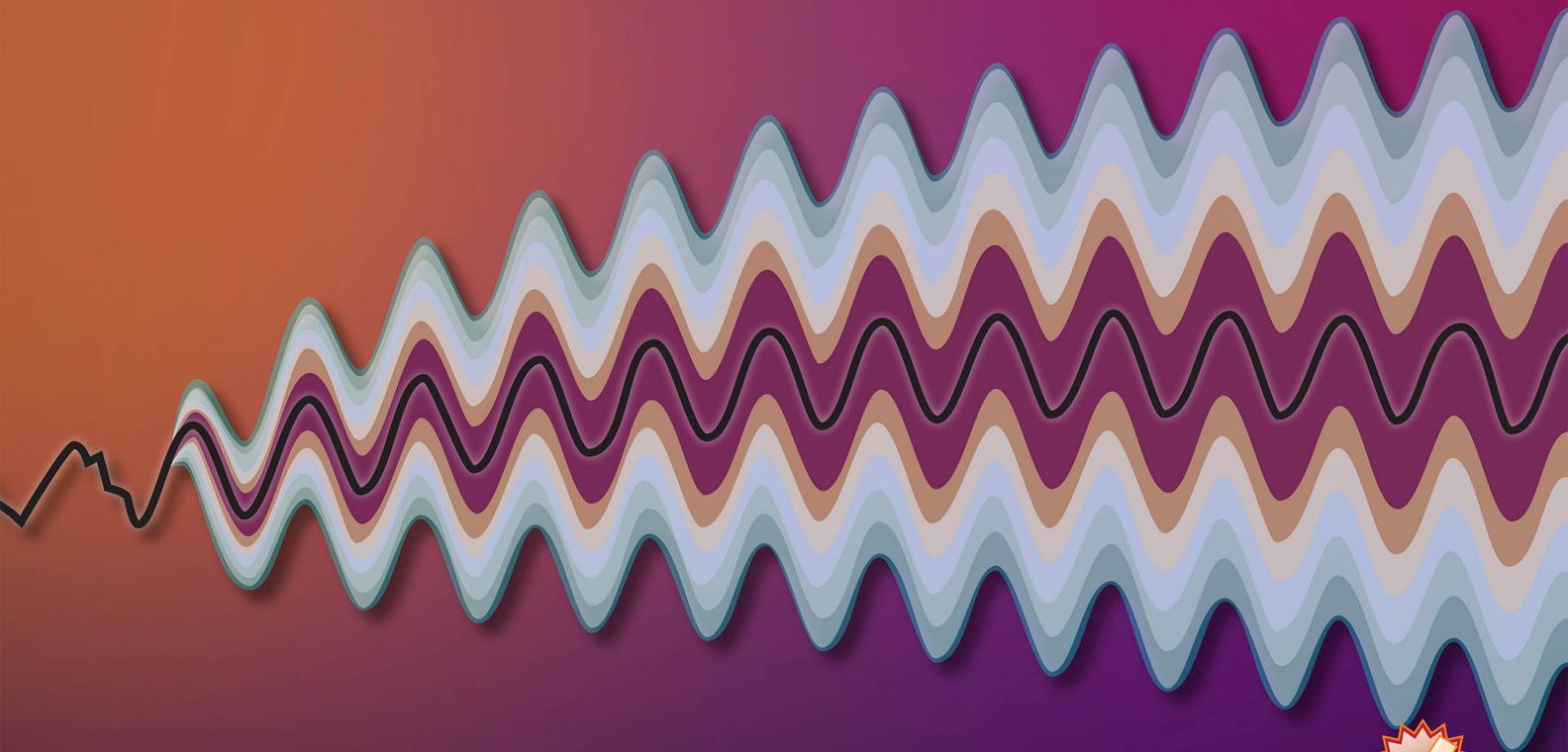




SAS® for Forecasting Time Series

Third Edition



John C. Brocklebank
David A. Dickey
Bong S. Choi

The correct bibliographic citation for this manual is as follows: Brocklebank, John C., David A. Dickey, and Bong S. Choi. 2018. *SAS® for Forecasting Time Series, Third Edition*. Cary, NC: SAS Institute Inc.

SAS® for Forecasting Time Series, Third Edition

Copyright © 2018, SAS Institute Inc., Cary, NC, USA

ISBN 978-1-62959-844-4 (Hard copy)

ISBN 978-1-62960-544-9 (EPUB)

ISBN 978-1-62960-545-6 (MOBI)

ISBN 978-1-62960-546-3 (PDF)

All Rights Reserved. Produced in the United States of America.

For a hard-copy book: No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, or otherwise, without the prior written permission of the publisher, SAS Institute Inc.

For a web download or e-book: Your use of this publication shall be governed by the terms established by the vendor at the time you acquire this publication.

The scanning, uploading, and distribution of this book via the Internet or any other means without the permission of the publisher is illegal and punishable by law. Please purchase only authorized electronic editions and do not participate in or encourage electronic piracy of copyrighted materials. Your support of others' rights is appreciated.

U.S. Government License Rights; Restricted Rights: The Software and its documentation is commercial computer software developed at private expense and is provided with RESTRICTED RIGHTS to the United States Government. Use, duplication, or disclosure of the Software by the United States Government is subject to the license terms of this Agreement pursuant to, as applicable, FAR 12.212, DFAR 227.7202-1(a), DFAR 227.7202-3(a), and DFAR 227.7202-4, and, to the extent required under U.S. federal law, the minimum restricted rights as set out in FAR 52.227-19 (DEC 2007). If FAR 52.227-19 is applicable, this provision serves as notice under clause (c) thereof and no other notice is required to be affixed to the Software or documentation. The Government's rights in Software and documentation shall be only those set forth in this Agreement.

SAS Institute Inc., SAS Campus Drive, Cary, NC 27513-2414

March 2018

SAS® and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are trademarks of their respective companies.

SAS software may be provided with certain third-party software, including but not limited to open-source software, which is licensed under its applicable third-party software license agreement. For license information about third-party software distributed with SAS software, refer to <http://support.sas.com/thirdpartylicenses>.

Contents

About This Book	ix
About The Authors	xi
Acknowledgments	xiii
Chapter 1: Overview of Time Series	1
1.1 Introduction.....	1
1.2 Analysis Methods and SAS/ETS Software.....	2
1.2.1 Options.....	2
1.2.2 How SAS/ETS Procedures Interrelate	3
1.3 Simple Models: Regression.....	5
1.3.1 Linear Regression	5
1.3.2 Highly Regular Seasonality	11
1.3.3 Regression with Transformed Data	17
Chapter 2: Simple Models: Autoregression.....	23
2.1 Introduction.....	23
2.1.1 Terminology and Notation.....	23
2.1.2 Statistical Background	23
2.2 Forecasting	24
2.2.1 PROC ARIMA for Forecasting.....	25
2.2.2 Backshift Notation B for Time Series	32
2.2.3 Yule-Walker Equations for Covariances.....	33
2.3 Fitting an AR Model in PROC REG.....	37
Chapter 3: The General ARIMA Model	41
3.1 Introduction.....	41
3.1.1 Statistical Background	41
3.1.2 Terminology and Notation.....	41
3.2 Prediction	42
3.2.1 One-Step-Ahead Predictions.....	42
3.2.2 Future Predictions	43
3.3 Model Identification.....	46
3.3.1 Stationarity and Invertibility	46
3.3.2 Time Series Identification	47
3.3.3 Chi-Square Check of Residuals	56
3.3.4 Summary of Model Identification	56
3.4 Examples and Instructions	56
3.4.1 IDENTIFY Statement for Series 1-8.....	57
3.4.2 Example: Iron and Steel Export Analysis.....	65

3.4.3 Estimation Methods Used in PROC ARIMA.....	70
3.4.4 ESTIMATE Statement for Series 8-A	72
3.4.5 Nonstationary Series.....	77
3.4.6 Effect of Differencing on Forecasts.....	78
3.4.7 Examples: Forecasting IBM Series and Silver Series	80
3.4.8 Models for Nonstationary Data	84
3.4.9 Differencing to Remove a Linear Trend	91
3.4.10 Other Identification Techniques.....	95
3.5 Summary of Steps for Analyzing Nonseasonal Univariate Series	104
Chapter 4: The ARIMA Model: Introductory Applications.....	107
4.1 Seasonal Time Series	107
4.1.1 Introduction to Seasonal Modeling	107
4.1.2 Model Identification.....	108
4.2 Models with Explanatory Variables	119
4.2.1 Case 1: Regression with Time Series Errors	120
4.2.2 Case 1A: Intervention	120
4.2.3 Case 2: Simple Transfer Functions.....	121
4.2.4 Case 3: General Transfer Functions	121
4.2.5 Case 3A: Leading Indicators	121
4.2.6 Case 3B: Intervention	121
4.3 Methodology and Example	122
4.3.1 Case 1: Regression with Time Series Errors	122
4.3.2 Case 2: Simple Transfer Functions.....	131
4.3.3 Case 3: General Transfer Functions	133
4.3.4 Case 3B: Intervention	155
4.4 Further Example	161
4.4.1 North Carolina Retail Sales	161
4.4.2 Construction Series Revisited.....	168
4.4.3 Milk Scare (Intervention).....	172
4.4.4 Terrorist Attack.....	175
Chapter 5: The ARIMA Model: Special Applications.....	177
5.1 Regression with Time Series Errors and Unequal Variances.....	177
5.1.1 Autoregressive Errors	177
5.1.2 Example: Energy Demand at a University.....	178
5.1.3 Unequal Variances.....	182
5.1.4 ARCH, GARCH, and IGARCH for Unequal Variances	184
5.2 Cointegration.....	189
5.2.1 Cointegration and Eigenvalues	191
5.2.2 Impulse Response Function.....	192
5.2.3 Roots in Higher-Order Models	192
5.2.4 Cointegration and Unit Roots.....	194
5.2.5 An Illustrative Example	196
5.2.6 Estimation of the Cointegrating Vector.....	199
5.2.7 Intercepts and More Lags	201
5.2.8 PROC VARMAX.....	202

5.2.9 Interpretation of the Estimates.....	205
5.2.10 Diagnostics and Forecasts	206
Chapter 6: Exponential Smoothing	209
6.1 Single Exponential Smoothing	209
6.1.1 The Smoothing Idea.....	209
6.1.2 Forecasting with Single Exponential Smoothing.....	210
6.1.3 Alternative Representations	210
6.1.4 Atlantic Ocean Tides: An Example	211
6.1.5 Improving the Tide Forecasts.....	213
6.2 Exponential Smoothing for Trending Data	216
6.2.1 Linear and Double Exponential Smoothing.....	216
6.2.2 Properties of the Forecasts	217
6.2.3 A Generated Multi-Series Example	217
6.2.4 Real Data Examples.....	219
6.2.5 Boundary Values in Linear Exponential Smoothing	222
6.2.6 Damped Trend Exponential Smoothing.....	228
6.2.7 Diagnostic Plots	229
6.2.8 Sums of Forecasts	231
6.3 Smoothing Seasonal Data	232
6.3.1 Seasonal Exponential Smoothing	232
6.3.2 Winters Method	234
6.4.1 Validation	236
6.4.2 Choosing a Model Visually	237
6.4.3 Choosing a Model Numerically.....	239
6.5 Advantages of Exponential Smoothing	240
6.6 How the Smoothing Equations Lead to ARIMA in the Linear Case	240
Chapter 7: Unobserved Components and State Space Models	243
7.1 Nonseasonal Unobserved Components Models.....	243
7.1.1 The Nature of Unobserved Components Models	243
7.1.2 A Look at the PROC UCM Output	246
7.1.3 A Note on Unit Roots in Practice	247
7.1.4 The Basic Structural Model Related to ARIMA Structures.....	247
7.1.5 A Follow-Up on the Example	249
7.2 Diffuse Likelihood and Kalman Filter: Overview and a Simple Case	250
7.2.1 Diffuse Likelihood in a Simple Model.....	251
7.2.2 Definition of a Diffuse Likelihood	251
7.2.3 A Numerical Example	252
7.3 Seasonality in Unobserved Components Models	254
7.3.1 Description of Seasonal Recursions.....	254
7.3.2 Tourism Example with Regular Seasonality.....	254
7.3.3 Decomposition	257
7.3.4 Another Seasonal Model: Sine and Cosine Terms	258
7.3.5 Example with Trigonometric Components.....	259
7.3.6 The Seasonal Component Made Local and Damped.....	261

7.4 A Brief Introduction to the SSM Procedure.....	265
7.4.1 Brief Overview.....	265
7.4.2 Simple Examples	265
7.4.3 Extensions of the AR(1) Model	266
7.4.4 Accommodation for Curvature	267
7.4.5 Models with Several Lags.....	270
7.4.6 Bivariate Examples.....	273
7.4.7 The Start-up Problem Revisited.....	274
7.4.8 Example and More Details on the State Space Approach	276
Chapter 8: Adjustment for Seasonality with PROC X13	285
8.1 Introduction	285
8.2 The X-11 Method	287
8.2.1 Moving Averages	287
8.2.2 Outline of the X-11 Method	290
8.2.3 Basic Seasonal Adjustment Using the X-11 Method	291
8.2.4 Tests for Seasonality.....	292
8.3 regARIMA Models and TRAMO	295
8.3.1 regARIMA Models.....	295
8.3.2 Automatic Selection of ARIMA Orders.....	296
8.4 Data Examples	296
8.4.1 Airline Passengers Revisited.....	296
8.4.3 Employment in the United States	299
Chapter 9: SAS Forecast Studio	305
9.1 Introduction	305
9.2 Creating a Project	305
9.3 Overview of Available Modes.....	310
9.4 Project Settings.....	312
9.4.1 Model Generation	312
9.4.2 Goodness of Fit and Honest Assessment.....	313
9.4.2 Transformation and Outlier Detection	314
9.5 Creating Custom Events	318
9.6 Hierarchical Time Series and Reconciliation	320
Chapter 10: Spectral Analysis	333
10.1 Introduction	333
10.2 Example: Plant Enzyme Activity	334
10.3 PROC SPECTRA.....	335
10.4 Tests for White Noise	337
10.5 Harmonic Frequencies	338
10.6 Extremely Fast Fluctuations and Aliasing	342
10.7 The Spectral Density.....	342
10.8 Some Mathematical Detail (Optional Reading)	345
10.9 Estimation of the Spectrum: The Smoothed Periodogram	345

10.10 Cross-Spectral Analysis	346
10.10.1 Interpretation of Cross-Spectral Quantities.....	346
10.10.2 Interpretation of Cross-Amplitude and Phase Spectra	348
10.10.3 PROC SPECTRA Statements.....	349
10.10.4 Cross-Spectral Analysis of the Neuse River Data	352
10.10.5 Details on Gain, Phase, and Pure Delay	354
References	359
Index	363

About This Book

What Does This Book Cover?

Starting from basics, this book shows you methods for modeling data taken over time—both univariate and multivariate. From the well-known ARIMA models to unobserved components, this book discusses and illustrates with engaging examples statistical methods that range from simple to complicated. Many of the newer methods are variations on the basic ARIMA structures, and the links are discussed where appropriate.

Unique to this book is its pragmatism. It bridges a gap between books on theoretical mathematical developments and books that provide only a high-level overview of applications. This book gives you the statistical underpinnings and the SAS applications that empower you to put the methods into practice. The collection of examples represents a broad range of SAS applications and explanation sufficient for you to understand the strengths of a method, the weaknesses of it, and situations best suited for it. Examples serve as templates that you can adjust to fit your specific forecasting needs.

Is This Book for You?

Successful enterprises want to have an idea of what is happening next. They therefore face a forecasting problem. If you want to use statistical methods to forecast future values of data taken over time, then you will need the methods in this book. Professionally, those who will benefit most from reading are statisticians, economists, or data scientists.

What Are the Prerequisites for This Book?

To gain the most benefit from this book, ideally you will have intermediate knowledge of SAS. More importantly, knowledge of some statistical ideas, such as multiple regression, will ensure that you gain the most value from reading.

What Is New in This Edition?

In addition to updating of all chapters and including fresh examples selected for their inherent interest, this third edition features four completely new chapters to teach the following topics:

- Exponential smoothing
- Unobserved components and state space models
- Adjustment for seasonality
- SAS Forecast Studio

What Should You Know about the Examples?

This book includes tutorials for you to follow to gain hands-on experience with SAS.

Software Used to Develop the Book's Content

Content for this book was developed with the following SAS products:

- Base SAS 9.4
 - SAS/ETS 14.3
 - SAS/STAT 14.3
 - SAS/GRAF 9.4
 - SAS/IML 14.3
 - SAS Forecast Studio
-

Example Code and Data

You can access the example code and data for this book by linking to its author page at <https://support.sas.com/authors>.

Output and Graphics

The figures showing output in this book were generated with a SAS Output Delivery System style customized for optimal print quality; therefore, your output will differ in appearance.

We Want to Hear from You

SAS Press books are written by SAS Users for SAS Users. We welcome your participation in their development and your feedback on SAS Press books that you are using. Please visit sas.com/books to do the following:

- Sign up to review a book
- Recommend a topic
- Request information on how to become a SAS Press author
- Provide feedback on a book

Do you have questions about a SAS Press book that you are reading? Contact the author through saspress@sas.com or https://support.sas.com/author_feedback.

SAS has many resources to help you find answers and expand your knowledge. If you need additional help, see our list of resources: sas.com/books.

About the Authors



John C. Brocklebank, PhD, is Executive Vice President, Global Hosting and US Professional Services, at SAS. Dr. Brocklebank brings more than 35 years of SAS programming and statistical experience to his leadership role at SAS. He holds 14 patents and directs the SAS Advanced Analytics Lab for State and Local Government, which devotes the resources of nearly 300 mostly doctoral-level SAS experts to devising technology solutions to critical state and local government issues. He also serves on the Board of Directors for the North Carolina State College of Sciences Foundation, where he advises the dean and college leaders on issues affecting the future direction of the college. In addition, he is a member of the Lipscomb University College of Computing and Technology Advancement Council and the Analytics Corporate Advisory Board, Analytics and Data Mining Programs, Spears School of Business at Oklahoma State University. Dr. Brocklebank holds an MS in biostatistics and in 1981 received a PhD in statistics and mathematics from North Carolina State University, where he now serves as a Physical and Mathematical Sciences Foundation board member and an adjunct professor of statistics.



David A. Dickey, PhD, is a William Neal Reynolds Distinguished Professor of Statistics at North Carolina State University, where he teaches graduate courses in statistical methods and time series. An accomplished SAS user since 1976, an award-winning teacher, and a prolific and highly cited author, he co-invented the famous Dickey-Fuller test used in SAS/ETS software. He is a fellow of the American Statistical Association, was a founding member of the NCSU Institute for Advanced Analytics, is a member of the Financial Math faculty, and fulfills an associate appointment in the Department of Agricultural and Resource Economics. Dr. Dickey holds an MS in mathematics from Miami University–Ohio, and in 1976 he received his PhD in statistics from Iowa State University.



Bong S. Choi, PhD, is a Senior Associate Analytical Consultant at SAS. He has worked on projects across a variety of industries, including health care, retail, and banking. A SAS certified advanced programmer, he has been a SAS user since 2011. Dr. Choi holds an MS in applied statistics from the University of Michigan at Ann Arbor and in 2016 received his PhD in statistics from North Carolina State University.

Learn more about these authors by visiting their author pages, where you can download free book excerpts, access example code and data, read the latest reviews, get updates, and more:

<http://support.sas.com/Brocklebank>

<http://support.sas.com/Dickey>

<http://support.sas.com/Choi>

Acknowledgments

We acknowledge Julie McAlpine Palmieri for encouraging the publication of the third edition of this book, Jenny Jennings Foerst for overseeing the editing and composition, Amy Wolfe for copyediting the final manuscript, Robert Harris for providing graphics support, Denise T. Jones for completing production, and Rajesh Selukar for providing his technical expertise. Dr. Selukar's help in understanding the SAS implementation of the state space algorithm was invaluable to this revised edition. We also thank Mark Little, Donna Woodward, Wen Bardsley, and Chip Wells for their technical reviews and their many suggestions for improving the manuscript. These SAS employees were generous with their time and always happy to assist.

John C. Brocklebank

I extend deep gratitude to Dr. Dickey for his substantial additions to this edition and his painstaking lead on several rounds of substantive revisions. In addition, I thank Dr. Choi for his work on the X13 procedure, for running and checking all programs, for optimizing output from the programs, and for contributing long hours to editing the manuscript and liaising with SAS Press to ensure a quality publication.

Finally, I thank my wife Vicki, as always, for her understanding and support.

David A. Dickey

I acknowledge my wife Barbara for her patience and encouragement throughout the book writing process and my whole career. My efforts on this book are dedicated to Barbara and our grandchildren Aliyah, Emmy, Declan and Gillian.

Bong S. Choi

I thank my wife Areum for her love and support, and my infant son Yuhnu for his beautiful smile.

Chapter 1: Overview of Time Series

1.1 Introduction	1
1.2 Analysis Methods and SAS/ETS Software	2
1.2.1 Options.....	2
1.2.2 How SAS/ETS Procedures Interrelate.....	3
1.3 Simple Models: Regression	5
1.3.1 Linear Regression	5
1.3.2 Highly Regular Seasonality	11
1.3.3 Regression with Transformed Data	17

1.1 Introduction

This book deals with data collected at equally spaced points in time. The discussion begins with a single observation at each point. It continues with k series being observed at each point, and then they are analyzed together in terms of their interrelationships.

One of the main goals of univariate time series analysis is to forecast future values of the series. For multivariate series, relationships among component series, as well as forecasts of these components, might be of interest. Secondary goals are smoothing, interpolating, and modeling the structure. Three important characteristics of time series are often encountered: seasonality, trend, and autocorrelation.

For example, seasonality occurs when data are collected monthly, and the value of the series in any given month is closely related to the value of the series in that same month in previous years. Seasonality can be very regular or can change slowly over a period of years.

A trend is a regular, slowly evolving change in the series level. Changes that can be modeled by low-order polynomials or low-frequency sinusoids fit into this category. For example, if a plot of sales over time shows a steady increase of \$500 per month, you might fit a linear trend to the sales data. A trend is a long-term movement in the series.

In contrast, autocorrelation is a local phenomenon. When deviations from an overall trend tend to be followed by deviations of a like sign, the deviations are positively autocorrelated. Autocorrelation is the phenomenon that distinguishes time series from other branches of statistical analysis.

For example, consider a manufacturing plant that produces computer parts. Normal production is 100 units per day, although actual production varies from this mean of 100. Variation can be caused by machine failure, absenteeism, or incentives such as bonuses or approaching deadlines. A machine might malfunction for several days, resulting in a run of low productivity. Similarly, an approaching deadline might increase production over several days. This is an example of *positive autocorrelation*, with data falling and staying below 100 for a few days, and then rising above 100 and staying high for a while, and then falling again, and so on.

Another example of positive autocorrelation is the flow rate of a river. Consider variation around the seasonal level: you might see high flow rates for several days following rain and low flow rates for several days during dry periods.

Negative autocorrelation occurs less often than positive autocorrelation. An example is a worker's attempt to control temperature in a furnace. The autocorrelation pattern depends on the worker's habits, but suppose he reads a low value of a furnace temperature and turns up the heat too far, and then similarly turns it down too far when the reading is high. If he reads and adjusts the temperature each minute, you can expect a low temperature reading to be followed by a high temperature reading. As a second example, an athlete might follow a long workout day with a short workout day, and vice versa. The time he spends exercising daily displays negative autocorrelation.

1.2 Analysis Methods and SAS/ETS Software

Modern statistical analysis is performed with software. SAS software offers the SAS/ETS suite of tools, introduced here and expanded on in subsequent chapters.

1.2.1 Options

When you perform univariate time series analysis, you observe a single series over time. The goal is to model the historic series, and then to use the model to forecast future values of the series. An early forecasting method that is still useful today is exponential smoothing. In this approach, data are downweighted more the further into the past they occur. This makes the forecast more responsive to recent data. The degree of downweighting is controlled by model parameters that can be estimated from the data. You can use some simple SAS/ETS procedures to model low-order polynomial trends and autocorrelation. For seasonal data, you might want to fit a Holt-Winters exponentially smoothed trend-seasonal model. If the trend is local and the data are nonseasonal, you might prefer one of several methods of nonseasonal exponential smoothing, which use exponential smoothing to fit a local linear or quadratic trend. For higher-order trends or for cases where the forecast variable Y_t is related to one or more explanatory variables X_t , PROC AUTOREG estimates this relationship and fits an AR series as an error term. Simpler versions of exponential smoothing and limited models with trend and autoregressive errors are available in PROC FORECAST, but the functionality is improved in PROC ESM and PROC AUTOREG. In this book, the focus is on the new ESM procedure, rather than PROC FORECAST.

Polynomials in time and seasonal indicator variables (see [section 1.3.2](#)) can be computed as far into the future as needed. However, if the explanatory variable is a nondeterministic time series, actual future values are not available. PROC AUTOREG treats future values of the explanatory variable as known, so user-supplied forecasts of future values with PROC AUTOREG might give overly optimistic standard errors of forecast estimates. More sophisticated procedures such as PROC SSM, PROC VARMAX, or PROC ARIMA, with their transfer function options, are preferable when the explanatory variable's future values are unknown.

One approach to modeling seasonality in time series is the use of seasonal indicator variables in PROC AUTOREG to model a highly regular seasonality. Also, the AR error series from PROC AUTOREG or from PROC FORECAST with METHOD=STEPAR can include some correlation at seasonal lags (that is, it can relate the deviation from trend at time t to the deviation at time $t - 12$ in monthly data). The WINTERS method of PROC ESM uses updating equations similar to those in simple exponential smoothing to fit a seasonal multiplicative model.

Another approach to seasonality is to remove it from the series and to forecast the seasonally adjusted series with other seasonally adjusted series used as inputs. The U.S. Census Bureau has adjusted thousands of series with its X-11 seasonal adjustment package. This package is the result of years of work by census researchers and is the basis for the seasonally adjusted figures that the federal government reports. A modification of the X11 procedure that incorporates ARIMA extensions of the data was initially developed by Statistics Canada and is in popular use in the United States. This and further developments by the Bank of Spain are now incorporated by the U.S. Census Bureau's updated X13 procedure. You can seasonally adjust your own data using PROC X13, which is the U.S. Census Bureau's program set up as a SAS procedure. If you are using seasonally adjusted figures as explanatory variables, this procedure is useful.

The X13 procedure can be thought of as dividing a single series into component series whose graphs are helpful in visually evaluating the historical behavior of seasonality, trend, cycles, and irregularity in a series. It also helps evaluate trading-day effects, effects due to the number of Saturdays, for example, in a month for a retailer. The usefulness of such a decomposition has led to similar approaches in PROC TIMESERIES, PROC UCM for unobserved components models, and PROC SSM for analyzing state space models. PROC SSM is a generalization of PROC UCM and is an advancement of the older PROC STATESPACE (the newer procedure is covered in this book). PROC SSM uses a component approach for vector-valued time series. PROC UCM uses a similar approach for univariate time series. In this chapter, descriptions of PROC SSM properties apply to PROC UCM, so the estimation method, the Kalman filter, applies to both.

An alternative to using PROC X13 is to model the seasonality as part of an ARIMA model. Or, if the seasonality is highly regular, you can model it with indicator variables or trigonometric functions as explanatory variables. The X13, UCM, SSM, and ARIMA procedures can identify and adjust for outliers.

If you are unsure about the presence of seasonality, you can use PROC SPECTRA to check for it. PROC SPECTRA decomposes a series into cyclical components of various periodicities. Monthly data with highly regular seasonality have a large ordinate at period 12 in the PROC SPECTRA output SAS data set. Other periodicities, such as multiyear business cycles, might appear in this analysis. PROC SPECTRA provides a check on model residuals to see whether they exhibit cyclical patterns over time. Often these cyclical patterns are not found by other procedures. Thus, it is good practice to

analyze residuals with this procedure. PROC SPECTRA relates an output time series Y_t to one or more input or explanatory series X_t in terms of cycles. Specifically, cross-spectral analysis estimates the change in amplitude and phase when a cyclical component of an input series is used to predict the corresponding component of an output series. This enables the analyst to separate long-term movements from short-term movements.

Without a doubt, the most powerful and sophisticated methodology for forecasting univariate series is the ARIMA modeling methodology popularized by Box and Jenkins (1976). A flexible class of models is introduced, and one member of the class is fit to the historic data. The model is used to forecast the series. Seasonal data can be accommodated, and seasonality can be local. That is, seasonality for month t can be closely related to seasonality for this same month one or two years previously, but less closely related to seasonality for this same month several years previously. Local trending and even long-term upward or downward drifting in the data can be accommodated in ARIMA models through differencing.

Explanatory time series as inputs to a transfer function model can be accommodated. Future values of nondeterministic, independent input series can be forecast by PROC ARIMA, which, unlike the previously mentioned procedures, accounts for the fact that these inputs are forecast when you compute prediction error variances and prediction limits for forecasts. PROC VARMAX models vector processes with possible explanatory variables—the “X” in VARMAX. As in PROC SSM, this approach assumes that at each time point, you observe a vector of responses, each entry of which depends on its own lagged values and lags of the other vector entries. PROC VARMAX allows explanatory variables X and cointegration among the elements of the response vector. Cointegration is an idea that has become popular in recent econometrics. The idea is that each element of the response vector might be a nonstationary process, one that has no tendency to return to a mean or deterministic trend function. Yet one or more linear combinations of the responses are stationary, remaining near some constant. An analogy is two lifeboats adrift in a stormy sea, but tied together by a rope. Their location might be expressible mathematically as a random walk with no tendency to return to a particular point. Over time, the boats drift arbitrarily far from any particular location. Nevertheless, because they are tied together, the difference in their positions would never be too far from 0. Prices of two similar stocks might vary over time according to a random walk with no tendency to return to a given mean. Yet, if they are indeed similar, their price difference might not get too far from 0.

1.2.2 How SAS/ETS Procedures Interrelate

PROC ARIMA emulates PROC AUTOREG if you choose not to model the inputs. ARIMA can fit a richer error structure. Specifically, the error structure can be an autoregressive (AR), moving average (MA), or mixed-model structure. PROC ARIMA can emulate the older PROC FORECAST with METHOD=STEPAR if you use polynomial inputs and AR error specifications. However, unlike PROC FORECAST, PROC ARIMA provides test statistics for the model parameters and checks model adequacy. PROC ARIMA can emulate PROC ESM if you fit a moving average of order d to the d th difference of the data. Instead of arbitrarily choosing a smoothing constant, as was suggested in the earliest years of exponential smoothing research, PROC ESM and the analogies in PROC ARIMA let the data tell you what smoothing constant to use. Furthermore, PROC ARIMA produces more reasonable forecast intervals than many of the simpler procedures. In short, PROC ARIMA does everything the simpler procedures do and does it better. However, more user involvement is required to correctly implement PROC ARIMA.

PROC ESM has some advantages of its own. Among other advantages is the accumulation of forecasts. For example, monthly forecasts can be added to get yearly totals, which is a simple operation. The more difficult part is getting the appropriate standard error for such a sum of autocorrelated data, which PROC ESM does for you.

To benefit from this additional flexibility and sophistication in software, you must have enough expertise and time to analyze the series. You must be able to identify and specify the form of the time series model using the autocorrelations, partial autocorrelations, inverse autocorrelations, and cross-correlations of the time series. Later chapters explain in detail what these terms mean and how to use them. Once you identify a model, fitting and forecasting are almost automatic.

The identification process is more complicated when you use input series. For proper identification, the ARIMA methodology requires that inputs be independent of each other and that there be no feedback from the output series to the input series. For example, if the temperature T_t in a room at time t is to be explained by current and lagged furnace temperatures F_t , lack of feedback corresponds to there being no thermostat in the room. A thermostat causes the furnace temperature to adjust to recent room temperatures. These ARIMA restrictions might be unrealistic in many examples. You can use PROC SSM and PROC VARMAX to model multiple time series without these restrictions.

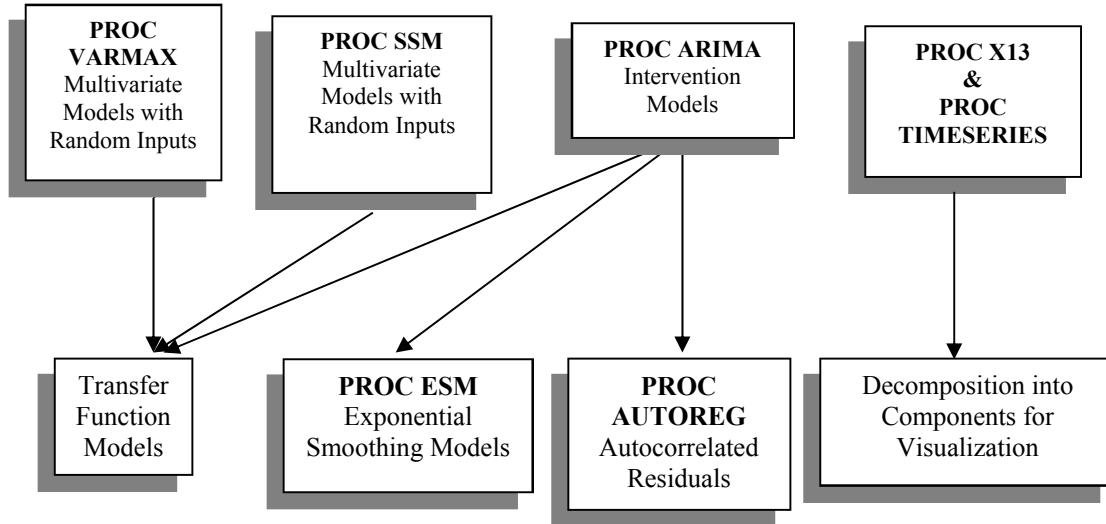
Although PROC SSM and PROC VARMAX are sophisticated in theory, they are fairly easy to run in their default mode. The theory enables you to model several time series together, accounting for relationships of individual component series with current and past values of the other series. Feedback and cross-correlated input series are allowed. Unlike PROC

ARIMA, PROC SSM uses certain prespecified components, many of which are nonstationary unit root processes, thus assuming a form rather than performing the difficult identification process as is required to correctly use PROC ARIMA. The stationarity and unit root concepts are discussed in Chapter 3, “The General ARIMA Model,” where you learn how to make nonstationary series stationary. It is possible to test data for the type of unit root nonstationarity assumed in PROC SSM. Some of the models in the SSM and UCM procedures have repeated unit roots, a condition almost always rejected by statistical tests on real data. Nevertheless, the forecasts from these models are often appealing when visualized and the SSM procedures are often run without checking for the assumed unit root structure.

This chapter introduces some techniques for analyzing and forecasting time series and it lists the SAS procedures for the appropriate computations. As you continue reading the rest of the book, you might want to refer back to this chapter to clarify the relationships among the various procedures.

Figure 1.1 shows the interrelationships among the SAS/ETS procedures mentioned.

Figure 1.1: How SAS/ETS Procedures Interrelate



The following table shows some common questions and answers concerning the procedures.

SAS/ETS Procedures	Q1	Q2	Q3	Q4	Q5	Q6	Q7
FORECAST	T [†]	Y ^{**}	Y	N [§]	Y	Y	Y
AUTOREG	T	Y [‡]	Y	Y	Y	Y	Y
ESM	T	Y	Y	N	Y	Y	Y
X13	T	Y [‡]	Y	Y	Y	N	Y
SPECTRA	F [#]	N [*]	N	N	Y	N	N
ARIMA	T	Y [‡]	Y	Y	N	N	N
SSM	T	Y	Y [‡]	Y	Y	N	N
VARMAX	T	Y	Y	Y	Y	N	N
MODEL	T	Y [‡]	Y	Y	Y	N	Y
SAS Forecast Server Procedures	T	Y	Y	Y	Y	Y	Y

^{*}No

^{**}Yes

[†]Requires user intervention

[‡]Supplied by the program

[§]Time domain analysis

[#]Frequency domain analysis

Q¹ Is a frequency domain analysis (F) or time domain analysis (T) conducted?

Q² Are forecasts automatically generated (or are they generated with a statement)?

Q³ Do predicted values have 95% confidence limits?

Q⁴ Can you supply leading indicator variables or explanatory variables?

Q⁵ Does the procedure run with little user intervention?

Q⁶ Is minimal time series background required for implementation?

Q⁷ Does the procedure handle series with embedded missing values?

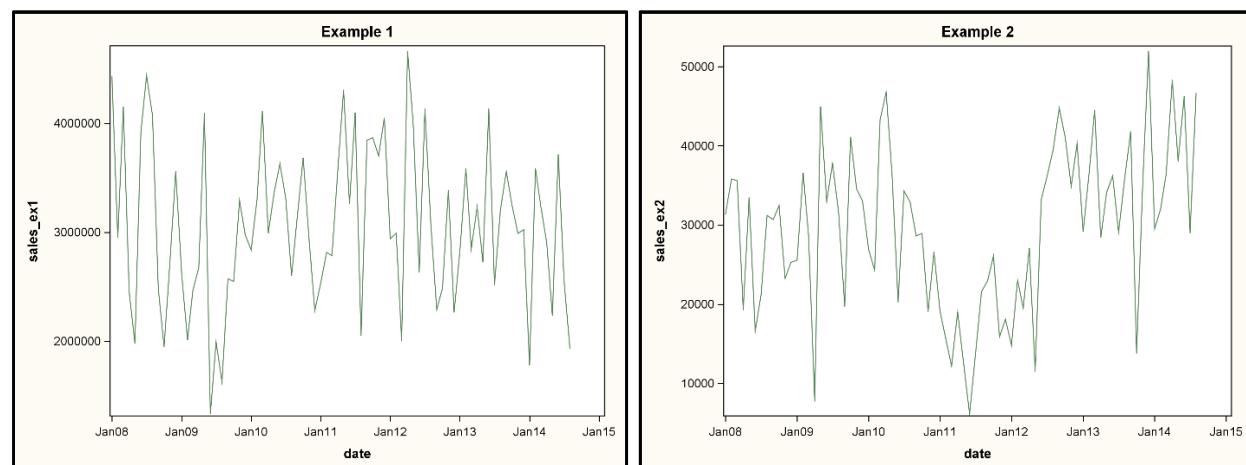
1.3 Simple Models: Regression

Perhaps the most common tool in a statistician's toolbox is regression. Regression offers methodology that can be applied to time series problems as well.

1.3.1 Linear Regression

This section introduces linear regression, an elementary but common method of mathematical modeling. Suppose that at time t you observe Y_t . You also observe explanatory variables X_{1t} , X_{2t} , and so on. For example, Y_t could be sales in month t , X_{1t} could be advertising expenditure in month t , and X_{2t} could be competitors' sales in month t . **Output 1.1** shows simple plots of monthly sales versus date for two generated series.

Output 1.1: Producing Simple Plots of Monthly Data



A multiple linear regression model relating the variables is as follows:

$$Y_t = \beta_0 + \beta_1 X_{1t} + \beta_2 X_{2t} + \varepsilon_t$$

For this model, assume the following characteristics about the errors:

- They have the same variance at all times t .
- They are uncorrelated with each other (ε_t and ε_s are uncorrelated for t different from s).
- They have a normal distribution.

These assumptions enable you to use standard regression methodology, such as PROC REG or PROC GLM. For example, suppose you have 80 observations and you issue the following statements:

```
title "Predicting Sales Using Advertising";
title2 "Expenditures and Competitors' Sales";
proc reg data=sales_ex1;
  model sales_ex1=adv comp / dw;
  output out=out1 p=p r=r;
run;
```

The statements produce **Output 1.2** for the example 1 data.

Output 1.2: Performing a Multiple Regression: Predicting Sales from Advertising Expenditures and Competitors' Sales

The REG Procedure
Model: MODEL1
Dependent Variable: sales_ex1

Number of Observations Read	80
Number of Observations Used	80

Analysis of Variance					
Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	2	2.526182E13	1.263091E13	51.14	<.0001
Error	77	1.901816E13	2.469891E11		
Corrected Total	79	4.427998E13			

Root MSE	496980	R-Square	0.5705
Dependent Mean	3064723	Adj R-Sq	0.5593
Coeff Var	16.21615		

Parameter Estimates					
Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > t
Intercept	1	2700165	373957	7.22	<.0001
ADV	1	10.17968	1.91705	5.31	<.0001
COMP	1	-0.60561	0.08465	-7.15	<.0001

Durbin-Watson D	1.394	❸
Number of Observations	80	
1st Order Autocorrelation	0.283	❹

In this output, the estimates of β_0 , β_1 , and β_2 ❶ are shown. The standard errors ❷ are incorrect if the assumptions on ε_t are not satisfied. You have created an output data set called OUT1 and have called for the Durbin-Watson option to check these error assumptions. The test statistics (❸ and ❹) produced by PROC REG are designed specifically to detect departures from the null hypothesis (H_0 : ε_t uncorrelated) of the following form:

$$H_1: \varepsilon_t = \rho \varepsilon_{t-1} + e_t$$

Here, $|\rho| < 1$ and e_t is an uncorrelated series. This type of error term, in which ε_t is related to ε_{t-1} , is called an AR (autoregressive) error of the first order.

The Durbin-Watson option in the MODEL statement produces the Durbin-Watson test statistic ❸:

$$d = \sum_{t=2}^n (\hat{\varepsilon}_t - \hat{\varepsilon}_{t-1})^2 / \sum_{t=1}^n \hat{\varepsilon}_t^2$$

where $\hat{\varepsilon}_t = Y_t - \hat{\beta}_0 - \hat{\beta}_1 X_{1t} - \hat{\beta}_2 X_{2t}$.

If the actual errors are uncorrelated, the numerator of d has an expected value of about $2(n - 1)\sigma^2$. The denominator has an expected value of approximately $n\sigma^2$. Thus, if the errors ε_t are uncorrelated, the ratio d should be approximately 2.

Positive autocorrelation means that ε_t is closer to ε_{t-1} than in the independent case, so $|\varepsilon_t - \varepsilon_{t-1}|$ should be smaller. It follows that d should also be smaller. The smallest possible value for d is 0. If d is significantly less than 2, positive autocorrelation is present.

When is a Durbin-Watson statistic significant? The answer depends on the number of coefficients in the regression and on the number of observations. In this case, you have $k = 3$ coefficients (β_0 , β_1 , and β_2 for the intercept, ADV, and COMP) and $n = 80$ observations. In general, if you want to test for positive autocorrelation at the 5% significance level, you must compare $d = 1.349$ to a critical value. Even with k and n fixed, the critical value can vary depending on actual values of the independent variables. The results of Durbin and Watson imply that if $k = 3$ and $n = 80$, the critical value must be between $d_L = 1.59$ and $d_U = 1.69$. Because d is less than d_L , you would reject the null hypotheses of uncorrelated errors in favor of the alternative—positive autocorrelation. If $d > 2$, which is evidence of negative autocorrelation, compute $d' = 4 - d$ and compare the results to d_L and d_U . Specifically, if d' (1.954) were greater than 1.69, you would be unable to reject the null hypothesis of uncorrelated errors. If d' were less than 1.59, you would reject the null hypothesis of uncorrelated errors in favor of the alternative—negative autocorrelation. If $1.59 < d < 1.69$, then you cannot be sure whether d is to the left or right of the actual critical value c , because you know only that $1.59 < c < 1.69$.

Durbin and Watson have constructed tables of bounds for the critical values. Most tables use $k' = k - 1$, which equals the number of explanatory variables, excluding the intercept and n (number of observations) to obtain the bounds d_L and d_U for any given regression (Draper and Smith 1998).

Three warnings apply to the Durbin-Watson test. First, it is designed to detect first-order AR errors. Although this type of autocorrelation is only one possibility, it seems to be the most common. The test has some power against other types of autocorrelation. Second, the Durbin-Watson bounds do not hold when lagged values of the dependent variable appear on the right side of the regression. Thus, if the example had used last month's sales to help explain this month's sales, you would not know correct bounds for the critical value. Third, if you incorrectly specify the model, the Durbin-Watson statistic often lies in the critical region, even though no real autocorrelation is present. Suppose an important variable, such as X_{3t} = product availability, had been omitted in the sales example. This omission could produce a significant d . Some practitioners use d as a lack-of-fit statistic, which is justified only if you assume a priori that a correctly specified model cannot have autocorrelated errors and that significance of d must be due to lack of fit.

The output produced a first-order autocorrelation ❹ denoted as $\hat{\rho} = 0.283$. When n is large and the errors are uncorrelated, then the following expression is approximately distributed as a standard normal variate:

$$n^{1/2} \hat{\rho} / (1 - \hat{\rho}^2)^{1/2}$$

Thus, a value $n^{1/2} \hat{\rho} / (1 - \hat{\rho}^2)^{1/2}$ exceeding 1.645 is significant evidence of positive autocorrelation at the 5% significance level. This is especially helpful when the number of observations exceeds the largest in the Durbin-Watson table, as in this example:

$$\sqrt{80}(0.283) / \sqrt{1 - 0.283^2} = 2.639$$

You should use this test only for large n values. It is subject to the three warnings given for the Durbin-Watson test. Because of the approximate nature of the $n^{1/2}\hat{\rho}/(1 - \hat{\rho}^2)^{1/2}$ test, the Durbin-Watson test is preferable. In general, d is approximately $2(1 - \hat{\rho})$. This is easily seen by noting the following:

$$\hat{\rho} = \sum \hat{\varepsilon}_t \hat{\varepsilon}_{t-1} / \sum \hat{\varepsilon}_t^2$$

and

$$d = \sum (\hat{\varepsilon}_t - \hat{\varepsilon}_{t-1})^2 / \sum \hat{\varepsilon}_t^2$$

Durbin and Watson gave a computer-intensive way to compute exact p -values for their test statistic d . This has been incorporated in PROC AUTOREG. For the sales data in example 2, you issue this code to fit a model for sales as a function of this-period and last-period advertising.

```
proc autoreg data=sales_ex2;
  model sales_ex2=adv adv1 / dwprob;
run;
```

The resulting **Output 1.2a** shows a significant $d = 0.5427$ (p -value $0.0001 < 0.05$). Could this be because of an omitted variable? Try the model with competitors' sales included:

```
proc autoreg data=sales_ex2;
  model sales_ex2=adv adv1 comp / dwprob;
run;
```

Now, in **Output 1.2b**, $d = 1.8728$ is insignificant (p -value $0.2239 > 0.05$). Note the increase in R -square (the proportion of variation explained by the model) from 39% to 82%. What is the effect of an increase of \$1 in advertising expenditure? It gives a sales increase estimated at \$6.04 this period, but a decrease of \$5.18 next period. You wonder if the true coefficients on ADV and ADV1 are the same with opposite signs. That is, you wonder if these coefficients add to 0. If they do, then the increase that you get this period from advertising is followed by a decrease of equal magnitude next period. This means the advertising dollar simply shifts the timing of sales rather than increasing the level of sales. Having no autocorrelation evident, you fit the model in PROC REG, asking for a test that the coefficients of ADV and ADV1 add to 0:

```
proc reg data = sales_ex2;
  model sales_ex2 = adv adv1 comp / dw;
  temp: test adv+adv1=0;
run;
```

Output 1.2c gives the results. The regression is exactly that given by PROC AUTOREG with no NLAG= specified. The p -value ($0.0766 > 0.05$) is not small enough to reject the hypothesis that the coefficients are of equal magnitude. It is possible that advertising just shifts the timing, which is a temporary effect. Note the label TEMP on the test.

Although you might have information about your company's plans to advertise, you would likely not know what your competitors' sales will be in future months. At best, you would have to substitute estimates of these future values in forecasting your sales. It appears that an increase of \$1.00 in your competitors' sales is associated with a \$0.56 decrease in your sales.

From **Output 1.2c**, the forecasting equation is as follows:

$$\text{PREDICTED SALES} = 35967 - 0.56323\text{COMP} + 6.03820\text{ADV} - 5.18838\text{ADV1}$$

Output 1.2a: Predicting Sales from Advertising Expenditures**The AUTOREG Procedure**

Dependent Variable	sales_ex2
---------------------------	------------------

Ordinary Least Squares Estimates			
SSE	5164550166	DFE	77
MSE	67072080	Root MSE	8190
SBC	1678.82082	AIC	1671.67474
MAE	6642.78606	AICC	1671.99053
MAPE	29.9747012	HQC	1674.53981
Durbin-Watson	0.5427	Regress R-square	0.3866
		Total R-square	0.3866

Durbin-Watson Statistic			
Order	DW	Pr < DW	Pr > DW
1	0.5427	<.0001	1.0000

NOTE: Pr<DW is the p-value for testing positive autocorrelation, and Pr>DW is the p-value for testing negative autocorrelation.

Parameter Estimates					
Variable	DF	Estimate	Standard Error	t Value	Approx Pr > t
Intercept	1	14466	8532	1.70	0.0940
ADV	1	6.5601	0.9641	6.80	<.0001
ADV1	1	-5.0152	0.9606	-5.22	<.0001

Output 1.2b: Predicting Sales from Advertising Expenditures and Competitors' Sales**The AUTOREG Procedure**

Dependent Variable	sales_ex2
---------------------------	------------------

Ordinary Least Squares Estimates			
SSE	1487719368	DFE	76
MSE	19575255	Root MSE	4424
SBC	1583.63695	AIC	1574.10885
MAE	3522.97712	AICC	1574.64218
MAPE	14.280071	HQC	1577.92894
Durbin-Watson	1.8728	Regress R-Square	0.8233
		Total R-Square	0.8233

Durbin-Watson Statistics				
Order	DW	Pr < DW	Pr > DW	
1	1.8728	0.2239	0.7761	

NOTE: Pr<DW is the p-value for testing positive autocorrelation, and Pr>DW is the p-value for testing negative autocorrelation.

Parameter Estimates					
Variable	DF	Estimate	Standard Error	t Value	Approx Pr > t
Intercept	1	35967	4869	7.39	<.0001
ADV	1	6.0382	0.5222	11.56	<.0001
ADV1	1	-5.1884	0.5191	-9.99	<.0001
COMP	1	-0.5632	0.0411	-13.71	<.0001

Output 1.2c: Predicting Sales from Advertising Expenditures and Competitors' Sales

The REG Procedure

Model: MODEL1

Dependent Variable: sales_ex2

Number of Observations Read	80
Number of Observations Used	80

Analysis of Variance					
Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	3	6931264991	2310421664	118.03	<.0001
Error	76	1487719368	19575255		
Corrected Total	79	8418984359			

Root MSE	4424.39316	R-Square	0.8233
Dependent Mean	29630	Adj R-Sq	0.8163
Coeff Var	14.93203		

Parameter Estimates					
Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > t
Intercept	1	35967	4869.00487	7.39	<.0001
ADV	1	6.03820	0.52224	11.56	<.0001
ADV1	1	-5.18838	0.51913	-9.99	<.0001
COMP	1	-0.56323	0.04110	-13.71	<.0001

Durbin-Watson D	1.873
Number of Observations	80
1st Order Autocorrelation	0.044

Test TEMPR Results for Dependent Variable SALES				
Source	DF	Mean Square	F Value	Pr > F
Numerator	1	63103884	3.22	0.0766
Denominator	76	19575255		

1.3.2 Highly Regular Seasonality

Occasionally, a very regular seasonality occurs in a series, such as an average monthly temperature at a given location. In this case, you can model seasonality by computing means. Specifically, the mean of all the January observations estimates the seasonal level for January. Similar means are used for other months throughout the year. An alternative to computing the 12 means is to run a regression on monthly indicator variables. An indicator variable has values of 0 or 1. For the January indicator, the 1s occur only for observations made in January. You can compute an indicator variable for each month, and regress Y_t on the 12 indicators with no intercept. You can also regress Y_t on a column of 1s and 11 of the indicator variables. The intercept now estimates the level for the month associated with the omitted indicator. The coefficient of any indicator column is added to the intercept to compute the seasonal level for that month.

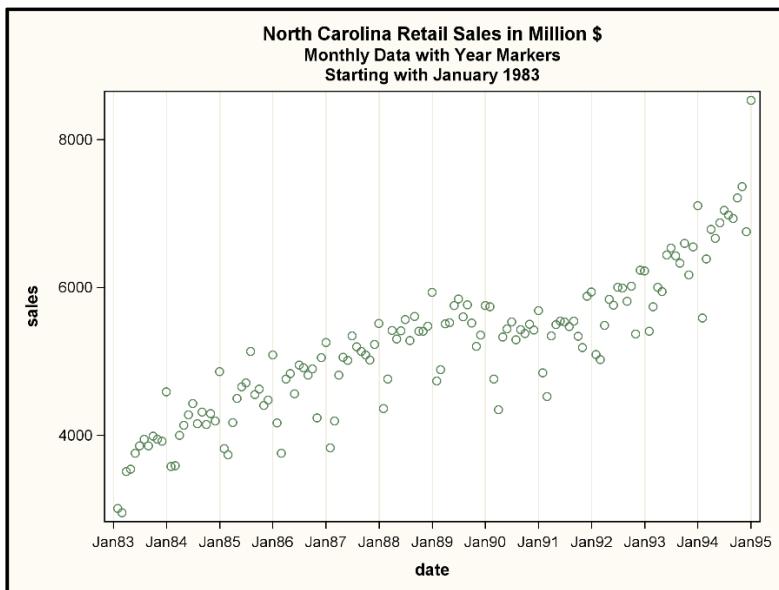
For further illustration, **Output 1.3** shows a series of quarterly increases in North Carolina retail sales. That is, each point is the sales for that quarter minus the sales for the previous quarter. **Output 1.4** shows a plot of the monthly sales through time. Quarterly sales were computed as averages of three consecutive months and are used to make the presentation brief. A model for the monthly data is shown in Chapter 4. There is a strong seasonal pattern and perhaps a mild trend over time. The change data are plotted in **Output 1.6**. To model the seasonality, use S1, S2, and S3. For the trend, use time, T1, and its square, T2. The S variables are often referred to as indicator variables, being indicators of the season, or as dummy variables. The first CHANGE value is missing because the sales data start in quarter 1 of 1983, so no increase can be computed for that quarter.

Output 1.3: Displaying North Carolina Retail Sales Data Set

Obs	DATE	CHANGE	S1	S2	S3	T1	T2
1	83Q1	.	1	0	0	1	1
2	83Q2	1678.41	0	1	0	2	4
3	83Q3	633.24	0	0	1	3	9
4	83Q4	662.35	0	0	0	4	16
5	84Q1	-1283.59	1	0	0	5	25

(Additional output omitted)

47	94Q3	543.61	0	0	1	47	2209
48	94Q4	1526.95	0	0	0	48	2304

Output 1.4: Plotting North Carolina Monthly Sales

Issue these commands:

```
proc autoreg data=all;
  model change = t1 t2 s1 s2 s3 / dwprob;
run;
```

This gives **Output 1.5**.

Output 1.5: Using PROC AUTOREG to Get the Durbin-Watson Statistic**The AUTOREG Procedure**

Dependent Variable	CHANGE
---------------------------	--------

Ordinary Least Squares Estimates			
SSE	5290127.6	DFE	41
MSE	129028	Root MSE	359.20398
SBC	703.147758	AIC	692.046872
MAE	267.796224	AICC	694.146872
MAPE	239.274858	HQC	696.22421
Durbin-Watson	2.3770	Regress R-Square	0.9221
		Total R-Square	0.9221

Durbin-Watson Statistics			
Order	DW	Pr < DW	Pr > DW
1	2.3770	0.8608	0.1392

NOTE: Pr<DW is the p-value for testing positive autocorrelation, and Pr>DW is the p-value for testing negative autocorrelation.

Parameter Estimates					
Variable	DF	Estimate	Standard Error	t Value	Approx Pr > t
Intercept	1	679.4273	200.1247	3.40	0.0015
T1	1	-44.9929	16.4428	-2.74	0.0091
T2	1	0.9915	0.3196	3.10	0.0035
S1	1	-1726	150.3312	-11.48	<.0001
S2	1	1504	146.8483	10.24	<.0001
S3	1	-221.2871	146.6958	-1.51	0.1391

PROC AUTOREG is intended for regression models with autoregressive errors. The following is an example of a model with autoregressive errors:

$$Y_t = \beta_0 + \beta_1 X_{1t} + \beta_2 X_{2t} + Z_t$$

where $Z_t = \rho Z_{t-1} + e_t$.

The error term Z_t is related to a lagged value of itself in an equation that resembles a regression equation—hence the term “autoregressive.” The term e_t represents the portion of Z_t that could not have been predicted from previous Z values. This is often called an unanticipated shock or white noise. It is assumed that the e series is independent and identically distributed. This one lag error model is fit using the NAG=1 option in the MODEL statement. Alternatively, the options /NLAG=5 BACKSTEP can be used to try 5 lags of Z , automatically deleting those deemed statistically insignificant.

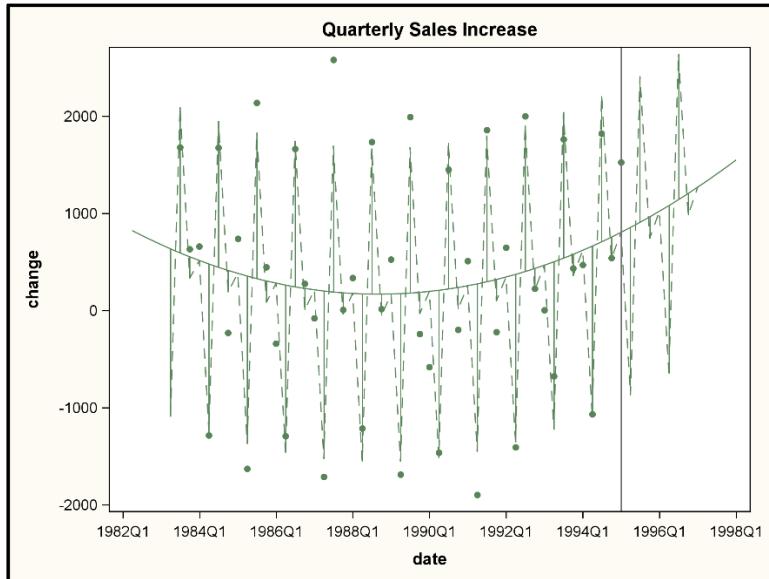
The retail sales change data require no autocorrelation adjustment. The Durbin-Watson test has a p -value $0.8608 > 0.05$. There is no evidence of positive autocorrelation in the errors. The statistic is not too close to 0. Note that $1 - 0.8608 = 0.1392$, so there is also no evidence for negative autocorrelation. The fitting of the model is the same as in PROC REG because no NLAG specification was issued in the MODEL statement. The parameter estimates are interpreted just as they would be in PROC REG. That is, the predicted change, PC, in quarter 4 (where S1 = S2 = S3 = 0) is given by the following:

$$PC = 679.4273 - 44.9929t + 0.9915t^2$$

In quarter 1 (where S1 = 1, S2 = S3 = 0), the PC is given by the following:

$$PC = 679.4273 - 1726 - 44.9929t + 0.9915t^2$$

The coefficients of S1, S2, and S3 represent shifts in the quadratic polynomial associated with the first through third quarters. The remaining coefficients calibrate the quadratic function to the fourth quarter level. In **Output 1.6**, the data are dots, and the fourth quarter quadratic predicting function is the smooth curve. Vertical lines extend from the quadratic, indicating the seasonal shifts required for the other three quarters. The broken line gives the predictions. The last data point for 1994Q4 is indicated with an extended vertical line. The shift for any quarter is the same every year. This is a property of the dummy variable model and might not be reasonable for some data (for example, sometimes seasonality is slowly changing over a period of years).

Output 1.6: Plotting Quarterly Sales Increases with Quadratic Predicting Function

To forecast into the future, extrapolate the linear and quadratic terms and the seasonal dummy variables the requisite number of periods. The data set EXTRA listed in **Output 1.7** contains these values. There is no question about the future values of these, unlike the case of competitors' sales that was considered in an earlier example. The PROC AUTOREG technology assumes perfectly known future values of the explanatory variables. Set the response variable, CHANGE, to missing.

Output 1.7: Data Appended for Forecasting

Obs	DATE	CHANGE	S1	S2	S3	T1	T2
1	95Q1	.	1	0	0	49	2401
2	95Q2	.	0	1	0	50	2500
3	95Q3	.	0	0	1	51	2601
4	95Q4	.	0	0	0	52	2704
5	96Q1	.	1	0	0	53	2809
6	96Q2	.	0	1	0	54	2916
7	96Q3	.	0	0	1	55	3025
8	96Q4	.	0	0	0	56	3136

Combine the original data set—call it NCSALES—with the data set EXTRA:

```
data all;
  set ncsales extra;
run;
```

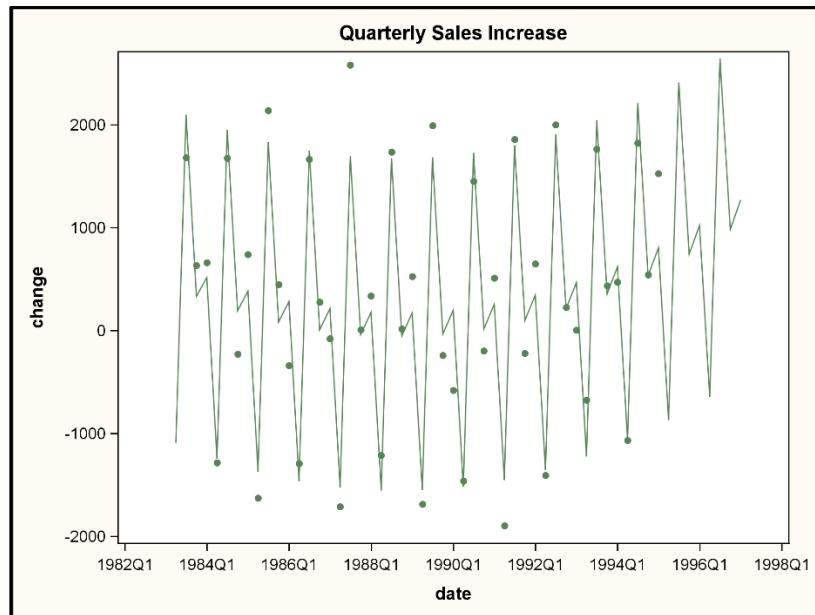
Run PROC AUTOREG on the combined data, noting that the EXTRA data cannot contribute to the estimation of the model parameters because CHANGE is missing. The EXTRA data have full information about the explanatory variables. Predicted values (forecasts) are produced. The predicted values P are output into a data set OUT1 using this statement in PROC AUTOREG:

```
output out=out1 pm=p;
```

Using PM= requests that the predicted values be computed only from the regression function without forecasting the error term Z . If NLAG= is specified, a model is fit to the regression residuals. This model can be used to forecast residuals into the future. Replacing PM= with P= adds forecasts of future Z values to the forecast of the regression function. The two types of forecast, with and without forecasting the residuals, point out the fact that part of the predictability comes from the explanatory variables and part comes from the autocorrelation (that is, from the momentum of the series). Thus, as seen in **Output 1.5**, there is a total R -square and a regression R -square, the latter measuring the predictability associated with the explanatory variables apart from contributions due to autocorrelation. In

the current example, with no autoregressive lags specified, these are the same, and P= and PM= create the same variable. The predicted values from PROC AUTOREG using data set ALL are displayed in **Output 1.8**.

Output 1.8: Plotting Quarterly Sales Increase with Prediction



Because this example shows no residual autocorrelation, analysis in PROC REG would be appropriate. Using the data set with the extended explanatory variables, add P and CLI to produce predicted values and associated prediction intervals.

```
proc reg data=all;
  model change = t t2 s1 s2 s3 / p cli;
  title "Quarterly Sales Increase";
run;
```

Output 1.9: Producing Forecasts and Prediction Intervals with the P and CLI Options in the Model Statement

QUARTERLY SALES INCREASE

The REG Procedure

Model: MODEL1

Dependent Variable: CHANGE

Number of Observations Read	56
Number of Observations Used	47
Number of Observations with Missing Values	9

Analysis of Variance					
Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	5	62618901	12523780	97.06	<.0001
Error	41	5290128	129028		
Corrected Total	46	67909029			

Root MSE	359.20398	R-Square	0.9221
Dependent Mean	280.25532	Adj R-Sq	0.9126
Coeff Var	128.17026		

Parameter Estimates					
Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > t
Intercept	1	679.42728	200.12467	3.40	0.0015
T1	1	-44.99289	16.44278	-2.74	0.0091
T2	1	0.99152	0.31963	3.10	0.0035
S1	1	-1725.83250	150.33121	-11.48	<.0001
S2	1	1503.71785	146.84832	10.24	<.0001
S3	1	-221.28706	146.69576	-1.51	0.1391

Output Statistics						
Obs	Dependent Variable	Predicted Value	Std Error Mean Predict	95% CL Predict		Residual
1	.	-1090	195.0057	-1916	-264.9732	.
2	1678	2097	172.1016	1293	2902	-418.7154
3	633.2400	332.0852	163.6584	-465.0876	1129	301.1548
4	662.3500	515.3200	156.0279	-275.5880	1306	147.0300
5	-1284	-1247	153.6192	-2036	-457.5991	-37.0083

(Additional output omitted)

49	.	-870.4182	195.0057	-1696	-44.9848	.
50	.	2412	200.1247	1582	3243	.
51	.	742.4454	211.9675	-99.8696	1585	.
52	.	1021	224.4171	165.4992	1876	.
53	.	-645.8498	251.4732	-1531	239.6822	.
54	.	2645	259.4076	1750	3540	.
55	.	982.8781	274.9919	69.2774	1896	.
56	.	1269	291.0057	335.6179	2203	.

Sum of Residuals	0
Sum of Squared Residuals	5290128
Predicted Residual SS (PRESS)	7067796

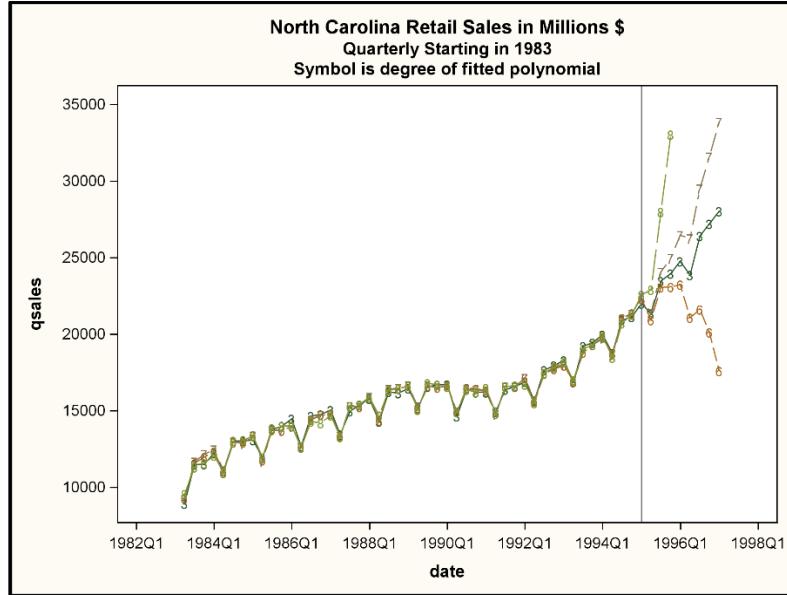
For observation 49, an increase in sales of -870.4182 (in essence, a decrease) is predicted for the next quarter with confidence intervals extending from -1696 to -44.9848. This is the typical after-Christmas sales slump.

What does this sales change model say about the levels of sales, and why were the levels of sales not used in the analysis? First, a cubic term in time, bt^3 , when differenced, becomes a quadratic term: $bt^3 - b(t-1)^3 = b(3t^2 - 3t + 1)$. A quadratic plus seasonal model in the differences is associated with a cubic plus seasonal model in the levels. However, if the error term in the differences satisfies the usual regression assumptions, which it seems to do for these data, then the error term in the original levels cannot possibly satisfy them. The levels appear to have a nonstationary error term. Ordinary regression statistics are invalid on the original level series. If you ignore this, the usual (incorrect here) regression statistics indicate that a degree 8 polynomial is required to get a good fit.

A plot of sales and the forecasts from polynomials of varying degree is shown in **Output 1.10**. The degree 8 polynomial, arrived at by inappropriate use of ordinary regression, gives a ridiculous forecast that extends vertically beyond the range of this graph just a few quarters into the future. The degree 3 polynomial gives a reasonable increase and the intermediate degree 6 polynomial actually forecasts a decrease. It is dangerous to forecast too far into the future using

polynomials, especially those of high degree. Time series models specifically designed for nonstationary data are discussed later. In summary, the differenced data seem to satisfy assumptions needed to justify regression.

Output 1.10: Plotting Sales and Forecasts of Polynomials of Varying Degree



1.3.3 Regression with Transformed Data

Often, you analyze some transformed version of the data rather than the original data. The logarithmic transformation is probably the most common, and it is the only transformation discussed in this book. Box and Cox (1964) suggest a family of transformations and a method of using the data to select one of them. This is discussed in the time series context in Box and Jenkins (1976, 1994).

Consider the following model:

$$Y_t = \beta_0 (\beta_1^{X_t}) \varepsilon_t$$

Taking logarithms on both sides, you obtain the following:

$$\log(Y_t) = \log(\beta_0) + \log(\beta_1) X_t + \log(\varepsilon_t)$$

Now, if $\log(\varepsilon_t)$ satisfies the standard regression assumptions, the regression of $\log(Y_t)$ on 1 and X_t produces the best estimates of $\log(\beta_0)$ and $\log(\beta_1)$.

As before, if the data consist of $(X_1, Y_1), (X_2, Y_2), \dots, (X_n, Y_n)$, then you can append future known values $X_{n+1}, X_{n+2}, \dots, X_{n+s}$ to the data if they are available and compute `ly=log(y)` in the DATA step. Set Y_{n+1} through Y_{n+s} to missing values (.). Use the MODEL statement in PROC REG:

```
model ly=x / p cli;
```

This produces predictions of future LY values and prediction limits for them. If, for example, you obtain an interval

$$-1.13 < \log(Y_{n+s}) < 2.7$$

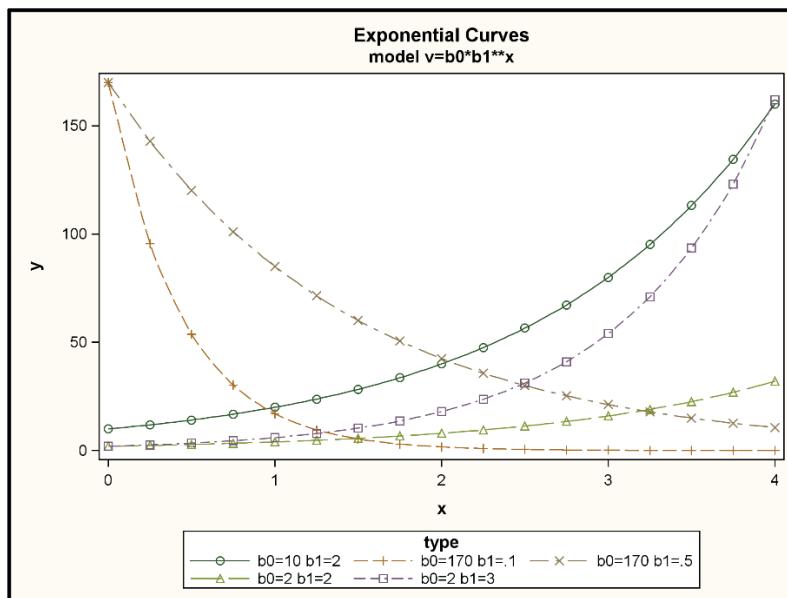
then you can compute $\exp(-1.13) = 0.323$ and $\exp(2.7) = 14.88$ to conclude that $0.323 < Y_{n+s} < 14.88$.

The original prediction interval had to be computed on the log scale, the only scale on which you can justify a t distribution or normal distribution.

When should you use logarithms? A quick check is to plot Y against X . When $Y_t = \beta_0 (\beta_1^{X_t}) \varepsilon_t$, then the overall shape of the plot resembles that of $Y = \beta_0 (\beta_1^X)$.

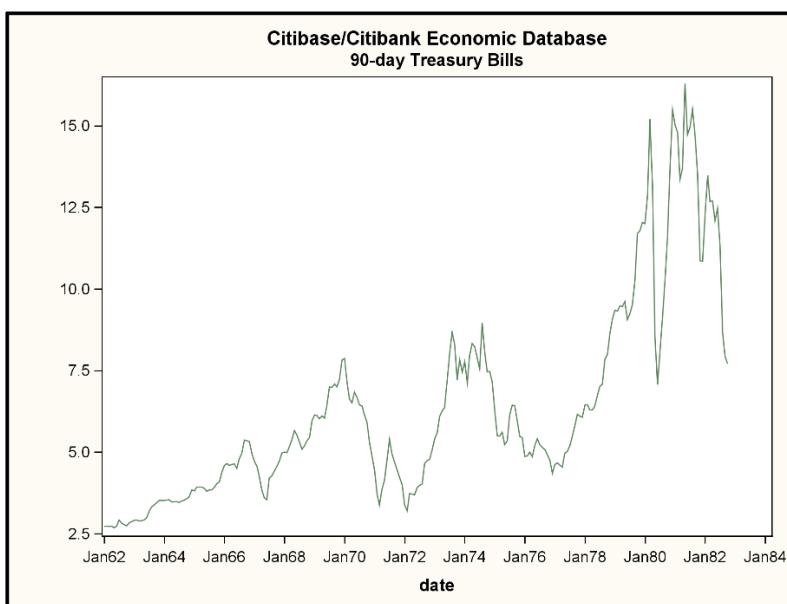
See **Output 1.11** for several examples of this type of plot. The curvature in the plot becomes more dramatic as β_1 moves away from 1 in either direction. The actual points are scattered around the appropriate curve. Because the error term ε is multiplied by $\beta_0(\beta_1^X)$, the variation around the curve is greater at the higher points and lesser at the lower points on the curve.

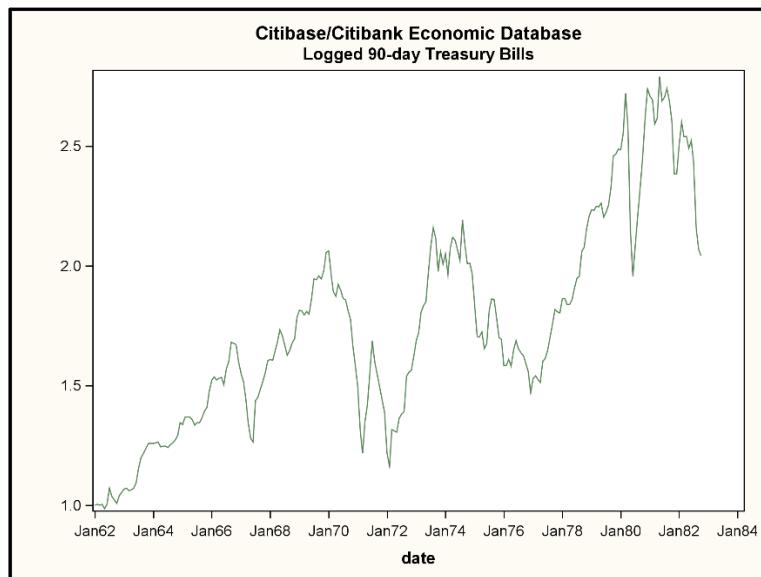
Output 1.11: Plotting Exponential Curves



Output 1.12 shows a plot of the U.S. Treasury bill rates against time. The curvature and especially the variability are similar to those just described. In this case, you simply have $X_t = t$. A plot of the logarithm of the rates appears in **Output 1.13**. Because this plot is straighter with more uniform variability, you decide to analyze the logarithms.

Output 1.12: Plotting 90-Day Treasury Bill Rates



Output 1.13: Plotting 90-Day Logged Treasury Bill Rates

To analyze and forecast the series with simple regression, you first create a data set with future values of time:

```
data tbills2;
  set tbills end=eof;
  time+1;
  output;
  if eof then do i=1 to 24;
    lfym3=.;
    time+1;
    date=intnx('month',date,1);
    output;
  end;
  drop i;
run;
```

Output 1.14 shows the last 24 observations of the data set TBILLS2. You regress the log Treasury bill rate, LFYGM3, on TIME to estimate $\log(\beta_0)$ and $\log(\beta_1)$ in the following model:

$$\text{LFYGM3} = \log(\beta_0) + \log(\beta_1) \times \text{TIME} + \log(\varepsilon_i)$$

Output 1.14: Displaying Future Date Values for U.S. Treasury Bill Data

Obs	DATE	LFYGM3	TIME
1	NOV82	.	251
2	DEC82	.	252
3	JAN83	.	253
4	FEB83	.	254
5	MAR83	.	255

(Additional output omitted)

20	JUN84	.	270
21	JUL84	.	271
22	AUG84	.	272
23	SEP84	.	273
24	OCT84	.	274

You also produce predicted values and check for autocorrelation by using these SAS statements:

```
proc reg data=tbills2;
  model lfygm3=time / dw p cli;
  id date;
  title 'Citibase/Citibank Economic Database';
  title2 'Regression with Transformed Data';
run;
```

The result is shown in **Output 1.15**.

Output 1.15: Producing Predicted Values and Checking Autocorrelation with the P, CLI, and DW Options in the MODEL Statement on the Log-Transformed Data

The REG Procedure
Model: MODEL1
Dependent Variable: LFYGM3

Number of Observations Read	274
Number of Observations Used	250
Number of Observations with Missing Values	24

Analysis of Variance					
Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	1	32.68570	32.68570	540.63	<.0001
Error	248	14.99365	0.06046		
Corrected Total	249	47.67935			

Root MSE	0.24588	R-Square	0.6855
Dependent Mean	1.74783	Adj R-Sq	0.6843
Coeff Var	14.06788		

Parameter Estimates					
Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > t
Intercept	1	1.11904	0.03120	35.87	<.0001
TIME	1	0.00501	0.00021548	23.25	<.0001

Durbin-Watson D	0.090	❶
Number of Observations	250	❷
1st Order Autocorrelation	0.951	❸

Output Statistics							
Obs	DATE	Dependent Variable	Predicted Value	Std Error Mean Predict	95% CL Predict	Residual	
1	JAN62	1.0006	1.1240	0.0310	0.6359	1.6122	-0.1234
2	FEB62	1.0043	1.1291	0.0308	0.6410	1.6171	-0.1248
3	MAR62	1.0006	1.1341	0.0306	0.6460	1.6221	-0.1334
4	APR62	1.0043	1.1391	0.0305	0.6511	1.6271	-0.1348
5	MAY62	0.9858	1.1441	0.0303	0.6562	1.6320	-0.1583

(Additional output omitted)

251	NOV82	.	2.3766	0.0312	1.8885	2.8648	.
-----	-------	---	--------	--------	--------	--------	---

(Additional output omitted)

270	JUN84	.	2.4718	0.0348	1.9827	2.9609	.
271	JUL84	.	2.4768	0.0350	1.9877	2.9660	.
272	AUG84	.	2.4818	0.0352	1.9926	2.9711	.
273	SEP84	.	2.4868	0.0354	1.9976	2.9761	.
274	OCT84	.	2.4919	0.0356	2.0025	2.9812	.

Sum of Residuals	0
Sum of Squared Residuals	14.99365
Predicted Residual SS (PRESS)	15.21335

Now, for example, you compute confidence limits as follows:

$$1.11904 - (1.96)(0.03120) < \log(\beta_0) < 1.11904 + (1.96)(0.03120)$$

Thus,

$$2.880 < \beta_0 < 3.255$$

is a 95% confidence interval for β_0 . Similarly, $1.0046 < \beta_1 < 1.0054$ is a 95% confidence interval for β_1 .The growth rate of Treasury bills is estimated from this model to be between 0.46% and 0.54% per time period. Your forecast for November 1982 can be obtained from $1.8885 < 2.3766 < 2.8648$ so that

$$6.61 < \text{FYGM3}_{251} < 17.55$$

is a 95% prediction interval for the November 1982 yield, and $\exp(2.3766) = 10.77$ is the predicted value. Because the distribution on the original levels is highly skewed, the prediction 10.77 does not lie midway between 6.61 and 17.55, nor would you want it to do so.The Durbin-Watson statistic in **Output 1.15** ① is $d = 0.090$. However, because $n = 250$ ② is beyond the range of the Durbin-Watson tables, you use $\hat{\rho} = 0.951$ ③ to compute as follows:

$$n^{1/2}\hat{\rho}/(1-\hat{\rho}^2)^{1/2} = 48.63$$

This is greater than 1.645. At the 5% level, you can conclude that positive autocorrelation is present (or that your model is misspecified in some other way). This is also evident in **Output 1.13**, in which the data fluctuate around the overall trend in a clearly dependent fashion. Therefore, you should recompute your forecasts and confidence intervals using some of the methods in this book that consider autocorrelation.

Suppose $X = \log(Y)$, and X is normal with mean M_x and variance σ_x^2 . Then, $Y = \exp(X)$ and Y has median $\exp(M_x)$ and mean $\exp(M_x + \sigma_x^2/2)$. For this reason, some experts suggest adding half the error variance to a log scale forecast prior to exponentiation. We prefer to simply exponentiate and think of the result—for example, $\exp(2.3766) = 10.77$ —as an estimate of the median, reasoning that this is a more credible central estimate for such a highly skewed distribution.

Chapter 2: Simple Models: Autoregression

2.1 Introduction	23
2.1.1 Terminology and Notation.....	23
2.1.2 Statistical Background	23
2.2 Forecasting	24
2.2.1 PROC ARIMA for Forecasting.....	25
2.2.2 Backshift Notation B for Time Series.....	32
2.2.3 Yule-Walker Equations for Covariances	33
2.3 Fitting an AR Model in PROC REG	37

2.1 Introduction

A simple and yet quite useful model, the order 1 autoregressive, AR(1), model is used in this chapter to introduce some of the basic ideas in time series analysis and forecasting.

2.1.1 Terminology and Notation

Often, you can forecast series Y_t simply based on past values Y_{t-1} , Y_{t-2} , For example, suppose Y_t satisfies the following:

$$Y_t - \mu = \rho(Y_{t-1} - \mu) + e_t \quad (2.1)$$

where e_t is a sequence of uncorrelated $N(0, \sigma^2)$ variables. The term for such an e_t sequence is “white noise.” Assuming equation 2.1 holds at all times t , you can write, for example, $Y_{t-1} - \mu = \rho(Y_{t-2} - \mu) + e_{t-1}$, and, when you substitute in equation 2.1, you obtain $Y_t - \mu = e_t + \rho e_{t-1} + \rho^2(Y_{t-2} - \mu)$. When you continue this way, you obtain the following:

$$Y_t - \mu = e_t + \rho e_{t-1} + \rho^2 e_{t-2} + \cdots + \rho^{t-1} e_1 + \rho^t (Y_0 - \mu) \quad (2.2)$$

If you assume $|\rho| < 1$, then the effect of the series values before you started collecting data (Y_0 , for example) is minimal. Furthermore, you see that the mean (expected value) of Y_t is μ .

Suppose the variance of Y_{t-1} is $\sigma^2/(1 - \rho^2)$. Then the variance of $\rho(Y_{t-1} - \mu) + e_t$ is as follows:

$$\rho^2 \sigma^2 / (1 - \rho^2) + \sigma^2 = \sigma^2 / (1 - \rho^2)$$

This shows that the variance of Y_t is also $\sigma^2/(1 - \rho^2)$.

2.1.2 Statistical Background

You can define Y_t as an accumulation of past shocks e_t to the system by writing the mathematical model shown in model equation 2.1 as follows:

$$Y_t = \mu + \sum_{j=0}^{\infty} \rho^j e_{t-j} \quad (2.3)$$

This is to say, you do so by extending equation 2.2 back into the infinite past, again showing that, if $|\rho| < 1$, then the effect of shocks in the past is minimal. Equation 2.3, in which the series is expressed in terms of a mean and past shocks,

is often called the *Wold representation* of the series. You can also compute a covariance between Y_t and Y_{t-j} from equation 2.3. Calling it $\gamma(j) = \text{cov}(Y_t, Y_{t-j})$, you have the following:

$$\gamma(j) = \rho^{|j|} \sigma^2 / (1 - \rho^2) = \rho^{|j|} \text{var}(Y_t)$$

An interesting feature is that $\gamma(j)$ does not depend on t . In other words, the covariance between Y_t and Y_s depends only on the time distance $|t - s|$ between these observations, and not on the values of t and s .

Why emphasize variances and covariances? They determine which model is appropriate for your data. One way to determine when model equation 2.1 is appropriate is to compute estimates of the covariances of your data and determine whether they are of the given form—that is, whether they decline exponentially at rate ρ as lag j increases. Suppose you observe the following $\gamma(j)$ sequence:

$$\gamma(0) = 243, \gamma(1) = 162, \gamma(2) = 108, \gamma(3) = 72, \gamma(4) = 48, \gamma(5) = 32, \gamma(6) = 21.3, \dots$$

You know the variance of your process, which is as follows:

$$\text{var}(Y_t) = \gamma(0) = 243$$

You note that $\gamma(1)/\gamma(0) = 2/3$. Also, $\gamma(2)/\gamma(1) = 2/3$. And, in fact, $\gamma(j)/\gamma(j-1) = 2/3$ all the way through the sequence. Thus, you decide that model 1 is appropriate and that $\rho = 2/3$. Because $\gamma(0) = \sigma^2 / (1 - \rho^2)$, you also know that the following is true:

$$\sigma^2 = (1 - (2/3)^2)(243) = 135$$

2.2 Forecasting

How does your knowledge of ρ help you forecast? Suppose you know $\mu = 100$ (in practice, you use an estimate such as the mean, \bar{Y} , of your observations). If you have data up to time n , you know that in the discussion in Section 2.1 the following holds:

$$Y_{n+1} - 100 = (2/3)(Y_n - 100) + e_{n+1}$$

At time n , e_{n+1} has not occurred. It is not correlated with anything that has occurred up to time n . You forecast e_{n+1} by its unconditional mean 0. Because Y_n is available, it is easy to compute the forecast of Y_{n+1} as follows:

$$\hat{Y}_{n+1} = 100 + (2/3)(Y_n - 100)$$

You can compute the forecast error as $Y_{n+1} - \hat{Y}_{n+1} = e_{n+1}$. Similarly, you can compute the following:

$$\begin{aligned} Y_{n+2} - 100 &= (2/3)(Y_{n+1} - 100) + e_{n+2} \\ &= (2/3)[(2/3)(Y_n - 100) + e_{n+1}] + e_{n+2} \end{aligned}$$

and you forecast Y_{n+2} as $100 + (2/3)^2(Y_n - 100)$, with forecast error $e_{n+2} + (2/3)e_{n+1}$.

Similarly, for a general μ and ρ , the forecast L steps into the future is $\mu + \rho^L(Y_n - \mu)$, with this error:

$$e_{n+L} + \rho e_{n+L-1} + \dots + \rho^{L-1} e_{n+1}$$

A forecasting strategy now becomes clear. You do the following:

1. Examine estimates of the autocovariances $\gamma(j)$ to see whether they decrease exponentially.
2. If they do, then assume equation 2.1 holds, and estimate μ and ρ .

3. Calculate the prediction $\hat{Y}_{n+L} = \mu + \rho^L (Y_n - \mu)$ and the forecast error variance:

$$\sigma^2 (1 + \rho^2 + \rho^4 + \dots + \rho^{2L-2})$$

You must substitute estimates, such as $\hat{\rho}$, for your parameters.

For example, if $\mu = 100$, $\rho = 2/3$, and $Y_n = 127$, then the forecasts become 118, 112, 108, 105.3, 103.6, 102.4, The forecast error variances, based on $\text{var}(e) = \sigma^2 = 135$, become 135, 195, 221.7, 233.5, 238.8, and 241.1. The forecasts decrease exponentially at the rate $\rho = 2/3$ to the series mean $\mu = 100$. The forecast error variance converges to the following series variance:

$$\sigma^2 / (1 - \rho^2) = 135 / [1 - (2/3)^2] = 243 = \gamma(0)$$

This shows that an equation such as equation 2.1 helps you forecast in the short run, but you might as well use the series mean to forecast a stationary series far into the future.

In this section, $Y_t - \mu = \rho(Y_{t-1} - \mu) + e_t$ was expanded as an infinite sum of past shocks e_t , showing how past shocks accumulate to determine the current deviation of Y from the mean. At time $n + L$, this expansion was as follows:

$$Y_{n+L} - \mu = (e_{n+L} + \rho e_{n+L-1} + \dots + \rho^{L-1} e_{n+1}) + \rho^L (e_n + \rho e_{n-1} + \dots)$$

Substituting $Y_n - \mu = e_n + \rho e_{n-1} + \dots$ shows the following:

- The best (minimum prediction error variance) prediction of Y_{n+L} is $\mu + \rho^L (Y_n - \mu)$.
- The error in that prediction is $(e_{n+L} + \rho e_{n+L-1} + \dots + \rho^{L-1} e_{n+1})$, so the prediction error variance is $\sigma^2 (1 + \rho^2 + \rho^4 + \dots + \rho^{2L-2})$.
- The effect of shocks that happened a long time ago has little effect on the present Y if $|\rho| < 1$.

The future shocks (e 's) in item 2 have not yet occurred, but from the historic residuals, an estimate of σ^2 can be obtained so that the error variance can be estimated and a prediction interval can be calculated. It will be shown that, for a whole class of models called *ARMA models*, such a decomposition of Y_{n+L} into a prediction that is a function of current and past Y s, plus a prediction error that is a linear combination of future shocks (e 's), is possible. The coefficients in these expressions are functions of the model parameters, such as ρ , which can be estimated.

2.2.1 PROC ARIMA for Forecasting

For example, 200 data values (Y_1, Y_2, \dots, Y_{200}) with mean $\bar{Y} = 90.091$ and last observation $Y_{200} = 140.246$ are analyzed with these statements:

```
proc arima data=example plots(only)=series(acf);
  identify var=y center outcov=cov;
  estimate p=1 noint;
  forecast lead=5;
quit;

proc print data=cov;
run;
```

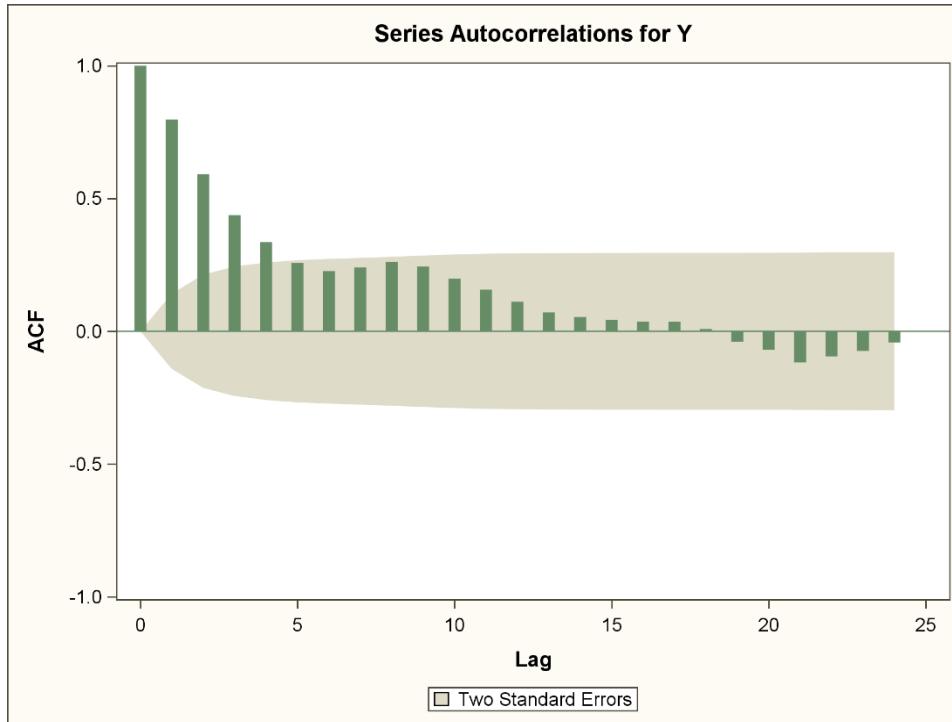
Output 2.1 shows the results when you use PROC ARIMA to identify, estimate, and forecast.

Output 2.1: Using PROC ARIMA to Identify, Estimate, and Forecast

The ARIMA Procedure

Name of Variable = Y	
Mean of Working Series	0
Standard Deviation	34.61987
Number of Observations	200

Autocorrelation Check for White Noise									
To Lag	Chi-Square	DF	Pr > ChiSq	Autocorrelations					
6	287.25	6	<.0001	0.797	0.591	0.437	0.336	0.258	0.227
12	342.46	12	<.0001	0.240	0.261	0.243	0.198	0.157	0.111
18	345.17	18	<.0001	0.071	0.054	0.042	0.037	0.036	0.008
24	353.39	24	<.0001	-0.040	-0.069	-0.117	-0.095	-0.074	-0.042



Conditional Least Squares Estimation					
Parameter	Estimate	① Standard Error	② t Value	Approx Pr > t	Lag
AR1,1	0.80575	0.04261	18.91	<.0001	1

Variance Estimate ③	430.7275
Std Error Estimate	20.75397
AIC	1781.668
SBC	1784.966
Number of Residuals	200

* AIC and SBC do not include log determinant.

Autocorrelation Check of Residuals										
To Lag	Chi-Square	DF	Pr > ChiSq	Autocorrelations						
6	5.46	5	0.3623	0.103	-0.051	-0.074	-0.020	-0.063	-0.060	
12	9.46	11	0.5791	0.014	0.110	0.074	0.007	0.034	-0.002	
18	11.30	17	0.8406	-0.048	-0.002	0.007	-0.001	0.065	0.042	
24	20.10	23	0.6359	-0.042	0.043	-0.185	-0.006	-0.032	-0.005	
30	24.49	29	0.7043	0.064	-0.007	0.033	0.028	-0.098	-0.056	
36	27.06	35	0.8290	0.029	0.029	0.074	0.002	-0.036	-0.046	

Model for variable Y	
Data have been centered by subtracting the value	90.09064

No mean term in this model.

Autoregressive Factors	
Factor 1:	1 - 0.80575 B**(1)

Forecasts for variable Y				
Obs	Forecast	Std Error	④	95% Confidence Limits
201	130.5036	20.7540	89.8265	171.1806
202	122.6533	26.6528	70.4149	174.8918
203	116.3280	29.8651	57.7936	174.8625
204	111.2314	31.7772	48.9492	173.5136
205	107.1248	32.9593	42.5257	171.7239

Obs	⑥ LAG	⑦ VAR	N	⑤ COV	⑦ CORR	STDERR	INVCORR	PARTCORR
1	0	Y	200	1198.54	1.00000	0.00000	1.00000	1.00000
2	1	Y	199	955.58	0.79729	0.07071	-0.57652	0.79729
3	2	Y	198	708.55	0.59118	0.10657	0.09622	-0.12213
4	3	Y	197	524.04	0.43723	0.12187	0.02752	0.01587
5	4	Y	196	402.37	0.33572	0.12947	-0.07210	0.03245
6	5	Y	195	308.94	0.25777	0.13376	0.04054	-0.00962
7	6	Y	194	271.51	0.22653	0.13622	0.02199	0.08239
8	7	Y	193	287.61	0.23997	0.13809	0.00313	0.10406
9	8	Y	192	313.13	0.26126	0.14016	-0.04048	0.04971
10	9	Y	191	291.72	0.24340	0.14257	-0.01001	-0.04771
11	10	Y	190	237.63	0.19827	0.14463	0.03317	-0.03747
12	11	Y	189	188.06	0.15691	0.14599	-0.02052	0.00538
13	12	Y	188	132.94	0.11092	0.14683	-0.01059	-0.03433
14	13	Y	187	85.07	0.07098	0.14725	0.03023	-0.00640
15	14	Y	186	64.18	0.05355	0.14742	-0.02472	0.02087
16	15	Y	185	50.93	0.04249	0.14751	-0.01468	-0.02806
17	16	Y	184	43.81	0.03656	0.14758	0.07028	-0.00672

Obs	⑥ LAG	⑥ VAR	N	⑥ COV	⑦ CORR	STDERR	INVCORR	PARTCORR
18	17	Y	183	43.52	0.03631	0.14762	-0.06341	0.01212
19	18	Y	182	10.11	0.00843	0.14767	-0.02383	-0.07674
20	19	Y	181	-47.37	-0.03952	0.14767	0.11277	-0.06557
21	20	Y	180	-82.87	-0.06914	0.14772	-0.18424	0.01500
22	21	Y	179	-140.53	-0.11725	0.14788	0.20996	-0.10473
23	22	Y	178	-113.55	-0.09474	0.14835	-0.12659	0.14816
24	23	Y	177	-88.68	-0.07399	0.14865	0.04992	-0.03625
25	24	Y	176	-50.80	-0.04239	0.14883	-0.01756	0.03510

The CENTER option tells PROC ARIMA to use the series mean, \bar{Y} , to estimate μ . The estimates of $\gamma(j)$ are called COV ⑤ with j labeled LAG ⑥ in the output. The covariances 1199, 956, 709, 524, 402, 309, ... decrease at a rate of about 0.8. Dividing each covariance by the variance 1198.54 (covariance at LAG 0) gives the estimated sequence of correlations ⑦. The correlation at lag 0 is always $\rho(0) = 1$, and in general, $\rho(j) = \gamma(j) / \gamma(0)$. The “Series Autocorrelations for Y” plot shows approximate exponential decay. The ESTIMATE statement produces an estimate ① $\hat{\rho} = 0.80575$ that you can test for significance with the t value. Because $t = 18.91$ ② exceeds the 5% critical value, $\hat{\rho}$ is significant. If ρ were 0, this t value would have approximately a standard normal distribution in large samples. Thus, a t exceeding 1.96 in magnitude would be considered significant at about the 5% level. Also, you have an estimate of σ^2 , 430.7275 ③.

You forecast Y_{201} by the following:

$$90.091 + .80575(140.246 - 90.091) = 130.503$$

You use the forecast standard error:

$$(430.7275)^{1/2} = 20.7540 \quad ④$$

Next, you forecast Y_{202} by the following:

$$90.091 + (.80575)^2(140.246 - 90.091) = 122.653$$

And you use forecast standard error as follows:

$$(430.7275(1 + 0.80575^2))^{1/2} = 26.6528 \quad ④$$

As previously illustrated, PROC ARIMA produced the forecasts and standard errors ④. The coefficients are estimated through the least squares (LS) method:

$$0.80575 = \sum(Y_t - \bar{Y})(Y_{t-1} - \bar{Y}) / \sum(Y_{t-1} - \bar{Y})^2$$

In this equation, \bar{Y} is the mean of the data set, and the sums run from 2 to 200. One alternative estimation scheme is the maximum likelihood (ML) method. Another alternative is unconditional least squares (ULS). A discussion of these methods in the context of the autoregressive order 1 AR(1) model follows. The likelihood function for a set of observations is simply their joint probability density viewed as a function of the parameters. The first observation Y_1 is normal with mean μ and variance $\sigma^2 / (1 - \rho^2)$. Its probability density function is as follows:

$$\frac{\sqrt{1-\rho^2}}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(Y_1 - \mu)^2(1-\rho^2)}{2\sigma^2}\right)$$

For the rest of the observations, $t = 2, 3, 4, \dots$, it is most convenient to note that $e_t = Y_t - \rho Y_{t-1}$ has a normal distribution with mean $\mu - \rho\mu = (1 - \rho)\mu$ and variance σ^2 .

Each of these probability densities is thus given by this expression:

$$\frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{[(Y_t - \mu) - \rho(Y_{t-1} - \mu)]^2}{2\sigma^2}\right)$$

Because $Y_1, e_2, e_3, \dots, e_n$ are independent, the joint likelihood is the product of these n probability density functions—namely, the following:

$$\frac{\sqrt{1-\rho^2}}{(\sqrt{2\pi\sigma^2})^n} \exp\left(-\frac{(1-\rho^2)(Y_1 - \mu)^2 + [(Y_2 - \mu) - \rho(Y_1 - \mu)]^2 + \dots + [(Y_n - \mu) - \rho(Y_{n-1} - \mu)]^2}{2\sigma^2}\right)$$

Substituting the observations for Y in this expression produces an expression involving μ , ρ , and σ^2 . Viewed in this way, the expression is called the likelihood function for the data. It clearly depends on assumptions about the model form. Using calculus, it can be shown that the estimate of σ^2 that maximizes the likelihood is USS/n , where USS represents the unconditional sum of squares:

$$\text{USS} = (1-\rho^2)(Y_1 - \mu)^2 + [(Y_2 - \mu) - \rho(Y_1 - \mu)]^2 + \dots + [(Y_n - \mu) - \rho(Y_{n-1} - \mu)]^2$$

The estimates that minimize USS are the unconditional least squares (ULS) estimates—that is, USS is the objective function to be minimized by the ULS method. The minimization can be modified in the current example by inserting \bar{Y} in place of μ , leaving only ρ to be estimated.

The conditional least squares (CLS) method results from assuming that Y_0 and all other Y s that occurred before you started observing the series are equal to the mean. Thus, it minimizes a slightly different objective function:

$$[Y_1 - \mu]^2 + [(Y_2 - \mu) - \rho(Y_1 - \mu)]^2 + \dots + [(Y_n - \mu) - \rho(Y_{n-1} - \mu)]^2$$

As with the other methods, it can be modified by substituting \bar{Y} for μ , leaving only ρ to be estimated. The first term cannot be changed by manipulating ρ , so the CLS method with \bar{Y} substituted also minimizes the following:

$$[(Y_2 - \bar{Y}) - \rho(Y_1 - \bar{Y})]^2 + \dots + [(Y_n - \bar{Y}) - \rho(Y_{n-1} - \bar{Y})]^2$$

In other words, the CLS estimate of ρ could be obtained by regressing deviations from the sample mean on their lags, with no intercept in this simple centered case.

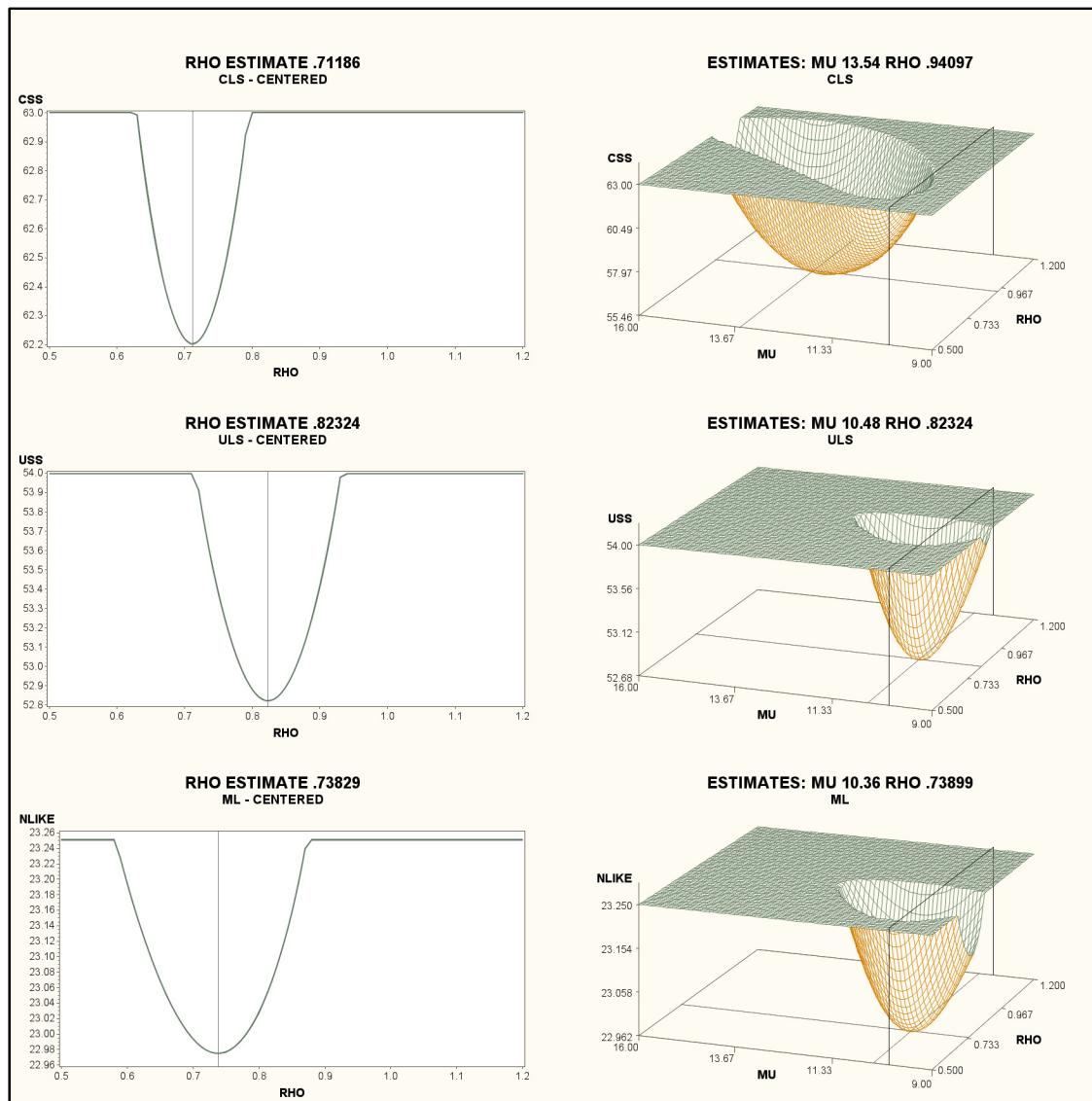
If full maximum likelihood estimation is desired, then the expression USS/n is substituted for σ^2 in the likelihood function. The resulting expression, called a *concentrated likelihood*, is maximized. The log of the likelihood is this:

$$-(n/2)\log(2\pi/n) - (n/2) - (n/2)\log(\text{USS}) + (1/2)\log(1-\rho^2)$$

The ML method can be run on centered data by substituting $Y_t - \bar{Y}$ in USS for $Y_t - \mu$.

For the series $\{14, 15, 14, 10, 12, 10, 5, 6, 6, 8\}$, the sample average is 10. The three rows in **Output 2.2** display the objective functions just discussed for conditional least squares, unconditional least squares, and maximum likelihood for an AR(1) model fit to these data. The negative of the likelihood is shown so that a minimum is sought in each case. The right panel in each row plots the function to be minimized over a floor of (ρ, μ) pairs, with each function truncated by a convenient ceiling plane. Crosshairs in the plot floors indicate the minimizing values, and it is seen that these estimates can vary somewhat from method to method when the sample size is very small. Each plot also shows a vertical slicing plane at $\mu = 10$, corresponding to the sample mean. The left-hand plots show the cross-section from the slicing planes. These are the objective functions to be minimized when the sample mean, 10, is used as an estimate of the population mean. The slicing plane does not meet the floor at the crosshair mark, so the sample mean differs somewhat from the estimate that minimizes the objective function. Likewise, the ρ that minimizes the cross section plot is not the same as the one minimizing the surface plot, although this difference is quite minor for ULS and ML in this small example.

Output 2.2: Objective Functions



The minimizing values for the right-side ULS plot are obtained from the following code METHOD=ML for maximum likelihood and there is no method specification for CLS:

```
proc arima data=estimate;
  identify var=y noprint;
  estimate p=1 method = uls outest=outuls printall;
run;
```

The OUTEST data set contains the estimates and related information. PRINTALL shows the iterative steps used to search for the minima. If you were to use the CENTER option in the IDENTIFY statement and the NOCONSTANT option in the ESTIMATE statement, the code would produce the ρ estimate that minimizes the objective function computed with the sample mean, 10. Partial output showing the iterations for the small series is shown in **Output 2.3**. The second column in each segment is the objective function that is being minimized. It should end with the height of the lowest point in each plot. The estimates correspond to the coordinates on the horizontal axis (or the floor) corresponding to the minimum. Up through the first ML METHOD output, the minimization routine searches estimates of both μ and ρ . In output after that, searches for ρ are performed after setting the μ estimate to the sample mean.

Output 2.3: Using PROC ARIMA to Get Iterations for Parameter Estimates**CLS METHOD****The ARIMA Procedure**

Conditional Least Squares Estimation						
Iteration	SSE	MU	AR1,1	Constant	Lambda	R Crit
0	62.2677	10.00000	0.68852	3.114754	0.00001	1
1	58.9831	11.23572	0.70534	3.310709	1E-6	0.216536
2	57.5765	11.48670	0.79565	2.347267	1E-7	0.1318
3	56.7205	12.11988	0.82108	2.168455	1E-8	0.0988
4	56.1715	12.42416	0.86430	1.686007	1E-9	0.074987
5	55.8188	12.81406	0.88581	1.463231	1E-10	0.060663
6	55.6237	13.04116	0.90725	1.209559	1E-11	0.045096
7	55.5279	13.23798	0.91913	1.070524	1E-12	0.03213
8	55.4864	13.35049	0.92816	0.959077	1E-12	0.021304
9	55.4698	13.43016	0.93320	0.897121	1E-12	0.013585
10	55.4635	13.47506	0.93656	0.854795	1E-12	0.008377
11	55.4612	13.50404	0.93846	0.831106	1E-12	0.005073
12	55.4604	13.52045	0.93964	0.816055	1E-12	0.003033
13	55.4601	13.53060	0.94032	0.807533	1E-12	0.001801
14	55.4600	13.53640	0.94073	0.802304	1E-12	0.001065
15	55.4600	13.53990	0.94097	0.79931	1E-12	0.000628

ULS METHOD**The ARIMA Procedure**

Conditional Least Squares Estimation						
Iteration	SSE	MU	AR1,1	Constant	Lambda	R Crit
0	62.2677	10.00000	0.68852	3.114754	0.00001	1
1	58.9831	11.23572	0.70534	3.310709	1E-6	0.216536
2	57.5765	11.48670	0.79565	2.347267	1E-7	0.1318
3	56.7205	12.11988	0.82108	2.168455	1E-8	0.0988
4	56.1715	12.42416	0.86430	1.686007	1E-9	0.074987

Unconditional Least Squares Estimation						
Iteration	SSE	MU	AR1,1	Constant	Lambda	R Crit
0	54.3164	12.42416	0.86430	1.686007	0.00001	1
1	52.9894	10.60067	0.87709	1.302965	1E-6	0.165164
2	52.7745	10.86905	0.83699	1.771771	1E-7	0.065079
3	52.7002	10.51067	0.83570	1.726955	1E-8	0.036672
4	52.6864	10.54790	0.82618	1.833486	1E-9	0.01526
5	52.6838	10.49047	0.82539	1.831752	1E-10	0.006528
6	52.6833	10.49285	0.82374	1.849434	1E-11	0.00267
7	52.6832	10.48409	0.82351	1.850351	1E-12	0.001104
8	52.6832	10.48384	0.82324	1.853158	1E-12	0.000458

ML METHOD

The ARIMA Procedure

Conditional Least Squares Estimation						
Iteration	SSE	MU	AR1,1	Constant	Lambda	R Crit
0	62.2677	10.00000	0.68852	3.114754	0.00001	1
1	58.9831	11.23572	0.70534	3.310709	1E-6	0.216536
2	57.5765	11.48670	0.79565	2.347267	1E-7	0.1318
3	56.7205	12.11988	0.82108	2.168455	1E-8	0.0988
4	56.1715	12.42416	0.86430	1.686007	1E-9	0.074987

Maximum Likelihood Estimation						
Iter	Loglike	MU	AR1,1	Constant	Lambda	R Crit
0	-23.33779	12.42416	0.86430	1.686007	0.00001	1
1	-22.97496	10.32117	0.76958	2.378179	1E-6	0.233964
2	-22.96465	10.50928	0.73277	2.808362	1E-7	0.058455
3	-22.96211	10.33518	0.74384	2.647467	1E-8	0.028078
4	-22.96176	10.38272	0.73739	2.726623	1E-9	0.010932
5	-22.96169	10.35478	0.73966	2.69579	1E-10	0.004795
6	-22.96168	10.36483	0.73855	2.709918	1E-11	0.002018
7	-22.96168	10.36003	0.73899	2.70409	1E-12	0.00087

Notice that each method begins with CLS starting with the sample mean and an estimate, 0.6885, of the autoregressive coefficient. The CLS estimates, after a few iterations, are substituted in the ULS or ML objective function when one of those methods is specified. In more complex models, the likelihood function is more involved, as are the other objective functions. Nevertheless, the basic ideas generalize for all models handled by PROC ARIMA.

You have no reason to believe that dependence of Y_t on past values should be limited to the previous observation Y_{t-1} . For example, you might have equation 2.4:

$$Y_t - \mu = \alpha_1(Y_{t-1} - \mu) + \alpha_2(Y_{t-2} - \mu) + e_t \quad (2.4)$$

This is a second-order autoregressive, AR(2), process. One way to determine whether you have this process is to examine the autocorrelation plot by using the following SAS statements:

```
proc arima data=estimate;
  identify var=y;
run;
```

You need to study the form of autocorrelations for such AR processes, which is facilitated by writing the models in backshift notation.

2.2.2 Backshift Notation B for Time Series

A convenient notation for time series is the backshift notation B , where the following defines the B operator:

$$B(Y_t) = Y_{t-1}$$

That is, B indicates a shifting back of the time subscript. Similarly,

$$B^2(Y_t) = B(Y_{t-1}) = Y_{t-2}$$

and

$$B^5(Y_t) = Y_{t-5}$$

Now, consider the process $Y_t = 0.8Y_{t-1} + e_t$. In backshift notation, this becomes $(1 - 0.8B)Y_t = e_t$. You can write

$$Y_t = (1 - 0.8B)^{-1}e_t$$

and, recalling that $(1 - X)^{-1} = 1 + X + X^2 + X^3 + \dots$, for $|X| < 1$, you obtain the following:

$$Y_t = (1 + 0.8B + 0.8^2B^2 + 0.8^3B^3 + \dots)e_t$$

or

$$Y_t = e_t + 0.8e_{t-1} + 0.64e_{t-2} + \dots$$

It becomes apparent that backshift notation enables you to execute the computations, linking equations 2.1 and 2.3, in a simplified manner. This technique extends to higher-order processes. Consider equation 2.5:

$$Y_t = 1.70Y_{t-1} - 0.72Y_{t-2} + e_t \quad (2.5)$$

Comparing equations 2.5 and 2.4 results in $\mu = 0$, $\alpha_1 = 1.70$, and $\alpha_2 = -0.72$. You can rewrite equation 2.5:

$$(1 - 1.70B + 0.72B^2)Y_t = e_t$$

Or, rewrite it as equation 2.6:

$$Y_t = (1 - 1.70B + 0.72B^2)^{-1}e_t \quad (2.6)$$

Algebraic combination shows the following:

$$9/(1 - 0.9B) - 8/(1 - 0.8B) = 1/(1 - 1.70B + 0.72B^2)$$

Thus, you can write Y_t as follows:

$$Y_t = \sum_{j=0}^{\infty} W_j e_{t-j}$$

where $W_j = 9(0.9^j) - 8(0.8^j)$.

You can see that the influence of early shocks e_{t-j} is minimal, because 0.9 and 0.8 are less than 1. Equation 2.7 enables you to write Y_t as follows:

$$\begin{aligned} Y_t = e_t + 1.7e_{t-1} + 2.17e_{t-2} + 2.47e_{t-3} + 2.63e_{t-4} \\ + 2.69e_{t-5} + 2.69e_{t-6} + 2.63e_{t-7} + 2.53e_{t-8} + \dots \end{aligned} \quad (2.7)$$

You could also accomplish this by repeated back substitution as in equation 2.3. Note that the weights W_j initially increase before tapering off toward 0.

2.2.3 Yule-Walker Equations for Covariances

You have learned how to use backshift notation to write a time series as a weighted sum of past shocks (as in equation 2.7). You are now ready to compute covariances $\gamma(j)$. You accomplish this by using Yule-Walker equations. These equations result from multiplying the time series equation (such as equation 2.5) by Y_{t-j} and computing expected values.

For equation 2.5, when you use $j = 0$, you obtain the following:

$$E(Y_t^2) = 1.70E(Y_t Y_{t-1}) - 0.72E(Y_t Y_{t-2}) + E(Y_t e_t)$$

or

$$\gamma(0) = 1.70\gamma(1) - 0.72\gamma(2) + \sigma^2$$

Here, E stands for expected value. Using equation 2.7 with all subscripts lagged by 1, you see that Y_{t-1} involves only e_{t-1} , e_{t-2} , Thus, $E(Y_{t-1} e_t) = 0$. When you use $j = 1$, you obtain the following:

$$E(Y_t Y_{t-1}) = 1.70E(Y_{t-1}^2) - 0.72(Y_{t-1} Y_{t-2}) + E(Y_{t-1} e_t)$$

Furthermore, $E(Y_{t-1} Y_{t-2}) = \gamma(1)$ because the difference in subscripts is $(t-1) - (t-2) = 1$. Also recall this:

$$\gamma(1) = \gamma(-1)$$

Using these ideas, write your second Yule-Walker equation as follows:

$$\gamma(1) = 1.70\gamma(0) - 0.72\gamma(1)$$

In the same manner, for all $j > 0$, you have the following:

$$\gamma(j) = 1.70\gamma(j-1) - 0.72\gamma(j-2) \quad (2.8)$$

If you assume a value for σ^2 (for example, $\sigma^2 = 10$), then you can use the Yule-Walker equations to compute autocovariances $\gamma(j)$ and autocorrelations:

$$\rho(j) = \gamma(j)/\gamma(0)$$

The autocorrelations do not depend on σ^2 . The Yule-Walker equations for $j = 0$, $j = 1$, and $j = 2$ are three equations in three unknowns: $\gamma(0)$, $\gamma(1)$, and $\gamma(2)$. Solving these (using $\sigma^2 = 10$), you get $\gamma(0) = 898.1$, $\gamma(1) = 887.6$, and $\gamma(2) = 862.4$. Using equation 2.8, you then compute as follows:

$$\gamma(3) = 1.7(862.4) - 0.72(887.6) = 827.0$$

and

$$\gamma(4) = 1.7(827.0) - 0.72(862.4) = 785$$

and so forth.

Thus, the Yule-Walker equations for an AR(2) process (see equation 2.4) are the following:

$$\gamma(0) = \alpha_1\gamma(1) + \alpha_2\gamma(2) + \sigma^2$$

and

$$\gamma(j) = \alpha_1\gamma(j-1) + \alpha_2\gamma(j-2), \quad j > 0$$

You have seen that PROC ARIMA gives estimates of $\gamma(j)$. With that in mind, suppose you have a time series with mean 100 and the following covariance sequence:

j	$\gamma(j)$
0	390.00
1	360.00
2	277.50
3	157.50
4	19.90
5	-113.80
6	-223.70
7	-294.50
8	-317.60
9	-292.20
10	-223.90
11	-125.50
12	-13.20
13	95.50
14	184.40

The last two observations are 130 and 132, and you want to predict five steps ahead. How do you do so? First, you need a model for the data. You can eliminate the AR(1) model because of the failure of $\gamma(j)$ to taper off at a constant exponential rate. For example, $\gamma(1)/\gamma(0) = 0.92$, but $\gamma(2)/\gamma(1) = 0.77$.

If the model is an AR(2) model as in equation 2.4, you have the following Yule-Walker equations:

$$\begin{aligned} 390 &= \alpha_1(360) + \alpha_2(277.5) + \sigma^2 \\ 360 &= \alpha_1(390) + \alpha_2(360) \end{aligned}$$

and

$$277.5 = \alpha_1(360) + \alpha_2(390)$$

These can be solved with $\alpha_1 = 1.80$, $\alpha_2 = -0.95$, and $\sigma^2 = 5.625$. Thus, in general, if you know or if you can estimate the $\gamma(j)$ s, then you can find or estimate the coefficients from the Yule-Walker equations. You can confirm this diagnosis by checking whether the following is true for $j = 3, 4, \dots, 14$:

$$\gamma(j) = 1.80\gamma(j-1) - 0.95\gamma(j-2)$$

To predict, you first write your equation:

$$Y_t - 100 = 1.80(Y_{t-1} - 100) - 0.95(Y_{t-2} - 100) + e_t \quad (2.9)$$

Assuming that your last observation is Y_n , you now write Y_{n+1} as follows:

$$Y_{n+1} = 100 + 1.80(Y_n - 100) - 0.95(Y_{n-1} - 100) + e_{n+1}$$

Thus, the forecast becomes this:

$$\hat{Y}_{n+1} = 100 + 1.80(132 - 100) - 0.95(130 - 100) = 129.1$$

Here you recall that 130 and 132 were the last two observations. The prediction error is

$$Y_{n+1} - \hat{Y}_{n+1} = e_{n+1}$$

with variance $\sigma^2 = 5.625$. You compute the one-step-ahead prediction interval from

$$129.1 - 1.96(5.625)^{1/2}$$

to

$$129.1 + 1.96(5.625)^{1/2}$$

The prediction of Y_{n+2} arises from the following expression:

$$Y_{n+2} = 100 + 1.80(Y_{n+1} - 100) - 0.95(Y_n - 100) + e_{n+2}$$

And it is given by the following:

$$\hat{Y}_{n+2} = 100 + 1.80(\hat{Y}_{n+1} - 100) - 0.95(Y_n - 100) = 122$$

The prediction error is $1.8(Y_{n+1} - \hat{Y}_{n+1}) + e_{n+2} = 1.8e_{n+1} + e_{n+2}$, with variance $\sigma^2(1 + 1.8^2) = 23.85$.

Using equation 2.9, you compute predictions, replacing unknown Y_{n+j} with predictions, and e_{n+j} with 0 for $j > 0$. You also can monitor prediction error variances. If you express Y_t in the form of equation 2.7, you get the following:

$$Y_t - 100 = e_t + 1.8e_{t-1} + 2.29e_{t-2} + 2.41e_{t-3} + \dots \quad (2.10)$$

The prediction error variances for one, two, three, and four steps ahead are then σ^2 , $\sigma^2(1 + 1.8^2)$, $\sigma^2(1 + 1.8^2 + 2.29^2)$, and $\sigma^2(1 + 1.8^2 + 2.29^2 + 2.41^2)$.

Surprisingly, the weights on e_{t-j} seem to increase as you move further into the past. However, if you continue to write out the expression for Y_t in terms of e_t , you see that the weights eventually taper off toward 0, just as in equation 2.7. You obtained equation 2.10 by writing the model

$$(1 - 1.80B + 0.95B^2)(Y_t - \mu) = e_t$$

as follows:

$$\begin{aligned} (Y_t - \mu) &= (1 - 1.80B + 0.95B^2)^{-1} e_t \\ &= (1 + 1.80B + 2.29B^2 + 2.41B^3 + \dots) e_t \end{aligned}$$

Now replace B with an algebraic variable M . The key to tapering off the weights involves this characteristic equation:

$$1 - 1.80M + 0.95M^2 = 0$$

If all values of M (roots) that solve this equation are larger than 1 in magnitude, the weights taper off. In this case, the roots are $M = 0.95 \pm 0.39i$, which is a complex pair of numbers with magnitude 1.03. In equation 2.5, the roots are 1.11 and 1.25. The condition of roots having a magnitude greater than 1 is called *stationarity* and ensures that shocks e_{t-j} in the distant past have little influence on the current observation Y_t .

The AR model of order p is written as follows:

$$(Y_t - \mu) = \alpha_1(Y_{t-1} - \mu) + \alpha_2(Y_{t-2} - \mu) + \dots + \alpha_p(Y_{t-p} - \mu) + e_t$$

A general review of this discussion indicates that this model can be written in backshift form as follows:

$$(1 - \alpha_1 B - \alpha_2 B^2 - \dots - \alpha_p B^p)(Y_t - \mu) = e_t$$

Then, rewrite it as an infinite weighted sum of current and past shocks e_t as:

$$\begin{aligned}(Y_t - \mu) &= (1 - \alpha_1 B - \alpha_2 B^2 - \cdots - \alpha_p B^p)^{-1} e_t \\ &= (1 + W_1 B + W_2 B^2 + W_3 B^3 + \cdots) e_t\end{aligned}$$

You can now find the W_j s. The W_j s taper off toward 0 if all M s satisfying the following are such that $|M| > 1$:

$$1 - \alpha_1 M - \alpha_2 M^2 - \cdots - \alpha_p M^p = 0$$

You have also learned how to compute the system of Yule-Walker equations by multiplying equation 2.11 on both sides by $(Y_{t-j} - \mu)$ for $j = 0, j = 1, j = 2, \dots$, and by computing expected values. You can use these Yule-Walker equations to estimate coefficients α_j when you know or you can estimate values of the covariances $\gamma(j)$. You have also used covariance patterns to distinguish the AR(2) model from the AR(1) model.

2.3 Fitting an AR Model in PROC REG

Chapter 3, “The General ARIMA Model,” shows that associating autocovariance patterns with models is crucial for determining an appropriate model for a data set. As you expand your set of models, remember that the primary way to distinguish among them is through their covariance functions. Thus, it is crucial to build a catalog of their covariance functions as you expand your repertoire of models. The covariance functions are like fingerprints, helping you identify the model form appropriate for your data.

Output 2.4 shows a plot of the stocks of silver at the New York Mercantile Exchange in 1000 troy ounces from December 1976 through May 1981 (Fairchild Publications 1981). If you deal only with AR processes, you can fit the models by ordinary regression techniques such as PROC REG or PROC GLM. You can simplify the choice of the model’s order and thus your analysis, as illustrated in **Output 2.5**.

Assuming an order 4 model is adequate, you regress Y_t on Y_{t-1} , Y_{t-2} , Y_{t-3} , and Y_{t-4} using these SAS statements:

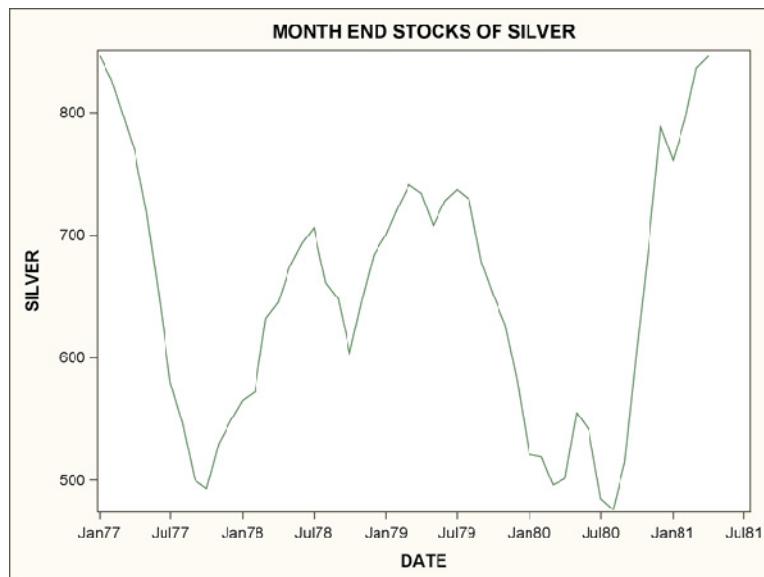
```
data silver;
  title 'MONTH END STOCKS OF SILVER';
  input silver @@;
  t=_n_;
  retain date '01DEC76'd lsilver1-lsilver4;
  date=intnx('MONTH',date,1);
  format date monyy.;
  output;
  lsilver4=lsilver3;
  lsilver3=lsilver2;
  lsilver2=lsilver1;
  lsilver1=silver;
  datalines;
846 827 799 768 719 652 580 546 500 493 530 548 565
572 632 645 674 693 706 661 648 604 647 684 700 723
741 734 708 728 737 729 678 651 627 582 521 519 496
501 555 541 485 476 515 606 694 788 761 794 836 846
;
run;

proc sgplot data=silver;
  series x=date y=silver;
  xaxis valuesformat=monyy5.;
quit;

proc print data=silver;
run;

proc reg data=silver;
  model silver=lsilver1 lsilver2 lsilver3 lsilver4 / ssl;
run;

proc reg data=silver;
  model silver=lsilver1 lsilver2;
run;
```

Output 2.4: Plotting Monthly Stock Values**Output 2.5: Using PROC PRINT to List the Data and Using PROC REG to Fit an AR Process****MONTH END STOCKS OF SILVER**

Obs	SILVER	T	DATE	LSILVER1	LSILVER2	LSILVER3	LSILVER4
1	846	1	JAN77
2	827	2	FEB77	846	.	.	.
3	799	3	MAR77	827	846	.	.
4	768	4	APR77	799	827	846	.
5	719	5	MAY77	768	799	827	846

(Additional output omitted)

48	788	48	DEC80	694	606	515	476
49	761	49	JAN81	788	694	606	515
50	794	50	FEB81	761	788	694	606
51	836	51	MAR81	794	761	788	694
52	846	52	APR81	836	794	761	788

The REG Procedure
Model: MODEL1
Dependent Variable: SILVER

Number of Observations Read	52
Number of Observations Used	48
Number of Observations with Missing Values	4

Analysis of Variance					
Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	4	417429	104357	95.30	<.0001
Error	43	47085	1095.00765		
Corrected Total	47	464514			

Root MSE	33.09090	R-Square	0.8986
Dependent Mean	636.89583	Adj R-Sq	0.8892
Coeff Var	5.19565		

Parameter Estimates						
Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > t	Type I SS
Intercept	1	102.84126	37.85904	2.72	0.0095	19470543
LSILVER1	1	1.38589	0.15156	9.14	<.0001	387295
LSILVER2	1	-0.44231	0.26078	-1.70	0.0971	28472
LSILVER3	1	0.00921	0.26137	0.04	0.9720	1061.93530
LSILVER4	1	-0.11236	0.15185	-0.74	0.4633	599.56290

The REG Procedure

Model: MODEL1

Dependent Variable: SILVER

Number of Observations Read	52
Number of Observations Used	50
Number of Observations with Missing Values	2

Analysis of Variance					
Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	2	457454	228727	220.26	<.0001
Error	47	48808	1038.45850		
Corrected Total	49	506261			

Root MSE	32.22512	R-Square	0.9036
Dependent Mean	642.76000	Adj R-Sq	0.8995
Coeff Var	5.01355		

Parameter Estimates					
Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > t
Intercept	1	77.95372	30.21038	2.58	0.0131
LSILVER1	1	1.49087	0.11589	12.86	<.0001
LSILVER2	1	-0.61144	0.11543	-5.30	<.0001

Output 2.5 shows that lags 3 and 4 might not be needed because the overall F value for these two lags is computed ❶❷ as follows:

$$\left[(1062 + 600) / 2 \right] / 1095 = 0.76$$

This is insignificant compared to the F distribution with 2 and 43 degrees of freedom. Alternatively, a TEST statement could be used to produce F .

You have identified the model through overfitting. Now, the final estimated model ❸ is this one:

$$Y_t = 77.9537 + 1.4909Y_{t-1} - 0.6114Y_{t-2} + e_t$$

It becomes this:

$$Y_t - 647 = 1.4909(Y_{t-1} - 647) - 0.6114(Y_{t-2} - 647) + e_t$$

All parameters are significant according to their t values ❹.

The fact that $M = 1$ almost solves the following characteristic equation suggests that this series might be nonstationary:

$$1 - 1.49M + 0.61M^2$$

In Chapter 3, you extend your class of models to include moving averages and mixed ARMA models. These models require more sophisticated fitting and identification techniques than did the simple regression with overfitting that was used in the previous silver example.

Chapter 3: The General ARIMA Model

3.1 Introduction.....	41
3.1.1 Statistical Background	41
3.1.2 Terminology and Notation.....	41
3.2 Prediction.....	42
3.2.1 One-Step-Ahead Predictions	42
3.2.2 Future Predictions	43
3.3 Model Identification.....	46
3.3.1 Stationarity and Invertibility	46
3.3.2 Time Series Identification.....	47
3.3.3 Chi-Square Check of Residuals.....	56
3.3.4 Summary of Model Identification	56
3.4 Examples and Instructions	56
3.4.1 IDENTIFY Statement for Series 1-8	57
3.4.2 Example: Iron and Steel Export Analysis	65
3.4.3 Estimation Methods Used in PROC ARIMA.....	70
3.4.4 ESTIMATE Statement for Series 8-A	72
3.4.5 Nonstationary Series	77
3.4.6 Effect of Differencing on Forecasts	78
3.4.7 Examples: Forecasting IBM Series and Silver Series	80
3.4.8 Models for Nonstationary Data.....	84
3.4.9 Differencing to Remove a Linear Trend	91
3.4.10 Other Identification Techniques	95
3.5 Summary of Steps for Analyzing Nonseasonal Univariate Series	104

3.1 Introduction

In this chapter, some general tools for analysis are presented based on statistical principles, and examples of their use are provided. The simple AR(1) model is extended to include a larger class—the autoregressive moving average or ARMA(p,q) class of stationary time series models.

3.1.1 Statistical Background

The general class of autoregressive moving average (ARMA) models is developed in this chapter. As each new model is introduced, its autocovariance function $\gamma(j)$ is given. This helps you use the estimated autocovariances $C(j)$ that PROC ARIMA produces to select an appropriate model for the data. Using estimated autocovariances to determine a model to be fit is called *model identification*. Once you select the model, you can use PROC ARIMA to fit the model, forecast future values, and provide forecast intervals.

3.1.2 Terminology and Notation

The moving average of order 1 is given by equation 3.1:

$$Y_t = \mu + e_t - \beta e_{t-1} \quad (3.1)$$

where e_t is a white noise (uncorrelated) sequence with mean 0 and variance σ^2 . Clearly, $\text{var}(Y_t) = \gamma(0) = \sigma^2(1 + \beta^2)$ and $\text{cov}(Y_t, Y_{t-1}) = \gamma(1) = E((e_t - \beta e_{t-1})(e_{t-1} - \beta e_{t-2})) = -\beta\sigma^2$ with $\text{cov}(Y_t, Y_{t-j}) = 0$ for $j > 1$.

If you observe the autocovariance sequence $\gamma(0) = 100$, $\gamma(1) = 40$, $\gamma(2) = 0$, $\gamma(3) = 0$, ..., then you see that you are dealing with a moving average (MA) process of order 1 because $\gamma(j)=0$ for $j > 1$. Also, you know that $-\beta\sigma^2 = 40$ and $(1 + \beta^2)\sigma^2 = 100$, so $\beta = -0.5$ and $\sigma^2 = 80$. The model is $Y_t = \mu + e_t + 0.5e_{t-1}$.

If each autocovariance $\gamma(j)$ is divided by $\gamma(0)$, then the resulting sequence of autocorrelations is $\rho(j)$. For a moving average like equation 3.1, $\rho(0) = 1$, $\rho(1) = -\beta / (1 + \beta^2)$, and $\rho(j) = 0$ for $j > 1$. Note that $-1/2 \leq -\beta / (1 + \beta^2) \leq 1/2$, regardless of the value β . In the example, the autocorrelations for lags 0 through 4 are 1, 0.4, 0, 0, and 0.

The general moving average of order q is written as follows:

$$Y_t = \mu + e_t - \beta_1 e_{t-1} - \cdots - \beta_q e_{t-q}$$

It is characterized by the fact that $\gamma(j)$ and $\rho(j)$ are 0 for $j > q$. In backshift notation, you write as follows:

$$Y_t = \mu + (1 - \beta_1 B - \beta_2 B^2 - \cdots - \beta_q B^q) e_t$$

Similarly, you write the mixed autoregressive moving average model ARMA(p,q) like so:

$$(Y_t - \mu) - \alpha_1 (Y_{t-1} - \mu) - \cdots - \alpha_p (Y_{t-p} - \mu) = e_t - \beta_1 e_{t-1} - \cdots - \beta_q e_{t-q}$$

Or, in backshift notation, it is as follows:

$$(1 - \alpha_1 B - \cdots - \alpha_p B^p)(Y_t - \mu) = (1 - \beta_1 B - \cdots - \beta_q B^q)e_t$$

For example, the following model is an ARMA(1,1) with mean $\mu = 0$:

$$(1 - 0.6B)Y_t = (1 + 0.4B)e_t$$

In practice, parameters are estimated and then used to estimate prediction error variances for several periods ahead. PROC ARIMA provides these computations.

3.2 Prediction

One of the most common reasons for fitting time series models is to produce forecasts. This section addresses forecasting with ARMA models.

3.2.1 One-Step-Ahead Predictions

You can further clarify the previous ARMA(1,1) example by predicting sequentially one step at a time. Let n denote the number of available observations. The next ($n + 1$) observation in the sequence satisfies the following:

$$Y_{n+1} = 0.6Y_n + e_{n+1} + 0.4e_n$$

First, predict Y_{n+1} by $\hat{Y}_{n+1} = 0.6Y_n + 0.4e_n$ with error variance σ^2 . Next, consider the following:

$$\begin{aligned} Y_{n+2} &= 0.6Y_{n+1} + e_{n+2} + 0.4e_{n+1} \\ &= 0.6(0.6Y_n + e_{n+1} + 0.4e_n) + e_{n+2} + 0.4e_{n+1} \end{aligned}$$

Therefore, predict Y_{n+2} by removing future e s (subscripts greater than n):

$$0.36Y_n + 0.24e_n = 0.6\hat{Y}_{n+1}$$

The prediction error is $e_{n+1} + e_{n+2}$, which has variance $2\sigma^2$. Finally, consider the following:

$$Y_{n+3} = 0.6Y_{n+2} + e_{n+3} + 0.4e_{n+2}$$

and

$$\hat{Y}_{n+3} = 0.6\hat{Y}_{n+2} + 0$$

Therefore, the prediction error is as follows:

$$\begin{aligned} Y_{n+3} - \hat{Y}_{n+3} &= 0.6(Y_{n+2} - \hat{Y}_{n+2}) + e_{n+3} + 0.4e_{n+2} \\ &= 0.6(e_{n+1} + e_{n+2}) + e_{n+3} + 0.4e_{n+2} \end{aligned}$$

The prediction error variance is $2.36\sigma^2$. This example shows that you can readily compute predictions and associated error variances after model parameters or their estimates are available.

The predictions for the model $Y_t = 0.6Y_{t-1} + e_t + 0.4e_{t-1}$ can be computed recursively as follows:

Observation	Prediction	Residual
10	(0)	10
5	10	-5
-3	1	-4
-8	-3.4	-4.6
1	-6.64	7.64
6	3.656	2.344
—	4.538	—
—	2.723	—
—	1.634	—

Start by assuming the mean (0) as a prediction of Y_1 with implied error $e_1 = 10$. Predict Y_2 by $0.6Y_1 + 0.4e_1 = 10$, using the assumed $e_1 = 10$. The residual is $r_2 = 5 - 10 = -5$. Using r_2 as an estimate of e_2 , predict Y_3 by the following:

$$0.6Y_2 + 0.4r_2 = 0.6(5) + 0.4(-5) = 1$$

The residual is $r_3 = Y_3 - 1 = -4$. Then, predict Y_4 by $0.6Y_3 + 0.4r_3 = 0.6(-3) + 0.4(-4) = -3.4$; Y_5 by -6.64 ; and Y_6 by 3.656.

These are one-step-ahead predictions for the historic data. For example, you use only the data up through $t = 3$ (and the assumed $e_1 = 10$) to predict Y_4 . The sum of squares of these residuals, $100 + 25 + \dots + 2.344^2 = 226.024$, is called the *conditional sum of squares* associated with the parameters 0.6 and 0.4. If you search in autoregressive (AR) and moving average (MA) parameters to find those that minimize this conditional sum of squares, you are performing conditional least squares (CLS) estimation, the default in PROC ARIMA. An estimate of the white noise variance is given by dividing the conditional sum of squares by n minus the number of estimated parameters; that is, $n - 2 = 6 - 2 = 4$ for this ARMA(1,1) with mean 0.

3.2.2 Future Predictions

Predictions into the future are of real interest. One-step-ahead computations are used to start the process. Continuing the previous process, estimate e_t s as 0 for t beyond n ($n = 6$ observations in the example). That is, estimate future Y s by their predictions.

The next three predictions are as follows:

$$\hat{Y}_7 \text{ is } 0.6(6) + 0.4(2.344) = 4.538 \text{ with error } e_7$$

$$\hat{Y}_8 \text{ is } 0.6(4.538) + 0.4(0) = 2.723 \text{ with error } e_8 + e_7$$

$$\hat{Y}_9 \text{ is } 0.6(2.723) + 0.4(0) = 1.634 \text{ with error } e_9 + e_8 + 0.6e_7$$

PROC ARIMA provides these computations; this illustration simply shows what PROC ARIMA is computing.

The prediction of Y_{7+j} is just this:

$$(0.6)^j \hat{Y}_7$$

It thus declines exponentially to the series mean (0 in the example). The prediction error variance increases from $\text{var}(e_t)$ to $\text{var}(Y_t)$. In a practical application, the form

$$Y_t - \alpha Y_{t-1} = e_t - \beta e_{t-1}$$

and parameter values

$$Y_t - 0.6Y_{t-1} = e_t + 0.4e_{t-1}$$

are not known. They can be determined through PROC ARIMA.

In practice, estimated parameters are used to compute predictions and standard errors. This procedure requires sample sizes much larger than those in the previous example.

Although they would not have to be, the forecasting methods used in PROC ARIMA are tied to the method of estimation. If you use CLS, the forecast is based on the expression of $Y_t - \mu$ as an infinite autoregression. For example, suppose $Y_t = \mu + e_t - \beta e_{t-1}$, a simple MA(1). Note that $e_t = Y_t - \mu + \beta e_{t-1}$, so at time $t-1$, you have $e_{t-1} = Y_{t-1} - \mu + \beta e_{t-2}$. Substituting this second expression into the first, you have $e_t = (Y_t - \mu) + \beta(Y_{t-1} - \mu) + \beta^2 e_{t-2}$. Continuing in this way and assuming that $|\beta| < 1$ so that $\beta^j e_{t-j}$ converges to 0 as j gets large, you find

$$e_t = \sum_{j=0}^{\infty} \beta^j (Y_{t-j} - \mu)$$

This can alternatively be expressed as follows:

$$(Y_t - \mu) = -\sum_{j=1}^{\infty} \beta^j (Y_{t-j} - \mu) + e_t.$$

Thus, the forecast of Y_t given data up to time $t-1$ is

$$\hat{Y}_t = \mu - \sum_{j=1}^{\infty} \beta^j (Y_{t-j} - \mu).$$

The expression

$$\sum_{j=1}^{\infty} \beta^j (Y_{t-j} - \mu)$$

depends on Y values prior to time 1, the infinite past. PROC ARIMA assumes Y values before time 1 are just equal to the mean μ and, of course, the parameters are replaced by estimates. The truncated sum

$$\sum_{j=1}^{t-1} \beta^j (Y_{t-j} - \mu)$$

is not necessarily the best linear combination of lagged Y values for predicting $Y_t - \mu$.

When maximum likelihood (ML) or unconditional least squares (ULS) estimation is used, optimal linear forecasts based on the finite past are computed. Suppose you want to minimize $E\{[(Y_t - \mu) - \phi_1(Y_{t-1} - \mu) - \phi_2(Y_{t-2} - \mu)]^2\}$ by finding ϕ_1 and ϕ_2 . That is, you want the minimum variance forecast of Y_t based on its two predecessors. Here, ϕ_1 and ϕ_2 are just coefficients. They do not necessarily represent AR or MA parameters.

Using calculus, you find the following:

$$\begin{pmatrix} \phi_1 \\ \phi_2 \end{pmatrix} = \begin{pmatrix} \gamma(0) & \gamma(1) \\ \gamma(1) & \gamma(0) \end{pmatrix}^{-1} \begin{pmatrix} \gamma(1) \\ \gamma(2) \end{pmatrix}$$

This gives the best forecast of Y_3 based on a linear combination of Y_1 and Y_2 :

$$\hat{Y}_3 = \mu + \phi_1(Y_2 - \mu) + \phi_2(Y_1 - \mu)$$

Likewise, to forecast Y_5 by using Y_1, Y_2, Y_3, Y_4 , the four ϕ_j s are computed as follows:

$$\begin{pmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \\ \phi_4 \end{pmatrix} = \begin{pmatrix} \gamma(0) & \gamma(1) & \gamma(2) & \gamma(3) \\ \gamma(1) & \gamma(0) & \gamma(1) & \gamma(2) \\ \gamma(2) & \gamma(1) & \gamma(0) & \gamma(1) \\ \gamma(3) & \gamma(2) & \gamma(1) & \gamma(0) \end{pmatrix}^{-1} \begin{pmatrix} \gamma(1) \\ \gamma(2) \\ \gamma(3) \\ \gamma(4) \end{pmatrix}$$

Here, $\gamma(h)$ is the autocovariance at lag h . The equations for the ϕ_j s can be set up for any ARMA structure and any number of lags. For the MA(1) example with parameter β , to predict the fifth Y , you have this:

$$\begin{pmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \\ \phi_4 \end{pmatrix} = \begin{pmatrix} 1+\beta^2 & -\beta & 0 & 0 \\ -\beta & 1+\beta^2 & -\beta & 0 \\ 0 & -\beta & 1+\beta^2 & -\beta \\ 0 & 0 & -\beta & 1+\beta^2 \end{pmatrix}^{-1} \begin{pmatrix} -\beta \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

For reasonably long time series whose parameters are well inside the stationarity and invertibility regions, the best linear combination forecast used when ML or ULS is specified does not differ by much from the truncated sum used when CLS is specified. (See [section 3.3.1](#).)

For an MA(1) process with lag 1 parameter $\beta = 0.8$, the weights on past Y , used in forecasting one step ahead, are listed below. The top row shows the first 14 weights assuming infinite past $(-(0.8)^t)$. The next two rows show finite past weights for $n = 7$ and $n = 14$ past observations.

Lag	Y_{t-1}	Y_{t-2}	Y_{t-3}	Y_{t-4}	Y_{t-5}	Y_{t-6}	Y_{t-7}	...	Y_{t-13}	Y_{t-14}
Infinite past	-0.80	-0.64	-0.51	-0.41	-0.33	-0.26	-0.21		-0.05	-0.04
$n = 7$, finite past	-0.79	-0.61	-0.47	-0.35	-0.25	-0.16	-0.08	—	—	—
$n = 14$, finite past	-0.80	-0.64	-0.51	-0.41	-0.32	-0.26	-0.20		-0.03	-0.02

Despite the fairly large β and small n values, the weights are quite similar. Increasing n to 25 produces weights indistinguishable, out to 2 decimal places, from those for the infinite past.

If $\beta = 1$, the series $Y_t = e_t - 1e_{t-1}$ is said to be noninvertible, indicating that you cannot get a nice, convergent series representation of e_t as a function of current and lagged Y values. Not only does this negate the previous discussion, but because a reasonable estimate of e_t cannot be extracted from the data, it eliminates any sensible model-based forecasting. The moving average of order q , $Y_t = e_t - \beta_1 e_{t-1} - \cdots - \beta_q e_{t-q}$ has an associated polynomial equation in the algebraic variable M , $1 - \beta_1 M - \cdots - \beta_q M^q = 0$ whose roots must satisfy $|M| > 1$ in order for the series to be invertible. Note the analogy with the characteristic equation computed from the autoregressive coefficients.

Fortunately, in practice, it is rare to encounter a naturally measured series that appears to be noninvertible. However, when differences are taken, noninvertibility can be artificially induced. For example, the time series $Y_t = \alpha_0 + \alpha_1 t + e_t$ is a

simple linear trend plus white noise. Some practitioners have the false impression that any sort of trend in a time series should be removed by taking differences. If that is done, then you see that the following results:

$$Y_t - Y_{t-1} = (\alpha_0 + \alpha_1 t + e_t) - (\alpha_0 + \alpha_1 (t-1) + e_{t-1}) = \alpha_1 + e_t - e_{t-1}$$

In the process of reducing the trend $\alpha_0 + \alpha_1 t$ to a constant α_1 , a noninvertible moving average has been produced. The parameters of $Y_t = \alpha_0 + \alpha_1 t + e_t$ are best estimated by the OLS regression of Y_t on t , this being a fundamental result of basic statistical theory. Perhaps, the practitioner was confused by thinking in a very narrow time series way.

3.3 Model Identification

In order to use the ARIMA procedure, you must first decide which model from the ARMA(p,q) class is to be fit. The process of making this decision is called *model identification*.

3.3.1 Stationarity and Invertibility

Consider the following ARMA model:

$$(1 - \alpha_1 B - \alpha_2 B^2 - \cdots - \alpha_p B^p)(Y_t - \mu) = (1 - \beta_1 B - \beta_2 B^2 - \cdots - \beta_q B^q)e_t$$

The model is stationary if all values of M such that

$$1 - \alpha_1 M - \alpha_2 M^2 - \cdots - \alpha_p M^p = 0$$

are larger than 1 in absolute value. Stationarity ensures that early values of e have little influence on the current value of Y . It also ensures that setting a few values of e to 0 at the beginning of a series does not affect the predictions very much, provided the series is moderately long. In the ARMA(1,1) example, the prediction of Y_6 with 0 as an estimate of e_1 differs from the prediction using the true e_1 by the quantity 0.01 e_1 . Any MA process is stationary. One AR example is $(1 - 1.3B + 0.3B^2)Y_t = e_t$, which is not stationary (the roots of $1 - 1.3M + 0.3M^2 = 0$ are $M = 1$ and $M = 10/3$). Another example is $(1 - 1.3B + 0.42B^2)Y_t = e_t$, which is stationary (the roots of $1 - 1.3M + 0.42M^2 = 0$ are $M = 10/7$ and $M = 10/6$).

A series satisfies the invertibility condition if all M s for which $1 - \beta_1 M - \beta_2 M^2 - \cdots - \beta_q M^q = 0$ are such that $|M| > 1$.

The invertibility condition ensures that Y_t can be expressed in terms of e_t and an infinite weighted sum of previous Y s. In the ARMA(1,1) example,

$$e_t = (1 + 0.4B)^{-1} (1 - 0.6B) Y_t$$

and

$$e_t = Y_t - Y_{t-1} + 0.4Y_{t-2} - 0.16Y_{t-3} + 0.064Y_{t-4} - \cdots$$

so

$$Y_t = e_t + Y_{t-1} - 0.4Y_{t-2} + 0.16Y_{t-3} - 0.064Y_{t-4} + \cdots$$

The decreasing weights on lagged values of Y enable you to estimate e_t from recent values of Y . Note that in **section 3.2.1**, the forecast of Y_{n+1} was $0.6Y_n + 0.4e_n$, so the ability to estimate e_n from the data was crucial.

3.3.2 Time Series Identification

You need to identify the form of the model. You can do this in PROC ARIMA by inspecting data-derived estimates of three functions:

- autocorrelation function (ACF)
- inverse autocorrelation function (IACF)
- partial autocorrelation function (PACF)

These functions are defined below. A short catalog of examples is developed. Properties useful for associating different forms of these functions with the corresponding time series forms are summarized.

In PROC ARIMA, an IDENTIFY statement produces estimates of all these functions. For example, the following SAS statements produce lists and plots of all three of these functions for the variable Y in the data set SERIES:

```
proc arima data=series;
  identify var=y;
run;
```

Autocovariance Function $\gamma(j)$

Recall that $\gamma(j)$ is the covariance between Y_t and Y_{t-j} , which is assumed to be the same for every t (stationarity). The autocovariance functions for Series 1–8 in the following table are discussed next (in these examples, e_t is white noise with variance $\sigma^2 = 1$):

Series	Model
1	$Y_t = 0.8Y_{t-1} + e_t$, AR(1), $\gamma(1) > 0$
2	$Y_t = -0.8Y_{t-1} + e_t$, AR(1), $\gamma(1) < 0$
3	$Y_t = 0.3Y_{t-1} + 0.4Y_{t-2} + e_t$, AR(2)
4	$Y_t = 0.7Y_{t-1} + 0.49Y_{t-2} + e_t$, AR(2)
5	$Y_t = e_t + 0.8e_{t-1}$, MA(1)
6	$Y_t = e_t - 0.3e_{t-1} + 0.4e_{t-2}$, MA(2)
7	$Y_t = e_t$, (white noise)
8	$Y_t = 0.6Y_{t-1} + e_t + 0.4e_{t-1}$, ARMA(1,1)

For an AR(1) series $Y_t - \rho Y_{t-1} = e_t$ (such as Series 1 and 2), the covariance sequence is $\gamma(j) = \rho^{|j|} \sigma^2 / (1 - \rho^2)$. For an AR(2) series $Y_t - \alpha_1 Y_{t-1} - \alpha_2 Y_{t-2} = e_t$ (such as Series 3 and 4), the covariance sequence begins with values $\gamma(0)$ and $\gamma(1)$, followed by $\gamma(j)$, that satisfy the following:

$$\gamma(j) - \alpha_1 \gamma(j-1) - \alpha_2 \gamma(j-2) = 0$$

The covariances might oscillate with a period, depending on α_1 and α_2 (such as Series 4). Beginning values are determined from the Yule-Walker equations.

For a general AR(p) series

$$Y_t - \alpha_1 Y_{t-1} - \alpha_2 Y_{t-2} - \cdots - \alpha_p Y_{t-p} = e_t$$

beginning values are $\gamma(0), \dots, \gamma(p-1)$, from which $\gamma(j)$ satisfies the following for $j > p$:

$$\gamma(j) - \alpha_1 \gamma(j-1) - \alpha_2 \gamma(j-2) - \cdots - \alpha_p \gamma(j-p) = 0$$

The fact that $\gamma(j)$ satisfies the same difference equation as the series ensures that $|\gamma(j)| < H\lambda^j$, where $0 < \lambda < 1$ and H is some finite constant. In other words, $\gamma(j)$ might oscillate, but it is bounded by a function that decreases exponentially to zero.

For an MA(1),

$$Y_t - \mu = e_t - \beta e_{t-1}$$

$$\gamma(0) = (1 + \beta^2) \sigma^2$$

$$\gamma(1) = \gamma(-1) = -\beta \sigma^2$$

and $\gamma(j) = 0$ for $|j| > 1$. For a general MA(q)

$$Y_t - \mu = e_t - \beta_1 e_{t-1} - \beta_2 e_{t-2} - \cdots - \beta_q e_{t-q}$$

the $q+1$ beginning values are $\gamma(0), \gamma(1), \dots, \gamma(q)$. Then $\gamma(j) = 0$ for $|j| > q$.

For an ARMA(1,1) process, you have the following:

$$(Y_t - \mu) - \alpha(Y_{t-1} - \mu) = e_t - \beta e_{t-1}$$

There is a drop-off from $\gamma(0)$ to $\gamma(1)$ determined by α and β . For $j > 1$, the pattern $\gamma(j) = \alpha \gamma(j-1)$ occurs. Thus, an apparently arbitrary drop followed by exponential decay characterizes the ARMA(1,1) covariance function.

Consider the ARMA(p,q) process:

$$(Y_t - \mu) - \alpha_1(Y_{t-1} - \mu) - \cdots - \alpha_p(Y_{t-p} - \mu) = e_t - \beta_1 e_{t-1} - \cdots - \beta_q e_{t-q}$$

There are q arbitrary beginning values that are affected by the moving average terms, followed by behavior characteristic of an AR(p) in the sense that the autocorrelations are of the form $\gamma(j) = \alpha_1 \gamma(j-1) + \alpha_2 \gamma(j-2) + \dots + \alpha_p \gamma(j-p)$, just as they are in the AR(p) case. For example a correlation sequence of the form 1, 0.4, 0.2, 0.1, 0.05, 0.025, ... suggests an ARMA(1,1) with $\alpha = 0.5$.

For a white noise sequence, $\gamma(j) = 0$ if $j \neq 0$.

Autocorrelation Function

The pattern, rather than the magnitude, of the sequence $\gamma(j)$ is associated with the model form. Normalize the autocovariance sequence $\gamma(j)$ by computing autocorrelations $\rho(j) = \gamma(j) / \gamma(0)$. Notably, $\rho(0) = 1$ for all series and $\rho(j) = \rho(-j)$.

