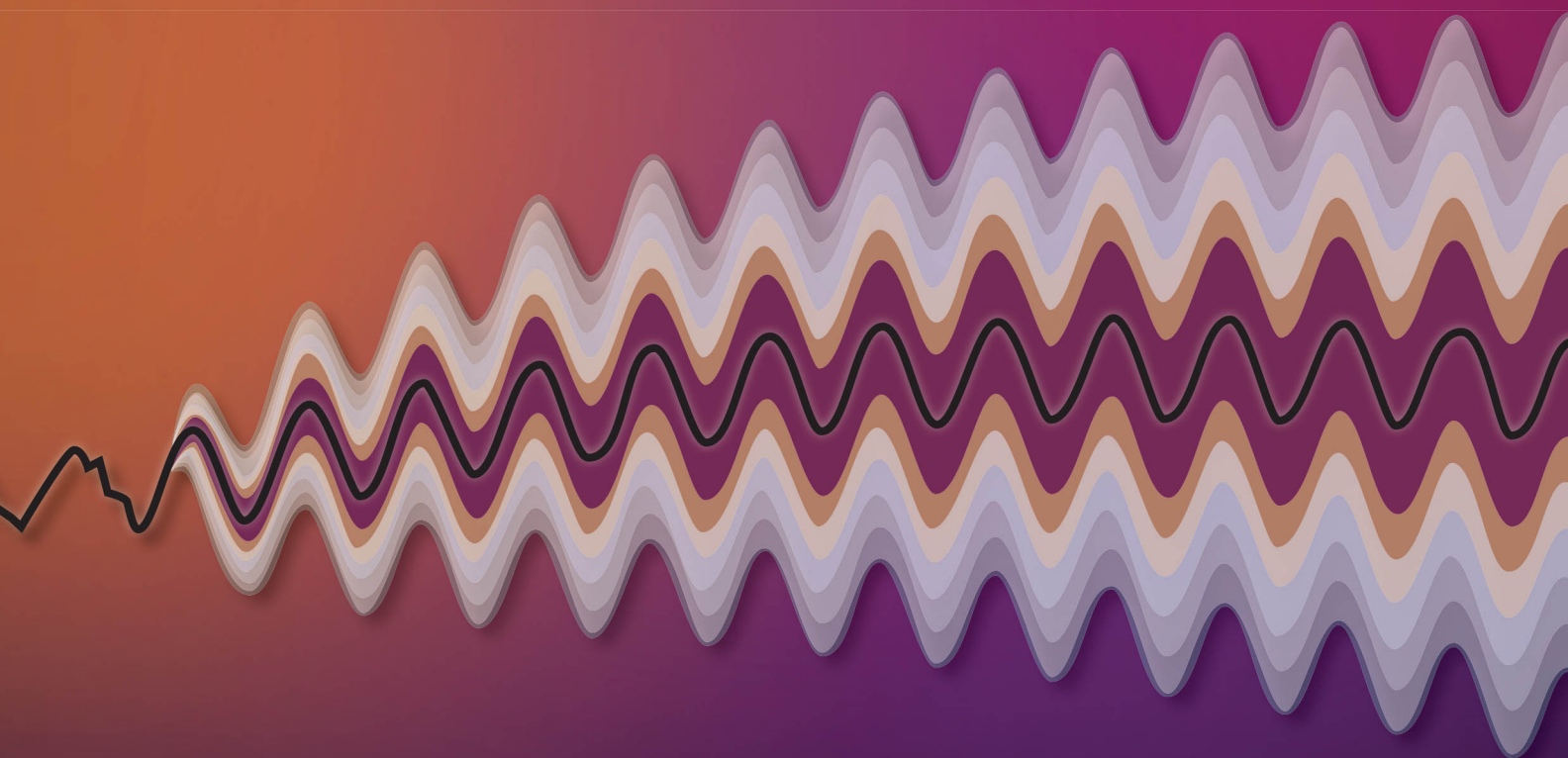


SAS[®] for Forecasting Time Series

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.

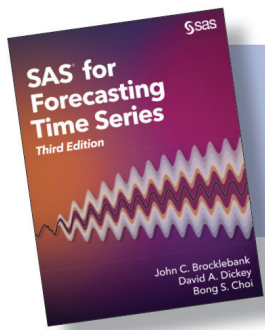
SAS Institute Inc., SAS Campus Drive, Cary, NC 27513-2414

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>.



SAS® for Forecasting Time Series, Third Edition. Full book available for purchase [here](#).

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/GRAPH 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 the author pages at <http://support.sas.com/brocklebank>, <http://support.sas.com/dickey>, or <http://support.sas.com/choi>.

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

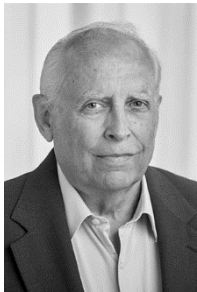
Do you have questions about a SAS Press book that you are reading? Contact the author through saspress@sas.com or https://support.sas.com/author_feedback.

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.



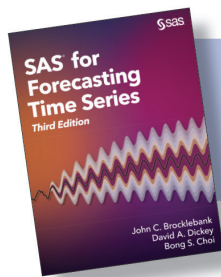
Bong S. Choi, PhD, is a Senior Associate Analytical Consultant at SAS. He has worked on projects across a variety of industries, including health care, retail, and banking. A SAS certified advanced programmer, he has been a SAS user since 2011. Dr. Choi holds an MS in applied statistics from the University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



SAS® for Forecasting Time Series, Third Edition. Full book available for purchase [here](#).

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 $\omega \sum_{j=1}^{\infty} (1-\omega)^j Y_{n-j}$, 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:

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

The last term approaches 0 with increasing n . Ignoring it and letting n increase results in an infinite sum. Rearranging gives $Y_t = ((1-\theta)Y_{t-1} + \theta(1-\theta)Y_{t-2} + \theta^2(1-\theta)Y_{t-3} + \dots) + e_t$, 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

$$\hat{Y}_{t+2} = (1-\theta) \left(Y_{t+1} + \theta \sum_{j=0}^{\infty} \theta^j Y_{t-j} \right) = (1-\theta) Y_{t+1} + \theta \hat{Y}_{t+1}$$

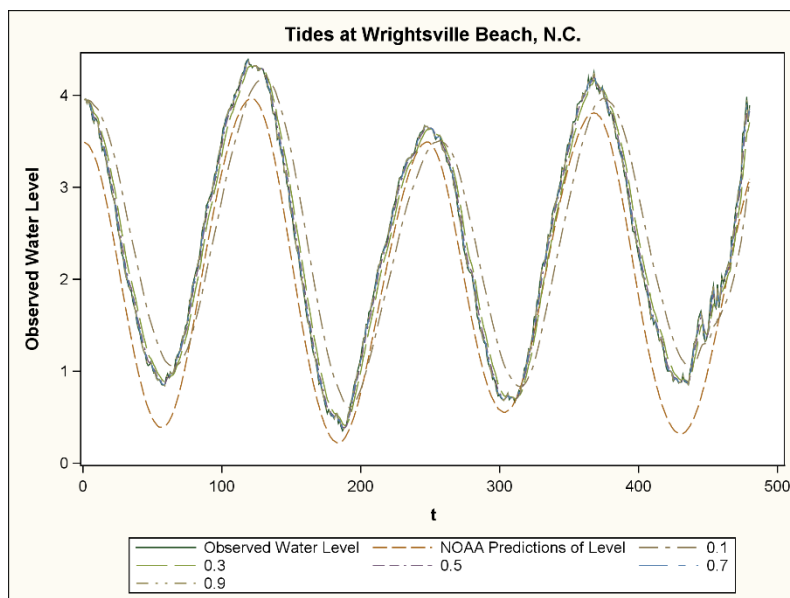
When period $t + 1$ arrives and Y_{t+1} is observed, the forecast of Y_{t+2} is a simple weighted average of the newly observed Y_{t+1} and its former forecast \hat{Y}_{t+1} . When presented in this fashion, $(1 - \theta)$ is typically symbolized ω , and ω is called the *smoothing weight*. The smoothed 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.

