344.879928	0.797585	0.912306	0.841272	13.8
26	-344.918583	0.798634	0.913355	0.842320	15.2
25	-344.965064	0.799896	0.914618	0.843583	16.6
24	-345.020936	0.801418	0.916139	0.845104	18.3
23	-345.088068	0.803250	0.917971	0.846936	20
22	-345.168691	0.805457	0.920178	0.849143	22
14	-346.646581	0.807200	0.883681	0.836324	46.3
21	-345.265457	0.808115	0.922837	0.851802	24.1
10	-348.042938	0.808973	0.847213	0.823535	67.2
20	-345.381517	0.811317	0.926039	0.855004	26.5
13	-346.916458	0.815092	0.891573	0.844217	50.8
19	-345.520600	0.815174	0.929895	0.858860	29.1
18	-345.687106	0.819819	0.934541	0.863506	31.9
9	-348.394640	0.819867	0.858108	0.834429	73.7
12	-347.237704	0.824599	0.901080	0.853724	55.8
17	-345.886201	0.825415	0.940136	0.869101	35
37	-344.570448	0.829249	0.982211	0.887498	5.45
36	-344.591380	0.829809	0.982771	0.888058	5.98
35	-344.616570	0.830485	0.983446	0.888733	6.56
34	-344.646877	0.831298	0.984260	0.889547	7.2

In **View/Model Selection Summary** we have also included some of the views from ENET. The selected models here, highlighted in red in the **Coefficient Matrix** and **Summary Path**

and with the dotted line in the **Lambda Coefficient Graph**, are based on AIC rather than MSE.

	Lambda	d.f.	L1Norm	R-squared	AIC
14	46.29198	2	401.6294	0.272800	0.807200
15	42.17952	3	389.3258	0.279727	0.840273
16	38.43241	3	402.5544	0.287845	0.832155
17	35.01818	3	414.6077	0.294585	0.825415
18	31.90726	3	425.5902	0.300181	0.819819
19	29.07271	3	435.5971	0.304826	0.815174
20	26.48997	3	444.7150	0.308683	0.811317
21	24.13667	3	453.0229	0.311885	0.808115
22	21.99244	3	460.5928	0.314543	0.805457
23	20.03869	3	467.4901	0.316750	0.803250
24	18.25851	3	473.7747	0.318582	0.801418
25	16.63647	3	479.5011	0.320104	0.799896
26	15.15854	3	484.7186	0.321366	0.798634
27	13.81189	3	489.4727	0.322415	0.797585
28	12.58488	3	493.8045	0.323285	0.796715
29	11.46688	3	497.7514	0.324008	0.795992
30	10.44819	3	501.3477	0.324608	0.795392
31	9.520002	4	504.6357	0.325121	0.834879
32	8.674271	4	508.4125	0.326542	0.833458
33	7.903673	4	511.8537	0.327722	0.832278
34	7.201532	4	514.9893	0.328702	0.831298
35	6.561768	4	517.8463	0.329515	0.830485
36	5.978839	4	520.4495	0.330191	0.829809
37	5.447695	4	522.8214	0.330751	0.829249
38	4.963737	5	524.2500	0.331222	0.868778
39	4.522772	5	523.7015	0.331626	0.868374
40	4.120982	5	523.2017	0.331962	0.868038
41	3.754885	5	522.7463	0.332240	0.867760
42					

References

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.

Escribano, Alvaro and Sucarrat, Genaro (2011). "Automated model selection in finance: General-to-specific modelling of the mean and volatility specifications," *Oxford Bulletin of Economics and Statistics*, 75, 716–735.

Friedman, J., Hastie, T., & Tibshirani, R. (2001). *The Elements of Statistical Learning* (Vol. 1, No. 10). New York: Springer Series in Statistics.

Friedman, J., Hastie, T., & Tibshirani, R. (2010). Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33(1), 1.

Hastie, T., Tibshirani, R., & Wainwright, M. (2015). Statistical learning with sparsity: the lasso and generalizations. CRC press.

Hoover, K. D. and S. J. Perez (1999). "Data Mining Reconsidered: Encompassing and the General-to-Specific Approach to Specification Search," *Econometrics Journal*, 2, 167–191.

Hurvich, C. M. and C. L. Tsai (1990). "The Impact of Model Selection on Inference in Linear Regression," *American Statistician*, 44, 214–217.

Roecker, E. B. (1991). "Prediction Error and its Estimation for Subset-Selection Models," *Technometrics*, 33, 459–469.

Uniejewski B., Nowotarski, J., and Weron, R (2016) "Automated variable selection and shrinkage for day-ahead electricity price forecasting," mimeo.

Zou, H., & Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2), 301-320.

Chapter 23. Instrumental Variables and GMM

This chapter describes EViews tools for estimating a single equation using Two-stage Least Squares (TSLS), Limited Information Maximum Likelihood (LIML) and K-Class Estimation, and Generalized Method of Moments (GMM).

There are countless references for the techniques described in this chapter. Notable textbook examples include Hayashi (2000), Hamilton (1994), Davidson and MacKinnon (1993). Less technical treatments may be found in Stock and Watson (2007) and Johnston and DiNardo (1997).

Background

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.

There are many different approaches to using instruments to eliminate the effect of variable and residual correlation. EViews offers three basic types of instrumental variable estimators: Two-stage Least Squares (TSLS), Limited Information Maximum Likelihood and K-Class Estimation (LIML), and Generalized Method of Moments (GMM).

Two-stage Least Squares

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. The linear TSLS objective function is given by:

$$\Psi(\beta) = (y - X\beta)' Z(Z'Z)^{-1} Z'(y - X\beta) \tag{23.1}$$

Then the coefficients computed in two-stage least squares are given by,

$$b_{TSLS} = (X'Z(Z'Z)^{-1} Z'X)^{-1} X'Z(Z'Z)^{-1} Z'y, \tag{23.2}$$

and the standard estimated covariance matrix of these coefficients may be computed using:

$$\hat{\Sigma}_{TSLS} = s^2 (X'Z(Z'Z)^{-1} Z'X)^{-1}, \tag{23.3}$$

where s^2 is the estimated residual variance (square of the standard error of the regression). If desired, s^2 may be replaced by the non-d.f. corrected estimator. Note also that EViews offers both White and HAC covariance matrix options for two-stage least squares.

Estimating TSLS in EViews

To estimate an equation using Two-stage Least Squares, open the equation specification box by choosing **Object/New Object.../Equation...** or **Quick/Estimate Equation...** Choose **TSLS** from the **Method:** dropdown menu and the dialog will change to include an edit window where you will list the instruments.

Alternately, type the `tsls` keyword in the command window and hit ENTER.

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

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.
- EViews will, by default, add a constant to the instrument list. If you do not wish a constant to be added to the instrument list, the **Include a constant** check box should be unchecked.

To illustrate the estimation of two-stage least squares, we use an example from Stock and Watson 2007 (p. 438), which estimates the demand for cigarettes in the United States in 1995. (The data are available in the workfile "Sw_cig.WF1".) The dependent variable is the per capita log of packs sold LOG(PACKPC). The exogenous variables are a constant, C, and the log of real per capita state income LOG(PERINC). The endogenous variable is the log of real after tax price per pack LOG(RAVGPRC). The additional instruments are average state sales tax RTAXSO, and cigarette specific taxes RTAXS. Stock and Watson use the White covariance estimator for the standard errors.

The equation specification is then,

```
log(packpc) c log(ravgprs) log(perinc)
```

and the instrument list is:

```
c log(perinc) rtaxso rtaxs
```

This specification satisfies the order condition for identification, which requires that there are at least as many instruments (four) as there are coefficients (three) in the equation specification. Note that listing C as an instrument is redundant, since by default, EViews automatically adds it to the instrument list.

To specify the use of White heteroskedasticity robust standard errors, we will select **White** in the **Coefficient covariance matrix** dropdown menu on the **Options** tab. By default, EViews will estimate the using the **Ordinary** method with **d.f. Adjustment** as specified in Equation (23.3).

Output from TSLS

Below we show the output from a regression of LOG(PACKPC) on a constant and LOG(RAVGPRS) and LOG(PERINC), with instrument list "LOG(PERINC) RTAXSO RTAXS".

Dependent Variable: LOG(PACKPC)
Method: Two-Stage Least Squares
Date: 04/15/09 Time: 14:17
Sample: 1 48
Included observations: 48
White heteroskedasticity-consistent standard errors & covariance
Instrument specification: LOG(PERINC) RTAXSO RTAXS
Constant added to instrument list

Variable	Coefficient	Std. Error	t-Statistic	Prob.
C	9.894956	0.959217	10.31566	0.0000
LOG(RAVGPRS)	-1.277424	0.249610	-5.117680	0.0000
LOG(PERINC)	0.280405	0.253890	1.104436	0.2753

R-squared	0.429422	Mean dependent var	4.538837
Adjusted R-squared	0.404063	S.D. dependent var	0.243346
S.E. of regression	0.187856	Sum squared resid	1.588044
F-statistic	13.28079	Durbin-Watson stat	1.946351
Prob(F-statistic)	0.000029	Second-Stage SSR	1.845868
Instrument rank	4	J-statistic	0.311833
Prob(J-statistic)	0.576557		

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 formulae 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).

Three other summary statistics are reported: "Instrument rank", the "J-statistic" and the "Prob(J-statistic)". The Instrument rank is simply the rank of the instrument matrix, and is equal to the number of instruments used in estimation. The J-statistic is calculated as:

$$\frac{1}{T} u' Z \left(s^2 Z' Z / T \right)^{-1} Z' u \tag{23.4}$$

where u are the regression residuals. See "Generalized Method of Moments," beginning on page 101 for additional discussion of the J-statistic.

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

$$s^2 = \sum_t u_t^2 / (T - k) . \tag{23.5}$$

These structural, or regression, 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 $\tilde{u}_t = y_t - x_t' b_{TSLS}$, where the y_t and 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.

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 using the default settings with a tighter convergence tolerance looks as follows:

Dependent Variable: LOG(PACKPC)
Method: Two-Stage Least Squares
Date: 08/25/09 Time: 15:04
Sample (adjusted): 2 48
Included observations: 47 after adjustments
White heteroskedasticity-consistent standard errors & covariance
Instrument specification: LOG(PERINC) RTAXSO RTAXS
Constant added to instrument list
Lagged dependent variable & regressors added to instrument list

Variable	Coefficient	Std. Error	t-Statistic	Prob.
C	10.02006	0.996752	10.05272	0.0000
LOG(RAVGPRS)	-1.309245	0.271683	-4.819022	0.0000
LOG(PERINC)	0.291047	0.290818	1.000785	0.3225
AR(1)	0.026532	0.133425	0.198852	0.8433

R-squared	0.431689	Mean dependent var	4.537196
Adjusted R-squared	0.392039	S.D. dependent var	0.245709
S.E. of regression	0.191584	Sum squared resid	1.578284
Durbin-Watson stat	1.951380	Instrument rank	7
J-statistic	1.494632	Prob(J-statistic)	0.683510

Inverted AR Roots	.03

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:

$$\begin{aligned} y_t &= x_t'\beta + w_t\gamma + u_t \\ u_t &= \rho_1 u_{t-1} + \epsilon_t \end{aligned} \tag{23.6}$$

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 cited 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:

$$(w_t, z_t, y_{t-1}, x_{t-1}, w_{t-1}) . \tag{23.7}$$

EViews estimates the model as a nonlinear regression model so that *Fair's warning does not apply*. Estimation of the model does, however, require specification of additional instruments to satisfy the instrument order condition for the transformed specification. By default, the first-stage instruments employed in TSLS are formed as if one were running Cochrane-Orcutt using Fair's prescription. Thus, if you omit the lagged left- and right-hand side terms from the instrument list, EViews will, by default, automatically add the lagged terms as instruments. This addition will be noted in your output.

You may instead instruct EViews not to add the lagged left- and right-hand side terms as instruments. In this case, you are responsible for adding sufficient instruments to ensure the order condition is satisfied.

Higher Order AR errors

The AR(1) results extend 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:

$$(w_t, z_t, y_{t-4}, x_{t-4}, w_{t-4}) \tag{23.8}$$

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

$$(w_t, z_t, y_{t-1}, \ldots, y_{t-4}, x_{t-1}, \ldots, x_{t-4}, w_{t-1}, \ldots, w_{t-4}) \tag{23.9}$$

Note that while conceptually 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.

In this case, you may wish to turn off the automatic lag instrument addition and handle the additional instrument specification directly.

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 to 4)
```

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

```
c gov log(m1) 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(m1) 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.

Recall that by default, EViews augments the instrument list by adding lagged dependent and regressor variables corresponding to the AR lags. 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.

Of course, you may instruct EViews not to add the extra instruments. In this case, you are responsible for adding enough instruments to ensure the instrument order condition is satisfied.

Nonlinear Two-stage Least Squares

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:

$$y_t = f(x_t, \beta) + \epsilon_t, \tag{23.10}$$

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:

$$\Psi(\beta) = (y - f(X, \beta))' Z(Z'Z)^{-1} Z'(y - f(X, \beta)) \tag{23.11}$$

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:

$$G(\beta)' Z(Z'Z)^{-1} Z'(y - f(X, \beta)) = 0 \tag{23.12}$$

with estimated covariance given by:

$$\hat{\Sigma}_{TSNLLS} = s^2 (G(b_{TSNLLS})' Z(Z'Z)^{-1} Z' G(b_{TSNLLS}))^{-1}. \tag{23.13}$$

How to Estimate Nonlinear TSLS in EViews

To estimate a Nonlinear equation using TSLS simply select **Object/New Object…/Equation…** or **Quick/Estimate Equation…** Choose **TSLS** from the **Method** dropdown menu, 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 the derivatives $G(\beta)$. 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.

Nonlinear Two-stage Least Squares with ARMA errors

While we will not go into much detail here, note that EViews can estimate non-linear TSLS models where there are ARMA error terms.

To estimate your model, simply open your equation specification window, and enter your nonlinear specification, including all ARMA 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), ma(1)=c(4)]
```

with the instrument list:

```
c gov
```

EViews will transform the nonlinear regression model as described in "Specifying AR Terms" on page 132, and then estimate nonlinear TSLS on the transformed specification. For nonlinear models with AR errors, EViews uses a Gauss-Newton algorithm. See "Optimization Algorithms" on page 1319 for further details.

Weighted Nonlinear Two-stage Least Squares

Weights may be used in nonlinear two-stage least squares estimation, provided there are no ARMA terms. Simply add weighting to your nonlinear TSLS specification above by pressing the **Options** button and entering the weight specification (see "Weighted Least Squares" on page 47).

The objective function for weighted TSLS is,

$$\Psi(\beta) = (y - f(X, \beta))' W' Z(Z' WZ)^{-1} Z' W'(y - f(X, \beta)) . \tag{23.14}$$

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

$$\hat{\Sigma}_{WTSNLLS} = s^2 (G(b)' WZ(Z' WZ)^{-1} Z' WG(b))^{-1} \tag{23.15}$$

where $b \equiv b_{WTSNLLS}$.

Limited Information Maximum Likelihood and K-Class Estimation

Limited Information Maximum Likelihood (LIML) is a form of instrumental variable estimation that is quite similar to TSLS. As with TSLS, LIML uses instruments to rectify the prob-

lem where one or more of the right hand side variables in the regression are correlated with residuals.

LIML was first introduced by Anderson and Rubin (1949), prior to the introduction of two-stage least squares. However traditionally TSLS has been favored by researchers over LIML as a method of instrumental variable estimation. If the equation is exactly identified, LIML and TSLS will be numerically identical. Recent studies (for example, Hahn and Inoue 2002) have, however, found that LIML performs better than TSLS in situations where there are many "weak" instruments.

The linear LIML estimator minimizes

$$\Psi(\beta) = T\frac{(y - X\beta)'Z(Z'Z)^{-1}Z'(y - X\beta)}{(y - X\beta)'(y - X\beta)} \tag{23.16}$$

with respect to β, where y is the dependent variable, X are explanatory variables, and Z are instrumental variables.

Computationally, it is often easier to write this minimization problem in a slightly different-form. Let $W = (y, X)$ and $\tilde{\beta} = (-1, \beta)'$. Then the linear LIML objective function can be written as:

$$\Psi(\beta) = T\frac{\tilde{\beta}'W'Z(Z'Z)^{-1}Z'W\tilde{\beta}}{\tilde{\beta}'W'W\tilde{\beta}} \tag{23.17}$$

Let λ be the smallest eigenvalue of $(W'W)^{-1}W'Z(Z'Z)^{-1}Z'W$. The LIML estimator of $\tilde{\beta}$ is the eigenvector corresponding to λ, with a normalization so that the first element of the eigenvector equals -1.

The non-linear LIML estimator maximizes the concentrated likelihood function:

$$L = -\frac{T}{2}(\log(u'u) + \log\left|X'AX - X'AZ(Z'AZ)^{-1}Z'AX\right|) \tag{23.18}$$

where $u_t = y_t - f(X_t, \beta)$ are the regression residuals and $A = I - u(u'u)^{-1}u'$.

The default estimate of covariance matrix of instrumental variables estimators is given by the TSLS estimate in Equation (23.3).

K-Class

K-Class estimation is a third form of instrumental variable estimation; in fact TSLS and LIML are special cases of K-Class estimation. The *linear* K-Class objective function is, for a fixed k, given by:

$$\Psi(\beta) = (y - X\beta)'(I - kM_Z)(y - X\beta) \tag{23.19}$$

The corresponding K-Class estimator may be written as:

$$\beta_k = (X'(I - kM_Z)X)^{-1}X'(I - kM_Z)y \tag{23.20}$$

where $P_Z = Z(Z'Z)^{-1}Z'$ and $M_Z = I - Z(Z'Z)^{-1}Z' = I - P_Z$.

If $k = 1$, then the K-Class estimator is the TSLS estimator. If $k = 0$, then the K-Class estimator is OLS. LIML is a K-Class estimator with $k = \lambda$, the minimum eigenvalue described above.

The obvious K-Class covariance matrix estimator is given by:

$$\hat{\Sigma}_k = s^2(X'(I - kM_Z)X)^{-1} \tag{23.21}$$

Bekker (1994) offers a covariance matrix estimator for K-Class estimators with normal error terms that is more robust to weak instruments. The Bekker covariance matrix estimate is given by:

$$\tilde{\Sigma}_{BEKK} = H^{-1}\tilde{\Sigma}H^{-1} \tag{23.22}$$

where

$$\begin{aligned} H &= X'P_ZX - \alpha(X'X) \\ \tilde{\Sigma} &= s^2((1-\alpha)^2\tilde{X}'P_Z\tilde{X} + \alpha^2\tilde{X}'M_Z\tilde{X}) \end{aligned} \tag{23.23}$$

for

$$\alpha = \frac{u'P_Zu}{u'u} \text{ and } \tilde{X} = X - \frac{uu'X}{u'u}.$$

Hansen, Hausman and Newey (2006) offer an extension to Bekker's covariance matrix estimate for cases with non-normal error terms.

Estimating LIML and K-Class in EViews

To estimate a LIML or K-Class equation in EViews, create an equation by choosing **Object/New Object…/Equation…** or **Quick/Estimate Equation**, and choose **LIML** from the **Method** box.

Alternately, you may enter the keyword `liml` in the command window then hit ENTER.

In the **Equation specification** edit box, specify your dependent variable and exogenous variables, and in the **Instrument list** edit box, provide a list of instruments. Endogenous variables should be entered in both the **Equation specification** box and the **Instrument list** box.

For K-Class estimation, enter the value of k in the box labeled **K (leave blank for LIML)**. If no value is entered in this box, LIML is performed.

If you wish to estimate a non-linear equation, then enter the expression for the non-linear equation in the **Equation specification** box. Note that non-linear K-Class estimation is currently not permitted; only non-linear LIML may be performed.

If you do not wish to include a constant as one of the instruments, uncheck the **Include a Constant** checkbox.

Different standard error calculations may be chosen by changing the **Standard Errors** drop-down menu on the **Options** tab of the estimation dialog. Note that if your equation was non-linear, only IV based standard errors may be calculated. For linear estimation you may also choose K-Class based, Bekker, or Hansen, Hausman and Newey standard errors.

As an example of LIML estimation, we estimate part of Klein's Model I, as published in Greene (2008, p. 385). We estimate the Consumption equation, where consumption (CONS) is regressed on a constant, private profits (Y), lagged private profits (Y(-1)), and wages (W) using data in the workfile "Klein.WF1". The instruments are a constant, lagged corporate profits (P(-1)), lagged capital stock (K(-1)), lagged GNP (X(-1)), a time trend (TM), Government wages (WG), Government spending (G) and taxes (T). In his reproduction of the Klein model, Greene uses K-Class standard errors. The results of this estimation are as follows:

Dependent Variable: CONS
Method: LIML / K-Class
Date: 05/27/09 Time: 11:16
Sample (adjusted): 1921 1941
Included observations: 21 after adjustments
Covariance type: K-Class
Instrument specification: C P(-1) K(-1) X(-1) TM WG G T

Variable	Coefficient	Std. Error	t-Statistic	Prob.
C	17.14765	1.840295	9.317882	0.0000
Y	-0.222513	0.201748	-1.102927	0.2854
Y(-1)	0.396027	0.173598	2.281293	0.0357
W	0.822559	0.055378	14.85347	0.0000

R-squared	0.956572	Mean dependent var	53.99524
Adjusted R-squared	0.948909	S.D. dependent var	6.860866
S.E. of regression	1.550791	Sum squared resid	40.88419
Durbin-Watson stat	1.487859	LIML min. eigenvalue	1.498746

EViews identifies the LIML estimation procedure, along with the choice of covariance matrix type and the list of instruments in the header. This information is followed by the usual coefficient, t-statistics, and asymptotic p-values.

The standard summary statistics reported at the bottom of the table are computed using the formulae outlined in "Summary Statistics" on page 13. Along with the standard statistics, the LIML minimum eigenvalue is also reported, if the estimation type was LIML.

Generalized Method of Moments

We offer here a brief description of the Generalized Method of Moments (GMM) estimator, paying particular attention to issues of weighting matrix estimation and coefficient covariance calculation. Or treatment parallels the excellent discussion in Hayashi (2000). Those interested in additional detail are encouraged to consult one of the many comprehensive surveys of the subject.

The GMM Estimator

The starting point of GMM estimation is the assumption that there are a set of L moment conditions that the K-dimensional parameters of interest, β should satisfy. These moment conditions can be quite general, and often a particular model has more specified moment conditions than parameters to be estimated. Thus, the vector of $L \geq K$ moment conditions may be written as:

$$E(m(y_t, \beta)) = 0 . \tag{23.24}$$

In EViews (as in most econometric applications), we restrict our attention to moment conditions that may be written as an orthogonality condition between the residuals of an equation, $u_t(\beta) = u(y_t, X_t, \beta)$, and a set of K instruments Z_t:

$$E(Z_t u_t(\beta)) = 0 \tag{23.25}$$

The traditional Method of Moments estimator is defined by replacing the moment conditions in Equation (23.24) with their sample analog:

$$m_T(\beta) = \frac{1}{T}\sum_t Z_t u_t(\beta) = \frac{1}{T} Z' u(\beta) = 0 \tag{23.26}$$

and finding the parameter vector β which solves this set of L equations.

When there are more moment conditions than parameters ($L > K$), the system of equations given in Equation (23.26) may not have an exact solution. Such as system is said to be *overidentified*. Though we cannot generally find an exact solution for an overidentified system, we can reformulate the problem as one of choosing a β so that the sample moment $m_T(\beta)$ is as "close" to zero as possible, where "close" is defined using the quadratic form:

$$\begin{aligned} J(\beta, \hat{W}_T) &= T\, m_T(\beta)' \hat{W}_T^{-1} m_T(\beta) \\ &= \frac{1}{T} u(\beta)' Z\, \hat{W}_T^{-1} Z' u(\beta) \end{aligned} \tag{23.27}$$

as a measure of distance. The possibly random, symmetric and positive-definite $L \times L$ matrix $\hat{W}_T$ is termed the *weighting matrix* since it acts to weight the various moment conditions in constructing the distance measure. The *Generalized* Method of Moments estimate is defined as the β that minimizes Equation (23.27).

As with other instrumental variable estimators, for the GMM estimator to be identified, there must be at least as many instruments as there are parameters in the model. In models where there are the same number of instruments as parameters, the value of the optimized objective function is zero. If there are more instruments than parameters, the value of the optimized objective function will be greater than zero. In fact, the value of the objective function, termed the *J*-statistic, can be used as a test of over-identifying moment conditions.

Under suitable regularity conditions, the GMM estimator is consistent and $\sqrt{T}$ asymptotically normally distributed,

$$\sqrt{T}(\hat{\beta} - \beta_0) \rightarrow N(0, V) \tag{23.28}$$

The asymptotic covariance matrix V of $\sqrt{T}(\hat{\beta} - \beta_0)$ is given by

$$V = (\Sigma' W^{-1}\Sigma)^{-1} \cdot \Sigma' W^{-1} S W^{-1} \Sigma \cdot (\Sigma' W^{-1}\Sigma)^{-1} \tag{23.29}$$

for

$$\begin{aligned} W &= \text{plim } \hat{W}_T \\ \Sigma &= \text{plim } \frac{1}{T} Z' \nabla u(\beta) \\ S &= \text{plim } \frac{1}{T} Z' u(\beta) u(\beta)' Z \end{aligned} \tag{23.30}$$

where S is both the asymptotic variance of $\sqrt{T} m_T(\hat{\beta})$ and the long-run covariance matrix of the vector process $\{ Z_t u_t(\beta) \}$.

In the leading case where the $u_t(\beta)$ are the residuals from a linear specification so that $u_t(\beta) = y_t - X_t'\beta$, the GMM objective function is given by

$$J(\beta, \hat{W}_T) = \frac{1}{T}(y - X\beta)' Z \hat{W}_T^{-1} Z'(y - X\beta) \tag{23.31}$$

and the GMM estimator yields the unique solution $\hat{\theta} = (X' Z \hat{W}_T^{-1} Z' X)^{-1} X' Z \hat{W}_T^{-1} Z' y$. The asymptotic covariance matrix is given by Equation (23.27), with

$$\Sigma = \text{plim } \frac{1}{T}(Z' X) \tag{23.32}$$

It can be seen from this formation that both two-stage least squares and ordinary least squares estimation are both special cases of GMM estimation. The two-stage least squares objective is simply the GMM objective function multiplied by $\hat{\sigma}^2$ using weighting matrix $\hat{W}_T = (\hat{\sigma}^2 Z' Z / T)$. Ordinary least squares is equivalent to two-stage least squares objective with the instruments set equal to the derivatives of $u_t(\beta)$, which in the linear case are the regressors.

Choice of Weighting Matrix

An important aspect of specifying a GMM estimator is the choice of the weighting matrix, $\hat{W}_T$. While any sequence of symmetric positive definite weighting matrices $\hat{W}_T$ will yield a consistent estimate of β, Equation (23.29) implies that the choice of $\hat{W}_T$ affects the asymptotic variance of the GMM estimator. Hansen (1992) shows that an *asymptotically efficient*, or *optimal* GMM estimator of β may be obtained by choosing $\hat{W}_T$ so that it converges to the inverse of the long-run covariance matrix S:

$$\text{plim } \hat{W}_T = S \tag{23.33}$$

Intuitively, this result follows since we naturally want to assign less weight to the moment conditions that are measured imprecisely. For a GMM estimator with an optimal weighting matrix, the asymptotic covariance matrix of $\hat{\beta}$ is given by

$$\begin{aligned} V &= (\Sigma' S^{-1} \Sigma)^{-1} \cdot \Sigma' S^{-1} S S^{-1} \Sigma \cdot (\Sigma' S \Sigma)^{-1} \\ &= (\Sigma' S^{-1} \Sigma)^{-1} \end{aligned} \tag{23.34}$$

Implementation of optimal GMM estimation requires that we obtain estimates of S^{-1}. EViews offers four basic methods for specifying a weighting matrix:

- **Two-stage least squares**: the two-stage least squares weighting matrix is given by $\hat{W}_T = (\hat{\sigma}^2 Z'Z/T)$ where $\hat{\sigma}^2$ is an estimator of the residual variance based on an initial estimate of β. The estimator for the variance will be s^2 or the no d.f. corrected equivalent, depending on your settings for the coefficient covariance calculation.
- **White**: the White weighting matrix is a heteroskedasticity consistent estimator of the long-run covariance matrix of $\{Z_t u_t(\beta)\}$ based on an initial estimate of β.
- **HAC - Newey-West**: the HAC weighting matrix is a heteroskedasticity and autocorrelation consistent estimator of the long-run covariance matrix of $\{Z_t u_t(\beta)\}$ based on an initial estimate of β.
- **User-specified**: this method allows you to provide your own weighting matrix (specified as a sym matrix containing a scaled estimate of the long-run covariance $\hat{U} = T\hat{S}$).

For related discussion of the **White** and **HAC - Newey West** robust standard error estimators, see "Robust Standard Errors" on page 32.

Weighting Matrix Iteration

As noted above, both the White and HAC weighting matrix estimators require an initial consistent estimate of β. (Technically, the two-stage least squares weighting matrix also requires an initial estimate of β, though these values are irrelevant since the resulting $\hat{\sigma}^2$ does not affect the resulting estimates).

Accordingly, computation of the optimal GMM estimator with White or HAC weights often employs a variant of the following procedure:

1. Calculate initial parameter estimates $\hat{\beta}_0$ using TSLS
2. Use the $\hat{\beta}_0$ estimates to form residuals $u_t(\hat{\beta}_0)$
3. Form an estimate of the long-run covariance matrix of $\{Z_t u_t(\hat{\beta}_0)\}$, $\hat{S}_T(\hat{\beta}_0)$, and use it to compute the optimal weighting matrix $\hat{W}_T = \hat{S}_T(\hat{\beta}_0)$
4. Minimize the GMM objective function with weighting matrix $\hat{W}_T = \hat{S}_T(\hat{\beta}_0)$

$$J(\beta_1, \hat{\beta}_0) = \frac{1}{T} u(\beta_1)' Z \hat{S}_T(\hat{\beta}_0)^{-1} Z' u(\beta_1) \tag{23.35}$$

with respect to β_1 to form updated parameter estimates.

We may generalize this procedure by repeating steps 2 through 4 using $\hat{\beta}_1$ as our initial parameter estimates, producing updated estimates $\hat{\beta}_2$. This iteration of weighting matrix and coefficient estimation may be performed a fixed number of times, or until the coefficients converge so that $\hat{\beta}_j = \hat{\beta}_{j-1}$ to a sufficient degree of precision.

An alternative approach due to Hansen, Heaton and Yaron (1996) notes that since the optimal weighting matrix is dependent on the parameters, we may rewrite the GMM objective function as

$$J(\beta) = \frac{1}{T} u(\beta)' Z \hat{S}_T(\beta)^{-1} Z' u(\beta) \tag{23.36}$$

where the weighting matrix is a direct function of the β being estimated. The estimator which minimizes Equation (23.36) with respect to β has been termed the *Continuously Updated Estimator* (CUE).

Linear Equation Weight Updating

For equations that are linear in their coefficients, EViews offers three weighting matrix updating options: the **N-step Iterative**, the **Iterate to Convergence**, and the **Continuously Updating** method.

As the names suggests, the **N-Step Iterative** method repeats steps 2-5 above N times, while the **Iterate to Convergence** repeats the steps until the parameter estimates converge. The **Continuously Updating** approach is based on Equation (23.36).

Somewhat confusingly, the **N-Step Iterative method** with a single weight step is sometimes referred to in the literature as the 2-step GMM estimator, the first step being defined as the initial TSLS estimation. EViews views this as a 1-step estimator since there is only a single optimal weight matrix computation.

Non-linear Equation Weight Updating

For equations that are non-linear in their coefficients, EViews offers five different updating algorithms: **Sequential N-Step Iterative**, **Sequential Iterate to Convergence**, **Simultaneous Iterate to Convergence**, **1-Step Weight Plus 1 Iteration**, and **Continuously Updating**. The methods for non-linear specifications are generally similar to their linear counterparts, with differences centering around the fact that the parameter estimates for a given weighting matrix in step 4 must now be calculated using a non-linear optimizer, which itself involves iteration.

All of the non-linear weighting matrix update methods begin with $\hat{\beta}_0$ obtained from two-stage least squares estimation in which the coefficients have been iterated to convergence.

The **Sequential N-Step Iterative** procedure is analogous to the linear **N-Step Iterative** procedure outlined above, but with the non-linear optimization for the parameters in each step 4 iterated to convergence. Similarly, the **Sequential Iterate to Convergence** method follows the same approach as the **Sequential N-Step Iterative** method, with full non-linear optimization of the parameters in each step 4.

The **Simultaneous Iterate to Convergence** method differs from **Sequential Iterate to Convergence** in that only a single iteration of the non-linear optimizer, rather than iteration to

convergence, is conducted in step 4. The iterations are therefore simultaneous in the sense that each weight iteration is paired with a coefficient iteration.

1-Step Weight Plus 1 Iteration performs a single weight iteration after the initial two-stage least squares estimates, and then a single iteration of the non-linear optimizer based on the updated weight matrix.

The **Continuously Updating** approach is again based on Equation (23.36).

Coefficient Covariance Calculation

Having estimated the coefficients of the model, all that is left is to specify a method of computing the coefficient covariance matrix. We will consider two basic approaches, one based on a family of estimators of the asymptotic covariance given in Equation (23.29), and a second, due to Windmeijer (2000, 2005), which employs a bias-corrected estimator which take into account the variation of the initial parameter estimates.

Conventional Estimators

Using Equation (23.29) and inserting estimators and sample moments, we obtain an estimator for the asymptotic covariance matrix of $\hat{\beta}_1$:

$$\hat{V}_T(\hat{\beta}_1, \hat{\beta}_0) = \hat{A}^{-1} \hat{B}(\hat{S}^*) \hat{A}^{-1} \tag{23.37}$$

where

$$\begin{aligned} \hat{A} &= \nabla u(\hat{\beta}_1)' Z \hat{S}_T(\hat{\beta}_0)^{-1} Z' \nabla u(\hat{\beta}_1) \\ \hat{B} &= \nabla u(\hat{\beta}_1)' Z \hat{S}_T(\hat{\beta}_0)^{-1} \hat{S}^* \hat{S}_T(\hat{\beta}_0)^{-1} Z' \nabla u(\hat{\beta}_1) \end{aligned} \tag{23.38}$$

Notice that the estimator depends on both the final coefficient estimates $\hat{\beta}_1$ and the $\hat{\beta}_0$ used to form the estimation weighting matrix, as well as an additional estimate of the long-run covariance matrix $\hat{S}^*$. For weight update methods which iterate the weights until the coefficients converge the two sets of coefficients will be identical.

EViews offers a number of different covariance specifications of this form, **Estimation default**, **Estimation updated**, **Two-stage Least Squares**, **White**, **HAC (Newey-West)**, **Windmeijer**, **Cluster-robust**, and **User defined**, each corresponding to a different estimator for $\hat{S}^*$.

Of these, **Estimation default** and **Estimation update** are the most commonly employed coefficient covariance methods. Both methods compute $\hat{S}^*$ using the estimation weighting matrix specification (*i.e.* if **White** was chosen as the estimation weighting matrix, then **White** will also be used for estimating $\hat{S}^*$).

- **Estimation default** uses the previously computed estimate of the long-run covariance matrix to form $\hat{S}^* = \hat{S}_T(\hat{\beta}_0)$. The asymptotic covariance matrix simplifies considerably in this case so that $\hat{V}_T(\hat{\beta}) = \hat{A}^{-1}$.

- **Estimation updated** performs one more step 3 in the iterative estimation procedure, computing an estimate of the long-run covariance using the final coefficient estimates to obtain $\hat{S}^* = \hat{S}_T(\hat{\beta}_1)$. Since this method relies on the iterative estimation procedure, it is not available for equations estimated by CUE.

In cases, where the weighting matrices are iterated to convergence, these two approaches will yield identical results.

The remaining specifications compute estimates of $\hat{S}^*$ at the final parameters $\hat{\beta}_1$ using the indicated long-run covariance method. You may use these methods to estimate your equation using one set of assumptions for the weighting matrix $\hat{W}_T = \hat{S}_T(\hat{\beta}_0)$, while you compute the coefficient covariance using a different set of assumptions for $\hat{S}^* = \hat{S}_T(\hat{\beta}_1)$.

The primary application for this mixed weighting approach is in computing robust standard errors. Suppose, for example, that you want to estimate your equation using TSLS weights, but with robust standard errors. Selecting **Two-stage least squares** for the estimation weighting matrix and **White** for the covariance calculation method will instruct EViews to compute TSLS estimates with White coefficient covariances and standard errors. Similarly, estimating with **Two-stage least squares** estimation weights and **HAC - Newey-West** covariance weights produces TSLS estimates with HAC coefficient covariances and standard errors.

Note that it is possible to choose combinations of estimation and covariance weights that, while reasonable, are not typically employed. You may, for example, elect to use White estimation weights with HAC covariance weights, or perhaps HAC estimation weights using one set of HAC options and HAC covariance weights with a different set of options. It is also possible, though not recommended, to construct odder pairings such as HAC estimation weights with TSLS covariance weights.

Windmeijer Estimator

Various Monte Carlo studies (*e.g.* Arellano and Bond 1991) have shown that the above covariance estimators can produce standard errors that are downward biased in small samples. Windmeijer (2000, 2005) observes that part of this downward bias is due to extra variation caused by the initial weight matrix estimation being itself based on consistent estimates of the equation parameters.

Following this insight it is possible to calculate bias-corrected standard error estimates which take into account the variation of the initial parameter estimates. Windmeijer provides two forms of bias corrected standard errors; one for GMM models estimated in a one-step (one optimal GMM weighting matrix) procedure, and one for GMM models estimated using an iterate-to-convergence procedure.

The Windmeijer corrected variance-covariance matrix of the one-step estimator is given by:

$$V_{W2Step} = \hat{V}_1 + D_{2S}\hat{V}_1 + \hat{V}_1 D_{2S}' + D_{2S}\hat{V}_2 D_{2S}' \tag{23.39}$$

where:

$\hat{V}_1 = \hat{A}^{-1}$, the estimation default covariance estimator

$\hat{W}_{2T} = \hat{S}_T(\hat{\beta}_1)$, the updated weighting matrix (at final parameter estimates)

$\hat{V}_2 = \hat{A}^{-1}\hat{B}\hat{A}^{-1}$, the estimation updated covariance estimator where $\hat{S}^* = \hat{S}_T(\hat{\beta}_1)$

$\hat{W}_{1T} = \hat{S}_T(\hat{\beta}_0)$, the estimation weighting matrix (at initial parameter estimates)

$\hat{W}_{0T} = (\hat{\sigma}^2 Z'Z/T)$, the initial weighting matrix

$\partial \hat{W}_j^{-1} = \partial \hat{W}_{1T}^{-1}/\partial \beta_j$

D_{2S} is a matrix whose jth column is given by $D_{2S,j}$:

$$D_{2S,j} = -\hat{V}_1 \nabla u(\hat{\beta}_1)' Z \hat{W}_{2T}^{-1} \partial \hat{W}_j^{-1} \hat{W}_{2T}^{-1} Z' u(\hat{\beta}_1) - V$$

The Windmeijer iterate-to-convergence variance-covariance matrix is given by:

$$\hat{V}_{WIC} = (I - D_C)^{-1} \hat{V}_C (I - D_C)^{-1} \tag{23.40}$$

where:

$V_C = (\nabla u(\hat{\beta})' Z \hat{W}_{CT}^{-1} Z' u(\hat{\beta}))^{-1}$, the estimation default covariance estimator

$\hat{W}_{CT} = \hat{S}_T(\hat{\beta})$, the GMM weighting matrix at converged parameter estimates

Weighted GMM

Weights may also be used in GMM estimation. The objective function for weighted GMM is,

$$S(\beta) = \frac{1}{T}(y - f(X, \beta))' \Lambda Z \hat{S}_T^{-1} Z' \Lambda (y - f(X, \beta)) \tag{23.41}$$

where $\hat{S}_T$ is the long-run covariance of $w_t^* Z_t \epsilon_t$ where we now use Λ to indicate the diagonal matrix with observation weights w_t^*.

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

$$\hat{\Sigma}_{WGMM} = (G(b)' \Lambda Z \hat{S}_T^{-1} Z' \Lambda G(b))^{-1} \tag{23.42}$$

where $b \equiv b_{WGMM}$.

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** edit 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, by default, add a constant to the instrument list. If you do not wish a constant to be added to the instrument list, the **Include a constant** check box should be unchecked.

For example, if you type,

Equation spec: `y c x`

Instrument list: `c z w`

the orthogonality conditions are given by:

$$\begin{aligned} \sum (y_t - c(1) - c(2)x_t) &= 0 \\ \sum (y_t - c(1) - c(2)x_t)z_t &= 0 \\ \sum (y_t - c(1) - c(2)x_t)w_t &= 0 \end{aligned} \tag{23.43}$$

If you enter the specification,

Equation spec: `c(1)*log(y)+x^c(2)`

Instrument list: `c z z(-1)`

the orthogonality conditions are:

$$\begin{aligned} \sum(c(1)\log y_t + x_t^{c(2)}) &= 0 \\ \sum(c(1)\log y_t + x_t^{c(2)})z_t &= 0 \\ \sum(c(1)\log y_t + x_t^{c(2)})z_{t-1} &= 0 \end{aligned} \tag{23.44}$$

Beneath the **Instrument list** box there are two dropdown menus that let you set the **Estimation weighting matrix** and the **Weight updating**.

The **Estimation weight matrix** dropdown specifies the type of GMM weighting matrix that will be used during estimation. You can choose from **Two-stage least squares**, **White**, **HAC (Newey-West)**, and **User-specified**. If you select **HAC (Newey West)** then a button appears that lets you set the weighting matrix computation options. If you select **Cluster-robust**, you will be prompted for a cluster series. If you select **User-specified** you must enter the name of a symmetric matrix in the workfile containing an estimate of the weighting matrix (long-run covariance) scaled by the number of observations $\hat{U} = T\hat{S}$). Note that the matrix must have as many columns as the number of instruments specified.

The $\hat{U}$ matrix can be retrieved from any equation estimated by GMM using the `@instwgt` data member (see "Equation Data Members" on page 40 of the *Command and Programming Reference*). `@instwgt` returns $\hat{U}$ which is an implicit estimator of the long-run covariance scaled by the number of observations.

For example, for GMM equations estimated using the **Two-stage least squares** weighting matrix, will contain $\hat{\sigma}^2(Z'Z)$ (where the estimator for the variance will use s^2 or the no d.f. corrected equivalent, depending on your options for coefficient covariance calculation). Equations estimated with a **White** weighting matrix will return $\sum \hat{\epsilon}^2 Z_t Z_t'$.

Storing the user weighting matrix from one equation, and using it during the estimation of a second equation may prove useful when computing diagnostics that involve comparing *J*-statistics between two different equations.

The **Weight updating** dropdown menu lets you set the estimation algorithm type. For linear equations, you can choose between **N-Step Iterative**, **Iterate to Convergence**, and **Continuously Updating**. For non-linear equations, the choice is between **Sequential N-Step Iterative**, **Sequential Iterate to Convergence**, **Simultaneous Iterate to Convergence, 1-Step Weight Plus 1 Iteration**, and **Continuously Updating**.

To illustrate estimation of GMM models in EViews, we estimate the same Klein model introduced in "Estimating LIML and K-Class in EViews," on page 99, as again replicated by Greene 2008 (p. 385). We again estimate the Consumption equation, where consumption (CONS) is regressed on a constant, private profits (Y), lagged private profits (Y(-1)), and wages (W) using data in "Klein.WF1". The instruments are a constant, lagged corporate

profits (P(-1)), lagged capital stock (K(-1)), lagged GNP (X(-1)), a time trend (TM), Government wages (WG), Government spending (G) and taxes (T). Greene uses the **White** weighting matrix, and an **N-Step Iterative** updating procedure, with **N** set to 2. The results of this estimation are shown below:

Dependent Variable: CONS
Method: Generalized Method of Moments
Date: 04/21/09 Time: 12:17
Sample (adjusted): 1921 1941
Included observations: 21 after adjustments
Linear estimation with 2 weight updates
Estimation weighting matrix: White
Standard errors & covariance computed using estimation weighting matrix
No d.f. adjustment for standard errors & covariance
Instrument specification: C P(-1) K(-1) X(-1) TM WG G T

Variable	Coefficient	Std. Error	t-Statistic	Prob.
C	14.31902	0.896606	15.97025	0.0000
Y	0.090243	0.061598	1.465032	0.1612
Y(-1)	0.143328	0.065493	2.188443	0.0429
W	0.863930	0.029250	29.53616	0.0000

R-squared	0.976762	Mean dependent var	53.99524
Adjusted R-squared	0.972661	S.D. dependent var	6.860866
S.E. of regression	1.134401	Sum squared resid	21.87670
Durbin-Watson stat	1.420878	Instrument rank	8
J-statistic	3.742084	Prob(J-statistic)	0.442035

The EViews output header shows a summary of the estimation type and settings, along with the instrument specification. Note that in this case the header shows that the equation was linear, with a 2 step iterative weighting update performed. It also shows that the weighing matrix type was White, and this weighting matrix was used for the covariance matrix, with no degree of freedom adjustment.

Following the header the standard coefficient estimates, standard errors, t-statistics and associated p-values are shown. Below that information are displayed the summary statistics. Apart from the standard statistics shown in an equation, the instrument rank (the number of linearly independent instruments used in estimation) is also shown (8 in this case), and the J-statistic and associated p-value is also shown.

As a second example, we also estimate the equation for Investment. Investment (I) is regressed on a constant, private profits (Y), lagged private profits (Y(-1)) and lagged capital stock (K-1)). The instruments are again a constant, lagged corporate profits (P(-1)), lagged capital stock (K(-1)), lagged GNP (X(-1)), a time trend (TM), Government wages (WG), Government spending (G) and taxes (T).

Unlike Greene, we will use a **HAC** weighting matrix, with pre-whitening (fixed at 1 lag), a Tukey-Hanning kernel with Andrews Automatic Bandwidth selection. We will also use the

Continuously Updating weighting updating procedure. The output from this equation is show below:

Dependent Variable: I
Method: Generalized Method of Moments
Date: 08/10/09 Time: 10:48
Sample (adjusted): 1921 1941
Included observations: 21 after adjustments
Continuously updating weights & coefficients
Estimation weighting matrix: HAC (Prewhitening with lags = 1, Tukey -Hanning kernel, Andrews bandwidth = 2.1803)
Standard errors & covariance computed using estimation weighting matrix
Convergence achieved after 30 iterations
No d.f. adjustment for standard errors & covariance
Instrument specification: C P(-1) K(-1) X(-1) TM WG GT

Variable	Coefficient	Std. Error	t-Statistic	Prob.
C	22.20609	5.693625	3.900168	0.0012
Y	-0.261377	0.277758	-0.941024	0.3599
Y(-1)	0.935801	0.235666	3.970878	0.0010
K(-1)	-0.157050	0.024042	-6.532236	0.0000

R-squared	0.659380	Mean dependent var	1.266667
Adjusted R-squared	0.599271	S.D. dependent var	3.551948
S.E. of regression	2.248495	Sum squared resid	85.94740
Durbin-Watson stat	1.804037	Instrument rank	8
J-statistic	1.949180	Prob(J-statistic)	0.745106

Note that the header information for this equation shows slightly different information from the previous estimation. The inclusion of the **HAC** weighting matrix yields information on the prewhitening choice (lags = 1), and on the kernel specification, including the bandwidth that was chosen by the Andrews procedure (2.1803). Since the **CUE** procedure is used, the number of optimization iterations that took place is reported (39).

IV Diagnostics and Tests

EViews offers several IV and GMM specific diagnostics and tests.

Instrument Summary

The Instrument Summary view of an equation is available for non-panel equations estimated by GMM, TSLS or LIML. The summary will display the number of instruments specified, the instrument specification, and a list of the instruments that were used in estimation.

For most equations, the instruments used will be the same as the instruments that were specified in the equation, however if two or more of the instruments are collinear, EViews will automatically drop instruments until the instrument matrix is of full rank. In cases where instruments have been dropped, the summary will list which instruments were dropped.

The Instrument Summary view may be found under **View/IV Diagnostics & Tests/Instrument Summary**.

Instrument Orthogonality Test

The Instrument Orthogonality test, also known as the C-test or Eichenbaum, Hansen and Singleton (EHS) Test, evaluates the othogonality condition of a sub-set of the instruments. This test is available for non-panel equations estimated by TSLS or GMM.

Recall that the central assumption of instrumental variable estimation is that the instruments are orthogonal to a function of the parameters of the model:

$$E(Z'u(\beta)) = 0 \tag{23.45}$$

The Instrument Orthogonality Test evaluates whether this condition possibly holds for a subset of the instruments but not for the remaining instruments

$$\begin{aligned} E(Z_1'u(\beta)) &= 0 \\ E(Z_2'u(\beta)) &\neq 0 \end{aligned} \tag{23.46}$$

Where $Z = (Z_1, Z_2)$, and Z_1 are instruments for which the condition is assumed to hold.

The test statistic, C_T, is calculated as the difference in J-statistics between the original equation and a secondary equation estimated using only Z_1 as instruments:

$$C_T = \frac{1}{T}u(\beta)'Z\hat{W}_T^{-1}Z'u(\beta) - \frac{1}{T}u(\tilde{\beta})'Z_1\hat{W}_{T1}^{-1}Z_1'u(\tilde{\beta}) \tag{23.47}$$

where $\hat{\beta}$ are the parameter estimates from the original TSLS or GMM estimation, and $\hat{W}_T$ is the original weighting matrix, $\tilde{\beta}$ are the estimates from the test equation, and $\hat{W}_{T1}^{-1}$ is the matrix for the test equation formed by taking the subset of $\hat{W}_T^{-1}$ corresponding to the instruments in Z_1. The test statistic is Chi-squared distributed with degrees of freedom equal to the number of instruments in Z_2.

To perform the Instrumental Orthogonality Test in EViews, click on **View/IV Diagnostics and Tests/Instrument Orthogonality Test**. A dialog box will the open up asking you to enter a list of the Z_2 instruments for which the orthogonality condition may not hold. Click on **OK** and the test results will be displayed.

Regressor Endogeneity Test

The Regressor Endogeneity Test, also known as the Durbin-Wu-Hausman Test, tests for the endogeneity of some, or all, of the equation regressors. This test is available for non-panel equations estimated by TSLS or GMM.

A regressor is endogenous if it is explained by the instruments in the model, whereas exogenous variables are those which are not explained by instruments. In EViews' TSLS and GMM estimation, exogenous variables may be specified by including a variable as both a regressor

and an instrument, whereas endogenous variable are those which are specified in the regressor list only.

The Endogeneity Test tests whether a subset of the endogenous variables are actually exogenous. This is calculated by running a secondary estimation where the test variables are treated as exogenous rather than endogenous, and then comparing the J-statistic between this secondary estimation and the original estimation:

$$H_T = \frac{1}{T} u(\tilde{\beta})' \tilde{Z} \tilde{W}_T^{-1} \tilde{Z}' u(\tilde{\beta}) - \frac{1}{T} u(\beta)' Z \tilde{W}_{T*}^{-1} Z' u(\beta) \tag{23.48}$$

where $\tilde{\beta}$ are the parameter estimates from the original TSLS or GMM estimation obtained using weights $\tilde{W}_T$, and $\tilde{\beta}$ are the estimates from the test equation estimated using $\tilde{Z}$, the instruments augmented by the variables which are being tested, and $\tilde{W}_T$ is the weighting matrix from the secondary estimation.

Note that in the case of GMM estimation, the matrix $\tilde{W}_{T*}^{-1}$ should be a sub-matrix of $\tilde{W}_T^{-1}$ to ensure positivity of the test statistic. Accordingly, in computing the test statistic, EViews first estimates the secondary equation to obtain $\tilde{\beta}$, and then forms a new matrix $\tilde{W}_{T*}^{-1}$, which is the subset of $\tilde{W}_T^{-1}$ corresponding to the original instruments Z. A third estimation is then performed using the subset matrix for weighting, and the test statistic is calculated as:

$$H_T = \frac{1}{T} u(\tilde{\beta})' \tilde{Z} \tilde{W}_T^{-1} \tilde{Z}' u(\tilde{\beta}) - \frac{1}{T} u(\beta^*)' Z' \tilde{W}_{T*}^{-1} Z' u(\beta^*) \tag{23.49}$$

The test statistic is distributed as a Chi-squared random variable with degrees of freedom equal to the number of regressors tested for endogeneity.

To perform the Regressor Endogeneity Test in EViews, click on **View/IV Diagnostics and Tests/Regressor Endogeneity Test**. A dialog box will the open up asking you to enter a list of regressors to test for endogeneity. Once you have entered those regressors, hit **OK** and the test results are shown.

Weak Instrument Diagnostics

The Weak Instrument Diagnostics view provides diagnostic information on the instruments used during estimation. This information includes the Cragg-Donald statistic, the associated Stock and Yugo critical values, and Moment Selection Criteria (MSC). The Cragg-Donald statistic and its critical values are available for equations estimated by TSLS, GMM or LIML, but the MSC are available for equations estimated by TSLS or GMM only.

The Cragg-Donald statistic is proposed by Stock and Yugo as a measure of the validity of the instruments in an IV regression. Instruments that are only marginally valid, known as weak instruments, can lead to biased inferences based on the IV estimates, thus testing for the presence of weak instruments is important. For a discussion of the properties of IV estima-

tion when the instruments are weak, see, for example, Moreira 2001, Stock and Yugo 2004 or Stock, Wright and Yugo 2002.

Although the Cragg-Donald statistic is only valid for TSLS and other K-class estimators, EViews also reports for equations estimated by GMM for comparative purposes.

The Cragg-Donald statistic is calculated as:

$$G_t = \left(\frac{(T-k_1-k_2)^2}{k_2}\right)(X_E{}'M_{XZ}X_E)^{-1/2}(M_XX_E)'M_XZ_Z((M_XZ_Z)'(M_XZ_Z))^{-1} \quad (23.50)$$

$$(M_XZ_Z)'(M_XX_E)(X_E{}'M_{XZ}X_E)^{-1/2}$$

where:

Z_Z = instruments that are not in the regressor list

$X_Z = (X_X | Z_Z)$

X_X = exogenous regressors (regressors in both the regressor and instrument lists)

X_E = endogenous regressors (regressors that are not in instrument list)

$M_{XZ} = I - X_Z(X_Z{}'X_Z)^{-1}X_Z{}'$

$M_X = I - X_X(X_X{}'X_X)^{-1}X_X{}'$

k_1 = number of columns of X_X

k_2 = number of columns of Z_Z

The statistic does not follow a standard distribution, however Stock and Yugo provide a table of critical values for certain combinations of instruments and endogenous variable numbers. EViews will report these critical values if they are available for the specified number of instruments and endogenous variables in the equation.

Moment Selection Criteria (MSC) are a form of Information Criteria that can be used to compare different instrument sets. Comparison of the MSC from equations estimated with different instruments can help determine which instruments perform the best. EViews reports three different MSCs: two proposed by Andrews (1999)—a Schwarz criterion based, and a Hannan-Quinn criterion based, and the third proposed by Hall, Inoue, Jana and Shin (2007)—the Relevant Moment Selection Criterion. They are calculated as follows:

SIC-based = $J_T - (c-k)\ln(T)$

HQIQ-based = $J_T - 2.01(c-k)\ln(\ln(T))$

Relevant MSC = $\ln(|T\Omega|)(1/\tau)(c-k)\ln(\tau)$

where c = the number of instruments, k = the number of regressors, T = the number of observations, Ω = the estimation covariance matrix,

$$\tau = \left(\frac{T}{b}\right)^{1/2}$$

and b is equal 1 for TSLS and White GMM estimation, and equal to the bandwidth used in HAC GMM estimation.

To view the Weak Instrument Diagnostics in EViews, click on **View/IV Diagnostics & Tests/ Weak Instrument Diagnostics**.

GMM Breakpoint Test

The GMM Breakpoint test is similar to the Chow Breakpoint Test, but it is geared towards equations estimated via GMM rather than least squares.

EViews calculates three different types of GMM breakpoint test statistics: the Andrews-Fair (1988) Wald Statistic, the Andrews-Fair LR-type Statistic, and the Hall and Sen (1999) O-Statistic. The first two statistics test the null hypothesis that there are no structural breaks in the equation parameters. The third statistic tests the null hypothesis that the over-identifying restrictions are stable over the entire sample.

All three statistics are calculated in a similar fashion to the Chow Statistic – the data are partitioned into different subsamples, and the original equation is re-estimated for each of these subsamples. However, unlike the Chow Statistic, which is calculated on the basis that the variance-covariance matrix of the error terms remains constant throughout the entire sample (i.e s^2. is the same between subsamples), the GMM breakpoint statistic lets the variance-covariance matrix of the error terms vary between the subsamples.

The Andrews-Fair Wald Statistic is calculated, in the single breakpoint case, as:

$$AF_1 = (\theta_1 - \theta_2)'\left(\frac{1}{T_1}V_1^{-1} + \frac{1}{T_2}V_2^{-1}\right)^{-1}(\theta_1 - \theta_2) \tag{23.51}$$

Where θ_i refers to the coefficient estimates from subsample i, T_i refers to the number of observations in subsample i, and V_i is the estimate of the variance-covariance matrix for subsample i.

The Andrews-Fair LR-type statistic is a comparison of the J-statistics from each of the subsample estimations:

$$AF_2 = J_R - (J_1 + J_2) \tag{23.52}$$

Where J_R is a J-statistic calculated with the original equation's residuals, but a GMM weighting matrix equal to the weighted (by number of observations) sum of the estimated weighting matrices from each of the subsample estimations.

The Hall and Sen O-Statistic is calculated as:

$$O_T = J_1 + J_2 \tag{23.53}$$

The first two statistics have an asymptotic χ^2 distribution with $(m-1)k$ degrees of freedom, where m is the number of subsamples, and k is the number of coefficients in the original equation. The O-statistic also follows an asymptotic χ^2 distribution, but with $2 \times (q-(m-1)k)$ degrees of freedom.

To apply the GMM Breakpoint test, click on **View/Breakpoint Test….** In the dialog box that appears simply enter the dates or observation numbers of the breakpoint you wish to test.

References

Amemiya, T. (1975). "The Nonlinear Limited-Information Maximum-Likelihood Estimator and the Modified Nonlinear Two-Stage Least-Squares Estimator," *Journal of Econometrics*, 3, 375-386.

Anderson, T.W. and H. Rubin (1950). "The Asymptotic Properties of Estimates of the Parameters of a Single Equation in a Complete System of Stochastic Equations," *The Annals of Mathematical Statistics*, 21(4), 570-582.

Andrews, D.W.K. (1999). "Consistent Moment Selection Procedures for Generalized Method of Moments Estimation," *Econometrica*, 67(3), 543-564.

Andrews, D.W.K. (Oct. 1988). "Inference in Nonlinear Econometric Models with Structural Change," *The Review of Economic Studies*, 55(4), 615-639.

Anderson, T. W. and H. Rubin (1949). "Estimation of the parameters of a single equation in a complete system of stochastic equations," *Annals of Mathematical Statistics*, 20, 46–63.

Arellano, M. and S. Bond (1991). "Some Tests of Specification For Panel Data: Monte Carlo Evidence and an Application to Employment Equations," *Review of Economic Studies*, 38, 277-297.

Bekker, P. A. (1994). "Alternative Approximations to the Distributions of Instrumental Variable Estimators," *Econometrica*, 62(3), 657-681.

Cragg, J.G. and S. G. Donald (1993). "Testing Identifiability and Specification in Instrumental Variable Models," *Econometric Theory*, 9(2), 222-240.

Eichenbaum, M., L.P. Hansen, and K.J. Singleton (1988). "A Time Series Analysis of Representative Agent Models of Consumption and Leisure Choice under Uncertainty," *The Quarterly Journal of Economics*, 103(1), 51-78.

Hahn, J. and A. Inoue (2002). "A Monte Carlo Comparison of Various Asymptotic Approximations to the Distribution of Instrumental Variables Estimators," *Econometric Reviews*, 21(3), 309-336

Hall, A.R., A. Inoue, K. Jana, and C. Shin (2007). "Information in Generalized Method of Moments Estimation and Entropy-based Moment Selection," *Journal of Econometrics*, 38, 488-512.

Hansen, C., J. Hausman, and W. Newey (2006). "Estimation with Many Instrumental Variables," *MIMEO*.

Hausman, J., J.H. Stock, and M. Yogo (2005). "Asymptotic Properties of the Han-Hausman Test for Weak Instruments," *Economics Letters*, 89, 333-342.

Moreira, M.J. (2001). "Tests With Correct Size When Instruments Can Be Arbitrarily Weak," *MIMEO*.

Stock, J.H. and M. Yogo (2004). "Testing for Weak Instruments in Linear IV Regression," *MIMEO*.

Stock, J.H., J.H. Wright, and M. Yogo (2002). "A Survey of Weak Instruments and Weak Identification in Generalized Method of Moments," *Journal of Business & Economic Statistics*, 20(4), 518-529.

Windmeijer, F. (2000). "A finite Sample Correction for the Variance of Linear Two-Step GMM Estimators," *The Institute for Fiscal Studies*, Working Paper 00/19.

Windmeijer, F. (2005). "A finite Sample Correction for the Variance of Linear efficient Two-Step GMM Estimators," *Journal of Econometrics,* 126, 25-51.

Chapter 24. Time Series Regression

In this chapter, we turn our attention to the analysis of single equation models for time series data, focusing on the estimation of Autoregressive-Moving Average (ARMA), Autoregressive-Integrated-Moving Average (ARIMA), and Autoregressive-Fractionally Integrated-Moving Average (ARFIMA) specifications, and the computation of equation diagnostics for these models.

Before turning to the EViews implementation of these features, we provide brief background for the models and related diagnostics. Those desiring additional detail are encouraged to consult one or more of the many book length treatments of time series methods (Box, Jenkins, and Reinsel, 2008; Hamilton, 1994).

Related topics are discussed elsewhere in this volume; see, for example, Chapter 42. "Univariate Time Series Analysis," on page 667, Chapter 44. "Vector Autoregression and Error Correction Models," on page 833, Chapter 48. "State Space Models and the Kalman Filter," on page 969 for material on additional time series topics.

Background

A common occurrence in time series regression is the presence of correlation between residuals and their lagged values. This *serial correlation* violates the standard assumption of regression theory which requires uncorrelated regression disturbances. Among the problems associated with unaccounted for serial correlation in a regression framework are:

- OLS is no longer efficient among linear estimators. Intuitively, 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 of the equation specification, OLS estimates are biased and inconsistent.

A popular framework for modeling serial dependence is the Autoregressive-Moving Average (ARMA) and Autoregressive-Integrated-Moving Average (ARIMA) models popularized by Box and Jenkins (1976) and generalized to Autoregressive-Fractionally Integrated-Moving Average (ARFIMA) specifications.

(Note that ARMA and ARIMA models which allow for explanatory variables in the mean are sometimes termed *ARIMAX* and *ARIMAX*. We will generally use *ARMA* to refer to models both with and without explanatory variables unless there is a specific reason to distinguish between the two types.)

Autoregressive (AR) Models

An autoregressive model of order p, denoted AR(p) has the form

$$\begin{aligned} Y_t &= \rho_1 Y_{t-1} + \rho_2 Y_{t-2} + \ldots + \rho_p Y_{t-p} + \epsilon_t \\ &= \sum_{j=1}^{p} \rho_j Y_{t-j} + \epsilon_t \end{aligned} \tag{24.1}$$

where ϵ_t are the independent and identically distributed innovations for the process and the autoregressive parameters ρ_i characterize the na ture of the dependence. Note that the autocorrelations of a stationary AR(p) are infinite, but decline geometrically so they die off quickly, and the partial autocorrelations for lags greater than p are zero.

It will be convenient for the discussion to follow to define a lag operator L such that:

$$L^k Y_t = Y_{t-k} \tag{24.2}$$

and to rewrite the AR(p) as

$$\begin{aligned} Y_t &= \sum_{j=1}^{p} \rho_j L^j Y_t + \epsilon_t \\ \rho(L) Y_t &= \epsilon_t \end{aligned} \tag{24.3}$$

where

$$\rho(L) = 1 - \sum_{j=1}^{p} \rho_j L^j \tag{24.4}$$

is a lag polynomial that characterizes the AR process.If we add a mean to the model, we obtain:

$$\rho(L)(Y_t - \mu_t) = \epsilon_t \tag{24.5}$$

The AR(1) Model

The simplest and most widely used regression model with serial correlation is the first-order autoregressive, or AR(1), model. If the mean $\mu_t = X_t'\beta$ is a linear combination of regressors X_t and parameters β, the AR(1) model may be written as:

$$\begin{aligned} Y_t &= X_t'\beta + u_t \\ u_t &= \rho u_{t-1} + \epsilon_t \end{aligned} \tag{24.6}$$

The parameter ρ is the first-order serial correlation coefficient.

Substituting the second equation into the first, we obtain the regression form

$$\begin{aligned} Y_t &= X_t'\beta + \rho L(Y_t - X_t'\beta) + \epsilon_t \\ &= X_t'\beta + \rho(Y_{t-1} - X_{t-1}'\beta) + \epsilon_t \end{aligned} \tag{24.7}$$

In the representation it is easy to see that the AR(1) model incorporates the residual from the previous observation into the regression model for the current observation.

Rearranging terms and using the lag operator, we have the polynomial form

$$(1 - \rho L)(Y_t - X_t'\beta) = \epsilon_t \tag{24.8}$$

Higher-Order AR Models

A regression model with an autoregressive process of order p, AR(p), is given by:

$$\begin{aligned} Y_t &= X_t'\beta + u_t \\ u_t &= \rho_1 u_{t-1} + \rho_2 u_{t-2} + \ldots + \rho_p u_{t-p} + \epsilon_t \end{aligned} \tag{24.9}$$

Substituting and rearranging terms, we get the regression

$$Y_t = X_t'\beta + \sum_{j=1}^{p} \rho_j (Y_{t-j} - X_{t-j}'\beta) + \epsilon_t \tag{24.10}$$

and the polynomial form

$$\left(1 - \sum_{j=1}^{p} \rho_j L^j\right)(Y_t - X_t'\beta) = \epsilon_t \tag{24.11}$$

Moving Average (MA) Models

A moving average model of order q, denoted MA(q) has the form

$$\begin{aligned} Y_t &= \epsilon_t + \theta_1 \epsilon_{t-1} + \rho\theta_2 \epsilon_{t-2} + \ldots + \theta_q Y_{t-q} \\ &= \epsilon_t + \sum_{j=1}^{q} \theta_j \epsilon_{t-j} \\ &= \theta(L)\epsilon_t \end{aligned} \tag{24.12}$$

where ϵ_t are the innovations, and

$$\theta(L) = 1 + \sum_{j=1}^{q} \theta_j L^j \tag{24.13}$$

is the moving average polynomial with parameters θ_i that characterize the MA process. Note that the autocorrelations of an MA model are zero for lags greater than q.

You should pay particular attention to the definition of the lag polynomial when comparing results across different papers, books, or software, as the opposite sign convention is sometimes employed for the θ coefficients.

Adding a mean to the model, we get the mean adjusted form:

$$Y_t - \mu = \theta(L)\epsilon_t \tag{24.14}$$

The MA(1) Model

The MA(1) model assumes that the current disturbance term u_t is a weighted sum of the current and lagged innovations ϵ_t and ϵ_{t-1}:

$$\begin{aligned} Y_t &= X_t'\beta + u_t \\ u_t &= \epsilon_t + \theta\epsilon_{t-1} \end{aligned} \tag{24.15}$$

The parameter θ is the first-order moving average coefficient. Substituting, the MA(1) may be written as

$$Y_t = X_t'\beta + \epsilon_t + \theta\epsilon_{t-1} \tag{24.16}$$

and

$$Y_t - X_t'\beta = (1 + \theta L)\epsilon_t \tag{24.17}$$

Autoregressive Moving Average (ARMA) Models

We may combine the AR and the MA specifications to define an autoregressive model moving average (ARMA) model:

$$\rho(L)(Y_t - \mu_t) = \theta(L)\epsilon_t \tag{24.18}$$

We term this model an ARMA(p, q) to indicate that there are p lags in the AR and q terms in the MA.

The ARMA(1, 1) Model

The simplest ARMA model is first-order autoregressive with a first-order moving average error:

$$\begin{aligned} Y_t &= X_t'\beta + u_t \\ u_t &= \rho u_{t-1} + \epsilon_t + \theta\epsilon_{t-1} \end{aligned} \tag{24.19}$$

The parameter ρ is the first-order serial correlation coefficient, and the θ is the moving average coefficient. Substituting, the ARMA(1, 1) may be written as

$$\begin{aligned} Y_t &= X_t'\beta + \rho u_{t-1} + \epsilon_t + \theta\epsilon_{t-1} \\ &= X_t'\beta + \rho(Y_{t-1} - X_{t-1}'\beta) + \epsilon_t + \theta\epsilon_{t-1} \end{aligned} \tag{24.20}$$

or equivalently,

$$(1-\rho L)(Y_t - X_t'\beta) = (1+\theta L)\epsilon_t \tag{24.21}$$

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. Processes with SAR and SMA terms are ARMA models constructed using products of lag polynomials. These products produce higher order ARMA models with nonlinear restrictions on the coefficients.

Seasonal AR Terms

A SAR(p) term is a seasonal autoregressive term with lag p. A SAR adds to an existing AR specification a polynomial with a lag of p:

$$1-\phi_p L^p \tag{24.22}$$

The SAR is not intended to be used alone. The SAR allows you to form the product of lag polynomials, with the resulting lag structure defined by the product of the AR and SAR lag polynomials.

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

$$Y_t = \rho_1 Y_{t-1} + \rho_2 Y_{t-2} + \epsilon_t, \tag{24.23}$$

which can be represented using the lag operator L as:

$$(1-\rho_1 L - \rho_2 L^2) Y_t = \epsilon_t \tag{24.24}$$

For quarterly data, we might wish to add a SAR(4) term because we believe that there is correlation between a quarter and the quarter the year previous. Then the resulting process would be:

$$(1-\rho_1 L - \rho_2 L^2)(1-\phi_4 L^4) Y_t = \epsilon_t. \tag{24.25}$$

Expanded terms, we see that the process is equivalent to:

$$Y_t = \rho_1 Y_{t-1} + \rho_2 Y_{t-2} + \phi_4 Y_{t-4} - \phi_4\rho_1 Y_{t-5} - \phi_4\rho_2 Y_{t-6} + \epsilon_t. \tag{24.26}$$

The parameter ϕ_4 is associated with the seasonal part of the process. Note that this is an AR(6) process with nonlinear restrictions on the coefficients.

Seasonal MA Terms

Similarly, SMA(q) can be included in your specification to specify a seasonal moving average term with lag q. The resulting the MA lag structure is obtained from the product of the lag polynomial specified by the MA terms and the one specified by any SMA terms.

For example, second-order MA process without seasonality may be written as

$$Y_t = \epsilon_t + \theta_1 \epsilon_{t-1} + \theta_2 \epsilon_{t-2}, \tag{24.27}$$

or using lag operators:

$$Y_t = (1 + \theta_1 L + \theta_2 L^2)\epsilon_t. \tag{24.28}$$

To take account of seasonality in a quarterly workfile you may wish to add an SMA(4). Then the resulting process is:

$$Y_t = (1 + \theta_1 L + \theta_2 L^2)(1 + \omega_4 L^4)\epsilon_t \tag{24.29}$$

The process is equivalent to:

$$Y_t = Y_t + \theta_1 Y_{t-1} + \theta_2 Y_{t-2} + \omega_4 Y_{t-4} + \omega_4 \theta_1 Y_{t-5} + \omega_4 \theta_2 Y_{t-6}. \tag{24.30}$$

The parameter w_4 is associated with the seasonal part of an MA(6) process which has non-linear restrictions on the coefficients.

Integrated Models

A time series Y_t is said to be integrated of order 0 or $I(0)$, if it may be written as a MA process $Y_t = \theta(L)\epsilon_t$, with coefficients such that

$$\sum_{i=1}^{\infty} |\theta_i| < \infty \tag{24.31}$$

Roughly speaking, an $I(0)$ process is a moving average with autocovariances that die off sufficiently quickly, a condition which is necessary for stationarity (Hamilton, 2004).

Y_t is said to be integrated of order d or $I(d)$, if its d-th integer difference, $(1-L)^d Y_t$ is $I(0)$, and the $d-1$ difference is not.

Typically, one assumes that d is an integer and that $d = 1$ or $d = 2$ so that first or second differencing the original series yields a stationary series. We will consider both integer and non-integer integration in turn.

ARIMA Model

An ARIMA(p, d, q) model is defined as an $I(d)$ process whose d-th integer difference follows a stationary ARMA(p, q) process. In polynomial form we have:

$$\rho(L)(1-L)^d(Y_t - \mu_t) = \theta(L)\epsilon_t \tag{24.32}$$

Example

The ARIMA(1,1,1) Model

An ARIMA(1,1,1) model for Y_t assumes that the first difference of Y_t is an ARMA(1,1).

$$(1-\rho L)(1-L)(Y_t - X_t'\beta) = \epsilon_t + \theta\epsilon_{t-1} \tag{24.33}$$

Rearranging, and noting that $(1-L)Y_t = \Delta Y_t$ and $(1-L)X_t = \Delta X_t$ we may write this specification as

$$\Delta Y_t = \Delta X_t'\beta + \rho(\Delta Y_{t-1} - \Delta X_{t-1}'\beta) + \epsilon_t + \theta\epsilon_{t-1} \tag{24.34}$$

or

$$\begin{aligned} \Delta Y_t &= \Delta X_t'\beta + u_t \\ u_t &= \rho u_{t-1} + \epsilon_t + \theta\epsilon_{t-1} \end{aligned} \tag{24.35}$$

ARFIMA Model

Stationary processes are said to have *long memory* when autocorrelations are persistent, decaying more slowly than the rate associated with ARMA models. Modeling long term dependence is difficult for standard ARMA specifications as it requires non-parsimonious, large-order ARMA representations that are generally accompanied by undesirable short-run dynamics (Sowell, 1992).

One popular approach to modeling long memory processes is to employ the notion of *fractional integration* (Granger and Joyeux, 1980; Hosking, 1981). A fractionally integrated series is one with long-memory that is not $I(1)$.

Following Granger and Joyeux (1981) and Hosking (1981), we may define a discrete time fractional difference operator which depends on the parameter d:

$$\nabla^d = (1-L)^d = \sum_{k=0}^{\infty} \binom{d}{k} (-1)^k L^k = \sum_{k=0}^{\infty} \frac{\Gamma(-d+k)}{\Gamma(-d)\Gamma(k+1)} L^k \tag{24.36}$$

for $-1/2 < d < 1/2$ and Γ the gamma function.

If the fractional d-difference operator applied to a process produces a random walk we say that the process is an ARFIMA(0, d, 0). Hosking notes that for an ARFIMA(0, d, 0):

- when $d = 1/2$, the process is non-stationary but invertible
- when $0 < d < 1/2$, the process has long memory but is stationary and invertible
- when $-1/2 < d < 0$, the process has short memory, with all negative auto-correlations and partial auto correlations, and is invertible
- when $d = 0$, the process is white noise

More generally, if d-th order fractional differencing results in an ARMA(p, q), the process is said to be ARFIMA(p, d, q). In polynomial form we have:

$$\rho(L)(1-L)^d(Y_t - \mu_t) = \theta(L)\epsilon_t \tag{24.37}$$

Notice that the ARFIMA specification is identical to the standard Box-Jenkins ARIMA formulation in Equation (24.32), but allowing for non-integer d. Note also that the range restric-

tion on d is non-binding as we may apply integer differencing or summing until d is in the desired range.

By combining fractional differencing with a traditional ARMA specification, the ARFIMA model allows for flexible dynamic patterns. Crucially, when $-1/2 < d < 1/2$, the autocorrelations and partial autocorrelations of the ARFIMA process decay more slowly (hyperbolically) than the rates associated with ARMA specifications. Thus, the ARFIMA model allows you to model slowing decaying long-run dependence using the d parameter and more rapidly decaying short-run dynamics using a parsimonious ARMA(p, q).

The Box-Jenkins (1976) Approach to ARIMA Modeling

In Box-Jenkins ARIMA modeling and forecasting, you assemble a complete forecasting model by using combinations of the three ARIMA 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 462 of *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 $u_1, \ldots, u_{t-5}$ 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. Along these lines, 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.

For example, we can examine the correlogram of the DRI Basics housing series in the "Hs.WF1" workfile by setting the sample to "1959m01 1984m12" then selecting **View/Correlogram...** from the HS series toolbar. Click on **OK** to accept the default settings and display the result.

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 several views for diagnostic checks after estimation.

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:

$$u_t = \rho u_{t-1} + \epsilon_t . \tag{24.38}$$

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 Diagnostics/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 464 of *User’s Guide I*.

Serial Correlation LM Test

Selecting **View/Residual Diagnostics/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 215 for further discussion of the serial correlation LM test.

Example

As an example of the application of serial correlation testing procedures, consider the following results from estimating a simple consumption function by ordinary least squares using data in the workfile "Uroot.WF1":

Dependent Variable: CS
Method: Least Squares
Date: 08/10/09 Time: 11:06
Sample: 1948Q3 1988Q4
Included observations: 162

Variable	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 Diagnostics/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 Diagnostics/Serial Correlation LM Test…** and entering a lag of 4 yields the following result (top portion only):

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 ARIMA and ARFIMA Models in EViews

EViews estimates ARIMA models for linear and nonlinear equations specifications defined by list or expression, and ARFIMA models for linear specifications defined by list.

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).

To estimate an ARMA, ARIMA, or ARFIMA model in EViews, open an equation object by clicking on **Quick/Estimate Equation…** or **Object/New Object…/Equation** in the main EViews menu, or type equation in the command line:

Putting aside the **Equation specification** for a moment, consider the **Estimation settings** section at the bottom of the dialog

- When estimating ARMA models, you may choose **LS – Least Squares (NLS and ARMA)**, **TSLS – Two-Stage Least Squares (TSNLS and ARMA)**, or **GMM - Generalized Method of Moments** in the estimation **Method** dropdown menu.

  Note that some estimation techniques and methods (notable maximum likelihood and fractional integration) are only available under the least squares option.

- Enter the sample specification in the **Sample** edit dialog.

As the focus of our discussion will be on the equation specification for standard ARIMA and ARFIMA models and on the corresponding settings on the **Options** tab, the remainder of our discussion will assume you have selected the **LS – Least Squares (NLS and ARMA)** method in the dropdown menu. We will make brief comments about other specifications when appropriate.

Equation Specification

EViews estimates general ARIMA and ARFIMA specifications that allow for right-hand side explanatory variables (ARIMAX and ARFIMAX).

You should enter your equation specification in the top edit field. As with other equation specifications, you may enter your equation by listing the dependent variable followed by

explanatory variables and special keywords, or you may provide an explicit expression for the equation.

To specify your ARIMA model, you will:

- Difference your dependent variable, *if necessary,* to account for the integer order of integration.
- Describe your structural regression model (dependent variables and mean regressors) and add AR, SAR, MA, SMA terms, as necessary.

To specify your ARFIMA model you will:

- Difference your dependent variable, *if necessary,* to account for an integer order of integration.
- Describe your structural regression model (dependent variables and regressors) and add any ordinary and seasonal ARMA terms, if desired.
- Add the `d` keyword to the specification to indicate that you would like to estimate and use a fractional difference parameter d.

Specifying AR Terms

To specify an AR term in EViews, you will use the keyword `ar`, followed by the desired lag or lag range enclosed in parentheses. You must explicitly instruct EViews to use *each AR lag you wish to include*.

First-Order AR

For specifications defined by list, simply add the `ar` keywords to the list. For example, to estimate a simple consumption function with AR(1) errors, and enter your list of variables as usual, adding the keyword expression `AR(1)` to the end of your list.

For the specification:

$$\begin{aligned} CS_t &= c_1 + c_2 GDP_t + u_t \\ u_t &= \rho u_{t-1} + \epsilon_t \end{aligned} \tag{24.39}$$

with the series CS and GDP in the workfile, you may specify your equation as:

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

For specifications defined by expression, specify your model using EViews expressions, followed by an additive term describing the AR lag coefficient assignment enclosed in square brackets. For the revised specification:

$$\begin{aligned} CS_t &= c_1 + GDP_t^{c_2} + u_t \\ u_t &= \rho u_{t-1} + \epsilon_t \end{aligned} \tag{24.40}$$

you would enter

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

Higher-Order AR

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 lag you wish to include*. You may use the `to` keyword to define a lag range.

If you wish to estimate a model with autocorrelations from one to five:

$$\begin{aligned} CS_t &= c_1 + c_2 GDP_t + u_t \\ u_t &= \rho_1 u_{t-1} + \rho_2 u_{t-2} + \ldots + \rho_5 u_{t-5} + \epsilon_t \end{aligned} \tag{24.41}$$

you may define your specification using:

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

or more concisely

```
cs c gdp ar(1 to 5)
```

The latter form specifies a lag range from 1 to 5 using the `to` keyword.

We emphasize the fact that you must explicitly list AR lags that you wish to include. By requiring that you enter all of the desired AR terms, EViews allows you the flexibility to restrict lower order correlations to be zero. For example, if you have quarterly data and want only to include a single term to account for seasonal autocorrelation, you could enter

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

For specifications defined by expression, you must list the coefficient assignment for each of the lags separately, separated by commas:

$$\begin{aligned} CS_t &= c_1 + GDP_t^{c_2} + u_t \\ u_t &= \rho_1 u_{t-1} + \rho_2 u_{t-2} + \ldots + \rho_5 u_{t-5} + \epsilon_t \end{aligned} \tag{24.42}$$

you would enter

```
cs = c(1) + gdp^c(2) + [ar(1)=c(3), ar(2)=c(4), ar(3)=c(5),
    ar(4)=c(6), ar(5)=c(7)]
```

Seasonal AR

Seasonal AR terms may be added using the `sar` keyword, followed by a lag or lag range enclosed in parentheses. The specification

```
cs c gdp ar(1) sar(4)
```

will define an AR(5) model with coefficient restrictions as described above ("Seasonal ARMA Terms" on page 123).

Note that in the absence ordinary AR terms, the `sar` is equivalent to an `ar`. Thus,

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

are equivalent specifications.

Specifying MA Terms

To specify an MA term in EViews, you will use the keyword `ma`, followed by the desired lag or lag range enclosed in parentheses. You must explicitly instruct EViews to use *each MA lag you wish to include*. You may use the `to` keyword to define a lag range.

For specifications defined by list,:

$$\begin{aligned} CS_t &= c_1 + c_2 GDP_t + u_t \\ u_t &= \epsilon_t + \theta_1 \epsilon_{t-1} + \theta_2 \epsilon_{t-2} \end{aligned} \tag{24.43}$$

you would specify the equation as

```
cs c gdp ma(1) ma(2)
```

or more concisely as

```
cs c gdp ma(1 to 2)
```

For specifications defined by expression, the MA keywords require coefficient assignments for each lag so that they must be entered individually. Thus, for

$$\begin{aligned} CS_t &= c_1 + GDP_t^{c_2} + u_t \\ u_t &= \epsilon_t + \theta_1 \epsilon_{t-1} + \theta_2 \epsilon_{t-2} \end{aligned} \tag{24.44}$$

you would enter

```
cs = c(1) + gdp^c(2) + [ma(1)=c(3), ma(2)=c(4)]
```

Seasonal MA terms may be added using the `sma` keyword, followed by a lag enclosed in parentheses. The specification

```
cs c gdp ma(1) ma(4)
```

will define an MA(5) model with coefficient restrictions as described above ("Seasonal ARMA Terms" on page 123). Note that in the absence of ordinary MA terms, the `sma` is equivalent to an `ma`. Thus,

```
cs c gdp ma(4)
cs c gdp sma(4)
```

are equivalent specifications.

Specifying Differencing

There are two distinct methods of specifying differencing in EViews:

- For integer differencing, you will apply the difference operator to the dependent and explanatory variables either before estimation, or by using series expressions in the equation specification.
- For fractional differencing, you will, include the `d` keyword in the by-list equation specification to indicate that the dependent and explanatory variables should be fractionally differenced.

Integer Differencing

The `d` operator may be used to specify integer 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).

Higher-order and seasonal differencing may be specified using the two optional parameters, n and s. `d(x,n)` specifies the n-th order difference of the series X:

$$d(x, n) = (1 - L)^n x, \tag{24.45}$$

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

```
d(gdp,2) = gdp - 2*gdp(-1) + gdp(-2)
```

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

$$d(x, n, s) = (1 - L)^n (1 - L^s) x. \tag{24.46}$$

For example, `d(gdp,0,4)` specifies zero ordinary differencing with a seasonal difference at lag 4, or GDP–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(GDP) or log(GDP)–log(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 ARIMA 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 first create the difference series by typing in the command line:

```
series dm1 = d(m1)
```

and then use this series when you enter your equation specification:

```
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:

```
d(m1) c ar(1) ma(1)
```

The latter method should generally be employed 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 produce 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.

Furthermore, if any other transformation or lags of the original series 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 with or without ARMA terms. Simply include the series expression in the list of regressors. 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.

Fractional Differencing

If you wish to perform fractional differencing as part of ARFIMA estimation, simply add the `d` keyword to the existing specification.

Note that fractional integration models may only be estimated in equations specified by list. You may not specify an ARFIMA model using expression.

Specification Examples

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 the dependent variable (INC) and your other regressors (in this case C and GOV):

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

Once again, you need not use AR and MA terms consecutively. For example, if you want to fit a fourth-order autoregressive model, you could use AR(4) by itself, resulting in a restricted ARMA(4, 0):

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

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

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

indicates an ARMA(0, 2) model for the errors of the INC equation.

The traditional Box-Jenkins or ARMA models do not have 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:

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

is a standard Box-Jenkins ARIMA (2, 1, 2) model for log INC.

You may specify an range of MA terms to include using the `to` keyword. The following ARFIMA(0, 1, 5) specification includes all of the MA terms from 1 to 5, along with the mean regressor DLOG(GDP):

```
dlog(inc) dlog(cs) c dlog(gdp) ma(1 to 5)
```

For equations specified by expression, simply enter the explicit equation involving the possibly differenced dependent variable, and add any expressions for AR and MA terms in square brackets:

```
dlog(cs) = c(1) + dlog(gdp)^c(2) + [ar(1)=c(3), ar(2)=c(4),
    ma(1)=c(5), ma(2)=c(6)]
```

To estimate an ARFIMA(2, d, 1) (fractionally integrated second-order autoregressive, first-order moving average error model), you would include expressions for the AR(1), AR(2), and MA(1) terms and the `d` keyword along with the dependent variable (INC) and other regressors (C and GOV):

```
log(inc) c log(gov) ar(1 to 2) ma(1) d
```

Estimation Options

Clicking on the **Options** tab displays a variety of estimation options. The available options will differ depending on whether your equation is specified by list or by expression and whether there are ARMA and fractional differencing components. For the remainder of this discussion, we will assume that you have included ARMA or fractional differencing in the equation specification, and we discuss in turn the settings available for each specification method.

Equations Specified By List

If your equation is specified by list, clicking on the **Options** tab displays a dialog page that offers settings for controlling the ARMA estimation, for computing the coefficient covariance, for optimization, and for setting the default coefficient name.

ARMA

The **ARMA** section of the page controls the method for estimating your ARMA components and setting starting values.

ARMA Method

The **Method** dropdown specifies the objective function used in the estimation method:

- For models without fractional differencing, you may choose between the default **ML** (maximum likelihood), **GLS** (generalized least squares), and **CLS** (conditional least squares) estimation.
- For models with fractional differencing, you may choose between the default **ML** and **GLS** estimation (**CLS** is not available for ARFIMA models).

See "Estimation Method Details" on page 160 for discussion of these objective functions.

Starting Values

The nonlinear estimation techniques used to estimate ARMA and ARFIMA models require starting values for all coefficient estimates. Normally, EViews determines its own starting values and for the most part this is an issue with which you need not be concerned. There are, however, occasions where 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 left over from previous estimation instructs EViews to continue from 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. You might also want to supply starting

values if you have a rough idea of what the answers should be, and want to speed up the estimation process.

The **Starting ARMA coefficient values** dropdown will offer choices for overriding the default EViews starting values. The available starting value options will differ depending on the ARMA method selected above:

- If you select ML or GLS estimation as your method, you will be presented with the choice of **Automatic**, **EViews fixed**, **Random**, and **User-specified**.

  For **User-specified**, all of the coefficients are taken from the values in the coefficient vector in the workfile as described below.

  For each of the remaining methods, the mean coefficients are obtained from simple OLS regression.

  The default EViews **Automatic** initializes the ARMA coefficients using least squares regression of residuals against lagged residuals (for AR terms) and innovations (for MA terms), where innovations are obtained by first regressing residuals against many lags of residuals. **EViews fixed** sets the ARMA coefficients to arbitrary fixed values of 0.0025 for ordinary ARMA and 0.01 for seasonal ARMA terms. **Random** generates randomized ARMA coefficients.

  For ARFIMA estimation, the fractional difference parameter is initialized using the Geweke and Porter-Hundlak (1983) log periodogram regression (**Automatic**), a fixed value of 0.1 (**EViews fixed**), or a randomly generated uniform $-[0.5, 0.5]$ (**Random**).

- If you select the **CLS** estimation method, the starting values dropdown will let you choose between **OLS/TLS, .8 x OLS/TSLS**, **.5 x OLS/TSLS**, **.3 x OLS/TSLS**, **Zero**, and **User-specified**.

  For the **User-specified** selection, all of the coefficients are initialized from the values in the coefficient vector in the workfile as described below.

For the variants of **OLS/TSLS**, EViews will initialize the mean coefficients at the specified fraction of the simple OLS or TSLS estimates while **Zero** sets the mean coefficients to zero.

Coefficients for ARMA terms are always set to arbitrary fixed values of 0.0025 for ordinary ARMA and 0.01 for seasonal ARMA terms.

For you to usefully set user-specified starting values, you will need a little more information about how EViews assigns coefficients for the ARMA terms.

EViews assigns coefficient numbers to the variables in the following order:

- First are the coefficients of the variables, in order of entry.
- Next is the ARFIMA coefficient.
- Next come the AR terms in the order of entry.
- The SAR, MA, and SMA coefficients follow, in that order.

(Following estimation, you can always see the assignment of coefficients by looking at the **Representations** view of your equation.)

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)
```

By default EViews uses the built-in C coefficient vector, but this may be overridden (see "Coefficient Name" on page 142). To set initial values, you may edit the corresponding elements of the coefficient vector in the workfile, or you may also assign values in the 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).

Backcasting

If your specification includes MA terms and the ARMA estimation method is CLS, EViews will display a checkbox for whether or not to use backcasting to initialize the MA innovations. By default, EViews performs backcasting as described in "Initializing MA Innovations" on page 164, but you can unselect this option to set the presample innovations to their unconditional expectation of zero.

Coefficient Covariance

The **Coefficient covariance** section of the page controls the computation of the estimates of the precision of your coefficient estimates.

The options that are available will depend on the ARMA estimation method.

- For ML or GLS estimation, covariances are always calculated by taking the inverse of the an estimate of the information matrix.

  The default setting for **Information matrix** estimation uses the outer product of the gradients (**OPG**), but you may instead use the dropdown to use the observed Hessian (**Hessian - observed**).

- For **CLS** estimation, you may choose a **Covariance method** using the dropdown menu.

  The default **Ordinary** method takes the inverse of the estimate of the information matrix. Alternately you may choose to compute **Huber-White** or **HAC (Newey-West)** sandwich covariances.

  In the latter case, EViews will display a **HAC options** button which you may use to access various settings for controlling the long-run covariance estimation.

  The **Information matrix** dropdown menu will offer you the choice between computing the information matrix estimate using the outer product of the gradients (**OPG**) or the observed Hessian (**Hessian - observed**).

If you select GLS or CLS estimation, the covariance matrix will, by default, employ a degree-of-freedom correction. If you select ML estimation the default computation will not employ degree-of-freedom correction. In all three cases, the **d.f. Adjustment** checkbox may be used to modify the computation.

Estimation Algorithm

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.

The **Estimation algorithm** section of the dialog contains settings for the numeric optimization of your likelihood or least squares objective function.

By default, EViews estimates ARMA and ARFIMA models using the Broyden, Fletcher, Goldfarb and Shanno (BFGS) algorithm. You may use the **Optimization method** dropdown to select a different method:

- For models estimated using ML and GLS, you may choose to estimate your model using **BFGS**, **OPG-BHHH** (Gauss-Newton using the outer-product of the gradient), **Kohn-Ansley** (transformation to pseudo-GLS regression model), and **Newton-Raphson**.
- For models estimated using CLS, you may choose between **BFGS**, **Gauss-Newton**, **Newton-Raphson**, and **EViews.** The latter employs OPG/BHHH with a Marquardt diagonal adjustment.
- For all but EViews, the **Step method** combo lets you choose between the default **Marquardt**, **Dogleg**, and **Line Search** determined steps. The default method is **Marquardt**.

In addition, you can use the **Maximum iterations** and **Convergence tolerance** edit fields to change the stopping rules from their global default settings. Checking the **Display settings** in output box instructs EViews to put information about starting values and other optimization settings at the top of your equation output.

Coefficient Name

For equations specified by list EViews will, by default, use the built-in C vector to hold coefficient estimates. You may change this assignment by entering the name of a coefficient object in the **Coefficient name** edit field.

If the coefficient does not exist, EViews will create it and size it appropriately. If the coefficient already exists, it will be resized if necessary so that it is large enough to hold the results. If an object of a different type with that name is present in the workfile, EViews will issue an error message.

Equation Specified By Expression

If your equation is specified by expression, clicking on the **Options** tab displays a dialog page that offers a subset of the settings that are available for equations specified by list.

You may use the page to control the computation of the coefficient covariance, the optimization method, the ARMA starting coefficient values, and the default coefficient name.

Coefficient Covariance

The **Coefficient covariance** section of the page allows you to specify a **Covariance method** using the dropdown menu. You may choose to compute the default **Ordinary**, or the **Huber-White**, or **HAC (Newey-West)** sandwich covariances. If you select **HAC (Newey-West)**, EViews will display a **HAC options** button which you may use to access various settings for controlling the long-run covariance estimation.

As before, the **Information matrix** dropdown menu will offer you the choice between computing the information matrix estimate using the outer product of the gradients (**OPG**) or the observed Hessian (**Hessian - observed**).

By default, EViews will apply a degree-of-freedom correction to the estimated covariance matrix. You may uncheck the **d.f. Adjustment** checkbox to remove this correction.

Estimation Algorithm

By default, EViews estimates by-expression ARMA and ARFIMA models using **BFGS**. You may use the **Optimization method** dropdown to choose between **BFGS**, **Gauss-Newton**, **Newton-Raphson**, and **EViews,** the latter of which employs Gauss-Newton with a Marquardt diagonal adjustment.

Where appropriate, the **Step method** combo lets you choose between the default **Marquardt**, **Dogleg**, and **Line Search** determined steps.

The **Maximum iterations** and **Convergence tolerance** edit fields may be used to limit the number of iterations and to set the algorithm stopping rule. Checking the **Display settings** in output box instructs EViews to put information about starting values and other optimization settings at the top of your equation output.

ARMA Method

You will not be able to specify an ARMA method as ARMA equations specified by expression may only use the CLS objective.

Starting Values

The starting value dropdown menu lets you choose between the default **OLS/TLS,** and **.8 x OLS/TSLS**, **.5 x OLS/TSLS**, **.3 x OLS/TSLS**, **Zero**, and **User-specified**.

For the variants of **OLS/TSLS**, EViews will initialize the mean coefficients at the specified fraction of the simple OLS or TSLS estimates (ignoring ARMA terms), while **Zero** sets the mean coefficients to zero. Coefficients for ARMA terms are always set to arbitrary fixed values of 0.0025 for ordinary ARMA and 0.01 for seasonal ARMA terms.

For the **User-specified** selection, the coefficients are initialized from the values in the coefficient vector in the workfile.

Estimation Output

EViews displays a variety of results in the output view following estimation.

The top portion of the output displays information about the optimization technique, ARMA estimation method, the coefficient covariance calculation, and if requested, the starting values used to initialize the optimization procedure.

```
Dependent Variable: DLOG(GNP)
Method: ARMA Maximum Likelihood (BFGS)
Date: 02/06/15   Time: 10:20
Sample: 1947Q2 1989Q4
Included observations: 171
Convergence achieved after 8 iterations
Coefficient covariance computed using outer product of gradients
```

The next section shows the estimated coefficients, coefficient standard errors, and *t*-statistics. In addition to the estimates of the ARMA coefficients, EViews will display estimates of the fractional integration parameter for ARFIMA models, and the estimate of the error variance if the ARMA estimation method is maximum likelihood, labeled SIGMASQ.

Variable	Coefficient	Std. Error	t-Statistic	Prob.
C	0.008051	0.003814	2.111053	0.0362
D	0.288756	0.057164	5.051407	0.0000

All of these results may be interpreted in the usual manner.

In the section directly below the coefficient estimates are the usual descriptive statistics for the dependent variable, along with a variety of summary and descriptive statistics for the estimated equation.

R-squared	0.099981	Mean dependent var	0.008032
Adjusted R-squared	0.094656	S.D. dependent var	0.010760
S.E. of regression	0.010238	Akaike info criterion	-6.312490
Sum squared resid	0.017713	Schwarz criterion	-6.275746
Log likelihood	541.7179	Hannan-Quinn criter.	-6.297581
F-statistic	18.77385	Durbin-Watson stat	1.824729
Prob(F-statistic)	0.000025		

Note that all of the equation summary results involving residuals differ from those computed in standard OLS settings so that some care should be taken in interpreting results. To understand the issues, keep in mind that there are two different residuals associated with an ARMA model. The first are the estimated *unconditional residuals*:

$$\hat{u}_t = Y_t - X_t'\hat{\beta}, \tag{24.47}$$

which are computed using the original explanatory variables and the estimated coefficients, $\hat{\beta}$. These residuals are the errors that you would obtain if you made a prediction of the value of Y_t using contemporaneous information while ignoring the information contained in the lagged residuals.

Generally, there is little reason to examine the unconditional residuals, and EViews does not automatically compute them following estimation.

The second set of residuals are the estimated *one-period ahead forecast errors*, ϵ. 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 ARMA models, the computed residuals, and all of 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 estimated one-period ahead forecast errors, $\hat{\epsilon}$.

Lastly, to aid in the interpretation of the results for ARMA and ARFIMA models, EViews displays a the reciprocal roots of the AR and MA polynomials in the lower block of the results. EViews reports these roots as **Inverted AR Roots** and **Inverted MA Roots** at the bottom of the regression output. For our general ARMA model with the lag polynomials $\rho(L)$ and $\theta(L)$, the reported roots are the roots of the polynomials:

$$\rho(x^{-1}) = 0 \qquad \text{and} \qquad \theta(x^{-1}) = 0 . \tag{24.48}$$

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.

For example, in 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).

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 try to 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 "Initializing MA Innovations" on page 164).

If the estimated MA process has roots with modulus close to one, it is a sign that you may have over-differenced the data, which introduced an MA unit root. 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, perhaps using ARFIMA to account for long-run dependence.

Consider the following example output from ARMA estimation:

Dependent Variable: CP
Method: ARMA Maximum Likelihood (BFGS)
Date: 03/01/15 Time: 15:25
Sample: 1954M01 1993M07
Included observations: 475
Convergence achieved after 117 iterations
Coefficient covariance computed using outer product of gradients

Variable	Coefficient	Std. Error	t-Statistic	Prob.
C	5.836704	1.750241	3.334801	0.0009
AR(1)	0.973815	0.007755	125.5649	0.0000
SAR(4)	0.225555	0.049713	4.537146	0.0000
MA(1)	0.466481	0.016635	28.04168	0.0000
MA(4)	-0.344940	0.043602	-7.911135	0.0000
SIGMASQ	0.249337	0.007393	33.72769	0.0000

R-squared	0.974433	Mean dependent var	6.331116
Adjusted R-squared	0.974161	S.D. dependent var	3.126173
S.E. of regression	0.502520	Akaike info criterion	1.483257
Sum squared resid	118.4349	Schwarz criterion	1.535846
Log likelihood	-346.2736	Hannan-Quinn criter.	1.503938
F-statistic	3575.030	Durbin-Watson stat	1.986429
Prob(F-statistic)	0.000000		

Inverted AR Roots	.97	.69	.00-.69i	-.00+.69i
	-.69			
Inverted MA Roots	.67	-.11+.74i	-.11-.74i	-.92

This estimation result corresponds to the following specification,

$$\begin{aligned} y_t &= 5.84 + u_t \\ (1-0.97L)(1-0.23L^4)u_t &= (1+0.47L)(1-0.34L^4)\epsilon_t \end{aligned} \tag{24.49}$$

or equivalently, to:

$$\begin{aligned} y_t = {} & 0.118 + 0.97y_{t-1} + 0.23y_{t-4} - 0.22y_{t-5} + \epsilon_t \\ & + 0.47\epsilon_{t-1} - 0.34\epsilon_{t-4} - 0.17\epsilon_{t-5} \end{aligned} \tag{24.50}$$

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

Equation Diagnostics

In addition to the usual views and procs for an equation such as coefficient confidence ellipses, Wald tests, omitted and redundant variables tests, EViews offers diagnostics for examining the properties of your ARMA model and the properties of the estimated innovations.

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 $a = \operatorname{atan}(i/r)$, 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) MA(4)
Date: 03/01/15 Time: 16:19
Sample: 1954M01 1994M12
Included observations: 470

AR Root(s)	Modulus	Cycle
0.987785	0.987785	
0.617381	0.617381	
-0.617381	0.617381	
2.60e-17 ± 0.617381i	0.617381	4.000000

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

MA Root(s)	Modulus	Cycle
-0.815844	0.815844	
-0.112642 ± 0.619634i	0.629790	3.589119
0.557503	0.557503	

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 interpreted 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.

ARMA Frequency Spectrum

The ARMA frequency spectrum view of an ARMA equation shows the spectrum of the estimated ARMA terms in the frequency domain, rather than the typical time domain. Whereas viewing the ARMA terms in the time domain lets you view the autocorrelation functions of the data, viewing them in the frequency domain lets you observe more complicated cyclical characteristics.

The spectrum of an ARMA process can be written as a function of its frequency, λ, where λ is measured in radians, and thus takes values from $-\pi$ to π. However since the spectrum is symmetric around 0, it is EViews displays it in the range $[0, \pi]$.

To show the frequency spectrum, select **View/ARMA Structure...** from the equation toolbar, choose **Frequency spectrum** from the **Select a diagnostic** list box, and then select a display format (**Graph** or **Table**).

If a series is white noise, the frequency spectrum should be flat, that is a horizontal line. Here we display the graph of a series generated as random normals, and indeed, the graph is approximately a flat line.

If a series has strong AR components, the shape of the frequency spectrum will contain peaks at points of high cyclical frequencies. Here we show a typical AR(2) model, where the data were generated such that $\rho_1 = 0.7$ and $\rho_2 = -0.5$.

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 128, more general tests for serial correlation in the residuals may be carried out with **View/Residual Diagnostics/Correlogram-Q-statistic** and **View/Residual Diagnostics/Serial Correlation LM Test…**.

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

Autocorrelation	Partial Correlation		AC	PAC	Q-Stat	Prob
		1	-0.010	-0.010	0.0506	
		2	-0.032	-0.033	0.5566	
		3	-0.060	-0.060	2.2667	
		4	0.022	0.020	2.4994	
		5	0.158	0.155	14.528	0.000
		6	-0.178	-0.182	29.840	0.000
		7	-0.126	-0.123	37.560	0.000
		8	-0.030	-0.021	37.986	0.000
		9	0.161	0.141	50.643	0.000
		10	-0.022	-0.058	50.875	0.000
		11	0.052	0.118	52.195	0.000
		12	-0.022	-0.001	52.426	0.000

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.

Examples

To illustrate the estimation of ARIMA and ARFIMA specifications in EViews we consider examples from Sowell (1992a) which model the natural logarithm of postwar quarterly U.S. real GDP from 1947q1 to 1989q4. Sowell estimates a number of models which are compared using AIC and SIC. We will focus on the ARMA(3, 2) and ARFIMA(3, d, 2) specifications (Table 2, p. 288 and Table 3, p. 289).

To estimate the ARMA(3, 2) we open an equation dialog by selecting **Object/New Object/Equation**, by selecting **Quick/Estimate Equation...**, or by typing the command keyword `equation` in the command line. EViews will display the least squares dialog:

We enter the expression for the dependent variable, followed by the AR and MA terms using ranges that include all of the desired terms, and C to indicate that we wish to include an intercept. Next, we click on the **Options** tab to display the estimation settings.

First, we instruct EViews to compute coefficient standard errors using the observed Hessian by setting the Information matrix dropdown to **Hessian - observed**. In addition, we set the **Optimization method** to **BFGS**, the **Convergence tolerance** to "1e-8", and the **ARMA Method** to **ML**. Click on **OK** to estimate the model.

EViews will perform the iterative maximum likelihood estimation using BFGS and will display the estimation results:

Dependent Variable: DLOG(GNP)
Method: ARMA Maximum Likelihood (BFGS)
Date: 03/01/15 Time: 20:18
Sample: 1947Q2 1989Q4
Included observations: 171
Convergence achieved after 18 iterations
Coefficient covariance computed using observed Hessian

Variable	Coefficient	Std. Error	t-Statistic	Prob.
C	0.008008	0.001189	6.732176	0.0000
AR(1)	0.599278	0.148087	4.046797	0.0001
AR(2)	-0.671335	0.178976	-3.750983	0.0002
AR(3)	0.137677	0.104632	1.315814	0.1901
MA(1)	-0.277643	0.121744	-2.280545	0.0239
MA(2)	0.793977	0.118172	6.718800	0.0000
SIGMASQ	9.24E-05	9.99E-06	9.245754	0.0000

R-squared	0.197098	Mean dependent var	0.008032
Adjusted R-squared	0.167724	S.D. dependent var	0.010760
S.E. of regression	0.009816	Akaike info criterion	-6.366372
Sum squared resid	0.015801	Schwarz criterion	-6.237766
Log likelihood	551.3248	Hannan-Quinn criter.	-6.314189
F-statistic	6.709856	Durbin-Watson stat	1.998281
Prob(F-statistic)	0.000002		

Inverted AR Roots	.24	.18+.74i	.18-.74i
Inverted MA Roots	.14+.88i	.14-.88i	

The top portion of the output displays information about the estimation method, optimization, and covariance calculation.

The next section contains the coefficient estimates, standard errors, t-statistics and corresponding p-value. (It is worth pointing out that the reported ARMA coefficients use a different sign convention than those in Sowell so that the ARMA coefficients all have the opposite sign).

Notice that since we estimated the model using ML, EViews displays the estimate of the error variance as one of the estimated coefficients. You should be aware that the EViews reported p-value for SIGMASQ is for the two-sided test, despite the fact that SIGMASQ must be non-negative. (If desired, you may use the reported coefficient, standard error, and the @CTDIST function to compute the appropriate one-sided p-value.)

The final section shows the inverted AR and MA roots.

It may be instructive to compare these results to those obtained from an alternative conditional least squares approach to estimating the specification. To reestimate your equation using CLS, click on the **Estimate** button to bring up the dialog, then on the **Options** tab to show the estimation options. In the **ARMA** section of the page, we have:

Select **CLS** in the **Method** dropdown, and click on **OK** to estimate the new specification.

Click on **OK** to accept the changes and re-estimate the model.

Dependent Variable: DLOG(GNP)
Method: ARMA Conditional Least Squares (BFGS / Marquardt steps)
Date: 03/01/15 Time: 21:00
Sample (adjusted): 1948Q1 1989Q4
Included observations: 168 after adjustments
Convergence achieved after 38 iterations
Coefficient covariance computed using observed Hessian
MA Backcast: 1947Q3 1947Q4

Variable	Coefficient	Std. Error	t-Statistic	Prob.
C	0.007994	0.001254	6.373517	0.0000
AR(1)	0.563811	0.176602	3.192556	0.0017
AR(2)	-0.673101	0.161797	-4.160159	0.0001
AR(3)	0.158506	0.108283	1.463812	0.1452
MA(1)	-0.242197	0.153644	-1.576346	0.1169
MA(2)	0.814550	0.098533	8.266750	0.0000

R-squared	0.200837	Mean dependent var	0.008045
Adjusted R-squared	0.176172	S.D. dependent var	0.010845
S.E. of regression	0.009844	Akaike info criterion	-6.368908
Sum squared resid	0.015698	Schwarz criterion	-6.257337
Log likelihood	540.9882	Hannan-Quinn criter.	-6.323627
F-statistic	8.142440	Durbin-Watson stat	1.994105
Prob(F-statistic)	0.000001		

Inverted AR Roots	.27	.15-.76i	.15+.76i
Inverted MA Roots	.12-.89i	.12+.89i	

The top of the new equation output now reports that estimation was performed using CLS and that the MA errors were initialized using backcasting. Despite the different objectives, we see that the CLS ARMA coefficient estimates are generally quite similar to those obtained from exact ML estimation. Lastly, we note that the estimate of the variance is not reported as part of the coefficient output for CLS estimation.

Next, following Sowell, we estimate an ARFIMA$(3, d, 2)$. Once again, click on the **Estimate** button to bring up the dialog:

and add the special `d` keyword to the list of regressors to tell EViews that you wish to estimate the fractional integration parameter. Click on **OK** to estimate the updated equation.

Dependent Variable: DLOG(GNP)
Method: ARMA Maximum Likelihood (BFGS)
Date: 03/01/15 Time: 21:18
Sample: 1947Q2 1989Q4
Included observations: 171
Convergence achieved after 29 iterations
Coefficient covariance computed using observed Hessian

Variable	Coefficient	Std. Error	t-Statistic	Prob.
C	0.007886	0.000373	21.15945	0.0000
D	-0.606793	0.306851	-1.977481	0.0497
AR(1)	1.195165	0.351233	3.402768	0.0008
AR(2)	-0.939049	0.295641	-3.176311	0.0018
AR(3)	0.516754	0.178254	2.898971	0.0043
MA(1)	-0.291411	0.125001	-2.331272	0.0210
MA(2)	0.811038	0.114772	7.066532	0.0000
SIGMASQ	9.02E-05	9.76E-06	9.239475	0.0000

R-squared	0.216684	Mean dependent var	0.008032
Adjusted R-squared	0.183044	S.D. dependent var	0.010760
S.E. of regression	0.009725	Akaike info criterion	-6.373358
Sum squared resid	0.015416	Schwarz criterion	-6.226380
Log likelihood	552.9221	Hannan-Quinn criter.	-6.313720
F-statistic	6.441378	Durbin-Watson stat	1.995509
Prob(F-statistic)	0.000001		

Inverted AR Roots	.82	.19+.77i	.19-.77i
Inverted MA Roots	.15-.89i	.15+.89i	

Notice first that EViews has switched from CLS estimation to ML since ARFIMA models may only be estimated using ML or GLS.

Turning to the estimate of the fractional differencing parameter, we see that it is negative and statistically significantly different from zero at the 5% level. Thus, we can reject the unit root hypothesis under this specification. Alternately, we cannot reject the time trend null hypothesis that $d = -1.0$.

(Note: the results reported in Sowell differ slightly, presumably due to differences in the nonlinear optimization procedure in general, and the estimate of the observed Hessian in particular—for what it is worth, the EViews likelihood is slightly higher than the likelihood reported by Sowell. Notably, Sowell's conclusions differ slightly from than those outlined here, as he finds that the unit root and trend hypotheses are both consistent with the ARFIMA estimates. Sowell does not reject the zero null at the 5% level, but does reject at the 10% level. See Sowell for detailed interpretation of results.)

Additional Topics

Dealing with Estimation Problems

Since EViews uses nonlinear estimation algorithms to estimate ARMA models, all of the discussion in Chapter 21, "Solving Estimation Problems" on page 59, 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 and ARFIMA 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 autocorrelation 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.

For a discussion of how to estimate TSLS specifications with ARMA errors, see "Nonlinear Two-stage Least Squares" on page 96.

Nonlinear Models with ARMA errors

EViews will estimate nonlinear ordinary and two-stage least squares models with autoregressive error terms. For details, see the discussion in "Nonlinear Least Squares," beginning on page 51.

Weighted Models with ARMA errors

EViews does not offer built-in procedures to automatically estimate weighted models with ARMA error terms. You can, of course, always construct weighted series and then perform estimation using the weighted data and ARMA terms. Note that this procedure implies a very specific assumption about the properties of your data.

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 93.

Nonlinear Models with ARMA Errors

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

$$\begin{aligned} CS_t &= c_1 + GDP_t^{c_2} + u_t \\ u_t &= c_3 u_{t-1} + c_4 u_{t-2} + \epsilon_t \end{aligned} \tag{24.51}$$

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^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 "Initializing the AR Errors" on page 162).

Estimation Method Details

In "ARMA Method" on page 138 we described how EViews lets you choose between maximum likelihood (ML), generalized least squares (GLS), and conditional least squares (CLS) estimation for ARIMA and ARFIMA estimation.

Recall that for the general ARIMA(p, d, q) model we have

$$\begin{aligned} \rho(L)(1-L)^d(Y_t-\mu_t) &= \theta(L)\epsilon_t \\ \rho(L)u_t &= \theta(L)\epsilon_t \end{aligned} \tag{24.52}$$

for the unconditional residuals

$$\begin{aligned} u_t &= (1-L)^d(Y_t-\mu_t) \\ &= \nabla^d Y_t - \nabla^d X_t'\beta \end{aligned} \tag{24.53}$$

and innovations

$$\epsilon_t = u_t - \rho_1 u_{t-1} - \ldots - \rho_p u_{t-p} + \theta_1\epsilon_{t-1} + \theta_q\epsilon_{t-q} \tag{24.54}$$

We will use the expressions for the unconditional residuals and innovations to describe three objective functions that may be used to estimate the ARIMA model.

(For simplicity of notation our discussion abstracts from SAR and SMA terms and coefficients. It is straightforward to allow for the inclusion of these seasonal terms).

Maximum Likelihood (ML)

Estimation of ARIMA and ARFIMA models is often performed by exact maximize likelihood assuming Gaussian innovations.

The exact Gaussian likelihood function for an ARIMA or ARFIMA model is given by

$$\begin{aligned} \log L(\beta, \rho, \theta, \sigma^2, d) &= -\frac{T}{2}\log(2\pi) - \frac{1}{2}\log|\Omega| - \frac{1}{2}u'\Omega^{-1}u \\ &= -\frac{T}{2}\log(2\pi) - \frac{1}{2}\log|\Omega| - S(\beta, \rho, \theta, d) \end{aligned} \tag{24.55}$$

where $Y = (Y_1, Y_2, \ldots, Y_T)'$ and $u = (u_1, u_2, \ldots, u_T)'$, where Ω is the symmetric Toeplitz covariance matrix for the T draws from the ARMA/ARFIMA process for the unconditional residuals (Doornik and Ooms 2003). Note that direct evaluation of this function requires the inversion of a large $T \times T$ matrix Ω which is impractical for large T for both storage and computational reasons.

The ARIMA model restricts d to be a known integer. The ARFIMA model treats d as an estimable parameter.

ARIMA ML

It is well-known that for ARIMA models where d is a known integer, we may employ the Kalman filter to efficiently evaluate the likelihood. The Kalman filter works with the state space prediction error decomposition form of the likelihood, which eliminates the need to invert the large matrix Ω.

See Hamilton (2004, Chapter 13, p. 372) or Box, Jenkins, and Reinsel (2008, 7.4, p. 275) for extensive discussion.

ARFIMA ML

Sowell (1992) and Doornik and Ooms (2003) offer detailed descriptions of the evaluation of the likelihood for ARFIMA models. In particular, practical evaluation of Equation (24.55) requires that we address several computational issues.

First, we must compute the autocovariances of the ARFIMA process that appear in the Ω which an involve an infinite order MA representation. Fortunately, Hosking (1981) and Sowell (1992) describe closed-form alternatives and Sowell (1992) derives efficient recursive algorithms using hypergeometric functions.

Second, we must compute the determinant of the variance matrix and generalized (inverse variance weighted) residuals in a manner that is computationally and storage efficient. Doornik and Ooms (2003) describe a Levinson-Durbin algorithm for efficiently performing this operation with minimal operation count while eliminating the need to store the full $T \times T$ matrix Ω.

Third, where possible we follow Doornik and Ooms (2003) in concentrate the likelihood with respect to the regression coefficients β and the scale parameter σ^2.

Generalized Least Squares (GLS)

Since the exact likelihood function in Equation (24.55) depends on the data, and the mean and ARMA parameters only through the last term in the expression, we may ignore the inessential constants and the log determinant term to define a generalized least squares objective function

$$S(\beta, \rho, \theta, d) = u'\Omega^{-1}u \tag{24.56}$$

and the ARFIMA estimates may be obtained by minimizing $S(\beta, \rho, \theta, d)$.

Conditional Least Squares (CLS)

Box and Jenkins (1976) and Box, Jenkins, and Reinsel (2008, Section 7.1.2 p 232.) point out that conditional on pre-sample values for the AR and MA errors, the normal conditional likelihood function may be maximized by minimizing the sum of squares of the innovations.

The recursive innovation equation in Equation (24.54) is easy to evaluate given parameter values, lagged values of the differenced Y_t, X_t, and estimates of the lagged innovations. Note, however that neither the u_t nor the can be substituted in the first period as they are not available until we start up the difference equation.

We discuss below methods for starting up the recursion by specifying presample values of u_t and ϵ_t. Given these presample values, the conditional likelihood function for normally distributed innovations is given by

$$\begin{aligned} \log l(\beta, \rho, \theta, \sigma^2) &= -\frac{T}{2}\log(2\pi\sigma^2) - \frac{1}{2\sigma^2}\sum_{t=1}^{T}\epsilon_t^2 \\ &= -\frac{T}{2}\log(2\pi\sigma^2) - \frac{1}{2\sigma^2}S(\beta, \rho, \theta) \end{aligned} \tag{24.57}$$

Notice that the conditional likelihood function depends on the data and the mean and ARMA parameters only through the conditional least squares function $S(\beta, \rho, \theta)$, so that the conditional likelihood may be maximized by minimizing $S(\beta, \rho, \theta)$.

Coefficient standard errors for the CLS estimation are the same as those for any other non-linear least squares routine: ordinary inverse of the estimate of the information matrix, or a White robust or Newey-West HAC sandwich covariance estimator. In all three cases, one can use either the Gauss-Newton outer-product of the Jacobians, or the Newton-Raphson negative of the Hessian to estimate the information matrix.

In the remainder of this section we discuss the initialization of the recursion. EViews initializes the AR errors using lagged data (adjusting the estimation sample if necessary), and initializes the MA innovations using backcasting or the unconditional (zero) expectation.

Initializing the AR Errors

Consider an AR(p) regression model of the form:

$$\begin{aligned} Y_t &= X_t'\beta + u_t \\ u_t &= \rho_1 u_{t-1} + \rho_2 u_{t-2} + \ldots + \rho_p u_{t-p} + \epsilon_t \end{aligned} \tag{24.58}$$

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

We can rewrite out model as

$$\epsilon_t = Y_t - X_t'\beta - (\rho_1 u_{t-1} + \rho_2 u_{t-2} + \ldots + \rho_p u_{t-p}) \tag{24.59}$$

so we can see that we require p pre-sample values to evaluate the AR process at $t = 1$

Typically conditional least squares employs lagged values of the variables in the model to initialize the process. For example, to estimate an AR(1) model, one may transforms the linear model,

$$\begin{aligned} Y_t &= X_t'\beta + u_t \\ u_t &= \rho u_{t-1} + \epsilon_t \end{aligned} \tag{24.60}$$

into a nonlinear model by substituting the second equation into the first, writing u_{t-1} in terms of observables and rearranging terms:

$$\begin{aligned} Y_t &= X_t'\beta + \rho u_{t-1} + \epsilon_t \\ &= X_t'\beta + \rho(Y_{t-1} - X_{t-1}'\beta) + \epsilon_t \\ &= \rho Y_{t-1} + (X_t - \rho X_{t-1})'\beta + \epsilon_t \end{aligned} \tag{24.61}$$

so that the innovation recursion written in terms of observables is given by

$$\epsilon_t = (Y_t - \rho Y_{t-1}) - (X_t - \rho X_{t-1})'\beta \tag{24.62}$$

Notice that we require observation on the Y_t and X_t in the period before the start of the recursion. If these values are not available, we must adjust the period of interest to begin at $t = 2$ so that the values of the observed data in $t = 1$ may be substituted into the equation to obtain an expression for u_1.

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

$$\begin{aligned} Y_t = {} & (\rho_1 Y_{t-1} + \rho_2 Y_{t-2} + \rho_3 Y_{t-3}) + f(X_t, \beta) - \rho_1 f(X_{t-1}, \beta) \\ & - \rho_2 f(X_{t-2}, \beta) - \rho_3 f(X_{t-3}, \beta) + \epsilon_t \end{aligned} \tag{24.63}$$

It is important to note that textbooks often describe techniques for estimating linear AR models like Equation (24.58). 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. These approaches proceed by obtaining an initial consistent estimate of the AR coefficients ρ and then estimating the remaining coefficients via a second-stage 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 (2008, p. 648–652).

In contrast, the EViews conditional least squares estimates the coefficients ρ and β are estimated simultaneously by minimizing the nonlinear sum-of-squares function $S(\beta, \rho, \theta)$ (which maximizes the conditional likelihood). The nonlinear least squares approach has the advantage of being easy-to-understand, generally applicable, and easily extended to models that contain endogenous right-hand side variables and to nonlinear mean specifications.

Thus, for a nonlinear mean AR(1) specification, EViews transforms the nonlinear model,

$$\begin{aligned} Y_t &= f(X_t, \beta) + u_t \\ u_t &= \rho u_{t-1} + \epsilon_t \end{aligned} \tag{24.64}$$

into the alternative nonlinear regression form

$$Y_t = \rho Y_{t-1} + f(X_t, \beta) - \rho f(X_{t-1}, \beta) + \epsilon_t \tag{24.65}$$

yielding the innovation specification:

$$\epsilon_t = (Y_t - \rho Y_{t-1}) - (f(X_t, \beta) - \rho f(X_{t-1}, \beta)) \tag{24.66}$$

Similarly, for higher order ARs, we have:

$$\begin{aligned} \epsilon_t = \; & Y_t - (\rho_1 Y_{t-1} + \rho_2 Y_{t-2} + \rho_3 Y_{t-3}) + f(X_t, \beta) \\ & - (\rho_1 f(X_{t-1}, \beta) + \rho_2 f(X_{t-2}, \beta) + \rho_3 f(X_{t-3}, \beta)) \end{aligned} \tag{24.67}$$

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

Initializing MA Innovations

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

$$\begin{aligned} Y_t &= X_t'\beta + u_t \\ u_t &= \epsilon_t + \theta_1 \epsilon_{t-1} + \theta_2 \epsilon_{t-2} + \ldots + \theta_q \epsilon_{t-q} \end{aligned} \tag{24.68}$$

for $t = 1, 2, \ldots, T$. 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:

$$\{\hat{\epsilon}_{-(q-1)}, \hat{\epsilon}_{-(q-2)}, \ldots, \hat{\epsilon}_0\} \tag{24.69}$$

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:

$$\hat{\epsilon}_t = \hat{u}_t - \hat{\theta}_1 \hat{\epsilon}_{t-1} - \ldots - \hat{\theta}_q \hat{\epsilon}_{t-q} \tag{24.70}$$

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

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

$$\{\hat{\epsilon}_{-(q-1)}, \hat{\epsilon}_{-(q-2)}, \ldots, \hat{\epsilon}_0\} \tag{24.71}$$

One may employ 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:

$$\tilde{\epsilon}_{T+1} = \tilde{\epsilon}_{T+2} = \ldots = \tilde{\epsilon}_{T+q} = 0 \tag{24.72}$$

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

$$\tilde{\epsilon}_t = \hat{u}_t - \theta_1 \tilde{\epsilon}_{t+1} - \ldots - \theta_q \tilde{\epsilon}_{t+q} \tag{24.73}$$

for $t = T, \ldots, 0, \ldots, -(q-1)$. The final q values, $\{\tilde{\epsilon}_0, \ldots, \tilde{\epsilon}_{-(q-2)}, \tilde{\epsilon}_{-(q-1)}\}$, 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.)

Alternately, one obvious method is to turn backcasting off and to set the pre-sample ϵ to their unconditional expected values of 0:

$$\hat{\epsilon}_{-(q-1)} = \ldots = \hat{\epsilon}_0 = 0, \tag{24.74}$$

Whichever methods is used to initialize the presample values, the sum-of-squared residuals (SSR) is formed recursively as a function of the β and θ, using the fitted values of the lagged innovations:

$$S(\beta, \theta) = \sum_{t=q+1}^{T} (Y_t - X_t'\beta - \theta_1 \hat{\epsilon}_{t-1} - \ldots - \theta_q \hat{\epsilon}_{t-q})^2 . \tag{24.75}$$

and the 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.

References

Baille, Richard (1996). "Long Memory Processes and Fractional Integration in Econometrics," *Journal of Econometrics*, 73, 5–59..

Box, George E. P. and Gwilym M. Jenkins (1976). *Time Series Analysis: Forecasting and Control*, Revised Edition, Oakland, CA: Holden-Day.

Box, George E.P., Jenkins, Gwilym M., and Gregory C. Reinsel (2008). *Time Series Analysis: Forecasting and Control*, Fourth Edition, Hoboken, New Jersey: John Wiley & Sons.

Doornik, Jurgen A. and Marius Ooms (2003). "Computational Aspects of Maximum Likelihood Estimation of Autoregressive Fractionally Integrated Moving Average Models," *Computational Statistics & Data Analysis*, 42, 333–348.

Fair, Ray C. (1984). *Specification, Estimation, and Analysis of Macroeconometric Models*, Cambridge, MA: Harvard University Press.

Geweke, J. F. and S. Porter-Hudak (1983). "The Estimation and Application of Long Memory Time Series Models" *Journal of Time Series Analysis*, 4, 221–238.

Granger, C. W. J. and Roselyne Joyeux (1980). "An Introduction to Long-Memory Time Series Models and Fractional Differencing," *Journal of Time Series Analysis,* 1, 15–29.

Greene, William H. (2008). *Econometric Analysis*, 6th Edition, Upper Saddle River, NJ: Prentice-Hall.

Hamilton, James D. (1994). *Time Series Analysis*, Princeton University Press.

Hayashi, Fumio. (2000). *Econometrics*, Princeton, NJ: Princeton University Press.

Hosking, J. R. M. (1981). "Fractional Differencing," *Biometrika*, 68, 165–176.

Johnston, Jack and John Enrico DiNardo (1997). *Econometric Methods*, 4th Edition, New York: McGraw-Hill.

Kohn, Robert and Craig F. Ansley (1985). "Efficient Estimation and Prediction in Time Series Regression Models," *Biometrika*, 72, 694–697.

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.

Sowell, Fallaw (1992). "Maximum Likelihood Estimation of Stationary Univariate Fractionally Integrated Time Series Models," *Journal of Econometrics*, 53, 165–188.

Sowell, Fallaw (1992a). "Modeling Long-run Behavior with the Fractional ARIMA Model," *Journal of Monetary Economics*, 29, 277–302.

Chapter 25. 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 553 of *User's Guide I*, and forecasting using multiple equations and models is described in Chapter 49. "Models," on page 995. Additional tools for performing forecast evaluation are described in "Forecast Evaluation" on page 466 of *User's Guide I*.

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 1959M01–1996M0. The data are contained in the workfile "House1.WF1" which contains observations for 1959M01–1998M12 so that we may perform out-of-sample forecasts.

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: 08/09/09 Time: 07:45
Sample (adjusted): 1959M03 1990M01
Included observations: 371 after adjustments
Convergence achieved after 6 iterations

Variable	Coefficient	Std. Error	t-Statistic	Prob.
C	0.321924	0.117278	2.744973	0.0063
HS(-1)	0.952653	0.016218	58.74151	0.0000
SP	0.005222	0.007588	0.688248	0.4917
AR(1)	-0.271254	0.052114	-5.205025	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 187.

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 27. "ARCH and GARCH Estimation," on page 263 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 180 and "Forecasting with ARMA Errors" on page 182). 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 (with either just the forecast values, or forecast values alongside actuals) 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 187 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. Select HS and HSF in the workfile window, then right-mouse click and select **Open/as Group**. Then select **View/Graph...** and select **Line & Symbol** in the **Graph Type/Basic type** page to display a graph of 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. Make certain that the forecast sample is set to "1959m01 1995m06". Click on **OK**. 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 175, 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 c x 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:

$$\hat{y}_t = \hat{c}(1) + \hat{c}(2)x_t + \hat{c}(3)z_t. \tag{25.1}$$

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 c y(-1) 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:

$$y_t = x_t'\beta + \epsilon_t, \tag{25.2}$$

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:

$$\hat{y}_t = x_t' b. \tag{25.3}$$

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 nonzero; 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:

$$\text{forecast se} = s\sqrt{1 + x_t'(X'X)^{-1}x_t} \tag{25.4}$$

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 180).

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 193).

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 eval-

uations in their own table, you should turn off the **Do graph** option in the Forecast dialog box.

Suppose the forecast sample is $j = T+1, T+2, \ldots, T+h$, 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	$\sqrt{\sum_{t=T+1}^{T+h} (\hat{y}_t - y_t)^2 / h}$
Mean Absolute Error	$\sum_{t=T+1}^{T+h} \lvert \hat{y}_t - y_t \rvert / h$
Mean Absolute Percentage Error	$100 \sum_{t=T+1}^{T+h} \left\lvert \frac{\hat{y}_t - y_t}{y_t} \right\rvert / h$
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:

$$\sum (\hat{y}_t - y_t)^2 / h = ((\sum \hat{y}_t / h) - \bar{y})^2 + (s_{\hat{y}} - s_y)^2 + 2(1-r) s_{\hat{y}} s_y \tag{25.5}$$

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	$\frac{(s_{\bar{y}} - s_y)^2}{\sum(\hat{y}_t - y_t)^2 / h}$
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:

```
smpl 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 x z 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:

$$\hat{y}_S = \hat{c}(1) + \hat{c}(2)x_S + \hat{c}(3)z_S + \hat{c}(4)y_{S-1}, \tag{25.6}$$

  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:

$$\hat{y}_{S+k} = \hat{c}(1) + \hat{c}(2)x_{S+k} + \hat{c}(3)z_{S+k} + \hat{c}(4)\hat{y}_{S+k-1}. \tag{25.7}$$

- 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.

Lastly, we note that for non-linear dynamic forecasting, EViews produces what Tong and Lim (1980) term the "eventual forecasting function" in which the lagged forecasted values are substituted recursively into the one-step ahead function. This approach differs from the simulation based approaches to multi-step forecasting which employs stochastic simulation. If you wish to obtain the latter forecasts, click on the **Stochastic simulation** checkbox and enter the number of **Repetitions** and **Failed reps prop. before halting** as desired.

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:

$$\hat{y}_{S+k} = \hat{c}(1) + \hat{c}(2)x_{S+k} + \hat{c}(3)z_{S+k} + \hat{c}(4)y_{S+k-1} \tag{25.8}$$

  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 175).

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:

$$\begin{aligned} y_t &= x_t'\beta + u_t \\ u_t &= \rho_1 u_{t-1} + \rho_2 u_{t-2} + \epsilon_t \end{aligned} \tag{25.9}$$

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 $t = S, S+1, \ldots$, 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:

$$\hat{y}_S = \hat{\phi}_1 \epsilon_{S-1} + \ldots + \hat{\phi}_q \epsilon_{S-q}, \tag{25.10}$$

you will need values for the pre-forecast sample innovations, $\epsilon_{S-1}, \epsilon_{S-2}, \ldots, \epsilon_{S-q}$. 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: $\epsilon_0, \epsilon_{-1}, \epsilon_{-2}, \ldots, \epsilon_{-q}$. (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** dropdown menu 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 ("Initializing MA Innovations" on page 164), the q values for the innovations *beyond* the estimation sample are set to zero:

$$\tilde{\epsilon}_{T+1} = \tilde{\epsilon}_{T+2} = \ldots = \tilde{\epsilon}_{T+q} = 0 \tag{25.11}$$

EViews then uses the unconditional residuals to perform the backward recursion:

$$\tilde{\epsilon}_t = \hat{u}_t - \hat{\theta}_1 \tilde{\epsilon}_{t+1} - \ldots - \hat{\theta}_q \tilde{\epsilon}_{t+q} \tag{25.12}$$

for $t = T, \ldots, 0, \ldots, -(q-1)$ 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 $\epsilon_{S-1}, \epsilon_{S-2}, \ldots, \epsilon_{S-q}$ using the recursion:

$$\hat{\epsilon}_t = \hat{u}_t - \hat{\theta}_1 \hat{\epsilon}_{t-1} - \ldots - \hat{\theta}_q \hat{\epsilon}_{t-q} \tag{25.13}$$

for $t = 1, \ldots, S-1$, 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:

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:

```
smpl 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 LOG(X), or use an auto-updating regressor series EXPZ that is defined using the expression 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 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 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(hs)+sp) 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, LOG(HS) + 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 LOG(HS) + 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 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 **Forecast 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) c sp
log(hs) c 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, LOG(HS)–LOG(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 LOG(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 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 LOG(HS) and LOG(HS(-1)):

```
log(hs) c log(hs(-1))
```

The **Forecast** dialog for the first equation specification (using LOGHS and LOGHSLAG) contains an additional dropdown menu 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 dropdown menu 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 dropdown menu 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.

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 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.

Note also that for non-linear dynamic forecasting, EViews produces what Tong and Lim (1980) term the "eventual forecasting function" in which the lagged forecasted values are substituted recursively into the one-step ahead function. If you wish to obtain simulation-based multi-step forecasting, you may create a model from your equation using **Proc/Make Model**, and then use the resulting model to perform the dynamic stochastic simulation.

References

Pindyck, Robert S. and Daniel L. Rubinfeld (1998). *Econometric Models and Economic Forecasts*, 4th edition, New York: McGraw-Hill.

Tong, H. and K. S. Lim (1980). "Threshold Autoregression, Limit Cycles and Cyclical Data," *Journal of the Royal Statistical Society. Series B (Methodological)*, 42, 245–292.

Chapter 26. 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. For some equations estimated using particular methods, only a subset of these categories will be available.

Additional tests are discussed elsewhere in the *User's Guide*. These tests include unit root tests ("Performing Unit Root Tests in EViews" on page 668), the Granger causality test ("Granger Causality" on page 661 of *User's Guide I*), tests specific to binary, order, censored, and count models (Chapter 31. "Discrete and Limited Dependent

Variable Models," on page 361), and the tests for cointegration ("Testing for Cointegration" on page 308).

Coefficient Diagnostics

These diagnostics provide information and evaluate restrictions on the estimated coefficients, including the special case of tests for omitted and redundant variables.

Scaled Coefficients

The **Scaled Coefficients** view displays the coefficient estimates, the standardized coefficient estimates and the elasticity at means. The standardized coefficients are the point estimates of the coefficients standardized by multiplying by the standard deviation of the dependent variable divided by the standard deviation of the regressor.

The elasticity at means are the point estimates of the coefficients scaled by the mean of the dependent variable divided by the mean of the regressor.

Confidence Intervals and Confidence Ellipses

The **Confidence Intervals** view displays a table of confidence intervals for each of the coefficients in the equation.

The **Confidence Intervals** dialog allows you to enter the size of the confidence levels. These can be entered a space delimited list of decimals, or as the name of a scalar or vector in the workfile containing confidence levels. You can also choose how you would like to display the confidence intervals. By default they will be shown in pairs where the low and high values for each confidence level are shown next to each other. By unchecking the **Arrange in pairs** checkbox you can choose to display the confidence intervals concentrically.

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.

We motivate our discussion of this view by pointing out that the Wald test view (**View/Coefficient Diagnostics/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 Diagnostics/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 using equation EQ01 from the workfile "Cellipse.WF1", 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 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:

$$(b - f(\hat{\beta}))'(V(\hat{\beta})^{-1})(b - f(\hat{\beta})) = c_\alpha \tag{26.1}$$

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).

Variance Inflation Factors

Variance Inflation Factors (VIFs) are a method of measuring the level of collinearity between the regressors in an equation. VIFs show how much of the variance of a coefficient estimate of a regressor has been inflated due to collinearity with the other regressors. They can be calculated by simply dividing the variance of a coefficient estimate by the variance of that coefficient had other regressors not been included in the equation.

There are two forms of the Variance Inflation Factor: centered and uncentered. The centered VIF is the ratio of the variance of the coefficient estimate from the original equation divided by the variance from a coefficient estimate from an equation with only that regressor and a constant. The uncentered VIF is the ratio of the variance of the coefficient estimate from the original equation divided by the variance from a coefficient estimate from an equation with only one regressor (and no constant). Note that if you original equation did not have a constant only the uncentered VIF will be displayed.

The VIF view for EQ01 from the "Cellipse.WF1" workfile contains:

Variance Inflation Factors
Date: 08/10/09 Time: 14:35
Sample: 1968 1982
Included observations: 15

Variable	Coefficient Variance	Uncentered VIF
X1	0.002909	1010.429
X2	3.72E-06	106.8991
X3	0.002894	1690.308
X4	1.43E-06	31.15205
X5	1.74E-06	28.87596

The centered VIF is numerically identical to $1/(1-R^2)$ where R^2 is the R-squared from the regression of that regressor on all of the other regressors in the equation.

Note that since the VIFs are calculated from the coefficient variance-covariance matrix, any robust standard error options will be present in the VIFs.

Coefficient Variance Decomposition

The Coefficient Variance Decomposition view of an equation provides information on the eigenvector decomposition of the coefficient covariance matrix. This decomposition is a useful tool to help diagnose potential collinearity problems amongst the regressors. The decomposition calculations follow those given in Belsley, Kuh and Welsch (BKW) 2004 (Section 3.2). Note that although BKW use the singular-value decomposition as their method to decompose the variance-covariance matrix, since this matrix is a square positive semi-definite matrix, using the eigenvalue decomposition will yield the same results.

In the case of a simple linear least squares regression, the coefficient variance-covariance matrix can be decomposed as follows:

$$\mathrm{var}(\beta) = \sigma^2 (X'X)^{-1} = \sigma^2 V S^{-1} V' \tag{26.2}$$

where S is a diagonal matrix containing the eigenvalues of $X'X$, and V is a matrix whose columns are equal to the corresponding eigenvectors.

The variance of an individual coefficient estimate is then:

$$\mathrm{var}(\beta_i) = \sigma^2 \sum_j v_{ij}^2 \tag{26.3}$$

where μ_j is the j-th eigenvalue, and v_{ij} is the (i,j)-th element of V.

We term the j-th condition number of the covariance matrix, κ_j:

$$\kappa_j \equiv \frac{\min(\mu_m)}{\mu_j} \tag{26.4}$$

If we let:

$$\phi_{ij} \equiv \frac{v_{ij}^2}{\mu_j} \tag{26.5}$$

and

$$\phi_i \equiv \sum_j \phi_{ij} \tag{26.6}$$

then we can term the variance-decomposition proportion as:

$$\pi_{ji} \equiv \frac{\phi_{ij}}{\phi_i} \tag{26.7}$$

These proportions, together with the condition numbers, can then be used as a diagnostic tool for determining collinearity between each of the coefficients.

Belsley, Kuh and Welsch recommend the following procedure:

- Check the condition numbers of the matrix. A condition number smaller than 1/900 (0.001) could signify the presence of collinearity. Note that BKW use a rule of any number greater than 30, but base it on the condition numbers of X, rather than $X'X^{-1}$.
- If there are one or more small condition numbers, then the variance-decomposition proportions should be investigated. Two or more variables with values greater than 0.5 associated with a small condition number indicate the possibility of collinearity between those two variables.

To view the coefficient variance decomposition in EViews, select **View/Coefficient Diagnostics/Coefficient Variance Decomposition**. EViews will then display a table showing the Eigenvalues, Condition Numbers, corresponding Variance Decomposition Proportions and, for comparison purposes, the corresponding Eigenvectors.

As an example, we estimate an equation using data from Longley (1967), as republished in Greene (2008). The workfile "Longley.WF1" contains macro economic variables for the US between 1947 and 1962, and is often used as an example of multicollinearity in a data set. The equation we estimate regresses Employment on Year (YEAR), the GNP Deflator (PRICE), GNP, and Armed Forces Size (ARMED). The coefficient variance decomposition for this equation is show below.

Coefficient Variance Decomposition
Date: 07/16/09 Time: 12:42
Sample: 1947 1962
Included observations: 16

Eigenvalues	17208.87	0.208842	0.054609	1.88E-07
Condition	1.09E-11	9.02E-07	3.45E-06	1.000000

Variance Decomposition Proportions

	Associated Eigenvalue			
Variable	1	2	3	4
YEAR	0.988939	0.010454	0.000607	2.60E-13
PRICE	1.000000	9.20E-09	5.75E-10	7.03E-19
GNP	0.978760	0.002518	0.017746	0.000975
ARMED	0.037677	0.441984	0.520339	9.31E-11

Eigenvectors

	Associated Eigenvalue			
Variable	1	2	3	4
YEAR	0.030636	-0.904160	-0.426067	-0.004751
PRICE	-0.999531	-0.027528	-0.013451	-0.000253
GNP	0.000105	0.001526	0.007921	-0.999967
ARMED	0.000434	0.426303	-0.904557	-0.006514

The top line of the table shows the eigenvalues, sorted from largest to smallest, with the condition numbers below. Note that the final condition number is always equal to 1. Three of the four eigenvalues have condition numbers smaller than 0.001, with the smallest condition number being very small: 1.09E-11, which would indicate a large amount of collinearity.

The second section of the table displays the decomposition proportions. The proportions associated with the smallest condition number are located in the first column. Three of these values are larger than 0.5, indeed they are very close to 1. This indicates that there is a high level of collinearity between those three variables, YEAR, PRICE and GNP.

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:

$$\log Q = A + \alpha \log L + \beta \log K + \epsilon , \tag{26.8}$$

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 in the workfile "Coef_test.WF1" provided the following result:

Dependent Variable: LOG(Q)
Method: Least Squares
Date: 08/10/09 Time: 11:46
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	Hannan-Quinn criter.	-3.278429
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 Diagnostics/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
Null Hypothesis: C(2) + C(3) = 1

Test Statistic	Value	df	Probability
t-statistic	10.95526	22	0.0000
F-statistic	120.0177	(1, 22)	0.0000
Chi-square	120.0177	1	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. In cases with a single restriction, EViews reports the t-statistic equivalent of the F-statistic. See "Wald Test Details" on page 207 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 F-statistic and Chi-square statistic 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, c(3)=1/3
```

and EViews reports:

Wald Test:
Equation: EQ1
Null Hypothesis: C(2)=2/3, C(3)=1/3

Test Statistic	Value	df	Probability
F-statistic	106.6113	(2, 22)	0.0000
Chi-square	213.2226	2	0.0000

Null Hypothesis Summary:

Normalized Restriction (= 0)	Value	Std. Err.
-2/3 + C(2)	0.924508	0.167740
-1/3 + C(3)	-0.093729	0.105390

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 general production function of the form:

$$\log Q = \beta_1 + \beta_2 \log(\beta_3 K^{\beta_4} + (1 - \beta_3) L^{\beta_4}) + \epsilon \tag{26.9}$$

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 may initialize the parameters using the command

```
param c(1) -2.6 c(2) 1.8 c(3) 1e-4 c(4) -6
```

then select **Quick/Estimate Equation…** and then estimate the following specification:

```
log(q) = c(1) + c(2)*log(c(3)*k^c(4)+(1-c(3))*l^c(4))
```

to obtain

Dependent Variable: LOG(Q)
Method: Least Squares
Date: 08/10/09 Time: 13:39
Sample: 1947 1971
Included observations: 25
Convergence achieved after 288 iterations
LOG(Q)=C(1)+C(2)*LOG(C(3)*K^C(4)+(1-C(3))*L^C(4))

	Coefficient	Std. Error	t-Statistic	Prob.
C(1)	-2.655953	0.337610	-7.866935	0.0000
C(2)	-0.301579	0.245596	-1.227944	0.2331
C(3)	4.37E-05	0.000318	0.137553	0.8919
C(4)	-6.121195	5.100604	-1.200092	0.2435

R-squared	0.985325	Mean dependent var	4.767586
Adjusted R-squared	0.983229	S.D. dependent var	0.326086
S.E. of regression	0.042229	Akaike info criterion	-3.345760
Sum squared resid	0.037450	Schwarz criterion	-3.150740
Log likelihood	45.82200	Hannan-Quinn criter.	-3.291670
F-statistic	470.0092	Durbin-Watson stat	0.725156
Prob(F-statistic)	0.000000		

To test the nonlinear restriction $\beta_2 = 1/\beta_4$, choose **View/Coefficient Diagnostics/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: Untitled
Null Hypothesis: C(2) = 1/C(4)

Test Statistic	Value	df	Probability
t-statistic	-1.259105	21	0.2218
F-statistic	1.585344	(1, 21)	0.2218
Chi-square	1.585344	1	0.2080

Null Hypothesis Summary:

Normalized Restriction (= 0)	Value	Std. Err.
C(2) - 1/C(4)	-0.138212	0.109770

Delta method computed using analytic derivatives.

We focus on the p-values for the statistics which show that we fail 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: Untitled
Null Hypothesis: C(2)*C(4)=1

Test Statistic	Value	df	Probability
t-statistic	11.11048	21	0.0000
F-statistic	123.4427	(1, 21)	0.0000
Chi-square	123.4427	1	0.0000

Null Hypothesis Summary:

Normalized Restriction (= 0)	Value	Std. Err.
-1 + C(2)*C(4)	0.846022	0.076146

Delta method computed using analytic derivatives.

so that the test now decisively rejects the null hypothesis. We hasten to add that this type of inconsistency in results 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:

$$y = f(\beta) + \epsilon \tag{26.10}$$

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:

$$H_0\colon\ g(\beta) = 0\,, \tag{26.11}$$

where g is a smooth function, $g\colon R^k \rightarrow R^q$, imposing q restrictions on β. The Wald statistic is then computed as:

$$W = g(\beta)'\left(\frac{\partial g(\beta)}{\partial \beta}\hat{V}(b)\frac{\partial g(\beta)}{\partial \beta'}\right)g(\beta)\big|_{\beta = b} \tag{26.12}$$

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:

$$\hat{V}(b) = s^2\left(\sum_i \frac{\partial f_i(\beta)}{\partial \beta}\frac{\partial f_i(\beta)}{\partial \beta'}\right)^{-1}\Bigg|_{\beta = b} \tag{26.13}$$

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,

$$y = X\beta + \epsilon \tag{26.14}$$

and linear restrictions:

$$H_0\colon\ R\beta - r = 0, \tag{26.15}$$

where R is a known $q \times k$ matrix, and r is a q-vector, respectively. The Wald statistic in Equation (26.12) reduces to:

$$W = (Rb - r)'(Rs^2(X'X)^{-1}R')^{-1}(Rb - r), \tag{26.16}$$

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:

$$F = \frac{W}{q} = \frac{(\tilde{u}'\tilde{u} - u'u)/q}{(u'u)/(T-k)}, \tag{26.17}$$

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 (26.17) 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, F-statistic (and corresponding t-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:

$$LR = -2(l_r - l_u) \tag{26.18}$$

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, 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.
- Equations estimated by Two-Stage Least Squares and GMM offer a variant of this test based on the difference in J-statistics.

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 Diagnostics/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 specification is:

```
log(q) c log(l) log(k)
```

If you enter the list:

```
log(l)^2 log(k)^2
```

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 (the bottom portion shows the estimated test equation):

Omitted Variables Test
Equation: EQ1
Specification: LOG(Q) C LOG(L) LOG(K)
Omitted Variables: LOG(L)^2 LOG(K)^2

	Value	df	Probability
F-statistic	2.490982	(2, 20)	0.1082
Likelihood ratio	5.560546	2	0.0620

F-test summary:

	Sum of Sq.	df	Mean Squares
Test SSR	0.008310	2	0.004155
Restricted SSR	0.041669	22	0.001894
Unrestricted SSR	0.033359	20	0.001668
Unrestricted SSR	0.033359	20	0.001668

LR test summary:

	Value	df
Restricted LogL	44.48746	22
Unrestricted LogL	47.26774	20

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, neither test rejects the null hypothesis that the two series do not belong to the equation at a 5% 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 Diagnostics/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 specification is:

```
log(q) c log(l) log(k) log(l)^2 log(k)^2
```

If you type the list:

```
log(l)^2 log(k)^2
```

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 top portion of the output is:

Redundant Variables Test
Equation: EQ1
Specification: LOG(Q) C LOG(L) LOG(K) LOG(L)^2 LOG(K)^2
Redundant Variables: LOG(L)^2 LOG(K)^2

	Value	df	Probability
F-statistic	2.490982	(2, 20)	0.1082
Likelihood ratio	5.560546	2	0.0620

F-test summary:

	Sum of Sq.	df	Mean Squares
Test SSR	0.008310	2	0.004155
Restricted SSR	0.041669	22	0.001894
Unrestricted SSR	0.033359	20	0.001668
Unrestricted SSR	0.033359	20	0.001668

LR test summary:

	Value	df
Restricted LogL	44.48746	22
Unrestricted LogL	47.26774	20

The reported 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:

$$F = \frac{(\tilde{u}'\tilde{u} - (u_1'u_1 + u_2'u_2))/k}{(u_1'u_1 + u_2'u_2)/(T - 2k)} \tag{26.19}$$

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 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.

For example, suppose we have estimated an equation specification of

```
lwage c grade age high
```

using data from the “Cps88.WF1” workfile.

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. To apply the breakpoint test, push **View/Coefficient Diagnostics/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. Since UNION contains values representing either union or non-union and MARRIED contains values for married and single, entering "union married" will specify 4 subsamples: non-union/married, non-union/single, union/married, and union/single. In the bottom portion of the dialog we indicate the names of the regressors that should be allowed to vary across breakpoints. By default, all of the variables will be allowed to vary.

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 = non-union, MARRIED = single UNION = non-union, MARRIED = married UNION = union, MARRIED = single UNION = union, MARRIED = married

Note all three statistics decisively reject the null hypothesis.

Residual Diagnostics

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 464 in *User's Guide I*.

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 Diagnostics/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 218, 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 "Q-Statistics" on page 464 of *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 Diagnostics/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 "Histogram and Stats" on page 440 of *User's Guide I*, for a discussion of the Jarque-Bera test.

To display the histogram and Jarque-Bera statistic, select **View/Residual Diagnostics/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 there are lagged dependent variables or not. 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;

$$y_t = X_t\beta + \epsilon_t \tag{26.20}$$

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 $e = y - X\hat{\beta}$:

$$e_t = X_t\gamma + \left(\sum_{s=1}^{p} \alpha_s e_{t-s}\right) + v_t. \tag{26.21}$$

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 Diagnostics/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.

To illustrate, consider the macroeconomic data in our "Basics.WF1" workfile. We begin by regressing money supply M1 on a constant, contemporaneous industrial production IP and three lags of IP using the equation specification

```
m1 c ip(0 to -3)
```

The serial correlation LM test results for this equation with 2 lags in the test equation strongly reject the null of no serial correlation:

Breusch-Godfrey Serial Correlation LM Test:

F-statistic	25280.60	Prob. F(2,353)	0.0000
Obs*R-squared	357.5040	Prob. Chi-Square(2)	0.0000

Test Equation:
Dependent Variable: RESID
Method: Least Squares
Date: 08/10/09 Time: 14:58
Sample: 1960M01 1989M12
Included observations: 360
Presample missing value lagged residuals set to zero.

Variable	Coefficient	Std. Error	t-Statistic	Prob.
C	-0.584837	1.294016	-0.451955	0.6516
IP	-11.36147	0.599613	-18.94800	0.0000
IP(-1)	17.13281	1.110223	15.43187	0.0000
IP(-2)	-5.029158	1.241122	-4.052107	0.0001
IP(-3)	-0.717490	0.629348	-1.140054	0.2550
RESID(-1)	1.158582	0.051233	22.61410	0.0000
RESID(-2)	-0.156513	0.051610	-3.032587	0.0026

R-squared	0.993067	Mean dependent var	-6.00E-15
Adjusted R-squared	0.992949	S.D. dependent var	76.48159
S.E. of regression	6.422212	Akaike info criterion	6.576655
Sum squared resid	14559.42	Schwarz criterion	6.652218
Log likelihood	-1176.798	Hannan-Quinn criter.	6.606700
F-statistic	8426.868	Durbin-Watson stat	1.582614
Prob(F-statistic)	0.000000		

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" on page 33) 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 squared residuals from the original equation 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

```
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 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 27. “ARCH and GARCH Estimation,” on page 263 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:

$$e_t^2 = \beta_0 + \left(\sum_{s=1}^{q} \beta_s e_{t-s}^2 \right) + v_t, \tag{26.22}$$

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 distri-

bution 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:

$$y_t = b_1 + b_2 x_t + b_3 z_t + e_t \tag{26.23}$$

where the b are the estimated parameters and e the residual. The test statistic is then based on the auxiliary regression:

$$e_t^2 = \alpha_0 + \alpha_1 x_t + \alpha_2 z_t + \alpha_3 x_t^2 + \alpha_4 z_t^2 + \alpha_5 x_t z_t + v_t . \tag{26.24}$$

Prior to EViews 6, White tests always included the level values of the regressors (*i.e.* the cross product of the regressors and a constant) whether the original regression included a constant term. 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 nonsignificant 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 Diagnostics/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 nonlinear 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** dropdown menu 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 and later 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.

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 Harvey (Breusch-Pagan-Godfrey/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:

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:

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. 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: 08/10/09 Time: 15:06
Sample (adjusted): 1959M07 1989M12
Included observations: 335 after adjustments

Variable	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.

Stability Diagnostics

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'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; equations estimated using GMM offer a related test (see "GMM Breakpoint Test" on page 116).

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:

$$F = \frac{(\tilde{u}'\tilde{u} - (u_1'u_1 + u_2'u_2))/k}{(u_1'u_1 + u_2'u_2)/(T-2k)}, \tag{26.25}$$

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 parame-

ters 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 Diagnostics/Chow Breakpoint Test…** on the equation toolbar. In the dialog that appears, list the dates or observation numbers for the breakpoints in the upper edit field, and the regressors that are allowed to vary across breakpoints in the lower edit field.

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.

The results of a test applied to EQ1 in the workfile “Coef_test.WF1”, using the settings above are:

Chow Breakpoint Test: 1960M01 1970M01
Null Hypothesis: No breaks at specified breakpoints
Varying regressors: All equation variables
Equation Sample: 1959M01 1989M12

F-statistic	186.8638	Prob. F(6,363)	0.0000
Log likelihood ratio	523.8566	Prob. Chi-Square(6)	0.0000
Wald Statistic	1121.183	Prob. Chi-Square(6)	0.0000

Indicating that the coefficients are not stable across regimes.

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 226.

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:

$$\text{MaxF} = \max_{\tau_1 \le \tau \le \tau_2} (F(\tau)) \tag{26.26}$$

The Exp statistic takes the form:

$$\text{ExpF} = \ln\left(\frac{1}{k}\sum_{\tau=\tau_1}^{\tau_2} \exp\left(\frac{1}{2}F(\tau)\right)\right) \tag{26.27}$$

The Ave statistic is the simple average of the individual F-statistics:

$$\text{AveF} = \frac{1}{k} \sum_{\tau = \tau_1}^{\tau_2} F(\tau) \tag{26.28}$$

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 15% 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 Diagnostics/ 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, EQ02 in the workfile "DEMO.WF1", using quarterly data from 1952Q1 to 1992Q4. 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:

Note all three of the summary statistic measures fail to reject the null hypothesis of no structural breaks at the 1% level within the 113 possible dates tested. The maximum statistic was in 1982Q2, and that is the most likely breakpoint location. Also, since the original equation was linear, note that the p-value for the LR F-statistic is identical to the Wald F-statistic.

Multiple Breakpoint Tests

Tests for parameter instability and structural change in regression models have been an important part of applied econometric work dating back to Chow (1960), who tested for regime change at *a priori* known dates using an F-statistic. To relax the requirement that the candidate breakdate be known, Quandt (1960) modified the Chow framework to consider the F-statistic with the largest value over all possible breakdates. Andrews (1993) and Andrews and Ploberger (1994) derived the limiting distribution of the Quandt and related test statistics. The EViews tools for performing these tests are described in "Chow's Breakpoint Test" on page 226 and "Quandt-Andrews Breakpoint Test" on page 228.

More recently, Bai (1997) and Bai and Perron (1998, 2003a) provide theoretical and computational results that further extend the Quandt-Andrews framework by allowing for *multiple* unknown breakpoints. The remainder of this section offers a brief outline of the Bai and Bai-Perron approach to structural break testing as implemented in EViews. Perron (2006) offers a useful survey of the literature and provides references for those requiring additional discussion.

Background

We consider a standard multiple linear regression model with T periods and m potential breaks (producing $m+1$ regimes). For the observations $T_j, T_j+1, \ldots, T_{j+1}-1$ in regime j we have the regression model

$$y_t = X_t'\beta + Z_t'\delta_j + \epsilon_t \tag{26.29}$$

for the regimes $j = 0, \ldots, m$. Note that the regressors are divided into two groups. The X variables are those whose parameters do not vary across regimes, while the Z variables have coefficients that are regime specific.

While it is slightly more convenient to define breakdates to be the last date of a regime, we follow EViews's convention in defining the breakdate to be the first date of the subsequent regime. We tie down the endpoints by setting $T_0 = 1$ and $T_{m+1} = T+1$.

The multiple breakpoint tests that we consider may broadly be divided into three categories: tests that employ global maximizers for the breakpoints, test that employ sequentially determined breakpoints, and hybrid tests, which combine the two approaches.

Global Maximizer Tests

Bai and Perron (1998) describe global optimization procedures for identifying the m multiple breaks which minimize the sums-of-squared residuals of the regression model Equation (26.29).

Briefly, for a *specific* set of m breakpoints, say $\{T\}_m = (T_1, \ldots, T_m)$, we may minimize the sum-of-squared residuals:

$$S(\beta, \delta | \{T\}) = \sum_{j=0}^{m} \left\{ \sum_{t=T_j}^{T_{j+1}-1} y_t - X_t'\beta - Z_t'\delta_j \right\}^2 \tag{26.30}$$

using standard least squares regression to obtain estimates $(\hat{\beta}, \hat{\delta})$. The global m-break optimizers are the set of breakpoints and corresponding coefficient estimates that minimize sum-of-squares across *all* possible sets of m-break partitions.

Note that the number of comparison models increases rapidly in both m and T so that efficient algorithms for computing the optimizers are required. Practical algorithms for computing the global optimizers for multiple breakpoint models are outlined in Bai and Perron (2003a).

These global breakpoint estimates are then used as the basis for several breakpoint tests. EViews supports both the Bai and Perron (1998) tests of l-breaks versus none test (along with the double maximum variants of this test in which l is determined as part of the testing procedure), and information criterion methods (Yao, 1988 and Liu, Wi, and Zidek, 1997) for determining the number of breaks.

Global L Breaks vs. None

Bai and Perron (1998) describe a generalization of the Quandt-Andrews test (Andrews, 1993) in which we test for equality of the δ_j across multiple regimes. For a test of the null of no breaks against an alternative of l breaks, we employ an F-statistic to evaluate the null hypothesis that $\delta_0 = \delta_1 = \ldots = \delta_{l+1}$. The general form of the statistic (Bai-Perron 2003a) is:

$$F(\hat{\delta}) = \frac{1}{T}\left(\frac{T-(l+1)q-p}{kq}\right)(R\hat{\delta})'(R\hat{V}(\hat{\delta})R')^{-1}R\hat{\delta} \tag{26.31}$$

where $\hat{\delta}$ is the optimal l-break estimate of δ, $(R\delta)' = (\delta_0' - \delta_1', \ldots, \delta_l' - \delta_{l+1}')$, and $\hat{V}(\hat{\delta})$ is an estimate of the variance covariance matrix of $\hat{\delta}$ which may be robust to serial correlation and heteroskedasticity, whose form depends on assumptions about the distribution of the data and the errors across segments. (We do not reproduce the formulae for the estimators of the variance matrices here as there are a large number of cases to consider; Bai-Perron (2003a) offer detailed descriptions of the various cases.)

A single test of no breaks against an alternative of l breaks assumes that the alternative number of breakpoints l is pre-specified. In cases where l is not known, we may test the null of no structural change against an unknown number of breaks up to some upper-bound, m^*. This type of testing is termed *double maximum* since it involves maximization both for a given l and across various values of the test statistic for l.

The equal-weighted version of the test, termed $UD\max$ chooses the alternative that maximizes the statistic across the number of breakpoints. An alternative approach, denoted

$WD\max$ applies weights to the individual statistics so that the implied marginal p-values are equal prior to taking the maximum.

The distributions of these test statistics are non-standard, but Bai and Perron (2003b) provide critical value and response surface computations for various trimming parameters (minimum sample sizes for estimating a break), numbers of regressors, and numbers of breaks.

Information Criteria

Yao (1988) shows that under relatively strong conditions, the number of breaks $\hat{l}$ that minimizes the Schwarz criterion is a consistent estimator of the true number of breaks in a breaking mean model.

More generally, Liu, Wu, and Zidek (1997) propose use of modified Schwarz criterion for determining the number of breaks in a regression framework. LWZ offer theoretical results showing consistency of the estimated number of breakpoints, and provide simulation results to guide the choice of the modified penalty criterion.

Sequential Testing Procedures

Bai (1997) describes an intuitive approach for detecting more than one break. The procedure involves *sequential* application of breakpoint tests.

- Begin with the full sample and perform a test of parameter constancy with unknown break.
- If the test rejects the null hypothesis of constancy, determine the breakdate, divide the sample into two samples and perform single unknown breakpoint tests in each subsample. Each of these tests may be viewed as a test of the alternative of $l+1 = 2$ versus the null hypothesis of $l = 1$ breaks. Add a breakpoint whenever a subsample null is rejected. (Alternately, one could test only the single subsample which shows the greatest improvement in the sum-of-squared residuals.)
- Repeat the procedure until all of the subsamples do not reject the null hypothesis, or until the maximum number of breakpoints allowed or maximum subsample intervals to test is reached.

If the number of breakpoints is pre-specified, we simply estimate the specified number of breakpoints using the one-at-a-time method.

Once the sequential breakpoints have been determined, Bai recommends a refinement procedure whereby breakpoints are re-estimated if they are obtained from a subsample containing more than one break. This procedure is required so that the breakpoint estimates have the same limiting distribution as those obtained from the global optimization procedure.

Note that EViews uses the (potentially robust) F-statistic in Equation (26.31) for the test in place of the difference in sums-of-squared residuals described in Bai (1997) and Bai and Per-

ron (1998). Critical value and response surface computations are again provided by Bai and Perron (2003b).

Global Plus Sequential Testing

Bai and Perron (1998) describe a modified Bai (1997) approach in which, at each test step, the l breakpoints under the null are obtained by global minimization of the sum-of-squared residuals. We may therefore view this approach as an $l+1$ versus l test procedure that combines the global and sequential testing approaches.

Each test begins with the set of l global optimizing breakpoints and performs a single test of parameter constancy using the subsample break that most reduces the-sum-of-squared residuals. Note that in this case, we only test for constancy in a single subsample.

Computing Multiple Breakpoint Tests in EViews

To use the EViews tools for testing for multiple breaks, you must use an equation that is specified by list and estimated by least squares. Note in particular that this latter restriction means that models with AR and MA terms are not eligible for multiple break testing.

From an estimated equation, bring up the multiple break testing dialog, by clicking on **View/Stability Diagnostics/Multiple Breakpoint Test...**.

The dialog is divided into the **Test specification**, **Breakpoint variables**, and **Options** sections.

Test Specification

The **Test specification** section contains a **Method** drop-down where you may specify the type of test you wish to perform. You may choose between:

- Sequential L + 1 breaks vs. L
- Sequential tests all subsets

- Global L breaks vs. none
- L + 1 breaks vs. global L
- Global information criteria

The two sequential tests are based on the Bai sequential methodology as described in "Sequential Testing Procedures" on page 232 above. The methods differ in whether, for a given l breakpoints, we test for an additional breakpoint in each of the $l+1$ segments (**Sequential tests all subsets**), or whether we test for the single added breakpoint that most reduces the sum-of-squares (**Sequential L + 1 breaks vs. L**).

The **Global L breaks vs. none** choice implements the Bai-Perron tests of l globally optimized breaks against the null of no structural breaks, along with the corresponding $UD\max$ and $WD\max$ tests ("Global L Breaks vs. None" on page 231).

The **L + 1 breaks vs. global L** choice implements the Bai-Perron l vs. $l+1$ testing procedure outlined in "Global Plus Sequential Testing" on page 233.

The **Global information criteria** uses the information criteria computed from the global optimizers to determine the number of breaks ("Information Criteria" on page 232).

Breakpoint Variables

EViews supports the testing of partial structural change models in which only a subset of the variables in the regression are subject to change across regimes. The variables which have regime specific coefficients should be listed in the **Regressors to vary across breakpoints** edit field.

By default, all of the variables in your specification will be included in this list. To treat some of these variables as non-varying X's, you may simply delete them from the list. Note that there must be at least one variable in the list.

Options

The **Options** section of the dialog allow you to specify the maximum number of breaks or break levels to consider, the trimming percentage of the sample, the significance level for any test computations (if relevant), and assumptions regarding the computation of the variance matrices used in testing (if relevant):

- The **Maximum breaks** limits the number of breakpoints allowed via global testing and in sequential or mixed l vs. $l+1$ testing. If you have selected the **Sequential tests all subsets** method, the edit field will be labeled **Maximum levels** to indicate that the restriction is on the maximum number of break levels allowed. This change in labeling reflects the fact that the Bai all subsets approach potentially adds $l+1$ breaks for a given set of l breaks.

- The **Trimming percentage,** $e = 100(h/T)$ implicitly determines h, the minimum segment length permitted when constructing a test. Small values of the trimming percentage can lead to estimates of coefficients and variances which are based on very few observations.

In testing settings, you will be prompted to specify a test size, and to make assumptions about the distribution of the errors across segments which determine the form for the estimator $\hat{V}(\hat{\delta})$ in Equation (26.31) on page 231.

- The **Significance level** drop-down menu should be used to choose between test size values of (0.01, 0.025, 0.05, and 0.10). This menu is not relevant for tests which select between models using information criteria.
- The **Allow error distributions to differ across breaks** lets you specify different error distributions for different regimes (which in turn implies using different estimators for $\hat{V}(\hat{\delta})$; see Bai and Perron, 2003b for details). Selecting this option will provide robustness of the test to error distribution variation at the cost of power if the error distributions are the same across regimes.

We remind you that EViews will employ the coefficient covariance settings in the original equation when determining whether to allow for heteroskedasticity alone or heteroskedasticity and serial correlation. Thus, if you estimated your original equation using White standard errors, EViews will compute the breakpoint tests using an statistic which is robust to heteroskedasticity. Similarly, if you estimated your original equation with Newey-West standard errors, EViews will compute the breakpoint tests using a HAC robust test statistic.

One final note. EViews will, by default, estimate the robust specification assuming heterogeneous distributions for the Z_t. Bai and Perron (2003a) who, with one exception, do not impose the restriction that the distribution of the Z_t is the same across regimes. In cases where you are testing using robust variances, EViews will offer you a choice of whether to assume a common distribution for the data across regimes.

Bai and Perron do impose the homogeneity data restriction when computing heteroskedasticity and HAC robust variances estimators assuming homogeneous errors. To match the Bai-Perron common error assumptions, you will have to select the **Assume common data distribution checkbox**.

(Note that EViews does not allow you to specify heterogeneous error distributions and robust covariances in partial switching models.)

Examples

To illustrate the use of these tools in practice, we consider a simple model of the U.S. *ex-post* real interest rate from Garcia and Perron (1996) that is used as an example by Bai and Perron

(2003a). The data, which consist of observations for the three-month treasury rate deflated by the CPI for the period 1961q1–1983q3, are provided in the series RATES in the workfile "realrate.WF1". The regression model consists of a constant regressor, and allows for serial correlation that differs across regimes through the use of HAC covariance estimation. We allow up to 5 breaks in the model, and employ a trimming percentage of 15% $(e = 15)$. Since there are 103 observations in the sample, the trimming value implies that regimes are restricted to have at least 15 observations.

Following Bai and Perron (2003a), we begin by estimating the equation specification using least squares. Our equation specification consists of the dependent variable and a single (constant) regressor, so we enter

```
rate c
```

in the specification dialog

Since we wish to allow for serial correlation in the errors, we specify a quadratic spectral kernel based HAC covariance estimation using prewhitened residuals. The kernel bandwith is determined automatically using the Andrews AR(1) method.

The covariance options may be specified in the **Equation Estimation** dialog by selecting the **Options** tab, clicking on the **HAC options** button and filling out the dialog as shown:

Click on **OK** to accept the HAC settings, and then on **OK** to estimate the equation. The estimation results should be as depicted below:

Dependent Variable: RATES
Method: Least Squares
Date: 12/03/12 Time: 14:09
Sample: 1961Q1 1986Q3
Included observations: 103
HAC standard errors & covariance (Prewhitening with lags = 1,
Quadratic-Spectral kernel, Andrews bandwidth = 1.9610)

Variable	Coefficient	Std. Error	t-Statistic	Prob.
C	1.375142	0.599818	2.292600	0.0239

R-squared	0.000000	Mean dependent var	1.375142
Adjusted R-squared	0.000000	S.D. dependent var	3.451231
S.E. of regression	3.451231	Akaike info criterion	5.325001
Sum squared resid	1214.922	Schwarz criterion	5.350580
Log likelihood	-273.2375	Hannan-Quinn criter.	5.335361
Durbin-Watson stat	0.745429		

To construct multiple breakpoint tests for this equation, select **View/Stability Diagnostics/ Multiple Breakpoint Test...** from the equation dialog. We consider examples for three different approaches for multiple breakpoint testing with this equation.

Sequential Bai-Perron

The default **Method** setting (**Sequential L + 1 breaks vs. L**) instructs EViews to perform sequential testing of $l+1$ versus l breaks using the methods outlined by Bai (1997) and Bai and Perron (1998).

There is a single regressor "C" which we require to be in the list of breaking variables.

By default, the tests allow for a maximum number of 5 breaks, employ a trimming percentage of 15%, and use the 0.05 significance level for the sequential testing. We will leave these options at their default settings. We do, however, select the **Allow error distributions to differ across breaks** checkbox to allow for error heterogeneity.

Click on **OK** to accept the test specification and display the test results. The top portion of the dialog shows the test settings, including the test method, breakpoint variables, test options, and method of computing test covariances. Note that the test employs the same HAC covariance settings used in the original equation but assume regime specific error distributions:

```
Multiple breakpoint tests
Bai-Perron tests of L+1 vs. L sequentially determined
    breaks
Date: 12/03/12   Time: 14:09
Sample: 1961Q1 1986Q3
Included observations: 103
Breakpoint variables: C
Break test options: Trimming 0.15, Max. breaks 5, Sig. level
    0.05
Test statistics employ HAC covariances (Prewhitening with
    lags = 1, Quadratic-Spectral kernel, Andrews
    bandwidth)
Allow heterogeneous error distributions across breaks
```

The middle section of the table presents the actual sequential test results:

Sequential F-statistic determined breaks: 3

Break Test	F-statistic	Scaled F-statistic	Critical Value**
0 vs. 1 *	57.90582	57.90582	8.58
1 vs. 2 *	33.92749	33.92749	10.13
2 vs. 3 *	14.72464	14.72464	11.14
3 vs. 4	0.033044	0.033044	11.83

* Significant at the 0.05 level.
** Bai-Perron (Econometric Journal, 2003) critical values.

EViews displays the F-statistic, along with the F-statistic scaled by the number of varying regressors (which is the same in this case, since we only have the single, varying regressor), and the Bai-Perron critical value for the scaled statistic. The sequential test results indicate that there are three breakpoints: we reject the nulls of 0, 1, and 2 breakpoints in favor of the alternatives of 1, 2, and 3 breakpoints, but the test of 4 versus 3 breakpoints does not reject the null.

The bottom portion of the output shows the estimated breakdates:

Break dates:

	Sequential	Repartition
1	1980Q4	1967Q1
2	1972Q4	1972Q4
3	1967Q1	1980Q4

EViews displays both the breakdates obtained from the original sequential procedure, and those obtained following the repartition procedure. In this case, the dates do not change. Again bear in mind that the results follow the EViews convention in defining breakdates to be the first date of the subsequent regime.

Global Bai-Perron L Breaks vs. None

To perform the Bai-Perron tests of l globally optimized breaks against the null of no structural breaks, along with the corresponding $UD\max$ and $WD\max$ tests, simply call up the dialog and change the **Method** drop-down to **Global L breaks vs. none**:

We again leave the remaining settings at their default values with the exception of the **Allow error distributions to differ across breaks** checkbox which is selected. Click on **OK** to perform the test.

The top portion of the output, which shows the test settings, is almost identical to the output for the previous example. The only difference is a line identifying the test method as being "Bai-Perron tests of 1 to M globally determined breaks."

The middle portion of the output contains the test results:

Sequential F-statistic determined breaks:	5
Significant F-statistic largest breaks:	5
UDmax determined breaks:	1
WDmax determined breaks:	1

Breaks	F-statistic	Scaled F-statistic	Weighted F-statistic	Critical Value
1 *	57.90582	57.90582	57.90582	8.58
2 *	43.01429	43.01429	51.11671	7.22
3 *	33.32281	33.32281	47.97143	5.96
4 *	24.77054	24.77054	42.59143	4.99
5 *	18.32587	18.32587	40.21381	3.91

UDMax statistic*	57.90582	UDMax critical value**	8.88
WDMax statistic*	57.90582	WDMax critical value**	9.91

* Significant at the 0.05 level.
** Bai-Perron (Econometric Journal, 2003) critical values.

The first four lines summarize the results for different approaches to determining the number of breaks. The "Sequential" result is obtained by performing tests from 1 to the maximum number until we cannot reject the null; the "Significant" result chooses the largest statistically significant breakpoint. In both cases, the multiple breakpoint test indicates that there are 5 breaks. The "UDmax" and "WDmax" results show the number of breakpoints as

determined by application of the unweighted and weighted maximized statistics. The maximized statistics both indicate the presence of a single break.

The remaining lines show the individual test statistics (original, scaled, weighted) along with the critical values for the scaled statistics. In each case, the statistics far exceed the critical value so that we reject the null of no breaks. Note that the values corresponding to the $UD\max$ and $WD\max$ statistics are shaded for easy identification.

The last two lines of output show the test results for double maximum statistics. In both cases, the maximized value clearly exceeds the critical value, so that we reject the null of no breaks in favor of the alternative of a single break.

The bottom of the portion shows the global optimizers for the breakpoints for each number of breaks:

```
Estimated break dates:
1:  1980Q4
2:  1972Q4,  1980Q4
3:  1967Q1,  1972Q4,  1980Q4
4:  1967Q1,  1972Q4,  1977Q1,  1980Q4
5:  1965Q1,  1968Q4,  1972Q4,  1977Q1,  1980Q4
```

Note that the three-break global optimizers are the same as those obtained in the sequential testing example ("Sequential Bai-Perron" on page 237). This equivalence will not hold in general.

Global Information Criteria

Lastly, we consider using information criteria to select the number of breaks.

Here we see the dialog when we select **Global information criteria** in the **Method** drop-down menu. Note that there are no options for computing the coefficient covariances since this method does not require their calculation. Click on **OK** to construct the table of results.

The top and bottom portions of the output are similar to the results seen previously so we focus only on the test summaries themselves:

Schwarz criterion selected breaks:	2
LWZ criterion selected breaks:	2

Breaks	# of Coefs.	Sum of Sq. Resids.	Log-L	Schwarz* Criterion	LWZ* Criterion
0	1	1214.922	-273.2375	2.512703	2.550154
1	3	644.9955	-240.6282	1.969506	2.082148
2	5	455.9502	-222.7649	1.712641	1.900875
3	7	445.1819	-221.5340	1.778735	2.042977
4	9	444.8797	-221.4990	1.868051	2.208735
5	11	449.6395	-222.0471	1.968688	2.386267

* Minimum information criterion values displayed with shading

The two summary rows show that both the Schwarz and the LWZ information criteria select 2 breaks. The remainder of the output shows, for each break, the number of estimated coefficients, the optimized sum-of-squared residuals and likelihood, and the values of the information criteria. The minimized Schwarz and LWZ values are shaded for easy identification.

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

$$F = \frac{(\tilde{u}'\tilde{u} - u'u)/T_2}{u'u/(T_1 - k)}, \tag{26.32}$$

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 Diagnostics/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
Equation: EQ1
Specification: LOG(CS) C LOG(GDP)
Test predictions for observations from 1973Q1 to 1994:4

	Value	df	Probability
F-statistic	0.708348	(88, 102)	0.9511
Likelihood ratio	91.57087	88	0.3761

F-test summary:

	Sum of Sq.	df	Mean Squares
Test SSR	0.061798	88	0.000702
Restricted SSR	0.162920	190	0.000857
Unrestricted SSR	0.101122	102	0.000991
Unrestricted SSR	0.101122	102	0.000991

LR test summary:

	Value	df
Restricted LogL	406.4749	190
Unrestricted LogL	452.2603	102

Unrestricted log likelihood adjusts test equation results to account for observations in forecast sample

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:

$$y = X\beta + \epsilon, \tag{26.33}$$

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:

$$\begin{aligned} H_0&: \ \epsilon \sim N(0, \sigma^2 I) \\ H_1&: \ \epsilon \sim N(\mu, \sigma^2 I) \ \ \mu \neq 0 \end{aligned} \tag{26.34}$$

The test is based on an augmented regression:

$$y = X\beta + Z\gamma + \epsilon. \tag{26.35}$$

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,

$$y = \beta_0 + \beta_1 X + \epsilon, \tag{26.36}$$

is specified instead of the true relation:

$$y = \beta_0 + \beta_1 X + \beta_2 X^2 + \epsilon \tag{26.37}$$

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,

$$y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \epsilon \tag{26.38}$$

instead of the (true) multiplicative relation:

$$y = \beta_0 X_1^{\beta_1} X_2^{\beta_2} + \epsilon \,. \tag{26.39}$$

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:

$$Z = [\hat{y}^2, \hat{y}^3, \hat{y}^4, \ldots] \tag{26.40}$$

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 Diagnostics/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 only applicable to equations estimated using selected methods.

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 $x_t' b_{t-1}$, where x_t' is the row vector of observations on the regressors in period t. The forecast error is $y_t - x_t' b_{t-1}$, and the forecast variance is given by:

$$\sigma^2 (1 + x_t'(X_{t-1}' X_{t-1})^{-1} x_t) . \tag{26.41}$$

The recursive residual w_t is defined in EViews as:

$$w_t = \frac{(y_t - x_{t-1}' b)}{(1 + x_t'(X_{t-1}' X_{t-1})^{-1} x_t)^{1/2}} . \tag{26.42}$$

These residuals can be computed for $t = k+1, \ldots, T$. 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 Diagnostics/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 249.

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:

$$W_t = \sum_{r=k+1}^{t} w_r / s, \tag{26.43}$$

for $t = k+1, \ldots, T$, where w is the recursive residual defined above, and s is the standard deviation of the recursive residuals w_t. If the β vector remains constant from period to period, $\mathrm{E}(W_t) = 0$, 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:

$$[k, \pm -0.948(T-k)^{1/2}] \qquad \text{and} \qquad [T, \pm 3 \times 0.948(T-k)^{1/2}]. \tag{26.44}$$

Movement of W_t outside the critical lines is suggestive of coefficient instability. A sample CUSUM 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:

$$S_t = \left(\sum_{r=k+1}^{t} w_r^2 \right) \Big/ \left(\sum_{r=k+1}^{T} w_r^2 \right). \tag{26.45}$$

The expected value of S_t under the hypothesis of parameter constancy is:

$$E(S_t) = (t-k)/(T-k) \tag{26.46}$$

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.

Leverage Plots

Leverage plots are the multivariate equivalent of a simple residual plot in a univariate regression. Like influence statistics, leverage plots can be used as a method for identifying influential observations or outliers, as well as a method of graphically diagnosing any potential failures of the underlying assumptions of a regression model.

Leverage plots are calculated by, in essence, turning a multivariate regression into a collection of univariate regressions. Following the notation given in Belsley, Kuh and Welsch 2004 (Section 2.1), the leverage plot for the k-th coefficient is computed as follows:

Let X_k be the k-th column of the data matrix (the k-th variable in a linear equation, or the k-th gradient in a non-linear), and $X[k]$ be the remaining columns. Let u_k be the residuals from a regression of the dependent variable, y on $X[k]$, and let v_k be the residuals from a regression of X_k on $X[k]$. The leverage plot for the k-th coefficient is then a scatter plot of u_k on v_k.

It can easily be shown that in an auxiliary regression of u_k on a constant and v_k, the coefficient on v_k will be identical to the k-th coefficient from the original regression. Thus the original regression can be represented as a series of these univariate auxiliary regressions.

In a univariate regression, a plot of the residuals against the explanatory variable is often used to check for outliers (any observation whose residual is far from the regression line), or to check whether the model is possibly mis-specified (for example to check for linearity). Leverage plots can be used in the same way in a multivariate regression, since each coefficient has been modelled in a univariate auxiliary regression.

To display leverage plots in EViews select **View/ Stability Diagnostics/Leverage Plots...**. EViews will then display a dialog which lets you choose some simple options for the leverage plots.

The **Variables to plot** box lets you enter which variables, or coefficients in a non-linear equation, you wish to plot. By default this box will be filled in with the original regressors from your equation. Note that EViews will let you enter variables that were not in the original equation, in which case the plot will simply show the original equation residuals plotted against the residuals from a regression of the new variable against the original regressors.

To add a regression line to each scatter plot, select the **Add fit lines** checkbox. If you do not wish to create plots of the partialed variables, but would rather plot the original regression residuals against the raw regressors, unselect the **Partial out variables** checkbox.

Finally, if you wish to save the partial residuals for each variable into a series in the workfile, you may enter a naming suffix in the **Enter a naming suffix to save the variables as a series** box. EViews will then append the name of each variable to the suffix you entered as the name of the created series.

We illustrate using an example taken from Wooldridge (2000, Example 9.8) for the regression of R&D expenditures (RDINTENS) on sales (SALES), profits (PROFITMARG), and a constant (using the workfile "Rdchem.WF1"). The leverage plots for equation E1 are displayed here:

Influence Statistics

Influence statistics are a method of discovering influential observations, or outliers. They are a measure of the difference that a single observation makes to the regression results, or how

different an observation is from the other observations in an equation's sample. EViews provides a selection of six different influence statistics: RStudent, DRResid, DFFITS, CovRatio, HatMatrix and DFBETAS.

- RStudent is the studentized residual; the residual of the equation at that observation divided by an estimate of its standard deviation:

$$\bar{e}_i = \frac{e_i}{s(i)\sqrt{1-h_i}} \tag{26.47}$$

  where e_i is the original residual for that observation, $s(i)$ is the variance of the residuals that would have resulted had observation i not been included in the estimation, and h_i is the i-th diagonal element of the Hat Matrix, *i.e.* $x_i(X'X)^{-1}x_i$. The RStudent is also numerically identical to the t-statistic that would result from putting a dummy variable in the original equation which is equal to 1 on that particular observation and zero elsewhere. Thus it can be interpreted as a test for the significance of that observation.

- DFFITS is the scaled difference in fitted values for that observation between the original equation and an equation estimated without that observation, where the scaling is done by dividing the difference by an estimate of the standard deviation of the fit:

$$DFFITS_i = \left[\frac{h_i}{1-h_i}\right]^{1/2} \frac{e_i}{s(i)\sqrt{1-h_i}} \tag{26.48}$$

- DRResid is the dropped residual, an estimate of the residual for that observation had the equation been run without that observation's data.

- COVRATIO is the ratio of the determinant of the covariance matrix of the coefficients from the original equation to the determinant of the covariance matrix from an equation without that observation.

- HatMatrix reports the i-th diagonal element of the Hat Matrix: $x_i(X'X)^{-1}x_i$.

- DFBETAS are the scaled difference in the estimated betas between the original equation and an equation estimated without that observation:

$$DFBETAS_{i,j} = \frac{\beta_j - \beta_j(i)}{s(i)\sqrt{\mathrm{var}(\beta_j)}} \tag{26.49}$$

  where β_j is the original equation's coefficient estimate, and $\beta_j(i)$ is the coefficient estimate from an equation without observation i.

To display influence statistics in EViews select **View/Stability Diagnostics/Influence Statistics**. EViews will bring up a dialog where you can choose how you wish to display the statistics. The **Output statistics** box lets you choose which statistics you would like to calculate, and whether to store them as a series in your workfile. Simply check the check box next to the statistics you would like to calculate, and, optionally, enter the name of the series you would like to be created. Note that for the DFBETAS statistics you should enter a naming suffix, rather than the name of the series. EViews will then create the series with the name of the coefficient followed by the naming suffix you provide.

The **Output type** box lets you select whether to display the statistics in graph form, or in table form, or both. If both boxes are checked, EViews will create a spool object containing both tables and graphs.

If you select to display the statistics in tabular form, then a new set of options will be enabled, governing how the table is formed. By default, EViews will only display 100 rows of the statistics in the table (although note that if your equation has less than 100 observations, all of the statistics will be displayed). You can change this number by changing the **Number of obs to include** dropdown menu. EViews will display the statistics sorted from highest to lowest, where the Residuals are used for the sort order. You can change which statistic is used to sort by using the **Select by** dropdown menu. Finally, you can change the sort order to be by observation order rather than by one of the statistics by using the **Display in observation order** check box.

We illustrate using the equation E1 from the “Rdchem.WF1” workfile. A plot of the DFFITS and COVRATIOs clearly shows that observation 10 is an outlier.

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:

$$W = (b_1 - b_2)'(V_1 + V_2)^{-1}(b_1 - b_2), \tag{26.50}$$

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, consider the quarterly workfile of macroeconomic data in the workfile "Coef_test2.WF1" (containing data for 1947q1–1994q4) and suppose wish to test whether there was a structural change in the consumption function in 1973q1. First, estimate the model in the first sample and save the results by the commands:

```
coef(2) b1
smpl 1947q1 1972q4
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 1973q1 1994q4
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 WALD statistic value of 53.1243 has an associated p-value of 2.9e-12 so that we decisively reject the null hypothesis of no structural change.