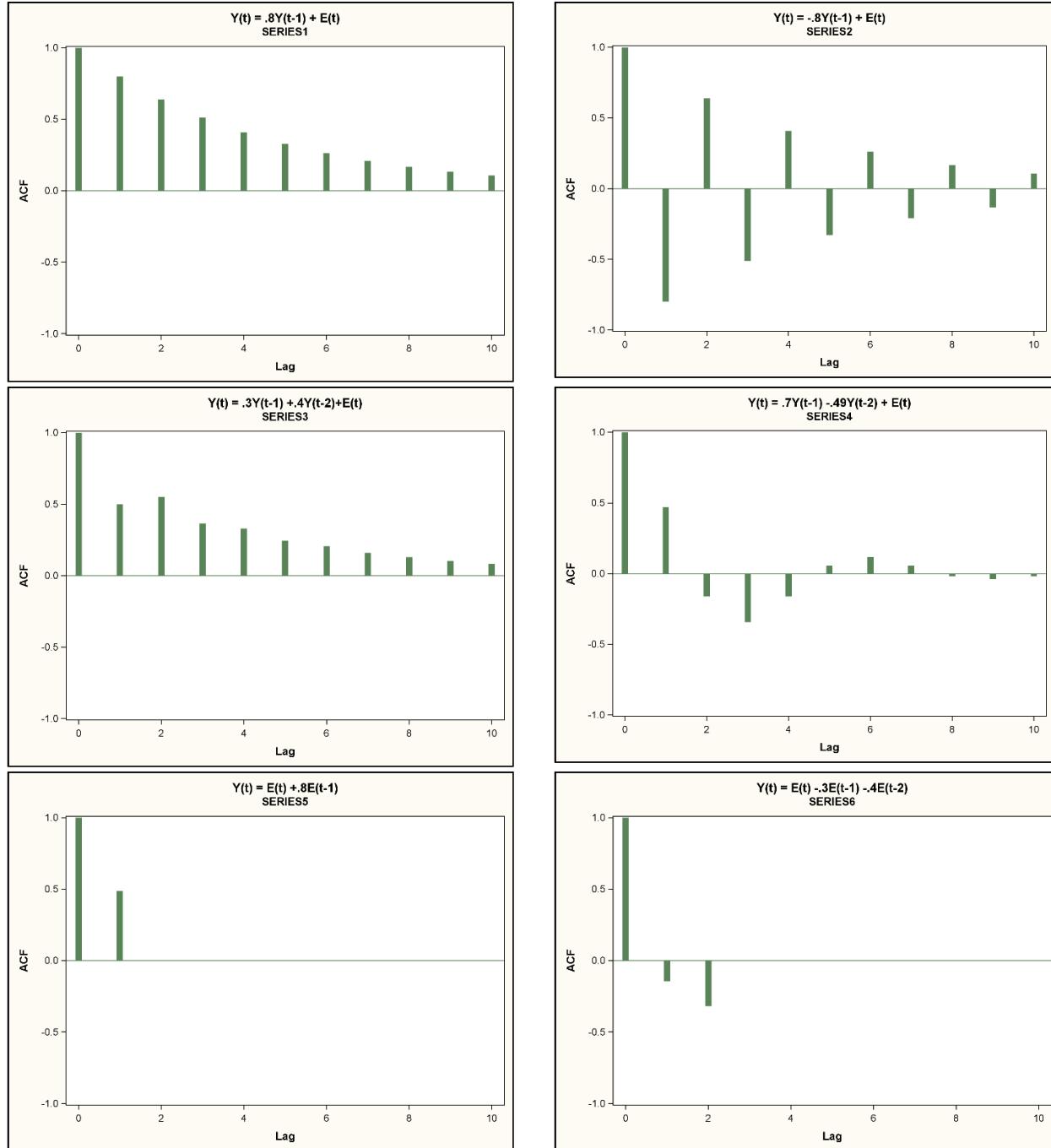
The ACFs for the eight series previously listed are in the following table.

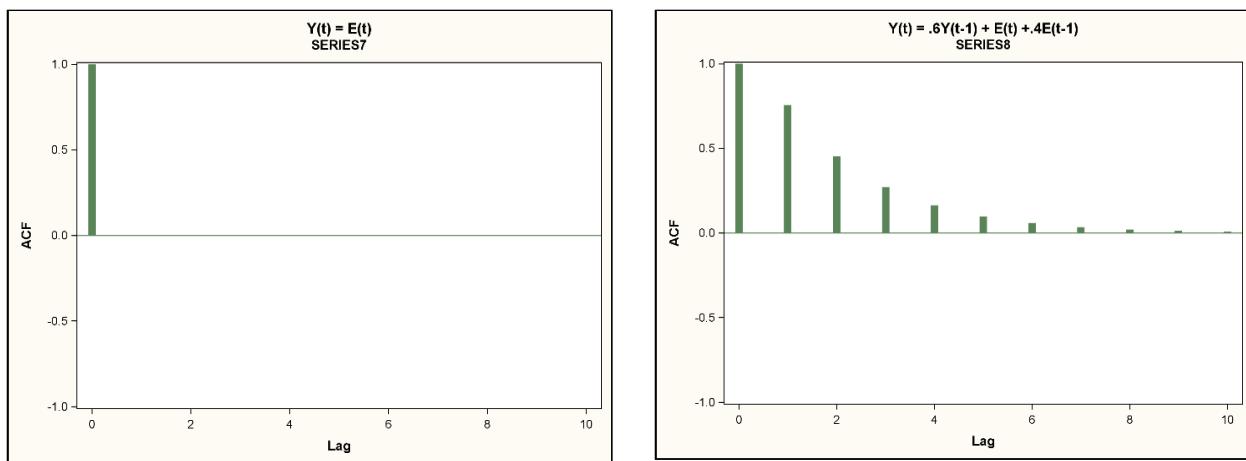
Series	Model, ACF
1	$Y_t = 0.8Y_{t-1} + e_t, \quad \rho(j) = 0.8^{ j }$
2	$Y_t = -0.8Y_{t-1} + e_t, \quad \rho(j) = (-0.8)^{ j }$
3	$Y_t = 0.3Y_{t-1} + 0.4Y_{t-2} + e_t, \quad \rho(1) = 0.5000,$ $\rho(j) = 0.3\rho(j-1) + 0.4\rho(j-2) \quad \text{for } j > 1$
4	$Y_t = 0.7Y_{t-1} - 0.49Y_{t-2} + e_t, \quad \rho(1) = 0.4698,$ $\rho(j) = 0.7\rho(j-1) - 0.49\rho(j-2) \quad \text{for } j > 1$
5	$Y_t = e_t + 0.8e_{t-1}, \quad \rho(1) = 0.4878, \quad \rho(j) = 0 \quad \text{for } j > 1$
6	$Y_t = e_t - 0.3e_{t-1} - 0.4e_{t-2}, \quad \rho(1) = -0.144,$ $\rho(2) = -0.32, \quad \rho(j) = 0 \quad \text{for } j > 2$
7	$Y_t = e_t, \quad \rho(0) = 1, \quad \rho(j) = 0 \quad \text{for } j > 0$

Series	Model, ACF
8	$Y_t - 0.6Y_{t-1} = e_t + 0.4e_{t-1}, \rho(0) = 1, \rho(1) = 0.7561,$ $\rho(j) = 0.6\rho(j-1) \text{ for } j > 1$

The ACFs are plotted in **Output 3.1**.

Output 3.1: Plotting Actual Autocorrelations for Series 1–8





Partial Autocorrelation Function

The PACF is motivated by the regression approach to the silver example in Chapter 2, “Simple Models: Autoregression.” First, regress Y_t on Y_{t-1} , and denote by $\hat{\pi}_1$, the coefficient on Y_{t-1} . Next, regress Y_t on Y_{t-1} , Y_{t-2} , and denote by $\hat{\pi}_2$, the coefficient on Y_{t-2} . Continue in this manner, regressing Y_t on Y_{t-1} , Y_{t-2} , ..., Y_{t-j} and calling the last coefficient $\hat{\pi}_j$. The $\hat{\pi}_j$ values are the estimated partial autocorrelations.

In an autoregression of order p , the coefficients $\hat{\pi}_j$ estimate 0s for all $j > p$. The theoretical partial autocorrelations π_j estimated by the $\hat{\pi}_j$ are obtained by solving equations similar to the regression normal equations:

$$\begin{bmatrix} \gamma(0) & \gamma(1) & \cdots & \gamma(j-1) \\ \gamma(1) & \gamma(0) & \cdots & \gamma(j-2) \\ \vdots & \vdots & & \vdots \\ \gamma(j-1) & \gamma(j-2) & \cdots & \gamma(0) \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_j \end{bmatrix} = \begin{bmatrix} \gamma(1) \\ \gamma(2) \\ \vdots \\ \gamma(j) \end{bmatrix}$$

For each j , let $\pi_j = b_j$. (A new set of equations is needed for each j .) As with autocorrelations, the π_j sequence is useful for identifying the form of a time series model. The PACF is most useful for identifying AR processes because, for an AR(p), the PACF is 0 beyond lag p . For MA or mixed (ARMA) processes, the theoretical PACF does not become 0 after a fixed number of lags.

You can solve the previous set of equations for the catalog of eight series currently under discussion. When you observe an estimated PACF $\hat{\pi}_j$, compare its behavior to the following behavior to choose a model. The following is a list of actual partial autocorrelations for Series 1–8:

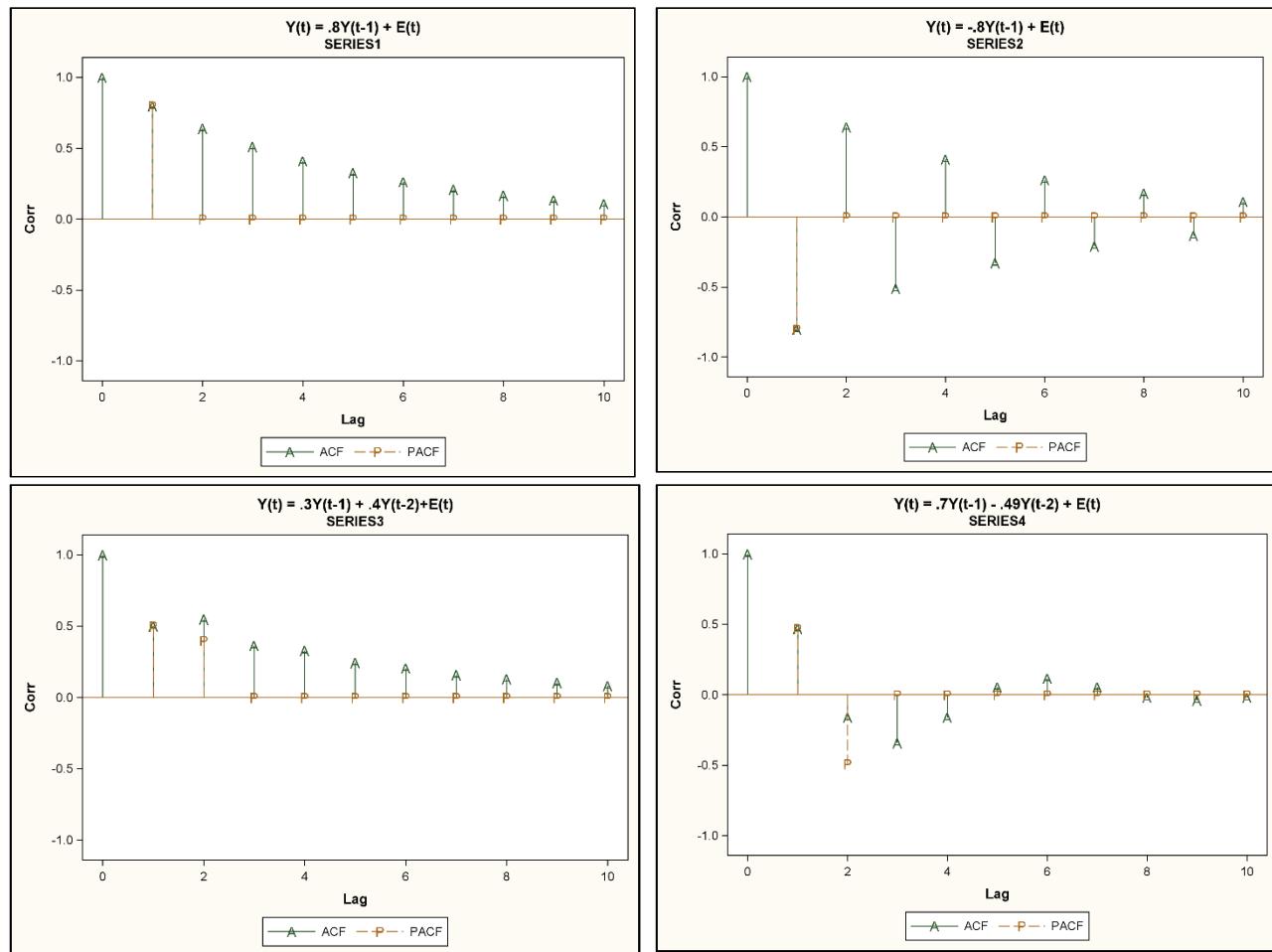
Series	Model	Lag 1	Lag 2	Lag 3	Lag 4	Lag 5
1	$Y_t = 0.8Y_{t-1} + e_t$	0.8	0	0	0	0
2	$Y_t = -0.8Y_{t-1} + e_t$	-0.8	0	0	0	0
3	$Y_t = 0.3Y_{t-1} + 0.4Y_{t-2} + e_t$	0.5	0.4	0	0	0
4	$Y_t = 0.7Y_{t-1} - 0.49Y_{t-2} + e_t$	0.4698	-0.4900	0	0	0
5	$Y_t = e_t + 0.8e_{t-1}$	0.4878	-0.3123	0.2215	-0.1652	0.1267
6	$Y_t = e_t - 0.3e_{t-1} - 0.4e_{t-2}$	-0.144	-0.3480	-0.1304	-0.1634	-0.0944
7	$Y_t = e_t$	0	0	0	0	0
8	$Y_t = 0.6Y_{t-1} + e_t + 0.4e_{t-1}$	0.7561	-0.2756	0.1087	-0.0434	0.0173

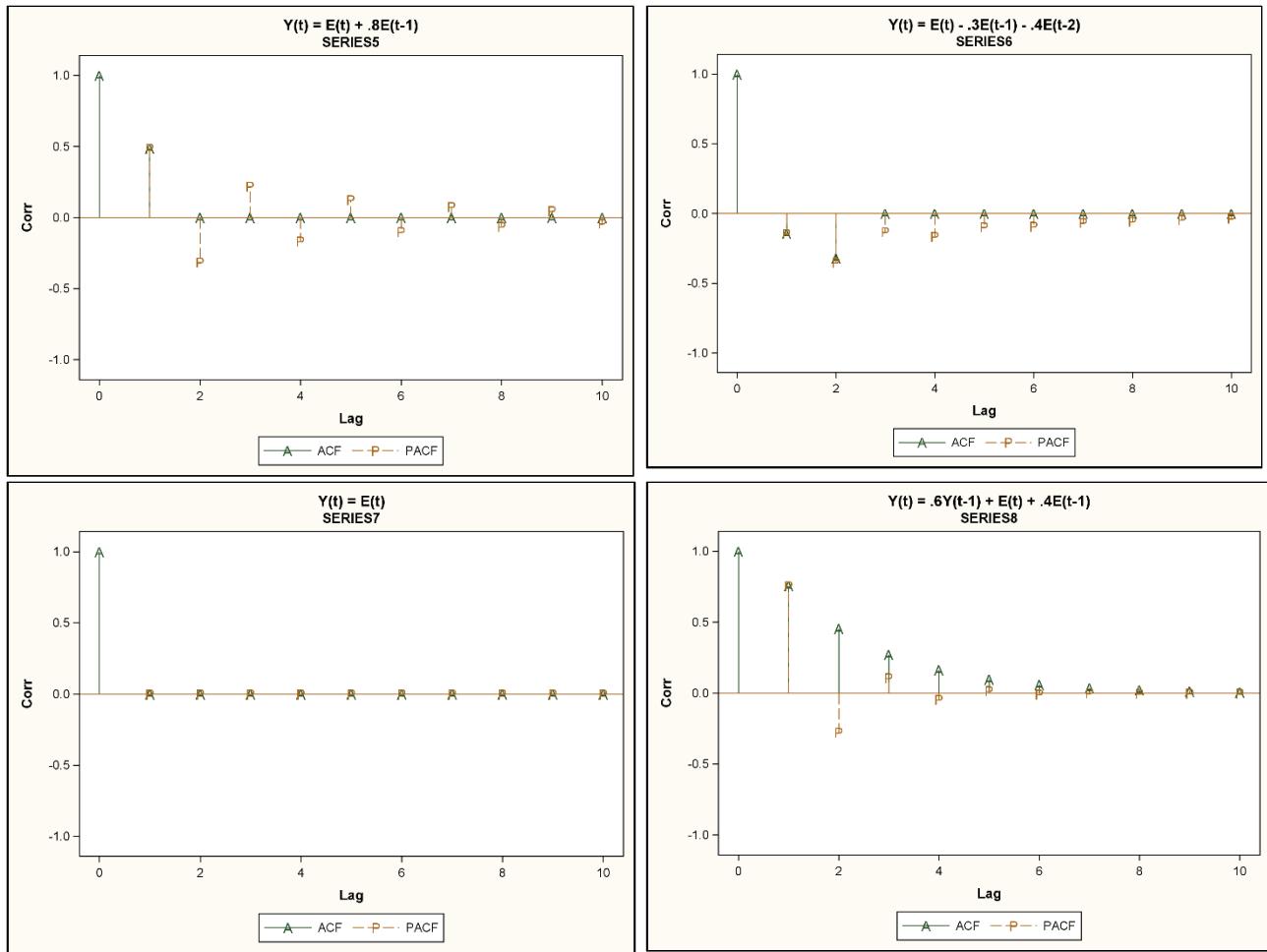
Plots of these values against lag number, with A used as a plot symbol for ACF and P for PACF, are given in **Output 3.2**. Here is a list of actual autocorrelations for Series 1–8:

Series	Model	Lag 1	Lag 2	Lag 3	Lag 4	Lag 5
1	$Y_t = 0.8Y_{t-1} + e_t$	0.8	0.64	0.512	0.410	0.328
2	$Y_t = -0.8Y_t + e_t$	-0.8	0.64	-0.512	0.410	-0.328
3	$Y_t = 0.3Y_{t-1} + 0.4Y_{t-2} + e_t$	0.500	0.550	0.365	0.330	0.245
4	$Y_t = 0.7Y_{t-1} - 0.49Y_{t-2} + e_t$	0.470	-0.161	-0.343	-0.161	0.055
5	$Y_t = e_t + 0.8e_{t-1}$	0.488	0	0	0	0
6	$Y_t = e_t - 0.3e_{t-1} - 0.4e_{t-2}$	-0.144	-0.32	0	0	0
7	$Y_t = e_t$	0	0	0	0	0
8	$Y_t = 0.6Y_{t-1} + e_t + 0.4e_{t-1}$	0.756	0.454	0.272	0.163	0.098

Output 3.2 shows the plots.

Output 3.2: Plotting Actual Autocorrelation Function and Actual Partial Autocorrelation Function for Series 1–8





Estimated ACF

Begin the PROC ARIMA analysis by estimating the three functions previously defined. Use these estimates to identify the form of the model. Define the estimated autocovariance $C(j)$ as follows:

$$C(j) = \frac{1}{n} \sum (Y_t - \bar{Y})(Y_{t+j} - \bar{Y})$$

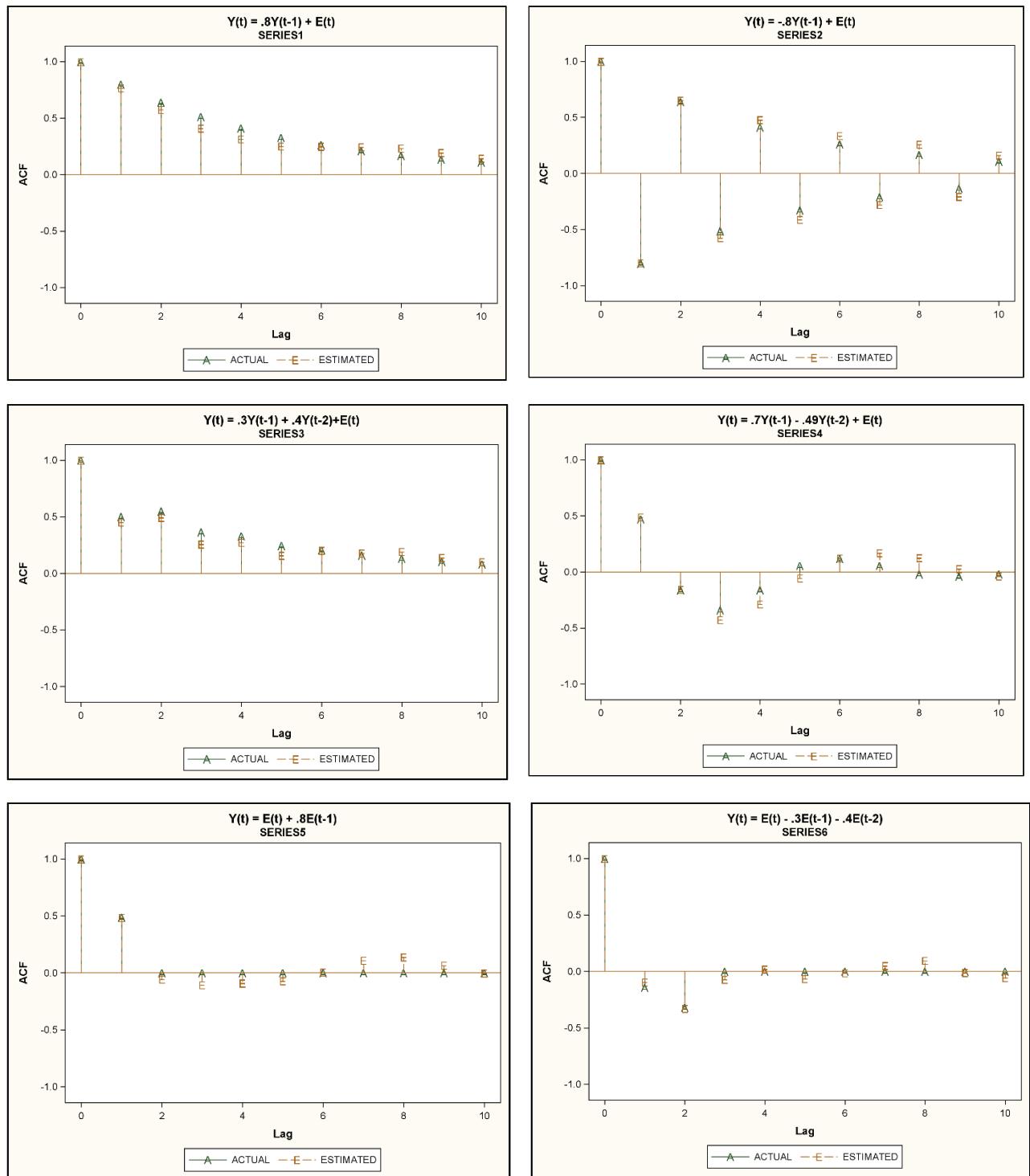
where the summation is from 1 to $n - j$ and \bar{Y} is the mean of the entire series. Define the estimated autocorrelation by the following:

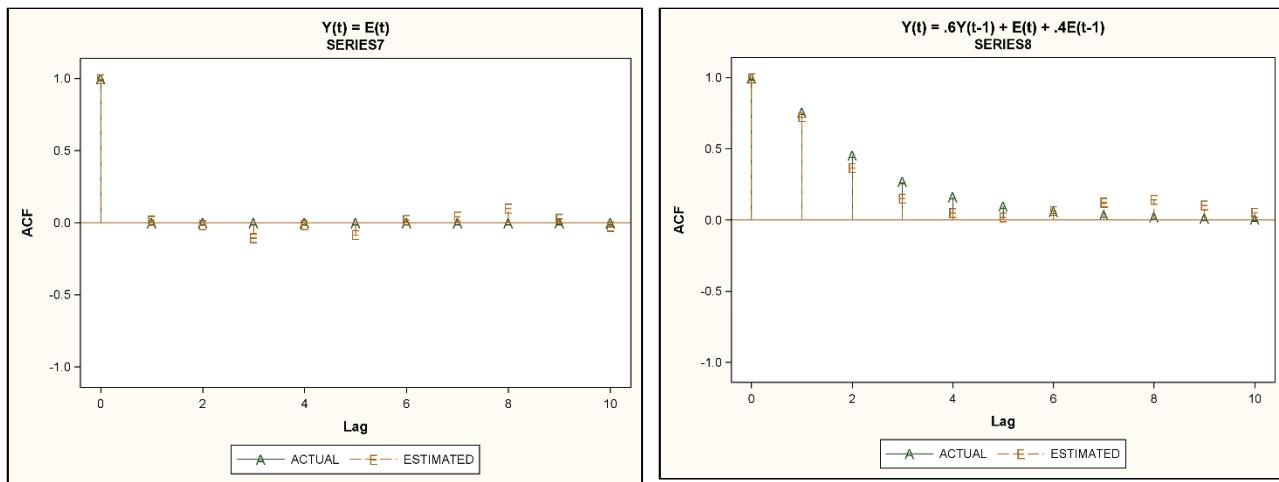
$$r(j) = \frac{C(j)}{C(0)}$$

Compute standard errors for autocorrelations in PROC ARIMA next:

- For autocorrelation $r(j)$, assign a variance $(\sum r^2(i))/n$ where the summation runs from $-j+1$ to $j-1$.
- The standard error is the square root of this variance.
- This variance $(\sum r^2(i))/n$ is appropriate under the hypothesis that $\gamma(i) = 0$ for $i \geq j$ while $\gamma(i) \neq 0$ for $i < j$.

The group of plots in **Output 3.3** illustrate the actual (A) and estimated (E) ACFs for the series. Each data series contains 150 observations. The purpose of the plots is to indicate the amount of sampling error in the estimates.

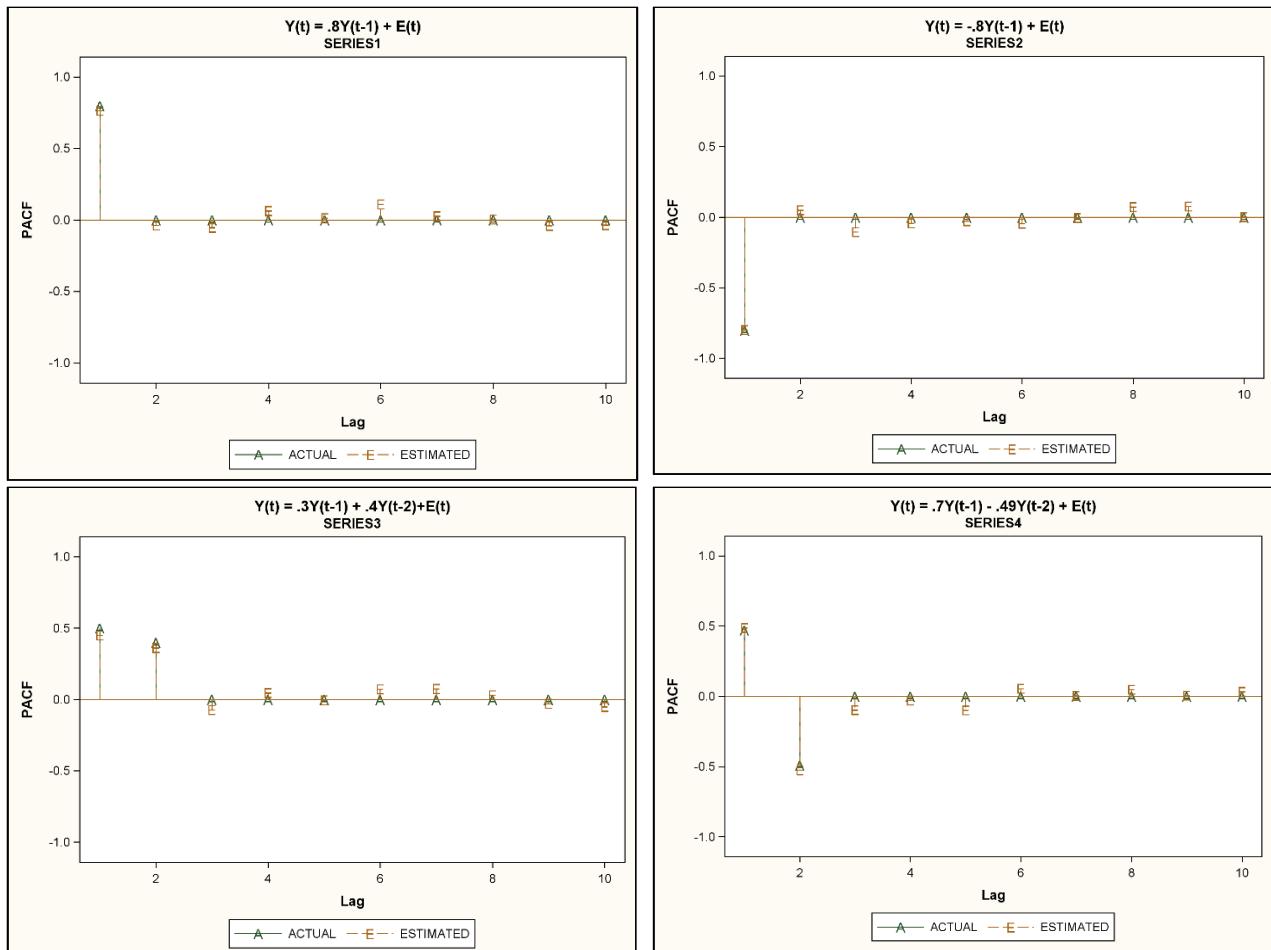
Output 3.3: Plotting Actual and Estimated Autocorrelation Functions for Series 1–4

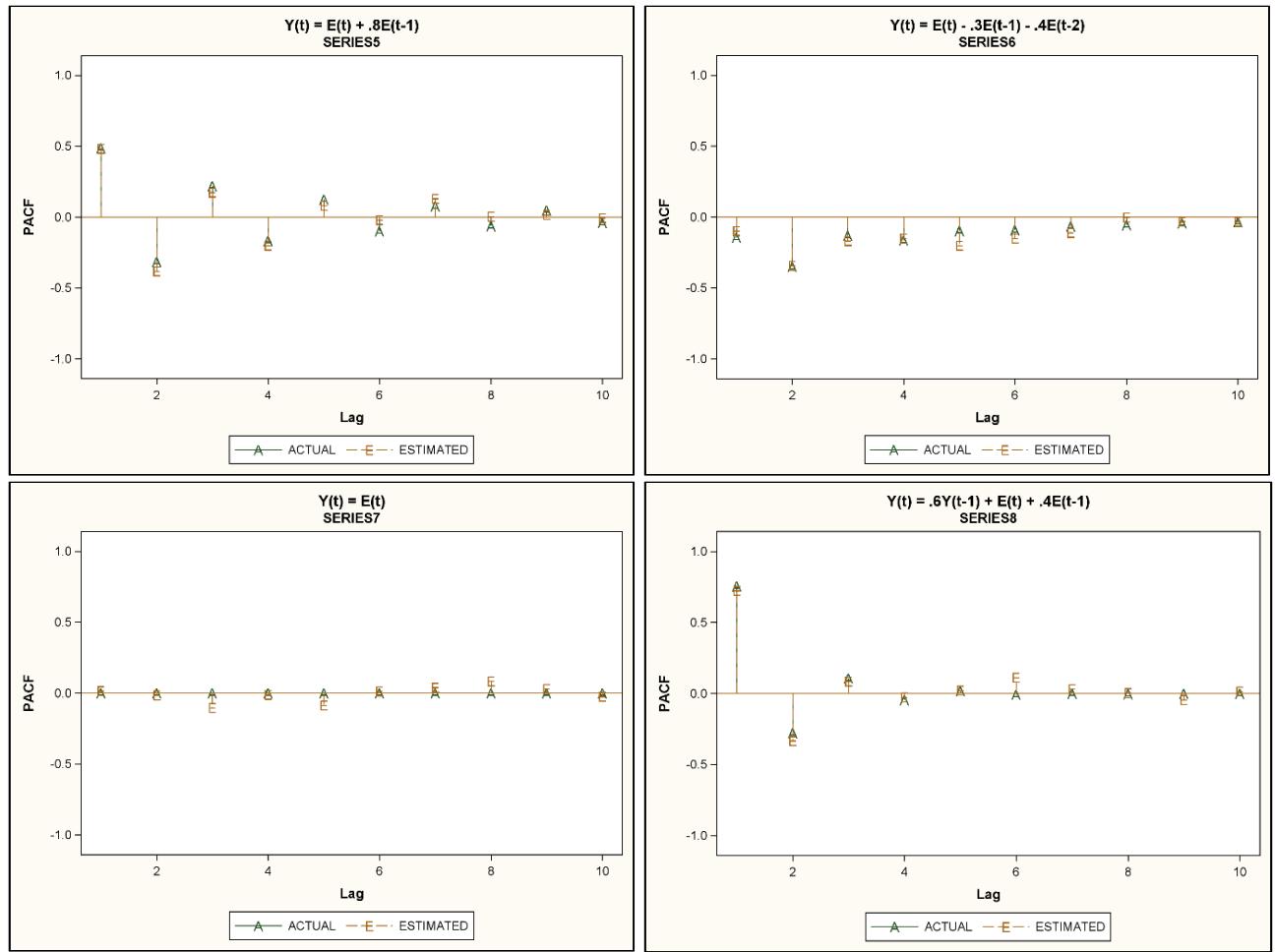


Estimated Partial Autocorrelation Function

The partial autocorrelations are defined in the PCAF section as solutions to equations involving the covariances $\gamma(j)$. To estimate these partial autocorrelations, substitute estimated covariances $C(j)$ for the actual covariances and solve. For j large enough that the actual partial autocorrelation π_j is 0 or nearly 0, an approximate standard error for the estimated partial autocorrelation is $n^{-1/2}$. The plots in **Output 3.4** illustrate the actual (A) and estimated (E) PACFs for the series.

Output 3.4: Plotting Actual and Estimated Partial Autocorrelation Functions for Series 1–8





Inverse Autocorrelation Function

The IACF of an ARMA(p,q) model is defined as the ACF of the ARMA(q,p) model that you obtain if you switch sides with the MA and AR operators. Thus, the inverse autocorrelation of $(1 - 0.8B)(Y_t - \mu) = e_t$ is defined as the ACF of $Y_t - \mu = e_t - 0.8e_{t-1}$.

In the catalog of Series 1–8, for example, the IACF of Series 3 is the same as the ACF of Series 6, and vice versa.

Estimated Inverse Autocorrelation Function

Suppose you know that a series comes from an AR(3) process. Fit an AR(3) model to obtain estimated coefficients. For example,

$$Y_t - \mu = 0.300(Y_{t-1} - \mu) + 0.340(Y_{t-2} - \mu) - 0.120(Y_{t-3} - \mu) + e_t$$

The inverse model is the moving average $Y_t - \mu = e_t - 0.300e_{t-1} - 0.340e_{t-2} + 0.120e_{t-3}$.

At lag 0 the inverse autocovariances are estimated by the following:

$$(1 + 0.300^2 + 0.340^2 + 0.120^2)\sigma^2$$

At lag 1, they are estimated by this:

$$(-0.300 + (0.300)(0.340) - (0.340)(0.120))\sigma^2$$

At lag 2, the estimation is by this:

$$(-0.340 - (0.300)(0.120))\sigma^2$$

And at lag 3, they are estimated by $0.120\sigma^2$.

In general, you do not know the order p of the process, nor do you know the form (it might be MA or ARMA). Use the fact that any invertible ARMA series can be represented as an infinite-order AR. Therefore, it can be approximated by an AR(p) with p large. (See **Section 3.3.1**.)

Set p to the minimum of the NLAG value and half of the number of observations after differencing. Then, do the following:

1. Fit an AR(p) to the data.
2. Using the estimated coefficients, compute covariances for the corresponding MA series as illustrated above for $p = 3$.
3. Assign standard errors of $n^{-1/2}$ to the resulting estimates.

3.3.3 Chi-Square Check of Residuals

In the identification stage, PROC ARIMA uses the autocorrelations to form a statistic whose approximate distribution is chi-square under the null hypothesis that the series is white noise. The test is the Ljung-Box modification of the Box-Pierce Q statistic. Both Q statistics are described in Box, Jenkins, and Riensel (1994). The Ljung-Box modification is described in Ljung and Box (1978, p. 297). The formula for this statistic is:

$$n(n+2) \sum_{j=1}^k r^2(j)/(n-j)$$

In this formula, $r(j)$ is the estimated autocorrelation at lag j and k can be any positive integer. In PROC ARIMA, several k s are used.

Later in the modeling stage, PROC ARIMA calculates the same statistic on the model residuals to test the hypothesis that they are white noise. The statistic is compared to critical values from a chi-square distribution. If your model is correct, the residuals should be white noise and the chi-square statistic should be small (the PROB value should be large). A significant chi-square statistic indicates that your model does not fit well.

3.3.4 Summary of Model Identification

At the identification stage, you compute the ACF, PACF, and IACF. Behavior of the estimated functions is the key to model identification. The behavior of functions for different processes is summarized in the following table:

	MA(q)	AR(p)	ARMA(p,q)	White Noise
ACF	D(q)	T	T	0
PACF	T	D(p)	T	0
IACF	T	D(p)	T	0

Notes: D(q) means that the function drops off to 0 after lag q ; T means the function tails off exponentially; and 0 means the function is 0 at all nonzero lags.

3.4 Examples and Instructions

The following pages contain results for 150 observations generated from each of the eight sample series previously discussed. The ACFs correspond to the Es in **Output 3.3**. Even with 150 observations, considerable variation occurs.

To obtain all of the output shown for the first series Y1, use these SAS statements:

```
proc arima data=series;
  identify var=y1 nlag=10;
run;
```

The VAR= option is required. The NLAG= option gives the number of autocorrelations to be computed and defaults to 24. When you fit an ARIMA(p,d,q), NLAG+1 must be greater than $p+d+q$ to obtain initial parameter estimates. For the ARMA(p,q) models discussed so far, d is 0.

The following options can also be used:

NOPRINT

suppresses printout. This is useful because you must use an IDENTIFY statement prior to an ESTIMATE statement. If you have seen the output on a previous run, you might want to suppress it with this option.

CENTER

subtracts the series mean from each observation prior to the analysis.

DATA=SASdataset

specifies the SAS data set to be analyzed (the default is the most recently created SAS data set).

3.4.1 IDENTIFY Statement for Series 1-8

The following SAS statements, when used on the generated data, produce **Output 3.5**:

```
proc arima data=series;
  identify var=y1 nlag=10;
  identify var=y2 nlag=10;
  more sas statements
  identify var=y8 nlag=10;
run;
```

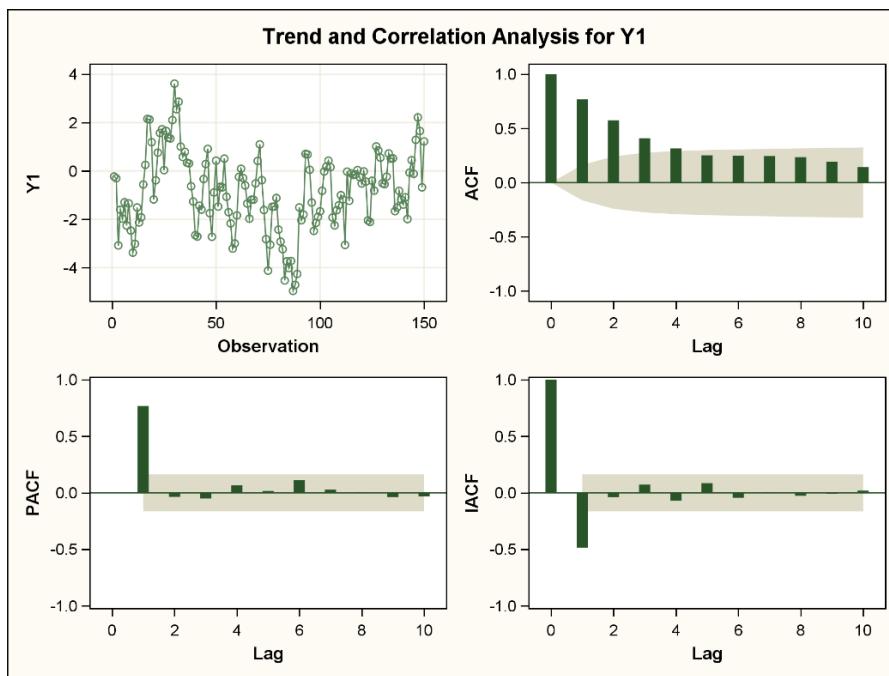
Try to identify all eight of these series. These are presented in **section 3.3.2**, so you can check your diagnosis against the actual model.

Output 3.5a: Using the IDENTIFY Statement for Series 1

The ARIMA Procedure

Name of Variable = Y1	
Mean of Working Series	-0.83571
Standard Deviation	1.610893
Number of Observations	150

Autocorrelation Check for White Noise							
To Lag	Chi-Square	DF	Pr > ChiSq	Autocorrelations			
6	202.72	6	<.0001	0.768	0.576	0.410	0.316
				0.251	0.248		

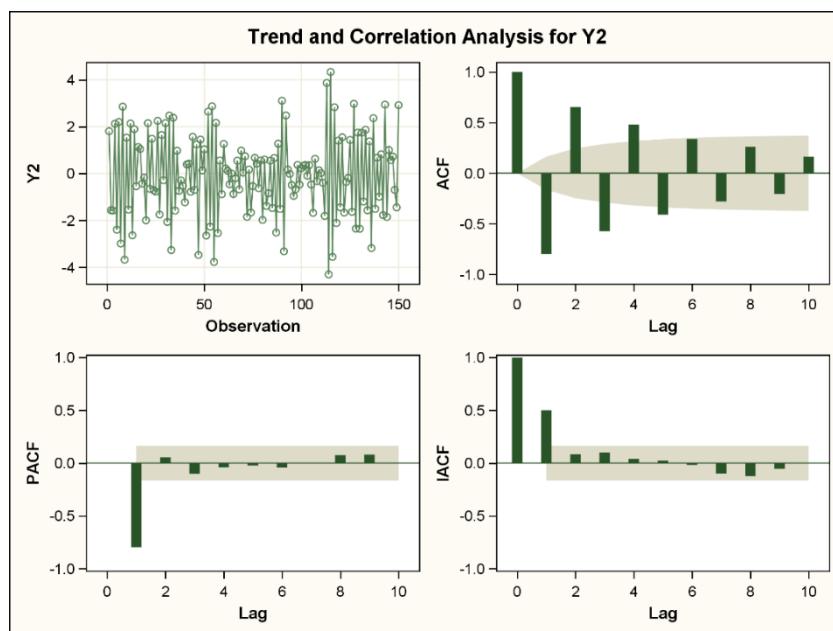


Output 3.5b: Using the IDENTIFY Statement for Series 2

The ARIMA Procedure

Name of Variable = Y2	
Mean of Working Series	-0.07304
Standard Deviation	1.740946
Number of Observations	150

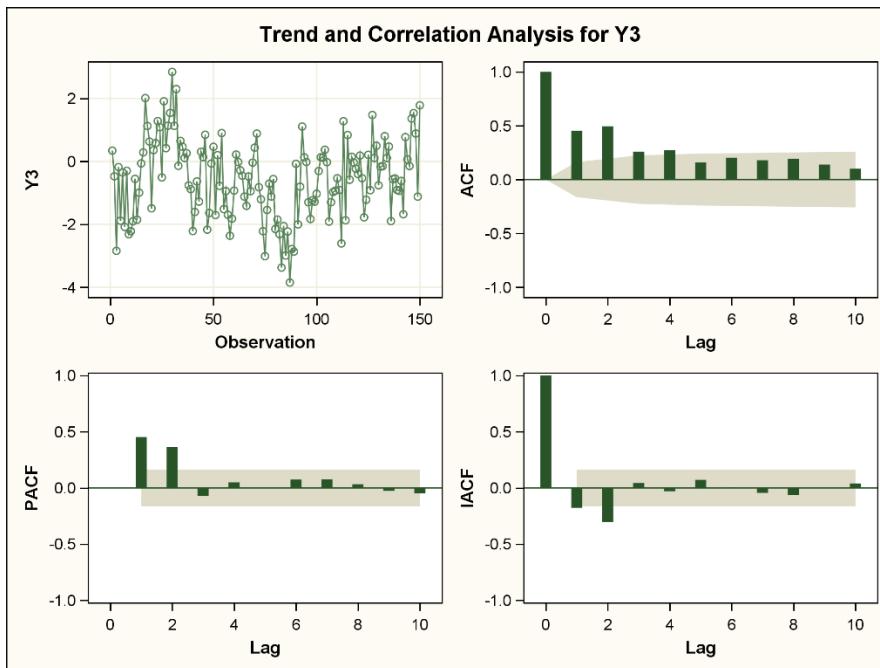
Autocorrelation Check for White Noise									
To Lag	Chi-Square	DF	Pr > ChiSq	Autocorrelations					
6	294.24	6	<.0001	-0.796	0.654	-0.573	0.480	-0.410	0.338



Output 3.5c: Using the IDENTIFY Statement for Series 3**The ARIMA Procedure**

Name of Variable = Y3	
Mean of Working Series	-0.55064
Standard Deviation	1.237272
Number of Observations	150

Autocorrelation Check for White Noise								
To Lag	Chi-Square	DF	Pr > ChiSq	Autocorrelations				
6	101.56	6	<.0001	0.453	0.494	0.258	0.273	0.159
				0.203				

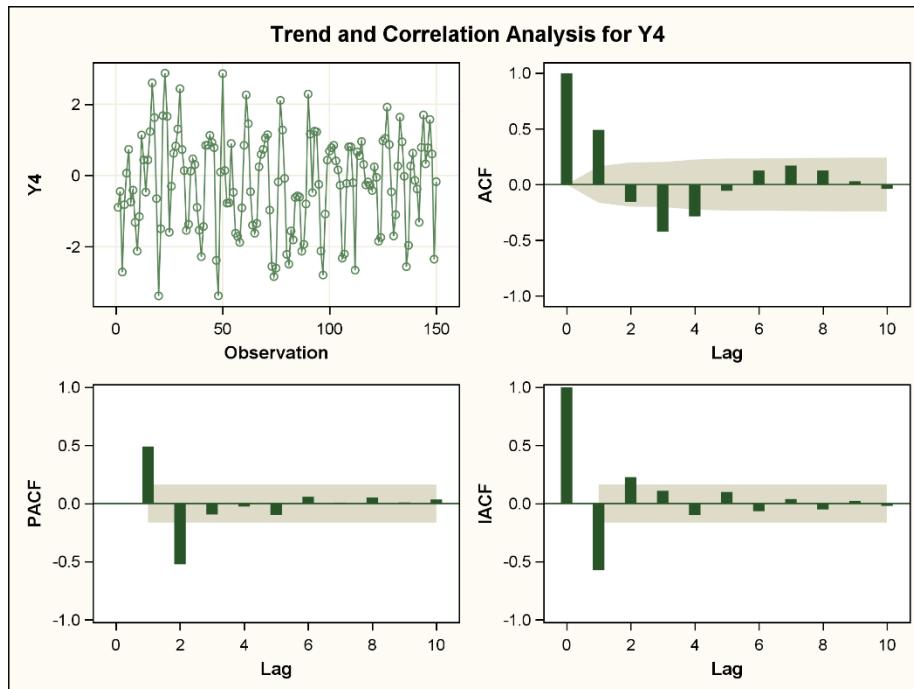


Output 3.5d: Using the IDENTIFY Statement for Series 4

The ARIMA Procedure

Name of Variable = Y4	
Mean of Working Series	-0.21583
Standard Deviation	1.381192
Number of Observations	150

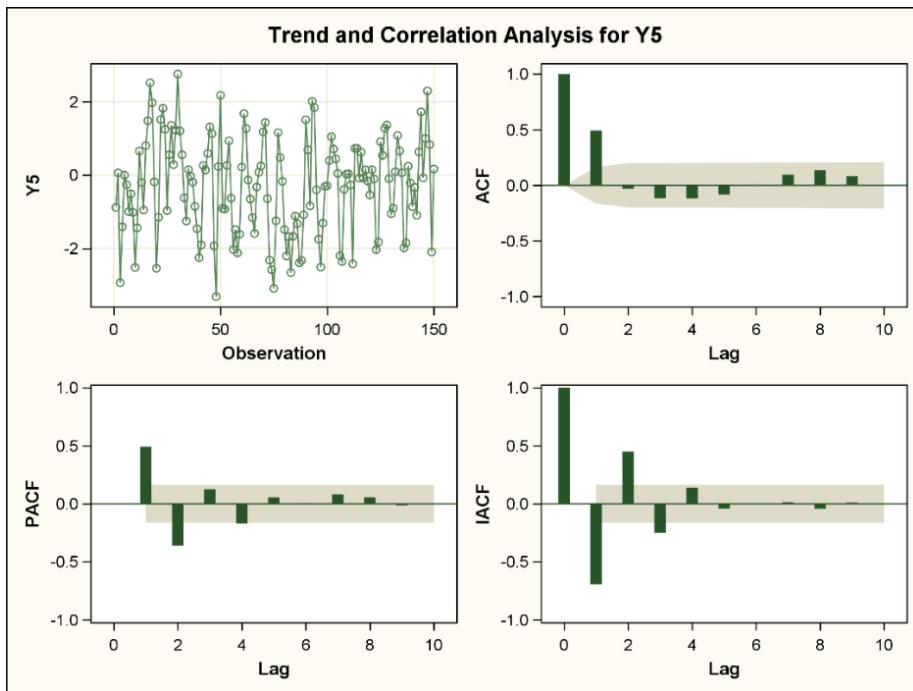
Autocorrelation Check for White Noise									
To Lag	Chi-Square	DF	Pr > ChiSq	Autocorrelations					
6	84.33	6	<.0001	0.490	-0.156	-0.425	-0.286	-0.056	0.125



Output 3.5e: Using the IDENTIFY Statement for Series 5**The ARIMA Procedure**

Name of Variable = Y5	
Mean of Working Series	-0.30048
Standard Deviation	1.316518
Number of Observations	150

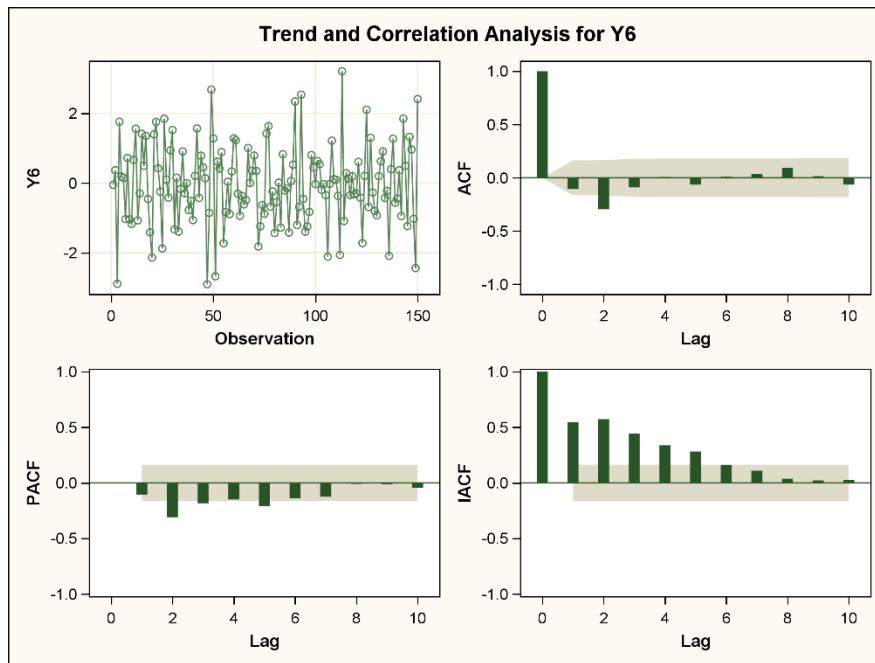
Autocorrelation Check for White Noise								
To Lag	Chi-Square	DF	Pr > ChiSq	Autocorrelations				
6	42.48	6	<.0001	0.492	-0.032	-0.116	-0.117	-0.084
								0.006



Output 3.5f: Using the IDENTIFY Statement for Series 6**The ARIMA Procedure**

Name of Variable = Y6	
Mean of Working Series	-0.04253
Standard Deviation	1.143359
Number of Observations	150

Autocorrelation Check for White Noise									
To Lag	Chi-Square	① DF	② Pr > ChiSq	Autocorrelations					
6	17.03	6	0.0092	-0.105	-0.295	-0.091	0.006	-0.065	0.009



Look at Y6. First, observe that the calculated Q statistic ① is 17.03, which would be compared to a chi-square distribution with six degrees of freedom. The 5% critical value is 12.59, so you have significant evidence against the null hypothesis that the considered model is adequate. Because no model is specified, this Q statistic simply tests the hypothesis that the original data are white noise. The number 0.0092 ② is the area under the chi-square distribution to the right of the calculated 17.03. Because 0.0092 is less than 0.05, without recourse to a chi-square table, you see that 17.03 is to the right of the 5% critical value. Either way, you decide that Y6 is not a white noise series.

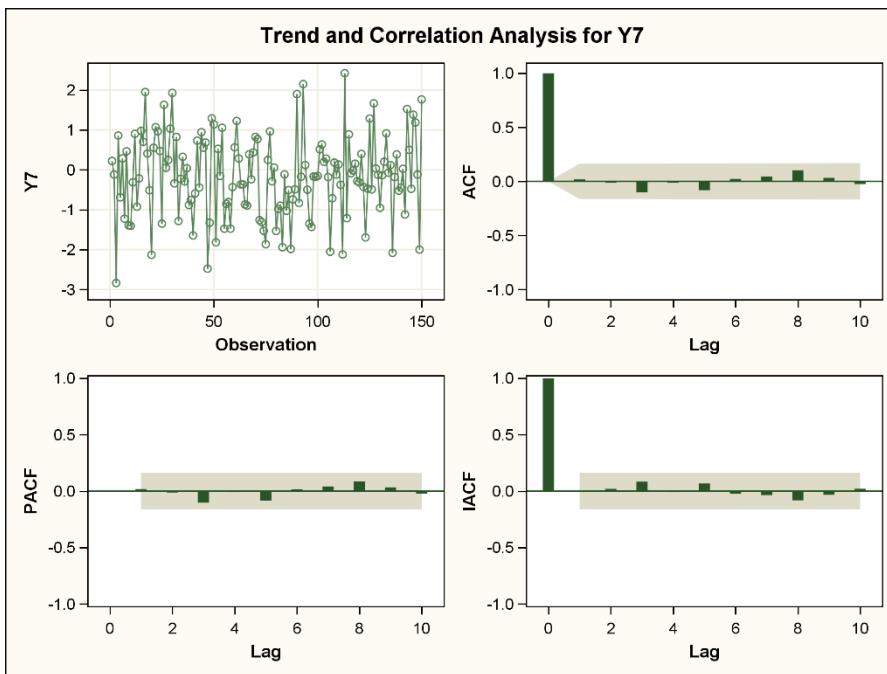
A model is needed for Y6. The PACF and IACF are nonzero through several lags, which means that an AR diagnosis requires perhaps seven lags. A model with few parameters is preferable. The ACF is near 0 after two lags, indicating that you can choose an MA(2). Because an MA model has a persistently nonzero PACF and IACF, the MA(2) diagnosis seems appropriate. At this stage, you have identified the form of the model and can assign the remainder of the analysis to PROC ARIMA. You must identify the model because PROC ARIMA does not do it automatically.

The generated series has 150 observations. Note the width of the standard error bands on the autocorrelations. Even with 150 observations, reading fine detail from the ACF is unlikely. Your goal is to use these functions to limit your search to a few plausible models, rather than to pinpoint one model at the identification stage.

Output 3.5g: Using the IDENTIFY Statement for Series 7**The ARIMA Procedure**

Name of Variable = Y7	
Mean of Working Series	-0.15762
Standard Deviation	1.023007
Number of Observations	150

Autocorrelation Check for White Noise								
To Lag	Chi-Square	③ DF	④ Pr > ChiSq	Autocorrelations				
6	2.85	6	0.8269	0.019	-0.012	-0.103	-0.012	-0.081
					0.022			

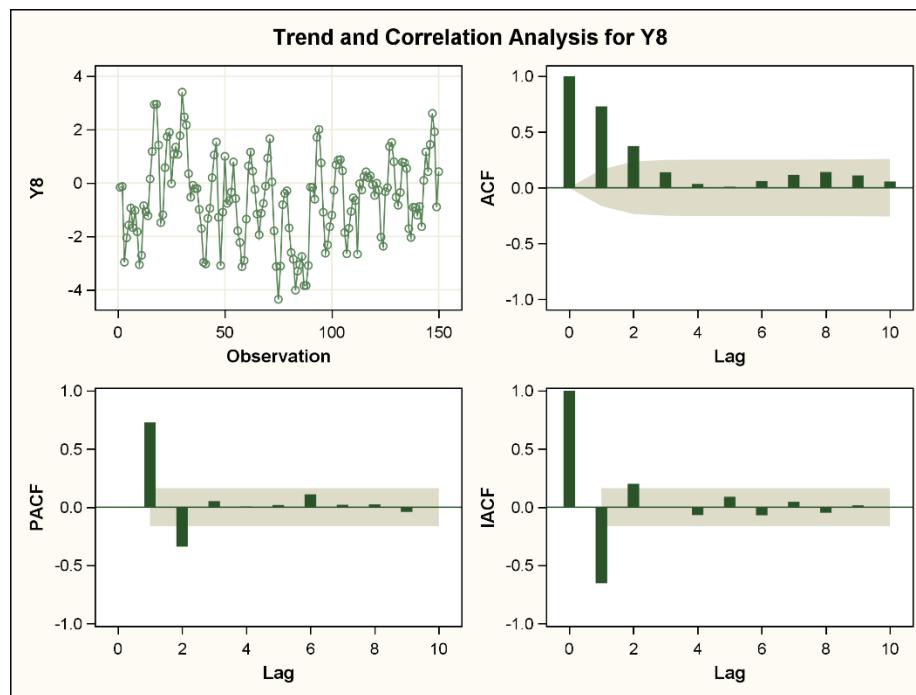


Contrast Y6 with Y7, where the calculated statistic 2.85 ③ has an area 0.8269 ④ to its right. The value 2.85 is far to the left of the critical value and nowhere near significance. Therefore, you decide that Y7 is a white noise series.

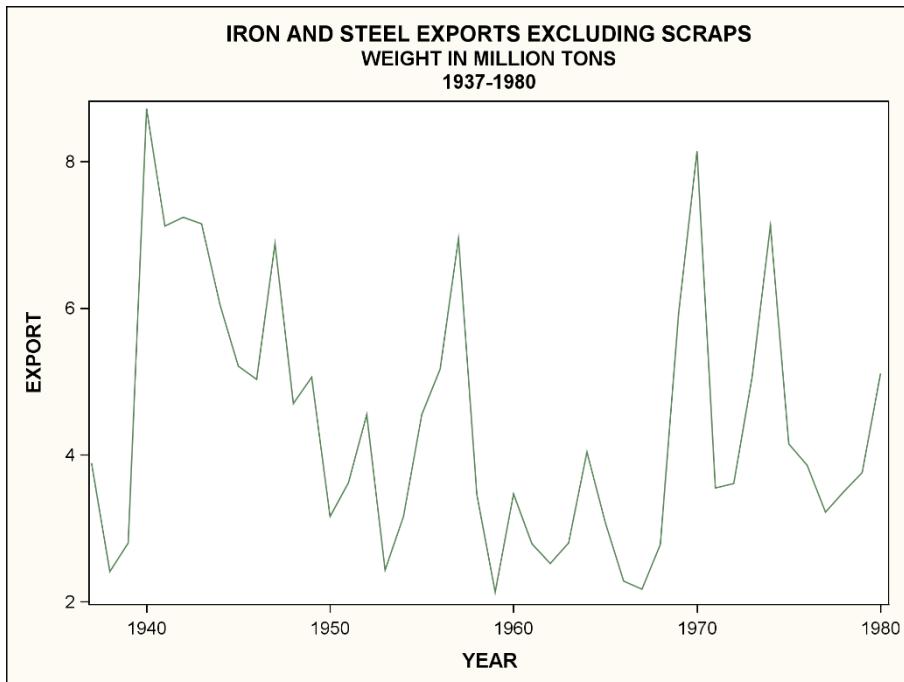
Output 3.5h: Using the IDENTIFY Statement for Series 8**The ARIMA Procedure**

Name of Variable = Y8	
Mean of Working Series	-0.57405
Standard Deviation	1.591833
Number of Observations	150

Autocorrelation Check for White Noise								
To Lag	Chi-Square	DF	Pr > ChiSq	Autocorrelations				
6	106.65	6	<.0001	0.729	0.373	0.139	0.034	0.010

**3.4.2 Example: Iron and Steel Export Analysis**

The U.S. iron and steel export yearly series (Fairchild Publications 1981) shown in **Output 3.6** is a good illustration of model identification.

Output 3.6: Plotting a Yearly Series

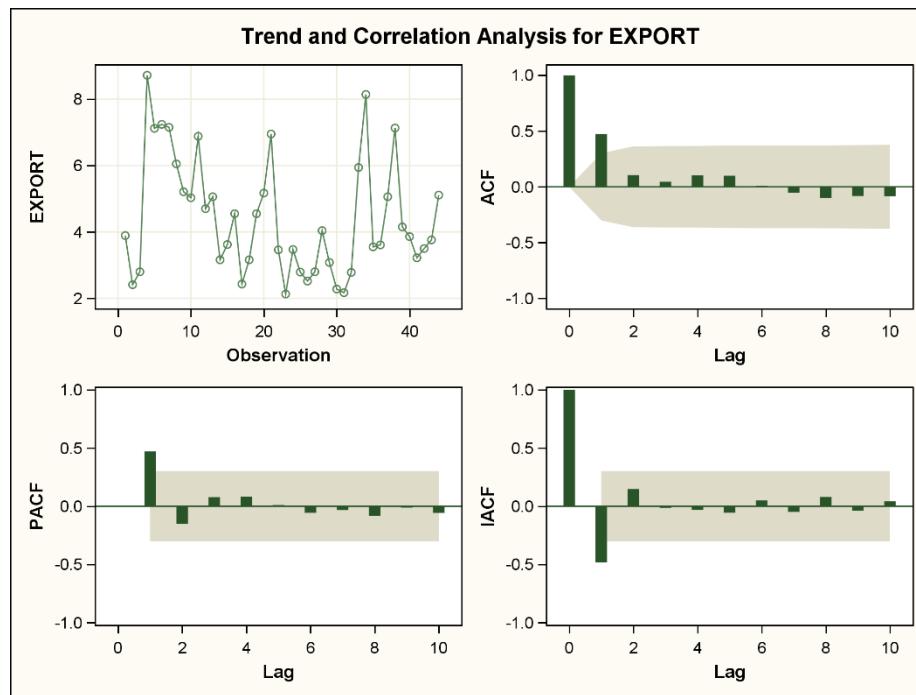
The following statements produce the results in **Output 3.7**:

```
proc arima data=steel;
  identify var=export nlag=10;
run;
```

Output 3.7: Identifying a Model Using the IDENTIFY Statement**The ARIMA Procedure**

Name of Variable = EXPORT	
Mean of Working Series	4.418182
Standard Deviation	1.73354
Number of Observations	44

Autocorrelation Check for White Noise									
To Lag	Chi-Square	DF	Pr > ChiSq	Autocorrelations					
6	12.15	6	0.0586	0.472	0.104	0.045	0.103	0.099	0.008



Although the Q (Chi-square) statistic ❶ fails by a slim margin to be significant, the lag 1 autocorrelation 0.472 ❷ is beyond the two standard error bands. Thus, you want to fit a model despite the Q value. From the ACF, it appears that an MA(1) is appropriate. From the PACF and IACF, an AR(1) also appears consistent with these data. You can fit both and select the one with the smallest error mean square. To fit the MA(1) model, use the statement:

```
ESTIMATE Q=1;
```

For the AR(1) model use the statement:

```
ESTIMATE P=1;
```

Suppose you overfit, using an MA(2) as an initial step. Specify these statements:

```
proc arima data=steel;
  identify var=export noprint;
  estimate q=2;
run;
```

Any ESTIMATE statement must be preceded by an IDENTIFY statement. In this example, NOPRINT suppresses the printout of ACF, IACF, and PACF.

Output 3.8: Fitting an MA(2) Model with the ESTIMATE Statement

The ARIMA Procedure

Conditional Least Squares Estimation					
Parameter	Estimate	Standard Error	t Value	Approx Pr > t	Lag
MU	4.43400	0.39137	11.33	<.0001	0
MA1,1	-0.56028	0.15542 ❷	-3.60	0.0008	1
MA1,2	-0.13242	0.15535	-0.85	0.3990	2

Constant Estimate	4.433999
Variance Estimate	2.433068
Std Error Estimate	1.559829

AIC	166.8821
SBC	172.2347
Number of Residuals	44

* AIC and SBC do not include log determinant.

Correlations of Parameter Estimates			
Parameter	MU	MA1,1	MA1,2
MU	1.000	-0.013	-0.011
MA1,1	-0.013	1.000	0.492
MA1,2	-0.011	0.492	1.000

Autocorrelation Check of Residuals									
To Lag	Chi-Square	DF	Pr > ChiSq	Autocorrelations					
6	0.58	4	0.9653	-0.002	-0.006	0.006	0.060	0.081	-0.032
12	2.81	10	0.9855	0.005	-0.077	-0.035	0.008	-0.163	0.057
18	6.24	16	0.9853	0.066	-0.005	0.036	-0.098	0.123	-0.125
24	12.10	22	0.9553	-0.207	-0.086	-0.102	-0.068	0.025	-0.060

Model for variable EXPORT	
Estimated Mean	4.433999

Moving Average Factors	
Factor 1:	$1 + 0.56028 B^{**}(1) + 0.13242 B^{**}(2)$

Note that the Q statistics ❶ in **Output 3.8** are quite small, indicating a good fit for the MA(2) model. However, when you examine the parameter estimates and their t statistics ❷, you see that more parameters were fit than necessary. An MA(1) model is appropriate because the t statistic for the lag 2 parameter is only -0.85 . Also, it is wise to ignore the fact that the previous Q was insignificant due to the large t value, -3.60 , associated with the lag 1 coefficient. In **Output 3.7** the Q was calculated from six autocorrelations, and the large lag 1 autocorrelation's effect was diminished by the other five small autocorrelations.

You now fit an MA(1) model using these statements:

```
proc arima data=steel;
  identify var=export noprint;
  estimate q=1;
run;
```

The results are shown in **Output 3.9**.

Output 3.9: Fitting an MA(1) Model with the ESTIMATE Statement

The ARIMA Procedure

Conditional Least Squares Estimation					
Parameter	Estimate	Standard Error	t Value	Approx Pr > t	Lag
MU	4.42102	0.34703	12.74	<.0001	0
MA1,1	-0.49827	0.13512	-3.69	0.0006	1

Constant Estimate	4.421016
Variance Estimate	2.412583
Std Error Estimate	1.553249
AIC	165.5704
SBC	169.1388
Number of Residuals	44

* AIC and SBC do not include log determinant.

Correlations of Parameter Estimates		
Parameter	MU	MA1,1
MU	1.000	-0.008
MA1,1	-0.008	1.000

Autocorrelation Check of Residuals									
To Lag	❶ Chi-Square	DF	Pr > ChiSq	Autocorrelations					
6	1.31	5	0.9336	0.059	0.094	-0.028	0.085	0.075	-0.020
12	3.23	11	0.9873	-0.006	-0.079	-0.052	-0.013	-0.146	0.039
18	6.68	17	0.9874	0.063	-0.001	0.044	-0.092	0.096	-0.149
24	14.00	23	0.9268	-0.206	-0.135	-0.114	-0.084	0.014	-0.072

❷ Model for variable EXPORT
Estimated Mean 4.421016

Moving Average Factors
Factor 1: 1 + 0.49827 B**(1)

The Q statistics ❶ are still small, so you have no evidence of a lack of fit for the MA(1) model. The estimated model is now ❷:

$$Y_t = 4.421 + e_t + .4983e_{t-1}$$

3.4.3 Estimation Methods Used in PROC ARIMA

How does PROC ARIMA estimate this MA coefficient? As in the AR case, three techniques are available:

- conditional least squares (CLS)
- unconditional least squares (ULS)
- maximum likelihood (ML)

In the CLS method, you attempt to minimize:

$$\sum_{t=p+1}^n e_t^2$$

Here, p is the order of the AR part of the process and e_t is a residual. Consider the following example:

$$e_t = Y_t - \hat{\mu} + \hat{\beta}(e_{t-1})$$

Here, $\hat{\mu}$ and $\hat{\beta}$ are parameter estimates. Begin by assuming $e_0 = 0$. ARIMA computations indicate that $\hat{\mu} = 4.421$ and $\hat{\beta} = -0.4983$ provide the minimum for the iron export data.

To illustrate further, suppose you are given data Y_1, Y_2, \dots, Y_6 , where you assume

$$Y_t = e_t - \beta e_{t-1}$$

Suppose you want to estimate β from the following data:

	Sum of Squares						
Y_t	-12	-3	7	9	4	-7	
$\hat{e}_t(-0.29)$	-12	0.48	6.86	7.01	1.97	-7.57	301.63
$\hat{e}_t(-0.30)$	-12	0.60	6.82	6.95	1.91	-7.57	300.26
$W_t(-0.30)$	0	-12	4	6	6	0	

You find that $\hat{\gamma}(0) = 57.89$ and $\hat{\gamma}(1) = 15.26$. Their ratio results in $\hat{\rho}(1) = -\beta / (1 + \beta^2) = 0.2636$, which yields the initial estimate $\beta = -0.29$. Now compute $\hat{e}_t = Y_t - 0.29\hat{e}_{t-1}$.

Starting with $e_0 = 0$, values of \hat{e}_t are listed under the Y_t values. Example computations are as follows:

$$\hat{e}_1 = Y_1 - 0.29(0) = -12$$

$$\hat{e}_2 = -3 - 0.29(-12) = 0.48$$

$$\hat{e}_3 = 7 - 0.29(0.48) = 6.86$$

Continuing and summing these estimated squared errors gives the following:

$$\hat{e}_1^2 + \hat{e}_2^2 + \cdots + \hat{e}_6^2 = 301.63$$

Perhaps you can improve on 301.63. For example, using $\hat{\beta} = -0.30$, you can add a second row of e_t values to the previous list and compute:

$$\hat{e}_1^2 + \hat{e}_2^2 + \cdots + \hat{e}_6^2 = 300.26$$

The larger (in magnitude) $\hat{\beta}$ gives a smaller sum of squares, so you would like to continue increasing the magnitude of $\hat{\beta}$, but by how much? Letting β_0 be the true value of the parameter, you can use Taylor series expansion to write the following expression:

$$e_t(\beta_0) = e_t(\hat{\beta}) + W_t(\hat{\beta})(\beta_0 - \hat{\beta}) + R_t \quad (3.2)$$

In this equation, W_t is the derivative of e_t with respect to β , and R_t is a remainder term. Rearranging equation 3.2 and ignoring the remainder yields the following:

$$e_t(\hat{\beta}) = -W_t(\hat{\beta})(\beta_0 - \hat{\beta}) + e_t(\beta_0)$$

Because $e_t(\beta_0)$ is white noise, this looks like a regression equation that you can use to estimate $\beta_0 - \hat{\beta}$. You need to compute the derivative W_t . Derivatives are defined as limits, for example:

$$W_t(\hat{\beta}) = \lim_{\delta \rightarrow 0} \frac{e_t(\hat{\beta} + \delta) - e_t(\hat{\beta})}{\delta}$$

You have now computed $e_t(-0.29)$ and $e_t(-0.30)$, so, using $\delta = 0.01$, you can approximate $W_t(-0.30)$ by the following:

$$\frac{e_t(-0.29) - e_t(-0.30)}{0.01}$$

Recall the third row of the previous table, where $\delta = 0.01$ and $\hat{\beta} = -0.30$. Now, regressing $e_t(-0.30)$ on $-W_t(-0.30)$ and using the 2-decimal accuracy from the table gives a coefficient:

$$\frac{(0.60)(12) + (6.82)(-4) + (6.95)(-6) + (1.91)(-6) + (7.57)(0)}{(12)^2 + (-4)^2 + (-6)^2 + (-6)^2 + (0)} = -0.3157$$

This is an estimate of $\beta_0 - \hat{\beta} = \beta_0 + 0.30$, so you compute an improved estimate of β_0 as $-0.30 - 0.3157 = -0.6157$. This estimated β results in a lower sum of squares:

$$\sum \hat{e}_t^2(-0.6157) = 271.87$$

Using $\hat{\beta} = -0.6157$ as a new initial value, you can again compute an improvement. Continue iterating the estimation improvement technique until the changes $\Delta\hat{\beta}$ become small. For this data set, and changing from $\delta = 0.01$ to $\delta = 0.001$ as is done in PROC ARIMA, you find that $\hat{\beta} = -0.6618$ minimizes the sum of squares at 271.153.

You can extend this method to higher-order and mixed processes. The technique used in PROC ARIMA is more sophisticated than the one here, but it operates under the same principle. The METHOD=ULS technique more accurately computes prediction error variances and finite sample predictions than METHOD=CLS. METHOD=CLS assumes a constant variance and the same linear combination of past values as the optimum prediction. Also, when you specify METHOD=ML, the quantity to be minimized is not the sum of squares. Instead, it is the negative log of the likelihood function. Although CLS, ULS, and ML should give similar results for reasonably large data sets, studies comparing the three methods indicate that ML is the most accurate. Initial values are computed from the Yule-Walker equations for the first round of the iterative procedure as in the previous example. See also **section 2.2.1**.

3.4.4 ESTIMATE Statement for Series 8-A

Finally, reexamine the generated series Y8,

$$Y_t - 0.6Y_{t-1} = e_t + 0.4e_{t-1}$$

Overfitting on both sides of a series generally leads to unstable estimates and often, but not always, to failures to converge. For series 8, an ARMA(2,2) model converges even though the series is ARMA(1,1). Convergence is not, in general, expected when overfitting on both sides like this. An ARMA(4,4) does fail to converge for series 8. With real data, the model is unknown, but when an ARMA(p,q) model fit is attempted and nonconvergence results, one possible cause is overfitting on both sides. Try starting with only the autoregressive side, and then add moving average terms one at a time.

To illustrate nonconvergence with less extreme overfitting than ARMA(4,4), a second series, series 8-A, is generated from the same ARMA(1,1) model as for series 8. An ARMA(2,2) is fit to this ARMA(1,1) data, resulting in **Output 3.10**. Because the data are generated from a known model, you know the model overfits the data.

The following statements produce **Output 3.10**:

```
proc arima data=newarma11;
  identify var=y8 noprint;
  estimate p=1 q=1 printall grid;
  estimate p=2 q=2;
quit;
```

The PRINTALL option shows the iterations. Because the iterations stop when the changes in parameter estimates are small, you have no guarantee that the final parameter estimates have minimized the residual sum of squares (or maximized the likelihood). To check this, use the GRID option to evaluate the sum of squares (or likelihood) on a grid surrounding the final parameter estimates.

Output 3.10: Using the ESTIMATE Statement for Series 8-A: PROC ARIMA

The ARIMA Procedure

Preliminary Estimation

Initial Autoregressive Estimates	
	Estimate
1	0.54140

Initial Moving Average Estimates	
	Estimate
1	-0.31029

Constant Term Estimate	0.093084
White Noise Variance Est	1.033422

Conditional Least Squares Estimation							
Iteration	SSE	MU	MA1,1	AR1,1	Constant	Lambda	R Crit
0	146.72	0.20298	-0.31029	0.54140	0.093084	0.00001	1
1	145.14	0.15545	-0.41396	0.53088	0.072923	1E-6	0.098128
2	145.13	0.15086	-0.42211	0.52918	0.071026	1E-7	0.007949
3	145.13	0.15085	-0.42358	0.52834	0.071115	1E-8	0.00125
4	145.13	0.15089	-0.42386	0.52817	0.071194	1E-9	0.000239

ARIMA Estimation Optimization Summary	
Estimation Method	Conditional Least Squares
Parameters Estimated	3
Termination Criteria	Maximum Relative Change in Estimates
Iteration Stopping Value	0.001
Criteria Value	0.000661
Alternate Criteria	Relative Change in Objective Function
Alternate Criteria Value	1.073E-7
Maximum Absolute Value of Gradient	0.029539
R-Square Change from Last Iteration	0.000239
Objective Function	Sum of Squared Residuals
Objective Function Value	145.1314
Marquardt's Lambda Coefficient	1E-9
Numerical Derivative Perturbation Delta	0.001
Iterations	4

Conditional Least Squares Estimation					
Parameter	Estimate	Standard Error	t Value	Approx Pr > t	Lag
MU	0.15089	0.23900	0.63	0.5288	0
MA1,1	-0.42386	0.09685	-4.38	<.0001	1
AR1,1	0.52817	0.09216	5.73	<.0001	1

Constant Estimate	0.071194
Variance Estimate	0.987289
Std Error Estimate	0.993624
AIC	426.7322
SBC	435.7642
Number of Residuals	150

* AIC and SBC do not include log determinant.

Correlations of Parameter Estimates			
Parameter	MU	MA1,1	AR1,1
MU	1.000	-0.020	-0.045
MA1,1	-0.020	1.000	0.634
AR1,1	-0.045	0.634	1.000

Autocorrelation Check of Residuals									
To Lag	Chi-Square	DF	Pr > ChiSq	Autocorrelations					
6	1.53	4	0.8216	-0.008	0.015	0.046	-0.086	-0.001	0.006
12	2.47	10	0.9912	0.005	-0.004	-0.002	-0.041	0.026	-0.058
18	7.87	16	0.9527	0.070	0.064	0.031	-0.102	-0.106	-0.015
24	15.96	22	0.8178	-0.102	0.038	-0.060	-0.168	0.036	0.027
30	24.99	28	0.6282	0.037	0.049	-0.126	0.054	0.022	-0.158

SSE Surface on Grid Near Estimates: MA1,1 (Y8A) ①			
MU (Y8A)	-0.42886	-0.42386	-0.41886
0.14589	145.14	145.13	145.14
0.15089	145.14	145.13	145.14
0.15589	145.14	145.13	145.14

SSE Surface on Grid Near Estimates: AR1,1 (Y8A) ②			
MU (Y8A)	0.52317	0.52817	0.53317
0.14589	145.14	145.13	145.14
0.15089	145.14	145.13	145.14
0.15589	145.14	145.13	145.14

SSE Surface on Grid Near Estimates: AR1,1 (Y8A) ③			
MA1,1 (Y8A)	0.52317	0.52817	0.53317
-0.42886	145.13	145.14	145.15
-0.42386	145.14	145.13	145.14
-0.41886	145.15	145.14	145.13

Model for variable Y8A	
Estimated Mean	0.15089

Autoregressive Factors	
Factor 1:	1 - 0.52817 B**(1)

Moving Average Factors	
Factor 1:	1 + 0.42386 B**(1)

Preliminary Estimation

Initial Autoregressive Estimates	
	Estimate
1	0.62234
2	-0.16633

Initial Moving Average Estimates	
	Estimate
1	-0.27357
2	-0.10806

Constant Term Estimate	0.110416
White Noise Variance Est	1.020326

Conditional Least Squares Estimation									
Iteration	SSE	MU	MA1,1	MA1,2	AR1,1	AR1,2	Constant	Lambda	R Crit
0	148.53	0.20298	-0.27357	-0.10806	0.62234	-0.16633	0.110416	0.00001	1
1	146.71	0.19160	-0.06149	0.01487	0.84488	-0.24439	0.076546	0.00001	0.174239
2	145.86	0.18266	0.14008	0.12380	1.05407	-0.33017	0.050432	0.00001	0.120744
3	145.51	0.17579	0.25644	0.19203	1.17726	-0.38160	0.03592	0.00001	0.075601
4	145.32	0.15472	0.53022	0.36998	1.47543	-0.50649	0.004806	1E-6	0.052896
5	145.06	0.14167	-0.51699	-0.03549	0.42463	0.07304	0.071165	1E-6	0.143854
6	144.79	0.14228	-1.10825	-0.26981	-0.16623	0.39478	0.109763	1E-6	0.095905
7	144.15	0.14394	-1.30881	-0.34367	-0.36509	0.50619	0.123632	1E-6	0.066706
8	143.62	0.14664	-1.34628	-0.35860	-0.39938	0.52432	0.128319	1E-6	0.062488
9	143.50	0.15104	-1.36216	-0.36775	-0.41065	0.52617	0.133592	1E-6	0.057159
10	143.33	0.16809	-1.37677	-0.38586	-0.40605	0.50265	0.151853	0.001	0.067524
11	143.33	0.17137	-1.37722	-0.38561	-0.40743	0.50379	0.154852	0.1	0.03361
12	143.33	0.17324	-1.37744	-0.38570	-0.40809	0.50425	0.156583	0.1	0.036957
13	143.33	0.17324	-1.37744	-0.38570	-0.40809	0.50425	0.156583	1E12	0.037211

Warning: Estimates did not improve after a ridge was encountered in the objective function. The iteration process has been terminated.

ARIMA Estimation Optimization Summary	
Estimation Method	Conditional Least Squares
Parameters Estimated	5
Termination Criteria	Maximum Relative Change in Estimates
Iteration Stopping Value	0.001
Criteria Value	9.87E-14
Maximum Absolute Value of Gradient	1.513393
R-Square Change from Last Iteration	0.037211
Objective Function	Sum of Squared Residuals
Objective Function Value	143.3278

ARIMA Estimation Optimization Summary	
Marquardt's Lambda Coefficient	1E12
Numerical Derivative Perturbation Delta	0.001
Iterations	13
Warning Message	Estimates may not have converged.

Conditional Least Squares Estimation					
Parameter	Estimate	Standard Error	t Value	Approx Pr > t	Lag
MU	0.17324	0.24140	0.72	0.4741	0
MA1,1	-1.37744	0.17371	-7.93	<.0001	1
MA1,2	-0.38570	0.13472	-2.86	0.0048	2
AR1,1	-0.40809	0.16404	-2.49	0.0140	1
AR1,2	0.50425	0.08255	6.11	<.0001	2

Constant Estimate	0.156583
Variance Estimate	0.988468
Std Error Estimate	0.994217
AIC	428.8564
SBC	443.9096
Number of Residuals	150

* AIC and SBC do not include log determinant.

Correlations of Parameter Estimates					
Parameter	MU	MA1,1	MA1,2	AR1,1	AR1,2
MU	1.000	-0.026	-0.022	-0.033	-0.017
MA1,1	-0.026	1.000	0.994	0.890	-0.065
MA1,2	-0.022	0.994	1.000	0.874	-0.015
AR1,1	-0.033	0.890	0.874	1.000	0.223
AR1,2	-0.017	-0.065	-0.015	0.223	1.000

Autocorrelation Check of Residuals									
To Lag	Chi-Square	DF	Pr > ChiSq	Autocorrelations					
6	0.97	2	0.6169	-0.011	0.029	0.017	-0.058	-0.028	0.028
12	1.58	8	0.9914	-0.015	0.013	-0.019	-0.026	0.011	-0.047
18	6.78	14	0.9427	0.060	0.071	0.024	-0.096	-0.110	-0.016
24	14.50	20	0.8043	-0.100	0.041	-0.064	-0.161	0.032	0.029
30	22.76	26	0.6466	0.042	0.040	-0.119	0.047	0.023	-0.154

Model for variable Y8A	
Estimated Mean	0.173242

Autoregressive Factors	
Factor 1:	1 + 0.40809 B**(1) - 0.50425 B**(2)

Moving Average Factors
Factor 1: $1 + 1.37744 B^{**}(1) + 0.3857 B^{**}(2)$

Examine the grids ❶ ❷ ❸ in **Output 3.10**, and verify that the middle sum of squares, 145.13, is the smallest of the nine tabulated values. For example, increasing the AR estimate 0.52817 to 0.53317 and decreasing the MA estimate -0.42386 to -0.42886 increases the sum of squares from 145.13 to 145.15.

As expected, the correct model fits well. Turning to the overfitted model, there are several indications of problems. To understand the failure to converge, note that

$$Y_t - 0.6Y_{t-1} = e_t + 0.4e_{t-1}$$

implies the following:

$$Y_{t-1} - 0.6Y_{t-2} = e_{t-1} + 0.4e_{t-2}$$

Now, multiply this last equation on both sides by φ , and add to the first equation, obtaining the following:

$$Y_t + (\varphi - 0.6)Y_{t-1} - 0.6\varphi Y_{t-2} = e_t + (\varphi + 0.4)e_{t-1} + 0.4\varphi e_{t-2}$$

Every φ yields a different ARMA(2,2), each equivalent to the original Y8. Thus, the procedure could not find one ARMA(2,2) model that seemed best. Although you sometimes overfit and test coefficients for significance to select a model (as illustrated with the iron and steel data), the previous example shows that this method fails when you overfit on both sides of the ARMA equation at once. Notice that $(1 + 0.4081B - 0.5043B^2)(y_t - \mu) = (1 + 1.3774B + 0.3857B^2)e_t$ is about the same as $(1 + 0.948B)(1 - 0.5361B)(Y_t - \mu) = (1 + 0.948B)(1 + 0.4187B)e_t$. Eliminating the common factor results in an ARMA(1,1), namely $(1 - 0.5361B)(Y_t - \mu) = (1 + 0.4187B)e_t$.

3.4.5 Nonstationary Series

The theory behind PROC ARIMA requires that a series be stationary. Theoretically, the stationarity of a series

$$(1 - \alpha_1 B - \alpha_2 B^2 - \cdots - \alpha_p B^p)(Y_t - \mu) = (1 - \beta_1 B - \beta_2 B^2 - \cdots - \beta_q B^q)e_t$$

hinges on the solutions M of the characteristic equation:

$$1 - \alpha_1 M - \alpha_2 M^2 - \cdots - \alpha_p M^p = 0$$

If all M s that satisfy this equation have $|M| > 1$, then the series is stationary. For example, the following series is stationary:

$$(1 - 1.5B + 0.64B^2)(Y_t - \mu) = (1 + 0.8B)e_t$$

But the following series is not:

$$(1 - 1.5B + 0.5B^2)(Y_t - \mu) = (1 + 0.8B)e_t$$

For the nonstationary example, the characteristic equation is this:

$$1 - 1.5M + 0.5M^2 = 0$$

The solutions are $M = 1$ and $M = 2$. These solutions are called *roots of the characteristic polynomial*, and because one of them is 1, the series is nonstationary. This unit root nonstationarity has several implications, which are explored. The overfit example at the end of the previous section ended when the common factor $(1 - \varphi B)$ neared $(1 - B)$, an unstable value.

First, expanding the model gives the following expression:

$$Y_t - 1.5Y_{t-1} + 0.5Y_{t-2} + (1 - 1.5 + 0.5)\mu = e_t + 0.8e_{t-1}$$

It shows that μ drops out of the equation. As a result, series forecasts do not tend to return to the historic series mean. This is in contrast to stationary series, where μ is estimated and where forecasts always approach this estimated mean.

In the nonstationary example, Y_t is the series level, and:

$$W_t = Y_t - Y_{t-1}$$

This W_t is the first difference or change in the series. By substitution:

$$W_t - 0.5W_{t-1} = e_t + 0.8e_{t-1}$$

When the levels Y_t satisfy an equation with a single unit root nonstationarity, the first differences W_t satisfy a stationary equation, often with mean 0. Similarly, you can eliminate a double unit root as in

$$(1 - 2B + B^2)(Y_t - \mu) = e_t + 0.8e_{t-1}$$

by computing and then analyzing the second difference, which is the following:

$$W_t - W_{t-1} = Y_t - 2Y_{t-1} + Y_{t-2}$$

The first and second differences are often written ∇Y_t and $\nabla^2 Y_t$. For nonseasonal data, you rarely difference more than twice.

Because you do not know the model, how do you know when to difference? You decide by examining the ACF or performing a test as in [section 3.4.8](#). If the ACF dies off very slowly, a unit root is indicated. The slow dying-off might occur after one or two substantial drops in the ACF. The sequence 1, 0.50, 0.48, 0.49, 0.45, 0.51, 0.47, ..., is considered to die off slowly in this context, even though the initial drop from 1 to 0.5 is large and the magnitude of each autocorrelation after the first is not near 1.

Using the IDENTIFY statement, you can accomplish differencing easily. The following statement produces the correlation function for W_t :

```
identify var=y(1);
```

In this function, $W_t = Y_t - Y_{t-1}$.

A subsequent ESTIMATE statement operates on W_t , so the NOCONSTANT option is normally used. The following statement specifies analysis of the second difference:

```
identify var=y(1,1);
```

This can be written as follows:

$$(Y_t - Y_{t-1}) - (Y_{t-1} - Y_{t-2})$$

The default is no differencing for the variables. Assuming a nonzero mean in the differenced data is equivalent to assuming a deterministic trend in the original data because $(\alpha + \beta t) - (\alpha + \beta(t-1)) = \beta$. You can fit this β easily by omitting the NOCONSTANT option.

3.4.6 Effect of Differencing on Forecasts

PROC ARIMA provides forecasts and 95% upper and lower confidence bounds for predictions for the general ARIMA model. If you specify differencing, modeling is done on the differenced series, but predictions are given for the original series levels. Also, when you specify a model with differencing, prediction error variances increase without bound as you predict further into the future.

In general, by using estimated parameters and by estimating σ^2 from the model residuals, you can easily derive the forecasts and their variances from the model. PROC ARIMA accomplishes this task for you automatically.

For example, in the model $Y_t - 1.5Y_{t-1} + 0.5Y_{t-2} = e_t$, note that $(Y_t - Y_{t-1}) - 0.5(Y_{t-1} - Y_{t-2})e_t$. Thus, the first differences

$$W_t = Y_t - Y_{t-1}$$

are stationary. Given data Y_1, Y_2, \dots, Y_n from this series, you predict future values by first predicting future values of W_{n+j} , using $(0.5)^j W_n$ as the prediction:

$$Y_{n+j} - Y_n = W_{n+1} + W_{n+2} + \dots + W_{n+j}$$

So the forecast of Y_{n+j} is expressed as follows:

$$Y_n + \sum_{i=1}^j (0.5)^i W_n$$

To illustrate further, the following computation of forecasts shows a few values of Y_t , W_t , and predictions \hat{Y}_j :

Actual			
t	98	99	100(n)
Y_t	475	518	550
W_t	28	43	32

Forecast				
t	101	102	103	104
Y_t	566	574	578	580
W_t	16	8	4	2

Note that the following expression approaches 1 as j increases:

$$\sum_{i=1}^j (0.5)^i$$

So the forecasts converge to $550 + (1)(32) = 582$,

Forecast errors can be computed from the forecast errors of the W_t s. For example,

$$Y_{n+2} = Y_n + W_{n+1} + W_{n+2}$$

and

$$\hat{Y}_{n+2} = Y_n + .5W_n + .25W_n$$

Rewriting $Y_{n+2} = Y_n + (.5W_n + e_{n+1}) + (.25W_n + .5e_{n+1} + e_{n+2})$ yields the forecast error, with the variance $3.25\sigma^2$ arising from this combined forecast error expression:

$$1.5e_{n+1} + e_{n+2}$$

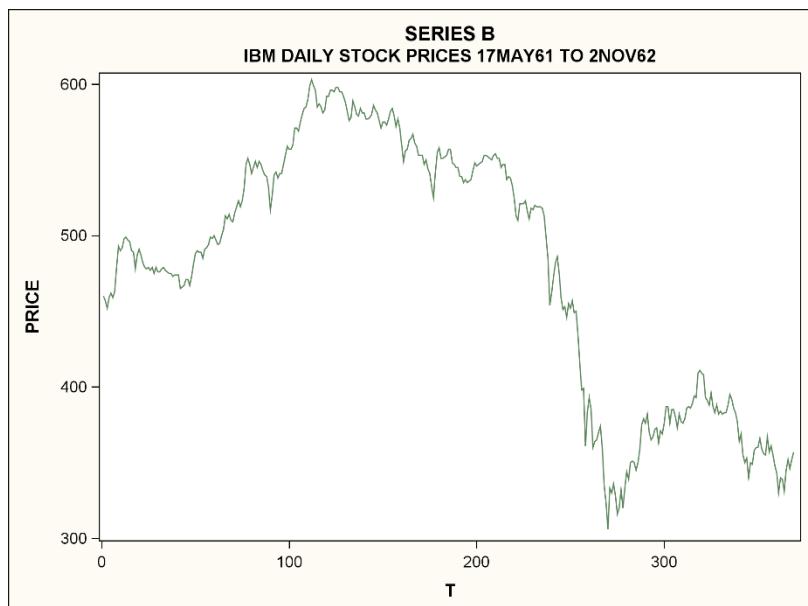
3.4.7 Examples: Forecasting IBM Series and Silver Series

An example that obviously needs differencing is the IBM stock closing price series reported by Box and Jenkins (1976). In this example, the data are analyzed with PROC ARIMA. They are forecast 15 periods ahead. You read in the series and check the ACF:

```
data ibm;
  input price @@;
  t+1;
  datalines;
data lines
;
run;
proc arima data=ibm;
  identify var=price center nlag=15;
  identify var=price(1) nlag=15;
run;
```

The plot of the original data is shown in **Output 3.11**, and the IDENTIFY results are shown in **Output 3.12**.

Output 3.11: Plotting the Original Data

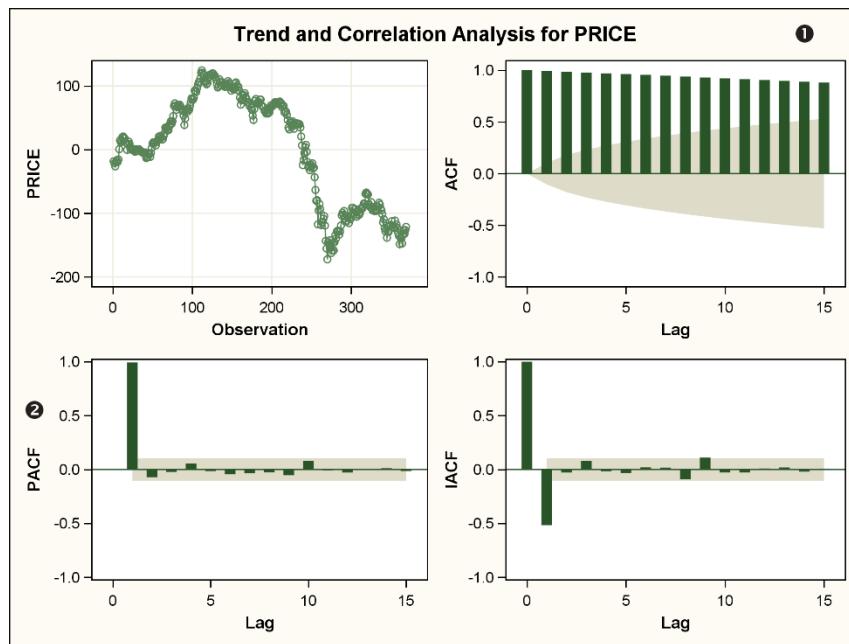


Output 3.12: Identifying the IBM Price Series

The ARIMA Procedure

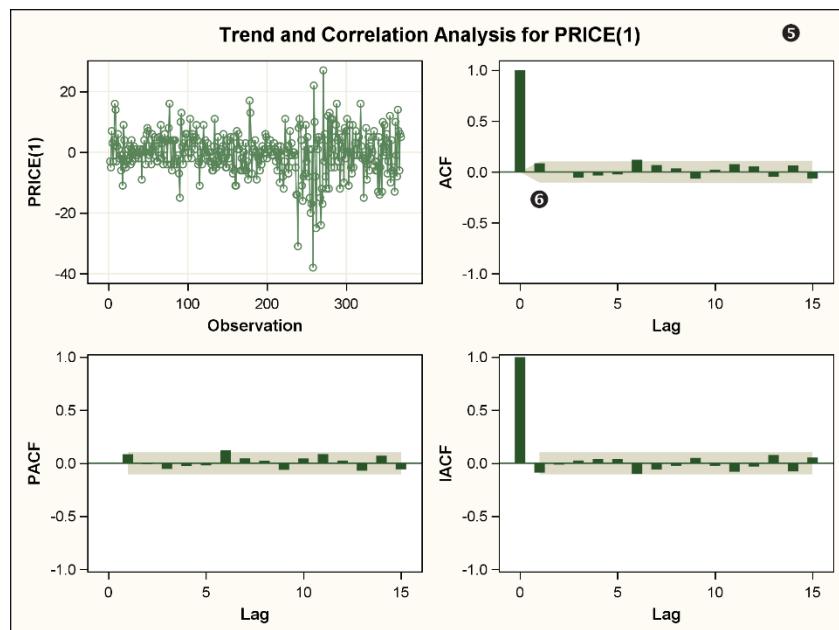
Name of Variable = PRICE	
Mean of Working Series	0
Standard Deviation	84.10504
Number of Observations	369

Autocorrelation Check for White Noise									
To Lag	Chi-Square	DF	Pr > ChiSq	Autocorrelations					
6	2135.31	6	<.0001	0.993	0.986	0.978	0.971	0.964	0.956
12	4097.40	12	<.0001	0.948	0.939	0.930	0.922	0.914	0.906



Name of Variable = PRICE	
Period(s) of Differencing	1
Mean of Working Series	-0.27989
Standard Deviation	7.248345
Number of Observations	368
Observation(s) eliminated by differencing	1

Autocorrelation Check for White Noise								
To Lag	③Chi-Square	DF	Pr > ChiSq	Autocorrelations				
6	9.98	6	0.1256	0.086	-0.001	-0.054	-0.035	-0.024
12	17.42	12	0.1344	0.068	0.036	-0.066	0.022	0.077



The ACF ❶ dies off very slowly. The PACF ❷ indicates a very high coefficient, 0.99340, in the regression of Y_t on Y_{t-1} . The ACF of the differenced series ❸ looks like white noise. In fact, the Q statistics 9.98 and 17.42 ❹ are not significant. For example, the probability of a value larger than 9.98 in a χ^2_6 distribution is 0.126, so 9.98 is to the left of the critical value and, therefore, is not significant. The Q statistics are computed with the first six (9.98) and first 12 (17.42) autocorrelations of the differenced series. With a first difference, it is common to find an indication of a lag 1 MA term. The first autocorrelation is 0.08558 ❺ with a standard error of about $1/(368)^{1/2} = 0.052$.

Next, suppress the printout with the IDENTIFY statement (you have already looked at it, but you still want PROC ARIMA to compute initial estimates), and estimate the model:

```
proc arima data=ibm;
  identify var=price(1) noprint;
  estimate q=1 noconstant;
run;
```

The results are shown in **Output 3.13**.

Output 3.13: Analyzing Daily Series with the ESTIMATE Statement: PROC ARIMA

The ARIMA Procedure

Conditional Least Squares Estimation					
Parameter	Estimate	Standard Error	t Value	Approx Pr > t	Lag
MA1,1	-0.08658	0.05203	-1.66	0.0970	1

Variance Estimate	52.36132
Std Error Estimate	7.236112
AIC	2501.943
SBC	2505.851
Number of Residuals	368

* AIC and SBC do not include log determinant.

Autocorrelation Check of Residuals										
To Lag	Chi-Square	DF	Pr > ChiSq	Autocorrelations						
6	6.99	5	0.2217	0.001	0.005	-0.051	-0.026	-0.030	0.120	
12	13.94	11	0.2365	0.056	0.039	-0.070	0.024	0.072	0.054	
❸ 18	31.04	17	0.0198	-0.057	0.079	-0.081	0.118	0.113	0.040	
24	39.05	23	0.0196	0.041	0.072	-0.089	-0.027	0.066	0.025	
30	49.83	29	0.0094	0.028	-0.100	-0.055	0.051	0.028	0.099	
36	56.47	35	0.0122	0.072	-0.074	-0.063	-0.007	0.022	0.035	
42	64.42	41	0.0112	0.066	-0.085	0.059	-0.060	0.018	0.017	
48	76.33	47	0.0044	-0.116	-0.037	0.073	0.005	0.069	0.057	

Model for variable PRICE	
Period(s) of Differencing	1

No mean term in this model.

Moving Average Factors	
Factor 1:	1 + 0.08658 B**(1)

Although the evidence is not strong enough to indicate that the series has a nonzero first-order autocorrelation, you nevertheless fit the MA(1) model. The t statistic -1.66 ⑦ is significant at the 10% level.

More attention should be paid to the lower-order and seasonal autocorrelations than to the other autocorrelations. In this example, you ignore an autocorrelation 0.121 ④ (found in **Output 3.12**) at lag 6 that was even bigger than the lag 1 autocorrelation. Similarly, residuals from the final fitted model show a Q statistic 31.04 ⑧ that attains significance because of autocorrelations 0.118 and 0.113 at lags 16 and 17. Ignore this significance in favor of the more parsimonious MA(1) model.

The model appears to fit. Therefore, make a third run to forecast:

```
proc arima data=ibm;
  identify var=price(1) noint;
  estimate q=1 noconstant noint;
  forecast lead=15;
run;
```

See the forecasts in **Output 3.14**.

Output 3.14: Forecasting Daily Series: PROC ARIMA

The ARIMA Procedure

Model for variable PRICE	
Period(s) of Differencing	1

No mean term in this model.

Moving Average Factors	
Factor 1:	1 + 0.08658 B**(1)

Forecasts for variable PRICE				
Obs	Forecast	Std Error	95% Confidence Limits	
370	357.3837	7.2361	343.2012	371.5662
371	357.3837	10.6856	336.4403	378.3270
372	357.3837	13.2666	331.3817	383.3857

(Additional output omitted)

382	357.3837	28.1816	302.1487	412.6187
383	357.3837	29.2579	300.0392	414.7282
384	357.3837	30.2960	298.0047	416.7627

If $Y_t - Y_{t-1} = e_t - \beta e_{t-1}$, as in the IBM example, then, by repeated back substitution, you have the following:

$$e_t = (Y_t - Y_{t-1}) + \beta(Y_{t-1} - Y_{t-2}) + \beta^2(Y_{t-2} - Y_{t-3}) + \dots$$

or

$$Y_t = e_t + (1 - \beta)(Y_{t-1} + \beta Y_{t-2} + \beta^2 Y_{t-3} + \dots)$$

This means that $\hat{Y}_t = (1 - \beta)(Y_{t-1} + \beta Y_{t-2} + \beta^2 Y_{t-3} + \dots)$.

Forecasting Y_t by such an exponentially weighted sum of past Y s is called *single exponential smoothing*. Higher degrees of differencing plus the inclusion of more MA terms is equivalent to higher-order exponential smoothing. However, PROC ARIMA, unlike PROC FORECAST with METHOD=EXPO, estimates the parameters from the data.

Dickey and Fuller (1979) give a formal test of the null hypothesis that an AR series has a unit root nonstationarity versus the alternative that it is stationary. Said and Dickey (1984) extend the test to ARIMA models. The test involves a regression of ∇Y_t (where $\nabla Y_t = Y_t - Y_{t-1}$) on $Y_{t-1} - \bar{Y}, \nabla Y_{t-1}, \dots, \nabla Y_{t-p}$. Here, p is at least as large as the order of the AR process. Or, in the case of the mixed process, it is large enough to give a good approximation to the model. The t test on $Y_{t-1} - \bar{Y}$ is called τ_μ because it does not have a Student's t distribution and must be compared to tables provided by Fuller (1996, p. 642). The silver series from Chapter 2, "Simple Models: Autoregression," is used as an illustration in the next section.

3.4.8 Models for Nonstationary Data

You can formally test for unit root nonstationarity with careful modeling and special distributions. Any autoregressive model such as the AR(2) model $Y_t - \mu = \alpha_1(Y_{t-1} - \mu) + \alpha_2(Y_{t-2} - \mu) + e_t$ can be written in terms of differences and the lagged level term ($Y_t - \mu$). With a little algebra, the AR(2) becomes:

$$Y_t - Y_{t-1} = -(1 - \alpha_1 - \alpha_2)(Y_{t-1} - \mu) - \alpha_2(Y_{t-1} - Y_{t-2}) + e_t$$

Stationarity depends on the roots of the characteristic equation $1 - \alpha_1 M - \alpha_2 M^2 = 0$, so if $M = 1$ is a root, then $(1 - \alpha_1 - \alpha_2) = 0$. In that case, the $(Y_{t-1} - \mu)$ term drops out of the model and forecasts do not revert to the mean. This discussion suggests a least squares regression of $Y_t - Y_{t-1}$ on Y_{t-1} and $(Y_{t-1} - Y_{t-2})$ with an intercept. It also suggests the use of the resulting coefficient or t test on the Y_{t-1} term as a test of the null hypothesis that the series has a unit root nonstationarity. If all roots M exceed 1 in magnitude, the coefficient of $(Y_{t-1} - \mu)$ will be negative, suggesting a one-tailed test to the left if stationarity is the alternative. However, there is one major problem with this idea: neither the estimated coefficient of $(Y_{t-1} - \mu)$ nor its t test has a standard distribution, even when the sample size becomes very large. This does not mean the test cannot be done, but it does require the tabulation of a new distribution for the test statistics.

Dickey and Fuller (1979, 1981) studied the distributions of estimators and t statistics in autoregressive models with unit roots. The leftmost columns of the following tables show the regressions that they studied. Here, $Y_t - Y_{t-1} = \nabla Y_t$ denotes a first difference.

Regress ∇Y_t on these	To yield AR(1) in deviations form
$Y_{t-1}, \nabla Y_{t-1} \dots \nabla Y_{t-k}$	$Y_t = \rho Y_{t-1} + e_t$
$Y_{t-1}, 1, \nabla Y_{t-1} \dots \nabla Y_{t-k}$	$Y_t - \mu = \rho(Y_{t-1} - \mu) + e_t$
$Y_{t-1}, 1, t, \nabla Y_{t-1} \dots \nabla Y_{t-k}$	$Y_t - \alpha - \beta t = \rho(Y_{t-1} - \alpha - \beta(t-1)) + e_t$

AR(1) in regression form	$H_0: \rho = 1$
$\nabla Y_t = (\rho - 1)Y_{t-1} + e_t$	$\nabla Y_t = e_t$
$\nabla Y_t = (1 - \rho)\mu + (\rho - 1)Y_{t-1} + e_t$	$\nabla Y_t = e_t$
$\nabla Y_t = (1 - \rho)(\alpha + \beta t) + \rho\beta + (\rho - 1)Y_{t-1} + e_t$	$\nabla Y_t = \beta + e_t$

The lagged differences are referred to as *augmenting lags* and the tests as *Augmented Dickey-Fuller* or *ADF* tests. The three regression models allow for three types of trends. For illustration, a lag 1 autoregressive model with autoregressive parameter ρ is shown in the preceding table, both in deviations form and in the algebraically equivalent regression form. The deviations form is most instructive. It shows that if $|\rho| < 1$ and if you have appropriate starting values, then the expected value of Y_t is 0, μ , or $\alpha + \beta t$, depending on which model is assumed.

Fit the first model only if you know the mean of your data is 0 (for example, Y_t might already be a difference of some observed variable). Use the third model if you suspect a regular trend up or down in your data. If you fit the third model when β is really 0, your tests will be valid, but not as powerful as those from the second model. The parameter β represents a trend slope when $|\rho| < 1$. It is called a *drift* when $\rho = 1$. For known parameters and n data points, the forecast of Y_{n+L} would be $\alpha + \beta(n + L) + \rho^L(Y_n - \alpha - \beta n)$ for $|\rho| < 1$ with forecast error variance $(1 + \rho^2 + \dots + \rho^{2L-2})\sigma^2$. As L increases, the forecast error variance approaches $\sigma^2/(1 - \rho^2)$, the variance of Y around the trend. However, if $\rho = 1$, the L

step ahead forecast is $Y_t + \beta L$ with forecast error variance $L\sigma^2$, so that the error variance increases without bound in this case. In both cases, the forecasts have a component that increases at the linear rate β .

For the regression under discussion, the distributions for the coefficients of Y_{t-1} , 1, and t are all nonstandard. Tables of critical values and discussion of the theory are given in Fuller (1996). One very nice feature of these regressions is that the coefficients of the lagged differences ∇Y_{t-j} have normal distributions in the limit. Thus, a standard F test to see whether a set of these lagged differences can be omitted is justified in large samples, as are the t statistics for the individual lagged difference coefficients. They converge to standard normal distributions. The coefficients of Y_{t-1} and the associated t tests have distributions that differ among the three regressions and are nonstandard. Fortunately, the t test statistics have the same limit distributions no matter how many augmenting lags are used.

For example, stocks of silver on the New York Mercantile Exchange are analyzed in Chapter 2 of this book. The data are re-analyzed here using DEL to denote the difference, DEL_i for its i th lag, and LSILVER for the lagged level of silver. The where part=1 statement restricts analysis to the data used in the first edition of this book.

```
proc reg data=silver;
  model del=lsilver dell1 dell2 dell3 dell4 /noprint;
  test dell2=0, dell3=0, dell4=0;
  where part=1;
run;
proc reg data=silver;
  model del=lsilver dell1;
  where part=1;
run;
```

The result of the TEST statement for the model with four augmenting lags is in **Output 3.15**.

Output 3.15: Test of Augmenting Lags

Test 1 Results for Dependent Variable DEL				
Source	DF	Mean Square	F Value	Pr > F
Numerator	3	1152.19711	1.32	0.2803
Denominator	41	871.51780		

Because this test involves only the lagged differences, the F distribution is justified in large samples. Although the sample size is not particularly large, the p -value 0.2803 is not even close to 0.05, providing no evidence against leaving out all but the first augmenting lag. The second PROC REG produces **Output 3.16**.

Output 3.16: PROC REG on Silver Data

Parameter Estimates					
Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > t
Intercept	1	75.58073	27.36395	2.76	0.0082
LSILVER	1	-0.11703	0.04216	-2.78	0.0079
DEL1	1	0.67115	0.10806	6.21	<.0001

Because the printed p -value 0.0079 is less than 0.05, the uninformed user might conclude that there is strong evidence against a unit root in favor of stationarity. This is an error because all p -values from PROC REG are computed from the t distribution. Under the null hypothesis of a unit root, this statistic has the distribution tabulated by Dickey and Fuller. The appropriate 5% left tail critical value of the limit distribution is -2.86 (Fuller 1996, p. 642), so the statistic is not far enough below 0 to reject the unit root null hypothesis. Nonstationarity (i.e., a unit root) cannot be rejected. This test is also available in PROC ARIMA starting with Version 6 of SAS and can be obtained as follows:

```
proc arima data=silver;
  i var = silver stationarity=(adf=(1)) outcov=adf;
run;
```

Output 3.17 contains several tests.

Output 3.17: Unit Root Tests on Silver Data

Augmented Dickey-Fuller Unit Root Tests							
Type	Lags	Rho	Pr < Rho	Tau	Pr < Tau	F	Pr > F
Zero Mean	1	-0.2461	0.6232	-0.28	0.5800		
Single Mean	1	-17.7945	0.0121	-2.78	0.0689	3.86	0.1197
Trend	1	-15.1102	0.1383	-2.63	0.2697	4.29	0.3484

Every observed data point exceeds 400, so any test from a model that assumes a 0 mean can be ignored. Also, the PROC REG output strongly indicated that one lagged difference was required. Thus, the tests with no lagged differences can be ignored and are not requested here. The output shows coefficient (or normalized bias) unit root tests that would be computed as $n(\hat{\rho} - 1)$ in an AR(1) model with coefficient ρ . For the AR(2) model with roots ρ and m , the regression model form

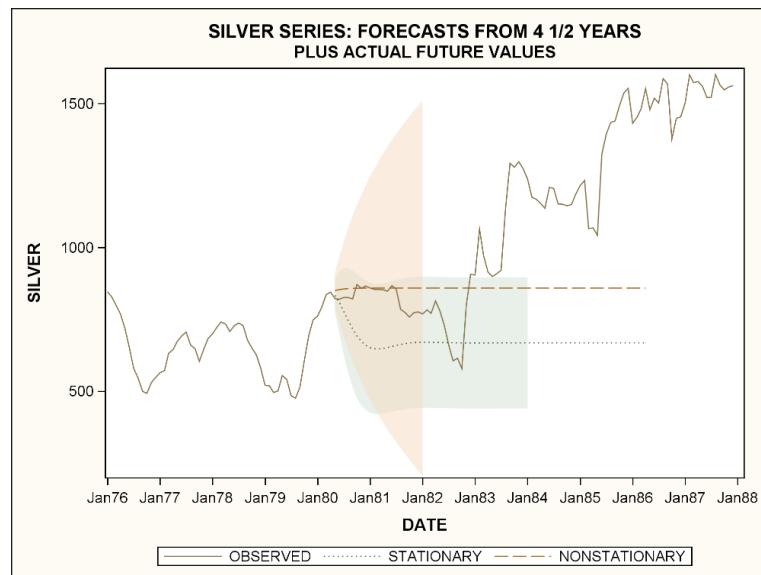
$$Y_t - Y_{t-1} = -(1 - \alpha_1 - \alpha_2)(Y_{t-1} - \mu) - \alpha_2(Y_{t-1} - Y_{t-2}) + e_t$$

becomes the following:

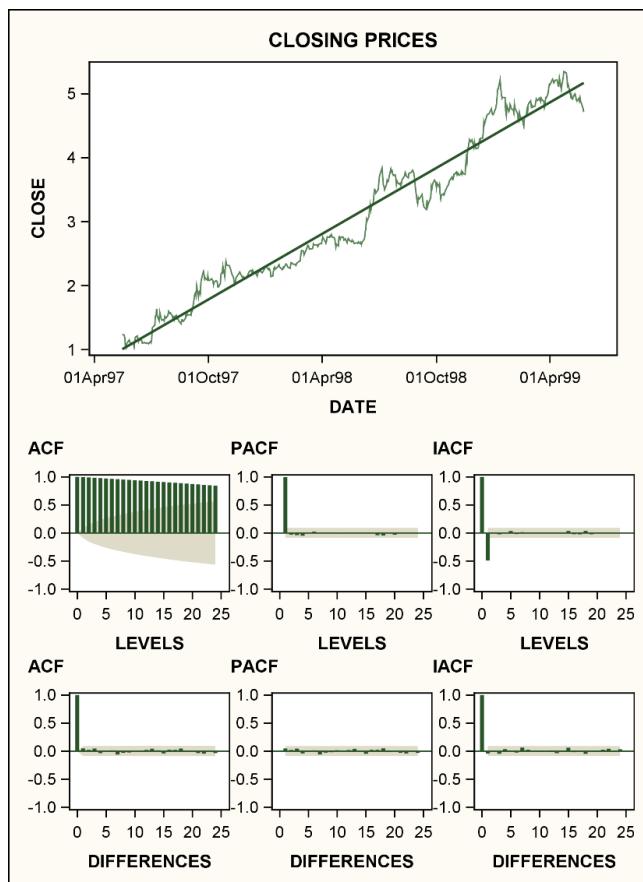
$$Y_t - Y_{t-1} = -(1 - m)(1 - \rho)(Y_{t-1} - \mu) + m\rho(Y_{t-1} - Y_{t-2}) + e_t$$

So the coefficient of Y_{t-1} is $-(1 - \rho)(1 - m)$ in terms of the roots. If $\rho = 1$, then the coefficient of $(Y_{t-1} - Y_{t-2})$, which is 0.67115 in the silver example, is an estimate of m . Consequently, it is not surprising that an adjustment using that statistic is required to get a test statistic that behaves like $n(\hat{\rho} - 1)$ under $H_0: \rho = 1$. Specifically, you divide the lag 1 coefficient (-0.11703) by (1 - 0.67115), and then multiply by n . Similar adjustments can be made in higher-order processes. For the silver data, $50(-0.11703)/(1 - 0.67115) = -17.7945$ is shown in the printout and has a p -value (0.0121) less than 0.05. However, based on simulated size and power results (Dickey 1984), the tau tests are preferable to these normalized bias tests. Furthermore, the adjustment for lagged differences is motivated by large sample theory, and $n = 50$ is not particularly large. The associated single mean tau test, -2.78, has a p -value exceeding 0.05. It fails to provide significant evidence at the usual 0.05 level against the unit root null hypothesis. The F type statistics are discussed in Dickey and Fuller (1981). If interest lies only in inference about ρ , there is no advantage to using the F statistics, which include restrictions on the intercept and trend as a part of H_0 . Simulations indicate that the polynomial deterministic trend should have as low a degree as is consistent with the data to get good power. The 50 observations studied so far do not display any noticeable trend, so the model with a constant mean seems reasonable. However, tests based on the model with linear trend would be valid and would guard against any unrecognized linear trend. These tests provide even less evidence against the unit root. In summary, getting a test with validity and good statistical power requires appropriate decisions about the model in terms of lags and trends. This is no surprise, as any statistical hypothesis test requires a realistic model for the data.

The data analyzed here were used in the first edition of this book. Since then, more data on this series have been collected. The full set of data makes it clear that the series is not stationary, which is in agreement with the single mean tau statistic. In **Output 3.18**, the original series of 50 is plotted, along with forecasts and confidence bands from an AR(2), which assumes stationarity in levels (solid lines), and an AR(1) fit to the differenced data (dashed lines). The more recent data are appended to the original 50. For a few months into the forecast, the series stays within the solid line bands. It appears that the analyst who chooses stationarity is the better forecaster. This analyst also has much tighter forecast bands. However, a little further ahead, the observations burst through their bands, never to return. The unit root forecast, although its bands might appear unpleasantly wide, does seem to give a more realistic assessment of the uncertainty inherent in this series.

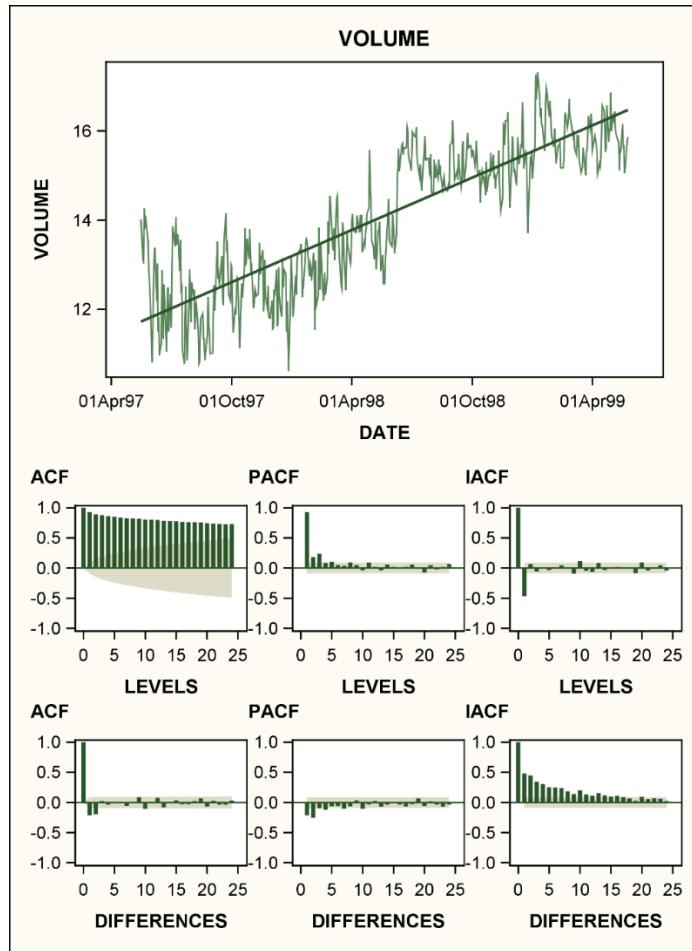
Output 3.18: Silver Series, Stationary and Nonstationary Models

To illustrate the effects of trends, **Output 3.19** shows the logarithm of the closing price of Amazon.com stock. The data were downloaded from the stock reports available through the web search engine Yahoo!. The closing prices are fairly tightly clustered around a linear trend, as displayed in the top part of the figure. The ACF, IACF, and PACF of the series are displayed just below the series plot. Those of the differenced series are displayed just below that. The ACF of the original series dies off very slowly. This could be due to a deterministic trend, a unit root, or both. The three plots along the bottom seem to indicate that differencing has reduced the series to stationarity.

Output 3.19: Amazon Closing Prices

In contrast, **Output 3.20** shows the volume of the same Amazon.com stocks. These, too, show a trend, but notice the IACF of the differenced series. If a series has a unit root on the moving average side, the IACF will die off slowly. This is in line with what you have learned about unit roots on the autoregressive side. For the model $Y_t = e_t - \rho e_{t-1}$, the dual model obtained by switching the backshift operator to the AR side is $(1 - \rho B)Y_t = e_t$. If ρ is (near) 1, you expect the IACF to behave like the ACF of a (near) unit root process—that is, to die off slowly.

Output 3.20: Amazon Volume



This behavior is expected anytime Y_t is the difference of an originally stationary series. Chang and Dickey (1993) give a detailed proof of what happens to the IACF when such overdifferencing occurs. They find that an essentially linear descent in the IACF is consistent with overdifferencing. This can follow an initial drop-off, which happens in the volume data. A linear trend is reduced to a constant by first differencing, so such a trend will not affect the behavior of the IACF of the differenced series. Of course, a linear trend in the data will make the ACF of the levels die off very slowly, as seen in the volume data. The apparent mixed-message-differencing indicated by the levels' ACF and too much differencing indicated by the differences' IACF are not really inconsistent. You just need to think outside the class of ARIMA models to models with time trends and ARIMA errors.

Regression of differences on 1, t , a lagged level, and lagged differences indicated that no lagged differences were needed for the log-transformed closing price series. Two were needed for volume. Using the indicated models, the parameter estimates from PROC REG are shown in **Output 3.21**. PROC REG uses the differenced series as a response, DATE as the time variable, LAGC and LAGV as the lag levels of closing price and volume, respectively, and lagged differences DV1 and DV2 for volume.

Output 3.21: Closing Price and Volume—Unit Root Test

Parameter Estimates						
Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > t	Type I SS
Intercept	1	-2.13939	0.87343	-2.45	0.0146	0.02462
DATE	1	0.00015950	0.00006472	2.46	0.0141	0.00052225
LAGC	1	-0.02910	0.01124	-2.59	0.0099	0.02501

Parameter Estimates						
Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > t	Type I SS
Intercept	1	-17.43463	3.11590	-5.60	<.0001	0.01588
DATE	1	0.00147	0.00025318	5.80	<.0001	0.00349
LAGV	1	-0.22354	0.03499	-6.39	<.0001	25.69204
DV1	1	-0.13996	0.04625	-3.03	0.0026	1.04315
DV2	1	-0.16621	0.04377	-3.80	0.0002	4.16502

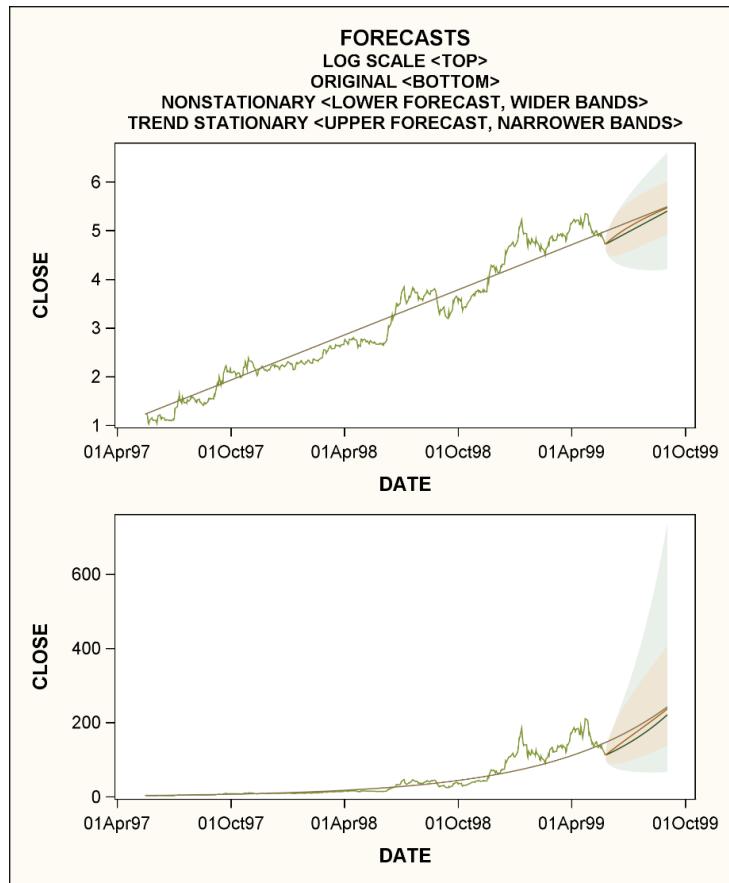
These tests can be automated using the IDENTIFY statement in PROC ARIMA. For these examples, clearly only the linear trend tests are to be considered. Although power is gained by using a lower-order polynomial when it is consistent with the data, the assumption that the trend is simply a constant is clearly inappropriate.

The trend tau statistics, for models with a time trend (see Fuller 1996), are -2.59 (LAGC) for closing price and -6.39 (LAGV) for volume. Using the large n critical values -3.13 at significance level 0.10, -3.41 at 0.05, and -3.96 at 0.01, unit roots are rejected even at the 0.01 level for volume. Thus, the volume series displays stationary fluctuations around a linear trend. Alternatively, the model can be fit in PROC ARIMA and the trend unit root test output consulted to get an approximate p -value.

There is not evidence for stationarity in closing prices even at the 0.10 level, so even though the series seems to hug the linear trend line closely, the deviations cannot be distinguished from a unit root process whose variance grows without bound. An investment strategy based on an assumption of reversion of log-transformed closing prices to the linear trend line does not seem to be supported. That is not to refute the undeniable upward trend in the data—it comes out in the intercept or drift term (estimate 0.0068318) of the model for the differenced series. The model (computations not shown) is $\nabla Y_t = 0.0068318 + e_t + 0.04547e_{t-1}$.

The differences, ∇Y_t , have this positive drift term as their average, so it implies a positive change on average with each passing unit of time. A daily increase of 0.0068318 in the logarithm implies a multiplicative $e^{0.0068318} = 1.00686$ or 0.68% daily increase, which compounds to a $e^{260(0.0068318)} = e^{1.78} = 5.91$, an almost sixfold increase over the approximately 260 trading days in a year. This was a period of phenomenal growth for many such technology stocks, with these data going from about 3.5 to about 120 over two years' time, approximately the predicted 35-fold increase.

The top panel of **Output 3.22** shows closing price forecasts and intervals for the unit root with drift model (forecast rising almost linearly from the last observation and outermost bands) and for a model with stationary residuals from a linear trend (forecast converging to trend line and interior bands) for the log scale data. The following plot, in which each of these has been transformed back to the original scale by exponentiation, deserves some comments. First, note the strong effect of the logarithmic transformation. Any attempt to model on the original scale would have to account for the obviously unequal variation in the data and would require a somewhat complex trend function. Once logs are taken, a rather simple model, random walk with drift, seems to suffice. There is a fairly long string of values starting around January 1999 that are far above the trend curve. Recall that this trend curve is simply an exponentiation of the linear trend on the log scale. As a result, it approximates a median, not a mean. This 50% probability number, the median, might be a more easily understood number for an investment strategist than the mean in a highly skewed distribution such as this. The chosen model, random walk with drift, does not even use this curve. Therefore, a forecast beginning on February 1, 1999, for example, would emanate from the February 1, 1999 data point and follow a path approximately parallel to this trend line. The residuals from this trend line would not represent forecasting errors from either model. Even for the model that assumes stationary but strongly correlated errors, the forecast consists of the trend plus an adjustment based on the error correlation structure.

Output 3.22: Amazon Closing Price (Two Models, Two Scales)

In fact, the plot actually contains forecasts throughout the historic series from both models, but they overlay the data so closely that they are hardly distinguishable from it. The combination of logs and differencing, although it makes the transformed series behave nicely statistically, produces very wide forecast intervals on the original scale. Although this might disappoint the analyst, it might be a reasonable assessment of uncertainty, given that 95% confidence is required and that this is a volatile series.

In summary, ignorance of unit roots and deterministic trends in time series can lead to clearly inappropriate mean-reverting forecasts. Careful modeling of unit roots and deterministic trends can lead to reasonable and informative forecasts. Note that p -values produced under the assumption of stationarity can be misleading when unit roots are in fact present as shown in the silver and stock closing price examples. Both of these show inappropriately small p -values when the p -values are computed from the t rather than from the Dickey-Fuller distributions. In the regression of differences on trend terms, lagged level, and lagged differences, the usual (t and F) distributions are appropriate in large samples for inference on the lagged differences. To get tests with the proper behavior, carefully deciding on the number of lagged differences is important. Hall (1992) studies several methods and finds that overfitting lagged differences, and then testing to leave some out, is a good method. This was illustrated in the silver example and was done for all examples here. Dickey, Bell, and Miller (1986) show that the addition of seasonal dummy variables to a model does not change the large sample (limit) behavior of the unit root tests discussed.

Some practitioners are under the false impression that differencing is justified anytime data appear to have a trend. In fact, such differencing might not be appropriate. This is discussed next.

3.4.9 Differencing to Remove a Linear Trend

Occasionally, practitioners difference data to remove a linear trend. If Y_t has a linear trend $\alpha + \beta t$, then the differenced series $W_t = Y_t - Y_{t-1}$ involves only the constant. For example, suppose $Y_t = \alpha + \beta t + e_t$ where e_t is white noise. Then you have

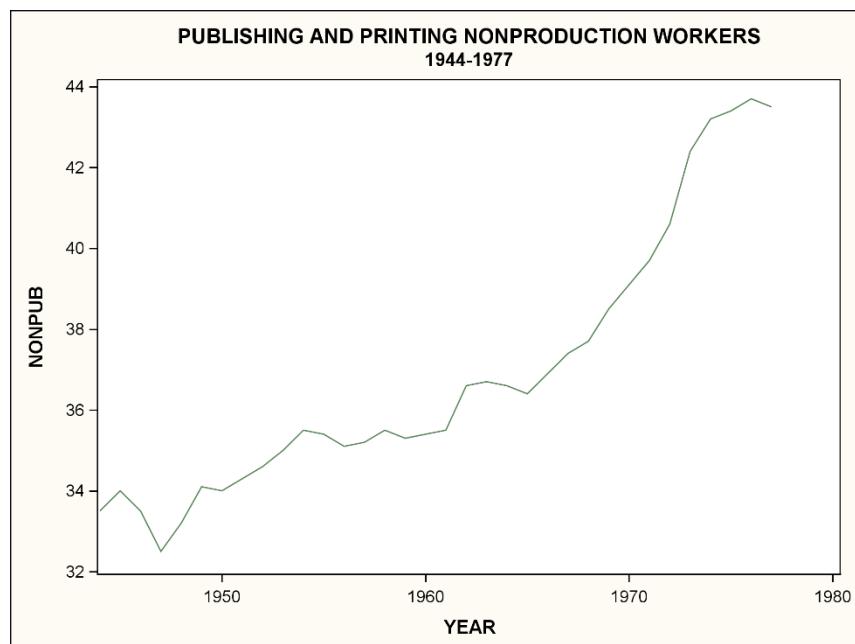
$$W_t = \beta + e_t - e_{t-1}$$

which does not have a trend but, unfortunately, is a noninvertible moving average. Thus, the data have been overdifferenced. Now, the IACF of W looks like the ACF of a time series with a unit root nonstationarity. That is, the IACF of W dies off very slowly. You can detect overdifferencing this way.

The linear trend plus white noise model previously presented is interesting. The ACF of the original data dies off slowly because of the trend. You respond by differencing, and then the IACF of the differenced series indicates that you have overdifferenced. This mixed signaling by the diagnostic functions simply tells you that the data do not fit an ARMA model on the original levels scale or on the differences scale. You can obtain the correct analysis in this particular case by regressing Y on t using PROC REG or PROC GLM. The situation is different if the error series e_t is not white noise, but is instead a nonstationary time series whose difference, $e_t - e_{t-1}$, is stationary.

In this case, a model in the differences is appropriate and has an intercept estimating β . This scenario seems to hold in the publishing and printing data that produce the plot (U.S. Bureau of Labor Statistics 1977) shown in **Output 3.23**. The data are the percentages of nonproduction workers in the industry over several years.

Output 3.23: Plotting the Original Series



The ACF shown in **Output 3.24a** is obtained by specifying the following statements:

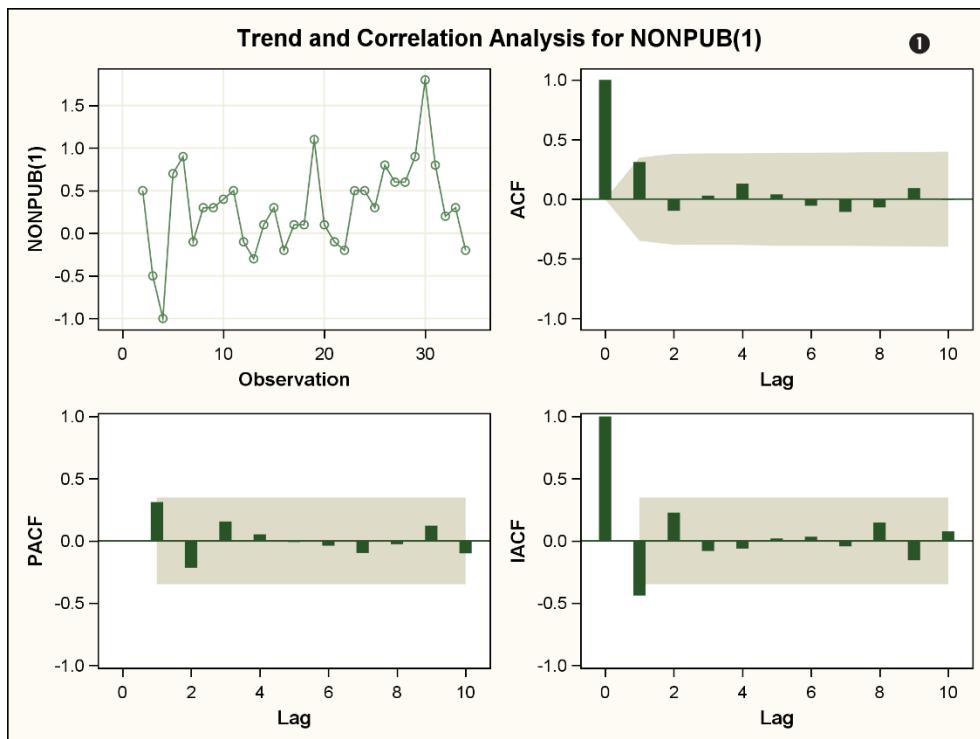
```
proc arima data=workers;
  identify var=nonpub(1) nlag=10;
  title 'PUBLISHING AND PRINTING NONPRODUCTION WORKERS';
  title2 '1944-1977';
run;
```

Output 3.24a: Modeling with the IDENTIFY Statement: PROC ARIMA**The ARIMA Procedure**

Warning: The value of NLAG is larger than 25% of the series length. The asymptotic approximations used for correlation based statistics and confidence intervals may be poor.

Name of Variable = NONPUB	
Period(s) of Differencing	1
Mean of Working Series	0.30303
Standard Deviation	0.513741
Number of Observations	33
Observation(s) eliminated by differencing	1

Autocorrelation Check for White Noise									
To Lag	Chi-Square	DF	Pr > ChiSq	Autocorrelations					
6	4.79	6	0.5716	0.312	-0.097	0.030	0.131	0.040	-0.055



Because the ACF ❶ looks like that of an MA(1) and because it is very common to fit an MA(1) term when a first difference is taken, you do that fitting by specifying these statements:

```
proc arima data=workers;
  identify var=nonpub(1) noint;
  estimate q=1;
  forecast lead=10;
run;
```

This program generates **Output 3.24b**.

Output 3.24b: Estimating and Forecasting with the ESTIMATE and FORECAST Statements: PROC ARIMA

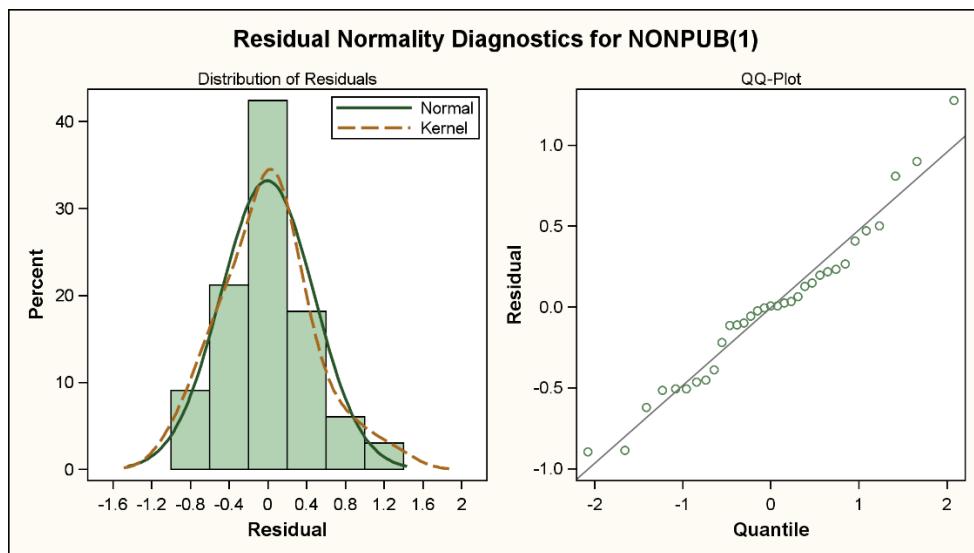
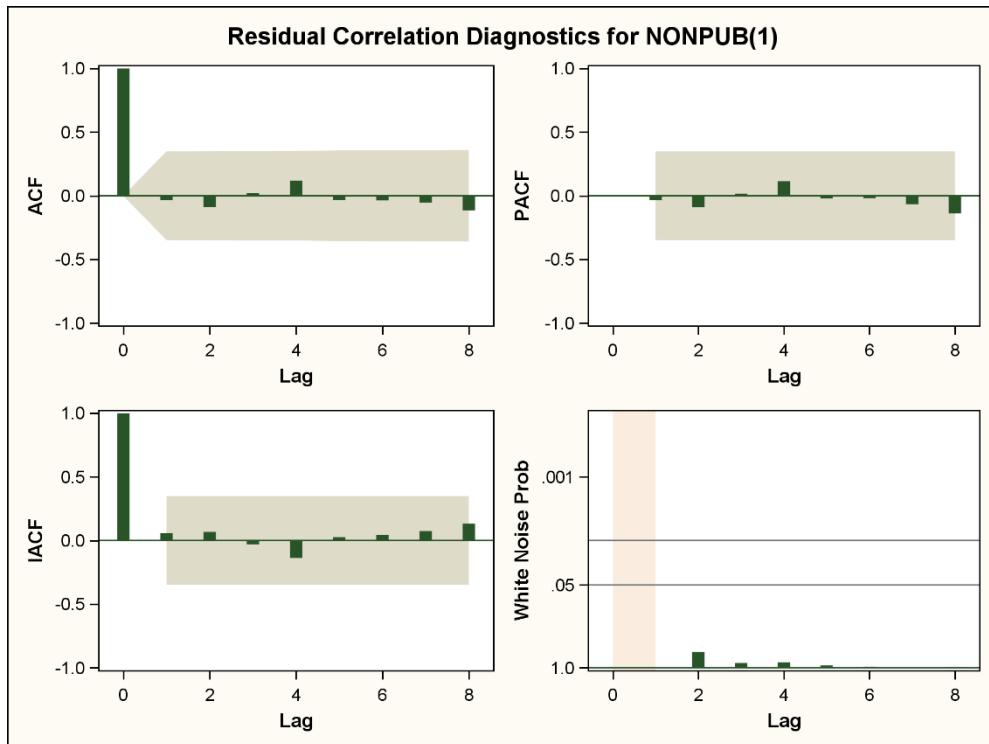
Conditional Least Squares Estimation					
Parameter	Estimate	Standard Error	t Value	Approx Pr > t	Lag
MU	0.30330	0.12294	2.47	0.0193	0
MA1,1	-0.46626	0.16148	-2.89	0.0070	1

Constant Estimate	0.3033
Variance Estimate	0.238422
Std Error Estimate	0.488284
AIC	48.27419
SBC	51.2672
Number of Residuals	33

* AIC and SBC do not include log determinant.

Correlations of Parameter Estimates		
Parameter	MU	MA1,1
MU	1.000	0.006
MA1,1	0.006	1.000

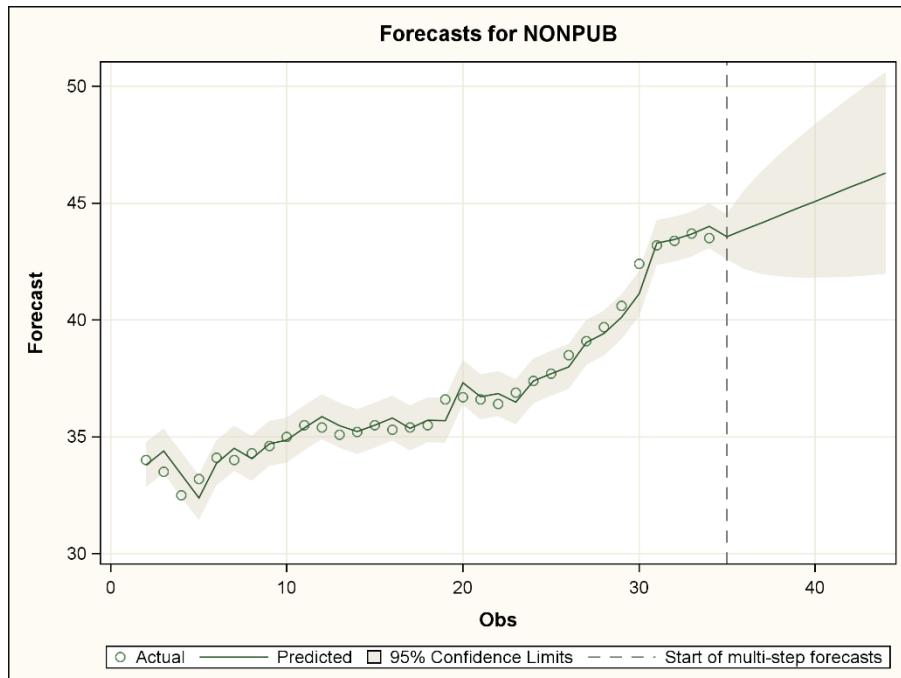
Autocorrelation Check of Residuals										
To Lag	Chi-Square	DF	Pr > ChiSq	Autocorrelations						
6	1.01	5	0.9619	-0.033	-0.089	0.020	0.119	-0.032	-0.036	
12	3.80	11	0.9754	-0.054	-0.114	0.157	-0.093	0.072	-0.057	
18	7.41	17	0.9776	0.064	0.001	-0.175	-0.108	-0.027	0.085	
24	10.07	23	0.9909	0.007	-0.023	0.067	-0.003	-0.123	0.057	



Model for variable NONPUB	
Estimated Mean	0.3033
Period(s) of Differencing	1

Moving Average Factors	
Factor 1:	$1 + 0.46626 B^{**}(1)$

Forecasts for variable NONPUB				
Obs	Forecast	Std Error	95% Confidence Limits	
35	43.5635	0.4883	42.6065	44.5206
36	43.8668	0.8666	42.1683	45.5654
37	44.1701	1.1241	41.9669	46.3733
38	44.4734	1.3327	41.8613	47.0855
39	44.7767	1.5129	41.8116	47.7419
40	45.0800	1.6737	41.7996	48.3605
41	45.3833	1.8204	41.8154	48.9513
42	45.6866	1.9562	41.8526	49.5206
43	45.9899	2.0831	41.9072	50.0727
44	46.2932	2.2027	41.9761	50.6104



The output shows a good fit based on the Q statistics ❷, the parameter estimates, and their t statistics ❸. The MU (0.3033) ❹ estimate is statistically significant and is approximately the slope in the plot of the data. Also, the MA coefficient is not near 1. In fact, it is a negative number. Thus, you have little evidence of overdifferencing. With only 33 observations, you have a lot of sampling variability (for example, look at the two standard error marks on the ACF). The number 0.3033 is sometimes called *drift*.

3.4.10 Other Identification Techniques

In addition to the ACF, IACF, and PACF, three methods called ESACF, SCAN, and MINIC are available for simultaneously identifying both the AR and MA orders. These consist of tables with rows labeled AR 0, AR 1, etc., and columns labeled MA 0, MA 1, etc. You look at the table entries to find the row and column whose labels give the correct p and q . Tsay and Tiao (1984, 1985) develop the ESACF and SCAN methods and show they even work when the autoregressive operator has roots on the unit circle. In that case, $p + d$ rather than p is found. For $(Y_t - Y_{t-2}) - 0.7(Y_{t-1} - Y_{t-3}) = e_t$, ESACF and SCAN should give 3 as the autoregressive order. The key to showing their results is that standard estimation techniques give consistent estimators of the autoregressive operator coefficients even in the presence of unit roots.

These methods can be understood through an ARMA(1,1) example. Suppose you have the ARMA(1,1) process $Z_t - \alpha Z_{t-1} = e_t - \beta e_{t-1}$, where Z_t is the deviation from the mean at time t . The autocorrelations $\rho(j)$ are $\rho(0) = 1$, $\rho(1) = [(\alpha - \beta)(1 - \alpha\beta)]/[1 - \alpha^2 + (\beta - \alpha)^2]$, and $\rho(j) = \alpha\rho(j - 1)$ for $j > 1$.

The partial autocorrelations are motivated by the problem of finding the best linear predictor of Z_t based on Z_{t-1}, \dots, Z_{t-k} . That is, you want to find coefficients φ_{kj} for which $E\{(Z_t - \varphi_{k1}Z_{t-1} - \varphi_{k2}Z_{t-2} - \dots - \varphi_{kk}Z_{t-k})^2\}$ is minimized. This is sometimes referred to as performing a theoretical regression of Z_t on $Z_{t-1}, Z_{t-2}, \dots, Z_{t-k}$ or projecting Z_t onto the space spanned by $Z_{t-1}, Z_{t-2}, \dots, Z_{t-k}$. It is accomplished by solving the matrix system of equations:

$$\begin{pmatrix} 1 & \rho(1) & \cdots & \rho(k-1) \\ \rho(1) & 1 & \cdots & \rho(k-2) \\ \vdots & \vdots & \ddots & \vdots \\ \rho(k-1) & \rho(k-2) & \cdots & 1 \end{pmatrix} \begin{pmatrix} \varphi_{k1} \\ \varphi_{k2} \\ \vdots \\ \varphi_{kk} \end{pmatrix} = \begin{pmatrix} \rho(1) \\ \rho(2) \\ \vdots \\ \rho(k) \end{pmatrix}$$

Letting $\pi_k = \varphi_{kk}$ for $k = 1, 2, \dots$ produces the sequence π_k of partial autocorrelations. (See the subsection “Partial Autocorrelation Function” in section 3.3.2.)

At $k = 1$ in the ARMA(1,1) example, you note that $\varphi_{11} = \pi_1 = \rho(1) = [(\alpha - \beta)(1 - \alpha\beta)]/[1 - \alpha^2 + (\beta - \alpha)^2]$, which does not in general equal α . Therefore, $Z_t - \varphi_{11}Z_{t-1}$ is not $Z_t - \alpha Z_{t-1}$ and does not equal $e_t - \beta e_{t-1}$. The autocorrelations of $Z_t - \varphi_{11}Z_{t-1}$ would not drop to 0 beyond the MA order. Increasing k beyond 1 does not solve the problem. Still, it is clear that there is some linear combination of Z_t and Z_{t-1} , namely $Z_t - \alpha Z_{t-1}$, whose autocorrelations theoretically identify the order of the moving average part of your model. In general, neither the π_k sequence nor any φ_{kj} sequence contains the autoregressive coefficients unless the process is a pure autoregression. You are looking for a linear combination $Z_t - C_1Z_{t-1} - C_2Z_{t-2} - \dots - C_pZ_{t-p}$ whose autocorrelation is 0 for j exceeding the MA order q (1 in this example). The trick is to discover p and the C_j s from the data.

The lagged residual from the theoretical regression of Z_t on Z_{t-1} is $R_{1,t-1} = Z_{t-1} - \varphi_{11}Z_{t-2}$, which is a linear combination of Z_{t-1} and Z_{t-2} , so regressing Z_t on Z_{t-1} and $R_{1,t-1}$ produces regression coefficients, for example C_{21} and C_{22} , which give the same fit or projection as regressing Z_t on Z_{t-1} and Z_{t-2} . That is, $C_{21}Z_{t-1} + C_{22}R_{1,t-1} = C_{21}Z_{t-1} + C_{22}(Z_{t-1} - \varphi_{11}Z_{t-2}) = \varphi_{21}Z_{t-1} + \varphi_{22}Z_{t-2}$. Thus, it must be that $\varphi_{21} = C_{21} + C_{22}$ and $\varphi_{22} = -\varphi_{11}C_{22}$. You have the following, in matrix form:

$$\begin{pmatrix} \varphi_{21} \\ \varphi_{22} \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 0 & -\varphi_{11} \end{pmatrix} \begin{pmatrix} C_{21} \\ C_{22} \end{pmatrix}$$

Noting that $\rho(2) = \alpha\rho(1)$, you see that the φ_{2j} coefficients satisfy the following:

$$\begin{pmatrix} 1 & \rho(1) \\ \rho(1) & 1 \end{pmatrix} \begin{pmatrix} \varphi_{21} \\ \varphi_{22} \end{pmatrix} = \rho(1) \begin{pmatrix} 1 \\ \alpha \end{pmatrix}$$

Relating this to the C s and noting that $\rho(1) = \varphi_{11}$, you have

$$\begin{pmatrix} 1 & \varphi_{11} \\ \varphi_{11} & 1 \end{pmatrix} \begin{pmatrix} 1 & 1 \\ 0 & -\varphi_{11} \end{pmatrix} \begin{pmatrix} C_{21} \\ C_{22} \end{pmatrix} = \varphi_{11} \begin{pmatrix} 1 \\ \alpha \end{pmatrix}$$

or

$$\begin{pmatrix} C_{21} \\ C_{22} \end{pmatrix} = \frac{\varphi_{11}}{\varphi_{11}(\varphi_{11}^2 - 1)} \begin{pmatrix} 0 & \varphi_{11}^2 - 1 \\ -\varphi_{11} & 1 \end{pmatrix} \begin{pmatrix} 1 \\ \alpha \end{pmatrix} = \begin{pmatrix} \alpha \\ \frac{\alpha - \varphi_{11}}{\varphi_{11}^2 - 1} \end{pmatrix}$$

You now filter Z by using only $C_{21} = \alpha$. That is, you compute $Z_t - C_{21}Z_{t-1}$, which is just $Z_t - \alpha Z_{t-1}$. This, in turn, is a moving average of order 1. Its lag 1 autocorrelation (it is nonzero) will appear in the AR 1 row and MA 0 column of the ESACF table. Let the residual from this regression be denoted $R_{2,t}$. The next step is to regress Z_t on $Z_{t-1}, R_{1,t-2}$, and $R_{2,t-1}$. In this regression, the theoretical coefficient of Z_{t-1} will again be α , but its estimate might differ somewhat from the one obtained previously. Notice the use of the lagged value of $R_{2,t}$ and the second lag of the first round residual $R_{1,t-2} = Z_{t-2} - \varphi_{11}Z_{t-3}$. The lag 2 autocorrelation of $Z_t - \alpha Z_{t-1}$, which is 0, will be written in the MA 1 column of the AR 1 row. For the ESACF of a general ARMA(p,q) in the AR p row, once your regression has at least q lagged residuals, the first p theoretical C_{kj} will be the p autoregressive coefficients and the filtered series will be a MA(q), so its autocorrelations will be 0 beyond lag q .

The entries in the AR k row of the ESACF table are computed as follows:

1. Regress Z_t on $Z_{t-1}, Z_{t-2}, \dots, Z_{t-k}$ with residual $R_{1,t}$. Coefficients are $C_{11}, C_{12}, \dots, C_{1k}$.
2. Regress Z_t on $Z_{t-1}, Z_{t-2}, \dots, Z_{t-k}, R_{1,t-1}$ with residual $R_{2,t}$. Second-round coefficients are C_{21}, \dots, C_{2k} , (and $C_{2,k+1}$). Record in MA 0 column the lag 1 autocorrelation of $Z_t - C_{21}Z_{t-1} - C_{22}Z_{t-2} - \dots - C_{2k}Z_{t-k}$.
3. Regress Z_t on $Z_{t-1}, Z_{t-2}, \dots, Z_{t-k}, R_{1,t-2}, R_{2,t-1}$ with residual $R_{3,t}$. Third-round coefficients are C_{31}, \dots, C_{3k} , (and $C_{3,k+1}, C_{3,k+2}$). Record in MA 1 column the lag 2 autocorrelation of $Z_t - C_{31}Z_{t-1} - C_{32}Z_{t-2} - \dots - C_{3k}Z_{t-k}$.

Continue in this way. At each step, you lag all residuals that were previously included as regressors and add the lag of the most recent residual to your regression. The estimated C coefficients and resulting filtered series differ at each step. Looking down the ESACF table of an AR(p,q), theoretically row p should be the first row in which a string of 0s appears. It should start at the MA q column. Finding that row and the first 0 entry in it puts you in row p column q of the ESACF. The model is now identified.

Here is a theoretical ESACF table for an ARMA(1,1) with X for nonzero numbers:

	MA 0	MA 1	MA 2	MA 3	MA 4	MA 5
AR 0	X	X	X	X	X	X
AR 1	X	0*	0	0	0	0
AR 2	X	X	0	0	0	0
AR 3	X	X	X	0	0	0
AR 4	X	X	X	X	0	0

The string of 0s slides to the right as the AR row number moves beyond p , so there appears a triangular array of 0s whose point 0* is at the correct (p,q) combination.

In practice, the theoretical regressions are replaced by least squares regressions, so the ESACF table will have only numbers near 0 where the theoretical ESACF table has 0s. A recursive algorithm is used to quickly compute the needed coefficients without having to compute so many actual regressions. PROC ARIMA will use asymptotically valid standard errors based on Bartlett's formula to deliver a table of approximate p -values for the ESACF entries and will suggest values of p and q as a tentative identification. See Tsay and Tiao (1984) for additional details.

Tsay and Tiao (1985) suggest a second table called SCAN. It is computed using canonical correlations. For the ARMA(1,1) model, recall that the autocovariances are $\gamma(0), \gamma(1), \gamma(2) = \alpha\gamma(1), \gamma(3) = \alpha^2\gamma(1), \gamma(4) = \alpha^3\gamma(1)$, and so forth, so the covariance matrix of $Y_t, Y_{t-1}, \dots, Y_{t-5}$ is this:

$$\Gamma = \begin{pmatrix} \gamma(0) & \gamma(1) & \alpha\gamma(1) & \alpha^2\gamma(1) & \alpha^3\gamma(1) & \alpha^4\gamma(1) \\ \gamma(1) & \gamma(0) & \gamma(1) & \alpha\gamma(1) & \alpha^2\gamma(1) & \alpha^3\gamma(1) \\ [\alpha\gamma(1)] & [\gamma(1)] & \gamma(0) & \gamma(1) & \alpha\gamma(1) & \alpha^2\gamma(1) \\ [\alpha^2\gamma(1)] & [\alpha\gamma(1)] & \gamma(1) & \gamma(0) & \gamma(1) & \alpha\gamma(1) \\ \alpha^3\gamma(1) & \alpha^2\gamma(1) & \alpha\gamma(1) & \gamma(1) & \gamma(0) & \gamma(1) \\ \alpha^4\gamma(1) & \alpha^3\gamma(1) & \alpha^2\gamma(1) & \alpha\gamma(1) & \gamma(1) & \gamma(0) \end{pmatrix}$$

The entries in square brackets form the 2×2 submatrix of covariances between the vectors $(Y_t, Y_{t-1})'$ and $(Y_{t-2}, Y_{t-3})'$. That submatrix \mathbf{A} , the variance matrix \mathbf{C}_{11} of $(Y_t, Y_{t-1})'$, and the variance matrix \mathbf{C}_{22} of $(Y_{t-2}, Y_{t-3})'$ are as follows:

$$\mathbf{A} = \gamma(1) \begin{pmatrix} \alpha & 1 \\ \alpha^2 & \alpha \end{pmatrix} \quad \mathbf{C}_{11} = \mathbf{C}_{22} = \begin{pmatrix} \gamma(0) & \gamma(1) \\ \gamma(1) & \gamma(0) \end{pmatrix}$$

The best linear predictor of $(Y_t, Y_{t-1})'$ based on $(Y_{t-2}, Y_{t-3})'$ is

$$\mathbf{A}' \mathbf{C}_{22}^{-1} (Y_{t-2}, Y_{t-3})'$$

with the following prediction error variance matrix:

$$\mathbf{C}_{11} - \mathbf{A}' \mathbf{C}_{22}^{-1} \mathbf{A}$$

Because matrix \mathbf{C}_{11} represents the variance of (Y_t, Y_{t-1}) , the matrix

$$\mathbf{C}_{11}^{-1}\mathbf{A}'\mathbf{C}_{22}^{-1}\mathbf{A}$$

is analogous to a regression R^2 statistic. Its eigenvalues are called *squared canonical correlations* between $(Y_t, Y_{t-1})'$ and $(Y_{t-2}, Y_{t-3})'$.