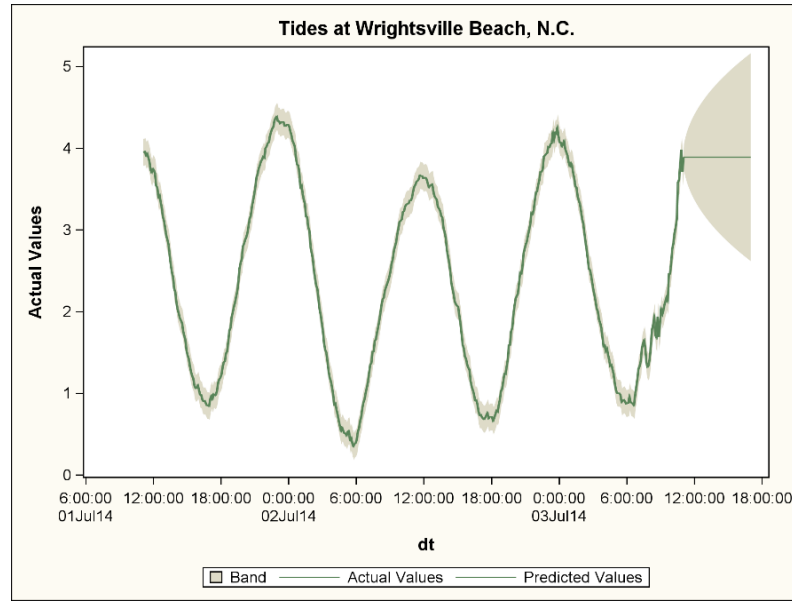
PROC ESM is applied to the data. The variable DT is a SAS datetime variable, and six-minute increments are specified in the ID statement. The LEAD option requests 60 forecasts into the future. The default model in PROC ESM is the simple or single exponential smoothing model, the one just discussed.

```
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.

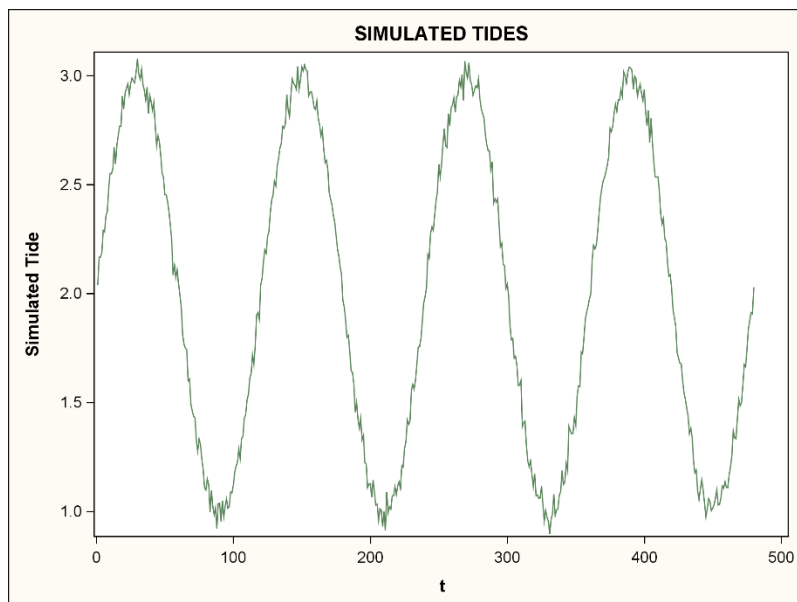
To see one way this might have occurred, **Output 6.4** shows a generated series of 480 observations from the model $Y_t = \beta_0 + \beta_1 X_{1t} + \beta_2 X_{2t} + e_t$, where X_1 and X_2 are a sine and cosine pair with period 120, and e_t is white noise. This is just the typical multiple regression 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



NAME	_TRANSFORM_	_MODEL_	_PARM_	_EST_	_STDERR_	_TVALUE_	_PVALUE_
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";
```

```

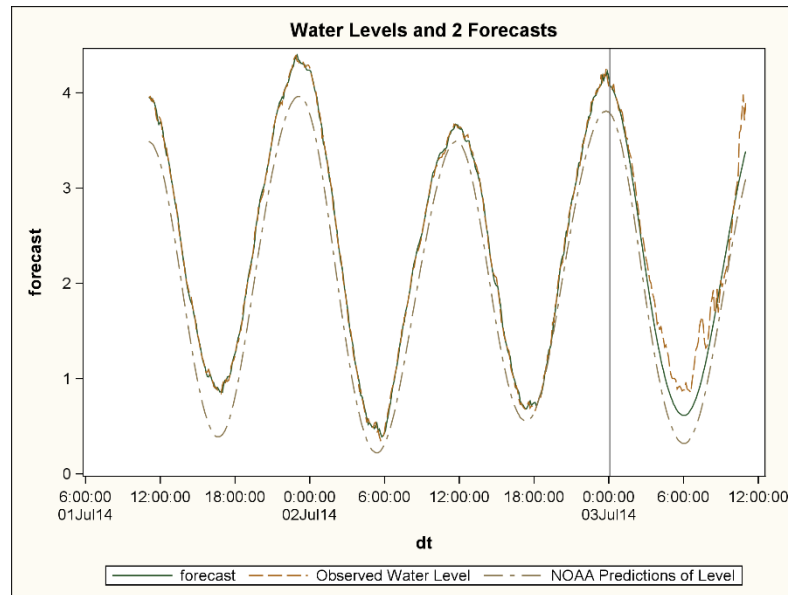
run;
* add predicted difference to noaa forecast ;
data all;
    merge tides esm_fore;
    forecast = noaa_pred+predict;
proc sgplot;
    title "water levels and 2 forecasts";
    series y=forecast x=dt;
    series y=water_level x=dt;
    series y=noaa_pred x=dt;
    refline "03jul14:00:06"dt / axis=x;
run;

```

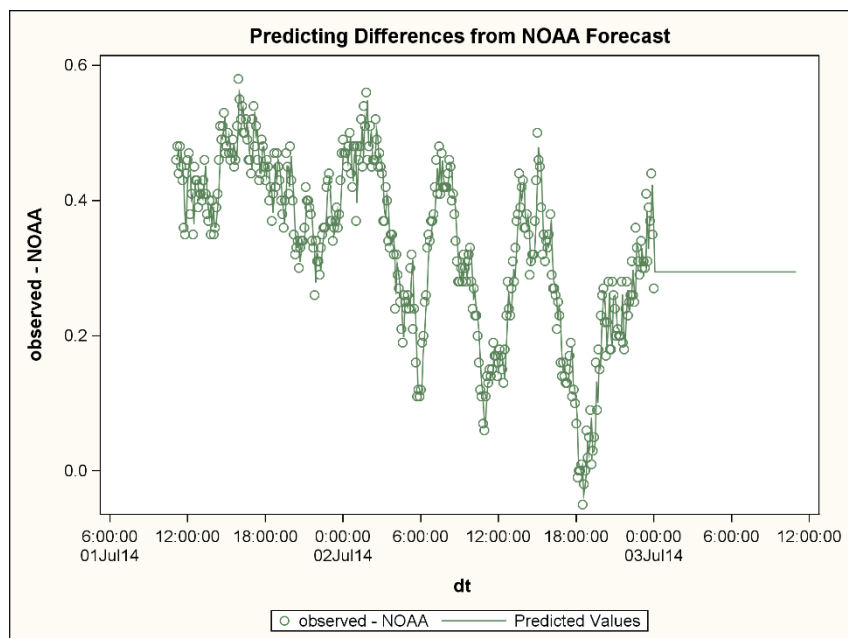
Output 6.5 shows the actual tides and relatively long-range predictions from the NOAA website. Clearly for a forecast of the next few hours, this NOAA forecast would be preferred to the horizontal line forecast from a simple exponential smoothing model fit in PROC ESM. An exponential smoothing model 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 property 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

NAME	_TRANSFORM_	_MODEL_	_PARAM_	_EST_	_STDERR_	_TVALUE_	_PVALUE_
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 $(1 - B)^2 Y_t = (1 -$

$\theta_1 B - \theta_2 B^2$) 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 $(1 - \theta_1 B - \theta_2 B^2)$. That is, assume that $(1 - \theta_1 B - \theta_2 B^2)$ cannot be expressed as $(1 + \theta_2 B)(1 - B)$. A factor $(1 - B)$ would cancel out one of the differencing operators, $(1 - B)$, on the left.