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 pro-

posed by Davidson and MacKinnon (1989, 1993), which carries out the test by running an auxiliary regression.

The following equation in the "Basics.WF1" workfile was estimated by OLS:

Dependent Variable: LOG(M1)
Method: Least Squares
Date: 08/10/09 Time: 16:08
Sample (adjusted): 1959M02 1995M04
Included observations: 435 after adjustments

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	Hannan-Quinn criter.	-7.895226
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:

```
equation eq_test.ls log(ip) c dlog(ppi) tb3 log(m1(-1)) urate aaa
eq_test.makeresid res_ip
```

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: 08/10/09 Time: 16:11
Sample (adjusted): 1959M02 1995M04
Included observations: 435 after adjustments

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	-7.924511
Sum squared resid	0.008963	Schwarz criterion	-7.868300
Log likelihood	1729.581	Hannan-Quinn criter.	-7.902326
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 rejects the hypothesis of consistent OLS estimates at conventional levels.

Note that an alternative form of a regressor endogeneity test may be computed using the Regressor Endogeneity Test view of an equation estimated by TSLS or GMM (see "Regressor Endogeneity Test" on page 113).

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:

$$\begin{aligned} H_1: \quad & CS_t = \alpha_1 + \alpha_2 GDP_t + \alpha_3 GDP_{t-1} + \epsilon_t \\ H_2: \quad & CS_t = \beta_1 + \beta_2 GDP_t + \beta_3 CS_{t-1} + \epsilon_t \end{aligned} \tag{26.51}$$

for the variables in the workfile "Coef_test2.WF1". 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(f=na) 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: 08/10/09 Time: 16:17
Sample (adjusted): 1947Q2 1994Q4
Included observations: 191 after adjustments

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	-727.9220	Hannan-Quinn criter.	7.691692
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 may also test model H_2 against model H_1. First, estimate model H_1 and retrieve the fitted values:

```
equation eq_cs1a.ls cs gdp gdp(-1)
eq_cs1a.fit(f=na) cs1
```

Then estimate model H_2 including the fitted values from model H_1. The results of this "reverse" test regression are given by:

Dependent Variable: CS
Method: Least Squares
Date: 08/10/09 Time: 16:46
Sample (adjusted): 1947Q2 1995Q1
Included observations: 192 after adjustments

Variable	Coefficient	Std. Error	t-Statistic	Prob.
C	-1413.901	130.6449	-10.82247	0.0000
GDP	5.131858	0.472770	10.85486	0.0000
CS(-1)	0.977604	0.018325	53.34810	0.0000
CS1F	-7.240322	0.673506	-10.75020	0.0000

R-squared	0.999836	Mean dependent var	1962.779
Adjusted R-squared	0.999833	S.D. dependent var	854.9810
S.E. of regression	11.04237	Akaike info criterion	7.661969
Sum squared resid	22923.56	Schwarz criterion	7.729833
Log likelihood	-731.5490	Hannan-Quinn criter.	7.689455
F-statistic	381618.5	Durbin-Watson stat	2.260786
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.

Bai, Jushan (1997). "Estimating Multiple Breaks One at a Time," *Econometric Theory*, 13, 315–352.

Bai, Jushan and Pierre Perron (1998). "Estimating and Testing Linear Models with Multiple Structural Changes," *Econometrica*, 66, 47–78.

Bai, Jushan and Pierre Perron (2003a). "Computation and Analysis of Multiple Structural Change Models," *Journal of Applied Econometrics*, 6, 72–78.

Bai, Jushan and Pierre Perron (2003b). "Critical Values for Multiple Structural Change Tests," *Econometrics Journal*, 18, 1–22.

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 Econometric*s, Oxford: Oxford University Press.

Engle, Robert F. (1982). "Autoregressive Conditional Heteroskedasticity with Estimates of the Variance of U.K. Inflation," *Econometrica*, 50, 987–1008.

Garcia, Rene and Pierre Perron (1996). "An Analysis of the Real Interest Rate Under Regime Shifts," *The Review of Economics and Statistics*, 78, 111–125.

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.

Liu, Jian, Wu, Shiying, and James V. Zidek (1997). "On Segmented Multivariate Regression," *Statistica Sinica*, 7, 497–525.

Longley, J. W. "An Appraisal of Least Squares Programs for the Electronic Computer from the Point of View of the User," *Journal of the American Statistical Association*, 62(319), 819-841.

Perron, Pierre (2006). "Dealing with Structural Breaks," in *Palgrave Handbook of Econometrics, Vol. 1: Econometric Theory*, T. C. Mills and K. Patterson (*eds.*). New York: Palgrave Macmillan.

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). "A Note on the Lagrange Multiplier and F-statistics for Two Stage Least Squares Regression," *Economics Letters*, 34, 151-155.

Wooldridge, Jeffrey M. (2000). *Introductory Econometrics: A Modern Approach*. Cincinnati, OH: South-Western College Publishing.

Yao, Yi-Ching (1988). "Estimating the Number of Change-points via Schwarz' Criterion," *Statistics & Probability Letters*, 6, 181–189.

Part VI. Advanced Single Equation Analysis

The following sections describe EViews tools for the estimation and analysis of advanced single equation models and time series analysis:

- Chapter 27. "ARCH and GARCH Estimation," beginning on page 263, outlines the EViews tools for ARCH and GARCH modeling of the conditional variance, or volatility, of a variable.
- Chapter 28. "Cointegrating Regression," on page 293 describes EViews' tools for estimating and testing single equation cointegrating relationships. Multiple equation tests for cointegration are described in Chapter 44. "Vector Autoregression and Error Correction Models," on page 833.
- Chapter 29. "Autoregressive Distributed Lag (ARDL) Models," beginning on page 321 describes the specification and estimation 0f Autoregressive Distributed Lag (ARDL) models.
- Chapter 30. "Midas Regression," on page 341 documents EViews tools for Mixed Data Sampling (MIDAS) regression, an estimation technique which allows for data sampled at different frequencies to be used in the same regression.
- Chapter 31. "Discrete and Limited Dependent Variable Models," on page 361 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.*), Heckman selection models, and integer valued (count data).
- Chapter 32. "Generalized Linear Models," on page 421 documents describes EViews tools for the class of Generalized Linear Models.
- Chapter 33. "Robust Least Squares," beginning on page 451 describes tools for robust least squares estimation which are designed to be *robust*, or less sensitive, to outliers.
- Chapter 34. "Least Squares with Breakpoints," beginning on page 471 outlines the EViews estimator for equations with one or more structural breaks.
- Chapter 35. "Discrete Threshold Regression," beginning on page 491 describes the analysis of discrete threshold regressions and autoregressions.
- Chapter 36. "Smooth Transition Regression," beginning on page 507 describes the analysis of smooth threshold regressions and autoregressions.
- Chapter 37. "Elastic Net and Lasso," beginning on page 535 describes the estimation of elastic net and LASSO regularization models.
- Chapter 38. "Functional Coefficient Regression," beginning on page 557 describes tools for estimating semi-parametric functional coefficients models in EViews.

- Chapter 39. "Switching Regression," beginning on page 583 describes estimation of regression models with nonlinearities arising from discrete changes in unobserved regimes.
- Chapter 40. "Quantile Regression," beginning on page 619 describes the estimation of quantile regression and least absolute deviations estimation in EViews.
- Chapter 41. "The Log Likelihood (LogL) Object," beginning on page 643 describes techniques for using EViews to estimate the parameters of maximum likelihood models where you may specify the form of the likelihood.
- Chapter 42. "Univariate Time Series Analysis," on page 667 describes tools for univariate time series analysis, including unit root tests in both conventional and panel data settings, variance ratio tests, and the BDS test for independence.

Chapter 27. 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 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 276.

The GARCH(1, 1) Model

We begin with the simplest GARCH(1,1) specification:

$$Y_t = X_t'\theta + \epsilon_t \tag{27.1}$$

$$\sigma_t^2 = \omega + \alpha\epsilon_{t-1}^2 + \beta\sigma_{t-1}^2 \tag{27.2}$$

in which the mean equation given in (27.1) 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 (27.2) 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 (27.2), we can express the conditional variance as a weighted average of all of the lagged squared residuals:

$$\sigma_t^2 = \frac{\omega}{(1-\beta)} + \alpha \sum_{j=1}^{\infty} \beta^{j-1} \epsilon_{t-j}^2 . \tag{27.3}$$

  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:

$$\epsilon_t^2 = \omega + (\alpha + \beta)\epsilon_{t-1}^2 + v_t - \beta v_{t-1} . \tag{27.4}$$

  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:

$$\sigma_t^2 = \omega + \sum_{j=1}^{q} \beta_j \sigma_{t-j}^2 + \sum_{i=1}^{p} \alpha_i \epsilon_{t-i}^2 \tag{27.5}$$

The GARCH-M Model

The X_t in equation Equation (27.2) 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):

$$Y_t = X_t'\theta + \lambda\sigma_t^2 + \epsilon_t. \tag{27.6}$$

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 (27.6).

$$Y_t = X_t'\theta + \lambda\sigma_t + \epsilon_t. \tag{27.7}$$

$$Y_t = X_t'\theta + \lambda\log(\sigma_t^2) + \epsilon_t \tag{27.8}$$

Regressors in the Variance Equation

Equation (27.5) may be extended to allow for the inclusion of exogenous or predetermined regressors, z, in the variance equation:

$$\sigma_t^2 = \omega + \sum_{j=1}^{q} \beta_j \sigma_{t-j}^2 + \sum_{i=1}^{p} \alpha_i \epsilon_{t-i}^2 + Z_t'\pi. \tag{27.9}$$

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:

$$l_t = -\frac{1}{2}\log(2\pi) - \frac{1}{2}\log\sigma_t^2 - \frac{1}{2}(y_t - X_t'\theta)^2/\sigma_t^2, \tag{27.10}$$

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:

$$l_t = -\frac{1}{2}\log\left(\frac{\pi(\nu-2)\Gamma(v/2)^2}{\Gamma((v+1)/2)^2}\right) - \frac{1}{2}\log\sigma_t^2 - \frac{(\nu+1)}{2}\log\left(1 + \frac{(y_t - X_t'\theta)^2}{\sigma_t^2(\nu-2)}\right) \tag{27.11}$$

where the degree of freedom $\nu > 2$ controls the tail behavior. The *t*-distribution approaches the normal as $\nu \to \infty$.

For the GED, we have:

$$l_t = -\frac{1}{2}\log\left(\frac{\Gamma(1/r)^3}{\Gamma(3/r)(r/2)^2}\right) - \frac{1}{2}\log\sigma_t^2 - \left(\frac{\Gamma(3/r)(y_t - X_t'\theta)^2}{\sigma_t^2\Gamma(1/r)}\right)^{r/2} \tag{27.12}$$

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…**, by selecting **Object/New Object…/Equation…**. Select **ARCH** from the method dropdown menu at the bottom of the dialog. Alternately, typing the keyword `arch` in the command line both creates the object and sets the estimation method.

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 dropdown menu in the upper right-hand side of the dialog. You may choose to include **the Std. Dev.**, **Variance**, or the **Log(Var)** in the mean equation.

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 in the **Model** dropdown menu. 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 276.

In the **Order** section, 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 with threshold order 0.

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 this list.

The distinction between the permanent and transitory regressors is discussed in "The Component GARCH (CGARCH) Model" on page 279.

Restrictions

If you choose the GARCH/TARCH model, you may restrict the parameters of the GARCH model in two ways. One option is to set the **Restrictions** dropdown to **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:

$$\omega = \hat{\sigma}^2\left(1 - \sum_{j=1}^{q} \beta_j - \sum_{i=1}^{p} \alpha_i\right) \tag{27.13}$$

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 **Error Distribution** dropdown menu.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 266 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 "Initializing MA Innovations" on page 164.

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,

$$\sigma_0^2 = \epsilon_0^2 = \lambda^T \hat{\sigma}^2 + (1-\lambda) \sum_{j=0}^{T} \lambda^{T-j-1} (\hat{\epsilon}_{T-j}^2), \tag{27.14}$$

where $\hat{\epsilon}$ are the residuals from the mean equation, $\hat{\sigma}^2$ is the unconditional variance estimate:

$$\hat{\sigma}^2 = \sum_{t=1}^{T} \hat{\epsilon}_t^2 / T \tag{27.15}$$

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:

$$\sigma_0^2 = \hat{\sigma}^2 . \tag{27.16}$$

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:

$$\begin{aligned} R_t &= c + \epsilon_t \\ \sigma_t^2 &= \omega + \alpha\epsilon_{t-1}^2 + \beta\sigma_{t-1}^2 \end{aligned} \tag{27.17}$$

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:

$$\begin{aligned} R_t &= \gamma_0 + \gamma_1 DUM_t + \gamma_2\sigma_t + \epsilon_t \\ \sigma_t^2 &= \omega + \alpha_1\epsilon_{t-1}^2 + \alpha_2\epsilon_{t-2}^2 + \alpha_3\epsilon_{t-3}^2 + \alpha_4\epsilon_{t-4}^2 + \gamma_3 DUM_t \end{aligned} \tag{27.18}$$

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 (symmetric)**.
- 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)) in the workfile “Stocks.WF1”, using backcast values for the initial variances and computing Bollerslev-Wooldridge standard errors. The output is presented below:

Dependent Variable: DLOG(SPX)
Method: ML ARCH - Normal distribution (Marquardt / EViews legacy)
Date: 03/09/15 Time: 13:23
Sample: 1/02/1990 12/31/1999
Included observations: 2528
Convergence achieved after 26 iterations
Presample variance: backcast (parameter = 0.7)
GARCH = C(2) + C(3)*RESID(-1)^2 + C(4)*GARCH(-1)

Variable	Coefficient	Std. Error	z-Statistic	Prob.
C	0.000597	0.000149	4.013882	0.0001
Variance Equation				
C	5.83E-07	1.37E-07	4.261215	0.0000
RESID(-1)^2	0.053317	0.005152	10.34861	0.0000
GARCH(-1)	0.939955	0.006125	153.4702	0.0000

R-squared	-0.000014	Mean dependent var	0.000564
Adjusted R-squared	-0.000014	S.D. dependent var	0.008888
S.E. of regression	0.008889	Akaike info criterion	-6.807476
Sum squared resid	0.199649	Schwarz criterion	-6.798243
Log likelihood	8608.650	Hannan-Quinn criter.	-6.804126
Durbin-Watson stat	1.964029		

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. Also noted is the method for computing the presample variance, in this case backcasting with smoothing parameter $\lambda = 0.7$.

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 (27.2) on page 263. 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

- The **Representations** view displays the estimation command as well as the estimation and substituted coefficients equations for the mean and variance specifications.
- The **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 "ARCH Model Procedures" on page 275).
- **GARCH Graph/Conditional Standard Deviation** and **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 Diagnostics** produces standard diagnostics (confidence intervals and ellipses, Wald tests, omitted and redundant variables tests), for the estimated coefficients. See "Coefficient Diagnostics" on page 196 for details. Note that the likelihood ratio tests are not appropriate under a quasi-maximum likelihood interpretation of your results.
- C**oefficient Diagnostics/Nyblom Stability Test** is a test of parameter stability or structural change. The test provides individual statistics on each parameter (both in the mean and variance equations) and a joint test of null hypothesis is that the parameters are constant through time.
- **Residual Diagnostics/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 462 of *User's Guide I* for an explanation of correlograms and Q-statistics.

- **Residual Diagnostics/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 462 of *User's Guide I* for an explanation of correlograms and Q-statistics. See also **Residual Diagnostics/ARCH LM Test**.

- **Residual Diagnostics/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 440 of *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 Diagnostics/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 218 for a discussion of testing. See also **Residual Diagnostics/Correlogram Squared Residuals.**

- **Residual Diagnostics/Sign-Bias Test** (Engle and Ng, 1993) provides a test for misspecification of the conditional variance model. The test performs a regression of the squared residuals against dummy variables based upon the sign previous residuals. If a GARCH model is correctly specified, the sign of previous residuals should have no impact upon the current squared residuals. The null hypothesis is that the parameters of the regression are zero.

ARCH Model Procedures

Various ARCH equation procedures allow you to produce results based on you estimated equation. Some of these procedures, for example the Make Gradient Group and Make Derivative Group behave the same as in other equations. Some of the procedures have ARCH specific elements:

- **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 **Forecast 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 replaced 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 to "2001m01 @last" so the dynamic forecast begins immediately following the estimation period. Unselect the **Forecast Evaluation** checkbox and click on **OK** to display the forecast results.

  It will be useful to display these results in two columns. Right-mouse click then select **Position and align graphs...**, enter "2" for the number of **Columns**, and select **Automatic** spacing. Click on OK to display the rearranged graph:

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.

- **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.**

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

$$\sigma_t^2 = \sum_{j=1}^{q} \beta_j \sigma^2{}_{t-j} + \sum_{i=1}^{p} \alpha_i \epsilon^2{}_{t-i} \tag{27.19}$$

such that

$$\sum_{j=1}^{q} \beta_j + \sum_{i=1}^{p} \alpha_i = 1 \tag{27.20}$$

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 Zakoïan (1994) and Glosten, Jaganathan, and Runkle (1993). The generalized specification for the conditional variance is given by:

$$\sigma_t^2 = \omega + \sum_{j=1}^{q} \beta_j \sigma_{t-j}^2 + \sum_{i=1}^{p} \alpha_i \epsilon_{t-i}^2 + \sum_{k=1}^{r} \gamma_k \epsilon_{t-k}^2 \bar{I}_{t-k} \tag{27.21}$$

where $\bar{I}_t = 1$ 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:

$$\log(\sigma_t^2) = \omega + \sum_{j=1}^{q} \beta_j \log(\sigma_{t-j}^2) + \sum_{i=1}^{p} \alpha_i \left| \frac{\epsilon_{t-i}}{\sigma_{t-i}} \right| + \sum_{k=1}^{r} \gamma_k \frac{\epsilon_{t-k}}{\sigma_{t-k}}. \tag{27.22}$$

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 variance 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 offers 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:

$$\log(\sigma_t^2) = \omega + \sum_{j=1}^{q} \beta_j \log(\sigma_{t-j}^2) + \sum_{i=1}^{p} \alpha_i \left(\left| \frac{\epsilon_{t-i}}{\sigma_{t-i}} \right| - E\left| \frac{\epsilon_{t-i}}{\sigma_{t-i}} \right| \right) + \sum_{k=1}^{r} \gamma_k \frac{\epsilon_{t-k}}{\sigma_{t-k}}$$

which is an alternative parameterization of the specification above. Estimating the latter 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 drop-down menu and enter the orders for the **ARCH**, **GARCH** and the **Asymmetry order**.

Notice that we have specified the mean equation using an explicit expression. Using the explicit expression is for illustration purposes only; we could just as well entered "dlog(ibm) c dlog(spx)" as our specification.

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:

$$\sigma_t^\delta = \omega + \sum_{j=1}^{q} \beta_j \sigma_{t-j}^\delta + \sum_{i=1}^{p} \alpha_i (|\epsilon_{t-i}| - \gamma_i \epsilon_{t-i})^\delta \tag{27.23}$$

where $\delta > 0$, $|\gamma_i| \leq 1$ for $i = 1, \ldots, r$, $\gamma_i = 0$ 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 dropdown menu 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:

$$\sigma_t^2 = \bar{\omega} + \alpha(\epsilon_{t-1}^2 - \bar{\omega}) + \beta(\sigma_{t-1}^2 - \bar{\omega}). \tag{27.24}$$

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:

$$\begin{aligned} \sigma_t^2 - m_t &= \alpha(\epsilon_{t-1}^2 - m_{t-1}) + \beta(\sigma_{t-1}^2 - m_{t-1}) \\ m_t &= \omega + \rho(m_{t-1} - \omega) + \phi(\epsilon_{t-1}^2 - \sigma_{t-1}^2). \end{aligned} \tag{27.25}$$

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:

$$\begin{aligned} \sigma_t^2 &= (1-\alpha-\beta)(1-\rho)\omega + (\alpha+\phi)\epsilon_{t-1}^2 - (\alpha\rho + (\alpha+\beta)\phi)\epsilon_{t-2}^2 \\ &\quad + (\beta-\phi)\sigma_{t-1}^2 - (\beta\rho - (\alpha+\beta)\phi)\sigma_{t-2}^2 \end{aligned} \tag{27.26}$$

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** dropdown menu. 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 TARCH model, introducing asymmetric effects in the transitory equation and estimates models of the form:

$$\begin{aligned} y_t &= x_t'\pi + \epsilon_t \\ m_t &= \omega + \rho(m_{t-1} - \omega) + \phi(\epsilon_{t-1}^2 - \sigma_{t-1}^2) + \theta_1 z_{1t} \\ \sigma_t^2 - m_t &= \alpha(\epsilon_{t-1}^2 - m_{t-1}) + \gamma(\epsilon_{t-1}^2 - m_{t-1})d_{t-1} + \beta(\sigma_{t-1}^2 - m_{t-1}) + \theta_2 z_{2t} \end{aligned} \tag{27.27}$$

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.

Fractionally Integrated GARCH and EGARCH

EViews 12 estimates Fractionally Integrated GARCH (FIGARCH) models, with support for both standard FIGARCH and Fractionally Integrated Exponential GARCH (FIEGARCH) specifications. The following discussion outlines the basic model and the interactive tools for estimating and working with FIGARCH models.

The Fractionally Integrated GARCH (FIGARCH) Model

While traditional GARCH models focus on short term dynamics of conditional variance, the fractionally integrated GARCH model (FIGARCH) model, introduced by Baillie, Bollerslev and Mikkelsen (1996, BBM), is designed to capture long run dependence properties of the variance.

Background

A standard GARCH model's specification of the variance may be written as,

$$\sigma_t^2 = \omega + \alpha(L)\epsilon_{t-1}^2 + \beta(L)\sigma_{t-1}^2 \tag{27.28}$$

where $\alpha(L)$ and $\beta(L)$ represent polynomial lags,

$$\begin{aligned}\alpha(L) &= \sum_{i=1}^{p} \alpha_i L^i \\ \beta(L) &= \sum_{j=1}^{q} \beta_j L^j\end{aligned} \tag{27.29}$$

and L is the lag operator.

The fractionally integrated GARCH (FIGARCH) model modifies this specification by introducing a fractional difference term. The FIGARCH variance may be written as,

$$\sigma_t^2 = \omega + (1 - \beta(L) - \phi(L)\pi(L))\epsilon_{t-1}^2 + \beta(L)\sigma_{t-1}^2 \tag{27.30}$$

where

$$\begin{aligned}\phi(L) &= 1 - \sum_{i=1}^{r} \alpha_i L^i \\ \beta(L) &= \sum_{j=1}^{q} \beta_j L^j\end{aligned} \tag{27.31}$$

and $\pi(L)$ is the infinite lag operator

$$\begin{aligned}\pi(L) &= (1 - L)^d \\ &= 1 + \pi^*(L)\end{aligned} \tag{27.32}$$

which uses the infinite lag expansion

$$\pi^*(L) = \sum_{i=1}^{\infty} \pi_k L^i \tag{27.33}$$

In practice, the infinite lag expansion is truncated to a finite number. For example, BBM suggest truncation at 1000 lags.

Estimating a FIGARCH model

To estimate a FIGARCH model, select **ARCH - Autoregressive Conditional Heteroskedasticity** in the **Method** dropdown at the bottom of the dialog to display the GARCH dialog, then, in the **Variance and distribution specification** section, choose **FIGARCH** in the **Model** dropdown menu:

As in other GARCH models, in the **Mean equation** section of the dialog, you should enter your variables or explicit equation and select your **ARCH-M** specification.

You may select the **ARCH** and **GARCH** orders, list any **Variance regressors**, and select the **Error distribution**.

Lastly, you may specify the FIGARCH **Lag truncation** number of terms in the FIGARCH infinite lag specification.

The Fractionally Integrated Exponential GARCH (FIEGARCH) Model

The FIEGARCH model of Bollerslev and Mikkelsen (1996) adds FIGARCH long-memory innovations to EGARCH processes.

Background

Before proceeding, note that while FIEGARCH models can be specified for arbitrary ARCH and GARCH orders, researchers typically restrict both to a single lag, and EViews currently supports only these FIEGARCH(1,1) models.

The starting point for FIEGARCH is an EGARCH model that differs slightly from the model typically estimated by EViews (Equation 26.22). The log conditional variance for this new form of the EGARCH model variance is

$$\log(\sigma_t^2) = \omega + \frac{\alpha(L)}{\beta(L)} g(z_{t-1}) \tag{27.34}$$

where

$$\begin{aligned}\alpha(L) &= 1 - \sum_{i=1}^{p} \alpha_i L^i \\ \beta(L) &= 1 - \sum_{j=1}^{q} \beta_j L^j\end{aligned} \tag{27.35}$$

and

$$\begin{aligned}g(z_t) &= \gamma_1 z_t + \gamma_2(|z_t| - E(|z_t|)) \\ \epsilon_t &= \sigma_t z_t\end{aligned} \tag{27.36}$$

For $z_t \sim N(0, 1)$, substituting $z_t = \epsilon_t / \sigma_t$ yields the EGARCH(1, 1) variance specification

$$\begin{aligned}\log(\sigma_t^2) &= (1-\beta)\omega + \gamma_1 \frac{\epsilon_{t-1}}{\sigma_{t-1}} + \gamma_2\left(\left|\frac{\epsilon_{t-1}}{\sigma_{t-1}}\right| - \sqrt{2/\pi}\right) \\ &- \alpha\left(\gamma_1 \frac{\epsilon_{t-2}}{\sigma_{t-2}} + \gamma_2\left(\left|\frac{\epsilon_{t-2}}{\sigma_{t-2}}\right| - \sqrt{2/\pi}\right)\right) + \beta\log(\sigma_{t-1}^2)\end{aligned} \tag{27.37}$$

which may be compared with the corresponding EGARCH(1, 1) using the EViews (Equation 26.22) alternative specification

$$\log(\sigma_t^2) = \omega + \gamma_1 \frac{\epsilon_{t-1}}{\sigma_{t-1}} + \alpha\left(\left|\frac{\epsilon_{t-1}}{\sigma_{t-1}}\right|\right) + \beta\log(\sigma_{t-1}^2) \tag{27.38}$$

The FIEGARCH model adds a long-run lag polynomial to Equation (27.34)

$$\log(\sigma_t^2) = \omega + \frac{\alpha(L)}{\beta(L)} \pi(L) g(z_{t-1}) \tag{27.39}$$

The corresponding FIEGARCH(1, 1) is given by

$$\begin{aligned}\log(\sigma_t^2) &= (1-\beta)\omega + g(z_{t-1}) + \sum_{k=1}^{\infty} \pi_k g(z_{t-1-k}) \\ &- \alpha g(z_{t-2}) - \alpha \sum_{k=1}^{\infty} \pi_k g(z_{t-2-k}) + \beta\log(\sigma_{t-1}^2)\end{aligned} \tag{27.40}$$

Substituting using $z_t = \epsilon_t / \sigma_t$ yields

$$\log(\sigma_t^2) = (1-\beta)\omega + \gamma_1 \frac{\epsilon_{t-1}}{\sigma_{t-1}} + \gamma_2\left(\left|\frac{\epsilon_{t-1}}{\sigma_{t-1}}\right| - \sqrt{2/\pi}\right)\right) \tag{27.41}$$

$$+ \sum_{k=1}^{\infty} \pi_k\left(\gamma_1 \frac{\epsilon_{t-1-k}}{\sigma_{t-1-k}} + \gamma_2\left(\left|\frac{\epsilon_{t-1-k}}{\sigma_{t-1-k}}\right| - \sqrt{2/\pi}\right)\right)$$

$$- \alpha\left(\gamma_1 \frac{\epsilon_{t-2}}{\sigma_{t-2}} + \gamma_2\left(\left|\frac{\epsilon_{t-2}}{\sigma_{t-2}}\right| - \sqrt{2/\pi}\right)\right)$$

$$- \alpha \sum_{k=1}^{\infty} \pi_k\left(\gamma_1 \frac{\epsilon_{t-2-k}}{\sigma_{t-2-k}} + \gamma_2\left(\left|\frac{\epsilon_{t-2-k}}{\sigma_{t-2-k}}\right| - \sqrt{2/\pi}\right)\right)$$

$$+ \beta \log(\sigma_{t-1}^2)$$

As with the FIGARCH model, in practice the infinite sums are truncated at a finite number, typically 1,000 terms.

Estimating a FIEGARCH model

To estimate a FIEGARCH model in EViews simply select **ARCH - Autoregressive Conditional Heteroskedasticity** in the **Method** dropdown at the bottom of the dialog to display the GARCH dialog, then choose **FIEGARCH** in the **Variance and distribution specification** section **Model** dropdown menu.

Note that since as EViews only supports the FIEGARCH(1, 1) specification, you may not specify the number of ARCH and GARCH terms.

You may, as for standard FIGARCH estimation, specify the **Lag truncation** number.

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 41. "The Log Likelihood (LogL) Object," beginning on page 643 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 (in the workfile "Stocks.WF1"). 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:

$$\log(s_t/s_{t-1}) = c_1 + \epsilon_t$$

For the variance specification, we employ an EGARCH(1, 1) model:

$$\log(\sigma_t^2) = \omega + \beta\log(\sigma_{t-1}^2) + \alpha\left|\frac{\epsilon_{t-1}}{\sigma_{t-1}}\right| + \gamma\frac{\epsilon_{t-1}}{\sigma_{t-1}} \tag{27.42}$$

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.

Dependent Variable: DLOG(SPX)
Method: ML ARCH - Student's t distribution (BFGS / Marquardt steps)
Date: 03/09/15 Time: 14:01
Sample: 1/02/1990 12/31/1999
Included observations: 2528
Convergence achieved after 71 iterations
Coefficient covariance computed using outer product of gradients
Presample variance: backcast (parameter = 0.7)
LOG(GARCH) = C(2) + C(3)*ABS(RESID(-1) / @SQRT(GARCH(-1))) + C(4)*RESID(-1) / @SQRT(GARCH(-1)) + C(5)*LOG(GARCH(-1))

Variable	Coefficient	Std. Error	z-Statistic	Prob.
C	0.000513	0.000135	3.810600	0.0001
Variance Equation				
C(2)	-0.196710	0.039150	-5.024490	0.0000
C(3)	0.113675	0.017550	6.477203	0.0000
C(4)	-0.064068	0.011575	-5.535009	0.0000
C(5)	0.988584	0.003360	294.2102	0.0000
T-DIST. DOF	6.703688	0.844702	7.936156	0.0000

R-squared	-0.000032	Mean dependent var	0.000564
Adjusted R-squared	-0.000032	S.D. dependent var	0.008888
S.E. of regression	0.008889	Akaike info criterion	-6.871798
Sum squared resid	0.199653	Schwarz criterion	-6.857949
Log likelihood	8691.953	Hannan-Quinn criter.	-6.866773
Durbin-Watson stat	1.963994		

To test whether there any remaining ARCH effects in the residuals, select **View/Residual Diagnostics/ARCH LM Test...** and specify the order to test. EViews will open the general Heteroskedasticity Tests dialog opened to the ARCH page. Enter “7” in the dialog for the number of lags and click on **OK**.

The top portion of the output from testing up-to an ARCH(7) is given by:

Heteroskedasticity Test: ARCH

F-statistic	0.398895	Prob. F(7,2513)	0.9034
Obs*R-squared	2.798042	Prob. Chi-Square(7)	0.9030

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/Theoretical** 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 775 of *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 expected, 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 the **Forecast** button on the equation toolbar, 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 (DLOG(SPX)^2) as a proxy for the realized volatility. A plot of the proxy against the forecasted volatility for the years 2000 and 2001 provides an indication of the model's ability to track variations in market volatility.

References

Baillie, Richard T., Bollerslev, Tim, Mikkelsen, Hans Ole (1996). "Fractionally integrated generalized autoregressive heteroscedasticity," Journal of Econometrics, 74, 3–30.

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 Mikkelsen, Hans Ole (1996). "Modeling and pricing long memory stock market volatility," Journal of Econometrics, 73, 151–184.

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, Zhuanxin, 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.

Engle, R.F., Ng, V.K. (1993). "Measuring and testing the impact of news on volatility," *Journal of Finance*, 48, 1749–1777.

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 28. Cointegrating Regression

This chapter describes EViews' tools for estimating and testing single equation cointegrating relationships. Three fully efficient estimation methods, Fully Modified OLS (Phillips and Hansen 1992), Canonical Cointegrating Regression (Park 1992), and Dynamic OLS (Saikkonen 1992, Stock and Watson 1993) are described, along with various cointegration testing procedures: Engle and Granger (1987) and Phillips and Ouliaris (1990) residual-based tests, Hansen's (1992b) instability test, and Park's (1992) added variables test.

Notably absent from the discussion is Johansen's (1991, 1995) system maximum likelihood approach to cointegration analysis and testing, which is supported using Var and Group objects, and fully documented in Chapter 44. "Vector Autoregression and Error Correction Models," on page 833 and Chapter 55. "Cointegration Testing," on page 1245. Also excluded are single equation error correction methods which may be estimated using the Equation object and conventional OLS routines (see Phillips and Loretan (1991) for a survey).

The study of cointegrating relationships has been a particularly active areEViews 12a of research. We offer here an abbreviated discussion of the methods used to estimate and test for single equation cointegration in EViews. Those desiring additional detail will find a wealth of sources. Among the many useful overviews of literature are the textbook chapters in Hamilton (1994) and Hayashi (2000), the book length treatment in Maddala and Kim (1999), and the Phillips and Loretan (1991) and Ogaki (1993) survey articles.

Background

It is well known that many economic time series are difference stationary. In general, a regression involving the levels of these I(1) series will produce misleading results, with conventional Wald tests for coefficient significance spuriously showing a significant relationship between unrelated series (Phillips 1986).

Engle and Granger (1987) note that a linear combination of two or more I(1) series may be stationary, or I(0), in which case we say the series are *cointegrated*. Such a linear combination defines a *cointegrating equation* with *cointegrating vector* of weights characterizing the long-run relationship between the variables.

We will work with the standard triangular representation of a regression specification and assume the existence of a single cointegrating vector (Hansen 1992b, Phillips and Hansen 1990). Consider the $n+1$ dimensional time series vector process (y_t, X_t'), with cointegrating equation

$$y_t = X_t'\beta + D_{1t}'\gamma_1 + u_{1t} \tag{28.1}$$

where $D_t = (D_{1t}', D_{2t}')'$ are deterministic trend regressors and the n stochastic regressors X_t are governed by the system of equations:

$$\begin{aligned} X_t &= \Gamma_{21}' D_{1t} + \Gamma_{22}' D_{2t} + \epsilon_{2t} \\ \Delta\epsilon_{2t} &= u_{2t} \end{aligned} \tag{28.2}$$

The p_1-vector of D_{1t} regressors enter into both the cointegrating equation and the regressors equations, while the p_2-vector of D_{2t} are deterministic trend regressors which are included in the regressors equations but excluded from the cointegrating equation (if a non-trending regressor such as the constant is present, it is assumed to be an element of D_{1t} so it is not in D_{2t}).

Following Hansen (1992b), we assume that the innovations $u_t = (u_{1t}, u_{2t}')'$ are strictly stationary and ergodic with zero mean, contemporaneous covariance matrix Σ, one-sided long-run covariance matrix Λ, and covariance matrix Ω, each of which we partition conformably with u_t

$$\begin{aligned} \Sigma &= E(u_t u_t') = \begin{bmatrix} \sigma_{11} & \sigma_{12} \\ \sigma_{21} & \Sigma_{22} \end{bmatrix} \\ \Lambda &= \sum_{j=0}^{\infty} E(u_t u_{t-j}') = \begin{bmatrix} \lambda_{11} & \lambda_{12} \\ \lambda_{21} & \Lambda_{22} \end{bmatrix} \\ \Omega &= \sum_{j=-\infty}^{\infty} E(u_t u_{t-j}') = \begin{bmatrix} \omega_{11} & \omega_{12} \\ \omega_{21} & \Omega_{22} \end{bmatrix} = \Lambda + \Lambda' - \Sigma \end{aligned} \tag{28.3}$$

In addition, we assume a rank n long-run covariance matrix Ω with non-singular submatrix Ω_{22}. Taken together, the assumptions imply that the elements of y_t and X_t are I(1) and cointegrated but exclude both cointegration amongst the elements of X_t and multi-cointegration. Discussions of additional and in some cases alternate assumptions for this specification are provided by Phillips and Hansen (1990), Hansen (1992b), and Park (1992).

It is well-known that if the series are cointegrated, ordinary least squares estimation (static OLS) of the cointegrating vector β in Equation (28.1) is consistent, converging at a faster rate than is standard (Hamilton 1994). One important shortcoming of static OLS (SOLS) is that the estimates have an asymptotic distribution that is *generally* non-Gaussian, exhibit asymptotic bias, asymmetry, and are a function of non-scalar nuisance parameters. Since conventional testing procedures are not valid unless modified substantially, SOLS is generally not recommended if one wishes to conduct inference on the cointegrating vector.

The problematic asymptotic distribution of SOLS arises due to the presence of long-run correlation between the cointegrating equation errors and regressor innovations and (ω_{12}), and cross-correlation between the cointegrating equation errors and the regressors (λ_{12}). In the special case where the X_t are strictly exogenous regressors so that $\omega_{12} = 0$ and $\lambda_{12} = 0$, the bias, asymmetry, and dependence on non-scalar nuisance parameters vanish, and the

SOLS estimator has a fully efficient asymptotic Gaussian mixture distribution which permits standard Wald testing using conventional limiting χ^2-distributions.

Alternately, SOLS has an asymptotic Gaussian mixture distribution if the number of deterministic trends excluded from the cointegrating equation p_2 is no less than the number of stochastic regressors n. Let $m_2 = \max(n - p_2, 0)$ represent the number of cointegrating regressors less the number of deterministic trend regressors excluded from the cointegrating equation. Then, roughly speaking, when $m_2 = 0$, the deterministic trends in the regressors asymptotically dominate the stochastic trend components in the cointegrating equation.

While Park (1992) notes that these two cases are rather exceptional, they are relevant in motivating the construction of our three asymptotically efficient estimators and computation of critical values for residual-based cointegration tests. Notably, the fully efficient estimation methods supported by EViews involve transformations of the data or modifications of the cointegrating equation specification to mimic the strictly exogenous X_t case.

Estimating a Cointegrating Regression

EViews offers three methods for estimating a single cointegrating vector: Fully Modified OLS (FMOLS), Canonical Cointegrating Regression (CCR), and Dynamic OLS (DOLS). Static OLS is supported as a special case of DOLS. We emphasize again that Johansen's (1991, 1995) system maximum likelihood approach is discussed in Chapter 44. "Vector Autoregression and Error Correction Models," on page 833.

The equation object is used to estimate a cointegrating equation. First, create an equation object, select **Object/New Object.../Equation** or **Quick/Estimate Equation...** then select **COINTREG - Cointegrating Regression** in the **Method** dropdown menu. The dialog will show settings appropriate for your cointegrating regression. Alternately, you may enter the `cointreg` keyword in the command window to perform both steps.

There are three parts to specifying your equation. First, you should use the first two sections of the dialog (**Equation specification** and **Cointegrating regressors specification**) to specify your triangular system of equations. Second, you will use the **Nonstationary estimation settings** section to specify the basic cointegrating regression estimation method. Lastly, you should enter a sample specification, then click on **OK** to estimate the equation. (We ignore, for a moment, the options settings on the **Options** tab.)

Specifying the Equation

The first two sections of the dialog (**Equation specification** and **Cointegrating regressors specification**) are used to describe your cointegrating and regressors equations.

Equation Specification

The cointegrating equation is described in the **Equation specification** section. You should enter the name of the dependent variable, y, followed by a list of cointegrating regressors, X, in the edit field, then use the **Trend specification** dropdown to choose from

a list of deterministic trend variable assumptions (**None**, **Constant (Level)**, **Linear Trend**, **Quadratic Trend**). The dropdown menu selections imply trends up to the specified order so that the **Quadratic Trend** selection depicted includes a constant and a linear trend term along with the quadratic.

If you wish to add deterministic regressors that are not offered in the pre-specified list to D_1, you may enter the series names in the **Deterministic regressors** edit box.

Cointegrating Regressors Specification

Cointegrating Regressors Specification section of the dialog completes the specification of the regressors equations.

First, if there are any D_2 deterministic regressors (regressors that are included in the regressors equations but not in the cointegrating equation), they should be specified here using the **Additional trends** dropdown menu or by entering regressors explicitly using the **Additional deterministic regressors** edit field.

Second, you should indicate whether you wish to estimate the regressors innovations u_{2t} indirectly by estimating the regressors equations in levels and then differencing the residuals or directly by estimating the regressors equations in differences. Check the box for **Estimate using differenced data** (which is only relevant and only appears if you are estimating your equation using FMOLS or CCR) to estimate the regressors equations in differences.

Specifying an Estimation Method

Once you specify your cointegrating and regressor equations you are ready to describe your estimation method. The EViews equation object offers three methods for estimating a single cointegrating vector: Fully Modified OLS (FMOLS), Canonical Cointegrating Regression (CCR), and Dynamic OLS (DOLS). We again emphasize that Johansen's (1991, 1995) system maximum likelihood approach is described elsewhere("Vector Error Correction (VEC) Models" on page 881).

The **Nonstationary estimation settings** section is used to describe your estimation method. First, you should use the **Method** dropdown menu to choose one of the three methods. Both the main dialog page and the options page will change to display the options associated with your selection.

Fully Modified OLS

Phillips and Hansen (1990) propose an estimator which employs a semi-parametric correction to eliminate the problems caused by the long run correlation between the cointegrating equation and stochastic regressors innovations. The resulting Fully Modified OLS (FMOLS) estimator is asymptotically unbiased and has fully efficient mixture normal asymptotics allowing for standard Wald tests using asymptotic Chi-square statistical inference.

The FMOLS estimator employs preliminary estimates of the symmetric and one-sided long-run covariance matrices of the residuals. Let $\hat{u}_{1t}$ be the residuals obtained after estimating Equation (28.1). The $\hat{u}_{2t}$ may be obtained indirectly as $\hat{u}_{2t} = \Delta\hat{\epsilon}_{2t}$ from the levels regressions

$$X_t = \hat{\Gamma}_{21}' D_{1t} + \hat{\Gamma}_{22}' D_{2t} + \hat{\epsilon}_{2t} \tag{28.4}$$

or directly from the difference regressions

$$\Delta X_t = \hat{\Gamma}_{21}' \Delta D_{1t} + \hat{\Gamma}_{22}' \Delta D_{2t} + \hat{u}_{2t} \tag{28.5}$$

Let $\hat{\Omega}$ and $\hat{\Lambda}$ be the long-run covariance matrices computed using the residuals $\hat{u}_t = (\hat{u}_{1t}, \hat{u}_{2t}')'$. Then we may define the modified data

$$y_t^+ = y_t - \hat{\omega}_{12}\hat{\Omega}_{22}^{-1}\hat{u}_2 \tag{28.6}$$

and an estimated bias correction term

$$\hat{\lambda}_{12}^+ = \hat{\lambda}_{12} - \hat{\omega}_{12}\hat{\Omega}_{22}^{-1}\hat{\Lambda}_{22} \tag{28.7}$$

The FMOLS estimator is given by

$$\hat{\theta} = \begin{bmatrix} \hat{\beta} \\ \hat{\gamma}_1 \end{bmatrix} = \left(\sum_{t=2}^{T} Z_t Z_t' \right)^{-1} \left(\sum_{t=2}^{T} Z_t y_t^+ - T \begin{bmatrix} \hat{\lambda}_{12}^{+\prime} \\ 0 \end{bmatrix} \right) \tag{28.8}$$

where $Z_t = (X_t', D_t')'$. The key to FMOLS estimation is the construction of long-run covariance matrix estimators $\hat{\Omega}$ and $\hat{\Lambda}$.

Before describing the options available for computing $\hat{\Omega}$ and $\hat{\Lambda}$, it will be useful to define the scalar estimator

$$\hat{\omega}_{1.2} = \hat{\omega}_{11} - \hat{\omega}_{12}\hat{\Omega}_{22}^{-1}\hat{\omega}_{21} \tag{28.9}$$

which may be interpreted as the estimated long-run variance of u_{1t} conditional on u_{2t}. We may, if desired, apply a degree-of-freedom correction to $\hat{\omega}_{1.2}$.

Hansen (1992) shows that the Wald statistic for the null hypothesis $R\theta = r$

$$W = (R\hat{\theta} - r)'(R\,V(\hat{\theta})R')^{-1}(R\hat{\theta} - r) \tag{28.10}$$

with

$$V(\hat{\theta}) = \hat{\omega}_{1.2}\left(\sum_{t=2}^{T} Z_t Z_t' \right)^{-1} \tag{28.11}$$

has an asymptotic χ_g^2-distribution, where g is the number of restrictions imposed by R. (You should bear in mind that restrictions on the constant term and any other non-trending variables are not testable using the theory underlying Equation (28.10).)

To estimate your equation using FMOLS, select **Fully-modified OLS (FMOLS)** in the **Nonstationary estimation settings** dropdown menu.

The main dialog and options pages will change to show the available settings.

To illustrate the FMOLS estimator, we employ data for (100 times) log real quarterly aggregate personal disposable income (LY) and personal consumption expenditures (LC) for the U.S. from 1947q1 to 1989q3 as described in Hamilton (2000, p. 600, 610) and contained in the workfile "Hamilton_coint.WF1".

We wish to estimate a model that includes an intercept in the cointegrating equation, has no additional deterministics in the regressors equations, and estimates the regressors equations in non-differenced form.

By default, EViews will estimate Ω and Λ using a (non-prewhitened) kernel approach with a Bartlett kernel and Newey-West fixed bandwidth. To change the whitening or kernel settings, click on the **Long-run variance calculation: Options** button and enter your changes in the sub-dialog.

Here we have specified that the long-run variances be computed using a nonparametric method with the Bartlett kernel and a real-valued bandwidth chosen by Andrews' automatic bandwidth selection method.

In addition, you may use the **Options** tab of the **Equation Estimation** dialog to modify the computation of the coefficient covariance. By default, EViews computes the coefficient covariance by rescaling the usual OLS covariances using the $\hat{\omega}_{1.2}$ obtained from the estimated $\hat{\Omega}$ after applying a degrees-of-freedom correction. In our example, we will use the checkbox on the **Options** tab (not depicted) to remove the d.f. correction.

The estimates for this specification are given by:

Dependent Variable: LC
Method: Fully Modified Least Squares (FMOLS)
Date: 08/11/09 Time: 13:19
Sample (adjusted): 1947Q2 1989Q3
Included observations: 170 after adjustments
Cointegrating equation deterministics: C
Long-run covariance estimate (Bartlett kernel, Andrews bandwidth = 14.9878)
No d.f. adjustment for standard errors & covariance

Variable	Coefficient	Std. Error	t-Statistic	Prob.
LY	0.987548	0.009188	107.4880	0.0000
C	-0.035023	6.715362	-0.005215	0.9958

R-squared	0.998171	Mean dependent var	720.5078
Adjusted R-squared	0.998160	S.D. dependent var	41.74069
S.E. of regression	1.790506	Sum squared resid	538.5929
Durbin-Watson stat	0.406259	Long-run variance	25.46653

The top portion of the results describe the settings used in estimation, in particular, the specification of the deterministic regressors in the cointegrating equation, the kernel nonparametric method used to compute the long-run variance estimators $\hat{\Omega}$ and $\hat{\Lambda}$, and the no-d.f. correction option used in the calculation of the coefficient covariance. Also displayed is the bandwidth of 14.9878 selected by the Andrews automatic bandwidth procedure.

The estimated coefficients are presented in the middle of the output. Of central importance is the coefficient on LY which implies that the estimated cointegrating vector for LC and LY

(1, -0.9875). Note that we present the standard error, t-statistic, and p-value for the constant even though they are not, strictly speaking, valid.

The summary statistic portion of the output is relatively familiar but does require a bit of comment. First, all of the descriptive and fit statistics are computed using the original data, not the FMOLS transformed data. Thus, while the measures of fit and the Durbin-Watson stat may be of casual interest, you should exercise extreme caution in using these measures. Second, EViews displays a "Long-run variance" value which is an estimate of the long-run variance of u_{1t} conditional on u_{2t}. This statistic, which takes the value of 25.47 in this example, is the $\hat{\omega}_{1.2}$ employed in forming the coefficient covariances, and is obtained from the $\hat{\Omega}$ and $\hat{\Lambda}$ used in estimation. Since we are not d.f. correcting the coefficient covariance matrix the $\hat{\omega}_{1.2}$ reported here is not d.f. corrected.

Once you have estimated your equation using FMOLS you may use the various cointegrating regression equation views and procedures. We will discuss these tools in greater depth in ("Working with an Equation" on page 317), but for now we focus on a simple Wald test for the coefficients. To test for whether the cointegrating vector is (1, -1), select **View/Coefficient Diagnostics/Wald Test - Coefficient Restrictions** and enter "C(1) = 1" in the dialog. EViews displays the output for the test:

Wald Test:
Equation: FMOLS
Null Hypothesis: C(1)=1

Test Statistic	Value	df	Probability
t-statistic	-1.355362	168	0.1771
F-statistic	1.837006	(1, 168)	0.1771
Chi-square	1.837006	1	0.1753

Null Hypothesis Summary:

Normalized Restriction (= 0)	Value	Std. Err.
-1 + C(1)	-0.012452	0.009188

Restrictions are linear in coefficients.

The t-statistic and Chi-square p-values are both around 0.17, indicating that the we cannot reject the null hypothesis that the cointegrating regressor coefficient value is equal to 1.

Note that this Wald test is for a simple linear restriction. Hansen points out that his theoretical results do not directly extend to testing nonlinear hypotheses in models with trend regressors, but EViews does allow tests with nonlinear restrictions since others, such as Phillips and Loretan (1991) and Park (1992) provide results in the absence of the trend regressors. We do urge caution in interpreting nonlinear restriction test results for equations involving such regressors.

Canonical Cointegrating Regression

Park's (1992) Canonical Cointegrating Regression (CCR) is closely related to FMOLS, but instead employs stationary transformations of the (y_{1t}, X_t') data to obtain least squares estimates to remove the long run dependence between the cointegrating equation and stochastic regressors innovations. Like FMOLS, CCR estimates follow a mixture normal distribution which is free of non-scalar nuisance parameters and permits asymptotic Chi-square testing.

As in FMOLS, the first step in CCR is to obtain estimates of the innovations $\hat{u}_t = (\hat{u}_{1t}, \hat{u}_{2t}')'$ and corresponding consistent estimates of the long-run covariance matrices $\hat{\Omega}$ and $\hat{\Lambda}$. Unlike FMOLS, CCR also requires a consistent estimator of the contemporaneous covariance matrix $\hat{\Sigma}$.

Following Park, we extract the columns of $\hat{\Lambda}$ corresponding to the one-sided long-run covariance matrix of $\hat{u}_t$ and (the levels and lags of) $\hat{u}_{2t}$

$$\hat{\Lambda}_2 = \begin{bmatrix} \hat{\lambda}_{12} \\ \hat{\Lambda}_{22} \end{bmatrix} \tag{28.12}$$

and transform the (y_{1t}, X_t') using

$$\begin{aligned} X_t^* &= X_t - (\hat{\Sigma}^{-1}\hat{\Lambda}_2)'\hat{u}_t \\ y_t^* &= y_t - \left(\hat{\Sigma}^{-1}\hat{\Lambda}_2\tilde{\beta} + \begin{bmatrix} 0 \\ \hat{\Omega}_{22}^{-1}\hat{\omega}_{21} \end{bmatrix}\right)'\hat{u}_t \end{aligned} \tag{28.13}$$

where the $\tilde{\beta}$ are estimates of the cointegrating equation coefficients, typically the SOLS estimates used to obtain the residuals $\hat{u}_{1t}$.

The CCR estimator is defined as ordinary least squares applied to the transformed data

$$\begin{bmatrix} \beta \\ \hat{\gamma}_1 \end{bmatrix} = \left(\sum_{t=1}^{T} Z_t^* Z_t^{*\prime}\right)^{-1} \sum_{t=1}^{T} Z_t^* y_t^* \tag{28.14}$$

where $Z_t^* = (Z_t^{*\prime}, D_{1t}')'$.

Park shows that the CCR transformations asymptotically eliminate the endogeneity caused by the long run correlation of the cointegrating equation errors and the stochastic regressors innovations, and simultaneously correct for asymptotic bias resulting from the contemporaneous correlation between the regression and stochastic regressor errors. Estimates based on the CCR are therefore fully efficient and have the same unbiased, mixture normal asymptotics as FMOLS. Wald testing may be carried out as in Equation (28.10) with Z_t^* used in place of Z_t in Equation (28.11).

To estimate your equation using CCR, select **Canonical Cointegrating Regression (CCR)** in the **Nonstationary estimation settings** dropdown menu. The main dialog and options pages for CCR are identical to those for FMOLS.

To continue with our consumption and disposable income example, suppose we wish to estimate the same specification as before by CCR, using prewhitened Quadratic-spectral kernel estimators of the long-run covariance matrices. Fill out the equation specification portion of the dialog as before, then click on the **Long-run variance calculation: Options** button to change the calculation method. Here, we have specified a (fixed lag) VAR(1) for the prewhitening method and have changed our kernel shape to quadratic spectral. Click on **OK** to accept the covariance options

Once again go to the **Options** tab to turn off d.f. correction for the coefficient covariances so that they match those from FMOLS. Click on **OK** again to accept the estimation options.

The results are presented below:

Dependent Variable: LC
Method: Canonical Cointegrating Regression (CCR)
Date: 08/11/09 Time: 13:25
Sample (adjusted): 1947Q2 1989Q3
Included observations: 170 after adjustments
Cointegrating equation deterministics: C
Long-run covariance estimate (Prewhitening with lags = 1, Quadratic
-Spectral kernel, Andrews bandwidth = 1.5911)
No d.f. adjustment for standard errors & covariance

Variable	Coefficient	Std. Error	t-Statistic	Prob.
LY	0.988975	0.007256	136.3069	0.0000
C	-1.958828	5.298819	-0.369673	0.7121

R-squared	0.997780	Mean dependent var	720.5078
Adjusted R-squared	0.997767	S.D. dependent var	41.74069
S.E. of regression	1.972481	Sum squared resid	653.6343
Durbin-Watson stat	0.335455	Long-run variance	15.91571

The first thing we note is that the VAR prewhitening has a strong effect on the kernel part of the calculation of the long-run covariances, shortening the Andrews optimal bandwidth from almost 15 down to 1.6. Furthermore, as a result of prewhitening, the estimate of the conditional long-run variance changes quite a bit, decreasing from 25.47 to 15.92. This decrease contributes to estimated coefficient standard errors for CCR that are smaller than their FMOLS counterparts. Differences aside, however, the estimates of the cointegrating vector are qualitatively similar. In particular, a Wald test of the null hypothesis that the cointegrating vector is equal to (1, -1) yields a p-value of 0.1305.

Dynamic OLS

A simple approach to constructing an asymptotically efficient estimator that eliminates the feedback in the cointegrating system has been advocated by Saikkonen (1992) and Stock and Watson (1993). Termed Dynamic OLS (DOLS), the method involves augmenting the cointegrating regression with lags *and leads* of ΔX_t so that the resulting cointegrating equation error term is orthogonal to the entire history of the stochastic regressor innovations:

$$y_t = X_t'\beta + D_{1t}'\gamma_1 + \sum_{j=-q}^{r} \Delta X_{t+j}'\delta + v_{1t} \tag{28.15}$$

Under the assumption that adding q lags and r leads of the differenced regressors soaks up all of the long-run correlation between u_{1t} and u_{2t}, least-squares estimates of $\theta = (\beta', \gamma')'$ using Equation (28.15) have the same asymptotic distribution as those obtained from FMOLS and CCR.

An estimator of the asymptotic variance matrix of $\hat{\theta}$ may be computed by computing the usual OLS coefficient covariance, but replacing the usual estimator for the residual variance of v_{1t} with an estimator of the long-run variance of the residuals. Alternately, you could compute a robust HAC estimator of the coefficient covariance matrix.

To estimate your equation using DOLS, first fill out the equation specification, then select **Dynamic OLS (DOLS)** in the **Nonstationary estimation settings** dropdown menu. The dialog will change to display settings for DOLS.

By default, the **Lag & lead method** is **Fixed** with **Lags** and **Leads** each set to 1. You may specify a different number of lags or leads or you can use the dropdown to elect automatic information criterion selection of the lag and lead orders by selecting **Akaike**, **Schwarz**, or **Hannan-Quinn**. If you select **None**, EViews will estimate SOLS.

If you select one of the info criterion selection methods, you will be prompted for a maximum lag and lead length. You may enter a value, or you may retain the default entry "*" which instructs EViews to use an arbitrary observation-based rule-of-thumb:

$$\mathrm{int}(\min((T-k)/3,\ 12)\cdot(T/100)^{1/4}) \tag{28.16}$$

to set the maximum, where k is the number of coefficients in the cointegrating equation. This rule-of-thumb is a slightly modified version of the rule suggested by Schwert (1989) in the context of unit root testing. (We urge careful thought in the use of automatic selection methods since the purpose of including leads and lags is to remove long-run dependence by orthogonalizing the equation residual with respect to the history of stochastic regressor innovations; the automatic methods were not designed to produce this effect.)

For DOLS estimation we may also specify the method used to compute the coefficient covariance matrix. Click on the **Options** tab of the dialog to see the relevant options.

The dropdown menu allows you to choose between the **Default (rescaled OLS)**, **Ordinary Least Squares**, **White**, or **HAC - Newey West.** The default computation method rescales the ordinary least squares coefficient covariance using an estimator of the long-run variance of DOLS residuals (multiplying by the ratio of the long-run variance to the ordinary squared standard error). Alternately, you may employ a sandwich-style **HAC (Newey-West)** covariance matrix estimator. In both cases, the **HAC Options** button may be used to override the default method for computing the long-run variance (non-prewhitened Bartlett kernel and a Newey-West fixed bandwidth). In addition, EViews offers options for estimating the coefficient covariance using the **White** covariance or **Ordinary Least Squares** methods. These methods are offered primarily for comparison purposes.

Lastly, the **Options** tab may be used to remove the degree-of-freedom correction that is applied to the estimate of the conditional long-run variance or robust coefficient covariance.

We illustrate the technique by estimating an example from Hamilton (19.3.31, p. 611) using the consumption and income data discussed earlier. The model employs an intercept-trend specification for the cointegrating equation, with no additional deterministics in the regressors equations, and four lags and leads of the differenced cointegrating regressor to eliminate long run correlation between the innovations.

Here, we have entered the cointegrating equation specification in the top portion of the dialog, and chosen **Dynamic OLS (DOLS)** as our estimation method, and specified a **Fixed** lag and lead length of 4.

In computing the covariance matrix, Hamilton computes the long-run variance of the residuals using an AR(2) whitening regression with no d.f. correction. To match Hamilton's computations, we click on the **Options** tab to display the covariance. First, turn off the adjustment for degrees of freedom by unchecking the **d.f. Adjustment** box. Next, with the dropdown set to **Default (rescaled OLS)**, click on the **HAC Options** button to display the **Long-run Variance Options** dialog. Select a **Fixed** lag specification of 2, and choose the **None** kernel. Click on **OK** to accept the HAC settings, then on **OK** again to estimate the equation.

The estimation results are given below:

Dependent Variable: LC
Method: Dynamic Least Squares (DOLS)
Date: 08/11/09 Time: 13:37
Sample (adjusted): 1948Q2 1988Q3
Included observations: 162 after adjustments
Cointegrating equation deterministics: C @TREND
Fixed leads and lags specification (lead=4, lag=4)
Long-run variance estimate (Prewhitening with lags = 2, None kernel)
No d.f. adjustment for standard errors & covariance

Variable	Coefficient	Std. Error	t-Statistic	Prob.
LY	0.681179	0.071981	9.463267	0.0000
C	199.1406	47.20878	4.218297	0.0000
@TREND	0.268957	0.062004	4.337740	0.0000

R-squared	0.999395	Mean dependent var	720.5532
Adjusted R-squared	0.999351	S.D. dependent var	39.92349
S.E. of regression	1.017016	Sum squared resid	155.1484
Durbin-Watson stat	0.422921	Long-run variance	10.19830

The top portion describes the settings used in estimation, showing the trend assumptions, the lag and lead specification, and method for computing the long-run variance used in forming the coefficient covariances. The actual estimate of the latter, in this case 10.198, is again displayed in the bottom portion of the output (if you had selected OLS as your coefficient covariance methods, this value would be simply be the ordinary S.E. of the regression; if you had selected White or HAC, the statistic would not have been computed).

The estimated coefficients are displayed in the middle of the output. First, note that EViews does not display the results for the lags and leads of the differenced cointegrating regressors since we cannot perform inference on these short-term dynamics nuisance parameters. Second, the coefficient on the linear trend is statistically different from zero at conventional levels, indicating that there is a deterministic time trend common to both LC and LY. Lastly, the estimated cointegrating vector for LC and LY is (1, -0.6812), which differs qualitatively from the earlier results. A Wald test of the restriction that the cointegrating vector is (1, -1) yields a *t*-statistic of -4.429, strongly rejecting that null hypothesis.

While EViews does not display the coefficients for the short-run dynamics, the short-run coefficients are used in constructing the fit statistics in the bottom portion of the results view (we again urge caution in using these measures). The short-run dynamics are also used in computing the residuals used by various equation views and procs such as the residual plot or the gradient view.

The short-run coefficients are not included in the representations view of the equation, which focuses only on the estimates for Equation (28.1). Furthermore, forecasting and model solution using an equation estimated by DOLS are also based on the long-run relationship. If you wish to construct forecasts that incorporate the short-run dynamics, you

may use least squares to estimate an equation that explicitly includes the lags and leads of the cointegrating regressors.

Testing for Cointegration

In the single equation setting, EViews provides views that perform Engle and Granger (1987) and Phillips and Ouliaris (1990) residual-based tests, Hansen's instability test (Hansen 1992b), and Park's $H(p, q)$ added variables test (Park 1992).

System cointegration testing using Johansen's methodology is described in "Johansen Cointegration Test" on page 1245.

Note that the Engle-Granger and Phillips-Perron tests may also be performed as a view of a Group object.

Residual-based Tests

The Engle-Granger and Phillips-Ouliaris residual-based tests for cointegration are simply unit root tests applied to the residuals obtained from SOLS estimation of Equation (28.1). Under the assumption that the series are *not* cointegrated, *all* linear combinations of (y_t, X_t'), including the residuals from SOLS, are unit root nonstationary. Therefore, a test of the *null hypothesis of no cointegration* against the *alternative of cointegration* corresponds to a unit root test of the null of nonstationarity against the alternative of stationarity.

The two tests differ in the method of accounting for serial correlation in the residual series; the Engle-Granger test uses a parametric, augmented Dickey-Fuller (ADF) approach, while the Phillips-Ouliaris test uses the nonparametric Phillips-Perron (PP) methodology.

The Engle-Granger test estimates a p-lag augmented regression of the form

$$\Delta \hat{u}_{1t} = (\rho - 1)\hat{u}_{1t-1} + \sum_{j=1}^{p} \delta_j \Delta \hat{u}_{1t-j} + v_t \tag{28.17}$$

The number of lagged differences p should increase to infinity with the (zero-lag) sample size T but at a rate slower than $T^{1/3}$.

We consider the two standard ADF test statistics, one based on the *t*-statistic for testing the null hypothesis of nonstationarity $(\rho = 1)$ and the other based directly on the normalized autocorrelation coefficient $\hat{\rho} - 1$:

$$\begin{aligned} \hat{\tau} &= \frac{\hat{\rho} - 1}{se(\hat{\rho})} \\ \hat{z} &= \frac{T(\hat{\rho} - 1)}{\left(1 - \sum_j \hat{\delta}_j\right)} \end{aligned} \tag{28.18}$$

where $se(\hat{\rho})$ is the usual OLS estimator of the standard error of the estimated $\hat{\rho}$

$$se(\hat{\rho}) = \hat{s}_v \left(\sum_t \hat{u}_{1t-1}^2 \right)^{-1/2} \tag{28.19}$$

(Stock 1986, Hayashi 2000). There is a practical question as to whether the standard error estimate in Equation (28.19) should employ a degree-of-freedom correction. Following common usage, EViews standalone unit root tests and the Engle-Granger cointegration tests both use the d.f.-corrected estimated standard error $\hat{s}_v$, with the latter test offering an option to turn off the correction.

In contrast to the Engle-Granger test, the Phillips-Ouliaris test obtains an estimate of ρ by running the unaugmented Dickey-Fuller regression

$$\Delta \hat{u}_{1t} = (\rho - 1)\hat{u}_{1t-1} + w_t \tag{28.20}$$

and using the results to compute estimates of the long-run variance ω_w and the strict one-sided long-run variance λ_{1w} of the residuals. By default, EViews d.f.-corrects the estimates of both long-run variances, but the correction may be turned off. (The d.f. correction employed in the Phillips-Ouliaris test differs slightly from the ones in FMOLS and CCR estimation since the former applies to the estimators of both long-run variances, while the latter apply only to the estimate of the conditional long-run variance).

The bias corrected autocorrelation coefficient is then given by

$$(\hat{\rho}^* - 1) = (\hat{\rho} - 1) - T\hat{\lambda}_{1w} \left(\sum_t \hat{u}_{1t-1}^2 \right)^{-1} \tag{28.21}$$

The test statistics corresponding to Equation (28.18) are

$$\begin{aligned} \hat{\tau} &= \frac{\hat{\rho}^* - 1}{se(\hat{\rho}^*)} \\ \hat{z} &= T(\hat{\rho}^* - 1) \end{aligned} \tag{28.22}$$

where

$$se(\hat{\rho}^*) = \hat{\omega}_w^{1/2} \left(\sum_t \hat{u}_{1t-1}^2 \right)^{-1/2} \tag{28.23}$$

As with ADF and PP statistics, the asymptotic distributions of the Engle-Granger and Phillips-Ouliaris z and τ statistics are non-standard and depend on the deterministic regressors specification, so that critical values for the statistics are obtained from simulation results. Note that the dependence on the deterministics occurs despite the fact that the auxiliary regressions themselves exclude the deterministics (since those terms have already been removed from the residuals). In addition, the critical values for the ADF and PP test statistics must account for the fact that the residuals used in the tests depend upon estimated coefficients.

MacKinnon (1996) provides response surface regression results for obtaining critical values for four different assumptions about the deterministic regressors in the cointegrating equation (none, constant (level), linear trend, quadratic trend) and values of $k = m_2 + 1$ from 1 to 12. (Recall that $m_2 = \max(n - p_2, 0)$ is the number of cointegrating regressors less the number of deterministic trend regressors excluded from the cointegrating equation.) When computing critical values, EViews will ignore the presence of any user-specified deterministic regressors since corresponding simulation results are not available. Furthermore, results for $k = 12$ will be used for cases that exceed that value.

Continuing with our consumption and income example from Hamilton, we construct Engle-Granger and Phillips-Ouliaris tests from an estimated equation where the deterministic regressors include a constant and linear trend. Since SOLS is used to obtain the first-stage residuals, the test results do not depend on the method used to estimate the original equation, only the specification itself is used in constructing the test.

To perform the Engle-Granger test, open an estimated equation and select **View/Cointegration and** select **Engle-Granger** in the **Test Method** dropdown. The dialog will change to display the options for this specifying the number p of augmenting lags in the ADF regression.

By default, EViews uses automatic lag-length selection using the Schwarz information criterion. The default number of lags is the observation-based rule given in Equation (28.16). Alternately you may specify a **Fixed (User-specified)** lag-length, select a different information criterion (**Akaike**, **Hannan-Quinn**, **Modified Akaike**, **Modified Schwarz**, or **Modified Hannan-Quinn**), or specify sequential testing of the highest order lag using a t-statistic and specified p-value threshold. For our purposes the default settings suffice so simply click on **OK**.

The Engle-Granger test results are divided into three distinct sections. The first portion displays the test specification and settings, along with the test values and corresponding p-values:

Cointegration Test - Engle-Granger
Date: 04/21/09 Time: 10:37
Equation: EQ_DOLS
Specification: LC LY C @TREND
Cointegrating equation deterministics: C @TREND
Null hypothesis: Series are not cointegrated
Automatic lag specification (lag=1 based on Schwarz Info Criterion, maxlag=13)

	Value	Prob.*
Engle-Granger tau-statistic	-4.536843	0.0070
Engle-Granger z-statistic	-33.43478	0.0108

*MacKinnon (1996) p-values.

The probability values are derived from the MacKinnon response surface simulation results. In settings where using the MacKinnon results may not be appropriate, for example when the cointegrating equation contains user-specified deterministic regressors or when there are more than 12 stochastic trends in the asymptotic distribution, EViews will display a warning message below these results.

Looking at the test description, we first confirm that the test statistic is computed using C and @TREND as deterministic regressors, and note that the choice to include a single lagged difference in the ADF regression was determined using automatic lag selection with a Schwarz criterion and a maximum lag of 13.

As to the tests themselves, the Engle-Granger tau-statistic (t-statistic) and normalized autocorrelation coefficient (which we term the z-statistic) both reject the null hypothesis of no cointegration (unit root in the residuals) at the 5% level. In addition, the tau-statistic rejects at a 1% significance level. On balance, the evidence clearly suggests that LC and LY are cointegrated.

The middle section of the output displays intermediate results used in constructing the test statistic that may be of interest:

Intermediate Results:	
Rho - 1	-0.241514
Rho S.E.	0.053234
Residual variance	0.642945
Long-run residual variance	0.431433
Number of lags	1
Number of observations	169
Number of stochastic trends**	2

**Number of stochastic trends in asymptotic distribution.

Most of the entries are self-explanatory, though a few deserve a bit of discussion. First, the "Rho S.E." and "Residual variance" are the (possibly) d.f. corrected coefficient standard error and the squared standard error of the regression. Next, the "Long-run residual variance" is the estimate of the long-run variance of the residual based on the estimated para-

metric model. The estimator is obtained by taking the residual variance and dividing it by the square of 1 minus the sum of the lag difference coefficients. These residual variance and long-run variances are used to obtain the denominator of the z-statistic (Equation (28.18)). Lastly, the "Number of stochastic trends" entry reports the $k = m_2 + 1$ value used to obtain the p-values. In the leading case, k is simply the number of cointegrating variables (including the dependent) in the system, but the value must generally account for deterministic trend terms in the system that are excluded from the cointegrating equation.

The bottom section of the output depicts the results for the actual ADF test equation:

Engle-Granger Test Equation:
Dependent Variable: D(RESID)
Method: Least Squares
Date: 04/21/09 Time: 10:37
Sample (adjusted): 1947Q3 1989Q3
Included observations: 169 after adjustments

Variable	Coefficient	Std. Error	t-Statistic	Prob.
RESID(-1)	-0.241514	0.053234	-4.536843	0.0000
D(RESID(-1))	-0.220759	0.071571	-3.084486	0.0024

R-squared	0.216944	Mean dependent var	-0.024433
Adjusted R-squared	0.212255	S.D. dependent var	0.903429
S.E. of regression	0.801838	Akaike info criterion	2.407945
Sum squared resid	107.3718	Schwarz criterion	2.444985
Log likelihood	-201.4713	Hannan-Quinn criter.	2.422976
Durbin-Watson stat	1.971405		

Alternately, you may compute the Phillips-Ouliaris test statistic. Simply select **View/Cointegration** and choose **Phillips-Ouliaris** in the **Test Method** dropdown.

The dialog changes to show a single **Options** button for controlling the estimation of the long-run variance ω_w and the strict one-sided long-run variance λ_{1w}. The default settings instruct EViews to compute these long-run variances using a non-prewhitened Bartlett kernel estimator with a fixed Newey-West bandwidth. To change these settings, click on the **Options** button and fill out the dialog. Since the default settings are sufficient for our needs, simply click on the **OK** button to compute the test statistics.

As before, the output may be divided into three parts; we will focus on the first two. The test results are given by:

Cointegration Test - Phillips-Ouliaris
Date: 02/08/13 Time: 13:11
Equation: EQ_19_3_29
Specification: LC LY C @TREND
Cointegrating equation deterministics: C @TREND
Null hypothesis: Series are not cointegrated
Long-run variance estimate (Bartlett kernel, Newey-West fixed bandwidth = 5.0000)
No d.f. adjustment for variances

	Value	Prob.*
Phillips-Ouliaris tau-statistic	-5.138345	0.0009
Phillips-Ouliaris z-statistic	-43.62100	0.0010

At the top of the output EViews notes that we estimated the long-run variance and one-sided long run variance using a Bartlett kernel and an number of observations based bandwidth of 5.0. More importantly, the test statistics show that, as with the Engle-Granger tests, the Phillips-Ouliaris tests reject the null hypothesis of no cointegration (unit root in the residuals) at roughly the 1% significance level.

The intermediate results are given by:

Intermediate Results:	
Rho - 1	-0.279221
Bias corrected Rho - 1 (Rho* - 1)	-0.256594
Rho* S.E.	0.049937
Residual variance	0.730377
Long-run residual variance	0.659931
Long-run residual autocovariance	-0.035223
Number of observations	170
Number of stochastic trends**	2

**Number of stochastic trends in asymptotic distribution.

There are a couple of new results. The "Bias corrected Rho - 1" reports the estimated value of Equation (28.21) and the "Rho* S.E." corresponds to Equation (28.23). The "Long-run residual variance" and "Long-run residual autocovariance" are the estimates of ω_w and λ_{1w}, respectively. It is worth noting that the ratio of $\hat{\omega}_w^{1/2}$ to the S.E. of the regression, which is a measure of the amount of residual autocorrelation in the long-run variance, is the scaling factor used in adjusting the raw t-statistic to form tau.

The bottom portion of the output displays results for the test equation.

Hansen's Instability Test

Hansen (1992) outlines a test of the null hypothesis of cointegration against the alternative of no cointegration. He notes that under the alternative hypothesis of no cointegration, one should expect to see evidence of parameter instability. He proposes (among others) use of

the L_c test statistic, which arises from the theory of Lagrange Multiplier tests for parameter instability, to evaluate the stability of the parameters.

The L_c statistic examines time-variation in the scores from the estimated equation. Let $\mathfrak{s}_t$ be the vector of estimated individual score contributions from the estimated equation, and define the partial sums,

$$\mathfrak{S}_t = \sum_{t=2}^{T} \mathfrak{s}_t \tag{28.24}$$

where $\mathfrak{S}_t = 0$ by construction. For FMOLS, we have

$$\mathfrak{s}_t = (Z_t \hat{u}_t^+) - \begin{bmatrix} \hat{\lambda}_{12}^{+\prime} \\ 0 \end{bmatrix} \tag{28.25}$$

where $\hat{u}_t^+ = y_t^+ - X_t'\hat{\theta}$ is the residual for the transformed regression. Then Hansen chooses a constant measure of the parameter instability $\hat{G}$ and forms the statistic

$$L_c = \sum_{t=2}^{T} \mathfrak{S}_t' G^{-1} \mathfrak{S}_t \tag{28.26}$$

For FMOLS, the natural estimator for G is

$$G = \hat{\omega}_{1.2} \left(\sum_{t=2}^{T} Z_t Z_t' \right) \tag{28.27}$$

The $\mathfrak{s}_t$ and G may be defined analogously to least squares for CCR using the transformed data. For DOLS $\mathfrak{s}_t$ is defined for the subset of original regressors Z_t, and G may be computed using the method employed in computing the original coefficient standard errors.

The distribution of L_c is nonstandard and depends on $m_2 = \max(n - p_2, 0)$, the number of cointegrating regressors less the number of deterministic trend regressors excluded from the cointegrating equation, and p the number of trending regressors in the system. Hansen (1992) has tabulated simulation results and provided polynomial functions allowing for computation of p-values for various values of m_2 and p. When computing p-values, EViews ignores the presence of user-specified deterministic regressors in your equation.

In contrast to the residual based cointegration tests, Hansen's test does rely on estimates from the original equation. We continue our illustration by considering an equation estimated on the consumption data using a constant and trend, FMOLS with a Quadratic Spectral kernel, Andrews automatic bandwidth selection, and no d.f. correction for the long-run variance and coefficient covariance estimates. The equation estimates are given by:

Dependent Variable: LC
Method: Fully Modified Least Squares (FMOLS)
Date: 08/11/09 Time: 13:45
Sample (adjusted): 1947Q2 1989Q3
Included observations: 170 after adjustments
Cointegrating equation deterministics: C @TREND
Long-run covariance estimate (Quadratic-Spectral kernel, Andrews bandwidth = 10.9793)
No d.f. adjustment for standard errors & covariance

Variable	Coefficient	Std. Error	t-Statistic	Prob.
LY	0.651766	0.057711	11.29361	0.0000
C	220.1345	37.89636	5.808855	0.0000
@TREND	0.289900	0.049542	5.851627	0.0000

R-squared	0.999098	Mean dependent var	720.5078
Adjusted R-squared	0.999087	S.D. dependent var	41.74069
S.E. of regression	1.261046	Sum squared resid	265.5695
Durbin-Watson stat	0.514132	Long-run variance	8.223497

There are no options for the Hansen test so you may simply click on **View/Cointegration Tests...**, select **Hansen Instability** in the dropdown menu, then click on **OK**.

Cointegration Test - Hansen Parameter Instability
Date: 08/11/09 Time: 13:48
Equation: EQ_19_3_31
Series: LC LY
Null hypothesis: Series are cointegrated
Cointegrating equation deterministics: C @TREND
No d.f. adjustment for score variance

Lc statistic	Stochastic Trends (m)	Deterministic Trends (k)	Excluded Trends (p2)	Prob.*
0.575537	1	1	0	0.0641

*Hansen (1992b) Lc(m2=1, k=1) p-values, where m2=m-p2 is the number of stochastic trends in the asymptotic distribution

The top portion of the output describes the test hypothesis, the deterministic regressors, and any relevant information about the construction of the score variances. In this case, we see that the original equation had both C and @TREND as deterministic regressors, and that the score variance is based on the usual FMOLS variance with no d.f. correction.

The results are displayed below. The test statistic value of 0.5755 is presented in the first column. The next three columns describe the trends that determine the asymptotic distribution. Here there is a single stochastic regressor (LY) and one deterministic trend (@TREND) in the cointegrating equation, and no additional trends in the regressors equations. Lastly, we see from the final column that the Hansen test does not reject the null hypothesis that the series

are cointegrated at conventional levels, though the relatively low p-value are cause for some concern, given the Engle-Granger and Phillips-Ouliaris results.

Park's Added Variables Test

Park's $H(p, q)$ test is an added variable test. The test is computed by testing for the significance of spurious time trends in a cointegrating equation estimated using one of the methods described above.

Suppose we estimate equation Equation (28.1) where, to simplify, we let D_{1t} consist solely of powers of trend up to order p. Then the Park test estimates the spurious regression model including from $p+1$ to q spurious powers of trend

$$y_t = X_t'\beta + \sum_{s=0}^{p} t^s \gamma_s + \sum_{s=p+1}^{q} t^s \gamma_s + u_{1t} \tag{28.28}$$

and tests for the joint significance of the coefficients $(\gamma_{p+1}, \ldots, \gamma_q)$. Under the null hypothesis of cointegration, the spurious trend coefficients should be insignificant since the residual is stationary, while under the alternative, the spurious trend terms will mimic the remaining stochastic trend in the residual. Note that unless you wish to treat the constant as one of your spurious regressors, it should be included in the original equation specification.

Since the additional variables are simply deterministic regressors, we may apply a joint Wald test of significance to $(\gamma_{p+1}, \ldots, \gamma_q)$. Under the maintained hypothesis that the original specification of the cointegrating equation is correct, the resulting test statistic is asymptotically χ^2_{q-p}.

While one could estimate an equation with spurious trends and then to test for their significance using a Wald test, EViews offers a view which performs these steps for you. First estimate an equation where you include all trends that are assumed to be in the cointegrating equation. Next, select **View/Cointegration Test...** and choose **Park Added Variables** in the dropdown menu. The dialog will change to allow you to specify the spurious trends.

There are two parts to the dialog. The dropdown menu allows you to specify a trend polynomial. By default, the dropdown will be set to two orders higher than the trend order in the original equation. In our example equation which includes a linear trend, the default setting will include quadratic and cubic trend terms in the test equation and test for the significance of the two coefficients. You may use the edit field to enter non power-of-trend deterministic regressors.

We will use the default settings to perform a Park test on the FMOLS linear trend consumption equation considered previously. The results are presented in two parts: the test specification and test results are displayed at the top of the output, and the results for the test equation (not depicted) are displayed at the bottom:

Cointegration Test - Park Added Variables
Date: 08/11/09 Time: 13:49
Equation: EQ_19_3_31
Series: LC LY
Null hypothesis: Series are cointegrated
Original trend specification: Linear trend
Added trends: Powers of trend up to 3
Added deterministics to test: @TREND^2 (@TREND/170)^3

	Value	df	Probability
Chi-square	12.72578	2	0.0017

The null hypothesis is that the series are cointegrated. The original specification includes a constant and linear trend and the test equation will include up to a cubic trend. The Park test evaluates the statistical significance of the @TREND^2 and the (@TREND/170)^3 terms using a conventional Wald test. (You may notice that the latter cubic trend term—and any higher order trends that you may include—uses the trend scaled by the number of observations in the sample.)

The test results reject the null hypothesis of cointegration, in direct contrast to the results for the Engle-Granger, Phillips-Ouliarias, and Hansen tests (though the latter, which also tests the null of cointegration, is borderline). Note however, adding a quadratic trend to the original equation and then testing for cointegration yields results that, for all four tests, point to cointegration between LC and LY.

Working with an Equation

Once you estimate your equation, EViews offers a variety of views and procedures for examining the properties of the equation, testing, forecasting, and generating new data. For the

most part, these views and procedures are a subset of those available in other estimation settings such as least squares estimation. (The one new view, for cointegration testing, is described in depth in "Testing for Cointegration," beginning on page 308.) In some cases there have been modifications to account for the nature of cointegrating regression.

Views

For the most part, the views of a cointegrating equation require little discussion. For example, the **Representations** view offers text descriptions of the estimated cointegrating equation, the **Covariance Matrix** displays the coefficient covariance, and the **Residual Diagnostics** (**Correlogram - Q-statistics**, **Correlogram Squared Residuals**, **Histogram - Normality Test**) offer statistics based on pooled residuals. That said, a few comments about the construction of these views are in order.

First, the **Representations** and **Covariance Matrix** views of an equation only show results for the cointegrating equation and the long-run coefficients. In particular, the short-run dynamics included in a DOLS equation are not incorporated into the equation. Similarly, **Coefficient Diagnostics** and **Gradients** views do not include any of the short-run coefficients.

Second, the computation of the residuals used in the **Actual, Fitted, Residual** views and the **Residual Diagnostics** views differs depending on the estimation method. For FMOLS and CCR, the residuals are derived simply by substituting the estimated coefficients into the cointegrating equation and computing the residuals. The values are *not* based on the transformed data. For DOLS, the residuals from the cointegrating equation are adjusted for the estimated short-run dynamics. In all cases, the test statistics results in the **Residual Diagnostics** should only be viewed is illustrative as they are not supported by asymptotic theory.

Note that standardized residuals are simply the residuals divided through by the long-run variance estimate.

The **Gradient** (score) views are based on the moment conditions implied by the particular estimation method. For FMOLS and CCR, these moment conditions *are* based on the transformed data (see Equation (28.25) for the expression for FMOLS scores). For DOLS, these values are simply proportional (-2 times) to the residuals times the regressors.

Procedures

The procs for an equation estimated using cointegrating regression are virtually identical to those found in least squares estimation.

Most of the relevant issues were discussed previously (*e.g.*, construction of residuals and gradients), however you should also note that forecasts constructed using the **Forecast...** procedure and models created using **Make Model** procedure follow the **Representations** view in omitting DOLS short-run dynamics. Furthermore, the forecast standard errors generated by the **Forecast...** proc and from solving models created using the **Make Model...** proc both employ the "S.E. of the regression" reported in the estimation output. This may not be appropriate.

Data Members

The summary statistics results in the bottom of the equation output may be accessed using data member functions (see "Equation Data Members" on page 40 for a list of common data members). For equations estimated using DOLS (with default standard errors), FMOLS, or CCR, EViews computes an estimate of the long-run variance of the residuals. This statistic may be accessed using the `@lrvar` member function, so that if you have an equation named FMOLS,

```
scalar mylrvar = fmols.@lrvar
```

will store the desired value in the scalar MYLRVAR.

References

Engle, R. F., and C. W. J. Granger (1987). "Co-integration and Error Correction: Representation, Estimation, and Testing," *Econometrica*, 55, 251-276.

Hamilton, James D. (1994). *Time Series Analysis*, Princeton: Princeton University Press.

Hansen, Bruce E. (1992a). "Efficient Estimation and Testing of Cointegrating Vectors in the Presence of Deterministic Trends," *Journal of Econometrics*, 53, 87-121.

Hansen, Bruce E. (1992b). "Tests for Parameter Instability in Regressions with I(1) Processes," *Journal of Business and Economic Statistics*, 10, 321-335.

Hayashi, Fumio (2000). *Econometrics*, Princeton: Princeton University Press.

MacKinnon, James G. (1996). "Numerical Distribution Functions for Unit Root and Cointegration Tests," *Journal of Applied Econometrics*, 11, 601-618.

Ogaki, Masao (1993). "Unit Roots in Macroeconometrics: A Survey," *Monetary and Economic Studies*, 11, 131-154.

Park, Joon Y. (1992). "Canonical Cointegrating Regressions," *Econometrica*, 60, 119-143.

Park, Joon Y. and Masao Ogaki (1991). "Inferences in Cointegrated Models Using VAR Prewhitening to Estimate Short-run Dynamics," Rochester Center for Economic Research Working Paper No. 281.

Phillips, Peter C. B. and Bruce E. Hansen (1990). "Statistical Inference in Instrumental Variables Regression with I(1) Processes," *Review of Economics Studies*, 57, 99-125.

Phillips, Peter C. B. and Hyungsik R. Moon (1999). "Linear Regression Limit Theory for Nonstationary Panel Data," *Econometrica*, 67, 1057-1111.

Phillips, Peter C. B. and Mico Loretan (1991). "Estimating Long-run Economic Equilibria," *Review of Economic Studies*, 59, 407-436.

Saikkonen, Pentti (1992). "Estimation and Testing of Cointegrated Systems by an Autoregressive Approximation," *Econometric Theory*, 8, 1-27.

Stock, James H. (1994). "Unit Roots, Structural Breaks and Trends," Chapter 46 in *Handbook of Econometrics, Volume 4*, R. F. Engle & D. McFadden (*eds.*), 2739-2841, Amsterdam: Elsevier Science Publishers B.V.

Stock, James H. and Mark Watson (1993). "A Simple Estimator Of Cointegrating Vectors In Higher Order Integrated Systems," *Econometrica*, 61, 783-820.

Chapter 29. Autoregressive Distributed Lag (ARDL) Models

EViews offers powerful time-saving tools for estimating and examining the properties of Autoregressive Distributed Lag (ARDL) models. ARDLs are standard least squares regressions that include lags of both the dependent variable and explanatory variables as regressors (Greene, 2008). Although ARDL models have been used in econometrics for decades, they have gained popularity in recent years as a method of examining cointegrating relationships between variables through the work of Pesaran and Shin (1998, PS(1998)) and Pesaran, Shin and Smith (2001, PSS(2001)).

While it is possible to use a standard least squares procedure to estimate an ARDL, the specialized ARDL estimator in EViews offers a number of useful features including model selection and the computation of post-estimation diagnostics.

Background

Specification

ARDL models are linear time series models in which both the dependent and independent variables are related not only contemporaneously, but across historical (lagged) values as well. In particular, if y_t is the dependent variable and $x_1, \ldots, x_k$ are k explanatory variables, a general ARDL$(p, q_1, \ldots, q_k)$ model is given by:

$$y_t = a_0 + a_1 t + \sum_{i=1}^{p} \psi_i y_{t-i} + \sum_{j=1}^{k} \sum_{l_j=0}^{q_j} \beta_{j, l_j} x_{j, t-l_j} + \epsilon_t \tag{29.1}$$

where ϵ_t are the usual innovations, a_0 is a constant term, and a_1, ψ_i, and β_{j, l_j} are respectively the coefficients associated with a linear trend, lags of y_t, and lags of the k regressors $x_{j, t}$ for $j = 1, \ldots, k$. Alternatively, let L denote the usual lag operator and define $\psi(L)$ and $\beta_j(L)$ as the lag polynomials:

$$\psi(L) = 1 - \sum_{i=1}^{p} \psi_i L^i \text{ and } \beta_j(L) = 1 - \sum_{l_j=1}^{q_j} \beta_{j, l_j} L^{l_j}$$

Equation (29.1) above can then be rewritten as:

$$\psi(L) y_t = a_0 + a_1 t + \sum_{j=1}^{k} \beta_j(L) x_{j, t} + \epsilon_t \tag{29.2}$$

Following this general formulation, three alternative representations can be made. While all three can be used for parameter estimation, the first is typically used for intertemporal dynamic estimation, the second for post-estimation derivation of the long-run (equilibrium)

relationship, while the third is a reduction of (29.1) to the conditional error correction (CEC) representation in the PSS(2001) bounds test. All three alternative representations make use of the Beveridge-Nelson decomposition.

Representation 1: Intertemporal Dynamics Regression

The typical starting point for most ARDL applications is the estimation of intertemporal dynamics. In this form, one is interested in estimating the relationship between y_t on both its own lags as well as the contemporaneous and lagged values of the k regressors $x_{j,t}$. This in fact the basis of the ARDL model studied in PS(1998). In particular, we cast (29.1) into the following representation:

$$\begin{aligned} y_t &= a_0 + a_1 t + \sum_{i=1}^{p} \psi_i y_{t-i} + \sum_{j=1}^{k} \beta_j(L) x_{j,t} + \epsilon_t \qquad (29.3) \\ &= a_0 + a_1 t + \sum_{i=1}^{p} \psi_i y_{t-i} + \sum_{j=1}^{k} (\beta_j(1) + (1-L)\tilde{\beta}_j(L)) x_{j,t} + \epsilon_t \\ &= a_0 + a_1 t + \sum_{i=1}^{p} \psi_i y_{t-i} + \sum_{j=1}^{k} \beta_j(1) x_{j,t} + \sum_{j=1}^{k} \tilde{\beta}_j(L) \Delta x_{j,t} + \epsilon_t \end{aligned}$$

where we have used the Beveridge-Nelson result to decompose $\beta_j(L)$ into $\beta_j(L) = \beta_j(1) + (1-L)\tilde{\beta}_j(L)$. Since this equation does not solve for y_t, it is typically interpreted as a regression for the intertemporal dynamics of the model.

Representation 2: Post-Regression Derivation of Long-Run Dynamics

The second representation is in essence an attempt to derive the long-run relationship between y_t and the k regressors. As such, the representation solves for y_t in terms of $x_{j,t}$.

$$\begin{aligned} y_t &= \psi^{-1}(1)\left(a^*_0 + a_1 t + \sum_{j=1}^{k} \beta_j(1) x_{j,t} + \sum_{j=1}^{k} \tilde{\beta}^*_j(L) \Delta x_{j,t} + \epsilon^*_t \right) \qquad (29.4) \\ &= \alpha_0 + \alpha_1 t + \sum_{j=1}^{k} \theta_j(1) x_{j,t} + \sum_{j=1}^{k} \tilde{\theta}_j(L) \ \Delta x_{j,t} + \xi_t \end{aligned}$$

where

$$\begin{aligned} a^*_0 &= a_0 - \tilde{\psi}(L)\psi^{-1}(L) a_1 \\ &= a_0 - \tilde{\psi}(1)\psi^{-1}(1) a_1 \\ \tilde{\beta}^*_j(L) &= \tilde{\beta}_j(L) - \tilde{\psi}(L)\psi^{-1}(L)\beta_j(L) \\ \epsilon^*_t &= \epsilon_t - \tilde{\psi}(L)\psi^{-1}(L)\Delta\epsilon_t \end{aligned}$$

and

$$
\begin{aligned}
\alpha_0 &= \psi^{-1}(a_0 - \tilde{\psi}(1)\alpha_1) \\
\alpha_1 &= \psi^{-1}(1)\, a_1 \\
\theta_j(1) &= \psi^{-1}(1)\beta_j(1) \\
\tilde{\theta}_j(L) &= \psi^{-1}(1)(\tilde{\beta}_j(L) - \tilde{\psi}(L)\psi^{-1}(L)\beta_j(L)) \\
\xi_t &= (\psi^{-1}(1)\epsilon_t - \tilde{\psi}(L)\tilde{\psi}^{-1}(L)\alpha_1 \Delta\epsilon_t)
\end{aligned}
$$

where we have again used the Beveridge-Nelson decomposition to express $\psi(L) = \psi(1) + (1 - L)\tilde{\psi}(L)$. Furthermore, from (29.1) we are typically interested in the long-run (trend) parameters captured by α_1 and $\theta_j(1)$, for $j = 1, \ldots, k$. In fact, given the one-to-one correspondence between the parameter estimates obtained in (29.1) and (29.5), it is possible to derive estimates of the long-run parameters post-estimation. In particular, if $\hat{a}_1, \hat{b}_{0,1}, \ldots, \hat{b}_{0,p}, \hat{b}_1, \ldots, \hat{b}_k$ denote the relevant subset of estimated coefficients from the regression model corresponding to the model in (29.4), in particular,

$$
y_t = a_0 + a_1 t + \sum_{i=1}^{p} b_{0,i} y_{t-i} + \sum_{j=1}^{k} b_j \; x_{j,t} + \sum_{j=1}^{k} \sum_{l_j=1}^{q_j - 1} c_{j,l_j} \Delta x_{j,t-l_j} + \epsilon_t
$$

then, a post-regression estimate of the long-run parameters is derived as follows:

$$
\begin{aligned}
\hat{\alpha}_1 &= \frac{\hat{a}_1}{1 - \sum_{i=1}^{p} \hat{b}_{0,i}} \\
\hat{\theta}_j(1) &= \frac{\hat{b}_j}{1 - \sum_{i=1}^{p} \hat{b}_{0,i}}
\end{aligned}
\tag{29.5}
$$

Representation 3: Conditional Error Correction Form and the Bounds Test

The final representation is arguably the most interesting and one that typically receives the most attention in applied work. The objective here is to test for cointegration by reducing a typical vector autoregression framework to its corresponding conditional error correction (CEC) form. This CEC model is in fact an ARDL model with a one-to-one correspondence with the model in (29.1).

In particular, Equation (29.1) may be re-written as:

$$\Delta y_t = a_0 + a_1 t - \psi(1) y_{t-1} + \sum_{j=1}^{k} \beta_j(1) x_{j,\,t-1} \tag{29.6}$$
$$+ \left(\tilde{\psi}^*(L) \Delta y_{t-1} + \sum_{j=1}^{k} \tilde{\beta}_j(L) \Delta x_{j,\,t-1} \right) + \sum_{j=1}^{k} \beta_j(L) \Delta x_{j,\,t} + \epsilon_t$$

where we have used the convention

$$\psi^*(L) = \sum_{i=1}^{p} \psi_i L^i = \left(\sum_{i=1}^{p} \psi_i L^{i-1} \right) L = (\psi^*(1) + (1-L)\tilde{\psi}^*(L)) L$$

relying once again on the Beveridge-Nelson decomposition.

Equation (29.7) above is the CEC form derived from the ARDL model in (29.1). Rewriting this equation as:

$$\Delta y_t = a_0 + a_1 t - \psi(1) \left(y_{t-1} - \sum_{j=1}^{k} \beta_j(1) x_{j,\,t-1} \right) \tag{29.7}$$
$$+ \left(\tilde{\psi}^*(L) \Delta y_{t-1} + \sum_{j=1}^{k} \tilde{\beta}_j(L) \Delta x_{j,\,t-1} \right) + \sum_{j=1}^{k} \beta_j(L) \Delta x_{j,\,t} + \epsilon_t$$
$$= a_0 + a_1 t - \psi(1) EC_{t-1} + \left(\tilde{\psi}^*(L) \Delta y_{t-1} + \sum_{j=1}^{k} \tilde{\beta}_j(L) \Delta x_{j,\,t-1} \right)$$
$$+ \sum_{j=1}^{k} \beta_j(L) \Delta x_{j,\,t} + \epsilon_t$$

it is readily verified that the error correction term, typically denoted as EC_t, is also the cointegrating relationship when y_t and $x_{1,\,t}, \ldots, x_{k,\,t}$ are cointegrated. PSS(2001) demonstrate that (29.7) is in fact (abstracting from differing lag values) the CEC of the VAR(p) model:

$$\Phi(L)(z_t - \mu - \gamma t) = \epsilon_t$$

where z_t is the (k +1)-vector $(y_t, x_{1,\,t}, \ldots, x_{k,\,t})^T$ and μ and γ are respectively the (k +1)-vectors of intercept and trend coefficients, and $\Phi(L) = I_{k+1} - \sum_{i=1} \Phi_i L^i$ is the (k + 1) square matrix lag polynomial.

Traditionally, the cointegration tests of Engle-Granger (1987), Phillips and Ouliaris (1990), Park (1990), or Johansen (1991; 1995), typically require all variables in the VAR to be I(1). This clearly requires a battery pre-testing for the presence of a unit root in each of the variables under consideration, and is subject to misclassification. In contrast, PSS(2001) propose a test for cointegration that is robust to whether variables of interest are I(0), I(1), or mutu-

ally cointegrated. In this regard, PSS(2001) discuss a bounds test for cointegration as a test on parameter significance in the cointegrating relationship of the CEC model (27.7). In other words, the test is a standard F. or Wald test for the following null and alternative hypotheses:

$$\begin{aligned} H_0&: (\psi(1) \cap \{\beta_j(1)\}_{j=1}^{k}) = 0 \\ H_A&: (\psi(1) \cap \{\beta_j(1)\}_{j=1}^{k}) \neq 0 \end{aligned} \tag{29.8}$$

Once the test statistic is computed, it is compared to two asymptotic critical values corresponding to polar cases of all variables being purely I(0) or purely I(1). When the test statistic is below the lower critical value, one fails to reject the null and concludes that cointegration is not possible. In contrast, when the test statistic is above the upper critical value, one rejects the null and concludes that cointegration is indeed possible. In either of these two cases, knowledge of the cointegrating rank is not necessary.

Alternatively, should the test statistic fall between the lower and upper critical values, testing is inconclusive, and knowledge of the cointegrating rank is required to proceed further.

Here it is also important to highlight that PSS(2001) offer five alternative interpretations of the CEC model (29.7), distinguished by whether deterministic terms integrate into the error correction term. When deterministic terms contribute to the error correction term, they are implicitly projected onto the span of the cointegrating vector. This implies that a_0 and a_1 in (29.7) must be restricted. Below are summaries of the regression (REG) models, for each of the five interpretations along with the appropriate cointegrating relationship EC_t and the bounds test null-hypothesis H_0.

Case 1:

$$\begin{aligned} \Delta y_t &= b_0 y_{t-1} + \sum_{j=1}^{k} b_j x_{j,t-1} \\ &+ \sum_{i=1}^{p-1} c_{0,i} \Delta y_{t-i} + \sum_{j=1}^{k} \sum_{l_j=1}^{q_j-1} c_{j,l_j} \Delta x_{j,t-l_j} + \sum_{j=1}^{k} d_j \Delta x_{j,t} + \epsilon_t \\ EC_t &= y_t - \sum_{j=1}^{k} \frac{b_j}{b_0} x_{j,t} \\ H_0&: b_0 = b_j = 0, \forall j \end{aligned} \tag{29.9}$$

Case 2:

$$\begin{aligned}\Delta y_t &= a_0 + b_0 y_{t-1} + \sum_{j=1}^{k} b_j x_{j,\,t-1} \\ &+ \sum_{i=1}^{p-1} c_{0,\,i} \Delta y_{t-i} + \sum_{j=1}^{k} \sum_{l_j=1}^{q_j-1} c_{j,\,l_j} \Delta x_{j,\,t-l_j} + \sum_{j=1}^{k} d_j \Delta x_{j,\,t} + \epsilon_t \\ EC_t &= y_t - \sum_{j=1}^{k} \frac{b_j}{b_0} x_{j,\,t} - \frac{a_0}{b_0} \\ H_0\colon\ b_0 &= b_j = a_0 = 0,\ \forall j\end{aligned} \tag{29.10}$$

Case 3:

$$\begin{aligned}\Delta y_t &= a_0 + b_0 y_{t-1} + \sum_{j=1}^{k} b_j x_{j,\,t-1} \\ &+ \sum_{i=1}^{p-1} c_{0,\,i} \Delta y_{t-i} + \sum_{j=1}^{k} \sum_{l_j=1}^{q_j-1} c_{j,\,l_j} \Delta x_{j,\,t-l_j} + \sum_{j=1}^{k} d_j \Delta x_{j,\,t} + \epsilon_t \\ EC_t &= y_t - \sum_{j=1}^{k} \frac{b_j}{b_0} x_{j,\,t} \\ H_0\colon\ b_0 &= b_j = 0,\ \forall j\end{aligned} \tag{29.11}$$

Case 4:

$$\begin{aligned}\Delta y_t &= a_0 + b_0 y_{t-1} + \sum_{j=1}^{k} b_j x_{j,\,t-1} \\ &+ \sum_{i=1}^{p-1} c_{0,\,i} \Delta y_{t-i} + \sum_{j=1}^{k} \sum_{l_j=1}^{q_j-1} c_{j,\,l_j} \Delta x_{j,\,t-l_j} + \sum_{j=1}^{k} d_j \Delta x_{j,\,t} + \epsilon_t \\ EC_t &= y_t - \sum_{j=1}^{k} \frac{b_j}{b_0} x_{j,\,t} - \frac{a_1}{b_0} t \\ H_0\colon\ b_0 &= b_j = a_1 = 0,\ \forall j\end{aligned} \tag{29.12}$$

Case 5:

$$\Delta y_t = a_0 + a_1 t + b_0 y_{t-1} + \sum_{j=1}^{k} b_j x_{j,\,t-1} \tag{29.13}$$

$$+ \sum_{i=1}^{p-1} c_{0,\,i} \Delta y_{t-i} + \sum_{j=1}^{k} \sum_{l_j=1}^{q_j-1} c_{j,\,l_j} \Delta x_{j,\,t-l_j} + \sum_{j=1}^{k} d_j \Delta x_{j,\,t} + \epsilon_t$$

$$EC_t = y_t - \sum_{j=1}^{k} \frac{b_j}{b_0} x_{j,\,t} - \frac{a_1}{b_0} t$$

$$H_0\colon\ b_0 = b_j = 0,\ \forall j$$

Estimating ARDL Models in EViews

Since ARDL models are least squares regressions using lags of the dependent and independent variables as regressors, they can be estimated in EViews using an equation object with the Least Squares estimation method.

However, EViews also offers a specialized estimator for handling ARDL models. This estimator offers built-in lag-length selection methods, as well as post-estimation views. To estimate an ARDL model using the ARDL estimator, open the equation dialog by selecting **Quick/Estimate Equation…**, or by selecting **Object/New Object…/Equation** and then selecting **ARDL** from the **Method** dropdown menu. EViews will then display the ARDL estimation dialog:

The **Specification** tab allows you to specify the variables used in the regression, and whether to let EViews automatically detect the appropriate number of lags for each variable.

To begin, enter the name of the dependent variable, followed by a space delimited list of dynamic regressors (*i.e.*, variables which will have lag terms in the model) in the **Dynamic Specification** edit box. You may then select whether you wish EViews to automatically select the number of lags for all variables by selecting the Automatic Selection radio button, fixing the independent variable and the regressors to a uniform fixed length by selecting the Fixed radio buttons, or by taking full control of granularity and specifying a specific lag for each of the independent and regressors variables. The latter can be specified via command in the Dynamic Specification edit box by replacing each variable by the Fixed Lag command @FL(VARIABLE, LAG). For instance, if the variable x_t should possess 3 lags, then one would specify this by writing @FL(x_t, 3). One can do this for all variables in order to estimate a specific structure, or specify some variables using the @FL command, and others without. In the latter case, if the Automatic Selection radio button is selected, EViews will fix the lags of the variables specified with @FL, and automatically select the lags for the variables which were not specified using the @FL function. Alternatively, if the Fixed radio button is selected, any variables not specified with @FL will have the specified fixed number.

If you choose automatic selection, you must then select the maximum number of lags to test for the dependent variable and regressors using the **Max lags** dropdowns. If you select to use a fixed number of lags, the same dropdowns can be used to select the number of lags for the dependent variable and regressors. Note that when using fixed lags for regressors, each regressor will be given the same number of lags.

The **Fixed regressors** area lets you specify any fixed/static variables (regressors without lags). The **Trend specification** dropdown may be used to specify which of the five cases for the trend and intercept should be included. Any other static regressors can be specified by entering their name in the **List of fixed regressors** box.

The Options tab allows you to specify the type of model selection to be used if you chose automatic selection on the **Specification** tab. You may choose between the Akaike Information Criterion (AIC), Schwarz Criterion (SC), Hannan-Quinn Criterion (HQ), or the Adjusted R-squared.

You may also select the type of covariance matrix to use in the final estimates, using the **Coefficient covariance matrix** dropdown. Note that this selection does not affect the model selection criteria.

ARDL Post-Estimation Views and Procedures

Since ARDL models are estimated by simple least squares, all of the views and procedures available to equation objects estimated by least squares are also available for ARDL models. In addition, there are a few ARDL specific views.

The **Model Selection Summary** item on the **View** menu allows you to view either a **Criteria Graph** or a **Criteria Table**. The graph shows the model selection value for the twenty "best" models. If you use either the Akaike Information Criterion (AIC), the Schwarz Criterion (BIC), or the Hannan-Quinn (HQ) criterion, the graph will show the twenty models with the lowest criterion value. If you choose the Adjusted R-squared as the model selection criteria, the graph will show the twenty models with the highest Adjusted R-squared. The table form of the view shows the log-likelihood value, the AIC, BIC and HQ values, and the Adjusted R-squareds of the top twenty models in tabular form.

The **View/Coefficient Diagnostics** menu offers the new item **Long Run Form and Bounds Test.** Every ARDL model is associated with a CEC model. This view displays a table of least squares estimates corresponding to this CEC regression. Note that the lag of the dependent variable in this regression will always be suffixed by a single asterisk while some other variables will be suffixed by a double asterisk. As summarized in notes below the regression output, the single asterisk indicates that the p-value associated with the relevant variable is incompatible with the t-Bounds distribution in Theorem 3.2 in PSS(2001). Moreover, any variables suffixed by a double asterisk indicates a dynamic regressor with an optimal lag of zero. As such, EViews does not include lags and differences of such variables, but estimates them contemporaneously. Accordingly, such variables should be reinterpreted in the context of the decomposition $z_t = z_{t-1} + \Delta z_t$ so that they can be included in the EC_{t-1} term which arises in each of the CEC regressions. In particular, least squares estimates of coefficients associated with such variables are simultaneously estimates of the coefficients associated with z_{t-1} as well as Δz_t. When this is the case, EViews augments the table of

regression estimates with a note that such variables should be interpreted as Z = Z(-1) + D(Z).

There are also several tables provided in this view. The first table is titled *Levels Equation* and displays the estimates of long-run variables, their standard errors computed using the delta method as in PS(98), their t-statistics, as well as the appropriate p-values. Moreover, just below this table is a line starting with EC. This expression lists the name of the dependent variable minus an expression enclosed in brackets. This is the long run, otherwise known as the error correction equation.

Below the table of long run coefficient estimates are two additional tables, respectively titled as the F-Bounds Test and the t-Bounds Test. These tables respectively display the the the F- and t- statistics along with their associated I(0) (lower) and I(1) (upper) critical value bounds for the null hypotheses of no levels relationship between the dependent variable and the regressors in the CEC model. The critical values are provided for significance levels 10%, 5%, 2.5%, and 1%, respectively. The t-Bounds test in particular is a parameter significance test on the lagged value of the dependent variable. Since the distribution of this test is non-standard, the p-value provided in the regression output of the CEC regression is not compatible with this distribution, although the t-statistic is valid. Accordingly, any inference must be conducted using the t-Bounds test critical values provided.

We also mention here that the F- critical value tables now present the critical values computed under an asymptotic regime (sample size equal to 1000) and referenced from PSS(2001), in addition to providing cirtical values for finite sample regimes (sample sizes running from 30 to 80 in increments of 5) and referenced from Narayan (2005).

Another view that is offered after estimation is **View/ Coefficient Diagnostics/Error Correction Form**. In this view, an error correction model which estimates the speed of adjustment to equilibrium in a cointegrating relationship. Here, the error correction term derived as the Levels Equation earlier, is included among the regressors and is denoted as *CointEq*. The coefficient associated with this regressor is typically the speed of adjustment to equilibrium in every period. If variables are indeed cointegrated, we typically expect this coefficient to be negative and highly significant. Here as well we find the the F-Bounds Test and the t-Bounds Test tables below the regression output. While the the F-Bounds Test will not have changed from the **Long Run Form and Bounds Test** view, the t-Bounds Test here reflects the t- statistic associated with the *CointEq* regressor. Again, since the distribution of this test is non-standard, the p-value provided in the regression output is not compatible with this distribution and any inference must be conducted using the the t-Bounds test critical values provided.

Issues with ARDL Model Selection

The ARDL model selection process will use the same sample for each estimation. Since the selection is over the number of lags, this means that observations will be dropped from each

estimation based on the maximum number of lags in the selection procedure. However the final estimation output will use all observations available for the selected model. Consequently, unless the selected model happens to be the model with the maximum number of lags, the final estimation will have more observations than when that model was estimated during selection, and will have different estimated results.

An Example

Greene (2008, page 685) uses an ARDL model on data from a number of quarterly US macroeconomic variables between 1950 and 2000. In particular, he estimates an ARDL model using the log of real consumption as the dependent variable, and the log of real GDP as a single regressor (along with a constant).

We can open the Greene data with the following EViews command:

```
wfopen http://www.stern.nyu.edu/~wgreene/Text/Edition7/TableF5-
    2.txt
```

Next we bring up the ARDL estimation dialog by clicking on **Quick/Estimate Equation** and using the **Method** combo to change the estimation method to **ARDL**.

Following Greene's example, we estimate an ARDL model with the log of real consumption as the dependent variable, and log GDP as the regressor, by entering:

```
log(realcons) log(realgdp)
```

in the **Dynamic Specification** area. We choose to perform **Automatic Selection**, with a maximum of 8 lags (two years) for both the dependent variable and dynamic regressors.

Greene includes a full set of quarterly dummies as fixed regressors, which we can include by choosing **Constant (Level)** as the trend specification, and then adding the expression "@expand(@quarter, @droplast)" in the **Fixed regressors** box.

We do not make any changes to the **Options** tab, leaving all settings at their default value. The results are shown below:

Dependent Variable: LOG(REALCONS)
Method: ARDL
Date: 03/10/15 Time: 23:38
Sample (adjusted): 1951Q2 2000Q4
Included observations: 199 after adjustments
Maximum dependent lags: 8 (Automatic selection)
Model selection method: Akaike info criterion (AIC)
Dynamic regressors (8 lags, automatic): LOG(REALGDP)
Fixed regressors: @EXPAND(@QUARTER, @DROPLAST) C
Number of models evalulated: 72
Selected Model: ARDL(5, 1)
Note: final equation sample is larger than selection sample

Variable	Coefficient	Std. Error	t-Statistic	Prob.*
LOG(REALCONS(-1))	0.854510	0.064428	13.26300	0.0000
LOG(REALCONS(-2))	0.258776	0.082121	3.151153	0.0019
LOG(REALCONS(-3))	-0.156598	0.071521	-2.189542	0.0298
LOG(REALCONS(-4))	-0.194069	0.070465	-2.754106	0.0065
LOG(REALCONS(-5))	0.169457	0.048486	3.494951	0.0006
LOG(REALGDP)	0.547615	0.048246	11.35042	0.0000
LOG(REALGDP(-1))	-0.475684	0.051091	-9.310547	0.0000
@QUARTER=1	-0.000348	0.001176	-0.295813	0.7677
@QUARTER=2	-0.000451	0.001165	-0.386775	0.6994
@QUARTER=3	0.000854	0.001171	0.729123	0.4668
C	-0.058209	0.027842	-2.090705	0.0379

R-squared	0.999873	Mean dependent var	7.902158
Adjusted R-squared	0.999867	S.D. dependent var	0.502623
S.E. of regression	0.005805	Akaike info criterion	-7.406420
Sum squared resid	0.006336	Schwarz criterion	-7.224378
Log likelihood	747.9388	Hannan-Quinn criter.	-7.332743
F-statistic	148407.0	Durbin-Watson stat	1.865392
Prob(F-statistic)	0.000000		

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

The first part of the output gives a summary of the settings used during estimation. Here we see that automatic selection (using the Akaike Information Criterion) was used with a maximum of 8 lags of both the dependent variable and the regressor. Out of the 72 models evaluated, the procedure has selected an ARDL(5,1) model - 5 lags of the dependent variable, LOG(REALCONS), and a single lag (along with the level value) of LOG(REALGDP).

EViews also notes that since the selected model has fewer lags than the maximum, the sample used in the final estimation will not match that used during selection.

The rest of the output is standard least squares output for the selected model. Note that each of the regressors (with the exception of the quarterly dummies) is significant, and that the coefficient on the one period lag of the dependent variable, LOG(REALCONS), is quite high, at 0.85.

To view the relative superiority of the selected model against alternatives, we click on V**iew/ Model Selection Summary/Criteria Graph** to view a graph of the AIC of the top twenty models.

The selected ARDL(5,1) model was only slightly better than an ARDL(5,2) model, which was in turn only slightly better than an ARDL(5,3). It is notable that the top three models all use five lags of the dependent variable.

Rather than using automatic selection to choose the best model, Greene (Example 20.4) analyzes these data with a fixed ARDL(3,3) model. We can replicate this by pressing the **Estimate** button to bring up the **Equation Estimation** dialog again. We change the number of lags on both dependent and regressors to 3, and then select the **Fixed** radio button to switch off automatic selection.

The results of this estimation are:

ARDL Cointegrating Form and Bounds Test
Original dep. variable: LOG(REALCONS)
Selected Model: ARDL(3, 3)
Date: 31/03/17 Time: 11:04
Sample: 1950Q1 2000Q4
Included observations: 201

Cointegrating Form

Variable	Coefficient	Std. Error	t-Statistic	Prob.
C	-0.110221	0.029258	-3.767248	0.0002
LOG(REALCONS(-1))	-0.118944	0.030474	-3.903191	0.0001
LOG(REALGDP(-1))	0.126497	0.032281	3.918624	0.0001
DLOG(REALCONS(-1))	-0.157714	0.069795	-2.259665	0.0250
DLOG(REALCONS(-2))	0.233653	0.068672	3.402444	0.0008
DLOG(REALGDP)	0.565088	0.051953	10.87699	0.0000
DLOG(REALGDP(-1))	0.047706	0.063725	0.748631	0.4550
DLOG(REALGDP(-2))	-0.190243	0.058922	-3.228753	0.0015
@QUARTER=2	4.66E-07	0.001270	0.000367	0.9997
@QUARTER=3	0.001174	0.001263	0.929288	0.3539
@QUARTER=4	0.000259	0.001266	0.204677	0.8380

Bounds Test Null Hypothesis: No cointegrating relationships exist

Test Statistic	Value	Signif.	I(0)	I(1)
F-statistic	10.45256	10%	3.02	3.51
k	1	5%	3.62	4.16
		2.5%	4.18	4.79
		1%	4.94	5.58

EC = LOG(REALCONS) - (1.0635*LOG(REALGDP) + 0.0000
(@QUARTER=2) + 0.0099(@QUARTER=3) + 0.0022
*(@QUARTER=4) -0.9267)

Long Run Coefficients

Variable	Coefficient	Std. Error	t-Statistic	Prob.
LOG(REALGDP)	1.063498	0.007908	134.4805	0.0000
@QUARTER=2	3.92E-06	0.010680	0.000367	0.9997
@QUARTER=3	0.009869	0.010947	0.901557	0.3684
@QUARTER=4	0.002178	0.010645	0.204601	0.8381
C	-0.926656	0.065892	-14.06325	0.0000

The one-period lag on the dependent variable remains high, at 0.72, and again all coefficients are significant (with the exception of the dummies).

We can then examine the long-run coefficients by selecting **View/Coefficient Diagnostics/ Cointegration Form and Bounds Test**.

The long-run coefficients, at the bottom of the output, show that the long-run impact of a change in log(REALGDP) on log(REALCONS) has essentially no lagged-effects. The long-run change is very close to being equal to the initial change (the coefficient is close to one).

In a second example, Example 20.5, Greene examines an ARDL(1,1) model's cointegrating form. To perform this in EViews, we again bring up the **Equation Estimation** dialog and change the number of lags to 1 for both dependent and regressors, remove the quarterly dummies, and then click **OK**.

Following estimation, we click on **View/Coefficient Diagnostics/Cointegration Form and Bounds Test** to bring up the cointegrating relationship view:

ARDL Cointegrating Form and Bounds Test
Original dep. variable: LOG(REALCONS)
Selected Model: ARDL(1, 1)
Date: 31/03/17 Time: 11:05
Sample: 1950Q1 2000Q4
Included observations: 203

Cointegrating Form

Variable	Coefficient	Std. Error	t-Statistic	Prob.
C	-0.085331	0.029285	-2.913823	0.0040
LOG(REALCONS(-1))	-0.095416	0.030589	-3.119291	0.0021
LOG(REALGDP(-1))	0.101173	0.032371	3.125408	0.0020
DLOG(REALGDP)	0.584210	0.051411	11.36351	0.0000

Bounds Test Null Hypothesis: No cointegrating relationships exist

Test Statistic	Value	Signif.	I(0)	I(1)
F-statistic	17.24754	10%	3.02	3.51
k	1	5%	3.62	4.16
		2.5%	4.18	4.79
		1%	4.94	5.58

EC = LOG(REALCONS) - (1.0603*LOG(REALGDP) -0.8943)

Long Run Coefficients

Variable	Coefficient	Std. Error	t-Statistic	Prob.
LOG(REALGDP)	1.060339	0.010630	99.75379	0.0000
C	-0.894307	0.089041	-10.04381	0.0000

References

Banerjee, A., Dolado, J. J., Galbraith, J. W., Hendry, D., et al. (1993). "Co-integration, error correction, and the econometric analysis of non-stationary data." *OUP Catalogue*.

Engle, R. F. and Granger, C. W. (1987). "Co-integration and error correction: representation, estimation, and testing." *Econometrica: Journal of the Econometric Society*, pages 251–276.

Johansen, S. (1991). "Estimation and hypothesis testing of cointegration vectors in gaussian vector autoregressive models." *Econometrica: Journal of the Econometric Society*, pages 1551–1580.

Johansen, S. (1995). "Likelihood-based inference in cointegrated vector autoregressive models." *Oxford University Press on Demand*.

Johansen, S. and Juselius, K. (1990). "Maximum likelihood estimation and inference on cointegration - with applications to the demand for money." *Oxford Bulletin of Economics and Statistics*, 52(2):169–210.

Narayan, P. K. (2005). "The saving and investment nexus for china: evidence from cointegration tests." *Applied Economics*, 37(17):1979–1990.

Park, J. (1990). "Testing for unit roots and cointegration by variable addition." *Advances in Econometrics*, 8(2):107–133.

Perron, P. and Ng, S. (1996). "Useful modifications to some unit root tests with dependent errors and their local asymptotic properties." *The Review of Economic Studies*, 63(3):435–463.

Pesaran, M. H. and Shin, Y. (1998). "An autoregressive distributed-lag modelling approach to cointegration analysis." *Econometric Society Monographs*, 31:371–413.

Pesaran, M. H., Shin, Y., and Smith, R. J. (2001). "Bounds testing approaches to the analysis of level relationships." *Journal of Applied Econometrics*, 16(3):289–326.

Phillips, P. C. and Ouliaris, S. (1990). "Asymptotic properties of residual based tests for cointegration." *Econometrica: Journal of the Econometric Society*, pages 165–193.

Stock, J. H. and Watson, M. W. (1988). "Testing for common trends." *Journal of the American statistical Association*, 83(404):1097–1107.

Chapter 30. Midas Regression

Mixed Data Sampling (MIDAS) regression is an estimation technique which allows for data sampled at different frequencies to be used in the same regression.

More specifically, the MIDAS methodology (Ghysels, Santa-Clara, and Valkanov, (2002) and Gyhsels, Santa-Clara, and Valkanaov (2006), and Andreou, Ghysels, and Kourtellos (2010)) addresses the situation where the dependent variable in the regression is sampled at a lower frequency than one or more of the regressors. The goal of the MIDAS approach is to incorporate the information in the higher frequency data into the lower frequency regression in a parsimonious, yet flexible fashion.

The following discussion describes EViews' easy-to-use tools for single equation MIDAS regression estimation. We begin by offering background on the approach. Next, we describe how to estimate a MIDAS regression in EViews. We conclude with examples.

Background

Standard regression models require the regressor data to follow the same frequency and structure as the dependent variable in the regression. This restriction is not always met in practice—as in economics, where major statistical releases occur on annual, quarterly, monthly and even daily frequencies.

Traditionally, there have been two approaches to estimation in mixed frequency data settings:

- The first approach involves introducing the sum or average of the higher frequency data into the lower frequency regression. This approach adds a single coefficient for each high frequency variable, implicitly applying equal weighting to each value in the sum.
- Alternately, the individual components of the higher frequency data may be added to the regression, allowing for a separate coefficient for each high frequency component. For example, in estimating an annual regression with monthly high frequency regressors, one could add each of the monthly components as a regressor. Note that this approach adds a large number of coefficients to the regression.

MIDAS estimation occupies the middle ground between these approaches, allowing for non-equal weights but reducing the number of coefficients by fitting functions to the parameters of the higher frequency data. Thus, MIDAS offers an approach to mixed frequency estimation featuring a flexible, parsimonious parameterization of the response of the lower frequency dependent variable to the higher frequency data.

Specifically, the model under consideration is:

$$y_t = X_t'\beta + f(\{X_{t/S}^H\}, \theta, \lambda) + \epsilon_t \tag{30.1}$$

where

- y_t is the dependent variable, sampled at a low frequency, at date t,
- X_t is the set of regressors sampled at the same low frequency as y_t,
- $\{X_{t/S}^H\}$ is a set of regressors sampled at a higher frequency with S values for each low frequency value. Note that $\{X_{t/S}^H\}$ is not restricted to the S values associated with the current t as it may include values corresponding to lagged low frequency values.
- f is a function describing the effect of the higher frequency data in the lower frequency regression
- β, λ, and θ vectors of parameters to be estimated.

The *individual coefficients* approach adds each of the higher frequency components as a regressor in the lower frequency regression. In the simple case where we only include high frequency data corresponding to the current low frequency observation, we have:

$$y_t = X_t'\beta + \sum_{\tau=0}^{S-1} X_{(t-\tau)/S}^H{}'\theta_\tau + \epsilon_t \tag{30.2}$$

where $X_{(t-\tau)/S}^H$ are the data τ high frequency periods prior to t (we will refer to these data as the τ-th high frequency lag at t). Notice that this approach estimates a distinct θ for each of the S high frequency lag regressors.

Alternately, the simple *aggregation* approach adds an equally weighted sum (or average) of the high frequency data as a regressor in the low frequency regression:

$$y_t = X_t'\beta + \left(\sum_{\tau=0}^{S-1} X_{(t-\tau)/S}^H\right)'\lambda + \epsilon_t \tag{30.3}$$

The approach estimates a single λ associated with the new regressor. Viewed differently, the aggregation approach may be thought of as one in which the component higher frequency lags all enter the low frequency regression with a common coefficient λ.

For a quarterly regression with a higher frequency monthly series, there are three months in each quarter so the individual coefficients approach adds three regressors to the low frequency regression. The first regressor contains values for the first month in the corresponding quarter (January, April, July, or October), the second regressor has values for the second month in the corresponding quarter (February, May, August, or November), and the third regressor contains values for the third month in the relevant quarter (March, June, September, December).

The aggregation approach adds the single regressor containing the sum of the monthly values over the corresponding quarter. For first quarter observations, the regressor will contain the sum of the higher frequency January, February, and March monthly values for that quarter. Similarly, the regressor will contain the sum of the October, November, and December values in fourth quarter observations.

We may think of these two approaches as polar extremes. The individual coefficients approach offers the greatest flexibility but requires large numbers of coefficients. The aggregation approach is parsimonious, but places quite significant equal weighting restrictions on the lagged high frequency data.

In contrast, MIDAS estimation offers several different weighting functions which occupy the middle ground between the unrestricted and the equally weighted aggregation approaches. The MIDAS weighting functions reduce the number of parameters in the model by placing restrictions on the effects of high frequency variables at various lags.

Step Weighting

The simplest weighting method employs the step function:

$$y_t = X_t'\beta + \sum_{\tau=0}^{k-1} X^H_{(t-\tau)/S}{}'\varphi_\tau + \epsilon_t \tag{30.4}$$

where

- k is a chosen number of lagged high frequency periods to use (where k may be less than or greater than S).
- η is a step length
- $\varphi_m = \theta_i$ for $\kappa = \text{int}(m/\eta)$

In this approach, the coefficients on the high frequency data are restricted using a step function, with high frequency lags within a given step sharing values for φ. For example, with $\eta = 3$, the first three lagged high frequency lags $X^H_{(t-\tau)/S}$, $\tau = 0, 1, 2$, employ the same coefficient θ_0, the next three lags use θ_1, and so on up to the maximum lag of k.

Notably, the number of high frequency coefficients in the step weighting model increases with the number of high frequency lags, but in comparison to an individual coefficient approach, the number of coefficients is reduced by a factor of roughly $1/\eta$.

Almon (PDL) Weighting

Almon lag weighting (also called polynomial distributed lag or PDL weighting) is widely used to place restrictions on lag coefficients in autoregressive models, and is a natural candidate for the mixed frequency weighting.

For each high frequency lag up to k, the regression coefficients are modeled as a p dimensional lag polynomial in the MIDAS parameters θ. We may write the resulting restricted regression model as:

$$y_t = X_t'\beta + \sum_{\tau=0}^{k-1} X^H_{(t-\tau)/S}{}'\left(\sum_{j=0}^{p} \tau^j \theta_j\right) + \epsilon_t \tag{30.5}$$

where p is the Almon polynomial order, and the chosen number of lags k may be less than or greater than S.

Importantly, the number of coefficients to be estimated depends on the polynomial order and not the number of high frequency lags. We can see this more clearly after rearranging terms and rewriting the model using a constructed variable:

$$\begin{aligned} y_t &= X_t'\beta + \sum_{i=0}^{p} Z_{i,t}'\theta_i + \epsilon_t \\ Z_{i,t} &= \sum_{\tau=0}^{k-1} \tau^i X^H_{(t-\tau)/S} \end{aligned} \tag{30.6}$$

It is easy to see the distinct coefficient θ_i associated with each of the p sets of constructed variables $Z_{i,t}$.

Exponential Almon Weighting

The normalized exponential Almon weighting approach uses exponential weights and a lag polynomial of degree 2, yielding:

$$y_t = X_t'\beta + \sum_{\tau=0}^{k-1} X^H_{(t-\tau)/S}{}'\left(\frac{\exp(\tau\theta_1 + \tau^2\theta_2)}{\sum_{j=0}^{k}\exp(j\theta_1 + j^2\theta_2)}\right)\lambda + \epsilon_t \tag{30.7}$$

$$y_t = X_t'\beta + \sum_{\tau=0}^{k-1} X^H_{(t-\tau)/S}{}'\left(\frac{\exp(\tau\theta_1 + \tau^2\theta_2)}{\sum_{j=0}^{k}\exp(j\theta_1 + j^2\theta_2)}\right)\lambda + \epsilon_t \tag{30.8}$$

where k is a chosen number of lags, λ is a slope coefficient that is common across lags, and the differential response comes via the exponential weighting function and the lag polynomial which depends on the two MIDAS coefficients θ_1 and θ_2.

In constructed variable form, we have

$$y_t = X_t'\beta + \sum_{i=0}^{k} Z_{i,t}'\lambda + \epsilon_t$$
$$Z_{i,t} = \left(\frac{\exp(i\theta_1 + i^2\theta_2)}{\sum_{j=0}^{k}\exp(j\theta_1 + j^2\theta_2)}\right) X^H_{(t-i)/S} \tag{30.9}$$

Note that this regression model is highly nonlinear in the parameters of the model.

Beta Weighting

Beta weighting was introduced by Ghysels, Santa-Clara and Valkanov and is based on the normalized beta weighting function. The corresponding regression model is given by

$$y_t = X_t'\beta + \sum_{\tau=0}^{k-1} X^H_{t-\tau}{}'\left(\frac{\omega_\tau^{\theta_1-1}(1-\omega_\tau)^{\theta_2-1}}{\sum_{j=0}^{k}\omega_j^{\theta_1-1}(1-\omega_j)^{\theta_2-1}} + \theta_3\right)\lambda + \epsilon_t \tag{30.10}$$

where k is a number of lags, λ is a slope coefficient that is common across lags, and

$$\omega_i = \begin{cases} \delta & i = 0 \\ i/(k-1) & i = 1, \ldots, k-2 \\ 1-\delta & i = k \end{cases} \tag{30.11}$$

where δ is a small number (in practice, approximately equal to $2.22e^{-16}$).

In constructed variable form, we have

$$y_t = X_t'\beta + \sum_{i=0}^{k} Z_{i,t}'\lambda + \epsilon_t$$
$$Z_{i,t} = \left(\frac{\omega_i^{\theta_1-1}(1-\omega_i)^{\theta_2-1}}{\sum_{j=0}^{k}\omega_j^{\theta_1-1}(1-\omega_j)^{\theta_2-1}} + \theta_3\right) X^H_{(t-i)/S} \tag{30.12}$$

The beta function is extremely flexible and can take many shapes, including gradually increasing or decreasing, flat, humped, or U-shaped, depending on the values of the three MIDAS parameters $(\theta_1, \theta_2, \theta_3)$.

In practice the parameters of the beta function are restricted further by imposing $\theta_1 = 1$, $\theta_3 = 0$, or $\theta_1 = 1$ and $\theta_3 = 0$.

- The restriction $\theta_1 = 1$ implies that the *shape* of the weight function depends on a single parameter, exhibiting slow decay when $\theta_2 > 1$ and slow increase when $\theta_2 < 1$.

- The restriction $\theta_3 = 0$ implies that there are zero weights at the high frequency lag endpoints (when $\tau = 0$ and $\tau = k-1$).
- The restriction $\theta_1 = 1$ and $\theta_3 = 0$ imposes both the shape and the zero endpoint weight restrictions.

(Please note that for specifications with a small number of MIDAS lags the zero endpoint restrictions is quite restrictive and may generate significant bias.)

Lastly, while the number of parameters of the beta weighting model is at most 3 so that it does not increase with the number of lags, estimation does involve optimization of a highly non-linear objective.

U-MIDAS

The U-MIDAS technique adds each of the higher frequency components as a regressor in the lower frequency regression. Notably, The U-MIDAS weighting method is simply the individual coefficients technique given by Equation (30.2). In contrast to other methods, there are no restrictions on the implied lag polynomial in the low frequency regression.

While extremely flexible, U-MIDAS requires a estimation of a large number of coefficients. U-MIDAS does not alleviate the issue of requiring a large number of coefficients, but can be used in cases where a small number of lags are required, and is often used for comparative purposes.

Auto/GETS

The Auto/GETS weighting scheme is an extension of U-MIDAS that uses variable selection to reduce down the number of individual coefficients by excluding individual lags. See "Auto-Search / GETS" on page 68 for discussion of the Auto/GETS procedure.

As a complement to the MIDAS variable selection, EViews also offers the ability to test for indicator variable inclusion. See "Indicator Saturation" on page 60 for additional detail on indicator saturation.

MIDAS Estimation in EViews

With built-in tools for working with multi-frequency data and an intrinsic understanding of the relationship between various time series frequencies, EViews offers an ideal platform for MIDAS estimation.

To perform MIDAS estimation in EViews, open the equation dialog by selecting **Quick/Estimate Equation…**, or by selecting **Object/New Object…/Equation** and then selecting **MIDAS** from the **Method** dropdown menu to bring up the MIDAS estimation dialog:

Specification

The **Specification** tab is used to specify the variables of and form of the MIDAS equation and to set the estimation sample.

The **Specification** edit field is used to specify the low frequency dependent variable followed by a list of low frequency regressors from the same page as the dependent variable. The low frequency regressors should include any desired lags of the dependent variable. Note that explicit ARMA terms are not permitted in this estimation method.

The **Higher frequency regressors** edit field is used to specify the higher-frequency regressors. The syntax for these variable is *pagename\seriesname* where *pagename* is the name of the page containing the series, and *seriesname* is the name of the series. Note also that series expressions are allowed, *e.g.* “mypage\log(x)”.

You may specify more than one higher-frequency series, and those series may be of different frequencies from different pages. However, we caution you that using more than one high frequency regressor oftens leads to multicollinearity issues and, in the case of the non-linear weighting, increases the complexity of estimation dramatically. An alternative approach suggested by Andreou, *et al.* (2013) would be to estimate several univariate models and then use forecast combination to produce a final forecast.

Date Timing

When specifying your high frequency variable, care should must be taken to ensure that you refer to the correct observations from the higher frequency page.

To illustrate, let's assume our dependent variable, Y, is quarterly, and our regressor, X, is monthly. We would like to use 4 lags (months) of X to explain each quarter of Y. EViews will use the 4 months up to, and including, the last month of the corresponding quarter. Quarter 1 will thus be explained by March, February, January and December. Quarter 2 will be explained by June, May, April and March.

If you wish to use different sets of months, you can use the lag operator when specifying the regressor. In our example, if we want Quarter 1 to be explained by January, December, November and October, and Quarter 2 to be explained by April, March, February and January, we would specify the regressor as "monthlypage\x(-2)"; *i.e.,* using the second lagged values of X.

Lag Selection

All of the MIDAS estimation methods require a value for k, the number of high frequency lags to be included in the low frequency regression equation.

Just below the **Higher frequency regressors** edit field are radio buttons that control the number of lags. You may provide a fixed number of lags by selecting the appropriate radio button and entering a value, or you can elect to determine the number of lags using minimal sum-of-squared residuals as the selection criterion. If you select the latter radio button, you will prompted to enter a value for the maximum number of lags. Note that automatic selection is only available for the Almon and Step weighting methods.

If you have entered more than one high frequency regressor you may enter a single lag or maximum lag value or you may enter a space delimited list of lags. If you enter a single value, it will be applied to all of the regressors.

As you make your choice, keep in mind that the maximum number of lags and selected lags from automatic selection will apply to all of the high frequency series.

Estimation Options

The **Options** tab of the dialog lets you specify some the MIDAS weighting function along with other estimation options:

The **MIDAS weighting method** dropdown menu controls specification of the MIDAS weighting. By default the **PDL/Almon** weighting method is selected, but **Step**, **Exponential Almon**, **Beta, U-MIDAS,** or **Auto/GETS** may also be chosen:

- If you select the **PDL/Almon** method, you must specify p, the degree of the Almon polynomial.
- If you select **Step**, you must specify the stepsize η.
- If you select **Beta**, you may, if desired, impose restrictions on the shape parameter θ_1, the endpoint parameter θ_3, or both θ_1 and θ_3. You may also specify covariance and optimization settings ("Coefficient Covariance and Optimization" on page 350).
- If you select **Exponential Almon**, you may specify covariance and optimization settings ("Coefficient Covariance and Optimization" on page 350).
- If you select **Auto/GETS**, and click on the **Options** button you will be prompted to specify options for the Auto/GETS selection algorithm, as well as settings for indicator saturation inclusion. See "Variable Selection and Indicator Saturation" on page 351 for extended discussion.

Frequency Conversion

The **Frequency Conversion Options** button produces a secondary dialog that allows you to change the way the different frequencies of the variables are matched. By default, EViews uses the last observation in the higher frequency periods as the 0th lag in the regression. You

can change this to instruct EViews to use the first observation, or to use arbitrary date series from each page to perform the date matching.

Coefficient Covariance and Optimization

Since the **Beta** and **Exponential Almon** weighting methods involve non-linear estimation, selecting either of these methods will enable the **Coefficient Covariance** and **Optimization**-method options:

The **Coefficient covariance** section offers standard EViews covariance settings for nonlinear regression ("Coefficient Covariance" on page 54).

The **Optimization method** dropdown menu offers standard EViews optimization settings for nonlinear regression ("Optimization" on page 55), with the exception of the default **Hybrid Optimization method**. This method is a combination of the **OPG** and **BFGS** methods, where OPG is used for an initial 50 iterations, then BFGS is used until convergence. We have found that the hybrid method often reaches convergence more successfully than OPG or BFGS alone.

For the nonlinear models, you may elect to have EViews obtain starting values, or you may specify your own.

For the exponential Almon method, EViews sets $\theta_1 = -1$ and $\theta_2 = 0$, then runs OLS with those values to obtain the remaining starting values. For beta weighting, EViews sets $\theta_1 = 1$, $\theta_2 = 5$, and $\theta_3 = 0$, then runs OLS to obtain the remaining values. Then, if not

performing shape restricted estimation, EViews updates the starting values by estimating a shape restricted beta weight model.

Variable Selection and Indicator Saturation

If you select **Auto/GETS** in the **MIDAS weights** dropdown, EViews will display an Auto/GETS **Options** button.

Clicking on this button displays a dialog showing options for variable selection and indicator saturation:

There are three sections in this dialog:

- In the **Model selection** section the **Criterion** dropdown specifies the information criteria used to select the final model from the candidate models.

  The **Include GUM** and **Include empty model** check boxes specify whether to include the full model (including all search regressors) or the empty model (including zero search regressors) as possible candidates.

  The **Blocks** edit field allows you to specify the number of blocks into which the search regressors will be split. Typically, if the number of search regressors is less than the number of observations in estimation, only a single block is required. However, if the number of search regressors exceeds the number of observations, they will be split into blocks. EViews will automatically determine the optimal number of blocks, but you may enter your own choice in this field to override the EViews default.

  The **Chronological blocking** and **Alternating blocking** radio buttons determine whether the search regressors are split into blocks in chronological order (the first group of lags/indicators in the first block, followed by the next group in the second block and so on), or alternating (the first variable in the first block, second variable in the second block and so on).

- The **Include indicators** area allows you to specify whether you would like to perform indicator detection, and the type of indicator you would like to detect.

- The **Diagnostics** options allows selection of which diagnostic tests to include, along with their p-values, and for the AR LM test and ARCH LM test, the number of lags to include.

An Example

As an example of using MIDAS regression in EViews, we analyze data used by Armesto, Engemann, and Owyang (2010). The data consists of log-differenced, seasonally adjusted quarterly real GDP between 1947 and 2011 and log-differenced monthly total non-farm payroll employment from 1939 to 2011. These data are in the workfile "Midas.WF1" with the real GDP data in series REALGDP on page "QuarterlyPage", and the employment data in series EMP on page "MonthlyPage".

Beta Weighting

We estimate a MIDAS model with real GDP as the dependent variable and a lagged value of real GDP as a regressor. Only data between 1985 and 2009q1 are used. Monthly employment with 9 lags is used as a set of higher-frequency regressors. The employment data lags are offset by 5 months (*i.e.*, to explain Quarter 1 real GDP, employment data from the previous year's February through October are used).

We use the beta weighting method, while restricting the endpoints coefficient θ_3 to be zero. The following dialog settings reflect this equation specification:

and the associated options

The results of this estimation are given by

Equation: EQ01 Workfile: MIDAS::QuarterlyPage\

View | Proc | Object | Print | Name | Freeze | Estimate | Forecast | Stats | Resids

Dependent Variable: REALGDP
Method: MIDAS
Date: 03/08/16 Time: 15:40
Sample: 1985Q1 2009Q1
Included observations: 97
Method: Beta (restricted endpoints)
Optimization method: initial OPG iterations followed by BFGS
Coefficient covariance computed using observed Hessian
Convergence achieved after 2 iterations

Variable	Coefficient	Std. Error	t-Statistic	Prob.
C	0.665560	0.138964	4.789442	0.0000
REALGDP(-1)	0.284700	0.118813	2.396208	0.0186
Page: MONTHLYPAGE Series: EMP(-5) Lags: 9				
SLOPE	1.912065	0.574790	3.326545	0.0013
BETA01	0.990377	0.073333	13.50516	0.0000
BETA02	6.615724	10.51712	0.629043	0.5309

R-squared	0.337155	Mean dependent var	1.274925
Adjusted R-squared	0.330178	S.D. dependent var	0.682517
S.E. of regression	0.558590	Akaike info criterion	1.755456
Sum squared resid	29.64215	Schwarz criterion	1.888173
Log likelihood	-80.13963	Hannan-Quinn criter.	1.809120
Durbin-Watson stat	2.047218		

MONTHLYPAGE\EMP...	Lag	Coefficient	Distribution
	0	1.232185	
	1	0.419821	
	2	0.175474	
	3	0.062785	
	4	0.017883	
	5	0.003547	
	6	0.000363	
	7	7.40E-06	
	8	1.08E-88	

The top portion of the output describes the estimation sample, MIDAS method, and other estimation settings. Here we see that beta weighting with restricted endpoints ($\theta_3 = 0$). Estimation uses the hybrid method of initial OPG iterations followed by BFGS estimation, with the coefficient covariance computed using the inverse of the negative Hessian.

The first section displays coefficients, standard errors, and t-statistics for the low frequency regressors. These results show standard regression output.

Next, we display the results for the high frequency variables. First, we describe the page and name of the variable and the number lags k used in the low frequency regression. Here, we are using EMP(-5) from the "monthlypage" and allowing for 9 high frequency lags of this variable in the low frequency regression.

The coefficient results for the common SLOPE coefficient (λ) and the free MIDAS beta weight coefficients (θ) are displayed directly below.

First, we see that monthly EMP(-5) has an overall positive effect on REALGDP as the SLOPE coefficient is a statistically significant 1.91.

The actual lag coefficients are obtained by applying weights to this overall slope. The shape of the weight function is determined by the remaining MIDAS coefficients. The θ_1 coefficient, labeled BETA01, is very close to 1, so that the lag pattern depends primarily on BETA02 (θ_2).

The large positive estimate of θ_2, is statistically different from 1, and value of 6.62, implies that the lag pattern is sharply decreasing as shown in the lag coefficient graph at the bottom of the output. We conclude that the coefficient that the zero high frequency lag of employment has a large impact on real GDP, but the effect dies off pretty quickly.

The endpoint coefficient θ_3 has been restricted to be zero so that it does not appear in the output.

The remaining output consists of the standard summary statistics and diagnostics.

To continue our example, we wish to perform a static forecast over the period after the estimation sample (2009q2) to the end of the workfile (2011q2). To forecast from our MIDAS equation, we do so in the usual manner, by first clicking the **Forecast** button:

and filling out the resulting dialog with the appropriate settings. Clicking on **OK** performs the forecast:

Almon Weighting

As a second example we will estimate the same model, but this time using the Almon weighting method, with a second degree polynomial, and we instruct EViews to select the most appropriate number of lags for employment (up to a maximum of 12):

and

Clicking on **OK** estimates the specified equation, and displays the results:

Equation: EQ01 Workfile: MIDAS::QuarterlyPa...

View | Proc | Object | Print | Name | Freeze | Estimate | Forecast | Stats | Resids

Dependent Variable: REALGDP
Method: MIDAS
Date: 03/08/16 Time: 16:45
Sample: 1985Q1 2009Q1
Included observations: 97
Method: PDL/Almon (polynomial degree: 2)
Automatic lag selection, max lags: 12
Chosen selection: 7

Variable	Coefficient	Std. Error	t-Statistic	Prob.
C	0.722293	0.138304	5.222492	0.0000
REALGDP(-1)	0.263092	0.114505	2.297643	0.0238
Page: MONTHLYPAGE Series: EMP(-5) Lags: 7				
PDL01	1.673965	0.488421	3.427299	0.0009
PDL02	-0.354906	0.117500	-3.020477	0.0033

R-squared	0.362509	Mean dependent var	1.274925
Adjusted R-squared	0.355798	S.D. dependent var	0.682517
S.E. of regression	0.547803	Akaike info criterion	1.695838
Sum squared resid	28.50836	Schwarz criterion	1.802011
Log likelihood	-78.24812	Hannan-Quinn criter.	1.738769
Durbin-Watson stat	2.065393		

MONTHLYPAGE\EMP...	Lag	Coefficient	Distribution
	0	1.319059	
	1	0.964153	
	2	0.609246	
	3	0.254340	
	4	-0.100566	
	5	-0.455472	
	6	-0.810378	

As with the beta weighting model estimated above, the overall effect of high frequency EMP(-5) on REALGDP is positive and decreasing in the lags. However, in contrast with the beta weights estimates, the Almon weights lag coefficients do not decline sharply; in fact, the pattern appears to be roughly linear.

Note that the MIDAS variable description line shows that EViews chose 7 as the most appropriate high frequency lag length. Clicking on **View/Model Selection/Criteria Graph** displays a graph of the sum of squared residuals from each of the different lag selections:

providing clear evidence that 7 is the optimal lag.

References

Andreou, Elena, Eric Ghysels, and Andros Kourtellos, 2013. "Should macroeconomic forecasters use daily financial data and how?" *Journal of Business and Economic Statistics,* 3, 240–251.

Armesto, Michelle T., Kristie M. Engemann, and Michael T. Owyang, 2010. "Forecasting with mixed frequencies," *Federal Reserve Bank of St. Louis Review,* 92, 521–36.

Ghysels, Eric, Pedro Santa-Clara, and Rossen Valkanov, 2004. "The MIDAS touch: Mixed data sampling regression models," *University of North Carolina and UCLA Discussion Paper.*

Ghysels, Eric, Pedro Santa-Clara, and Rossen Valkanov, 2006. "Predicting volatility: getting the most out of return data sampled at different frequencies," *Journal of Econometrics*, 131, 59–95.

Ghysels, Eric, Arthur Sinko, and Rossen Valkanov, 2007. "MIDAS regressions: Further results and new directions," *Econometric Reviews*, 26, 53–90.

Chapter 31. Discrete and Limited Dependent Variable Models

The regression methods described in Chapter 20. "Basic Regression Analysis" 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.

EViews offers related tools for estimation of a number of these models under the GLM framework (see Chapter 32. "Generalized Linear Models," beginning on page 421). In some cases, the GLM tools are more general than those provided here; in other cases, they are more restrictive.

Standard introductory discussion for the models presented in this chapter may be found in Greene (2008), Johnston and DiNardo (1997), and Maddala (1983). Wooldridge (1997) 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:

$$\Pr(y_i = 1 | x_i, \beta) = 1 - F(-x_i'\beta), \tag{31.1}$$

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:

$$\Pr(y_i = 0 | x_i, \beta) = F(-x_i'\beta). \tag{31.2}$$

Given such a specification, we can estimate the parameters of this model using the method of maximum likelihood. The likelihood function is given by:

$$l(\beta) = \sum_{i=0}^{n} y_i \log(1 - F(-x_i'\beta)) + (1 - y_i)\log(F(-x_i'\beta)). \tag{31.3}$$

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 1319 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:

$$y_i^* = x_i'\beta + u_i \tag{31.4}$$

where u_i is a random disturbance. Then the observed dependent variable is determined by whether y_i^* exceeds a threshold value:

$$y_i = \begin{cases} 1 & \text{if } y_i^* > 0 \\ 0 & \text{if } y_i^* \le 0. \end{cases} \tag{31.5}$$

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:

$$\Pr(y_i = 1 | x_i, \beta) = \Pr(y_i^* > 0) = \Pr(x_i'\beta + u_i > 0) = 1 - F_u(-x_i'\beta) \tag{31.6}$$

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 $y = 1$:

$$\begin{aligned} \mathrm{E}(y_i | x_i, \beta) &= 1 \cdot \Pr(y_i = 1 | x_i, \beta) + 0 \cdot \Pr(y_i = 0 | x_i, \beta) \\ &= \Pr(y_i = 1 | x_i, \beta). \end{aligned} \tag{31.7}$$

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:

$$y_i = (1 - F(-x_i'\beta)) + \epsilon_i, \tag{31.8}$$

where ϵ_i is a residual representing the deviation of the binary y_i from its conditional mean. Then:

$$\begin{aligned} &\mathrm{E}(\epsilon_i | x_i, \beta) = 0 \\ &\mathrm{var}(\epsilon_i | x_i, \beta) = F(-x_i'\beta)(1 - F(-x_i'\beta)). \end{aligned} \tag{31.9}$$

We will use the conditional mean interpretation in our discussion of binary model residuals (see "Make Residual Series" on page 375).

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 - Binary Choice (Logit, Probit, Extreme Value)** estimation method. The dialog will change to reflect your choice. Alternately, enter the keyword `binary` in the command line and press ENTER.

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	$\Pr(y_i = 1 \mid x_i, \beta) = 1 - \Phi(-x_i'\beta) = \Phi(x_i'\beta)$ where Φ is the cumulative distribution function of the standard normal distribution.
Logit	$\Pr(y_i = 1 \mid x_i, \beta) = 1 - (e^{-x_i'\beta}/(1 + e^{-x_i'\beta}))$ $= e^{x_i'\beta}/(1 + e^{x_i'\beta})$ which is based upon the cumulative distribution function for the logistic distribution.
Extreme value (Gompit)	$\Pr(y_i = 1 \mid x_i, \beta) = 1 - (1 - \exp(-e^{-x_i'\beta}))$ $= \exp(-e^{-x_i'\beta})$ which is based upon the CDF for the Type-I extreme value distribution. Note that this distribution is skewed.

For example, consider the probit specification example described in Greene (2008, p. 781-783) 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 (the data are provided in the workfile "Binary.WF1"). 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 (BFGS / Marquardt steps)
Date: 03/09/15 Time: 15:54
Sample: 1 32
Included observations: 32
Convergence achieved after 23 iterations
Coefficient covariance computed using observed Hessian

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:

$$\frac{\partial \mathrm{E}(y_i | x_i, \beta)}{\partial x_{ij}} = f(-x_i'\beta)\beta_j, \tag{31.10}$$

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 375, 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:

$$\frac{\beta_j}{\beta_k} = \frac{\partial \mathrm{E}(y_i | x_i, \beta) / \partial x_{ij}}{\partial \mathrm{E}(y_i | x_i, \beta) / \partial x_{ik}}. \tag{31.11}$$

In addition to the summary statistics of the dependent variable, EViews also presents the following summary statistics:

McFadden R-squared	0.377478	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
LR statistic	15.54585	Avg. log likelihood	-0.400588
Prob(LR statistic)	0.001405		

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 ordinary residuals:

$$e_i = y_i - \mathrm{E}(y_i | x_i, \beta) = y_i - (1 - F(-x_i'\beta)). \tag{31.12}$$

Additionally, there are several likelihood based statistics:

- **Log likelihood** is the maximized value of the log likelihood function $l(\beta)$.
- **Avg. log likelihood** is the log likelihood $l(\beta)$ 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 $-2(l(\tilde{\beta}) - l(\beta))$. 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 degrees of freedom is one less than the number of coefficients in the equation, 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 $1 - l(\beta) / l(\tilde{\beta})$, 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 E. "Information Criteria," beginning on page 1335. For additional discussion, see Grasa (1989).

Estimation Options

The iteration limit, convergence criterion, and coefficient name 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.

Optimization

By default, EViews uses Newton-Raphson with Marquardt steps to obtain parameter estimates.

If you wish, you can use the **Optimization method** dropdown menu to select a different method. In addition to **Newton-Raphson**, you may select **BFGS**, **OPG - BHHH**, or **EViews legacy**.

For non-legacy estimation, the **Step method** may be chosen between Marquardt, Dogleg, and Line search. For legacy estimation the **Legacy method** is set to the default **Quadratic hill climbing** (Marquardt steps) or **BHHH** (line search).

Note that for legacy estimation, the default optimization algorithm does influence the default method of computing coefficient covariances.

See "Optimization Method" on page 1314 and "Technical Notes" on page 417 for discussion.

Coefficient Covariances

For binary dependent variable models, EViews allows you to estimate the standard errors using the default (inverse of the estimated information matrix), quasi-maximum likelihood (**Huber/White**), cluster quasi-ML (**Cluster-robust**), or generalized linear model (**GLM**) methods. Note that in this setting, the quasi-ML standard errors are associated with misspecified models.

In addition, for ordinary and GLM covariances, you may choose to compute the information matrix estimate using the outer-product of the gradients (**OPG)** or using the negative of the matrix of log-likelihood second derivatives (**Hessian - observed**).

You may elect to compute your covariances with or without a **d.f. Adjustment**.

Note that for legacy estimation, the default algorithm does influence the default method of computing coefficient covariances.

See "Technical Notes" on page 417 for discussion.

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 dropdown menu. 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 dropdown menu.

The EViews default values are selected using a 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 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 at 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, 2008).

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 Diagnostics**, 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.

Dependent Variable Frequencies

This view displays a frequency and cumulative frequency table for the dependent variable in the 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:

Categorical Descriptive Statistics for Explanatory Variables
Equation: EQ_PROBIT
Date: 03/09/15 Time: 16:14

Variable	Mean Dep=0	Mean Dep=1	Mean All
C	1.000000	1.000000	1.000000
GPA	2.951905	3.432727	3.117188
TUCE	21.09524	23.54545	21.93750
PSI	0.285714	0.727273	0.437500

Variable	Standard Deviation Dep=0	Standard Deviation Dep=1	Standard Deviation All
C	0.000000	0.000000	0.000000
GPA	0.357220	0.503132	0.466713
TUCE	3.780275	3.777926	3.901509
PSI	0.462910	0.467099	0.504016
Observations	21	11	32

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:

Expectation-Prediction Evaluation for Binary Specification
Equation: EQ_PROBIT
Date: 03/09/15 Time: 16:15
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	65.63
% Incorrect	14.29	27.27	18.75	0.00	100.00	34.38
Total Gain*	-14.29	72.73	15.63			
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 Dep = 0 observations and 8 of the 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 Dep = 0 and 72.73% of the 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 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 Dep = 1 predictions by 72.73 percentage points, but does more poorly on the Dep = 0 predictions (-14.29 percentage points). Overall, the esti-

mated 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	65.63	34.38	54.88
% Incorrect	19.58	37.68	25.80	34.38	65.63	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:

$$\sum_i \Pr(y_i = 0 \mid x_i, \beta) = \sum_i F(-x_i'\beta), \tag{31.13}$$

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, **E(# 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 417. 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:

Goodness-of-Fit Evaluation for Binary Specification
Andrews and Hosmer-Lemeshow Tests
Equation: EQ_PROBIT
Date: 03/09/15 Time: 16:16
Grouping based upon predicted risk (randomize ties)

	Quantile of Risk		Dep=0		Dep=1		Total	H-L
	Low	High	Actual	Expect	Actual	Expect	Obs	Value
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 obtained by displaying the covariance matrix view using **View/Covariance Matrix** or using the `@covariance` member 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:

$$\frac{\partial \mathrm{E}(y_i | x_i, \beta)}{\partial x_{ij}} = f(-x_i'\beta)\beta_j. \tag{31.14}$$

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	$e_{si} = \dfrac{y_i - \hat{p}_i}{\sqrt{\hat{p}_i(1-\hat{p}_i)}}$
Generalized	$e_{gi} = \dfrac{(y_i - \hat{p}_i)f(-x_i'\hat{\beta})}{\hat{p}_i(1-\hat{p}_i)}$

where $p_i = 1 - F(-x_i'\beta)$ 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.

$$\frac{\partial l(\beta)}{\partial \beta} = \sum_{i=1}^{N} \frac{(y_i - (1 - F(-x_i'\beta)))f(-x_i'\beta)}{F(-x_i'\beta)(1 - F(-x_i'\beta))} \cdot x_i = \sum_{i=1}^{N} e_{g,i} \cdot x_i . \tag{31.15}$$

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. (The following discussion skims over many of the useful features of EViews models. Those wishing greater detail should consult Chapter 49. “Models,” beginning on page 995.)

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.

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 linked to the estimated equation and will open the model window.

What we will do is to use the model to solve for values of the probabilities for various values of GPA, with TUCE equal to the mean value, and PSI equal to 0 in one case, and PSI equal to 1 in a second case. We will define scenarios in the model so that calculations are performed using the desired values. Click on the **Scenarios** button on the model toolbar to display the **Scenario Specification** dialog and click on **Scenario 1** to define the settings for that scenario.

The **Scenario Specification** dialog allows us to define a set of assumptions under which we will solve the model. Click on the **Overrides** tab and enter "GPA PSI TUCE". Defining these overrides tells EViews to use the values in the series GPA_1, PSI_1, and TUCE_1 instead of the original GPA, PSI, and TUCE when solving for GRADE under Scenario 1.

Having defined the first scenario, we must create the series GPA_1, PSI_1 and TUCE_1 in our workfile. We wish to use these series to evaluate the GRADE probabilities for various values of GPA holding TUCE equal to its mean value and PSI equal to 0.

First, we will use the command line to fill GPA_1 with a grid of values ranging from 2 to 4. The easiest way to do this is to use the `@trend` function:

```
series gpa_1 = 2+(4-2)*@trend/(@obs(@trend)-1)
```

Recall that `@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 create series TUCE_1 containing the mean values of TUCE and a series PSI_1 which we set to zero:

```
series tuce_1 = @mean(tuce)
series psi_1 = 0
```

Having prepared our data for the first scenario, we will now use the model object to define an alternate scenario where PSI = 1. Return to the **Select Scenario** tab, select **Copy Scenario**, then select **Scenario 1** as the **Source**, and **New Scenario** as the **Destination**. Copying Scenario 1 creates a new scenario, Scenario 2, that instructs EViews to use the values in the series GPA_2, PSI_2, and TUCE_2 when solving for GRADE. These values are initialized from the corresponding Scenario 1 series defined previously. We then set PSI_2 equal to 1 by issuing the command

```
series psi_2 = 1
```

We are now ready to solve the model under the two scenarios. Click on the **Solve** button and set the **Active** solution scenario to **Scenario 1** and the **Alternate** solution scenario to **Scenario 2**. Be sure to click on the checkbox **Solve for Alternate along with Active** so that EViews knows to solve for both. You can safely ignore the remaining solution settings and simply click on **OK**.

EViews will report that your model has solved successfully and will place the solutions in the series GRADE_1 and GRADE_2, respectively. To display the results, select **Object/New Object.../Group**, and enter:

```
gpa_1 grade_1 grade_2
```

EViews will open an untitled group window containing these three series. Select **View/Graph/XY line** to display a graph of the fitted GRADE probabilities plotted against GPA for those with PSI = 0 (GRADE_1) and with PSI = 1 (GRADE_2), both computed with TUCE evaluated at means.

We have annotated the graph slightly so that you can better judge the effect of the new teaching methods (PSI) on the probability of grade improvement for various values of the student's GPA.

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:

$$\mathrm{var}(u_i) = \exp(2z_i'\gamma), \tag{31.16}$$

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:

$$\frac{(y_i - p_i)}{\sqrt{p_i(1-p_i)}} = \frac{f(-x_i'\beta)}{\sqrt{p_i(1-p_i)}} x_i' b_1 + \frac{f(-x_i'\beta)(-x_i'\beta)}{\sqrt{p_i(1-p_i)}} z_i' b_2 + v_i, \tag{31.17}$$

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 p_i and fitted index $x_i'\beta$. 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:

$$y_i^* = x_i'\beta + \epsilon_i \tag{31.18}$$

where is ϵ_i are independent and identically distributed random variables. The observed y_i is determined from y_i^* using the rule:

$$y_i = \begin{cases} 0 & \text{if } y_i^* \le \gamma_1 \\ 1 & \text{if } \gamma_1 < y_i^* \le \gamma_2 \\ 2 & \text{if } \gamma_2 < y_i^* \le \gamma_3 \\ \dots & \dots \\ M & \text{if } \gamma_M < y_i^* \end{cases} \tag{31.19}$$

so that there are $M+1$ distinct categories. 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

$$\begin{aligned} \Pr(y_i = 0 | x_i, \beta, \gamma) &= F(\gamma_1 - x_i'\beta) \\ \Pr(y_i = 1 | x_i, \beta, \gamma) &= F(\gamma_2 - x_i'\beta) - F(\gamma_1 - x_i'\beta) \\ \Pr(y_i = 2 | x_i, \beta, \gamma) &= F(\gamma_3 - x_i'\beta) - F(\gamma_2 - x_i'\beta) \\ &\dots \\ \Pr(y_i = M | x_i, \beta, \gamma) &= 1 - F(\gamma_M - x_i'\beta) \end{aligned} \tag{31.20}$$

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:

$$l(\beta, \gamma) = \sum_{i=1}^{N} \sum_{j=0}^{M} \log(\Pr(y_i = j | x_i, \beta, \gamma)) \cdot 1(y_i = j) \tag{31.21}$$

where $1(.)$ 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 1320).

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.

(The data, which are from Allison, Truett, and D.V. Cicchetti (1976).“Sleep in Mammals: Ecological and Constitutional Correlates,” *Science*, 194, 732-734, are available in the “Order.WF1” dataset. A more complete version of the data may be obtained from StatLib: http://lib.stat.cmu.edu/datasets/sleep).

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.

You may click on the **Options** tab to set the iteration limit, convergence criterion, optimization algorithm, and most importantly, method for computing coefficient covariances. See “Technical Notes” on page 417 for a discussion of these methods.

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 informa-

tion, number of distinct values for y, and the method of computing the coefficient covariance matrix.

Dependent Variable: DANGER
Method: ML - Ordered Probit (Quadratic hill climbing)
Date: 08/12/09 Time: 00:13
Sample (adjusted): 1 61
Included observations: 58 after adjustments
Number of ordered indicator values: 5
Convergence achieved after 7 iterations
Covariance matrix computed using second derivatives

Variable	Coefficient	Std. Error	z-Statistic	Prob.
BODY	0.000247	0.000421	0.587475	0.5569
BRAIN	-0.000397	0.000418	-0.950366	0.3419
SLEEP	-0.199508	0.041641	-4.791138	0.0000

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 (2008, section 23.10) 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:

$$\frac{\partial \Pr(y = k)}{\partial \beta_j} = \frac{\partial F(\gamma_{k+1} - x_i'\beta)}{\partial \beta_j} - \frac{\partial F(\gamma_k - x_i'\beta)}{\partial \beta_j} \tag{31.22}$$

for $k = 1, 2, \ldots, M-1$. 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:

Limit Points

LIMIT_2:C(4)	-2.798449	0.514784	-5.436166	0.0000
LIMIT_3:C(5)	-2.038945	0.492198	-4.142527	0.0000
LIMIT_4:C(6)	-1.434567	0.473679	-3.028563	0.0025
LIMIT_5:C(7)	-0.601211	0.449109	-1.338675	0.1807

Pseudo R-squared	0.147588	Akaike info criterion	2.890028
Schwarz criterion	3.138702	Log likelihood	-76.81081
Hannan-Quinn criter.	2.986891	Restr. log likelihood	-90.10996
LR statistic	26.59830	Avg. log likelihood	-1.324324
Prob(LR statistic)	0.000007		

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 368) 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.
- **Prediction Evaluation**— classifies observations on the basis of the predicted response. EViews performs the classification on the basis of the category with the maximum predicted probability.

  The first portion of the output shows results for the estimated equation and for the constant probability (no regressor) specifications.

Prediction Evaluation for Ordered Specification
Equation: EQ_ORDER
Date: 08/12/09 Time: 00:20

Estimated Equation					
Dep. Value	Obs.	Correct	Incorrect	% Correct	% Incorrect
1	18	10	8	55.556	44.444
2	14	6	8	42.857	57.143
3	10	0	10	0.000	100.000
4	9	3	6	33.333	66.667
5	7	6	1	85.714	14.286
Total	58	25	33	43.103	56.897
Constant Probability Spec.					
Dep. Value	Obs.	Correct	Incorrect	% Correct	% Incorrect
1	18	18	0	100.000	0.000
2	14	0	14	0.000	100.000
3	10	0	10	0.000	100.000
4	9	0	9	0.000	100.000
5	7	0	7	0.000	100.000
Total	58	18	40	31.034	68.966

Each row represents a distinct value for the dependent variable. The "Obs" column indicates the number of observations with that value. Of those, the number of "Correct" observations are those for which the predicted probability of the response is the highest. Thus, 10 of the 18 individuals with a DANGER value of 1 were correctly specified. Overall, 43% of the observations were correctly specified for the fitted model versus 31% for the constant probability model.

The bottom portion of the output shows additional statistics measuring this improvement

Gain over Constant Prob. Spec.					
Dep. Value	Obs.	Equation % Incorrect	Constant % Incorrect	Total Gain*	Pct. Gain**
1	18	44.444	0.000	-44.444	NA
2	14	57.143	100.000	42.857	42.857
3	10	100.000	100.000	0.000	0.000
4	9	66.667	100.000	33.333	33.333
5	7	14.286	100.000	85.714	85.714
Total	58	56.897	68.966	12.069	17.500

Note the improvement in the prediction for DANGER values 2, 4, and especially 5 comes from refinement of the constant only prediction of DANGER = 1.

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 containing 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 49. "Models," beginning on page 995 for additional detail on modifying and solving models.

For this example, the series I_DANGER_1 will contain the fitted linear index $x_i'\beta$. 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:

$$e_{gi} = \frac{f(\gamma_{y_{i+1}} - x_i'\beta) - f(\gamma_{y_i} - x_i'\beta)}{F(\gamma_{y_{i+1}} - x_i'\beta) - F(\gamma_{y_i} - x_i'\beta)}, \tag{31.23}$$

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:

$$y_i^* = x_i'\beta + \sigma\epsilon_i, \tag{31.24}$$

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:

$$y_i = \begin{cases} 0 & \text{if } y_i^* \le 0 \\ y_i^* & \text{if } y_i^* > 0 \end{cases} \tag{31.25}$$

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:

$$y_i = \begin{cases} \underline{c}_i & \text{if } y_i^* \le \underline{c}_i \\ y_i^* & \text{if } \underline{c}_i < y_i^* \le \bar{c}_i \\ \bar{c}_i & \text{if } \bar{c}_i < y_i^* \end{cases} \tag{31.26}$$

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:

$$\begin{aligned} l(\beta, \sigma) &= \sum_{i=1}^{N} \log f((y_i - x_i'\beta)/\sigma) \cdot 1(\underline{c}_i < y_i < \bar{c}_i) \\ &+ \sum_{i=1}^{N} \log(F((\underline{c}_i - x_i'\beta)/\sigma)) \cdot 1(y_i = \underline{c}_i) \\ &+ \sum_{i=1}^{N} \log(1 - F((\bar{c}_i - x_i'\beta)/\sigma)) \cdot 1(y_i = \bar{c}_i) \end{aligned} \tag{31.27}$$

where f, F are the density and cumulative distribution functions of ϵ, respectively.

Estimating Censored Models in EViews

Suppose that we wish to estimate the model:

$$\mathrm{HRS}_i = \beta_1 + \beta_2 \mathrm{AGE}_i + \beta_3 \mathrm{EDU}_i + \beta_4 \mathrm{KID1}_i + \epsilon_i, \tag{31.28}$$

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 - Censored or Truncated Data (including Tobit)** estimation method. Alternately, enter the keyword `censored` in the command line and press ENTER. 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	$\mathrm{E}(\epsilon) = 0, \mathrm{var}(\epsilon) = 1$
Logistic	$\mathrm{E}(\epsilon) = 0, \mathrm{var}(\epsilon) = \pi^2/3$
Extreme value (Type I)	$\mathrm{E}(\epsilon) \approx -0.5772$ (Euler's constant), $\mathrm{var}(\epsilon) = \pi^2/6$

Bear in mind that the extreme value distribution is asymmetric.

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 ($\underline{c}_i$ 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 unemployment 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 see an example of header output from a left censored model (our Fair's (1978) tobit model described below) where the censoring is specified by value:

```
Dependent Variable: Y_PT
Method: ML - Censored Normal (TOBIT)  (Newton-Raphson /
      Marquardt steps)
Date: 03/09/15   Time: 16:23
Sample: 1 601
Included observations: 601
Left censoring (value) at zero
Convergence achieved after 8 iterations
Coefficient covariance computed using observed Hessian
```

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 0.4766 for a model with extreme value errors, the implied standard error of the error term is $0.5977 = 0.4766\pi/\sqrt{6}$.

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 $\hat{\epsilon}_i = y_i - \mathrm{E}(y_i | x_i, \hat{\beta}, \hat{\sigma})$ 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	$e_{oi} = y_i - \mathrm{E}(y_i \mid x_i, \beta, \hat{\sigma})\, f'((y_i - x_i'\beta)/\hat{\sigma})$
Standardized	$e_{si} = \dfrac{y_i - \mathrm{E}(y_i \mid x_i, \beta, \hat{\sigma})}{\sqrt{\mathrm{var}(y_i \mid x_i, \beta, \hat{\sigma})}}$
Generalized	$e_{gi} = -\dfrac{f((\underline{c}_i - x_i'\beta)/\hat{\sigma})}{\hat{\sigma}F((\underline{c}_i - x_i'\beta)/\hat{\sigma})} \cdot 1(y_i \le \underline{c}_i)$ $- \dfrac{f'((y_i - x_i'\beta)/\hat{\sigma})}{\hat{\sigma}f((y_i - x_i'\beta)/\hat{\sigma})} \cdot 1(\underline{c}_i < y_i < \bar{c}_i)$ $+ \dfrac{f((\bar{c}_i - x_i'\beta)/\hat{\sigma})}{\hat{\sigma}(1-F((\bar{c}_i - x_i'\beta)/\hat{\sigma}))} \cdot 1(y_i \ge \bar{c}_i)$

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 $\mathrm{E}(y_i \mid x_i, \beta, \sigma)$.

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 (2008), provides a brief discussion and additional references).

Forecasting

EViews provides you with the option of forecasting the expected dependent variable, $\mathrm{E}(y_i \mid x_i, \beta, \sigma)$, or the expected latent variable, $\mathrm{E}(y_i^* \mid x_i, \beta, \sigma)$. 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 $\mathrm{E}(y_i^* \mid x_i, \beta, \sigma)$ may be derived from the latent model using the relationship:

$$\hat{y}_i^* = \mathrm{E}(y_i^* \mid x_i, \beta, \hat{\sigma}) = x_i'\beta - \hat{\sigma}\gamma\,. \tag{31.29}$$

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:

$$\begin{aligned}\hat{y}_i = \mathrm{E}(y_i \mid x_i, \beta, \hat{\sigma}) &= \underline{c}_i \cdot \Pr(y_i = \underline{c}_i \mid x_i, \beta, \hat{\sigma}) \\ &+ \mathrm{E}(y_i^* \mid \underline{c}_i < y_i^* < \bar{c}_i\,;\, x_i, \beta, \hat{\sigma}) \cdot \Pr(\underline{c}_i < y_i^* < \bar{c}_i \mid x_i, \beta, \hat{\sigma}) \\ &+ \bar{c}_i \cdot \Pr(y_i = \underline{c}_i \mid x_i, \beta, \hat{\sigma})\end{aligned} \tag{31.30}$$

Note that these forecasts always satisfy $\underline{c}_i \le \hat{y}_i \le \bar{c}_i$. 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:

$$\Pr(y_i = \underline{c}_i | x_i, \beta, \hat{\sigma}) = \Pr(y_i^* \le \underline{c}_i | x_i, \beta, \hat{\sigma}) = F((\underline{c}_i - x_i'\beta)/\hat{\sigma}). \tag{31.31}$$

Censored Model Example

As an example, we replicate Fair's (1978) tobit model that estimates the incidence of extramarital affairs ("Tobit_Fair.WF1). The dependent variable, number of extramarital affairs (Y_PT), is left censored at zero and the errors are assumed to be normally distributed. The top portion of the output was shown earlier; bottom portion of the output is presented below:

Variable	Coefficient	Std. Error	z-Statistic	Prob.
C	7.608487	3.905987	1.947904	0.0514
Z1	0.945787	1.062866	0.889847	0.3735
Z2	-0.192698	0.080968	-2.379921	0.0173
Z3	0.533190	0.146607	3.636852	0.0003
Z4	1.019182	1.279575	0.796500	0.4257
Z5	-1.699000	0.405483	-4.190061	0.0000
Z6	0.025361	0.227667	0.111394	0.9113
Z7	0.212983	0.321157	0.663173	0.5072
Z8	-2.273284	0.415407	-5.472429	0.0000
Error Distribution				
SCALE:C(10)	8.258432	0.554581	14.89131	0.0000

Mean dependent var	1.455907	S.D. dependent var	3.298758
S.E. of regression	3.058957	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 Diagnostics/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 Diagnostics/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.

