Forecasting Using SAS® Software: A Programming Approach

Course Notes

Forecasting Using SAS® Software: A Programming Approach Course Notes was developed by David Dickey of North Carolina State University and Terry Woodfield of SAS. Additional contributions were made by Bob Lucas, Michael Leonard, Chip Wells IV, and Donna Woodward. Editing and production support was provided by the Curriculum Development and Support Department.

Forecasting Using SAS® Software: A Programming Approach Course Notes

Book code E2067, course code LWFETSP/FETSP, prepared date 14Nov2011. LWFETSP_002

ISBN 978-1-61290-099-5

Table of Contents

Course Description

This course teaches analysts how to use SAS/ETS software to create forecasting models, evaluate the model for accuracy, and forecast future values using the model.

To learn more…

For information about other courses in the curriculum, contact the SAS Education Division at 1-800-333-7660, or send e-mail to training@sas.com. You can also find this information on the Web at support.sas.com/training/ as well as in the Training Course Catalog.

For a list of other SAS books that relate to the topics covered in this Course Notes, USA customers can contact our SAS Publishing Department at 1-800-727-3228 or send e-mail to sasbook@sas.com. Customers outside the USA, please contact your local SAS office.

Also, see the Publications Catalog on the Web at support.sas.com/pubs for a complete list of books and a convenient order form.

Prerequisites

Before attending this course, you should have experience using SAS to enter or transfer data and to perform elementary analyses, such as computing row and column totals and averages, and producing charts and plots. You can gain this experience by completing the SAS® Programming 1: Essentials and SAS® Programming 2: Data Manipulation Techniques courses. Knowledge of SAS Macro language programming is useful, but not required. A student with no experience in data analysis and statistical modeling can gain the prerequisite knowledge by completing the Statistics 2: ANOVA and Regression course.

Chapter 1 Introduction to Forecasting

1.1 Time Series and Forecasting

Preliminary Remarks

This course uses a programming approach for forecasting. As such, the course goes well beyond teaching proper syntax for SAS forecasting procedures. SAS software includes a powerful programming language, a versatile macro language, and a sophisticated Output Delivery System (ODS) to support forecasting projects. Of course, SAS/ETS software contains many procedures that implement specific forecasting methodologies. You can combine the SAS programming language with SAS procedures to enhance quality and productivity for a forecasting project.

The programs used in this course are designed to run in SAS 9.3, but most programs will also run successfully in SAS 9.2. The SAS macro variable SYSVER provides a mechanism for programmatically checking and conditionally executing code consistent with the release required for the code to function properly.

Objectives

- Introduce time series and forecasting using example data sets.
- Explain the analytic workflow.
- Illustrate the use of ODS Graphics procedures for plotting time series data.
- Describe time series with respect to the concepts of signal (systematic variation) and noise (random variation).

3

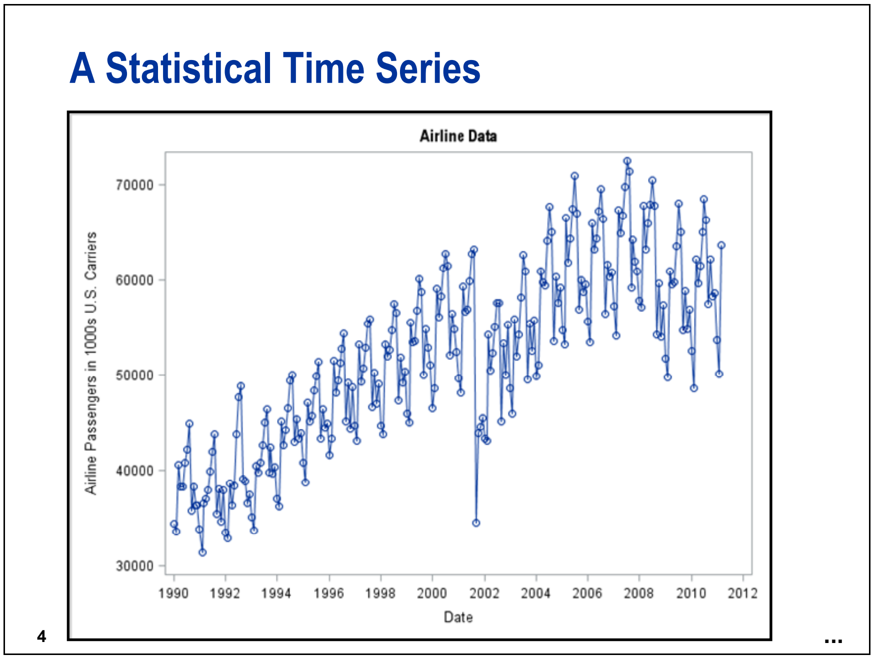
The United States Department of Transportation through its Bureau of Transportation Statistics publishes airline transportation statistics on the following Web site:

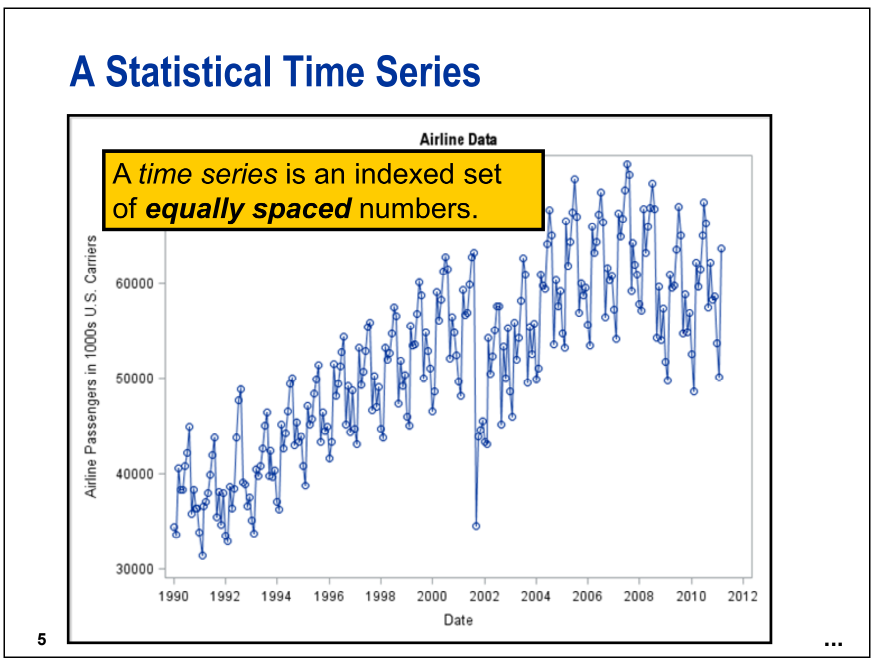
http://www.transtats.bts.gov

Web addresses can change, and content can change over time.

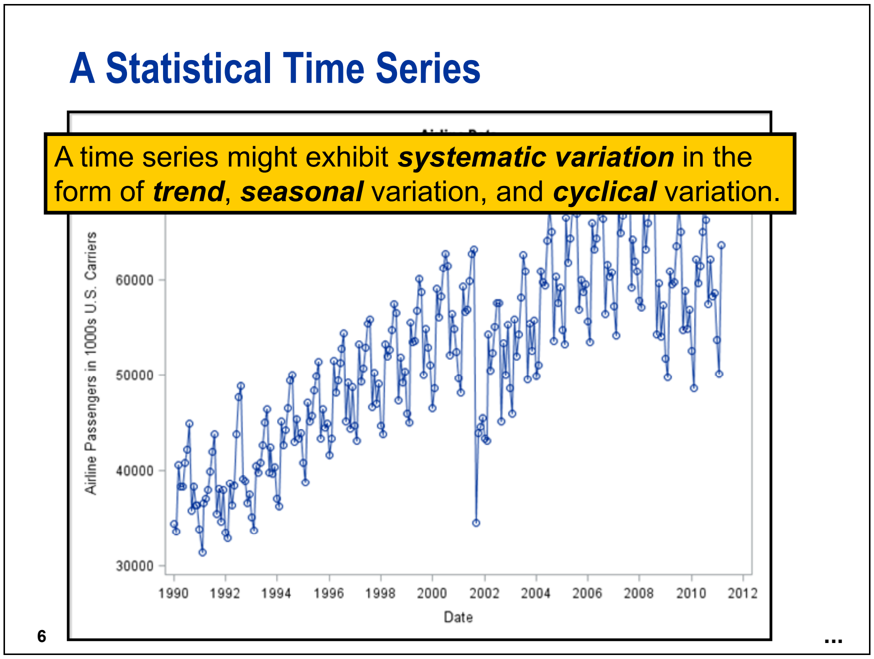
Monthly data for January 1990 through March 2011 was extracted for analysis. The data set contains monthly passenger totals for scheduled flights of domestic U.S. carriers. The data helps illustrate concepts of statistical time series analysis.

A visual examination of time series data helps suggest strategies for forecasting future values of the series. Program **Ch01_01.sas** provides SAS code for producing most of the plots that are included in this chapter.

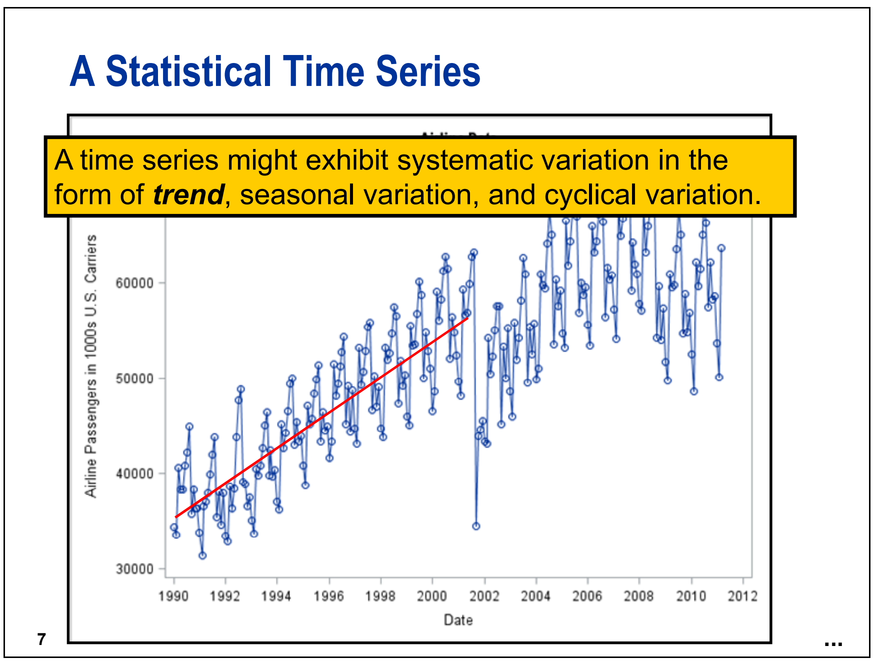




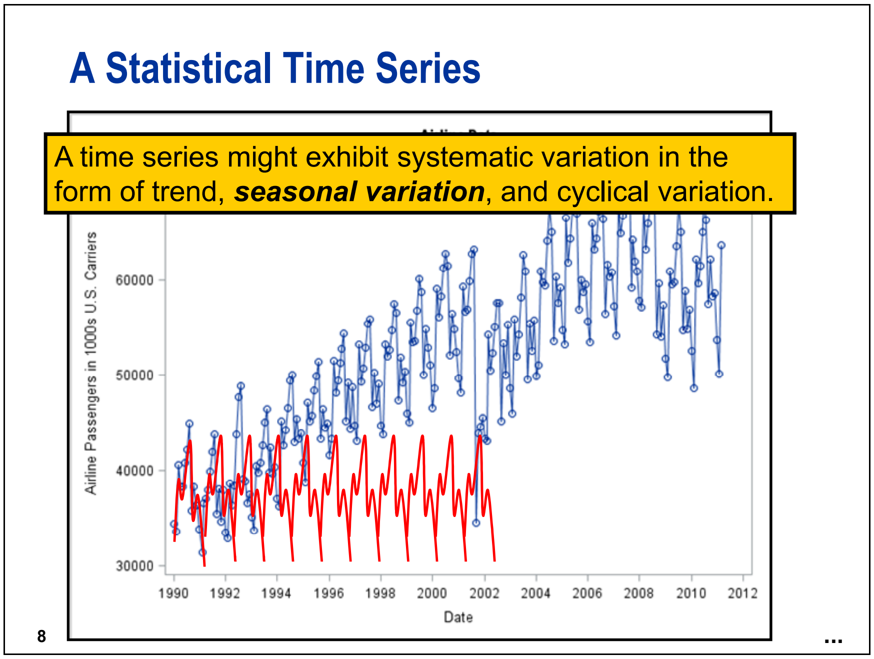
The index that orders a time series is usually a time index, but it could also, for example, be measurements at equidistant points on a metal rod. Equal spacing is critical for the methods described here. Equal spacing need not be mathematically rigid, because, for example, monthly data is considered to be equally spaced even though months have a different number of days. To account for unequal numbers of days, practitioners use trading day adjustments, weekend day adjustments, or similar modifications of the original series.



A time series can be decomposed into trend, seasonal, cyclical, and irregular components. Discussion of the irregular component must wait for additional background information.



Trend represents systematic variation over time, often in the form of deterministic mathematical functions such as polynomials of time, logarithms of time, and exponential functions of time. In the above plot, the time series through the year 2000 seems to follow a linear trend pattern. A linear trend is a first-degree polynomial of time.

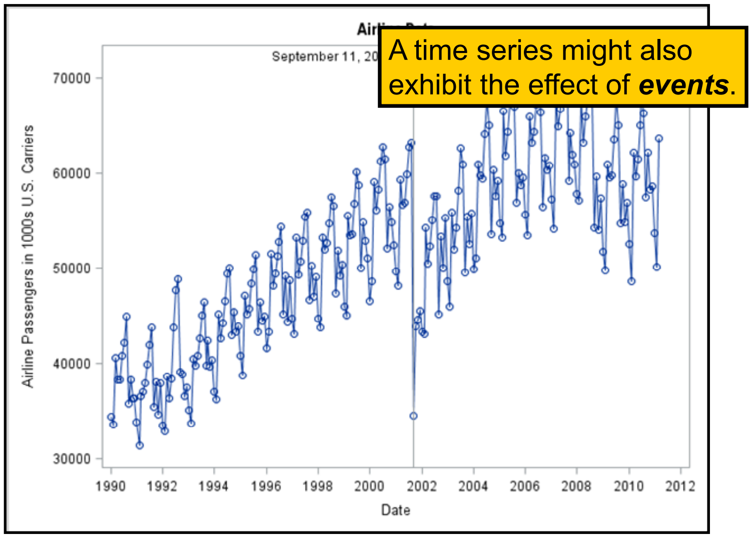


A *seasonal* component exhibits repeat behavior across periods of time. A seasonal monthly time series has a period of 12 and repeats behavior every 12 months. A *cyclical* component is identical to a seasonal component except that the period of the cycle cannot be determined in advance of looking at the data. Seasonal fluctuations usually have a period that is determined in one of three ways:

- By cosmological constants such as the solar cycle of 12 months and the lunar cycle of approximately 28 days
- By social convention such as observing an arbitrary seven-day week
- By mechanical features or physical constants such as a turbine making a complete revolution every 37 seconds

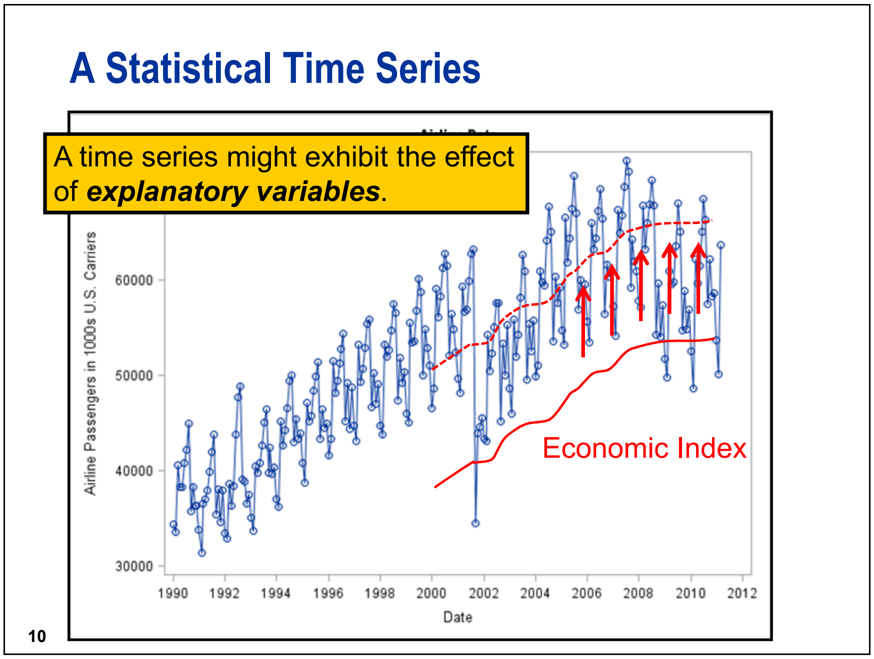
A classic example of a cyclical component is sunspot activity, with peaks occurring approximately every 11 years. If scientists discover a physical cause of this phenomenon, such as information about solar fusion, solar rotation, or another factor implying an 11-year period, then the sunspot cyclical component will be reclassified as a seasonal component. Some physical constants reflecting periodicity no doubt started as empirical observations.

A Statistical Time Series



9 ...

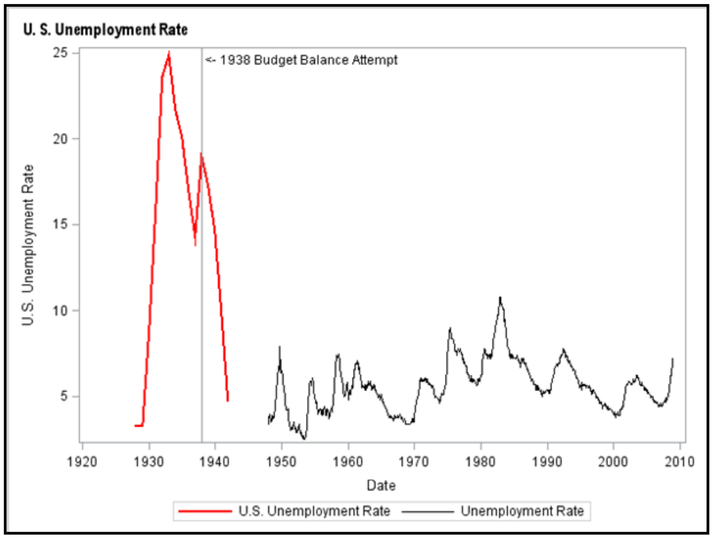
Events often cause structural change in a time series. Events can be random or unexpected occurrences, or they can be planned. Examples of random or unexpected events are cyclones, earthquakes, terrorist attacks, scandal, and illness. Examples of planned events include promotions, organizational restructuring, acquisitions, and price changes.



Time series analysis can be univariate/univariable, where a time series is modeled simply using functions of time and past values of the series. Time series analysis can also be univariate/multivariable, where explanatory variables are added to help explain the behavior of the series. Time series analysis can also be multivariate, which is always multivariable, where two or more time series are modeled together to produce forecasts for each series.

This course addresses univariate time series analysis. A later chapter focuses on multivariable analysis, framing the topic as *time series regression analysis*, also called *dynamic regression analysis*. In time series regression analysis, a regression model having one or more explanatory variables is used to forecast the series of interest.

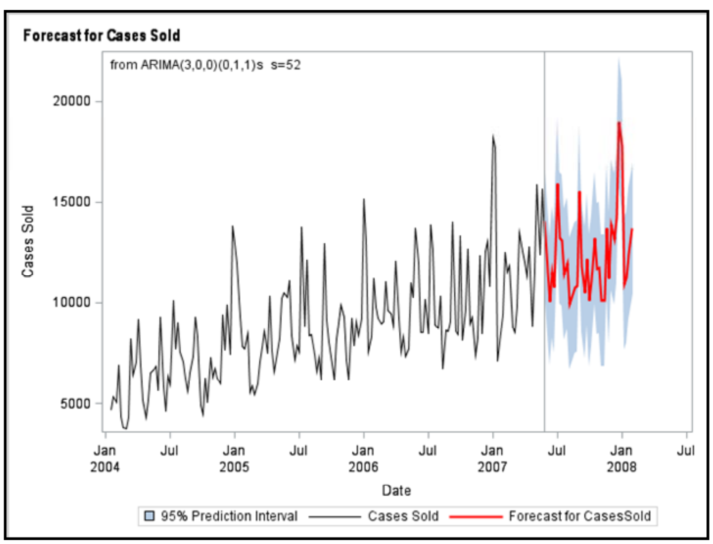
Time Series Analysis Can Be Used to *Explain*



11

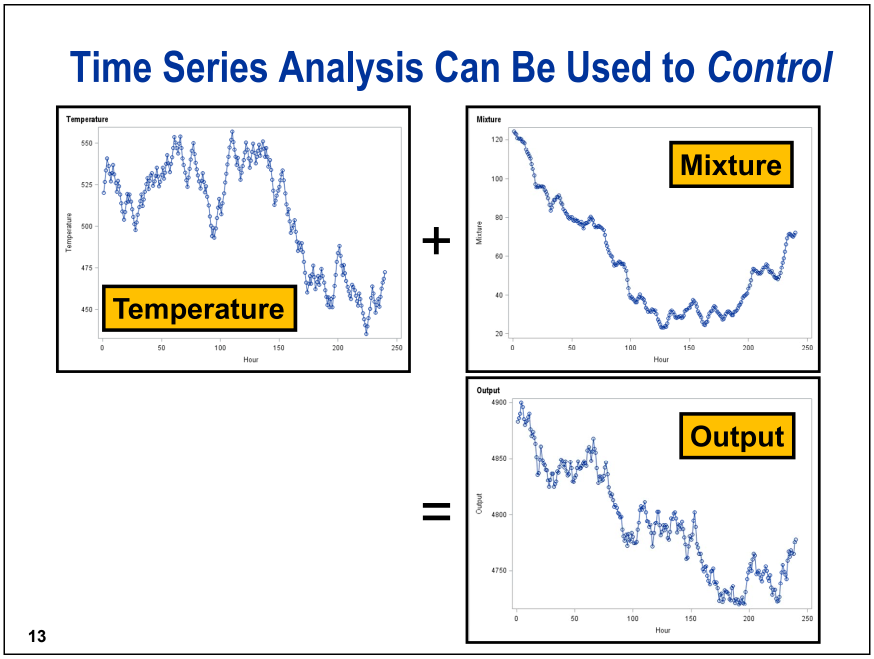
One purpose of time series analysis is to explain or describe the behavior of a time series. Examining the annual time series of civilian unemployment in the United States helps explain effects of public policy during the Great Depression. For example, in understanding the spike in unemployment in 1938, analysis reveals that efforts to balance the budget might have had a negative effect on employment.

Time Series Analysis Can Be Used to *Forecast*



12

Time series analysis is often used in the retail sector to forecast future demand. Confidence intervals help quantify the uncertainty of the forecast.



In a production process, examination of the effects of settings that are at least partially under the control of the process manager help the manager to optimize production. The process must be viewed as a time series because temporal effects contribute to the process. For example, reducing a temperature setting does not have an immediate effect on dropping the temperature to the desired point.

1.01 Multiple Answer Poll

Which of the following are statistical time series?

a. The number of federal tax returns processed annually for the past 10 years
b. Individual ATM deposit amounts for the last 12 months
c. The ages of all current pilots for an airline when the pilots obtained the airline transport pilots' licenses
d. Total circulation for the Sunday edition of a newspaper for each week over the last six months
e. A life table for Belgian males listing the probability of survival to a specified age in years

15

Visualization contributes to understanding a time series. This work emphasizes the use of SAS ODS Graphics, available in SAS 9.2 and later. (An experimental version is available in SAS 9.1.3.) In addition to ODS Graphics, SAS/GRAPH procedures can be employed. SAS Enterprise Guide also supports numerous graphing tasks. This course primarily uses ODS Graphics procedures and the ODS Statistical Graphics output produced by most SAS analytic procedures.

1.02 Multiple Choice Poll

Which environment are you most comfortable working in?

a. SAS Windowing Environment (formerly SAS Display Manager)
b. SAS Enterprise Guide
c. Using a batch program editor and submitting SAS programs in batch
d. None of the above

18

SGPLOT Procedure

PROC SGPLOT DATA=*SAS-data-set*;
    BAND X=*variable* UPPER=*variable*
        LOWER=*variable* / *<options>*;
    SCATTER X=*variable* Y=*variable* / *<options>*;
    SERIES X=*variable* Y=*variable* / *<options>*;
    LOESS X=*variable* Y=*variable* / *<options>*;
    PBSPLINE X=*variable* Y=*variable* / *<options>*;
RUN;

Plot elements are overlaid in the order given.

19

Plotting a Time Series

This demonstration illustrates how to produce a time plot using ODS Graphics procedures.

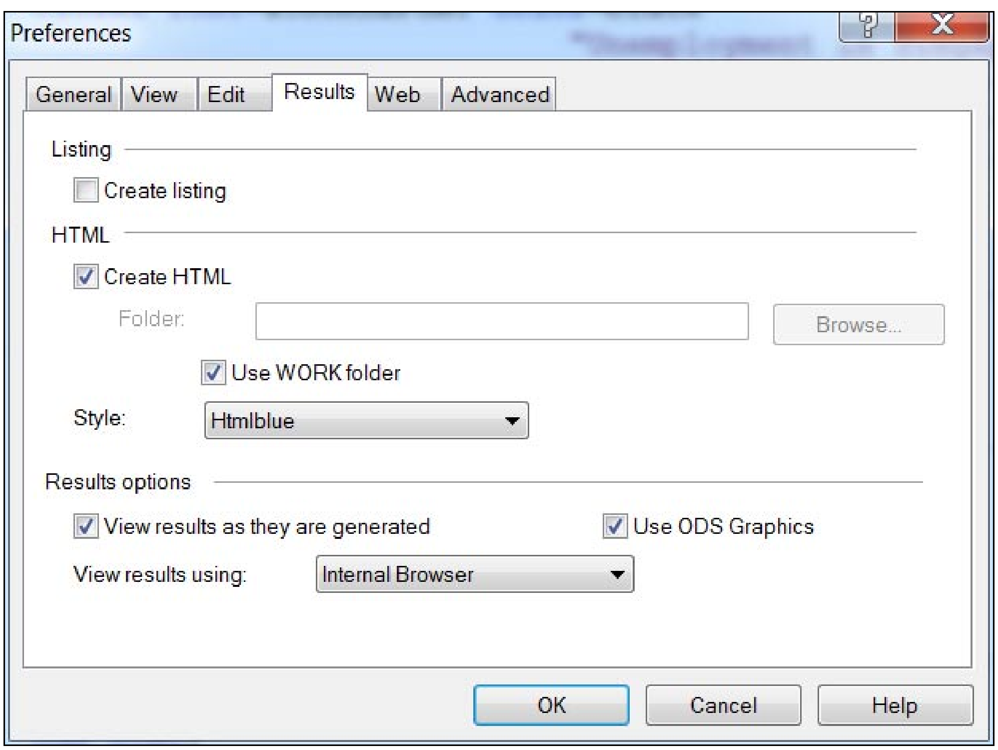
Preliminary Information about Running SAS

The SAS windowing environment is usually invoked by selecting **Start** ⇨ **All Programs** ⇨ **SAS** ⇨ **SAS 9.3**.