Any model of the form $(1 - B)^2 Y_t = (1 - \theta_1 B - \theta_2 B^2)e_t$ results in a linear forecast. If the last observation is at time n , then the one-step-ahead forecast is $\hat{Y}_{n+1} = Y_n + (Y_n - Y_{n-1}) - \theta_1 e_n - \theta_2 e_{n-1}$, so that $\hat{Y}_{n+1} - Y_n = (Y_n - Y_{n-1}) - \theta_1 e_n - \theta_2 e_{n-1}$. 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 $\hat{Y}_{n+2} = \hat{Y}_{n+1} + (\hat{Y}_{n+1} - Y_n) - \theta_2 e_n = \hat{Y}_{n+1} + (Y_n - Y_{n-1}) - (\theta_1 + \theta_2)e_n - \theta_2 e_{n-1}$, which is an increase of $\beta = (Y_n - Y_{n-1}) - (\theta_1 + \theta_2)e_n - \theta_2 e_{n-1}$ 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 :

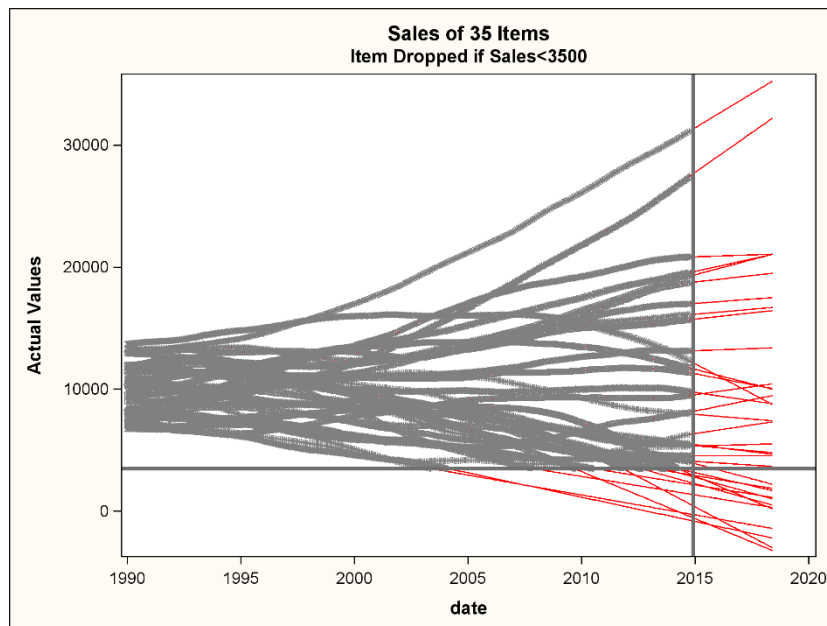
$$\begin{aligned} L_t &= L_{t-1} + T_{t-1} + \omega e_t \\ T_t &= T_{t-1} + \omega \gamma e_t \end{aligned}$$

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 / lineattrs=(color=red thickness=1 pattern=solid)
        group=_name_;
    scatter y=actual x=date / markerattrs=(color=gray size=4) group=_name_;
    refline 3500 / axis=y lineattrs=(thickness=3);
    refline "01dec2014"d /axis=x lineattrs=(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 supresses 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:

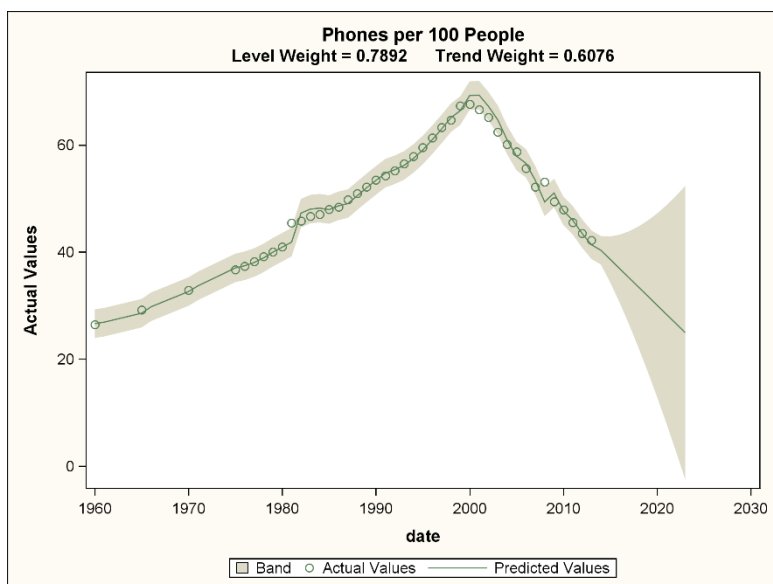
```

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
;

%wldbnc(phones_100, lines);

```

Output 6.9: Phone Lines per 100 People



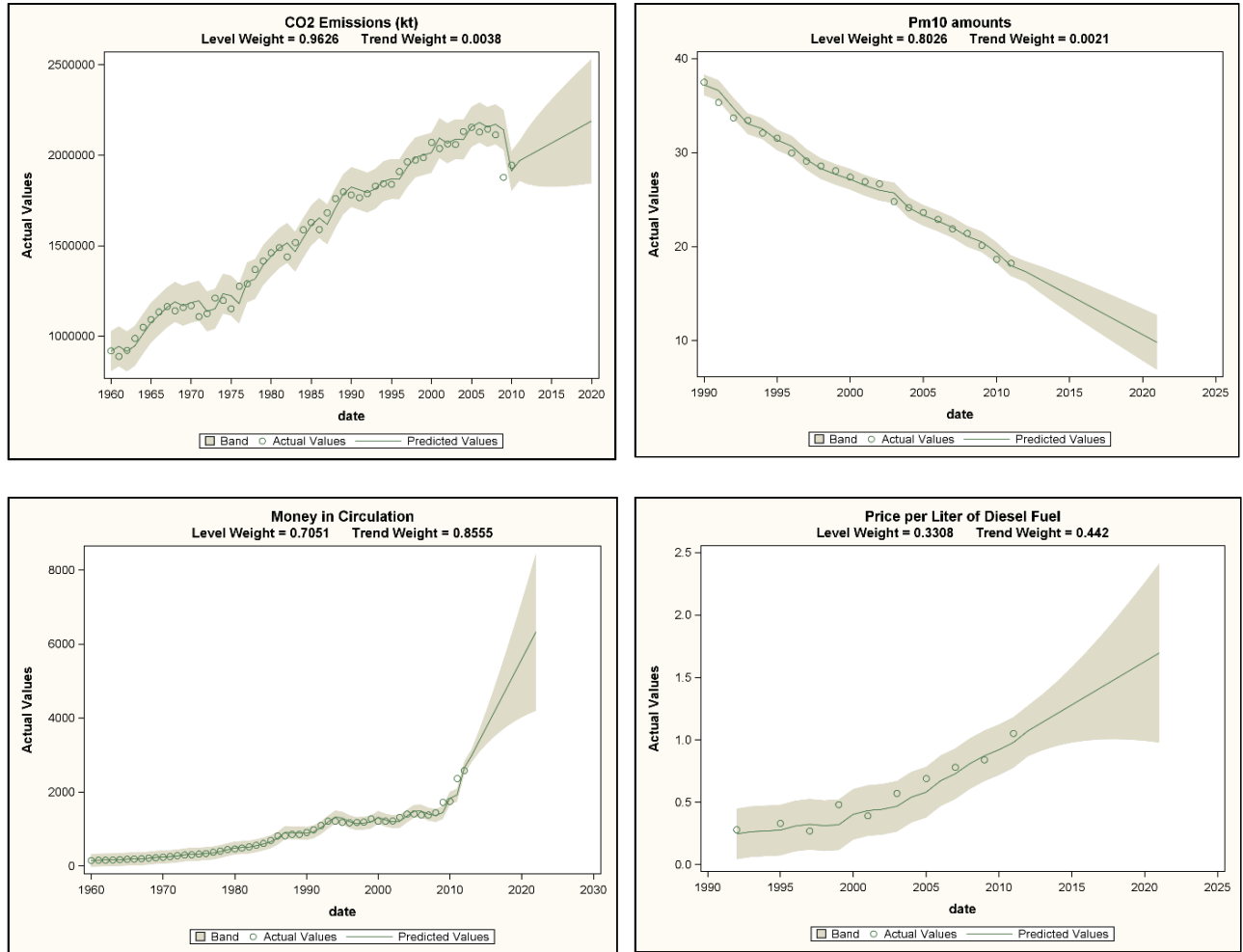
NAME	_TRANSFORM_	_MODEL_	_PARAM_	_EST_	_STDERR_	_TVALUE_	_PVALUE_
LINES	NONE	LINEAR	LEVEL	0.78915	0.10908	7.23458	.000000009
LINES	NONE	LINEAR	TREND	0.60761	0.17668	3.43904	.001377721

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.