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 $x_i'\beta < 0$, and replacing y_i with $2(x_i'\beta)$ for all observations where $y_i < 2(x_i'\beta)$. 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 conditional 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**, edit the specification, replacing the dependent variable "Y_PT" with "Y_C", 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…** again 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 (2008, 23.3.4) 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:

$$y_i^* = x_i'\beta + \sigma\epsilon_i \tag{31.32}$$

where $y_i = y_i^*$ is only observed if:

$$\underline{c}_i < y_i^* < \bar{c}_i . \tag{31.33}$$

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:

$$\begin{aligned} l(\beta, \sigma) &= \sum_{i=1}^{N} \log f((y_i - x_i'\beta)/\sigma) \cdot 1(\underline{c}_i < y_i < \bar{c}_i) \\ &- \sum_{i=1}^{N} \log(F((\bar{c}_i - x_i'\beta)/\sigma) - F((\underline{c}_i - x_i'\beta)/\sigma)). \end{aligned} \tag{31.34}$$

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	$e_{oi} = y_i - \mathrm{E}(y_i^* \mid \underline{c}_i < y_i^* < \bar{c}_i \,;\, x_i, \beta, \hat{\sigma})$
Standardized	$e_{si} = \dfrac{y_i - \mathrm{E}(y_i^* \mid \underline{c}_i < y_i^* < \bar{c}_i \,;\, x_i, \beta, \hat{\sigma})}{\sqrt{\mathrm{var}(y_i^* \mid \underline{c}_i < y_i^* < \bar{c}_i \,;\, x_i, \beta, \hat{\sigma})}}$

Generalized	$e_{gi} = -\frac{f'((y_i - x_i'\beta)/\hat{\sigma})}{\sigma f((y_i - x_i'\beta)/\hat{\sigma})} - \frac{f((\bar{c}_i - x_i'\beta)/\hat{\sigma}) - f((\underline{c}_i - x_i'\beta)/\hat{\sigma})}{\sigma(F((\bar{c}_i - x_i'\beta)/\hat{\sigma}) - F((\underline{c}_i - x_i'\beta)/\hat{\sigma}))}$

where f, F, are the density and distribution functions. Details on the computation of $\mathrm{E}(y_i | \underline{c}_i < y_i^* < \bar{c}_i;\ x_i, \hat{\beta}, \hat{\sigma})$ 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 Gourieroux, Monfort and Trognon (1987); Greene (2008) provides a brief discussion and additional references).

Forecasting

EViews provides you with the option of forecasting the expected observed dependent variable, $\mathrm{E}(y_i | x_i, \hat{\beta}, \hat{\sigma})$, or the expected latent variable, $\mathrm{E}(y_i^* | x_i, \hat{\beta}, \hat{\sigma})$.

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 $\mathrm{E}(y_i^* | x_i, \hat{\beta}, \hat{\sigma})$ are computed using:

$$\hat{y}_i^* = \mathrm{E}(y_i^* | x_i, \hat{\beta}, \hat{\sigma}) = x_i'\hat{\beta} - \hat{\sigma}\gamma . \tag{31.35}$$

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:

$$\hat{y}_i = \mathrm{E}(y_i^* | \underline{c}_i < y_i^* < \bar{c}_i;\ x_i, \hat{\beta}, \hat{\sigma}) \tag{31.36}$$

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 $\underline{c}_i < \hat{y}_i < \bar{c}_i$.

It is instructive to compare this latter expected value with the expected value derived for the censored model in Equation (31.30) above (repeated here for convenience):

$$\begin{aligned} \hat{y}_i = \mathrm{E}(y_i | x_i, \hat{\beta}, \hat{\sigma}) &= \underline{c}_i \cdot \Pr(y_i = \underline{c}_i | x_i, \hat{\beta}, \hat{\sigma}) \\ &+ \mathrm{E}(y_i^* | \underline{c}_i < y_i^* < \bar{c}_i;\ x_i, \hat{\beta}, \hat{\sigma}) \cdot \Pr(\underline{c}_i < y_i^* < \bar{c}_i | x_i, \hat{\beta}, \hat{\sigma}) \\ &+ \bar{c}_i \cdot \Pr(y_i = \underline{c}_i | x_i, \hat{\beta}, \hat{\sigma}). \end{aligned} \tag{31.37}$$

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 $\underline{c}_i$ 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) (Newton-Raphson / Marquardt steps)
Date: 03/09/15 Time: 16:26
Sample (adjusted): 452 601
Included observations: 150 after adjustments
Truncated sample
Left censoring (value) at zero
Convergence achieved after 11 iterations
Coefficient covariance computed using observed Hessian

Variable	Coefficient	Std. Error	z-Statistic	Prob.
C	12.37287	5.178533	2.389261	0.0169
Z1	-1.336854	1.451426	-0.921063	0.3570
Z2	-0.044791	0.116125	-0.385719	0.6997
Z3	0.544174	0.217885	2.497527	0.0125
Z4	-2.142868	1.784389	-1.200897	0.2298
Z5	-1.423107	0.594582	-2.393459	0.0167
Z6	-0.316717	0.321882	-0.983953	0.3251
Z7	0.621418	0.477420	1.301618	0.1930
Z8	-1.210020	0.547810	-2.208833	0.0272
Error Distribution				
SCALE:C(10)	5.379485	0.623787	8.623910	0.0000

Mean dependent var	5.833333	S.D. dependent var	4.255934
S.E. of regression	4.013126	Akaike info criterion	5.344456
Sum squared resid	2254.725	Schwarz criterion	5.545165
Log likelihood	-390.8342	Hannan-Quinn criter.	5.425998
Avg. log likelihood	-2.605561		

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 with a sample that is adjusted accordingly, and that the frequency information at the bottom of the screen shows that there are no left and right censored observations.

Heckman Selection Model

The Heckman (1976) selection model, sometimes called the Heckit model, is a method for estimating regression models which suffer from sample selection bias. Under the Heckman selection framework, the dependent variable y_i is only observable for a portion of the data. A classic example, in economics, of the sample selection problem is the wage equation for women, whereby a woman's wage is only observed if she makes the decision to enter the work place, and is unobservable if she does not. Heckman's (1976) paper that introduced the Heckman Selection model worked on this very problem.

EViews provides an estimator for the simple linear Heckman Selection Model. This model is specified as:

$$y_i = X_i'\beta + \epsilon_i \tag{31.38}$$

$$z_i = W_i\gamma + u_i \tag{31.39}$$

where is z_i a binary variable, with y_i only observed when $z_i = 1$. ϵ_t and u_t are error terms which follow a bivariate normal distribution:

$$\begin{bmatrix} \epsilon_i \\ u_i \end{bmatrix} \sim N \begin{bmatrix} \sigma^2 & \rho\sigma \\ \rho\sigma & 1 \end{bmatrix} \tag{31.40}$$

with scale parameter σ and correlation coefficient ρ. Note that we have normalized the variance of u_t to 1 since this variance is not identified in this model.

Equation Equation (31.38) is generally referred to as the response equation, with y_i the variable of interest. Equation Equation (31.39) is termed the selection equation and determines whether is y_i observed or not.

EViews offers two different methods of estimating this model: Heckman's original two-step method and a Maximum Likelihood method.

The Heckman Two-Step Method

The Heckman two-step method is based around the observation that:

$$E(y_i | Z_i = 1) = X_i'\beta + \rho\sigma\lambda_i(W_i\gamma) \tag{31.41}$$

where $\lambda(X) = \phi(X)/\Phi(x)$ is the Inverse Mills Ratio (Greene, 2008), and ϕ and Φ are the standard normal density and cumulative distribution function, respectively. Then we may specify a regression model:

$$y_i = X_i'\beta + \rho\sigma\lambda_i(W_i\gamma) + v_i \tag{31.42}$$

The two-step method proceeds by first estimating a Probit regression for Equation (31.39) to obtain an estimate of $\hat{\gamma}$, from which $\lambda_i(W_i\hat{\gamma})$ may be calculated. A least squares regression of y_i on $X_i'\beta$ and $\hat{\lambda}_i$

$$y_i = X_i'\beta + \rho\sigma\hat{\lambda}_i + v_i \tag{31.43}$$

is then computed, yielding consistent estimates of β and $\theta = \rho\sigma$. An estimator for the error standard deviation σ may be obtained from the usual standard error of the regression s, and the ratio estimator $\hat{\rho} = \hat{\theta}/s$ follows.

The estimator of the coefficient covariance matrix of the two-step method is given by:

$$\hat{\Omega} = \hat{\sigma}^2(X^{*\prime}X^*)^{-1}(X^{*\prime}(I - \hat{\rho}^2\hat{\Delta})X^* + Q)(X^{*\prime}X^*)^{-1} \tag{31.44}$$

where $X^*_i = (X_i', \hat{\lambda}_i)'$, $\hat{\Delta}$ is a diagonal matrix with $\hat{\delta}_i = \hat{\lambda}_i(\hat{\lambda}_i - W_i\hat{\gamma})$ on the diagonals, I is an identity matrix, $Q = \hat{\rho}^2(X^{*\prime}\hat{\Delta}W)\hat{V}(X^{*\prime}\hat{\Delta}W)$, and $\hat{V}$ is the coefficient covariance matrix from the Probit estimation of Equation (31.39).

Maximum Likelihood

The maximum likelihood method of estimating the Heckman Selection Model is performed using the log-likelihood function given by:

$$\log L(\beta, \gamma, \rho, \sigma | X, W) = \sum_{i|z_i=0} (\log 1 - \Phi(W_i\gamma)) + \tag{31.45}$$

$$\sum_{i|z_i=1} -\log(\sigma) + \log\left(\phi\left(\frac{y_i - X_i\beta}{\sigma}\right)\right) + \log\left(1 - \Phi\left(\frac{-\left(W_i\gamma + \rho\left(\frac{y_i - X_i\beta}{\sigma}\right)\right)}{\sqrt{1-\rho^2}}\right)\right)$$

where the first summation is over observations for which $z_i = 0$ (*i.e.*, when y_i is unobserved), and the second for observations for which $z_i = 1$ (*i.e.*, when y_i is observed).

It is straightforward to maximize this log-likelihood function with respect to the parameters, $\beta, \gamma, \rho, \sigma$. However, this maximization is unrestricted with regards to ρ and σ, when, in fact, there are restrictions of the form $-(1 < \rho < 1)$ and $\sigma > 1$ imposed on the parameters. EViews optimizes the model using transformed versions of the parameters:

$$\sigma = \exp(\sigma^*) \tag{31.46}$$

$$\rho = \arctan(\rho^*)(2/\pi) \tag{31.47}$$

to impose the restrictions.

Starting values for the optimization can be obtained using the Heckman two-step method outlined above.

As with most maximum likelihood estimations, the covariance matrix of the estimated parameters can be calculated as either $(-H)^{-1}$ (where H is the Hessian matrix, the infor-

mation matrix), $(GG')^{-1}$ (where G is the matrix of gradients), or as $H^{-1}GG'H^{-1}$ (the Huber/White matrix).

Estimating the Heckman Selection Model in EViews

To estimate the Heckman Selection Model, open the equation dialog by selecting **Quick/Estimate Equation…** or **Object/New Object…/Equation** in the main EViews menu and selecting **Heckit** from the **Method** dropdown menu. Alternately, you may enter the command heckit in the command line.

The first page of the dialog, the **Specification** tab, lets you specify both the response equation Equation (31.38) and the selection equation Equation (31.39). Both equations should be specified as the dependent variable followed by a space delimited list of regressors. Note that the dependent variable for the selection equation should be series containing only zeros and ones.

The **Specification** page also lets you select the type of estimation method by selecting one of the two radio buttons; either **Maximum Likelihood** or **Heckman two-step**.

If you have chosen to estimate via maximum likelihood, the **Options** tab of the dialog lets you specify the type of covariance matrix, by using the **Coefficient covariance matrix** dropdown menu. You may choose from **Outer Product of Gradients**, **Information Matrix**, and **Huber/White**. You may also choose starting values for the maximum likelihood procedure. You may select **EViews Supplied** to perform the Heckman two-step procedure to obtain starting values, or you can down-weight those starting values by choosing a multiple of them. The **User Supplied** option will use the specified coefficient vector in the workfile to obtain starting values.

An Example

As an example of the estimation of the Heckman Selection model, we take one of the results from Econometric Analysis by William H. Greene (6th Edition, p. 888, Example 24.8), which uses data from the Mroz (1987) study of the labor supply of married women to estimate a wage equation for women. Only 428 of the 753 women studied participated in the labor force, so a selection equation is provided to model the sample selection behavior of married women.

The wage equation is given by:

$$\text{Wage} = \beta_1 + \beta_2\text{Exper} + \beta_3\text{Exper}^2 + \beta_4\text{Educ} + \beta_5\text{City} + \epsilon \tag{31.48}$$

where EXPER is a measure of each woman's experience, EDUC is her level of education, and CITY is a dummy variable for whether she lives in a city or not.

The selection equation is given by:

$$\text{LFP} = \gamma_1 + \gamma_2\text{Age} + \gamma_3\text{Age}^2 + \gamma_4\text{Faminc} + \gamma_5\text{Educ} + \gamma_6\text{Kids} + u \tag{31.49}$$

where LFP is a binary variable taking a value of 1 if the woman is in the labor force, and 0 otherwise, AGE is her age, FAMINC is the level of household income not earned by the woman, and KIDS is a dummy variable for whether she has children.

You can bring the Mroz data directly into EViews from Greene's website, using the following EViews command:

```
wfopen http://www.stern.nyu.edu/~wgreene/Text/Edition7/TableF5-
    1.txt
```

In this data, the wage data are in the series WW, experience is AX, education is in WE, the city dummy is CIT, labor force participation is LFP, age is WA, and family income is FAMINC. There is no kids dummy variable, but there are two variables containing the number of children below K6 education (KL6), and the number of kids between K6 education and 18 (K618). We can create the dummy variable simply by testing whether the sum of those two variables is greater than 0.

To estimate this equation in EViews, we click on **Quick/Estimate Equation**…, and then change the equation method to **Heckit**. In the **Response Equation** box we type:

```
ww c ax ax^2 we cit
```

And in the **Selection Equation** box we type:

```
lfp c wa wa^2 faminc we (kl6+k618)>0
```

To begin we select the **Heckman two-step** estimation method. After clicking **OK**, the estimation results show and replicate the results in the first pane of Table 24.3 in Greene (note that Greene only shows the estimates of the Wage equation, plus ρ and σ).

Dependent Variable: WW
Method: Two-Step Heckman Selection
Date: 03/09/15 Time: 16:31
Sample: 1 753
Included observations: 753
Selection Variable: LFP
Coefficient covariance computed using two-step Heckman method

Variable	Coefficient	Std. Error	t-Statistic	Prob.
		Response Equation - WW		
C	-0.971200	2.132849	-0.455353	0.6490
AX	0.021061	0.062532	0.336804	0.7364
AX^2	0.000137	0.001882	0.072842	0.9420
WE	0.417017	0.104157	4.003746	0.0001
CIT	0.443838	0.316531	1.402194	0.1613
		Selection Equation - LFP		
C	-4.156807	1.402086	-2.964730	0.0031
WA	0.185395	0.065967	2.810436	0.0051
WA^2	-0.002426	0.000774	-3.136096	0.0018
FAMINC	4.58E-06	4.21E-06	1.088918	0.2765
WE	0.098182	0.022984	4.271744	0.0000
(KL6+K618)>0	-0.448987	0.130911	-3.429697	0.0006

Mean dependent var	4.177682	S.D. dependent var	3.310282
S.E. of regression	2.418304	Akaike info criterion	6.017314
Sum squared resid	4327.663	Schwarz criterion	6.084863
Log likelihood	-2254.519	Hannan-Quinn criter.	6.043337

We can modify our equation to use as the estimation method. Click on the **Estimate** button to bring up the estimation dialog and change the method to **Maximum Likelihood**. Next, click on the **Options** tab and change the **Information matrix** to **OPG** and click on **OK** to estimate the equation. The results match the second pane of Table 24.3 in Greene.

Dependent Variable: WW
Method: ML Heckman Selection (Newton-Raphson / Marquardt steps)
Date: 03/09/15 Time: 16:34
Sample: 1 753
Included observations: 753
Selection Variable: LFP
Convergence achieved after 6 iterations
Coefficient covariance computed using outer product of gradients

Variable	Coefficient	Std. Error	t-Statistic	Prob.
Response Equation - WW				
C	-1.963024	1.680330	-1.168237	0.2431
AX	0.027868	0.075614	0.368562	0.7126
AX^2	-0.000104	0.002341	-0.044369	0.9646
WE	0.457005	0.096271	4.747067	0.0000
CIT	0.446529	0.426937	1.045889	0.2960
Selection Equation - LFP				
C	-4.119692	1.410456	-2.920822	0.0036
WA	0.184015	0.065841	2.794837	0.0053
WA^2	-0.002409	0.000773	-3.114124	0.0019
FAMINC	5.68E-06	3.89E-06	1.460278	0.1446
WE	0.095281	0.023999	3.970163	0.0001
(KL6+K618)>0	-0.450615	0.136668	-3.297155	0.0010
Interaction terms				
@LOG(SIGMA)	1.134100	0.026909	42.14565	0.0000
TFORM(RHO)	-0.210301	0.367061	-0.572931	0.5669
SIGMA	3.108376	0.083644	37.16219	0.0000
RHO	-0.131959	0.223781	-0.589676	0.5556

Mean dependent var	4.177682	S.D. dependent var	3.310282
S.E. of regression	2.361759	Akaike info criterion	4.234416
Sum squared resid	4127.650	Schwarz criterion	4.314247
Log likelihood	-1581.258	Hannan-Quinn criter.	4.265171

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 - Integer Count Data** 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:

$$m(x_i, \beta) = \mathrm{E}(y_i | x_i, \beta) = \exp(x_i'\beta). \tag{31.50}$$

- 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:

$$f(y_i|x_i, \beta) = e^{-m(x_i, \beta)} m(x_i, \beta)^{y_i} / y_i! \tag{31.51}$$

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:

$$l(\beta) = \sum_{i=1}^{N} y_i \log m(x_i, \beta) - m(x_i, \beta) - \log(y_i!) . \tag{31.52}$$

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 coefficient variance matrix consistently estimated by the inverse of the Hessian:

$$V = \mathrm{var}(\hat{\beta}) = \left(\sum_{i=1}^{N} \hat{m}_i x_i x_i' \right)^{-1} \tag{31.53}$$

where $\hat{m}_i = m(x_i, \hat{\beta})$. Alternately, one could estimate the coefficient covariance using the inverse of the outer-product of the scores:

$$V = \mathrm{var}(\hat{\beta}) = \left(\sum_{i=1}^{N} (y_i - \hat{m}_i)^2 x_i x_i' \right)^{-1} \tag{31.54}$$

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:

$$v(x_i, \beta) = \mathrm{var}(y_i|x_i, \beta) = \mathrm{E}(y_i|x_i, \beta) = m(x_i, \beta) . \tag{31.55}$$

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:

$$l(\beta, \eta) = \sum_{i=1}^{N} y_i \log(\eta^2 m(x_i, \beta)) - (y_i + 1/\eta^2) \log(1 + \eta^2 m(x_i, \beta)) \tag{31.56}$$

$$+ \log\Gamma(y_i + 1/\eta^2) - \log(y_i!) - \log\Gamma(1/\eta^2)$$

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:

$$\begin{aligned} \mathrm{E}(y_i | x_i, \beta) &= m(x_i, \beta) \\ \mathrm{var}(y_i | x_i, \beta) &= m(x_i, \beta)(1 + \eta^2 m(x_i, \beta)) \end{aligned} \tag{31.57}$$

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 417 for details on these options.

Further details on QML estimation are provided by Gourioux, Monfort, and Trognon (1994a, 1994b). Wooldridge (1997) 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:

$$l(\beta) = \sum_{i=1}^{N} -\log m(x_i, \beta) - y_i / m(x_i, \beta) . \tag{31.58}$$

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:

$$l(\beta) = \sum_{i=1}^{N} -\frac{1}{2}\left(\frac{y_i - m(x_i, \beta)}{\sigma}\right)^2 - \frac{1}{2}\log(\sigma^2) - \frac{1}{2}\log(2\pi) . \tag{31.59}$$

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:

$$y_i = m(x_i, \beta) + \epsilon_i . \tag{31.60}$$

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 vari-**

ance 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 Diagnostics/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 (1997).

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 $x_i'\hat{\beta}$. Note that for all of these models the forecasts of y_i are given by $\hat{y}_i = m(x_i, \hat{\beta})$ where $m(x_i, \hat{\beta}) = \exp(x_i'\hat{\beta})$.
- **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)	$e_{si} = \dfrac{y_i - m(x_i, \hat{\beta})}{\sqrt{v(x_i, \hat{\beta}, \hat{\gamma})}}$
Generalized	e_g = (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 (1997).

Demonstrations

A Specification Test for Overdispersion

Consider the model:

$$\mathrm{NUMB}_i = \beta_1 + \beta_2 \mathrm{IP}_i + \beta_3 \mathrm{FEB}_i + \epsilon_i, \tag{31.61}$$

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) and provided in the workfile "Strike.WF1".

The results from Poisson estimation of this model are presented below:

Dependent Variable: NUMB
Method: ML/QML - Poisson Count (Newton-Raphson / Marquardt steps)
Date: 03/09/15 Time: 16:43
Sample: 1 103
Included observations: 103
Convergence achieved after 3 iterations
Coefficient covariance computed using observed Hessian

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	LR statistic	16.84645
Avg. log likelihood	-2.762584	Prob(LR statistic)	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:

```
equation testeq.ls (numb-numb_f)^2-numb numb_f^2
```

yielding the following results:

Dependent Variable: (NUMB-NUMB_F)^2-NUMB
Method: Least Squares
Date: 03/09/15 Time: 16:38
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	Hannan-Quinn criter.	8.555269
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 (1997), 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: 08/12/09 Time: 10:55
Sample: 1 103
Included observations: 103

Variable	Coefficient	Std. Error	t-Statistic	Prob.
NUMB_F	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	987.3785	Schwarz criterion	5.143199
Log likelihood	-262.5574	Hannan-Quinn criter.	5.127980
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, 1984b)) using the estimate of $\hat{\eta}^2$ from the Wooldridge test derived above. To compute this estimator, simply select **Negative Binomial (QML)** and enter "0.221238" 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 choose the **GLM** option in the Covariance method dropdown menu. The estimation results are shown below:

Dependent Variable: NUMB
Method: QML - Negative Binomial Count (Newton-Raphson / Marquardt steps)
Date: 03/09/15 Time: 16:48
Sample: 1 103
Included observations: 103
QML parameter used in estimation: 0.22124
Convergence achieved after 4 iterations
Coefficient covariance computed using observed Hessian
GLM adjusted covariance (variance factor =0.961161659819)

Variable	Coefficient	Std. Error	z-Statistic	Prob.
C	1.724906	0.064023	26.94197	0.0000
IP	2.833103	1.198416	2.364039	0.0181
FEB	-0.369558	0.235617	-1.568474	0.1168

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	LR statistic	519.0330
Avg. log likelihood	-2.558066	Prob(LR statistic)	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 (1997), 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 currently 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. (Alternately, you may use the GLM estimators for count models described in Chapter 32. "Generalized Linear Models," on page 421, which do compute the QLR statistics automatically.)

Suppose that the estimated QML equation is named EQ1 and that the results are given by:

Dependent Variable: NUMB
Method: ML/QML - Poisson Count (Newton-Raphson / Marquardt steps)
Date: 03/09/15 Time: 21:42
Sample: 1 103
Included observations: 103
Convergence achieved after 3 iterations
Coefficient covariance computed using observed Hessian
GLM adjusted covariance (variance factor =2.22642047526)
d.f. adjustment for standard errors & covariance

Variable	Coefficient	Std. Error	z-Statistic	Prob.
C	1.725630	0.065140	26.49094	0.0000
IP	2.775334	1.222202	2.270766	0.0232
FEB	-0.377407	0.260405	-1.449307	0.1473

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	LR statistic	16.84645
Avg. log likelihood	-2.762584	Prob(LR statistic)	0.000220

Note that when you select the GLM robust standard errors, EViews reports the estimated, here d.f. corrected, variance factor. 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.@lrstat/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. 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

Default Standard Errors

The default standard errors are obtained by taking the inverse of the estimated information matrix. If you estimate your equation using a Newton-Raphson or Quadratic Hill Climbing method, EViews will use the inverse of the Hessian, $\hat{H}^{-1}$, to form your coefficient covariance estimate. If you employ BHHH, the coefficient covariance will be estimated using the inverse of the outer product of the scores $(\hat{g}\hat{g}')^{-1}$, where $\hat{g}$ and $\hat{H}$ are the gradient (or score) and Hessian of the log likelihood evaluated at the ML estimates.

Huber/White (QML) and Cluster-robust (QML) Standard Errors

The Huber/White options for robust standard errors computes the quasi-maximum likelihood (or pseudo-ML) standard errors:

$$\mathrm{var}_{QML}(\hat{\beta}) = (-\hat{H})^{-1}\hat{g}\hat{g}'(-\hat{H})^{-1}, \tag{31.62}$$

The Cluster-robust variants of these errors replace the inner matrix with a cluster-aware version of the moment.

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, but as with all QML estimation, caution is advised.

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 $x_i'\beta$:

$$\mathrm{E}(y_i | x_i, \beta) = h(x_i'\beta). \tag{31.63}$$

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:

$$\mathrm{var}(y_i | x_i, \beta) = \sigma^2 \mathrm{var}_{ML}(y_i | x_i, \beta). \tag{31.64}$$

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:

$$\mathrm{var}_{GLM}(\hat{\beta}) = \hat{\sigma}^2 \mathrm{var}_{ML}(\hat{\beta}), \tag{31.65}$$

where the d.f. corrected variance factor estimator is

$$\hat{\sigma}^2 = \frac{1}{N-K} \cdot \sum_{i=1}^{N} \frac{(y_i - \hat{y}_i)^2}{\sqrt{v(x_i, \hat{\beta}, \hat{\gamma})}} = \frac{1}{N-K} \cdot \sum_{i=1}^{N} \frac{\hat{u}_i^2}{\sqrt{(v(x_i, \hat{\beta}, \hat{\gamma}))}}. \tag{31.66}$$

If you do not choose to d.f. correct, the leading term in Equation (31.66) is $1/N$. When you select GLM standard errors, the estimated proportionality term $\hat{\sigma}^2$ is reported as the variance factor estimate in EViews.

(Note that the EViews legacy estimator always estimates a d.f. corrected variance factor, while the other estimators permit you to choose whether to override the default of no correction. Since the default behavior has changed, you will need to explicitly request d.f. correction to match the legacy covariance results.)

For detailed discussion on GLMs and the phenomenon of overdispersion, see McCullaugh and Nelder (1989).

The Hosmer-Lemeshow Test

Let the data be grouped into $j = 1, 2, \ldots, J$ 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:

$$\begin{aligned} y(j) &= \sum_{i \in j} y_i \\ \bar{p}(j) &= \sum_{i \in j} p_i / n_j = \sum_{i \in j} (1 - F(-x_i'\hat{\beta})) / n_j \end{aligned} \tag{31.67}$$

The Hosmer-Lemeshow test statistic is computed as:

$$HL = \sum_{j=1}^{J} \frac{(y(j) - n_j \bar{p}(j))^2}{n_j \bar{p}(j)(1 - \bar{p}(j))}. \tag{31.68}$$

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 $j = 1, 2, \ldots, J$ 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 Econometric*s, 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. (2008). *Econometric Analysis*, 6th 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.

Heckman, James (1976). "The Common Structure of Statistical Models of Truncation, Sample Selection, and Limited Dependent Variables and a Simple Estimator for Such Models," *Annals of Economic and Social Measurement,* 5, 475-492.

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, Peter, and J. A. Nelder (1989). *Generalized Linear Models, Second Edition*. London: Chapman & Hall.

McDonald, J. and R. Moffitt (1980). "The Uses of Tobit Analysis," *Review of Economic and Statistics*, 62, 318–321.

Mroz, Thomas (1987). "The Sensitivity of an Empirical Model of Married Women's Hours of Work to Economic and Statistical Assumptions", Econometrica, 55, 765-799.

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. (1997). "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 32. Generalized Linear Models

Nelder and McCullagh (1972) describe a class of *Generalized Linear Models* (GLMs) that extends linear regression to permit non-normal stochastic and non-linear systematic components. GLMs encompass a broad and empirically useful range of specifications that includes linear regression, logistic and probit analysis, and Poisson models.

GLMs offer a common framework in which we may place all of these specification, facilitating development of broadly applicable tools for estimation and inference. In addition, the GLM framework encourages the relaxation of distributional assumptions associated with these models, motivating development of robust *quasi-maximum likelihood* (QML) estimators and robust covariance estimators for use in these settings.

The following discussion offers an overview of GLMs and describes the basics of estimating and working with GLMs in EViews. Those wishing additional background and technical information are encouraged to consult one of the many excellent summaries that are available (McCullagh and Nelder 1989, Hardin and Hilbe 2007, Agresti 1990).

Overview

Suppose we have $i = 1, \ldots, N$ independent response variables Y_i, each of whose conditional mean depends on k-vectors of explanatory variables X_i and unknown coefficients β. We may decompose Y_i into a systematic mean component, μ_i, and a stochastic component ϵ_i

$$Y_i = \mu_i + \epsilon_i \tag{32.1}$$

The *conventional linear regression* model assumes that the μ_i is a linear predictor formed from the explanatory variables and coefficients, $\mu_i = X_i'\beta$, and that ϵ_i is normally distributed with zero mean and constant variance $V_i = \sigma^2$.

The GLM framework of Nelder and McCullagh (1972) generalizes linear regression by allowing the mean component μ_i to depend on a linear predictor through a nonlinear function, and the distribution of the stochastic component ϵ_i be any member of the linear exponential family. Specifically, a GLM specification consists of:

- A *linear predictor* or *index* $\eta_i = X_i'\beta + o_i$ where o_i is an optional *offset* term.
- A distribution for Y_i belonging to the linear exponential family.
- A smooth, invertible *link function,* $g(\mu_i) = \eta_i$, relating the mean μ_i and the linear predictor η_i.

A wide range of familiar models may be cast in the form of a GLM by choosing an appropriate distribution and link function. For example:

Model	Family	Link
Linear Regression	Normal	Identity: $g(\mu) = \mu$
Exponential Regression	Normal	Log: $g(\mu) = \log(\mu)$
Logistic Regression	Binomial	Logit: $g(\mu) = \log(\mu/(1-\mu))$
Probit Regression	Binomial	Probit: $g(\mu) = \Phi^{-1}(\mu)$
Poisson Count	Poisson	Log: $g(\mu) = \log(\mu)$

For a detailed description of these and other familiar specifications, see McCullagh and Nelder (1981) and Hardin and Hilbe (2007). It is worth noting that the GLM framework is able to nest models for continuous (normal), proportion (logistic and probit), and discrete count (Poisson) data.

Taken together, the GLM assumptions imply that the first two moments of Y_i may be written as functions of the linear predictor:

$$\begin{aligned} \mu_i &= g^{-1}(\eta_i) \\ V_i &= (\phi / w_i)\, V_\mu(g^{-1}(\eta_i)) \end{aligned} \tag{32.2}$$

where $V_\mu(\mu)$ is a distribution-specific variance function describing the mean-variance relationship, the *dispersion* constant $\phi > 0$ is a possibly known scale factor, and $w_i > 0$ is a known *prior weight* that corrects for unequal scaling between observations.

Crucially, the properties of the GLM maximum likelihood estimator depend only on these two moments. Thus, a GLM specification is principally a vehicle for specifying a mean and variance, where the mean is determined by the link assumption, and the mean-variance relationship is governed by the distributional assumption. In this respect, the distributional assumption of the standard GLM is overly restrictive.

Accordingly, Wedderburn (1974) shows that one need only specify a mean and variance specification as in Equation (32.2) to define a quasi-likelihood that may be used for coefficient and covariance estimation. Not surprisingly, for variance functions derived from exponential family distributions, the likelihood and quasi-likelihood functions coincide. McCullagh (1983) offers a full set of distributional results for the quasi-maximum likelihood (QML) estimator that mirror those for ordinary maximum likelihood.

QML estimators are an important tool for the analysis of GLM and related models. In particular, these estimators permit us to estimate GLM-like models involving mean-variance specifications that extend beyond those for known exponential family distributions, and to estimate models where the mean-variance specification is of exponential family form, but

the observed data do not satisfy the distributional requirements (Agresti 1990, 13.2.3 offers a nice non-technical overview of QML).

Alternately, Gourioux, Monfort, and Trognon (1984) show that consistency of the GLM maximum likelihood estimator requires only correct specification of the conditional mean. Misspecification of the variance relationship does, however, lead to invalid inference, though this may be corrected using robust coefficient covariance estimation. In contrast to the QML results, the robust covariance correction does not require correction specification of a GLM conditional variance.

How to Estimate a GLM in EViews

To estimate a GLM model in EViews you must first create an equation object. You may select **Object/New Object…/Equation** or **Quick/Estimate Equation…** from the main menu, or enter the keyword `equation` in the command window. Next select **GLM - Generalized Linear Model** in the **Method** dropdown menu. Alternately, entering the keyword `glm` in the command window will both create the object and automatically set the estimation method. The dialog will change to show settings appropriate for specifying a GLM.

Specification

The main page of the dialog is used to describe the basic GLM specification.

We will focus attention on the **GLM Equation specification** section since the **Estimation settings** section in the bottom of the dialog is should be self-explanatory.

Dependent Variable and Linear Predictor

In the main edit field you should specify your dependent variable and the linear predictor.

There are two ways in which you may enter this information. The easiest method is to list the dependent response variable followed by all of the regressors that enter into the predictor. PDL specifications are permitted in this list, but ARMA terms are not. If you wish to include an offset in your predictor, it should be entered on the **Options** page (see "Specification Options" on page 425).

Alternately, you may enter an explicit *linear* specification like "Y = C(1) + C(2)*X". The response variable will be taken to be the variable on the left-hand side of the equality ("Y") and the linear predictor will be taken from the right-hand side of the expression ("C(1) + C(2)*X"). Offsets may be entered directly in the expression or they may be entered on the **Options** page. Note that this specification should not be taken as a literal description of the mean equation; it is merely a convenient syntax for specifying both the response and the linear predictor.

Family

Next, you should use the **Family** dropdown to specify your distribution. The default family is the **Normal** distribution, but you are free to choose from the list of linear exponential family and quasi-likelihood distributions. Note that the last three entries (**Exponential Mean**, **Power Mean (p)**, **Binomial Squared**) are for quasi-likelihood specifications not associated with exponential families.

If the selected distribution requires specification of an ancillary parameter, you will be prompted to provide the values. For example, the **Binomial Count** and **Binomial Proportion** distributions both require specification of the number of trials n_i, while the **Negative Binomial** requires specification of the excess-variance parameter k_i.

For descriptions of the various exponential and quasi-likelihood families, see "Distribution," beginning on page 439.

Link

Lastly, you should use the **Link** dropdown to specify a link function.

EViews will initialize the **Link** setting to the default for to the selected family. In general, the canonical link is used as the default link, however, the **Log** link is used as the default for the **Negative Binomial** family. The **Exponential Mean**, **Power Mean (p)**, and **Binomial Squared** quasi-likelihood families will default to use the **Identity**, **Log**, and **Logit** links, respectively.

If the link that you select requires specification of parameter values, you will be prompted to enter the values.

For detailed descriptions of the link functions, see "Link," beginning on page 441.

Options

Click on the **Options** tab to display additional settings for the GLM specification. You may use this page to augment the equation specification, to choose a dispersion estimator, to specify the estimation algorithm and associated settings, or to define a coefficient covariance estimator.

Specification Options

The **Specification Options** section of the **Options** tab allows you to augment the GLM specification.

To include an offset in your linear predictor, simply enter a series name or expression in the **Offset** edit field.

The **Frequency weights** edit field should be used to specify replicates for each observation in the workfile. In practical terms, the frequency weights act as a form of variance weighting and inflate the number of "observations" associated with the data records.

You may also specify prior variance weights in the using the **Weights** dropdown and associated edit fields. To specify your weights, simply select a description for the form of the weighting series (**Inverse std. dev.**, **Inverse variance**, **Std. deviation**, **Variance**), then enter the corresponding weight series name or expression. EViews will translate the values in the weighting series into the appropriate values for w_i. For example, to specify w_i directly, you should select **Inverse variance** then enter the series or expression containing the w_i values. If you instead choose **Variance**, EViews will set w_i to the inverse of the values in the weight series. "Weighted Least Squares" on page 47 for additional discussion.

Dispersion Options

The **Method** dropdown may be used to select the dispersion computation method. You will always be given the opportunity to choose between the **Default** setting or **Pearson Chi-Sq.**, **Fixed at 1**, and **User-Specified**. Additionally, if the specified distribution is in the linear exponential family, you may choose to use the **Deviance** statistic.

The **Default** entry instructs EViews to use the default method for computing the dispersion, which will depend on the specified family. For families with a free dispersion parameter, the default is to use the **Pearson Chi-Sq.** statistic, otherwise the default is **Fixed at 1**. The current default setting will be displayed directly below the dropdown.

Coefficient Covariance Options

The **Covariance method** dropdown specifies the estimator for the coefficient covariance matrix. You may choose between the **Ordinary** method, which uses the inverse of the estimated information matrix, or you may elect to use **Huber/White** sandwich estimator, the heteroskedasticity and auto-correlation consistent **HAC (Newey-West)** approach, or a Cluster-robust QML sandwich estimator.

If you select the HAC covariance method, a HAC options button will appear prompting so that you may customize the whitening and kernel settings. By default, EViews HAC estimation will employ a Bartlett kernel with fixed Newey-West sample-size based bandwidth and no pre-whitening (see "HAC Consistent Covariances (Newey-West)" on page 45 for additional discussion).

If you select the cluster-robust method, an edit field will prompt you for the cluster identifier variable.

The **Information matrix** dropdown allows you to specify the method for estimating the information matrix. For covariances computed in the standard fashion, you may choose between the default **Hessian - observed**, **Hessian - expected**, and **OPG - BHHH**. If you are computing Huber/White covariances, only the two Hessian based selections will be displayed.

(Note that in some earlier versions of EViews, the information matrix default method was tied in with the optimization method. This default dependence has been eliminated.)

Lastly you may use the **d.f. Adjustment** checkbox choose whether to apply a degree-of-freedom correction to the coefficient covariance. By default, EViews will perform this adjustment.

Estimation Options

The **Estimation** section of the page lets you specify the optimization algorithm, starting values, and other estimation settings.

The **Optimization Algorithm** and **Step method** dropdown menus control your estimation method.

- The default **Optimization Algorithm** is **Newton-Raphson**, but you may instead select **BFGS**, **OPG - BHHH**, **Fisher Scoring** (IRLS), or **EViews legacy**.
- The default **Step method** is **Marquardt**, but you may use the menu to select **Dogleg** or **Line search**.

If you select optimization using **EViews legacy**, you will be prompted to select a legacy method in place of a step method. The **Legacy method** dropdown offers the choice of the default **Quadratic Hill Climbing** (Newton-Raphson with Marquardt steps), **Newton-Raphson** with line search, **IRLS - Fisher Scoring**, and **BHHH** (OPG with line search).

By default, the **Starting Values** dropdown is set to **EViews Supplied**. The EViews default starting values for β are obtained using the suggestion of McCullagh and Nelder to initialize the IRLS algorithm at $\hat{\mu}_i = (n_i y_i + 0.5)/(n_i + 1)$ for the binomial proportion family, and $\hat{\mu}_i = (y_i + \bar{y})/2$ otherwise, then running a single IRLS coefficient update to obtain the initial β. Alternately, you may specify starting values that are a fraction of the default values, or you may instruct EViews to use your own values.

You may use the **IRLS iterations** edit field to instruct EViews to perform a fixed number of additional IRLS updates to refine coefficient values prior to starting the specified estimation algorithm.

The **Max Iterations** and **Convergence** edit fields are self-explanatory. Selecting the **Display settings** checkbox instructs EViews to show detailed information on tolerances and initial values in the equation output.

Coefficient Name

You may use the Coefficient name section of the dialog to change the coefficient vector from the default C. EViews will create and resize the vector if necessary.

Examples

In this section, we offer three examples illustrating GLM estimation in EViews.

Exponential Regression

Our first example uses the Kennen (1983) dataset ("Strike.WF1") on number of strikes (NUMB), industrial production (IP), and dummy variable representing the month of February (FEB). To account for the non-negative response variable NUMB, we may estimate a nonlinear specification of the form:

$$\mathrm{NUMB}_i = \exp(\beta_1 + \beta_2 \mathrm{IP}_i + \beta_3 \mathrm{FEB}_i) + \epsilon_i \tag{32.3}$$

where $\epsilon_i \sim N(0, \sigma^2)$. This model falls into the GLM framework with a log link and normal family. To estimate this specification, bring up the GLM dialog and fill out the equation specification page as follows:

```
numb c ip feb
```

then change the **Link function** to **Log**. For the moment, we leave the remaining settings and those on the **Options** page at their default values. Click on **OK** to accept the specification and estimate the model. EViews displays the following results:

Dependent Variable: NUMB
Method: Generalized Linear Model (Newton-Raphson / Marquardt steps)
Included observations: 103
Family: Normal
Link: Log
Dispersion computed using Pearson Chi-Square
Convergence achieved after 6 iterations
Coefficient covariance computed using observed Hessian

Variable	Coefficient	Std. Error	z-Statistic	Prob.
C	1.727368	0.066206	26.09097	0.0000
IP	2.664874	1.237904	2.152732	0.0313
FEB	-0.391015	0.313445	-1.247476	0.2122

Mean dependent var	5.495146	S.D. dependent var	3.653829
Sum squared resid	1273.783	Log likelihood	-275.6964
Akaike info criterion	5.411580	Schwarz criterion	5.488319
Hannan-Quinn criter.	5.442662	Deviance	1273.783
Deviance statistic	12.73783	Restr. deviance	1361.748
LR statistic	6.905754	Prob(LR statistic)	0.031654
Pearson SSR	1273.783	Pearson statistic	12.73783
Dispersion	12.73783		

The top portion of the output displays the estimation settings and basic results, in particular the choice of algorithm (Newton-Raphson with Marquardt steps), distribution family (Normal), and link function (Log), as well as the dispersion estimator, coefficient covariance estimator, and estimation status. We see that the dispersion estimator is based on the Pearson χ^2 statistic and the coefficient covariance is computed using the inverse of the (negative of the) observed Hessian.

The coefficient estimates indicate that IP is positively related to the number of strikes, and that the relationship is statistically significant at conventional levels. The FEB dummy variable is negatively related to NUMB, but the relationship is not statistically significant.

The bottom portion of the output displays various descriptive statistics. Note that in place of some of the more familiar statistics, EViews reports the deviance, deviance statistic (deviance divided by the degrees-of-freedom) restricted deviance (for the model with only a constant), and the corresponding LR test statistic and probability. The test indicates that the IP and FEB variables are jointly significant at roughly the 3% level. Also displayed are the sum-of-squared Pearson residuals and the estimate of the dispersion, which in this example is the Pearson statistic.

It may be instructive to examine the representations view of this equation. Simply go to the equation toolbar or the main menu and click on **View/Representations** to display the view.

Notably, the representations view displays both the specification of the linear predictor (I_NUMB) as well as the mean specification (EXP(I_NUMB)) in terms of the EViews coefficient names, and in terms of the estimated values. These are the expressions used when forecasting the index or the dependent variable using the **Forecast** procedure (see "Forecasting" on page 436).

Binomial

We illustrate the estimation of GLM binomial logistic regression using a simple example from Agresti (2007, Table 3.1, p. 69) examining the relationship between snoring and heart disease. The data in the first page of the workfile "Snoring.WF1" consist of grouped binomial response data for 2,484 subjects divided into four risk factor groups for snoring level (SNORING), coded as 0, 2, 4, 5. Associated with each of the four groups is the number of individuals in the group exhibiting heart disease (DISEASE) as well as a total group size (TOTAL).

SNORING	DISEASE	TOTAL
0	24	1379
2	35	638
4	21	213
5	30	254

We may estimate a logistic regression model for these data in either raw frequency or proportions form.

To estimate the model in raw frequency form, bring up the GLM equation dialog, enter the linear predictor specification:

```
disease c snore
```

select **Binomial Count** in the **Family** dropdown, and enter “TOTAL” in the **Number of trials** edit field. Next switch over to the **Options** page and turn off the **d.f. Adjustment** for the coefficient covariance. Click on **OK** to estimate the equation.

```
Dependent Variable: DISEASE
Method: Generalized Linear Model (Newton-Raphson / Marquardt
   steps)
Date: 03/10/15   Time: 15:19
Sample: 1 4
Included observations: 4
Family: Binomial Count (n = TOTAL)
Link: Logit
Dispersion fixed at 1
Summary statistics are for the binomial proportions and implicit
      variance weights used in estimation
Convergence achieved after 2 iterations
Coefficient covariance computed using observed Hessian
No d.f. adjustment for standard errors & covariance
```

The output header shows relevant information for the estimation procedure. Note in particular the EViews message that summary statistics are computed for the binomial proportions data. This message is a hint at the fact that EViews estimates the binomial count model by scaling the dependent variable by the number of trials, and estimating the corresponding proportions specification.

Accordingly, you could have specified the model in proportions form. Simply enter the linear predictor specification:

```
disease/total c snoring
```

with **Binomial Proportions** specified in the **Family** dropdown and “TOTAL” entered in the **Number of trials** edit field.

Dependent Variable: DISEASE/TOTAL
Method: Generalized Linear Model (Newton-Raphson / Marquardt steps)
Date: 03/10/15 Time: 15:21
Sample: 1 4
Included observations: 4
Family: Binomial Proportion (trials = TOTAL)
Link: Logit
Dispersion fixed at 1
Convergence achieved after 2 iterations
Coefficient covariance computed using observed Hessian
No d.f. adjustment for standard errors & covariance

Variable	Coefficient	Std. Error	z-Statistic	Prob.
C	-3.866248	0.166214	-23.26061	0.0000
SNORING	0.397337	0.050011	7.945039	0.0000

Mean dependent var	0.023490	S.D. dependent var	0.001736
Sum squared resid	0.000357	Log likelihood	-11.53073
Akaike info criterion	6.765367	Schwarz criterion	6.458514
Hannan-Quinn criter.	6.092001	Deviance	2.808912
Deviance statistic	1.404456	Restr. deviance	65.90448
LR statistic	63.09557	Prob(LR statistic)	0.000000
Pearson SSR	2.874323	Pearson statistic	1.437162
Dispersion	1.000000		

The top portion of the output changes to show the different settings, but the remaining output is identical. In particular, there is strong evidence that SNORING is related to heart disease in these data, with the estimated probability of heart disease increasing with the level of snoring.

It is worth mentioning that data of this form are sometimes represented in a frequency weighted form in which the data each group is divided into two records, one for the binomial successes, and one for the failures. Each each record contains the number of repeats in the group and a binary indicator for success (the total number of records is G, where G is the number of groups) The FREQ page of the “Snoring.WF1” workfile contains the data represented in this fashion:

SNORING	DISEASE	N
0	1	24
2	1	35
4	1	21
5	1	30
0	0	1355

2	0	603
4	0	192
5	0	224

In this representation, DISEASE is an indicator for whether the record corresponds to individuals with heart disease or not, and N is the number of individuals in the category.

Estimation of the equivalent GLM model specified using the frequency weighted data is straightforward. Simply enter the linear predictor specification:

```
disease c snoring
```

with either **Binomial Proportions** or **Binomial Count** specified in the **Family** dropdown. Since each observation corresponds to a binary indicator, you should enter "1" enter as the **Number of trials** edit field. The multiple individuals in the category are handled by entering "N" in the **Frequency weights** field in the **Options** page.

Dependent Variable: DISEASE
Method: Generalized Linear Model (Newton-Raphson / Marquardt steps)
Date: 03/10/15 Time: 15:16
Sample: 1 8
Included cases: 8
Total observations: 2484
Family: Binomial Count (n = 1)
Link: Logit
Frequency weight series: N
Dispersion fixed at 1
Convergence achieved after 6 iterations
Coefficient covariance computed using observed Hessian
No d.f. adjustment for standard errors & covariance

Variable	Coefficient	Std. Error	z-Statistic	Prob.
C	-3.866248	0.166214	-23.26061	0.0000
SNORING	0.397337	0.050011	7.945039	0.0000

Mean dependent var	0.044283	S.D. dependent var	0.205765
Sum squared resid	102.1917	Log likelihood	-418.8658
Akaike info criterion	0.338861	Schwarz criterion	0.343545
Hannan-Quinn criter.	0.340562	Deviance	837.7316
Deviance statistic	0.337523	Restr. deviance	900.8272
LR statistic	63.09557	Prob(LR statistic)	0.000000
Pearson SSR	2412.870	Pearson statistic	0.972147
Dispersion	1.000000		

Note that while a number of the summary statistics differ due to the different representation of the data (notably the Deviance and Pearson SSRs), the coefficient estimates and LR test statistics in this case are identical to those outlined above. There will, however, be substan-

tive differences between the two results in settings when the dispersion is estimated since the effective number of observations differs in the two settings.

Lastly the data may be represented in individual trial form, which expands observations for each trial in the group into a separate record. The total number of records in the data is $\sum n_i$, where n_i is the number of trials in the i-th (of G) group. This representation is the traditional ungrouped binary response form for the data. Results for data in this representation should match those for the frequency weighted data.

Binomial Proportions

Papke and Wooldridge (1996) apply GLM techniques to the analysis of fractional response data for 401K tax advantaged savings plan participation rates ("401kjae.WF1"). Their analysis focuses on the relationship between plan participation rates (PRATE) and the employer matching contribution rates (MRATE), accounting for the log of total employment (LOG(TOTEMP), LOG(TOTEMP)^2), plan age (AGE, AGE^2), and a binary indicator for whether the plan is the only pension plan offered by the plan sponsor (SOLE).

We focus on two of the equations estimated in the paper. In both, the authors employ a GLM specification using a binomial proportion family and logit link. Information on the binomial group size n_i is ignored, but variance misspecification is accounted for in two ways: first using a binomial QMLE with GLM standard errors, and second using the robust Huber-White covariance approach.

To estimate the GLM standard error specification, we first call up the GLM dialog and enter the linear predictor specification:

```
prate mprate log(totemp) log(totemp)^2 age age^2 sole
```

Next, select the **Binomial Proportion** family, and enter the sample description

```
@all if mrate<=1
```

Lastly, we leave the **Number of trials** edit field at the default value of 1, but correct for heterogeneity by going to the **Options** page and specifying **Pearson Chi-Sq.** dispersion estimates. Click on **OK** to continue.

The resulting estimates correspond the coefficient estimates and first set of standard errors in Papke and Wooldridge (Table II, column 2):

Dependent Variable: PRATE
Method: Generalized Linear Model (Newton-Raphson / Marquardt steps)
Date: 03/10/15 Time: 15:26
Sample: 1 4735 IF MRATE<=1
Included observations: 3784
Family: Binomial Proportion (trials = 1) Quasi-likelihood
Link: Logit
Dispersion computed using Pearson Chi-Square
Convergence achieved after 4 iterations
Coefficient covariance computed using observed Hessian

Variable	Coefficient	Std. Error	z-Statistic	Prob.
MRATE	1.390080	0.100368	13.84981	0.0000
LOG(TOTEMP)	-1.001875	0.111222	-9.007914	0.0000
LOG(TOTEMP)^2	0.052186	0.007105	7.345545	0.0000
AGE	0.050113	0.008710	5.753136	0.0000
AGE^2	-0.000515	0.000211	-2.444532	0.0145
SOLE	0.007947	0.046785	0.169860	0.8651
C	5.057998	0.426942	11.84703	0.0000

Mean dependent var	0.847769	S.D. dependent var	0.169961
Sum squared resid	92.69516	Quasi-log likelihood	-8075.397
Deviance	765.0353	Deviance statistic	0.202551
Restr. deviance	895.5505	Quasi-LR statistic	680.4838
Prob(Quasi-LR stat)	0.000000	Pearson SSR	724.4200
Pearson statistic	0.191798	Dispersion	0.191798

Papke and Wooldridge offer a detailed analysis of the results (p. 628–629), which we will not duplicate here. We will point out that the estimate of the dispersion (0.191798) taken from the Pearson statistic is far from the restricted value of 1.0.

The results using the QML with GLM standard errors rely on validity of the GLM assumption for the variance given in Equation (32.2), an assumption that may be too restrictive. We may instead estimate the equation without imposing a particular conditional variance specification by computing our estimates using a robust Huber-White sandwich method. Click on **Estimate** to bring up the equation dialog, select the **Options** tab, then change the **Covariance method** from **Default** to **Huber/White**. Click on **OK** to estimate the revised specification:

Dependent Variable: PRATE
Method: Generalized Linear Model (Newton-Raphson / Marquardt steps)
Date: 03/10/15 Time: 15:27
Sample: 1 4735 IF MRATE<=1
Included observations: 3784
Family: Binomial Proportion (trials = 1)
Link: Logit
Dispersion fixed at 1
Convergence achieved after 5 iterations
Coefficient covariance computed using the Huber-White method with observed Hessian

Variable	Coefficient	Std. Error	z-Statistic	Prob.
MRATE	1.390080	0.107792	12.89596	0.0000
LOG(TOTEMP)	-1.001875	0.110524	-9.064757	0.0000
LOG(TOTEMP)^2	0.052186	0.007134	7.315681	0.0000
AGE	0.050113	0.008852	5.661091	0.0000
AGE^2	-0.000515	0.000212	-2.432326	0.0150
SOLE	0.007947	0.050242	0.158172	0.8743
C	5.057998	0.421199	12.00858	0.0000

Mean dependent var	0.847769	S.D. dependent var	0.169961
Sum squared resid	92.69516	Log likelihood	-1179.279
Akaike info criterion	0.626997	Schwarz criterion	0.638538
Hannan-Quinn criter.	0.631100	Deviance	765.0353
Deviance statistic	0.202551	Restr. deviance	895.5505
LR statistic	130.5153	Prob(LR statistic)	0.000000
Pearson SSR	724.4200	Pearson statistic	0.191798
Dispersion	1.000000		

EViews reports the new method of computing the coefficient covariance in the header. The coefficient estimates are unchanged, since the alternative computation of the coefficient covariance is a post-estimation procedure, and the new standard estimates correspond the second set of standard errors in Papke and Wooldridge (Table II, column 2). Notably, the use of an alternative estimator for the coefficient covariance has little substantive effect on the results.

Working with a GLM Equation

EViews offers various views and procedures for a estimated GLM equation. Some, like the **Gradient Summary** or the coefficient **Covariance Matrix** view are self-explanatory. In this section, we offer relevant comment on the remaining views.

Residuals

The main equation output offers summary statistics for the sum-of-squared response residuals (“Sum squared resid”), and the sum-of-squared Pearson residuals (“Pearson SSR”).

The **Actual, Fitted, Residual** views and **Residual Diagnostics** allow you to examine properties of your residuals. The **Actual, Fitted, Residual Table** and **Graph**, show the fit of the unweighted data. As the name suggests, the **Standardized Residual Graph** displays the standardized (scaled Pearson) residuals.

The **Residual Diagnostics** show **Histograms** of the standardized residuals and **Correlograms** of the standardized residuals and the squared standardized residuals.

The **Make Residuals** proc allows you to save the **Ordinary** (response), **Standardized** (scaled Pearson), or **Generalized** (score) residuals into the workfile. The latter may be useful for constructing test statistics (note, however, that in some cases, it may be more useful to compute the gradients of the model directly using **Proc/Make Gradient Group**).

Given standardized residuals SRES for equation EQ1, the unscaled Pearson residuals may be obtained using the command

```
series pearson = sres * @sqrt(eq1.@dispersion)
```

Forecasting

EViews offers built-in tools for producing in and out-of-sample forecasts (fits) from your GLM estimated equation. Simply click on the **Forecast** button on your estimated equation to bring up the forecast dialog, then enter the desired settings.

You should first use the radio buttons to specify whether you wish to forecast the expected dependent variable μ_i or the linear index η_i.

Next, enter the name of the series to hold the forecast output, and set the forecast sample.

Lastly, specify whether you wish to produce a forecast graph and whether you wish to fill non-forecast values in the workfile with actual values or to fill them with NAs. For most cross-section applications, we recommend that you uncheck this box.

Click on **OK** to produce the forecast.

Note that while EViews does not presently offer a menu item for saving the fitted GLM variances or scaled variances, you can easily obtain results by saving the ordinary and standardized residuals and taking ratios ("Residuals" on page 448). If ORESID are the ordinary and SRESID are the standardized residuals for equation EQ1, then the commands

```
series glmsvar = (oresid / sresid)^2
series glmvar = glmvar * eq1.@dispersion
```

produce the scaled variance and unscaled variances, respectively.

Lastly, you should use **Proc/Make Model** to create a model object for more complicated simulation from your GLM equation.

Testing

You may perform Wald tests of coefficient restrictions. Simply select **View/Coefficient Diagnostics/Wald - Coefficient Restrictions**, then enter your restrictions in the edit field. For the Papke-Wooldridge example above with Huber-White robust covariances, we may use a Wald test to evaluate the joint significance of AGE^2 and SOLE by entering the restriction "C(5) = C(6) = 0" and clicking on OK to perform the test.

Wald Test:
Equation: EQ2_QMLE_R
Null Hypothesis: C(5)=C(6)=0

Test Statistic	Value	df	Probability
F-statistic	2.970226	(2, 3777)	0.0514
Chi-square	5.940451	2	0.0513

Null Hypothesis Summary:

Normalized Restriction (= 0)	Value	Std. Err.
C(5)	-0.000515	0.000212
C(6)	0.007947	0.050242

Restrictions are linear in coefficients.

The test results show joint-significance at just above the 5% level. The **Confidence Intervals** and **Confidence Ellipses...** views will also employ the robust covariance matrix estimates.

The **Omitted Variables...** and **Redundant Variables...** views and the **Ramsey RESET Test...** views are likelihood ratio based tests. Note that the RESET test is a special case of an omitted variables test where the omitted variables are powers of the fitted values from the original equation.

We illustrate these tests by performing the RESET test on the first Papke-Wooldridge QMLE equation with GLM covariances. Select **View/Stability Diagnostics/Ramsey Reset Test...** and change the default to include 2 fitted terms in the test equation.

Ramsey RESET Test
Equation: EQ2_QMLE
Specification: PRATE MRATE LOG(TOTEMP) LOG(TOTEMP)^2 AGE AGE^2 SOLE C
Omitted Variables: Powers of fitted values from 2 to 3

	Value	df	Probability
F-statistic	0.311140	(2, 3775)	0.7326
QLR* statistic	0.622280	2	0.7326

F-test summary:

	Sum of Sq.	df	Mean Squares
Test Deviance	0.119389	2	0.059694
Restricted Deviance	765.0353	3777	0.202551
Unrestricted Deviance	764.9159	3775	0.202627
Dispersion SSR	724.2589	3775	0.191857

QLR* test summary:

	Value	df
Restricted Deviance	765.0353	3777
Unrestricted Deviance	764.9159	3775
Dispersion	0.191857	

The top portion of the output shows the test settings, and the test summaries. The bottom portion of the output shows the estimated test equation. The results show little evidence of nonlinearity.

Notice that in contrast to LR tests in most other equation views, the likelihood ratio test statistics in GLM equations are obtained from analysis of the deviances or quasi-deviances. Suppose D_0 is the unscaled deviance under the null and D_1 is the corresponding statistic under the alternative hypothesis. The usual asymptotic χ^2 likelihood ratio test statistic may be written in terms of the difference of deviances with common scaling,

$$\frac{D_0 - D_1}{\hat{\phi}} \sim \chi_r^2 \tag{32.4}$$

as $N \to \infty$, where $\hat{\phi}$ is an estimate of the dispersion and r is the fixed number of restrictions imposed by the null hypothesis. $\hat{\phi}$ is either a specified fixed value or an estimate under the alternative hypothesis using the specified dispersion method. When D_0 and D_1 contain the quasi-deviances, the resulting statistic is the quasi-likelihood ratio (QLR) statistic (Wooldridge, 1997).

If ϕ is estimated, we may also employ the F-statistic variant of the test statistic:

$$\frac{(D_0 - D_1)/r}{\hat{\phi}} \sim F_{r,\, N-p} \tag{32.5}$$

where $N - p$ is the degrees-of-freedom under the alternative and $\hat{\phi}$ is an estimate of the dispersion. EViews will estimate $\hat{\phi}$ under the alternative hypothesis using the method specified in your equation.

We point out that the Ramsey test results (and all other GLM LR test statistics) presented here may be problematic since they rely on the GLM variance assumption, Papke and Wooldridge offer a robust LM formulation for the Ramsey RESET test. This test is not currently built-into EViews, but which may be constructed with some effort using auxiliary results provided by EViews (see Papke and Wooldridge, p. 625 for details on the test construction).

Technical Details

The following discussion offers a brief technical summary of GLMs, describing specification, estimation, and hypothesis testing in this framework. Those wishing greater detail should consult the McCullagh and Nelder's (1989) monograph or the book-length survey by Hardin and Hilbe (2007).

Distribution

A GLM assumes that Y_i are independent random variables following a linear exponential family distribution with density:

$$f(y_i, \theta_i, \phi, w_i) = \exp\left(\frac{y_i\theta_i - b(\theta_i)}{\phi / w_i} + c(y_i, \phi, w_i)\right) \tag{32.6}$$

where b and c are distribution specific functions. $\theta_i = \theta(\mu_i)$, which is termed the *canonical parameter*, fully parameterizes the distribution in terms of the conditional mean, the *dispersion* value ϕ is a possibly known scale nuisance parameter, and w_i is a known *prior weight* that corrects for unequal scaling between observations with otherwise constant ϕ.

The exponential family assumption implies that the mean and variance of Y_i may be written as

$$\begin{aligned} E(Y_i) &= b'(\theta_i) = \mu_i \\ Var(Y_i) &= (\phi / \boldsymbol{w_i})\, b''(\theta_i) = (\phi / \boldsymbol{w_i})\, V_\mu(\mu_i) \end{aligned} \tag{32.7}$$

where $b'(\theta_i)$ and $b''(\theta_i)$ are the first and second derivatives of the b function, respectively, and V_μ is a distribution-specific variance function that depends only on μ_i.

EViews supports the following exponential family distributions:

Family	θ_i	$b(\theta_i)$	V_μ	ϕ
Normal	μ_i	$\theta_i^2/2$	1	σ^2
Gamma	$-1/\mu_i$	$-\log(-\theta_i)$	μ^2	ν

Inverse Gaussian	$-1/(2\mu_i^2)$	$-(-2\theta)^{1/2}$	μ^3	λ
Poisson	$\log(\mu_i)$	e^{θ_i}	μ	1
Binomial Proportion (n_i trials)	$\log\left(\frac{p_i}{1-p_i}\right)$	$\log(1+e^{\theta_i})$	$\mu(1-\mu_i)$	1
Negative Binomial (k_i is known)	$\log\left(\frac{k_i\mu_i}{1+k_i\mu_i}\right)$	$\frac{-\log(1-e^{\theta_i})}{k_i}$	$\mu(1+k_i\mu)$	1

The corresponding density functions for each of these distributions are given by:

- Normal

$$f(y_i, \mu_i, \sigma^2, w_i) = (2\pi\sigma^2/w_i)^{-1/2}\exp\left(\frac{-(y_i^2 - 2y_i\mu_i + \mu_i^2)}{2\sigma^2/w_i}\right) \tag{32.8}$$

for $-\infty < y_i < \infty$.

- Gamma

$$f(y_i, \mu_i, r_i) = \frac{(y_i r_i/\mu_i)^{r_i}\exp(-y_i/(\mu_i/r_i))}{y_i\Gamma(r_i)} \tag{32.9}$$

for $y_i > 0$ where $r_i = w_i/\nu$.

- Inverse Gaussian

$$f(y_i, \mu_i, \lambda, w_i) = (2\pi y_i^3\lambda/w_i)^{-1/2}\exp\left(\frac{-(y_i-\mu_i)^2}{2y_i\mu_i^2(\lambda/w_i)}\right) \tag{32.10}$$

for $y_i > 0$.

- Poisson

$$f(y_i, \mu_i) = \frac{\mu_i^{y_i}\exp(-\mu_i)}{y_i!} \tag{32.11}$$

for $y_i = 0, 1, 2, \ldots$ The dispersion is restricted to be 1 and prior weighting is not permitted.

- Binomial Proportion

$$f(y_i, n_i, \mu_i) = \binom{n_i}{n_i y_i}\mu_i^{n_i y_i}(1-\mu_i)^{n_i(1-y_i)} \tag{32.12}$$

for $0 \le y_i \le 1$ where $n_i = 1, 2, \ldots$ is the number of binomial trials. The dispersion is restricted to be 1 and the prior weights $w_i = n_i$.

- Negative Binomial

$$f(y_i, \mu_i, k_i) = \frac{\Gamma(y_i + 1/k_i)}{\Gamma(y_i + 1)\Gamma(1/k_i)} \left(\frac{k_i \mu_i}{1 + k_i \mu_i}\right)^{y_i} \left(\frac{1}{1 + k_i \mu_i}\right)^{1/k_i} \tag{32.13}$$

for $y_i = 0, 1, 2, \ldots$ The dispersion is restricted to be 1 and prior weighting is not permitted.

In addition, EViews offers support for the following quasi-likelihood families:

Quasi-Likelihood Family	V_μ
Poisson	μ
Binomial Proportion	$\mu(1-\mu)$
Negative Binomial (k)	$\mu(1+k\mu)$
Power Mean (r)	μ^r
Exponential Mean	ϵ^μ
Binomial Squared	$\mu^2(1-\mu)^2$

The first three entries in the table correspond to overdispersed or prior weighted versions of the specified distribution. The last three entries are pure quasi-likelihood distributions that do not correspond to exponential family distributions. See "Quasi-likelihoods," beginning on page 443 for additional discussion.

Link

The following table lists the names, functions, and associated range restrictions for the supported links:

Name	**Link Function** $g(\mu)$	**Range of** μ
Identity	μ	$(-\infty, \infty)$
Log	$\log(\mu)$	$(0, \infty)$
Log-Complement	$\log(1-\mu)$	$(-\infty, 1)$
Logit	$\log(\mu/(1-\mu))$	$(0, 1)$
Probit	$\Phi^{-1}(\mu)$	$(0, 1)$

Log-Log	$-\log(-\log(\mu))$	$(0, 1)$
Complementary Log-Log	$\log(-\log(1-\mu))$	$(0, 1)$
Inverse	$1/\mu$	$(-\infty, \infty)$
Power (p)	μ^p if $p \neq 0$ $\log(\mu)$ if $p = 0$	$(0, \infty)$
Power Odds Ratio (p)	$(\mu/(1-\mu))^p$ if $p \neq 0$ $\log(\mu/(1-\mu))$ if $p = 0$	$(0, 1)$
Box-Cox (p)	$(\mu^p - 1)/p$ if $p \neq 0$ $\log(\mu)$ if $p = 0$	$(0, \infty)$
Box-Cox Odds Ratio (p)	$((\mu/(1-\mu))^p - 1)/p$ if $p \neq 0$ $\log(\mu/(1-\mu))$ if $p = 0$	$(0, 1)$

EViews does not restrict the link choices associated with a given distributional family. Thus, it is possible for you to choose a link function that returns invalid mean values for the specified distribution at some parameter values, in which case your likelihood evaluation and estimation will fail.

One important role of the inverse link function is to map the real number domain of the linear index into the range of the dependent variable. Consequently the choice of link function is often governed in part by the desire to enforce range restrictions on the fitted mean. For example, the mean of a binomial proportions or negative binomial model must be between 0 and 1, while the Poisson and Gamma distributions require a positive mean value. Accordingly, the use of a Logit, Probit, Log-Log, Complementary Log-Log, Power Odds Ratio, or Box-Cox Odds Ratio is common with a binomial distribution, while the Log, Power, and Box-Cox families are generally viewed as more appropriate for Poisson or Gamma distribution data.

EViews will default to use the *canonical link* for a given distribution. The canonical link is the function that equates the canonical parameter θ of the exponential family distribution and the linear predictor $\eta = g(\mu) = \theta(\mu)$. The canonical links for relevant distributions are given by:

Family	**Canonical Link**
Normal	Identity
Gamma	Inverse
Inverse Gaussian	Power $(p = -2)$

Poisson	Log
Binomial Proportion	Logit

The negative binomial canonical link is not supported in EViews so the log link is used as the default choice in this case. We note that while the canonical link offers computational and conceptual convenience, it is not necessarily the best choice for a given problem.

Quasi-likelihoods

Wedderburn (1974) proposed the method of maximum quasi-likelihood for estimating regression parameters when one has knowledge of a mean-variance relationship for the response, but is unwilling or unable to commit to a valid fully specified distribution function.

Under the assumption that the Y_i are independent with mean μ_i and variance $Var(Y_i) = V_\mu(\mu_i)(\phi / w_i)$, the function,

$$U_i = u(\mu_i, y_i, \phi, w_i) = \frac{y_i - \mu_i}{(\phi / w_i) V_\mu(\mu_i)} \tag{32.14}$$

has the properties of an individual contribution to a score. Accordingly, the integral,

$$Q(\mu_i, y_i, \phi, w_i) = \int_y^{\mu_i} \frac{y_i - t}{(\phi / w_i) V_\mu(t)} dt \tag{32.15}$$

if it exists, should behave very much like a log-likelihood contribution. We may use to the individual contributions Q_i to define the *quasi-log-likelihood*, and the scaled and unscaled *quasi-deviance* functions

$$\begin{aligned} q(\mu, y, \phi, w) &= \sum_{i=1}^{N} Q(\mu_i, y_i, \phi, w_i) \\ D^*(\mu, y, \phi, w) &= -2\, q(\mu, y, \phi, w) \\ D(\mu, y, w) &= -2\phi D^*(\mu, y, \phi, w) \end{aligned} \tag{32.16}$$

We may obtain estimates of the coefficients by treating the quasi-likelihood $q(\mu, y, \phi, w)$ as though it were a conventional likelihood and maximizing it respect to β. As with conventional GLM likelihoods, the quasi-ML estimate of β does not depend on the value of the dispersion parameter ϕ. The dispersion parameter is conventionally estimated using the Pearson χ^2 statistic, but if the mean-variance assumption corresponds to a valid exponential family distribution, one may also employ the deviance statistic.

For some mean-variance specifications, the quasi-likelihood function corresponds to an ordinary likelihood in the linear exponential family, and the method of maximum quasi-likelihood is equivalent to ordinary maximum likelihood. For other specifications, there is no

corresponding likelihood function. In both cases, the distributional properties of the maximum quasi-likelihood estimator will be analogous to those obtained from maximizing a valid likelihood (McCullagh 1983).

We emphasize the fact that quasi-likelihoods offer flexibility in the mean-variance specification, allowing for variance assumptions that extend beyond those implied by exponential family distribution functions. One important example occurs when we modify the variance function for a Poisson, Binomial Proportion, or Negative Binomial distribution to allow a free dispersion parameter.

Furthermore, since the quasi-likelihood framework only requires specification of the mean and variance, it may be used to relax distributional restrictions on the form of the response data. For example, while we are unable to evaluate the Poisson likelihood for non-integer data, there are no such problems for the corresponding quasi-likelihood based on mean-variance equality.

A list of common quasi-likelihood mean-variance assumptions is provided below, along with names for the corresponding exponential family distribution:

$V_\mu(\mu)$	**Restrictions**	**Distribution**
1	None	Normal
μ	$\mu > 0,\ y \geq 0$	Poisson
μ^2	$\mu > 0,\ y > 0$	Gamma
μ^r	$\mu > 0,\ r \neq 0,\ 1,\ 2$	---
e^μ	None	---
$\mu(1-\mu)$	$0 < \mu < 1,\ 0 \leq y \leq 1$	Binomial Proportion
$\mu^2(1-\mu)^2$	$0 < \mu < 1,\ 0 \leq y \leq 1$	---
$\mu(1+k\mu)$	$\mu > 0,\ y \geq 0$	Negative Binomial

Note that the power-mean μ^r, exponential mean $\exp(\mu)$, and squared binomial proportion $\mu^2(1-\mu)^2$ variance assumptions do not correspond to exponential family distributions.

Estimation

Estimation of GLM models may be divided into the estimation of three basic components: the β coefficients, the coefficient covariance matrix Σ, and the dispersion parameter ϕ.

Coefficient Estimation

The estimation of β is accomplished using the method of maximum likelihood (ML). Let $y = (y_1, \ldots, y_N)'$ and $\mu = (\mu_1, \ldots, \mu_N)'$. We may write the log-likelihood function as

$$l(\mu, y, \phi, w) = \sum_{i=1}^{N} \log f(y_i, \theta_i, w_i) \tag{32.17}$$

Differentiating $l(\mu, y, \phi, w)$ with respect to β yields

$$\begin{aligned} \frac{\partial l}{\partial \beta} &= \sum_{i=1}^{N} \frac{\partial \log f(y_i, \theta_i, \phi, w_i)}{\partial \theta_i} \left(\frac{\partial \theta_i}{\partial \beta}\right) \\ &= \sum_{i=1}^{N} \left(\frac{y_i - b'(\theta_i)}{\phi / w_i}\right)\left(\frac{\partial \theta_i}{\partial \mu}\right)\left(\frac{\partial \mu_i}{\partial \eta}\right)\left(\frac{\partial \eta_i}{\partial \beta}\right) \\ &= \sum_{i=1}^{N} \frac{w_i}{\phi}\left(\frac{y_i - \mu_i}{V_\mu(\mu_i)}\right)\left(\frac{\partial \mu_i}{\partial \eta}\right) X_i \end{aligned} \tag{32.18}$$

where the last equality uses the fact that $\partial \theta_i / \partial \mu = V_\mu(\mu_i)^{-1}$. Since the scalar dispersion parameter ϕ is incidental to the first-order conditions, we may ignore it when estimating β. In practice this is accomplished by evaluating the likelihood function at $\phi = 1$.

It will prove useful in our discussion to define the *scaled deviance* D^* and the *unscaled deviance* D as

$$\begin{aligned} D^*(\mu, y, \phi, w) &= -2\{l(\mu, y, \phi, w) - l(y, y, \phi, w)\} \\ D(\mu, y, w) &= \phi D^*(\mu, y, \phi, w) \end{aligned} \tag{32.19}$$

respectively. The scaled deviance D^* compares the likelihood function for the saturated (unrestricted) log-likelihood, $l(y, y, \phi, w)$, with the log-likelihood function evaluated at an arbitrary μ, $l(\mu, y, \phi, w)$.

The unscaled deviance D is simply the scaled deviance multiplied by the dispersion, or equivalently, the scaled deviance evaluated at $\phi = 1$. It is easy to see that minimizing either deviance with respect to β is equivalent to maximizing the log-likelihood with respect to the β.

In general, solving for the first-order conditions for β requires an iterative approach. EViews offers three different algorithms for obtaining solutions: Newton-Raphson, BHHH, and IRLS - Fisher Scoring. All of these methods are variants of Newton's method but differ in the method for computing the gradient weighting matrix used in coefficient updates (see "Optimization Algorithms" on page 1319).

IRLS, which stands for *Iterated Reweighted Least Squares*, is a commonly used algorithm for estimating GLM models. IRLS is equivalent to Fisher Scoring, a Newton-method variant that employs the Fisher Information (negative of the *expected* Hessian matrix) as the update weighting matrix in place of the negative of the *observed* Hessian matrix used in standard Newton-Raphson, or the outer-product of the gradients (OPG) used in BHHH.

In the GLM context, the IRLS-Fisher Scoring coefficient updates have a particularly simple form that may be implemented using weighted least squares, where the weights are known functions of the fitted mean that are updated at each iteration. For this reason, IRLS is particularly attractive in cases where one does not have access to custom software for estimating GLMs. Moreover, in cases where one's preference is for an observed-Hessian Newton method, the least squares nature of the IRLS updates make the latter well-suited to refining starting values prior to employing one of the other methods.

Coefficient Covariance Estimation

You may choose from a variety of estimators for Σ, the covariance matrix of $\hat{\beta}$. In describing the various approaches, it will be useful to have expressions at hand for the *expected Hessian* (I), the *observed Hessian* (H), and the *outer-product of the gradients* (J) for GLM models. Let $X = (X_1, X_2, \ldots, X_N)'$. Then given estimates of $\hat{\beta}$ and the dispersion $\hat{\phi}$ (See "Dispersion Estimation," on page 447), we may write

$$\begin{aligned}
\hat{I} &= -E\left(\frac{\partial^2 l}{\partial\beta\partial\beta'}\right)\bigg|_{\hat{\beta}} = X'\hat{\Lambda}_I X \\
\hat{H} &= -\left(\frac{\partial^2 l}{\partial\beta\partial\beta'}\right)\bigg|_{\hat{\beta}} = X'\hat{\Lambda}_H X \\
\hat{J} &= \sum_{i=1}^{N}\left(\frac{\partial \log f_i}{\partial\beta}\frac{\partial \log f_i}{\partial\beta'}\right)\bigg|_{\hat{\beta}} = X'\hat{\Lambda}_J X
\end{aligned} \tag{32.20}$$

where $\hat{\Lambda}_I$, $\hat{\Lambda}_H$, and $\hat{\Lambda}_J$ are diagonal matrices with corresponding i-th diagonal elements

$$\begin{aligned}
\hat{\lambda}_{I,i} &= (w_i/\hat{\phi})\, V_\mu(\hat{\mu}_i)^{-1}\left(\frac{\partial\mu_i}{\partial\eta}\right)^2 \\
\hat{\lambda}_{H,i} &= \lambda_{I,i} + (w_i/\hat{\phi})(y_i - \hat{\mu}_i)\left\{V_\mu(\hat{\mu}_i)^{-2}\left(\frac{\partial\mu_i}{\partial\eta}\right)^2\left(\frac{\partial V_\mu(\hat{\mu}_i)}{\partial\mu}\right) - V_\mu(\hat{\mu}_i)^{-1}\left(\frac{\partial^2\mu_i}{\partial\eta^2}\right)\right\} \\
\hat{\lambda}_{J,i} &= \left\{(w_i/\hat{\phi})(y_i - \hat{\mu}_i)\, V_\mu(\hat{\mu}_i)^{-1}\left(\frac{\partial\mu_i}{\partial\eta}\right)\right\}^2
\end{aligned} \tag{32.21}$$

Given correct specification of the likelihood, asymptotically consistent estimators for the Σ may be obtained by taking the inverse of one of these estimators of the information matrix.

In practice, one typically matches the covariance matrix estimator with the method of estimation (*i.e.*, using the inverse of the expected information estimator $\hat{\Sigma}_I = \hat{\mathcal{I}}^{-1}$ when estimation is performed using IRLS) but mirroring is not required. By default, EViews will pair the estimation and covariance methods, but you are free to mix and match as you see fit.

If the variance function is incorrectly specified, the GLM inverse information covariance estimators are no longer consistent for Σ. The Huber-White Sandwich estimator (Huber 1967, White 1980) permits non GLM-variances and is robust to misspecification of the variance function. EViews offers two forms for the estimator; you may choose between one that employs the expected information ($\hat{\Sigma}_{IJ} = \hat{\mathcal{I}}^{-1}\hat{\mathcal{J}}\hat{\mathcal{I}}^{-1}$) or one that uses the observed Hessian ($\hat{\Sigma}_{HJ} = \hat{H}^{-1}\hat{\mathcal{J}}\hat{H}^{-1}$).

Lastly, you may choose to estimate the coefficient covariance with or without a degree-of-freedom correction. In practical terms, this computation is most easily handled by using a non d.f.-corrected version of $\hat{\phi}$ in the basic calculation, then multiplying the coefficient covariance matrix by $N/(N-k)$ when you want to apply the correction.

Dispersion Estimation

Recall that the dispersion parameter ϕ may be ignored when estimating β. Once we have obtained $\hat{\beta}$, we may turn attention to obtaining an estimate of ϕ. With respect to the estimation of ϕ, we may divide the distribution families into two classes: distributions with a free dispersion parameter, and distributions where the dispersion is fixed.

For distributions with a free dispersion parameter (Normal, Gamma, Inverse Gaussian), we must estimate ϕ. An estimate of the free dispersion parameter ϕ may be obtained using the generalized Pearson χ^2 statistic (Wedderburn 1972, McCullagh 1983),

$$\hat{\phi}_P = \frac{1}{N-k}\sum_{i=1}^{N}\frac{w_i(y_i-\hat{\mu}_i)^2}{V_\mu(\hat{\mu}_i)} \tag{32.22}$$

where k is the number of estimated coefficients. In linear exponential family settings, ϕ may also be estimated using the unscaled deviance statistic (McCullagh 1983),

$$\hat{\phi}_D = \frac{D(\mu, y, w)}{N-k} \tag{32.23}$$

For distributions where the dispersion is fixed (Poisson, Binomial, Negative Binomial), ϕ is naturally set to the theoretically proscribed value of 1.0.

In fixed dispersion settings, the theoretical restriction on the dispersion is sometimes violated in the data. This situation is generically termed *overdispersion* since ϕ typically exceeds 1.0 (though *underdispersion* is a possibility). At a minimum, unaccounted for overdispersion leads to invalid inference, with estimated standard errors of the $\hat{\beta}$ typically understating the variability of the coefficient estimates.

The easiest way to correct for overdispersion is by allowing a free dispersion parameter in the variance function, estimating ϕ using one of the methods described above, and using the estimate when computing the covariance matrix as described in "Coefficient Covariance Estimation," on page 446. The resulting covariance matrix yields what are sometimes termed GLM standard errors.

Bear in mind that estimating $\hat{\phi}$ given a fixed dispersion distribution violates the assumptions of the likelihood so that standard ML theory does not apply. This approach is, however, consistent with a quasi-likelihood estimation framework (Wedderburn 1974), under which the coefficient estimator and covariance calculations are theoretically justified (see "Quasi-likelihoods," beginning on page 443). We also caution that overdispersion may be evidence of more serious problems with your specification. You should take care to evaluate the appropriateness of your model.

Computational Details

The following provides additional details for the computation of results:

Residuals

There are several different types of residuals that are computed for a GLM specification:

- The *ordinary* or *response residuals* are defined as

$$\hat{\epsilon}_{oi} = (y_i - \hat{\mu}_i) \tag{32.24}$$

  The ordinary residuals are simply the deviations from the mean in the original scale of the responses.

- The *weighted* or *Pearson residuals* are given by

$$\hat{\epsilon}_{pi} = [(1/w_i)\, V_\mu(\hat{\mu}_i)]^{-1/2}(y_i - \hat{\mu}_i) \tag{32.25}$$

  The weighted residuals divide the ordinary response variables by the square root of the unscaled variance. For models with fixed dispersion, the resulting residuals should have unit variance. For models with free dispersion, the weighted residuals may be used to form an estimator of ϕ.

- The *standardized* or *scaled Pearson residuals)* are computed as

$$\hat{\epsilon}_{si} = [(\hat{\phi}/w_i)\, V_\mu(\hat{\mu}_i)]^{-1/2}(y_i - \hat{\mu}_i) \tag{32.26}$$

  The standardized residuals are constructed to have approximately unit variance.

- The *generalized* or *score residuals* are given by

$$\hat{\epsilon}_{gi} = [(\hat{\phi}/w_i)\, V_\mu(\hat{\mu}_i)]^{-1}(\partial\hat{\mu}_i/\partial\eta)(y_i - \hat{\mu}_i) \tag{32.27}$$

The scores of the GLM specification are obtained by multiplying the explanatory variables by the generalized residuals (Equation (32.18)). Not surprisingly, the generalized residuals may be used in the construction of LM hypothesis tests.

Sum of Squared Residuals

EViews reports two different sums-of-squared residuals: a basic sum of squared residuals, $SSR = \sum \hat{\epsilon}_{oi}^2$, and the Pearson SSR, $SSR_P = \sum \hat{\epsilon}_{pi}^2$.

Dividing the Pearson SSR by $(N - k)$ produces the Pearson χ^2 statistic which may be used as an estimator of ϕ, ("Dispersion Estimation" on page 447) and, in some cases, as a measure of goodness-of-fit.

Log-likelihood and Information Criteria

EViews always computes GLM log-likelihoods using the full specification of the density function: scale factors, inessential constants, and all. The likelihood functions are listed in "Distribution," beginning on page 439.

If your dispersion specification calls for a fixed value for ϕ, the fixed value will be used to compute the likelihood. If the distribution and dispersion specification call for ϕ to be estimated, $\hat{\phi}$ will be used in the evaluation of the likelihood. If the specified distribution calls for a fixed value for ϕ but you have asked EViews to estimate the dispersion, or if the specified value is not consistent with a valid likelihood, the log-likelihood will not be computed.

The AIC, SIC, and Hannan-Quinn information criteria are computed using the log-likelihood value and the usual definitions (Appendix E. "Information Criteria," on page 1335).

It is worth mentioning that computed GLM likelihood value for the normal family will differ slightly from the likelihood reported by the corresponding LS estimator. The GLM likelihood follows convention in using a degree-of-freedom corrected estimator for the dispersion while the LS likelihood uses the uncorrected ML estimator of the residual variance. Accordingly, you should take care not compare likelihood functions estimated using the two methods.

Deviance and Quasi-likelihood

EViews reports the *unscaled* deviance $D(\mu, y, w)$ or quasi-deviance. The quasi-deviance and quasi-likelihood will be reported if the evaluation of the likelihood function is invalid. You may divide the reported deviance by $(N - k)$ to obtain an estimator of the dispersion, or use the deviance to construct likelihood ratio or F-tests.

In addition, you may divide the deviance by the dispersion to obtain the scaled deviance. In some cases, the scaled deviance may be used as a measure of goodness-of-fit.

Restricted Deviance and LR Statistic

The restricted deviance and restricted quasi-likelihood reported on the main page are the values for the constant only model.

The entries for "LR statistic" and "Prob(LR statistic)" reported in the output are the corresponding χ^2_{k-1} likelihood ratio tests for the constant only null against the alternative given by the estimated equation. They are the analogues to the "F-statistics" results reported in EViews least squares estimation. As with the latter F-statistics, the test entries will not be reported if the estimated equation does not contain an intercept.

References

Agresti, Alan (1990). *Categorical Data Analysis*. New York: John Wiley & Sons.

Agresti, Alan (2007). *An Introduction to Categorical Data Analysis, 2nd Edition*. New York: John Wiley & Sons.

Hardin, James W. and Joseph M. Hilbe (2007). *Generalized Linear Models and Extensions, 2nd Edition*.

McCullagh, Peter (1983). "Quasi-Likelihood Functions," *Annals of Statistics*, 11, 59-67.

McCullagh, Peter, and J. A. Nelder (1989). *Generalized Linear Models, Second Edition*. London: Chapman & Hall.

Papke, Leslie E. and Jeffrey M. Wooldridge (1996). "Econometric Methods for Fractional Variables With an Application to 401 (K) Plan Participation Rates," *Journal of Applied Econometrics*, 11, 619-632.

Nelder, J. A. and R. W. M. Wedderburn (1972). "Generalized Linear Models," *Journal of the Royal Statistical Society, A*, 135, 370-384.

Wedderburn, R. W. M. (1974). "Quasi-Likelihood Functions, Generalized Linear Models and the Gauss-Newton Method," *Biometrika*, 61, 439-447.

Wooldridge, Jeffrey M. (1997). "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 33. Robust Least Squares

Ordinary least squares estimators are sensitive to the presence of observations that lie outside the norm for the regression model of interest. The sensitivity of conventional regression methods to these *outlier* observations can result in coefficient estimates that do not accurately reflect the underlying statistical relationship.

Robust least squares refers to a variety of regression methods designed to be *robust*, or less sensitive, to outliers. EViews offers three different methods for robust least squares: M-estimation (Huber, 1973), S-estimation (Rousseeuw and Yohai, 1984), and MM-estimation (Yohai 1987). The three methods differ in their emphases:

- M-estimation addresses dependent variable outliers where the value of the dependent variable differs markedly from the regression model norm (large residuals).
- S-estimation is a computationally intensive procedure that focuses on outliers in the regressor variables (high leverages).
- MM-estimation is a combination of S-estimation and M-estimation. The procedure starts by performing S-estimation, and then uses the estimates obtained from S-estimation as the starting point for M-estimation. Since MM-estimation is a combination of the other two methods, it addresses outliers in both the dependent and independent variables.

Least squares diagnostics for outlier detection are described in greater detail in "Leverage Plots" on page 250 and "Influence Statistics" on page 251.

Background

Before describing the mechanics of estimating robust regression models in EViews, it will be useful to review the basics of the three estimation methods and to outline alternative approaches for computing the covariance matrix of the coefficient estimates.

M-estimation

The traditional least squares estimator is computed by finding coefficient values that minimize the sum of the squared residuals:

$$\hat{\beta}_{LS} = \text{argmin}_{\beta} \sum_{i=1}^{N} r_i(\beta)^2 \tag{33.1}$$

where the residual function r is given by

$$r_i(\beta) = r_i = y_i - X_i'\beta \tag{33.2}$$

Since the residuals r_i enter the objective function on the right-hand side of Equation (33.1) after squaring, the effects of outliers are magnified accordingly.

M-estimator definition

One obvious approach to robust regression replaces squaring of residuals in Equation (33.1) with a function that provides less weight to outliers. The Huber M-estimator ("M" for "maximum likelihood estimator-like") computes the coefficient values that minimize the summed values of a function ρ of the residuals:

$$\hat{\beta}_M = \operatorname{argmin}_\beta \sum_{i=1}^{N} \rho_c\left(\frac{r_i(\beta)}{\sigma w_i}\right) \tag{33.3}$$

where σ is a measure of the scale of the residuals, c is an arbitrary positive *tuning constant* associated with the function, and where w_i are individual weights that are generally set to 1, but may be set to:

$$w_i = \sqrt{1 - X_i(X'X)^{-1}X_i'} \tag{33.4}$$

to down-weight observations with high leverage (large diagonals of the Hat Matrix).

The potential choices for the function ρ (Andrews, Bisquare, Cauchy, Fair, Huber-Bisquare, Logistic, Median, Talworth, Welsch) are outlined below along with the default values of the tuning constants:

Name	$\rho_c(X)$	Default c
Andrews	$\begin{cases} c^2\left(1-\cos\left(\frac{X}{c}\right)\right) & \text{if } \lvert X\rvert \le \pi c \\ 2c^2 & \text{otherwise} \end{cases}$	1.339
Bisquare	$\begin{cases} \frac{c^2}{6}\left(1-\left(1-\left(\frac{X}{c}\right)^2\right)^3\right) & \text{if } \lvert X\rvert \le c \\ \frac{c^2}{6} & \text{otherwise} \end{cases}$	4.685
Cauchy	$\left(\frac{c^2}{2}\right)\log\left(1+\left(\frac{X}{c}\right)^2\right)$	2.385
Fairl	$c^2\log\left(\frac{\lvert X\rvert}{c} - \log\left(1+\left(\frac{X}{c}\right)\right)\right)$	1.4

Huber	$\begin{cases} \frac{X^2}{2} & \text{if } \lvert X\rvert \le c \\ c\lvert X\rvert - \frac{c^2}{2} & \text{otherwise} \end{cases}$	1.345
Logistic	$c^2 \log\left(\cosh\left(\frac{X}{c}\right)\right)$	1.205
Median	$\begin{cases} \frac{X^2}{2c} & \text{if } \lvert X\rvert \le c \\ \frac{Xc^2}{2} & \text{otherwise} \end{cases}$	0.01
Talworth	$\begin{cases} \frac{X^2}{2} & \text{if } \lvert X\rvert \le c \\ \frac{c^2}{2} & \text{otherwise} \end{cases}$	2.796
Welsch	$\frac{c^2}{2}\left(1 - \exp\left(-\left(\frac{X}{c}\right)^2\right)\right)$	2.985

The default tuning constants for each function are taken from Holland and Welsch (1977), and are chosen so that the estimator achieves 95% asymptotic efficiency under residual normality.

M-estimator calculation

If the scale σ is known, then the k-vector of coefficient estimates $\hat{\beta}_M$ may be found using standard iterative techniques for solving the k nonlinear first-order equations:

$$\sum_{i=1}^{N} \psi_c\left(\frac{r_i(\beta)}{\sigma w_i}\right)\frac{x_{ij}}{w_i} = 0 \qquad j = 1, \ldots, k \tag{33.5}$$

for β, where $\psi_c(\cdot) = \rho_c{}'(\cdot)$, the derivative of the $\rho_c(\cdot)$ function, and x_{ij} is the value of the j-th regressor for observation i.

Since σ is not known, a sequential procedure is used that alternates between: (1) computing updated estimates of the scale $\hat{\sigma}^{(j+1)}$ given coefficient estimates $\hat{\beta}^{(j)}$, and (2) using iterative methods to find the $\hat{\beta}^{(j)}$ that solves Equation (33.5) for a given $\hat{\sigma}^{(j)}$. The initial

$\hat{\beta}_M^{(0)}$ are obtained from ordinary least squares. The initial coefficients are used to compute a scale estimate, $\hat{\sigma}^{(1)}$, and from that are formed new coefficient estimates $\hat{\beta}_M^{(1)}$, followed by a new scale estimate $\hat{\sigma}^{(2)}$, and so on until convergence is reached.

Given an estimate $\hat{\beta}_M^{(s-1)}$, the updated scale $\hat{\sigma}^{(s)}$ is estimated using one of three different methods: Mean Absolute Deviation – Zero Centered (MADZERO), Median Absolute Deviation – Median Centered (MADMED), or Huber Scaling:

MADZERO	$\hat{\sigma}^{(s)} = \text{median}\left[\frac{\text{abs}(r_i^{(s-1)})}{0.6745}\right]$
MADMED	$\hat{\sigma}^{(s)} = \text{median}\left[\frac{\text{abs}(r_i^{(s-1)} - \text{median}[r_i^{(s-1)}])}{0.6745}\right]$
Huber	$\hat{\sigma}^{(s)} = \sqrt{\frac{1}{hN}(\hat{\sigma}^{(j-1)})^2 \sum_{i=1}^{N} \xi\left(\frac{r_i^{(s-1)}}{\hat{\sigma}^{(s-1)}}\right)}$ where $\xi(v) = \min(\frac{v^2}{2}, \frac{2.5^2}{2})$ $h = 0.48878N$

where $r_i^{(s-1)}$ are the residuals associated with $\hat{\beta}_M^{(s-1)}$ and where the initial scale required for the Huber method is estimated by:

$$\hat{\sigma}^{(0)} = \sqrt{\sum_{i=0}^{N} (r_i^{(0)})^2} \tag{33.6}$$

M-estimator summary statistics

EViews automatically computes a variety of robust summary statistics for equations estimated using M-estimators.

R-squared

Maronna (1996, p. 171) defines the robust R^2 statistic as

$$R^2 = \frac{\sum_{i=1}^{N} \rho_c\left(\frac{y_i - \tilde{\mu}}{\hat{\sigma} w_i}\right) - \sum_{i=1}^{N} \rho_c\left(\frac{\hat{r}_i}{\hat{\sigma} w_i}\right)}{\sum_{i=1}^{N} \rho_c\left(\frac{y_i - \tilde{\mu}}{\hat{\sigma} w_i}\right)}$$

where $\tilde{\mu}$ is the M-estimate from the constant-only specification.

The adjusted R^2 is calculated as:

$$\overline{R}^2 = 1 - (1 - R^2)\frac{N-1}{N-k} \tag{33.7}$$

Both of these statistics can be highly sensitive to the choice of function, even when the coefficient estimates and standard errors are not. Studies have also found that these statistics may be upwardly biased (see, for example, Renaud and Victoria-Feser (2010)).

Rw-squared

Renaud and Victoria-Feser (2010) propose the R_W^2 statistic, and provide simulation results showing R_W^2 to be a better measure of fit than the robust R^2 outlined above. The R_W^2 statistic is defined as

$$R_W^2 = \frac{\sum_{i=1}^{N} \rho_{ci} \cdot (y_i - \bar{y}_W)(y_i - \overline{\hat{y}_W})}{\sqrt{(\sum_{i=1}^{N} \rho_{ci} \cdot (y_i - \bar{y}_W))(\sum_{i=1}^{N} \rho_{ci} \cdot (y_i - \overline{\hat{y}_W}))}} \tag{33.8}$$

where $\rho_{ci} = \rho_c(r_i(\hat{\beta})/(\hat{\sigma} w_i))$ is the function of the residual value and

$$\bar{y}_W = \sum_{i=1}^{N} \rho_{ci} \cdot y_i, \quad \overline{\hat{y}_W} = \sum_{i=1}^{N} \rho_{ci} \cdot \hat{y}_i \tag{33.9}$$

As with the robust R^2, an adjusted value of $\overline{R}_W^2$ may be calculated from the unadjusted statistic

$$\overline{R}_W^2 = 1 - (1 - R_W^2)\frac{N-1}{N-k} \tag{33.10}$$

Rn-squared Statistic

The R_N^2 statistic is a robust version of a Wald test of the hypothesis that all of the coefficients are equal to zero. It is calculated using the standard Wald test quadratic form:

$$R_N = \beta_1{}'\hat{\Omega}_1^{-1}\beta_1 \tag{33.11}$$

where β_1 are the k_1 non-intercept robust coefficient estimates and $\hat{\Omega}$ is the corresponding estimated covariance. Under the null hypothesis that all of the coefficients are equal to zero, the R_N^2 statistic is asymptotically distributed as a $\chi^2(k_1)$.

Deviance

The deviance is the value of the objective function Equation (33.3) evaluated at the final coefficient estimates and estimate of the scale:

$$\text{Deviance} = 2\hat{\sigma}^2 \sum_{i=1}^{N} \rho_c\left(\frac{r_i(\hat{\beta})}{\hat{\sigma} w_i}\right) \tag{33.12}$$

Information Criteria

EViews reports two information criteria for M-estimated equations: the robust equivalent of the Akaike Information Criterion (AIC_R), and a corresponding robust Schwarz Information Criterion (BIC_R):

$$AIC_R = 2\sum_{i=1}^{N} \rho_c\left(\frac{r_i(\hat{\beta})}{\hat{\sigma} w_i}\right) + 2k\left\{\frac{\sum_{i=1}^{N} \psi_c\left(\frac{r_i(\hat{\beta})}{\sigma w_i}\right)^2}{\sum_{i=1}^{N} \psi_c'\left(\frac{r_i(\hat{\beta})}{\sigma w_i}\right)}\right\} \tag{33.13}$$

$$BIC_R = 2\sum_{i=1}^{N} \rho_c\left(\frac{r_i(\hat{\beta})}{\hat{\sigma} w_i}\right) + 2k \cdot \log(T)$$

where ψ_c' is the derivative of ψ_c as outlined in Holland and Welsch (1977). See Ronchetti (1985) for details.

S-estimation

The S-estimator ("S" for "scale statistic") is a member of the class of high-breakdown-value estimators introduced by Rousseeuw and Yohai (1984). The breakdown-value of an estimator can be seen as a measure of an estimator's robustness to outliers. (A good description of breakdown-values and high-breakdown-value estimators can be found in Hubert and Debruyne (2009)).

S-estimator definition

S-estimators find the set of coefficients β that provide the smallest estimate of the scale S such that:

$$\frac{1}{N-k}\sum_{i=1}^{N} h_c\left(\frac{r_i(\beta)}{S}\right) = b \tag{33.14}$$

for the function $h_c(\cdot)$ with tuning constant $c > 0$, where b is taken to be $E_\phi(h_c)$ with ϕ the standard normal. The breakdown value B for this estimator is $B = b/max(h_c)$.

Following Rousseeuw and Yohai, we choose a function based on the integral of the Biweight function

$$h_c(X) = \begin{cases} \left(\frac{X}{c}\right)^6 - 3\left(\frac{X}{c}\right)^4 + 3\left(\frac{X}{c}\right)^2 & \text{if } |X| \le c \\ 1 & \text{otherwise} \end{cases} \tag{33.15}$$

and estimate the scale S using the Median Absolute Deviation, Zero Centered (MADZERO) method.

Note that c affects the objective function through h_c and b. c is typically chosen to achieve a desired breakdown value. EViews defaults to a c value of 1.5476 implying a breakdown value of 0.5. Other notable values for c (with associated B) are:

c	B
5.1824	0.10
4.0963	0.15
3.4207	0.20
2.9370	0.25
2.5608	0.30
1.9880	0.40
1.5476	0.50

S-estimator calculation

Calculation of S-estimates is computationally intensive, and there exist a number of fast algorithms that provide accurate approximations. EViews uses the Fast-S algorithm of Salibian-Barrera and Yohai (2006):

1. Obtain a random subsample of size m from the data and compute the least squares regression to obtain a $\hat{\beta}^{(0)}$. By default m is set equal to k, the number of regressors. (Note that with the default $m = k$, the regression will produce an exact fit for the subsample.)

2. Using the full sample, perform a set of $r \geq 0$ refinements to the initial coefficient estimates using a variant of M-estimation which takes a single step toward the solution of Equation (33.5) at every $\hat{\beta}^{(s)}$ update. These modified M-estimate refinements employ the Bisquare function h_c with tuning parameter and scale estimator

$$\hat{\sigma}^{(s)} = \sqrt{\frac{1}{N-k}\frac{(\hat{\sigma}^{(s-1)})^2}{B}\sum_{i=1}^{N} h_c\left(\frac{r_i^{(s-1)}}{\sigma^{(s-1)}}\right)} \tag{33.16}$$

   where $\hat{\sigma}^{(s-1)}$ is the previous iteration's estimate of the scale and B is the breakdown value defined earlier.

   The initial scale estimator $\hat{\sigma}^{(0)}$ is obtained using MADZERO

3. Compute a new set of residuals over the entire sample using the possibly refined initial coefficient estimates, compute an estimate of the scale $\hat{S}^{(0)}$ using MADZERO, and produce a final estimate of S by iterating Equation (33.16) (with S in place of σ) to convergence or until $\hat{S}^{(j)} < B$.

4. Steps 1-3 are repeated Q times. The best (smallest) q scale estimates are refined using M-estimation as in Step 2 with $r = 50$ (or until convergence). The smallest scale from those refined scales is the final estimate of S, and the final coefficient estimates are the corresponding estimates of β.

S-estimator summary statistics

The following summary statistics are available for equations estimated by S-estimation:

R-squared

The robust version of R^2 for S-estimation is given by:

$$R^2 = 1 - \frac{(N - kS_F^2)}{NS_C^2} \tag{33.17}$$

where S_F is the estimate of the scale from the final estimation, and S_C is an estimate of the scale from S-estimation with only a constant as a regressor.

Deviance

The S-estimator deviance value is given by:

$$Deviance = 2S^2 \tag{33.18}$$

Rn-squared Statistic

The R_N^2 statistic is identical to the one computed for M-estimation. See "Rn-squared Statistic" on page 455 for discussion.

MM Estimation

MM-estimation addresses outliers in both the dependent and the independent variables by combining S-estimation with M-estimation.

The MM-estimator first computes S-estimates of the coefficients and scale, then uses the estimate of the scale as a fixed value in iterating to find a solution to Equation (33.5). The second stage M-estimation in EViews uses the Bisquare function with a default tuning parameter value of 4.684 which gives 95% relative efficiency for normal errors (Yohai, 1987).

The summary statistics for MM-estimation are obtained from the second-stage M-estimation procedure.

Coefficient Covariance Methods

EViews offers three different methods for computing the coefficient covariance matrix taken from Huber (1981, p. 173, equations 6.5, 6.6 and 6.7). All three methods provide unbiased estimates of the covariance matrix, with none having better properties than the others.

Type I (default)	$\Lambda^2 \dfrac{[1/(N-K)]\sum_{i=1}^{N}\psi_c(r_i)^2}{[(1/N)\sum_{i=1}^{N}\psi_c'(r_i)]^2}(X'X)^{-1}$
Type II	$\Lambda \dfrac{[1/(N-K)]\sum_{i=1}^{N}\psi_c(r_i)^2}{[(1/N)\sum_{i=1}^{N}\psi_c'(r_i)]^2} W^{-1}$
Type III	$\Lambda^{-1}\dfrac{1}{N-k}[\sum_{i=1}^{N}\psi_c(r_i)^2]\, W^{-1}(X'X)\, W^{-1}$

with

$$\Lambda = 1 + \frac{N}{k}\frac{\sum_{i=1}^{N}[\psi_c'(r_i) - \overline{\psi_c'}]}{(\overline{\psi_c'})^2}$$

$$\overline{\psi'} = \frac{1}{N}\sum_{i=1}^{N}\psi_c'(r_i) \tag{33.19}$$

$$W_{js} = \sum_{i=1}^{N}\psi_c'(r_i)x_{ij}x_{is} \qquad j, s = 1, \ldots, k$$

where as before, $\psi_c(\cdot) = \rho_c'(\cdot)$ and x_{ij} is the value of the j-th regressor for observation i.

The first method (which is the easiest computationally) is the default choice.

Estimating Robust Regression in EViews

To estimate an equation using robust regression, open the equation dialog by selecting **Quick/Estimate Equation…**, or by selecting **Object/New Object…/Equation** and selecting **ROBUSTLS** from the **Method** dropdown menu. EViews will display the robust regression dialog:

The **Specification** tab lets you enter the basic regression specification and the type of robust regression to be performed:

- Enter the regression specification in list form (dependent variable followed by the list of regressors) in the **Equation specification** variable edit field.
- Specify the estimation type by choosing one of the three estimation types **M-estimation**, **S-estimation**, or **MM-estimation** in the **Robust estimation type** dropdown. By default, EViews will perform **M-estimation**.
- Enter the estimation **Sample** in the edit field

Click on **OK** to estimate the equation using the default settings, or click on **Options** to inspect settings for advanced options.

Options

Clicking on the **Options** tab of the dialog lets you specify advanced estimation options. The tab will display different settings depending on whether you choose M-estimation, S-estimation, or MM-estimation in the **Robust estimation type** dropdown.

M-estimation options

For M-estimation, you will be offered choices the for objective specification, scale estimator, and covariance type.

Objective specification

The **Objective specification** section of the dialog controls the choice of function and the tuning constant:

- You should use the **Function** dropdown to choose from among the 10 available functions: Andrews, Bisquare, Cauchy, Fair, Huber, Huber-Bisquare, Logistic, Median, Talworth, and Welsch (Bisquare is the default).
- The **Scale using H-matrix** checkbox may be used to define individual weights w_i as described in Equation (33.4) on page 452.
- The **Default constant** and **User-specified** constant radio buttons should be used to specify the value of the tuning constant. Choosing **Default constant** will use the Holland and Welsch (1977) values of the tuning constant as described on page 452. To provide your own tuning value, select **User-specified constant** and enter a positive number or name of a scalar object in the **Tuning value** edit field.

Scale estimates

The **Scale estimates** dropdown is used to select between Mean Absolute Deviation (MAD) with either zero or median centering, Huber scaling, or a user-specified scale. The default estimator is MAD with median centering. To provide a user-specified scale, select **Fixed user** in the dropdown and enter a positive number or name of a scalar object in the **User scale** edit field

Other settings

The **Covariance** type dropdown allows you to choose between the three types of Huber covariance methods.

The **Iteration** control section controls the maximum iterations and convergence tolerances used when solving the nonlinear equations in Equation (33.5). Click on **Display Settings** to show information in the EViews output.

You may use the **Coefficient name** edit field to specify a coefficient vector other than the default C to hold the results from estimation.

S-estimation options

The S-estimator offers a set of estimation options than differs markedly from those offered by the M-estimator. In contrast to the M-estimator, there is no option for choosing the scale estimator. You will, however, be offered a slightly modified set of **Objective specification** options and a new set of **S options**.

Objective specification

The **Objective specification** section of the dialog allows you to specify the values of the tuning and breakdown constants:

- You should select **Default constant** to use the default c value of 1.5476 (0.5 breakdown) or you may select **User-specified constant** and enter a value or name of a scalar in the **Tuning value** edit field. The tuning value must be positive.

- Note that the function choice dropdown is disabled since the S-estimation function is restricted to be the Tukey Bisquare.

S settings

The **S options** portion of the dialog allows you to control the settings for the Fast-S algorithm:

- **Number of trials** controls the number Q of S subsample estimates to be computed. By default, EViews will compute 200 estimates.
- **Initial sample size** specifies the size m of each random subsample used in the S initializing regression. By default, this field will be initialized at the number of regressors.
- **Max refinements** controls the number of refinements r to each initial subsample regression estimate. Each refinement consists of a single modified M-estimator step toward the solution of the nonlinear equations.
- **Number of comparisons** is the number of best estimates q that are candidates for refinement and comparison to find the final estimate.
- The **Random generator** and **Seed** fields control the construction of the random subsamples required for the Fast-S algorithm. You may the leave the **Seed** field blank, in which case EViews will use the clock to obtain a seed at the time of estimation, or you may provide an integer from 0 to 2,147,483,647. The **Clear** button may be used to clear the seed used by a previously estimated equation.

For additional discussion of these settings, see "S-estimator calculation" on page 457.

Other settings

The **Coefficient name**, **Covariance type**, and **Iteration control** settings are as described in "M-estimation options" on page 460.

MM-estimation options

The options for the MM-estimator are closely related to the options for the S-estimator described in "S-estimation options" on page 462.

The main difference between the MM and S options is in the settings for the tuning parameters. Since the MM estimator combines both S and M estimation, the dialog has separate fields for the tuning values used in the S-estimation and the tuning value used in the M-estimation.

The **Default constants** setting sets an S tuning parameter of 1.5476 (0.5 breakdown) and a default M tuning value of 4.684 (for 0.95 relative efficiency under normal errors).

An Illustration

For our example of robust regression estimation, we employ the "salinity" data set taken from Rousseeuw and Leroy (1987, page 82), which has been used in many studies of robust regression and outlier effects. See, for example, Rousseeuw and van Zomeren (1992) and Fung (1993). The data consist of 28 observations on water salinity (salt concentration) and river discharge measurements taken from Pamlico Sound in North Carolina.

We are interested in modeling the relationship between the amount of discharge and the level of salinity. The regression model of interest is:

$$S_t = \beta_1 + \beta_2 S_{t-1} + \beta_3 t + \beta_4 D_t + \epsilon_t \tag{33.20}$$

where S is the salinity level, D is discharge and t represents a time trend.

The data are provided in the workfile "Rousseeuw and Leroy.wf1" located in the EViews application data directory. The series SALINITY and DISCHARGE contain the salt and discharge measurements, TREND contains the number of biweekly periods since the start of spring, and LAGSEL is contains the lagged value of SALINITY.

We begin with ordinary least squares estimation of this specification. The equation EQ01 in the workfile contains these least squares estimates:

Dependent Variable: SALINITY
Method: Least Squares
Date: 07/23/12 Time: 16:19
Sample: 1 28
Included observations: 28

Variable	Coefficient	Std. Error	t-Statistic	Prob.
C	9.590263	3.125086	3.068799	0.0053
LAGSAL	0.777105	0.086222	9.012849	0.0000
TREND	-0.025512	0.161079	-0.158384	0.8755
DISCHARGE	-0.295036	0.106804	-2.762410	0.0108

R-squared	0.826388	Mean dependent var	10.55357
Adjusted R-squared	0.804687	S.D. dependent var	3.010166
S.E. of regression	1.330320	Akaike info criterion	3.540279
Sum squared resid	42.47401	Schwarz criterion	3.730594
Log likelihood	-45.56391	Hannan-Quinn criter.	3.598460
F-statistic	38.07988	Durbin-Watson stat	2.660574
Prob(F-statistic)	0.000000		

Rousseeuw and Leroy identify observation 16 as being an outlier. We can confirm this finding by looking at the influence statistics and leverages for this equation. From the EQ01 menu, display the influence statistics dialog by selecting **View/Stability Diagnostics/Influence Statistics...**

Check the box labeled **Hat Matrix** to tell EViews that you want to view the diagonals of the matrix along with the default results, then click on **OK** to display the graphs:

The spikes in the graphs for all four influence measures point to observation 16 as being an outlier. This finding is confirmed by the leverage plot view of EQ01. Select **View/Stability Diagnostics/Leverage Plots...** and click on **OK** to accept the default settings:

The graphs support the view that observation 16 has high leverage, especially in the relationship between SALINITY and DISCHARGE. Using the mouse pointer to hover over the outlier confirms the identity of the outlier observation. (For additional discussion of these diagnostics, see "Leverage Plots" on page 250 and "Influence Statistics" on page 251.)

M-estimation example

Given the presence of outliers, we re-estimate the regression using robust M-estimation. Create a new equation object by clicking on **Quick/Estimate Equation…**, or by selecting **Object/New Object…/Equation** and then select **ROBUSTLS** from the **Method** dropdown menu. Enter the dependent variable followed by the list of regressor variables in the **Equation specification** edit field:

```
salinity c lagsel trend discharge
```

and click on **OK** to instruct EViews to estimate the specification using the default estimator and settings. (For convenience, we have included the equation object EQ02 estimated using these settings in your workfile.)

Dependent Variable: SALINITY
Method: Robust Least Squares
Date: 08/15/12 Time: 15:32
Sample: 1 28
Included observations: 28
Method: M-estimation
M settings: weight=Bisquare, tuning=4.685, scale=MAD
(median centered)
Huber Type I Standard Errors & Covariance

Variable	Coefficient	Std. Error	z-Statistic	Prob.
C	18.29590	2.660551	6.876732	0.0000
LAGSAL	0.722380	0.073405	9.840981	0.0000
TREND	-0.202283	0.137135	-1.475064	0.1402
DISCHARGE	-0.623040	0.090928	-6.852032	0.0000

Robust Statistics

R-squared	0.620467	Adjusted R-squared	0.573026
Rw-squared	0.933495	Adjust Rw-squared	0.933495
Akaike info criterion	52.08135	Schwarz criterion	57.40400
Deviance	23.76607	Scale	0.734314
Rn-squared statistic	202.9058	Prob(Rn-squared stat.)	0.000000

Non-robust Statistics

Mean dependent var	10.55357	S.D. dependent var	3.010166
S.E. of regression	1.577654	Sum squared resid	59.73579

A description of the settings used in the M-estimation is presented at the top of the output. Here we see that the Bisquare function with a default tuning parameter value of 4.685 was used, that the scale was estimated using the median centered, median absolute deviation method, and that the z-statistics in the output are based on Huber Type I covariance estimates.

Turning to the coefficient estimates, we see the effect on the coefficient estimates of moving from least squares to robust M-estimation. The M-estimator produces a much larger negative impact of DISCHARGE on SALINITY than does ordinary least squares (-0.623 versus -0.295) with the M-estimator coefficient estimated with similar precision (0.091 versus 0.107). The sensitivity of the DISCHARGE coefficient estimates to robust estimation is in accord with the earlier EQ01 diagnostic suggesting that observation 16 had high leverage for the relationship between SALINITY and DISCHARGE.

The bottom portion of the output displays the R^2 and R^2_W goodness-of-fit and adjusted measures, along which indicate that the model accounts for roughly 60-90% of the variation in the constant-only model. The R^2_N statistic of 202.906 and corresponding p-value of 0.00 indicate strong rejection of the null hypothesis that all non-intercept coefficients are equal to

zero. Lastly, the output shows the value of the deviance, information criteria, and the estimated scale. These measures may be of use when comparing models. See "M-estimator summary statistics" on page 454 for formulae and discussion.

MM-estimation example

Next, we estimate the equation using MM-estimation. In the **Specification** tab:

- Specify your equation estimation method as **ROBUSTLS - Robust Least Squares**,
- Change the **Robust Estimation Type dropdown** to **MM-estimation**
- Fill out the **Specification** edit field with "salinity c lagsel trend discharge" as before.

Next, we will provide values for the tuning and breakdown values, and will specify options to control the S-estimation refinement.

Click on the **Options** tab to display the additional estimation settings:

- For the objective specification, we will provide tuning and breakdown values. Select the **User-specified constants** radio button and enter the values as depicted. The S-tuning value of 2.937 is chosen to provide a breakdown of 0.25; the M-tuning value of 3.44 is chosen to produce relative efficiency of 0.85.
- Select **Huber Type II** standard errors.
- Under **S options**, enter values for the **Number of trials,** and **Max refinements**, as depicted. The **Initial sample size** will be pre-filled with the number of regressor variables specified on the first tab of the dialog—we will leave this at the default setting.

Enter "5" in the **Number of comparisons** edit field so that EViews will refine the best 5 of the 200 trials.

Additional detail on these settings are provided in "S-estimation" on page 456 and "S-estimation options" on page 462.

Click on **OK** to accept the specification and options and to estimate the equation. EViews will display the results of the MM-estimation.

Dependent Variable: SALINITY
Method: Robust Least Squares
Date: 08/15/12 Time: 16:41
Sample: 1 28
Included observations: 28
Method: MM-estimation
S settings: tuning=2.937, breakdown=0.25, trials=200, subsmpl=4,
refine=2, compare=5
M settings: weight=Bisquare, tuning=3.44
Random number generator: rng=kn, seed=801674785
Huber Type II Standard Errors & Covariance

Variable	Coefficient	Std. Error	z-Statistic	Prob.
C	18.35831	3.589233	5.114829	0.0000
LAGSAL	0.715431	0.066356	10.78163	0.0000
TREND	-0.187484	0.139891	-1.340216	0.1802
DISCHARGE	-0.625891	0.133533	-4.687167	0.0000

Robust Statistics

R-squared	0.661056	Adjusted R-squared	0.618688
Rw-squared	0.922073	Adjust Rw-squared	0.922073
Akaike info criterion	23.15291	Schwarz criterion	32.83208
Deviance	26.94331	Scale	1.175363
Rn-squared statistic	202.9223	Prob(Rn-squared stat.)	0.000000

Non-robust Statistics

Mean dependent var	10.55357	S.D. dependent var	3.010166
S.E. of regression	1.586231	Sum squared resid	60.38708

Notice that the top of the output displays various settings for both the S and the M-portions of the MM-estimation. In addition to showing the S-tuning value of 2.937 and associated breakdown value of 0.25, EViews reports that the S-estimation consists of 200 trials with initial coefficients obtained from a random initial sample size of 4 and 2 initial refinement steps. The final comparison involves fully refining 5 sets of the scale estimates and choosing the smallest scale estimate.

Once the scale estimate is obtained, EViews performs fixed scale M-estimation using the reported 3.44 tuning parameter.

EViews also reports information on the random number generator used to obtain the random subsamples, and the method used to obtain coefficient estimate covariances.

Turning to the results, we note that despite the difference in robust estimation method, relative efficiency settings, and method of computing standard errors, the results from the M-estimation and the MM-estimation are generally quite similar. Most importantly, both estimates show statistically significant DISCHARGE coefficients of around -0.62 with roughly comparable coefficient standard errors (0.089 versus 0.13). The results for other coefficients are even closer.

The MM-estimate of the scale is considerably larger than that obtained from M-estimation (1.16 versus 0.73), but the overall goodness-of-fit measures and R_N^2 statistic and test results are quite similar.

References

Croux, C., G. Dhaene, and D. Hoorelbeke (2003). "Robust standard errors for robust estimators," Discussion Papers Series 03.16, K.U. Leuven, CES.

Fung, Wing-Kam (1993). "Unmasking Outliers and Leverage Points: A Confirmation," *Journal of the American Statistical Association*, 88(422), 515-519.

Holland, Paul W. and Roy E. Welsch (1977). "Robust regression using iteratively reweighted least squares," *Communications in Statistics - Theory and Methods*, 6(9), 813–827.

Huber, Peter J. (1973). "Robust Regression: Asymptotics, Conjectures and Monte Carlo," *The Annals of Statistics*, 1(5), 799–821.

Huber, Peter J. (1981). *Robust Statistics*. New York: John Wiley & Sons.

Hubert, Mia and Michiel Debruyne (2009). "Breakdown Value," *Wiley Interdisciplinary Reviews: Computational Statistics,* 1(3), 296–302.

Maronna, Ricardo A., R. Douglas Martin, and Victor J. Yohai (2006). *Robust Statistics*. Chichester, England: John Wiley & Sons, Ltd.

Renaud, Olivier and Maria-Pia Victoria-Feser (2010). "A Robust Coefficient of Determination for Regression," *Journal of Statistical Planning and Inference*, 140, 1852–1862

Ronchetti, Elvezio (1985). "Robust Model Selection in Regression," *Statistics & Probability Letters*, 3, 21–23.

Rousseeuw, P.J. and A. M. Leroy (1987). *Robust Regression and Outlier Detection*. New York: John Wiley & Sons, Inc.

Rousseeuw, P. J. and Bert C. van Zomeren (1992). "A Comparison of Some Quick Algorithms for Robust Regression," *Computational Statistics & Data Analysis*, 14(1), 107–116

Rousseeuw, P. J. and V. J. Yohai (1984), "Robust Regression by Means of S-Estimators," in *Robust and Nonlinear Time Series*, J. Franke, W. Härdle, and D. Martin, eds., *Lecture Notes in Statistics No. 26*, Berlin: Springer-Verlag.

Saliban-Barrera, Matías, and Víctor J. Yohai (2006). "A Fast Algorithm for S-Regression Estimates," *Journal of Computational and Graphical Statistics*, 15(2), Pages 414–427.

Yohai, Víctor J. (1987). "High Breakdown-Point and High Efficiency Robust Estimates for Regression," *The Annals of Statistics*, 15(2), 642-656.

Chapter 34. Least Squares with Breakpoints

The standard linear regression model described in Chapter 20. "Basic Regression Analysis," on page 5 assumes that the parameters of the model do not vary across observations.

Despite this assumption. *structural change*, the changing of parameters at dates in the sample period, plays an empirically relevant role in applied time series analysis. Accordingly, there has been a large volume of work targeted at developing testing and estimation methodologies for regression models which allow for change. Hansen (2001) and Perron (2006) offer useful overviews of the literature.

This chapter describes the EViews tools for estimating linear regression models that are subject to structural change. The regime breakpoints may be known and specified *a priori*, or they may be estimated using the Bai (1997), Bai and Perron (1998), and related techniques.

Note that much of the material in this chapter parallels the closely related subject of testing for multiple breakpoints outlined in "Multiple Breakpoint Tests" on page 230.

Background

We consider a standard multiple linear regression model with T periods and m potential breaks (producing $m+1$ regimes). For the observations $T_j, T_j+1, \ldots, T_{j+1}-1$ in regime j we have the regression model

$$y_t = X_t'\beta + Z_t'\delta_j + \epsilon_t \tag{34.1}$$

for the regimes $j = 0, \ldots, m$. Note that the regressors are divided into two groups. The X variables are those whose parameters do not vary across regimes, while the Z variables have coefficients that are regime-specific.

While it is slightly more convenient to define breakdates to be the last date of a regime, we follow EViews's convention in defining the breakdate to be the first date of the subsequent regime. We tie down the endpoints by setting $T_0 = 1$ and $T_{m+1} = T+1$.

Once the number and identity of the breakpoints is determined, the model may be estimated using standard regression techniques. We may rewrite the equation specification as a standard regression equation

$$y_t = X_t'\beta + \bar{Z}_t'\bar{\delta} + \epsilon_t \tag{34.2}$$

with fixed parameter vectors β and $\bar{\delta} = (\delta_0', \delta_1', \ldots, \delta_m')$ where $\bar{Z}_t'$ is an expanded set of regressors interacted with the set of dummy variables corresponding to each of the $m+1$ regime segments.

The breakpoints may be known *a priori* or they be estimated using a variety of approaches. The breakpoint estimation methods that we consider may broadly be divided into two categories: global maximizers for the breakpoints and sequentially determined breakpoints.

Global Maximization

Bai and Perron (1998) describe global optimization procedures for identifying the m multiple breaks and associated coefficients which minimize the sums-of-squared residuals of the regression model Equation (34.1).

If the desired number of breakpoints is known, the global m-break optimizers are the set of breakpoints and corresponding coefficient estimates that minimize the sum-of-squares for that model.

If the desired number of breakpoints is not known, we may specify a maximum number of breakpoints and employ testing to determine the "optimal" number of breakpoints. The various test approaches are outlined in detail in "Global Maximizer Tests" on page 230, but briefly speaking, involve:

- Global tests of l breaks versus none (Bai-Perron 1998). The test of l versus no breaks procedure may be applied sequentially beginning with a single break until the null is not rejected. Alternately, it may be applied to all breaks with the selected break being the highest statistically significant number of breaks, or it may employ the unweighted or weighted double maximum statistics ($UD\max$ or $WD\max$).
- Information criteria based model selection of the number of breaks (Yao 1988; Liu, Wi, and Zidek 1997), where we minimize the specified information criteria with respect to the number of breaks.
- Sequential tests of $l+1$ versus l globally determined breakpoints. The procedure is applied sequentially, beginning with a single break, until the null is not rejected. This approach is a modified Bai (1997) method in which, at each test step, the l breakpoints under the null are obtained by global optimization, and the candidate breakpoints are obtained by sequential estimation.

Sequential Determination

Bai (1997) describes an intuitive approach for obtaining estimates for more than one break. The procedure involves *sequential* application of breakpoint tests.

- Begin with the full sample and perform a test of parameter constancy with unknown break. At each stage, test for breakpoints in breakpoint tests in each subsample. Add a breakpoint whenever a subsample null is rejected. (Alternately, one could test only the single subsample which shows the greatest improvement in the sum-of-squared residuals.) If any of the tests reject, add the specified breakpoint to the current set.

- Repeat the procedure until all of the subsamples do not reject the null hypothesis, or until the maximum number of breakpoints allowed or maximum subsample intervals to test is reached.
- Perform refinement so that breakpoints are re-estimated if they are obtained from a subsample containing more than one break. This procedure is required so that the breakpoint estimates have the same limiting distribution as those obtained from the global optimization procedure.

If the number of breakpoints is pre-specified, we simply estimate the specified number of breakpoints using the one-at-a-time method.

Estimating Least Squares with Breakpoints in EViews

To estimate an equation using least squares with breakpoints, select **Object/New Object…/ Equation** or **Quick/Estimate Equation…** from the main EViews menu, then select **BREAKLS - Least Squares with Breakpoints** in the **Method** drop-down menu, or simply type the keyword `breakls` in the command window:

You should enter the dependent variable followed by a list of variables with breaking regressors in the top edit field, and optionally, a list of non-breaking regressors in the bottom edit field.

Next, click on the **Options** tab to display additional settings for calculation of the coefficient covariance matrix, specification of the breakpoints, weighting, and the coefficient name.

The weighting and coefficient name settings are common to other EViews estimators so we focus on the covariance computation and break specification.

Coefficient Covariance Matrix

The **Coefficient** covariance matrix section of the page offers various computation settings.

The top drop-down menu should be used to specify the estimator for the coefficient covariances. The **Estimation default** is to use the conventional estimator for least squares regression. You may instead elect to use heteroskedasticity robust **White** or **HAC (Newey-West)** covariance calculations. If you specify HAC covariances, EViews will display the **HAC options** button which you may press to bring up a dialog for customizing your calculation.

By default, the covariances will be computed assuming homogeneous errors variances with a common distribution across regimes. You may use the **Allow error distributions to differ across breaks** to relax this common distribution restriction.

If you specify either the White or HAC form of robust covariance, EViews will commonly display the **Assume common data distributions** checkbox. EViews generally follow Bai and Perron (2003a) who, with one exception, do not impose the restriction that the distribution of the Z_t is the same across regimes. In cases where you employ robust variances, EViews will offer you a choice of whether to assume a common distribution for the data across regimes.

Bai and Perron do impose the homogeneity data restriction when computing HAC robust variances estimators with homogeneous errors. To match the Bai-Perron common error assumptions, you will have to select the **Assume common data distribution checkbox**.

(Note that EViews does not allow you to specify heterogeneous error distributions and robust covariances in partial breaking models.)

Break Specification

The **Break specification** section of the dialog contains a **Method** drop-down where you may specify the type of test you wish to perform. You may choose between:

- Sequential L + 1 breaks vs. L
- Sequential tests all subsets
- Global L breaks vs. none
- L + 1 breaks vs. global L
- Global information criteria
- Fixed number - sequential
- Fixed number - global
- User-specified

The first two entries determine the optimal number of breaks based on the sequential methodology as described in "Sequential Determination" on page 472 above. The methods differ in whether, for a given l breakpoints, we test for an additional breakpoint in each of the $l+1$ segments (**Sequential tests all subsets**), or whether we test the single added breakpoint that most reduces the sum-of-squares (**Sequential L + 1 breaks vs. L**).

The next three methods employ the global optimizers to determine the number and identities of breaks as described in "Global Maximization" on page 472. If you select one of the global methods, you will see a second drop-down prompting you to specify a sub-method.

- For the **Global L breaks vs. none** method, there are four possible sub-methods. The **Sequential evaluation** method chooses the last significant number of breaks, determined sequentially. Selecting **Highest significant** chooses the number of breaks that is largest from amongst the significant tests. The latter two settings choose the number of breaks using the corresponding double max test.

- Similarly, if you select the **L + 1 breaks vs. none** method, a drop-down offers a choice between **Sequential evaluation** and **Highest significant**.
- The **Global information criteria** method lets you choose between using the **Schwarz criterion** or the **LWZ criterion**.

The next two methods, **Fixed number - sequential** and **Fixed number - global**, pre-specify the number of breaks and choose the breakpoint dates using the specified method.

The **User-specified** method allows you to specify your own break dates.

Depending on your choice of method, you may be prompted to provide information on one or more additional settings:

- If you specify one of the two fixed number of break methods, you will be prompted for the number of breakpoints (not depicted).
- The **Trimming percentage,** $e = 100(h/T)$ implicitly determines h, the minimum segment length permitted when constructing a test. Small values of the trimming percentage can lead to estimates of coefficients and variances which are based on very few observations.
- The **Maximum breaks** and **Maximum levels** setting limits the number of breakpoints allowed via global testing, and in sequential or mixed l vs. $l+1$ testing.
- The **Significance level** drop-down menu should be used to choose between test size values of (0.01, 0.025, 0.05, and 0.10). This setting is not relevant for methods which do not employ testing.

Additional detail on all of the methodologies outlined above is provided in "Multiple Breakpoint Tests" on page 230.

Working with Breakpoint Equations

Before describing EViews tools for working with an equation estimated using breakpoint least squares, it is important to point out that the resulting estimated equation is simply a linear regression model in which some of the variables are interacted with regime dummy variables, as in Equation (34.2) on page 471. Thus, most of the discussion related to the lin-

ear regression model in Chapter 20. "Basic Regression Analysis," on page 5 and Chapter 26. "Specification and Diagnostic Tests," on page 195 applies. We focus our attention in this section on the unique aspects of the breakpoint equation.

Estimation Output

To illustrate the output from estimation of a breakpoint equation, we employ data from Hansen's (2001) labor productivity example. Hansen's example uses monthly (February 1947 to April 2001) U. S. labor productivity in the manufacturing durables sector as measured by the growth rate of the ratio of the Industrial Production Index to average weekly labor hours. The data are in the series DDUR in the workfile "hansen_jep.wf1".

We estimate a breakpoint model with DDUR regressed on its lag DDUR(-1) and a constant. The output is presented below:

Dependent Variable: DDUR
Method: Least Squares with Breaks
Date: 12/10/12 Time: 10:55
Sample (adjusted): 1947M03 2001M04
Included observations: 650 after adjustments
Break type: Bai-Perron tests of L+1 vs. L sequentially determined breaks
Break selection: Trimming 0.05, Max. breaks 5, Sig. level 0.10
Breaks: 1963M12, 1994M12
White heteroskedasticity-consistent standard errors & covariances
No d.f. adjustment for covariances

Variable	Coefficient	Std. Error	t-Statistic	Prob.
1947M03 - 1963M11 -- 201 obs				
DDUR(-1)	0.089257	0.097749	0.913124	0.3615
C	3.034604	1.086232	2.793697	0.0054
1963M12 - 1994M11 -- 372 obs				
DDUR(-1)	-0.240237	0.060248	-3.987492	0.0001
C	3.891848	0.553035	7.037262	0.0000
1994M12 - 2001M04 -- 77 obs				
DDUR(-1)	-0.186131	0.140463	-1.325130	0.1856
C	9.329578	1.562201	5.972073	0.0000

R-squared	0.049806	Mean dependent var	3.756325
Adjusted R-squared	0.042429	S.D. dependent var	11.15357
S.E. of regression	10.91439	Akaike info criterion	7.627229
Sum squared resid	76715.79	Schwarz criterion	7.668555
Log likelihood	-2472.849	Hannan-Quinn criter.	7.643258
F-statistic	6.751313	Durbin-Watson stat	2.027022
Prob(F-statistic)	0.000004		

The top portion of the output shows equation specification information. As we see, the two estimated breakdates 1963m12 and 1994m12 were determined using the Bai-Perron sequential breakpoint methodology, with a maximum of 5 breaks, 5% trimming, and a test size of 0.10. Coefficient covariances for the tests and estimates are computed using White's method with no d.f. correction.

The middle section labels each regime and shows the corresponding coefficient estimates, standard errors, and p-values.

The bottom portion of the dialog shows the standard summary statistics. Most are self-explanatory. We do note that the R-square, the F-statistic, and the corresponding probability are all based on a comparison with the full restricted, no breakpoint, constant only model. Note also the F-statistic is based on the difference of the sums-of-squares so, despite the presence of White coefficient standard errors, it is not robust to heteroskedasticity.

Equation Views and Procs

As noted above, estimated equation is simply a linear regression model in which some of the variables are interacted with regime dummy variables. Thus, most of the equation views and procs are defined exactly as in the standard least square regression. There are, however, some specific breakpoint regression features that deserve discussion.

Representations View

The representations view shows you the equation specification estimated by EViews:

Note in particular the use of the `@before`, `@during`, and `@after` functions to create regime dummy variables that interact with the regressors. You could have used these functions to specify an equivalent model using the ordinary least squares estimator.

Breakpoint Specification View

The breakpoint specification view (**View/Breakpoint Specification**) displays a summary of the breakpoint specification, the method used to determine the break dates, and if appropriate, re-computes and displays the test statistics used to obtain the optimal breaks.

The top portion of the output shows the breakpoint summary:

Breakpoint Specification
Description of the breakpoint specification used in estimation
Equation: BREAKEQ
Date: 12/11/12 Time: 10:09

Summary
Estimated number of breaks: 2
Method: Bai-Perron tests of L+1 vs. L sequentially determined breaks
Maximum number of breaks: 5
Breaks: 1965M08, 1991M06

The remaining portion shows the intermediate results for the breakpoint determination:

Current breakpoint calculations:

Multiple breakpoint tests

Bai-Perron tests of L+1 vs. L sequentially determined breaks
Date: 12/11/12 Time: 10:09
Sample: 1947M03 2001M04
Included observations: 650
Breakpoint variables: DDUR(-1) C
Break test options: Trimming 0.15, Max. breaks 5, Sig. level 0.05
Test statistics employ White heteroskedasticity-consistent covariances assuming common data distribution
No d.f. adjustment for covariances

Sequential F-statistic determined breaks:			2
Break Test	F-statistic	Scaled F-statistic	Critical Value**
0 vs. 1 *	8.967659	17.93532	11.47
1 vs. 2 *	7.710196	15.42039	12.95
2 vs. 3	3.817321	7.634643	14.03

* Significant at the 0.05 level.
** Bai-Perron (Econometric Journal, 2003) critical values.

Break dates:

	Sequential	Repartition
1	1982M01	1965M08
2	1991M06	1991M06

Coefficient Labeling in Various Views

EViews uses various methods for labeling the coefficient results in view output.

For example, the matrix-based coefficient covariance view (**View/Covariance matrix**) uses minimum labeling information to identify columns and rows of the covariance matrix. The coefficients are listed in the order that they appear in the equation results view, but without overt regime identification:

	DDUR(-1)	C	DDUR(-1)	C	DDUR(-1)	C
DDUR(-1)	0.009148	-0.054648	0.000000	0.000000	0.000000	0.000000
C	-0.054648	1.042239	0.000000	0.000000	0.000000	0.000000
DDUR(-1)	0.000000	0.000000	0.004145	-0.018084	0.000000	0.000000
C	0.000000	0.000000	-0.018084	0.368065	0.000000	0.000000
DDUR(-1)	0.000000	0.000000	0.000000	0.000000	0.012956	-0.084330
C	0.000000	0.000000	0.000000	0.000000	-0.084330	1.264747

Here, the first to columns correspond to the coefficients on DDUR(-1) and the intercept in the first regime, the next two columns are the results for the second regime, and so on. Any non-regime specific coefficients appear at the end of the blocks of varying coefficients.

In most table output, EViews groups and labels the variables by regimes. For example, the scaled coefficients view (**View/Coefficient Diagnostics/Scaled Coefficients**) mirrors the format of the equation results output:

Scaled Coefficients
Date: 12/10/12 Time: 12:21
Sample: 1947M01 2001M04
Included observations: 650

Variable	Coefficient	Standardized Coefficient	Elasticity at Means
1947M03 - 1965M07 -- 221 obs			
DDUR(-1)	0.081718	0.057258	0.026049
C	3.256673	0.138422	0.294775
1965M08 - 1991M05 -- 310 obs			
DDUR(-1)	-0.239991	-0.150939	-0.082734
C	3.342257	0.149785	0.424351
1991M06 - 2001M04 -- 119 obs			
DDUR(-1)	-0.209215	-0.091133	-0.071046
C	8.383666	0.290912	0.408606

Alternately, in leverage plots (**View/Stability Diagnostics/Leverage Plots...**) EViews displays graphs that are labeled with the full dummy variable interaction variables previously seen in the representations view:

Specifying Regime-Specific or Common Variables

EViews dialogs will prompt you, where relevant, to indicate whether variables should have regime-specific or common coefficients. For example, the omitted variables test (**View/Coefficient Diagnostics/Omitted Variables Test - Likelihood Ratio...**) dialog offers separate edit fields corresponding to the two types of variables:

Similarly, if you select **View/Stability Diagnostics/Leverage Plots...**, EViews will prompt you to identify the variables for which you wish to display diagnostics.

Notice that in both cases, you should specify these variables in terms of the original, non-breaking variables.

Forecasting Proc

Forecasting in breakpoint least squares works the same way as in the least squares estimator. Click on the **Forecast** button on the toolbar or select **Proc/Forecast...** to bring up the dialog.

All of the settings are as in the standard linear regression case. However, it is worth noting that the saved forecast S.E. assumes a common variance across regimes even if you have relaxed this assumption in the computation of the coefficient covariances. Note also that out-of-sample forecasts will use the first regime specific coefficients for periods prior to the estimation period, and the last regime specific coefficients for periods after the estimation period.

Example

To illustrate the use of these tools in practice, we employ the simple model of the U.S. *ex-post* real interest rate from Garcia and Perron (1996) that is used as an example by Bai and Perron (2003a). The data, which consist of observations for the three-month treasury rate deflated by the CPI for the period 1961q1–1983q3, are provided in the series RATES in the workfile "realrate.WF1".

Select **Object/New Object…/Equation** or **Quick/Estimate Equation…** from the main menu or enter the command **breakls** in the command line and hit Enter.

The regression model consists of a regime-specific constant regressor so we enter the dependent variable RATES and C in the topmost edit field. The sample is set to the full workfile range.

Next, click on the **Options** tab and specify **HAC (Newey-West)** standard errors, check **Allow error distributions to differ across breaks**, choose the **Bai-Perron Global L breaks vs. none** method using the **Unweighted-Max F (UDMax)** test to determine the number of breaks, and set a **Trimming percentage** of 15, and a **Significance level** of 0.05.

Lastly, to match the test example in Bai and Perron (2003a), we click on the **HAC Options** button and set the options to use a **Quadratic-Spectral** kernel with **Andrews automatic** bandwidth and single pre-whitening lag:

Click on **OK** to accept the settings and estimate the model. EViews displays the results of the breakpoint selection and coefficient estimation:

Dependent Variable: RATES
Method: Least Squares with Breaks
Date: 12/11/12 Time: 14:38
Sample: 1961Q1 1986Q3
Included observations: 103
Break type: Bai-Perron tests of 1 to M globally determined breaks
Break selection: Unweighted max-F (UDmax), Trimming 0.15, Max. breaks 5, Sig. level 0.05
Breaks: 1980Q4
HAC standard errors & covariance (Prewhitening with lags = 1, Quadratic-Spectral kernel, Andrews bandwidth)
Allow heterogeneous error distributions across breaks

Variable	Coefficient	Std. Error	t-Statistic	Prob.
	1961Q1 - 1980Q3 -- 79 obs			
C	0.078612	0.404959	0.194122	0.8465
	1980Q4 - 1986Q3 -- 24 obs			
C	5.642890	0.608843	9.268219	0.0000

R-squared	0.469105	Mean dependent var	1.375142
Adjusted R-squared	0.463849	S.D. dependent var	3.451231
S.E. of regression	2.527072	Akaike info criterion	4.711226
Sum squared resid	644.9955	Schwarz criterion	4.762386
Log likelihood	-240.6282	Hannan-Quinn criter.	4.731948
F-statistic	89.24490	Durbin-Watson stat	1.382941
Prob(F-statistic)	0.000000		

The $UDMax$ methodology selects a single statistically significant break at 1980Q4. The results clearly show a significant difference in the mean RATES prior to and after 1980Q4.

Click on **View/Actual, Fitted, Residual/Actual, Fitted, Residual Graph**, to see in-sample fitted data alongside the original series and the residuals:

Casual inspection of the residuals suggests that the model might be improved with the addition of another breakpoint in the early 1970s. Click on the **Estimate** button, select the **Options** tab, and modify the **Method** to use the **Global information criteria** with **LWZ criterion**. Click on **OK** to re-estimate the equation using the new method.

EViews reports new estimates featuring two breaks(1972Q4, 1980Q4) defining a medium, low, and a high rate regime, respectively:

Dependent Variable: RATES
Method: Least Squares with Breaks
Date: 12/11/12 Time: 15:39
Sample: 1961Q1 1986Q3
Included observations: 103
Break type: Compare information criteria for 0 to M globally determined breaks
Break selection: LWZ criterion, Trimming 0.15, Max. breaks 5
Breaks: 1972Q4, 1980Q4
HAC standard errors & covariance (Prewhitening with lags = 1, Quadratic-Spectral kernel, Andrews bandwidth)
Allow heterogeneous error distributions across breaks

Variable	Coefficient	Std. Error	t-Statistic	Prob.
	1961Q1 - 1972Q3 -- 47 obs			
C	1.355037	0.157766	8.588880	0.0000
	1972Q4 - 1980Q3 -- 32 obs			
C	-1.796138	0.518593	-3.463485	0.0008
	1980Q4 - 1986Q3 -- 24 obs			
C	5.642890	0.611880	9.222223	0.0000

R-squared	0.624708	Mean dependent var	1.375142
Adjusted R-squared	0.617202	S.D. dependent var	3.451231
S.E. of regression	2.135299	Akaike info criterion	4.383784
Sum squared resid	455.9502	Schwarz criterion	4.460524
Log likelihood	-222.7649	Hannan-Quinn criter.	4.414866
F-statistic	83.22967	Durbin-Watson stat	1.942392
Prob(F-statistic)	0.000000		

The corresponding actual, fitted, residual plot is given by

References

Bai, Jushan (1997). "Estimating Multiple Breaks One at a Time," *Econometric Theory*, 13, 315–352.

Bai, Jushan and Pierre Perron (1998). "Estimating and Testing Linear Models with Multiple Structural Changes," *Econometrica*, 66, 47–78.

Bai, Jushan and Pierre Perron (2003a). "Computation and Analysis of Multiple Structural Change Models," *Journal of Applied Econometrics*, 6, 72–78.

Bai, Jushan and Pierre Perron (2003b). "Critical Values for Multiple Structural Change Tests," *Econometrics Journal*, 18, 1–22.

Garcia, Rene and Pierre Perron (1996). "An Analysis of the Real Interest Rate Under Regime Shifts," *The Review of Economics and Statistics*, 78, 111–125.

Hansen, Bruce E. (2001). "The New Econometrics of Structural Change: Dating Breaks in U.S. Labor Productivity," *Journal of Economic Perspectives*, 15, 117–128.

Liu, Jian, Wu, Shiying, and James V. Zidek (1997). "On Segmented Multivariate Regression," *Statistica Sinica*, 7, 497–525.

Perron, Pierre (2006). "Dealing with Structural Breaks," in *Palgrave Handbook of Econometrics, Vol. 1: Econometric Theory*, T. C. Mills and K. Patterson (*eds.*). New York: Palgrave Macmillan.

Yao, Yi-Ching (1988). "Estimating the Number of Change-points via Schwarz' Criterion," *Statistics & Probability Letters*, 6, 181–189.

Chapter 35. Discrete Threshold Regression

The discrete Threshold Regression (TR) model describes a simple form of nonlinear regression featuring piecewise linear specifications and regime switching that occurs when an observed variable crosses unknown thresholds. TR specifications are quite popular as they are easy to estimate and interpret, and able to produce interesting nonlinearities and rich dynamics. Among the applications of TR are models for sample splitting, multiple equilibria, and the very popular Threshold Autoregression (TAR) and self-exciting Threshold Autoregression (SETAR) specifications (Hansen 1999, 2011; Potter 2003).

This chapter describes tools for estimating TR models with known or unknown thresholds. Among the powerful features are model selection tools for selecting the best threshold variable from a candidate list, and the ability to specify both regime varying and non-varying variables. You may, for example, easily specify a two-regime SETAR model and allow EViews to estimate the optimal delay parameter, threshold values, and coefficients and covariance estimates for the varying and regression parameters.

Background

We begin with a standard multiple linear regression model with T observations and m potential thresholds (producing $m+1$ regimes). (While we will use t to index the T observations, there is nothing in the structure of the model that requires time series data.)

For the observations in regime $j = 0, 1, \ldots, m$ we have the linear regression specification

$$y_t = X_t'\beta + Z_t'\delta_j + \epsilon_t \tag{35.1}$$

Note that the regressors are divided into two groups. The X variables are those whose parameters do not vary across regimes, while the Z variables have coefficients that are regime-specific.

Suppose that there is an observable *threshold variable* q_t and strictly increasing *threshold values* $(\gamma_1 < \gamma_2 < \ldots < \gamma_m)$ such that we are in regime j if and only if:

$$\gamma_j \leq q_t < \gamma_{j+1}$$

where we set $\gamma_0 = -\infty$ and $\gamma_{m+1} = \infty$. Thus, we are in regime j if the value of the threshold variable is at least as large as the j-th threshold value, but not as large as the $(j+1)$-th threshold. (Note that we follow EViews convention by defining the threshold values as the first values of each regime.)

For example, in the single threshold, two regime model, we have:

$$\begin{aligned} y_t &= X_t'\beta + Z_t'\delta_1 + \epsilon_t \qquad \text{if } -\infty < q_t < \gamma_1 \\ y_t &= X_t'\beta + Z_t'\delta_2 + \epsilon_t \qquad \text{if } \gamma_1 \leq q_t < \infty \end{aligned} \tag{35.2}$$

Using an indicator function $1(\cdot)$ which takes the value 1 if the expression is true and 0 otherwise and defining $1_j(q_t, \gamma) = 1(\gamma_j \leq q_t < \gamma_{j+1})$, we may combine the $m+1$ individual regime specifications into a single equation:

$$y_t = X_t'\beta + \sum_{j=0}^{m} 1_j(q_t, \gamma) \cdot Z_t'\delta_j + \epsilon_t \tag{35.3}$$

The identity of the threshold variable q_t and the regressors X_t and Z_t will determine the type of TR specification. If q_t is the d-th lagged value of y, Equation (35.3) is a self-exciting (SE) model with *delay* d; if it's not a lagged dependent, it's a conventional TR model. If the regressors X_t and Z_t contain only a constant and lags of the dependent variable, we have an autoregressive (AR) model. Thus, a SETAR model is a threshold regression that combines an autoregressive specification with a lagged dependent threshold variable.

Given the threshold variable and the regression specification in Equation (35.1), we wish to find the coefficients δ and β, and usually, the threshold values γ. We may also use model selection to identify the threshold variable q_t.

Nonlinear least squares is an natural approach for estimating the parameters of the model. If we define the sum-of-squares objective function

$$S(\delta, \beta, \gamma) = \sum_{t=1}^{T} \left(y_t - X_t'\beta - \sum_{j=0}^{m} 1_j(q_t, \gamma) \cdot Z_t'\delta_j \right)^2 \tag{35.4}$$

and we may obtain threshold regression estimates by minimizing $S(\delta, \beta, \gamma)$ with respect to the parameters.

Taking advantage of the fact that for a given γ, say $\tilde{\gamma}$, minimization of the concentrated objective $S(\delta, \beta, \tilde{\gamma})$ is a simple least squares problem, we can view estimation as finding the set of thresholds and corresponding OLS coefficient estimates that minimize the sum-of-squares across *all* possible sets of m-threshold partitions.

This basic estimation setup is well known from the breakpoint testing and regression literature (see, for example, Hansen, 2001 and Perron, 2006), and indeed, by permuting the observation index so that the threshold variable is non-decreasing, one sees that estimation of the threshold and breakpoint models are fundamentally equivalent (Bai and Perron, 2003), In essence, threshold regressions can be thought of as breakpoint least squares regressions with data reordered with respect to the threshold variable. Alternately, breakpoint regressions may be thought of as threshold regressions with time as the threshold variable.

Accordingly, the discussion of breakpoint testing ("Multiple Breakpoint Tests" on page 230) and estimation (Chapter 34. "Least Squares with Breakpoints," on page 471) may generally be applied in the current context. We will assume for our purposes that you are familiar

with, or can refer to this material, and in the interest of brevity, we will minimize the amount of repetition in our discussion below.

Threshold Estimation in EViews

To estimate a threshold regression in EViews, select **Object/New Object.../Equation** or **Quick/Estimate Equation...** from the main EViews menu, then select **Threshold - Threshold Regression** in the **Method** drop-down menu. Alternatively, type `threshold` in the command window and press **Enter**. You will see the following dialog:

There are two tabs in the threshold regression dialog: **Specification** and **Options**. We discuss each of the pages in turn.

Specification

There are three distinct sections in the threshold regression **Specification** page: **Equation specification**, **Threshold specification**, and **Sample** specification. Since the sample specification should be familiar, we will focus on the first two sections.

In the first edit field of the **Equation specification** section you should enter the dependent variable followed by a list of variables with threshold specific coefficients. The list of explanatory variables may include lagged series and ranges of lagged series specified with the word "`to`" (lag ranges are common in threshold regression models). In the second edit field, you may optionally specify a list of non-threshold varying regressors.

Next, in the **Threshold variable specification** edit field, you should a specification for one or more threshold variables. You may enter this specification as a single integer or integer pairs, or you may provide a list of variables:

- If you enter a single integer, EViews will interpret the value as the delay parameter in a SETAR model. Thus, if your dependent variable is Y and you enter a "3" in the edit field, EViews will use Y(-3) as the threshold variable.
- If you enter a single variable name, EViews will use that variable as the threshold variable. Thus, if you enter "W", EViews will estimate the specification using the series W as the threshold variable.
- If you enter one or more lag pairs, EViews will use model selection to determine the best delay parameter amongst all of the implied lag values. Thus, if you enter "1 4 7 9", EViews will estimate SETAR models with delay parameters between 1 and 4 and between 7 and 9, (threshold variables {Y(-1), Y(-2), Y(-3), Y(-4), Y(-7), Y(-8), Y(-9)}), and determine the specification that minimizes the sum-of-squared residuals.
- If you specify more than one variable, by providing a list of names, entering a group name, or using wildcard expressions, EViews will estimate TR models using each variable as the threshold variable and will employ model selection to choose the specification that minimizes the sum-of-squares.

The model selection criterion is the sum of the squares of the residuals (SSR). Also note that your threshold specification may not mix integer specifications and explicit variable lists.

In the example depicted above, we specify a threshold regime specific AR(11) specification for LYNX_TRANSF and enter the range pair "1 5" in the **Threshold variable specification** edit field. The result is a SETAR model where we will perform model selection for the threshold variable using lags of LYNX_TRANSF from -1 to -5 .

Options

The **Options** page contains additional settings for the calculation of the coefficient covariance matrix, the determination of thresholds, and the coefficient name. Most of the settings are identical to those found in breakpoint least squares, and extensive discussion may be found elsewhere ("Estimating Least Squares with Breakpoints in EViews" on page 473).

We offer a brief description of the threshold specification methods below.

Threshold Value Estimation Methods

EViews offers a number of options for choosing the number and location of the threshold values.

If the number of thresholds is known it can be entered as one of the "Fixed number" specification options.

Before describing the other methods, it is important to note that EViews uses the methodologies of Bai and Perron (1998), and not the fixed regressor bootstrap testing proposed by Hansen (1999), to determine the number of thresholds. We caution you that the approaches based on testing should be viewed as informal in the TAR setting as the lagged endogenous regressors in the model are themselves subject to structural breaks which violates the assumptions for the Sup-F statistics (Hansen, 2000; Hansen, 1999).

For a given m, global estimation of thresholds compares the SSRs for all possible sets of m threshold values. The following global methods are used to identify threshold values and the associated regression coefficients. In the first two methods the number of thresholds is unknown and user must specify the maximum number of thresholds allowed. In the last case the desired number of thresholds must be entered.

- Global L thresholds versus none
- Minimizing the information criteria
- Fixed number - global

Threshold values may also be estimated sequentially by finding an initial threshold value that minimizes the residual sums of squares, then searching for additional values (given the initial value) that minimize the SSR until the desired number of thresholds, possibly determined through testing, is obtained. Sequential tests are used in the following methods. Again, in the first two methods, the number of thresholds is not known and the user must enter the maximum number of thresholds allowed. In the last case, the user must enter the desired number of thresholds:

- Sequential $L+1$ breaks vs. L
- Sequential tests all subsets
- Fixed number - sequential

The global tests are mixed with sequential testing in the $L+1$ versus global L method.

Additional details for each of these methods may be found in the discussion of breakpoint regression ("Background," beginning on page 471) and breakpoint testing ("Multiple Breakpoint Tests," beginning on page 230).

Working with Threshold Equations

The threshold equation produced by the EViews threshold estimation procedure is, in essence, a linear regression model with regime dummies on some or all of the variables. For the most part, views and procs will work as in standard models, though some care should be taken in interpreting results.

We describe below the basics of working with your threshold equation, highlighting views and procs for which there are important differences, along with threshold specific tools.

Estimation Output

Suppose we estimate a two-regime threshold regression model with an AR(11) in each regime and model selection over threshold dependent variable lags from -1 to -5.

The top part of the output shows equation specification information.

```
Dependent Variable: LYNX_TRANSF
Method: Threshold Regression
Date: 03/05/15   Time: 10:29
Sample (adjusted): 1832 1934
Included observations: 103 after adjustments
Threshold type: Fixed number of sequentially determined thresholds
Threshold variables considered: LYNX_TRANSF(-1)
LYNX_TRANSF(-2) LYNX_TRANSF(-3) LYNX_TRANSF(-4)
LYNX_TRANSF(-5)
Threshold variable chosen: LYNX_TRANSF(-3)
Threshold value used: 3.404149
```

In addition to the usual dependent variable, method, date, and sample information, EViews displays information about the threshold specification. Here we see that the threshold value was found using the fixed number (one) of sequentially determined thresholds. Since we instructed EViews to perform model selection using 1 to 5 lags of the LYNX_TRANSF series, EViews displays the names of all of the candidate series. Lastly, EViews displays the selected threshold variable and the estimated threshold value.

Some comments on the reported threshold value are in order. Recall that the threshold values are only identified up to an interval defined by adjacent values of the sorted threshold variable (Tsay, 1989). For purposes of display, EViews reports the observed value of the threshold variable at the beginning of a regime, truncated to a more readable form, while ensuring that the representation satisfies the threshold inequalities.

The middle part of the output labels displays coefficient values and associated statistics for each regime. The bottom portion of the output contains the usual summary statistics.

Variable	Coefficient	Std. Error	t-Statistic	Prob.
LYNX_TRANSF(-3) < 3.404149 -- 78 obs				
C	0.901519	0.326423	2.761809	0.0071
LYNX_TRANSF(-1)	1.059387	0.111013	9.542922	0.0000
LYNX_TRANSF(-2)	-0.179744	0.156881	-1.145737	0.2554
LYNX_TRANSF(-3)	-0.054279	0.149394	-0.363329	0.7173
LYNX_TRANSF(-4)	-0.150385	0.150522	-0.999085	0.3208
LYNX_TRANSF(-5)	0.047953	0.155663	0.308059	0.7588
LYNX_TRANSF(-6)	-0.041777	0.155923	-0.267935	0.7894
LYNX_TRANSF(-7)	-0.036509	0.158275	-0.230671	0.8182
LYNX_TRANSF(-8)	0.159765	0.162665	0.982170	0.3290
LYNX_TRANSF(-9)	0.009293	0.162477	0.057197	0.9545
LYNX_TRANSF(-10)	0.184069	0.154652	1.190213	0.2375
LYNX_TRANSF(-11)	-0.308074	0.098137	-3.139209	0.0024
3.404149 <= LYNX_TRANSF(-3) -- 25 obs				
C	1.074224	1.128350	0.952031	0.3440
LYNX_TRANSF(-1)	1.625837	0.191162	8.505018	0.0000
LYNX_TRANSF(-2)	-1.980078	0.307855	-6.431855	0.0000
LYNX_TRANSF(-3)	1.512621	0.445142	3.398063	0.0011
LYNX_TRANSF(-4)	-1.033482	0.444150	-2.326876	0.0225
LYNX_TRANSF(-5)	0.755411	0.357410	2.113569	0.0377
LYNX_TRANSF(-6)	0.424123	0.411788	1.029955	0.3062
LYNX_TRANSF(-7)	-0.946509	0.438604	-2.158001	0.0340
LYNX_TRANSF(-8)	-0.086284	0.278625	-0.309677	0.7576
LYNX_TRANSF(-9)	0.414445	0.257290	1.610811	0.1112
LYNX_TRANSF(-10)	0.141073	0.242708	0.581244	0.5627
LYNX_TRANSF(-11)	-0.242090	0.173382	-1.396282	0.1665

R-squared	0.928938	Mean dependent var	2.879249
Adjusted R-squared	0.908249	S.D. dependent var	0.562305
S.E. of regression	0.170325	Akaike info criterion	-0.501483
Sum squared resid	2.291827	Schwarz criterion	0.112434
Log likelihood	49.82638	Hannan-Quinn criter.	-0.252825
F-statistic	44.90026	Durbin-Watson stat	2.109955
Prob(F-statistic)	0.000000		

Most of the summary statistics are self-explanatory. We do note that the R-square, the F-statistic, and the corresponding probability are all based on a comparison with the fully restricted, no threshold, constant only model.

Equation Views and Procs

Not surprisingly, the available views and procs for threshold equations parallel those in breakpoint least squares regression, with some additions as described below. Additionally we address some subtle and not-so-subtle issues that you should bear in mind when working with the familiar routines.

Criteria Graph and Table

If you select **View/Model Selection Summary** from an estimated threshold equation you will be offered a choice of displaying a **Criteria Graph** or a **Criteria Table**:

These two views display the model selection criteria used to select the threshold variable in a line plot or a table, ordered by the selection criterion.

For example, the criteria graph for this equation is shown below:

In this figure, the threshold variable whose model has the lowest AIC is clearly visible on the left of the graph.

Here, we see the same set of results in table form. This view also includes information about the common sample used for model selection estimation, and the number of regimes employed for each candidate model.

Threshold Variable	SSR	Regimes
LYNX_TRANSF(-3)	2.291827	2
LYNX_TRANSF(-2)	2.528967	2
LYNX_TRANSF(-4)	2.733194	2
LYNX_TRANSF(-1)	2.799304	2
LYNX_TRANSF(-5)	3.041367	2

Representations View

The representations view (**View/Representations**) shows the expanded equation specification, which combines the coefficients from different regimes with the threshold variable and limits and various inequalities into a single equation.

Note that estimating this single equation specification via ordinary least squares will produce the same coefficients as the estimated threshold model.

Threshold Specification

The threshold specification view displays more detailed information about the threshold variable, values, along with information about the method of selecting the number of thresholds. To display this view, click on **View/Threshold Specification** from the equation menu.

The top portion of the output displays information about the threshold and threshold values:

```
Threshold Specification
Description of the threshold specification used in estimation
Equation: TAR
Date: 03/05/15   Time: 11:28
==============================================================
                          Summary
==============================================================
Threshold variable: LYNX_TRANSF(-3)
Specified number of thresholds: 1
Method: Fixed number of sequentially determined thresholds
Threshold data value: 3.40414924921
Adjacent data value: 3.39984671271
Threshold value used: 3.404149
==============================================================
```

The detailed information on the threshold values includes the actual data value corresponding to the break (in this case 3.40414924921), the actual data value for the next highest data value (here 3.39984671271), and the truncated value EViews uses for display and representation purposes (3.404149). (Note that any value between the lower adjacent data value and the threshold data value would produce the same observed fit).

The lower portion of the output displays calculations used in determining the thresholds:

Current threshold calculations:

Multiple threshold tests
Bai-Perron tests of L+1 vs. L sequentially determined thresholds
Date: 03/05/15 Time: 11:28
Sample: 1832 1934
Included observations: 103
Threshold variable: LYNX_TRANSF(-3)
Threshold varying variables: C LYNX_TRANSF(-1) LYNX_TRANSF(-2) LYNX_TRANSF(-3) LYNX_TRANSF(-4) LYNX_TRANSF(-5) LYNX_TRANSF(-6) LYNX_TRANSF(-7) LYNX_TRANSF(-8) LYNX_TRANSF(-9) LYNX_TRANSF(-10) LYNX_TRANSF(-11)
Threshold test options: Trimming 0.15, Max. thresholds 5, Sig. level 0.05

Sequential F-statistic determined thresholds:			2
Threshold Test	F-statistic	Scaled F-statistic	Critical Value**
0 vs. 1 *	4.201074	50.41289	27.03
1 vs. 2 *	2.900849	34.81019	29.24
2 vs. 3	1.528676	18.34411	30.45

* Significant at the 0.05 level.
** Bai-Perron (Econometric Journal, 2003) critical values.

Threshold values:

	Sequential	Repartition
1	3.404149	2.328379
2	2.328379	3.404149

In this case, EViews displays the results for sequentially determined thresholds using the Bai-Perron Sup-F test statistics. We caution again that since this TAR specification contains lagged endogenous variables, that the conditions required for distributional results is violated (Hansen, 1999; Hansen, 2000).

Omitted and Redundant Variables Testing

The omitted variables test (**View/Coefficient Diagnostics/Omitted Variables Test/Likelihood Ratio...**), requires the specification of regime-specific or common variables:

Simply enter the variables you wish to add in the appropriate edit field.

Alternately, the redundant variables test will prompt you to enter variables from the original specification that you wish to drop.

We point out that these tests are performed conditionally on the thresholds identified in the estimation step. EViews will use the threshold variables and values previously determined and perform the test on the conditional linear specification. This may not be the test you wish to perform.

Forecasting

Static or one-step ahead forecasting from a TR estimated equation is straightforward, and involves conditioning on the observed regressors, including any lagged endogenous and computing the forecast.

For TAR and other models with lagged endogenous variables, n-step ahead nonlinear dynamic forecasting is considerably trickier (see, for example the discussion in Potter 1999, or Tong and Lim, 1980 who distinguish between the "eventual forecasting function" and the n-step ahead forecasts). For dynamic threshold regression models, EViews computes forecasts by stochastic simulation with the forecasts and forecast standard errors obtained from the sample average and standard deviation of the simulated values.

To perform the forecast simulation, click on the **Forecast** button on the equation toolbar or select **Proc/Forecast...** from the equation menu to display the dialog:

Most of this **Forecast** dialog should be familiar. Note, however, that under the **Method** section are two new edit fields for controlling the stochastic simulation, one for the number of **Repetitions**, and the second for the **% failed reps before halting**.

Change the settings as desired and click on **OK** to continue. Here we tell EViews we wish to compute the simulated dynamic forecasts from 1900 to 1934. EViews displays the results of the forecast along with evaluation statistics:

We may use the saved values of the forecast and standard error series to show a comparison with the actuals. First, display a graph containing the series using the command:

```
plot lynx_transf lynx_transfcst lynx_transfcst+2*lynx_transfse
    lynx_transfcst-2*lynx_transfse
```

After a bit of editing to change line colors, patterns, and legend entries, we have:

References

Bai, Jushan and Pierre Perron (1998). "Estimating and Testing Linear Models with Multiple Structural Changes," *Econometrica*, 66, 47–78.

Bai, Jushan, and Pierre Perron (2003). "Computation and Analysis of Multiple Structural Change Models." *Journal of Applied Econometrics* 18(1), 1–22.

Hansen, Bruce (1999). "Testing for Linearity." *Journal of Economic Surveys,* 13, 551-576.

Hansen, Bruce (2000). "Testing for Structural Change in Conditional Models." *Journal of Econometrics*, 97, 93-115.

Hansen, Bruce (2011). "Threshold Autoregression in Economics." Statistics and Its Interface," 4, 123–127.

Potter, Simon (1999). "Nonlinear Time Series Modelling: An Introduction." *Journal of Economic Surveys* 13, 505–528.

Tong, H. and K. S. Lim (1980). "Threshold Autoregression, Limit Cycles and Cyclical Data," *Journal of the Royal Statistical Society. Series B (Methodological)*, 42, 245–292.

Tsay, Ruey S. (1989). "Testing and Modeling Threshold Autoregressive Processes," *Journal of the American Statistical Association*, 84, 231–240.

Chapter 36. Smooth Transition Regression

Smooth Transition Autoregressive (STAR) modeling (Teräsvirta, 1994) is an extremely popular approach for nonlinear time series analysis. STAR models, which are a special case of Smooth Transition Regression (STR) models, embed regime-dependent linear auto-regression specifications in a smooth transition nonlinear regression framework.

Similar in concept to discrete Threshold Regression (TR) models, STR models differ in that regime switching occurs smoothly when an observed variable crosses the unobserved thresholds. As a result of this smooth response, STR models are often thought to have more "realistic" dynamics than their discrete TR counterparts. STR models have been applied to a wide range of datasets, from macroeconomic (Teräsvirta and Anderson 1992, Teräsvirta 1994) to the well-known Canadian lynx data (Teräsvirta, 1994).

This chapter describes EViews tools for estimation of two-regime STR models with unknown parameters for the shape and location of the smooth threshold. EViews estimation supports several different transition functions, provides model selection tools for selecting the best threshold variable from a candidate list, and offers the ability to specify regime varying and non-varying variables and variables that appear in only one regime.

Following estimation, EViews offers specialized views of the transition function and weights and offers tests for linearity against STR alternatives and tests of no remaining nonlinearity and parameter constancy, alongside conventional tests for heteroskedasticity and serial correlation.

Portions of this chapteroffer brief background for STR models. Those desiring additional detail should consult the more detailed discussions in Teräsvirta (1994) and van Dijk, Teräsvirta, and Franses (2002), and the textbook treatment in Martin, Hurn, and Harris (2013, p. 720–726, 744–745).

Background

We begin with a slightly modified version of Equation (33.3) defining the discrete TR model:

$$y_t = \sum_{j=0}^{m-1} 1_j(s_t; c, \gamma) \cdot Z_t{}'\delta_j + X_t\alpha + \epsilon_t \tag{36.1}$$

where $1_j(\cdot)$ is a (0, 1) indicator for regime that depends on the observed variable s_t, where c represents one or more thresholds and $\gamma > 0$ is a threshold slope parameter. Note that we have divided the regressors into two groups— Z variables whose coefficients vary across the m regimes, and X variables with coefficients that are regime invariant.

Restricting ourselves to $m = 2$, as using the fact that $1_j(\cdot) = 1$ for exactly one j, we may rewrite the discrete TR equation as:

$$\begin{aligned} y_t &= 1_0(s_t; c, \gamma) \cdot Z_t'\delta_0 + 1_1(s_t; c, \gamma) \cdot Z_t'\delta_1 + X_t\alpha + \epsilon_t \\ &= (1 - 1_1(s_t; c, \gamma)) \cdot Z_t'\delta_0 + 1_1(s_t; c, \gamma) \cdot Z_t'\delta_1 + X_t\alpha + \epsilon_t \end{aligned} \tag{36.2}$$

To construct the two-regime STR model, we replace the indicator function with a continuous transition function G that returns values between 0 to 1. Then we have

$$y_t = (1 - G(s_t; c, \gamma)) \cdot Z_t'\delta_0 + G(s_t; c, \gamma) \cdot Z_t'\delta_1 + X_t\alpha + \epsilon_t \tag{36.3}$$

where G has different properties as $s \to -\infty$, $s \to \infty$, and $s = c$, depending on the specific functional form.

The key modeling choices in Equation (36.3) are the choice of the threshold variable s and the selection of a transition function G. For a given s and G, we may estimate the regression parameters $(\delta_0, \delta_1, \alpha)$ and the threshold values and slope (c, γ) via nonlinear least squares. Additionally, given a list of candidate variables for s, we can select a threshold variable using model selection techniques.

Common transition function choices are given by:

- Logistic (LSTR):

$$G(s; c, \gamma) = \frac{1}{1 + \exp(-\gamma(s - c))} \tag{36.4}$$

- Normal (NSTR or STR):

$$G(s; c, \gamma) = \int_{-\infty}^{\gamma(s-c)} \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{x^2}{2}\right) dx = \Phi(\gamma(s - c)) \tag{36.5}$$

- Exponential (ESTR):

$$G(s; c, \gamma) = 1 - \exp(-\gamma(s - c)^2) \tag{36.6}$$

- Logistic, second-order (L2STR):

$$G(s; c, \gamma) = \frac{1}{1 + \exp(-\gamma(s - c_1)(s - c_2))} \tag{36.7}$$

for $\gamma > 0$ in all cases.

Some brief comments about properties of the specific transition functions:

- The logistic and normal transition functions are monotonically increasing in s so that the two regimes correspond to high and low values of the threshold variable. The threshold value c determines the point at which the regimes are equally weighted, while γ controls the speed and smoothness of the transition. As $\gamma \to \infty$, the transi-

tion function approaches the indicator function and the model approaches the discrete threshold model.

- The exponential transition function is increasing in absolute deviations of s from the threshold c. Furthermore, $G(s; c, \gamma) = 0$ when $s = c$, and G approaches 1 as $s \to \infty$ and $s \to -\infty$. The ESTR model does not nest the discrete TR model since, as $\gamma \to 0$ or $\gamma \to \infty$, the specification becomes linear since G approaches a constant function returning 0 or 1.
- The second-order logistic function does nest the discrete TR model. As $\gamma \to \infty$ with distinct $c_1 < c_2$, G approaches 1 for $s < c_1$ and $s > c_2$, and G approaches 0 for s in-between. Thus, the L2STR model nests a three-regime discrete TR model where the outer regimes have a common linear specification.

  For finite γ, G attains its minimum value at $s = (c_1 + c_2)/2$ with a non-zero value.

For further discussion of the implications of various choices for G, see van Dijk, Teräsvirta, and Franses (2002) who offer extensive commentary on the properties of these transition functions and provide concrete examples of their use in empirical settings.

Following estimation of the coefficients using nonlinear least squares (van Dijk, Terasvirt-Teräsvirta, and Franses, 2002; Leybourne, Newbold and Vougas, 1998), we may estimate the coefficient covariance matrix using conventional methods available for nonlinear least squares (ordinary, heteroskedasticity consistent, heteroskedasticity and autocorrelation consistent).

The general form of the NLLS covariance is given by:

$$\hat{\Sigma}_{NLLS} = cA^{-1}BA^{-1} \tag{36.8}$$

where A is an estimate of the information, B is the variance of the residual weighted gradients, and c is a scale parameter.

For the ordinary covariance estimator, we assume that $A = B$. Then we have

$$\hat{\Sigma}_{NLLS} = cA^{-1} \tag{36.9}$$

where c is an estimator of the residual variance (with or without degree-of-freedom correction).

We may estimate A using the outer-product of the gradients (OPG) of the error function or the one-half of the Hessian matrix of second derivatives of the sum-of-squares function:

$$\hat{\Sigma}_{NLLS} = c \cdot \left(\frac{\partial f(\beta)'}{\partial \beta}\frac{\partial f(\beta)}{\partial \beta}\right)^{-1} \tag{36.10}$$

or

$$\hat{\Sigma}_{NLLS} = c \cdot \left(\frac{1}{2} \cdot \frac{\partial^2 S(\beta)}{\partial\beta\partial\beta'}\right)^{-1} \tag{36.11}$$

evaluated at the estimated coefficients.

Alternately, we may assume distinct A and B, and employ a White or HAC sandwich estimator for the coefficient covariance as in "Robust Standard Errors," beginning on page 32 of *User's Guide II*.

Lastly, following specification and estimation of an STR equation, the modeling cycle typically involves performing a number of hypothesis tests using the STR specification and results. Background information for the various tests performed by EViews will be discussed below.

Estimating a Smooth Transition Regression in EViews

Estimation of a STR model is performed using the same dialog as discrete Threshold Regression. Open an existing equation or select **Quick/Estimate Equation...** from the main EViews menu, select **THRESHOLD - Threshold Regression** from the main **Method** dropdown menu near the bottom of the dialog, and click on the **Smooth** radio button in the **Threshold type** setting.

EViews will display a dialog with two tabs: the **Specification** tab and the **Options** tab.

Specification

Focusing first on the **Specification** page, we see that it is virtually identical to the dialog for discrete threshold models. There are again three sections: **Equation specification**, **Threshold specification**, and **Sample** specification. As the Sample specification should be familiar, we focus our attention on the first two sections.

Equation Specification

The **Equation specification** section has two edit fields, one for the dependent variable and the threshold varying regressors, and one for threshold non-varying regressors.

In the first edit field, you should enter the dependent variable, followed by a list of Z variables with threshold specific coefficients. The list of explanatory variables may include lagged series and ranges of lagged series specified with the word "`to`" (lag ranges commonly appear in STAR models).

You may employ two varying regressor variable modifiers, `@base`, and `@alt` to specify variables which only appear in the base or in the alternative specification, respectively. For example, adding the varying regressors specification

```
@alt(c) @alt(ys(-7))
```

indicates that the Z variables C and YS(-7) should only appear in the portion of the specification weighted by G, or equivalently, that the elements of δ_0 corresponding to those two variables will be zero. Note that you may include more than one variable inside a `@base` or `@alt` tag.

In the second edit field, you may optionally specify a list of threshold non-varying X regressors corresponding with coefficients α.

Threshold Specification

Next, in the **Threshold variable specification** edit field, you should enter a specification for the threshold variables. You may enter this specification as a single integer or integer pairs, or you may provide a list of variables:

- If you enter a single integer, EViews will interpret the value as the delay parameter in a self-exciting model. Thus, if your dependent variable is Y and you enter a "3" in the edit field, EViews will use Y(-3) as the threshold variable.
- If you enter a single variable name, EViews will use that variable as the threshold variable. Thus, if you enter "W", EViews will estimate the specification using the series W as the threshold variable.
- If you enter one or more lag pairs, EViews will use model selection to determine the best delay parameter amongst all of the implied lag values. Thus, if you enter "1 4 7 9", EViews will estimate self-exciting STR models with delay parameters between 1

and 4 and between 7 and 9, (threshold variables {YS(-1), YS(-2), YS(-3), YS(-4), YS(-7), YS(-8), YS(-9)}), and determine the specification that minimizes the sum-of-squared residuals.

- If you specify more than one variable by providing a list of names, entering a group name, or using wildcard expressions, EViews will estimate a STR model using each candidate threshold variable and will choose the specification that minimizes the sum-of-squares.

Note that your threshold specification may not mix integer specifications and explicit variable lists.

Options

The options dialog for STR models differs substantively from its discrete counterpart.

The most important STR options are in the **Smooth threshold specification**, and the **Coefficient covariance matrix** settings:

- The **Smooth threshold specification** should be used to select the form of the transition function (see "Background" on page 507). By default, EViews uses the **Logistic**, but you may use the dropdown to select the **Exponential**, **Normal**, or **Logistic - 2nd Order** functions.

- By default, EViews computes standard estimates of the coefficient covariance using the d.f. corrected estimator of the residual variance and the inverse outer-product of

the gradients. The **Coefficient covariance matrix** may be used to modify the estimator to use the Hessian, or to compute White or HAC covariance estimates.

Estimation of smooth transition models involves nonlinear optimization. The settings for **Starting parameter values** and **Estimation algorithm** control the optimization starting values and method:

- The **Starting parameter values** may be used to choose between different starting value methods.

  The default method is **Grid search - concentrated coefs**, in which the regression coefficients are concentrated out of the likelihood and grid search is performed over the transition parameters. Alternately, you may choose **Grid search - zero coefs** which is analogous to the first method, but sets the regression coefficients to zero, **Data-moments and OLS** which uses OLS regression on the linear portion of the specification, sets the nonlinear coefficient values to zero, and uses data moments to obtain starting values for the slope and threshold, or **User-specified**. The latter will use the coefficients in the coefficient vector specified in the **Coefficient name** edit field.

- The **Estimation algorithm** section offers standard nonlinear least squares optimization settings. By default, EViews uses BFGS with Marquardt steps and sets the maximum iterations and convergence tolerance to their global defaults.

Lastly, EViews offers an option to control the behavior of threshold variable model selection when it encounters models that do not converge. By default, EViews will drop those models from consideration, but you may use the **Include nonconverged models** checkbox to force inclusion.

Working with Smooth Threshold Equations

The smooth threshold equation estimated by EViews is a particular nonlinear regression specification. Accordingly, EViews supports the most of the views and procs for a nonlinear regression equation, alongside a number of tools that are specific to STR regression.

We describe below the basics of working with your STR, highlighting views and procs that are specific to STR and those which differ substantively from the standard nonlinear regression implementation.

Estimation Output

Following estimation, EViews displays the STR output. For example, we might have:

Dependent Variable: YS
Method: Smooth Threshold Regression
Transition function: Exponential
Date: 05/06/17 Time: 09:38
Sample (adjusted): 1869 1988
Included observations: 120 after adjustments
Threshold variable: YS(-3)
Starting values: Grid search with concentrated regression coefficients
Ordinary standard errors & covariance using outer product of gradients
Convergence achieved after 9 iterations

Variable	Coefficient	Std. Error	t-Statistic	Prob.
Threshold Variables (linear part)				
YS(-1)	2.609656	1.302386	2.003750	0.0477
YS(-2)	1.190593	0.557769	2.134564	0.0351
YS(-3)	1.733120	1.224012	1.415934	0.1598
YS(-4)	0.385758	0.402546	0.958295	0.3401
YS(-5)	-0.917191	0.437967	-2.094198	0.0387
YS(-6)	-0.529917	0.319152	-1.660392	0.0998
Threshold Variables (nonlinear part)				
YS(-1)	-2.618711	1.304748	-2.007062	0.0473
YS(-2)	-1.243749	0.560841	-2.217650	0.0288
YS(-3)	-1.975486	1.221529	-1.617224	0.1089
YS(-4)	-0.454079	0.422803	-1.073972	0.2853
YS(-5)	0.935453	0.461735	2.025952	0.0453
YS(-6)	0.587556	0.339849	1.728875	0.0868
C	0.076220	0.014286	5.335444	0.0000
YS(-7)	-0.322125	0.097125	-3.316622	0.0013
Slopes				
SLOPE	339.9926	133.2895	2.550782	0.0122
Thresholds				
THRESHOLD	-0.087539	0.011879	-7.369287	0.0000

R-squared	0.238193	Mean dependent var	0.043943
Adjusted R-squared	0.128317	S.D. dependent var	0.071276
S.E. of regression	0.066546	Akaike info criterion	-2.458292
Sum squared resid	0.460545	Schwarz criterion	-2.086626
Log likelihood	163.4975	Hannan-Quinn criter.	-2.307357
F-statistic	2.167837	Durbin-Watson stat	1.912573
Prob(F-statistic)	0.011921		

The top portion of the output displays the dependent variable, method, date, sample information, description of the threshold specification, and information about the estimation procedure. In this example, we see that the logistic STR (LSTR) is self-exciting, as the threshold variable is the first lag of the dependent variable. Estimation converged after 9 iterations,

from employed starting values obtained from grid search after concentrating out the regression coefficients. The coefficient standard errors and covariance uses the standard inverse outer product of the gradients times the d.f. corrected estimator of the residual variance.

The middle part of the output displays coefficient values and associated statistics for both the base (linear part) and alternative (nonlinear par) of the specification. EViews also reports the estimates slope ($\hat{\gamma}$) and threshold ($\hat{c}$) parameters.

The bottom portion of the output contains the usual summary statistics.Most of the summary statistics are self-explanatory. Note that the R-square, the F-statistic, and the corresponding probability are all based on a comparison with the no threshold, constant only linear model.

Equation Views and Procs

The views and procs for smooth threshold equations parallel those in nonlinear least squares regression, with some additions as documented below. We will not discuss familiar residuals and residual diagnostics, gradients and derivatives, and coefficient covariance views and procedures, but will focus instead of views which are new or differ for equations estimated by STR.

Representations View

As always, the representations view (**View/Representations**) shows the estimation command. In addition, the view depicts the estimation equation and substituted coefficients sections which show the nonlinear regression specification which combines the coefficients from different regimes with the slope and threshold variable into a single equation.