Recall that for a square matrix \mathbf{M} , if a column vector \mathbf{H} exists such that $\mathbf{MH} = b\mathbf{H}$, then \mathbf{H} is called an *eigenvector* and the scalar b is the corresponding eigenvalue of matrix \mathbf{M} . Using $\mathbf{H} = (1, -\alpha)'$, you see that $\mathbf{AH} = (0, 0)'$, so

$$\mathbf{C}_{11}^{-1}\mathbf{A}'\mathbf{C}_{22}^{-1}\mathbf{AH} = \mathbf{0H}$$

That is,

$$\mathbf{C}_{11}^{-1}\mathbf{A}'\mathbf{C}_{22}^{-1}\mathbf{A}$$

has an eigenvalue 0. The number of 0 eigenvalues of \mathbf{A} is the same as the number of 0 eigenvalues of

$$\mathbf{C}_{11}^{-1}\mathbf{A}'\mathbf{C}_{22}^{-1}\mathbf{A}.$$

This is true for general time series covariance matrices.

The matrix \mathbf{A} has a first column that is α times the second, which implies these equivalent statements:

- The 2×2 matrix \mathbf{A} is not of full rank (its rank is 1).
- The 2×2 matrix \mathbf{A} has at least one eigenvalue 0.
- The 2×2 matrix $\mathbf{C}_{11}^{-1}\mathbf{A}'\mathbf{C}_{22}^{-1}\mathbf{A}$ has at least one eigenvalue 0.
- The vectors (Y_t, Y_{t-1}) and (Y_{t-2}, Y_{t-3}) have at least one squared canonical correlation that is 0.

The fourth of these statements is easily seen. The linear combinations $Y_t - \alpha Y_{t-1}$ and its second lag $Y_{t-2} - \alpha Y_{t-3}$ have correlation 0 because each is an MA(1). The smallest canonical correlation is obtained by taking linear combinations of (Y_t, Y_{t-1}) and (Y_{t-2}, Y_{t-3}) and finding the pair with correlation closest to 0. Because linear combinations exist in the two sets that are uncorrelated, the smallest canonical correlation must be 0. Again, you have a method of finding a linear combination whose autocorrelation sequence is 0 beyond the moving average lag q .

In general, construct an arbitrarily large covariance matrix of $Y_t, Y_{t-1}, Y_{t-2}, \dots$, and let $\mathbf{A}_{j,m}$ be the $m \times m$ matrix whose upper left element is in row $j+1$, column 1 of the original matrix. In this notation, the \mathbf{A} with square-bracketed elements is denoted $\mathbf{A}_{2,2}$, and the bottom left 3×3 matrix of Γ is $\mathbf{A}_{3,3}$. There is a full-rank 3×2 matrix \mathbf{H} for which $\mathbf{A}_{3,3}\mathbf{H}$ has all 0 elements—namely, the following form:

$$\mathbf{A}_{3,3}\mathbf{H} = \begin{pmatrix} \alpha^2\gamma(1) & \alpha\gamma(1) & \gamma(1) \\ \alpha^3\gamma(1) & \alpha^2\gamma(1) & \alpha\gamma(1) \\ \alpha^4\gamma(1) & \alpha^3\gamma(1) & \alpha^2\gamma(1) \end{pmatrix} \begin{pmatrix} 1 & 0 \\ -\alpha & 1 \\ 0 & -\alpha \end{pmatrix} = \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{pmatrix}$$

This shows that matrix $\mathbf{A}_{3,3}$ has (at least) two eigenvalues that are 0 with the columns of \mathbf{H} being the corresponding eigenvectors. Similarly, using $\mathbf{A}_{3,2}$ and $\mathbf{H} = (1, -\alpha)$, you have the following:

$$\mathbf{A}_{3,2}\mathbf{H} = \begin{pmatrix} \alpha^2\gamma(1) & \alpha\gamma(1) \\ \alpha^3\gamma(1) & \alpha^2\gamma(1) \end{pmatrix} \begin{pmatrix} 1 \\ -\alpha \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

So, $\mathbf{A}_{3,2}$ has (at least) one 0 eigenvalue, as does $\mathbf{A}_{j,2}$ for all $j > 1$. In fact, all $\mathbf{A}_{j,m}$ with $j > 1$ and $m > 1$ have at least one 0 eigenvalue for this example. For general ARIMA(p,q) models, all $\mathbf{A}_{j,m}$ with $j > q$ and $m > p$ have at least one 0 eigenvalue. This provides the key to the SCAN table. If you make a table whose m th row, j th column entry is the smallest canonical correlation derived from $\mathbf{A}_{j,m}$, you have this table for the current example:

	$m = 1$	$m = 2$	$m = 3$	$m = 4$
$j = 1$	X	X	X	X
$j = 2$	X	0	0	0
$j = 3$	X	0	0	0
$j = 4$	X	0	0	0

	$p = 0$	$p = 1$	$p = 2$	$p = 3$
$q = 0$	X	X	X	X
$q = 1$	X	0	0	0
$q = 2$	X	0	0	0
$q = 3$	X	0	0	0

In this table, the Xs represent nonzero numbers. Relabeling the rows and columns with $q = j - 1$ and $p = m - 1$ provides the SCAN (smallest canonical correlation) table. It has a rectangular array of 0s whose upper left corner is at the p and q corresponding to the correct model, ARMA(1,1), for the current example. The first column of the SCAN table consists of the autocorrelations and the first row consists of the partial autocorrelations.

In PROC ARIMA, entries of the previous 6×6 variance-covariance matrix Γ would be replaced by estimated autocovariances. To see why the 0s appear for an ARMA(p,q) whose autoregressive coefficients are α_i , you notice from the Yule-Walker equations that $\gamma(j) - \alpha_1\gamma(j-1) - \alpha_2\gamma(j-2) - \dots - \alpha_p\gamma(j-p)$ is zero for $j > q$. Therefore, in the variance covariance matrix for such a process, any $m \times m$ submatrix with $m > p$ whose upper left element is at row j , column 1 of the original matrix will have at least one 0 eigenvalue with eigenvector $(1, -\alpha_1, -\alpha_2, \dots, -\alpha_p, 0, 0, \dots, 0)'$ if $j > q$. Therefore, 0 will appear in the theoretical table whenever $m > p$ and $j > q$. Approximate standard errors are obtained by applying Bartlett's formula to the series filtered by the autoregressive coefficients, which in turn can be extracted from the \mathbf{H} matrix (eigenvectors). An asymptotically valid test, using Bartlett's formula, is available and PROC ARIMA displays a table of the resulting p -values.

The MINIC method simply attempts to fit models over a grid of p and q choices. It records the SBC information criterion for each fit in a table. The Schwarz Bayesian Information Criterion is $SBC = n \ln(s^2) + (p+q)\ln(n)$, where p and q are the AR and MA orders of the candidate model. It is an estimate of the innovations variance. Some sources refer to Schwarz's criterion, perhaps normalized by n , as *BIC*. Here, the symbol SBC is used so that Schwarz's criterion will not be confused with the BIC criterion of Sawa (1978). Sawa's BIC, used as a model selection tool in PROC REG, is this:

$$n \ln(s^2) + 2 \left[(k+2) \frac{n}{n-k} - \left(\frac{n}{n-k} \right)^2 \right]$$

It is for a full regression model with n observations and k parameters. The MINIC technique chooses p and q giving the smallest SBC. It is possible that the fitting will fail because of singularities. In this case, SBC is set to missing.

The fitting of models in computing MINIC follows a clever algorithm suggested by Hannan and Rissanen (1982) using ideas dating back to Durbin (1960). First, using the Yule-Walker equations, a long autoregressive model is fit to the data. For the ARMA(1,1) example of this section, the following is seen:

$$Y_t = (\alpha - \beta) [Y_{t-1} + \beta Y_{t-2} + \beta^2 Y_{t-3} + \dots] + e_t$$

As long as $|\beta| < 1$, the coefficients on lagged Y will die off quickly, indicating that a truncated version of this infinite autoregression will approximate the e_t process well. To the extent that this is true, the Yule-Walker equations for a length k (k large) autoregression can be solved to give estimates, for example $\hat{\beta}_j$, of the coefficients of the Y_{t-j} terms and a

residual series $\hat{e}_t = Y_t - \hat{b}_1 Y_{t-1} - \hat{b}_2 Y_{t-2} - \dots - \hat{b}_k Y_{t-k}$ that is close to the actual e_t series. Next, for a candidate model of order p,q , regress Y_t on $Y_{t-1}, \dots, Y_{t-p}, \hat{e}_{t-1}, \hat{e}_{t-2}, \dots, \hat{e}_{t-q}$. Letting

$$\hat{\sigma}_{pq}^2$$

be $1/n$ times the error sum of squares for this regression, choose p and q to minimize the SBC criterion:

$$\text{SBC} = n \ln(\hat{\sigma}_{pq}^2) + (p+q) \ln(n)$$

You can select the length of the autoregressive model for the \hat{e}_t series by minimizing the AIC criterion.

To illustrate, 1000 observations on an ARMA(1,1) with $\alpha=0.8$ and $\beta=0.4$ are generated and analyzed. The following code generates **Output 3.25**:

```
proc arima data=a;
  i var=y nlag=1 minic p=(0:5) q=(0:5);
  i var=y nlag=1 esacf p=(0:5) q=(0:5);
  i var=y nlag=1 scan p=(0:5) q=(0:5);
run;
```

Output 3.25: ESACF, SCAN, and MINIC Displays

Minimum Information Criterion						
Lags	MA 0	MA 1	MA 2	MA 3	MA 4	MA 5
AR 0	0.28456	0.177502	0.117561	0.059353	0.028157	0.003877
AR 1	-0.0088	-0.04753	-0.04502	-0.0403	-0.03565	-0.03028
AR 2	-0.03958	-0.04404	-0.04121	-0.0352	-0.03027	-0.02428
AR 3	-0.04837	-0.04168	-0.03537	-0.02854	-0.02366	-0.01792
AR 4	-0.04386	-0.03696	-0.03047	-0.02372	-0.01711	-0.01153
AR 5	-0.03833	-0.03145	-0.02461	-0.0177	-0.01176	-0.00497

Error series model: AR(9)
Minimum Table Value: BIC(3,0) = -0.04837

Extended Sample Autocorrelation Function						
Lags	MA 0	MA 1	MA 2	MA 3	MA 4	MA 5
AR 0	0.5055	0.3944	0.3407	0.2575	0.2184	0.1567
AR 1	-0.3326	-0.0514	0.0564	-0.0360	0.0417	-0.0242
AR 2	-0.4574	-0.2993	0.0197	0.0184	0.0186	-0.0217
AR 3	-0.1207	-0.2357	0.1902	0.0020	0.0116	0.0006
AR 4	-0.4074	-0.1753	0.1942	-0.0132	0.0119	0.0015
AR 5	0.4836	0.1777	-0.0733	0.0336	0.0388	-0.0051

ESACF Probability Values						
Lags	MA 0	MA 1	MA 2	MA 3	MA 4	MA 5
AR 0	<.0001	<.0001	<.0001	<.0001	<.0001	0.0010
AR 1	<.0001	0.1489	0.1045	0.3129	0.2263	0.4951
AR 2	<.0001	<.0001	0.5640	0.6013	0.5793	0.6003
AR 3	0.0001	<.0001	<.0001	0.9598	0.7634	0.9874
AR 4	<.0001	<.0001	<.0001	0.7445	0.7580	0.9692
AR 5	<.0001	<.0001	0.0831	0.3789	0.2880	0.8851

ARMA(p+d,q) Tentative Order Selection Tests	
ESACF	
p+d	q
1	1
4	3
5	2

(5% Significance Level)

Squared Canonical Correlation Estimates						
Lags	MA 0	MA 1	MA 2	MA 3	MA 4	MA 5
AR 0	0.2567	0.1563	0.1170	0.0670	0.0483	0.0249
AR 1	0.0347	0.0018	0.0021	0.0008	0.0011	0.0003
AR 2	0.0140	0.0023	0.0002	0.0002	0.0002	0.0010
AR 3	0.0002	0.0007	0.0002	<.0001	0.0002	<.0001
AR 4	0.0008	0.0010	0.0002	0.0002	0.0002	0.0002
AR 5	0.0005	<.0001	0.0002	<.0001	0.0002	0.0004

SCAN Chi-Square[1] Probability Values						
Lags	MA 0	MA 1	MA 2	MA 3	MA 4	MA 5
AR 0	<.0001	<.0001	<.0001	<.0001	<.0001	0.0010
AR 1	<.0001	0.2263	0.1945	0.4097	0.3513	0.5935
AR 2	0.0002	0.1849	0.7141	0.6767	0.7220	0.3455
AR 3	0.6467	0.4280	0.6670	0.9731	0.6766	0.9877
AR 4	0.3741	0.3922	0.6795	0.6631	0.7331	0.7080
AR 5	0.4933	0.8558	0.7413	0.9111	0.6878	0.6004

ARMA(p+d,q) Tentative Order Selection Tests	
SCAN	
p+d	q
1	1
3	0

(5% Significance Level)

The tentative order selections in ESACF and SCAN simply look at all triangles (rectangles) for which every element is insignificant at the specified level (.05, by default). These are listed in descending order of size (below the tables), with size being the number of elements in the triangle or rectangle. In this example, ESACF and SCAN list the correct (1,1) order at the top of the list. The MINIC criterion uses $k = 9$, a preliminary AR(9) model, to create the estimated white noise series, and then selects $(p,q) = (3,0)$ as the order. This is also one choice given by the SCAN option. The second smallest SBC, -0.04753 , occurs at the correct $(p,q) = (1,1)$.

As a check on the relative merits of these methods, 50 ARMA(1,1) series, each of length 500, are generated for each of the 12 (α, β) pairs obtained by choosing α and β from $\{-0.9, -0.3, 0.3, 0.9\}$ such that $\alpha \neq \beta$. This gives 600 series. For each, the ESACF, SCAN, and MINIC methods are used, the results are saved, and the estimated p and q are extracted for each method. The whole experiment is repeated with series of length 50. A final set of 600 runs for $Y_t = 0.5Y_{t-4} + e_t + 0.3e_{t-1}$ using $n = 50$ gives the last three columns. Asterisks indicate the correct model.

Output 3.26: Comparison of ESACF, SCAN, and MINIC

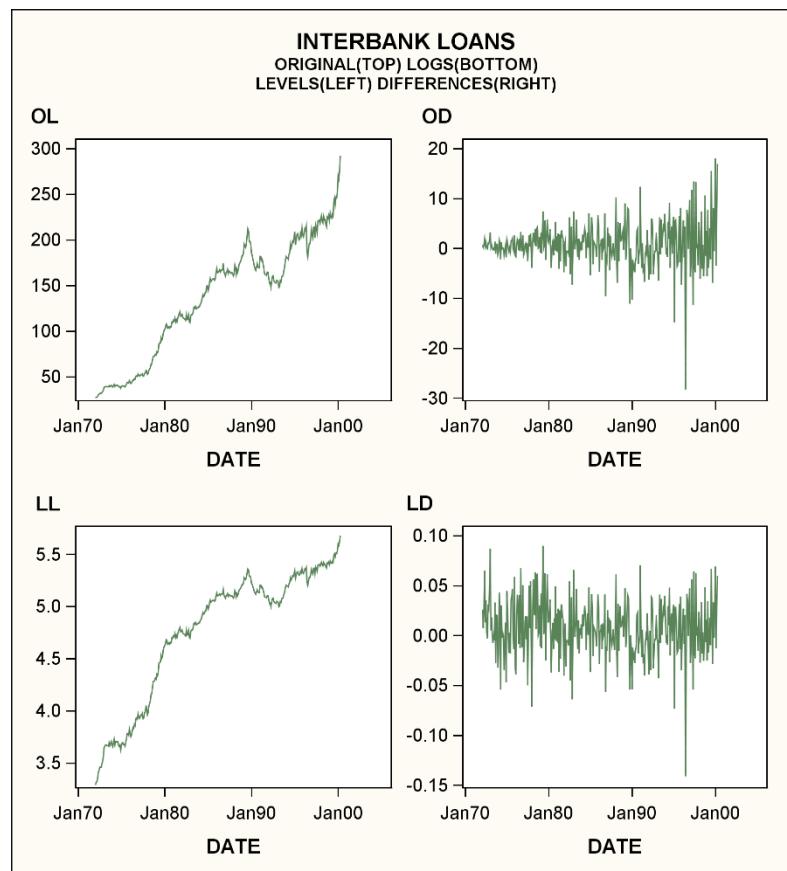
	Data Generating Process								
	ARIMA(1,1) n=50			ARIMA(1,1) n=500			ARIMA(4,1) n=50		
	BIC	ESACF	SCAN	BIC	ESACF	SCAN	BIC	ESACF	SCAN
pq									
00	25	40	25	2	1	1	69	64	35
01	48	146	126	0	0	0	28	46	33
02	17	21	8	0	0	0	5	9	11
03	7	4	16	0	0	0	4	6	2
04	7	3	2	0	0	0	41	20	35
05	6	1	0	0	0	0	14	0	2
10	112	101	145	1	0	0	28	15	38
11	**** 53	*** 165	*** 203	*** 252	*** 441	*** 461	5	47	78
12	16	7	9	13	23	8	1	10	30
13	12	0	2	13	18	8	0	1	3
14	2	0	1	17	5	3	3	0	0
15	5	0	0	53	6	0	4	0	0
20	91	41	18	95	6	12	26	16	19
21	9	22	14	9	6	25	2	42	25
22	4	8	7	24	46	32	3	62	121
23	1	2	1	1	0	1	1	2	8
24	3	0	1	4	0	2	2	1	0
25	6	0	0	10	0	1	0	0	0
30	50	6	8	35	2	9	30	9	21
31	1	10	3	5	3	11	3	23	27
32	3	4	0	3	6	1	3	21	7
33	0	2	0	3	15	13	1	16	2
34	1	0	0	5	2	0	0	0	0
35	2	0	0	4	0	0	0	0	0
40	61	6	6	5	0	0	170	66	98
41	3	4	0	3	0	5	**** 10	**** 52	***** 0
42	1	2	0	2	4	2	4	24	0
43	0	0	0	5	3	0	0	22	0
44	1	0	0	1	4	1	0	0	0
45	5	0	0	6	0	0	1	0	0
50	32	3	2	5	0	0	116	6	5
51	10	1	0	0	1	2	18	13	0
52	2	0	0	3	0	1	2	6	0
53	2	0	0	9	2	0	5	0	0
54	2	1	0	6	1	0	1	0	0
55	0	0	0	6	0	0	0	0	0
Totals	600	600	597	600	595	599	600	599	600

It is reassuring that the methods almost never underestimate p or q when n is 500. For the ARMA(1,1) with parameters in this range, it appears that SCAN does slightly better than ESACF, with both being superior to MINIC. The SCAN and ESACF columns do not always add to 600 because, for some cases, no rectangle or triangle can be found with all elements insignificant. Because SCAN compares the smallest normalized squared canonical correlation to a distribution (x_1^2) that is appropriate for a randomly selected one, it is also very conservative. By analogy, even if 5% of men exceed 6 feet in height, finding a random sample of 10 men whose shortest member exceeds 6 feet in height would be extremely rare. Thus, the appearance of a significant bottom-right-corner element in the SCAN table, which would imply no rectangle of insignificant values, happens rarely—not the 30 times you would expect from $600(0.05) = 30$.

The conservatism of the test implies that for moderately large p and q , there is a fairly good chance that a rectangle (triangle) of insignificant terms will appear by chance having p or q too small. Indeed, for 600 replicates of the model $Y_t = 0.5Y_{t-4} + e_t + 0.3e_{t-1}$ using $n = 50$, you see that $(p, q) = (4, 1)$ is rarely chosen by any technique with SCAN giving no correct choices. There does not seem to be a universally preferable choice among the three.

As a real data example, **Output 3.27** shows monthly interbank loans in billions of dollars. The data were downloaded from the Federal Reserve website. Also shown are the differences (upper right corner) and the corresponding log scale graphs. The data require differencing and the right-side graphs seem to indicate the need for logarithms to stabilize the variance.

Output 3.27: Loans



To identify the log-transformed variable (called LOANS in the data set), use this code to get the SCAN table:

```
proc arima data=ibl;
  identify var=loans scan p=(0:5) q=(0:5);
run;
```

Output 3.28 shows the SCAN results. They indicate several possible models.

Output 3.28: SCAN Table for Interbank Loans

Lags	MA 0	MA 1	MA 2	MA 3	MA 4	MA 5
AR 0	0.9976	0.9952	0.9931	0.9899	0.9868	0.9835
AR 1	<.0001	0.0037	0.0397	0.0007	0.0024	0.0308
AR 2	0.0037	0.0003	0.0317	0.0020	<.0001	0.0133
AR 3	0.0407	0.0309	0.0274	0.0126	0.0134	0.0125
AR 4	0.0004	0.0053	0.0076	0.0004	0.0022	0.0084
AR 5	0.0058	0.0003	0.0067	0.0022	<.0001	0.0078

SCAN Chi-Square[1] Probability Values						
Lags	MA 0	MA 1	MA 2	MA 3	MA 4	MA 5
AR 0	<.0001	<.0001	<.0001	<.0001	<.0001	<.0001
AR 1	0.9125	0.2653	0.0003	0.6474	0.3936	0.0019
AR 2	0.2618	0.7467	0.0033	0.4419	0.9227	0.0940
AR 3	0.0002	0.0043	0.0136	0.0856	0.0881	0.1302
AR 4	0.7231	0.1942	0.1562	0.7588	0.4753	0.1589
AR 5	0.1613	0.7678	0.1901	0.4709	0.9708	0.1836

ARMA($p+d,q$) Tentative Order Selection Tests	
SCAN	
$p+d$	q
4	0
2	3

(5% Significance Level)

The SCAN table was computed on log-transformed undifferenced data. Therefore, the listed number $p + d$ represents $p + 1$, and SCAN suggests ARIMA(3,1,0) or ARIMA(1,1,3).

```
identify var=loans(1) noint;
  estimate p=3 ml;
  estimate p=1 q=3 ml;
run;
```

The chi-square checks for both of these models are insignificant at all lags, indicating both models fit well. Both models have some insignificant parameters and could be refined by omitting some lags.

3.5 Summary of Steps for Analyzing Nonseasonal Univariate Series

The steps for analyzing nonseasonal univariate series are summarized as follows:

1. Check for nonstationarity using:
 - a. A data plot to monitor slow level shifts in the data (as in the IBM example).
 - b. ACF to monitor very slow decay (as in the IBM or publishing and printing example).
 - c. Dickey-Fuller test for stationarity (as in the silver example).

If any of these tests indicate nonstationarity, difference the series using VAR=Y(1) in the IDENTIFY statement and repeat step 1. If necessary, difference again by specifying VAR=Y(1,1).

2. Check the Q statistic (chi-square) at the bottom of the printout. If Q is small (in other words, PROB is fairly large) and if the first few autocorrelations are small, you might want to assume that your (possibly differenced) series is just white noise.
3. Check the ACF, IACF, and PACF to identify a model. If the ACF drops to 0 after q lags, this indicates an MA(q) model. If the IACF or PACF drops to 0 after p lags, this indicates an AR(p) model. If you have differenced the series once or twice, one or two MA lags are likely to be indicated.
4. You can use the SCAN, ESACF, and MINIC tables to determine initial starting models to try in an ESTIMATE statement. Using the ESTIMATE statement, specify the model that you chose (or several candidate models). For example, you fit the model $(Y_t - Y_{t-1}) = (1 - \theta_1 B) e_t$ by specifying these statements:

```
proc arima data=sasds;
  identify var=y(1);
  estimate q=1 noconstant;
run;
```

Although the default CLS estimation method has been used in these examples, the slightly more accurate maximum likelihood method can be invoked using the ML option in the ESTIMATE statement. With modern computing power, the slightly increased computation time is usually not even noticeable.

5. Check the Q statistic (chi-square) at the bottom of the ESTIMATE printout. If it is insignificant, your model fits reasonably well according to this criterion. Otherwise, return to the original ACF, IACF, and PACF of your (possibly differenced) data to determine whether you have missed something. This is generally more advisable than plotting the ACF of the residuals from this misspecified model. If you have differenced, the mean is often (IBM data), but not always (publishing and printing data), 0. Use the NOCONSTANT option to suppress the fitting of a constant. Fitting extra lags and excluding insignificant lags in an attempt to bypass identification causes unstable parameter estimates and possible convergence problems if you overfit on both sides (AR and MA) at once. Correlations of parameter estimates are extremely high in this case (if, in fact, the estimation algorithm converges). Overfitting on one side at a time to check the model is not a problem.
6. Use the FORECAST statement with LEAD= k to produce forecasts from the fitted model. It is a good idea to specify BACK= b to start the forecast b steps before the end of the series. You can then compare the last b forecasts to data values at the end of the series. If you note a large discrepancy, you might want to adjust your forecasts. You omit the BACK= option on your final forecast. It is used only as a diagnostic tool.
7. Examine plots of residuals and possibly use PROC UNIVARIATE to examine the distribution and PROC SPECTRA to test the white noise assumption further. (See Chapter 10, "Spectral Analysis," for more information.)

Chapter 4: The ARIMA Model: Introductory Applications

4.1 Seasonal Time Series	107
4.1.1 Introduction to Seasonal Modeling	107
4.1.2 Model Identification	108
4.2 Models with Explanatory Variables	119
4.2.1 Case 1: Regression with Time Series Errors	120
4.2.2 Case 1A: Intervention	120
4.2.3 Case 2: Simple Transfer Functions.....	121
4.2.4 Case 3: General Transfer Functions.....	121
4.2.5 Case 3A: Leading Indicators	121
4.2.6 Case 3B: Intervention	121
4.3 Methodology and Example.....	122
4.3.1 Case 1: Regression with Time Series Errors	122
4.3.2 Case 2: Simple Transfer Functions.....	131
4.3.3 Case 3: General Transfer Functions	133
4.3.4 Case 3B: Intervention	155
4.4 Further Example	161
4.4.1 North Carolina Retail Sales	161
4.4.2 Construction Series Revisited	168
4.4.3 Milk Scare (Intervention)	172
4.4.4 Terrorist Attack	175

4.1 Seasonal Time Series

Some time series display approximately periodic behavior, such as the increase in retail sales associated with the Christmas season. Seasonality can be very regular or slowly changing over time. No universally accepted rigorous mathematical definition of the term seems to exist, but the description of seasonality as approximately periodic behavior provides a general idea. As a result, several modeling techniques were created to handle seasonality.

4.1.1 Introduction to Seasonal Modeling

The first priority in seasonal modeling is to specify correct differencing and appropriate transformations. These are discussed first, followed by model identification. The potential behavior of autocorrelation functions (ACFs) for seasonal models is not easy to characterize, but ACFs are given for a few seasonal models. You should find a pattern that matches your data among these diagnostic plots.

Consider the following model, where e_t is white noise:

$$Y_t - \mu = \alpha(Y_{t-12} - \mu) + e_t$$

This model is applied to monthly data and expresses this December's Y (for example, as μ plus a proportion of last December's deviation from μ). If $\mu = 100$, $\alpha = 0.8$, and last December's $Y = 120$, the model forecasts this December's Y as $100 + 0.8(20) = 116$. The forecast for next December's Y is $100 + 0.64(20)$, and the forecast for j Decembers ahead is $100 + 0.8^j(20)$.

The model responds to change in the series because it uses only the most recent December to forecast the future. This approach contrasts with the indicator variables in the regression approach discussed in Chapter 1, "Overview of Time Series," where the average of all December values goes into the forecast for this December. For the previous autoregressive (AR) seasonal model, the further into the future that you forecast, the closer your forecast is to the mean μ . Suppose you allow α to be 1 in the AR seasonal model. Your model is nonstationary and reduces to $Y_t = Y_{t-12} + e_t$.

This model uses last December's Y as the forecast for next December (and for any other future December). The difference $Y_t - Y_{t-12}$ is stationary (white noise, in this case) and is specified using the PROC ARIMA statement:

```
identify var=y(12);
```

This is called a *span 12 difference*. The forecast does not tend to return to the historical series mean, as evidenced by the lack of a μ term in the model.

When you encounter a span 12 difference, often the differenced series is not white noise. Instead, it is a moving average of the form $e_t - \beta e_{t-12}$.

For example, if $Y_t - Y_{t-12} = e_t - 0.5e_{t-12}$, then you see that the following is true:

$$\begin{aligned} e_{t-12} &= Y_{t-12} - Y_{t-24} + 0.5e_{t-24} \\ &= Y_{t-12} - Y_{t-24} + 0.5(Y_{t-24} - Y_{t-36} + 0.5e_{t-36}) \end{aligned}$$

If you continue in this fashion, you can express Y_t as e_t plus an infinite weighted sum of past Y values—namely, the following:

$$Y_t = e_t + 0.5(Y_{t-12} + 0.5Y_{t-24} + \dots)$$

Thus, the forecast for any future December is a weighted sum of past December values, with weights decreasing exponentially as you move further into the past. Although the forecast involves many past Decembers, the decreasing weights make it respond more to recent December values than to those in the distant past.

Differencing over seasonal spans is indicated when the ACF at the seasonal lags dies off very slowly. Often this behavior is masked in the original ACF, which dies off slowly at all lags. In that case, you should difference, as the ACF seems to indicate, by specifying the PROC ARIMA statement:

```
identify var=y(1);
```

Now look at the ACF of the differenced series, considering only seasonal lags (12, 24, 36, and so on). If these ACF values die off very slowly, you want to take a span 12 difference in addition to the first difference. You accomplish this by specifying:

```
identify var=y(1,12);
```

Note how the differencing specification works. For example, identify var=y(1,1) specifies a second difference, $(Y_t - Y_{t-1}) - (Y_{t-1} - Y_{t-2}) = Y_t - 2Y_{t-1} + Y_{t-2}$ whereas identify var=y(2) creates the span 2 difference, $Y_t - Y_{t-2}$.

Calling the span 1 and span 12 differenced series V_t , you create $V_t = (Y_t - Y_{t-1}) - (Y_{t-12} - Y_{t-13})$ and consider models for V_t .

4.1.2 Model Identification

If V_t appears to be white noise, then the model becomes the following:

$$Y_t = Y_{t-1} + (Y_{t-12} - Y_{t-13}) + e_t$$

Thus, with data through this November, you forecast this December's Y_t as the November value (Y_{t-1}) plus last year's November-to-December change ($Y_{t-12} - Y_{t-13}$).

More commonly, you find that the differenced series V_t satisfies the following:

$$V_t = (1 - \theta_1 B)(1 - \theta_2 B^{12}) e_t$$

This is called a *seasonal multiplicative moving average*. The meaning of a product of backshift factors such as this one is simply $V_t = e_t - \theta_1 e_{t-1} - \theta_2 e_{t-12} - \delta e_{t-13}$ where $\delta = -\theta_1 \theta_2$. If you are not sure about the multiplicative structure, you can specify the following:

```
estimate q=(1,12,13);
```

You can then check to see whether the third estimated moving average (MA) coefficient δ is approximately the negative of the product of the other two ($\theta_1\theta_2$). To specify the multiplicative structure, issue the PROC ARIMA statement:

```
estimate q=(1) (12);
```

As before, the inclusion of this moving average structure has the practical effect of incorporating all past seasonality with weights declining in magnitude as you move further into the past. After differencing, the intercept is probably 0, so you can use the NOCONSTANT option. You can fit seasonal multiplicative factors on the AR side, also. For example, you can specify the following:

```
estimate p=(1,2) (12) noconstant;
```

Doing so causes the following model to be fit to the data:

$$(1 - \alpha_1 B - \alpha_2 B^2)(1 - \alpha_3 B^{12})V_t = e_t$$

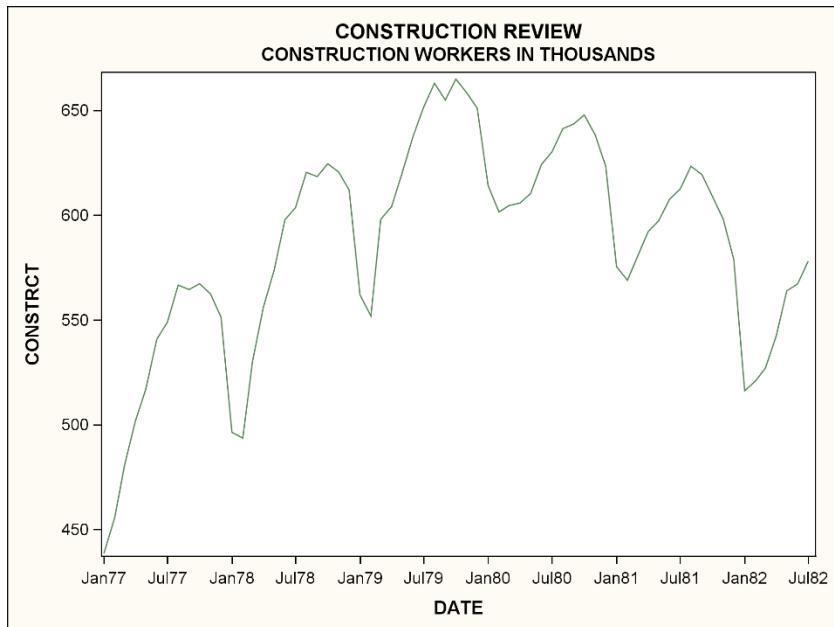
Consider the monthly number of U.S. masonry and electrical construction workers in thousands (U.S. Census Bureau 1982). You issue the following SAS statements to plot the data and compute the ACF for the original series, first differenced series, and first and seasonally differenced series:

```
proc sgplot data=const;
  series x=date y=constrct;
  xaxis valuesformat=monyy5.;
quit;

proc arima data=const;
  identify var=constrct nlag=36;
  identify var=constrct(1) nlag=36;
  identify var=constrct(1,12) nlag=36;
run;
```

The plot is shown in **Output 4.1**. The ACFs are shown in **Output 4.2**.

Output 4.1: Plotting the Original Data

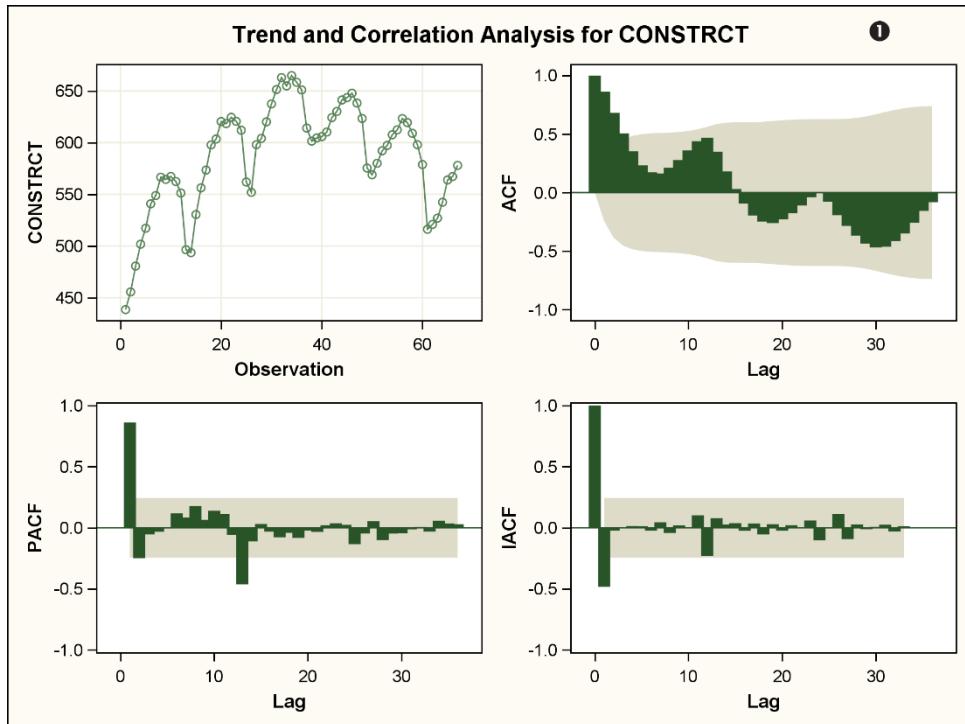


Output 4.2: Computing the ACF with the IDENTIFY Statement: PROC ARIMA**The ARIMA Procedure**

Warning: The value of NLAG is larger than 25% of the series length. The asymptotic approximations used for correlation based statistics and confidence intervals may be poor.

Name of Variable = CONSTRCT	
Mean of Working Series	585.4149
Standard Deviation	50.65318
Number of Observations	67

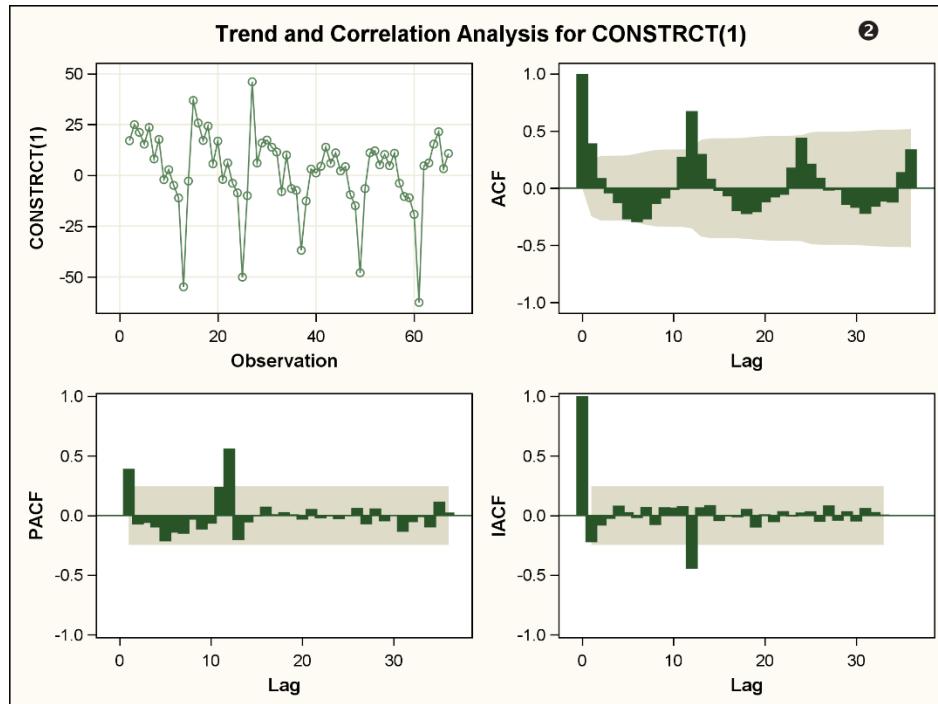
Autocorrelation Check for White Noise									
To Lag	Chi-Square	DF	Pr > ChiSq	Autocorrelations					
6	119.03	6	<.0001	0.863	0.681	0.505	0.354	0.233	0.173
12	175.54	12	<.0001	0.163	0.211	0.278	0.360	0.439	0.468
18	199.05	18	<.0001	0.349	0.181	0.031	-0.095	-0.196	-0.248
24	215.31	24	<.0001	-0.261	-0.227	-0.177	-0.111	-0.040	-0.004
30	296.18	30	<.0001	-0.078	-0.193	-0.284	-0.368	-0.436	-0.468
36	376.88	36	<.0001	-0.463	-0.414	-0.349	-0.259	-0.157	-0.082



Warning: The value of NLAG is larger than 25% of the series length. The asymptotic approximations used for correlation-based statistics and confidence intervals might be poor.

Name of Variable = CONSTRCT	
Period(s) of Differencing	1
Mean of Working Series	2.113636
Standard Deviation	19.56132
Number of Observations	66
Observation(s) eliminated by differencing	1

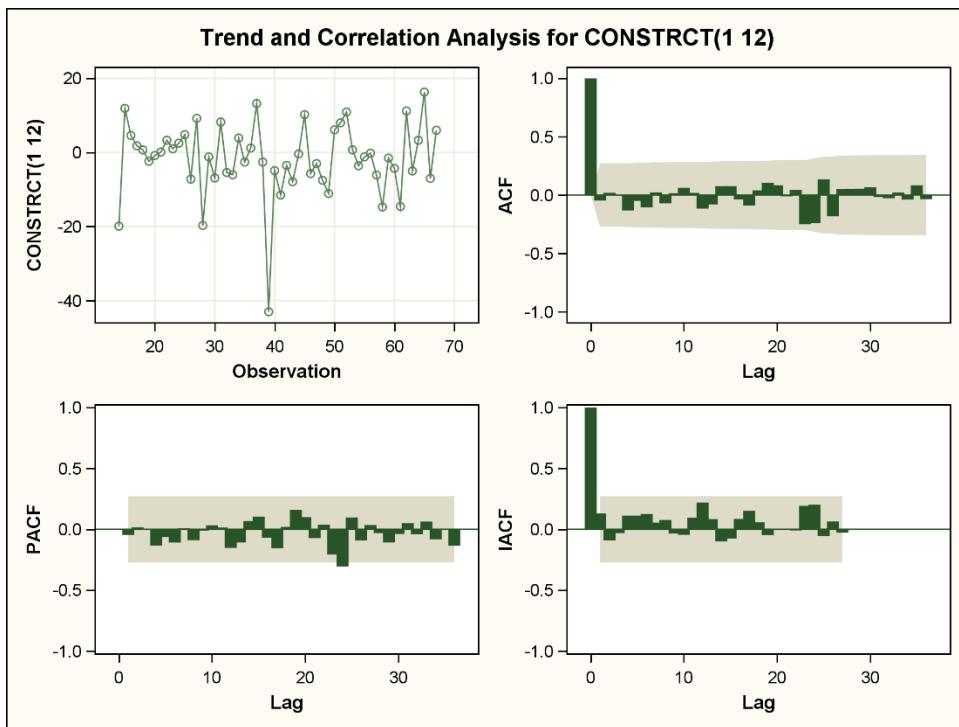
Autocorrelation Check for White Noise									
To Lag	Chi-Square	DF	Pr > ChiSq	Autocorrelations					
6	24.63	6	0.0004	0.392	0.090	-0.045	-0.131	-0.273	-0.297
12	76.44	12	<.0001	-0.273	-0.139	-0.089	-0.015	0.274	0.675
18	93.66	18	<.0001	0.300	0.083	-0.024	-0.071	-0.202	-0.226
24	124.70	24	<.0001	-0.211	-0.124	-0.081	-0.058	0.181	0.442
30	137.06	30	<.0001	0.214	0.091	-0.022	-0.016	-0.148	-0.172
36	171.23	36	<.0001	-0.224	-0.161	-0.117	-0.125	0.140	0.341



Warning: The value of NLAG is larger than 25% of the series length. The asymptotic approximations used for correlation based statistics and confidence intervals may be poor.

Name of Variable = CONSTRCT	
Period(s) of Differencing	1,12
Mean of Working Series	-1.70926
Standard Deviation	9.624434
Number of Observations	54
Observation(s) eliminated by differencing	13

Autocorrelation Check for White Noise									
To Lag	Chi-Square	DF	Pr > ChiSq	Autocorrelations					
6	2.05	6	0.9149	-0.046	0.019	-0.005	-0.132	-0.050	-0.105
12	3.70	12	0.9883	0.021	-0.071	0.016	0.062	0.019	-0.117
18	5.97	18	0.9963	-0.081	0.076	0.076	-0.037	-0.090	0.040
24	19.59	24	0.7196	0.104	0.084	-0.011	0.045	-0.249	-0.240
30	26.63	30	0.6424	0.135	-0.182	0.051	0.053	0.055	0.066
36	28.38	36	0.8134	-0.017	-0.027	0.021	-0.038	0.084	-0.035



The plot of the data displays nonstationary behavior (nonconstant mean). The original ACF ❶ shows slow decay, indicating a first differencing. The ACF of the first differenced series ❷ shows slow decay at the seasonal lags, indicating a span 12 difference. The Q statistics ❸ on the CONSTRCT(1,12) differenced variable indicate that no AR or MA terms are needed.

To forecast the seasonal data, use the following statements:

```
proc arima data=const;
  identify var=constrct(1,12) noprint;
  estimate noconstant method=ml;
  forecast lead=12 interval=month id=date out=outf;
run;
```

The results are shown in **Output 4.3**.

Output 4.3: Forecasting Seasonal Data with the IDENTIFY, ESTIMATE, and FORECAST Statements: PROC ARIMA

The ARIMA Procedure

Variance Estimate	95.5513
Std Error Estimate	9.775034
AIC	399.4672
SBC	399.4672
Number of Residuals	54

To Lag	Autocorrelation Check of Residuals								
	Chi-Square	DF	Pr > ChiSq	Autocorrelations					
6	0.93	6	0.9880	-0.019	0.048	0.031	-0.089	-0.011	-0.061
12	2.62	12	0.9977	0.057	-0.033	0.052	0.093	0.050	-0.079
18	5.22	18	0.9985	-0.046	0.108	0.102	-0.003	-0.054	0.078
24	16.49	24	0.8695	0.135	0.112	0.020	0.072	-0.209	-0.195

Model for variable CONSTRCT			
Period(s) of Differencing		1,12	

No mean term in this model.

Forecasts for variable CONSTRCT				
Obs	Forecast	Std Error	95% Confidence Limits	
68	588.9000	9.7750	569.7413	608.0587
69	585.0000	13.8240	557.9055	612.0945
70	574.6000	16.9309	541.4161	607.7839

(Additional output omitted)

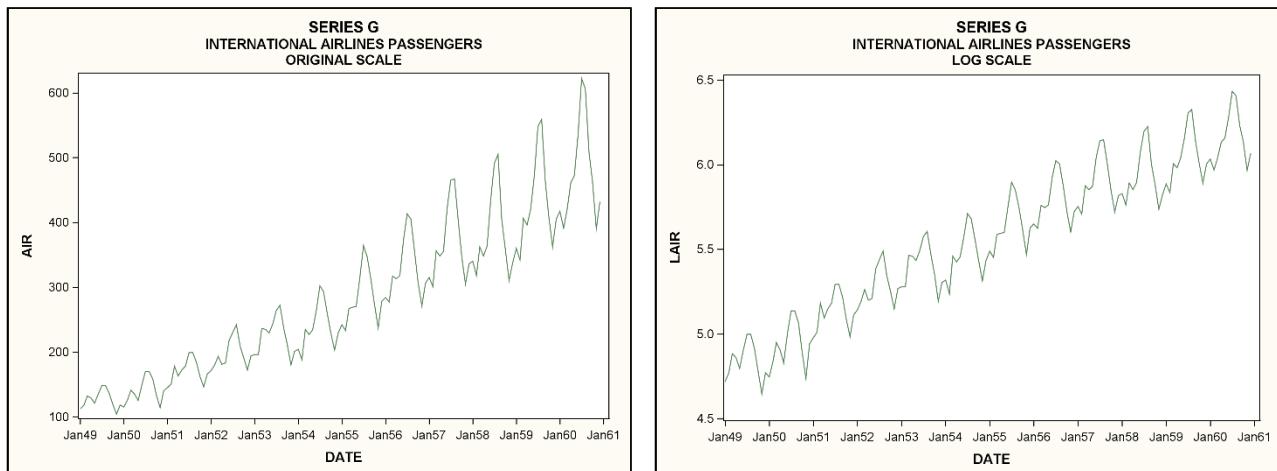
77	529.5000	30.9114	468.9148	590.0852
78	532.8000	32.4201	469.2577	596.3423
79	543.6000	33.8617	477.2323	609.9677

The following model is known as the *airline model*:

$$(1 - B)(1 - B^{12})Y_t = (1 - \theta_1 B)(1 - \theta_2 B^{12})e_t$$

Its popularity started when Box and Jenkins (1976) used it to model sales of international airline tickets on a logarithmic scale. Output 4.4 shows plots of the original and log scale data from Box and Jenkins's text.

Output 4.4: Plotting the Original and Log-Transformed Box and Jenkins Airline Data



Now, analyze the logarithms, which have the more stable seasonal pattern, using these SAS statements:

```
proc arima data=airline;
  identify var=lair;
  identify var=lair(1);
  title "SERIES G";
  title2 "INTERNATIONAL AIRLINES PASSENGERS";
run;
```

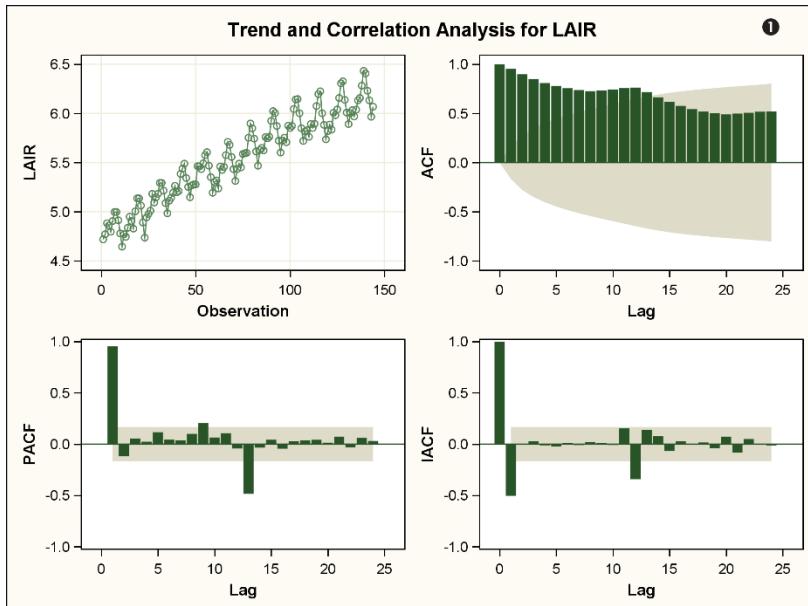
The results are shown in **Output 4.5a**.

Output 4.5a: Identifying the Logarithms with the IDENTIFY Statement: PROC ARIMA

The ARIMA Procedure

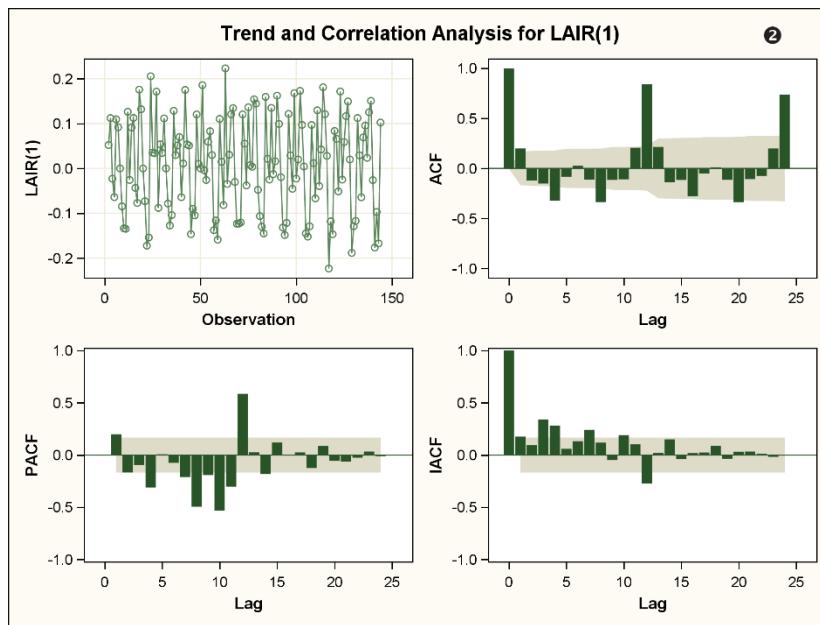
Name of Variable = LAIR	
Mean of Working Series	5.542176
Standard Deviation	0.439921
Number of Observations	144

Autocorrelation Check for White Noise									
To Lag	Chi-Square	DF	Pr > ChiSq	Autocorrelations					
6	638.37	6	<.0001	0.954	0.899	0.851	0.808	0.779	0.756
12	1157.62	12	<.0001	0.738	0.727	0.734	0.744	0.758	0.762
18	1521.94	18	<.0001	0.717	0.663	0.618	0.576	0.544	0.519
24	1785.32	24	<.0001	0.501	0.490	0.498	0.506	0.517	0.520



Name of Variable = LAIR	
Period(s) of Differencing	1
Mean of Working Series	0.00944
Standard Deviation	0.106183
Number of Observations	143
Observation(s) eliminated by differencing	1

Autocorrelation Check for White Noise									
To Lag	Chi-Square	DF	Pr > ChiSq	Autocorrelations					
6	27.95	6	<.0001	0.200	-0.120	-0.151	-0.322	-0.084	0.026
12	169.89	12	<.0001	-0.111	-0.337	-0.116	-0.109	0.206	0.841
18	195.75	18	<.0001	0.215	-0.140	-0.116	-0.279	-0.052	0.012
24	321.53	24	<.0001	-0.114	-0.337	-0.107	-0.075	0.199	0.737



It is difficult to detect seasonality in the ACF of the original series ❶ because all the values are so near 1. The slow decay is much more evident here than in the construction example. Once you take the first difference, you obtain the ACF ❷. Looking at the seasonal lags (12, 24), you see little decay, indicating that you should consider a span 12 difference. Create the following variable and its ACF, inverse autocorrelation function (IACF), and partial autocorrelation function (PACF):

$$V_t = (Y_t - Y_{t-1}) - (Y_{t-12} - Y_{t-13})$$

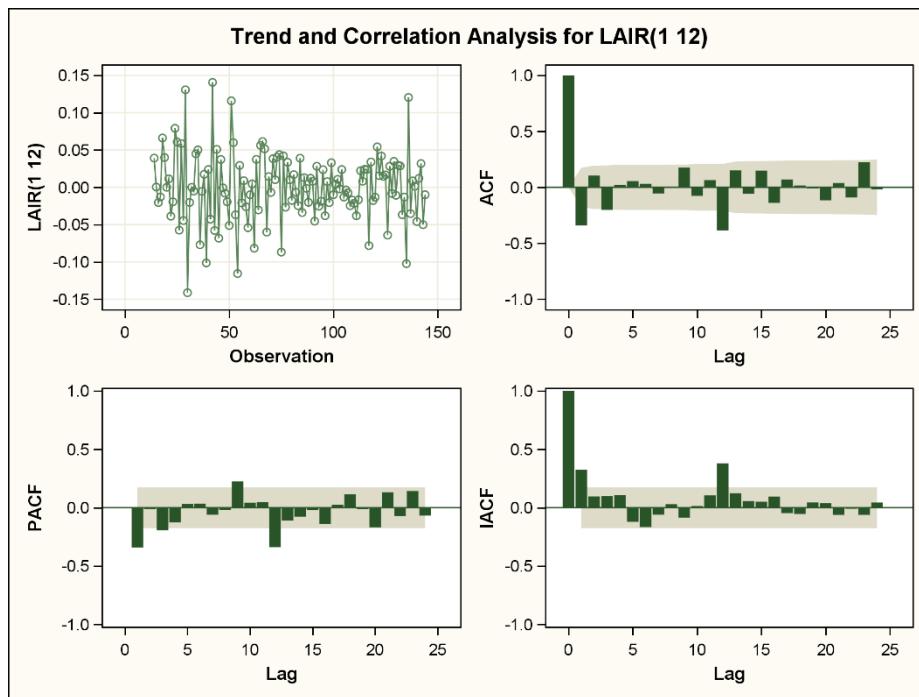
Do so by issuing the following SAS statements to generate **Output 4.5b**:

```
proc arima data=airline;
  identify var=lair(1,12);
```

Output 4.5b: Identifying the Logarithms with the IDENTIFY Statement: PROC ARIMA

Name of Variable = LAIR	
Period(s) of Differencing	1,12
Mean of Working Series	0.000291
Standard Deviation	0.045673
Number of Observations	131
Observation(s) eliminated by differencing	13

Autocorrelation Check for White Noise									
To Lag	Chi-Square	DF	Pr > ChiSq	Autocorrelations					
6	23.27	6	0.0007	-0.341	0.105	-0.202	0.021	0.056	0.031
12	51.47	12	<.0001	-0.056	-0.001	0.176	-0.076	0.064	-0.387
18	62.44	18	<.0001	0.152	-0.058	0.150	-0.139	0.070	0.016
24	74.27	24	<.0001	-0.011	-0.117	0.039	-0.091	0.223	-0.018



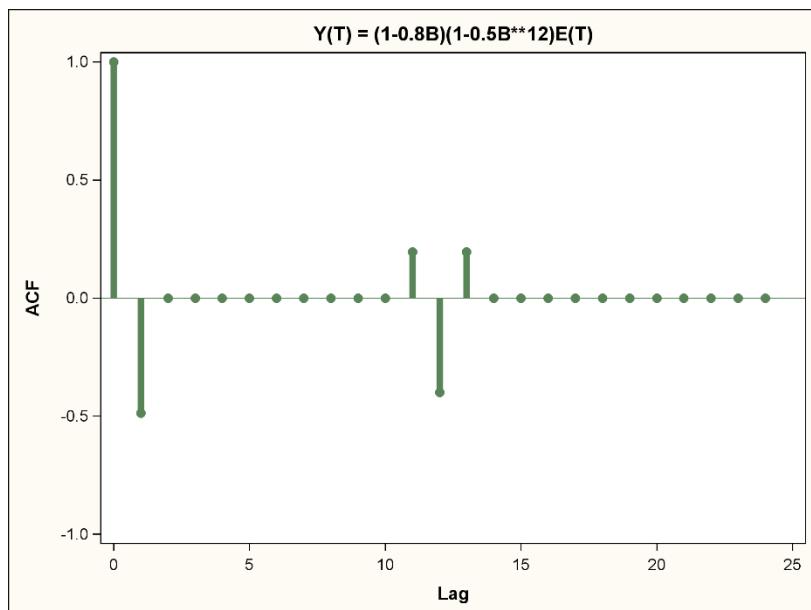
The model is identified from the autocorrelations. Identification depends on pattern recognition in the plot of the ACF values against the lags. The nonzero ACF values are called spikes to draw to mind the plots PROC ARIMA produces in the IDENTIFY stage. For the airline model, if $\theta_1 > 0$ and $\theta_2 > 0$, the theoretical autocorrelations of the series

$$V_t = (1 - \theta_1 B)(1 - \theta_2 B^{12}) e_t$$

- a (negative) spike at lag 1
- a (negative) spike at lag 12
- equal (and positive) spikes at lags 11 and 13 that are called side lobes of the lag 12 spike
- all other lag correlations 0

The pattern shown in **Output 4.5c** represents the theoretical ACF of $V_t = (1 - \theta_1 B)(1 - \theta_2 B^{12}) e_t$, or the IACF of $(1 - \theta_1 B)(1 - \theta_2 B^{12}) V_t = e_t$:

Output 4.5c: Theoretical ACF of V_t



When you compare this pattern to the ACF of the LAIR(1,12) variable, you find reasonable agreement. If the signs of the parameters are changed, the spikes and side lobes have different signs, but remain at the same lags. The spike and side lobes at the seasonal lag are characteristic of seasonal multiplicative models. If the multiplicative factor is on the AR side, this pattern appears in the IACF instead of in the ACF. In that case, the IACF and PACF behave differently, and the IACF is easier to interpret.

If the model is changed to the following, then the spike and side lobes are visible at the seasonal lag (for example, 12) and at its multiples (24, 36, and so on), but the magnitudes of the spikes at the multiples decrease exponentially at rate α :

$$V_t - \alpha V_{t-12} = (1 - \theta_1 B)(1 - \theta_2 B^{12}) e_t$$

If the decay is extremely slow, an additional seasonal difference is needed ($\alpha = 1$). If the pattern appears in the IACF, the following model is indicated:

$$(1 - \theta_1 B)(1 - \theta_2 B^{12}) V_t = (1 - \alpha B^{12}) e_t$$

The SAS code for the airline data is as follows:

```
proc arima data=airline plots=none;
  identify var=lair(1,12) noprint;
  estimate q=(1) (12) noconstant;
  forecast lead=12 out=fore id=date interval=month;
quit;

proc spectra p whitetest data=fore (where=(residual ne .)) out=resid;
  var residual;
quit;

title "FORECASTS";
proc sgplot data=fore(firstobs=120);
  band x=date lower=195 upper=u95;
  series x=date y=lair / legendlabel="ACTUAL";
  series x=date y=forecast / lineattrs=graphdata2 legendlabel="FORECAST";
  xaxis valuesformat=monyy5.;
quit;

title "PERIODOGRAM OF RESIDUALS";
proc sgplot data=resid;
  series x=freq y=p_01 / markers markerattrs=(symbol=circlefilled);
quit;
proc arima data=airline;
  identify var=lair(1,12) noprint;
  estimate q=(1) (12) noconstant;
  forecast lead=12 out=fore id=date interval=month;
run;
```

The results are shown in **Output 4.6** and **Output 4.7**.

Output 4.6: Fitting the Airline Model: PROC ARIMA

The ARIMA Procedure

Conditional Least Squares Estimation					
Parameter	Estimate	Standard Error	t Value	Approx Pr > t	Lag
MA1,1	0.37727	0.08196	4.60	<.0001	1
MA2,1	0.57236	0.07802	7.34	<.0001	12

Variance Estimate	0.00141
Std Error Estimate	0.037554
AIC	-486.133

SBC	-480.383
Number of Residuals	131

* AIC and SBC do not include log determinant.

Correlations of Parameter Estimates		
Parameter	MA1,1	MA2,1
MA1,1	1.000	-0.091
MA2,1	-0.091	1.000

Autocorrelation Check of Residuals									
To Lag	Chi-Square	DF	Pr > ChiSq	Autocorrelations					
6	5.15	4	0.2723	0.010	0.028	-0.119	-0.100	0.081	0.077
12	7.89	10	0.6400	-0.049	-0.023	0.114	-0.045	0.025	-0.023
18	11.98	16	0.7452	0.012	0.036	0.064	-0.136	0.055	0.011
24	22.56	22	0.4272	-0.098	-0.096	-0.031	-0.021	0.214	0.013

Model for variable LAIR	
Period(s) of Differencing	1,12

No mean term in this model.

Moving Average Factors	
Factor 1:	1 - 0.37727 B ^{**} (1)
Factor 2:	1 - 0.57236 B ^{**} (12)

Forecasts for variable LAIR				
Obs	Forecast	Std Error	95% Confidence Limits	
145	6.1095	0.0376	6.0359	6.1831
146	6.0536	0.0442	5.9669	6.1404
147	6.1728	0.0500	6.0747	6.2709

(Additional output omitted)

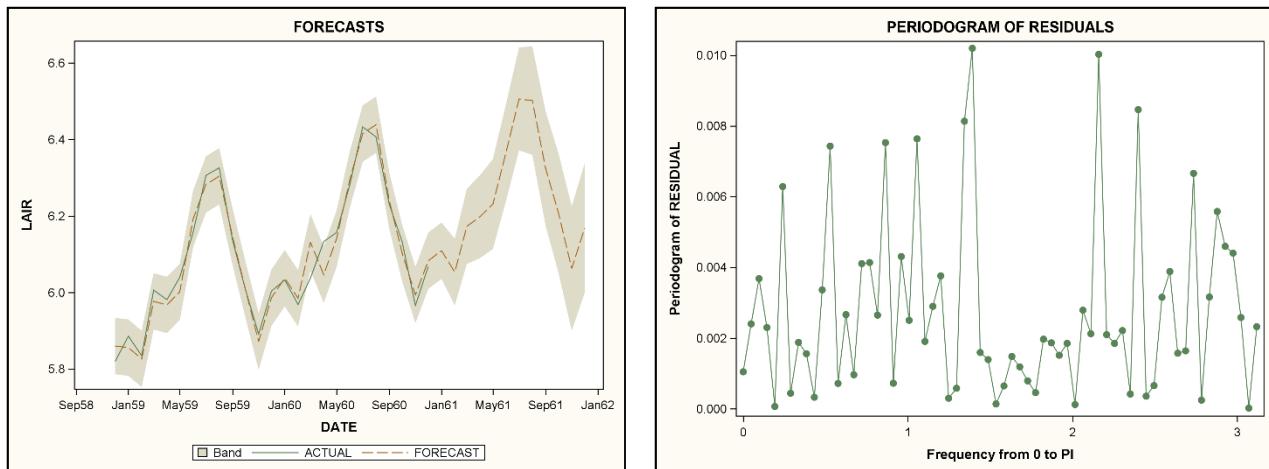
154	6.2081	0.0796	6.0521	6.3641
155	6.0631	0.0829	5.9005	6.2256
156	6.1678	0.0862	5.9989	6.3367

SERIES G INTERNATIONAL AIRLINES PASSENGERS

The SPECTRA Procedure

Test for White Noise for Variable RESIDUAL	
M =	65
Max(P[*])	0.0102
Sum(P[*])	0.181402

Fisher's Kappa: M*MAX(P ^(*))/SUM(P ^(*))	
Kappa	3.655039
Bartlett's Kolmogorov-Smirnov Statistic: Maximum absolute difference of the standardized partial sums of the periodogram and the CDF of a uniform(0,1) random variable.	
Test Statistic	0.089019
Approximate P-Value	0.6816

Output 4.7: Plotting the Forecasts and the Periodogram: PROC ARIMA and PROC SPECTRA

PROC SPECTRA is also used to search for hidden periodicities in the airline residuals. No periodicities are indicated in the periodogram plot or in the white noise tests produced by PROC SPECTRA. In general, PROC SPECTRA expresses a series as a sum of sinusoidal waves at equally spaced (Fourier) frequencies between 0 and π radians. The mathematical technique used here is referred to as a Fourier decomposition of the series. The periodogram, the right-hand graph in **Output 4.7**, is simply a plot of the sum of squares at each frequency versus the frequency. The two tests shown in **Output 4.6** test to see whether all frequencies are equally important, a condition characterizing white noise. This is analogous to the fact that white light contains equal contributions from all light spectrum frequencies. Fuller (1996) discusses the tests, neither of which is significant. The model seems to have accounted for any periodic behavior in the data. Refer to Chapter 10, "Spectral Analysis," for more information about PROC SPECTRA.

4.2 Models with Explanatory Variables

Sometimes you can improve forecasts by relating the series of interest to other explanatory variables. Obviously, forecasting in such situations requires knowledge (or at least forecasts) of future values of such variables. The nature of the explanatory variables and of the model relating them to the target series determines the optimal forecasting method. Explanatory variables are addressed in Chapter 2, "Simple Models: Autoregression." There, they are deterministic, meaning that their future values are determined without error. Seasonal indicator variables and time t are deterministic. Explanatory variables like interest rates and unemployment are not deterministic because their future values are unknown.

Chapter 2 assumes that the relationship between the target series Y_t and the explanatory series $X_{1t}, X_{2t}, \dots, X_{kt}$ satisfies the usual regression model assumptions, as follows, where e_t is white noise:

$$Y_t = \beta_0 + \beta_1 X_{1t} + \dots + \beta_k X_{kt} + e_t$$

The Durbin-Watson statistic is used in Chapter 2 to detect departures from the assumptions on e_t . The following methods are appropriate when the Durbin-Watson statistic from PROC REG or PROC GLM shows significant autocorrelation.

Recall that if the regression analysis from PROC REG or PROC GLM shows no autocorrelation and if known future values (as opposed to forecasts) are available for all X s, you can forecast with appropriate prediction intervals by performing the following tasks:

- supplying future X s and missing values (.) for future Y s
- regressing Y on the X s with the CLI option in the MODEL statement or the keywords U95= and L95= in the OUTPUT statement

This chapter combines regression with time series errors to provide a richer class of forecasting models. Three cases are delineated, presented in order of increasing complexity. Examples are included, and special cases are highlighted.

4.2.1 Case 1: Regression with Time Series Errors

The model is as follows, where Z_t is an ARIMA time series:

$$Y_t = \beta_0 + \beta_1 X_{1t} + \beta_2 X_{2t} + \dots + \beta_k X_{kt} + Z_t$$

This is a typical regression, except that you allow for autocorrelation in the error term Z . The Y series does not depend on lagged values of the X s. If the error series is purely autoregressive of order p , the following SAS code (where p would be replaced by a nonnegative integer) properly fits a model to $k = 3$ explanatory variables:

```
proc autoreg data=examp;
   model y=x1 x2 x3 / nlag=p;
run;
```

Because PROC ARIMA can do this and can also accommodate mixed models and differencing, it is used instead of PROC AUTOREG in the analyses below.

In case 1, forecasts of Y and forecast intervals are produced whenever future values of the X s are supplied. If these future X s are user-supplied forecasts, the procedure cannot incorporate the uncertainty of these future X s into the intervals around the forecasts of Y . Thus, the Y forecast intervals are too narrow. Valid intervals are produced when you supply future values of deterministic X s or when PROC ARIMA forecasts the X s in a transfer function setting as in cases 2 and 3.

4.2.2 Case 1A: Intervention

If one of the X variables is an indicator variable (each value 1 or 0), the modeling above is called *intervention analysis*. The reason for this term is that X usually changes from 0 to 1 during periods of expected change in the level of Y , such as strikes, power outages, and war. For example, suppose Y is the daily death rate from automobile accidents in the United States. Suppose that on day 50, the speed limit is reduced from 65 mph to 55 mph. Suppose you have another 100 days of data after this intervention. In that case, designate X_t as 0 before day 50 and as 1 on and following day 50. The model $Y_t = \beta_0 + \beta_1 X_t + Z_t$ explains Y in terms of two means (plus the error term). Before day 50, the mean is $\beta_0 + (\beta_1)(0) = \beta_0$ and on and following day 50, the mean is $\beta_0 + \beta_1$. Thus, β_1 is the effect of a lower speed limit, and its statistical significance can be judged based on the t test for $H_0: \beta_1 = 0$.

If the model is fit by ordinary regression, but the Z s are autocorrelated, this t test is not valid. Using PROC ARIMA to fit the model allows a valid test. Supplying future values of the deterministic X produces forecasts with valid forecast intervals.

The 1s and 0s can occur in any meaningful place in X . For example, if the speed limit reverts to 65 mph on day 70, you set X back to 0 starting on day 70.

If a data point is considered an outlier, you can use an indicator variable that is 1 only for that data point in order to eliminate its influence on the ARMA parameter estimates. Deleting the point results in a missing value (.) in the series. Closing the gap with a DELETE statement makes the lags across the gap incorrect. You can avoid these problems with the indicator variable approach. PROC ARIMA also provides an outlier detection routine. If you choose a point to consider as an outlier and supply a dummy variable based on data inspection, it is suggested that you multiply the dummy variable's p -value by the sample size n (Bonferroni correction) before comparing to your significance level (for example, $\alpha = 0.05$) as this is not a preplanned comparison.

4.2.3 Case 2: Simple Transfer Functions

In this case, the model is as follows, where X_t and Z_t are independent ARIMA processes:

$$Y_t = \beta_0 + \beta_1 X_t + Z_t$$

Because X is an ARIMA process, you can estimate a model for X in PROC ARIMA and use it to forecast future X s. The algorithm enables you to compute forecast error variances for these future X s, which are automatically incorporated later into the Y forecast intervals. However, first you must identify a model and fit it to the Z series. You accomplish this by studying the ACF, IACF, and PACF of residuals from a regression of Y on X . In fact, you can accomplish this entire procedure within PROC ARIMA. Once you have identified and fit models for X and Z , you can produce forecasts and associated intervals easily.

You can use several explanatory variables, but for proper forecasting, they should be independent of one another. If the explanatory variables contain arbitrary correlations, use the STATESPACE, SSM, or VARMAX procedure, each of which takes advantage of these correlations to produce forecast intervals.

4.2.4 Case 3: General Transfer Functions

In case 3, you allow the target series Y_t to depend on current and past values of the explanatory variable X . The model is as follows, where X and Z are independent ARIMA time series:

$$Y_t = \alpha + \sum_{j=0}^{\infty} \beta_j X_{t-j} + Z_t$$

Because it is impossible to fit an infinite number of unrestricted β s to a finite data set, you restrict the β s to have certain functional forms depending on only a few parameters. The appropriate form for a given data set is determined by an identification process for the β s that is very similar to the usual identification process with the ACFs. Instead of inspecting autocorrelations, you inspect cross-correlations. But, you are looking for the same patterns that are in univariate ARIMA modeling. The β s are called *transfer function weights* or *impulse-response weights*.

You can use several explanatory X s, but they should be independent of one another for proper forecasting and identification of the β s. Even if you can identify the model properly, correlation among explanatory variables causes incorrect forecast intervals because the procedure assumes independence when it computes forecast error variances.

Because you need forecasts of explanatory variables to forecast the target series, it is crucial that X does not depend on past values of Y . Such a dependency is called *feedback*. Feedback puts you in a circular situation where you need forecasts of X to forecast Y and forecasts of Y to forecast X . You can use PROC STATESPACE, SSM, or VARMAX to model a series with arbitrary forms of feedback and cross-correlated inputs. Strictly AR models, including feedback, can be fit by multiple regression as proved by Fuller (1996). A general approach to AR modeling by nonlinear regression is also given by Fuller (1986).

4.2.5 Case 3A: Leading Indicators

Suppose in the previous model you find as follows: $\beta_0 = \beta_1 = 0$, $\beta_2 \neq 0$. Then, Y responds two periods later to movements in X . X is called a leading indicator for Y because its movements enable you to predict movements in Y two periods ahead using observed X values. The lead of two periods is also called a *shift* or a *pure delay* in the response of Y to X . Such models are highly desirable for forecasting.

4.2.6 Case 3B: Intervention

You can use an indicator variable as input in case 3B, as was suggested in case 1A. However, you identify the pattern of the β s differently than in case 3. In case 3, cross-correlations are the key to identifying the β pattern, but in case 3B, cross-correlations are virtually useless.

4.3 Methodology and Example

Combining regression with time series gives the analyst the ability to forecast based on known or estimated future inputs and on the momentum (autocorrelation) in the errors.

4.3.1 Case 1: Regression with Time Series Errors

In this example, a manufacturer of building supplies monitors sales (S) for one of its product lines in terms of disposable income (D), U.S. housing starts (H), and mortgage rates (M). The data are obtained quarterly. Plots of the four series are given in **Output 4.8**.

The first task is to determine the differencing desired. Each series has a fairly slowly decaying ACF, and you decide to use a differenced series. Each first differenced series has an ACF consistent with the assumption of stationarity. The D series has differences that display a slight, upward trend. This trend is not of concern unless you plan to model D . Currently, you are using it just as an explanatory variable. The fact that you differenced all the series (including sales) implies an assumption about the error term. Your model in the original levels of the variable is as follows:

$$S_t = \beta_0 + \beta_1 D_t + \beta_2 H_t + \beta_3 M_t + \eta_t$$

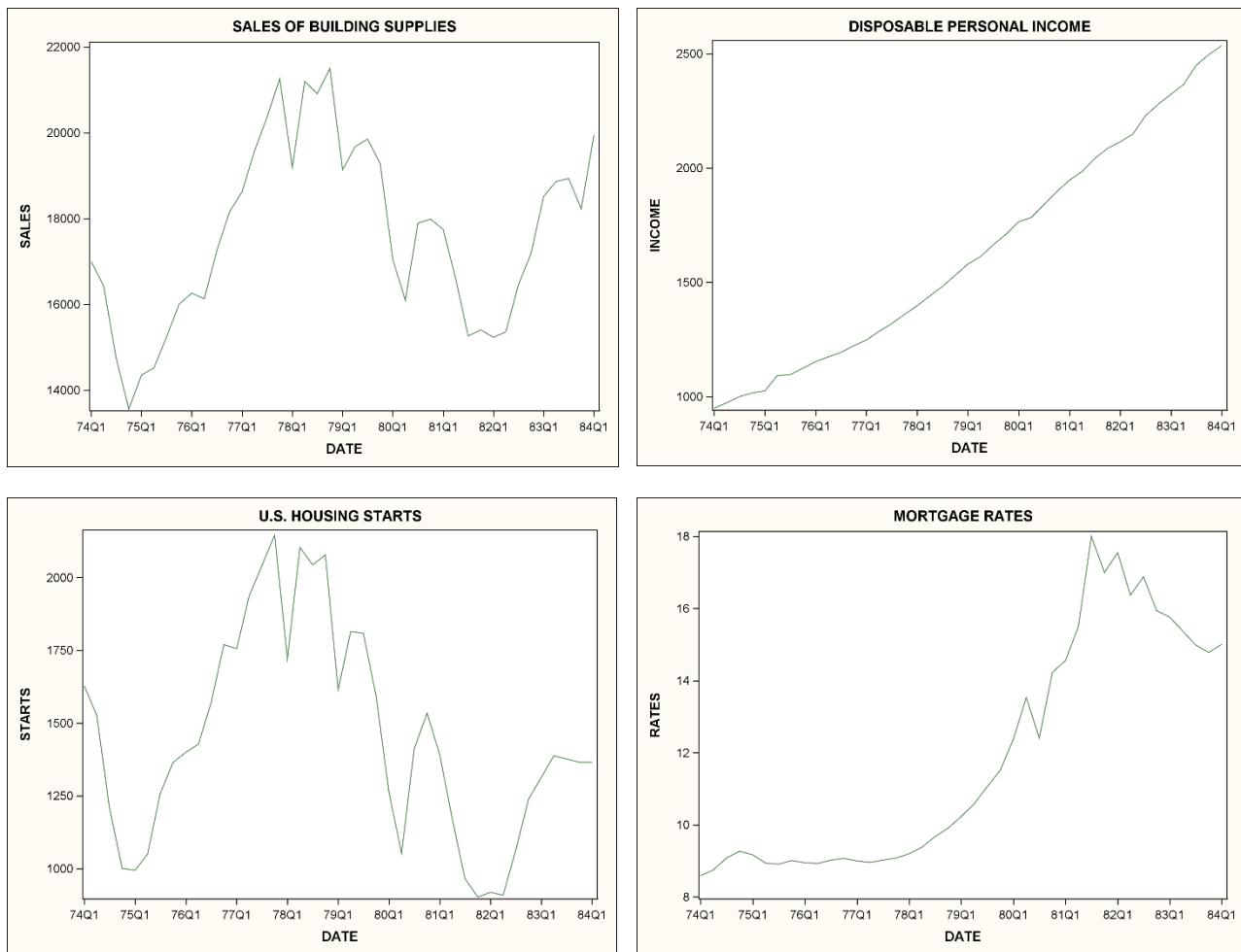
When you lag by 1, you get the following:

$$S_{t-1} = \beta_0 + \beta_1 D_{t-1} + \beta_2 H_{t-1} + \beta_3 M_{t-1} + \eta_{t-1}$$

When you subtract, you get the following:

$$\nabla S_t = 0 + \beta_1 \nabla D_t + \beta_2 \nabla H_t + \beta_3 \nabla M_t + \nabla \eta_t$$

Thus, differencing implies that η_t had a unit root nonstationarity, so the differenced error series is stationary. This assumption, unlike assumptions about the explanatory series, is crucial. If you do not want to make this assumption, you can model the series in the original levels. Also, in the development above, you assume a simple intercept β_0 that canceled out of the differenced model. If, in fact, a trend $\beta_0 + \psi t$ is present, the differenced series has intercept ψ . If you had decided to fit the model in the original levels and to allow only AR error structures, PROC AUTOREG or Fuller's PROC NLIN method (1986) would have been an appropriate tool for the fitting.

Output 4.8: Plotting Building- and Manufacturing-Related Quarterly Data

Assuming differencing is appropriate, your next task is to output the residuals from regression, and to choose a time series model for the error structure $\nabla\eta_t$. To accomplish this in PROC ARIMA, you must modify your IDENTIFY and ESTIMATE statements. The IDENTIFY statement is used to call in all explanatory variables of interest and to declare the degree of differencing for each. The CROSSCOR= option accomplishes this goal. You specify the following SAS statements:

```
proc arima data=housing;
  title 'model in first differences';
  identify var=sales(1) crosscor=(mort(1) dpic(1) starts(1)) noint;
run;
```

The NOPRINT option eliminates the printing of the cross-correlation function. Because you assume a contemporaneous relationship between sales and the explanatory variables, you do not check the cross-correlation function for dependence of sales on lagged values of the explanatory variables. If you want to check for lagged dependencies, you need to model the explanatory series to perform prewhitening. This is the only way you can get clear information from the cross-correlations.

To run a regression of SALES(1) on MORT(1), DPIC(1), and STARTS(1), add the following statement to your PROC ARIMA code:

```
estimate input=(mort dpic starts) plot method=ml;
run;
```

The INPUT= option denotes which variables in the CROSSCOR= list are to be used in the regression. Specifying differencing in the INPUT= option is not allowed. The order of differencing in the CROSSCOR= list is the order used. The PLOT option creates and plots the ACF, IACF, and PACF of the residuals. These are produced automatically if ODS graphics are turned on. The results are shown in **Output 4.9**.

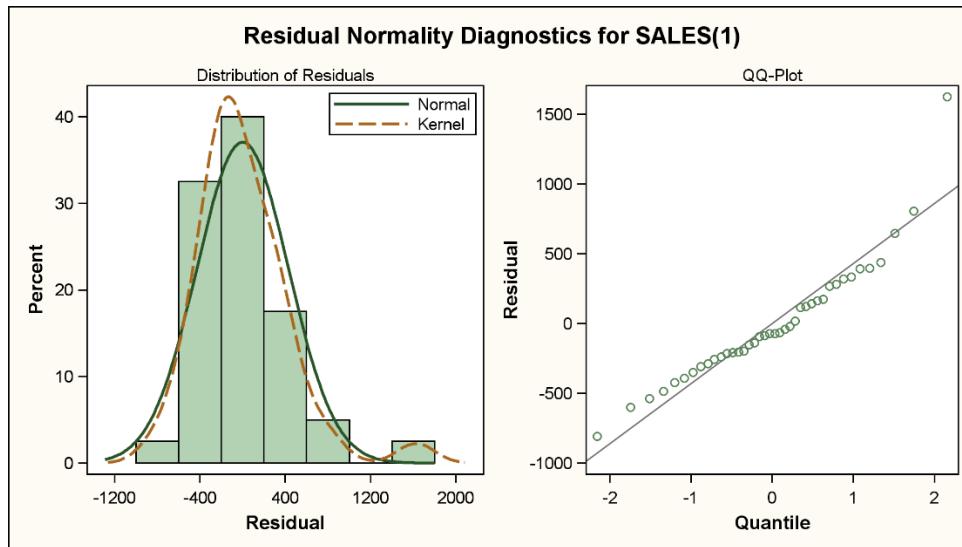
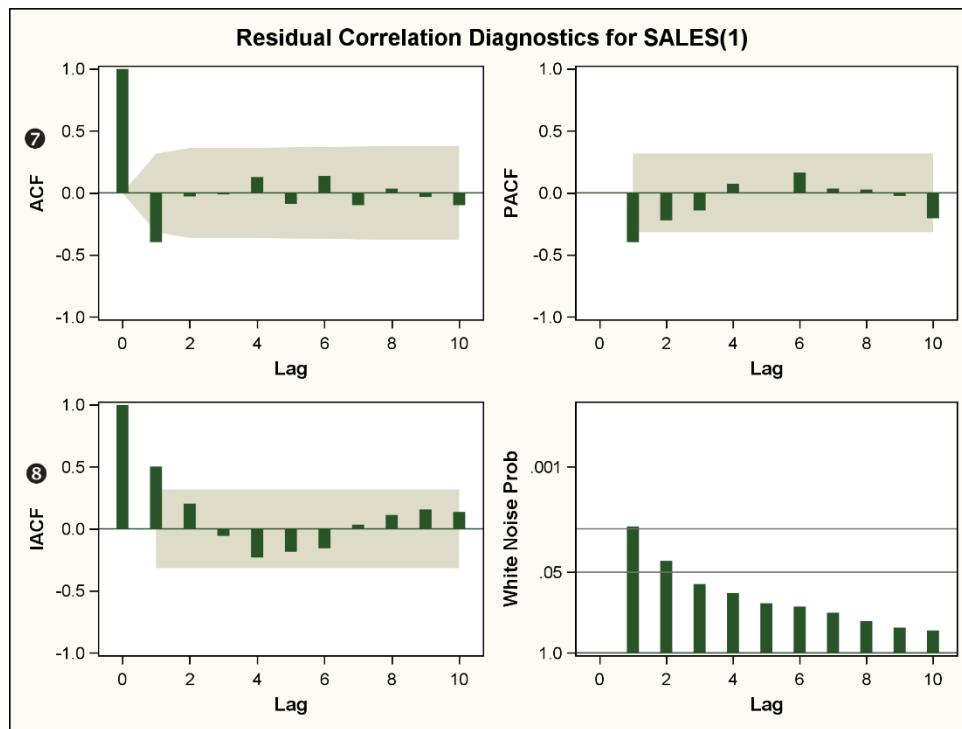
Output 4.9: Using the INPUT= Option of the ESTIMATE Statement to Run a Regression: PROC ARIMA**The ARIMA Procedure**

Maximum Likelihood Estimation							
Parameter	Estimate	Standard Error	t Value	Approx Pr > t	Lag	Variable	Shift
MU	170.03857	181.63744	0.94	0.3492	0	SALES	0
NUM1	-151.07400	112.12506	-1.35	0.1779	0	MORT	0
NUM2	-1.00212	4.22489	-0.24	0.8125	0	DPIC	0
NUM3	4.93009	0.39852	12.37	<.0001	0	STARTS	0

Constant Estimate	170.0386
Variance Estimate	200686.5
Std Error Estimate	447.9805
AIC	605.6806
SBC	612.4362
Number of Residuals	40

Correlations of Parameter Estimates					
Variable Parameter	SALES MU	MORT NUM1	DPIC NUM2	STARTS NUM3	
SALES MU	1.000	-0.028	-0.916	0.016	
MORT NUM1	-0.028	1.000	-0.071	0.349	
DPIC NUM2	-0.916	-0.071	1.000	-0.039	
STARTS NUM3	0.016	0.349	-0.039	1.000	

Autocorrelation Check of Residuals									
To Lag	Chi-Square	DF	Pr > ChiSq	Autocorrelations					
6	8.90	6	0.1793	-0.397	-0.028	-0.012	0.128	-0.088	0.138
12	10.27	12	0.5923	-0.099	0.035	-0.033	-0.099	0.053	0.026
18	16.98	18	0.5245	-0.011	-0.122	0.143	-0.189	0.014	0.156
24	23.37	24	0.4981	-0.033	-0.169	0.174	-0.043	-0.039	0.093



① Model for variable SALES	
Estimated Intercept	170.0386
Period(s) of Differencing	1

② Input Number 1	
Input Variable	MORT
Period(s) of Differencing	1
Overall Regression Factor	-151.074

③ Input Number 2	
Input Variable	DPIC
Period(s) of Differencing	1
Overall Regression Factor	-1.00212

② Input Number 3	
Input Variable	STARTS
Period(s) of Differencing	1
Overall Regression Factor	4.930094

Output from the ESTIMATE statement for the sales data indicates that sales ① are positively related to housing starts ②, but negatively related to mortgage rates ③ and disposable personal income ④. In terms of significance, only the t statistic ⑤ for housing starts exceeds 2.

However, unless you fit the correct model, the t statistics are meaningless. The correct model includes specifying the error structure, which you have not yet done. For the moment, ignore these t statistics. You might argue based on the chi-square checks ⑥ that the residuals are not autocorrelated. However, because the first chi-square statistic uses six correlations, the influence of a reasonably large correlation at lag 1 might be lessened to such an extent by the other five small correlations that significance is lost. Look separately at the first few autocorrelations, and remember that differencing is often accompanied by an MA term. Further, the graph of the white noise test does show significance at lags 1 and 2. The plot computes the Q statistic using sums of squared correlations summing up through lags 1, 2, 3, etc., rather than in groups of 6. Thus, you fit a model to the error series and wait to judge the significance of your t statistics until all important variables (including lagged error values) have been incorporated into the model. You use the same procedure here as you do in regression settings, where you do not use the t statistic for a variable in a model with an important explanatory variable omitted.

Based on the ACF of the differenced series, you fit an MA(1) model to the errors. You interpret the ACF of the differenced series as having a nonzero value (-0.39698) ⑦ at lag 1 and a near-zero value at the other lags. Also, check the IACF ⑧ to see whether you have overdifferenced the series. If you have, the IACF dies off very slowly. Suppose you decide the IACF dies off rapidly enough and that you were correct to difference. If $Y = \alpha + \beta X + \eta$, where X and η are unit root processes, regression of Y on X produces an inconsistent estimate of β . This makes it impossible for you to use the PLOT option in a model in the original levels of the series to determine whether you should difference. Residuals from the model might not resemble the true errors in the series because the estimate of β is inconsistent. Because the explanatory series seems to require differencing, you decide to model the SALES series in differences also, and then to check for overdifferencing with the PLOT option. Overdifferencing results in an MA coefficient that is an estimate of 1.

The next step is to fit the regression model with an MA error term. You can accomplish this in PROC ARIMA by replacing the ESTIMATE statement above:

```
estimate input=(mort dpic starts) q=1 method=ml;
run;
```

The results are shown in **Output 4.10**.

Output 4.10: Fitting the Regression Model with an MA Error Term: PROC ARIMA

The ARIMA Procedure

ARIMA Estimation Optimization Summary	
Estimation Method	Maximum Likelihood
Parameters Estimated	5
Termination Criteria	Maximum Relative Change in Estimates
Iteration Stopping Value	0.001
Criteria Value	27.51793
Maximum Absolute Value of Gradient	234254.1
R-Square Change from Last Iteration	0.153278
Objective Function	Log Gaussian Likelihood
Objective Function Value	-289.068
Marquardt's Lambda Coefficient	1E-8
Numerical Derivative Perturbation Delta	0.001

ARIMA Estimation Optimization Summary	
Iterations	13
Warning Message	Estimates may not have converged.

Maximum Likelihood Estimation								
Parameter	Estimate	Standard Error	t Value	Approx Pr > t	Lag	Variable	Shift	
MU	91.38149	47.30628	1.93	0.0534	0	SALES	0	
MA1,1 ①	0.99973	30.48149	0.03	0.9738	1	SALES	0	
NUM1	-202.26240	60.63966	-3.34	0.0009	0	MORT	0	
NUM2	0.89566	1.14910	0.78	0.4357	0	DPIC	0	
NUM3	5.13054	0.28083	18.27	<.0001	0	STARTS	0	

Constant Estimate	91.38149
Variance Estimate	115435.7
Std Error Estimate	339.7582
AIC	588.1353
SBC	596.5797
Number of Residuals	40

Correlations of Parameter Estimates						
Variable Parameter	SALES MU	SALES MA1,1	MORT NUM1	DPIC NUM2	STARTS NUM3	
SALES MU	1.000	0.360	-0.219	-0.958	-0.612	
SALES MA1,1	0.360	1.000	0.159	-0.402	-0.314	
MORT NUM1	-0.219	0.159	1.000	-0.052	0.651	
DPIC NUM2	-0.958	-0.402	-0.052	1.000	0.457	
STARTS NUM3	-0.612	-0.314	0.651	0.457	1.000	

Autocorrelation Check of Residuals									
To Lag	Chi-Square	DF	Pr > ChiSq	Autocorrelations					
6	2.08	5	0.8382	0.017	0.079	0.076	0.147	-0.066	0.080
12	6.50	11	0.8383	-0.128	-0.079	-0.144	-0.186	-0.037	-0.044
18	11.43	17	0.8336	-0.120	-0.183	0.024	-0.137	0.040	0.074
24	14.65	23	0.9067	-0.073	-0.045	0.108	-0.053	-0.073	0.088

Model for variable SALES	
Estimated Intercept	91.38149
Period(s) of Differencing	1

Moving Average Factors	
Factor 1:	1 - 0.99973 B**(1)

Input Number 1	
Input Variable	MORT
Period(s) of Differencing	1
Overall Regression Factor	-202.262

Input Number 2	
Input Variable	DPIC
Period(s) of Differencing	1
Overall Regression Factor	0.895659

Input Number 3	
Input Variable	STARTS
Period(s) of Differencing	1
Overall Regression Factor	5.130543

You have used the generally more accurate maximum likelihood (ML) method of estimation on the differenced series. Remember that the IDENTIFY statement determines the degree of differencing used. The MA parameter 0.99973 ① is not significant (p -value > 0.05). The calculated t statistics ② on the explanatory variables have changed from the values that they had in the regression with no model for the error series. Also, PROC AUTOREG, another SAS procedure for regression with time series errors, cannot be used here because it does not allow for differencing (a problem that can be alleviated in the DATA step, but could be very cumbersome for handling the forecasts and standard errors) and because it works only with AR error terms.

Something has happened here that can happen in practice and is worth noting. The MA parameter estimate is almost 1. A MA parameter of 1 is exactly what would be expected if the original regression model in series levels had a white noise error term. This indicates that just an ordinary regression would suffice to fit the model without any differencing being required. However, further inspection of the printout reveals that this number might not be a good estimate of the true MA parameter, this coming from the message about estimates not converging. The standard error of the MA parameter estimate is very large and is not to be trusted due to nonconvergence. Decisions made on the basis of these numbers cannot be supported.

It is worth noting that since the first edition of this book, in which the example first appeared, some relevant developments have taken place. If a regression model with stationary errors is appropriate for data in which the variables themselves appear to be nonstationary, then these errors are a stationary linear combination of nonstationary variables. The variables, independent and dependent, are said to be *cointegrated*. Tests for cointegration are available in PROC VARMAX, discussed in section 5.2 of Chapter 5. It will be seen that elimination of some seemingly unimportant input variables in the example results in a model that does not show this problem. This is the route that will be taken here. However, a test for cointegration could be used to make a more informed decision as to whether the differencing was appropriate.

Any model that can be fit in PROC AUTOREG can also be fit in PROC ARIMA, which makes PROC ARIMA more generally applicable than PROC AUTOREG. The only advantage of PROC AUTOREG in this setting is its automatic selection of an AR model.

A final modeling step is to delete insignificant explanatory variables. Do not calculate SALES forecasts based on forecasts of unrelated series. If you do, the forecast error variance is unnecessarily large because the forecast responds to fluctuations in irrelevant variables. Is it acceptable to eliminate simultaneously all variables with insignificant t statistics? No, it is not acceptable. Eliminating a single insignificant regressor, like DPIC, can change the t statistics on all remaining parameters.

In the previous example, DPIC drifts upward along with SALES. A nonzero MU in the differenced model corresponds to drift in the original levels. The t statistic on MU is currently insignificant because DPIC takes over as the explainer of drift if MU is removed. Similarly, MU takes over if DPIC is removed. However, if you remove both terms from the model, the fit deteriorates significantly. DPIC and MU have the lowest t statistics. Remove DPIC and leave MU in the model because it is much easier to forecast than DPIC.

When DPIC is removed from the INPUT= list in your ESTIMATE statement, what happens to the t test for MU? Omitting the insignificant DPIC results in a t statistic 3.86 on MU. The other t statistics change, but mortgage rates are

still not statistically significant. Removing the mortgage rates from the INPUT= list results in a fairly simple model. Review the progression of your modeling thus far:

- You noticed that the inputs and the dependent variable SALES were nonstationary.
- You checked the residuals from a regression of differenced SALES on differenced DPIC, STARTS, and MORT. The residuals seemed stationary and reasonably invertible (in other words, the IACF died down reasonably fast).
- You used the PLOT option to identify an error term model that was MA(1). This term was problematic in that its estimate was near 1, it had a huge standard error, and the estimation procedure might not have converged.
- You used *t* statistics to sequentially remove insignificant terms and obtain $\nabla S_t = \nabla H_t + \psi + e_t - \beta e_{t-1}$, where the following define the symbols:

∇ indicates a first difference

S_t is sales at time t

H_t is U.S. housing starts at time t

ψ is a constant (drift) that corresponds to the slope in a plot of the undifferenced series against time

Consider two scenarios for forecasting this series. First, suppose you are supplied with future values of housing starts H_{t+1} from some source. You incorporate these into your data set, along with missing values for the unknown future values of SALES, and you call for a forecast. You do not supply information about the forecast accuracy of future housing start values, nor can the procedure use such information. It simply treats these futures as known values. In the second scenario, you model housing starts, and then forecast them from within PROC ARIMA. This provides an example of a case 2 problem.

For the first scenario, suppose you have been given future values of U.S. housing starts (the values are actually those that would be forecast from PROC ARIMA, giving you an opportunity to see the effect of treating forecasts as perfectly known values). The first step is to create a data set with future values for DATE and STARTS and missing values for SALES. This data set is concatenated to the original data set. The combined data set COMB has eight values of future STARTS. Use the following SAS statements:

```
proc arima data=comb;
  identify var=sales(1) crosscor=(starts(1)) noprint;
  estimate q=1 input=(starts) method=ml;
  forecast lead=8 id=date interval=qtr out=for1;
  title 'DATA WITH FORECASTS OF STARTS APPENDED AND SALES=.';
run;
```

The results are shown in **Output 4.11**.

Output 4.11: Forecasting with Future Input Values and Missing Future Sales Values

The ARIMA Procedure

Maximum Likelihood Estimation							
Parameter	Estimate	Standard Error	t Value	Approx Pr > t	Lag	Variable	Shift
MU	91.99669	25.60324	3.59	0.0003	0	SALES	0
MA1,1 Θ	0.60397	0.15203	3.97	<.0001	1	SALES	0
NUM1	5.45100	0.26085	20.90	<.0001	0	STARTS	0

Constant Estimate	91.99669
Variance Estimate	152042.2
Std Error Estimate	389.9259
AIC	594.1269
SBC	599.1936
Number of Residuals	40

Correlations of Parameter Estimates				
Variable Parameter		SALES MU	SALES MA1,1	STARTS NUM1
SALES MU		1.000	-0.170	0.010
SALES MA1,1		-0.170	1.000	0.068
STARTS NUM1		0.010	0.068	1.000

Autocorrelation Check of Residuals									
To Lag	Chi-Square	DF	Pr > ChiSq	Autocorrelations					
6	2.07	5	0.8388	-0.100	0.030	0.142	0.112	-0.044	0.011
12	6.81	11	0.8140	-0.134	-0.057	-0.092	-0.229	-0.061	-0.029
18	11.77	17	0.8140	-0.187	-0.130	0.007	-0.078	0.059	0.115
24	19.29	23	0.6844	-0.025	0.051	0.218	-0.010	-0.072	0.155

Model for variable SALES	
Estimated Intercept	91.99669
Period(s) of Differencing	1

Moving Average Factors	
Factor 1:	1 - 0.60397 B**(1)

Input Number 1	
Input Variable	STARTS
Period(s) of Differencing	1
Overall Regression Factor	5.451005

Forecasts for variable SALES				
Obs	Forecast	Std Error	95% Confidence Limits	
42	19322.8041	389.9259	18558.5634	20087.0447
43	19493.4413	419.3912	18671.4496	20315.4329
44	19484.7198	446.9181	18608.7764	20360.6631
45	19576.7165	472.8452	18649.9570	20503.4759
46	19679.9629	497.4227	18705.0324	20654.8935
47	19794.3720	520.8417	18773.5409	20815.2030
48	19857.6642	543.2521	18792.9096	20922.4189
49	19949.6609	564.7740	18842.7242	21056.5976

The final MA estimate 0.60397 is not particularly close to 1, giving you some confidence that you have not overdifferenced. No convergence problems remain at this point.

The estimation is exactly the same as in the original data set because SALES has missing values for all future quarters. These points cannot be used in the estimation. Because future values are available for all inputs, forecasts are generated. A request of LEAD=10 gives only eight forecasts because only eight future STARTS are supplied. Future values were supplied to and not generated by the procedure. Forecast intervals are valid if you can guarantee the future values supplied for housing starts. Otherwise, they are too small. Section 4.3.2 displays a plot of the forecasts from this procedure and displays a similar plot in which PROC ARIMA is used to forecast the input variable. See **Output 4.13**.

Predicted SALES are the same (recall that future values of STARTS in this example are the same as those produced in PROC ARIMA), but forecast intervals differ considerably. The general increase in predicted SALES is caused by including the drift term ψ .

4.3.2 Case 2: Simple Transfer Functions

In case 2, housing starts H_t are used as an explanatory variable for a company's sales. Using fitting and diagnostic checking, you obtain the model $\nabla S_t = \psi + \beta \nabla H_t + \eta_t$, where η is the moving average $\eta_t = e_t - \theta e_{t-1}$.

In case 1, you supplied future values of H_t to PROC ARIMA and obtained forecasts and forecast intervals. The forecasts were valid, but the intervals were not large enough because future values of housing starts were forecasts. In addition, you have the problem of obtaining these future values for housing starts. PROC ARIMA correctly incorporates the uncertainty of future housing start values into the sales forecast.

Step 1 in this methodology identifies and estimates a model for the explanatory variable H_t , U.S. housing starts. The data are quarterly and, based on the usual criteria, the series should be differenced. The differenced series ∇H_t shows some correlation at lag 4, but not enough to warrant a span 4 difference. Use an AR factor to handle the seasonality of this series.

Diagnostic checking was done on the STARTS series H_t . The model, as follows, fits well:

$$(1 - \alpha B^4) \nabla H_t = (1 - \theta B^3) e_t$$

In section 4.3.1, the series was forecast eight periods ahead to obtain future values. You do not need to request forecasts of your inputs (explanatory series) if your goal is only to forecast target series (SALES, in this case). The procedure automatically generates forecasts of inputs that it needs, but you do not see them unless you request them.

In step 2, an input series is used in an input option to identify and estimate a model for the target series S_t . This part of the SAS code is the same as it was in the previous example. The two steps must be together in a single PROC ARIMA segment. The entire set of code follows. Some of the output is shown in Output 4.12. Some of the output has been suppressed (because it was displayed earlier). Also, forecast intervals are wider than in case 1, where forecasts of H_t were taken from this run and concatenated to the end of the data set instead of being forecast by the procedure. This made it impossible to incorporate forecast errors for H_t into the forecast of S_t . Here is the SAS code:

```
proc arima data=housing;
  title 'FORECASTING STARTS AND SALES';
  identify var=starts(1) noprint;
  estimate p=(4) q=(3) method=ml noconstant;
  forecast lead=8;
  identify var=sales(1) crosscor=(starts(1)) noprint;
  estimate q=1 input=(starts) method=ml noprint;
  forecast lead=8 id=date interval=qtr out=for2 noprint;
run;
```

Output 4.12: Estimating with Use of Maximum Likelihood: PROC ARIMA

The ARIMA Procedure

Maximum Likelihood Estimation					
Parameter	Estimate	Standard Error	t Value	Approx Pr > t	Lag
MA1,1	0.42332	0.15283	2.77	0.0056	3
AR1,1	0.28500	0.15582	1.83	0.0674	4

Variance Estimate	30360.5
Std Error Estimate	174.2426
AIC	529.229
SBC	532.6068
Number of Residuals	40

Correlations of Parameter Estimates		
Parameter	MA1,1	AR1,1
MA1,1	1.000	0.193
AR1,1	0.193	1.000

Autocorrelation Check of Residuals									
To Lag	Chi-Square	DF	Pr > ChiSq	Autocorrelations					
6	2.55	4	0.6351	0.197	0.106	0.034	0.053	0.004	-0.063
12	9.90	10	0.4496	-0.021	-0.019	-0.033	-0.232	-0.167	-0.208
18	14.58	16	0.5558	-0.153	-0.117	-0.114	-0.116	-0.058	-0.060
24	18.62	22	0.6686	-0.096	0.118	0.109	0.068	0.066	-0.047

Model for variable STARTS	
Period(s) of Differencing	1

No mean term in this model.

Autoregressive Factors	
Factor 1:	1 - 0.285 B**(4)

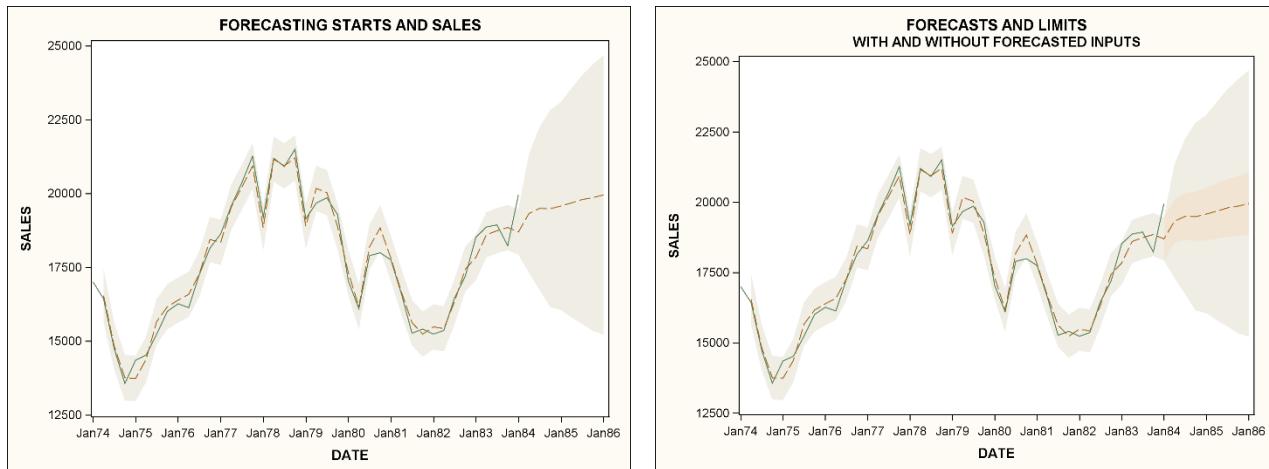
Moving Average Factors	
Factor 1:	1 - 0.42332 B**(3)

Forecasts for variable STARTS				
Obs	Forecast	Std Error	95% Confidence Limits	
42	1373.2426	174.2426	1031.7333	1714.7519
43	1387.6693	246.4163	904.7022	1870.6364
44	1369.1927	301.7971	777.6812	1960.7041
45	1369.1927	318.0853	745.7569	1992.6284
46	1371.2568	351.7394	681.8602	2060.6534
47	1375.3683	382.4434	625.7930	2124.9437
48	1370.1026	410.8593	564.8332	2175.3719
49	1370.1026	430.6707	526.0035	2214.2016

You can now merge the data sets FOR1 and FOR2 from the previous two examples and plot the forecasts and intervals on the same graph. This is illustrated in **Output 4.13** to indicate the difference in interval widths for these data.

The first plot gives forecast intervals that arose from using PROC ARIMA to forecast housing starts. The second plot adds a narrower forecast band that results from using the same future values for housing starts as given (assumed known) input rather than being forecast by PROC ARIMA. The inner interval drastically underestimates the uncertainty in the forecasts.

Output 4.13: Plotting Forecast Intervals



4.3.3 Case 3: General Transfer Functions

As introduced in section 4.2.4, in case 3 the target series can depend on the past values of the input series, as well as on the current value. Modeling strategy with general transfer function is illustrated here with real-world examples.

Model Identification

You have specified the ARMA model with backshift operators. For example, you can write the ARMA(1,1) model $Y_t - \alpha Y_{t-1} = e_t - \theta e_{t-1}$ as the following:

$$(1 - \alpha B) Y_t = (1 - \theta B) e_t$$

$$Y_t = (1 - \theta B) / (1 - \alpha B) e_t$$

or

$$Y_t = e_t + (\alpha - \theta) e_{t-1} + \alpha(\alpha - \theta) e_{t-2} + \alpha^2(\alpha - \theta) e_{t-3} + \dots$$

The pattern of the weights (coefficients on the e_t s) determines that the process has one AR and one MA parameter in the same way the ACF does. For example, if the following model holds, then the weights are 1, 1.2, 0.6, 0.3, 0.15, 0.075, ...:

$$Y_t = e_t + 1.2e_{t-1} + 0.6e_{t-2} + 0.3e_{t-3} + 0.15e_{t-4} + 0.075e_{t-5} + \dots$$

The pattern is characterized by one arbitrary change (from 1 to 1.2), followed by exponential decay at the rate 0.5 (0.6 = (0.5)(1.2), 0.3 = (0.5)(0.6), ...). The exponential decay tells you to put a factor $(1 - 0.5B)$ in the denominator of the expression multiplying e_t (in other words, $\alpha = 0.5$). Because $1.2 = \alpha - \theta$, you see that $\theta = -0.7$. The model, then, is $Y_t = (1 + 0.7B) / (1 - 0.5B) e_t$.

What have you learned from this exercise? First, you can write any ARMA model by setting Y_t equal to a ratio of polynomial factors in the backshift operator B operating on e_t . Next, you see that if you can estimate the sequence of weights on the e_t s, you can determine how many AR and MA lags you need. Finally, in this representation, the numerator polynomial corresponds to MA factors and the denominator corresponds to AR factors.

If you can apply a ratio of backshift polynomials to an unobserved error series e_t , why not apply one to an observable input? This is exactly what you do in case 3. For example, suppose you write the following expression, where η_t is the moving average $\eta_t = e_t + .6e_{t-1}$:

$$Y_t - 0.8Y_{t-1} = 3(X_{t-1} - 0.4X_{t-2}) + \eta_t$$

You then obtain the following:

$$(1 - .08B)Y_t = 3(1 - 0.4B)X_{t-1} + (1 + 0.6B)e_t$$

or

$$Y_t = 0 + 3(1 - 0.4B)/(1 - 0.8B)X_{t-1} + (1 + 0.6B)/(1 - 0.8B)e_t$$

This is called a *transfer function*. Y_t is modeled as a function of lagged values of the input series X_t and current and lagged values of the shocks e_t . Usually the intercept is not 0, although for simplicity, 0 is used in the preceding example.

You now have a potentially useful model, but how is it used? With real data, how will you know the form of the backshift expression that multiplies X_{t-1} ? The answer is in the cross-correlations. Define the cross-covariance as $\gamma_{xy}(j) = \text{cov}(Y_t, X_{t+j})$ and $\gamma_{yy}(j) = \text{cov}(X_t, Y_{t+j}) = \text{cov}(X_{t-j}, Y_t) = \gamma_{yx}(-j)$.

Estimate $\gamma_{xy}(j)$ by the following:

$$C_{xy}(j) = \sum (X_t - \bar{X})(Y_{t+j} - \bar{Y})/n$$

Define the cross-correlation as follows:

$$\rho_{xy}(j) = \gamma_{xy}(j)/(\gamma_{xx}(0)\gamma_{yy}(0))^{1/2}$$

Then estimate ρ_{xy} by the following:

$$r_{xy}(j) = C_{xy}(j)/(\gamma_{xx}(0)\gamma_{yy}(0))^{1/2}$$

To illustrate the theoretical cross-covariances for a transfer function, assume that X_t is a white noise process independent of the error series η_t . The cross-covariances are computed below and are direct multiples of $\gamma_{xx}(0)$, the variance of X . (This holds only when X is white noise.)

$$Y_t - 0.8Y_{t-1} = 3X_{t-1} - 1.2X_{t-2} + \eta_t$$

and

$$Y_t = 3X_{t-1} + 1.2X_{t-2} + 0.96X_{t-3} + 0.768X_{t-4} + \dots + \text{noise}$$

Multiplying both sides by X_{t-j} , $j = 0, 1, 2, 3$, and computing expected values, gives the following:

$$\begin{aligned}\gamma_{xy}(0) &= E(X_t Y_t) = 0 \\ \gamma_{xy}(1) &= E(X_t Y_{t+1}) = E(X_{t-1} Y_t) = 3\gamma_{xx}(0) \\ \gamma_{xy}(2) &= E(X_t Y_{t+2}) = E(X_{t-2} Y_t) = 1.2\gamma_{xx}(0)\end{aligned}$$

and

$$\gamma_{xy}(3) = E(X_t Y_{t+3}) = .96\gamma_{xx}(0)$$

When you divide each term in the cross-covariance sequence by $\gamma_{xx}(0)$, you obtain the following weights:

Lag J	-1	0	1	2	3	4	5
Weight β_j	0	0	3	1.2	9.6	$(1.2)(0.8)^2$	$(1.2)(0.8)^3$

The model involves the following:

$$3(1 - 0.4B)(1 - 0.8B)X_{t-1} = 0X_t + 3X_{t-1} + 1.2X_{t-2} + 0.96X_{t-3} + \dots$$

If X is white noise, then the cross-covariances are proportional to the transfer function weights β_j . These weights β_j are known as the *impulse-response function*. The reason for this name is clear if you ignore the error term in the model and let X_t be a pulse. That is, $X_t = 0$ except at $t = 10$, where $X_{10} = 1$. Ignoring the white noise term, you have the following:

$$Y_t = 3X_{t-1} + 1.2X_{t-2} + 0.96X_{t-3} + \dots$$

This gives the following:

$$\begin{aligned} Y_0 &= Y_1 = Y_2 = \dots = Y_{10} = 0 \\ Y_{11} &= 3X_{10} + 1.2X_9 + 0.96X_8 + \dots = 3 \\ Y_{12} &= 3(0) + 1.2(1) + 0.96(0) + \dots = 1.2 \end{aligned}$$

and $Y_{13} = 0.96$.

The weights are the expected responses to a pulse input. The pulse is delayed by one period. Its effect continues to be felt, starting with $t = 11$, but diminishes quickly because of the stationary denominator (in other words, AR-type operator $(1 - 0.8B)^{-1}$ on X_{t-1}).

The crucial point is that if you can obtain the cross-correlations, you have the impulse-response weight pattern, which you can then analyze by the same rules used for the ACFs. In the example just shown, the 0 on X_t indicates a pure delay. The arbitrary jump from 3 to 1.2, followed by exponential decay at rate 0.8, indicate that the multiplier on X_{t-1} has one numerator (MA) lag and one denominator (AR) lag. The only problem is the requirement that X_t be white noise, which is unlikely in practice and is addressed next.

Suppose you have the same transfer function, but X_t is AR(1) with parameter α . You have the following:

$$Y_t = 0 + 3(1 - 0.4B)(1 - 0.8B)X_{t-1} + (1 + 0.6B)(1 - 0.8B)e_t$$

Here, $X_t = \alpha X_{t-1} + \varepsilon_t$, where the X_t s are independent of the e_t s, and where e_t and ε_t are two (independent) white noise sequences.

Note that $Y_t = 3X_{t-1} + 1.2X_{t-2} + 0.96X_{t-3} + \dots + \text{Noise}_t$, so $\alpha Y_{t-1} = 3\alpha X_{t-2} + 1.2\alpha X_{t-3} + 0.96\alpha X_{t-4} + \dots + \alpha(\text{Noise}_{t-1})$.

Consequently, $Y_t - \alpha Y_{t-1} = 3(X_{t-1} - \alpha X_{t-2}) + 1.2(X_{t-2} - \alpha X_{t-3}) + 0.96(X_{t-3} - \alpha X_{t-4}) + \dots + N'_t$, where N'_t is a noise term.

Set $Y_t - \alpha Y_{t-1} = Y'_t$ and note that $X_t - \alpha X_{t-1} = \varepsilon_t$ is a white noise sequence, so the expression becomes the following:

$$Y'_t = 3\varepsilon_{t-1} + 1.2\varepsilon_{t-2} + 0.96\varepsilon_{t-3} + \dots + N'_t$$

The impulse-response function is exactly what you want, and $X_t - \alpha X_{t-1} = \varepsilon_t$ is a white noise sequence.

You want to model X and use that model to estimate Y'_t and ε_t . This process is known as *prewhitening*, although it really whitens only X . Next, compute the cross-correlations of the prewhitened X and Y (in other words, the estimated Y'_t and ε_t). The prewhitened variables are used only to compute the cross-correlations. The parameter estimation in PROC ARIMA is always performed on the original variables.

Statements for Transfer Function Modeling in the IDENTIFY Stage

Use the IDENTIFY and ESTIMATE statements in PROC ARIMA to model X . A subsequent IDENTIFY statement for Y with the CROSSCOR=(X) option automatically prewhitens X and Y , using the previously estimated model for X . For this example, you specify the following SAS statements:

```
proc arima data=transfer;
  title 'FITTING A TRANSFER FUNCTION';
  identify var=x;
```

```
estimate p=1;
identify var=y crosscor=(x) nlag=10;
run;
```

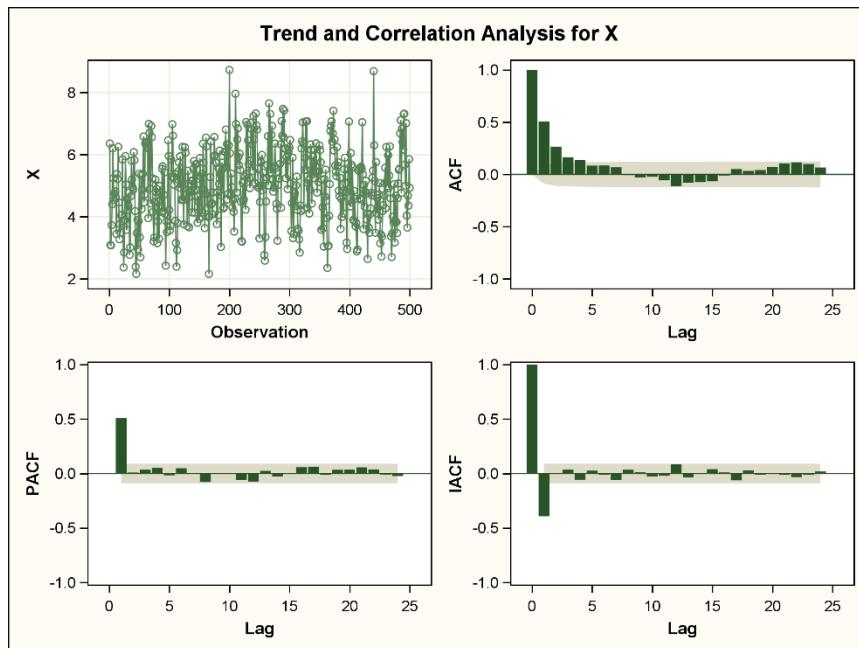
The results are shown in **Output 4.14**.

Output 4.14: Fitting a Transfer Function with the IDENTIFY and ESTIMATE Statements: PROC ARIMA

The ARIMA Procedure

Name of Variable = X	
Mean of Working Series	5.000159
Standard Deviation	1.168982
Number of Observations	500

Autocorrelation Check for White Noise									
To Lag	Chi-Square	DF	Pr > ChiSq	Autocorrelations					
6	196.16	6	<.0001	0.509	0.265	0.165	0.137	0.084	0.086
12	207.41	12	<.0001	0.069	-0.007	-0.028	-0.021	-0.057	-0.112
18	217.70	18	<.0001	-0.078	-0.074	-0.066	-0.011	0.051	0.033
24	241.42	24	<.0001	0.041	0.071	0.104	0.115	0.100	0.066



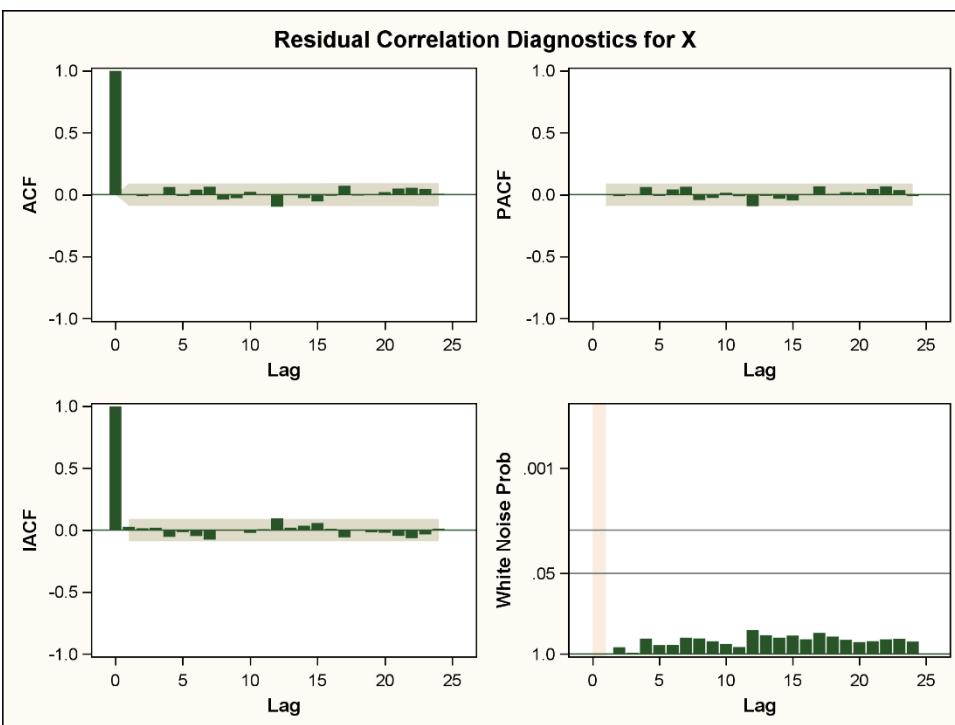
Conditional Least Squares Estimation					
Parameter	Estimate	Standard Error	t Value	Approx Pr > t	Lag
MU	5.00569	0.09149	54.71	<.0001	0
AR1,1	0.50854	0.03858	13.18	<.0001	1

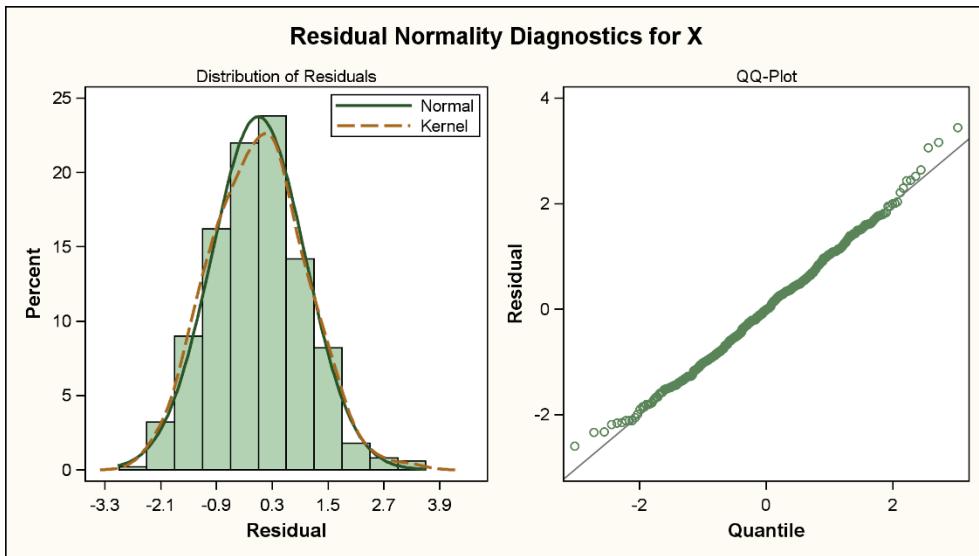
Constant Estimate	2.460094
Variance Estimate	1.017213
Std Error Estimate	1.00857
AIC	1429.468
SBC	1437.897
Number of Residuals	500

* AIC and SBC do not include log determinant.

Correlations of Parameter Estimates		
Parameter	MU	AR1,1
MU	1.000	0.005
AR1,1	0.005	1.000

Autocorrelation Check of Residuals										
To Lag	Chi-Square	DF	Pr > ChiSq	Autocorrelations						
6	2.92	5	0.7120	-0.005	-0.012	0.004	0.062	-0.010	0.041	
12	11.37	11	0.4127	0.063	-0.041	-0.028	0.022	-0.006	-0.097	
18	15.99	17	0.5246	-0.005	-0.027	-0.054	-0.008	0.072	-0.007	
24	20.23	23	0.6278	-0.001	0.020	0.050	0.055	0.046	0.008	
30	24.29	29	0.7144	-0.013	0.050	0.042	-0.011	-0.005	0.055	
36	26.44	35	0.8506	0.016	-0.030	-0.014	-0.023	0.045	0.011	
42	29.67	41	0.9055	-0.033	0.043	0.012	0.017	-0.009	0.049	
48	34.37	47	0.9148	0.025	-0.021	0.018	0.024	0.038	0.071	





Model for variable X	
Estimated Mean	5.005691

Autoregressive Factors	
Factor 1:	$1 - 0.50854 B^{**}(1)$

Name of Variable = Y	
Mean of Working Series	10.05915
Standard Deviation	6.141561
Number of Observations	500

Autocorrelation Check for White Noise								
To Lag	Chi-Square	DF	Pr > ChiSq	Autocorrelations				
6	1103.03	6	<.0001	0.856	0.720	0.607	0.514	0.422
								0.351

Correlation of Y and X								
Number of Observations				Autocorrelations				
500								
Variance of transformed series Y				14.62306				
Variance of transformed series X				1.013152				

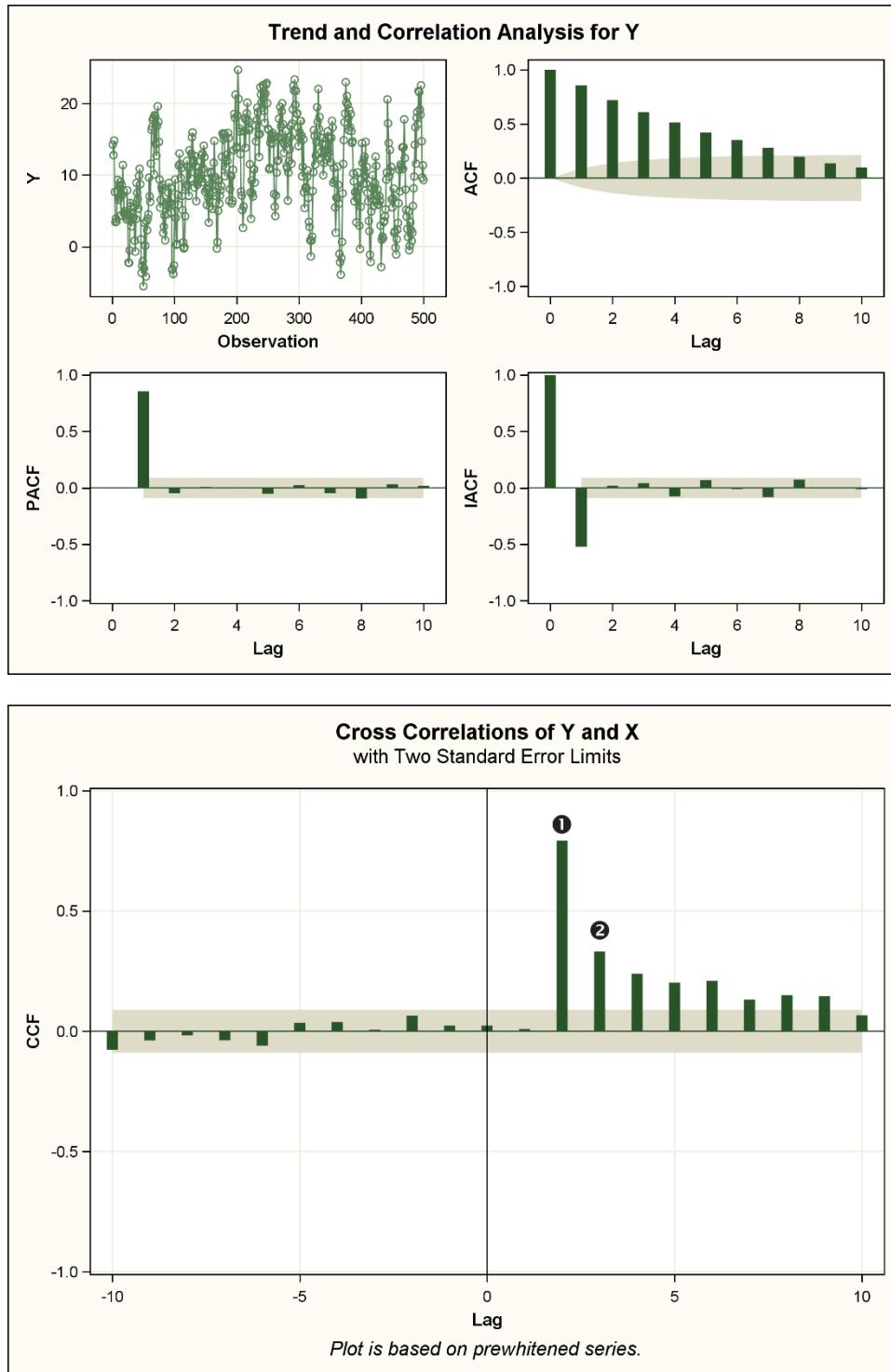
Both series have been prewhitened.

Crosscorrelation Check Between Series								
To Lag	Chi-Square	DF	Pr > ChiSq	Crosscorrelations				
5	418.85	6	<.0001	0.023	0.009	0.793	0.331	0.239
								0.202

Both variables have been prewhitened by the following filter:

Prewitening Filter

Autoregressive Factors	
Factor 1:	$1 - 0.50854 B^{**}(1)$



Data for this example are generated from the following model:

$$(Y_t - 10) - 0.8(Y_{t-1} - 10) = 3((X_{t-2} - 5) - 0.4(X_{t-3} - 5)) + N_t$$

Here, $(X_t - 5) = 0.5(X_{t-1} - 5) + e_t$.

The cross-correlations are near 0 until you reach lag 2. You see a spike (0.79327) ① followed by an arbitrary drop to 0.33146 ②, followed by an approximately exponential decay. The one arbitrary drop corresponds to one numerator

(MA) lag $(1 - \theta B)$ and the exponential decay corresponds to one denominator (AR) lag $(1 - \alpha B)$. The form of the transfer function is then as follows:

$$C(1 - \theta B)(1 - \alpha B)X_{t-2} = (C - (C\theta)B)(1 - \alpha B)X_{t-2}$$

Note the pure delay of two periods. The default in PROC ARIMA is to estimate the model with the C multiplied through the numerator (as shown on the right). The ALTPARM option gives the factored C form (as shown on the left).

Now review the PROC ARIMA instructions needed to run this example. In INPUT=(*form1 variable1 form2 variable2...*), the specification for the transfer function form is the following:

$$S\$((L_{1,1}, L_{1,2}, \dots) \dots (L_{k,1} \dots)) / ((L_{k+1,1} \dots) \dots (\dots))$$

Here, the following characteristics apply:

- S is the shift or pure delay (2 in the example).
- Lag polynomials are written in multiplicative form.
- Variable j is not followed by differencing numbers. (This is done in CROSSCOR.)

For example, INPUT=(2\$(1,3)(1)/(1)x) ALTPARM indicates the following:

$$Y_t = \theta_0 + (C(1 - \theta_1 B - \theta_2 B^3)(1 - \alpha B)(1 - \delta B))X_{t-2} + \text{Noise}$$

Several numerator and denominator factors can be multiplied together. Note the absence of a transfer function form in the sales and housing starts example, which assumes that only contemporaneous relationships exist among sales, S_t , and the input variables.

For the current (generated data) example, the transfer function form should indicate a pure delay of two (2\$), one numerator (MA) lag (2\$(1)) and one denominator lag (2\$(1)/(1)). Use the automatically generated ODS graphics to analyze the residuals, and then estimate the transfer function with the noise model. The PLOT option, formerly required to get the plots, is not required in recent SAS releases as long as the (default) ODS graphics have not been turned off.

To continue with the generated data, add these SAS statements to statements used previously to identify and estimate the X model and to identify the Y model:

```
estimate input=(2$(1)/(1)x) maxit=30 altparm method=ml;
```

This code produces **Output 4.15**. Note the AR(1) nature of the autocorrelation plot of residuals. Continue with the following code to produce **Output 4.16**:

```
proc arima data=transfer;
  identify var=x noprint;
  estimate p=1 noprint;
  identify var=y crosscor=x nlag=10 noprint;
  estimate p=1 input=(2$(1)/(1)x) printall altparm method=ml;
  forecast lead=10 out=outdata (where=(t>480)) id=t;
quit;

title "FORECAST OUTPUT DATA SET";
proc print data=outdata;
run;

data next;
  set outdata;
  if t>480;
run;

proc print data=next;
  title 'FORECAST OUTPUT DATA SET';
run;

proc sgplot data=outdata noautolegend;
  band x=t lower=195 upper=u95 /
  legendlabel="95% Forecast Band" fillatrrs=graphconfidence
```

```

transparency=0.5 fill outline;
series x=t y=y / markers;
series x=t y=forecast / lineattrs=graphdata2;
quit;

```

Model Evaluation

The estimated model is as follows:

$$Y_t = -32.46 + 2.99(1 - 0.78B)^{-1}(1 - 0.37B)X_{t-2} + (1 - 0.79B)^{-1}\eta_t$$

Results are given in **Output 4.15** and **Output 4.16**.

Output 4.15: Fitting a Transfer Function: PROC ARIMA

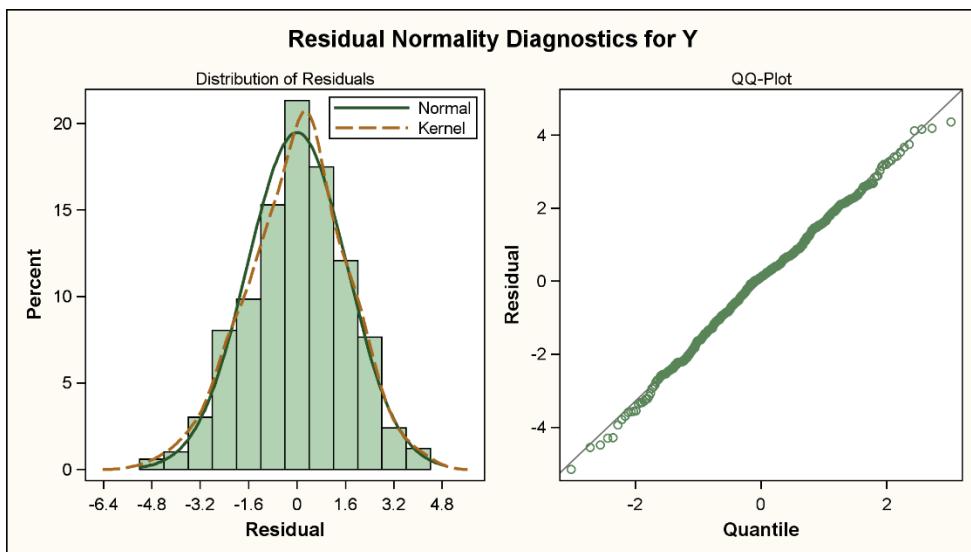
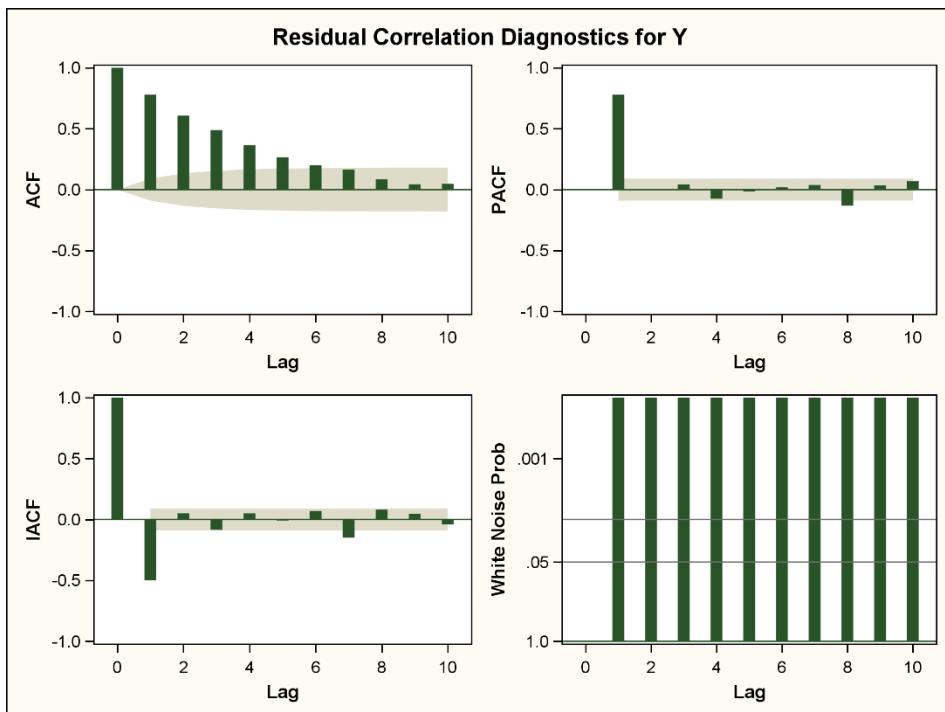
The ARIMA Procedure

Maximum Likelihood Estimation							
Parameter	Estimate	Standard Error	t Value	Approx Pr > t	Lag	Variable	Shift
MU	-33.99761	0.77553	-43.84	<.0001	0	Y	0
SCALE1	3.06271	0.07035	43.53	<.0001	0	X	2
NUM1,1	0.39865	0.02465	16.17	<.0001	1	X	2
DEN1,1	0.79069	0.0079330	99.67	<.0001	1	X	2

Constant Estimate	-33.9976
Variance Estimate	2.702153
Std Error Estimate	1.643823
AIC	1908.451
SBC	1925.285
Number of Residuals	497

Correlations of Parameter Estimates					
Variable Parameter	Y MU	X SCALE1	X NUM1,1	X DEN1,1	X
Y MU	1.000	-0.328	-0.347	-0.634	
X SCALE1	-0.328	1.000	0.689	0.291	
X NUM1,1	-0.347	0.689	1.000	0.821	
X DEN1,1	-0.634	0.291	0.821	1.000	

② Autocorrelation Check of Residuals									
To Lag	Chi-Square	DF	Pr > ChiSq	Autocorrelations					
6	731.66	6	<.0001	0.780	0.607	0.489	0.365	0.265	0.201
12	751.66	12	<.0001	0.165	0.084	0.043	0.048	0.031	0.002
18	771.14	18	<.0001	-0.027	-0.047	-0.096	-0.096	-0.095	-0.086
24	784.80	24	<.0001	-0.087	-0.071	-0.062	-0.075	-0.059	-0.025
30	802.67	30	<.0001	-0.021	-0.019	-0.048	-0.083	-0.111	-0.107
36	862.27	36	<.0001	-0.116	-0.141	-0.147	-0.143	-0.142	-0.125
42	892.19	42	<.0001	-0.108	-0.093	-0.106	-0.088	-0.090	-0.088
48	913.92	48	<.0001	-0.041	-0.011	0.018	0.053	0.116	0.145



To Lag	Chi-Square	DF	Pr > ChiSq	Crosscorrelations						
				-0.031	0.008	-0.004	-0.007	-0.006	0.001	
5	0.57	4	0.9668	-0.031	0.008	-0.004	-0.007	-0.006	0.001	
11	1.93	10	0.9969	0.007	-0.009	0.004	-0.002	-0.018	0.048	
17	7.25	16	0.9682	0.056	0.042	0.037	0.043	0.044	0.027	
23	15.71	22	0.8300	0.032	0.072	0.055	0.058	0.057	0.035	
29	16.35	28	0.9603	0.019	-0.003	0.016	0.012	-0.019	0.013	
35	19.87	34	0.9743	-0.075	-0.022	-0.002	0.029	-0.013	0.003	
41	23.89	40	0.9796	0.002	0.002	-0.019	0.026	0.070	0.047	
47	26.13	46	0.9919	0.021	0.023	-0.001	-0.019	0.032	0.047	

Model for variable Y	
Estimated Intercept	-33.9976

Input Number 1	
Input Variable	X
Shift	2
Overall Regression Factor	3.062708

Numerator Factors	
Factor 1: 1 - 0.39865 B**(1)	

Denominator Factors	
Factor 1: 1 - 0.79069 B**(1)	

Output 4.16: Modeling and Plotting Forecasts for Generated Data**The ARIMA Procedure
Preliminary Estimation**

Initial Autoregressive Estimates	
	Estimate
1	0.85629

Constant Term Estimate	1.445571
White Noise Variance Est	10.06195

Conditional Least Squares Estimation										
Iteration	SSE	MU	AR1,1	SCALE1	NUM1,1	DEN1,1	Constant	Lambda	R Crit	
0	3908.38	10.05915	0.85629	3.01371	0.10000	0.10000	1.445571	0.00001	1	
1	3583.47	9.16756	0.85945	2.99948	0.39860	0.41798	1.288464	0.00001	0.909428	
2	2552.90	4.14633	0.86958	2.97390	0.48026	0.58951	0.540763	0.001	0.91803	
3	2023.55	-2.96769	0.86982	2.97054	0.53937	0.72057	-0.38634	0.001	0.890669	
4	547.94	-25.1088	0.87029	2.92351	0.37808	0.74910	-3.25698	0.001	0.865045	

Maximum Likelihood Estimation										
Iter	Loglike	MU	AR1,1	SCALE1	NUM1,1	DEN1,1	Constant	Lambda	R Crit	
0	-725.24385	-25.1088	0.87029	2.92351	0.37808	0.74910	-3.25698	0.00001	1	
1	-716.89325	-33.4591	0.80592	2.98642	0.38655	0.79275	-6.49364	1E-6	0.23266	
2	-710.59597	-32.3938	0.78804	2.99291	0.37260	0.77873	-6.8662	1E-7	0.159835	
3	-710.53839	-32.4846	0.79232	2.99361	0.37277	0.77889	-6.7465	1E-8	0.015088	
4	-710.53820	-32.4651	0.79257	2.99341	0.37260	0.77873	-6.73437	1E-9	0.000805	

ARIMA Estimation Optimization Summary										
Estimation Method						Maximum Likelihood				
Parameters Estimated						5				
Termination Criteria						Maximum Relative Change in Estimates				
Iteration Stopping Value						0.001				
Criteria Value						0.0006				
Alternate Criteria						Relative Change in Objective Function				
Alternate Criteria Value						0.000035				
Maximum Absolute Value of Gradient						9.391771				
R-Square Change from Last Iteration						0.000805				

ARIMA Estimation Optimization Summary	
Objective Function	Log Gaussian Likelihood
Objective Function Value	-710.538
Marquardt's Lambda Coefficient	1E-9
Numerical Derivative Perturbation Delta	0.001
Iterations	4

Maximum Likelihood Estimation							
Parameter	Estimate	Standard Error	t Value	Approx Pr > t	Lag	Variable	Shift
MU	-32.46515	1.72585	-18.81	<.0001	0	Y	0
AR1,1	0.79257	0.02739	28.93	<.0001	1	Y	0
SCALE1	2.99341	0.04554	65.74	<.0001	0	X	2
NUM1,1	0.37260	0.02284	16.31	<.0001	1	X	2
DEN1,1	0.77873	0.01367	56.96	<.0001	1	X	2

Constant Estimate	-6.73437
Variance Estimate	1.029993
Std Error Estimate	1.014886
AIC	1431.076
SBC	1452.119
Number of Residuals	497

Correlations of Parameter Estimates						
Variable Parameter	Y MU	Y AR1,1	X SCALE1	X NUM1,1	X DEN1,1	X
Y MU	1.000	0.027	-0.288	-0.383	-0.795	
Y AR1,1	0.027	1.000	-0.021	0.003	-0.011	
X SCALE1	-0.288	-0.021	1.000	0.070	-0.016	
X NUM1,1	-0.383	0.003	0.070	1.000	0.818	
X DEN1,1	-0.795	-0.011	-0.016	0.818	1.000	

To Lag	Chi-Square	DF	Pr > ChiSq	Autocorrelations							
				-0.011	-0.040	0.063	-0.012	-0.037	-0.034		
6	4.22	5	0.5187	-0.011	-0.040	0.063	-0.012	-0.037	-0.034		
12	19.06	11	0.0600	0.118	-0.070	-0.084	0.044	0.037	-0.003		
18	26.15	17	0.0719	-0.017	0.052	-0.098	-0.005	-0.029	0.017		
24	34.15	23	0.0630	-0.043	0.006	0.046	-0.073	-0.040	0.067		
30	39.57	29	0.0912	-0.006	0.072	0.010	-0.025	-0.066	0.002		
36	42.81	35	0.1710	0.019	-0.040	-0.038	-0.003	-0.049	-0.018		
42	52.32	41	0.1108	-0.007	0.036	-0.067	0.030	-0.010	-0.104		
48	58.30	47	0.1248	0.028	-0.003	-0.012	-0.045	0.085	0.026		

⑤ Crosscorrelation Check of Residuals with Input X									
To Lag	Chi-Square	DF	Pr > ChiSq	Crosscorrelations					
5	0.75	4	0.9448	-0.002	0.019	-0.020	0.005	0.018	0.020
11	6.96	10	0.7296	0.026	-0.010	0.023	-0.002	-0.015	0.105
17	7.70	16	0.9573	0.026	0.003	0.007	0.020	0.018	-0.008
23	11.08	22	0.9736	0.013	0.074	-0.003	0.024	0.019	-0.015
29	14.34	28	0.9846	-0.015	-0.029	0.035	0.001	-0.046	0.047
35	28.14	34	0.7499	-0.131	0.060	0.024	0.051	-0.056	0.027
41	33.95	40	0.7385	-0.001	-0.002	-0.028	0.067	0.079	-0.015
47	38.77	46	0.7665	-0.021	0.011	-0.033	-0.026	0.077	0.038

Model for variable Y	
Estimated Intercept	-32.4651

Autoregressive Factors	
Factor 1:	1 - 0.79257 B**(1)

Input Number 1	
Input Variable	X
Shift	2
Overall Regression Factor	2.993406

Numerator Factors	
Factor 1:	1 - 0.3726 B**(1)

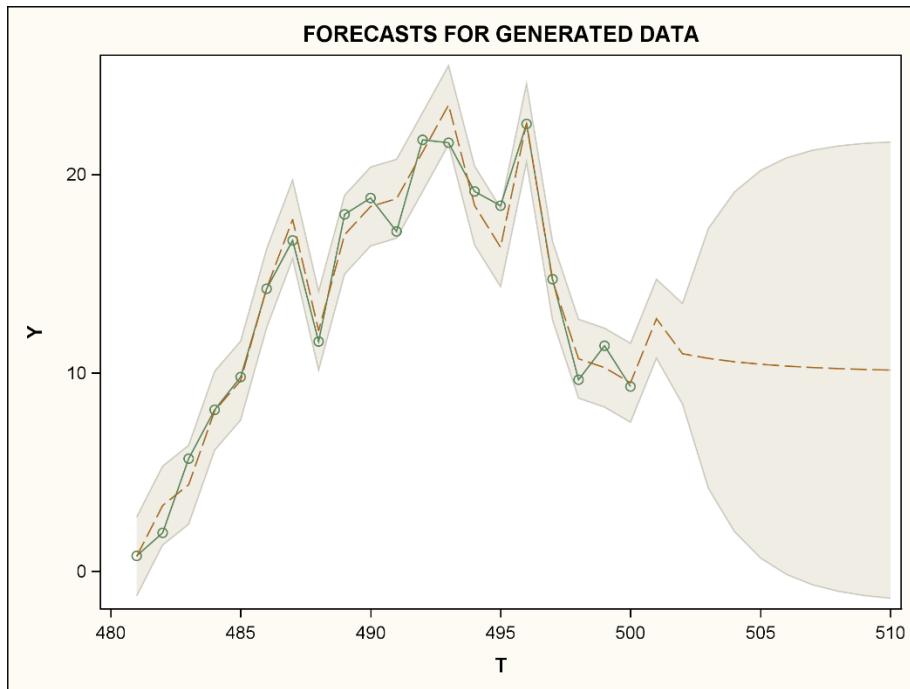
Denominator Factors	
Factor 1:	1 - 0.77873 B**(1)

Forecasts for variable Y				
Obs	Forecast	Std Error	95% Confidence Limits	
501	12.7292	1.0149	10.7401	14.7184
502	10.9660	1.2950	8.4279	13.5042
503	10.7302	3.3464	4.1715	17.2890
504	10.5609	4.3680	1.9999	19.1220
505	10.4362	4.9805	0.6746	20.1977
506	10.3424	5.3556	-0.1543	20.8392
507	10.2710	5.5859	-0.6771	21.2191
508	10.2161	5.7270	-1.0087	21.4409
509	10.1736	5.8134	-1.2204	21.5677
510	10.1406	5.8661	-1.3568	21.6381

Obs	T	Y	FORECAST	STD	L95	U95	RESIDUAL
1	481	0.7780	0.7495	1.01489	-1.2396	2.7387	0.02853
2	482	1.9415	3.3176	1.01489	1.3284	5.3067	-1.37604
3	483	5.6783	4.3649	1.01489	2.3758	6.3541	1.31335
4	484	8.1388	8.0926	1.01489	6.1035	10.0818	0.04618
5	485	9.7920	9.6121	1.01489	7.6230	11.6013	0.17983

(Additional output omitted)

26	506	.	10.3424	5.35558	-0.1543	20.8392	.
27	507	.	10.2710	5.58586	-0.6771	21.2191	.
28	508	.	10.2161	5.72704	-1.0087	21.4409	.
29	509	.	10.1736	5.81341	-1.2204	21.5677	.
30	510	.	10.1406	5.86614	-1.3568	21.6381	.



As shown in **Output 4.16 ①**, standard errors are (1.73), (0.05), (0.01), (0.02), and (0.03).

In the autocorrelation and cross-correlation checks of residuals and input, note the following facts:

- Chi-square statistics automatically printed by PROC ARIMA are like the Q statistics discussed earlier for standard PROC ARIMA models.
- Cross-correlation of residuals with input implies improper identification of the transfer function model. This is often accompanied by autocorrelation in residuals.
- Autocorrelation of residuals not accompanied by cross-correlation of residuals with X indicates that the transfer function is right, but that the noise model is not properly identified.

See **Output 4.15 ②**:

```
estimate input=(2$(1)/(1)x) ... ;
```

This is in contrast to **Output 4.16 ③**:

```
estimate p=1 ... ;
```

Neither cross-correlation check ④ or ⑤ indicates any problem with the transfer specification. First, the inputs are forecast. Then, they are used to forecast Y . In an example without prewhitening, future values of X must be in the original data set.

In addition to generated data, logarithms of flow rates for the Neuse River in Goldsboro, North Carolina, and 30 miles downstream in Kinston, North Carolina, are analyzed. These data include 400 daily observations. Obviously, the flow rates develop a seasonal pattern over the 365 days in a year, causing the ACF to die off slowly. Taking differences of the logarithmic observations produces ACFs that seem well behaved. The goal is to relate flow rates at Kinston to those at

Goldsboro. The differenced data should suffice for transfer function identification even though nonstationarity is probably caused by the 365-day seasonal periodicity in flows.

You can obtain a model for the logarithms of the Goldsboro flow rates by using the following SAS statements:

```
title "FLOW RATES OF NEUSE RIVER AT GOLDSBORO AND KINSTON";
proc arima data=river;
  identify var=lgold(1) noint;
  estimate q=1 p=3 method=ml noint;
  identify var=lkins(1) crosscor=(lgold(1));
quit;
```

The results are shown in **Output 4.17**.

Output 4.17: Analyzing Logarithms of Flow Data with the IDENTIFY and ESTIMATE Statements: PROC ARIMA

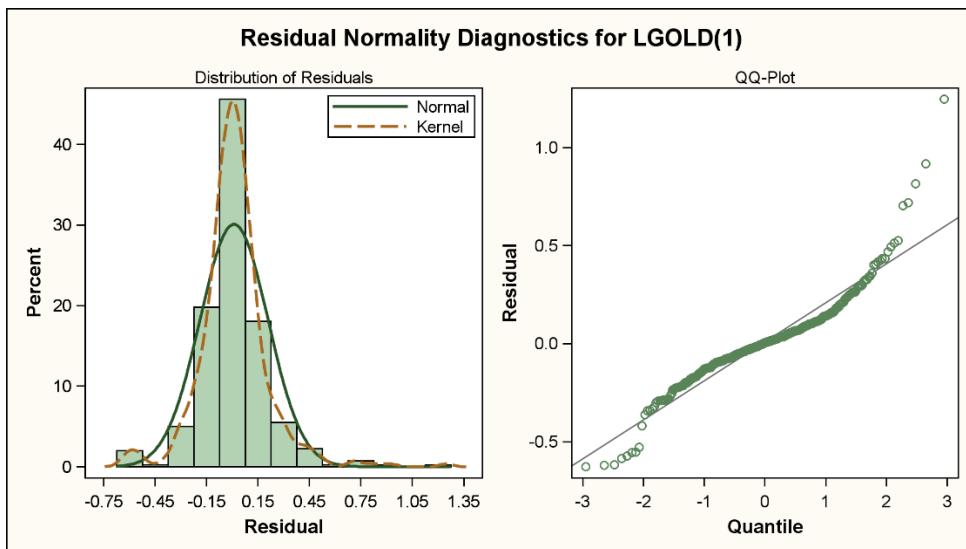
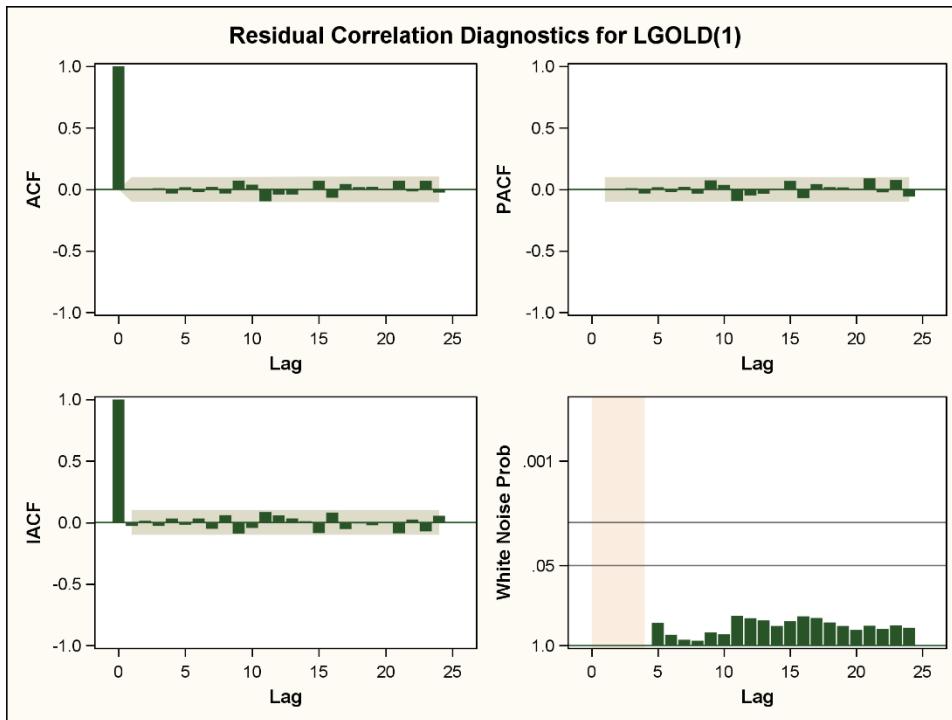
The ARIMA Procedure

Maximum Likelihood Estimation					
Parameter	Estimate	Standard Error	t Value	Approx Pr > t	Lag
MA1,1	0.87394	0.05878	14.87	<.0001	1
AR1,1	1.24083	0.07467	16.62	<.0001	1
AR1,2	-0.29074	0.08442	-3.44	0.0006	2
AR1,3	-0.11724	0.05394	-2.17	0.0297	3

Variance Estimate	0.039916
Std Error Estimate	0.199791
AIC	-148.394
SBC	-132.438
Number of Residuals	399

Correlations of Parameter Estimates				
Parameter	MA1,1	AR1,1	AR1,2	AR1,3
MA1,1	1.000	0.745	-0.367	0.374
AR1,1	0.745	1.000	-0.783	0.554
AR1,2	-0.367	-0.783	1.000	-0.847
AR1,3	0.374	0.554	-0.847	1.000

Autocorrelation Check of Residuals										
To Lag	Chi-Square	DF	Pr > ChiSq	Autocorrelations						
6	0.77	2	0.6819	-0.001	0.003	0.012	-0.032	0.020	-0.019	
12	8.67	8	0.3711	0.023	-0.032	0.074	0.041	-0.095	-0.040	
18	14.28	14	0.4287	-0.040	0.006	0.071	-0.066	0.045	0.020	
24	19.19	20	0.5097	0.024	0.004	0.072	-0.014	0.071	-0.024	
30	22.15	26	0.6807	-0.018	0.058	-0.052	-0.021	0.007	-0.001	
36	26.18	32	0.7555	0.005	0.025	0.022	-0.089	-0.008	-0.005	
42	32.18	38	0.7347	-0.065	0.047	0.080	-0.016	0.006	-0.023	
48	34.21	44	0.8555	-0.029	0.005	-0.007	-0.010	0.042	0.041	



Model for variable LGOLD	
Period(s) of Differencing	1

No mean term in this model.