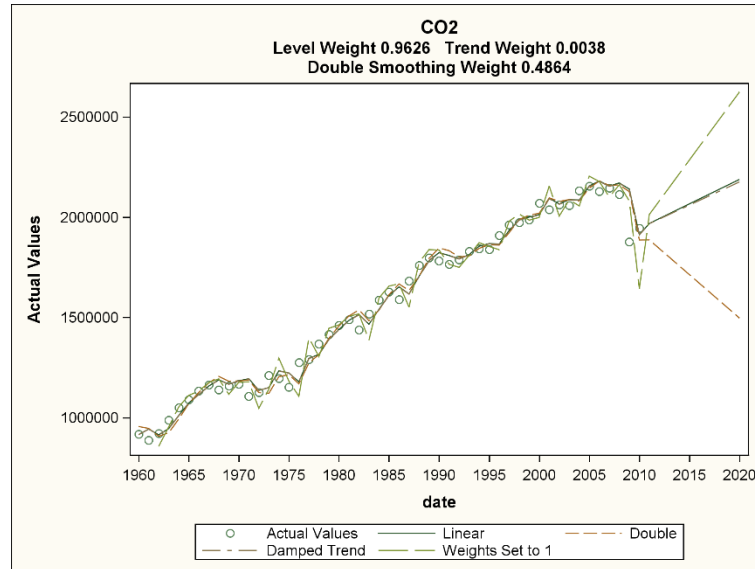
Four more examples, using the World Bank data and macro, are shown in **Output 6.10**.

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 $\hat{Y}_{t+1} = Y_t + (Y_t - Y_{t-1})$.

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.

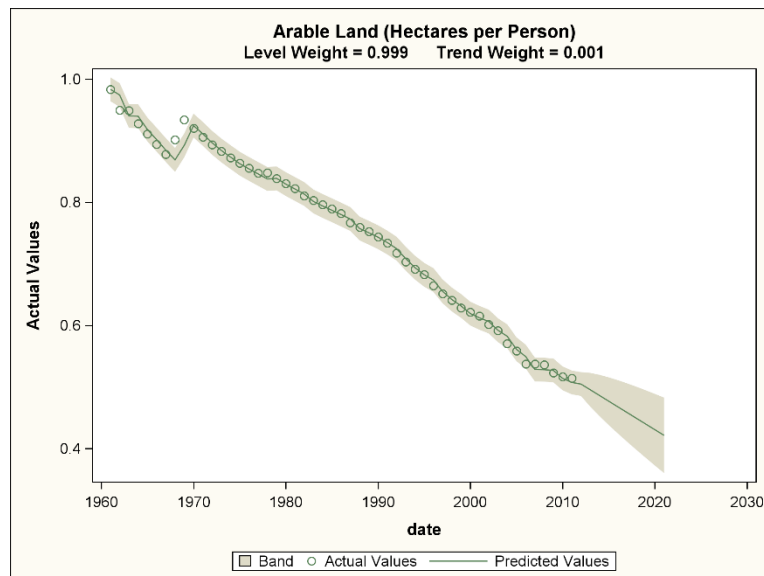
Suppose both smoothing parameters are at the limit value 1, so that the moving average parameters $\theta_1 = (2 - \omega - \omega\gamma)$ and $\theta_2 = (\omega - 1)$ are both 0. The ARIMA model becomes $Y_t = Y_{t-1} + (Y_{t-1} - Y_{t-2}) + e_t$. A one-step-ahead forecast from time n is $\hat{Y}_{n+1} = Y_n + (Y_n - Y_{n-1})$. 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
;

%wldbknk(land,land);
```

Output 6.12: Arable Land in the United States (Hectares per Person)



NAME	_TRANSFORM_	_MODEL_	_PARM_	_EST_	_STDERR_	_TVALUE_	_PVALUE_
LAND	NONE	LINEAR	LEVEL	0.999	0.097685	10.2268	0.00000
LAND	NONE	LINEAR	TREND	0.001	0.046650	0.0214	0.98298

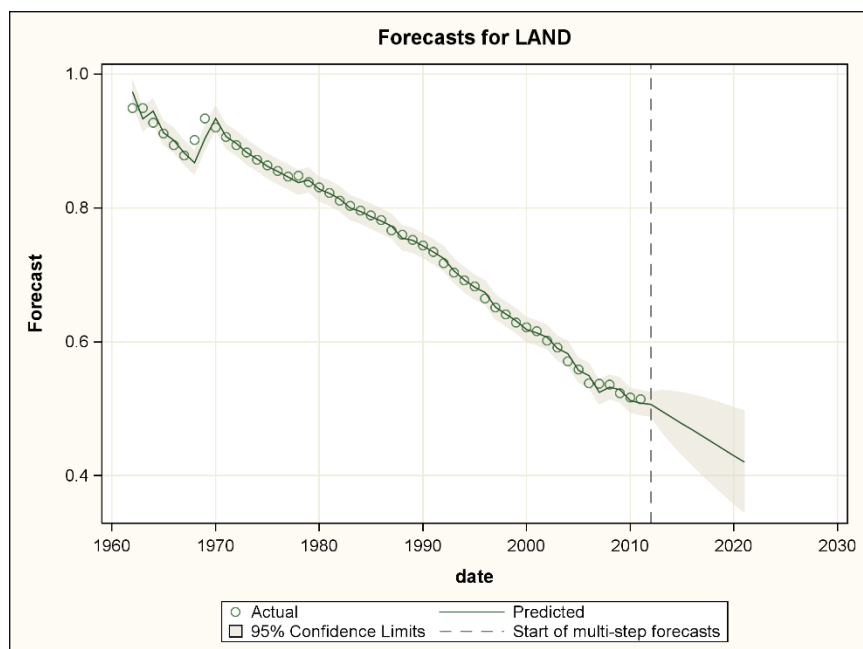
The level smoothing weight has hit the boundary on the $\omega = 1$ side. The trend weight has hit the boundary on the $\gamma = 0$ side. Despite this, the forecast looks reasonable. Without convergence, however, neither the forecast nor the width of the prediction intervals is justified by statistical 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.

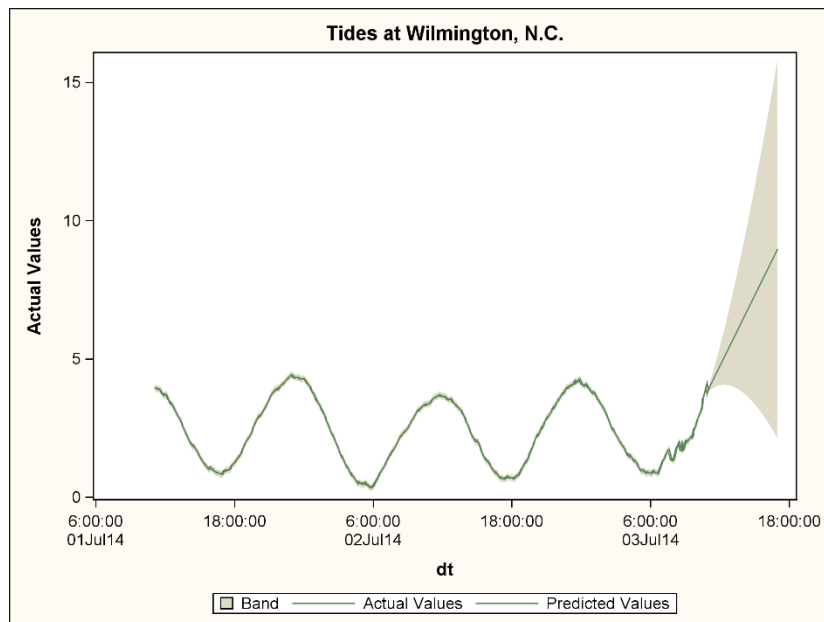
Returning to the tides data, a linear exponential smoothing model gives an even worse forecast far into the future for the original tides data than the single exponential smoothing model. However, the smoothing parameters are well within the allowable range in **Output 6.14**.