Note that estimating the estimation equation via ordinary least squares will produce the same coefficients as the estimated smooth threshold model.

Threshold Smoothing Weights

The threshold function occupies a central place in the specification of STR models. Accordingly, EViews offers easy-to-use tools for examining the weights associated with your estimated equation

Threshold Weights View

The threshold weights view allows you to examine at the shape of the smoothing function or the values of the smoothing weights for each observation in the estimation sample.

Select **View/Threshold Smoothing Weights...** to display the dialog:

By default, this view will display a **Function graph** showing the values returned by G for various values of s, and will display **Boxplots** that summarize the distribution of the observed individual threshold values and smoothing weights along borders of graph.

You may use the **Display type** dropdown menu to instead display a **Graph**, **Spreadsheet**, or **Summary Statistics** for the individual weights.

- The default **Function graph** display will show the shape of the estimated smoothing function.

  When displaying a **Function graph**, you may use the **Axis borders** dropdown to display summary graphs for the individual weights along the axis of the graph. You may choose to display a **Boxplot**, **Histogram**, or a **Kernel density**, or **None**.

  You may save the function plot data to a matrix in the workfile by providing the name of a new or existing **Output matrix** in the corresponding edit field.

- The **Graph** and **Spreadsheet** display the individual weight data for each observation in graph or spreadsheet form.

- The **Summary Statistics** setting shows descriptive statistics for the individual weights.

Save Threshold Weights Proc

The threshold weights proc lets you save the individual threshold weight values as a series in your workfile.

To save a series in the workfile containing the individual smoothing weights, click on **Proc/ Make Threshold Weights Group...** and enter a name for the weight series in the edit field. Click on **OK** to save the weight series.

Model Selection Summary

If you specify more than one potential threshold variable, EViews will estimate a STR model using each candidate threshold variable and will choose the specification that minimizes the sum-of-squares. You may examine the results of this model selection procedure using the model selection views.

To display a graph of the results, ordered from best to worst, select **View/Model Selection Summary/Criteria Graph**:

To display a table of the results, click on **View/Model Selection Summary/Criteria Table**:

Threshold Variable	SSR	Regimes
YS(-3)	0.460545	2
YS(-2)	0.486196	2
YS(-1)	0.489824	2

Stability Diagnostics

EViews provides easy-to-use tests for linearity against STR alternatives as well as tests for misspecification of the STR model by considering the hypotheses of no remaining nonlinearity, of parameter constancy, of no serial correlation, and of homoskedasticity,

Linearity Testing

We may rewrite the basic form of the STR specification Equation (36.3) on page 508 as:

$$y_t = Z_t'\delta_0 + G(s_t; c, \gamma) \cdot Z_t'\theta + X_t\alpha + \epsilon_t \tag{36.12}$$

where $\theta = \delta_1 - \delta_0$. For convenience, we will assume that $G(0, c, h_t) = 0$. This is not an important restriction as the functions above may be modified slightly to enforce this condition.

We can test for linearity under the null hypothesis using either $\gamma = 0$ or $\theta = 0$, and as it is more convenient to test for $\gamma = 0$, the literature focuses on this restriction. Note, however, that under the null hypothesis, the parameters c and θ are not identified, so that standard theory cannot be used to obtain the null distribution of the test statistic.

Luukkonen, Saikkonen, and Teräsvirta (1988) propose an approach which replaces G by a Taylor series expansion which is estimable under the null. The resulting test procedure involves taking the linear portion of the model and adding terms representing the interaction of the linear regressors with the polynomial terms in the Taylor expansion and then testing for the statistical significance of sets of the interaction coefficients.

It is worth noting that the resulting Taylor series expansion will differ depending on the form for G and that the specific terms of the Taylor expansion that are asymptotically relevant under the alternative differ across G. These differences allow us to use the probabilities of rejection of various null hypothesis to discriminate between alternative choices for G.

There is a large literature discussing the properties of these tests, offering elaboration and various refinements. For details, see Teräsvirta (1994), Eitrheim and Teräsvirta (1996), Escribano and Jorda (1999), van Dijk, Teräsvirta, and Franses (2002).

To perform these tests, simply click on **View/Stability Diagnostics/Linearity Test**. EViews will perform linearity tests against nonlinear alternatives using the selected threshold variable.

Smooth Threshold Linearity Tests
Date: 05/06/17 Time: 10:28
Sample: 1861 1988
Included observations: 120
Test for nonlinearity using YS(-3) as the threshold variable
Taylor series alternatives: b0 + b1*s [+ b2*s^2 + b3*s^3 + b4 *s^4]

Linearity Tests

Null Hypothesis	F-statistic	d.f.	p-value
H04: b1=b2=b3=b4=0	2.719220	(30, 84)	0.0002
H03: b1=b2=b3=0	2.854853	(23, 91)	0.0002
H02: b1=b2=0	3.039417	(16, 98)	0.0004
H01: b1=0	3.257240	(9, 105)	0.0016

The H0i test uses the i-th order Taylor expansion (bj=0 for all j>i).

Terasvirta Sequential Tests

Null Hypothesis	F-statistic	d.f.	p-value
H3: b3=0	1.957734	(7, 91)	0.0695
H2: b2=0 | b3=0	2.375368	(7, 98)	0.0276
H1: b1=0 | b2=b3=0	3.257240	(9, 105)	0.0016

All tests are based on the third-order Taylor expansion (b4=0).
Linear model is rejected at the 5% level using H03.
Recommended model: first-order logistic.
. Pr(H1) <= Pr(H2)

Escribano-Jorda Tests

Null Hypothesis	F-statistic	d.f.	p-value
H0L: b2=b4=0	1.394541	(13, 84)	0.1792
H0E: b1=b3=0	2.456645	(12, 84)	0.0085

All tests are based on the fourth-order Taylor expansion.
Linear model is rejected at the 5% level using H04.
Recommended model: first-order logistic with zero threshold.
. Pr(H0L) >= Pr(H0E) with Pr(H0L) >= .05 and Pr(H0E) < .05).

EViews displays results from three different sets of tests.

- The Luukkonen, Saikkonen, and Teräsvirta linearity tests at the top of the output are joint hypothesis test for significance of the elements of the Taylor expansion.

  The listed hypothesis refers to the coefficients of the expansion under test assuming that the higher order terms are restricted to zero. Imposing these restrictions may produce tests with higher power.

  For example, the H02 null hypothesis tests $b1 = b2 = 0$ assuming that $b3 = b4 = 0$.

- The Teräsvirta tests are a sequential set of general-to-specific tests for terms of the Taylor expansion that allow for transition function selection. Following Teräsvirta, EViews carries out the three tests in the following sequence:

$$\begin{aligned} H_{03}&: \quad b_3 = 0 \\ H_{02}&: \quad b_2 = 0 \mid b_3 = 0 \\ H_{01}&: \quad b_1 = 0 \mid b_2 = b_3 = 0 \end{aligned} \tag{36.13}$$

  If the null of linearity is rejected, we may use the results to determine a preferred transition function. A useful rule for discriminating between ESTAR and LSTAR models is to select the ESTAR model if the p-value for H2 is smallest, otherwise the LSTAR model is to be preferred.

- The Escribano-Jorda tests use the different properties of the Taylor expansion both to test for nonlinearity and to discriminate between alternative choices for G. The HOL test evaluates linearity against a logistic alternative, while the HOE test compares linearity to an exponential alternative.

Note that EViews computes the F-statistic forms of all of these tests.

Further, note that we assume that the threshold variable is the one used in estimation of your STR model. Thus, EViews implementation does not allow for the forms of the linearity test where one tests against multiple thresholds.

Remaining Nonlinearity Tests

Given estimation of a two-regime STR model we may wish to test for whether there is additional unmodeled nonlinearity. One popular approach is to test the estimated model against a model with additional regimes. The testing methodology is analogous to the linearity tests outlined above ("Linearity Testing" on page 519) in which we consider a Taylor series expansion of the transition weighting function, and then test for the significance of interactions with the variables of the specification.

Following the discussion in van Dijk, Teräsvirta, and Franses (2002) we distinguish between two forms of the test, the additive and the encapsulated tests, which offer different specifications of the multiple regime alternative.

Additive Multiple Regime STR (AMRSTR)

We may specify a three regime model by adding a second nonlinear component, as in:

$$y_t = Z_t'\delta_0 + G_1(s_t; c_1, \gamma_1) \cdot Z_t'\theta_1 + X_t'\alpha + G_2(s_t; c_2, \gamma_2) \cdot Z_t'\theta_2 + \epsilon_t \tag{36.14}$$

Eitrheim and Terasvirta (1998) develop LM tests to test the two-regime LSTAR model against this alternative. The test involves expanding the G_2 function in a third-order Taylor expansion and then performing a standard hypothesis test on polynomial terms interacted with Z.

To perform a remaining nonlinearity test against this three regime alternative, you should click on **View/Stability Diagnostics/Remaining Nonlinearity Test/Additive Nonlinearity Test**:

Smooth Threshold Remaining Nonlinearity Tests
Date: 05/06/17 Time: 10:45
Sample: 1861 1988
Included observations: 120
Additive nonlinearity tests using YS(-3) as the threshold variable
Taylor series alternatives: b0 + b1*s [+ b2*s^2 + b3*s^3 + b4*s^4]

Additive Nonlinearity Tests

Null Hypothesis	F-statistic	d.f.	p-value
H04: b1=b2=b3=b4=0	1.427627	(30, 74)	0.1099
H03: b1=b2=b3=0	1.401097	(23, 81)	0.1364
H02: b1=b2=0	1.461854	(16, 88)	0.1328
H01: b1=0	1.855505	(9, 95)	0.0682

The H0i test uses the i-th order Taylor expansion (bj=0 for all j>i).

Terasvirta Sequential Tests

Null Hypothesis	F-statistic	d.f.	p-value
H3: b3=0	1.207163	(7, 81)	0.3084
H2: b2=0 | b3=0	0.962351	(7, 88)	0.4637
H1: b1=0 | b2=b3=0	1.855505	(9, 95)	0.0682

All tests are based on the third-order Taylor expansion (b4=0).
Original model is not rejected at the 5% level using H03.

Escribano-Jorda Tests

Null Hypothesis	F-statistic	d.f.	p-value
H0L: b2=b4=0	1.224852	(13, 74)	0.2796
H0E: b1=b3=0	1.464289	(12, 74)	0.1574

All tests are based on the fourth-order Taylor expansion.
Original model is not rejected at the 5% level using H04.

Multiple Regime STR (MRSTR)

The Multiple Regime STR model encapsulates a STR model in a higher regime STR model. For our three regime alternative, we have

$$y_t = Z_t'\delta_0 + G_1(s_t; c_1, \gamma_1) \cdot Z_t'\theta_1 + G_2(s_t; c_2, \gamma_2) \cdot \{Z_t'\theta_2 + G_1(s_t; c_1, \gamma_1) Z_t'\theta_3\} + X_t'\alpha + \epsilon_t \quad (36.15)$$

Note that both the linear and nonlinear terms in the first node appear in the second node, and that we have kept the non-varying regressors outside of the second nonlinear node.

van Dijk and Franses (1999) develop an LM test in this framework for testing the null of the two regime LSTAR model against the MRSTAR alternative.

To perform a remaining nonlinearity test against the encapsulated three regime alternative, select **View/Stability Diagnostics/Remaining Nonlinearity Test/Encapsulated Nonlinearity Test**:

Date: 05/06/17 Time: 10:57
Sample: 1861 1988
Included observations: 120
Encapsulated nonlinearity tests using YS(-3) as the threshold variable
Taylor series alternatives: b0 + b1*s [+ b2*s^2 + b3*s^3 + b4 *s^4]

Encapsulated Nonlinearity Tests

Null Hypothesis	F-statistic	d.f.	p-value
H04: b1=b2=b3=b4=0	1.720985	(51, 53)	0.0261
H03: b1=b2=b3=0	1.407513	(39, 65)	0.1102
H02: b1=b2=0	1.573896	(26, 78)	0.0653
H01: b1=0	2.053239	(13, 91)	0.0247

The H0i test uses the i-th order Taylor expansion (bj=0 for all j>i).

Terasvirta Sequential Tests

Null Hypothesis	F-statistic	d.f.	p-value
H3: b3=0	1.049026	(13, 65)	0.4180
H2: b2=0 | b3=0	1.073109	(13, 78)	0.3943
H1: b1=0 | b2=b3=0	2.053239	(13, 91)	0.0247

All tests are based on the third-order Taylor expansion (b4=0).
Original model is not rejected at the 5% level using H03.

Escribano-Jorda Tests

Null Hypothesis	F-statistic	d.f.	p-value
H0L: b2=b4=0	1.707621	(24, 53)	0.0531
H0E: b1=b3=0	1.708217	(23, 53)	0.0550

All tests are based on the fourth-order Taylor expansion.
Original model is rejected at the 5% level using H04.
Recommended model: exponential with nonzero threshold.
. Pr(H0L) < Pr(H0E) with Pr(H0L) >= .05

Parameter Constancy Test

One interesting variant of the STR model is the time-varying coefficient specification which is obtained by choosing time to be the threshold variable. This model allows for structural instability in which regression parameters evolve smoothly over time.

If we take the test methodology outlined in "Linearity Testing" on page 519 and test against the Taylor series expansion using the threshold variable t, we obtain a test for parameter constancy.

To perform this test, select **View/Stability Diagnostics/Parameter Constancy Test**:

Smooth Threshold Parameter Constancy Test
Date: 05/06/17 Time: 10:29
Sample: 1861 1988
Included observations: 120
Encapsulated nonlinearity test using trend as the threshold variable
Taylor series alternatives: b0 + b1*s [+ b2*s^2 + b3*s^3 + b4*s^4]

Parameter Constancy Tests

Null Hypothesis	F-statistic	d.f.	p-value
H04: b1=b2=b3=b4=0	0.962281	(56, 48)	0.5576
H03: b1=b2=b3=0	0.911618	(42, 62)	0.6204
H02: b1=b2=0	1.443235	(28, 76)	0.1063
H01: b1=0	2.164006	(14, 90)	0.0152

The H0i test uses the i-th order Taylor expansion (bj=0 for all j>i).

Note that EViews only performs the first form of the linearity test and not the remaining tests which discriminate between threshold variables. The latter may, however, be obtained by first estimating the STR time-varying coefficient specification and performing the standard linearity test on that equation.

Forecasting

Static and one-step ahead forecasting from an STR estimated equation is straightforward. These simple forecasts all involve conditioning on the observed regressors (including any lagged endogenous regressors) and using the estimated specification to evaluate the forecast.

n-step ahead forecasting of STAR and other nonlinear dynamic models is considerably more difficult. For these dynamic forecasting problems, EViews computes the forecasts by stochastic simulation, with the forecasts and forecast standard errors obtained from the sample average and the standard deviation of the simulated values.

To perform the forecast simulation, click on the **Forecast** button on the equation toolbar or select **Proc/Forecast...** from the equation menu to display the dialog.

If your equation has dynamic components, you will offered a choice between producing a **Dynamic forecast** or a **Static forecast**. If you select **Dynamic forecast**, EViews will display options for choosing the number of simulation **Repetitions**, and for specifying a **Failed reps frac. before halting** the simulation.

Example

We consider here an example of LSTAR estimation using a multiple equilibrium model of unemployment and data described in Martin, Hurn, and Harris (2013, 19.9.1, p. 744). The model is a simple two-regime error correction model of the change in the unemployment rate y_t:

$$\Delta y_t = \eta_0 + \delta_0 y_{t-1} + G(s_t; c, \gamma)(\eta_1 + \delta_1 y_{t-1}) + \epsilon_t \tag{36.16}$$

where $s_t = y_{t-1} - y_{t-13}$ is the lagged annual change in the rate.

The data for unemployment consist of the U.S. Civilian Unemployment Rate (UNRATE) for persons 16 and older from 1948m01 to 2010m03 (2010-10-08 vintage). The “unemp_m-hh19.wf1” workfile containing these data obtained from BLS is available in your EViews example data.

To estimate this model, open the smooth threshold dialog and fill it out as depicted:

We can leave the options at their default values. Click on **OK** to estimate this specification and display the results.

Dependent Variable: D(UNRATE)
Method: Smooth Threshold Regression
Transition function: Logistic
Date: 05/06/17 Time: 12:44
Sample (adjusted): 1949M02 2010M03
Included observations: 734 after adjustments
Threshold variable: UNRATE(-1)-UNRATE(-13)
Starting values: Grid search with zero regression coefficents
Ordinary standard errors & covariance using observed Hessian
Convergence achieved after 14 iterations

Variable	Coefficient	Std. Error	t-Statistic	Prob.
Threshold Variables (linear part)				
C	0.077212	0.040566	1.903344	0.0574
UNRATE(-1)	-0.020922	0.007250	-2.885710	0.0040
Threshold Variables (nonlinear part)				
C	0.281957	0.083136	3.391519	0.0007
UNRATE(-1)	-0.017162	0.012472	-1.376091	0.1692
Slopes				
SLOPE	5.540962	2.644380	2.095373	0.0365
Thresholds				
THRESHOLD	0.420043	0.115051	3.650926	0.0003

R-squared	0.087165	Mean dependent var	0.007357
Adjusted R-squared	0.080896	S.D. dependent var	0.215941
S.E. of regression	0.207023	Akaike info criterion	-0.303835
Sum squared resid	31.20094	Schwarz criterion	-0.266245
Log likelihood	117.5075	Hannan-Quinn criter.	-0.289336
F-statistic	13.90315	Durbin-Watson stat	1.949541
Prob(F-statistic)	0.000000		

Comparison of these results to those reported in Table 19.6 of Martin, Hurn, and Harris shows that while the estimates of the η and δ match, the estimates of the smoothing function slope parameter does not match. This difference stems from the fact that like some authors, Martin, Hurn, and Harris estimate the threshold weighting function employing a scaling factor ω equal to the sample standard deviation of the threshold variable. The resulting parameterization is given by

$$G\left(\frac{s_t}{\omega};\frac{c}{\omega},\gamma\right) \tag{36.17}$$

We can obtain the equivalent result by dividing the estimator of γ and its corresponding standard error by the standard deviation of the threshold, or we can modify the existing specification to account for the weighting.

We can modify our estimates to match the weighted γ results by either by creating a new scaled threshold series, or by directly entering the scaled expression

```
(unrate(-1)-unrate(-13))/@stdev(unrate(-1)-unrate(-13))
```

in the dialog and estimating the resulting specification:

Dependent Variable: D(UNRATE)
Method: Smooth Threshold Regression
Transition function: Logistic
Date: 05/06/17 Time: 12:45
Sample (adjusted): 1949M02 2010M03
Included observations: 734 after adjustments
Threshold variable: (UNRATE(-1)-UNRATE(-13))/@STDEV(UNRATE(-1)-UNRATE(-13),"1948m01 2010m03")

Starting values: Grid search with zero regression coefficents
Ordinary standard errors & covariance using observed Hessian
Convergence achieved after 13 iterations

Variable	Coefficient	Std. Error	t-Statistic	Prob.
Threshold Variables (linear part)				
C	0.077212	0.040567	1.903318	0.0574
UNRATE(-1)	-0.020922	0.007250	-2.885669	0.0040
Threshold Variables (nonlinear part)				
C	0.281957	0.083138	3.391434	0.0007
UNRATE(-1)	-0.017162	0.012472	-1.376053	0.1692
Slopes				
SLOPE	6.886386	3.286494	2.095359	0.0365
Thresholds				
THRESHOLD	0.337978	0.092573	3.650920	0.0003

R-squared	0.087165	Mean dependent var	0.007357
Adjusted R-squared	0.080896	S.D. dependent var	0.215941
S.E. of regression	0.207023	Akaike info criterion	-0.303835
Sum squared resid	31.20094	Schwarz criterion	-0.266245
Log likelihood	117.5075	Hannan-Quinn criter.	-0.289336
F-statistic	13.90315	Durbin-Watson stat	1.949541
Prob(F-statistic)	0.000000		

The slopes now match, but the new estimates of c do not.We can obtain equivalent results for c by multiplying by the standard error by the standard deviation of the threshold.

It is worth commenting that the reason that some authors perform this scaling is to improve the numeric stability of the estimator procedure. We note that EViews always scales internally, but displays the results in the original metric.

We may display the transition function by clicking on **View/Threshold Smoothing Weights...**, and clicking on **OK** to accept the default settings:

Visually, we see that the median transition weight is near 0.3 and that the upper end of the IQR only extends to around 0.75. Moreover, the weight function is rather steep, with most of the variation occurring well within one standard deviation of the lagged annual change in the rate.

Clicking on **View/Residual Diagnostics/Serial Correlation Test...** and accepting the defaults, computes a Breusch-Godfrey serial correlation LM test using a default lag of 2:

Breusch-Godfrey Serial Correlation LM Test:

F-statistic	23.02572	Prob. F(2,726)	0.0000
Obs*R-squared	43.78175	Prob. Chi-Square(2)	0.0000

Test Equation:
Dependent Variable: RESID
Method: Least Squares
Date: 05/06/17 Time: 13:06
Sample: 1949M02 2010M03
Included observations: 734
Coefficient covariance computed using observed Hessian
Presample missing value lagged residuals set to zero.

Variable	Coefficient	Std. Error	t-Statistic	Prob.
C(1)	0.025105	0.040313	0.622743	0.5336
C(2)	-0.002826	0.007134	-0.396101	0.6921
C(3)	-0.008365	0.087390	-0.095723	0.9238
C(4)	-0.001797	0.012675	-0.141772	0.8873
C(5)	1.489169	2.952912	0.504305	0.6142
C(6)	0.024106	0.119306	0.202054	0.8399
RESID(-1)	0.027900	0.036637	0.761520	0.4466
RESID(-2)	0.247504	0.036663	6.750750	0.0000

R-squared	0.059648	Mean dependent var	-1.08E-16
Adjusted R-squared	0.050581	S.D. dependent var	0.206315
S.E. of regression	0.201030	Akaike info criterion	-0.359887
Sum squared resid	29.33986	Schwarz criterion	-0.309767
Log likelihood	140.0784	Hannan-Quinn criter.	-0.340555
F-statistic	6.578778	Durbin-Watson stat	2.082711
Prob(F-statistic)	0.000000		

There is strong evidence against the null of no serial correlation, suggesting that the lag structure of the STAR is insufficient to account for autocorrelation.

We may perform a linearity test by clicking on **View/Stability Diagnostics/Linearity Test**:

Smooth Threshold Linearity Tests
Date: 05/06/17 Time: 13:10
Sample: 1948M01 2010M03
Included observations: 734
Test for nonlinearity using UNRATE(-1)-UNRATE(-13) as the threshold variable
Taylor series alternatives: b0 + b1*s [+ b2*s^2 + b3*s^3 + b4 *s^4]

Linearity Tests

Null Hypothesis	F-statistic	d.f.	p-value
H04: b1=b2=b3=b4=0	12.66027	(8, 724)	0.0000
H03: b1=b2=b3=0	11.43959	(6, 726)	0.0000
H02: b1=b2=0	11.09069	(4, 728)	0.0000
H01: b1=0	21.69341	(2, 730)	0.0000

The H0i test uses the i-th order Taylor expansion (bj=0 for all j>i).

Terasvirta Sequential Tests

Null Hypothesis	F-statistic	d.f.	p-value
H3: b3=0	11.49768	(2, 726)	0.0000
H2: b2=0 | b3=0	0.516693	(2, 728)	0.5967
H1: b1=0 | b2=b3=0	21.69341	(2, 730)	0.0000

All tests are based on the third-order Taylor expansion (b4=0).
Linear model is rejected at the 5% level using H03.
Recommended model: first-order logistic.
. Pr(H3) <= Pr(H2) or Pr(H1) <= Pr(H2)

Escribano-Jorda Tests

Null Hypothesis	F-statistic	d.f.	p-value
H0L: b2=b4=0	8.688837	(4, 724)	0.0000
H0E: b1=b3=0	15.10224	(4, 724)	0.0000

All tests are based on the fourth-order Taylor expansion.
Linear model is rejected at the 5% level using H04.
Recommended model: first-order logistic with nonzero threshold.
. Pr(H0L) >= Pr(H0E) with Pr(H0L) < .05

The tests strongly reject the null of linearity against the smooth transition alternatives. Moreover, both the Terasvirta and the Escribano-Jordan discrimination tests favor the logistic threshold function.

We may test against remaining nonlinearity using either the additive or encapsulated nonlinearity tests. Performing the encapsulated test by selecting **View/Stability Diagnostics/ Remaining Nonlinearity Test/Encapsulated Nonlinearity Test** yields:

Smooth Threshold Remaining Nonlinearity Tests
Date: 05/06/17 Time: 13:13
Sample: 1948M01 2010M03
Included observations: 734
Encapsulated nonlinearity tests using UNRATE(-1)-UNRATE(
-13) as the threshold variable
Taylor series alternatives: b0 + b1*s [+ b2*s^2 + b3*s^3 + b4
*s^4]

Encapsulated Nonlinearity Tests

Null Hypothesis	F-statistic	d.f.	p-value
H04: b1=b2=b3=b4=0	5.262447	(16, 712)	0.0000
H03: b1=b2=b3=0	5.976939	(12, 716)	0.0000
H02: b1=b2=0	3.012346	(8, 720)	0.0025
H01: b1=0	1.661824	(4, 724)	0.1570

The H0i test uses the i-th order Taylor expansion (bj=0 for all
j>i).

Terasvirta Sequential Tests

Null Hypothesis	F-statistic	d.f.	p-value
H3: b3=0	11.55291	(4, 716)	0.0000
H2: b2=0 | b3=0	4.332274	(4, 720)	0.0018
H1: b1=0 | b2=b3=0	1.661824	(4, 724)	0.1570

All tests are based on the third-order Taylor expansion (b4=0).
Original model is rejected at the 5% level using H03.
Recommended model: first-order logistic.
. Pr(H3) <= Pr(H2)

Escribano-Jorda Tests

Null Hypothesis	F-statistic	d.f.	p-value
H0L: b2=b4=0	6.469522	(8, 712)	0.0000
H0E: b1=b3=0	4.657659	(8, 712)	0.0000

All tests are based on the fourth-order Taylor expansion.
Original model is rejected at the 5% level using H04.
Recommended model: exponential with nonzero threshold.
. Pr(H0L) < Pr(H0E) with Pr(H0E) < .05

suggesting that the current model is inadequate to capture the nonlinear dynamics.

References

Eitrheim, Øyvind and Timo Teräsvirta (1996). "Testing the Adequacy of Smooth Transition Autoregressive Models," *Journal of Econometrics*, 74, 59–75.

Escribano, Álvaro and Oscar Jordá (1999). "Improved Testing and Specification of Smooth Transition Regression Models, in *Nonlinear Time Series Analysis of Economic and Financial Data*, Rothman, P., *ed.* ,Klewer: Boston. 289–319.

Leybourne, S., P. Newbold, and D. Vougas (1998). Unit roots and smooth transitions, *Journal of Time Series Analysis,* 19, 83–97.

Lin, Chien-Fu Jeff and Timo Teräsvirta (1994). "Testing the Constancy of Regression Parameters Against Continuous Structural Change," *Journal of Econometrics*, 62, 211–228.

Luukkonen, Ritva, Pentti Saikkonen, and Timo Teräsvirta (1988). Testing Linearity Against Smooth Transition Autoregressive Models," *Biometrika*, 75, 491–499.

Martin, Vance, Stan Hurn, and David Harris (2013). *Econometric Modelling with Time Series.* New York: Cambridge University Press.

Teräsvirta, Timo (1994). "Specification, Estimation, and Evaluation of Smooth Transition Autoregressive Models," *Journal of the American Statistical Association*, 89, 208–218.

van Dijk, Dick, and Philip Hans Franses (1999). "Modeling Multiple Regimes in the Business Cycle," *Macroeconomic Dynamics*, 3, 311–340.

van Dijk, Dick and Timo Teräsvirta, and Philip Hans Franses (2002). "Smooth Transition Autoregressive Models—A Survey of Recent Developments," *Econometric Reviews*, 21, 1–47.

Chapter 37. Elastic Net and Lasso

Elastic net regularization is a popular solution to the overfitting problem, where a model fits training data well but does not generalize easily to new test data. Depending on the particular parameters chosen for the elastic net model, some or all of the regressors are preserved, and their magnitudes are reduced.

Below, we describe EViews tools for estimation of elastic net regression, and the special cases of ridge and Lasso models. You may perform estimation over a single lambda penalization parameter and a grid search over multiple penalization parameters. When multiple parameters are used, EViews also supports options for automatic generation of penalization parameters, as well as cross-validation tools for choosing the parameter with the lowest error.

Following estimation, EViews offers specialized views of tables of the coefficients and other summary statistics, graphs of coefficient evolution with respect to the penalty parameter and other statistics, and diagnostics for cross validation.

The discussion includes brief background for elastic net, Lasso, and ridge regression methods. More detailed discussions may be found in Zou and Hastie (2005) and the textbook by Hastie, Tibshirani, and Friedman (2001).

Background

While the ordinary least squares estimator has many desirable properties, particularly unbiasedness, in certain settings, it can suffer from large variance. For example, if the data have more regressors than the length of the dataset (frequently referred to as the large p, small n problem), or if there are many correlated regressors, the least squares estimates are very sensitive to random errors and may have high variance. We can use *regularization* to reduce this variance by introducing bias, while lowering the total error, with an elastic net regression model.

Elastic net, Lasso, and ridge regression are all penalized regression methods that work by shrinking the magnitudes of the regressors in the model. The usual approach is to modify the standard cost function for linear regression with a penalty term.

$$J = \frac{1}{2m}\sum_{i=1}^{m}\left(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j\right)^2 + \lambda\left[\frac{(1-\alpha)}{2}\sum_{j=1}^{p}\beta_j^2 + \alpha\sum_{j=1}^{p}\left|\beta_j\right|\right] \tag{37.1}$$

Depending on the value of α in the penalty term, Equation (37.1) becomes a ridge regression model, a Lasso model, or an elastic net model. The magnitude of the penalty parameter $\lambda \geq 0$ controls the impact of the penalty. If λ is chosen to be a “large” value, the minimization of this cost function:

$$min_{\beta} J \tag{37.2}$$

will shrink or even set to zero the values of β . A model with smaller (or zero) coefficients is less complex and less prone to overfitting.

Note that the penalization does not include the constant (β_0) term. We only want to reduce the magnitudes of the regressors.

Ridge

The ridge estimator is the ordinary least squares estimator with an L2 penalty term attached:

$$J = \frac{1}{2m}\sum_{i=1}^{m}\left(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j\right)^2 + \lambda\sum_{j=1}^{p}\beta_j^2 \tag{37.3}$$

The penalty shrinks the size of the coefficients but does not reduce them to zero. A bigger λ results in more shrinkage when the cost function J is minimized.

While there are many ways of choosing the penalty parameter, for multiple λ EViews uses cross-validation. In short, this involves partitioning the data into training and test sets, looping over the list of parameters and estimating a set of coefficients with the training data. These coefficients are then used on the test data set to predict the dependent and calculate an error measure. The penalty parameter with the minimum error is λ_{min}. For cross-validation procedures with more than one test set we also calculate the standard error of the error measures across both the training and test sets (giving the largest penalty parameters within one and two standard errors of λ_{min} as λ_{1SE} and λ_{2SE}). Additional information about the cross-validation options available in EViews will be discussed in the next section.

Analytic Solution

Solving the right-hand side of Equation (37.3) for the coefficients in the standard way yields:

$$\beta = (X'X + \lambda I)^{-1}(X'Y) \tag{37.4}$$

Note that when $\lambda = 0$, Equation (37.4) becomes the equation for the OLS coefficients, and when penalization parameter increases ($\lambda \to \infty$), the coefficients are more heavily penalized ($\beta \to 0$). Conveniently, the addition of the positive constant λ to the diagonal makes the matrix $X'X + \lambda I$ nonsingular and invertible.

Bias/Variance Tradeoff

Bias is the difference between the expected value of the estimate and the actual value, and variance is the uncertainty in those estimates. The goal of regularization is to reduce model complexity to the extent that the model fits training data well but also generalizes well to test data. Low bias, more complex models tend to fit training data well, while low variance, less complex models tend to generalize better to test data. In OLS, for example, model complexity is related to the number of regressors. Reducing the number of regressors reduces the

variance of the estimator at the cost of introducing bias. This is called the *bias-variance tradeoff*. For elastic net, ridge, and Lasso models, the equivalent reduction in complexity is achieved by a reduction in the magnitudes of the coefficients, not just in the elimination of coefficients themselves.

We can see this in more detail by calculating the expectation of the ridge coefficient with Equation (37.4):

$$\begin{aligned} E(\hat{\beta}^{ridge}) &= E((X'X+\lambda I)^{-1}(X'Y)) \\ &= (X'X+\lambda I)^{-1}(X'X)\beta \end{aligned} \tag{37.5}$$

Since $E((\hat{\beta}^{ridge}) \neq \beta)$, the ridge estimator is biased. We also see that as $\lambda \to \infty$, $E(\hat{\beta}^{ridge}) \to 0$, as expected.

While we will not derive them here, the bias and variance of the ridge estimator are given by:

$$bias(\hat{\beta}^{ridge}) = -\lambda(X'X+\lambda I)^{-1}\beta \tag{37.6}$$

$$var(\hat{\beta}^{ridge}) = \sigma^2(X'X+\lambda I)^{-1}X'X\{(X'X+\lambda I)^{-1}\}' \tag{37.7}$$

As λ increases, the bias in Equation (37.6) increases, while the variance in Equation (37.7) (σ^2 is error variance from the residuals) decreases.

Lasso

The Lasso (Least Absolute Shrinkage and Selection Operator) estimator is the OLS estimator with an L1 penalty term:

$$J = \frac{1}{2m}\sum_{i=1}^{m}\left(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j\right)^2 + \lambda\sum_{j=1}^{p}|\beta_j| \tag{37.8}$$

Since the penalty term is a sum of absolute values the solution in nonlinear and there is no analytic solution. The Lasso equation must be solved numerically. Unlike in ridge regression, the coefficients can shrink to zero.

It is worth noting that if a group of regressors are correlated, Lasso will tend to favor one out of the group and shrink the others. Ridge regression will proportionally shrink the coefficients of the entire group.

Elastic Net

The elastic net model is a combination of the ridge and Lasso models. Repeating Equation (37.1):

$$J = \frac{1}{2m}\sum_{i=1}^{m}\left(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j\right)^2 + \lambda\left[\frac{(1-\alpha)}{2}\sum_{j=1}^{p}\beta_j^2 + \alpha\sum_{j=1}^{p}|\beta_j|\right]$$

The regularization term is a combination of the L1 and L2 penalties, with the amount of mixing controlled by the parameter α. When $\alpha = 0$, this becomes a ridge regression model. When $\alpha = 1$, it becomes a Lasso model. The compromise between these two models works well for groups of correlated regressors, since the ridge term shrinks them proportionally while the Lasso term pushes them towards zero.

Estimating an Elastic Net Regression in EViews

To estimate an elastic net model, open an existing equation or select **Quick/Estimate Equation...** from the main EViews menu and select **ENET - Elastic Net Regularization** from the main **Method** dropdown menu near the bottom of the dialog.

EViews will display a dialog with two tabs: the **Specification** tab and the **Options** tab.

Specification

Focusing first on the **Specification** page, we see that it has three sections: **Equation specification**, **Penalty specification**, and **Sample** specification. As the **Sample specification** should be familiar, we focus our attention on the first two sections.

Equation Specification

In the **Equation specification** section edit field you should enter the dependent variable followed by the regressors. In order to add weights to the variables, use the keyword `@vw()` around the variable being weighted, followed by the weight. The weight must be between zero and one and multiplies the λ penalty parameter. For example, `@vw(v2, 0)` means the variable V2 is not shrunk.

Penalty Specification

Next, in the **Penalty specification** edit field you should choose from the elastic net, ridge, and Lasso penalties, and enter the desired parameters.

- If you choose the elastic net penalty, EViews will fill in the alpha field with the default value of 0.5, or you may enter a number of your choice. This value must be between zero and one.
- If you choose the ridge penalty, EViews will automatically fill in the alpha field with the ridge parameter $\alpha = 0$. Note that when ridge regression is chosen, EViews displays the analytic solution.
- If you choose the Lasso penalty, EViews will automatically fill in the alpha field with the Lasso parameter $\alpha = 1$.

You are free to fill in the lambda field with a single value, multiple values separated by spaces, or a series object from the workfile. If multiple values of lambda are entered, either directly as a list or through a series object, cross-validation will be performed to determine

the value giving the model with the lowest measurement error. Note that EViews does not allow multiple values of the α parameter.

You may also leave the lambda field blank. If you choose to do this EViews will generate its own list of values based on the data.

Options

The options dialog allows you specify the data transformation options for the regressors and, potentially, the properties of the lambda path, optimization settings, and details of the cross-validation. Depending of the values of lambda chosen in the **Specification** window, various options will be available:

- For single values of lambda, the available options will be limited to **Regressor transformation** in the **Regularization specification** section of the dialog and options in the **Estimation algorithm** section. You may choose between **None** (the default), **Standardization** by sample and population, **L1** and **L2** normalizations, and **Min-max** transformation.
- For multiple *supplied* values of lambda, the available options will expand to include the **Cross-validation options**. Certain options within this section can be chosen depending on the particular method of cross-validation chosen (**Folds** for **K-Fold cross-validation**, for example).

- For multiple *EViews-supplied* values of lambda (i.e., leaving the **Lambda** text box blank on the **Specification** window), the remaining options for **Min/max lambda ratio** and **No. lambdas on path** will become available. In the **Min/max lambda ratio** text box, supply the ratio of the minimum to the maximum value of lambda desired; the default is 0.0001. In the **No. lambdas on path** text box, supply the number of values of lambda on the path; the default is 100.

The **Weights** portion of the dialog is the same as that used in other estimators such as least squares. For more information, please see "Weighted Least Squares".

Cross-validation options

The available cross-validation options include:

- **Shuffle**: After the dataset is divided into training and test sets, the ordering is shuffled. The **Random generator** and **Seed** fields control the details of the shuffling. You may the leave the **Seed** field blank, in which case EViews will use the clock to obtain a seed at the time of estimation, or you may provide an integer from 0 to 2,147,483,647. The **Clear** button may be used to clear the seed used by a previously estimated equation. By changing the value of the **Shuffle reps** field from the default of 1 you may create multiple datasets with different, shuffled ordering of training and test sets.
- **K-Fold**: The dataset is divided into K evenly spaced "folds" and the ordering is shuffled (the details of the shuffling are determined by the **Random generator** and **Seed** fields). One fold is held out as the test set while the remaining K-1 folds are combined into the training set. This process is then repeated, with each fold being held out in turn as the test set. After model estimation the statistics are averaged over all K folds.
- **Leave One Out**: This is the same as K-Fold, but with the number of folds equal to the number of observations.
- **Leave P Out**: This is similar to K-Fold, with the exception that a test set of size P is held out, with the remaining data forming the training set. This process is repeated over all remaining combinations.
- **Rolling Window**: After a window size is chosen for the dataset, the window is divided into training and test sets (the default test set size is 1, and you may also choose the test set size as a fraction), with the test set in each window always coming *after* the training set. The window "rolls" through the dataset until it reaches the end of the dataset. You may also choose how far ahead of the training set you want the test set to be (the horizon), as well as an initial period in the dataset to hold out of the cross-validation (the initial period).
- **Expanding Window**: The dataset is divided into training and test sets (the default test set size is 1, and only integer test set sizes are allowed), with the test set in each

window always coming *after* the training set. With each iteration the training set size expands by one while the test set stays the same size. You may choose how far ahead of the training set you want the test set to be (the horizon), as well as an initial period in the dataset to hold out of the cross-validation (the initial period).

Working with Elastic Net Equations

The elastic net equation estimated by EViews is a penalized regression specification. As a result, most of the views you may be accustomed to for other linear and nonlinear regression equations are not available. Fortunately, we have supplied several additional views applicable to elastic net models and cross-validation. We describe below the basics of working with your elastic net equation, focusing on the views specific to this estimation method.

Estimation Output

Following estimation, EViews displays the elastic net, Lasso, or ridge output. For example, we might have:

Equation: TEST_STDZ Workfile: EXAMPLES::me...

View | Proc | Object | Print | Name | Freeze | Estimate | Forecast | Stats | Resids

Dependent Variable: YSTDZ
Method: Elastic Net Regularization
Date: 03/08/19 Time: 11:14
Sample (adjusted): 2 100
Included observations: 99 after adjustments
Penalty type: Elastic Net (alpha = 0.5)
Lambda at minimum error: 0.04776
Regressor transformation: None
Cross-validation method: K-Fold (number of folds = 5), rng=kn, seed=1004242618
Selection measure: Mean Squared Error

	(minimum)	(+1 SE)	(+2 SE)
Lambda	0.04776	0.1211	0.1458

Variable	Coefficients		
C	-2.48E-17	-2.40E-17	-2.29E-17
VSTDZ1	0.466449	0.434821	0.417471
VSTDZ2	0.000000	1.48E-08	0.000000
VSTDZ3	0.236656	0.207337	0.179262
VSTDZ4	0.000000	0.000000	0.000000
VSTDZ5	-0.264975	-0.239067	-0.217906
VSTDZ6	0.200015	0.175951	0.154894
VSTDZ7	0.011771	3.38E-08	0.000000
VSTDZ8	0.118996	0.094587	0.076866
VSTDZ9	0.000000	0.000000	0.000000
VSTDZ10	0.000000	0.000000	0.000000
VSTDZ11	0.060476	0.036995	0.008426
VSTDZ12	0.000000	0.000000	0.000000
VSTDZ13	0.000000	0.000000	0.000000
VSTDZ14	-0.397264	-0.373173	-0.349798
VSTDZ15	0.000000	1.32E-09	0.000000
VSTDZ16	0.000000	0.000000	0.000000
VSTDZ17	0.000000	0.000000	0.000000
VSTDZ18	0.000000	0.000000	0.000000
VSTDZ19	0.000000	0.000000	0.000000
VSTDZ20	-0.313419	-0.285465	-0.247656
YSTDZ(-1)	0.000000	0.000000	0.000000
d.f.	9	11	8
L1 Norm	2.070021	1.847395	1.652278
R-squared	0.901515	0.881398	0.852507

The top portion of the output displays the dependent variable, method, date, sample information, description of the penalty type and alpha parameter, information about cross-validation if it has been used, the transformation of the regressors, and information about the estimation procedure. In this example, we used the default elastic net penalty with an alpha value of 0.5, midway between a ridge and Lasso model, as well as the default 5-fold cross-validation. Out of the list of lambda values used in the cross-validation procedure (available

in the views, described below) EViews has chosen the value 0.04776 as having the lowest mean squared error. No transformation was done on the regressors.

The middle part of the output displays coefficient values for the certain values of lambda. If cross-validation has been done with multiple training and test sets, the first column supplies the coefficients associated with the minimum measurement error (associated with λ_{min}), and the second and third columns give the coefficients for the largest values of the penalty parameter within one and two standard errors of the minimum error (associated with λ_{1SE} and λ_{2SE}). If one, two, or three values of lambda are provided, the output will simply display the coefficients for these values.

The bottom portion of the output is very different from what you may be used to in EViews. The first row contains the model degrees of freedom, the second is the L1 norm, and the third is the R-squared.

Elastic Net Views and Procs

There is little overlap between the views and procs for elastic net equations and many of the other estimators in EViews such as least squares. We will not discuss the handful of similar views, such as the residuals and derivatives views, but will focus instead on new views specific to elastic net, ridge, and Lasso estimators.

Representations View

As always, the representations view (**View/Representations**) shows the estimation command, estimation equation, and substituted coefficients sections.

Coefficient Matrix

The **Coefficient Matrix** view is one of two views that allow you to easily see the behavior of the model over the entire lambda path. The leftmost column lists all the lambdas used in the estimation. The remaining columns display the coefficients at each value of lambda.

Lambda	C	VSTDZ1	VSTDZ2	VSTDZ3	VSTDZ4
0.797904	0.000000	0.000000	0.000000	0.000000	0.00
0.72702	5.84E-19	0.040773	0.000000	0.000000	0.00
0.662434	-7.92E-21	0.080096	0.000000	0.000000	0.00
0.603585	-5.65E-19	0.117077	0.000000	0.000000	0.00
0.549964	-1.09E-18	0.151774	0.000000	0.000000	0.00
0.501107	-1.58E-18	0.184256	0.000000	0.000000	0.00
0.45659	-2.93E-18	0.212213	0.000000	0.000000	0.00
0.416028	-4.87E-18	0.234645	0.000000	0.000000	0.00
0.379069	-6.58E-18	0.254644	0.000000	0.000000	0.00
0.345394	-8.69E-18	0.274014	0.000000	0.015682	0.00
0.31471	-1.09E-17	0.292274	0.000000	0.036622	0.00
0.286752	-1.29E-17	0.309134	0.000000	0.056034	0.00
0.261278	-1.47E-17	0.324684	0.000000	0.074005	0.00
0.238066	-1.65E-17	0.339010	0.000000	0.090619	0.00
0.216917	-1.77E-17	0.353619	0.000000	0.105376	0.00
0.197647	-1.88E-17	0.367335	0.000000	0.118876	0.00
0.180089	-1.98E-17	0.379958	0.000000	0.131311	0.00
0.16409	-2.07E-17	0.391567	0.000000	0.142756	0.00
0.149513	-2.16E-17	0.402233	0.000000	0.153279	0.00

Summary Path

Select **View/Summary Path** to show the lambda path in the leftmost column, the model degrees of freedom in the second column, the L1 norm of the coefficients in the next column, and the R-squared of the model.

Equation: TEST_STDZ Workfile: EXAMPLES::med\

View | Proc | Object | Print | Name | Freeze | Estimate | Forecast | Stats | Resids

Summary path

	Lambda	d.f.	L1Norm	R-squared	AIC
1	0.806591	1	0.006697	5.43E-10	1.02
2	0.734935	2	0.047916	0.044545	1.00
3	0.669646	2	0.125943	0.122155	0.92
4	0.610156	2	0.199327	0.189725	0.85
5	0.555952	2	0.268179	0.248344	0.79
6	0.506563	2	0.332636	0.299029	0.74
7	0.461561	3	0.395925	0.344709	0.71
8	0.420557	5	0.503878	0.414117	0.69
9	0.383196	5	0.607261	0.475195	0.62
10	0.349154	6	0.721239	0.536331	0.58
11	0.318136	6	0.831017	0.590364	0.53
12	0.289874	6	0.932532	0.636226	0.48
13	0.264122	6	1.026293	0.675085	0.44
14	0.240658	7	1.113063	0.708041	0.43
15	0.219279	7	1.203209	0.739066	0.40
16	0.199799	7	1.286180	0.765216	0.37
17	0.182049	7	1.362483	0.787229	0.35
18	0.165876	7	1.432598	0.805739	0.33
19	0.151140	7	1.496981	0.821286	0.32
20	0.137713	7	1.556062	0.834331	0.30
21	0.125479	7	1.610244	0.845268	0.29
22	0.114332	8	1.660048	0.854450	0.30
23					

Graphs

The **Coefficient Graphs** section of the **View** menu has options for displaying the evolution of the coefficients according to various features of the model.

Lambda Coefficient

To display a graph of the coefficients, ordered by the values of lambda in the lambda path, select **View/Coefficient Graphs/Lambda**:

This is the graph you would see by plotting each column (excluding the constant, if included) of the **View/Coefficient Matrix** table against the leftmost lambda path column. In this example we see that as the value of the regularization penalty lambda increases, the absolute value of each of the coefficients decreases. This is as we would expect for our penalized regression. The thin black vertical line is the minimum value of lambda selected by cross-validation.

L1 Norm Coefficient

To display a graph of the coefficients against the L1 norm of the coefficients (again excluding the constant if included), select **View/Coefficient Graphs/L1 Norm**:

In this graph we can see how the proportion of each coefficient in the norm changes as the norm of all the coefficients changes.

R-squared coefficient

The plot of the coefficients against the R-squared value of the coefficients is under **View/ Coefficient Graphs/R-squared**:

We can see in this example that the R-squared statistic is larger where the absolute values of the coefficients are larger.

Diagnostics

If cross-validation has been done EViews provides diagnostic tools for evaluating the results. These consist of a graph and table of the errors from the training and test sets, graphs of the evolution of the objective function for each value of lambda, and tables of the indices for the training and test sets.

Training/Test Error Graph

To display a graph of the errors produced by cross-validation, choose **View/Diagnostics/ Training/Test Error Graph**:

In this graph we see the lambda path on the x-axis and the error measure means of the training and test sets on the y-axis. As we would expect, the training error is consistently smaller than the test error. This graph is what we see by plotting the first three columns of the **Training/Test Error Table**, below.

Training/Test Error Table

Under **View/Diagnostics/Training/Test Error Table** you can view a table containing the information in the **Training/Test Error Graph** graph.

Equation: TEST_STDZ Workfile: EXAMPLES::med\

View | Proc | Object | Print | Name | Freeze | Estimate | Forecast | Stats | Resids

Means and Standard Errors over cross-validation sets

	Lambda	TestSetMeans	TrainSetMeans	TestSetErrors	TrainSetEr
1	0.806591	1.010328	0.995827	0.104540	0.02
2	0.734935	0.965374	0.943336	0.109512	0.01
3	0.669646	0.907006	0.879406	0.111675	0.01
4	0.610156	0.842383	0.814151	0.108229	0.01
5	0.555952	0.782581	0.750606	0.100323	0.01
6	0.506563	0.730242	0.691666	0.093619	0.01
7	0.461561	0.682416	0.634872	0.088528	0.01
8	0.420557	0.633376	0.578042	0.082281	0.00
9	0.383196	0.580266	0.519901	0.076605	0.00
10	0.349154	0.522557	0.461930	0.074097	0.00
11	0.318136	0.469712	0.410764	0.071052	0.00
12	0.289874	0.421966	0.366276	0.065793	0.00
13	0.264122	0.377340	0.326303	0.057880	0.00
14	0.240658	0.339407	0.292280	0.051268	0.00
15	0.219279	0.305960	0.262369	0.045499	0.00
16	0.199799	0.276322	0.236304	0.040700	0.00
17	0.182049	0.250399	0.213925	0.036345	0.00
18	0.165876	0.228723	0.194933	0.032584	0.00
19	0.151140	0.210831	0.178594	0.029287	0.00
20	0.137713	0.195867	0.164786	0.026499	0.00
21	0.125479	0.183293	0.153208	0.024161	0.00
22	0.114332	0.172735	0.143508	0.022199	0.00
23					

The first column contains the lambda path, the second and third columns list the means over the test and training sets from the cross-validation, and the fourth and fifth columns are the associated standard errors. For example, 5-fold cross-validation performed on a dataset of 100 values divides the data into five sets of twenty data points. For each division of training and test sets, EViews calculates an error measure over the residuals. When repeated over each training/test set, we are left with a series of five error measures that we then use for the mean and standard error.

Train/Test Set Indices

In **View/Diagnostics/Training/Training Sets Table** and **Test Sets Table** you can see tables of the data indices selected by cross-validation for the training and test sets.

Equation: TEST_STDZ Workfile: EXAMPLES::med\

View | Proc | Object | Print | Name | Freeze | Estimate | Forecast | Stats | Resids

Test set indices

Index	Set1	Set2	Set3	Set4	Set5
1	96	53	86	88	6
2	82	91	95	62	3
3	21	15	85	74	5
4	3	19	98	70	3
5	39	5	94	64	7
6	35	49	8	60	2
7	92	29	83	23	4
8	97	32	7	22	7
9	48	45	93	6	7
10	73	79	26	4	5
11	16	2	71	52	9
12	72	27	66	81	5
13	24	18	56	87	7
14	89	47	61	55	1
15	14	65	10	80	6
16	46	1	51	44	3
17	41	99	12	58	2
18	17	38	57	42	1
19	84	37	25	9	6
20	33	43	63	34	

At the top of the **Training Sets Table** are the indices of each cross-validation set (for example, 1 through 5 for 5-fold cross-validation). Down each column are the indices of the data points selected for each cross-validation training set.

Forecasting

Static and dynamic forecasting from an elastic net equation is as straightforward as forecasting from a least squares equation object. To perform the forecast simulation, click on the **Forecast** button on the equation toolbar or select **Proc/Forecast...** from the equation menu to display the dialog.

Example

We consider here an example of Lasso estimation using the prostate cancer data described in multiple sources such as the textbook by Hastie, Tibshirani, and Friedman (2017, 3.4, p. 61). The regressors are log(cancer volume) (LCAVOL), log(prostate weight) (LWEIGHT), age, the logarithm of the amount of benign prostatic hyperplasia (LBPH), seminal vesicle invasion (SVI), log(capsular penetration) (LCP), Gleason score (GLEASON) and percentage Gleason score 4 or 5 (PGG45). The dependent variable is the logarithm of prostate-specific antigen (LPSA). The “prostate.wf1” workfile containing these data obtained from the authors’ website is available in your EViews example data.

To estimate this model, open the elastic net dialog and fill it out as depicted:

In order to match the directions given in the accompanying file "prostate.info.txt," also from the authors' website, we have already standardized the regressors across the length of the dataset. The lambda field is blank to allow EViews to calculate the lambda path and use cross-validation to find the model with the smallest error. We have also specified the sample to include only the training data. On the options page we choose 10-fold cross-validation as directed and leave the other values at their defaults. Click on **OK** to estimate this specification and display the results.

Equation: LASSO Workfile: EXAMPLES::Prostate\

View | Proc | Object | Print | Name | Freeze | Estimate | Forecast | Stats | Resids

Dependent Variable: LPSA
Method: Elastic Net Regularization
Date: 03/15/19 Time: 16:39
Sample: 1 97 IF TRAIN="T"
Included observations: 67
Penalty type: LASSO (alpha = 1)
Lambda at minimum error: 0.0116
Regressor transformation: None
Cross-validation method: K-Fold (number of folds = 10), rng=kn, seed=237641960
Selection measure: Mean Squared Error

	(minimum)	(+1 SE)	(+2 SE)
Lambda	0.0116	0.1891	0.3011

Variable	Coefficients		
LCAVOL_STDZ	0.632114	0.536828	0.487696
AGE_STDZ	-0.123978	0.000000	0.000000
GLEASON_STDZ	0.023213	0.000000	0.000000
LBPH_STDZ	0.123224	0.000000	0.000000
LCP_STDZ	-0.069564	0.000000	0.000000
LWEIGHT_STDZ	0.255763	0.134843	0.046791
PGG45_STDZ	0.096141	0.000000	0.000000
SVI_STDZ	0.282077	0.152288	0.079262
C	2.478387	2.478387	2.478387
d.f.	8	3	3
L1 Norm	4.084461	3.302346	3.092136
R-squared	0.674522	0.587053	0.507137

Comparison with Table 3.3 reported in Hastie, Tibshirani, and Friedman shows that while the coefficients in the +1SE column in the EViews output are very close, they are not an exact match. While there are a few potential sources for this difference, it is most likely due to randomness associated with the cross-validation procedure.

Next we will view the evolution of the coefficients with respect to the lambda penalty:

As expected, as the penalization increases, the complexity of the model decreases and the coefficients gradually shrink to zero. The model at +1SE (lambda = 0.189) chosen by cross-validation is after most of the coefficients have fallen out of the model.

By forecasting over the test data, EViews calculates a root mean squared error for the Lasso model of 0.632.

For comparison, we have also analyzed the same data with an OLS model:

Equation: OLS Workfile: EXAMPLES::Prostate\

View | Proc | Object | Print | Name | Freeze | Estimate | Forecast | Stats | Resids

Dependent Variable: LPSA
Method: Least Squares
Date: 03/12/19 Time: 12:47
Sample: 1 97 IF TRAIN="T"
Included observations: 67

Variable	Coefficient	Std. Error	t-Statistic	Prob.
LCAVOL_STDZ	0.679528	0.126629	5.366290	0.0000
AGE_STDZ	-0.141465	0.101342	-1.395909	0.1681
GLEASON_STDZ	-0.021305	0.145247	-0.146681	0.8839
LBPH_STDZ	0.210147	0.102219	2.055846	0.0443
LCP_STDZ	-0.288493	0.154529	-1.866913	0.0670
LWEIGHT_STDZ	0.263053	0.095628	2.750789	0.0079
PGG45_STDZ	0.266956	0.153614	1.737840	0.0875
SVI_STDZ	0.305201	0.123600	2.469255	0.0165
C	2.464933	0.089315	27.59820	0.0000

R-squared	0.694371	Mean dependent var	2.452345
Adjusted R-squared	0.652215	S.D. dependent var	1.207812
S.E. of regression	0.712286	Akaike info criterion	2.283733
Sum squared resid	29.42638	Schwarz criterion	2.579886
Log likelihood	-67.50505	Hannan-Quinn criter.	2.400921
F-statistic	16.47158	Durbin-Watson stat	1.774180
Prob(F-statistic)	0.000000		

After forecasting over the test data EViews finds a higher root mean squared error for the OLS model of 0.722, compared to 0.632 for the Lasso model. A similar calculation for the ridge model also leads to a lower root mean squared error (0.640) compared to least squares. The regularized regression models have superior forecast performance compared to the OLS model.

References

Friedman, J., Hastie, T., & Tibshirani, R. (2001). *The Elements of Statistical Learning* (Vol. 1, No. 10). New York: Springer Series in Statistics.

Friedman, J., Hastie, T., & Tibshirani, R. (2010). Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33(1), 1.

Zou, H., & Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2), 301-320.

Chapter 38. Functional Coefficient Regression

The following discussion describes EViews tools for estimating and working with functional coefficients regression models. Functional coefficient regression is a semi-parametric approach to extend the standard regression framework to allow for non-linear, dynamic coefficients in regression estimation (Fan and Gijbels (1996); Cai, Fan, and Yao (2000)).

Background

Traditional linear regression postulates that the relationship between a dependent variable Y and explanatory variables X_i is linear:

$$Y_t = \beta_0 + \sum_{i=1}^{k} \beta_i X_{it} + \epsilon_t \tag{38.1}$$

While this framework is typically sufficient for most applications, the requirement that coefficients be the same for all observations is quite restrictive and often violated in practice.

Alternatively, nonparametric modeling is agnostic as to the nature of the relationship between variables, assuming a general functional relationship between the dependent and explanatory variables:

$$Y_t = f(X_{1t}, \ldots, X_{kt}) \tag{38.2}$$

The flexibility of this specification comes at a cost as it can be difficult to interpret nonparametric estimates. For example, describing the marginal effects of a given variable upon Y_t can be challenging.

A flexible, middle ground between these two extremes is the functional coefficients model:

$$Y_t = \beta_0(Z_t) + \sum_{i=1}^{k} \beta_i(Z_t) X_{it} + \epsilon_t \tag{38.3}$$

where the β_j are no longer simple coefficients, but are instead functions of the variable Z_t. Here, the relationship is linear in variables, but non-linear in parameters. In contrast to the linear regression specification Equation (38.1), the coefficients are no longer constant, but instead vary across observations. Non-linear phenomena are easily accommodated in this framework, coefficient relationships are dynamic, and interpretation of coefficient relationships is still intuitive.

Estimation of functional coefficient models is based on local polynomial regression, and incorporates two distinct techniques:

- Approximate the non-linear functions $\beta_j(z)$ using Taylor's Theorem

- Estimate local regressions where we penalize observations using a kernel function

The basic idea is that for each z of interest, we estimate a local regression with kernel weighted squared residuals. Then, estimating this regression for a set of z traces out the functional coefficients relationship.

For example, suppose we have the single regressor functional coefficients model:

$$y_t = \beta_0(Z_t) + \beta_1(Z_t)X_t + \epsilon_t \tag{38.4}$$

We will approximate the coefficient functions $\beta_0(Z_t)$ and $\beta_1(Z_t)$ with a linear Taylor expansion at z. The resulting objective function is:

$$S(z) = \sum_{t=1}^{T} \{ y_t - (\beta_0 + \beta_1 X_t) - (\beta_0^{(1)} + \beta_1^{(1)} X_t))(Z_t - z)\}^2 \cdot K\left(\frac{Z_t - z}{h}\right) \tag{38.5}$$

Minimizing this objective provides estimates of the coefficients at that point,

$$\min_{\beta_0, \beta_1, \beta_0^{(1)}, \beta_1^{(1)}} S(z) \rightarrow \beta_0(z), \beta_1(z), \beta_0^{(1)}(z), \beta_1^{(1)}(z)$$

We may repeat this minimization for various z.

There are several points that we wish to emphasize:

- The functional coefficient estimate is a set of coefficients estimated at a corresponding set of points z.
- The objective function depends on the kernel *bandwidth* h.
- In this example, the objective contains twice as many coefficients as the base model to account for the presence of the first derivatives in the Taylor approximation. We are, however, typically interested only in β_0 and β_1, and not in the corresponding $\beta_0^{(1)}$ and $\beta_1^{(1)}$.

In this example, we employ a linear Taylor expansion, but EViews supports an arbitrary polynomial degree p in Equation (38.5).

Bandwidth Selection

By far the most important step in estimating functional coefficient regressions is optimal selection of the bandwidth parameter $h \in (0, \infty)$. At one extreme, when $h \rightarrow 0$, the functional coefficient estimator reduces to interpolating the data points in y_t (small bias, large variance). Alternatively, when $h \rightarrow \infty$, the functional coefficient estimator reduces to the mean of y_t (large bias, small variance). Between these two extremes we may select a bandwidth that balances bias and variance.

Final Bandwidth

The bandwidth employed in Equation (38.5) may be termed the estimation *final bandwidth*. There are several popular methods used to select an optimal final bandwidth:

- **Minimize Integrated Mean Square Error (IMSE)**: selects for the value which minimizes the integrated (or summed) mean squared error (IMSE) of the functional coefficient estimates, where the IMSE is defined as the average of the squared bias and variance contributions from each functional coefficient estimate at each function evaluation point z.
- **Leave-One-Out (LOO)**: a leave-one-out variant of the IMSE bandwidth optimizer which minimizes the IMSE defined by averaging the squared bias and variance contributions from leave-one-out functional coefficients obtained at each observation evaluation point Z_i.
- **Non-parametric Akaike Information Criterion (AIC)**: computes the optimal bandwidth using a non-parametric AIC with non-parametric degrees of freedom. The idea stems from the Hastie and Tibshirani (1990) degrees of freedom smoothers literature, with the actual bandwidth methodology suggested in Cai, Fan and Yao (2000) and Cai (2003). Briefly, for each evaluation point z, and some bandwidth h, we estimate functional coefficients and use these to compute the standard error of the local polynomial regression residuals. The standard error and an estimated non-parametric degrees of freedom is then used to obtain a functional AIC value. The optimal bandwidth is obtained as the h which minimizes the AIC summed over the z.

In addition to these methods, you may employ any of the pilot methods described below which do not require preliminary functional coefficient estimates.

Pilot Bandwidth

It is important to note that the optimal estimation bandwidth estimators themselves require functional coefficient estimates to obtain standard errors, covariance matrix, and bias estimates. These preliminary estimates require their own *pilot bandwidth* h_{pilot}.

The pilot bandwidth is often determined using one of the following methods which do not depend on other bandwidths:

- **(Simple) Rule of Thumb (ROT)**: This method selects the pilot bandwidth using the asymptotic IMSE (AIMSE) Gaussian kernel as a reference. For a given bandwidth h, we compute the residual standard error of the functional coefficients regression obtained at each evaluation point z and sum these values. We search for the value of h^* which minimizes this sum, and use this value to obtain an optimal h_{pilot} value. For non-Gaussian kernels, the optimal bandwidth employs canonical

kernel transformations as outlined in Marron and Nolan (1988), using the constants in Härdle, *et al.*(1991, p. 76).

- **(Robust) Rule of Thumb (RROT):** A modified version of ROT which computes the objective by summing over the minimum of the residual standard error and $IQR/1.34$, where IQR is the inter-quartile range of the residuals.
- **Residual Squares Criterion (RSC)**: A bandwidth estimator proposed by Fan and Gijbels (1995b). For each evaluation point z and bandwidth h, we compute a functional RSC value that depends on the residual variance, the polynomial degree of the functional coefficients estimation, and a number representing the effective number of local observations. The optional pilot bandwidth h_{pilot} is chosen to be the value that minimizes the sum of these values.
- **Modified Multi-Cross-Validation (MMCV):** The MMCV was proposed by Cai, Fan, and Yao (2000). The cross-validation procedure estimates, for each evaluation point z, a functional coefficient model using Q different sub-series of lengths $T - qm$ and then computes the average mean square error (AMSE) from the m-step ahead forecast errors starting at observation $T - qm + 1$. The resulting Q AMSE values are summed to form a functional objective value at z. The optimal pilot bandwidth h_{pilot} is the value that minimizes the sum of these functional objectives across the z.

Auxiliary Polynomial Degree

The estimation auxiliary regressions using pilot bandwidths also require the specification of a pilot polynomial degree. For reasons outlined in Fan and Gijbels (1995a) and Fan and Gijbels (1996), the pilot polynomial order $p + k$ should exceed the estimation order p by an even integer, $k \in \{2, 4, \ldots\}$.

Forecasting

Forecasting functional coefficient models, and in general, forecasting non-linear models, is considerably more difficult than forecasting linear models.

In particular, consider a series y_t is observed over the period $t = 1, \ldots, T$, and suppose a forecast is required for y_{T+r} for some integer $r \geq 1$.

Point Forecasts

The optimal predictor is given by the familiar conditional expectation of the forecast value conditional on the observable information up to time T. We may evaluate this predictor in one of the following ways:

- **Plug-in Method**: Following Fan and Yao (2003), a r-step ahead forecast can be obtained by substituting for the conditional expectation the corresponding fitted values

$$\hat{Y}_{T+r} = \beta_0(\hat{Z}_{T+r}) + \sum_{i=1}^{k} \beta_i(\hat{Z}_{T+r})\hat{X}_{iT+r} \tag{38.6}$$

For static forecasting, $\hat{Z}_{T+r}$ and $\hat{X}_{iT+r}$ are set to the available actual values.

For dynamic forecasting where the forecast sample evaluation points $\hat{Z}_{T+r}$ or data $\hat{X}_{T+r}$, depend on the lagged endogenous variable, they may be set to the actual lagged endogenous Y_{T+r-v} or lagged forecast values $\hat{Y}_{T+r-v}$ as appropriate.

- **Monte Carlo – Bootstrap**: For each forecast step r, the idea is to simulate a large number, say M of draws of $\hat{Y}_{T+r}$.

  The r-step ahead forecast is obtained as the mean of the M simulated series at that horizon of interest,

$$\hat{Y}_{T+r} = \hat{\beta}_0(\hat{Z}_{T+r}) + \sum_{i=1}^{k} \hat{\beta}_i(\hat{Z}_{T+r})\hat{X}_{iT+r} + \frac{1}{M}\sum_{m=1}^{M} \epsilon^{*m}_{T-r} \tag{38.7}$$

  where the simulated residual ϵ^{*m}_{T-r} is the m-th draw (with replacement) from the within sample estimation residuals (Huang and Shen, 2004; Harvill and Ray, 2005).

  The $\hat{\beta}_0$ and $\hat{\beta}_i$ are functional coefficients estimates computed at the evaluation points $\hat{Z}_{T+r}$ using the original data $\{Y_s, Z_s, X_s\}_{s=1,\ldots T}$.

  As with the Plug-in Method, the $\hat{Z}_{T+r}$ and $\hat{X}_{iT+r}$ are actual values, if available, for static forecasting. For dynamic forecasting, $\hat{Z}_{T+r}$ or $\hat{X}_{T+r}$ will be set to the actual lagged endogenous Y_{T+r-v} or lagged forecast values $\hat{Y}_{T+r-v}$ as appropriate.

- **Monte Carlo – Asymptotic**: Similar to the **Monte Carlo – Bootstrap** approach, but with the simulated residual ϵ^{*m}_{T-r} drawn from a mean zero Gaussian distribution with standard deviation $\hat{\sigma}_\epsilon$ obtained as the standard error of the estimation residuals.

- **Full Bootstrap**: In contrast to the Monte Carlo bootstrap methods which generate simulated values through the residual process, the full bootstrap forecast is the average of forecasts using bootstrap draws of the dependent variable.

  As in the Monte Carlo - Bootstrap, for the m-th draw, we generate a bootstrap draw of the dependent variable Y^{*m}_{T-r} for $t = 1, \ldots T$ using

$$Y^{*m}_t = \hat{\beta}_0(Z_t) + \sum_{i=1}^{k} \hat{\beta}_i(Z_t)X_{it} + \epsilon^{*m}_t \tag{38.8}$$

  where the set of simulated residuals $\{\epsilon^{*m}_s\}_{s=1,\ldots T}$, are a draw (with replacement) from the within sample estimation residuals.

The $\hat{\beta}_0$ and $\hat{\beta}_i$ are functional coefficients estimates computed at the evaluation points Z_t using the original data $\{Y_s, Z_s, X_s\}_{s=1,\ldots T}$.

The data for the m-th simulation are then given by $\{Y^{*m}_s, Z^{*m}_s, X^{*m}_s\}_{s=1,\ldots T}$, where the Z^{*m}_s and X^{*m}_s are the original Z_s and X_s, or are lags of Y^{*m}_s if the originals contain lags of Y_s.

The corresponding r-step ahead individual forecast for the m-th simulation is given by the fitted values:

$$\hat{Y}^{*m}_{T+r} = \hat{\beta}^{*m}_0(\hat{Z}^{*m}_{T+r}) + \sum_{i=1}^{k} \hat{\beta}^{*m}_i(\hat{Z}^{*m}_{T+r})\hat{X}^{*m}_{iT+r} \tag{38.9}$$

The $\hat{\beta}^{*m}_0$ and $\hat{\beta}^{*m}_i$ are functional coefficients estimated using the bootstrap simulation data $\{Y^{*m}_s, Z^{*m}_s, X^{*m}_s\}_{s=1,\ldots T}$ and evaluated at $\hat{Z}^{*m}_{T+r}$.

Once again, in dynamic forecasting settings, $\hat{Z}^{*m}_{T+r}$ or $\hat{X}^{*m}_{T+r}$ that depend on lagged endogenous variables will be set to actual values or lagged forecast values $\hat{Y}^{*m}_{T+r-v}$, as appropriate. For static forecasting, $\hat{Z}^{*m}_{T+r}$ or $\hat{X}^{*m}_{T+r}$ will be set to actuals.

The forecast is obtained by averaging over the individual simulated forecasts:

$$\hat{Y}_{T+r} = \frac{1}{M}\sum_{m=1}^{M} \hat{Y}^{*m}_{T+r} \tag{38.10}$$

Note that the two bootstrap methods are not available in cases where the equation is specified using a dependent variable expression that is not normalizable.

Confidence Intervals

For forecasts that are obtained by Monte Carlo simulation methods, forecast confidence intervals may be obtained using the empirical distribution of the simulation results.

Monte Carlo forecast confidence intervals only consider residual uncertainty since resampling only involves a resampling from the residuals.

Confidence intervals for the full bootstrap are obtained by adapting the results in Davidson and Hinkley (1997) to functional coefficient models. Note that the full bootstrap introduces both coefficient and residual uncertainty into the calculation.

Estimating Functional Coefficients Models in EViews

To estimate a functional coefficient model, **Quick/Estimate Equation...** from the main EViews menu and select **FUNCOEF - Functional Coefficients** from the main **Method** dropdown menu near the bottom of the dialog to display the functional coefficients dialog:

- In the **Specification** edit field you should enter your dependent variable followed by a list of functional coefficient regressors. Here, LOGLYNX is the dependent variable, and the two lag variables LOGLYNX(-1) and LOGLYNX(-2) will have functional coefficients.
- In the **Functional coefficient series** variable edit field, you will enter the Z_t variable upon which the functional coefficients depend. Here the coefficients will vary with LOGLYNX(-2).
- In the **Local curvature** section, the **Polynomial Degree** dropdown allows you to choose the degree of the regression approximation. For example, local linear regression will set this to 1.
- The **Functional evaluation points** options group allows you to control the evaluation points for the estimation. The **Type** dropdown specifies the source of the evaluation points. By default, the values at which we evaluate the functional coefficients are set to the actual data **Series values** Z_t. If you wish to control the evaluation points, you may select **Uniform grid** and provide a minimum, maximum, and length, to create the corresponding evenly spaced grid. Alternatively, if the custom grid is stored as a vector in the workfile, select **Vector values** and enter the name of the vector.

The **Options** tab allows you to specify settings associated with bandwidth selection methods and kernel types.

- From the **Final bandwidth** group **Method** dropdown, you may select the method for choosing the estimation bandwidth. You may choose between: **Rule of thumb**, **Rule of thumb - robust**, **Residual squares criterion**, **Multi Cross-validation** (default), **Mean square error**, **Leave one out**, **Nonparametric AIC**, and **User-specified**. If a **User-specified** bandwidth is requested, you will be prompted to enter a bandwidth.

- Associated with the automated bandwidth methods are bandwidth search options specifying the **Min**, **Max**, **Length**, **Step%**, and **Stop** settings. The first three set the minimum, maximum, and length of the bandwidth grid over which to search for an optimal bandwidth. The last two determine the increment by which to increase the grid value when searching, and how many consecutive failures to improve upon the optimal bandwidth are permitted before the search is terminated. Note here that the search also terminates whenever either the maximum is reached, or the number of bandwidths searched reaches the length specified. Furthermore, when the bandwidth mechanism is **Multi Cross-Validation**, the number of forecast steps is specified in the **m** edit field, and the number of sub-series to evaluate may be specified under **Q** (see "Pilot Bandwidth," on page 559).

- Next, the type of kernel used in estimation may be specified in the **Kernel type** dropdown. You may choose between: **Uniform**, **Triangular**, **Epanechnikov** (default), **Biweight**, **Triweight**, and **Gaussian**. Note that the types are ordered from least to most smooth.

- Lastly, when the estimation bandwidth requires a pilot bandwidth, the latter may be specified **Pilot bandwidth** group **Method** dropdown. The options here follow the same principles outlined for the main bandwidth selection method. Since a pilot bandwidth procedure also requires an auxiliary estimation degree, the latter can be specified from the **Auxiliary local curvature** group by selecting an appropriate value from the **Added polynomial degree** dropdown. Note here that this auxiliary degree is in addition to the main polynomial degree specified on the **Specification** tab. In other words, if the main polynomial degree is 1, and the auxiliary polynomial degree is 2, then the pilot polynomial degree will be equal to 3.

Once you have specified your functional coefficients regression specification and set the options, clicking on **OK** estimates the model and displays results in the form of spool output:

The **Equation Summary** shows the functional coefficients estimation settings. Below these settings are graphs showing the functional coefficients relationships for each of the regressors.

Post-Estimation Views and Procs

After estimating the model, EViews offers several post-estimation views.

Functional Residuals

You may examine the fitted values and residuals from the fitted model by clicking on **View/ Actual, Fitted, Residual** and then choosing to display a graph or table view of the residuals, actuals, and fitted values, or a residual graph:

Functional Bias

To display estimates of the functional bias, select **View/Functional Bias...** from the equation menu. EViews will open a dialog showing you options associated with estimating the bias:

Functional biases must be estimated using a pilot bandwidth ("Pilot Bandwidth," on page 559). The method of selecting a pilot bandwidth for the bias calculation may be specified by selecting desired options. The **Update local pilot bandwidth** checkbox may be used to control whether the results of the bandwidth computation are saved as the local pilot.

Alternately, if a pilot bandwidth has previously been set, it may be used for this calculation by clicking on the **Use local pilot bandwidth** checkbox (see "Setting and Clearing the Local Pilot Bandwidth," on page 577).

Once your settings have been chosen, click on **OK**.

Functional Covariance/Correlation Matrix

Selecting **View/Functional Covariances...** brings up a dialog with options associated with estimating the functional covariance functions:

Functional covariances estimated using a pilot bandwidth ("Pilot Bandwidth," on page 559). A pilot bandwidth for the covariance calculation may be specified using options specified in this dialog. The **Update local pilot bandwidth** checkbox may be used to control whether the results of the bandwidth computation are saved as the local pilot.

Alternately, if a pilot bandwidth has previously been set, it may be used for this calculation by clicking on **Use local pilot bandwidth** (see "Setting and Clearing the Local Pilot Bandwidth," on page 577).

If you wish to compute correlations instead of covariances, select the **Functional correlations** checkbox. Click on **OK** to display the graphs.

Bandwidth Views

Bandwidth selection is a critical part of functional coefficient estimation. EViews offers views for examining your estimation and pilot bandwidth choices.

Estimation Bandwidth

To see how the estimation objective function changes along the bandwidth search grid, select **View/Estimation Bandwidth/Criteria Graph** or **Criteria Table**. Selecting the graph, we have:

shows the graph displaying the relationship between the final bandwidth and the value of the objective.

The table display provides an alternative view of the relationship:

Equation: EQ Workfile: LYNX_FUNCOEF::Ly...

View Proc Object Print Name Freeze Estimate Forecast Stats R

Functional Coefficient: Estimation Final Bandwidth
Bandwidth method: Multi cross-validation (m=10% of sample, Q=4 blocks)
Bandwidth selected: 0.805255
Grid: 0.500 (min), 1.000 (max), 100 (length)
Search: 10.00 (step %), 10 (stop count)
Final polynomial degree: 1
Kernel: Epanechnikov

Bandwidth	Objective
0.500000	0.048412
0.550000	0.048134
0.605000	0.047335
0.665500	0.046711
0.732050	0.046272
0.805255	0.046207 *
0.885781	0.046537
0.974359	0.046830
1.071794	0.047210

* Minimum value of objective.

If the estimation required computation of a pilot bandwidth, the graph or table will also display the relationship between *both* the final and the pilot bandwidths and the objective.

Pilot Bandwidth

To see how the estimation objective function changes along the pilot search grid, select **View/Pilot Bandwidth/Criteria Graph** or **Criteria Table**.

These views will display the pilot bandwidth from estimation, if available, and the local pilot bandwidth, if available (see "Setting and Clearing the Local Pilot Bandwidth," on page 577).

Confidence Intervals

Clicking on **View/Coefficient Diagnostics/Confidence Intervals...** brings up a dialog prompting you to specify options for computing the confidence interval and the pilot bandwidth:

The left-hand side of the dialog contains options associated with computation of the confidence intervals.

- If you prefer to compute standard error bands in place of confidence intervals, click on the **Make SE bands** checkbox and select the number of standard error bands to display in the **Standard deviations** dropbox.
- If instead you are computing confidence intervals, you may specify one or more levels in the **Confidence levels** textbox. The default value is 0.95.
- If critical values are to be computed from Gaussian critical values as opposed to the true limiting distribution, check the **Use Gaussian critical values** box.
- Finally, if bias is to be ignored in deriving confidence intervals, select the **Don't account for bias** setting.

The right portion of the dialog shows options associated with the required pilot stage estimation. You may specify pilot bandwidth options in this dialog or choose to use a previously determined local pilot bandwidth (see "Setting and Clearing the Local Pilot Bandwidth," on page 577).

Click on **OK** to display the output.

Stability and Significance Tests

Selecting **View/Coefficient Diagnostics/Stability and Significance Tests...** brings up a dialog prompting you specify the tests and options of interest:

You may provide one or more restrictions specify a test for:

- whether coefficient j is functionally equivalent to a specific numeric value (example, "c(1) = 2").
- whether coefficient j is constant on the estimated grid (using the **@const** keyword; example, "c(1) = @const").
- whether coefficient j is equal to the values in a series in the workfile (example, "c(1) = s1").
- whether coefficient j is equal to the values in a vector in the workfile (for estimation on a custom grid; example "c(1) = v1").
- whether all coefficients are restricted (using the **@all** keyword; examples, "@all = 2", "@all = @const").

The remainder of the dialog is reserved for options associated with the pilot stage estimation. You may specify pilot options in this dialog or choose to use a previously determined local pilot bandwidth (see "Setting and Clearing the Local Pilot Bandwidth," on page 577).

When all options have been specified, click on **OK** to display the table of results:

Forecasting

To perform a forecast using the estimated functional coefficient model, click on the **Forecast** button in the equation object, or click on **Proc/Forecast...** to display the forecast dialog:

Most of these options should be familiar from typical forecast dialogs. The most important difference lies in the choice of forecast **Stochastic method**. You may use the **Type** dropdown to choose between **Plug-in**, **Monte Carlo (asymptotic)**, **Monte Carlo (bootstrap)**, and **Full bootstrap** methods (see "Point Forecasts," on page 560 for discussion).

The available computation options depend on the type selected:

- For the Monte Carlo and bootstrap methods, the number of simulations may be set in the **Replications** textbox.

- When the stochastic method employs a bootstrap, problematic re-sampling of the functional coefficient variable values can be mitigated by selecting the lower and upper data proportions to conform re-sampled observations to said range of original observations.
- For the full bootstrap, computationally complexity can be achieved by choosing to bootstrap only the dependent variable by ensuring the **Bootstrap response only** box is checked.
- Finally, when any of the stochastic methods is employed (other than **Plug-in**) is chosen, forecast confidence levels will be derived from the distribution of the simulation results. You may specify the confidence interval level in the **Confidence level** edit field. You will also be prompted to enter a name in the **CI prefix (optional)** edit field to save the confidence intervals in the workfile using names formed by appending "_LWR" and "_UPR" to the specified prefix).

Once all the options are specified, click on **OK** to compute the forecasts and display the results.

Setting and Clearing the Local Pilot Bandwidth

Many of the functional coefficient equation views and procedures rely on preliminary computation using a pilot bandwidth. For each of those views and procs, the corresponding dialog offers settings which allow you to specify a method for computing the pilot bandwidth.

A pilot bandwidth may, however, be expensive to compute, and it makes sense to allow users to define a *local pilot bandwidth* that is computed once and then made available for use in all of subsequent views and procedures.

The local pilot bandwidth may be set in two distinct ways:

- By default, when a estimation final stage bandwidth procedure such as MSE, LOO, or non-parametric AIC is employed, the local pilot bandwidth will be initialized to the pilot bandwidth obtained in that procedure.

  (For other estimation methods, no pilot bandwidth is computed so the local pilot bandwidth is not initially defined.)

- You may use the **Set Local Pilot Bandwidth...** proc to initialize (or replace) the local pilot bandwidth.

To set the local pilot bandwidth, click on **Proc/Set Local Pilot Bandwidth...** from the equation menu to display the dialog:

From here, select the desired options and click on **OK** to compute and save a local pilot bandwidth and to display a summary of the bandwidth selection results:

The local pilot bandwidth will be available in all views and procs that employ a pilot bandwidth. You may elect to use this bandwidth in those cases by selecting the **Use existing local pilot bandwidth** checkbox.

You may display a table or a graph of both the estimation pilot bandwidth (if available) and the local pilot bandwidth results by clicking on **View/Pilot Bandwidth** and selecting either **Criteria Graph** or **Criteria Table** ("Bandwidth Views," on page 570).

To clear the local pilot bandwidth, select **Proc/Clear Local Pilot Bandwidth.**

Output Objects to Workfile

Selecting **Proc/Make Functional Objects...** allows you to export various functional coefficients results to the workfile:

The type of functional object to be saved may be chosen from the **Type** dropdown. In most cases, you will be prompted for an **Output name** to be used in saving the object to the workfile.

The output of the functional object is saved to a matrix in the workfile where the rows correspond to the evaluation points specified in estimation. If estimation uses the default evaluation grid, functional values will be computed at the data values of the functional coefficient series for each observation in the estimation sample. Alternately, if you specified a uniform grid or provided a workfile vector, the rows will correspond to the elements of that custom grid.

There are several options associated with the composition of this matrix:

- You should check **Align to sample length** box if you the wish to expand the number of rows of the output matrix are to match the workfile length. Aligning the rows of the output matrix to match the workfile will make it easier to export the results to series.
- Duplicate evaluation points are sometimes present in series or vector-based functional coefficient evaluation. If you wish to only evaluate the functional values at unique points, you should click on the **Remove duplicate grid obs.** checkbox.

- By default, the rows of the matrix will contain evaluation point values as they are observed. For series or vector-based functional coefficient evaluation, the rows will therefore be unordered with respect to these values. If you wish to sort the rows by the evaluation point values, select the **Sort grid obs** checkbox. The rows of the output matrix will be ordered from low to high values of the evaluation points.

Recall that the Taylor expansion in Equation (38.5) creates coefficients associated with the degree of the expansion. By default EViews will save only the original equation coefficients, but you may elect to save results associated these derivatives:

- To include derivative coefficient results in the output, check the **Include derivative coefs.** box.

For each output type, there will be different available options:

- When the output object requires a pilot bandwidth for computation, the **Pilot bandwidth options** will be displayed.
- Covariances and correlations are produced with respect to some reference coefficient which should be specified in the **Reference coef.** dropdown.
- For confidence intervals, the options associated with computation of confidence intervals or standard errors will be displayed ("Confidence Intervals," on page 572). Note that the confidence intervals for coefficients will be saved in pairs to adjacent columns of the output matrix.

The remaining portion of the dialog shows options associated with the required pilot stage estimation. You may specify pilot bandwidth options in this dialog or choose to use a previously determined local pilot bandwidth (see "Setting and Clearing the Local Pilot Bandwidth," on page 577).

References

Cai, Zongwu (2003). "Trending Time-Varying Coefficient Models With Serially Correlated Errors," *SFB 373 Discussion Paper*.

Cai, Zongwu, Fan, Jianqing and Qiwei Yao (2000). "Functional-coefficient regression models for nonlinear time series," *Journal of the American Statistical Association*, 95, 941–956.

Chen, Rong, and Ruey S. Tsay (1993). "Functional-coefficient autoregressive models," *Journal of the American Statistical Association,* 88, 298–308.

Davison, A. C. and D. V. Hinkley (1997). *Bootstrap Methods and Their Applications*, Cambridge: Cambridge University Press.

Fan, Jianqing and Irene Gijbels (1995a). "Adaptive order polynomial fitting: Bandwidth robustification and bias reduction," *Journal of the Computational and Graphical Statistics*, 4, 213–227.

Fan, Jianqing and Irene Gijbels (1995b). "Data-driven bandwidth selection in local polynomial fitting: Variable bandwidth and spatial adaptation," *Journal of the Royal Statistical Society: Series B (Methodological)*, 57, 371–394.

Fan, Jianqing and Irene Gijbels (1996). *Local Polynomial Modelling and its Applications*, London: Chapman & Hall.

Fan, Jianqing and Qiwei Yao (2003). *Nonlinear Time Series*, New York: Springer.

Fan, Jianqing and Wenyang Zhang (2008a). "Simultaneous confidence bands and hypothesis testing in varying-coefficient models," *Scandanavian Journal of Statistics,* 27, 715–731.

Fan, Jianqing and Wenyang Zhang (2008b). "Statistical methods with varying coefficient models," *Statistics and Its Interface,* 1, 179–195.

Härdle, Wolfgang (1991). *Smoothing Techniques with Implementation in S*, New York: Springer Verlag.

Harvill, Jane L. and Bonnie K. Ray (2005). "A note on multi-step forecasting with functional coefficient autoregressive models," *International Journal of Forecasting,* 21, 721–727.

Hastie, Trevor J. and Robert J. Tibshirani (1990). *Generalized Additive Models*. London: Chapman & Hall.

Huang, Jianhua Z. and Haipeng Shen (2004). "Functional coefficient regression models for non-linear time series: A polynomial spline approach," *Scandinavian Journal of Statistics,* 31, 515–534.

Huang, Jianhua Z. and Haipeng Shen (2004). "Functional coefficient regression models for non-linear time series: A polynomial spline approach," *Scandinavian Journal of Statistics,* 31, 515–534.

Marron, J. S. and D. Nolan (1989). "Canonical Kernels for Density Estimation," *Statistics and Probability Letters*, 7, 191–195.

Zhang, Wenyang and Sik-Yum Lee (2000). "Variable bandwidth selection in varying-coefficient models," *Journal of Multivariate Analysis*, 74, 116–134.

Chapter 40. 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 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 dropdown menu 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 dropdown menu is an section **Weight**, where you may define observations weights. The data will be transformed prior to estimation using this specification. (See "Weighted Least Squares" on page 47 for a discussion of the settings).

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 633 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 dropdown menu 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** dropdown, the right side of the dialog changes to offer a set of bootstrap options.

You may use the **Method** dropdown menu to choose from one of four bootstrap methods: **Residual**, **XY-pair**, **MCMB**, **MCMB-A**. See "Bootstrapping," beginning on page 637 for a discussion of the various methods. The default method is **XY-pair**.

Just below the dropdown menu 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 smaller than the original data may produce more accurate results, especially for very large sample sizes; Koenker (2005, p. 108) provides a brief summary.

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** dropdown should be self-explanatory. Simply use the dropdown to choose your desired generator. EViews will initialize the dropdown using the default settings for the choice of generator.

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: 08/12/09 Time: 11:46
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

Variable	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	Restr. objective	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 ("Restr. objective"), 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 compu-

tation of the $\Lambda_n(\tau)$ form of the QLR statistics (see "Standard Views and Procedures" on page 625 and "Quasi-Likelihood Ratio Tests" on page 639).

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: 08/12/09 Time: 11:49
Sample: 1 235
Included observations: 235
Bootstrap Standard Errors & Covariance
Bootstrap method: MCMB-A, reps=500, rng=kn, seed=47500547
Sparsity method: Siddiqui using fitted quantiles
Bandwidth method: Hall-Sheather, bw=0.15744
Estimation successfully identifies unique optimal solution

Variable	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	Restr. objective	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 substantive impact on the probability of 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 18).

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: $\hat{\epsilon}_i(\tau) = Y_i - X_i'\hat{\beta}(\tau)$. Standardized residuals are the ratios of the residuals to 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 $\hat{\sigma}(\tau) = n^{-1} V(\tau)$. This latter estimator is used in forming quasi-likelihood ratio (QLR) tests ("Quasi-Likelihood Ratio Tests" on page 639).

- 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** tab 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 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, $\tau = \{0.1, 0.2, \ldots, 0.9\}$). 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.

The **Coefficient specification** radio buttons permit you to choose a subset of the coefficients to display. By default, EViews will produce results for all of the coefficients in your model. You may select **Intercept only** to produce results only for the intercept, or you may select **User-specified coefficients** and enter a list of coefficient names to show results for specific coefficients. Entering, for example, "C(2) C(3)" will produce process results only for the second and third coefficients.

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 for 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 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.

In addition, you may use the **Coefficient specification** section to specify a subset of coefficients to test. Simply click on the **User-specified coefficients** radio and enter a list of coefficient names to perform the tests on a specific set of coefficients. Entering, for example, "C(2) C(3)" will produce test equality only for the second and third coefficients.

The **Output** page allows you to save the results from the supplementary process estimation. As in "Process Coefficients" on page 626, 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 first equation relating food expenditure and household income in the Engel data set are provided below. 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) test 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:

$$\frac{\beta(\tau) + \beta(1-\tau)}{2} = \beta(1/2) \tag{40.1}$$

To perform the test, select **View/Quantile Process/Symmetric Quantiles Test...** and fill out the dialog.

By default, EViews will test for symmetry using the estimated quantile and the quartiles as 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 quantile, 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. Alternately, you may click on **User-specified coefficients** and enter a list of coefficient names (*e.g.* "C(3) C(4)") to perform tests for specific coefficients.

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(tau) + b(1-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

$$F(y) = \text{Prob}(Y \leq y) \tag{40.2}$$

so that for $0 < \tau < 1$, the τ-th quantile of Y may be defined as the smallest y satisfying $F(y) \geq \tau$:

$$Q(\tau) = \inf\{y\colon F(y) \geq \tau\} \tag{40.3}$$

Given a set of n observations on Y, the traditional empirical distribution function is given by:

$$F_n(y) = \sum_k 1(Y_i \leq y) \tag{40.4}$$

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,

$$Q_n(\tau) = \inf\{y\colon F_n(y) \geq \tau\} \tag{40.5}$$

or equivalently, in the form of a simple optimization problem:

$$\begin{aligned} Q_n(\tau) &= \text{argmin}_\xi \left\{ \sum_{i\colon Y_i \geq \xi} \tau \left| Y_i - \xi \right| + \sum_{i\colon Y_i < \xi} (1-\tau) \left| Y_i - \xi \right| \right\} \\ &= \text{argmin}_\xi \left\{ \sum_i \rho_\tau(Y_i - \xi) \right\} \end{aligned} \tag{40.6}$$

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:

$$Q(\tau | X_i, \beta(\tau)) = X_i' \beta(\tau) \tag{40.7}$$

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:

$$\hat{\beta}_n(\tau) = \operatorname{argmin}_{\beta(\tau)}\left\{\sum_i \rho_\tau(Y_i - X_i'\beta(\tau))\right\} \tag{40.8}$$

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 our *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 possibly 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 algorithm 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,

$$\sqrt{n}(\beta(\tau) - \beta(\tau)) \sim N(0, \tau(1-\tau)s(\tau)^2 J^{-1}) \tag{40.9}$$

where:

$$\begin{aligned} J &= \lim_{n\to\infty}\left(\sum_i X_i X_i'/n\right) = \lim_{n\to\infty}(X'X/n) \\ s(\tau) &= F^{-1\prime}(\tau) = 1/f(F^{-1}(\tau)) \end{aligned} \tag{40.10}$$

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 (40.9) 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:

$$\hat{s}(\tau) = [\hat{F}^{-1}(\tau + h_n) - \hat{F}^{-1}(\tau - h_n)]/(2h_n) \tag{40.11}$$

for some bandwidth h_n tending to zero as the sample size $n \to \infty$. $\hat{s}(\tau)$ 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:

$$\hat{s}(\tau) = X^{*\prime}(\hat{\beta}(\tau + h_n) - \hat{\beta}(\tau - h_n))/(2h_n) \tag{40.12}$$

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 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:

$$h_n = n^{-1/5}\left(\frac{4.5(\phi(\Phi^{-1}(\tau)))^4}{[2(\Phi^{-1}(\tau))^2 + 1]^2}\right)^{1/5} \tag{40.13}$$

(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:

$$h_n = n^{-1/3} z_\alpha^{2/3} \left(\frac{1.5(\phi(\Phi^{-1}(\tau)))^2}{2(\Phi^{-1}(\tau))^2 + 1} \right)^{1/3} \tag{40.14}$$

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:

$$h_n = z_\alpha \sqrt{\frac{\tau(1-\tau)}{n}} \tag{40.15}$$

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 the derivative $F^{-1}{}'(\tau)$ 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:

$$\hat{s}(\tau) = 1 / \left[(1/n) \sum_{i=1}^{n} c_n^{-1} K(\hat{u}_i(\tau)/c_n) \right] \tag{40.16}$$

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:

$$c_n = \kappa(\Phi^{-1}(\tau + h_n) - \Phi^{-1}(\tau - h_n)) \tag{40.17}$$

where $\kappa = \min(s, IQR/1.34)$ is the Silverman (1986) robust estimate of scale (where s 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):

$$\sqrt{n}(\hat{\beta}(\tau) - \beta(\tau)) \sim N(0, \tau(1-\tau) H(\tau)^{-1} J H(\tau)^{-1}) \tag{40.18}$$

where J is as defined earlier,

$$J = \lim_{n \to \infty}\left(\sum_i X_i X_i' / n\right) \tag{40.19}$$

and:

$$H(\tau) = \lim_{n \to \infty}\left(\sum_i X_i X_i' f_i(q_i(\tau)) / n\right) \tag{40.20}$$

$f_i(q_i(\tau))$ 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 (40.18) reduces to the simple scalar sparsity form given in Equation (40.9).

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 634). 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:

$$\begin{aligned} \hat{f}_i(q_i(\tau)) &= 2h_n / (\hat{F}_i^{-1}(q_i(\tau + h)) - \hat{F}_i^{-1}(q_i(\tau - h))) \\ &= 2h_n / (X_i'(\hat{\beta}(\tau + h) - \hat{\beta}(\tau - h))) \end{aligned} \tag{40.21}$$

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 Equation (40.21) 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:

$$\hat{f}_i(q_i(\tau)) = \max(0, 2h_n / (X_i'(\hat{\beta}(\tau + h) - \hat{\beta}(\tau - h)) - \delta)) \tag{40.22}$$

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:

$$\hat{H}_n = \sum_i \hat{f}_i(q_i(\tau)) X_i X_i' / n \tag{40.23}$$

The Powell (1984, 1989) kernel approach replaces the Siddiqui difference with a kernel density estimator using the residuals of the original fitted model:

$$\hat{H}_n = (1/n)\sum_i c_n^{-1} K(\hat{u}_i(\tau)/c_n) X_i X_i' \tag{40.24}$$

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 635 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 (**MCMB** 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:

$$\hat{V}(\hat{\beta}) = n\left(\frac{m}{n}\right)\frac{1}{B}\sum_{j=1}^{B}(\hat{\beta}_j(\tau) - \overline{\beta(\tau)})(\hat{\beta}_j(\tau) - \overline{\beta(\tau)})' \tag{40.25}$$

where $\overline{\beta(\tau)}$ 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 (40.25).

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 (40.25), 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 statistic 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

$$Q(\tau|X_i, \beta(\tau)) = \beta_0(\tau) + X_{i1}'\beta_1(\tau) \tag{40.26}$$

We may then define:

$$\begin{aligned} \hat{V}(\tau) &= \min_{\beta(\tau)} \sum_i \rho_\tau(Y_i - \beta_0(\tau) - X_{i1}'\beta_1(\tau)) \\ \tilde{V}(\tau) &= \min_{\beta_0(\tau)} \sum_i \rho_\tau(Y_i - \beta_0(\tau)) \end{aligned} \tag{40.27}$$

the minimized unrestricted and intercept-only objective functions. The Koenker and Machado goodness-of-fit criterion is given by:

$$R^1(\tau) = 1 - \hat{V}(\tau)/\tilde{V}(\tau) \tag{40.28}$$

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:

$$\begin{aligned} L_n(\tau) &= \frac{2(\tilde{V}(\tau) - \hat{V}(\tau))}{\tau(1-\tau)s(\tau)} \\ \Lambda_n(\tau) &= \frac{2\hat{V}(\tau)}{\tau(1-\tau)s(\tau)} \log(\tilde{V}(\tau)/\hat{V}(\tau)) \end{aligned} \tag{40.29}$$

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 633 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 Diagnostics," beginning on page 196.

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:

$$\beta = (\beta(\tau_1)', \beta(\tau_2)', \ldots, \beta(\tau_K)')' \tag{40.30}$$

Then

$$\sqrt{n}(\hat{\beta} - \beta) \sim N(0, \Omega) \tag{40.31}$$

where Ω has blocks of the form:

$$\Omega_{ij} = [\min(\tau_i, \tau_j) - \tau_i \tau_j] H^{-1}(\tau_i) J H^{-1}(\tau_j) \tag{40.32}$$

In the *i.i.d.* setting, Ω simplifies to,

$$\Omega = \Omega_0 \otimes J \tag{40.33}$$

where Ω_0 has representative element:

$$\omega_{ij} = \frac{\min(\tau_i, \tau_j) - \tau_i \tau_j}{f(F^{-1}(\tau_i))(f(F^{-1}(\tau_j)))} \tag{40.34}$$

Estimation of Ω may be performed directly using (40.32), (40.33) and (40.34), 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:

$$H_0\colon \beta_1(\tau_1) = \beta_1(\tau_2) = \ldots = \beta_1(\tau_K) \tag{40.35}$$

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:

$$\frac{\beta(\tau) + \beta(1-\tau)}{2} = \beta(1/2) \tag{40.36}$$

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 $j = 1, \ldots, (K-1)/2$. Then the Newey and Powell test null is the joint hypothesis that:

$$H_0\colon \frac{\beta(\tau_j) + \beta(\tau_{K-j-1})}{2} = \beta(1/2) \tag{40.37}$$

for $j = 1, \ldots, (K-1)/2$.

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 39. Switching Regression

Linear regression is one of the primary tools for econometric and statistical analysis. There is, however, considerable evidence that nonlinear modeling is sometimes appropriate, especially in the analysis of macroeconomic relationships that are subject to regime change.

We describe here EViews tools for *switching regression* models—linear regression models with nonlinearities arising from discrete changes in regime. We consider settings with both independent and Markov switching where the sample separation into regimes is not observed. Dynamics specifications are permitted through the use of lagged dependent variables as explanatory variables and through the presence of auto-correlated errors.

While this chapter focuses only on models where the regimes are unobserved, related models with observed regimes are discussed in the Heckman chapter.

Background

The following discussion describes only the basic features of switching models. Switching models have a long history in economics that is detailed in numerous surveys (Goldfeld and Quandt, 1973, 1976; Maddala, 1986; Hamilton, 1994; Frühwirth-Schnatter, 2006), and we encourage you to explore these resources for additional discussion.

The Basic Model

Suppose that the random variable of interest, y_t follows a process that depends on the value of an unobserved discrete state variable s_t. We assume there are M possible regimes, and we are said to be in state or regime m in period t when $s_t = m$, for $m = 1, \ldots, M$.

The switching model assumes that there is a different regression model associated with each regime. Given regressors X_t and Z_t, the conditional mean of y_t in regime m is assumed to be the linear specification:

$$\mu_t(m) = X_t'\beta_m + Z_t'\gamma \tag{39.1}$$

where β_m and γ are k_X and k_Z vectors of coefficients. Note that the β_m coefficients for X_t are indexed by regime and that the γ coefficients associated with Z_t are regime invariant.

Lastly, we assume that the regression errors are normally distributed with variance that may depend on the regime. Then we have the model:

$$y_t = \mu_t(m) + \sigma(m)\epsilon_t \tag{39.2}$$

when $s_t = m$, where ϵ_t is iid standard normally distributed. Note that the standard deviation σ may be regime dependent, $\sigma(m) = \sigma_m$.

The likelihood contribution for a given observation may be formed by weighting the density function in each of the regimes by the one-step ahead probability of being in that regime:

$$L_t(\beta, \gamma, \sigma, \delta) = \sum_{m=1}^{M} \frac{1}{\sigma_m} \phi\left(\frac{y_t - \mu_t(m)}{\sigma(m)}\right) \cdot P(s_t = m | \Im_{t-1}, \delta) \tag{39.3}$$

$\beta = (\beta_1, \ldots, \beta_M)$, $\sigma = (\sigma_1, \ldots, \sigma_M)$, δ are parameters that determine the regime probabilities, $\phi(\cdot)$ is the standard normal density function, and $\Im_{t-1}$ is the information set in period $t-1$. In the simplest case, the δ represent the regime probabilities themselves.

The full log-likelihood is a normal mixture

$$l(\beta, \gamma, \sigma, \delta) = \sum_{t=1}^{T} \log\left\{ \sum_{m=1}^{M} \frac{1}{\sigma_m} \phi\left(\frac{y_t - \mu_t(m)}{\sigma(m)}\right) \cdot P(s_t = m | \Im_{t-1}, \delta) \right\} \tag{39.4}$$

which may be maximized with respect to $(\beta, \gamma, \sigma, \delta)$.

Simple Switching

To this point, we have treated the regime probabilities $P(s_t = m | \Im_{t-1}, \delta)$ in an abstract fashion. This section considers a simple switching model featuring independent regime probabilities. We begin by focusing on the specification of the regime probabilities, then describe likelihood evaluation and estimation of those probabilities.

It should be emphasized that the following discussion is valid only for specifications with uncorrelated errors. Models with correlated errors are described in "Serial Correlation" on page 589.

Regime Probabilities

In the case where the probabilities are constant values, we could simply treat them as additional parameters in the likelihood in Equation (39.4). More generally, we may allow for varying probabilities by assuming that p_m is a function of vectors of exogenous observables G_{t-1} and coefficients δ parameterized using a multinomial logit specification:

$$P(s_t = m | \Im_{t-1}, \delta) \equiv p_m(G_{t-1}, \delta) = \frac{\exp(G_{t-1}'\delta_m)}{\sum_{j=1}^{M} \exp(G_{t-1}'\delta_j)} \tag{39.5}$$

for $\delta = (\delta_1, \delta_2, \ldots, \delta_M)$ with the identifying normalization $\delta_M = 0$. The special case of constant probabilities is handled by choosing G_{t-1} to be identically equal to 1.

Likelihood Evaluation

We may use Equation (39.4) and Equation (39.5) to obtain a normal mixture log-likelihood function:

$$l(\beta, \gamma, \sigma, \delta) = \sum_{t=1}^{T} \log\left\{ \sum_{m=1}^{M} \frac{1}{\sigma_m} \phi\left(\frac{y_t - \mu_t(m)}{\sigma_m}\right) \cdot p_m(G_{t-1}, \delta) \right\} \tag{39.6}$$

This likelihood may be maximized with respect to the parameters $(\beta, \gamma, \sigma, \delta)$ using iterative methods.

It is worth noting that the likelihood function for this normal mixture model is unbounded for certain parameter values. However, local optima have the usual consistency, asymptotic normality, and efficiency properties. See Maddala (1986) for discussion of this issue as well as a survey of different algorithms and approaches for estimating the parameters.

Given parameter point-estimates, coefficient covariances may be estimated using conventional methods, *e.g.*, inverse negative Hessian, inverse outer-product of the scores, and robust sandwich.

Filtering

The likelihood expression in Equation (39.6) depends on the one-step ahead probabilities of being in a regime: $P(s_t = m | \Im_{t-1})$. Note, however, that observing the value of the dependent variable in a given period provides additional information about which regime is in effect. We may use this contemporaneous information to obtain updated estimates of the regime probabilities

The process by which the probability estimates are updated is commonly termed *filtering*. By Bayes' theorem and the laws of conditional probability, we have the filtering expressions:

$$P(s_t = m | \Im_t) = P(s_t = m | y_t, \Im_{t-1}) = \frac{f(y_t | s_t = m, \Im_{t-1}) \cdot P(s_t = m | \Im_{t-1})}{f(y_t | \Im_{t-1})} \tag{39.7}$$

The expressions on the right-hand side are obtained as a by-product of the densities obtained during likelihood evaluation. Substituting, we have:

$$P(s_t = m | \Im_t) = \frac{\frac{1}{\sigma_m} \phi\left(\frac{y_t - \mu_t(m)}{\sigma(m)}\right) \cdot p_m(G_{t-1}, \delta)}{\sum_{j=1}^{M} \frac{1}{\sigma_j} \phi\left(\frac{y_t - \mu_t(j)}{\sigma(j)}\right) \cdot p_j(G_{t-1}, \delta)} \tag{39.8}$$

Markov Switching

The Markov switching regression model extends the simple exogenous probability framework by specifying a first-order Markov process for the regime probabilities. We begin by describing the regime probability specification, then discuss likelihood computation, filtering, and smoothing.

Regime Probabilities

The first-order Markov assumption requires that the probability of being in a regime depends on the previous state, so that

$$P(s_t = j | s_{t-1} = i) = p_{ij}(t) \tag{39.9}$$

Typically, these probabilities are assumed to be time-invariant so that $p_{ij}(t) = p_{ij}$ for all t, but this restriction is not required.

We may write these probabilities in a transition matrix

$$p(t) = \begin{bmatrix} p_{11}(t) & \dots & p_{1M}(t) \\ . & \dots & . \\ p_{M1}(t) & \dots & p_{MM}(t) \end{bmatrix} \tag{39.10}$$

where the ij-th element represents the probability of transitioning from regime i in period $t-1$ to regime j in period t. (Note that some authors use the transpose of $p(t)$ so that all of their indices are reversed from those used here.)

As in the simple switching model, we may parameterize the probabilities in terms of a multinomial logit. Note that since each row of the transition matrix specifies a full set of conditional probabilities, we define a separate multinomial specification for each row i of the matrix

$$p_{ij}(G_{t-1}, \delta_i) = \frac{\exp(G_{t-1}'\delta_{ij})}{\sum_{s=1}^{M} \exp(G_{t-1}'\delta_{is})} \tag{39.11}$$

for $j = 1, \dots, M$ and $i = 1, \dots, M$ with the normalizations $\delta_{iM} = 0$.

As noted earlier, Markov switching models are generally specified with constant probabilities so that G_{t-1} contains only a constant. Hamilton's (1989) model of GDP is a notable example of a constant transition probability specification. Alternately, Diebold, Lee, and Weinbach (1994), and Filardo (1994) adopt two-state models that employ time-varying logistic parameterized probabilities.

Likelihood Evaluation and Filtering

The Markov property of the transition probabilities implies that the expressions on the right-hand side of Equation (39.4) must be evaluated recursively.

Briefly, each recursion step begins with filtered estimates of the regime probabilities for the previous period. Given filtered probabilities, $P(s_{t-1} = m | \Im_{t-1})$, the recursion may broken down into three steps:

1. We first form the *one-step ahead predictions* of the regime probabilities using basic rules of probability and the Markov transition matrix:

$$\begin{aligned} P(s_t = m|\Im_{t-1}) &= \sum_{j=1}^{M} P(s_t = m|s_{t-1} = j) \cdot P(s_{t-1} = j|\Im_{t-1}) \\ &= \sum_{j=1}^{M} p_{jm}(G_{t-1}, \delta_j) \cdot P(s_{t-1} = j|\Im_{t-1}) \end{aligned} \tag{39.12}$$

2. Next, we use these one-step ahead probabilities to form the one-step ahead joint densities of the data and regimes in period t:

$$f(y_t, s_t = m|\Im_{t-1}) = \frac{1}{\sigma_m}\phi\left(\frac{y_t - \mu_t(m)}{\sigma(m)}\right) \cdot P(s_t = m|\Im_{t-1}) \tag{39.13}$$

3. The likelihood contribution for period t is obtained by summing the joint probabilities across unobserved states to obtain the marginal distribution of the observed data

$$L_t(\beta, \gamma, \sigma, \delta) = f(y_t|\Im_{t-1}) = \sum_{j=1}^{M} f(y_t, s_t = j|\Im_{t-1}) \tag{39.14}$$

4. The final step is to filter the probabilities by using the results in Equation (39.13) to update one-step ahead predictions of the probabilities:

$$P(s_t = m|\Im_t) = \frac{f(y_t, s_t = m|\Im_{t-1})}{\sum_{j=1}^{M} f(y_t, s_t = j|\Im_{t-1})} \tag{39.15}$$

These steps are repeated successively for each period, $t = 1, \ldots, T$. All that we require for implementation are the initial filtered probabilities, $P(s_0 = m|\Im_0)$, or alternately, the initial one-step ahead regime probabilities $P(s_1 = m|\Im_0)$. See "Initial Probabilities" on page 588 for discussion.

The likelihood obtained by summing the terms in Equation (39.14) may be maximized with respect to the parameters $(\beta, \gamma, \sigma, \delta)$ using iterative methods. Coefficient covariances may be estimated using standard approaches.

Smoothing

Estimates of the regime probabilities may be improved by using all of the information in the sample. The *smoothed* estimates for the regime probabilities in period t use the information set in the final period, $\Im_T$, in contrast to the filtered estimates which employ only contemporaneous information, $\Im_t$. Intuitively, using information about future realizations of the dependent variable y_s $(s > t)$ improves our estimates of being in regime m in period t because the Markov transition probabilities link together the likelihood of the observed data in different periods.

Kim (2004) provides an efficient smoothing algorithm that requires only a single backward recursion through the data. Under the Markov assumption, Kim shows that the joint probability is given by

$$\begin{aligned} P(s_t = i, s_{t+1} = j | \Im_T) &= P(s_t = i | s_{t+1} = j, \Im_T) \cdot P(s_{t+1} = j | \Im_T) \\ &= \frac{P(s_t = i, s_{t+1} = j | \Im_t)}{P(s_{t+1} = j | \Im_t)} \cdot P(s_{t+1} = j | \Im_T) \end{aligned} \tag{39.16}$$

The key in moving from the first to the second line of Equation (39.16) is the fact that under appropriate assumptions, if s_{t+1} were known, there is no additional information about s_t in the future data $(y_{t+1}, \ldots, y_T)$.

The smoothed probability in period t is then obtained by marginalizing the joint probability with respect to s_{t+1}:

$$P(s_t = i | \Im_T) = \sum_{j=1}^{M} P(s_t = i, s_{t+1} = j | \Im_T) \tag{39.17}$$

Note that apart from the smoothed probability terms, $P(s_{t+1} = j | \Im_T)$, all of the terms on the right-hand side of Equation (39.16) are obtained as part of the filtering computations. Given the set of filtered probabilities, we initialize the smoother using $P(s_T = j | \Im_T)$, and iterate computation of Equation (39.16) and Equation (39.17) for $t = T-1, \ldots, 1$ to obtain the smoothed values.

Initial Probabilities

The Markov switching filter requires initialization of the filtered regime probabilities in period 0, $P(s_0 = m | \Im_0)$.

There are a few ways to proceed. Most commonly, the initial regime probabilities are set to the ergodic (steady state) values implied by the Markov transition matrix (see, for example Hamilton (1999, p. 192) or Kim and Nelson (1999, p. 70) for discussion and results). The values are thus treated as functions of the parameters that determine the transition matrix.

Alternately, we may use prior knowledge to specify regime probability values, or we can be agnostic and assign equal probabilities to regimes. Lastly, we may treat the initial probabilities as parameters to be estimated.

Note that the initialization to ergodic values using period 0 information is somewhat arbitrary in the case of time-varying transition probabilities.

Dynamic Models

We may extend the basic switching model to allow for dynamics in the form of lagged endogenous variables and serially correlated errors. The two methods require different assumptions about the dynamic response to changes in regime.

Our discussion is very brief. Frühwirth-Schnatter (2006) offers a nice overview of the differences between these two approaches, and provides further discussion and references.

Dynamic Regression

The most straightforward method of adding dynamics to the switching model is to include lagged endogenous variables. For a model with p lagged endogenous regressors, and random state variable s_t taking the value m we have:

$$\begin{aligned} \mu_t(m) &= X_t'\beta_m + Z_t'\gamma + \sum_{r=1}^{p} \varphi_{rm} y_{t-r} \\ y_t &= \mu_t(m) + \sigma(m)\epsilon_t \end{aligned} \tag{39.18}$$

where ϵ_t is again iid standard normally distributed. The coefficients on the lagged endogenous variable are allowed to be regime-varying, but this generality is not required.

In the Markov switching context, this model has been termed the "Markov switching dynamic regression" (MSDR) model (Frühwirth-Schnatter, 2006). In the special case where the lagged endogenous coefficients are regime-invariant, the model may be viewed as a variant of the "Markov switching intercept" (MSI) specification (Krolzig, 1997).

Of central importance is the fact that the mean specification depends only on the contemporaneous state variable s_t so that lagged endogenous regressors may be treated as additional regime specific X_t or invariant Z_t for purposes of likelihood evaluation, filtering, and smoothing. Thus, the discussions in "Simple Switching" on page 584 and "Markov Switching" on page 585 are directly applicable in MSDR settings.

Serial Correlation

An alternative dynamic approach assumes that the errors are serially correlated (Hamilton, 1989). With serial correlation of order p, we have the AR specification

$$\left(1 - \sum_{r=1}^{p} \rho_r(s_t) L^r\right)(y_t - \mu_t(s_t)) = \sigma(s_t)\epsilon_t \tag{39.19}$$

Rearranging terms and applying the lag operator, we have:

$$y_t = \mu_t(s_t) + \sum_{r=1}^{p} \rho_r(s_t)(y_{t-r} - \mu_{t-r}(s_{t-r})) + \sigma(s_t)\epsilon_t \tag{39.20}$$

In the Markov switching literature, this specification has been termed the "Markov switching autoregressive" (MSAR) (Frühwirth-Schnatter, 2006) or the "Markov switching mean" (MSM) model (Krolzig, 1997). The MSAR model is perhaps most commonly referred to as the "Hamilton model" of switching with dynamics.

Note that, in contrast to the MSDR specification, the mean equation in the MSAR model depends on lag states. The presence of the regime-specific lagged mean adjustments on the right-hand side of Equation (39.20) implies that probabilities for a $p+1$ dimensional state

vector representing the current and p previous regimes are required to obtain a representation of the likelihood.

For example, in a two regime model with an AR(1), we have the standard prediction error representation of the likelihood:

$$l(\beta, \gamma, \sigma, \delta, \rho) = \sum_{t=2}^{T} \log\left\{ \sum_{i=1}^{2} \sum_{j=1}^{2} \frac{1}{\sigma(i)} \phi\left(\frac{y_t - \mu_t(i) - \rho_r(i)(y_{t-1} - \mu_{t-1}(j))}{\sigma(i)} \right) \right. \tag{39.21}$$

$$\left. \cdot P(s_t = i, s_{t-1} = j | \Im_{t-1}) \right\}$$

which requires that we consider probabilities for the four potential regime outcomes for the state vector (s_t, s_{t-1}).

More generally, since there is a $p+1$ dimensional state vector and M regimes, the number of potential realizations is $M^* = M^{p+1}$. The description of the basic Markov switching model above ("Markov Switching" on page 585) is no longer valid since it does not handle the filtering and smoothing for the full M^* vector of probabilities.

Markov Switching AR

Hamilton (1989) derived the form of the MSAR specification and outlined an operational filtering procedure for evaluating the likelihood function. Hamilton (1989), Kim (1994), and Kim and Nelson (1999, Chapter 4) all offer excellent descriptions of the construction of this lagged-state filtering procedure.

Briefly, the Hamilton filter extends the analysis in "Markov Switching" on page 585 to handle the larger $p+1$ dimensional state vector. While the mechanics of the procedure are a bit more involved, the concepts follow directly from the simple filter described above ("Likelihood Evaluation and Filtering" on page 586). The filtered probabilities for lagged values of the states, $s_{t-1}, \ldots, s_{t-p}$ conditional on the information set $\Im_{t-1}$ are obtained from the previous iteration of the filter, and the one-step ahead joint probabilities for the state vector are obtained by applying the Markov updates to the filtered probabilities. These joint probabilities are used to evaluate a likelihood contribution and in obtaining updated filtered probabilities.

Hamilton also offers a modified lag-state smoothing algorithm that may be used with the MSAR model, but the approach is computationally unwieldy. Kim (1994) improves significantly on the Hamilton smoother with an efficient smoothing filter that handles the M^* probabilities using a single backward recursion pass through the data. This approach is a straightforward extension of the basic Kim smoother (see "Smoothing" on page 587).

Simple Switching AR

The simple switching results outlined earlier ("Simple Switching" on page 584) do not hold for the simple switching with autocorrelation (SSAR) model. As with the MSAR specification, the presence of lagged states in the specification complicates the dynamics and requires handling a $p+1$ dimensional state variable representing current and lag states.

Conveniently, we may obtain results for the specification by treating it as a restricted Markov switching model with transition probabilities that do not depend on the origin regime:

$$P(s_t = j | s_{t-1} = i) = p_{ij}(t) = p_j(t) \tag{39.22}$$

so that the rows of the transition matrix are the identical

$$p(t) = \begin{bmatrix} p_1(t) & \dots & p_M(t) \\ . & \dots & . \\ p_1(t) & \dots & p_M(t) \end{bmatrix} \tag{39.23}$$

We may then apply the Hamilton filter and Kim smoother to this restricted specification to obtain the one-step ahead, likelihood, filtered, and smoothed values.

Initial Probabilities

In the serial correlation setting, the Markov switching filter requires initialization of the vector of probabilities associated with the M^{p+1} dimensional state vector. We may proceed as in the uncorrelated model by setting M initial probabilities in period $(-p+1)$ using one of the methods described in "Initial Probabilities" on page 588 and recursively applying Markov transition updates to obtain the joint initial probabilities for the M^* dimensional initial probability vector in period 0.

Again note that the initialization to steady state values using the period $(-p+1)$ information is somewhat arbitrary in the case of time-varying transition probabilities.

Estimating Switching Regressions in EViews

To display the switching regression dialog, first open an equation by selecting **Quick/Estimate Equation…** from the main menu and select **SWITCHREG - Switching Regression** in the **Method** dropdown, or enter the command `switchreg` in the command line:

There are two tabs in this dialog. The first tab is used for basic specification of your switching regression. The second tab contains options for modifying selected features of the specification and for controlling computational aspects of the estimation.

Specification

The **Specification** page contains three sections: **Equation specification**, **Switching specification**, and **Estimation settings**. We focus on the first two sections since the last should already be familiar.

Equation Specification

The top portion of the page contains the **Equation specification** section where you should specify the behavior of the model in the different regimes.

- You should enter the name of the dependent variable series (y_t) followed by any regressors with switching coefficients (X_t) in the first edit field.
- Regressors which have non-varying coefficients (Z_t) should be listed in the second edit field.

- You should select the **Regime specific error variances** option if you wish to allow for heteroskedasticity across regimes.

You may specify dynamics by including lags of the dependent variable series as regressors or by including AR terms to allow for serially correlated errors. Bear in mind that the dimension of the state probability vector, M^{p+1}, increases rapidly in p, where p is the highest AR order in the model.

Note that the dependent variable may not be specified by expression, and that the regressor lists may not contain expressions containing the dependent variable or its lags.

Switching Specification

The **Switching specification** section controls the specification of the regime probabilities.

- The **Switching** type drop-down allows you to choose between **Simple** and **Markov** switching. The default setting is to estimate a simple switching model.
- You should specify the number of regimes $M > 1$ in the edit field. By default, EViews assumes that you have two regimes. Bear in mind that switching models with more than a few regimes may be difficult to estimate.
- You may specify additional regressors that determine the unconditional regime probabilities (for simple switching) or the regime transition probability matrix (for Markov switching). By default, EViews sets the list so that there is a single constant term resulting in time-invariant probabilities.

  Important note: the data for the probability regressors that determine the transition or regime probabilities for period t, should be located in period t of the workfile. That is, the G_{t-1} data should be in period t of the workfile, not period $t-1$. You may, of course, employ standard EViews lag expressions to refer to data in the previous period.

Additional options for setting the initial regime probabilities and restricting the elements of the probability vector or transition matrix are located on the **Options** tab and described in "Switching" on page 594.

Options

Clicking on the **Options** tab displays options for modifying features of the switching specification and for controlling various aspects of the computation.

Switching

The **Switching** options section may be used to specify the initial state probabilities and any restrictions to the regime probability vector or transition matrix.

Recall that evaluation of the likelihood in Markov switching and SSAR models requires presample values for the filtered probabilities ("Initial Probabilities" on page 588). The **Initial regime probabilities** dropdown lets you choose the method of initializing these values (**Ergodic solution** (default), **Estimated**, **Uniform**, **User-specified**). If you select **User-specified**, you will be prompted for the name of an M-element vector in the workfile that contains the initial probabilities.

The **Probability restriction vector/Transition restriction matrix** edit field allows you to specify restrictions on the regime probabilities. Markov switching models, in particular, will sometime require restrictions on transition matrix probabilities. For example, we may have $p_{wr} = 0$ if it is impossible to transition directly from state w to state r. Similarly, if state w is an absorbing state, then $p_{ww} = 1$ and $p_{wr} = 0$ for $r \neq w$.

To specify restrictions, you should enter the name of an M-element vector in the workfile (for a SSAR model), or an $M \times M$ matrix in the workfile (for Markov switching) in the edit field. The vector or matrix should contain valid probability values for elements that are restricted and NAs for elements that are to be estimated. For example, in a three regime Markov switching model where state 3 is an absorbing state, you would have

$$\begin{bmatrix} NA & NA & NA \\ NA & NA & NA \\ 0 & 0 & 1 \end{bmatrix} \tag{39.24}$$

You should take care not to specify invalid or inconsistent restrictions. For example, rows of a Markov transition matrix may not be specified so that there is a single unrestricted cell since the adding up condition for the row places a restriction on that cell. Similarly, fixed values should be valid probabilities that do not have generate row sums greater than 1. EViews will detect these types of errors and will refuse to estimate the model.

In "Regime Heteroskedasticity" on page 612, we offer an example that employs restrictions and discuss several practical issues associated with estimating the model.

Coefficient Covariance Options

EViews estimates the parameters of the likelihood using the Broyden, Fletcher, Goldfarb and Shanno (BFGS) method. By default, the coefficient covariance is computed using the d.f. corrected inverse of the negative of the observed Hessian.

You may use the **Covariance method** dropdown to instruct EViews to use the Hessian, the outer product of the gradient (OPG) method, or a Huber-White robust covariance that forms a sandwich using the Hessian and gradients.

The **d.f. Adjustment** checkbox may be used to remove the d.f. correction.

Starting Value and Iteration Options

The **Iteration control** section contains the standard fields for setting the maximum number of iterations and convergence tolerance. By default, EViews will use the estimation settings from your global options.You may select the **Display settings** checkbox to display starting values and other estimation settings in the top section of your estimation output.

You may use the **Starting values** section to control the setting of initial parameter estimates. Switching regression models often have local roots and may be difficult to estimate so EViews offers a range of tools for choosing starting values.

Start Method

The **Start method** dropdown allows you to specify a basic method for choosing starting values (**EViews Supplied**, **.8 x EViews Supplied**, **.5 x EViews Supplied**, **.3 x EViews Supplied**, **Zero**, **User-Supplied**).

The EViews supplied methods employ simple least squares coefficient estimates or the specified fraction of those estimates. AR coefficients are arbitrarily initialized to zero.

The **Zero** and **User-Supplied** methods are self-explanatory, with the latter taken from the default coefficient vector specified in the dialog (typically the coefficient vector C).

Randomized Estimates

The first two edit fields under **Randomized estimates** allow you to choose random starting values based on the those specified in the **Start method** dropdown.

- EViews will randomly choose the number of random starts as specified in **No. of random starts** and for each random start perform the number of iterations as specified in **Iterations for starts**. The coefficients with the highest likelihood value will be chosen as the starting values.

  For non user-supplied starting values, EViews will, by default, generate 25 sets of random starting values and refine each with 10 iterations before choosing the best set as the starting values. By default, there is no randomization of user-supplied values.

In addition to randomizing based on the initial values, you may randomize based on the final coefficient estimates. The **No. of random from final** edit field determines the number of random coefficients to try following estimation.

- The random starting values are chosen by taking the best estimated values to date and adding random normals with scale given by the **Random scale** fraction of the final coefficient standard deviations. The estimates with the highest likelihood become the final estimates.

  By default, EViews does not perform randomization based on the final coefficient estimates.

For both initial and final randomization, the random starting values are chosen by taking the base values and adding random normals with scale given by the **Random scale** fraction of the root of the estimated coefficient variances (or the scale fraction itself if the variances are not available). The random values will be generated using the **Generator** specified in the dropdown and the random **Seed** specified in the edit field. If a random seed is not specified, EViews will obtain one from a single draw from the generator.

Optimization Options

You can use the **Optimization method** dropdown menu to select a different method.

By default, EViews uses BFGS with Marquardt steps to obtain parameter estimates. You may use the **Optimization method** dropdown to choose between **BFGS**, **OPG - BHHH**, or **Newton-Raphson**. The **Step method** dropdown offers the choice of **Marquardt**, **Dogleg**, and **Line search**.

See "Optimization Method" on page 1314 for discussion.

The remainder of the section allows you to specify a convergence criterion and the maximum number of iterations, and to instruct EViews to display starting value and other optimization information in the output.

Estimation Output

Estimating the equation produces the standard estimation output view. Here we see example output from a simple switching regression model, estimated using data in the workfile "GNP_hamilton.WF1":

Dependent Variable: G
Method: Simple Switching Regression (BFGS / Marquardt steps)
Date: 03/10/15 Time: 16:03
Sample (adjusted): 1951Q4 1984Q4
Included observations: 133 after adjustments
Number of states: 2
Standard errors & covariance computed using observed Hessian
Random search: 25 starting values with 10 iterations using 1 standard deviation (rng=kn, seed=216937)
Convergence achieved after 16 iterations

Variable	Coefficient	Std. Error	z-Statistic	Prob.
		Regime 1		
C	-0.769089	0.233452	-3.294420	0.0010
G(-1)	0.493466	0.140850	3.503494	0.0005
		Regime 2		
C	0.951307	0.138822	6.852728	0.0000
G(-1)	0.272289	0.090619	3.004760	0.0027
		Common		
G(-2)	-0.012522	0.081574	-0.153503	0.8780
LOG(SIGMA)	-0.342578	0.117598	-2.913118	0.0036
		Probabilities Parameters		
P1-C	-0.985846	0.440799	-2.236495	0.0253

Mean dependent var	0.719740	S.D. dependent var	1.058739
S.E. of regression	1.020420	Sum squared resid	132.2396
Durbin-Watson stat	2.043722	Log likelihood	-184.7977
Akaike info criterion	2.884176	Schwarz criterion	3.036300
Hannan-Quinn criter.	2.945993		

The top portion of the output describes the type of switching model and basic sample information, along with information about the computation of the coefficient covariance and the method of producing coefficient estimates.

The middle section displays coefficient estimates. Regime specific coefficients are presented in blocks at the top, followed by any common coefficients, and then the logistic coefficients for the regime probabilities. Note that we have specified G(-1) to be a regime specific regressor, G(-2) to be common, and assume a common error variance. In this example of a simple switching model with two regimes and no probability regressors, there is only a single probability regressor.

The bottom section shows the standard descriptive statistics for the equation. Most are self-explanatory. Of note are the residual-based statistics which employ the expected value of the

residuals obtained by taking the sum of the regime specific residuals weighted by the one-step ahead (unfiltered) regime probabilities (Maheu and McCurdy, 2000).

Switching Views

Once you have estimated your VAR, EViews offers a variety of views for examining your results.

Most of these routines are familiar tools for working with a VAR. You may, for example, examine **Residuals** plots for your estimated equation, examine the **Lag Structure** of your VAR, or use the **Residual Tests** submenu to examine properties of your residuals.

Since the presence of switching and multiple regimes creates a few wrinkles, we offer a few comments on select views.

Regime Results

EViews offers specialized tools for examining the regime transition results and predicted regime probabilities.

Transition Results

To display the transition results dialog, select **View/Regime Results/Transition Results...** EViews offers to display different types of output: **Summary**, **Transition probabilities**, and **Expected durations**.

The default **Summary** display shows a table containing both the transition matrix and the expected durations (Kim and Nelson, 1999, p. 71-72) implied by the transition matrix. For example,

Equation: EQ01
Date: 10/16/12 Time: 11:18
Transition summary: Constant simple switching
transition probabilities and expected durations
Sample (adjusted): 1951Q4 1984Q4
Included observations: 133 after adjustments

Constant transition probabilities:
P(i, k) = P(s(t) = k | s(t-1) = i)
(row = i / column = j)

	1	2
1	0.271733	0.728267
2	0.271733	0.728267

Constant expected durations:

	1	2
	1.373124	3.680077

Here, we see results from the simple switching model with constant transition probabilities. Note that since the model assumes simple switching, the probabilities of being in regime 1 and regime 2 (approximately 0.27 and 0.73, respectively) do not depend on the origin state. These probabilities imply that the expected duration in a regime is roughly 1.37 quarters in regime 1 and 3.68 quarters in regime 2.

In models with varying transition probabilities, the transition probability summary will instead show the means and standard deviations of the transition probabilities and the expected durations.

The latter two output type choices may prove useful in models with time-varying transition probabilities:

- Selecting **Transition probabilities** allows you to see the transition matrix in every period. You will be offered a choice of displaying the transition probabilities in **Graph** (default), **Sheet**, or **Table** form.

  The **Graph** display shows a multiple graph showing each transition probability. For purposes of this display simple switching models are treated as restricted Markov switching models. Thus, a two-regime switching model will always show four separate graphs, one for each transition. Note that for constant transition probability models, each graph will be a straight line.

  The **Sheet** display shows the same results in spreadsheet format while the **Table** form displays results for each period in a table form similar to that used in the summary display. For constant probability models, the spreadsheet format will contain identical rows and the table form will show a single transition matrix.

- Lastly, you may display the **Expected durations** associated with the transition matrix in each period. The results may be displayed in **Graph** (default), **Sheet**, or **Table** form. For constant probability models, the table form will only show the single set of expected durations.

Regime Probabilities

To display the estimated regime probabilities select **View/Regime Results/Regime Probabilities...**

EViews offers a choice between the **One-step-ahead**, **Filtered**, and **Smoothed** probabilities. For simple switching models without AR terms, the filtered and smoothed results are the same. In addition, you may display the results in **Multiple graphs**, in a **Single graph**, or in **Sheet** form.

For graphical output you may also select which regimes to plot by entering the corresponding indices in the **Regimes to plot** edit field. While the default is to show the probabilities for all regimes, you may use the edit field to remove redundant probabilities or to focus on a specific regime of interest.

Residuals and Structural Residuals Diagnostics

The **Residuals** and **Structural Residuals** submenus provides various views of the residuals and associated fitted values. The fits and residuals are formed by taking expectations using regime specific values and the one-step ahead regime probabilities. The standardized residuals are computed similarly with the regime specific residuals scaled by the estimate of the regime standard error prior to forming the expectations (Maheu and McCurdy, 2000).

EViews offers select diagnostics in the **Residuals Tests** submenu. Some care should be taken in the use of these views. Maheu and McCurdy note, for example, that the standardized residuals may be used in Ljung-Box tests, but urge caution in interpreting the results since asymptotic results using these residuals are not available.

Regime Specific Lagged Endogenous

Switching VAR Procs

EViews offers several procs for working with switching equations. Most of these procs are self-explanatory, but we offer brief comments about forecasting and saving regime results to the workfile.

Make Regime Results

The **Make Regime Results** submenu offers routines for saving transition matrix results and predicted regime probabilities into series and matrices in the workfile.

Transition Results

To save transition probability or expected duration results in the workfile click on **Proc/ Make Regime Results/Make Transition Results...**

In addition to prompting you to choose between saving the **Transition probabilities** or the **Expected durations**, you must select an output format. By default, EViews will save the results in a group of series in the workfile. The series names will be formed using the base name specified in the edit field, as in “TPROB12”, “TPROB22”, etc. for transitions, and “TPROB1”, “TPROB2”, etc. for expected durations.

You may instead elect to save the results in a matrix. In this case, EViews will prompt you for the name of the matrix and for an observation at which to evaluate the transition matrix or expected duration. By default, the dialog will be filled with the first observation in the estimation sample.

Regime Probabilities

To save the regime probabilities, select **Proc/Make Regime Results/Make Regime Probabilities Group...**

Select the type of probability you wish to compute (**One-step-ahead**, **Filtered**, or

Smoothed), and enter the names of series to hold the results, one for each probability you wish to retain.

Examples

As illustrations of switching regression estimation, we consider three examples: Hamilton's (1989) MSAR(4) specification for post-war U.S. GNP, Kim and Nelson's (1999) example of a time-varying transition probability model of industrial production, and Kim and Nelson's (1999) three state Markov model of regime heteroskedasticity.

Markov Switching AR

Hamilton (1989) specifies a two-state Markov switching model in which the mean growth rate of GNP is subject to regime switching, and where the errors follow a regime-invariant AR(4) process. The data for this example, which consists of the series G containing (100 times) the log difference of quarterly U.S. GNP for 1951q1–1984q4, may be found in the workfile "GNP_hamilton.WF1".

To estimate the Hamilton model, open an switching equation dialog and enter the specification as depicted below:

The equation specification consists of a two-state Markov switching model with a single switching mean regressor C and the four non-switching AR terms. The error variance is assumed to be common across the regimes. The only probability regressor is the constant C since we have time-invariant regime transition probabilities.

With the exception of the convergence tolerance which we set to 1e-8, we leave the rest of the settings at their default values. Click on **OK** to estimate the equation and display the estimation results.

The top portion of the output describes the estimation settings:

```
Dependent Variable: G
Method: Markov Switching Regression (BFGS / Marquardt
steps)
Date: 03/10/15   Time: 16:04
Sample (adjusted): 1952Q2 1984Q4
Included observations: 131 after adjustments
Number of states: 2
Initial probabilities obtained from ergodic solution
Standard errors & covariance computed using observed Hessian
Random search: 25 starting values with 10 iterations using 1 standard
      deviation (rng=kn, seed=216937)
Convergence achieved after 36 iterations
```

(Bear in mind that if you attempt replicate this estimation using the default settings, you may obtain different results due to the different set of random starting values. You may use the random number generator seed settings to obtain the same starting values.)

The middle section displays the coefficients for the regime specific mean and the invariant error distribution coefficients. We see, in the differences in the regime specific means, what Hamilton terms the fast and slow growth rates for the U.S. economy.

Variable	Coefficient	Std. Error	z-Statistic	Prob.
		Regime 1		
C	1.163517	0.077218	15.06795	0.0000
		Regime 2		
C	-0.358813	0.274116	-1.308983	0.1905
		Common		
AR(1)	0.013487	0.124301	0.108504	0.9136
AR(2)	-0.057521	0.142645	-0.403246	0.6868
AR(3)	-0.246983	0.110779	-2.229508	0.0258
AR(4)	-0.212921	0.114528	-1.859115	0.0630
LOG(SIGMA)	-0.262658	0.089928	-2.920758	0.0035

The remaining results show the parameters of the transition matrix and summary statistics for the estimated equation.

Transition Matrix Parameters				
P11-C	2.243457	0.450936	4.975108	0.0000
P21-C	-1.123682	0.540208	-2.080092	0.0375

Mean dependent var	0.719835	S.D. dependent var	1.066382
S.E. of regression	1.005677	Sum squared resid	125.4118
Durbin-Watson stat	1.923927	Log likelihood	-181.2634
Akaike info criterion	2.904785	Schwarz criterion	3.102317
Hannan-Quinn criter.	2.985051		

Inverted AR Roots	.48-.62i	.48+.62i	-.47+.35i	-.47-.35i

Instead of focusing on the transition matrix parameters, we examine the transition matrix probabilities directly by selecting **View/Regime Results/Transition Results...** and clicking on **OK** to display the default summary view:

Equation: MSAR4
Date: 03/10/15 Time: 16:07
Transition summary: Constant Markov transition
probabilities and expected durations
Sample (adjusted): 1951Q2 1984Q4
Included observations: 135 after adjustments

Constant transition probabilities:
P(i, k) = P(s(t) = k | s(t-1) = i)
(row = i / column = j)

	1	2
1	0.904085	0.095915
2	0.245329	0.754671

Constant expected durations:

	1	2
	10.42586	4.076158

Here, we see the transition probability matrix and the expected durations. Note that there is considerable state dependence in the transition probabilities with a relatively higher probability of remaining in the origin regime (0.90 for the high output state, 0.75 for the low output state). The corresponding expected durations in a regime are approximately 10.4 and 4.1 quarters, respectively.

Lastly, we display the filtered and full sample estimates of the probabilities of being in the two regimes. First select **View/Regime Results/Regime Probabilities...** and choose the filtered results. We will display the results only for the first regime. Then repeat the procedure choosing the smoothed results.

After saving the two views as graphs, editing the labels, and applying the RecShade add-in (http://www.eviews.com/Addins/addins.shtml) to label the NBER recessions, we see that the predicted probabilities of being in the low output state coincide nicely with the commonly employed definition of recessions:

Time-Varying Transitions

Kim and Nelson (1999, p. 93) provide example data for estimating an MSAR(4) model with time-varying transition probabilities, as in Filardo (1994). Filardo models the log growth rate of industrial production (DLOGIP) using an MSAR(4) switching mean specification, using (among other variables) the log growth rate of the Composite Index of Eleven Leading Indicators (DLOGIDX) as a business-cycle predictor.

The Kim and Nelson monthly data for 1948m01 to 1991m04 are included the workfile "kimnelson_tvp.WF1". Note that these data correspond to, but differ slightly from the data used in Filardo (1994).

We open the switching regression dialog and fill out the **Specification** tab with the switching mean AR(4) spec and the switching probability specification:

Note that use have used the lag of the leading indicator variable as our probability regressor so that the period t data for the regressor corresponds to the values influencing the transitions for $t-1$ to t.

We leave the remaining settings at their defaults and click on **OK** to estimate the equation. Following estimation, EViews displays the results:

Dependent Variable: DLOGIP
Method: Markov Switching Regression (BFGS / Marquardt steps)
Date: 03/10/15 Time: 16:11
Sample (adjusted): 1948M07 1991M04
Included observations: 514 after adjustments
Number of states: 2
Initial probabilities obtained from ergodic solution
Standard errors & covariance computed using observed Hessian
Random search: 25 starting values with 10 iterations using 1 standard deviation (rng=kn, seed=1081838492)
Convergence achieved after 15 iterations

Variable	Coefficient	Std. Error	z-Statistic	Prob.
		Regime 1		
C	0.517304	0.077974	6.634341	0.0000
		Regime 2		
C	-0.865887	0.154691	-5.597543	0.0000
		Common		
AR(1)	0.189474	0.050923	3.720827	0.0002
AR(2)	0.079344	0.051670	1.535587	0.1246
AR(3)	0.110945	0.052516	2.112572	0.0346
AR(4)	0.122252	0.051106	2.392135	0.0168
LOG(SIGMA)	-0.362469	0.038370	-9.446653	0.0000
		Transition Matrix Parameters		
P11-C	4.359390	0.755435	5.770703	0.0000
P11-DLOGIDX(-1)	1.770205	0.513537	3.447085	0.0006
P21-C	-1.649359	0.450407	-3.661931	0.0003
P21-DLOGIDX(-1)	0.994559	0.571838	1.739232	0.0820

Mean dependent var	0.245376	S.D. dependent var	0.878113
S.E. of regression	0.772775	Sum squared resid	303.3684
Durbin-Watson stat	2.059688	Log likelihood	-586.5718
Akaike info criterion	2.320667	Schwarz criterion	2.411319
Hannan-Quinn criter.	2.356194		

Inverted AR Roots	.76	-.04+.57i	-.04-.57i	-.50

The results are broadly similar to, but differ slightly from those reported by Filardo. The coefficients on the intercept in the mean equation both differ from zero with opposite (statistically significant) signs. As to the transition matrix parameters, we see that increases in the log growth rate are associated with higher probabilities of being in the high production growth regime, lowering the transition probability out of regime 1 and increasing the transition probability from regime 2 into regime 1.

We can examine more directly the transition probabilities. The default transition probability summary (**View/Regime Results/Transition Results...**) shows descriptive statistics for the elements of the transition matrix:

Equation: MSAR4_TVP
Date: 03/10/15 Time: 16:13
Transition summary: Time-varying Markov
transition probabilities and expected durations
Sample (adjusted): 1948M03
1991M04
Included observations: 518 after adjustments

Time-varying transition probabilities:
P(i, k) = P(s(t) = k | s(t-1) = i)
(row = i / column = j)

		1	2
Mean	1	0.951776	0.048224
	2	0.200155	0.799845

		1	2
Std. Dev.	1	0.112571	0.112571
	2	0.145082	0.145082

Time-varying expected durations:

	1	2
Mean	503.5106	10.50145
Std. Dev.	2948.119	16.90769

We can see the variation in the time-varying probabilities by examining graphs of the transition probabilities for each observation. Select **View/Regime Results/Transition Results...** and click on the **Transition probabilities** radio button. The default view for showing the probabilities is a graph so you may simply click on **OK** to show graphs for all four probabilities.

The filtered probabilities of being in the low production regime are presented below, with NBER recession shading.

To create this graph, select **View/Regime Results/Regime Probabilities...**, select the **Filtered** radio button and enter "2" in the **Regimes to plot** edit field to only show the second (low production) regime.

Click on **OK** to display the graph view, then click on the **Freeze** button to save the view as a graph object. Select **Proc/Add-ins/Add USA Recession Shading** in the graph menu (you must first install the RecShade Add-in to use the automatic shading feature—http://www.eviews.com/Addins/addins.shtml).

Regime Heteroskedasticity

Kim and Nelson (1999) offer an example (Section 4.6, p. 86) of a three state Markov switching model of regime heteroskedastic stock returns from 1926m1–1986m12. The data, which consist of monthly CRSP equal-weighted excess returns are in the series EXCESS, provided in the workfile "ew_excs.WF1".

The specification of the model is as depicted in the dialog below:

The excess returns are assumed to have mean zero so we enter only the name of the dependent variable in the topmost edit field. Since we wish to model regime heteroskedasticity, **Regime specific error variances** box is checked. The model assumes Markov switching probabilities with 3 regimes and constant transition probabilities.

Preliminary analysis indicates that this model is particularly difficult to estimate with a number of local roots exhibiting coefficient singularity.

To obtain estimates we instruct EViews to perform extra randomized starting value estimation. Click on the **Options** tab and change the starting value settings so EViews generates 200 sets of random starts (instead of 25) with 50 (instead of 10) iteration refinements. Set the convergence tolerance to 1e-5. You may need to set the random seed as depicted below. Click on **OK** to proceed with the estimation.

EViews estimates the model and displays the standard switching regression output:

Dependent Variable: EXCESS
Method: Markov Switching Regression (BFGS / Marquardt steps)
Date: 03/18/15 Time: 11:25
Sample: 1926M01 1986M12
Included observations: 732
Number of states: 3
Initial probabilities obtained from ergodic solution
Standard errors & covariance computed using observed Hessian
Random search: 200 starting values with 50 iterations using 1 standard deviation (rng=kn, seed=1727802456)
Convergence achieved after 1 iteration

Variable	Coefficient	Std. Error	z-Statistic	Prob.
		Regime 1		
LOG(SIGMA)	-3.352112	0.077154	-43.44684	0.0000
		Regime 2		
LOG(SIGMA)	-1.736500	0.092396	-18.79415	0.0000
		Regime 3		
LOG(SIGMA)	-2.760187	0.059783	-46.16981	0.0000
		Transition Matrix Parameters		
P11-C	3.608554	0.700534	5.151145	0.0000
P12-C	-16.92853	1799.045	-0.009410	0.9925
P21-C	-2.818798	8.158197	-0.345517	0.7297
P22-C	3.028018	0.875604	3.458206	0.0005
P31-C	-3.894605	0.630337	-6.178607	0.0000
P32-C	-4.417053	0.649727	-6.798325	0.0000

Mean dependent var	-2.44E-18	S.D. dependent var	0.080088
S.E. of regression	0.080198	Sum squared resid	4.688728
Durbin-Watson stat	1.616117	Log likelihood	1001.904
Akaike info criterion	-2.712851	Schwarz criterion	-2.656346
Hannan-Quinn criter.	-2.691054		

The results show estimates of the log standard deviations in the low, high, and medium volatility regimes. The implied standard deviations are 0.035, 0.176, and 0.063, respectively.

We may view the estimated transition probability matrix by clicking on **View/Regime Results/Transition Results...** and clicking on **OK** to display the summary:

Equation: HETEROSK1
Date: 03/10/15 Time: 16:49
Transition summary: Constant Markov transition probabilities and expected durations
Sample: 1926M01 1986M12
Included observations: 732

Constant transition probabilities:
P(i, k) = P(s(t) = k | s(t-1) = i)
(row = i / column = j)

	1	2	3
1	0.973624	2.46E-12	0.026376
2	0.002748	0.951202	0.046050
3	0.019712	0.011691	0.968597

Constant expected durations:

	1	2	3
	37.91315	20.49264	31.84411

The transition probabilities point to a possible explanation of the difficulty in estimating the model. The transition probability from the low volatility regime 1 to the high volatility regime 2 is essentially zero.

We may impose a zero restriction on transitions from the high to low state by creating a 3×3 matrix object RESTR containing:

$$\begin{bmatrix} NA & 0 & NA \\ NA & NA & NA \\ NA & NA & NA \end{bmatrix} \tag{39.25}$$

and re-estimating the model using this restriction matrix.

One tricky aspect of imposing restrictions is that the regime identities are not fixed and can change for different starting values. Accordingly, we re-estimate the model using the prior coefficient estimates as user-specified starting values, keeping in mind that the P12-C coefficient no longer exists.

Your example workfile includes the vector object STARTCOEFS which contains the coefficient estimates from the unrestricted equation with the P12-C coefficient removed. Select **Proc/Set Global Coefs** from the STARTCOEFS toolbar to place these values in the C coefficient vector.

Next, go to the original equation and make a copy by selecting **Object/Copy Object**. Click on the **Estimate** button go to the **Options** tab, enter the name of the restriction vector in the **Transition prob restr. matrix** edit field, and change the **Start method** to **User Specified**. Click on **OK** to accept the changes and estimate the equation.

The update results are displayed below:

Dependent Variable: EXCESS
Method: Markov Switching Regression (OPG - BHHH / Marquardt
steps)
Date: 03/18/15 Time: 11:29
Sample: 1926M01 1986M12
Included observations: 732
Number of states: 3
Fixed probability matrix: RESTR
Initial probabilities obtained from ergodic solution
Standard errors & covariance computed using observed Hessian
Random search: 200 starting values with 50 iterations using 1
standard deviation (rng=kn, seed=1727802456)
Convergence achieved after 56 iterations

Variable	Coefficient	Std. Error	z-Statistic	Prob.
		Regime 1		
LOG(SIGMA)	-3.352111	0.077102	-43.47652	0.0000
		Regime 2		
LOG(SIGMA)	-1.736499	0.092333	-18.80698	0.0000
		Regime 3		
LOG(SIGMA)	-2.760187	0.059743	-46.20125	0.0000
		Transition Matrix Parameters		
P11-C	3.608567	0.700107	5.154310	0.0000
P21-C	-2.818959	8.157176	-0.345580	0.7297
P22-C	3.027993	0.875130	3.460047	0.0005
P31-C	-3.894608	0.629907	-6.182827	0.0000
P32-C	-4.417052	0.649280	-6.802999	0.0000

Mean dependent var	-2.44E-18	S.D. dependent var	0.080088
S.E. of regression	0.080198	Sum squared resid	4.688728
Durbin-Watson stat	1.616117	Log likelihood	1001.904
Akaike info criterion	-2.715584	Schwarz criterion	-2.665356
Hannan-Quinn criter.	-2.696208		

Note that the coefficient associated with the restricted probability p_{12} has been removed.

The transition probabilities view shows us the updated probabilities with the restricted value for p_{12}:

Equation: HETEROSK2
Date: 03/10/15 Time: 16:52
Transition summary: Constant Markov transition probabilities and expected durations
Sample: 1926M01 1986M12
Included observations: 732

Constant transition probabilities:
P(i, k) = P(s(t) = k | s(t-1) = i)
(row = i / column = j)

	1	2	3
1	0.973624	0.000000	0.026376
2	0.002748	0.951202	0.046050
3	0.019712	0.011691	0.968597

Constant expected durations:

	1	2	3
	37.91315	20.49264	31.84411

Lastly, we display the smoothed regime probabilities **View/Regime Results/Regime Probabilities...** selecting **Smoothed** and clicking on **OK** to accept the remaining settings. After rearranging, we have:

References

Davidson, James (2004). "Forecasting Markov-switching Dynamic, Conditionally Heteroscedastic Processes," *Statistics & Probability Letters,* 68, 137-147.

Diebold, Francis X., Lee, Joon-Haeng, and Gretchen C. Weinbach (1994). "Regime Switching with Time-Varying Transition Probabilities," in C. Hargreaves (*ed.*), *Nonstationary Time Series Analysis and Cointegration*, Oxford: Oxford University Press, 283–302.

Filardo, Andrew J. (1994). "Business-Cycle Phases and Their Transitional Dynamics," *Journal of Business & Economic Statistics*, 12, 299-308.

Frühwirth-Schnatter, Sylvia (2006). *Finite Mixture and Markov Switching Models,* New York: Springer Science + Business Media LLC.

Goldfeld, Stephen M. and Richard E. Quandt (1973). "A Markov Model for Switching Regressions," *Journal of Econometrics*, 3–16.

Goldfeld, Stephen M. and Richard E. Quandt (1976), *Studies in Nonlinear Estimation*, Cambridge, MA: Ballinger Publishing Company.

Hamilton, James D. (1989). "A New Approach to the Economic Analysis of Nonstationary Time Series and the Business Cycle," *Econometrica*, 57, 357–384.

Hamilton, James D. (1990). "Analysis of Time Series Subject to Changes in Regime," *Journal of Econometrics*, 45, 39–70.

Hamilton, James D. (1994). *Time Series Analysis*, Chapter 22, Princeton: Princeton University Press.

Hamilton, James D. (1996). "Specification Testing in Markov-switching Time-series Models," *Journal of Econometrics*, 70, 127–157.

Hansen, B. E. (1992). "The Likelihood Ratio Test Under Nonstandard Conditions: Testing the Markov Switching Model of GNP," *Journal of Applied Econometrics*, 7, S6–S82.

Kim, Chang-Jin (1994). "Dynamic Linear Models with Markov-Switching," *Journal of Econometrics*, 60, 1–22.

Kim, Chang-Jin and Charles R. Nelson (1999). *State-Space Models With Regime Switching,* Cambridge: The MIT Press.

Krolzig, Hans-Martin (1997). *Markov-Switching Vector Autoregressions: Modelling, Statistical Inference, and Application to Business Cycle Analysis,* Berlin: Springer-Verlag.

Maddala, G. S. (1986). "Disequilibrium, Self-Selection, and Switching Models," *Handbook of Econometrics*, Chapter 28 in Z. Griliches & M. D. Intriligator (*eds.*), *Handbook of Econometrics*, *Volume 3*, Amsterdam: North- Holland.

Maheu, John M., and Thomas H. McCurdy (2000). "Identifying Bull and Bear Markets in Stock Returns," *Journal of Business & Economic Statistics*, 18, 100–112.

Smith, Daniel R. (2008). "Evaluating Specification Tests for Markov-switching Time-series Models," *Journal of Time Series Analysis*, 29, 629–652.

Chapter 41. The Log Likelihood (LogL) Object

EViews contains customized procedures which help solve a wide range 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.

You should note that while useful in a wide range of settings, the Logl object is nevertheless restricted in the types of functions that it can handle. In particular, the Logl requires that all computations be specified using series expressions, and that the log-likelihood objective can be expressed as a series containing log-likelihood contributions for each observation. For more general optimization problems, you should consider the `optimize` command (see Chapter 10. "User-Defined Optimization," beginning on page 243 of *Command and Programming Reference*).

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:

$$\begin{aligned} y_t &= \beta_1 + \beta_2 x_t + \beta_3 z_t + \epsilon_t \\ \epsilon_t &\sim N(0, \sigma^2 z_t^{\alpha}) \end{aligned} \tag{41.1}$$

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:

$$\begin{aligned} l(\beta, \alpha, \sigma) &= -\frac{T}{2}(\log(2\pi) + \log\sigma^2) - \frac{\alpha}{2}\sum_{t=1}^{T}\log(z_t) - \sum_{t=1}^{T}\frac{(y_t - \beta_1 - \beta_2 x_t - \beta_3 z_t)^2}{\sigma^2 z_t^{\alpha}} \\ &= \sum_{t=1}^{T}\left\{\log\phi\left(\frac{y_t - \beta_1 - \beta_2 x_t - \beta_3 z_t}{\sigma z_t^{\alpha/2}}\right) - \frac{1}{2}\log(\sigma^2 z_t^{\alpha})\right\} \end{aligned} \tag{41.2}$$

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:

$$l(\beta, \alpha, \sigma) = \sum_{t=1}^{T} l_t(\beta, \alpha, \sigma) \tag{41.3}$$

where the individual contributions are given by:

$$l_t(\beta, \alpha, \sigma) = \log\phi\left(\frac{y_t - \beta_1 - \beta_2 x_t - \beta_3 z_t}{\sigma z_t^{\alpha/2}}\right) - \frac{1}{2}\log(\sigma^2 z_t^{\alpha}) \tag{41.4}$$

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 parameters 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 the keyword `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 contribution 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 control the method of evaluation 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 (2008), Chapter 23.11.1 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 655.

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:

$$s^{(i+1)} = \max(r\theta^{(i)}, m) \tag{41.5}$$

The two-sided numeric derivative is evaluated as:

$$\frac{f(\theta^{(i)} + s^{(i+1)}) - f(\theta^{(i)} - s^{(i+1)})}{2s^{(i+1)}} \tag{41.6}$$

The one-sided numeric derivative is evaluated as:

$$\frac{f(\theta^{(i)} + s^{(i+1)}) - f(\theta^{(i)})}{s^{(i+1)}} \tag{41.7}$$

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.

You may also use the **Estimation options** section of the dialog to specify the estimation method, step method, maximum number of iterations, convergence tolerance, and whether to display information about settings in the output.

Note that the default estimation method for new logl objects is BFGS.

The **Coefficient covariance** section of the dialog provides options for controlling the coefficient covariance estimate. The default **Covariance method** estimates the coefficient covariance using the inverse of the matrix specified in the **Information matrix** dropdown menu. Alternately, you may compute the sandwich covariance by selecting **Huber – White** in the **Covariance method** menu. By default, EViews uses the outer-product of the gradients (**OPG**) as the information matrix estimator, but if you are performing non-legacy optimization, you may substitute the (negative of the) observed Hessian by selecting **Hessian – Observed**.

See "Setting Estimation Options" on page 1313 for a discussion of these options.

The default settings should, however, 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 logl 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 1315.

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 symmetric matrix object, you may use the `@coefcov` data member.
- **Wald Coefficient Tests…**: performs the Wald coefficient restriction test. See "Wald Test (Coefficient Restrictions)" on page 202, 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, information about the convergence of the estimation procedure, and the method used to compute the coefficient covariance matrix.

LogL: MLOGIT
Method: Maximum Likelihood (BFGS / Marquardt steps)
Date: 03/10/15 Time: 21:47
Sample: 1 1000
Included observations: 1000
Evaluation order: By equation
Convergence achieved after 22 iterations
Coefficient covariance computed using outer product of gradients

	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 D. “Gradients and Derivatives,” on page 1327 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.

Logl: MLOGIT Workfile: MLOGIT::Mlogit\

View | Proc | Object | Print | Name | Freeze | InsertTxt | Estimate | Stats | Spec

Gradients at estimated parameters

	B2(1)	B2(2)	B2(3)	B3(1)	B3(2)
1	0.617633	0.446035	0.449061	-0.332790	-0.240330
2	0.660916	0.310714	0.074241	-0.308290	-0.144935
3	0.665234	0.330282	0.518881	-0.355333	-0.176420
4	-0.420696	-0.360283	-0.076988	-0.284167	-0.243360
5	0.576206	0.516712	0.306708	-0.303744	-0.272382
6	-0.394766	-0.296582	-0.147463	0.694417	0.521706
7	-0.386750	-0.269690	-0.047604	-0.292850	-0.204211
8	0.582613	0.509899	0.362163	-0.311741	-0.272834
9	-0.288308	-0.076096	-0.252996	-0.379763	-0.100234
10	0.553928	0.549244	0.256273	-0.290475	-0.288018
11	-0.301615	-0.086085	-0.056264	0.674758	0.192585
12	-0.259970	-0.029439	-0.251003	-0.396446	-0.044893
13	0.689356	0.244054	0.363363	-0.346518	-0.122679
14	0.735086	0.057760	0.005284	-0.322644	-0.025352
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 the logl specification. 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.

Logl: MLOGIT Workfile: MLOGIT::Mlogit\

View | Proc | Object | Print | Name | Freeze | InsertTxt | Estimate | Stats | Spec

Log-Likelihood derivative testing
LogL: MLOGIT
Evaluated at estimated parameters

Coefficient	User	Coef. Value
B2(1)	GRAD21	-0.521793
B2(2)	GRAD22	0.994358
B2(3)	GRAD23	0.134983
B3(1)	GRAD31	-0.262307
B3(2)	GRAD32	0.176770
B3(3)	GRAD33	0.399166

	Sum Over Observations		Maximum Difference*	
Coefficient	Numeric	User	Absolute	Percent
B2(1)	2.37E-08	2.36E-08	-5.62E-12	1.65E-09
B2(2)	-5.43E-08	-5.43E-08	-2.86E-12	7.38E-08
B2(3)	-2.90E-08	-2.90E-08	1.81E-11	1.43E-06
B3(1)	-7.40E-08	-7.42E-08	-9.14E-12	2.66E-09
B3(2)	-6.11E-08	-6.11E-08	1.30E-11	5.67E-07
B3(3)	2.09E-07	2.09E-07	6.52E-12	2.04E-06

*Note: analytic and numeric derivatives may differ slightly at estimated parameters due to limits on numeric precision

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.

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 `@atan` 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 (2008).

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, X1 and X2 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:

$$\Pr(y_i = j) = \frac{\exp(\beta_{0j} + \beta_{1j}x_{1i} + \beta_{2j}x_{2i})}{\sum_{k=1}^{3} \exp(\beta_{0k} + \beta_{1k}x_{1i} + \beta_{2k}x_{2i})} = P_{ij} \tag{41.8}$$

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, so we normalize the parameters of the first choice category $j = 1$ to be all zero: $\beta_{0,1} = \beta_{1,1} = \beta_{2,1} = 0$ (see, for example, Greene (2008, Section 23.11.1).

The log likelihood function for the multinomial logit can be written as:

$$l = \sum_{i=1}^{N} \sum_{j=1}^{3} d_{ij} \log(P_{ij}) \tag{41.9}$$

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:

$$\frac{\partial l}{\partial \beta_{kj}} = \sum_{i=1}^{N} (d_{ij} - P_{ij}) x_{ki} \tag{41.10}$$

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 logl1 = (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=l) dd2 c x1 x2
b2 = eq2.@coefs
equation eq3.binary(d=l) 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 using the logl to compute full maximum likelihood estimates of an AR(1). This logl example replicates the ML estimator that is built-into the least squares estimator for an equation (Chapter 24. "Time Series Regression," beginning on page 119).

To illustrate, 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:

$$f(y,\theta) = \begin{cases} \dfrac{1}{\sigma\sqrt{2\pi(1-\rho^2)}}\exp\left\{-\dfrac{(y_t - c/(1-\rho^2))^2}{2(\sigma^2/(1-\rho^2))}\right\} & t = 1 \\ \dfrac{1}{\sigma\sqrt{2\pi}}\exp\left\{-\dfrac{(y_t - c - \rho y_{t-1})^2}{2(\sigma^2)}\right\} & t > 0 \end{cases} \tag{41.11}$$

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, 1994, 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 (2008, Chapter 23.11.1) 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, p. 644–646). Note that there are some typos in Judge *et al.* (1985, p. 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.
- **Probit with heteroskedasticit**y (hprobit1.prg): estimates a probit specification with multiplicative heteroskedasticity.
- **Probit with grouped data** (gprobit1.prg): estimates a probit with grouped data (proportions data).
- **Nested logit** (nlogit1.prg): estimates a nested logit model with 2 branches. Tests the IIA assumption by a Wald test. See Greene (2008, Chapter 23.11.4) 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 (2008, Chapter 25.4) 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 (2008, example 25.4).
- **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 (1994, 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 27. "ARCH and GARCH Estimation," on page 263).
- **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 (1994, p. 668–669). Note that this model may more easily be estimated using the standard ARCH estimation tools provided in EViews (Chapter 27. "ARCH and GARCH Estimation," on page 263).
- **Multivariate GARCH** (bv_garch.prg and tv_garch.prg): estimates the bi- or the trivariate 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 791).

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. (2008). *Econometric Analysis*, 6th Edition, Upper Saddle River, NJ: Prentice-Hall.

Hamilton, James D. (1994). *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, 307–333.

Part VII. Advanced Univariate Analysis

The following section describe EViews tools for advanced univariate analysis:

- Chapter 42. "Univariate Time Series Analysis," on page 667 describes advanced tools for univariate time series analysis, including unit root tests in both conventional and panel data settings, variance ratio tests, and the BDS test for independence.

Chapter 42. Univariate Time Series Analysis

In this section, we discuss a several advanced tools for testing properties of univariate time series. Among the topics considered are unit root tests in both conventional and panel data settings, variance ratio tests, the BDS test for independence.

Unit Root Testing

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*:

$$y_t = y_{t-1} + \epsilon_t, \tag{42.1}$$

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:

$$y_t - y_{t-1} = (1 - L)y_t = \epsilon_t. \tag{42.2}$$

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.

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 Augmented Dickey-Fuller (1979) and Phillips-Perron (1988) tests, EViews 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 671, 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 Tests/Standard 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 dropdown menu 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 ("Hayashi_92.WF1"), 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 dropdown menu. Note that the right-hand side of the dialog has changed, and now features a dropdown menu for selecting the spectral estimation method. You may use this dropdown menu 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 678.

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 using the Andrews bandwidth 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 automatic) 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. See Chapter 55. “Cointegration Testing,” on page 1254 for EViews routines to perform testing in this setting.

Basic Unit Root Theory

The following discussion outlines the basics 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, 1994, Chapter 17, and Hayashi, 2000, Chapter 9).

Consider a simple AR(1) process:

$$y_t = \rho y_{t-1} + x_t'\delta + \epsilon_t, \tag{42.3}$$

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 (42.3) after subtracting y_{t-1} from both sides of the equation:

$$\Delta y_t = \alpha y_{t-1} + x_t'\delta + \epsilon_t, \tag{42.4}$$

where $\alpha = \rho - 1$. The null and alternative hypotheses may be written as,

$$\begin{aligned} H_0&: \alpha = 0 \\ H_1&: \alpha < 0 \end{aligned} \tag{42.5}$$

and evaluated using the conventional t-ratio for α:

$$t_\alpha = \hat{\alpha}/(se(\hat{\alpha})) \tag{42.6}$$

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:

$$\Delta y_t = \alpha y_{t-1} + x_t'\delta + \beta_1 \Delta y_{t-1} + \beta_2 \Delta y_{t-2} + \ldots + \beta_p \Delta y_{t-p} + v_t. \tag{42.7}$$

This augmented specification is then used to test (42.5) using the t-ratio (42.6). 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 (1994, 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 678.

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:

$$d(y_t|a) = \begin{cases} y_t & \text{if } t = 1 \\ y_t - ay_{t-1} & \text{if } t > 1 \end{cases} \tag{42.8}$$

Next, consider an OLS regression of the quasi-differenced data $d(y_t|a)$ on the quasi-differenced $d(x_t|a)$:

$$d(y_t|a) = d(x_t|a)'\delta(a) + \eta_t \tag{42.9}$$

where x_t contains either a constant, or a constant and trend, and let $\delta(a)$ 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:

$$\bar{a} = \begin{cases} 1 - 7/T & \text{if } x_t = \{1\} \\ 1 - 13.5/T & \text{if } x_t = \{1, t\} \end{cases} \tag{42.10}$$

We now define the *GLS detrended data,* y_t^d using the estimates associated with the $\bar{a}$:

$$y_t^d \equiv y_t - x_t'\delta(\bar{a}) \tag{42.11}$$

Then the DFGLS test involves estimating the standard ADF test equation, (42.7), after substituting the GLS detrended y_t^d for the original y_t:

$$\Delta y_t^d = \alpha y_{t-1}^d + \beta_1 \Delta y_{t-1}^d + \ldots + \beta_p \Delta y_{t-p}^d + v_t \tag{42.12}$$

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 (42.4), 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:

$$\tilde{t}_\alpha = t_\alpha \left(\frac{\gamma_0}{f_0}\right)^{1/2} - \frac{T(f_0 - \gamma_0)(se(\hat{\alpha}))}{2 f_0^{1/2} s} \tag{42.13}$$

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 (42.4) (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 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 676 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:

$$y_t = x_t'\delta + u_t \tag{42.14}$$

The LM statistic is be defined as:

$$LM = \sum_t S(t)^2 / (T^2 f_0) \tag{42.15}$$

where f_0, is an estimator of the residual spectrum at frequency zero and where $S(t)$ is a cumulative residual function:

$$S(t) = \sum_{r=1}^{t} \hat{u}_r \tag{42.16}$$

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 676 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 (42.9). Define the residuals from (42.9) as $\hat{\eta}_t(a) = d(y_t|a) - d(x_t|a)'\hat{\delta}(a)$, and let $\mathrm{SSR}(a) = \sum \hat{\eta}_t^2(a)$ 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:

$$P_T = (\mathrm{SSR}(\bar{a}) - \bar{a}\mathrm{SSR}(1)) / f_0 \tag{42.17}$$

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 676).

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:

$$\kappa = \sum_{t=2}^{T} (y_{t-1}^d)^2 / T^2 \tag{42.18}$$

The modified statistics may then be written as,

$$\begin{aligned}
MZ_\alpha^d &= (T^{-1}(y_T^d)^2 - f_0)/(2\kappa) \\
MZ_t^d &= MZ_\alpha \times MSB \\
MSB^d &= (\kappa / f_0)^{1/2} \\
MP_T^d &= \begin{cases} (\bar{c}^2\kappa - \bar{c}\,T^{-1}(y_T^d)^2)/f_0 & \text{if } x_t = \{1\} \\ (\bar{c}^2\kappa + (1-\bar{c})\,T^{-1}(y_T^d)^2)/f_0 & \text{if } x_t = \{1, t\} \end{cases}
\end{aligned} \tag{42.19}$$

where:

$$\bar{c} = \begin{cases} -7 & \text{if } x_t = \{1\} \\ -13.5 & \text{if } x_t = \{1, t\} \end{cases} \tag{42.20}$$

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 676).

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,

$$\hat{f}_0 = \sum_{j=-(T-1)}^{T-1} \hat{\gamma}(j) \cdot K(j/l) \tag{42.21}$$

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:

$$\hat{\gamma}(j) = \sum_{t=j+1}^{T} (\tilde{u}_t \tilde{u}_{t-j}) / T \tag{42.22}$$

Note that the residuals $\tilde{u}_t$ that EViews uses in estimating the autocovariance functions in (42.22) will differ depending on the specified unit root test:

Unit root test	**Source of $\tilde{u}_t$ residuals for kernel estimator**
ADF, DFGLS	*not applicable.*
PP, ERS Point Optimal, NP	residuals from the Dickey-Fuller test equation, (42.4).
KPSS	residuals from the OLS test equation, (42.14).

EViews supports the following kernel functions:

Bartlett:	$K(x) = \begin{cases} 1 - \lvert x \rvert & \text{if } \lvert x \rvert \le 1.0 \\ 0 & \text{otherwise} \end{cases}$
Parzen:	$K(x) = \begin{cases} 1 - 6x^2(1 - \lvert x \rvert) & \text{if } 0.0 \le \lvert x \rvert \le 0.5 \\ 2(1 - \lvert x \rvert)^3 & \text{if } 0.5 < \lvert x \rvert \le 1.0 \\ 0 & \text{otherwise} \end{cases}$
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 678.

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:

$$\Delta\tilde{y}_t = \alpha\tilde{y}_{t-1} + \varphi \cdot \tilde{x}_t{}'\delta + \beta_1\Delta\tilde{y}_{t-1} + \ldots + \beta_p\Delta\tilde{y}_{t-p} + u_t \tag{42.23}$$

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	$\tilde{y}_t = y_t - x_t'\delta(0)$, and $\varphi = 0$.
GLS detrended	$\tilde{y}_t = y_t - x_t'\delta(\bar{a}) = y_t^d$. and $\varphi = 0$.

where $\delta(a)$ are the coefficient estimates from the regression defined in (42.9).

The AR spectral estimator of the frequency zero spectrum is defined as:

$$\tilde{f}_0 = \hat{\sigma}_u^2 / (1 - \hat{\beta}_1 - \hat{\beta}_2 - \ldots - \hat{\beta}_p) \tag{42.24}$$

where $\hat{\sigma}_u^2 = \sum \tilde{u}_t^2 / T$ is the residual variance, and $\hat{\beta}$ are the estimates from (42.23). 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 678.

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	*not applicable*
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. (See "Automatic Bandwidth Selection" on page 1343 for discussion.)

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:

$$\tau = \alpha^2 \sum_t \tilde{y}_{t-1}^2 / \hat{\sigma}_u^2 \tag{42.25}$$

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 677, when estimating f_0. Ng and Perron (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:

$$k_{\max} = \mathrm{int}(\min(T/3, 12) \cdot (T/100)^{1/4}) \tag{42.26}$$

See Hayashi (2000, p. 594) for a discussion of the selection of this upper bound.

Unit Root Tests with a Breakpoint

The use of unit root tests to distinguish between trend and difference stationary data has become an essential tool in applied research. Accordingly, EViews offers a variety of standard unit root tests, including augmented Dickey-Fuller (ADF), Phillips-Perron (PP), Elliot,

Rothenberg, and Stock (ERS), Ng and Perron (NP), and Kwiatkowski, Phillips, Schmidt, and Shin (KPSS) tests ("Unit Root Testing," on page 667).

However, as Perron (1989) points out, structural change and unit roots are closely related, and researchers should bear in mind that conventional unit root tests are biased toward a false unit root null when the data are trend stationary with a structural break. This observation has spurred development of a large literature outlining various unit root tests that remain valid in the presence of a break (see Hansen, 2001 for an overview).

EViews offers support for several types of modified augmented Dickey-Fuller tests which allow for levels and trends that differ across a single break date. You may compute unit root tests with a single break where:

- The break can occur slowly or immediately.
- The break consists of a level shift, a trend break, or both a shift and break.
- The break date is known, or the break date is unknown and estimated from the data.
- The data are non-trending or trending.

Background

We begin with a brief discussion of the specifications underlining the testing methodology. As always, our discussion is necessarily brief and we encourage you to consult the enclosed references for additional detail.

Our discussion follows the basic framework outlined in Perron (1989), Vogelsang and Perron (1998), Zivot and Andrews (1992), Banerjee *et al.* (1992) and others. For a useful overview of the literature, see Perron (2006). Note that our notation differs slightly from the above sources.

Break Variables

Before proceeding, it will be useful to define a few variables which allow us to characterize the breaks. Let $1(\cdot)$ be an indicator function that takes the value 1 if the argument $(\cdot)$ is true, and 0 otherwise. Then the following variables are defined in terms of a specified break date T_b,

- An *intercept break* variable

$$DU_t(T_b) = 1(t \geq T_b) \tag{42.27}$$

  that takes the value 0 for all dates prior to the break, and 1 thereafter.

- A *trend break* variable

$$DT_t(T_b) = 1(t \geq T_b) \cdot (t - T_b + 1) \tag{42.28}$$

which takes the value 0 for all dates prior to the break, and is a break date re-based trend for all subsequent dates.

- A one-time *break dummy* variable

$$D_t(T_b) = 1(t = T_b) \tag{42.29}$$

which takes the value of 1 only on the break date and 0 otherwise.

Note that following EViews convention, we define the break date as the first date for the new regime. This is in contrast to much of the literature which defines the break date as the last date of the previous regime.

The Model

Following Perron (1989), we consider four basic models for data with a one-time break. For non-trending data, we have a model with (O) a one-time change in level; for trending data, we have models with (A) a change in level, (B) a change in both level and trend, and (C) a change in trend.

In addition, we consider two versions of the four models which differ in their treatment of the break dynamics: the *innovational outlier* (IO) model assumes that the break occurs gradually, with the breaks following the same dynamic path as the innovations, while the *additive outlier* (AO) model assumes the breaks occur immediately. The tests considered here evaluate the null hypothesis that the data follow a unit root process, possibly with a break, against a trend stationary with break alternative.

Within this basic framework there are a variety of specifications for the null and alternative hypotheses, depending on the assumptions one wishes to make about the break dynamics, trend behavior, and whether the break date is known or determined endogenously.

As in Perron (1989), we consider two distinct approaches to modeling the break dynamics.

Innovational Outlier Tests

For the IO model, we consider the following general null hypothesis:

$$y_t = y_{t-1} + \beta + \psi(L)(\theta D_t(T_b) + \gamma DU_t(T_b) + \epsilon_t) \tag{42.30}$$

where ϵ_t are *i.i.d.* innovations, and $\psi(L)$ is a lag polynomial representing the dynamics of the stationary and invertible ARMA error process. Note that the break variables enter the model with the same dynamics as the ϵ_t innovations.

For our alternative hypothesis, we assume a trend stationary model with breaks in the intercept and trend:

$$y_t = \mu + \beta t + \psi(L)(\theta DU_t(T_b) + \gamma DT_t(T_b) + \epsilon_t) \tag{42.31}$$

with the breaks again following the innovation dynamics.

We may construct a general Dickey-Fuller test equation which nests the two hypotheses:

$$y_t = \mu + \beta t + \theta DU_t(T_b) + \gamma DT_t(T_b) + \omega D_t(T_b) + \alpha y_{t-1} + \sum_{i=1}^{k} c_i \Delta y_{t-i} + u_t \tag{42.32}$$

and use the t-statistic for comparing $\hat{\alpha}$ to 1 ($t_{\hat{\alpha}}$) to evaluate the null hypothesis. As with conventional Dickey-Fuller unit root test equations, the k lagged differences of the y are included in the test equations to eliminate the effect of the error correlation structure on the asymptotic distribution of the statistic.

Within this general framework, we may specify different models for the null and alternative by placing zero restrictions on one or more of the trend and break parameters β, θ, γ. ω. Following Perron (1989), Perron and Vogelsang (1992a, 1992b), and Vogelsang and Perron (1998), we consider four distinct specifications for the Dickey-Fuller regression which correspond to different assumptions for the trend and break behavior:

- Model 0: non-trending data with intercept break:

  $$y_t = \mu + \theta DU_t(T_b) + \omega D_t(T_b) + \alpha y_{t-1} + \sum_{i=1}^{k} c_i \Delta y_{t-i} + u_t \tag{42.33}$$

  Setting the trend and trend break coefficients β and γ to zero yields a test of a random walk against a stationary model with intercept break.

- Model 1: trending data with intercept break:

  $$y_t = \mu + \beta t + \theta DU_t(T_b) + \omega D_t(T_b) + \alpha y_{t-1} + \sum_{i=1}^{k} c_i \Delta y_{t-i} + u_t \tag{42.34}$$

  Setting the trend break coefficient γ to zero produces a test of a random walk with drift against a trend stationary model with intercept break.

- Model 2: trending data with intercept and trend break:

  $$y_t = \mu + \beta t + \theta DU_t(T_b) + \gamma DT_t(T_b) + \omega D_t(T_b) + \alpha y_{t-1} + \sum_{i=1}^{k} c_i \Delta y_{t-i} + u_t \tag{42.35}$$

  The unrestricted Dickey-Fuller equation tests the random walk with drift against a trend stationary with intercept and trend break alternative.

- Model 3: trending data with trend break:

  $$y_t = \mu + \beta t + \gamma DT_t(T_b) + \alpha y_{t-1} + \sum_{i=1}^{k} c_i \Delta y_{t-i} + u_t \tag{42.36}$$

  Setting the intercept break and break dummy coefficients θ and ω to zero tests a random walk with drift null against a trend stationary with trend break alternative.

  Note that the test equation for Model 3 follows the methodology of Zivot and Andrews (1992) and Banerjee *et al.* (1992) which does not nest the null and alternatives, as $DU_t(T_b)$ is absent from the test equation; see Vogelsang and Perron (1998), p. 1077 for discussion.

You should bear in mind that whether one specifies a known break date or estimates the break date from the data affects the allowable specifications for the null hypothesis.

If the break date is known as in Perron (1989), Models 0, 1, and 2 allow for breaks under the null hypothesis. Model 3 does not allow for a break under the null.

If the break date is estimated, the test statistics considered here do not permit a breaking trend under the null. Vogelsang and Perron (1998) offer a detailed discussion of this point, noting that this undesirable restriction is required to obtain distributional results for the resulting Dickey-Fuller t-statistic. They offer practical advice for testing in the case where you wish to allow $\gamma \neq 0$ under the null. See also Kim and Perron (2009) for more recent work that directly tackles this issue.

Additive Outlier Tests

The general AO model null hypothesis is:

$$y_t = y_{t-1} + \beta + \theta D_t(T_b) + \gamma DU_t(T_b) + \psi(L)\epsilon_t \tag{42.37}$$

where ϵ_t are *i.i.d.* innovations, and $\psi(L)$ is a lag polynomial representing the dynamics of the stationary and invertible ARMA error process, and β is a drift parameter. Note that the full impact of the break variables occurs immediately.

The alternative hypothesis is for a trend stationary model with possible breaks in the intercept and trend:

$$y_t = \mu + \beta t + \theta DU_t(T_b) + \gamma DT_t(T_b) + \psi(L)\epsilon_t \tag{42.38}$$

Testing for a unit root in the AO framework is a two-step procedure where we first use the intercept, trend, and breaking variables to detrend the series using OLS, and then use the detrended series to test for a unit root using a modified Dickey-Fuller regression.