The Virtual Lab for this course in a Live Web environment usually has a desktop icon for SAS 9.3. Unless otherwise indicated, all demonstrations use the SAS windowing environment.

If you run programs in the SAS windowing environment, there are a few issues related to the version of SAS you use.

- In SAS 9.2 and later versions, ODS Graphics is available. In SAS 9.1.3, an experimental version of ODS Graphics is available. In prior versions of SAS, ODS Graphics is not available.
- In SAS 9.2, the default output is LIST output, which sends results to the SAS windowing environment ***Output*** window. In SAS 9.3, the default output is HTML, which sends results to an HTML file and opens the file in an HTML viewer, which appears in the SAS windowing environment as the *Results Viewer*. You can change settings by selecting **Tools** ⇨ **Options** ⇨ **Preferences** ⇨ **Results**. A display of a sample Results window is shown below:



You can also use ODS statements, for example, ODS HTML and ODS HTML CLOSE, if you want to control the output in a program.

- In SAS 9.2, ODS Graphics is turned ***off*** by default. In SAS 9.3, ODS graphics is turned ***on*** by default. The following statements control ODS Graphics:

```
ODS GRAPHICS ON;
ODS GRAPHICS OFF;
```

  If ODS Graphics is turned on, then SAS procedures that automatically produce ODS Graphics output do so in the format determined by the current graphics settings. In SAS 9.2, the default behavior is to create a portable network graphics (PNG) file that can be accessed through the Results Explorer. Double-clicking on a PNG result causes the default viewer for the PNG file to open. In SAS 9.3, the graphics files are still PNG files, but they are embedded in the HTML file that is displayed in the Results Viewer. You can access the PNG file directly using the Results Explorer as in SAS 9.2.

- Graphics files and HTML files are stored in your SAS home folder. The SAS home folder can get very cluttered very quickly. The **FETSP.sas** file mentioned below uses the following SAS statement to route all graphics output to the temporary Work library:

```
ods listing gpath="%sysfunc(pathname(work))";
```

  You can also use the Preferences window that is shown above.

This course assumes that the default settings of SAS 9.3 will be used.

If you want to use SAS Enterprise Guide, select **Start** ⇨ **All Programs** ⇨ **SAS** ⇨ **SAS Enterprise Guide 4.3**.

If you run programs in SAS Enterprise Guide, there are a few issues that must be addressed.

- Libraries can be assigned automatically by selecting **File** ⇨ **Open** ⇨ **Data**, or you can select **Tools** ⇨ **Assign Project Library**. Automatically assigned libraries are usually temporary SAS Work libraries. Accessing data sets that are created by one program and accessed by another can pose problems unless you manage the libraries where the data sets are stored and store data sets as permanent data sets.
- SAS Enterprise Guide spawns a new SAS process when you run a program, so macros and macro variables might not persist across programs.
- When you select only a portion of a SAS program to run, the results overwrite previous results, unlike the SAS windowing environment, where results are only added to the Output or Results window.
- SAS Enterprise Guide uses a default output setting that is similar to the default ODS settings used in the SAS windowing environment for SAS 9.3.

When using the SAS windowing environment, after you invoked SAS, while in the Program Editor, select **File** ⇨ **Open Program** to navigate to the **FETSP.sas** program. Run this program first, and run this program every time that you restart your SAS session. Because this program defines macro variables and includes programs that compile course macros, and because SAS Enterprise Guide persists macros and macro variables differently than the SAS windowing environment, you might need to include this program in programs run using SAS Enterprise Guide.

Start of the Demonstration

Data about American Airlines stocks (AMR) was downloaded from the following Web site:

www.dailyfinance.com

Data about stocks typically consists of opening, closing, high, and low prices along with volume. For this stock, as often happens, volume seems to be ***stationary***. For forecasting, it is desirable to have ***normally distributed*** data as well. This is because normal distribution percentiles are used to construct forecast intervals.

The raw data downloaded from the Internet consists of daily stock prices and volume. When the U.S. stock exchanges are closed, for example, for the U.S. Thanksgiving Day holiday, the volume is recorded as zero (0). The data stored in **SASUSER.AMRDAILY** contains the daily stock prices and volume. The data stored in **SASUSER.AMRMONTHLY** represents average monthly values, with calculations based on values set to missing when the volume is zero.

The program **Demo1_01Plot.sas** contains the SAS source code for this demonstration. The program **FETSP_CreateCourseData.sas** contains the code used to read the daily data and create the monthly data.

To illustrate tasks relevant for time series analysis, a subset of the daily data is used. The original data spans September 1995 to March 2011. Such a long range of values tends to produce cluttered graphs, so a subset of the data is chosen from January 2010 to March 2011. The following code subsets the data:

```
data work.amr;
   set sasuser.amrdaily;
   where Date>='01JAN2010'd;
run;
```

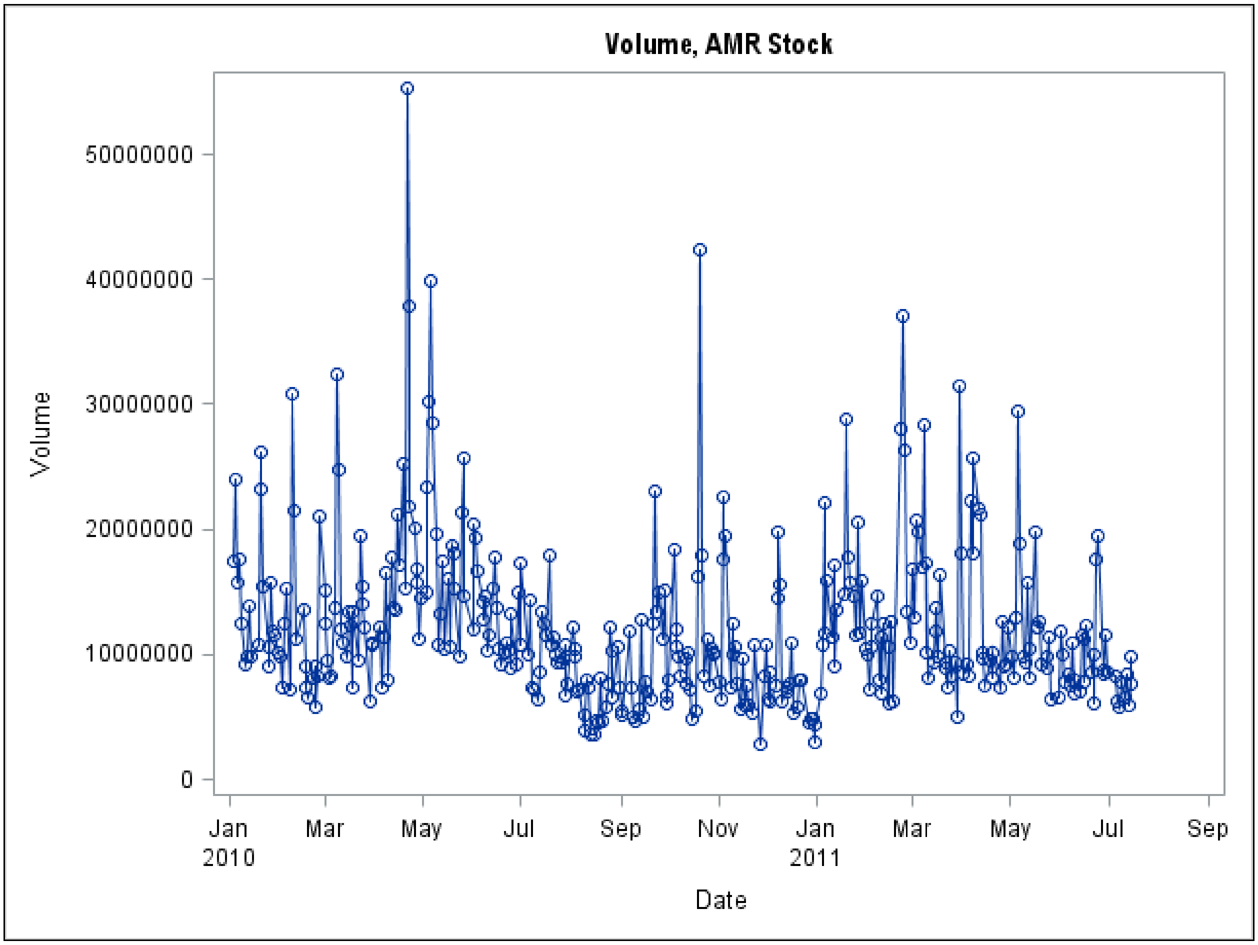
To produce a time series plot, use the SERIES statement in PROC SGPLOT. Because time series analysts typically like to place markers at each data point and then “connect the dots,” you can use the MARKER option or you can combine SCATTER and SERIES statements.

```
title1 font=&COURSEFONT color=black "Volume, AMR Stock";
proc sgplot data=work.amr;
   series x=Date y=Volume / markers;
run;
```

The identical plot is produced with the following code:

```
proc sgplot data=work.amr;
   scatter x=Date y=Volume;
   series x=Date y=Volume;
run;
```

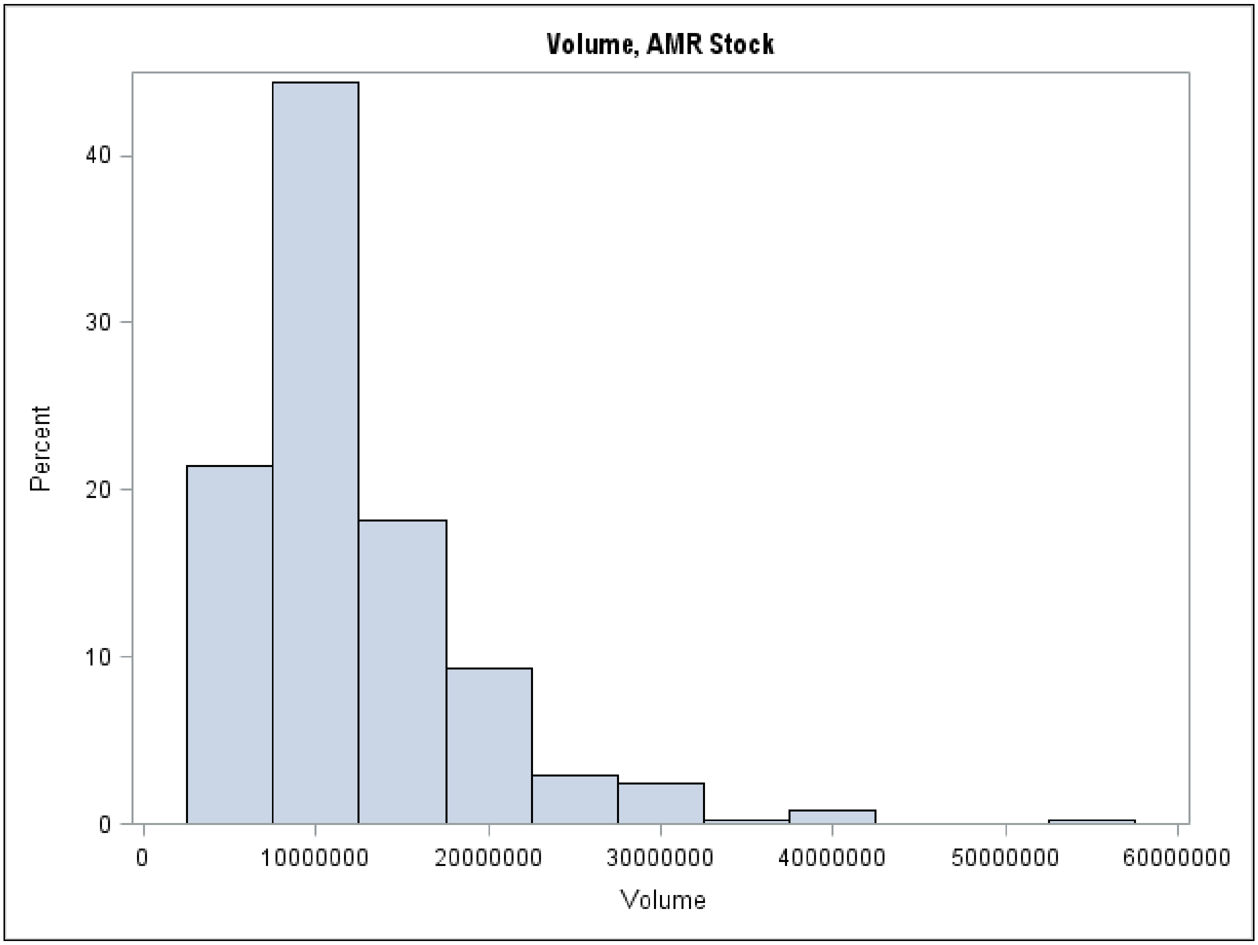
The plot appears below:



A histogram can suggest symmetry, a requirement for a normal distribution, or skewness. The following code produces a histogram for AMR stock volume:

```
proc sgplot data=work.amr;
   histogram Volume;
run;
```

The histogram follows:

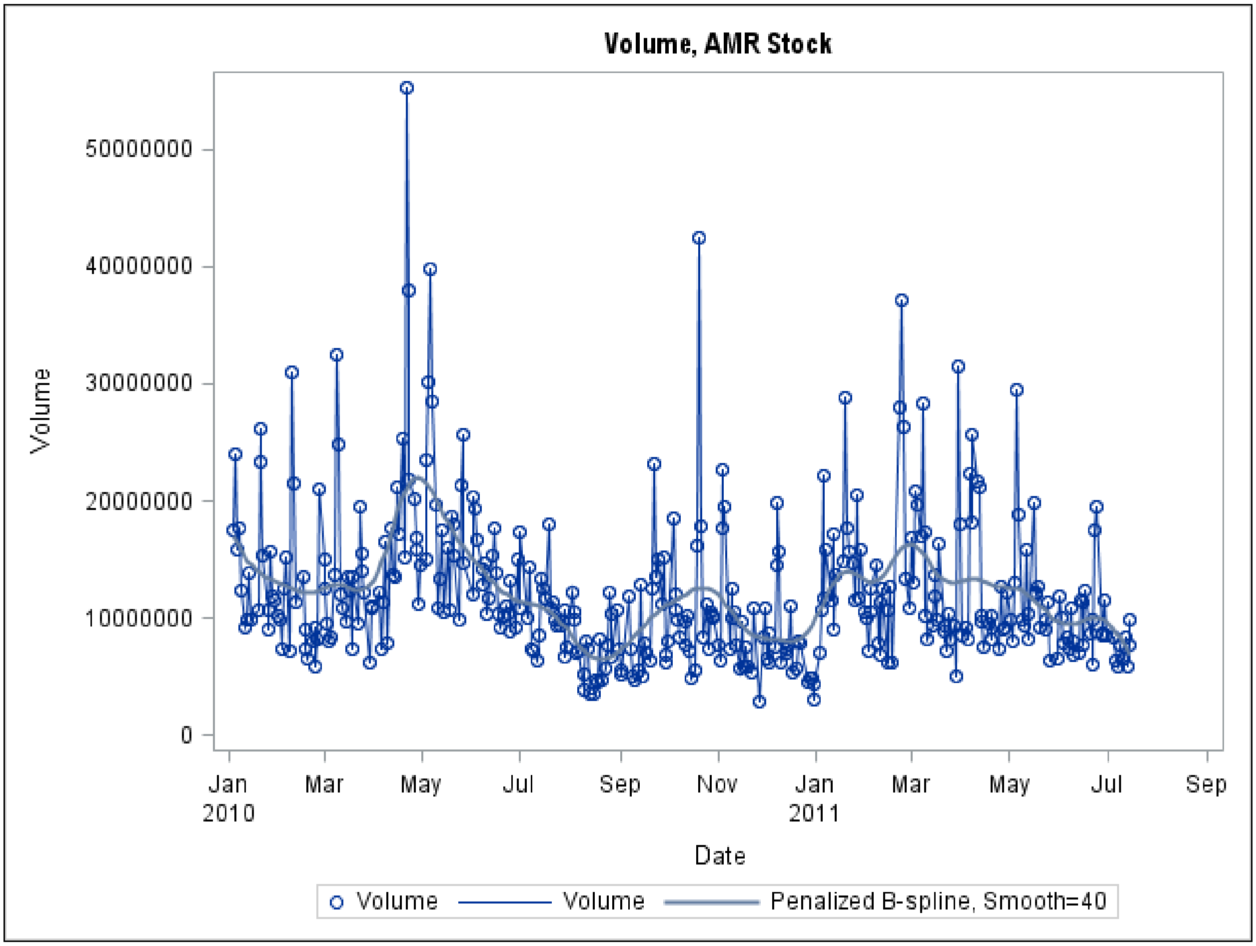


The stock volume is clearly skewed to the right.

With noisy data, you can smooth the plot by using a penalized B-spline. The following code smooths the original time series plot:

```
proc sgplot data=work.amr;
   scatter x=Date y=Volume;
   series x=Date y=Volume;
   pbspline x=Date y=Volume / smooth=40;
run;
```

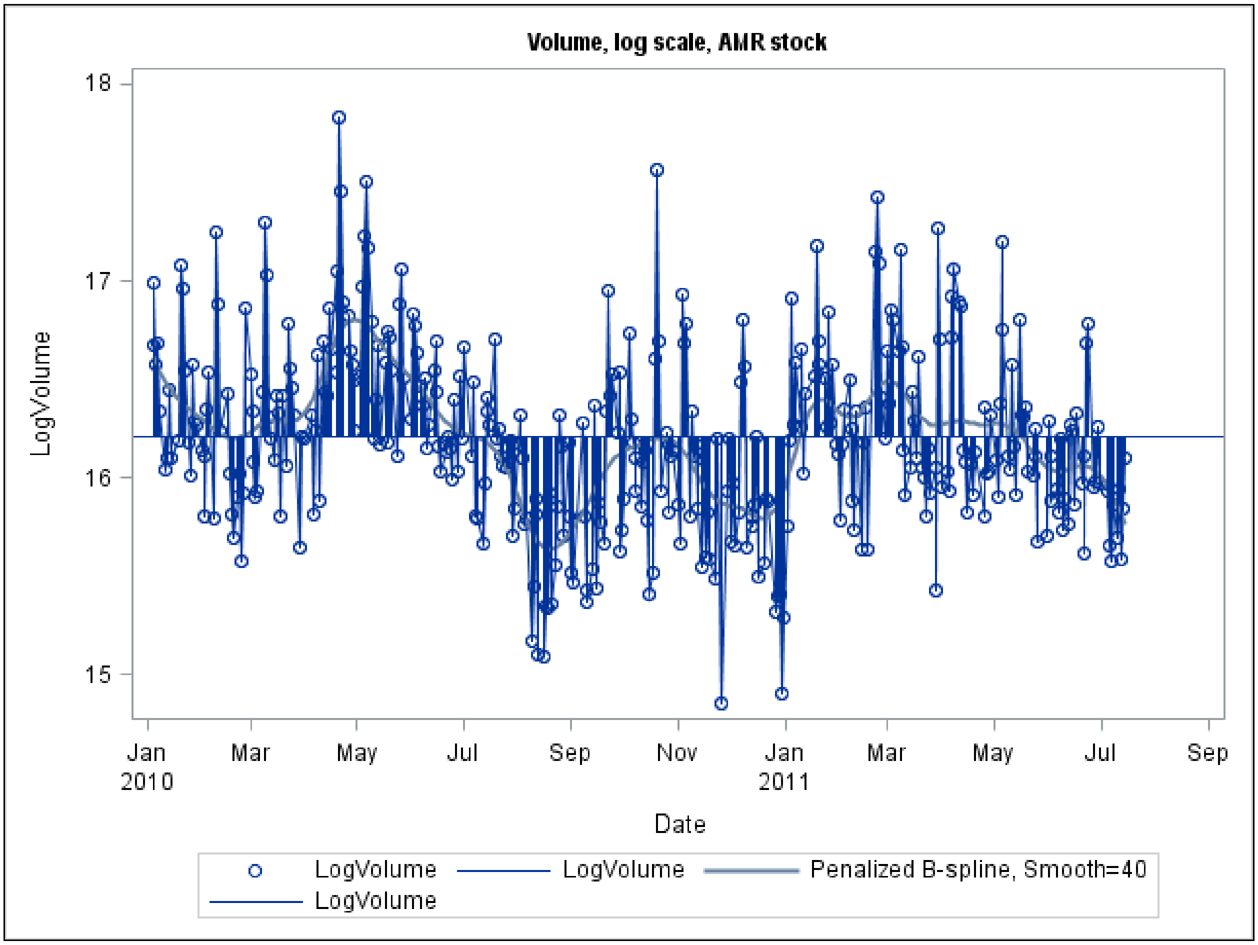
The plot elements are overlaid. First, SGPLOT produces a scatter plot. Next, a series plot is overlaid on top of the scatter plot. Finally, a smooth curve is overlaid on top of the series plot. The results follow.



Notice that the SMOOTH function appears to fluctuate around a fairly stable level, which might be treated as a constant. Notice also that the distribution around that SMOOTH function appears to be skewed to the right, with several much larger deviations above the curve than those below. The normal distribution is symmetric and not skewed, so a natural logarithm might be helpful. The resulting plot shows less skewness. A NEEDLE statement is added to show the up and down movements of the series. The needle plot assumes a baseline of zero, which is not very useful, so the mean of the log series is calculated and used as the baseline.

```
proc sql;
   select mean(LogVolume) into :LVMean
   from work.amr;
quit;
title1 h=1.2 f=&coursefont "Volume, log scale, AMR stock";
proc sgplot data=work.amr;
   scatter x=Date y=LogVolume;
   series x=Date y=LogVolume;
   pbspline x=Date y=LogVolume / smooth=40;
   needle x=Date y=LogVolume / baseline=&LVMean;
run;
```

The plot appears below:

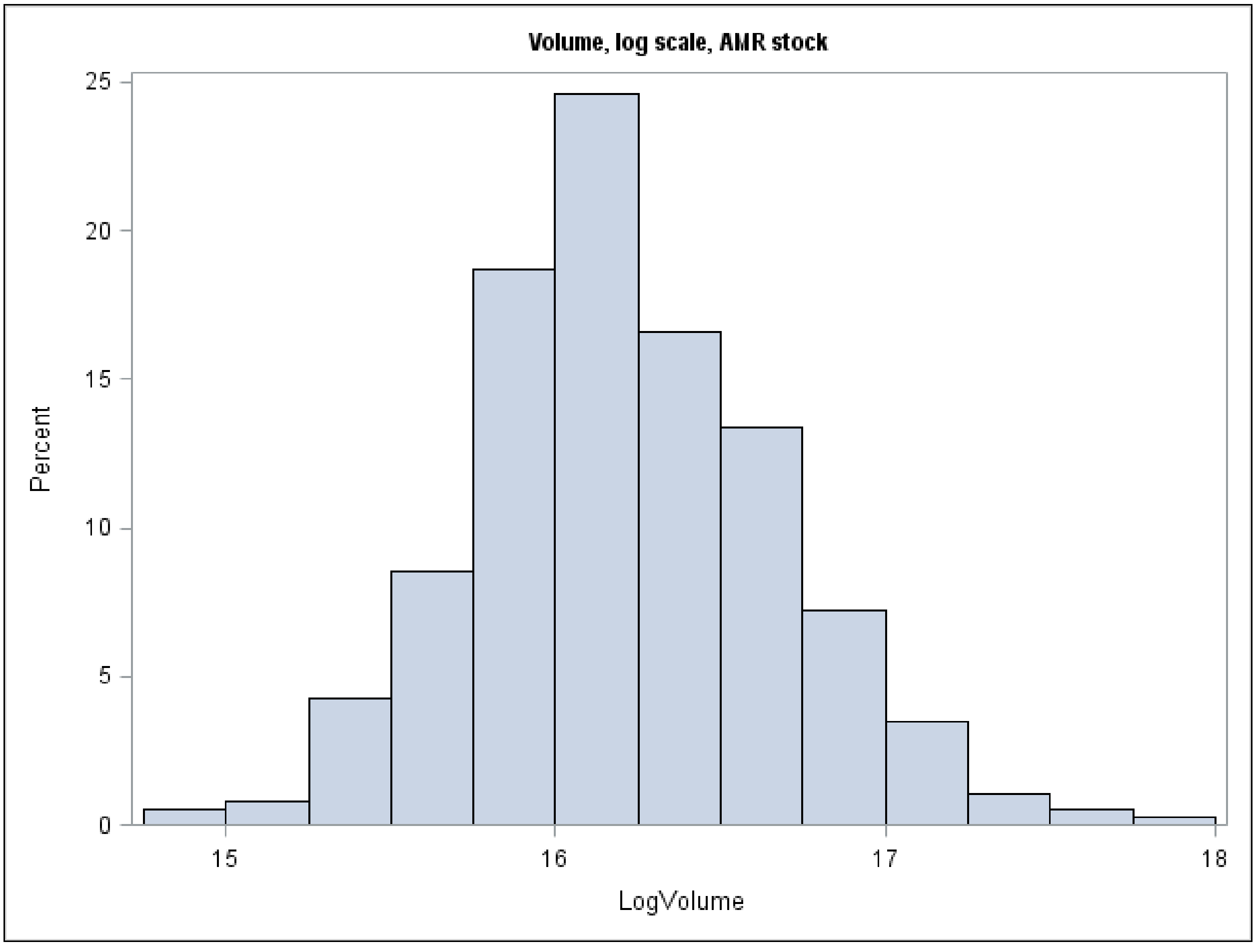


Volume cannot be negative. An advantage of analyzing the data on the log scale is that a negative forecast or confidence limit on the log scale is exponentiated to get it back on the original scale. Even a negative log scale number produces a positive value on the original scale of measurement.