```
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



NAME	_TRANSFORM_	_MODEL_	_PARM_	_EST_	_STDERR_	_TVALUE_	_PVALUE_
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;

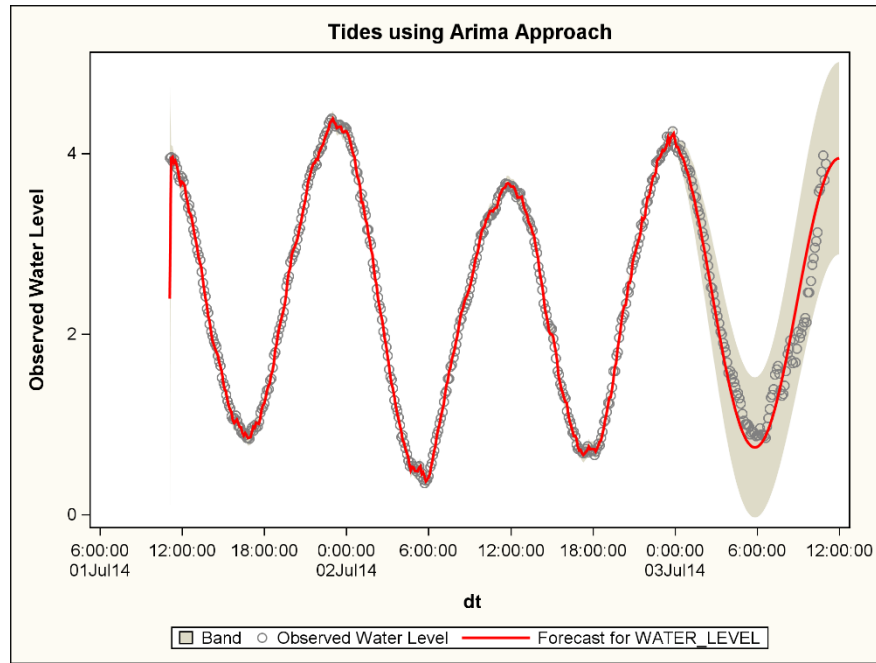
data all;
  merge outarima tides(keep = water_level dt);
  by dt;
run;

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

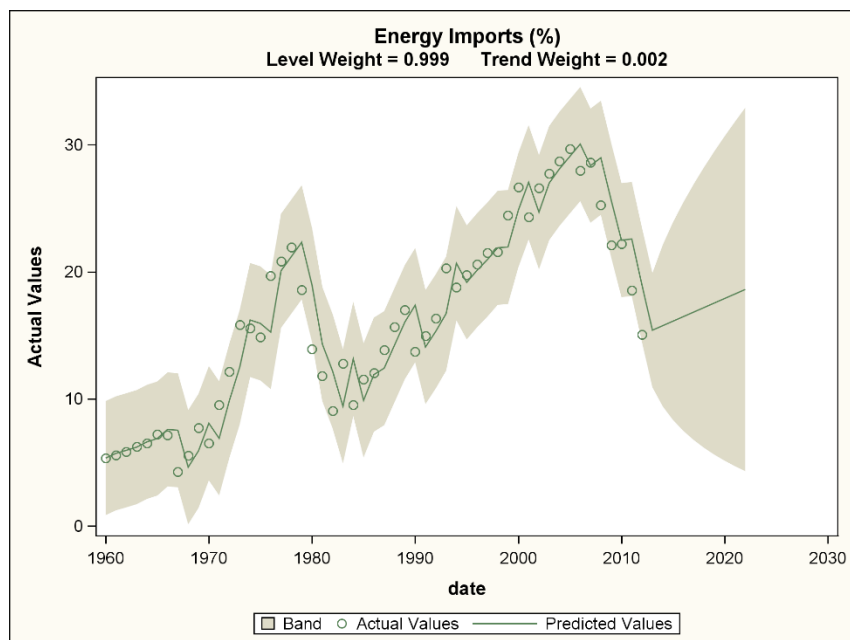


The fit is excellent with all parameter estimates significant and no sign of correlation in the residuals. The forecast intervals are fairly narrow. Yet, they contain all of the withheld data points. The forecasts seem reasonable in regard to the observed past data. A message stating that no improvement 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



NAME	_TRANSFORM_	_MODEL_	_PARM_	_EST_	_STDERR_	_TVALUE_	_PVALUE_
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.