Autoregressive Factors	
Factor 1:	$1 - 1.24083 B^{**}(1) + 0.29074 B^{**}(2) + 0.11724 B^{**}(3)$

Moving Average Factors	
Factor 1:	$1 - 0.87394 B^{**}(1)$

Name of Variable = LKINS	
Period(s) of Differencing	1
Mean of Working Series	0.006805
Standard Deviation	0.152423
Number of Observations	399
Observation(s) eliminated by differencing	1

Autocorrelation Check for White Noise									
To Lag	Chi-Square	DF	Pr > ChiSq	Autocorrelations					
6	153.89	6	<.0001	0.548	0.187	0.096	-0.011	-0.108	-0.161
12	212.40	12	<.0001	-0.205	-0.175	-0.121	-0.143	-0.149	-0.115
18	222.41	18	<.0001	-0.053	0.032	0.063	0.002	0.076	0.101
24	231.81	24	<.0001	0.074	0.074	0.069	0.057	0.030	-0.050

Variable LGOLD has been differenced.

Correlation of LKINS and LGOLD	
Period(s) of Differencing	1
Number of Observations	399
Observation(s) eliminated by differencing	1
Variance of transformed series LKINS	0.016725
Variance of transformed series LGOLD	0.039507

Both series have been prewhitened.

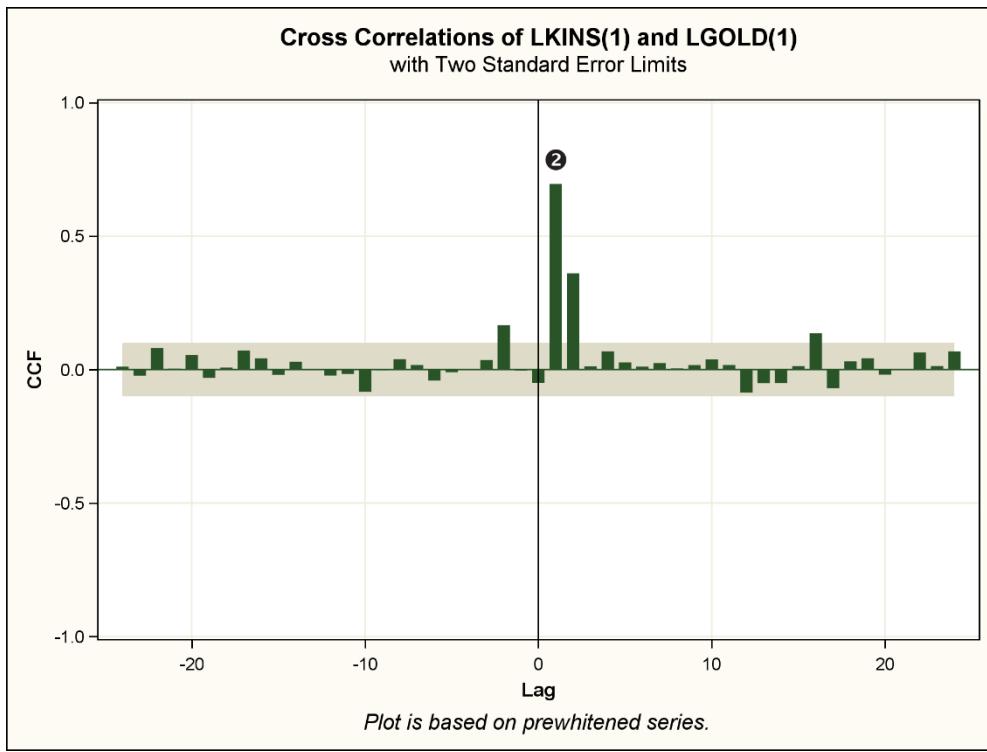
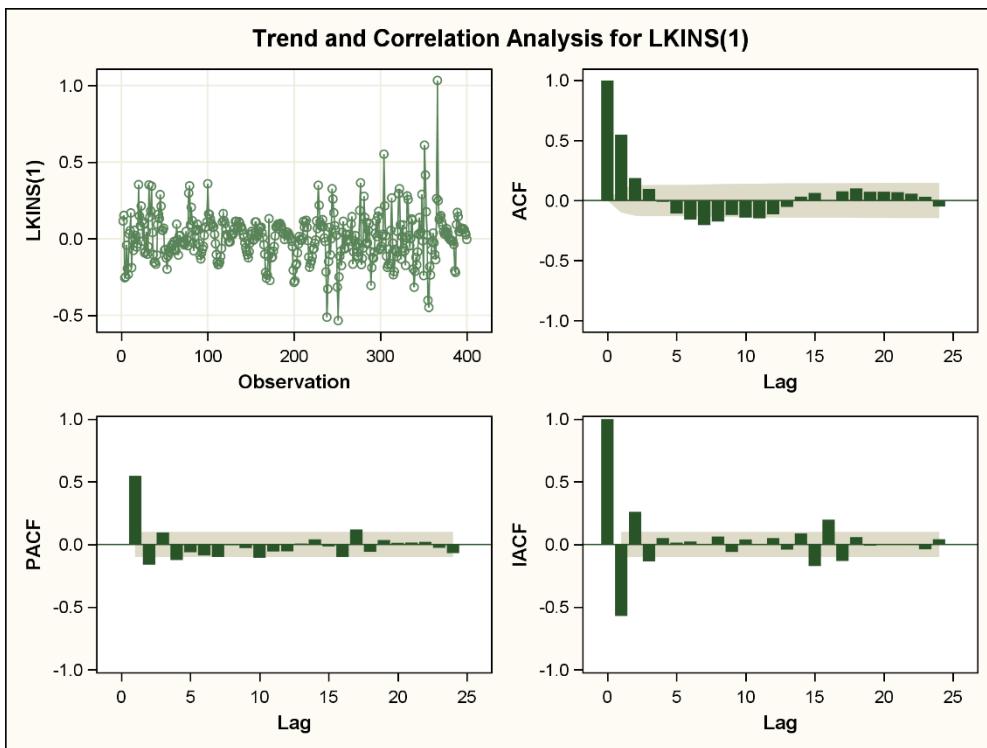
Crosscorrelation Check Between Series									
To Lag	Chi-Square	DF	Pr > ChiSq	Crosscorrelations					
5	248.39	6	<.0001	-0.050	0.696	0.361	0.012	0.069	0.027
11	249.50	12	<.0001	0.011	0.024	0.005	0.017	0.039	0.017
17	263.88	18	<.0001	-0.086	-0.051	-0.051	0.013	0.136	-0.070
23	266.85	24	<.0001	0.031	0.043	-0.019	0.001	0.064	0.013

Both variables have been prewhitened by the following filter:

Prewhitening Filter

Autoregressive Factors	
Factor 1:	$1 - 1.24083 B^{**}(1) + 0.29074 B^{**}(2) + 0.11724 B^{**}(3)$

Moving Average Factors	
Factor 1:	$1 - 0.87394 B^{**}(1)$



The output from the ESTIMATE statement ❶ shows a reasonable fit. Cross-correlations from the second IDENTIFY statement ❷ show that a change in flow rates in Goldsboro affects the flow in Kinston one and two days later, with little other effect. This suggests $C(1 - \theta B)X_{t-1}$ as a transfer function. Add the following SAS statements to the previous code:

```
estimate input=(1$(1)lgold) method=ml;
run;
```

Results are shown in **Output 4.18**.

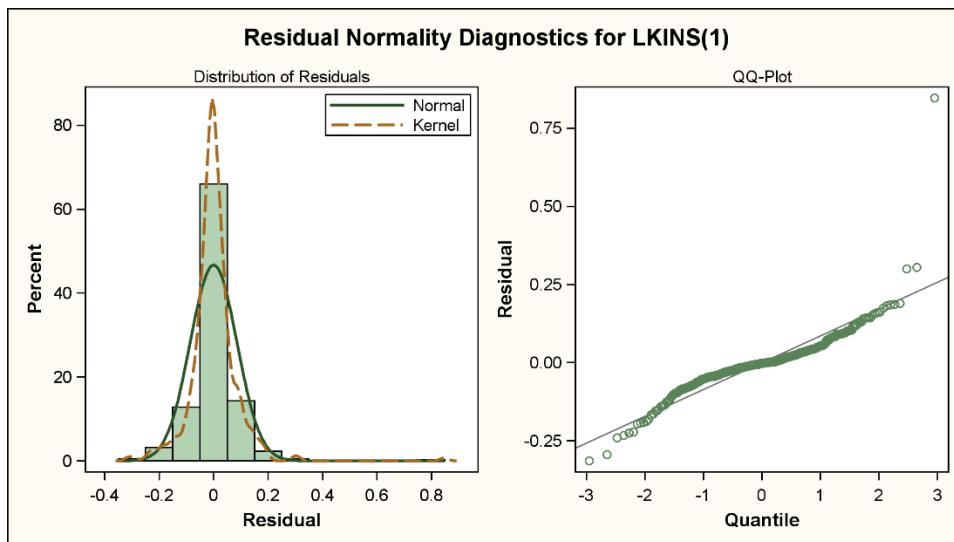
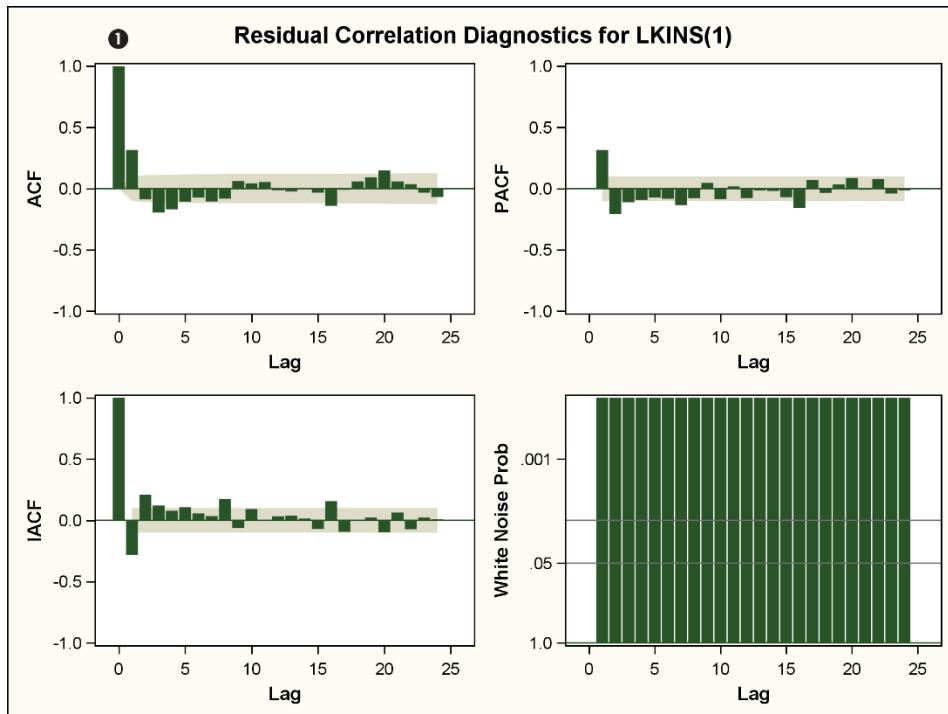
Output 4.18: Modeling Flow Rates: Identifying an Error Model through the Residual Plots**The ARIMA Procedure**

Maximum Likelihood Estimation							
Parameter	Estimate	Standard Error	t Value	Approx Pr > t	Lag	Variable	Shift
MU	0.0018976	0.0043054	0.44	0.6594	0	LKINS	0
NUM1	0.43109	0.02107	20.46	<.0001	0	LGOLD	1
NUM1,1	-0.22837	0.02106	-10.84	<.0001	1	LGOLD	1

Constant Estimate	0.001898
Variance Estimate	0.00735
Std Error Estimate	0.085733
AIC	-820.849
SBC	-808.897
Number of Residuals	397

Correlations of Parameter Estimates				
Variable Parameter	LKINS MU	LGOLD NUM1	LGOLD NUM1,1	
LKINS MU	1.000	-0.018	0.020	
LGOLD NUM1	-0.018	1.000	0.410	
LGOLD NUM1,1	0.020	0.410	1.000	

To Lag	Chi-Square	DF	Pr > ChiSq	Autocorrelations							
				-6	-12	-18	-24	-30	-36	-42	-48
6	76.40	6	<.0001	0.315	-0.088	-0.195	-0.169	-0.106	-0.073		
12	87.15	12	<.0001	-0.105	-0.081	0.062	0.042	0.054	-0.013		
18	97.51	18	<.0001	-0.023	-0.005	-0.033	-0.140	-0.010	0.059		
24	114.97	24	<.0001	0.092	0.150	0.059	0.036	-0.033	-0.069		
30	120.35	30	<.0001	-0.044	-0.053	-0.056	-0.062	-0.023	-0.019		
36	131.49	36	<.0001	0.031	0.042	0.087	0.099	0.063	-0.039		
42	137.18	42	<.0001	-0.049	-0.042	-0.041	-0.004	-0.048	-0.068		
48	152.24	48	<.0001	0.018	0.128	0.103	0.056	0.013	-0.054		



Crosscorrelation Check of Residuals with Input LGOLD									
To Lag	Chi-Square	DF	Pr > ChiSq	Crosscorrelations Θ					
5	4.08	5	0.5373	0.011	-0.012	-0.010	0.071	0.065	0.027
11	11.11	11	0.4345	0.036	-0.007	0.019	-0.008	-0.076	-0.101
17	21.14	17	0.2202	-0.033	-0.025	0.027	0.140	-0.007	0.056
23	24.04	23	0.4014	0.036	-0.048	-0.041	-0.015	-0.002	0.042
29	27.77	29	0.5303	-0.024	-0.065	-0.063	-0.014	-0.006	0.020
35	33.12	35	0.5593	0.023	0.007	0.011	-0.063	0.016	0.093
41	44.70	41	0.3193	0.126	0.045	-0.038	-0.067	-0.047	0.055
47	50.83	47	0.3252	-0.044	-0.057	0.062	0.072	-0.005	-0.033

Model for variable LKINS	
Estimated Intercept	0.001898
Period(s) of Differencing	1

Input Number 1	
Input Variable	LGOLD
Shift	1
Period(s) of Differencing	1

Numerator Factors	
Factor 1:	$0.43109 + 0.22837 B^{**}(1)$

Diagnostics automatically produced by ODS graphics ① are used to identify an error model. The cross-correlations check ② looks reasonable, but the autocorrelations check ③ indicates that the error is not white noise. This implies that t statistics for the model parameters are computed from improper standard errors.

An ARMA(2,1) model fits the error term. Make the final estimation of the transfer function with noise by replacing the ESTIMATE statement with the following:

```
estimate p=2 q=1 input=(1$(1)lgold) method=ml noconstant
      altparm;
run;
```

Output 4.19 shows the results.

Output 4.19: Estimating the Final Transfer Function: PROC ARIMA

The ARIMA Procedure

Maximum Likelihood Estimation							
Parameter	Estimate	Standard Error	t Value	Approx Pr > t	Lag	Variable	Shift
MA1,1	0.88776	0.03506	25.32	<.0001	1	LKINS	0
AR1,1	1.16325	0.05046	23.05	<.0001	1	LKINS	0
AR1,2	-0.47963	0.04564	-10.51	<.0001	2	LKINS	0
SCALE1	0.49539	0.01847	26.83	<.0001	0	LGOLD	1
NUM1,1	-0.55026	0.04540	-12.12	<.0001	1	LGOLD	1

Variance Estimate	0.005838
Std Error Estimate	0.076407
AIC	-909.399
SBC	-889.48
Number of Residuals	397

Correlations of Parameter Estimates					
Variable Parameter	LKINS MA1,1	LKINS AR1,1	LKINS AR1,2	LGOLD SCALE1	LGOLD NUM1,1
LKINS MA1,1	1.000	0.478	0.126	0.108	-0.091
LKINS AR1,1	0.478	1.000	-0.618	0.056	-0.097
LKINS AR1,2	0.126	-0.618	1.000	-0.121	0.042
LGOLD SCALE1	0.108	0.056	-0.121	1.000	0.590
LGOLD NUM1,1	-0.091	-0.097	0.042	0.590	1.000

Autocorrelation Check of Residuals									
To Lag	Chi-Square	DF	Pr > ChiSq	Autocorrelations					
6	0.37	3	0.9468	0.005	-0.008	-0.010	-0.001	0.018	0.020
12	10.07	9	0.3446	-0.043	-0.094	0.092	-0.020	0.044	-0.048
18	20.04	15	0.1703	-0.030	0.004	0.036	-0.132	0.048	0.045
24	31.10	21	0.0720	0.023	0.145	0.001	0.064	-0.005	-0.025
30	33.30	27	0.1874	0.019	-0.010	-0.018	-0.056	0.002	-0.035
36	37.66	33	0.2645	0.024	-0.007	0.045	0.044	0.052	-0.052
42	39.40	39	0.4522	0.003	0.018	0.005	0.058	-0.005	-0.012
48	47.78	45	0.3605	0.028	0.111	0.054	0.035	0.035	-0.010

Crosscorrelation Check of Residuals with Input LGOLD									
To Lag	Chi-Square	DF	Pr > ChiSq	Crosscorrelations					
5	6.41	4	0.1705	-0.044	-0.062	0.010	0.079	0.054	0.034
11	14.95	10	0.1340	0.083	0.038	0.100	0.042	-0.031	-0.024
17	28.12	16	0.0306	0.041	-0.002	0.040	0.130	-0.051	0.102
23	29.97	22	0.1192	0.044	-0.036	0.006	0.003	0.003	0.038
29	33.48	28	0.2185	-0.047	-0.053	-0.056	-0.006	-0.026	-0.002
35	40.58	34	0.2028	-0.008	-0.018	-0.003	-0.087	0.045	0.089
41	48.60	40	0.1650	0.104	0.040	-0.008	-0.025	-0.001	0.085
47	55.14	46	0.1674	-0.069	-0.032	0.088	0.048	-0.019	-0.017

Model for variable LKINS ①	
Period(s) of Differencing	1

No mean term in this model.

Autoregressive Factors	
Factor 1:	1 - 1.16325 B**(1) + 0.47963 B**2)

Moving Average Factors	
Factor 1:	1 - 0.88776 B**1)

Input Number 1	
Input Variable	LGOLD
Shift	1
Period(s) of Differencing	1
Overall Regression Factor	0.495394

Numerator Factors	
Factor 1:	$1 + 0.55026 B^{**}(1)$

The model becomes ①:

$$\nabla LKINS_t = 0.49539(1 + 0.55B)\nabla LGOLD_{t-1} + (1 - 0.8877B) \\ / (1 - 1.16325B + 0.47963B^2)e_t$$

Because you encountered a pure delay, this is an example of a leading indicator, although this term is generally reserved for economic data.

Summary of Modeling Strategy

Follow these steps in case 3 to complete your modeling:

1. Identify and estimate a model for input X (IDENTIFY, ESTIMATE).
2. Prewhiten Y and X using model from item 1 (IDENTIFY).
3. Compute cross-correlations, $\gamma_{XY}(j)$, to identify transfer function form (IDENTIFY).
4. Fit a transfer function and compute and analyze residuals (ESTIMATE, PLOT).
5. Fit a transfer function with noise model (ESTIMATE).
6. Forecast X and Y (FORECAST).

4.3.4 Case 3B: Intervention

Suppose you use as an input X_t a sequence that is 0 through time 20 and 1 from time 21 onward. If the model is $Y_t = \alpha + \beta X_t + \text{Noise}$, then you have $Y_t = \alpha + \text{Noise}$ through time 20 and $Y_t = \alpha + \beta + \text{Noise}$ after time 20. Thus, Y experiences an immediate level shift (from α to $\alpha + \beta$) at time 21.

Change the model to $Y_t - \rho Y_{t-1} = \alpha + \beta X_t + \text{Noise}$ or $Y_t = \alpha' + \beta / (1 - \rho B) X_t + \text{Noise}$, where $\alpha' = \alpha / (1 - \rho)$ (the expected value of Y when X is 0). You can also write the following:

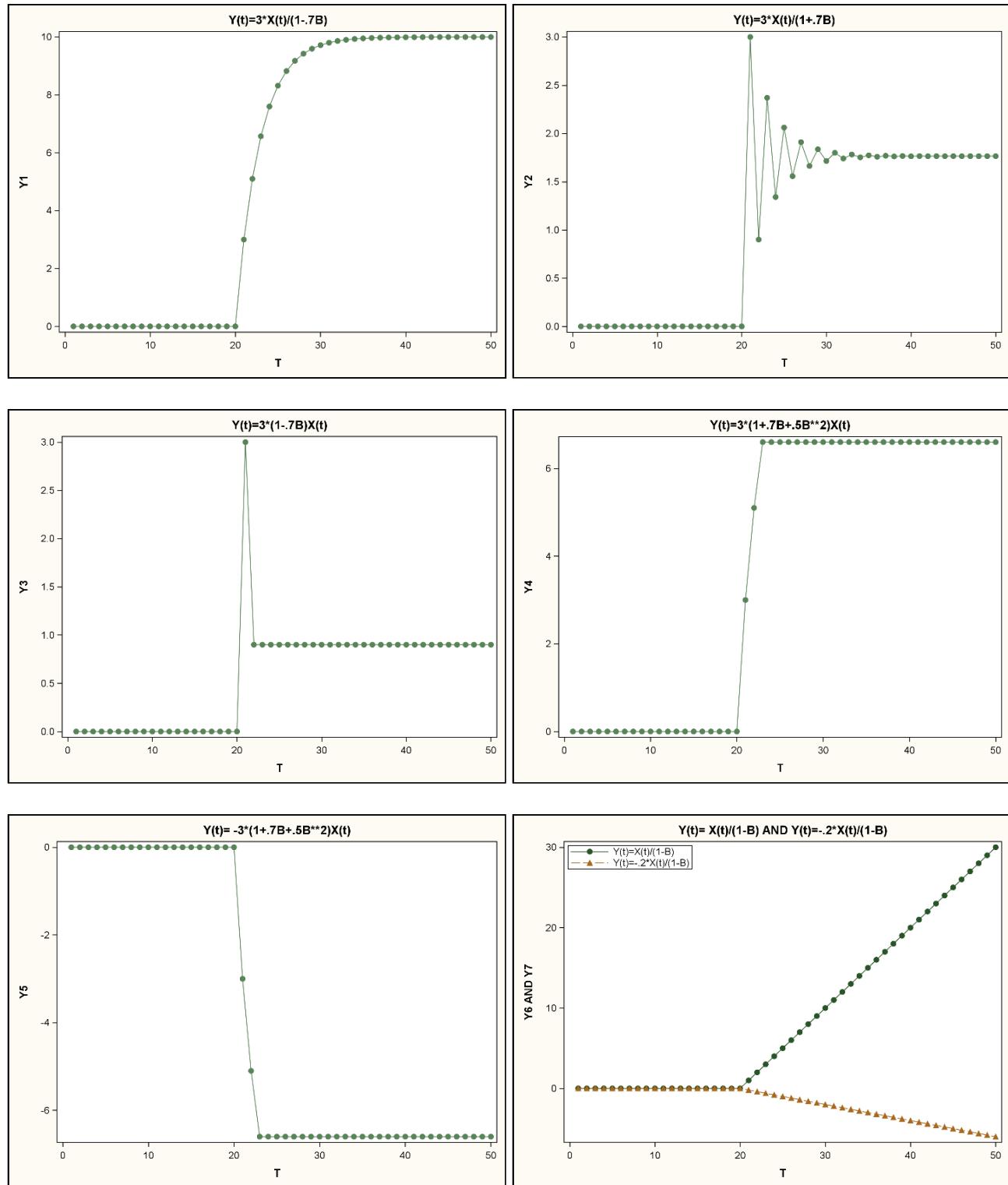
$$Y_t = \alpha' + \beta(X_t + \rho X_{t-1} + \rho^2 X_{t-2} + \dots) + \text{Noise}$$

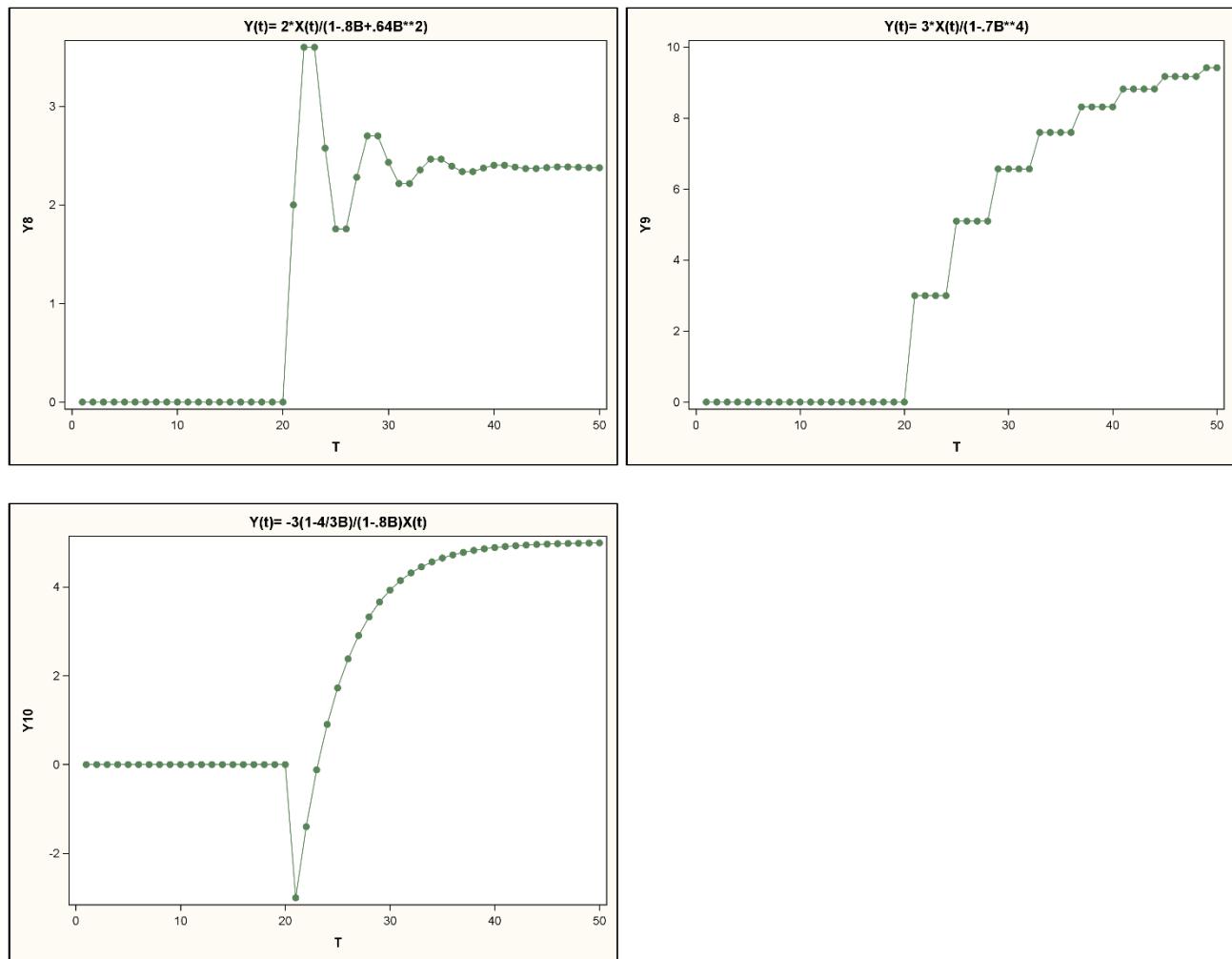
At time 21, $X_{21} = 1$ and the previous X s are 0, so $Y_{21} = \alpha' + \beta + \text{Noise}$. At time 22, you get $Y_{22} = \alpha' + \beta(1 + \rho) + \text{Noise}$. Y_t eventually approaches the following if you ignore the noise term:

$$\alpha' + \beta(1 + \rho + \rho^2 + \dots) = \alpha' + \beta / (1 - \rho)$$

Thus, you see that ratios of polynomials in the backshift operator B can provide interesting approaches to new levels.

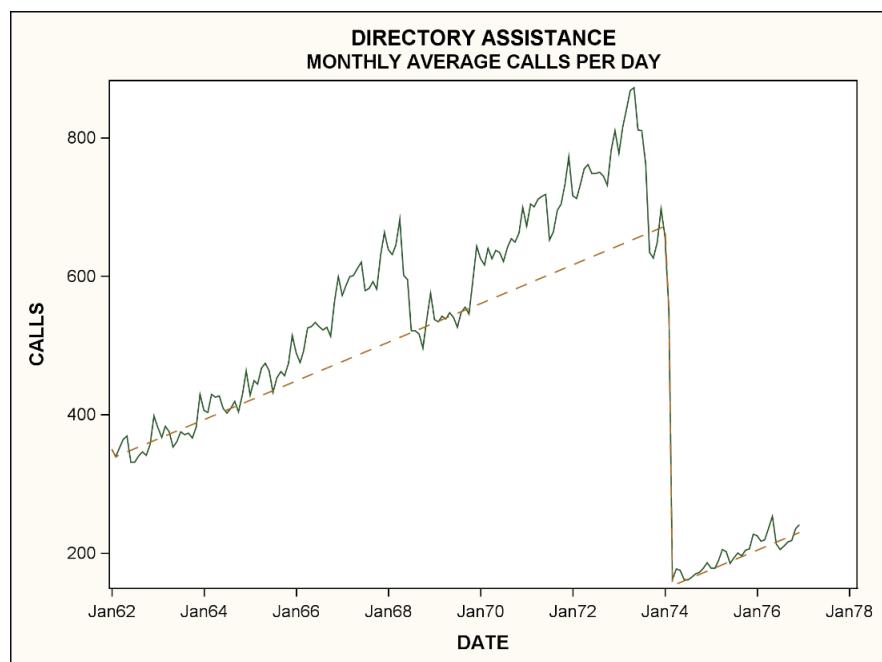
When you use an indicator input, you cannot prewhiten. Therefore, impulse-response weights are not proportional to cross-covariances. You make the identification by comparing the behavior of Y_t near the intervention point with a catalog of typical behaviors for various transfer function forms. Several such response functions for $X_t = 1$ when $t > 20$ and 0 otherwise are shown in **Output 4.20**.

Output 4.20: Plotting Intervention Models



Output 4.21 shows calls for directory assistance in Cincinnati, Ohio (McSweeny, 1978).

Output 4.21: Plotting the Original Data



Prior to March 1974, directory assistance was free, but from that day on, a charge was imposed. The data seem to show an initial falling off of demand starting in February, which might be an anticipation effect. The data clearly show an upward trend. You check the pre-intervention data for stationarity with the following code:

```
proc arima data=calls;
  identify var=calls stationarity = (adf=(2,3,12,13));
  identify var=calls(1);
  estimate p=(12) method=ml;
  where date < '01feb74'd;
run;
```

Some of the results are shown in **Output 4.22**. Only the trend tests are of interest because there is clearly a trend. However, none of the other tests could reject a unit root either. Tests with 12 and 13 lagged differences are requested in anticipation of seasonality. Below this are the chi-square checks for a seasonal AR model for the first differences. The fit is excellent and the seasonal AR parameter 0.5693 is not too close to 1. With this information, you see that only the unit root tests with 12 or more lags are valid.

Output 4.22: Unit Root Tests, Pre-intervention Calls Data

The ARIMA Procedure

Augmented Dickey-Fuller Unit Root Tests							
Type	Lags	Rho	Pr < Rho	Tau	Pr < Tau	F	Pr > F
Zero Mean	2	0.3448	0.7653	0.71	0.8681		
	3	0.3145	0.7576	0.59	0.8433		
	12	0.3421	0.7645	0.57	0.8390		
	13	0.3038	0.7547	0.46	0.8121		
Single Mean	2	-3.0059	0.6526	-1.48	0.5421	1.70	0.6370
	3	-3.3438	0.6113	-1.49	0.5379	1.58	0.6670
	12	-3.4458	0.5988	-1.46	0.5515	1.58	0.6693
	13	-4.1907	0.5129	-1.56	0.4996	1.65	0.6513
Trend	2	-18.1708	0.0888	-2.48	0.3378	3.37	0.5037
	3	-31.1136	0.0045	-3.04	0.1252	4.84	0.2091
	12	124.1892	0.9999	-2.96	0.1489	4.70	0.2387
	13	52.3936	0.9999	-3.18	0.0924	5.46	0.0964

Autocorrelation Check of Residuals									
To Lag	Chi-Square	DF	Pr > ChiSq	Autocorrelations					
6	2.71	5	0.7451	-0.015	-0.001	0.065	0.019	-0.111	-0.030
12	6.07	11	0.8683	-0.004	0.060	-0.039	-0.085	0.067	-0.067
18	11.10	17	0.8511	0.018	-0.067	0.029	-0.095	0.035	-0.120
24	16.64	23	0.8263	-0.002	-0.028	0.082	-0.149	-0.006	0.049

Model for variable CALLS	
Estimated Mean	1.077355
Period(s) of Differencing	1

Autoregressive Factors	
Factor 1:	1 - 0.56934 B**(12)

A first difference will reduce a linear trend to a constant, so calls tend to increase by 1.077 per month. The intervention variable IMPACT is created, having value 1 from February 1974 onward. Because the majority of the drop is seen in March, you fit an intervention model of the form $(\beta_0 - \beta_1 B)X_t$, where X_t is the IMPACT variable at time t . The first time X_t is 1, the effect is β_0 , and after that, both X_t and X_{t-1} will be 1 so that the effect is $\beta_0 - \beta_1$. You anticipate a negative β_1 .

and a larger-in-magnitude and positive β_1 . A test that $\beta_0 = 0$ is a test for an anticipation effect. Motivated by the pre-intervention analysis, you try the same seasonal AR(1) error structure and check the diagnostics to see whether it suffices. The code is:

```
proc arima;
  identify var=calls(1) crosscor= (impact(1)) noprint;
  estimate input = ((1)impact) p=(12) method=ml;
run;
```

In **Output 4.23**, all terms except mu are significant. The trend part of the fitted model is overlaid on the data in **Output 4.24**. Because the model has a unit root, the data can wander fairly far from this trend. This, indeed, happens. It also explains why the standard error for mu is so large. That is, with random walk errors, it is difficult to accurately estimate the drift term. Despite this, the model seems to capture the intervention well and seems poised to offer an accurate forecast of the next few values. The drop of -123 in calls the month prior to the charge is significant, so there was an anticipation effect. An additional drop of 400 leaves the calls at 523 below the previous levels.

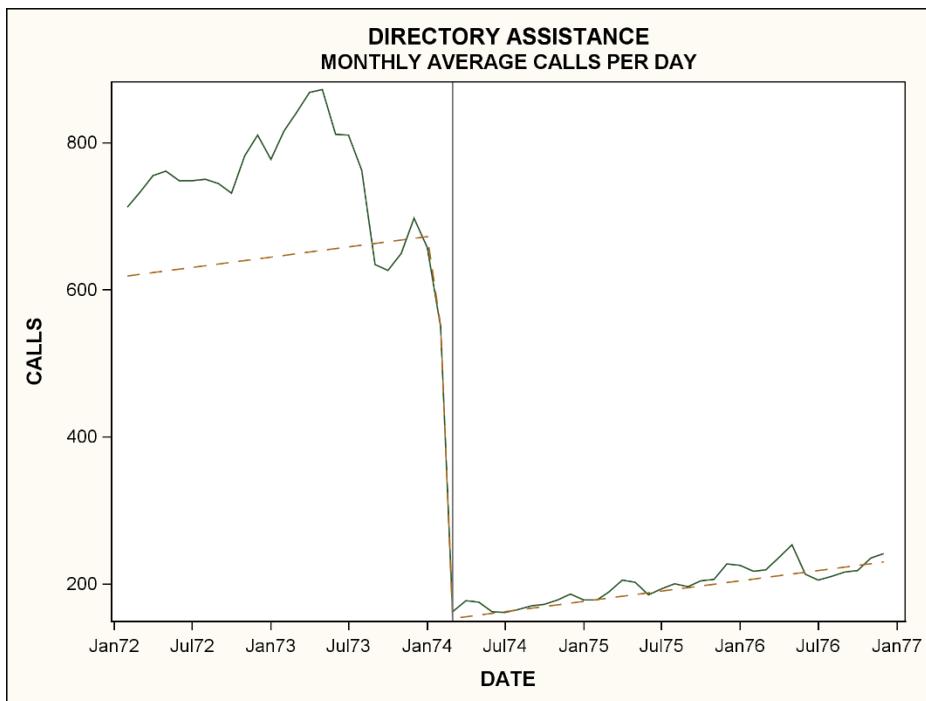
Output 4.23: PROC ARIMA for Calls Data

The ARIMA Procedure

Maximum Likelihood Estimation							
Parameter	Estimate	Standard Error	t Value	Approx Pr > t	Lag	Variable	Shift
MU	2.32863	2.88679	0.81	0.4199	0	CALLS	0
AR1,1	0.45045	0.06740	6.68	<.0001	12	CALLS	0
NUM1	-123.18861	20.39502	-6.04	<.0001	0	IMPACT	0
NUM1,1	400.69122	20.34270	19.70	<.0001	1	IMPACT	0

Constant Estimate	1.279694
Variance Estimate	503.7929
Std Error Estimate	22.44533
AIC	1619.363
SBC	1632.09
Number of Residuals	178

Autocorrelation Check of Residuals									
To Lag	Chi-Square	DF	Pr > ChiSq	Autocorrelations					
6	3.43	5	0.6342	0.009	-0.046	0.058	0.016	-0.109	-0.030
12	7.91	11	0.7209	-0.026	0.048	-0.022	-0.110	0.015	-0.088
18	11.47	17	0.8312	-0.024	-0.093	0.021	-0.055	0.025	-0.068
24	19.83	23	0.6518	0.006	0.001	0.062	-0.162	-0.029	0.098
30	22.73	29	0.7886	-0.021	0.019	0.016	-0.025	-0.026	0.105

Output 4.24: Effect of Charge for Directory Assistance

To forecast the next few months, you extend the data set with missing values for calls and set the intervention variable to 1, assuming the charge will remain in effect. The following code produces the plot in **Output 4.25**. The forecasts and intervals for the historical data have been deleted from the plot. The intervals are wide due to the unit root structure of the errors. Recall that even the historical data have produced some notable departures from trend. Adding other predictor variables, such as population or new phone installations, might help reduce the size of these intervals, but the predictors would need to be extrapolated into the future.

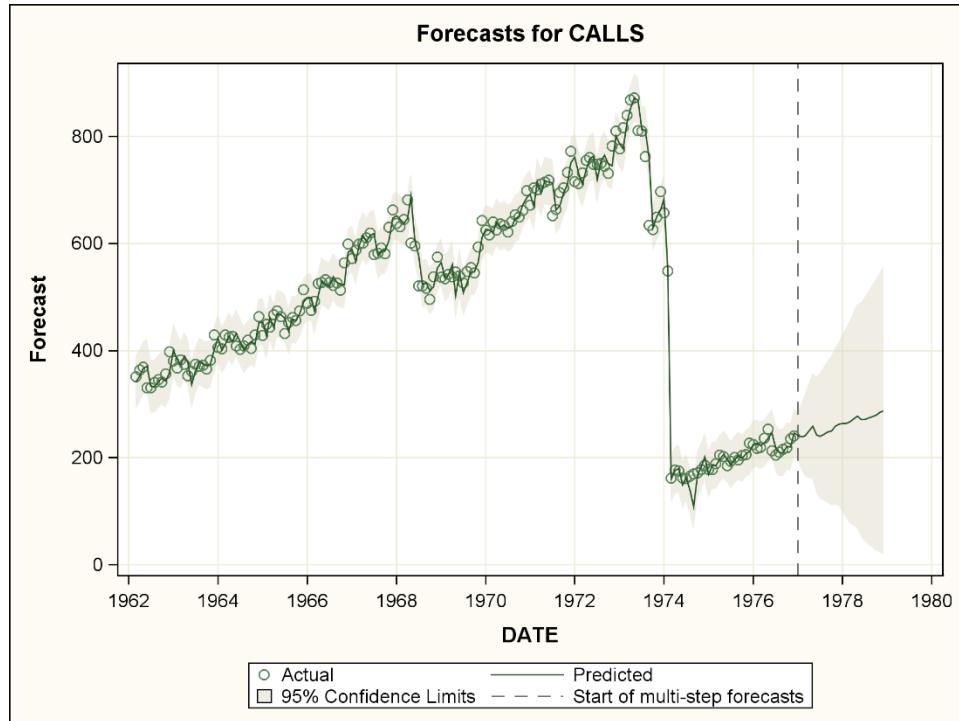
```

data extra;
  do t=1 to 24;
    date = intnx('month','01dec76'd,t);
    impact=1;
  output;
  end;
run;

data all;
  set calls extra;
run;

ods select forecastsplot;
proc arima data=all plots(only)=forecast(forecast);
  identify var=calls(1) crosscor=(impact(1)) noprint;
  estimate input = ((1)impact) p=(12) method=ml noprint;
  forecast lead=24 out=graph id=date interval=month;
quit;

```

Output 4.25: Forecasts from Intervention Model

4.4 Further Example

Earlier, a data set of NC retail sales was presented. This section revisits those data.

4.4.1 North Carolina Retail Sales

Consider again the North Carolina retail sales data investigated in Chapter 1. Recall that in that chapter, the quarterly sales increases were modeled using seasonal dummy variables. Seasonal dummy variables were fit to the first differences of quarterly sales. The models discussed in this section potentially provide an alternative approach.

Here, the full monthly data (from which the quarterly numbers were computed as averages) is used. This is an example in which the airline model seems a good choice at first, but later runs into some problems. Recall that when a first difference is found, often a moving average at lag 1 is appropriate. Likewise, a multiplicative moving average structure, specified by ESTIMATE Q = (1)(12), often works well when the first and span 12 difference, $(Y_t - Y_{t-1}) - (Y_{t-12} - Y_{t-13})$, has been taken. You can think of these moving average terms as somewhat mitigating the impact of the rather heavy-handed differencing operator. As in the IBM example in section 3.4.7, the fitting of these moving average terms causes forecasts to be weighted averages of seasonal patterns over all past years where the weights decrease exponentially as you move further into the past. Thus, the forecast is influenced somewhat by all past patterns, but most substantially by those of the most recent years.

The airline model is written as $(1 - B)(1 - B^{12})Y_t = (1 - \theta_{1,1}B)(1 - \theta_{2,1}B^{12})e_t$, introducing double subscripts to indicate which factor and which lag within that factor is being modeled. This double-subscript notation corresponds to PROC ARIMA output. The airline model is often a good first try when seasonal data are encountered. Now if, for example, $\theta_{2,1} = 1$, then there is cancellation on both sides of the model, and it reduces to $(1 - B)Y_t = (1 - \theta_{1,1}B)e_t$. Surprisingly, this can happen even with strongly seasonal data. If it does, as it will for the retail sales, it suggests considering a model outside the ARIMA class.

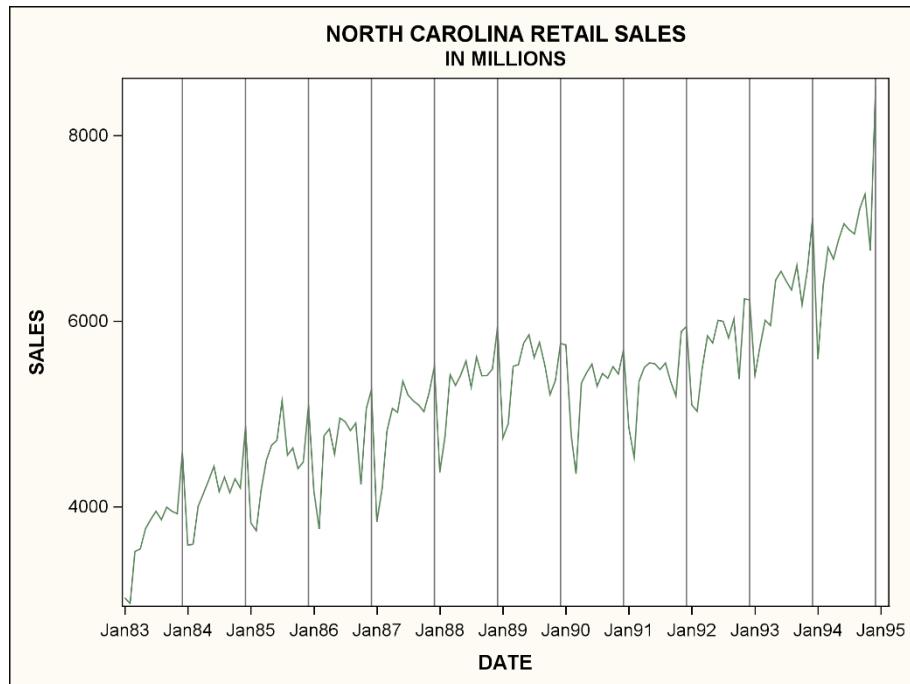
Consider a model $Y_t = \mu + S_t + Z_t$ where $S_t = S_{t-12}$, and Z_t has some ARIMA structure, perhaps even having unit roots. Note that S_t forms an exactly repeating seasonal pattern, as would be modeled using dummy variables. Because of S_t , the autocorrelation function will have spikes at lag 12, as will that of the ordinary first differences because $S_t - S_{t-1}$ is also periodic. However, the span 12 difference $Y_t - Y_{t-12}$ will involve $(1 - B^{12})Z_t$. Unless Z_t has a unit root at lag 12, estimates of the coefficient of Z_{t-12} will be forced toward the moving average boundary. This overdifferencing often results in failure to converge.

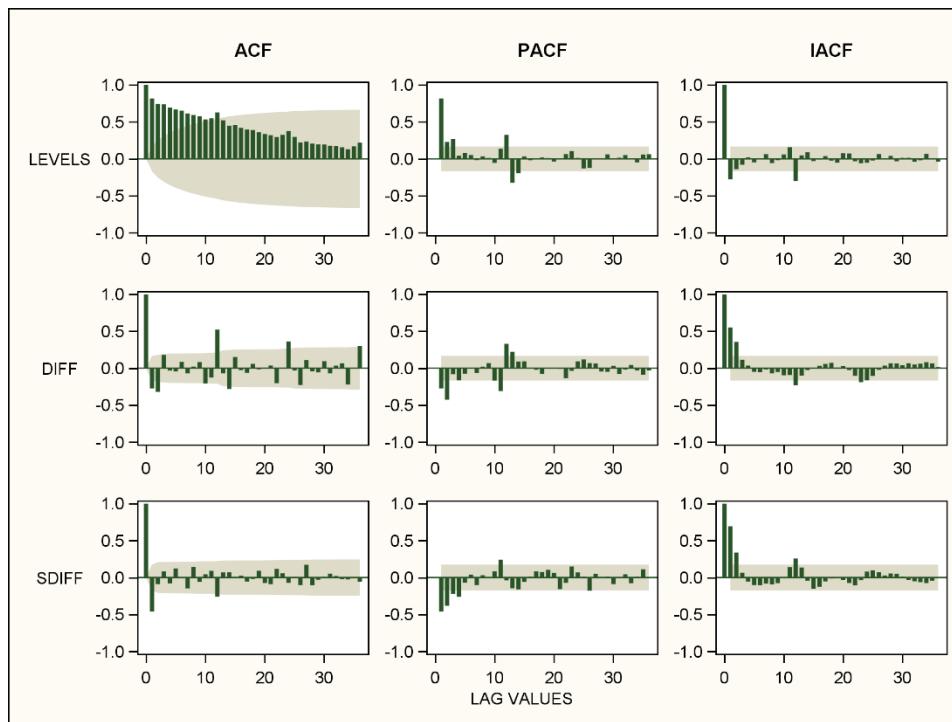
You issue the following SAS statements to plot the data and compute the ACF of the original series, first differenced series, and first and seasonally differenced series:

```
title 'NORTH CAROLINA RETAIL SALES';
title2 'IN MILLIONS';
proc sgplot data=ncretail (where=(date lt '01jan95'd));
  series x=date y=sales;
  refline refdate / axis=x;
  xaxis valuesformat=monyy5.;
quit;
proc arima data=ncretail ;
  identify var=sales outcov=levels nlag=36;
  identify var=sales(1) outcov=diff nlag=36;
  identify var=sales(1,12) outcov=seas nlag=36;
run;
```

The data plot is shown in **Output 4.26**. The ACF, IACF, and PACF have been saved with the OUTCOV=option. **Output 4.27** uses this with SAS/GRAFH and a template to produce a matrix of plots with rows representing original, (1), and (1,12) differenced data and columns representing (from left to right) the ACF, IACF, and PACF.

Output 4.26: Plotting the Original Data



Output 4.27: Computing the ACF with the IDENTIFY Statement: PROC ARIMA

The plot of the data displays nonstationary behavior (nonconstant mean). The original ACF shows slow decay, indicating a first differencing. The ACF of the differenced series shows slow decay at the seasonal lags, indicating a possible span 12 difference. The Q statistics and ACF on the SALES(1,12) differenced variable indicate that some MA terms are needed, with the ACF spikes at 1 and 12 indicating MA terms at lags 1 and 12. Heeding the remarks at the beginning of this section, you try a multiplicative structure, even though the expected side lobes at 11 and 13 (that such a structure implies) are not evident in the ACF. Such a structure serves as a check on the differencing, as you will see. To the previous code, add:

```
estimate q=(1) (12) ml;
```

This requests that maximum likelihood estimates of the multiplicative MA be fitted to the first and span 12 differenced data. The results are in **Output 4.28**.

Output 4.28: Fitting the Multiplicative MA Structure

Warning: The model defined by the new estimates is unstable. The iteration process has been terminated.

Warning: Estimates may not have converged.

ARIMA Estimation Optimization Summary	
Estimation Method	Maximum Likelihood
Parameters Estimated	3
Termination Criteria	Maximum Relative Change in Estimates
Iteration Stopping Value	0.001
Criteria Value	69.27795
Maximum Absolute Value of Gradient	365546.8
R-Square Change from Last Iteration	0.13941
Objective Function	Log Gaussian Likelihood
Objective Function Value	-923.09
Marquardt's Lambda Coefficient	0.00001
Numerical Derivative Perturbation Delta	0.001

ARIMA Estimation Optimization Summary	
Iterations	8
Warning Message	Estimates may not have converged.

Maximum Likelihood Estimation					
Parameter	Estimate	Standard Error	t Value	Approx Pr > t	Lag
MU	-0.65905	1.52987	-0.43	0.6666	0
MA1,1	0.74136	0.05997	12.36	<.0001	1
MA2,1	0.99979	83.86694	0.01	0.9905	12

Constant Estimate	-0.65905
Variance Estimate	62632.24
Std Error Estimate	250.2643
AIC	1852.18
SBC	1860.805
Number of Residuals	131

Correlations of Parameter Estimates			
Parameter	MU	MA1,1	MA2,1
MU	1.000	0.205	-0.117
MA1,1	0.205	1.000	-0.089
MA2,1	-0.117	-0.089	1.000

Autocorrelation Check of Residuals									
To Lag	Chi-Square	DF	Pr > ChiSq	Autocorrelations					
6	9.12	4	0.0582	-0.131	-0.056	0.175	0.046	0.115	0.032
12	16.16	10	0.0950	0.047	0.115	0.062	0.120	-0.047	0.114
18	22.96	16	0.1149	0.009	0.003	0.211	0.014	-0.001	-0.014
24	33.26	22	0.0583	0.075	-0.026	-0.074	0.163	0.087	-0.137
30	37.20	28	0.1144	0.028	-0.055	0.103	-0.092	-0.023	-0.007
36	44.61	34	0.1053	0.031	-0.010	-0.060	0.024	-0.031	-0.185

Model for variable SALES	
Estimated Mean	-0.65905
Period(s) of Differencing	1,12

Moving Average Factors	
Factor 1:	1 - 0.74136 B**(1)
Factor 2:	1 - 0.99979 B**(12)

In **Output 4.28**, you see that there seems to be a problem. The procedure had trouble converging, the standard error on the lag 12 coefficient is extremely large, and the estimate itself is almost 1, indicating a possibly noninvertible model. You can think of a near 1.00 moving average coefficient at lag 12 as trying to undo the span 12 differencing. Of course, trying to make inferences when convergence has not been verified is, at best, questionable. Returning to the discussion at the beginning of this section, a possible explanation is that the seasonality S_t is regular enough to be accounted for by seasonal dummy variables. That scenario is consistent with all that has been observed about these data. The first difference plus dummy variable model of section 1 did seem to fit the data pretty well.

The dummy variables can be incorporated in PROC ARIMA using techniques in **section 4.2**. Letting $S_{1,t}$ through $S_{12,t}$ denote monthly indicator variables (dummy variables), your model is as follows:

$$Y_t = \alpha + \beta t + \delta_1 S_{1,t} + \delta_2 S_{2,t} + \cdots + \delta_{11} S_{11,t} + Z_t$$

From your previous modeling, Z_t seems to have a (nonsseasonal) unit root. You interpret $\alpha + \beta t$ as a December line in that, for December, each S_{jt} is 0. For January, the expected value of Y is $(\alpha + \delta_1) + \beta t$. That is, δ_1 is a shift in the trend line that is included for all January data, and similar δ_j values allow shifts for the other 10 months up through November. Because Christmas sales are always relatively high, you anticipate that all these δ s and especially δ_1 will be negative.

Using ∇ to denote a first difference, write the model at time t and at time $t - 1$, and subtract to get the following:

$$\nabla Y_t = \nabla \alpha + \beta (\nabla t) + \delta_1 \nabla S_{1,t} + \delta_2 \nabla S_{2,t} + \cdots + \delta_{11} \nabla S_{11,t} + \nabla Z_t$$

Now, ∇Z_t is stationary if Z_t has a unit root, $\nabla \alpha = \alpha - \alpha = 0$ and $\nabla t = t - (t - 1) = 1$. Because errors should be stationary for proper modeling in PROC ARIMA, the model is specified in first differences as follows:

$$\nabla Y_t = \beta + \delta_1 \nabla S_{1,t} + \delta_2 \nabla S_{2,t} + \cdots + \delta_{11} \nabla S_{11,t} + \nabla Z_t$$

The parameters have the same interpretations as before. This code fits the model with ∇Z_t specified as ARMA(2,1) and plots forecasts. The data set had 24 missing values for sales at the end with seasonal indicator variables S_{jt} nonmissing. The seasonal indicator variables can be generated without error, so they are valid deterministic inputs.

The following code produces **Output 4.29** and **Output 4.30**:

```
proc arima data=ncretail plots(only)=forecast(forecast);
  identify var=sales(1) crosscor=(s1(1) s2(1) s3(1) s4(1) s5(1)
    s6(1) s7(1) s8(1) s9(1) s10(1) s11(1))
    noprint;
  estimate input=(s1 s2 s3 s4 s5 s6 s7 s8 s9 s10 s11) p=2 q=1 ml;
  forecast lead=24 out=out1 id=date interval=month;
quit;
```

Output 4.29: Seasonal Model for North Carolina Retail Sales

Maximum Likelihood Estimation							
Parameter	Estimate	Standard Error	t Value	Approx Pr > t	Lag	Variable	Shift
MU	26.82314	5.89090	4.55	<.0001	0	SALES	0
MA1,1	0.50693	0.14084	3.60	0.0003	1	SALES	0
AR1,1	-0.39666	0.14155	-2.80	0.0051	1	SALES	0
AR1,2	-0.29811	0.12524	-2.38	0.0173	2	SALES	0
NUM1	-1068.4	95.36731	-11.20	<.0001	0	S1	0
NUM2	-1092.1	93.36419	-11.70	<.0001	0	S2	0
NUM3	-611.48245	82.36315	-7.42	<.0001	0	S3	0
NUM4	-476.60662	89.45164	-5.33	<.0001	0	S4	0
NUM5	-396.94536	90.59402	-4.38	<.0001	0	S5	0
NUM6	-264.63164	87.94063	-3.01	0.0026	0	S6	0
NUM7	-371.30277	90.53160	-4.10	<.0001	0	S7	0
NUM8	-424.65711	88.92377	-4.78	<.0001	0	S8	0
NUM9	-440.79196	81.05429	-5.44	<.0001	0	S9	0
NUM10	-642.50812	92.86014	-6.92	<.0001	0	S10	0
NUM11	-467.54818	94.61205	-4.94	<.0001	0	S11	0

Constant Estimate	45.45892
Variance Estimate	57397.62
Std Error Estimate	239.578
AIC	1988.001
SBC	2032.443
Number of Residuals	143

Autocorrelation Check of Residuals									
To Lag	Chi-Square	DF	Pr > ChiSq	Autocorrelations					
6	1.88	3	0.5973	-0.009	-0.021	0.008	0.001	0.108	0.016
12	7.55	9	0.5801	0.034	0.119	0.057	0.108	-0.076	0.024
18	16.08	15	0.3770	0.013	0.030	0.222	0.030	0.014	-0.037
24	27.84	21	0.1446	0.013	-0.040	-0.020	0.168	0.096	-0.169

Model for variable SALES	
Estimated Intercept	26.82314
Period(s) of Differencing	1

Autoregressive Factors	
Factor 1:	$1 + 0.39666 B^{**}(1) + 0.29811 B^{**}(2)$

Moving Average Factors	
Factor 1:	$1 - 0.50693 B^{**}(1)$

Input Number 1	
Input Variable	S1
Period(s) of Differencing	1
Overall Regression Factor	-1068.41

Input Number 2	
Input Variable	S2
Period(s) of Differencing	1
Overall Regression Factor	-1092.13

Input Number 3	
Input Variable	S3
Period(s) of Differencing	1
Overall Regression Factor	-611.482

Input Number 4	
Input Variable	S4
Period(s) of Differencing	1
Overall Regression Factor	-476.607

Input Number 5	
Input Variable	S5
Period(s) of Differencing	1
Overall Regression Factor	-396.945

Input Number 6	
Input Variable	S6
Period(s) of Differencing	1
Overall Regression Factor	-264.632

Input Number 7	
Input Variable	S7
Period(s) of Differencing	1
Overall Regression Factor	-371.303

Input Number 8	
Input Variable	S8
Period(s) of Differencing	1
Overall Regression Factor	-424.657

Input Number 9	
Input Variable	S9
Period(s) of Differencing	1
Overall Regression Factor	-440.792

Input Number 10	
Input Variable	S10
Period(s) of Differencing	1
Overall Regression Factor	-642.508

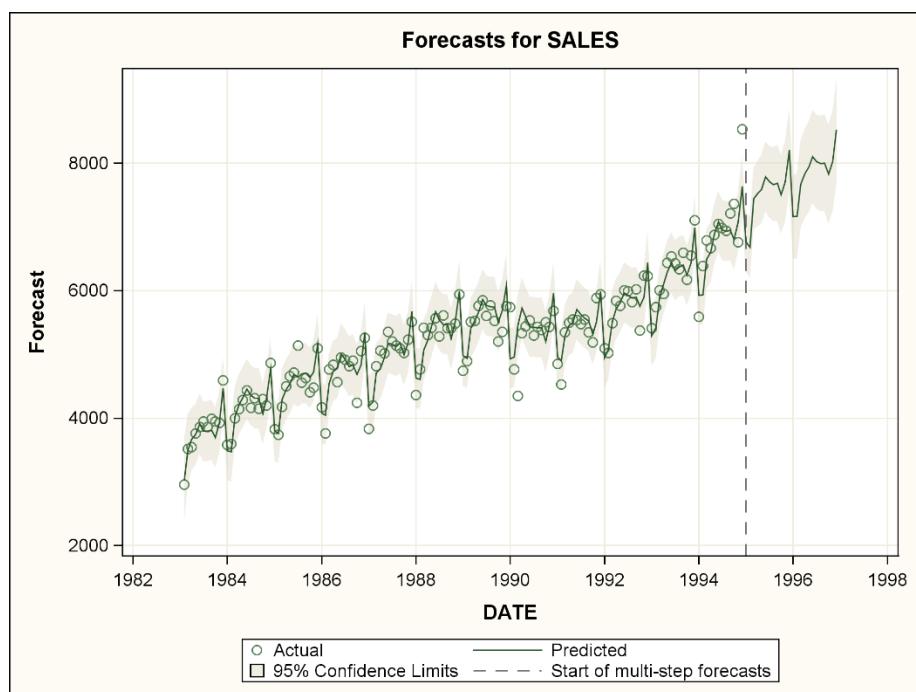
Input Number 11	
Input Variable	S11
Period(s) of Differencing	1
Overall Regression Factor	-467.548

Forecasts for variable SALES				
Obs	Forecast	Std Error	95% Confidence Limits	
145	6769.7677	239.5780	6300.2034	7239.3320
146	6677.4766	240.6889	6205.7350	7149.2182
147	7438.1332	243.5997	6960.6865	7915.5799
148	7527.8413	261.9620	7014.4053	8041.2774
149	7587.4027	270.8251	7056.5952	8118.2102
150	7786.6130	277.8644	7242.0088	8331.2172
151	7704.8579	287.2914	7141.7771	8267.9387
152	7667.1369	295.8507	7087.2802	8246.9935
153	7682.8322	303.6424	7087.7040	8277.9604
154	7509.2889	311.5971	6898.5697	8120.0081
155	7709.0439	319.3641	7083.1018	8334.9861
156	8203.8173	326.8401	7563.2225	8844.4121
157	7162.6783	334.1872	6507.6835	7817.6731
158	7165.4868	341.3916	6496.3715	7834.6021
159	7672.9379	348.4302	6990.0272	8355.8485
160	7834.7312	355.3315	7138.2942	8531.1682
161	7941.1827	362.1054	7231.4692	8650.8962

Forecasts for variable SALES				
Obs	Forecast	Std Error	95% Confidence Limits	
162	8100.3044	368.7526	7377.5626	8823.0463
163	8020.4722	375.2818	7284.9333	8756.0111
164	7993.9393	381.7001	7245.8207	8742.0578
165	8004.6236	388.0121	7244.1338	8765.1133
166	7829.7326	394.2228	7057.0701	8602.3952
167	8031.5161	400.3374	7246.8692	8816.1629
168	8525.8866	406.3599	7729.4359	9322.3374

Output 4.30 shows the resulting graph.

Output 4.30: Forecasts from Seasonal Model



4.4.2 Construction Series Revisited

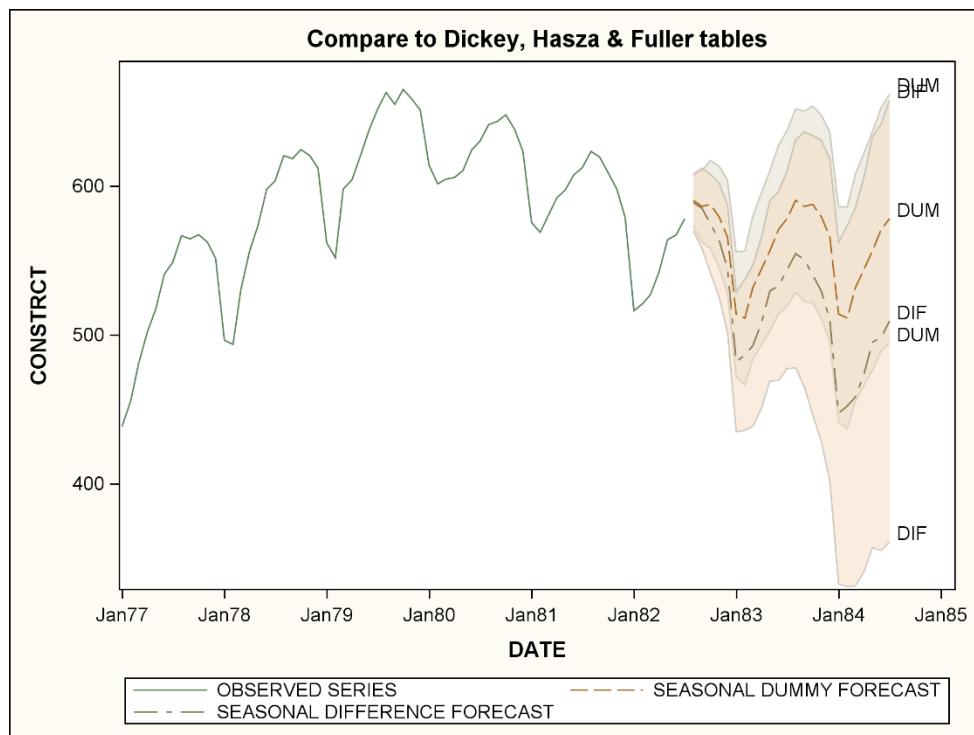
Returning to the construction worker series at the beginning of **section 4.1.2**, you can fit two models, both having a first difference. Let one incorporate a seasonal difference and the other incorporate seasonal dummy variables S1 through S12 to model the seasonal pattern. This code produces two forecast data sets, OUTDUM and OUTDIF, that have 24 forecasts from the two models. The data set ALL has the original construction data, along with seasonal dummy variables S1 through S12 that extend 24 periods into the future. In **section 4.4.1**, the December indicator S12 was dropped to avoid a collinearity problem involving the intercept. An equally valid approach is to drop the intercept (NOCONSTANT) and retain all 12 seasonal indicators. That approach is used here:

```
proc arima data=all;
  identify var=constuct nlag=36 noprint;
  identify var=constuct(1) stationarity=(adf=(0,1,2,3) dlag=12);
  identify var=constuct(1) noprint
    crosscor = (s1(1) s2(1) s3(1) s4(1) s5(1) s6(1)
                s7(1) s8(1) s9(1) s10(1) s11(1) s12(1));
  estimate input = (s1 s2 s3 s4 s5 s6 s7 s8 s9 s10 s11 s12 )
    noconstant method=ml noprint;
  forecast lead=24 id=date interval=month out=outdum;
  identify var=constuct(1,12) noprint;
```

```
estimate noconstant method=ml noint;
forecast lead=24 interval=month id=date out=outdif noint;
run;
```

In **Output 4.31**, the forecast data sets have been merged. Forecasts 24 periods ahead have been plotted. The forecasts and intervals for the span 12 differenced series are shown as dash-dotted lines labeled DIF, and those for the dummy variable model are shown as dashed lines labeled DUM on the far right. The forecasts are quite different. The seasonally differenced series gives much wider intervals and a general pattern of decline. The seasonal dummy variables produce forecast intervals that are less pessimistic and, 24 periods into the future, are about half the width of the others. Of course, wide intervals are expected with differencing. Upper forecast limits are close to each other at the end of the forecast period, and the lower DUM limit is quite close to the DIFF forecast. The lower DIFF limit is much further down. Is there a way to see which model is more appropriate? The chi-square statistics for both models show no problems with the models. Note the code STATIONARITY=(ADF=(0,1,2,3) DLAG=12) for the first differenced series. This DLAG=12 option requests a seasonal unit root test. Dickey, Hasza, and Fuller (1984) develop this and other seasonal unit root tests.

Output 4.31: Seasonal Dummy and Seasonal Difference Forecasts



Output 4.32 shows the results.

Output 4.32: Seasonal Unit Root Tests for Construction Data

Seasonal Augmented Dickey-Fuller Unit Root Tests					
Type	Lags	Rho	Pr < Rho	Tau	Pr < Tau
Zero Mean	0	-7.1995	0.1458	-1.98	0.0321
	1	-6.2101	0.1810	-1.77	0.0499
	2	-7.4267	0.1389	-2.09	0.0251
	3	-7.7560	0.1291	-2.20	0.0198
Single Mean	0	-6.5816	0.2238	-1.79	0.0740
	1	-5.7554	0.2606	-1.61	0.0991
	2	-6.9994	0.2068	-1.95	0.0541
	3	-7.3604	0.1930	-2.07	0.0428

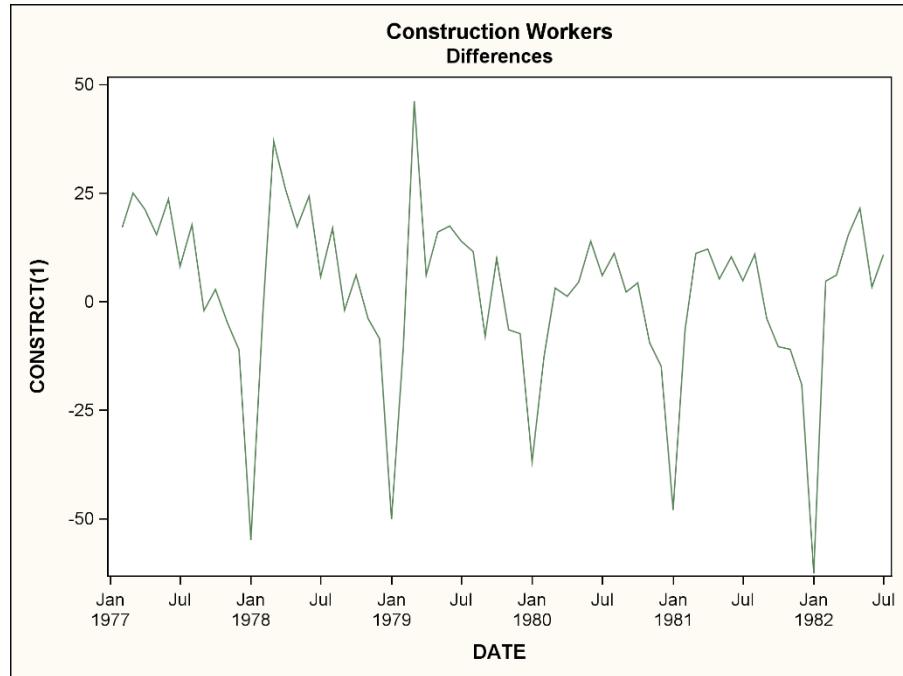
The seasonal dummy variable model does not lose as much data to differencing, is a little easier to understand, has narrower intervals, and does more averaging of past seasonal behavior. In fact, the first and span 12 difference model has forecast $\hat{Y}_t = Y_{t-1} + (Y_{t-12} - Y_{t-13})$, so the forecast for this August is just this July's value with last year's July-to-August change added in. The forecast effectively makes a copy of last year's seasonal pattern and attaches it to the end of the series as a forecast. Without moving average terms, last year's pattern alone gives the forecast.

Only the single mean tests in **Output 4.32** are relevant, as the data clearly have a nonzero mean. The Lags column counts the number of augmenting lags p (lagged seasonal differences) in seasonal multiplicative models. More details follow. With three of these lags there is mild evidence in favor of stationary seasonality. This is in contrast to the analysis in **Output 4.2**, in which the first and span 12 difference model seemed to fit the data well without augmenting lags. Notice, however, that with 0, 1, or 2 augmenting lags, the evidence against the first and span 12 differences model is not significant, agreeing with **Output 4.2**.

While these seasonal unit root tests can sometimes be helpful, you must carefully consider the alternative hypothesis to ensure that the test is in fact addressing the question of interest. The model here does not allow deterministic seasonal behavior under the null or the alternative hypothesis. In **Output 4.32**, the model without augmenting lags ($ADF=(0)$) is $(Y_t - \mu) = \alpha(Y_{t-12} - \mu) + e_t$, where μ is either assumed to be 0 or estimated. The tests shown are from regressions of $Y_t - Y_{t-1}$ on Y_{t-1} , without or with an intercept. Notably, no seasonal dummy variables are included. The test has no option to capture seasonality with dummy variables—that is, with stationary residuals around periodic seasonal effects.

There is, however, a way to test unit root seasonality versus stationary residuals around a periodic seasonal pattern. This is the alternative of interest here. Assume monthly data for illustration. Start with $p = 0$ (using $ADF=(0)$), so the model is now $Y_t - S_j = \alpha(Y_{t-12} - S_j) + e_t$. If $\alpha = 1$, then the S_j values cancel out, giving $Y_t = \alpha Y_{t-12} + e_t$, a seasonal random walk. In contrast, if $|\alpha| < 1$, then $Y_t = S_j + Z_t$, where $Z_t = \alpha Z_{t-12} + e_t$, a regression model with seasonal dummy variables (to model the S_j effects) and seasonal autoregressive errors that can be fit in PROC AUTOREG or PROC ARIMA. That is, inserting $Y_t - S_j = Z_t$ into $Z_t = \alpha Z_{t-12} + e_t$ gives the original model $Y_t - S_j = \alpha(Y_{t-12} - S_j) + e_t$. In the construction worker data, first differences are taken and tested for an additional seasonal unit root. That is, Y_t is the series of first differenced data. **Output 4.33** shows the first differences. **Output 4.2** showed no evidence that ARMA terms are needed once the first and span 12 difference is taken. Therefore, the following code will produce the test. Here D is $Y_t - Y_{t-12}$ and LAG12 is Y_{t-12} . PROC GLM uses monthly dummy variables and Y_{t-12} to produce the test statistic shown in **Output 4.34**, but special tables are needed to judge significance.

```
proc glm data=const;
  class month;
  model D=month lag12/solution;
  output out=deseasonalized residual=Y;
  title "Compare to Dickey, Hasza & Fuller tables";
  estimate "Seasonal_tau" lag12 1;
run;
```

Output 4.33: First Differences of Construction Data**Output 4.34 Unit Root Versus Seasonal Dummy Variable Model.**

Parameter	Estimate	Standard Error	t Value	Pr > t
Seasonal_tau	-0.75421420	0.16291118	-4.63	<.0001

Because the test statistic -4.63 does not have a t distribution, the reported p -value is not appropriate and is struck through in the output. Instead, Dickey, Hasza, and Fuller (1984, Table 7) shows tenth percentile $t = -5.49$, regardless of sample size. Although the smallest sample size in that table is 120, it suggests that the true p -value for -4.63 is larger than 0.10. There is not significant evidence against the seasonal unit root. To get a more accurate p -value, 2 million simulated time series with seasonal unit roots (not shown) were run. The t test was less than -4.63 about 29.4% of the time, giving a correct p -value of 0.294. This is consistent with a seasonal unit root even when seasonal dummy variables are present. Once the alternative of interest is included, there is much less evidence against the seasonal unit root.

The seasonal multiplicative model for an arbitrary number of augmenting lags p is as follows:

$$(1 - \alpha B^{12})(1 - \gamma_1 B - \gamma_2 B^2 - \gamma_3 B^3 - \dots - \gamma_p B^p)(Y_t - S_j) = e_t$$

The product of backshift polynomials gives rise to the name *multiplicative seasonal model*, and p is the entry in the Lags column of **Output 4.32**.

If $\Phi_t = (1 - \gamma_1 B - \gamma_2 B^2 - \gamma_3 B^3 - \dots - \gamma_p B^p)(Y_t - S_j)$, then the model is expressible as $(1 - \alpha B^{12})\Phi_t = e_t$. Clearly, if Φ_t (known as a “filtered” version of $(Y_t - S_j)$) were known, then the same regression just discussed could be used. The null hypothesis is $\alpha = 1$ as usual. Under that hypothesis, $(1 - \gamma_1 B - \gamma_2 B^2 - \gamma_3 B^3 - \dots - \gamma_p B^p)(1 - \alpha B^{12})(Y_t - S_j) = (1 - \gamma_1 B - \gamma_2 B^2 - \gamma_3 B^3 - \dots - \gamma_p B^p)(Y_t - Y_{t-12}) = e_t$. This AR(p) model for the span 12 differences suggests that initial estimates of the γ_j coefficients can be obtained by regressing the span 12 difference $D_t = (Y_t - Y_{t-12})$ on p of its lags. When this was done on the construction data, no lags were significant, consistent with the use of the Lags = 0 row of **Output 4.34**.

The residuals r_t from this regression serve as estimates of e_t , and the estimated coefficients—call them $g_j, j = 1, 2, \dots, p$ —are next used to approximate Φ_t by F_t , where F_t is calculated by substituting g_j for γ_j and substituting sample seasonal means for S_j . Let $D_t = Y_t - Y_{t-12}$. Now regress the residuals r_t on $F_{t-12}, D_{t-1}, D_{t-2}, \dots, D_{t-p}$. The coefficients on the lagged D s give one-step Taylor series improvements for the g_j estimates and the regression coefficient on F_{t-12} estimates $\alpha - 1$. The studentized test (t test) for $\alpha - 1$ is compared with the same distribution that was used for the $p = 0$ case. Although the subtraction of seasonal means from the data initially is not quite the same as using seasonal dummy variables, the Dickey, Hasza, and Fuller table provides a reasonable approximation for the general case with augmenting lags.

4.4.3 Milk Scare (Intervention)

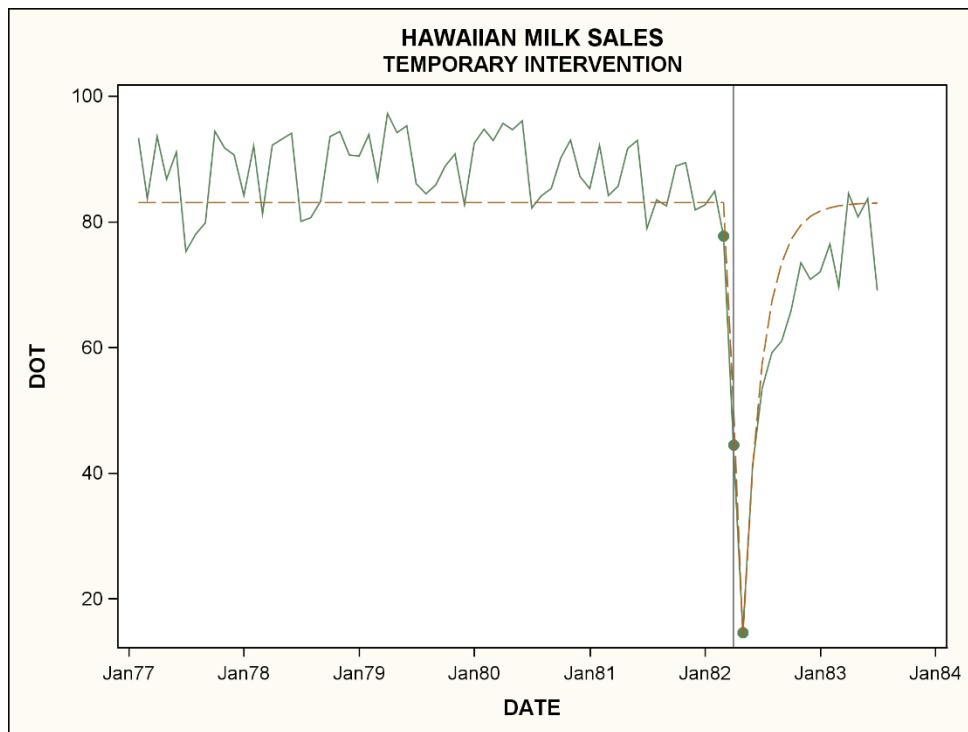
Liu et al. (1998) discuss milk sales in Oahu, Hawaii, during a time period in which the discovery of high pesticide levels in milk was publicized. Liu (through personal communication) provided the data here. The data indicate April 1982 as the month of first impact, although some tainted milk was found in March. **Output 4.35** shows a graph with March, April, and May 1982 indicated by dots. Ultimately, eight recalls were issued and publicized, with more than 36 million pounds of contaminated milk found. It might be reasonable to expect a resulting drop in milk sales that might have a long-term effect. It appears that, with the multiple recalls and escalating publicity, the full impact was not realized until May 1982, after which recovery began.

Initially, a model was fit to the data before the intervention. A seasonal pattern was detected, but no ordinary or seasonal differencing seemed necessary. A P=(1)(12) specification left a somewhat large correlation at lag 2, so Q=(2) was added, and the resulting model fit the pre-intervention data nicely. The intervention response showed an arbitrary value after the first drop, in fact another drop, followed by exponential increase upward. The second drop suggests a numerator lag, and the exponential increase suggests a denominator lag in the transfer function operator. X is a variable that is 1 for April 1982 and 0 otherwise. The following code produces an intervention model with this pattern:

```
proc arima data=liu;
  identify var=sales noprint crosscor=(x);
  estimate input=( (1) /(1) x ) p=(1)(12) q=(2) method=ml;
run;
```

Output 4.35 and **Output 4.36** show the results.

Output 4.35: Effect of Tainted Milk



Output 4.36: Model for Milk Sales Intervention: Effect of Negative Publicity**The ARIMA Procedure**

Maximum Likelihood Estimation							
Parameter	Estimate	Standard Error	t Value	Approx Pr > t	Lag	Variable	Shift
MU	83.01547	3.98012	20.86	<.0001	0	SALES	0
MA1,1	-0.34929	0.11980	-2.92	0.0035	2	SALES	0
AR1,1	0.53417	0.10566	5.06	<.0001	1	SALES	0
AR2,1	0.78929	0.06964	11.33	<.0001	12	SALES	0
NUM1	-39.89628	2.83209	-14.09	<.0001	0	X	0
NUM1,1	49.55974	2.96258	16.73	<.0001	1	X	0
DEN1,1	0.61051	0.03289	18.56	<.0001	1	X	0

Constant Estimate	8.148306
Variance Estimate	15.88346
Std Error Estimate	3.985406
AIC	450.5909
SBC	466.9975
Number of Residuals	77

Autocorrelation Check of Residuals									
To Lag	Chi-Square	DF	Pr > ChiSq	Autocorrelations					
6	3.98	3	0.2636	-0.027	0.030	-0.015	0.110	0.019	0.180
12	14.35	9	0.1104	0.116	0.141	0.149	0.000	0.235	-0.063
18	18.96	15	0.2157	0.052	0.133	-0.006	0.005	0.154	-0.046
24	23.01	21	0.3434	0.118	0.064	0.035	0.017	0.010	-0.131

By specifying INPUT=((1)/(1) X), where X is 1 for April 1982 and 0 otherwise, you are fitting an intervention model whose form is $(\beta_0 - \beta_1 B) / (1 - \alpha_1 B) X_t$.

Filling in the estimates, you have the following:

$$\begin{aligned} (-40 - 50B) / (1 - 0.61B) X_t &= (-40 - 50B)(1 + 0.61B + 0.61^2 B^2 + \dots) X_t \\ &= -40X_t - 74.4X_{t-1} + 0.61(-74.4)X_{t-2} + .061^2(-74.4)X_{t-3} + \dots \end{aligned}$$

So, when X_t is 1, the estimated effect is -40. The next month, X_{t-1} is 1, and the effect is -74.4. Two months after the intervention, the estimated effect is 0.61(-74.4) as recovery begins. This model forces a return to the original level. In **Output 4.35**, a horizontal line at the intercept 83 has been drawn, and the intervention effects -40, -74.4, and so on, have been added in. Notice how the intercept line underestimates the pre-intervention level, and how the estimated recovery seems faster than the data suggest. Had you plotted the forecasts, including the autoregressive components, this failure of the mean structure in the model might not have been noticed. The importance of plotting cannot be overemphasized. It is a critical component of data analysis. The statistics in **Output 4.36** give no warning signs of any problems. Again, you might think of the autoregressive structure as compensating for some lack of fit.

Might there be some permanent effect of this incident? The model now under consideration does not allow it. To investigate this, you add a level shift variable.

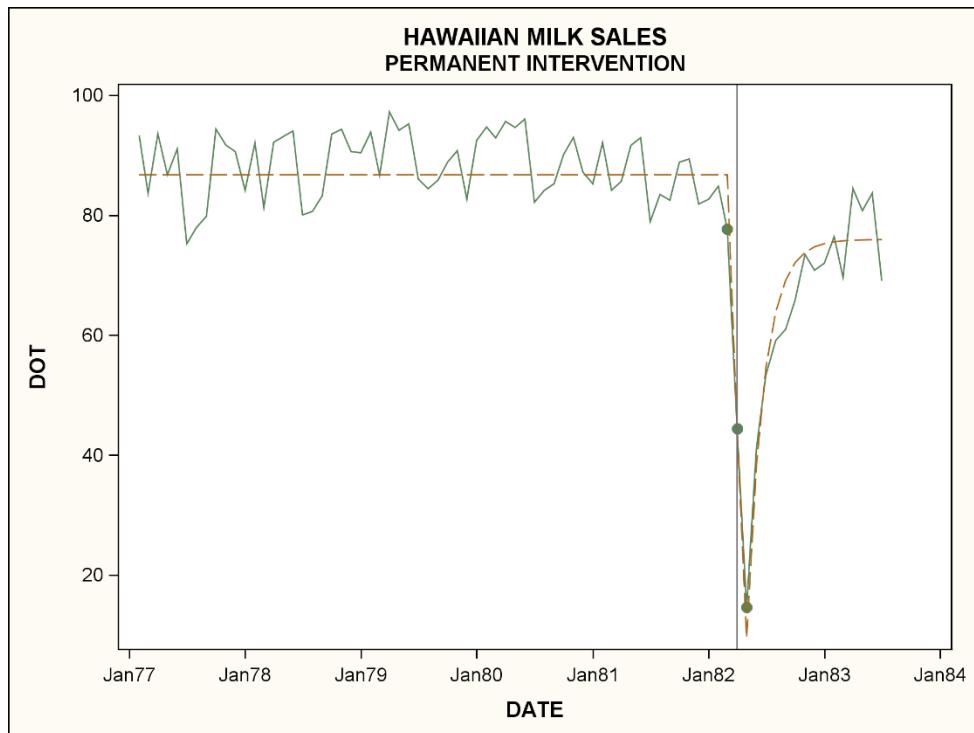
Define the variable LEVEL1 to be 1 prior to April 1982 and 0 otherwise. This adds a constant, the coefficient of the column, for the pre-intervention period. It represents the difference between the pre-intervention mean and the level to which the post-intervention trend is moving—that is, the level attained long after the intervention. If this shift is not significantly different from 0, then the model shows no permanent effect. If the shift (coefficient) is significantly larger

than 0, then a permanent decrease in sales is suggested by the model. If the coefficient happens to be negative, then the pre-intervention level is less than the level toward which the data are now moving. You issue the following code to fit a model with both temporary effects (X) and a permanent level shift (LEVEL1):

```
proc arima;
  identify var=sales noprint crosscor=(x level1);
  estimate input=((1)/(1)x level1) p=(1) (12) q=(2) method=ml;
run;
```

Output 4.37 and Output 4.38 show the results.

Output 4.37: Model Allowing Permanent Effect



Output 4.38: Intervention Model with Permanent Shift

Maximum Likelihood Estimation							
Parameter	Estimate	Standard Error	t Value	Approx Pr > t	Lag	Variable	Shift
MU	75.95488	2.71752	27.95	<.0001	0	SALES	0
MA1,1	-0.31251	0.12035	-2.60	0.0094	2	SALES	0
AR1,1	0.29447	0.11635	2.53	0.0114	1	SALES	0
AR2,1	0.77633	0.07042	11.02	<.0001	12	SALES	0
NUM1	-31.67599	3.24331	-9.77	<.0001	0	X	0
NUM1,1	48.55647	2.99268	16.23	<.0001	1	X	0
DEN1,1	0.56564	0.03353	16.87	<.0001	1	X	0
NUM2	10.79115	2.22675	4.85	<.0001	0	LEVEL1	0

Constant Estimate	11.98599
Variance Estimate	13.73471
Std Error Estimate	3.706037
AIC	439.2273
SBC	457.9778
Number of Residuals	77

Autocorrelation Check of Residuals										
To Lag	Chi-Square	DF	Pr > ChiSq	Autocorrelations						
6	2.46	3	0.4830	0.020	-0.010	-0.091	-0.011	-0.090	0.110	
12	9.47	9	0.3948	0.094	0.125	0.099	-0.033	0.188	-0.083	
18	12.28	15	0.6581	-0.010	0.067	-0.069	-0.030	0.114	-0.068	
24	18.29	21	0.6308	0.080	0.029	-0.012	-0.048	-0.072	-0.196	

It appears that the pre-intervention level is about $75.95 + 10.79$ and the ultimate level to which sales will return is 75.95, according to this model. All estimates, including the estimated 10.79 permanent loss in sales, are significant. The geometric rate of approach to the new level is 0.56565, indicating a faster approach to the new level than that from the first model. Of course, at this point, it is clear that the old model was misspecified, as it did not include LEVEL1. The AR1,1 coefficient 0.29 is quite a bit smaller than 0.53 from the first model. That is consistent with the idea that the autoregressive structure there was in part compensating for the poor fit of the mean function. You can add and subtract 1.96(2.2268) from 10.79 to get an approximate 95% confidence interval for the permanent component of the sales loss due to the contamination scare.

Other models can be tried. Seasonal dummy variables might be tried in place of the seasonal AR factor. Liu et al. suggest that some sort of trend might be added to account for a decline in consumer preference for milk. A simple linear trend gives a mild negative slope, but it is not statistically significant. The estimated permanent level shift is about the same and still significant in its presence.

4.4.4 Terrorist Attack

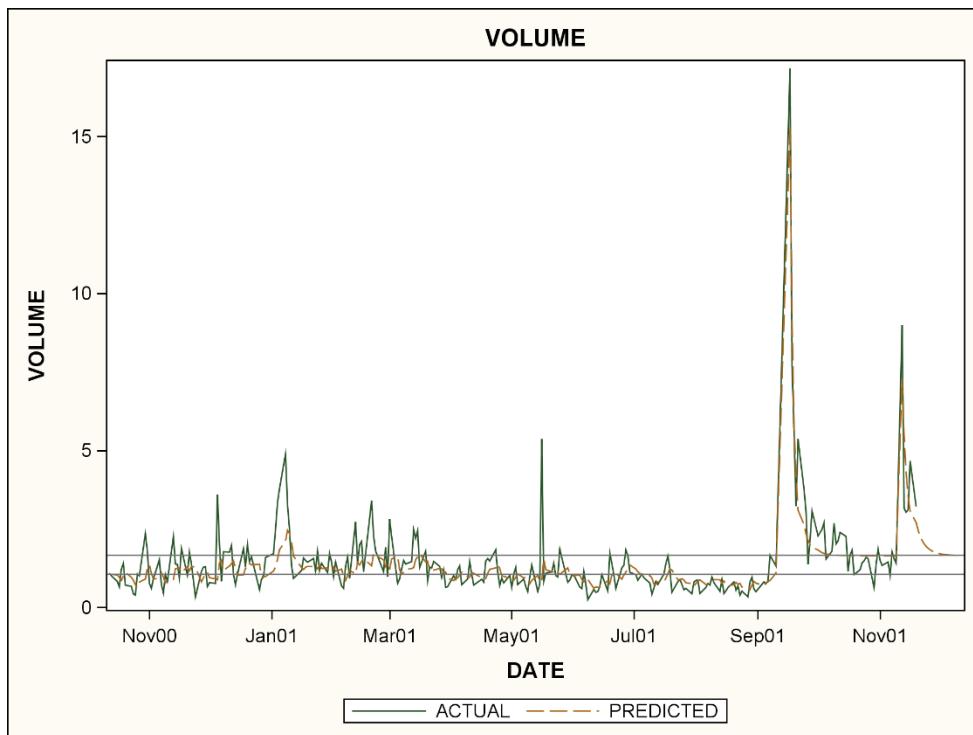
On September 11, 2001, terrorists used commercial airliners as weapons to attack targets in the United States, resulting in the collapse of the World Trade Center in New York City. American Airlines flights were among those involved. The stock market was closed following this incident and reopened September 17. In a second incident, an American Airlines jet crashed on November 12, 2001, in Queens, New York. An intervention analysis of American Airlines stock trading volume (in millions) is now done, incorporating a pulse and level shift intervention for each of these events, defined similarly to those of the milk example in [section 4.4.3](#). Data through November 19 are used here, so there is not a lot of information about the nature of the response to the second incident. A model that seems to fit the data reasonably well, with parameters estimated from PROC ARIMA, is as follows:

$$\log(\text{Volume}) = 0.05 + (2.58 - 2.48B) / (1 - 0.76B) X_t + 1.49 / (1 - 0.80B) P_t + (1 - 0.52B) / (1 - 0.84B) e_t$$

Here, X_t is a level shift variable that is 1 after September 11 and 0 before, and P_t is a pulse variable that is 1 only on the day of the second incident. The p -values for all estimates except the intercept were less than 0.0005. Those for the chi-square check of residuals were all larger than 0.35, indicating an excellent fit for the 275 log-transformed volume values in the data set.

This model allows for a permanent effect of the terrorist attack of September 11, but forces the effect of the second incident to decline exponentially to 0 over time. The second incident sparked a $\log(\text{volume})$ increase 1.49 on the day that it happened, but j days later, $\log(\text{volume})$ is $(0.80)^j(1.49)$ above what it would have otherwise been, according to the model. The permanent effect of the events of September 11 on log volume would be $(2.59 - 2.48)/(1 - 0.76) = 0.46$ according to the model. The numerator lag for X allows a single arbitrary change from the initial shock (followed by an exponential approach at rate 0.76 to the eventual new level). In that sense, the inclusion of this lag acts like a pulse variable and likely explains why the pulse variable for September 11 was not needed in the model. The level shift variable for the second incident did not seem to be needed either, but with so little data after November 12, the existence of a permanent effect remains in question.

Output 4.39 shows a graph of the data and a forecast from this model.

Output 4.39: American Airlines Stock Volume

Calculations from the log model were exponentiated to produce the graph. The model was fit to the full data set, but the option BACK=42 was used in the FORECAST statement so that the data following September 11 were not used to adjust the forecasts. That is, only the X and P parts of the model are used in the post-September 11 forecasts. With that in mind (with no adjustments based on recent residuals), it is striking how closely these forecasts mimic the behavior of the data after this incident. It is also interesting how similar the decay rates (denominator terms) are for the two incidents. Two horizontal lines, one at the pre-intervention level $\exp(0.05) = 1.05$ and one at the ultimate level $\exp(0.05 + (2.59 - 2.48) / (1 - 0.76)) = \exp(0.51) = 1.66$, are drawn. The permanent effect of the event of September 11 is an increase of $(2.59 - 2.48) / (1 - 0.76)$ in log-transformed volume, according to the model. That becomes a multiplicative increase of $\exp((2.59 - 2.48) / (1 - 0.76)) = 1.58$, a 58% increase in volume.

Chapter 5: The ARIMA Model: Special Applications

5.1 Regression with Time Series Errors and Unequal Variances	177
5.1.1 Autoregressive Errors.....	177
5.1.2 Example: Energy Demand at a University.....	178
5.1.3 Unequal Variances	182
5.1.4 ARCH, GARCH, and IGARCH for Unequal Variances.....	184
5.2 Cointegration.....	189
5.2.1 Cointegration and Eigenvalues.....	191
5.2.2 Impulse Response Function.....	192
5.2.3 Roots in Higher-Order Models.....	192
5.2.4 Cointegration and Unit Roots	194
5.2.5 An Illustrative Example	196
5.2.6 Estimation of the Cointegrating Vector	199
5.2.7 Intercepts and More Lags	201
5.2.8 PROC VARMAX	202
5.2.9 Interpretation of the Estimates	205
5.2.10 Diagnostics and Forecasts	206

5.1 Regression with Time Series Errors and Unequal Variances

Regression models can contain a wide variety of inputs, but for ordinary least squares (OLS) estimation, uncorrelated errors are required. Autoregressive and ARMA models take advantage of autocorrelation for forecasting, but most models previously discussed involve a limited collection of inputs. In this chapter, the flexibility of regression and the ability of AR or ARMA models to capture autocorrelation are combined to provide a powerful set of analysis tools. The basic approach is to identify a correlation structure from OLS residuals, then switch from OLS to generalized least squares (GLS) or maximum likelihood (ML) for the final estimation stage.

5.1.1 Autoregressive Errors

The SAS AUTOREG procedure provides a tool to fit a regression model with autoregressive time series errors. Such a model can be written in two steps. With a response Y_t related to a single input X_t and with an AR(1) error, you can write $Y_t = \beta_0 + \beta_1 X_t + Z_t$ and $Z_t = \alpha Z_{t-1} + e_t$.

In this expression, $|\alpha| < 1$ for stationarity and $e_t \sim N(0, \sigma^2)$ with obvious extensions to multiple regression and AR(p) series. The variance of Z_t is $\sigma^2/(1 - \alpha^2)$, from which the normal density function of $Z_1 = Y_1 - \beta_0 - \beta_1 X_1$ can be derived. Furthermore, substitution of $Z_t = Y_t - \beta_0 - \beta_1 X_t$ and its lag into $e_t = Z_t - \alpha Z_{t-1}$ shows the following:

$$e_t = (Y_t - \beta_0 - \beta_1 X_t) - \alpha(Y_{t-1} - \beta_0 - \beta_1 X_{t-1})$$

Because $e_t \sim N(0, \sigma^2)$, the normal density of this expression can also be written for $t = 2, 3, \dots, n$. Because $Z_1, e_2, e_3, \dots, e_n$ are independent of each other, the product of these n normal densities constitutes the so-called joint density function of the Y values. If the observed data values Y and X are plugged into this function, the only unknowns remaining in the function are the parameters α, β_0, β_1 , and σ^2 . The resulting function $L(\alpha, \beta_0, \beta_1, \sigma^2)$ is called the *likelihood function*, with the values of α, β_0, β_1 , and σ^2 that maximize it referred to as *maximum likelihood estimates*. This is the best way to estimate the model parameters. Other methods (described below) have evolved as less computationally burdensome approximations.

From the expression for e_t , you can see that the following is true:

$$Y_t = \alpha Y_{t-1} + \beta_0 (1 - \alpha) + \beta_1 (X_t - \alpha X_{t-1}) + e_t$$

This suggests that you could use some form of nonlinear least squares to estimate the coefficients.

A third, less-used approach to estimate the parameters is much less computer intensive, but it does not make as efficient use of the data as maximum likelihood does. It is called the Cochrane-Orcutt method and consists of (1) running a least squares regression of Y on X , (2) fitting an autoregressive model to the residuals, and (3) using that model to filter the data. You write, as above:

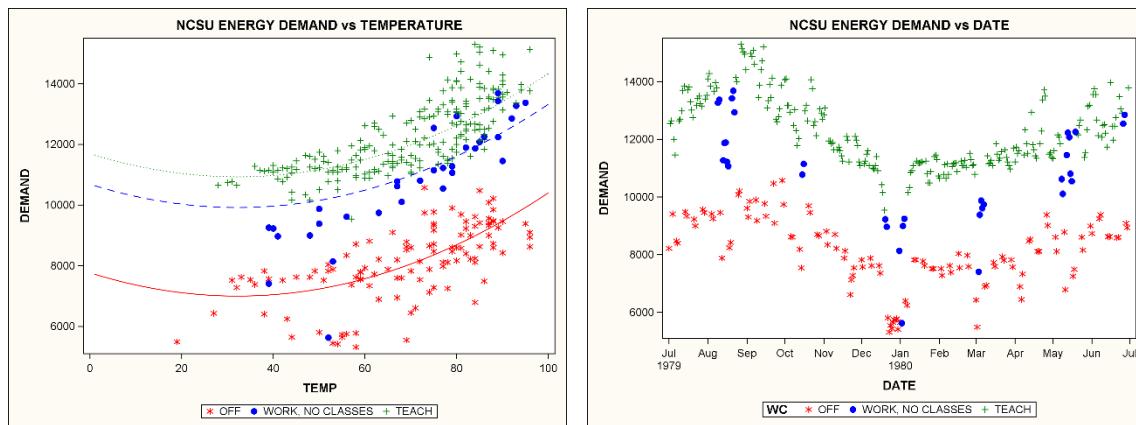
$$Y_t = \alpha Y_{t-1} + \beta_0(1-\alpha) + \beta_1(X_t - \alpha X_{t-1}) + e_t$$

You observe that this is the equation for a regression of the transformed (or filtered) variable $Y_t - \alpha Y_{t-1}$ on transformed variables $(1 - \alpha)$ and $(X_t - \alpha X_{t-1})$. Because e_t satisfies the usual regression properties, this regression, done with OLS, satisfies all of the usual conditions for inference. The resulting estimates of the parameters would be unbiased, with proper standard errors and valid t tests given by the OLS formulas applied to these transformed variables. When α is replaced by an estimate from a model for the residuals, the previous statements are approximately true. The Cochrane-Orcutt method can be modified to include an equation for the first observation as well. The method can be iterated, using the new regression estimates to produce new estimates of Z , and as a result, new estimates of α , and so on. However, the simultaneous iteration of all parameters done by maximum likelihood would generally be preferred. An autoregressive order 1 AR(1) error structure has been used for illustration, but not all error structures are AR(1). Regardless of the method used for parameter estimation, the autocorrelation structure needs to be identified.

If the error autocorrelation is ignored and a regression of Y on X is done, the estimated slope and intercept will be unbiased and will (under general assumptions on X) be consistent. That is, they will converge to their true values as the sample size increases. However, the standard errors reported in this regression, unlike those for the filtered variable regression, will be wrong, as will the p -values and any inference with them. Thus, OLS residuals can be used to estimate the error autocorrelation structure, but the OLS t tests and associated p -values for the intercept and slopes cannot be trusted. In PROC AUTOREG, the user sees the initial OLS regression, the estimated autocorrelation function computed from the residuals, the autoregressive parameter estimates (with insignificant ones being omitted if BACKSTEP is specified), and the final estimation of parameters, including standard errors and tests, that are valid based on large sample theory.

5.1.2 Example: Energy Demand at a University

Output 5.1 shows energy demand plotted against temperature and against date. Data were collected at North Carolina State University during the 1979–1980 academic year. Three plot symbols are used to indicate non-workdays (*), workdays with no classes (●), and teaching days (+). The goal is to relate demand for energy to temperature and type of day. The coefficient of the variable WORK will be seen in **Output 5.4** to be 2919. The variable WORK is 0 for non-workdays and 1 for workdays, indicating that 1(2919) is to be added to every prediction for a workday. A similar 0,1 variable called TEACH has a coefficient of 1011, which indicates that 1(1011) should be added to teaching days. Because all teaching days are workdays, teaching day demand is $2919 + 1011 = 3930$ higher than non-workdays for any given temperature. Workdays that are not teaching days have demand 2919 higher than non-workdays for a given temperature. As temperatures rise, demand increases at an increasing rate. The three curves on the graphs come from a model to be discussed. The plot of demand against date shows that there were a couple of workdays during class break periods (for example, December 31) where demand was more like non-workdays, as you might expect. You might want to group these with the non-workdays. Also, day-of-week dummy variables can be added. A model without these modifications will be used.

Output 5.1: NCSU Energy Demand

The model has today's temperature TEMP, its square TEMPSQ, yesterday's temperature TEMP1, teaching day indicator TEACH, and workday indicator WORK as explanatory variables. Future values of TEACH and WORK would be known, but future values of the temperature variables would have to be estimated to forecast energy demand into the future. Future values of such inputs need to be provided (with missing values for the response) in the data set in order to forecast. No accounting for forecast inaccuracy in future values of the inputs is done by PROC AUTOREG.

To fit the model, issue this code:

```
proc autoreg data=energy plots(only)=acf;
  model demand = temp tempsq teach work temp1
    /nlag= 15 backstep method=ml dwprob;
run;
```

Output 5.2 contains the OLS regression portion of the PROC AUTOREG output.

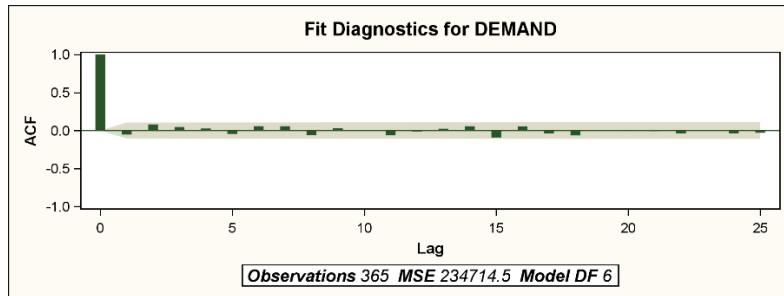
Output 5.2: OLS Regression**The AUTOREG Procedure**

Ordinary Least Squares Estimates			
SSE	231077196	DFE	359
MSE	643669	Root MSE	802.28989
SBC	5947.02775	AIC	5923.62837
MAE	628.751802	AICC	5923.86301
MAPE	6.42812843	HQC	5932.92759
Durbin-Watson	0.5331	Total R-Square	0.8794

Durbin-Watson Statistics			
Order	DW	Pr < DW	Pr > DW
1	0.5331	<.0001	1.0000

NOTE: Pr<DW is the p-value for testing positive autocorrelation, and Pr>DW is the p-value for testing negative autocorrelation.

Parameter Estimates					
Variable	DF	Estimate	Standard Error	t Value	Approx Pr > t
Intercept	1	3593	202.8272	17.71	<.0001
TEMP	1	27.3512	5.7938	4.72	<.0001
TEMPSQ	1	0.7988	0.1458	5.48	<.0001
TEACH	1	1533	150.2740	10.20	<.0001
WORK	1	2685	158.8292	16.91	<.0001
TEMP1	1	34.0833	5.7923	5.88	<.0001



This regression displays strongly autocorrelated residuals r_t . Consider the following Durbin-Watson statistic:

$$DW = \sum_2^n (r_t - r_{t-1})^2 / \sum_1^n r_t^2 = .5331$$

It is significantly less than 2 ($p < 0.0001$), indicating a positive lag 1 autocorrelation in the errors. If r_t and r_{t-1} were alike (strong positive correlation), then $r_t - r_{t-1}$ would be near 0, showing that the Durbin-Watson (DW) statistic tends to be less than 2 under positive autocorrelation. The DW is expected to be near 2 for uncorrelated data. Extensions of DW to lags of more than 1 are available in PROC AUTOREG. A more general approach than DW is given by PROC AUTOREG's automatic AR order selection process as shown in Output 5.3. The autocorrelation plot shows strong autocorrelations. The correction for autocorrelation reveals that lags 7 and 14 are present, indicating some sort of weekly effect. Lag 1 is also sensible, but lags 5 and 12 are a little harder to justify with intuition. All of the other 15 lags that you started with are eliminated automatically by the BACKSTEP option.

Output 5.3: AR Parameter Estimates

Estimates of Autoregressive Parameters				
Lag	Coefficient	Standard Error	t Value	
1	-0.580119	0.040821	-14.21	
5	-0.154947	0.045799	-3.38	
7	-0.157783	0.048440	-3.26	
12	0.127772	0.045687	2.80	
14	-0.116690	0.044817	-2.60	

In PROC AUTOREG, the model for the error Z_t is written with plus signs rather than minus signs—that is, $(1 + \alpha_1 B + \alpha_2 B^2 + \dots + \alpha_p B^p)Z_t = e_t$ is the notation used. Therefore, the AR(14) error model in **Output 5.3** is as follows:

$$Z_t = 0.58Z_{t-1} + 0.15Z_{t-5} + 0.16Z_{t-7} - 0.13Z_{t-12} + 0.12Z_{t-14} + e_t$$

Using this AR(14) structure and these estimates as initial values, the likelihood function is computed and maximized, producing the final estimates with correct (justified by large sample theory) standard errors in **Output 5.4**.

Output 5.4: Final Estimates

Parameter Estimates					
Variable	DF	Estimate	Standard Error	t Value	Approx Pr > t
Intercept	1	4638	407.4300	11.38	<.0001
TEMP	1	23.4711	3.5389	6.63	<.0001
TEMPSQ	1	0.7405	0.1162	6.37	<.0001
TEACH	1	1011	114.2874	8.84	<.0001
WORK	1	2919	115.5118	25.27	<.0001
TEMP1	1	25.0550	3.5831	6.99	<.0001
AR1	1	-0.6490	0.0401	-16.18	<.0001
AR5	1	-0.1418	0.0478	-2.97	0.0032
AR7	1	-0.1318	0.0504	-2.62	0.0093
AR12	1	0.1420	0.0481	2.95	0.0033
AR14	1	-0.1145	0.0469	-2.44	0.0151

Using OLS estimation from **Output 5.2** or from PROC REG, a 95% confidence interval for the effect on energy demand of teaching classes would have been incorrectly computed as $1533 \pm 1.96(150)$, whereas the correct interval is $1011 \pm 1.96(114)$. Using the same regression inputs in PROC ARIMA, a model with $p = (1)$ and $q = (1, 7, 14)$ showed no lack of fit in **Output 5.5**. This error model is more aesthetically pleasing than AUTOREG because it does not include the unusual lags 5 and 12. Note that AUTOREG cannot fit moving average terms.

Output 5.5: PROC ARIMA for Energy Data**The ARIMA Procedure**

Maximum Likelihood Estimation							
Parameter	Estimate	Standard Error	t Value	Approx Pr > t	Lag	Variable	Shift
MU	4468.1	363.30832	12.30	<.0001	0	DEMAND	0
MA1,1	0.25723	0.06257	4.11	<.0001	1	DEMAND	0
MA1,2	-0.19657	0.05391	-3.65	0.0003	7	DEMAND	0
MA1,3	-0.18440	0.05289	-3.49	0.0005	14	DEMAND	0
AR1,1	0.84729	0.03622	23.39	<.0001	1	DEMAND	0
NUM1	25.49771	3.45642	7.38	<.0001	0	TEMP	0
NUM2	0.74724	0.11173	6.69	<.0001	0	TEMPSQ	0
NUM3	838.08859	111.69438	7.50	<.0001	0	TEACH	0
NUM4	3085.4	114.90204	26.85	<.0001	0	WORK	0
NUM5	25.32913	3.44973	7.34	<.0001	0	TEMP1	0

Autocorrelation Check of Residuals									
To Lag	Chi-Square	DF	Pr > ChiSq	Autocorrelations					
6	3.85	2	0.1461	0.018	-0.007	-0.042	-0.052	0.022	0.071
12	11.04	8	0.1995	-0.000	0.008	0.077	0.040	-0.036	-0.101
18	14.65	14	0.4022	-0.007	0.010	-0.019	0.084	-0.014	-0.042
24	15.71	20	0.7347	-0.003	0.011	0.049	-0.002	0.013	-0.005
30	20.87	26	0.7487	0.002	0.074	-0.073	0.043	0.007	0.012
36	28.24	32	0.6575	0.021	-0.017	0.075	0.002	0.106	-0.025
42	36.98	38	0.5163	0.052	0.015	0.050	0.018	0.042	0.117
48	38.90	44	0.6895	-0.014	0.046	-0.033	-0.021	-0.022	-0.016

The effect on energy demand of teaching classes is estimated from PROC ARIMA as 838 with standard error 112, somewhat different from PROC AUTOREG and quite different from the OLS estimates. The purely autoregressive model from PROC AUTOREG and the mixed ARMA error model can both be estimated in PROC ARIMA. Doing so shows the AIC and SBC criteria to be smaller (better) for the model with the mixed ARMA error. The chi-square white noise tests, although acceptable in both, have higher (better) p -values for the mixed ARMA error structure. There is a lesson here. With its automated error structure identification, PROC AUTOREG can give a final model or serve as a starting point for a more sophisticated approach such as that available in PROC ARIMA.

5.1.3 Unequal Variances

The models previously discussed involve white noise innovations or shocks that are assumed to have constant variance. For long data sets, it can be apparent just from a graph that this constant variance assumption is unreasonable. PROC AUTOREG provides methods for handling such situations. In **Output 5.6**, you see graphs of 8892 daily values from January 1, 1920 (with $Y_1 = 108.76$), to December 31, 1949 (with $Y_{8892} = 200.13$), of Y_t = the Dow Jones Industrial Average, $L_t = \log(Y_t)$ and $D_t = \log(Y_t) - \log(Y_{t-1})$. Clearly, the log transformation improves the statistical properties and gives a clearer idea of the long-term increase than does the untransformed series. Many macroeconomic time series are better understood on the logarithmic scale over long periods of time. By the properties of logarithms, note that $D_t = \log(Y_t / Y_{t-1})$. If Y_t / Y_{t-1} is near 1, then D_t is approximately $(Y_t - Y_{t-1})/Y_{t-1}$. That is, 100 D_t represents the daily percentage change in the Dow Jones average.

To demonstrate how this works, let $\Delta = ((Y_t - Y_{t-1}) / Y_{t-1})$, so $1 + \Delta = Y_t / Y_{t-1}$. Using a Taylor series expansion of $\log(X)$ at $X = 1$, you can represent $\log(1 + \Delta) = \log(1) + 1^{-1}\Delta - (1^{-2}/2)\Delta^2 + \dots$ because $(\partial/\partial X)\log(X) = 1/X$, $(\partial/\partial X)(1/X) = (-1/(X^2))$, and so on. Because $\log(1) = 0$, it follows that $\log(1 + \Delta) = \log(Y_t / Y_{t-1})$ can be approximated by $0 + \Delta = (Y_t - Y_{t-1}) / Y_{t-1}$. This also shows that $\log(Y_t / Y_{t-1})$ is essentially the overnight return on a \$1 investment.

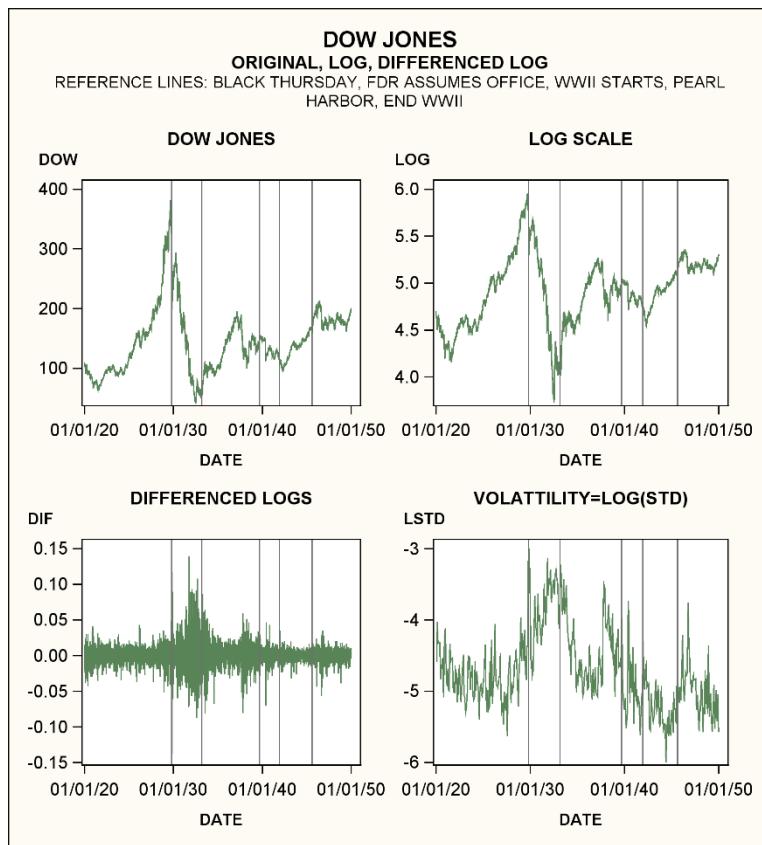
The graph of D_t , in **Output 5.6**, shows some periods of high volatility in these returns. The five vertical graph lines represent, from left to right, Black Thursday (October 24, 1929, when the stock market crashed), the inauguration of President Franklin D. Roosevelt (FDR), the start of World War II, the bombing of Pearl Harbor, and the end of World War II. Note especially the era from Black Thursday until somewhat after FDR assumed office, known as the Great Depression.

The mean of the D_t values is as follows:

$$\bar{D} = n^{-1} \sum_{t=2}^n (\log(Y_t) - \log(Y_{t-1})) = n^{-1} (\log(Y_n) - \log(Y_1))$$

This results in $e^{n\bar{D}} = Y_n/Y_1$, the increase in the series over the entire time period. For the data at hand, the ratio of the last to first data point is $200.13/108.76 = 1.84$, so the series did not quite double during this 30-year period. You might argue that subperiods like the depression, in which extreme volatility is present, are not typical and should be ignored or at least downweighted in computing a rate of return that has some relevance for future periods. You decide to look at the variability of the D_t series.

Because there is so much data, the reduction of each month's D_t numbers to a standard deviation still leaves a relatively long time series of 360 monthly numbers. These standard deviations have a histogram with a long tail to the right. Again, a logarithmic transform is used to produce a monthly series $S_t = \log(\text{standard deviation})$, variable LSTD, that has a more symmetric distribution. Thus, S_t measures the volatility in the series, and a plot of LSTD versus time is the fourth graph in **Output 5.6**.

Output 5.6: Dow Jones Industrial Average on Several Scales

Now, apply a time series model to the S_t series. The tau test for stationarity suggests a unit root process when six augmenting lags are used. The reason for choosing six lags is that the partial autocorrelation function for S_t is near 0 after lag 6. Furthermore, a regression of $S_t - S_{t-1}$ on S_{t-1} and 20 lagged differences ($S_{t-j} - S_{t-j-1}$ for $j = 1$ to 20) in PROC REG gave an insignificant F test for lags 7 through 20. A similar regression using six lagged differences showed all six to be significant according to their t tests. Dickey and Fuller (1979) show that such t tests on lagged differences are valid in large samples—only the test for the coefficient on the lagged level S_{t-1} has a nonstandard distribution. That test cannot reject the unit root hypothesis. This is consistent with a model in first differences for the log-transformed standard deviation series S_t . The results are not displayed. At this point, you are ready to model S_t . You have seen that a lag 6 autoregressive model for $S_t - S_{t-1}$ seems to provide an adequate fit. Perhaps this long autoregression is an approximation of a mixed (ARMA) model. The following code, again using LSTD as the variable name for S_t , seems to provide a reasonable ARMA(1,1) model:

```
proc arima data=out1;
  i var=lstd(1) stationarity=(adf=(6));
  e p=1 q=1 ml noconstant;
run;
```

The constant was suppressed (NOCONSTANT) after an initial check showed it to be insignificant. The tau test for unit roots suggests stationarity of the differenced series ($p = 0.0001$) when six lagged differences are used. That is, no further differencing seems to be needed. Said and Dickey (1984) show that even for mixed models, these stationarity tests are valid as long as sufficient lagged differences are included in the model. In summary, the S series appears to be well modeled as an ARIMA(1,1,1) series with parameters as shown in **Output 5.7**.

Output 5.7: ARIMA Model for S

Augmented Dickey-Fuller Unit Root Tests								
Type	Lags	Rho	Pr < Rho	Tau	Pr < Tau	F	Pr > F	
Zero Mean	6	322.9722	0.9999	-11.07	<.0001			
Single Mean	6	322.2521	0.9999	-11.06	<.0001	61.14	0.0010	
Trend	6	322.0197	0.9999	-11.05	<.0001	61.05	0.0010	

Maximum Likelihood Estimation					
Parameter	Estimate	Standard Error	t Value	Approx Pr > t	Lag
MA1,1	0.82365	0.04413	18.66	<.0001	1
AR1,1	0.32328	0.07338	4.41	<.0001	1

Autocorrelation Check of Residuals									
To Lag	Chi-Square	DF	Pr > ChiSq	Autocorrelations					
6	2.07	4	0.7230	0.010	-0.016	-0.045	-0.053	0.015	0.016
12	7.95	10	0.6341	0.072	0.060	-0.018	0.044	-0.005	0.070
18	19.92	16	0.2236	-0.013	-0.014	-0.134	0.065	0.078	0.055
24	23.96	22	0.3494	0.053	-0.022	-0.080	-0.018	0.015	0.017
30	34.37	28	0.1890	-0.045	0.016	-0.131	-0.081	-0.002	-0.024
36	37.38	34	0.3164	0.014	0.027	0.056	-0.010	-0.057	0.010
42	38.83	40	0.5227	-0.018	0.005	-0.050	-0.016	-0.007	-0.022
48	41.69	46	0.6533	-0.015	-0.010	-0.028	-0.070	-0.004	-0.030

The model suggests the predicting equation $\hat{S}_t = S_{t-1} + 0.3233(S_{t-1} - S_{t-2}) - 0.82365\hat{e}_{t-1}$ where \hat{e}_{t-1} would be replaced by the residual $S_{t-1} - \hat{S}_{t-1}$. Exponentiation of \hat{S}_t gives a conditional standard deviation for month t . Notice that because \hat{S}_t is a logarithm, the resulting standard deviations will all be positive regardless of the sign of \hat{S}_t . This allows the variance to change over time in a way that can be predicted from the most recent few variances. The theory underlying ARIMA models is based on large sample arguments and does not require normality, so the use of log-transformed standard deviations as data does not necessarily invalidate this approach. However, there are at least two major problems with approaching heterogeneous variation in the manner previously used with the Dow Jones series. First, you will not often have so much data to start with. Second, the use of a month as a period for computing a standard deviation is quite arbitrary. A more statistically rigorous approach is presented next. The discussion thus far has been a review of unit root test methodology as well as a motivation for fitting a nonconstant variance model that might involve a unit root. An analyst would most likely use the more sophisticated approach shown in the next section.

5.1.4 ARCH, GARCH, and IGARCH for Unequal Variances

The series D_t , whose variability is measured by S_t , has nonconstant conditional variance. Engle (1982) introduced a model in which the variance at time t is modeled as a linear combination of past squared residuals and called it an ARCH (autoregressive conditionally heteroscedastic) process. Bolerslev (1986) introduced a more general structure in which the variance model looks more like an ARMA model than an AR model. He called it a GARCH (generalized ARCH) process. Thus, the usual approach to modeling ARCH or GARCH processes improves the method previously shown in substantial ways. The purpose of the monthly standard deviation approach is to illustrate the idea of an ARMA type of structure for standard deviations or variances. Robert F. Engle (for ARCH) and Clive W. J. Granger (for cointegration, discussed in section 5.2) jointly won the Nobel Prize in Economics in 2003.

The usual approach to GARCH(p,q) models is to model an error term ε_t in terms of a standard white noise $e_t \sim N(0,1)$, as $\varepsilon_t = \sqrt{h_t}e_t$ where h_t satisfies the type of recursion used in an ARMA model:

$$h_t = \omega + \sum_{i=1}^q \alpha_i \varepsilon_{t-i}^2 + \sum_{j=1}^p \gamma_j h_{t-j}$$

In this way, the error term has a conditional variance that is a function of the magnitudes of past errors. Engle's original ARCH structure has $\gamma_j = 0$. Because h_t is the variance rather than its logarithm, certain restrictions must be placed on the α_i , γ_j , and ω to ensure positive variances. For example, if these are all restricted to be positive, then positive initial values of h_t will ensure that all h_t are positive. For this reason, Nelson (1991) suggested replacing h_t with $\log(h_t)$ and an additional modification. He called the resulting process EGARCH. These approaches allow the standard deviation to change with each observation. Nelson and Cao (1992) give constraints on the α and γ values that ensure nonnegative

estimates of h_t . These are the default in PROC AUTOREG. More details are provided in PROC AUTOREG documentation and in Hamilton (1994), which is a detailed reference for time series.

Recall that PROC AUTOREG will fit a regression model with autoregressive errors using the ML method based on a normal distribution. In place of the white noise shocks in the autoregressive error model, you can specify a GARCH(p,q) process. If it appears, as suggested by your analysis of the Dow Jones standard deviations, that the process describing the error variances is a unit root process, then the resulting model is referred to as an integrated GARCH or IGARCH. If the usual stationarity conditions are satisfied, then for a GARCH process, forecasts of h_t will revert to a long-run mean. In an IGARCH model, mean reversion is no longer a property of h_t , so forecasts of h_t will tend to reflect the most recent variation rather than the average historical variation. You would expect the variation during the Great Depression to have little effect on future h_t values in an IGARCH model of the Dow Jones data.

To investigate models of the daily percentage change in the Dow Jones Industrial Average Y_t , you use $D_t = \log(Y_t) - \log(Y_{t-1})$. Calling this variable DDOW, you issue this code:

```
proc autoreg data=more;
  model ddow = / nlag=2
    garch=(p=2, q=1, type=integ, noint);
  output out=out2 ht=ht p=f lcli=l ucli=u;
run;
```

PROC AUTOREG allows the use of regression inputs. However, there is no apparent time trend or seasonality, and no other regressors are readily available. The model statement DDOW = (with no inputs) specifies that the regression part of your model is only a mean. Note how the h_t sequence, predicted values, and default upper and lower forecast limits have been requested in the data set called OUT2.

In **Output 5.8**, the estimate of the mean is 0.000363. Because DDOW is a difference, a mean is interpreted as a drift in the data. Because the data are log differences, the number $e^{0.000363} = 1.0003631$ is an estimate of the long-run daily growth over this time period. With 8892 days in the study period, the number $e^{(0.000363)(8891)} = 25$ represents a 25-fold increase, approximately an 11.3% yearly growth rate! This is not remotely like the rate of growth, except in certain portions of the graph. PROC AUTOREG starts with OLS estimates so that the average DDOW over the period is the OLS intercept 0.0000702 from **Output 5.8**. This gives $e^{(0.0000702)(8891)} = 1.87$, indicating 87% growth for the full 30-year period. This has to be more in line with the graph because (as you saw earlier) except for rounding error, it is Y_n / Y_1 .

Note the strong rejection of normality. The normality test used here is the one developed by Jarque and Bera (1980). This is a general test of normality based on a measurement of skewness b_1 and kurtosis $b_2 - 3$ using residuals r_t , where

$$b_1 = \frac{\sum_{t=1}^n r_t^3 / n}{\left(\sum_{t=1}^n r_t^2 / n \right)^{3/2}}$$

and

$$b_2 - 3 = \frac{\sum_{t=1}^n r_t^4 / n}{\left(\sum_{t=1}^n r_t^2 / n \right)^2} - 3$$

The expression $\sum_{t=1}^n r_t^j / n$ is sometimes called the (raw) j th moment of r . The fractions involve third and fourth moments scaled by the sample variance. The numerators are sums of approximately independent terms. Thus, they satisfy a central limit theorem. Both have approximately mean 0 when the true errors are normally distributed. Approximate variances of the skewness and kurtosis are $6 / n$ and $24 / n$. Odd and even powers of normal errors are uncorrelated, so squaring each of these approximately normal variates and dividing by its variance produces a pair of squares of approximately independent $N(0,1)$ variates. Therefore, the sum of these squared variates follows a chi-square distribution with two degrees of freedom under the normality null hypothesis. The Jarque-Bera test

$$JB = n \left(b_1^2 / 6 + (b_2 - 3)^2 / 24 \right)$$

has (approximately) a chi-square distribution with two degrees of freedom under the null hypothesis of normal errors.

Why is the IGARCH model giving a 25-fold increase? It seems unreasonable. The model and the data indicate large variability during periods when there were steep drops in the Dow Jones average. A method that accounts for different variances tends to downweight observations with high variability. In fact, there are some periods in which the 11.3% annual rate required for a 25-fold increase ($1.113^{20} = 25$) was actually exceeded, such as in the periods leading up to the Great Depression, after FDR assumed office, and toward the end of WWII. The extremely large variances associated with periods of decrease or slow growth give them low weight. That would tend to increase the estimated growth rate, but it is still not enough to explain the results.

Output 5.8: IGARCH Model for Dow Jones

The AUTOREG Procedure

Dependent Variable	DDOW
--------------------	------

The AUTOREG Procedure

Ordinary Least Squares Estimates			
SSE	1.542981	DFE	8891
MSE	0.0001735	Root MSE	0.01317
SBC	-51754.031	AIC	-51761.124
MAE	0.00831381	AICC	-51761.123
MAPE	99.8614695	HQC	-51758.709
Durbin-Watson	1.9427	Regress R-Square	0.0000
		Total R-Square	0.0000

Parameter Estimates					
Variable	DF	Estimate	Standard Error	t Value	Approx Pr > t
Intercept	1	0.0000702	0.000140	0.50	0.6155

Estimates of Autoregressive Parameters			
Lag	Coefficient	Standard Error	t Value
1	-0.029621	0.010599	-2.79
2	0.037124	0.010599	3.50

The AUTOREG Procedure

Integrated GARCH Estimates			
SSE	1.5453787	Observations	8892
MSE	0.0001738	Uncond Var	.
Log Likelihood	28466.2335	Total R-Square	.
SBC	-56887.003	AIC	-56922.467
MAE	0.00831069	AICC	-56922.46
MAPE	114.79916	HQC	-56910.392
		Normality Test	3886.0451
		Pr > ChiSq	<.0001

Parameter Estimates					
Variable	DF	Estimate	Standard Error	t Value	Approx Pr > t
Intercept	1	0.000363	0.0000748	4.85	<.0001
AR1	1	-0.0868	0.009734	-8.91	<.0001
AR2	1	0.0323	0.009579	3.37	0.0008
ARCH1	1	0.0698	0.003965	17.60	<.0001
GARCH1	1	0.7078	0.0609	11.62	<.0001
GARCH2	1	0.2224	0.0573	3.88	0.0001

Perhaps more importantly, the rejection of normality by the Jarque-Bera test introduces the possibility of bias in the estimated mean. In an OLS regression of a column Y of responses in a matrix \mathbf{X} of explanatory variables, the model is $\mathbf{Y} = \mathbf{X}\beta + \mathbf{e}$. The estimated parameter vector $\hat{\beta} = (\mathbf{X}'\mathbf{X})^{-1}(\mathbf{X}'\mathbf{Y}) = \beta + (\mathbf{X}'\mathbf{X})^{-1}(\mathbf{X}'\mathbf{e})$ is unbiased whenever the random vector \mathbf{e} has mean 0. In regard to bias, it does not matter whether the variances are unequal or whether there is correlation among the errors. These features affect only the variance of the estimates, causing biases in the standard errors for $\hat{\beta}$, but not in the estimates of β , themselves. In contrast to OLS, GARCH and IGARCH models are fit by maximum likelihood assuming a normal distribution. Failure to meet this assumption could produce bias in parameter estimates such as the estimated mean.

As a check to see whether bias can be induced by nonnormal errors, data from a model having the same h_t sequence as estimated for the Dow Jones log differences data were generated for innovations $e_t \sim N(0,1)$, and again for innovations $(e_t^2 - 1)/\sqrt{2}$. This second set of innovations used the same normal variables in a way that gave a skewed distribution, yet still having mean 0 and variance 1. The mean was set at 0.00007 for the simulation, and 50 such data sets were created. For each data set, IGARCH models were fit for each of the two generated series, and the estimated means were output to a data set. The overall mean and standard deviation of each set of 50 means were as follows:

Kind of Error	Mean	Standard Deviation
Normal	0.000071	0.000907
Skewed	0.000358	0.001496

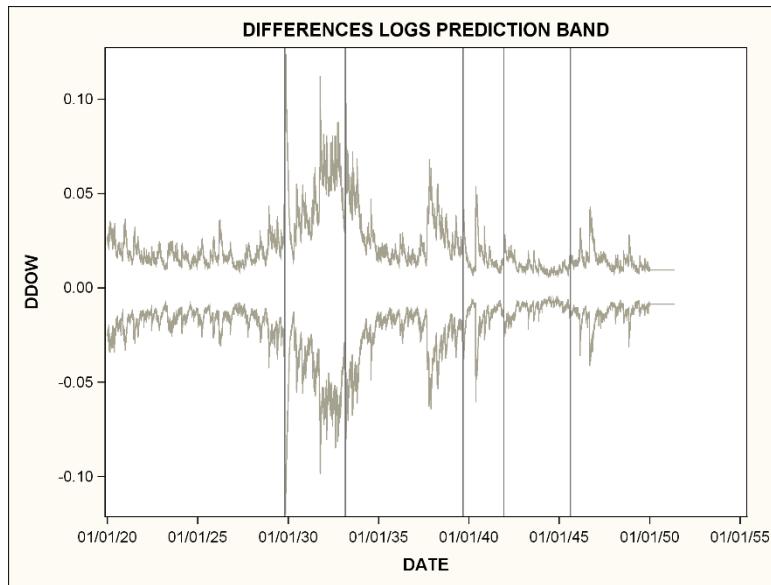
Thus, it seems that finding a factor of 5 bias in the estimate of the mean (of the log differences) could be simply a result of error skewness. In fact, the factor of 5 is almost exactly what the simulation shows. The simulation results show that if the errors had been normal, good estimates of the true value, known in the simulation to be 0.00007, would have resulted.

Using TYPE=INTEG specifies an IGARCH model for h_t , which, like any unit root model, will have a linearly changing forecast if an intercept is present. Therefore, you use NOINT to suppress the intercept. Using $p = 2$ and $q = 1$, your h_t model has the following form:

$$h_t = h_{t-1} + 0.7078(h_{t-1} - h_{t-2}) + 0.2224(h_{t-2} - h_{t-3}) + 0.0698\varepsilon_{t-1}^2$$

You can look at h_t as a smoothed local estimate of the variance. It was computed by adding to the previous smoothed value (h_{t-1}) a weighted average of the two most recent changes in these smoothed values and the square of the most recent shock.

PROC AUTOREG produces h_t estimates. The resulting 95% prediction limits are shown in **Output 5.9**. The data set MORE used for **Output 5.8** and **Output 5.9** has the historical data and 500 additional days with dates, but no additional values of D_t . PROC AUTOREG produces h_t values and prediction limits for these. In general, future values of all inputs need to be included for this to work, but here the only input is the intercept.

Output 5.9: h_t -Based Intervals

It appears that $(h_t - h_{t-1})$, $(h_{t-1} - h_{t-2})$, and ε_t^2 were fairly small at the end of the series, contributing very little to h_{t+1} so that h_{n+1} is approximately h_n , as are all h_{n+j} for $j > 0$. The forecast intervals coming off the end of the series have about the same width as the last forecast interval in the historical data. They are almost, but not exactly, two horizontal lines.

The autoregressive error model is $Z_t = 0.0868Z_{t-1} - 0.0323Z_{t-2} + \varepsilon_t^2$ where $-\varepsilon_t = \sqrt{h_t}e_t$.

Although the lag Z coefficients are statistically significant, they are small, so their contribution to forecasts and to the width of prediction intervals into the future is imperceptible in the graph.

Clearly, the IGARCH estimated mean 0.000363 is unacceptable because of the nonnormality, the resulting danger of bias, and its failure to represent the observed growth over the period. The ordinary mean 0.00007 is an unbiased estimate and exactly reproduces the observed growth. The usual conditions leading to the OLS formula for the standard error of a mean do not hold here. (More will be said about this shortly.) The problem is not with IGARCH versus GARCH. In fact, a GARCH(2,1) model fits the series nicely, but still gives an unacceptable estimate of the mean of D_t . The average of n independent values of ε_t has variance

$$n^{-2} \sum_{t=1}^n h_t$$

if e_t has mean 0 and variance 1. The AR(2) error series $Z_t = \alpha_1 Z_{t-1} + \alpha_2 Z_{t-2} + \varepsilon_t$ can be summed from 1 to n on both sides and divided by n to get $(1 - \alpha_1 - \alpha_2)\bar{Z}$, approximately equal to $\bar{\varepsilon}$. From $\varepsilon_t = \sqrt{h_t}e_t$, it follows that the (approximate) variance of $(1 - \alpha_1 - \alpha_2)\bar{Z}$ is $n^{-2} \sum_{t=1}^n h_t$ and Z is $(1 - \alpha_1 - \alpha_2)^{-2} n^{-2} \sum_{t=1}^n h_t$. Hamilton (1994, p. 663) indicates that maximum likelihood estimates of h_t are reasonable under rather mild assumptions for ARCH models even when the errors are not normal. Also, the graphical evidence indicates that the estimated h_t series has captured the variability in the data nicely. Proceeding on that basis, you sum the estimated h_t series and use estimated autoregressive coefficients to estimate the standard deviation of the mean:

$$n^{-1} \sqrt{(1 - \alpha_1 - \alpha_2)^2 \sum_{t=1}^n h_t}$$

as

$$8892^{-1} \sqrt{(1 - 0.0868 + 0.0323)^2 1.55846} = 0.0001485$$

In this way, you get the following:

$$t = \frac{0.00007}{0.0001845} = 0.5$$

This is not significant at any reasonable level.

Interestingly (and despite the previous comments), a simple t test on the D_t data, ignoring all of the variance structure, gives about the same t . A little thought shows that this could be anticipated for the special case of this model. The summing of h_t and division by n yield what might be thought of as an average variance over the period. Because the α s are small, the average of h_t divided by n is a reasonable approximation of the variance of \bar{Z} and \bar{D} . To the extent that the squared residuals $(D_t - \bar{D})^2$ provide approximate estimates of the corresponding conditional variances h_t , the usual OLS formula gives an estimate of the standard error of the mean:

$$\sqrt{n^{-1} \sum_{t=1}^n (D_t - \bar{D})^2 / (n-1)}$$

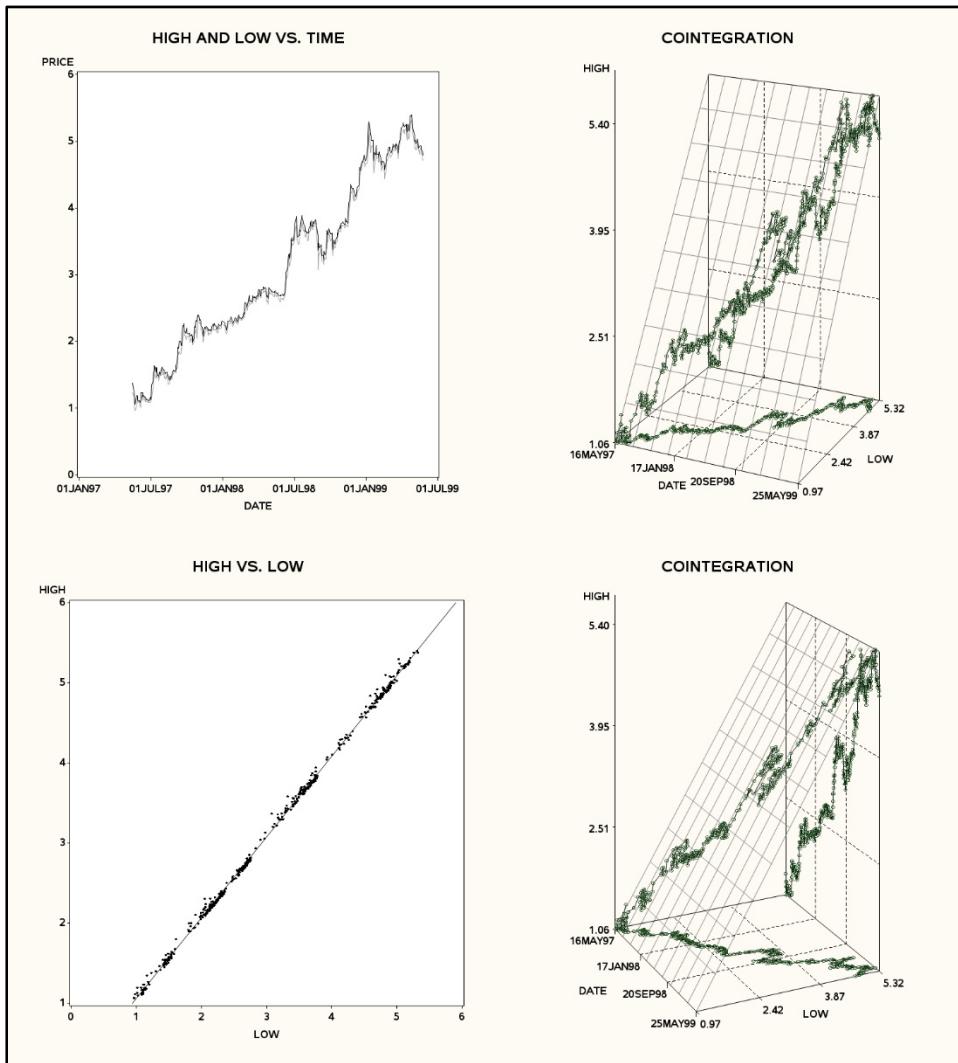
Additional care is required, such as consideration of the assumed unit root structure for h_t and the error introduced by ignoring the α s, to make this into a rigorous argument. However, this line of reasoning does suggest that the naive t test, produced by PROC MEANS, for example, might be reasonable for these particular data. There is no reason to expect the naive approach to work well in general.

This example serves to illustrate several important points. One is that careful checking of model implications against what happens in the data is a crucial component of proper analysis. This typically involves some graphics. Another is that failure to meet assumptions is sometimes not so important but, at other times, can render estimates meaningless. Careful thinking and a knowledge of statistical principles are crucial. The naive use of statistical methods without understanding the underlying assumptions and limitations can lead to ridiculous claims. Computational software is not a replacement for knowledge.

5.2 Cointegration

As previously mentioned, the 2003 Nobel Prize in Economics was given for ARCH models and cointegration. In this section, cointegration is discussed and implemented using the VARMAX procedure.

In this section, you study a dimension k vector \mathbf{V}_t of time series. The model $\mathbf{V}_t = \mathbf{A}_1 \mathbf{V}_{t-1} + \mathbf{A}_2 \mathbf{V}_{t-2} + \mathbf{e}_t$ is called a vector autoregression or a VAR of dimension k and order $p = 2$ (2 lags). It is assumed that \mathbf{e}_t has a multivariate normal distribution with k dimensional mean vector $\mathbf{0}$, a vector of 0s, and $k \times k$ variance matrix Σ . The i th element of \mathbf{V}_t is the time series $Y_{it} - \mu_i$, so the deviation of each series from its mean is expressed by the model as a linear function of previous deviations of all series from their means. For example, the upper-left panel of **Output 5.10** shows the logarithms of some high and low prices for stock of the electronic retailer Amazon.com, extracted by the Internet search engine Yahoo!

Output 5.10: Amazon.com Data with Cointegrating Plane

One way of fitting a vector model is to simply regress each Y_{it} on lags of all Y s including its own lags to produce estimates of row i of the \mathbf{A} coefficient matrices. Using just one lag, you specify as follows:

```
proc reg data=amazon;
  model high low = high1 low1;
run;
```

In this code, high1 and low1 are lagged values of the log-transformed high and low prices. The partial output **Output 5.11** shows the estimates.

Output 5.11: PROC REG on Amazon.com Data

The REG Procedure

Model: MODEL1

Dependent Variable: HIGH

Parameter Estimates					
Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > t
Intercept	1	0.01573	0.00730	2.15	0.0317
HIGH1	1	0.88411	0.05922	14.93	<.0001
LOW1	1	0.11583	0.05979	1.94	0.0533

The REG Procedure
Model: MODEL1
Dependent Variable: LOW

Parameter Estimates					
Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > t
Intercept	1	-0.01042	0.00739	-1.41	0.1590
HIGH1	1	0.45231	0.05990	7.55	<.0001
LOW1	1	0.54209	0.06047	8.96	<.0001

The estimated model becomes the following:

$$\begin{pmatrix} Y_{1t} - \mu_1 \\ Y_{2t} - \mu_2 \end{pmatrix} = \begin{pmatrix} 0.8841 & 0.1158 \\ 0.4523 & 0.5421 \end{pmatrix} \begin{pmatrix} Y_{1,t-1} - \mu_1 \\ Y_{2,t-1} - \mu_2 \end{pmatrix} + \begin{pmatrix} e_{1t} \\ e_{2t} \end{pmatrix}$$

Recall that in a univariate AR(1) process, $Y_t = \alpha Y_{t-1} + e_t$, the requirement $|\alpha| < 1$ was imposed so that the expression $Y_t = \sum_{j=0}^{\infty} \alpha^j e_{t-j}$ for Y_t in terms of past shocks e_t would converge. That is, it would have weights on past shocks that decay exponentially as you move further into the past. What is the analogous requirement for the vector process $\mathbf{V}_t = \mathbf{AV}_{t-1} + \mathbf{e}_t$? The answer lies in the eigenvalues of the coefficient matrix \mathbf{A} .

5.2.1 Cointegration and Eigenvalues

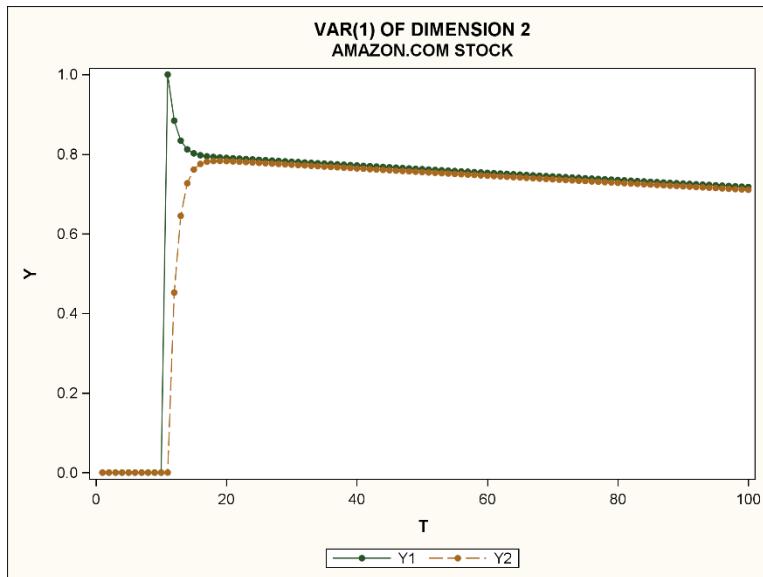
Any $k \times k$ matrix has k possibly complex numbers called eigenvalues or roots that determine certain properties of the matrix. The eigenvalues of \mathbf{A} are defined to be the roots of the polynomial $|m\mathbf{I} - \mathbf{A}|$, where \mathbf{A} is the $k \times k$ coefficient matrix, \mathbf{I} is a $k \times k$ identity, and $||$ denotes a determinant. For the previous fitted 2×2 matrix, you find the following:

$$\left| m \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} - \begin{pmatrix} 0.8841 & 0.1158 \\ 0.4523 & 0.5421 \end{pmatrix} \right| = (m - 0.8841)(m - 0.5421) - (0.4523)(0.1158)$$

This becomes $m^2 - (0.8841 + 0.5421)m + 0.4269 = (m - 0.9988)(m - 0.4274)$. The roots of this matrix are the real numbers 0.9988 and 0.42740. A matrix with unique eigenvalues can be expressed as $\mathbf{A} = \mathbf{ZDZ}^{-1}$, where \mathbf{D} is a matrix with the eigenvalues of \mathbf{A} on the main diagonal and 0 everywhere else and \mathbf{Z} is the matrix of eigenvectors of \mathbf{A} . Note that $\mathbf{A}^L = (\mathbf{ZDZ}^{-1})(\mathbf{ZDZ}^{-1}) \cdots (\mathbf{ZDZ}^{-1}) = \mathbf{ZD}^L \mathbf{Z}^{-1}$. By the same reasoning as in the univariate case, the predicted deviations from the means L steps into the future are $\hat{\mathbf{V}}_{n+L} = \mathbf{A}^L \mathbf{V}_n$, where \mathbf{V}_n is the last observed vector of deviations. For the 2×2 matrix, you have the following:

$$\begin{aligned} \mathbf{A}^L &= \begin{pmatrix} 0.8841 & 0.1158 \\ 0.4523 & 0.5421 \end{pmatrix}^L = \mathbf{ZD}^L \mathbf{Z}^{-1} \\ &= \mathbf{Z} \begin{pmatrix} 0.9988 & 0 \\ 0 & 0.4274 \end{pmatrix}^L \mathbf{Z}^{-1} = \mathbf{Z} \begin{pmatrix} 0.9988^L & 0 \\ 0 & 0.4274^L \end{pmatrix} \mathbf{Z}^{-1} \end{aligned}$$

The elements of \mathbf{A}^L all converge to 0 as seen in **Output 5.12**.

Output 5.12: Impulse Response, Lag 1 Model**5.2.2 Impulse Response Function**

To illustrate, **Output 5.12** shows a bivariate series with both Y_{1t} and Y_{2t} being 0 up to time $t = 11$, mimicking constant high and low stock price (log transformed and mean corrected). At time $t = 11$, Y_{1t} is shifted to 1 with Y_{2t} remaining at 0, representing a shock to the high price—that is, $\mathbf{V}_{11} = (1, 0)'$. From then on, $\hat{\mathbf{V}}_{11+L} = \mathbf{A}^L \mathbf{V}_{11} = \mathbf{A}^L (1, 0)'$. In other words, $\hat{\mathbf{V}}_{11+L}$ traces out the path that would be followed with increasing lead L in absence of further shocks. The sequence computed is called an impulse response function. It is seen that at time $t = 12$, $Y_{2,t}$ responded to the jump in $Y_{1,t}$ and increased to about 0.45, while $Y_{1,t}$ decreased following the initial jump to about 0.88. Continuing through time, the two series come close together, then descend very slowly toward 0. This demonstrates the effect of a unit shock to the log of high price. The equilibrium, 0 deviations of both series from their mean, is approached slowly due to the eigenvalue 0.9988 being so close to 1. Clearly, if it were exactly 1.000, then 1.000^L would not decrease at all, and the forecasts would not converge to the mean (0). Similarly, any attempt to represent the vector of deviations from the mean in terms of an infinite weighted sum of past error vectors will fail (for example, not converge) if the eigenvalues or roots of the coefficient matrix \mathbf{A} are one—that is, if \mathbf{A} has any unit roots.

When all the eigenvalues of \mathbf{A} are less than 1, the vector autoregressive process of order 1, or VAR(1), is stationary, following the terminology from univariate processes. When the true \mathbf{A} has unit roots, nonstandard distributions of estimates will arise just as in the univariate case. The largest eigenvalue of the estimated matrix, $\hat{\rho} = 0.9988$, is uncomfortably close to 1. It would not be at all surprising to find that the true \mathbf{A} matrix has a unit root. The roots are analogous to the reciprocals of the roots you found for univariate series. Therefore, there is the requirement that these roots be less than 1, not greater than 1 (in magnitude).

5.2.3 Roots in Higher-Order Models

The requirement that the roots be less than 1 in magnitude is called the *stationarity condition*. Series satisfying this requirement are said to be stationary, although technically, certain conditions on the initial observations are required to ensure constant mean and covariances that depend only on the time separation of observations (which is the mathematical definition of stationarity).

In higher-order vector processes, it is still the roots of a determinantal equation that determine stationarity. In an order 2 VAR, $\mathbf{V}_t = \mathbf{A}_1 \mathbf{V}_{t-1} + \mathbf{A}_2 \mathbf{V}_{t-2} + \mathbf{e}_t$, the characteristic polynomial is $|m^2 \mathbf{I} - \mathbf{A}_1 m - \mathbf{A}_2|$. If all values of m that make this determinant 0 satisfy $|m| < 1$, then the vector process satisfies the stationarity condition. Regressing on lag 1 and 2 terms in the Amazon.com high and low price series, this estimated model is found:

$$\hat{\mathbf{V}}_t = \begin{pmatrix} 0.98486 & 0.23545 \\ 0.63107 & 0.65258 \end{pmatrix} \mathbf{V}_{t-1} + \begin{pmatrix} -0.08173 & -0.13927 \\ -0.22514 & -0.06414 \end{pmatrix} \mathbf{V}_{t-2}$$

The matrix entries are estimates coming from two PROC REG outputs as seen in **Output 5.13**.

Output 5.13: Process of Order 2

The REG Procedure
Model: MODEL1
Dependent Variable: HIGH

Parameter Estimates					
Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > t
Intercept	1	0.01487	0.00733	2.03	0.0430
HIGH1	1	0.98486	0.06896	14.28	<.0001
LOW1	1	0.23545	0.06814	3.46	0.0006
HIGH2	1	-0.08173	0.07138	-1.14	0.2528
LOW2	1	-0.13927	0.06614	-2.11	0.0357

The REG Procedure
Model: MODEL1
Dependent Variable: LOW

Parameter Estimates					
Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > t
Intercept	1	-0.00862	0.00731	-1.18	0.2386
HIGH1	1	0.63107	0.06877	9.18	<.0001
LOW1	1	0.65258	0.06795	9.60	<.0001
HIGH2	1	-0.22514	0.07118	-3.16	0.0017
LOW2	1	-0.06414	0.06595	-0.97	0.3312

Inclusion of lag 3 terms seems to improve the model even further. For simplicity of exposition, the lag 2 model is discussed. Keeping all the coefficient estimates, the characteristic equation, whose roots determine stationarity, is written as follows:

$$\begin{aligned} & \left| m^2 \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} - m \begin{pmatrix} 0.98486 & 0.23545 \\ 0.63107 & 0.65258 \end{pmatrix} - \begin{pmatrix} -0.08173 & -0.13927 \\ -0.22514 & -0.06414 \end{pmatrix} \right| \\ & = (m - 0.99787)(m - 0.54974)(m - 0.26767)(m + 0.17784) \end{aligned}$$

Again, the largest eigenvalue, $\hat{\rho} = 0.99787$, is very close to 1. It would not be at all surprising to find that the characteristic equation $|m^2\mathbf{I} - m\mathbf{A}_1 - \mathbf{A}_2| = 0$ using the true coefficient matrices has a unit root. Fountis and Dickey (1989) show that if a vector AR process has a single unit root, then the largest estimated root, normalized as $n(\hat{\rho} - 1)$, has the same limit distribution as the univariate AR(1) case. Comparing $n(\hat{\rho} - 1) = 509(0.99787 - 1) = -1.08$ to the 5% critical value (-11.3), the unit root hypothesis is not rejected. This provides a test for 1 versus no unit roots. Hence, it is not as general as tests to be discussed later. Also, no diagnostics have been performed to check the model adequacy, a prerequisite for validity of any statistical test.

Using this vector AR(2) model, a bivariate vector of 0 deviations up to time $t = 11$ is generated. Then, a unit shock is imposed on the first component, the one corresponding to the high price, and on the AR(2) used to extrapolate into the future. The code is:

```
data shock;
y12=0; y22=0; y11=0; y21=0;
do t=1 to 100;
  y1 = 0.98486*y11 + 0.23545*y21 - 0.08173*y12 - 0.13927*y22;
  y2 = 0.63107*y11 + 0.65258*y21 - 0.22514*y12 - 0.06414*y22;
  if t=11 then y1=1;
  output;
  y22=y21; y21=y2; y12=y11; y11=y1;
end;
```

```

run;

proc gplot data=shock; plot (y1 y2)*t/overlay href=11;
  symbol1 v=dot i=join c=red;
  symbol2 v=dot i=join c=green;
run;
quit;

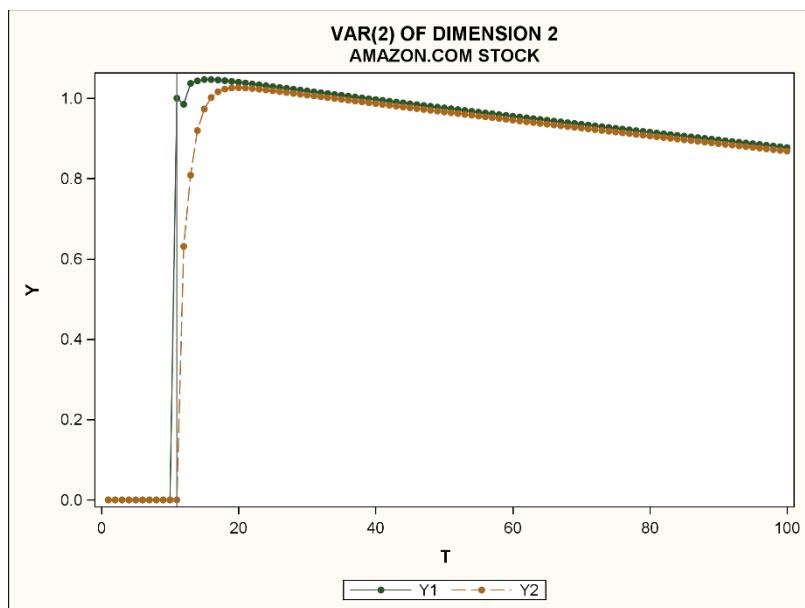
```

The graph of this impulse response function is shown in **Output 5.14**.

The addition of the second lag produces a more interesting pattern immediately following the shock to the high price logarithm series. But, in the long run, the series again approach each other and descend in tandem to the (0,0) equilibrium deviation from the mean.

The forecasts might not have returned to the (0,0) equilibrium point if the true coefficient matrices (rather than estimates) had been used. The behavior in the estimated model could simply be the result of the highest estimated root 0.99787 being a slight underestimate of a root that is really 1. A number even slightly smaller than 1 will reduce to nearly 0 when raised to a large exponent, as happens when the impulse response is extrapolated into the future. Models that allow exact unit roots in vector processes are discussed next.