In the first-step of the AO test, we detrend the data using a model with appropriate trend and break variables:

- Model 0: non-trending data with intercept break:

$$y_t = \mu + \theta DU_t(T_b) + y_t^* \tag{42.39}$$

- Model 1: trending data with intercept break:

$$y_t = \mu + \beta t + \theta DU_t(T_b) + y_t^* \tag{42.40}$$

- Model 2: trending data with intercept and trend break:

$$y_t = \mu + \beta t + \theta DU_t(T_b) + \gamma DT_t(T_b) + y_t^* \tag{42.41}$$

- Model 3: trending data with trend break:

$$y_t = \mu + \beta t + \gamma DT_t(T_b) + y_t^* \tag{42.42}$$

In the second-step, let $\hat{y}_t^*$ be the residuals obtained from the detrending equation. The resulting Dickey-Fuller unit root test equation is given by,

- Models 0, 1, 2:

$$\hat{y}_t^* = \sum_{i=0}^{k} \omega_i D_{t-i}(T_b) + \alpha \hat{y}_{t-1}^* + \sum_{i=1}^{k} c_i \Delta \hat{y}_{t-i}^* + u_t \tag{42.43}$$

- Model 3:

$$\hat{y}_t^* = \alpha y_{t-1} + \sum_{i=1}^{k} c_i \Delta \hat{y}_{t-i}^* + u_t \tag{42.44}$$

where we use the t-statistic for comparing $\hat{\alpha}$ to 1, $t_{\hat{\alpha}}$, to evaluate the null hypothesis.

These are standard augmented Dickey-Fuller equations with the addition of $k+1$ break dummy variables $D_{t-i}(T_b)$ in Equation (42.44) to eliminate the asymptotic dependence of the test statistic on the correlation structure of the errors and to ensure that the asymptotic distribution is identical to that of the corresponding IO specification. See Perron and Vogelsang (1992b) for discussion.

As with the IO tests, when we estimate the break date from the data, the distributional results require that there be no trend break under the null hypothesis. See Vogelsang and Perron (1998) and Kim and Perron (2009) for discussion.

Test Options

For a given test equation described above, you must choose a number of lags k to include in the test equation, and you must specify the candidate date T_b at which to evaluate the break. EViews offers a number of tools for you to use when making these choices.

Lag Selection

The theoretical properties of the test statistics requires that we choose the number of lag terms in the Dickey-Fuller equations k to be large enough to eliminate the effect of the correlation structure of the errors on the asymptotic distribution of the statistic

- Fixed (with observation-based suggestion from Said and Dickey, 1984).

All of the remaining methods are data dependent, and require specification of a maximum lag length k_{max}. A different optimal lag length k^* is obtained for each candidate break date.

- t-test.

  Following Perron (1989), Perron and Vogelsang (1992a, 1992b), and Vogelsang and Perron (1998), k^* is chosen so that the coefficient on the last included dependent variable lag difference is significant at a specified probability value, while the coefficients on the last included lag difference in higher-order autoregressions up to k_{max}

are all insignificant at the same level. The probability values for the t-statistics are computed using the t-distribution.

The t-test method requires the specification of a p-value for use in evaluating significance. The default p-value of 0.10 may be changed by the user.

- F-test.

  Based on an approach of Said and Dickey (1984) (see also Perron and Vogelsang, 1992a, 1992b), the approach uses an F-test of the joint significance of the lag coefficients for a given k^o against all higher lags up to k_{max}. If any of the tests against higher-order lags are significant at a specified probability level, we set $k^* = k^o + 1$. If none of the test statistics is significant, we lower k^o by 1 and continue. We begin the procedure with $k^o = k_{max} - 1$ and continue until we achieve a rejection with $k^* = k^o + 1$, or until the lower bound $k^o = 0$ is evaluated without rejection and we set $k^* = 0$.

  The F-test method requires the specification of a p-value for use in evaluating significance. The default p-value of 0.10 may be changed by the user.

- Information criterion

  Following the approach of Hall (1994) and Ng and Perron (1995), k^* is chosen to minimize the specified information criterion amongst models with 0 to k_{max} lags.

  You may choose between the Akaike, Schwarz, Hannan-Quinn, Modified Akaike, Modified Schwarz, Modified Hannan-Quinn. Note that the sample used for model selection excludes data using full set of lag differences up to k_{max}.

Break Date Selection

Perron (1989) specified an *a priori* fixed break date. Subsequent research (Zivot and Andrews, 1992; Banerjee *et al.*, 1992; Vogelsang and Perron, 1998) has focused on endogenously determining break dates from the data. EViews supports the following break date selection methods:

- Minimize the Dickey-Fuller t-statistic $t_{\hat{\alpha}}$.

  Select the date providing the most evidence against the null hypothesis of a unit root and in favor of the breaking trend alternative hypothesis.

- Minimize or maximize θ t-statistic ($t_{\hat{\theta}}$) Maximize $|\theta|$ t-statistic ($t_{|\hat{\theta}|}$), Minimize or Maximize γ t-statistic ($t_{\hat{\gamma}}$), Maximize $|\gamma|$ t-statistic ($t_{|\hat{\gamma}|}$), Maximize (θ, γ) F-statistic ($F_{(\hat{\theta}, \hat{\gamma})}$).

  Choose the date with the strongest evidence of a break. The alternative minimize and maximize options are provided to allow for evaluation of one-sided alternatives, and will produce different critical values for the final Dickey-Fuller test statistic and tests with greater power than the non-directional alternatives.

- User-specified break date.

  Specify a fixed break date. This option allows you to carry out the original Perron (1989) test.

For the automatic break selection methods, the following procedure is carried out. For each possible break date, the optimal number of lags k is chosen using the specified method, and the test statistic of interest is computed. The procedure is repeated for each possible break date, and the optimal break date is chosen from the candidate dates.

When the method is minimize $t_{\hat{\alpha}}$, all possible break dates are considered. For the methods involving θ or γ, trimming is performed to remove some endpoint values from consideration as the break date.

Computing a Unit Root with Breakpoint Test

To compute a breakpoint unit root test, open a series window and select **View/Unit Root Tests/Breakpoint Unit Root Test...** to display the dialog:

The dialog is divided into six sections.

- The first section tells EViews whether you wish to compute the test using the raw data (**Level**), or whether to test for higher order integration using differences (**1st difference** or **2nd difference**) of the original data.

- The **Trend specification** section determines the trend components that are included in the test. Using the **Basic** dropdown, you may choose between an **Intercept** only or an **Intercept and trend** specification. If you include a trend in the specification you

will be prompted to indicate which deterministic components are breaking by choosing **Intercept**, **Intercept and trend**, or **Trend** in the **Breaking** dropdown menu.

- The **Lag length** section describes the method for selecting lags k for each of the augmented Dickey-Fuller test specifications ("Lag Selection" on page 684). You may choose between **Akaike criterion** (AIC), **Schwarz criterion** (BIC), **Hannan-Quinn criterion** (HQC), **Modified Akaike**, **Modified Schwarz**, **Modified Hannan-Quinn**, **t-statistic**, **F-statistic**, and **Fixed** lag specifications. For all but the **Fixed** lag method, you must provide a **Max. lag** to test; by default, EViews will suggest a maximum lag based on the number of observations in the series. For the test methods (**t-statistic**, **F-statistic**), you must specify a p-value for the tests; for the **Fixed** lag method, you must specify the actual number of use using the **User lags** edit field.

- The **Break type** section allows you to choose between the default **Innovation outlier** and the **Additive outlier** specifications ("The Model" on page 681).

- The **Breakpoint selection** section specifies the method for determining the identity of the breakpoint ("Break Date Selection" on page 685).

  For a model with an intercept break, you may choose between minimizing the t-statistic for α in the ADF test (**Dickey-Fuller min-t**), minimizing the t-statistic for the intercept break coefficient (**Intercept break min-t**), maximizing the t-statistic for the break coefficient (**Intercept break max-t**), maximizing the absolute value of the t-statistic for the intercept break coefficient (**Intercept break max-abs-t**), or providing a specific date (**User-specified**).

  For models with a trend break, there will be corresponding entries for minimizing and maximizing the t-statistic or absolute value of the t-statistic for the trend break coefficient. For models with both an intercept and trend break you will be offered an addi-

tional choice of using the F-statistic for the break coefficients (**Incpt. + trend break max-F**) to select the breakpoint.

You will be prompted for specify a trimming percentage when employing methods that involve the t-statistic or F-statistic of the break coefficients, EViews will remove from consideration as the breakpoint this percentage of the observations from each endpoint.

For the **User-specified** break choice you will be prompted to specify a single date.

- Lastly, the **Additional output** controls the output produced by the view. The check-box **Display test and selection graphs** controls whether to show only the test results with the selected break, or to show the test results and graphs depicting the break selection criterion results for each candidate break.

If you provide a name in the **Results matrix** edit field, EViews will save the results from each of the candidate augmented Dickey-Fuller tests in workfile. The first column contains the observation identifier for the break; the second through fifth columns contain the autoregressive coefficient, autoregressive coefficient standard error, number of observations, number of variables, and number of selected lags in the Dickey-Fuller regressions.

If appropriate, the remaining columns contain results for the breakpoint selection, with the contents varying with the method chosen. When minimizing the Dickey-Fuller $t_{\hat{\alpha}}$, the output consists of a single column containing the $t_{\hat{\alpha}}$ statistics. For methods involving one of θ or γ, the output contains the coefficient value, standard error, and the corresponding t-statistic; for the F-statistic method, the output columns consist of the estimates of θ, the standard error of θ, the estimates of γ, the standard error of γ, and the F-statistic for testing the significance of the two coefficients.

Examples

As examples, we replicate some of the results given in Perron (1997), using data originally provided by Nelson and Plosser (1982). The dataset contains fourteen annual macroeconomic series with values between 1860 and 1988. These data are provided in the workfile "nelson_plosser.wf1".

Real GNP

To begin, we replicate the results in the second row of Table 3 in Perron (1997), which tests for a unit root in the log of real GNP using data between 1909 and 1970. We display the log of real GDP, and set the workfile sample to dates from 1909 to 1970 with the commands

```
smpl 1909 1970
show log(rgnp)
```

To perform the unit root test with breakpoints, we click on **View/Breakpoint Unit Root Test...** which brings up the test dialog. In this example Perron tests for the existence of a unit root of the data in levels. The test assumes an innovation outlier break, with a trend specification given by Model 2 (Equation (42.35), above); trending data with both intercept and trend break.

Perron selects a breakpoint by minimizing the Dickey-Fuller t-statistic, and selects a lag length using the F-test.

We can match these settings by clicking the **Level** and **Innovation Outlier** buttons, changing the **Basic Trend specification** to **Trend and Intercept** and the **Breaking Trend specification** to **Intercept**, selecting **Dickey-Fuller min-t** as the **Breakpoint selection**, and changing the **Lag length Method** to **F-statistic**:

Clicking **OK** produces the following results:

The top section of this output describes the test that was performed, with a description of the underlying series, the trend and break specification, and the break type. The second section displays the selected break date, which in this case is 1929. Recall that, unlike Perron, EViews reports the break date for the start of the new regime instead of the last date before of the old regime, so the EViews reported date of 1929 matches Perron's 1928 result. Lastly, we see that the selected number of lags for corresponding test regression, selected on the basis of F-statistic selection is eight.

The lower section reports the Augmented Dickey-Fuller t-statistic for the unit root test, along with Vogelsang's asymptotic p-values. Our test resulted in a statistic of -5.50, with a p-value less than 0.01, leading us to reject the null hypothesis of a unit root.

EViews also provides a graph of the Augmented Dickey-Fuller statistics and AR coefficients at each test date:

Both graphs show a large dip in 1929, leaving little doubt as to which date should be selected as the break point.

Employment

Our second example replicates row nine of Table 3 in Perron (2007). This example performs a unit root test on the log of employment using data from 1890 to 1970. We again begin with issuing commands to set the sample and display the log of employment:

```
smpl 1890 1970
show log(totalemp)
```

In this test, Perron again assumes an innovation outlier break, with a trend specification given by Model 2 (Equation (42.35), above); trending data with intercept and trend break. However Perron now selects the breakpoint corresponding to the minimum intercept break *t*-statistic, and selects the lag-length using the *t*-statistic method. We replicate these choices with the following dialog settings:

The first section of the results of this test are shown below:

Again, the top section of this output describes the test that was performed, notably the underlying series, the trend and break specifications, and the break type. From the second section we can see that again a date of 1929 was chosen as the most likely break date. The t-statistic based lag selection selected seven lags for this test regression.

The second section displays the test statistic and associated p-value. The statistic value of -4.918 matches the value report by Perron, and the p-value again means that we reject (at a 5% significance level) the null hypothesis of a unit root.

GNP Deflator

Our final example replicates row 12 of Table 3 in Perron (1997), and performs a unit root test with breaks on the log of the GNP deflator between 1889 and 1970. We set the workfile sample and display the log of the GNP deflator by issuing the commands

```
smpl 1889 1970
show log(gnpdeflat)
```

We can mimic Perron's results with the following dialog settings:

which yield results of:

Here, 1920 was selected at the most likely break date, and the automatic lag selection routine selected 9 lags.

The t-statistic of -3.869 matches that reported by Perron, and the corresponding p-value of 0.27 indicates we cannot reject the hypothesis that the log of the GNP deflator has a unit root.

Seasonal Unit Root Testing

An important element of time series data is *seasonality* or *cyclicality*. Typically, seasonality is treated as a stationary feature in most time series models. Nevertheless, non-stationarity, particularly of the unit-root kind, can be an important feature within the cyclical components themselves, and can give rise to similar inferential inaccuracies and concerns one often encounters with traditional unit root series. Accordingly, identifying the presence of unit roots at one or more seasonal frequencies is the subject of the battery of tests known as *seasonal unit root* tests.

EViews offers several seasonal unit root tests, including the classical Hylleberg, *et al.* (1990) test, the Smith and Taylor (1999) likelihood ratio test, the Canova and Hansen (1995) test, the Taylor (2003) robust stationarity test, and the Taylor (2005) variance ratio test.

The discussion to follow assumes familiarity with the theory of seasonal unit root tests.

Performing a Seasonal Unit Root Test in EViews

To begin, double click on a series name to open the series window. From there, select **View/ Unit Root Tests/Seasonal Unit Root Tests...**

To carry out a seasonal unit root test, there are three settings which must be specified.

First, a test type should be chosen from the dropdown **Test type**. Here, one of four tests can be selected.

- Traditional Hylleberg, Engle, Granger, and Yoo (HEGY) test
- HEGY Likelihood Ratio test
- Canova-Hansen test
- Variance ratio test

In the **Options** group, three options should be specified: **Periodicity**, **Non-Seasonal Deterministics**, and **Seasonal Deterministics**:

- Under **Periodicity** one can select the seasonal frequency. EViews offers $2, 4, 5, 6, 7, 12$ as choices.
- Next, under the **Non-Seasonal Deterministics**, one can select either **None**, **Constant**, or **Constant and Trend**.
- Similarly, for **Seasonal Deterministics**, one can select **None**, **Seasonal Dummies**, **Spectral Intercepts**, or **Spectral Intercepts and Trends**. Note that some options are not available for all tests. For instance, the traditional HEGY test does not support spectral intercepts (and trends), and the Canova-Hansen test does not support non-seasonal trends.

Furthermore, the traditional HEGY and the HEGY Likelihood Ratio test allow for configuration of the **Lag selection** procedures. In particular, one can select either automatic or user-specified lag selection. The default procedure is automatic selection using the AIC, but automatic selection supports:

- Akaike (AIC)
- Schwarz (SIC)
- Hannan-Quinn (HQ)
- t-Statistic

For the automatic selection methods, one must specify the maximum number of lags to consider (by default, 12), and, in the case of the t-statistic procedure, the significance level (by default, 0.05).

Note that additional options are displayed when carrying out the Canova-Hansen test:

- First, one can also configure long-run covariance estimation options by clicking on the **HAC Options** button. Here, you may tweak the kernel, bandwidth, and pre-whitening settings.
- Further, note that the Canova-Hansen test also defaults to including a lag of the dependent variable as an additional exogenous regressor. This can be removed by deselecting the appropriate checkbox beneath the **HAC Options** button.
- Specification of spectral frequencies to be tested can be selected in the **Restriction selection** checkbox. Note that when no frequencies are selected (default), all frequencies are tested.

Lastly, note that critical values for the traditional and likelihood-ratio HEGY tests have been computed by simulation, and in cases where relevant, linear interpolation is provided to adjust to actual sample sizes.

Examples

To illustrate the computation of seasonal root tests, we will use the famous airline passenger data from ("airpassengers.wf1") which is a monthly dataset of total international airline passengers from 1949 to 1960.

We start with the traditional HEGY test using the default settings. Open the AIRPASSENGERS series and click on **View/Unit Root Tests/Seasonal Unit Root Tests...** to display the dialog:

Click on **OK** to compute the tests at the default settings and display the results.

The top portion of the results, which describes the basics settings for the test and provides relevant test statistics for frequency $\omega_0 = 0$, frequency $\omega_6 = \pi$, and the remaining 5 harmonic pair frequencies $(\omega_k, 2\pi - \omega_k)$ for $1 \le k \le 5$. of the output is given below:

Seasonal Unit Root Test for AIRPASSENGERS
Method: Traditional HEGY
Periodicity (Seasons): 12
Non-Seasonal Deterministics: None
Seasonal Deterministics: None
Lag Selection: 2 (Automatic: AIC, maxlags=12)
Sample Size: 130

Null Hypothesis		Significance Level		
Unit root at:	Test Stat.	1%	5%	10%
Frequency 0	-2.390602			
n=120		-2.48	-1.92	-1.60
n=140		-2.55	-1.93	-1.60
n=130*		-2.52	-1.92	-1.60
Frequency 2PI/12 and 22PI/12	5.281402			
n=120		33.98	8.15	3.75
n=140		30.48	8.15	3.77
n=130*		32.23	8.15	3.76
Frequency 4PI/12 and 20PI/12	2.343020			
n=120		33.98	8.15	3.75
n=140		30.48	8.15	3.77
n=130*		32.23	8.15	3.76
Frequency 6PI/12 and 18PI/12	1.059733			
n=120		33.98	8.15	3.75
n=140		30.48	8.15	3.77
n=130*		32.23	8.15	3.76
Frequency 8PI/12 and 16PI/12	0.327604			
n=120		33.98	8.15	3.75
n=140		30.48	8.15	3.77
n=130*		32.23	8.15	3.76
Frequency 10PI/12 and 14PI/12	0.796306			
n=120		33.98	8.15	3.75
n=140		30.48	8.15	3.77
n=130*		32.23	8.15	3.76
Frequency PI	-2.519200			
n=120		-2.48	-1.92	-1.60
n=140		-2.55	-1.93	-1.60
n=130*		-2.52	-1.92	-1.60

For each test statistic, critical values are given at the 1%, 5%, and 10% levels. In addition, since critical values are computed via simulation for sample sizes in intervals of 20, three sets of critical values are provided for each test. The interpolated value for the actual sample

size, which in this case is $n = 130$, and the closest lower and upper values for which simulations exist. In this case, the latter two are $n = 120$ and $n = 140$:

The basic results are followed by results for joint tests of all seasonal frequencies and all frequencies, respectively:

All seasonal frequencies	2.176090			
n=120		30.98	7.50	3.48
n=140		27.88	7.51	3.51
n=130*		29.43	7.50	3.50
All frequencies	2.696959			
n=120		28.70	6.99	3.27
n=140		25.66	7.00	3.30
n=130*		27.18	7.00	3.29

The bottom portion shows the HEGY test regression output:

Dependent Variable: AIRPASSENGERS-AIRPASSENGERS(-12)
Method: Least Squares
Date: 02/14/19 Time: 11:01
Sample (adjusted): 1950M03 1960M12
Included observations: 130 after adjustments

Variable	Coefficient	Std. Error	t-Statistic	Prob.
OMEGA(0)	0.001604	0.000671	2.390602	0.0184
OMEGA(2PI/12)	0.010796	0.004636	2.328882	0.0216
OMEGA(22PI/12)	0.010260	0.004691	2.187309	0.0307
OMEGA(4PI/12)	0.006190	0.008734	0.708742	0.4799
OMEGA(20PI/12)	0.017791	0.008648	2.057264	0.0419
OMEGA(6PI/12)	0.014627	0.020420	0.716284	0.4753
OMEGA(18PI/12)	-0.026313	0.020687	-1.271952	0.2059
OMEGA(8PI/12)	0.014948	0.025895	0.577251	0.5649
OMEGA(16PI/12)	-0.014704	0.025880	-0.568154	0.5710
OMEGA(10PI/12)	-0.036253	0.029952	-1.210371	0.2286
OMEGA(14PI/12)	0.010163	0.030128	0.337344	0.7365
OMEGA(PI)	-0.144661	0.057424	-2.519200	0.0131
DEP(-1)	0.382001	0.110308	3.463036	0.0007
DEP(-2)	0.396210	0.098827	4.009115	0.0001

R-squared	0.649238	Mean dependent var	32.17692
Adjusted R-squared	0.609928	S.D. dependent var	17.48021
S.E. of regression	10.91739	Akaike info criterion	7.720031
Sum squared resid	13825.97	Schwarz criterion	8.028843
Log likelihood	-487.8020	Hannan-Quinn criter.	7.845512
Durbin-Watson stat	1.953684		

Observe that in this example, the automatic lag selection has chosen 2 lags of the dependent variable to be used as additional regressors. These are captured as variables DEP(-1) and DEP(-2).

Next, we compute the HEGY likelihood ratio test, with a constant and trend, and spectral intercepts and trends:

The top portion of output is:

Seasonal Unit Root Test for AIRPASSENGERS
Method: Likelihood-Ratio HEGY
Periodicity (Seasons): 12
Non-Seasonal Deterministics: Constant and trend
Seasonal Deterministics: Spectral intercepts and trends
Lag Selection: 11 (Automatic: AIC, maxlags=12)
Sample Size: 121

Null Hypothesis		Significance Level		
Unit root at:	Test Stat.	1%	5%	10%
Frequency 0	3.377344			
n=120		7.51	5.62	4.72
n=140		7.58	5.66	4.74
n=121*		7.52	5.62	4.73
Frequency 2PI/12 and 22PI/12	5.180804			
n=120		5.97	4.78	4.22
n=140		5.91	4.75	4.21
n=121*		5.96	4.78	4.22
Frequency 4PI/12 and 20PI/12	2.765391			
n=120		5.97	4.78	4.22
n=140		5.91	4.75	4.21
n=121*		5.96	4.78	4.22
Frequency 6PI/12 and 18PI/12	2.258036			
n=120		5.97	4.78	4.22
n=140		5.91	4.75	4.21
n=121*		5.96	4.78	4.22
Frequency 8PI/12 and 16PI/12	3.450085			
n=120		5.97	4.78	4.22
n=140		5.91	4.75	4.21
n=121*		5.96	4.78	4.22
Frequency 10PI/12 and 14PI/12	2.508899			
n=120		5.97	4.78	4.22
n=140		5.91	4.75	4.21
n=121*		5.96	4.78	4.22
Frequency PI	3.108038			
n=120		5.64	3.94	3.19
n=140		5.76	4.00	3.23
n=121*		5.64	3.94	3.19

The output is similar to that of the traditional HEGY test. The primary difference is that statistics are all computed as F-statistic and so all statistics are positive. Furthermore, since we have included spectral intercepts and trends, the latter variables are shown in the regression output in the second table. Not depicted here are the results for the joint tests, and the test regression equation.

Next, consider the Canova-Hansen test performed on the same data. Here, we specify spectral intercepts and use defaults for everything else:

For the Canova-Hansen test, there are four sections of output.

The first section displays the joint test statistic, along with the total number of restrictions imposed, and the relevant critical values. In the second section, each of the harmonic pairs used in the joint test statistic is tested individually. Note here that each of the harmonic restrictions has two test statistics. The statistic without the asterisk is the usual CH test statistic. The asterisk marked statistic is robust to unattended unit roots.

Seasonal Unit Root Test for AIRPASSENGERS
Method: Canova-Hansen
Null Hypothesis: No seasonal unit root.
Periodicity (Seasons): 12
Non-Seasonal Deterministics: Constant
Seasonal Deterministics: Seasonal constants
Sample Size: 143

		Significance Level		
Joint LM Statistic	K	1%	5%	10%
3.393862	12	3.510	2.960	2.690

Note: The K restrictions used in the regression are identified below.

		Significance Level		
Harmonic Pairs	LM Stat.	1%	5%	10%
2PI/12 and 22PI/12	2.261366	1.070	0.749	0.610
	2.749517*			
4PI/12 and 20PI/12	1.599241	1.070	0.749	0.610
	2.576787*			
6PI/12 and 18PI/12	1.404161	1.070	0.749	0.610
	2.562663*			
8PI/12 and 16PI/12	1.346824	1.070	0.749	0.610
	2.121649*			
10PI/12 and 14PI/12	1.015918	1.070	0.749	0.610
	1.594240*			

(*) The statistic is robust to unattended unit roots.

The third section shows individual test statistics (in this case 12 of them), where the restrictions are tested on an individual basis. Note that the test statistics that are robust to unattended unit roots are reported here for frequencies $\omega_0 = 0$ and $\omega_{s/2} = \pi$:

		Significance Level		
Individual Frequencies	LM Stat.	1%	5%	10%
0	0.104963	0.748	0.470	0.353
	0.084804*			
2PI/12	1.712091	0.748	0.470	0.353
22PI/12	1.689205	0.748	0.470	0.353
4PI/12	1.563690	0.748	0.470	0.353
20PI/12	0.576865	0.748	0.470	0.353
6PI/12	0.312542	0.748	0.470	0.353
18PI/12	0.890876	0.748	0.470	0.353
8PI/12	0.169422	0.748	0.470	0.353
16PI/12	1.178735	0.748	0.470	0.353
10PI/12	0.141598	0.748	0.470	0.353
14PI/12	0.914546	0.748	0.470	0.353
PI	0.109890	0.748	0.470	0.353
	0.204633*			

(*) The statistic is robust to unattended unit roots.

Finally, the last table consists of the test regression output:

Dependent Variable: AIRPASSENGERS
Method: Least Squares
Date: 02/14/19 Time: 11:03
Sample (adjusted): 1949M02 1960M12
Included observations: 143 after adjustments

Variable	Coefficient	Std. Error	t-Statistic	Prob.
C	4.388369	3.951141	1.110659	0.2688
SIN(2PI/12)	-15.49239	2.113800	-7.329168	0.0000
COS(2PI/12)	17.91936	2.168222	8.264539	0.0000
SIN(4PI/12)	18.88631	2.101338	8.987751	0.0000
COS(4PI/12)	15.60368	2.094874	7.448503	0.0000
SIN(6PI/12)	4.784571	2.083045	2.296911	0.0232
COS(6PI/12)	-12.90487	2.100065	-6.144985	0.0000
SIN(8PI/12)	11.51329	2.086945	5.516812	0.0000
COS(8PI/12)	6.345242	2.093160	3.031418	0.0029
SIN(10PI/12)	5.143249	2.093637	2.456610	0.0153
COS(10PI/12)	10.65219	2.086049	5.106397	0.0000
COS(PI)	1.542641	1.477327	1.044211	0.2983
DEP(-1)	0.992415	0.013134	75.55948	0.0000

R-squared	0.980021	Mean dependent var	281.4755
Adjusted R-squared	0.978177	S.D. dependent var	119.5509
S.E. of regression	17.66088	Akaike info criterion	8.667089
Sum squared resid	40547.87	Schwarz criterion	8.936439
Log likelihood	-606.6969	Hannan-Quinn criter.	8.776540
Durbin-Watson stat	1.619057		

Note: * indicates which variables constitute restrictions.

Our final example computes the Taylor variance ratio test. We will again stick with the default settings:

In this case, the table provides test statistics for each of the frequencies $\omega_0 = 0$, frequency $\omega_6 = \pi$, and the 5 harmonic pair frequencies $(\omega_k, 2\pi - \omega_k)$ for $1 \le k \le 5$, along with the relevant critical values:

Seasonal Unit Root Test for AIRPASSENGERS
Method: Taylor Variance Ratio Test
Periodicity (Seasons): 12
Non-Seasonal Deterministics: Constant
Seasonal Deterministics: Seasonal trigonometric constants
Sample Size: 133

Null Hypothesis		Significance Level		
Seasonal unit root at:	VR Stat.	1%	5%	10%
Frequency 0	0.099	0.005	0.010	0.015
Frequencies 2PI/12 and 22PI/12	0.062	0.012	0.019	0.024
Frequencies 4PI/12 and 20PI/12	0.059	0.012	0.019	0.024
Frequencies 6PI/12 and 18PI/12	0.055	0.012	0.019	0.024
Frequencies 8PI/12 and 16PI/12	0.039	0.012	0.019	0.024
Frequencies 10PI/12 and 14PI/12	0.036	0.012	0.019	0.024
Frequency PI	0.006	0.005	0.010	0.015

Note: Critical values are obtained under asymptotic sample sizes.

Cross-sectionally Independent Panel Unit Root Testing

The unit root test is a cornerstone of modern time series analysis. Nevertheless, a well-known drawback of many popular univariate unit root tests is a lack of statistical power in the presence of small to medium sized samples.

Since many economic time series have short samples but are observed over many cross-sections, multivariate unit root tests that combine results for different cross-sections or series, offer improved statistical power over their univariate counterparts. 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)

EViews computes five different independent cross-section 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). Notably all of these tests require cross-sectional independence. Tests which relax this assumption are described in "Cross-sectionally Dependent Panel Unit Root Tests" on page 716.

Bear in mind that 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 Independent 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. Additional background is provided in "Independent Panel Unit Root Tests Background" on page 710.

To begin, select **View/Unit Root Tests/Cross-sectionally Independent…** 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 testing using a 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 **Use 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 dropdown menu 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 "**Hadri**", corresponding to one or more of the tests listed above. The dropdown menu 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 dropdown menu (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 dropdown menu (**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 perform a panel unit root tests on real gross investment data (I) in the oft-cited Grunfeld data containing data on R&D expenditure and other economic measures for 10 firms for the years 1935 to 1954 found in “Grunfeld_Baltagi.WF1”. 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
Series: I
Date: 08/12/09 Time: 14:17
Sample: 1935 1954
Exogenous variables: Individual effects
Automatic selection of maximum lags
Automatic lag length selection based on SIC: 0 to 3
Newey-West automatic bandwidth selection and 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** dropdown menu. 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	Band-width	Obs
1	0.22672	11314.	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.

Independent Panel Unit Root Tests Background

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:

$$y_{it} = \rho_i y_{it-1} + X_{it}\delta_i + \epsilon_{it} \tag{42.45}$$

where $i = 1, 2, \ldots, N$ cross-section units or series, that are observed over periods $t = 1, 2, \ldots, T_i$.

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:

$$\Delta y_{it} = \alpha y_{it-1} + \sum_{j=1}^{p_i} \beta_{ij} \Delta y_{it-j} + X'_{it}\delta + \epsilon_{it} \tag{42.46}$$

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:

$$H_0\colon\ \alpha = 0 \tag{42.47}$$

$$H_1\colon\ \alpha < 0 \tag{42.48}$$

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 $j = 1, \ldots, p_i$) and the exogenous variables X_{it}. The estimated coefficients from these two regressions will be denoted $(\hat{\beta}, \hat{\delta})$ and $(\dot{\beta}, \dot{\delta})$, 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:

$$\Delta \bar{y}_{it} = \Delta y_{it} - \sum_{j=1}^{p_i} \hat{\beta}_{ij} \Delta y_{it-j} - X'_{it}\hat{\delta} \tag{42.49}$$

Likewise, we may define the analogous $\bar{y}_{it-1}$ using the second set of coefficients:

$$\bar{y}_{it-1} = y_{it-1} - \sum_{j=1}^{p_i} \dot{\beta}_{ij} \Delta y_{it-j} - X'_{it}\dot{\delta} \tag{42.50}$$

Next, we obtain our proxies by standardizing both $\Delta \bar{y}_{it}$ and $\bar{y}_{it-1}$, dividing by the regression standard error:

$$\begin{aligned} \Delta \tilde{y}_{it} &= (\Delta \bar{y}_{it} / s_i) \\ \tilde{y}_{it-1} &= (\bar{y}_{it-1} / s_i) \end{aligned} \tag{42.51}$$

where s_i are the estimated standard errors from estimating each ADF in Equation (42.46).

Lastly, an estimate of the coefficient α may be obtained from the pooled proxy equation:

$$\Delta \tilde{y}_{it} = \alpha \tilde{y}_{it-1} + \eta_{it} \tag{42.52}$$

LLC show that under the null, a modified t-statistic for the resulting $\hat{\alpha}$ is asymptotically normally distributed

$$t_\alpha^* = \frac{t_\alpha - (N\tilde{T}) S_N \hat{\sigma}^{-2} se(\hat{\alpha}) \mu_{m\tilde{T}}^*}{\sigma_{m\tilde{T}}^*} \rightarrow N(0, 1) \tag{42.53}$$

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:

$$\tilde{T} = T - \left(\sum_i p_i / N \right) - 1 \tag{42.54}$$

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:

$$\begin{aligned} \Delta \tilde{y}_{it} &= \left(\Delta y_{it} - \sum_{j=1}^{p_i} \hat{\beta}_{ij} \Delta y_{it-j} \right) / s_i \\ \tilde{y}_{it-1} &= \left(y_{it-1} - \sum_{j=1}^{p_i} \hat{\beta}_{ij} \Delta y_{it-j} \right) / s_i \end{aligned} \tag{42.55}$$

where $\hat{\beta}$, $\dot{\beta}$, and s_i are as defined for LLC.

Second, the proxies are transformed and detrended,

$$\begin{aligned} \Delta y_{it}^* &= \sqrt{\frac{(T-t)}{(T-t+1)}}\left(\Delta \tilde{y}_{it} - \frac{\Delta \tilde{y}_{it+1} + \ldots + \Delta \tilde{y}_{iT}}{T-t}\right) \\ y_{it}^* &= \tilde{y}_{it} - \tilde{y}_{i1} - \frac{t-1}{T-1}(\tilde{y}_{iT} - \tilde{y}_{i1}) \end{aligned} \tag{42.56}$$

The persistence parameter α is estimated from the pooled proxy equation:

$$\Delta y_{it}^* = \alpha y_{it-1}^* + \nu_{it} \tag{42.57}$$

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:

$$y_{it} = \delta_i + \eta_i t + \epsilon_{it} \tag{42.58}$$

Given the residuals $\hat{\epsilon}$ from the individual regressions, we form the LM statistic:

$$LM_1 = \frac{1}{N}\left(\sum_{i=1}^{N}\left(\sum_t S_i(t)^2 / T^2\right) / \dot{f}_0\right) \tag{42.59}$$

where $S_i(t)$ are the cumulative sums of the residuals,

$$S_i(t) = \sum_{s=1}^{t} \hat{\epsilon}_{it} \tag{42.60}$$

and $\dot{f}_0$ is the average of the individual estimators of the residual spectrum at frequency zero:

$$\dot{f}_0 = \sum_{i=1}^{N} f_{i0} / N \tag{42.61}$$

EViews provides several methods for estimating the f_{i0}. See "Unit Root Testing" on page 667 for additional details.

An alternative form of the LM statistic allows for heteroskedasticity across i:

$$LM_2 = \frac{1}{N}\left(\sum_{i=1}^{N}\left(\sum_t S_i(t)^2 / T^2\right) / f_{i0}\right) \tag{42.62}$$

Hadri shows that under mild assumptions,

$$Z = \frac{\sqrt{N}(LM - \xi)}{\zeta} \rightarrow N(0, 1) \tag{42.63}$$

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:

$$\Delta y_{it} = \alpha y_{it-1} + \sum_{j=1}^{p_i} \beta_{ij} \Delta y_{it-j} + X'_{it}\delta + \epsilon_{it} \tag{42.64}$$

The null hypothesis may be written as,

$$H_0\colon\ \alpha_i = 0, \text{ for all } i \tag{42.65}$$

while the alternative hypothesis is given by:

$$H_1\colon \begin{cases} \alpha_i = 0 & \text{for } i = 1, 2, \ldots, N_1 \\ \alpha_i < 0 & \text{for } i = N+1, N+2, \ldots, N \end{cases} \tag{42.66}$$

(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, $t_{iT_i}(p_i)$:

$$\overline{t_{NT}} = \left(\sum_{i=1}^{N} t_{iT_i}(p_i) \right) / N \tag{42.67}$$

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 $\overline{t_{NT}}$ 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 (42.64) may be non-zero for some cross-sections, IPS show that a properly standardized $\overline{t_{NT}}$ has an asymptotic standard normal distribution:

$$W_{\overline{t_{NT}}} = \frac{\sqrt{N}\left(\overline{t_{NT}} - N^{-1} \sum_{i=1}^{N} E(\overline{t_{iT}}(p_i)) \right)}{\sqrt{N^{-1} \sum_{i=1}^{N} Var(\overline{t_{iT}}(p_i))}} \rightarrow N(0, 1) \tag{42.68}$$

The expressions for the expected mean and variance of the ADF regression t-statistics, $E(\overline{t_{iT}}(p_i))$ and $Var(\overline{t_{iT}}(p_i))$, 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

$$-2\sum_{i=1}^{N}\log(\pi_i) \rightarrow \chi^2_{2N} \tag{42.69}$$

In addition, Choi demonstrates that:

$$Z = \frac{1}{\sqrt{N}}\sum_{i=1}^{N}\Phi^{-1}(\pi_i) \rightarrow N(0,1) \tag{42.70}$$

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 676 for details.

Cross-sectionally Dependent Panel Unit Root Tests

In "Cross-sectionally Independent Panel Unit Root Testing" on page 705, we described first generation panel unit root tests u on pooled panel data, with (possibly) individual trend, intercepts, and lag coefficients when there is cross-sectional independence.

The assumption of cross-sectional independence can be difficult to justify, as cross-sections are often influenced by common forces, termed factors. For instance, cross-country purchasing power parity (PPP) data is often anchored to a common numeraire currency such as the US dollar, so that movements in the latter affect the cross-sectional PPP in correlated fashion. Tests which account for cross-sectional dependence have been termed *second generation panel unit root tests*.

EViews currently supports two important second generation contributions: Bai and Ng's (2004) Panel Analysis of Nonstationarity in Idiosyncratic and Common Components (PANIC), and Pesaran's (2007) Cross-sectionally Augmented IPS (CIPS).

As with independent panel unit root tests, EViews supports testing in different settings involving multiple series: as a series view (if the workfile is panel structured), as a group view, or as a pool view.

Performing Dependent Panel Unit Root Tests in EViews

Second generation panel unit root tests may be performed on a single series in a panel workfile or on a group of series in a workfile. To perform the test in the panel setting, you should open the series and then click on **View/Unit Root Tests/Cross-Sectionally Dependent...** In the group setting, open the group and click on the same view entry. EViews will display the dialog:

There is a lot to this dialog so we will examine each section in turn.

Test

The **Test** section of the dialog is used to specify the basic test type:

The **Type** dropdown lets you choose between the two second generation panel unit root test types: **Bai and Ng - PANIC** and **Pesaran - CIPS**.

The **Deterministics** dropdown offers three choices for the deterministic terms to be included in the specification: **None**, **Constant**, or **Constant and trend**.

ADF Lag Selection

The **ADF lag selection** section controls the specification of the ADF test specifications in the PANIC and CIPS tests.

The **Method** dropdown offers several options for determining the number of lagged difference terms p to use in the ADF specifications. You may use the dropdown to select between using information criteria (Akaike, Schwarz, Hannan-Quinn, Modified Akaike, Modified, Schwarz, Modified Hannan-Quinn), performing *t*-tests of lag difference variable significance, or providing a user-specified fixed value.

For the information criteria and *t*-test options, you will be prompted for the maximum number of lags to consider; for the user-specified fixed value, you will be prompted to enter the value itself.

Note that EViews pre-fills the **Maximum lags** edit field with a default value derived from a variation of the Schwert (1989) rule: let there be a lag choice function $L(k) = k(T/100)^{1/4}$ for $k \in \{2, 4, 6, 8, 10, 12\}$ and let $T^* = \min(N, T)$; then the default maximum lag $L^* = \max_k\{L(k) | L(k) < T^*\}$.

PANIC MQ Options

If you choose to perform a PANIC test the **PANIC MQ options** section offers options for controlling the computation:

The **Type** dropdown lets you choose between the **MQC** and **MQF** statistics.

The **Significance level** edit field lets you choose the level of significance at which EViews will perform the test as the p-value for these statistics is obtained by simulation.

The remaining options will change with your selection of the PANIC type:

- If you choose to compute the MQ_c, the **Long-Run Variance Options** button will be enabled.

- If you choose MQ_f, the **Lag selection options** will be enabled so that you may control the order of the VAR(p). The settings are as described in "ADF Lag Selection," beginning on page 718.

Factor Selection

The **Factor selection** options control the selection of number of factors method used in the PANIC procedure:

The **Method** dropdown lets you choose between the **Simple**, **Bai and Ng**, and **Ahn and Horenstein**. In turn, each of these choices offers different options.

- If you select **Simple**, you will choose a **Criterion** from the dropdown. You may choose from the following selections: **Minimum of...**, **Maximum of...**, **Average of criteria**.

For the first two selections, the number of factors is determined as the minimum or maximum of the **Max. Factors** (discussed below), the specified **Minimum eigenvalue**, and **Cumulative proportion** of eigenvalues. The **Average of criteria** will use the average of the other two results.

- If you select **Bai and Ng**, EViews will compute the Bai and Ng (2002) as discussed in "Bai and Ng" on page 652 of *User's Guide I*.

The **Criterion** dropdown offers all of the information criteria described by Bai and Ng (**ICP1**, **ICP2**, **ICP3**, **PCP1**, **PCP2**, **PCP3**) as well as the **Average of criteria** which averages the criteria prior to determining the optimal number of factors.

- If you select **Ahn and Horenstein**, EViews will compute the Ahn and Horenstein (2013) number of factor determination procedure ("Ahn and Horenstein" on page 653 of *User's Guide I*). The dialog looks the same as for **Bai and Ng**, but the **Criterion** dropdown now offers **Eigenvalue Ratio**, **Growth ratio**, and **Average of criteria**.
- For **User**, the number of factors is specified by the user in the edit box

In all of the automatic factor selection procedures above is the selection of maximum factors to be considered. While ultimately an arbitrary selection, EViews offers some suggestions typically seen in the literature. In particular, as in the case of ADF selection, consider the **Max. factors** dropdown which provides the following options:

- **Schwert**: let there be a lag choice function $L(k) = k(T/100)^{1/4}$ for $k \in \{2, 4, 6, 8, 10, 12\}$ and let $T^* = \min(N, T)$; then returns the value $L^* = \max_k\{L(k) | L(k) < T^*\}$.
- **Ahn and Horenstein**: uses the suggestion made on page 1208 in Ahn and Horenstein (2013).
- **min(root(N), root(T))**: uses $\min(\sqrt{N}, \sqrt{T})$.
- **min(N,T):** uses $\min(N, T)$.
- **User**: users can specify an arbitrary value in the **User maximum factors** edit field.

Furthermore, as discussed in Bai and Ng (2002) and Ahn and Horenstein (2013), factor selection procedures can be improved dramatically if time and/or cross-section dimensions are demeaned and/or standardized. To allow for these possibilities, there are four check-boxes associated with each of these options and combinations:

- **Time-demean**: demeans the time dimension.
- **Time-standardize**: standardizes the time dimension.
- **Cross-demean**: demeans the cross-section dimension.
- **Cross-standardize**: standardizes the cross-section dimension.

p-value Simulation

In case of the PANIC tests, inference is based on simulated p-values. In this setting, you can may specify simulation options in the p-value **simulations** section.

In particular, the number of **Monte Carlo replications**, **Asymptotic length** and **Random seed** can be set in their respective edit fields. Note that the latter is optional. Furthermore, the **Random generator** dropdown offers various random number generators. Details associated with the latter options can be obtained in the EViews documentation for rndseed.

Dependent Panel Unit Root Test Examples

To illustrate some of the features of second generation panel unit root tests in EViews, we consider three examples using data from Pesaran (2007). These panel data feature quarterly real exchange rates from 17 OECD countries for the period 1973q1 to 1998q4. Data can be downloaded from

http://qed.econ.queensu.ca/jae/2007-v22.2/pesaran/

in the file "real-xrate.zip".

You may unzip the file then drag-and-drop it onto your EViews to create the new workfile. As is, however, the resulting unstacked, undated workfile is not set up for panel or group unit root testing.

From the command line you can set up the group form of the unstacked data with the following command:

```
pagestruct(freq=q, start=1973q1)
group oecd_rer *_rate
```

The first line specifies the frequency of the data (the EViews auto-input was unable to do so since there are no dates in the file). The next line creates a group containing all of the rate series. We can conduct the unit root tests using the group object OECD_RER.

Alternately, we can choose to stack the data in a new page, and conduct the test on the stacked panel data. The command

```
pagestack ?real_exchange_rate @ ?*
```

creates a new page containing a stacked panel data series REAL_EXCHANGE_RATE.

Whichever form with which we wish to work, we will use the sample statement

```
smpl 1974 1998
```

to the example in Pesaran (2007).

Lastly, note that the examples below use the REAL_EXCHANGE_RATE series in the workfile tab "Realexchangerates7398_STK. All of the examples may also be performed using the group object OECD_RER in the workfile tab "Realexchangerates7398".

PANIC MQc Example

This example will use default values for all inputs. To use the panel data form of the test, click on the workfile tab "Realexchangerates7398_STK", and then open the panel series

REAL_EXCHANGE_RATE. Click on View/Unit Root Tests/Cross-Sectionally Dependent..., to access the second generation panel unit root tests.

As PANIC MQ_c is the default test, at this point, one can simply click on **OK** to compute this test.

The output displays a spool with the spool tree listing the summary, factor selection, common factors, and individual and pooled idiosyncratic elements. The first of these is a summary of the PANIC procedure performed. The second is a table listing the details of the factor selection procedure. In particular, since the selection was the average of the Bai and Ng (2002) test statistics, listed in the table are the factor selections based on each of the six individual statistic variants as well as their average which is used in the remainder of the PANIC procedure.

Next, the common factors table displays the PANIC test for how many common factors influence the panel. The test statistic and associated p-value are reported. In this particular case, that number selected 7, which is also the maximum number of factors allowed using the Schwert (1989) rule.

The table that follows displays the individual ADF test statistics for the idiosyncratic elements associated with each cross-section. The ADF lags are selected for each cross-section independently using the same automatic selection rule, which in this case happens to be AIC with 7 maximum lags. Furthermore, the columns **t-stat** and **p-value** report the ADF t-statistic and associated p-value corresponding to the null hypothesis that the idiosyncratic element exhibit a unit root. It is readily seen that apart from Norway, all other cross-sections exhibit a unit root at the 5% significance level.

Lastly, the final table is a pooled version of the individual ADF test statistics in the previous table. Here, we cannot reject the null hypothesis that all of the cross-sections are simultaneously co-integrated.

PANIC MQf Example

For the second example, using the same data, the objective is to perform a PANIC unit root test using the MQ_f test statistic and the Ahn and Horenstein (2013) factor selection procedure.

From the series window click on **View/Unit Root Tests/Cross-Sectionally Dependent...**. In the **PANIC MQ options** section select **MQF** in the **Type** dropdown. Next, under the **Factor selection** group, select **Ahn and Horenstein** from the **Method** dropdown. Furthermore, for the **Max. factors**, select **Ahn and Horenstein**. Lastly, ensure that **Time-demean**, **Time-standardize**, **Cross-demean**, and **Cross-standardize** are all checked. Click on **OK**.

The spool output display is analogous to that in "PANIC MQc Example" on page 721. Here, the factor selection procedure is based on Ahn and Horenstein (2013).

Evidently, both the eigenvalue ratio and growth ratio statistics select a single common factor using the Ahn and Horenstein (2013) suggestion for maximum factors, which here happens to be 1.

Since the number of common factors selected is 1, the test associated with the common factors is based on an ADF regression. The next table displays the results of this test.

Here, we see that the test fails to reject the null hypothesis that the single common factor is non-stationary.

Next, as in the previous example, table that follows displays the individual ADF test statistics for the idiosyncratic elements associated with each cross-section. Unlike the previous example, there are significantly more cross-sections for which one can argue rejection of the unit root null hypothesis.

Lastly, the pooled version of the latter tests has a p-value which effectively zero, and therefore we fail to reject the null hypothesis that all cross-sections are not cointegrated.

CIPS Example

The final example uses the same data. Here, the objective is to perform the Pesaran (2007) CIPS test. From the series window click on **View/Unit Root Tests/Cross-Sectionally Dependent...**. From the **Type** dropdown, select **Pesaran – CIPS**. To conform with the analysis in the original paper, change the **Method** dropdown in the **ADF lag selection** group to **User**. We will stick with a single lag for the ADF regression. Hit **OK**.

The output displays a spool with the spool tree listing the summary, CIPS and CADF unit root tests, as well as CIPS and CADF critical value interpolations.

The first of the spool tables is a summary of the CIPS procedure performed. The second is a table listing the details of the CIPS test. In particular, the first part of this table displays the critical values for the usual CIPS statistic as well as its truncated version. The second portion of this table summarizes the test results. Note that the t-statistic is displayed along with the associated p-values which are summarized categorically based on the critical values tabulated in Pesaran (2007). Here, we cannot reject the null hypothesis below the 10% significance level. Furthermore, the outcome of this table matches the results reported in Table XI of the original paper.

The next table summarizes the ADF test statistics that are used in the computation of the CIPS statistic.

Cross-sectional ADF unit root test
Null hypothesis: Unit root for specified cross-section
Lag selection: 1 (USER)

Critical values:

Level	CADF	Trunc. CADF
1%	-3.86	-3.86
5%	-3.25	-3.25
10%	-2.92	-2.92

Test results:

	CADF		Truncated CADF	
Cross-section	t-stat	p-value	t-stat	p-value
AUSTRALIA	-1.79377	>=10%	-1.79377	>=10%
AUSTRIA	-1.22496	>=10%	-1.22496	>=10%
BELGIUM	-1.33671	>=10%	-1.33671	>=10%
CANADA	-0.23423	>=10%	-0.23423	>=10%
DENMARK	-1.97252	>=10%	-1.97252	>=10%
FINLAND	-2.45363	>=10%	-2.45363	>=10%
FRANCE	-3.23595	>=10%	-3.23595	>=10%
GERMANY	-2.16387	>=10%	-2.16387	>=10%
ITALY	-2.51329	>=10%	-2.51329	>=10%
JAPAN	-1.79572	>=10%	-1.79572	>=10%
NETHERLANDS	-1.80378	>=10%	-1.80378	>=10%
NEW_ZEALAND	-3.12134	>=10%	-3.12134	>=10%
NORWAY	-2.56687	>=10%	-2.56687	>=10%
SPAIN	-2.72901	>=10%	-2.72901	>=10%
SWEDEN	-1.72785	>=10%	-1.72785	>=10%
SWITZERLAND	-2.56148	>=10%	-2.56148	>=10%
UK	-1.98414	>=10%	-1.98414	>=10%

Here, the first portion of the table summarizes the critical values associated with the CADF statistic as well as its truncated version. The second portion of this table summarizes the test results for each of the cross-sections. In particular, these are t-statistics associated with the cross-sectionally augmented ADF regressions for each of the cross-sections. The table summarizes the t-statistic and p-value category for each of the CADF and truncated CADF test statistics. In particular, here we cannot reject the unit root null hypotheses at significance levels less than 10% for any of the cross-sections.

At last, the remaining two tables summarize the interpolation computation used to derive the critical values reported earlier.

FINLAND	-2.45363	>=10%	-2.45363	>=10%
FRANCE	-3.23595	>=10%	-3.23595	>=10%
GERMANY	-2.16387	>=10%	-2.16387	>=10%
ITALY	-2.51329	>=10%	-2.51329	>=10%
JAPAN	-1.79572	>=10%	-1.79572	>=10%
NETHERLANDS	-1.80378	>=10%	-1.80378	>=10%
NEW_ZEALAND	-3.12134	>=10%	-3.12134	>=10%
NORWAY	-2.56687	>=10%	-2.56687	>=10%
SPAIN	-2.72901	>=10%	-2.72901	>=10%
SWEDEN	-1.72785	>=10%	-1.72785	>=10%
SWITZERLAND	-2.56148	>=10%	-2.56148	>=10%
UK	-1.98414	>=10%	-1.98414	>=10%

In particular, if the T or N columns display values, it implies that interpolation was done on that dimension of the data.

Dependent Panel Unit Root Tests Background

To formalize the discussion, let $i = 1, \ldots, N$ and $t = 1, \ldots, T$, respectively, denote the cross-section and time indices, and consider the following integrated panel data model:

$$Y_{it} = D_{it} + u_{it} \tag{42.71}$$

where Y_{it} is the observed data, D_{it} are the model deterministic dynamics, and u_{it} are the innovations.

The traditional first generation panel unit root tests of Maddala and Wu (1999), Hadri (2000), Levin, Lin and Chu (2002), and Im, Pesaran and Shin (2003), while differing in unit root parameter heterogeneity assumptions and in test construction approach, all assume that u_{it} are independent across i.

To introduce cross-sectional dependence, we model u_{it} as having a factor model structure,

$$u_{it} = \lambda_i' F_t + \epsilon_{it} \tag{42.72}$$

where F_t is a vector of r common factors generated by a multivariate white noise process η_t,

$$(1 - L) F_t = C(L) \eta_t \tag{42.73}$$

λ_i denotes a vector of cross-section specific factor loadings, and e_{it} is a multivariate idiosyncratic component which is cross-sectionally independent as F_t is assumed to account for all inter-cross-sectional correlations. When e_{it} are not cross-sectionally independent, the factor model governing u_{it} is said to be *approximate*. As usual, L is the lag operator, and we have the polynomial:

$$C(z) = \sum_{j=0}^{\infty} C_j z^j \tag{42.74}$$

Substituting, an integrated panel model with cross-sectional dependence may be formulated as

$$Y_{it} = D_{it} + \lambda_i' F_t + \epsilon_{it} \tag{42.75}$$

Assumptions imposed on F_t and v_{it} govern the nature of the process for Y_{it}. For example, when $C(1)$ has rank 0, F_t is non-stochastic and stationary, or $I(0)$. Alternately, when $C(1)$ has rank $0 < r_1 \leq r$, F_t is an $I(1)$ process driven by r_1 linearly independent stochastic trends. In particular, when $r_1 < r$, F_t has r_1 common trends and $r_0 = r - r_1$ cointegrated stationary factors.

A crucial feature of these models is that the common factors F_t and idiosyncratic components v_{it} can each be $I(0)$ or $I(1)$. This generality permits a wide spectrum of possible outcomes for the properties of the observed Y_{it}. For instance, Y_{it} is considered $I(1)$ if *either* F_t or ϵ_{it} is $I(1)$. Alternately, if the number of common trends $r_1 > 0$, Y_{it} is deemed conditionally stationary or non-stationary, depending on the properties of the idiosyncratic component $\epsilon_{it} = Y_{it} - \lambda_i' F_t$.

For further detail, see Bai and Ng (2004) and Banerjee and Wagner (2009).

PANIC Unit Root Testing

Bai and Ng's (2004) PANIC (Panel Analysis of Nonstationarity in Idiosyncratic and Common Components) test is widely considered to be the first unit root test for panel data with cross-sectional dependence. The PANIC test is based on a factor model in which non-stationarity can arise from common factors, idiosyncratic components, or both.

The observed series is modeled using Equation (42.75):

$$Y_{it} = D_{it} + \lambda_i' F_t + \epsilon_{it} \tag{42.76}$$

where

$$\begin{aligned}(1-L)F_t &= C(L)\eta_t \\ \epsilon_{it} &= \rho\epsilon_{it-1} + v_{it}\end{aligned} \tag{42.77}$$

and v_{it} is a mean zero, stationary, and invertible MA process. PANIC assumes a balanced sample where both N and T are large, such that $N/T \to 0$ as both $N, T \to \infty$.

Estimation requires a suitable stationary transformation of Y_{it}.

- If D_{it} contains at most a constant, we take the first differences of Y_{it}

$$y_{it} = \Delta Y_{it} = Y_{it} - Y_{it-1} \tag{42.78}$$

- If D_{it} contains a linear trend, Y_{it} is both demeaned and differenced,

$$y_{it} = \Delta Y_{it} - \Delta \overline{Y}_{it}, \tag{42.79}$$

  where

$$\Delta \overline{Y}_{it} = \frac{1}{T-1}\sum_{s=2}^{T} Y_{is} \tag{42.80}$$

Then the transformed equation is

$$y_{it} = \lambda_i' f_t + \omega_{it} \tag{42.81}$$

where $f_t = \Delta F_t$ and $\omega_{it} = \Delta\epsilon_{it}$ for $i = 1, \ldots N$ and $t = 2, \ldots, T$.

Before we turn to the computation of the test, note that an important advantage of PANIC is that the idiosyncratic components can be tested for stationarity regardless of the (non-)stationarity of the common factors. In other words, tests on common factors are asymptotically independent of tests on idiosyncratic components.

PANIC Unit Root Test Algorithm

The algorithm for computing the PANIC unit root test is quite involved and we encourage those interested to seek out more expansive treatments than that given here. Briefly, however, the algorithm involves three parts: computing the factor and idiosyncratic components, testing for a unit root in the idiosyncratic components, testing for a unit root in the common factors.

Computing test data

The first step in PANIC is computing estimates of the common factors $\hat{F}$ and idiosyncratic components $\hat{\epsilon}$:

- Arbitrarily specify the number r of common factors, or use y and $Q = y'y$ to determine r, the number of factors to use ("Determining the Number of Components" on page 650).

- Compute the principal components decomposition of the $Q = y'y$ matrix, producing the $(T-1) \times \min(N, T-1)$ matrix of eigenvectors H.
- Using H, form a matrix of r differenced common factors $\hat{f}$ and run a regression of y on $\hat{f}$ to obtain the factor loadings matrix $\hat{\Lambda}$.
- Compute estimates of the common factors $\hat{F}$ and idiosyncratic components $\hat{\epsilon}$ by taking cumulative sums of terms involving $\hat{f}$ and $\hat{\Lambda}$.

Testing idiosyncratic components

Next, PANIC tests for a unit root in the idiosyncratic components:

- Compute an ADF regression with p lags on the $\hat{\epsilon}$ for each cross-section and test for a unit root in the idiosyncratic component.

If we are unable to reject the null hypothesis for a given cross-section, we conclude that there is a unit root in the data for that cross section.

Recall too that early panel unit root tests conducted pooled tests to addressed the low power of univariate unit root tests in face of small T. Similarly, we may compute a pooled test statistic based on Maddala and Wu (1999) and Choi (2001),

$$P_N^e = \frac{2\sum_{i=1}^{N} \log(p_i^e) - 2N}{\sqrt{4N}} \rightarrow N(0, 1) \tag{42.82}$$

where p_i^e is the p-value associated with the idiosyncratic component ADF test for the i-th cross-section.

Observe that although the individual ADF test statistics have a non-standard distribution characterized by functionals of Brownian motions, under cross-sectional independence, the pooled test converges to the standard Gaussian distribution. Furthermore, note that the null hypothesis of this test, that all cross-sections have a unit root, holds only if no stationary combination of the Y_i exists. As such, the pooled test is also, in fact, a panel test for no cointegration.

Bear in mind, however, that the pooled version of the idiosyncratic ADF test does require cross-sectional independence, even though the overall PANIC procedure allows for slight correlation between the idiosyncratic components.

Testing common factors

Lastly, PANIC tests for unit roots in the common factors:

- If there is only one common factor ($r = 1$), PANIC computes an ADF regression with p lags and tests for a unit root in the common factor.

- In the multivariate common factor case ($r > 1$) perform an iterative testing procedure for determining the number of common trends. This procedure is reminiscent of the Johansen (1991) trace test for cointegrating rank. Bai and Ng consider two test statistics:
  1. The first, denoted as MQ_c, is based on a VAR(1) representation and corrects for serial correlation of arbitrary form through a non-parametric estimation.
  2. The second, denoted as MQ_f, is based on a VAR(p) representation, and corrects for the presence of serial correlation through a filter procedure.

  A brief description of the MQ statistics is warranted here. In particular, these statistics are modified versions of similarly named statistics in Stock and Watson (1988) that were designed to test the null hypothesis that s common factors have at most s common stochastic trends, against the alternative that they have less than s common trends. In this regard, the MQ_f test adapts the Dickey and Fuller (1979) unit root test to the multivariate settings. On the other hand, the MQ_c parallels in the multivariate the Phillips (1987) test.

  In practice, for each MQ statistic, the testing procedure first computes the relevant MQ for $m = r$ and tests whether the number of common stochastic trends is equal to m. If the null is not rejected, we conclude that all $m = r$ common factors are nonstationary. If the null is rejected, we decrement m by 1 and repeat until we fail to reject or until $m = 1$.

  If fail to reject the null at some $m = r_1$, we conclude that among the r common factors, r_1 are nonstationary, and $r_0 = r - r_1$ are stationary.

  If we reject the null at $m = 1$, we conclude that all r common factors are stationary.

CIPS Test

It is widely acknowledged that Bai and Ng (2004) is among the most general frameworks for panel unit root testing. In this regard, several specializations of PANIC exist and among the most popular and accessible is the Pesaran (2007) test (Cross-sectionally Augmented IPS).

The framework is again the factor augmented integrated panel,

$$\begin{aligned} Y_{it} &= D_{it} + \lambda_i' F_t + \epsilon_{it} \\ (1-L)F_t &= C(L)\eta_t \end{aligned} \tag{42.83}$$

but with the cross-sectional dependency restricted to a single *stationary* common factor ($r = 1$ and F_t is $I(0)$).

This model reduces to a simple heterogeneous dynamic linear model. When the deterministics include a trend and intercept we have

$$\begin{aligned} Y_{it} &= \mu_i + \delta_i(1-\psi_i)t + \psi_i Y_{it-1} + u_{it} \\ u_{it} &= \lambda_i F_t + \epsilon_{it} \end{aligned} \tag{42.84}$$

where ϵ_{it} are mean zero, stationary ARMA processes that are cross-sectionally and intra-sectionally independent. In addition to being $I(0)$, F_t is assumed to be serially uncorrelated with mean zero and constant variance, and ϵ_{it}, λ_t, and F_t are mutually independent for all i and t.

We may rewrite Equation (42.84) in Augmented Dickey-Fuller form as

$$\Delta Y_{it} = \mu_i + \gamma_i t - (1-\psi_i) Y_{it-1} + \sum_{k=1}^{p} \phi_{ik} \Delta Y_{it-k} + \lambda_i F_t + \epsilon_{it} \tag{42.85}$$

where $\gamma_i = \delta_i(1-\psi_i)$, the p lagged differences are added to account for correlation in the ϵ_{it}, and $\lambda_i F_t$ represents factor augmentation.

It is important to highlight the fact that non-stationarity arises in this model only when the $\rho_i = -(1-\psi_i)$ autoregressive coefficient is equal to 0. Accordingly the null hypothesis for the unit root test is

$$H_0\colon\ \psi_i = 1 \text{ for all } i,$$

against possibly heterogeneous alternatives,

$$H_A\colon\ \psi_i < 1 \text{ for } i \le N_1 \text{ and } \psi_i = 1 \text{ for } i > N_1$$

such that $\lim_{N\to\infty} N_1/N = \kappa$ for $0 < \kappa \le 1$ and $1 \le N_1 \le N$ is a positive integer, where, without loss of generality, we reorder the cross-sections to facilitate partitioning by index i.

For N sufficiently large, Pesaran (2007) argues that $\{\overline{Y}_t, \overline{Y}_{t-1}, \ldots\}$ where

$$\overline{Y}_t = (1/N) \sum_{s=1}^{N} Y_{st} \tag{42.86}$$

are proxies for the common factor F_t. It follows that $\{\overline{Y}_t\}_{t=1}$ and $\{\Delta\overline{Y}_t\}_{t=1}$ may be used to filter out the effects of the unobserved common factor.

Using these proxies we obtain the cross-sectionally augmented ADF (CADF) equation:

$$\begin{aligned} \Delta Y_{it} &= \mu_i + \gamma_i t + \rho_i Y_{it-1} + \sum_{k=1}^{p} \phi_{ik} \Delta Y_{it-k} \\ &+ \alpha_i \overline{Y}_{it-1} + \sum_{k=1}^{p} \beta_{ik} \Delta \overline{Y}_{it-k} + \epsilon_{it} \end{aligned} \tag{42.87}$$

Let τ_i^{CADF} denote the t-statistic associated with the traditional ADF null hypothesis $H_0\colon\ \rho_i = 0$ for cross-section i. Following Im, Pesaran, and Shin (2003), the panel unit root

test of interest is a pooled version of individual CADF statistics, or the cross-sectionally augmented (CIPS) statistic:

$$\tau^{CIPS} = \frac{1}{N}\sum_{i=1}^{N}\tau_i^{CADF} \tag{42.88}$$

A truncated version of this test is proposed to counter the influence of extreme outcomes that may arise when T is sufficiently small, say $10 \le T \le 20$. Define the constants $K_1 = -6.19$ and $K_2 = 2.61$ obtained by simulation in Pesaran (2007), and use these values to define a truncated version of τ^{CIPS},

$$\tau_i^{CADF*} = \begin{cases} K_1 & \text{if } \tau^{CADF} < K_1 \\ \tau^{CADF} & \text{if } K_1 \le \tau^{CADF} < K_2 \\ K_2 & \text{if } \tau^{CADF} \ge K_2 \end{cases} \tag{42.89}$$

$$\tau^{CIPS*} = \frac{1}{N}\sum_{i=1}^{N}\tau_i^{CADF*}$$

Neither CIPS nor truncated CIPS tests have standard limiting distributions. Critical values for popular scenarios have been derived via simulation and tabulated in Pesaran (2007).

Variance Ratio Test

The question of whether asset prices are predictable has long been the subject of considerable interest. One popular approach to answering this question, the Lo and MacKinlay (1988, 1989) overlapping variance ratio test, examines the predictability of time series data by comparing variances of differences of the data (returns) calculated over different intervals. If we assume the data follow a random walk, the variance of a q-period difference should be q times the variance of the one-period difference. Evaluating the empirical evidence for or against this restriction is the basis of the variance ratio test.

EViews allows you to perform the Lo and MacKinlay variance ratio test for homoskedastic and heteroskedastic random walks, using the asymptotic normal distribution (Lo and MacKinlay, 1988) or wild bootstrap (Kim, 2006) to evaluate statistical significance. In addition, you may compute the rank, rank-score, or sign-based forms of the test (Wright, 2000), with bootstrap evaluation of significance. In addition, EViews offers Wald and multiple comparison variance ratio tests (Richardson and Smith, 1991; Chow and Denning, 1993), so you may perform joint tests of the variance ratio restriction for several intervals.

Performing Variance Ratio Tests in EViews

First, open the series which contains the data which you wish to test and click on **View/ Variance Ratio Test…** Note that EViews allows you to perform the test using the differences, log differences, or original data in your series as the random walk innovations.

The **Output** dropdown determines whether you wish to see your test output in **Table** or **Graph** form. (As we discuss below, the choices differ slightly in a panel workfile.)

The **Data specification** section describes the properties of the data in the series. By default, EViews assumes you wish to test whether the data in the series follow a **Random walk**, so that variances are computed for differences of the data. Alternately, you may assume that the data follow an **Exponential random walk** so that the innovations are obtained by taking log differences, or that the series contains the **Random walk innovations** themselves.

The **Test specification** section describes the method used to compute your test. By default, EViews computes the basic Lo and MacKinlay variance ratio statistic assuming heteroskedastic increments to the random walk. The default calculations also allow for a non-zero innovation mean and bias correct the variance estimates.

The **Compute using** dropdown, which defaults to **Original data**, instructs EViews to use the original Lo and MacKinlay test statistic based on the innovations obtained from the original data. You may instead use the **Compute using** dropdown to instruct EViews to perform the variance ratio test using **Ranks**, **Rank scores** (van der Waerden scores), or **Signs** of the data.

For the Lo and MacKinlay test statistic, the three checkboxes directly beneath the dropdown allow you to choose whether to bias-correct the variance estimates, to construct the test using the heteroskedasticity robust test standard error, and to allow for non-zero means in the innovations. The **Probabilities** dropdown may be used to select between computing the test probabilities using the default **Asymptotic normal** results (Lo and MacKinlay 1988), or using the **Wild bootstrap** (Kim 2006). If you choose to perform a wild bootstrap, the

Options portion on the lower right of the dialog will prompt you to choose a bootstrap error distribution (**Two-point**, **Rademacher**, **Normal**), number of replications, random number generator, and to specify an optional random number generator seed.

For variance ratio test computed using **Ranks**, **Rank scores** (van der Waerden scores), or **Signs** of the data, the probabilities will be computed by permutation bootstrapping using the settings specified under **Options**. For the ranks and rank scores tests, there is an additional **Tie handling** option for the method of assigning ranks in the presence of tied data.

Lastly, the **Test periods** section identifies the intervals whose variances you wish to compare to the variance of the one-period innovations. You may specify a single period or more than one period; if there is more than one period, EViews will perform one ore more joint tests of the variance ratio restrictions for the specified periods.

There are two ways to specify the periods to test. First, you may provide a user-specified list of values or name of a vector containing the values. The default settings, depicted above, are to compute the test for periods “2 4 8 16.” Alternately, you may click on the **Equal-spaced grid** radio, and enter a minimum, maximum, and step.

If you are performing your test on a series in a panel workfile, the **Output** options differ slightly. If you wish to produce output in tabular form, you can choose to compute individual variance ratio tests for each cross-section and form a Fisher Combined test (**Table - Fisher Combined**), or you can choose to stack the cross-sections into a single series and perform the test on the stacked panel (**Table - Stacked Panel**). Note that the stacked panel method assumes that all means and variances are the same across all cross-sections; the only adjustment for the panel structure is in data handling that insures that lags never cross the seams between cross-sections. There are two graphical counterparts to the table choices: **Graph - Individual**, which produces a graph for each cross-section, and **Graph - Stacked Panel**, which produces a graph of the results for the stacked analysis.

An Example

In our example, we employ the time series data on nominal exchange rates used by Wright (2000) to illustrate his modified variance ratio tests ("Wright.WF1"). The data in the first page (WRIGHT) of the workfile provide the relative-to-U.S. exchange rates for the Canadian dollar, French franc, German mark, Japanese yen, and the British pound for the 1,139 weeks from August 1974 through May 1996. Of interest is whether the exchange rate returns, as measured by the log differences of the rates, are *i.i.d.* or martingale difference, or alternately, whether the exchange rates themselves follow an exponential random walk.

We begin by performing tests on the Japanese yen. Open the JP series, then select **View/Variance Ratio**... to display the dialog. We will make a few changes to the default settings to match Wright's calculations. First, select **Exponential random walk** in the **Data specification** section to tell EViews that you wish to work with the log returns. Next, uncheck the **Use unbiased variances** and **Use heteroskedastic robust S.E.** checkboxes to perform the *i.i.d.* version of the Lo-MacKinlay test with no bias correction. Lastly, change the user-specified test periods to "2 5 10 30" to match the test periods examined by Wright. Click on **OK** to compute and display the results.

The top portion of the output shows the test settings and basic test results.

Null Hypothesis: Log JP is a random walk
Date: 04/21/09 Time: 15:15
Sample: 8/07/1974 5/29/1996
Included observations: 1138 (after adjustments)
Standard error estimates assume no heteroskedasticity
Use biased variance estimates
User-specified lags: 2 5 10 30

Joint Tests	Value	df	Probability
Max |z| (at period 5)*	4.295371	1138	0.0001
Wald (Chi-Square)	22.63414	4	0.0001

Individual Tests

Period	Var. Ratio	Std. Error	z-Statistic	Probability
2	1.056126	0.029643	1.893376	0.0583
5	1.278965	0.064946	4.295371	0.0000
10	1.395415	0.100088	3.950676	0.0001
30	1.576815	0.182788	3.155651	0.0016

*Probability approximation using studentized maximum modulus with parameter value 4 and infinite degrees of freedom

Since we have specified more than one test period, there are two sets of test results. The "Joint Tests" are the tests of the joint null hypothesis for all periods, while the "Individual Tests" are the variance ratio tests applied to individual periods. Here, the Chow-Denning maximum $|z|$ statistic of 4.295 is associated with the period 5 individual test. The approximate p-value of 0.0001 is obtained using the studentized maximum modulus with infinite degrees of freedom so that we strongly reject the null of a random walk. The results are quite similar for the Wald test statistic for the joint hypotheses. The individual statistics generally reject the null hypothesis, though the period 2 variance ratio statistic p-value is slightly greater than 0.05.

The bottom portion of the output shows the intermediate results for the variance ratio test calculations, including the estimated mean, individual variances, and number of observations used in each calculation.

Test Details (Mean = -0.000892835617901)

Period	Variance	Var. Ratio	Obs.
1	0.00021	--	1138
2	0.00022	1.05613	1137
5	0.00027	1.27897	1134
10	0.00029	1.39541	1129
30	0.00033	1.57682	1109

Alternately, we may display a graph of the test statistics using the same settings. Simply click again on **View/Variance Ratio Test...**, change the **Output** dropdown from **Table** to **Graph**, then fill out the dialog as before and click on **OK**:

EViews displays a graph of the variance ratio statistics and plus or minus two asymptotic standard error bands, along with a horizontal reference line at 1 representing the null hypothesis. Here, we see a graphical representation of the fact that with the exception of the test against period 2, the null reference line lies outside the bands.

Next, we repeat the previous analysis but allow for heteroskedasticity in the data and use bootstrapping to evaluate the statistical significance. Fill out the dialog as before, but enable the **Use heteroskedastic robust S.E.** checkbox and use the **Probabilities** dropdown to select **Wild bootstrap** (with the two-point distribution, 5000 replications, the Knuth generator, and a seed for the random number generator of 1000 specified in the **Options** section). The top portion of the results is depicted here:

Null Hypothesis: Log JP is a martingale
Date: 04/21/09 Time: 15:15
Sample: 8/07/1974 5/29/1996
Included observations: 1138 (after adjustments)
Heteroskedasticity robust standard error estimates
Use biased variance estimates
User-specified lags: 2 5 10 30
Test probabilities computed using wild bootstrap: dist=twopoint,
reps=5000, rng=kn, seed=1000

Joint Tests	Value	df	Probability
Max |z| (at period 5)	3.646683	1138	0.0012

Individual Tests

Period	Var. Ratio	Std. Error	z-Statistic	Probability
2	1.056126	0.037086	1.513412	0.1316
5	1.278965	0.076498	3.646683	0.0004
10	1.395415	0.115533	3.422512	0.0010
30	1.576815	0.205582	2.805766	0.0058

Note that the Wald test is no longer displayed since the test methodology is not consistent with the use of heteroskedastic robust standard errors in the individual tests. The p-values for the individual variance ratio tests, which are all generated using the wild bootstrap, are generally consistent with the previous results, albeit with probabilities that are slightly higher than before. The individual period 2 test, which was borderline (in)significant in the

homoskedastic test, is no longer significant at conventional levels. The Chow-Denning joint test statistic of 3.647 has a bootstrap p-value of 0.0012 and strongly rejects the null hypothesis that the log of JP is a martingale.

Lastly, we perform Wright's rank variance ratio test with ties replaced by the average of the tied ranks. The test probabilities for this test are computed using the permutation bootstrap, whose settings we select to match those for the previous bootstrap:

Null Hypothesis: Log JP is a random walk
Date: 04/21/09 Time: 15:16
Sample: 8/07/1974 5/29/1996
Included observations: 1138 (after adjustments)
Standard error estimates assume no heteroskedasticity
User-specified lags: 2 5 10 30
Test probabilities computed using permutation bootstrap: reps=5000, rng=kn, seed=1000

Joint Tests	Value	df	Probability
Max |z| (at period 5)	5.415582	1138	0.0000
Wald (Chi-Square)	37.92402	4	0.0000

Individual Tests

Period	Var. Ratio	Std. Error	z-Statistic	Probability
2	1.081907	0.029643	2.763085	0.0050
5	1.351718	0.064946	5.415582	0.0000
10	1.466929	0.100088	4.665193	0.0000
30	1.790412	0.182788	4.324203	0.0000

The standard errors employed in forming the individual z-statistics (and those displayed in the corresponding graph view) are obtained from the asymptotic normal results. The probabilities for the individual z-statistics and the joint max $|z|$ and Wald statistics, which all strongly reject the null hypothesis, are obtained from the permutation bootstrap.

The preceding analysis may be extended to tests that jointly consider all five exchange rates in a panel setting. The second page (WRIGHT_STK) of the "Wright.WF1" workfile contains the panel dataset of the relative-to-U.S. exchange rates described above (Canada, Germany, France, Japan, U.K.). Click on the WRIGHT_STK tab to make the second page active, double click on the EXCHANGE series to open the stacked exchange rates series, then select **View/ Variance Ratio Test...**

We will redo the heterogeneous Lo and MacKinlay test example from above using the panel data series. Select **Table - Fisher Combined** in the **Output** dropdown then fill out the remainder of the dialog as before, then click on **OK**. The output, which takes a moment to generate since we are performing 5000 bootstrap replications for each cross-section, consists of two distinct parts. The top portion of the output:

Null Hypothesis: Log EXCHANGE is a martingale
Date: 04/21/09 Time: 15:18
Sample: 8/07/1974 5/29/1996
Cross-sections included: 5
Total panel observations: 5690 (after adjustments)
Heteroskedasticity robust standard error estimates
Use biased variance estimates
User-specified lags: 2 5 10 30
Test probabilities computed using wild bootstrap:
dist=Two-point, reps=5000, rng=kn, seed=1000

Summary Statistics

Statistics	Max |z|	Prob.	df
Fisher Combined	28.252	0.0016	10

shows the test settings and provides the joint Fisher combined test statistic which, in this case, strongly rejects the joint null hypothesis that all of the cross-sections are martingales.

The bottom portion of the output:

Cross-section Joint Tests

Cross-section	Max |z|	Prob.	Obs.
CAN	2.0413	0.0952	1138
DEU	1.7230	0.1952	1138
FRA	2.0825	0.0946	1138
JP	3.6467	0.0016	1138
UK	1.5670	0.2606	1138

depicts the max $|z|$ statistics for the individual cross-sections, along with corresponding wild bootstrap probabilities. Note that four of the five individual test statistics do not reject the joint hypothesis at conventional levels. It would therefore appear that the Japanese yen result is the driving force behind the Fisher combined test rejection.

Technical Details

Suppose we have the time series $\{Y_t\} = (Y_0, Y_1, Y_2, \ldots, Y_T)$ satisfying

$$\Delta Y_t = \mu + \epsilon_t \tag{42.90}$$

where μ is an arbitrary drift parameter. The key properties of a random walk that we would like to test are $E(\epsilon_t) = 0$ for all t and $E(\epsilon_t \epsilon_{t-j}) = 0$ for any positive j.

The Basic Test Statistic

Lo and MacKinlay (1988) formulate two test statistics for the random walk properties that are applicable under different sets of null hypothesis assumptions about ϵ_t:

First, Lo and MacKinlay make the strong assumption that the ϵ_t are i.i.d. Gaussian with variance σ^2 (though the normality assumption is not strictly necessary). Lo and MacKinlay term this the homoskedastic random walk hypothesis, though others refer to this as the *i.i.d.* null.

Alternately, Lo and MacKinlay outline a heteroskedastic random walk hypothesis where they weaken the *i.i.d.* assumption and allow for fairly general forms of conditional heteroskedasticity and dependence. This hypothesis is sometimes termed the martingale null, since it offers a set of sufficient (but not necessary), conditions for ϵ_t to be a martingale difference sequence (*m.d.s.*).

We may define estimators for the mean of first difference and the scaled variance of the q-th difference:

$$\begin{aligned} \hat{\mu} &= \frac{1}{T}\sum_{t=1}^{T}(Y_t - Y_{t-1}) \\ \hat{\sigma}^2(q) &= \frac{1}{Tq}\sum_{t=1}^{T}(Y_t - Y_{t-q} - q\hat{\mu})^2 \end{aligned} \tag{42.91}$$

and the corresponding variance ratio $VR(q) = \hat{\sigma}^2(q)/\hat{\sigma}^2(1)$. The variance estimators may be adjusted for bias, as suggested by Lo and MacKinlay, by replacing T in Equation (42.91) with $(T - q + 1)$ in the no-drift case, or with $(T - q + 1)(1 - q/T)$ in the drift case.

Lo and MacKinlay show that the variance ratio z-statistic:

$$z(q) = (VR(q) - 1) \cdot [\mathfrak{s}^2(q)]^{-1/2} \tag{42.92}$$

is asymptotically $N(0, 1)$ for appropriate choice of estimator $\mathfrak{s}^2(q)$.

Under the *i.i.d.* hypothesis we have the estimator,

$$\mathfrak{s}^2(q) = \frac{2(2q-1)(q-1)}{3qT} \tag{42.93}$$

while under the *m.d.s.* assumption we may use the kernel estimator,

$$\mathfrak{s}^2(q) = \sum_{j=1}^{q-1}\left(\frac{2(q-j)}{q}\right)^2 \cdot \hat{\delta}_j \tag{42.94}$$

where

$$\hat{\delta}_j = \left\{\sum_{t=j+1}^{T}(y_{t-j} - \hat{\mu})^2(y_t - \hat{\mu})^2\right\} \Big/ \left\{\sum_{t=j+1}^{T}(y_{t-j} - \hat{\mu})^2\right\}^2 \tag{42.95}$$

Joint Variance Ratio Tests

Since the variance ratio restriction holds for every difference $q > 1$, it is common to evaluate the statistic at several selected values of q.

To control the size of the joint test, Chow and Denning (1993) propose a (conservative) test statistic that examines the maximum absolute value of a set of multiple variance ratio statistics. The p-value for the Chow-Denning statistic using m variance ratio statistics is bounded from above by the probability for the Studentized Maximum Modulus (SMM) distribution with parameter m and T degrees-of-freedom. Following Chow and Denning, we approximate this bound using the asymptotic ($T = \infty$) SMM distribution.

An second approach is available for variance ratio tests of the *i.i.d.* null. Under this set of assumptions, we may form the joint covariance matrix of the variance ratio test statistics as in Richardson and Smith (1991), and compute the standard Wald statistic for the joint hypothesis that all m variance ratio statistics equal 1. Under the null, the Wald statistic is asymptotic Chi-square with m degrees-of-freedom.

For a detailed discussion of these tests, see Fong, Koh, and Ouliaris (1997).

Wild Bootstrap

Kim (2006) offers a wild bootstrap approach to improving the small sample properties of variance ratio tests. The approach involves computing the individual (Lo and MacKinlay) and joint (Chow and Denning, Wald) variance ratio test statistics on samples of T observations formed by weighting the original data by mean 0 and variance 1 random variables, and using the results to form bootstrap distributions of the test statistics. The bootstrap p-values are computed directly from the fraction of replications falling outside the bounds defined by the estimated statistic.

EViews offers three distributions for constructing wild bootstrap weights: the two-point, the Rademacher, and the normal. Kim's simulations indicate that the test results are generally insensitive to the choice of wild bootstrap distribution.

Rank and Rank Score Tests

Wright (2000) proposes modifying the usual variance ratio tests using standardized ranks of the increments, ΔY_t. Letting $r(\Delta Y_t)$ be the rank of the ΔY_t among all T values, we define the standardized rank (r_{1t}) and van der Waerden rank scores (r_{2t}):

$$\begin{aligned} r_{1t} &= \left(r(\Delta Y_t) - \frac{T+1}{2}\right) / \sqrt{\frac{(T-1)(T+1)}{12}} \\ r_{2t} &= \Phi^{-1}(r(\Delta Y_t)/(T+1)) \end{aligned} \tag{42.96}$$

In cases where there are tied ranks, the denominator in r_{1t} may be modified slightly to account for the tie handling.

The Wright variance ratio test statistics are obtained by computing the Lo and MacKinlay homoskedastic test statistic using the ranks or rank scores in place of the original data. Under the *i.i.d.* null hypothesis, the exact sampling distribution of the statistics may be approximated using a permutation bootstrap.

Sign Test

Wright also proposes a modification of the homoskedastic Lo and MacKinlay statistic in which each ΔY_t is replaced by its sign. This statistic is valid under the *m.d.s.* null hypothesis, and under the assumption that $\mu = 0$, the exact sampling distribution may also be approximated using a permutation bootstrap. (EViews does not allow for non-zero means when performing the sign test since allowing $\mu \neq 0$ introduces a nuisance parameter into the sampling distribution.)

Panel Statistics

EViews offers two approaches to variance ratio testing in panel settings.

First, under the assumption that cross-sections are independent, with cross-section heterogeneity of the processes, we may compute separate joint variance ratio tests for each cross-section, then combine the p-values from cross-section results using the Fisher approach as in Maddala and Wu (1999). If we define π_i to be a p-value from the i-th cross-section, then under the hypothesis that the null hypothesis holds for all N cross-sections,

$$-2 \sum_{i=1}^{N} \log(\pi_i) \rightarrow \chi^2_{2N} \tag{42.97}$$

as $T \rightarrow \infty$.

Alternately, if we assume homogeneity across all cross-sections, we may stack the panel observations and compute the variance ratio test for the stacked data. In this approach, the only adjustment for the panel nature of the stacked data is in ensuring that lag calculations do not span cross-section boundaries.

BDS Independence Test

This series view carries out the BDS test for independence as described in Brock, Dechert, Scheinkman and LeBaron (1996).

The BDS test is a portmanteau test for time based dependence in a series. It can be used for testing against a variety of possible deviations from independence including linear dependence, non-linear dependence, or chaos.

The test can be applied to a series of estimated residuals to check whether the residuals are independent and identically distributed (*iid*). For example, the residuals from an ARMA

model can be tested to see if there is any non-linear dependence in the series after the linear ARMA model has been fitted.

The idea behind the test is fairly simple. To perform the test, we first choose a distance, ϵ. We then consider a pair of points. If the observations of the series truly are *iid*, then for any pair of points, the probability of the distance between these points being less than or equal to epsilon will be constant. We denote this probability by $c_1(\epsilon)$.

We can also consider sets consisting of multiple pairs of points. One way we can choose sets of pairs is to move through the consecutive observations of the sample in order. That is, given an observation s, and an observation t of a series X, we can construct a set of pairs of the form:

$$\{\{X_s, X_t\}, \{X_{s+1}, X_{t+1}\}, \{X_{s+2}, X_{t+2}\}, \ldots, \{X_{s+m-1}, X_{t+m-1}\}\} \tag{42.98}$$

where m is the number of consecutive points used in the set, or *embedding dimension*. We denote the joint probability of every pair of points in the set satisfying the epsilon condition by the probability $c_m(\epsilon)$.

The BDS test proceeds by noting that under the assumption of independence, this probability will simply be the product of the individual probabilities for each pair. That is, if the observations are independent,

$$c_m(\epsilon) = c_1^{\,m}(\epsilon). \tag{42.99}$$

When working with sample data, we do not directly observe $c_1(\epsilon)$ or $c_m(\epsilon)$. We can only estimate them from the sample. As a result, we do not expect this relationship to hold exactly, but only with some error. The larger the error, the less likely it is that the error is caused by random sample variation. The BDS test provides a formal basis for judging the size of this error.

To estimate the probability for a particular dimension, we simply go through all the possible sets of that length that can be drawn from the sample and count the number of sets which satisfy the ϵ condition. The ratio of the number of sets satisfying the condition divided by the total number of sets provides the estimate of the probability. Given a sample of n observations of a series X, we can state this condition in mathematical notation,

$$c_{m,\,n}(\epsilon) = \frac{2}{(n-m+1)(n-m)} \sum_{s=1}^{n-m+1} \sum_{t=s+1}^{n-m+1} \prod_{j=0}^{m-1} I_\epsilon(X_{s+j}, X_{t+j}) \tag{42.100}$$

where I_ϵ is the indicator function:

$$I_\epsilon(x, y) = \begin{cases} 1 & \text{if } |x-y| \le \epsilon \\ 0 & \text{otherwise.} \end{cases} \tag{42.101}$$

Note that the statistics $c_{m,\,n}$ are often referred to as *correlation integrals*.

We can then use these sample estimates of the probabilities to construct a test statistic for independence:

$$b_{m,\,n}(\epsilon) \;=\; c_{m,\,n}(\epsilon) - c_{1,\,n-m+1}(\epsilon)^{m} \tag{42.102}$$

where the second term discards the last $m-1$ observations from the sample so that it is based on the same number of terms as the first statistic.

Under the assumption of independence, we would expect this statistic to be close to zero. In fact, it is shown in Brock *et al.* (1996) that

$$(\sqrt{n-m+1})\frac{b_{m,\,n}(\epsilon)}{\sigma_{m,\,n}(\epsilon)} \rightarrow N(0,\,1) \tag{42.103}$$

where

$$\sigma^{2}_{m,\,n}(\epsilon) \;=\; 4\left(k^{m} + 2\sum_{j=1}^{m-1} k^{m-j} c_1^{\,2j} + (m-1)^{2} c_1^{\,2m} - m^{2} k c_1^{\,2m-2}\right) \tag{42.104}$$

and where c_1 can be estimated using $c_{1,\,n}$. k is the probability of any triplet of points lying within ϵ of each other, and is estimated by counting the number of sets satisfying the sample condition:

$$k_n(\epsilon) \;=\; \frac{2}{n(n-1)(n-2)} \sum_{t=1}^{n} \sum_{s=t+1}^{n} \sum_{r=s+1}^{n} \tag{42.105}$$

$$(I_\epsilon(X_t,\,X_s)I_\epsilon(X_s,\,X_r) + I_\epsilon(X_t,\,X_r)I_\epsilon(X_r,\,X_s) + I_\epsilon(X_s,\,X_t)I_\epsilon(X_t,\,X_r))$$

To calculate the BDS test statistic in EViews, simply open the series you would like to test in a window, and choose **View/BDS Independence Test...**. A dialog will appear prompting you to input options.

To carry out the test, we must choose ϵ, the distance used for testing proximity of the data points, and the dimension m, the number of consecutive data points to include in the set.

The dialog provides several choices for how to specify ϵ :

- **Fraction of pairs**: ϵ is calculated so as to ensure a certain fraction of the total number of pairs of points in the sample lie within ϵ of each other.
- **Fixed value**: ϵ is fixed at a raw value specified in the units as the data series.
- **Standard deviations**: ϵ is calculated as a multiple of the standard deviation of the series.
- **Fraction of range**: ϵ is calculated as a fraction of the range (the difference between the maximum and minimum value) of the series.

The default is to specify ϵ as a fraction of pairs, since this method is most invariant to different distributions of the underlying series.

You must also specify the value used in calculating ϵ . The meaning of this value varies based on the choice of method. The default value of 0.7 provides a good starting point for the default method when testing shorter dimensions. For testing longer dimensions, you should generally increase the value of ϵ to improve the power of the test.

EViews also allows you to specify the maximum correlation dimension for which to calculate the test statistic. EViews will calculate the BDS test statistic for all dimensions from 2 to the specified value, using the same value of ϵ or each dimension. Note the same ϵ is used only because of calculational efficiency. It may be better to vary ϵ with the correlation dimension to maximize the power of the test.

In small samples or in series that have unusual distributions, the distribution of the BDS test statistic can be quite different from the asymptotic normal distribution. To compensate for this, EViews offers you the option of calculating bootstrapped p-values for the test statistic. To request bootstrapped p-values, simply check the **Use bootstrap** box, then specify the number of repetitions in the field below. A greater number of repetitions will provide a more accurate estimate of the p-values, but the procedure will take longer to perform.

When bootstrapped p-values are requested, EViews first calculates the test statistic for the data in the order in which it appears in the sample. EViews then carries out a set of repetitions where for each repetition a set of observations is randomly drawn with replacement from the original data. Also note that the set of observations will be of the same size as the original data. For each repetition, EViews recalculates the BDS test statistic for the randomly drawn data, then compares the statistic to that obtained from the original data. When all the repetitions are complete, EViews forms the final estimate of the bootstrapped p-value by dividing the lesser of the number of repetitions above or below the original statistic by the total number of repetitions, then multiplying by two (to account for the two tails).

As an example of a series where the BDS statistic will reject independence, consider a series generated by the non-linear moving average model:

$$y_t = u_t + 8 u_{t-1} u_{t-2} \tag{42.106}$$

where u_t is a normal random variable. On simulated data, the correlogram of this series shows no statistically significant correlations, yet the BDS test strongly rejects the hypothesis that the observations of the series are independent (note that the Q-statistics on the squared levels of the series also reject independence).

Wavelet Analysis

Many economic time series feature time-varying characteristics such as non-stationarity, volatility, seasonality, and structural discontinuities. Wavelet analysis is a natural framework for analyzing these phenomena without imposing simplifying assumptions such as stationarity. In particular, wavelet filters can decompose and reconstruct a time series, as well as its correlation structure, across time scales.

While wavelets have applications in diverse areas such as such as regression, unit root testing, bootstrapping (wavestrapping), and fractional integration order estimation, EViews focuses on four popular areas of application:

- Wavelet transforms
- Variance decomposition
- Outlier detection
- Thresholding (denoising)

Additional discussion of wavelet analysis is provided in "Technical Background" on page 773.

Performing Wavelet Analysis in EViews

To perform wavelet analysis in EViews, open the series of interest and click on **View/Wavelet Analysis**. There are four sub-menu entries corresponding to the four areas of application. We consider each in turn.

Wavelet Transforms

Typically, the first step in wavelet analysis is to perform a wavelet transform of some input series (discrete wavelet transform analysis is described in "Discrete Wavelet Transform" on page 775).

Click on **View/Wavelet Analysis/Transforms...** to display the wavelet transform analysis dialog:

The first step is to choose a discrete wavelet decomposition type.

- The **Method** dropdown offers you a choice between the four supported methods:
  1. **Orthogonal transform - DWT** (orthogonal discrete wavelet transform)
  2. **Overlap transform - MODWT** (non-orthogonal maximum overlap discrete wavelet transform)
  3. **Orthogonal multiresolution - MRA** (multiresolution analysis using a DWT transform)
  4. **Overlap multires. - MODWT MRA** (multiresolution analysis using a MODWT transform)

Discussion is provided in "Discrete Wavelet Transform" on page 775 and "Multiresolution Analysis" on page 777.

Next you may specify scale options:

- If you select a method that employs DWT and the input series is not of dyadic (power of 2) length, the **Dyadic** dropdown will offer several options for adjusting the sample to derive the transform ("Adjusting Non-dyadic Length Time Series" on page 783). You may choose between **Pad with zeros**, **Pad with mean**, **Pad with median**, and **Shorten**

  The padding methods add the specified value to the end of the input series to achieve the next dyadic length. The **Shorten** method truncates the input series at the preceding dyadic length.

- The **Max scale** dropdown lets you choose the maximum wavelet scale up to which the transform will be computed. The default is the maximum possible based on sam-

ple size. While the DWT is limited by sample size, the MODWT is not. Accordingly, if the transform method is one of MODWT or MODWT MRA, the maximum wavelet scale is specied as an input value.

Lastly, you may use the **Filter** group settings to specify properties of the filter

- The **Class** dropdown specifies the choice of wavelet filter. You may select between **Haar**, **Daubechies**, and **Least asymmetric** filters.

  If you select either of the latter two filters, you will be prompted to provide a **Length**. The Daubechies class offers filter lengths running from 4 to 20 in increments of 2, whereas the symmlets offers filter lengths running from 8 to 20 in increments of 2

  For discussion, see "Wavelet Filters" on page 776 and "Choice of Wavelet Filter" on page 783.

- The **Boundary** dropdown allows you to choose the method for handling boundary conditions resulting from circular recycling of observations. You may choose **Periodic** and **Reflective**.

  The **Periodic** setting has the filter wrap around the end of the input series to handle endpoint missing observations. The **Reflective** setting concatenates the reverse of the input series to form an input of length $2T$.

  See "Boundary Conditions" on page 777 and "Handling Boundary Conditions" on page 783.

- By default, EViews will highlight boundary coefficients in the output. If you do not wish to highlight these values, uncheck the **Show boundary conditions** checkbox.

Example (Discrete Wavelet Transform)

As a first example, consider the Canadian real exchange rate data extracted from the data set in Pesaran (2007) in "wavelets.WF1". This is a quarterly series running from 1973q1 to 1998q4. We will use the default settings in the dialog. To display the transform, open the CANADA_RER series and select from the series window, click on **OK**.

Note that since the number of data points is 104, a dyadic adjustment is needed and the default setting pads the input series using the series mean is used to achieve dyadic length.

The output is a spool object with the spool tree listing the summary, original series, as well as wavelet and scaling coefficients for scales 1 through 7. The first of these is a summary of the wavelet transformation performed.

The second spool entry is a plot of the original series along with the padded values.

The remaining entries in the spool are wavelet and scaling coefficients plotted for each scale. Scroll down or click on an entry to bring it into view:

Here, we see the wavelet coefficients at scales 1 and 2. Notice that standard deviation of the coefficients at each plot has two dashed red lines. These represent the +/– standard deviation of coefficients at that scale. This is particularly useful in visualizing which wavelet coefficients should be shrunk to zero in wavelet shrinkage applications. Recall that coefficients exceeding some threshold bound (in this case the standard deviation) ought to be retained, whereas the remaining coefficients ought to be shrunk to zero.

Continuing to higher scales, we see much smaller relative values

Looking at the wavelet coefficients at each scale, it is clear that the majority of relevant information can be captured only in the first few scales and using very few coefficients. For instance, at scale 1 we see roughly 3 or 4 wavelet coefficients which exceed the threshold bounds. Those are the coefficients which will contribute most to explaining variation at that scale and will capture the most transitory (noise) features of the underlying series. At scale 2, there are 6 coefficients which exceed the threshold bounds. While these coefficients are in the lower half of the frequency range, they are not in the immediate vicinity of frequency 0. We can conclude that this is a series which has persistence, but is not necessarily non-stationary. A quick unit root test will also confirm this belief.

Finally, notice that the time-axis in the DWT is not actually aligned with time, but rather the number of observations. This offset occurs because the filters used are in fact not zero-phase filters and therefore their transformation will not be aligned with time.

Example (Maximal Overlap Discrete Wavelet Transform)

For the second example, use the input series from the first example and perform a MODWT with the Daubechies filter of length 6. From the series window, click on **View/Wavelet Analysis/Transforms...** to display the dialog.

Then in the **Method** dropdown select **Overlap transform - MODWT**. In the **Class** dropdown select **Daubechies** and in the **Length** dropdown select 6. Click on **OK**.

As before, the output is a spool object with wavelet and scaling coefficients as individual spool elements.

- The first spool output is a summary of the transformation procedure.
- The second displays the series transform.
- The remainder of the output are the wavelet and scaling coefficients.

Note that since the MODWT is not an orthonormal transform and since it uses all of the available observations, wavelet and scaling coefficients are of input series length and do not require length adjustments.

Scrolling down or clicking on **Wavelet 1** in the tree to bring the graph into view displays:

Next, notice that for each scale those coefficients affected by the boundary are displayed in red, and their count reported in the legends. Moreover, there is a dashed vertical line indicating the end of the boundary coefficients. Furthermore, since the MODWT is a redundant transform, the number of boundary coefficients standard deviation bounds will always be greater than those in the orthonormal DWT. As before the are available for reference. Finally, note that the time-axis in the MODWT is aligned with time. This is an advantage of the MODWT over the DWT at the expense of redundancy.

As in the DWT example, it's clear that the majority of relevant wavelet coefficients (those exceeding threshold bounds) are concentrated in the first few scales. We also observe that there are significantly more such coefficients in the MODWT transformation as opposed to the DWT since the MODWT is a redundant transformation and wavelet coefficients across scales are correlated. This behavior is also seen in the wave-like patterns in the output

Wavelet Variance Decomposition

Wavelet variance decomposition is described in "Wavelet Analysis of Variance" on page 778. To perform wavelet decomposition of variance in EViews, open a series window and proceed to **View/Wavelet Analysis/Variance Decomposition...** to display the dialog.

The **Variance** group settings control the computation of the variances and confidence intervals:

- The **Method** dropdown offers two variance decomposition choices ("Wavelet Analysis of Variance" on page 778):
  1. **Unbiased** – per-scale variance contributions using the unbiased variance form.
  2. **Biased** – per-scale variance contributions using the biased variance form.

When the variance decomposition is unbiased the maximum filter scale is adjusted automatically to reflect dropping boundary coefficients.

If the decomposition is unbiased, one can also form confidence intervals for per-scale variance contributions.

- The **CI Type** dropdown offers four choices:
  1. **None** – no confidence intervals are produced.
  2. **Asymp. Normal** – confidence intervals are produced using Gaussian critical values
  3. **Asymp. Chi-Squared** – confidence intervals are produced using Chi-squared critical values using equation and EDOF based on asymptotic estimation.
  4. **Asymp. Band-Limited** – confidence intervals are produced using Chi-squared critical values using equation and EDOF based on band-pass estimation.

  See "Variance Confidence Intervals" on page 778 for details.

- If a confidence interval will be computed, the **Levels edit** field specifies the confidence interval coverage probability.

Lastly, the **Decomposition** and **Filter** group elements were discussed in detail in "Wavelet Transforms" on page 753.

Example (Maximal Overlap Unbiased Variance Decomposition)

For this example, we use Japanese real exchange rate data from 1973Q1 to 1988Q4 extracted from the data set in Pesaran (2007). The objective is to produce a scale-by-scale decomposition of variance contributions using the MODWT with a Daubechies filter of length 4. Furthermore, we'll produce a 95%confidence intervals using the asymptotic Chi-squared distribution using a band-pass estimate for the EDOF. The band-pass EDOF is preferred here since the sample size is less than 128 and the asymptotic approximation to the EDOF requires a sample size of at least 128 for decent results.

To proceed, open the series window and click on **View/Wavelet Analysis/Variance Decomposition...** First, set the **CI type** dropdown to **Asymp. Band-Limited**. Next, set the dropdown in the **Decomposition** group to **Overlap transform - MODWT.** Lastly, change the **Filter** group **Class** dropdown to **Daubechies** and click on **OK** to display the results.

The output is a spool object with the spool tree listing the procedure summary, energy spectrum, variance distribution across-scales, CIs across scales, and the cumulative variance.

Scale	Variance	Rel. Proport.	Cum. Proport.	95% CI Lower	95% CI Upper
W1	0.000430	0.0425	0.0425	0.000298	0.000658
W2	0.000934	0.0923	0.1349	0.000552	0.001751
W3	0.002685	0.2654	0.4003	0.001240	0.007125
W4	0.006068	0.5997	1.0000	0.001940	0.044624

The first spool entry is a summary of the wavelet variance decomposition performed including information about the decomposition and filter, sample, and the confidence intervals settings.

The second entry is a table of the variance spectrum listing the contribution to overall variance by wavelet coefficients at each scale. The column titled “Variance” shows the variance contributed to the total at a given scale. Columns titled “Relative Proportion” and “Cumulative Proportion” display the proportion of overall variance contributing to the total at a given scale and its cumulative total, respectively.

Here, since confidence intervals have been produced, the last two columns display the lower and upper confidence interval values at each scale.

We may scroll down or click on the **Variance Distribution** selector to view the next items in the spool:

The next object in the spool is the **Variance Spectrum Distribution** graph, which is a bar plot of variances at each given scale.

Following this is the **Variance CI Distribution** graph, which shows the variance values along with their 95% confidence intervals at each scale.

Scrolling down to the last graph or clicking on **Cumulative Variance**, we see a display showing the cumulative variances across scales and, if computed, the cumulative confidence interval.

Wavelet Threshold (Denoising)

As in the previous wavelet examples, wavelet thresholding in EViews is performed on a series. In particular, from an open series window, click on **View/Wavelet Analysis/Thresholding (Denoising)...** to display the dialog.

The **Threshold** group options control the specification of the threshold:

- The **Method** dropdown allows you to choose between **Hard** and **Soft** threshold rules ("Thresholding Rule" on page 780).
- The **Limit** dropdown controls the method for determining the optimal threshold value. You may select between **Universal**, **Universal adaptive**, **MiniMax**, **SureShrink**, and **False Discovery Rate** ("Threshold Value Methods" on page 781)

The **Wavelet coefficient variance** group specifies the method for computing the estimate of the variance (standard deviation) of the wavelet coefficients

- The **Method** dropdown offers you several choices: **Mean Abs. Dev**, **Median (Gaussian)**, **Median Abs. Dev.**, and **Mean Median Abs. Dev.** ("Wavelet Coefficients Variance" on page 781)

The remaining **Decomposition** and **Filter** group elements were discussed in detail in "Wavelet Transforms" on page 753.

Example (Discrete Wavelet Hard Thresholding)

For our example, we use the French real exchange rate data from 1973q1 to 1988q4 extracted from the data set in Pesaran (2007). The objective is to de-noise the input function by extracting its signal using the DWT with a Daubechies filter of length 6. We will employ a hard thresholding rule using a false discovery rate threshold with significance level 05. The wavelet variances will be estimated using the median with a Guassian correction.

To proceed, open the series window and click on **View/Wavelet Analysis/Denoising (Thresholding)...**

Under the **Threshold** group, set the **Method** dropdown to **Hard** and set the **Limit** dropdown to **False Discovery Rate**. Next, Under the **Wavelet coefficient variance** group set the **Method** dropdown to **Median (Gaussian)**. Lastly, change the **Class** dropdown to **Daubechies** and set the **Length** dropdown to 6. Click on **OK**.

EViews displays spool output with a tree listing the summary, denoised function, and noise.

Finally, the remaining plot is the noise process extracted from the original series.

At the top of the spool is a summary of the denoising procedure describing the decomposition method and filter, sample information, threshold method, and wavelet coefficient variance computation.

Below the summary information is a plot of the denoised function (signal) superimposed over the original series.

Scrolling down or clicking on the **Noise** entry on the left-hand side of the spool brings the graph of the noise (difference) into view.

Wavelet Outlier Detection

Wavelet outlier detection is described in "Wavelet Outlier Detection" on page 779. To perform wavelet outlier detection in EViews, open a series window and proceed to **View/Wavelet Analysis/Outlier Detection...** to display the dialog.

The dialog is identical to the **Wavelet Thresholding** dialog described in

Example (Discrete Wavelet Transform Outlier Detection)

To demonstrate outlier detection, we employ data obtained from the U.S. Geological Survey website https://www.usgs.gov/.

As discussed in Bilen and Huzurbazar (2002), data collected in this database comes from many different sources and is generally notorious for input errors. Here we focus on a monthly dataset, collected at irregular intervals from Mary 19876 to June 2020, measuring water conductance at the Green River near Greendale, UT, identified by site number 09234500.

A quick glance at the series indicates that there is an unusually large drop from typical values (500 to 800 units) in September 1999 which drops to roughly 7.4 units. This is an unusually large drop and is almost certainly an outlying observation. In an attempt to identify the aforementioned outlier and perhaps uncover others, the example conducts the Bilen and Huzurbazar (2002) wavelet outlier detection using a DWT transform with a Haar filter, universal threshold, and a mean median absolute deviation estimator for wavelet coefficient variance. Furthermore, the maximum decomposition level is restricted to 3.

To proceed, open the series window and click on **View/Wavelet Analysis/Outlier Detection...**. Change the **Max Scale** to **4** and under the **MAD** group, set the **Method** to **Mean Median Abs. Dev.**, and click on **OK**.

EViews will display output in the form of a spool object with the spool tree listing the summary, outlier table, and outlier graphs for each of the relevant scales.

The first spool entry is a summary of the outlier detection procedure describing the decomposition method and filter, sample information, threshold method, and wavelet coefficient variance computation.

Next is an outlier overview table listing the exact location of a detected outlier along with its value and absolute deviation from the series mean and median, respectively.

We may scroll down or click on one of the scale entries on the left-hand side to display graphs of the outliers.

These graphs show, for each scale, plots of the original series with red dots and dotted vertical lines identifying outlying observations.

Saving Results to the Workfile

If you wish to save the wavelet results to the workfile you may click on **Proc/Make Wavelet Objects...**

For discussion, see "Wavelet Objects" on page 586.

Technical Background

Many economic time series feature time-varying characteristics such as non-stationarity, volatility, seasonality, and structural discontinuities. Wavelet analysis is a natural framework for analyzing these phenomena without imposing simplifying assumptions such as stationarity. In particular, wavelet filters can decompose and reconstruct a time series, as well as its correlation structure, across time scales. Moreover, these decompositions can be orthogonalized so that a decomposition at one scale is uncorrelated with a decomposition at another. This property is clearly useful in isolating features associated with a given timescale, as well

as achieving *de facto* decomposition and preservation of variation akin to eigenvalue diagonalization in principal component analysis.

Wavelet analysis is analogous, in many ways, to Fourier spectral analysis as both methods can represent a time series signal in a different space. This is achieved by expressing, to a desired degree of precision, the time series signal as a linear combination of basis functions.

For Fourier analysis, these basis functions are sines and cosines. These functions approximate global variation well, but are poorly adapted to modeling local time-variation, and to capturing discontinuous, non-linear, and non-stationary processes whose impulses persist and evolve with time.

Accordingly, Fourier basis functions approximate global variation well. Nevertheless, local variation, otherwise known as time-variation in time-series, is poorly adapted to cylical functions such as the sinus and cosinus. Accordingly, the trigonometric basis functions are ideally adapted to stationary processes characterized by impulses which wane with time, but are poorly adapted to discontinuous, non-linear, and non-stationary processes whose impulses persist and evolve with time. The Fourier transform is localized well in frequency, but lacks good time resolution. Furthermore, if components captured at a given frequency are non-stationary and time-dependent, then attempting to capture all such components with a single frequency (which is itself time independent), will clearly not work very well. To surmount this xed time-frequency relationship, a new set of basis

functions are needed.Alternatively, the wavelet transform, and therefore the wavelet basis functions, are localized both in scale and time. In particular, the latter form a set of shifts and dilations of a *mother wavelet function* $\psi(t)$ which is any function satisfying:

$$\begin{aligned} \int_{-\infty}^{\infty} \psi(x)\, dx &= 0 \\ \int_{-\infty}^{\infty} \psi(x)^2\, dx &= 1 \end{aligned} \tag{42.107}$$

We say that wavelets have mean zero and *unit energy*. The term energy originates in the signal processing literature, and is formalized as

$$\int_{-\infty}^{\infty} \left| f(t)^2 \right| dt \tag{42.108}$$

for some function $f(t)$. The term is interchangeable with the familiar *variance* for non-complex functions. Wavelet basis functions assume the form:

$$\psi_{a,\, b}(t) = \frac{1}{\sqrt{b}} \psi\left(\frac{t-b}{a}\right) \tag{42.109}$$

where b is the location constant, and a is the scaling factor (which corresponds to the notion of frequency in Fourier analysis). Wavelet transforms are well-adapted to non-stationary and discontinuous phenomena since both scale and time resolutions are preserved.

Since wavelet basis functions are *de facto* location and scale transformations of a single function, they are an ideal tool for *multiresolution analysis* (MRA)–the ability to analyze a signal at different frequencies with varying resolutions.

Moving along the time domain, MRA allows one to zoom to a desired level of detail such that high (low) frequencies yield good (poor) time resolutions and poor (good) frequency resolutions. Since economic time series often exhibit multiscale features, wavelet techniques can decompose these series into constituent processes associated with different time scales.

Discrete Wavelet Transform

In the context of continuous functions, the continuous wavelet transform (CWT) of a time series $y(t)$ is defined as

$$W(a, b) = \int_{-\infty}^{\infty} y(t)\, \psi_{a,\, b}(t)\, dt \tag{42.110}$$

and the inverse transformation for reconstructing the original process is

$$y(t) = \int_{-\infty}^{\infty} \int_{0}^{\infty} W(a, b)\, \psi_{a,\, b}(t)\, da\, db \tag{42.111}$$

See Percival and Walden (2000) for a detailed discussion.

Since continuous functions are rarely observed, a discretized analogue of the CWT is generally employed. The discrete wavelet transform (DWT) is a collection of CWT "slices" at nodes $(a, b) = (a_k, b_j)$ such that $a_k = 2^j k$, $b_j = 2^j$, where $j = 1, \ldots, M$, for dyadic observation length $T = 2^M$ for $M \geq 0$. The resulting discrete wavelet basis function is of the form

$$\psi_{k,\, j}(t) = 2^{-j/2} \psi(2^{-j} t - k) \tag{42.112}$$

Unlike the CWT which is highly redundant in both location and scale, the DWT can be designed as an orthonormal transformation. If the location discretization is restricted to the index $k = 1, \ldots, 2^{-j} T$, at each scale $\lambda_j = 2^{j-1}$, half the available observations are lost in exchange for orthonormality. This is the classical DWT framework. Alternatively, if the location index is restricted to the full set of available observations with $k = 1, \ldots, T$, the discretized transform is no longer orthonormal, but does not suffer from observation loss. The latter framework is typically referred to as the *maximal overlap discrete wavelet transform* (MODWT), and sometimes as the *non-decimated* DWT.

Mallat's Pyramid Algorithm

In practice, the DWT proceeds as a filtering operation (both low and high pass) on a time series, following the pyramid algorithm of Mallat (1989).

In case of the classical DWT with $T = 2^M$, let $y = \{y_t\}_{t=1}^T$ and define the matrix of DWT coefficients $W = [W_1, \ldots, W_M, V_M]'$. Here, W_j is a vector of wavelet coefficients of length $T/2^j$ and is associated with changes on a scale of length $\lambda_j = 2^{j-1}$. Moreover, V_M is a vector of scaling coefficients of length $T/2^j$ and is associated with averages on a scale of length -$\lambda_M = 2^{M-1}$. Furthermore, $W = wy$ where $w = [w_1, \ldots, w_M, v_M]$ is some $T \times T$ orthonormal matrix generating the DWT coefficients.

The algorithm convolves an input signal with wavelet high and low pass filter coefficients to derive the j-th level DWT matrix $[W_1, \ldots, W_j, V_j]'$. In particular, each iteration j convolves the scaling coefficients from the preceding iteration, namely $V_{t,j-1}$, with both the high and low pass filter coefficients. The input signal in the first iteration is y_t. The entire algorithm continues until the M-th iteration, although it can be stopped earlier.

Wavelet Filters

Like traditional filters which are used to extract components of a time series such as trends, seasonalities, business cycles, and noise, wavelets filters perform a similar role. Like traditional filters, wavelet filters

are also designed to capture low and high frequencies, and have a particular length. This length governs how much of the original series information is used to extract low and high frequency phenomena. The role of the length is very similar to the role of the autoregressive (AR) order in traditional time series models where higher AR orders imply more historical observations influence the present.

The simplest and shortest wavelet filter is of length $L = 2$ and is called the Haar wavelet. This is a sequence of rescaled rectangular functions and is therefore ideally suited to analyzing signals with sudden and discontinuous changes. In this regard, it is ideally suited for outlier detection. Unfortunately, the Haar filter is typically too simple for most other applications.

In contrast, Daubechies (1992) introduced a family of filters of even length that are indexed by the polynomial degree they are able to capture – rather the number of vanishing moments. Thus, the Haar filter, which is of length 2, can only capture constants and linear functions. The Daubechies wavelet filter of length 4 can capture everything from a constant to a cubic function. Accordingly, higher filter lengths are associated with higher smoothness.

It should be noted that Daubechies (1992) filters, also known as daublets, are typically not symmetric. If a more symmetric version of the daublet filters is required, then the class known as least asymmetric, or *symmlets*, is used.

Boundary Conditions

It's important to note that both the DWT and the MODWT make use of circular filtering. In particular, when a filtering operation reaches the beginning or end of an input series, otherwise known as the boundaries, the filter treats the input time series as periodic with period T. In other words, we assume that $y_{T-1}, y_{T-2}, \ldots$, are useful surrogates for unobserved values $y_{-1}, y_{-2}, \ldots$. Those wavelet coefficients which are affected by this circularity are also known as *boundary coefficients*. Note that the number of boundary coefficients only depends on the filter length L and is independent of the input series length T. Furthermore, the number of boundary coefficients increases with filter length L. Both DWT and MODWT boundary coefficients will appear the beginning of W_j and V_j. You may refer to Percival and Walden (2000) for further details.

Multiresolution Analysis

Similar to Fourier, spline, and linear approximations, a principal feature of the DWT is the ability approximate an input series as a function of wavelet basis functions. The general idea in wavelet theory is termed *multiresolution analysis* (MRA) and refers to the approximation of an input series at each scale (and up to all scales) λ_j. For each $m = 1, \ldots, M$,

$$\begin{aligned} y &= w' W \\ &= \sum_{j=1}^{m} D_j + S_m \end{aligned} \tag{42.113}$$

where the $D_j = w_j' W_j$ and $S_m = v_m' V_m$ are T-dimensional vectors of the j-th level *detail* and the m-th level *smooth* series. Furthermore, since the low-pass (high-pass) wavelet coefficients are associated with changes (averages) at scale λ_j, the detail and smooth series are associated with changes and average at scale λ_j, respectively, in the input series y.

The MRA is typically used to derive approximations for the original series using its lower and upper frequency components. Since upper frequency components are associated with transient features and are captured by the wavelet coefficients, the detail series will in fact extract those features of the original series which are typically associated with "noise". Alternatively, since lower frequency components are associated with perpetual features and are captured by the scaling coefficients, the smooth series will in fact extract those features of the original series which are typically associated with the "signal".

It's worth mentioning here that since wavelet filtering can result in boundary coefficients, the detail and smooth series will also have observations affected by the boundary.

Wavelet Analysis of Variance

The orthonormality of the DWT generating matrix w has important implications. In particular, the DWT is an energy (variance) preserving transformation. Coupled with this preservation of energy is the decomposition of energy on a scale-by-scale basis. Then,

$$\|y\|^2 = \sum_{j=1}^{M} \|W_j\|^2 + \|V_M\|^2 \tag{42.114}$$

where $\|\cdot\|$ is the Euclidean norm. Thus, $\|W_j\|^2$ quantifies the energy of y_t accounted for at scale λ_j.

This energy decomposition is known as the wavelet power spectrum (WPS) and is arguably the most important of the properties of the DWT, and bears resemblance to the spectral density function (SDF) used in Fourier analysis.

The total variance in y can be decomposed as

$$\sum_{j=0}^{\infty} \nu^2(\lambda_j) = \mathrm{var}(y) \tag{42.115}$$

where $\nu^2(\lambda_j)$ is the contribution to $\mathrm{var}(y)$ due to scale λ_j, and is estimated as

$$\hat{\nu}^2(\lambda_j) = \frac{1}{T} \sum_{t=1}^{T} W_{t,j}^2 \tag{42.116}$$

since $\|W_j\|^2 / T$ is the contribution to the sample variance of y_t at scale λ_j.

Unfortunately, this estimator is biased due to the presence of boundary coefficients. To derive an unbiased estimate, boundary coefficients should be dropped from consideration. Accordingly, an unbiased estimate of the variance contributed at scale λ_j is given by

$$\tilde{\nu}^2(\lambda_j) = \frac{1}{M_j} \sum_{t=\kappa_j+1}^{T} W_{t,j}^2 \tag{42.117}$$

where $M_j = T - \kappa_j$ and κ_j represents the number of coefficients affected by boundary conditions.

Variance Confidence Intervals

It is also possible to derive confidence intervals for the contribution to the overall variance at each scale.

An asymptotic normal confidence interval is given by

$$[\tilde{\nu}^2(\lambda_j) - R,\ \tilde{\nu}^2(\lambda_j) + R] \tag{42.118}$$

where

$$R = \Phi^{-1}(1-\alpha)\left(\frac{2A_j}{M_j}\right)^{1/2} \tag{42.119}$$

and A_j is the integral of the squared spectral density function of wavelet coefficients W_j excluding any boundary conditions. As shown in Percival and Walden (2000), A_j can be estimated as the sum of squared serial correlations among W_j excluding boundary coefficients. Priestly (1981) notes that there is no condition that prevents the lower bound of this confidence interval from becoming negative.

Percival and Waldon (2000) employ an asymptotic Chi-square confidence interval

$$[\frac{\eta\tilde{\nu}^2(\lambda_j)\,C}{Q_\eta(1-\alpha)},\ \frac{\eta\tilde{\nu}^2(\lambda_j)}{Q_\eta(\alpha)}] \tag{42.120}$$

where $Q_\eta(\alpha)$ is the α-quantile of the χ^2 distribution and η is the equivalent degrees of freedom (EDOF) which may be estimated using one of two suggestions

$$\begin{aligned} \eta_1 &= M_j\tilde{\nu}^4(\lambda_j) \\ \eta_2 &= \max\{2^{-j}M_j,\ 1\} \end{aligned} \tag{42.121}$$

The first estimator, η_1, relies on large sample theory, and in practice requires a sample of at least $T = 128$ to yield a decent approximation. The second estimator, η_2, assumes that the spectral density function of the wavelet coefficients at scale λ_j is a *band-limited*. See Percival and Walden (2000) for detail.

Wavelet Outlier Detection

A particularly important and useful application of wavelets is outlier detection. While the subject matter has received some attention over the years starting with Greenblatt (1996), here we focus on a rather simple and appealing contribution by Bilen and Huzurbazar (2002).

The appeal of the Bilen and Huzurbazar approach is that it doesn't require model estimation, is not restricted to processes generated via ARIMA, and works in the presence of both additive and innovational outliers. The approach also assumes that wavelet coefficients are approximately independent and identically normal variates. While EViews offers the ability to perform this procedure using a MODWT, it's generally better suited to the DWT. Moreover, Bilen and Huzurbazar suggest that Haar is the preferred filter since the latter yields coefficients large in magnitude in the presence of jumps or outliers. They also suggest that the transformation be carried out only at the first scale.

The overall procedure works on the principle of thresholding and the authors suggest the use of the universal threshold. The idea here is that extreme (outlying) values will register as noticeable spikes in the spectrum. These spike generating values are outlier candidates.

For m_j, the number of wavelet coefficients at scale λ_j, the outlier detection algorithm is given by:

1. Compute a wavelet transformation of the original data up to some scale $J^* \leq M$.
2. Select an optimal threshold η.
3. For $j = 1, \ldots, J^*$ find the set of indices $S = \{s_1, \ldots, s_{m_j}\}$ such that $|W_{i,j}| > \eta$ for $i = 1, \ldots, m_j$
4. Find the exact location of the outlier among the original observations. For instance, if s_i is an index associated with an outlier:
   - If the wavelet transform is the DWT, the original observation associated with that outlier is either $2^j s_i$ or $2^j s_i - 1$. To choose between the two, let $\tilde{\mu}$ denote the mean of the original observations excluding those two values. Between the two candidates, the outlier is the one which has the largest absolute deviation from $\tilde{\mu}$.
   - If the wavelet transform is MODWT, the outlier is the observation indexed by s_i.

See Bilen and Huzurbazar (2002) for additional detail and discussion.

Wavelet Shrinkage (Thresholding)

A key objective in any empirical work is to discriminate noise from useful information. Suppose that the observed time series $y_t = x_t + \epsilon_t$ where x_t is an unknown signal of interest obscured by the presence of unwanted noise ϵ_t. Traditionally, signal discernment was typically achieved using discrete Fourier transforms. Naturally, this assumes that any signal is an infinite superposition of sinusoidal functions; a strong assumption in empirical econometrics where most data exhibits unit roots, jumps, kinks, and various other non-linearities.

The principle behind wavelet-based signal extraction, otherwise known as *wavelet shrinkage*, is to shrink any wavelet coefficients not exceeding some threshold to zero and then exploit the MRA to synthesize the signal of interest using the modified wavelet coefficients. In other words, only those wavelet coefficients associated with very pronounced spectra are retained with the additional benefit of deriving a very sparse wavelet matrix.

Thresholding Rule

The two most popular thresholding rules are

- Hard Thresholding ("kill/keep" strategy)

$$\delta_\eta^H(x) = \begin{cases} x & \text{if } |x| > \eta \\ 0 & \text{otherwise} \end{cases} \tag{42.122}$$

- Soft Thresholding

$$\delta_\eta^S(x) = \text{sign}(x) \cdot \max\{0, |x| - \eta\} \tag{42.123}$$

Threshold Value Methods

The threshold value η is the key to wavelet shrinkage. In particular, optimal thresholding is achieved when $\eta = \sigma_\epsilon$, the standard deviation of the noise process ϵ. There are several strategies for determining the optimal threshold:

- **Universal Threshold** (Donoho and Johnstone 1994),

$$\eta = \hat{\sigma}_\epsilon \sqrt{2\log(T)} \tag{42.124}$$

  where $\hat{\sigma}_\epsilon$ is estimated using wavelet coefficients only at scale λ_1, regardless of the scale under consideration. When this threshold rule is coupled with soft thresholding, the combination is commonly referred to as *VisuShrink*.

- **Minimax Estimation** (Donoho and Johnstone 1994),

$$\inf \sup R \tag{42.125}$$

  where R is the familiar average mean squared error measurement of risk. Unfortunately, a closed form solution for this approach is not available, although tabulated values exist. When this threshold rule is coupled with soft thresholding, the combination is commonly referred to as *RiskShrink*.

- **Stein's Unbiased Risk Estimate** - SURE (Donoho and Johnstone 1994; Jansen 2010),

$$\min_{\hat{\mu}} \|\mu - \hat{\mu}\|^2 \tag{42.126}$$

  where $\mu = (\mu_1, \ldots, \mu_s)'$ and μ_k is the mean of some variable of interest, $q_k \sim N(\mu_k, 1)$ for $k = 1, \ldots, s$. In the framework of wavelet coefficients, q_k represents the wavelet coefficients at a given scale.

  Furthermore, while the optimal threshold η based on this rule depends on the thresholding rule used, the solution may not be unique, so the SURE threshold value is the minimum of the η. In case of the soft thresholding rule, the solution was proposed in Donoho and Johnstone (1994). For the hard thresholding rule, the solution was proposed in Jansen (2010).

- **False Discovery Rate** (Abramovich and Benjamini 1995), determines the optimal threshold value through a multiple hypothesis testing problem. For further detail, see Donoho and Johnstone (1995), Donoho, Johnstone *et al.* (1998), Gençay, Selçuk andWhitcher (2001), and Percival and Walden (2000).

Wavelet Coefficients Variance

There remains the issue of how to estimate the variance of the wavelet coefficients σ_ϵ^2. If the assumption is that the observed data y is obscured by some noise process ϵ, the usual estimator of variance will exhibit extreme sensitivity to noisy observations.

Let μ_j and ζ_j denote the mean and median, respectively, of the wavelet coefficients W_j at scale λ_j, and let M_j denote its cardinality (total number of coefficients at that scale). Then, several common estimators have been proposed in the literature:

- Mean Absolute Deviation

$$\hat{\sigma}_{\epsilon,j} = \frac{1}{m_j}\sum_{i=1}^{m_j} |W_{i,j} - \mu_j| \tag{42.127}$$

- Median Absolute Deviation

$$\hat{\sigma}^2_{\epsilon,j} = \mathrm{med}(W_{1,j} - \zeta_j, \ldots, W_{i,j} - \zeta_j) \tag{42.128}$$

- Mean Median Absolute Deviation

$$\hat{\sigma}^2_{\epsilon,j} = \frac{1}{m_j}\sum_{i=1}^{m_j} |W_{i,j} - \zeta_j| \tag{42.129}$$

- Median (Gaussian)

$$\hat{\sigma}^2_{\epsilon,j} = \frac{\mathrm{med}(W_{1,j}, \ldots, W_{m_j,j})}{0.6745} \tag{42.130}$$

Thresholding Implementation

Given choice of the thresholding rules and optimal threshold value, all wavelet thresholding procedures involve the following:

1. Compute a wavelet transformation of the original data up to some scale $J^* \leq M$ so that you obtain the wavelet and scaling coefficients $W_1, \ldots, W_{J^*}, V_{J^*}$.
2. Select an optimal threshold η.
3. Threshold the coefficients at each scale λ_j for $j = 1, \ldots, J^*$ using the selected threshold value and rule. This procedure will generate a set of modified wavelet and scaling coefficients $W_1^*, \ldots, W_{J^*}^*, V_{J^*}^*$.
4. Use MRA with the thresholded coefficients to generate an estimate of the signal:

$$\hat{y} = \sum_{j=1}^{J^*} D_j^* + S_{J^*}^* \tag{42.131}$$

where the $D_j^* = w_j^{*\prime} W_j^*$ and $S_{J^*} = v_{J^*}^{*\prime} V_{J^*}^*$

Practical Considerations

The discussion above outlines the basic theory underlying wavelet analysis. Nevertheless, there are several practical (empirical) considerations which should be addressed. We focus

here on three in particular: 1) selecting a wavelet filter, 2) handling non-dyadic series lengths, and 3) awareness of boundary conditions.

Choice of Wavelet Filter

The type of wavelet filter is typically chosen to mimic the data to which it is applied. Shorter filters don't approximate the ideal band pass filter well, but longer ones do. On the other hand, if the data are derived from piecewise constant functions, the Haar wavelet or other shorter wavelets may be more appropriate. Alternatively, if the underlying data is smooth, longer filters may be more appropriate. In this regard, it's important to note that longer filters expose more coefficients to boundary condition effects than

shorter ones. Accordingly, the rule of thumb strategy is to use the filter with the smallest length that gives reasonable results. Furthermore, since the MODWT is not orthogonal and its wavelet coefficients are correlated, wavelet filter choice is not as vital as for orthogonal DWT. Nevertheless, if alignment to time is important (*i.e.* zero phase filters), the least asymmetric family of filters may be a good choice.

Handling Boundary Conditions

Recall that wavelet filters exhibit boundary conditions due to circular recycling of observations. Although this may be an appropriate assumption for series with naturally exhibiting cyclical effects, it is not appropriate in all circumstances. An alternative approach is to reflect the original series to generate a series of length $2T$ so that filtering proceeds using observations $y_1, \ldots, y_T, y_T, \ldots, y_1$.

Whatever the approach, wavelet analysis should quantify how many wavelet coefficients are affected by boundary conditions.

Adjusting Non-dyadic Length Time Series

Recall that the DWT requires an input series of dyadic length. Naturally, this condition is rarely satisfied in practice. Broadly speaking, there are two approaches: either shorten the input series to dyadic length at the expense of losing observations, or "pad" the input series with observations to achieve dyadic length.

Although the choice of padding values is ultimately arbitrary, there are three popular choices, none of which has proven superior: 1) pad with zeros, 2) pad with mean, 3) pad with median.

References

Abramovich, Felix, and Yoav Benjamini (1995). "Thresholding of wavelet coefficients as multiple hypotheses testing procedure." In: Antoniadis A., Oppenheim G. (*eds.*) *Wavelets and Statistics. Lecture Notes in Statistics, vol. 103*. New York: Springer, 5–14.

Ahn, Seung C, and Alex R. Horenstein (2013). "Eigenvalue ratio test for the number of factors." E*conometrica*, 81, 1203–1227.

Bai, Jushan, and Serena Ng (2002). "Determining the number of factors in approximate factor models." *Econometrica*, 70, 191–221.

Bai, Jushan, and Serena Ng (2004). "A PANIC attack on unit roots and cointegration." *Econometrica*, 72, 1127–1177.

Banerjee, Anindya, Robin L. Lumsdaine, and James H. Stock (1992). "Recursive and sequential tests of the unit-root and trend-break hypotheses: Theory and international evidence," *Journal of Business & Economic Statistics*, 10, 271–287.

Banerjee, Anindya, and Martin Wagner (2009). "Panel methods to test for unit roots and cointegration," in *Palgrave Handbook of Econometrics*, 632–726. Springer.

Bhargava, A. (1986). "On the theory of testing for unit roots in observed time series," *Review of Economic Studies*, 53, 369-384.

Bilen, Canan, and S. Huzurbazar (2002). "Wavelet-based detection of outliers in time series." *Journal of Computational and Graphical Statistics*, 11, 311–327.

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.

Breitung, J. (2002). "Nonparametric tests for unit roots and cointegration," *Journal of Econometrics,* 108, 343–363.

Brock, William, Davis Dechert, Jose Sheinkman and Blake LeBaron (1996). "A test for independence based on the correlation dimension," *Econometric Reviews*, August, 15(3), 197–235.

Canova, F. and Hansen, B. E. (1995). "Are seasonal patterns constant over time? a test for seasonal stability," *Journal of Business & Economic Statistics,* 13, 237–252.

Choi, In (2001). "Unit root tests for panel data," *Journal of International Money and Finance*, 20, 249–272.

Chow, K. Victor and Karen C. Denning (1993). "A simple multiple variance ratio test," *Journal of Econometrics*, 58, 385–401.

Daubechies, I. (1992). *Ten lectures on Wavelets: CBMS-NSF Conference Series in Applied Mathematics*. Society for Industrial and Applied Mathematics.

Davidson, Russell and James G. MacKinnon (1993). *Estimation and Inference in Econometric*s, 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.

Dickey, D. A. and Fuller, W. A. (1981). "Likelihood ratio statistics for autoregressive time series with a unit root," *Econometrica*, 49, 1057–1072.

Donoho, David L, and Iain M. Johnstone (1994). "Ideal spatial adaptation by wavelet shrinkage," B*iometrika*, 81, 425–455.

Donoho, David L, and Iain M. Johnstone (1995). "Adapting to unknown smoothness via wavelet shrinkage," *Journal of the American Statistical Association*, 90, 1200–1224.

Donoho, David L. and Iain M Johnstone (1998). "Minimax estimation via wavelet shrinkage," *The Annals of Statistics*, 26 879–921.

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.

Fong, Wai Mun, See Kee Koh, and Sam Ouliaris (1997). "Joint variance-ratio tests of the martingale hypothesis for exchange rates," *Journal of Business and Economic Statistics*, 15, 51-59.

Fisher, R. A. (1932). *Statistical Methods for Research Workers, 4th Edition*, Edinburgh: Oliver & Boyd.

Gençay, R., F. Selçuk, and B. J.Whitcher (2001). *An Introduction to Wavelets and Other Filtering methods in Finance and Economics*. San Diego: Academic Press.

Greenblatt, Seth A. 1996. "Wavelets in Econometrics." In *Computational Economic Systems: Advances in Computational Economics, vol 5.*, Dordrecht: Springer, 139–160.

Hadri, Kaddour (2000). "Testing for stationarity in heterogeneous panel data." *The Econometrics Journal*, 3, 148–161.

Hall, Alistair (1994). "Testing for a unit root in time series with pretest data based model selection," *Journal of Business and Economic Statistics*, 12, 461–470.

Hamilton, James D. (1994). *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.

Hylleberg, S., Engle, R. F., Granger, C. W., and Yoo, B. S. (1990). "Seasonal integration and cointegration," *Journal of Econometrics,* 44, 215–238.

Im, Kyung So, M. Hashem Pesaran, and Yongcheol Shin (2003). "Testing for unit roots in heterogeneous panels." *Journal of Econometrics*, 115, 53–74.

Jansen, Maarten (2010). "Minimum risk methods in the estimation of unknown sparsity." Technical report.

Johansen, Søren (1991). "Estimation and hypothesis testing of cointegration vectors in Gaussian vector autoregressive models," *Econometrica*, 1551–1580.

Kim, Dukpa and Pierre Perron (2009). "Unit root tests allowing for a break in the trend function at an unknown time under both the null and alternative hypotheses," *Journal of Econometrics*, 148, 1–13.

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.

Lo, Andrew W. and A. Craig MacKinlay (1988). "Stock market prices do not follow random walks: evidence from a simple specification test," *The Review of Financial Studies*, 1, 41–66.

Lo, Andrew W. and A. Craig MacKinlay (1989). "The size and power of the variance ratio test in finite samples," *Journal of Econometrics*, 40, 203-238.

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 Shaowen Wu (1999). "A comparative study of unit root tests with panel data and a new simple test," *Oxford Bulletin of Economics and Statistics*, 61, 631-652.

Mallat, S. G. (1989). "A theory for multiresolution signal decomposition: The wavelet representation." *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 11 674–693.

Nelson, Charles R. and Charles I. Plosser (1982). "Trends and random walks in macroeconomic time series," *Journal of Monetary Economics*, 10, 139–162.

Newey, W., and K. West (1987). "A simple positive semi-definite heteroscedasticity and autocorrelation consistent covariance matrix," *Econometrica*, 55,150–161.

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, 1519-1554.

Oppenheim, A, and R. Schafer (1989). *Discrete-Time Signal Processing*. PTR Prentice Hall. Inc.

Percival, D. B., and A. T. Walden. 2000. *Wavelet Methods for Time Series Analysis*. New York: Cambridge University Press.

Perron, Pierre (1989). "The Great Crash, the oil price shock, and the unit root hypothesis," *Econometrica*, 57, 1361-1401.

Perron, Pierre (1997). "Further evidence on breaking trend functions in macroeconomic variables," *Journal of Econometrics*, 80, 355–385.

Perron, Pierre (2006). "Dealing with Structural Breaks," in *Palgrave Handbook of Econometrics, Vol. 1: Econometric Theory*, K. Patterson and T. C. Mills (eds.), Palgrave Macmillan, 278-352.

Perron, Pierre and Timothy J. Vogelsang (1992a). "Nonstationarity and level shifts with an application to purchasing power parity," *Journal of Business & Economic Statistics*, 10, 301–320.

Perron, Pierre and Timothy J. Vogelsang (1992b). "Testing for a unit root in a time series with a changing mean: corrections and extensions," *Journal of Business & Economic Statistics*, 10, 467–470.

Pesaran, M Hashem (2006). "Estimation and inference in large heterogeneous panels with a multifactor error structure," *Econometrica*, 74, 967–1012.

Pesaran, M Hashem (2007). "A simple panel unit root test in the presence of cross-section dependence," *Journal of Applied Econometrics*, 22, 265–312.

Phillips, P.C.B. (1987). "Time series regression with a unit root." *Econometrica*, 277–301.

Phillips, P.C.B. and P. Perron (1988). "Testing for a unit root in time series regression," *Biometrika*, 75, 335–346.

Priestley, M. B. (1981). *Spectral Analysis and Time Series*. Academic Press.

Richardson, Matthew and Tom Smith (1991). "Tests of financial models in the presence of overlapping observations," *The Review of Financial Studies*, 4, 227–254.

Said, Said E. and David A. Dickey (1984). "Testing for unit roots in autoregressive moving average models of unknown order," *Biometrika*, 71, 599–607.

Schwert, G William (1989. "Tests for unit roots: A Monte Carlo investigation." *Journal of Business & Economic Statistics*, 7, 147–59.

Smith, R. J. and Taylor, A. R. (1999). "Likelihood ratio tests for seasonal unit roots," *Journal of Time Series Analysis*, 20, 453–476.

Stock, James H, and Mark W. Watson (1988. "Testing for common trends." *Journal of the American Statistical Association*, 83, 1097–1107.

Taylor, A. R. (2005). "Variance ratio tests of the seasonal unit root hypothesis," *Journal of Econometrics*, 124, 33–54.

Taylor, R. A. (2003). "Robust stationarity tests in seasonal time series processes," *Journal of Business & Economic Statistics,* 21, 156–163.

Vogelsang, Timothy J. and Pierre Perron (1998). "Additional test for unit root allowing for a break in the trend function at an unknown time," *International Economic Review*, 39, 1073–1100.

Vogelsang, Timothy J. (1993). Unpublished computer program.

Wright, Jonathan H. (2000). "Alternative variance-ratio tests using ranks and signs," *Journal of Business and Economic Statistics*, 18, 1–9.

Zivot, Eric and Donald W. K. Andrews (1992). "Further evidence on the Great Crash, the oil-price shock, and the unit-root hypothesis," *Journal of Business & Economic Statistics*, 10, 251–270.

Part VIII. Multiple Equation Analysis

In this section, we document EViews tools for multiple equation estimation, forecasting and data analysis.

- Chapter 43. "System Estimation," on page 791 describes estimation techniques for systems of equations ().
- Chapter 44. "Vector Autoregression and Error Correction Models," on page 833 describes estimation of standard VARs, structural VARs, and VECs.
- Next, we describe various extensions of the VAR model: Bayesian VARs (Chapter 45. "Bayesian VAR Models," on page 895), mixed frequency VARs (Chapter 46. "Mixed Frequency VAR," on page 927), and switching VARs (Chapter 47. "Switching VAR," on page 939).
- Chapter 48. "State Space Models and the Kalman Filter," on page 969 describes the use of EViews' state space and Kalman filter tools for modeling structural time series models.
- Chapter 49. "Models," beginning on page 995 describes the use of model objects to forecast from multiple equation estimates, or to perform multivariate simulation.

Chapter 43. 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 49. "Models," on page 995 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:

$$f(y_t, x_t, \beta) = \epsilon_t, \tag{43.1}$$

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 49. "Models," on page 995.

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) (with both unrestricted and restricted covariance matrices).
- 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 820.

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 (STSLS) estimator is the system version of the single equation two-stage least squares estimator described above. STSLS 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 STSLS 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 (WTSLS) estimator is the two-stage version of the weighted least squares estimator. WTSLS 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 STSLS 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 specified correctly, 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.

Autogressive 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: Constant Conditional 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. There are three ways to specify the system: manually by entering a specification, by inserting a text file containing the specification, or by letting EViews create a system automatically from a selected list of variables,

To create a new system manually or by inserting a text file, click on **Object/New Object…/System** or type `system` in the command window. A blank system object window should appear. You will fill the system specification window with text describing the equations, and potentially, lines describing the instruments and the parameter starting values. You may enter the text by typing in the specification, or clicking on the **InsertTxt** button and loading

a specification from a text file. You may also insert a text file using the right-mouse button menu and selecting **Insert Text File...**

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 selected 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 808 for additional details on this dialog.

The **Make System** proc is also available from a Group object (see "Make System," on page 663).

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)*x1 + c(2)*x2 + c(3)*x3
  ```

  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) procedure, 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 equations, 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 vari-

ables 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:

$$\begin{bmatrix} Z1 & M1 & IA & C & 0 & 0 & 0 & 0 & 0 \\ Z2 & M1 & 0 & 0 & IA & C & 0 & 0 & 0 \\ Z3 & M1 & 0 & 0 & 0 & 0 & IA & C & IC \end{bmatrix} \tag{43.2}$$

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 spec-

ification. 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.

GMM Settings

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 820 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.

ARCH Settings

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 or the constant term. For 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.

FIML Settings

For systems estimated using FIML, you will be prompted to specify estimation settings for the parameterization of the residual covariance matrix.

The **Residual Covariance** drop down allows you to choose between the default **Unrestricted**, and **Diagonal**, **User-covariance**, and **User-factor** settings.

- The **Unrestricted** setting specifies standard FIML estimation in which all of the elements of the residual covariance matrix are estimated along with the coefficients of the mean equation. Note that the variance parameters for FIML are treated as nuisance parameters.
- The **Diagonal** setting places zero restrictions on the off-diagonals of the residual covariance matrix. Only the diagonal elements of the residual covariance matrix corresponding to the variances are estimated.
- **User-covariance** allows you to provide the name of a sym matrix object containing values for all of the residual covariances. All elements of the residual covariance Σ must be specified; no elements will be estimated.
- **User-factor** offers an alternative method of specifying the residual covariance. Instead of providing a sym matrix containing values of Σ, you will provide the name of a matrix P where $PP' = \Sigma$. Again, no elements of the residual covariance matrix are estimated.

Note that system objects estimated using a restricted FIML (diagonal, user-specified, or user-factor) estimator are not backward compatible with earlier versions of EViews, and will be dropped from the workfile if opened in a version prior to 9.5.

Options

For weighted least squares, SUR, weighted TSLS, 3SLS, GMM, and nonlinear systems of equations, there are additional options controlling 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 **Option**s 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,

$$H_0 = \epsilon_0 \epsilon_0' = \lambda^T \hat{H} + (1-\lambda) \sum_{j=0}^{T} \lambda^{T-j-1} \epsilon_{T-j} \epsilon_{T-j}' \tag{43.3}$$

where:

$$\hat{H} = \sum_{t=1}^{T} (\epsilon_t \epsilon_t') / T \tag{43.4}$$

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 1313 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 20, page 16 for a discussion of the use of these functions, and Chapter 1. "Object View and Procedure Reference," on page 2 of the *Command and Programming 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 887 of the *Command and Programming 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 D. "Gradients and Derivatives," on page 1327.
- **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.
- **Estimation Covariance Matrix** displays the covariance matrix used in estimation:
  1. identity matrix for OLS and TSLS
  2. diagonal matrix with equation variances used to compute WOLS and WTSLS

3. residual covariance matrix used to compute SUR and 3SLS
4. residual covariance matrix for unrestricted FIML; diagonal residual covariance matrix for diagonal FIML; user-specified covariance for user-covariance FIML
5. long-run covariance of the moments used to compute the weighting matrix for GMM estimates
6. matrix of missing values for ARCH

- **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 Intervals and Confidence Ellipses" on page 196 and "Wald Test (Coefficient Restrictions)" on page 202.
- A number of **Residual Diagnostics** 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 849 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 1013.

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 presented in Greene (1997). The data are provided in "G_cost.WF1". The translog cost function has four factors with three equations of the form:

$$\begin{aligned} c_K &= \beta_K + \delta_{KK}\log\left(\frac{p_K}{p_M}\right) + \delta_{KL}\log\left(\frac{p_L}{p_M}\right) + \delta_{KE}\log\left(\frac{p_E}{p_M}\right) + \epsilon_K \\ c_L &= \beta_L + \delta_{LK}\log\left(\frac{p_K}{p_M}\right) + \delta_{LL}\log\left(\frac{p_L}{p_M}\right) + \delta_{LE}\log\left(\frac{p_E}{p_M}\right) + \epsilon_L \\ c_E &= \beta_E + \delta_{EK}\log\left(\frac{p_K}{p_M}\right) + \delta_{EL}\log\left(\frac{p_L}{p_M}\right) + \delta_{EE}\log\left(\frac{p_E}{p_M}\right) + \epsilon_E \end{aligned} \tag{43.5}$$

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**. Click on **OK** to perform the estimation. 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: 08/13/09 Time: 09:10
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

Log likelihood	349.0326	Schwarz criterion	-26.37755
Avg. log likelihood	4.653769	Hannan-Quinn criter.	-26.80034
Akaike info criterion	-26.96261		
Determinant residual covariance		1.50E-16	

while the bottom portion of the output (not depicted) describes equation specific statistics.

To test the symmetry restrictions, select **View/Coefficient Diagnostics/Wald Coefficient Tests…**, fill in the dialog:

and click **OK**. The test result:

Wald Test:
System: SYS_UR
Null Hypothesis: C(3)=C(6), C(4)=C(10), C(8)=C(11)

Test Statistic	Value	df	Probability
Chi-square	0.418796	3	0.9363

Null Hypothesis Summary:

Normalized Restriction (= 0)	Value	Std. Err.
C(3) - C(6)	-0.010622	0.039838
C(4) - C(10)	0.031340	0.077783
C(8) - C(11)	0.058594	0.090758

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**. We will leave the **Residual covariance** combo at the default setting of **Unrestricted** to allow for estimation of an unrestricted covariance matrix. Click on **OK** to estimate the equation.

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 (BFGS / Marquardt steps)
Date: 03/10/15 Time: 22:09
Sample: 1947 1971
Included observations: 25
Total system (balanced) observations 75
Convergence achieved after 38 iterations
Coefficient covariance computed using outer product of gradients

	Coefficient	Std. Error	z-Statistic	Prob.
C(1)	0.057022	0.003306	17.24913	0.0000
C(2)	0.029742	0.012583	2.363697	0.0181
C(3)	-0.000369	0.011205	-0.032971	0.9737
C(4)	-0.010228	0.006027	-1.697176	0.0897
C(5)	0.253398	0.005050	50.17733	0.0000
C(6)	0.075427	0.015483	4.871616	0.0000
C(7)	-0.004414	0.009141	-0.482900	0.6292
C(8)	0.044286	0.003349	13.22339	0.0000
C(9)	0.018767	0.014894	1.260009	0.2077

Log likelihood	344.5916	Schwarz criterion	-26.40853
Avg. log likelihood	4.594555	Hannan-Quinn criter.	-26.72563
Akaike info criterion	-26.84733		
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** to display the residuals dialog. You may choose to save the ordinary or standardized residuals. If you choose the latter, you can elect to standardize the residuals using the Cholesky factor of the (conditional) covariance, the square root of the (conditional) correlation matrix, or the square root of the (conditional) covariance matrix. You must enter a basename for saving the residuals. The residuals will be named using the next available names in the workfile, in this case “RESID01”, “RESID02”, ..., if those names are not already used.

In this example, we elect to produce ordinary residuals. EViews opens an untitled group window containing the residuals of each equation in the system. To compute descriptive statistics for each residual in the group, select **View/Descriptive Stats/Common Sample** from the group window toolbar.

Group: UNTITLED Workfile: G_COST::G_cost\

View | Proc | Object | Print | Name | Freeze | Sample | Sheet | Stats | Spec

	RESID01	RESID02	RESID03
Mean	6.82E-09	5.63E-10	1.76E-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	0.664427	0.835613	0.116650
Kurtosis	3.013741	6.437974	1.488611
Jarque-Bera	1.839625	15.22152	2.436171
Probability	0.398594	0.000495	0.295796
Sum	1.70E-07	1.41E-08	4.40E-08
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 + sys_tlog.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/Graph/Line & Symbol**:

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:

$$\begin{aligned} \log(jy_t/jy_{t-1}) &= c_1 + \epsilon_{1t} \\ \log(sf_t/sf_{t-1}) &= c_2 + \epsilon_{2t} \\ \log(bp_t/bp_{t-1}) &= c_3 + \epsilon_{3t} \end{aligned} \tag{43.6}$$

where $\epsilon_t = [\epsilon_{1t}, \epsilon_{2t}, \epsilon_{3t}]'$ is assumed to distributed normally with mean zero and covariance H_t. The conditional covariance is modeled with a basic Diagonal VECH model:

$$H_t = \Omega + A \otimes \epsilon_{t-1}\epsilon_{t-1}' + B \otimes H_{t-1} \tag{43.7}$$

To estimate this model, create a system SYS01 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 VECH model we leave most of the settings 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: SYSTEM01
Estimation Method: ARCH Maximum Likelihood (BFGS / Marquardt steps)
Covariance specification: Diagonal VECH
Date: 03/10/15 Time: 22:15
Sample: 12/31/1979 12/25/2000
Included observations: 1096
Total system (balanced) observations 3288
Presample covariance: backcast (parameter =0.7)
Convergence achieved after 68 iterations
Coefficient covariance computed using outer product of gradients

	Coefficient	Std. Error	z-Statistic	Prob.
C(1)	-0.000865	0.000446	-1.936740	0.0528
C(2)	5.43E-05	0.000454	0.119510	0.9049
C(3)	-3.49E-05	0.000378	-0.092282	0.9265
Variance Equation Coefficients				
C(4)	6.49E-06	1.10E-06	5.919901	0.0000
C(5)	3.64E-06	9.67E-07	3.759945	0.0002
C(6)	-2.64E-06	7.39E-07	-3.575569	0.0003
C(7)	1.04E-05	2.28E-06	4.550943	0.0000
C(8)	-8.03E-06	1.62E-06	-4.972743	0.0000
C(9)	1.39E-05	2.49E-06	5.590122	0.0000
C(10)	0.059566	0.007893	7.546435	0.0000
C(11)	0.052100	0.007282	7.154661	0.0000
C(12)	0.046822	0.008259	5.668999	0.0000
C(13)	0.058630	0.007199	8.144178	0.0000
C(14)	0.067051	0.007508	8.931132	0.0000
C(15)	0.112734	0.008091	13.93397	0.0000
C(16)	0.917973	0.010867	84.47647	0.0000
C(17)	0.928844	0.009860	94.20352	0.0000
C(18)	0.924802	0.010562	87.55899	0.0000
C(19)	0.908492	0.011498	79.01305	0.0000
C(20)	0.886249	0.011892	74.52704	0.0000
C(21)	0.829154	0.012741	65.07734	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. To aid you in interpreting the results, EViews provides a covariance specification section at the bottom of the estimation output that re-labels and transforms coefficients:

Covariance specification: Diagonal VECH
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	5.919903	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	7.546440	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	8.931139	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	94.20361	0.0000
B1(1,3)	0.924802	0.010562	87.55915	0.0000
B1(2,2)	0.908492	0.011498	79.01313	0.0000
B1(2,3)	0.886249	0.011892	74.52720	0.0000
B1(3,3)	0.829154	0.012741	65.07757	0.0000

* Coefficient matrix is not PSD.

The first line of this section states the covariance model used in estimation, in this case Diagonal VECH. 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 scrolling down to 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:

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 with specification identical to the one above. We'll select **Constant Conditional Correlation** this time as the **Model type** for estimation and leave the remaining settings as they are. The basic results:

System: UNTITLED
Estimation Method: ARCH Maximum Likelihood (BFGS / Marquardt steps)
Covariance specification: Constant Conditional Correlation
Date: 03/10/15 Time: 22:29
Sample: 12/31/1979 12/25/2000
Included observations: 1096
Total system (balanced) observations 3288
Presample covariance: backcast (parameter =0.7)
Convergence achieved after 48 iterations
Coefficient covariance computed using outer product of gradients

	Coefficient	Std. Error	z-Statistic	Prob.
C(1)	-0.000804	0.000450	-1.788285	0.0737
C(2)	-0.000232	0.000467	-0.497008	0.6192
C(3)	8.56E-05	0.000377	0.226828	0.8206
Variance Equation Coefficients				
C(4)	5.84E-06	1.30E-06	4.482923	0.0000
C(5)	0.062911	0.010085	6.238135	0.0000
C(6)	0.916958	0.013613	67.35993	0.0000
C(7)	4.89E-05	1.72E-05	2.836869	0.0046
C(8)	0.063178	0.012988	4.864468	0.0000
C(9)	0.772214	0.064005	12.06496	0.0000
C(10)	1.47E-05	3.11E-06	4.735844	0.0000
C(11)	0.104348	0.009262	11.26665	0.0000
C(12)	0.828535	0.017936	46.19303	0.0000
C(13)	0.571323	0.018238	31.32550	0.0000
C(14)	-0.403219	0.023634	-17.06082	0.0000
C(15)	-0.677329	0.014588	-46.43001	0.0000

Log likelihood	9593.125	Schwarz criterion	-17.40991
Avg. log likelihood	2.917617	Hannan-Quinn criter.	-17.45244
Akaike info criterion	-17.47833		

Note that this specification has only 12 free parameters in the variance equation, as compared with 18 in the previous model. The extended variance section represents the variance equation as,

```
GARCH(i) = M(i) + A1(i)*RESID(i)(-1)^2 + B1(i)*GARCH(i)(-1)
```

while the model for the covariance equation is:

```
COV(i,j) = R(i,j)*@SQRT(GARCH(i)*GARCH(j))
```

The lower portion of the output shows that the correlations, R(1, 2), R(1, 3), and R(2, 3) are 0.5713, -0.4032, and -0.6773, respectively:

Covariance specification: Constant Conditional Correlation
GARCH(i) = M(i) + A1(i)*RESID(i)(-1)^2 + B1(i)*GARCH(i)(-1)
COV(i,j) = R(i,j)*@SQRT(GARCH(i)*GARCH(j))

Transformed Variance Coefficients				
	Coefficient	Std. Error	z-Statistic	Prob.
M(1)	5.84E-06	1.30E-06	4.482923	0.0000
A1(1)	0.062911	0.010085	6.238137	0.0000
B1(1)	0.916958	0.013613	67.35994	0.0000
M(2)	4.89E-05	1.72E-05	2.836869	0.0046
A1(2)	0.063178	0.012988	4.864469	0.0000
B1(2)	0.772214	0.064005	12.06496	0.0000
M(3)	1.47E-05	3.11E-06	4.735844	0.0000
A1(3)	0.104348	0.009262	11.26665	0.0000
B1(3)	0.828536	0.017936	46.19308	0.0000
R(1,2)	0.571323	0.018238	31.32550	0.0000
R(1,3)	-0.403219	0.023634	-17.06082	0.0000
R(2,3)	-0.677329	0.014588	-46.43002	0.0000

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:

$$\begin{bmatrix} y_1 \\ y_2 \\ \mathrm{M} \\ y_M \end{bmatrix} = \begin{bmatrix} X_1 & 0 & \dots & 0 \\ 0 & X_2 & & \mathrm{M} \\ & & \mathrm{O} & 0 \\ 0 & \dots & 0 & X_M \end{bmatrix} \begin{bmatrix} \beta_1 \\ \beta_2 \\ \mathrm{M} \\ \beta_M \end{bmatrix} + \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \mathrm{M} \\ \epsilon_M \end{bmatrix} \tag{43.8}$$

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:

$$y = X\beta + \epsilon . \tag{43.9}$$

Under the standard assumptions, the residual variance matrix from this stacked system is given by:

$$V = \mathrm{E}(\epsilon\epsilon') = \sigma^2(I_M \otimes I_T). \tag{43.10}$$

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 $\sigma_{ij} = \mathrm{E}(\epsilon_{it}\epsilon_{jt})$ for all t. If the errors are contemporaneously uncorrelated, then, $\sigma_{ij} = 0$ for $i \neq j$, and we can write:

$$V = \mathrm{diag}(\sigma_1^2, \sigma_2^2, \ldots, \sigma_M^2) \otimes I_T \tag{43.11}$$

More generally, if the errors are heteroskedastic and contemporaneously correlated:

$$V = \Sigma \otimes I_T. \tag{43.12}$$

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:

$$V = \begin{pmatrix} \sigma_{11}\Sigma_{11} & \sigma_{12}\Sigma_{12} & \cdots & \sigma_{1M}\Sigma_{1M} \\ \sigma_{21}\Sigma_{21} & \sigma_{22}\Sigma_{22} & & \mathrm{M} \\ & & \mathrm{O} & \\ \sigma_{M1}\Sigma_{M1} & \cdots & & \sigma_{MM}\Sigma_{MM} \end{pmatrix} \tag{43.13}$$

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,

$$b_{LS} = (X'X)^{-1}X'y \tag{43.14}$$

and the variance estimator is given by:

$$\mathrm{var}(b_{LS}) = s^2(X'X)^{-1} \tag{43.15}$$

where s^2 is the residual variance estimate for the stacked system.

Weighted Least Squares

The weighted least squares estimator is given by:

$$b_{WLS} = (X'\hat{V}^{-1}X)^{-1}X'\hat{V}^{-1}y \tag{43.16}$$

where $\hat{V} = \mathrm{diag}(s_{11}, s_{22}, \ldots, s_{MM}) \otimes I_T$ is a consistent estimator of V, and s_{ii} is the residual variance estimator:

$$s_{ij} = (y_i - X_i b_{LS})'(y_j - X_j b_{LS}) / \max(T_i, T_j) \tag{43.17}$$

where the inner product is taken over the non-missing common elements of i and j. The max function in Equation (43.17) 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:

$$\mathrm{var}(b_{WLS}) = (X' \hat{V}^{-1} X)^{-1}. \tag{43.18}$$

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:

$$\begin{aligned} y_1 &= X_1 \beta_1 + \epsilon_1 \\ y_2 &= X_2 \beta_2 + \epsilon_2 \end{aligned} \tag{43.19}$$

If β_1 and β_2 are unrestricted, the WLS estimator given in Equation (43.18) yields:

$$b_{WLS} = \begin{bmatrix} ((X_1'X_1)/s_{11})^{-1}((X_1'y_1)/s_{11}) \\ ((X_2'X_2)/s_{22})^{-1}((X_2'y_2)/s_{22}) \end{bmatrix} = \begin{bmatrix} (X_1'X_1)^{-1} X_1' y_1 \\ (X_2'X_2)^{-1} X_2' y_2 \end{bmatrix}. \tag{43.20}$$

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:

$$b_{SUR} = (X'(\hat{\Sigma} \otimes I_T)^{-1} X)^{-1} X'(\hat{\Sigma} \otimes I_T)^{-1} y, \tag{43.21}$$

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 "Specifying AR Terms" on page 132) and estimates the following equation:

$$y_{jt} = X_{jt}\beta_j + \left(\sum_{r=1}^{p_j} \rho_{jr}(y_{j(t-r)} - X_{j(t-r)}) \right) + \epsilon_{jt} \tag{43.22}$$

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,

$$Y\Gamma_j + XB_j + \epsilon_j = 0 \tag{43.23}$$

or, alternatively:

$$y_j = Y_j\gamma_j + X_j\beta_j + \epsilon_j = Z_j\delta_j + \epsilon_j \tag{43.24}$$

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; Y_j is the matrix of endogenous variables not including y_j.

In the first stage, we regress the right-hand side endogenous variables y_j on all exogenous variables X and get the fitted values:

$$\hat{Y}_j = X(X'X)^{-1}X'Y_j. \tag{43.25}$$

In the second stage, we regress y_j on $\hat{Y}_j$ and X_j to get:

$$\hat{\delta}_{2SLS} = (\hat{Z}_j{}'\hat{Z}_j)^{-1}\hat{Z}_j{}'y. \tag{43.26}$$

where $\hat{Z}_j = (\hat{Y}_j, X_j)$. The residuals from an equation using these coefficients are used for form weights.

Weighted TSLS applies the weights in the second stage so that:

$$\hat{\delta}_{W2SLS} = (\hat{Z}_j{}'\hat{V}^{-1}\hat{Z}_j)^{-1}\hat{Z}_j{}'\hat{V}^{-1}y \tag{43.27}$$

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,

$$\hat{\delta}_{3SLS} = (Z(\hat{\Sigma}^{-1} \otimes X(X'X)^{-1}X')Z)^{-1} Z(\hat{\Sigma}^{-1} \otimes X(X'X)^{-1}X')y \tag{43.28}$$

where $\hat{\Sigma}$ has typical element:

$$s_{ij} = ((y_i - Z_i\hat{\gamma}_{2SLS})'(y_j - Z_j\hat{\gamma}_{2SLS}))/\max(T_i, T_j) \tag{43.29}$$

If you choose to iterate the weights, the current coefficients and residuals will be used to estimate $\hat{\Sigma}$.

Full Information Maximum Likelihood)

Following the discussion in Amemiya (1997), recall that we have

$$f(y_t, x_t, \beta) = \epsilon_t, \tag{43.30}$$

where y_t is a vector of endogenous variables, x_t is a vector of exogenous variables. The Full Information Maximum Likelihood (FIML) estimator finds the vector of parameters β by maximizing the likelihood under the assumption that ϵ_t is a vector of *i.i.d.* multivariate normal random variables with covariance matrix Σ.

Under the normality assumption, the log-likelihood is given by

$$\log L^* = -\frac{T}{2}\log|\Sigma| + \sum_{t=1}^{T} \log\left\|\frac{\partial f_t}{\partial y_t'}\right\| - \frac{1}{2}\sum_{t=1}^{T} f_t'\Sigma^{-1}f_t \tag{43.31}$$

where $f_t = f(y_t, x_t, \beta)$. Note that the log determinant of the derivatives of f_t captures the simultaneity in the system of equations.

For the unrestricted and diagonal restricted covariance variants of the model, we may use the first-order conditions for the variance parameters and rewrite the likelihood in concentrated form:

$$\log L = \sum_{t=1}^{T} \log \left\| \frac{\partial f_t}{\partial y_t'} \right\| - \frac{T}{2} \log \left(T^{-1} \sum_{t=1}^{T} f_t f_t' \right) \tag{43.32}$$

The diagonal restricted estimator replaces the off diagonal terms in the latter matrix with zeros. The corresponding FIML estimator maximizes the concentrated likelihood with respect to the β (or equivalently, the full likelihood with respect to β and the free parameters of Σ).

The FIML estimator for models with user restricted covariances maximizes the full likelihood in Equation (43.31) with respect to β given the user specified value for Σ.

The estimator for β is asymptotically normally distributed with coefficient covariance typically computed using the partitioned inverse of the outer-product of the gradient of the full likelihood (OPG) or the inverse of the negative of the observed Hessian of the concentrated likelihood. EViews employs the OPG covariance by default, but there is evidence that one should take seriously the choice of method (Calzolari and Panattoni, 1988). In addition, EViews offers a QML covariance computation that employs a Huber-White sandwich using both the OPG and the inverse negative Hessian.

Over the years, a number of approaches for FIML estimation have been proposed (see, for example, Parke 1982, Belsley 1980, Dagenais 1978, or Amemiya 1977). EViews offers standard BFGS, Newton-Raphson, and OPG/BHHH algorithms with various step methods in trust region form, as well as a simple implementation of BHHH with Marquardt and line search steps ("Optimization Algorithms" on page 1319). See Calzolari and Panattoni (1987) and Weihs, Calzolari, and Panattoni (1986) for simulation results for the performance of various estimators.

Whichever method you select, we encourage you to perform sensitivity analysis.

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:

$$\mathrm{E}(m(y, \theta)) = 0 . \tag{43.33}$$

The method of moments estimator is defined by replacing the moment condition (43.33) by its sample analog:

$$\left(\sum_t m(y_t, \theta) \right) / T = 0 . \tag{43.34}$$

However, condition (43.34) 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:

$$\sum_t m(y_t, \theta) A(y_t, \theta) m(y_t, \theta) \tag{43.35}$$

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 (43.33) as an orthogonality condition between the residuals of a regression equation, $u(y, \theta, X)$, and a set of instrumental variables, Z, so that:

$$m(\theta, y, X, Z) = Z' u(\theta, y, X) \tag{43.36}$$

For example, the OLS estimator is obtained as a GMM estimator with the orthogonality conditions:

$$X'(y - X\beta) = 0 . \tag{43.37}$$

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," beginning on page 101 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 $A = \hat{\Omega}^{-1}$, where $\hat{\Omega}$ is the estimated long-run 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:

$$\hat{\Omega}_W = \hat{\Gamma}(0) = \frac{1}{T-k}\left(\sum_{t=1}^{T} Z_t' u_t u_t' Z_t\right) \tag{43.38}$$

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,

$$\hat{\Omega}_{HAC} = \hat{\Gamma}(0) + \left(\sum_{j=1}^{T-1} k(j, q)(\hat{\Gamma}(j) + \hat{\Gamma}'(j)) \right) \tag{43.39}$$

where:

$$\hat{\Gamma}(j) = \frac{1}{T-k} \left(\sum_{t=j+1}^{T} Z_{t-j}' u_{t-j} u_t' Z_t \right). \tag{43.40}$$

You also need to specify the *kernel function* κ and the *bandwidth* q.

Kernel Options

The kernel function 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:

$$\kappa(x) = \begin{cases} 1 - x & 0 \le x \le 1 \\ 0 & \text{otherwise} \end{cases} \tag{43.41}$$

while the quadratic spectral (QS) kernel is given by:

$$k(j/q) = \frac{25}{12(\pi x)^2} \left(\frac{\sin(6\pi x/5)}{6\pi x/5} - \cos(6\pi x/5) \right) \tag{43.42}$$

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:

$$q = \text{int}(4(T/100)^{2/9}) \tag{43.43}$$

where int( ) 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 the rule:

$$q = \begin{cases} \text{int}(1.1447(\hat{\alpha}(1)\,T)^{1/3}) & \text{for the Bartlett kernel} \\ 1.3221(\hat{\alpha}(2)\,T)^{1/5} & \text{for the QS kernel} \end{cases} \tag{43.44}$$

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 (43.36) and estimate the autocorrelation coefficients $\hat{\rho}_i$ and the residual variances $\hat{\sigma}_i^2$ for $i = 1, 2, \ldots, p$. Then $\hat{\alpha}(1)$ and $\hat{\alpha}(2)$ are estimated by:

$$\begin{aligned} \hat{\alpha}(1) &= \left(\sum_{i=1}^{zn} \frac{4\hat{\sigma}_i^4\hat{\rho}_i^2}{(1-\hat{\rho}_i)^6(1+\hat{\rho}_i)^2}\right) \Big/ \left(\sum_{i=1}^{zn} \frac{\hat{\sigma}_i^4}{(1-\hat{\rho}_i)^4}\right) \\ \hat{\alpha}(2) &= \left(\sum_{i=1}^{zn} \frac{4\hat{\sigma}_i^4\hat{\rho}_i^2}{(1-\hat{\rho}_i)^8}\right) \Big/ \left(\sum_{i=1}^{zn} \frac{\hat{\sigma}_i^4}{(1-\hat{\rho}_i)^4}\right) \end{aligned} \tag{43.45}$$

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,

$$\hat{\alpha}(p) = \left(\frac{l'F(p)l}{l'F(0)l}\right)^2 \tag{43.46}$$

where l is a vector of ones and:

$$F(p) = (p-1)\hat{\Gamma}(0) + \sum_{i=1}^{L} i^p(\hat{\Gamma}(i) + \hat{\Gamma}'(i)), \tag{43.47}$$

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:

$$L = \text{int}(4(T/100)^a) \tag{43.48}$$

where $a = 2/9$ for the Bartlett kernel and $a = 4/25$ for the quadratic spectral kernel.

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:

$$m_t = A m_{t-1} + v_t. \tag{43.49}$$

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 long-run variance of the residuals v_t computed using any of the above methods. The GMM estimator is then found by minimizing the criterion function:

$$u' Z \hat{\Omega}^{-1} Z' u \tag{43.50}$$

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:

$$l_t = -\frac{1}{2} m \log(2\pi) - \frac{1}{2}\log(|H_t|) - \frac{1}{2}\epsilon_t' H_t^{-1} \epsilon_t \tag{43.51}$$

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:

$$l_t = \log\left\{\frac{\Gamma\left(\frac{v+m}{2}\right) v^{m/2}}{(v\pi)^{m/2}\Gamma\left(\frac{v}{2}\right)(v-2)^{m/2}}\right\} - \frac{1}{2}\log(|H_t|) - \frac{1}{2}(v+m)\log\left[1 + \frac{\epsilon_t' H_t^{-1}\epsilon_t}{v-2}\right] \tag{43.52}$$

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 VECH, Constant Conditional Correlation (CCC), and Diagonal BEKK.

Diagonal VECH

Bollerslev, *et. al* (1988) introduce a restricted version of the general multivariate VECH model of the conditional covariance with the following formulation:

$$H_t = \Omega + A \bullet \epsilon_{t-1}\epsilon_{t-1}' + B \bullet H_{t-1} \tag{43.53}$$

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 matri-

ces 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:

$$(H_t)_{ij} = (\Omega)_{ij} + (A_{ij})\epsilon_{jt-1}\epsilon_{it-1} + (\boldsymbol{B})_{ij}(H_{t-1})_{ij} \tag{43.54}$$

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:

$$H_t = \tilde{\Omega}\Omega' + \tilde{A}A' \bullet \epsilon_{t-1}\epsilon_{t-1}' + \tilde{B}B' \otimes H_{t-1} \tag{43.55}$$

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:

$$B = b \cdot ii' \tag{43.56}$$

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:

$$\Omega = \Omega_0 \bullet (ii' - A - B) \tag{43.57}$$

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 $3N((N+1)/2)$.

Constant Conditional Correlation (CCC)

Bollerslev (1990) specifies the elements of the conditional covariance matrix as follows:

$$\begin{aligned} h_{iit} &= c_i + a_i\epsilon_{it-1}^2 + d_i I_{it-1}^- \epsilon_{it-1}^2 + b_i h_{iit-1} \\ h_{ijt} &= \rho_{ij}\sqrt{h_{iit}h_{jjt}} \end{aligned} \tag{43.58}$$

Restrictions may be imposed on the constant term using variance targeting so that:

$$c_i = \sigma_0^2(1 - a_i - b_i) \tag{43.59}$$

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.

$$h_{iit} = c_i + a_i\epsilon_{it-1}^2 + d_i I_{it-1}^- \epsilon_{it-1}^2 + b_i h_{iit-1} + e_i x_{1t} + g x_{2t} \tag{43.60}$$

Diagonal BEKK

BEKK (Engle and Kroner, 1995) is defined as:

$$H_t = \Omega\Omega' + A\epsilon_{t-1}\epsilon_{t-1}{}'A' + BH_{t-1}B' \tag{43.61}$$

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 VECH model where the coefficient matrices are rank one matrices. For convenience, EViews provides an option to estimate the Diagonal VECH model, but display the result in Diagonal BEKK form.

References

Amemiya, Takeshi (1977). "The Maximum Likelihood and the Nonlinear Three-Stage Least Squares Estimator in the General Nonlinear Simultaneous Equation Model," *Econometrica*, 45, 955–966.

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.

Belsley, David (1980). "On the Efficient Computation of the Nonlinear Full-information Maximum-likelihood Estimator, *Journal of Econometrics*, 14, 203–225.

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, 116–131.

Calzolari, Giorgio and Lorenzo Panattoni (1987). "Computational Efficiency of FIML Estimation," *Journal of Econometrics*, 36, 299–310.

Calzolari, Giorgio and Lorenzo Panattoni (1988). "Alternative Estimators of FIML Covariance Matrix: A Monte Carlo Study, *Econometrica*, 56, 701–714.

Dagenais, Marcel G. (1978). "The Computation of FIML Estimates as Iterative Generalized Least Squares Estimates in Linear and Nonlinear Simultaneous Equations Models," *Econometrica*, 46, 1351–1362.

Ding, Zhuanxin 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.

Parke, William R. (1982). "An Algorithm for FIML and 3SLS Estimation of Large Nonlinear Models", *Econometrica*, 50, 81–95.

Weis, C. Calzolari, G., and L. Panattoni (1987). "The Behavior of Trust-Region Methods in FIML-Estimation," Computing, 38, 89–100.

Chapter 44. 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 reduced form VAR approach sidesteps the need for structural modeling by treating every endogenous variable in the system as a function of p-lagged values of all of the endogenous variables in the system.

Below, we offer an abbreviated description of important features of the model. There are countless treatments of the basics of VAR analysis that one can consult for additional detail (see, for example, Lütkepohl, 2006).

We may write the stationary, k-dimensional, VAR(p) process as

$$y_t = A_1 y_{t-1} + \cdots + A_p y_{t-p} + C x_t + \epsilon_t \tag{44.1}$$

where

- $y_t = (y_{1t}, y_{2t}, \ldots, y_{Kt})'$ is a $k \times 1$ vector of endogenous variables,
- $x_t = (x_{1t}, x_{2t}, \ldots, x_{dt})'$ is a $d \times 1$ vector of exogenous variables,
- $A_1, \ldots, A_p$ are $k \times k$ matrices of lag coefficients to be estimated,
- C is a $k \times d$ matrix of exogenous variable coefficients to be estimated,
- $\epsilon_t = (\epsilon_{1t}, \epsilon_{2t}, \ldots, \epsilon_{Kt})'$ is a $k \times 1$ white noise innovation process, with $E(\epsilon_t) = 0$, $E(\epsilon_t \epsilon_t{}') = \Sigma_\epsilon$, and $E(\epsilon_t \epsilon_s{}') = 0$ for $t \neq s$.

The last statement implies that the vector of innovations are contemporaneously correlated with full rank matrix Σ_ϵ, but are uncorrelated with their leads and lags of the innovations

and (assuming the usual x_t orthogonality) uncorrelated with all of the right-hand side variables.

Let the $(pk + d) \times 1$ vector

$$Z_t = (y_{t-1}', \ldots, y_{t-p}', x_t')' \tag{44.2}$$

represent all of the period t regressors in the VAR. Then for observations $t = 1, \ldots, T$, we may write this model in compact system form as:

$$Y = BZ + E \tag{44.3}$$

where $Y = (y_1, y_2, \ldots, y_T)$ and $E = (\epsilon_1, \epsilon_2, \ldots, \epsilon_T)$ are $k \times T$ matrices of endogenous variables and innovations, and

$$\begin{aligned} B &= (A_1, A_2, \ldots, A_p, C) \\ Z &= (Z_1, Z_2, \ldots, Z_T) \end{aligned} \tag{44.4}$$

are the $k(pk + d)$ matrix of system coefficients and the $(pk + d) \times T$ matrix of regressor data, respectively.

In stacked form, we have

$$\tilde{y} = (Z' \otimes I_K)\beta + \tilde{\epsilon} \tag{44.5}$$

where $\tilde{y} = \mathrm{vec(Y)}$, $\beta = \mathrm{vec(B)}$, and $\tilde{\epsilon} = \mathrm{vec(E)}$, where $E(\tilde{\epsilon}\tilde{\epsilon}') = (I_K \otimes \Sigma_\epsilon)$.

Since only lagged values of the endogenous variables appear on the right-hand side of the VAR equations Equation (44.1) and the innovations are assumed to be uncorrelated with lagged innovations and the exogenous regressors, standard orthogonality conditions hold and OLS yields consistent estimates.

Furthermore, even though the innovations ϵ_t may be contemporaneously correlated, all of the equations in the system have identical Z_t regressors so that OLS is both equivalent to GLS and efficient. Accordingly, estimation of the standard VAR model in EViews is performed using simple OLS applied to each equation. (Later, when we describe estimation of restricted VAR models, we relax the identical regressors assumption so that OLS is no longer efficient.)

Applying least squares estimation to the stacked representation yields the least squares estimator

$$\hat{\beta} = ((ZZ')^{-1} Z \otimes I_K)\tilde{y} \tag{44.6}$$

which has covariance matrix

$$V(\hat{\beta}) = (ZZ')^{-1} \otimes \Sigma_\epsilon \tag{44.7}$$

To obtain an estimator of the covariance matrix we require an estimate of Σ_ϵ, which is typically obtained using the d.f. corrected residual moment estimator:

$$\hat{\Sigma}_{\epsilon} = \frac{\hat{E}\hat{E}'}{T-(pk+d)} \tag{44.8}$$

where $\hat{E} = Y - \hat{B}Z$ for $\hat{\beta} = \text{vec}(\hat{B})$

As an example, suppose that industrial production (IP) and money supply (M1) are jointly determined by a VAR(2) and let a constant be the only exogenous variable. Then the VAR may be written as:

$$\begin{aligned} IP_t &= a_{11}IP_{t-1} + a_{12}M1_{t-1} + b_{11}IP_{t-2} + b_{12}M1_{t-2} + c_1 + \epsilon_{1t} \\ M1_t &= a_{21}IP_{t-1} + a_{22}M1_{t-1} + b_{21}IP_{t-2} + b_{22}M1_{t-2} + c_2 + \epsilon_{2t} \end{aligned} \tag{44.9}$$

where a_{ij}, b_{ij}, c_i are the parameters to be estimated.

Alternately, in terms of Equation (44.1), we have

$$y_t = A_1 y_{t-1} + A_2 y_{t-2} + C + \epsilon_t \tag{44.10}$$

where $y_t = (IP_t, M1_t)'$ and $\epsilon_t = (\epsilon_{1t}, \epsilon_{2t})'$ are 2×1, A_1 and A_2 are 2×2 matrices, and C is 2×1.

Further, we may write the specification in system or stacked form using Equation (44.3), Equation (44.4), and Equation (44.5) with $Z_t = (IP_{t-1}, M1_{t-1}, IP_{t-2}, M1_{t-2}, 1)$ and $B = (A_1, A_2, C)$.

Estimates of the coefficients of the VAR may be obtained by regressing IP on an intercept and two lags of IP and M1 and by regressing M1 on an intercept and two lags of IP and M1.

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 to display the estimation dialog

The **Basics** tab of the VAR Specification dialog will prompt you to define the structure of your VAR.

We begin with a simple reduced form VAR. You should fill out the dialog with the appropriate information:

- Select the VAR type: **Standard VAR.**
- Set the estimation sample.
- Enter a list of k **Endogenous variables** in the corresponding edit field. You may list the series individually or you may include one or more series in a group object and enter the group name.

  Here we have listed M1, IP, and TB3 as endogenous series
- Enter the lag specification in the **Lag Intervals for Endogenous** edit box. This information is entered in pairs where 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.

  Through use of multiple lag pairs you may place zero restrictions on particular lag coefficient matrices A_i, as desired. Simply add any number of lag intervals, all entered in pairs, omitting those lags you wish to restrict. For example, he lag specification:

  ```
  2 4 6 9 12 12
  ```

uses lags 2–4, 6–9, and 12.

Note that restrictions entered in this fashion restrict the entire A_i matrix to be zero. Below, we describe tools that allow for finer discrimination.

- Enter the specification for the exogenous series in the appropriate edit box, using group names if convenient.

  Here, we have used the special series C to indicate that the VAR has a single constant exogenous term.

For the moment, we will ignore the VAR Restrictions tab of the dialog. Once you have specified your VAR, click on **OK** to have EViews estimate the coefficient matrices using least squares.

Estimation Output

Following estimation, EViews will estimate the model EViews will display the estimation results in the VAR window.

	IP	M1	TB3
IP(-1)	1.253934	0.253215	0.095984
	(0.05401)	(0.17769)	(0.05021)
	[23.2147]	[1.42501]	[1.91170]
IP(-2)	-0.187774	-0.230187	0.015590
	(0.08557)	(0.28149)	(0.07954)
	[-2.19448]	[-0.81774]	[0.19601]
IP(-3)	-0.003780	-0.153515	-0.173824
	(0.08556)	(0.28146)	(0.07953)
	[-0.04418]	[-0.54543]	[-2.18570]
IP(-4)	-0.059071	0.160097	0.071298
	(0.05343)	(0.17578)	(0.04967)
	[-1.10554]	[0.91080]	[1.43551]
M1(-1)	0.018813	1.305608	0.069550
	(0.01538)	(0.05060)	(0.01430)

Recall that our VAR specification has three ($k = 3$) endogenous variables, IP, M1, and TB3, the exogenous intercept C ($d = 1$), and includes lags 1 to 4 ($p = 4$). Thus, there are ($kp + d = 13$) regressors in each of the three equations in the VAR.

As you can see, EViews displays the coefficient results in table. Each column in the table corresponds to an equation in the VAR, and each row corresponds to a regressor in the equa-

tion. Note that the regressors are grouped by variable, so that all of the lags for the first variable, here IP, are followed by all of the lags for the second variable, M1, and so on. The exogenous variables appear last.

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, the standard error is 0.05021, and the corresponding *t*-statistic is 1.91170.

EViews displays additional information below the coefficient results. This information is divided into two parts:

- The first part of the additional output presents standard OLS regression summary statistics at the bottom of the column for the corresponding equation (recall that for the standard VAR each equation is estimated separately by OLS).
- The second part consists of summary statistics for the VAR system as a whole. These statistics include the determinant of the residual covariance, log-likelihood and associated information criteria, and number of coefficients:

Vector Autoregression Estimates

TB3(-4)	0.067962	0.243272	-0.008569
	(0.05850)	(0.19245)	(0.05438)
	[1.16175]	[1.26410]	[-0.15758]
C	0.162892	-0.992798	-0.199477
	(0.17343)	(0.57055)	(0.16121)
	[0.93922]	[-1.74008]	[-1.23735]
R-squared	0.999221	0.999915	0.968018
Adj. R-squared	0.999195	0.999912	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.)		0.289218	
Determinant resid covariance		0.259637	
Log likelihood		-1318.390	
Akaike information criterion		7.377118	
Schwarz criterion		7.791290	
Number of coefficients		39	

A couple of comments on the calculation of the system statistics.

- The determinant of the residual covariance (degree-of-freedom adjusted) is computed as:

$$\left|\hat{\Sigma}_\epsilon\right| = \det\left(\frac{1}{T-(pk+d)}\sum_t \hat{\epsilon}_t\hat{\epsilon}_t'\right) \tag{44.11}$$

where you will recall that $(pk+d)$ is the number of parameters per equation in the VAR. The unadjusted calculation ignores the number of parameters and divides by T.

- The log likelihood value is computed assuming a multivariate normal (Gaussian) distribution as:

$$l = -\frac{T}{2}\{k(1+\log 2\pi)+\log\left|\hat{\Sigma}_\epsilon\right|\} \tag{44.12}$$

using the unadjusted determinant.

- The two information criteria are computed as:

$$\begin{aligned} AIC &= -2l/T+2n/T \\ SC &= -2l/T+n\log T/T \end{aligned} \tag{44.13}$$

where $n = k(pk+d)$ is the total number of estimated parameters in the VAR. The 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. Bear in mind when comparing the IC with other sources that some define the AIC/SC differently, either omitting the "inessential" constant terms from the likelihood, or by not dividing by T (see also Appendix E. "Information Criteria," on page 1335 for additional discussion of information criteria).

VARs With Linear Constraints

The basic k-variable VAR(p) specification has $k(pk+d)$ coefficients so that even moderate sized VARs require estimation of a large number of parameters. When VARs are applied to macroeconomic data with limited sample sizes, model over-parameterization is a frequent problem as there are too few observations to estimate precisely the VAR parameters.

EViews offers two approaches to handling this over-parameterization problem:

- One approach, Bayesian estimation, is described in detail below ("Bayesian VAR Models" on page 895).
- Alternately, you may reduce the number of free parameters by imposing linear constraints on the elements of B.

We outline here EViews's tools for imposing linear restrictions. Note that we previously explored a variant of the constraint approach where specification of the VAR lag structure using multiple lag pairs sets a subset of lag matrices A_j to zero ("Estimating a VAR in EViews" on page 835). In contrast, the more general methods described in this section per-

mit finely targeted restriction on the parameters, allowing us to incorporate *a priori* information about the parameters of the model which does not fit into the lag exclusion framework.

Briefly, to echo the discussion in Lütkepohl (2006), suppose we have linear restrictions of the form:

$$\beta = R\gamma + r \tag{44.14}$$

where $\beta = \text{vec}(B)$ is the $k(pk+d)\times 1$ vector of VAR coefficients, R is a $k(pk+d)\times M$ restriction matrix, γ is a $M\times 1$ vector of unconstrained parameters and r is a $k(pk+d)\times 1$ vector of known constants.

Note that while this is not the most common form for expressing linear restrictions, the conventional (S, s) form of linear restrictions, $S\beta = s$, may readily be transformed to the (R, r) form (see Lütkepohl 2006, p. 195).

Substituting Equation (44.14) into Equation (44.5) produces an unrestricted regression specification

$$\begin{aligned} y &= (Z' \otimes I_K)(R\gamma + r) + \tilde{\epsilon} \\ z &= (Z' \otimes I_K)R\gamma + \tilde{\epsilon} \end{aligned} \tag{44.15}$$

where $z = \tilde{y} - (Z' \otimes I_K)r$.

The OLS estimator for γ is given by

$$\hat{\gamma}_{OLS} = (R'(ZZ' \otimes I_K)R)^{-1}R'(Z \otimes I_K)z \tag{44.16}$$

and a estimator of the variance of $\hat{\gamma}_{OLS}$ is given by

$$\hat{V}(\hat{\gamma}_{OLS}) = (R'(ZZ' \otimes I_K)R)^{-1}R'(ZZ' \otimes \hat{\Sigma}_\epsilon)R(R'(ZZ' \otimes I_K)R)^{-1} \tag{44.17}$$

for the consistent estimator

$$\hat{\Sigma}_\epsilon = \frac{\hat{E}\hat{E}'}{T} \tag{44.18}$$

Note that we employ the unadjusted residual variance matrix estimator since the presence of restrictions produces ambiguity in the finite sample correction.

Efficiency gains may be realized by accounting for the contemporaneous correlation in the errors. For $\hat{\Sigma}_\epsilon$, a consistent estimator of Σ_ϵ the estimated GLS estimator is given by

$$\hat{\gamma}_{GLS} = (R'(ZZ' \otimes \hat{\Sigma}_\epsilon^{-1})R)^{-1}R'(Z \otimes \hat{\Sigma}_\epsilon^{-1})z \tag{44.19}$$

Note that any consistent estimator for Σ_ϵ may be employed. By default, EViews will iterate over estimates of $\hat{\gamma}_{GLS}$ and the $\hat{\Sigma}_\epsilon$ obtained using the current residuals until convergence, or the maximum number of iterations is reached. Alternately, you may elect to perform only a single GLS iteration.

A consistent estimator of the coefficient covariance is given by

$$\hat{V}(\hat{\gamma}_{GLS}) = (R'(ZZ' \otimes \hat{\Sigma}_{\epsilon}^{-1})R)^{-1} \tag{44.20}$$

The corresponding estimators and covariances for the full coefficient vector β are then

$$\begin{aligned} \hat{\beta}_{OLS} &= R\hat{\gamma}_{OLS} + r \\ \hat{V}(\hat{\beta}_{OLS}) &= R(R'(ZZ' \otimes I_K)R)^{-1}R'(ZZ' \otimes \hat{\Sigma}_{\epsilon})R(R'(ZZ' \otimes I_K)R)^{-1}R' \end{aligned} \tag{44.21}$$

and

$$\begin{aligned} \hat{\beta}_{GLS} &= R\hat{\gamma}_{GLS} + r \\ \hat{V}(\hat{\beta}_{GLS}) &= R(R'(ZZ' \otimes \hat{\Sigma}_{\epsilon}^{-1})R)^{-1}R' \end{aligned} \tag{44.22}$$

Bear in mind that by construction, $\hat{V}(\hat{\beta}_{GLS})$ has reduced rank.

Estimating a VAR with Linear Restrictions in EViews

Select **Quick/Estimate VAR...** or type `var` in the command window to display the estimation dialog.

The **Basics** tab of the VAR Specification dialog outlines the basic specification, and should be filled out as for a basic specification ("Estimating a VAR in EViews" on page 835).

Click on the **VAR Restrictions** tab to specify your restrictions:

The **Restrictions** section on the left should be used to select the VAR elements that you wish to restrict. You may click on the entries for the lag matrices (**L1**, **L2**, **L3**, **L4**) and the vectors of coefficients associated with each exogenous variable (**C**) to select an element to restrict. The right side of the dialog will change to show the current settings for the selected element.

- If the **Manual** method is selected, you will place restrictions on the selection by editing its matrix representation in the dialog (rows of the matrix represent VAR equations while columns represent the endogenous variable). The ordering of the matrix elements follows the order of the endogenous variables in the original specification. Elements of the matrix or vector that are NA will be estimated while the remaining elements will be restricted to the specified value.

  Here, we have selected the matrix for the first lag of the endogenous variables (**L1**) and set two elements to zero so that the lag of the first endogenous variable has no effect on the second and third endogenous variables.

- If the **Workfile matrix** method is selected, you should enter the name of a matrix object in the workfile containing the restriction matrix for the selection. The matrix should be of the correct dimension and should have NAs for unrestricted elements and numeric values for restricted elements.

In settings where restricting a large number individual elements is inconvenient, or where you wish to impose restrictions across lag or exogenous variable coefficients, you may find it more convenient to specify restrictions using text expressions. In this case, click on the **Text** node of the **Restrictions** tree to display the **Restriction Text** edit field.

Text expressions allow you to use a function-like syntax to employ text to specify restrictions on one or more matrix elements. The text restrictions will use the following "@" keywords that you may use to refer to individual coefficient matrix elements,

@l#(r, c)	Element *(r, c)* of the lag # coefficient matrix.
@e#(r)	Element *r* of the exogenous variable # coefficient vector.
@e(X, r)	Element *r* of the exogenous variable X coefficient vector

Note that in this syntax, the canonical names ("L#", "E#", "E(X)") that refer to lag matrices and exogenous variable vectors are preceded by "@" to avoid ambiguity.

For example, we may have:

```
@L1(1,1) = 0
@L2(2,2) = @L1(3,3) / 2
@L2(1,1) + @L4(2,1) = 1
@E(C, 1) = 0
@E(X, 2) = @E(C, 2)
@E1(1) + @E1(2) = 1
```

In addition, you may use text expressions to refer to parts of lag coefficient matrices and to impose specialized restrictions,

@vec(W) = n_1 , n_2 , n_3 , ...	Restricts all elements of matrix W similar to a pattern matrix. Element ordering matches the vectorization of the matrix, *i.e.*, the elements of the first column, followed by the second column, followed by the third column, *etc.*
@diag(W)	Restricts W to be a diagonal matrix, *i.e.*, off-diagonal elements are zero. The diagonal elements are unrestricted.
@diag(W) = n	Restricts W to be a diagonal matrix with elements on the diagonal restricted to be n.
@lower(W)	Restricts W to be a lower triangular matrix, *i.e.*, elements above the diagonal are zero.
@unitlower(W)	Restricts W to be a unit lower triangular matrix, *i.e.*, elements above the diagonal are zero and elements on the diagonal are one.
@upper(W)	Restricts W to be an upper triangular matrix, *i.e.*, elements below the diagonal are zero.

@unitupper(W)	Restricts W to be a unit upper triangular matrix, *i.e.*, elements below the diagonal are zero and elements on the diagonal are one.
@row(W, r) = n	Restricts the elements in row r of W to be n.
@col(W, c) = n	Restricts the elements in column c of W to be n.

where W is a reference to a canonical matrix name (*e.g.*, "L1", "L3").

For example:

```
@row(L1, 2) = 0
@lower(L5)
```

You may specify your restrictions with any combination of direct assignment or text expressions. EViews will analyze the specification and determine if there are any restrictions that are incompatible.

If restrictions have been applied to any elements, the left side of the dialog will note this fact with the line of text "* restrictions applied", and the **Clear All** button will be enabled. The button allows you to easily remove all of the current restrictions.

When restrictions have been applied, the **Basics** tab of the dialog will change to offer additional options. In the newly displayed **Restriction estimation section** in the bottom left o the dialog, you will be prompted to specify the method of estimating your restricted coefficients:

In you do not select the **Iterate GLS weighting** option, EViews will estimate the parameters of the model using one-step GLS estimation. If you do select **Iterate GLS weighting you will be prompted** to specify the iteration and convergence properties of the estimation. To estimate with OLS, you may select **Iterate GLS weighting** and specify a **Max Iterations** of 0.

Estimation Output (Linear Restrictions)

Following estimation, EViews will display the results of restricted estimation. For the most part, the results are as described in "Estimation Output" on page 837, but there are some differences of note.

	IP	M1	TB3
IP(-1)	1.230677	0.000000	0.000000
	(0.05189)	---	---
	[23.7168]		
IP(-2)	-0.153998	0.070473	0.156826
	(0.08299)	(0.17463)	(0.05084)
	[-1.85569]	[0.40354]	[3.08445]
IP(-3)	-0.013155	-0.186360	-0.214413
	(0.08415)	(0.27570)	(0.08047)
	[-0.15634]	[-0.67595]	[-2.66455]
IP(-4)	-0.059631	0.144715	0.069240
	(0.05257)	(0.17298)	(0.05058)
	[-1.13441]	[0.83661]	[1.36903]
M1(-1)	0.003315	1.341040	0.000000
	(0.01478)	(0.04937)	---
	[0.22426]	[27.1643]	

First, the top of the estimation output shows that the VAR estimates were estimated with restrictions, and provides a text description of the restrictions. Here we see that in our illustration, we have placed zero restrictions on the first lag coefficients for IP in the M1 equation, and for IP and M1 in the TB3 equation.

Just below the restrictions specification, EViews displays information about the restricted estimation method. Here, we see that EViews estimated the VAR using iterated GLS and that the coefficients and GLS weight matrix converged after 11 iterations.

The coefficients display shows the full set of VAR coefficient estimates with restrictions imposed as appropriate. In cases, where the estimated standard error of the coefficient is zero, the display will show "---" and the *t*-statistic is suppressed.

As for unrestricted VARs, the bottom of the output show additional information on the individual equations and summary statistics for the VAR as a whole:

As before, there are standard OLS regression summary statistics at the bottom of the column for the corresponding equation. Since linearly restricted VAR regression is not computed on an equation by equation basis, EViews only displays a subset of the equation specific statistics that are available for unrestricted VAR estimation.

The remaining output consists of summary statistics for the VAR system as a whole. These statistics include the determinant of the residual covariance (non-d.f. corrected), log-likelihood and associated information criteria, the number of coefficients, and number of restrictions.

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 43. "System Estimation," beginning on page 791.

Residual Views

You may use the entries under the **Residuals** and **Structural Residuals** menus to examine the residuals of the estimated VAR in graph or spreadsheet form, or you may examine the covariance and correlation matrix of those residuals.

The views listed under **Residuals** will display results using the raw residuals from the estimated VAR.

Alternately, you may display the **Structural Residuals** views to examine the these transformed estimated residuals. If the $\hat{\epsilon}_t$ are the ordinary residuals, we may plot the structural residuals $\hat{u}_t$ based on factor loadings F,

$$\hat{\epsilon}_t = F\hat{u}_t \tag{44.23}$$

or based on weights W,

$$W\hat{\epsilon}_t = \hat{u}_t \tag{44.24}$$

When producing results for the **Structural Residuals** views, you will be prompted to choose a transformation.

Note that there is a corresponding procs which will save the residuals series in the workfile (**Proc/Make Residuals** and **Proc/Make Structural Residuals...**).

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

EViews offers several views for investigating the lag structure of your equation.

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.

Note: for VAR models with linear restrictions, EViews will only perform tests involving endogenous variable lags for which there are no restrictions.

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).

Note: for VAR models with linear restrictions, EViews will only perform tests involving endogenous variable lags for which there are no restrictions.

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:

$$LR = (T - m)\{\log|\Sigma_{\epsilon,\, l-1}| - \log|\Sigma_{\epsilon,\, l}|\} \sim \chi^2(k^2) \tag{44.25}$$

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).

Note: for VAR models with linear restrictions, EViews perform lag length tests using models in which the VAR linear restrictions are not applied.

Residual Tests

You may use these views to examine the properties of the residuals from your estimated VAR.

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 approximate asymptotic standard errors of the lagged correlations (ignoring coefficient estimation and 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 if there are no coefficient restrictions, and $k^2 h - r$ where r contains adjustments for the restrictions.

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 $h < p$.

Note that adjustments are made to the computation of the p-value of these statistics when the model is a VECM (Brüggemann, Lütkepohl, and Saikkonen, 2006).

Autocorrelation LM Test

Reports the multivariate LM test statistics for residual serial correlation up to the specified order. A Breusch-Godfrey test statistic for autocorrelation at 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 LR version of this LM test. The form of this statistic employs the Edgeworth expansion correction (Edgerton and Shukur 1999). 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.

A variant of this test tests for autocorrelation for lags 1 to h. The test modifies the LM statistic above by including all of the modified lagged residual regressors from $s = 1, \ldots, h$. Under the null hypothesis, the LM statistic is asymptotically χ^2 with hk^2 degrees of freedom.

In addition to the LR version of the test, EViews computes the Rao F-test version of the LM statistic as described Edgerton and Shukur (1999) since their simulations suggest it performs best among the many variants they consider.

Normality Test

This view 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 853 for additional discussion of the need for orthogonalization).

Let P be a $k \times k$ factorization matrix such that:

$$v_t = Pu_t \sim N(0, I_k) \tag{44.26}$$

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:

$$\sqrt{T}\begin{bmatrix} m_3 \\ m_4 - 3 \end{bmatrix} \to N\left(0, \begin{bmatrix} 6I_k & 0 \\ 0 & 24I_k \end{bmatrix}\right) \tag{44.27}$$

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 869, 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:

$$T\left\{\frac{m_3^2}{6}+\frac{(m_4-3)^2}{24}\right\} \tag{44.28}$$

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:

$$\begin{aligned}\lambda_3 &= Tm_3{}'m_3/6 \rightarrow \chi^2(k)\\ \lambda_4 &= T(m_4-3)'(m_4-3)/24 \rightarrow \chi^2(k)\\ \lambda &= \lambda_3+\lambda_4 \rightarrow \chi^2(2k).\end{aligned} \tag{44.29}$$

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.

Cointegration Test

This view performs the Johansen cointegration test for the variables in your VAR. See "Johansen Cointegration Test," on page 1245 for a description of the basic test methodology.

Note that Johansen cointegration tests may also be performed from a Group object, however, tests performed using the latter do not permit you to impose identifying restrictions on the cointegrating vector.

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:

$$u_t = P\epsilon_t \sim (0, D) \tag{44.30}$$

where D is a *diagonal* covariance matrix. As explained below, EViews provides several options for the choice of P which produce a diagonal matrix.

To compute and display impulse responses, click on **View/Impulse Response...** on the toolbar of an estimated VAR object to display the two tab dialog interface:

There are sections for **Display information** and **Response standard errors** on the first tab, and settings for **Impulse Definition** on the second tab.

Display Information

In the **Display information information** section you will define the **Impulses** and **Responses** you initially wish to display, the **Horizon length** over which you wish to compute those impulse responses, whether you wish to compute cumulative responses using the **Accumulate responses** checkbox, and a **Response standard errors** section for controlling the computation of standard errors and confidence intervals.

Note that the **Impulses** and **Responses** edit fields are for specifying the initial results to display. By default, these fields are filled out with the names of all of the variables in the VAR, but you will later be able to modify the display interactively. Note also when you specifying the **Horizon length** that in EViews, the original impulse is numbered as period 1 so that the default is to display the impulse and following 9 periods.

Response Standard Errors and Confidence Intervals

The **Method** dropdown in the **Response standard errors** group offers several method for computing IR standard errors (SEs). By extension, the latter are used to derive the corresponding confidence intervals. In particular, the following choices are available:

- **None**—no standard errors and no CIs are produced.
- **Analytic (asymptotic)**—SEs based on the response asymptotic distribution (Lütkepohl, 1990); SEs condence bands; CIs computed as + /– 2 SE confidence bands.

- **Monte Carlo**—SEs derived by simulating impulses by drawing innovations from the standard normal distribution; CIs computed as $+-2$ SE confidence bands.
- **Bootstrap: Standard percentile**—SEs and CIs derived using the standard percentile bootstrap ("Standard Percentile Confidence Interval" on page 890); CIs computed as traditional CIs with $(1-\alpha)$ % coverage.
- **Bootstrap: Hall's percentile**—SEs and CIs derived using Hall's percentile bootstrap ("Hall's Percentile Confidence Interval" on page 890); CIs computed as traditional CIs with $(1-\alpha)$ % coverage.
- **Bootstrap: Hall's studentized**—SEs and CIs derived using Hall's studentized bootstrap ("Hall's Studentized Confidence Interval" on page 891); CIs computed as traditional CIs with $(1-\alpha)$ % coverage.
- **Bootstrap: Kilian's studentized**—SEs and CIs derived using Kilian's unbiased confidence intervals (1998) ("Killian's Unbiased Confidence Interval" on page 891); CIs computed as traditional CIs with $(1-\alpha)$ % coverage.

When you select one of the bootstrap methods, the coverage for the CIs may be specified in the **Confidence intervals levels** edit field. You may provide one or more values in the form of a space delimited list.

For Monte Carlo and bootstrap methods, the number of replications is specified in the **Response standard error options** group. For instance, when performing the MC procedure, replications are specified in the Monte Carlo replication edit field. Alternatively, when doing a bootstrap procedure, the first (possibly the only) stage bootstrap replications are specified in the S**ingle bootstrap replications** edit field. Should a bootstrap procedure require a nested bootstrap loop, the second stage bootstrap replications are set in the D**ouble bootstrap replications** edit field. Should the use of a full double bootstrap be too computationally expensive, fast bootstrap iteration can be achieved by checking the **Fast double bootstrap approximation** checkbox.

Impulse Definitions

The **Impulse** tab provides the following options for specifying 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 $\sum_t e_{i,t}\, e_{j,t} / (T - m)$ where m 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 $\sum_t e_{i,t}\, e_{j,t} / T$.

- **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 869.

- **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.

Displaying Impulse Responses in EViews

To display the impulse responses, for now click on **OK** to accept the default settings and display the impulse responses. Here, EViews displays 9 impulse response graphs including +/–2 analytic asymptotic standard error bands:

Note that the impulse response output is an interactive spool object with the spool tree listing two collapsible categories (**Variables** and **Output Views**) and selectable entries on the left of the window. A category is said to be collapsible when its contents can be collapsed or expanded by clicking on, respectively, the minus or plus box to the left of the category name.

The **Output Views** node may be collapsed or expanded to hide or show the three entries which control the type of output displayed:

- **Table** — if selected, will display a table of the impulse response values and standard errors in the right pane.
- **Multiple Graphs** — if selected, will plot impulse response graphs with one for each requested impulse-response combination.

- **Combined Graphs** — if selected, will plot separate graphs for each requested impulse, with all of the requested responses.

Selecting one of these entries will display the corresponding contents in the right pane of the output window. For example, clicking on **Table** changes the display to show numeric values for the impulse responses and standard errors:

Response of IP:

Period	IP	M1	TB3
1	0.566385	0.000000	0.000000
	(0.02088)	(0.00000)	(0.00000)
2	0.716251	0.030681	0.029848
	(0.03999)	(0.02851)	(0.02987)
3	0.799069	0.073425	0.045671
	(0.05663)	(0.04468)	(0.04802)
4	0.856966	0.131627	-0.028484
	(0.06991)	(0.05339)	(0.06382)
5	0.868171	0.162640	-0.113295
	(0.08291)	(0.06890)	(0.07472)
6	0.855147	0.187397	-0.178115
	(0.09276)	(0.08202)	(0.08091)
7	0.841120	0.210964	-0.232474
	(0.10013)	(0.09224)	(0.08674)
8	0.828129	0.221598	-0.282099
	(0.10468)	(0.10071)	(0.09296)
9	0.813613	0.225829	-0.325101
	(0.10748)	(0.10721)	(0.09928)
10	0.799104	0.228910	-0.361405
	(0.10975)	(0.11239)	(0.10549)

Response of M1:

Period	IP	M1	TB3
1	-0.018372	1.863159	0.000000
	(0.09713)	(0.06868)	(0.00000)
2	0.011380	2.506491	-0.504939

Clicking on the **Combined graphs** entry yields:

showing the impulse responses in graphs grouped by response variables.

The **Variables** node may be collapsed or expanded to hide or show the three node entries: **Impulses**, **Responses**, and if appropriate, **Cholesky ordering**.

The **Impulses** and **Responses** nodes contain entries for the variables in the VAR with which you may interact to change the display behavior.

The **Cholesky ordering** node is informational and does not allow for interaction with the entries in the node.

You may interact with entries in the **Impulses** and **Responses** nodes by:

- Clicking on (toggling) each variable to show or not show the impulses or response of the variable in the output views
- Drag the entries within the node to change the order of variables in the output views.

Entries with "*" by their name indicate that they will be displayed in the output views. Clicking on a "*" entry will remove it from the output view. Clicking on an entry without a "*" will add it to the output view. Here, we see that all impulses and responses will be displayed.

If we double click on the **Impulses** entries for **IP** and **TB3** to remove the "*", and double click on **Multiple graphs**, the view will change to show only responses to the **M1** impulses:

We may click-and-drag the TB3 entry in the **Responses** node to rearrange the order of the graphs

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 the **Cholesky Decomposition** orthogonal factorizations or the **Structural Decomposition**, if available.

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 869. 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.

As with the output for impulse responses ("Impulse Responses" on page 853), the resulting output is interactive. You may click on any of the **Output Views** node entries to change the display type. Here we see a Stacked graphs representation of the decomposition.

Similarly, you may click on the **Decompositions** node under **Variables** and toggle on the variable names to show or hide the display of the decomposition of specific variables.

Historical Decomposition

An alternative method of innovation accounting is to decompose the observed series into the components corresponding to each structural shock. Burbridge and Harrison (1985) propose transforming observed residuals to structural residuals, and then for each observation beyond some point in the estimation sample, computing the contribution of the different accumulated structural shocks to each observed variable.

To obtain the historical decomposition, select **View/Historical Decomposition...** from the var object toolbar. You should provide the same information as for impulse responses and variance decomposition above.

In addition, you may use the **Include baseline in decomposition** to choose whether to include the base projection in the decomposition, or whether to plot only the stochastic accumulations and total.

Lastly, you may use the edit fields to specify an optional **Start** and **End** period to the decomposition. Both periods must be within the range of the original estimation period. If you leave the fields blank, EViews will perform the decomposition from the start to the end of the full estimation period.

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 808). Here, we discuss only those procedures that are unique to the VAR object.

Forecasting

You may produce forecasts directly from an estimated VAR object by clicking on the **Forecast** button or by selecting **Proc/Forecast**. EViews will display the forecast dialog:

Most of the dialog should be familiar from the standard equation forecast dialog. There are, however, a few minor differences.

First, the fields in which you enter the forecast name and optional S.E. series names now refer to the *character suffix* which you will use to form output series names. By default, as depicted here, EViews will append the letter "f" to the end of the original series names to form the output series names. If necessary, the original name will be converted into a valid EViews series name.

Second, if you choose to compute standard errors of the forecast, EViews will obtain those values via simulation. You will be prompted for the number of **Simulation repetitions**, and the "**% failed reps before halting**" the simulation setting.

Lastly, in addition to a **Forecast evaluation**, you are given a choice of whether to display the output graphs as **Individual graphs**, as **Multiple graphs**, or both.

Clicking on **OK** instructs EViews to perform the forecast and, if appropriate to display the output:

In this case, the output consists of a spool containing the forecast evaluation of the series in the VAR, along with individual graphs of the forecasts and the corresponding actuals series.

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 details for this procedure are provided in "Structural (Identified) VARs" below. 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

A structural VAR (SVAR) uses additional identifying restrictions and estimation of structural matrices to transform VAR errors into uncorrelated structural shocks. Obtaining structural shocks is central to a wide range of VAR analysis, including impulse response, forecast variance decomposition, historical decomposition, and other forms of causal analysis. See, for example, Amisano and Giannini (1997), Martin, Hurn and Harris (2013).

We begin with the SVAR specification

$$A y_t = A_1^s y_{t-1} + \cdots + A_p^s y_{t-p} + C^s x_t + B u_t \tag{44.31}$$

where A, all of the A_i^s, and C^s are the structural coefficients, and the u_t are the orthonormal unobserved structural innovations with $E(u_t u_t') = I_k$.

It is easy to see the relationship between the SVAR specification and the corresponding reduced-form VAR. Assuming that A is invertible, we have:

$$\begin{aligned} y_t &= A^{-1} A_1^s y_{t-1} + \cdots + A^{-1} A_p^s y_{t-p} + A^{-1} C^s x_t + A^{-1} B u_t \\ &= A_1 y_{t-1} + \cdots + A_p y_{t-p} + C x_t + \epsilon_t \end{aligned} \tag{44.32}$$

so the reduced-form lag matrices $A_i = A^{-1} A_i^s$ and $C = A^{-1} C^s$, and the reduced form error structure is given by

$$\begin{aligned} \epsilon_t &= A^{-1} B u_t = S u_t \\ E(\epsilon_t \epsilon_t') &= \Sigma_\epsilon = A^{-1} B B' A^{-1\prime} = S S' \end{aligned} \tag{44.33}$$

where $S = A^{-1} B$.

SVAR estimation uses estimates $\hat{\Sigma}_\epsilon$ obtained from the reduced form VAR, the short-run covariance relationships and any restrictions in Equation (44.33), and long-run restrictions

on the accumulated impulse responses (as described below), to identify and estimate the model. The challenge in SVAR estimation is that there are only $k(k+1)/2$ moments in Σ_ϵ and more than $k(k+1)/2$ elements in A and B, or in S so that those matrices are not identified unless additional restrictions are provided.

SVAR Restrictions

Prior knowledge and theory will often suggest restrictions on structural matrices, allowing you to identify and estimate the parameters of the SVAR. EViews allows you to specify restrictions in different ways, with support for restrictions using two different short-run representations, and restrictions on the long-run impulse-responses.

Our discussion of SVAR restrictions and estimation is necessarily brief. We encourage those interested in greater detail to consult Rubio-Ramirez, Waggoner, and Zha (2010) for detailed discussion of identification and other related issues.

A-B Restrictions (Short-run)

From Equation (44.33) we may write the short-run A-B model as:

$$\begin{aligned} \epsilon_t &= A^{-1}Bu_t \\ A\epsilon_t &= Bu_t \\ \Sigma_\epsilon &= A^{-1}BB'A^{-1\prime} \end{aligned} \tag{44.34}$$

We may use the estimated moments $\hat{\Sigma}_\epsilon$ along with the $k(k+1)/2$ unique covariance equations in Equation (44.34) to estimate the $2k^2$ elements in A and B. Satisfying the order condition requires an additional $2k^2 - k(k+1)/2 = k^2 + k(k-1)/2$ restrictions for identification.

Restrictions on A and B take the form of assumptions about the structure of contemporaneous feedback of variables in the SVAR and assumptions about the correlation structure of the errors, respectively.

S Restrictions (Short-run)

The short-run S model is given by:

$$\begin{aligned} \epsilon_t &= Su_t \\ \Sigma_\epsilon &= SS' \end{aligned} \tag{44.35}$$

We may use estimates $\hat{\Sigma}_\epsilon$ along with covariance restrictions in Equation (44.35) to estimate the elements of S.

The S model may be more convenient to specify in that it involves the k^2 element product $S = A^{-1}B$ matrix and not the $2k^2$ elements in the individual A and B. Thus, the order

condition only requires an additional $k^2 - k(k+1)/2 = k(k-1)/2$ restrictions for identification.

This convenience comes at a cost, however, as the individual A and B matrices as the latter are not identified from S alone. For example, a given S model with S is equivalent to an A-B model with $A = I_k$ and $B = S$, or an A-B model with $A = S^{-1}$ and $B = I$.

Thus, restrictions on S take the form of restrictions on the composite factor loadings, but offer no insight into the decomposition into endogeneity A and error loading components B.

F Restrictions (Long-run)

The identifying restrictions embodied in the relations $A\epsilon_t = Bu_t$ and $\epsilon_t = Su_t$ are commonly referred to as short-run restrictions. Blanchard and Quah (1989) proposed an alternative identification method using restrictions on the long-run properties of the accumulated impulse responses.

We may write these long-run restrictions as:

$$\begin{gathered}(I - A_1 - A_2 - \ldots - A_p)^{-1}\epsilon_t = \Psi\epsilon_t = Fu_t \\ \Sigma_\epsilon = \Psi^{-1}FF'\Psi^{-1\prime}\end{gathered} \tag{44.36}$$

where $\Psi = (I - A_1 - A_2 - \ldots - A_p)^{-1}$ is the long-run multiplier, which may be estimated using the reduced form VAR parameter estimates. Note that the long-run F model is related to the S model through $F = \Psi S$ and that as in the S model, the order condition requires an additional $k^2 - k(k+1)/2 = k(k-1)/2$ restrictions.

The F model employs estimates of the moments $\hat{\Sigma}_\epsilon$ along with covariance relationships and restrictions from Equation (44.33) to estimate the k^2 elements in F. Thus, long-run identifying restrictions are specified in terms of the elements of this F matrix, typically in the form of zero restrictions. The restriction $F_{i,j} = 0$ means that the (accumulated) response of the i-th variable to the j-th structural shock is zero in the long-run.

Note that as with the S model, knowledge of A and B is sufficient to compute F, but the converse is not true.

Specifying SVAR Restrictions in EViews

EViews supports linear restrictions among the elements of matrices A, B, S, and F. In addition to commonly employed restrictions on single elements of the structural matrices, you may specify restrictions across elements of a given matrix, and you may even specify restrictions span A, B, S, and F.

For example, you may specify:

- a single element constant restriction, *e.g.* $A_{1,1} = 2.5$

- a general linear relationship within a single matrix, *e.g.* $B_{2,2} = B_{3,3}/2$ or $A_{1,1} + A_{2,1} = 1$
- a general linear relationship that spans multiple matrices, *e.g.* $A_{1,2} = 3 \cdot B_{3,3}$ or $S_{1,1} + S_{2,2} - F_{3,3} = 1.5$

Restrictions in EViews are specified via pattern matrices and/or text expressions. Pattern matrices are a convenient way to place simple constant restrictions on individual elements of a structural matrix, while text expressions provide for the full range of supported restrictions.

A pattern matrix is a $k \times k$ matrix whose non-missing values, *i.e.*, non-NAs, specify constant restrictions on the corresponding matrix elements. All missing values, *i.e.*, NAs, place no restrictions on the corresponding matrix elements (such elements many still be restricted via text expressions). For example, suppose you want to restrict A to be unit lower triangular and B to be diagonal. With $k = 3$ the following pattern matrices could be employed:

$$A = \begin{pmatrix} 1 & 0 & 0 \\ \text{NA} & 1 & 0 \\ \text{NA} & \text{NA} & 1 \end{pmatrix}, \qquad B = \begin{pmatrix} \text{NA} & 0 & 0 \\ 0 & \text{NA} & 0 \\ 0 & 0 & \text{NA} \end{pmatrix}. \tag{44.37}$$

Text expressions allow you to write linear equations or use a function-like syntax to specify restrictions on one or more matrix elements. For example, replicating some of the earlier restrictions:

```
@a(1,1) = 2.5
@b(2,2) = @b(3,3) / 2
@a(1,1) + @a(2,1) = 1
@a(1,2) = 3 * @b(3,3)
@s(1,1) + @s(2,2) - @f(3,3) = 1.5
```

Notice that the EViews text syntax requires that the canonical structural matrix names are preceded by "@" to avoid ambiguity with workfile objects, *e.g.*, scalars or matrix object elements.

In addition, EViews offers function-like expressions that concisely specify popular sets of restrictions. In the following list, the token X can be substituted with any of the canonical matrices A, B, S, and F. The canonical names should not be preceded by "@" in this context since there is no potential workfile object ambiguity in the function argument(s).

@X = *mat*	Use *mat* as a pattern matrix for matrix X, *e.g.*, "@a = mat1", "@b = @mat2".
@vec(X) = n_1 , n_2 , n_3 , …	Restricts all elements of matrix X similar using the specified pattern matrix (provided in list form). Element ordering matches the vectorization of the matrix, *i.e.*, the elements of the first column, followed by the second column, followed by the third column, etc.
@diag(X)	Restricts X to be a diagonal matrix, *i.e.*, off-diagonal elements are zero. The diagonal elements are unrestricted.
@diag(X) = n	Restricts X to be a diagonal matrix with elements on the diagonal restricted to be n.
@lower(X)	Restricts X to be a lower triangular matrix, *i.e.*, elements above the diagonal are zero.
@unitlower(X)	Restricts X to be a unit lower triangular matrix, *i.e.*, elements above the diagonal are zero and elements on the diagonal are one.
@upper(X)	Restricts X to be an upper triangular matrix, *i.e.*, elements below the diagonal are zero.
@unitupper(X)	Restricts X to be a unit upper triangular matrix, *i.e.*, elements below the diagonal are zero and elements on the diagonal are one.
@row(X, r) = n	Restricts the elements in row r of X to equal n.
@col(X, c) = n	Restricts the elements in column c of X equal n.

Examples

Suppose we wish to recreate a recursive Cholesky orthogonalization (using the order of the variables in the VAR specification). This restriction is equivalent to requiring that the S matrix is lower triangular. In the SVAR dialog ("Restrictions" on page 875) there is **Restriction Preset** for exactly this scenario, but we can also use a pattern matrix or text expression to restrict S. For simplicity we'll assume an underlying VAR named V1 with three endogenous variables ($k = 3$). Each of the following sequences of commands produces the same set of S model restrictions and then estimates the model:

Approach #1: Using a pattern matrix

```
matrix(3, 3) pattern
pattern.fill na, na, na, 0, na, na, 0, 0, na
v1.svar(s=pattern)
```

Approach #2: Using a text expression equivalent to a pattern matrix

```
v1.append(svar) @vec(s) = na, na, na, 0, na, na, 0, 0, na
v1.svar
```

Approach #3: Using a text expression with specialized function

```
v1.append(svar) @lower(s)
v1.svar
```

For the next example, suppose we wish to estimate an A-B model where B captures only the standard deviations of the structural innovations so that the off-diagonal elements of B are set to zero. For $k = 3$, these six restrictions fall short of the twelve restrictions necessary to satisfy the order condition, so (at least) six additional restrictions are required. To satisfy the order condition, we will restrict matrix A to have ones along the diagonal and require $A_{1,2} = 0$, $A_{1,3} = 0$, and $A_{2,1} = A_{3,1}$.

Approach #1: Using pattern matrices when possible

```
matrix(3, 3) patternA
patternA.fill 1, na, na, 0, 1, na, 0, na, 1
matrix(3, 3) patternB
patternB.fill na, 0, 0, 0, na, 0, 0, 0, na
v1.append(svar) @a(2,1) = @a(3,1)
v1.svar(a=patternA, b=patternB)
```

Approach #2: Using text expressions when possible

```
v1.append(svar) @vec(a) = 1, na, na, 0, 1, na, 0, na, 1
v1.append(svar) @a(2,1) = @a(3,1)
v1.append(svar) @diag(b)
v1.svar
```

Finally, suppose we wish to restrict the long-run impulse responses of an SVAR system. For illustrative purposes, suppose the responses of some variables to innovations are precisely half the responses of other variables, specifically $F_{1,1} = F_{2,1}/2$, $F_{2,1} = F_{3,2}/2$, and $F_{1,3} = F_{2,3}/2$. Since these restrictions all involve more than one element F, we must use text expressions to specify the linear restrictions

```
v1.append(svar) @f(1,1) = @f(2,1) / 2
v1.append(svar) @f(2,1) = @f(3,2) / 2
v1.append(svar) @f(1,3) = @f(2,3) / 2
v1.svar
```

Estimating an SVAR in EViews

Once you have estimated a reduced form VAR, the SVAR specification and estimation dialog may be displayed by selecting **Proc/Estimate Structural Factorization...** You may use the

dialog to specify a base collection of restrictions, customize restrictions on the four canonical matrices, and adjust options for the estimation procedure.

Restrictions

The **Identifying Restrictions** tab allows you to specify your SVAR restrictions.

The **Restriction Preset** drop-down menu provides a variety of pre-built restriction templates that can be applied to the SVAR model. Several of the presets can be used as is, while others require additional restrictions to meet the order condition.

The selection area on the left of the dialog allows you to view the current pattern matrices for any and all of the four canonical matrices as well as the current collection of text expression restrictions. Selecting a matrix element will change the dialog display to show the current restrictions for that matrix. There are two methods for specifying restrictions:

- Select **Manual** and enter the elements directly in the dialog.
- Use the **Workfile matrix** edit field to provide the name of a matrix in the workfile that contains the desired patterns.

To you click on the **Text** entry in the selection area, you may enter the specification of your restrictions using the syntax described above ("Specifying SVAR Restrictions in EViews" on page 871).

Note that when a restriction has been applied to any of the selections, the left-hand side of the dialog will display the message "* restriction applied" and the names in the selection area will display a matching "*" as appropriate.

Lastly, we caution you that using the dropdown to select a preset will specify appropriate initial matrix restrictions, so that any existing customization will be overwritten.

Optimization

The log likelihood is maximized using Newton-Raphson with the Marquardt trust-region technique, analytic gradients, and numeric Hessians. See "Marquardt," on page 1321). See Amisano and Giannini (1997) for the analytic expression of the first derivatives.

Estimation of the SVAR model is based on the relation $\epsilon_t = S(\theta)u_t$, with S specified in terms of elements of A and B, or S, or F, which in turn depend on θ. We estimate θ via maximum likelihood using the concentrated log-likelihood function:

$$\begin{aligned} \ln L(\theta;\, \epsilon) &= \ln\left(\prod_{t=1}^{T} f(\epsilon_t|\theta)\right) \\ &= -\frac{T}{2}(k\ln(2\pi) + \ln|S(\theta)S(\theta)'| + \mathrm{tr}(\hat{\Sigma}_\epsilon(S(\theta)S(\theta)')^{-1})) \end{aligned} \tag{44.38}$$

where f is the probability density function of the multivariate normal distribution with zero mean and tr is the matrix trace operation.

As the underlying parameterization of the model in EViews involves a few idiosyncrasies, a few comments are in order:

- The underlying estimation parameters θ will be used to form elements of A and B, or elements of S, or elements of F (mirroring the A-B, S, and F models), with the parameterization chosen to reflect the specified restrictions.
- EViews parametrically enforces your restrictions by analytically determining a set of unconstrained free parameters in one of the three above models. The models are prioritized, from highest to lowest: A-B, then S, and then F.

  If, for example, the restrictions are in terms of a simple A-B model, then the parameterization will be in terms of the elements of A and B. If the restrictions are in terms of S and F, then the restrictions on F would be converted to restrictions on elements of S.

Regardless of which parameterization is used, the parameters can be used to calculate $S(\theta)$ and the likelihood.

In the special case of an A-B model with additional restrictions on S or F there is an additional complication, since restrictions on S or F cannot, in general, be converted to restric-

tions on A and B. In this setting, restrictions on S and F are enforced numerically through inclusion of a penalty term in the log-likelihood function.

As the penalty function approach does not actually reduce the number of parameters, this scenario produces a situation where the parameters may technically be under-identified. Ideally, the penalized log-likelihood function can guide the optimizer to convergence as if the extra parameters did not exist. However, convergence can be more difficult to achieve, even when the model is correctly identified. We therefore recommend that when using an A-B model, the number of restrictions on S and F be kept to a minimum.

Once you provide identifying restrictions, you are ready to estimate the SVAR. Simply click on **OK** to estimate the model.

Prior to estimation, you may first to click on the **Optimization Control** tab in the **SVAR Options** dialog to show and modify the estimation settings:

There are three sets of settings for you to consider.

Starting Values

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 field. **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.

The **Draw from...** options randomly draw the starting values for the free parameters from the specified distributions.

Iteration 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 maximum number of iterations, and the convergence criterion.

Sign Restrictions

For some restrictions, the signs of the matrices are not identified; see Christiano, Eichenbaum, and Evans (1999) for a discussion of this issue. When the sign is indeterminate, EViews will choose a post-estimation normalization such that the diagonal elements of matrix S are positive, with a further preference for the diagonal elements of matrix B to be positive in model. While it is not always possible to make the diagonal elements positive, this normalization procedure attempts to give all structural impulses positive signs (as well as the Cholesky factorization). The EViews default behavior applies this normalization rule whenever applicable. If you do not want to normalize the signs, deselect the **Normalize signs** option.

Ignore Errors

The **Ignore errors** checkbox instructs EViews to suppress "Near Singular Matrix" and other error messages during estimation. Consequently, if an error occurs, estimation results may be incomplete and inaccurate.

Optimization Trace

The **Optimization trace period** tells EViews to summarize the ongoing optimization (iteration number, log-likelihood, parameter values) at the user-specified interval. Summary information is displayed in an unnamed text object.

Estimation Output

Once optimization 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 parameter estimates.

For over-identified models, we also report the LR test for over-identification. The LR test statistic is computed as:

$$LR = 2(l_u - l_r) = T(\mathrm{tr}(P) - \log|P| - k)$$

where $P = S^{-T}S^{-1}\Sigma$. Under the null hypothesis that the restrictions are valid, the LR statistic is asymptotically distributed as $\chi^2(q-k)$, where q is the number of identifying restrictions.

The top portion of your SVAR output will show the restrictions, parameterization, and estimated coefficients,

```
Structural VAR Estimates
Date: 06/15/17   Time: 16:15
Sample: 1955Q1 1987Q3
Included observations: 131
Estimation method: Least squares via Gauss-Newton (analytic derivatives)
Convergence achieved after 16 iterations
Structural VAR is just-identified
```

Model: Ae = Bu where E[uu']=I

A =

C(1)	C(5)	C(9)	C(13)
C(2)	C(6)	C(10)	-C(10)
C(3)	C(7)	C(11)	C(14)
C(4)	C(8)	C(12)	C(15)

B =

1	0	0	0
0	1	0	0
0	0	1	0
0	0	0	1

@F(1,2) = 0
@F(1,3) = 0
@F(1,4) = 0
@S(1,2) = 0
@S(1,3) = 0
@A(2,3) + @A(2,4) = 0
@DIAG(B) = 1
including the restriction(s)

	Coefficient	Std. Error	z-Statistic	Prob.
C(1)	0.938016	0.059131	15.86337	0.0000
C(2)	0.141273	0.101338	1.394080	0.1633
C(3)	-0.240158	0.102263	-2.348425	0.0189
C(4)	0.569856	0.092085	6.188399	0.0000
C(5)	-0.614865	0.045403	-13.54246	0.0000
C(6)	0.590929	0.195760	3.018647	0.0025
C(7)	0.768293	0.146066	5.259897	0.0000
C(8)	0.928173	0.094779	9.793037	0.0000
C(9)	0.275181	0.019167	14.35711	0.0000
C(10)	0.178343	0.036007	4.953047	0.0000
C(11)	0.154229	0.056652	2.722407	0.0065
C(12)	-0.415401	0.037699	-11.01892	0.0000
C(13)	-0.081105	0.007293	-11.12095	0.0000
C(14)	0.187028	0.030663	6.099560	0.0000
C(15)	0.122432	0.017324	7.067029	0.0000

Objective value -952.9302

while the bottom portion shows the structural matrix estimates,

Estimated A matrix:			
0.938016	-0.614865	0.275181	-0.081105
0.141273	0.590929	0.178343	-0.178343
-0.240158	0.768293	0.154229	0.187028
0.569856	0.928173	-0.415401	0.122432
Estimated B matrix:			
1.000000	0.000000	0.000000	0.000000
0.000000	1.000000	0.000000	0.000000
0.000000	0.000000	1.000000	0.000000
0.000000	0.000000	0.000000	1.000000
Estimated S matrix:			
0.760158	0.000000	0.000000	0.503565
-0.176944	0.628329	0.339939	0.278754
0.924631	0.517439	1.547698	-0.998017
0.940489	-3.007808	2.674070	0.324515
Estimated F matrix:			
1.236121	0.000000	0.000000	0.000000
-0.127932	-0.520269	-0.027516	0.654769
5.272415	7.150987	4.518060	-2.794496
7.864187	4.919853	6.432775	-7.228512

If you switch the view of the VAR window, you can come back to the previous results (without reestimating) by selecting **View/Structural Factorization** 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 992 of the *Command and Programming Reference* for a list of available VAR data members.

Compatibility Notes

The increased flexibility of SVAR restrictions in EViews 10 is accompanied by streamlined options in the `svar` estimation procedure and a new text language for specifying restrictions.

The options and text syntax used by previous versions of EViews will continue to work in EViews 10 with the same limitations, *e.g.*, disallowing simultaneous short-run and long-run restrictions.

The new restriction capabilities of EViews 10 are only available via the new interface. Furthermore, SVAR estimation results (**View/Structural Factorization**) produced in EViews 10 and saved to a workfile will be accessible in previous version of EViews only in the following cases:

- The SVAR model was specified using only legacy options and text restrictions (if present). You may therefore use, for example, EViews 9 syntax to maintain workfile compatibility with EViews 9.
- The SVAR A-B-only or F-only model was estimated by pattern matrix restrictions specified in matrix objects.

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. You may test for cointegration using an estimated VAR object, Equation object estimated using nonstationary regression methods, or using a Group object (see Chapter 55. "Cointegration Testing," on page 1245).

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:

$$y_{2,\,t} = \beta y_{1,\,t} \tag{44.39}$$

The corresponding VEC model is:

$$\begin{aligned} \Delta y_{1,\,t} &= \alpha_1(y_{2,\,t-1} - \beta y_{1,\,t-1}) + \epsilon_{1,\,t} \\ \Delta y_{2,\,t} &= \alpha_2(y_{2,\,t-1} - \beta y_{1,\,t-1}) + \epsilon_{2,\,t} \end{aligned} \tag{44.40}$$

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 Johansen (1995) trend specifications as explained in "Deterministic Trend Specification" on page 1246. 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. "Imposing VEC Restrictions" on page 884 describes these restriction in greater detail. Note that the contents of this tab are 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 "Estimation Output" on page 837, 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 (44.11). 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

To forecast from your VEC, click on the Forecast button on the toolbar and fill out the dialog as described in "Forecasting," on page 866

Data Members

Various results from the estimated VAR/VEC can be retrieved through the command line data members. "Var Data Members" on page 992 of the *Command and Programming 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.

Imposing VEC Restrictions

Since the cointegrating vector β is not fully identified, you may wish to impose your own identifying restrictions when performing estimation.

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 estima-

tion, open the test, select **Vector Error Correction** in the main VAR estimation dialog, then click on 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)*y1 + B(i,2)*y2 + ... + B(i,k)*yk
```

where `y1`, `y2`, ... are the (lagged) endogenous variable. Then, if you want to impose the restriction that the coefficient on `y1` 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 `y1` 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 + ... + A(i,r)*CointEqr
```

Restrictions on the adjustment coefficients are currently limited to linear homogeneous restrictions so that you must be able to write your restriction as $R \cdot \mathrm{vec}(\alpha) = 0$, 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.

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.

Technical Discussion

Impulse Response Standard Errors and Confidence Intervals

Traditionally, confidence intervals for VAR impulse responses have been based on the Lütkepohl (1990) asymptotic approximation and EViews provides these values for ordinary VAR estimators. Nevertheless, it is well known that asymptotic validity is often questionable in small samples. In this regard, Monte Carlo and bootstrap methods are of considerable practical interest as they often perform better when the number of observations is limit

The VAR Bootstrap

To formalize matters, consider a system of time series of length T. If the system is modelled as a vector autoregression (VAR) process of order p, the data generating process (DGP) for the system is written as:

$$y_t = \boldsymbol{\nu} + \sum_{j=1}^{p} A_j y_{t-j} + u_t \tag{44.41}$$

where y_t and $\boldsymbol{\nu}$ are K-element vectors, $A_1, \ldots, A_p$ are $K \times K$ matrices, and $u_t \sim N(0, \boldsymbol{\Sigma}_u)$ where $\boldsymbol{\Sigma}_u$ is a $K \times K$ covariance matrix.

Next suppose that the object of interest (*e.g.*, estimator, test statistic, etc.) is the functional:

$$q_T = q(y_1, \ldots, y_T, \boldsymbol{\nu}, A_1, \ldots, A_p) \tag{44.42}$$

The distribution $F_{q,T}$ of q_T may be approximated by generating a large number, say B, of different outcome paths ($q_T{}^{*b}$ for $b = 1, \ldots, B$) for the "*" is used to indicate that the object is derived from simulation. In particular, the B outcomes $q_T{}^{*b}$ are obtained by simulating from the DGP in Equation (44.41) to generate B independent outcomes $y_t{}^{*b}$ of y_t and for each such draw, compute the corresponding $q_T{}^{*b}$ using Equation (44.42) but with the generated data. The distribution of the q_T is then estimated as the empirical distribution of the set of simulated outcomes $\{y_t{}^{*b}\}_{b=1}^{B}$.

There are a variety of simulating data from the DGP. EViews supports three: residual based bootstrap, residual based double bootstrap, fast double bootstrap. We consider each in turn.

Residual Based Bootstrap

A commonly used method of generating data from Equation (44.41) is the *residual based bootstrap*. The procedure is straightforward:

1. Using the original data y_t and the specification Equation (44.41), obtain estimates of the coefficients $(\hat{\boldsymbol{\nu}}, \hat{A}_1, \ldots, \hat{A}_p)$ and the corresponding the residual series $\hat{u}_t$.
2. Derive the centered residual series $\tilde{u}_t = \hat{u}_t - \bar{\hat{u}}_t$, where $\bar{\hat{u}}_t = (1/T)\sum_{t=1}^{T} \hat{u}_t$.
3. For each $b = 1, \ldots, B$:
   - Draw a series of length T of bootstrap innovations ${u_t}^{*b}$ by randomly drawing with replacement from the $\tilde{u}_t$.
   - Generate a bootstrap DGP of ${y_t}^{*b}$ as follows:

$$ {y_t}^{*b} = \hat{\boldsymbol{\nu}} + \sum_{j=1}^{p} \hat{A}_j {y_{t-j}}^{*b} + {u_t}^{*b} \tag{44.43} $$

   - Using the generated ${y_t}^{*b}$, estimate coefficients to obtain $(\hat{\boldsymbol{\nu}}^{*b}, {\hat{A}_1}^{*b}, \ldots, {\hat{A}_p}^{*b})$.
   - Obtain a bootstrap estimate of the object of interest

$$ {q_T}^{*b} = q({y_1}^{*b}, \ldots, {y_T}^{*b}, \hat{\boldsymbol{\nu}}^{*b}, {\hat{A}_1}^{*b}, \ldots, {\hat{A}_p}^{*b}) \tag{44.44} $$

4. The final bootstrap estimator of q_T, typically denoted as $\hat{q}_T^*$ is obtained as the average of the individual bootstrap estimates ${\hat{q}_T}^{*b}$. In other words

$$ \hat{q}_T^* = \frac{1}{B}\sum_{b=1}^{B} {\hat{q}_T}^{*b} \tag{44.45} $$

Residual Based Double Bootstrap

It is sometimes necessary to obtain bootstrap statistics of a bootstrap statistic, for example, when the object of interest is the distribution ${F_{q,T}}^{*b}$ of the b-th bootstrap estimate ${\hat{q}_T}^{*b}$. This distribution is typically needed in deriving the variance of a bootstrap statistics.

In this case, for each $b = 1, \ldots, B$, the distribution ${F_{q,T}}^{*b}$ is approximated by generating a large number, say R, of different outcome paths (${q_T}^{**r}$ for $r = 1, \ldots, R$), where the superscript "**" indicates that the object is derived from a second stage bootstrap simulation.

In particular, the R outcomes ${q_T}^{**r}$ are obtained by simulating, from the bootstrap DGP, using a bootstrap draw of b and Equation (44.43) and generating new R outcomes ${y_t}^{**r}$ and then for each of those outcomes, a corresponding ${q_T}^{**r}$. The distribution of the ${\hat{q}_T}^{*b}$

is then estimated as the empirical distribution of the set of simulated outcomes $\{y_t^{**r}\}_{r=1}^{R}$.

The resampling procedure used to derive these second stage bootstrap statistics is the residual bootstrap double bootstrap. The algorithm is summarized below:

1. Using the original data y_t and the specification Equation (44.41), obtain estimates of the coefficients $(\hat{\boldsymbol{\nu}}, \hat{A}_1, \ldots, \hat{A}_p)$ and the corresponding the residual series $\hat{u}_t$.
2. Derive the centered residual series $\tilde{u}_t = \hat{u}_t - \overline{\hat{u}}_t$, where $\overline{\hat{u}}_t = (1/T)\sum_{t=1}^{T} \hat{u}_t$.
3. For each $b = 1, \ldots, B$:
   - Perform a residual bootstrap: draw bootstrap residuals u_t^{*b} from the $\tilde{u}_t$, generate bootstrap data y_t^{*b}, estimate $(\hat{\boldsymbol{\nu}}^{*b}, \hat{A}_1^{*b}, \ldots, \hat{A}_p^{*b})$, and obtain the first stage bootstrap estimate

     $$q_T^{*b} = q(y_1^{*b}, \ldots, y_T^{*b}, \hat{\boldsymbol{\nu}}^{*b}, \hat{A}_1^{*b}, \ldots, \hat{A}_p^{*b}) \tag{44.46}$$

     This step is the same as each step in the standard residual bootstrap.
   - Derive first stage bootstrap residuals based on the bootstrap data

     $$\hat{u}_t^{*b} = y_t^{*b} - \left(\hat{\boldsymbol{\nu}}^{*b} + \sum_{j=1}^{p} \hat{A}_j^{*b} y_{t-j}^{*b}\right) \tag{44.47}$$

     and obtain centered first stage bootstrap residuals by subtracting off means,

     $$\tilde{u}_t^{*b} = \hat{u}_t^{*b} - \overline{\hat{u}_t^{*}}, \text{ where } \overline{\hat{u}}_t = (1/T)\sum_{t=1}^{T} \hat{u}_t. \tag{44.48}$$

   - For each $r = 1, \ldots, R$ perform a residual bootstrap: draw bootstrap residuals u_t^{**r} from the $\tilde{u}_t^{*b}$, generate bootstrap data y_t^{**r}, estimate $(\hat{\boldsymbol{\nu}}^{**r}, \hat{A}_1^{**r}, \ldots, \hat{A}_p^{**r})$, and obtain a second stage bootstrap estimate

     $$q_T^{**r} = q(y_1^{**r}, \ldots, y_T^{*r}, \hat{\boldsymbol{\nu}}^{*r}, \hat{A}_1^{*r}, \ldots, \hat{A}_p^{*r}) \tag{44.49}$$

Fast Double Bootstrap

Performing a full double bootstrap algorithm is extremely computationally demanding. A single bootstrap algorithm typically requires the computation of $1 + B$ test statistics —a single statistic performed on the original data and an additional B statistics obtained in the bootstrap loop. On the other hand, a full double bootstrap requires $1 + B + BR$ statistics, with the additional BR statistics computed in the inner loop of the double bootstrap algorithm. For instance, setting $B = 999$ and $R = 499$, the double bootstrap will require the computation of no less than 499,501 test statistics. In this regard, several attempts have been proposed to circumvent this computational burden, among the most popular of which is the fast double bootstrap (FDB) algorithm of Davidson and MacKinnon (2002) and general iterated variants proposed in Davidson and Trokic (2020). The principle governing the success

of these algorithms relies on setting $R = 1$. For instance, the FDB has proved particularly useful in approximating full double bootstrap algorithms with roughly less than the twice the computational burden of the traditional bootstrap: $1 + 2B$ to be precise.

Bootstrapping Impulse Responses

Let ϕ represent an impulse response coefficient that depends on the data. In terms of the discussion above we may think of $\phi = q_T$ and contemplate obtaining bootstrap measures of the precision of ϕ.

VAR Impulse Responses

To derive bootstrap confidence intervals for impulse responses, let ϕ, $\hat{\phi}$, and $\hat{\phi}^*$ denote, respectively, some general impulse response coefficient, its estimator derived using the original data y_t and the DGP in Equation (44.41), and the associated bootstrap estimator derived using a procedure like the residual based bootstrap "Residual Based Bootstrap" on page 888.

Next, let denote a given significance level so that the confidence interval of interest has coverage $1 - \alpha$. Then, the following bootstrap confidence intervals have been considered in the literature. See also Lütkepohl (2005) for further details.

Standard Percentile Confidence Interval

The most common bootstrap algorithm for impulse response CIs is based on the percentile confidence interval described in Efron and Tibshirani (1993). The bootstrap CI is derived as:

$$CI_{SP} = \left[s_{\frac{\alpha}{2}}{}^*, s_{\left(1-\frac{\alpha}{2}\right)}{}^* \right] \tag{44.50}$$

where $s_{\alpha/2}{}^*$ and $s_{(1-\alpha/2)}$ are the $\alpha/2$ and $(1-\alpha/2)$ quantiles of the empirical distribution of $\hat{\phi}^*$, respectively.

Hall's Percentile Confidence Interval

An alternative to the standard percentile CI is the Hall (1992) confidence interval. The idea behind this method is that the distribution of $(\hat{\phi} - \phi)$ is asymptotically approximated by the distribution of $(\hat{\phi}^* - \hat{\phi})$. Accordingly, the Hall bootstrap CI is given by

$$CI_{HP} = \left[\hat{\phi} - t^*{}_{\left(1-\frac{\alpha}{2}\right)}, \hat{\phi} - t^*{}_{\frac{\alpha}{2}} \right] \tag{44.51}$$

where $t^*{}_{\alpha/2}$ and $t^*{}_{(1-\alpha/2)}$ are the $(\alpha/2)$ and $(1-\alpha/2)$ quantiles of the empirical distribution of the bootstrap draws

$$\{\hat{\phi}^{*b} - \hat{\phi}^b\}_{b=1}^B \tag{44.52}$$

See Hall (1986) for details.

Hall's Studentized Confidence Interval

Another possible approach to computing the CI is to approximate the distribution of the studentized statistic $\hat{\tau}$,

$$\hat{\tau} = \frac{\hat{\phi} - \phi}{\sqrt{\hat{\text{var}}(\hat{\phi})}} \tag{44.53}$$

with the distribution of the bootstrap studentized statistic:

$$\hat{\tau}^* = \frac{\hat{\phi}^* - \hat{\phi}}{\sqrt{\hat{\text{var}}(\hat{\phi}^*)}} \tag{44.54}$$

In this case, the bootstrap CI is derived as:

$$CI_{HP} = \left[\hat{\phi} - t^{**}_{\left(1-\frac{\alpha}{2}\right)}\hat{\text{var}}(\hat{\phi}),\ \hat{\phi} - t^{**}_{\frac{\alpha}{2}}\hat{\text{var}}(\hat{\phi})\right] \tag{44.55}$$

where $t^{**}_{\alpha/2}$ and $t^{**}_{B(1-\alpha/2)}$ are the $(\alpha/2)$ and $(1-\alpha/2)$ quantiles of the empirical distribution of $\{\hat{\tau}^{*b}\}_{b=1}$, where

$$\hat{\tau}^{*b} = \frac{\hat{\phi}^{*b} - \hat{\phi}}{\sqrt{\hat{\text{var}}(\hat{\phi}^{*b})}} \tag{44.56}$$

See Hall (1986) for details.

Note that $\hat{\text{var}}(\hat{\phi})$ can be obtained from the empirical variance of the B bootstrap estimates $\hat{\phi}^{*b}$ and that , $\hat{\text{var}}(\hat{\phi}^{*b})$ may be estimated from the empirical variance of the R double bootstrap estimates $\hat{\phi}^{**r}$.

Clearly, the computations above require a full double bootstrap. An alternative is to use the significantly less computationally demanding fast double bootstrap (FDB) approximation. In this regard, Davidson and Trokic (2020) offer an algorithm that relies on the inverse relationship between p-values and confidence intervals and only requires $R = 1$ double bootstrap evaluations to compute empirical quantiles.

While the FDB is computationally appealing, care should be taken with its use as Chang and Hall (2015) show that the FDB does not improve asymptotically on the single bootstrap for the construction of confidence intervals.

Killian's Unbiased Confidence Interval

The Kilian (1998) confidence intervals are specifically designed for VAR impulse response estimates. In particular, this approach explicitly corrects for the bias and skewness in the impulse response estimator that arises due to insufficient observations.

We skip over most of the details of the Kilian computation. Roughly, however, the procedure involves:

- Estimating a VAR to obtain coefficients and then running a residual bootstrap to obtain bootstrap coefficient estimates which are used to compute a bias correction adjustment to the original estimates.
- Using the bias corrected coefficients as the basis of a residual double bootstrap to bias correct each of the bootstrap coefficients.
- Constructing the impulse responses using the bias corrected bootstrap coefficients.
- Deriving the $(\alpha/2)$ and $(1-\alpha/2)$ empirical quantiles of the bias corrected bootstrap impulse responses.

Note that the nested bootstrap computations required for this method can be costly. A fast bootstrap approximation may be obtained by computing the bias adjustments only once per loop and using it across replications.

VEC Impulse Responses

The principles governing the computation of bootstrap CIs for VAR impulse responses remain valid for VEC impulse responses as well. Since every VAR(p) has an equivalent representation as a VEC($p-1$) process, one can simply convert bootstrap VAR estimates into their VEC equivalents, and proceed analogously to derive the VEC impulse responses using the latter coefficients. Alternatively, one can start from a VEC representation, obtain bootstrap coefficient estimates, and then integrate the VEC model to its VAR equivalent. The converted bootstrap coefficient estimates can then be used to derive bootstrap CIs for the corresponding VAR.

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.

Brüggemann, Ralf, Helmut Lütkepohl, and Pentti Saikkonen (2006). "Residual autocorrelation testing for vector error correction models," *Journal of Econometrics*, 134, 579–604.

Burbridge, John and Alan Harrison (1985). "A Historical Decomposition of the Great Depression to Determine the Role of Money," *Journal of Monetary Econometrics*, 16, 45–54.

Chang, Jinyuan, and Peter Hall (2015). "Double-bootstrap methods that use a single double-bootstrap simulation." *Biometrika*, 102, 203–214.

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.

Davidson, Russell, and James G MacKinnon (2002). "Fast double bootstrap tests of nonnested linear regression models." *Econometric Reviews*, 21, 419–429.

Davidson, Russell, and Mirza Trokic (2020). "The fast iterated bootstrap." J*ournal of Econometrics*, 218, 451–475.

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.

Edgerton, David and Ghazi Shukur (19991). "Testing autocorrelation in a system perspective," *Econometric Reviews*, 18, 343–386.

Efron, Bradley, and Robert J Tibshirani (1993). *An Introduction To The Bootstrap*. Monographs on Statistics and Applied Probability, 57, 1–436.

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.

Hall, Peter (1986). "On the bootstrap and confidence intervals." *The Annals of Statistics,* 14, 1431–1452.

Hall, Peter (1992). *The Bootstrap and Edgeworth Expansion*. Springer-Verlag: New York.

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 Statistic*s, 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.

Kilian, Lutz (1998). "Small-sample confidence intervals for impulse response functions." *Review of Economics and Statistics*, 80, 218–230.

Lütkepohl, Helmut (1990). "Asymptotic distributions of impulse response functions and forecast error variance decompositions of vector autoregressive models." *The Review Of Economics And Statistics*, 72, 116–125.

Lütkepohl, Helmut (1991). *Introduction to Multiple Time Series Analysis*, New York: Springer-Verlag.

Lütkepohl, Helmut (2007). *New 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.

Martin, Vance, Stan Hurn, and David Harris (2013). *Econometric Modelling with Time Series.* New York: Cambridge University Press.

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.

Rao, C. Radhakrishna (1973). *Linear Statistical Inference and Its Applications, Second Edition*. New York: John Wiley & Sons.

Rubio-Ramirez, F., Daniel F. Waggoner, and Tao Zha (2010). "Structural Vector Autogregressions: Theory of Identification and Algorithms for Inference," *The Review of Economic Studies*, 77, 665–696.

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, Christopher (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.

Chapter 45. Bayesian VAR Models

In Chapter 44. "Vector Autoregression and Error Correction Models", we outlined the estimation of Vector Autoregression (VAR) models, which are frequently used in the study of macroeconomic data. Since VARs frequently require estimation of a large number of parameters, over-parameterization (too few observations to estimate the parameters of the model) is a common problem.

One approach to solving this problem is *shrinkage*, where we impose restrictions on parameters to reduce the parameter set. Bayesian VAR (BVAR) methods (Litterman 1986; Doan, Litterman, and Sims 1984; Sims and Zha 1998; Koop and Korobilis 2010; Giannone, Lenza and Primiceri 2014) are a popular approach for achieving shrinkage, since Bayesian priors provide a logical and consistent method of imposing parameter restrictions.

The remainder of this discussion describes the estimation of VARs with Bayesian restrictions shrinkage. We first describe the set of EViews tools for estimating and working with BVARs and provide examples of the approach. This first section assumes that you are familiar with the various methods outlined in the literature. The remaining section outlines the methods in somewhat more detail.

Estimating a Bayesian VAR in EViews

To estimate a Bayesian VAR in EViews, click on **Quick/Estimate VAR...** or type `var` in the command window to bring up the **VAR Specification** dialog. Select the **Bayesian VAR** as the **VAR type** in the radio buttons on the left-hand side of the dialog.

The dialog will change to the BVAR version of the VAR Specification dialog. As with a standard VAR, you may use the **Basics** page to list of endogenous variables, the included lags, and any exogenous variables, and to specify the estimation sample:

The three BVAR specific tabs, **Prior type** and **Hyper-parameters** and **Options**, allow you to customize your specification. The following discussion of the settings on these three tabs assumes that you are familiar with the basics of the various prior types and associated settings. For additional detail, see "Technical Background" on page 912.

Prior Type

The **Prior type** tab lets you specify the type of prior you wish to use, whether to include dummy observations, and options for calculating the initial residual covariance matrix.

You may use the drop-down menu to choose between **Litterman/Minnesota**, **normal-Flat**, **normal-Wishart**, **independent normal-Wishart**, **Sims-Zha normal-flat**, **Sims-Zha normal-Wishart**, and **Giannone, Lenza & Primiceri** priors.

The **Dummy observations** check boxes allow you to include dummy/additional observations to the data matrices of the VAR. The **Sum-of-coefficients** setting adds observations to account for possible unit-root issues, while the **Dummy-Initial-observations** setting adds observations to account for possible cointegration issues (see "Dummy Observations" on page 923).

For the priors other than **normal-Flat** or n**ormal-Wishart**, you may select options for estimating the initial (or prior) residual covariance matrix. A dropdown menu allows you to specify the type of estimate used to calculate the initial covariance:

- **Univariate AR** estimates a univariate AR model (with number of lags matching those specified for the VAR) for each endogenous variable, then constructs the residual

covariance matrix as a diagonal matrix with diagonal elements equal to the residual variance from the estimated univariate models.

- **Diagonal VAR** uses the covariance from an estimated classical VAR model, but zeros out the off-diagonals.
- **Full VAR** uses the covariance from an estimated classical VAR model.
- **AR(1)** is computed in the same way as the **Univariate AR**, but only one lag is used.

The **Exogenous variables** radio buttons specify whether to include any exogenous variables specified in the VAR in the calculation of the initial covariance estimates.

- If you select either of the **Dummy observations** choices, the **Include dummy observations in calculation** checkbox specifies whether to include those observations in the initial covariance calculation.
- The **Degree of freedom correction** selection determines whether to use a degrees-of-freedom correction in the covariance estimate.

If the **Giannone, Lenza & Primiceri** prior type was selected in the drop-down menu, the **Optimize** checkbox selects whether to use the initial covariances as hyper-parameters, or as the starting values for an optimized hyper-parameter selection.

Finally, the **Sample:** box is used to specify the sample that is used to estimate the covariance. If left blank, the same sample as the overall VAR estimation.

Hyper-parameters

The **Hyper-parameters** tab allows specification of the hyper-parameters of the prior distribution.

If the **Giannone, Lenza & Primiceri** prior type was selected on the **Prior type** tab, the **Optimize** checkbox selects whether to use the values as hyper-parameters, or as the starting values for an optimized hyper-parameter selection.

Note tat only hyper-parameters that are available for the current prior type and choice of initial dummy observations are available for selection.

Options

The **Options** tab offers options for the Giannone, Lenza & Primiceri (GLP) prior and the inverse normal-Wishart prior. The GLP prior requires optimization of the hyper-parameters, and so options relating to the optimization algorithm; **Convergence tolerance** and **Maximum iterations**. The latter requires estimation through a Gibbs Sampler, and so offers options for the number of draws from the sampler, the percentage of draws to discard as burn-in draws, and the random number generator's seed value.

Post-Estimation Procedures

Most of the post-estimation views and procs available in standard VARs are also available for Bayesian VARs. However, with a few exceptions, these procedures simply take the estimate of the mean or covariance of the posterior distributions and applies these to classical techniques as if they were classical point estimates. To make this explicit, the view and proc menu items are labeled "Classical…"

Some procedures have been modified to allow a more Bayesian approach. These procedures make the distinction between the posterior residual moments (estimated moments of the posterior distribution and the empirical residual moments (residual moments calculated using the classical approach: $\hat{E} = Y - X\hat{B}$ using Equation (45.4) on page 913).

Residuals View

The residuals view (**View/Residuals**) now offers two choices for the residual covariance display: **Empirical Covariance Matrix** (the covariance of the empirical residuals) and **Posterior Covariance Matrix** (the covariance estimated from the posterior distribution).

Impulse Response

To produce impulse response graphs and tables, you may select **Proc/Impulse Response…** As with other VAR estimation methods, the first two tabs of the dialog (**Display** and **Impulse Definition**) will prompt you to select which impulses and responses to display in graph or table form, and to define the impulses you wish to produce.

The impulse response dialog for Bayesian VARs has an additional **BVAR Options** tab that contains Bayesian Specific settings:

The **Impulse method** dropdown menu selects the method for computing the impulses:

- **Bayesian sampling** will use a Gibbs or Metropolis-Hasting sampling algorithm to generate draws from the posterior distributions for the impulses (see below).
- **Classical – posterior res cov.** performs classical impulse response analysis where the impulses are calculated based upon the posterior residual covariance.
- **Classical – empirical res cov.** performs classical impulses response analysis where the impulses are calculated based upon the empirical residual covariance.

If **Bayesian sampling** is selected as the impulse method, the **Sampling** and **Output** options allow you to customize the sampling and store output:

- **Draws** sets the number of iterations of the sampler
- **Percent burn-in** sets the number of draws to discard.
- **Random seed** sets the random number generator's seed.
- The **Drop unstable draws** checkbox determines whether EViews will throw out and re-sample any draw which results in VAR coefficients that produce an unstable VAR. Note that checking this box may severely increase the time taken to produce the impulses if the VAR is close to unstable.
- The **Distribution graphs** check box will produce kernel-distribution plots of the draws for each variable pair and each period of the responses.
- The **Store each simulation in page** edit field allows you to enter the name of a page in which to store each individual draw of the sampler. If a name is entered into this box, a new page with that name will be created in the workfile, structured as a panel with one dimension representing then umber of response periods, and the other corresponding to the number of non-burned in sample draws. EViews will create a series for each variable pair in the new page.

Forecasting

Selection **Proc/Forecast...** displays the forecast dialog which offers a choice between Bayesian or classical forecasting:

By default, EViews will produce forecasts using **Bayesian sampling**, but you may select the **Classical point forecast** radio button to use the estimated coefficients to produce a point forecast.

If **Bayesian sampling** is selected, the **Sampling** and **Output** options allow you to customize the sampling and store output as in Bayesian impulse response calculation:

- **Draws** sets the number of iterations of the sampler
- **Percent burn-in** sets the number of draws to discard.
- **Random seed** sets the random number generator's seed.
- The **Drop unstable draws** checkbox determines whether EViews will throw out and re-sample any draw which results in VAR coefficients that produce an unstable VAR. Note that checking this box may severely increase the time taken to produce the impulses if the VAR is close to unstable.
- The **Distribution graphs** check box will produce kernel-distribution plots of the draws for each variable pair and each period of the responses.
- The **Store each simulation in page** edit field allows you to enter the name of a page in which to store each individual draw of the sampler. If a name is entered into this box, a new page with that name will be created in the workfile, structured as a panel with one dimension representing then umber of response periods, and the other corresponding to the number of non-burned in sample draws. EViews will create a series for each variable pair in the new page.

Impulse and Forecast Sampling

The Bayesian sampling approach to either impulse responses or forecasting from the VAR will differ depending upon the type of prior that was used during estimation.

For priors that have a closed-form solution for the posterior: Litterman, normal-flat, normal-Wishart, Sims-Zha normal-flat and Sims-Zha normal-Wishart, although the posterior distribution of the VAR coefficients and covariances can be calculated without simulation, the posterior distribution of non-linear functions of the coefficients and covariance will still require calculation through Monte-Carlo integration simulations.

Following Koop and Korobilis 2010, EViews performs this integration by pulling a draw of Sigma from its posterior distribution, and then using that draw of Sigma to pull a draw of alpha from its conditional posterior distribution, and, in the case of forecasting, pull a draw of the forecast errors. These draws are then used to compute an impulse or forecast. This process is repeated many times, and the overall mean of the responses or forecasts is computed.

A closed form solution for the posterior distributions of the independent normal-Wishart prior does not exist, so estimation is performed using a Gibbs' sampler. Consequently, calculation of impulses or forecasts from the VAR also require use of a Gibbs' sampler. Specifically, the sampler is run identically as to during estimation, but at each draw of alpha and Sigma during the sampler, an impulse response or forecast is computed from the draws. Note that the same random number generator seed used during estimation will be used for draws of alpha and Sigma. For forecasting the seed specified on the forecast dialog is used for draws of the residuals at each stage of the sampler.

The nature of the Giannone, Lenza and Primiceri prior requires the use of a Metropolis-Hastings sampler for estimation of the posteriors, and consequently for sampling impulse responses or forecasts. Specifically, the sampler is run identically as to during estimation, but at each draw of alpha and Sigma during the sampler, an impulse response or forecast is computed from the draws. Note that the same random number generator seed used during estimation will be used for draws of alpha and Sigma. For forecasting the seed specified on the forecast dialog is used for draws of the residuals at each stage of the sampler.

Drawing from Posterior Distribution

EViews allows you to take draws from the posterior distributions of either the coefficients or the residual covariance matrix by selecting **Proc/Draw from Coefficient Posterior** or **Proc/Draw from Residual Covariance Posterior**.

For all prior types apart from independent normal-Wishart, this procedure involves simply taking draws from the full posterior distributions. For the independent normal-Wishart prior, a Gibbs' sampler is employed.

Examples

Investment Example

To illustrate the Bayesian approach, we first estimate the coefficients of a VAR(2) model using the first differences of the logarithm of the DLINVESTMENT (investment), DLINCOME (income), and DLCONCUMPTION (consumption) example data. The raw data are provided in the EViews workfile "wgmacro.WF1". This data set was examined by Lütkepohl (2007, page 228).

Click on **Quick/Estimate VAR...** to open the main VAR specification dialog. In the VAR type box, select **Bayesian VAR** and in the **Endogenous Variables** box, type:

```
dlincome dlinvestment dlconsumption
```

Here, you will see the pre-filled settings including the variable names. You may change the default settings, but for now on, we assume that the default settings are used.

Basic Settings

Next, click on the **Prior type** tab to select the prior type for the VAR. By default, EViews will choose the **Litterman/Minnesota** for the **Prior type** and the **Univariate AR** estimate for the **Initial residual covariance**, but you can change the prior type and the initial covariance estimation option from the menus.

The **Hyper-parameters** tab shows the hyper-parameter settings. Note that the settings may vary depending on the prior type. We will only change the hyper-parameter, **Mu1** to be equal to zero (since the data are all differenced):

Click on **OK** to accept the dialog settings. EViews estimates the VAR and displays the results view. The results are shown below:

Var: P228 Workfile: WGMACRO::Wgmacro\

View | Proc | Object | Print | Name | Freeze | Estimate | Forecast | Stats | Impulse | Resids

Bayesian VAR Estimates

Bayesian VAR Estimates
Date: 01/24/19 Time: 12:32
Sample (adjusted): 1960Q4 1978Q4
Included observations: 73 after adjustments
Prior type: Litterman / Minnesota
Initial residual covariance: Univariate AR
Constant included in covariance calculation
Hyper-parameters: Mu1: 0, L1: 0.1, L2: 0.99, L3: 1, L4: inf
Standard errors in ( )

	DLINVESTM...	DLINCOME	DLCONSUM...
DLINVESTMENT(-1)	-0.090174	0.017196	-0.003778
	(0.07759)	(0.01977)	(0.01671)
DLINVESTMENT(-2)	-0.010162	0.005236	0.006872
	(0.04631)	(0.01177)	(0.00995)
DLINCOME(-1)	0.146227	-0.015809	0.064302
	(0.30489)	(0.07867)	(0.06607)
DLINCOME(-2)	0.057278	0.010228	0.045137
	(0.17924)	(0.04636)	(0.03884)
DLCONSUMPTION(-1)	0.262211	0.100037	-0.044995
	(0.35965)	(0.09224)	(0.07840)
DLCONSUMPTION(-2)	0.083518	0.002805	0.034859
	(0.21201)	(0.05437)	(0.04633)
C	0.008850	0.017922	0.017673
	(0.01159)	(0.00298)	(0.00252)
R-squared	0.056066	0.057402	0.094320
Adj. R-squared	-0.029746	-0.028288	0.011986
Sum sq. resids	0.152248	0.009645	0.007122
S.E. equation	0.048029	0.012089	0.010388
F-statistic	0.653357	0.669880	1.145575
Mean dependent	0.018229	0.020283	0.019802
S.D. dependent	0.047330	0.011921	0.010451

The heading information provides the basic information about the settings used in estimation, and the basic prior information, including hyper-parameter values. Just below the heading, the mean and standard errors from the posterior distribution are displayed.

The bottom portion of the output displays standard classical summary statistics for a VAR. Although they do not have a Bayesian interpretation, they are displayed for comparison purposes.

In his study of this data, Lütkepohl chose to use a diagonal VAR to estimate the initial residual covariance. We can replicate his results by setting the Diagonal VAR covariance type on the **Prior type** tab of the dialog:

Since the estimates in the third row of Table 5.3 of Lütkepohl's example may be obtained using EViews' default hyper-parameter values (other than the previously changed Mu1), click on **OK** to estimate the modified BVAR specification:

Var: P228 Workfile: WGMACRO::Wgmacro\

View | Proc | Object | Print | Name | Freeze | Estimate | Forecast | Stats | Impulse | Resids

Bayesian VAR Estimates

Bayesian VAR Estimates
Date: 01/24/19 Time: 12:42
Sample (adjusted): 1960Q4 1978Q4
Included observations: 73 after adjustments
Prior type: Litterman / Minnesota
Initial residual covariance: Diagonal VAR
Constant included in covariance calculation
Hyper-parameters: Mu1: 0, L1: 0.1, L2: 0.99, L3: 1, L4: inf
Standard errors in ( )

	DLINVESTM...	DLINCOME	DLCONSUM...
DLINVESTMENT(-1)	-0.096453	0.017885	-0.003959
	(0.07622)	(0.01924)	(0.01551)
DLINVESTMENT(-2)	-0.011337	0.005721	0.007308
	(0.04601)	(0.01159)	(0.00934)
DLINCOME(-1)	0.150439	-0.019351	0.069184
	(0.30206)	(0.07717)	(0.06183)
DLINCOME(-2)	0.061511	0.010797	0.049405
	(0.17965)	(0.04601)	(0.03677)
DLCONSUMPTION(-1)	0.297322	0.112852	-0.051735
	(0.36589)	(0.09293)	(0.07531)
DLCONSUMPTION(-2)	0.100237	0.003454	0.040620
	(0.22109)	(0.05615)	(0.04563)
C	0.007766	0.017691	0.017498
	(0.01147)	(0.00292)	(0.00235)
R-squared	0.060117	0.061359	0.102341
Adj. R-squared	-0.025327	-0.023972	0.020736
Sum sq. resids	0.151595	0.009605	0.007059
S.E. equation	0.047926	0.012064	0.010342
F-statistic	0.703587	0.719073	1.254100
Mean dependent	0.018229	0.020283	0.019802
S.D. dependent	0.047330	0.011921	0.010451

The results in the other rows of table Table 5.3 may be obtained by changing the hyper-parameters.

Alternate priors

To illustrate the importance of the prior selection, we estimate the same model using the Sims-Zha normal-Wishart prior, with a univariate AR estimate for the initial residual covariance, and the default hyper-parameter settings:

We can see that, for these data and VAR specification, changing the prior type did not have a large impact on any of the parameter values:

Var: P228 Workfile: WGMACRO::Wgmacro\

View | Proc | Object | Print | Name | Freeze | Estimate | Forecast | Stats | Impulse | Resids

Bayesian VAR Estimates

Bayesian VAR Estimates
Date: 01/24/19 Time: 13:19
Sample (adjusted): 1960Q4 1978Q4
Included observations: 73 after adjustments
Prior type: Sims-Zha (normal-Wishart)
Initial residual covariance: Univariate AR
Constant included in covariance calculation
Hyper-parameters: Mu1: 0, L0: 1, L1: 0.1, L3: 1, L4: inf, L5: inf, C3: 4
Standard errors in ( )

	DLINVESTM...	DLINCOME	DLCONSUM...
DLINVESTMENT(-1)	-0.090287	0.017385	-0.003825
	(0.63862)	(0.16069)	(0.13946)
DLINVESTMENT(-2)	-0.010185	0.005336	0.006982
	(0.38117)	(0.09591)	(0.08324)
DLINCOME(-1)	0.147456	-0.016053	0.065085
	(2.52562)	(0.63549)	(0.55154)
DLINCOME(-2)	0.058125	0.010181	0.045923
	(1.48811)	(0.37444)	(0.32497)
DLCONSUMPTION(-1)	0.265090	0.101195	-0.045209
	(2.97914)	(0.74961)	(0.65058)
DLCONSUMPTION(-2)	0.084906	0.002845	0.034751
	(1.76015)	(0.44289)	(0.38438)
C	0.008724	0.017898	0.017646
	(0.09581)	(0.02411)	(0.02092)
R-squared	0.056339	0.057909	0.095264
Adj. R-squared	-0.029449	-0.027736	0.013015
Sum sq. resids	0.152204	0.009640	0.007114
S.E. equation	0.048022	0.012086	0.010382
F-statistic	0.656726	0.676153	1.158238
Mean dependent	0.018229	0.020283	0.019802
S.D. dependent	0.047330	0.011921	0.010451

Macroeconomic Variables Example

As a second example of using Bayesian VARs in EViews, we work with a dataset created by Stock and Watson (2008) which provides data on a number of US macroeconomic variables. The dataset was obtained from Mark Watson's website: http://www.princeton.edu/~mwatson/publi.html, and a truncated version, with only variables used in this example, is provided as the EViews workfile "stockwatson.wf1". These data are used in Giannone, Lenza and Primiceri 2012, among other studies.

We will estimate a VAR using the same series as Giannone, Lenza and Primiceri's "Medium-scale model". This model contains seven series: Gross domestic product (GDP), the GDP deflator (GDPI), consumption (CONS), investment (INV), hours worked (HOURS), wages (WAGES), and the federal funds rate (FED). The first six variables were reported at a quar-

terly frequency. The federal funds rate is reported at a monthly rate, so the quarterly average is used to convert the data to quarterly. All variables have been logged and multiplied by four (to give annualized values), with the exception of the funds rate which is left in levels.

To estimate the VAR we click on **Quick/Estimate VAR…** to open the main VAR specification dialog. We select **Bayesian VAR** as the **VAR type**, and in the **Endogenous Variables** box we type:

```
gdp gdpi cons inv hours wages fed
```

To estimate the VAR with 5 lags, in the **Lags for endogenous** edit field we enter:

```
1 5
```

To begin we will estimate with a normal-Wishart prior, including dummy observations to account for both unit roots and cointegration:

We keep the hyper-parameters at their default values. Clicking **OK** will estimate the VAR and display the results (the top portion of which are displayed below):

Var: VAR01 Workfile: STOCKWATSON::quarterly\

View | Proc | Object | Print | Name | Freeze | Estimate | Forecast | Stats | Impulse | Resids

Bayesian VAR Estimates

Bayesian VAR Estimates
Date: 02/15/19 Time: 12:59
Sample (adjusted): 1960Q2 2006Q4
Included observations: 187 after adjustments
Prior type: Normal-Wishart with initial obs. and sum-of-coef. dummies
Initial residual covariance: Univariate AR
Constant included in covariance calculation
Hyper-parameters: Mu1: 1, Mu5: 0.1, Mu6: 0.1, C1: 0.1, C2: 0.1, C3: 8
Standard errors in ( )

	GDP	GDPI	CONS	INV	HOURS	WAGES	FED
GDP(-1)	1.011156	0.000112	0.003027	0.089184	0.018495	0.003163	0.005018
	(0.02696)	(0.01869)	(0.02387)	(0.11443)	(0.02421)	(0.02418)	(0.01774)
GDP(-2)	0.040102	0.018061	0.008468	0.300525	0.089240	-0.005749	0.015896
	(0.08790)	(0.06093)	(0.07783)	(0.37304)	(0.07891)	(0.07883)	(0.05783)
GDP(-3)	-0.037082	0.008107	-0.007942	-0.093589	0.022809	-0.010089	0.000939

Having estimated the VAR, we will look at the impulse responses of to a shock in GDP. We bring up the Impulse Response dialog by clicking the **Impulse** button on the VAR. We enter "GDP" in the **Impulses** box and keep all other settings at their defaults.

Clicking **OK** will produce a graph of the responses:

We would like to compare these results with those obtained assuming independent normal-Wishart priors.

We follow the steps above a second time, beginning with estimation, but after changing the prior to **Independent normal-Wishart**. The resulting responses are:

Technical Background

For brevity, we will assume the reader has some knowledge of Bayesian statistics, and is familiar with the notions of the prior and posterior distributions and marginal likelihoods.

To relate the general Bayesian framework to Vector Autoregressions, we first write the VAR as:

$$y_t = \delta_0 + \sum_{j=1}^{p} \Pi_j y_{t-j} + \gamma F_t + \epsilon_t \tag{45.1}$$

where

- $y_t = (y_{1t}, y_{2t}, \ldots, y_{Mt})'$ is an M vector of endogenous variables (*i.e.* the VAR has M variables and p lags)
- δ_0 is an M vector of intercept coefficients
- Π_j are $M \times M$ matrices of lag coefficients

- F_t is a $(d-1)$ vector of exogenous variables
- γ is a $M \times (d-1)$ matrix of exogenous coefficients
- ϵ_t is an M vector of errors where we assume $\epsilon_t \sim N(0, \Sigma)$

Collecting the right-hand side variables into the $(Mp+d)$ vector x_t and corresponding coefficients into B,

$$\begin{aligned} x_t &= (y_{t-1}', \ldots, y_{t-p}', 1, F_t')' \\ B &= (\Pi_1, \ldots, \Pi_p, \delta_0, \gamma)' \end{aligned} \tag{45.2}$$

we may rewrite Equation (45.1) as

$$y_t' = x_t' B + \epsilon_t' \tag{45.3}$$

In matrix form, we may stack the observations so we have:

$$Y = XB + E \tag{45.4}$$

where

$$\begin{aligned} Y &= (y_1, \ldots, y_T)' \\ X &= (x_1, \ldots, x_T)' \\ E &= (\epsilon_1, \ldots, \epsilon_T)' \end{aligned} \tag{45.5}$$

Y and E are $T \times M$, X is $T \times k$, and B is $k \times M$, for $k = (Mp+d)$ when p presample observations on y are available.

The multivariate normal assumption on the ϵ_t leads to a multivariate normal distribution for Y. Let $y = \text{vec}(Y')$ and $\beta = \text{vec}(B')$. Then $\text{vec}(B'X') = (X \otimes I_M)\beta$ and

$$(y|\beta) \sim N((X \otimes I_M)\beta, I_T \otimes \Sigma) \tag{45.6}$$

Bayesian estimation of VAR models centers around forming prior distributions for β and Σ.

Priors

EViews offers seven class of priors which have been popular in the BVAR literature, with each incorporating differing beliefs on the underlying model.

1. The Litterman/Minnesota prior: A normal prior on β with fixed Σ.
2. The normal-flat prior: A normal prior on β that is independent of the distribution for Σ.
3. The normal-Wishart prior: A normal prior on β and a Wishart prior on Σ.
4. The independent normal-Wishart prior. A normal prior on β and a Wishart prior on Σ, where each endogenous equation's coefficients' distributions may be independent from each other.

5. The Sims-Zha normal-flat. A structural VAR equivalent of the normal-flat prior.
6. The Sims-Zha normal-Wishart prior. A structural VAR equivalent of the normal-Wishart prior.
7. The Giannone, Lenza and Primiceri prior. A prior that treats the hyper-parameters as parameters that can be selected through an optimization procedure.

It is worth noting that EViews only offers conjugate priors (whose posterior has the same distributional family as the prior distribution). For five of these priors, this restriction allows for analytical calculation of the Bayesian VAR, rather than simulation-based estimation (*e.g.* the MCMC method) as is generally required.

Litterman Prior

Early work on Bayesian VAR priors was done by researchers at the University of Minnesota and the Federal Reserve Bank of Minneapolis (see Litterman (1986) and Doan, Litterman, and Sims (1984)), and these early priors are often referred to as the "Litterman prior" or the "Minnesota prior".

This family of priors is based on an assumption that Σ is known; replacing Σ with an estimate Σ. This assumption yields simplifications in prior elicitation and computation of the posterior.

Let $k = (Mp + d)$. The prior on the Mk elements of β is assumed to be:

$$\beta \sim N(\underline{\beta}_{Mn}, \underline{V}_{Mn}) \tag{45.7}$$

where $\underline{\beta}_{Mn} = \text{vec}(\overline{B}_{Mn}{}')$ and $\underline{V}_{Mn}$ is $Mk \times Mk$.

The Litterman prior yields a posterior of:

$$(\beta \mid y)_{Mn} \sim N(\bar{\beta}_{Mn}, \overline{V}_{Mn}) \tag{45.8}$$

where

$$\begin{aligned} \overline{V}_{Mn} &= (\underline{V}_{Mn}^{-1} + (\hat{\Sigma}^{-1} \otimes (X'X)^{-1}))^{-1} \\ \bar{\beta}_{Mn} &= \overline{V}_{Mn}(\underline{V}_{Mn}^{-1}\bar{\beta}_{Mn} + (\hat{\Sigma}^{-1} \otimes X')y) \end{aligned} \tag{45.9}$$

These posterior distributions are algebraically tractable and only require specification of the prior mean $\bar{\beta}_{Mn}$ and covariance $\overline{V}_{Mn}$, along with the estimate of Σ.

The combination of likelihood, prior and posteriors provide the marginal log-likelihood of:

$$\log(m_{Mn}(y)) = \tag{45.10}$$
$$-\frac{TM}{2}\log(2\pi) - \frac{T}{2}\log|\Sigma| - \frac{1}{2}(y-(X\otimes I_M)\bar{\beta}_{Mn})'(\Sigma^{-1}\otimes I_T)(y-(X\otimes I_M)\bar{\beta}_{Mn})$$
$$+ -\frac{1}{2}\log|\underline{V}_{Mn}| - \frac{1}{2}(\bar{\beta}_{Mn}-\underline{\beta}_{Mn})'\underline{V}_{Mn}^{-1}(\bar{\beta}_{Mn}-\underline{\beta}_{Mn}) + \frac{1}{2}\log|\overline{V}_{Mn}|$$

While there are infinitely many valid choices for $\underline{\beta}_{Mn}$ and $\underline{V}_{Mn}$, the standard implementations of the Litterman prior employ simple formations that produce easy computation of the posterior distributions.

First, $\underline{\beta}_{Mn}$ is set to a vector of nearly all zeros, with only the elements corresponding to the coefficient of a variable's own first lag being non-zero. Those own lag elements are generally set to 1 or 0, as specified by the hyper-parameter μ_1.

Second, $\underline{V}_{Mn}$ is assumed to be a diagonal matrix:

- The diagonal elements corresponding to the j-th endogenous variables in the i-th equation at lag l are given by:

$$\underline{V}^{l}_{Mn(i,j)} = \begin{cases} \left(\dfrac{\lambda_1}{l^{\lambda_3}}\right)^2 & \text{for } i = j \\ \left(\dfrac{\lambda_1\lambda_2\sigma_i}{l^{\lambda_3}\sigma_j}\right)^2 & \text{for } i \neq j \end{cases} \tag{45.11}$$

- The diagonal elements corresponding to all of the exogenous variables (including the constant) in the r-th equation are:

$$\underline{V}_{Mn(r)} = \lambda_4\sigma_r \tag{45.12}$$

  λ_1, λ_2, λ_3, and λ_4 are hyper-parameters, and σ_j is the square root of the corresponding diagonal element of Σ.

Normal-Flat Prior

The normal-flat relaxes the assumption of Σ being known but imposes no prior beliefs about Σ. Thus, Σ is taken from an "uninformative" or Jeffrey's prior, and as such has no prior distribution.

For β, the prior is from a normal distribution conditional on Σ:

$$\beta|\Sigma \sim N(\underline{\beta}_{NF}, \underline{H}_{NF}\otimes\Sigma) \tag{45.13}$$

where $\underline{\beta}_{NF} = \text{vec}(\bar{B}_{NF}')$ and $\underline{H}_{NF}$ is $k\times k$.

For Σ, all of the prior information is contained in the degrees-of-freedom parameter $\underline{\nu}_{NF}$.

The conditional posterior of β is also from a normal distribution:

$$(\beta \mid y, \Sigma)_{NF} \sim N(\mathrm{vec}(\overline{B}_{NF}'), \overline{H}_{NF} \otimes \Sigma) \tag{45.14}$$

where

$$\begin{aligned} \overline{H}_{NF} &= (\underline{H}_{NF}^{-1} + X'X)^{-1} \\ \overline{B}_{NF} &= \overline{H}_{NF}(\underline{B}_{NF} + X'Y) \end{aligned} \tag{45.15}$$

The posterior of Σ is inverse-Wishart:

$$\begin{aligned} (\Sigma \mid y)_{NF} &\sim IW(\overline{S}_{NF}, \overline{\nu}_{NF}) \\ \overline{S}_{NF} &= Y'Y + \underline{B}_{NF}'\underline{H}_{NF}^{-1}\underline{B}_{NF} - \overline{B}_{NF}'\overline{H}_{NF}^{-1}\overline{B}_{NF} \\ \overline{\nu}_{NF} &= \underline{\nu}_{NF} + T \end{aligned} \tag{45.16}$$

where IW denotes the inverse-Wishart distribution. Since Σ has an uninformative prior, the marginal log-likelihood is non-tractable.

The conditional posteriors only require specification of $\underline{\beta}_{NF}$, , $\underline{H}_{NF}$, and $\underline{\nu}_{NF}$.

$\underline{\beta}_{NF}$ is constructed identically to the Litterman prior as a vector of nearly all zeros, with only the elements corresponding to the coefficient of an endogenous variable's own first lag, optionally set to a non-zero value specified by the hyper-parameter μ_1.

The covariance terms are constructed as:

$$\begin{aligned} \underline{H}_{NF}^{-1} &= c_2 I_k \\ \underline{\nu}_{NF} &= c_3 \end{aligned} \tag{45.17}$$

Where c_2 and c_3 are hyper-parameters.

Normal-Wishart Prior

The normal-Wishart prior again relaxes the assumption of Σ being known, this time imposing a prior distribution.

For the normal-Wishart model, the priors for alpha and Sigma are given by:

$$\begin{aligned} \beta \mid \Sigma &\sim N(\underline{\beta}_{NW}, \underline{H}_{NW} \otimes \Sigma) \\ \Sigma &\sim IW(\underline{S}_{NW}, \underline{\nu}_{NW}) \end{aligned} \tag{45.18}$$

where $\underline{\beta}_{NW} = \mathrm{vec}(\underline{B}_{NW}')$, $\underline{H}_{NW}$ is $k \times k$, and $\underline{S}_{NW}$ is $M \times M$.

The conditional posterior for β is

$$
\begin{aligned}
(\beta \mid y, \Sigma)_{NW} &\sim N(\mathrm{vec}(\bar{B}_{NW}'), \bar{H}_{NW} \otimes \Sigma) \\
\bar{H}_{NW} &= (\underline{H}_{NW}^{-1} + X'X)^{-1} \\
\bar{B}_{NW} &= \bar{H}_{NW}(\underline{H}_{NW}^{-1}\underline{B}_{NW} + X'Y)
\end{aligned}
\tag{45.19}
$$

and the posterior of Σ is,

$$
\begin{aligned}
(\Sigma \mid y)_{NW} &\sim IW(\bar{S}_{NW}, \bar{\nu}_{NW}) \\
\bar{S}_{NW} &= Y'Y + \underline{S}_{NW} + \underline{B}_{NW}'\underline{H}_{NW}^{-1}\underline{B}_{NW} - \bar{B}_{NW}'\bar{H}_{NW}^{-1}\bar{B}_{NW} \\
&= \underline{S}_{NW} + \bar{E}_{NW}'\bar{E}_{NW} + (\underline{B}_{NW} - \bar{B}_{NW})'\underline{H}_{NW}^{-1}(\underline{B}_{NW} - \bar{B}_{NW}) \\
&= S_{OLS} + \underline{S}_{NW} + B_{OLS}'(X'X)B_{OLS} + \underline{B}_{NW}'\underline{H}_{NW}^{-1}\underline{B}_{NW} - \bar{B}_{NW}'\bar{H}_{NW}^{-1}\bar{B}_{NW} \\
\bar{\nu}_{NW} &= \underline{\nu}_{NW} + T
\end{aligned}
\tag{45.20}
$$

where B_{OLS} and S_{OLS} are estimates from a conventional VAR, and $\bar{E}_{NW} = Y - X\bar{B}_{NW}$. Note the three calculations of $\bar{S}_{NW}$ are algebraically equivalent, and all three appear in the literature.

The corresponding marginal log-likelihood is:

$$
\begin{aligned}
\log(m_{NW}(y)) &= -\frac{TM}{2}\log(2\pi) \\
&- \frac{M}{2}\log\left|\underline{H}_{NW}\right| + \frac{\underline{\nu}_{NW}}{2}\log\left|\underline{S}_{NW}\right| - \log\left(\Gamma_M\left(\frac{\underline{\nu}_{NW}}{2}\right)\right) \\
&+ \frac{M}{2}\log\left|\bar{H}_{NW}\right| - \frac{\bar{\nu}_{NW}}{2}\log\left|\bar{S}_{NW}\right| + \log\left(\Gamma_M\left(\frac{\bar{\nu}_{NW}}{2}\right)\right)
\end{aligned}
\tag{45.21}
$$

Again, these posterior distributions and likelihood are algebraically tractable and only require specification of the prior mean $\underline{\beta}_{NW}$ and covariance $\underline{H}_{NW}$, along with prior specifications of $\underline{S}_{NW}$ and $\underline{H}_{NW}$.

$\underline{\beta}_{NW}$ is constructed identically to the Litterman prior as a vector of nearly all zeros, with only the elements corresponding to the coefficient of an endogenous variable's own first lag, optionally set to a non-zero value specified by the hyper-parameter μ_1.

The covariance terms are constructed as:

$$
\begin{aligned}
\underline{S}_{NW}^{-1} &= c_1 I_M \\
\underline{H}_{NW}^{-1} &= c_2 I_k \\
\underline{\nu}_{NW} &= c_3
\end{aligned}
\tag{45.22}
$$

where c_1, c_2 and c_3 are hyper-parameters.

Independent Normal-Wishart Prior

The independent normal-Wishart prior is similar to the normal-Wishart prior, but allows cross-equation independence of the coefficient distributions and removes the dependence of β on Σ.

(It is worth noting that the independent normal-Wishart prior typically referenced in the literature (notably, Koop and Korobilis 2009) also allows for differing variables in each endogenous variable's equation (restricted VARs). EViews currently does not allow for restricted VARs under this prior.)

The independent normal-Wishart priors for β and Σ are:

$$\begin{aligned} \beta &\sim N(\underline{\beta}_{INW}, \underline{V}_{INW}) \\ \Sigma &\sim IW(\underline{S}_{INW}, \underline{\upsilon}_{INW}) \end{aligned} \tag{45.23}$$

where $\underline{\beta}_{INW} = \text{vec}(\underline{B}_{INW}')$, $\underline{V}_{INW}$ is $Mk \times Mk$, and $\underline{S}_{INW}$ is $M \times M$. Note the cross-equation independence of the prior implies that $\underline{V}_{INW}$ is of full size.

The unconditional posterior distributions from these priors are non-tractable. However, conditional posterior distributions are:

$$\begin{aligned} (\beta \mid y, \Sigma)_{INW} &\sim N(\bar{\beta}_{INW}, \bar{V}_{INW}) \\ (\Sigma \mid y, \beta)_{INW} &\sim IW(\bar{S}_{INW}, \bar{\nu}_{INW}) \end{aligned} \tag{45.24}$$

where

$$\begin{aligned} \bar{V}_{INW} &= \left(\underline{V}_{INW}^{-1} + (\hat{\Sigma}^{-1} \otimes (X'X)^{-1})\right)^{-1} \\ \bar{\beta}_{INW} &= \bar{V}_{INW}(\underline{V}_{INW}^{-1}\bar{\beta}_{INW} + (\hat{\Sigma}^{-1} \otimes X')y) \\ \bar{S}_{INW} &= \underline{S}_{INW} + E'E \\ \bar{\nu}_{INW} &= \underline{\upsilon}_{INW} + T \end{aligned} \tag{45.25}$$

These conditional posteriors can be used in a Gibbs' sampler to obtain properties of the unconditional posteriors. For estimation, this involves sequentially drawing from the conditional distributions (starting with an initial estimate of Σ to draw β), then, after discarding a number of burn-in draws, taking the mean of the draws of $\bar{\beta}_{INW}$ and $\bar{V}_{INW}$.

The marginal log-likelihood also does not have an analytical solution, but an estimate of its value can be computed from the outputs of the Gibbs' sampler using Chib's method (1995).

The prior matrices are constructed in a similar manner to the normal-Wishart prior; $\underline{\beta}_{INW}$ is a vector of nearly all zeros, with only the elements corresponding to the coefficient of a variable's own first lag, the hyper-parameter μ_1, optionally set to non-zero values.

The covariance terms are constructed as:

$$\begin{aligned} \underline{S}_{INW}^{-1} &= c_2 I_M \\ \underline{V}_{INW}^{-1} &= c_3 I_{Mk} \\ \underline{\nu}_{INW} &= c_3 \end{aligned} \tag{45.26}$$

where c_1, c_2 and c_3 are hyper-parameters.

Sims-Zha Priors

Sims-Zha (1998), is one of the earliest expansions of the Litterman prior to allow for an unknown Σ.

(Note that the Sims-Zha approach simultaneously uses dummy observations to account for unit-root and cointegration issues ("Dummy Observations" on page 923), alongside a specific Bayesian prior form. We have separated these two concepts to allow dummy observations to be used with any prior, however it should be noted that researchers referring to the "Sims-Zha" priors generally imply use of both their prior and the inclusion of dummy observations.)

The construction of their prior employs the structural form of the VAR. To begin recall the standard reduced form VAR representation:

$$y_t = \delta_0 + \sum_{j=1}^{p} \Pi_j y_{t-j} + \gamma F_t + \epsilon_t \tag{45.27}$$

It follows that

$$A_0 y_t = A_0 \delta_0 + \sum_{j=1}^{p} A_0 \Pi_j y_{t-j} + A_0 \gamma F_t + A_0 \epsilon_t \tag{45.28}$$

or

$$A_0 y_t = D + \sum_{j=1}^{p} A_j y_{t-j} + G F_t + u_t \tag{45.29}$$

which is the structural representation of the VAR. We may write this equation as:

$$y_t' A_0' = x_t' A + u_t' \tag{45.30}$$

where

$$A = (A_1, \ldots, A_p, D, G)' \tag{45.31}$$

and we assume with $u \sim N(0, A_0 \Sigma A_0')$.

In matrix form, we may stack the observations so we have:

$$YA_0' = XA + U \tag{45.32}$$

Y and X are defined as before, U is $T \times M$, X is $T \times k$, and A is $k \times M$, for $k = (Mp + d)$ when p presample observations on y are available.

Sims-Zha place a conditional prior on A given A_0, using either the normal-flat or normal-Wishart distributions.

For both distributions, the conditional prior of α is given by:

$$\alpha | A_0 \sim N(\underline{\alpha}_{SZ}, \underline{H}_{SZ} \otimes \Sigma) \tag{45.33}$$

where $\underline{\alpha}_{SZ} = \text{vec}(\overline{A}_{SZ}')$ and $\underline{H}_{SZ}$ is $k \times k$.

For the normal-Wishart, the prior on Σ is:

$$\Sigma \sim IW(\underline{S}_{SZ}, \bar{\nu}_{SZ}) \tag{45.34}$$

where $\underline{S}_{SZ}$ is $M \times M$

The resulting conditional posterior for α is then:

$$\begin{aligned} (\alpha | y, A_0, \Sigma)_{SZ} &\sim N(\text{vec}(\overline{A}_{SZ}'), \overline{H}_{SZ} \otimes \Sigma) \\ \overline{H}_{SZ} &= (\underline{H}_{SZ}^{-1} + X'X)^{-1} \\ \overline{A}_{SZ} &= \overline{H}_{SZ}(\underline{H}_{SZ}^{-1}\underline{A}_{SZ} + X'Y) \end{aligned} \tag{45.35}$$

And the posterior for Σ is:

$$\begin{aligned} (\Sigma | y)_{SZ} &\sim IW(\bar{S}_{SZ}, \bar{\nu}_{SZ}) \\ \bar{\nu}_{SZ} &= \underline{\nu}_{SZ} + T \end{aligned} \tag{45.36}$$

where, for the normal-flat prior,

$$\bar{S}_{SZ} = Y'Y + \underline{A}_{SZ}'\underline{H}_{SZ}^{-1}\underline{A}_{SZ} - \overline{A}_{SZ}'(\underline{H}_{SZ}^{-1} + X'X)\overline{A}_{SZ} \tag{45.37}$$

and for the normal-Wishart prior,

$$\begin{aligned} \bar{S}_{SZ} &= Y'Y + \underline{S}_{SZ} + \underline{A}_{SZ}'\underline{V}_{SZ}^{-1}\underline{A}_{SZ} - \overline{A}_{SZ}'\overline{H}_{SZ}^{-1}\overline{A}_{SZ} \\ &= \underline{S}_{SZ} + \overline{U}_{SZ}'\overline{U}_{SZ} + (\underline{A}_{SZ} - \overline{A}_{SZ})'\underline{H}_{SZ}^{-1}(\underline{A}_{SZ} - \overline{A}_{SZ}) \\ &= S_{OLS} + \underline{S}_{SZ} + A_{OLS}'(X'X)A_{OLS} + \underline{A}_{SZ}'\underline{H}_{SZ}^{-1}\underline{A}_{SZ} - \overline{A}_{SZ}'\overline{H}_{SZ}^{-1}\overline{A}_{SZ} \end{aligned} \tag{45.38}$$

where A_{OLS} and S_{OLS} are estimates obtained from a conventional VAR, and $\overline{U}_{SZ} = YA_0' - X\overline{A}_{SZ}$.

The expressions for the posterior distribution do not require specification of a prior on A_0. However, formulation of the marginal log-likelihood does require a prior on A_0, and as such, EViews does not report a log-likelihood.

Sims-Zha suggest a form of $\underline{\alpha}_{SZ}$ and $\underline{H}_{SZ}$ along similar lines to the Litterman priors.

First, $\underline{\alpha}_{SZ}$ is set to a vector of nearly all zeros, with only the elements corresponding to the coefficient of a variable's own first lag, hyper-parameter mu_1, being non-zero.

Second $\underline{H}_{SZ}$ is assumed to be diagonal:

- The diagonal elements corresponding to the j-th endogenous variables in the i-th equation at lag l are given by:

$$H^{l}_{SZ(i,j)} = \left(\frac{\lambda_0 \lambda_1}{\sigma_j l^{\lambda_3}}\right)^2 \tag{45.39}$$

- The diagonal elements corresponding to the constants are:

$$H^{i}_{SZ(r)} = (\lambda_0 \lambda_4)^2 \tag{45.40}$$

- The diagonal elements corresponding to the remaining exogenous are:

$$H^{i}_{SZ(s)} = (\lambda_0 \lambda_5)^2 \tag{45.41}$$

  λ_0, λ_1, λ_3, λ_4 and λ_5 are hyper-parameters, and σ_j is the square root of the corresponding diagonal element of Σ.

Note that the Sims-Zha prior treats the elements of the covariance prior slightly differently from the Litterman prior—there is no cross-variable relative weight term, λ_2, and there are different weights for the constant compared with other exogenous variables.

Giannone, Lenza and Primiceri (GLP) Prior

The Giannone, Lenza and Primiceri (GLP) approach construction of the prior, and its hyper-parameters, in a different manner than the previously outlined methods. GLP begin with a normal-Wishart prior where the specification of the coefficient covariance is akin to a modified Litterman approach. They note that since the marginal likelihood can be derived analytically, and the likelihood is a function of the hyper-parameters, rather than specifying exact values for those parameters, and optimization algorithm can be used to find the optimal specification of the prior.

The prior takes the form:

$$\begin{aligned} \beta | \Sigma &\sim N(\underline{\beta}_{GLP}, \underline{H}_{GLP} \otimes \Sigma) \\ \Sigma &\sim IW(\underline{S}_{GLP}, \bar{\nu}_{GLP}) \end{aligned} \tag{45.42}$$

where

$$\underline{V}^{l}_{GLP(i,j)} = \begin{cases} \left(\dfrac{\lambda_1}{l^{\lambda_3}\phi_j}\right)^2 & \text{for the } j\text{-th endogenous regressor} \\ \infty & \text{for all exogenous variables} \end{cases} \tag{45.43}$$

$\underline{\beta}_{GLP}$ is again all zero other than for an endogenous variable's own lag.

The covariance terms are constructed so that $\underline{S}_{GLP}$ is diagonal, with each of the diagonal elements a hyper-parameter that is either fixed at an initial covariance estimate or optimized. $\underline{\nu}_{GLP}$ is a prespecified scalar hyper-parameter:

$$\underline{\nu}_{GLP} = c_3 \tag{45.44}$$

The conditional posterior for β is then:

$$\begin{aligned} (\beta \mid y, \Sigma)_{GLP} &\sim N(\mathrm{vec}(\overline{B}_{GLP}'), \overline{H}_{GLP} \otimes \Sigma) \\ \overline{H}_{GLP} &= (\underline{H}^{-1}_{GLP} + X'X)^{-1} \\ \overline{B}_{GLP} &= \overline{H}_{GLP}(\underline{H}^{-1}_{GLP}\underline{B}_{GLP} + X'Y) \end{aligned} \tag{45.45}$$

And the posterior for Σ is:

$$(\Sigma \mid y)_{GLP} \sim IW(\overline{S}_{GLP}, \overline{\nu}_{GLP})$$

$$\begin{aligned} \overline{S}_{GLP} &= Y'Y + \underline{S}_{GLP} + \underline{B}_{GLP}'\underline{H}^{-1}_{GLP}\underline{B}_{GLP} - \overline{B}_{GLP}'\overline{H}^{-1}_{GLP}\overline{B}_{GLP} \\ &= \underline{S}_{NW} + \overline{E}_{GLP}'\overline{E}_{GLP} + (\underline{B}_{GLP} - \overline{B}_{GLP})'\underline{H}^{-1}_{GLP}(\underline{B}_{GLP} - \overline{B}_{GLP}) \\ &= S_{OLS} + \underline{S}_{GLP} + B_{OLS}'(X'X)B_{OLS} + \underline{B}_{GLP}'\underline{H}^{-1}_{GLP}\underline{B}_{GLP} - \overline{B}_{GLP}'\overline{H}^{-1}_{GLP}\overline{B}_{GL} \\ \overline{\nu}_{GLP} &= \underline{\nu}_{GLP} + T \end{aligned} \tag{45.46}$$

where $\overline{E}_{GLP} = Y - X\overline{B}_{GLP}$.

The marginal log-likelihood is:

$$\begin{aligned} \log(m_{GLP}(y)) &= -\frac{TM}{2}\log(2\pi) \\ &- \frac{M}{2}\log\left|\underline{H}_{GLP}\right| + \frac{\underline{\nu}_{NW}}{2}\log\left|\underline{S}_{GLP}\right| - \log\left(\Gamma_M\left(\frac{\underline{\nu}_{GLP}}{2}\right)\right) \\ &+ \frac{M}{2}\log\left|\overline{H}_{GLP}\right| - \frac{\overline{\nu}_{NW}}{2}\log\left|\overline{S}_{GLP}\right| + \log\left(\Gamma_M\left(\frac{\overline{\nu}_{GLP}}{2}\right)\right) \end{aligned} \tag{45.47}$$

This likelihood is maximized with respect to the hyper-parameters to find the optimal parameter values.

Dummy Observations

Sims-Zha (1998) also popularized the idea of complimenting the prior by adding dummy observations to the data matrices in order to improve the forecasting performance of the Bayesian VAR. These dummy observations are made up of two components; the *sum-of-coefficient* component and the *dummy-initial-observation* component. Both sets of observations can be applied to any of the above priors.

Sum-of-coefficient

The sum-of-coefficient component of a prior was introduced by Doan, Litterman and Sims (1984) and demonstrates the belief that the average of lagged values of a variable can act as a good forecast of that variable. It also expresses that knowing the average of lagged values of a second variable does not improve the forecast of the first variable.

The prior is constructed by adding additional observations at the top (pre-sample) of the data matrices. Specifically, the following observations are constructed:

$$
\begin{aligned}
Y_1^d &= \begin{bmatrix} \mu_5 \bar{y}_{1,0} & 0 & \dots & 0 \\ 0 & \mu_5 \bar{y}_{2,0} & 0 & \dots \\ \dots & 0 & \dots & 0 \\ 0 & \dots & 0 & \mu_5 \bar{y}_{M,0} \end{bmatrix} \\
X_1^d &= \begin{bmatrix} (Y_1^d, \dots, Y_1^d)_{M \times Mp} & 0_{M \times 1} \end{bmatrix}
\end{aligned}
\tag{45.48}
$$

where $\bar{y}_{1,0}$ is the mean of the i-th variable over the first p-th periods.

These observations introduce correlation between each variable in each equation. μ_5 is a hyper-parameter. If $\mu_5 = 0$, the observations are uninformative. As $\mu_5 \to \infty$, the observations imply unit roots in each equation, ruling out cointegration.

Dummy- initial-observation

The dummy-initial-observation component (Sims, 1993) creates a single dummy observation which is equal to the scaled average of initial conditions, and reflects the belief that the average of initial values of a variable is likely to be a good forecast of that variable. The observation is formed as:

$$
\begin{aligned}
Y_2^d &= \begin{bmatrix} \mu_6 \bar{y}_{1,0} & \mu_6 \bar{y}_{2,0} & \dots & \mu_6 \bar{y}_{M,0} \end{bmatrix} \\
X_2^d &= \begin{bmatrix} (Y_2^d, \dots, Y_2^d)_{1 \times Mp} & \mu_6 \end{bmatrix}
\end{aligned}
\tag{45.49}
$$

The hyper-parameter μ_5, is a scaling factor. If $\mu_6 = 0$, the observation is uninformative. As $\mu_6 \to \infty$, all endogenous variables in the VAR are set to their unconditional mean, the VAR exhibits unit roots without drift, and the observation is consistent with cointegration.

Augmented Data

The dummy observations are stacked on top of each other, and on top of the original data matrices:

$$\begin{bmatrix} Y_1^d \\ Y_2^d \\ Y \end{bmatrix} = \begin{bmatrix} X_1^d \\ X_2^d \\ X \end{bmatrix} B^* + E \tag{45.50}$$

It should be noted that the dummy observations require a correction to the marginal likelihood calculations for each prior.

A Note on Initial Residual Covariance Estimation

Most of the prior types allowed by EViews require an initial estimate of the residual covariance matrix, Σ to form the prior on β.

For the Litterman prior, this estimate is taken as known and is used as the final covariance estimate, as well as being used in the construction of the prior covariance for β, $\underline{V}_{Mn}$. For the independent normal Wishart prior, it is used as the starting point for the Gibbs sampler iterations.

Both Sims-Zha priors use the initial covariance estimate to form the prior covariance for β, $\underline{H}_{SZ} \otimes \Sigma$. The Giannone, Lenza and Primiceri prior uses the initial estimate of Σ to form the prior covariance matrix, either as a fixed element, or as the starting point for the optimization routine.

There is no standard method in the literature for estimating the residual covariance used to form the priors. Further, the choice of method can lead to stark differences in the posterior estimates, and is as important as the specification of hyper-parameters.

Researchers wishing to replicate existing studies should pay careful attention to the initial covariance estimation method. The wide variety of options offered by EViews should cover most cases.

References

Chib, Siddhartha, 1995. “Marginal Likelihood from the Gibbs Output,” *Journal of the American Statistical Association,* 90m 1313–1321.

Giannone, Domencio, Lenza, Michele, and Primiceri, Giorgio, E, 2012. “Prior Selection for Vector Autoregressions”, *ECB Working Paper Series No. 1494.*

Kadiyala, R, and Karlsson, S, 1997, "Numerical Methods for Estimation and Inference in Bayesian VAR Models," *Journal of Applied Econometrics,* 12, 99–132

Koop, Gary and Korobilis, Dimitris, 2010. "Bayesian Multivariate Time Series Methods for Empirical Macroeconomics," *Foundations and Trends in Econometrics.*

Sims, Christopher and Zha, Tao, 1998. "Bayesian Methods for Dynamic Multivariate Models," *International Economic Review*, 39, 949–968.

Stock, J. H., and M. W. Watson, 2002a. "Forecasting Using Principal Components from a Large Number of Predictors," *Journal of the American Statistical Association*, 97, 147–162.

Chapter 46. Mixed Frequency VAR

Standard Vector Autoregression (VAR) models require all variables in the VAR to be of the same frequency—all variables in the VAR are monthly, or all are quarterly, *etc.* The single frequency requirement is generally not met in practice as macroeconomic researchers typically possess variables of different frequencies.

The traditional solution to mixed frequency estimation uses frequency conversion methods to convert the high frequency variables to the lowest frequency in the VAR. This aggregation leads to a loss of information as multiple observations are combined into a single data point.

In univariate models this loss of information has been overcome through the use of MIDAS estimation (see Chapter 30. "Midas Regression" in *User's Guide II*). MIDAS creates multiple low frequency variables for each high frequency variable, allowing retention of the high frequency fidelity. The downside to this approach can be a dramatic increase in the number of variables in the model and the corresponding loss of degrees of freedom.

Although MIDAS models can alleviate this by using a polynomial weighting scheme to reduce the number of variables to a more manageable level, in cases where the number of added variables is small, as with quarterly and monthly data, the unweighted, or U-MIDAS approach (Ghysels 2016), is generally employed.

In VAR models, different approaches to alleviate the data aggregation problem have recently become popular. There are three broad categories of methods for estimating these mixed frequency VARs:

- State-space approaches such as Schorfheide and Song (2013).
- U-MIDAS approaches as outlined in Ghysels (2016).
- MIDAS with polynomial weights, Ghysels (2016).

EViews offers support for the unweighted U-MIDAS approach which has been expanded to include both Bayesian and Classical VAR techniques.

We begin with a discussion of how to estimate a Mixed Frequency VAR in EViews. Technical details are provided in "Technical Background" on page 934.

Estimating a Mixed Frequency VAR in EViews

Although a single EViews workfile page can contain only one frequency of data, workfiles support multiple frequencies by allowing multiple pages. To estimate a mixed frequency VAR (MFVAR) in EViews, your workfile must be setup with different frequency data on different pages.

To estimate a MFVAR in EViews, select the page with the *lowest* frequency data, and click on **Quick/Estimate VAR...** or type `var` in the command window to bring up the **VAR Specification** dialog. Select **Mixed Frequency** as the **VAR type** in the drop-down box on the left-hand side of the dialog.

The dialog will change to the MFVAR version of the VAR Specification dialog. As with a standard VAR, you may use the **Basics** page to list of endogenous variables, the included lags, and any exogenous variables, and to specify the estimation sample:

Note that the **Endogenous variables** and **Exogenous variables** boxes should be used to enter the names of the series that are located on the current *lowest* frequency page. EViews does not allow for exogenous variables in a differing frequency.

The **High frequency variables** edit field is used to specify the higher-frequency endogenous variables. The syntax for high frequency variables is *pagename\seriesname* where *pagename* is the name of the page containing the series, and *seriesname* is the name of the series. Note also that series expressions are allowed, *e.g.* "mypage\log(x)".

The **Estimation type** radio buttons change the estimation type between simple U-MIDAS and Bayesian estimation.

Hyper-parameters

If **Bayesian** is selected as the **Estimation type**, the **Hyper-parameters** page of the dialog allows specification of the hyper-parameters used during estimation:

Options

The **Options** page offers options on the frequency conversion method used, as well as options for the sampler used during Bayesian estimation:

The **Number of observations** field in the **Frequency Conversion** box specifies the number of high frequency observations to use for each low frequency period. For example, when dealing with monthly and quarterly data, the field can be used to specify that only two months out of each quarter should be used, rather than all three. Note that in contrast to univariate MIDAS estimation, mixed frequency VAR estimation does not allow specification of high frequency observations that fall outside of the corresponding low frequency period.

If the field is left blank, EViews will use the minimum number of high-frequency observations per low frequency period. For monthly to quarterly data this will be three. For seven-day daily to monthly, this will be 28 (if the monthly data cover a non-leap year February). Care should be taken if choosing to increase the value to 30 or 31, since for a number of months the observation will be NA, and that NA will be propagated multiple times due to the lag nature of the VAR, resulting in a large number of observations being dropped from estimation.

The **First observations** and **Last observations** radio boxes specify whether to use the first set of high-frequency observations from each low frequency period, or the last. For example, with monthly and quarterly data, if two observations are being used, these radio buttons select whether the first two months in the quarter are used or the last two months.

The **Initial Covariance** and **Sampler** boxes are only available if **Bayesian** was selected as the **Estimation type** on the **Basics** tab.

Initial Covariance specifies the initial estimate of the residual covariance matrix used for formulation of the prior. Either the covariance matrix from a U-MIDAS classical VAR can be used, or an identity matrix.

Sampler specifies the number of draws of the Gibbs' Sampler, the percentage of draws to discard as burn-in, and the random number seed.

Post Estimation Procedures

Although **Proc/Forecast** can be used for mixed-frequency VARs, it should be noted that only forecasts for the low-frequency variables will be produced.

If the mixed-frequency VAR is estimated using Bayesian methods, the **View/Impulse Responses…** and **Proc/Forecast…** menu items will offer the Bayesian form of these procedures as outlined in Chapter 45. "Bayesian VAR Models".

An Example

As an example of using Mixed Frequency VARs in EViews, we will perform similar analysis to that in Ghysels (2016). The analysis involves monthly industrial production, inflation and unemployment data for the US between 1949 and 2011, and quarterly US GDP data over the same period. The data are provided in the two-page workfile "Kansfed.WF1".

Following Ghysels, we will estimate a mixed frequency VAR model with a single lag. Ensuring we have the **quarterly** page selected, we click on **Quick/Estimate VAR…** menu item and fill in the details of our VAR. To begin we will estimate a U-MIDAS model with the single quarterly variable, GDP, and three monthly variables:

Clicking on **OK** produces the result:

Var: UNTITLED Workfile: KANSFED::quarterly\

View | Proc | Object | Print | Name | Freeze | Estimate | Forecast | Stats | Impulse | Resids

Mixed Frequency VAR Estimates

Mixed Frequency VAR Estimates
Date: 02/12/19 Time: 08:19
Sample (adjusted): 1949Q2 2011Q4
Included observations: 251 after adjustments
Est type: U-MIDAS
Frequency conversion: First
Standard errors in ( ) & t-statistics in [ ]

	IP_1	IP_2	IP_3	INFL_1	INFL_2	INFL_3	UNEMP_1	UNEMP_2	UNEMP_3	GDP
IP_1(-1)	-0.169740	-0.319103	-0.363007	-0.026159	-0.001162	0.002922	0.000483	0.012255	0.021883	-0.218015
	(0.06443)	(0.09659)	(0.13080)	(0.02285)	(0.03396)	(0.04484)	(0.01024)	(0.01339)	(0.01678)	(0.05090)
	[-2.63448]	[-3.30360]	[-2.77526]	[-1.14469]	[-0.03421]	[0.06517]	[0.04713]	[0.91501]	[1.30414]	[-4.28303]
IP_2(-1)	-0.201394	-0.357907	-0.504241	0.021627	-0.091963	-0.083282	0.046570	0.072868	0.093076	-0.208785
	(0.10641)	(0.15953)	(0.21603)	(0.03774)	(0.05609)	(0.07405)	(0.01692)	(0.02212)	(0.02771)	(0.08407)
	[-1.89253]	[-2.24344]	[-2.33408]	[0.57300]	[-1.63969]	[-1.12463]	[2.75298]	[3.29409]	[3.35842]	[-2.48343]
IP_3(-1)	1.325540	1.431397	1.425214	0.032059	0.150372	0.171003	-0.055970	-0.088859	-0.106035	0.432182
	(0.07221)	(0.10826)	(0.14660)	(0.02561)	(0.03806)	(0.05025)	(0.01148)	(0.01501)	(0.01881)	(0.05705)
	[18.3560]	[13.2219]	[9.72177]	[1.25172]	[3.95097]	[3.40292]	[-4.87576]	[-5.91951]	[-5.63813]	[7.57544]
INFL_1(-1)	-0.192972	0.160473	0.122214	0.038855	0.183060	0.166587	0.017043	0.010173	0.009479	0.088347
	(0.19510)	(0.29249)	(0.39608)	(0.06920)	(0.10283)	(0.13577)	(0.03101)	(0.04056)	(0.05081)	(0.15414)
	[-0.98909]	[0.54865]	[0.30856]	[0.56151]	[1.78027]	[1.22701]	[0.54954]	[0.25083]	[0.18655]	[0.57317]
INFL_2(-1)	0.629418	-0.352145	-0.750712	-0.337800	-0.734052	-0.665105	-0.086698	0.000830	-0.017136	0.164765
	(0.33691)	(0.50509)	(0.68396)	(0.11949)	(0.17757)	(0.23445)	(0.05356)	(0.07003)	(0.08774)	(0.26617)
	[1.86821]	[-0.69720]	[-1.09759]	[-2.82692]	[-4.13394]	[-2.83687]	[-1.61882]	[0.01185]	[-0.19529]	[0.61902]
INFL_3(-1)	-0.470254	0.075620	0.416060	1.293264	1.523835	1.459440	0.071449	0.001613	0.024308	-0.295087
	(0.22030)	(0.33026)	(0.44723)	(0.07813)	(0.11611)	(0.15330)	(0.03502)	(0.04579)	(0.05737)	(0.17404)
	[-2.13464]	[0.22897]	[0.93031]	[16.5519]	[13.1245]	[9.52011]	[2.04029]	[0.03523]	[0.42368]	[-1.69550]
UNEMP_1(-1)	-0.102279	0.719532	2.246071	0.160064	0.316864	0.387693	-0.039384	-0.058090	-0.080903	1.086391
	(0.40387)	(0.60548)	(0.81991)	(0.14324)	(0.21286)	(0.28105)	(0.06420)	(0.08396)	(0.10518)	(0.31907)
	[-0.25324]	[1.18837]	[2.73941]	[1.11741]	[1.48860]	[1.37945]	[-0.61344]	[-0.69191]	[-0.76916]	[3.40482]
UNEMP_2(-1)	-0.491032	-0.527665	-1.082383	-0.411676	-0.995390	-0.958318	0.205843	0.002780	-0.018652	-1.237504
	(0.58746)	(0.88071)	(1.19261)	(0.20836)	(0.30962)	(0.40881)	(0.09338)	(0.12212)	(0.15300)	(0.46411)
	[-0.83585]	[-0.59914]	[-0.90757]	[-1.97580]	[-3.21488]	[-2.34419]	[2.20424]	[0.02277]	[-0.12191]	[-2.66638]
UNEMP_3(-1)	0.667567	-0.045994	-0.903346	0.267763	0.697560	0.582167	0.805016	1.016698	1.041913	0.237347
	(0.50560)	(0.75798)	(1.02642)	(0.17932)	(0.26647)	(0.35184)	(0.08037)	(0.10510)	(0.13168)	(0.39944)
	[1.32035]	[-0.06068]	[-0.88009]	[1.49318]	[2.61774]	[1.65464]	[10.0162]	[9.67351]	[7.91273]	[0.59420]
GDP(-1)	0.074211	0.280546	0.431746	-0.018483	-0.049048	-0.089134	-0.024142	-0.046122	-0.082763	0.841704
	(0.07232)	(0.10843)	(0.14683)	(0.02565)	(0.03812)	(0.05033)	(0.01150)	(0.01503)	(0.01884)	(0.05714)
	[1.02609]	[2.58744]	[2.94054]	[-0.72056]	[-1.28675]	[-1.77102]	[-2.09989]	[-3.06779]	[-4.39396]	[14.7310]
C	-0.412398	-0.578305	-0.766537	-0.105714	-0.036511	0.078153	0.272232	0.341607	0.522622	0.125116
	(0.37389)	(0.56053)	(0.75904)	(0.13261)	(0.19706)	(0.26019)	(0.05944)	(0.07772)	(0.09737)	(0.29539)
	[-1.10298]	[-1.03171]	[-1.00987]	[-0.79718]	[-0.18528]	[0.30037]	[4.58031]	[4.39520]	[5.36713]	[0.42357]
R-squared	0.960314	0.907589	0.829703	0.978222	0.950991	0.915371	0.987100	0.978289	0.965686	0.880089
Adj. R-squared	0.960472	0.907957	0.830382	0.978308	0.951186	0.915708	0.987152	0.978375	0.965823	0.880567
Sum sq. resids	346.3454	778.4207	1427.413	43.56864	96.20717	167.7194	8.751911	14.96620	23.49127	216.1718
S.E. equation	1.174675	1.761044	2.384723	0.416630	0.619109	0.817438	0.186730	0.244185	0.305926	0.928031
F-statistic	607.3668	246.5131	122.2899	1127.422	487.0516	271.4896	1920.670	1130.982	706.3883	184.2223
Log likelihood	-396.5625	-498.1962	-574.2935	-136.3886	-235.8055	-305.5569	65.04713	-2.286427	-58.86615	-337.4064
Akaike AIC	3.247510	4.057341	4.663693	1.174411	1.966577	2.522366	-0.430654	0.105868	0.556702	2.776147
Schwarz SC	3.402012	4.211843	4.818195	1.328913	2.121079	2.676868	-0.276152	0.260370	0.711204	2.930649
Mean dependent	2.988788	2.993225	3.010187	3.552193	3.560096	3.561486	5.804781	5.801992	5.809960	3.138497
S.D. dependent	5.908357	5.804645	5.790311	2.828817	2.802181	2.815547	1.647367	1.660517	1.654817	2.685347

Determinant resid covariance (dof adj.)	1.98E-07
Determinant resid covariance	1.26E-07
Log likelihood	-1567.894
Akaike information criterion	13.36967
Schwarz criterion	14.91469
Number of coefficients	110

Each of the monthly variables, IP, INFL and UNEMP have been transformed into three separate variables in the quarterly VAR, one variable for each month in the quarter. The "_i" suffix is used to distinguish each month.

In his paper, rather than estimating with U-MIDAS, Ghysels estimates using a Bayesian prior, which we can replicate by clicking **Estimate** to bring up the **VAR Specification** dialog and selecting **Bayesian** as the **Estimation type**:

Clicking on **OK** produces:

Var: UNTITLED Workfile: KANSFED::quarterly\

View | Proc | Object | Print | Name | Freeze | Estimate | Forecast | Stats | Impulse | Resids

Mixed Frequency VAR Estimates

Mixed Frequency VAR Estimates
Date: 02/12/19 Time: 08:36
Sample (adjusted): 1949Q2 2011Q4
Included observations: 251 after adjustments
Est type: Bayesian
Frequency conversion: First
Initial covariance: Classical VAR
Hyper-parameters: Lambda: 5, UpsilonHL: 0.5, UpsilonLH: 0.5, RhoH: 0.2, RhoL: 0, Kappa: 1
10000 draws, 10% burnin, seed=1
Standard errors in ( )

	IP_1	IP_2	IP_3	INFL_1	INFL_2	INFL_3	UNEMP_1	UNEMP_2	UNEMP_3	GDP
IP_1(-1)	-0.252423	-0.445662	-0.532491	-0.012557	0.023719	0.029064	0.009793	0.024444	0.038727	-0.296421
	(0.05778)	(0.08455)	(0.11541)	(0.02255)	(0.03343)	(0.04413)	(0.00998)	(0.01298)	(0.01604)	(0.04480)
IP_2(-1)	0.042470	-0.050393	-0.151450	0.011119	-0.118923	-0.108062	0.031998	0.055712	0.068753	-0.043821
	(0.09415)	(0.13742)	(0.18861)	(0.03736)	(0.05538)	(0.07309)	(0.01648)	(0.02145)	(0.02648)	(0.07504)
IP_3(-1)	1.150757	1.244115	1.249213	0.030058	0.154199	0.172798	-0.051441	-0.085736	-0.101180	0.357213
	(0.06103)	(0.08975)	(0.12362)	(0.02470)	(0.03665)	(0.04850)	(0.01105)	(0.01439)	(0.01770)	(0.04886)
INFL_1(-1)	-0.172182	-0.009208	-0.175140	-0.039999	0.059507	0.013980	0.023399	0.025317	0.033593	0.013394
	(0.14457)	(0.20032)	(0.26851)	(0.06521)	(0.09547)	(0.12685)	(0.02859)	(0.03651)	(0.04383)	(0.05827)
INFL_2(-1)	0.251381	-0.286141	-0.482327	-0.031725	-0.252847	-0.097970	-0.082852	-0.009442	-0.036456	0.000788
	(0.19504)	(0.27057)	(0.37434)	(0.10340)	(0.15070)	(0.20243)	(0.04814)	(0.06126)	(0.07235)	(0.06400)
INFL_3(-1)	-0.124142	0.160106	0.420447	1.064326	1.164154	1.041991	0.062080	-0.002035	0.021296	-0.065384
	(0.12524)	(0.18445)	(0.25947)	(0.06772)	(0.09947)	(0.13375)	(0.03158)	(0.04037)	(0.04791)	(0.05854)
UNEMP_1(-1)	-0.294072	-0.103432	0.603936	0.073328	0.193490	0.162664	0.049316	0.086495	0.122880	0.034026
	(0.24100)	(0.31849)	(0.43464)	(0.12530)	(0.18072)	(0.24189)	(0.05778)	(0.07319)	(0.08621)	(0.03490)
UNEMP_2(-1)	0.231742	0.226309	0.026465	-0.116232	-0.553858	-0.319258	0.190566	-0.028731	-0.069631	0.011516
	(0.25028)	(0.33875)	(0.47107)	(0.14879)	(0.21258)	(0.29158)	(0.07953)	(0.10122)	(0.11787)	(0.03501)
UNEMP_3(-1)	0.118434	-0.012148	-0.420009	0.055872	0.375436	0.162714	0.732384	0.905375	0.892138	0.018184
	(0.16896)	(0.26904)	(0.39052)	(0.11709)	(0.17223)	(0.24081)	(0.06703)	(0.08613)	(0.10030)	(0.03501)
GDP(-1)	0.063136	0.277917	0.425467	-0.017019	-0.046007	-0.087374	-0.025019	-0.046677	-0.083049	0.816566
	(0.06659)	(0.09714)	(0.13211)	(0.02575)	(0.03822)	(0.05034)	(0.01130)	(0.01471)	(0.01820)	(0.05273)
C	-0.180453	-0.265956	-0.380253	-0.085951	-0.016738	0.110704	0.269691	0.334734	0.507929	0.337233
	(0.30771)	(0.38687)	(0.51040)	(0.13085)	(0.19315)	(0.25466)	(0.05663)	(0.07274)	(0.08719)	(0.23142)
R-squared	0.958750	0.904774	0.823820	0.977280	0.948655	0.912041	0.986907	0.977864	0.964837	0.870466
Adj. R-squared	0.958914	0.905153	0.824522	0.977371	0.948859	0.912391	0.986960	0.977952	0.964977	0.870982
Sum sq. resids	359.9953	802.1363	1476.728	45.45180	100.7938	174.3193	8.882726	15.25933	24.07283	233.5196
S.E. equation	1.197599	1.787669	2.425568	0.425538	0.633695	0.833366	0.188121	0.246565	0.309690	0.964550
F-statistic	583.3856	238.4827	117.3679	1079.671	463.7460	260.2604	1892.014	1108.774	688.7167	168.6721
Mean dependent	2.988788	2.993225	3.010187	3.552193	3.560096	3.561486	5.804781	5.801992	5.809960	3.138497
S.D. dependent	5.908357	5.804645	5.790311	2.828817	2.802181	2.815547	1.647367	1.660517	1.654817	2.685347

These results are in line with those reported in Table 5 of Ghysels and show a reasonable difference from the U-MIDAS/classical VAR we first estimated – for example the coefficient on the first monthly observation of IP on a single quarterly lag of itself changes from -0.17 to -0.25.

Technical Background

For notational simplicity we will assume a mixed frequency VAR model with no constant or exogenous variables. We also assume only two frequencies of data, low and high, and that there are m high frequency periods per low frequency period.

The VAR contains k_L variables y_L^i, observed at low frequency and k_H variables y_H^i observed at the high frequency. Let

- $y^{i}_{L,\, t_L}$ represent the i-th low frequency variable observed during low-frequency period t_L
- $y^{i}_{H,\, t_L,\, t}$ represent the i-th high frequency variable observed in the t-th high-frequency period during low-frequency period t_L

Stacking the k_L and k_H variables into the matrices Y_L and Y_H, respectively, and for brevity ignoring the intercepts and exogenous variables, we may write the VAR as:

$$\begin{bmatrix} Y_{H,\, t_L,\, 1} \\ Y_{H,\, t_L,\, 2} \\ \dots \\ Y_{H,\, t_L,\, m} \\ Y_{L,\, t_L} \end{bmatrix} = \sum_{j=1}^{p} \begin{bmatrix} \Pi_j^{1,\,1} & \Pi_j^{1,\,2} & \dots & \Pi_j^{1,\,m} & \Pi_j^{1,\,m+1} \\ \Pi_j^{2,\,1} & \dots & \dots & \dots & \Pi_j^{2,\,m+1} \\ \dots & \dots & \dots & \dots & \dots \\ \Pi_j^{m,\,1} & \dots & \dots & \Pi_j^{m,\,m} & \Pi_j^{m,\,m+1} \\ \Pi_j^{m+1,\,1} & \dots & \dots & \Pi_j^{m+1,\,m} & \Pi_j^{m+1,\,m+1} \end{bmatrix} \begin{bmatrix} Y_{H,\, t_L-j,\, 1} \\ Y_{H,\, t_L-j,\, 2} \\ \dots \\ Y_{H,\, t_L-j,\, m} \\ Y_{L,\, t_{L-j}} \end{bmatrix} + \begin{bmatrix} E_{H,\, t_L,\, 1} \\ E_{H,\, t_L,\, 2} \\ \dots \\ E_{H,\, t_L,\, m} \\ E_L,\, t_L \end{bmatrix} \tag{46.1}$$

where $\Pi_j^{a,\,b}$ is $k_H \times k_H$, $\Pi_j^{m+1,\,b}$ is $k_L \times k_H$, and $\Pi_j^{a,\,m+1}$ is $k_H \times k_L$, for all j, $a,\, b = 1, \dots, m$, and $\Pi_j^{m+1,\,m+1}$ is $k_L \times k_L$.

The U-MIDAS approach constructs these data and simply uses classical VAR least squares to estimate B, the stacked matrix of coefficients.

Bayesian mixed-frequency VAR estimation (Ghysels 2016), analogously to Bayesian VAR estimation, requires you to specify prior distributions for $\beta = \mathrm{vec}(B')$ and the residual covariance matrix, Σ ("Technical Background" on page 934).

$$\begin{aligned} \beta &\sim N(\underline{\beta},\, \underline{V}) \\ \Sigma &\sim IW(\underline{S},\, \bar{\nu}) \end{aligned} \tag{46.2}$$

These priors coupled with the standard likelihood for a VAR model yields conditional posteriors for β and Σ of:

$$(\beta \,|\, y,\, \Sigma) \sim N(\bar{\beta},\, \bar{V}) \tag{46.3}$$

$$(\Sigma \,|\, y,\, \beta) \sim IW(\bar{S},\, \bar{\nu}) \tag{46.4}$$

where

$$\begin{aligned} \bar{V} &= (\underline{V}^{-1} + (\hat{\Sigma}^{-1} \otimes (X'X)^{-1}))^{-1} \\ \bar{\beta} &= \bar{V}(\underline{V}^{-1}\underline{\beta} + (\hat{\Sigma}^{-1} \otimes X')y) \\ \bar{S} &= \underline{S} + E'E \\ \bar{\nu} &= \underline{\upsilon} + T \end{aligned} \tag{46.5}$$

The prior on β is constructed like the Litterman prior in standard Bayesian VARs.

As with Litterman, the prior mean of β is assumed to be all zero, other than the own first lag terms. However the Ghysels prior differs slightly by allowing a different hyper-parameter for the own lag term depending on whether a variable is high or low frequency. Another crucial difference is that the variables are assumed to follow an AR(1) process in the high-frequency, so that when estimating in low-frequency space, the parameter is exponential.

The prior variance of β is similar to a Litterman prior, but with subtle differences to allow for the cross-frequency variances.

Thus, for the coefficient priors we have:

Element	Mean	Variance	Dimension
$\Pi_1^{a,\,m}$	$\mathrm{diag}(\rho_H^a)$	$\frac{\lambda^2}{a^2}$	$k_H \times k_H,\ 1 \le a \le m$
$\Pi_1^{m+1,\,m+1}$	$\mathrm{diag}(\rho_L^m)$	λ^2	$k_L \times k_L$
$\Pi_j^{a,\,b}$	0	$\frac{\lambda^2}{((j-1)m+(m-b-a))^2}$	$k_H \times k_H,\ 1 \le a \le m,$ $1 \le b \le m-1$
$\Pi_1^{m+1,\,b}$	0	$\nu_{HL}\frac{\lambda^2}{((j-1)m+b)^2}S_{HL}$	$k_L \times k_H,\ 1 \le b \le m$
$\Pi_j^{a,\,b}$	0	$\frac{\lambda^2}{((j-1)m+a)^2}$	$k_H \times k_H,\ 1 \le a \le m,$ $j > 1$
$\Pi_j^{a,\,m+1}$	0	$\nu_{HL}\frac{\lambda^2}{j^2}S_{HL}$	$k_H \times k_L,\ 1 \le a \le m$
$\Pi_j^{m+1,\,m+1}$	0	$\frac{\lambda^2}{j^2}$	$k_L \times k_L,\ j > 1$

where ρ_H, ρ_L, ν_{HL} and ν_{LH} are hyper-parameters, and S_{LH} is a $k_L \times k_H$ matrix with the (i,j)-th element equal to $\sigma_{i,\,L}^2/\sigma_{j,\,L}^2$, $i = 1, \ldots, k_L$, $j = 1, \ldots, k_H$ where the σ are taken from a univariate AR(1) estimation.

The covariance term between the i-th endogenous variable and an exogenous variable is set to $\kappa\sigma_i$.

$\underline{S}$ is formed by taking a scaled identity matrix or by scaling an initial estimate of the variance taken from a classical least squares U-MIDAS estimation:

$$\underline{S} = c_1\hat{S} \tag{46.6}$$

where $\hat{S} = I_{m(K_L + K_H)}$ or $\hat{S} = S_{OLS}$.

References

Ghysels, Eric, 2016. "Macroeconomics and the reality of mixed frequency data," *Journal of Econometrics*, 193, 294–314.

Schorfheide, F. and Song, D, 2013. "Real-Time Forecasting with a Mixed-Frequency VAR", *NBER Working Paper No. 19712*.

Chapter 47. Switching VAR

We describe here EViews tools for estimating *switching VAR* models—VAR models with nonlinearities arising from discrete changes in regime. As in their single equation counterparts, we consider settings with both independent and Markov switching where the sample separation into regimes is not observed.

The following discussion describes only the basic features of switching VARs. For a book length discussion of switching VAR models, see Krolzig (1997). In addition, switching models have a long history in economics that is detailed in numerous surveys (Goldfeld and Quandt, 1973, 1976; Hamilton, 1994; Frühwirth-Schnatter, 2006), and we encourage you to explore these resources for additional discussion.

Background

We offer a brief background for switching VAR models. Recall the standard k-dimensional, VAR(p) process

$$y_t = A_1 y_{t-1} + \cdots + A_p y_{t-p} + \mu_t + \epsilon_t \tag{47.1}$$

where

- $y_t = (y_{1t}, y_{2t}, \ldots, y_{Kt})'$ is a $k \times 1$ vector of endogenous variables,
- $A_1, \ldots, A_p$ are $k \times k$ matrices of lag coefficients to be estimated,
- μ_t is a $k \times 1$ vector of intercepts,
- $\epsilon_t = (\epsilon_{1t}, \epsilon_{2t}, \ldots, \epsilon_{Kt})'$ is a $k \times 1$ white noise innovation process, with $E(\epsilon_t) = 0$, $E(\epsilon_t \epsilon_t{}') = \Sigma_\epsilon$, and $E(\epsilon_t \epsilon_s{}') = 0$ for $t \neq s$.

The vector of innovations are contemporaneously correlated with full rank matrix Σ_ϵ, but are uncorrelated with their leads and lags of the innovations and assumed to be uncorrelated with all of the right-hand side variables.

Switching Specification

Following Krolzig, we modify Equation (47.1) to allow for regime change so that y_t follows a VAR process that depends on the value of an unobserved discrete state variable s_t. We assume there are $m = 1, \ldots, M$ possible regimes, and we are said to be in regime m in period t when $s_t = m$.

As in Krolzig, the VAR regime dependence is assumed to take one of two forms:

- switching intercept (SI):

$$y_t = \sum_{j=1}^{p} A_j(s_t) y_{t-1} + \mu_t(s_t) + \epsilon_t \tag{47.2}$$

- switching mean (SM):

$$y_t - \mu_t(s_t) = \sum_{j=1}^{p} A_j(s_t)(y_{t-j} - \mu_{t-j}(s_{t-j})) + \epsilon_t \tag{47.3}$$

Regime change in the SI model produces smooth changes in the time series, while the SM specification produces an immediate jump in mean.

Further, we will assume that the errors are distributed as $\epsilon_t \sim N(0, \Sigma_\epsilon(s_t))$ for $s_t = 1, \ldots, M$, with density function

$$f(\epsilon|\Sigma) = \frac{1}{(2\pi)^{k/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}\epsilon'\Sigma^{-1}\epsilon\right) \tag{47.4}$$

Common practice divides the parameters in the VAR specification into three groups: the intercept parameters μ, the endogenous variables parameters A, the error variance parameters Σ, Typically, only a subset of the groups is allowed to vary across regimes. For example, a common restriction is that only the intercepts, or only the intercepts and the error variances are regime specific.

Lastly, we may allow for exogenous variables by defining the intercepts as functions of exogenous variables and coefficients:

$$\mu_t = C_m x_t \tag{47.5}$$

- C_m is a $k \times d$ matrix of exogenous variable coefficients to be estimated,
- $x_t = (x_{1t}, x_{2t}, \ldots, x_{dt})'$ is a $d \times 1$ vector of exogenous variables,

so the μ are parameterized in terms of exogenous variable parameters $C = (C_1, \ldots, C_M)$. The remainder of our discussion will be in terms of μ, but the analysis can extend to underlying C parameters.

Regime Dependence

Central to the analysis of a switching VAR model is the notion that error term depends on an unobserved state variable. The nature of this state dependence differs dramatically between the switching intercept (SI) and switching mean (SM) specifications introduced earlier.

This difference creates some notational challenges. To facilitate discussion, the remainder of our discussion will organized around a new variable S_t that is defined in terms of current and lagged s_t and has M^* possible states.

We define S_t for both specifications in the discussion below.

Switching Intercept (SI) Specification

We may use Equation (47.2) to obtain an expression for the switching intercept error in terms of the observed data and current unobserved state:

$$\epsilon_t(s_t) = y_t - \sum_{j=1}^{p} A_j(s_t) y_{t-1} - \mu_t(s_t) \tag{47.6}$$

Note that the expression for ϵ_t depends only on the current state. Accordingly, we have $S_t = s_t$ and $M^* = M$. It follows that $S_t = m$ is equivalent to the statement $s_t = m$.

Switching Mean (SM) Specification

Similarly, we may use Equation (47.3) to obtain an expression for the error in terms of the observed data and a set of current and past unobserved states:

$$\epsilon_t(S_t) = y_t - \mu_t(s_t) - \sum_{j=1}^{p} A_j(s_t)(y_{t-1} - \mu_{t-1}(s_{t-j})) \tag{47.7}$$

where $S_t = (s_t, s_{t-1}, \ldots, s_{t-p})$ is a $p+1$ dimensional state vector representing the current and p previous regimes, with $M^* = M^{p+1}$.

To simplify notation, in switching mean specifications $S_t = m$ should be interpreted as shorthand for S_t being equal to the m-th possible realization of the $p+1$ dimensional vector, as in

$$S_t = (a_0(m), a_{-1}(m), \ldots, a_{-p}(m)) \tag{47.8}$$

where $a_k(m)$ is the value of the k-th lagged state in the m-th possible realization, for $m = 1, \ldots, M^*$.

Log Likelihood

The likelihood contribution for a given observation may be formed by weighting the state specific multivariate normal density Equation (47.4) by the one-step ahead prediction of the probability of being in the given state:

$$L_t(\mu, A, \Sigma, \delta) = \sum_{m=1}^{M^*} f(\epsilon_t | S_t = m) \cdot P(S_t = m | \Im_{t-1}) \tag{47.9}$$

where $\epsilon_t(m) = (\epsilon_t | S_t = m)$ is obtained from the regime specific specifications Equation (47.6) and Equation (47.7).

$\mu = (\mu_1, \ldots, \mu_M)$, $A = (A_{11}, \ldots, A_{1p}, \ldots, A_{pM})$, $\Sigma = (\Sigma_1, \ldots, \Sigma_M)$, are the VAR parameters, δ are parameters that determine the regime probabilities.

Defining $\epsilon_t(m) = (\epsilon_t | S_t = m)$, we have the full normal mixture log-likelihood

$$l(\mu, A, \Sigma, \delta) = \sum_{t=1}^{T} \log \sum_{m=1}^{M^*} \left\{ \frac{1}{(2\pi)^{k/2} |\Sigma(m)|^{1/2}} \exp\left(-\frac{1}{2}\epsilon_t(m)'\Sigma(m)^{-1}\epsilon_t(m)\right) \right. \tag{47.10}$$
$$\left. \cdot P(S_t = m | \Im_{t-1}, \delta) \right\}$$

which may be maximized with respect to (μ, A, Σ, δ).

It is worth noting that the likelihood function for this normal mixture model is unbounded for certain parameter values. However, local optima have the usual consistency, asymptotic normality, and efficiency properties. See Maddala (1986) for discussion of this issue as well as a survey of different algorithms and approaches for estimating the parameters.

Given parameter point-estimates, coefficient covariances may be estimated using conventional methods, *e.g.*, inverse negative Hessian, inverse outer-product of the scores, and robust sandwich.

Regime Probabilities

To finish our likelihood specification, we must specify the regime probabilities function

$$P(S_t = i | S_{t-1} = j, S_{t-2} = k, \ldots)$$

There are two commonly employed forms: simple switching and Markov switching.

Simple Switching

The simple switching model features independent regime probabilities which do not depend on past states.

$$\begin{aligned} P(S_t = j | S_{t-1} = i, S_{t-2} = k, \ldots) &= P(S_t = j) \\ &= p_j(t) \end{aligned} \tag{47.11}$$

More generally, we may allow for varying probabilities by assuming that p_m is a function of vectors of exogenous observables G_{t-1} and coefficients δ parameterized using a multinomial logit specification:

$$P(S_t = m | \Im_{t-1}, \delta) \equiv p_m(G_{t-1}, \delta) = \frac{\exp(G_{t-1}'\delta_m)}{\sum_{j=1}^{M} \exp(G_{t-1}'\delta_j)} \tag{47.12}$$

for $\delta = (\delta_1, \delta_2, \ldots, \delta_M)$ with the identifying normalization $\delta_M = 0$. The special case of constant probabilities is handled by choosing G_{t-1} to be identically equal to 1.

Markov Switching

The first-order Markov assumption requires that the probability of being in a regime depends only on the previous state, so that

$$P(S_t = j | S_{t-1} = i, S_{t-2} = k, \ldots) = P(S_t = j | S_{t-1} = i) = p_{ij}(t) \tag{47.13}$$

Typically, these transition probabilities are assumed to be time-invariant so that $p_{ij}(t) = p_{ij}$ for all t, but this restriction is not required.

We may write these probabilities in a transition matrix

$$p(t) = \begin{bmatrix} p_{11}(t) & \ldots & p_{1M}(t) \\ . & \ldots & . \\ p_{M1}(t) & \ldots & p_{MM}(t) \end{bmatrix} \tag{47.14}$$

where the ij-th element represents the probability of transitioning from regime i in period $t-1$ to regime j in period t. (Note that some authors use the transpose of $p(t)$ so that all of their indices are reversed from those used here.)

As in the simple switching model, we may parameterize the probabilities in terms of a multinomial logit. Note that since each row of the transition matrix specifies a full set of conditional probabilities, we define a separate multinomial specification for each row i of the matrix

$$p_{ij}(G_{t-1}, \delta_i) = \frac{\exp(G_{t-1}'\delta_{ij})}{\sum_{s=1}^{M} \exp(G_{t-1}'\delta_{is})} \tag{47.15}$$

for $j = 1, \ldots, M$ and $i = 1, \ldots, M$ with the normalizations $\delta_{iM} = 0$.

Probability Prediction and Filtering

The likelihood function in Equation (47.10) depends on the one-step ahead predicted probabilities of being in a regime: $P(S_t = m | \Im_{t-1})$. Obtaining these predicted probabilities is central to the evaluation of the likelihood.

Of related interest are the contemporaneous estimates of the regime probabilities: $P(S_t = m | \Im_t)$. The observed value of the dependent variable provides information about which regime is in effect in a given period, and we may use this contemporaneous information to updated our estimates of the regime probabilities. The process by which the predicted $P(S_t = m | \Im_{t-1})$ probability estimates are updated to form $P(S_t = m | \Im_t)$ is commonly termed *filtering*.

In the following sections, we outline the basics of one-step ahead prediction and filtering for both the simple switching specification and Markov switching.

Simple Switching

One-step ahead prediction is straightforward for simple switching since the one-step ahead predicted probabilities are simply the specified probability functions:

$$P(S_t = j|\Im_{t-1}) = P(S_t = j)$$

In the switching intercept case, substituting the general form of the simple switching function Equation (47.12) into Equation (47.10), we get

$$l(\mu, A, \Sigma, \delta) = \sum_{t=1}^{T} \log \sum_{m=1}^{M} \left\{ \frac{1}{(2\pi)^{k/2}|\Sigma(m)|^{1/2}} \exp\left(-\frac{1}{2}\epsilon_t(m)'\Sigma(m)^{-1}\epsilon_t(m)\right) \right. \tag{47.16}$$

$$\left. \cdot\, p_m(G_{t-1}, \delta) \right\}$$

By Bayes' theorem and the laws of conditional probability, we have the filtering expressions:

$$\begin{aligned} P(S_t = m|\Im_t) &= P(S_t = m|\epsilon_t, \Im_{t-1}) \\ &= \frac{f(\epsilon_t|S_t = m, \Im_{t-1}) \cdot P(S_t = m|\Im_{t-1})}{f(\epsilon_t|\Im_{t-1})} \end{aligned}$$

Substituting, we obtain the filtering update

$$P(S_t = m|\Im_t) = \frac{f(\epsilon_t|S_t = m, \Im_{t-1}) \cdot p_m(G_{t-1}, \delta)}{\sum_{j=1}^{M^*} f(\epsilon_t|S_t = j, \Im_{t-1}) \cdot p_j(G_{t-1}, \delta)} \tag{47.17}$$

Note that in the switching model setting, the state variable is $(p+1)$-dimensional, so the above relationship does not apply. We must instead treat this model as a restricted form of the Markov switching model (as described below) where there no state dependence in the probability function.

Markov Switching

The Markov property of the transition probabilities implies that the expressions on the right-hand side of Equation (47.10) must be evaluated recursively.

Briefly, each recursion step begins with filtered estimates of the regime probabilities for the previous period. Given filtered probabilities, $P(S_{t-1} = m|\Im_{t-1})$, the recursion may broken down into three steps:

1. We first form the *one-step ahead predictions* of the regime probabilities using basic rules of probability and the Markov transition matrix:

$$P(S_t = m|\Im_{t-1}) = \sum_{j=1}^{M^*} P(S_t = m|S_{t-1} = j) \cdot P(S_{t-1} = j|\Im_{t-1}) \tag{47.18}$$

2. Next, we use these one-step ahead probabilities to form the one-step ahead joint densities of the data and regimes in period t:

$$f(\epsilon, S_t = m|\Im_{t-1}) = f(\epsilon_t|S_t = m, \Im_{t-1}) \cdot P(S_t = m|\Im_{t-1}) \tag{47.19}$$

3. The likelihood contribution for period t is obtained by summing the joint probabilities across unobserved states to obtain the marginal distribution of the observed data

$$L_t(\mu, A, \Sigma, \delta) = f(\epsilon_t|\Im_{t-1}) = \sum_{j=1}^{M^*} f(\epsilon_t, S_t = j|\Im_{t-1}) \tag{47.20}$$

4. The final step is to filter the probabilities by using the results in Equation (47.19) to update one-step ahead predictions of the probabilities:

$$P(S_t = m|\Im_t) = \frac{f(\epsilon_t, S = m|\Im_{t-1})}{\sum_{j=1}^{M^*} f(\epsilon_t, S_t = j|\Im_{t-1})} \tag{47.21}$$

These steps are repeated successively for each period, $t = 1, \ldots, T$. All that we require for implementation are the initial filtered probabilities, $P(S_0 = m|\Im_0)$, or alternately, the initial one-step ahead regime probabilities $P(S_1 = m|\Im_0)$. See "Initial Probabilities" on page 945 for discussion.

The likelihood obtained by summing the terms in Equation (47.20) yields

$$l(\mu, A, \Sigma, \delta) = \sum_{t=1}^{T} \log \sum_{m=1}^{M^*} \left\{ \frac{1}{(2\pi)^{k/2}|\Sigma(m)|^{1/2}} \exp\left(-\frac{1}{2}\epsilon_t(m)'\Sigma(m)^{-1}\epsilon_t(m)\right) \right. \tag{47.22}$$

$$\left. \cdot \left(\sum_{j=1}^{M^*} P(S_t = m|S_{t-1} = j) \cdot P(S_{t-1} = j|\Im_{t-1}) \right) \right\}$$

The likelihood may be maximized with respect to the parameters (μ, A, Σ, δ) using iterative methods. Coefficient covariances may be estimated using standard approaches.

Initial Probabilities

In the switching intercept form, the Markov switching filter requires initialization of the filtered regime probabilities in period 0, $P(s_0 = m|\Im_0)$.

There are a few ways to proceed. Most commonly, the initial regime probabilities are set to the ergodic (steady state) values implied by the Markov transition matrix (see, for example Hamilton (1999, p. 192) or Kim and Nelson (1999, p. 70) for discussion and results). The values are thus treated as functions of the parameters that determine the transition matrix.

Alternately, we may use prior knowledge to specify regime probability values, or we can be agnostic and assign equal probabilities to regimes. Lastly, we may treat the initial probabilities as parameters to be estimated.

Note that the initialization to ergodic values using period 0 information is somewhat arbitrary in the case of time-varying transition probabilities.

In the switching means setting, the Markov switching filter requires initialization of the vector of probabilities associated with the M^{p+1} dimensional state vector. We may proceed as in the uncorrelated model by setting M initial probabilities in period $(-p+1)$ as described above, and recursively applying Markov transition updates to obtain the joint initial probabilities for the M^* dimensional initial probability vector in period 0.

Again note that the initialization to steady state values using the period $(-p+1)$ information is somewhat arbitrary in the case of time-varying transition probabilities.

Smoothing

For the Markov switching specification, estimates of the regime probabilities may be improved by using all of the information in the sample. The *smoothed* estimates for the regime probabilities in period t use the information set in the final period, $\Im_T$, in contrast to the filtered estimates which employ contemporaneous information, $\Im_t$.

Intuitively, using information about future realizations of the dependent variable y_r $(r > t)$ improves our estimates of being in regime m in period t because the Markov transition probabilities link together the likelihood of the observed data in different periods.

Kim (2004) provides an efficient smoothing algorithm that requires only a single backward recursion through the data. Under the Markov assumption, Kim shows that the joint probability is given by

$$\begin{aligned} P(S_t = i, S_{t+1} = j | \Im_T) &= P(S_t = i | S_{t+1} = j, \Im_T) \cdot P(S_{t+1} = j | \Im_T) \\ &= \frac{P(S_t = i, S_{t+1} = j | \Im_t)}{P(S_{t+1} = j | \Im_t)} \cdot P(S_{t+1} = j | \Im_T) \end{aligned} \tag{47.23}$$

The key in moving from the first to the second line of Equation (47.23) is the fact that under appropriate assumptions, if S_{t+1} were known, there is no additional information about S_t in the future data $(y_{t+1}, \ldots, y_T)$.

The smoothed probability in period t is then obtained by marginalizing the joint probability with respect to S_{t+1}:

$$P(S_t = i | \Im_T) = \sum_{j=1}^{M^*} P(S_t = i, S_{t+1} = j | \Im_T) \tag{47.24}$$

Note that apart from the smoothed probability terms, $P(S_{t+1} = j | \Im_T)$, all of the terms on the right-hand side of Equation (47.23) are obtained as part of the filtering computations. Given the set of filtered probabilities, we initialize the smoother using $P(S_T = j | \Im_T)$, and

iterate computation of Equation (47.23) and Equation (47.24) for $t = T-1, \ldots, 1$ to obtain the smoothed values.

Specifying a Switching VAR in EViews

To estimate a switching VAR model, open the VAR estimation dialog and select **Switching VAR** from the **VAR type** dropdown menu. The dialog changes to show the settings for switching VARs:

There are two tabs in this dialog. The first tab is used for basic specification of your switching regression. The second tab contains options for modifying selected features of the specification and for controlling computational aspects of the estimation.

Specification

The first tab contains the basic specification. The **Estimation sample** edit field on the left-hand side and most of the edit fields on the right are familiar from standard VAR estimation. You should enter the endogenous variables in the **Endogenous variables** list, a list of lag pairs to specify the endogenous lags, and a list of Exogenous variables.

Switching Specification

The **Switching specification** section controls the specification of the regime probabilities.

- The **Switching** type dropdown allows you to choose between **Simple** and **Markov** switching. The default setting is to estimate a simple switching model. Select **Markov** to specify a Markov switching vector autoregressive (MS-VAR) model.
- You should specify the number of regimes $M > 1$ in the edit field. By default, EViews assumes that you have two regimes. Bear in mind that switching models with more than a few regimes may be difficult to estimate.
- You should specify the coefficients that are allowed to vary across regimes. Note that you may only choose between grouped settings: whether the lagged endogenous switch or not, whether the exogenous switch or not, and whether the error covariance matrix switches or not.

  By default, only the exogenous variables will be switching.
- Lastly the edit field on the bottom right-hand side specifies the **Probability regressors**. You may specify additional regressors that determine the unconditional regime probabilities (for simple switching) or the regime transition probability matrix (for Markov switching). By default, EViews will assume non-varying switching probabilities and set the list to include a single constant term.

  Important note: the data for the probability regressors that determine the transition or regime probabilities for period t, should be located in period t of the workfile. That is, the G_{t-1} data should be in period t of the workfile, not period $t-1$. You may, of course, employ standard EViews lag expressions to refer to data in the previous period.

Additional options for setting the initial regime probabilities and restricting the elements of the probability vector or transition matrix are located on the **Options** tab and described in "Switching Options" on page 949.

Options

Clicking on the **Options** tab displays options for modifying features of the switching specification and for controlling various aspects of the computation.

Switching Options

The **Switching** options section may be used to specify the initial state probabilities and any restrictions to the regime probability vector or transition matrix.

Recall that evaluation of the likelihood in Markov switching and SSAR models requires presample values for the filtered probabilities ("Initial Probabilities" on page 945). The **Initial regime probabilities** dropdown lets you choose the method of initializing these values (**Ergodic solution** (default), **Estimated**, **Uniform**, **User-specified**). If you select **User-specified**, you will be prompted for the name of an M-element vector in the workfile that contains the initial probabilities.

The **Probability restriction vector/Transition restriction matrix** edit field allows you to specify restrictions on the regime probabilities. Markov switching models, in particular, will sometime require restrictions on transition matrix probabilities. For example, we may have $p_{wr} = 0$ if it is impossible to transition directly from state w to state r. Similarly, if state w is an absorbing state, then $p_{ww} = 1$ and $p_{wr} = 0$ for $r \neq w$.

To specify restrictions, you should enter the name of an M-element vector in the workfile (for a SSAR model), or an $M \times M$ matrix in the workfile (for Markov switching) in the edit field. The vector or matrix should contain valid probability values for elements that are restricted and NAs for elements that are to be estimated. For example, in a three regime Markov switching model where state 3 is an absorbing state, you would have

$$\begin{bmatrix} NA & NA & NA \\ NA & NA & NA \\ 0 & 0 & 1 \end{bmatrix} \tag{47.25}$$

You should take care not to specify invalid or inconsistent restrictions. For example, rows of a Markov transition matrix may not be specified so that there is a single unrestricted cell since the adding up condition for the row places a restriction on that cell. Similarly, fixed values should be valid probabilities that do not have generate row sums greater than 1. EViews will detect these types of errors and will refuse to estimate the model.

Starting Value and Iteration Options

The **Iteration control** section contains the standard fields for setting the maximum number of iterations and convergence tolerance. By default, EViews will use the estimation settings from your global options.You may select the **Display settings** checkbox to display starting values and other estimation settings in the top section of your estimation output.

You may use the **Starting values** section to control the setting of initial parameter estimates. Switching regression models often have local roots and may be difficult to estimate so EViews offers a range of tools for choosing starting values.

Start Method

The **Start method** dropdown allows you to specify a basic method for choosing starting values (**EViews Supplied**, **.8 x EViews Supplied**, **.5 x EViews Supplied**, **.3 x EViews Supplied**, **Zero**, **User-Supplied**).

The EViews supplied methods employ simple least squares coefficient estimates or the specified fraction of those estimates. AR coefficients are arbitrarily initialized to zero.

The **Zero** and **User-Supplied** methods are self-explanatory, with the latter taken from the default coefficient vector specified in the dialog (typically the coefficient vector C).

Randomized Estimates

The first two edit fields under **Randomized estimates** allow you to choose random starting values based on the those specified in the **Start method** dropdown.

- EViews will randomly choose the number of random starts as specified in **No. of random starts** and for each random start perform the number of iterations as specified in **Iterations for starts**. The coefficients with the highest likelihood value will be chosen as the starting values.

For non user-supplied starting values, EViews will, by default, generate 25 sets of random starting values and refine each with 10 iterations before choosing the best set as the starting values. By default, there is no randomization of user-supplied values.

In addition to randomizing based on the initial values, you may randomize based on the final coefficient estimates. The **No. of random from final** edit field determines the number of random coefficients to try following estimation.

- The random starting values are chosen by taking the best estimated values to date and adding random normals with scale given by the **Random scale** fraction of the final coefficient standard deviations. The estimates with the highest likelihood become the final estimates.

  By default, EViews does not perform randomization based on the final coefficient estimates.

For both initial and final randomization, the random starting values are chosen by taking the base values and adding random normals with scale given by the **Random scale** fraction of the root of the estimated coefficient variances (or the scale fraction itself if the variances are not available). The random values will be generated using the **Generator** specified in the dropdown and the random **Seed** specified in the edit field. If a random seed is not specified, EViews will obtain one from a single draw from the generator.

Coefficient Covariance Options

EViews estimates the parameters of the likelihood using the Broyden, Fletcher, Goldfarb and Shanno (BFGS) method. By default, the coefficient covariance is computed using a non d.f. corrected inverse of the negative of the observed Hessian.

You may use the **Covariance method** dropdown to instruct EViews to use the Hessian, the outer product of the gradient (OPG) method, or a Huber-White robust covariance that forms a sandwich using the Hessian and gradients.

The **d.f. Adjustment** checkbox may be used to apply a d.f. correction.

Optimization Options

You may use the **Optimization method** dropdown menu to select a different method.

By default, EViews uses BFGS with Marquardt steps to obtain parameter estimates. You may use the **Optimization method** dropdown to choose between **BFGS**, **OPG - BHHH**, or **Newton-Raphson**. The **Step method** dropdown offers the choice of **Marquardt**, **Dogleg**, and **Line search**.

See "Optimization Method" on page 1314 for discussion.

The remainder of the section allows you to specify a convergence criterion and the maximum number of iterations, and to instruct EViews to display starting value and other optimization information in the output.

Post-Estimation Results

As with single equation switching estimation, EViews allows you to compute one-step ahead predicted, filtered, and smoothed estimates of the regime probabilities and to display graphs and tables of these results.

Additionally, many, but not all of the standard VAR views are available.

Switching Views

Once you have estimated your VAR, EViews offers a variety of views for examining your results.

Most of these routines are familiar tools for working with an estimated VAR. You may, for example, examine **Residuals** or **Structural Residuals** views for your estimated VAR, examine the **Impulse Response...** or **Historical Decomposition...** views or examine the **Lag Structure** of your VAR.

Since the presence of multiple regimes creates a few wrinkles, we offer a few comments on select views.

Regime Results

EViews offers specialized tools for examining the regime transition results and predicted regime probabilities.

Transition Results

To display the transition results dialog, select **View/Regime Results/Transition Results...** EViews offers to display different types of output: **Summary**, **Transition probabilities**, and **Expected durations**.

The default **Summary** display shows a table containing both the transition matrix and the expected durations (Kim and Nelson, 1999, p. 71-72) implied by the transition matrix. For example:

VAR: SVAR_SWITCH0
Date: 03/26/19 Time: 10:07
Transition summary: Constant simple switching
transition probabilities and expected durations
Sample: 1962Q1 1991Q4
Included observations: 120

Constant transition probabilities:
P(i, k) = P(s(t) = k | s(t-1) = i)
(row = i / column = k)

	1	2
1	0.831672	0.168328
2	0.831672	0.168328

Constant expected durations:

	1	2
	5.940789	1.202397

Here, we see results from a simple switching model with constant transition probabilities. Note that since the model assumes simple switching, the probabilities of being in regime 1 and regime 2 (approximately 0.83 and 0.17, respectively) do not depend on the origin state. These probabilities imply that the expected duration in a regime is roughly 5.9 quarters in regime 1 and 1.2 quarters in regime 2.

In models with varying transition probabilities, the transition probability summary will instead show the means and standard deviations of the transition probabilities and the expected durations.

The latter two output type choices may prove useful in models with time-varying transition probabilities:

- Selecting **Transition probabilities** allows you to see the transition matrix in every period. You will be offered a choice of displaying the transition probabilities in **Graph** (default), **Sheet**, or **Table** form.

  The **Graph** display shows a multiple graph showing each transition probability. For purposes of this display simple switching models are treated as restricted Markov switching models. Thus, a two-regime switching model will always show four separate graphs, one for each transition. Note that for constant transition probability models, each graph will be a straight line.

  The **Sheet** display shows the same results in spreadsheet format while the **Table** form displays results for each period in a table form similar to that used in the summary display. For constant probability models, the spreadsheet format will contain identical rows and the table form will show a single transition matrix.

- Lastly, you may display the **Expected durations** associated with the transition matrix in each period. The results may be displayed in **Graph** (default), **Sheet**, or **Table** form. For constant probability models, the table form will only show the single set of expected durations.

Regime Probabilities

To display the estimated regime probabilities select **View/Regime Results/Regime Probabilities...**

EViews offers a choice between the **One-step-ahead**, **Filtered**, and **Smoothed** probabilities. For simple switching models without AR terms, the filtered and smoothed results are the same. In addition, you may display the results in **Multiple graphs**, in a **Single graph**, or in **Sheet** form.

For graphical output you may also select which regimes to plot by entering the corresponding indices in the **Regimes to plot** edit field. While the default is to show the probabilities for all regimes, you may use the edit field to remove redundant probabilities or to focus on a specific regime of interest.

Other Views

The lag structure and the error structure of the VAR play a central role in a number of VAR views (lag structure, impulse response, variance decomposition, historical decomposition).

In cases where the dynamic coefficients and the error covariance matrix are regime independent, these views will produce results identical to those obtained in a standard VAR setting.

In cases where there are regime specific values, EViews will compute each set of regime specific results. For example, if there are regime specific VAR lag coefficients, EViews will display lag structure results or produce impulse responses for each regime.

Switching Procs

EViews offers several procs for working with switching VARs. Most of these procs are self-explanatory, but we offer brief comments about saving regime results to the workfile.

Make Regime Results

The **Make Regime Results** submenu offers routines for saving transition matrix results and predicted regime probabilities into series and matrices in the workfile.

Transition Results

To save transition probability or expected duration results in the workfile click on **Proc/ Make Regime Results/Make Transition Results...**

In addition to prompting you to choose between saving the **Transition probabilities** or the **Expected durations**, you must select an output format. By default, EViews will save the results in a group of series in the workfile. The series names will be formed using the base name specified in the edit field, as in “TPROB12”, “TPROB22”, etc. for transitions, and “TPROB1”, “TPROB2”, etc. for expected durations.

You may instead elect to save the results in a matrix. In this case, EViews will prompt you for the name of the matrix and for an observation at which to evaluate the transition matrix or expected duration. By default, the dialog will be filled with the first observation in the estimation sample.

Regime Probabilities

To save the regime probabilities, select **Proc/Make Regime Results/Make Regime Probabilities Group...**

Select the type of probability you wish to compute (**One-step-ahead**, **Filtered**, or **Smoothed**), and enter the names of series to hold the results, one for each probability you wish to retain.

Examples

To illustrate switching VAR estimation, we consider examples using quarterly GDP data for a subset of G7 countries obtained from the OECD National Accounts Statistics (https://data.oecd.org/gdp/quarterly-gdp.htm).

We first estimate a Markov switching intercept model with varying intercepts, and then a Markov switching mean model with regime specific intercepts and lagged endogenous.

Markov Switching Intercept

The data for this example consists of 1960q4–2018q4 quarterly percent change in GDP for Canada (DPCAN), Germany (DPDEU), Japan (DPJPN), and the US (DPUSA). The data may be found in the workfile "msvar_G7.WF1".

To estimate the first model, open the switching VAR dialog and enter the specification as depicted below:

The specification consists of a two-state Markov switching intercept VAR(2) with a single switching mean regressor C. The error variance is assumed to be common across the regimes. Adopting the general Krolzig (1997) taxonomy, we may term this an MSI-VAR(2) specification—Markov switching intercept, VAR(2). The only probability regressor is the constant C since we have time-invariant regime transition probabilities.

With the exception of the convergence tolerance which we set to 1e-8, we leave the rest of the settings at their default values. Click on **OK** to estimate the equation and display the estimation results.

The top portion of the output describes the estimation settings:

```
Markov Switching Intercepts VAR Estimates (BFGS / Marquardt steps)
Date: 03/26/19   Time: 11:55
Sample (adjusted): 1961Q4 2018Q4
Included observations: 229 after adjustments
Number of states: 2
Initial probabilities obtained from ergodic solution
Standard errors & covariance computed using observed Hessian
Random search: 25 starting values with 10 iterations using 1 standard deviation
      (rng=kn, seed=1551063419)
Convergence achieved after 39 iterations
Standard errors in ( ) & z-statistics in [ ]
```

(Bear in mind that if you attempt replicate this estimation using the default settings, you may obtain different results due to the different set of random starting values. You may use the random number generator seed settings to obtain the same starting values.)

The middle section displays the coefficients for the regime specific mean and the invariant error distribution coefficients with columns corresponding to endogenous variables. We see, in the differences in the regime specific means, corresponding to fast and slow growth rates:

	DPCAN	DPUSA	DPDEU	DPJPN
		Regime 1		
C	0.243118	0.424911	0.324078	0.273245
	(0.07609)	(0.08164)	(0.09849)	(0.12145)
	[3.19510]	[5.20487]	[3.29032]	[2.24993]
		Regime 2		
C	0.118511	-0.554926	-1.828623	0.147562
	(0.21083)	(0.21997)	(0.33735)	(0.33308)
	[0.56213]	[-2.52269]	[-5.42057]	[0.44302]

Next are the regime invariant (common) VAR lag coefficients and error covariance matrix:

	Common			
DPCAN(-1)	0.087721	0.066251	0.007997	0.027347
	(0.07045)	(0.07135)	(0.08431)	(0.11266)
	[1.24518]	[0.92851]	[0.09485]	[0.24273]
DPCAN(-2)	0.000909	0.067051	-0.018585	0.075223
	(0.06423)	(0.06520)	(0.07664)	(0.10290)
	[0.01416]	[1.02834]	[-0.24252]	[0.73100]
DPUSA(-1)	0.421243	0.140479	0.142676	0.148158
	(0.06728)	(0.06955)	(0.08199)	(0.10758)
	[6.26116]	[2.01969]	[1.74006]	[1.37721]
DPUSA(-2)	0.067068	0.145516	0.108525	-0.035717
	(0.07210)	(0.07360)	(0.08936)	(0.11535)
	[0.93020]	[1.97709]	[1.21442]	[-0.30964]
DPDEU(-1)	0.057956	-0.047926	-0.141830	0.052416
	(0.04803)	(0.04899)	(0.05824)	(0.07680)
	[1.20677]	[-0.97824]	[-2.43514]	[0.68251]
DPDEU(-2)	-0.027297	-0.064217	-0.146389	-0.057118
	(0.04610)	(0.04650)	(0.05436)	(0.07379)
	[-0.59218]	[-1.38094]	[-2.69311]	[-0.77411]
DPJPN(-1)	0.134863	0.146054	0.271693	0.240272
	(0.04117)	(0.04142)	(0.04887)	(0.06590)
	[3.27605]	[3.52601]	[5.55960]	[3.64614]
DPJPN(-2)	-0.051127	0.018629	0.183025	0.274969
	(0.04304)	(0.04450)	(0.05224)	(0.06882)
	[-1.18779]	[0.41861]	[3.50329]	[3.99531]
SIGMA-DPCAN	0.473418	0.170626	0.033663	0.120773
	(0.04428)	(0.03429)	(0.03956)	(0.05078)
	[10.6908]	[4.97574]	[0.85087]	[2.37853]
SIGMA-DPUSA	0.170626	0.473529	-0.079511	0.108050
	(0.03429)	(0.04930)	(0.04176)	(0.05180)
	[4.97574]	[9.60517]	[-1.90384]	[2.08606]
SIGMA-DPDEU	0.033663	-0.079511	0.627890	0.142532
	(0.03956)	(0.04176)	(0.06687)	(0.06318)
	[0.85087]	[-1.90384]	[9.38965]	[2.25604]
SIGMA-DPJPN	0.120773	0.108050	0.142532	1.214430
	(0.05078)	(0.05180)	(0.06318)	(0.11353)
	[2.37853]	[2.08606]	[2.25604]	[10.6973]

The remaining results show the parameters of the transition matrix and summary statistics for the estimated VAR.

Variable	Transition Matrix Parameters Coefficient	Std. Error	z-Statistic	Prob.
P11-C	2.962861	0.382110	7.753938	0.0000
P21-C	0.442247	0.552673	0.800197	0.4236

Determinant resid covariance	0.239115
Log likelihood	-1113.597
Akaike info criterion	10.17988
Schwarz criterion	10.95959
Number of coefficients	52

Instead of focusing on the transition matrix parameters, we examine the transition matrix probabilities directly by selecting **View/Regime Results/Transition Results...** and clicking on **OK** to display the default summary view:

VAR: MSI_VAR2
Date: 03/26/19 Time: 12:30
Transition summary: Constant Markov transition
probabilities and expected durations
Sample (adjusted): 1961Q4 2018Q4
Included observations: 229 after adjustments

Constant transition probabilities:
P(i, k) = P(s(t) = k | s(t-1) = i)
(row = i / column = k)

	1	2
1	0.950868	0.049132
2	0.608794	0.391206

Constant expected durations:

	1	2
	20.35325	1.642591

Here, we see the transition probability matrix and the expected durations. Note that there is considerable state dependence in the transition probabilities with a higher probability of remaining in the high growth regime (0.95), but less so for the low growth regime (0.39). The corresponding expected durations in a regime are approximately 10.4 and 4.1 quarters, respectively.

Lastly, we display the filtered and full sample estimates of the probabilities of being in the two regimes. First select **View/Regime Results/Regime Probabilities...** and choose the smoothed results. We will display the results only for the second regime.

After saving the view as a graph and applying the RecShade add-in,

http://www.eviews.com/Addins/addins.shtml

to label the NBER recessions, we see that the predicted probabilities of being in the low output state generally coincide with the commonly employed definition of US recessions:

We may examine the impulse responses for this VAR in the usual fashion. Click on **View/ Impulse Response...** edit the fields to show the impulses and responses for the first two endogenous variables, and click on **OK** to display the results:

Markov Switching Mean

We may estimate a two-state Markov switching mean VAR(2) with switching mean regressor C, switching VAR lag coefficients. and common error variance (MSMA-VAR(2)).

We leave the remaining settings at their defaults and click on **OK** to estimate the equation. Following estimation, EViews displays the results.

First, the top portion shows the specification settings:

```
Markov Switching Means VAR Estimates (BFGS / Marquardt steps)
Date: 03/26/19   Time: 12:00
Sample (adjusted): 1961Q4 2018Q4
Included observations: 229 after adjustments
Number of states: 2
Initial probabilities obtained from ergodic solution
Standard errors & covariance computed using observed Hessian
Random search: 25 starting values with 10 iterations using 1 standard deviation
      (rng=kn, seed=1551063419)
Convergence achieved after 102 iterations
Standard errors in ( ) & z-statistics in [ ]
```

The next section contains results for the regime varying results; first for regime 1,

	DPCAN	DPUSA	DPDEU	DPJPN
		Regime 1		
DPCAN(-1)	-0.391048	0.035171	-0.019714	-0.007598
	(0.10884)	(0.12365)	(0.13464)	(0.16501)
	[-3.59275]	[0.28444]	[-0.14642]	[-0.04605]
DPCAN(-2)	-0.019083	0.019645	-0.124572	0.152827
	(0.09927)	(0.11616)	(0.12844)	(0.15461)
	[-0.19222]	[0.16911]	[-0.96990]	[0.98849]
DPUSA(-1)	0.808374	0.122241	0.711618	0.481926
	(0.08271)	(0.10178)	(0.11866)	(0.14185)
	[9.77328]	[1.20107]	[5.99708]	[3.39740]
DPUSA(-2)	-0.020501	-0.074936	-0.026362	-0.306532
	(0.10604)	(0.13416)	(0.14327)	(0.18321)
	[-0.19334]	[-0.55854]	[-0.18401]	[-1.67308]
DPDEU(-1)	0.159700	0.074446	0.315942	0.301823
	(0.06745)	(0.08157)	(0.09903)	(0.11389)
	[2.36763]	[0.91270]	[3.19051]	[2.65015]
DPDEU(-2)	0.159358	0.061623	-0.226353	0.089392
	(0.06765)	(0.07872)	(0.09182)	(0.11614)
	[2.35559]	[0.78283]	[-2.46508]	[0.76967]
DPJPN(-1)	0.321490	0.238193	0.080134	0.400013
	(0.06359)	(0.07642)	(0.08752)	(0.10966)
	[5.05532]	[3.11677]	[0.91559]	[3.64782]
DPJPN(-2)	-0.234563	-0.118519	-0.062310	0.296559
	(0.06429)	(0.07962)	(0.09059)	(0.10607)
	[-3.64861]	[-1.48863]	[-0.68780]	[2.79581]
C	0.763286	0.753963	0.229268	0.859171
	(0.10282)	(0.09250)	(0.11035)	(0.21868)
	[7.42345]	[8.15054]	[2.07766]	[3.92896]

and then for regime 2,

	Regime 2			
DPCAN(-1)	0.445794	0.097977	0.039130	0.026633
	(0.08695)	(0.10397)	(0.11295)	(0.13519)
	[5.12704]	[0.94233]	[0.34644]	[0.19700]
DPCAN(-2)	0.072499	0.025905	0.128669	0.049386
	(0.07673)	(0.09508)	(0.10509)	(0.13182)
	[0.94492]	[0.27245]	[1.22433]	[0.37465]
DPUSA(-1)	0.057907	0.271717	-0.178131	-0.178506
	(0.08057)	(0.09976)	(0.11733)	(0.13815)
	[0.71870]	[2.72372]	[-1.51815]	[-1.29209]
DPUSA(-2)	0.147858	0.352028	0.135499	0.289226
	(0.09341)	(0.11292)	(0.12366)	(0.14993)
	[1.58295]	[3.11750]	[1.09574]	[1.92902]
DPDEU(-1)	0.032779	-0.089371	-0.328781	-0.266046
	(0.06672)	(0.07662)	(0.08815)	(0.11327)
	[0.49131]	[-1.16640]	[-3.72971]	[-2.34879]
DPDEU(-2)	-0.193798	-0.129043	-0.103311	-0.044428
	(0.06237)	(0.07464)	(0.08559)	(0.11218)
	[-3.10714]	[-1.72880]	[-1.20703]	[-0.39603]
DPJPN(-1)	0.020410	0.051312	0.328461	0.151985
	(0.05184)	(0.05958)	(0.06563)	(0.08078)
	[0.39373]	[0.86118]	[5.00468]	[1.88138]
DPJPN(-2)	0.060159	0.060018	0.239172	0.113742
	(0.05521)	(0.06572)	(0.07356)	(0.09092)
	[1.08959]	[0.91319]	[3.25131]	[1.25102]
C	0.734701	0.707277	0.788342	0.693111
	(0.10146)	(0.11757)	(0.10851)	(0.12144)
	[7.24131]	[6.01575]	[7.26534]	[5.70766]

We can examine directly the transition probabilities. The default transition probability summary (**View/Regime Results/Transition Results...**) shows descriptive statistics for the elements of the transition matrix:

VAR: MSMA_VAR2
Date: 03/26/19 Time: 13:03
Transition summary: Constant Markov transition
probabilities and expected durations
Sample (adjusted): 1961Q4 2018Q4
Included observations: 229 after adjustments

Constant transition probabilities:
P(i, k) = P(s(t) = k | s(t-1) = i)
(row = i / column = k)

	1	2
1	0.321569	0.678431
2	0.544430	0.455570

Constant expected durations:

	1	2
	1.473989	1.836782

The filtered probabilities of being in the regime 2, with NBER recession shading are:

Examining the impulse responses, we click on **View/Impulse Response...** edit the fields to show the impulses and responses for the first two endogenous variables, and click on **OK** to display the results:

Notice that the results are displayed in a spool, with impulse responses computed for each of the regime coefficients.

References

Frühwirth-Schnatter, Sylvia (2006). *Finite Mixture and Markov Switching Models,* New York: Springer Science + Business Media LLC.

Goldfeld, Stephen M. and Richard E. Quandt (1973). "A Markov Model for Switching Regressions," *Journal of Econometrics*, 3–16.

Goldfeld, Stephen M. and Richard E. Quandt (1976), *Studies in Nonlinear Estimation*, Cambridge, MA: Ballinger Publishing Company.

Hamilton, James D. (1989). "A New Approach to the Economic Analysis of Nonstationary Time Series and the Business Cycle," *Econometrica*, 57, 357–384.

Hamilton, James D. (1990). "Analysis of Time Series Subject to Changes in Regime," *Journal of Econometrics*, 45, 39–70.

Hamilton, James D. (1994). *Time Series Analysis*, Chapter 22, Princeton: Princeton University Press.

Krolzig, H.-M. (1997). *Markov-Switching Vector Autoregressions*. Berlin: Springer.

Chapter 48. 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), and time varying (random) coefficient models.

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), whose approach we largely follow.

Specification

A linear state space representation of the dynamics of the $n \times 1$ vector y_t is given by the system of equations:

$$y_t = c_t + Z_t \alpha_t + \epsilon_t \tag{48.1}$$

$$\alpha_{t+1} = d_t + T_t \alpha_t + v_t \tag{48.2}$$

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:

$$\Omega_t = \text{var}\begin{bmatrix}\epsilon_t \\ v_t\end{bmatrix} = \begin{bmatrix}H_t & G_t \\ G_t' & Q_t\end{bmatrix} \tag{48.3}$$

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.

Note that the updating equation for the states is for the states in period $t+1$, given the errors specified in period t. This particular timing convention, which follows Koopman, Shephard, and Doornik (1999), has important implications for the interpretation of correlations between errors in the signal and state equations G_t as discussed in "A Note on Correlated Errors" on page 978.

In the discussion that follows, we will generalize the specification given in (48.1)—(48.3) 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 973.

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:

$$a_{t|s} \equiv E_s(\alpha_t) \tag{48.4}$$

$$P_{t|s} \equiv E_s[(\alpha_t - a_{t|s})(\alpha_t - a_{t|s})'] \tag{48.5}$$

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:

$$\tilde{y}_t = y_{t|t-1} \equiv E_{t-1}(y_t) = E(y_t | a_{t|t-1}) = c_t + Z_t a_{t|t-1} \tag{48.6}$$

The *one-step ahead prediction error* is given by,

$$\tilde{\epsilon}_t = \epsilon_{t|t-1} \equiv y_t - \tilde{y}_{t|t-1} \tag{48.7}$$

and the *prediction error variance* is defined as:

$$\tilde{F}_t = F_{t|t-1} \equiv \mathrm{var}(\epsilon_{t|t-1}) = Z_t P_{t|t-1} Z_t' + H_t \tag{48.8}$$

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 Ξ_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,* $\hat{\alpha}_t \equiv a_{t|T} \equiv \mathrm{E}_T(\alpha_t)$, and *smoothed estimates of the state variances*, $V_t \equiv \mathrm{var}_T(\alpha_t)$. 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 s*moothed estimates of the signal variables*,

$$\hat{y}_t \equiv E(y_t | \hat{\alpha}_t) = c_t + Z_t \hat{\alpha}_t \tag{48.9}$$

and to compute the *variance of the smoothed signal estimates:*

$$S_t \equiv \mathrm{var}(\hat{y}_{t|T}) = Z_t V_t Z_t' . \tag{48.10}$$

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*:

$$\hat{\Omega}_t = \mathrm{var}_T\left(\begin{bmatrix} \epsilon_t \\ v_t \end{bmatrix}\right) \tag{48.11}$$

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 $\hat{\nu}_t$.

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 (48.4)—(48.8) yields the *n-step ahead state conditional mean and variance:*

$$a_{t+n|t} \equiv E_t(\alpha_{t+n}) , \tag{48.12}$$

$$P_{t+n|t} \equiv E_t[(\alpha_{t+n} - a_{t+n|t})(\alpha_{t+n} - a_{t+n|t})'] \tag{48.13}$$

the *n-step ahead forecast,*

$$y_{t+n|t} \equiv E_t(y_{t+n}) \; = \; c_t + Z_t a_{t+n|t} \tag{48.14}$$

and the corresponding *n-step ahead forecast MSE matrix:*

$$F_{t+n|t} \equiv \mathrm{MSE}(\tilde{y}_{t+n|t}) \; = \; Z_{t+n} P_{t+n|t} Z_{t+n}{}' + H_t \tag{48.15}$$

for $n = 1, 2, \ldots$. 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, $s = t + 1, \ldots, t + n^*$.

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 $n = 1, \ldots, n^*$ 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 989) 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:

$$\log L(\theta) = -\frac{nT}{2}\log 2\pi - \frac{1}{2}\sum_t \log\left|F_t(\theta)\right| - \frac{1}{2}\sum_t \tilde{\epsilon}_t'(\theta)F_t(\theta)^{-1}\tilde{\epsilon}_t(\theta) \tag{48.16}$$

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 C. "Estimation and Solution Options," on page 1319).

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 982.

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. Note that you can insert a state space specification from an existing text file by clicking on the Spec button to display the state space specification, then pressing the right-mouse button menu and selecting **Insert Text File...**

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 vari-

ables. An example of this technique is provided in the example "ARMAX(2, 3) with a Random Coefficient" on page 978.

- 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 976.

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 976.

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)
```

A Note on Correlated Errors

We caution that you should think very carefully about the timing of your errors if you allow for correlation between errors in the state and signal equations. The timing in the signal equation and state equation updates specified in Equation (48.2) and Equation (48.1), implies that correlation between the signal and state errors is defined to be between the errors in the signal at time t, and the errors in the states dated in $t+1$. This timing may not be what you intend. To allow for correlation in the contemporaneous states and signals in time t, you will need to specify errors in the lagged states, and define correlation between the lagged state errors and the errors in the signal equation.