A histogram reveals properties more consistent with a normal distribution. The following code produces a histogram of **Volume** on the log scale:

```
proc sgplot data=work.amr;
   histogram LogVolume;
run;
```

The histogram is shown below:



Compare this to the histogram on the original scale. **Volume** is skewed to the right, primarily because there were several unusually high trading days. The logarithm transformation seems to convert skewed data into data that is more symmetric.

Preparing Data for Forecasting

- Read the raw transactional data.
- Convert the transactional data to time series data by accumulating the data to equally spaced time points.
- Plot the data.
  - Identify data pathologies.
  - Suggest forecasting approaches.
- Address data pathologies.
  - Transform skewed data.
  - Impute missing values.
  - Detect unusual observations (outliers).

21

The demonstration focused on plotting the data, but the data had to originally be processed as stock transactions that were converted to daily values. The daily values eventually have to be accumulated to monthly values to be consistent with another time series, namely the time series of U.S. airline passengers.

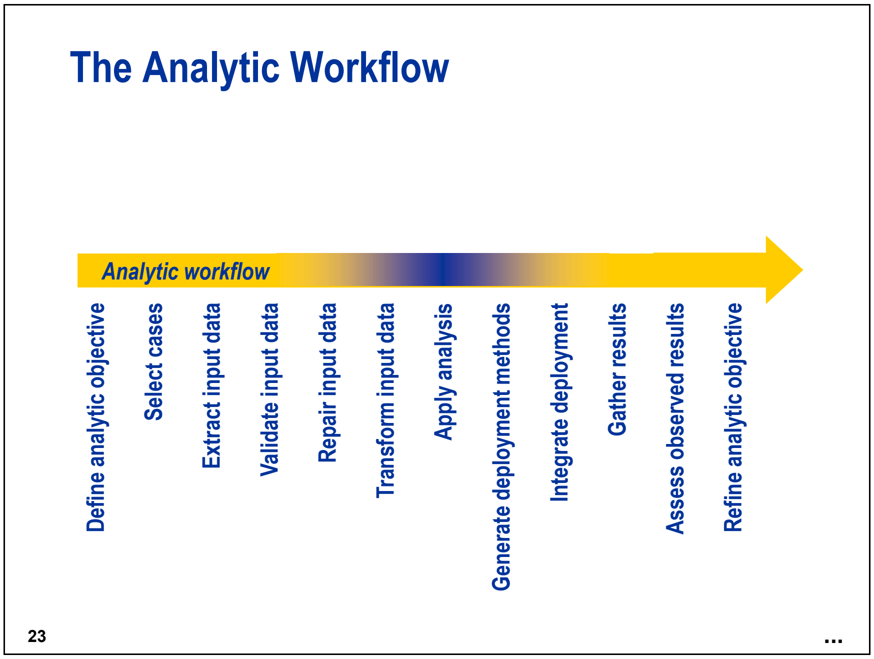
Data preparation represents the most important component of forecasting. Forecasting can be viewed as a process.

The Forecasting Process

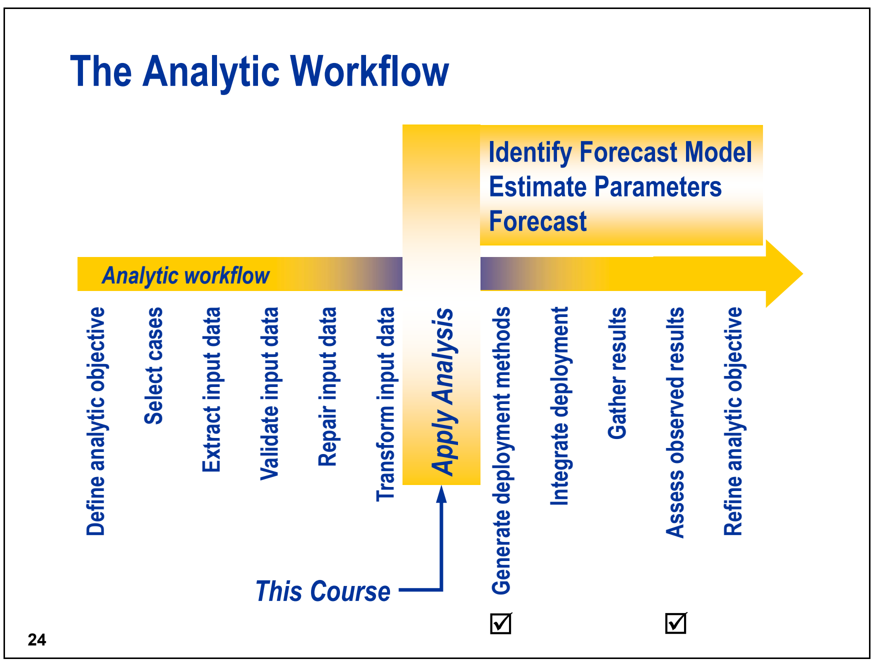
1. Prepare the data.
2. Create diagnostic plots and derive diagnostic statistics.
3. Propose forecast models.
4. Fit proposed models to the data.
5. Diagnose the adequacy of the fitted models.
6. Refine the list of proposed models and return to Step 5. When no more models are suggested, continue.
7. Select a model to use to generate the final forecasts.
8. Produce the forecasts.
9. Distribute, implement, or publish the forecasts.

22

The forecasting process is one realization of the more general analytic workflow.



This course focuses on the analytics of time series forecasting.



To understand forecast models, you need to understand the general approach to modeling.

Statistical Forecasting—The Math

Time Series=Signal + Noise

Time Series=Systematic Variation + Uncertainty

Forecast=Extrapolated Signal

Confidence Interval=Signal +/- Uncertainty

25

In the standard statistical notation, a simple linear trend model can be represented as follows:

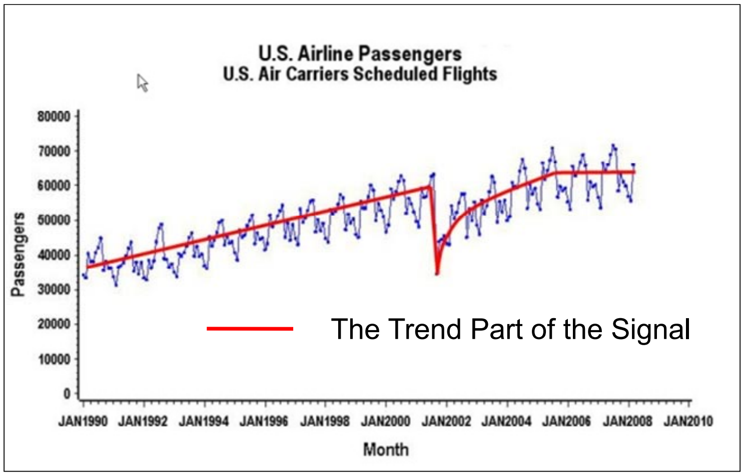
$$Y_t = \beta_0 + \beta_1 t + \varepsilon_t$$

A programming language formulation might look like the following:

TARGET=INTERCEPT+SLOPE*TIME+ERROR

The signal is represented by the INTERCEPT+SLOPE*TIME portion of the model and the ERROR (Greek epsilon) represents the noise component. In statistical linear regression analysis, the noise component is typically called the *error* component. Standard linear regression theory assumes that the error component has a Gaussian normal distribution. Furthermore, errors at different time points are assumed to be independent of each other. The assumption of independence and normality for the error component is called *white noise* error. (The term "white noise" derives from acoustic engineering.) Model formulation becomes a problem of identifying the mathematical form of the signal.

Time Series=Signal+Noise

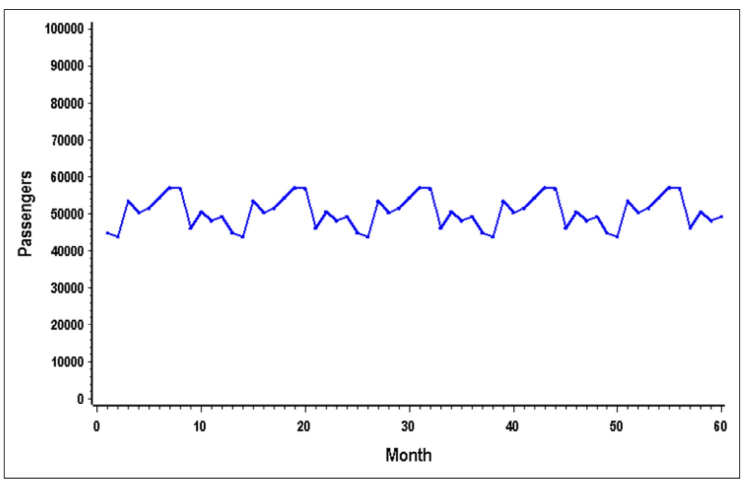


Signal=Systematic Variation (Predictable)

26

For the airline passengers data, as you saw, the signal is a combination of trend and seasonal components as well as regression components for events and economic conditions.

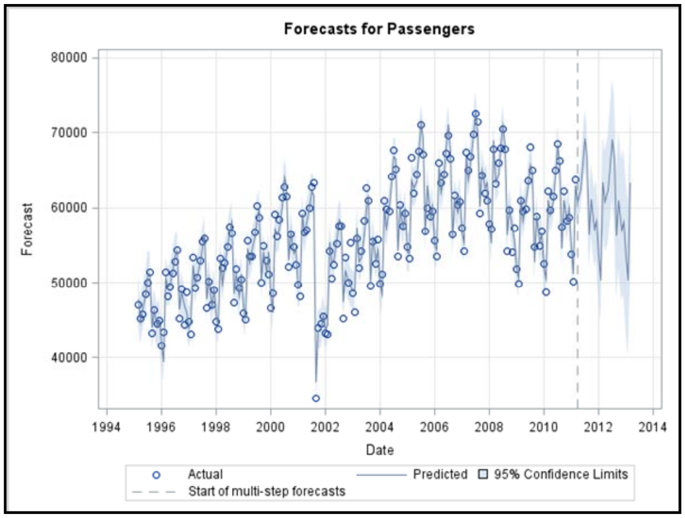
Time Series=Signal+Noise



The Seasonal Component of the Signal (5 Periods)

27

Forecast=Extrapolated Signal



28 Confidence Intervals Reflect Variation in Noise

Forecasting Using Statistical Models

Box-Jenkins Modeling Methodology

- IDENTIFY
  - Estimate and evaluate diagnostic functions.
  - Diagnose trend and seasonal components.
  - Select input variables and determine a dynamic relationship with the target variable.
- ESTIMATE
  - Derive estimates for model parameters.
  - Evaluate estimates and goodness-of-fit statistics.
- FORECAST
  - Derive forecasts of deterministic inputs.
  - Predict non-deterministic inputs.
  - Forecast the target variable.

29

1.2 Introduction to Forecasting with SAS Software

Objectives

- Outline the products and procedures in SAS that implement methodology for solving forecasting problems.
- Use sample time series data to exemplify forecasting concepts.

32

SAS Software for Forecasting

- Base SAS
  - DATA step programming
- SAS/STAT
  - Ordinary least squares regression
  - Regression with correlated errors
- SAS/ETS
  - Univariate and multivariate time series forecasting
  - Dynamic regression with correlated errors
  - Econometric modeling
  - Spectral analysis
  - time series cross-sectional modeling
- SAS Forecast Server
  - High performance large-scale forecasting

33 ...

SAS/ETS Procedures Used in This Course

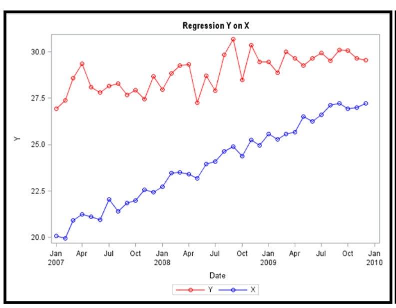
- TIMESERIES
  - Exploratory plots and statistics
  - Simple decomposition models
- FORECAST
  - Stepwise autoregression
  - Exponential smoothing (user-supplied weights, not covered)
- ARIMA
  - AutoRegressive Integrated Moving Average models
  - Dynamic regression models (transfer function models)
- AUTOREG
  - Simple regression models with autoregressive errors
  - ARCH and GARCH models (not covered)
- ESM – Exponential smoothing models
- SPECTRA – Spectral analysis

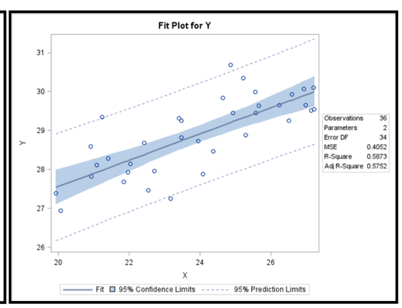
35

Regression of Y on X

Time Series at Time t: $Y_t = Y(t)$

Linear Regression Model: $Y_t = \beta_0 + \beta_1 X_t + \varepsilon_t$





36

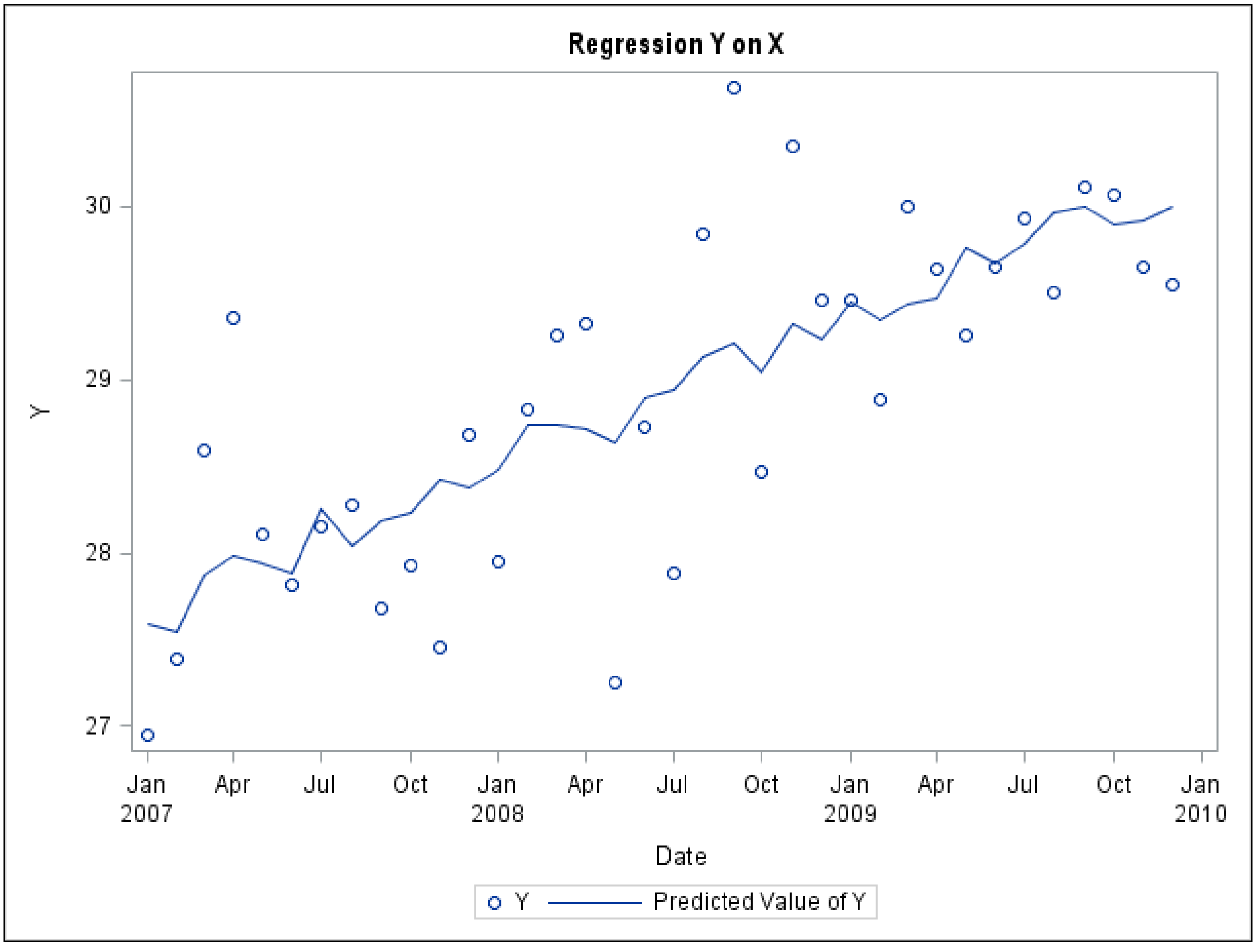
Many of the concepts of time series analysis can build on the concepts from regression analysis. With a simple regression model, you can use PROC REG in SAS/STAT software to fit the model to data. The above plots derive from a simulated set of data. With ODS Graphics turned on, the following code produces the plot of the regression curve:

```
proc reg data=work.tempyx;
   model Y=X;
   output out=work.regout p=Predicted;
quit;
```

In time series analysis, you want to view the data as a function of time. Because the data set is already sorted by the **Date** variable, you can use the following code to plot the regression curve as a function of time rather than of the input variable X:

```
proc sgplot data=work.regout;
   scatter x=Date y=Y;
   series x=Date y=Predicted;
run;
```

The predictions plotted with respect to time take a different shape.

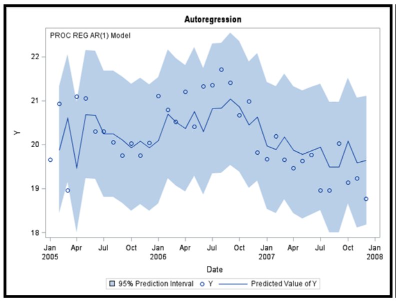


You can take what you know of regression analysis and add the time dimension to the analysis.

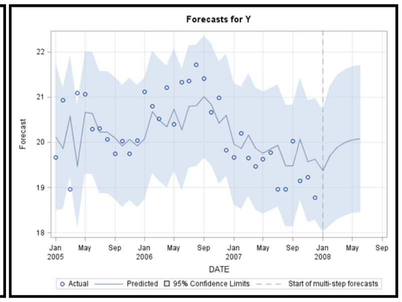
Regression of Y on Past Y: *Auto*regression

Autoregressive (Order 1) Model:

$$Y_t = \beta_0 + \beta_1 Y_{t-1} + \varepsilon_t$$



PROC REG



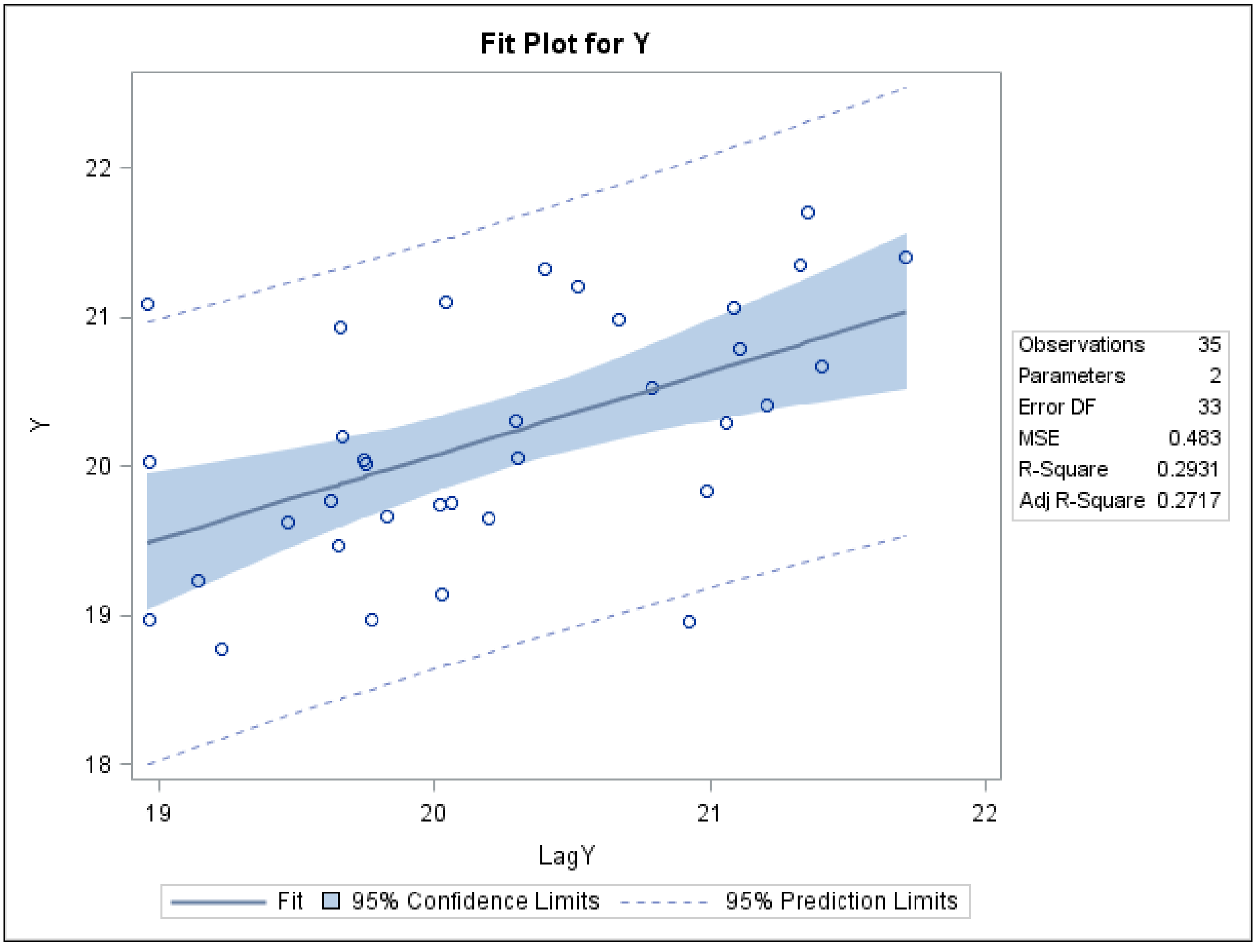
PROC ARIMA

37

Autoregression represents a time series as a function of its own past. You could use PROC REG to fit such models, but PROC REG cannot produce forecasts that are extrapolated in the time dimension. The following code reveals how to use PROC REG to fit an autoregressive model to data:

```
data work.ar1;
   set work.ar1;
   LagY=lag(Y);
   format Date date9.;
run;
proc reg data=work.ar1;
   model Y=LagY;
   output out=work.regout p=Predicted lcl=L95 ucl=U95;
quit;
```

You must add the LagY variable to the SAS table before using PROC REG. The prediction plot produced by PROC REG uses the independent variable, LagY, as the X-axis variable.



Again, viewing data using the time dimension becomes relevant. The following code produces the plot shown in the slide with title "Regression of Y on Past X: Autoregression":

```
proc sgplot data=work.regout;
   band x=Date lower=L95 upper=U95 /
        legendlabel="95% Prediction Interval";
   scatter x=Date y=Y;
   series x=Date y=Predicted;
   inset "PROC REG AR(1) Model";
run;
```

You can use PROC ARIMA to fit the same model, and you can get extrapolated forecasts. (The syntax and sample code for PROC ARIMA appears later.)

Time Series Models — Notation

Autoregressive Model of Order 1 = AR(1) Model:

$$Y_t = \phi_0 + \phi_1 Y_{t-1} + \varepsilon_t$$

Backshift Operator: $B(Y_t) = BY_t = Y_{t-1}$

Alternate Representation of AR(1) Model:

$$(1-\phi_1 B)(Y_t - \mu) = \varepsilon_t \qquad \phi_0 = \mu(1-\phi_1)$$

Mean: μ
Error Term: ε_t is "white noise."

38

The notation changes as you move from ordinary regression analysis to time series analysis. The Greek letters anticipate the use of regression coefficients in the general ***dynamic*** regression model. The shuffling of Greek letters can add confusion. Individual authors often select notation consistent with a more general subject matter area, such as engineering or economics. The notation employed in these course notes closely follows that of Box, Jenkins, and Reinsel (2008).

The use of backshift notation offers convenience with respect to model specification, and it facilitates examining theoretical issues in the context of properties of polynomials. For example, consider the polynomial shown below:

$\mathrm{f(x)} = 1 - \phi_1 x$,

This corresponds with the term $1-\phi_1 B$ in the AR(1) model. This polynomial has a zero at $x = 1/\phi_1$, that is, $f(1/\phi_1) = 0$. Later in the course you see that requiring the zeros (roots) of a polynomial to fall outside of the unit circle ensures a property called *stationarity*. This leads to the following algebraic conclusion:

$|1/\phi_1| > 1 \Rightarrow |\phi_1| < 1$

Thus, AR(1) models satisfying $|\phi_1| < 1$ are stationary. The type of stationarity that is discussed in this course requires that a time series have a constant mean and variance at all time points. Models with trend or seasonality are not stationary. (Stationary time series are the subject of Chapter 2.)

Naïve Models

Naïve Regression Model: $Y_t = \mu + \varepsilon_t$

Naïve Time Series Model: $Y_t = Y_{t-1} + \varepsilon_t$

$$R^2 = 1 - \frac{SSE}{SST},$$

$$SSE = \sum (Y_t - \hat{Y}_t)^2, \; \hat{Y}_t = \text{model prediction}$$

$$SST = \sum (Y_t - \overline{Y})^2 \quad \hat{\mu} = \overline{Y} = \text{naïve prediction}$$

39

A naïve model can be thought of as the model you would use if you had no statistical forecasting training. Realistically, many naïve models exist in practice. For example, a ratio model forecasts the sales for the next month by multiplying the sales for this month by the average change in sales over the past 12 months. If the company saw an average increase in sales of 5% over the last 12 months, and if this month's sales were $100,000, then the forecast for next month is 1.05*100,000=$105,000. This model might perform well on average, but it misses unique behavior by month, such as the increase in sales due to holidays.