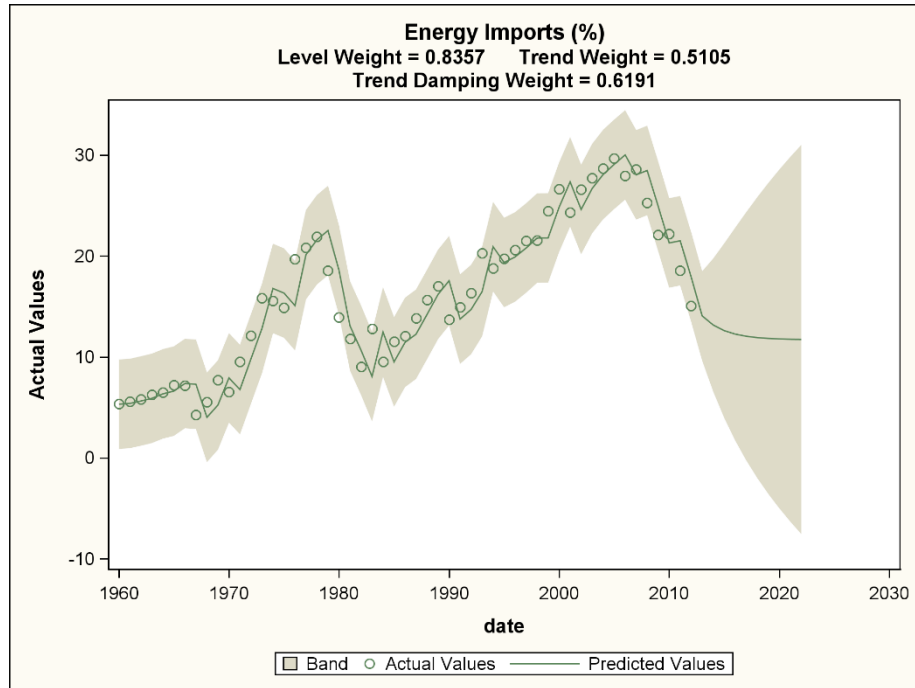
```
proc esm data=energy outest=betas outfor=for lead=10;
    forecast imports / model = damptrend;
    id date interval = year;
run;

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



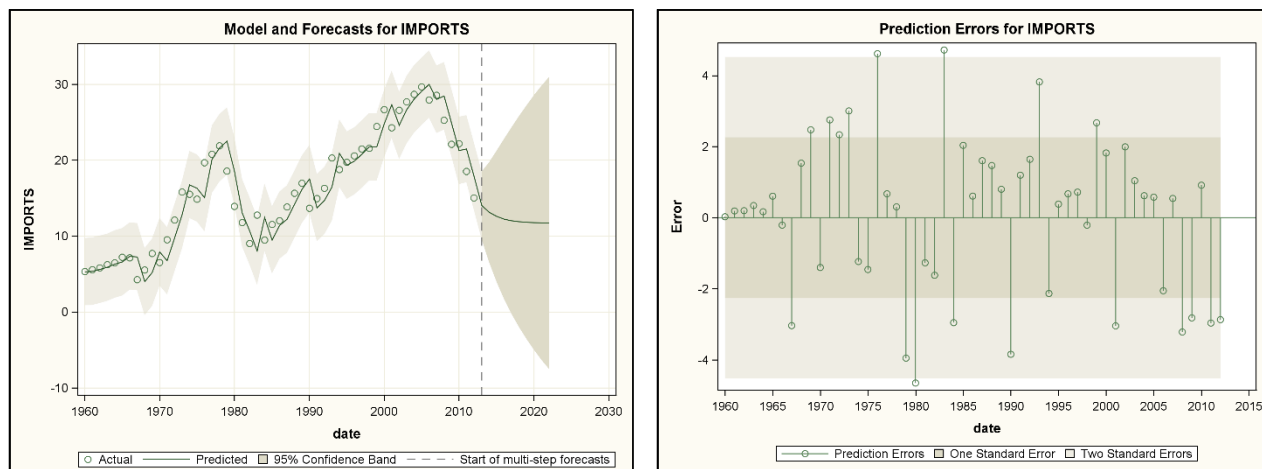
NAME	_TRANSFORM_	_MODEL_	_PARM_	_EST_	_STDERR_	_TVALUE_	_PVALUE_
IMPORTS	NONE	DAMPTREND	LEVEL	0.83568	0.27264	3.06517	0.00350
IMPORTS	NONE	DAMPTREND	TREND	0.51055	0.92948	0.54929	0.58525
IMPORTS	NONE	DAMPTREND	DAMPING	0.61910	0.38760	1.59727	0.11651

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.

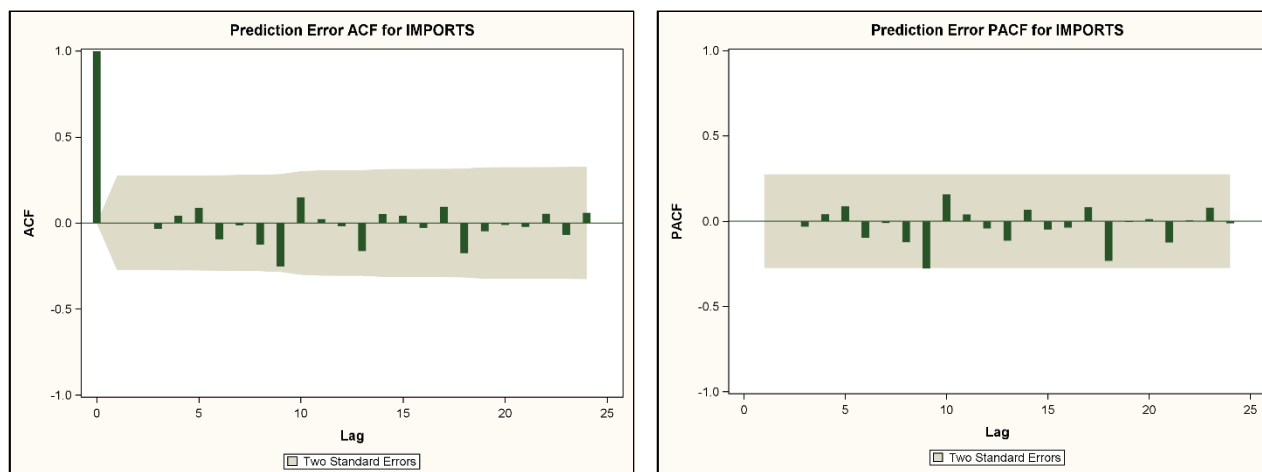
6.2.7 Diagnostic Plots

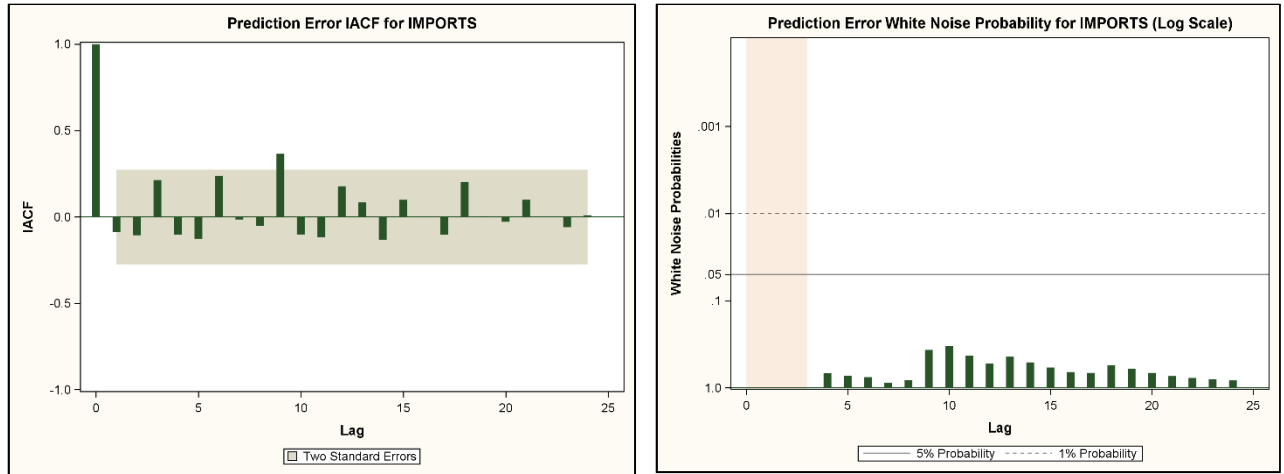
Adding the PLOTS=ALL option to the PROC ESM statement gives several useful plots for the energy imports data.

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

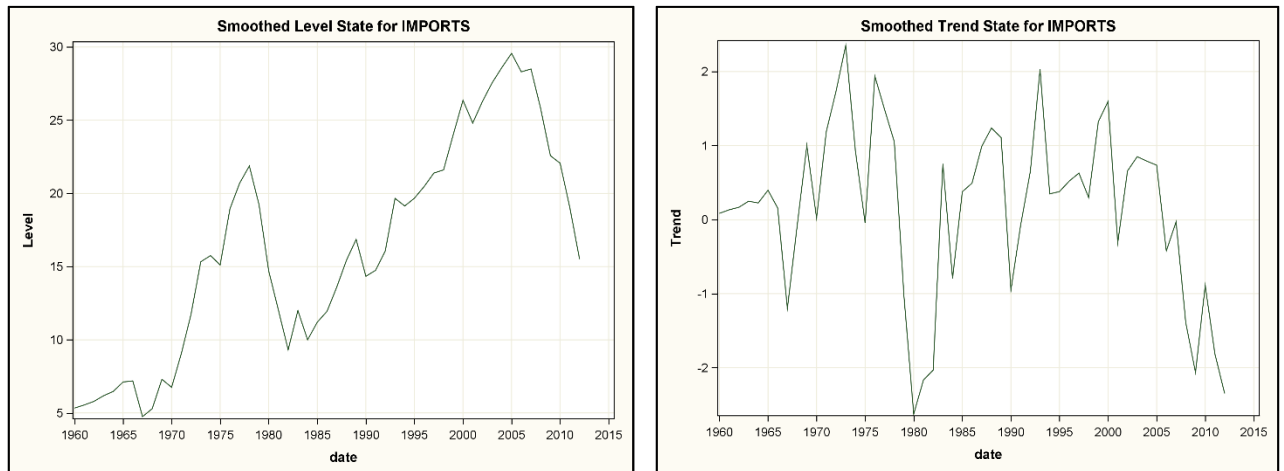
The usual correlation diagnostic panels are shown in **Output 6.19**. They confirm the impressions from the prediction errors plot. The upper left panel is the autocorrelation function. To its right is the partial autocorrelation function. The lower left panel is the inverse autocorrelation function. 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.

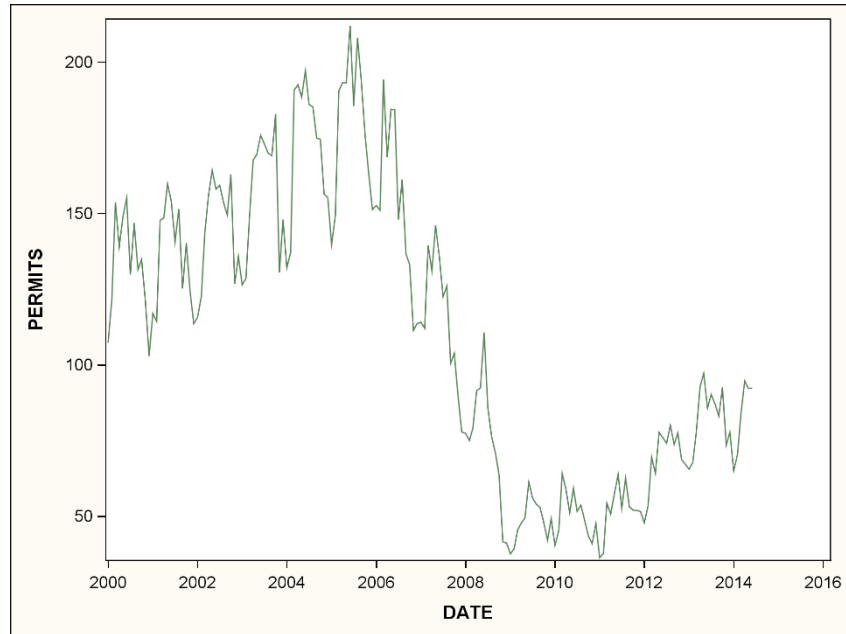
6.3 Smoothing Seasonal Data

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;
```

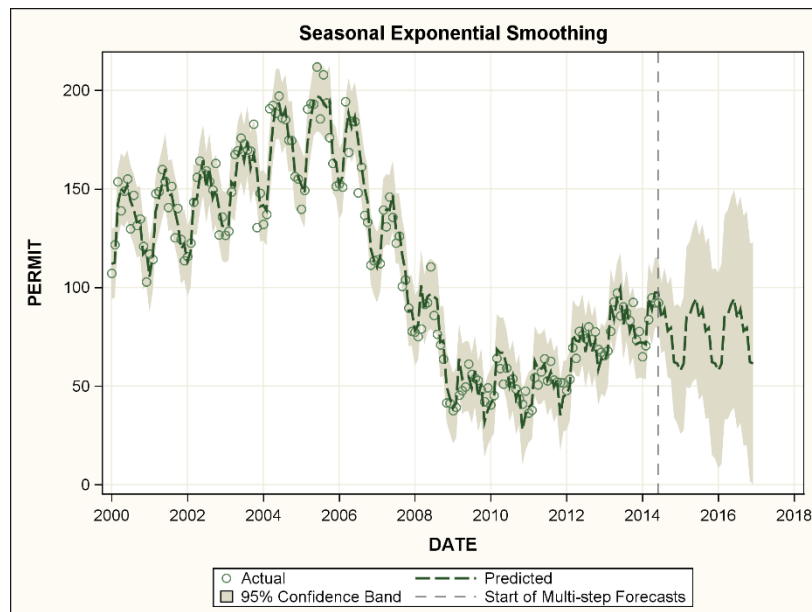
Output 6.22: US Housing Permits Issued (in Thousands)

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

NAME	_TRANSFORM_	_MODEL_	_PARM_	_EST_	_STDERR_	_TVALUE_	_PVALUE_
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.