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 800.

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 993).

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:

$$\begin{bmatrix} SV1 \\ SV2 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \qquad \text{var}\begin{bmatrix} SV1 \\ SV2 \end{bmatrix} = \begin{bmatrix} 1 & 0.5 \\ 0.5 & 2 \end{bmatrix} \tag{48.17}$$

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:

```
ss1.append @mprior svec0
ss1.append @vprior svar0
```

For more details on matrix objects and the `fill` and `append` commands, see Chapter 11. "Matrix Language," on page 261 of the *Command and Programming Reference*.

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 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.

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, optimization and coefficient covariance methods, the maximum number of iterations, convergence tolerance, the estimation algorithm, derivative settings and whether to display the starting values.

The default optimization method for new sspace objects is BFGS, but you may also select Newton-Raphson, OPG - BHHH, or EViews legacy. For non-legacy estimation, the **Step method** allows you to choose between the default **Marquardt**, **Dogleg**, and Line search determined steps.

The **Ordinary** default **Covariance method** employs the inverse of the matrix specified in the **Information matrix** dropdown menu. Alternately, you may compute the sandwich covariance by selecting **Huber – White** in the **Covariance method** menu.

The outer-product of the gradients (**OPG**) is the default information matrix estimator. If you are performing non-legacy optimization, you may use the observed Hessian by selecting **Hessian – Observed**.

The default settings should provide a good start for most problems; if you choose to change the settings, see "Setting Estimation Options" on page 1313 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 display the estimation output view:

Sspace: SS_ARMA21
Method: Maximum likelihood (BFGS / Marquardt steps)
Date: 03/16/15 Time: 11:42
Sample: 1949M01 1960M12
Included observations: 144
Convergence achieved after 9 iterations
Coefficient covariance computed using outer product of gradients

	Coefficient	Std. Error	z-Statistic	Prob.
C(1)	5.499767	0.257517	21.35687	0.0000
C(2)	0.409013	0.167201	2.446239	0.0144
C(3)	0.547165	0.164608	3.324055	0.0009
C(4)	1.188382	0.141461	8.400799	0.0000
C(5)	-4.934585	0.308276	-16.00704	0.0000

	Final State	Root MSE	z-Statistic	Prob.
SV1	0.245396	0.084850	2.892117	0.0038
SV2	0.319569	0.047896	6.672101	0.0000

Log likelihood	124.3367	Akaike info criterion	-1.657454
Parameters	5	Schwarz criterion	-1.554336
Diffuse priors	0	Hannan-Quinn criter.	-1.615553

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 993.

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 980.
- The **Estimation Output** view displays the coefficient estimates and summary statistics as described above in "Interpreting the estimation results" on page 985. 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 202.
- **Label** allows you to annotate your object. See "Labeling Objects" on page 116 of *User's Guide I*.

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:

- **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, $\epsilon_{t|t-1}$, or standardized one-step residuals, $e_{t|t-1}$, the smoothed signals, $\hat{y}_t$, smoothed signal disturbances, $\hat{\epsilon}_t$, or the standardized smoothed signal disturbances, $\hat{e}_t$. ± 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.

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, $\hat{\alpha}_t$, smoothed state disturbance estimates, $\hat{v}_t$, 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 982). 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 984).

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 972. 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.

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 wild-card 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: $(I - T \otimes T) \times \mathrm{vec}(P) = \mathrm{vec}(Q)$. 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 49. 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:

$$F(y, x) = 0 \tag{49.1}$$

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:

$$y_i = f_i(y, x) \tag{49.2}$$

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 o*verridden*, 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.

(A second, simpler example may be found in "Plotting Probability Response Curves" on page 376).

The macro model contains three stochastic equations and one identity. In EViews notation, these can be written:

```
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
```

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.

Estimating the Equations

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 89 and system "Two-Stage Least Squares" and related sections beginning on page 793).

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.

Creating the Model

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.

Linking the Equations

To add our estimated stochastic equations to the model, we can simply copy-and-paste or drag-and-drop them 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. To drag-and-drop, simply select the equation objects, then drag into the model window. Click on **OK** when prompted to link the equation to the object.

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**.

Adding the Identity

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.

Performing a Static Solution

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.

Examining the Solution Results

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 workfile 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 select **View/Variables** or 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 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 are 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 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.

(The names of the four series will be pre-filled in the **Model variables** section. Alternately, we could have simply selected **Proc/Make Graph...** then set the **Model variables** dropdown to **Endogenous variables**.)

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 "1960 1999", 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.

Performing a Dynamic Solution

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 will 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 "1985 1999".

Click on **OK** to solve the model. To examine the results, we will use **Proc/Make Graph…** exactly as above to display the actuals and the baseline solutions for the endogenous variables. 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.

Forecasting

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.

Filling in Exogenous Data

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 for G from 2000Q1 to 2005Q4, saving the results back into G itself; see Chapter 25. "Forecasting from an Equation," on page 167 for details. Later we will show you how to instruct the model to use the data in a different series, say G_1, in place of the data in G ("Using Scenarios for Alternate Assumptions" on page 1011), so that you may preserve the original state of the series G.

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.

Producing Endogenous Forecasts

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, select **Endogenous variables** in the **Model variables** section, then set the sample to 1995Q1 to 2005Q4 (so that we include five years of historical data). We will only display the baseline results so uncheck the **Actuals** box, then click on **OK** to produce the graphs.

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.

Using Add Factors to Model Equation Residuals

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 (if the series I_A does not already exist in the workfile). Click on **OK** 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.

Performing a Stochastic Simulation

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 coefficients 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 and choose **Dynamic** solution for the sample "2000 2005". In the **Solution scenarios & output** box on the right-hand side of the dialog, make sure that the **Std. Dev.** checkbox in the **Active** section is checked. Click on **OK** to begin the simulation.

Status messages will appear to indicate progress of the simulation. When the simulation is complete select **Proc/Make Graph…** to display the results. As before, we will set the **Model variables** to **Endogenous variables** and the sample to "1995 2005". In addition, you should choose **Mean + - 2 standard deviations** in the **Solution Series** box, check the **Actuals** and **Active** scenario boxes, and set the latter to **Baseline**. Click on **OK** to produce the graph.

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.

Using Scenarios for Alternate Assumptions

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 defined in the model. You can use this dialog to select which scenario is currently active, or to create, rename, copy, and delete scenarios.

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. The two scenarios 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 names appended by 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 with the default name "Scenario 1". Once we have created the scenario, we can modify the scenario from the baseline settings by overriding one of our exogenous variables. To add an override for the series M, first make certain that "Scenario 1" is active (highlighted in the **Scenario Specification** dialog) then click on the Overrides tab and enter "M" in the dialog. Click on **OK** to accept your changes.

Alternately, after exiting the **Scenario Specification** dialog, you may **View/Variables** to 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 using "Scenario 1".

(You may use the **Aliasing** tab to change the extension from "_1". Note also that depending on how you created the override, you may still need to create the series M_1 in your workfile by copying the values of M.)

For example, 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, and "Baseline" is the alternate scenario, and that **Solve for Alternate along with Active** is checked. Set the solution sample to "2000 2005". Click on **OK** to solve.

Once the solution is complete, we can use **Proc/Make Graph…** to display the results following the same procedure as above. First, set the **Model variables** selection to display the **Endogenous variables**. Next, set the **Solution series** list box to the setting **Deterministic solutions**, then check both the **Active** and **Compare** solution check boxes, making sure that the active scenario is set to "Scenario 1", and the comparison scenario is set to "Baseline". Set the sample to 1995Q1 to 2005Q4, then click on **OK**. 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, system, var, or equations you would like to add to the model, then copy-and-paste or drag-and-drop 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 1001.

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.

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 1038 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.

In addition, the equation view offers search capability, allowing you to quickly locate and work with an equation of interest. You may access equation find access it either by clicking the **Find** button (on the far right of the button bar), or pressing **CTRL-F** to bring up the **Model Find** dialog:

You may search for equations by name, by exogenous variable or by endogenous variable. For example if you have a linked equation called EQ01, you can type "Eq01" in the **Find** edit field, click on **Find Next**, and EViews will select that equation in the equation view. Similarly you can type in **GDP**, and EViews will select the next equation (either linked or written as an inline expression in the model) that contains the variable GDP.

The check boxes on the left allow you to modify which type of objects/variables to search. If we unselect **Exogenous**, and then search for **GDP**, EViews will find only equations which contain **GDP** as an endogenous variable.

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 "Examining the Solution Results" (p. 1002)). 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 1028 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 1042.

Users of early 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.

Print View

Print view (available from the **PrintView** button, or from **View/Print View**) offers a new way of looking at the underlying equations in your model. While similar to text view, print view offers display enhancements, but does not allow editing of the model. Selecting print view will bring up the **Model Print View** dialog:

The dialog settings let you determine how you would like to view the equations.

- Checking the **Break links** check box tells EViews to display linked equations as expressions rather than as linked objects. Note that this is only for display—the linked equations will remain linked in your model, but will be displayed as though they were broken.
- The **Include identities**, **Include @innov statements**, **Include add-factors**, and **Include comments** check boxes determine whether the respective model elements will be displayed in the view.
- Checking the **Use Display names** check box will replace all series names in the model with their display name.
- The **Number format** dropdown allows you to specify the display format for coefficients in the equations.

For example, the traditional text view of the model presented in "Text View," on page 1019 may be compared to the print view which substitutes embedded equation and model definitions and displays fewer significant digits:

Dependency Graph View

This view provides a graphical representation of the relationships among the model's variables. The model is shown as a directed graph with nodes for each variable and edges between pairs of related variables. Edges follow the direction of information flow. For example, in the below graph variable NOW depends on variable HILO (the equation for NOW includes HILO), resulting in a directed edge from HILO to NOW. Variables may be mutually dependent, in which case the edge between them will have arrowheads at both ends, e.g., variables BTE and XBTE.

Computing the layout for the graph is computationally expensive and may take some time. If you select this view but wish to abort before the layout calculations are completed, hit the F1 key.

The color of each edge summarizes the temporal (lag/lead) attributes of the dependency along a warm-cool spectrum:

Orange: Only lagged values are present.

Red: Lagged and contemporaneous values are present.

Black: Only contemporaneous values are present.

Blue: Lead and contemporaneous values are present.

Cyan: Only lead values are present.

If both lagged and lead values are present, the edge is shown in a either light or dark purple. If a lag or lead of length four or more is present, the edge is shown as a dotted line, e.g., the edge between variables BTE and XBTE.

The are several ways to adjust your view of the dependency graph and inspect additional information about the variables:

- Moving the mouse cursor over a node will highlight that node, all incident edges, and all adjacent nodes.
- Toggle hiding of a legend for edge colors.
- You may enlarge and shrink the graph using the mouse wheel (zoom in / zoom out) within a range of 1000% and 10% of the graph's natural size.

- Left-click and hold allows you draw a zoom selection box. When you release the left mouse button the selected area will be enlarged and centered within the window.
- When the graph is larger than the size of the window, scroll bars will appear. In addition to clicking on the scroll bars, you can right-click and hold within the window to pan the graph.
- You may press the ESC key to resize the graph to fit within the window (subject to the aforementioned zoom limits).
- Single left-clicking on a node will show a tool tip containing more detailed information about the associated variable's relationships.
- Double left-clicking on a node for an endogenous variable will open the properties dialog for that variable.
- Single right-clicking on the graph brings up a context menu. Through this menu you can:

  Toggle hiding of exogenous variables in the graph.

  Have EViews compute a different layout for the graph.

  Export an image of the graph at the current zoom level.

  Export a Graphviz DOT file for use with the GraphViz suite of open-source graph tools (not included or affiliated with EViews).

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 1011.

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** tab 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** tab 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** tab 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 exog-

enous 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.

The **Description** tab may be used to store notes describing the purpose of each scenario and, if desired, to include the description in the solution series. If the **Export description to solution series** check box is checked, any series that are generated during a solve under the scenario's description added to their own description attribute.

Scenario Editing Tools

The **Edit override** right button menu item, available from the variables view of a model, lets you quickly exclude, override and edit endogenous and exogenous variables for the current scenario.

To simultaneously create an override for an exogenous variable and open it for editing, first select **View/Variables** to display the model in variables view, then select **Edit override** from the right mouse button menu. EViews will ask you to confirm that you wish to override the variable, and will then open an untitled group containing the original series and the newly created override series for the current scenario. Again the group is opened in compare mode ("Group Comparison Tools," on page 595) to facilitate easy editing of the new series.

Edit override may also be used for endogenous variables, in which case EViews will simultaneously exclude and override the variable for the current scenario.

You may quickly revert an overriden variable back to its original non-overriden values by using the **Revert** right button menu item.

Scenario List

The scenario list view of a model (available from **View/Scenario View**) displays a list of each of the model's scenarios, along with their alias, overrides or excludes, and descriptions:

Scenario	Alias	Excludes	Overrides	Description
Baseline	_0			
Scenario 1	_1			Default oil price scenario
Scenario 2	_2		X1	High oil price scenario
Scenario 3	_3	Y1		US GDP excluded

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:

$$f(y_i) = f_i(y, x) \tag{49.3}$$

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:

$$f(y_i) = f_i(y, x) + a \tag{49.4}$$

Alternatively, we may provide an add factor for the endogenous variable of the model by using the add factor as an offset:

$$f(y_i - a) = f_i(y, x) \tag{49.5}$$

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 Active 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.

Locking the Model

Model locking allows you to disable changes to your model. To toggle model locking, simply select **Protect/Unprotect model** from the **Proc** menu. When you first lock the model, you will be prompted to enter a password (of at least 4 characters). To unlock a previously locked model, simply enter the previously entered password.

Once password protected, a number of features of the model are locked down. Notably, the text view of the model cannot be accessed, all *existing* scenarios become write-protected, equations and variables cannot be added/deleted or modified, links cannot be broken, and add-factors cannot be assigned or modified.

Note that the model may still be solved, new scenarios can still be created, modified, and deleted, and you may create an unprotected copy of the model.

It is important to note that once a model is password protected it may not be opened in versions of EViews prior to 9.5.

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:

$$F(y(-\text{maxlag}), \ldots, y(-1), y, y(1), \ldots, y(\text{maxlead}), x) = 0 \tag{49.6}$$

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:

$$\begin{aligned} F(y_{s-1}, y_s, y_{s+1}, x) &= 0 \\ F(y_s, y_{s+1}, y_{s+2}, x) &= 0 \\ F(y_{s+1}, y_{s+2}, y_{s+3}, x) &= 0 \\ &\dots \\ F(y_{t-2}, y_{t-1}, y_t, x) &= 0 \\ F(y_{t-1}, y_t, y_{t+1}, x) &= 0 \end{aligned} \tag{49.7}$$

where the unknowns are y_s, y_{s+1},... 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):

$$\hat{y}_S = \hat{\phi}_1 \epsilon_{S-1} + \ldots + \hat{\phi}_q \epsilon_{S-q}, \tag{49.8}$$

you require values for the pre-forecast sample innovations, $\epsilon_{S-1}, \epsilon_{S-2}, \ldots, \epsilon_{S-q}$. 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 183 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 post-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 1038 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 distribu-

tions 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.

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.

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 1016 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.

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

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.

Boundaries

EViews 12 introduced the ability to specify boundaries for endogenous variables in a model through a **Boundaries** dialog page. Although the solver will not enforce the boundaries while solving the model, EViews will warn you if any variable crosses its boundaries (i.e., solves to a value higher than the upper boundary or less than the lower boundary) for any observation in the solve sample. Boundary violations can also be examined in more detail using the **Check Solution Bounds** view after the solve is complete.

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 1322.)

- **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 1323.)
- **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 1324.)

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 1031.

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:

$$F(y, x) = 0 \tag{49.9}$$

the reordering is chosen to partition the system into two parts:

$$\begin{aligned} F_1(y_1, y_2, x) &= 0 \\ F_2(y_1, y_2, x) &= 0 \end{aligned} \tag{49.10}$$

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 contro**l 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 $t = T, T+1, \ldots, T+k-1$), 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:

$$y_t - y_{t-1} = y_{t-1} - y_{t-2} \tag{49.11}$$

for $t = T, T+1, \ldots, T+k-1$). 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:

$$(y_t - y_{t-1})/y_{t-1} = (y_{t-1} - y_{t-2})/y_{t-2} \tag{49.12}$$

for $t = T, T+1, \ldots, T+k-1$). 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. EViews can address this challenge in two ways. The first is to perform a numerical

optimization of the exogenous variable values, automating the search for exogenous values that lead to the desired endogenous values. The second approach respecifies the model in such a way that we can directly solve for the exogenous values.

After creating series holding the trajectory values for our endogenous target variables, we can conduct a search for the precipitating values of our exogenous control variables via the **Solve Controls for Targets…** item under the model's **Proc** menu.

The required parameters are a list of control variables (in the upper text area) and a list of target variables and trajectory series (in the lower text area). These lists may include group objects. The sample over which new control values will be calculated defaults to the current workfile sample. The solved control values will be stored in the original control series by default. The **Store solution values…** checkbox lets you indicate that the results should be stored in an alternative set of series with the adjoining suffix. This suffix defaults to the active scenario alias suffix, which allows the results to be easily applied to the model via a scenario override. As a convenience, if the trajectory series names are just the target variable names with a suffix applied, you may use the **Omit trajectory series…** checkbox and adjoining field to shorten the required parameters.

After clicking the **OK** button, EViews will determine an appropriate lag for each control variable, conduct the search over control values, and then display a summary of the results in the open model window. The image below shows an example results summary for a single control named DGF.

This summary is a transient rather than a persistent model view and will be lost on the next view change if not frozen.

There are three notable limitations of this numeric approach.

- The model may not contain leads (positive lags).
- There must be an equal number of control and target variables. The overdetermined case with more target variables than control variables is not supported currently.
- The procedure may take some time to complete. Depending on the dependencies among the variables and the control lags selected, the optimization may require multiple bidirectional passes over the sample observations, causing the solution time to be far greater than a standard model solve.

It is also possible for the search to fail, with EViews unable to discovery a set of satisfactory control variable values. You can always alter the control series values (the search's starting values) and try again.

Endogenous Variable Specification

The alternative to a numerical search for exogenous values is to respecify the model such that the exogenous control variables are made endogenous and the endogenous target series are made exogenous. The values of the control series can then be computed using the standard model solve procedure (with the trajectory series overriding the target series). You can

automatically rewrite a model's equations to select a new set of endogenous variables via the **Flip Endog/Exog Variables…** item under the model's **Proc** menu.

The required parameter is a list of endogenous and exogenous variables to flip (in the upper text area). There must be an equal number of endogenous and exogenous variables (but not add factors) that may be specified in any order. The list may include group objects.

EViews will automatically determine a minimal set of equations to rewrite symbolically. The number of equations rewritten will be at least the number of specified exogenous variables, but could be much greater. If a rewritten equation was included in the model via a linked object, e.g., a VAR, system, or another model, then the link to that object will be broken.

If there are equations that you wish to exclude from being rewritten, you may list the endogenous variable for these "locked" equations in the lower text area. Additionally, you may lock all equations that have associated add factors or innovation variances with the appropriate check box. Naturally, endogenous variables being flipped should not be locked.

Once you click the **OK** button, EViews will determine which equations need to be rewritten and then modify the model object accordingly. A summary of the changes made to the model is displayed in the model window.

This approach has several limitations to be aware of:

- Equations containing ARMA specifications will not be altered by this procedure and are treated as locked.
- Currently, all rewritten equations lose their add factors. This deficiency will hopefully be addressed in a future update, but for the time being locking all equations with add factors is the only way to prevent such information loss.
- Rewriting equations is a process of algebraic manipulation, since we are essentially solving equations for a new set of left-hand side variables. Consequently, we are limited by the set of invertible mathematical operations.

It is possible that EViews will be unable to rewrite the model's equations to produce the desired set of endogenous variables. If you have explicitly locked any equations, you may retry with fewer locked equations, hopefully providing EViews with enough options to be successful.

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 A. "Wildcards," on page 817 of the *Command and Programming Reference*).

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.

Comparing Solution Data

The **Compare solutions** menu item, available from the **View** menu, allows you to quickly view any differences between the solution values for different scenarios.

Once you have solved your model for different scenarios, you may wish to quickly compare the results between those scenarios to see which variables differ. Clicking on the menu item **View/Compare solutions**... brings up a dialog that allows you to do this. The first part of the dialog is similar to that of the data display dialog above. Select which variables you would like to compare by using the **Select** drop-down box, and the **From** edit field.

Having selected your variable, you may select which scenarios to compare using the **Series to compare** area. The first choice, using the **Solution series** drop-down is whether you wish to compare the deterministic solutions, or compare the means from a stochastic solve. Note you must have already performed the type of solve you choose prior to comparing it.

The **Compare the Active Scenario** drop-down lets you choose the set of variables for comparison. By default the drop-down will be set at the model's current active scenario. Note

changing this drop-down to another entry will change the active scenario for the model, as well as for comparison.

There are two choices for specifying the second set of variables. You may either select a comparison scenario (by selecting the **Scenario** radio button, and then selecting the scenario in the drop-down), or you may specify a pattern matching scheme by selecting the **Pattern** radio button. With pattern matching, you should use the "*" wildcard to represent the variable names in the pattern. For example, if you wish to compare I_0 (the current active scenario) with a series called I_OLD, you would enter a pattern of "*_OLD", having specified I as the variable to compare. Note that you may reference series stored in a database using the standard `dbname::` syntax, or series in another page using the standard `pagename\` syntax.

The series used for comparison should already exist in the workfile (or storage location if you specified another container with pattern matching). Note this means that you should have already solved the model for the specified scenario if applicable.

The **Include threshold** edit box lets you set the tolerance level for detecting differences between the solutions. By default it is set to 0.1% - i.e. any relative difference less than 0.001 will be ignored. You can specify a value of 0 to tell EViews to show all differences, no matter how small.

Finally the **Comparison sample** edit field lets you set the sample over which you wish to compare the series.

Clicking **OK** produces the comparison table:

The table lists any variables for which the percentage difference between the two series for each scenario is greater than the specified tolerance.

In this case we are comparing the solution between the Baseline scenario (_0) and Scenario 1 (_1), and two variables, M and G, have a difference greater than the specified tolerance of 1e-04. The four columns in the table show details about the variables. The first

shows the variable name. The second, Delta%, shows the maximum difference between the two series for each variable. The third and fourth columns, "First" and "Last", give the date (or observation number) of the first period in which the two series differ and the last period in which the two series differ. Here, the first period in which M_0 differs from M_1 is 1960Q1, and the last period in which they differ is 1999Q4.

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, Hanoch 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 IX. 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 50. "Pooled Time Series, Cross-Section Data," on page 1061 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.

Data where the panel cross-sections are stacked are described separately:

- In Chapter 9. "Advanced Workfiles," beginning on page 271 of *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 51. "Working with Panel Data," beginning on page 1111, outlines the basics of working with panel workfiles.
- Chapter 52. "Panel Estimation," beginning on page 1135 describes least squares, instrumental variables, and GMM estimation in panel structured workfiles.
- Chapter 53. "Panel Cointegration Estimation," beginning on page 1191 documents tools for panel cointegrating regression estimation in EViews.
- Chapter 54. "Panel Statistics," beginning on page 1213 outlines the computation of other "panel workfile aware" statistics and tests.

Chapter 50. 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 51. "Working with Panel Data," on page 1111 and Chapter 52. "Panel Estimation," beginning on page 1135.

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 specific 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 1067).

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 1074, and the role of the pool object in estimation is the focus of "Pooled Estimation," beginning on page 1082.

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 different 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 1066.

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, and either select **Object/Copy Selected…** in the main workfile toolbar, or right mouse-click and select **Object/Copy…** 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", "_JPNGDPIN", "_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", "_JPNGDP", 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 A. "Wildcards," on page 817 in the *Command and Programming Reference*). 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 43 of *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** dropdown menu, 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 1062.

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 multi-

ple columns of unstacked source data. You may read unstacked data directly into EViews using the standard workfile creation procedures described in "Creating a Workfile by Reading from a Foreign Data Source" on page 47 of *User's Guide I*. Each cross-section specific variable should be read as an individual series, with 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 should use the standard EViews 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.

There are to primary approaches to importing data into your pool series: you may read the data in stacked form then use EViews tools to restructure the data in pool form, or you may directly read or copy the data into a stacked representation of the pooled series.

Indirect Setup (Restructuring) of Stacked Data

The easiest approach to reading stacked pool data is to create an EViews workfile containing the data in stacked form, and then use the built-in workfile reshaping tools to create a pool workfile with the desired structure and data. (Alternately, you can perform the first step and simply work with the data in stacked form: see Chapter 51. "Working with Panel Data," on page 1111 for details.)

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 47 of *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 306 of *User's Guide I* for a general discussion of reshaping, and "Unstacking a Workfile" on page 309 of *User's Guide I* for a more specific discussion of the unstack procedure.

The indirect method is generally 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.

Direct Import of Stacked Data

An alternative approach is to enter or read the data directly into the workfile using a pool object. You may enter or copy-and-paste data from the source into and a stacked representation of your data, or you may use the pool object to describe how to read the stacked data into the unstacked workfile.

To enter data or copy-and-paste, you use the pool object to create a 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.

	C?	G?	YEAR
	C?	G?	YEAR
_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
_USA-1963	NA	NA	NA
_USA-1964	NA	NA	NA
_USA-1965	NA	NA	NA
_USA-1966			

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.

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, located on the pool proc menu, 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.

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.

For a discussion of the text specific settings in the dialog, see "References" on page 186 of *User's Guide I*.

Alternately, you may read in the data into a panel workfile, and then use

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 1064), then to use the @INGRP function:

```
dum? = @ingrp(us)
```

to generate and implicitly refer to the four series (see "Pool Series" on page 1066).

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 1069, 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** + /– 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 save the results in series objects. See "Making Period Stats" on page 1079.

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 independent cross-section tests from 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), or you may compute dependent cross-sections tests from Bai and Ng's (2004) Panel Analysis

of Nonstationarity in Idiosyncratic and Common Components (PANIC), and Pesaran's (2007) Cross-sectionally Augmented IPS (CIPS).

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 considerable detail in "Cross-sectionally Independent Panel Unit Root Testing" on page 705.

Performing Cointegration Tests

Panel cointegration tests are available as a view of a group in a panel workfile or for a group of pooled series defined using a pool object. EViews allows you to conduct several different 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 "References," beginning on page 1263.

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/Covariance Analysis…** to compute the covariance matrix of the series in the group. EViews will perform the analysis using all of the individual series in the group.

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 315 of *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 1097.

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** dropdown menu, 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 1088).

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 "Specifying AR Terms" on page 132 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 1079).

Fixed and Random Effects

You should account for individual and period effects using the **Fixed and Random Effects** dropdown menus. By default, EViews assumes that there are no effects so that the dropdown menus 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.

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 dropdown menu 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.

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 dropdown menu at the top of the page 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. Select **White cross-section (period cluster)** for one-way period clustering, **White period (cross-section clustering)** for one-way cross-section clustering, **White (diagonal)** for unstructured heteroskedasticity robust covariances, or **White two-way cluster** for clustering by both cross-section and period. Each of the methods is described in greater detail in "Robust Coefficient Covariances" on page 1107.

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** dropdown menu 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 1101 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 (2005), 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.

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** dropdown menu 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** dropdown menu 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, 2005). The workfile ("Grun-

feld_Baltagi_pool.WF1”) contains annual observations on investment (I?), firm value (F?), and capital stock (K?) 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 obviously cannot demonstrate all of the specifications that may be estimated using these data, but we provide a few illustrative examples.

Fixed Effects

First, we estimate a model regressing I? on the common regressors F? and K?, 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 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.

Robust Standard Errors

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 non-zero covariances are allowed across cross-sections (clustering by period). 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

(clustering by cross-section). Select the **Options** page and choose **White period** as the coefficient covariance method. The coefficient results are given by.

White period standard errors & covariance (d.f. corrected)
WARNING: estimated coefficient covariance matrix is of reduced rank

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 caution that the White period results assume that the number of cross-sections is large, which is not the case in this example. In fact, the resulting coefficient covariance matrix is of reduced rank, a fact that EViews notes in the output.

AR Estimation

We may add an AR(1) term to the specification, and compute estimates using Cross-section SUR PCSE methods to compute standard errors that are robust to more contemporaneous 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: 08/17/09 Time: 14:45
Sample (adjusted): 1936 1954
Included observations: 19 after adjustments
Cross-sections included: 10
Total pool (balanced) observations: 190
Cross-section SUR (PCSE) standard errors & covariance (d.f. corrected)
Convergence achieved after 14 iterations

Variable	Coefficient	Std. Error	t-Statistic	Prob.
C	-63.45169	28.79868	-2.203285	0.0289
F?	0.094744	0.015577	6.082374	0.0000
K?	0.350205	0.050155	6.982469	0.0000
AR(1)	0.686108	0.105119	6.526979	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.

Random Effects

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 dropdown menus 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 estimate the coefficient covariance using the **Ordinary** method and 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:

$$0.7315 \;=\; 89.26257^2/(89.26257^2 + 15.77783^2 + 51.72452^2) \tag{50.1}$$

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.

Cross-section Specific Regressors

Suppose instead that we elect to estimate a specification with I? as the dependent variable, C and F? as the common regressors, and K? as the cross-section specific regressor, using cross-section weighted least squares. The top portion of the output is given by:

Dependent Variable: I?
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?	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.

Group Dummy Variables

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 1066). 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 1162

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 **Covariance Matrix** or **Correlation 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 49. "Models," on page 995.

Estimation Background

The basic class of models that can be estimated using a pool object may be written as:

$$Y_{it} = \alpha + X_{it}'\beta_{it} + \delta_i + \gamma_t + \epsilon_{it}, \tag{50.2}$$

where Y_{it} is the dependent variable, and X_{it} is a k-vector of regressors, and ϵ_{it} are the error terms for $i = 1, 2, \ldots, M$ cross-sectional units observed for dated periods $t = 1, 2, \ldots, T$. 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:

$$Y_i = \alpha l_T + X_i'\beta_{it} + \delta_i l_T + I_T\gamma + \epsilon_i \tag{50.3}$$

for $i = 1, \ldots, M$, 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, $\gamma' = (\gamma_1, \gamma_2, \ldots, \gamma_T)$.

Analogously, we may write the specification as a set of T period specific equations, each with M observations stacked on top of one another.

$$Y_t = \alpha l_M + X_t'\beta_{it} + I_M\delta + \gamma_t l_M + \epsilon_t \tag{50.4}$$

for $t = 1, \ldots, T$, 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, $\delta' = (\delta_1, \delta_2, \ldots, \delta_M)$.

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:

$$Y = \alpha l_{MT} + X\beta + (I_M \otimes l_T)\delta + (l_M \otimes I_T)\gamma + \epsilon \tag{50.5}$$

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:

$$\Omega = E(\epsilon\epsilon') = E\begin{pmatrix} \epsilon_1\epsilon_1' & \epsilon_2\epsilon_1' & \ldots & \epsilon_M\epsilon_1' \\ \epsilon_2\epsilon_1' & \epsilon_2\epsilon_2' & & \mathrm{M} \\ & & \mathrm{O} & \\ \epsilon_M\epsilon_1' & \ldots & & \epsilon_M\epsilon_M' \end{pmatrix} \tag{50.6}$$

If instead we treat the specification as a set of period specific equations, the stacked (by period) representation is given by,

$$Y = \alpha l_{MT} + X\beta + (l_M \otimes I_T)\delta + (I_M \otimes l_T)\gamma + \epsilon \tag{50.7}$$

with error covariance,

$$\Omega = E(\epsilon\epsilon') = E\begin{pmatrix} \epsilon_1\epsilon_1' & \epsilon_2\epsilon_1' & \ldots & \epsilon_T\epsilon_1' \\ \epsilon_2\epsilon_1' & \epsilon_2\epsilon_2' & & \mathrm{M} \\ & & \mathrm{O} & \\ \epsilon_T\epsilon_1' & \ldots & & \epsilon_T\epsilon_T' \end{pmatrix} \tag{50.8}$$

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 (50.2) allows for β slope coefficients that are common to all individuals and periods, as well as coefficients that are either cross-sec-

tion 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 (50.2) to:

$$Y_{it} = \alpha + X_{it}'\beta + \delta_i + \gamma_t + \epsilon_{it} \tag{50.9}$$

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:

$$Y_{it} = \alpha + X_{it}'\beta_i + \delta_i + \gamma_t + \epsilon_{it} \tag{50.10}$$

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:

$$Y_{it} = \alpha + X_{it}'\beta_t + \delta_i + \gamma_t + \epsilon_{it} \tag{50.11}$$

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:

$$Y_{it} = \alpha + X_{0it}'\beta_0 + X_{1it}'\beta_{1i} + X_{2it}'\beta_{2t} + \delta_i + \gamma_t + \epsilon_{it} \tag{50.12}$$

If there are k_1 common regressors, k_2 cross-section specific regressors, and k_3 period specific regressors, there are a total of $k_0 = k_1 + k_2 M + k_3 T$ 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 (50.2) on page 1097 that does not admit period specific regressors or effects,

$$Y_{it} = \alpha + X_{it}'\beta_i + \delta_i + \gamma_t + \epsilon_{it} \tag{50.13}$$

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:

$$\epsilon_{it} = \sum_{r=1}^{p} \rho_{ri}\epsilon_{it-r} + \eta_{it} \tag{50.14}$$

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:

$$Y_{it} = \rho_{1i} Y_{it-1} + \alpha(1-\rho_{1i}) + (X_{it} - \rho_{1i} X_{it-1})'\beta_i + \delta_i(1-\rho_{1i}) + \eta_{it} \tag{50.15}$$

using iterative techniques to estimate $(\alpha, \beta_i, \rho_i)$ for all i. See "Specifying AR Terms" on page 132 for additional discussion.

We emphasize that EViews does place are 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 using the demeaned data (see, for example Baltagi, 2005). 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 (2005), 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 (2005) for discussion, especially "Note 3" on p. 28).

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 (2005, p. 113-116) and "Random Effects and GLS" on page 1105 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:

$$\begin{aligned} E(\epsilon_{it}\epsilon_{it} | X_i^*) &= \sigma_i^2 \\ E(\epsilon_{is}\epsilon_{jt} | X_i^*) &= 0 \end{aligned} \tag{50.16}$$

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:

$$E(\epsilon_i \epsilon_i' | X_i^*) = \sigma_i^2 I_T \tag{50.17}$$

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:

$$\begin{aligned} E(\epsilon_{it}\epsilon_{jt} | X_t^*) &= \sigma_t^2 \\ E(\epsilon_{is}\epsilon_{jt} | X_t^*) &= 0 \end{aligned} \tag{50.18}$$

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:

$$E(\epsilon_t \epsilon_t' | X_t^*) = \sigma_t^2 I_M \tag{50.19}$$

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. Specifically, we assume that:

$$\begin{aligned} E(\epsilon_{it}\epsilon_{jt} | X_t^*) &= \sigma_{ij} \\ E(\epsilon_{is}\epsilon_{jt} | X_t^*) &= 0 \end{aligned} \tag{50.20}$$

for all i, j, s and t with $s \neq t$. The errors may be thought of as cross-sectionally correlated. Alternately, this error structure is sometimes referred to as clustered by period since observations for a given period are correlated (form a cluster). Note that in this specification the contemporaneous covariances do not vary over t.

Using the period specific residual vectors, we may rewrite this assumption as,

$$E(\epsilon_t \epsilon_t' | X_t^*) = \Omega_M \tag{50.21}$$

for all t, where,

$$\Omega_M = \begin{pmatrix} \sigma_{11} & \sigma_{12} & \cdots & \sigma_{1M} \\ \sigma_{12} & \sigma_{22} & & \cdots \\ & & \cdots & \\ \sigma_{M1} & \cdots & & \sigma_{MM} \end{pmatrix} \tag{50.22}$$

We term this 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 generalized 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 1107.

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.

Serial Correlation (Period SUR)

This class of covariance structures allows for arbitrary heteroskedasticity and serial correlation between the residuals for a given cross-section, but restricts residuals in different cross-sections to be uncorrelated. This error structure is sometimes referred to as clustered by cross-section since observations in a given cross-section are correlated (form a cluster).

Accordingly, we assume that:

$$\begin{aligned} E(\epsilon_{is}\epsilon_{it} | X_i^*) &= \sigma_{st} \\ E(\epsilon_{is}\epsilon_{jt} | X_i^*) &= 0 \end{aligned} \tag{50.23}$$

for all i, j, s and t with $i \neq j$. Note that in this specification 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,

$$E(\epsilon_i \epsilon_i' | X_i^*) = \Omega_T \tag{50.24}$$

for all i, where,

$$\Omega_T = \begin{pmatrix} \sigma_{11} & \sigma_{12} & \dots & \sigma_{1T} \\ \sigma_{12} & \sigma_{22} & & \dots \\ & & \dots & \\ \sigma_{T1} & \dots & & \sigma_{TT} \end{pmatrix} \tag{50.25}$$

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 1103 for related discussion of errors clustered-by-period.

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:

$$\hat{\beta}_{OLS} = \left(\sum_i X_i' X_i\right)^{-1}\left(\sum_i X_i' Y_i\right) \tag{50.26}$$

while the corresponding IV estimator is given by:

$$\hat{\beta}_{IV} = \left(\sum_i X_i' P_{Z_i} X_i\right)^{-1}\left(\sum_i X_i' P_{Z_i} Y_i\right) \tag{50.27}$$

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 $Z_{0_{it}}$, 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:

$$\begin{aligned} \hat{\beta}_{OLS} &= \left(\sum_i X_i' Q X_i\right)^{-1}\left(\sum_i X_i' Q Y_i\right) \\ \hat{\beta}_{IV} &= \left(\sum_i X_i Q P_{Z_i} Q X_i\right)^{-1}\left(\sum_i X_i' Q P_{Z_i} Q Y_i\right) \end{aligned} \tag{50.28}$$

where $Z_i = Q Z_i$.

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:

$$\begin{aligned}\hat{\beta}_{GLS} &= \left(\sum_i X_i'\Omega_M^{-1}X_i\right)^{-1}\left(\sum_i X_i'\Omega_M^{-1}Y_i\right)\\ \hat{\beta}_{GIV} &= \left(\sum_i X_i\Omega_M^{-1/2}P_{Z_i*}\Omega_M^{-1/2}X_i\right)^{-1}\left(\sum_i X_i'\Omega_M^{-1/2}P_{Z_i*}\Omega_M^{-1/2}Y_i\right)\end{aligned} \tag{50.29}$$

where $Z_i* = \Omega_M^{-1/2}Z_i$.

In the context of random effects specifications, this approach to IV estimation is termed generalized two-stage least squares (G2SLS) method (see Baltagi (2005, p. 113-116) 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 1185 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,

$$Y_{it} = \alpha + X_{it}'\beta_i + \delta_i + \epsilon_{it} \tag{50.30}$$

and the transformed equation,

$$Y_{it} = \rho_{1i}Y_{it-1} + \alpha(1-\rho_{1i}) + (X_{it} - \rho_{1i}X_{it-1})'\beta_i + \delta_i(1-\rho_{1i}) + \eta_{it} \tag{50.31}$$

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 automati-

cally, 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 (period cluster)* method assumes that the errors are contemporaneously (cross-sectionally) correlated. The method treats the pool regression as a multivariate regression (with an equation for each cross-section), and computes robust standard errors for the system of equations. This estimator is robust to cross-equation (contemporaneous) correlation and heteroskedasticity. See Wooldridge (2002, p. 148-153) and Arellano (1987).

Alternatively, the *White period (cross-section cluster)* method assumes that the errors for a cross-section are heteroskedastic and serially correlated. The estimator is designed to accommodate arbitrary heteroskedasticity and within cross-section serial correlation.

The *White two-way cluster* method allows for both cross-section and period clustering.

In contrast, the *White (diagonal)* method is robust to observation specific heteroskedasticity in the disturbances, but not to correlation between residuals for different observations.

Each of the coefficient covariance methods is described in greater detail in "Cluster-Robust Covariances" on page 39.

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 variance matrices Ω_M and Ω_T, but place additional restrictions on the conditional variance matrices.

A sufficient (though not necessary) condition for use of PCSE is that the conditional and unconditional variances are the same. (Note also that as with the SUR estimators above, we require that Ω_M and Ω_T *not* vary with t and i, respectively.)

For example, the *Cross-section SUR (PCSE)* method handles cross-section correlation (period clustering) by replacing the outer product of the cross-section residuals in the White cross-section (period clustered) computation with an estimate of the (contemporaneous) cross-section residual covariance matrix Ω_M:

Analogously, the *Period SUR (PCSE)* handles between period correlation (cross-section clustering) by replacing the outer product of the period residuals in the White period (cross-section clustered) computation with an estimate of the period covariance Ω_T:

The two diagonal forms of these estimators, Cross-section weights (PCSE), and Period weights (PCSE), 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. (2005). *Econometric Analysis of Panel Data, Third 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 51. 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 274 of *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 52. "Panel Estimation," on page 1135 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 46, "Dated Panels" on page 288, and "Undated Panels" on page 293 of *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** dropdown menu, and fill out the dialog as desired. Here, we create a balanced quarterly panel (ranging from 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), which are provided in "Jtrain.WF1".

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/Structure/Resize Current Page...** 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.

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.

	FN	DATEID
1 - 35	1	1935-01-01
1 - 36	1	1936-01-01
1 - 37	1	1937-01-01
1 - 38	1	1938-01-01
1 - 39	1	1939-01-01
1 - 40	1	1940-01-01
1 - 41	1	1941-01-01
1 - 42	1	1942-01-01
1 - 43	1	1943-01-01
1 - 44	1	1944-01-01
1 - 45	1	1945-01-01
1 - 46	1	1946-01-01
1 - 47	1	1947-01-01
1 - 48	1	1948-01-01
1 - 49	1	1949-01-01
1 - 50	1	1950-01-01
1 - 51		

Here, we see the observation labels in an annual panel workfile formed using the cross-section identifiers and a two-digit year identifier.

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: 06/17/07 Time: 16:04
Name: GRUNFELD_BALTAGI_PANEL
Number of pages: 1

Page: Untitled
Workfile structure: Panel - Annual
Indices: FN x DATEID
Panel dimension: 10 x 20
Range: 1935 1954 x 10 -- 200 obs

Object	Count	Data Points
series	7	1400
coef	1	751
Total	8	2151

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 alpha 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 Count	Cumulative 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	Cumulative 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 cxid = @crossid
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.

	CXID	CELLID	WITHINID
A - 91	1	1	1
A - 92	1	2	2
A - 93	1	3	3
A - 94	1	4	4
A - 99	1	7	5
B - 93	2	3	1
B - 96	2	5	2
B - 98	2	6	3

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.

```
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 275 of *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:

```
smpl 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:

```
smpl 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:

```
smpl @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:

```
smpl @first @first+5
smpl @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:

```
smpl @all
smpl @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:

```
smpl @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 140 of *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.

Panel Spreadsheets

When looking at the spreadsheet view of a series in a panel workfile the default view will be to show the "stacked" form of the series - each cross-section's data will be below the previous cross-section's data.

You may change this by clicking on the **Wide +/-** button (you will almost certainly need to widen the window to see the button as it is far to the right of the more commonly used buttons). The first time you click the button, EViews will change the display of the series such that each row of the spreadsheet contains data for a specific date, and each column contains data for a cross-section.

Clicking the **Wide +/-** button a second time transposes this so cross-sections are now shown per row, and dates per column.

	1935	1936	1937	1938	1939
1	3078.50	4661.70	5387.10	2792.20	4313.20
2	1362.40	1807.10	2676.30	1801.90	1957.30
3	1170.60	2015.80	2803.30	2039.70	2256.20
4	417.50	837.80	883.90	437.90	679.70
5	157.70	167.90	192.90	156.70	191.40
6	197.00	210.30	223.10	216.70	286.40
7	138.00	200.10	210.10	161.20	161.70
8	191.50	516.00	729.00	560.40	519.90
9					

A third click of the button takes the view back to the original stacked form.

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 calender defined by the observed dates in the workfile.

See also "Panel Workfile Functions" on page 640 and "Panel Trend Functions" on page 640 of the *Command and Programming Reference* for discussion.

By-Group Statistics

The "by-group" statistical functions ("By-Group Statistics" on page 598 of the *Command and Programming 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:

```
series temp = lstat - @meansby(lstat, townid, "@all")
scalar within_std = @stdev(temp)
```

while the "between" standard deviation may be calculated from

```
smpl if @obsid = 1
series temp = lstat - @meansby(lstat, @crossid, "@all")
scalar between_std = @stdev(temp)
```

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 1124, 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 (provided in the workfile "Grunfeld_Baltagi.WF1") 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 for the series of interest 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. Alternately, 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.

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 241 of *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...**. Alternately, right-click to select the series then drag the selected series onto the tab for the destination page. EViews will open the **Link Dialog**, prompting you to specify a method for summarizing the data.

Suppose, for example, that you select the C01, F, and I series from the panel page and then **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 FN cross-section ID series. The default **Contraction method** is set to compute the mean values of the series for each value of the 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:

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.

Panel Data 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.

Discussion of specific features may be found in:

- Chapter 52. "Panel Estimation," on page 1135.
- Chapter 53. "Panel Cointegration Estimation," on page 1191.
- Chapter 54. "Panel Statistics," on page 1213.

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 52. 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 the 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 1111).

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 1097, and in "Estimation Background," beginning on page 1184.

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** dropdown menu to choose between **LS - Least Squares (LS and AR)**, **TSLS - Two-Stage Least Squares (TSLS and AR)**, and **GMM / DPD - Generalized Method of Moments / Dynamic Panel Data** techniques. If you select the 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 1138).

The remaining estimation supported estimation techniques do not account for the panel structure of your workfile, save for lags not crossing the boundaries between cross-section units.

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** dropdown menus. By default, EViews assumes that there are no effects so that both dropdown menus are set to **None**. You may change the default settings to allow for either **Fixed** or **Random** effects in either the cross-sec-

tion or period dimension, or both. See the pool discussion of "Fixed and Random Effects" on page 1100 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 (clustering by period), while the **Period SUR** allows for general correlation of residuals across periods for a specific cross-section (clustering by individual). **Cross-section weights** and **Period weights** allow for heteroskedasticity in the relevant dimension.

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 heteroskedasticity and contemporaneous correlation. Similarly, **Period weights** allows for period heteroskedasticity, while **Period SUR** corrects for heteroskedasticity and general correlation of observations within a 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 1101 for additional details.

Lastly, you should specify a method for computing coefficient covariances. You may use the dropdown menu 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. Select **White cross-section (period cluster)** for one-way period clustering, **White period (cross-section clustering)** for one-way cross-section clustering, **White (diagonal)** for unstructured heteroskedasticity robust covariances, or **White two-way cluster** for clustering by both cross-section and period. Click on the checkbox **No d.f. correction** to perform the calculations without the leading degree of freedom correction term.

Each of the coefficient covariance methods is described in greater detail in "Robust Coefficient Covariances" on page 1107.

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 1086.

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** dropdown menu 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 of "Instrumental Variables" on page 1105 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** dropdown menu 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. As in 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 1155).

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.

Second, the dialog presents you with a new dropdown menu 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 1136). If you select **White cross-section**, for example, EViews uses GMM weights that are formed assuming that there is contemporaneous correlation between cross-sections.

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 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), use 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 1138). 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.

Pooled Mean Group ARDL Estimation

Autoregressive Distributed Lag (ARDL) models are standard least squares regressions which include lags of both the dependent variable and independent variables as regressors.

In panel settings with individual effects, standard regression estimation of ARDL models is problematic due to bias caused by correlation between the mean-differenced regressors and the error term. This bias only vanishes for large numbers of observations T, and cannot be corrected by increasing the number of cross-sections, N. To address this problem, a number of small T–large N, dynamic panel data GMM estimators have been developed (*e.g.*, Arellano-Bond, 1991).

In large T datasets, this assumptions underlying dynamic GMM are often inappropriate, and the estimator breaks down. In these cases, a popular alternative is the Pooled Mean Group (PMG) estimator of Pesaran, Shin and Smith (PSS, 1999). This model takes the cointegration form of the simple ARDL model and adapts it for a panel setting by allowing the intercepts, short-run coefficients and cointegrating terms to differ across cross-sections.

Specifically, the PMG model can be written as:

$$\Delta y_{i,t} = \phi_i EC_{i,t} + \sum_{j=0}^{q-1} \Delta X_{i,t-j}{}'\beta_{i,j} + \sum_{j=1}^{p-1} \lambda_{i,j}{}^{*}\Delta y_{i,t-j} + \epsilon_{i,t} \tag{52.1}$$

where

$$EC_{i,t} = y_{i,t-1} - X_{i,t}{}'\theta \tag{52.2}$$

Note that it is assumed that both the dependent variable and the regressors have the same number of lags in each cross-section. For notational convenience, it is also assumed that the regressors X, have the same number of lags q in each cross-section, but this assumption is not strictly required for estimation.

PSS derive the concentrated (with respect to the long-run coefficients, θ, and the adjustment coefficients, ϕ_i) log-likelihood function:

$$l_t(\varphi) = -\frac{T_i}{2}\sum_{i=1}^{N}\log(2\pi\sigma_i^2) - \frac{1}{2}\sum_{i=1}^{N}\frac{1}{\sigma_i^2}(\Delta Y_i - \phi_i EC_i)'H_i(\Delta Y_i - \phi_i EC_i) \tag{52.3}$$

where

$$
\begin{aligned}
\Delta Y_i &= (\Delta y_{i,1}, \Delta y_{i,2}, \ldots, \Delta y_{i,T_i})' \\
EC_i &= (EC_{i,1}, EC_{i,2}, \ldots, EC_{i,T_i})' \\
H_i &= (I_{T_i} - W_i(W_i' W_i)^{-1} W_i')^{-1} \\
W_i &= (\Delta Y_{i,-1}, \ldots, \Delta Y_{i,-p+1}, \Delta X_i, \Delta X_{i,-1}, \ldots, \Delta X_{i,-q+1}) \\
\Delta X_i &= (\Delta X_{i,1}, \Delta X_{i,2}, \ldots, \Delta X_{i,T_i})'
\end{aligned}
\tag{52.4}
$$

where, with some abuse of notation, we define the j-th lags of ΔY_i and ΔX_i as $\Delta Y_{i,-j}$ and $\Delta X_{i,-j}$, respectively.

This log-likelihood can be maximized directly. However, PSS suggest an iterative procedure based upon the first derivatives of (2). Initial least squares estimates of θ based on the regression $Y_t = \theta X_t$ (where Y_t and X_t are the stacked forms of $y_{i,t}$ and $x_{i,t}$) are used to compute estimates, using the first-derivative relationships, of ϕ_i and σ_i^2. These estimates are then used to compute new estimates of θ, and the process continues until convergence. Given the final estimates of θ, ϕ_i and σ_i^2, estimates of $\beta_{i,j}$ and $\lambda_{i,j}^*$ may be computed.

Although this iterative procedure's estimates converge to the full likelihood estimates, their covariance matrix does not. Fortunately, PSS (equation 13, page 625) provide the analytical form of the estimate of the covariance matrix based upon the coefficient estimates.

Estimating PMG Models in EViews

To estimate a Panel ARDL/PMG model in EViews, open the equation dialog by selecting **Quick/Estimate Equation…**, or by selecting **Object/New Object…/Equation** and selecting **PMG/ARDL** from the **Method** dropdown menu. EViews will then display the ARDL estimation dialog:

The first tab of the dialog, the **Specification** tab, allows you to specify the variables used in the regression, and whether to let EViews automatically detect the number of lags for each variable. Enter the dependent variable, followed by a space delimited list of dynamic regressors (i.e. regressors which will have lag terms in the model) in the **Dynamic Specification** edit box. You may then select whether you wish EViews to automatically select the number of lags for each variable, or whether the number of lags is fixed, using the **Automatic Selection** and **Fixed** radio buttons.

If you choose automatic selection, you must then select the maximum number of lags to test for the dependent variable and regressors using the **Max lags** dropdown menu. If you select to use a fixed number of lags, the same menu may be used to select the number of lags for the dependent variable and regressors. Note that, unlike the non-panel form of ARDL model selection in EViews, each regressor will be given the same number of lags even when using automatic model selection.

The **Fixed regressors** area lets you specify any fixed/static variables (regressors without lags). The **Trend specification** dropdown may be used to specify whether the model includes a constant term, or a constant and trend, or neither. Finally, any other static regressors should be entered in the **List of fixed regressors box**.

The **Options** tab of the dialog lets you specify the type of model selection to be used if you chose automatic selection on the **Specification** tab. You may choose between the Akaike Information Criterion (AIC), Schwarz Criterion (SC), or Hannan-Quinn Criterion (HQ) as methods for selection.

Once you have clicked the **OK** button on the estimation dialog, EViews will present you with the estimation results for both the long-run and short-run coefficients. The presented short-run coefficients (and standard errors) are the mean (and standard deviation) of the cross-section specific coefficients. A separate **View** menu item allows you to see the cross-section specific coefficients in detail.

Post-Estimation Views and Procedures

To view the individual cross-section short run coefficients, you can click on **View- > Cross-section Short Run Coefficients**. The resulting display shows a spool containing each cross-section's coefficients, standard errors, t-statistics and p-values.

The **Model Selection Summary** item on the **View** menu allows you to view either a **Criteria Graph** or a **Criteria Table**. The graph shows the model selection value for the twenty "best" with the lowest criterion value. The table form of the view shows the log-likelihood value, AIC, BIC and HQ values of the best twenty models in tabular form.

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 ("Harrison_panel.WF1"). The data are well known and used as an example dataset in many sources (*e.g.*, Baltagi (2005), p. 171).

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, 2005), 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 (ZN, INDUS, RAD, TAX, PTRATIO) 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 ("Jtrain.WF1") used in examples from Wooldridge (2002, p. 276 and 282).

As before, the first step is to structure the workfile as a panel work-

file. 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 dropdown menu 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. Click on **OK** to accept the options. EViews displays the results from estimation:

Dependent Variable: LSCRAP
Method: Panel Least Squares
Date: 08/18/09 Time: 12:03
Sample: 1987 1989
Periods included: 3
Cross-sections included: 54
Total panel (balanced) observations: 162
White period standard errors & covariance (no d.f. correction)
WARNING: estimated coefficient covariance matrix is of reduced rank

Variable	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. Notice that EViews warns you that the estimated coefficient covariances is not of full rank which occurs in this case since the number of periods is less than the number of cross-sections.

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 dropdown menu, **White period** and **No d.f. correction** for the coefficient covariance method on the **Panel Options** page. The results are given by:

Dependent Variable: D(LSCRAP)
Method: Panel Least Squares
Date: 08/18/09 Time: 12:05
Sample (adjusted): 1988 1989
Periods included: 2
Cross-sections included: 54
Total panel (balanced) observations: 108
White period standard errors & covariance (no d.f. correction)

Variable	Coefficient	Std. Error	t-Statistic	Prob.
C	-0.090607	0.088082	-1.028671	0.3060
D89	-0.096208	0.111002	-0.866721	0.3881
D(GRANT)	-0.222781	0.128580	-1.732624	0.0861
D(GRANT_1)	-0.351246	0.264662	-1.327147	0.1874

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: 08/18/09 Time: 12:11
Sample (adjusted): 1989 1989
Periods included: 1
Cross-sections included: 54
Total panel (balanced) observations: 54

Variable	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 on the test equation by clicking on **View/Coefficient Diagnostics/Wald-Coefficient Restrictions...** and entering the restriction "C(1) = -0.5" in the edit box:

Wald Test:
Equation: Untitled
Null Hypothesis: C(1)=-0.5

Test Statistic	Value	df	Probability
t-statistic	5.525812	53	0.0000
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.

The formal test confirms our casual observation, strongly rejecting the null hypothesis.

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 entering the keyword `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** dropdown menu 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 (LUCLMS) 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 ("Abond_pan.WF1") 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 the keyword `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** dropdown menu 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 dropdown menu. Click on **Next** to continue to the next page, where you will specify the remaining explanatory variables.

In the next 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.

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.

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 drop-down menu, 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)".

Pooled Mean Group Estimation

As an example of the pooled mean group estimator, we follow the application given in PSS estimating consumption functions for OECD countries in the years 1962–1993. The data are in the file "OECD.wf1". The series CONS contains per capita real private consumption for each country. INF is a measure of inflation, and INC is per capita real disposable income. PSS estimate an ARDL(1,1,1) model with the natural log of CONS as the dependent variable and the logs of INF and INC as the two dynamic regressors, with a constant as a static regressor.

We can estimate this in EViews by clicking on **Quick/Estimate Equation** and then selecting **PMG/ARDL** in the method dropdown. We enter

```
log(cons) log(inf) log(inc)
```

in the **Dynamic Specification** box, select the **Fixed** radio button to turn off model selection, and change the number of dependent lags and regressor lags to 1. To match PSS, we change the estimation sample to "1962 1993". Leaving all other options at their default values, we click **OK** to estimate the model.

The results are as shown:

Dependent Variable: DLOG(CONS)
Method: ARDL
Date: 03/11/15 Time: 09:44
Sample: 1962 1993
Included observations: 767
Dependent lags: 1 (Fixed)
Dynamic regressors (1 lag, fixed): LOG(INF) LOG(INC)
Fixed regressors: C

Variable	Coefficient	Std. Error	t-Statistic	Prob.*
		Long Run Equation		
LOG(INF)	-0.464856	0.056705	-8.197798	0.0000
LOG(INC)	0.904398	0.008687	104.1114	0.0000
		Short Run Equation		
COINTEQ01	-0.199949	0.032126	-6.223894	0.0000
D(INF)	-0.018883	0.025371	-0.744295	0.4570
DLOG(INC)	0.327691	0.057425	5.706388	0.0000
C	0.154571	0.021679	7.130068	0.0000
Log likelihood	2327.084			

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

The coefficient on the log of inflation is an estimate of the long-run inflation elasticity, and is negative (and strongly significant), as economic theory, and PSS, expect. We would expect the long run income elasticity (the coefficient on the log of income) to be equal to one, but the estimated value is slightly less at 0.904. We can perform a Wald test of unit elasticity by clicking on **View/Coefficient Diagnostics/Wald Test**, and entering the restriction of `C(2)=1`. This test rejects the null hypothesis of unit elasticity.

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 Diagnostics/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 Diagnostics/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 Test
Equation: UNTITLED
Specification: D(LSCRAP) C D89
Omitted Variables: GRANT GRANT_1

	Value	df	Probability
F-statistic	1.529525	(2, 104)	0.2215
Likelihood ratio	3.130883	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 remainder of the results shows summary information and the test equation estimated under the unrestricted alternative:

F-test summary:

	Sum of Sq.	df	Mean Squares
Test SSR	1.017443	2	0.508721
Restricted SSR	35.60793	106	0.335924
Unrestricted SSR	34.59049	104	0.332601
Unrestricted SSR	34.59049	104	0.332601

LR test summary:

	Value	df
Restricted LogL	-93.32896	106
Unrestricted LogL	-91.76352	104

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 Diagnostics/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 Test
Equation: UNTITLED
Specification: LSCRAP C D88 D89 UNION GRANT GRANT_1
Redundant Variables: GRANT GRANT_1

	Value	df	Probability
F-statistic	1.832264	(2, 156)	0.1635

F-test summary:

	Sum of Sq.	df	Mean Squares
Test SSR	0.911380	2	0.455690
Restricted SSR	39.70907	158	0.251323
Unrestricted SSR	38.79769	156	0.248703
Unrestricted SSR	38.79769	156	0.248703

Here we see that the statistic value of 1.832 does not, at conventional significance levels, lead us to reject 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:

Restricted Test Equation:
Dependent Variable: LSCRAP
Method: Panel EGLS (Cross-section random effects)
Date: 08/18/09 Time: 12:39
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

Variable	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 important 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 (2005), in which we estimate a two-way fixed effects model using data in "Gasoline.WF1". 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 ("Cross-section F" and "Cross-section Chi-square") 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 cross-section effects are redundant.

The next two tests evaluate the significance of the period dummies in the unrestricted model against a restricted specification in which there are cross-section effects only. Both forms of the statistic strongly reject the null of no period effects.

The remaining results evaluate the joint significance of all of the effects, respectively. Both of the test statistics reject the restricted model in which there is only a single intercept.

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 2005, p. 66).

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 (2005) considers an example of Hausman testing (Example 1, p. 70), in which the results for a Swamy-Arora random effects estimator for the Grunfeld data ("Grunfeld_baltagi_panel.WF1") are compared with those obtained from the corresponding fixed effects estimator. To perform this test we *must first estimate a random effects estimator*, obtaining the results:

Dependent Variable: I
Method: Panel EGLS (Cross-section random effects)
Date: 08/18/09 Time: 12:50
Sample: 1935 1954
Periods included: 20
Cross-sections included: 10
Total panel (balanced) observations: 200
Swamy and Arora estimator of component variances

Variable	Coefficient	Std. Error	t-Statistic	Prob.
C	-57.83441	28.88930	-2.001932	0.0467
F	0.109781	0.010489	10.46615	0.0000
C01	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
C01	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: I
Method: Panel Least Squares
Date: 08/18/09 Time: 12:51
Sample: 1935 1954
Periods included: 20
Cross-sections included: 10
Total panel (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
C01	0.310065	0.017355	17.86656	0.0000

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		

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.

LM Tests for Random Effects

Testing for the existence of cross-section (individual) and time effects is important in panel and pool regression settings since accounting for the presence of these effects is necessary for correct specification of the regression and proper inference. As a result, a large number of effects tests have been considered in the literature. See, for example, the survey by Baltagi (2008).

EViews offers testing for individual and time effects using both F-statistic (likelihood ratio) and Lagrange multiplier (LM) tests. The follow discussion describes LM testing for random effects (*the F-statistic tests for fixed effects are described elsewhere in this manual*).

The most popular random effects test is the Breusch-Pagan (1980) LM test. Honda (1985) derives component LM tests with one-sided alternatives, obtaining a uniformly most powerful (UMP) test statistic. Moulton and Randolph (1989) propose a standardized version of the Honda test that has improved asymptotic size. King and Wu (1997) introduce a locally mean most powerful (LMMP) one-sided LM test. In addition, Baltagi and Li (1992), Baltagi, Chang and Li (1999) extend the Breusch-Pagan, Honda, and King and Wu approaches to unbalanced designs.

The EViews panel effects (PE) test view computes the following LM tests:

- Conventional LM (Breusch-Pagan, 1980)
- Uniformly most powerful LM (Honda, 1985)
- Standardized LM (Moulton and Randolph, 1989; Honda, 1991; Baltagi et al., 1999)
- Locally mean most powerful (LMMP) (King and Wu, 1997)
- Gourieroux, Holly, and Monfort (1982)

All of these tests may be computed from estimated regressions for equation objects in a panel structured workfile, or estimated pool objects in a non-panel workfile. Note that EViews offers these tests for equations estimated using both regression and instrumental variables so long as the equations are free of estimated effects, AR terms, and GLS weighting, despite the fact that these LM tests are not, strictly speaking, applicable in the instrumental variables case. One should employ appropriate caution in the use of such results in this setting.

Background

Our discussion follows closely the survey by Baltagi (2008). We consider two-way error components disturbances:

$$u_{it} = \mu_i + \lambda_t + \nu_{it} \tag{52.5}$$

for cross-sections $i = 1, \ldots, N$ and periods $t = 1, 2, \ldots, T_i$ where μ_i are unobservable individual effects, λ_i are unobservable time effects, and ν_{it} is the remaining idiosyncratic disturbance.

The LM tests are derived under the assumption that the unobserved individual effects are distributed as independent $N(0, \sigma_\mu^2)$, the unobservable time effects are independent $N(0, \sigma_\lambda^2)$, and the idiosyncratic disturbances are independent $N(0, \sigma_\nu^2)$. The null hypotheses to be tested are: no individual effects ($H_0^\mu : \sigma_\mu^2 = 0$); no time effects ($H_0^\lambda : \sigma_\lambda^2 = 0$); and no individual and time effects ($H_0^{\mu\lambda} : \sigma_\mu^2 = \sigma_\lambda^2 = 0$).

For the remaining discussion, it will be useful to write the component specification in stacked matrix form. Let $T = max(T_1, T_2, \ldots, T_N)$. Then, define the cross-section residual vector $u_i = (u_{i1}, u_{i2}, \ldots, u_{iT_i})'$ and stacked residuals $u = (u_1', u_2', \ldots, u_N')'$, we have

$$u = D_\mu \mu + D_\lambda \lambda + \nu \tag{52.6}$$

where D_μ is a $(\sum T_i) \times N$ matrix of cross-section dummies, D_λ is a $(\sum T_i) \times T$ matrix of period dummies, and ν is defined analogously to u.

For a balanced panel, $D_\mu = (I_N \otimes \iota_T)$ and $D_\lambda = (\iota_N \otimes I_T)$, where I_k is a k-dimensional identity matrix and i_k a k-dimensional unit vector. We also have $D_\mu D_\mu' = (I_N \otimes J_T)$ and $D_\lambda D_\lambda' = (J_N \otimes I_T)$, for $J_k = \iota_k \iota_k'$ a $k \times k$ matrix of ones.

Breusch-Pagan Two-Sided Test

Breusch and Pagan (1980) derive the two-sided LM test for error components in balanced panels. Define

$$A_r = \left[\left(\frac{\tilde{u}' D_r D_r' \tilde{u}}{\tilde{u}' \tilde{u}}\right) - 1\right] \text{ for } r = (\mu, \lambda) \tag{52.7}$$

where $\tilde{u}$ are the residuals obtained from the restricted model. Then defining

$$LM_\mu = \frac{NT}{2(T-1)} A_\mu^2 \tag{52.8}$$

and

$$LM_\lambda = \frac{NT}{2(N-1)} A_\lambda^2 \tag{52.9}$$

we obtain the LM statistic for testing $H_0^{\mu\lambda}$:

$$LM_{\mu\lambda} = LM_\mu + LM_\lambda$$

which is distributed as a χ_2^2 under the null. To test H_0^μ and H_0^λ, we may use LM_μ and LM_λ respectively, both of which are distributed as a χ_1^2 under corresponding null.

Baltagi and Li (1990) derived corresponding statistics for unbalanced samples:

$$\begin{aligned} LM_\mu &= \frac{n^2}{2} \frac{A_\mu^2}{(M_\mu - n)} \\ LM_\lambda &= \frac{n^2}{2} \frac{A_\lambda^2}{(M_\lambda - n)} \end{aligned} \tag{52.10}$$

where

$$n = \sum_{i=1}^{N} T_i$$
$$M_\mu = \sum_{i=1}^{N} T_i^2 \tag{52.11}$$
$$M_\lambda = \sum_{t=1}^{T} N_t^2$$

which simplifies for the balanced case where $T_i = T$ and $N_t = N$.

Honda UMP One-Sided Test

One shortcoming of the Breusch-Pagan test is that it assumes that the alternative hypothesis is two-sided even though the variance components cannot be negative.

Honda(1985) derives a uniformly most powerful LM_μ statistic for $H_0^\mu : \sigma_\mu^2 = 0$ against a one-sided $H_1^\mu : \sigma_\mu^2 > 0$:

$$HO_\mu = \sqrt{\frac{NT}{2(T-1)}} A_\mu \to N(0, 1) \tag{52.12}$$

Similarly, for the one-sided test of $H_0^\lambda : \sigma_\lambda^2 = 0$ against $H_1^\lambda : \sigma_\lambda^2 > 0$ we have:

$$HO_\lambda = \sqrt{\frac{NT}{2(T-1)}} A_\lambda \to N(0, 1) \tag{52.13}$$

Note that both of these statistics are the square roots of the corresponding Breusch-Pagan LM statistics.

Honda's statistics can be generalized to the unbalanced case yielding square roots of the unbalanced Breusch-Pagan LM statistics:

$$HO_\mu = \frac{n}{\sqrt{2(M_\mu - n)}} A_\mu$$
$$HO_\lambda = \frac{n}{\sqrt{2(M_\lambda - n)}} A_\lambda \tag{52.14}$$

Honda does not derive a uniformly most powerful statistic $H_0^{\mu\lambda} : \sigma_\mu^2 = \sigma_\lambda^2 = 0$ against the one-sided alternative, but does suggest a "*handy*" one-sided test statistic:

$$HO_{\mu\lambda} = (HO_\mu + HO_\lambda) / \sqrt{2} \tag{52.15}$$

which also converges to a $N(0, 1)$.

King and Wu

King and Wu (1997) propose locally mean most powerful (LMMP) one-sided test statistics KW_μ and KW_λ for $H_0^\mu : \sigma_\mu^2 = 0$ against a one-sided $H_1^\mu : \sigma_\mu^2 > 0$ and for $H_0^\lambda : \sigma_\lambda^2 = 0$ against $H_1^\lambda : \sigma_\lambda^2 > 0$. These two statistics are identical to the corresponding Honda UMP statistics.

Baltagi, Chang, and Li (1992) derive the corresponding LMMP test for $H_0 : \sigma_\mu^2 = \sigma_\lambda^2 = 0$ against the one-sided alternative:

$$KW_{\mu\lambda} = \sqrt{\frac{T-1}{N+T-2}} A_\mu + \sqrt{\frac{N-1}{N+T-2}} A_\lambda \rightarrow N(0, 1) \tag{52.16}$$

and Baltagi, Chang, and Li (1999) obtain results for unbalanced case:

$$KW_{\mu\lambda} = \sqrt{\frac{M_\mu - n}{M_\mu + M_\lambda - 2n}} A_\mu + \sqrt{\frac{M_\lambda - n}{M_\mu + M_\lambda - 2n}} A_\lambda \tag{52.17}$$

Standardized LM Tests

Moulton and Randolph (1989) showed that the asymptotic approximation for the one-sided statistics can be poor when the number of regressors is large or the inter-correlation of regressors is high. Alternatively, they propose a standardized one-sided LM (SLM) statistic which centers and scales the statistic so that its mean is zero and its variance is one.

For $H_0^\mu : \sigma_\mu^2 = 0$ against a one-sided $H_1^\mu : \sigma_\mu^2 > 0$, they show that the standardized Honda (or King-Wu statistic) is given by:

$$SLM_\mu = \frac{HO_\mu - E(HO_\mu)}{\sqrt{Var(HO_\mu)}} \rightarrow N(0, 1) \tag{52.18}$$

Expressions for the expected value and variance may be found in Moulton and Randolph (1989) and Baltagi (2008).

The one-sided statistic SLM_λ for $H_0^\lambda : \sigma_\lambda^2 = 0$ against a one-sided $H_1^\lambda : \sigma_\gamma^2 > 0$ is defined analogously:

$$SLM_\lambda = \frac{HO_\lambda - E(HO_\lambda)}{\sqrt{Var(HO_\lambda)}} \rightarrow N(0, 1) \tag{52.19}$$

For the two-way model, Honda (1991) proposes a standardized Honda-type SLM test statistic, and Baltagi, Chang and Li (1999) describe a standardized King-Wu statistic. Under $H_0 : \sigma_\mu^2 = \sigma_\lambda^2 = 0$, these SLM statistics are asymptotically distributed as $N(0, 1)$ and their critical values should be more accurate than those of the corresponding unstandardized tests. See Baltagi, Chang, and Li (1999) and Baltagi (2008) for details.

Gourieroux, Holly, Monfort test

Gourieroux, Holly and Monfort (1982) and Baltagi, Chang and Li (1992) account for the possibility of negative estimates of the variance components with the following modification of the LM test under the null hypothesis $H_0: \sigma_\mu^2 = \sigma_\lambda^2 = 0$ against the two-sided alternative

$$GHM = \begin{cases} LM_\mu + LM_\lambda & \text{if } \sqrt{LM_\mu} > 0,\ \sqrt{LM_\lambda} > 0 \\ LM_\mu & \text{if } \sqrt{LM_\mu} > 0,\ \sqrt{LM_\lambda} \le 0 \\ LM_\lambda & \text{if } \sqrt{LM_\mu} \le 0,\ \sqrt{LM_\lambda} > 0 \\ 0 & \text{if } \sqrt{LM_\mu} \le 0,\ \sqrt{LM_\lambda} \le 0 \end{cases} \rightarrow \chi^2_{\text{mixed}} \tag{52.20}$$

where χ^2_{mixed} is a mixed χ^2 distribution with $\chi^2_{\text{mixed}} \sim (1/4)\chi_0^2 + (1/2)\chi_1^2 + (1/4)\chi_2^2$. The weights $1/4$, $1/2$ and $1/4$ are from Gourieroux *et al.* (1982), χ_0^2 is zero with probability one, and LM_μ and LM_λ are asymptotically independent of each other.

Example

The LM test for random effects view implements Lagrange multiplier tests of individual or/and time effects based on the results of the pooling model. As an example we use the Grunfeld (1958) data which contains 10 large US manufacturing firms over 20 years (1935–1954), which is available in the workfile "Grunfeld_Baltagi_panel.wf1" in the "Working with Panel Data" folder in your "Example Files" directory.

Following Grunfeld (1958), we consider the following investment equation:

$$I_{it} = \alpha + \beta_1 F_{it} + \beta_2 C_{it} + u_{it} \tag{52.21}$$

where I_{it} denotes real gross investment for firm i in year t; F_{it} is the real value of the firm (share outstanding); and C_{it} is the real value of the capital stock. We estimate this model using ordinary pooled least squares on the specification:

```
i f c c01
```

and name our equation EQ01. The results are shown as below:

Equation: EQ01 Workfile: GRUNFELD_BALTAGI...

View | Proc | Object | Print | Name | Freeze | Estimate | Forecast | Stats | Resids

Dependent Variable: I
Method: Panel Least Squares
Date: 02/06/15 Time: 11:17
Sample: 1935 1954
Periods included: 20
Cross-sections included: 10
Total panel (balanced) observations: 200

Variable	Coefficient	Std. Error	t-Statistic	Prob.
F	0.115562	0.005836	19.80259	0.0000
C	-42.71437	9.511676	-4.490730	0.0000
C01	0.230678	0.025476	9.054808	0.0000

R-squared	0.812408	Mean dependent var	145.9583
Adjusted R-squared	0.810504	S.D. dependent var	216.8753
S.E. of regression	94.40840	Akaike info criterion	11.94802
Sum squared resid	1755850.	Schwarz criterion	11.99750
Log likelihood	-1191.802	Hannan-Quinn criter.	11.96805
F-statistic	426.5757	Durbin-Watson stat	0.219599
Prob(F-statistic)	0.000000		

To test for the presence of individual and time effects in this model, we can click on **View/ Fixed-Random Effects Testing/Omitted Random Effects - Lagrange Multiplier** menu item. The results of the LM tests are shown as below:

Lagrange Multiplier Tests for Random Effects
Null hypotheses: No effects
Alternative hypotheses: Two-sided (Breusch-Pagan) and one -sided (all others) alternatives

	Test Hypothesis		
	Cross-section	Time	Both
Breusch-Pagan	798.1615 (0.0000)	6.453882 (0.0111)	804.6154 (0.0000)
Honda	28.25175 (0.0000)	-2.540449 (0.9945)	18.18064 (0.0000)
King-Wu	28.25175 (0.0000)	-2.540449 (0.9945)	21.83221 (0.0000)
Standardized Honda	32.66605 (0.0000)	-2.432565 (0.9925)	16.29814 (0.0000)
Standardized King-Wu	32.66605 (0.0000)	-2.432565 (0.9925)	20.96591 (0.0000)
Gourieroux, et al.	--	--	798.1615 (0.0000)

From the first column, we see that there is strong evidence that there are unaccounted for cross-section random effects in the pooled estimator residuals. All three of the cross-section tests have p-values well below conventional significance levels.

However, for testing time-specific effects, there is a marked difference between the results for the two-sided Breusch-Pagan and the one-sided tests, with the former suggesting the presence of effects, and the latter with negative values indicating that there are no time-effects. These data clearly show the benefits of using one-sided tests in an empirical setting.

Panel Cross-section Dependence Test

It is commonly assumed that disturbances in panel data models are cross-sectionally independent, especially when the cross-section dimension (N) is large. There is, however, considerable evidence that cross-sectional dependence is often present in panel regression settings.

Ignoring cross-sectional dependence in estimation can have serious consequences, with unaccounted for residual dependence resulting in estimator efficiency loss and invalid test statistics.

There are a variety of tests for cross-section dependence in the literature. EViews offers the following tests:

- Breusch-Pagan (1980) LM
- Pesaran (2004) scaled LM
- Baltagi, Feng, and Kao (2012) bias-corrected scaled LM
- Pesaran (2004) CD

These four tests may be computed from panel and pool equations estimated by least squares and instrumental variables. They may also be computed for series in a panel workfile.

Background

Following Pesaran (2004), suppose that we have a panel data model

$$y_{it} = \beta_{it}{}' x_{it} + u_{it} \tag{52.22}$$

for $i = 1, 2, \ldots, N$ and $t = 1, 2, \ldots, T$ where x_{it} is a k-dimensional column vector of regressors, β_i are the corresponding cross-section specific vectors of parameters to be estimated. (Pesaran points out that while this specification has cross-section specific coefficients, the tests described below are also applicable to the more restrictive fixed and random effects models).

The general null hypothesis of no cross-section dependence may be stated in terms of the correlations between the disturbances in different cross-section units:

$$H_0 : \rho_{ij} = \mathrm{Corr}(u_{it}, u_{jt}) = 0 \quad \text{for } i \neq j \tag{52.23}$$

For balanced samples, ρ_{ij} is the product-moment correlation coefficients of the residuals

$$\hat{\rho}_{ij} = \frac{\sum_{t \in (i,j)}^{T_{kj}} \hat{u}_{it}\hat{u}_{jt}}{\left(\sum_{t \in (i,j)}^{T_{kj}} \hat{u}_{it}^2\right)^{1/2}\left(\sum_{t \in (i,j)}^{T_{kj}} \hat{u}_{jt}^2\right)^{1/2}} \tag{52.24}$$

In the unbalanced case, Pesaran proposes use of the centered correlation coefficient

$$\hat{\rho}_{ij} = \frac{\sum_{t \in (i,j)} (\hat{u}_{it} - \tilde{u}_i)(\hat{u}_{jt} - \tilde{u}_i)}{\sqrt{\sum_{t \in (i,j)} (\hat{u}_{it} - \tilde{u}_i)^2}\sqrt{\sum_{t \in (i,j)} (\hat{u}_{jt} - \tilde{u}_i)^2}} \tag{52.25}$$

where the notation $t \in (i, j)$ is used to indicate that we sum over the subset of T_{ij} observations common to i and j, and the pairwise mean

$$\tilde{u}_i = \frac{\sum_{t \in (i,j)} \hat{u}_{it}}{T_{ij}} \tag{52.26}$$

is used to adjust for the fact that the residuals in pairwise subsets are not necessarily mean zero.

(Note that in practice EViews always employs centered correlations as in Equation (52.25) as this allows for estimation methods where the residuals are not constrained to have zero means in each cross-section. These results may differ from those that would have been obtained using the non-centered correlations in Equation (52.24). EViews will provide a message informing you when non-zero means are found.)

Breusch-Pagan LM

The most well-known cross-section dependence diagnostic is the Breusch-Pagan (1980) Lagrange Multiplier (LM) test statistic. In a seemingly unrelated regressions context, Breusch and Pagan show that under the null hypothesis in Equation (52.23), a LM statistic for dependence is given by:

$$LM = \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} T_{ij}\hat{\rho}_{ij}^2 \rightarrow \chi^2_{\frac{N(N-1)}{2}} \tag{52.27}$$

where the $\hat{\rho}_{ij}$ are the correlation coefficients obtained from the residuals of the model as described above. The asymptotic χ^2 distribution is obtained for N fixed as $T_{ij} \rightarrow \infty$ for all (i, j), and follows from a normality assumption on the errors.

Pesaran Scaled LM

It is well known that the standard Breusch-Pagan LM test statistic is not appropriate for testing in large N settings. To address this shortcoming, Pesaran (2004) proposes a standardized version of the LM statistic

$$LM_S = \sqrt{\frac{1}{N(N-1)}} \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} (T_{ij}\hat{\rho}_{ij}^2 - 1) \to N(0, 1) \tag{52.28}$$

which is asymptotically standard normal as first $T_{ij} \to \infty$ and then $N \to \infty$.

Pesaran notes one shortcoming of the scaled LM which is that $E(T_{ij}\hat{\rho}_{ij}^2 - 1)$ is not centered at zero for finite T_{ij}, so that the statistic is likely to exhibit size distortion for small T_{ij}, and that the distortion will worsen for larger N.

Pesaran CD

To address the size distortion of LM and LM_S, Pesaran (2004) proposes an alternative statistic based on the average of the pairwise correlation coefficients $\hat{\rho}_{ij}$:

$$CD_p = \sqrt{\frac{2}{N(N-1)}} \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} T_{ij}\hat{\rho}_{ij} \to N(0, 1) \tag{52.29}$$

which is asymptotically standard normal for $T_{ij} \to \infty$ and $N \to \infty$ in any order.

Further, Pesaran points out that for a wide array of panel data models, the mean of CD is exactly equal to zero for all $T_{ij} > k + 1$ and all N, so that the CD test is likely to have good properties for both N and T_{ij} small, and he provides Monte Carlo evidence to support this claim.

Baltagi, Feng, and Kao Bias-corrected Scaled LM

Baltagi, Feng, and Kao (2012) offer a simple asymptotic bias correction for the scaled LM test statistic:

$$LM_{BC} = \sqrt{\frac{1}{N(N-1)}} \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} (T_{ij}\hat{\rho}_{ij}^2 - 1) - \frac{N}{2(T-1)} \to N(0, 1) \tag{52.30}$$

For a fixed effects homogeneous panel data model with $T_{ij} \to \infty$, $N \to \infty$, and $N/T_{ij} \to c_{ij} \in (0, \infty)$, Baltagi, *et al.* show that the scaled LM has an asymptotic bias term of $N/(2(T-1))$ resulting from the incidental parameters problem since, for small T_{ij}, the within residuals are estimated imprecisely. (Note that in Equation (52.30) we extend the slightly Baltagi, *et al.* scaled LM test to unbalanced designs by using the maximum of T_{ij} for T and requiring that $T_{ij}/T \to 1$ as $T_{ij} \to \infty$).

Note that EViews will not compute the biased corrected LM statistic unless the equation was estimated with cross-section fixed effects.

Example

We illustrate the use of cross-section dependence tests for equation objects using an empirical example from Baltagi (2008) examining gasoline demand in 18 OECD countries over the period 1960–1978 (Table 2.8, p. 29).

We download the data and create a panel-structured workfile by entering the following command in the EViews command window

```
wfopen http://www.wiley.com///wileychi/baltagi/supp/Gasoline.dat
    lastobs=342
```

and clicking on **Finish** in the import wizard to accept the default settings.

The equation of interest is a cross-section fixed effects regression of log motor gasoline consumption per auto (LNGASPCAR), on log real per capital income (LINCOMEP), log real motor gasoline price (LRPMG), log real motor gasoline price the log of the stock of cars per capita (LCARPCAP).

We estimate this fixed effect specification by entering the command:

```
equation gas.ls(cx=f) lgaspcar c lincomep lrpmg lcarpcap
```

which creates the equation object GAS and displays the estimation results:

Dependent Variable: LGASPCAR
Method: Panel Least Squares
Date: 02/05/15 Time: 17:22
Sample: 1960 1978
Periods included: 19
Cross-sections included: 18
Total panel (balanced) observations: 342

Variable	Coefficient	Std. Error	t-Statistic	Prob.
C	2.402670	0.225309	10.66387	0.0000
LINCOMEP	0.662250	0.073386	9.024191	0.0000
LRPMG	-0.321702	0.044099	-7.294964	0.0000
LCARPCAP	-0.640483	0.029679	-21.58045	0.0000

Effects Specification

Cross-section fixed (dummy variables)

R-squared	0.973366	Mean dependent var	4.296242
Adjusted R-squared	0.971706	S.D. dependent var	0.548907
S.E. of regression	0.092330	Akaike info criterion	-1.867450
Sum squared resid	2.736491	Schwarz criterion	-1.631979
Log likelihood	340.3340	Hannan-Quinn criter.	-1.773645
F-statistic	586.5556	Durbin-Watson stat	0.326578
Prob(F-statistic)	0.000000		

Implicit in our approach to estimation in this example and in the validity of the computed *t*-statistics is the assumption that the errors for different cross-sectional units are uncorrelated.

To test for the presence of cross-sectional dependence, we click on **View/Residual Diagnostics/Cross-section Dependence Test**

EViews will compute the cross-section dependence tests and display the results in the object window:

The top of the table displays the test hypothesis and information about the number of cross-section and period observations in the panel. The bottom portion of the table contains the test results.

The first line contains results for the Breusch-Pagan LM test. EViews shows the test statistic value, test degree-of-freedom, and the associated p-value. In this case, the value of the test statistic, 1027.14 is well into the upper tail of a χ^2_{153}, and we strongly reject the null of no correlation at conventional significance levels.

The next two lines present results for the two scaled Breusch-Pagan tests. Both the Pesaran scaled Breusch-Pagan LM, and the Baltagi *et al.* bias-adjusted LM tests are asymptotically standard normal, and the test statistic results of 49.97 and 49.47 respectively, strongly reject the null at conventional levels. Note that in this example, the bias correction has a relatively small effect on the scaled LM statistic as N and T are of similar magnitude.

Since T is relatively small, we may instead wish to focus on the results for the asymptotically standard normal Pesaran CD test which are presented in the final line of the table. While the test statistic value of 3.25 is significantly below that of the scaled LM tests, the Pesaran CD test still rejects the null at conventional significance levels.

Arellano-Bond Serial Correlation Testing

For models estimated by GMM, you may compute the first and second order serial correlation statistics proposed by Arellano and Bond (1991) as one method of testing for serial correlation. The test is actually two separate statistics, one for first order correlation and one for second. If the innovations are *i.i.d.* we expect the first order statistic to be significant (with a negative auto-correlation coefficient), and the second order statistic to be insignificant.

The statistics are calculated as:

$$m_j = \frac{\rho_j}{\sqrt{VAR(\rho_j)}} \tag{52.31}$$

$$\rho_j = \frac{1}{T-3-j} \sum_{t=4+j}^{T} \rho_{tj} \tag{52.32}$$

$$\rho_{tj} = E(\Delta\epsilon_{i,t}, \Delta\epsilon_{i,t-j}) \tag{52.33}$$

where ρ_j is the average j-th order autocovariance.

(Note that this test is only available for equations estimated by GMM using first difference cross-section effects.)

To perform the test click on **View/Residual Diagnostics/Arellano-Bond Serial Correlation Test**. EViews will then calculate the test statistics for both first and second order correlation and display them in one table.

As an illustration, we again use the workfile ABOND_PAN.WF1 which contains data on a firm level panel, as examined in Arellano and Bond (1991), and Doornik, Bond and Arellano (2006). The following command estimates the GMM example used in the "GMM Example"

on page 1155 section, but uses ordinary standard error estimates, instead of the White period standard errors used above:

```
gmm(cx=fd, per=f, gmm=perwhite, iter=oneb, levelper) n n(-1) n(-2)
    w w(-1) k ys ys(-1) @ @dyn(n,-2) w w(-1) k ys ys(-1)
```

This equation replicates the estimates shown in Table 4(b), page 290, of Arellano Bond (1991).

Dependent Variable: N
Method: Panel Generalized Method of Moments
Transformation: First Differences
Date: 12/20/12 Time: 14:47
Sample (adjusted): 1979 1984
Periods included: 6
Cross-sections included: 140
Total panel (unbalanced) observations: 611
White period instrument weighting matrix
Instrument specification: @DYN(N,-2) W W(-1) K YS YS(-1)
@LEV(@SYSPER)
Constant added to instrument list

Variable	Coefficient	Std. Error	t-Statistic	Prob.
N(-1)	0.474150	0.085303	5.558409	0.0000
N(-2)	-0.052968	0.027284	-1.941324	0.0527
W	-0.513205	0.049345	-10.40027	0.0000
W(-1)	0.224640	0.080063	2.805796	0.0052
K	0.292723	0.039463	7.417748	0.0000
YS	0.609775	0.108524	5.618813	0.0000
YS(-1)	-0.446371	0.124815	-3.576272	0.0004
@LEV(@ISPERIOD("1979"))	0.010509	0.007251	1.449224	0.1478
@LEV(@ISPERIOD("1980"))	0.014142	0.009959	1.420077	0.1561
@LEV(@ISPERIOD("1981"))	-0.040453	0.011551	-3.502122	0.0005
@LEV(@ISPERIOD("1982"))	-0.021640	0.011891	-1.819843	0.0693
@LEV(@ISPERIOD("1983"))	-0.001847	0.010412	-0.177358	0.8593
@LEV(@ISPERIOD("1984"))	-0.010221	0.011468	-0.891270	0.3731

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
Prob(J-statistic)	0.220105		

Once estimated we click on **View/Residual Diagnostics/Arellano-Bond Serial Correlation Test** to view the serial correlation test results. The table displays the results for a test of both first and second order serial correlation:

Arellano-Bond Serial Correlation Test
Equation: Untitled
Date: 02/04/13 Time: 23:58
Sample: 1976 1984
Included observations: 611

Test order	m-Statistic	rho	SE(rho)	Prob.
AR(1)	-2.427825	-2.106427	0.867619	0.0152
AR(2)	-0.332535	-0.075912	0.228281	0.7395

Although the original 1991 Arellano Bond paper does not display results for the first order test, the same data are used as an example in Doornik, Bond and Arellano 2006 (page 11), which does display corrected results for both tests.

The tests show that the first order statistic is statistically significant, whereas the second order statistic is not, which is what we would expect if the model error terms are serial uncorrelated in levels.

Estimation Background

The basic class of models that can be estimated using panel techniques may be written as:

$$Y_{it} = f(X_{it}, \beta) + \delta_i + \gamma_t + \epsilon_{it} \tag{52.34}$$

The leading case involves a linear conditional mean specification, so that we have:

$$Y_{it} = \alpha + X_{it}'\beta + \delta_i + \gamma_t + \epsilon_{it} \tag{52.35}$$

where Y_{it} is the dependent variable, and X_{it} is a k-vector of regressors, and ϵ_{it} are the error terms for $i = 1, 2, \ldots, M$ cross-sectional units observed for dated periods $t = 1, 2, \ldots, T$. 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 (50.2) on page 1097, 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 1097 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 (2005).

Background

The basic GMM panel estimators are based on moments of the form,

$$g(\beta) = \sum_{i=1}^{M} g_i(\beta) = \sum_{i=1}^{M} Z_i'\epsilon_i(\beta) \tag{52.36}$$

where Z_i is a $T_i \times p$ matrix of instruments for cross-section i, and,

$$\epsilon_i(\beta) = (Y_i - f(X_{it}, \beta)) \tag{52.37}$$

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:

$$\begin{aligned} S(\beta) &= \left(\sum_{i=1}^{M} Z_i'\epsilon_i(\beta)\right)' H \left(\sum_{i=1}^{M} Z_i'\epsilon_i(\beta)\right) \\ &= g(\beta)'Hg(\beta) \end{aligned} \tag{52.38}$$

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,

$$V(\hat{\beta}) = (G'HG)^{-1}(G'H\Lambda HG)(G'HG)^{-1} \tag{52.39}$$

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:

$$G(\beta) = \left(-\sum_{i=1}^{M} Z_i'\nabla f_i(\beta)\right) \tag{52.40}$$

In the simple linear case where $f(X_{it}, \beta) = X_{it}'\beta$, we may write the coefficient estimator in closed form as,

$$\begin{aligned}\hat{\beta} &= \left(\left(\sum_{i=1}^{M} Z_i' X_i\right)' H \left(\sum_{i=1}^{M} Z_i' X_i\right)\right)^{-1} \left(\left(\sum_{i=1}^{M} Z_i' X_i\right)' H \left(\sum_{i=1}^{M} Z_i' Y_i\right)\right) \\ &= (M_{ZX}' H M_{ZX})^{-1} (M_{ZX}' H M_{ZY})\end{aligned} \tag{52.41}$$

with variance estimator,

$$V(\hat{\beta}) = (M_{ZX}' H M_{ZX})^{-1} (M_{ZX}' H \Lambda H M_{ZX}) (M_{ZX}' H M_{ZX})^{-1} \tag{52.42}$$

for M_{AB} of the general form:

$$M_{AB} = M^{-1}\left(\sum_{i=1}^{M} A_i' B_i\right) \tag{52.43}$$

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:

$$\begin{aligned} H &= (\hat{\sigma}^2 M_{ZZ})^{-1} \\ \Lambda &= \hat{\sigma}^2 M_{ZZ} \end{aligned} \tag{52.44}$$

Substituting, we have the familiar expressions,

$$\begin{aligned}\hat{\beta} &= (M_{ZX}'(\hat{\sigma}^2 M_{ZZ})^{-1} M_{ZX})^{-1} (M_{ZX}'(\hat{\sigma}^2 M_{ZZ})^{-1} M_{ZY}) \\ &= (M_{ZX}' M_{ZZ}^{-1} M_{ZX})^{-1} (M_{ZX}' M_{ZZ}^{-1} M_{ZY})\end{aligned} \tag{52.45}$$

and,

$$V(\hat{\beta}) = \hat{\sigma}^2 (M_{ZX}' M_{ZZ}^{-1} M_{ZX})^{-1} \tag{52.46}$$

Standard errors that are robust to conditional or unconditional heteroskedasticity and contemporaneous correlation may be computed by substituting a new expression for Λ,

$$\Lambda = T^{-1}\left(\sum_{t=1}^{T} Z_t' \hat{\epsilon}_t \hat{\epsilon}_t' Z_t\right) \tag{52.47}$$

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 1107.

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 1107. For example, the Cross-section SUR (3SLS) weighting matrix is computed as:

$$H = \left(T^{-1} \sum_{t=1}^{T} Z_t' \hat{\Omega}_M Z_t \right)^{-1} \tag{52.48}$$

where $\hat{\Omega}_M$ is an estimator of the contemporaneous covariance matrix. Similarly, the White period weights are given by:

$$H = \left(M^{-1} \sum_{i=1}^{M} Z_i' \hat{\epsilon}_i \hat{\epsilon}_i' Z_i \right)^{-1} \tag{52.49}$$

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:

$$g(\beta) = \sum_{i=1}^{M} g_i(\beta) = \sum_{i=1}^{M} Z_i' \hat{\Omega}^{-1} \epsilon_i(\beta) \tag{52.50}$$

Dynamic Panel Data

Consider the linear dynamic panel data specification given by:

$$Y_{it} = \sum_{j=1}^{p} \rho_j Y_{it-j} + X_{it}'\beta + \delta_i + \epsilon_{it} \tag{52.51}$$

First-differencing this specification eliminates the individual effect and produces an equation of the form:

$$\Delta Y_{it} = \sum_{j=1}^{p} \rho_j \Delta Y_{it-j} + \Delta X_{it}'\beta + \Delta\epsilon_{it} \tag{52.52}$$

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 (52.52). 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:

$$W_i = \begin{bmatrix} Y_{i1} & 0 & 0 & \dots & \dots & \dots & \dots & 0 \\ 0 & Y_{i1} & Y_{i2} & \dots & \dots & \dots & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & Y_{i1} & Y_{i2} & \dots & Y_{iT_i-2} \end{bmatrix} \tag{52.53}$$

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,

$$H^d = \left(M^{-1} \sum_{i=1}^{M} Z_i' \Xi Z_i \right)^{-1} \tag{52.54}$$

where Ξ is the matrix,

$$\Xi = \frac{1}{2} \begin{bmatrix} 2 & -1 & 0 & \dots & 0 & 0 \\ -1 & 2 & 0 & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & 2 & -1 \\ 0 & 0 & 0 & \dots & -1 & 2 \end{bmatrix} \sigma^2 \tag{52.55}$$

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:

$$H = \left(M^{-1} \sum_{i=1}^{M} Z_i' \Delta\epsilon_i \Delta\epsilon_i' Z_i \right)^{-1} \tag{52.56}$$

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:

$$H = \left(M^{-1} \sum_{i=1}^{M} Z_i' Z_i \right)^{-1} \tag{52.57}$$

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, B. H. (2005). *Econometric Analysis of Panel Data, Third Edition*. New York: John Wiley & Sons.

Baltagi, B. H. (2008). *Econometric Analysis of Panel Data*, 4th ed. New York: John Wiley & Sons.

Baltagi, B. 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.

Baltagi, B. H., Chang, Y. J., and Q. Li (1999), "Testing For Random Individual And Time Effects Using Unbalanced Panel Data," *Advances in Econometrics,* 13, 1–20.

Baltagi, B. H., Chang, Y. J., and Q. Li (1992). "Monte Carlo Results on Several New and Existing Tests for the Error Components Model," *Journal of Econometrics,* 54, 95–120.

Baltagi, B. H, Feng, Q., and C. Kao (2012). "A Lagrange Multiplier test for Cross-sectional Dependence in a Fixed Effects Panel Data Model," *Journal of the Econometrics*, 170, 164–177.

Breusch, T., and A. Pagan (1980). "The Lagrange Multiplier Test and its Application to Model Specification in Econometrics," *Review of Economic Studies*, 47, 239–253.

Gourieroux, C., A. Holly, and A. Monfort (1982). "Likelihood Ratio Test, Wald Test, and Kuhn-Tucker Test in Linear Models with Inequality Constraints on the Regression Parameters," *Econometrica* 50, 63–80.

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, J. A. (1978). "Specification Tests in Econometrics," *Econometrica*, 46, 1251–1272.

Honda, Y. (1985) "Testing the Error Components Model with Non-normal Disturbances," *Review of Economic Studies*, 52, 681–690.

Honda, Y. (1991) "A Standardized Test For the Error Components Model with the Two-Way Layout," *Economics Letters*, 37, 125–128.

Hsiao, C., Pesaran M. H., and Andreas Pick (2012). "Diagnostic Tests of Cross Section Independence in Limited Dependent Variable Panel Data Models," *Oxford Bulletin of Economics & Statistics*, 74, 253–277.

King, M. L. and P. X. Wu (1997). "Locally Optimal One Sided Tests for Multiparameter Hypotheses," *Econometric Reviews*, 16, 131–156.

Layard, R. and S. J. Nickell (1986). "Unemployment in Britain," *Economica*, 53, S121–S169.

Moulton, B. R. and W. C. Randolph (1989). "Alternative Tests of the Error Components Model," *Econometrica,* 57, 685–93.

Papke, L. E. (1994). "Tax Policy and Urban Development: Evidence From the Indiana Enterprise Zone Program," *Journal of Public Economics*, 54, 37-49.

Pesaran, M. H. (2004) "General Diagnostic Tests for Cross Section Dependence in Panels," University of Cambridge, Faculty of Economics, Cambridge Working Papers in Economics No. 0435.

White, H. (1980)."A Heteroskedasticity-Consistent Covariance Matrix and a Direct Test for Heteroskedasticity," *Econometrica*, 48, 817–838.

Wooldridge, J. M. (2002). *Econometric Analysis of Cross Section and Panel Data*, Cambridge, MA: The MIT Press.

Chapter 53. Panel Cointegration Estimation

The analysis of long-run cointegrating relationships has received considerable attention in modern time series analysis. In this chapter, we describe EViews' tools for estimating cointegrating relationships using panel data. We consider various forms of the residual-based panel Fully Modified OLS (FMOLS) and Dynamic OLS (DOLS) estimators (Phillips and Moon, 1999; Pedroni, 2000, 2001; Kao and Chiang, 2000; Mark and Sul, 2003) that produce asymptotically unbiased, normally distributed coefficient estimates.

If you have not already done so, we recommend you familiarize yourself with the material in Chapter 28. "Cointegrating Regression," beginning on page 293 which describes the closely related on-panel versions of the estimators in this chapter. In addition, Baltagi (2008), Baltagi and Kao (2000), and Breitung and Pesaran (2005) offer useful surveys of the literature.

Background

We will work with the standard triangular representation of a regression specification and assume the existence of a single cointegrating vector as in Hansen (1992). Consider a panel structure for the $n+1$ dimensional time series vector process (y_{it}, X_{it}'), with cointegrating equation

$$y_{it} = X_{it}'\beta + D_{1it}'\gamma_{1i} + u_{1it} \tag{53.1}$$

for cross-sections i and periods t, where $D_{it} = (D_{1it}', D_{2it}')'$ are deterministic trend regressors and the n stochastic regressors X_t are governed by the system of equations:

$$\begin{aligned} X_{it} &= \Gamma_{21i}'D_{1it} + \Gamma_{22i}'D_{2it} + \epsilon_{2it} \\ \Delta\epsilon_{2it} &= u_{2it} \end{aligned} \tag{53.2}$$

The p_1-vector of D_{1it} regressors enter into both the cointegrating equation and the regressors equations, while the p_2-vector of D_{2it} are deterministic trend regressors which are included in the regressors equations but excluded from the cointegrating equation (see "Cointegrating Regression" on page 293 for further discussion).

It is worth mentioning that most authors have focused attention on the leading case in which the deterministic trend terms in the panel cointegrating equation consist only of cross-section dummy variables:

$$\begin{aligned} y_{it} &= X_{it}'\beta + \gamma_i + u_{1it} \\ \Delta X_{it} &= u_{2it} \end{aligned} \tag{53.3}$$

Notice that the cointegrating relationship between y and X is assumed to be homogeneous across cross-sections, and that the specification allows for cross-section specific deterministic effects.

Following Phillips and Moon (1999), we define the long run covariance matrices for the errors in cross-section $u_{it} = (u_{1it}, u_{2it}')'$ are strictly stationary and ergodic with zero mean, contemporaneous covariance matrix Σ_i, one-sided long-run covariance matrix Λ_i, and long-run covariance matrix Ω_i, each of which we partition conformably with u_{it}

$$\Sigma_i = E(u_{it}u_{it}') = \begin{bmatrix} \sigma_{11i} & \sigma_{12i} \\ \sigma_{21i} & \Sigma_{22i} \end{bmatrix}$$

$$\Lambda_i = \sum_{j=0}^{\infty} E(u_{it}u_{it-j}') = \begin{bmatrix} \lambda_{11i} & \lambda_{12i} \\ \lambda_{21i} & \Lambda_{22i} \end{bmatrix} \tag{53.4}$$

$$\Omega_i = \sum_{j=-\infty}^{\infty} E(u_{it}u_{it-j}') = \begin{bmatrix} \omega_{11i} & \omega_{12i} \\ \omega_{21i} & \Omega_{22i} \end{bmatrix} = \Lambda_i + \Lambda_i' - \Sigma_i$$

and define what Phillips and Moon term the long-run average covariance matrices $\Lambda = E(\Lambda_i)$, and $\Omega = E(\Omega_i)$.

Lastly, and perhaps most importantly, we assume that there is independence in the errors across cross-sections.

Given this basic structure, we may define panel estimators of the cointegrating relationship coefficient β using extensions of single-equation FMOLS and DOLS methods. There are different variants for each of the estimators depending on the assumptions that one wishes to make about the long-run covariances and how one wishes to use the panel structure of the data.

We begin by describing how to estimate panel FMOLS and DOLS models in EViews. We then discuss the views and procedures associated with the panel equation and offer a simple example. Lastly, we provide technical details on the estimation methods.

Estimating Panel Cointegration in EViews

EViews offers two basic methods for estimating a single cointegrating vector in panel settings: Fully Modified OLS (FMOLS) and Dynamic OLS (DOLS).

To estimate your panel equation you must first be working with a panel structured workfile. Next, select **Object/New Object…/Equation** or **Quick/Estimate Equation…** then select **COINTREG - Cointegrating Regression** in the **Method** dropdown menu. Alternately, you may enter the `cointreg` keyword in the command window.

EViews will open the panel version of the cointegrating equation dialog.

This dialog is almost identical to the dialog for cointegrating regression in non-panel settings.

As before, there are three parts of the equation specification: the equation and cointegrating regressors specification, the nonstationary estimation settings, and the sample settings (We ignore, for a moment, the related settings on the **Options** tab.)

Since the sample specification should be self-explanatory, we focus attention on the specification of the cointegrating and regressors equations and the nonstationary estimation settings.

Specifying the Equation

The **Equation specification** and **Cointegrating regressors specification** are used to describe your triangular regression specification.

Equation Specification

You should enter the name of the dependent variable, y, followed by a list of cointegrating regressors, X, in the **Equation specification** edit field, then use the **Trend specification** drop-down menu to specify the deterministic trend components (**None**, **Constant (Level)**, **Linear Trend**, **Quadratic Trend**). Your selection will include all trends up to the specified order. You may use the **Deterministic regressors** edit box to add deterministic trend regressors that are not offered in the pre-specified list.

Cointegrating Regressors Specification

When performing FMOLS estimation you may use the **Additional trends** drop-down menu and the **Additional deterministic regressors** edit field to specify D_2 deterministic trends that are included in the regressors equations but not in the cointegrating equation.

You may also indicate whether you wish to estimate the regressors innovations u_{2t} in Equation (53.2) indirectly by estimating the regressors equations in levels and then differencing the residuals or directly obtain $\hat{u}_{2t}$, by estimating the differenced regressors equations. Check the **Estimate using differenced data** box to estimate the regressors equations in differences.

Estimation Methods and Settings

The **Nonstationary estimation settings** section is used to specify the estimation method. You should use the **Method** dropdown menu to choose between FMOLS and DOLS estimation. When you make your selection, the main dialog page and the options page will both change to display the options associated with your selection.

Fully-Modified OLS

To estimate your equation using panel FMOLS, select **Fully-modified OLS (FMOLS)** in the **Nonstationary estimation settings** dropdown menu. The main dialog and options pages will change to show the available settings.

First, you should choose between the pooled, weighted, and group mean (averaged) FMOLS estimators:

- **Pooled** estimation performs standard FMOLS on the pooled sample after removing the deterministic components from both the dependent variable and the regressors.
- **Pooled (weighted)** estimation accounts for heterogeneity by using cross-section specific estimates of the long-run covariances to reweight the data prior to computing pooled FMOLS.
- **Grouped** mean estimation computes the cross-section average of the individual cross-section FMOLS estimates.

See "Fully-Modified OLS," on page 1205 for a detailed description of the methods

Additionally, you may click on the **Long-run variances: Options** button to specify options for computing the long-run covariances. By default, EViews will estimate the individual and long-run average covariance matrices using a (non-prewhitened) kernel approach with a

Bartlett kernel function and Newey-West fixed bandwidth. To change the whitening or kernel settings, click on the **Options** button and enter your changes in the sub-dialog.

Here we have specified that the long-run covariances be computed using a nonparametric method with the quadratic-spectral kernel and a real-valued bandwidth chosen by Andrews' automatic bandwidth selection method. Click on **OK** to accept the updated settings.

Lastly, you can specify the form of the first-stage cointegrating equation regression that EViews uses to obtain $\hat{u}_{it}$ for computing the long-run covariances. By default, the first-stage regression assumes homogeneous long-run coefficients, but you may allow for different coefficients by selecting the **Heterogeneous first-stage long-run coefficients** checkbox.

Clicking on the **Options** tab of the estimation dialog shows the settings for computing the coefficient covariance for the long-run coefficients and specifying the default coefficient name:

For pooled estimation, you may choose between the **Default (homogeneous variances)** moment estimator or a **Sandwich (heterogeneous variances)** method as described in

"Pooled FMOLS," on page 1205. You may also elect to apply or not apply a degrees-of-freedom adjustment to the estimated coefficient covariance.

The pooled weighted and grouped methods only offer the **d.f. Adjustment** option.

Dynamic OLS

To estimate your equation using DOLS, first fill out the equation specification, then select **Dynamic OLS (DOLS)** in the **Nonstationary estimation settings** dropdown menu. The dialog will change to display the settings for DOLS.

You should use the **Panel method** drop-down to choose between the pooled, weighted, and group mean (averaged) DOLS estimators:

- **Pooled** estimation performs standard DOLS on the pooled sample of data after first removing the deterministic components from both the dependent variable and the regressors.
- **Pooled (weighted)** estimation accounts for heterogeneity by using cross-section specific estimates of the conditional long-run residual variances to reweight the moments for each cross-section when computing the pooled DOLS estimator.
- **Grouped** mean estimation computes the cross-section average of the individual cross-section DOLS estimates.

If you specify pooled weighted estimation, EViews will display a **Long-run var wgts: Options** button which will allow you to specify the settings used in computing the long-run variances for use as weights.

Next, you should specify the method of selecting leads and lags. By default, the **Lag & lead method** is **Fixed** with **Lags** and **Leads** each set to 1. You may specify a different number of lags or leads or you can use the dropdown to enable automatic information criterion selection of the lag and lead orders for each cross-section by selecting **Akaike**, **Schwarz**, or **Hannan-Quinn**. Note that the automatic lag selection method is conducted by estimating separate regressions for each cross-section. If you select **None**, EViews will estimate static OLS.

If you select one of the info criterion selection methods, you will be prompted for a maximum lag and lead length. You may enter a value, or you may retain the default entry "*" which instructs EViews to use an arbitrary observation-based rule-of-thumb for each cross-section:

$$\mathrm{int}(\min((T_i - k)/3,\ 12) \cdot (T_i/100)^{1/4}) \tag{53.5}$$

to set the maximum, where k is the number of coefficients in the equation. As in the non-panel setting, we urge careful thought in the use of automatic selection methods since the purpose of including leads and lags is to remove long-run dependence, and automatic methods were not designed for this purpose.

When you are done modifying the main estimation settings, click on the **Options** tab of the dialog to see the options for computing the long-run coefficient covariance matrix estimates and specifying the default coefficient name:

For pooled estimation you will be prompted to specify the **Default (homogeneous variances)** moment estimator or a **Sandwich (heterogeneous variances)** method as described in "Pooled DOLS," on page 1208. You will also be prompted to specify the use of a **Long-run variance** estimator or **Ordinary variance** estimator for use in scaling the moment matrix or in computing the individual variance weights sandwich estimator, and to choose whether to perform a **d.f. Adjustment**.

Pooled weighted estimation offers only a choice of whether to perform the degree-of-freedom correction (since the long-run variance settings are specified on the first page of the dialog).

Grouped estimation offers a variety of choices for computing the individual coefficient covariance matrices.You may use the **Individual covariances method** drop-down to choose between the **Default (rescaled OLS)**, **Ordinary Least Squares**, **White**, or **HAC - Newey West.**

The **Default (rescaled OLS)** method re-scales the ordinary least squares coefficient covariance using an estimator of the long-run variance of DOLS residuals (multiplying by the ratio of the long-run variance to the ordinary squared standard error). Alternately, you may employ a sandwich-style **HAC (Newey-West)** covariance matrix estimator.

In both cases, you may use the options button (labeled **Options** or **HAC Options,** respectively) to override the default method for computing the long-run variance (non-prewhitened Bartlett kernel and a Newey-West fixed bandwidth). You may also select **White** covariances or **Ordinary Least Squares** covariances. The latter two methods are offered primarily for comparison purposes.

Working With a Panel Cointegrating Equation

Once you estimate your equation, EViews offers a variety of views and procedures for examining the properties of the equation, testing, forecasting, and generating data. Generally speaking, these views and procedures are the same as those available in other estimation settings such as least squares estimation. In some cases there have been modifications to account for the nature of panel cointegrating regression.

Views

For the most part, the views of a cointegrating equation require little discussion.

For example, the **Estimation Output** shows the estimated coefficients and summary statistics of the equation, the **Representations** view offers text descriptions of the estimated cointegrating equation, the **Covariance Matrix** displays the coefficient covariance, and the **Residual Diagnostics** (**Correlogram - Q-statistics**, **Correlogram Squared Residuals**, **Histogram - Normality Test**) offer statistics based on residuals.

That said, a few comments about the construction of these views are in order.

- The **Estimation Output**, **Representations**, **Covariance Matrix** views of an equation only show results for the cointegrating equation and the long-run coefficients. In the representations view, the presence of individual trend coefficients is represented by the presence of the expression "[CX = DETERM]". Similarly, the **Coefficient Diagnostics** do not include any of the deterministics. Note also that the short-run dynamics added in DOLS estimation are not included in these views.

  (Note that EViews while does not display the coefficients for the deterministics and short-run dynamics, the coefficients are used in constructing relevant measures such as fit statistics and residuals.)

- You may use the **Individual Coefficients** view to examine the estimated trend coefficients for each cross-section.

- The method used to compute residuals in the **Actual, Fitted, Residual** views and the **Residual Diagnostics** views differs depending on the estimation method. For FMOLS, the values are *not* based on the transformed data; the residuals are derived by substi-

tuting the estimated coefficients into the original cointegrating equation and computing the residuals. For DOLS, the residuals from the cointegrating equation are adjusted for the deterministics and estimated short-run dynamics. Standardized residuals are simply the residuals divided through by the long-run variance estimate.

- The test statistics in the **Residual Diagnostics** are computed using the pooled residual data and probably should be used only for illustrative purposes.

Procedures

The procs for an equation estimated using panel cointegrating regression are a subset of those found in least squares estimation.

While most of the relevant issues were discussed in the previous section (*e.g.*, construction of residuals), you should note that:

- Forecasts constructed using the **Forecast...** procedure and models created using **Make Model** procedure follow the **Representations** view in omitting DOLS short-run dynamics. If you wish to construct forecasts that incorporate the short-run dynamics, you should use ordinary least squares to estimate an equation that explicitly includes the lags and leads of the cointegrating regressors.
- The forecast standard errors generated by the **Forecast...** proc and those obtained from solving models created using the **Make Model...** proc both employ the "S.E. of the regression" reported in the estimation output. This may not be appropriate.
- When creating a model from a panel equation with deterministic trends, EViews will create a series in the workfile containing the fitted values of the trend terms and will incorporate this series in the equation specification. If you wish to solve for your model with out-of-sample values, you will need to fill in the appropriate fitted values in the series.

Examples

To illustrate the estimation of panel cointegration models in EViews, we follow Kao, Chiang, and Chen (KCC, 1999) who apply panel cointegration analysis to the study of economic growth by estimating the cointegrating relationship for total factor productivity and domestic and foreign R&D capital stock.

The KCC data, which we provide in the workfile "tfpcoint.WF1" consist of annual data on log total factor productivity (LTFP), log domestic (LRD), and log foreign (LFRD) R&D capital stock for 22 countries for the years 1971 to 1990. We consider estimation of simple pooled FMOLS and DOLS estimators for the cointegrating vectors as in Table 4(i) (p. 703) and Table 5(i) (p. 704).

To begin, display the panel cointegrating equation dialog, and fill out the top portion of the dialog as depicted below:

Following KCC, we assume a fixed effect specification with LTFP as the dependent variable and LRD and LFRD as the cointegrating regressors. To handle the fixed effect we specify a **Constant (Level)** in the **Trend specification** drop-down menu.

The default panel cointegration estimation method **Pooled estimation** using **Fully-modified OLS (FMOLS)** corresponds to the estimates in Table 4(i) of KCC, so we leave those settings unchanged.

To match the KCC estimates, we click on **Long-run variances: Options** button to display the long-run covariance settings, and change the **Kernel options** by setting a user-specified bandwidth value of 6:

Click on **OK** to accept the changes. Since we wish to estimate the equation using the default coefficient covariances, we simply click on **OK** again to estimate the equation using the specified settings. EViews estimates the equation and displays the results:

Dependent Variable: LTFP
Method: Panel Fully Modified Least Squares (FMOLS)
Date: 01/14/13 Time: 15:23
Sample (adjusted): 1972 1990
Periods included: 19
Cross-sections included: 22
Total panel (balanced) observations: 418
Panel method: Pooled estimation
Cointegrating equation deterministics: C
Coefficient covariance computed using default method
Long-run covariance estimates (Bartlett kernel, User bandwidth = 6.0000)

Variable	Coefficient	Std. Error	t-Statistic	Prob.
LRD	0.082284	0.017282	4.761167	0.0000
LFRD	0.114272	0.029055	3.933005	0.0001

R-squared	0.608017	Mean dependent var	-0.016190
Adjusted R-squared	0.585135	S.D. dependent var	0.031831
S.E. of regression	0.020502	Sum squared resid	0.165613
Durbin-Watson stat	0.286810	Long-run variance	0.001347

The top portion of the dialog displays the estimation method and information about the sample employed in estimation. Just below the sample information EViews shows that the estimates are based on pooled estimation using only a constant as the cross-section specific trend regressor. The coefficient covariances are computed using the default settings, and the long-run covariances used a Bartlett kernel with the user-specified bandwidth.

The middle section shows the coefficient estimates, standard errors, and t-statistics, which differ a bit from the results in KCC Table 4(i), as KCC report estimates for a slightly different model. As in KCC, both R&D variables, LRD and LFRD are positively related to LTFP, and the coefficients are statistically significant.

The bottom portion of the output shows various summary statistics. Note in particular, the reported "Long-run variance" which shows $\hat{\omega}_{1.2}$, the estimated long-run average variance of u_{1it} conditional on u_{2it}, obtained from the DOLS residuals. The square root of this variance, 0.0367, is somewhat higher than the "S.E. of the regression" value of 0.0205, which is based on the ordinary estimator of the residual variance.

Clicking **View/Representations** shows the commands used to estimate the equation, along with a text representation of the long-run relationship:

Note the "[CX = DETERM]" component which shows that there are additional heterogeneous trend terms in the relationship (in this case the fixed effects). The presence of this term instructs EViews to use this information when constructing models, and when computing fits and forecasts.

Suppose, for example, we select **Proc/Make Model** from our estimated equation. EViews will create a model object containing the equation results:

Notice the presence of EQ4I_EFCT in the model equation and in the workfile. Double-clicking on EQ4I in the model object displays the equation:

Notice that we have replaced the deterministic components "[CX = DETERM]" in the equation specification with the fitted values in the series EQ4I_EFCT. In this case EQ4I_EFCT just holds the estimated fixed effects, but more generally it will hold the fitted values for the deterministic terms in your regression.

To estimate the model using DOLS, we again display the equation dialog, fill out the top portion as before:

and change the **Method** to **Dynamic OLS (DOLS).** To match the settings in KCC, we set the **Panel Method** to **Pooled**, and specify the **Fixed** lags and leads, with 2 lags and 1 lead:

Click on OK to estimate the equation using the default covariance method. EViews will display the results:

Dependent Variable: LTFP
Method: Panel Dynamic Least Squares (DOLS)
Date: 01/15/13 Time: 15:36
Sample (adjusted): 1974 1989
Periods included: 16
Cross-sections included: 22
Total panel (balanced) observations: 352
Panel method: Pooled estimation
Cointegrating equation deterministics: C
Fixed leads and lags specification (lead=1, lag=2)
Coefficient covariance computed using default method
Long-run variance (Bartlett kernel, Newey-West fixed bandwidth) used
for coefficient covariances

Variable	Coefficient	Std. Error	t-Statistic	Prob.
LRD	0.109353	0.023067	4.740719	0.0000
LFRD	0.047674	0.037756	1.262690	0.2082

R-squared	0.932997	Mean dependent var	-0.018869
Adjusted R-squared	0.851443	S.D. dependent var	0.034313
S.E. of regression	0.013225	Sum squared resid	0.034632
Long-run variance	0.000156		

Again, the top portion of the dialog shows the estimation method, sample, and information about settings employed in estimation. Note in particular that the default coefficient covariance matrix computation uses an estimator of the long-run variance computed using a Bartlett kernel and fixed Newey-West bandwidth.

The long-run coefficients, standard errors, and *t*-statistics are close to their counterparts in KCC Table 5(i).

We may contrast these results to the group-mean estimates of the same specifications. The group-mean FMOLS results may be obtained by calling up the original FMOLS equation and selecting **Grouped** in the **Panel method** drop-down menu. The group-mean FMOLS coefficient results are given by:

Variable	Coefficient	Std. Error	t-Statistic	Prob.
LRD	0.319009	0.021539	14.81044	0.0000
LFRD	-0.061544	0.022454	-2.740894	0.0064

which differ markedly from the pooled estimates, suggesting that heterogeneity in the cointegrating equation or the long-run covariances may be important. Likewise, the corresponding group-mean DOLS results,

Variable	Coefficient	Std. Error	t-Statistic	Prob.
LRD	0.401746	0.063727	6.304167	0.0000
LFRD	-0.093889	0.055137	-1.702828	0.0902

differ by a fair amount from the pooled results.

Technical Details

Fully-Modified OLS

Phillips and Moon (1999), Pedroni (2000), and Kao and Chiang (2000) offer extensions of the Phillips and Hansen (1990) fully modified OLS estimator to panel settings.

Pooled FMOLS

The pooled FMOLS estimator outlined by Phillips and Moon (1999) is a straightforward extension of the standard Phillips and Hansen estimator. Given estimates of the average long-run covariances, $\hat{\Lambda}$ and $\hat{\Omega}$, we may define the modified dependent variable and serial correlation correction terms

$$\tilde{y}_{it}^{+} = \tilde{y}_{it} - \hat{\omega}_{12}\hat{\Omega}_{22}^{-1}\hat{u}_2 \tag{53.6}$$

and

$$\hat{\lambda}_{12}^{+} = \hat{\lambda}_{12} - \hat{\omega}_{12}\hat{\Omega}_{22}^{-1}\hat{\Lambda}_{22} \tag{53.7}$$

where $\tilde{y}_{it}$ and $\tilde{X}_{it}$ are the corresponding data purged of the individual deterministic trends, and $\hat{\omega}_{1.2}$ is the long-run average variance of u_{1it} conditional on u_{2it}. In the leading case of individual specific intercepts, $\tilde{y}_{it} = y_{it} - \bar{y}_i$ and $\tilde{X}_{it} = X_{it} - \bar{X}_i$ are the demeaned variables.

The pooled FMOLS estimator is then given by

$$\hat{\beta}_{FP} = \left(\sum_{i=1}^{N} \sum_{t=1}^{T} \tilde{X}_{it}\tilde{X}_{it}' \right)^{-1} \sum_{i=1}^{N} \sum_{t=1}^{T} (\tilde{X}_{it}\tilde{y}_{it}^{+} - \hat{\lambda}_{12}^{+}{}') \tag{53.8}$$

It is worth noting the pooled estimator simply sums across cross-sections separately in the numerator and denominator.

The estimates of the long-run covariances may be obtained by standard approaches using the $\hat{u}_{it}$ residuals obtained from estimating Equation (53.1) and after removing the deterministic components in Equation (53.2). Note that EViews allows you to relax the assumption of common β in these first stage estimates. Given estimates of the individual long-run covariances for each cross-section, $\hat{\Lambda}_i$ and $\hat{\Omega}_i$, we form our estimators by taking simple cross-section averages:

$$\hat{\Lambda} = \sum_{i=1}^{N} \hat{\Lambda}_i \qquad \hat{\Omega} = \sum_{i=1}^{N} \hat{\Omega}_i \tag{53.9}$$

Phillips and Moon (1999) show that under appropriate assumptions, the asymptotic distribution of the pooled estimator is asymptotically normal under sequential limits as $(T, N \to \infty)$. Then

$$\lim_{T,\, N \to \infty} N^{-1/2} T(\hat{\beta}_{FP} - \beta) \to N(0, \omega_{1.2} \cdot a\Omega_{22}^{-1}) \tag{53.10}$$

for a constant a that depends on the deterministic variable specification, where $\omega_{1.2}$ is the long-run variance of u_{1t} conditional on u_{2t}, given by $\omega_{1.2} = \omega_{11} - \omega_{12}\Omega_{22}^{-1}\omega_{21}$.

Instead of estimating the asymptotic variance directly using estimates of $\omega_{1.2}$, Ω_{22}^{-1} and the corresponding a for every possible deterministic specification, EViews adopts the Pedroni (2000) and Mark and Sul (2003) approach of forming a consistent estimator using moments of the regressors:

$$\hat{V}_{FP} = \hat{\omega}_{1.2} \cdot \hat{M}_{FP}^{-1} \tag{53.11}$$

where

$$\hat{M}_{FP} = \frac{1}{N} \sum_{i=1}^{N} \left(\frac{1}{T^2} \sum_{t=1}^{T} X_{it} X_{it}' \right) \tag{53.12}$$

In related work, Mark and Sul (2003) propose a sandwich form of this estimator which allows for heterogeneous variances:

$$\hat{V}_{FP} = \hat{M}_{FP}^{-1} \cdot \hat{D}_{FP} \cdot \hat{M}_{FP}^{-1} \tag{53.13}$$

where

$$\hat{D}_{FP} = \frac{1}{N} \sum_{i=1}^{N} \left(\hat{\omega}_{1.2i} \frac{1}{T^2} \sum_{t=1}^{T} X_{it} X_{it}' \right) \tag{53.14}$$

and the long-run variance estimates $\hat{\omega}_{1.2i} = \hat{\omega}_{11i} - \hat{\omega}_{12i}\hat{\Omega}_{22i}^{-1}\hat{\omega}_{21i}$ are computed for each cross-section. Note that degree-of-freedom corrections may be applied to the $\hat{\omega}_{1.2}$ and $\hat{\omega}_{1.2i}$ for comparability with standard regression standard error of the regression estimators.

Weighted FMOLS

Pedroni (2000) and Kao and Chiang (2000) describe feasible pooled FMOLS estimators for heterogeneous cointegrated panels where the long-run variances differ across cross-sections.

We again use first-stage estimates of the long-run and regressors equations to obtain the residuals, estimate the individual long-run variances $\hat{\Lambda}_i$ and $\hat{\Omega}_i$, and let

$$\hat{\lambda}_{12i}^{+} = \hat{\lambda}_{12i} - \hat{\omega}_{12i}\hat{\Omega}_{22i}^{-1}\hat{\Lambda}_{22i} \tag{53.15}$$

and

$$\tilde{y}_{it}^{++} = \tilde{y}_{it} - \hat{\omega}_{12}\hat{\Omega}_{22}^{-1}\hat{u}_2 - \hat{\omega}_{1.2i}^{1/2}\,(\hat{\omega}_{1.2i}^{-1/2}X_{it}' - (\hat{\omega}_{12i}X_{it})')\tilde{\beta}_0 \tag{53.16}$$

for $\tilde{\beta}_0$, a preliminary estimate of the long-run coefficient.

Next, we form the weighted variables:

$$\begin{aligned} \tilde{X}_{it}^{*} &= \hat{\Omega}_{22i}^{-1/2} \cdot \tilde{X}_{it} \\ \tilde{y}_{it}^{*} &= \hat{\omega}_{1.2i}^{-1/2} \cdot \tilde{y}_{it}^{++} \\ \hat{\lambda}_{12i}^{*} &= \hat{\omega}_{1.2i}^{-1/2} \cdot \hat{\lambda}_{12i}^{+} \cdot \hat{\Omega}_{22i}^{-1/2\,\prime} \end{aligned} \tag{53.17}$$

Then the estimator is given by

$$\hat{\beta}_{FW} = \left(\sum_{i=1}^{N}\sum_{t=1}^{T}\tilde{X}_{it}^{*}\tilde{X}_{it}^{*\prime}\right)^{-1}\sum_{i=1}^{N}\sum_{t=1}^{T}(\tilde{X}_{it}^{*}\tilde{y}_{it}^{*} - \hat{\lambda}_{12i}^{*\prime}) \tag{53.18}$$

and the asymptotic covariance is estimated using a moment estimator as in Pedroni (2000):

$$\hat{V}_{FW} = \left(\frac{1}{N}\sum_{i=1}^{N}\left(\frac{1}{T^2}\sum_{t=1}^{T}\tilde{X}_{it}^{*}\tilde{X}_{it}^{*\prime}\right)\right)^{-1} \tag{53.19}$$

Group-Mean FMOLS

Pedroni (2000, 2001) proposes a grouped-mean FMOLS estimator which averages over the individual cross-section FMOLS estimates:

$$\hat{\beta}_{FG} = \frac{1}{N}\sum_{i=1}^{N}\left\{\left(\sum_{t=1}^{T}\tilde{X}_{it}\tilde{X}_{it}'\right)^{-1}\sum_{t=1}^{T}(\tilde{X}_{it}\tilde{y}_{it} - \hat{\lambda}_{12i}')\right\} \tag{53.20}$$

Pedroni (1990) notes that in the presence of heterogeneity in the cointegrating relationships, the grouped-mean estimator offers the desirable property of providing consistent estimates

of the sample mean of the cointegrating vectors, in contrast to the pooled and weighted estimators.

We estimate the asymptotic covariance matrix for this estimator by computing the variance of the average of the individual estimates:

$$\hat{V}_{FG} = \frac{1}{N^2}\sum_{i=1}^{N}\left(\hat{\omega}_{1.2i}\frac{1}{T^2}\sum_{t=1}^{T}\tilde{X}_{it}\tilde{X}_{it}'\right)^{-1} \tag{53.21}$$

It is worth noting that the basic t-statistics obtained using this covariance estimator differ from the t-statistic proposed by Pedroni (1991), which aggregates individual statistics across the cross-section dimension.

Dynamic OLS (DOLS)

Kao and Chiang (2000), Mark and Sul (1999, 2003), and Pedroni (2001) propose extensions of the Saikkonen (1992) and Stock and Watson (1993) DOLS estimator to panel data settings. Panel DOLS involves augmenting the panel cointegrating regression equation with cross-section specific lags *and leads* of ΔX_{it} to eliminate the asymptotic endogenity and serial correlation.

Pooled DOLS

Kao and Chiang (2000) describe the pooled DOLS estimator in which we use ordinary least squares to estimate an augmented cointegrating regression equation:

$$\tilde{y}_{it} = \tilde{X}_{it}'\beta + \sum_{j=-q_i}^{r_i}\Delta\tilde{X}_{it+j}'\delta_i - v_{1it} \tag{53.22}$$

where $\tilde{y}_{it}$ and $\tilde{X}_{it}$ are the data purged of the individual deterministic trends. Note that the short-run dynamics coefficients δ_i are allowed to be cross-section specific.

Let $\tilde{Z}_{it}$ be regressors formed by interacting the $\Delta\tilde{X}_{it+j}$ terms with cross-section dummy variables, and let $\tilde{W}_{it}' = (\tilde{X}_{it}', \tilde{Z}_{it}')'$. Then the pooled DOLS estimator may be written as

$$\begin{bmatrix}\hat{\beta}_{DP}\\ \hat{\gamma}_{DP}\end{bmatrix} = \left(\sum_{i=1}^{N}\sum_{t=1}^{T}\tilde{W}_{it}\tilde{W}_{it}'\right)^{-1}\left(\sum_{i=1}^{N}\sum_{t=1}^{T}\tilde{W}_{it}\tilde{y}_{it}'\right) \tag{53.23}$$

Kao and Chiang (2000) show that the asymptotic distribution of this estimator is the same as for pooled FMOLS. We may estimate the asymptotic covariance matrix of the $\hat{\beta}_{DP}$ using the corresponding sub-matrix of:

$$\hat{V}_{DP} = \hat{\omega}_{1.2}\cdot\hat{M}_{DP}^{-1} \tag{53.24}$$

where

$$\hat{M}_{DP} = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{1}{T^2}\sum_{t=1}^{T}\tilde{W}_{it}\tilde{W}_{it}'\right) \tag{53.25}$$

and $\hat{\omega}_{1.2}$ is an estimator of the long-run residual variance.

Alternately, Mark and Sul (2003) employ a sandwich estimator

$$\hat{V}_{DP} = \hat{M}_{DP}^{-1}\cdot\hat{D}_{DP}\cdot\hat{M}_{DP}^{-1} \tag{53.26}$$

where

$$\hat{D}_{DP} = \frac{1}{N}\sum_{i=1}^{N}\left(\hat{\omega}_{1.2i}\frac{1}{T^2}\sum_{t=1}^{T}\tilde{W}_{it}\tilde{W}_{it}'\right) \tag{53.27}$$

employs the individual long-run variance estimates $\hat{\omega}_{1.2i}$.

Weighted DOLS

Mark and Sul (1999) describe a simple weighted DOLS estimator which allows for heterogeneity in the long-run variances. Define the weighted regression:

$$\begin{bmatrix}\hat{\beta}_{DW}\\ \hat{\gamma}_{DW}\end{bmatrix} = \left(\sum_{i=1}^{N}\hat{\omega}_{1.2i}^{-1}\sum_{t=1}^{T}\tilde{W}_{it}\tilde{W}_{it}'\right)^{-1}\left(\sum_{i=1}^{N}\hat{\omega}_{1.2i}^{-1}\sum_{t=1}^{T}\tilde{W}_{it}\tilde{y}_{it}'\right) \tag{53.28}$$

for individual long-run variance estimates $\hat{\omega}_{1.2i}$ obtained after preliminary DOLS estimation.

In EViews, we estimate the asymptotic covariance matrix of the $\hat{\beta}_{DW}$ using the corresponding sub-matrix of:

$$\hat{V}_{DW} = \left(\frac{1}{N}\sum_{i=1}^{N}\hat{\omega}_{1.2i}^{-1}\left(\frac{1}{T^2}\sum_{t=1}^{T}\tilde{W}_{it}\tilde{W}_{it}'\right)\right)^{-1} \tag{53.29}$$

Note that this very simple form of weighted estimation differs from the more complex estimator described Kao and Chiang (2000), which mixes the FMOLS endogenity correction, weighting of both dependent variable and regressors, and the DOLS serial correlation correction.

Group-mean DOLS

Pedroni (2001) extends the grouped estimator concept to DOLS estimation by averaging over the individual cross-section DOLS estimates:

$$\begin{bmatrix} \hat{\beta}_{DG} \\ \hat{\gamma}_{DG} \end{bmatrix} = \frac{1}{N} \sum_{i=1}^{N} \left\{ \left(\sum_{t=1}^{T} \tilde{W}_{it} \tilde{W}_{it}' \right)^{-1} \sum_{t=1}^{T} \tilde{W}_{it} \tilde{y}_{it}' \right\} \tag{53.30}$$

The asymptotic covariance matrix is obtained from the corresponding sub-matrix of the variance of the average of the individual estimators:

$$\hat{V}_{FG} = \frac{1}{N^2} \sum_{i=1}^{N} \left(\hat{\omega}_{1.2i} \frac{1}{T^2} \sum_{t=1}^{T} \tilde{W}_{it} \tilde{W}_{it}' \right)^{-1} \tag{53.31}$$

We again note that the basic *t*-statistics involving this covariance estimator differ from the *t*-statistic proposed by Pedroni (1991) which aggregates individual statistics across the cross-section dimension.

References

Baltagi, Badi (2008). *Econometric Analysis of Panel Data*, New York: John Wiley & Sons.

Baltagi, Badi and Chihwa Kao (2000). "Nonstationary Panels, Cointegration in Panels and Dynamic Panels: A Survey," in Baltagi, B. H. *ed.*, *Nonstationary Panels, Panel Cointegration and Dynamic Panels,* 15, Amsterdam: Elsevier, 7–51.

Breitung, Jörg and M. Hashem Pesaran (2008). "Unit Roots and Cointegration in Panels," in Mátyás, László and Patrick Sevestre, *eds. The Econometrics of Panel Data*, Berlin: Springer-Verlag Berlin Heidelberg.

Hansen, Bruce E. (1992). "Efficient Estimation and Testing of Cointegrating Vectors in the Presence of Deterministic Trends," *Journal of Econometrics*, 53, 87-121.

Kao, Chihwa and Min-Hsien Chiang (2000). "On the Estimation and Inference of a Cointegrated Regression in Panel Data," in Baltagi, B. H. et al. *eds.*, *Nonstationary Panels, Panel Cointegration and Dynamic Panels,* 15, Amsterdam: Elsevier, 179–222.

Kao, Chihwa, Chiang, Min-Hsien, and Bangtian Chen (1999). "International R&D Spillovers: An Application of Estimation and Inference in Panel Cointegration, Oxford Bulletin of Economics and Statistics, 61, 693–711.

Mark, Nelson C. and Donggyu Sul (1999). "A Computationally Simple Cointegration Vector Estimator for Panel Data," Ohio State University manuscript.

Mark, Nelson C. and Donggyu Sul (2003). "Cointegration Vector Estimation by Panel DOLS and Long-run Money Demand," *Oxford Bulletin of Economics and Statistics*, 65, 655–680.

Pedroni, Peter (2000). "Fully Modified OLS for Heterogeneous Cointegrated Panels," in Baltagi, B. H. *ed.*, *Nonstationary Panels, Panel Cointegration and Dynamic Panels,* 15, Amsterdam: Elsevier, 93–130.

Pedroni, Peter (2001). "Purchasing Power Parity Tests in Cointegrated Panels," *The Review of Economics and Statistics*, 83, 727–731.

Phillips, Peter C. B. and Bruce E. Hansen (1990). "Statistical Inference in Instrumental Variables Regression with I(1) Processes," *Review of Economics Studies*, 57, 99-125.

Phillips, Peter C. B. and Hyungsik R. Moon (1999). "Linear Regression Limit Theory for Nonstationary Panel Data," *Econometrica*, 67, 1057-1111.

Saikkonen, Pentti (1992). "Estimation and Testing of Cointegrated Systems by an Autoregressive Approximation," *Econometric Theory*, 8, 1-27.

Stock, James H. and Mark Watson (1993). "A Simple Estimator Of Cointegrating Vectors In Higher Order Integrated Systems," *Econometrica*, 61, 783-820.

Chapter 54. Panel Statistics

EViews offers varying levels of support for computations in panel workfiles.

At the most basic level, in the absence of panel specific settings, if computation of a statistic is available in a panel workfile, EViews computes the statistic on the stacked data, accounting for seams between the data when evaluating lags and leads of variables. This class of computation implicitly assumes homogeneity across cross-sections.

In more sophisticated settings, such as the equation estimators seen in Chapter 52. "Panel Estimation," beginning on page 1135 and Chapter 53. "Panel Cointegration Estimation," on page 1191, EViews supports various procedures that account specifically for various aspects of the panel structure of the data.

In the remainder of this chapter we describe a few other calculations in EViews that are "panel aware" in the sense that there are settings that account for the panel structure of the data. Some of these topics are discussed elsewhere; in these cases, we simply provide a link to the relevant discussion.

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 or group of series in a dated workfile, open the series or group 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.

Here we see the dialog for graphing a single series. Note in particular the panel workfile specific **Panel options** section which controls how the multiple cross-sections in your panel should be handled. If you select **Stack cross sections** EViews will display a single graph of the stacked data, labeled with both the cross-section and date. For example, with a **Line & Symbol** type graph, we have

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:

Nevertheless, the graph does offer you the ability examine all of your cross-sections at-a-glance.

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.

By-Statistics

While not specifically panel aware, there are a variety of places in EViews where you may use a classification variable to compute statistics by-group. In these cases, you may use the `@crossid` identifier to compute statistics for each cross-section.

For example, you may open a series object and select **View/Stats by Classification...** to display summary statistics for various groups:

Enter `@crossid` to compute statistics by cross-section:

Descriptive Statistics for FRD
Categorized by values of @CROSSID
Date: 02/05/13 Time: 15:43
Sample (adjusted): 1971 1990
Included observations: 440 after adjustments

@CROSSID	Mean	Std. Dev.	Obs.
1	0.823050	0.329523	20
2	0.857000	0.169526	20
3	0.893250	0.153560	20
4	0.870250	0.170139	20
5	0.913400	0.106301	20
6	0.839050	0.184148	20
7	0.865650	0.174743	20
8	0.860900	0.205281	20
9	0.844000	0.226664	20
10	0.882750	0.147099	20
11	0.940250	0.199533	20
12	0.876000	0.232249	20
13	0.973600	0.124876	20
14	0.781800	0.238805	20
15	0.820250	0.132656	20
16	0.834050	0.167274	20
17	0.863000	0.247619	20
18	0.799700	0.185269	20
19	0.662100	0.160519	20
20	0.853350	0.116893	20
21	0.854800	0.198757	20
22	0.865100	0.184872	20
All	0.853332	0.195387	440

Similarly, you may test equality of means across cross-sections (**View/Equality Tests by Classification...**). 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.

One particularly useful set of non-panel specific tools that may be used for panel analysis are the by-group statistics functions (Chapter 13. "Operator and Function Reference," beginning on page 598 of the *Command and Programming Reference*). The by-group statistics which allow you to compute statistics by cross-section ID and match merge those results back to the original data. For example the simple expression

```
series ydemean = y - @meansby(y, @crossid)
```

computes the deviations from the cross-section means for the series Y and places the results in the series YDEMEAN.

Panel Covariances

Panel structured data employ more than one dimension to identify a given observation. In the most common case where the panel combines time series and cross-sectional data, we have data for cross-section units $i = 1, \ldots, N$ and periods $t = 1, \ldots, T$. In this setting, we focus on a single random variable X, with individual observations denoted X_{it}.

It is sometimes convenient to view the X for different cross-sections (or time periods) as being distinct random variables. This unstacking of a single random variable into multiple random variables permits us to define measures of association between cross-sections or periods for a given panel series.

For example, we may define the *contemporaneous* or *between cross-section covariances* for X:

$$\sigma_{ij} = E\{(X_i - E(X_i))(X_j - E(X_j))\} \tag{54.1}$$

where $X_i' = (X_{i1}, X_{i2}, \ldots, X_{iT})$ is the random variable associated with the X for the i-th cross-section, $i = 1, \ldots, N$. The contemporaneous covariances are a measure of association (dependence) between the data for different cross-sections at a given point in time.

Similarly, we may define the *period* or *within cross-section covariances* for X:

$$\sigma_{st} = E\{(X_s - E(X_s))(X_t - E(X_t))\} \tag{54.2}$$

where $X_t' = (X_{1t}, X_{2t}, \ldots, X_{Nt})$, $t = 1, \ldots, T$. The within cross-section covariances measure the association between the data in different periods for a given cross-section.

Panel covariances and correlations are widely used in panel data analysis. For example:

- Contemporaneous correlations between macroeconomic variables are often used to examine the nature of relationships between different countries (see for example, Obstfeld and Rogoff, 2001, p. 368).
- The contemporaneous covariances of residuals from panel regression are used in computing cross-sectional Zellner SUR-type estimators (Johnston and Dinardo, 1997, p. 318) and in tests of cross-section dependence (Pesaran, 2004). Similarly, panel covariances are used as a first step in obtaining common factors for unit root and other tests (Bai and Ng, 2004).
- Analogously, period covariances of residuals may be used to compute feasible GLS estimators that correct for within cross-section (cluster) correlation.

Once we unstack the data, the computation of estimates of panel measures of association for a single series follows the standard methods (cross-reference to groups). For example, the standard Pearson estimators for the contemporaneous cross-section covariance use variation across time to obtain estimates:

$$\hat{\sigma}_{ij} = \frac{1}{T}\sum_{t=1}^{T}(X_{it} - \overline{X}_i)(X_{jt} - \overline{X}_j) \tag{54.3}$$

where $\overline{X}_i = T^{-1}\sum_{t=1}^{T} X_{it}$ and $\overline{X}_j = T^{-1}\sum_{t=1}^{T} X_{jt}$.

The corresponding Pearson estimators of the period covariances use variation in the cross-section dimension to provide estimates:

$$\hat{\sigma}_{st} = \frac{1}{N}\sum_{i=1}^{N}(X_{it} - \overline{X}_t)(X_{is} - \overline{X}_s) \tag{54.4}$$

where $\overline{X}_t = N^{-1}\sum_{i=1}^{N} X_{it}$ and $\overline{X}_s = N^{-1}\sum_{i=1}^{N} X_{is}$.

Other measures of association may be defined similarly. For discussion of the various methods that EViews supports, see "Covariance Analysis," beginning on page 616 of *User's Guide I*.

Telling EViews to compute measures of association for a series in a panel structured workfile is straightforward. Simply open the series, and select **View/Panel Covariance...** to display the dialog. *Note that the workfile must be structured as a panel for the panel covariance menu entry to be available.*

EViews will open the **Covariance Analysis** dialog which provides options for controlling the computation, display, and saving of results.

For the most part, the dialog is unchanged from the covariance dialog for a group of series and the discussion of settings there is directly relevant (see "Covariance Analysis," beginning on page 616) of *User's Guide I*.

The one notable difference in the current dialog are the radio buttons that allow you to choose whether to compute **Contemporaneous covariances** or **Between periods covariances**.

By changing the settings in the **Statistics** portion dialog, you may instruct EViews to compute a variety of other measures of association (uncentered Pearson, Spearman rank correlations and Kendall's tau), as well as test statistics for whether the measure of association is zero.

In addition, you may specify a sample to be used in computation and select whether you wish EViews to employ listwise deletion to balance the sample in the event that there are missing values in the series. If you will be working with series with missing observations you should bear in mind that:

- EViews will compute covariances for all of the cross-sections (for contemporaneous covariances) or periods (for between-period covariances) in the specified sample, even if there are no valid observations for a relevant cross-section or period. If you wish to exclude periods or cross-sections from the analysis, you should do so by setting the sample.
- For cross-section covariances, checking the **Balance sample - (listwise deletion)** setting instructs EViews to balance the data by removing data for periods where there are missing values for any cross-section.
- For period covariances, the **Balance sample - (listwise deletion)** setting will remove data for entire *cross-sections* where there are missing observations for any period.

To illustrate, we follow Obstfeld and Rogoff (2001) in computing the cross-country correlations for per capita consumption growth (DCPCH) for the Group of Seven countries over the period from 1973 to 1992. The data, which are from the Penn World Table 5.6, are provided for you in the workfile "PWT56_CONSUMP.wf1" in the Example Files folder in your EViews installation directory.

Open the workfile and the series DCPCH, select **View/Panel covariance...** and fill in the dialog as depicted above. Click on **OK** to compute the requested statistics and display the results.

Panel Covariance Analysis: Ordinary
Series: DCPCH
Date: 08/17/12 Time: 14:12
Sample: 1973 1992
Included observations: 140
Analysis of contemporaneous (between cross-sections) relationships
Number of cross-sections employed: 7

Covariance
Correlation

Observations	CANADA	U.S.A.	JAPAN	FRANCE	GERMANY, WEST	ITALY	U.K.
CANADA	0.000929						
	1.000000						
	20						
U.S.A.	0.000364	0.000344					
	0.643323	1.000000					
	20	20					
JAPAN	2.39E-05	0.000156	0.000280				
	0.046872	0.504028	1.000000				
	20	20	20				
FRANCE	7.83E-05	9.83E-05	0.000117	0.000107			
	0.248871	0.513199	0.680323	1.000000			
	20	20	20	20			
GERMANY, WEST	0.000211	0.000173	0.000114	9.58E-05	0.000283		
	0.411637	0.554703	0.403295	0.551423	1.000000		
	20	20	20	20	20		
ITALY	0.000236	4.42E-05	6.15E-05	4.88E-05	8.84E-05	0.000315	
	0.436899	0.134410	0.207144	0.266630	0.296038	1.000000	
	20	20	20	20	20	20	
U.K.	0.000364	0.000358	0.000293	0.000133	0.000198	0.000160	0.000885
	0.401499	0.648652	0.588630	0.434526	0.394636	0.302919	1.000000
	20	20	20	20	20	20	20

These results show the correlations in the values of DCPCH between cross-sections.

Likewise, we may instruct EViews to compute the between period covariances, we obtain correlations between periods. Fill in the dialog as in the previous example, changing the **Type** to **Between period covariances**, and change the sample to “1973 1992” since data for DCPCH in 1972 are not available (due to the lag in the difference).

If you were to use the original sample of "1972 1992" the resulting between period correlation matrix would contain only NAs (since the balanced sample option would remove all observations). Click on **OK** to accept the settings and compute the between covariances and correlations.

Panel Principal Components

In "Panel Covariances," beginning on page 1219, we discussed how a single panel variable could be unstacked into multiple random variables, allowing us to compute measures of association between different cross-sections or periods. We may extend this basic approach by computing the principal components of the panel variable using one of the measures of association.

The following discussion assumes that you are familiar with the tools for principal components analysis in EViews. Background on the computation, display, and saving of principal components and scores is provided in "Principal Components," beginning on page 634 of *User's Guide I*.

Viewing Principal Components

To compute and display the principal components results for a panel series, open the series, and select **View/Panel Principal Components...** to display the dialog. *Note that the workfile must be structured as a panel for the panel covariance menu entry to be available.*

The first tab of the dialog, labeled **Components**, specifies the display, output, and selection settings for the principal components. The tab is virtually identical to the one displayed when you compute the principal components for a group of series (see "Performing Covariance Analysis," beginning on page 617). The one minor difference is in the edit field for the **Maximum number** of components to be retained. In the panel setting the edit field is filled with "*" which is a stand-in for the maximum number of components (number of cross-sections or periods); in the group setting, this edit field is filled in with the number of variables.

The second tab, labeled **Calculation**, controls the computation of the measure of association used in the principal components analysis.

Again, the dialog is mostly unchanged from the one used in the group principal components setting with the notable exception of the radio buttons that allow you to choose whether to compute **Contemporaneous covariances** or **Between periods covariances**.

The default calculation computes the principal components using the contemporaneous (between cross-section) correlation matrix, but you may use the **Type** and **Method** dropdowns choose to use the uncentered Pearson, Spearman rank correlations and Kendall's tau.

Saving Component Scores

One common task is saving the principal components scores for use in subsequent analysis. Accordingly, EViews provides easy to use tools for saving the scores from your panel principal components analysis in the workfile. As these tools are virtually identical to those documented in "Saving Component Scores" on page 645, here, we offer only an abbreviated description.

To save the component scores in the workfile, you will use panel principal components procedure. Click on **Proc/Make Panel Principal Components...** to display the dialog.

The dialog is virtually unchanged from the one displayed for saving principal components scores of a group; indeed the first tab is identical.

In the first tab you will describe the output you wish EViews:

- You should provide names for the series in which you wish to save the scores, and optionally, names for the loadings and eigenvector matrices, and the eigenvalues vector.
- Importantly, the **Scaling** dropdown on the bottom of the dialog is used to determine the properties of your scores. By default, the scaling is set to **Normalize loadings** so that the scores have variances equal to the eigenvalues of the decomposition. You may instead elect to save normalized scores (**Normalize scores**), equal weighted scores and loadings (**Symmetric weights**), or user weighted loadings (**User loading weight**).

The second tab is used to describe the computation of the measure of association (used in the computation). The options are those for computing panel covariances as described in "Viewing Principal Components" on page 1224.

An Illustration

To illustrate, we compute principal components of the cross-country correlations for per capita consumption growth (DCPCH) for the Group of Seven countries over the period from 1973 to 1992 (Obstfeld and Rogoff, 2001). The data, which are from the Penn World Table 5.6, are provided for you in the workfile "PWT56_CONSUMP.wf1" in the Example Files folder in your EViews installation directory.

Open the workfile and the series DCPCH, select **View/Panel Principal Components...** and click on **OK** to compute the principal components using the default settings and to display the basic results in table form:

Panel Principal Components Analysis
Series: DCPCH
Date: 08/29/12 Time: 15:55
Sample: 1972 1992
Included observations: 147
Analysis of contemporaneous (between cross-section) relationships
Computed using: Ordinary correlations
Extracting 7 of 7 possible components

Eigenvalues: (Sum = 7, Average = 1)

Number	Value	Difference	Proportion	Cumulative Value	Cumulative Proportion
1	3.735308	2.644171	0.5336	3.735308	0.5336
2	1.091137	0.246793	0.1559	4.826445	0.6895
3	0.844344	0.214297	0.1206	5.670789	0.8101
4	0.630047	0.258421	0.0900	6.300836	0.9001
5	0.371627	0.163062	0.0531	6.672463	0.9532
6	0.208565	0.089592	0.0298	6.881027	0.9830
7	0.118973	---	0.0170	7.000000	1.0000

Eigenvectors (loadings):

Cross-section	PC 1	PC 2	PC 3	PC 4	PC 5	PC 6	PC 7
CANADA	0.326828	0.680374	-0.151109	-0.035987	0.329621	-0.022346	0.544974
U.S.A.	0.429951	0.113511	-0.499405	-0.018909	0.172759	0.356079	-0.629172
JAPAN	0.386426	-0.539826	0.095487	0.196637	0.118652	0.556905	0.432733
FRANCE	0.408552	-0.346313	0.187099	-0.283874	0.506102	-0.574024	-0.109172
GERMANY, WEST	0.392449	0.019595	0.049549	-0.656370	-0.634238	0.030900	0.095454
ITALY	0.280698	0.330954	0.793017	0.245221	-0.047112	0.154858	-0.310595
U.K.	0.399098	-0.054291	-0.228706	0.623011	-0.432216	-0.456208	0.048876

Here we see header information describing the computation and two out of three of the sections of output. The first section provides a summary of the eigenvalues of the correlation matrix, while the second section shows the corresponding eigenvectors. Not depicted here, but present in the actual output, is the estimated correlation matrix itself.

These results in the first section show that the first four components account for about 90% of the total scaled variance in the values of DCPCH between cross-sections. The second section describes the linear combination coefficients. We see that the first principal component (labeled "PC1") is a roughly-equal linear combination of all seven of the country per capita consumption growth. This component might be thought of as representing the common component in G7 consumption growth.

Alternately, we may instruct EViews to compute and graph the eigenvalues associated with the between period correlations. Click on **View/Panel Principal Components...** to display the dialog

go to the **Display** section, select **Eigenvalues plots** and check all of the display checkboxes so that EViews will display all three of the eigenvalue plots. Next, click on the **Calculation** tab and click on the **Between periods covariances** button so that EViews will unstack the data into different periods.

It is important to note that you must change the sample to "1973 1992" since data for DCPCH in 1972 are not available (DCPCH is a lagged difference). If you were to use the original sample of "1972 1992", the balanced sample option would remove all observations and the resulting between period correlation matrix would contain only NAs. Principal components analysis on this matrix would fail.

Click on **OK** to accept the settings. The results of this view (after rearranging the graphs slightly) are depicted below:

Panel Causality Testing

EViews offers panel specific forms of Granger causality tests ("Granger Causality" on page 661) of *User's Guide I*.

In panel workfile settings, EViews performs panel data specific causality testing. In these settings, least squares regressions can take a number of different forms, depending upon assumptions made about the structure of the panel data. Since Granger Causality is computed by running bivariate regressions, there are a number of different approaches to testing for Granger Causality in a panel context.

In general, the bivariate regressions in a panel data context take the form:

$$y_{i,\,t} = \alpha_{0,\,i} + \alpha_{1,\,i}\, y_{i,\,t-1} + \ldots + \alpha_{k,\,i}\, y_{i,\,t-k} + \beta_{1,\,i}\, x_{i,\,t-1} + \ldots + \beta_{k,\,i}\, x_{i,\,t-k} + \epsilon_{i,\,t} \tag{54.5}$$

$$x_{i,\,t} = \alpha_{0,\,i} + \alpha_{1,\,i}\, x_{i,\,t-1} + \ldots + \alpha_{k,\,i}\, x_{i,\,t-k} + \beta_{1,\,i}\, y_{i,\,t-1} + \ldots + \beta_{k,\,i}\, y_{i,\,t-k} + \epsilon_{i,\,t} \tag{54.6}$$

Where t denotes the time period dimension of the panel, and i denotes the cross-sectional dimension.

The different forms of panel causality test differ on the assumptions made about the homogeneity of the coefficients across cross-sections.

EViews offers two of the simplest approaches to causality testing in panels. The first is to treat the panel data as one large stacked set of data, and then perform the Granger Causality test in the standard way, with the exception of not letting data from one cross-section enter the lagged values of data from the next cross-section. This method assumes that all coefficients are same across all cross-sections, i.e.:

$$\alpha_{0,\,i} = \alpha_{0,\,j},\ \alpha_{1,\,i} = \alpha_{1,\,j},\ \ldots,\ \alpha_{l,\,i} = \alpha_{l,\,j},\ \forall i, j \tag{54.7}$$

$$\beta_{1,\,i} = \beta_{1,\,j},\ \ldots,\ \beta_{l,\,i} = \beta_{lj} \forall i, j \tag{54.8}$$

A second approach adopted by Dumitrescu-Hurlin (2012), makes an extreme opposite assumption, allowing all coefficients to be different across cross-sections:

$$\alpha_{0,\,i} \neq \alpha_{0,\,j},\ \alpha_{1,\,i} \neq \alpha_{1,\,j},\ \ldots,\ \alpha_{l,\,i} \neq \alpha_{l,\,j},\ \forall i, j \tag{54.9}$$

$$\beta_{1,\,i} \neq \beta_{1,\,j},\ \ldots,\ \beta_{l,\,i} \neq \beta_{lj} \forall i, j \tag{54.10}$$

This test is calculated by simply running standard Granger Causality regressions for each cross-section individually. The next step is to take the average of the test statistics, which are termed the $\bar{Wbar}$ statistic. They show that the standardized version of this statistic, appropriately weighted in unbalanced panels, follows a standard normal distribution. This is termed the $\bar{Zbar}$ statistic.

(EViews does not provide built-in versions of other panel-causality tests since they are often based upon regressions using some assumptions on Equation (54.5), or in some cases two-stage least squares regressions, often using a fixed or a random effects model. It is possible to perform these test by estimating the models using an EViews equation object and then perform Wald test coefficient restrictions on the appropriate coefficients.)

To perform the test, create a group containing the series of interest, then select **View/ Granger Causality...** to display the test dialog:

Select the **Test type** using the radio buttons and provide a number of **Lags to include**. Click on **OK** to accept the settings and compute the test.

Pairwise Dumitrescu Hurlin Panel Causality Tests
Date: 02/05/13 Time: 15:58
Sample: 1960 1978
Lags: 2

Null Hypothesis:	W-Stat.	Zbar-Stat.	Prob.
LGASPCAR does not homogeneously cause LCARPCAP	2.81017	0.59203	0.5538
LCARPCAP does not homogeneously cause LGASPCAR	4.59763	3.17200	0.0015

Here we show results for the pairwise Dumitrescu-Hurlin tests using data from "gasoline.WF1" (which is available in your examples directory). We reject the null that LCARPCAP does not homogeneously cause LGASPCAR, but do not direct in the opposite direction.

Panel Long-run Variances

The computation of long-run covariances is described in great detail in Appendix F, "Long-run Covariance Estimation" on page 1339 of *User's Guide I*. The group view for computing the covariances is documented in "Long-run Covariance" on page 655 of *User's Guide I*. In panel workfiles, EViews computes the Phillips and Moon (1999) *long-run average covariance matrix* obtained by averaging the long-run covariances across cross-sections.

There is little difference between the settings for long-run covariances and variances in non-panel and panel settings. You may, however, provide a name in the **Panel matrix** edit field to EViews to save a matrix containing the individual covariance estimates. Each row will contain the vec or vech of the results matrix for the corresponding cross-section.

Suppose, for example, that we create a group using the LCARPCAP and LGASPCAR series from the "gasoline.WF1" workfile. Select **View/Long-run Covariance...** to display the dialog, enter "pan_results" in the **Panel matrix** edit field, and leave the remaining settings at their defaults:

The resulting long-run average covariances are shown in the group window:

and the individual cross-section results are stored in the matrix PAN_RESULTS, with the vech of the individual cross-section covariances stored in each row:

	C1	C2	C3
R1	0.537980	-0.029571	0.008358
R2	0.424434	-0.087366	0.022836
R3	0.093846	0.003435	0.001313
R4	0.290093	-0.130620	0.059394
R5	0.282459	0.000789	0.004159
R6	0.390378	0.012386	0.001170
R7	1.727992	-0.510083	0.156401
R8	0.286481	0.013635	0.003882
R9	0.983011	-0.328386	0.118235
R10	3.513741	-2.011752	1.152877
R11	0.935524	-0.430496	0.200284
R12	0.454496	-0.117188	0.032589
R13	2.266039	-0.717614	0.232317
R14	0.141264	-0.009397	0.001772
R15	0.344595	-0.080767	0.022269
R16	1.380315	-0.576724	0.249664
R17	0.191708	-0.022287	0.004425
R18	0.064955	0.002941	0.001001

Panel Unit Root Testing

EViews provides convenient tools for computing panel unit root tests both under the assumption of cross-section independence and allowing for cross-section dependence.

EViews computes five different independent cross-section 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).

EViews currently supports two important dependent cross-section tests: Bai and Ng's (2004) Panel Analysis of Nonstationarity in Idiosyncratic and Common Components (PANIC), and Pesaran's (2007) Cross-sectionally Augmented IPS (CIPS).

These tests are described in detail in "Cross-sectionally Independent Panel Unit Root Testing," beginning on page 705 and "Cross-sectionally Dependent Panel Unit Root Tests," beginning on page 716.

To compute the independent unit root test on a series, simply select **View/Unit Root Tests/ Cross Sectionally Independent…** 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 dropdown menu 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 several 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 asympotic 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			

To compute a dependent unit root test on a series, simply select **View/Unit Root Tests/ Cross Sectionally Dependent…** from the menu of a series object:

For the default PANIC options, we have the results

The results are displayed in a spool. For a discussion of the results, additional examples, and discussion, see "Cross-sectionally Dependent Panel Unit Root Tests," beginning on page 716.

Panel Cointegration Testing

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 Details," beginning on page 1260.

To compute a panel cointegration test, select **View/Cointegration Test/Panel 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)

	Statistic	Prob.	Weighted Statistic	Prob.
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)

	Statistic	Prob.
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

Panel Cross-section Dependence Test

You may test for cross-section dependence in a series in a panel structured workfile. There are a variety of tests for cross-section dependence in the literature, and EViews offers the following tests:

- Breusch-Pagan (1980) LM
- Pesaran (2004) scaled LM
- Baltagi, Feng, and Kao (2012) bias-corrected scaled LM
- Pesaran (2004) CD

See "Panel Cross-section Dependence Test" on page 1177 for discussion.

Panel Resampling

Resample... performs resampling on all of the series in the group. A description of the resampling procedure is provided in "Resample" on page 482 of *User's Guide I*. When you resample from a panel workfile, EViews offers you an additional option of whether to resample across cross-sections or not. The default assumes that cross-sections are not identical so that the resampling is not done across cross-sections, but is instead performed on a cross-section by cross-section basis.

Panel 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.

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 ARCH or state space models with the stacked panel.

References

Dumitrescu, Elena-Ivona and Christophe Hurlin (2012). "Testing for Granger Non-causality in Heterogeneous Panels," *Economic Modeling*, 29, 1450-1460.

Phillips, Peter C. B. and Hyungsik R. Moon (1999). "Linear Regression Limit Theory for Nonstationary Panel Data," *Econometrica*, 67, 1057-1111.

Part X. Advanced Multivariate Analysis

The following chapters describe specialized tools for multivariate analysis:

- Chapter 55. "Cointegration Testing," on page 1245 documents testing for the presence of cointegrating relationships among non-stationary variables in non-panel and panel settings.
- Chapter 56. "Factor Analysis," on page 1265 describes tools for multivariate analysis using factor analysis.

General tools for multivariate analysis using the group object, including summary statistics, covariance analysis and principal components, are discussed in Chapter 12. "Groups," beginning on page 591 of *User's Guide I*.

Chapter 55. 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.

This chapter describes several tools for testing for the presence of cointegrating relationships among non-stationary variables in non-panel and panel settings.

The first two parts of this chapter focus on cointegration tests employing the Johansen (1991, 1995) system framework or Engle-Granger (1987) or Phillips-Ouliaris (1990) residual based test statistics. The final section describes cointegration tests in panel settings where you may compute the Pedroni (1999), Pedroni (2004), and Kao (1999) tests as well as a Fisher-type test using an underlying Johansen methodology (Maddala and Wu, 1999).

The Johansen tests may be performed using a Group object or an estimated Var object. The residual tests may be computed using a Group object or an Equation object estimated using nonstationary regression methods. The panel tests may be conducted using a Pool object or a Group object in a panel workfile setting. Note that additional cointegration tests are offered as part of the diagnostics for an equation estimated using nonstationary methods. See "Testing for Cointegration" on page 308.

If cointegration is detected, Vector Error Correction (VEC) or nonstationary regression methods may be used to estimate the cointegrating equation. See "Vector Error Correction (VEC) Models" on page 881 and Chapter 28. "Cointegrating Regression," beginning on page 293 for details.

Johansen Cointegration Test

EViews supports VAR-based cointegration tests using the methodology developed in Johansen (1991, 1995) performed using a Group object or an estimated Var object.

Consider a VAR of order p:

$$y_t = A_1 y_{t-1} + \cdots + A_p y_{t-p} + B x_t + \epsilon_t \tag{55.1}$$

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,

$$\Delta y_t = \Pi y_{t-1} + \sum_{i=1}^{p-1} \Gamma_i \Delta y_{t-i} + B x_t + \epsilon_t \tag{55.2}$$

where:

$$\Pi = \sum_{i=1}^{p} A_i - I, \qquad \Gamma_i = -\sum_{j=i+1}^{p} A_j \tag{55.3}$$

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 Johansen Cointegration Test

To carry out the Johansen cointegration test, select **View/Cointegration Test/Johansen System Cointegration Test...** from a group window or **View/Cointegration Test...** from a Var object window. The **Cointegration Test Specification** page prompts you for information about the test.

The dialog will differ slightly depending on whether you are using a group or an estimated Var object to perform your test. We show here the group dialog; the Var dialog has an additional page as described in "Imposing Restrictions" on page 1252.

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 Testing" on page 667 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.

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.

We may summarize the five deterministic trend cases considered by Johansen (1995, p. 80–84) as:

1. The level data y_t have no deterministic trends and the cointegrating equations do not have intercepts:
$$H_2(r)\colon \Pi y_{t-1} + Bx_t = \alpha\beta' y_{t-1}$$
2. The level data y_t have no deterministic trends and the cointegrating equations have intercepts:
$$H_1^*(r)\colon \Pi y_{t-1} + Bx_t = \alpha(\beta' y_{t-1} + \rho_0)$$
3. The level data y_t have linear trends but the cointegrating equations have only intercepts:
$$H_1(r)\colon \Pi y_{t-1} + Bx_t = \alpha(\beta' y_{t-1} + \rho_0) + \alpha_\perp \gamma_0$$
4. The level data y_t and the cointegrating equations have linear trends:
$$H^*(r)\colon \Pi y_{t-1} + Bx_t = \alpha(\beta' y_{t-1} + \rho_0 + \rho_1 t) + \alpha_\perp \gamma_0$$
5. The level data y_t have quadratic trends and the cointegrating equations have linear trends:
$$H(r)\colon \Pi y_{t-1} + Bx_t = \alpha(\beta' y_{t-1} + \rho_0 + \rho_1 t) + \alpha_\perp(\gamma_0 + \gamma_1 t)$$

The terms associated with $\alpha_\perp$ 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 $\alpha_\perp$ 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 Johansen Cointegration Test

As an example, the header portion of the cointegration test output for the four-variable system used by Johansen and Juselius (1990) for the Danish data is shown below.