You are probably familiar with *R square*. While it is usually described as a measure that shows "the percentage of variability in the data explained by the model," it can also be viewed as a measure that compares the model to a naïve model that has only a mean term and an error term. For certain linear regression models, ***all*** models are guaranteed to produce an error sum of squares (SSE) that is no larger than SST. The quantity SST is the numerator of the variance formula for the target variable, hence the "explained variation" interpretation of R square.

R square can be defined to be the squared correlation between the actual target variable and the predicted value produced by a model. This definition leads to the simplified formula given above when certain assumptions are met. When these assumptions are met, the calculated R-square value is guaranteed to be between zero and one. However, if you abandon the original definition and only use the above formula as the definition, then R square can actually be negative. This occurs when the predicted value produced by the model is actually ***worse*** than using the mean as the predicted value. The assumptions mentioned above prevent such models from being considered, but in time series analysis, the assumptions are often violated, and models that are worse than the naïve mean model can arise.

The mean model is not necessarily the best choice for a naïve model. In time series analysis, the naïve model is often taken to be a random walk model. A mean model uses the sample mean as the predicted value at all time points. The random walk model uses the previous observed value as the predicted value for the current time point.

The Random Walk Model

Random Walk: $Y_t = Y_{t-1} + \varepsilon_t$

$$R^2_{RW} = 1 - \frac{SSE}{SSRW},$$

$$SSE = \sum (Y_t - \hat{Y}_t)^2, \qquad \hat{Y}_t = \text{model prediction}$$

$$SSRW = \sum (Y_t - Y_{t-1})^2 \; Y_{t-1} = \text{naïve prediction}$$

40

The Random Walk R square is negative when a model produces predictions that are worse than random walk predictions.

The discussion of naïve models relates to a general philosophical consideration in forecasting. Some organizations say that they never engaged in forecasting in the past when they enter a new world of analytics. However, if an organization manufacturers 1,000 widgets, then it is forecasting that it will sell 1,000 widgets. The number 1,000 came from somewhere. It might be a number thought up by a product manager, which is called a *judgmental forecast*. It might come from an antiquated software system with hardcoded model parameters that were not estimated from real data for years. Wherever the number comes from, it becomes the naïve forecast that the new sophisticated forecasting system must beat. If you cannot define or quantify your naïve forecasts, then the random walk provides a convenient benchmark. Gilliland (2010) provides an excellent nontechnical discussion of naïve models and benchmarking.

To introduce concepts and terminology relevant for forecasting, the naïve random walk model, the autoregressive order 1 model, and the moving average order 1 model are used.

Three Simple Time-Dependent Models

Random Walk with Drift (RWD)

$$Y_t = \phi_0 + Y_{t-1} + \varepsilon_t$$

First Order Autoregression AR(1)

$$Y_t = \phi_0 + \phi_1 Y_{t-1} + \varepsilon_t$$

First Order Moving Average MA(1)

$$Y_t = \theta_0 + \theta_1 \varepsilon_{t-1} + \varepsilon_t$$

41

Notice that if $\phi_1 = 1$, then the autoregressive model simplifies to the random walk with drift model. If the drift parameter is zero, $\phi_0 = 0$, and if $\phi_1 = 1$, then the model simplifies to the naïve random walk model. The error component is white noise.

The simple AR(1) model is a special case of the general autoregressive order *p* model, denoted AR(*p*).

$$Y_t = \phi_0 + \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + \cdots + \phi_p Y_{t-p} + \varepsilon_t$$

The AR(*p*) model provides the foundation for several diagnostic functions used to identify forecast models. Similarly, the general MA(q) model is given by the following:

$$Y_t = \theta_0 + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + \cdots + \theta_q \varepsilon_{t-q} + \varepsilon_t$$

Random Walks and Differencing

Random Walk (RW)

$$Y_t = Y_{t-1} + \varepsilon_t$$

Random Walk with Drift

$$Y_t = \phi_0 + Y_{t-1} + \varepsilon_t$$

42

You can rewrite the above equations as difference equations.

Random Walks and Differencing

Random Walk:

$$Y_t - Y_{t-1} = \varepsilon_t$$

$$\Delta Y_t = Y_t - Y_{t-1} \longleftarrow \text{First Difference}$$

$$BY_t = Y_{t-1} \longleftarrow \text{Backshift Operator}$$

$$\Delta Y_t = (1-B)Y_t$$

43

PROC ARIMA accommodates differencing as part of the modeling process.

ARIMA Procedure

```
PROC ARIMA DATA=SAS-data-set <options>;
      BY variables;
      IDENTIFY VAR=variable CROSS=(variables)
                    NLAGS=n <options>;
      ESTIMATE P=n Q=n
                     INPUT=(variables)
                     METHOD=CLS|ML|ULS <options>;
       FORECAST OUT=SAS-data-set
                       ID=variable INTERVAL=interval
                       LEAD=n <options>;
RUN;
QUIT;
```

44

The design of PROC ARIMA is based on the pioneering work of Box and Jenkins (1976). They proposed a three-step approach to forecasting:

- Identify
- Estimate
- Forecast

The Box-Jenkins approach was outlined in the previous section.

The VAR= option identifies the target variable to be forecast. The CROSS and INPUT options are for dynamic regression models. The keyword CROSS is an accepted shorthand version of CROSSCORR, for CROSS CORRelations. The P= option identifies the order of the autoregression employed for forecasting. The Q= option specifies the order of the moving average component. METHOD=ML requests maximum likelihood estimation, and can be shortened to ML. The LEAD= option indicates how many time units to predict into the future. If ID= and INTERVAL= options are supplied, PROC ARIMA extends the ID variable into the future as well. The OUT= option identifies the name of the data set that contains forecasts. The NLAGS= option specifies the number of lags to use.

The AMR stock price data set was introduced earlier. The code in the next slide fits an autoregressive order 1 model to the time series **LogVolume**.

Forecast AMR Stock Volume

```
proc arima data=work.amrdaily
   plots(only)=(forecast(forecast));
   identify var=LogVolume;
   estimate p=1 ml; /*ML => METHOD=ML*/
   forecast lead=15 out=work.outAR1
            id=Date interval=weekday;
quit;
```

45

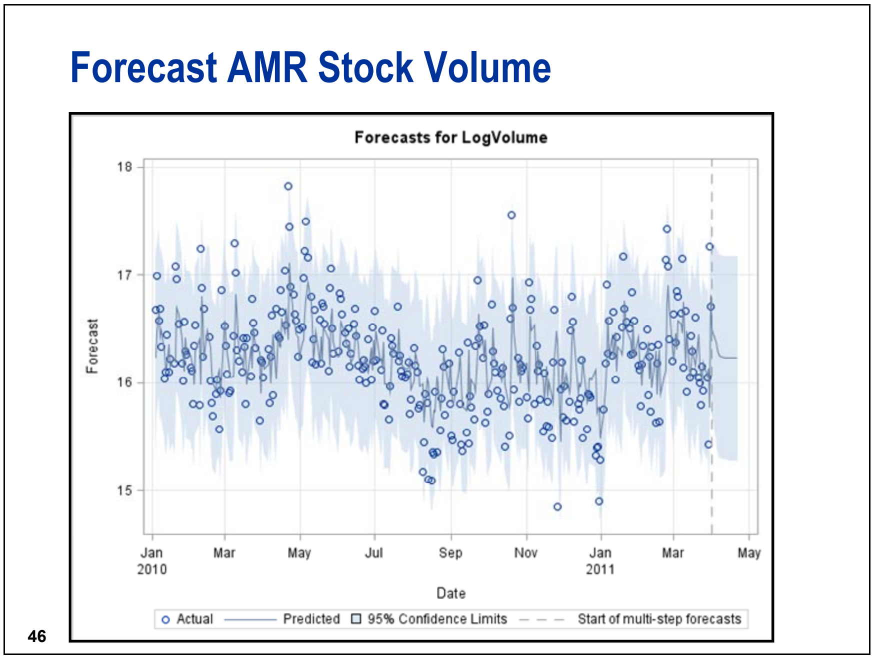
Special data handling is required with stock market data because the market is closed on weekends and certain holidays. You could treat all the open dates to be contiguous in terms of modeling and ignore all other dates. This approach is valid as long as you only use a model that does not depend on the weekly cycle of five business days, or any other cycle. You could define a time index variable **TIME** representing "trading day" where **TIME** is missing (.) on closed dates and increments by 1 otherwise. The lag 1 value of volume is taken to be the volume from the most recent *trading* day. This approach is common, and perhaps more "honest" than the approach presented below.

The AMR data that was created for this course was read directly from data obtained from the Internet as described previously. As is typical with stock data, there are some anomalies in the original data. Most data vendors adjust for splits in the stock, so that historic values are relative to the current stock price adjusted for splits. Also, data vendors can differ in how stock prices and volume are represented on days that the stock exchanges are closed. The data obtained for this course inconsistently treats closed dates, with the most recent data showing a volume of zero on closed dates, whereas older data is missing observations for closed dates. To make sure that the data is equally spaced with respect to the SAS weekday specification, you can use PROC EXPAND.

The following code adds weekday dates that were not in the original data and sets the stock price and volume values to missing:

```
proc expand data=sasuser.amrdaily
            out=work.amrdaily
            from=weekday
            align=beginning
            method=none
            ;
   id Date;
   convert Volume;
   convert LogVolume;
run;
```

When PROC ARIMA encounters a missing value, it assumes that the value is missing completely at random and attempts to forecast the missing value. Consequently, the forecast data has nonmissing forecasts for historic dates when the markets were closed. Clearly, these forecasts are invalid, because, for example, the volume should be exactly zero on those dates. While it might seem counter-intuitive to recommend such an approach, treating closed dates as missing values actually has little or no negative impact on the accuracy of the forecasts, and by not deleting the dates, equal spacing is preserved, and SAS date values can be used to label future forecasts. Furthermore, if cyclic variation exists, this methodology preserves the cycles, for example, a five day weekday cycle. Data post-processing is required, because you have to identify forecasts for closed dates and set them to zero or missing.



PROC ARIMA uses the AR(1) model to derive forecasts. The stock market is closed on the Friday before the Easter holiday, so the forecast of April 22, 2011 would have to be changed. Because LEAD=15 was used, the forecasts only go to April 21, 2011.

AR(1) forecasts have one arbitrary value followed by an exponential decrease toward the series mean. The arbitrary value and rate of decrease (AR parameter) are determined by the fitted model parameters, which in turn are determined by the data. Eventual mean reversion is characteristic of all stationary time series, which is a concept that is detailed later.

Mean reversion is sometimes viewed as a disappointment, rendering long term forecasts “uninteresting.” Notice, however, that forecasts can be updated daily as new data comes in. It is not necessary to stick with a long-term forecast computed from old data when it is simply a matter of data entry and rerunning a program to update the forecasts.

Forecast AMR Stock Volume

Maximum Likelihood Estimation					
Parameter	Estimate	Standard Error	t Value	Approx Pr > |t|	Lag
MU	16.22641	0.05005	324.20	<.0001	0
AR1,1	0.55754	0.04688	11.89	<.0001	1

Constant Estimate	7.17947
Variance Estimate	0.160525
Std Error Estimate	0.400656
AIC	319.8894
SBC	327.3754
Number of Residuals	312

$$\text{Model}: Y_t = 7.17947 + 0.55754 Y_{t-1} + \varepsilon_t$$

$$\text{Forecast Equation}: \hat{Y}_t = 7.17947 + 0.55754 Y_{t-1}$$

47

The mean form with the backshift operator of the above model is shown below:

$$(1 - 0.56612\mathrm{B})(\mathrm{Y}_t - 16.22279) = \varepsilon_t$$

The backshift operator can be applied to produce the following:

$$(\mathrm{Y}_t - 16.22279) = 0.56612(\mathrm{Y}_{t-1} - 16.22279) + \varepsilon_t$$

One way to validate this form of the model is too look for the model specification produced by the FORECAST statement.

Model for variable LogVolume	
Estimated Mean	16.22279

Autoregressive Factors	
Factor 1:	1 - 0.56612 B**(1)

Notice that **Constant Estimate** was labeled ϕ_0 in the AR(1) model definition presented earlier. The constant satisfies the following:

$$\phi_0 = \mu(1 - \phi_1)$$

Substituting the estimated values, you obtain this formula:

$$\hat{\phi}_0 = \hat{\mu}(1 - \hat{\phi}_1) = 16.22279(1 - 0.56612) = 7.038818$$

This is the value given in the output.

The **Variance Estimate** is the prediction error variance estimate, that is, the estimated variance of the error term ε_t.

The model is for **LogVolume**. The corresponding prediction equation for **Volume** is as follows:

PREDICTED=EXP(FORECAST)

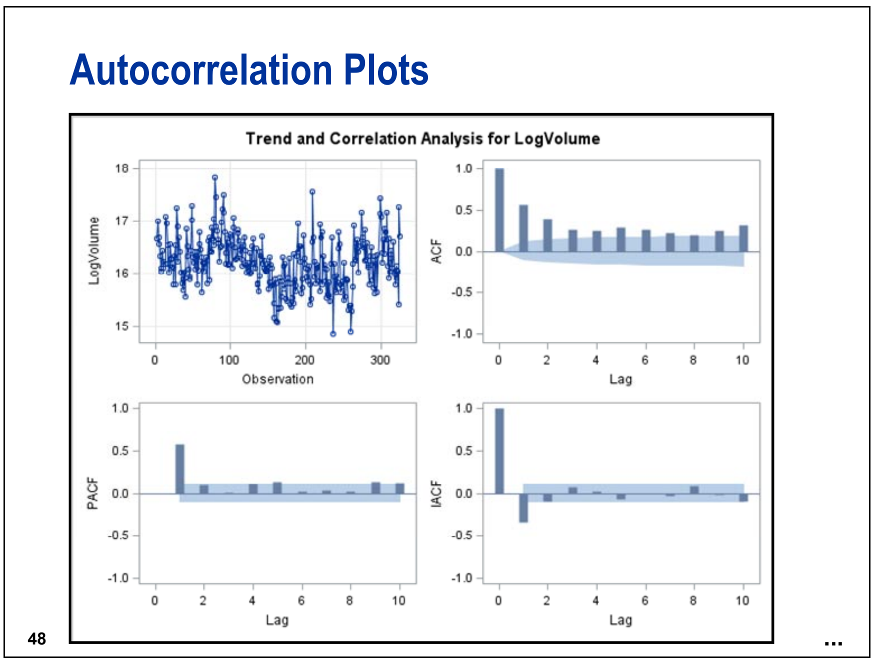
This represents the median prediction. If the logarithm of a random variable has a normal distribution, then the original variable is said to have a lognormal distribution. It can be shown that if the mean of the logarithm of the variable is MU and the variance is SIGMA squared, then the mean of the original variable is the following:

MEAN=EXP(MEAN+0.5*SIGMA**2)

For forecasting lognormal data, the predicted value is calculated as follows:

PREDICTED=EXP(FORECAST+0.5*STD*STD)

STD is the prediction standard deviation produced by PROC ARIMA, and FORECAST is the forecast value produced by PROC ARIMA for the logarithm of the original series.



The choice of model depends on the characteristics of the time series. Three sample functions provide insight into what models might produce good forecasts for the series. These are indicated in the plots in the slide above as the ACF (AutoCorrelation Function), the PACF (Partial AutoCorrelation Function), and the IACF (Inverse AutoCorrelation Function). The plot was produced using the following PROC ARIMA statement:

```
identify var=LogVolume nlags=10;
```

As you learn about the characteristic shapes of these three functions for forecast models, you will learn that the above shapes are characteristic of two models, one of which is an AR(1) model. (The other is an ARMA(1,1) model which is introduced later.)

The Autocorrelation Function (ACF)

- The autocorrelation function (ACF) measures the dependence among observations in a time series.
- The autocorrelation at lag *k* is the correlation of observations *k* times units apart.
- The sample autocorrelation function estimates the unknown population autocorrelation function.
- The sample autocorrelation function is a biased estimator.
- The sample ACF can identify features that should be included in a forecast model.
- The sample ACF can be used to provide estimates of forecast model parameters.

50

The ACF was thoroughly studied for autoregressive models. For example, it can be shown that the population autocorrelation function for a pure AR(1) process is given by this formula:

$$\mathrm{ACF}(k) = \mathrm{Corr}(Y_t, Y_{t-k}) = \phi_1^k$$

The parameter ϕ_1 is the coefficient for the lag-1 term in the AR(1) model. Given an observed time series $y_1, y_2, \ldots, y_n$, calculate the sample ACF as follows:

$$r_k = \frac{\sum_{t=1}^{n-k}(y_t - \bar{y})(y_{t+k} - \bar{y})}{\sum_{t=1}^{n}(y_t - \bar{y})^2}, \quad \bar{y} = \frac{1}{n}\sum_{t=1}^{n} y_t$$

Interpreting the ACF

- If the population ACF(1)>0, then a time series value is affected by the previous time series value and will tend to be "close" to the previous value.
- If the population ACF(1)>0, then, for example,
  - February values will be affected by January values, March values will be affected by February values, April values will be affected by March values, and so on.
  - Tuesday values will be affected by Monday values, and so on.
  - 2000 affects 2001, 2001 affects 2002, and so on.
- If the sample ACF(1)>0 and the sample ACF(1) is above the upper 95% confidence interval, then evidence suggests that the population ACF(1)>0.

51

continued...

Interpreting the ACF

- If the population ACF(1)<0, then a time series value is affected by the previous time series-value and will tend to be "distant" from the previous value. The previous value "pushes" the current value away so that it will tend to ***not*** be close.
- If the population ACF(1)<0, and if January is "high," then February will tend to be "low," and if February is "low," then March will tend to be "high," and so on.
- If the sample ACF(1)<0 and the sample ACF(1) is below the lower 95% confidence interval, then evidence suggests that the population ACF(1)<0.

52

continued...

Interpreting the ACF

- ACF(1) is called first-order autocorrelation, lag-1 autocorrelation, span-1 autocorrelation, or *serial correlation*. (Other names are also used.)
- The Durbin-Watson test statistic is one of several tests that can be used to test the null hypothesis that the population ACF(1)=0 against one-side or two-side alternatives. The Durbin-Watson test is standard in most commercial software for regression analysis. (For example, see PROC REG in SAS/STAT software.)

53

continued...

Interpreting the ACF

- If population ACF(*k*)>0, then a time series value is affected by the time series value *k* time units in the past and will tend to be "close" to this value.
- If the sample ACF(*k*)>0 and the sample ACF(*k*) is above the upper 95% confidence interval, then evidence suggests that the population ACF(*k*)>0.
- If population ACF(*k*)<0, then a time series value is affected by the time series value *k* time units in the past and will tend to be "distant" from this value.
- If the sample ACF(*k*)<0 and the sample ACF(*k*) is below the lower 95% confidence interval, then evidence suggests that the population ACF(*k*)<0.

54

continued...

Interpreting the ACF

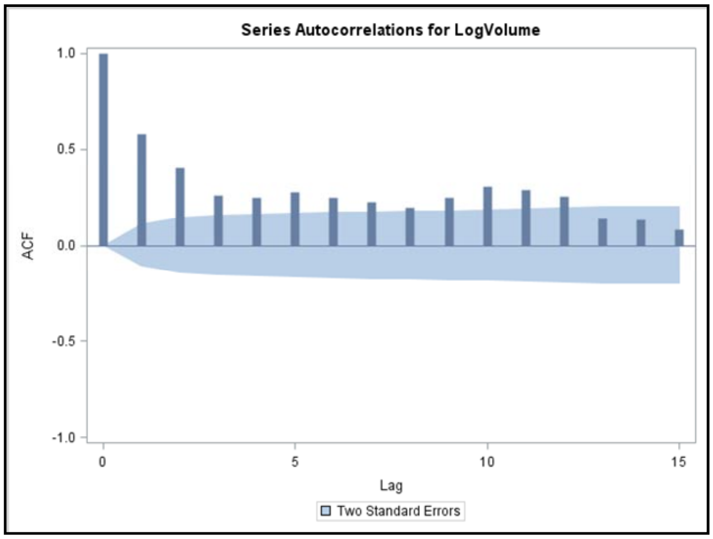
- The choice of 95% confidence interval for the sample ACF is arbitrary.
- A statistically significant result can be *spurious*. One cause of a spurious result is the *proximity effect*. Another cause of a spurious result is multiple testing, that is, testing many hypotheses about ACF(*k*) at the 5% level tends to produce, on average, 5% spurious results when there is no autocorrelation at the lags that are tested.

55 *continued...*

Interpreting the sample ACF would be straightforward if not for spurious results. These spurious results might actually be legitimate values that would be expected theoretically, but they hamper interpretation of the ACF with respect to the behavior of the time series.

The proximity effect can be explained using an example. Suppose the true population autocorrelation function is zero at all lags except lag 1. Suppose ACF(1) is large, for example, approximately 0.9. Then adjacent series values will be highly correlated. If January is large, then February also tends to be large. If February is large, then March also tends to be large. The sample ACF sees a large January value followed by a large March value and concludes that ACF(2) is positive, when in fact it is actually zero. The values that are two months apart appear to be correlated because of the strong correlation for values one month apart. The “proximity” of January and March to February causes the spurious result.

Interpreting the ACF



56

The Partial Autocorrelation Function (PACF)

- The partial autocorrelation function at lag k adjusts for all previous autocorrelations at lags 1, 2, …, k-1.
- The sample partial autocorrelation function estimates the population partial autocorrelation function.
- Because lag 1 has no previous lags, ACF(1)=PACF(1).
- The sample partial autocorrelation function helps determine whether significant lags in the sample autocorrelation function might due to the proximity effect.

57

The sample partial autocorrelation function (PACF) is calculated using the Yule-Walker equations, which are derived from treating the autoregressive model as an ordinary regression model.

The theoretical centered regression of Y_t on Y_{t-1} and Y_{t-2} has coefficients β_1 and β_2, which is given by solving an equation involving the autocorrelations at lag *k*, ρ_k =PACF(*k*), as follows:

$$\begin{pmatrix} 1 & \rho_1 \\ \rho_1 & 1 \end{pmatrix} \begin{pmatrix} \beta_1 \\ \beta_2 \end{pmatrix} = \begin{pmatrix} \rho_1 \\ \rho_2 \end{pmatrix}$$

This resolves to a system of two equations and two unknowns.

$$\beta_1 + \beta_2 \rho_1 = \rho_1$$
$$\beta_1 \rho_1 + \beta_2 = \rho_2$$

The solution for this system of equations, solving for the β_2 as a function of the ρs, is as follows:

$$\beta_2 = \frac{\rho_2 - \rho_1^2}{1 - \rho_1^2}$$

The value of this calculation derives from the fact that PACF(2)= β_2. Systems of equations such as this exist for any autoregression. For example, if you regress Y_t on Y_{t-1}, Y_{t-2}, and Y_{t-3} to get coefficients β_1, β_2, and β_3, then PACF(3)= β_3.

It is easy to see that PACF(1)= $\beta_1 = \rho_1$ =ACF(1) is obtained from regressing Y_t on Y_{t-1}.

The PACF is only the highest order parameter obtained from the sequence of autoregressions. The Yule-Walker equations write the system of equations in recursive form so that you can solve first for PACF(1), then for PACF(2), and so on, in an efficient recursive fashion. The sample PACF uses the sample ACF and substitutes these for the population ACF values in the formulas.

The regressions that are described are autoregressions, so in fact an estimated AR(1) model produces PACF(1), an estimated AR(2) model produces PACF(2), and so on. The Yule-Walker equations enable you to get an estimate for the highest order autoregressive parameter to use as the PACF value.

The use of parameters $\beta_1, \beta_2, \ldots$ was used for comparison to regression models. Textbooks about time series analysis often use parameters $\pi_1, \pi_2, \ldots$, when defining the PACF. To emphasize the use of multiple models of increasing order, you will also see an autoregressive model with *k* parameters represented by the following:

$$Y_t = \phi_{k0} + \phi_{k1} Y_{t-1} + \phi_{k2} Y_{t-2} + \cdots + \phi_{kk} Y_{t-k} + \varepsilon_t$$

The PACF is defined to be $\phi_{11}, \phi_{22}, \ldots, \phi_{kk}, \ldots$.

There are statistical considerations when you apply the PACF to forecasting problems. Notice that an AR(2) model, which can be written as shown below:

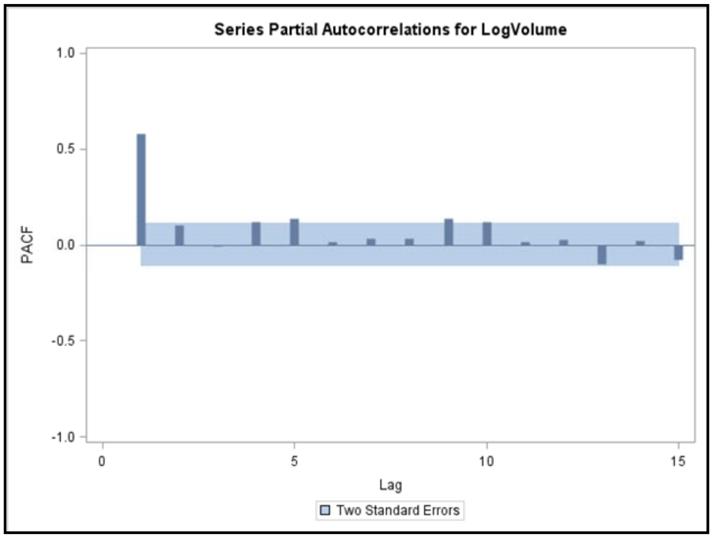
$$(Y_t - \mu) = \phi_{20} + \phi_{21}(Y_{t-1} - \mu) + \phi_{22}(Y_{t-2} - \mu) + \varepsilon_t$$

is the same as the AR(3) model, shown here:

$$(Y_t - \mu) = \phi_{30} + \phi_{31}(Y_{t-1} - \mu) + \phi_{32}(Y_{t-2} - \mu) + 0(Y_{t-3} - \mu) + \varepsilon_t$$

That is, an AR(3) model with $\phi_{33} = 0$. The theoretical regression with one lag (Y_t on Y_{t-1}) gives a nonzero coefficient ϕ_{11}, which equals the autocorrelation ρ_1, but does not equal ϕ_{21} unless ϕ_{22} is 0 and the model becomes only an AR(1). The regression of Y_t on Y_{t-1} and Y_{t-2} gives the lag 2 partial autocorrelation coefficient as its last coefficient, which ***is*** equal to ϕ_{22}, and $\phi_{22} = \phi_{32}$ because $\phi_{33} = 0$. Because the third coefficient is zero, $\phi_{33} = 0$, and all higher order coefficients will be zero for AR(4), AR(5), and all subsequent AR models, then the PACF will be zero at lags 3, 4, and so on. When the regression is run with estimated correlations substituted for theoretical ones, you can only expect to get within a statistical variation of zero. It is proven that the estimates so derived are approximately normal in large samples with variances approximately $1/n$, the reciprocal of the sample size, from which 95% error bands around zero can be computed to help judge the number of autoregressive lags *p* needed in a pure autoregressive AR(*p*) model. The last nonzero PACF yields the order of the autoregressive model.