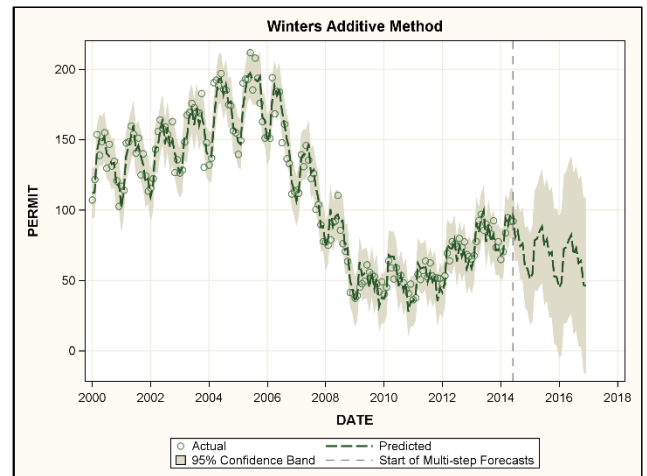
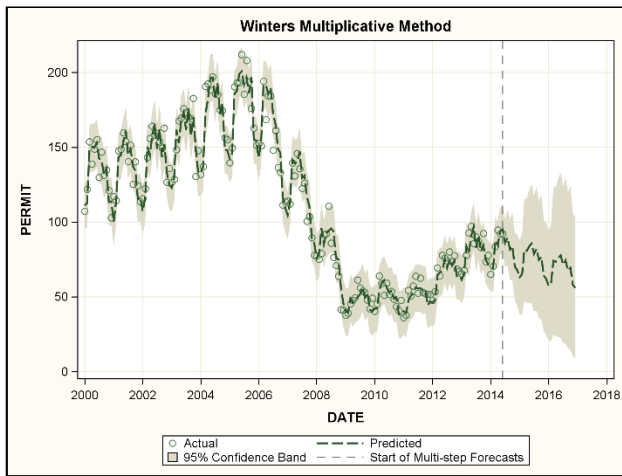
```
proc esm data=permits outest=betas
  outfor=for lead=24 plot = (forecasts modelforecasts);
  forecast permits/method=winters;
  **or use forecast permits/method=addwinters;
  id date interval=month;
run;
```

Output 6.24: Winters Multiplicative and Additive Method results**SMOOTHING BY WINTERS**

NAME	_TRANSFORM_	_MODEL_	_PARAM_	_EST_	_STDERR_	_TVALUE_	_PVALUE_
PERMITS	NONE	WINTERS	LEVEL	0.58656	0.050807	11.5449	0.00000
PERMITS	NONE	WINTERS	TREND	0.00100	0.002969	0.3368	0.73667
PERMITS	NONE	WINTERS	SEASON	0.01156	0.031181	0.3708	0.71122

SMOOTHING BY ADDWINTERS

NAME	_TRANSFORM_	_MODEL_	_PARAM_	_EST_	_STDERR_	_TVALUE_	_PVALUE_
PERMITS	NONE	ADDWINTERS	LEVEL	0.60949	0.050547	12.0579	0.00000
PERMITS	NONE	ADDWINTERS	TREND	0.00177	0.003890	0.4549	0.64973
PERMITS	NONE	ADDWINTERS	SEASON	0.00100	0.039016	0.0256	0.97958



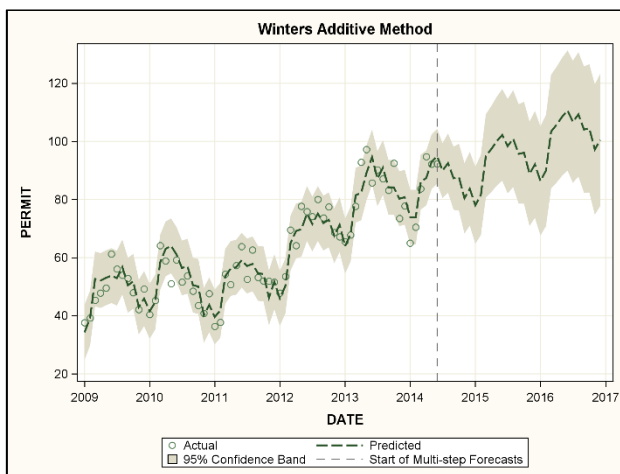
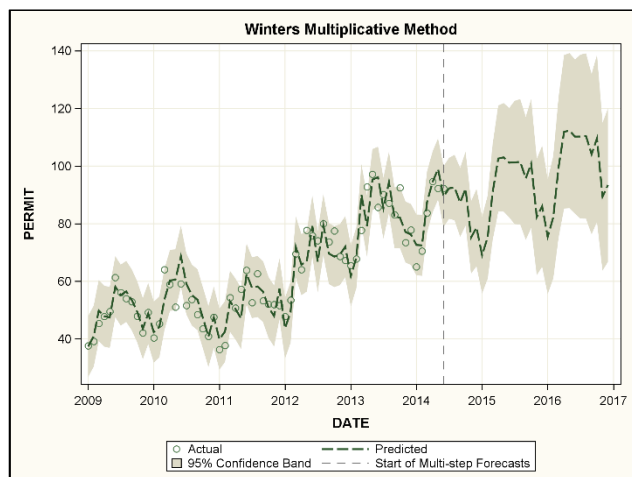
In either case, the level is responsive to local changes as is indicated by the level weight, intermediate between 0 and 1. The forecasts come off the end of the series at the right level. In contrast, the trend (slope) and seasonal weights are very small, indicating that these features are much influenced 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 WINTERS**

NAME	_TRANSFORM_	_MODEL_	_PARAM_	_EST_	_STDERR_	_TVALUE_	_PVALUE_
PERMITS	NONE	WINTERS	LEVEL	0.39383	0.05468	7.20251	0.00000
PERMITS	NONE	WINTERS	TREND	0.00100	0.02369	0.04222	0.96646
PERMITS	NONE	WINTERS	SEASON	0.90927	0.12868	7.06594	0.00000

SMOOTHING BY ADDWINTERS

NAME	_TRANSFORM_	_MODEL_	_PARAM_	_EST_	_STDERR_	_TVALUE_	_PVALUE_
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.

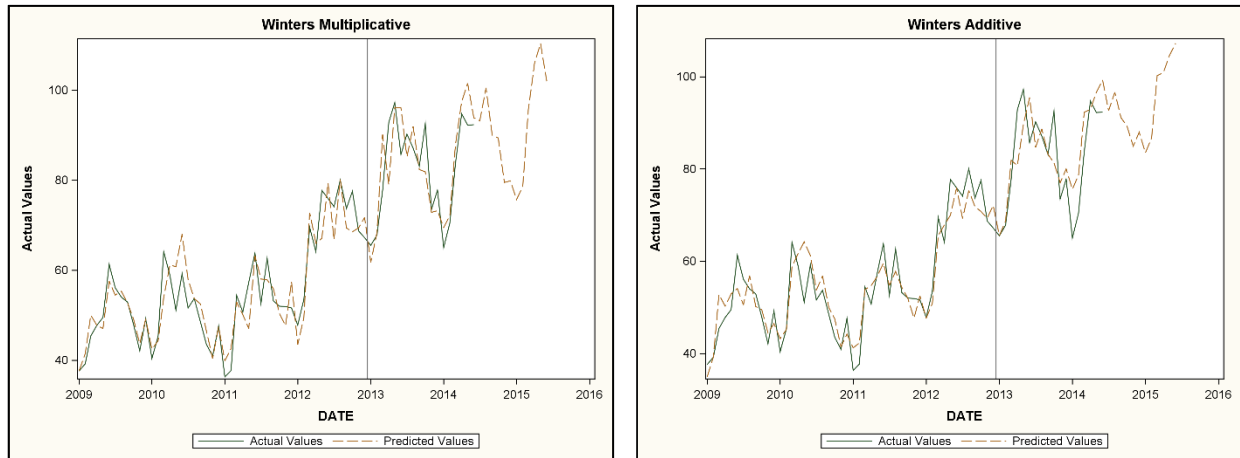
6.4.2 Choosing a Model Visually

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**.