Output 5.14: Impulse Response, Lag 2 Model



5.2.4 Cointegration and Unit Roots

An interesting class of models with exact unit roots is the class of cointegrated vector processes that can be represented in a type of model called the *error correction model*. Cointegration refers to a case in which a vector process, such as the one with logarithms of high and low prices, has individually nonstationary components, but there is some linear combination of them that is stationary. To make things a little clearer, suppose it is hypothesized that the ratio of high to low prices is stable. Specifically, the daily price ratio series $\log(\text{high}/\text{low}) = \log(\text{high}) - \log(\text{low})$ is stationary even though the $\log(\text{high})$ and $\log(\text{low})$ series each have unit roots. In this case, a shock to the high price series results in an impulse response in which both series move as before, but they will not move back toward any historical mean values. Rather, they will move toward some equilibrium pair of values for which $\log(\text{high}) - \log(\text{low})$ equals its long-term mean.

You can check $\text{spread} = \log(\text{high}) - \log(\text{low})$ for stationarity with no new tools. Simply create the daily spread series and perform a unit root test on it. Here is some code to do the test and to check whether 3 autoregressive lags (and 2 lagged differences) are sufficient to reduce the errors to white noise:

```

proc arima data=amazon;
  i var=spread stationarity = (adf=(2));
  e p=3;
run;

```

As shown in **Output 5.15**, the tests strongly reject the unit root null hypothesis and indicate stationarity. The zero mean test would be useful if you are willing to assume a zero mean for $\log(\text{high}) - \log(\text{low})$. Because $\text{high} > \text{low}$ always, such an assumption is untenable for these data. Chi-square tests for a lag 3 autoregression are shown. They indicate that lagged differences beyond the second, $Y_{t-2} - Y_{t-3}$, are unnecessary and that the fit appears to be excellent. This suggests that an increase in the bivariate system to 3 lags might be helpful, as previously mentioned.

It appears that spread = $\log(\text{high}) - \log(\text{low})$ is stationary according to the unit root tests. That means that the standard distribution theory should provide accurate tests because the sample size $n = 502$ is not too small. In that regard, notice that the mean estimate 0.07652 for spread is significantly different from 0. An estimate of the number toward which the ratio of high to low prices tends to return is $e^{0.07652} = 1.08$ with a 95% confidence interval extending from $e^{0.07652-(1.96)(0.004387)} = 1.07$ to $e^{0.07652+(1.96)(0.004387)} = 1.09$. You conclude that the high tends to be 7% to 9% higher than the low in the long run.

Output 5.15: Stationary Test for High-Low Spread

Augmented Dickey-Fuller Unit Root Tests							
Type	Lags	Rho	Pr < Rho	Tau	Pr < Tau	F	Pr > F
Zero Mean	2	-18.3544	0.0026	-3.00	0.0028		
Single Mean	2	-133.290	0.0001	-7.65	<.0001	29.24	0.0010
Trend	2	-149.588	0.0001	-8.05	<.0001	32.41	0.0010

Conditional Least Squares Estimation					
Parameter	Estimate	Standard Error	t Value	Approx Pr > t	Lag
MU	0.07652	0.0043870	17.44	<.0001	0
AR1,1	0.38917	0.04370	8.91	<.0001	1
AR1,2	0.04592	0.04702	0.98	0.3293	2
AR1,3	0.18888	0.04378	4.31	<.0001	3

Autocorrelation Check of Residuals									
To Lag	Chi-Square	DF	Pr > ChiSq	Autocorrelations					
6	4.63	3	0.2013	-0.001	-0.018	-0.009	-0.047	0.072	-0.033
12	9.43	9	0.3988	0.037	0.041	0.035	0.025	0.029	0.058
18	12.71	15	0.6248	-0.018	-0.046	0.025	-0.014	0.016	0.052
24	21.14	21	0.4506	0.017	0.023	-0.067	-0.074	0.059	0.038
30	25.13	27	0.5669	0.026	0.049	0.014	-0.012	-0.006	0.063
36	28.86	33	0.6734	0.013	0.038	0.049	-0.023	0.016	-0.045
42	33.05	39	0.7372	0.049	0.055	0.023	0.010	0.039	0.003
48	36.51	45	0.8125	0.030	-0.035	-0.050	-0.038	0.006	-0.004

Testing for cointegration is easy if you can prespecify the linear combination (for example, $S_t = \text{spread} = \log(\text{high}) - \log(\text{low})$). Often, it is suspected that *some* linear combination $Y_{1t} - \beta Y_{2t}$ is stationary, where (Y_{1t}, Y_{2t}) is a bivariate time series. The problem involves estimating β as well as testing the resulting linear combination for stationarity. Engle and Granger (1987) argue that if you use regression to estimate β , your method is somewhat like sorting through all linear combinations of $\log(\text{high})$ and $\log(\text{low})$ to find the most stationary-looking linear combination. Therefore, if you use the standard critical values for this test as if you knew β from some external source, your nominal level 0.05 would underestimate the true probability of falsely rejecting the unit root null hypothesis. Engle and Granger's solution was to compute residuals $r_t = Y_{1t} - \hat{\beta}Y_{2t}$ from a least squares regression of Y_{1t} on Y_{2t} and run a unit root test on these residuals. Then, compare the test statistic to special critical values that they supplied. This is a relatively easy and intuitively pleasing approach. However, it is not clear which of two or more series to use as the dependent variable in such a regression.

More symmetric approaches were suggested by Stock and Watson (1988) and Johansen (1988, 1991). Stock and Watson base their approach on a principal components decomposition of the vector time series. Johansen's method involves

calculating standard quantities and canonical correlations from a multivariate multiple regression and determining what distributions these would have in the vector time series case with multiple unit roots. Both strategies allow testing for multiple unit roots. For further comparisons among these approaches and an application to a macroeconomic vector series, see Dickey, Janssen, and Thornton (1991).

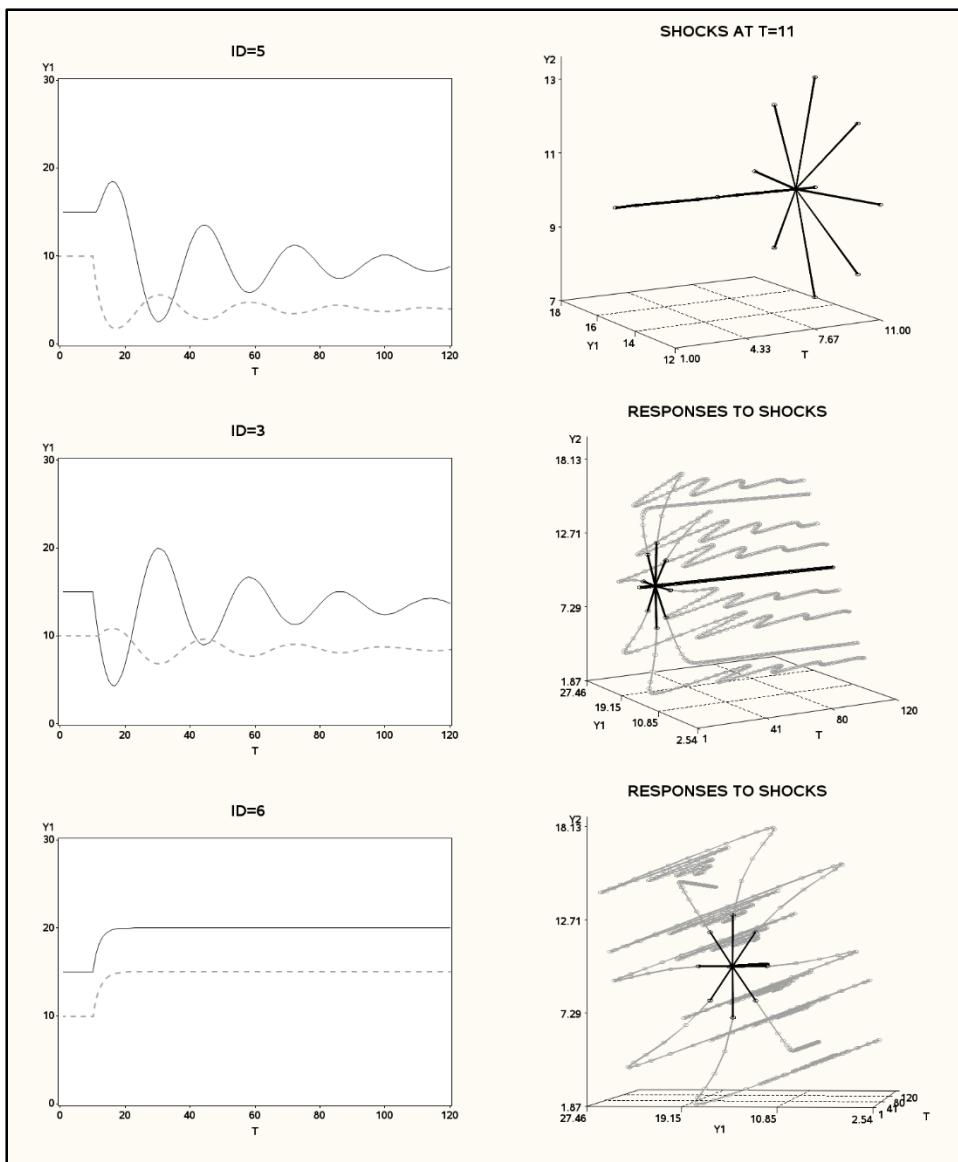
5.2.5 An Illustrative Example

To get a better feeling for cointegration, consider this system with known coefficients:

$$\begin{pmatrix} Y_{1,t} - 15 \\ Y_{2,t} - 10 \end{pmatrix} = \begin{pmatrix} 1.84 & -0.24 \\ -0.06 & 1.66 \end{pmatrix} \begin{pmatrix} Y_{1,t-1} - 15 \\ Y_{2,t-1} - 10 \end{pmatrix} + \begin{pmatrix} -0.88 & 0.28 \\ 0.07 & -0.67 \end{pmatrix} \begin{pmatrix} Y_{1,t-2} - 15 \\ Y_{2,t-2} - 10 \end{pmatrix} + \begin{pmatrix} e_{1,t} \\ e_{2,t} \end{pmatrix}$$

Suppose $Y_{1,t} = 15$ and $Y_{2,t} = 10$ up to time 11, where a shock takes place. What happens after time 11 if no further shocks occur? That is, what does the impulse response function look like? Look at **Output 5.16**.

Output 5.16: Impulse Responses



The left panels of **Output 5.16** show the results of setting at time $t = 11$ the pair (Y_1, Y_2) to $(15, 7)$, $(15, 13)$, and $(17, 12)$. A change in either coordinate at time 11 results in the ultimate shifting of both coordinates. There can be a lot of wiggling as the new levels are approached. Or, there can be a relatively monotone approach of each coordinate to its new level.

These plots give insight of the behavior of these three impulse response functions and several others in the three-dimensional plots in the right panels.

The axes represent Y_1 , Y_2 , and time t . All series set $(Y_1, Y_2) = (15, 10)$ up to time $t = 11$, thus forming a *means axis*. The top right plot shows eight possible shocks at time $t = 11$, fanning out in an asterisk-shaped pattern. The middle-right plot adds in the eight resulting impulse response curves. The bottom right plot is just a rotated view of the middle plot, with time measured by depth into the plot. In the first and second plots, time increases with movement to the right, the height of a point is Y_1 , and its depth back into the plot is Y_2 . The plots include a 0 shock case that forms a continuation of the means axis. For a while after the shock at time 11, there can be substantial wiggling or relatively smooth movement. What is striking is that as time passes, the points all seem to align in a plane.

This plane is interpreted as a long-term relationship that will be approached over time after a shock bumps the point off of it (the plane). This gives rise to the term *error correction*, meaning that movement off the plane is an *error*. In the long run in the absence of shocks, the points will move back to the equilibrium represented by the plane—an *error correction*. The means axis is a horizontal line in this plane, but the bivariate forecast vector is only guaranteed to approach some horizontal line in the plane, not necessarily to the means line where the points started. There is not mean reversion in the individual coordinates. A single shock can send the system into fairly wild fluctuations that, depending on what the series represent, might frighten investors, but these are temporary and the vector ultimately will settle near the plane of equilibrium. This equilibrium plane is interpreted as a relationship that cannot be dramatically violated for long periods of time by the system. Envision the plane as an “attractor,” exerting a force like gravity on the points to settle them down after a shock.

Further insights are given by mathematics. A vector VAR(2) model of dimension k , $\mathbf{V}_t = \mathbf{A}_1 \mathbf{V}_{t-1} + \mathbf{A}_2 \mathbf{V}_{t-2} + \mathbf{e}_t$, can be algebraically written in terms of differenced vectors $\nabla \mathbf{V}_t = \mathbf{V}_t - \mathbf{V}_{t-1}$ and a lagged vector \mathbf{V}_{t-1} as $\nabla \mathbf{V}_t = -(\mathbf{I} - \mathbf{A}_1 - \mathbf{A}_2)\mathbf{V}_{t-1} - \mathbf{A}_2 \nabla \mathbf{V}_{t-1} + \mathbf{e}_t$, where $(\mathbf{I} - \mathbf{A}_1 - \mathbf{A}_2)$ is $\mathbf{I}m^2 - \mathbf{A}_1m - \mathbf{A}_2$ evaluated at $m = 1$. So, if $|\mathbf{I} - \mathbf{A}_1 - \mathbf{A}_2| = 0$ (that is, if this matrix is less than full rank), then the time series has a unit root $m = 1$. Any $k \times k$ matrix $\mathbf{\Pi}$ that has rank $r < k$ can be written as $\mathbf{\Pi} = \mathbf{a}\mathbf{\beta}'$, where \mathbf{a} and $\mathbf{\beta}$ are full-rank $k \times r$ matrices. Using the \mathbf{A} matrices currently under discussion, consider the following model:

$$\begin{pmatrix} Y_{1,t} - 15 \\ Y_{2,t} - 10 \end{pmatrix} = \begin{pmatrix} 1.84 & -0.24 \\ -0.06 & 1.66 \end{pmatrix} \begin{pmatrix} Y_{1,t-1} - 15 \\ Y_{2,t-1} - 10 \end{pmatrix} + \begin{pmatrix} -0.88 & 0.28 \\ 0.07 & -0.67 \end{pmatrix} \begin{pmatrix} Y_{1,t-2} - 15 \\ Y_{2,t-2} - 10 \end{pmatrix} + \begin{pmatrix} e_{1,t} \\ e_{2,t} \end{pmatrix}$$

It becomes

$$\begin{aligned} \begin{pmatrix} \nabla Y_{1,t} \\ \nabla Y_{2,t} \end{pmatrix} &= - \begin{pmatrix} 0.04 & -0.04 \\ -0.01 & 0.01 \end{pmatrix} \begin{pmatrix} Y_{1,t-1} - 15 \\ Y_{2,t-1} - 10 \end{pmatrix} + \begin{pmatrix} 0.88 & -0.28 \\ -0.07 & 0.67 \end{pmatrix} \begin{pmatrix} \nabla Y_{1,t-1} \\ \nabla Y_{2,t-1} \end{pmatrix} + \begin{pmatrix} e_{1,t} \\ e_{2,t} \end{pmatrix} \\ &= \begin{pmatrix} -0.04 \\ 0.01 \end{pmatrix} (1 - 1) \begin{pmatrix} Y_{1,t-1} - 15 \\ Y_{2,t-1} - 10 \end{pmatrix} + \begin{pmatrix} 0.88 & -0.28 \\ -0.07 & 0.67 \end{pmatrix} \begin{pmatrix} \nabla Y_{1,t-1} \\ \nabla Y_{2,t-1} \end{pmatrix} + \begin{pmatrix} e_{1,t} \\ e_{2,t} \end{pmatrix} \\ &= \begin{pmatrix} -0.04 \\ 0.01 \end{pmatrix} (Y_{1,t-1} - Y_{2,t-1} - 5) + \begin{pmatrix} 0.88 & -0.28 \\ -0.07 & 0.67 \end{pmatrix} \begin{pmatrix} \nabla Y_{1,t-1} \\ \nabla Y_{2,t-1} \end{pmatrix} + \begin{pmatrix} e_{1,t} \\ e_{2,t} \end{pmatrix} \end{aligned}$$

so that

$$\mathbf{\Pi} = - \begin{pmatrix} 0.04 & -0.04 \\ -0.01 & 0.01 \end{pmatrix} = \begin{pmatrix} -0.04 \\ 0.01 \end{pmatrix} (1 - 1) = \mathbf{a}\mathbf{\beta}'$$

The interpretation is that $S_t = Y_{1,t} - Y_{2,t} - 5$ is stationary—that is, it tends to be near 0 so that the difference $Y_{1,t} - Y_{2,t}$ tends to be near 5. This algebraic form of the model is known as the *error correction model* or ECM. The plane satisfying $Y_{1,t} = Y_{2,t} + 5$ at every t is the attractor toward which all the impulse response functions are moving in the three-dimensional plots. A vector (a, b) such that $(a, b)(Y_{1,t}, Y_{2,t})'$ is stationary is called a *cointegrating vector*. In this case, $\mathbf{\beta}' = (-1, 1)$ is such a vector as are $(-2, 2)$, $(1, -1)$, and any nonzero vector of the form $(-\phi, \phi) = \phi(-1, 1)$. The set of all linear combinations of the rows of $\mathbf{\beta}'$ constitutes the set of all possible cointegrating vectors in the general case.

Next, consider $N_t = Y_{1t} + 4Y_{2t} = (1, 4)(Y_{1t}, Y_{2t})'$. Note that $\nabla N_t = (1, 4)(\nabla Y_{1t}, \nabla Y_{2t})'$ due to the fact that $(1, 4)\Pi = 0$. Multiplying the vector equation on both sides by the row vector $(1, 4)$, it is seen that ∇N_t involves lagged levels only through the following term:

$$(1, 4) \begin{pmatrix} -0.04 \\ 0.01 \end{pmatrix} (Y_{1,t-1} - Y_{2,t-1} - 5) = 0$$

That is, ∇N_t in fact does not involve the lag levels of the variables at all. It is strictly expressible in terms of differences, so N_t is a unit root process. Also, because the only constant in the model, -5 , is captured in the term $Y_{1,t-1} - Y_{2,t-1} - 5$ and is thus annihilated in the ∇N_t equation, it follows that N_t has no drift. N_t is a stochastic *common trend* shared by Y_{1t} and Y_{2t} . The two interesting linear combinations, the nonstationary $N_t = Y_{1t} + 4Y_{2t}$ and the stationary $S_t = Y_{1t} - Y_{2t} - 5$, can be written as follows:

$$\begin{pmatrix} N_t \\ S_t \end{pmatrix} = \begin{pmatrix} 1 & 4 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} Y_{1t} \\ Y_{2t} \end{pmatrix} + \begin{pmatrix} 0 \\ -5 \end{pmatrix} = \mathbf{T} \begin{pmatrix} Y_{1t} \\ Y_{2t} \end{pmatrix} + \begin{pmatrix} 0 \\ -5 \end{pmatrix}$$

From this, you see that the following is true:

$$\begin{pmatrix} Y_{1t} \\ Y_{2t} \end{pmatrix} = \mathbf{T}^{-1} \begin{pmatrix} N_t \\ S_t + 5 \end{pmatrix} = \begin{pmatrix} 0.2 & 0.8 \\ 0.2 & -0.2 \end{pmatrix} \begin{pmatrix} N_t \\ S_t + 5 \end{pmatrix}$$

It becomes clear exactly how the nonstationary common trend N_t is part of both Y series. For $\Pi = \alpha\beta'$, with α and β both $k \times r$ matrices, the matrix \mathbf{T} can always be constructed by stacking α_p' above β' , where α_p' is a $(k-r) \times k$ matrix such that $\alpha_p'\alpha = 0$.

As a final insight, multiply both sides of the VAR in error correction form by the transformation matrix \mathbf{T} to get the following:

$$\begin{aligned} \mathbf{T} \begin{pmatrix} \nabla Y_{1t} \\ \nabla Y_{2t} \end{pmatrix} &= \begin{pmatrix} \nabla N_t \\ \nabla S_t \end{pmatrix} \\ &= \begin{pmatrix} 0 \\ -0.05 \end{pmatrix} (Y_{1,t-1} - Y_{2,t-1} - 5) + \mathbf{T} \begin{pmatrix} 0.88 & -0.28 \\ -0.07 & 0.67 \end{pmatrix} \mathbf{T}^{-1} \mathbf{T} \begin{pmatrix} \nabla Y_{1,t-1} \\ \nabla Y_{2,t-1} \end{pmatrix} + \mathbf{T} \begin{pmatrix} e_{1,t} \\ e_{2,t} \end{pmatrix} \\ &= \begin{pmatrix} 0 \\ -0.05 \end{pmatrix} S_{t-1} + \mathbf{T} \begin{pmatrix} 0.88 & -0.28 \\ -0.07 & 0.67 \end{pmatrix} \mathbf{T}^{-1} \begin{pmatrix} \nabla N_{t-1} \\ \nabla S_{t-1} \end{pmatrix} + \mathbf{T} \begin{pmatrix} e_{1,t} \\ e_{2,t} \end{pmatrix} \\ &= \begin{pmatrix} 0 \\ -0.05 \end{pmatrix} S_{t-1} + \begin{pmatrix} 0.60 & 0 \\ 0 & 0.95 \end{pmatrix} \begin{pmatrix} \nabla N_{t-1} \\ \nabla S_{t-1} \end{pmatrix} + \begin{pmatrix} z_{1,t} \\ z_{2,t} \end{pmatrix} \end{aligned}$$

Here the z white noise errors are linear combinations of the e errors. The coefficient matrix for the lagged differences of N and S is diagonal, which would not be the case in general. Nevertheless, there does always exist a transformation matrix \mathbf{T} such that the vector \mathbf{TV}_t contains the following processes:

- As many unit root processes as the series has unit roots.
- These are followed by stationary processes (provided none of the original Y series requires second differencing to achieve stationarity).

The period of the sinusoidal waves follows from the mathematical model. With the diagonal coefficient matrix in this example, it is easy to describe the stationary component as $\nabla S_t = -0.05S_{t-1} + 0.95\nabla S_{t-1} + z_{2,t}$ or $S_t = 1.9S_{t-1} - 0.95S_{t-2} + z_{2,t}$ with characteristic polynomial $m^2 - 1.9m + 0.95 = (m - \sqrt{0.95}e^{i\theta})(m - \sqrt{0.95}e^{-i\theta})$. Here the representation of a complex number as $re^{i\theta} = r[\cos(\theta) + i\sin(\theta)]$ can be used with the fact that $\sin(-\theta) = -\sin(\theta)$ to show that $\sqrt{0.95}(e^{i\theta} + e^{-i\theta}) = 2\sqrt{0.95} \cos \theta$ must equal 1.9 so that the angle θ is $\theta = \arccos(1.90/2\sqrt{0.95}) = 12.92$ degrees. In the graphs, you would expect $110(12.92 / 360) = 4$ cycles in the 110 observations after the shock. This is precisely what the graphs in **Output 5.16** show, with the following value giving the amplitude damping factor L periods after the shock:

$$\sqrt{0.95}^L$$

The relationship of S_t to Y_{1t} and Y_{2t} determines the orientation of this sinusoidal fluctuation in the three-dimensional plots. For the cases with equal shocks to Y_{1t} and Y_{2t} , no fluctuations were seen. That is because for these cases, $S_t = Y_{1t} - Y_{2t} - 5$ is no different after the shocks than before, so the shocked points are still in the cointegrating plane. With $\nabla N_t = 0.60 \nabla N_{t-1}$ describing the component of motion in the cointegrating plane, you would expect an exponential increase in the equal shock cases to a new horizontal line contained in the cointegrating plane. That, indeed, is what happens. The cases with unequal shocks to the two Y components force the point off the cointegrating plane, initiating a ripple-like fluctuation about the plane as the new levels are approached. In the bottom-right plot of **Output 5.16**, where passing time moves you toward the back of the plot, the cointegrating plane slopes from the upper left to the lower right while the sinusoidal fluctuations seem to move from lower left to upper right and back again, repeatedly, as time passes.

You have learned some of the terminology and seen some geometric implications of cointegration in a hypothetical model with known parameters. You have seen from the graphs and in more detail from the mathematical analysis that the error correction model defines a simple linear attractor (a line, plane, or hyperplane) toward which forecasts gravitate. It can capture some fairly complicated short-term dynamics. You now look for cointegrating relationships such as the S_t formula and for common trends such as N_t in the Amazon.com data.

For the Amazon.com stocks, it appeared that the relationship $\log(\text{high}) - \log(\text{low})$ was stationary with average value about 0.0765. In a three-dimensional plot of $H_t = \log(\text{high})$, $L_t = \log(\text{low})$, and time t , you would expect the points to stay close to a plane having $\log(\text{high}) = \log(\text{low}) + 0.0765$ over time. That plane and the data were seen in **Output 5.10**. In the upper-left panel, both series are plotted against time, and they almost overlay each other. The upper-right panel plots L_t versus t in the floor and H_t versus t in the back wall. These are projections into the floor and back wall of the points (t, L_t, H_t) , which are seen moving from the lower left to upper right while staying close to a sloping plane. This is the cointegrating plane. The lower-right panel shows this same output rotated so that points move out toward the observer as time passes. To its left, the rotation continues, so you are now looking directly down the edge of the cointegrating plane. This is also the graph of H_t versus L_t and motivates the estimation of the cointegrating plane by regression, as suggested by Engle and Granger (1987).

Using OLS regression, you estimate an error correction model of the following form:

$$\begin{pmatrix} \nabla L_t \\ \nabla H_t \end{pmatrix} = \begin{pmatrix} -0.4185 \\ 0.1799 \end{pmatrix} S_{t-1} + \begin{pmatrix} -0.0429 & 0.2787 \\ 0.4170 & -0.1344 \end{pmatrix} \begin{pmatrix} \nabla L_{t-1} \\ \nabla H_{t-1} \end{pmatrix}$$

Here H_t and L_t are log-transformed high and low prices, $S_t = H_t - L_t - 0.0765$, and 0.0765 is the estimated mean of $H_t - L_t$. Thus, $0.18L_t + 0.42H_t$ is the common trend unit root process. It can be divided by 0.6 to make the weights sum to 1, in which case the graph will, of course, look like those of the original series that were so similar to each other in this example. It is a weighted average of things that are almost the same as each other.

5.2.6 Estimation of the Cointegrating Vector

In the Amazon.com data, it was easy to guess that $\log(\text{high}/\text{low})$ would be stationary and that $\log(\text{high}) - \log(\text{low})$ is the cointegrating relationship between these two series. This made the analysis pretty straightforward. A simple unit root test on $\log(\text{high}) - \log(\text{low})$ sufficed as a cointegration test. The high and low prices are so tightly cointegrated that it is clear from the outset that the data will produce a nice example. In other cases, the data might not be so nice, and the nature of the cointegrating plane might not be easily anticipated as it was here. The complete cointegration machinery includes tests that several series are cointegrated and methods for estimating the cointegrating relationships. If you estimate that $H_t - bL_t$ is stationary in the Amazon.com example, you might want to test to see whether b is an estimate of 1.00 to justify the coefficient of L_t in $S_t = H_t - L_t - 0.0765$. The techniques include tests of such hypotheses about the cointegrating parameters.

The number of cointegrating relations in a process with known parameters is the rank of the coefficient matrix, $\mathbf{\Pi}$, on the lagged levels in the error correction representation. In the previous hypothetical known-parameter example, you saw that this matrix was the following:

$$\mathbf{\Pi} = - \begin{pmatrix} 0.04 & -0.04 \\ -0.01 & 0.01 \end{pmatrix} = \begin{pmatrix} -0.04 \\ 0.01 \end{pmatrix} (1 \quad -1) = \mathbf{\alpha} \mathbf{\beta}'$$

This is clearly a rank-one matrix. This factoring of the matrix not only shows that there is one cointegrating relationship, but it reveals its nature. From the vector $(1 \quad -1)$, the difference in the bivariate vector's elements is the linear combination that is stable—that is, it stays close to a constant. This happens to be the same cointegrating relationship that seemed to apply to the Amazon.com case. (It was displayed in the lower-left corner of **Output 5.10**.)

A vector time series of dimension 3 could move around anywhere in three-dimensional space as time passes. However, the points (Y_{1t}, Y_{2t}, Y_{3t}) will stay near the plane $Y_{1t} - 0.5Y_{2t} - 0.5Y_{3t} = C$ for some constant C as time passes if its lag level coefficient matrix is the following:

$$\boldsymbol{\Pi} = \begin{pmatrix} 0.1 \\ -0.3 \\ -0.2 \end{pmatrix} \begin{pmatrix} 1 & -0.5 & -0.5 \end{pmatrix}$$

This is a plane running obliquely through three-dimensional (Y_{1t}, Y_{2t}, Y_{3t}) space just as the line in the lower-left corner of **Output 5.10** runs obliquely through two-dimensional space. In this case, there is one cointegrating vector $(1, -0.5, -0.5)$ and therefore two common trends. You can think of these as two dimensions in which the series is free to float without experiencing a “gravitational pull” back toward the plane, just as the bivariate series was free to float up and down along the diagonal line in the lower-left corner of **Output 5.10**. Because time added to Y_{1t}, Y_{2t}, Y_{3t} introduces a fourth dimension, no graph analogous to the plane in the Amazon.com example is possible.

As a second example, consider this:

$$\boldsymbol{\Pi} = \begin{pmatrix} 0.2 & 0.2 \\ 0.1 & -0.5 \\ 0.8 & 0.1 \end{pmatrix} \begin{pmatrix} 1 & -0.5 & -0.5 \\ 1 & 0.2 & -0.6 \end{pmatrix}$$

Here, the points (Y_{1t}, Y_{2t}, Y_{3t}) will stay near the line formed by the intersection of two planes: $Y_{1t} - 0.5Y_{2t} - 0.5Y_{3t} = C_1$ and $Y_{1t} + 0.2Y_{2t} - 0.6Y_{3t} = C_2$. In this last example, there are two cointegrating vectors and one common trend. That is, there is one dimension, the line of intersection of the planes, along which points are free to float.

SAS/ETS software provides PROC VARMAX to do this kind of modeling as well as allowing exogenous variables and moving average terms (hence, the “X” and “MA” in VARMAX). A lag 1 and a lag 2 bivariate autoregression have been fit to the Amazon.com data, but no check has yet been provided as to whether 2 lags are sufficient. In fact, a regression of the log-transformed high and low stock prices on their lags indicates that 3 lags might in fact be needed. A popular method by Johansen is described next. It involves squared canonical correlations. Let \mathbf{W} and \mathbf{Y} be two random vectors. Pick a linear combination of elements of \mathbf{W} and one of \mathbf{Y} in such a way as to maximize the correlation. That correlation is the highest canonical correlation. Using only the linear combinations of \mathbf{W} that are not correlated with the first and similarly for \mathbf{Y} , pick the linear combination from each set that produces the most highly correlated pair. That's the second highest canonical correlation, etc.

Let \mathbf{W} and \mathbf{Y} be two random mean 0 vectors related by $\mathbf{Y} = \boldsymbol{\Pi}\mathbf{W} + \mathbf{e}$, where $\boldsymbol{\Pi}$ is a $k \times k$ matrix of rank r . Let $\boldsymbol{\Sigma}_{YY}$, $\boldsymbol{\Sigma}_{WW}$, and $\boldsymbol{\Sigma}$ denote the variance matrices of \mathbf{Y} , \mathbf{W} , and \mathbf{e} , and assume \mathbf{W} and \mathbf{e} are uncorrelated. Let $\boldsymbol{\Sigma}_{YW} = E\{\mathbf{Y}\mathbf{W}'\} = \boldsymbol{\Pi}\boldsymbol{\Sigma}_{WW}$. The problem of finding vectors $\boldsymbol{\gamma}_j$ and scalars λ_j for which $(\boldsymbol{\Sigma}_{YW}\boldsymbol{\Sigma}_{WW}^{-1}\boldsymbol{\Sigma}'_{YW} - \lambda_j\boldsymbol{\Sigma}_{YY})\boldsymbol{\gamma}_j = \mathbf{0}$ or, equivalently, $(\boldsymbol{\Sigma}_{YW}\boldsymbol{\Sigma}_{WW}^{-1}\boldsymbol{\Sigma}'_{YW} - \lambda_j\mathbf{I})\boldsymbol{\gamma}_j = \mathbf{0}$ is an eigenvalue problem. The solutions λ_j are the squared canonical correlations between \mathbf{Y} and \mathbf{W} , and because the rank of $\boldsymbol{\Pi}$ is r , there must be $k - r$ linearly independent vectors $\boldsymbol{\gamma}_j$ such that $\boldsymbol{\Sigma}'_{YW}\boldsymbol{\gamma}_j = \boldsymbol{\Sigma}'_{WW}\boldsymbol{\Pi}'\boldsymbol{\gamma}_j = 0$. For these, you can solve the eigenvalue equation using $\lambda_j = 0$. That is, there are $k - r$ eigenvalues equal to 0. Finding the number of cointegrating vectors r is equivalent to finding the number of nonzero eigenvalues for the matrix $\boldsymbol{\Sigma}_{YW}\boldsymbol{\Sigma}_{WW}^{-1}\boldsymbol{\Sigma}'_{YW}\boldsymbol{\Sigma}_{YY}^{-1}$. Johansen's test involves estimating these variance and covariance matrices and testing the resulting estimated eigenvalues.

Begin with a lag 1 model $\mathbf{V}_t = \mathbf{AV}_{t-1} + \mathbf{e}_t$ or $\nabla\mathbf{V}_t = -(\mathbf{I} - \mathbf{A})\mathbf{V}_{t-1} + \mathbf{e}_t$. Johansen's method (1988, 1991) consists of a regression of $\nabla\mathbf{V}_t$ on \mathbf{V}_{t-1} . That is, each element of $\nabla\mathbf{V}_t$ is regressed on all the elements in \mathbf{V}_{t-1} to produce the rows of the estimated $-(\mathbf{I} - \mathbf{A})$ coefficient matrix. For a lag 1 model, $(\mathbf{I} - \mathbf{A}) = \boldsymbol{\Pi} = \mathbf{a}\boldsymbol{\beta}'$ where the rows of $\boldsymbol{\beta}'$ are the cointegrating vectors and the following three numbers are all the same:

- r = the rank of $\mathbf{I} - \mathbf{A}$.
- r = the number of cointegrating vectors or rows of $\boldsymbol{\beta}'$.
- r = the number of nonzero squared canonical correlations between the element $\nabla\mathbf{V}_t$ and those of \mathbf{V}_{t-1} .

Johansen suggested studying the estimated squared canonical correlation coefficients to decide how many of them are significantly different from 0 and, thereby, estimate r . Standard procedures such as PROC CANCORR will deliver the desired estimates, just as an ordinary regression program will deliver the test statistics for a univariate unit root test, but not the right p -values. As with the univariate unit root tests, the distributions of tests based on the squared canonical

correlation coefficients are nonstandard for unit root processes, such as those found in the error correction model. Johansen tabulated the required distributions, enabling a test for r , the number of cointegrating vectors.

In the Amazon.com data, it appeared that β' could be taken as any multiple of the vector $\mathbf{H}' = (1, -1)$. Johansen also provides a test of the null hypothesis that $\beta = \mathbf{H}\phi$, where \mathbf{H} is (as in the case of the Amazon.com data) a matrix of known constants. The test essentially compares the squared canonical correlations between $\nabla\mathbf{V}_t$ and \mathbf{V}_{t-1} to those between $\nabla\mathbf{V}_t$ and $\mathbf{H}'\mathbf{V}_{t-1}$. If $\nabla\mathbf{V}_t = \alpha\beta'\mathbf{V}_{t-1} + \mathbf{e}_t$ and $\beta = \mathbf{H}\phi$, you can easily see that $\nabla\mathbf{V}_t = \alpha\phi'(\mathbf{H}'\mathbf{V}_{t-1}) + \mathbf{e}_t$, which motivates the test. In the Amazon.com data, if β' is some multiple of $\mathbf{H}' = (1, -1)$, you would expect the two squared canonical correlations between $\nabla\mathbf{V}_t$ and \mathbf{V}_{t-1} to consist of one number near 0 and another number nearly equal to the squared canonical correlation between $\nabla\mathbf{V}_t$ and $S_t = \text{spread} = \mathbf{H}'\mathbf{V}_{t-1} = (1, -1)\mathbf{V}_{t-1} = \log(\text{high}) - \log(\text{low})$. The test that the first number is near 0 is a test for the cointegrating rank and involves nonstandard distributions. Given that there is one cointegrating vector, the test that its form is $(1, -1)$ is the one involving comparison of two eigenvalues and, interestingly, is shown by Johansen to have a standard chi-square distribution under the null hypothesis in large samples.

5.2.7 Intercepts and More Lags

PROC VARMAX gives these tests and a lot of additional information for this type of model. Before using PROC VARMAX on the Amazon.com data, some comments about higher-order processes and the role of the intercept are needed. Up to now, vector \mathbf{V}_t was assumed to have mean vector $\mathbf{0}$, implying that no intercept was needed in the model. Suppose now that

$$\mathbf{V}_t - \boldsymbol{\mu} = \mathbf{A}(\mathbf{V}_{t-1} - \boldsymbol{\mu}) + \mathbf{e}_t$$

and

$$\mathbf{V}_t = \boldsymbol{\lambda} + \mathbf{A}\mathbf{V}_{t-1} + \mathbf{e}_t$$

where $\boldsymbol{\mu}$ is a vector of means. The first equation is referred to as the *deviations form* for the model. In order for these two equations to be equivalent, the intercept restriction $\boldsymbol{\lambda} = (\mathbf{I} - \mathbf{A})\boldsymbol{\mu}$ must hold. In other words, the assumption that the model can be written in this deviations form implies that $\boldsymbol{\lambda} = \mathbf{0}$. Subtracting $(\mathbf{V}_{t-1} - \boldsymbol{\mu})$ from both sides of the first equation and subtracting \mathbf{V}_{t-1} from both sides of the second gives the following equations:

$$\nabla\mathbf{V}_t = (\mathbf{A} - \mathbf{I})(\mathbf{V}_{t-1} - \boldsymbol{\mu}) + \mathbf{e}_t$$

and

$$\nabla\mathbf{V}_t = \boldsymbol{\lambda} + (\mathbf{A} - \mathbf{I})\mathbf{V}_{t-1} + \mathbf{e}_t$$

In the cointegration case, recall that $\mathbf{A} - \mathbf{I} = \alpha\beta'$, with α_p representing a matrix of the same dimensions as α such that $\alpha'_p\alpha = \mathbf{0}$ and $\alpha'_p(\mathbf{A} - \mathbf{I}) = \alpha'_p\alpha\beta' = \mathbf{0}$. Multiplying by α'_p displays the common trends in the vector process. The equations become the following:

$$\alpha'_p\nabla\mathbf{V}_t = \mathbf{0} + \alpha'_p\mathbf{e}_t$$

and

$$\alpha'_p\nabla\mathbf{V}_t = \alpha'_p\boldsymbol{\lambda} + \mathbf{0} + \alpha'_p\mathbf{e}_t$$

The elements of vector $\alpha'_p\mathbf{V}_t$ are seen to be driftless, random walks in the first equation because their first differences are white noise processes. The second equation appears to describe random walks with drift terms given by the elements of vector $\alpha'_p\boldsymbol{\lambda}$. Of course, once you remember the intercept restriction $\boldsymbol{\lambda} = (\mathbf{I} - \mathbf{A})\boldsymbol{\mu}$, you see that $\alpha'_p\boldsymbol{\lambda} = \alpha'_p(\mathbf{I} - \mathbf{A})\boldsymbol{\mu} = \mathbf{0}$. Nevertheless, some practitioners are interested in the possibility of an unrestricted (nonzero) drift in such data. Such data will display regular upward or downward trends. As in the case of univariate unit root tests, you might prefer to associate the unrestricted drift case with a deviations form that allows for such trends. Subtracting $\mathbf{V}_{t-1} - \boldsymbol{\mu} - \boldsymbol{\lambda}(t-1)$ from both sides of

$$\mathbf{V}_t - \boldsymbol{\mu} - \boldsymbol{\lambda}(t-1) = \mathbf{A}(\mathbf{V}_{t-1} - \boldsymbol{\mu} - \boldsymbol{\lambda}(t-1)) + \mathbf{e}_t$$

gives

$$\nabla \mathbf{V}_t - \boldsymbol{\lambda} = -(\mathbf{I} - \mathbf{A})(\mathbf{V}_{t-1} - \boldsymbol{\mu} - \boldsymbol{\lambda}(t-1)) + \mathbf{e}_t$$

Multiplying by $\boldsymbol{\alpha}'_p$ on both sides and remembering that $\boldsymbol{\alpha}'_p(\mathbf{I} - \mathbf{A})\boldsymbol{\mu} = \mathbf{0}$, you see that the common trends for this model are given by $\boldsymbol{\alpha}'_p \nabla \mathbf{V}_t = \boldsymbol{\alpha}'_p \boldsymbol{\lambda} + \mathbf{0} + \boldsymbol{\alpha}'_p \mathbf{e}_t$.

Existence of a vector of linear trends in the original model, when written in this new deviations form, could imply drifts ($\boldsymbol{\alpha}'_p \boldsymbol{\lambda} \neq \mathbf{0}$) in some of the common trend elements or not ($\boldsymbol{\alpha}'_p \boldsymbol{\lambda} = \mathbf{0}$). This is analogous to the suggestion to pair the unit root with drift null hypothesis with an alternative of stationarity around a linear trend in the univariate case. Further discussion about the role of the intercept in cointegration can be found in Johansen (1994).

In the case of higher-order models, such as $\mathbf{V}_t = \mathbf{A}_1 \mathbf{V}_{t-1} + \mathbf{A}_2 \mathbf{V}_{t-2} + \mathbf{e}_t$ or $\nabla \mathbf{V}_t = (\mathbf{A}_1 + \mathbf{A}_2 - \mathbf{I}) \mathbf{V}_{t-1} - \mathbf{A}_2 \nabla \mathbf{V}_{t-1} + \mathbf{e}_t$, the estimate of $(\mathbf{A}_1 + \mathbf{A}_2 - \mathbf{I})$ that would be obtained by multivariate multiple regression can be obtained in three steps as follows:

1. Regress $\nabla \mathbf{V}_t$ on $\nabla \mathbf{V}_{t-1}$, getting residual matrix $\mathbf{R}_{1,t}$.
2. Regress \mathbf{V}_{t-1} on $\nabla \mathbf{V}_{t-1}$, getting residuals $\mathbf{R}_{2,t-1}$.
3. Regress $\mathbf{R}_{1,t}$ on $\mathbf{R}_{2,t-1}$.

In higher-order models, then, you simply replace $\nabla \mathbf{V}_t$ and \mathbf{V}_{t-1} with $\mathbf{R}_{1,t}$ and $\mathbf{R}_{2,t-1}$, and follow the same steps as described earlier for a lag 1 model. In a lag p model, steps 1 and 2 would have regressors $\nabla \mathbf{V}_{t-1}, \dots, \nabla \mathbf{V}_{t-p+1}$. Furthermore, Johansen shows that seasonal dummy variables can be added as regressors without altering the limit distributions of his tests. The procedure has been described in a manner that emphasizes its similarity to univariate unit root testing. The reader familiar with Johansen's method might note that he uses a slightly different parameterization that places the lag levels at the furthest lag, rather than lag 1. For example, $\mathbf{V}_t = \mathbf{A}_1 \mathbf{V}_{t-1} + \mathbf{A}_2 \mathbf{V}_{t-2} + \mathbf{e}_t$ becomes $\nabla \mathbf{V}_t = (\mathbf{A}_1 - \mathbf{I}) \nabla \mathbf{V}_{t-1} + (\mathbf{A}_1 + \mathbf{A}_2 - \mathbf{I}) \mathbf{V}_{t-2} + \mathbf{e}_t$. The same *impact matrix* as it is called, $\boldsymbol{\Pi} = \mathbf{A}_1 + \mathbf{A}_2 - \mathbf{I}$, appears in either format. Inferences about its rank are the same either way.

5.2.8 PROC VARMAX

Returning to the Amazon.com data, you see that PROC VARMAX is used to produce some of the cointegration computations that have just been discussed:

```
proc varmax data=amazon;
  model high low/p=3 lagmax=5 ecm=(rank=1 normalize=high)
    cointtest;
  cointeg rank=1 h=(1, -1);
  output out=out1 lead=50;
  id t interval=day;
run;
```

This requests a vector autoregressive model of order 3, VAR(3), on variables HIGH and LOW. They are the log-transformed high and low prices for Amazon.com stock. Diagnostics of fit will be given up to lagmax = 5. The error correction model (ECM) is assigned a rank 1, meaning that the impact matrix $\mathbf{A}_1 + \mathbf{A}_2 + \mathbf{A}_3 - \mathbf{I} = \boldsymbol{\Pi} = \boldsymbol{\alpha}\boldsymbol{\beta}'$ is such that $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are column vectors (rank 1). Here, \mathbf{A}_1 , \mathbf{A}_2 , and \mathbf{A}_3 represent the VAR coefficient matrices. The NORMALIZE option asks PROC VARMAX to report the multiple of $\boldsymbol{\beta}'$ that has 1 as the coefficient of high. Recall that if $\boldsymbol{\beta}'\mathbf{V}_t$ is a stationary linear combination of elements of the random vector \mathbf{V}_t , then so is any multiple of it. The COINTTEST option asks for a test of the cointegrating rank, while the COINTEG statement tests the hypothesis that the cointegrating vector $\boldsymbol{\beta}'$ can be expressed as a multiple of $\mathbf{H}' = (1, -1)$. Only a few of the many items produced by PROC VARMAX are shown in **Output 5.17**.

Output 5.17: PROC VARMAX on Amazon.com Data, Part 1

Number of Observations	509
Number of Pairwise Missing	0

Simple Summary Statistics						
Variable	Type	N	Mean	Standard Deviation	Min	Max
HIGH	Dependent	509	3.12665	1.23624	1.06326	5.39929
LOW	Dependent	509	3.05067	1.22461	0.96508	5.31812

Cointegration Rank Test Using Trace						
H0: Rank=r	H1: Rank>r	Eigenvalue	Trace	Pr > Trace	Drift in ECM	Drift in Process
0	0	0.1203	65.4985	<.0001	Constant	Linear
1	1	0.0013	0.6559	0.4178		

Cointegration Rank Test Using Trace Under Restriction						
H0: Rank=r	H1: Rank>r	Eigenvalue	Trace	Pr > Trace	Drift in ECM	Drift in Process
0	0	0.1204	71.1499	<.0001	Constant	Constant
1	1	0.0123	6.2589	0.1714		

Whether or not the intercept restriction (that α'_p annihilates the intercept) is imposed, the hypothesis of $r = 0$ cointegrating vectors is rejected. For example, in the unrestricted case, Johansen's trace test has value 65.50 with p -value less than 0.0001, so $r = 0$ is rejected. The test for $r = 1$ versus $r > 1$ does not reject the $r = 1$ null hypothesis. Thus, Johansen's test indicates a single ($r = 1$) cointegrating vector and a single ($k - r = 2 - 1 = 1$) common trend. The tests are based on eigenvalues, as might be anticipated from the earlier discussion linking squared canonical correlations to eigenvalues. From the graph, it seems that a drift or linear trend term would be appropriate here, so the test without the restriction seems appropriate, though both tests agree that $r = 1$ anyway. Assuming a rank $r = 1$, the null hypothesis that α'_p annihilates the intercept is tested by comparing eigenvalues of certain matrices with and without this intercept restriction. The null hypothesis that the intercept restriction holds is rejected (shown in **Output 5.17a**) using the chi-square test 5.60 with 1 degree of freedom. Having seen **Output 5.10**, it is not surprising to find a drift in the common trend.

Output 5.17a: PROC VARMAX on Amazon.com Data, Part 2

Hypothesis Test of the Restriction					
Rank	Eigenvalue	Restricted Eigenvalue	DF	Chi-Square	Pr > ChiSq
0	0.1203	0.1204	2	5.65	0.0593
1	0.0013	0.0123	1	5.60	0.0179

Long-Run Parameter Beta Estimates		
Variable	1	2
HIGH	1.00000	1.00000
LOW	-1.01036	-0.24344

Adjustment Coefficient Alpha Estimates		
Variable	1	2
HIGH	-0.06411	-0.00209
LOW	0.35013	-0.00174

The long-run parameter estimates in **Output 5.17a** enable the user to estimate impact matrices $\Pi = \alpha\beta'$ of various ranks. For these data, the rank 1 and rank 2 versions of Π are the following two equations:

$$\begin{pmatrix} -0.06411 \\ 0.35013 \end{pmatrix} \begin{pmatrix} 1.00000 & -1.01036 \end{pmatrix} = \begin{pmatrix} -0.064 & 0.0648 \\ 0.350 & -0.3548 \end{pmatrix}$$

and

$$\begin{pmatrix} -0.06411 & -0.00209 \\ 0.35013 & -0.00174 \end{pmatrix} \begin{pmatrix} 1.00000 & -1.01036 \\ 1.00000 & -0.24344 \end{pmatrix} = \begin{pmatrix} -0.0662 & 0.065283 \\ 0.34839 & -0.353334 \end{pmatrix}$$

These are almost the same, as might be expected, because there was very little evidence from the test that the rank is greater than 1. In this computation, no restriction is made on the intercept.

Now, suppose \mathbf{W}_{t-1} is an augmented version of \mathbf{V}_{t-1} , namely, a vector whose last entry is 1 and whose first entries are the same as those of \mathbf{V}_{t-1} . For simplicity, consider the lag 1 model. Write the model as $\nabla\mathbf{V}_t = \alpha\beta'_+ \mathbf{W}_{t-1} + \mathbf{e}_t$, where β'_+ is the same as β' except for its last column. Recall the previous transformation matrix \mathbf{T} constructed by stacking α'_p above β' , where α'_p is a $(k-r) \times k$ matrix such that $\alpha'_p \alpha = \mathbf{0}$. Because $\nabla\alpha'_p \mathbf{V}_t = \alpha'_p \alpha \beta'_+ \mathbf{W}_{t-1} + \alpha'_p \mathbf{e}_t = \alpha'_p \mathbf{e}_t$, it follows that the elements of $\alpha'_p \mathbf{V}_t$ are driftless unit root processes. These are the first $k-r$ elements of $\mathbf{T}\mathbf{V}_t$. The last r elements are the stationary linear combinations. They satisfy $\nabla\beta' \mathbf{V}_t = \beta' \alpha \beta'_+ \mathbf{W}_{t-1} + \beta' \mathbf{e}_t$. The elements in the last column of $\beta' \alpha \beta'_+$ get multiplied by 1, the last entry of \mathbf{W}_{t-1} . In other words, they represent the intercepts for the stationary linear combinations. This shows how the addition of an extra element, a 1, to \mathbf{V}_{t-1} forces a model in which the unit root components do not drift. The result is the same in higher-order models. PROC VARMAX gives dummy variables for this case as well. Having previously rejected the restriction of no drift in the common trends, you are not really interested in these results that assume the restriction. In another data set, they might be of interest. They are shown here for completeness.

Output 5.17b: PROC VARMAX on Amazon.com Data, Part 3

Long-Run Coefficient Beta Based on the Restricted Trend		
Variable	1	2
HIGH	1.00000	1.00000
LOW	-1.01039	-0.79433
1	-0.04276	-1.48420

Adjustment Coefficient Alpha Based on the Restricted Trend		
Variable	1	2
HIGH	-0.05877	-0.00744
LOW	0.35453	-0.00614

5.2.9 Interpretation of the Estimates

A list of estimates is given in **Output 5.17c**.

Output 5.17c: PROC VARMAX on Amazon.com Data, Part 4

Model Parameter Estimates						
Equation	Parameter	❶ Estimate	Standard Error	t Value	Pr > t	Variable
D_HIGH	CONST1	0.00857	0.00414	2.07	0.0392	1
	AR1_1_1	-0.06411	0.07614			HIGH(t-1)
	AR1_1_2	0.06478	0.07693			LOW(t-1)
	AR2_1_1	0.04391	0.08169	0.54	0.5912	D_HIGH(t-1)
	AR2_1_2	0.19078	0.07785	2.45	0.0146	D_LOW(t-1)
	AR3_1_1	-0.09703	0.07200	-1.35	0.1784	D_HIGH(t-2)
	AR3_1_2	0.03941	0.06626	0.59	0.5523	D_LOW(t-2)
D_LOW	CONST2	-0.01019	0.00412	-2.47	0.0137	1
	AR1_2_1	0.35013	0.07571			HIGH(t-1)
	AR1_2_2	-0.35376	0.07649			LOW(t-1)
	AR2_2_1	0.29375	0.08123	3.62	0.0003	D_HIGH(t-1)
	AR2_2_2	0.01272	0.07741	0.16	0.8696	D_LOW(t-1)
	AR3_2_1	0.06793	0.07160	0.95	0.3432	D_HIGH(t-2)
	AR3_2_2	-0.13209	0.06589	-2.00	0.0455	D_LOW(t-2)

Covariances of Innovations		
Variable	HIGH	LOW
HIGH	0.00298	0.00229
LOW	0.00229	0.00295

The list of estimates ❶ shows that the fitted rank 1 model for the log-transformed high and low prices, H_t and L_t , is as follows:

$$\begin{pmatrix} \nabla H_t \\ \nabla L_t \end{pmatrix} = \begin{pmatrix} 0.00857 \\ -0.01019 \end{pmatrix} + \begin{pmatrix} -0.064 & 0.065 \\ 0.350 & -0.354 \end{pmatrix} \begin{pmatrix} H_{t-1} \\ L_{t-1} \end{pmatrix} + \begin{pmatrix} 0.044 & 0.191 \\ 0.294 & 0.013 \end{pmatrix} \begin{pmatrix} \nabla H_{t-1} \\ \nabla L_{t-1} \end{pmatrix} + \begin{pmatrix} -0.097 & 0.039 \\ 0.068 & -0.132 \end{pmatrix} \begin{pmatrix} \nabla H_{t-2} \\ \nabla L_{t-2} \end{pmatrix} + \begin{pmatrix} e_{1t} \\ e_{2t} \end{pmatrix}$$

Error variance matrix ❷ is:

$$\Sigma = \frac{1}{1000} \begin{pmatrix} 2.98 & 2.29 \\ 2.29 & 2.95 \end{pmatrix}$$

This indicates a correlation 0.77 between the errors.

5.2.10 Diagnostics and Forecasts

Output 5.17 is a series of diagnostics.

Output 5.17d: PROC VARMAX on Amazon.com Data, Part 5

Univariate Model ANOVA Diagnostics				
Variable	R-Square	Standard Deviation	F Value	Pr > F
HIGH	❷ 0.0558	0.05461	❶ 4.91	<.0001
LOW	0.1800	0.05430	18.25	<.0001

Univariate Model White Noise Diagnostics					
Variable	Durbin Watson	Normality		ARCH	
		Chi-Square	Pr > ChiSq	F Value	Pr > F
HIGH	1.97999	82.93	<.0001	19.06	<.0001
LOW	1.97865	469.45	<.0001	144.47	<.0001

Univariate Model AR Diagnostics								
Variable	AR1		AR2		AR3		AR4	
	F Value	Pr > F						
HIGH	0.02	0.8749	0.07	0.9294	0.97	0.4057	1.31	0.2666
LOW	0.00	0.9871	0.33	0.7165	0.65	0.5847	0.81	0.5182

The regression of ∇H_t on the lagged levels and two lagged differences of both H and L have a model F test 4.91 ❶ and R square 0.0558 ❷, and a similar line describing the ∇L_t regression is found just below this. The residuals from these models are checked for normality and unequal variance of the autoregressive conditional heteroscedastic (ARCH) type. Both of these departures from assumptions are found. The Durbin-Watson DW(1) statistics are near 2 for both residual series, and autoregressive models fit to these residuals up to 4 lags show no significance. These tests indicate uncorrelated residuals.

Recall that the spread $H_t - L_t$ was found to be stationary using a standard unit root test, and that the estimated cointegrating relationship was $H_t - 1.01L_t$. Given these findings, it is surprising that the test that $\beta' = \phi(1, -1)$ rejects that hypothesis. However, the sample size $n = 509$ is somewhat large. So, rather small and practically insignificant departures from the null hypotheses might still be statistically significant. In a similar vein, you might look at plots of residual histograms to see whether they are approximately bell shaped before worrying too much about the rejection of normality. The test that $\beta' = \phi(1, -1)$ is referred to as a test of the restriction matrix \mathbf{H} .

Output 5.17e: PROC VARMAX on Amazon.com Data, Part 6

Restriction Matrix H with Respect to Beta	
Variable	1
HIGH	1.00000
LOW	-1.00000

Long-Run Coefficient Beta with Respect to Hypothesis on Beta	
Variable	1
HIGH	1.00000
LOW	-1.00000

Adjustment Coefficient Alpha with Respect to Hypothesis on Beta	
Variable	1
HIGH	-0.07746
LOW	0.28786

Test for Restricted Long-Run Coefficient Beta					
Index	Eigenvalue	Restricted Eigenvalue	DF	Chi-Square	Pr > ChiSq
1	0.1203	0.1038	1	❶ 9.40	0.0022

The test compares eigenvalues 0.1038 and 0.1203 by comparing the following to a chi-square with 1 degree of freedom ❶:

$$(n-3)[\log(1-0.1038) - \log(1-0.1203)] = 506(0.01858) = 9.40$$

The fitted model implies one common trend that is a unit root with drift process and one cointegrating vector. The last bit of code requests forecasts using the VAR(3) in rank 1 error correction form. These are put into an output data set, a few observations from which are shown in **Output 5.17f**. An additional complication with these data is that the market is closed on the weekends, so the use of the actual dates as ID variables causes a missing data message to be produced. An easy fix is to use t = observation number as an ID variable, making the implicit assumption that the correlation between a Monday and the previous Friday is the same as between adjacent days. A portion of these data, including standard errors and upper and lower 95% confidence limits, is shown.

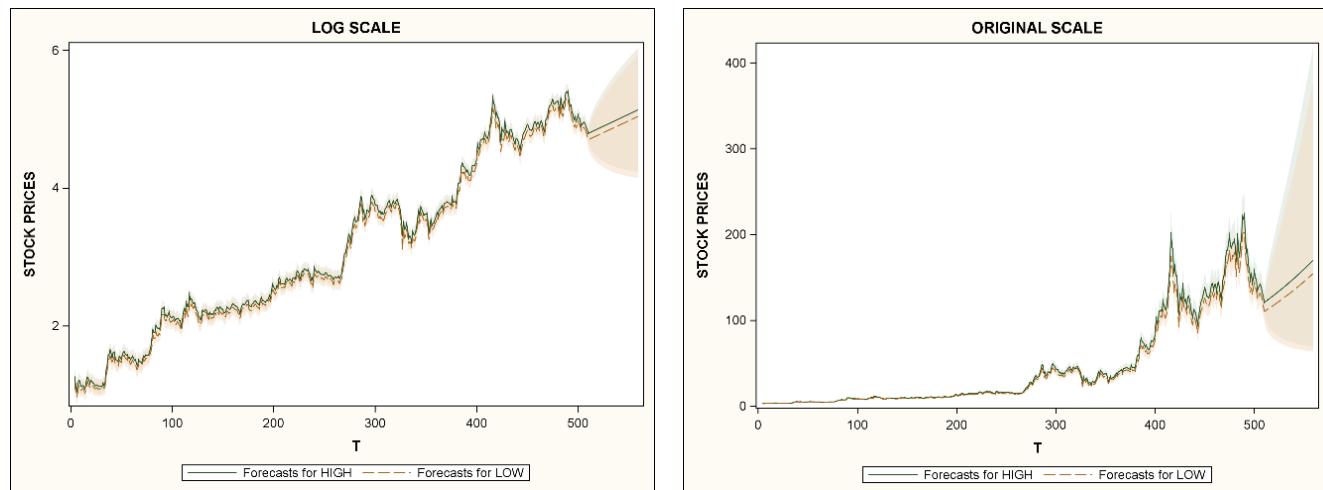
Output 5.17f: PROC VARMAX on Amazon.com Data, Last Part

Obs	T	HIGH	FOR1	RES1	STD1	LCI1	UCI1	LOW	FOR2
508	508	4.86272	4.88268	-0.019970	0.05461	4.77565	4.98972	4.75359	4.81222
509	509	4.79682	4.85402	-0.057193	0.05461	4.74698	4.96105	4.71290	4.75931
510	510	.	4.79125	.	0.05461	4.68422	4.89829	.	4.70442
511	511	.	4.80030	.	0.08475	4.63420	4.96640	.	4.70671
512	512	.	4.80704	.	0.10715	4.59704	5.01704	.	4.71564

Obs	RES2	STD2	LCI2	UCI2
508	-0.058634	0.05430	4.70580	4.91865
509	-0.046406	0.05430	4.65288	4.86574
510	.	0.05430	4.59799	4.81084
511	.	0.08605	4.53806	4.87537
512	.	0.10855	4.50288	4.92840

You can observe the quick spreading of confidence intervals, typical of data whose logarithms contain a unit root. The fact that the unit root is in some sense shared between the two series does not do much to narrow the intervals. The drift in the underlying unit root process or common trend is apparent in the forecasts. The short-term dynamics do not seem to contribute much to the forecasts, suggesting that the last few observations were quite near the cointegrating plane as seen in **Output 5.18**.

Output 5.18: Forecasts Using Cointegration



Chapter 6: Exponential Smoothing

6.1 Single Exponential Smoothing	209
6.1.1 The Smoothing Idea.....	209
6.1.2 Forecasting with Single Exponential Smoothing	210
6.1.3 Alternative Representations.....	210
6.1.4 Atlantic Ocean Tides: An Example	211
6.1.5 Improving the Tide Forecasts	213
6.2 Exponential Smoothing for Trending Data	216
6.2.1 Linear and Double Exponential Smoothing	216
6.2.2 Properties of the Forecasts	217
6.2.3 A Generated Multi-Series Example	217
6.2.4 Real Data Examples.....	219
6.2.5 Boundary Values in Linear Exponential Smoothing	222
6.2.6 Damped Trend Exponential Smoothing	228
6.2.7 Diagnostic Plots	229
6.2.8 Sums of Forecasts	231
6.3 Smoothing Seasonal Data	232
6.3.1 Seasonal Exponential Smoothing.....	232
6.3.2 Winters Method.....	234
6.4 Diagnostics	236
6.4.1 Validation	236
6.4.2 Choosing a Model Visually	237
6.4.3 Choosing a Model Numerically	239
6.5 Advantages of Exponential Smoothing	240
6.6 How the Smoothing Equations Lead to ARIMA in the Linear Case	240

6.1 Single Exponential Smoothing

The idea of exponential smoothing is that often, time series evolve in such a way that the level, trend, or seasonality changes slowly over time. This renders older data less relevant for forecasting than more recent data. The data are nonstationary. Thus far, differencing has been the only tool for dealing with such series. Exponential smoothing was developed as a method for downweighting past data. Perhaps surprisingly, the resulting forecast formulas from many of these models match those from ARIMA(p,1,q) models.

6.1.1 The Smoothing Idea

The idea of exponential smoothing is to estimate a model locally by using all the observations, but weighting the most recent ones more than those in the past. Models can have a local mean, local trend, and local seasonal pattern. Interestingly, many of these models are equivalent to certain subsets of the ARIMA models. This equivalency is exploited in this section. An example of a local-level model is the random walk model

$$Y_t = Y_{t-1} + e_t$$

that forecasts all future values as being the same as the current value. It gives all the weight to the most recent value and none to its predecessors. The current value is the best estimate of all future values in a random walk model. The single (or simple) exponential smoothing algorithm predicts all future values as the weighted sum $\omega \sum_{j=1}^n (1-\omega)^j Y_{n-j}$, where n is the number of observed Y values. In addition, $0 < \omega < 1$ and terms involving values of Y before the beginning of the

observed series are ignored. The algorithm gives a forecast only. To get forecast standard errors, the relationship between this algorithm and the forecasts from an ARIMA(0,1,1) are exploited in the next section. Viewing exponential smoothing as arising from an ARIMA model allows estimation of the parameters. Estimation is constrained to enforce the $0 < \omega < 1$ requirement.

6.1.2 Forecasting with Single Exponential Smoothing

An ARIMA model that is like the random walk model, but more flexible is $Y_t - Y_{t-1} = e_t - \theta e_{t-1}$ with $|\theta| < 1$. This is the integrated moving average model of order 1, ARIMA(0,1,1) or IMA(1,1). In fact, when differencing, it is a good idea to start the modeling by using a moving average term at the lag implied by whatever orders of differencing are used. In that way, a value of θ close to 1 can be an indicator of over-differencing. Assuming a positive value $0 < \theta < 1$, a series of back substitutions yields an interesting representation for Y as follows. The IMA(1,1) model holds at all times, so that $Y_{t-j} - Y_{t-j-1} = e_{t-j} - \theta e_{t-j-1}$ for any j . Thus, each e_{t-j} is $e_{t-j} = Y_{t-j} - Y_{t-j-1} + \theta e_{t-j-1}$. Substituting for e_{t-1} in $e_t = (Y_t - Y_{t-1}) + \theta e_{t-1}$ results in the following:

$$e_t = (Y_t - Y_{t-1}) + \theta e_{t-1} = (Y_t - Y_{t-1}) + \theta((Y_{t-1} - Y_{t-2}) + \theta e_{t-2}) = (Y_t - Y_{t-1}) + \theta(Y_{t-1} - Y_{t-2}) + \theta^2 e_{t-2}$$

Substituting for e_{t-2} gives this:

$$e_t = (Y_t - Y_{t-1}) + \theta(Y_{t-1} - Y_{t-2}) + \theta^2 e_{t-2} = Y_t - (1-\theta)Y_{t-1} - \theta(1-\theta)Y_{t-2} - \theta^2 Y_{t-3} + \theta^3 e_{t-3}$$

Continuing through n such back substitutions, you have the following expression:

$$e_t = Y_t - (1-\theta)Y_{t-1} - \theta(1-\theta)Y_{t-2} - \theta^2(1-\theta)Y_{t-3} - \cdots - \theta^{n-1}(1-\theta)Y_{t-(n+1)} - \theta^n Y_{t-n} - \theta^n e_{t-(n+1)}$$

The last term approaches 0 with increasing n . Ignoring it and letting n increase results in an infinite sum. Rearranging gives $Y_t = ((1-\theta)Y_{t-1} + \theta(1-\theta)Y_{t-2} + \theta^2(1-\theta)Y_{t-3} + \cdots) + e_t$, so that a forecast of Y_{t+1} from data up through Y_t could be written as the following infinite sum:

$$\hat{Y}_{t+1} = (1-\theta) \sum_{j=0}^{\infty} \theta^j Y_{t-j}$$

This forecast formula is exponential smoothing with $\omega = (1-\theta)$. Because $0 < \theta < 1$, the geometric series sums to a finite number:

$$\sum_{j=0}^{\infty} \theta^j = 1 / (1-\theta)$$

All the coefficients on lagged Y s add to 1 and they are all positive, implying that the forecast is a proper weighted average of all past Y values with exponentially declining weights. The sum can be truncated with hardly any loss of accuracy. Because the moving average order 1 error term has no correlation with any future values beyond the first one, the forecast of $Y_{t+2} - Y_{t+1}$ is 0, indicating that the forecast of Y_{t+2} is the same as Y_{t+1} , as is the forecast for any Y_{t+j} with $j > 1$. This model can produce only a horizontal line forecast. Further, because the model involves a difference, the forecast standard error bands will grow without bound as they do for any such nonstationary process. The process of using an exponentially weighted average of past data as a forecast is attributed to R. G. Brown. It was developed long before ARIMA models came into fashion (Brown 1956 and Holt 1957). Authors have suggested various schemes for selecting θ , such as simple rules of thumb, but one advantage of estimating θ from the equivalent ARIMA representation is the associated prediction standard error formula.

6.1.3 Alternative Representations

Another thing to notice about the one-step-ahead forecast at time t ,

$$\hat{Y}_{t+1} = (1-\theta) \sum_{j=0}^{\infty} \theta^j Y_{t-j}$$

is that the next forecast (at time $t + 1$ for the time $t + 2$ value) is as follows:

$$\hat{Y}_{t+2} = (1 - \theta) \sum_{j=0}^{\infty} \theta^j Y_{t+1-j}$$

which becomes

$$\hat{Y}_{t+2} = (1 - \theta) \left(Y_{t+1} + \theta \sum_{j=0}^{\infty} \theta^j Y_{t-j} \right) = (1 - \theta) Y_{t+1} + \theta \hat{Y}_{t+1}$$

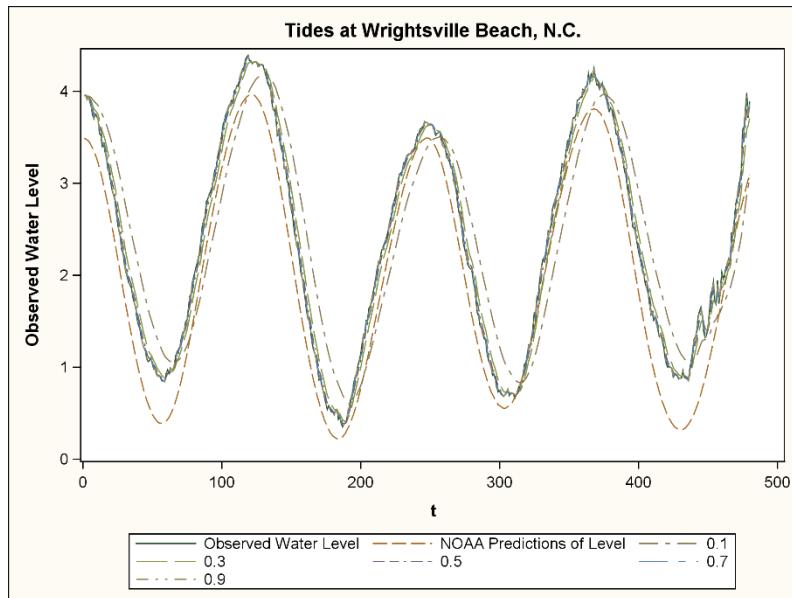
When period $t + 1$ arrives and Y_{t+1} is observed, the forecast of \hat{Y}_{t+2} is a simple weighted average of the newly observed Y_{t+1} and its former forecast \hat{Y}_{t+1} . When presented in this fashion, $(1 - \theta)$ is typically symbolized ω , and ω is called the *smoothing weight*. The smoothed value at time $t + 1$, which is the forecast for time $t + 2$, is written $S_{t+1} = \hat{Y}_{t+2} = \omega Y_{t+1} + (1 - \omega) \hat{Y}_{t+1}$. Because this holds at all times, authors typically drop the subscripts back one, writing $S_t = \omega Y_t + (1 - \omega) S_{t-1}$, where S_{t-1} is the smoothed value at time $t - 1$, which serves as a forecast for time t . This is the parameterization underlying the SAS procedures that do exponential smoothing. Because this forecasting process requires only the previous forecast and the current data, the computations were very easy before modern computers were available. They could easily be used, for example, in process control situations where you are looking for observations that differ substantially from their forecast. Because of the differencing, the mean of the series drops out. Forecasts are not mean reverting and tend to be reasonable for forecasting a step or two ahead in many practical situations. One last point about $S_{t+1} = \hat{Y}_{t+2} = \omega Y_{t+1} + (1 - \omega) \hat{Y}_{t+1}$ is that it can be re-expressed as $S_{t+1} = \hat{Y}_{t+2} = \hat{Y}_{t+1} + \omega(Y_{t+1} - \hat{Y}_{t+1})$ so that each forecast is its predecessor, plus some multiple of the most recent forecast error, which is $e_{t+1} = (Y_{t+1} - \hat{Y}_{t+1})$. For that reason, this formulation is called the error correction form and is typically written $S_t = \hat{Y}_t + \omega e_{t-1} = S_{t-1} + \omega e_{t-1}$. Clearly, the larger ω is, the more responsive the smoothed value is to the most recent change. The smaller ω is, the more stable the smoothed series of forecasts is and the more weight it gives to the historical data.

6.1.4 Atlantic Ocean Tides: An Example

As an example of the last point, consider the Atlantic Ocean's water levels and their predictions by National Oceanic and Atmospheric Administration (NOAA) at Wrightsville Beach, North Carolina, just before the passage of Hurricane Arthur in 2014. (Data were downloaded from a now discontinued NOAA database.) At that time, the site contained a disclaimer that the data were preliminary. Each point represents a six-minute time interval. There are 480 data points starting at 11:06 a.m., local time, on July 1, 2014, a 48-hour period. The data are plotted and analyzed with PROC SGPlot and PROC ESM.

The plot in **Output 6.1** consists of the observed water levels (solid line); smoothed levels using $\omega = 0.1, 0.3, 0.5, 0.7$, and 0.9 ; and the levels predicted by NOAA (lowest plot, equal dashes). The smooth plot that appears right-most (dashes and dots) uses $\omega = 0.1$, meaning that it gives low weight to the incoming observation and high weight to the past forecast. It is less responsive to incoming data and, as a result, is smooth and delayed when compared to the data, especially considering that the smoothed value at time t is a forecast of the time $t + 1$ water level.

```
proc sgplot data=tides;
  series x=t y=water_level;
  series x=t y=noaa_pred;
  series x=t y=smooth1;
  series x=t y=smooth3;
  series x=t y=smooth5;
  series x=t y=smooth7;
  series x=t y=smooth9;
run;
```

Output 6.1: Tides at Wrightsville Beach, NC

The plot shows the smoothed value at time t , which is a forecast of the time $t + 1$ observation. The actual observations would need to be shifted one period to the left to interpret the smoothed values as forecasts. This causes an even further separation between the actual water level and the forecast using $\omega = 0.1$. The larger weights give forecasts that are more responsive to the incoming data, wiggle more, and are better aligned with the data. They are almost indistinguishable from the data plot, meaning that they forecast the next value with something close to the current one. You see something very close to a random walk forecast when the weight ω is very close to 1.

Which level of smoothing is best among the five? The sums of squared differences between forecasts and observed water levels are 141.7, 21.2, 8.7, 5.1, and 3.7 for $\omega = 0.1, 0.3, 0.5, 0.7$, and 0.9, respectively. It seems that the largest weight $\omega = 0.9$ worked the best. This is the least squares estimate among these five, but knowing that the model can be thought of as an IMA(1,1) model, a program could be used to estimate the weight across all possibilities. Perhaps an even larger weight on the incoming data would be better, giving a forecast even closer to a random walk forecast. Recall that a random walk forecast uses each observation as a forecast of the next and it has weight $\omega = 1$. That is, the moving average coefficient is $\theta = 0$. In **Output 6.1**, it is clear that such a random walk forecast would not be too bad if a one-step-ahead forecast is all that is desired. This is because these are frequent measurements of a slowly and smoothly varying series.

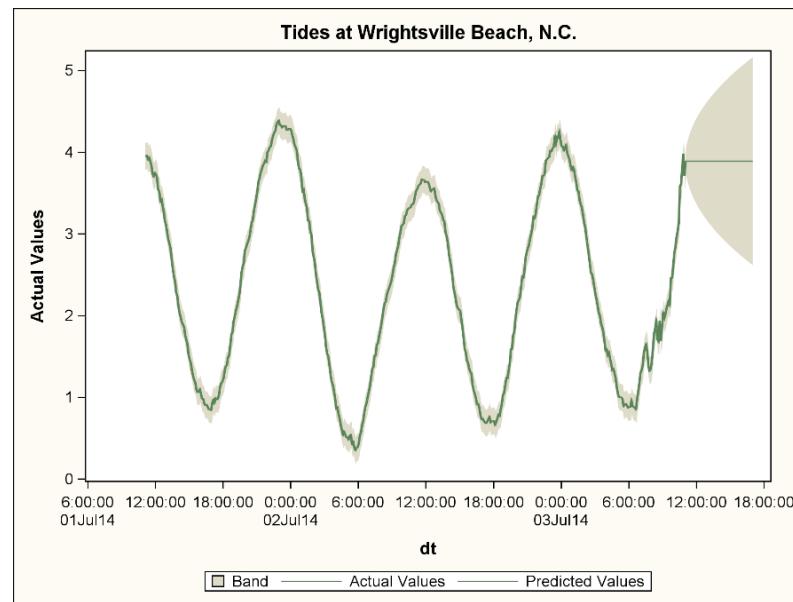
PROC ESM is applied to the data. The variable DT is a SAS datetime variable, and six-minute increments are specified in the ID statement. The LEAD option requests 60 forecasts into the future. The default model in PROC ESM is the simple or single exponential smoothing model, the one just discussed.

```
proc esm data = tides outfor=esm_fore outest=betas lead=60;
  id dt interval=minute6;
  forecast water_level;
run;
proc print data=betas noobs;
run;
```

Output 6.2: Parameter Estimates from PROC ESM

<u>_NAME_</u>	<u>_TRANSFORM_</u>	<u>_MODEL_</u>	<u>_PARM_</u>	<u>_EST_</u>	<u>_STDERR_</u>	<u>_TVALUE_</u>	<u>_PVALUE_</u>
WATER_LEVEL	NONE	SIMPLE	LEVEL	0.999	0.032450	30.7862	1.3217E-115

The estimated smoothing weight $\omega = 0.999$ is cause for concern. Can it be trusted? This will be discussed after first looking at a plot of the forecasts, data, and 95% prediction intervals. For illustration, forecasts 6 hours (60 observations) ahead are requested.

Output 6.3: PROC ESM Forecasts for One-Step-Ahead Historically and 60 Periods into the Future

The one-step-ahead forecasts are very close to random walk forecasts in which each observation is the predictor of the next. The forecasts and one-step-ahead prediction intervals are very accurate during the course of the data, which is to be expected. After all, this is just a prediction of what will happen in the next six minutes, unlike the longer-range forecasts that NOAA produces. The error bands grow rapidly in the forecast horizon as the lead time increases. This is because of the unit root ARIMA model, IMA(1,1), that underlies the estimation. A simple exponential smoothing model always produces a horizontal line forecast as has been seen in the earlier discussion. At least the prediction intervals warn the analyst that there is a lot of uncertainty in forecasts more than a step or two ahead. In some applications, with the easy updating of forecasts, these short-term forecasts might be sufficient. One-step-ahead forecasts are likely of little interest for six-minute tide measurements.

6.1.5 Improving the Tide Forecasts

Quite a bit is known about tides. The pull of the moon on the tides is obvious in **Output 6.1** and **Output 6.3**. The naïve model used here does not incorporate any of that knowledge. It has no way of selecting the pattern that is obvious to the viewer. Another problem with this model is the estimated value 0.999 for ω . This most likely came from the software restriction that keeps $0 < \omega < 1$. That is, the estimate hit the software-imposed boundary as opposed to having converged to this number. Without convergence, inference based on standard errors and p -values is not rigorously justified by statistical theory.

To see one way this might have occurred, **Output 6.4** shows a generated series of 480 observations from the model $Y_t = \beta_0 + \beta_1 X_{1t} + \beta_2 X_{2t} + e_t$, where X_1 and X_2 are a sine and cosine pair with period 120, and e_t is white noise. This is just the typical multiple regression model. The plot is similar to the tides and the form of the model implies that the best prediction is obtained by simply regressing Y on X_1 and X_2 . Future values of X_1 and X_2 are easily obtained. Here is the code for producing **Output 6.4**:

```

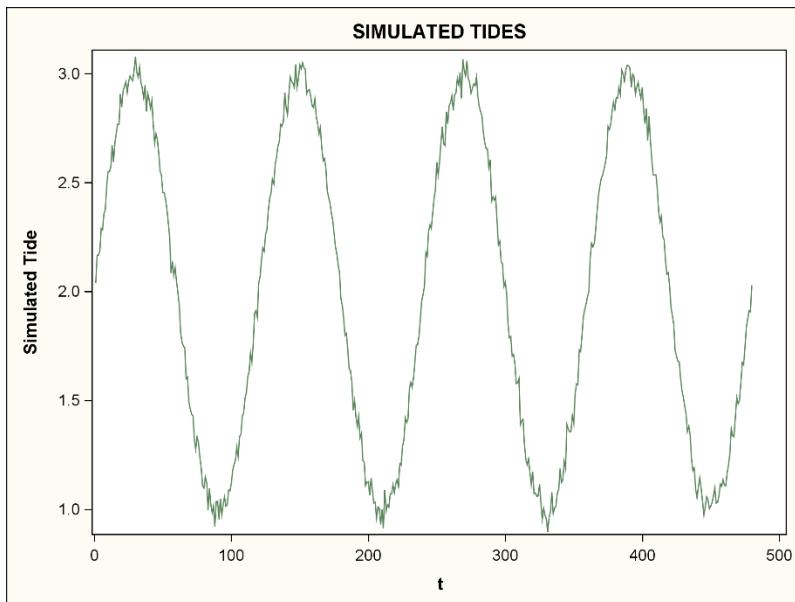
data simexp;
  do t=1 to 600;
    if t<481 then y = 2 + sin(2*constant("pi")*t/120) + 0.04*normal(123);
    else y=.;
    label y="simulated tide";
    output;
  end;
run;
proc sgplot;
  where t<481;
  series y=y x=t;
  title "simulated tides";
run;
proc esm data = simexp out=esm_fore outest=betas;
  forecast y;
  title "simulated tides";
run;

```

```
proc print data=betas noobs;
run;
```

Output 6.4 shows the results of estimating the smoothing weight with PROC ESM. It shows that an estimate 0.999 might have arisen from ignoring the sine-like feature of the data.

Output 6.4: Simulated Tides



<u>_NAME_</u>	<u>_TRANSFORM_</u>	<u>_MODEL_</u>	<u>_PARM_</u>	<u>_EST_</u>	<u>_STDERR_</u>	<u>_TVALUE_</u>	<u>_PVALUE_</u>
Y	NONE	SIMPLE	LEVEL	0.999	0.032298	30.9307	2.9672E-116

For this model, a random walk forecast for one step ahead would not be too bad because X_1 and X_2 change by a very small amount as time increases by 1. The error variance is relatively small so that Y is slowly changing. However, the random walk does not give the best forecast. Statistical theory (assuming this known model form) shows that the best forecast is from a multiple regression and would involve no autocorrelation. Printed results in **Output 6.4**, using the same type of code, gives very similar results to the actual tides.

The example shows that in at least some cases for short-term forecasts, but not for long-range forecasts, exponential smoothing does a reasonable job. When data are actually from $Y_t - Y_{t-1} = e_t - \theta e_{t-1}$ with $0 < \theta < 1$, it in fact gives the best forecast. The example suggests that when the weight ω hits the boundary $\omega = 0.999$, one possible reason is that there is some input that the model is ignoring. PROC ESM does not accommodate inputs. A possible approach to the tides forecasting problem is to use the deviations of the observations Y from the predicted values from NOAA as the response variable. Call this deviation series D_t . The NOAA series captures the large-scale movements in the series and contains future values. Adding forecasts of D_t to the NOAA numbers might provide an improved forecast as well as delivering an estimated weight parameter well within the $(0,1)$ interval.

To implement the analysis incorporating D_t , the following code is used. The response variable DIFFERENCE is D_t .

```
* smooth difference (observed - noaa prediction);
proc esm data=tides outfor=esm_fore outest=betas lead=110;
  where dt < "03jul14:00:06"dt;
  id dt interval=minute6;
  forecast difference;
run;
proc print data=betas noobs;
run;
proc sgplot data=esm_fore;
  scatter x=dt y=actual;
  label actual = "observed - noaa";
  series x=dt y=predict;
title "Predicting Differences From NOAA Forecast";
```

```

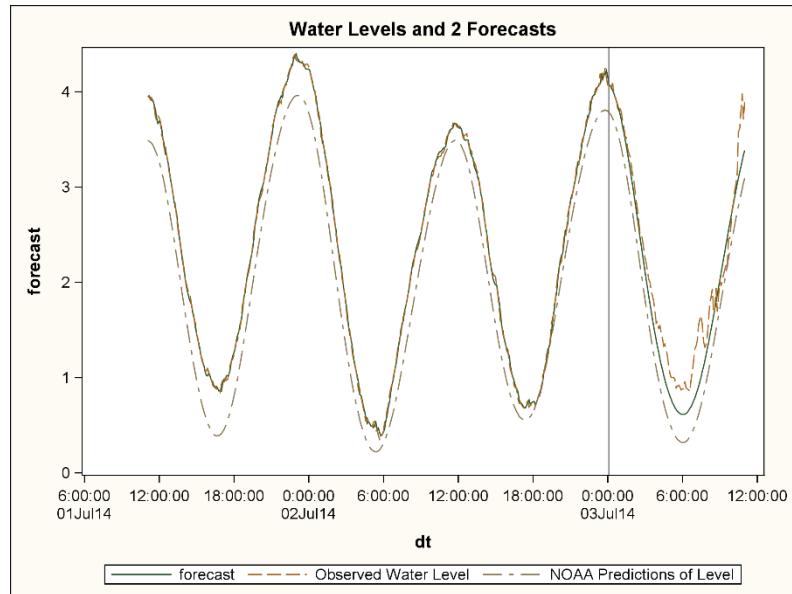
run;
* add predicted difference to noaa forecast ;
data all;
  merge tides esm_fore;
  forecast = noaa_pred+predict;
proc sgplot;
  title "water levels and 2 forecasts";
  series y=forecast x=dt;
  series y=water_level x=dt;
  series y=noaa_pred x=dt;
  reftime "03jul14:00:06" dt / axis=x;
run;

```

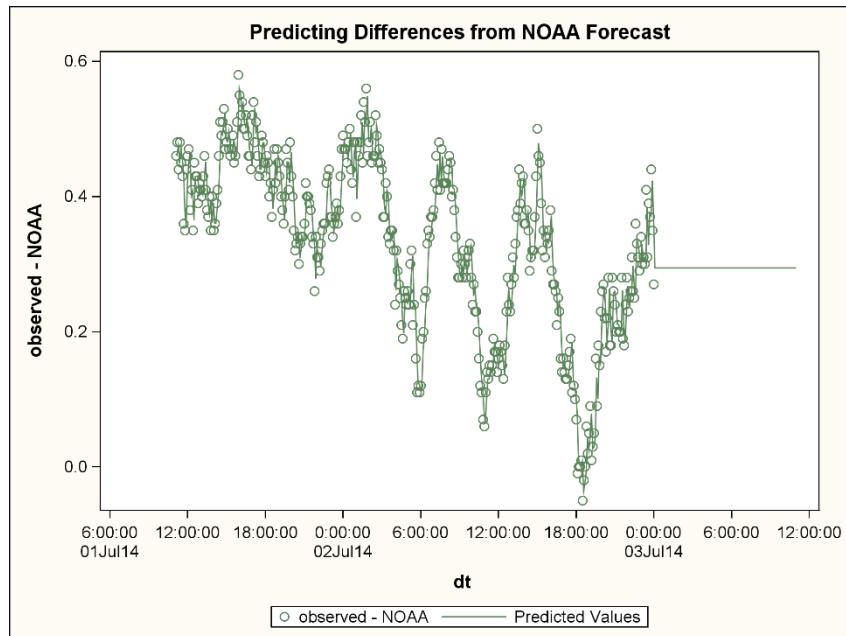
Output 6.5 shows the actual tides and relatively long-range predictions from the NOAA website. Clearly for a forecast of the next few hours, this NOAA forecast would be preferred to the horizontal line forecast from a simple exponential smoothing model fit in PROC ESM. An exponential smoothing model for the differences D_t between the observations and the NOAA forecasts might provide a mechanism to help assimilate the actual observations into the forecast.

Output 6.5 includes three series. The solid line is the result of running an exponential smoothing model on the differences of the actual data and NOAA forecasts up through the end of July 2, and then adding that forecast (recall that it is constant) of the differences to the NOAA forecasts. The vertical line is at the end of July 2. Throughout the plot, the lower dashed line shows the NOAA forecasts. To the left of the vertical line, the one-step-ahead exponential smoothing adjusted forecasts and data are almost indistinguishable. To the right, the data are above both forecasts and are represented by a dashed line. The solid line in the middle is the modified forecast, which is seen in every case to be closer to the actual data than the NOAA long-run forecast. The modification does not use the data after July 2 in any way, so the modified forecast is a true forecast, based only on observed data as of July 2. The modification seems to help, and the forecast is now for a long-enough period ahead to be of some interest to boaters and beach owners.

Output 6.5: NOAA Long-Run Forecasts of Tides with and without Modifications from PROC ESM



The forecast error sums of squares over the 110 forecasted values are 8.12 for the modified NOAA forecasts and 29.41 for the unmodified forecasts. The exponential smoothing of the differences between observed and predicted has resulted in one-step-ahead forecasts visually indistinguishable from the observed data. It seems to have improved the long-run forecasts quite a bit for this particular set of data. The smoothing weight is $\omega = 0.75$, so now $\theta = 0.25$, which is well within $0 < \theta < 1$, rather than being near either the 0 or 1 boundary. **Output 6.6** shows the differences D_t between the observations and long-run NOAA predictions. This is the series that was smoothed to assimilate the data. The increased volatility of that series with increasing time might be due to the approaching storm. The simple exponential smoothing predictions from PROC ESM are overlaid on these differences and extended into the future as well.

Output 6.6: Adjustments to NOAA Forecasts

<u>_NAME_</u>	<u>_TRANSFORM_</u>	<u>_MODEL_</u>	<u>_PARM_</u>	<u>_EST_</u>	<u>_STDERR_</u>	<u>_TVALUE_</u>	<u>_PVALUE_</u>
DIFFERENCE	NONE	SIMPLE	LEVEL	0.75450	0.036042	20.9339	1.0601E-64

The differences produce a nice smoothing value $\omega = 0.75$, which is somewhat responsive to the most recent forecast error. In **Output 6.5**, the constant forecast is a helpful addition, at least for several hours ahead, to the NOAA long-run forecasts. Additional features like the periodic behavior with increasing amplitude as the storm approaches might improve the data assimilation even further, but the simple constant forecast already provides a substantial improvement in the forecast. In essence, the exponential smoothing has provided a local shift to the long-run NOAA forecast that helps at least for several hours.

6.2 Exponential Smoothing for Trending Data

Simple exponential smoothing is useful in many situations. As the name implies, it is *simple*. Variations of the method have been suggested to extend it to data sets that are more complicated than those for which simple smoothing is appropriate. Linear and double exponential smoothing models allow for local linear trends that change over time. The level and trend at the end of the series are used to establish a linear forecast into the future. Estimates of the weights that hit the software-imposed boundary might suggest a possibility for model improvement, but do not necessarily imply unreasonable forecasts.

6.2.1 Linear and Double Exponential Smoothing

Single exponential smoothing produces smoothed versions S_t of the series Y_t at time t , each of which serves as a prediction of the series at time $t + 1$. This is done by taking a weighted average of Y_t and the previous prediction S_{t-1} , both of which could be considered as estimates of the time t smoothed value S_t , which in turn is an estimate of the local level of the series.

In a series with both a local level L_t and a local trend (for example, local slope) T_t , you might think of smoothing both the level and the trend. To see how this works, consider moving from time $t-1$ to time t . Using the trend at time $t-1$, one estimate of the next level L_t would be $L_{t-1} + T_{t-1}$ because T_{t-1} is the local slope at time $t-1$. It is an expected one-step-ahead increment in L_{t-1} . Taking a weighted average of $(L_{t-1} + T_{t-1})$ and Y_t (thinking of Y_t as another estimate of the level, available at time t) gives the smoothing formula for the level at time t , namely $L_t = \omega Y_t + (1 - \omega)(L_{t-1} + T_{t-1})$. As before, this is both the smoothed level estimate at time t and a forecast of the local level at time $t+1$. The change in level, $L_t - L_{t-1}$, is an estimate of the slope available at time t . Starting with the smoothed estimate T_{t-1} , the trend estimate is updated to time t using the same type of formula, $T_t = \gamma(L_t - L_{t-1}) + (1 - \gamma)T_{t-1}$. The equivalent ARIMA model is $(1 - B)^2 Y_t = (1 -$

$\theta_1B - \theta_2B^2)e_t$. This is an ARIMA(0,2,2) or just IMA(2,2) because the second difference is set equal to a moving average of order 2. The process is known as *linear exponential smoothing* or *Holt's method*. This involves two smoothing weights, ω and γ . Details of the relationship between linear smoothing and the associated ARIMA model require some algebra and are shown in the **section 6.6.1**.

The simple exponential smoothing operator applies a differencing operator $(1 - B)$ to Y_t and a moving average operator $(1 - \theta B)$ to e_t . The special case $(1 - B)^2 Y_t = (1 - \theta B)^2 e_t$ uses this operation twice. The smoothed values have been smoothed again using the same weight $\omega = 1 - \theta$ both times. This special case is referred to as *double exponential smoothing*.

6.2.2 Properties of the Forecasts

Assume there is not a $(1 - B)$ factor in $(1 - \theta_1B - \theta_2B^2)$. That is, assume that $(1 - \theta_1B - \theta_2B^2)$ cannot be expressed as $(1 + \theta_2B)(1 - B)$. A factor $(1 - B)$ would cancel out one of the differencing operators, $(1 - B)$, on the left.

Any model of the form $(1 - B)^2 Y_t = (1 - \theta_1B - \theta_2B^2)e_t$ results in a linear forecast. If the last observation is at time n , then the one-step-ahead forecast is $\hat{Y}_{n+1} = Y_n + (Y_n - Y_{n-1}) - \theta_1e_n - \theta_2e_{n-1}$, so that $\hat{Y}_{n+1} - Y_n = (Y_n - Y_{n-1}) - \theta_1e_n - \theta_2e_{n-1}$. The e terms could be replaced by residuals in practice. Do the forecasts increase linearly from here? Moving to a two-step-ahead forecast, you see that $\hat{Y}_{n+2} = \hat{Y}_{n+1} + (\hat{Y}_{n+1} - Y_n) - \theta_2e_n = \hat{Y}_{n+1} + (Y_n - Y_{n-1}) - (\theta_1 + \theta_2)e_n - \theta_2e_{n-1}$, which is an increase of $\beta = (Y_n - Y_{n-1}) - (\theta_1 + \theta_2)e_n - \theta_2e_{n-1}$ over the one-step-ahead forecast. Now it is a question of whether the subsequent changes are this same number β . Because the moving average part is of order 2, forecasts beyond two steps ahead do not involve any e values that have been observed or estimated with residuals, so $\hat{Y}_{n+L} = \hat{Y}_{n+L-1} + (\hat{Y}_{n+L-1} - \hat{Y}_{n+L-2})$ for $L > 2$. For example, if $L = 3$, the forecast is $\hat{Y}_{n+3} = \hat{Y}_{n+2} + (\hat{Y}_{n+2} - \hat{Y}_{n+1}) = \hat{Y}_{n+2} + \beta$, so $\hat{Y}_{n+3} - \hat{Y}_{n+2} = \beta$, another increase of β . Each subsequent forecast is now its predecessor plus the previous change in predictions, which will continue to be β . The forecast is linear.

The error correction formulation for the models being currently considered, the two with linear forecasts, consists of two equations, one for the level L and one for the trend T :

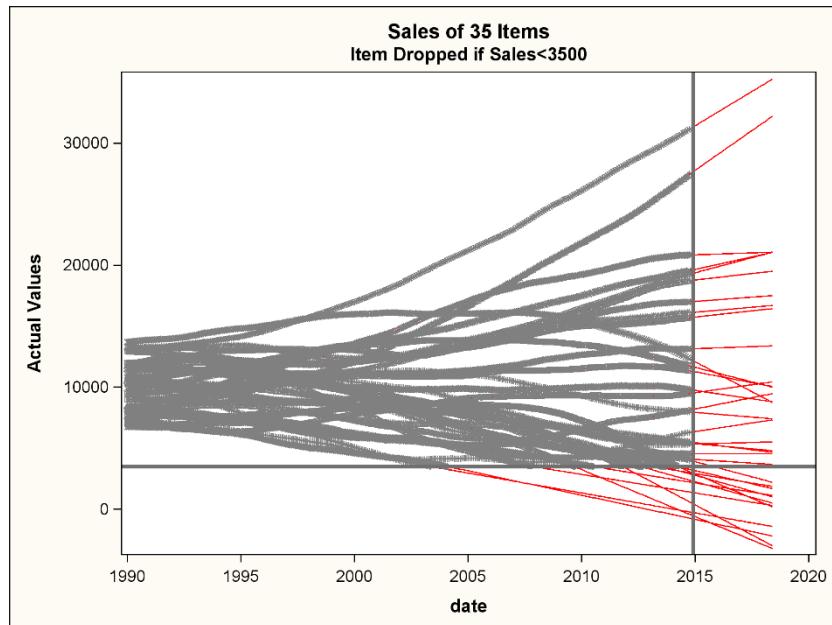
$$\begin{aligned} L_t &= L_{t-1} + T_{t-1} + \omega e_t \\ T_t &= T_{t-1} + \omega \gamma e_t \end{aligned}$$

Setting the trend smoothing parameter γ equal to ω produces the special case called *double exponential smoothing*. The choice is a matter of whether extra flexibility is desired by adding a second parameter γ to the forecasting equations versus just reusing ω .

6.2.3 A Generated Multi-Series Example

PROC ESM enables the smoothing of many series variables at once. Consider an artificial (generated data) example of a retail store chain analyzing sales of 35 items in its lineup of products. **Output 6.7** shows the graph of the monthly number of units sold for each item and some linear forecasts. The historical data cover January 1990 through December 2014. A horizontal line appears at sales of 3500 units. The company's policy is to drop a product from its lineup if monthly sales fall below 3500. This has happened for some items. These series still appear in the data set with missing values after the item was dropped. Including the missing values, there are the same number of records for each series. A request for an L -step-ahead forecast gives $m + L$ forecasts for a series with m missing values at the end, the first m coming from forecasting through the m missing values. That is why the forecasts for all series extend equally far to the right in the plot.

```
proc esm data=sales outest=betas outfor=for lead=42;
  forecast item1-item35 / method=linear;
  id date interval=month;
run;
proc sgplot data=for noautolegend;
  series y=predict x=date / lineatrs=(color=red thickness=1 pattern=solid)
    group=_name_;
  scatter y=actual x=date / markeratrs=(color=gray size=4) group=_name_;
  refline 3500 / axis=y lineatrs=(thickness=3);
  refline "01dec2014"d / axis=x lineatrs=(thickness=3);
  title "Sales of 35 Items";
  title2 "Item Dropped if Sales<3500";
run;
```

Output 6.7: Forecasting Sales by Product

The 300 historical data points lie on and to the left of the vertical reference line, with nonmissing values plotted above the horizontal reference line. By default, the nonmissing data points are plotted as circles. With so many having size set at 4 (SIZE=4), they are not easily distinguished from each other. Forecasts are produced throughout the historical data and into the future, but the forecasts are so close to the observed values that the one-step-ahead historical forecasts are hidden by the data points. Lines to the right of the vertical reference line at the end of 2014 or below the horizontal reference line are forecasts.

Note the relative smoothness of the data. This smoothness is characteristic of time series with double unit roots. There are a few series that are predicted to fall below the boundary in the 42-month forecast horizon, although they are currently above the 3500-unit boundary. Perhaps those should be highlighted for company executives. Perhaps those with projected sales above 20,000, for example, should be highlighted. The variable _NAME_ has values ITEM1, ITEM11, ITEM2, and so on, where these three example names are in ascending order. They are character, so ITEM11 appears before ITEM2. To keep the original ordering, the NOTSORTED option is invoked in the BY statement. It is the user's responsibility to make sure that all members of a group appear contiguously. If occurrences of ITEM2 were scattered through the data, each new occurrence would be considered a new group when the NOTSORTED option is in effect.

```

data flag;
  set for;
  by notsorted _name_;
  length status $ 22;
  if last._name_ then do;
    if predict < 3500 then status = "consider for drop";
    if predict > 20000 then status = "predicted sales leader";
    if predict < 3500 or predict > 20000 then output;
  end;

proc sort data=flag;
  by descending predict;
run;

proc print data=flag;
  var status _name_ date predict ;
run;

```

Output 6.8: Selecting Products of Interest

Obs	STATUS	_NAME_	DATE	PREDICT
1	PREDICTED SALES LEADER	ITEM27	JUN2018	35246.08
2	PREDICTED SALES LEADER	ITEM32	JUN2018	32203.72
3	PREDICTED SALES LEADER	ITEM35	JUN2018	21062.91
4	PREDICTED SALES LEADER	ITEM6	JUN2018	21057.22
5	PREDICTED SALES LEADER	ITEM3	JUN2018	21029.53
6	CONSIDER FOR DROP	ITEM20	JUN2018	2199.83
7	CONSIDER FOR DROP	ITEM13	JUN2018	1856.37
8	CONSIDER FOR DROP	ITEM9	JUN2018	1683.19
9	CONSIDER FOR DROP	ITEM14	JUN2018	1122.75
10	CONSIDER FOR DROP	ITEM4	JUN2018	1002.70
11	CONSIDER FOR DROP	ITEM33	JUN2018	490.87
12	CONSIDER FOR DROP	ITEM15	JUN2018	297.53
13	CONSIDER FOR DROP	ITEM19	JUN2018	173.21
14	CONSIDER FOR DROP	ITEM2	JUN2018	-1420.42
15	CONSIDER FOR DROP	ITEM21	JUN2018	-2199.18
16	CONSIDER FOR DROP	ITEM26	JUN2018	-3006.51
17	CONSIDER FOR DROP	ITEM1	JUN2018	-3234.49

Based on **Output 6.8**, several items are headed toward drop status, some should be promoted, and some should be phased out. Items 27 and 32 are clearly leaders and could be studied to see why they are so popular. Perhaps their price is too low and potential profit is being missed.

Note the nature of these data. Each series is a new variable. Another data format that can be handled by PROC ESM is one in which the responses for all series appear in one column. A second variable, to be used in a BY statement, contains an identifier to label the different series. For example, if the sales data contained a variable SALES that had observations for all the items and a variable ITEM that had the item labels, then PROC ESM would have been run with a BY ITEM command.

6.2.4 Real Data Examples

From the World Bank website (2017), you can download series. Here about 1300 time series for the United States were chosen. Of these, 826 were deemed by PROC ESM to have enough observations to do forecasting. Of these 826, all but 180 had one or both weights on the boundary when doing general linear smoothing. Hitting the boundary rather than converging is a common phenomenon in this collection of time series. As in the tides data, slow non-local movements such as the cyclical nature of tides cannot be exactly accommodated with exponential smoothing models. In this case, the best approximation from the models that fall within the exponential smoothing class might be one that uses each observation as the forecast of its successor, an exponential smoothing model with parameters exactly on the boundary. The software-imposed boundary constraints do not allow exact boundary values.

One of the interesting series is the number of telephone lines per 100 people, yearly data from 1960 through 2013. It provides a nice illustration of the local nature of the level and trend. A macro that could be used for all of the World Bank series is:

```
%macro wldbnk(dsn, var, band=1);
  * band=0 suppresses default prediction band;
  proc esm data=&dsn outest=betas outfor=for lead=10;
    forecast &var / method =linear;
    id date interval =year;
  run;

  data _null_; set betas;
  if _parm_="level" then call symput("level_wt", strip(round(_est_, 0.0001)));
  if _parm_="trend" then call symput("trend_wt", strip(round(_est_, 0.0001)));

  proc print data=betas;
  run;
```

```

proc sgplot data=for;
%if &band %then %do;
  band upper=upper lower=lower x=date;
%end;
  scatter x=date y=actual;
  series x=date y=predict;
  title2 "LEVEL WEIGHT  &LEVEL_WT TREND WEIGHT  &TREND_WT";
run;

%mend;

```

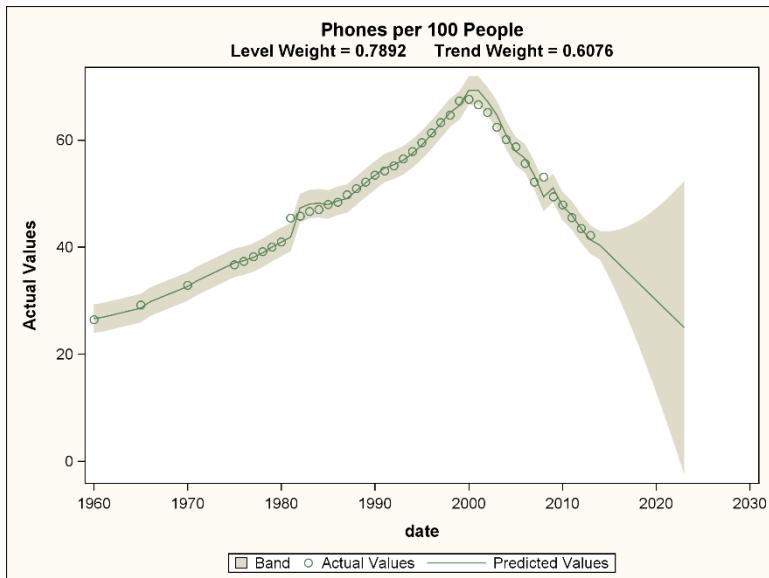
The first two positional parameters specify the data set and variable. The non-positional parameter BAND=1 sets the default behavior to include a prediction band, which would be omitted if BAND were changed to 0. Using this macro for the phone lines data and taking the default non-positional parameter value can be done:

```

data phones_100;
  input lines @@;
  retain date "01jan1959"d; date=intnx("year",date,1);
  format date year.;
  title "phones per 100 people";
  datalines;
  26.4372717 . . . . 29.19030177 . . . . 32.89273314 . . . . 36.69128488
  (more data)
  42.22487039
;
%wldbnk(phones_100, lines);

```

Output 6.9: Phone Lines per 100 People



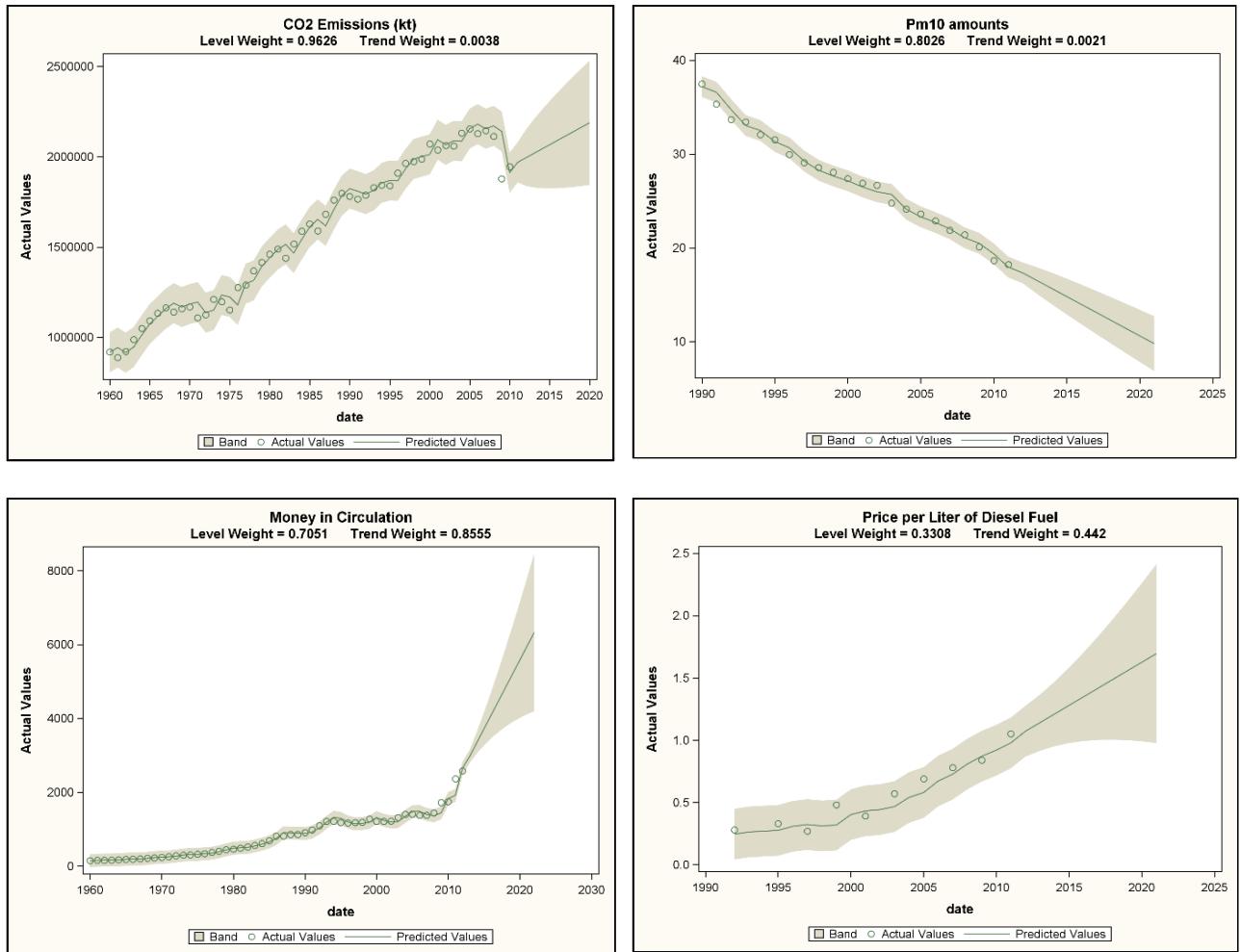
NAME	_TRANSFORM_	_MODEL_	_PARM_	_EST_	_STDERR_	_TVALUE_	_PVALUE_
LINES	NONE	LINEAR	LEVEL	0.78915	0.10908	7.23458	.000000009
LINES	NONE	LINEAR	TREND	0.60761	0.17668	3.43904	.001377721

The early part of the data contains several missing values (.), but the procedure has no trouble handling them. The forecasts plotted in **Output 6.9** extend nicely from the last part of the data, illustrating the local nature of the level and slope. It takes a period or two for the model to react to the jumps in phone line numbers in 1981 and 2008, as well as the turning point around 2000, but it soon gets back on target. Both of the smoothing weights are nearer to 1 than to 0, indicating a smoother that is fairly responsive to changes. It gives rapid, though not immediate, responses to changes in the series level and slope. Error bands widen quickly as forecast lead time increases. The flexibility of this double unit

root model that allows it to track the historical data well is accompanied, as usual, by large uncertainty in the forecast path.

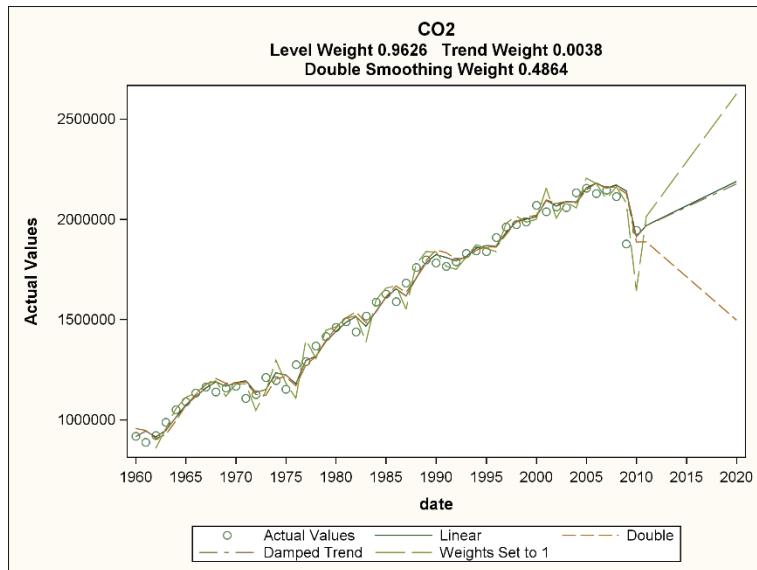
Four more examples, using the World Bank data and macro, are shown in **Output 6.10**.

Output 6.10: Four US Time Series



On the top left is a plot of CO₂ emissions from solid fuels. The level and trend weights are 0.963 and 0.004. The model is very responsive to changes in level, but it uses a consistent trend (slope) estimate through the course of the series. This enables a quick level drop toward the end of the series, while keeping the slope fairly constant. On the upper right is PM10, a measure of particulate matter of a certain size in the air. The level and trend weights are 0.803 and 0.002. The slope is only slightly more flexible than the slope for CO₂. On the bottom left is a measure of the amount of money in circulation. The level and trend weights are 0.705 and 0.856. This model is very responsive to both level and slope changes. This is helpful if the last two observations are truly the start of a new trend. If they are outliers, the forecasts must be used cautiously, if at all. On the bottom right is the price per liter at the pump for diesel fuel. The observations start in 1990 and are taken in alternate years, so it is an example of a shorter time series. The level and trend weights are 0.331 and 0.442. The response to changes in slope and level are moderate, near neither the 0 or 1 extreme. The level and slope changes are accommodating what appears to be a somewhat convex up shape in the data.

The two weights for CO₂ emissions are different, one being fairly close to 0 and the other to 1. To illustrate the effect of weights, two more forecasts are added to the CO₂ plot in **Output 6.11**. One is the double exponential smoothing forecast in which both weights are forced to be the same. The other is the result of setting the moving average θ parameters to 0. The relationships of the parameters in the ARIMA-equivalent model for linear smoothing, $(1 - B)^2 Y_t = (1 - \theta_1 B - \theta_2 B^2) e_t$, to the smoothing weights ω and γ are $\theta_1 = (2 - \omega - \omega\gamma)$ and $\theta_2 = (\omega - 1)$. Setting the moving average parameters to 0 is the same as setting both smoothing weights ω and γ to 1. The forecast becomes $\tilde{Y}_{t+1} = Y_t + (Y_t - Y_{t-1})$.

Output 6.11: Three Forecasting Methods for the CO₂ data

The last value in the plot is 67850.5 units above its predecessor, causing the forecast one period into the future, $\hat{Y}_{t+1} = Y_t + (Y_t - Y_{t-1})$, to be 67850.5 units higher than the last observation. That difference is then added to the one-step-ahead forecast to get the two-step-ahead forecast, and so on (in the linear smoothing case with both smoothing parameters set to 1). The result of this is that the forecast follows the extension of the line connecting the last two observations as can be seen by looking at the highest of the three forecasts. The middle forecast is from the linear method fit with estimated smoothing parameters. It has a more conservative slope, as it is responding to the one-step-ahead changes in all of the past data, not just the last change. If the smoothers for both level and trend are forced to be the same, this common value is 0.4864 and produces the forecast with negative slope. The level weight is less responsive to incoming data than in the linear smoothing case. As a result, the last level estimate is not as close to the last value. The trend estimate is more responsive to incoming data than in the linear smoothing case. Thus, it gives more weight to the relatively large drop near the end of the series, so much so that the slope of the forecast becomes negative. In contrast, the linear method with trend weight 0.0038 is giving much more weight to distant past one-step changes, most of which are positive. One more model, the damped trend model of [section 6.2.6](#), was fit. Its forecasts are visually almost indistinguishable over the forecast horizon from the linear trend model. This results from the damping factor being estimated at 0.999, suggesting almost no damping over the forecast horizon and explaining why forecasts are so close to those of the linear smoothing model.

There are three main points. First, the different smoothing methods are somewhat restricted in terms of what behaviors they can exhibit. Second, an analyst should look at the series and forecasts together in a plot if possible. Third, the smoothing weights can make a big difference, especially if some unusual behavior occurs near the end of the series.

6.2.5 Boundary Values in Linear Exponential Smoothing

The exponential smoothing methods so far have equivalent ARIMA models. With the right settings of their parameters, they will reproduce the exponential smoothing forecasts. For linear exponential smoothing, the ARIMA model is $(1 - B)^2 Y_t = (1 - \theta_1 B - \theta_2 B^2) e_t$, where the moving average parameters are related to the smoothing weights ω and γ by $\theta_1 = (2 - \omega - \omega\gamma)$ and $\theta_2 = (\omega - 1)$. The smoothing parameters are held between 0.001 and 0.999 by the software and end up at those boundary values sometimes in practice, as in [Output 6.2](#) and [Output 6.4](#).

The class of ARIMA models is broad enough that the models can be thought of as universal approximators. But, the ARIMA equivalents to exponential smoothing are a restricted subclass. In particular, none of those presented so far have intercepts. It is possible that the best model for a set of data is not in this restricted class, which, in turn, could result in the best approximation within the class having weights near the 0 or 1 boundary. In this section, a few cases suggest that when estimates are near these boundary values, one possible response is to try a model outside the class of exponential smoothing models.

Occasionally, users will difference data whenever a trend is visible. This is not a good practice. Suppose $Y_t = \alpha + \beta t + e_t$. This is a simple linear trend model with white noise errors that would be best modeled using a regression on time. Differencing reduces the linear trend to $\alpha + \beta t - (\alpha + \beta(t-1)) = \beta$, which is a constant. A second difference reduces it to $\beta - \beta = 0$. The second difference has 0 intercept, thus reproducing one feature of the exponential smoothing models. But,

the model has become $(1 - B)^2 Y_t = (1 - B)^2 e_t$, so $\theta_1 = 2$ and $\theta_2 = -1$. Recalling that $\theta_1 = (2 - \omega - \omega\gamma)$ and $\theta_2 = (\omega - 1)$, these parameters can be achieved by setting $\omega = 0$. This suggests that when the estimated level smoothing parameter ω is near 0 (for example, when it hits the software-imposed lower boundary), one possible cause is that the analyst over-differenced the data. The particular model analyzed here should not be differenced at all. It should just be analyzed by regression on time.

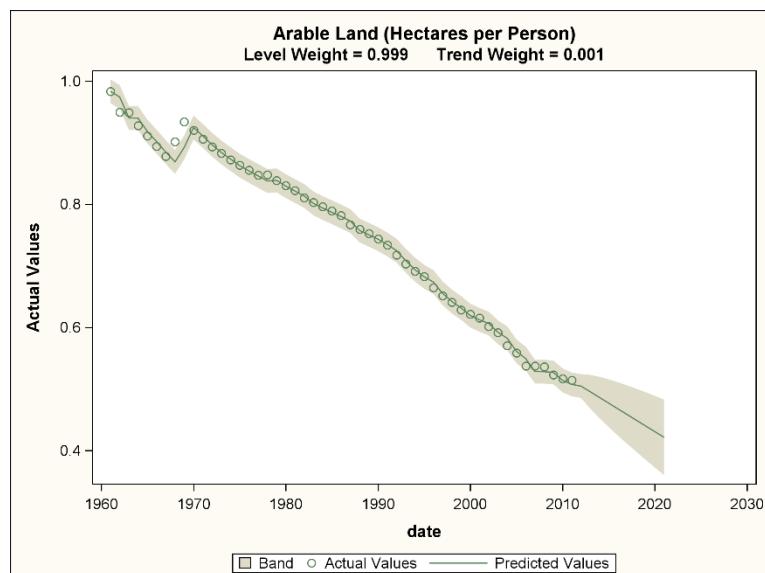
Suppose $Y_t = \beta + Y_{t-1} + e_t$, a random walk with drift β . It is best estimated by fitting an intercept (the drift) to the differences. That is, the model is $Y_t - Y_{t-1} = \beta + e_t$. Only one difference is really needed, not two. However, because it has an intercept, this model does not fall into the class of ARIMA models that are equivalent to exponential smoothing models. The intercept problem is easily solved by differencing the differences, getting $Y_t - 2Y_{t-1} + Y_{t-2} = 0 + e_t - 1e_{t-1}$ so that $\theta_1 = 1$ and $\theta_2 = 0$. These parameters result whenever the smoothing parameters are $\omega = 1$ and $\gamma = 0$. One possible cause of a level smoothing weight being near 1 and the corresponding trend smoothing weight being near 0 is that the data are from a random walk with drift.

Suppose both smoothing parameters are at the limit value 1, so that the moving average parameters $\theta_1 = (2 - \omega - \omega\gamma)$ and $\theta_2 = (\omega - 1)$ are both 0. The ARIMA model becomes $Y_t = Y_{t-1} + (Y_{t-1} - Y_{t-2}) + e_t$. A one-step-ahead forecast from time n is $\hat{Y}_{n+1} = Y_n + (Y_n - Y_{n-1})$. The forecast is just the current observation plus the most recently observed change. The level and slope of the forecast depend on only the last two values. If the series is not especially smooth, the forecasts are quite unstable. This was seen in the highest CO₂ forecast in **Output 6.11**. The two-step-ahead forecast $\hat{Y}_{n+2} = \hat{Y}_{n+1} + (\hat{Y}_{n+1} - Y_n)$ has the one-step-ahead forecast as its level. The new slope $(\hat{Y}_{n+1} - Y_n) = (Y_n - Y_{n-1})$ is the same as the old slope. In fact, the slope (trend) is $(Y_n - Y_{n-1})$ no matter how far ahead you are forecasting. Each subsequent forecast is its predecessor plus this slope. Simply put, the forecast runs along the extension of the line, connecting the last two observations. This could be quite unstable.

The World Bank data collection contains a series of arable land measurements in hectares per person in the United States over 51 years. Use the %WLDBNK macro to invoke linear exponential smoothing on the data. **Output 6.12** shows the results.

```
data land;
label land="ARABLE LAND (HECTARES PER PERSON)";
input land @@;
retain date "01jan1960"d;
date=intnx("year",date,1);
format date year.;
datalines;
0.983336146      0.949377607 0.948911975 0.927442428 0.910948364 0.893900081
      (more data)
0.516713266 0.514029084
;
%wldbnk(land,land);
```

Output 6.12: Arable Land in the United States (Hectares per Person)



<u>_NAME_</u>	<u>TRANSFORM_</u>	<u>MODEL_</u>	<u>PARM_</u>	<u>EST_</u>	<u>STDERR_</u>	<u>TVALUE_</u>	<u>PVALUE_</u>
LAND	NONE	LINEAR	LEVEL	0.999	0.097685	10.2268	0.000000
LAND	NONE	LINEAR	TREND	0.001	0.046650	0.0214	0.98298

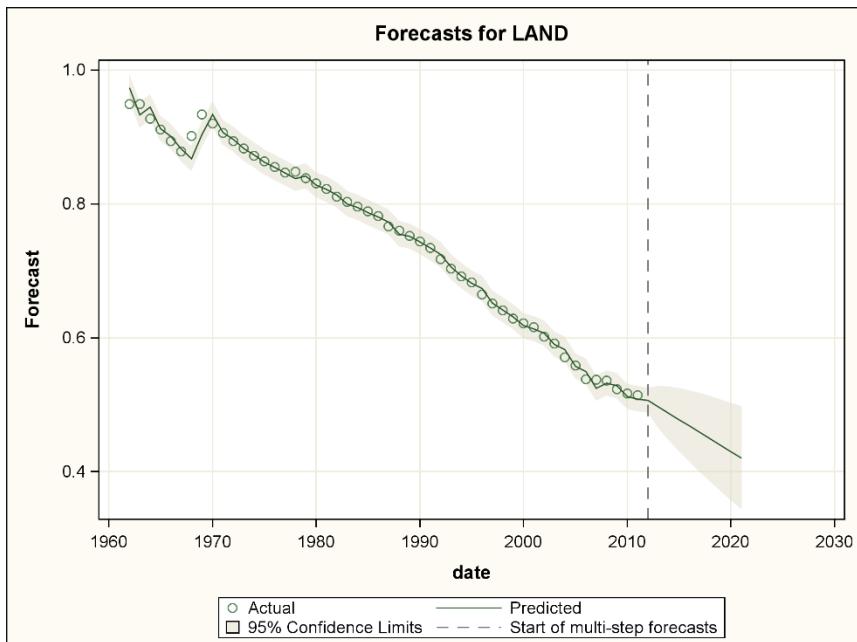
The level smoothing weight has hit the boundary on the $\omega = 1$ side. The trend weight has hit the boundary on the $\gamma = 0$ side. Despite this, the forecast looks reasonable. Without convergence, however, neither the forecast nor the width of the prediction intervals is justified by statistical theory. The suggested alternative was a single difference with an intercept. A moving average order 1 term is included to modify the effect of the single difference. In other words, the model is a single exponential smoothing model modified with an intercept term. Because models with intercepts are not in the class of exponential smoothing models, the fitting is done in PROC ARIMA.

```
proc arima data=land plots=forecast(forecast);
  identify var=land(1);
  estimate q=1 method=ml;
  forecast lead=10 id=date interval=year;
run;
```

Output 6.13: PROC ARIMA Analysis of Arable Land Data

Maximum Likelihood Estimation					
Parameter	Estimate	Standard Error	t Value	Approx Pr > t	Lag
MU	-0.0095340	0.0017846	-5.34	<.0001	0
MA1,1	-0.32204	0.13925	-2.31	0.0207	1

Autocorrelation Check of Residuals									
To Lag	Chi-Square	DF	Pr > ChiSq	Autocorrelations					
6	1.73	5	0.8855	-0.071	-0.040	-0.070	-0.133	0.030	-0.027
12	5.41	11	0.9099	-0.183	0.071	0.050	0.117	-0.056	0.037
18	8.05	17	0.9654	-0.027	-0.073	0.047	-0.086	0.092	-0.099
24	10.10	23	0.9907	-0.040	0.048	-0.027	0.037	-0.021	-0.122



When differencing is used with an intercept, the intercept is estimating the average period to period change in the series, which is the slope in a linearly trending case. The intercept here is -0.009534 and is highly statistically significant. It represents the average year-to-year change in the series. Even though the ID variable DATE is measured in days, the differences are year-to-year differences, so it is a loss in land per person per year. This loss could come from conversion of arable land to other uses and from an increase in population. Over the 10-year forecast period, the drop is $10(-0.009534) = -0.09534$ hectares per person, which is consistent with the graph.

The two graphs are qualitatively similar, although the ARIMA model suggests slightly wider prediction intervals: $(0.34345, 0.49742)$ versus $(0.36022, 0.48271)$ for ESM at lead 10. The moving average parameter estimate is negative, a value that is disallowed in exponential smoothing. It would correspond to a smoothing weight 1.322, a value exceeding 1. Both parameters fit to the data are significant and there is no sign of residual correlation. That is, there is no need to modify the model. Furthermore, convergence was achieved. This model is justified by statistical theory and cannot be reproduced by exponential smoothing methods because of its intercept and negative parameter estimate. Hitting the boundary does not necessarily mean that a terrible forecast is obtained, but it does mean that the results are not justified by statistical theory.

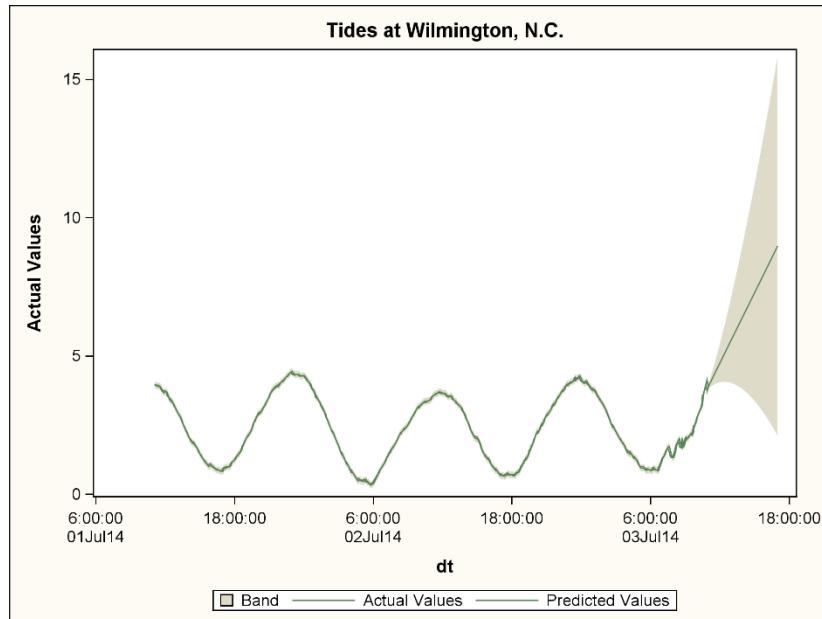
Returning to the tides data, a linear exponential smoothing model gives an even worse forecast far into the future for the original tides data than the single exponential smoothing model. However, the smoothing parameters are well within the allowable range in **Output 6.14**.

```
proc esm data = tides outfor=esm_fore outest=betas lead=60;
  id dt interval=minute6;
  forecast water_level / method=linear;

proc print data=betas noobs;
run;

proc sgplot data=esm_fore;
  band x=dt upper=upper lower=lower;
  series x=dt y=actual;
  series x=dt y=predict;
run;
```

Output 6.14: Linear Exponential Smoothing Forecast for Tides Data



<u>_NAME_</u>	<u>_TRANSFORM_</u>	<u>_MODEL_</u>	<u>_PARM_</u>	<u>_EST_</u>	<u>_STDERR_</u>	<u>_TVALUE_</u>	<u>_PVALUE_</u>
WATER_LEVEL	NONE	LINEAR	LEVEL	0.81700	0.031941	25.5787	1.4786E-91
WATER_LEVEL	NONE	LINEAR	TREND	0.23197	0.026455	8.7682	3.1996E-17

Thus far, no exponential smoothing model has given an acceptable forecast for the tides. Only the smoothing of differences from the NOAA forecasts has been acceptable. Before leaving this example, an ARIMA approach that works well is shown in **Output 6.15**. All parameters are statistically significant. There is no evidence of correlation in the table labeled “Autocorrelation Check of the Residuals.”

```

proc arima data=tides plot=(forecast(forecast));
  where dt <"03jul14:00:06"dt;
  identify var= water_level nlags=18;
  estimate p=(1 2 4 5 6) q=(4 11) method=ml maxiter=500;
  forecast lead = 120 id=dt interval = minute6 out=outarima;
run;

data all;
  merge outarima tides(keep = water_level dt);
  by dt;
run;

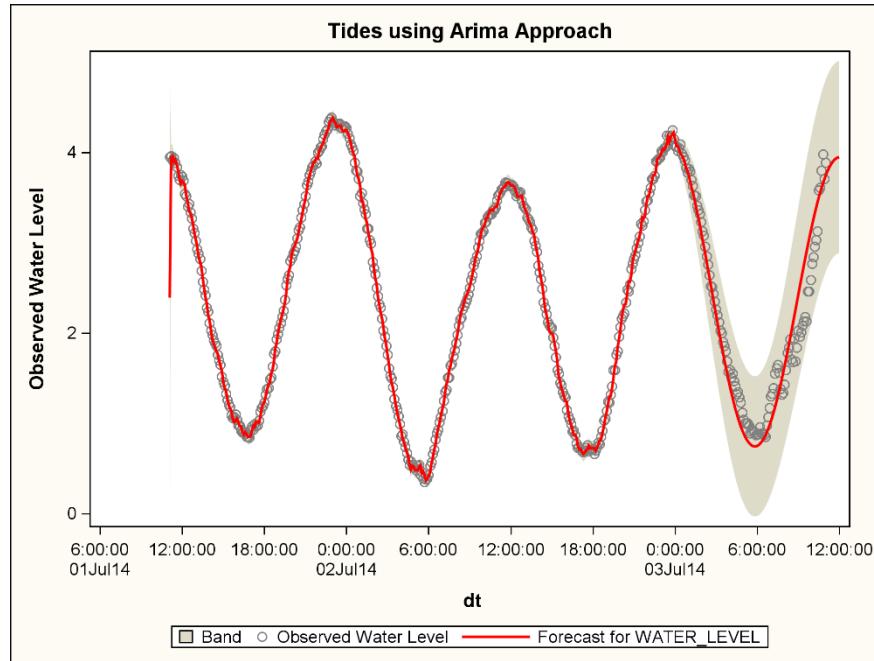
proc sgplot data=all;
  title "TIDES USING ARIMA APPROACH";
  band upper = u95 lower=l95 x=dt;
  scatter y=water_level x=dt / markerattrs=(color=gray);
  series y=forecast x=dt / lineattrs=(color=red thickness=2);
run;

```

Output 6.15: ARIMA Approach to the Tides Data

Maximum Likelihood Estimation					
Parameter	Estimate	Standard Error	t Value	Approx Pr > t	Lag
MU	2.39157	0.07279	32.85	<.0001	0
MA1,1	0.69896	0.07258	9.63	<.0001	4
MA1,2	-0.09047	0.03970	-2.28	0.0227	11
AR1,1	0.81157	0.04421	18.36	<.0001	1
AR1,2	0.22077	0.05409	4.08	<.0001	2
AR1,3	0.81811	0.07323	11.17	<.0001	4
AR1,4	-0.62589	0.07615	-8.22	<.0001	5
AR1,5	-0.23657	0.04740	-4.99	<.0001	6

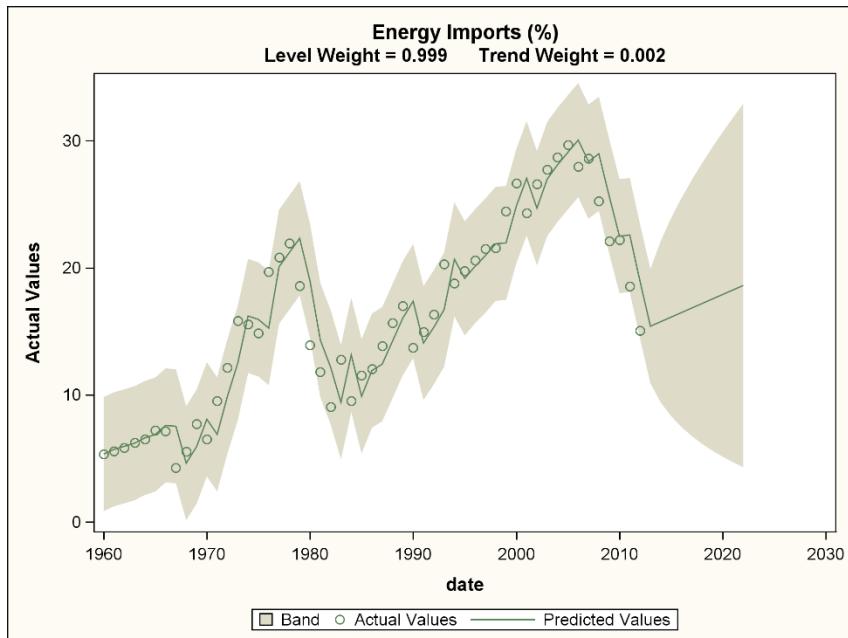
Autocorrelation Check of Residuals									
To Lag	Chi-Square	DF	Pr > ChiSq	Autocorrelations					
6	.	0	.	-0.005	-0.026	0.006	0.025	-0.056	-0.066
12	4.75	5	0.4474	-0.031	-0.031	-0.009	0.027	0.031	-0.004
18	10.18	11	0.5139	0.008	0.099	-0.008	-0.007	-0.010	0.063
24	13.59	17	0.6956	-0.001	-0.063	-0.033	-0.020	-0.047	0.031
30	17.30	23	0.7944	0.062	-0.016	-0.042	0.008	0.056	0.014
36	21.18	29	0.8526	-0.033	-0.007	0.034	-0.001	0.008	-0.084
42	30.72	35	0.6751	0.027	0.059	0.130	-0.004	-0.019	-0.037
48	38.00	41	0.6048	-0.077	0.013	0.044	-0.022	-0.045	0.081



The fit is excellent with all parameter estimates significant and no sign of correlation in the residuals. The forecast intervals are fairly narrow. Yet, they contain all of the withheld data points. The forecasts seem reasonable in regard to the observed past data. A message stating that no improvement in fit could be obtained indicates that the procedure stopped before the standard convergence criteria were met. The parameter estimates, no matter how they were derived, have reduced the error term to white noise and delivered a reasonable forecast. The model specified was derived by fitting a sequence of long autoregressions, and then trying to improve them with moving average terms by trial and error. It only makes sense that careful use of the entire suite of ARIMA models should give at least as good results as using the subset that comprise the exponential smoothing equivalents.

Although sinusoidal-looking forecasts from an autoregressive model may be surprising, the lag 2 recursion $Y_t = \alpha Y_{t-1} - Y_{t-2}$ will produce an exact sinusoid of frequency ω when $\alpha = 2\cos(\omega)$ and 2 different starting values are given. Adding an error term e_t gives an autoregressive type equation, but it is nonstationary. Its characteristic equation has a complex conjugate pair of roots with magnitude 1. Thus it is possible for an autoregressive model with complex roots of magnitude near 1 to approximate sinusoidal behavior.

A final trend example is the percent of US-consumed energy that is imported. **Output 6.16** shows output from a linear exponential smoother using the same type of PROC ESM code as in the previous examples. The trend smoothing parameter is almost 0, indicating that the trend is determined by all of the data with little weight given to incoming data. Because the data have been increasing in general, a positive slope is seen. The idea that the most recent observation just happens to be a turning point and that the series will immediately start increasing again seems somewhat unlikely. This is especially unlikely in light of the fact that the level smoothing weight has hit the boundary.

Output 6.16: Percent of Energy Use from Imports

<u>_NAME_</u>	<u>_TRANSFORM_</u>	<u>_MODEL_</u>	<u>_PARM_</u>	<u>_EST_</u>	<u>_STDERR_</u>	<u>_TVALUE_</u>	<u>_PVALUE_</u>
IMPORTS	NONE	LINEAR	LEVEL	0.99900	0.10393	9.61270	0.00000
IMPORTS	NONE	LINEAR	TREND	0.00199	0.02750	0.07227	0.94267

6.2.6 Damped Trend Exponential Smoothing

The recent trend in the data shows a decreasing demand for energy imports, and the forecast runs counter to this momentum. This serves to introduce one last trend smoothing model, the *damped trend exponential smoothing model*. The idea is to let the slope move toward 0 as the forecast moves farther into the future. The ARIMA equivalent of this model is the same as the linear trend, except that one of the two difference operators $(1 - B)$ is replaced by an autoregressive operator $(1 - \phi B)$, where $0 < \phi < 1$. The model is $(1 - \phi B)(1 - B)Y_t = (1 - \theta_1 B - \theta_2 B^2)e_t$, and the relationships to the smoothing weights are $\theta_1 = 1 + \phi - \omega - \omega\gamma\phi$ and $\theta_2 = (\omega - 1)\phi$. Substituting 1 for ϕ returns the corresponding equations for linear smoothing. The autoregressive parameter ϕ itself is the trend damping weight in damped trend exponential smoothing.

```

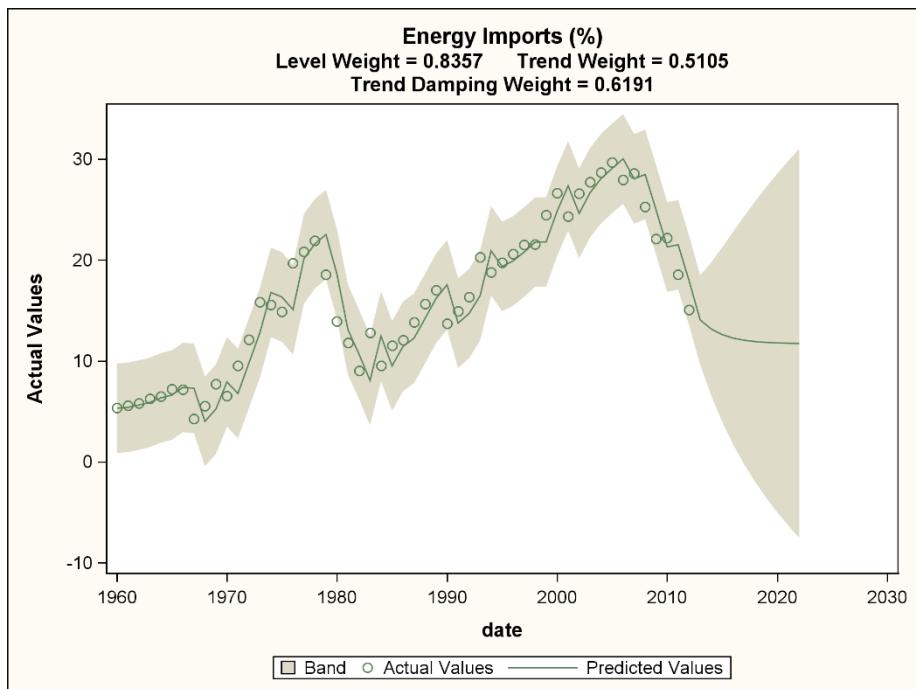
proc esm data=energy outest=betas outfor=for lead=10;
  forecast imports / model = damprend;
  id date interval = year;
run;

data _null_;
  set betas;
  if _parm_="level" then call symput("level_wt", strip(round(_est_, 0.0001)));
  if _parm_="trend" then call symput("trend_wt", strip(round(_est_, 0.0001)));
  if _parm_="damping" then call symput("damping_wt", strip(round(_est_, 0.0001)));

proc print data=betas noobs;
run;

proc sgplot data=for;
  band upper=upper lower=lower x=date;
  scatter x=date y=actual;
  series x=date y=predict;
  title2 "LEVEL WEIGHT = &LEVEL_WT      TREND WEIGHT = &TREND_WT";
  title3 "TREND DAMPING WEIGHT = &DAMPING_WT";
run;

```

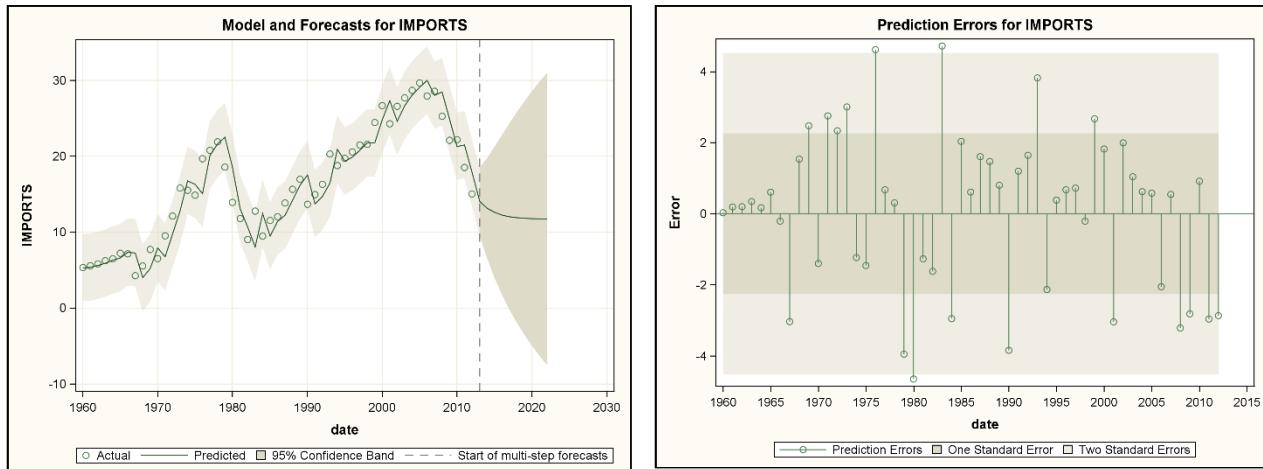
Output 6.17: Damped Trend Model for Energy Imports

<u>_NAME_</u>	<u>_TRANSFORM_</u>	<u>_MODEL_</u>	<u>_PARM_</u>	<u>_EST_</u>	<u>_STDERR_</u>	<u>_TVALUE_</u>	<u>_PVALUE_</u>
IMPORTS	NONE	DAMPTREND	LEVEL	0.83568	0.27264	3.06517	0.00350
IMPORTS	NONE	DAMPTREND	TREND	0.51055	0.92948	0.54929	0.58525
IMPORTS	NONE	DAMPTREND	DAMPING	0.61910	0.38760	1.59727	0.11651

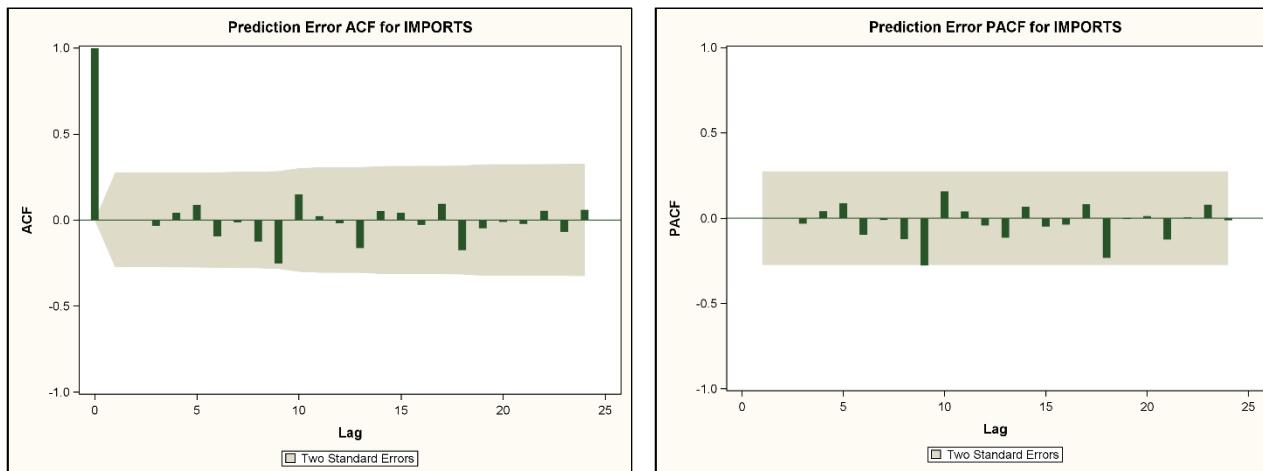
This plot tells a different story. All of the parameter estimates are well within the 0 to 1 region, but they have large standard errors. The forecast suggests a continued decline in dependence on imported energy but at a steadily less decreasing rate. PROC ESM has some features that make checking the models relatively easy.

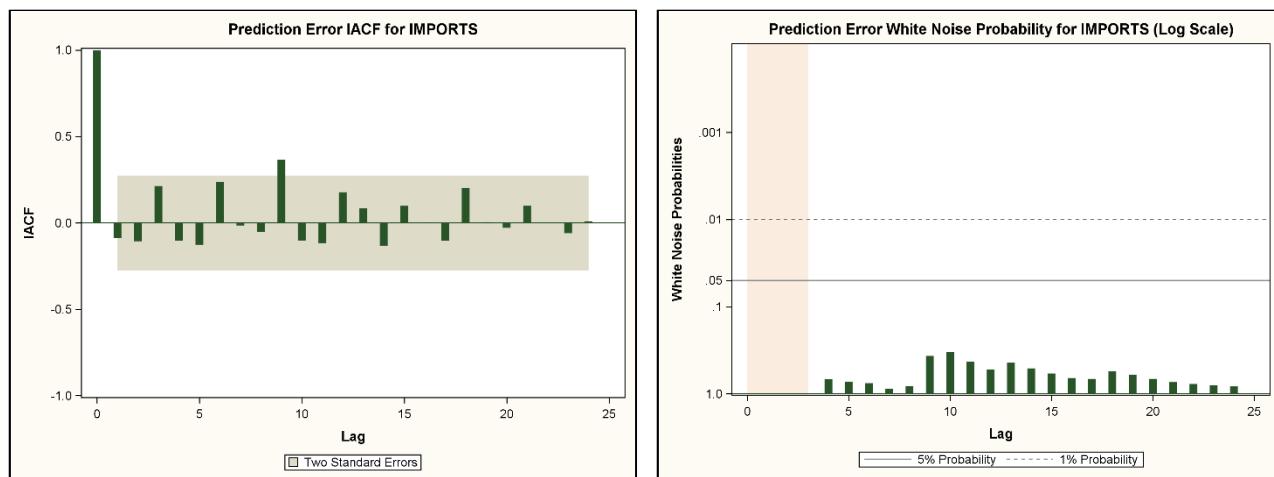
6.2.7 Diagnostic Plots

Adding the PLOTS=ALL option to the PROC ESM statement gives several useful plots for the energy imports data. **Output 6.18** shows the forecast plot in the left panel. It is much like the %WLDBNK macro, but it does not list the parameter estimates in its title. The right panel contains the prediction errors with one and two standard error bands. No outliers or autocorrelation are evident.

Output 6.18: Forecasts and Forecast Errors for Energy Imports

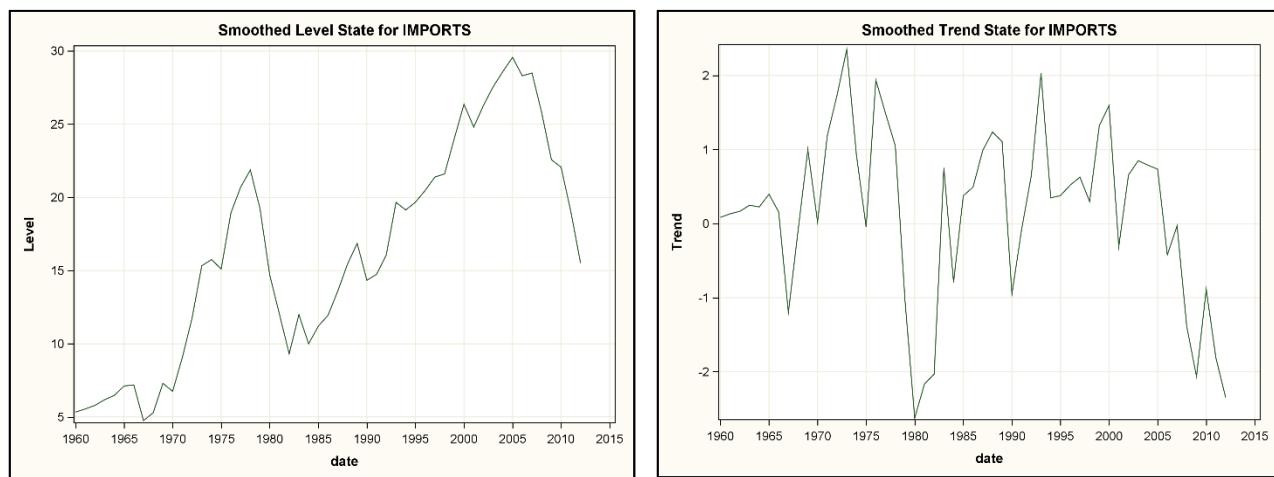
The usual correlation diagnostic panels are shown in **Output 6.19**. They confirm the impressions from the prediction errors plot. The upper left panel is the autocorrelation function. To its right is the partial autocorrelation function. The lower left panel is the inverse autocorrelation function. To its right is the test for white noise. Note the inverted logarithmic scale on the white noise test vertical axis. By plotting in this way, large bars show significant correlation, so small bars are desirable for prediction error data. The 5% and 1% lines indicate that the bars are small enough that the errors can be considered uncorrelated. Turning to the autocorrelation functions, even if there is no true autocorrelation, one in 20 of the bars would be expected to exceed the 5% bands shown in their plots. The plots are consistent with uncorrelated errors.

Output 6.19: Four Diagnostic Plots for the Energy Import Data



The linear and damped trend methods involve a smoothed level L_t and smoothed trend T_t series. **Output 6.20** shows the smoothed level (left) and smoothed trend (right). Because the level smoothing weight 0.84 from **Output 6.17** is close to 1, the smoothed level is responsive to the recent data. The plot looks like the data plot. The local trends have historically mostly been above 0. However, starting in 2005, negative trends have been consistently seen, with the most recent one being about as low as the dramatically low 1980 value. As a result, U.S. dependency on foreign energy has been decreasing at an increasingly sharp rate for the last several observations.

Output 6.20: Smoothed Level and Smoothed Trend



Several other plots are available, such as the periodogram that searches for periodic behavior in the prediction errors. In summary, all of the diagnostics look good for this series and model. The only small cloud on the horizon is that two standard errors in **Output 6.17** are quite large, rendering the trend and damping weights statistically insignificant. When a null hypothesis fails to be rejected (as is the case here), it does not mean that the hypothesis is true. The analyst is not obliged to enforce it. There is no obligation to set these weights to 0, but there is not strong enough evidence to convince a skeptic that they are not zero.

6.2.8 Sums of Forecasts

Analysts occasionally are interested in sums over the forecast period. For the hypothetical sales data in **Output 6.7**, a manager might want to know the forecasted total sales over the forecast period. Adding up the forecasts is an easy task. However, the variances of the forecast errors are different for each lead time. Furthermore, the forecast errors are not independent of each other. The computation of appropriate standard errors for these forecast totals is difficult. PROC ESM can do this computation. Returning to the sales data, you need to insert only OUTSUM=TOTALS to get a data set

(TOTALS, in this case) with the desired information. This code prints out a few columns of three of the rows in the TOTALS data. The variable PREDICT is the desired estimate and comes with upper and lower confidence limits.

```
proc esm data=sales outest=betas outfor=for lead=42 outsum=totals;
  forecast item1-item35/ method=linear;
  id date interval=month;
run;
proc print data=totals;
  title "PARTIAL OUTPUT FROM OUTSUM=TOTALS";
  var _name_ lower predict upper _lead1_ _lead2_ _lead42_;
  where _name_="item1" or _name_="item2" or _name_="item27";
run;
```

Output 6.21: Partial OUTSUM= Data Set

Obs	_NAME_	LOWER	PREDICT	UPPER	_LEAD1_	_LEAD2_	_LEAD42_
1	ITEM1	-184068.53	-80629.81	22808.90	-605.02	-669.16	-3234.49
2	ITEM2	-305732.83	-36221.87	233289.10	-304.43	-331.65	-1420.42
27	ITEM27	1382550.54	1399741.40	1416932.26	31408.27	31501.88	35246.08

The data set contains all 42 requested forecasts for all 35 items, but only the first two forecasts and the last one are requested for just three of the items. The predictions for items 1 and 2 for all 42 periods past the end of the data are negative, as is their sum. The totals are for the same period as the series with full data. These were among the items discontinued early for low sales. The negative forecasts are not of interest. Item 27 is of interest because it has such high and increasing sales with a forecasted total over the 42 periods of just under 1.4 million.

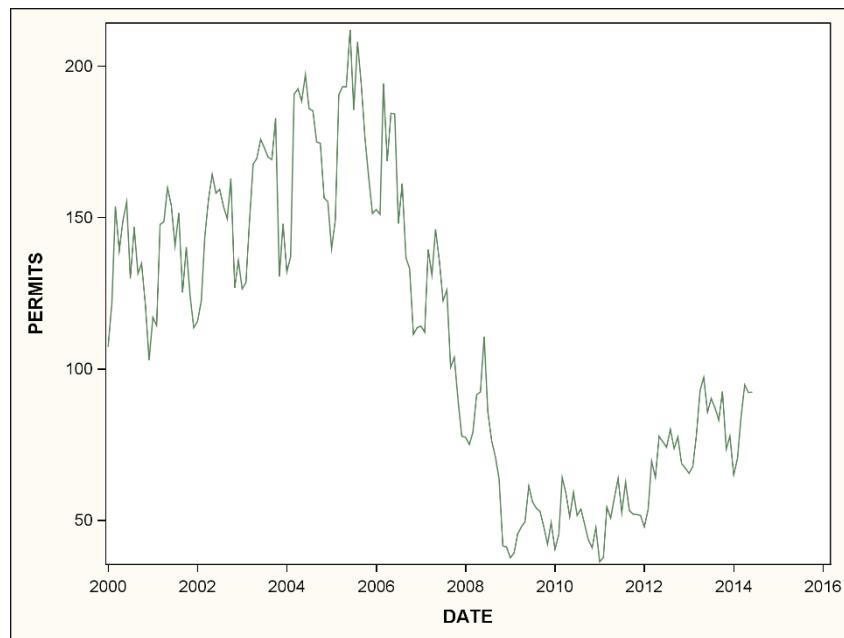
6.3 Smoothing Seasonal Data

An obvious extension of the simple exponential smoothing that was discussed at the beginning of this chapter is the same smoothing applied separately to each month. That is, the sequence of all Januaries is smoothed, as is the sequence of all Februaries, and so on. This provides 12 horizontal forecast lines, one for each month.