```
Date: 09/21/09   Time: 11:12
Sample (adjusted): 1974Q3 1987Q3
Included observations: 53 after adjustments
Trend assumption: No deterministic trend (restricted constant)
Series: LRM LRY IBO IDE
Exogenous series: D1 D2 D3
Warning: Critical values assume no exogenous series
Lags interval (in first differences): 1 to 1
```

As indicated in the header of the output, the test assumes no trend in the series with a restricted intercept in the cointegration relation (We computed the test using assumption 2 in the dialog, **Intercept (no trend) in CE - no intercept in VAR**), 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 next portion 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 (55.3), the third column is the test statistic, the fourth column is the 5% critical value, and the last column is the p-value. The (nonstandard distribution) critical values are taken from MacKinnon-Haug-Michelis (1999) so they differ slightly from those reported in Johansen and Juselius (1990).

Unrestricted Cointegration Rank Test (Trace)

Hypothesized No. of CE(s)	Eigenvalue	Trace Statistic	0.05 Critical Value	Prob.**
None	0.433165	49.14436	54.07904	0.1282
At most 1	0.177584	19.05691	35.19275	0.7836
At most 2	0.112791	8.694964	20.26184	0.7644
At most 3	0.043411	2.352233	9.164546	0.7071

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

Unrestricted Cointegration Rank Test (Maximum Eigenvalue)

Hypothesized No. of CE(s)	Eigenvalue	Max-Eigen Statistic	0.05 Critical Value	Prob.**
None *	0.433165	30.08745	28.58808	0.0319
At most 1	0.177584	10.36195	22.29962	0.8059
At most 2	0.112791	6.342731	15.89210	0.7486
At most 3	0.043411	2.352233	9.164546	0.7071

Max-eigenvalue test indicates 1 cointegrating eqn(s) at the 0.05 level
* denotes rejection of the hypothesis at the 0.05 level
**MacKinnon-Haug-Michelis (1999) p-values

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 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 $r = 0, 1, \ldots, k-1$. 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:

$$LR_{\text{tr}}(r|k) = -T \sum_{i=r+1}^{k} \log(1-\lambda_i) \tag{55.4}$$

where λ_i is the i-th largest eigenvalue of the Π matrix in (55.3) 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:

$$\begin{aligned} LR_{\max}(r|r+1) &= -T\log(1-\lambda_{r+1}) \\ &= LR_{\text{tr}}(r|k) - LR_{\text{tr}}(r+1|k) \end{aligned} \tag{55.5}$$

for $r = 0, 1, \ldots, k-1$.

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.

Unrestricted Cointegrating Coefficients (normalized by b*S11*b=I):

LRM	LRY	IBO	IDE	C
-21.97409	22.69811	-114.4173	92.64010	133.1615
14.65598	-20.05089	3.561148	100.2632	-62.59345
7.946552	-25.64080	4.277513	-44.87727	62.74888
1.024493	-1.929761	24.99712	-14.64825	-2.318655

Unrestricted Adjustment Coefficients (alpha):

D(LRM)	0.009691	-0.000329	0.004406	0.001980
D(LRY)	-0.005234	0.001348	0.006284	0.001082
D(IBO)	-0.001055	-0.000723	0.000438	-0.001536
D(IDE)	-0.001338	-0.002063	-0.000354	-4.65E-05

The remaining blocks report estimates from a different normalization for each possible number of cointegrating relations $r = 0, 1, \ldots, k-1$. 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.

In our example, for one cointegrating equation we have:

1 Cointegrating Equation(s): Log likelihood 669.1154

Normalized cointegrating coefficients (standard error in parentheses)

LRM	LRY	IBO	IDE	C
1.000000	-1.032949	5.206919	-4.215880	-6.059932
	(0.13897)	(0.55060)	(1.09082)	(0.86239)

Adjustment coefficients (standard error in parentheses)

D(LRM)	-0.212955
	(0.06435)
D(LRY)	0.115022
	(0.06739)
D(IBO)	0.023177
	(0.02547)
D(IDE)	0.029411
	(0.01717)

Imposing Restrictions

Since the cointegrating vector β is not fully identified, you may wish to impose your own identifying restrictions. *If you are performing your Johansen cointegration test using an estimated Var object*, EViews offers you the opportunity to impose restrictions on β. 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 perform the cointegration test from a Var object, you will first need to estimate a VAR with your variables as described in "Estimating a VAR in EViews" on page 835. Next, select **View/Cointegration Test...** from the Var menu 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:

A full description of how to enter your restrictions is provided in "Imposing VEC Restrictions" on page 884.

Results of Restricted Cointegration Test

If you impose restrictions in the **Cointegration Test** view, the top portion of the output will display the unrestricted test results 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-likehood	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 $r = 2, 3, \ldots, k-1$ (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.

Single-Equation Cointegration Tests

You may use a group or an equation object estimated using `cointreg` to perform Engle and Granger (1987) or Phillips and Ouliaris (1990) single-equation residual-based cointegration tests. A description of the single-equation model underlying these tests is provided in "Background" on page 293. Details on the computation of the tests and the associated options may be found in "Residual-based Tests," on page 308.

Briefly, the Engle-Granger and Phillips-Ouliaris residual-based tests for cointegration are simply unit root tests applied to the residuals obtained from a static OLS cointegrating regression. Under the assumption that the series are *not* cointegrated, the residuals are unit root nonstationary. Therefore, a test of the *null hypothesis of no cointegration* against the *alternative of cointegration* may be constructed by computing a unit root test of the null of residual nonstationarity against the alternative of residual stationarity.

How to Perform a Residual-Based Cointegration Test

We illustrate the single-equation cointegration tests using Hamilton's (1994) purchasing power parity example (p. 598) for the log of the U.S. price level (P_T), log of the Dollar-Lira exchange rate (S_T), and the log of the Italian price level (PSTAR_T) from 1973m1 to 1989m10. We will use these data, which are provided in "Hamilton_rates.WF1", to construct Engle-Granger and Phillips-Ouliaris tests assuming the constant is the only deterministic regressor in the cointegrating equation.

To carry out the Engle-Granger of Phillips-Ouliaris cointegration tests, first create a group, say G1, containing the series P_T, S_T, and PSTAR_T, then select **View/Cointegration Test/**

Single-Equation Cointegration Test from the group toolbar or main menu. The **Cointegration Test Specification** page opens to prompt you for information about the test.

The dropdown menu at the top allows you to choose between the default **Engle-Granger** test or the **Phillips-Ouliaris** test. Below the dropdown are the options for the test statistic. The Engle-Granger test requires a specification for the number of lagged differences to include in the test regression, and whether to d.f. adjust the standard error estimate when forming the ADF test statistics. To match Hamilton's example, we specify a **Fixed (User-specified)** lag specification of 12, and retain the default d.f. correction of the standard error estimate.

The right-side of the dialog is used to specify the form of the cointegrating equation. The main cointegrating equation is described in the **Equation specification** section. You should use the **Trend specification** dropdown to choose from the list of pre-specified deterministic trend variable assumptions (**None**, **Constant (Level)**, **Linear Trend**, **Quadratic Trend**). If you wish to include deterministic regressors that are not offered in the pre-specified list, you may enter the series names or expressions in the **Deterministic regressors** edit box. For our example, we will leave the settings at their default values, with the **Trend specification** set to **Constant (Level)**, and no additional deterministic regressors specified.

The **Regressors specification** section should be used to specify any deterministic trends or other regressors that should be included in the regressors equations but not in the cointegrating equation. In our example, Hamilton points to evidence of non-zero drift in the regressors, so we will select **Linear trend** in the **Additional trends** dropdown.

Click on **OK** to compute and display the test results.

Date: 05/11/09 Time: 15:52
Series: P_T S_T PSTAR_T
Sample: 1973M01 1989M10
Included observations: 202
Null hypothesis: Series are not cointegrated
Cointegrating equation deterministics: C
Additional regressor deterministics: @TREND
Fixed lag specification (lag=12)

Dependent	tau-statistic	Prob.*	z-statistic	Prob.*
P_T	-2.730940	0.4021	-26.42791	0.0479
S_T	-2.069678	0.7444	-13.83563	0.4088
PSTAR_T	-2.631078	0.4548	-22.75737	0.0962

*MacKinnon (1996) p-values.

Intermediate Results:

	P_T	S_T	PSTAR_T
Rho - 1	-0.030478	-0.030082	-0.031846
Rho S.E.	0.011160	0.014535	0.012104
Residual variance	0.114656	5.934605	0.468376
Long-run residual variance	2.413438	35.14397	6.695884
Number of lags	12	12	12
Number of observations	189	189	189
Number of stochastic trends**	2	2	2

**Number of stochastic trends in asymptotic distribution

The top two portions of the output describe the test setup and summarize the test results. Regarding the test results, note that EViews computes both the Engle-Granger tau-statistic (t-statistic) and normalized autocorrelation coefficient (which we term the z-statistic) for residuals obtained using each series in the group as the dependent variable in a cointegrating regression. Here we see that the test results are broadly similar for different dependent variables, with the tau-statistic uniformly failing to reject the null of no cointegration at conventional levels. The results for the z-statistics are mixed, with the residuals from the P_T equation rejecting the unit root null at the 5% level. On balance, however, the test statistics suggest that we cannot reject the null hypothesis of no cointegration.

The bottom portion of the results show intermediate calculations for the test corresponding to each dependent variable. "Residual-based Tests," on page 308 offers a discussion of these statistics. We do note that there are only 2 stochastic trends in the asymptotic distribution (instead of the 3 corresponding to the number of variables in the group) as a result of our assumption of a non-zero drift in the regressors.

Alternately, you may compute the Phillips-Ouliaris test statistic. Once again select **View/ Cointegration Test/Single-Equation Cointegration Test** from the Group toolbar or main menu, but this time choose **Phillips-Ouliaris** in the **Test Method** dropdown.

The right-hand side of the dialog, which describes the cointegrating regression and regressors specifications, should be specified as before.

The left-hand side of the dialog changes to show a single **Options** button for controlling the estimation of the **Long-run variance** used in the Phillips-Ouliaris test, and the checkbox for **d.f Adjustment** of the variance estimates. The default settings instruct EViews to compute these long-run variances using a non-prewhitened Bartlett kernel estimator with a fixed Newey-West bandwidth. We match the Hamilton example settings by turning off the d.f. adjustment and by clicking on the **Options** button and using the **Bandwidth method** dropdown to specify a **User-specified** bandwidth value of 13.

Click on the **OK** button to accept the **Options**, then on **OK** again to compute the test statistics and display the results:

Date: 05/11/09 Time: 16:01
Series: P_T S_T PSTAR_T
Sample: 1973M01 1989M10
Included observations: 202
Null hypothesis: Series are not cointegrated
Cointegrating equation deterministics: C
Additional regressor deterministics: @TREND
Long-run variance estimate (Bartlett kernel, User bandwidth = 13.0000)
No d.f. adjustment for variances

Dependent	tau-statistic	Prob.*	z-statistic	Prob.*
P_T	-2.023222	0.7645	-7.542281	0.8039
S_T	-1.723248	0.8710	-6.457868	0.8638
PSTAR_T	-1.997466	0.7753	-7.474681	0.8078

*MacKinnon (1996) p-values.

Intermediate Results:

	P_T	S_T	PSTAR_T
Rho - 1	-0.016689	-0.014395	-0.017550
Bias corrected Rho - 1 (Rho* - 1)	-0.037524	-0.032129	-0.037187
Rho* S.E.	0.018547	0.018644	0.018617
Residual variance	0.162192	6.411674	0.619376
Long-run residual variance	0.408224	13.02214	1.419722
Long-run residual autocovariance	0.123016	3.305234	0.400173
Bandwidth	13.00000	13.00000	13.00000
Number of observations	201	201	201
Number of stochastic trends**	2	2	2

**Number of stochastic trends in asymptotic distribution

In contrast with the Engle-Granger tests, the results are quite similar for all six of the tests with the Phillips-Ouliaris test not rejecting the null hypothesis that the series are not cointegrated. As before, the bottom portion of the output displays intermediate results for the test associated with each dependent variable.

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; conducting a cointegration test using a Pool

involves only minor differences in specification (see "Performing Cointegration Tests," beginning on page 1078 for a discussion of testing in the pooled data setting).

To perform the panel cointegration test using a Group object you should first make certain you are in a panel structured workfile (Chapter 51. "Working with Panel Data," on page 1111). If you have a panel workfile with a single cross-section in the sample, you may perform one of the standard single-equation cointegration tests using your subsample.

Next, open an EViews group containing the series of interest, and select **Views/Cointegration Test/Panel Cointegration Test…** to display the cointegration dialog.

The dropdown menu 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 ("Cross-sectionally Independent Panel Unit Root Testing" on page 705).

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 dropdown menu (**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:

$$\mathrm{int}(\min((T_i - k)/3, 12) \cdot (T_i/100)^{1/4})$$

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.

The Pedroni test employs both parametric and non-parametric kernel estimation of the long run variance. You may use the **Variance calculation** and **Lag length** sections to control the computation of the parametric variance estimators. The **Spectral estimation** portion of the dialog allows you to specify settings for the non-parametric 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**, **Newey-West fixed**, **User specified**). The Newey-West fixed bandwidth is given by $4(T_i/100)^{2/9}$. The Kao test uses the **Lag length** and the **Spectral estimation** portion of the dialog settings as described below.

Here, we see the options for the Fisher test selection. These options are similar to the options available in the Johansen cointegration test ("Johansen Cointegration Test," beginning on page 1245).

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.

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

$$y_{it} = \alpha_i + \delta_i t + \beta_{1i} x_{1i,t} + \beta_{2i} x_{2i,t} + \ldots + \beta_{Mi} x_{Mi,t} + e_{i,t} \tag{55.6}$$

for $t = 1, \ldots, T$; $i = 1, \ldots, N$; $m = 1, \ldots, M$; 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 (55.6) and then to test whether residuals are I(1) by running the auxiliary regression,

$$e_{it} = \rho_i e_{it-1} + u_{it} \tag{55.7}$$

or

$$e_{it} = \rho_i e_{it-1} + \sum_{j=1}^{p_i} \psi_{ij} \Delta e_{it-j} + v_{it} \tag{55.8}$$

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 $\aleph_{N,T}$ is constructed from the residuals from either Equation (55.7) or Equation (55.8). 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,

$$\frac{\aleph_{N,T} - \mu\sqrt{N}}{\sqrt{v}} \Rightarrow N(0, 1) \tag{55.9}$$

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

$$y_{it} = \alpha_i + \beta x_{it} + e_{it} \tag{55.10}$$

for

$$y_{it} = y_{it-1} + u_{i,t} \tag{55.11}$$

$$x_{it} = x_{it-1} + \epsilon_{i,t} \tag{55.12}$$

for $t = 1, \ldots, T$; $i = 1, \ldots, N$. More generally, we may consider running the first stage regression Equation (55.6), 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,

$$e_{it} = \rho e_{it-1} + v_{it} \tag{55.13}$$

or the augmented version of the pooled specification,

$$e_{it} = \tilde{\rho} e_{it-1} + \sum_{j=1}^{p} \psi_j \Delta e_{it-j} + v_{it} \tag{55.14}$$

Under the null of no cointegration, Kao shows that following the statistics,

$$DF_\rho = \frac{T\sqrt{N}(\hat{\rho} - 1) + 3\sqrt{N}}{\sqrt{10.2}} \tag{55.15}$$

$$DF_t = \sqrt{1.25}\, t_\rho + \sqrt{1.875 N} \tag{55.16}$$

$$DF_\rho^* = \frac{\sqrt{N}T(\hat{\rho} - 1) + 3\sqrt{N}\hat{\sigma}_v^2/\hat{\sigma}_{0v}^2}{\sqrt{3 + 36\hat{\sigma}_v^4/(5\hat{\sigma}_{0v}^4)}} \tag{55.17}$$

$$DF_t^* = \frac{t_\rho + \sqrt{6N}\hat{\sigma}_v/(2\hat{\sigma}_{0v})}{\sqrt{\hat{\sigma}_{0v}^2/(2\hat{\sigma}_v^2) + 3\hat{\sigma}_v^2/(10\hat{\sigma}_{0v}^2)}} \tag{55.18}$$

and for $p > 0$ (*i.e.* the augmented version),

$$ADF = \frac{t_{\tilde{\rho}} + \sqrt{6N}\hat{\sigma}_v/(2\hat{\sigma}_{0v})}{\sqrt{\hat{\sigma}_{0v}^2/(2\hat{\sigma}_v^2) + 3\hat{\sigma}_v^2/(10\hat{\sigma}_{0v}^2)}} \tag{55.19}$$

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

$$w_{it} = \begin{bmatrix} u_{it} \\ \epsilon_{it} \end{bmatrix} \tag{55.20}$$

is estimated as

$$\hat{\Sigma} = \begin{bmatrix} \hat{\sigma}_u^2 & \hat{\sigma}_{u\epsilon} \\ \hat{\sigma}_{u\epsilon} & \hat{\sigma}_\epsilon^2 \end{bmatrix} = \frac{1}{NT} \sum_{i=1}^{N} \sum_{t=1}^{T} \hat{w}_{it} \hat{w}_{it}' \tag{55.21}$$

and the long run covariance is estimated using the usual kernel estimator

$$\begin{aligned} \hat{\Omega} &= \begin{bmatrix} \hat{\sigma}_{0u}^2 & \hat{\sigma}_{0u\epsilon} \\ \hat{\sigma}_{0u\epsilon} & \hat{\sigma}_{0\epsilon}^2 \end{bmatrix} \\ &= \frac{1}{N} \sum_{i=1}^{N} \left[\frac{1}{T} \sum_{t=1}^{T} \hat{w}_{it} \hat{w}_{it}' + \frac{1}{T} \sum_{\tau=1}^{\infty} \kappa(\tau/b) \sum_{t=\tau+1}^{T} (\hat{w}_{it} \hat{w}_{it-\tau}' + \hat{w}_{it-\tau} \hat{w}_{it}') \right] \end{aligned} \tag{55.22}$$

where κ is one of the supported kernel functions and b is the bandwidth.

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 at 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,

$$-2 \sum_{i=1}^{N} \log(\pi_i) \rightarrow \chi^2_{2N} \tag{55.23}$$

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.

References

Boswijk, H. Peter (1995). "Identifiability of Cointegrated Systems," Technical Report, Tinbergen Institute.

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.

Hamilton, James D. (1994). *Time Series Analysis*, Princeton: Princeton University Press.

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 Statistic*s, 52, 169–210.

Kao, Chinwa D. (1999). "Spurious Regression and Residual-Based Tests for Cointegration in Panel Data," *Journal of Econometrics*, 90, 1–44.

MacKinnon, James G. (1996). "Numerical Distribution Functions for Unit Root and Cointegration Tests," *Journal of Applied Econometrics*, 11, 601-618.

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.

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.

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.

Chapter 56. 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 1298.

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** dropdown menu, 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 616.

Method

You may use the **Method** dropdown 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 1309) 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

User-matrix in the Type dropdown, 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.

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** dropdown menu, 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 1299 for a brief summary of the various methods.

You may change the default by selecting an alternative method from the dropdown menu. 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 dropdown menu, the dialog page will change to provide you with a number of additional options.

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** dropdown 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.

- If you select either **Bai and Ng** or **Ahn and Horenstein** from the dropdown menu, the dialog page will offer different options.

The dialog settings are described in detail in "Bai and Ng" on page 652 and "Ahn and Horenstein" on page 653 of *User's Guide I*.

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 1302).

By default, EViews will compute SMC based estimates of the communalities. You may select a different method using the **Initial communalities** dropdown menu. Most of the methods should be self-explanatory; a few require additional comment.

- **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 1302 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** dropdowns may be used to specify the basic rotation method (see "Types of Rotation," on page 1306 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** dropdown menu 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** dropdown. 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** dropdown, the 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 1309. 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** (Thur-

stone'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 coefs** 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 642).

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 1303.

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:

$$A = \mathrm{diag}(S^{-1})^{-1} S^{-1} \mathrm{diag}(S^{-1})^{-1}$$

- **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 1272 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 .

Factor Selection

If you estimated your factor model using the Bai and Ng or the Ahn and Horenstein number of factor selection method, you may use this view to display information related to the selection procedure.

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 1274.

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** dropdown.
- 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 1302). 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 1266.
- **Rotate...** is used to perform factor rotation using the **Factor Rotation** dialog. See "Rotating Factors" on page 1273.
- **Make Scores...** is used to save estimated factor scores as series in the workfile. See "Estimating Scores" on page 1274.
- **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 218 in the *Command and Programming 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 the factor object 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 pre-filled settings. Here, we see that EViews has entered in the names of the seven series in G7.

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 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			
	F1	F2	Communality	Uniqueness
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 variance accounted for by the two factors is 3.55, which is close to 51% $(3.55/7.0)$ of the total variance (sum of the diagonals of the correlation matrix). Furthermore, we see that the first factor F1 accounts for 77% $(2.72/3.55)$ of the *common* variance and the second factor F2 accounts for the remaining 23% $(0.82/3.55)$.

Factor	Variance	Cumulative	Difference	Proportion	Cumulative
F1	2.719663	2.719663	1.892380	0.766762	0.766762
F2	0.827282	3.546945	---	0.233238	1.000000
Total	3.546945	3.546945		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 Noncentralilty	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 1303 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: HOLZINGER24::Untitled\

View | Proc | Object | Print | Name | Freeze | Estimate | Rotate | Score | Stats

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
PARAG...	-0.007661	0.017623	-1.14E-12	0.000427	0.001141	0.023294	-0.018990
SENTE...	-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 (near) zero since we have subtracted off the total fitted covariance (which includes the uniquenesses). To replace the (near) 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

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** dropdown 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**. Set the random generator options as depicted and 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: FACTOR01
Date: 10/16/09 Time: 11:12
Initial loadings: Oblique Random (reps=25, rng=kn, seed=1000)
Results obtained from random draw 1 of 25
Failure to improve after 18 iterations

Rotated loadings: L * 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: inv(T)'

	F1	F2
F1	0.875271	0.207044
F2	-0.786498	1.158366

Initial rotation objective: 0.288147
Final rotation objective: 0.010096

Note that in the absence of orthogonality, the factor rotation 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 oblique Quartimax 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.

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

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:

Indeterminancy 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:

	Validity
VERBAL	0.940103
SPATIAL	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:

$$X_i - \mu = LF_i + \epsilon_i \tag{56.1}$$

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:

$$\begin{aligned} var(X) &= E[(X_i - \mu)(X_i - \mu)'] \\ &= E[(LF_i + \epsilon_i)(LF_i + \epsilon_i)'] \\ &= L\Phi L' + \Psi \end{aligned} \tag{56.2}$$

The variances of the individual variables may be decomposed into:

$$\sigma_{jj} = h_j^2 + \psi_j \tag{56.3}$$

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:

$$\begin{aligned} var(X, F) &= E[(X_i - \mu)F_i'] \\ &= E[(LF_i + \epsilon_i)F_i'] \\ &= L\Phi \end{aligned} \tag{56.4}$$

Initially, we make the further assumption that the factors are orthogonal so that $\Phi = I$ (we will relax this assumption shortly). Then:

$$\begin{aligned} var(X) &= LL' + \Psi \\ var(X, F) &= L \end{aligned} \tag{56.5}$$

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 are 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 partialed out (for $m = 0, \ldots, p-1$). The number of factor 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).

Bai and Ng

Bai and Ng (2002) propose a model selection approach to determining the number of factors in a principal components framework. The technique involves least squares regression using

different numbers of eigenvalues obtained from a principal components decomposition. See "Bai and Ng" on page 652 of *User's Guide I* for details.

Ahn and Horenstein

Ahn and Horenstein (AH, 2013) provide a method for obtaining the number of factors that exploits the fact that the r largest eigenvalues of a given matrix grow without bounds as the rank of the matrix increases, whereas the other eigenvalues remain bounded. The optimization strategy is then simply to find the maximum of the ratio of two adjacent eigenvalues. See "Ahn and Horenstein" on page 653 of *User's Guide I* for discussion.

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:

$$\begin{aligned} D_{ML}(S, \Sigma) &= tr\left|\Sigma^{-1}S\right| - \ln\left|\Sigma^{-1}S\right| - p \\ D_{GLS}(S, \Sigma) &= \mathrm{tr}([I_p - S^{-1}\Sigma]^2)/2 \\ D_{ULS}(S, \Sigma) &= \mathrm{tr}([S - \Sigma]^2)/2 \end{aligned} \tag{56.6}$$

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 $\mathbf{\Psi}_0$, we may define the reduced dispersion matrix $S_R(\mathbf{\Psi}_0) = S - \mathbf{\Psi}_0$, 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 be 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 absolution 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 partioning.

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 (56.6). 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 of 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:

$$\begin{aligned} AIC &= (N-k)D/N-(2/N)df \\ SC &= (N-k)D/N-(\ln(N)/N)df \\ HQ &= (N-k)D/N-(2\ln(\ln(N))/N)df \end{aligned} \tag{56.7}$$

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:

$$X_i - \mu = LF_i + \epsilon_i \tag{56.8}$$

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 (56.1) as:

$$X_i - \mu = L(T^{-1})' T' F_i + \epsilon_i = \tilde{L}\tilde{F}_i + \epsilon_i \tag{56.9}$$

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:

$$E(\tilde{F}_i \tilde{F}_i') = T'T = \Phi \tag{56.10}$$

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:

$$f(L) = (1-\kappa)\sum_{i=1}^{p}\left(\sum_{j=1}^{m}\sum_{k \neq j}^{m}\lambda_{ij}\lambda_{ik}\right) + \kappa\sum_{j=1}^{m}\left(\sum_{i=1}^{p}\sum_{p \neq i}^{p}\lambda_{ij}\lambda_{pj}\right) \tag{56.11}$$

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 (*default* = 0, Quartimax).
Generalized Crawford-Ferguson	4	Vector of weights for (in order): total squares, variable complexity, factor complexity, diagonal quartics (*no default*).
Geomin	1	Epsilon offset (*default* = 0.01).
Harris-Kaiser (case II)	2	Power parameter (*default* = 0, independent cluster solution).
Oblimin	1	Deviation from orthogonality (*default* = 0, Quartimin).
Orthomax	1	Factor complexity weight (*default* = 1, Varimax).
Promax	1	Power parameter (*default* = 3).
Simplimax	1	Fraction of near-zero loadings (*default* = 0.75).
Target	1	$p \times m$ matrix of target loadings. Missing values correspond to unrestricted elements. (*No default.*)

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 pos-

sibility 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:

$$\hat{G}_i = W'(Z_i - \mu_Z) \tag{56.12}$$

where 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 $\hat{\epsilon}$ 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 $W'\Sigma W$, 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 pri-

mary 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 $R_{ff} = \bar{W}'\Sigma\bar{W}$. 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 C. 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 947), 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.

Optimization Method

A majority of the EViews nonlinear estimators offer you the choice of optimization method. For these estimators, the **Optimization method** dropdown menu lets you choose between the **BFGS**, **Gauss-Newton**, **Newton-Raphson**, and **EViews Legacy** methods. The default method is estimator specific.

In general, the differences between the estimates should be small for well-behaved nonlinear specifications, but if you are experiencing optimization difficulties, you may wish to experiment with methods. Note that EViews legacy is a particular implementation of Gauss-Newton with Marquardt or line search steps, and is provided for backward estimation compatibility.

The **Step method** allow you to choose the approach for choosing candidate iterative steps. The default method is **Marquardt**, but you may instead select **Dogleg** or **Line Search**.

See "Optimization Algorithms" on page 1319 for extensive discussion.

Iteration and Convergence

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:

$$\frac{\left\|\theta_{(i+1)} - \theta_{(i)}\right\|_2}{\left\|\theta_{(i)}\right\|_2} \leq tol \tag{C.1}$$

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.

Starting Coefficient Values

Iterative estimation procedures require starting values for the coefficients of the model. There are no general rules for selecting 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** dropdown menu 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 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 func-

tion 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.

Derivative Computation

In many EViews estimation procedures, you can specify the form of the function for the mean equation or the objective function. 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 needs to compute derivatives of the user-specified function. EViews uses two techniques for evaluating derivatives: numeric (finite difference) and analytic.

In most cases, you need not worry about the settings for the derivative computation. The EViews estimation engine will generally employ analytic expressions for the derivatives, if possible, or will compute high numeric derivatives, switching between lower precision computation early in the iterative procedure and higher precision computation for later iterations and final computation.

For the legacy optimizer, EViews may offer 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)

The some cases, EViews will offer you settings for controlling the derivative taking:

- 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.
- In a limited number of cases, EViews will always use numeric derivatives. For example, selected GARCH (see "Derivative Methods" on page 270) 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. If you elect to use EViews legacy estimation, the dialog will also allow you 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 21. "Additional Regression Tools," beginning on page 23).

Computing the more accurate numeric derivatives requires additional objective function evaluations. EViews legacy 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). The newer EViews engine computes derivatives in an adaptive method to achieve high precision.

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, $\theta_{(i)}$:

$$\begin{aligned} g_{(i)} + H_{(i)}(\theta_{(i+1)} - \theta_{(i)}) &= 0 \\ \theta_{(i+1)} &= \theta_{(i)} - H_{(i)}^{-1} g_{(i)} \end{aligned} \tag{C.2}$$

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:

$$\theta_{(i+1)} = \theta_{(i)} - \tilde{H}_{(i)}^{-1} g_{(i)} \qquad \text{where } -\tilde{H}_{(i)} = -H_{(i)} + \alpha I \tag{C.3}$$

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 selected other nonlinear models (ARCH and GARCH, GMM, State Space), EViews provides two first derivative methods: Gauss-Newton/BHHH or Marquardt.

Nonlinear single equation and system models are estimated using the Marquardt method.

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. In many cases, 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:

$$\begin{aligned} x_1 &= f_1(x_1, x_2, \ldots, x_N, z) \\ x_2 &= f_2(x_1, x_2, \ldots, x_N, z) \\ &\mathrm{M} \\ x_N &= f_N(x_1, x_2, \ldots, x_N, z) \end{aligned} \tag{C.4}$$

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:

$$x^{(i+1)} = f(x^{(i)}, z) . \tag{C.5}$$

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:

$$x_k^{(i)} = f_k(x_1^{(i)}, x_2^{(i)}, \ldots, x_{k-1}^{(i)}, x_k^{(i-1)}, x_{k+1}^{(i-1)}, \ldots, x_N^{(i-1)}, z) . \tag{C.6}$$

The performance of the Gauss-Seidel method can be affected be 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:

$$F(x, z) = 0 \tag{C.7}$$

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^* :

$$F(x, z) = F(x^*, z^*) + \frac{\partial}{\partial x} F(x^*, z^*)\Delta x = 0 \tag{C.8}$$

and then use this approximation to construct an iterative procedure for updating our current guess for x:

$$x_{t+1} = x_t - \left[\frac{\partial}{\partial x} F(x_t, z^*)\right]^{-1} F(x_t, z^*) \tag{C.9}$$

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:

$$x_{t+1} = x_t - J_t^{-1} F(x_t, z^*) \tag{C.10}$$

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:

$$J_{t+1} = J_t + \frac{(F(x_{t+1}, z^*) - F(x_t, z^*) - J_t \Delta x)\Delta x'}{\Delta x' \Delta x} \tag{C.11}$$

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.

Berndt, E., Hall, B., Hall, R., and Hausman, J. (1974). *Estimation and Inference in Nonlinear Structural Models*, Annals of Economic and Social Measurement, Vol.. 3, 653–665.

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.

More and Sorensen (1983). "Computing a Trust Region Step," *SIAM Journal of Scientific Statistical Computing,* 4, 553–572.

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 D. 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:

$$S(\beta) = \sum_t (y_t - X_t'\beta)^2 \tag{D.1}$$

The first-order conditions for this objective function are obtained by differentiating with respect to β, yielding

$$\sum_t -2(y_t - X_t'\beta)X_t \tag{D.2}$$

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 23. "Instrumental Variables and GMM," beginning on page 89, and 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 1320).

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:

$$\tilde{u}_t \quad \text{on} \left(\frac{\partial f_t(X_t, \tilde{\beta})}{\partial \beta} \right) \quad \text{and} \quad Z_t \tag{D.3}$$

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" on page 1317.

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:

$$y_t = c(1)(1 - \exp(-c(2)x_t)) + \epsilon_t \tag{D.4}$$

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:

$$y_t = c(1)(1 - \exp(-\phi(c(2)x_t))) + \epsilon_t \tag{D.5}$$

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:

Gradients of the Objective Function

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 E. 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:

$$\tau = \alpha^2 \sum_t \tilde{y}_{t-1}^2 / \sigma^2 \tag{E.1}$$

for $\tilde{y}_t \equiv y_t$ when computing the ADF test equation (42.7), and for $\tilde{y}_t$ as defined in ("Autoregressive Spectral Density Estimator," beginning on page 677) 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:

  $$l = -\frac{TM}{2}(1+\log 2\pi) - \frac{T}{2}\log|\hat{\Omega}| \tag{E.2}$$

  where

  $$|\hat{\Omega}| = \det\left(\sum_i \hat{\epsilon}\hat{\epsilon}'/T\right) \tag{E.3}$$

  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.

- Many estimation methods, including least squares regression, do not treat the error variance term, sigma, as an estimated coefficient, and as such omit this term from the calculation of k.

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.

Appendix F. Long-run Covariance Estimation

The long-run (variance) covariance matrix (LRCOV) occupies an important role in modern econometric analysis. This matrix is, for example, central to calculation of efficient GMM weighting matrices (Hansen 1982), heteroskedastic and autocorrelation (HAC) robust standard errors (Newey and West 1987), and is employed in unit root (Phillips and Perron 1988) and cointegration analysis (Phillips and Hansen 1990, Hansen 1992b).

EViews offers tools for computing symmetric LRCOV and the one-sided LRCOV using nonparametric kernel (Newey-West 1987, Andrews 1991), parametric VARHAC (Den Haan and Levin 1997), and prewhitened kernel (Andrews and Monahan 1992) methods. In addition, EViews supports Andrews (1991) and Newey-West (1994) automatic bandwidth selection methods for kernel estimators, and information criteria based lag length selection methods for VARHAC and prewhitening estimation.

Technical Discussion

Our basic discussion and notation follows the framework of Andrews (1991) and Hansen (1992a).

Consider a sequence of mean-zero random p-vectors $\{V_t(\theta)\}$ that may depend on a K-vector of parameters θ, and let $V_t \equiv V_t(\theta_0)$ where θ_0 is the true value of θ. We are interested in estimating the LRCOV matrix Ω,

$$\Omega = \sum_{j=-\infty}^{\infty} \Gamma(j) \tag{F.1}$$

where

$$\begin{aligned} \Gamma(j) &= E(V_t V_{t-j}') \qquad j \geq 0 \\ \Gamma(j) &= \Gamma(-j)' \qquad j < 0 \end{aligned} \tag{F.2}$$

is the autocovariance matrix of V_t at lag j. When V_t is second-order stationary, Ω equals 2π times the spectral density matrix of V_t evaluated at frequency zero (Hansen 1982, Andrews 1991, Hamilton 1994).

Closely related to Ω are two measures of the *one-sided* LRCOV matrix:

$$\begin{aligned} \Lambda_1 &= \sum_{j=1}^{\infty} \Gamma(j) \\ \Lambda_0 &= \sum_{j=0}^{\infty} \Gamma(j) = \Gamma(0) + \Lambda_1 \end{aligned} \tag{F.3}$$

The matrix Λ_1, which we term the *strict* one-sided LRCOV, is the sum of the lag covariances, while the Λ_0 also includes the contemporaneous covariance $\Gamma(0)$. The two-sided LRCOV matrix Ω is related to the one-sided matrices through $\Omega = \Gamma(0) + \Lambda_1 + \Lambda_1'$ and $\Omega = \Lambda_0 + \Lambda_0' - \Gamma(0)$.

Despite the important role the one-sided LRCOV matrix plays in the literature, we will focus our attention on Ω, since results are generally applicable to all three measures; exception will be made for specific issues that require additional comment.

In the econometric literature, methods for using a consistent estimator $\hat{\theta}$ and the corresponding $\hat{V}_t \equiv V_t(\hat{\theta})$ to form a consistent estimate of Ω are often referred to as *heteroskedasticity and autocorrelation consistent* (HAC) covariance matrix estimators.

There have been three primary approaches to estimating Ω:

1. The *nonparametric kernel* approach (Andrews 1991, Newey-West 1987) forms estimates of Ω by taking a weighted sum of the sample autocovariances of the observed data.
2. The *parametric VARHAC* approach (Den Haan and Levin 1997) specifies and fits a parametric time series model to the data, then uses the estimated model to obtain the implied autocovariances and corresponding Ω.
3. The *prewhitened kernel* approach (Andrews and Monahan 1992) is a hybrid method that combines the first two approaches, using a parametric model to obtain residuals that "whiten" the data, and a nonparametric kernel estimator to obtain an estimate of the LRCOV of the whitened data. The estimate of Ω is obtained by "recoloring" the prewhitened LRCOV to undo the effects of the whitening transformation.

Below, we offer a brief description of each of these approaches, paying particular attention to issues of kernel choice, bandwidth selection, and lag selection.

Nonparametric Kernel

The class of kernel HAC covariance matrix estimators in Andrews (1991) may be written as:

$$\hat{\Omega} = \frac{T}{T-K} \sum_{j=-\infty}^{\infty} k(j/b_T) \cdot \hat{\Gamma}(j) \tag{F.4}$$

where the sample autocovariances $\hat{\Gamma}(j)$ are given by

$$\begin{aligned} \hat{\Gamma}(j) &= \frac{1}{T} \sum_{t=j+1}^{T} \hat{V}_t \hat{V}_{t-j}' \qquad j \geq 0 \\ \hat{\Gamma}(j) &= \hat{\Gamma}(-j)' \qquad j < 0 \end{aligned} \tag{F.5}$$

k is a symmetric kernel (or lag window) function that, among other conditions, is continous at the origin and satisfies $|k(x)| \leq 1$ for all x with $k(0) = 1$, and $b_T > 0$ is a bandwidth parameter. The leading $T/(T-K)$ term is an *optional* correction for degrees-of-freedom associated with the estimation of the K parameters in θ.

The choice of a kernel function and a value for the bandwidth parameter completely characterizes the kernel HAC estimator.

Kernel Functions

There are a large number of kernel functions that satisfy the required conditions. EViews supports use of the following kernel shapes:

Truncated uniform	$k(x) = \begin{cases} 1 & \text{if } \lvert x\rvert \leq 1.0 \\ 0 & \text{otherwise} \end{cases}$
Bartlett	$k(x) = \begin{cases} 1-\lvert x\rvert & \text{if } \lvert x\rvert \leq 1.0 \\ 0 & \text{otherwise} \end{cases}$
Bohman	$k(x) = \begin{cases} (1-\lvert x\rvert)\cos(\pi x) + \dfrac{\sin(\pi\lvert x\rvert)}{\pi} & \text{if } \lvert x\rvert \leq 1.0 \\ 0 & \text{otherwise} \end{cases}$
Daniell	$k(x) = \sin(\pi x)/(\pi x)$
Parzen	$k(x) = \begin{cases} 1-6x^2(1-\lvert x\rvert) & \text{if } 0.0 \leq \lvert x\rvert \leq 0.5 \\ 2(1-\lvert x\rvert)^3 & \text{if } 0.5 < \lvert x\rvert \leq 1.0 \\ 0 & \text{otherwise} \end{cases}$
Parzen-Riesz	$k(x) = \begin{cases} 1-x^2 & \text{if } \lvert x\rvert \leq 1.0 \\ 0 & \text{otherwise} \end{cases}$
Parzen-Geometric	$k(x) = \begin{cases} 1/(1+\lvert x\rvert) & \text{if } \lvert x\rvert \leq 1.0 \\ 0 & \text{otherwise} \end{cases}$
Parzen-Cauchy	$k(x) = \begin{cases} 1/(1+x^2) & \text{if } \lvert x\rvert \leq 1.0 \\ 0 & \text{otherwise} \end{cases}$

Quadratic Spectral	$k(x) = \frac{25}{12\pi^2 x^2}\left(\frac{\sin(6\pi x/5)}{6\pi x/5} - \cos(6\pi x/5)\right)$
Tukey-Hamming	$k(x) = \begin{cases} 0.54 + 0.46\cos(\pi x) & \text{if } \lvert x\rvert \le 1.0 \\ 0 & \text{otherwise} \end{cases}$
Tukey-Hanning	$k(x) = \begin{cases} 0.50 + 0.50\cos(\pi x) & \text{if } \lvert x\rvert \le 1.0 \\ 0 & \text{otherwise} \end{cases}$
Tukey-Parzen	$k(x) = \begin{cases} 0.436 + 0.564\cos(\pi x) & \text{if } \lvert x\rvert \le 1.0 \\ 0 & \text{otherwise} \end{cases}$

Note that $k(x) = 0$ for $|x| > 1$ for all kernels with the exception of the Daniell and the Quadratic Spectral. The Daniell kernel is presented in truncated form in Neave (1972), but EViews uses the more common untruncated form. The Bartlett kernel is sometimes referred to as the Fejer kernel (Neave 1972).

A wide range of kernels have been employed in HAC estimation. The truncated uniform is used by Hansen (1982) and White (1984), the Bartlett kernel is used by Newey and West (1987), and the Parzen is used by Gallant (1987). The Tukey-Hanning and Quadratic Spectral were introduced to the econometrics literature by Andrews (1991), who shows that the latter is optimal in the sense of minimizing the asymptotic truncated MSE of the estimator (within a particular class of kernels). The remaining kernels are discussed in Parzen (1958, 1961, 1967).

Bandwidth

The bandwidth b_T operates in concert with the kernel function to determine the weights for the various sample autocovariances in Equation (F.4). While some authors restrict the bandwidth values to integers, we follow Andrews (1991) who argues in favor of allowing real valued bandwidths.

To construct an operational nonparametric kernel estimator, we must choose a value for the bandwidth b_T. Under general conditions (Andrews 1991), consistency of the kernel estimator requires that b_T is chosen so that $b_T \to \infty$ and $b_T / T \to 0$ as $T \to \infty$. Alternately, Kiefer and Vogelsang (2002) propose setting $b_T = T$ in a testing context.

For the great majority of supported kernels $k(j / b_T) = 0$ for $|j| > b_T$ so that the bandwidth acts indirectly as a lag truncation parameter. Relating b_T to the corresponding integer lag number of included lags m requires, however, examining the properties of the kernel at the endpoints $(|j / b_T| = 1)$. For kernel functions where $k(1) \neq 0$ (*e.g.*, Truncated,

Parzen-Geometric, Tukey-Hanning), b_T is simply a real-valued truncation lag, with at most $m = \text{floor}(b_T)$ autocovariances having non-zero weight. Alternately, for kernel functions where $k(1) = 0$ (*e.g.*, Bartlett, Bohman, Parzen), the relationship is slightly more complex, with $m = \text{ceil}(b_T) - 1$ autocovariances entering the estimator with non-zero weights.

The varying relationship between the bandwidth and the lag-truncation parameter implies that one should examine the kernel function when choosing bandwidth values to match computations that are quoted in lag truncation form. For example, matching Newey-West's (1987) Bartlett kernel estimator which uses m weighted autocovariance lags requires setting $b_T = m + 1$. In contrast, Hansen's (1982) or White's (1984) estimators, which sum the first m unweighted autocovariances, should be implemented using the Truncated kernel with $b_T = m$.

Automatic Bandwidth Selection

Theoretical results on the relationship between bandwidths and the asymptotic truncated MSE of the kernel estimator provide finer discrimination in the rates at which bandwidths should increase. The optimal bandwidths may be written in the form:

$$b_T = \gamma T^{1/(2q+1)} \tag{F.6}$$

where γ is a constant, and q is a parameter that depends on the kernel function that you select (Parzen 1958, Andrews 1991). For the Bartlett and Parzen-Geometric kernels $(q = 1)$ b should grow (at most) at the rate $T^{1/3}$. The Truncated kernel does not have an theoretical optimal rate, but Andrews (1991) reports Monte Carlo simulations that suggest that $T^{1/5}$ works well. The remaining EViews supported kernels have $(q = 2)$ so their optimal bandwidths grow at rate $T^{1/5}$ (though we point out that Daniell kernel does not satisfy the conditions for the optimal bandwidth theorems).

While theoretically useful, knowledge of the rate at which bandwidths should increase as $T \to \infty$ does not tell us the optimal bandwidth for a given sample size, since the constant γ remains unspecified.

Andrews (1991) and Newey and West (1994) offer two approaches to estimating γ. We may term these techniques *automatic bandwidth selection methods*, since they involve estimating the optimal bandwidth from the data, rather than specifying a value *a priori*. Both the Andrews and Newey-West estimators for γ may be written as:

$$\hat{\gamma}(q) = c_k \hat{\alpha}(q)^{1/(2q+1)} \tag{F.7}$$

where q and the constant c_k depend on properties of the selected kernel and $\hat{\alpha}(q)$ is an estimator of $\alpha(q)$, a measure of the smoothness of the spectral density at frequency zero that depends on the autocovariances $\Gamma(j)$. Substituting into Equation (F.6), the resulting plug-in estimator for the optimal automatic bandwidth is given by:

$$\hat{b}_T{}^* = c_k(\hat{\alpha}(q)T)^{1/(2q+1)} \tag{F.8}$$

The q that one uses depends on properties of the selected kernel function. The Bartlett and Parzen-Geometric kernels should use $\hat{\alpha}(1)$ since they have $q = 1$. $\hat{\alpha}(2)$ should be used for the other EViews supported kernels which have $q = 2$. The Truncated kernel does not have a theoretically proscribed choice, but Andrews recommends using $\hat{\alpha}(2)$. The Daniell kernel has $q = 2$, though we remind you that it does not satisfy the conditions for Andrews's theorems. "Kernel Function Properties" on page 1349 summarizes the values of c_k and q for the various kernel functions.

It is of note that the Andrews and Newey-West estimators both require an estimate of $\alpha(q)$ that requires forming preliminary estimates of Ω and the smoothness of Ω. Andrews and Newey-West offer alternative methods for forming these estimates.

Andrews Automatic Selection

The Andrews (1991) method estimates $\alpha(q)$ parametrically: fitting a simple parametric time series model to the original data, then deriving the autocovariances $\Gamma(j)$ and corresponding $\alpha(q)$ implied by the estimated model.

Andrews derives $\hat{\alpha}(q)$ formulae for several parametric models, noting that the choice between specifications depends on a tradeoff between simplicity and parsimony on one hand and flexibility on the other. EViews employs the parsimonius approach used by Andrews in his Monte Carlo simulations, estimating p-univariate AR(1) models (one for each element of $\hat{V}_t$), then combining the estimated coefficients into an estimator for $\alpha(q)$.

For the univariate AR(1) approach, we have:

$$\hat{\alpha}(q) = \left(\sum_{s=1}^{p} w_s (\hat{f}_s^{(q)})^2 \right) \Big/ \left(\sum_{s=1}^{p} w_s (\hat{f}_s^{(0)})^2 \right) \tag{F.9}$$

where $\hat{f}_s^{(q)}$ are parametric estimators of the smoothness of the spectral density for the s-th variable (Parzen's (1957) q-th generalized spectral derivatives) at frequency zero. Estimators for $\hat{f}_s^{(q)}$ are given by:

$$\hat{f}_s^{(q)} = \frac{1}{2\pi} \sum_{j=-\infty}^{\infty} |j|^q \cdot \tilde{\Gamma}_s(j) \tag{F.10}$$

for $s = 1, \ldots, p$ and $q = 0, 1, 2$, where $\tilde{\Gamma}_s(j)$ are the estimated autocovariances at lag j implied by the univariate AR(1) specification for the s-th variable.

Substituting the univariate AR(1) estimated coefficients $\hat{\rho}_s$ and standard errors $\hat{s}_s$ into the theoretical expressions for $\tilde{\Gamma}_s(j)$, we have:

$$\hat{\alpha}(1) = \left(\sum_{s=1}^{p} w_s \frac{4\mathfrak{s}_s^4 \hat{\rho}_s^2}{(1-\hat{\rho}_s)^6 (1+\hat{\rho}_s)^2} \right) \Big/ \left(\sum_{s=1}^{p} w_s \frac{\mathfrak{s}_s^4}{(1-\hat{\rho}_s)^4} \right)$$
$$\hat{\alpha}(2) = \left(\sum_{s=1}^{p} w_s \frac{4\mathfrak{s}_s^4 \hat{\rho}_s^2}{(1-\hat{\rho}_s)^8} \right) \Big/ \left(\sum_{s=1}^{p} w_s \frac{\mathfrak{s}_s^4}{(1-\hat{\rho}_s)^4} \right) \tag{F.11}$$

which may be inserted into Equation (F.8) to obtain expressions for the optimal bandwidths.

Lastly, we note that the expressions for $\hat{\alpha}(q)$ depend on the weighting vector w which governs how we combine the individual $\hat{f}_s^{(q)}$ into a single measure of relative smoothness. Andrews suggests using either $w_s = 1$ for all s or $w_s = 1$ for all but the instrument corresponding to the intercept in regression settings. EViews adopts the first suggestion, setting $w_s = 1$ for all s.

Newey-West Automatic Selection

Newey-West (1994) employ a nonparametric approach to estimating $\alpha(q)$. In contrast to Andrews who computes parametric estimates of the individual $f_s^{(q)}$, Newey-West uses a Truncated kernel estimator to estimate the $f^{(q)}$ corresponding to aggregated data.

First, Newey and West define, for various lags, the *scalar* autocovariance estimators:

$$\hat{\sigma}_j = \frac{1}{T} \sum_{t=j+1}^{T} w' \hat{V}_t \hat{V}_{t-j}' w = w' \hat{\Gamma}(j) w \tag{F.12}$$

The $\hat{\sigma}_j$ may either be viewed as the sample autocovariance of a weighted linear combination of the data using weights w, or as a weighted combination of the sample autocovariances.

Next, Newey and West use the $\hat{\sigma}_j$ to compute nonparametric truncated kernel estimators of the Parzen measures of smoothness:

$$\hat{\mathcal{J}}^{(q)} = \frac{1}{2\pi} \sum_{j=-n}^{n} |j|^q \cdot \hat{\sigma}_j \tag{F.13}$$

for $q = 0, 1, 2$. These nonparametric estimators are weighted sums of the scalar autocovariances $\hat{\sigma}_j$ obtained above for j from $-n$ to n, where n, which Newey and West term the *lag selection parameter*, may be viewed as the bandwidth of a kernel estimator for $\hat{\mathcal{J}}^{(q)}$.

The Newey and West estimator for $\hat{\alpha}(q)$ may then be written as:

$$\hat{\alpha}(q) = (\hat{\mathcal{J}}^{(q)} / \hat{\mathcal{J}}^{(0)})^2 \tag{F.14}$$

for $q = 1, 2$. This expression may be inserted into Equation (F.8) to obtain the expression for the plug-in optimal bandwidth estimator.

In comparing the Andrews estimator Equation (F.11) with the Newey-West estimator Equation (F.14) we see two very different methods of distilling results from the p-dimensions of the original data into a scalar measure $\alpha(q)$. Andrews computes parametric estimates of the generalized derivatives for the p individual elements, then aggregates the estimates into a single measure. In contrast, Newey and West aggregate early, forming linear combinations of the autocovariance matrices, then use the scalar results to compute nonparametric estimators of the Parzen smoothness measures.

To implement the Newey-West optimal bandwidth selection method we require a value for n, the lag-selection parameter, which governs how many autocovariances to use in forming the nonparametric estimates of $f^{(q)}$. Newey and West show that n should increase at (less than) a rate that depends on the properties of the kernel. For the Bartlett and the Parzen-Geometric kernels, the rate is $T^{2/9}$. For the Quadratic Spectral kernel, the rate is $T^{2/25}$. For the remaining kernels, the rate is $T^{4/25}$ (with the exception of the Truncated and the Daniell kernels, for which the Newey-West theorems do not apply).

In addition, one must choose a weight vector w. Newey-West (1987) leave open the choice of w, but follow Andrew's (1991) suggestion of $w_s = 1$ for all but the intercept in their Monte Carlo simulations. EViews differs from this choice slightly, setting $w_s = 1$ for all s.

Parametric VARHAC

Den Haan and Levin (1997) advocate the use of parametric methods, notably VARs, for LRCOV estimation. The VAR spectral density estimator, which they term *VARHAC*, involves estimating a parametric VAR model to filter the $\hat{V}_t$, computing the contemporaneous covariance of the filtered data, then using the estimates from the VAR model to obtain the implied autocovariances and corresponding LRCOV matrix of the original data.

Suppose we fit a VAR(q) model to the $\{\hat{V}_t\}$. Let $\hat{A}_j$ be the $p \times p$ matrix of estimated j-th order AR coefficients, $j = 1, \ldots, q$. Then we may define the innovation (filtered) data and estimated innovation covariance matrix as:

$$V_t^* = \hat{V}_t - \sum_{j=1}^{q} \hat{A}_j \hat{V}_{t-j} \tag{F.15}$$

and

$$\hat{\Gamma}^*(0) = \frac{1}{T-q} \sum_{t=q+1}^{T} V_t^* V_t^{*\prime} \tag{F.16}$$

Given an estimate of the innovation contemporaneous variance matrix $\hat{\Gamma}^*(0)$ and the VAR coefficients $\hat{A}_j$, we can compute the implied theoretical autocovariances $\Gamma(j)$ of $\hat{V}_t$. Summing the autocovariances yields a parametric estimator for Ω, given by:

$$\hat{\Omega} = \frac{T-q}{T-q-K} \hat{D} \hat{\Gamma}^*(0) \hat{D} \tag{F.17}$$

where

$$\hat{D} = \left(I_p - \sum_{j=1}^{q} \hat{A}_j \right)^{-1} \tag{F.18}$$

Implementing VARHAC requires a specification for q, the order of the VAR. Den Haan and Levin use model selection criteria (AIC or BIC-Schwarz) using a maximum lag of $T^{1/3}$ to determine the lag order, and provide simulations of the performance of estimator using data-dependent lag order.

The corresponding VARHAC estimators for the one-sided matrices Λ_1 and Λ_0 do not have simple expressions in terms of $\hat{A}_j$ and $\hat{\Gamma}^*(0)$. We can, however, obtain insight into the construction of the one-sided VARHAC LRCOVs by examining results for the VAR(1) case. Given estimation of a VAR(1) specification, the estimators for the one-sided long-run variances may be written as:

$$\begin{aligned} \hat{\Lambda}_1 &= \frac{T-q}{T-q-K} \sum_{j=1}^{\infty} (\hat{A}_1)^j \hat{\Gamma}(0) = \frac{T-q}{T-q-K} \hat{A}_1 (I_p - \hat{A}_1)^{-1} \hat{\Gamma}(0) \\ \hat{\Lambda}_0 &= \frac{T-q}{T-q-K} \sum_{j=0}^{\infty} (\hat{A}_1)^j \hat{\Gamma}(0) = \frac{T-q}{T-q-K} (I_p - \hat{A}_1)^{-1} \hat{\Gamma}(0) \end{aligned} \tag{F.19}$$

Both estimators require estimates of the VAR(1) coefficient estimates $\hat{A}_1$, as well as an estimate of $\Gamma(0)$, the contemporaneous covariance matrix of $\hat{V}_t$.

One could, as in Park and Ogaki (1991) and Hansen (1992b), use the sample covariance matrix $\hat{\Gamma}(0) = (1/T)\sum \hat{V}_t \hat{V}_t'$ so that the estimates of Λ_1 and Λ_0 employ a mix of parametric and non-parametric autocovariance estimates. Alternately, in keeping with the spirit of the parametric methodology, EViews constructs a parametric estimator $\tilde{\Gamma}(0)$ using the estimated VAR(1) coefficients $\hat{A}_1$ and $\hat{\Gamma}^*(0)$.

Prewhitened Kernel

Andrews and Monahan (1992) propose a simple modification of the kernel estimator which performs a parametric VAR *prewhitening* step to reduce autocorrelation in the data followed by kernel estimation performed on the whitened data. The resulting prewhitened LRVAR estimate is then *recolored* to undo the effects of the transformation. The Andrews and Monahan approach is a hybrid that combines the parametric VARHAC and nonparametric kernel techniques.

There is evidence (Andrews and Monahan 1992, Newey-West 1994) that this prewhitening approach has desirable properties, reducing bias, improving confidence interval coverage probabilities and improving sizes of test statistics constructed using the kernel HAC estimators.

The Andrews and Monahan estimator follows directly from our earlier discussion. As in a VARHAC, we first fit a VAR(q) model to the $\hat{V}_t$ and obtain the whitened data (residuals):

$$\hat{V}_t^* = \hat{V}_t - \sum_{j=1}^{q} \hat{A}_j \hat{V}_{t-j} \tag{F.20}$$

In contrast to the VAR specification in the VARHAC estimator, the prewhitening VAR specification is not necessarily believed to be the true time series model, but is merely a tool for obtaining V_t^* values that are closer to white-noise. (In addition, Andrews and Monahan adjust their VAR(1) estimates to avoid singularity when the VAR is near unstable, but EViews does not perform this eigenvalue adjustment.)

Next, we obtain an estimate of the LRCOV of the whitened data by applying a kernel estimator to the residuals:

$$\hat{\Omega}^* = \sum_{j=-\infty}^{\infty} k(j/b_T) \cdot \hat{\Gamma}^*(j) \tag{F.21}$$

where the sample autocovariances $\hat{\Gamma}^*(j)$ are given by

$$\begin{aligned} \hat{\Gamma}^*(j) &= \frac{1}{T-q} \sum_{t=j+q+1}^{T} \hat{V}_t^* \hat{V}_{t-j}^{*\prime} \qquad j \geq 0 \\ \hat{\Gamma}^*(j) &= \hat{\Gamma}^*(-j)' \qquad j < 0 \end{aligned} \tag{F.22}$$

Lastly, we recolor the estimator to obtain the VAR prewhitened kernel LRCOV estimator:

$$\hat{\Omega} = \frac{T-q}{T-q-K} \hat{D} \hat{\Omega}^* \hat{D} \tag{F.23}$$

The prewhitened kernel procedure differs from VARHAC only in the computation of the LRCOV of the residuals. The VARHAC estimator in Equation (F.17) assumes that the residuals V_t^* are white noise so that the LRCOV may be estimated using the contemporaneous variance matrix $\hat{\Gamma}^*(0)$, while the prewhitening kernel estimator in Equation (F.21) allows for residual heteroskedasticity and serial dependence through its use of the HAC estimator $\hat{\Omega}^*$. Accordingly, it may be useful to view the VARHAC procedure as a special case of the prewhitened kernel with $k(0) = 1$ and $k(x) = 0$ for $x \neq 0$.

The recoloring step for one-sided prewhitened kernel estimators is complicated when we allow for HAC estimation of $\hat{\Lambda}_1^*$ (Park and Ogaki, 1991). As in the VARHAC setting, the

expressions for one-sided LRCOVs are quite involved but the VAR(1) specification may be used to provide insight. Suppose that the VARHAC estimators of the one-sided LRCOV matrices defined in Equation (F.19) are given by $\bar{\Lambda}_1$ and $\bar{\Lambda}_0$, and let $\hat{\Lambda}_1^*$ be the *strict* one-sided kernel estimator computed using the prewhitened data:

$$\hat{\Lambda}_1^* = \sum_{j=1}^{\infty} k(j/b_T) \cdot \hat{\Gamma}^*(j) \tag{F.24}$$

Then the prewhitened kernel one-sided LRCOV estimators are given by:

$$\begin{aligned} \hat{\Lambda}_1 &= \bar{\Lambda}_1 + \frac{T-q}{T-q-K} \hat{D} \hat{\Lambda}_1^* \hat{D} \\ \hat{\Lambda}_0 &= \bar{\Lambda}_0 + \frac{T-q}{T-q-K} \hat{D} \hat{\Lambda}_1^* \hat{D} \end{aligned} \tag{F.25}$$

Kernel Function Properties

	q	c_k	r_B	r_n
Truncated uniform	0	0.6611	$1/5$	---
Bartlett	1	1.1447	$1/3$	$2/9$
Bohman	2	2.4201	$1/5$	$4/25$
Daniell	2	0.4462	$1/5$	---
Parzen	2	2.6614	$1/5$	$4/25$
Parzen-Riesz	2	1.1340	$1/5$	$4/25$
Parzen-Geometric	1	1.0000	$1/3$	$2/9$
Parzen-Cauchy	2	1.0924	$1/5$	$4/25$
Quadratic Spectral	2	1.3221	$1/5$	$2/25$
Tukey-Hamming	2	1.6694	$1/5$	$4/25$
Tukey-Hanning	2	1.7462	$1/5$	$4/25$
Tukey-Parzen	2	1.8576	$1/5$	$4/25$

Notes: $r_B = 1/(2q+1)$ is the optimal rate of increase for the LRCOV kernel bandwidth. r_n is the optimal rate of increase for the lag selection parameter in the Newey-West (1987) automatic bandwidth selection procedure. The Truncated uniform kernel does not have the-

oretically proscribed values for c_k and r_B, but Andrews (1991) reports Monte Carlo simulations that suggest that these values work well. The Daniell kernel value for r_B does not follow from the theory since the kernel does not satisfy the conditions of the optimal bandwidth theorems.

References

Andrews, Donald W. K, and J. Christopher Monahan (1992). "An Improved Heteroskedasticity and Autocorrelation Consistent Covariance Matrix Estimator," *Econometrica*, 60, 953-966.

Andrews, Donald W. K. (1991). "Heteroskedaticity and Autocorrelation Consistent Covariance Matrix Estimation," *Econometrica*, 59, 817-858.

den Haan, Wouter J. and Andrew Levin (1997). "A Practitioner's Guide to Robust Covariance Matrix Estimation," Chapter 12 in Maddala, G. S. and C. R. Rao (eds.), *Handbook of Statistics Vol. 15, Robust Inference*, North-Holland: Amsterdam, 291-341.

Gallant, A. Ronald (1987). *Nonlinear Statistical Models*. New York: John Wiley & Sons.

Hamilton, James D. (1994). *Time Series Analysis*, Princeton University Press.

Hansen, Bruce E. (1992a). "Consistent Covariance Matrix Estimation for Dependent Heterogeneous Processes," *Econometrica*, 60, 967-972.

Hansen, Bruce E. (1992b). "Tests for Parameter Instability in Regressions with I(1) Processes," *Journal of Business and Economic Statistics*, 10, 321-335.

Hansen, Lars Peter (1982). "Large Sample Properties of Generalized Method of Moments Estimators," *Econometrica*, 50, 1029-1054.

Kiefer, Nicholas M., and Timothy J. Vogelsang (2002). "Heteroskedasticity-Autocorrelation Robust Standard Errors Using the Bartlett Kernel Without Truncation," *Econometrica*, 70, 2093-2095.

Neave, Henry R. (1972). "A Comparison of Lag Window Generators," *Journal of the American Statistical Association*, 67, 152-158.

Newey, Whitney K. and Kenneth D. West (1987). "A Simple, Positive Semi-Definite, Heteroskedasticity and Autocorrelation Consistent Covariance Matrix," *Econometrica*, 55, 703-708.

Newey, Whitney K. and Kenneth D. West (1994). "Automatic Lag Length Selection in Covariance Matrix Estimation," *Review of Economic Studies*, 61, 631-653.

Park, Joon Y. and Masao Ogaki (1991). "Inferences in Cointegrated Models Using VAR Prewhitening to Estimate Shortrun Dynamics," Rochester Center for Economic Research Working Paper No. 281.

Parzen, Emanual (1957). "Consistent Estimates of the Spectrum of a Stationary Time Series," *The Annals of Mathematical Statistics*, 28, 329-348.

Parzen, Emanuel (1958). "On Asymptotically Efficient Consistent Estimates of the Spectral Density Function of a Stationary Time Series," *Journal of the Royal Statistical Society, B*, 20, 303-322.

Parzen, Emanual (1961). "Mathematical Considerations in the Estimation of Spectra," *Technometrics*, 3, 167-190.

Parzen, Emanual (1967). "On Empirical Multiple Time Series Analysis," in Lucien M. Le Cam and Jerzy Neyman (*eds.*), *Proceedings of the Fifth Berkely Symposium on Mathematical Statistics and Probability*, 1, 305-340.

White, Halbert (1984). *Asymptotic Theory for Econometricians*. Orlando: Academic Press.

Index

(Key: I = User's Guide I; II = User's Guide II)

Symbols

?
  pool cross section identifier *UI:1066*
.DB? files *UI:330*
.EDB file *UI:326*
.RTF file *UI:875*
.WF1 file *UI:76*
`@all` *UI:138*
`@cellid` *UI:1117*
`@clear` *UI:214*
`@count` *UI:207*
`@crossid` *UI:1116*
`@elem` *UI:190*
`@eqnq` *UI:196*
`@expand` *UI:208*, *UI:29*
`@first` *UI:138*
`@firstmax` *UI:1122*
`@firstmin` *UI:1122*
`@ingrp` *UI:1066*
`@isna` *UI:196*
`@last` *UI:138*
`@lastmax` *UI:1122*
`@lastmin` *UI:1122*
`@map` *UI:236*
`@neqna` *UI:196*
`@obsid` *UI:1118*
`@obsnum`
  panel observation numbering *UI:1118*
`@ranks` *UI:194*
`@seriesname` *UI:207*
`@unmap` *UI:237*
`@unmaptxt` *UI:237*
~, in backup file name *UI:76*, *UI:948*

Numerics

1-step GMM
  single equation *UI:105*, *UI:110*
2sls (Two-Stage Least Squares) *UI:89*, *UI:96*
  diagnostics *UI:112*
  dropped instruments *UI:112*
  in systems *UI:793*
  instrument orthogonality test *UI:113*
  instrument summary *UI:112*
  J-statistic *UI:92*
  nonlinear *UI:96*
  nonlinear with AR specification *UI:97*
  order condition *UI:91*
  panels *UI:1138*
  rank condition *UI:91*
  regressor endogeneity test *UI:113*
  residuals *UI:92*
  system estimation *UI:823*
  weak instruments *UI:114*
  weighted in systems *UI:793*, *UI:823*
  weighted nonlinear *UI:97*, *UI:108*
  with AR specification *UI:93*, *UI:159*
  with MA specification *UI:95*
  with pooled data *UI:1105*
3sls (Three Stage Least Squares) *UI:793*, *UI:824*
64-bit version *UI:954*

A

Abort key *UI:14*
Across factors *UI:813*
Active window *UI:113*
Add factor *UI:995*, *UI:1008*
Add text to graph *UI:833*
Adding data *UI:299*
Add-ins *I:899*
ADF
  See also Unit root tests.
Adjusted R-squared
  for regression *UI:13*
Advanced database query *UI:343*
AIC *UII:1335*
  See also Akaike criterion.
Akaike criterion *UI:15*, *UII:1335*
  for equation *UI:15*
Alias *UI:997*
  database *UI:340*
  OBALIAS.INI file *UI:351*

object *UI:349*
Almon lag *UI:24*
Almon weights *UI:343*
Alpha series *UI:216*
additional views *UI:223*
declaring and creating *UI:217*
maximum length *UI:218*, *UI:942*
spreadsheet view *UI:223*
truncation *UI:218*, *UI:942*
Analysis of variance *UI:450*
by ranks *UI:453*
Analytical derivatives *UII:1330*
logl *UI:648*
Analytical graphs *UI:761*
And operator *UI:139*, *UI:191*
Anderson-Darling test *UI:455*
Andrew's automatic bandwidth *UII:1344*
cointegrating regression *UI:305*
GMM estimation *UI:110*
long-run covariance estimation *UI:658*
panel cointegrating regression *UI:1197*
system GMM *UI:827*
Andrews function *UI:452*
Andrews test *UI:372*, *UI:419*
Andrews-Quandt breakpoint test *UI:228*
ANOVA *UI:450*
by ranks *UI:453*
Appending data *UI:299*
AR roots
inverted *UI:146*
AR Roots (VAR) *UI:848*
AR specification
forecast *UI:182*
in 2SLS *UI:93*
in ARIMA models *UI:120*, *UI:132*
in nonlinear 2SLS *UI:97*
in nonlinear least squares *UI:58*
in pool *UI:1084*
in systems *UI:796*
AR(1)
coefficient *UI:120*
Durbin-Watson statistic *UI:127*
estimation *UI:132*
AR(p) *UI:121*
estimation *UI:133*
ARCH *UI:263*
See also GARCH.
correlogram test *UI:214*
LM test *UI:218*
multivariate *UI:794*
system *UI:794*
ARCH test *UI:218*
ARCH-M *UI:265*
ARDL
bounds testing *UI:323*
cointegrating relationships *UI:323*
long-run relationships *UI:322*
panel *UI:1143*
pooled mean group estimation *UI:1143*
Area band graph *UI:748*
Area graph *UI:746*
Arellano-Bond serial correlation test *UI:1182*
AREMOS data *UI:356*
data banks *UI:355*
ARFIMA *UI:125*, *UI:136*
ARFIMA models *UI:124*
ARIMA *UI:124*
ARIMA models *UI:124*
automatic forecasting *UI:538*
automatic selection *UI:538*
automatic selection using X-13 *UI:494*, *UI:497*
Box-Jenkins approach *UI:126*
correlogram *UI:149*
diagnostic checking *UI:147*
difference operator *UI:135*
frequency spectrum *UI:151*
identification *UI:126*
impulse response *UI:150*
roots *UI:148*
specification *UI:132*
starting values *UI:138*, *UI:144*
structure *UI:148*
X-13 *UI:493*
ARIMAX *UI:131*
ARMA terms
in models *UI:1037*
seasonal *UI:123*
testing *UI:215*
using state spaces models for *UI:978*
ARMAX *UI:978*
Array expressions *1:961*
Arrows
adding to a graph *UI:835*
Artificial regression *UI:219*, *UI:256*

ASCII file
  import *UI:150*
  open as workfile *UI:48*
Asymptotic test *UI:195*
Attributes *UI:62*
  adding *UI:65*
  replacing *UI:67*
  viewing *UI:62*
Augmented Dickey-Fuller test *UI:672*
  See also Unit root tests.
Augmented regression *UI:244*
Auto tab indent *UI:948*
Autocorrelation *UI:462*, *UI:14*
  robust standard errors *UI:32*
Autocorrelation test *See* Serial correlation test
Automatic bandwidth selection
  cointegrating regression *UI:305*
  GMM estimation *UI:110*
  long-run covariance estimation *UI:658*
  panel cointegrating regression *UI:1197*
  robust standard errors *UI:45*
  technical details *UII:1343*
Automatic forecast
  ARIMA *UI:538*
  ETS smoothing *UI:568*, *UI:570*
  using X-13 *UI:498*
Automatic variable selection *UII:65*
Autoregressive distributed lag models
  See ARDL.
Autoregressive spectral density estimator *UI:677*
Auto-search
  database *UI:341*
Auto-series *UI:201*
  forecasting *UI:188*
  generate new series *UI:201*, *UI:338*
  in estimation *UI:205*
  in groups *UI:205*
  in regression *UI:338*
  with database *UI:337*
Auto-updating graph *UI:828*
Auto-updating series *UI:211*
  and databases *UI:215*
  converting to ordinary series *UI:214*
Auxiliary graphs *UI:780*
Auxiliary regression *UI:215*, *UI:218*
Average log likelihood *UI:366*
Average shifted histogram *UI:766*
Axis *UI:727*
  assignment *UI:728*
  characteristics *UI:730*, *UI:733*
  custom obs labels *UI:849*
  data ticks and lines *UI:731*
  date labels *UI:737*
  date ticks *UI:737*
  duplicating *UI:732*
  format *UI:732*
  labels *UI:730*, *UI:731*, *UI:732*
  remove custom date labels *UI:851*
  scale *UI:733*

B

Backcast
  in GARCH models *UI:269*
  MA terms *UI:140*
Backup files *UI:76*, *UI:948*
Bai Perron breakpoint test *UI:230*
  computing in EViews *UI:233*
  examples *UI:235*
Bai sequential breakpoint
  estimation with *UI:472*
  test *UI:232*
Balanced data *UI:1072*
Balanced sample *UI:1083*
Baltagi, Fend and Kao test *UI:1177*, *UI:1239*
Band-Pass filter *UI:581*
Bandwidth
  Andrews *UI:828*, *UII:1344*
  automatic selection *See* Automatic bandwidth selection
  bracketing *UI:769*, *UI:785*, *UI:786*
  cointegrating regression *UI:305*
  GMM estimation *UI:110*
  kernel - technical details *UII:1342*
  kernel graph *UI:769*, *UI:784*
  local regression *UI:786*
  long-run covariance estimation *UI:658*
  Newey-West (automatic) *UI:828*, *UII:1345*
  Newey-West (fixed) *UI:827*
  panel cointegrating regression *UI:1197*
  robust standard errors *UI:45*
  selection in system GMM *UI:801*, *UI:827*
Bar graph *UI:746*
Bartlett kernel *UI:827*
  cointegrating regression *UI:305*

GMM estimation *UI:110*
long-run covariance estimation *UI:658*
panel cointegrating regression *UI:1197*
robust standard errors *UI:45*
technical details *UII:1341*
Bartlett test *UI:454*
Baxter-King band-pass filter *UI:581*
Bayesian model averaging *UI:549*
Bayesian VAR *See* BVAR
BDS test *UI:466*, *UI:749*
BEA data *UI:360*
Bekker standard errors *UI:99*
Berndt-Hall-Hall-Hausman (BHHH). *See* Optimization algorithms.
Bias proportion *UI:178*
BIC *UII:1335*
See also Schwarz criterion.
Bin width
histograms *UI:762*
See also Binning
Binary dependent variable *UI:361*
categorical regressors stats *UI:369*
error messages *UI:368*
estimation *UI:363*
estimation statistics *UI:366*
expectation-prediction table *UI:370*
fitted index *UI:375*
forecasting *UI:375*
goodness-of-fit *UI:372*
interpretation of coefficient *UI:365*
log likelihood *UI:362*
predicted probabilities *UI:376*
residuals *UI:375*
response curve *UI:376*
views *UI:369*
Binary estimation
dependent variable frequencies *UI:369*
perfect predictor *UI:368*
Binary file *UI:48*
Binning *UI:769*, *UI:783*, *UI:785*
categorical graphs *UI:811*
classifications *UI:444*, *UI:629*
Binomial sign test *UI:448*
Bisquare function *UI:452*
Blom *UI:772*
Bloomberg data *UI:356*
BLS data *UI:364*
BLS MoveReg *UI:514*
BMA *UI:549*
BMP *UI:863*
Bohman kernel
cointegrating regression *UI:305*
GMM estimation *UI:110*
long-run covariance estimation *UI:658*
panel cointegrating regression *UI:1197*
robust standard errors *UI:45*
technical details *UII:1341*
Bonferroni *UI:623*
Bootstrap *UI:482*
in quantile regression *UI:621*, *UI:622*, *UI:637*
Bounds testing (ARDL models) *UI:323*
Box-Cox transformation *UI:781*
Box-Jenkins *UI:126*
Boxplot *UI:777*
as axis *UI:687*
Break *UI:14*
Breakpoint estimation *UI:471*
coefficient labeling *UI:480*
estimation output *UI:477*
example *UI:483*
in EViews *UI:473*
specification view *UI:479*
Breakpoint test
See also Breakpoint estimation
Bai and Perron *UI:230*
Chow *UI:226*
estimation after *UI:471*
factor *UI:211*
for unit roots *UI:679*
GMM *UI:116*
multiple *UI:230*
Quandt-Andrews *UI:228*
unequal variance *UI:254*
unknown *UI:228*
Breitung *UI:712*
Breusch-Godfrey test *UI:128*, *UI:215*, *UI:850*
Breusch-Pagan test *UI:217*
cross-section dependence *UI:1177*, *UI:1239*
Brown-Forsythe test *UI:455*
Broyden's method *UI:1044*, *UII:1324*
Bubble plot *UI:753*
Bureau of Economic Analysis data *UI:360*
Bureau of Labor Statistics data *UI:364*
BVAR *UI:895*

By-group statistics *UI:443*, *UI:1124*, *UI:1133*

C

C
  coef vector *UI:6*
  constant in regression *UI:6*
Cache *UI:432*
Cancel keystroke *UI:14*
Canonical cointegrating regression *UI:295*, *UI:302*
Categorical graphs *UI:793*
  See also Graphs.
  analytical *UI:802*
  binning *UI:811*
  category summaries *UI:794*
  descriptive statistics *UI:794*
  factor display settings *UI:813*
  factor labeling *UI:822*
  factor ordering *UI:814*
  factors *UI:810*
  identifying categories *UI:805*
  line *UI:798*
  specifying factors *UI:810*
  summaries *UI:794*
Categorical regressor stats *UI:369*, *UI:392*
Cauchy function *UI:452*
Causality
  Dumitrescu-Hurlin *UI:1231*
  Granger's test *UI:661*
  panel data *UI:1230*
CD test *UI:1177*, *UI:1239*
CEIC *UI:367*
CEIC data *UI:367*
Cell
  annotation *UI:872*
  formatting *UI:870*
  merging *UI:873*
  selection *UI:865*
Censored dependent variable *UI:387*
  estimation *UI:388*
  expected dependent variable *UI:393*
  fitted index *UI:393*
  forecasting *UI:393*
  goodness-of-fit tests *UI:394*
  interpretation of coefficient *UI:392*
  log likelihood *UI:388*
  residuals *UI:392*
  scale factor *UI:392*
  specifying the censoring point *UI:389*
  views *UI:392*
Census *UI:368*
Census data *UI:368*
Census X-11
  using X-12 *UI:525*
Census X-12 *UI:523*
  seasonal adjustment options *UI:524*
Census X-13 *UI:487*
CGARCH *UI:279*
Change default directory *UI:95*
Chi-square
  independence test in tabulation *UI:631*
  statistic for Wald test *UI:204*
  test for independence in n-way table *UI:632*
  test for the median *UI:452*
Cholesky factor
  in VAR impulse responses *UI:855*
  in VAR normality test *UI:851*
Chow test
  breakpoint *UI:226*
  forecast *UI:242*
  n-step forecast *UI:248*
  one-step forecast *UI:248*
Chow-Lin
  frequency conversion method *UI:173*, *UI:176*
Christiano-Fitzgerald band-pass filter *UI:581*
Circular filtering *UI:777*
Classification
  from series *UI:477*
Classification table
  binary models *UI:370*
  ordered models *UI:384*
  sensitivity *UI:371*
  specificity *UI:371*
Cleveland subsampling *UI:788*
Close
  EViews *UI:14*
  object *UI:937*
Cloud drive *UI:96*
Cluster
  robust standard errors *UI:39*
Clustering
  by cross-section *UI:1104*, *UI:1107*, *UI:1108*
  by period *UI:1103*, *UI:1107*
Cochrane-Orcutt *UI:94*, *UI:163*

Coef (coefficient vector)
default *UI:6*
update from equation *UI:19*
Coefficient
common (pool) *UI:1084*
covariance matrix *UI:17*
covariance matrix of estimates *UI:19*
cross-section specific (pool) *UI:1084*
diagnostics *UI:196*
elasticity at means *UI:196*
estimated from regression *UI:11*
heteroskedasticity consistent covariances *UI:36*
maximum number in default *UI:384*
recursive estimates *UI:249*
regression *UI:11*
restrictions *UI:8*
scaled *UI:196*
setting initial values *UI:57*, *UII:1315*
standard error *UI:12*
standardized *UI:196*
tests *UI:196*, *UI:202*
variance decomposition *UI:200*
vectors of *UI:20*
Coefficient restrictions *UI:658*
cross-equation *UI:796*
Coefficient uncertainty *UI:1015*, *UI:1031*, *UI:1041*
Cointegrating regression *UI:293*
equation specification *UI:296*
panel *See Panel cointegrating regression.*
Cointegrating regressors *UI:297*
panel *UI:1194*
Cointegration *UI:1245*
See also Cointegrating regression.
ARDL models *UI:323*
Hansen instability test *UI:313*
panel testing *UI:1238*, *UI:1258*
Park added variable test *UI:316*
pooled data testing *UI:1078*
regression *UI:293*
residual tests *UI:308*, *UI:1254*
restrictions *UI:884*, *UI:1252*
test *UI:308*, *UI:1254*
test critical values *UI:1248*, *UI:1263*
VAR *UI:1245*
Collinearity *UI:23*
coefficient variance decomposition *UI:200*
test of *UI:199*, *UI:200*
variance inflation factors *UI:199*
Color
EViews application *UI:936*
graph frame *UI:725*
printing in *UI:862*
tables *UI:871*
Column width *UI:870*
Combination testing *UI:467*
Command window *UI:7*
history of *UI:8*
Commands
history of *UI:8*
Comments *UI:116*
spool *UI:882*
tables *UI:872*
Common sample *UI:195*
Communalities *UI:1271*
Comparing workfiles and pages *UI:92*
Comparison operators *UI:190*
with missing values *UI:196*
Component GARCH *UI:279*
Component plots *UI:642*
Conditional independence *UI:632*
Conditional standard deviation
display graph of *UI:273*
Conditional variance *UI:261*, *UI:263*, *UI:264*
forecast *UI:275*
in the mean equation *UI:265*
make series from ARCH *UI:276*
Confidence ellipses *UI:789*, *UI:196*
Confidence interval *UI:196*
ellipses *UI:196*
for forecast *UI:177*
for stochastic model solution *UI:1039*
Constant
in equation *UI:6*, *UI:12*
in ordered models *UI:382*
Contemporaneous covariance (in panels) *UI:1219*
Contingency coefficient *UI:632*
Continuously updating GMM
single equation *UI:105*, *UI:110*
Contracting data *UI:302*
Convergence criterion *UII:1314*, *UII:1327*
default setting *UI:947*

in nonlinear least squares *UI:55*, *UI:60*
in pool estimation *UI:1087*
Convert
panel to pool *UI:309*
pool to panel *UI:315*
Copy *UI:302*
and paste *UI:118*, *UI:874*, *1:903*
and paste *See also* OLE.
by link *UI:303*
by value *UI:304*
command *UI:171*
data *UI:162*
data cut-and-paste *UI:148*
database *UI:351*
objects *UI:117*
pool objects *UI:1064*
table to clipboard *UI:874*
to and from database *UI:333*
to spool *UI:879*
Copy special *1:930*
Correlogram *UI:462*, *UI:465*, *UI:128*
ARMA models *UI:149*
autocorrelation function *UI:462*
cross *UI:655*
partial autocorrelation function *UI:463*
Q-statistic *UI:464*
squared residuals *UI:214*, *UI:274*
VAR *UI:849*
Count models *UI:407*
estimation *UI:408*
forecasting *UI:412*
negative binomial (ML) *UI:409*
Poisson *UI:408*
QML *UI:410*
residuals *UI:412*
Covariance
matrix, of estimated coefficients *UI:17*
matrix, systems *UI:807*
Covariance analysis *UI:616*
details *UI:624*
panel *UI:1219*
Covariance proportion *UI:179*
Cragg-Donald *UI:114*
Cramer's V *UI:632*
Cramer-von Mises test *UI:455*
Create
database *UI:327*
dated data table *UI:599*
factor *UI:1266*
graph *UI:827*
group *UI:133*, *UI:206*
link *UI:255*
objects *UI:105*
page *UI:82*
series *UI:124*, *UI:198*
spool *UI:877*
table *UI:865*
text object *UI:876*
workfile *UI:42*
Cross correlation *UI:655*
Cross correlogram *UI:655*
Cross section
pool identifiers *UI:1063*
specific series *UI:1065*
summaries *UI:1126*
SUR *UI:1103*
Cross section dependence test *UI:1177*, *UI:1239*
Cross validation
diagnostics *UI:549*
Cross-equation
coefficient restriction *UI:792*, *UI:796*
correlation *UI:793*, *UI:794*
weighting *UI:792*
CSV *UI:875*
C-test *UI:113*
Cubic
frequency conversion method *UI:173*, *UI:174*
CUE (continuously updating GMM) *See* Continuously updating GMM
Cumulative distribution *UI:772*
computation *UI:772*
Cumulative statistics
functions *UI:193*
Customization
graphs *UI:724*
CUSUM
sum of recursive residuals test *UI:246*
sum of recursive squared residuals test *UI:247*

D

Daniell kernel
cointegrating regression *UI:305*
GMM estimation *UI:110*
long-run covariance estimation *UI:658*

panel cointegrating regression *UI:1197*
robust standard errors *UI:45*
technical details *UII:1341*
Data
appending more *UI:299*
cut and paste *UI:150*, *UI:162*
duplicate observations *UI:458*
enter from keyboard *UI:146*, *1:961*
export *UI:161*, *UI:163*
Federal Reserve Economic data *UI:397*
FRED *UI:397*
import *UI:150*
import as matrix *UI:161*
import as table *UI:161*, *UI:876*
irregular *UI:271*
keyboard entry *UI:148*
pool *UI:1068*
regular *UI:271*
remove *UI:302*
structure *UI:271*
Database
alias *UI:340*
AREMOS *UI:355*
auto-search *UI:341*
auto-series *UI:339*
BEA *UI:360*
Bloomberg *UI:356*
BLS *UI:364*
cache *UI:432*
CEIC *UI:367*
copy *UI:351*
copy objects *UI:333*
create *UI:327*
data storage precision *UI:946*
Datastream *UI:375*
DBnomics *UI:375*
default *UI:330*
default in search order *UI:341*
delete *UI:351*
delete objects *UI:335*
display all objects *UI:328*
DRIBase *UI:382*
DRIPro *UI:382*
ECB SDMX *UI:417*
Economy.com *UI:411*
EcoWin *UI:383*
EIA *UI:387*
export *UI:333*
FactSet *UI:396*
FAME *UI:396*
fetch objects *UI:331*
field *UI:344*
foreign formats *UI:353*
FRED *UI:397*
frequency in query *UI:346*
group storage options *UI:946*
Haver Analytics *UI:401*
IHS Global Insight *UI:402*
IHS Magellan *UI:403*
IHSMarkit *UI:406*
IMF SDMX *UI:417*
link *UI:332*, *UI:333*, *UI:431*
link options *UI:433*
maintenance *UI:351*
match operator in query *UI:345*
NOAA *UI:412*
OECD SDMX *UI:417*
open *UI:327*
packing *UI:352*
previewing contents *UI:107*
queries *UI:341*
rebuild *UI:353*
registry *UI:955*
rename *UI:351*
rename object *UI:335*
repair *UI:353*
sharing violation *UI:328*
statistics *UI:352*
store objects *UI:330*
test integrity *UI:353*
UN SDMX *UI:417*
US Census *UI:368*
using auto-updating series with *UI:215*
window *UI:327*
World Bank *UI:425*
Database registry *UI:339*, *UI:955*
Datastream data *UI:375*
Date pairs *UI:137*
Date series *UI:224*
Dated data table *UI:598*
create *UI:599*
customization *UI:611*
customize *UI:599*
data format *UI:603*
example *UI:612*
fonts *UI:607*

formatting options *UI:606*
frequency conversion *UI:605*
headers *UI:609*
table options *UI:600*
templates *UI:611*
transformation methods *UI:604*
Dated import *UI:152*
Dates
default display format *UI:947*
display format *UI:224*
format in a spreadsheet *See* Display format
global options *UI:940*
match merging using *UI:247*
DBnomics *UI:375*
Decomposition
wavelet *UI:753*
Default
database *UI:9*, *UI:330*
database in search order *UI:341*
directory *UI:9*, *UI:953*
set directory *UI:95*
setting global options *UI:935*
update directory *UI:95*
window appearance *UI:936*
Delete *UI:119*
database *UI:351*
graph element *UI:843*
objects from database *UI:335*
observation in series *UI:132*
page *UI:91*
series using pool *UI:1080*
spool objects *UI:886*
Demonstration
estimation *UI:27*
examining data *UI:20*
forecasting *UI:34*
getting data into EViews *UI:17*
specification test *UI:30*
Den Haan and Levin *UII:1346*
Denoising *UI:766*, *UI:780*
Denton
frequency conversion method *UI:173*, *UI:175*
Dependent variable
no variance in binary models *UI:368*
Derivatives *UII:1317*, *UII:1330*
checking *UI:655*
default methods *UI:947*
description *UII:1330*
in equation *UI:19*
in logl *UI:648*
in system *UI:807*
saving in series *UII:1334*
Description
field in database query *UI:347*
Descriptive statistics *UI:191*
balanced sample (pool) *UI:1076*
by classification *UI:443*
by group *UI:443*
categorical graphs of *UI:794*
common sample (group) *UI:616*
common sample (pool) *UI:1076*
cross-section specific *UI:1077*
for a series *UI:445*
graphs of *UI:794*
group *UI:616*
individual samples (group) *UI:616*
individual samples (pool) *UI:1076*
pooled *UI:1076*
series *UI:440*
stacked data *UI:1077*
tests *UI:446*
time period specific *UI:1077*
Deselect all *UI:106*
Deterministic regressors *UI:297*
panel *UI:1193*
DFBetas *UI:251*
DFGLS *UI:673*
Dickey-Fuller test *UI:672*
See also Unit root tests.
Diebold-Mariano Test *UI:468*
Difference from moving-average *UI:537*
Difference operator *UI:192*, *UI:135*
seasonal *UI:193*, *UI:135*
Differencing
fractional *UI:125*
Discrete wavelet transform *UI:775*
Display filter *UI:61*, *UI:271*
Display format *UI:125*
group *UI:134*
Display mode
spools *UI:891*
Display name
field in database query *UI:347*
`distdata` *UI:585*

Distribution
  empirical distribution function tests *UI:455*
  tests *UI:455*
Distribution plot *UI:771*
  save data *UI:585*
DOLS *See* Dynamic OLS (DOLS)
Doornik and Hansen factorization matrix *UI:851*
Dot plot *UI:751*
Drag and drop
  existing file onto a new workfile *UI:89*
  existing file onto an existing workfile *UI:89*
  into a model *UI:1000*
  series into a group *UI:135*
  within the same workfile *UI:171*
Drag(ging)
  text in graph *UI:834*
DRI database
  DRIpro *UI:430*
  frequency *UI:434*
  illegal names *UI:433*
  object alias *UI:349*
  queries *UI:435*
  shadowing of object names *UI:350*
  troubleshooting *UI:435*
DRIBase database *UI:382*
DRIPro link *UI:382*
Dropbox *UI:96*
Dual processor *UI:954*
Dumitrescu-Hurlin test *UI:1231*
Dummy variables *UI:208*
  as binary dependent variable *UI:361*
  as censoring point in estimation *UI:390*
  automatic creation *UI:29*
  generating pool series dummies *UI:1075*
  pools *UI:1075*
  using @GROUP to create pool dummies *UI:1075*
Dunn-Sidak *UI:623*
Duplicate observations *UI:458*
Durbin-Watson statistic *UI:127*
  demonstration *UI:30*
  for regression *UI:14*
  lagged dependent variable *UI:128*
Durbin-Wu-Hausman test *UI:113*
Dynamic forecasting *UI:180*
Dynamic OLS (DOLS) *UI:295*, *UI:304*
  panel *UI:1196*, *UI:1208*
Dynamic panel data *UI:1139*
Dynamic switching models *UI:589*
Dynamic switching regression *UI:588*

E

Easy query *UI:342*
Economy.com *UI:411*
EcoWin database *UI:383*
Edit
  group *UI:135*
  series *UI:131*, *UI:593*
  table *UI:867*
EGARCH *UI:277*
  See also GARCH
EGLS (estimated GLS) *UI:1085*, *UI:1103*, *UI:1137*
EHS test *UI:113*
EIA *UI:387*
Eigenvalues
  factor analysis *UI:1280*
  plots *UI:642*
Elasticity at means *UI:196*
Elliot, Rothenberg, and Stock point optimal test *UI:675*
  See also Unit root tests.
Embedded spools *UI:880*
Empirical CDF
  graph *UI:772*
Empirical distribution tests *UI:455*
Empirical quantile graph *UI:774*
Empirical survivor graph *UI:773*
End field *UI:64*, *UI:346*
Endogeneity *UI:255*
  test of *UI:113*
Endogenous variables *UI:89*
  in models *UI:995*
Engle and Ng
  sign bias *UI:275*
Engle-Granger cointegration test *UI:1254*
Enterprise Edition *UI:356*, *UI:357*, *UI:367*, *UI:375*, *UI:382*, *UI:383*, *UI:396*, *UI:402*, *UI:411*
Epanechnikov kernel *UI:768*
Equality tests *UI:449*
  groups *UI:633*
  mean *UI:450*

median *UI:452*
variance *UI:454*
Equation *UI:5*
add to model *UI:1000*
automatic dummy variables in *UI:29*
coefficient covariance matrix *UI:17*
coefficient covariance scalar *UI:16*
coefficient p-values vector *UI:17*
coefficient standard error vector *UI:17*
coefficient t-statistic scalar *UI:17*
coefficient t-statistic vector *UI:17*
coefficient vector *UI:17*, *UI:20*
command string *UI:17*
create *UI:5*
derivatives *UI:19*
estimating in models *UI:999*
gradients *UI:19*
procedures *UI:19*
regression coefficients *UI:11*
regression summary statistics *UI:13*
residuals *UI:20*
results *UI:11*
retrieve previously estimated *UI:20*
r-squared *UI:13*
sample string *UI:17*
saved results *UI:16*
scalar results *UI:16*
specification *UI:6*
specification by list *UI:6*
specify by formula *UI:7*
specify with non-default coefs *UI:9*
specify with restrictions *UI:8*
specify without dependent variable *UI:8*
specifying a constant *UI:6*
store *UI:20*
text representation *UI:18*
t-statistic *UI:12*
updatetime *UI:17*
vector and matrix results *UI:17*
views *UI:18*
Ergodic probabilities *UI:588*, *UI:594*, *UI:949*
Error bar graph *UI:752*
Error-trend-seasonal smoothing
See ETS smoothing
Estimation *UI:9*
as part of model *UI:999*
auto-series *UI:205*
behavior *UII:1327*
binary dependent variable *UI:363*
censored models *UI:388*
collinearity *UI:22*
convergence *UII:1327*
convergence problems *UII:1315*, *UII:1316*
count models *UI:408*
demonstration *UI:27*
derivative computation options *UII:1317*
derivatives *UII:1330*
failure to improve *UII:1315*
for pool *UI:1082*
GLM *UI:423*
GMM *UI:101*
log likelihood *UI:650*
logl *UI:650*
missing data *UI:10*
multi-equation *UI:792*
near singular matrix problems *UII:1317*
nonlinear least squares *UI:51*
options *UII:1313*
ordered models *UI:381*
output *UI:11*
panel *UI:1135*
residuals from equation *UI:20*
robust regression *UI:459*
sample *UI:9*
sample adjustment *UI:10*
single equation methods *UI:9*
starting values *UII:1315*
state space *UI:973*, *UI:984*
systems *UI:792*, *UI:800*
truncated models *UI:397*
two-stage least squares *UI:89*
user-defined *I:899*
VAR *UI:835*, *UI:841*
VEC *UI:881*
ETS smoothing *UI:559*
AMSE based *UI:566*
example *UI:572*
forecast details *UI:568*, *UI:570*
initial states *UI:565*
MLE based *UI:566*
model selection *UI:567*, *UI:570*
output *UI:571*
parameters *UI:565*, *UI:570*
performing in EViews *UI:568*
specification *UI:569*
technical details *UI:560*

Eurostat SDMX *UI:417*
Evaluating forecasts *UI:466*
Evaluation order *UI:188*
  logl *UI:647*
EViews
  auto-update *UI:15*, *UI:955*
EViews Databases *UI:325*
EViews Enterprise Edition *UI:383*, *UI:396*, *UI:401*, *UI:411*
EViews Forum *UI:15*
Examining data
  demonstration *UI:20*
Excel
  Add-in *UI:164*
  reading EViews data in *UI:164*, *1:903*
Excel file
  importing data into matrix *UI:161*
  importing data into table *UI:161*, *UI:876*
  importing data into workfile *UI:150*
  linking data from *UI:55*
  opening as workfile *UI:48*
  opening as workfile demo *UI:17*
  saving as *UI:161*
Exogenous variable *UI:89*
  in models *UI:995*
  uncertainty *UI:1031*, *UI:1042*
Expectation-prediction table
  binary models *UI:370*
  ordered models *UI:384*
Expectations consistency in models *UI:1033*
Expected dependent variable
  censored models *UI:393*
  truncated models *UI:399*
Expected latent variable
  censored models *UI:393*
  truncated models *UI:399*
Exponential GARCH (EGARCH) *UI:277*
  See also GARCH
Exponential regression *UI:422*
Exponential smoothing *UI:553*, *UI:559*
  See also ETS smoothing
  See also Smoothing.
  double *UI:555*
  Holt-Winters additive *UI:556*
  Holt-Winters multiplicative *UI:556*
  Holt-Winters no seasonal *UI:557*
  single *UI:555*
Export *UI:163*, *UI:323*
  database *UI:333*
  pool data *UI:1080*
Expression *UI:187*
  for database fields *UI:345*
  parentheses *UI:188*
Extending EViews
  See Add-ins.
Extreme value
  binary model *UI:364*
  censored dependent variable models *UI:389*

F

Factor analysis *UI:1265*
  background *UI:1298*
  communalities *UI:1271*
  creation *UI:1266*
  data members *UI:1282*
  details *UI:1298*
  eigenvalues *UI:1280*
  example *UI:1282*
  factor selection *UI:1279*
  goodness of fit *UI:1278*, *UI:1303*
  graph scores *UI:1276*
  Kaiser's measure of sampling adequacy *UI:1281*
  loading views *UI:1279*
  method *UI:1267*, *UI:1269*
  method details *UI:1301*
  model evaluation *UI:1303*
  number of factors *UI:1279*
  PACE *UI:1269*
  procedures *UI:1281*
  reduced covariance *UI:1279*
  rotation *UI:1273*
  rotation (theory) *UI:1305*
  scaling *UI:1272*
  score estimation *UI:1274*
  specification *UI:1266*
  theory of *UI:1298*
  views *UI:1277*
Factor and graph layout options *UI:817*
Factor breakpoint test *UI:211*
Factor display settings *UI:813*
Factset data *UI:396*
Fair function *UI:452*
Fair-Taylor model solution *UI:1032*

FAME database *UI:396*
Federal Reserve Economic Data *UI:397*
Fetch *UI:120*
  from database *UI:331*
  from pool *UI:1080*
`fetch` *UI:171*
Fields in database *UI:344*
  description *UI:347*
  display_name *UI:347*
  end *UI:346*
  expressions *UI:345*
  freq *UI:346*
  history *UI:347*
  last_update *UI:347*
  last_write *UI:347*
  name *UI:345*
  remarks *UI:347*
  source *UI:347*
  start *UI:346*
  type *UI:345*
  units *UI:347*
Files
  default locations *UI:953*
  open session on double click *UI:938*
  opening/saving on a cloud location *UI:96*
Filter
  Hodrick-Prescott *UI:580*
  Markov switching *UI:586*
  state space models *UI:970*
  switching regression *UI:585*
  workfile objects *UI:73*
FIML *UI:794*
  system *UI:824*
First derivative methods *UII:1320*
Fisher-ADF *UI:715*
Fisher-Johansen *UI:1263*
Fisher-PP *UI:715*
Fit lines (graph) *UI:702*
Fitted index
  binary models *UI:375*
  censored models *UI:393*
  truncated models *UI:399*
Fitted probability
  binary models *UI:375*
Fitted values
  of equation *UI:18*
Fixed effects
  panel estimation *UI:1136*
  pool *UI:1085*
  pool description *UI:1100*
  test *UI:1166*
Fixed variance parameter
  negative binomial QML *UI:412*
  normal QML *UI:411*
Flatten
  spools *UI:887*
FMOLS *See* Fully modified OLS (FMOLS)
Fonts
  defaults *UI:940*
  tables *UI:871*
  text in graph *UI:833*, *UI:858*
Forecast
  AR specification *UI:182*
  ARIMA *UI:538*
  ARIMA using X-13 *UI:498*
  automatic with ARIMA models *UI:538*
  automatic with ETS smoothing *UI:568*, *UI:570*
  auto-series *UI:188*
  averaging *UI:547*
  backcasting *UI:183*
  binary models *UI:375*
  by exponential smoothing *UI:559*
  censored models *UI:393*
  Chow test *UI:242*
  combination testing *UI:467*
  combining *UI:547*
  conditional variance *UI:275*
  count models *UI:412*
  demonstration *UI:34*
  dynamic *UI:180*, *UI:972*
  equations with formula *UI:187*
  error *UI:175*
  ETS smoothing *UI:568*, *UI:570*
  evaluation *UI:466*, *UI:177*
  example *UI:170*
  expressions and auto-updating series *UI:187*
  fitted values *UI:174*
  from estimated equation *UI:167*
  GLM *UI:436*
  innovation initialization in models *UI:1033*
  interval *UI:177*
  lagged dependent variables *UI:180*
  MA specification *UI:183*
  missing values *UI:175*
  models *UI:1004*

nonlinear models *UI:193*
n-step ahead *UI:972*
n-step test *UI:248*
one-step test *UI:248*
ordered models *UI:386*
out-of-sample *UI:174*
PDLs *UI:193*
smoothed *UI:973*
standard error *UI:176*, *UI:190*
state space *UI:972*
static *UI:181*
structural *UI:182*
system *UI:808*
truncated models *UI:399*
VAR/VEC *UI:866*, *UI:883*
variance *UI:175*
with AR errors *UI:183*
Foreign data
import into workfile *UI:150*
open as workfile *UI:17*
Format
tables *UI:870*
Formula
forecast *UI:187*
implicit assignment *UI:199*
normalize *UI:200*
specify equation by *UI:7*
Forward solution for models *UI:1031*
Fractional difference
Specification *UI:136*
Fractional integration *UI:125*
Frame *UI:725*
size *UI:726*
FRED *UI:397*
Freedman-Diaconis *UI:762*
Freeze *UI:119*
create graph from view *UI:827*
Freq
field in database query *UI:346*
Frequency (Band-Pass) filter *UI:581*
Frequency conversion *UI:118*, *UI:170*, *UI:940*
Chow-Lin *UI:173*, *UI:176*
constant match *UI:173*
cubic *UI:173*, *UI:174*
dated data table *UI:605*
default settings *UI:177*
Denton *UI:173*
DRI database *UI:434*
high to low split observations *UI:172*
linear *UI:173*, *UI:174*
links *UI:262*
Litterman *UI:173*, *UI:176*
low to high *UI:173*
methods *UI:171*
panels *UI:253*
point *UI:173*, *UI:175*
propagate NAs *UI:172*
quadratic *UI:174*
quandratic *UI:173*
undated series *UI:177*
using links *UI:251*
Frequency spectrum *UI:151*
Frequency weights *UI:425*
Frequency zero spectrum estimation *UI:676*
F-statistic *UI:204*, *UI:209*
for regression *UI:15*
F-test
for variance equality *UI:454*
Full Information Maximum Likelihood *UI:824*
Full information maximum likelihood *UI:794*
Fully modified OLS (FMOLS) *UI:295*, *UI:297*
panel *UI:1194*, *UI:1205*
Functional coefficients regression *UI:557*

G

GARCH *UI:263*, *UI:273*, *UI:275*
ARCH-M model *UI:265*
asymmetric component model *UI:280*
backcasting *UI:269*
component models (CGARCH) *UI:279*
estimation in EViews *UI:266*
examples *UI:271*
exponential GARCH (EGARCH) *UI:277*
GARCH(1,1) model *UI:263*
GARCH(p,q) model *UI:265*
initialization *UI:269*
Integrated GARCH (IGARCH) *UI:276*
mean equation *UI:267*
multivariate *UI:664*
power ARCH (PARCH) *UI:278*
procedures *UI:275*
robust standard errors *UI:270*
sign bias test *UI:275*
test for *UI:218*

threshold (TARCH) *UI:277*
variance equation *UI:267*
Gauss file *UI:48*
Gauss-Newton *UII:1321*
Gauss-Seidel algorithm *UI:1043*, *UII:1322*
Generalized error distribution *UI:278*
Generalized least squares *See* GLS
Generalized linear models
example *UI:427*
forecasting *UI:436*
link function *UI:424*, *UI:441*
overview *UI:421*
performing in EViews *UI:423*
quasi-likelihood ratio test *UI:412*
residuals *UI:435*
robust standard errors *UI:426*
specification *UI:423*
technical details *UI:439*
testing *UI:437*
variance factor *UI:418*
Generalized method of moments, *See* GMM.
Generalized residual
binary models *UI:375*
censored models *UI:393*
count models *UI:412*
GLM *UI:435*
ordered models *UI:386*
score vector *UI:376*
truncated models *UI:399*
Generate series *UI:198*
by command *UI:200*
dynamic assignment *UI:199*
for pool *UI:1074*
implicit assignment *UI:199*
implicit formula *UI:199*
using samples *UI:198*
Geographic maps *UI:669*
Geomap *UI:669*
Geometric moving average *UI:205*
GiveWin data *UI:401*
Glejser heteroskedasticity test *UI:218*
GLM
See Generalized linear models.
Global breakpoint
estimation with *UI:472*
tests *UI:230*
Global optimum *UII:1317*
GLS
detrending *UI:673*
pool estimation details *UI:1101*
weights *UI:1137*
GMM *UI:101*, *UI:825*
bandwidth selection (single equation) *UI:110*
bandwidth selection (system) *UI:801*
breakpoint test *UI:116*
continuously updating (single equation) *UI:105*, *UI:110*
diagnostics *UI:112*
dropped instruments *UI:112*
estimate single equation by *UI:101*
estimate system by *UI:794*
HAC weighting matrix (single equation) *UI:110*
HAC weighting matrix (system) *UI:827*
instrument orthogonality test *UI:113*
instrument summary *UI:112*
iterate to convergence (single equation) *UI:105*, *UI:110*
J-statistic (single equation) *UI:102*
kernel options (single equation) *UI:110*
kernel options (system) *UI:801*
multi-equation *UI:794*
N-step (single equation) *UI:105*, *UI:110*
one-step (single equation) *UI:105*, *UI:110*
panels *UI:1139*
prewhitening option (single equation) *UI:110*
prewhitening option (system) *UI:802*, *UI:828*
regressor endogeneity test *UI:113*
robust standard errors *UI:106*
system *UI:825*
tests *UI:112*
user-specified weight matrix *UI:110*
weak instruments *UI:114*
White weighting matrix (single equation) *UI:110*
White weighting matrix (system) *UI:826*
Windmeijer standard errors *UI:107*
Godfrey heteroskedasticity test *UI:217*
Goldfeld-Quandt *UII:1320*
Gompit models *UI:364*
Goodness-of-fit
adjusted R-squared *UI:13*
Andrews test *UI:372*, *UI:419*
factor analysis *UI:1278*
forecast *UI:177*
Hosmer-Lemeshow test *UI:372*, *UI:418*

R-squared *UI:13*
Google Drive *UI:96*
Gradients *UII:1327*
details *UII:1327*
in equation *UI:19*, *UI:807*
in logl *UI:654*
saving in series *UII:1329*
summary *UII:1328*
Granger causality test *UI:661*
panel *UI:1230*
VAR *UI:848*
Graph
align multiple *UI:860*
analytical graphs *UI:761*
Animation *UI:710*
area band *UI:748*
area graph *UI:746*
arrows *UI:835*
automating *UI:864*
auto-updating *UI:828*
auxiliary graphs *UI:780*
average shifted histogram *UI:766*
axis borders *UI:687*
axis control *UI:848*
axis label format *UI:732*
axis *See also* Axis.
background color *UI:847*
background printing *UI:847*
bar graph *UI:746*
basic customization *UI:724*
border *UI:847*
boxplot *UI:777*
bubble plot *UI:753*
categorical *UI:847*
categorical *See also* Categorical graphs.
color settings *UI:847*
combining *UI:832*
combining graphs *UI:832*
confidence ellipse *UI:789*
coordinates for positioning elements *UI:833*
creating *UI:827*, *UI:828*
custom obs labels *UI:849*
customization *UI:832*
customize axis labels *UI:732*
customizing lines and symbols *UI:852*
data option *UI:685*
date label format *UI:734*
date label frequency *UI:733*
date label positioning *UI:736*
dot plot *UI:751*
drawing lines and shaded areas *UI:834*
empirical CDF *UI:772*
empirical log survivor *UI:773*
empirical quantile *UI:774*
empirical survivor *UI:773*
error bar *UI:752*
fill areas *UI:743*
first vs. all *UI:699*
fit lines *UI:702*
font *UI:833*
font options *UI:858*
frame *UI:725*
frame border *UI:726*
frame color *UI:725*
frame fill *UI:847*
freeze *UI:827*
freezing *UI:828*
frequency *UI:687*
grid lines *UI:847*
groups *UI:692*
high-low-open-close *UI:752*
histogram *UI:761*
histogram edge polygon *UI:765*
histogram polygon *UI:764*
identifying points *UI:681*
indentation *UI:847*
kernel density *UI:767*
kernel regression *UI:783*
legend *UI:739*
legend font *UI:741*
legend options *UI:851*
legend placement *UI:740*
legend settings *UI:851*
legend text *UI:741*
line formats *UI:741*
line graph *UI:745*
lines *UI:835*
link frequency *UI:687*
location *UI:727*
means *UI:685*
merging multiple *UI:106*
mixed frequency data *UI:696*
mixed line *UI:749*
modifying *UI:844*
multiple graph options *UI:860*
multiple series option *UI:694*

nearest neighbor regression *UI:785*
non-consecutive observations *UI:847*
observation graphs *UI:745*
observations to label *UI:733*
orientation *UI:686*
orthogonal regression *UI:788*
pairwise data *UI:698*
panel data *UI:1213*
panel data options *UI:689*
pie *UI:759*
place text in *UI:833*
position *UI:727*, *UI:860*
print in color *UI:862*
printing *UI:862*
quantile-quantile *UI:775*, *UI:776*, *UI:777*
raw data *UI:685*
regression line *UI:780*
remove custom date labels *UI:851*
remove elements *UI:843*
rotate *UI:686*
rotation *UI:733*
sample break plotting options *UI:847*
saving *UI:863*
scale *UI:733*
scatter *UI:753*
scatterplot matrix *UI:700*
scores *UI:1276*
seasonal *UI:759*
series *UI:683*
series view *UI:440*
settings for multiple graphs *UI:859*
shade options *UI:858*
size *UI:726*
slider bar (pasting with) *1:910*
sorting *UI:842*
sorting observations *UI:842*
spike *UI:749*
stacked *UI:699*
symbol graph *UI:745*
symbols *UI:741*
templates *UI:854*
text justification *UI:833*
text options *UI:858*
theoretical distribution *UI:771*
type *UI:684*, *UI:693*, *UI:745*, *UI:845*
update settings *UI:829*
XY area *UI:757*
XY bar *UI:758*
XY line *UI:756*
XY pairs *UI:699*
graph animation *UI:710*
Grid lines *UI:737*
table *UI:870*
Grid search *UII:1322*
Group *UI:206*, *UI:591*
add member *UI:591*
adding series *UI:592*
adding to *UI:135*
auto-series *UI:205*
create *UI:133*, *UI:206*
display format *UI:134*
display type *UI:125*
edit mode default *UI:944*
edit series *UI:593*
editing *UI:135*
element *UI:207*
graph view *UI:616*
graphing *UI:692*
make system of equations *UI:663*
number of series *UI:207*
pool *UI:1066*
rearranging series *UI:592*
row functions *UI:207*
spreadsheet view *UI:592*
spreadsheet view defaults *UI:944*
summaries *UI:598*
Group into bins option *UI:444*, *UI:629*
Group-mean DOLS *UI:1209*
Group-mean FMOLS *UI:1207*
Groupwise heteroskedasticity *UI:633*
Gumbel *UI:772*

H

HAC
cointegrating regression *UI:305*
GMM estimation *UI:110*
panel cointegrating regression *UI:1197*
robust standard errors *UI:32*, *UI:45*
system GMM *UI:827*
Hadri *UI:713*
Hannan-Quinn criterion *UII:1335*
for equation *UI:15*
Hansen instability test *UI:313*
Harvey heteroskedasticity test *UI:218*
Hat matrix *UI:251*

Hatanaka two-step estimator *UI:163*
Hausman test *UI:255*, *UI:1168*
Haver Analytics data *UI:401*
Heckit model*See* Heckman selection
Heckman selection *UI:401*
  example *UI:404*
  ML estimation *UI:402*
  performing in EViews *UI:403*
  two-step model *UI:401*
Heckman two-step *UI:401*
Help *UI:14*
  EViews Forum *UI:15*
  help system *UI:14*
  World Wide Web *UI:15*
Heteroskedasticity
  binary models *UI:379*
  cross-sectional details *UI:1102*
  groupwise *UI:633*
  of known form *UI:47*
  period details *UI:1102*
  robust standard errors *UI:32*
  tests of *UI:217*
  White's test *UI:219*
  wizard *UI:220*
Heteroskedasticity consistent covariances *UI:36*
Heywood cases *UI:1272*
Hide
  objects in spool *UI:883*
High frequency data *UI:45*
High-breakdown estimation *UI:456*
High-low-open-close graph *UI:752*
Hildreth-Lu *UI:163*
Histogram *UI:440*
  as axis *UI:687*
  average shifted graph *UI:766*
  bin width *UI:762*
  edge polygon graph *UI:765*
  graph *UI:761*
  normality test *UI:214*
  polygon graph *UI:764*
  save data *UI:585*
  variable width *UI:758*
Historical decomposition *UI:864*
History
  command window *UI:8*
  field in database query *UI:347*
Hodrick-Prescott filter *UI:580*, *UI:581*
Holt-Winters
  additive *UI:556*
  ETS framework *UI:562*
  likelihood based *UI:562*
  multiplicative *UI:556*
  no-seasonal *UI:557*
Honda random effects test *UI:1170*
Hosmer-Lemeshow test *UI:372*, *UI:418*
HTML *UI:875*
  open page as workfile *UI:48*
  save table as web page *UI:875*
Huber covariance *UI:458*
Huber function *UI:453*
Huber M-estimator *UI:451*, *UI:452*
Huber/White standard errors *UI:417*, *UI:426*
Hypothesis test
  stability *UII:519*, *UI:549*
Hypothesis tests
  See also Test.
  ARCH *UI:218*
  Bartlett test *UI:454*
  BDS independence *UI:466*, *UI:749*
  binomial sign test *UI:448*
  Brown-Forsythe *UI:455*
  chi-square test *UI:452*
  Chow breakpoint *UI:226*
  coefficient based *UI:196*
  coefficient p-value *UI:12*
  CUSUM *UI:246*
  CUSUM of squares *UI:247*
  demonstration *UI:30*
  descriptive statistic tests *UI:446*
  distribution *UI:455*
  F-test *UI:454*
  Hausman test *UI:255*
  heteroskedasticity *UI:217*
  irrelevant or redundant variable *UI:210*
  Kruskal-Wallis test *UI:453*
  Levene test *UI:455*
  mean *UI:446*
  median *UI:448*
  multi-sample equality *UI:449*
  nonnested *UI:257*
  normality *UI:214*
  omitted variables *UI:209*
  Ramsey RESET *UI:244*
  residual based *UI:213*
  Siegel-Tukey test *UI:454*

single sample *UI:446*
stability test *UI:225*
unit root *UI:465*, *UI:667*
unknown breakpoint *UI:228*
Van der Waerden test *UI:448*, *UI:453*
variance *UI:447*
Wald coefficient restriction test *UI:202*
White heteroskedasticity *UI:219*
Wilcoxon rank sum test *UI:452*
Wilcoxon signed ranks test *UI:448*

I

Icon *UI:103*
Identification
Box-Jenkins *UI:126*
GMM *UI:102*
nonlinear models *UI:59*
Identity
in model *UI:996*
in system *UI:797*
If condition in samples *UI:138*
IGARCH *UI:276*
IHS Global Insight data *UI:402*, *UI:403*
IHS Magellan data *UI:403*
Im, Pesaran and Shin *UI:714*
Import data *UI:146*
append to end *UI:157*
as matrix *UI:161*
as table *UI:161*, *UI:876*
dated read *UI:152*
for pool objects *UI:1068*
from ASCII *UI:150*
from spreadsheet *UI:150*
from workfile *UI:150*
matched read *UI:155*
name conflicts *UI:161*
options *UI:159*
See also Foreign data.
sequential read *UI:157*
using a pool object *UI:1072*
Impulse response *UI:853*
See also VAR.
ARMA models *UI:150*
generalized impulses *UI:856*
standard errors *UI:854*
structural decomposition *UI:856*
transformation of impulses *UI:855*
user specified impulses *UI:856*
Incorrect functional form *UI:219*, *UI:244*
Indentation
spools *UI:886*
Independence test *UI:466*, *UI:749*
Index
fitted from binary models *UI:375*
fitted from censored models *UI:393*
fitted from truncated models *UI:399*
Indicator saturation *UI:60*
Individual sample *UI:195*
Influence statistics *UI:251*
Information criterion
Akaike *UI:15*, *UII:1335*
Hannan-Quinn *UII:1335*
Schwarz *UI:15*, *UII:1335*
Innovation *UI:120*
Insert
observation *UI:132*
Insertion point *UI:7*
Instrumental variable *UI:89*
dropped instruments *UI:112*
for 2SLS with AR specification *UI:94*
for nonlinear 2SLS *UI:97*
identification (single equation) *UI:91*
identification (systems) *UI:799*
in systems *UI:797*
order condition *UI:91*
rank *UI:92*
summary of *UI:112*
tests *UI:112*
using PDL specifications *UI:25*
weak *UI:98*
weak instruments *UI:114*
with pooled data *UI:1105*
Integer dependent variable *UI:407*
Integrated series *UI:667*
Integrity (database) *UI:353*
Intercept in equation *UI:6*, *UI:12*
Interpolate *UI:484*
Intraday data *UI:45*
in samples *UI:140*
Invalid date identifiers *UI:295*
Inverted AR roots *UI:146*
Inverted MA roots *UI:146*
Irregular data *UI:271*

Irrelevant variable test *UI:210*
Iterate to convergence GMM
  single equation *UI:105*, *UI:110*
Iteration *UII:1314*
  failure to improve message *UII:1315*
  in models *UI:1046*
  in nonlinear least squares *UI:55*
  optimization method *UII:1314*

J

Jarque-Bera statistic *UI:442*, *UI:214*, *UI:274*
  in VAR *UI:850*
JPEG *UI:863*
J-statistic
  2sls *UI:92*
  GMM *UI:102*
  panel equation *UI:1160*
J-test *UI:257*

K

Kaiser's measure of sampling adequacy *UI:1281*
Kaiser-Guttman *UI:1299*
Kalman filter *UI:971*
Kao panel cointegration test *UI:1261*
K-class *UI:97*
  estimation of *UI:99*
Kendall's tau *UI:616*
  theory *UI:625*
Kernel
  cointegrating regression *UI:305*
  functions *UII:1341*
  GMM estimation *UI:110*
  graph *UI:784*
  long-run covariance estimation *UI:658*
  panel cointegrating regression *UI:1197*
  robust standard errors *UI:45*
  system GMM HAC *UI:801*, *UI:827*
  technical details *UII:1341*
Kernel density graph *UI:767*
  save data *UI:585*
Kernel functions *UI:768*
Kernel regression *UI:783*
  save data *UI:585*
Keyboard
  data entry *UI:146*
  focus option *UI:938*
Keyboard focus *UI:938*
Klein model
  GMM *UI:110*
  LIML *UI:100*
Kolmogorov-Smirnov test *UI:455*
KPSS unit root test *UI:675*
Kruskal-Wallis test *UI:453*
Kullback-Leibler *UII:1335*
Kurtosis *UI:442*
Kwiatkowski, Phillips, Schmidt, and Shin test
  UI:675

L

Label
  See Label object
Label object *UI:116*
  automatic update option *UI:941*
  capitalization *UI:116*
LAD *UI:619*
  output *UI:622*
  performing in EViews *UI:619*
  quantile process views *UI:625*
Lag
  dynamic assignment *UI:199*
  exclusion test *UI:848*
  forecasting *UI:180*
  panel data *UI:1119*
  series *UI:192*
Lag length
  VAR *UI:849*
Lag structure
  VAR *UI:848*
Lagged dependent variable
  and serial correlation *UI:119*
  Durbin-Watson statistic *UI:128*
Lagged dependent variable models
  See ARD
Lagged series in equation *UI:7*
Lagrange multiplier
  test for serial correlation *UI:128*
Large sample test *UI:195*
Last_update
  field in database query *UI:347*
Last_write
  field in database query *UI:347*
Latent variable

binary model *UI:362*
censored models *UI:387*
ordered models *UI:380*
LaTeX
save graph as *UI:863*
Lead
series *UI:192*
Least absolute deviations estimation *See* LAD
Least squares
panels *UI:1136*
See also Equation.
See also OLS.
Levene test *UI:455*
Leverage plots *UI:250*
Leverages
estimation in presence of high *UI:451*
Levin, Lin and Chu *UI:711*
Likelihood *UI:14*
Likelihood specification *UI:653*
Lilliefors test *UI:455*
Limit points *UI:383*
censored dependent variables *UI:389*
make covariance matrix *UI:386*
make vector *UI:386*
non-ascending *UI:384*
Limited dependent variable *UI:361*
Limited information maximum likelihood (LIML)
See LIML
LIML *UI:97*
Bekker standard errors *UI:99*
dropped instruments *UI:112*
estimation of *UI:99*
instrument summary *UI:112*
linear objective *UI:98*
minimum eigenvalue *UI:98*, *UI:101*
nonlinear objective *UI:98*
weak instruments *UI:114*
Line drawing *UI:834*
Line graph *UI:745*
Linear
frequency conversion method *UI:173*, *UI:174*
Linearity tests
smooth transition *UII:519*, *UI:549*
Link *UI:241*
basic concepts *UI:241*
breaking *UI:268*
create by command *UI:263*
create by copy-and-paste *UI:118*
creation *UI:255*
data from foreign formats *UI:48*, *UI:55*, *UI:150*, *UI:154*
frequency conversion *UI:251*, *UI:262*
match merging *UI:242*
modifying *UI:267*
to databases *UI:332*, *UI:333*
working with *UI:265*
Link function *UI:424*, *UI:441*
Linked equations in models *UI:1014*
Linking objects from Excel and other file formats *UI:48*, *UI:55*, *UI:150*, *UI:154*
Linking objects to Excel, Word *See* OLE.
List
specifying equation by *UI:6*
Litterman
frequency conversion method *UI:173*, *UI:176*
Ljung-Box Q-statistic *UI:464*
serial correlation test *UI:128*
LM test
ARCH *UI:218*
auxiliary regression *UI:215*, *UII:1330*
serial correlation *UI:128*, *UI:215*
LMMP test for random effects *UI:1170*
Lo and MacKinlay variance ratio test *UI:739*
Load
workfile *UI:81*
Loadings *UI:1279*
Local optimum *UII:1317*
Local regression *UI:786*
Local weighting option *UI:787*
LOESS *UI:786*, *UI:787*
Log likelihood
See also Logl.
average *UI:366*
censored models *UI:388*
exponential *UI:411*
for binary models *UI:362*
for regression (normal errors) *UI:14*
negative binomial *UI:409*
normal *UI:411*
ordered models *UI:381*
Poisson model *UI:409*
restricted *UI:366*
truncated models *UI:397*
Logical expression *UI:190*

in easy query *UI:343*
Logistic function *UI:453*
Logistic regression *UI:422*
Logit models *UI:364*
Logl *UI:643*
analytical derivatives *UI:648*
convergence *UI:655*
derivatives *UI:648*
errors *UI:656*
estimation *UI:650*
examples *UI:658*
gradients *UI:654*
limitations *UI:657*
order of evaluation *UI:647*
parameters *UI:646*
specification *UI:645*
starting values *UI:651*
step size *UI:649*
troubleshooting *UI:656*
views *UI:653*
Long name *UI:115*
for series *UI:474*
Long-run covariance *UII:1339*
cointegrating regression *UI:305*
GMM estimation *UI:110*
group *UI:655*
panel cointegrating regression *UI:1197*
series *UI:465*
technical discussion *UII:1339*
Long-run relationships
ARDL models *UI:322*
Long-run variance*See* Long-run covariance
LOWESS *UI:786*, *UI:787*
LR statistic *UI:209*, *UI:366*, *UI:395*, *UI:396*, *UI:412*
QLR *UI:416*

M

MA roots
inverted *UI:146*
MA specification
backcasting *UI:140*
forecast *UI:183*
in ARIMA models *UI:121*, *UI:134*
in model solution *UI:1033*
in two stage least squares *UI:95*
MADMED
definition *UI:454*
MADZERO
definition *UI:454*
MAE *UI:467*
Mallat's pyramid algorithm *UI:776*
Mann-Whitney test *UI:452*
MAPE *UI:467*
Maps
geographic *UI:669*
Marginal significance level *UI:12*, *UI:195*
Markdown
save graph as *UI:863*
Markov switching *UI:583*, *UI:585*
AR *UI:589*, *UI:590*
autoregressive models *UI:589*, *UI:590*
dynamic regression *UI:589*
estimation in EViews *UI:591*
example *UI:603*
expected durations *UI:599*, *UI:953*
filtering *UI:586*
initial probabilities *UI:588*, *UI:594*, *UI:949*
mean models *UI:589*
regime probabilities *UI:586*, *UI:601*, *UI:602*, *UI:942*, *UI:954*, *UI:956*
smoothing *UI:587*, *UI:946*
transition probabilities *UI:599*, *UI:953*
transition results *UI:599*, *UI:602*, *UI:953*, *UI:955*
VAR *UI:939*
views available *UI:599*, *UI:952*
Markov switching VAR
MSI-VAR(2) example *UI:956*
Marquardt *UII:1321*
Match merge *UI:242*
by date *UI:247*
many-to-many *UI:245*
many-to-one *UI:244*
one-to-many *UI:243*
panels *UI:249*
using links *UI:242*
Match operator in database query *UI:345*
Match-merge
as import *UI:155*
Matlab *I:899*
Maximization *See* Optimization (user-defined).
Maximum
number of observations *UI:953*

Maximum likelihood
See also Logl.
See also Optimization (user-defined).
full information *UI:794*
quasi-generalized pseudo-maximum likelihood *UI:415*, *UI:433*
quasi-maximum likelihood *UI:410*, *UI:421*
user specified *UI:643*
McFadden R-squared *UI:366*
Mean *UI:441*
equality test *UI:450*
hypothesis test of *UI:446*
Mean absolute error *UI:467*, *UI:178*
Mean absolute percentage error *UI:467*, *UI:178*
Mean equation (GARCH) *UI:267*
Mean square error *UI:548*, *UI:568*, *UI:178*
Measurement equation *UI:970*
Measurement error *UI:89*, *UI:244*
Median *UI:441*
equality test *UI:452*
hypothesis test of *UI:448*
Median function *UI:453*
Memory allocation *UI:953*
Memory, running out of *UI:953*
Menu *UI:114*
objects *UI:115*
Merge *UI:118*
See Match merge.
graphs *UI:106*
into panel workfiles *UI:1132*
store option *UI:331*
Messages *UI:935*
M-estimation *UI:451*
performing in EViews *UI:459*
tuning constants *UI:452*
weight functions *UI:452*
Meta data
See Attributes
Metafile
save graph as Windows metafile. *UI:863*
Micro TSP
opening workfiles *UI:323*
Microsoft Excel
See Excel.
Microsoft PowerPoint
pasting graphs and data into *1:903*
Microsoft Word
pasting graphs and data into *1:903*
MIDAS
Almon weighting *UI:343*
beta weighting *UI:344*, *UI:345*
example *UI:353*
exponential Almon weighting *UI:344*
PDL weighting *UI:343*
regression *UI:341*
step weighting *UI:343*
Minimization *See* Optimization (user-defined).
Minimum discrepancy *UI:1301*
Missing values *UI:195*
forecasting *UI:175*
handling in estimation *UI:10*
in frequency conversion *UI:172*
in models *UI:1046*
in observation graphs *UI:688*
interpolate *UI:484*
recoding *UI:197*
relational comparisons involving *UI:196*
test *UI:196*
MIsspecification test
sign bias *UI:275*
Misspecification test
conditional variance *UI:275*
misspecification test *UI:275*
Mixed data sampling
regression *UI:341*
Mixed frequency graph *UI:696*
Mixed frequency VAR *UI:927*
Mixed line graph *UI:749*
MLE
See Logl
MM-estimation *UI:458*
performing in EViews *UI:459*
Model
excluding variables *UI:1027*
overriding variables *UI:1026*
Model averaging *UI:547*
Model consistent expectations *UI:1033*
Model selection
ARDL models *UI:328*
ARIMA models *UI:538*
ARIMA using X-13 *UI:494*, *UI:497*
ETS smoothing *UI:567*, *UI:570*
TAR estimation *UI:498*

Models
add factors *UI:995*, *UI:1008*, *UI:1028*
adding equations *UI:1000*
aliasing *UI:997*, *UI:1025*
binding variables *UI:997*
block structure *UI:1018*
boundaries *UI:1043*
Broyden solution *UII:1324*
Broyden solver *UI:1044*
coefficient uncertainty *UI:1015*, *UI:1031*, *UI:1041*
comparing solutions *UI:1055*
convergence test *UI:1046*
creating *UI:1013*
definition *UI:791*
demonstration *UI:377*
dependency graph *UI:1022*
derivatives *UI:1046*
diagnostic messages and iteration history *UI:1042*
dynamic solution *UI:1036*
dynamic solve *UI:1003*
editing scenarios *UI:1026*
endogenous variables *UI:995*
equation view *UI:1015*
estimating equations *UI:999*
excluding variables *UI:1024*
exogenous variable *UI:995*
exogenous variable uncertainty *UI:1031*, *UI:1042*
Fair-Taylor solution *UI:1032*
fit option for solution *UI:1037*
forecasting with *UI:1004*
future values *UI:1031*
Gauss-Seidel solution *UII:1322*
Gauss-Seidel solver *UI:1043*
handling of ARMA terms *UI:1037*
identities *UI:996*
initialize excluded variables *UI:1044*
inline equations *UI:1014*
intercept shift add factor *UI:1028*
linked equations *UI:1014*
MA error terms *UI:1033*
missing value handling *UI:1046*
Newton solution *UII:1323*
Newton's method *UI:1044*
overriding variables *UI:997*, *UI:1025*, *UI:1031*
properties of equations *UI:1016*
roundoff of solution *UI:1047*
scenarios *UI:377*, *UI:1011*, *UI:1024*
scenarios (example) *UI:377*
simultaneous and recursive blocks *UI:1018*
solution methods *UI:1043*
solve (dynamic) *UI:1003*
solve (static) *UI:1001*
solving *UI:1030*
solving to match target *UI:1047*
starting values *UI:1046*
static solution *UI:1037*
static solve *UI:1001*
stochastic equations *UI:996*
stochastic simulation *UI:1036*
stochastic solution *UI:1038*
text description of *UI:1019*, *UI:1020*
text keywords *UI:1019*, *UI:1020*
tracking variables *UI:1042*
updating links *UI:1015*
variable dependencies *UI:1018*
variable shift add factor *UI:1028*
variable view *UI:1017*
Moment condition *UI:102*
Moment selection criteria *UI:115*
Moody's Economy.com data *UI:411*
MoveReg decomposition *UI:514*
Moving statistics
functions *UI:193*
geometric mean *UI:205*
MSAR *UI:589*, *UI:590*
MSE *UI:548*, *UI:568*, *UI:178*
MSI *UI:589*
MSM *UI:589*
Multicollinearity *UI:22*
coefficient variance decomposition *UI:200*
test of *UI:199*, *UI:200*
Multiple processors *UI:954*
Multiresolution analysis *UI:777*
Multivariate ARCH *UI:794*

N

NA *See* NAs *and* Missing data.
Nadaraya-Watson *UI:784*
Name
object *UI:115*
reserved *UI:115*
Name field in database query *UI:345*

Naming objects
spool *UI:882*
NAs *UI:195*
forecasting *UI:175*
inequality comparison *UI:196*
See also Missing data
test *UI:196*
National Oceanic and Atmospheric Administration data *UI:412*
Near singular matrix *UI:22*
binary models *UI:369*
logl *UI:647*, *UI:655*, *UI:657*
nonlinear models *UI:59*, *UII:1317*
polynomial distributed lag *UI:24*
RESET test *UI:245*
Nearest neighbor regression *UI:785*, *UI:786*
Negative binomial count model *UI:409*
Network proxy server *UI:952*
Newey-West automatic bandwidth
cointegrating regression *UI:305*
GMM estimation *UI:110*
long-run covariance estimation *UI:658*, *UII:1345*
panel cointegrating regression *UI:1197*
system GMM *UI:827*
Newey-West consistent covariance
cointegrating regression *UI:305*
GLM estimation *UI:426*
GMM estimation *UI:110*
panel cointegrating regression *UI:1197*
robust standard errors *UI:45*
system GMM *UI:827*
Newton's method *UI:1044*, *UII:1323*
Newton-Raphson *UII:1319*, *UII:1321*
NOAA data *UI:412*
Noninvertible MA process *UI:146*, *UI:158*
Nonlinear coefficient restriction
Wald test *UI:207*
Nonlinear least squares *UI:51*
coefficient covariance *UI:54*
convergence criterion *UI:55*
forecast standard errors *UI:177*
iteration option *UI:55*
optimization method option *UI:55*
specification *UI:53*
starting values *UI:56*
two stage *UI:96*
two stage with AR specification *UI:97*
weighted *UI:58*
weighted two stage *UI:97*, *UI:108*
with AR specification *UI:58*, *UI:159*
Nonnested tests *UI:257*
Nonparametric kernel
technical details *UII:1340*
Non-unique identifiers *UI:295*
Normality test *UI:442*, *UI:455*, *UI:214*, *UI:274*, *UI:850*
VAR *UI:850*
Normalize formula *UI:200*
Nowcasting *UI:341*
N-step forecast test *UI:248*
N-step GMM
single equation *UI:105*, *UI:110*
Null hypothesis *UI:195*
Number format
See Display format
Numbers
relational comparison *UI:190*
N-way table *UI:633*
chi-square tests *UI:631*
Nyblom *UI:273*
Nyblom test *UI:273*

O

Object *UI:101*
allow multiple untitled *UI:937*
basics *UI:102*
closing untitled *UI:937*
copy *UI:117*
create *UI:105*
data *UI:102*, *UI:123*
delete *UI:119*
freeze *UI:119*
icon *UI:103*
label *See* Label object
naming *UI:116*
open *UI:106*
preview *UI:107*, *UI:333*
print *UI:119*
procedure *UI:103*
sample *UI:145*
show *UI:107*
store *UI:120*
type *UI:104*

window *UI:112*
Object linking and embedding
See OLE.
Objects menu *UI:115*
Observation equation *UI:970*, *UI:975*
Observation graphs *UI:688*, *UI:745*
missing values *UI:688*
Observation identifiers *UI:312*
Observation number *UI:128*
Observation scale *UI:733*
Observations, number of
maximum *UI:953*
ODBC *UI:48*
OLE *1:903*
copy special *1:930*
embedding (definition) *1:904*
linking (definition) *1:904*
paste EViews object *1:908*
pasting graphs *1:905*
pasting numerical data *1:919*
pasting with the workfile sample *1:926*
using *1:904*
OLS (ordinary least squares)
See also Equation.
adjusted R-squared *UI:13*
coefficient standard error *UI:12*
coefficient t-statistic *UI:12*
coefficients *UI:11*
standard error of regression *UI:14*
sum of squared residuals *UI:14*
system estimation *UI:792*, *UI:821*
Omitted variables test *UI:209*, *UI:244*
panel *UI:1162*
OneDrive *UI:96*
One-step forecast test *UI:248*
One-step GMM
single equation *UI:105*, *UI:110*
One-way frequency table *UI:458*
Open
database *UI:327*
foreign data as matrix *UI:161*
foreign data as table *UI:161*, *UI:876*
multiple objects *UI:106*
object *UI:106*
options *UI:323*
workfile *UI:81*
Operator *UI:187*
arithmetic *UI:187*
conjunction (and, or) *UI:191*
difference *UI:192*
lag *UI:192*
lead *UI:192*
parentheses *UI:188*
relational *UI:190*
Optimization
methods *UII:1314*
Optimization algorithms
BHHH *UII:1321*
first derivative methods *UII:1320*
Gauss-Newton *UII:1321*
Goldfeld-Quandt *UII:1320*
grid search *UII:1322*
Marquardt *UII:1321*
Newton-Raphson *UII:1319*
second derivative methods *UII:1319*
starting values *UII:1315*
step size *UII:1321*
Option settings
allow only one untitled *UI:937*
backup workfiles *UI:78*, *UI:946*
date notation *UI:940*
default fonts *UI:940*
EViews sessions on open *UI:938*
external program interface *UI:951*
fonts *UI:940*
frequency conversion *UI:940*
keyboard focus *UI:938*
messages *UI:935*
network proxy server *UI:952*
print setup *UI:956*
program execution mode *UI:948*
series auto label *UI:941*
snapshots *UI:939*
spreadsheet data display *UI:944*
spreadsheet view defaults *UI:943*
start page *UI:935*
warn on close *UI:937*
window appearance *UI:936*
Optmization (user-defined) *I:899*
Or operator *UI:139*, *UI:191*
Order condition
2sls *UI:91*
GMM *UI:102*
Order of evaluation
logl *UI:647*

Order of stacked data *UI:1071*
Ordered dependent variable *UI:380*
  error messages *UI:384*
  estimation *UI:381*
  expectation-prediction tables *UI:384*
  forecasting *UI:386*
  limit points *UI:386*
  log likelihood *UI:381*
  variable frequencies *UI:384*
  views *UI:384*
Ordinary residual
  binary models *UI:375*
  censored models *UI:393*
  count models *UI:412*
  GLM *UI:435*
  truncated models *UI:398*
Orientation *UI:686*
Orthogonal regression *UI:788*
Orthogonality condition *UI:102*, *UI:826*
Ourlier analysis *UI:60*
Outlier detection
  wavelets *UI:769*, *UI:779*
Outliers
  detection of *UI:250*, *UI:251*
  detection of in X-13 *UI:492*
  robust estimation in presence of *UI:451*
Over identification *UI:102*
Overdispersion *UI:410*, *UI:418*, *UI:447*
  specification test *UI:413*

P

PACE *UI:1269*
  details *UI:1303*
Pack database *UI:352*
Packable space *UI:328*, *UI:352*
Page
  create new *UI:82*
  delete page *UI:91*
  rename *UI:91*
  reorder *UI:91*
Page breaks *UI:895*
Pairwise graphs *UI:698*
Panel
  random components test *UI:1170*
  residual cross-section dependence test
    UI:1177, *UI:1239*
  unit root tests *UI:716*
Panel cointegrating regression *UI:1191*
  equation specification *UI:1193*
  examples *UI:1199*
  performing in EViews *UI:1192*
  PMG models *UI:1143*
  technical details *UI:1191*, *UI:1205*
Panel data *UI:1111*
  analysis *UI:1133*
  balanced *UI:278*
  cell identifier *UI:1117*
  cointegration testing *UI:1238*, *UI:1258*
  convert to pool *UI:309*
  covariance analysis *UI:1219*
  create workfile of *UI:46*
  cross-section identifiers *UI:1116*
  cross-section summaries *UI:1126*
  dated *UI:277*
  duplicate identifiers *UI:276*, *UI:293*
  dynamic panel data *UI:1139*
  estimation *See* Panel estimation.
  fixed effects test *UI:1166*
  frequency conversion *UI:253*
  GMM estimation *UI:1139*
  graphs *UI:1213*
  group identifier *UI:1116*
  Hausman test *UI:1168*
  identifiers *UI:274*
  instrumental variables estimation *UI:1138*
  irregular *UI:278*
  lags *UI:275*, *UI:1119*
  lags and leads *UI:1119*
  least squares estimation *UI:1136*
  merging data into *UI:1132*
  nested *UI:280*
  period summaries *UI:1126*
  pool comparison *UI:1061*
  regular *UI:278*
  samples in panel workfiles *UI:1120*
  See also Panel workfile.
  statistics *UI:1124*
  testing *UI:1162*
  time trend *UI:1124*
  trends *UI:1124*
  unbalanced *UI:278*
  undated *UI:277*
  unit root tests *UI:705*, *UI:1234*
  within-group identifier *UI:1118*
  workfile structure *UI:274*

Panel estimation *UI:1135*
examples *UI:1146*
GLS weights *UI:1137*
GMM *UI:1139*
GMM (example) *UI:1155*
GMM details *UI:1185*
least squares *UI:1136*
PMG *UI:1143*
TSLS *UI:1138*
Panel unit root *See* Panel data - unit root tests.
Panel vs. pool *UI:1061*
Panel workfile
See also Panel data.
create *UI:1111*
dated *UI:288*
display *UI:1114*
nested *UI:280*
structure *UI:1111*, *UI:1116*
undated *UI:293*
undated with ID *UI:292*
Parallel analysis *UI:1270*
Param (command) *UI:57*, *UI:800*, *UII:1316*
Parameters
logl *UI:646*
PARCH *UI:278*
Park added variable test *UI:316*
Parks estimator *UI:1103*
Partial analysis *UI:621*
Partial autocorrelation *UI:463*, *UI:126*
Partial covariance analysis *UI:621*
Parzen kernel
cointegrating regression *UI:305*
GMM estimation *UI:110*
long-run covariance estimation *UI:658*
panel cointegrating regression *UI:1197*
robust standard errors *UI:45*
technical details *UII:1341*
Parzen-Cauchy kernel
cointegrating regression *UI:305*
GMM estimation *UI:110*
long-run covariance estimation *UI:658*
panel cointegrating regression *UI:1197*
robust standard errors *UI:45*
technical details *UII:1341*
Parzen-Geometric kernel
cointegrating regression *UI:305*
GMM estimation *UI:110*
long-run covariance estimation *UI:658*
panel cointegrating regression *UI:1197*
robust standard errors *UI:45*
technical details *UII:1341*
Parzen-Riesz kernel
cointegrating regression *UI:305*
GMM estimation *UI:110*
long-run covariance estimation *UI:658*
panel cointegrating regression *UI:1197*
robust standard errors *UI:45*
technical details *UII:1341*
Paste *UI:118*
data as new workfile *UI:47*
existing series *UI:149*
into Excel, Word *See* OLE.
new series *UI:148*
Paste special
See also OLE.
EViews object *1:908*
graphs *1:906*
spreadsheets *1:911*
tables *1:911*
Paste special *See* OLE.
PcGive data *UI:401*
PDF
save graph as *UI:863*
PDL
mixed frequencies *UI:343*
PDL (polynomial distributed lag) *UI:23*, *UI:177*
far end restriction *UI:24*
forecast standard errors *UI:177*
instrumental variables *UI:25*
near end restriction *UI:24*
specification *UI:25*
Pearson covariance *UI:616*
Pedroni panel cointegration test *UI:1239*, *UI:1260*
Period
summaries *UI:1126*
SUR *UI:1104*
Perron unit root test *UI:679*
Pesaran scaled LM test *UI:1177*, *UI:1239*
Pesaran, Shin and Smith *UI:1143*
Phillips-Ouliaris cointegration test *UI:1254*
Phillips-Perron test *UI:674*
Pie graph *UI:759*
PMG *UI:1143*

PNG *UI:863*
Point
  frequency conversion method *UI:173*, *UI:175*
Poisson count model *UI:408*
Polynomial distributed lags, *See* PDL.
Pool *UI:1061*
  ? placeholder *UI:1066*
  and cross-section specific series *UI:1065*
  AR specification *UI:1084*
  balanced data *UI:1072*, *UI:1076*
  balanced sample *UI:1083*
  base name *UI:1065*
  coefficient test *UI:1096*
  cointegration *UI:1078*
  common coefficients *UI:1084*
  convergence criterion *UI:1087*
  convert to panel *UI:315*
  copy *UI:1064*
  creating *UI:1067*
  cross-section *UI:1063*
  cross-section specific coefficients *UI:1084*
  defining *UI:1063*
  defining groups of identifiers *UI:1064*
  descriptive statistics *UI:1076*
  dummy variable *UI:1075*
  editing definitions *UI:1064*
  estimation *UI:1082*
  estimation details *UI:1097*
  export data *UI:1080*
  fixed effects *UI:1085*, *UI:1100*
  generate series *UI:1074*
  group *UI:1066*
  import *UI:1068*
  import data *UI:1068*
  import stacked data *UI:1072*
  instrumental variables *UI:1088*, *UI:1105*
  make group *UI:1079*
  make system *UI:1079*
  naming series *UI:1065*
  object *UI:1062*
  options *UI:1086*
  order *UI:1071*
  period-specific coefficients *UI:1084*
  pool series *UI:1066*
  procedures *UI:1096*
  random effects *UI:1085*, *UI:1101*
  residuals *UI:1097*
  restructure *UI:1070*
  series *UI:1066*
  setup *UI:1067*
  special group identity series *UI:1066*
  specification *UI:1063*
  stacked data *UI:1069*
  tests *UI:1096*
  unstacked data *UI:1068*
  workfile *UI:1061*
Pool data
  panel comparison *UI:1061*
Pool vs. panel *UI:1061*
Pooled Mean Group estimation *UI:1143*
Portmanteau test
  VAR *UI:850*
PostScript *UI:863*
  save graph as PostScript file *UI:863*
PowerPoint
  pasting graphs and data into *1:903*
Prais-Winsten *UI:163*
Precedence of evaluation *UI:188*
Predetermined variable *UI:89*
Prediction table
  binary models *UI:370*
  ordered models *UI:384*
Preview objects *UI:107*, *UI:333*
Prewhitening
  cointegrating regression *UI:305*
  GMM estimation *UI:110*
  long-run covariance estimation *UII:1347*
  panel cointegrating regression *UI:1197*
  robust standard errors *UI:45*
  system GMM *UI:802*, *UI:828*
  technical details *UII:1347*
Principal components *UI:634*
Principal factors *UI:1302*
Print
  graphs *UI:862*
  mode *UI:895*
  objects *UI:119*
  settings *UI:956*
  setup options *UI:956*
  spool *UI:895*, *UI:957*
  tables *UI:874*
  to a spool *UI:878*
Probability response curve *UI:376*
Probit models *UI:364*
Procedures *UI:103*

Process graphs (quantile regression) *UI:626*
Processors
  multiple *UI:954*
Program
  auto indent *UI:948*
  backup files *UI:948*
  execution option *UI:948*, *UI:951*, *UI:952*
  syntax coloring *UI:948*
  tab settings *UI:948*
Proxy server *UI:952*
P-value *UI:195*
  for coefficient t-statistic *UI:12*
Pyramid algorithm *UI:776*

Q

QML *UI:410*, *UI:421*, *UI:443*
QQ-plot *UI:775*, *UI:776*, *UI:777*
  save data *UI:585*
Q-statistic
  Ljung-Box *UI:464*
  residual serial correlation test *UI:850*
  serial correlation test *UI:128*
Quadratic
  frequency conversion method *UI:173*, *UI:174*
Quadratic hill-climbing *UII:1320*
Quadratic spectral kernel *UI:827*
  cointegrating regression *UI:305*
  GMM estimation *UI:110*
  long-run covariance estimation *UI:658*
  panel cointegrating regression *UI:1197*
  robust standard errors *UI:45*
  technical details *UII:1342*
Qualitative dependent variable *UI:361*
Quandt breakpoint test *UI:228*
Quantile method *UI:772*
Quantile process coefficients *UI:626*
  technical details *UI:640*
Quantile process views *UI:625*
  technical details *UI:640*
Quantile regression *UI:619*
  example *UI:622*
  output *UI:622*
  performing in EViews *UI:619*
  process coefficients *UI:626*
  quantile process views *UI:625*
  slope equality test *UI:628*
  symmetric quantiles test *UI:629*
  technical details *UI:631*
Quantiles
  from series *UI:477*, *UI:479*
Quasi-generalized pseudo-maximum likelihood *UI:415*
Quasi-likelihood ratio test *UI:412*, *UI:416*
Quasi-maximum likelihood *UI:410*
  robust standard errors *UI:417*
Queries on database *UI:341*
  advanced query *UI:343*
  DRI *UI:435*
  easy query *UI:342*
  examples *UI:348*
  logical expressions *UI:343*
  wildcard characters *UI:342*
Quiet mode *UI:949*

R

R *I:899*
R project *I:899*
Ramsey RESET test *UI:244*
Random components test *UI:1170*
Random effects
  LM test for *UI:1170*
  panel estimation *UI:1136*
  pool *UI:1085*
  pool descriptions *UI:1101*
  test for correlated effects (Hausman) *UI:1168*
Random walk *UI:667*
Rank condition for identification *UI:91*
Ranks
  observations in series or vector *UI:194*
Ratio to moving-average *UI:537*
RATS data
  4.x native format *UI:421*
  portable format *UI:422*
Read *UI:1068*
  data from foreign file as matrix *UI:161*
  data from foreign file as table *UI:161*, *UI:876*
Reading EViews data (in other applications) *UI:164*
Rebuild database *UI:353*
Recursive coefficient *UI:249*
  save as series *UI:249*
Recursive estimation
  least squares *UI:245*

using state space *UI:978*
Recursive least squares *UI:245*
Recursive residual *UI:245*, *UI:246*
CUSUM *UI:246*
CUSUM of squares *UI:247*
n-step forecast test *UI:248*
one-step forecast test *UI:248*
save as series *UI:249*
Reduced covariance *UI:1279*
Redundant variables test *UI:210*
panel *UI:1164*
Regime probabilities *UI:601*, *UI:954*
outputting *UI:602*, *UI:956*
Regime switching *UI:584*
Registry *UI:339*
Regression
See also Equation.
adjusted R-squared *UI:13*
breakpoint estimation *UI:471*
coefficient standard error *UI:12*
coefficients *UI:11*
collinearity *UI:22*
forecast *UI:167*
F-statistic *UI:15*
functional coefficients *UI:557*
line on graph *UI:780*
log likelihood *UI:14*
quantile *UI:619*
residuals from *UI:20*
standard error of *UI:14*
sum of squared residuals *UI:14*
t-statistic for coefficient *UI:12*
Regular data *UI:271*
Relational operators
and missing values *UI:196*
Remarks
field in database query *UI:347*
Removing data *UI:302*
Rename *UI:115*
database *UI:351*
objects in database *UI:335*
page *UI:91*
workfile page *UI:91*
Reorder
page *UI:91*
Repair database *UI:353*
Representations view
equation *UI:18*
Resample *UI:482*
Reserved names *UI:115*
RESET test *UI:244*
Reshaping a workfile *UI:306*
Residuals
binary models *UI:375*
censored dependent variable models *UI:392*
count models *UI:412*
default series RESID *UI:20*
display of in equation *UI:20*
estimation in presence of large *UI:451*
from estimated equation *UI:20*
from two stage least squares *UI:92*
generalized *UI:375*, *UI:393*, *UI:399*, *UI:412*
GLM *UI:435*
make series or group containing *UI:19*
of equation *UI:18*
ordinary *UI:375*, *UI:393*, *UI:398*, *UI:412*
plot *UI:18*
plots of *UI:250*
pool *UI:1097*
recursive *UI:245*, *UI:246*
standardized *UI:18*, *UI:375*, *UI:393*, *UI:398*, *UI:412*
studentized *UI:251*
sum of squares *UI:14*
symmetrically trimmed *UI:395*
system *UI:809*
tests of *UI:213*
truncated dependent variable *UI:398*
unconditional *UI:145*
Resize
spools *UI:885*
table columns and rows *UI:870*
workfile *UI:283*, *UI:296*
Restricted estimation *UI:8*
Restricted log likelihood *UI:366*
Restructuring *UI:1070*
Results
display or retrieve *UI:16*
Rich Text Format *UI:875*
RMSE *UI:467*, *UI:178*
Rn-squared statistic
definition *UI:455*
Robust least squares *UI:451*
Andrews function *UI:452*

Bisquare function *UI:452*
Cauchy function *UI:452*
example *UI:464*
Fair function *UI:452*
Huber function *UI:453*
Logistic function *UI:453*
Median function *UI:453*
M-estimation *UI:451*
Talworth function *UI:453*
Welsch function *UI:453*
Robust regression *See* Robust least squares.
Robust standard errors *UI:32*
Bollerslev-Wooldridge for GARCH *UI:270*
cluster *UI:39*
clustered *UI:1137*
GLM *UI:417*, *UI:426*
GMM *UI:106*
Huber-White (QML) *UI:417*, *UI:426*
Robustness iterations *UI:782*, *UI:787*
Root mean square error *UI:467*, *UI:178*
Rotate
factors *UI:1273*, *UI:1305*
graphs *UI:686*
Rotation of factors *UI:1273*
details *UI:1305*
Row
functions *UI:207*
height *UI:870*
R-squared
adjusted *UI:13*
for regression *UI:13*
from two stage least squares *UI:93*
McFadden *UI:366*
negative *UI:272*
uncentered *UI:215*, *UI:219*
with AR specification *UI:145*
RTF *UI:875*
create *UI:957*
redirecting print to *UI:957*
Rw-squared statistic
definition *UI:455*

S

SAIC *UI:549*
Sample
`@all` *UI:138*
`@first` *UI:138*
adjustment in estimation *UI:10*
all observations *UI:138*
balanced *UI:1083*
breaks *UI:688*
change *UI:137*
command *UI:139*
common *UI:195*
current *UI:61*
date pairs *UI:137*
first observation *UI:138*
if condition *UI:138*
individual *UI:195*
intraday data *UI:140*
last observation *UI:138*
range pairs *UI:137*
selection and missing values *UI:139*
specifying sample object *UI:145*
specifying samples in panel workfiles *UI:1120*
used in estimation *UI:9*
using sample objects in expressions *UI:145*
with expressions *UI:140*
workfile *UI:136*
SAR specification *UI:123*, *UI:126*
SAR(p)
estimation *UI:133*
SARMA *UI:123*
SAS file *UI:48*
Save
backup workfile *UI:76*
graphs *UI:863*
options *UI:323*
save as new workfile *UI:76*
spool *UI:896*
tables *UI:875*
workfile *UI:75*
workfile as foreign file *UI:161*
workfile precision and compression *UI:77*
Scalar *UI:210*
Scale factor *UI:392*
Scaled coefficients *UI:196*
Scaling
factor analysis *UI:1272*
Scatterplot *UI:753*
categorical *UI:805*
matrix of *UI:700*
with confidence ellipse *UI:789*
with kernel regression fit *UI:783*
with nearest neighbor fit *UI:785*

with orthogonal regression line *UI:788*
with regression line *UI:780*
Scenarios *UI:1011*
simple example *UI:377*
Schwarz criterion *UII:1335*
for equation *UI:15*
Score coefficients *UI:1275*
Score vector *UI:376*
Scores *UI:1274*
SDMX data *UI:422*
Seasonal
ARMA terms *UI:123*
difference *UI:193*, *UI:135*
graphs *UI:759*
Seasonal adjustment *UI:487*
additive *UI:537*
Census X-11 (historical) *UI:532*
Census X-12 *UI:523*
Census X-13 *UI:487*
MoveReg decomposition *UI:514*
multiplicative *UI:537*
STL decomposition *UI:506*
Tramo/Seats *UI:532*
weekly *UI:514*
Seasonal unit root test *UI:695*
Seasonal unit roots *UI:695*
Second derivative methods *UII:1319*
Second-generation panel unit r oot tests *UI:716*
Seemingly unrelated regression *UI:793*, *UI:822*
Select
all *UI:106*
object *UI:105*
Selection model *See* Heckman selection
Sensitivity of binary prediction *UI:371*
Sequential breakpoint
estimation with *UI:472*
tests *UI:232*
Serial correlation
ARIMA models *UI:124*
Durbin-Watson statistic *UI:14*, *UI:127*
first order *UI:120*
higher order *UI:121*
nonlinear models *UI:159*
switching models *UI:589*
theory *UI:119*
two stage regression *UI:159*
Serial correlation test
equations *UI:127*, *UI:215*
panels *UI:1182*
VARs *UI:850*
Series *UI:439*
adjust values *UI:475*
auto-series *UI:201*
auto-updating *UI:211*
auto-updating and databases *UI:215*
auto-updating and forecasting *UI:187*
binning *UI:477*
classification *UI:477*
comparison *UI:475*
create *UI:124*, *UI:198*
cross-section specific *UI:1065*
delete observation *UI:132*
description of *UI:123*
descriptive statistics *UI:440*
difference *UI:192*
display format *UI:125*
display type *UI:125*
dynamic assignment *UI:199*
edit in differences *UI:593*
edit mode default *UI:944*
editing *UI:131*, *1:961*
fill values *UI:475*
functions *UI:189*
generate by command *UI:200*
graph *UI:440*, *UI:683*
implicit assignment *UI:199*
in pool objects *UI:1066*
insert observation *UI:132*
interpolate *UI:484*
lag *UI:192*
lead *UI:192*
pooled *UI:1066*
previewing contents *UI:107*
procs *UI:477*
properties *UI:474*
ranking *UI:194*
resample *UI:482*
setting graph axis *UI:728*
smpl + /- *UI:128*
spreadsheet view *UI:440*
spreadsheet view defaults *UI:943*
using expressions in place of *UI:201*
S-estimation *UI:456*
performing in EViews *UI:459*
tuning constants *UI:457*

weight function *UI:456*
SETAR *UI:491*
Shade region of graph *UI:834*
Shadowing of object names *UI:350*
Sharing violation *UI:328*
Show object view *UI:106*
Siddiqui difference quotient *UI:621*, *UI:634*
Siegel-Tukey test *UI:454*
Sign bias test *UI:275*
Sign test *UI:448*
Signal equation *UI:975*
Signal variables
views *UI:988*
Silverman bandwidth *UI:769*
Simple switching
regime probabilities *UI:942*
Simultaneous equations *See* systems.
Singular matrix
error in binary estimation *UI:369*
error in estimation *UI:22*, *UI:59*, *UII:1317*
error in logl *UI:647*, *UI:655*, *UI:657*
error in PDL estimation *UI:24*
error in RESET test *UI:245*
Skewness *UI:441*
Slope equality test (quantile regression) *UI:628*
technical details *UI:640*
SMA specification *UI:123*, *UI:126*
Smooth threshold autoregression *UII:507*
Smooth transition regression
transition functions *UII:508*
Smoothed AIC weights *UI:549*
Smoothing
ETS model *UI:559*
likelihood based *UI:559*
Markov switching *UI:587*, *UI:946*
methods *UI:553*
parameters *UI:554*
state space *UI:971*
Smpl command *UI:139*
Smpl + /- *UI:128*
Snapshots
defaults *UI:939*
Solve
Broyden *UI:1044*
Gauss-Seidel *UII:1322*
Newton-Raphson *UII:1319*
Sort
display *UI:595*
observations in a graph *UI:687*, *UI:842*
spreadsheet display *UI:595*
valmaps *UI:231*
workfile *UI:323*
Source
field in database query *UI:347*
Sparse label option *UI:444*, *UI:630*
Spearman rank correlation *UI:616*
Spearman rank-order
theory *UI:625*
Specification
by formula *UI:7*
by list *UI:6*
of equation *UI:6*
of nonlinear equation *UI:53*
of systems *UI:796*
Specification test
for binary models *UI:379*
for overdispersion *UI:413*
for tobit *UI:395*
of equation *UI:195*
RESET (Ramsey) *UI:244*
White *UI:219*
Specificity of binary prediction *UI:371*
Spectrum estimation *UI:676*, *UI:677*
Spike graph *UI:749*
Spool *UI:877*
add to *UI:878*
appending *UI:879*
comments *UI:882*
copying to *UI:879*
create *UI:877*
customization *UI:889*
delete objects *UI:886*
display mode *UI:891*
embedding *UI:880*
extract *UI:886*
flatten tree hierarchy *UI:887*
hiding objects *UI:883*
indentation *UI:886*
management *UI:878*
naming objects *UI:882*
order *UI:886*
print *UI:957*
print size *UI:896*
print to *UI:878*

printing *UI:895*
properties *UI:889*
rearrange *UI:886*
redirecting print to *UI:957*
resize *UI:885*
saving *UI:896*
Spreadsheet
file import *UI:150*
file import as matrix *UI:161*
file import as table *UI:161*, *UI:876*
series *UI:440*
sort display default *UI:944*
sort display order *UI:595*
view option *UI:943*, *UI:944*
Spreadsheet view
alpha *UI:223*
display type *UI:125*
group *UI:592*
SPSS file *UI:48*
SSAR *UI:591*
SSCP *UI:618*
Stability test *UI:225*, *UI:273*
Bai Perron tests *UI:230*
Chow breakpoint *UI:226*
Chow forecast *UI:242*
RESET *UI:244*
smooth transition *UII:524*
with unequal variance *UI:254*
stability test *UI:273*
Stacked data *UI:1069*
balanced *UI:1072*
descriptive statistics *UI:1077*
order *UI:1071*
Stacking data *UI:315*
Standard deviation *UI:441*
Standard error
for estimated coefficient *UI:12*
forecast *UI:176*, *UI:190*
impulse response
Confidence interval
impulse response *UI:854*
of the regression *UI:14*
See also Robust standard errors.
Standard errors
cluster-robust *UI:39*
Standardized coefficients *UI:196*
Standardized residual *UI:18*
binary models *UI:375*
censored models *UI:393*
count models *UI:412*
GLM *UI:435*
truncated models *UI:398*
STAR *UII:507*
Start
field in database query *UI:346*
field in workfile details *UI:64*
Start page *UI:935*
Starting values
(G)ARCH models *UI:269*
binary models *UI:368*
for ARMA estimation *UI:138*, *UI:144*
for coefficients *UI:57*, *UII:1315*
for nonlinear least squares *UI:54*, *UI:56*
for systems *UI:800*
logl *UI:651*
param statement *UI:57*, *UII:1316*
state space *UI:979*
user supplied *UI:139*, *UI:144*
Stata file *UI:48*
State equation *UI:970*, *UI:974*
State space *UI:969*
`@mprior` *UI:979*
`@vprior` *UI:979*
estimation *UI:973*, *UI:984*
filtering *UI:970*
forecasting *UI:972*
interpreting *UI:985*
observation equation *UI:970*
representation *UI:969*
specification *UI:969*, *UI:974*
specification (automatic) *UI:982*
starting values *UI:979*
state equation *UI:970*
views *UI:987*
State variables *UI:969*
State views *UI:988*
Static forecast *UI:181*
Static OLS *UI:294*, *UI:295*
Stationary time series *UI:667*
Status line *UI:9*
Step size *UII:1321*
logl *UI:649*
Step weights *UI:343*
Stepwise *UII:65*

STL decomposition *UI:506*
Stochastic equations
  in model *UI:996*
Store *UI:120*
  as .DB? file *UI:331*
  from pool *UI:1080*
  in database *UI:330*
  merge objects *UI:331*
Structural breaks *UI:60*
Structural change
  estimation in the presence of *UI:471*
  tests of *UI:225*, *UI:230*
Structural forecast *UI:182*
Structural solution of models *UI:1037*
Structural VAR *UI:869*
  estimation *UI:878*
  factorization matrix *UI:851*
Structuring a workfile *UI:271*
Studentized residual *UI:251*
Subtitle
  Breusch-Pagan LM test *UI:1178*
Sum of squared residuals
  for regression *UI:14*
Summarizing data *UI:598*
Summary statistics
  for regression variables *UI:13*
SUR *UI:793*, *UI:822*
Survivor function *UI:773*
  log *UI:773*
  save data *UI:585*
Switching regression *UI:583*
  dynamic models *UI:588*
  estimation in EViews *UI:591*
  expected durations *UI:599*, *UI:953*
  filtering *UI:585*
  initial probabilities *UI:594*, *UI:949*
  regime probabilities *UI:584*, *UI:601*, *UI:602*, *UI:954*, *UI:956*
  serial correlation *UI:589*
  transition probabilities *UI:599*, *UI:953*
  transition results *UI:599*, *UI:602*, *UI:953*, *UI:955*
  views available *UI:599*, *UI:952*
Switching VAR *UI:939*
SwitchingVAR
  switching intercept likelihood *UI:941*
Symbol graph *UI:745*
Symmetric quantiles test *UI:629*
  technical details *UI:641*
Symmetrically trimmed residuals *UI:395*
Syntax coloring *UI:948*
System *UI:791*
  ARCH *UI:794*
  covariance matrix *UI:807*
  create *UI:794*, *UI:795*
  cross-equation weighting *UI:792*
  definition *UI:791*
  derivatives *UI:807*
  estimation *UI:792*, *UI:800*
  estimation methods (technical) *UI:820*
  FIML *UI:824*
  forecast *UI:808*
  full information maximum likelihood *UI:794*
  GMM *UI:825*
  gradients *UI:807*
  Instruments *UI:797*
  make system from group *UI:663*
  OLS *UI:792*, *UI:821*
  options *UI:804*
  residuals *UI:809*
  specification *UI:796*
  specify from VAR *UI:868*
  SUR *UI:793*, *UI:822*
  three stage least squares *UI:793*, *UI:824*
  two stage least squares *UI:793*, *UI:823*
  views *UI:807*
  weighted least squares *UI:793*, *UI:821*
System options *UI:953*

T

Tab settings *UI:948*
Table *UI:865*
  cell annotation *UI:872*
  cell format *UI:870*
  cell merging *UI:873*
  color *UI:871*
  column resize *UI:870*
  column width *See* Column width.
  comments *UI:872*
  copy *UI:874*
  copy to other windows programs *UI:874*
  customization *UI:869*
  edit *UI:867*
  editing *UI:867*
  font *UI:871*

formatting *UI:870*
gridlines *UI:870*
merging *UI:873*
paste as unformatted text *UI:875*
print *UI:874*
read data from foreign source *UI:876*
row resize *UI:870*
save to disk *UI:875*
selecting cells *UI:865*
sort rows *UI:868*
title *UI:870*
Tabs
See Page
Tabulation
n-way *UI:629*
one-way *UI:458*
Talworth function *UI:453*
TAR *UI:491*
TARCH *UI:277*
Template
dated data tables *UI:611*
graphs *UI:854*
Test
See also Hypothesis tests, Specification test and Goodness of fit
ARCH *UI:218*
Arrelano-Bond serial correlation *UI:1182*
breakpoint *UI:226*, *UI:228*, *UI:230*
coefficient *UI:196*
cross-section dependence *UI:1177*, *UI:1239*
Durbin-Wu-Hausman *UI:113*
Granger causality *UI:661*, *UI:1230*
Hansen instability *UI:313*
heteroskedasticity *UI:217*
multiple breakpoint *UI:230*
Park added variable *UI:316*
pooled *UI:1096*
RESET *UI:244*
residual *UI:213*
seasonal unit root *UI:695*
stability tests *UI:225*
unit root with break *UI:679*
variance ratio *UI:739*
White *UI:219*
Text *UI:876*
Text file
import as matrix *UI:161*
import as table *UI:161*, *UI:876*
open as workfile *UI:48*
Theil inequality coefficient *UI:467*, *UI:178*
Themes *UI:936*
Theoretical distribution graph *UI:771*
save data *UI:585*
Three stage least squares *See* 3sls (Three Stage Least Squares)
Threshold autoregression *UI:491*
Threshold GARCH (TARCH) *UI:277*
Threshold regression *UI:491*
smooth *UII:507*
Thresholding *UI:766*, *UI:780*
Time series functions *UI:192*
Title bar *UI:6*, *UI:61*, *UI:113*
To (lag range) *UI:7*
Tobit *UI:387*
Toolbar *UI:61*, *UI:113*
Tracking model variables *UI:1042*
TRAMO/SEATS
in X-13 *UI:497*, *UI:501*
Tramo/Seats *UI:532*
Transition equation *UI:970*
Transition results
Markov switching *UI:599*, *UI:953*
outputting *UI:602*, *UI:955*
switching regression *UI:599*, *UI:953*
Transpose *UI:592*
Trend
panel data *UI:1124*
See also `@trend`.
Truncated dependent variable *UI:397*
estimation *UI:397*
fitted index *UI:399*
forecasting *UI:399*
log likelihood *UI:397*
residuals *UI:398*
Truncation point *UI:398*
TSD data format *UI:356*
TSP portable data format *UI:424*
t-statistics
retrieve from equation *UI:12*
Tukey *UI:772*
Tukey-Hamming kernel
cointegrating regression *UI:305*
GMM estimation *UI:110*
long-run covariance estimation *UI:658*

panel cointegrating regression *UI:1197*
robust standard errors *UI:45*
technical details *UIII:1342*
Tukey-Hanning kernel
cointegrating regression *UI:305*
GMM estimation *UI:110*
long-run covariance estimation *UI:658*
panel cointegrating regression *UI:1197*
robust standard errors *UI:45*
technical details *UIII:1342*
Tukey-Parzen kernel
cointegrating regression *UI:305*
GMM estimation *UI:110*
long-run covariance estimation *UI:658*
panel cointegrating regression *UI:1197*
robust standard errors *UI:45*
technical details *UIII:1342*
Tuning constants
M-estimation *UI:452*
S-estimation *UI:457*
Type
field in database query *UI:345*

U

U.S. Energy Administration *UI:387*
UMP random effects test *UI:1170*
Unconditional residual *UI:145*
Undo *UI:475*
Uniit roots
seasonal *UI:695*
Unit root test *UI:465*, *UI:667*
augmented Dickey-Fuller *UI:672*
Dickey-Fuller *UI:672*
Dickey-Fuller GLS detrended *UI:673*
Elliot, Rothenberg, and Stock *UI:675*
KPSS *UI:675*
panel data *UI:705*, *UI:1234*
Phillips-Perron *UI:674*, *UI:675*
pooled data *UI:1077*
seasonal *UI:695*
trend assumption *UI:673*
with breakpoints *UI:679*
Unit root tests
panel dependent *UI:716*
Units
field in database query *UI:347*
Unstacked data *UI:1068*
Unstacking data *UI:309*
Unstacking identifiers *UI:311*
Untitled *UI:115*, *UI:116*
Update
automatic *UI:211*
coefficient vector *UI:19*, *UIII:1316*
from Database *UI:120*
graphs *UI:829*
group *UI:592*
Updating graphs *UI:828*
Urzua factorization matrix *UI:851*
US Census *UI:368*
US Census data *UI:368*
User defined menus
See Add-ins.
User objects *I:899*
User specified GMM weight matrix *UI:110*
User supplied starting values *UI:139*, *UI:144*
User-defined optimization *See* Optimization (user-defined).

V

Valmap *UI:227*
cautions *UI:238*
find label for value *UI:236*
find numeric value for label *UI:237*
find string value for label *UI:237*
functions *UI:236*
properties *UI:232*
sorting *UI:231*
Value map *See* Valmap.
Van der Waerden *UI:772*
Van der Waerden test *UI:448*, *UI:453*
VAR *UI:854*
AR roots *UI:848*
autocorrelation LM test *UI:850*
autocorrelation test *UI:850*
coefficients *UI:883*
cointegration *UI:1245*
correlograms *UI:849*
estimation *UI:835*, *UI:841*
estimation output *UI:837*, *UI:845*
factorization matrix in normality test *UI:851*
forecasting *UI:866*, *UI:883*
Granger causality test *UI:848*
historical decomposition *UI:864*
impulse response *UI:853*

Jarque-Bera normality test *UI:850*
lag exclusion test *UI:848*
lag length *UI:849*
lag length choice *UI:849*
lag structure *UI:848*
linear restrictions *UI:839*
mathematical model *UI:833*
Mixed frequency *UI:927*
response confidence intervals *UI:854*
response standard errors *UI:854*
See also Impulse response, Structural VAR.
variance decomposition *UI:862*
VARHAC *UI:655*
technical details *UII:1346*
Variance
equality test *UI:454*
hypothesis test of *UI:447*
wavelet decomposition *UI:761*, *UI:778*
Variance decomposition *UI:200*, *UI:862*
Variance equation *See* ARCH and GARCH.
Variance factor *UI:418*
Variance inflation factor (VIF) *UI:199*
Variance proportion *UI:179*
Variance ratio test *UI:739*
example *UI:742*
technical details *UI:746*
VEC *UI:881*
estimating *UI:881*
Vector autoregression
See VAR.
Vector error correction model *See* VEC and VAR.
Verbose mode *UI:949*
View
default *UI:106*
Vogelsang-Perron unit root tests *UI:679*
Volatility *UI:264*

W

Wald test *UI:202*
coefficient restriction *UI:202*
demonstration *UI:30*
formula *UI:207*
F-statistic *UI:208*
joint restriction *UI:204*
nonlinear restriction *UI:207*
structural change with unequal variance *UI:254*
Warning on close option *UI:937*
Watson test *UI:455*
Wavelet
anova *UI:761*, *UI:778*
denoising *UI:766*, *UI:780*
multiresolution analysis *UI:777*
outlier detection *UI:769*, *UI:779*
save results *UI:586*
thresholding *UI:766*, *UI:780*
thresholding rule *UI:780*
transformation *UI:753*
variance confidence intervals *UI:778*
variance decomposition *UI:761*, *UI:778*
Wavelet transforms
boundary conditions *UI:777*
Wavelets
discrete transform *UI:775*
Weak instruments *UI:98*, *UI:114*
Weight functions
M-estimation *UI:452*
S-estimation *UI:456*
Weighted least squares *UI:47*
cross-equation weighting *UI:792*
nonlinear *UI:58*
nonlinear two stage *UI:97*, *UI:108*
pool *UI:1085*
system estimation *UI:821*
two stage in systems *UI:793*, *UI:823*
weight scaling *UI:49*
weight type *UI:49*
Weighting matrix
GMM *UI:103*, *UI:110*
heteroskedasticity and autocorrelation consistent (HAC) in system GMM *UI:827*
heteroskedasticity and autocorrelation consistent (HAC) robust standard errors *UI:45*
kernel options (system) *UI:827*
system GMM *UI:826*
White (cointegrating regression) *UI:305*
White (GMM) *UI:110*
White (panel cointegrating regression) *UI:1197*
White (robust standard errors) *UI:33*
White (system GMM) *UI:826*
Welsch function *UI:453*
White heteroskedasticity consistent covariance matrix
cointegrating regression *UI:305*
GMM *UI:110*

panel cointegrating regression *UI:1197*
robust standard errors *UI:33*
system GMM *UI:826*
White heteroskedasticity test *UI:219*
VAR *UI:852*
Whitening *UI:657*, *UI:664*
Width of table column *UI:870*
Wilcoxon test
rank sum *UI:452*
signed ranks *UI:448*
Wildcard characters *UI:73*
in easy query *UI:342*
Windmeijer standard errors *UI:107*
Window
active *UI:113*
database *UI:327*
EViews main *UI:5*
object *UI:115*
Within deviations *UI:1124*, *UI:1133*
Within factors *UI:813*
identification *UI:817*
Word
pasting graphs and data into *1:903*
Work area *UI:10*
Workfile
append to *UI:299*
applying structure to *UI:281*
attributes *UI:62*
automatic backup *UI:78*, *UI:946*
common structure errors *UI:294*
comparing *UI:92*
contract *UI:302*
copy from *UI:302*
create *UI:42*
description of *UI:41*
details display *UI:62*
directory *UI:61*
export *UI:323*
filtering objects *UI:73*
load existing from disk *UI:81*
multi-page *UI:82*
observation numbers *UI:128*
panel *UI:1111*
pool *UI:1061*, *UI:1070*
remove structure *UI:296*
reshape *UI:306*
resize *UI:283*, *UI:296*
sample *UI:136*
save *UI:75*
sorting *UI:323*
stacking *UI:315*
statistics *UI:75*
storage defaults *UI:945*
storage precision and compression *UI:945*
structure settings *UI:282*
structuring *UI:271*
summary view *UI:75*
undated *UI:273*
unstacking *UI:309*
window *UI:60*
World Bank data *UI:425*
Write *UI:1080*

X

X-11 *UI:532*
using X-12 *UI:525*
using X-13 *UI:494*
X-12 *UI:523*
X-13 *UI:487*, *UI:493*
ARIMA estimation *UI:498*
ARIMA forcasting *UI:498*
automatic outliers *UI:492*
example *UI:504*
manual ARIMA *UI:493*
output options *UI:502*
seasonal adjustment method *UI:499*
TRAMO based ARIMA *UI:497*
transformations *UI:489*
user defined regressors *UI:490*
variable options *UI:488*
X-11 based ARIMA *UI:494*
XY (area) graph *UI:757*
XY (bar) graph *UI:758*
XY (line) graph *UI:756*

Y

Yates' continuity correction *UI:453*

Z

Zivot-Andrews unit root test *UI:679*