Interpreting the PACF



58

For the AMR stock volume data, the PACF for **LogVolume** obtained using ODS Graphics shows significant bars, that is, bars that extend beyond the 95% confidence band. For convenience, such significant bars are called *spikes* to differentiate them from bars that do not extend beyond the confidence bands. Spikes can be seen at lags 1, 4, 5, 9, and 10.

A more primitive plot of the PACF is available if you turn off ODS graphics. Use the following statement to do so:

```
ods graphics off;
```

To turn ODS graphics on, use the keyword **ON** instead of **OFF**.

The following plot is called an ASCII plot or a “printer plot” because it can be produced without high resolution plotting hardware, for example, with an old impact printer.

```
                              AMR Daily Stock Volume

                               The ARIMA Procedure

                             Partial Autocorrelations

        Lag    Correlation    -1 9 8 7 6 5 4 3 2 1 0 1 2 3 4 5 6 7 8 9 1

          1        0.57650    |                   . |************        |
          2        0.10021    |                   . |**                  |
          3       -0.00976    |                   . | .                  |
          4        0.11430    |                   . |**                  |
          5        0.13570    |                   . |***                 |
          6        0.01551    |                   . | .                  |
          7        0.02976    |                   . |*.                  |
          8        0.03241    |                   . |*.                  |
          9        0.13250    |                   . |***                 |
         10        0.11609    |                   . |**                  |
         11        0.01677    |                   . | .                  |
         12        0.02354    |                   . | .                  |
         13       -0.10201    |                  **| .                  |
         14        0.02284    |                   . | .                  |
         15       -0.07431    |                  .*| .                  |
```

The advantage of the ASCII plot is that the actual values of the PACF can be seen. The marginal results at lags 2 and 13 are relatively close to the significant results at lags 4 and 10.

Judgment should be used in interpreting the PACF. Large values indicate lags that should be considered for a pure AR model. Lags whose correlations extend slightly past the two standard error bands at small lags should be considered. Even those correlations near the bands might be considered if they are at low lags (for example, at lags 1 or 2) or if they are at sensible lags such as lag 24 in hourly data, lag 4 in quarterly data, and so on. Here the lag 13 correlation -0.10201 is at too large a lag. The volume, 13 trading days ago, seems an unlikely predictor.

✎ One in 20 (5%) of the estimates are expected to cross the bands only by chance (the definition of 95% intervals).

While the ACF is often displayed and viewed first, the insight obtained from one correlation plot often needs to be weighed against the evidence produced by the other plots. Armed with the above insight, you should revisit the ACF plot. The low resolution version is shown next.

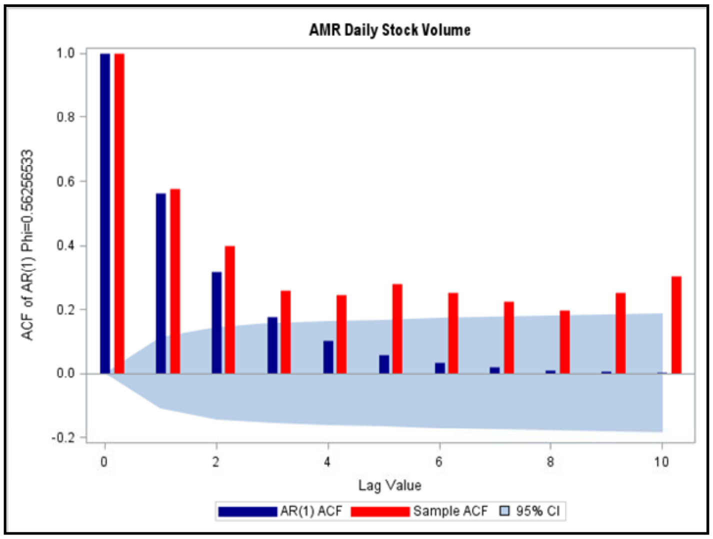
```
                              AMR Daily Stock Volume

                              The ARIMA Procedure

                          Name of Variable = LogVolume

          Mean of Working Series                        16.22554
          Standard Deviation                             0.47795
          Number of Observations                             324
          Embedded missing values in working series           12

                                Autocorrelations

      Lag  Covariance  Correlation  -1 9 8 7 6 5 4 3 2 1 0 1 2 3 4 5 6 7 8 9 1

        0    0.228436      1.00000  |                    |********************|
        1    0.131693      0.57650  |                  . |************        |
        2    0.091205      0.39926  |                 .  |********            |
        3    0.059033      0.25842  |                 .  |*****               |
        4    0.056252      0.24625  |                 .  |*****               |
        5    0.063764      0.27913  |                 .  |******              |
        6    0.057282      0.25076  |                 .  |*****               |
        7    0.051332      0.22471  |                .   |****                |
        8    0.044784      0.19605  |                .   |****                |
        9    0.057181      0.25031  |                .   |*****               |
       10    0.069138      0.30266  |                .   |******              |
       11    0.065444      0.28648  |                .   |******              |
       12    0.058153      0.25457  |                .   |*****               |
       13    0.031955      0.13989  |                .   |***.                |
       14    0.030663      0.13423  |                .   |***.                |
       15    0.019304      0.08450  |                .   |** .                |

                          "." marks two standard errors
```

You can argue that an AR(1) model is suggested by the PACF plot, although other models are also suggested. Knowledge of the theoretical finding presented before is of value when you examine the ACF plot. Recall that a pure AR(1) process produces an ACF such as the following:

$$\mathrm{ACF}(k) = \phi_1^k$$

This corresponds to a plot with exponentially decaying bars, given that the AR(1) coefficient must be between -1 and 1. You can see that for the first few lags, the ACF is dropping off at a rate that looks somewhat exponential, consistent with the AR(1) diagnosis from the PACF.

Sample versus Theoretical AR(1) ACF



60

A second type of simple model, the moving average of order 1, MA(1), predicts the current Y_t from the past error term, that is, from the part of the most recent past observation that was not anticipated by the model. The model for an MA(1) is as follows:

$$Y_t = \mu + \varepsilon_t - \theta_1 \varepsilon_{t-1}$$

Y_t is predicted as μ minus θ_1 times the most recent residual with μ and $\theta_1$replaced, of course, by estimates in practice. The covariance between Y_t and Y_{t-j} is the expected value, that is, the long run average value, of the product $(Y_t - \mu)(Y_{t-j} - \mu)$, which is the expected value of $(\varepsilon_t - \theta_1\varepsilon_{t-1})(\varepsilon_{t-j} - \theta_1\varepsilon_{t-j-1})$. Because the εs are independent, the covariance is 0 unless some subscripts match in the factors of $(\varepsilon_t - \theta_1\varepsilon_{t-1})(\varepsilon_{t-j} - \theta_1\varepsilon_{t-j-1})$. For the MA(1) model, this happens only when j=0 (that covariance is the variance of Y_t), or when j=1. The ACF is the covariance divided by the variance of Y_t. For j>1, that is, when j exceeds the moving average order (1), the ACF will be 0 because its numerator, the covariance at lag j, is 0. This then gives a way to identify the number q of moving average lags in a pure moving average model. For example, the model below:

$$Y_t = \mu + \varepsilon_t - \theta_1\varepsilon_{t-1} - \theta_2\varepsilon_{t-2}$$

is called a moving average of order 2. For this model, the covariance at lag j, the expected value of the following:

$$(\varepsilon_t - \theta_1\varepsilon_{t-1} - \theta_2\varepsilon_{t-2})(\varepsilon_{t-j} - \theta_1\varepsilon_{t-j-1} - \theta_2\varepsilon_{t-j-2})$$

has matching subscripts only if j=0, j=1 or j=2. Beyond the moving average order 2, the covariance and hence the ACF are 0. The ACF seems to preclude an MA(1) model but for illustration (of a poorly fitting model), an MA(1) model is used.

An MA(1) Model for the AMR Data

Maximum Likelihood Estimation					
Parameter	Estimate	Standard Error	t Value	Approx Pr > |t|	Lag
MU	16.22473	0.03348	484.61	<.0001	0
MA1,1	-0.42619	0.05291	-8.06	<.0001	1

Constant Estimate	16.22473
Variance Estimate	0.17681
Std Error Estimate	0.420488
AIC	349.022
SBC	356.508
Number of Residuals	312

Model for variable LogVolume	
Estimated Mean	16.22473

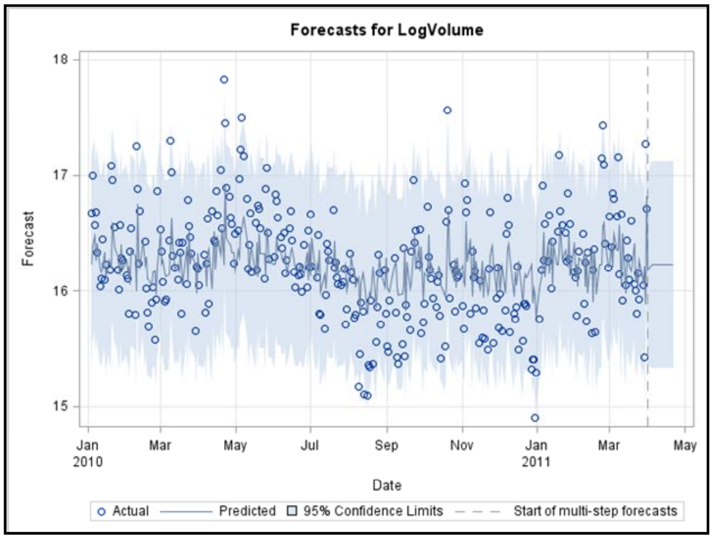
Moving Average Factors	
Factor 1:	1 + 0.42619 B**(1)

$$\text{Model}: Y_t = 16.22473 + \varepsilon_t + 0.42619\varepsilon_{t-1}$$

$$\text{Forecast Equation}: \hat{Y}_t = 16.22473 + 0.42619(Y_{t-1} - \hat{Y}_{t-1})$$

61

An MA(1) Model for the AMR Data



62

The last observation (trading day t=325) was slightly below its prediction leaving a small negative estimated error ε_{325}. As the target time t increases to 326 and t-1 becomes 325, the small 325th residual is multiplied by 0.42619 and added to the estimated mean 16.22473 to forecast the log(volume) for trading day 326. Because t=326 is one step into the future, there is no actual value there and hence no residual to estimate the error that is needed for forecasting time t=327. If you had to guess a future value, it would be your forecast, so value minus forecast becomes only forecast minus forecast, which is 0, that is, you replace any errors in the future by 0. Multiplying this (0) by 0.42619 and adding to the estimated mean 16.22473 gives the mean 16.22473 as the time t=327 forecast. The t-1=326 error is estimated to be 0 as will happen for all future ts from there on. The future errors are uncorrelated with anything that thus far occurred and come from a distribution with mean zero. This is the nature of an MA(1) forecast. There is one interesting forecast followed by forecasts equal to the mean. In a moving average of order 2, two lagged errors are involved and two interesting forecasts are produced followed by forecasts equal to the series mean.

Moving averages of order 2, 3, and so on, are obvious extensions and are discussed later. For now, notice that an MA(q) is identified by the ACF dropping to 0 after the appropriate lag q. PACF (and IACF) functions are not very useful for identifying moving average models. Forecasts from MA(q) models have q interesting forecasts followed by forecasts equal to the series mean.

Another identifying function is called the *Inverse Autocorrelation Function* (IACF). The IACF is the ACF of a different model called the *inverse* or *dual* model. This dual model is defined using the backshift operator. The idea is to switch the moving average (ε_t) backshift operator, if any, to the autoregressive (Y_t) side and the autoregressive (Y_t) backshift operator, if any, to the moving average (ε_t) side. As an example, the AR(1) model shown below:

$$(1-0.8B)(Y_t-10)=\varepsilon_t$$

has this dual model:

$$(Y_t-10)=(1-0.8B)\varepsilon_t$$

This is an MA(1) model whose ACF has thus only one nonzero lag correlation at lag 1. Remembering that the ACF of this dual, MA(1), model is the IACF of the original Y_t series, you switch the MA(1) diagnosis to AR(1). In general the strategy is as follows:

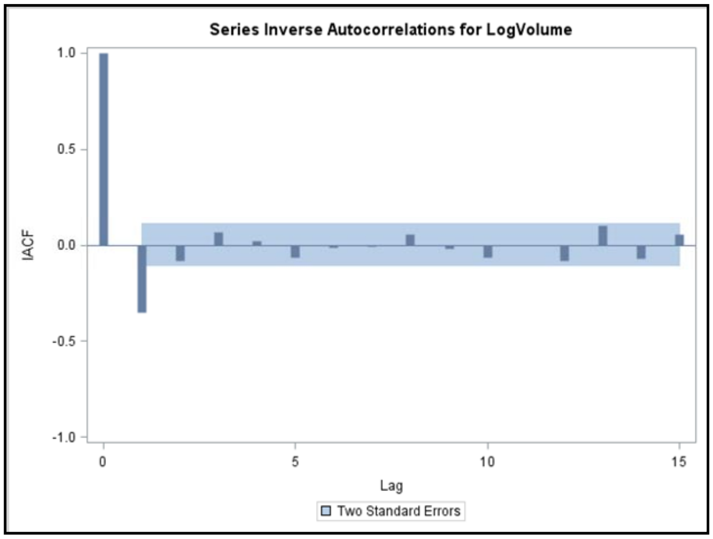
1. Diagnose the IACF as if it were an ACF.
2. Switch the operators so that the AR operators move to the MA side and the MA operators move to the AR side.

The Inverse Autocorrelation Function (IACF)

- The inverse autocorrelation function at lag k adjusts for all previous autocorrelations at lags 1, 2, …, k-1.
- The sample inverse autocorrelation function estimates the population inverse autocorrelation function.
- Typically IACF(k) is opposite in sign to PACF(k).
- The inverse autocorrelation function can help to identify model components when the PACF is ambiguous, and vice versa.
- The inverse autocorrelation function does not have a specific sample formula in that it requires the user to provide an approximating model for the data. Different users (and different software) can choose different approximating models.

63

Interpreting the IACF



64

Notice that an AR(2) has dual model MA(2), AR(3) has dual model MA(3), and so on, so all pure autoregressive models have an IACF that drops to 0 after the appropriate autoregressive lag just as the PACF does. Both are attempts to give the same information. As with the PACF, the IACF is not very helpful in general for moving average identification. For the special case of an MA(1), however, the IACF is the ACF of an AR(1) so if the IACF dies off exponentially , the dual model is AR(1) and thus the original data satisfies an MA(1) model. Anything you know about an ACF pattern does double duty. If you see a familiar IACF pattern, diagnose it as if it were an ACF, and then switch the operators.

The sample IACF is not well defined algorithmically, meaning that two different software vendors can produce two different IACF tables for the exact same data set. This occurs because this discussion of the IACF referred to a model, but when the IACF is calculated, presumably there is no candidate model, so you are using a tool to diagnose a model that requires you to specify a model. The dilemma is resolved by using what is called an *approximating autoregressive model*. There are several useful criteria for automatically selecting the order p of an approximating AR(p) model. These include Parzen's CAT criteria, Akaike's A information criteria (AIC), and Schwarz's Bayesian information criteria (SBC), to name a few of the more popular criteria. If all vendors could agree to use, for example, AIC, then all IACF plots would be identical. PROC ARIMA does not employ a criteria; it simply selects as p the value supplied for the NLAGS option. You can easily see how the IACF changes as you change the value supplied for NLAGS.

Fortunately, even if different approximating models are used, plots of the IACF are usually similar even if the numbers are not exactly the same.

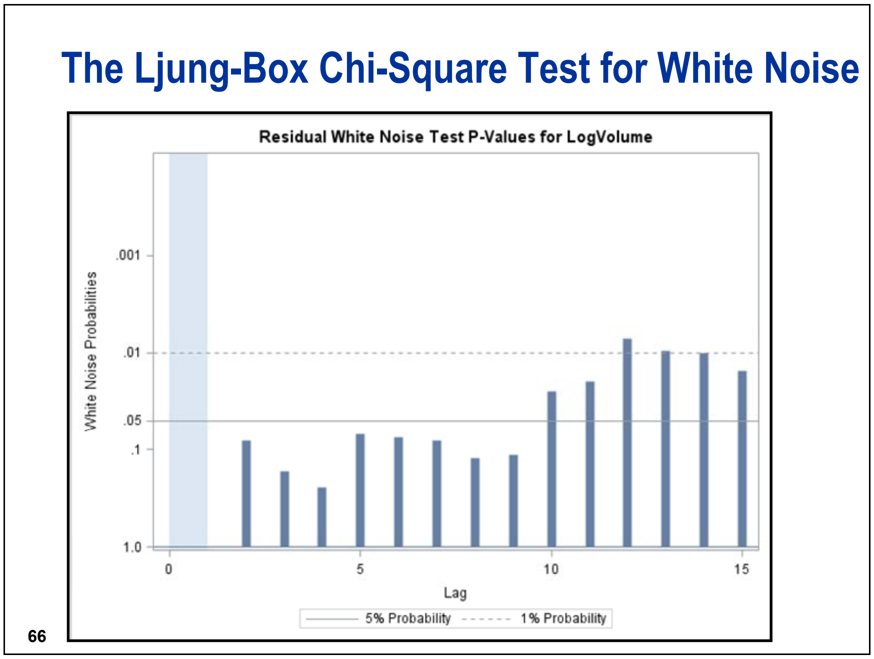
The Ljung-Box Chi-Square Test for White Noise

- A *white noise* time series is a Gaussian (normal, bell-shaped) time series with mean zero and positive fixed variance in which all observations are independent of each other.
- The null hypothesis is that the series is white noise, and the alternative hypothesis is that one or more autocorrelations up to lag *m* are not zero.
- The Ljung-Box test can be applied to the original series or to the residuals after fitting a model.

$$\chi_m^2 = n(n+2)\sum_{k=1}^{m} \frac{r_k^2}{n-k}, r_k = \mathrm{ACF}(k) \text{ given } \mu = 0.$$

65

The Ljung-Box test helps ensure that models fit to data are adequate. The residuals of a forecast model should resemble white noise. If not, then the model is missing a feature that would enable it to better represent the systematic variation in the series. There are competing white noise tests. Popular competitors include Bartlett's K-S test and Fisher's Kappa test, which are both available using PROC SPECTRA. Evidence suggests that the Ljung-Box test might be a more powerful test for situations that are more likely to be encountered. There is no ***most*** powerful test for all situations, which explains why competing tests exist.



The chart associated with the Ljung-Box chi-square test is often presented on a log scale. The bars are shown going from 1.0 to 0.0 with the height of a bar representing one minus the *p*-value associated with the test statistic. Thus, a large bar means a small *p*-value (statistical significance) and a small bar indicates a large *p*-value. The "white means white" memory device interprets a chart that is mostly white (small bars) as implying that the underlying time series is white noise. While the Ljung-Box test might be relevant for the original time series, it is more often applied to the residuals after a forecast model is fit to the series. A model is adequate if the residuals appear to be white noise.

The Ljung-Box test is actually a series of tests at different lags. Plots including multiple tests are routinely used, but you need not require that all tests have a *p*-value larger than 0.05. Some judgment is required.

The Ljung-Box Chi-Square Test for White Noise

Autocorrelation Check of Residuals									
To Lag	Chi-Square	DF	Pr > ChiSq	Autocorrelations					
6	10.06	5	0.0736	-0.062	0.080	-0.043	0.042	0.125	0.068
12	25.72	11	0.0072	0.065	-0.021	0.073	0.145	0.086	0.125
18	29.58	17	0.0296	-0.046	0.068	-0.018	0.053	0.004	-0.061
24	41.63	23	0.0100	0.149	0.075	0.083	-0.052	0.062	0.041
30	50.87	29	0.0073	0.083	0.112	-0.027	0.002	0.054	0.102
36	60.21	35	0.0051	0.161	-0.050	0.022	0.016	0.073	-0.003
42	61.63	41	0.0201	0.006	-0.024	0.030	-0.057	0.023	-0.008
48	69.35	47	0.0186	-0.055	0.084	0.014	-0.040	-0.003	-0.134

67

PROC ARIMA provides a table of chi-square values using lags that are a multiple of six. The above table and the previous plot are associated with an AR(1) model for **LogVolume**. The p-values for lags 12, 24, and so on are all less than 0.05, suggesting that the residuals are not white noise and the model is not adequate. The small p-values provide evidence of ***lack of fit*** for the AR(1) model.

To understand the impact that the autocorrelations have on the chi-square statistic, observe that ACF(10)=0.145 and ACF(12)=0.125. The contribution of ACF(10) to the lag 12 chi-square statistic is as follows:

$$\frac{n(n+2)}{n-10} r_{10}^2 = \frac{312(314)}{302} 0.145^2 = 6.82$$

The contribution of ACF(12) is the following:

$$\frac{n(n+2)}{n-12} r_{12}^2 = \frac{312(314)}{300} 0.125^2 = 5.10$$

The lag 12 chi-square statistic is 25.72. Almost half of this value comes from the contribution of the lag 10 and lag 12 autocorrelation values. These two autocorrelations are included in all the subsequent chi-square values, that is, chi-square is cumulative.

✐ The presence of missing values in the data impacts the calculation of the sample ACF as well as the Ljung-Box chi-square statistic. The above calculations that employ ACF(10) and ACF(12) do not include corrections for missing values in the estimation of the sample ACF. However, the values that are used approximate the actual calculations and provide a reasonable approximation to the contribution that ACF10) and ACF(12) make to the overall chi-square statistic.

Surprisingly, despite evidence from the PACF and IACF that an AR(1) model is appropriate, an AR(4) model is required to get residuals that pass a white noise test. (The next demonstration completes the analysis.)

The data and graphs produced by PROC ARIMA help the trained analyst identify some potential models for data. PROC TIMESERIES is capable of producing some of the same diagnostics. In addition, PROC TIMESERIES supports two straightforward seasonal decomposition strategies that are exploited in later chapters.

Many SAS procedures can automatically produce graphs with ODS Graphics. A programmer only needs to turn on the ODS Graphics functionality to automatically produce high quality graphics. These can be sent as HTML files with the appropriate ODS statements. For this course, the default viewer for the particular machine being used is employed and HTML results are discarded after each analysis. In addition to popular output formats such as HTML and Adobe PDF, exporting graphs can be a simple matter of using copy and paste.

TIMESERIES Procedure

```
PROC TIMESERIES DATA=SAS-data-set
                PRINT=(options)
                PLOT=(options)
                SEASONALITY=n <options>;
    BY variables;
    ID variable INTERVAL=interval <options>;
    VAR variables / <options>;
    DECOMP components / <options>;
RUN;
QUIT;
```

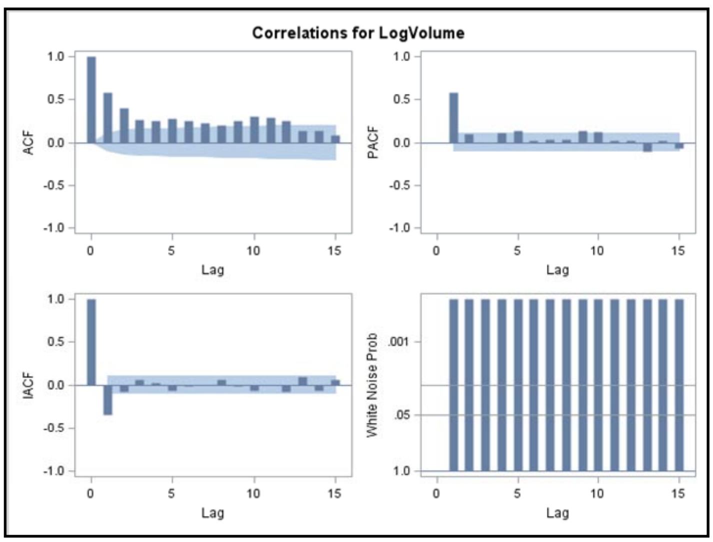
68

PROC TIMESERIES Diagnostics

```
proc timeseries
     data=work.AMR
     print=(descstats)
     plot=(series corr acf pacf iacf wn)
     seasonality=5;
   id Date interval=weekday;
   var LogVolume;
run;
```

69

PROC TIMESERIES CORR Panel



70

Diagnosing and Fitting Models to Time Series Data

This demonstration illustrates how to use SAS procedures to diagnose and fit models to a time series. The program code can be found in **Demo1_02ACF.sas**.

The goal of this analysis is to identify candidate forecast models for the AMR daily stock volume data. The AMR stock volume data has several problems that can be addressed in the data preparation stage.

1. The data has missing values due to dates that the stock market is closed.
2. The AMR stock volume time series is skewed to the right with a few unusually high trading days.
3. The stock volume is low for days that the market is only open for a half day. The period covered by the extracted data, January 1, 2010 through March 31, 2011, has only one day, November 26, 2010, that the market was open a half day.

The above three issues are addressed as follows:

1. The missing value features of the SAS procedures used will be exploited. There will be no preprocessing of the data to impute missing values. (A methodology for dealing with the unique nature of missing values in stock market data was mentioned earlier.)
2. The natural logarithm transformation will be applied to the original stock volume.
3. The one affected day will be adjusted by multiplying the stock volume by 2. (Prior analysis shown in the course notes did not make the half-day correction, so some of the results in this demonstration will differ from previous results that were displayed.)