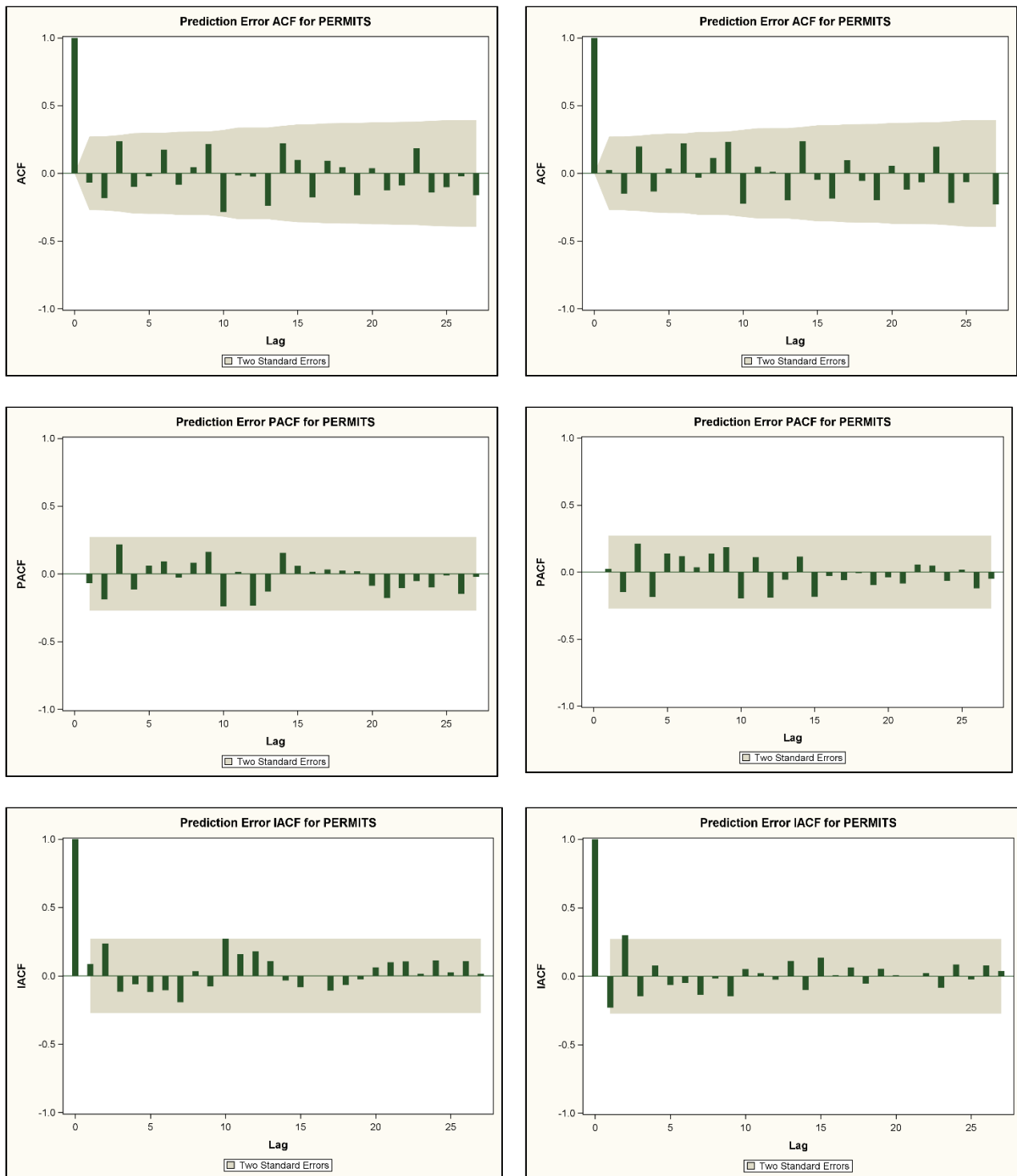
Output 6.26: Multistep Forecasts for Winters Multiplicative (left) and Winters Additive Models (right)

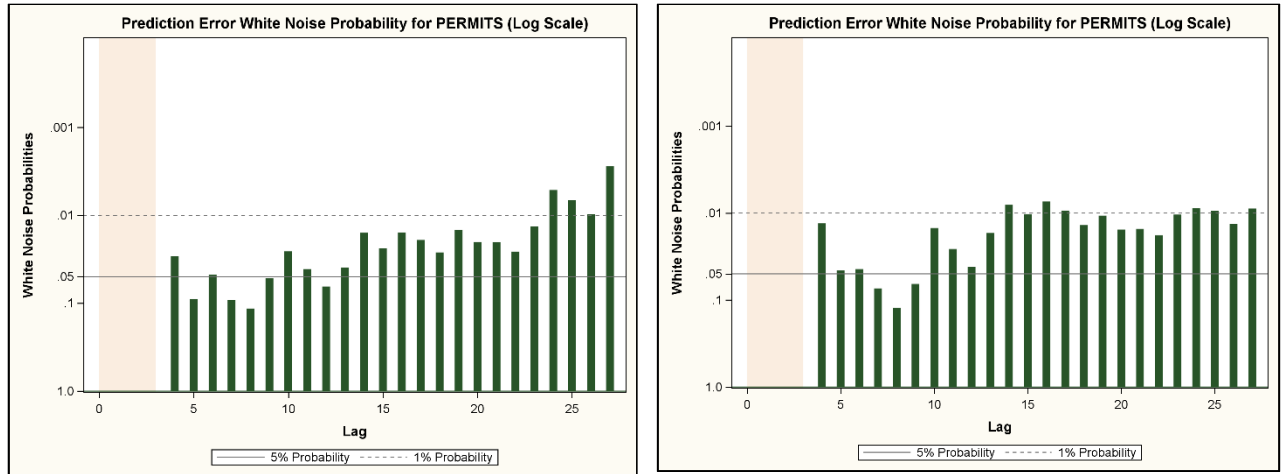


The leftmost plot forecasts (from the multiplicative method) are slightly closer to the last 18 observations. Both methods do a nice job of forecasting.

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;

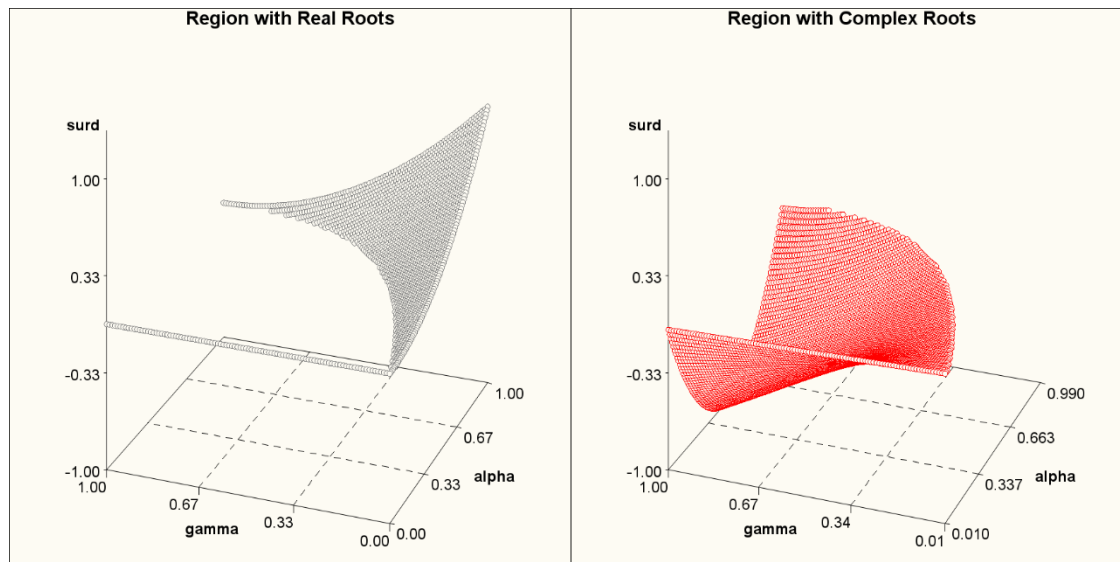
proc g3d; where roots="real";
  scatter gamma*omega=surd / shape = "balloon" size=0.6 noneedle zmin=-1 zmax=1
  color="gray";
  title "REGION WITH REAL ROOTS";;
run;
```

```

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)



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.®

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. © 2017 SAS Institute Inc. All rights reserved. M1588358 US.0217