6.3.1 Seasonal Exponential Smoothing

PROC ESM refers to the smoothing of each months as *seasonal exponential smoothing*, and it uses the same smoothing weight for each month. It is equivalent to the model $Y_t - Y_{t-12} = e_t - \theta e_{t-12}$ with $0 < \theta < 1$. The seasonal smoothing weight is $\omega = 1 - \theta$. To illustrate, **Output 6.22** shows the number of building permits issued monthly from January 2000 to June 2014. The data are from the US Census Bureau web page (<http://www.census.gov/construction/bps/uspermits.html>). Begin by using the seasonal exponential smoothing model.

```
proc sgplot data=permits;
  series x=date y=permits;
run;
```

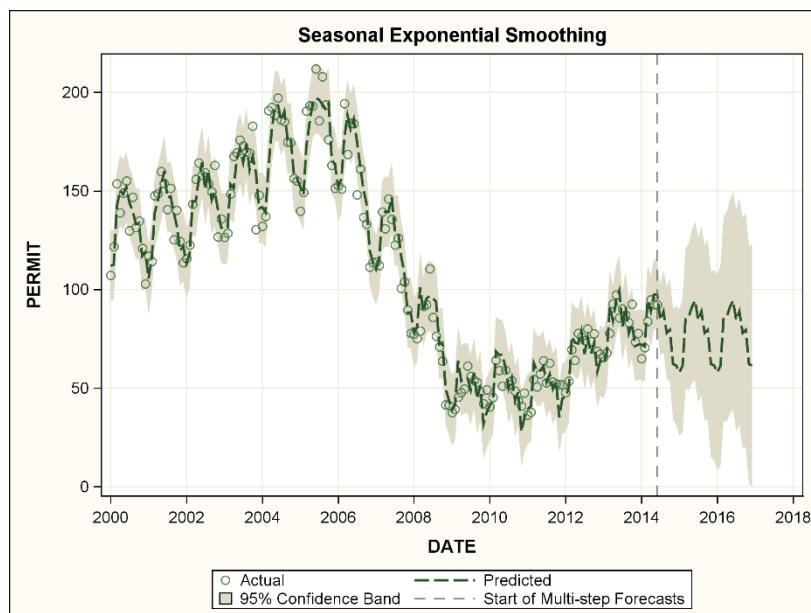
Output 6.22: US Housing Permits Issued (in Thousands)

Despite seeing that a horizontal forecast with seasonality does not seem desirable, you do it anyway to see the simple smoothing model. **Output 6.23** contains the PROC ESM weight estimates and a plot resulting from the PLOT= option.

```
proc esm data=housing outest=betas
  outfor=for lead=24 plot = (forecasts modelforecasts);
  forecast permits / method=seasonal;
  id date interval=month;
run;
proc print data=betas noobs; run;
```

Output 6.23: Smoothing Weights for Seasonal Exponential Smoothing

<u>_NAME_</u>	<u>TRANSFORM_</u>	<u>MODEL_</u>	<u>PARM_</u>	<u>EST_</u>	<u>STDERR_</u>	<u>TVALUE_</u>	<u>PVALUE_</u>
PERMITS	NONE	SEASONAL	LEVEL	0.61185	0.050046	12.2257	0.00000
PERMITS	NONE	SEASONAL	SEASON	0.00100	0.048362	0.0207	0.98353



The seasonal parameter is very small, indicating a small weight for the incoming observation and a relatively large weight to data in the past. That is, the model is incorporating all past seasonal patterns with almost equal weight. The plot shows that the forecast seasonal pattern is not exactly the seasonal pattern of the most recent year. The forecast has a horizontal (or 0) trend not because of the data, but because that is the only choice available to this model. It is clear that there have been historically different local trends that change occasionally.

6.3.2 Winters Method

There is enough of an upward trend toward the end of the data to make a local linear trend appealing. Two seasonal methods, the Winters multiplicative method and the Winters additive method, are available. The idea of the additive method can be understood by imagining a local linear trend line. Add a January effect to every January, a February effect to every February, and so on.

Set up these effects so that they add to 0 over a year. This sounds like, and is like, regression with seasonal dummy variables. Unlike regression, in this case, local rather than global estimates of the trend and seasonal parameter are computed. The degree to which these are local is determined by the smoothing weights. The smaller the weight on the incoming data point, the more the forecast is determined by data from long ago. The larger the weight, the more the forecast behaves like a seasonal random walk that uses only the most recent year to compute the forecast.

Like the additive method, the multiplicative method is understood by imagining a trend line. This trend line is multiplied by a January effect every January, a February effect every February, and so on. These are arranged so that they average to 1. For example, every January might be 1.1 times the overall trend, and every February might be 1.2 times the overall trend. Note that 1.1 times 100 is 110 and 1.1 times 500 is 550. That is, the seasonal variation is more pronounced at high levels than at low levels of the series.

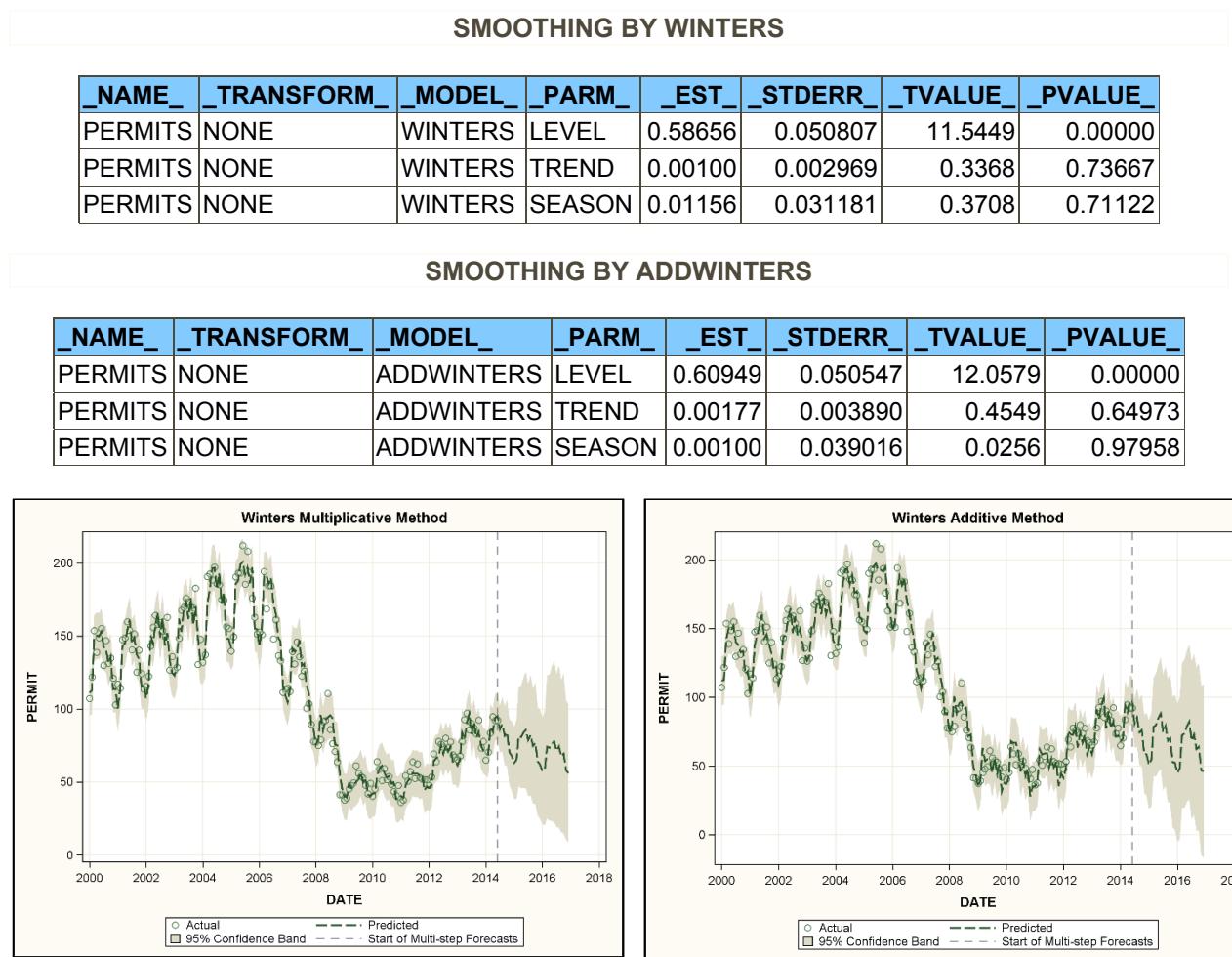
Neither of the two Winters methods is associated with a simple ARIMA equivalent model. However, the smoothing equations help explain how the weights work. For the Winters additive method, the forecasting equation has a level term L , a trend term T , and a seasonal term S with periodicity p . The k -step-ahead forecast from time t is $\hat{Y}_{t+k} = L_t + kT_t + S_{t-p+k}$, where you see that the seasonal subscript, $t - p + k$, indicates the most recent observation or forecast from the same season as time $t + k$. The level smoothing equation is $L_t = \omega(Y_t - S_{t-p}) + (1 - \omega)(L_{t-1} + T_{t-1})$, which is the same as the Holt's method except that a deseasonalized version, $(Y_t - S_{t-p})$, of the current observation is used. The trend smoothing equation is the same as the Holt's method, $T_t = \gamma(L_t - L_{t-1}) + (1 - \gamma)T_{t-1}$, so the level and trend smoothing weights have the same interpretation and implications as they did in the Holt's method.

The current Y would be an estimate of the current level in Holt's method, but for seasonal data, the current observation deviates from the level because of the seasonal effect. The difference between the observation and the current level is one estimate of the appropriate seasonal component. In the usual way, mixing that with the previous estimate of the seasonal component gives the seasonal smoothing equation $S_t = \delta(Y_t - L_t) + (1 - \delta)S_{t-p}$. The Winters method is a logical extension of Holt's method. The larger δ , the more responsive is the smoothed seasonal component to the most recent observation. Smaller values of δ imply stronger influence of past observations. As with the other smoothing methods, hitting the lower bound for the seasonal weight might happen when data have constant, rather than slowly changing, seasonal effects such as those estimated in a regression with seasonal dummy variables. Hitting the upper bound might arise when analyzing data from a seasonal random walk.

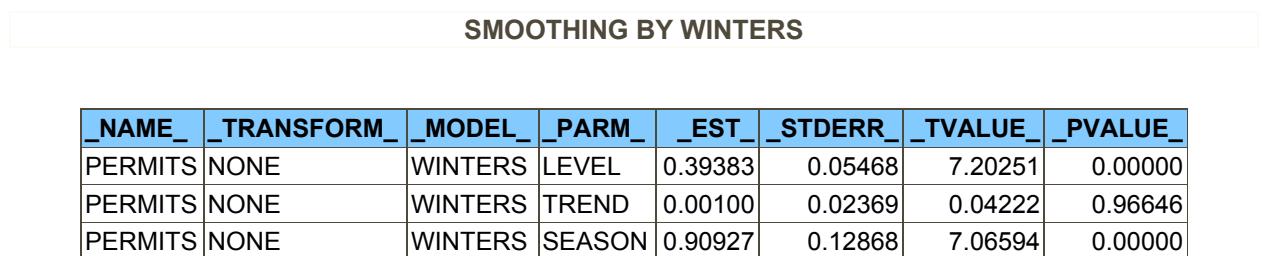
The smoothing equations for the Winters multiplicative method are very similar to the additive method. But, they treat the seasonal component as a multiplier. The ratio of the current observation to the current level replaces the difference between them that was used in the additive method. The resulting smoothing equations are $L_t = \omega(Y_t / S_{t-p}) + (1 - \omega)(L_{t-1} + T_{t-1})$ for the level, $T_t = \gamma(L_t - L_{t-1}) + (1 - \gamma)T_{t-1}$ for the trend, and $S_t = \delta(Y_t / L_t) + (1 - \delta)S_{t-p}$ for the seasonal component. Both methods are applied to the seasonal housing permits data.

The plot in **Output 6.23** seems to have a more pronounced seasonal pattern in the earlier historical data where the levels are higher than in the more recent historical data. If relevant, this feature favors a multiplicative approach. **Output 6.24** displays the parameter estimates and forecast graphs resulting from each of these approaches.

```
proc esm data=permits outest=betas
  outfor=for lead=24 plot = (forecasts modelforecasts);
  forecast permits/method=winters;
  **or use forecast permits/method=addwinters;
  id date interval=month;
run;
```

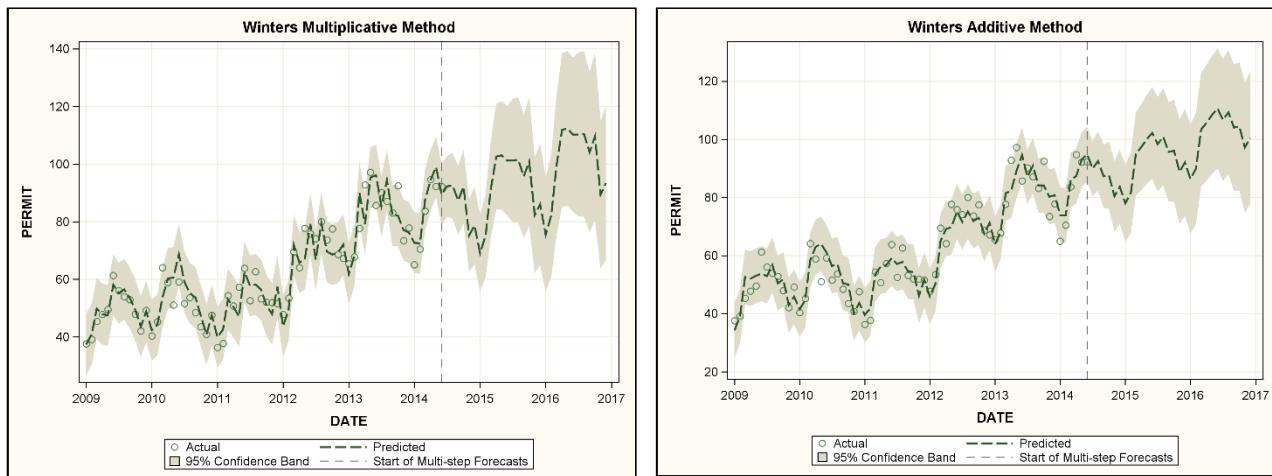
Output 6.24: Winters Multiplicative and Additive Method results

In either case, the level is responsive to local changes as is indicated by the level weight, intermediate between 0 and 1. The forecasts come off the end of the series at the right level. In contrast, the trend (slope) and seasonal weights are very small, indicating that these features are much influenced by the earlier data. Note especially that the strong descent in the middle years of the plot is still having an influence on the nature of the trend. This is so much so that the forecasts imply that the data end just before a turning point in the recovery period. The future will return to a decline in building permits issued. It would be a hard sell to convince someone that the data just so happens to end right at the end of a recovery and will immediately begin to descend. This brings up a question. Is more data always better? Are you willing, for example, to state that data from the downturn in building from year 2006 to 2009 are relevant for forecasting from the middle of 2014 onward? Perhaps conditions have changed. That is, the earlier data are more of a distraction, not relevant for current market conditions, and irrelevant for the task at hand. Analyzing only the data from 2009 onward, a different and more believable forecast is achieved. **Output 6.25** results from applying the code just described to the reduced data set.

Output 6.25: Winters Multiplicative (left) and Winters Additive (right) Methods on Data Starting in 2009

SMOOTHING BY ADDWINTERS

<u>_NAME_</u>	<u>TRANSFORM_</u>	<u>MODEL_</u>	<u>PARM_</u>	<u>EST_</u>	<u>STDERR_</u>	<u>TVALUE_</u>	<u>PVALUE_</u>
PERMITS	NONE	ADDWINTERS	LEVEL	0.40134	0.072828	5.51082	0.00000
PERMITS	NONE	ADDWINTERS	TREND	0.00100	0.018784	0.05324	0.95771
PERMITS	NONE	ADDWINTERS	SEASON	0.00100	0.094037	0.01063	0.99155



On the left, the Winters multiplicative method has a seasonal smoothing parameter 0.90927, giving much weight to the most recent seasonal pattern. This is reflected in the forecasts whose seasonal pattern is much closer to that of the most recent year than earlier years. The Winters additive method, in contrast, has a very small weight 0.001 (probably a boundary value). It uses more of an equally weighted average of all past years. The spread and seasonal fluctuations are larger for the multiplicative approach.

6.4 Diagnostics

Choosing among several models can be challenging. One option is to withhold some data at the end of the series, and then compare multistep forecasts to actual observations. Alternatively, the BACK= option can be used.

6.4.1 Validation

Validation typically refers to withholding a subset of the data, usually the last part in time series. The earlier part of the data, the *training data*, is then used to do the fitting. Multistep forecasts from the training data are compared to the observed values in the withheld data. This was done in the program that produced **Output 6.5**. There are some caveats. If the last part of the data is atypical (for example, if there is a visually apparent level shift or change in slope in the last few observations), then withholding those from the model-fitting procedure and using them for evaluation dooms the model's forecasts to look bad when compared to the withheld data. For seasonal data, a good rule is to withhold at least one full season. For this and other reasons, withholding might not leave a sufficient training set to get good parameter estimates, and the withholding approach must be abandoned.

In the ESM procedure, all of the historical data are used for the estimation of parameters. No data are withheld. This is different from what was done in the program that produced **Output 6.5**. With the BACK=18 option in PROC ESM, for example, all historical data including the last 18 observations are used to fit the parameters. But, the forecasts for the last 18 points are not all one-step-ahead forecasts as in the earlier data. They are multistep-ahead instead, just as what would have been done if the last 18 observations had been missing. The last forecast is an 18-step-ahead forecast, but the parameter estimates are from the full data. In that sense, the BACK= option allows some influence of the last observations on their own predictions, unlike a pure validation approach in which the last observations are withheld from the estimation stage.

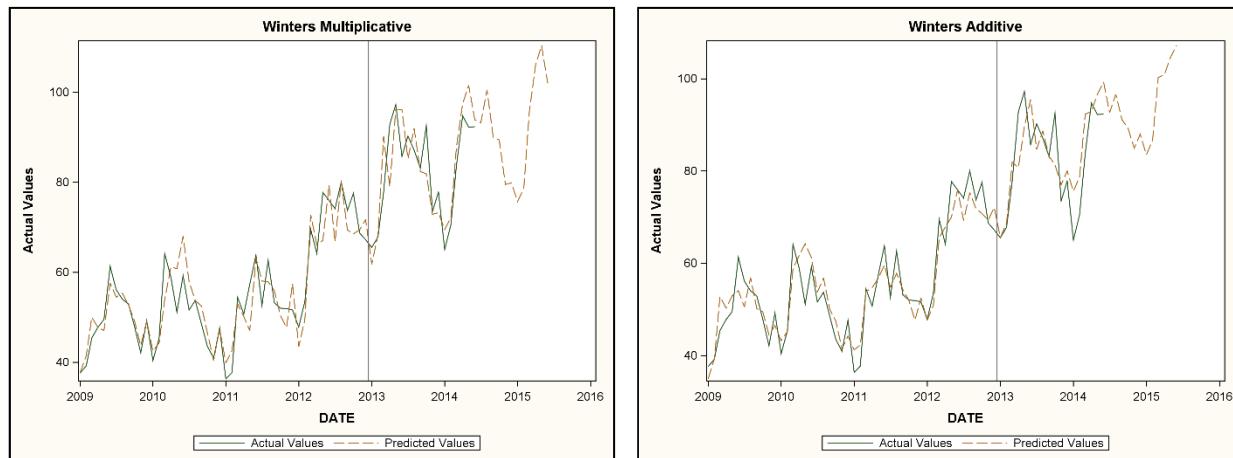
6.4.2 Choosing a Model Visually

The following code gives multistep-ahead forecasts for the last 18 observations in the housing permits data in **Output 6.22**. Plots of the forecasts and actual values are shown in the right panel of **Output 6.26**. Multistep forecasts to the right of the vertical reference line follow the data well. The Winters additive method is used.

```
proc esm data=housing outest=betas outfor=for_add lead=24 back=18 plot = (all);
  forecast permits/method=addwinters;
  id date interval=month;
run;
proc sgplot data=for_add;
  series y=actual x=date;
  series y=predict x=date;
  refline "15dec2012" d/axis=x;
run;
```

Changing the method to METHOD=WINTERS produces the multiplicative Winters results, which appear in the left panel of **Output 6.26**.

Output 6.26: Multistep Forecasts for Winters Multiplicative (left) and Winters Additive Models (right)

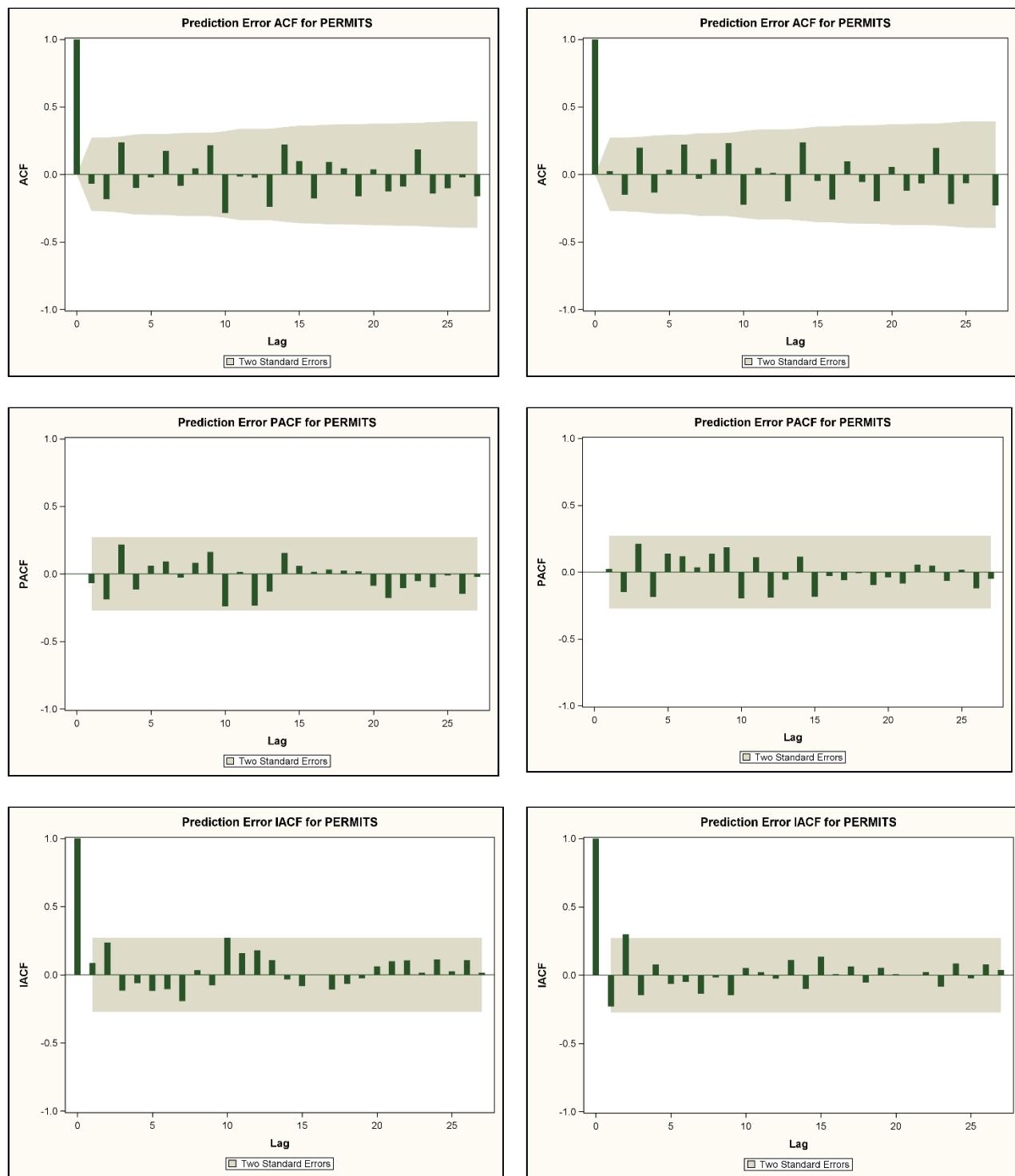


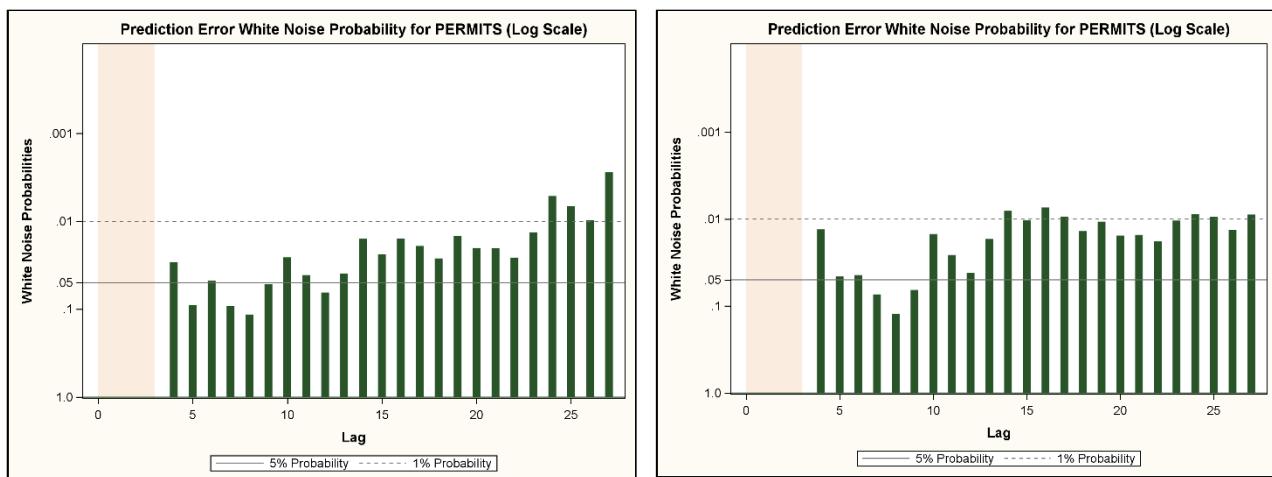
The leftmost plot forecasts (from the multiplicative method) are slightly closer to the last 18 observations. Both methods do a nice job of forecasting.

As always, an analyst strives for uncorrelated residuals. The PLOT=(ALL) option produces several diagnostics, including the familiar ACF, PACF, and IACF functions, as well as the chi-square test for white noise. These are shown in **Output 6.27** and are interpreted as they are in PROC ARIMA output. The chi-square test produces *p*-values that are graphed on an inverted logarithmic scale. Changing the method to METHOD=WINTERS produces the corresponding plots for the Winters additive method.

Output 6.27 shows the ACF, PACF, IACF, and chi-square graphs in four rows. In each row, the multiplicative model results are on the left and the additive model results are on the right.

Output 6.27: Diagnostic Graphs: ACF, PACF, IACF, and Chi-Square with Multiplicative Model Results on the Left and Additive Model Results on the Right





The ACF, PACF, and IACF functions show little or no evidence of autocorrelation. The white noise probability values in the lower left panel, on the inverted logarithmic vertical scale, have a few values above the 0.05 horizontal reference line. Probability values less than 0.05 suggest autocorrelation. A few are above the 0.01 reference line, indicating significance at the 0.01 level. In the chi-square tests on the left, there is some evidence of correlation. These refer to residual correlation in the multiplicative model.

The bottom right panel of **Output 6.27** shows a similar result for the additive model. However, the lag numbers at which the 0.05 and 0.01 reference lines are exceeded, as well as the number of exceedances, are not the same as the multiplicative model. In summary, neither model is perfect and both are reasonable. You might note that the *p*-values less than 0.01 arise earlier in the left panel than in the right panel in the bottom row of **Output 6.27**. If you are forced to choose, you might want to select the additive model. If the maximum exceedance (regardless of lag number) were of concern, the multiplicative model should be chosen. In this example, there is no outstanding reason to prefer one model over the other.

If there is enough evidence of lack of fit, then another model might be fit to the data. In general, that might be a different exponential smoothing model or it might be something outside the class of models available in PROC ESM. In the current case, the analyst would likely look to a seasonal ARIMA model to see whether the fit could be improved.

6.4.3 Choosing a Model Numerically

In addition to **Output 6.27**, the previous program produced an output data set by specifying OUTFOR=FOR_ADD. Using METHOD=WINTERS and OUTFOR=FOR_MULT in the previous code produces a second output data set containing the results of using the Winters multiplicative model. These data sets have residuals and actual values so that numerical evaluations of fit, restricted to the last 18 observations, can be computed. Several evaluation statistics have been proposed in literature, such as the average squared error and the mean absolute percentage error or MAPE.

With a true validation data set, one that has no influence on the fitted model, the average squared error would be the sum of squared errors divided by the number of errors used, not by degrees of freedom. Dividing by degrees of freedom rather than the number of errors used is appropriate when the same data are used to fit the model and evaluate it. For example, in a random sample of size n from a distribution, the sample mean fits the data better than the true overall mean. The sum of squared deviations from the sample mean estimates $n-1$ times the variance. So, $n-1$ is the divisor, where n is the number of deviations used. If the true mean were known, the squared deviations from it would simply be averaged.

Although the BACK= option does not affect the number of observations used in the model-fitting process, the computations that follow treat the last 18 observations as if they were actually withheld. First, squared errors and absolute percentage errors of the form $|\text{error}|/\text{observation}$ are produced for each observation using the following code:

```
data for2;
  set for_add;
  if date > "01dec2012"d then do;
    e2_add = error**2;
    pcterr_add = 100*abs(error)/actual;
  end;
run;
```

This code produces a data set, FOR2, for the additive case. Similarly, it produces a data set, FOR3, for the multiplicative case. These are merged, and average squared errors and mean absolute percentage errors (MAPEs) are computed.

Output 6.28 gives the results, showing that both evaluation statistics slightly prefer the multiplicative model. Here is the code:

```
data all;
  merge for2 for3;
run;

proc means n mean data=all;
  title "model comparison";
  var e2_add e2_mult pcterr_add pcterr_mult;
  label e2_add = "Squared Errors, Additive";
  label e2_mult = "Squared Errors, Multiplicative";
  label pcterr_add = "Percentage Errors, Additive";
  label pcterr_mult = "Percentage Errors, Multiplicative";
run;
```

Output 6.28: Average Squared Errors and MAPEs for the Two Models

Variable	Label	N	Mean
E2_ADD	Squared Errors, Additive	18	45.3496203
E2_MULT	Squared Errors, Multiplicative	18	43.6995169
PCTERR_ADD	Percentage Errors, Additive	18	6.6100106
PCTERR_MULT	Percentage Errors, Multiplicative	18	6.0877979

6.5 Advantages of Exponential Smoothing

A strong selling point of exponential smoothing models and PROC ESM, in particular, is the ability to forecast a large collection of time series all at once. This is similar to what was done in the example with 35 new products. For example, with thousands of time series to forecast, it would be hard to view graphs of each. Checking for boundary conditions (if done at all) would likely need to be done by a computer. Several examples of exponential smoothing have been presented. Many of these have quite reasonable forecasts, even when the estimated weights land on the software-imposed boundary that keeps the estimates inside the (0,1) interval.

When exponential smoothing models are run automatically, the hope is that the series are the type that result in reasonable forecasts. There are examples (for example, the housing permits, tides, and money circulation data) in which the graphs of the forecasts are unreasonable and would likely give a practitioner very strong reservations about their applicability, to say the least. This shows the danger of reporting these (or any other statistical analyses for that matter) without looking at the results.

These and most other time series methods assume that the underlying model structure is consistent across the entire analysis period. The housing permits data analysis is an example in which this assumption seems untenable. A price is paid when the Winters methods are used on the full data. Depending on the weights, the forecasts can go from almost ignoring data in the distant past to almost weighing it as much as the current data. The housing permits example has given sufficient weight to the long yet temporary period of decline that results in a decreasing forecast, despite several recent years of increasing numbers of permits issued.

6.6 How the Smoothing Equations Lead to ARIMA in the Linear Case

This section is optional reading. It is intended for readers who want to see an algebraic development of the relationship between the linear exponential smoothing model and ARIMA models. There are restrictions on the ARIMA parameters corresponding to the requirement that the smoothing weights lie between 0 and 1.

For the linear smoothing case, showing the equivalence between the smoothing equations and ARIMA models requires a bit of algebra. The forecast of Y at time t is the level at time $t - 1$ plus 1 times the trend (slope) estimate at time $t - 1$. The level and trend smoothing equations and the associated first differences are as follows:

$$L_t = \omega Y_t + (1 - \omega)(L_{t-1} + T_{t-1}) = L_{t-1} + T_{t-1} + \omega(Y_t - \hat{Y}_t) = L_{t-1} + T_{t-1} + \omega(Y_t - L_{t-1} - T_{t-1})$$

As a result, $L_t - L_{t-1} = T_{t-1} + \omega(Y_t - L_{t-1} - T_{t-1})$. Using this equality, for the trend you have the following:

$$T_t = \gamma(L_t - L_{t-1}) + (1-\gamma)T_{t-1} = \gamma(T_{t-1} + \omega(Y_t - L_{t-1} - T_{t-1})) + (1-\gamma)T_{t-1} = \omega\gamma(Y_t - L_{t-1} - T_{t-1}) + T_{t-1}$$

Rearranging this gives $T_t - T_{t-1} = \omega\gamma(Y_t - L_{t-1} - T_{t-1}) = \omega\gamma e_t$.

Assuming that the difference $Y_t - L_{t-1} - T_{t-1}$ between Y_t and its predictions from its predecessor is a white noise process e_t , it follows that $Y_t - \hat{Y}_t - (Y_{t-1} - \hat{Y}_{t-1}) = e_t - e_{t-1}$. Substituting the smoothing equation forecasts of the form $L_{t-1} + T_{t-1}$, this becomes the following:

$$\begin{aligned} Y_t - Y_{t-1} - (\hat{Y}_t - \hat{Y}_{t-1}) &= Y_t - Y_{t-1} - (L_{t-1} - L_{t-2}) - (T_{t-1} - T_{t-2}) \\ &= Y_t - Y_{t-1} - (T_{t-2} + \omega e_{t-1}) - \omega\gamma e_{t-1} \\ &= e_t - e_{t-1} \end{aligned}$$

Having shown in the last equality that $Y_t - Y_{t-1} = T_{t-2} + \omega(1+\gamma)e_{t-1} + e_t - e_{t-1}$ and knowing that the first difference of T is just a multiple $\omega\gamma$ of white noise, one more difference will express Y in terms of the white noise series. That is, it will give the ARIMA equivalent representation. This becomes the following:

$$\begin{aligned} Y_t - 2Y_{t-1} + Y_{t-2} &= \omega\gamma e_{t-2} + \omega(1+\gamma)(e_{t-1} - e_{t-2}) + e_t - 2e_{t-1} + e_{t-2} \\ &= e_t - (2 - \omega - \omega\gamma)e_{t-1} - (\omega - 1)e_{t-2} \end{aligned}$$

In typical ARIMA notation, this is $(1 - B)^2 Y_t = e_t - \theta_1 e_{t-1} - \theta_2 e_{t-2}$, where $\theta_1 = 2 - \omega - \omega\gamma$ and $\theta_2 = \omega - 1$.

Double exponential smoothing applies an operator $(1 - B)Y_t = (1 - \omega B)e_t$, and then applies that same operator again, getting $(1 - B)^2 Y_t = (1 - \omega B)^2 e_t$. Is the linear method the same idea using a different coefficient for the second smooth? This would be the case if the moving average backshift operator factored into two factors, each with a real number coefficient. Using the quadratic formula and $\theta_1 = 2 - \omega - \omega\gamma$ and $\theta_2 = \omega - 1$, the question is one of whether the roots of the polynomial $1 - \theta_1 B - \theta_2 B^2$ (or alternatively those of $m^2 - \theta_1 m - \theta_2$) are real rather than a complex pair.

The quadratic equation $f(m) = m^2 - \theta_1 m - \theta_2$ is an upward-opening parabola. The quadratic formula involves the following expression: $(\theta_1^2 + 4\theta_2)^{1/2}$.

If the expression in parentheses is negative for any combination of $0 < \omega < 1$, $0 < \gamma < 1$, then for that combination, the roots are not real. The linear smoother is not the result of two consecutive applications of single exponential smoothing operators as would be the case if $\omega = \gamma$. An easy non-algebraic way to check this is to generate the values under the square root over a grid of (ω, γ) and plot the results. The value whose square root is desired will be 0 if $\omega = 4\gamma/(1 + \gamma)^2$, which is the boundary between the complex and real root regions shown in **Output 6.29**. The real root region is the set of points with $\omega \geq 4\gamma/(1 + \gamma)^2$. The conclusion is that Holt's method of linear exponential smoothing cannot always be duplicated by repeated applications of single smoothers with weights that are real numbers. The exact boundary values, 0 and 1, of the smoothing parameters are included.

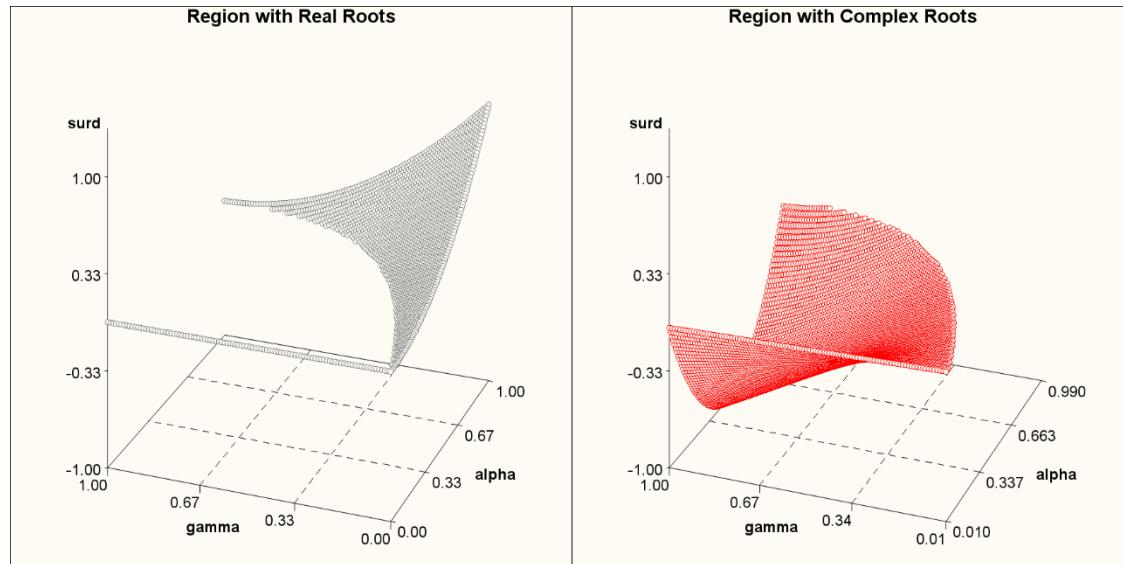
```
data complex;
length roots $ 7 cval $ 5;
do omega = 0 to 1 by 0.01;
  do gamma=0 to 1 by 0.01;
    theta1=2-omega-omega*gamma;
    theta2=omega-1;
    surd=theta1*theta1+4*theta2;
    if surd<0 then roots="complex";
    else roots="real";
    if surd<0 then cval = "red"; else cval="gray";
    if surd=0 then cval="green";
    output;
  end;
end;
run;

proc g3d; where roots="real";
  scatter gamma*omega=surd / shape = "balloon" size=0.6 noneedle zmin=-1 zmax=1
    color="gray";
  title "REGION WITH REAL ROOTS";
run;
```

```

proc g3d; where roots="complex";
  scatter gamma*omega=surd / shape = "balloon" size=0.6 noneedle zmin=-1 zmax=1
    color="red";
  title "REGION WITH COMPLEX ROOTS";
run;

```

Output 6.29: Portion of Quadratic Formula Versus Gamma (Right to Left) and Omega (Front to Back)

Chapter 7: Unobserved Components and State Space Models

7.1 Nonseasonal Unobserved Components Models	243
7.1.1 The Nature of Unobserved Components Models.....	243
7.1.2 A Look at the PROC UCM Output.....	246
7.1.3 A Note on Unit Roots in Practice	247
7.1.4 The Basic Structural Model Related to ARIMA Structures.....	247
7.1.5 A Follow-Up on the Example.....	249
7.2 Diffuse Likelihood and Kalman Filter: Overview and a Simple Case	250
7.2.1 Diffuse Likelihood in a Simple Model.....	251
7.2.2 Definition of a Diffuse Likelihood	251
7.2.3 A Numerical Example	252
7.3 Seasonality in Unobserved Components Models	254
7.3.1 Description of Seasonal Recursions	254
7.3.2 Tourism Example with Regular Seasonality	254
7.3.3 Decomposition	257
7.3.4 Another Seasonal Model: Sine and Cosine Terms.....	258
7.3.5 Example with Trigonometric Components	259
7.3.6 The Seasonal Component Made Local and Damped	261
7.4 A Brief Introduction to the SSM Procedure.....	265
7.4.1 Brief Overview	265
7.4.2 Simple Examples.....	265
7.4.3 Extensions of the AR(1) Model.....	266
7.4.4 Accommodation for Curvature	267
7.4.5 Models with Several Lags	270
7.4.6 Bivariate Examples	273
7.4.7 The Start-up Problem Revisited	274
7.4.8 Example and More Details on the State Space Approach	276

7.1 Nonseasonal Unobserved Components Models

It is common to think of a time series as a sum of components (for example, trend, seasonal, and irregular components). In this section, particular forms of these components are proposed and used to decompose a series, producing interesting insights into the series behavior and creating informative graphs.

7.1.1 The Nature of Unobserved Components Models

The SSM (state space models) and UCM (unobserved components models) procedures are very general. They are best understood by starting with some simple, familiar cases. This chapter serves as an introduction. It does not paint the full picture of the broad functionality of these procedures for univariate and multivariate applications.

A key feature of the state space approach to modeling that is used in the UCM and SSM procedures is the assumption of a state vector α_t that can be partly or entirely unobserved or latent. There is also the assumption of a state vector transition equation that shows how the state vector changes from time t to $t + 1$. It allows for regression type inputs through $\mathbf{W}_{t+1}\gamma$, deterministic inputs \mathbf{c}_{t+1} , and a random error vector $\eta_{t+1} \sim N(0, \mathbf{Q})$. This constitutes the general *state vector transition equation*:

$$\alpha_{t+1} = \mathbf{T}_t \alpha_t + \mathbf{W}_{t+1} \gamma + \mathbf{c}_{t+1} + \eta_{t+1}$$

Typically, the state vector is not the vector of observations Y_{t+1} . The two vectors are not necessarily the same size. The state vector might contain the observed vector as a subvector. In any case, the vector of observations is related to the state vector and possibly a matrix of additional explanatory variables X_{t+1} through an *observation equation* given by the following expression:

$$\mathbf{Y}_{t+1} = \mathbf{Z}_{t+1} \boldsymbol{\alpha}_{t+1} + \mathbf{X}_{t+1} \boldsymbol{\beta} + \boldsymbol{\epsilon}_{t+1}$$

Here, \mathbf{Z}_{t+1} is a matrix linking the state vector to the observations and $\boldsymbol{\epsilon}_{t+1}$ is an error vector that has a diagonal variance matrix. In addition to the state vector transition equation and the measurement equation, a third *initial condition equation* is needed to get the recursion started. See the SSM procedure documentation for additional details.

A linear time trend with white noise errors is simple and can be handled by PROC REG and many other SAS procedures. The state space approach, as implemented in the UCM and SSM procedures, defines the model as the sum of components. Each component follows a recursion. Because a linear trend increases by its slope β with each unit increase in time, the mean μ_t of the data Y_t at time t can be written recursively as $\mu_{t+1} = \mu_t + \beta_t$, where $\beta_{t+1} = \beta_t$. This, in turn, can be summarized in a (matrix) state vector transition equation:

$$\begin{pmatrix} \mu_{t+1} \\ \beta_{t+1} \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \mu_t \\ \beta_t \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

The second recursion keeps the slope β constant and equal to whatever is set as its initial value, for example, β_0 . An initial value μ_0 serves as the intercept. Each level μ_t is its predecessor plus an additional β . These recursions describe the level at time $t+1$ as $\mu_{t+1} = \mu_0 + \beta_0(t+1)$. The observation equation then becomes the following, which is just the usual linear regression model:

$$Y_{t+1} = (1 \ 0) \begin{pmatrix} \mu_{t+1} \\ \beta_{t+1} \end{pmatrix} + e_{1,t+1}$$

The model at time t is $Y_t = \mu_0 + \beta_0 t + e_{1,t}$. If the errors are normal white noise, the usual least squares regression estimators maximize the likelihood. The models handled by the UCM and SSM procedures are much more general, and initial values for the component states are required.

This seems at first glance to be making a simple task harder. Indeed, it is for the simple regression case, but it allows for some appealing variations. Instead of a fixed intercept and slope for all time, it might be more appropriate, for some data, to allow these parameters to be local in nature. That is accomplished by adding a random error term (innovation) at each step of the recursive calculation. The recursions then become

$$\beta_{t+1} = \beta_t + e_{3,t+1}$$

and

$$\mu_{t+1} = \mu_t + \beta_t + e_{2,t+1}$$

The state vector at time t consists of elements μ_t and β_t . Thus, in matrix form, the state vector transition equation becomes the following:

$$\begin{pmatrix} \mu_{t+1} \\ \beta_{t+1} \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \mu_t \\ \beta_t \end{pmatrix} + \begin{pmatrix} e_{2,t+1} \\ e_{3,t+1} \end{pmatrix}$$

This explains why the double subscript was used initially on the white noise error term and illustrates the role of the state transition equation's error term. The smaller the variances of these two white noise terms, the closer the data are to following a simple linear regression. The larger the variances, the more the local level and slope vary. This model is known as the *basic structural model*.

The second line, $\beta_{t+1} = \beta_t + e_{3,t+1}$, of the transition equation produces the following sequence, where $W_3 = e_{3,1} + e_{3,2} + e_{3,3}$:

$$\begin{aligned}\beta_1 &= \beta_0 + e_{3,1} \\ \beta_2 &= \beta_1 + e_{3,2} = \beta_0 + e_{3,1} + e_{3,2} \\ \beta_3 &= \beta_2 + e_{3,3} = (\beta_0 + e_{3,1} + e_{3,2}) + e_{3,3} = \beta_0 + W_3\end{aligned}$$

In general, $\beta_{t+1} = \beta_0 + W_{t+1}$, where $W_{t+1} = e_{3,1} + e_{3,2} + e_{3,3} + \dots + e_{3,t+1}$ is a random walk. Notice the following:

$$\sum'_{j=0} \beta_j = \sum'_{j=0} (\beta_0 + W_j) = (t+1)\beta_0 + \sum'_{j=1} W_j$$

Similarly, the first line $\mu_{t+1} = \mu_t + \beta_t + e_{2,t+1}$ of the transition equation produces the following sequence:

$$\begin{aligned}\mu_1 &= \mu_0 + \beta_0 + e_{2,1} \\ \mu_2 &= \mu_1 + \beta_1 + e_{2,2} = (\mu_0 + \beta_0 + e_{2,1}) + (\beta_0 + e_{3,1}) + e_{2,2} = \mu_0 + 2\beta_0 + W_1 + (e_{2,1} + e_{2,2})\end{aligned}$$

In general,

$$\mu_{t+1} = \mu_0 + \sum'_{j=0} \beta_j + (e_{2,1} + \dots + e_{2,t+1}) = \mu_0 + (t+1)\beta_0 + \sum'_{j=1} W_j + (e_{2,1} + \dots + e_{2,t+1})$$

This is the linear trend $\mu_0 + (t+1)\beta_0$, plus the cumulative sum of a random walk:

$$\sum'_{j=1} W_j$$

In addition, there is another independent random walk $(e_{2,1} + \dots + e_{2,t+1})$. The sum of a random walk has a repeated unit root. Consider the random portion of the level μ_{t+1} :

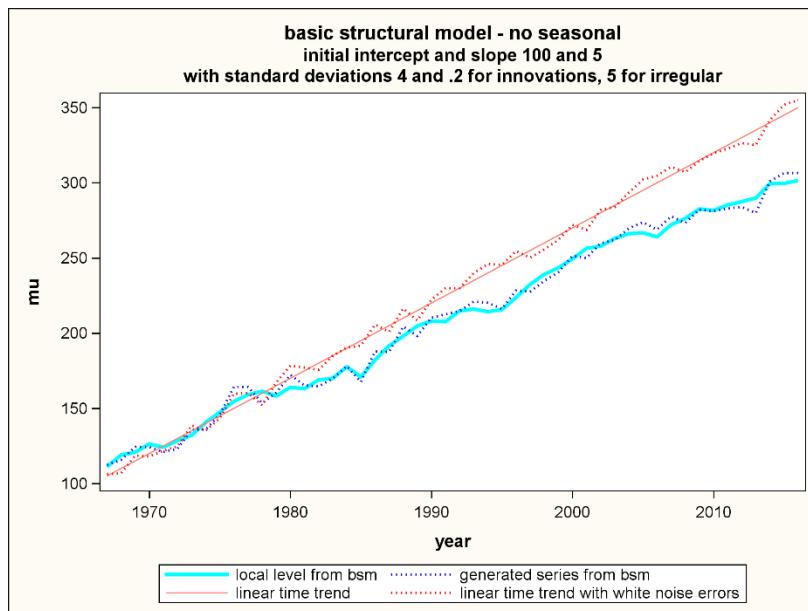
$$\sum'_{j=1} W_j + (e_{2,1} + \dots + e_{2,t+1})$$

It allows the level to deviate from the linear trend part $\mu_0 + (t+1)\beta_0$ increasingly as time goes by.

Output 7.1 represents 50 observations generated from a basic structural model. The initial intercept and slope, 100 and 5, produce the displayed straight line that would occur if no innovations e were present in the recursions. Adding the observation equation's error term produces the data scattered around the straight line. For the dynamic case, the level and slope innovations, $e_{2,t+1}$ and $e_{3,t+1}$ respectively, have standard deviations 4 and 0.2, both exceeding 0. As previously shown, β_{t+1} is β_0 plus the random walk sequence W_{t+1} . It produces local slopes that can vary from the initial slope more as time goes by. Furthermore, μ_{t+1} includes the cumulative sum of the W_{t+1} random walk. The cumulative sum of a random walk is a series with a repeated unit root of multiplicity 2. It also includes the cumulative sum of $e_{2,t}$, an additional random walk. The local level can be far from the line $100 + 5t$ as time goes by. In **Output 7.1**, the light, thick, and wavy line is the local level μ_t , which shows the characteristic smoothness of a process with multiple unit roots. Overlaid on this are the data $Y_t = \mu_t + e_{1,t}$, generated from the UCM. The linear predictions from the corresponding straight line model $Y_t = 100 + 5t + e_{1,t}$ lie above the UCM data. Both data series use the same error terms $e_{1,t}$.

The UCM procedure fits such a recursive model. For the **Output 7.1** data, the appropriate code is:

```
proc ucm data=ucm1;
  id date interval=year;
  model y;
  irregular;
  level;
  slope;
  forecast lead=24 print=decomp out=out1;
run;
```

Output 7.1: Example of Basic Structural Model Data Compared to Linear Model Data

Partial output is shown in **Output 7.2**.

Output 7.2: Partial Results from PROC UCM

Final Estimates of the Free Parameters					
Component	Parameter	Estimate	Approx Std Error	t Value	Approx Pr > t
Irregular	Error Variance	21.57741	7.79316	2.77	0.0056
Level	Error Variance	11.75836	7.54524	1.56	0.1191
Slope	Error Variance	1.948434E-7	0.0001275	0.00	0.9988

Significance Analysis of Components (Based on the Final State)			
Component	DF	Chi-Square	Pr > ChiSq
Irregular	1	0.00	0.9642
Level	1	8306.57	<.0001
Slope	1	62.97	<.0001

Trend Information (Based on the Final State)		
Name	Estimate	Standard Error
Level	306.671099	3.3648215
Slope	3.964446419	0.4995859

7.1.2 A Look at the PROC UCM Output

Output 7.2 is organized in three tables. The top table shows the estimated innovation variances for the transition equation's two recursions and for the error ("irregular" in UCM syntax) variance. Although the data were generated with a positive variance 0.04 for the slope recursion, the variance is quite small. With only 50 data points, the estimated slope variance is less than 0.000001 and not statistically different from 0 ($p=0.9988$).

Not only does the small sample size affect the power of the test, it also affects the accuracy of the estimates and standard errors as they are based on asymptotic theory. This could explain why the estimate is so many approximate standard errors from 0.04. Further, it is worth noting that a test that a variance parameter is on the boundary of feasible values (for

example, variance=0) no longer has the usual distribution (for example, see Self and Liang 1987). You must view these results only as a guide rather than as a rigorously justified hypothesis test. For a test of a single variance, the Self and Liang results imply that the p -value can be divided in half so that the level variance p -value changes from 0.1191 to 0.0595.

The level variance that generated the data was 16. Its estimate is 11.76, which is less than a single standard error from the true value 16. If the sample size is increased to 5000, the estimates change to 0.0469 and 15.837, both much closer to the true values 0.04 and 16. This is consistent with the asymptotic theory. The first row in the second table shows that the final irregular term is not significantly different from 0. This implies that the last observation is close to the final local level. The final level and slope estimates are given in the third table. They are significantly different from 0 based on their chi-square tests. Unlike the variance components, the slope and level components have no a priori bounds on their feasible values. Thus, the usual asymptotic theory holds for these. The third table allows the reader to divide each estimate by its standard error, giving a ratio that would be an approximately standard normal statistic in large samples. Just as an F test statistic with 1 numerator degree of freedom is the square of a corresponding t , so are the chi-square statistics in the second table the squares of these ratios from the third table. Based on this third table, the forecast L periods into the future is $306.67 + 3.96L$.

If the innovation variance of the slope were really 0, then the slope recursion would become $\beta_{t+1} = \beta_t$. That is, there would be a common slope β at all times, and β_t would no longer contain a unit root time series. Based on the second and third tables, β appears to be nonzero. The level recursion would become $\mu_{t+1} = \beta + \mu_t + e_{3,t}$, which is a random walk with drift β . It has only a single unit root, not a repeated (multiplicity 2) unit root. This implies that, with a constant slope, a second difference of the response variable would be too much and would introduce a unit root on the moving average side in the corresponding ARIMA representation. The UCM procedure allows the user to restrict a component's variance to 0.

7.1.3 A Note on Unit Roots in Practice

The UCM basic structural model with all variances positive is a series with a double unit root. Finding a unit root of multiplicity 2 through model identification methods is rare in practice. Either the original or first differenced series usually appears to be stationary. Many exponential smoothing models, close relatives of UCMs and SSMs, also have theoretically equivalent ARIMA representations with unit roots. For example, the single exponential smoothing model is a version of the integrated moving average model $Y_t - Y_{t-1} = e_t - \theta e_{t-1}$ with $0 < \theta < 1$. Because this model produces a weighted average of past Y values, with values further back receiving lower weight, it has common-sense appeal as a way to compute a forecast consisting of a local series level.

The common-sense appeal remains even when such a process is rejected by unit root testing. For this reason, exponential smoothing models, also known as exponentially weighted moving averages, are routinely used in quality control applications without prior unit root testing. The point is that the UCMs and SSMs can produce reasonable forecasts even in some cases where there is statistical evidence that a series does not have the theoretically appropriate number of unit roots. They have the same intuitive appeal as exponential smoothing models—forecasts that weight recent data more heavily than data in the past. There is also a disadvantage. Forecast standard errors for unit root processes grow rapidly as the lead time L increases. The unappealing wide forecast error bands of unit root models could be avoided if a linear trend model with stationary errors were a reasonable alternative. Such an alternative model could be fit in the UCM, SSM, or ARIMA procedure.

7.1.4 The Basic Structural Model Related to ARIMA Structures

Again, consider the basic structural model. The previous notation, $\beta_{t+1} = \beta_t + e_{3,t+1}$, becomes:

$$\beta_{t+1} = \beta_0 + \sum_{j=1}^{t+1} e_{3,j}$$

In addition, $\mu_{t+1} - \mu_t = \beta_t + e_{2,t+1}$, which is the following:

$$\mu_{t+1} - \mu_t = \beta_0 + \sum_{j=1}^t e_{3,j} + e_{2,t+1}$$

The first equation describes a random walk that starts at β_0 . A first difference makes it stationary. In the second equation, as is shown, the first difference of μ_t is a constant plus a random walk that starts from 0 plus a white noise process.

Because of the random walk component, it requires yet another difference to make it stationary. Doing so eliminates the constant β_0 and produces:

$$(1 - B)^2 \mu_{t+1} = e_{3,t+1} + e_{2,t+1} - e_{2,t}$$

With $Y_{t+1} = \mu_{t+1} + e_{1,t+1}$, it follows that Y is an ARIMA(0,2,2) given by:

$$(1 - B)^2 Y_{t+1} = e_{3,t+1} + (e_{2,t+1} - e_{2,t}) + (e_{1,t+1} - 2e_{1,t} + e_{1,t-1})$$

The autocovariances of the combination of white noise terms on the right side of this model are:

$$\begin{aligned}\gamma(0) &= \sigma_3^2 + 2\sigma_2^2 + 6\sigma_1^2 \\ \gamma(1) &= -\sigma_2^2 - 4\sigma_1^2 \\ \gamma(2) &= \sigma_1^2 \\ \gamma(j) &= 0 \quad \text{for } j > 2\end{aligned}$$

Such dependence at lag 2, but not after 2, defines a moving average of order 2. Thus, the basic structural model, theoretically, is an ARIMA(0,2,2) model. This model might produce useful forecasts even when such a structure is rejected by standard identification techniques.

Because the data in **Output 7.1** were generated exactly from a UCM basic structural model, the equivalence to an ARIMA(0,2,2) model should hold up. The small sample size could affect the results. For example, the slope variance was already shown to be non-detectable statistically.

The following code results in **Output 7.3**, in which the moving average component is very close to a unit root:

```
proc arima data=ucm1;
  identify var=y(1,1);
  estimate q=2 method=ml noconstant;
run;
```

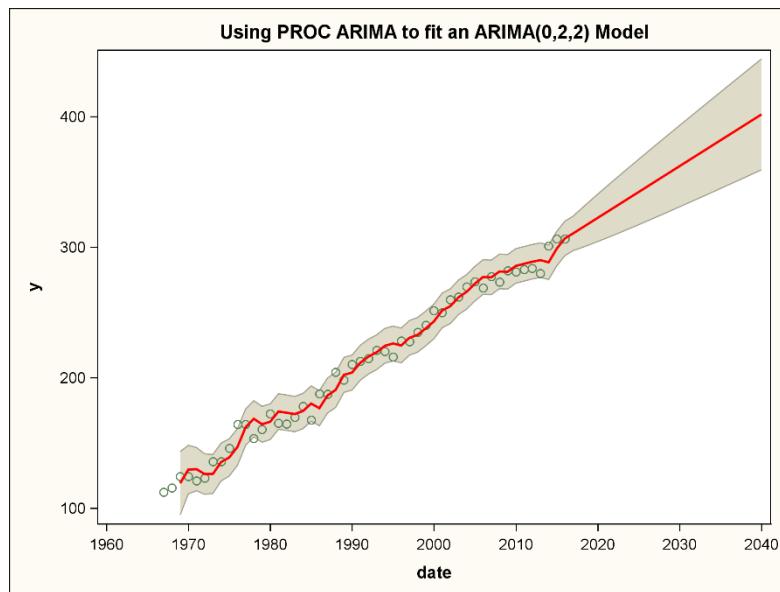
This is consistent with the result that the slope innovation variance was not significantly different from 0, suggesting that a second difference might appear to be overdifferencing (therefore, introducing a unit root on the moving average side). With slight rounding, the moving average operator factors as $1 - 1.325B + 0.325B^2 = (1 - 0.325B)(1 - B)$, consistent with overdifferencing.

Output 7.3. Estimated MA(2) for Second Differenced Data

Moving Average Factors	
Factor 1:	1 - 1.32462 B**(1) + 0.32482 B**(2)

A plot of the data with 12 forecasts into the future and forecast intervals in **Output 7.4** appears reasonable. The forecasts are consistent with the intercept and slope shown at the bottom of **Output 7.2**.

```
proc sgplot data=out1 noautolegend;
  band lower=lcl upper=ucl x=date / outline fill;
  scatter y=y x=date;
  series y=forecast x=date/lineattrs=(color=red thickness=2);
  title "Using PROC ARIMA to fit an ARIMA(0,2,2) Model";
run;
```

Output 7.4: Plot of Basic Structural Model Forecasts Using an ARIMA Model**7.1.5 A Follow-Up on the Example**

The data from **Output 7.1** suggested that perhaps a UCM with no slope innovations or, equivalently, an ARIMA(0,1,1) model with a drift might be fit to the data. Code useful for investigating these possibilities is:

```

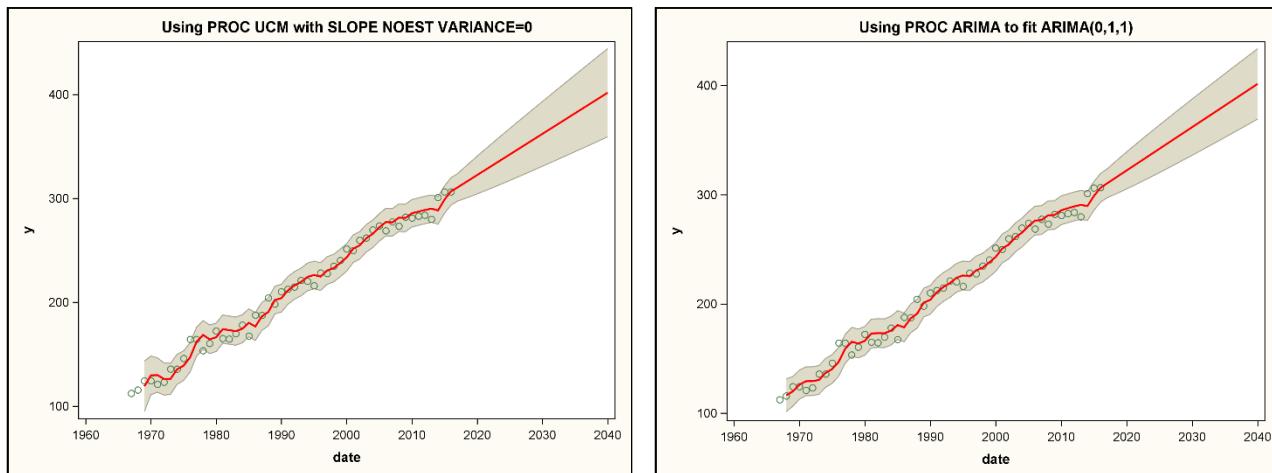
proc ucm data=ucml;
  id date interval=year;
  model y;
  irregular;
  level;
  slope noest variance=0;
  forecast lead=24 print=decomp out=out2;
run;

proc arima data=ucml;
  identify var=y(1) noprint;
  estimate q=1 method=ml;
  forecast lead=24 out=out3 id=date interval=year;
run;

```

Notice the SLOPE statement. A slope is needed, so a SLOPE statement is required. The initial variance value 0 is also the final value due to the NOEST option. This means the model's slope will be constant. Using graphics code similar to that previously shown, results from both UCM and ARIMA analyses are shown in **Output 7.5**. These results are very similar to each other and to those in **Output 7.4**, in which the slope is allowed to vary locally but with a very small estimated variance.

Output 7.5: Constant Slope Analyses with PROCs UCM and ARIMA



Although the forecasts are similar, notice that there is some difference in the width of the forecast intervals far past the end of the data. The year 2040 forecast standard errors for the two plots are 21.68 and 16.44. This difference results from the initial value estimation problem. Unit root processes underlie these models, and if the process started from some initial value long before the observations were taken, the initial level could be far from where the series started. PROC UCM makes less restrictive assumptions on the initial values than does PROC ARIMA, resulting in more uncertainty in the forecasts and a slightly later time of first forecast in the observed data.

7.2 Diffuse Likelihood and Kalman Filter: Overview and a Simple Case

The Kalman filter is named for the engineer Rudolph E. Kalman, who developed the method in 1960. The examples given in this section are simpler than the more general models of Kalman. They do not use the full functionality of the SSM and UCM procedures. Some matrices with elements estimated here are sometimes considered known from the laws of physics in the general Kalman filter. An example of this second type of application is the tracking of an aircraft using a vector, called the state vector of position, thrust, velocity, heading, acceleration, and the like. It contains all of the information needed to forecast the position of the aircraft into the future (apart from future values of deterministic inputs).

The transition of the state vector going from time t to $t+1$ uses the laws of physics under idealized conditions. In the real world, there is random buffeting from winds or other shocks (an error term) affecting the actual state vector at time $t+1$. The tracking mechanism (for example, remote radar measurements) might introduce a second random term, measurement error, in the observation equation. The SSM procedure provides more general models than does the UCM procedure.

For each successive time period t , the Kalman filter computes a forecast, based on data up through time t , and a forecast error variance estimate for the state vector at time $t+1$. When the observation at time $t+1$ comes in, update formulas are applied to get new forecasts and forecast error variances. This process produces a sequence of independent residuals (residual vectors) and their estimated variances. These can be used to sequentially construct a likelihood function. If there are unknown parameters (called *hyperparameters*) to be estimated, then for each choice of these a likelihood can be computed. Estimation consists of searching for the parameters that maximize the likelihood. Once the full data are obtained, past and future observations are allowed to influence the prediction at time t , resulting in smoothed states. In addition, once the estimation process converges, a diffuse likelihood and a profile likelihood are used to compute various fit statistics such as AIC. The relationship of diffuse to profile likelihood is similar to that between the REML likelihood and full blown maximum likelihood (in, for example, the MIXED procedure).

Because the procedure is recursive in nature, there is a question of how to get started. In most applications, start-up information is not available and a conservative method, the diffuse Kalman filter, is used. The idea, as in some Bayesian applications, is to put a prior distribution on the initial state that is uninformative. This is accomplished by proposing a normal distribution with large variance, making a modification that allows for a limit, and then taking the limit as the variance increases without bound. The result is sometimes called a *normal prior with infinite variance*, but a statistical purist would insist that a true normal distribution has a finite variance by definition.

7.2.1 Diffuse Likelihood in a Simple Model

Suppose in the basic structural model that $Y_t = \mu_t$. In this case, the model is just a random walk $Y_t = Y_{t-1} + e_{2,t}$, where $e_{2,t} \sim N(0, \sigma_2^2)$ and the $e_{2,t}$ form an independent white noise sequence. There is no slope or irregular component in the state transition equation. If an initial level μ is assumed and if the series started in the past at, for example, $Y_{-2} = \mu$, then $Y_{-1} = \mu + e_{2,-1}$, $Y_0 = \mu + e_{2,-1} + e_{2,0}$. The first observed value of the series is $Y_1 = \mu + \gamma$, where $\gamma = e_{2,-1} + e_{2,0} + e_{2,1} \sim N(0, 3\sigma_2^2)$ and, therefore, $Y_1 = \mu + \gamma \sim N(\mu, 3\sigma_2^2)$. The contribution of the first observation to $-2\ln(\text{Likelihood})$ is thus $\ln(2\pi) + \ln(3\sigma_2^2) + (Y_1 - \mu)^2 / 3\sigma_2^2$. There are $n - 1$ differences $d_t = Y_t - Y_{t-1} = e_{2,t}$, $t = 2, 3, \dots, n$ that contribute to $-2\ln(\text{Likelihood})$ as well. These are independent $N(0, \sigma_2^2)$ variables because the differences are white noise. Because Y_1 is a function only of μ and white noise terms up through $t=1$, the objective function $-2\ln(\text{Likelihood})$ is the sum of independent terms, one from Y_1 and others from the sequence of differences. The objective function, $-2\ln(\text{Likelihood})$, in this extremely simple example is expressed as follows:

$$[\ln(2\pi) + \ln(k\sigma_2^2) + \frac{(Y_1 - \mu)^2}{k\sigma_2^2}] + (n-1)\ln(2\pi) + (n-1)\ln(\sigma_2^2) + \sum_{t=2}^n \frac{(Y_t - Y_{t-1})^2}{\sigma_2^2}$$

Here, $k=3$ for the current example and $\gamma \sim N(0, k\sigma_2^2)$. Knowledge that the series started $k=2$ periods ago at a fixed level would almost never be available. In this particular model, no term other than $(Y_1 - \mu)^2 / k\sigma_2^2$ supplies any information on the initial level. The objective function can be computed sequentially as the data come in. The Kalman filter takes advantage of this sequential construction.

7.2.2 Definition of a Diffuse Likelihood

The variance $k\sigma_2^2$ of $\gamma = e_{2,1} + e_{2,0} + e_{2,-1} + \dots + e_{2,-k+2}$ allows for an estimate of the initial mean, albeit a not very precise one because it is computed from the first observation Y_1 , where $Y_1 - \mu$ has variance $k\sigma_2^2$, which could be quite large. Most likely, there is no known upper limit for k . Rather than saying $\gamma = e_{2,1} + e_{2,0} + e_{2,-1} + \dots + e_{2,-k+2}$, it makes sense to simply let $\gamma \sim N(0, C\sigma_2^2)$, where C could be arbitrarily large rather than being a known integer k . This gives an objective function:

$$[\ln(2\pi) + \ln(C) + \ln(\sigma_2^2) + \frac{(Y_1 - \mu)^2}{C\sigma_2^2}] + (n-1)\ln(2\pi) + (n-1)\ln(\sigma_2^2) + \sum_{t=2}^n \frac{(Y_t - Y_{t-1})^2}{\sigma_2^2}$$

Admitting that C is unbounded and attempting to take a limit as C approaches infinity, it is seen that $\ln(C) \rightarrow \infty$, resulting in an undefined limit. For any proposed (μ, σ_2^2) combination, the function diverges (pointwise) to infinity. The term $(Y_1 - \mu)^2 / (C\sigma_2^2)$ implies that in the limit, no useful information is provided by the first observation. Suppose that the part of the likelihood associated with the first observation is ignored:

$$[\ln(2\pi) + \ln(C) + \ln(\sigma_2^2) + \frac{(Y_1 - \mu)^2}{C\sigma_2^2}]$$

Then, the function becomes the pointwise limit of the following:

$$(n-1)\ln(2\pi) + (n-1)\ln(\sigma_2^2) + \sum_{t=2}^n \frac{(Y_t - Y_{t-1})^2}{\sigma_2^2}$$

The reader familiar with REML estimation might see an analogy with the way in which fixed parts of a mixed model, such as μ , are initially eliminated in PROC MIXED. The result of modifying the likelihood by omitting observation 1's contribution to $-2\ln(\text{Likelihood})$ and taking the limit as the variance (matrix) of the initial state increases without bound leads to what De Jong (1991) refers to as the *diffuse log likelihood*. The implementation of this idea used in the SSM procedure and in Durbin and Koopman (2012) eliminates the entire first observation's contribution to the objective function:

$$[\ln(2\pi) + \ln(C) + \ln(\sigma_2^2) + \frac{(Y_1 - \mu)^2}{C\sigma_2^2}]$$

This leaves $(n - 1)\ln(\sigma_2^2)$ in the objective function. In summary, the use of the diffuse Kalman filter leads to a likelihood based on $n - 1$ differences and takes this modified sample size into account. With this adjustment, the estimate of σ_2^2 consists of the sum of squared differences divided by $n - 1$. De Jong shows how to compute likelihood maximizing

estimates using what he terms the diffuse Kalman filter. This is used in the UCM and SSM procedures. Both procedures, especially the SSM procedure, enable fitting a very wide array of dynamic linear models.

7.2.3 A Numerical Example

In the general Kalman filter approach, the initial analysis moves forward using the data from time 1 up through t to compute the current state, to forecast one step ahead to time $t + 1$, and to compute the forecast error variance. Recall that when the time $t + 1$ data come, update formulas allow the new forecast and forecast error variance to be quickly computed. These lead to a sequential formulation of the log likelihood function. Once the full data are read in, the states based on full data are used to make a second set of component series referred to as the *smoothed series* and prediction error variances. These smoothed series are informative and can be viewed as output in the procedures.

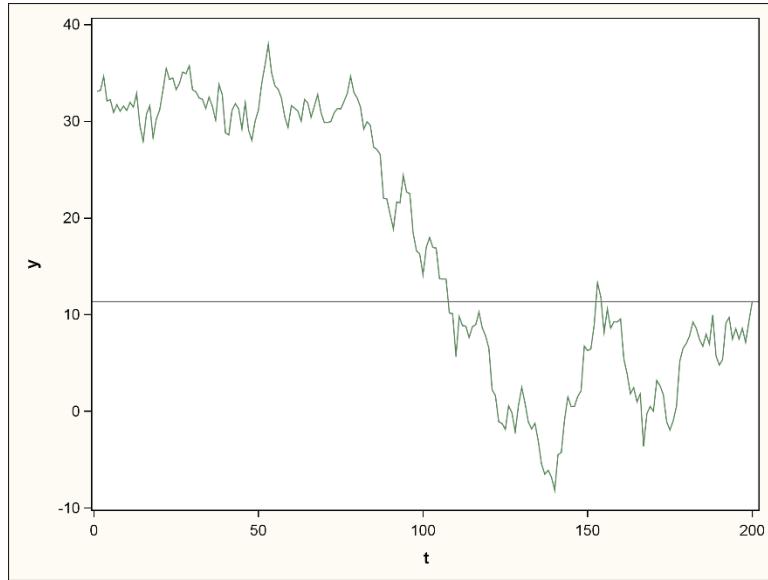
The following code produces 200 observations from a random walk model. The data start at time $t = -100$ at level $\mu_0 = 50$. A PROC SGLOT step shows the data. The LISTING statement with the GPATH= option sends the graphs to the Work directory rather than storing them permanently on the computer. A reference line is added to the plot at the final smoothed value of Y , 11.327, which serves as a forecast for all future values in this random walk. The resulting plot is in **Output 7.6**.

```
data a; y=50; * initial mean 50 *;
do t=-100 to 200;
  y = y + 2*normal(123); * error variance 4 *;
  if t>0 then output;
end; * omit startup *;

ods listing gpath = "%sysfunc(pathname(work))";
proc sgplot;
  series x=t y=y;
  refline 11.327 / axis = y;
run;
```

Run this code to get output that includes the error variance estimate.

Output 7.6: Plot of Random Walk Data with Reference Line at the Final UCM Level Estimate



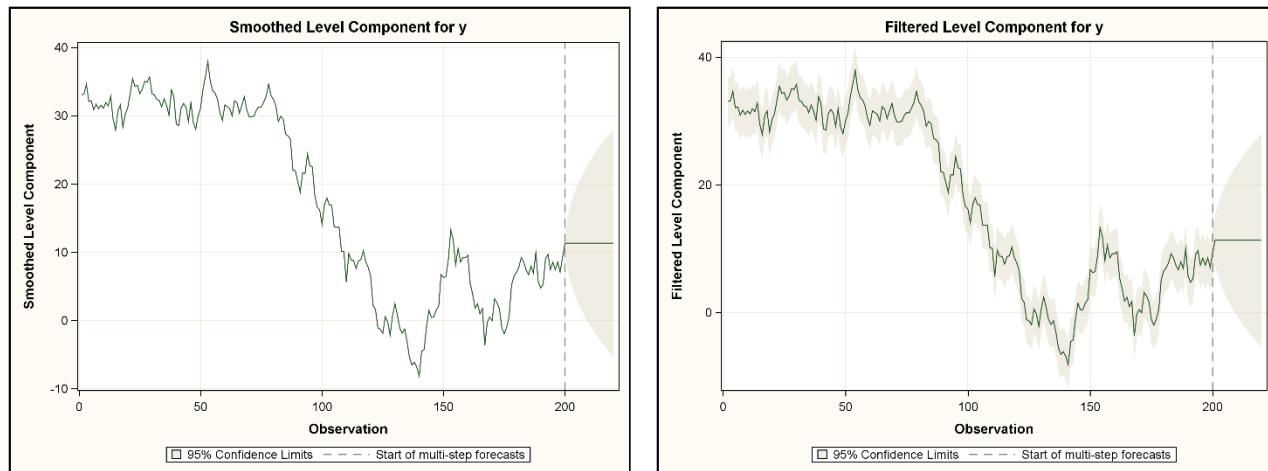
After time $t = 2$, the errors in this simple random walk model are just the first differences. The diffuse likelihood is maximized by setting the error variance to the sum of squared differences divided by $n - 1$. The final UCM level error variance estimate is 3.63964. There are no other parameters to estimate in this model. After all the model parameters are estimated, the UCM procedure uses the final parameter estimates to make two passes through the data—the forward pass produces the one-step-ahead forecasts and the backward pass produces the smoothed quantities, such as smoothed level estimates. The one-step-ahead prediction (filtered estimate) at time t predicts one step ahead, from data up through time t .

In this simple case, the one-step-ahead forecast is the current observation (that is, the prediction at $(t + 1)$ is the observation at t), and the standard deviation of the prediction is the square root of the level error variance, 3.63964. The

backward pass computes a smoothed level series from the full data set. Because Y equals this level, the level component reproduces Y and has standard error 0. The confidence limits both equal the observed series for this reason. Plots of these two (smoothed and filtered) series with error bands are computed by the UCM procedure when the PLOTS option is invoked. The following code, which fits the model that generated the data (no slope or irregular components), illustrates the idea. Below the code are the plots (**Output 7.7**).

```
proc ucm data=a ;
model y;
level plots=(smooth filter) print=(smooth filter);
forecast lead=20 print = all;
run;
```

Output 7.7: Two Plots, Smoothed Level and Filtered Level, from PROC UCM



As mentioned, the level component reproduces the data. With no IRREGULAR statement, the historical data's smoothed level plot on the left is just the data series. The standard error is 0. On the right, data through time t are used to forecast the time $t + 1$ value. Recalling that the final estimate of the error variance is used in this calculation, the forecast standard error is constant. Thus, the forecast intervals all have equal width through the historical series in this so-called filtered level component plot.

The PRINT option in the FORECAST statement shows that the forecast intervals past the end of the historical series are the same for both plots, as both are now forecasting into the future with the same estimated error variance. That is, with full data, the smoothed plot on the left can perfectly predict the smoothed component in the historical data because it is just a data point. After the observed data end, both plots show predictions of unobserved values using the same variance estimate from the full data.

The final variance estimate from the UCM procedure output is shown in **Output 7.8**, which also shows the output from the following program. The program computes the estimate in a DATA step by summing the squared first differences and dividing by $n - 1 = 199$. This reinforces the comments about the effect of using a diffuse likelihood.

```
data next;
set a end=eof;
diff = y-lag(y);
sse+diff*diff;
if _n_=1 then sse=0;
if eof then do;
variance = sse/(t-1);
output; end;
proc print;
var t variance;
run;
```

Output 7.8: Level Error from PROC UCM and DATA Step

Obs	t	variance
1	200	3.63964

Final Estimates of the Free Parameters					
Component	Parameter	Estimate	Approx Std Error	t Value	Approx Pr > t
Level	Error Variance	3.63964	0.36488	9.97	<.0001

7.3 Seasonality in Unobserved Components Models

A general modeling technique should address seasonality. These models are covered in this section.

7.3.1 Description of Seasonal Recursions

The idea of UCM and state space models is to model effects recursively. For an exactly periodic function, consider $f(t)$ with period p , as in $f(t) = f(t - p)$. This is a recursive equation that defines the term *periodic*. Because this is a recursion, it provides a way to handle seasonality in a manner compatible with the component approach. Seasonality can be made flexible and local by defining a seasonal component $S(t)$ of period p through the recursion $S(t) = S(t - p) + e_{4,t}$, where

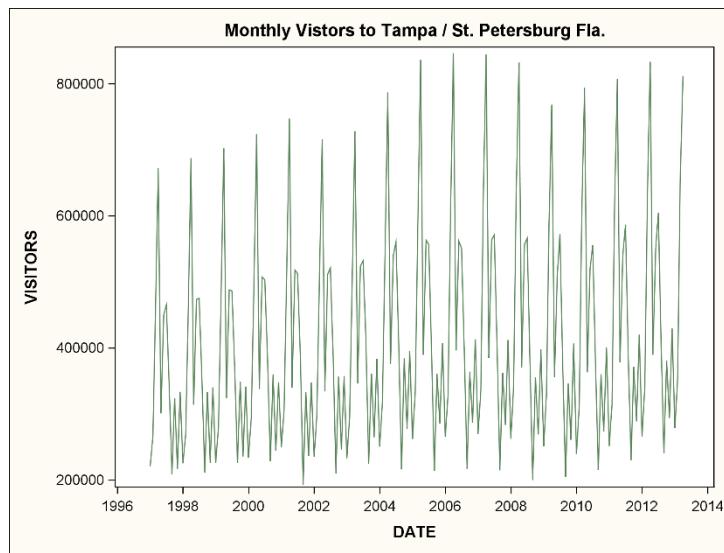
$$e_{4,t} \sim N(0, \sigma_4^2)$$

If $\sigma_4^2 = 0$, then the seasonality is an exactly periodic function as would be assumed when seasonal dummy variables are used in a regression. If $\sigma_4^2 > 0$, then the seasonality changes slowly over time. In the next section, a SEASON statement enables the user to estimate the seasonal coefficients and to decide whether $\sigma_4^2 = 0$. That is, the user decides whether the seasonality is exactly periodic. This idea, as well as ideas about diffuse initial values, follow the same logic as the previously discussed component.

7.3.2 Tourism Example with Regular Seasonality

Monthly visitor counts for the Tampa, Florida, and St. Petersburg, Florida, area are plotted in **Output 7.9**.

Output 7.9: Visitors to Tampa-St. Petersburg



There is a strong seasonal effect. It appears that the number of visitors has increased over time, but with occasional leveling off or declining as in 2006 through 2010. This might be modeled as a local level plus a local slope, a local level plus a deterministic slope, or just a local level with no slope. In preparation for what is about to come, some comments are in order. The irregular term in PROC UCM is specified in a statement rather than being assumed by default as it is, for example, in the ARIMA, REG, and GLM procedures. In these procedures, specifying a model with no error term would not be sensible. In UCMs, however, there are already error terms built in to the local level, slope, and seasonal components.

For example, if there is no seasonal, slope, or irregular term, the model becomes simply $\mu_{t+1} = 0 + \mu_t + e_{3,t+1}$ with $Y_t = \mu_t$. In other words, $Y_t = Y_{t-1} + e_{3,t}$, which is a random walk. The level component's innovation now serves as an error term. If an irregular term is added explicitly with an IRREGULAR statement, then $Y_t = \mu_t + e_{1,t}$. So, $Y_t - Y_{t-1} = e_{3,t} + e_{1,t} - e_{1,t-1}$. The first differences have covariance $-\sigma_1^2$ at lag 1 and 0 at higher lags so that the first differences now form a moving average of order 1. Adding the irregular component changed the model from a random walk to an ARIMA(0,1,1). The UCM procedure output includes the sum of all the components except the irregular. If no IRREGULAR statement is used, this sum reproduces the historical data.

The data were converted to units of thousands of visitors, variable V , to avoid extremely large numbers in the output. This code will fit a seasonal UCM model to the data with local level, slope, and seasonality, as well as an irregular term:

```
proc ucm data=stpete;
  id date interval=month;
  model v;
  level;
  slope;
  season length=12 type=dummy;
  irregular;
run;
```

Partial output, **Output 7.10**, suggests that the irregular term might be omitted. Recall that this does not mean that there are no random error terms in the model.

Output 7.10: First Analysis of Tourism Data

Final Estimates of the Free Parameters					
Component	Parameter	Estimate	Approx Std Error	t Value	Approx Pr > t
Irregular	Error Variance	0.00220	0.01707	0.13	0.8972
Level	Error Variance	79.69477	19.75965	4.03	<.0001
Slope	Error Variance	0.00000105	.	.	.
Season	Error Variance	32.51178	9.36988	3.47	0.0005

Significance Analysis of Components (Based on the Final State)			
Component	DF	Chi-Square	Pr > ChiSq
Irregular	1	0.00	0.9949
Level	1	4640.98	<.0001
Slope	1	0.57	0.4495
Season	11	7613.88	<.0001

The first thing to note is that neither the irregular component nor the slope component is local in nature. This conclusion is based on the fact that their variances, shown in the **Estimate** column, are very small, especially in relationship to 79.69 and 32.51. Their **Approx Pr>|t|** probabilities are much larger than 0.05 or missing. There is no evidence of an error term in either component where the presence of an error term would introduce a local nature to the component. Possibly because of the closeness of the slope error variance estimate to the lower bound 0 for variances, the standard error and test statistic of the slope variance are missing. Because the irregular component has assumed mean 0 and insignificant error variance ($t = 0.13$), the conclusion is to omit it.

The extremely small slope variance suggests a constant as well. The final state's slope component is not significantly different from 0, which, alone, would not usually suggest omission unless the slope was assumed constant. In other words, just because the local slope at the end of the observed data is 0 does not mean that all the local slopes should be

forced to 0 by omitting the SLOPE statement. Nevertheless, there is a suggestion that the slope might be constant and possibly even equal to 0. Therefore, a more careful approach was applied after removing the irregular term:

```
slope variance=0 noest;
```

The result was that the overall slope, when forced to be constant, was still not significantly different from 0. This validated the omission of the slope term as was initially suspected. This leaves a simple PROC UCM approach consisting of a local level and local seasonal pattern. Estimation is accomplished using the following code:

```
proc ucm data=stpete;
  id date interval=month;
  model v;
  level;
  season type=dummy length=12;
  forecast lead=24 out=out1;
run;
```

This code produces **Output 7.11**.

Output 7.11: Partial Results for Tourism Analysis

Final Estimates of the Free Parameters					
Component	Parameter	Estimate	Approx Std Error	t Value	Approx Pr > t
Level	Error Variance	78.72997	19.16377	4.11	<.0001
Season	Error Variance	32.82267	9.27960	3.54	0.0004

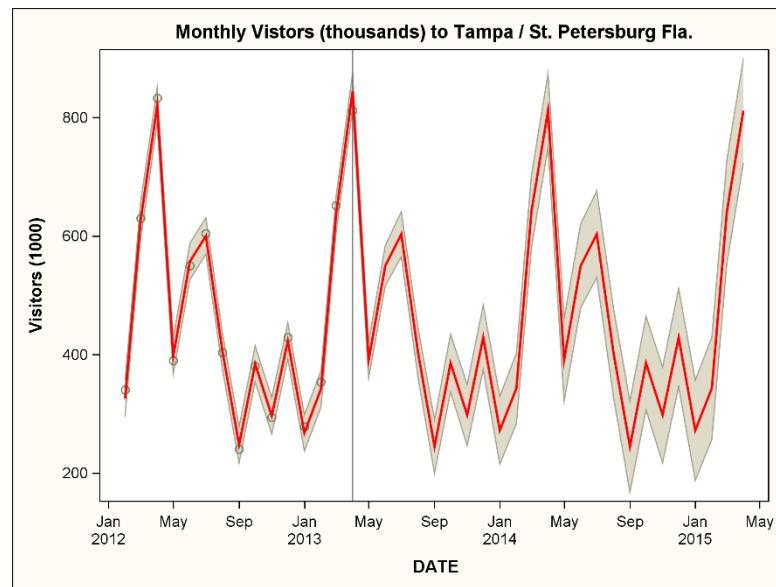
Significance Analysis of Components (Based on the Final State)			
Component	DF	Chi-Square	Pr > ChiSq
Level	1	4660.80	<.0001
Season	11	7662.65	<.0001

Trend Information (Based on the Final State)		
Name	Estimate	Standard Error
Level	447.8518087	6.560005

The table in **Output 7.11** shows that the error variances for the two components (**Level** and **Season**) are significantly larger than zero. The two components are both local in nature rather than constant. The significance analysis shows that, at the end of the series in the final state, there is a significantly nonzero level and significant seasonality. The local level is estimated as 447.85. Because the future errors in the level term are independent of the historical errors, the forecast will show seasonal variation around a level of 447.85 thousand visitors. For the same reason, the final state's local seasonal pattern serves as the seasonal pattern for the forecasts. After the series ends, the innovations (component error terms) are assumed to be 0 (0 being the mean of their distribution). The last 12 local seasonal components remain constant in the forecasts, producing an exactly periodic pattern around 447.85 for the forecasts. Because of the unit root nature of the components, the forecast standard errors will expand as the forecasts extend into the future.

The following code produces **Output 7.12** from the OUT= output data set specified in the FORECAST statement in PROC UCM. This can be used to generate a custom forecast graph.

```
proc sgplot data=out1 noautolegend;
  where date>"01JAN2012"d;
  title 'Monthly visitors (thousands) to Tampa / St. Petersburg Fla.';
  band lower=lcl upper=ucl x=date / fill outline;
  refline "01apr2013"d / axis = x;
  scatter y=v x=date;
  series y=forecast x=date/lineattrs=(color=red thickness=2);
run;
```

Output 7.12: Final Analysis of Tourism Data with Forecasts

Although the forecast error variances grow linearly without bound in such unit root models, the error standard deviations are small enough, relative to the seasonal swings in the series, that the forecast error bands appear impressively tight for the first couple of years of forecasts. Note the exactly periodic nature of the forecasts.

7.3.3 Decomposition

One appealing thing about the components type of model is the ability to look at the components individually, a task not easily available in PROC ARIMA, for example. The level component can be thought of as a type of seasonally adjusted series. Using the same previous output data set (OUT1) and the following code produces a graph of the level component, named S_LEVEL in the OUT1 data set:

```
%macro april;
%do yr = 1997 %to 2013;
  "01apr&yr"d
%end;
%mend;

options mprint; ** check log for the code generated by %april **;
proc sgplot data=out1 noautolegend;
  refline %april / axis = x;
  scatter y=v x=date;
  series y=s_level x=date/lineatrs=(color=red thickness=2);
run;
```

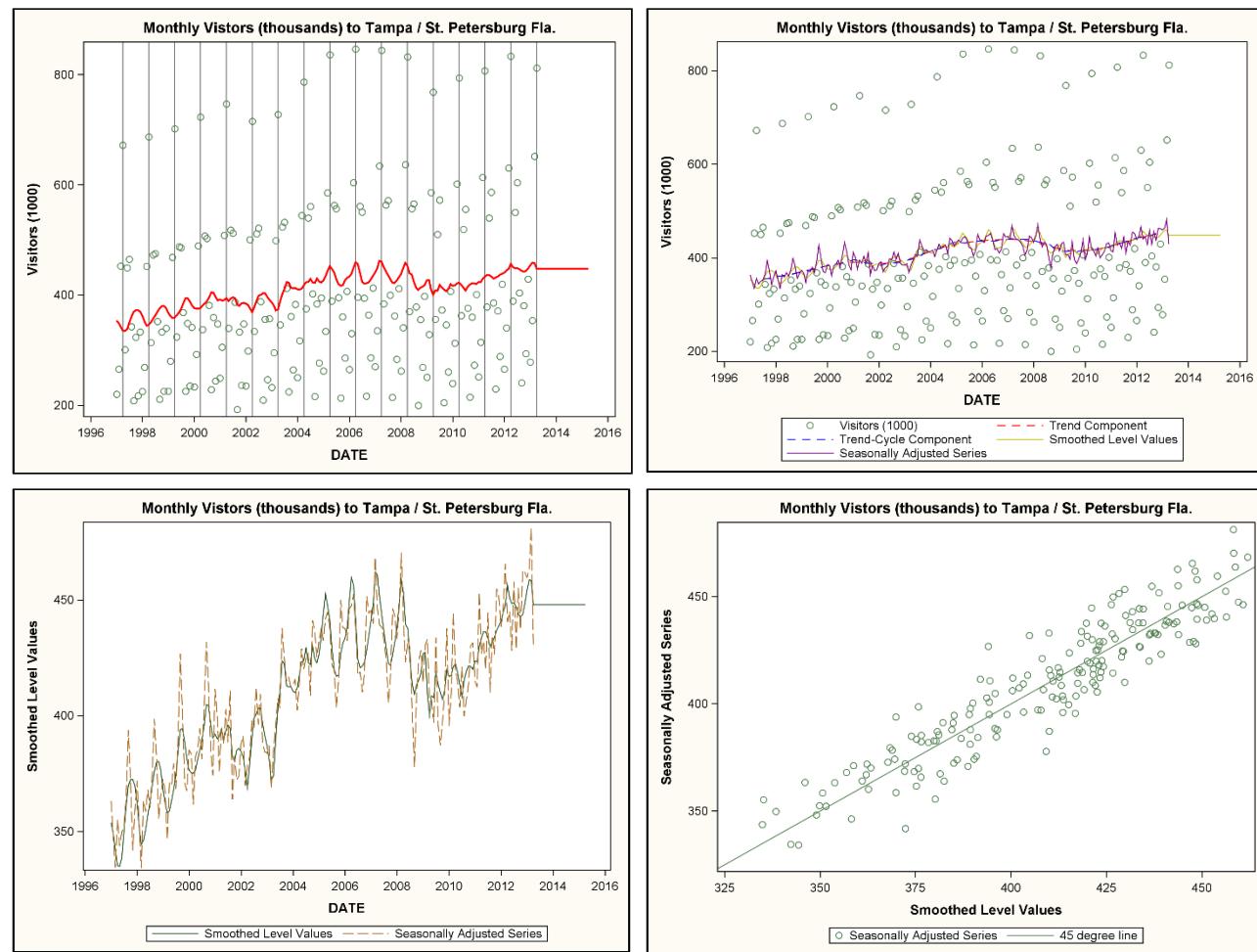
The %APRIL macro produces a set of character strings of the form “01APRxxxx”d, where each xxxx is a year from 1997 to 2013. This allows reference lines at each April. The resulting (resolved) code is displayed in the log window due to the MPRINT option. The resulting upper left panel of **Output 7.13** shows the local level component with the April reference lines. It seems visually that for a few years around 2005 through 2008, the local level has retained some influence of the high April visitor counts.

An alternative decomposition is given by PROC TIMESERIES, which uses classical decomposition methods to extract components. The components are an underlying trend component (TC), which is similar to the local level component; a trend-cycle component (TCC), which adds to TC any underlying business cycle that the procedure has detected; and a seasonally adjusted series (SA), which is the series with the seasonality removed, but still retaining the TCC and irregular components. These are put into the OUT2 data set by the OUTDECOMP statement. The TC and TCC components are almost indistinguishable smooth, dashed lines in the upper right panel of **Output 7.13**, which also includes a scatter plot of the original data, as well as the SA component from PROC TIMESERIES and the smoothed level series from PROC UCM. These are the irregular lines around the smoothed trend and are similar to each other. A plot with only the smoothed level and the seasonally adjusted series appears in the bottom left panel, where it is clear that both methods of seasonal adjustment show influences of the strong seasonality in 2005 through 2008, as well as simultaneous peaks and troughs elsewhere around the TC and TCC components.

Completing **Output 7.13** in the bottom right corner is a scatter plot of the SA series from PROC TIMESERIES against the smoothed level series. A 45-degree line indicates that they are similar, but not exactly the same. The three plots containing the smoothed level series are extended into the future as horizontal lines at a height based more on the recent series values than on the distant past values. Recall that the null hypothesis of 0 slope is not rejected, which is not the same as saying it is accepted. If a constant slope is fit, its estimate, 0.4848, produces a linearly increasing forecast that might appear more consistent with the last few years of data than does the constant level forecast. The associated *p*-value, *p* = 0.45, implies that such an increasing forecast must be justified on grounds other than statistical significance, such as visual appeal. The PROC TIMESERIES code to put the components into the data set OUT2 is shown. That data set is then merged by date with the OUT1 data set produced by PROC UCM to produce the four panels of **Output 7.13**.

```
proc timeseries data=stpete outdecomp=out2 seasonality=12;
  var v;
  decomp sa tc tcc;
  id date interval = month;
run;
```

Output 7.13: Components from PROCs UCM and TIMESERIES



7.3.4 Another Seasonal Model: Sine and Cosine Terms

Seasonal dummy variables are not the only way to model seasonality. A sinusoidal cycle component $\alpha \sin(\omega t + \delta)$ can be introduced recursively using the trigonometric formulas for the sum of two angles, namely $\sin(A + B) = \sin(A)\cos(B) + \cos(A)\sin(B)$. For cosine terms, $\cos(A + B) = \cos(A)\cos(B) - \sin(A)\sin(B)$. In time series terminology, the term *seasonal* implies periodic behavior where the period is a known quantity such as 12 months or 4 quarters. The term *cycle* is more general, for example, a diurnal cycle in biology or a business cycle. A cycle could have a seasonal period. Therefore, this methodology can be used for modeling seasonality or a general cycle. In $\alpha \sin(\omega t + \delta)$, the amplitude is represented by α ,

the frequency by ω , and δ is known as the *phase shift*. If the frequency ω is known, then $\sin(\omega t)$ and $\cos(\omega t)$ are also known. Cycle can be expressed as follows:

$$\alpha \sin(\omega t + \delta) = [\alpha \cos(\delta)]\sin(\omega t) + [\alpha \sin(\delta)]\cos(\omega t)$$

This is the sum of two known predictor variables, each multiplied by an unknown coefficient. A regression on $\sin(\omega t)$ and $\cos(\omega t)$ would estimate the cycle in PROC REG where the two regression coefficients would be estimates of the quantities in square brackets [] in the previous displayed equation. To express the two input variables recursively, again apply the sum of angles formulas:

$$\begin{pmatrix} \sin(\omega(t+1)) \\ \cos(\omega(t+1)) \end{pmatrix} = \begin{pmatrix} \cos(\omega) & \sin(\omega) \\ -\sin(\omega) & \cos(\omega) \end{pmatrix} \begin{pmatrix} \sin(\omega t) \\ \cos(\omega t) \end{pmatrix}$$

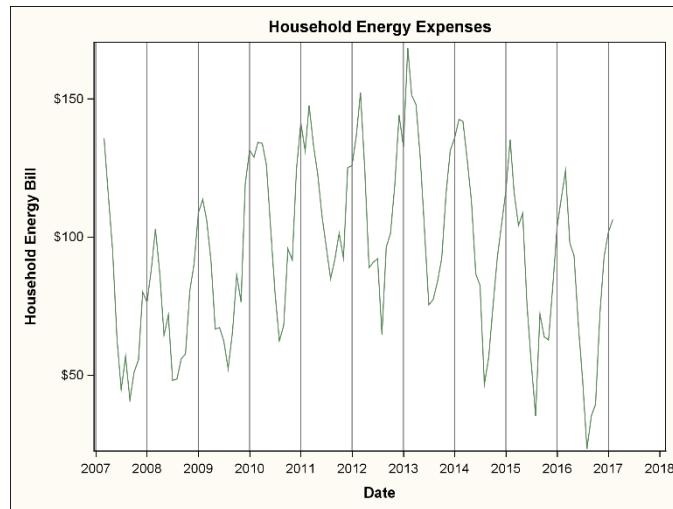
7.3.5 Example with Trigonometric Components

Suppose a homeowner records utility bills monthly and notices what appears to be periodic behavior. A graph of the data produced by the following code is in **Output 7.14a**.

```
* create reference lines ;
%macro ref;
  %do i=2007 %to 2017;
    "01Jan&i."d
  %end;
%mend;

proc sgplot data=household;
  Title "Household Energy Expenses";
  series X=date Y=amount;
  reftime %ref /axis=X;
run;
```

Output 7.14a: Household Energy Bills with Reference Lines for each December



The seasonality appears regular with a single peak and trough each year. This suggests a trigonometric component, possibly with 0 error variance. That is, it might be regular enough that a deterministic sinusoidal component is suffice. This can be tested. The pattern seems similar to a single sine wave, suggesting that a sinusoidal component using only the fundamental period of 12 months might suffice, as opposed to adding cycles of periods 6, 4, 3 and so on, called *harmonics*. Harmonics leave the overall periodicity at 12, but change the shape of seasonal pattern. There are 6 harmonics in all, each being at period $12/j$, where $j = 1, 2, \dots, 6$, and where $j = 1$ gives the fundamental period, the overall period of the combined sinusoids.

The level of the series varies slowly as well, suggesting a level component with nonzero error variance. The general appearance of the level might suggest a quadratic deterministic trend as an alternative. However, this would force the forecasts to decrease over time. Without additional information to suggest a continually decreasing trend, such a

component is difficult to justify and is not pursued. A model with local level and possibly local trigonometric seasonal pattern is fit using the following code:

```
proc ucm plots=(all) data=household;
  model amount;
  level;
  season length=12 type=trig dropharmonics=2 to 6;
  irregular;
  forecast lead=60;
run;
```

The TYPE=TRIG option invokes the fitting of sinusoids and, by default, includes the full set of harmonics. The DROPHARMONICS option omits all but the fundamental period sinusoid.

Output 7.14b: Significance Check with Seasonal Error Variance Estimated

Final Estimates of the Free Parameters					
Component	Parameter	Estimate	Approx Std Error	t Value	Approx Pr > t
Irregular	Error Variance	68.31425	13.04022	5.24	<.0001
Level	Error Variance	24.78905	9.72176	2.55	0.0108
Season	Error Variance	0.05993	0.23790	0.25	0.8011

The seasonal component's error variance is insignificant ($p = 0.80$). The seasonal component can be made deterministic (global, rather than local) by initializing its error variance to 0 and using a NOEST option to hold it at 0. The resulting SEASON statement is:

```
season length=12 type=trig dropharmonics=2 to 6 variance=0 noest;
```

The relevant output is in **Output 7.14c**.

Output 7.14c: Significance Check with Seasonal Error Variance Held at 0

Final Estimates of the Free Parameters					
Component	Parameter	Estimate	Approx Std Error	t Value	Approx Pr > t
Irregular	Error Variance	68.20599	13.14561	5.19	<.0001
Level	Error Variance	25.56317	9.65173	2.65	0.0081

Significance tests of the final component values are listed in **Output 7.14d**. The last irregular term is not unusual in that it does not differ significantly from 0. The **Level** and **Season** components differ significantly from 0. The **Season** component has 2 degrees of freedom because the sinusoid consists of a sine and a cosine term. Following these tests is a table with the final estimate of the level, indicating that the forecasts varies around a level of 73.70.

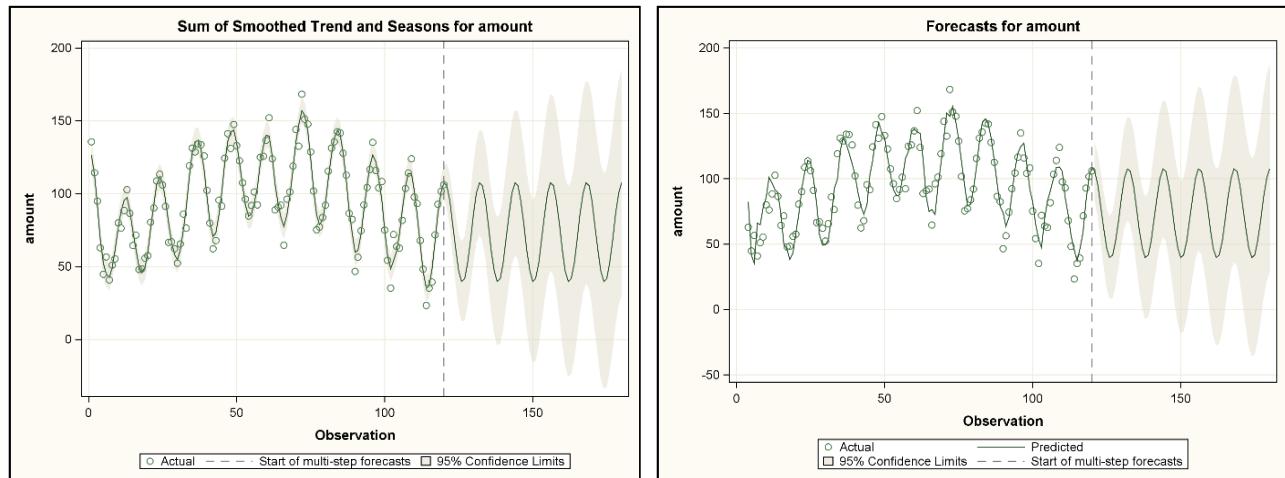
Output 7.14d: Some Output Tables for a Seasonal UCM

Significance Analysis of Components (Based on the Final State)			
Component	DF	Chi-Square	Pr > ChiSq
Irregular	1	0.06	0.8030
Level	1	167.12	<.0001
Season	2	431.64	<.0001

Trend Information (Based on the Final State)		
Name	Estimate	Standard Error
Level	73.70034692	5.7010332

The PLOTS=(ALL) option produces a large array of plots, among which the plots of **Output 7.14e** are of particular interest.

Output 7.14e: Two of Many Plots Produced by the PLOTS=(ALL) Option



The plot on the left is the sum of all filtered components other than the irregular component. These are extended 60 periods (five years) into the future with appropriate prediction intervals. Recall that filtered components give one-step-ahead forecasts and intervals through the historical data. When forecasting future observations, as in the plot to the right, the variance of the irregular term must be incorporated. Although not dramatic in this example, this results in a widening of the forecast intervals. With only a LEVEL statement to model trend, the forecast varies around the final level 73.07. The rapidly expanding prediction intervals are typical of unit root components. Models beyond the basic structural model such as this one can be, at best, difficult to fit with programs that run only ARIMA models.

The example in this section is not real data. It was generated using precisely the type of model just fit. It is interesting that this model produced a plot with a quadratic appearance and that the UCM procedure correctly diagnosed the seasonal error variance as 0 and the others as nonzero, just as in the data-generating mechanism.

7.3.6 The Seasonal Component Made Local and Damped

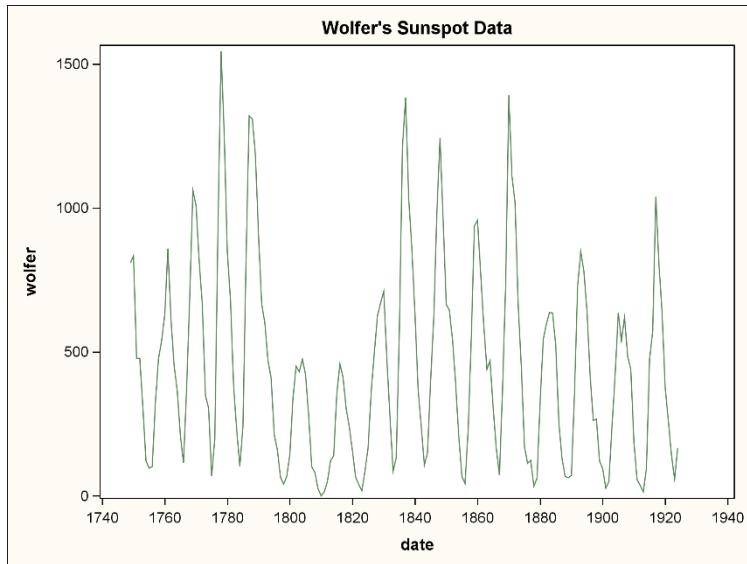
In time series terminology, a cycle indicates a general, often unknown, period. The word “seasonal” usually implies a known periodicity on which most people would agree. For example, the periodicity for monthly data is 12. Daily data might have a period 7 seasonal pattern when there is a day of the week effect. On the other hand, cycles such as a business cycle do not have a commonly agreed-on period. A well-known example is the so-called “Wolfer sunspot” numbers. These numbers (counts) historically have peaks of activity that are about 9 to 11 years apart. The peaks are not of uniform height.

To make the cycle local, a bivariate vector of independent errors is added to the previous matrix recursion. Because with the added errors the seasonal is local, it might help in some cases to allow the seasonality to decay over time at some rate ρ (the *damping factor*), rather than simply copying the final seasonal component pattern repeatedly into the future to get the forecast. After all, if seasonality is local, it makes sense that the timing of peaks and troughs might be less predictable as time goes by, so damped seasonal components would be appropriate. With these modifications, the general form of a cycle component, ignoring amplitude, is as follows:

$$\begin{pmatrix} \sin(\omega(t+1)) \\ \cos(\omega(t+1)) \end{pmatrix} = \rho \begin{pmatrix} \cos(\omega) & \sin(\omega) \\ -\sin(\omega) & \cos(\omega) \end{pmatrix} \begin{pmatrix} \sin(\omega t) \\ \cos(\omega t) \end{pmatrix} + \begin{pmatrix} e_{4,t+1} \\ e_{5,t+1} \end{pmatrix}$$

The Wolfer sunspot series is used in the SAS documentation for PROC SPECTRA. Sunspots have been linked to various weather patterns on earth. **Output 7.15** shows the yearly data in which an approximate 11-year cycle with apparently changing amplitude over time is seen. An exactly periodic function is clearly inappropriate here.

Output 7.15: Wolfer Sunspot Series



Initially, a UCM model with LEVEL, SLOPE, IRREGULAR, and CYCLE statements was fit. Perhaps not surprisingly, the slope term appeared unnecessary, as did the irregular term. Omitting these one at a time left this PROC UCM code:

```
proc ucm data=sunspot;
  id date interval=year;
  model wolfer;
  level;
  cycle period=11 plot=smooth;
  forecast lead=44 plots=decomp print=decomp out=out1;
run;
```

Output 7.16 shows partial results. Note the use of a CYCLE statement, rather than a SEASON statement, to handle the periodicity in the data.

Output 7.16: Sunspot Analysis Results

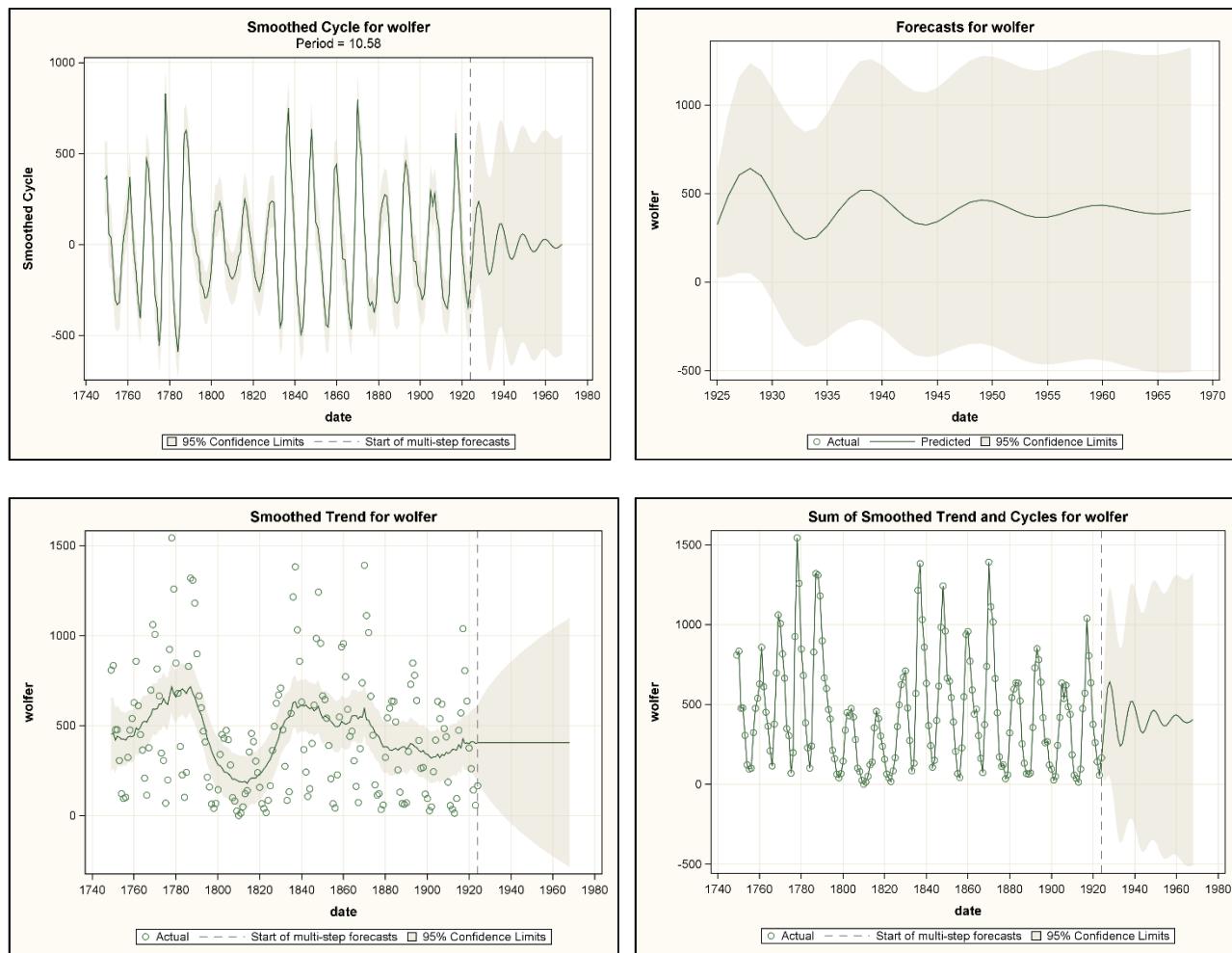
Final Estimates of the Free Parameters					
Component	Parameter	Estimate	Approx Std Error	t Value	Approx Pr > t
Level	Error Variance	2576.40098	1174.7	2.19	0.0283
Cycle	Damping Factor	0.93606	0.01955	47.89	<.0001
Cycle	Period	10.58312	0.41233	25.67	<.0001
Cycle	Error Variance	11677	3354.3	3.48	0.0005

Significance Analysis of Components (Based on the Final State)			
Component	DF	Chi-Square	Pr > ChiSq
Level	1	14.79	0.0001
Cycle	2	5.45	0.0655

Trend Information (Based on the Final State)		
Name	Estimate	Standard Error
Level	405.5843438	105.46693

The first table shows significant error variances for the **Level** and **Cycle** components. The damping factor rho (ρ) is significantly different from 0 and, more interestingly, it is $t = (1 - 0.93606)/0.01955 = 3.27$ standard errors less than 1. This is consistent with the idea that damping is appropriate. Statistical theory does not justify a stronger term than “consistent.” The hypothesis that $\rho = 1$ makes the seasonal recursion a bivariate unit root process. There is no current theoretical justification for comparing $t = 3.27$ to a known distribution to get a p -value. The cyclical period, initialized at 11 years, becomes 10.58 in the final analysis, only about 1 standard error below 11. The PLOTS=DECOMP option in the FORECAST statement and the PLOT=SMOOTH option in the CYCLE statement produce four nice summary plots as shown in **Output 7.17**. No additional graphics code is needed.

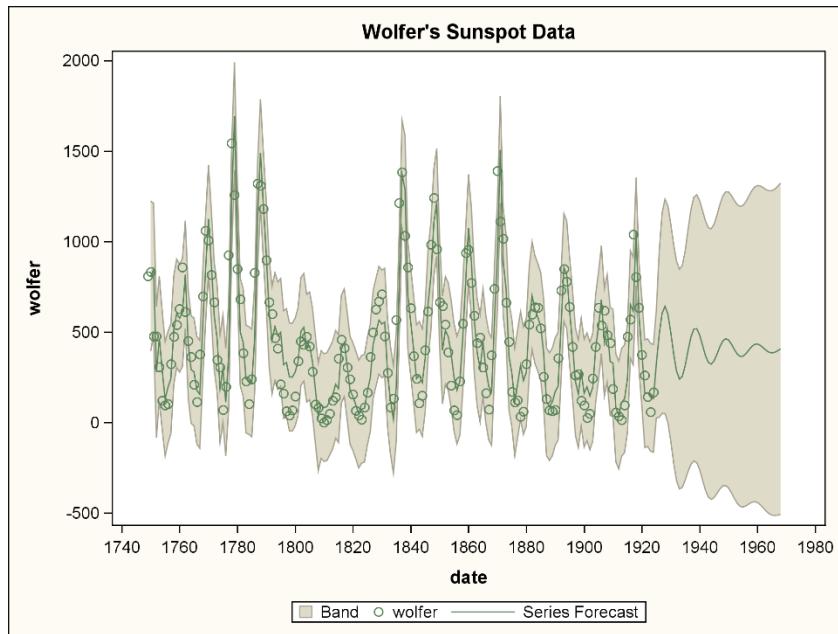
Output 7.17: Plots Produced by the UCM Procedure for the Damped Seasonal Model



The smoothed trend plot in the bottom left shows the driftless random walk nature of the trend as specified by the PROC UCM code. Its forecast is constant, and the error variance seems to be increasing without bound. Because there is no irregular component, the model's smoothed components add up to the historical data points as shown in the bottom right plot. In other words, the sum of the smoothed components at time t takes into account all information up through time t , thus reproducing the observation itself. Because the cyclical periods are only approximately 11 years on average and differ from cycle to cycle, it is reasonable to have the damping shown in the bottom right panel forecasts. Uncertainty about where the peaks and troughs are and their amplitudes increases as the lead time increases.

When forecasts into the future are computed, all future error terms, as always, are replaced by 0s. **Output 7.18** shows one-step-ahead predictions in the historical data and forecasts into the future.

Output 7.18: UCM Forecasts for Sunspot Data

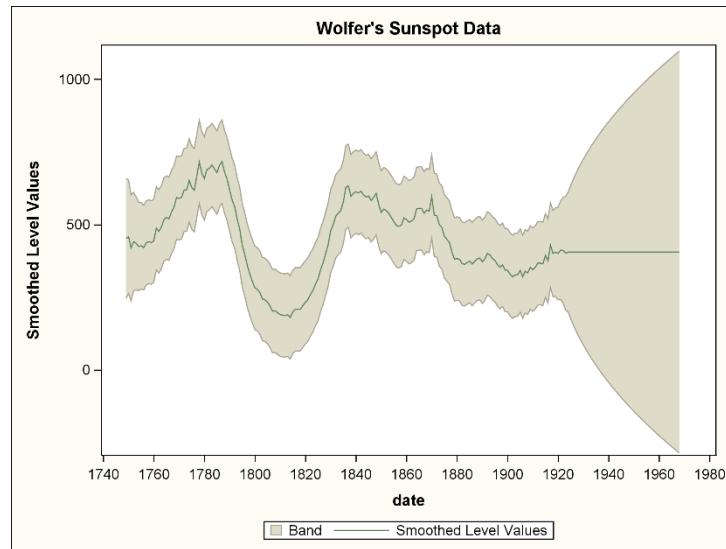


Forecast error bands must account for the fact that future unknown errors are set to 0. In **Output 7.18**, the one-step-ahead forecasts through the historical data have nontrivial one-step-ahead forecast error bands. For example, the innovation (error term) in the local level series at time $t+1$ is unknown at time t . It is uncorrelated with any information available at time t . This must be accounted for in the one-step-ahead error band, which has nonzero width even though the model did not include an IRREGULAR statement. After one step beyond the observed data, the forecast error bands increase in width due to their multi-step nature.

The sunspot OUT1 data set has the smoothed level, S_LEVEL, and its variance, VS_LEVEL, from which a deseasonalized series with a (pointwise) confidence band is computed in a DATA step and subsequently plotted. **Output 7.19** shows a graph. The code to generate **Output 7.19** is the following:

```
data smoothlevel;
  set out1;
  low_sl = s_level - 1.96*sqrt(vs_level);
  high_sl = s_level + 1.96*sqrt(vs_level);

proc sgplot data=smoothlevel;
  band lower=low_sl upper=high_sl x=date;
  series y=s_level x=date;
  series y=low_sl x=date;
  series y=high_sl x=date;
run;
```

Output 7.19: Smoothed Level Component of Sunspot Counts

Seasonal period and damping factors can be fixed at their initial values. For example, in the previous household energy usage data shown in **Output 7.14a**, this can be done by replacing the SEASON statement with the following code, thereby estimating the same model as discussed in that section:

```
cycle period=12 rho=1 noest(period rho);
```

7.4 A Brief Introduction to the SSM Procedure

The general statespace model was introduced in the first part of Section 7.1.1. A more flexible procedure than PROC UCM is the StateSpace Modeling procedure, PROC SSM. It can be thought of as an extension of PROC UCM that uses the full statespace functionality.

7.4.1 Brief Overview

The SSM procedure is a more general recursive procedure than the UCM procedure, but it is also based on the idea of a state space model. The STATESPACE procedure handles a subset of the models that PROC SSM handles. The new SSM procedure handles an extremely wide array of univariate and multivariate dynamic linear models, typically including some unobserved components handled by the state vector. Some simple examples are used in this section to introduce the idea, and then the state space model formulation and some more interesting examples are shown.

7.4.2 Simple Examples

One of the simplest of the ARIMA models is the autoregressive order 1 model, $Y_{t+1} - \mu = \rho(Y_t - \mu) + e_{t+1}$, where $|\rho| < 1$. An algebraic rearrangement of terms gives $Y_{t+1} = (1 - \rho)\mu + \rho Y_t + e_{t+1}$, which in turn shows that Y_{t+1} can be expressed as a sum of three components, namely the constant $\lambda_0 = (1 - \rho)\mu$ plus ρY_t plus a white noise term. The idea, just as in UCM, is to express the model's components as recursions. The first term can be expressed recursively simply as $\lambda_{t+1} = \lambda_t$ so that the value of all λ s is set by the initial value λ_0 . The first two components form a simple vector valued recursive relationship summarized in a *transition equation*:

$$\begin{pmatrix} Y_{t+1} \\ \lambda_{t+1} \end{pmatrix} = \begin{pmatrix} \rho & 1 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} Y_t \\ \lambda_t \end{pmatrix} + \begin{pmatrix} \eta_{1,t+1} \\ 0 \end{pmatrix}$$

In the state space model notation, the 2×2 coefficient matrix with three structural parameters and one parameter, ρ , to be estimated is called the *transition matrix* and symbolized \mathbf{T}_t . The transition equation's error vector, $\boldsymbol{\eta}_{t+1}$, is assumed to form an independent normal white noise sequence with mean vector 0 and variance matrix \mathbf{Q}_t . In this case, \mathbf{Q}_t is a 2×2 matrix with just one nonzero element in the upper left corner. \mathbf{Q}_t can be any valid covariance matrix. It can handle correlated error terms. The first row of the matrix equation is the entire model, with the second ensuring the constancy of λ_t . The column vector on the left is called the *state vector*, symbolized \mathbf{a}_{t+1} , and its lag \mathbf{a}_t appears on the right, thus the

name *transition equation*. Recall from **section 7.1.1** that the complete transition equation in PROC SSM notation, is as follows:

$$\alpha_{t+1} = \mathbf{T}_t \alpha_t + \mathbf{W}_{t+1} \gamma + \mathbf{c}_{t+1} + \eta_{t+1}$$

In this case, it reduces to $\alpha_{t+1} = \mathbf{T}_t \alpha_t + \eta_{t+1}$. Here, the transition matrix \mathbf{T} is constant over time. $\mathbf{W}_{t+1} \gamma$ and \mathbf{c}_{t+1} are zero vectors. If not omitted, $\mathbf{W}_{t+1} \gamma$ allows for a regression input with coefficient vector γ in the state vector α_t . The \mathbf{c}_{t+1} allows for a known deterministic input vector. Note the similarity of the transition equation to a vector AR(1) process. A process such as this, in which a vector depends on only 1 lag of itself, is typically called a *Markov process* in statistics and mathematics.

In this simple case, the first element of the state vector is the future response whose value is to be predicted. The observation equation selects the state vector's first element:

$$Y_{t+1} = (1 \ 0) \begin{pmatrix} Y_{t+1} \\ \lambda_{t+1} \end{pmatrix} = \mathbf{Z} \alpha_{t+1}$$

This equation defines a matrix \mathbf{Z} that specifies a component based on the state vector. The basic idea of state space models (SSM) and unobserved components (UCM) models is to mimic and expand this structure to more complex situations. The full-blown observation equation, as with the transition equation, contains additional terms and is given by $\mathbf{Y}_{t+1} = \mathbf{Z}_{t+1} \alpha_{t+1} + \mathbf{X}_{t+1} \beta + \varepsilon_{t+1}$. This allows deterministic regressor terms and a random normal error term, $\mathbf{X}_{t+1} \beta$ and ε_{t+1} , respectively, where ε_{t+1} forms an independent normal vector sequence with diagonal variance matrix \mathbf{G} . These affect the observed variables rather than the state vector. The error terms in the transition equation and observation equation are independent of each other. Only the transition equation error allows correlation in its error vector's random elements.

7.4.3 Extensions of the AR(1) Model

Two more comments are relevant. First, with a recursion, the question of how to get started always arises. The use of a diffuse likelihood enables you to be indifferent about the starting values for parameters such as the intercept λ . The basic idea is to write the likelihood with a prior distribution for λ , modify it so that a limit can be taken, and then take that limit as the variance of the normal prior increases without bound. This is accomplished through a third equation, the initialization equation. Second, only components defined in COMPONENT statements are eligible for inclusion in the model's state vector. These define the nature of $\mathbf{Z}_{t+1} \alpha_{t+1}$ where, in the AR(1) example, the \mathbf{Z} matrix, $\mathbf{Z} = (1 \ 0)$, is assumed to be constant over time.

One simple expansion that yields another familiar model, the random walk with ($j = 1$) or without ($j = 0$) drift, is obtained by allowing $\rho = 1$, thus arriving at this pair of equations:

$$\begin{pmatrix} Y_{t+1} \\ \lambda_{t+1} \end{pmatrix} = \begin{pmatrix} 1 & j \\ 0 & 1 \end{pmatrix} \begin{pmatrix} Y_t \\ \lambda_t \end{pmatrix} + \begin{pmatrix} e_{1,t+1} \\ 0 \end{pmatrix}$$

The second row holds λ constant at its initial value. And, as before,

$$Y_{t+1} = (1 \ 0) \begin{pmatrix} Y_{t+1} \\ \lambda_{t+1} \end{pmatrix}$$

With no drift, the simple random walk recursion $Y_{t+1} = Y_t + e_{t+1}$ suffices as a one-dimensional transition equation.

A third example is a slight generalization of the exponential smoothing model, this being equivalent to a model whose first difference is a moving average. The model, put into state space form via another pair of recursive equations, is as follows:

$$\begin{pmatrix} Y_{t+1} \\ \lambda_{t+1} \\ e_{1,t+1} \end{pmatrix} = \begin{pmatrix} 1 & 0 & -\theta \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} Y_t \\ \lambda_t \\ e_{1,t} \end{pmatrix} + \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} e_{1,t+1} \\ 0 \\ 0 \end{pmatrix}$$

If (as in the usual exponential smoothing case) no drift term is involved, then the initial value of λ is set to 0, the middle term of the state vector is no longer needed, and the recursion simplifies to the following expression:

$$\begin{pmatrix} Y_{t+1} \\ e_{1,t+1} \end{pmatrix} = \begin{pmatrix} 1 & -\theta \\ 0 & 0 \end{pmatrix} \begin{pmatrix} Y_t \\ e_{1,t} \end{pmatrix} + \begin{pmatrix} 1 \\ 1 \end{pmatrix} (e_{1,t+1})$$

The error term includes a multiplier—a vector of 1s. This is accommodated by structuring the \mathbf{Q} variance matrix as follows:

$$\mathbf{Q} = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \sigma^2 \begin{pmatrix} 1 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} \sigma^2$$

That is, multipliers in the state transition equation's error term can be implicitly accommodated through the specification of \mathbf{Q} . The error term can be written without a multiplier. The first row of the state transition equation is the model, and the second is just an identity. The measurement equation is (assuming no drift) the following:

$$Y_{t+1} = (1 \ 0) \begin{pmatrix} Y_{t+1} \\ e_{t+1} \end{pmatrix}$$

The exact forms written here have many equivalent representations, which appear in the literature. The previous expressions are an attempt to make the models as familiar as possible.

The observation equation can include a random error term. A fourth example illustrates this. Adding a white noise series $e_{2,t}$ to an AR(1) model, $(X_t - \mu) = \rho(X_{t-1} - \mu) + e_{1,t}$, where the two white noise series are independent of each other, gives an observed series $Y_t = X_t + e_{2,t}$. Or, in deviations form, $(Y_t - \mu) = (X_t - \mu) + e_{2,t}$. Thus, X_t becomes a time-varying input to Y_t . Notice that $Y_t - \mu - \rho(Y_{t-1} - \mu) = (X_t - \mu) - \rho(X_{t-1} - \mu) + e_{2,t} - \rho e_{2,t-1}$. From the X_t model, this becomes $Y_t - \mu - \rho(Y_{t-1} - \mu) = e_{1,t} + e_{2,t} - \rho e_{2,t-1}$. The right side has nonzero variance and nonzero covariance at lag 1. But at lag 2 and higher, the covariance and correlation are 0. This covariance structure defines a MA(1) model for $Y_t - \mu - \rho(Y_{t-1} - \mu)$ and an ARMA(1,1) model for Y_t . This, in turn, shows that an autoregressive order 1 series, AR(1), with added white noise, becomes an ARMA(1,1) process. Adding white noise to the state space model transition equation for an AR(1) gives this pair of transition and measurement equations, where $x_t = X_t - \mu$ and $y_t = Y_t - \mu$:

$$\begin{aligned} \begin{pmatrix} x_{t+1} \\ \lambda_{t+1} \end{pmatrix} &= \begin{pmatrix} \rho & 1 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x_t \\ \lambda_t \end{pmatrix} + \begin{pmatrix} e_{1,t+1} \\ 0 \end{pmatrix} \\ y_{t+1} &= (1 \ 0) \begin{pmatrix} x_{t+1} \\ \lambda_{t+1} \end{pmatrix} + \begin{pmatrix} e_{2,t} \\ 0 \end{pmatrix} \end{aligned}$$

With x and y being deviations, the λ component could be omitted for simplicity, leading to the univariate transition equation $x_{t+1} = \rho x_t + e_{1,t+1}$ and observation equation $y_{t+1} = x_{t+1} + e_{2,t}$. In this example, X might be the true speed of a car and Y might be the speedometer reading or a radar reading or an estimated speed from an observer. It might be of more interest to extract an estimate of X from the series of Y observations (that is, to compute the state vector than to forecast Y). This shows an advantage of the state space approach, where it is a component of the state vector that is of interest.

7.4.4 Accommodation for Curvature

A fifth example is a quadratic regression with white noise errors. The components $\zeta_{t+1} = \zeta_t + e_{4,t}$, $\beta_{t+1} = \beta_t + \zeta_t + e_{3,t}$, and $\mu_{t+1} = \mu_t + \beta_t + e_{2,t}$, with fixed starting values ζ_0 , β_0 , and μ_0 , produce constant, linear, and quadratic expected values at time t given by the following:

$$\zeta_0, \quad \beta_0 + t\zeta_0, \quad \mu_0 + \sum_{j=1}^t (\beta_0 + j\zeta_0) = \mu_0 + t\beta_0 + \frac{t(t+1)}{2}\zeta_0 = \mu_0 + (\beta_0 + \zeta_0/2)t + (\zeta_0/2)t^2$$

Each is accompanied by random deviations with increasingly many unit roots (unless the error variances have been set to 0). Consider a transition equation of the following form:

$$\begin{pmatrix} \mu_{t+1} \\ \beta_{t+1} \\ \zeta_{t+1} \end{pmatrix} = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \mu_t \\ \beta_t \\ \zeta_t \end{pmatrix} + \begin{pmatrix} e_{2,t} \\ e_{3,t} \\ e_{4,t} \end{pmatrix}$$

Add a measurement equation:

$$Y_t = (1 \ 0 \ 0) \begin{pmatrix} \mu_t \\ \beta_t \\ \zeta_t \end{pmatrix} + e_{1,t}$$

This gives an overall or local quadratic, depending on which variances are set to 0, with white noise error $e_{1,t}$, where each e_t series is a white noise, and they are independent of each other.

For the case of a global (rather than local) quadratic, it is easier to compute the linear and quadratic predictors within the SSM procedure. This procedure allows most DATA step programming statements. The following example with generated data illustrates the method:

```
data a;
  do t=1 to 100;
    y = 100 - .01*(t-20)**2 + 4*normal(123);
    output;
  end;

proc ssm data=a;
  id t;
  parms eps / lower=(1.e-8);

  * data step programming allowed;
  one=1; tsq=t*t;

  * state vector has regression parameters-use identity t matrix, t(i);
  state qtrend(3) t(i) a1(3) print=(t cov);
  component zalpha = (one t tsq)*qtrend;
  irregular wn;
  component mu    = qtrend[1]/print=(smooth); * just to look at;
  component slope = qtrend[2]/print=(smooth); * just to look at;
  component curve = qtrend[3]/print=(smooth); * just to look at;
  model y = zalpha wn;
  output out=quadout;
run;

proc print data=quadout(obs=5);
  var forecast;
  run;
proc print data=quadout(obs=5);
  var smoothed_y smoothed_zalpha smoothed_mu smoothed_slope smoothed_curve;
run;

proc sgplot data=quadout noautolegend;
  series x=t y=y;
  series x=t y=forecast_zalpha;
  series x=t y=smoothed_zalpha;
  refline 4/axis=x;
  title "QUADRATIC REGRESSION EXAMPLE";
  title2 "FIRST THREE OBSERVATIONS ESTIMATE QUADRATIC";
  title3 "THEREFORE FIRST FORECAST IS AT T=4";
run;
```

The STATE statement gives the nature of the state transition equation. The STATE statement implies a state vector, called **Qtrend**, of dimension 3; an identity transition matrix **T(I)**; and three diffuse starting values **A1(3)**. In the COMPONENT statement, the first three elements of the state vector are constant, linear, and quadratic coefficients. The MODEL statement contains **Zalpha** but not **Qtrend**, illustrating that it is the COMPONENT statement, not the STATE

statement, that describes how the state vector enters the model. Other components not used in the MODEL statement can be computed just to look at, as described in the code, but not graphed. As a check on the specification in the STATE statement, the PRINT = (T COV) results are shown in **Output 7.20**. The transition matrix is an identity as requested. There is no disturbance added to the state vector in the transition step. The state vector just contains 1, t , and t^2 at time t .

Output 7.20: Transition and Disturbance Covariance Matrices

Transition Matrix for qtrend			
	Col1	Col2	Col3
Row1	1	0	0
Row2	0	1	0
Row3	0	0	1

Disturbance Covariance for qtrend			
	Col1	Col2	Col3
Row1	0	0	0
Row2	0	0	0
Row3	0	0	0

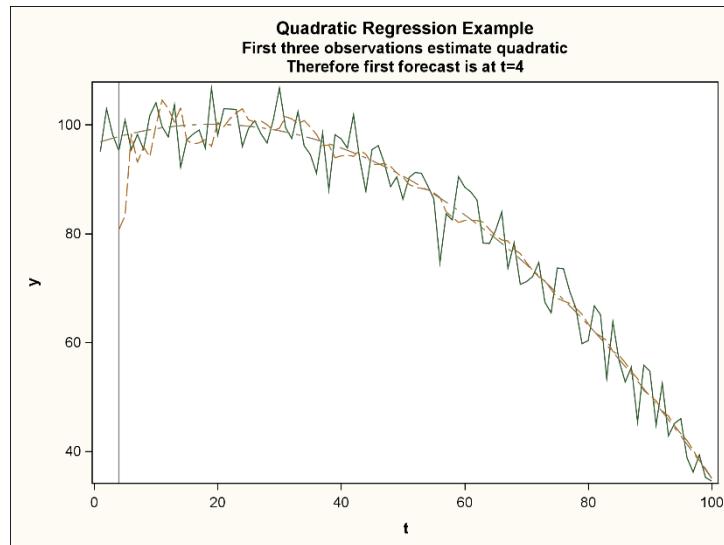
Only after three observations are read is it possible to fit a quadratic. The first forward-looking forecast is for observation 4. This results from the use of the diffuse prior. On the other hand, the smoothed series is based on all the data. This is the difference between smoothed and forecast series, a few observations of which appear in **Output 7.21**. As each new data point comes in, the mu, slope, and curve estimates change as they are based on more data. Once the full data are available, estimates based on full data are used to smooth the series at all times. The smoothed series shows constant mu, slope, and curve components based on full data.

Output 7.21: Data Sets with Forecast and Smoothed Series

Obs	FORECAST_Y	FORECAST_ZALPHA	FORECAST_MU	FORECAST_SLOPE	FORECAST_CURVE
1
2
3
4	80.7327	80.7327	74.6078	26.7907	-6.31488
5	83.2784	83.2784	85.4707	13.0310	-2.69389

Obs	Smoothed_Y	Smoothed_ZALPHA	Smoothed_MU	Smoothed_SLOPE	Smoothed_CURVE
1	95.084	96.8490	96.4807	0.37824	-0.009927217
2	102.930	97.1975	96.4807	0.37824	-0.009927217
3	98.146	97.5261	96.4807	0.37824	-0.009927217
4	95.217	97.8348	96.4807	0.37824	-0.009927217
5	100.865	98.1237	96.4807	0.37824	-0.009927217

Output 7.22 shows the results from the first SG PLOT.

Output 7.22: Data (Quadratic Plus Noise), Smoothed Series (Quadratic), and Forecast Series for SSM

The data appear to have constant variance through the time interval. The smoothed series, being based on full data, plots as a quadratic curve through time, although the forecast series (dashed line) varies around the quadratic at first, but by time 60, it is almost indistinguishable from the smoothed series. The PRINT=(SMOOTH) option on each of the mu, slope, and curve COMPONENT statements in the program produces a table of smoothed estimates with as many rows as there are observations (100). Because the components are smoothed as opposed to forecasted, they are based on the full data. Each table has identical rows in this simple example. Estimates for mu (96.5), slope (0.378), and curve ($-0.0099t^2$), each appearing repeatedly in its table, imply that the quadratic is $96.5 + 0.378t - 0.0099t^2$.

7.4.5 Models with Several Lags

Even though the state transition equation has just one lag, it is possible to fit, for example, an AR(3) model using a mean estimate and model deviations from it that satisfy a transition equation. The model formulated in the SSM is a result of Akaike's formulation (1976).

One way (see Akaike 1974, 1976) to write the state vector in an ARIMA model has the current value Y_t in the first entry, plus enough forecasts into the future so that every forecast into the future, as far as you like, is a linear combination of the state vector's entries. The notation $Y_{t+L|t}$ is then used as a notation for the L -step-ahead forecast from data up through time t . For the AR(3) model here, the standard way to forecast gives $Y_{t+1|t} = \phi_1 Y_t + \phi_2 Y_{t-1} + \phi_3 Y_{t-2}$, where everything on the right is known at time t . For the two-step-ahead forecast, because Y_{t+1} is unknown at time t , its forecast is substituted, giving $Y_{t+2|t} = \phi_1 Y_{t+1|t} + \phi_2 Y_t + \phi_3 Y_{t-1}$. In the same fashion, it follows that $Y_{t+3|t} = \phi_1 Y_{t+2|t} + \phi_2 Y_{t+1|t} + \phi_3 Y_t$ and $Y_{t+4|t} = \phi_1 Y_{t+3|t} + \phi_2 Y_{t+2|t} + \phi_3 Y_{t+1|t}$. The last of these is a linear combination of forecasts as would be any $Y_{t+L|t}$ with $L > 4$. Therefore, the vector $(Y_t, Y_{t+1|t}, Y_{t+2|t})$ has everything needed to compute $Y_{t+3|t}$ and, subsequently, the sequence of all forecasts into the infinite future as linear combinations. Hence, it serves as a state vector in Akaike's formulation.

To complete the state space formulation as Akaike gives it, the transition matrix \mathbf{T} and transition error vector are needed. To get from the state vector at time t , $(Y_t, Y_{t+1|t}, Y_{t+2|t})$, to $(Y_{t+1}, Y_{t+2|t+1}, Y_{t+3|t+1})$ (that is, to the state vector that includes information up through time $t+1$), the (constant) transition matrix \mathbf{T} is needed. Notice that Y_{t+1} is $Y_{t+1|t} + e_{t+1}$. In other words, the forecast of Y_{t+1} that is computed at time t contains all of Y_{t+1} except the error e_{t+1} , which is unknown at time t . Therefore, the first row of the transition is the following:

$$Y_{t+1} = (0 \quad 1 \quad 0) \begin{pmatrix} Y_t \\ Y_{t+1|t} \\ Y_{t+2|t} \end{pmatrix} + e_{t+1}$$

Next, note that $Y_{t+2|t+1} = \phi_1 Y_{t+1} + \phi_2 Y_t + \phi_3 Y_{t-1} = \phi_1(Y_{t+1|t} + e_{t+1}) + \phi_2 Y_t + \phi_3 Y_{t-1}$, which is $\phi_1 e_{t+1}$ plus three terms that are known at time t . Because $Y_{t+2|t} = \phi_1 Y_{t+1|t} + \phi_2 Y_t + \phi_3 Y_{t-1}$ so that $Y_{t+2|t+1} = Y_{t+2|t} + \phi_1 e_{t+1}$, the first two transition rows are as follows:

$$\begin{pmatrix} Y_{t+1} \\ Y_{t+2|t+1} \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} Y_t \\ Y_{t+1|t} \\ Y_{t+2|t} \end{pmatrix} + \begin{pmatrix} 1 \\ \phi_1 \end{pmatrix} e_{t+1}$$

The last transition equation row can be discerned by noting that $Y_{t+3|t+1} = \phi_1 Y_{t+2|t+1} + \phi_2 Y_{t+1} + \phi_3 Y_t$, in terms of things that are known at time $t+1$. To relate this forecast to things that are known at time t (and thus to the time t state vector), note that the previous two-row equation relates the time $t+1$ terms needed to time t terms. From that, $Y_{t+3|t+1} = \phi_1(Y_{t+2|t} + \phi_1 e_{t+1}) + \phi_2(Y_{t+1|t} + e_{t+1}) + \phi_3 Y_t$. This gives the full transition equation:

$$\begin{pmatrix} Y_{t+1} \\ Y_{t+2|t+1} \\ Y_{t+3|t+1} \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ \phi_3 & \phi_2 & \phi_1 \end{pmatrix} \begin{pmatrix} Y_t \\ Y_{t+1|t} \\ Y_{t+2|t} \end{pmatrix} + \begin{pmatrix} 1 \\ \phi_1 \\ \phi_1^2 + \phi_2 \end{pmatrix} e_{t+1}$$

The error vector is of dimension 3, where each element has mean 0 and each is a different multiple of e_t . Therefore, the error vector can be described as a mean 0 vector with variance matrix:

$$\mathbf{Q} = \begin{pmatrix} 1 \\ \phi_1 \\ \phi_1^2 + \phi_2 \end{pmatrix} \sigma^2 (1 \quad \phi_1 \quad \phi_1^2 + \phi_2)$$

As before, the vector multiple of e_{t+1} can be described by specifying the previous rank 1 \mathbf{Q} matrix. As a check, the SAS Output Delivery System can be used to output the parameter estimates table from **Output 7.24** into a data set called Betas, which is pulled into the IML procedure as the vector **Vec**. The IML procedure computes \mathbf{Q} . The // command concatenates the vector entries to form the estimate of the column $(1 \quad \phi_1 \quad \phi_1^2 + \phi_2)'$, which is multiplied by its transpose and by the estimate of σ^2 . This is the first entry in **Vec**, symbolized **vec[1,1]**:

```
proc iml;
  use betas;
  read all var {estimate} into vec;
  col = {1}//vec[2,1]//vec[2,1]*vec[2,1]+vec[3,1];
  cov = vec[1,1]*col`*col;
  print vec;
  print cov;
quit;
```

The PRINT command gives the entries in **Vec** and the covariance matrix **Cov** shown in **Output 7.23**. This matches the **Q** matrix in **Output 7.24**, verifying the previous computations. **Vec** is the estimate of $(\sigma^2 \quad \phi_1 \quad \phi_2 \quad \phi_3)'$.

Output 7.23: PROC IML Results

VEC
2.8265434
0.8493949
0.0641285
-0.095726

COV		
2.8265434	2.4008515	2.2205329
2.4008515	2.0392709	1.8861092
2.2205329	1.8861092	1.7444509

Consider the series $(Y_t - \mu) = 0.9(Y_{t-1} - \mu) + 0.12(Y_{t-2} - \mu) - 0.16(Y_{t-3} - \mu) + \eta_t$, where the error term is white noise with variance 4 and the mean is $\mu = 100$. An equivalent representation is $(1 + 0.4B)(1 - 0.8B)(1 - 0.5B)(Y_t - \mu) = \eta_t$, which has no unit roots and implies stationarity. Fortunately, for ARIMA components, a built-in structure is available, eliminating the need to specify T_t and Q_t matrices. The next example illustrates the use of a TREND statement to specify this autoregressive model. The TREND statement is used because it has an ARIMA option and can handle the full suite of ARIMA structures. The TREND statement also has random walk and spline options, as well as growth and decay curve options. Alternatively, a DEPLAG statement could be used because an autoregressive model involves lags only of the dependent variable.

The following program analyzes 100 observations, generated from this series, with the SSM procedure. It produces the data set named Betas, used in the previous IML code:

```
data ar3;
  input y @@; t+1;
  datalines;
98.00 98.65 100.53 98.08 98.68 97.19 98.42 97.79 98.79 98.45
99.64 99.19 100.89 97.43 96.30 99.06 99.97 97.17 99.52 100.15
102.58 104.66 103.35 103.32 101.44 101.79 102.43 102.01 102.62 99.72
99.56 98.50 98.50 97.60 99.07 98.18 97.19 101.09 99.91 96.50
96.47 98.99 99.89 99.80 97.88 100.82 97.67 97.29 99.14 100.41
103.44 105.20 107.16 103.59 101.92 100.65 99.31 97.22 96.21 98.63
98.49 98.78 97.90 100.44 99.99 98.89 100.10 101.16 99.37 98.52
98.41 98.58 99.68 100.24 100.35 101.18 101.89 103.55 101.58 100.97
99.55 97.17 98.01 97.56 95.77 95.98 95.75 91.83 92.73 91.56
91.31 95.13 95.82 99.90 98.48 99.11 94.95 93.77 93.59 92.13
;

ods output ParameterEstimates=betas;
proc ssm data=ar3 optimizer(maxiter=100);
  int=1;
  trend lagsto3(arima(p=3)) print=(t cov);
  model y = int lagsto3;
run;
```

Output 7.24 (partial output) shows that the estimates are within 2 standard errors of the actual values used to generate the data, but standard errors are not particularly small. The transition matrix is as expected.

Output 7.24: Results from the ARIMA Option in the SSM Procedure

Regression Parameter Estimates					
Response Variable	Regression Variable	Estimate	Standard Error	t Value	Pr > t
Y	INT	98.7	0.89	110.90	<.0001

Model Parameter Estimates					
Component	Type	Parameter	Estimate	Standard Error	t Value
LAGSTO3	ARMA Trend	Error Variance	2.8265	0.4021	7.03
LAGSTO3	ARMA Trend	AR_1	0.8494	0.0877	9.68
LAGSTO3	ARMA Trend	AR_2	0.0641	0.2046	0.31
LAGSTO3	ARMA Trend	AR_3	-0.0957	0.1773	-0.54

Likelihood optimization algorithm converged in 76 iterations.

Transition Matrix for LAGSTO3			
	Col1	Col2	Col3
Row1	0	1	0
Row2	0	0	1
Row3	-0.09573	0.064128	0.849395

Disturbance Covariance for LAGSTO3			
	Col1	Col2	Col3
Row1	2.826543	2.400851	2.220533
Row2	2.400851	2.039271	1.886109
Row3	2.220533	1.886109	1.744451

7.4.6 Bivariate Examples

A bivariate vector autoregression can be fit as well. Consider this VAR dimension 2 lag 1 model:

$$\begin{pmatrix} X_{t+1} \\ W_{t+1} \end{pmatrix} = \begin{pmatrix} \lambda_{X,t} \\ \lambda_{W,t} \end{pmatrix} + \begin{pmatrix} \phi_{11} & \phi_{12} \\ \phi_{21} & \phi_{22} \end{pmatrix} \begin{pmatrix} X_t \\ W_t \end{pmatrix} + \begin{pmatrix} e_{1,t+1} \\ e_{2,t+1} \end{pmatrix}$$

Here, the intercept vector is a bivariate random walk that has the same structure as the basic structural model's local level:

$$\begin{pmatrix} \lambda_{X,t+1} \\ \lambda_{W,t+1} \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \lambda_{X,t} \\ \lambda_{W,t} \end{pmatrix} + \begin{pmatrix} v_{1,t+1} \\ v_{2,t+1} \end{pmatrix}$$

The 2×2 disturbance variance matrix is, for example, \mathbf{Q}_V . Elimination of error terms in this equation (assuming that \mathbf{Q}_V has all elements 0) changes the vector of λ s from a bivariate random walk to a constant vector.

The state vector and transition equation are written in terms of this local intercept vector concatenated with the associated bivariate observation:

$$\begin{pmatrix} \lambda_{X,t+1} \\ \lambda_{W,t+1} \\ X_{t+1} \\ W_{t+1} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & \phi_{11} & \phi_{12} \\ 0 & 1 & \phi_{21} & \phi_{22} \end{pmatrix} \begin{pmatrix} \lambda_{X,t} \\ \lambda_{W,t} \\ X_t \\ W_t \end{pmatrix} + \begin{pmatrix} v_{1,t+1} \\ v_{2,t+1} \\ v_{1,t+1} + e_{1,t+1} \\ v_{2,t+1} + e_{2,t+1} \end{pmatrix}$$

At time $t+1$, the model is expressed as follows:

$$\begin{pmatrix} X_{t+1} \\ W_{t+1} \end{pmatrix} = \begin{pmatrix} \lambda_{X,t+1} \\ \lambda_{W,t+1} \end{pmatrix} + \begin{pmatrix} \phi_{11} & \phi_{12} \\ \phi_{21} & \phi_{22} \end{pmatrix} \begin{pmatrix} X_t \\ W_t \end{pmatrix} + \begin{pmatrix} e_{1,t+1} \\ e_{2,t+1} \end{pmatrix}$$

The error variance is, for example, \mathbf{Q}_E . Except for the error term, the right side of a state space model would be expected to involve time t terms, not time $t+1$ terms. As it stands, the local intercept vector is at time $t+1$, but it can be decomposed by using its value at time t plus the increment at time $t+1$. This gives the following:

$$\begin{pmatrix} X_{t+1} \\ W_{t+1} \end{pmatrix} = \begin{pmatrix} \lambda_{X,t} \\ \lambda_{W,t} \end{pmatrix} + \begin{pmatrix} v_{1,t+1} \\ v_{2,t+1} \end{pmatrix} + \begin{pmatrix} \phi_{11} & \phi_{12} \\ \phi_{21} & \phi_{22} \end{pmatrix} \begin{pmatrix} X_t \\ W_t \end{pmatrix} + \begin{pmatrix} e_{1,t+1} \\ e_{2,t+1} \end{pmatrix}$$

This justifies the last two rows of the transition equation, which becomes the aforementioned expression:

$$\begin{pmatrix} \lambda_{X,t+1} \\ \lambda_{W,t+1} \\ X_{t+1} \\ W_{t+1} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & \phi_{11} & \phi_{12} \\ 0 & 1 & \phi_{21} & \phi_{22} \end{pmatrix} \begin{pmatrix} \lambda_{X,t} \\ \lambda_{W,t} \\ X_t \\ W_t \end{pmatrix} + \begin{pmatrix} v_{1,t+1} \\ v_{2,t+1} \\ v_{1,t+1} + e_{1,t+1} \\ v_{2,t+1} + e_{2,t+1} \end{pmatrix}$$

What is the 4×4 disturbance covariance matrix \mathbf{Q} here? The upper left 2×2 corner is \mathbf{Q}_V . The lower right is the sum of variance matrices $\mathbf{Q}_V + \mathbf{Q}_E$. What is the upper right 2×2 submatrix? Because the v and e subvectors are independent of

each other, the covariance between the two v s and the $(v + e)$ s would also be \mathbf{Q}_V (the two-dimensional \mathbf{V} vector is common to both). That gives a 4×4 \mathbf{Q} matrix form with 2×2 submatrices:

$$\mathbf{Q} = \begin{pmatrix} \mathbf{Q}_V & \mathbf{Q}_V \\ \mathbf{Q}_V & \mathbf{Q}_V + \mathbf{Q}_E \end{pmatrix}$$

Such a model can be used in predator (Y) and prey (X) studies, where both series would be expected to have autocorrelation, and current predator numbers would be expected to be high (positive lag 1 cross correlation) in years following high prey abundance. Prey numbers would be expected to be low following years of high predator counts. In other words, you would expect a bivariate autoregression with parameters having these signs:

$$\begin{pmatrix} X_{t+1} \\ W_{t+1} \end{pmatrix} = \begin{pmatrix} \lambda_{X,t+1} \\ \lambda_{W,t+1} \end{pmatrix} + \begin{pmatrix} + & - \\ + & + \end{pmatrix} \begin{pmatrix} X_t \\ W_t \end{pmatrix} + \begin{pmatrix} e_{1,t+1} \\ e_{2,t+1} \end{pmatrix}$$

7.4.7 The Start-up Problem Revisited

As in the UCM procedure, the idea of a diffuse initialization crops up in the SSM procedure. For example, in the univariate AR(1) case, where $|\rho| < 1$, the variance of Y is $\sigma^2 / (1 - \rho^2)$, which could be very large for ρ near 1. Because the intercept is $\lambda = \mu(1 - \rho^2)$, the mean of Y is $\mu = \lambda / (1 - \rho^2)$ so that the contribution of the first variable to -2Log(Likelihood) is the following:

$$\ln(2\pi) + \ln(\sigma^2) - \ln(1 - \rho^2) + (Y_1 - \mu)^2 / (1 - \rho^2) / \sigma^2$$

This would be useful if only stationary autoregressive models were of interest. But, consider a simple random walk model. If ρ approaches 1, the term $\ln(1 - \rho^2)$ diverges, and Y_1 is multiplied by a value that approaches 0 for any given σ^2 value. As with UCM, following this logic leads to the diffuse likelihood that omits the observation 1 term from the likelihood.

The syntax uses components. It can fit far more complex models than previously described. To demonstrate with something simple, 1000 observations from an AR(1) model with $\mu = 100$, $\rho = 0.9$, and $\sigma^2 = 16$ are generated and analyzed with this code:

```
%let rho=0.9;
%let mu=100;
%let var_e=16;

data ar1;
  s = sqrt(&var_e);
  y = &mu + normal(12321)*sqrt(&var_e/(1-&rho**2));
  do t=1 to 1000;
    y = &mu + &rho*(y-&mu)+s*normal(12321);
    output;
  end;

**(1);
proc arima data=ar1; title "arima ml";
  identify var=y noprint;
  estimate p=1 method=ml;
run;

**(2);
proc mixed data=ar1; title "mixed";
  ods output covparms=mixed;
  model y= /outpm=mean ddfm=kr;
  repeated /subject=int type=ar(1);
run;

proc print data=mean(obs=4);
proc print data=mixed;
run;

**(3);
proc ssm data=ar1; title "ssm";
  parms a1 /lower=-0.999 upper=0.9999;
```

```

parms vare /lower=1.e-6;
varl = vare/(1-a1*a1);
state   y_state(1)
        cov(g) =(vare)
        covl(g)=(varl)
print= (cov t)
t(g) = (a1);
component ar1 = y_state[1];
intercept=1;
model y = ar1 intercept;
run;
title;

```

The ARIMA approach uses maximum likelihood on all parameters, including the mean. PROC MIXED is a general program for fitting mixed models and uses restricted maximum likelihood (REML) by default. The REPEATED statement's SUBJECT=INT option causes the MIXED procedure to assume that the entire record comes from one subject. It suggests that the subject dummy variables consist of just one column—a column of all 1s (for example, an intercept column).

The PROC SSM PARMS statement specifies estimation of the AR(1) parameter and sets lower bounds for the variances of the white noise error and the response variable. It bounds the correlation between the response and its lag (for example, the autoregressive parameter estimate) to be within the interval $[-0.99, 0.99]$. This keeps it away from hitting or crossing the unit root boundary. The STATE statement declares Y_STATE to be the name of the subvector (1 element here) in the first position or postions of the state vector.

For comparison, portions of the outputs are given in **Output 7.25**.