The original data covered the period 1995 to 2011. The following code expands missing dates to produce missing values. The data is truncated to include the period of interest, January 1, 2010, to March 31, 2011.

```
proc expand data=sasuser.amrdaily
            out=work.amrdaily
            from=weekday
            align=beginning
            method=none
            ;
   id Date;
   convert Volume;
   convert LogVolume;
run;
```

PROC EXPAND adds missing weekday dates that were not in the original data, and sets the value of **Volume** and **LogVolume** to missing on those dates.

```
data work.amr;
   set work.amrdaily(where=('01JAN2010'd<=Date<='31MAR2011'd));
   keep Date Volume LogVolume;
run;
```

Step 3 mentioned above requires multiplying the volume by 2 on the affected day.

```
data sasuser.amr;
   set work.amr;
   if (Date='26NOV2010'd) then do;
      Volume=2*Volume;
      LogVolume=log(Volume);
   end;
run;
```

The data set is permanently written to the Sasuser library for use in a later demonstration.

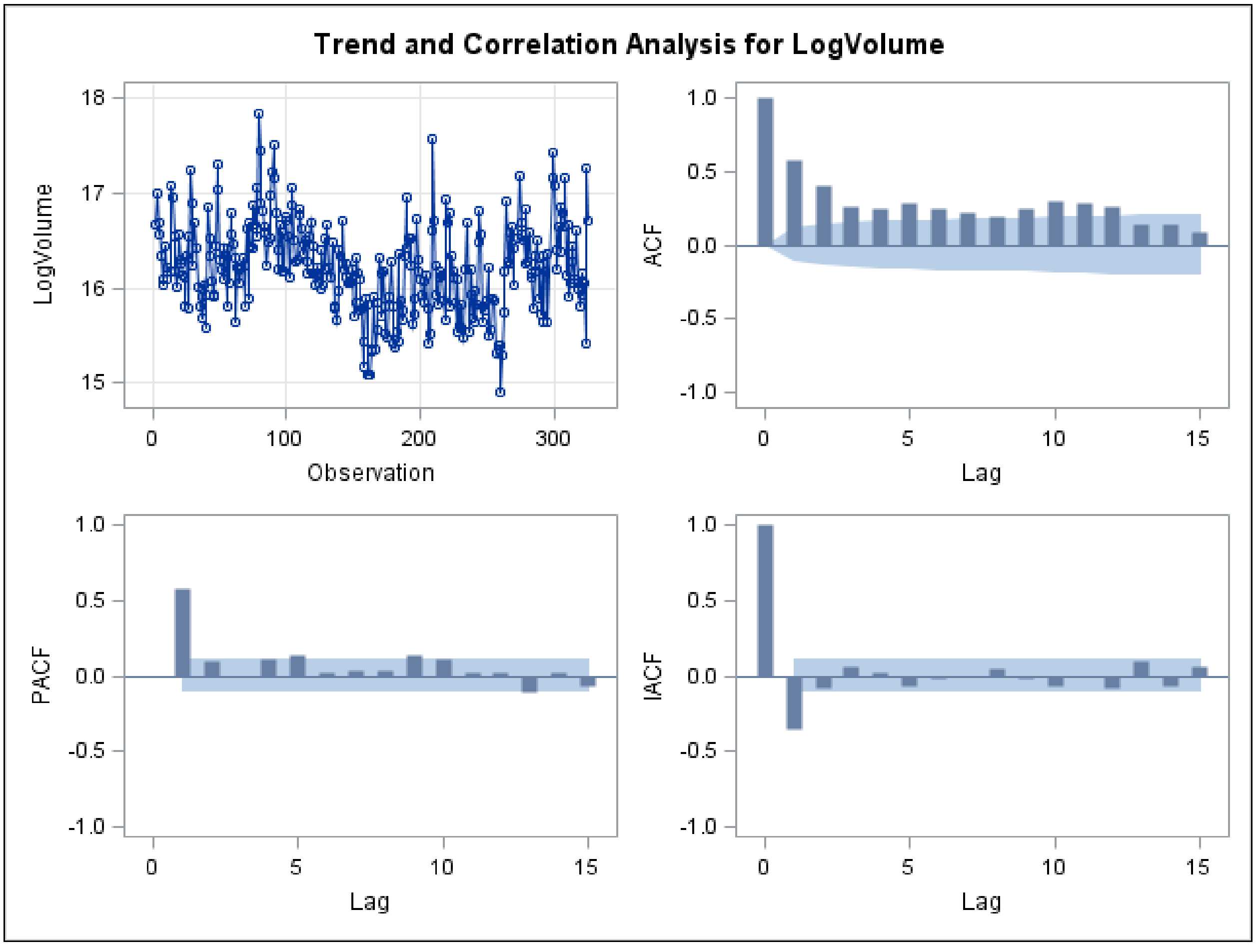
The following code produces sample correlation plots to help diagnose a forecast model for the data:

```
proc arima data=sasuser.amr plots=all;
   identify var=LogVolume nlags=15;
quit;
```

If you want the enlarged plots, use the UNPACK option as in the following code:

```
proc arima data=sasuser.amr plots=all plots(unpack);
   identify var=LogVolume nlags=15
            outcov=sasuser.amrcov;
quit;
```

The smaller plots appear below:



The course notes addressed the exponential decay of the sample ACF for the first few lags. The decay does not persist for higher lags. The spike at lag 1 of the PACF and IACF suggest an AR(1) model, which is supported by the decay pattern in the ACF, but the significant ACF values at higher lags seem to contradict a simple model.

If you want to scrutinize actual autocorrelation values, you can display the low resolution plots with the actual value using the following code. The code also outputs sample ACF, PACF, and IACF values into temporary SAS tables using ODS output tables:

```
ods graphics off;
ods output AutoCorrGraph=sasuser.amracf
           PACFGraph=sasuser.amrpacf
           IACFGraph=sasuser.amriacf;
proc arima data=sasuser.amr;
   identify var=LogVolume nlags=15
            outcov=sasuser.amrcov;
quit;
ods graphics on;
```

You must turn ODS Graphics off to prevent production of the high resolution plots. The low resolution plots are suppressed when the high resolution plots are requested. Here is the ACF plot:

```
                          AMR Daily Stock Volume

                           The ARIMA Procedure

                        Name of Variable = LogVolume

            Mean of Working Series                      16.22554
            Standard Deviation                           0.47795
            Number of Observations                           324
            Embedded missing values in working series         12

                             Autocorrelations

   Lag  Covariance  Correlation  -1 9 8 7 6 5 4 3 2 1 0 1 2 3 4 5 6 7 8 9 1

     0    0.228436      1.00000  |                    |********************|
     1    0.131693      0.57650  |                  . |************        |
     2    0.091205      0.39926  |                 .  |********            |
     3    0.059033      0.25842  |                 .  |*****               |
     4    0.056252      0.24625  |                 .  |*****               |
     5    0.063764      0.27913  |                 .  |******              |
     6    0.057282      0.25076  |                 .  |*****               |
     7    0.051332      0.22471  |                .   |****                |
     8    0.044784      0.19605  |                .   |****                |
     9    0.057181      0.25031  |                .   |*****               |
    10    0.069138      0.30266  |                .   |******              |
    11    0.065444      0.28648  |                .   |******              |
    12    0.058153      0.25457  |                .   |*****               |
    13    0.031955      0.13989  |                .   |***.                |
    14    0.030663      0.13423  |                .   |***.                |
    15    0.019304      0.08450  |                .   |** .                |
                         "." marks two standard errors
```

The PACF plot suggests an AR(1) model.

```
                         Partial Autocorrelations

        Lag    Correlation    -1 9 8 7 6 5 4 3 2 1 0 1 2 3 4 5 6 7 8 9 1

          1        0.57650    |                  . |************        |
          2        0.10021    |                  . |**                  |
          3       -0.00976    |                  . | .                  |
          4        0.11430    |                  . |**                  |
          5        0.13570    |                  . |***                 |
          6        0.01551    |                  . | .                  |
          7        0.02976    |                  . |*.                  |
          8        0.03241    |                  . |*.                  |
          9        0.13250    |                  . |***                 |
         10        0.11609    |                  . |**                  |
         11        0.01677    |                  . | .                  |
         12        0.02354    |                  . | .                  |
         13       -0.10201    |                  **| .                  |
         14        0.02284    |                  . | .                  |
         15       -0.07431    |                  .*| .                  |
```

The IACF plot produces the same diagnostics.

```
                          Inverse Autocorrelations

     Lag    Correlation    -1 9 8 7 6 5 4 3 2 1 0 1 2 3 4 5 6 7 8 9 1

       1       -0.35011    |              *******| .                  |
       2       -0.08509    |                   **| .                  |
       3        0.06360    |                   . |*.                  |
       4        0.01791    |                   . | .                  |
       5       -0.06561    |                   .*| .                  |
       6       -0.01426    |                   . | .                  |
       7       -0.00976    |                   . | .                  |
       8        0.05243    |                   . |*.                  |
       9       -0.02008    |                   . | .                  |
      10       -0.06682    |                   .*| .                  |
      11       -0.00290    |                   . | .                  |
      12       -0.08106    |                   **| .                  |
      13        0.09820    |                   . |**                  |
      14       -0.07282    |                   .*| .                  |
      15        0.05698    |                   . |*.                  |
```

PROC ARIMA can be used to fit and assess an AR(1) model.

```
proc arima data=sasuser.amr;
   identify var=LogVolume nlags=15 noprint;
   estimate p=1 ml;
quit;
```

Because diagnostic results were already examined, the NOPRINT option is used in the IDENTIFY statement.

The estimated model and related statistics appear below:

AMR Daily Stock Volume

The ARIMA Procedure

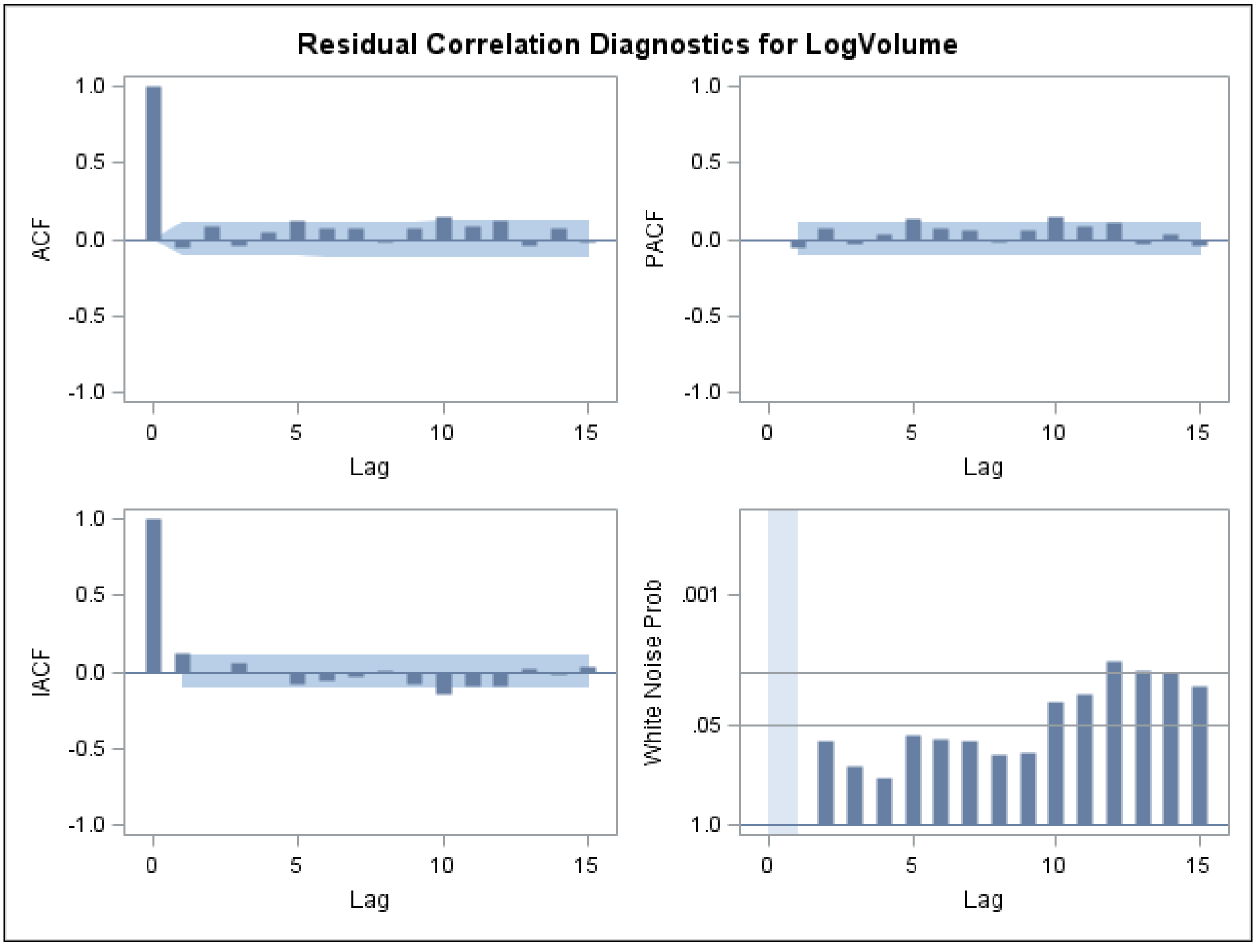
Maximum Likelihood Estimation					
Parameter	Estimate	Standard Error	t Value	Approx Pr > |t|	Lag
MU	16.22954	0.04985	325.56	<.0001	0
AR1,1	0.56257	0.04663	12.06	<.0001	1

Constant Estimate	7.099364
Variance Estimate	0.155802
Std Error Estimate	0.394718
AIC	310.6273
SBC	318.1134
Number of Residuals	312

The Ljung-Box chi-square statistics suggest a lack of fit for the AR(1) model.

Autocorrelation Check of Residuals									
To Lag	Chi-Square	DF	Pr > ChiSq	Autocorrelations					
6	10.06	5	0.0736	-0.062	0.080	-0.043	0.042	0.125	0.068
12	25.72	11	0.0072	0.065	-0.021	0.073	0.145	0.086	0.125
18	29.58	17	0.0296	-0.046	0.068	-0.018	0.053	0.004	-0.061
24	41.63	23	0.0100	0.149	0.075	0.083	-0.052	0.062	0.041
30	50.87	29	0.0073	0.083	0.112	-0.027	0.002	0.054	0.102
36	60.21	35	0.0051	0.161	-0.050	0.022	0.016	0.073	-0.003
42	61.63	41	0.0201	0.006	-0.024	0.030	-0.057	0.023	-0.008
48	69.35	47	0.0186	-0.055	0.084	0.014	-0.040	-0.003	-0.134

The graph plots *p*-values for the chi-square white noise test.



The large autocorrelations at lags 10 and 12 cause a significant deviation from the white noise null hypothesis, as shown earlier in the notes. Additional residual diagnostics confirm that the model should be disqualified.

You can examine the fit of an MA(1) model for insight into the diagnostics, even though an MA model is not indicated by any of the autocorrelation plots. The following code fits an MA(1) model to the data:

```
proc arima data=sasuser.amr;
   identify var=LogVolume nlags=15 noprint;
   estimate q=1 ml;
quit;
```

A subset of the output follows:

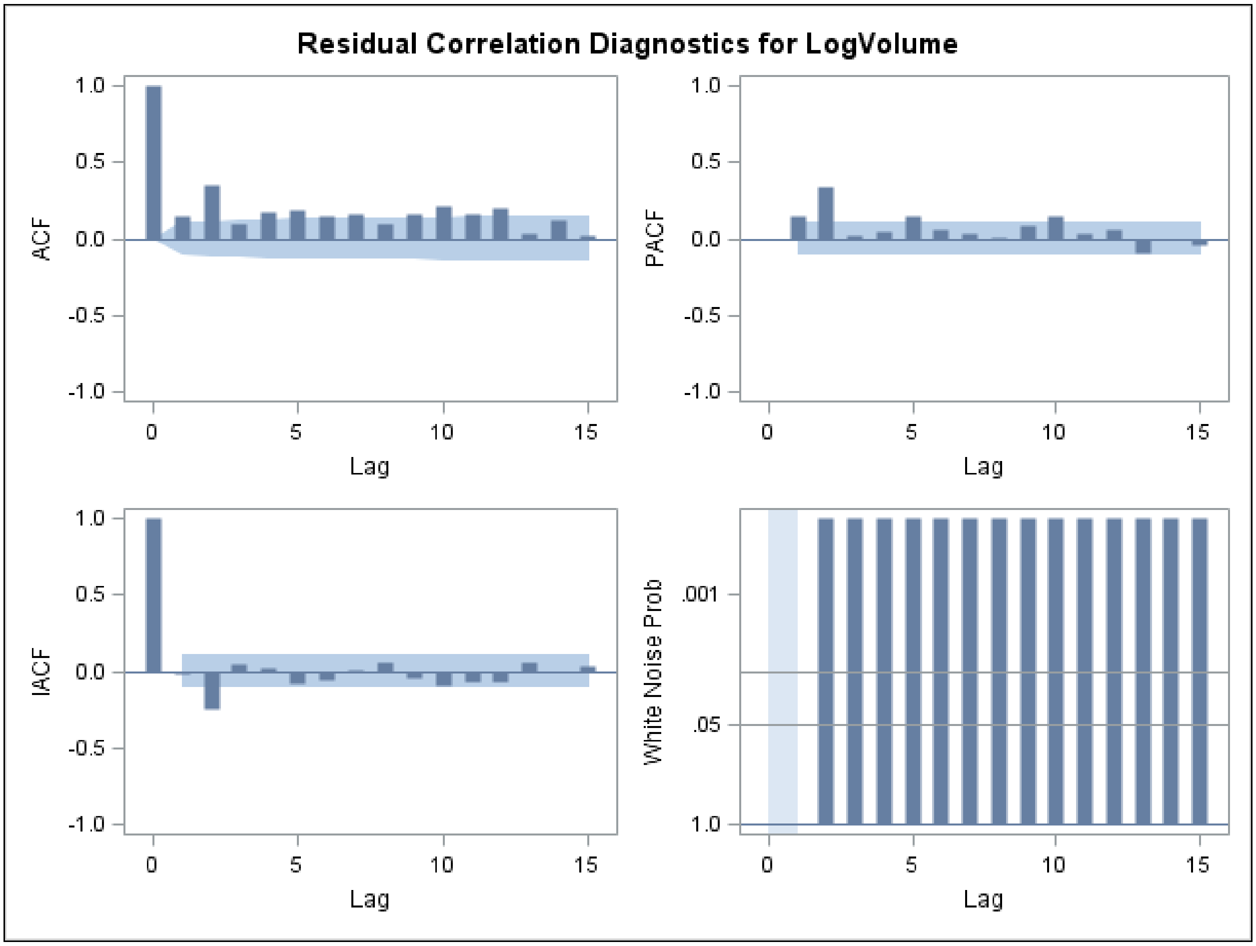
AMR Daily Stock Volume

The ARIMA Procedure

Maximum Likelihood Estimation					
Parameter	Estimate	Standard Error	t Value	Approx Pr > |t|	Lag
MU	16.22473	0.03348	484.61	<.0001	0
MA1,1	-0.42619	0.05291	-8.06	<.0001	1

Constant Estimate	16.22473
Variance Estimate	0.17681
Std Error Estimate	0.420488
AIC	349.022
SBC	356.508
Number of Residuals	312

Autocorrelation Check of Residuals									
To Lag	Chi-Square	DF	Pr > ChiSq	Autocorrelations					
6	70.58	5	<.0001	0.142	0.347	0.099	0.168	0.190	0.151
12	120.35	11	<.0001	0.155	0.100	0.161	0.213	0.159	0.202
18	127.87	17	<.0001	0.031	0.123	0.022	0.078	0.058	0.013
24	153.29	23	<.0001	0.177	0.103	0.145	0.030	0.132	0.098
30	176.56	29	<.0001	0.142	0.148	0.051	0.076	0.120	0.140
36	193.03	35	<.0001	0.202	0.029	0.091	0.041	0.093	0.012
42	194.51	41	<.0001	0.027	-0.028	0.018	-0.060	0.004	-0.013
48	201.17	47	<.0001	-0.037	0.047	-0.006	-0.055	-0.015	-0.138



The MA(1) model is clearly inadequate.

You learn about the general family of autoregressive moving average (ARMA) models in the next chapter. The fact that the residuals of the AR(1) model are autocorrelated suggests that a model that looks more than one time unit into the past is required. Several steps might be necessary to find an adequate model. To simplify the demonstration, an AR(4) model will be employed without justification.

```
proc arima data=sasuser.amr;
   identify var=LogVolume nlags=15 noprint;
   estimate p=4 ml;
quit;
```

The results follow:

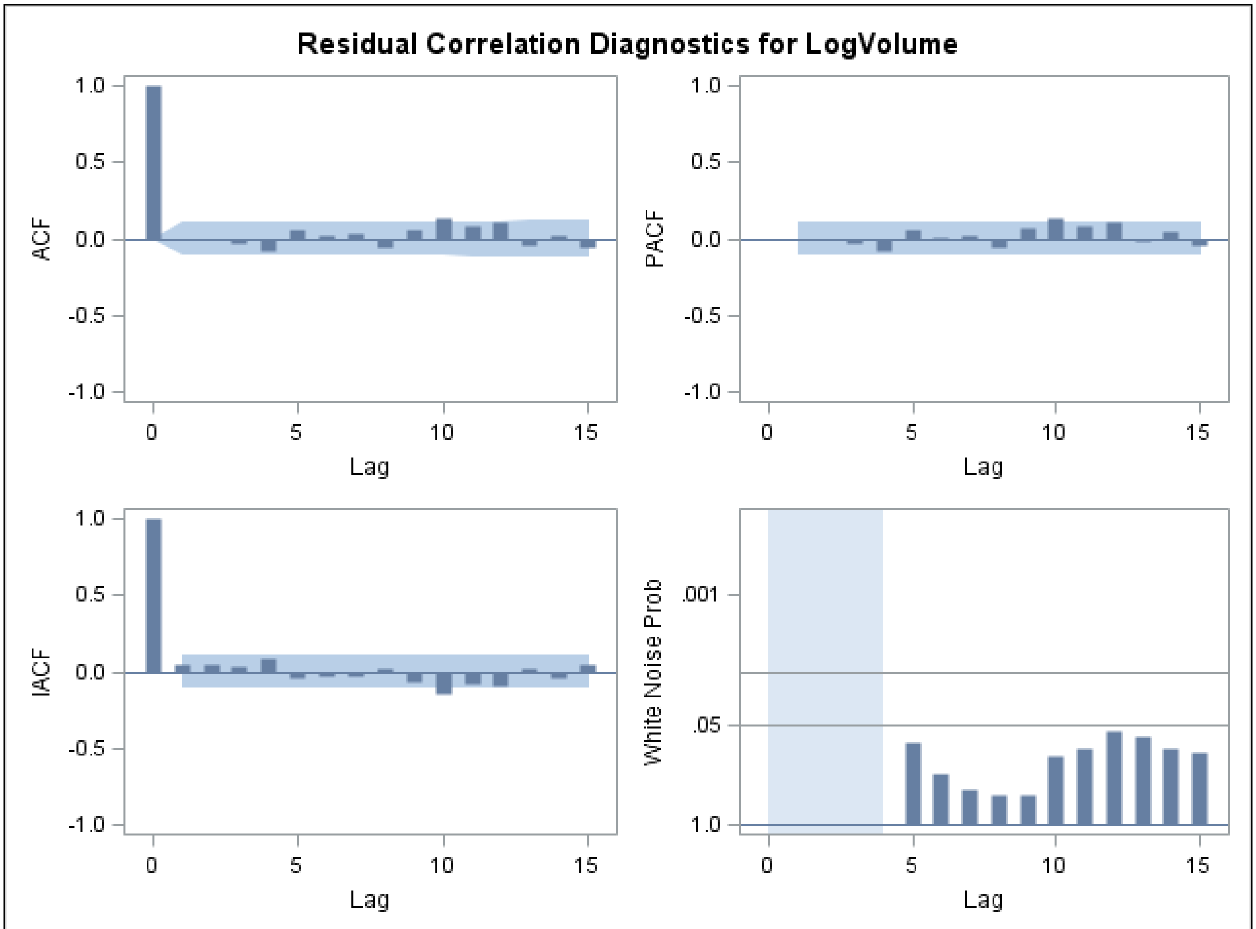
AMR Daily Stock Volume

The ARIMA Procedure

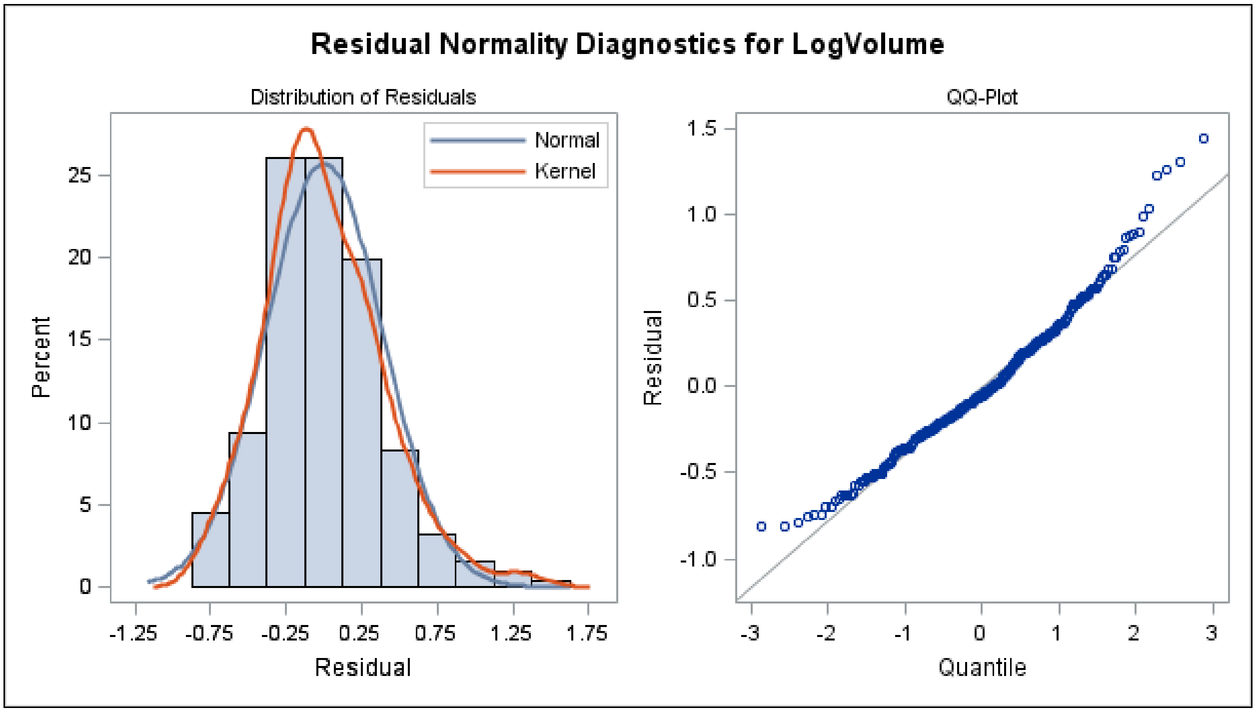
Maximum Likelihood Estimation					
Parameter	Estimate	Standard Error	t Value	Approx Pr > |t|	Lag
MU	16.23350	0.06364	255.08	<.0001	0
AR1,1	0.49677	0.05704	8.71	<.0001	1
AR1,2	0.10778	0.06511	1.66	0.0979	2
AR1,3	-0.06905	0.06733	-1.03	0.3052	3
AR1,4	0.12737	0.05897	2.16	0.0308	4

Constant Estimate	5.472892
Variance Estimate	0.152881
Std Error Estimate	0.391
AIC	307.5684
SBC	326.2834
Number of Residuals	312

Autocorrelation Check of Residuals									
To Lag	Chi-Square	DF	Pr > ChiSq	Autocorrelations					
6	3.05	2	0.2178	-0.007	-0.008	-0.035	-0.075	0.055	0.015
12	15.07	8	0.0578	0.027	-0.052	0.056	0.132	0.084	0.103
18	18.51	14	0.1846	-0.048	0.025	-0.054	0.018	-0.020	-0.074
24	27.58	20	0.1198	0.133	0.081	0.055	-0.055	0.038	0.025
30	34.87	26	0.1146	0.057	0.097	-0.047	-0.008	0.034	0.102
36	42.39	32	0.1035	0.152	-0.044	-0.002	0.017	0.048	-0.005
42	44.07	38	0.2302	-0.013	-0.026	0.015	-0.067	0.026	-0.002
48	52.38	44	0.1808	-0.057	0.106	0.014	-0.040	-0.014	-0.126



The AR(4) model does not exhibit any lack of fit problems. Additional diagnostics are supplied to judge how well the residuals conform to a white noise hypothesis beyond only looking at autocorrelations.

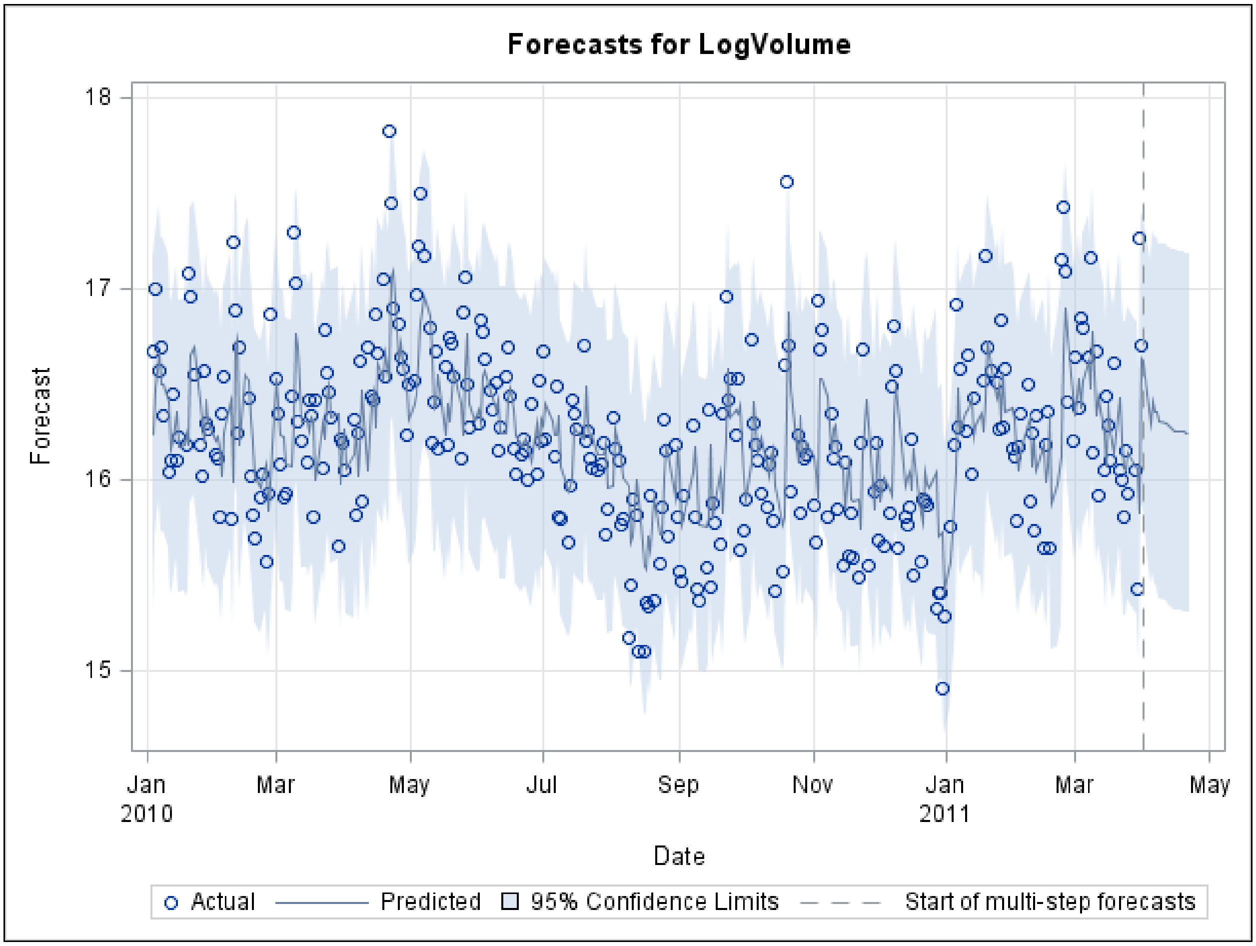


The kernel density estimate can be thought of as a locally smoothed version of a high precision histogram. The QQ or quantile-quantile plot plots the residual on the vertical axis versus a standardized version of what should theoretically be obtained if the data were normal. Truly normal data would plot close to a straight line on the QQ plot. Several of the normality tests in PROC UNIVARIATE measure departures from the straight line. Here the plot looks reasonably good with one or two points possibly being a bit extreme under the normality assumption.

The forecasts for the AR(4) model are obtained using the following code:

```
proc arima data=sasuser.amr plots(only)=(forecast(forecast));
   identify var=LogVolume nlags=15 noprint;
   estimate p=4 ml noprint;
   forecast lead=15 id=Date interval=weekday;
quit;
```

A plot of the forecasts follows:



Because of the coefficients going back to lag 4, the forecasts revert more slowly to the mean.

1.3 Evaluating Forecasts

Objectives

- Introduce and summarize popular goodness-of-fit/ accuracy statistics.
- Illustrate how to derive the statistics from SAS procedures and from SAS DATA step code.
- Produce goodness-of-fit/accuracy statistics for select time series.

74

Forecasting

If someone asks you if you can forecast something, your answer should always be “Yes.”

If someone asks you if you can forecast something ***accurately***, you cannot answer until you established what accuracy means and until you performed preliminary modeling of the data.

75

Liability

"Will you stake your reputation on the accuracy of these forecasts?"

"No, but I will stake my reputation on the methodology that was used to generate the forecasts."

- You might have no control over data accuracy and validity.
- You have no control over future events such as catastrophes, economic downturns, war, the integrity of key players, the survival of key players, and so on.

76

Picking a Winning Set of Forecasts

Good forecasts should have the following characteristics:

- Be highly correlated with the actual series values
- Exhibit small forecast errors
- Capture the salient features of the original time series

In addition, forecast quality should be based on the business, engineering, or scientific problem being addressed.

77

The Use of Judgment in Forecasting

- *Judgmental forecasting* refers to subjective human manipulation of forecasts.
- A manager might know something that cannot be expressed as part of a statistical model.
  - A labor strike will limit parts availability.
  - An earthquake that occurred in a region having many manufacturing facilities will cause shortages when inventories expire.
  - Data might not be available.
- Judgmental forecasts must be identified and assessed in the same way as statistical forecasts.
- Judgmental forecasting is risky and often leads to inferior forecasts.

78

Amos Tversky pioneered the study of human judgment and contributed to one of the first comprehensive studies of judgment (Kahneman, Slavic, and Tversky 1982). The follow-up work by Gilovich, Griffin, and Kahneman (2002) provides additional insight into how humans make decisions. The general finding is that human judgment in forecasting tends to be very poor when compared to analytic techniques.

Accuracy versus Goodness-of-Fit

- A diagnostic statistic calculated using the same sample that was used to fit the model is a ***goodness-of-fit*** statistic.
- A diagnostic statistic calculated using a holdout sample that was not used in modeling is an ***accuracy*** statistic.

79

continued...

Accuracy versus Goodness-of-Fit

- Assessing a predictive model using accuracy statistics calculated for a holdout sample is called ***honest assessment***.
- In general, an accuracy statistic provides an unbiased estimate of implementation accuracy, that is, the accuracy actually experienced when the forecast model is deployed.
- The ***optimism principal***: goodness-of-fit statistics tend to give an optimistic estimate of implementation accuracy.

80

A fair assessment of predictability would use a holdout sample at the end of the data. Without this, the model is assessed on the data that "gave it birth." It likely fits that data better than it will fit the future observations that it is meant to forecast. One possibility is to limit the fit range with a WHERE statement, and then merge the forecast data with the original data and apply the GOFstats macro as is done in **Ch01_d07.sas**.

✐ The penalty terms in AIC and SBC are meant to account for this overfitting phenomenon when no holdout is available. AIC and SBC are thus not of interest when computed on a holdout sample.

When you assess on a holdout sample, the RMSE denominator degrees of freedom should be the withhold data sample size, not adjusted for degrees of freedom so setting NumParms = 0 in the GOFstats macro is appropriate. The reason for adjusting error degrees of freedom for the number of estimated parameters is the same as the reason for using a holdout sample, that is, the errors in the fit data are on average a bit too small because the model is custom fit to that particular data sample.

Model Diagnostic Statistics

Notation for Basic Quantities

Series Length: *n*

Number of Model Parameters to Estimate: *k*

Model Likelihood Function Evaluated at Maximum: *L*

Sum of Squared Errors: SSE

Predicted Target Value Y_t at Time *t*: $\hat{Y}_t$

81

continued...

Model Diagnostic Statistics

R-Square: $R^2 = 1 - \sum_{t=1}^{n}(Y_t - \hat{Y}_t)^2 \Big/ \sum_{t=1}^{n}(Y_t - \bar{Y})^2$

Root Mean Squared Error:

$$\text{MSE} = \frac{1}{n-k}\sum_{t=1}^{n}(Y_t - \hat{Y}_t)^2 \; *$$

$$\text{RMSE} = \sqrt{\text{MSE}}$$

* For holdout samples, use divisor *n* rather than *n-k*.

82

continued...

As described before, R square compares the sum of squared errors (SSE) of the forecast model to the SSE for the forecast model that forecasts every future value as the mean of the historic data. Consequently, if a model produces forecasts worse than the mean, then R square is negative. This seems to contradict R square being a squared quantity that must be positive. Some practitioners prefer to define R square so that it can never be negative.

Model Diagnostic Statistics

Mean Absolute Percent Error:

$$\text{MAPE} = \frac{1}{n}\sum_{t=1}^{n} | Y_t - \hat{Y}_t | / Y_t$$

Mean Absolute Error:

$$\text{MAE} = \frac{1}{n}\sum_{t=1}^{n} | Y_t - \hat{Y}_t |$$

83

continued...

MAPE requires that the time series values be positive.

Model Diagnostic Statistics

Symmetric Mean Absolute Percent Error:

$$\text{SMAPE} = \frac{1}{n}\sum_{t=1}^{n} | Y_t - \hat{Y}_t | /[(Y_t + \hat{Y}_t)/2]$$

Mean Absolute Error/Mean:

$$\text{MAE/Mean} = \frac{\frac{1}{n}\sum_{t=1}^{n} | Y_t - \hat{Y}_t |}{\frac{1}{n}\sum_{t=1}^{n} Y_t}$$

84

continued...

The symmetric MAPE (SMAPE) is recommended for situations where the actual value of the time series can be zero. The MAE/Mean is a special case of the Weighted Mean Absolute percent Error (WMAPE). SMAPE and MAE/Mean are not popular in the business world, because MAPE seems to dominate business forecasting.

Model Diagnostic Statistics

Likelihood-Based Information Criteria

General Formula:

IC = Accuracy + Penalty

Akaike's A Information Criteria:

$$\mathrm{AIC} = -2\log(L) + 2k$$

Schwarz's Bayesian Information Criteria:

$$\mathrm{SBC} = -2\log(L) + k\log(n)$$

85

continued...

AIC enables more parameters than SBC when $2k$ is smaller than $k\log(n)$. This occurs for all series with more than seven observations. Schwarz showed that AIC is biased toward over-parameterization. This might actually be a property that favors AIC over SBC.

Model Diagnostic Statistics

Error-Based Information Criteria

Akaike's A Information Criteria:

$$\text{AIC} = n\log(\text{SSE}/n) + 2k$$

Schwarz's Bayesian Information Criteria:

$$\text{SBC} = n\log(\text{SSE}/n) + k\log(n)$$

86

The likelihood form of AIC and SBC requires that models be *hierarchically consistent* before they can be compared. The error-based form of AIC and SBC enables comparison of any two models.

There are variations of the classic R square measure that are designed to decrease when adding complexity to a model does not produce a commensurate increase in accuracy. Adjusted R square is a popular alternative to R square.

$$\text{ADJRSQ} = 1 - \frac{(n-1)(1-R^2)}{n-k}$$

While R square always increases when inputs are added to a model, the variations of adjusted R square might decrease when an input is added. The random walk R square simply compares a forecast model to the random walk forecast model.

$$\text{RWRSQ} = 1 - \frac{\sum (Y-\hat{Y})^2}{\sum (Y-\hat{Y}_{RW})^2}$$

The choice of accuracy statistic for model selection might be a critical step in a given forecasting project.

1.03 Multiple Choice Poll

Which goodness-of-fit/accuracy statistic do you currently use most often?

a. MAPE
b. RMSE
c. SMAPE
d. R-Square
e. AIC
f. SBC
g. Another statistic
h. No statistic

88

Obtaining Model Diagnostic Statistics

This demonstration illustrates how to obtain model diagnostic statistics. The code can be found in **Demo1_03GOF.sas**.

The three models considered before are revisited, even though two of the models were disqualified.

```
proc arima data=sasuser.amr;
   identify var=LogVolume noprint;
   /*----  AR(1)  ----*/
   estimate p=1 method=ml noprint
            outstat=work.Stat_AR1;
   forecast lead=12 id=Date interval=weekday
            out=work.amr_AR1 noprint;
   /*----  MA(1)  ----*/
   estimate q=1 method=ml noprint
            outstat=work.Stat_MA1;
   forecast lead=12 id=Date interval=weekday
            out=work.amr_MA1 noprint;
   /*----  AR(4)  ----*/
   estimate p=4 method=ml noprint
            outstat=work.Stat_AR4;
   forecast lead=12 id=Date interval=weekday
            out=work.amr_AR4 noprint;
quit;
```

The three forecast data sets can be employed to calculate goodness-of-fit statistics that are not provided by PROC ARIMA. The **OUTSTAT** data set contains the statistics AIC and SBC, and it also contains sufficient information to calculate RMSE, namely SSE, number of residuals, and number of parameters in the model.

View of the **WORK.STAT_AR1** Data Set

VIEWTABLE: Work.Stat_ar1 (Model ...

	Type of Estimation Method Used	Name of Statistic	Value of Statistic
1	ML	AIC	310.627347
2	ML	SBC	318.113353
3	ML	LOGLIK	-153.31367
4	ML	SSE	48.298651
5	ML	NUMRESID	312
6	ML	NPARMS	2
7	ML	NDIFS	0
8	ML	ERRORVAR	0.1558021
9	ML	MU	16.2295417
10	ML	CONV	0
11	ML	NITER	3

The following DATA step calculates MAPE and RMSE and uses the **forecast** data set:

```
data work.ar1gof;
   attrib Model length=$12
          MAPE  length=8
          NMAPE length=8
          MSE   length=8
          RMSE  length=8
          NMSE  length=8
          NumParm length=8;
   set work.amr_AR1 end=lastobs;
   retain MAPE MSE NMAPE NMSE 0 NumParm 2;
   Actual=exp(LogVolume);
   Forecast=exp(Forecast+0.5*STD*STD);
   Residual=Actual-Forecast;
   /*----  SUM function necessary to handle missing  ----*/
   MAPE=sum(MAPE,100*abs(Residual)/Actual);
   NMAPE=NMAPE+N(100*abs(Residual)/Actual);
   MSE=sum(MSE,Residual**2);
   NMSE=NMSE+N(Residual);
   if (lastobs) then do;
      Model="AR(1)";
      MAPE=MAPE/NMAPE;
      RMSE=sqrt(MSE/NMSE);
      if (NumParm>0) and (NMSE>NumParm) then
         RMSE=sqrt(MSE/(NMSE-NumParm));
      else RMSE=sqrt(MSE/NMSE);
      output;
   end;
   keep Model MAPE RMSE NumParm;
run;
```

The resulting table can be viewed, printed, or combined with other goodness-of-fit tables. To calculate accuracy for the original scale of the data rather than for the log scale, the actual and forecast values are converted back using the exponential function. The forecast was adjusted based on the distributional properties of the lognormal distribution as mentioned before. Theoretical details are provided at the end of the chapter.

Because the calculation of MAPE and RMSE is a standard activity, you should convert the code into a macro that can be used for any data set and forecast model coming from PROC ARIMA. The following code shows the above code after it is converted into a macro:

```
%macro GOFstats(ModelName=,DSName=,OutDS=,NumParms=0,
                ActualVar=Actual,ForecastVar=Forecast);
data &OutDS;
   attrib Model length=$12
          MAPE  length=8
          NMAPE length=8
          MSE   length=8
          RMSE  length=8
          NMSE  length=8
          NumParm length=8;
   set &DSName end=lastobs;
   retain MAPE MSE NMAPE NMSE 0 NumParm &NumParms;
   Residual=&ActualVar-&ForecastVar;
   /*----  SUM function necessary to handle missing  ----*/
   MAPE=sum(MAPE,100*abs(Residual)/&ActualVar);
   NMAPE=NMAPE+N(100*abs(Residual)/&ActualVar);
   MSE=sum(MSE,Residual**2);
   NMSE=NMSE+N(Residual);
   if (lastobs) then do;
      Model="&ModelName";
      MAPE=MAPE/NMAPE;
      RMSE=sqrt(MSE/NMSE);
      if (NumParm>0) and (NMSE>NumParm) then
         RMSE=sqrt(MSE/(NMSE-NumParm));
      else RMSE=sqrt(MSE/NMSE);
      output;
   end;
   keep Model MAPE RMSE NumParm;
run;
%mend GOFstats;
```

The code for converting the log-scaled data was removed, so before using the macro, you must make the forecast adjustments to the forecast data. The following code prepares the data for use by the GOFstats macro:

```
data work.holdar1;
   set work.amr_AR1;
   Actual=exp(LogVolume);
   Forecast=exp(Forecast+0.5*STD*STD);
   keep Actual Forecast;
run;
```

Calling the macro is easy.

```
%GOFstats(ModelName=AR(1),DSName=work.holdar1,OutDS=work.ar1gofm,
          NumParms=2,ActualVar=Actual,ForecastVar=Forecast);
```

A different output data set is specified so that you can test the macro and compare it to the original code.

You are now ready to perform the calculations for all three models.

```
data work.holdma1;
   set work.amr_MA1;
   Actual=exp(LogVolume);
   Forecast=exp(Forecast+0.5*STD*STD);
   keep Actual Forecast;
run;

%GOFstats(ModelName=MA(1),DSName=work.holdma1,OutDS=work.ma1gofm,
          NumParms=2,ActualVar=Actual,ForecastVar=Forecast);

data work.holdar4;
   set work.amr_AR4;
   Actual=exp(LogVolume);
   Forecast=exp(Forecast+0.5*STD*STD);
   keep Actual Forecast;
run;

%GOFstats(ModelName=AR(4),DSName=work.holdar4,OutDS=work.ar4gofm,
          NumParms=5,ActualVar=Actual,ForecastVar=Forecast);
```

You can combine the results into one data set, and then sort by MAPE.

```
data work.gof;
   set work.ar1gofm
       work.ma1gofm
       work.ar4gofm;
run;

proc sort data=work.gof;
  by MAPE;
run;

title2 font=&COURSEFONT color=black "Model Summary Sorted by MAPE";
proc print data=work.gof noobs;
   var Model MAPE RMSE;
   format MAPE 6.2 RMSE 9.1;
run;
```

The following goodness-of-fit table shows that AR(4) is the preferred model based on MAPE.

AMR Daily Stock Volume
Model Summary Sorted by MAPE

Model	MAPE	RMSE
AR(4)	34.82	5876685.8
AR(1)	35.29	5858703.1
MA(1)	38.65	6132210.5

There is a risk in using MAPE to select a model using the same data that was used to fit the model. The statistics AIC and SBC help prevent an overfitting bias and would usually be recommended if you cannot use a holdout sample. For the AMR stock volume data, there is plenty of data to use a holdout sample. The three months of 2011 are added to a holdout sample.

```
data work.fitsample;
   set sasuser.amr;
   if (Date<='31DEC2010'd) then Y=LogVolume;
   else Y=.;
run;
```

The three models can be estimated using the fit sample, and forecasts can be obtained for the holdout sample.

```
proc arima data=work.fitsample;
   identify var=Y noprint;
   /*----  AR(1)  ----*/
   estimate p=1 method=ml noprint
            outstat=work.Stat_AR1;
   forecast lead=64 id=Date interval=weekday
            out=work.amr_AR1 noprint;
   /*----  MA(1)  ----*/
   estimate q=1 method=ml noprint
            outstat=work.Stat_MA1;
   forecast lead=64 id=Date interval=weekday
            out=work.amr_MA1 noprint;
   /*----  AR(4)  ----*/
   estimate p=4 method=ml noprint
            outstat=work.Stat_AR4;
   forecast lead=64 id=Date interval=weekday
            out=work.amr_AR4 noprint;
quit;
```

The previous temporary data sets are overwritten with the new data based on using a holdout sample.

```
data work.holdar1;
   merge work.fitsample work.amr_AR1;
   by Date;
   Actual=Volume;
   Forecast=exp(Forecast+0.5*STD*STD);
   if (Date>='01JAN2011'd) then output;
   keep Actual Forecast;
run;

%GOFstats(ModelName=AR(1),DSName=work.holdar1,OutDS=work.ar1gofm,
          NumParms=0,ActualVar=Actual,ForecastVar=Forecast);

data work.holdma1;
   merge work.fitsample work.amr_ma1;
   by Date;
   Actual=Volume;
   Forecast=exp(Forecast+0.5*STD*STD);
   if (Date>='01JAN2011'd) then output;
   keep Actual Forecast;
run;

%GOFstats(ModelName=AR(4),DSName=work.holdma1,OutDS=work.ma1gofm,
          NumParms=0,ActualVar=Actual,ForecastVar=Forecast);

data work.holdar4;
   merge work.fitsample work.amr_AR4;
   by Date;
   Actual=Volume;
   Forecast=exp(Forecast+0.5*STD*STD);
   if (Date>='01JAN2011'd) then output;
   keep Actual Forecast;
run;

%GOFstats(ModelName=AR(4),DSName=work.holdar4,OutDS=work.ar4gofm,
          NumParms=0,ActualVar=Actual,ForecastVar=Forecast);
```

The GOFstats macro is applied to the holdout data after replacing the missing actual values with the values from the original data. When you use a holdout sample, the number of parameters (NumParms) is set to zero.

```
data work.gof;
   set work.ar1gofm
       work.ma1gofm
       work.ar4gofm;
run;

proc sort data=work.gof;
  by MAPE;
run;

title2 font=&COURSEFONT color=black "Model Summary Sorted by MAPE";
proc print data=work.gof noobs;
   var Model MAPE RMSE;
   format MAPE 6.2 RMSE 9.1;
run;
```

The table of holdout MAPE and RMSE statistics follows:

AMR Daily Stock Volume
Model Summary Sorted by MAPE

Model	MAPE	RMSE
AR(4)	32.38	6980718.4
AR(1)	32.53	7012776.7
AR(4)	34.14	7237249.7

The AR(1) model looks competitive using a holdout sample.

Simulating a Retrospective Study

1. Divide the time series data into two segments. The *fit sample* is used to derive a forecast model. The *holdout sample* is used to evaluate forecast accuracy.
2. Derive a set of candidate models.
3. Calculate the chosen model accuracy statistic by forecasting the holdout sample.
4. Pick the model with the best accuracy statistic.

90

Choosing the Holdout Sample

- If possible, choose enough time points to cover a complete seasonal period. For example, for monthly data, hold out at least 12 observations.
- The holdout sample is always at the end of the series.
- If unique behavior occurs within the holdout sample, do not use a holdout sample. Instead, base accuracy calculations on the entire series.
- If there is insufficient data to fit a model without the holdout sample, then do not use a holdout sample. Again, base accuracy calculations on the entire series.

91

Rules of Thumb

- At least four time points are required for every parameter to be estimated in a model.
- Anything above the minimum series length can be used to create a holdout sample.
- Holdout samples should rarely contain over 25% of the series.

92

Summary of Data Used for Forecast Model Building

Fit Sample	Holdout Sample
Used to estimate model parameters for accuracy evaluation **Used to forecast values in holdout sample**	**Used to evaluate model accuracy** **Simulates retrospective study**

⚠ **Full = Fit + Holdout** data is used to fit a deployment model.