Output 7.25: Comparison of Estimates in an AR(1) Model

ARIMA ML

Maximum Likelihood Estimation					
Parameter	Estimate	Standard Error	t Value	Approx Pr > t	Lag
MU	100.01156	1.22431	81.69	<.0001	0
AR1,1	0.89346	0.01436	62.23	<.0001	1

MIXED

Covariance Parameter Estimates		
Cov Parm	Subject	Estimate
AR(1)	Intercept	0.8954
Residual		87.2656

SSM

Regression Parameter Estimates					
Response Variable	Regression Variable	Estimate	Standard Error	t Value	Pr > t
Y	INTERCEPT	100	1.25	80.21	<.0001

Estimates of Named Parameters			
Parameter	Estimate	Standard Error	t Value
A1	0.895	0.0144	62.05
VARE	17.302	0.7742	22.35

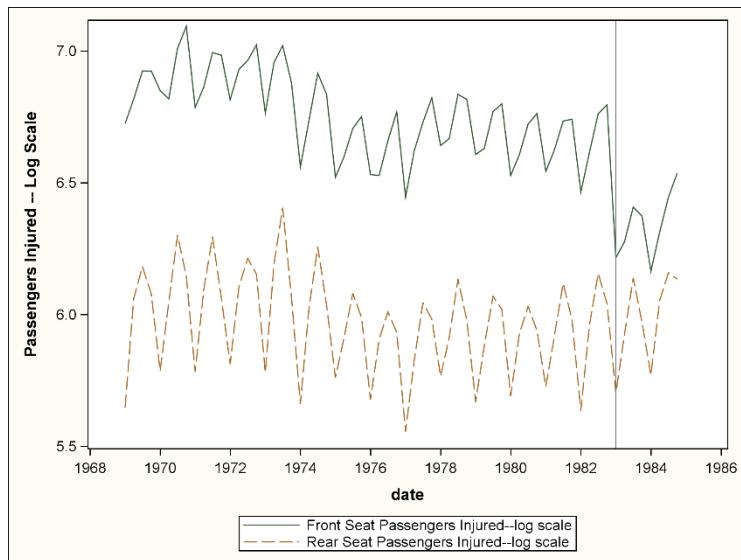
The estimates of the autoregressive parameter from these three approaches are all close. Those of PROC MIXED and PROC SSM match exactly for this model. The MIXED procedure's residual variance is an estimate of $\sigma^2 / (1 - \rho^2)$ where, compared to the SSM output, the estimate of σ^2 is 17.302. The autoregressive parameter ρ is 0.895, thus showing that both parameter estimates match. The approaches follow similar philosophies. The REML approach in PROC MIXED and the diffuse likelihood in PROC SSM eliminate the information about the mean initially. This results in residuals with which to estimate the autoregressive parameter. The two procedures produce the same estimate in this simple case. In contrast, PROC ARIMA maximizes the likelihood on all parameters, but is restricted to stationary models. For most cases in PROC MIXED, the default REML approach is preferred.

7.4.8 Example and More Details on the State Space Approach

The SSM procedure can fit vector (multivariate) responses. In addition, it allows deterministic (regression type) inputs. For example, **Output 7.26** shows quarterly data on the log scale for front seat (higher) and rear seat (lower) injuries in automobile crashes. The data come from SAS documentation for PROC SSM. With trigonometric quarterly bivariate variables, there are three seasonal variables for each of the two responses, six variables in all. The data set has 64 nonmissing and four missing observations at the end for each seat type.

A vertical line at January 1, 1983, shows the date of enactment of a law requiring front-seat passengers to wear seat belts. It appears to have an effect on front-seat passenger injuries, but no obvious effect on back-seat passenger injuries. Another obvious feature that needs to be modeled is the strong seasonal pattern. A possible downward trend is suggested in the upper (front seat) plot, but that could easily result from a local level component. For example, note the relatively horizontal overall pattern from 1971 through 1973, and again from 1975 to the intervention point. Trend is ignored here. Because accidents can happen with no back-seat passengers, but almost never without front-seat passengers, it is not surprising that the overall levels of the two plots differ. It is not obvious whether an exactly periodic function or one with local features is more appropriate. Previous examples have shown ways of dealing with this decision in SSMs and UCMs.

Output 7.26: Quarterly Front-Seat and Back-Seat Injuries from Auto Accidents



The general state space equations are simplified for the traffic accident data. A bivariate basic structural model is fit. Terms not used in this model and terms dealing with diffuse initial conditions are omitted from this discussion. The state vector has 10 elements:

$$\alpha_t = (e_{1,t} \quad e_{2,t} \quad L_{1,t} \quad L_{2,t} \quad S_{1,1,t} \quad S_{1,2,t} \quad S_{1,3,t} \quad S_{2,1,t} \quad S_{2,2,t} \quad S_{2,3,t})'$$

The state transition equation is $\alpha_{t+1} = \mathbf{T}\alpha_t + \boldsymbol{\eta}_{t+1}$, where $\boldsymbol{\eta}_{t+1}$ is a 10-element column vector consisting of zeros and mean 0 random variables. It is called the transition equation *disturbance vector*. The \mathbf{T} matrix is block diagonal with a 2×2 block of 0s for the white noise e terms, a 2×2 block for the level variables L , and a 6×6 block for the seasonal variables S . For the seasonal terms, trigonometric computations are used, and the 6×6 coefficient matrix further decomposes into blocks of dimensions 2×2 , 1×1 , 2×2 , and 1×1 . The first subscript on each S corresponds to the

relevant response variable, front seat or back seat. The second subscript corresponds to the appropriate frequency. Recall that the sum of angles formula gives the following:

$$S_{1,1,t+1} = A \cos(2\pi(t+1)/4) = \cos(2\pi/4)A \cos(2\pi t/4) - \sin(2\pi/4)A \sin(2\pi t/4)$$

and

$$S_{1,2,t+1} = B \sin(2\pi(t+1)/4) = \sin(2\pi/4)B \cos(2\pi t/4) + \cos(2\pi/4)B \sin(2\pi t/4)$$

Here, A and B are amplitudes so that the (sine, cosine) pair for frequency $\pi/4$ and front-seat passengers transitions from time t to $t+1$ as:

$$\begin{pmatrix} S_{1,1,t+1} \\ S_{1,2,t+1} \end{pmatrix} = \begin{pmatrix} \cos(2\pi/4) & -\sin(2\pi/4) \\ \sin(2\pi/4) & \cos(2\pi/4) \end{pmatrix} \begin{pmatrix} S_{1,1,t} \\ S_{1,2,t} \end{pmatrix} = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} S_{1,1,t} \\ S_{1,2,t} \end{pmatrix}$$

This shows the 2×2 submatrix of \mathbf{T} associated with the subvector $(S_{1,1,t}, S_{1,2,t})'$ of the state. At frequency $2\pi/4$, there is this (sine, cosine) pair, but at the next frequency, $2(2\pi/4) = \pi$, the sine function $\sin(4\pi t/4) = \sin(\pi t)$ is identically 0, so only the cosine function enters the model. Because $S_{1,3,t+1} = A \cos(\pi(t+1)) = A \cos(\pi) \cos(\pi t) = -A \cos(\pi t) = -S_{1,3,t}$, a simple univariate transition $S_{1,3,t+1} = -S_{1,3,t}$ holds for the third element of this state subvector. These three S values are for the front seat. Changing the first subscripts from 1 to 2 gives the transitions for the back-seat seasonal variables.

A level component of a random walk nature, as in the basic structural model discussed for the UCM procedure, is specified using 1s on the diagonal of \mathbf{T} for the L components. The transition equation would then be the following:

$$\begin{pmatrix} e_{1,t+1} \\ e_{2,t+1} \\ L_{1,t+1} \\ L_{2,t+1} \\ S_{1,1,t+1} \\ S_{1,2,t+1} \\ S_{1,3,t+1} \\ S_{2,1,t+1} \\ S_{2,2,t+1} \\ S_{2,3,t+1} \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \end{pmatrix} \begin{pmatrix} e_{1,t} \\ e_{2,t} \\ L_{1,t} \\ L_{2,t} \\ S_{1,1,t} \\ S_{1,2,t} \\ S_{1,3,t} \\ S_{2,1,t} \\ S_{2,2,t} \\ S_{2,3,t} \end{pmatrix} + \begin{pmatrix} e_{1,t+1} \\ e_{2,t+1} \\ u_{1,t+1} \\ u_{2,t+1} \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

On the left is the state vector \mathbf{a}_{t+1} at time $t+1$. On the right is the transition matrix \mathbf{T} times the state vector \mathbf{a}_t at time t , plus a matrix of mean 0 random terms, the *transition input vector* \mathbf{u}_{t+1} . This input vector uses u for the inputs to the level equations and e for the noise inputs to the state vector.

Recall the observation equation $\mathbf{Y}_t = \mathbf{Z}_t \mathbf{a}_t + \mathbf{X}_t \boldsymbol{\beta} + \boldsymbol{\varepsilon}_t$, relating the response vector \mathbf{Y}_t to the state vector \mathbf{a}_t through $\mathbf{Z}_t \mathbf{a}_t$, a regression term $\mathbf{X}_t \boldsymbol{\beta}$, and a random vector of observation errors $\boldsymbol{\varepsilon}_t$. In the SSM procedure, a separate MODEL statement is issued for each response variable, thus implying independence of elements in the vector $\boldsymbol{\varepsilon}_t$. In the auto injury data, some crashes have both front-seat and back-seat passengers in the same crash, making independence of the two response vector elements unlikely. Instead, what might have otherwise been independent observation errors specified in IRREGULAR statements are included in the state vector to accommodate this likely correlation.

When included in the state vector, as is done with $e_{1,t}$ and $e_{2,t}$, nonzero covariances are allowed. As a result, in this example, there is no additional observation error to be specified. For that reason, no IRREGULAR statement appears in the upcoming code. Error terms in the IRREGULAR statement are assumed independent. To relate the two-dimensional data vector without $\mathbf{X}_t \boldsymbol{\beta}$ (the seatbelt law effect) to the state vector \mathbf{a}_t for the auto injury example, the observation equation is as follows:

$$\mathbf{Y}_t - \mathbf{X}_t \boldsymbol{\beta} = \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \end{pmatrix} \mathbf{a}_t + \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

The first row relates the appropriate error, level, and three seasonal terms for front-seat passengers to the front-seat injury counts. Likewise, the second row of the 2×10 transition matrix picks up the back-seat injuries. The added vector of 0s is a reminder that the general state space theory allows for added observational errors in the observation equation, which are suppressed in the auto injury data. The intervention variable for the seatbelt law is added to the model as a regressor. Thus, it is ignored (subtracted out of the response vector) in the state vector discussion. Its effect is modeled in $\mathbf{X}_t\beta$.

The random walk level vector $(L_{1,t}, L_{2,t})$ in the model might be a pair of cointegrated unit root processes. To do this, some restriction must be imposed. With bivariate input (u_1, u_2) to the level vector, the covariance matrix has the following form:

$$\Sigma_U = \begin{pmatrix} \sigma_1^2 & \sigma_{12} \\ \sigma_{12} & \sigma_2^2 \end{pmatrix}$$

If the variances are known or estimated, the matrix or its estimate can be forced to have rank 1 by selecting the covariance to force the ratios of elements in its two columns to be the same. That is:

$$\sigma_1^2 / \sigma_{12} = \sigma_{12} / \sigma_2^2$$

or

$$\sigma_{12} = \pm \sqrt{\sigma_1^2 \sigma_2^2}$$

This ensures that there is a linear combination of the inputs (u_1, u_2) that is always 0. Thus, the sum over time of that linear combination is 0. This implies that the same linear combination of L_1 and L_2 is 0 at all times. For example, consider the following equation:

$$\Sigma_U = \begin{pmatrix} 12 & \sigma_{12} \\ \sigma_{12} & 3 \end{pmatrix}$$

Here, the covariance σ_{12} would be 6, and it is easily seen that the variance of $u_{1,t} - 2u_{2,t}$ is the following:

$$(1 \quad -2)\Sigma_U \begin{pmatrix} 1 \\ -2 \end{pmatrix} = (1 \quad -2) \begin{pmatrix} 12 & 6 \\ 6 & 3 \end{pmatrix} \begin{pmatrix} 1 \\ -2 \end{pmatrix} = 0$$

With variance 0, $u_{1,t} - 2u_{2,t}$ is constant and equal to its mean (0) with probability 1. Now, suppose that the response variable is the sum of several components, including the vector random effect, for example, \mathbf{E}_t . This is the case in the accident data. This is given by the following sum:

$$\mathbf{E}_t = \begin{pmatrix} L_{1,t} \\ L_{2,t} \end{pmatrix} + \begin{pmatrix} e_{1,t} \\ e_{2,t} \end{pmatrix} = \begin{pmatrix} L_{1,t} + e_{1,t} \\ L_{2,t} + e_{2,t} \end{pmatrix}$$

Suppose you want to model this nonseasonal part E_t of the response vector as a cointegrated series. This means that you have to find a linear combination of the nonseasonal part that is stationary. Assuming the rank 1 covariance matrix previously shown, the linear combination $L_{1,t} - 2L_{2,t}$ is a vector of 0s because

$$L_{1,t} - 2L_{2,t} = \sum_{j=1}^t (u_{1,j} - 2u_{2,j}) = \sum_{j=1}^t (0) = 0$$

This implies that

$$(1 \quad -2) \begin{pmatrix} L_{1,t} + e_{1,t} \\ L_{2,t} + e_{2,t} \end{pmatrix} = 0 + e_{1,t} - 2e_{2,t}$$

This is a white noise sequence and is stationary. By definition, the two vector components, which are individually unit root processes, are cointegrated with $(1, -2)$ serving as the cointegrating vector. Cointegration might be appropriate for the accident data as injury counts from cars with both front-seat and back-seat passengers are coming from the same

accidents. That is, they are paired by cars. The COV(RANK=1) option forces cointegration of the associated (level) model components.

This code produces the analysis:

```
proc ssm data=seatbelt stateinfo;
  id date interval=quarter;
  q1_83_shift = (date >= '1jan1983'd);

  state error(2) type=wn cov(g) print=cov;
  component wn1 = error[1];
  component wn2 = error[2];

  state level(2) type=rw cov(rank=1)print=cov;
  component rw1 = level[1];
  component rw2 = level[2];

  state season(2) type=season(length=4);
  component s1 = season[1];
  component s2 = season[2];

  model f_ksi = q1_83_shift rw1 s1 wn1 / print=(smooth);
  model r_ksi = rw2 s2 wn2;
  eval f_ksi_sa = rw1 + q1_83_shift;
  eval r_ksi_sa = rw2;
  output out=for1;
run;
```

After creating a descriptive date and the regression variable Q1_83_SHIFT, the program uses three sets of three statements to describe the three subvectors that constitute the state vector with (2) indicating that there are two response variables. With LENGTH=4 in the seasonal section, three seasonal terms for each response are created. The MODEL statements allow different effects from the state vector to affect each response. It adds any additional regressor variables to the model. EVAL statements allow the user to compute combinations of the modeled effects, such as the F_KSI_SA combination, which omits the seasonal component and shows a deseasonalized series.

Output 7.27 shows partial results. Some parts deal with diffuse initial conditions, which are the methods used to deal with the uncertainty in initial component values. The number of diffuse initializations, 9 (an intervention, two levels, and six seasonal values), and the number of components, two level, two error, and six seasonal components, in the state vector are produced by the STATEINFO option.

Output 7.27: SSM Results for Seatbelt Data

Model Summary	
Model Property	Value
Number of Model Equations	2
State Dimension	10
Dimension of the Diffuse Initial Condition	9
Number of Parameters	5

State Vector Composition	
Subsection	Dimension
ERROR	2
LEVEL	2
SEASON	6

Diffuse Initial State Composition (Including Regressors)	
Subsection	Dimension
LEVEL	2
SEASON	6
Q1_83_SHIFT	1

ID Variable Information					
Name	Start	End	Max Delta	NDistinct	Type
DATE	1969:1	1985:4	1	68	Regular

Response Variable Information							
Name	Number of Observations			Minimum	Maximum	Mean	Std Deviation
	Total	Missing	Induced Missing				
F_KSI	68	4	0	6.16	7.09	6.71	0.206
R_KSI	68	4	0	5.56	6.41	5.97	0.186

Regression Parameter Estimates					
Response Variable	Regression Variable	Estimate	Standard Error	t Value	Pr > t
F_KSI	Q1_83_SHIFT	-0.408	0.0259	-15.74	<.0001

Model Parameter Estimates					
Component	Type	Parameter	Estimate	Standard Error	t Value
ERROR	Disturbance Covariance	RootCov[1, 1]	0.0361	0.00736	4.91
ERROR	Disturbance Covariance	RootCov[2, 1]	0.0338	0.01131	2.99
ERROR	Disturbance Covariance	RootCov[2, 2]	0.0462	0.00470	9.84
LEVEL	Disturbance Covariance	RootCov[1, 1]	0.0375	0.00843	4.45
LEVEL	Disturbance Covariance	RootCov[2, 1]	0.0223	0.00569	3.92

Likelihood optimization algorithm converged in 21 iterations.

Likelihood Computation Summary	
Statistic	Value
Nonmissing Response Values Used	128
Estimated Parameters	5
Initialized Diffuse State Elements	9
Normalized Residual Sum of Squares	119
Diffuse Log Likelihood	166.15755
Profile Log Likelihood	199.91165

Information Criteria		
Statistic	Diffuse Likelihood Based	Profile Likelihood Based
AIC (lower is better)	-322.3151	-371.8233
BIC (lower is better)	-308.4195	-331.8949
AICC (lower is better)	-321.7841	-368.1065
HQIC (lower is better)	-316.6725	-355.6002
CAIC (lower is better)	-303.4195	-317.8949

Disturbance Covariance for ERROR		
	Col1	Col2
Row1	0.001307	0.001222
Row2	0.001222	0.003277

Disturbance Covariance for LEVEL		
	Col1	Col2
Row1	0.001408	0.000837
Row2	0.000837	0.000497

Additive Outlier Summary						
Obs	ID	Response Variable	Estimate	Standard Error	Chi-Square	Pr > ChiSq
5	1970:1	F_KSI	0.126	0.0393	10.34	0.0013

Full-Sample Prediction of Missing Values for F_KSI					
Obs	ID	Estimate	Standard Error	95% Confidence Limits	
65	1985:1	6.26	0.0607	6.14	6.38
66	1985:2	6.35	0.0713	6.21	6.49
67	1985:3	6.48	0.0802	6.32	6.64
68	1985:4	6.50	0.0880	6.33	6.67

The regression parameter estimates show a highly significant permanent drop, -0.408 , in injuries to front-seat passengers consistent with the introduction of the seatbelt law. Computing the ratio of the two elements in column 1 of the disturbance covariance for levels matrix and the same for column 2 shows the same ratio, which enforces cointegration of the elements of the random E_t vector. This was accomplished using the COV(RANK=1) option. The 1970:1 front-seat observation is tagged as an outlier. Because that observation was not in question before looking at the data, you might argue for multiplying the p -value by the number of nonmissing observations in the data set, a Bonferroni correction, to be conservative. This becomes $128(0.0013)$, where 128 is the number of nonmissing observations in the combined front-seat and back-seat data.

The output data set can be used for graphing and further analyzing the estimates. Some examples are given here. The following code produces means of the smoothed front-seat and back-seat seasonal factors by quarter:

```
data graph1; set for1; qtr=qtr(date);
proc means mean std data=graph1; class qtr;
  var smoothed_s1 smoothed_s2;
  label smoothed_s1="front";
```

```

label smoothed_s2="rear";
run;

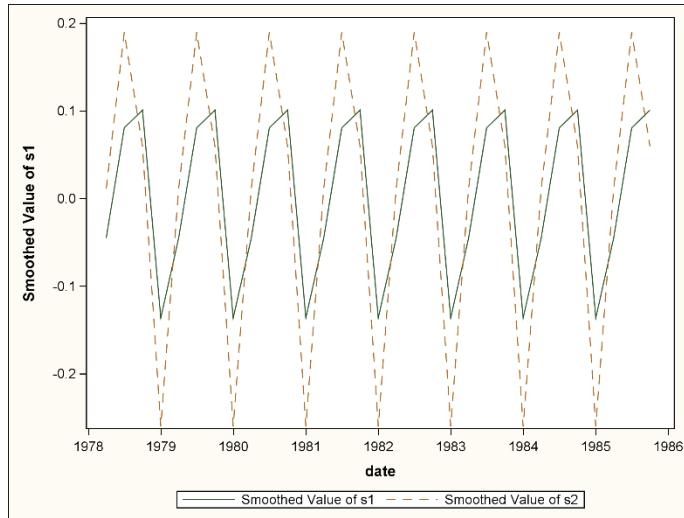
proc sgplot data=graph1; where date>"01jan78"d;
  series y=smoothed_s1 x=date;
  series y=smoothed_s2 x=date / lineattrs=(pattern=dash);
run;

```

Interestingly, the fluctuation (-0.26 in quarter 1 to 0.19 in quarter 3) in back-seat seasonal effects is larger than that in front-seat data (-0.14 in quarter 1 to 0.10 in quarter 4) and peaks one quarter earlier, though statistical significance of these differences has not been addressed.

Output 7.28: Smoothed Seasonal Series for Front-Seat and Back-Seat Passengers

QTR	N	Variable	Label	Mean	Std Dev
	Obs				
1	17	Smoothed_S1	FRONT	-0.1363284	0
		Smoothed_S2	REAR	-0.2602021	0
2	17	Smoothed_S1	FRONT	-0.0452122	0
		Smoothed_S2	REAR	0.0110545	0
3	17	Smoothed_S1	FRONT	0.0806572	0
		Smoothed_S2	REAR	0.1897149	0
4	17	Smoothed_S1	FRONT	0.1008834	0
		Smoothed_S2	REAR	0.0594328	0



Recall that the seasonal factors had no input variances. That is, they are fixed and global, rather than being local in nature. It is not surprising to see that the quarterly effects are all the same as indicated by the 0 standard deviations around these means. The graph shows the same thing with dashed lines for the back-seat passenger injuries and solid lines for the front-seat passengers in **Output 7.29**.

Two EVAL statements produced seasonally adjusted series that have the regressors, level, and white noise error components, but not the seasonal components. Some code to graph these two seasonally adjusted series and corresponding confidence intervals (followed by the resulting graph) are shown here:

```

proc sgplot data=graph1 noautolegend;
  band x=date
    upper=smoothed_upper_f_ksi_sa
    lower=smoothed_lower_f_ksi_sa / fill outline;
  series y=smoothed_f_ksi_sa           x=date
    /lineattrs=(color=red);

  band x=date
    upper=smoothed_upper_r_ksi_sa
    lower=smoothed_lower_r_ksi_sa / fill outline;

```

```
series y=smoothed_r_ksi_sa      x=date
      /lineattrs=(color=red);
run;
```

Output 7.29: Injury Forecasts for Front-Seat (Top Line) and Back-Seat Passengers

In this chapter, the UCM and SSM procedures were introduced and illustrated. The auto accident example shows some of the extensive functionality of the SSM procedure, including using regressors, cointegrated unit root processes, and seasonality, all in one bivariate model. The details of the Kalman filter likelihood computations, the effect of diffuse initial conditions, an impressive list of additional functionality, and more examples are available in the online documentation for each procedure.

Chapter 8: Adjustment for Seasonality with PROC X13

8.1 Introduction.....	285
8.2 The X-11 Method	287
8.2.1 Moving Averages	287
8.2.2 Outline of the X-11 Method	290
8.2.3 Basic Seasonal Adjustment Using the X-11 Method	291
8.2.4 Tests for Seasonality.....	292
8.3 regARIMA Models and TRAMO.....	295
8.3.1 regARIMA Models	295
8.3.2 Automatic Selection of ARIMA Orders.....	296
8.4 Data Examples.....	296
8.4.1 Airline Passengers Revisited	296
8.4.3 Employment in the United States	299

8.1 Introduction

The X13 procedure is an adaptation from SAS of the X-13ARIMA-SEATS seasonal adjustment program developed by the U.S. Bureau of the Census. Although it has many functionalities, the main purpose of the procedure is to seasonally adjust monthly or quarterly time series data. Denote by Y_t the observed response at time t . Then, the response series can be decomposed into three main components: trend-cyclical component (T_t), seasonal component (S_t), and irregular component (I_t). PROC TIMESERIES decomposes time series data into its underlying components using a different methodology.

The trend-cycle component (TCC) contains information about trend and cycle. There is no clear-cut definition for trend and cycle, but a business cycle theory suggests that the growth trend can be decomposed into stable, longer-term fluctuations (a linear trend for example) and into shorter-term fluctuations of unknown and perhaps varying periodicity. The former can be considered trend and the latter can be considered cycle. For example, when business falls off, employees can be laid off, leading to less income and less purchasing and further decline, until a bottom is reached and things begin to improve. This requires rehiring of workers and more money for spending and improved sales. This is a cycle. The model underlying the X13 procedure does not separate the two components. Rather, it uses a combined TCC.

The seasonal component contains regular fluctuations observed from year to year. The major difference between seasonality and cycle is this regularity and known periodicity. Thus, a series with seasonality tends to show similarly shaped fluctuations from year to year after being adjusted for the TCC. A regular period, such as 12 fluctuations in monthly data, is an example of a seasonal component. Also, a single cycle can span multiple years, whereas seasonal variation occurs within each year.

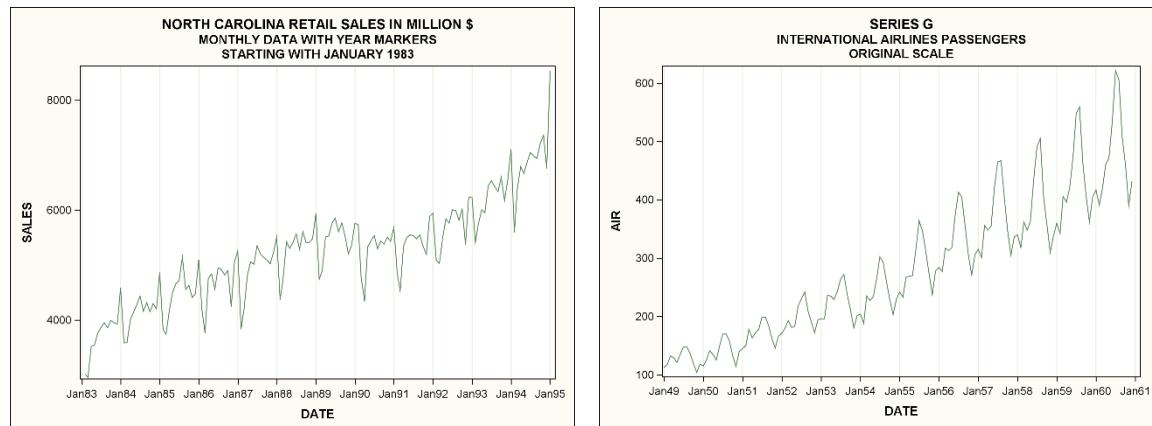
Two main models for the relationship between the observed series and unobserved components are the additive model and the multiplicative model:

- Additive model: $Y_t = T_t + S_t + I_t$
- Multiplicative model: $Y_t = T_t S_t I_t$

The seasonally adjusted (SA) series derived from the additive model is $\tilde{Y}_t = Y_t - S_t = T_t + I_t$, and that derived from the multiplicative model is $\tilde{Y}_t = Y_t / S_t = T_t I_t$. Additive models are good for series that show a constant magnitude of fluctuations. Multiplicative models are good for series with positive values that exhibit an increasing magnitude of fluctuations as the value of the series grows. **Output 8.1** displays the series plots of the monthly volume of North Carolina retail sales and number of international airline passengers. These examples have been used in Chapter 4. Both

plots show clear evidence of seasonality as the shape of the fluctuations within a year is similar across years. However, the magnitude of fluctuations increases with the level of the series for the airline passenger data. Apparently, the additive model is appropriate for the retail sales data, and the multiplicative model is appropriate for the airline passenger data.

Output 8.1: North Carolina Retail Sales and International Airline Passengers Data Revisited



The main purpose of seasonal adjustment is to study the series without the effect of seasonality. For example, policymakers do not want to be concerned every year when new graduates hit the market and cannot immediately find jobs, thus raising the unemployment rate. They do not want to pat themselves on the back at Christmas time when temporary sales jobs open up. If it happens the same month every year, then it is masking the true underlying unemployment rate or sales.

The US Census Bureau is tasked with numerically removing seasonality from various series of importance to the economy so that trends and cycles can be better assessed. A sequence of methodologies was developed over time. The development stabilized with the X-11 method, which used certain local averages to make the adjustments flexible and was available to the public. After a fairly long period of X-11 experience, two updated methods, X-12 and X-13, were released by the US Census Bureau. The X-13 version includes components developed at Statistics Canada and Bank of Spain.

The X13 procedure or X-13ARIMA-SEATS methodology can be decomposed into three parts:

- an X part represented by the X-11 method
- an ARIMA part represented by regARIMA (regression-ARIMA) models and TRAMO (Time Series Regression with ARIMA Noise, Missing Observations, and Outliers)
- a SEATS (Signal Extraction in ARIMA Time Series) part

The X-11 method applies moving averages to the series to estimate the seasonally adjusted series while adjusting for outliers. It adjusts for other effects such as trading day and holidays. One problem with this method is that symmetric moving averages cannot be applied to the end points of the series because there are not enough observations. Thus, an asymmetric moving average is used instead for those points. Studies show that the estimated series has a large inaccuracy at those end points although these values are usually of most interest.

The X-12 method improves the X-11 method by using regARIMA models to adjust the series for deterministic effects and to forecast and backcast the series so that symmetric moving averages can be computed for all points. Moreover, the method automatically selects ARIMA orders while adjusting for outliers. This automatic routine is based on TRAMO. Another benefit of regARIMA models is that users have more flexibility by optionally supplying user-defined inputs.

The X-13ARIMA-SEATS method further enriches the method by adding the SEATS routine to decompose the time series using an ARIMA model. The SEATS decomposition offers an alternative method of seasonal decomposition to the X-11 method.

The X13 procedure requires a minimum of three years of data (12 for quarterly and 36 for monthly).

8.2 The X-11 Method

The details of the original X-11 method are well illustrated in the book by Ladiray and Quenneville, *Seasonal Adjustment with the X-11 Method*. Using a step-by-step approach, the book describes exactly what computation is done at each step. The X-11 method has been modified over time through the stages of development to incorporate the added functionalities. The X-11 method used by the X13 procedure is a bit different from what is described in the book. But, the core idea remains the same: nonparametric estimation of components through moving averages. In this section, the building blocks of the X-11 method are briefly introduced, and an application of the method using the X13 procedure is illustrated on the North Carolina retail sales data.

8.2.1 Moving Averages

A moving average is a non-parametric estimation technique that uses weighted averages. A moving average operator and a seasonal moving average operator are as follows:

$$M(Y_t) = \sum_{i=-p}^q w_i Y_{t+i}$$

and

$$M^s(Y_t) = \sum_{i=-p}^q w_i Y_{t+s+i}$$

with

$$\sum_{i=-p}^q w_i = 1$$

Here, s is the season length and $p + q + 1$ is the moving average order, the number of terms appearing in the summation. When $p = q$, the moving average is said to be centered. A centered moving average with $w_i = w_{-i}$ is said to be symmetric.

A desirable property for moving averages is trend preservation. Suppose a trend component has a quadratic form: $T_t = \beta_0 + \beta_1 t + \beta_2 t^2$. What follows is this:

$$\begin{aligned} M(T_t) &= \sum_{i=-p}^q w_i T_{t+i} \\ &= \beta_0 \sum_{i=-p}^q w_i + \beta_1 \sum_{i=-p}^q w_i (t+i) + \beta_2 \sum_{i=-p}^q w_i (t+i)^2 \\ &= (\beta_0 + \beta_1 t + \beta_2 t^2) \sum_{i=-p}^q w_i + (\beta_1 + 2t\beta_2) \sum_{i=-p}^q w_i i + \beta_2 \sum_{i=-p}^q w_i i^2 \\ &= T_t \sum_{i=-p}^q w_i + (\beta_1 + 2t\beta_2) \sum_{i=-p}^q w_i i + \beta_2 \sum_{i=-p}^q w_i i^2 \end{aligned}$$

Note that $M(T_t) = T_t$ if the following holds:

$$\sum_{i=-p}^q w_i = 1$$

$$\sum_{i=-p}^q i w_i = 0$$

$$\sum_{i=-p}^q i^2 w_i = 0$$

These constraints preserve a quadratic trend. For a linear trend with $\beta_2 = 0$, only the first two constraints need to be satisfied. For a constant trend with $\beta_1 = 0$ and $\beta_2 = 0$, only the first constraint needs to be satisfied to preserve trend. In general, to preserve a polynomial trend of degree d , the required constraints are the following:

$$\sum_{i=-p}^q w_i = 1, \quad \sum_{i=-p}^q i^j w_i = 0 \quad \text{for } j = 1, \dots, d$$

The first constraint can be enforced easily by making the sum of the weights equal to 1. Also, the constraints are satisfied for all odd j if a symmetric moving average filter is used. For an even j , some negative weights are needed.

A class of symmetric moving averages, called *centered moving average filters*, is used by the X-11 method. Without loss of generality, suppose $u < v$:

$$M_{uvv}(Y_t) = \sum_{i=-p}^p w_i Y_{t+i} \quad \text{where } p = \frac{u+v}{2} - 1 \text{ and } w_i = \begin{cases} \frac{p+i+1}{uv} & \text{if } -p \leq i < -p+u-1 \\ \frac{u}{uv} & \text{if } -p+u-1 \leq i \leq p-u+1 \\ \frac{p-i+1}{uv} & \text{if } p-u+1 < i \leq p \end{cases}$$

$$M_{uvv}^s(Y_t) = \sum_{i=-p}^p w_i Y_{t+s+i}$$

The formula works only when $u + v$ is an even number, a condition satisfied by the X-11 method. The expression for weights might seem daunting at first, but it is just saying that for $u < v$, the numerators of the first u weights increase linearly from 1 to u , those of the last u weights decrease linearly from u to 1, and those of the middle weights are all u . Because weights are symmetric and the sum of the weights is 1, centered moving average filters preserve a linear trend.

Alternatively, this class of moving averages can be derived by applying simple averages of v terms for u neighboring points, including the given point itself, and taking the simple average of resulting averages. With this definition, u has to be an odd number, which is the case for the moving averages that the X-11 method chooses. For example, $M_{2 \times 12}(Y_t)$ is the average of two ordinary averages, the first from $t - 6$ to $t + 5$ and the second from $t - 5$ to $t + 6$, giving 13 weights:

$$\frac{1}{24}(1, 2, 2, \dots, 2, 1)$$

Another class of moving averages used by the X-11 method are Henderson symmetric moving averages. The $2p + 1$ weights for Henderson symmetric moving averages are obtained by minimizing the following objective function (smoothness criterion) over w_i :

$$\sum_{i=-p}^p ((1-B)^3 w_i)^2$$

with the following constraints:

$$\sum_{i=-p}^p w_i = 1, \quad \sum_{i=-p}^p i w_i = 0, \quad \sum_{i=-p}^p i^2 w_i = 0$$

$$w_i = 0 \quad \text{if } i > p \quad \text{or } i < p$$

The constraints preserve quadratic trends. In the smoothness criterion, B represents the backshift operator. The optimization problem is to find moving average coefficients with the following characteristics:

- They are smooth.
- They preserve a quadratic trend.
- They are symmetric with all nonzero coefficients within a fixed range.

The resulting weight for i from $-p$ to p can be expressed as follows:

$$w_i = \frac{315((r-1)^2 - i^2)(r^2 - i^2)((r+1)^2 - i^2)(3r^2 - 16 - 11i^2)}{8r(r^2 - 1)(4r^2 - 1)(4r^2 - 9)(4r^2 - 25)}$$

Here, $r = p + 2$. Henceforth, H_{2p+1} denotes the Henderson symmetric moving average filter of order $2p+1$.

For the end points where not enough data points are available for symmetric moving average, asymmetric moving averages are used. Musgrave moving averages are used in place of Henderson symmetric moving averages, and certain moving average filters chosen by the US Census Bureau are used in place of composite moving averages. Asymmetric moving averages are not pursued because they are made less important by regARIMA forecasts and backcasts in the X-12 ARIMA method.

The following program and **Output 8.2** illustrate Henderson weights for $p = 3, 6, 9, 12$:

```

data henderson;
  do p = 3, 6, 9, 12 ; jitter = (p-9)/30;
    r = p+2; r2=r*r;
    do i=-p to p; i2=i*i;
      num = 315*((r-1)**2-i2)*(r2-i2)*((r+1)**2-i2)*(3*r2-16-11*i2);
      den = 8*r*(r2 - 1)*(4*r2-1)*(4*r2-9)*(4*r2-25);
      henderson = num/den;
      output;
    end;
  end;
run;

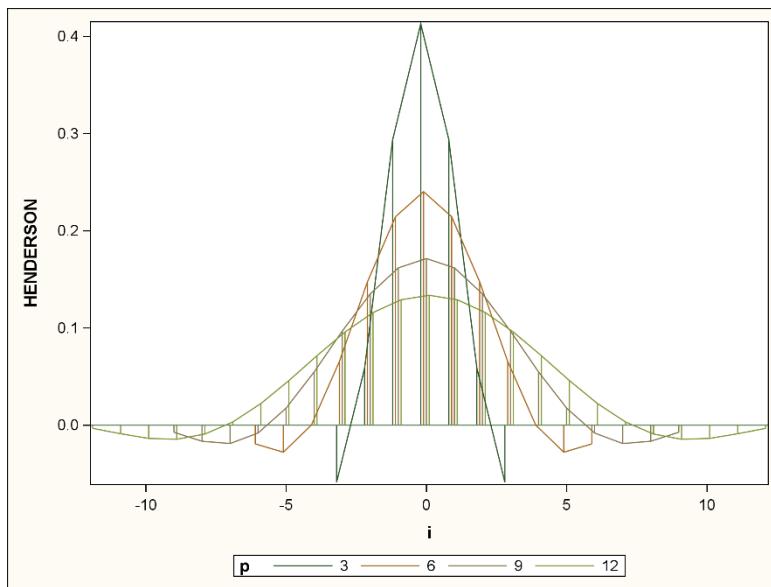
title "CHECK: WEIGHTS SHOULD SUM TO 1";
proc means sum;
  var henderson;
  by p;
run;

proc print; run;

data jitter;
  set henderson;
  i=i+jitter;
run;

proc sgplot;
  needle y=henderson x=i/group=p lineattrs=(pattern=solid);
  series y=henderson x=i/group=p lineattrs=(pattern=solid);
  title "Henderson Weights for p = 3, 6, 9, 12";
run;

```

Output 8.2: Henderson Weights for $p = 3, 6, 9, 12$ **8.2.2 Outline of the X-11 Method**

The X-11 method estimates the components by repeatedly applying moving average filters and adjustments to obtain a refined estimate of the seasonally adjusted series and other components. Letting (M) denote the multiplicative model and (A) the additive model, a monthly series is considered. With the X-11 method, there is a block of calculations repeatedly applied with a few modifications. This basic algorithm of the X-11 method is presented in the following steps:

1. Initial estimates
 - a. Initial trend component
 $T_t^{(1)} = M_{2 \times 12}(Y_t)$
 - b. Initial seasonal-irregular (SI) component
 (A): $SI_t^{(1)} = Y_t - T_t^{(1)}$ (M): $SI_t^{(1)} = Y_t / T_t^{(1)}$
 - c. Initial preliminary seasonal component
 $\tilde{S}_t^{(1)} = M_{3 \times 3}(SI_t^{(1)})$
 - d. Initial seasonal component
 $S_t^{(1)} = \tilde{S}_t^{(1)} - M_{2 \times 12}(\tilde{S}_t^{(1)})$
 - e. Initial irregular component
 (A): $I_t^{(1)} = SI_t^{(1)} - S_t^{(1)}$ (M): $I_t^{(1)} = SI_t^{(1)} / S_t^{(1)}$
 - f. Initial seasonally adjusted series
 (A): $\tilde{Y}_t^{(1)} = Y_t - \tilde{S}_t^{(1)}$
 (M): $\tilde{Y}_t^{(1)} = Y_t / \tilde{S}_t^{(1)}$
2. Seasonal components and seasonally adjusted series
 - a. Intermediate trend component
 $T_t^{(2)} = H_{13}(\tilde{Y}_t^{(1)})$
 - b. Intermediate seasonal-irregular (SI) component
 (A): $SI_t^{(2)} = Y_t - T_t^{(2)}$ (M): $SI_t^{(2)} = Y_t / T_t^{(2)}$
 - c. Preliminary seasonal component
 $\tilde{S}_t^{(2)} = M_{3 \times 5}(SI_t^{(2)})$
 - d. Seasonal component
 $S_t^{(2)} = \tilde{S}_t^{(2)} - M_{2 \times 12}(\tilde{S}_t^{(2)})$

- e. Intermediate irregular component

$$(A) : I_t^{(2)} = SI_t^{(2)} - S_t^{(2)} \quad (M) : I_t^{(2)} = SI_t^{(2)} / S_t^{(2)}$$
 - f. Seasonally adjusted series

$$\tilde{Y}_t^{(2)} = Y_t - S_t^{(2)}$$
3. Final trend and irregular component
- a. Trend component

$$T_t^{(3)} = H_{13}(\tilde{Y}_t^{(2)})$$
 - b. Irregular component

$$(A) : I_t^{(3)} = \tilde{Y}_t^{(2)} - T_t^{(3)} \quad (M) : I_t^{(3)} = \tilde{Y}_t^{(2)} / T_t^{(3)}$$

The actual order of the Henderson symmetric moving average filters and seasonal moving average filters in step 2.d might be different because they are chosen by the data. The first two irregular components, $I_t^{(1)}$ and $I_t^{(2)}$, are used to detect outliers and adjust for outliers to obtain a refined estimate of the seasonal components. The last irregular component, $I_t^{(3)}$, is used to adjust for outliers and calendar effects, such as trading day and holidays.

Now, the X-11 method in the X13 procedure consists of seven parts from part A to part G.

- Part A: Preadjustment
- Part B: Preliminary seasonal adjustment
- Part C: Second seasonal adjustment
- Part D: Final seasonal adjustment
- Part E: Analytical tables
- Part F: Diagnostics
- Part G: Spectral plots

In part A, users can supply user-defined factors and predefined factors such as trading day and holiday effects to adjust the raw series. If there is no input from users, the raw series is passed without modification. The resulting series is B1.

In part B, the basic algorithm is applied to B1, and calendar effects are estimated. Then, B1 is adjusted for the effects. The resulting series is C1.

In part C, the basic algorithm is applied to C1, and refined estimates of effects are obtained. Then, B1 is adjusted with the new estimates. This is series D1.

In part D, the basic algorithm is applied to D1 to obtain final estimates.

This is a high-level overview of the X-11 method. In each part, there might be slight variations of the basic algorithm. See the SAS documentation for the X13 procedure for more information.

8.2.3 Basic Seasonal Adjustment Using the X-11 Method

The X-11 additive method with the X13 procedure is illustrated by the following SAS code:

```
proc x13 data=wh.sales date=date interval=month;
  var sales;
  x11 mode=add;
  output out=out1 a1 d10 d11 d12 d13;
run;
```

A long list of outputs is produced, showing intermediate calculations and diagnostics. The MODE=ADD option invokes the additive model. If this option is not specified, a multiplicative model is assumed by default. The OUTPUT statement generates an output data set containing the following series:

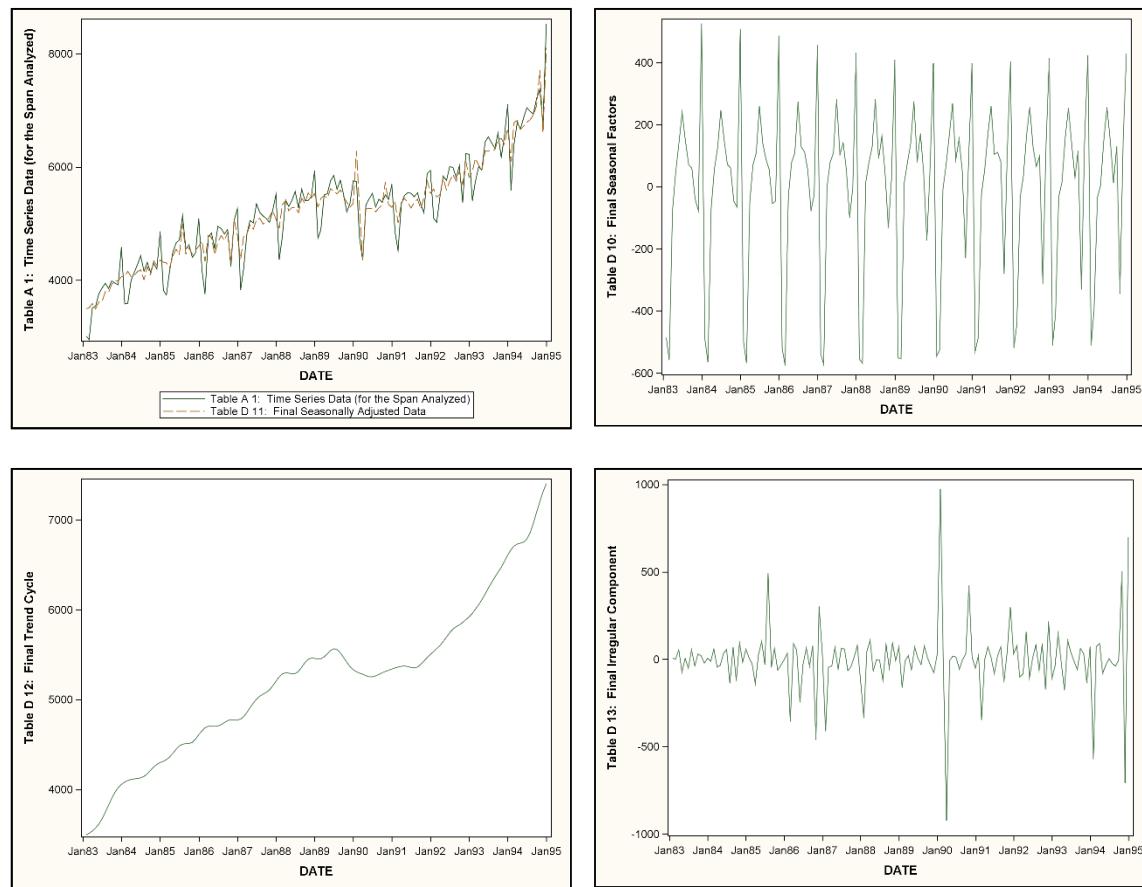
- A1: Original series Y_t
- D10: Final seasonal factors S_t
- D11: Seasonally adjusted series $\tilde{Y}_t = T_t + I_t$

- D12: Trend-cycle component T_t
- D13: Irregular component I_t

A full list of available table names and descriptions can be found in the *SAS/ETS User's Guide*. **Output 8.3** displays the seasonal decomposition of the North Carolina retail sales. The top-left panel shows the original series together with the seasonally adjusted series. The top-right panel shows the seasonal factor. The bottom-left panel shows the trend-cycle component. And, the bottom-right panel shows the irregular component.

There is a temporary decrease in the trend-cycle during the early 1990s combined with the highly volatile irregular component. During the period, there was an economic recession due in most part to the savings and loan crises (S&L crises). In addition, the Gulf War broke out in August of 1990. Also, the rapid growth of the trend component and high volatility of the irregular component at the end of the series might indicate the advent of the dot-com bubble. By decomposing the series and removing the seasonality factor, you can observe the pattern more clearly.

Output 8.3: Basic Seasonal Decomposition of the NC Retail Sales Data



8.2.4 Tests for Seasonality

One approach to model seasonality is to assume stable seasonal effects over time. For a monthly series, under the stability assumption, the seasonal component in a month of this year is the same as last year. Another approach is to assume that the seasonal component can randomly change year after year. This changing seasonality is called *moving seasonality*. The model underlying the X-11 method, by its nonparametric and local nature, includes moving seasonality.

Output 8.4 shows the seasonality test portion from the output of the North Carolina retail sales data ([section 8.4.3](#)) produced by the following code:

```
proc x13 data=sales date=date interval=month;
  ods select d8a;
  var sales;
```

```
x11 mode=add;
output out=d8 d8;
run;
```

The ODS SELECT statement comes in handy when only a specific portion of large output is needed. The program is requesting table D8A (*F* tests for Seasonality) to be printed and series D8 (SI component) to be output to a data set named D8.

Output 8.4: Tests for Seasonality on the North Carolina Retail Sales Data

Test for the Presence of Seasonality Assuming Stability					
	Sum of Squares	DF	Mean Square	F-Value	
Between Months	12379092	11	1125372	27.82921	**
Residual	5337885	132	40438.52		
Total	17716976	143			

** Seasonality present at the 0.1 percent level.

Nonparametric Test for the Presence of Seasonality Assuming Stability		
Kruskal-Wallis Statistic	DF	Probability Level
95.96753	11	.00%

Seasonality present at the one percent level.

Moving Seasonality Test					
	Sum of Squares	DF	Mean Square	F-Value	
Between Years	212874.9	11	19352.27	0.746689	
Error	3136012	121	25917.45		

No evidence of moving seasonality at the five percent level.

Summary of Results and Combined Test for the Presence of Identifiable Seasonality	
Seasonality Tests:	Probability Level
Stable Seasonality F-test	0.000
Moving Seasonality F-test	0.692
Kruskal-Wallis Chi-square Test	0.000
Combined Measures:	
T1 = 7/F_Stable	0.25
T2 = 3*F_Moving/F_Stable	0.08
T = (T1 + T2)/2	0.17
Combined Test of Identifiable Seasonality:	
	Present

The X13 procedure uses the SI component to conduct the seasonality test. The first two tables correspond to two seasonality tests under the assumption of stable seasonality. In the simple linear regression context, they are equivalent to testing if month of year is a significant predictor of the SI component. To see this, run the following code to get **Output 8.5**:

```
data d8;
  set d8;
  abs_d8=abs(sales_d8);
  month=month(date);
  year=year(date);
run;

proc glm data=d8;
  ods select overallanova;
  class month;
  model sales_d8 = month;
run;
quit;

proc nparlway data=d8 wilcoxon;
  ods select kruskalwallistest;
  class month;
  var sales_d8;
run;
```

Output 8.5: F and Kruskal-Wallis Test for Seasonality under Stability Assumption (New)

The GLM Procedure

Dependent Variable: SALES_D8 Table D 8: Final Unmodified SI Ratios

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	11	12379091.95	1125372.00	27.83	<.0001
Error	132	5337884.53	40438.52		
Corrected Total	143	17716976.48			

The NPAR1WAY Procedure

Kruskal-Wallis Test	
Chi-Square	95.9675
DF	11
Pr > Chi-Square	<.0001

The two tables in **Output 8.5** match the first two tables in **Output 8.4**. The test for the presence of seasonality assuming stability is a one-way ANOVA testing the statistical significance of month of year on the SI component. The Kruskal-Wallis test is a nonparametric counterpart of the *F* test in a one-way ANOVA.

The moving seasonality test (the third table in **Output 8.4**) is based on a two-way ANOVA where the absolute value of the SI component is the dependent variable and two independent variables—year and month of year. The *F* test on the effect of year is equivalent to the moving seasonality test. The following code produces **Output 8.6**:

```
proc glm data=d8;
  ods select overallanova "Type III Model ANOVA";
  class month year;
  model abs_d8 = month year;
run;
quit;
```

Output 8.6: Moving Seasonality Test Using PROC GLM**The GLM Procedure****Dependent Variable: abs_d8**

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	22	5121522.265	232796.467	8.98	<.0001
Error	121	3136011.920	25917.454		
Corrected Total	143	8257534.185			

Source	DF	Type III SS	Mean Square	F Value	Pr > F
month	11	4908647.319	446240.665	17.22	<.0001
year	11	212874.946	19352.268	0.75	0.6918

The row for **Between Years** in the moving seasonality table in **Output 8.4** is equal to the row for **Year** in the Type III SS table in **Output 8.6**, and rows for **Error** match.

The X13 procedure uses a combined test using the three test statistics to determine the presence of seasonality. In this example, the stable seasonality test has a small *p*-value (less than 0.001). The moving seasonality test is conducted. The *p*-value for moving seasonality is large (greater than or equal to 0.05). It is decided that there is no significant moving seasonality. Then, $T_1=7/27.82921=0.25$ is less than 1 and $T_2=3 \times 0.746689/27.82921=0.08$ is less than 1, so the Kruskal-Wallis test is conducted. The *p*-value is less than 0.001, and it is determined that identifiable seasonality exists. The full flow diagram of the decision process is illustrated in the *SAS/ETS User's Guide* in the section, "Combined Test for the Presence of Identifiable Seasonality."

8.3 regARIMA Models and TRAMO

The X-11 program was based on intuitively reasonable but ad hoc local smoothing methods using moving averages. In that sense, it was not model based. Because regression and ARIMA models have been well studied, incorporating their functionality in seasonal adjustment is appealing.

8.3.1 regARIMA Models

The regression-ARIMA (regARIMA) models are seasonal ARIMA models with input series. The general form of $\text{ARIMA}((p, d, q) \times (P, D, Q)s)$ models with k input series can be stated as follows:

$$\phi(B)\Phi(B^s)(1-B)^d(1-B^s)^D \left(f(Y_t) - \sum_{j=1}^k \beta_j X_{jt} \right) = \theta(B)\Theta(B^s)\varepsilon_t$$

where

$$\begin{aligned}\phi(B) &= (1 - \phi_1 B - \phi_2 B^2 - \cdots - \phi_p B^p) \\ \Phi(B^s) &= (1 - \Phi_1 B^s - \Phi_2 B^{2s} - \cdots - \Phi_p B^{ps}) \\ \theta(B) &= (1 - \theta_1 B - \theta_2 B^2 - \cdots - \theta_q B^q) \\ \Theta(B^s) &= (1 - \Theta_1 B^s - \Theta_2 B^{2s} - \cdots - \Theta_q B^{qs})\end{aligned}$$

and ε_t is a white noise series. The X13 procedure offers several options for the transformation function $f(\cdot)$ such as log, square root, inverse, logistic, and Box-Cox transformation. You can choose not to transform the response.

The expression might look daunting at first glance, but ARIMA $((0, 1, 1) \times (0, 1, 1)_{12})$ without input series and transformation of the response series is the familiar airline model:

$$(1 - B)(1 - B^{12})Y_t = (1 - \theta_1 B)(1 - \Theta_1 B^{12})\varepsilon_t$$

There are many benefits of using regARIMA models in seasonal adjustment. First, regARIMA models provide a unified treatment of deterministic effects and outlier effects. The X-11 method adjusts for additive outliers (AO), which affect only a single time point several times through the estimation stages. On the other hand, regARIMA models offer ways to handle level shifts (LS), which are permanent shifts in the level of a series, and to handle temporary change outliers (TC), which are outliers that change the level of the series and return to the original level in a short period of time.

Second, the model allows extension of the series with forecasts and backcasts. The X-11 method uses asymmetric moving averages to seasonally adjust points near the ends of the series because there are not enough observations to apply a symmetric moving average, centered on the point of interest, all the way through the last data point. The coefficients for the asymmetric moving averages corresponding to Henderson symmetric moving average filters are derived using a reasonable criterion. With the extended series, symmetric moving averages are available at all points. That is, enough ARIMA-based forecasts are added to the end of the series (and backcasts are added to the beginning) that a symmetric filter centered on the last (first) data point has enough data to be applied. Studies show that this method reduces revisions on average, a measure of change in estimates with an addition of new data.

Last, regARIMA models enable estimation of deterministic effects directly from the response series. The X-11 method estimates the calendar effects by means of ordinary least squares (OLS) on the irregular component. OLS is valid because the irregular component consists of these calendar effects plus an almost uncorrelated series. However, it creates a complication in that the trend and seasonality have to be removed from the regression model. In X-11, certain adjustments are made to account for this issue, but regARIMA models provide a more natural-model-based treatment of the problem.

8.3.2 Automatic Selection of ARIMA Orders

TRAMO is an automatic routine developed by the Bank of Spain to select an optimal regARIMA model and preadjust the original series for deterministic effects, outliers, and missing observations. Then, seasonal adjustment methods such as X-11 or SEATS can be applied to obtain a seasonally adjusted series. The X13 procedure uses a similar routine based on TRAMO. Recall the notation for ARIMA orders: ARIMA $((p, d, q) \times (P, D, Q)_s)$. By default, the maximum order is 2 for the non-seasonal difference (d), 1 for the seasonal difference (D), 3 for the non-seasonal ARMA orders (p and q), and 1 for the seasonal ARMA orders (P and Q).

The orders of the non-seasonal and seasonal differences are first selected. With the orders of differences fixed, goodness of fit statistics are calculated for all possible models with the maximum order constraints for seasonal and non-seasonal ARIMA orders. Then, the best model is selected, and a final check for the model is run to check for unit roots, non-seasonal overdifferencing, and insignificant ARMA coefficients. If the selected model passes all the tests, the unknown parameters are estimated with the maximum likelihood method on this model. This complex model selection can be requested by simply adding the AUTOMDL statement as shown in the next section.

8.4 Data Examples

In this section, applications of the X13 procedure on real data examples are illustrated.

8.4.1 Airline Passengers Revisited

The airline series that was modeled for forecasting in [Chapter 4](#) had a strong seasonal component. It also serves as a great example of seasonal adjustment. Depending on the application, repeated up and down seasonal movement could mask subtle features and be viewed as a distraction to be removed.

The international airline passengers data was displayed in [Output 8.1](#). To decompose and seasonally adjust this data, the X13 procedure is invoked by the following code:

```
proc x13 data=wh.airline date=date interval=month;
  var air;
  transform function=auto;
  regression predefined=td;
  outlier;
  automdl;
```

```

x11 final=all;
output out=out2 a1 d10 d11 d12 d13;
run;

```

The TRANSFORM statement can be used to transform the response. The FUNCTION=AUTO option tells the procedure to automatically choose between a log-transformation or no transformation. If the log-transformation is chosen, a multiplicative model is assumed. Otherwise, an additive model is assumed.

The REGRESSION statement can be used to list explanatory variables to be added to the regARIMA model. Here, trading-day effects are included. The idea of trading-day effects is to take into account the number of occurrences of each day of the week as predefined regressors. For example, if a retailer tends to experience high sales on Saturdays, then a month with five Saturdays would show higher sales than one with four Saturdays.

The OUTLIER statement can be used to automatically detect different types of outliers. If outliers are detected, appropriate regression variables are created and incorporated into the regARIMA model. The model is reestimated. This process is repeated until no outlier is found. By default, the largest outliers are accounted for in the model one at a time. The METHOD=ADDALL option can be specified to remove the effects of all identified outliers at each iteration, rather than removing them one at a time.

The AUTOMDL statement requests the procedure to automatically select ARIMA orders and preadjust the raw series using the routine based on TRAMO.

Last, the FINAL=ALL option in the X11 statement tells the procedure to remove outliers and effects of user-defined regressors from the final seasonally adjusted series.

Output 8.7 shows the choice of the transformation for the response series. Based on AICC, the log-transformation is selected. A multiplicative model is assumed.

Output 8.8 shows the results of the automatic ARIMA order selection. An ARIMA((0,1,0) × (0,1,1)12) model is selected. This is slightly different from the airline model, whose order of non-seasonal MA is 1 rather than 0.

Output 8.9 shows the estimated effects for trading day and outliers and the chi-square test for the combined trading-day effects. One additive outlier has been detected in May, 1951. There is no sign of level shifts and temporary changes. The AO effect is significant, but only Thursday and Friday have significant effects at the 10% level. The null hypothesis of the chi-square is that all trading-day effects are 0. The null hypothesis is rejected. At least one of the effects is significant.

After the regARIMA model is selected and effects are estimated, the raw series is adjusted for the effects and extended using the selected model. Also, the underlying input series for trading day is extended. Then, the preadjustment is complete. The adjusted series is decomposed by the X-11 method.

Output 8.10 displays the resulting seasonal decomposition. The structure of the plots is the same as **Output 8.3**. There is a steady, increasing trend-cycle. The magnitude of the seasonal factors has been steadily increasing, which is an expected phenomenon for multiplicative models. Curiously, there is high volatility around May, 1951, which is the time point identified as an additive outlier. The Korean War occurred between June 1950 and July 1953. April and May of 1951 were the times when the war was most heated before entering a stalemate in July 1951. The war events might have caused the disruption in the series.

Output 8.7: Choice of Transformation

Results of Automatic Transformation Selection	
For Variable AIR	
AICC (with AICDIFF=-2.00) prefers	Log(AIR)
Adjustment will be	Multiplicative

Output 8.8: Automatic ARIMA Order Selection

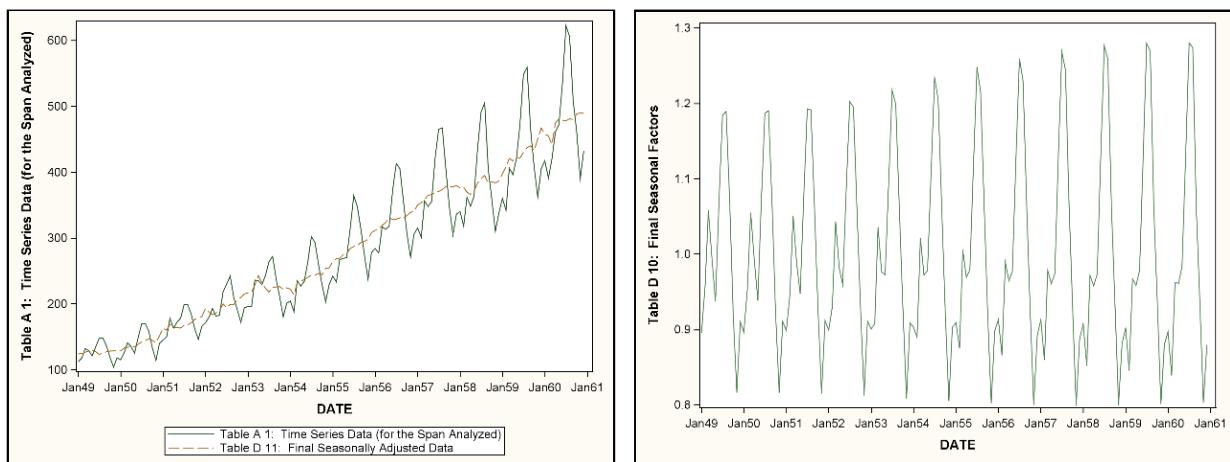
Final Automatic Model Selection		
For Variable AIR		
Source of Model	Orders Altered	Estimated Model
Automatic Model Choice	No	(0,1,0) (0,1,1)

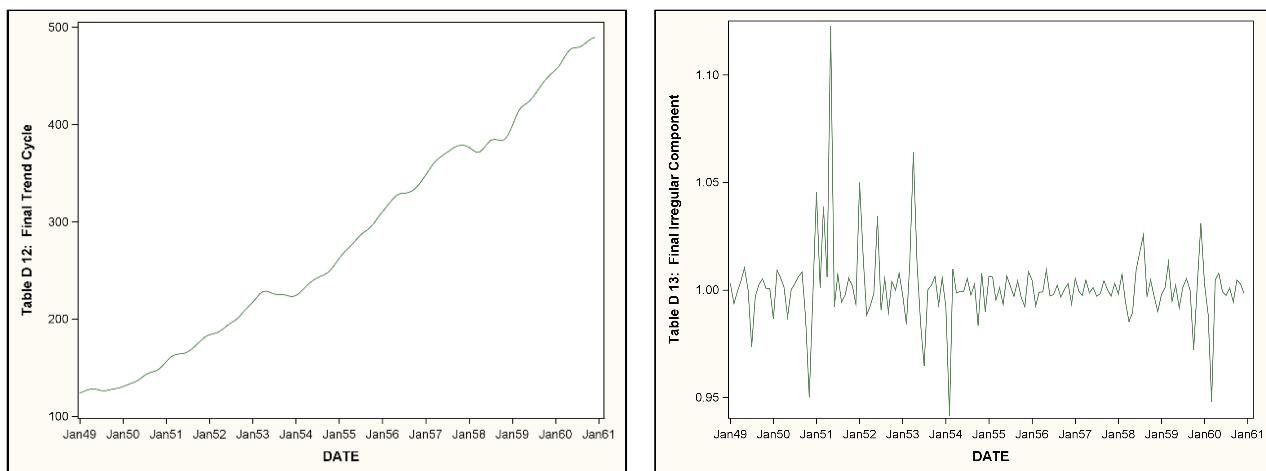
Output 8.9: Estimated Trading-Day and Outlier Effects

Regression Model Parameter Estimates						
For Variable AIR						
Type	Parameter	NoEst	Estimate	Standard Error	t Value	Pr > t
Trading Day	MON	Est	-0.00226	0.00355	-0.64	0.5258
	TUE	Est	-0.00494	0.00354	-1.39	0.1657
	WED	Est	-0.00298	0.00349	-0.85	0.3946
	THU	Est	-0.00633	0.00351	-1.81	0.0734
	FRI	Est	0.00607	0.00351	1.73	0.0861
	SAT	Est	0.00498	0.00359	1.39	0.1681
	SUN(derived)*	Est	0.00546	0.00364	1.50	0.1357
Automatically Identified	AO MAY1951	Est	0.11857	0.01981	5.98	<.0001

*Note: For trading-day and fixed seasonal effects, the derived parameter estimate is obtained indirectly as minus the sum of the directly estimated parameters that define the effect.

Chi-squared Tests for Groups of Regressors			
For Variable AIR			
Regression Effect	DF	Chi-Square	Pr > ChiSq
Trading Day	6	47.6870	<.0001

Output 8.10: Seasonal Decomposition of the International Airline Passengers Data



8.4.3 Employment in the United States

The power of the X13 procedure lies in its fully automated schemes. It is especially useful when an investigator needs to process many series, although only some of them show certain characteristics of interest. Several U.S. employment series are used as examples. The data downloaded from the Bureau of Labor Statistics contains 918 monthly series for the number of employees across industries. The data covers most of the non-agricultural industries, both private and public, including construction; telecommunication; data processing; hosting and related services; finance and insurance; gambling; accommodation; human rights organizations; federal, state, and local government; and US postal services. The data range of the series differs by industry. Data between January 2001 and May 2016 are taken to enforce an equal range for all series.

Among 918 series, the X13 procedure chooses the additive model for 839 series or 91.4%. Significant seasonality is detected for 659 or 71.8% of the series. Among the 659 series that exhibit significant seasonality, 92.1% choose the additive model.

The X13 procedure consists mainly of two phases: the preadjustment phase and the seasonal decomposition phase. In the preadjustment phase, regARIMA models and the automatic model selection scheme based on TRAMO are used to adjust the series for calendar effects, user-defined inputs, and outliers so that the preadjusted series is thought to be composed of only three components: trend-cycle, seasonal, and irregular components. In the seasonal decomposition phase, users can choose between the enhanced X-11 method and SEATS.

At the end of the preadjustment phase, forecasts are produced based on the automatically selected ARIMA model. These forecasts are referred to as X-13 forecasts. Individual components after seasonal decomposition can be used to produce forecasts. One way to do this is:

1. Use the estimated trend series to fit an autoregressive model with a cubic trend. The maximum number of lags for autoregressive errors is set to 13. A backward selection with 10% significance level on the number of lags is used.
2. If an additive model is used, add the forecasted seasonal factors and other effects such as calendar effects to obtain forecasts on the original scale. If a multiplicative model is used, multiply the effects.

Forecasts from this method are referred to as X13-Autoreg-Trend forecasts.

The root mean squared error (RMSE) and mean absolute error (MAE) were computed using a 12-month holdout sample. For around 60% of the 918 series considered, the X-13 forecasts outperformed the X13-Autoreg-Trend forecasts. Also, 95% forecast bands from the X-13 forecasts captured all 12 data points in the holdout sample for more than 75% of the series.

Consider one of the individual series, which contains employment data for the nuclear power plant and other electric power generation industries. The X13 procedure is used to analyze this series with the following code:

```
proc x13 data=nuclear date=date seasons=12;
  var value;
  regression predefined=(td);
  transform function=auto;
```

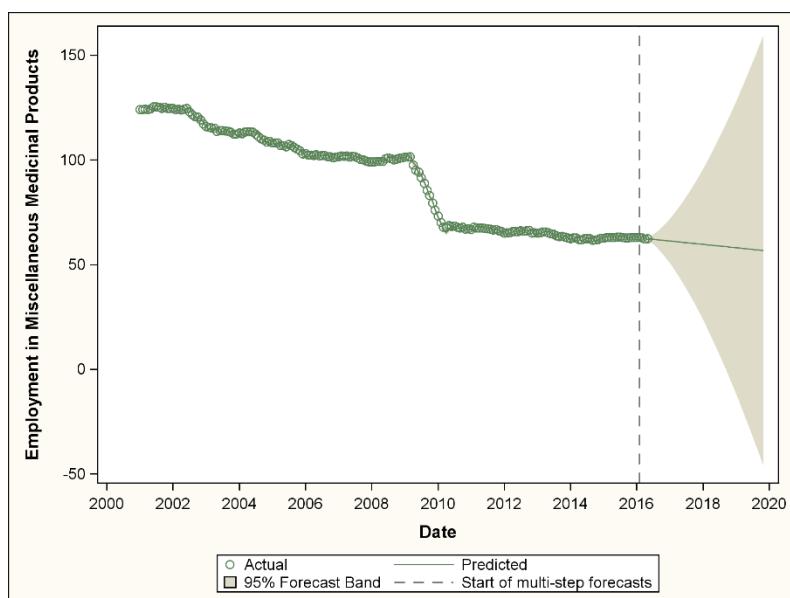
```

automdl maxdiff=(2,1);
outlier type=(ao ls);
forecast lead=42 nbackcast=42;
x11 outforecast final=all;
output out=out3 a1 d10 d11 d12 d13 d18;
run;

```

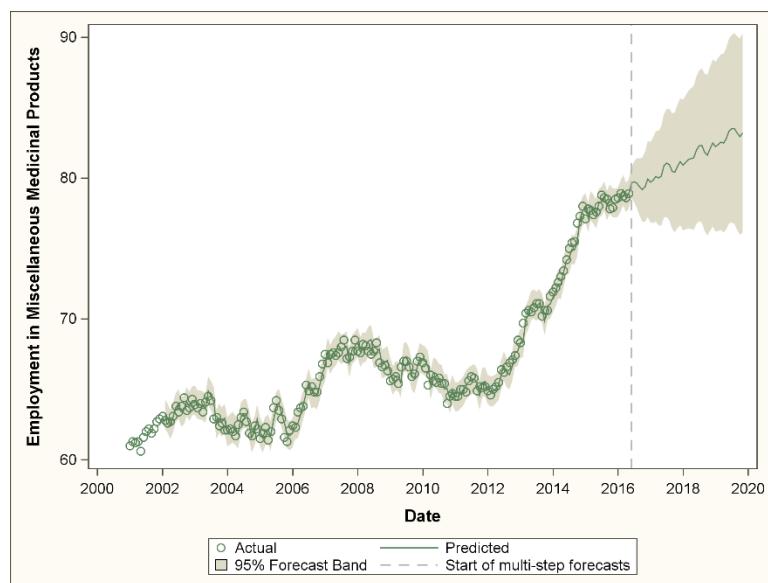
Output 8.11 shows the forecast plot using the automatically selected model, ARIMA(0,2,1)(0,0,0)12. An additive model is selected by the transformation test, and no significant seasonality is identified. Nevertheless, the automatic model selection and forecasts produced by the X13 procedure can still be used because they are done in the preadjustment phase and no seasonal decomposition is involved at this phase. The employment series has been decreasing. A sharp downward trend starts around 2009, and a level shift outlier is detected by the X13 procedure for April 2009. The series might be suggesting that the United States is either slowly moving away from the nuclear power plants or using automation to reduce the number of jobs for the industry. The increasingly wide confidence band on the original series is a typical phenomenon for a model with second-order differencing. This is one of the reasons why high-order differencing is typically avoided in time series analysis.

Output 8.11: Forecast Plot of Employment (in Thousands) in the Nuclear Power Plant and in Other Electric Power Generation Industries

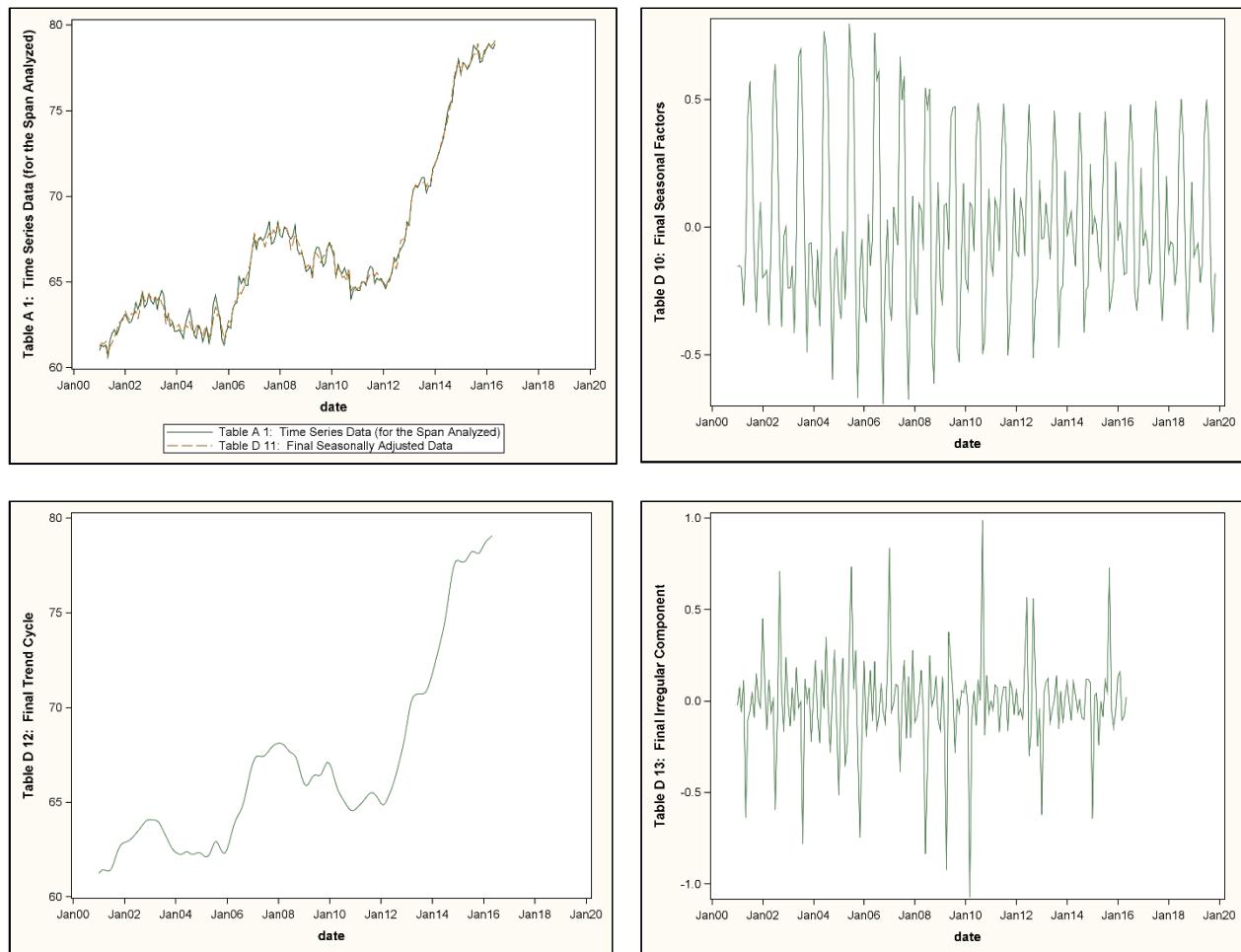


Another industry that shows relatively low RMSE is miscellaneous medicinal and biological products. The X13 procedure chose an ARIMA(0,1,0)(0,1,1)12 model for the series. A significant seasonality is detected, but no significant trading-day or leap-year effect is found. The forecast plot in **Output 8.12** shows that the employment in the industry is expected to continue to grow. **Output 8.13** shows the seasonal decomposition of the series. In the bottom left panel, there is rapid growth of employment since 2012, becoming less rapid around 2015.

Output 8.12: Forecast Plot of Employment (in Thousands) in the Miscellaneous Medicinal and Biological Products Industry

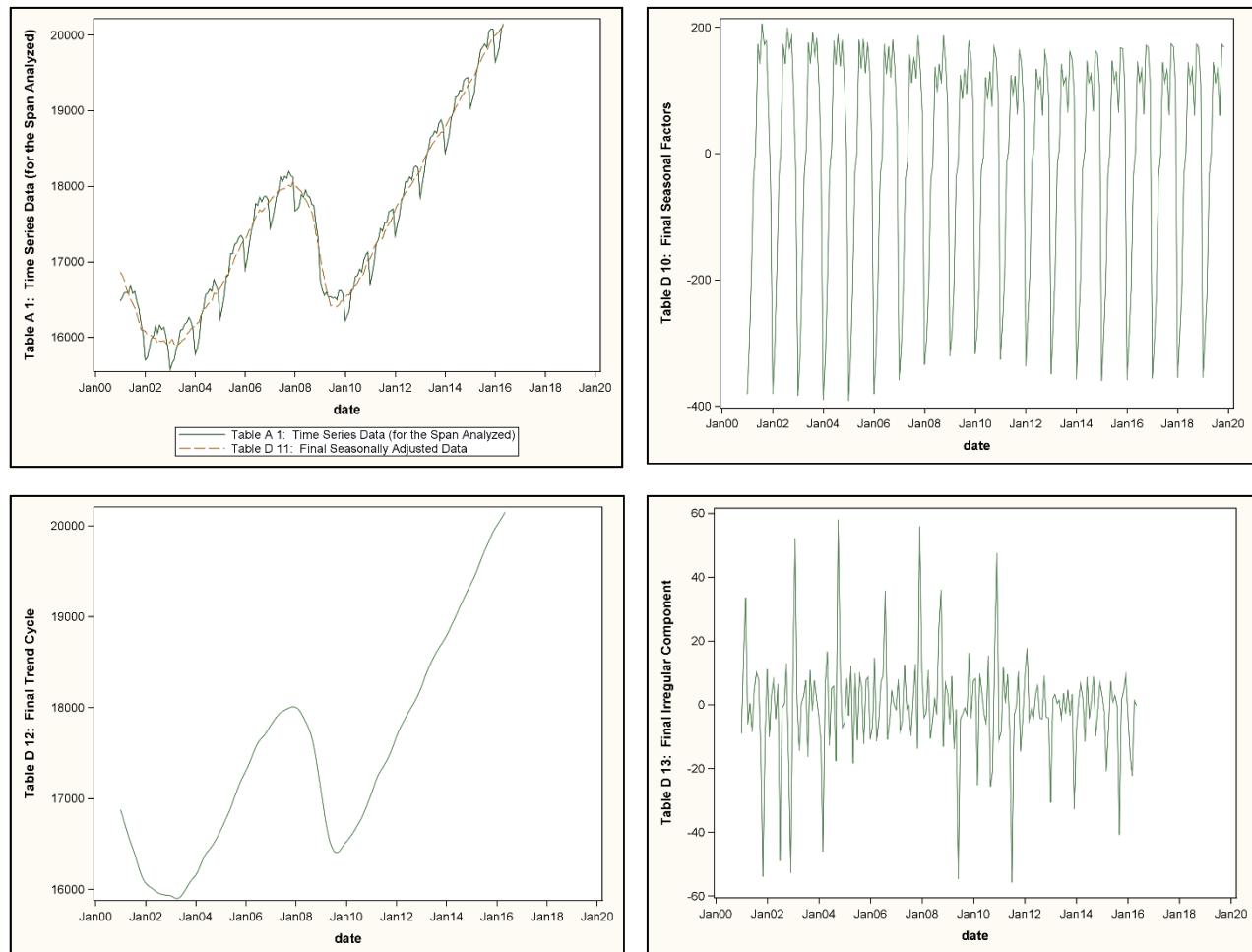


Output 8.13: Seasonal Decomposition of the Employment in the Miscellaneous Medicinal and Biological Products Industry



One of the industries that show relatively high RMSE is professional and business services. The X13 procedure selected an ARIMA(2, 2, 0)(0, 1, 1)12 additive model, but detected no outlier. **Output 8.14** shows the seasonal decomposition of the series. Since January 2010, there has been a steady growth of employment in the industry. Unlike employment in the nuclear plant industry, there are clear seasonal patterns. The employment is usually the highest in October and lowest in January. There are significant trading-day effects. If there are more Fridays (Sundays) in a given month, employment is higher (lower).

Output 8.14: Seasonal Decomposition of the Employment in the Professional and Business Services Industry



Output 8.15: Trading-Day and Leap-Year Effects on Employment in the Professional and Business Services Industry

Regression Model Parameter Estimates							
For Variable value							
Type	Parameter	NoEst	Estimate	Standard Error	t Value	Pr > t	
Trading Day	MON		Est	-1.57569	3.79205	-0.42	0.6783
	TUE		Est	-2.99945	3.85249	-0.78	0.4374
	WED		Est	0.45180	3.85700	0.12	0.9069
	THU		Est	-0.80586	3.89681	-0.21	0.8364
	FRI		Est	11.50685	3.84308	2.99	0.0032
	SAT		Est	1.09587	3.87882	0.28	0.7779

Regression Model Parameter Estimates						
For Variable value						
Type	Parameter	NoEst	Estimate	Standard Error	t Value	Pr > t
	SUN(derived)*	Est	-7.67353	3.76032	-2.04	0.0429
Leap Year	Leap Year	Est	-4.50332	12.92468	-0.35	0.7280

Chapter 9: SAS Forecast Studio

9.1 Introduction.....	305
9.2 Creating a Project	305
9.3 Overview of Available Modes.....	310
9.4 Project Settings.....	312
9.4.1 Model Generation	312
9.4.2 Goodness of Fit and Honest Assessment.....	313
9.4.2 Transformation and Outlier Detection	314
9.5 Creating Custom Events	318
9.6 Hierarchical Time Series and Reconciliation	320

9.1 Introduction

SAS Forecast Server provides a sophisticated framework for analyzing univariate time series. With the software, analysts can process a large number of series through built-in automated routines. Although experienced analysts might be able to study each series carefully and build better models, they might not have enough time to go through each series.

Automatically produced models tend to generate sufficiently good forecasts for most series, so analysts can use their time more efficiently by focusing on problematic series or important series for the business.

SAS Forecast Studio provides a user-friendly GUI. With SAS Forecast Studio, analysts without prior SAS programming knowledge can easily build models and forecast by point and click. Moreover, the underlying SAS programs can be exported so that experienced SAS programmers can modify them and interact with utilities and forecasting tools provided by other SAS products such as Base SAS, SAS/STAT, and SAS/ETS. SAS Forecast Studio uses optimized procedures provided by SAS Forecast Server that can generate millions of forecasts in a reasonable turnaround time. A simpler GUI, the Time Series Forecasting System, is available from a menu in the SAS/ETS package when using the windowing interface in SAS Foundation.

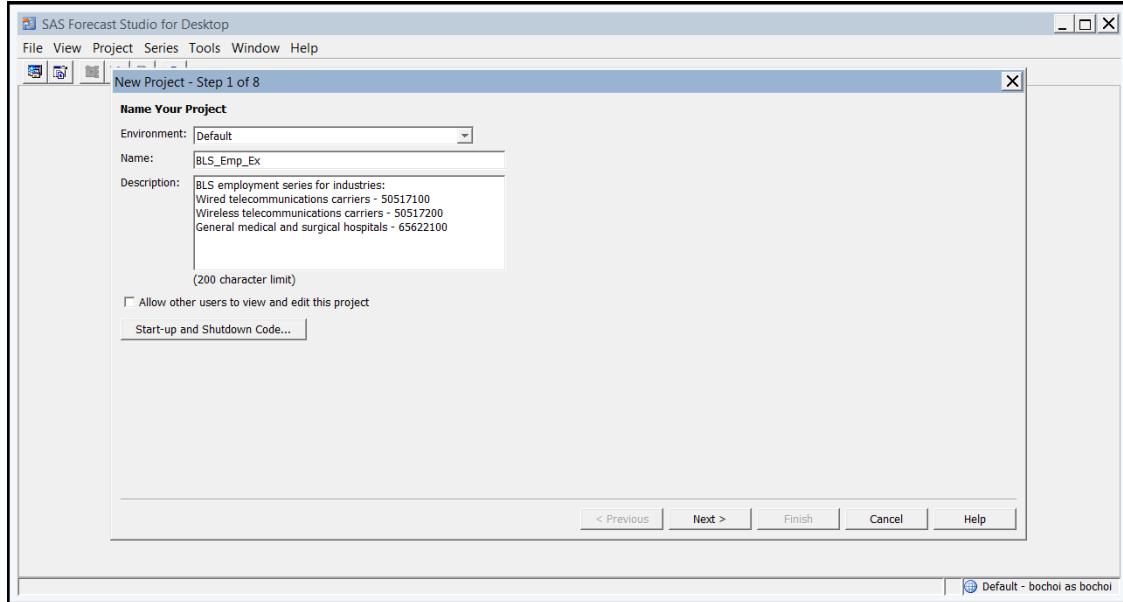
This chapter is designed to provide a basic tutorial and high-level overview of SAS Forecast Studio 14.1. Interested readers can refer to the *SAS Forecast Studio: User's Guide*, *SAS Forecast Server Procedures: User's Guide*, and references therein for more details. Also, the desktop version of the software (SAS Forecasting for Desktop) is used. It has limited functionalities. Some of the limitations are as follows:

- Input data sets cannot include more than 1,000 time series.
- Running the generated code in batch mode is disabled.
- SAS Forecast Server procedures cannot be used outside of the SAS Forecast Studio environment.

The purpose of this chapter is to show the interface. The underlying models are described in previous chapters. Once the basics are illustrated on a few series, scaling up to larger collections of series should be straightforward. These limitations should not hinder the purpose.

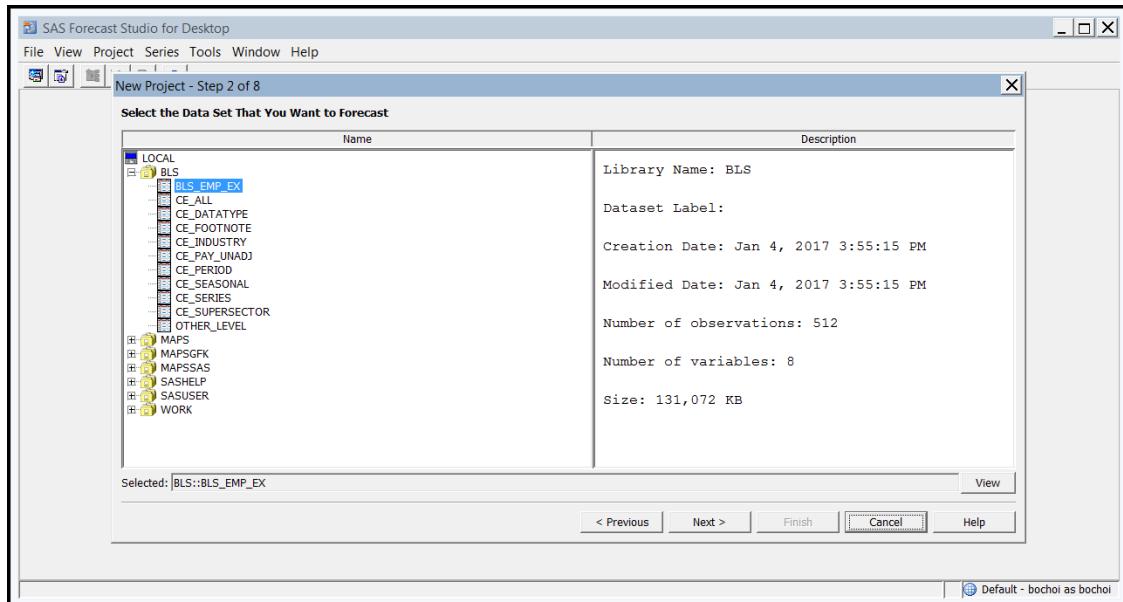
9.2 Creating a Project

Before creating a project, SAS libraries for the data sources must be assigned. Contact your SAS administer. Or, if you have administrative privileges, refer to *SAS Forecast Server: Administrator's Guide*. Start SAS Forecast Studio for Desktop. **Figure 9.1** shows the screen for creating a new project.

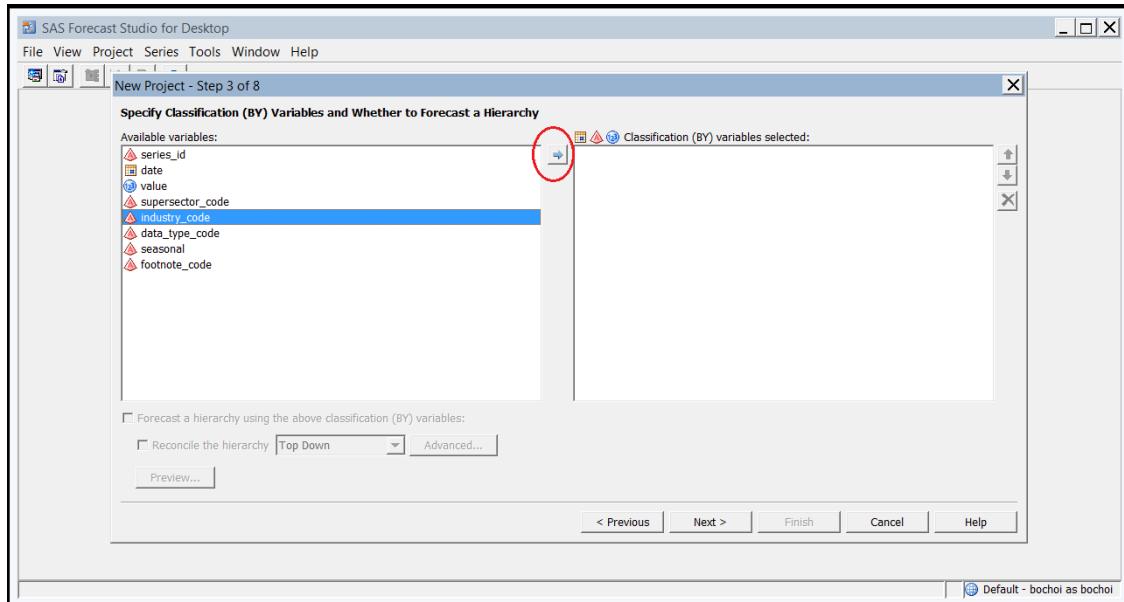
Figure 9.1: Create a New Project

Choose the environment that you want to use for the project. Name the project using a valid SAS name. It must have less than or equal to 32 characters and consist of numbers, letters, and underscores. It cannot start with a number. Last, it is a best practice to write the description in detail so that you can remember or your coworkers can understand what the project is about. When you are ready, click **Next**.

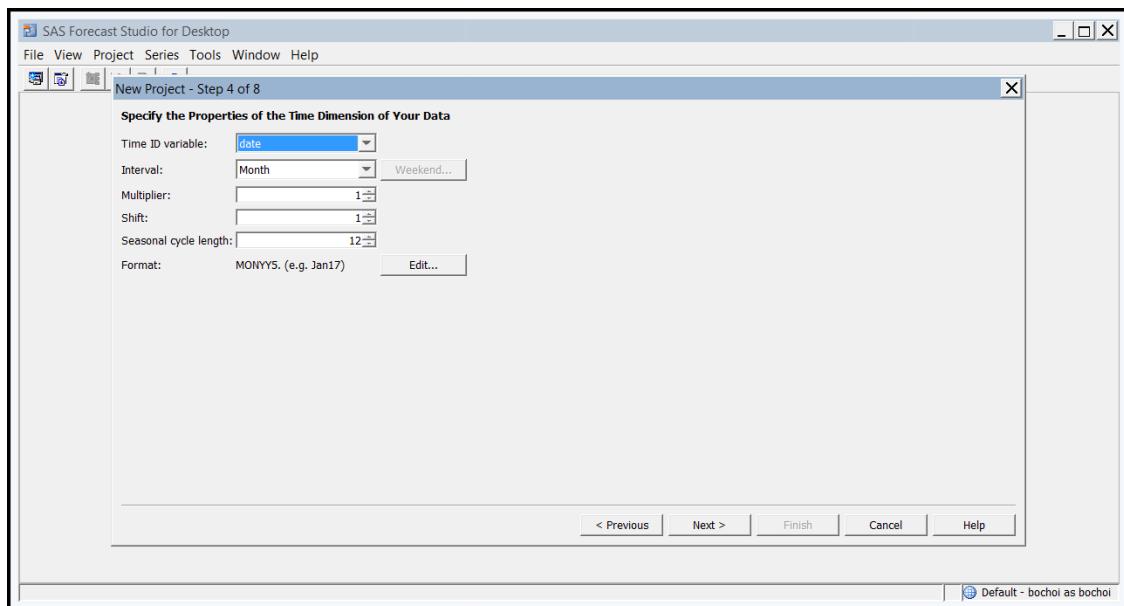
In step 2 (**Figure 9.2**), select a data source that you want to use for the project. For the first tutorial, the data set **BLS_EMP_EX** in the **BLS** library is used. The data was downloaded from the Bureau of Labor Statistics website and is stored in the BLS library. It contains monthly employment series for four industries: petroleum refineries (32324110), wired telecommunications carriers (50517100), wireless telecommunications carriers (50517200), and general medical and surgical hospitals (65622100). All series have an equal range of dates from March 1, 2006 to October 1, 2016. The right panel provides a summary for the selected data set. Click **Next**.

Figure 9.2: Select a Data Source

In step 3 (**Figure 9.3**), click **industry_code**, and then click the right arrow to use the variable as a BY variable. If needed, multiple BY variables can be used. SAS Forecast Studio supports hierarchical series through reconciliation, which is discussed later in the chapter. Uncheck **Forecast a hierarchy using the above classification (BY) variable**.

Figure 9.3: Select the BY Variable

In step 4 (**Figure 9.4**), select the time variable. Using the time ID variable and equal-length intervals defined in this step, with the accumulation statistic selected in the next step, SAS Forecast Studio converts the original series into series that have exactly one value within each interval. To avoid confusion, refer to the converted series and the equal-length intervals by analysis series and analysis intervals, respectively.

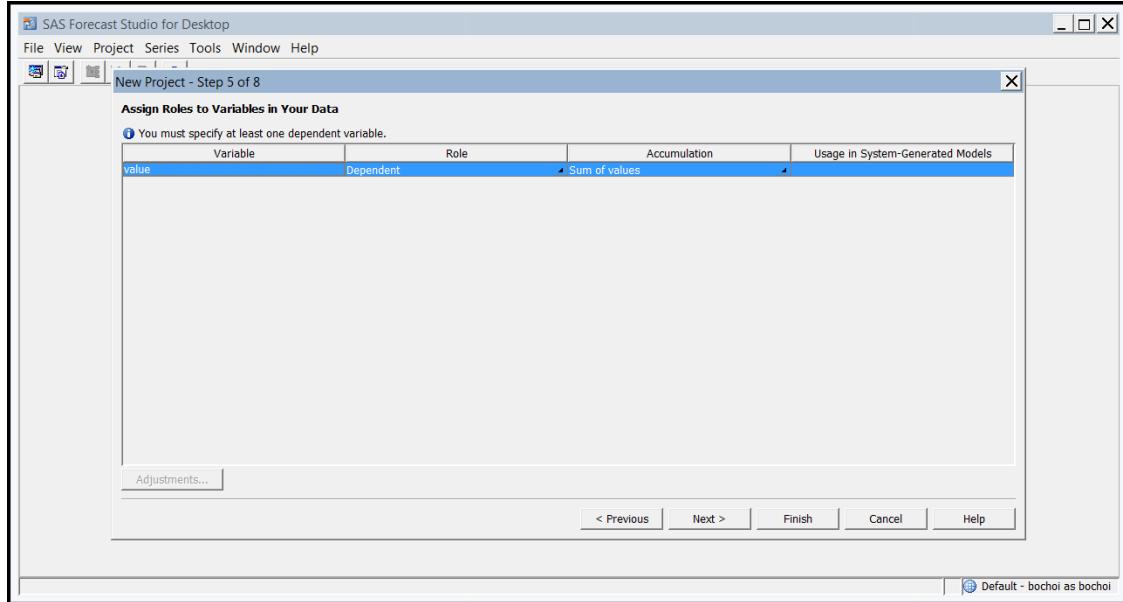
Figure 9.4: Select the Time Variable

In this example, the time variable is **date**, which is recorded at a monthly interval in the original series. The length of each analysis interval is the time unit specified in the **Interval** option multiplied by the number specified in the **Multiplier** option. With Interval = Month and Multiplier = 1, the analysis series is at a monthly interval. With Interval = Year and Multiplier = 2, the analysis series is at a biannual interval. The **Shift** option defines the subperiod, which is used to shift the analysis interval to a later starting point. The **Shift** option depends on the interval specification. For example, Year, Qtr, and Month intervals are shifted by calendar months. Week and Day intervals are shifted by days. Interval = Qtr and Shift = 3 reflects quarterly data shifted to the third month in each quarter (March, June, September, and December). Interval = Year and Shift = 7 means each analysis interval starts on July 1 of the year.

For the current example, set **Interval = Month**, **Multiplier = 1**, and **Shift = 1**. Because this specifies a monthly analysis interval, the natural seasonal length is 12. Click **Next**.

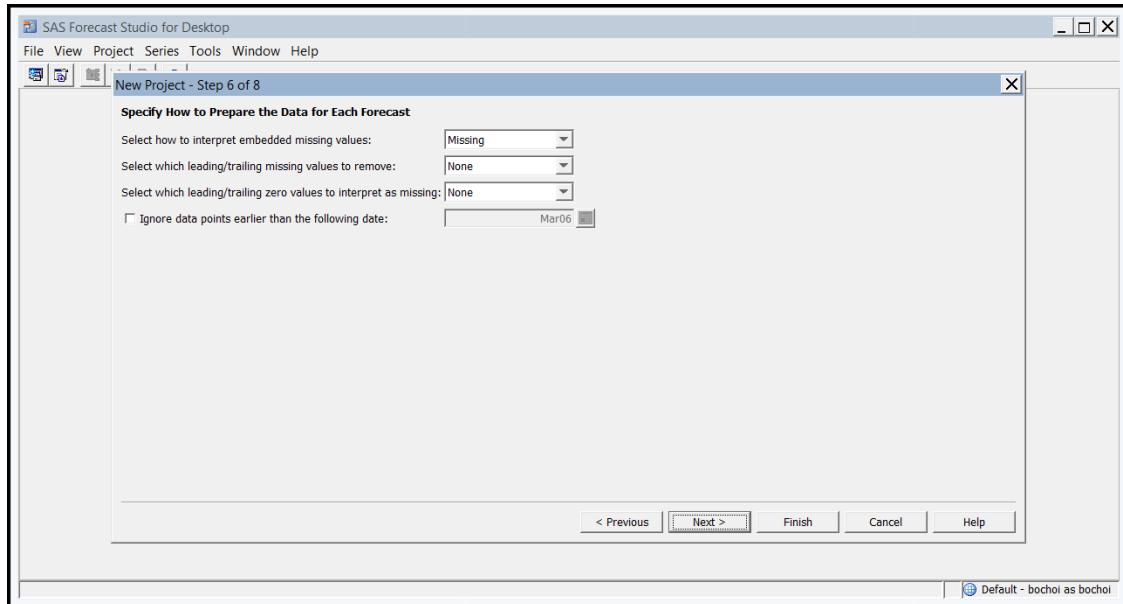
Figure 9.5 shows the fifth step that sets the roles of the variables. Depending on how you define analysis intervals, there can be multiple observations in one interval. To force the analysis series to have exactly one value for each interval, the selected summary statistic under **Accumulation** is used as the analysis value. Select **Dependent** for **Role** and **Sum of values** for **Accumulation**. Click **Next**.

Figure 9.5: Variable Roles and Accumulation Statistic

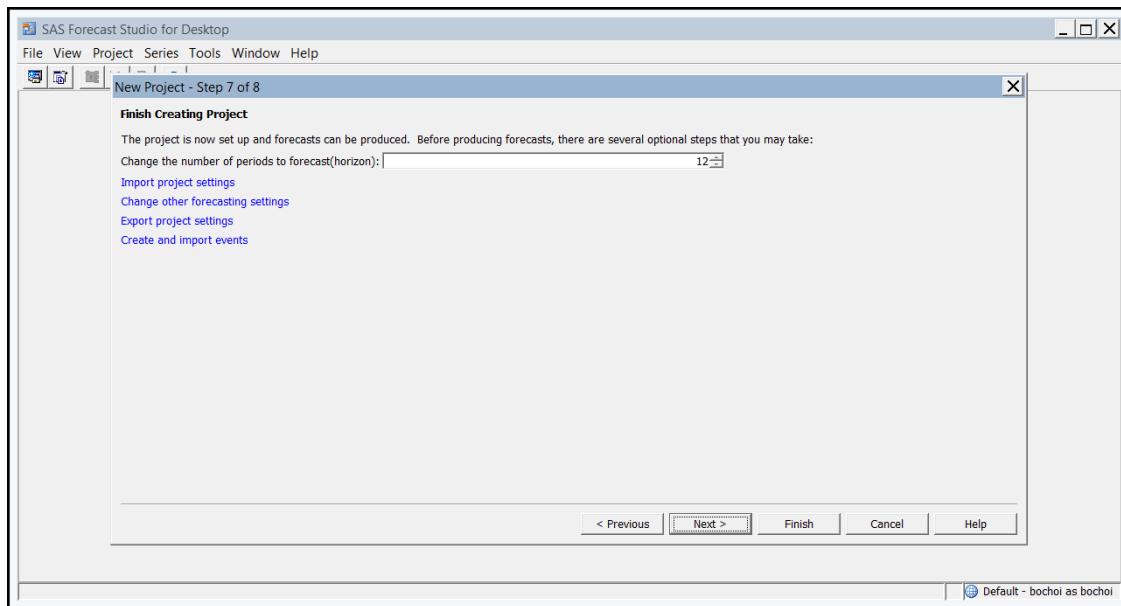


In step 6 (**Figure 9.6**), you can choose how to interpret and treat missing values. The example data set is preprocessed so that all series have the same range of dates with no missing values. Leave the default settings, and click **Next**.

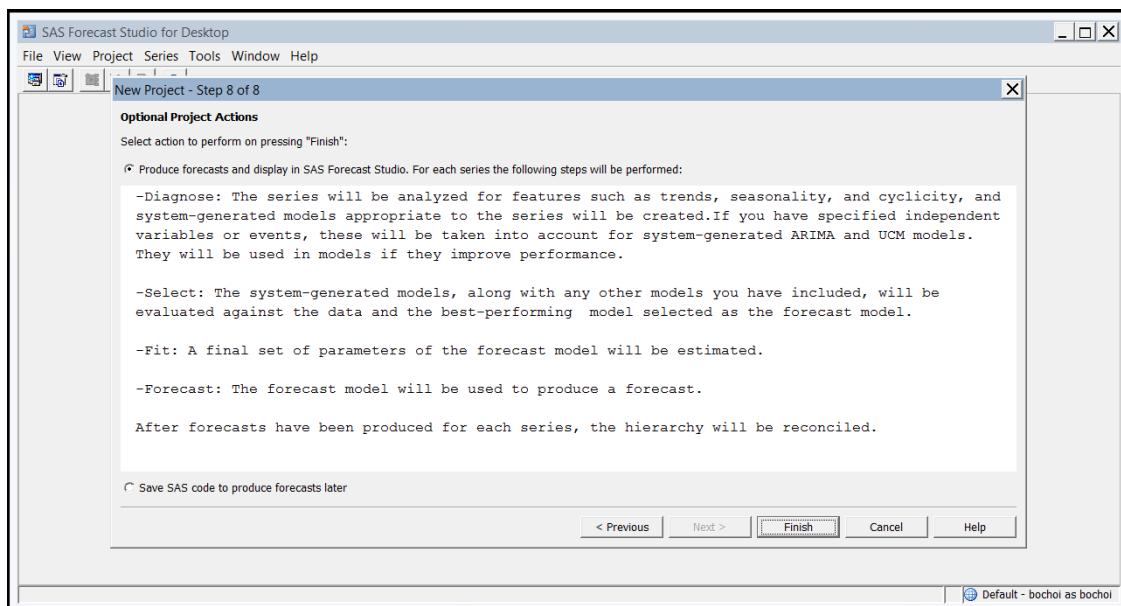
Figure 9.6: Missing Values



In step 7 (**Figure 9.7**), determine how many future points you would like to forecast. In this step, you can choose to import and export project settings, change other forecasting settings (such as model generation and selection, diagnostics, and many more), and create events for the project. These settings can be modified after the project is created, but that require refitting all series. More details are discussed throughout the chapter. Leave the default settings, and click **Next**.

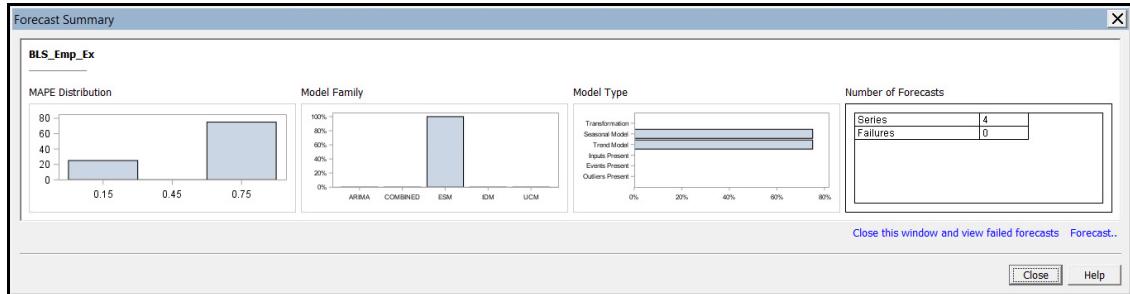
Figure 9.7: Forecast Horizon and Project Settings

In the final step (**Figure 9.8**), choose whether to produce outputs now or create SAS code to be run later. When the first option is chosen, SAS Forecast Studio goes through four steps: Diagnose, Select, Fit, and Forecast. In the Diagnose step, each series is examined, and based on diagnostics, SAS Forecast Studio builds candidate models. Trends, seasonality, cycles, inputs, events, and possible transformation of the dependent series might be considered. In the Select step, the candidate models and user-defined models (if any) are compared based on a goodness of fit measure, and the best model is chosen as the forecast model. Then, the forecast model is fit to the data in the Fit step, and corresponding forecasts are produced in the Forecast step. Select **Produce forecasts and display in SAS Forecast Studio**, and click **Finish**.

Figure 9.8: Run Options

Once the project is created and models are fit for each series specified by the BY variable, a forecast summary is produced as shown in **Figure 9.9**.

Figure 9.9: Forecast Summary

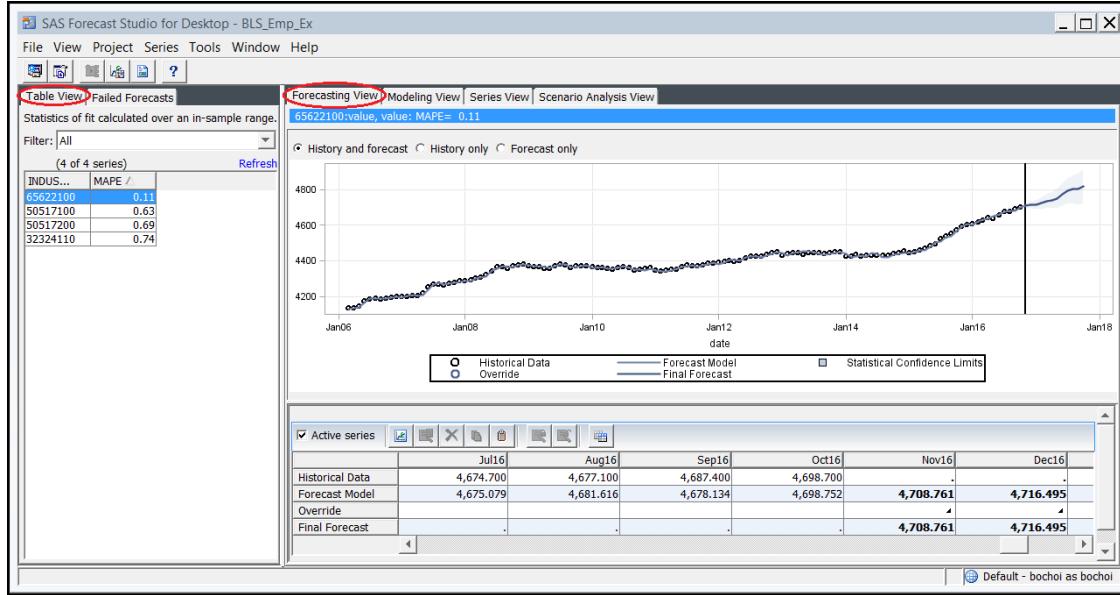


This example includes four series. One series has the mean absolute percentage error (MAPE) around 0.15, and the other three have MAPE around 0.75. With the default settings, an exponential smoothing model (ESM), including seasonal exponential smoothing, additive Holt-Winters multiplicative method, and damped-trend exponential smoothing, was selected for all series as the best-fitting model. Three of these models include seasonal terms and three include trend terms. Last, all four series were fit successfully (for example, no error messages). Close the Forecast Summary window.

9.3 Overview of Available Modes

The **Table View** tab in **Figure 9.10** shows the filtered list of series. Filters are used to control which series are shown. The current filter is **All**, listing all available series that were successfully fit. User-defined filters can be created using **Tools ▶ Filters**. Creating new filters provides a way to focus on a subset of the series to extract useful information when there are many series. Series that failed to be fit are listed on the **Failed Forecasts** tab.

Figure 9.10: Table View and Forecasting View



The first series in the left panel list is currently selected. It is the number of employees (in thousands) in general medical and surgical hospitals (65622100). The right panel shows the **Forecasting View** of the selected series. The forecast plot and table appear. An override can be specified to change some of the forecasts to known values for specific future time points. This is just a substitution and does not affect the model or other forecasts.

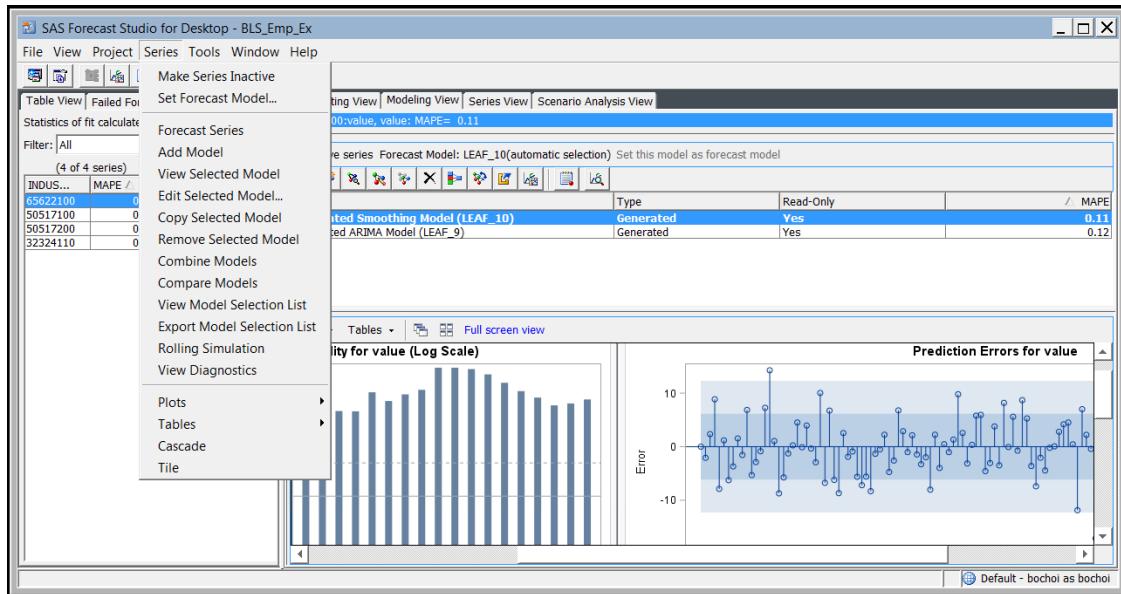
For example, if an analyst knew that there would be a strike of 30,000 employees in the next month (November 2016), the override value of $4709 - 30 = 4679$ can be supplied to modify the forecast at the time point. If you would like to view the numbers rounded to the third decimal place as in **Figure 9.10**, go to **View ▶ Edit Forecasting View**

Properties. Then, uncheck **Auto-formatting Enabled**, and supply the value e for **Set number of decimal places in forecasts**.

In the modeling view (**Figure 9.11**), model details of and diagnostics on the selected (forecast) model for the series and other competing models can be viewed. A number of plots and tables containing the information are available.

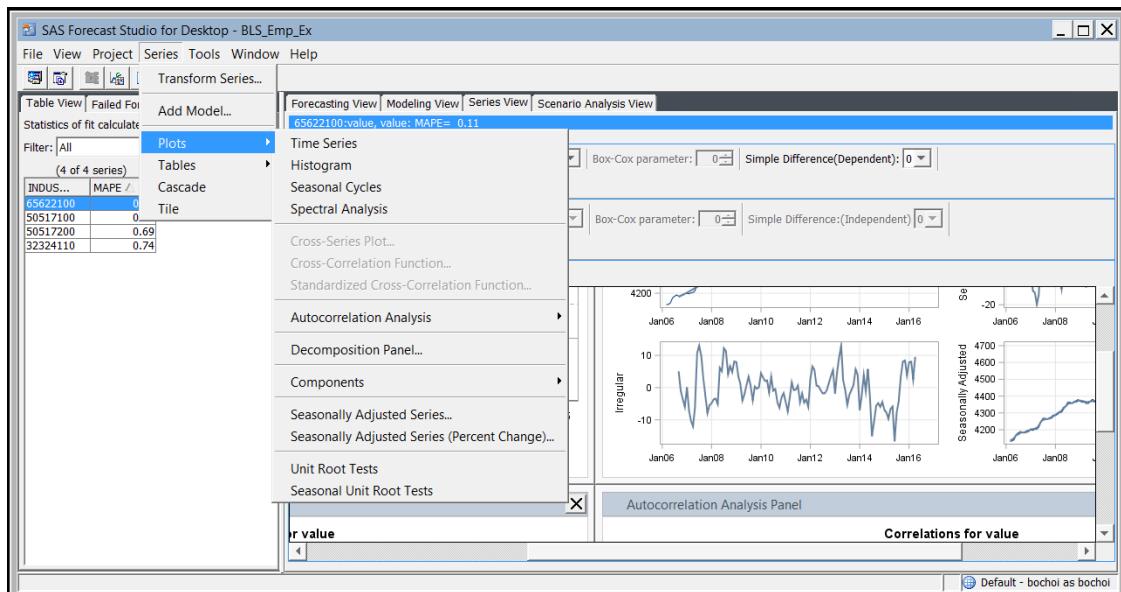
Moreover, you can create new models and manually choose a forecast model that overrides the automatically selected model. Currently, SAS Forecast Studio is selecting the model with the lowest MAPE. More details about the criterion are discussed in section 9.4.2. Tools for the **Modeling View** are listed on the **Series** tab as shown in **Figure 9.11**. Some of these tools are illustrated in more detail later.

Figure 9.11: Modeling View



The **Modeling View** focuses on model properties and diagnostics, and the **Series View** focuses on series properties such as transformation, decomposition, and unit roots (**Figure 9.12**). In this view, analysts can easily see by pointing and clicking how the series and diagnostics change if a log, square root, logistic, or Box-Cox transformation is used. Effects of simple differencing and seasonal differencing can be seen as well. In this way, you can efficiently experiment with different ways to transform the series in search of the optimal model for the series.

Figure 9.12: Series View



The **Scenario Analysis View** provides a convenient tool for what-if analysis—how the forecasts of dependent series change for different future values of an input series. Using this view, analysts can better understand the dynamics between the input and dependent series and test the robustness of the model. Moreover, this intuitive analysis scheme can serve as a communication bridge between mathematically minded analysts and executives with limited statistical background. This view is disabled for the current example because it does not have any input series.

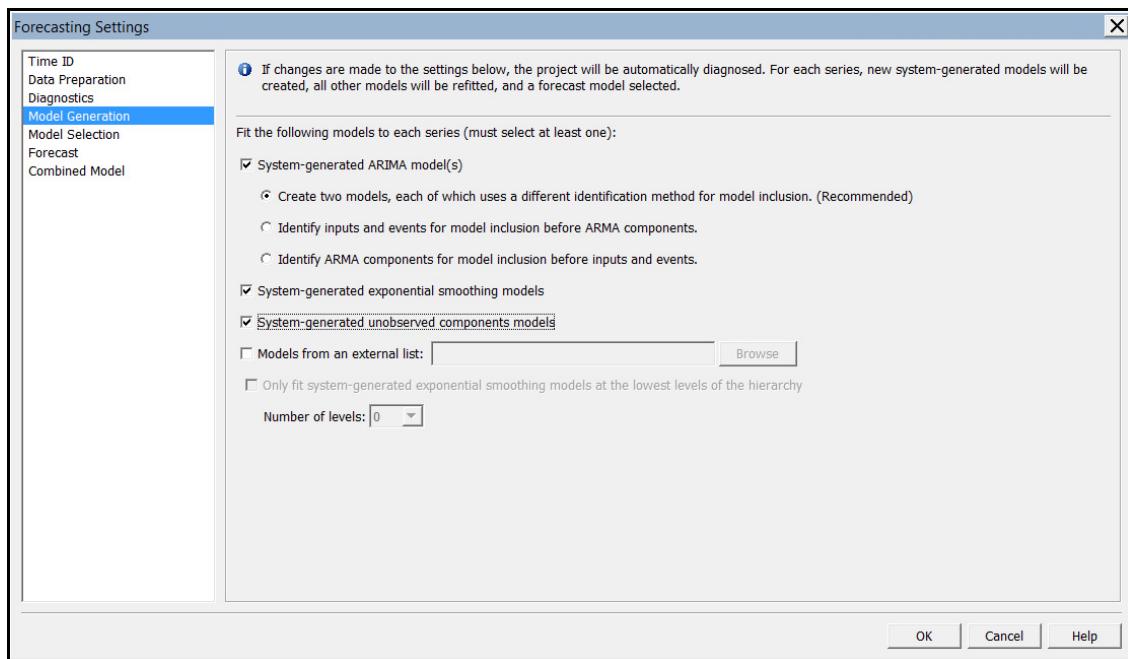
9.4 Project Settings

Even after the project is created, most of the initial settings can be modified. To change the project description and view the data source and project directory, select **File ▶ Project Properties**. In the project directory, the underlying SAS data sets generating the report can be found, providing a way for SAS programmers to extract additional information not present in the output. To change the data source, select **File ▶ Change Data Source**. To change the accumulation statistic, select **Project ▶ Hierarchy and Variable Settings**. This section is focused on forecasting settings that can be found in **Project ▶ Forecasting Settings**.

9.4.1 Model Generation

Figure 9.13 shows the settings for model generation. SAS Forecast Studio provides three broad classes of models for continuous series: ARIMA models (see **Chapters 1 through 5**), ESMs (see **Chapter 6**), and UCMs (see **Chapter 7**). Thus, the model selection in SAS Forecast Studio has more flexibility than the automatic model selection routine provided by the X13 procedure (see **Chapter 8**) that uses only ARIMA models. Often in transactional data, intermittent series are observed. That is, the series are sparsely observed or contain too many zeros, so they are difficult to be viewed as continuous. In that case, none of the three classes of models for continuous series can be used, and SAS Forecast Studio applies intermittent demand models. This last class of models is beyond the scope of this chapter. Interested readers can refer to *SAS Forecast Server Procedures: User's Guide* and references therein.

There are two ways of fitting ARIMA models. First, ARIMA models can be fit, and then based on the residuals, the effects of inputs and events can be estimated. Second, the effects of inputs and events can be selected and estimated first, and then using the adjusted series, ARIMA orders can be selected. Often, the effects of events and especially of inputs have a stronger influence on the response than does the ARMA structure of residuals. Estimates of these effects obtained by ignoring such autocorrelations are typically unbiased, though not fully efficient. These considerations suggest that, at least in theory, estimating effects of inputs first is preferable. This is the approach taken in PROC AUTOREG, where least squares residuals are used to diagnose the autocorrelation structure. Of course, an analyst could try both ways. For computation, ESMs tend to be simplest, then ARIMA models, and then UCMs. By default, SAS Forecast Studio fits ARIMA models and ESMs because UCMs are computationally intensive. They must be explicitly selected as shown in **Figure 9.13**. The decision of which class of models is used should depend on the analyst's time, computational resources, size of the data, and subject knowledge.

Figure 9.13: Model Generation Settings

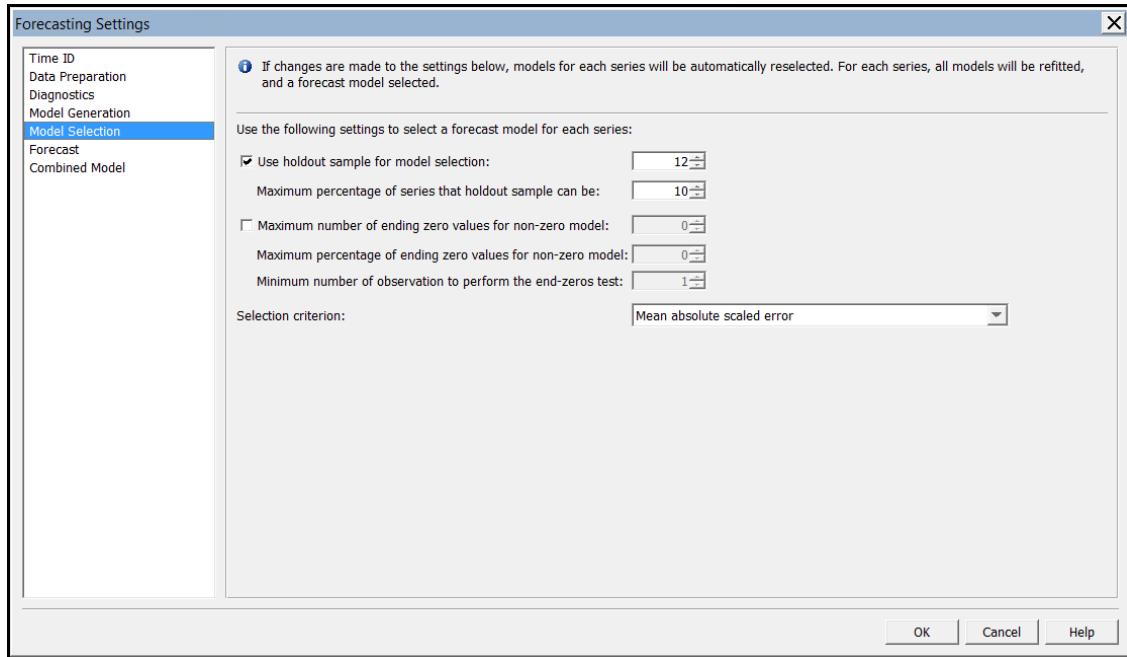
There are three types of models: default, custom, and external. Default models are provided by SAS Forecast Studio, custom models are created by users in SAS Forecast Studio, and external models are imported from an external model repository. Although users can freely edit custom models, default and external models cannot be modified. However, a user can always copy a model to create a new custom model. To view the current model repository for the project, select **Project ▶ Model Repository**.

9.4.2 Goodness of Fit and Honest Assessment

Goodness of fit statistics measure the overall quality of the model. SAS Forecast Studio supports a number of statistics. Among them are:

1. Mean squared error (MSE): $\frac{1}{T} \sum_{t=1}^T (Y_t - \hat{Y}_t)^2$
2. Mean absolute error (MAE): $\frac{1}{T} \sum_{t=1}^T |Y_t - \hat{Y}_t|$
3. Mean absolute percentage error (MAPE): $\frac{100}{T} \sum_{t=1}^T \frac{|Y_t - \hat{Y}_t|}{|Y_t|}$
4. Mean absolute scaled error (MASE): $\frac{T-1}{T} \frac{\sum_{t=1}^T |Y_t - \hat{Y}_t|}{\sum_{t=2}^T |Y_t - Y_{t-1}|}$

In these statistics, Y_t is the dependent series, \hat{Y}_t is the corresponding forecast series, and T is the number of time points. MSE is a natural goodness of fit in a theoretical sense because many statistical estimation methods try to minimize this measure. However, MSE is heavily influenced by large outliers. On the other hand, MAE is robust to outliers. Neither MSE nor MAE takes into account the scale of the series, so they cannot be used to compare series of different scale. That is, a question such as, "What industries have the highest MSE or MAE?" might not be very meaningful. MAPE gained its popularity because the statistic is simple and intuitive. However, it cannot be computed when any of the actual values in the series is zero. Moreover, given the same deviation, the contribution from a point is larger when the observed value is smaller. Studies show that models selected by MAPE tend to systematically underforecast. Mean absolute scaled error (MASE) is a relatively recent measure that overcomes the limitations. It is simply the MAE of the model divided by the MAE of a random walk. Thus, the statistic has an appealing interpretation: how well the model forecasts in comparison to the simplest model, an idea analogous to the familiar R^2 in the regression context. MASE is used as the goodness-of-fit statistic for this example. Nevertheless, the choice of statistics should depend on the nature of the problem and business needs.

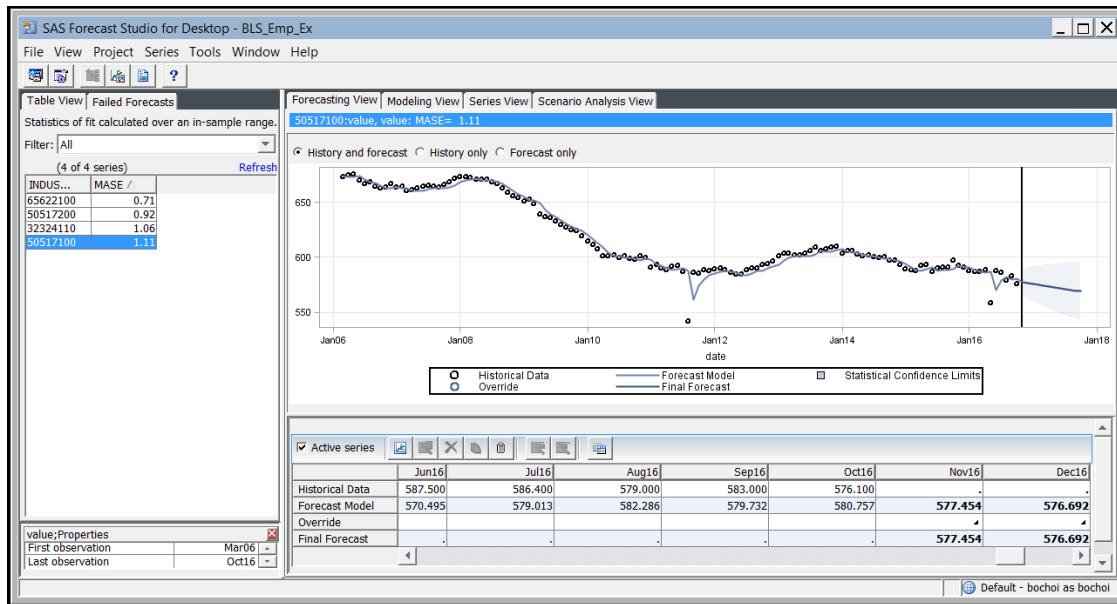
Figure 9.14: Model Selection Settings

All of the statistics previously mentioned (when calculated in the sample) tend to prefer more complex models, creating overfitting problems. Although statistics such as AIC and BIC take into account the complexity of the model, the comparison based on them across different transformations and differencing is often invalid. A general approach to overcome the difficulties is to use a holdout sample. That is, a portion of the data is held out, and the remaining data is used to fit the models. Then, using the estimated parameters, the goodness-of-fit statistic is calculated for the holdout sample to evaluate the model performance. This process is sometimes called *honest assessment*. Using a holdout sample, you are making an assumption that there is no significant change in the underlying process between the partitions. For time series, generally the last time points are held out, and the period for the holdout sample is called the *holdout-sample period*. The remaining portion is called the *in-sample period*. Because the current example is monthly data and some industries exhibit seasonality over a year, a reasonable choice is to hold out at least the last 12 time points. This amounts to around 9% of the observed time points. The options for model selection can be modified in **Project ▶ Forecasting Settings ▶ Model Selection** as shown in **Figure 9.14**. Once you make all of your changes, click **OK**.

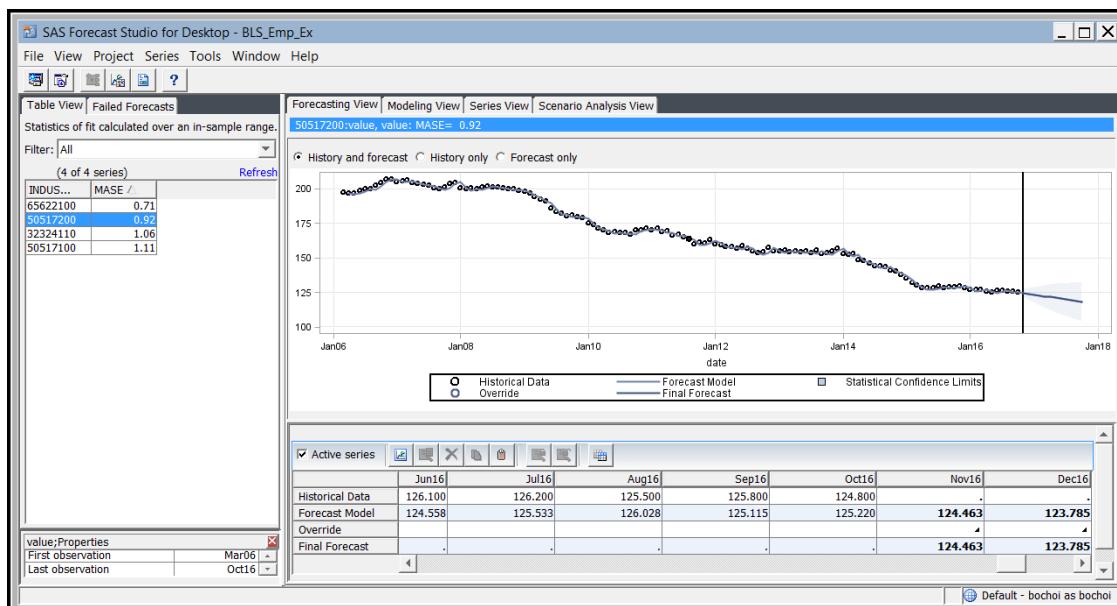
9.4.2 Transformation and Outlier Detection

For some series, transformation is required to enforce stable variability across time. SAS Forecast Studio offers log, square root, logistic, and Box-Cox transformation, as well as tests for seasonality. Moreover, large outliers can create a bias in estimation, so adjusting for outliers might be necessary. SAS Forecast Studio provides an option for automatic outlier detection for ARIMA models and UCMs. Outlier detection is not supported for ESMs. The automatic detection supports two types of outliers: level shift outliers and additive outliers. Level shift outliers shift the level (mean) of the series permanently. Additive outliers affect only a single time point—this point does not adversely affect the estimation. An example of level shift outliers is shown in **Chapter 4**, where the effect of a change in policy for directory assistance is investigated. (See **Output 4.21** through **4.25**.)

Figure 9.15 shows an example of additive outliers for the employment series in wired telecommunication carriers. There are two obvious outliers in August 2011 and May 2016. The exact date and value can be viewed in the plot by holding your pointer over the points of interest or by looking at the table in the bottom portion of the right panel. According to an article in *The New York Times*, there was a large strike by 45,000 Verizon employees in August 2011, which was the nation's largest strike since 2007. There was another strike in May 2016 by 40,000 Verizon employees. Not accounting for outliers, it takes several months for the forecasts to get back to a reasonable level. The reason is that at the time of the strike, the forecast is based only on past data, so the residual is a very large negative number. The ESM is a weighted average of the most recent data point and the associated residual. It is also too low. This behavior persists in a diminishing way for several steps past the intervention. This suggests a possible model improvement.

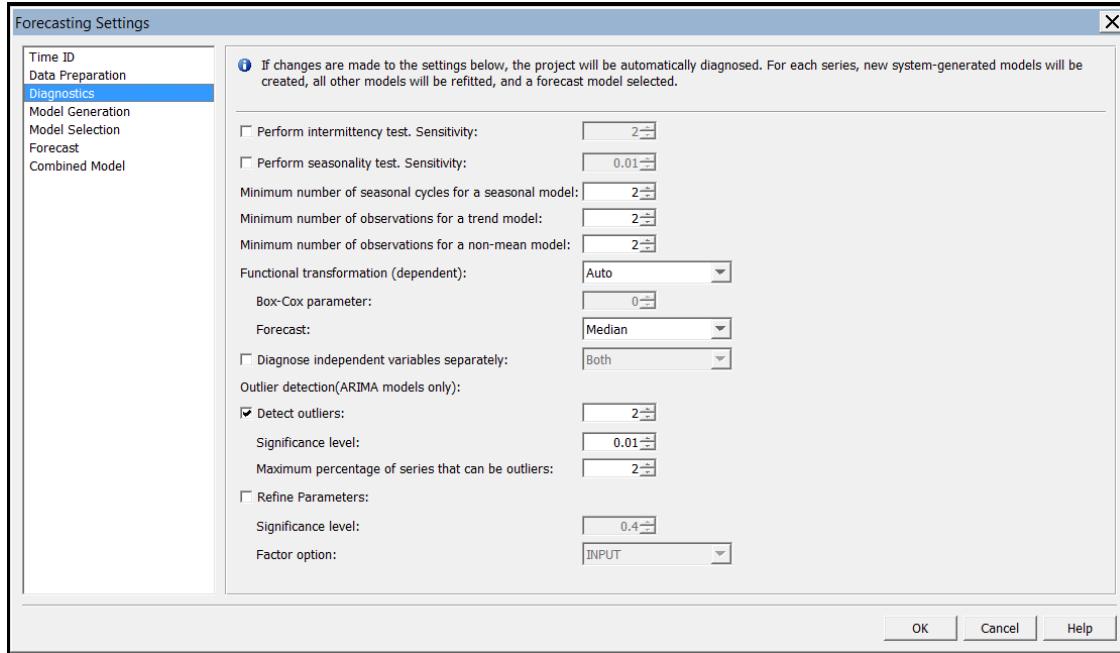
Figure 9.15: Forecasting View of Number of Employees (in Thousands) in Wired Telecommunication Carriers

Another interesting feature is shown in **Figure 9.16** for wireless telecommunication carriers. The strikes seem to have affected only the wired telecommunication carriers industry.

Figure 9.16: Forecasting View of Number of Employees (in Thousands) in Wireless Telecommunication Carriers

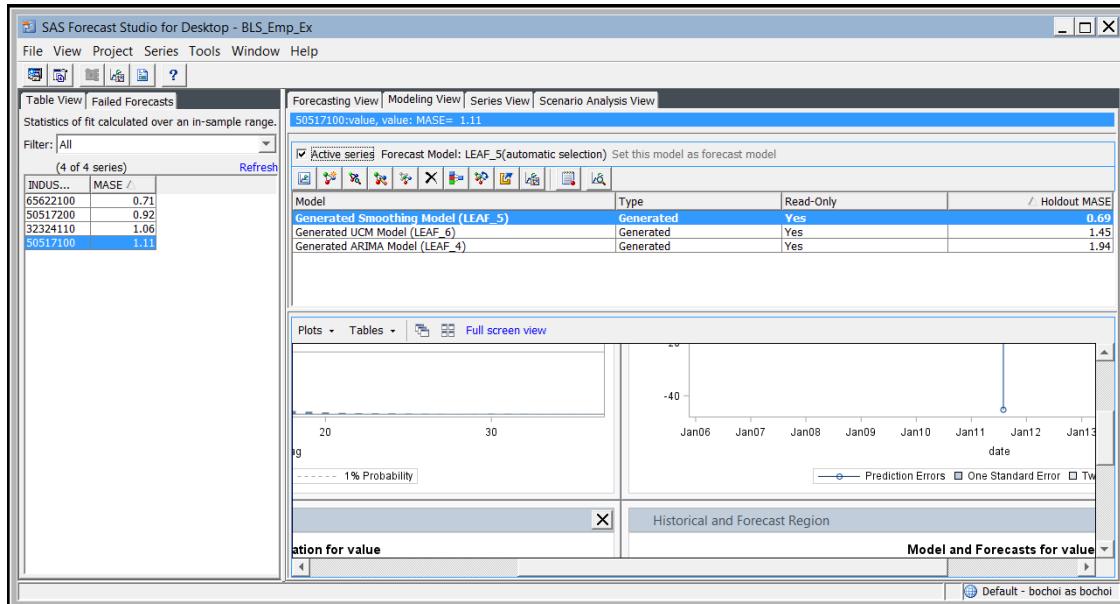
Select Project ▶ Forecasting Settings ▶ Diagnostics to select SAS Forecast Studio options for data transformation and outlier detection as shown in **Figure 9.17**. Click OK.

Figure 9.17: Diagnostics Settings



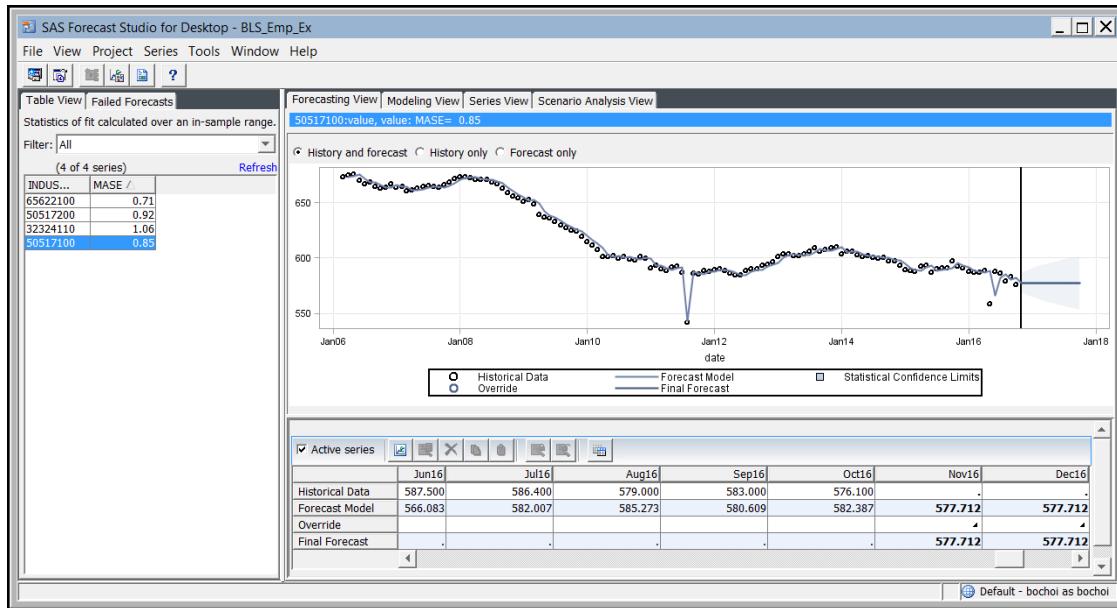
After models are refitted accounting for the August 2011 outlier, you will not see any change on the **Forecasting View** tab. The ESM is still the best in terms of holdout MASE, as shown in **Figure 9.18** on the **Modeling View** tab. This class of models does not account for outliers. Click **Generated ARIMA Model (LEAF_4)** on the **Modeling View** tab. Then, select **Series ▶ Copy Selected Model**, and click OK in the pop-up window. A copy of the system-generated ARIMA model called LEAF_4COPY1 is created. Select **Series ▶ Set Forecast Model**, and choose LEAF_4COPY1 for the forecast model from the drop-down menu.

Figure 9.18: Modeling View of Number of Employees (in Thousands) in Wired Telecommunication Carriers



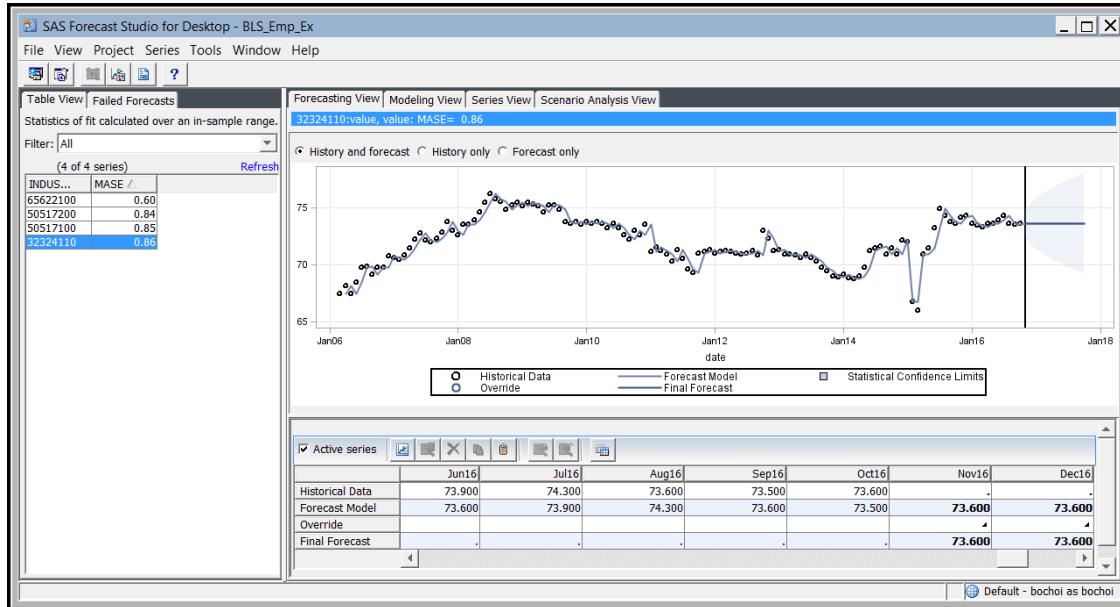
Go back to the **Forecasting View** as shown in **Figure 9.19**. If a copy was not made and the system-generated model was set to forecast model, when project settings are changed later to have all series diagnosed again, the change will be lost. Notice how the incorporation of an additive outlier for August 2011 has eliminated the carryover effect on subsequent forecasts.

Figure 9.19: Revised Forecasting View of Number of Employees (in Thousands) in Wired Telecommunication Carriers



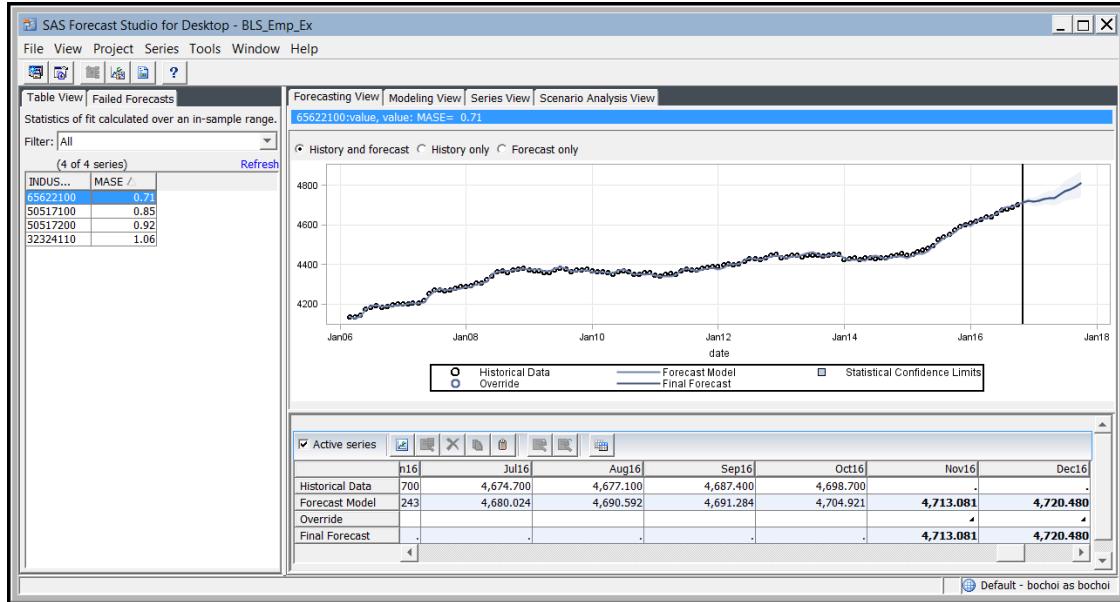
From the plot, the additive outlier is well captured for August 2011. However, the additive outlier is not detected for May 2016. This can be confirmed by selecting **Series ▶ View Forecast Model Details**. At the end of the third line, you will see AO01AUG2011D, meaning that AO on August 1, 2011 is incorporated in the model. Nothing for May 2016 can be found in the model details. This is because May 2016 is in the holdout period. Only the data portion before the holdout period is used to fit the model, so the additive outlier in May 2016 is not detected. If you go back to **Model Selection**, uncheck **Use holdout sample for model selection**, and click **OK**, then you will see that the new system-generated ARIMA model (that has captured both additive outliers) has a much lower in-sample MASE (0.65) than ESM (1.11). This example provides a caution against using a holdout sample for model selection: analysts should check that the holdout sample does not contain any abnormalities or display any change in the underlying process.

Another example in line with this caution is petroleum refineries shown in **Figure 9.20**. Starting in February 1, 2015, the oil refinery workers in the United Steelworkers union launched a long strike. Apparently, this struggle has led to a shift of level in the series. SAS Forecast Studio detected level shifts in February 2015 due to the prolonged strike, and in April 2015 that is perhaps due to the terms reached during the strike. The level shifted downward, then shifted back up above the previous level. If a holdout sample is used, a limited sample for estimating the latter level shift effect would be available, resulting in compromised holdout-sample performance. Select **Project ▶ Forecasting Settings ▶ Model Selection**, and recheck **Use holdout sample for model selection** for the analysis in the next section.

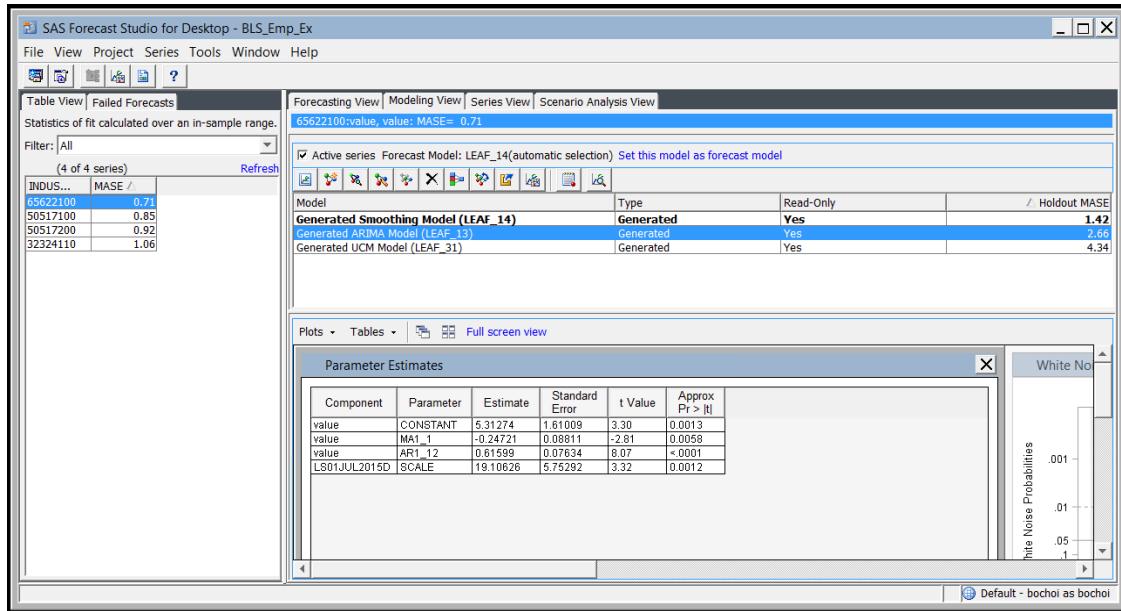
Figure 9.20: Forecasting View of Number of Employees (in Thousands) in Petroleum Refineries

9.5 Creating Custom Events

Figure 9.21 and Figure 9.22 show the Forecasting View and Modeling View, respectively, of the number of employees in general medical and surgical hospitals. Apparently, the growth in number of employees has been relatively slow between July 2008 and April 2015. The selected model is ESM with Holt-Winters multiplicative method.

Figure 9.21: Forecasting View of Number of Employees (in Thousands) in General Medical and Surgical Hospitals

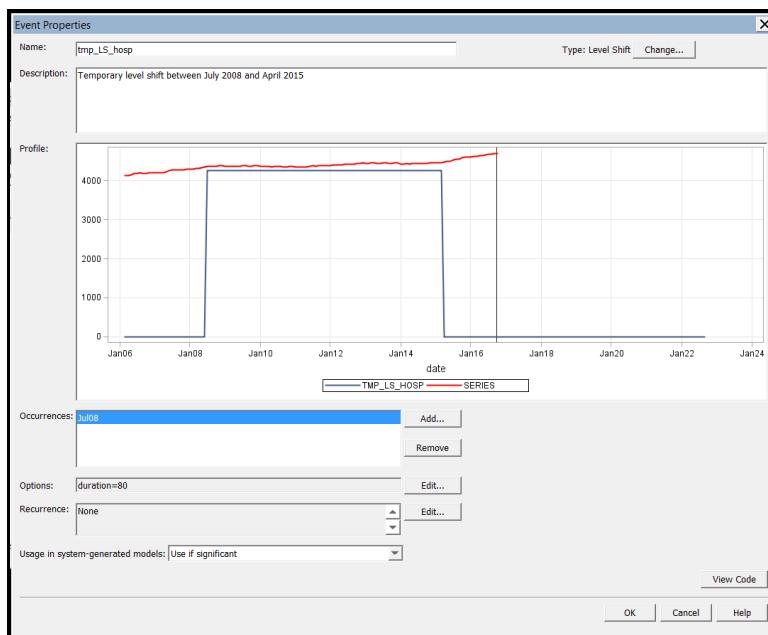
The runner-up is ARIMA(0,1,1)(1,0,0)12 plus a level shift on July 1, 2015. The parameter estimates table in the bottom panel of **Modeling View** can be requested using the drop-down menu next to **Tables**, and selecting **Parameter Estimates**.

Figure 9.22: Modeling View of Number of Employees (in Thousands) in General Medical and Surgical Hospitals

Instead of placing a level shift on July 2015, suppose that you want to model the series so that there is a temporary level shift between June 2008 and February 2015 in the slope of the series. Because you cannot change default models, first make a copy of the ARIMA model by selecting the model, and selecting **Series ▶ Copy Selected Model**.

In the pop-up window, click **Outlier Variables**. Click the X mark to the right of 01JUL2015D to remove this level shift outlier indicator from the model. Select **Events**, and click **New** to open the **Event Properties** window. On the first line, change **Name** to **tmp_LS_hosp** and **Type** from **Pulse** to **Level Shift**. Enter a **Description**, and click **Add** next to **Occurrences**. Choose **July** for **Month** and **2008** for **Year**, and click the right arrow. **Jul08** is added to **Occurrences**. Click **OK**.

Click **Edit** next to **Options**. Uncheck **All**, enter **80** for **Periods**, and click **OK**. This limits the duration of level shift to 80 months. At this point, the **Event Properties** window should look like **Figure 9.23**. Click **OK** to close the window.

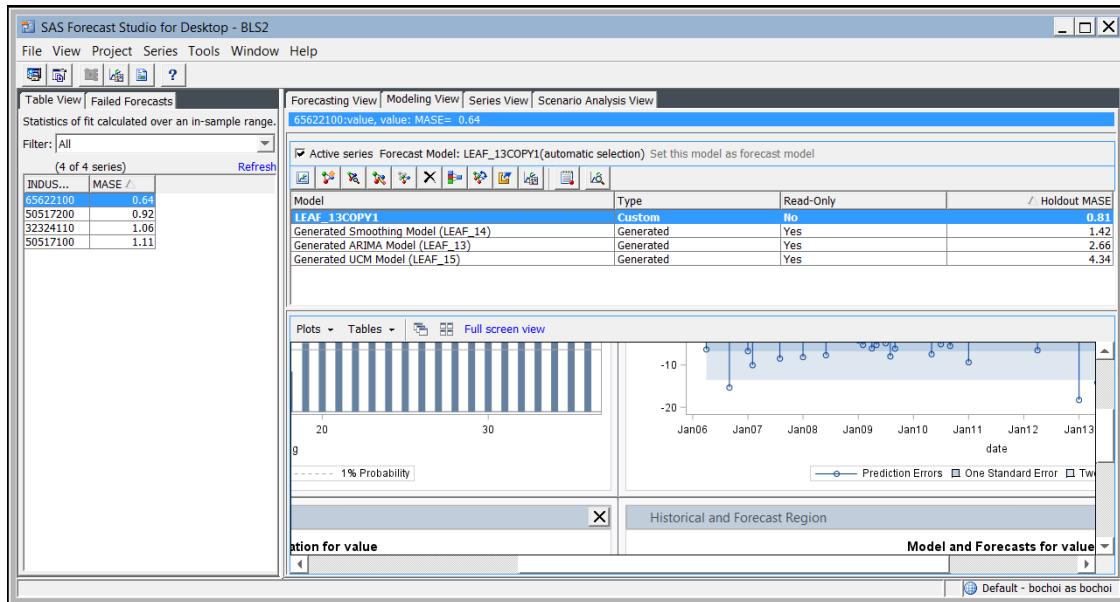
Figure 9.23: Creating a Fixed Duration Level Shift Outlier Indicator

Because you need to place a shift on the slope, which is equivalent to a level shift in the differences, and the model already has differencing order 1, the transfer function should be an identity. That is, the first difference $a + bt - (a+b(t-$

1)) of any line is just b , the slope. A change in slope is a level shift in the differenced series. Click the highlighted words below **Transfer Function**, and delete (1) to leave **Differencing order** of the level shift variable blank in the pop-up window. This shows how to enforce levels of differencing that are not the same in the response as they are in the input. Click **OK** to close the window and click **OK** again to complete the change. The model gives the same slope at the beginning and at the end of the data, with a different slope in the middle.

The holdout MASE for this custom model has decreased from 2.66 to 0.81, making it the best model as shown in **Figure 9.24**. The newly created event is automatically added to the event repository, which can be checked by selecting **Project ▶ Event Repository**. This event can now be reused for other models.

Figure 9.24: New Custom Model with the Temporary Level Shift



9.6 Hierarchical Time Series and Reconciliation

Often, time series can be broken down into several series, which can be further broken down into even smaller series. This creates a hierarchical structure. For example, consider the US housing inventory series downloaded from the US Census Bureau (Table 10a, “Quarterly Estimates of the Total Housing Inventory for the United States by Region”). The data contain the quarterly series of housing units from the second quarter of 2000 to the fourth quarter of 2016 by four broad regions in the United States (Northeast, Midwest, South, and West) and four status categories (Year-round vacant, Seasonal, Owner, and Renter).

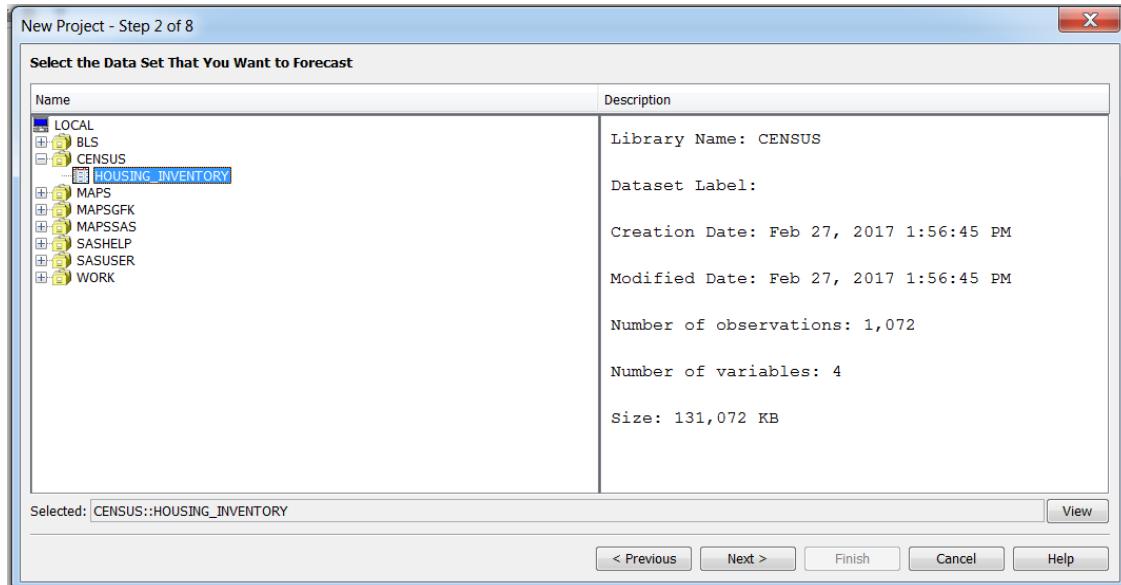
The US housing inventory series is an aggregate series that combines smaller series from different regions and status. An individual series might follow a very different process. Suppose that you take the hierarchy that divides the US series by the four regions, and you further subdivide each regional series into the four status categories. First, you can model the series at each level of hierarchy. But, the problem is that forecasts at the top level of hierarchy would not equal the sum of the forecasts at the middle level or at the bottom level. By reconciling the differences, you can use the information contained in the hierarchical structure that might improve the forecasts. This process is called *reconciliation*.

SAS Forecast Studio offers three options to choose the base reconciliation level: top down, bottom up, and middle out. Forecasts at the base level are kept unchanged. Forecasts at other levels are reconciled according to the base-level forecasts. The top-down method chooses the highest level as the base level, the bottom-up method chooses the lowest level, and the middle-out method chooses the middle level. When there are more than three levels, users can choose the specific middle level as the base level. It might not necessarily be true for all series, but in many applications, series are the smoothest at the top level and contain more noise at the bottom. Thus, the top-down approach does a better job in removing the noise at the lower levels. The bottom-up approach does a better job in identifying certain patterns such as seasonality.

Create a project for the US housing inventory data. Select **File ▶ New Project**. In step 1, enter a project name and description. In step 2, choose the **HOUSING_INVENTORY** data set in the **CENSUS** library as shown in **Figure 9.25**.

In step 3, choose **region** and **status** as classification variables. Check **Forecast a hierarchy using the above classification (BY) variables** and check **Reconcile the hierarchy**.

Figure 9.25: Step 2 in the New Project Wizard for the US Housing Inventory Data



From the **Reconcile the hierarchy** drop-down menu, select **Middle Out—region** as shown in **Figure 9.26**. Click the **Preview** button if you want to see the hierarchy in a tree view.

Figure 9.26: Step 3 in the New Project Wizard for the US Housing Inventory Data

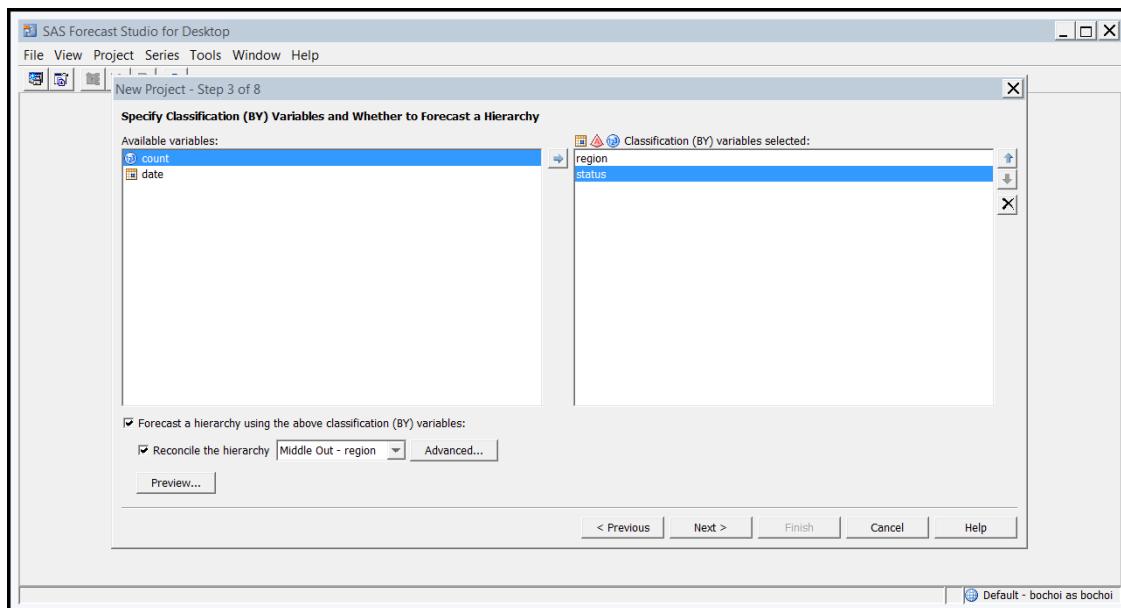
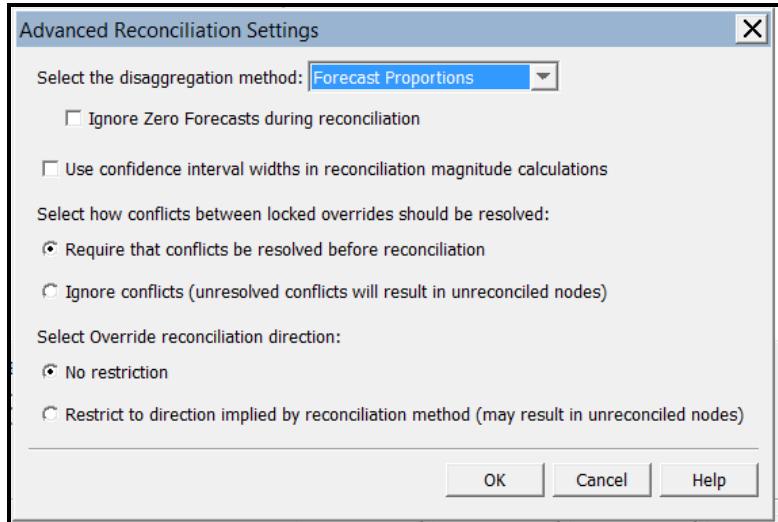


Figure 9.27 shows additional settings for reconciliation methods. To access these, click **Advanced** next to **Reconcile the hierarchy**.

Figure 9.27: Advanced Reconciliation Settings



Because you have selected **Middle Out – region** as the base forecast, the relationship between the top level and the region would be bottom up. With no other constraints specified for overrides, the top-level forecasts are simply the sum of the forecasts for each region. On the other hand, the relationship between the region and status series is top down. The problem is a little more complicated because there are infinitely many ways to distribute the middle-level forecasts into the bottom level. To resolve the identifiability problem with disaggregation, SAS Forecast Studio minimizes a loss function of user's choice: **Equal Split of the Difference** and **Forecast Proportions**. Let \hat{R}_t denote the forecasts for the middle level—region (which has also been chosen as the base level). Let $\hat{S}_{t,I}$ denote the forecasts for status i at the bottom level. Also, denote by I the number of distinct levels of status and $\hat{S}_{t,i}$ the reconciled forecasts.

The **Equal Split of the Difference** option computes $\hat{S}_{t,i}$ that minimize the squared loss function:

$$\sum_{i=1}^I (\hat{S}_{t,i} - \tilde{S}_{t,i})^2$$

This is under the constraint that $\hat{R}_t = \sum_{i=1}^I \tilde{S}_{t,i}$. Then, the solution must satisfy the following result:

$$\tilde{S}_{t,1} - \hat{S}_{t,1} = \tilde{S}_{t,2} - \hat{S}_{t,2} = \dots = \tilde{S}_{t,I} - \hat{S}_{t,I} = \frac{1}{I}(\hat{R}_t - \sum_{i=1}^I \hat{S}_{t,i})$$

This leads, in turn, to the following result:

$$\tilde{S}_{t,i} = \hat{S}_{t,1} + \frac{1}{I}(\hat{R}_t - \sum_{i=1}^I \hat{S}_{t,i})$$

The option name comes from the fact that the differences between reconciled and unreconciled forecasts are the same across the levels of status.

The **Forecast Proportions** option minimizes the loss function:

$$\sum_{i=1}^I \frac{(\hat{S}_{t,i} - \tilde{S}_{t,i})^2}{|\hat{S}_{t,i}|}$$

This is under the same constraint. Then, the solution must satisfy the following result:

$$\frac{\tilde{S}_{t,1} - \hat{S}_{t,1}}{|\hat{S}_{t,1}|} = \frac{\tilde{S}_{t,2} - \hat{S}_{t,2}}{|\hat{S}_{t,2}|} = \dots = \frac{\tilde{S}_{t,I} - \hat{S}_{t,I}}{|\hat{S}_{t,I}|} = \frac{\hat{R}_t - \sum_{i=1}^I \hat{S}_{t,i}}{\sum_{i=1}^I |\hat{S}_{t,i}|}$$

This leads to:

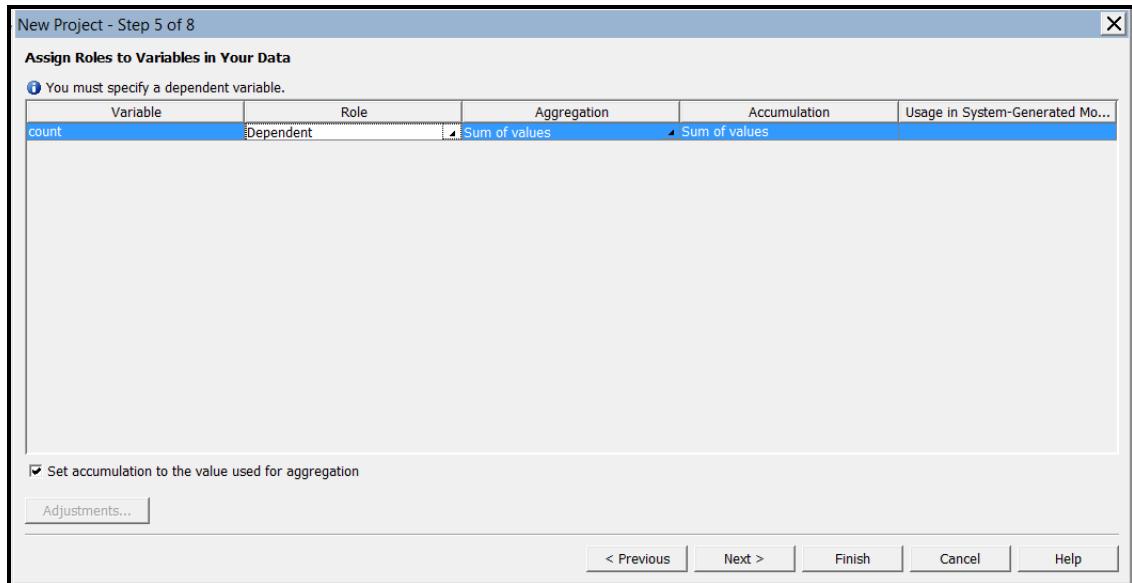
$$\tilde{S}_{t,i} = \hat{S}_{t,i} + \frac{|\hat{S}_{t,i}|}{\sum_{i=1}^I |\hat{S}_{t,i}|} (\hat{R}_t - \sum_{i=1}^I \hat{S}_{t,i})$$

The option keeps the forecast proportions the same across the levels of status, letting the bottom-level series of larger magnitude have larger deviations.

If **Use confidence interval widths in reconciliation magnitude calculations** is checked, then instead of the simple summation in the loss function, a weighted sum based on the confidence interval widths is used. Remaining settings deal with locked overrides (which were briefly mentioned in the beginning of section 9.3). Overrides might create additional constraints for the optimization problem and cause reconciliation failures. Requiring that conflicts be resolved before reconciliation and putting no restriction on the override reconciliation direction (see **Figure 9.27**), you can have reconciled forecasts for all series. Click **OK** to close the window. Click **Next**.

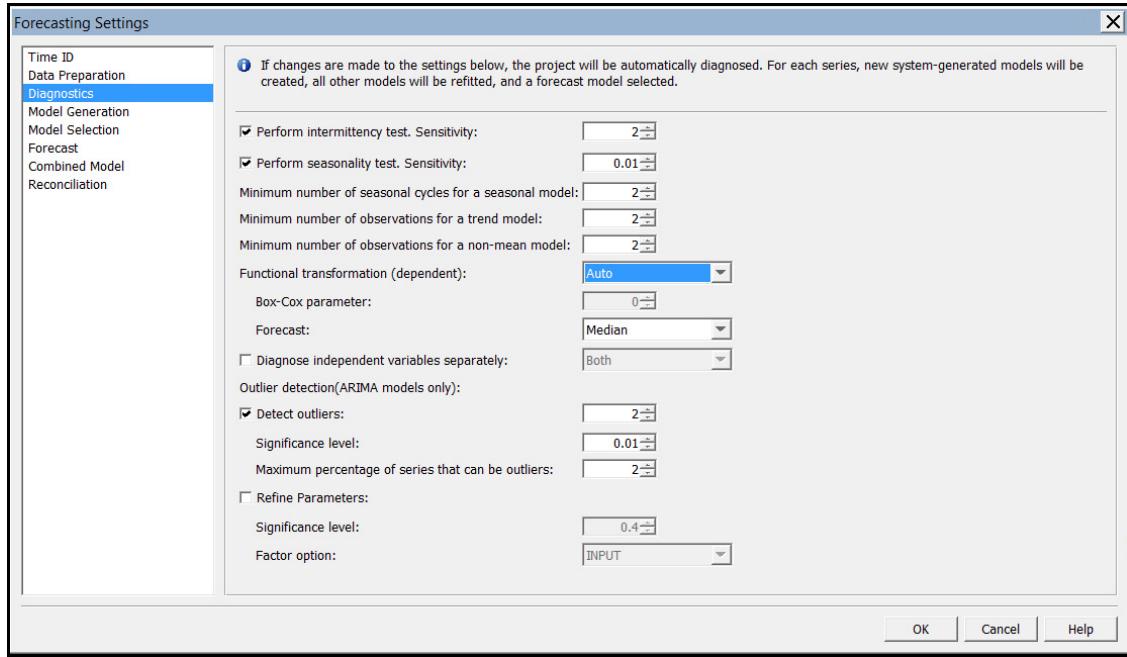
In step 4, choose **date** as the **Time ID variable**. Leave the default settings and click next. In step 5, choose the role **Dependent** for the variable **count** and set the **Aggregation** and **Accumulation** statistics as **Sum of values** (see **Figure 9.28**). Unlike the corresponding step with no hierarchy as shown in **Figure 9.5**, you have a new option for the aggregation statistic, which dictates how the upper-level series is aggregated from the lower-level series. The equations and solutions derived in this section assume that **Sum of values** is selected for the aggregation statistic. When **Average** is selected for aggregation, a few modifications are needed. Interested readers can refer to *SAS Forecast Server Procedures: User's Guide*.

Figure 9.28: Step 5 in the New Project Wizard for the US Housing Inventory Data



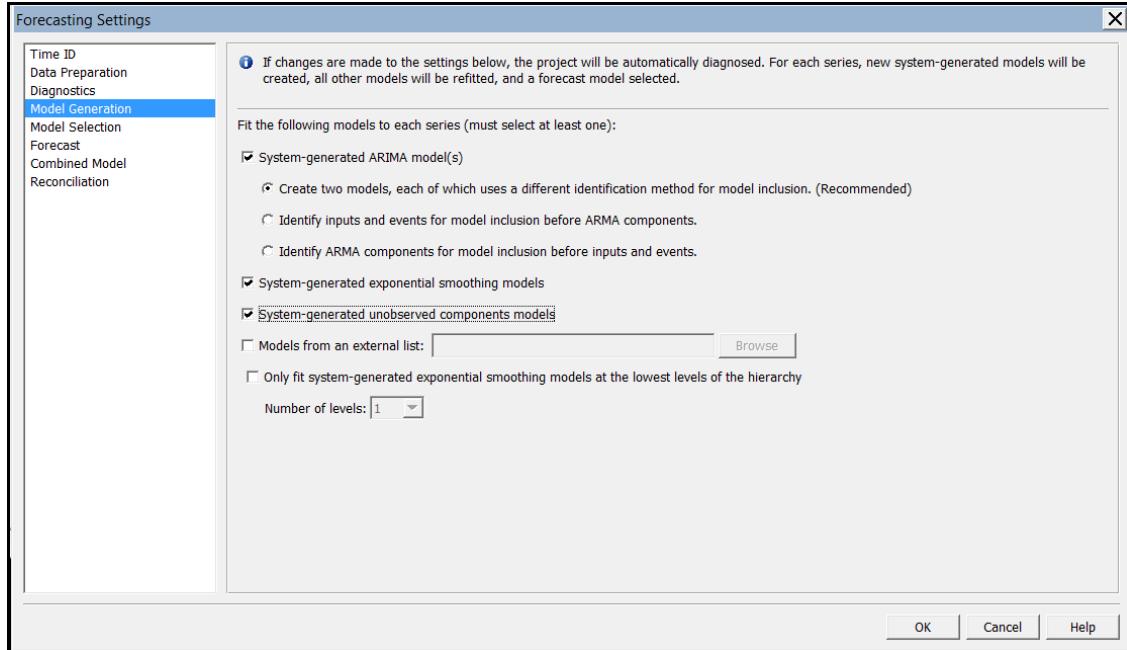
In step 6, leave the default settings, and click **Next**. In step 7, click on **Change other forecasting settings**. Select **Diagnostics**, and change the settings as shown in **Figure 9.29**.

Figure 9.29: Diagnostics Settings



Select **Model Generation**, and choose ARIMA, ESM, and UCM, as shown in **Figure 9.30**.

Figure 9.30: Model Generation Settings



Select **Model Selection**. The example data set has quarterly series, so choose the last four time points as the holdout sample. Select **Mean absolute scaled error** as the selection criterion as shown in **Figure 9.31**. Click **OK** to close **Forecast Settings** window. Click next to go to Step 8. Then, click **Finish** to complete the project creation.

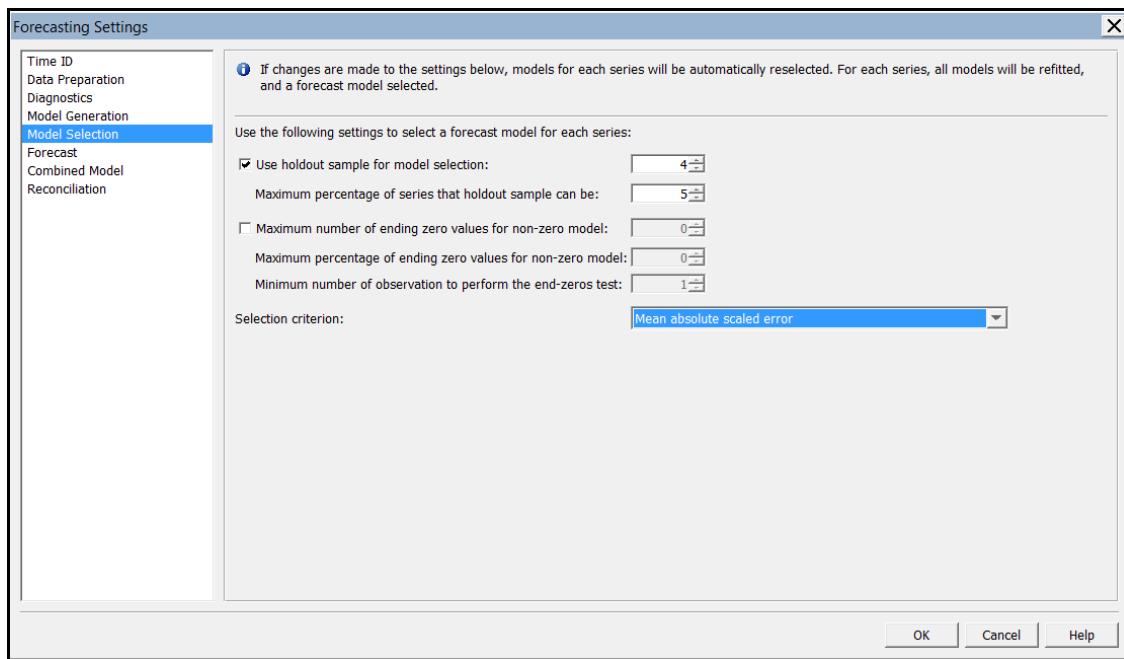
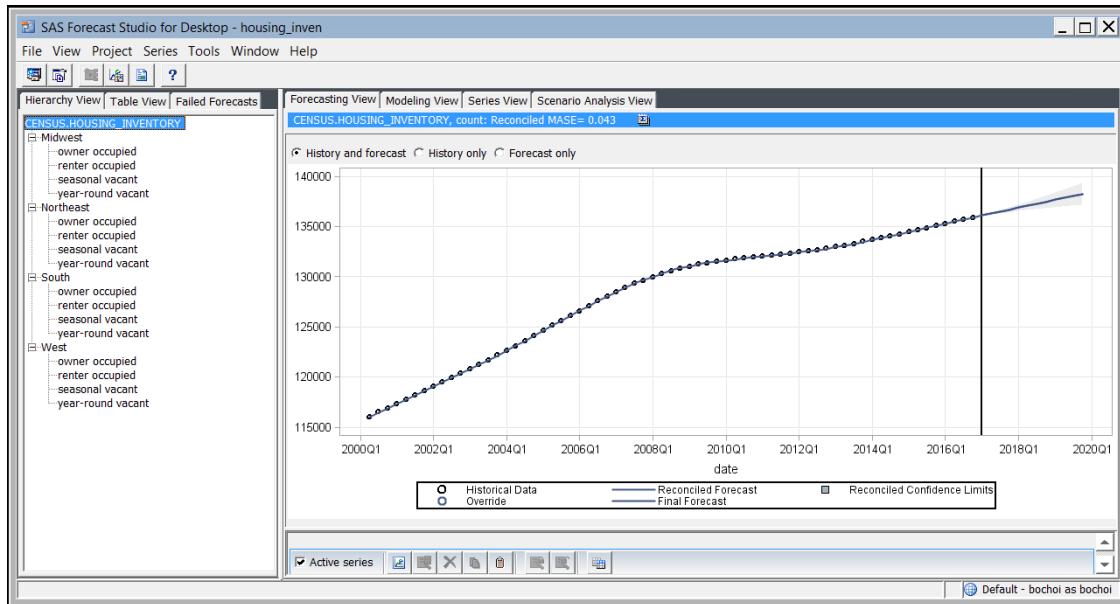
Figure 9.31: Model Selection Settings

Figure 9.32 shows the forecast summary. The summary is presented at each level of the hierarchy. At the middle level (region), ESM with the trend component is selected for all four series. At the bottom level (status), some series selected ARIMA and UCM, although ESM is still selected for the majority. Most series exhibit trend and/or seasonality, and outliers are present in one series. Click **Close**.

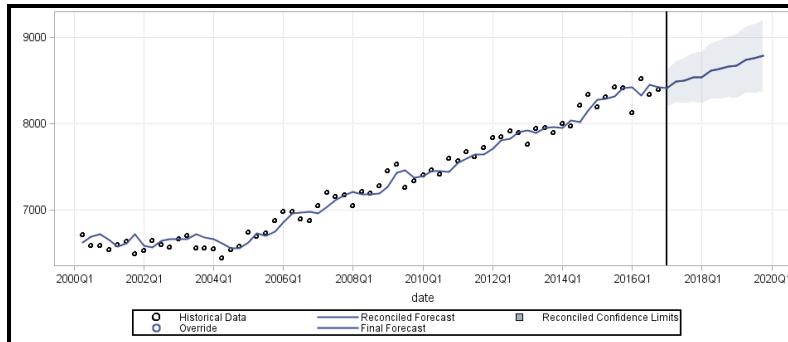
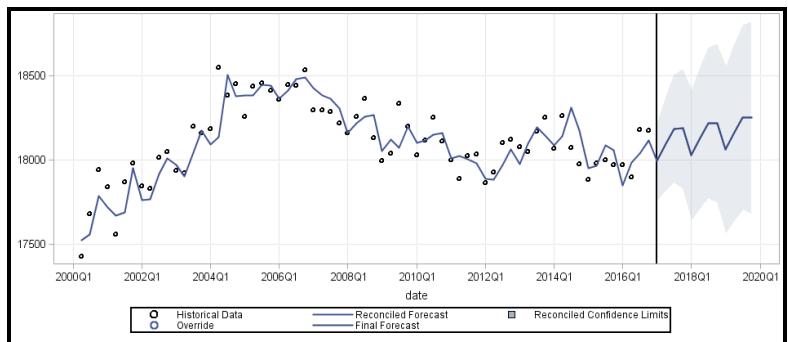
Figure 9.32 Forecast Summary

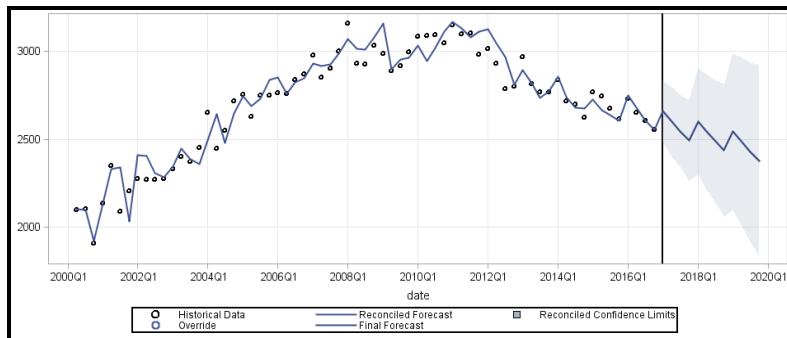
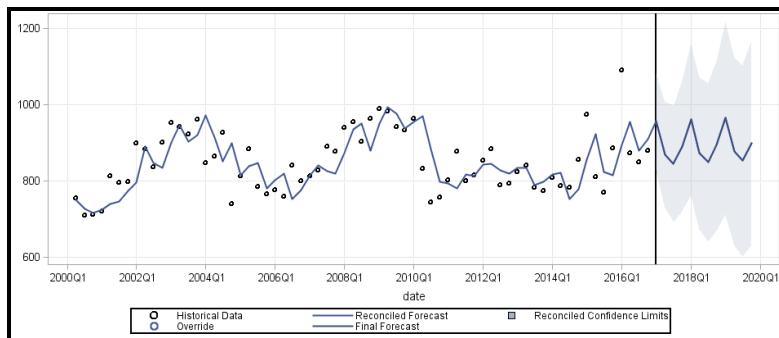
As shown in **Figure 9.33**, there is the new **Hierarchy View** that facilitates moving between different levels of hierarchy. In the series plot, the values are in thousands. The US housing inventory has been growing at an increasing rate since 2012, but not as fast as the precrisis level before 2008. The impact of the Great Recession has been lingering for a decade. A similar story can be told at the middle level across regions.

Figure 9.33: Forecast View of US Housing Inventory Data

Output 9.1 shows the housing inventory in the Midwest broken down into the four status categories: owner, renter, seasonal, and year-round vacant. You see a steady growth in renter-occupied units and a decline in year-round vacant housing. A recent jump in owner-occupied units is observed, but not enough observations are available to see whether it is suggesting a new level or temporary fluctuations.

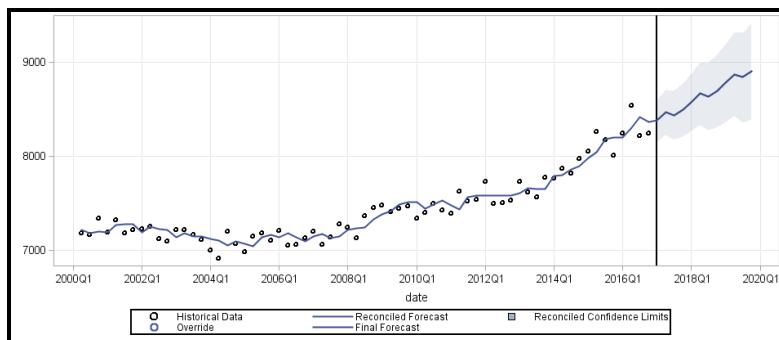
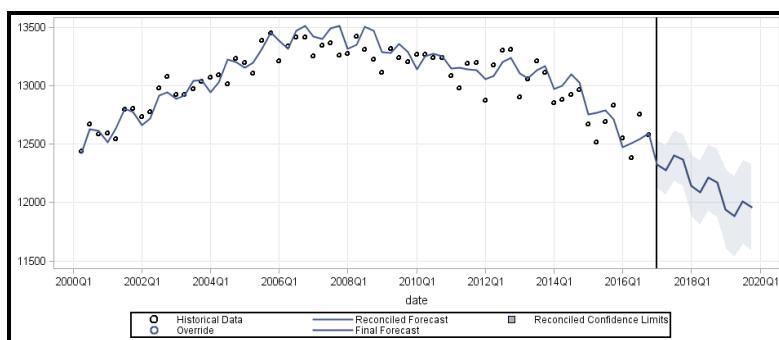
Output 9.1: Midwest Housing Inventory by Status (Owner, Renter, Seasonal, and Year-Round Vacant)

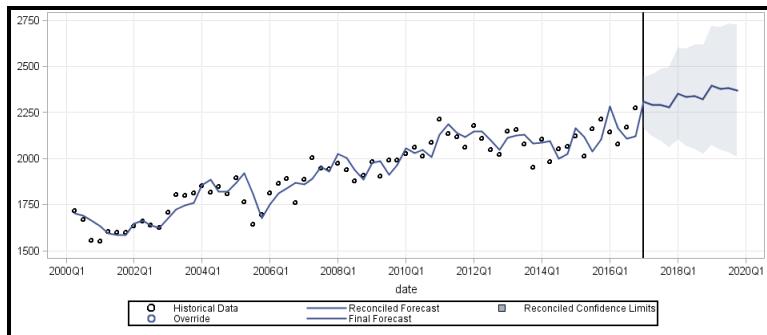
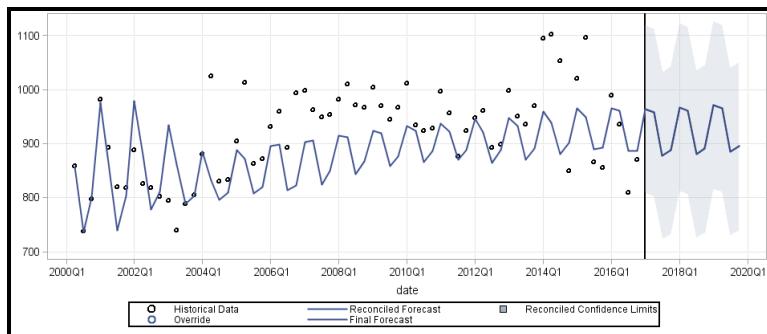




Output 9.2 shows the same set of plots for the Northeast. Owner-occupied units have been steadily declining since 2008, while renter-occupied units are increasing. The poor model fit on seasonal suggests that reconciliation does not necessarily produce better results. Unlike the Midwest region, the forecasts on year-round vacant units is growing.

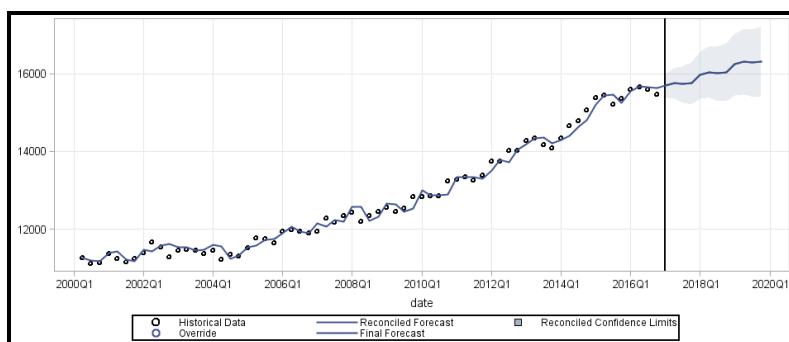
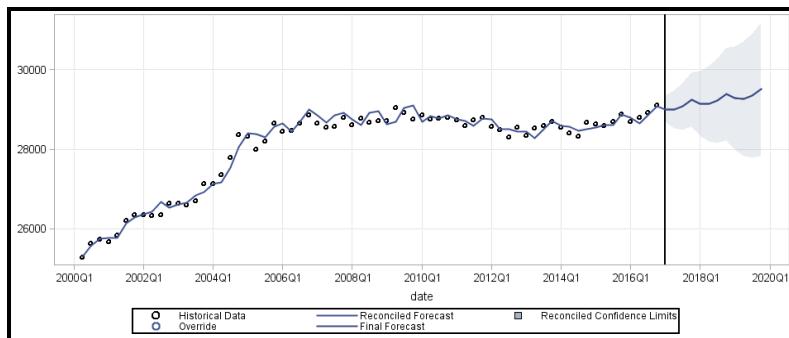
Output 9.2: Northeast Housing Inventory by Status (Owner, Renter, Seasonal, Year-Round Vacant)

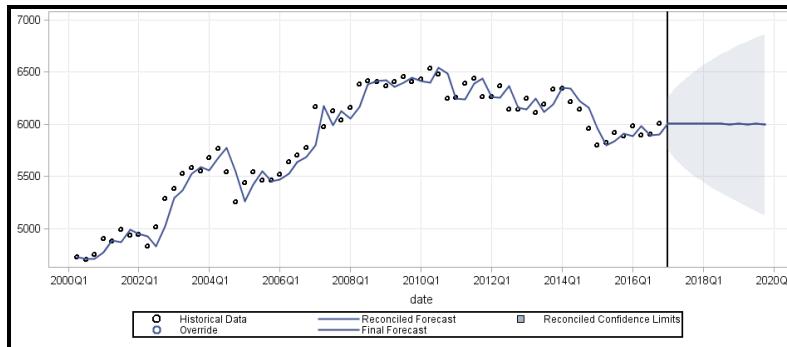
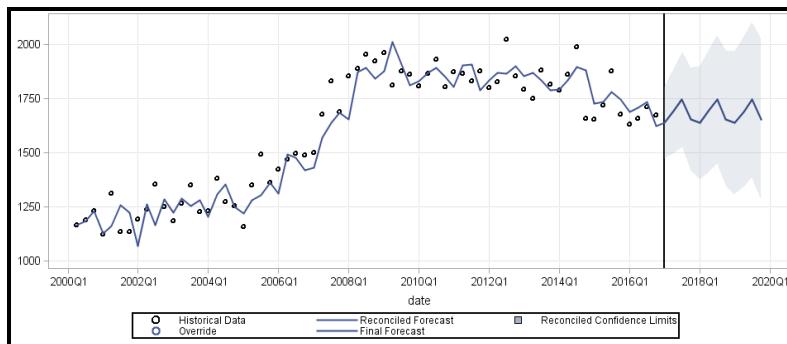




Output 9.3 shows that, in the South, owner-occupied units have been stable since 2006 with a slight recent increase. On the other hand, renter-occupied units have been increasing, but a recent decline in the growth rate is observed. The seasonal units have also been declining since 2008, and year-round vacant units had a rapid drop in 2014, but have been stable since then.

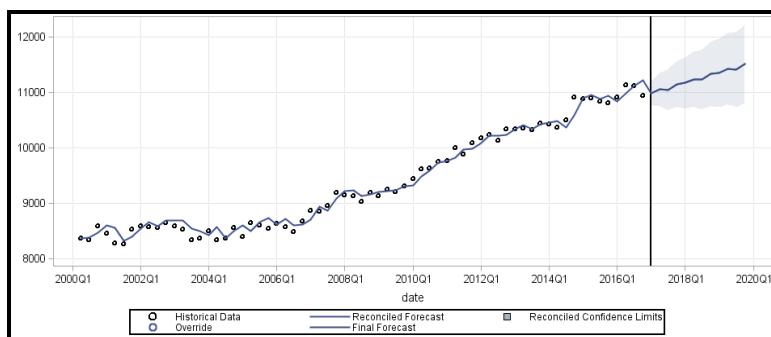
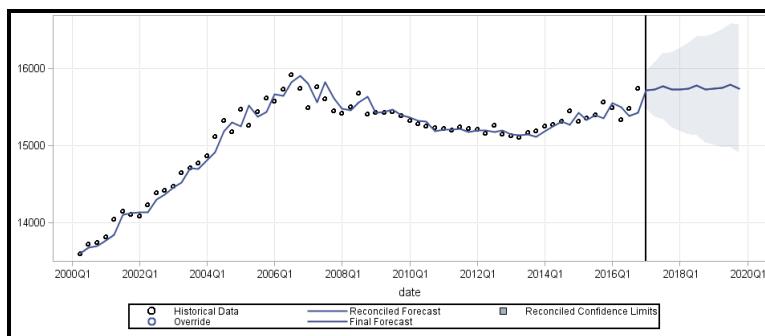
Output 9.3: South Housing Inventory by Status (Owner, Renter, Seasonal, and Year-Round Vacant)

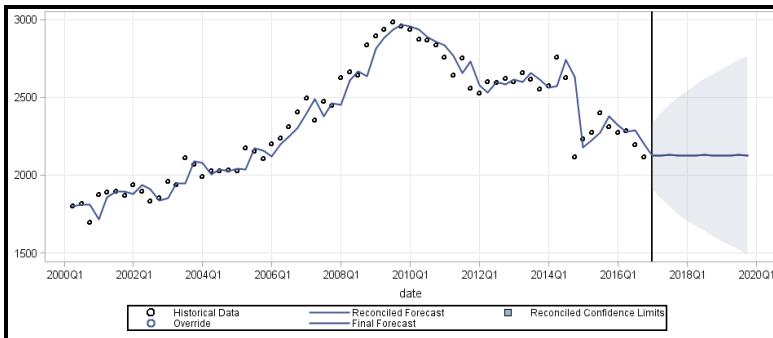
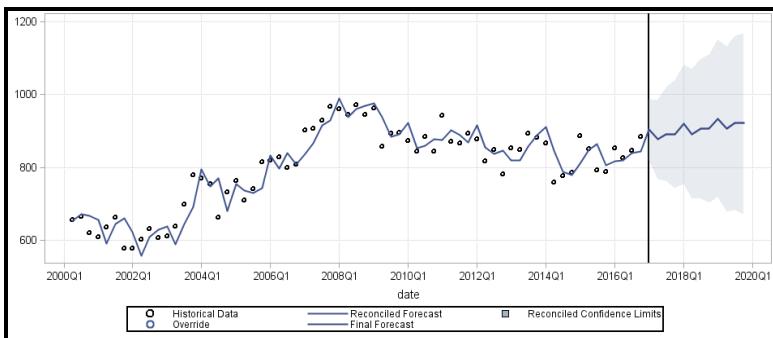




In the West (**Output 9.4**), owner-occupied units have been declining since 2007, but started to grow in 2014. Renter-occupied units have been increasing at a stable rate since 2007. Seasonal units have been declining since 2008, but with the recent increase, the forecasts are increasing. As is the case with the South, year-round vacant units have seen a rapid drop in 2014, but have been stable since then.

Output 9.4: West Housing Inventory by Status (Owner, Renter, Seasonal, and Year-Round Vacant)



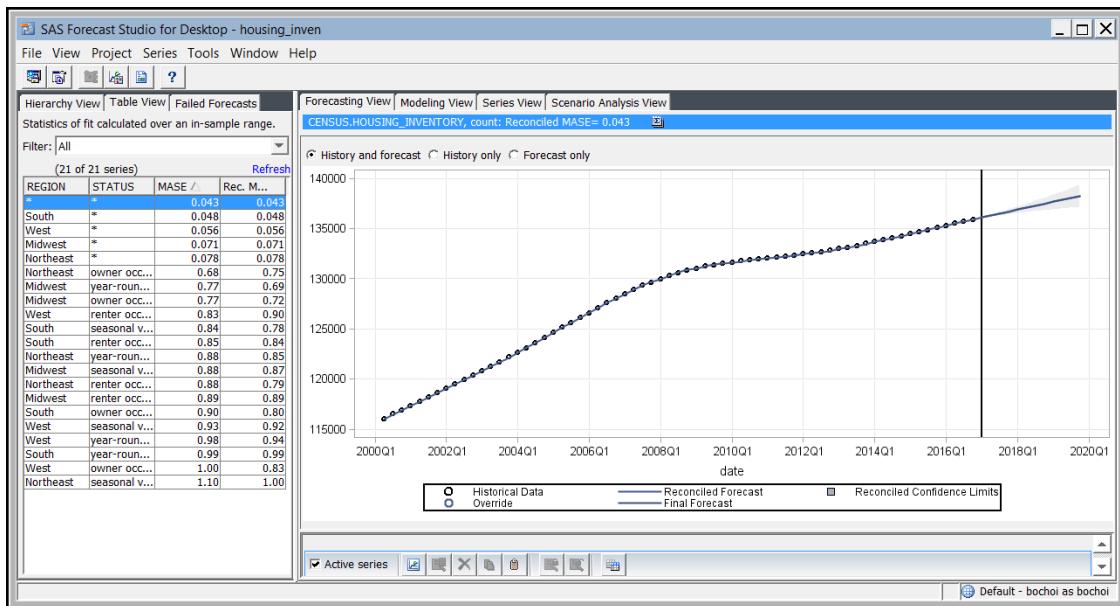


Overall, the most conspicuous region might be the Northeast, with owner-occupied units declining and renter-occupied units increasing, together leading to a rapidly declining homeownership rate. In the other three regions, owner-occupied units are slowly increasing, while renter-occupied units are rapidly increasing. Thus, it seems a general phenomenon across the country that there are more renters since the Great Recession. The increasing number made sense for several years after 2008 because many citizens of the United States were forced to sell their homes because of financial hardships. In recent years, however, homeowners have enjoyed better financial health. What, then, makes people prefer renting? There has been a rapid increase in home prices, reaching beyond the precrisis peak. Some might find homes unaffordable at the current price level. In fear of another crash, some might be choosing not to own a home while others might have rushed to realize the accumulated capital gain.

In any case, the rising number of rental units might suggest an increasing proportion of investors in the housing market. One concern is that homes tend to be more liquid for investors than for people using the properties they own as their primary residence. When prices stabilize with the Federal Reserve System hiking the interest rates in 2017, and the return on investment becomes sufficiently low or turns negative, if investors suddenly list their properties on the market altogether, will it have a similar effect as what was seen when homeowners with subprime mortgages in 2008 were forced to simultaneously sell their homes due to rising interest rates?

Last, the left panel in **Figure 9.34** shows the **Table View**. The third and fourth columns display the MASE and reconciled MASE, respectively. You observe that at the bottom level, the reconciled MASE is smaller for most of the series. Because a middle-out method is being used, the two statistics must coincide at the middle level. No gain in MASE is seen at the top level.

Figure 9.34: Table View of Hierarchical Series



In summary, SAS Forecast Studio provides the ability to forecast multiple time series quickly using reasonable defaults or to intervene by choosing from a large array of options. Informative graphical summaries of results are instantly available. In cases of hierarchical collections of series, reconciliation using any of several options is available.

Chapter 10: Spectral Analysis

10.1 Introduction	333
10.2 Example: Plant Enzyme Activity	334
10.3 PROC SPECTRA	335
10.4 Tests for White Noise	337
10.5 Harmonic Frequencies.....	338
10.6 Extremely Fast Fluctuations and Aliasing	342
10.7 The Spectral Density	342
10.8 Some Mathematical Detail (Optional Reading).....	345
10.9 Estimation of the Spectrum: The Smoothed Periodogram	345
10.10 Cross-Spectral Analysis.....	346
10.10.1 Interpretation of Cross-Spectral Quantities	346
10.10.2 Interpretation of Cross-Amplitude and Phase Spectra	348
10.10.3 PROC SPECTRA Statements	349
10.10.4 Cross-Spectral Analysis of the Neuse River Data.....	352
10.10.5 Details on Gain, Phase, and Pure Delay.....	354

10.1 Introduction

The modeling of time series data using sinusoidal components is called *spectral analysis*. The main tool is the *periodogram*. A very simple model appropriate for spectral analysis is a mean plus a sinusoidal wave plus white noise:

$$Y_t = \mu + \alpha(\sin(\omega t + \delta)) + e_t = \mu + \alpha(\sin(\delta)\cos(\omega t) + \cos(\delta)\sin(\omega t)) + e_t$$

Here, the formula, $\sin(A + B) = \cos(A)\sin(B) + \sin(A)\cos(B)$ for the sine of the sum of two angles, has been applied. The function $\mu + \alpha(\sin(\omega t + \delta))$ oscillates between $\mu - \alpha$ and $\mu + \alpha$ in a smooth and exactly periodic fashion. The number α is called the *amplitude*. The number δ , in radians, is called the *phase shift* or *phase angle*. The number ω is called the *frequency* and is also measured in radians. If an arc of length r is measured along the circumference of a circle whose radius is r , then the angle obtained by connecting the arc's ends to the circle center is one radian. There are 2π radians in a full 360-degree circle, and one radian is $360/(2\pi) = 360/6.2832 = 57.3$ degrees. A plot of $\mu + \alpha(\sin(\omega t + \delta))$ versus t is a sine wave that repeats every $2\pi / \omega$ time units. That is, the period is $2\pi / \omega$. A sinusoid of period 12 would go through $\omega = 2\pi / 12 = 0.52$ radians per observation.

Letting $A = \alpha\sin(\delta)$ and $B = \alpha\cos(\delta)$, you see the following:

$$Y_t = \mu + A \cos(\omega t) + B \sin(\omega t) + e_t$$

This is a nice expression in that if ω is known, variables $\sin(\omega t)$ and $\cos(\omega t)$ can be constructed in a DATA step. The parameters μ , A , and B can be estimated by ordinary least squares as in PROC REG. From the expressions for A and B , it is seen that $B / A = \tan(\delta)$ and

$$\sqrt{A^2 + B^2} = \alpha \sqrt{\cos^2(\delta) + \sin^2(\delta)} = \alpha$$

Therefore, phase angle and amplitude estimates can be constructed from estimates of A and B .

10.2 Example: Plant Enzyme Activity

For example, Chiu-Yueh Hung, a researcher in the Department of Genetics at North Carolina State University, collected observations on leaf enzyme activity Y every 4 hours over 5 days. There are 6 observations per day and 30 observations in all. Each observation is an average of several harvested leaves. The researcher anticipated a 12-hour enzyme cycle, which corresponds to 3 observations. To focus this discussion on periodic components, the original data have been detrended using linear regression.

First, read in the data, creating the sine and cosine variables for a period 3 (frequency $2\pi / 3$ cycles per observation), and then regress Y on these two variables.

```
data plants;
  title "ENZYME ACTIVITY";
  title2 "(DETRENDED)";
  do t=1 to 30; input y @@; pi=3.1415926;
    s1=sin(2*pi*t/3); c1=cos(2*pi*t/3);
    output;
  end;
  datalines;
265.945 290.385 251.099 285.870 379.370 301.173
283.096 306.199 341.696 246.352 310.648 276.348
234.870 314.744 261.363 321.780 313.289 253.460
307.988 303.909 284.128 252.886 317.432 287.160
213.168 308.458 296.351 283.666 333.544 316.998
;
run;

proc reg data=plants;
  model y = s1 c1/ssl;
  output out=out1 predicted=p residual=r;
run;
```

The analysis of variance table is shown in **Output 10.1**.

Output 10.1: Plant Enzyme Sinusoidal Model

The REG Procedure
Model: MODEL1
Dependent Variable: Y

Number of Observations Read	30
Number of Observations Used	30

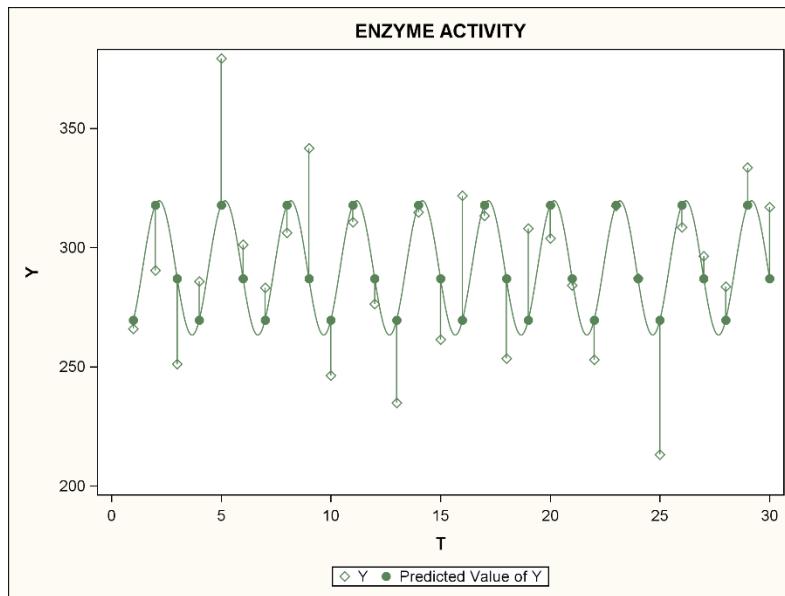
Analysis of Variance					
Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	2	11933	5966.44520	7.09	0.0034
Error	27	22724	841.62133		
Corrected Total	29	34657			

Root MSE	29.01071	R-Square	0.3443
Dependent Mean	291.44583	Adj R-Sq	0.2957
Coeff Var	9.95407		

Parameter Estimates						
Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > t	Type I SS
Intercept	1	291.44583	5.29661	55.03	<.0001	2548220
S1	1	-27.84889	7.49053	-3.72	0.0009	11633
C1	1	-4.46823	7.49053	-0.60	0.5558	299.47664

The sum of squares for the intercept is $n\bar{Y}^2 = 30(291.44583^2) = 2548220$. The sum of squares for the model, which is the sum of squares associated with frequency $\omega = 2\pi / 3$, is 11933 and has 2 degrees of freedom. It is statistically significant based on the F test, $F = 7.09$ ($p = 0.0034$). It appears that the sine term is significant, but not the cosine term. However, splitting the 2 degrees of freedom sum of squares is not meaningful in that if $t = 0$ had been used as the first time index rather than $t = 1$, both would have been significant. The sum of squares 11933 would not change with this time shift. The sum of squares 11933 associated with frequency $\omega = 2\pi / 3$ is called the *periodogram ordinate* at that frequency. A given set of data might have important fluctuations at several frequencies. **Output 10.2** shows the actual and fitted values for the plant enzyme data.

Output 10.2: Data and Sinusoidal Predictions



10.3 PROC SPECTRA

Periodogram ordinates are calculated for a collection of frequencies known as the *Fourier frequencies*. With $2m + 1$ observations Y , there are m of these, each with 2 degrees of freedom, so that a multiple regression of Y on the $2m$ sine and cosine columns fits the data perfectly. That is, there are no degrees of freedom for error. The Fourier frequencies are $(2\pi j / n)$, where j runs from 1 to m . For each j , two columns $\sin(2\pi jt / n)$ and $\cos(2\pi jt / n)$, $t = 1, 2, \dots, n$, are created. The model sum of squares, when the data are regressed on these two columns, is the j th periodogram ordinate. At the j th Fourier frequency, the sine and cosine run through j cycles in the time period covered by the data. If $n = 2m$, an even number, there are still m periodogram ordinates and j still runs from 1 to m , but when $j = m$, the frequency $2\pi j / n$ becomes $2\pi m / (2m) = \pi$ and $\sin(\pi t) = 0$. Thus, for even n , the last Fourier frequency has only one degree of freedom associated with it, arising from the cosine term, $\cos(\pi t) = (-1)^t$, only. It does not matter whether a multiple regression using all the Fourier sine and cosine columns or m bivariate regressions, one for each j , are run. The columns are all orthogonal to each other, and the sums of squares (periodogram ordinates) are the same either way.

PROC SPECTRA calculates periodogram ordinates at all the Fourier frequencies. With the 30 plant enzyme measurements, there are 15 periodogram ordinates, the last having 1 degree of freedom and the others having 2 degrees of freedom, each. Because $2\pi 10 / 30 = 2\pi / 3$, the Fourier frequency for $j = 10$ should have a periodogram ordinate equal to the previously computed model sum of squares, 11933. You might expect the other periodogram ordinates to add to 22724, the error sum of squares. However, PROC SPECTRA associates twice the correction term, $2n\bar{Y}^2 = 5096440$, with frequency 0, and twice the sum of squares at frequency π (when n is even) with that frequency. So, you must divide the frequency π ordinate by 2 to get its contribution to the error sum of squares from regression. This doubling is done here because after doubling some, division of all ordinates by 2 becomes the same as dividing unadjusted numbers by their degrees of freedom. The frequency 0 ordinate is replaced with 0 when the option ADJMEAN is used in PROC SPECTRA. These ideas are illustrated in the following code for the plant enzyme data:

```
proc spectra data=plants out=out2 coeff;
  var y;
run;
data out2; set out2; sse = p_01;
  title j=center "ENZYME DATA";
  if period=3 or period=. then sse=0;
  if round (freq, 0.0001) = 3.1416 then sse = 0.5*p_01;
run;
proc print data=out2;
  sum sse;
run;
```

The option COEFF in PROC SPECTRA adds the regression coefficients (\cos_01 and \sin_01) to the data. Looking at the period 3.0000 line of **Output 10.3**, you see the regression sum of squares $11933 = P_01$, which matches the regression output. The coefficients $A = -21.88$ and $B = 17.79$ are those that would have been obtained if time t had been labeled as 0, 1, ..., 29 (as PROC SPECTRA does), instead of 1, 2, ..., 30. Any periodogram ordinate with 2 degrees of freedom can be computed as $(n/2)(A^2 + B^2)$, where A and B are its Fourier coefficients. You see that $(30/2)((-21.88)^2 + (17.79)^2) = 11933$ in **Output 10.3**.

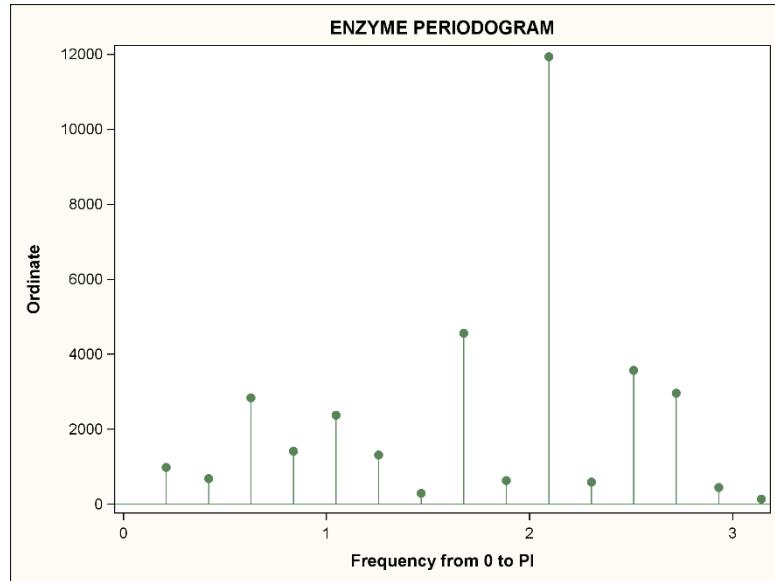
Output 10.3: OUT Data Set from PROC SPECTRA

Obs	FREQ	PERIOD	COS_01	SIN_01	P_01	SSE
1	0.00000	.	582.892	0.0000	5096440.43	0.00
2	0.20944	30.0000	5.086	6.3073	984.78	984.78
3	0.41888	15.0000	1.249	6.6450	685.74	685.74
4	0.62832	10.0000	-6.792	-11.9721	2841.87	2841.87
5	0.83776	7.5000	-1.380	-9.6111	1414.15	1414.15
6	1.04720	6.0000	-10.035	-7.5685	2369.85	2369.85
7	1.25664	5.0000	0.483	-9.3467	1313.91	1313.91
8	1.46608	4.2857	4.288	-1.0588	292.58	292.58
9	1.67552	3.7500	17.210	-2.7958	4560.07	4560.07
10	1.88496	3.3333	-6.477	-0.1780	629.66	629.66
11	2.09440	3.0000	-21.884	17.7941	11932.89	0.00
12	2.30383	2.7273	-5.644	-2.7635	592.35	592.35
13	2.51327	2.5000	6.760	13.8749	3573.13	3573.13
14	2.72271	2.3077	-9.830	-10.0278	2957.94	2957.94
15	2.93215	2.1429	-0.022	-5.4249	441.45	441.45
16	3.14159	2.0000	2.973	0.0000	132.60	66.30
						22723.78

PROC SPECTRA automatically creates the column FREQ of Fourier frequencies equally spaced in the interval 0 to π and the column PERIOD of corresponding periods. It is customary to plot the periodogram versus frequency or period, omitting frequency 0.

Output 10.4 shows the unusually large ordinate 11933 at the anticipated frequency of one cycle per 12 hours—that is, one cycle per 3 observations.

Output 10.4: Periodogram with a Single Important Frequency



The researcher was specifically looking for this cycle and took sufficient observations to make the frequency of interest a Fourier frequency. If the important frequency is not a Fourier frequency, the periodogram ordinates with frequencies near the important one will be large. By creating their own sine and cosine columns, researchers can always investigate any frequency using regression. The beauty of the Fourier frequencies is the orthogonality of the resulting collection of regression columns (sine and cosine functions).

10.4 Tests for White Noise

For a normal white noise series with variance σ^2 , the periodogram ordinates are independent and, when divided by σ^2 , have chi-square distributions with 2 degrees of freedom (df). These properties lead to tests of the white noise null hypothesis.

You are justified in using an F test for the single sinusoid plus white noise model when the appropriate ω is known in advance, as in section 10.2. You would not be justified in testing the largest observed ordinate (just because it is the largest) with F . If you test for a period 3 component in multiple sets of white noise data (your null hypothesis), the F test statistic will have an F distribution. However, if you always test the largest ordinate whether or not it occurs at period 3, then this new F statistic will never be less than the F for period 3, and it will usually be larger. Clearly, this new F statistic cannot have the same F distribution.

Fisher computed the distribution for the largest periodogram ordinate divided by the mean of all the 2 degrees of freedom ordinates under the white noise null hypothesis. In the plant enzyme data, omission of the 1 degree of freedom ordinate 132.6 gives Fisher's κ test statistic $11933 / [(22723.8 - 132.6 / 2 + 11933) / 14] = 4.83$. Fuller (1996) discusses this test and the *cumulative periodogram* test. The latter uses C_k , which is the ratio of the sum of the first k periodogram ordinates to the sum of all the ordinates (again dropping any 1 degree of freedom ordinate). The set of these C_k should behave like an ordered sample from a uniform distribution if the data are white noise. Therefore, a standard distributional test, such as those in PROC UNIVARIATE, can be applied to these cumulative C_k ratios, resulting in a test of the white noise null hypothesis. Traditionally, the Kolmogorov-Smirnov test is applied. For more details, see Fuller (1996, p. 363).

Interpolating in Fuller's table of critical values for Fisher's κ with 14 ordinates gives 4.385 as the 10% critical value and 4.877 as the 5% critical value. The value 4.83 is significant at 10%, but not quite at 5%. Therefore, if you were just searching for a large ordinate rather than focusing from the start on a 12-hour cycle, your evidence for a 12-hour cycle would be nowhere near as impressive. This illustrates the increase in statistical power that can be obtained when you

know something about your subject matter. You obtain both white noise tests using the WHITETEST option, as shown in **Output 10.5**.

```
proc spectra data=plants whitetest;
  var y;
run;
```

Output 10.5: Periodogram-Based White Noise Tests

The SPECTRA Procedure

Test for White Noise for Variable Y	
M-1	14
Max(P ^(*))	11932.89
Sum(P ^(*))	34590.36

Fisher's Kappa: (M- 1)*Max(P ^(*))/Sum(P ^(*))	
Kappa	4.829682

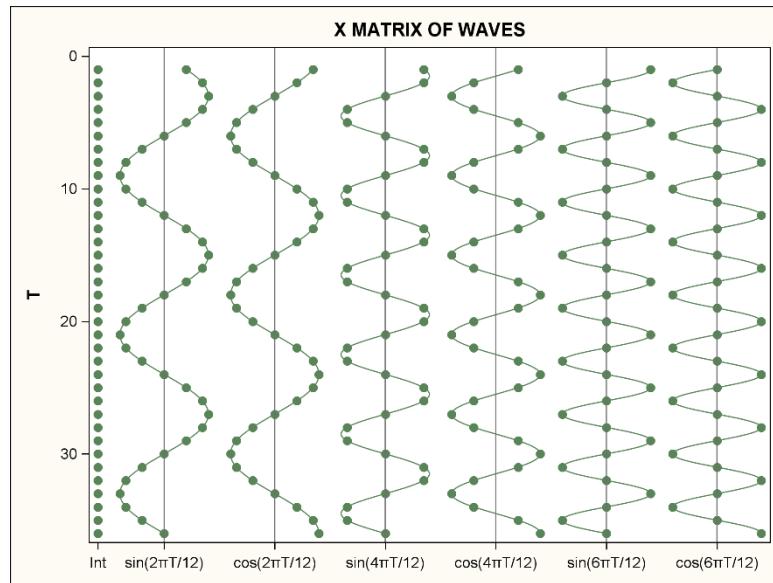
Bartlett's Kolmogorov-Smirnov Statistic: Maximum absolute difference of the standardized partial sums of the periodogram and the CDF of a uniform(0,1) random variable.	
Test Statistic	0.255984
Approximate P-Value	0.3180

For 14 periodogram ordinates, tables of the Kolmogorov-Smirnov statistic indicate that a value larger than about 0.36 would be needed for significance at the 5% level, so 0.256 is not big enough. Fisher's test is designed to detect a single sinusoid buried in white noise. It would be expected to be more powerful under the model proposed than the Kolmogorov-Smirnov test, which is designed to have some power against any departure from white noise.

10.5 Harmonic Frequencies

Just because a function is periodic does not necessarily mean it is a pure sinusoid. For example, the sum of a sinusoid of period k and another of period $k/2$ is a periodic function of period k , but it is not expressible as a single sinusoid. On the other hand, any periodic function of period k defined on the integers can be represented as the sum of sinusoids of period $k, k/2, k/3$, and so on. For a fundamental period k , periods k/j for $j = 2, 3, \dots$, are called *harmonics*. Harmonics affect the wave shape, but not the period. A period of 2 is the shortest period detectable in a periodogram, and its associated frequency, π , is sometimes called the *Nyquist frequency*. Thus, the plant enzyme measurements were not taken frequently enough to investigate harmonics of the fundamental frequency $2\pi/3$ (period 3). Even the first harmonic has period $3/2 < 2$ and frequency $4\pi/3$, which exceeds the Nyquist frequency π .

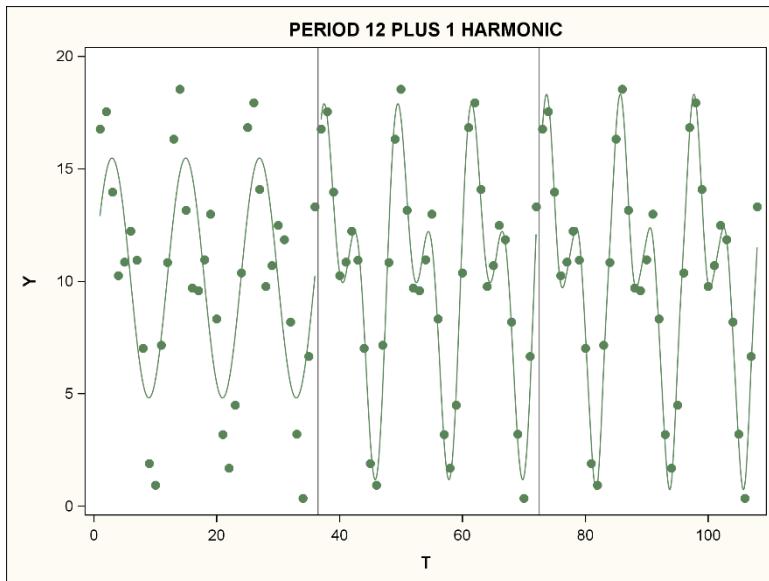
To further illustrate the idea of harmonics, imagine $n = 36$ monthly observations where there is a fundamental frequency $2\pi/12$ and possibly contributions from the harmonic frequencies $2(2\pi)/12$ and $3(2\pi)/12$ plus white noise. To fit the model, you create three sine and three cosine columns. The sine column for the fundamental frequency would have t th entry $\sin(2\pi t/12)$ and would go through 3 cycles in 36 observations. Look at **Output 10.6**.

Output 10.6: Fundamental and Harmonic Sinusoids

Output 10.6 is a schematic representation of the regression X matrix and is interpreted as follows. On the left, a vertical column of dots represents the intercept column, a column of 1s. Just to its right is a wave that represents $\sin(2\pi t / 12)$, and to its right is another wave representing $\cos(2\pi t / 12)$. Run your finger down one of these two waves. Your finger cycles between one unit left and one unit right of the wave's center line. There are three cycles in each of these two columns. Writing the deviations of dots from the center as numbers provides the entries of the corresponding column of the X matrix. These two waves or columns of X will have regression coefficients A_1, B_1 . With these coefficients, the regression will exactly fit any sinusoid of frequency $2\pi / 12$ regardless of its amplitude and phase.

Similar comments apply to the other two pairs of waves, but note that as you run your finger down these, the left-to-right oscillation is faster. There are more cycles: $36 / 6 = 6$ for the middle pair and $36 / 4 = 9$ for the right-most pair, where $12 / 2 = 6$ and $12 / 3 = 4$ are the periods corresponding to the two harmonic frequencies. Three more pairs of columns, with periodicities $12 / 4 = 3$, $12 / 5$, and $12 / 6 = 2$, complete a full set of harmonics for a period 12 function measured at integer time points. They would add 6 more columns for a total of 12 waves, seeming to contradict the fact that a period 12 function has 11, not 12, degrees of freedom. However, at period $12 / 6 = 2$, the sine column becomes $\sin(2\pi t / 2) = \sin(\pi t) = 0$ for all t . Such a column of 0s would be omitted, leaving 11 columns (11 degrees of freedom), plus an intercept column associated with the period 12 function. If 36 consecutive observations from any period 12 function were regressed on this 12-column X matrix, the fit would be perfect at the observed points, but it would not necessarily interpolate well between them. A perfect fit at the observation points would result even if the sequence Y_t were repeated sets of six 1s followed by six -1s. The fitted values would exactly match the observed -1,1 pattern at integer values of t , but interpolated values at time $t = 5.9$ for example, would not be restricted to -1 or 1. You might envision the harmonics as fine-tuning the wave shape as you move up through the higher harmonic frequencies (shorter period fluctuations). This motivates the statistical problem of separating the frequencies that contribute to the true process from those that are fitting just random noise so that a good picture of the wave shape results. Periodograms and associated tests are useful here.

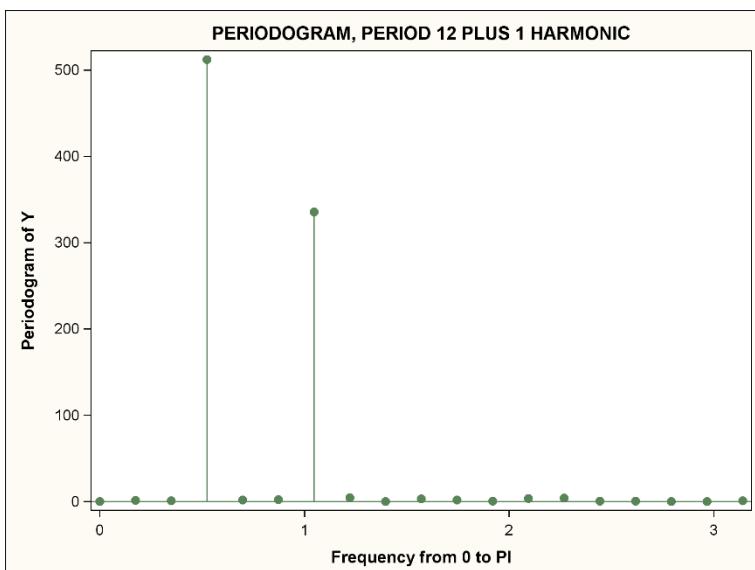
The following outputs are generated from a sinusoid of period $k = 12$ plus another at the first harmonic, period $12 / 2 = 6$. Each sinusoid is the sum of a sine and cosine component, thus allowing an arbitrary phase angle. For interpolation purposes, sine and cosine terms are generated for t in increments of 0.1, but Y exists only at integer t .

Output 10.7: Increased Resolution Using Harmonics

Output 10.7 shows three sets of fitted values. The sine and cosine at the fundamental frequency $\omega_1 = 2\pi / 12$ are used to produce the fitted values on the left side. These fitted values do not capture the double peak in each interval of 12 time points, and they miss the low and high extremes. Including the first harmonic $\omega_2 = 2(2\pi / 12)$ gives a better fit and creates an idea of what the data-generating function looks like. The fitted values on the right side are those coming from the fundamental and all harmonic frequencies $j(2\pi / 12)$ for $j = 1, 2, \dots, 6$, omitting the sine at $j = 6$. The minor wiggles are due to the frequencies with $j > 2$. Adding all those extra parameters does not seem to have produced any useful new features in the fitted values. From PROC REG (as will be seen in **Output 10.9** later), the F test 1.53 for frequencies with $j = 3, 4, \dots, 6$, is not significant. The Type I sums of squares for $j = 1$ and 2 are large enough that neither the $j = 1$ nor $j = 2$ frequency can be omitted. Recall that you would not eliminate just a sine or cosine—they are treated as pairs. Rearrangement of terms or deletion of some terms would not affect the sums of squares because the sine and cosine columns correspond to Fourier frequencies, so they are orthogonal to each other.

The following PROC SPECTRA code is used to generate **Output 10.8** and **Output 10.10**:

```
proc spectra data=compress p s adjmean out=outspectra;
  var y;
  weights 1 2 3 4 3 2 1;
run;
```

Output 10.8: Periodogram with Two Independent Frequencies

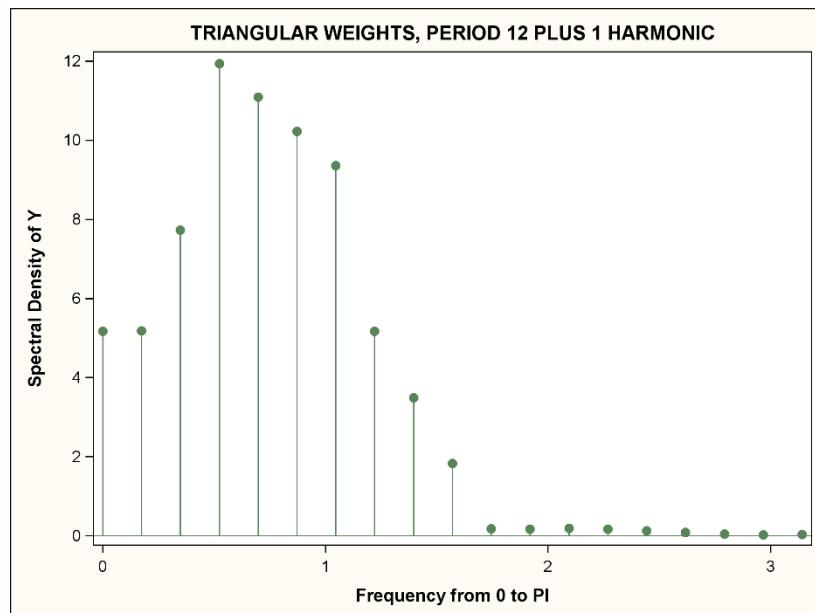
The periodogram in **Output 10.8** makes it clear that there are two dominant frequencies, $2\pi / 12$ and its first harmonic, $4\pi / 12$. The last few lines of the program deliver a smoothed version of the periodogram, shown in **Output 10.10**, that is discussed in section 10.9. Smoothing is not helpful in this particular example.

Output 10.9: Regression Estimates and F Test

Parameter Estimates						
Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > t	Type I SS
Intercept	1	10.13786	0.14369	70.55	<.0001	3699.94503
S1	1	5.33380	0.20321	26.25	<.0001	512.09042
C1	1	0.10205	0.20321	0.50	0.6201	0.18745
S2	1	3.90502	0.20321	19.22	<.0001	274.48586
C2	1	1.84027	0.20321	9.06	<.0001	60.95867
S3	1	-0.33017	0.20321	-1.62	0.1173	1.96220
S4	1	-0.41438	0.20321	-2.04	0.0526	3.09082
S5	1	-0.17587	0.20321	-0.87	0.3954	0.55672
C3	1	-0.27112	0.20321	-1.33	0.1947	1.32314
C4	1	-0.16725	0.20321	-0.82	0.4186	0.50351
C5	1	-0.02326	0.20321	-0.11	0.9098	0.00974
C6	1	-0.11841	0.14369	-0.82	0.4180	0.50473

Test HARMONICS Results for Dependent Variable Y				
Source	DF	Mean Square	F Value	Pr > F
Numerator	7	1.13584	1.53	0.2054
Denominator	24	0.74330		

Output 10.10: Smoothed Periodogram



10.6 Extremely Fast Fluctuations and Aliasing

Suppose a series actually has a frequency larger (faster fluctuations) than the Nyquist frequency π radians per observation (for example, $4\pi / 3 > \pi$). Imagine a wheel with a dot on its edge, and an observer who looks at the wheel each second. If the wheel rotates clockwise $4\pi / 3$ radians per second, at the first observation, the dot will now be $2\pi / 3$ radians counterclockwise (for example, $-2\pi / 3$ radians) from its previous position, and similarly for subsequent observations. Based on the dot's position, the observer knows only that the frequency of rotation is $-2\pi / 3 + 2\pi j$ for some integer j . These frequencies are all said to be *aliased* with $-2\pi / 3$, where this frequency was selected because it is in the interval $[-\pi, \pi]$. Another alias is $2\pi / 3$ (as if the observer has moved to the other side of the wheel).

Because $A\cos(\omega t) + B\sin(\omega t) = A\cos(-\omega t) - B\sin(-\omega t)$, it is not possible to distinguish a cycle of frequency ω from one of $-\omega$ using the periodogram. Thus, it is sufficient and customary to compute periodogram ordinates at the Fourier frequencies $2\pi j / n$ with $j = 1, 2, \dots, m$ so that $0 \leq 2\pi j / n \leq \pi$. Recall that the number of periodogram ordinates m is either $(n - 1) / 2$ if n is odd or $n / 2$ if n is even.

Imagine a number line with reference points at πj for all integers j , positive, negative, and zero. Folding that line back and forth in accordion fashion at these reference points maps the whole line into the interval $[0, \pi]$. The set of points that map into any ω are its aliases. For that reason, the Nyquist frequency π is also referred to as the *folding frequency*. The reason that this frequency has names instead of always being called π is that some people prefer radians or cycles per second, per hour, per day, and so on, rather than radians per observations as a unit of measure. If observations are taken every 15 minutes, the Nyquist frequency π radians per observation would convert to 4π radians per hour, or 2 cycles per hour. In this book, radians per observation and the Nyquist frequency π are the standard.

When the periodogram is plotted over $[0, \pi]$ and there appears to be a cycle at a bizarre frequency in $[0, \pi]$, ask yourself if this might be coming from a cycle beyond the Nyquist frequency.

10.7 The Spectral Density

Consider three processes: $W_t = 10 + (5 / 3)e_t$, $Y_t = 10 + 0.8(Y_{t-1} - 10) + e_t$, and $Z_t = 10 - 0.8(Z_{t-1} - 10) + e_t$, where $e_t \sim N(0, 0.36)$ is white noise. Each process has mean 10 and variance 1.

The spectral density function of a process is defined as

$$f(\omega) = \frac{1}{2\pi} \sum_{h=-\infty}^{\infty} \gamma(h) \cos(\omega h)$$

where $\gamma(h)$ is the autocovariance function. The function is symmetric: $f(\omega) = f(-\omega)$. For W_t , the variance is $\gamma(0) = \sigma^2 = 1$ and $\gamma(h) = 0$ if h is not 0. The spectral density for W_t becomes the following:

$$f_w(\omega) = \frac{1}{2\pi} \sum_{h=-\infty}^{\infty} \gamma(h) \cos(\omega h) = \frac{1}{2\pi} \gamma(0) \cos(0) = \frac{1}{2\pi}$$

This is $\sigma^2 / 2\pi$ for a general white noise series with variance σ^2 . Sometimes the spectral density is plotted over the interval $-\pi \leq \omega \leq \pi$. Because for white noise, $f(\omega)$ is $\sigma^2 / 2\pi$, the plot is a rectangle of height $\sigma^2 / 2\pi$ over an interval of width 2π . The area of the rectangle, $2\pi\sigma^2 / 2\pi = \sigma^2$, is the variance of the series. This (area = variance) will be true in general of a spectral density whether or not it is plotted as a rectangle.

Because the plot of $f_w(\omega)$ has equal height at each ω , it is said that all frequencies contribute equally to the variance of W_t . This is the same idea as white light, where all frequencies of the light spectrum are equally represented, or as white noise in acoustics. In other words, the time series is conceptualized as the sum of sinusoids at various frequencies with white noise having equal contributions for all frequencies. In general, the interpretation of the spectral density is the decomposition of the variance of a process into components at different frequencies.

An interesting mathematical fact is that if the periodogram is computed for data from any ARMA model, the periodogram ordinate at any Fourier frequency $0 < \omega < \pi$ estimates $4\pi f(\omega)$. That is, $4\pi f(\omega)$ is (approximately) the periodogram ordinate's expected value. Dividing the periodogram ordinate by 4π gives an almost unbiased estimate of $f(\omega)$. If the plot over $-\pi \leq \omega \leq \pi$ is desired (so that the area under the curve is the variance), use the symmetry of $f(\omega)$ and plot the estimate at both ω and $-\omega$. For white noise, $f(\omega)$ estimates the same thing ($\sigma^2 / 2\pi$) at each ω . Averaging several $f(\omega)$ values gives an even better estimate. Local averaging of estimates frequently, but not always, improves estimation.

Often only the positive frequency half of the estimated spectral density is plotted. It is left to the reader to remember that the variance is twice the area of the plot.

What do the spectral densities of Y_t and Z_t look like? Using a little intuition, you would expect the positively autocorrelated series Y_t to fluctuate at a slower rate around its mean than does W_t . Likewise, you would expect the negatively autocorrelated series Z_t to fluctuate faster than W_t because, for Z_t , positive deviations tend to be followed by negative and negative by positive. The slower fluctuation in Y_t should show up as longer period waves—that is, higher periodogram ordinates at low frequencies. For Z_t , you would expect the opposite—large contributions to the variance from frequencies near $-\pi$ or π .

The three graphs at the top of **Output 10.11** show the symmetrized periodograms for W , Y , and Z , each computed from 1000 simulated values. Each graph has each ordinate plotted at the associated ω and its negative to show the full symmetric spectrum. The behavior is as expected—high values near $\omega = 0$ indicating low-frequency waves in Y_t ; high values near the extreme ω s for Z_t indicating high-frequency fluctuations; and a flat spectrum for W_t . Two other periodograms are shown. The first, in the bottom left corner, is for $D_t = Y_t - Y_{t-1}$. Because D_t is a moving linear combination of Y_t values, D_t is referred to as a filtered version of Y_t . If the filter $Y_t - 0.8Y_{t-1}$ had been applied, the filtered series would just be white noise, and the spectral density would just be a horizontal line. Linear filtering of this sort is a way of altering the spectral density of a process. The differencing filter has overcompensated for the autocorrelation, depressing the middle (near 0 frequency) periodogram ordinates of D_t too much so that instead of being level, the periodogram dips down to 0 at the middle.

In the wide middle panel of **Output 10.11**, a sinusoid $0.2\sin(2\pi t / 25 + 0.1)$ has been added to Y_t . The first 200 observations from both the original and altered series have been plotted. Because the amplitude of the sinusoid is so small, the plot of the altered Y_t is nearly indistinguishable from the original Y_t . The same is true of the autocorrelations. In contrast, the periodogram in the middle of the bottom row shows a strong spike at the Fourier frequency $2\pi / 25 = 40(2\pi / 1000) = 0.2513$ radians, clearly exposing the modification to Y_t . The middle graphs in the top (original Y_t) and bottom rows are identical except for the spikes at frequency ± 0.2513 .

The bottom right graph of **Output 10.11** contains plots of three smoothed spectral density estimates with the theoretical spectral densities that they estimate. See **section 10.8** for information about smoothing the periodogram to estimate the spectral density. The low horizontal line associated with white noise has been discussed already. For autoregressive order 1 series, AR(1) such as Y and Z , the theoretical spectral density is as follows:

$$f(\omega) = \frac{\sigma^2}{2\pi} / (1 + \rho^2 - 2\rho \cos(\omega))$$

Here, ρ is the lag 1 autoregressive coefficient, 0.8 for Y and -0.8 for Z . For a moving average (MA) such as $X_t = e_t - \theta e_{t-1}$, the spectral density is the following:

$$f_x(\omega) = \frac{\sigma^2}{2\pi} (1 + \theta^2 - 2\theta \cos(\omega))$$

Both the AR and MA spectral densities involve the white noise spectral density $\sigma^2/2\pi$. It is either multiplied (MA) or divided (AR) by a trigonometric function involving the ARMA coefficients. X_t is a filtered version of e_t . If X_t had been defined in terms of a more general time series V_t as $X_t = V_t - \theta V_{t-1}$, the spectral density of X_t would have been similarly related to that of V_t as $f_x(\omega) = f_V(\omega)(1 + \theta^2 - 2\theta \cos(\omega))$, where $f_V(\omega)$ is the spectral density of V_t . Suppose Y_t has the following spectral density:

$$f_Y(\omega) = \frac{\sigma^2}{2\pi} / (1 + \rho^2 - 2\rho \cos(\omega))$$

This is filtered to get $D_t = Y_t - \theta Y_{t-1}$. The spectral density of D_t is expressed as follows:

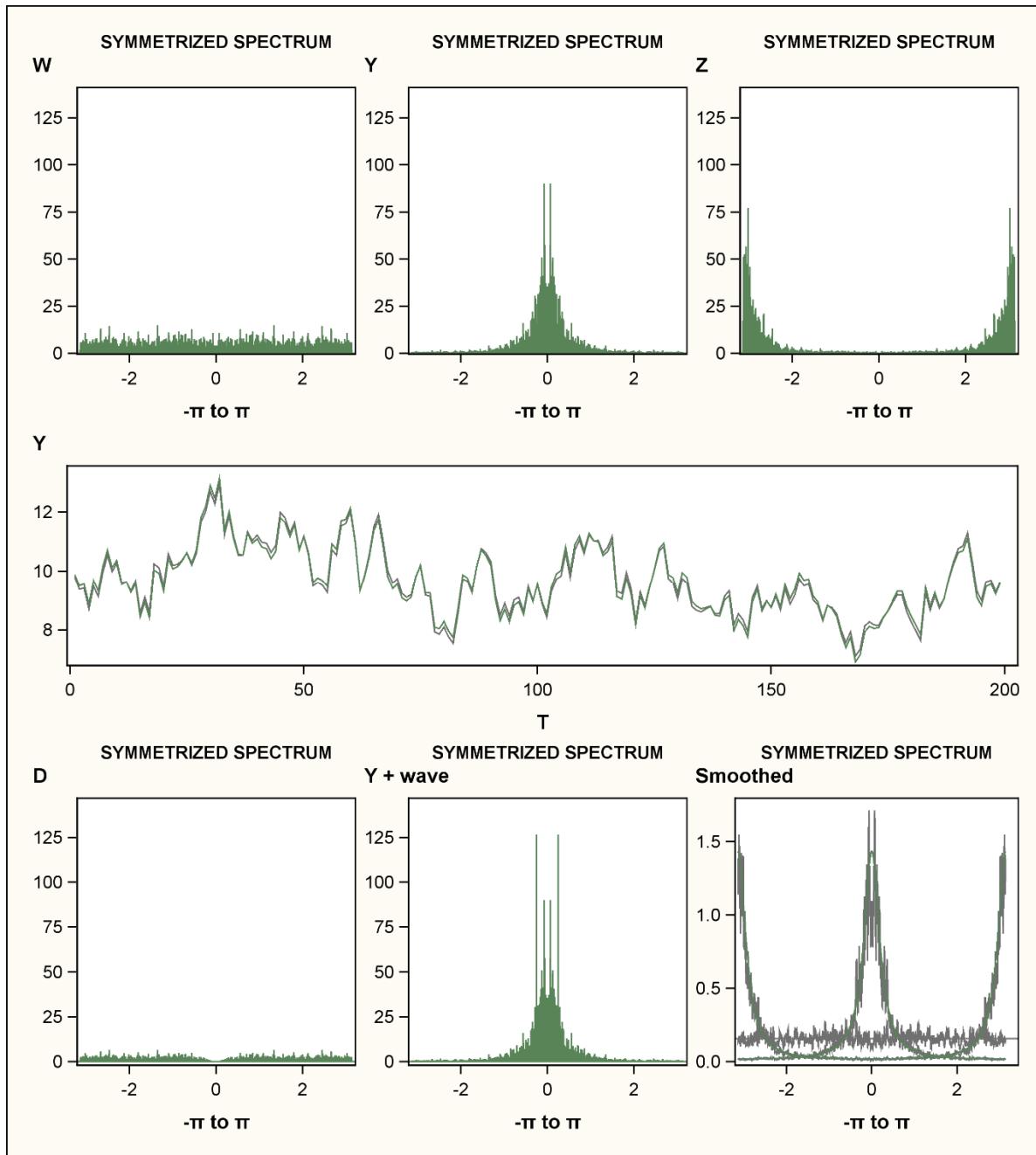
$$\begin{aligned} f_D(\omega) &= (1 + \theta^2 - 2\theta \cos(\omega)) f_Y(\omega) \\ &= \frac{\sigma^2}{2\pi} (1 + \theta^2 - 2\theta \cos(\omega)) / (1 + \rho^2 - 2\rho \cos(\omega)) \end{aligned}$$

If $D_t = Y_t - Y_{t-1}$, then $\theta = 1$, so

$$f_D(\omega) = (2 - 2\cos(\omega)) \frac{\sigma^2}{2\pi} / (1 + \rho^2 - 2\rho\cos(\omega))$$

This is 0 at frequency $\omega = 0$, and gives some insight into the behavior of the D_t periodogram displayed in the bottom left corner of **Output 10.11**. Filtering affects different frequencies in different ways. The multiplier associated with the filter, such as $(1 + \theta^2 - 2\theta\cos(\omega))$ in the previous examples, is sometimes called the *squared gain* of the filter in that amplitudes of some waves are increased (gain > 1) and some are reduced (gain < 1). Designing filters to amplify certain frequencies and reduce or eliminate others has been studied in some fields. The term *squared gain* is used because the spectral density decomposes the variance, not the standard deviation, into frequency components.

Output 10.11: Spectral Graphics



10.8 Some Mathematical Detail (Optional Reading)

The spectral densities for general ARMA processes can be defined in terms of complex exponentials $e^{i\omega} = \cos(\omega) + i\sin(\omega)$. Here, i represents an imaginary number whose square is -1 . Although that concept might be hard to grasp, calculations done with such terms often result in expressions not involving i , so the use of i along the way is a convenient mechanism for calculation of quantities that ultimately do not involve imaginary numbers.

Using the backshift operator B , the ARMA(p,q) model is expressed as follows:

$$(1 - \alpha_1 B - \cdots - \alpha_p B^p) Y_t = (1 - \theta_1 B - \cdots - \theta_q B^q) e_t$$

You now understand that these expressions in backshift can be correctly referred to as *filters*. Replace B^j with $e^{i\omega j}$, getting $A(\omega) = (1 - \alpha_1 e^{i\omega} - \alpha_2 e^{i\omega 2} - \cdots - \alpha_p e^{i\omega p})$ on the autoregressive and $M(\omega) = (1 - \theta_1 e^{i\omega} - \theta_2 e^{i\omega 2} - \cdots - \theta_q e^{i\omega q})$ on the moving average side. The complex polynomials $A(\omega)$ and $M(\omega)$ have corresponding complex conjugate expressions $A^*(\omega)$ and $M^*(\omega)$, obtained by replacing $e^{i\omega} = \cos(\omega) + i\sin(\omega)$ everywhere with $e^{-i\omega} = \cos(\omega) - i\sin(\omega)$. Start with the spectral density of e_t , which is $\sigma^2 / 2\pi$. The spectral density for the ARMA(p,q) process Y_t becomes the following:

$$f_Y(\omega) = \frac{\sigma^2}{2\pi} \frac{M(\omega) M^*(\omega)}{A(\omega) A^*(\omega)}$$

When a complex expression is multiplied by its complex conjugate, the product involves only real numbers and is positive.

If there are unit roots on the autoregressive side, the denominator $A(\omega)A^*(\omega)$ will be zero for some ω . The theoretical expression for $f_Y(\omega)$ will be undefined. This process does not have a covariance function $\gamma(h)$ that is a function of h only, so the following spectral density cannot exist either:

$$f(\omega) = \frac{1}{2\pi} \sum_{h=-\infty}^{\infty} \gamma(h) \cos(\omega h)$$

Despite the fact that $f(\omega)$ does not exist for unit root autoregressions, the periodogram can still be computed. Akdi and Dickey (1997) and Evans (1998) discuss normalization and distributional properties for the periodogram in this situation. Although they find the expected overall gross behavior (extremely large ordinates near frequency 0), they also find some interesting distributional departures from the stationary case. Unit roots on the moving average side are not a problem—they simply cause $f_Y(\omega)$ to be 0 at some ω values. An example of this is D_t , whose periodogram is shown in the bottom left corner of **Output 10.11** and whose spectral density is 0 at $\omega = 0$.

10.9 Estimation of the Spectrum: The Smoothed Periodogram

The graph in the bottom right corner of **Output 10.11** contains theoretical spectral densities for W_t , Y_t , and Z_t . It also contains estimates derived from the periodogram plotted symmetrically over the full interval $-\pi \leq \omega \leq \pi$. These estimates are derived by locally smoothing the periodogram. Smoothed estimates are local weighted averages. In that picture, a simple average of 21 ordinates centered at the frequency of interest is taken and divided by 4π . These are good estimates of the theoretical spectral densities that are overlaid in the plot. The 4π divisor is used so that the area under the spectral density curve over $-\pi \leq \omega \leq \pi$ will be the series variance. Weighted averages concentrated more on the ordinates near the one of interest can also be used. In the PROC SPECTRA output data set, the spectral density estimates are named S_01 , S_02 , and so on. The periodogram ordinates are named P_01 , P_02 , and so on, for variables in the order listed in the VAR statement.

Let $I_n(\omega_j)$ denote the periodogram ordinate at Fourier frequency $\omega_j = 2\pi j / n$ constructed from n observations on some time series Y_t . Suppose you issue these statements:

```
proc spectra p s adjmean out=outspec;
  weights 1 2 3 4 3 2 1;
  var x r y;
run;
```

The smoothed spectral density estimate for Y will have variable name S_03 in the data set OUTSPEC. For $j > 2$, it will be computed as follows:

$$\frac{1}{16} [I_n(\omega_{j-3}) + 2I_n(\omega_{j-2}) + 3I_n(\omega_{j-1}) + 4I_n(\omega_j) + 3I_n(\omega_{j+1}) + 2I_n(\omega_{j+2}) + I_n(\omega_{j+3})] / (4\pi)$$

Here, the divisor 16 is the sum of the numbers in the WEIGHT statement. Modifications are needed for $j < 4$ and $j > m - 3$, where m is the number of ordinates. **Output 10.10** shows the results of smoothing the **Output 10.8** periodogram. Much of the detail has been lost.

From the graphs in **Output 10.8** and **Output 10.10**, the sinusoids are indicated more strongly by the unsmoothed P_01 than by the smoothed spectrum S_01. That is because the smoothing spreads the effect of the sinusoid into neighboring frequencies as opposed to the periodogram, which concentrates it entirely on the true underlying Fourier frequency. On the other hand, when the true spectrum is fairly smooth, as with X , Y , Z , and D in **Output 10.11**, the estimator should be smoothed. This presents a dilemma for the researcher who is trying to discover the nature of the true spectrum. The best way to smooth the spectrum for inspection is not known without knowing the nature of the true spectrum. In that case, inspecting its estimate is of no interest. To address this, several graphs are made using different degrees of smoothing. The less smooth ones reveal spikes and the more smooth ones reveal the shape of the smooth regions of the spectrum.

Dividing each periodogram ordinate by the corresponding spectral density $f(\omega)$ results in a set of almost independent variables, each with approximately (exactly if the data are normal white noise) a chi-square distribution with 2 degrees of freedom, a highly variable distribution. The weights applied to produce the spectral density lower the variance and usually introduce a bias. The set of weights is called a *spectral window*, and the effective number of periodogram ordinates involved in an average is called the *bandwidth* of the window. The estimated spectral density approximates a weighted average of the true spectral density in an interval surrounding the target frequency rather than just at the target frequency. The interval is larger for larger bandwidths. Therefore, the resulting potential for bias increased, whereas the variance of the estimate is decreased by increasing the bandwidth.

10.10 Cross-Spectral Analysis

Just as cross-correlations can measure relationships between two series in the time domain, cross-spectral quantities display relationships between two series in the frequency domain.

10.10.1 Interpretation of Cross-Spectral Quantities

Interpreting cross-spectral quantities is closely related to the transfer function model in which an output time series, Y_t , is related to an input time series, X_t , through the following equation:

$$Y_t = \sum_{j=-\infty}^{\infty} v_j X_{t-j} + \eta_t$$

In this equation, η_t is a time series independent of the input, X_t . For the moment, assume $\eta_t = 0$.

For example, let Y_t and X_t be related by the transfer function:

$$Y_t - .8Y_{t-1} = X_t$$

Then, consider the following, which is a weighted sum of current and previous inputs:

$$Y_t = \sum_{j=0}^{\infty} (.8)^j X_{t-j}$$

Cross-spectral quantities tell you what happens to sinusoidal inputs. In the example, suppose X_t is the sinusoid:

$$X_t = \sin(\omega t)$$

In this sinusoid, $\omega = 2\pi/12$. Using trigonometric identities shows that Y_t satisfying $Y_t - .8Y_{t-1} = \sin(\omega t)$ must be of the form $Y_t = A \sin(\omega t - B)$.

Next, solve the following:

$$\begin{aligned} & A \sin(\omega t - B) - 0.8A \sin(\omega t - B - \omega) \\ &= (A \cos(B) - 0.8A \cos(B + \omega)) \sin(\omega t) \\ &\quad + (-A \sin(B) + 0.8A \sin(B + \omega)) \cos(\omega t) \\ &= \sin(\omega t) \end{aligned}$$

The solution must have $A \cos B - 0.8 \cos(B + \omega) = 1$ and $-A \sin B + 0.8 \sin(B + \omega) = 0$. The solution then follows:

$$\tan(B) = .08 \sin(\omega) / (1 - 0.8 \cos(\omega)) = 0.4 / (1 - (0.4)\sqrt{3}) = 1.3022$$

and

$$A = 1 / \left\{ 0.6091 - 0.8 \left[0.6091 (\sqrt{3}/2) - 0.7931 (1/2) \right] \right\} = 1.9828$$

The transfer function produces output with amplitude 1.9828 times that of the input. It has the same frequency and a phase shift of $\arctan(1.3022) = 52.5^\circ = 0.92$ radians. These results hold only for $\omega = 2\pi/12$. The output for any noiseless linear transfer function is a sinusoid of frequency ω when the input X is such a sinusoid. Only the amplitude and phase are changed.

In cross-spectral analysis, using arbitrary input and its associated output, you simultaneously estimate the gain and phase at all Fourier frequencies. An intermediate step is the computation of quantities called *cospectrum* and *quadrature spectrum*.

The theoretical cross-spectrum, $f_{xy}(\omega)$, is the Fourier transform of the cross-covariance function $\gamma_{xy}(h)$, where the following holds:

$$\gamma_{xy}(h) = E \{ [X_t - E(X_t)][Y_{t+h} - E(Y_t)] \}$$

The real part of $f_{xy}(\omega) = c(\omega) - iq(\omega)$ is the cospectrum, $c(\omega)$, and the imaginary part gives the quadrature spectrum, $q(\omega)$. In the following example, multiply both sides by X_{t-h} and take the expected value:

$$Y_t - 0.8Y_{t-1} = X_t$$

You obtain this result:

$$\gamma_{xy}(h) - 0.8\gamma_{xy}(h-1) = \gamma_{xx}(h)$$

Here, $\gamma_{xx}(h)$ is the autocorrelation function (ACF) for x . Now, when $\gamma(h)$ is absolutely summable, then the following holds true:

$$f(\omega) = (2\pi)^{-1} \sum_{h=-\infty}^{\infty} (\gamma(h) e^{-i\omega h})$$

From these last two equations, the following sequence of equivalent equations is obtained:

$$\begin{aligned} & (2\pi)^{-1} \sum_{h=-\infty}^{\infty} (\gamma_{xy}(h) e^{-i\omega h} - 0.8\gamma_{xy}(h-1) e^{-i\omega h}) \\ &= (2\pi)^{-1} \sum_{h=-\infty}^{\infty} (\gamma_{xx}(h) e^{-i\omega h}) \end{aligned}$$

or

$$\begin{aligned} & (2\pi)^{-1} \sum_{h=-\infty}^{\infty} (\gamma_{xy}(h) e^{-i\omega h} - 0.8\gamma_{xy}(h-1) e^{-i\omega(h-1)} e^{-i\omega}) \\ &= (2\pi)^{-1} \sum_{h=-\infty}^{\infty} (\gamma_{xx}(h) e^{-i\omega h}) \end{aligned}$$

which becomes

$$f_{xy}(\omega) - 0.8f_{xy}(\omega)e^{-i\omega} = f_{xx}(\omega)$$

However, $e^{-i\omega} = \cos(\omega) - i\sin(\omega)$. Consequently, $f_{xy}(\omega)(1 - 0.8\cos(\omega) + 0.8i\sin(\omega)) = f_{xx}(\omega)$.

Multiplying and dividing the left side by the complex conjugate $(1 - 0.8\cos(\omega) + 0.8i\sin(\omega))$, you obtain the result:

$$f_{xy}(\omega) = (1 - 0.8\cos(\omega) - 0.8i\sin(\omega)) / (1.64 - 1.6\cos(\omega)) f_{xx}(\omega)$$

You then have the cospectrum of X by Y (that of Y by X is the same):

$$c(\omega) = f_{xx}(\omega)(1 - 0.8\cos(\omega)) / (1.64 - 1.6\cos(\omega))$$

You also have the quadrature spectrum of X by Y (that of Y by X is $-q(\omega)$):

$$q(\omega) = f_{xx}(\omega) \{0.8\sin(\omega) / [1.64 - 1.6\cos(\omega)]\}$$

In **Output 10.12**, the cospectrum and quadrature spectrum of Y by X and their estimates from PROC SPECTRA are graphed for the case $X_t = 0.5X_{t-1} + e_t$.

10.10.2 Interpretation of Cross-Amplitude and Phase Spectra

The cross-amplitude spectrum is defined as follows:

$$A_{xy}(\omega) = |f_{xy}(\omega)| = (c^2(\omega) + q^2(\omega))^{1/2}$$

In this example, this becomes the following:

$$A_{xy}(\omega) = (1.64 - 1.6\cos(\omega))^{-1/2} f_{xx}(\omega)$$

The gain is defined as the amplitude divided by the spectral density of X , or the following:

$$A_{xy}(\omega) / f_{xx}(\omega)$$

This is assuming $f_{xx}(\omega) \neq 0$. Thus, the gain is the multiplier applied to the sinusoidal component of X at frequency ω to obtain the amplitude of the frequency ω component of Y in a noiseless transfer function, in this case $(1.64 - 1.6\cos(\omega))^{-1/2}$.

The phase spectrum $\Psi_{xy}(\omega)$ of X by Y is defined as follows:

$$\Psi_{xy}(\omega) = \arctan(q(\omega) / c(\omega))$$

The phase spectrum of Y by X is $\arctan(-q(\omega) / c(\omega))$. This is the phase difference between the output and input at frequency ω . In this example,

$$\Psi_{yx}(\omega) = \arctan\{-0.8\sin(\omega) / [1 - 0.8\cos(\omega)]\}$$

These cross-amplitude and phase spectra are graphed with their estimates from PROC SPECTRA in **Output 10.12**. The graphs explain the effect of the transfer function on a sinusoidal input. Its amplitude is changed ($A_{xy}(\omega)$), and it undergoes a phase shift ($\Psi_{xy}(\omega)$). The graphs show how these changes are a function of frequency (ω). The cross-spectrum can be expressed as follows:

$$f_{xy}(\omega) = A_{xy}(\omega) \exp(i\Psi_{xy}(\omega))$$

Transfer function relationships are not perfect (noiseless), so an error series is introduced into the model as follows:

$$Y_t = \sum_{j=-\infty}^{\infty} v_j X_{t-j} + \eta_t$$

Here, η_t is uncorrelated with X_t . Now, analogous to the squared correlation coefficient, the squared coherency is defined as the following:

$$K_{xy}^2(\omega) = |f_{xy}(\omega)|^2 / (f_{xx}(\omega)f_{yy}(\omega))$$

This measures the strength of the relationship between X and Y as a function of frequency. The spectrum $f_\eta(\omega)$ of η_t satisfies the following expression, analogous to regression in which the error sum of squares is the total sum of squares minus the regression sum of squares which can also be expressed as the total sum of squares multiplied by $(1 - R^2)$:

$$f_\eta(\omega) = f_{yy}(\omega) - f_{xy}(\omega)f_{xx}^{-1}(\omega)f_{xy}(\omega) = f_{yy}(\omega)(1 - K_{xy}^2(\omega))$$

To compute the theoretical coherency for the example, you need assumptions on X and η . Assume the following equation with $\text{var}(e_t) = 1$, and assume that η_t is white noise with variance 1:

$$X_t = 0.5X_{t-1} + e_t$$

Then, you have the following result:

$$K_{xy}^2(\omega) = \left\{ 1 + [1.64 - 1.6 \cos(\omega)] / [1.25 - \cos(\omega)] \right\}^{-1}$$

The true squared coherency and its estimate from PROC SPECTRA are graphed in **Output 10.12**.

10.10.3 PROC SPECTRA Statements

PROC SPECTRA gives the names shown in Table 10.1 to estimates of the cross-spectral quantities for the first two variables in the VAR list:

Table 10.1: Estimate Names of the Cross-Spectral Quantities

Estimate Label	Name
cospectrum	CS_01_02
quadrature spectrum	QS_01_02
cross-amplitude spectrum	A_01_02
phase spectrum	PH_01_02
squared coherency	K_01_02

PROC SPECTRA options for cross-spectral analysis are illustrated here:

```
proc spectra data=in out=o1 coef p s
  cross a k ph whitetest adjmean;
  var y1 y2;
  weights 1 1 1 1 1 1;
run;
```

The CROSS option indicates that cross-spectral analysis is to be done. It produces the cospectrum C_01_02 and the quadrature spectrum Q_01_02 when used in conjunction with the S option. CROSS produces the real part RP_01_02 and the imaginary part IP_01_02 of the cross-periodogram when used in conjunction with the P option. Thus, RP and IP are unweighted estimates, and C and Q are weighted and normalized estimates of the cospectrum and quadrature spectrum. The A, K, and PH options request, respectively, estimation of cross-amplitude, squared coherency, and phase spectra (CROSS must be specified also). Weighting is necessary to obtain a valid estimate of the squared coherency.

Consider the following 512 observations Y_t generated from the following model (the noiseless transfer function):

$$V_t = 0.8V_{t-1} + X_t$$

And, adding a noise term, you have the following:

$$Y_t = V_t + \eta_t$$

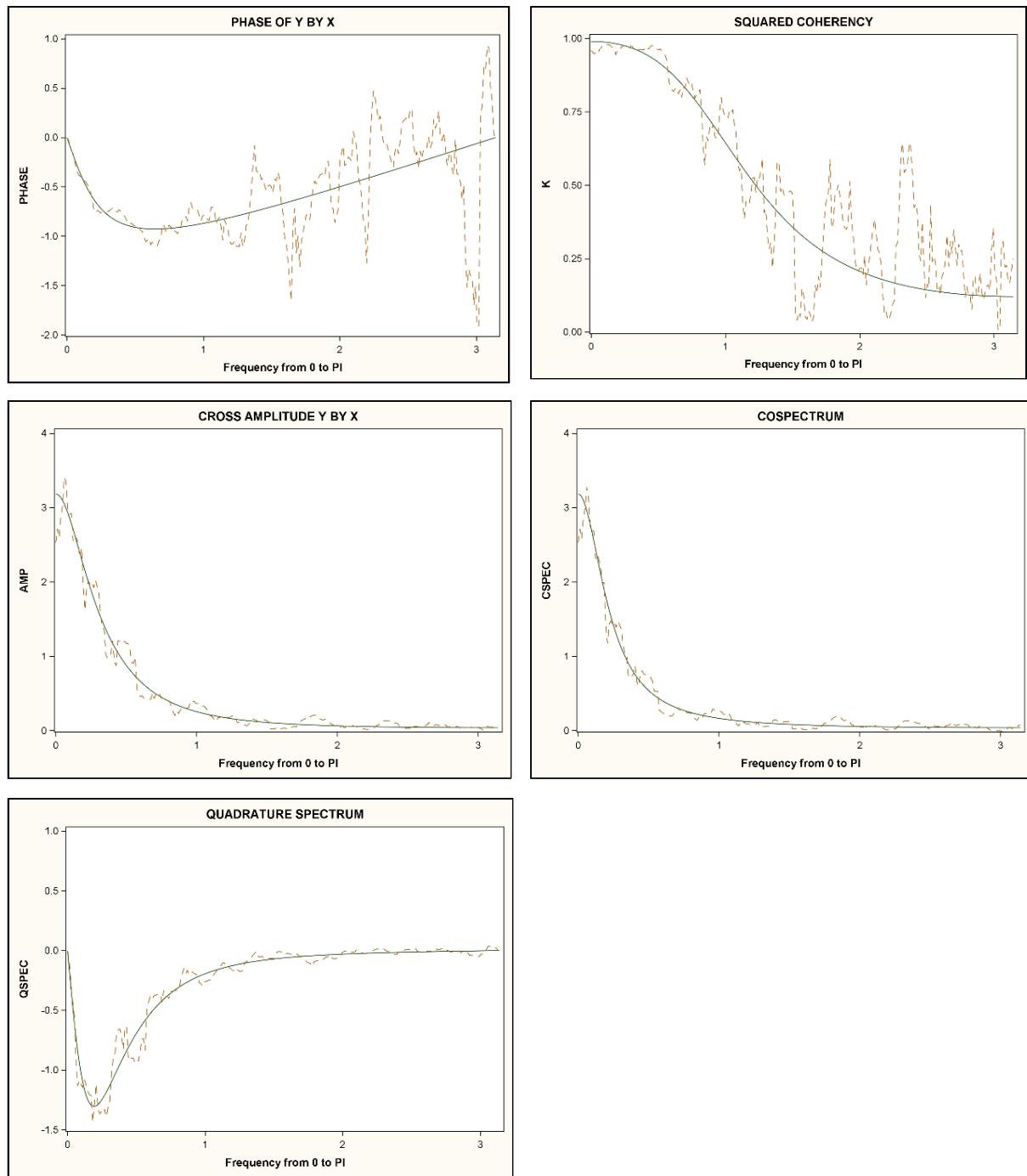
Here, X_t is an autoregressive (AR) process with variance 1.3333 and η_t is white noise with variance 1:

$$X_t = 0.5X_{t-1} + e_t$$

The following SAS code produces appropriate spectral estimates:

```
proc spectra data=a out=ooo p s cross a k ph;
  weights 1 1 1 1 1 1 1 1 1 1 1 1;
  var y x;
run;
```

Plots of estimated and true spectra are overlaid in **Output 10.12**.

Output 10.12: Plots of Estimated and True Spectra

Although the data are artificial, think of X and Y as representing furnace and room temperatures in a building. The phase spectrum shows that long-term fluctuations (ω near zero) and short-term fluctuations (ω near π) for furnace and room temperatures are nearly in phase. The phase spectrum starts at zero and then decreases, indicating that X (the furnace temperature) tends to peak slightly before room temperature at intermediate frequencies. This makes sense if the furnace is connected to the room by a reasonably long pipe.

The squared coherency is near one at low frequencies, indicating a strong correlation between room temperature and furnace temperature at low frequencies. The squared coherency becomes smaller at the higher frequencies in this example. The estimated phase spectrum can vary at high frequencies as a result of this low correlation between furnace and room temperatures at high frequencies. Because of mixing as the air travels from the furnace to the room, high-frequency oscillations in furnace temperatures tend not to be strongly associated with temperature fluctuations in the room.

The gain, A_01_02 / S_02, behaves like the cross-amplitude spectrum A_01_02 for this example. This behavior shows that low-frequency fluctuations in the furnace produce high-amplitude fluctuations at room temperature, and high-frequency fluctuations produce low-amplitude (small variance) fluctuations at room temperature. The transfer function tends to smooth the high-frequency fluctuations. Because of mixing in the pipe leading from the furnace to the room, it is not surprising that high-frequency (fast oscillation) temperature changes in the furnace are not transferred to the room.

10.10.4 Cross-Spectral Analysis of the Neuse River Data

In Chapter 4, “The ARIMA Model: Introductory Applications,” the differenced log flow rates of the Neuse River at Kinston (Y) and Goldsboro (X) are analyzed with the following transfer function model shown in two equivalent representations:

$$(1 - 1.241B + 0.291B^2 + 0.117B^3)X_t = (1 - 0.874B)e_t$$

or

$$X_t - 1.241X_{t-1} + 0.291X_{t-2} + 0.117X_{t-3} = e_t - 0.874e_{t-1}$$

This assumes $\sigma_e^2 = 0.0399$ and $Y_t = 0.495X_{t-1} + 0.273X_{t-2} + \varepsilon_t$.

Here, $\varepsilon_t = 1.163\varepsilon_{t-1} - 0.48\varepsilon_{t-2} + v_t - 0.888v_{t-1}$, and v_t is a white noise series with $\sigma_v^2 = 0.0058$.

You compute the spectral quantities and plot them by using the estimated model parameters. First, the model-based spectral quantities are developed. Then, the direct estimates (no model) of the spectral quantities from PROC SPECTRA are plotted.

When the previous models are used, the spectrum of Goldsboro is written as follows:

$$\begin{aligned} f_{xx}(\omega) &= \left((1 - 0.0874e^{i\omega})(1 - 0.874e^{-i\omega}) \right) / \left((1 - 1.241e^{i\omega} + 0.291e^{2i\omega} + 0.117e^{3i\omega}) \right. \\ &\quad \left. (1 - 1.241e^{-i\omega} + 0.291e^{-2i\omega} + 0.117e^{-3i\omega}) \right) (0.0399 / (2\pi)) \\ &= [1 + 0.874^2 - 2(0.874)\cos(\omega)] / \{[1 + 1.241^2 + 0.291^2 + 0.117^2] \\ &\quad - 2\cos(\omega)[1.241 + 1.241(0.291) - 0.291(0.117)] - 2\cos(2\omega)[1.241(0.117) - 0.291] \\ &\quad + 2\cos(3\omega)[0.117]\} [0.0399 / (2\pi)] \end{aligned}$$

The cross-covariance of Y_t with X_{t-j} is the same as the cross-covariance of X_{t-j} with $0.495X_{t-1} + 0.273X_{t-2}$, so you obtain the following as a result:

$$\gamma_{xy}(j) = 0.495\gamma_{xx}(j-1) + 0.237\gamma_{xx}(j-2)$$

Thus, the cross-spectrum is as follows:

$$f_{xy}(\omega) = (0.495e^{-i\omega} + 0.273e^{-2i\omega})f_{xx}(\omega)$$

The real part (cospectrum) is this:

$$c(\omega) = (0.495\cos(\omega) + 0.273\cos(2\omega))f_{xx}(\omega)$$

And the quadrature spectrum is the following:

$$q(\omega) = (0.495\sin(\omega) + 0.273\sin(2\omega))f_{xx}(\omega)$$

The phase spectrum is expressed like so:

$$\Psi_{xy}(\omega) = \arctan(q(\omega)/c(\omega))$$

The spectrum of Kinston (Y) is written as follows:

$$f_{yy}(\omega)(0.495e^{i\omega} + 0.273e^{2i\omega})(0.495e^{-i\omega} + 0.273e^{-2i\omega})f_{xx}(\omega) + f_{ee}(\omega)$$

where

$$f_{ee}(\omega) = \left(\frac{(1 - 0.888e^{i\omega})(1 - 0.888e^{-i\omega})}{((1 - 1.163e^{i\omega} + 0.48e^{2i\omega})(1 - 1.163e^{-i\omega} + 0.48e^{-2i\omega}))} \right) \left(\frac{0.0058}{2\pi} \right)$$

The squared coherency is written simply as follows:

$$K_{xy}^2(\omega) = |f_{xy}(\omega)|^2 / (f_{xx}(\omega)f_{yy}(\omega))$$

Consider this pure delay transfer function model:

$$Y_t = \beta X_{t-c}$$

Using the Fourier transform, you can show the following relationship:

$$f_{xy}(\omega) = \sum_{h=-\infty}^{\infty} (\gamma_{xy}(h)e^{i\omega h}) = \beta \sum_{h=-\infty}^{\infty} (\gamma_{xx}(h-c)e^{i\omega(h-c)}e^{ic\omega})$$

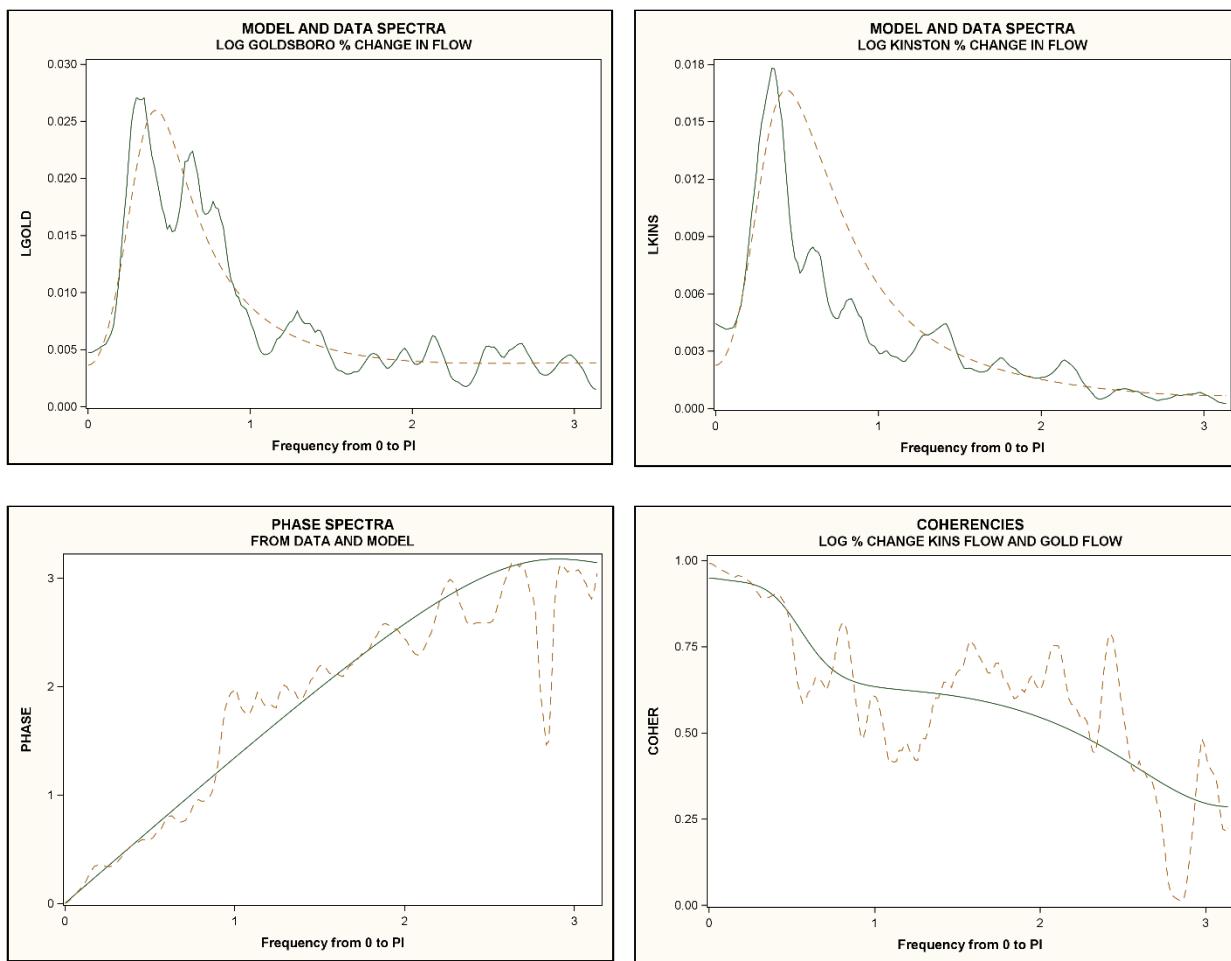
so

$$f_{xy}(\omega) = f_{xx}(\omega)\beta(\cos(c\omega) + i\sin(c\omega))$$

Thus, the phase spectrum is $\Psi_{xy}(\omega) = \arctan(\tan(c\omega)) = c\omega$.

When you use the ordinates in the plot of the phase spectrum as dependent variable values and frequency as the independent variable, a simple linear regression using a few low frequencies gives 1.34 as an estimate of c . This indicates a lag of 1.34 days between Goldsboro and Kinston. Because ARIMA models contain only integer lags, this information appears as two spikes, at lags 1 and 2, in the prewhitened cross-correlations. However, with the cross-spectral approach, you are not restricted to integer lags. In **Output 10.13**, the irregular plots are the cross-spectral estimates from PROC SPECTRA. These are overlaid on the (smooth) plots computed from the transfer function fitted by PROC ARIMA.

Output 10.13: Overlaying the Smoothed Model-Derived Plots from PROC ARIMA and the Irregular PROC SPECTRA Plots



From one viewpoint, the closeness of the PROC SPECTRA plots to the model-derived plots provides a check on the ARIMA transfer function model and estimates. From another viewpoint, the model-based spectral plots provide a highly smoothed version of the PROC SPECTRA output.

10.10.5 Details on Gain, Phase, and Pure Delay

Suppose X_t is a perfect sine wave $X_t = \sin(\omega t + \delta)$. Now, suppose $Y_t = 3X_{t-1} = 3\sin(\omega t - \omega + \delta)$. Y is also a perfect sine wave. The phase $-\omega + \delta$ of Y is ω radians less than the phase of X , and the amplitude of Y is 3 times that of X . You could say that the phase of X is ω radians more than the phase of Y and the amplitude of X is 1/3 that of Y . The idea of cross-spectral analysis is to think of a general pair of series X and Y as each being composed of sinusoidal terms. Then, you estimate how the sinusoidal components of Y are related in terms of amplitude and phase to those of the corresponding sinusoidal component of X .

With two series, Y and X , there is a phase of Y by X and a phase of X by Y . If $Y_t = 3X_{t-1}$, then Y is behind X by 1 time unit. That is, the value of X at time t is a perfect predictor of Y at time $t+1$. Similarly, X is ahead of Y by 1 time unit. This program creates $X_t = e_t$ and $Y_t = 3X_{t-1}$, so it is an example of a simple noiseless transfer function. With $e_t \sim N(0, 1)$, the spectrum $f_{XX}(\omega)$ of X is $f_{XX}(\omega) = 1 / (2\pi) = 0.1592$ at all frequencies ω . And, Y_t has spectrum $9 / (2\pi) = 1.4324$.

```

data a;
pi = 4*atan(1);
x=0;
do t = 1 to 64;
  y = 3*x; *y is 3 times previous x*;
  x=normal(1827655);
  if t=64 then x=0;
  output;
end;
  
```

```

run;
proc spectra data=a p s cross a k ph out=out1 coeff;
  var x y;
run;
proc print label data=out1;
  where period > 12;
  id period freq;
run;

```

Output 10.14: X and Y Series

Period	Frequency from 0 to PI	Cosine Transform of X	Sine Transform of X	Cosine Transform of Y	Sine Transform of Y	Periodogram of X
64.0000	0.09817	0.16213	-0.09548	0.51212	-0.23739	1.13287
32.0000	0.19635	-0.24721	-0.13649	-0.64748	-0.54628	2.55166
21.3333	0.29452	0.09053	0.26364	0.03031	0.83572	2.48656
16.0000	0.39270	0.37786	-0.15790	1.22856	-0.00383	5.36666
12.8000	0.49087	-0.32669	-0.20429	-0.57543	-1.00251	4.75075

Period	Frequency from 0 to PI	Periodogram of Y	Spectral Density of X	Spectral Density of Y	Real Periodogram of X by Y	Imag Periodogram of X by Y
64.0000	0.09817	10.1959	0.09015	0.81136	3.3823	0.33312
32.0000	0.19635	22.9650	0.20305	1.82749	7.5079	1.49341
21.3333	0.29452	22.3791	0.19787	1.78087	7.1385	2.16543
16.0000	0.39270	48.2999	0.42707	3.84359	14.8744	6.16119
12.8000	0.49087	42.7567	0.37805	3.40247	12.5694	6.71846

Period	Frequency from 0 to PI	Cospectra of X by Y	Quadrature of X by Y	Coherency**2 of X by Y	Amplitude of X by Y	Phase of X by Y
64.0000	0.09817	0.26915	0.02651		1	0.27045
32.0000	0.19635	0.59746	0.11884		1	0.60916
21.3333	0.29452	0.56806	0.17232		1	0.59362
16.0000	0.39270	1.18367	0.49029		1	1.28120
12.8000	0.49087	1.00024	0.53464		1	1.13416

Because no weights were specified, no smoothing has been done. Only a few frequencies are printed out.

At period 64, X has a component:

$$0.16213 \cos(2\pi t / 64) - 0.09548 \sin(2\pi t / 64) = 0.188156 \sin(2\pi t / 64 + 2.10302)$$

Y has a component:

$$0.51212 \cos(2\pi t / 64) - 0.23739 \sin(2\pi t / 64) = 0.564465 \sin(2\pi t / 64 - 2.00486)$$

Here, $0.564465 / 0.188156 = 3$ is the amplitude increase in going from X to Y . The phase shift is $2.10302 - 2.00486 = 0.09817$ radians. Each periodogram ordinate is $(n / 2)$ times the sum of squares of the two coefficients, $(64 / 2)[(0.16213)^2 + (0.09548)^2] = 1.13287$ for X at period 64, for example.

Each Y periodogram ordinate is 3^2 times the corresponding X periodogram ordinate. This exact relationship would not hold if noise were added to Y . Within the class of ARMA models, the periodogram $I_n(\omega)$ divided by $2\pi f(\omega)$ (where the true spectral density of the process is $f(\omega)$) has approximately a chi-square distribution with 2 degrees of freedom, a

distribution with mean 2. This motivates $I_n(\omega) / 4\pi$ as an estimator of $f(\omega)$ for both Y and X . Each spectral density estimator is the corresponding periodogram ordinate divided by 4π . For example, $1.13287 / (4\pi) = 0.0902$ for X at period 64.

In the VAR statement of PROC SPECTRA, the order of variables is $X Y$. This produces the phase of X by Y , not Y by X . The phase $-\omega + \delta$ of Y is ω radians less than the phase δ of X . Thus, the entries in the phase column are exactly the same as the frequencies. The plot of phase by frequency is a straight line with slope 1. This slope gives the pure delay d for $Y_t = CX_{t-d}$, so $d = 1$. If the variables had been listed in the order $Y X$, $-\omega$ would have appeared as the phase spectrum estimate.

The slope of the phase plot near the origin gives some idea of the lag relationship between Y and X in a transfer function model with or without added noise, as long as the coherency there is reasonably strong. The delay does not need to be an integer, as was illustrated with the Neuse River data. The phase plot of the generated data that simulated furnace and room temperatures had a negative slope near the origin. The room temperature Y is related to lagged furnace temperature X . With the variables listed in the order $Y X$, the phase of Y by X is produced, giving the negative slope. If the order had been $X Y$, the plot would be reflected about the phase = 0 horizontal line, and an initial positive slope would have been seen. For the Neuse River data, the sites must have been listed in the order Goldsboro Kinston in PROC SPECTRA because the phase slope is positive and Goldsboro (X) is upstream from Kinston (Y).

If $Y_t = 3X_{t-1}$, and if X_t has an absolutely summable covariance function $\gamma_{XX}(h)$, which is the case in the current example, then Y also has a covariance function:

$$\begin{aligned}\gamma_{YY}(h) &= E\{Y_t Y_{t+h}\} \\ &= 9E\{X_{t-1} X_{t-1+h}\} \\ &= 9\gamma_{XX}(h)\end{aligned}$$

By definition, the theoretical spectral density $f_{XX}(\omega)$ of X is the Fourier transform of the covariance sequence:

$$f_{XX}(\omega) = \frac{1}{2\pi} \sum_{h=-\infty}^{\infty} e^{-i\omega h} \gamma_{XX}(h)$$

Similarly, $f_{YY}(\omega) = 9f_{XX}(\omega)$. The absolute summability assumption ensures the existence of the theoretical spectral densities. The processes have a cross-covariance function:

$$\begin{aligned}\gamma_{XY}(h) &= E\{X_t Y_{t+h}\} \\ &= 3E\{X_t X_{t-1+h}\} \\ &= 3\gamma_{XX}(h-1)\end{aligned}$$

Its Fourier transform is the cross-spectral density of Y by X :

$$\begin{aligned}f_{XY}(\omega) &= \frac{1}{2\pi} \sum_{h=-\infty}^{\infty} e^{-i\omega h} \gamma_{XY}(h) = \frac{3}{2\pi} \sum_{h=-\infty}^{\infty} e^{-i\omega h} e^{-i\omega(h-1)} \gamma_{XX}(h-1) \\ &= \frac{3}{2\pi} \sum_{h=-\infty}^{\infty} [\cos(\omega) - i \sin(\omega)] e^{-i\omega(h-1)} \gamma_{XX}(h-1) \\ &= 3[\cos(\omega) - i \sin(\omega)] f_{XX}(\omega)\end{aligned}$$

Consider the following expression:

$$\frac{1}{2\pi} \sum_{h=-\infty}^{\infty} e^{-i\omega h} \gamma_{XY}(h)$$

If you write it as $c(\omega) - iq(\omega)$, the real part $c(\omega)$ is the cospectrum of X by Y and the coefficient of $-i$ is the quadrature spectrum $q(\omega)$. In this example, $c(\omega) = 3\cos(\omega)f_{XX}(\omega)$ and $q(\omega) = 3\sin(\omega)f_{XX}(\omega)$. For example, at period 32, you find $3\cos(2\pi/32) = 2.9424$ and $3\sin(2\pi/32) = 0.5853$. Multiplying these by the estimated X spectral density gives $(2.9424)(0.20305) = 0.5974$, the estimated cospectrum of X by Y for period 32. Similarly, you get $(0.5853)(0.20305) = 0.1188$, the estimated quadrature spectrum of X by Y on the printout.

The phase and amplitude spectra are transformations of $q(\omega)$ and $c(\omega)$ and are often easier to interpret. The phase of X by Y is $\text{Atan}(q(\omega) / c(\omega)) = \text{Atan}(\sin(\omega) / \cos(\omega)) = \omega$. The phase of Y by X is $-\omega$, as would be expected from the previous discussion of phase diagrams. The phase shows you the lag relationship between the variables. For $f_{XY}(\omega) = 3[\cos(\omega) - i\sin(\omega)]f_{XX}(\omega)$, the amplitude of the frequency ω component is expressed as follows:

$$\sqrt{c^2(\omega) + q^2(\omega)} = A(\omega) = 3\sqrt{\cos^2(\omega) + \sin^2(\omega)}f_{XX}(\omega) = 3f_{XX}(\omega)$$

This is called the *amplitude* of X by Y , and in the printout, each of these entries is the corresponding spectral density of the X estimate multiplied by 3. The quantity $A^2(\omega) / f_{XX}(\omega)$ is the spectral density for that part of Y that is exactly related to X , without any added noise. Because Y is related to X by a noiseless transfer function, the spectral density of Y should be $A^2(\omega) / f_{XX}(\omega)$. For example, at period 32, you find $(0.60916)^2 / (0.20305) = 1.82749$. Recall that the quantity $A(\omega) / f_{XX}(\omega)$ has been referred to earlier as the *gain*. It represents the amplitude multiplier for the frequency ω component in going from X to Y in a model where Y is related to X without noise. In this case, the gain is thus:

$$3\sqrt{\cos^2(\omega) + \sin^2(\omega)} = 3$$

A more realistic scenario is that an observed series W_t consists of Y_t plus an added noise component N_t independent of X (and thus, Y). The phase, amplitude, and gain using W and X as data have their same interpretation. But, they refer to relationships between X and Y —that is, between X and the part of W that is a direct transfer function of X . You can think of fluctuations in X over time as providing energy that is transferred into Y , such as vibrations in an airplane engine transferred to the wing or fuselage. The fluctuations in that object consist of the transferred energy plus independent fluctuations such as wind movements while flying. The spectral density $f_{WW}(\omega)$ of W will no longer be $A^2(\omega) / f_{XX}(\omega)$. It will be this plus the noise spectrum. In a system with noise, the quantity $A^2(\omega) / [f_{XX}(\omega)f_{WW}(\omega)]$ provides an R^2 measure as a function of frequency. Its symbol is $\kappa^2(\omega)$, and it is called the *squared coherency*. In a noiseless transfer function, such as $Y_t = 3X_{t-1}$, the squared coherency between Y and X would be 1 at all frequencies because $[A^2(\omega) / f_{XX}(\omega)] / f_{YY}(\omega) = f_{YY}(\omega) / f_{YY}(\omega) = 1$ in that case. This appears in the output. However, if there are no smoothing weights, the squared coherency is really meaningless, as would be an R^2 of 1 in a simple linear regression with only 2 points.

This small example without smoothing is presented to show and interpret the cross-spectral calculations. In practice, smoothing weights are usually applied so that more accurate estimates can be obtained. Another practical problem arises with the phase. The phase is usually computed as the angle in $[-\pi/2, \pi/2]$ whose tangent is $q(\omega) / c(\omega)$. If a phase angle a little less than $\pi/2$ is followed by one just a bit bigger than $\pi/2$, the interval restriction will cause this second angle to be reported as an angle just a little bigger than $-\pi/2$. The phase diagram can show phases jumping back and forth between $-\pi/2$ and $\pi/2$ when, in fact, they could be represented as not changing much at all. Some practitioners choose to add and subtract multiples of π from the phase at selected frequencies in order to avoid excessive fluctuations in the plot.

Fuller (1996) gives formulas for the cross-spectral estimates and confidence intervals for these quantities in the case that there are $2d + 1$ equal smoothing weights.

References

- Akaike, H. 1974. "Markovian Representation of Stochastic Processes and Its Application to the Analysis of Autoregressive Moving Average Processes." *Annals of the Institute of Statistical Mathematics*, 26: 363–387.
- Akaike, H. 1976. "Canonical Correlations Analysis of Time Series and the Use of an Information Criterion." In *Advances and Case Studies in System Identification*, eds. R. Mehra and D. G. Lainiotis. New York: Academic Press.
- Akdi, Y., and D. A. Dickey. 1997. "Periodograms of Unit Root Time Series: Distributions and Tests." *Communications in Statistics* 27:69–87.
- Anderson, T. W. 1971. *The Statistical Analysis of Time Series*. New York: Wiley.
- Bailey, C. T. 1984. "Forecasting Industrial Production 1981–1984." *Proceedings of the Ninth Annual SAS Users Group International Conference*, Hollywood Beach, FL, 50–57.
- Bartlett, M. S. 1947. "Multivariate Analysis." *Supplement to the Journal of the Royal Statistical Society*, Series B, IX:176–197.
- Bartlett, M. S. 1966. *An Introduction to Stochastic Processes*. 2d ed. Cambridge: Cambridge University Press.
- Bolerslev, Tim. 1986. "Generalized Autoregressive Conditional Heteroskedasticity." *Journal of Econometrics* 31:307–327.
- Box, G. E. P., and D. R. Cox. 1964. "An Analysis of Transformations." *Journal of the Royal Statistical Society* B26:211.
- Box, G. E. P., and G. M. Jenkins. 1976. *Time Series Analysis: Forecasting and Control*. Rev. ed. Oakland: Holden-Day.
- Box, G. E. P., G. M. Jenkins, and G. C. Reinsel. 1994. *Time Series Analysis: Forecasting and Control*. 3d ed. Englewood Cliffs, NJ: Prentice Hall.
- Brillinger, D. R. 1975. *Time Series: Data Analysis and Theory*. New York: Holt, Rinehart & Winston.
- Brocklebank, J., and D. A. Dickey. 1984. *SAS Views: SAS Applied Time Series Analysis and Forecasting*. Cary, NC: SAS Institute Inc.
- Chang, M. C., and D. A. Dickey. 1993. "Recognizing Overdifferenced Time Series." *Journal of Time Series Analysis* 15:1–8.
- Chavern, J. 1984. "On the Limitations of Akaike's Information Criterion and Its Use in PROC STATESPACE." *Proceedings of the Ninth Annual SAS Users Group International Conference*, Hollywood Beach, FL, 106–111.
- Cohen, H., ed. 1981. *Metal Statistics*. New York: Fairchild Publications.
- Croston, J. D. 1977. "Forecasting and Stock Control for Intermittent Demands." *Operations Research Quarterly* 23, no. 3.
- Davis, H. T. 1941. *The Analysis of Economic Time Series*. Chicago: Principia Press.
- De Jong, P. 1991. "The Diffuse Kalman Filter." *Annals of Statistics*, 19(2):1073–1083.
- 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*, 427–431.
- Dickey, D. A., and W. A. Fuller. 1981. "Likelihood Ratio Statistics for Autoregressive Time Series with a Unit Root." *Econometrica* 49:1057–1072.
- Dickey, D. A., D. P. Hasza, and W. A. Fuller. 1984. "Testing for Unit Roots in Seasonal Time Series." *Journal of the American Statistical Association* 79:355–367.
- Dickey, D. A., D. W. Janssen, and D. L. Thornton. 1991. "A Primer on Cointegration with an Application to Money and Income." *Review of the Federal Reserve Bank of St. Louis* 73:58–78.
- Dickey, D. A., W. R. Bell, and R. B. Miller. 1986. "Unit Roots in Time Series Models: Tests and Implications." *American Statistician* 40:12–26.
- Draper, N., and H. Smith. 1998. *Applied Regression Analysis*. 3d ed. New York: Wiley.

- Durbin, J. 1960. "The Fitting of Time Series Models." *International Statistical Review* 28:233–244.
- Durbin, J., and S. J. Koopman. 2012. *Time Series Analysis by State Space Methods*, 2nd ed. Oxford University Press, Oxford, UK: Oxford University Press.
- Engle, R. F., and C. W. J. Granger. 1987. "Cointegration and Error Correction: Representation, Estimation, and Testing." *Econometrica* 55:251–276.
- Engle, Robert. 1982. "Autoregressive Conditional Heteroskedasticity with Estimates of the Variance of United Kingdom Inflation." *Econometrica* 50:987–1007.
- Evans, B. 1998. "Estimation and Hypothesis Testing in Nonstationary Time Series Using Frequency Domain Methods." Ph.D. diss., North Carolina State University.
- Fountis, N. G., and D. A. Dickey. 1989. "Testing for a Unit Root Nonstationarity in Multivariate Autoregressive Time Series." *Annals of Statistics* 17:419–428.
- Fuller, W. A. 1976. *Introduction to Statistical Time Series*. New York: Wiley.
- Fuller, W. A. 1986. "Using PROC NLIN for Time Series Prediction." *Proceedings of the Eleventh Annual SAS Users Group International Conference*, Atlanta, GA, 63–68.
- Fuller, W. A. 1996. *Introduction to Statistical Time Series*. 2d ed. New York: Wiley.
- Hall, A. 1992. "Testing for a Unit Root in Time Series with Data-Based Model Selection." *Journal of Business and Economic Statistics* 12:461–470.
- Hamilton, J. D. 1994. *Time Series Analysis*. Princeton, NJ: Princeton University Press.
- Hannan, E. J., and J. Rissanen. 1982. "Recursive Estimation of Mixed Autoregressive-Moving Average Order." *Biometrika* 69, no. 1 (April): 81–94.
- Harvey, A. C. 1981. *Time Series Models*. Oxford: Philip Allan Publishers.
- Jarque, C. M., and A. K. Bera. 1980. "Efficient Tests for Normality, Homoskedasticity and Serial Independence of Regression Residuals." *Economics Letters* 6:255–259.
- Jenkins, G. M., and D. G. Watts. 1968. *Spectral Analysis and Its Applications*. Oakland: Holden-Day.
- Johansen, S. 1988. "Statistical Analysis of Cointegrating Vectors." *Journal of Economic Dynamics and Control* 12:312–254.
- Johansen, S. 1991. "Estimation and Hypothesis Testing of Cointegrating Vectors in Gaussian Vector Autoregressive Models." *Econometrica* 59:1551–1580.
- Johansen, S. 1994. "The Role of the Constant and Linear Terms in Cointegration Analysis of Non-Stationary Variables." *Econometric Reviews* 13:205–230.
- Johnston, J. 1972. *Econometric Methods*. 2d ed. New York: McGraw-Hill.
- Jones, R. H. 1974. "Identification and Autoregressive Spectrum Estimation." *IEEE Transactions on Automatic Control*, AC-19:894–897.
- Liu, Shiping, Ju-Chin Huang, and Gregory L. Brown. 1988. "Information and Risk Perception: A Dynamic Adjustment Process." *Risk Analysis* 18:689–699.
- Ljung, G. M., and G. E. P. Box. 1978. "On a Measure of Lack of Fit in Time Series Models." *Biometrika* 65:297–303.
- McSweeney, A. J. 1978. "Effects of Response Cost on the Behavior of a Million Persons: Charging for Directory Assistance in Cincinnati." *Journal of Applied Behavior Analysis* 11:47–51.
- Nelson, D. B. 1991. "Conditional Heteroskedasticity in Asset Returns: A New Approach." *Econometrica* 59:347–370.
- Nelson, D. B., and C. Q. Cao. 1992. "Inequality Constraints in the Univariate GARCH Model." *Journal of Business and Economic Statistics* 10:229–235.
- Pham, D. T. 1978. "On the Fitting of Multivariate Process of the Autoregressive-Moving Average Type." *Biometrika* 65:99–107.
- Priestley, M. B. 1980. "System Identification, Kalman Filtering, and Stochastic Control." In *Directions in Time Series*, ed. D. R. Brillinger and G. C. Tiao. Hayward CA: Institute of Mathematical Statistics.
- Priestley, M. B. 1981. *Spectra Analysis and Time Series. Volume 1: Univariate Series*. New York: Academic Press.
- Robinson, P. M. 1973. "Generalized Canonical Analysis for Time Series." *Journal of Multivariate Analysis* 3:141–160.
- Said, S. E., and D. A. Dickey. 1984. "Testing for Unit Roots in Autoregressive Moving Average Models of Unknown Order." *Biometrika* 71, no. 3: 599–607.

- SAS Institute Inc. 1985. *SAS Introductory Guide*. 3d ed. Cary, NC: SAS Institute Inc.
- SAS Institute Inc. 2016. *Base SAS 9.4 Procedures Guide: Statistical Procedures*, 5th ed. Cary, NC: SAS Institute Inc.
- SAS Institute Inc. 2016. *SAS/ETS 14.2 User's Guide*. Cary, NC: SAS Institute Inc.
- SAS Institute Inc. 2016. *SAS/GRAF 9.4 Reference*. Cary, NC: SAS Institute Inc.
- SAS Institute Inc. 2017. *SAS/STAT 14.3 User's Guide*. Cary, NC: SAS Institute Inc.
- Self, S. G., and K-Y Liang. 1987. "Asymptotic Properties of Maximum Likelihood Estimators and Likelihood Ratio Tests under Nonstandard Conditions." *Journal of the American Statistical Association*, 82(392):605–610.
- Singleton, R. C. 1969. "An Algorithm for Computing the Mixed Radix Fast Fourier Transform." *IEEE Transactions of Audio and Electroacoustics*, AU-17:93–103.
- Stock, J. H., and W. W. Watson. 1988. "Testing for Common Trends." *Journal of the American Statistical Association* 83:1097–1107.
- The World Bank Group. 2018. United States Data. <http://data.worldbank.org/country/united-states>.
- Tsay, Ruey S., and George C. Tiao. 1984. "Consistent Estimates of Autoregressive Parameters and Extended Sample Autocorrelation Function for Stationary and Nonstationary ARMA Models." *Journal of the American Statistical Association* 79, no. 385 (March): 84–96.
- Tsay, Ruey S., and George C. Tiao. 1985. "Use of Canonical Analysis in Time Series Model Identification." *Biometrika* 72, no. 2 (August): 299–315.
- US Bureau of Census. 1982. "Construction Workers in Thousands." *Construction Review*.
- US Census Bureau. N.D. Table 10a. Quarterly Estimates of the Total Housing Inventory for the United States by Regions. https://www.census.gov/housing/hvs/data/hist_tab10a_v2016.xlsx.
- US Department of Labor. 1977. "Publishing and Printing Nonproduction Workers 1944–1977." *Handbook of Labor Statistics*.
- Whittle, P. 1963. "On the Fitting of Multivariate Autoregressions and the Approximate Canonical Factorization of a Spectral Density Matrix." *Biometrika* 50:129–134.

Index

A

ACF (autocorrelation function) 47, 48–50, 52–54
Additive model 285
ADF (Augmented Dickey-Fuller) 84
ADJMEAN option 336
Airline Passengers example 296–299
aliasing 342
ALTPARM option 140
amplitude 333
analysis methods 2–5
See also spectral analysis
AR models
 extensions of 266–267
 fitting in REG procedure 37–40
ARCH procedure 184–189
ARIMA model
 about 25, 41
 cointegration 189–208
 example and instructions 56–104
 methodology and example 122–161
 model identification 46–56
 models with explanatory variables 119–121
 predictions 42–46
 regression with time series errors and unequal variances
 177–189
 seasonal time series 107–120
 statistical background 41
 steps for analyzing nonseasonal univariate series 104–105
 terminology and notation 41–42
ARIMA procedure 2–5, 25–32, 35, 63, 70–71, 89, 254
See also ARIMA model
Atlantic Ocean Tides example 211–216
Augmented Dickey-Fuller (ADF tests) 84
augmenting lags 84
autocorrelation 1
autocorrelation function (ACF) 47, 48–50, 52–54
autocovariance function 47–48
AUTOMDL statement 297
AUTOREG procedure 2, 3, 5, 8, 13–15, 120, 122, 128, 170, 177–178, 179–182, 185, 187, 310
autoregression
 about 23
 fitting AR models in REG procedure 37–40
 forecasting 24–37
 statistical background 23–24
 terminology and notation 23
autoregressive errors 177–178

B

BACK option 236, 239–240
backshift notation B , for time series 32–33
BACKSTEP option 180
bandwidth 346
basic structural model 244, 247–249
bivariate examples 273–274
boundary values, in linear exponential smoothing 222–228
Box-Cox transformation 295
Box-Pierce Q statistic 56
BY statement 219

C

CANCORR procedure 200–201
CENTER option 57
centered moving average filters 288
chi-square check of residuals 56
CLS (conditional least squares) method 29, 43, 70
Cochrane-Orcutt method 178
COEFF option 336
COINTEG statement 202
cointegrated variables 128
cointegrating vector 197, 199–201
cointegration
 about 3, 189–191
 diagnostics 206–208
 eigenvalues and 191–192
 estimation of cointegrating vector 199–201
 example 196–199
 forecasts 206–208
 impulse response function 192
 intercepts 201–202
 interpretation of estimates 205
 lags 201–202
 roots in higher-order models 192–194
 unit roots and 194–196
 VARMAX procedure 202–204
COINTTEST option 202
common trend 198
COMPONENT statement 266, 268, 270
concentrated likelihood 29
conditional least squares (CLS) method 29, 43, 70
conditional sum of squares 43
Construction Series example 168–171
cosine terms 258–259
cospectrum 347
COV option 278, 281
covariances, Yule-Walker equations for 33–37
CROSSCOR= option 123, 135–136
cross-spectral analysis 346–347

cumulative periodogram test 337
 curvature, accommodation for 267–270
 cycle 258
 CYCLE statement 262, 263

D

damped trend exponential smoothing 228–229
 damping factor 261
 data
 nonstationary 84–90
 training 236
 transformed 17–22
 trending 216–232
 DATA option 57
 DDOW statement 185
 decomposition 257–258
 DELETE statement 120
 deviations form 201
 diagnostic plots 229–231
 Dickey-Fuller distributions 90
 differencing
 effect of on forecasts 78–79
 to remove linear trends 91–95
 diffuse likelihood 250–254, 251–252
 diffuse log likelihood 251–252
 disturbance vector 276
 double exponential smoothing 216–217
 drift 84, 95
 Durbin-Watson (DW) statistic 7–8, 119–120, 206

E

ECM (error correction model) 194, 197
 eigenvalues
 about 98
 cointegration and 191–192
 Employment in the United States example 299–303
 Energy Demand at a University example 178–182
 Engle, Robert F. 184
 error 177–178, 197
 error correction 197
 error correction model (ECM) 194, 197
 ESACF procedure 95, 101–103
 ESM procedure 2, 3, 5, 212, 215, 217–222, 227–229, 231–232, 236, 239, 240
 estimate inverse autocorrelation function (IACF) 47, 55–56
 ESTIMATE statement 28, 30, 57, 67, 72–78, 105, 123, 126, 135–136, 150, 153
 estimated autocorrelation function (ACF) 52–54
 estimated partial autocorrelation function (PACF) 54–55
 estimation
 in ARIMA procedure 70–71
 of cointegrating vector 199–201
 interpretation of 205
 of spectrum 345–346
 EVAL statement 282

examples

Airline Passengers 296–299
 ARIMA model 56–104, 122–161
 Atlantic Ocean Tides 211–216
 bivariate 273–274
 cointegration 196–199
 Construction Series 168–171
 Employment in the United States 299–303
 Energy Demand at a University 178–182
 Iron and Steel Export Analysis 65–70
 Milk Scare 172–175
 Multi-Series 217–219
 Neuse River Data 352–354
 North Carolina Retail Sales 161–168
 Plant Enzyme Activity 334–335
 real data 219–222
 Terrorist Attack 175–176
 Tourism 254–257
 Trigonometric Components 259–261
 explanatory variables, models with 119–121
 exponential smoothing
 See also single exponential smoothing
 advantages of 240
 damped trend 228–229
 diagnostic plots 229–231
 diagnostics 236–240
 double 216–217
 how smoothing leads to ARIMA 240–242
 linear 216–217, 222–228
 Multi-Series example 217–219
 real data examples 219–222
 seasonal 232–234
 sums of forecasts 231–232
 for trending data 216–232
 Winters method 234–236

F

F test 85
 feedback 121
 FINAL=ALL option 297
 Fisher's kappa test statistic 337
 fluctuations 342
 folding frequency 342
 FORECAST procedure 2, 3, 5, 84
 FORECAST statement 105, 176, 253, 256
 forecasts/forecasting
 about 24–25
 ARIMA procedure for 25–32
 backshift notation *B* for time series 32–33
 cointegration 206–208
 effect of differencing on 78–79
 IBM series 80–84
 properties of 217
 Silver series 80–84
 with single exponential smoothing 210
 sums of 231–232
 Yule-Walker equations for covariances 33–37

Fourier frequencies 335
 frequency 333
 FUNCTION=AUTO option 297
 future predictions 43–46

G

gain 354–357
 GARCH procedure 184–189
 GLM procedure 6, 37, 91–95, 119–120, 170, 254
 goodness of fit 313–314
 GPATH= option 252
 Granger, Clive W.J. 184
 GRID option 72

H

harmonics 259–261, 338
 hierarchical time series 320–331
 highly regular seasonality 11–17
 holdout-sample period 314
 Holt's method 216–217
 Holt-Winters model 2
 honest assessment 313–314
 hyperparameters 250

I

IACF (inverse autocorrelation function) 47, 55–56
 IBM series 80–84
 identification, techniques for 95–104
See also model identification
 IDENTIFY statement 30, 47, 57–65, 67, 78, 80–84, 89, 123, 128, 135–136, 150
 IGARCH procedure 184–189
 IML procedure 271
 impact matrix 202
 impulse-response function 135, 192
 impulse-response weights 121
 initial condition equation 244
 INPUT= option 123
 in-sample period 314
 instructions, for ARIMA model 56–104
 intercepts 201–202
 intervention analysis 120, 121, 155–161
 inverse autocorrelation function (IACF) 47, 55–56
 invertibility 46
 Iron and Steel Export Analysis example 65–70
 IRREGULAR statement 253, 255, 262, 264, 277

K

Kalman, Rudolph E. 250
 Kalman filter 250–254
 Kolmogorov-Smirnov test 337, 338

L

lags 84, 201–202, 270–273
 leading indicators 121
 LEVEL statement 261, 262
 likelihood factors 177
 linear exponential smoothing 216–217, 222–228
 linear regression 5–11
 linear trends, differencing to remove 91–95
 LISTING statement 252
 Ljung-Box modification 56

M

MAE (mean absolute error) 299
 Markov process 266
 MASE (mean absolute scaled error) 313
 maximum likelihood (ML) method 28, 70, 177
 mean absolute error (MAE) 299
 mean absolute scaled error (MASE) 313
 means axis 197
 MEANS procedure 189
 METHOD-ADDALL option 297
 Milk Scare example 172–175
 MINIC procedure 95, 101–103
 MIXED procedure 250, 251–252, 275
 ML (maximum likelihood) method 28, 70, 177
 MODE option 291
 model identification
 about 46, 133–135
 chi-square check of residuals 56
 invertibility 46
 stationarity 46
 summary of 56
 time series identification 47–56
 MODEL procedure 5
 MODEL statement 7, 13, 17, 120, 269, 277, 278
 models
See also ARIMA model
 Additive model 285
 AR models 37–40, 266–267
 basic structural model 244, 247–249
 choosing numerically 239–240
 choosing visually 237–239
 error correction model (ECM) 194, 197
 with explanatory variables 119–121
 Holt-Winters model 2
 identification of 108–119
 Multiplicative model 171, 285
 nonseasonal unobserved components models (UCM) 243–250
 for nonstationary data 84–90
 regARIMA models 295–296
 state space model (SSM) 243
 unobserved components model (UCM) 254–255
 moving averages 287–290
 moving seasonality 292
 Multiplicative model 171, 285
 Multi-Series example 217–219

N

negative autocorrelation 1
 Neuse River Data example 352–354
 NLAG= option 57
 NLIN procedure 122
 NOCONSTANT option 30, 78, 105, 109
 NOEST option 249, 260
 nonseasonal univariate series, analyzing 104–105
 nonseasonal unobserved components models (UCM) 243–250
 nonstationary data, models for 84–90
 nonstationary series 77–78
 NOPRINT option 57
 NOPRINT statement 67, 123
 normal prior with infinitive variance 250
 NORMALIZE option 202
 North Carolina Retail Sales example 161–168
 NOTSORTED option 218
 Nyquist frequency 338

O

observation equation 244
 one-step-ahead predictions 42–43
 one-way ANOVA 294
 options
See also specific options
 OUTCOV= option 162
 OUTDECOMP statement 257
 outlier detection, in SAS Forecast Studio 314–318
 OUTLIER statement 297
 OUTPUT statement 120, 291

P

PACF (partial autocorrelation function) 47, 50–52, 54–55
 PARMS statement 275–276
 partial autocorrelation function (PACF) 47, 50–52, 54–55
 periodogram 333
 periodogram ordinate 335
 phase angle 333, 354–357
 phase shift 259, 333
 Plant Enzyme Activity example 334–335
 PLOT option 123, 126, 129, 140, 229–231, 233–234, 237, 253, 261, 263
 positive autocorrelation 1
 predictions
 about 42
 future 43–46
 one-step-ahead 42–43
 prewhitening 135
 PRINT option 253, 270
 PRINTALL option 30, 72
 procedures
See also specific procedures

projects

creating in SAS Forecast Studio 305–310
 settings for in SAS Forecast Studio 310–318
 pure delay 121, 354–357
p-values 90

Q

quadratic spectrum 347

R

reconciliation 320–331
 REG procedure 6–8, 13, 15, 17, 37–40, 85–86, 88, 91–95, 100, 119–120, 181, 183, 192–193, 244, 254, 259, 334, 340
 regARIMA models 295–296
 regression
 about 5
 autoregressive errors 177–178
 highly regular seasonality 11–17
 linear 5–11
 with time series errors 120, 122–130, 177–189
 with time series errors and unequal variances 177–189
 with transformed data 17–22
 REGRESSION statement 297
 REPEATED statement 275
 residuals, chi-square check of 56
 root mean squared error (RMSE) 299
 roots
 of the characteristic polynomial 77
 in higher-order models 192–194
 unit 194–196, 247

S

SAS Forecast Studio
 about 305
 creating custom events 318–320
 creating projects 305–310
 hierarchical time series 320–331
 modes in 310–312
 project settings 310–318
 reconciliation 320–331
 SAS/ETS software 2–5
 SBC (Schwarz Bayesian Information Criteria) 100
 SCAN procedure 95, 101–103
 Schwarz Bayesian Information Criteria (SBC) 100
 SEASON statement 254, 260, 262, 265
 seasonal exponential smoothing 232–234
 seasonal modeling 107–108
 seasonal multiplicative moving average 108
 seasonal recursions 254
 seasonal time series
 about 107
 identification of models 108–119
 seasonal modeling 107–108

seasonality
 adjustment for with X13 procedure 285–303
 tests for 292–295
 in unobserved components model 254–255

SELECT statement 293

SGPLOT procedure 211, 252
 shift 121

Silver series 80–84

sine terms 258–259

single exponential smoothing
 about 209
 alternative representations of 210–211
 Atlantic Ocean Tides example 211–216
 forecasting with 210
 idea of 209–210

SLOPE statement 249, 256, 262

smoothed periodogram 345–346

smoothed series 252

smoothing weight 211

span 12 difference 108

SPECTRA procedure 2–3, 5, 105, 119, 262, 335–337, 345–346, 348–354, 356

spectral analysis
 about 333–334
 aliasing 342
 cross-spectral analysis 346–347
 estimation of spectrum 345–346
 fluctuations 342
 harmonic frequencies 338–341
 mathematical detail 345
 Plant Enzyme Activity example 334–335
 smoothed periodogram 345–346
 SPECTRA procedure 335–337
 spectral density 342–344
 tests for white noise 337–338

spectral density 342–344

spectral window 346

squared canonical correlations 98

squared coherency 357

squared gain 344

SSM (state space model) 243

SSM procedure 2, 3–4, 5, 121, 265–283, 275–276, 283

state space model (SSM) 243

STATE statement 268, 269, 275–276

state vector 265

state vector transition equation 243

STATEINFO option 278

statements
See also specific statements

STATESPACE procedure 2, 121

stationarity 36, 46, 192

T

t test 85

TCC (trend-cycle component) 257, 285

Terrorist Attack example 175–176

TEST statement 85

tests
 Augmented Dickey-Fuller (ADF tests) 84
 cumulative periodogram test 337
F test 85
 Kolmogorov-Smirnov test 337, 338
 for seasonality 292–295
t test 85
 for white noise 337–338

time series
See also specific topics
 about 1
 analysis methods 2–5
 backshift notation B for 32–33
 hierarchical 320–331
 identification of 47–56
 regression 5–22, 120, 122–130, 177–189
 regression with errors 120, 122–130, 177–189
 SAS/ETS software 2–5
 simple models 5–22

TIMESERIES procedure 2, 257–258, 285

Tourism example 254–257

training data 236

TRAMO 295–296

transfer function weights 121

transfer functions 121, 131–155

TRANSFORM statement 297

transformation, in SAS Forecast Studio 314–318

transformed data, regression with 17–22

transition equation 265–266

transition input vector 277

transition matrix 265

TREND statement 272

trend-cycle component (TCC) 257, 285

trending data, exponential smoothing for 216–232

Trigonometric Components example 259–261

U

UCM (unobserved components model) 254–255

UCM procedure 2, 3–4, 246–247, 253, 254, 256, 258, 262, 263, 265, 283

unconditional least squares (ULS) 28, 70

unconditional sum of squares (USS) 29

unequal variances, regression with 177–189

unit roots 194–196, 247

UNIVARIATE procedure 105, 337

unobserved components model (UCM) 254–255

USS (unconditional sum of squares) 29

V

validation 236

VAR= option 57

VARMAX procedure 2, 3, 5, 121, 128, 189, 200, 202–204

W

WEIGHT statement 346
 white noise 337–338
 WHITETEST option 338
 Winters method 234–236
 %WLDBNK macro 223, 229–231
 Wold representation, of the series 23–24
 "Wolfer sunspot" 261

X

X-11 method
 about 287
 basic seasonal adjustment using 291–292
 moving averages 287–290
 outline of 290–291
 tests for seasonality 292–295
 X13 procedure
 about 2, 5, 285–286
 adjustment for seasonality with 285–303
 data examples 296–303
 regARIMA models 295–296
 X-11 method 287–295

Y

Yule-Walker equations 33–37, 100

Ready to take your SAS® and JMP® skills up a notch?



Be among the first to know about new books,
special events, and exclusive discounts.

support.sas.com/newbooks

Share your expertise. Write a book with SAS.

support.sas.com/publish

 sas.com/books
for additional books and resources.


THE POWER TO KNOW®