93

Forecasting the Holdout Sample

Forecast for	Beginning of						
	DEC 2007	JAN 2008	FEB 2008	MAR 2008	APR 2008	MAY 2008	JUN 2008
DEC 2007	1-step	Actual	Actual	Actual	Actual	Actual	Actual
JAN 2008	2-step	1-step	Actual	Actual	Actual	Actual	Actual
FEB 2008	3-step	2-step	1-step	Actual	Actual	Actual	Actual
MAR 2008	4-step	3-step	2-step	1-step	Actual	Actual	Actual
APR 2008	5-step	4-step	3-step	2-step	1-step	Actual	Actual
MAY 2008	6-step	5-step	4-step	3-step	2-step	1-step	Actual
JUN 2008	7-step	6-step	5-step	4-step	3-step	2-step	1-step

94 ...

Forecasting the Holdout Sample

Forecast for	Beginning of						
	DEC 2007	JAN 2008	FEB 2008	MAR 2008	APR 2008	MAY 2008	JUN 2008
DEC 2007	1-step	Actual	Actual	Actual	Actual	Actual	Actual
JAN 2008	2-step	1-step	Actual	Actual	Actual	Actual	Actual
FEB 2008	3-step	2-step	1-step	Actual	Actual	Actual	Actual
MAR 2008	4-step	3-step	2-step	1-step	Actual	Actual	Actual
APR 2008	5-step	4-step	3-step	2-step	1-step	Actual	Actual
MAY 2008	6-step	5-step	4-step	3-step	2-step	1-step	Actual
JUN 2008	7-step	6-step	5-step	4-step	3-step	2-step	1-step

Column: Multi-step ahead forecast accuracy
Diagonal: 1-step ahead forecast accuracy
Off-diagonal: *k*-step ahead forecast accuracy

95

A common practice involves creating forecast triangles that are similar to loss triangles in insurance loss reserving. As each time period passes, forecasts are evaluated and updated.

1.04 Multiple Choice Poll

Which of the following statements is ***true*** with respect to the Ljung-Box statistical test for white noise?

a. The Ljung-Box chi-square test assumes Gaussian normality and tests for independence.
b. You need only calculate one chi-square statistic for a given time series or set of residuals.
c. The number of sample autocorrelations to use for the chi-square statistic must be a multiple of 6.
d. The statistic has an exact chi-square distribution as long as the number of autocorrelations m is small with respect to the length of the series n.

Details

Aesthetics for the Stock Market Data

The stock market volumes in the previous sections were log transformed, giving the series superior statistical properties such as less skewness and thus supporting the forecast interval computations done in PROC ARIMA. Transforming back to the original scale is desirable when you show the results to a non-statistical audience. The logarithm is a monotone increasing transformation, meaning that percentiles, such as the median or the endpoints of a confidence interval on the log scale, can simply be exponentiated. To see this, suppose, for example, that on the log scale, 50% of the observations are less than the log scale median M. All observations X that are less than M have exp(X) < exp(M). Because 50% of the X values are less than M, you find that 50% of the exp(X) values are less than exp(M). The same logic applies to any percentile.

Although percentiles and thus confidence limits on the log scale can simply be exponentiated to get back to the original scale, the same is not true of the mean. The numbers X = 10, 0, and -10 on the (natural or base e) log scale average out to 0, and exp(0)= e^0=1. You see that the median on the original scale is 1, but is 1 also the mean of the three exp(X) values? The values when exponentiated give e^{10} = 22028, 1, and 1/22028 = 0.000045, the average of which is approximately 22029/3=7757, a far cry from e^0=1. If X=log(Y) is normal with mean μ and variance σ^2 (written as $X \sim N(\mu,\sigma^2)$), then Y has a lognormal distribution with mean $\exp(\mu + \sigma^2/2)$, which is greater than $\exp(\mu)$. For this reason, some forecasters add one half of the variance of the white noise error term to the log scale forecast before exponentiation. Generally speaking, when the forecast alone is exponentiated, this can be interpreted as an estimate of the median of the distribution from which the next observation comes. With the adjustment for variance applied, the exponentiated value can be thought of as an estimate of that distribution's mean. (For the mathematically oriented reader, the development of this mean formula is at the end of this section.) Also notice that for this adjustment to be helpful, you want the estimated white noise variance to be close to the true white noise variance. Otherwise, the adjustment might do more harm than good. This suggests the use of the adjustment for large data sets (with errors that appear to be normal on the log scale as is the case for the AMR data).

Data whose log values are normally distributed are called *lognormal* data and are said to be *lognormally distributed*.

A feature of the stock market data is the closure of the market on the weekends and holidays. If you chose to count in trading day time rather than calendar time, you have to align forecasts with the correct dates.

Mean of a Lognormal Variable

For a lognormal variable *Y* with (normally distributed) logarithm *X*=log(*Y*), suppose *X* has mean μ and variance σ^2. The mean of *Y* is exp(μ + σ^2/2). For those interested in the mathematical detail, the expected value (mean) of *Y*=exp(*X*) is developed as follows:

$$E\{Y\} = E\{e^x\} = \frac{1}{\sqrt{2\pi}}\int e^x e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2} dx = \frac{1}{\sqrt{2\pi}}\int e^{-\left(\frac{x^2-2\mu x+\mu^2-2x\sigma^2}{2\sigma^2}\right)} dx =$$

$$\frac{1}{\sqrt{2\pi}}\int e^{-\left(\frac{x^2-2(\mu+\sigma^2)x+(\mu^2+2\mu\sigma^2+\sigma^4)-(2\mu\sigma^2+\sigma^4)}{2\sigma^2}\right)} dx = e^{\left(\frac{(2\mu\sigma^2+\sigma^4)}{2\sigma^2}\right)} \frac{1}{\sqrt{2\pi}}\int e^{-\frac{1}{2}\left(\frac{x-(\mu+\sigma^2)}{\sigma}\right)} dx = e^{\mu+\sigma^2/2}$$

The last step follows because the expression before the = sign involves the integral of a new normal distribution, and that integral is 1.

Exercises

1. Complete demonstration 1.

2. Complete demonstration 2.

3. Complete demonstration 3.

4. Combine demonstration 2 and 3, except use **SASUSER.AMRMONTHLY**. Consider using only the AR(1) and MA(1) models for **Volume**. Employ a log transformation if you determine that the volumes are skewed.

1.4 Chapter Summary

There are many competing strategies and methods for forecasting. SAS software supports most major forecasting methodologies. The Box-Jenkins forecasting methodology of IDENTIFY-ESTIMATE-FORECAST provides a useful framework for carrying out a forecasting project.

PROC ARIMA provides basic Box-Jenkins forecasting capabilities. In particular, PROC ARIMA provides diagnostic statistics and plots for the IDENTIFY step.

PROC TIMESERIES can also be used for exploring the properties of a time series.

SAS/GRAPH software and ODS Graphics provide the capability for customizing plotted output.

Time series analysis includes proposing models and the testing accuracy of the models. In general, a model contains one or more of the four time series components: cyclical, seasonal, trend, error. Accuracy statistics help to guide the forecasting professional to a good model.

For Additional Information

Armstrong, J.S., editor. 2001. *Principles of Forecasting*. Boston: Kluwer Academic Publishers.

Bowerman, B.L., and R.T. O'Connell. 1993. *Forecasting and Time Series: An Applied Approach*. Belmont, California: Duxbury Press.

Box, G.E.P., and G.M. Jenkins. 1976. *Time Series Analysis: Forecasting and Control*. Oakland, California: Holden-Day.

Box, G.E.P., G.M. Jenkins, and G.C. Reinsel. 2008. *Time Series Analysis: Forecasting and Control, Fourth Edition*. Hoboken, New Jersey: John Wiley and Sons.

Darby, Michael R. 1976. "Three-and-a-half Million Employees Have Been Mislaid: or an Explanation of Unemployment 1934-1941," *Journal of Political Economy*, February, 1976, 84, 1-16.

Gilliland, Michael. 2010. *The Business Forecasting Deal: Exposing Myths, Eliminating Bad Practices, Providing Practical Solutions*. Hoboken, New Jersey: John Wiley and Sons.

Gilovich, Thomas, Dale Griffin, and Daniel Kahneman. 2002. *Heuristics and Biases: The Psychology of Intuitive Judgment*. New York: Cambridge University Press.

Kahneman, D., P. Slavic, and A. Tversky. 1982. *Judgment under Uncertainty: Heuristics and Biases*. New York: Cambridge University Press.

Margo, Robert A. 1993. "Employment and Unemployment in the 1930's," *Journal of Economic Perspectives*, 1993, 7, 41-59.

Tufte, Edward R. 1983. *The Visual Display of Quantitative Information*. Cheshire, Connecticut: Graphics Press.

Walton, Gary M., and Hugh Rockoff. 2005. *History of the American Economy, Tenth Edition*. Mason, Ohio: Thomson South-Western.

1.5 Solutions

Solutions to Exercises

1. See course notes.
2. See course notes.
3. See course notes.
4. See **Exercises_Ch1.sas**.

Solutions to Student Activities (Polls/Quizzes)

1.01 Multiple Answer Poll – Correct Answers

Which of the following are statistical time series?

a. The number of federal tax returns processed annually for the past 10 years
b. Individual ATM deposit amounts for the last 12 months
c. The ages of all current pilots for an airline when the pilots obtained the airline transport pilots' licenses
d. Total circulation for the Sunday edition of a newspaper for each week over the last six months
e. A life table for Belgian males listing the probability of survival to a specified age in years

16

1.04 Multiple Choice Poll – Correct Answer

Which of the following statements is ***true*** with respect to the Ljung-Box statistical test for white noise?

a. The Ljung-Box chi-square test assumes Gaussian normality and tests for independence.
b. You need only calculate one chi-square statistic for a given time series or set of residuals.
c. The number of sample autocorrelations to use for the chi-square statistic must be a multiple of 6.
d. The statistic has an exact chi-square distribution as long as the number of autocorrelations m is small with respect to the length of the series n.

99

Chapter 2 Stationary Time Series Models

2.1 Introduction to Stationary Time Series

Objectives

- Illustrate and define a stationary time series.
- Relate the diagnostics provided by the three autocorrelation functions to the analysis of a stationary time series.
- Relate the Box-Jenkins forecasting methodology to the modeling of a stationary time series.

3

Stationarity

- A stationary time series has a constant mean and variance.
- A time series with long-term trend or seasonal components cannot be stationary because the mean of the series depends on the time that the value is observed.
  For example, mean(December)>mean(February), meaning that you expect sales in December to be higher than sales in February.
- PROC ARIMA assumes that pure autoregressive models are for stationary time series and constrains the parameter estimates to enforce this assumption.

4

Stationary models produce forecasts that always converge to the constant mean of the stationary series. (Reversion to the mean was introduced in Chapter 1.)

Stationary Time Series

- To understand the general univariate single variable model and the general univariate multiple variable model, you must understand how to model a stationary time series.
- A stationary time series can be short, moderate, or long memory.
- A long memory stationary time series generates forecasts that revert to the mean much more slowly than a short memory time series.
- To obtain accurate forecasts for many steps into the future, you want a time series to be nonstationary, but if it is stationary, you want it to be long memory.

5

A long memory stationary process "remembers" many time points in the past. The definition of long memory can be vague. Suppose a stationary monthly time series has nonzero autocorrelation at lag 24. The current value is influenced by what happened 24 months in the past. Forecasts for this long memory process can go 24 months into the future before converging to the mean of the series.

2.01 Multiple Choice Poll

Which of the following statements is true?

a. Time series forecasting methods require that the time series of interest be stationary.
b. A trigonometric sine wave is stationary because it cycles about a constant value.
c. Forecasts for stationary time series always converge to the mean of the series.
d. Long-term forecasts are more accurate for stationary time series than for nonstationary series.
e. Politicians want to help you.

7

The Autocorrelation Function

- The autocorrelation function (ACF) measures the dependence among observations in a time series.
- The autocorrelation at lag *k* is the correlation of observation *k* time units apart.
- The sample autocorrelation function is defined so that it is always the autocorrelation of a stationary process.
- A long memory stationary time series often has spikes at large lags of the ACF, but values near one for small lags also suggest long memory.
- When the sample ACF suggests that the time series is long memory, this also suggests that the series might be nonstationary.

9

Autocorrelation Plots—Review

- The autocorrelation function (ACF) plot is one diagnostic tool available to detect the presence of autocorrelation in data as well as to estimate the magnitude and sign of the autocorrelation.
- The partial autocorrelation function (PACF) at lag *k* is the autocorrelation between observation *k* time units apart adjusted for all autocorrelation for observations less than *k* time units apart.
- The inverse autocorrelation function (IACF) is similar to the PACF, but the IACF might be able to detect features when the PACF is ambiguous, and vice versa.
- One of the primary purposes of the autocorrelation plots is to suggest candidate models to use for forecasting.

10

Autocorrelation Plots—PROC TIMESERIES

Abbreviated general form of the TIMESERIES procedure:

```
PROC TIMESERIES <options>
                    PLOTS=(corr acf pacf iacf);
    VAR variable;
    ID variable INTERVAL=name;
RUN;
```

11

PROC ARIMA provides the essential diagnostic plots, so your use of PROC TIMESERIES might be limited. When you investigate nonstationary series, PROC TIMESERIES provides some useful decomposition options that are not available in PROC ARIMA.

Autocorrelation Plots—PROC ARIMA

General form of the ARIMA procedure:

```
PROC ARIMA <options>;
    IDENTIFY VAR=variable <options>;
    ESTIMATE <options>;
    OUTLIER <options>;
    FORECAST OUT=SAS-data-set <options>;
RUN;
```

12

continued...

Box-Jenkins ARMA Models

- A stationary (Gaussian) time series with an autocorrelation function that is zero at all positive lags is a *white noise* time series.
- A stationary time series that has at least one nonzero autocorrelation can be approximated by an ARMA model.
- AR: Autoregressive ⇒ Time series is a function of its own past.
- MA: Moving Average ⇒ Time series is a function of past shocks (deviations, innovations, errors, and so on).

13

The theory of ARMA modeling includes the important property that ARMA models are *universal approximators*. The technical definition of a universal approximator is beyond the scope of this course. A simple intuitive explanation captures the essence of the technical definition. If you are confronted with any arbitrary stationary process, that process has a true (population) autocorrelation function. Given that true autocorrelation function and a “closeness” metric D, there exists an ARMA process that has the same autocorrelation function to within D units of the true autocorrelation function. A version of the theory proves that any stationary process can be approximated by an ARMA model in the sense that the *spectral density function* of the ARMA model will be “close” to the true spectral density function.

✎ Universal approximators are pervasive in mathematical and statistical literature. Neural networks are universal approximators for a large class of functions. Many advanced calculus courses discuss Fourier series and Taylor series as universal approximators of a large class of mathematical functions. Rational polynomials are universal approximators and provide the foundation for numerical algorithms to approximate mathematical functions. For example, there is a rational polynomial approximation for calculating Gaussian probabilities, which otherwise would require tedious numerical integration techniques. Universal approximators are nice, but the theory only guarantees that an approximator exists, ***not*** that you can actually find one for a given model and data set.

Autoregressive Models

A time series that is a linear function of *p* past values plus error is called an *autoregressive process of order p*, denoted AR(*p*).

$$Y_t = \theta_0 + \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + \cdots + \phi_p Y_{t-p} + \varepsilon_t$$

14

The constant term is defined by the parameter θ_0 rather than ϕ_0 to be consistent with the more general ARMA model notation presented later. The alternate mean form of the model is given by the following:

$$(Y_t - \mu) = \phi_1 (Y_{t-1} - \mu) + \phi_2 (Y_{t-2} - \mu) + \cdots + \phi_p (Y_{t-p} - \mu) + \varepsilon_t$$

where

$$\theta_0 = \mu(1 - \phi_1 - \phi_2 - \cdots - \phi_p)$$

Recall that the population (theoretical) PACF and IACF drop to zero after lag p for an AR(p) model, so you would expect the sample functions to produce values that are not statistically significantly different from zero after lag p.

Moving Average Models

A time series that is a linear function of *q* past errors is called a *moving average process of order q*, denoted MA(*q*).

$$Y_t = \theta_0 + \varepsilon_t - \theta_1\varepsilon_{t-1} - \theta_2\varepsilon_{t-2} - \cdots - \theta_q\varepsilon_{t-q}$$

15

Recall that the population (theoretical) ACF drops to zero after lag q for an MA(q) model, so you would expect the sample ACF to produce values that are not statistically significantly different from zero after lag q. Also, recall that the mean and constant term for an MA(q) model are the same, $\mu = \theta_0$.

The Inverse Autocorrelation Function

If the model is…

$$Y_t = \theta_0 + \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + \varepsilon_t$$

…the dual model is…

$$Y_t = \theta_0 + \phi_1\varepsilon_{t-1} + \phi_2\varepsilon_{t-2} + \varepsilon_t$$

16

Interpretation of Diagnostic Plots (Review)

Autocorrelation Function

- ACF(k) that is the correlation between Y_t and Y_{t-k}
- Drops to zero after lag *q* for an MA(*q*) model
- Decays to zero for AR models
- Simple exponential decay for an AR(1) model

Partial Autocorrelation Function

- PACF(k) that is the last coefficient in regression of Y_t on Y_{t-1} , Y_{t-2} , …, Y_{t-k}
- Drops to zero after lag *p* for an AR(*p*) model
- Decays to zero for MA models

Inverse Autocorrelation Function

- ACF of the dual model
- Interpret as ACF then switch backshift operators

17

The mixed autoregressive moving average (ARMA) model produces autocorrelation functions that cannot be easily diagnosed, although they can be explained in the context of the AR and MA behavior described above.

The ARMA(1,1) Model

$$Y_t - \mu = \phi_1(Y_{t-1} - \mu) + \varepsilon_t - \theta_1\varepsilon_{t-1}$$

Alternate Parameterization:

$$Y_t = \theta_0 + \phi_1 Y_{t-1} + \varepsilon_t - \theta_1\varepsilon_{t-1}$$

$$\theta_0 = \mu(1 - \phi_1)$$

18

The ARMA(1,1) model can be considered when the autocorrelation function plots suggest either an MA(1) or an AR(1) model. For larger values of *p* and *q*, you must resort to other strategies for determining the ARMA orders, but after selecting a candidate model, you can examine the autocorrelation plots for consistency. For example, if you propose an ARMA(3,2) model, you should see spikes in the PACF and IACF at lag 3, and you should see a spike in the ACF at lag 2. The theoretical autocorrelation functions will have non-zero values at all lags; the above statement refers to the sample functions, where non-significant values at any lag are possible, and spikes tend to be pronounced for the lags indicated by the maximum order.

Autoregressive Moving Average Models

A time series that is a linear function of p past values plus a linear combination of q past errors is called an *autoregressive moving average process of order* (p,q), denoted ARMA(p,q).

$$Y_t = \theta_0 + \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + \cdots + \phi_p Y_{t-p} + \varepsilon_t - \theta_1 \varepsilon_{t-1} - \theta_2 \varepsilon_{t-2} - \cdots - \theta_q \varepsilon_{t-q}$$

19

continued...

Autoregressive Moving Average Models

Box and Jenkins place the series values on the left of the equal sign and error terms on the right of the equal sign, with every parameter except the constant term preceded by a minus sign.

$$Y_t - \phi_1 Y_{t-1} - \phi_2 Y_{t-2} - \cdots - \phi_p Y_{t-p} = \theta_0 + \varepsilon_t - \theta_1 \varepsilon_{t-1} - \theta_2 \varepsilon_{t-2} - \cdots - \theta_q \varepsilon_{t-q}$$

20

The above notation is employed by Box and Jenkins (1976). Many variations exist in the literature. Some authors use alpha instead of phi, and beta instead of theta. Some authors switch the signs. Letters such as X, Z, and W are used instead of Y. A common mistake made by users who want to write the model or the forecast equation is that the signs are reversed. PROC ARIMA produces the *factored form* of the model using polynomial expressions in B, the backshift operator. While this practice should clarify the correct sign to employ, it adds another level of complexity, namely writing a model using polynomial operators.

Box-Jenkins Methodology and ARMA Models

- Accurate forecasts can be obtained by using ARMA models to approximate the underlying process that generates the data. Finding the true model that generates the data is unnecessary.
- AR, MA, and ARMA models are universal approximators for stationary time series.
- Forecasters should choose the most parsimonious model to approximate the true model.
- An ARMA model might require fewer parameters than either a pure AR or a pure MA model to approximate the true model.

21

Box-Jenkins Modeling Methodology

Identify	Determine ARMA orders (p,q) using ACF, PACF, and the IACF.
Estimate	Fit the ARMA(p,q) model and assess the fit of the model.
Forecast	Produce forecasts using the best ARMA model that passes assessment.

The general Box-Jenkins modeling methodology addresses trend, seasonality, and regressors as well.

22

Box and Jenkins (1976) published a textbook that provides a comprehensive framework for forecasting. The textbook includes much of the research of the authors, but also accumulates research from a wealth of different sources. Box and Jenkins did not invent nor discover ARMA models, but their textbook is the primary reason that ARMA models and the Box-Jenkins method are so popular.

ARMA Model Properties

This demonstration illustrates the properties of eight different simulated stationary time series.

The program **Demo2_01ARMAacf.sas** creates autocorrelation plots for eight time series stored in the SAS table **SASUSER.ARMAEXAMPLES**. Labels associated with each series describe the model for the series. For example, the label for **Y1** is AR(1). PROC CONTENTS reveals the variables and their labels.

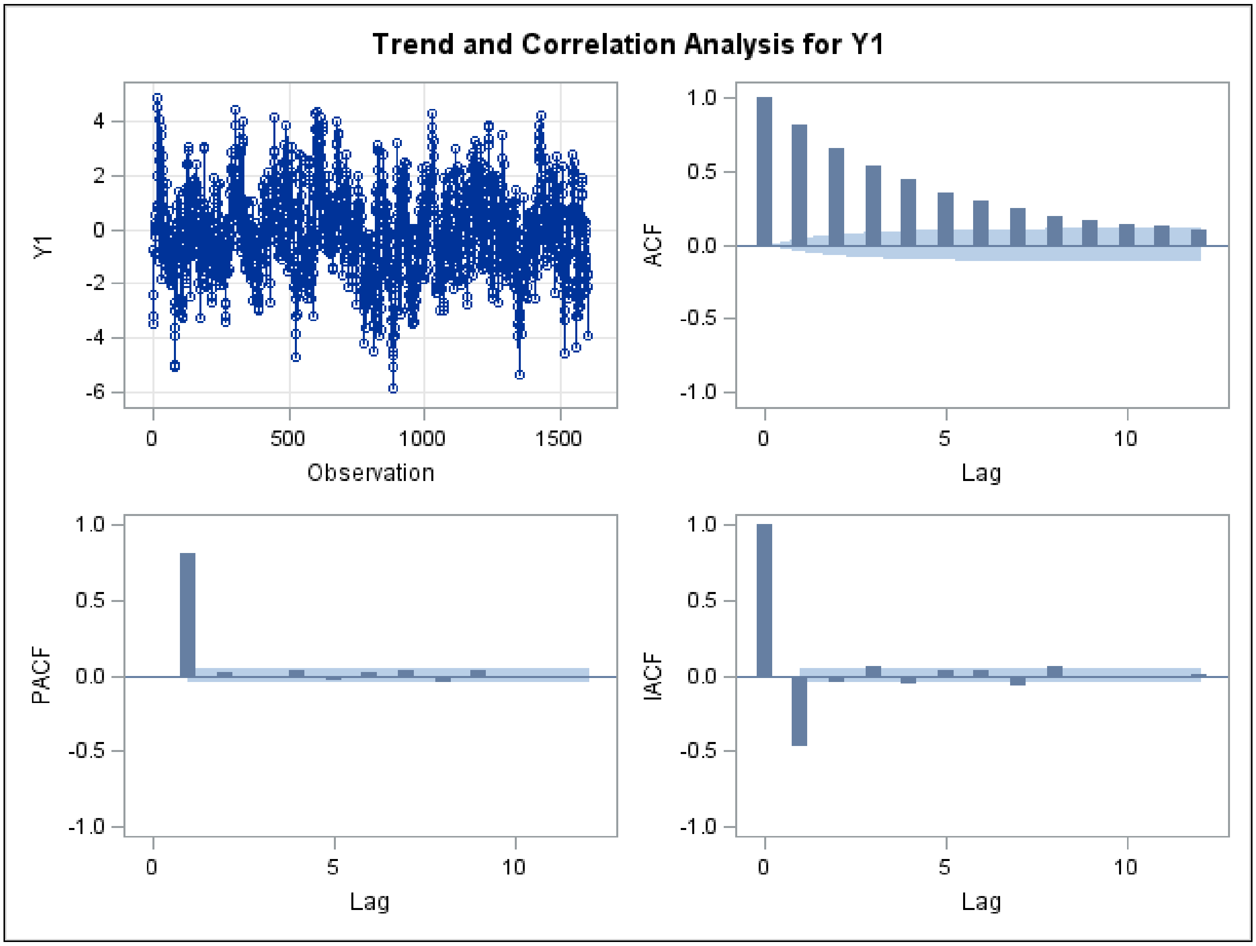
Alphabetic List of Variables and Attributes

#	Variable	Type	Len	Format	Label
2	Date	Num	8	DATE9.	Date
3	Y1	Num	8		AR(1)
4	Y2	Num	8		AR(2) roots .9, -.8
5	Y3	Num	8		AR(2) complex roots, period 9
6	Y4	Num	8		MA(1)
7	Y5	Num	8		MA(2) roots .9, -.8
8	Y6	Num	8		MA(2) complex roots, period 9
9	Y7	Num	8		ARMA(1,1)
10	Y8	Num	8		White Noise
1	t	Num	8		Time Index

PROC ARIMA produces the autocorrelation panel of plots for each series.

```
proc arima data=sasuser.armaExamples plots=series(corr);
   identify var=y1  nlag=12;
   identify var=y2  nlag=12;
   identify var=y3  nlag=12;
   identify var=y4  nlag=12;
   identify var=y5  nlag=12;
   identify var=y6  nlag=12;
   identify var=y7  nlag=12;
   identify var=y8  nlag=12;
quit;
```

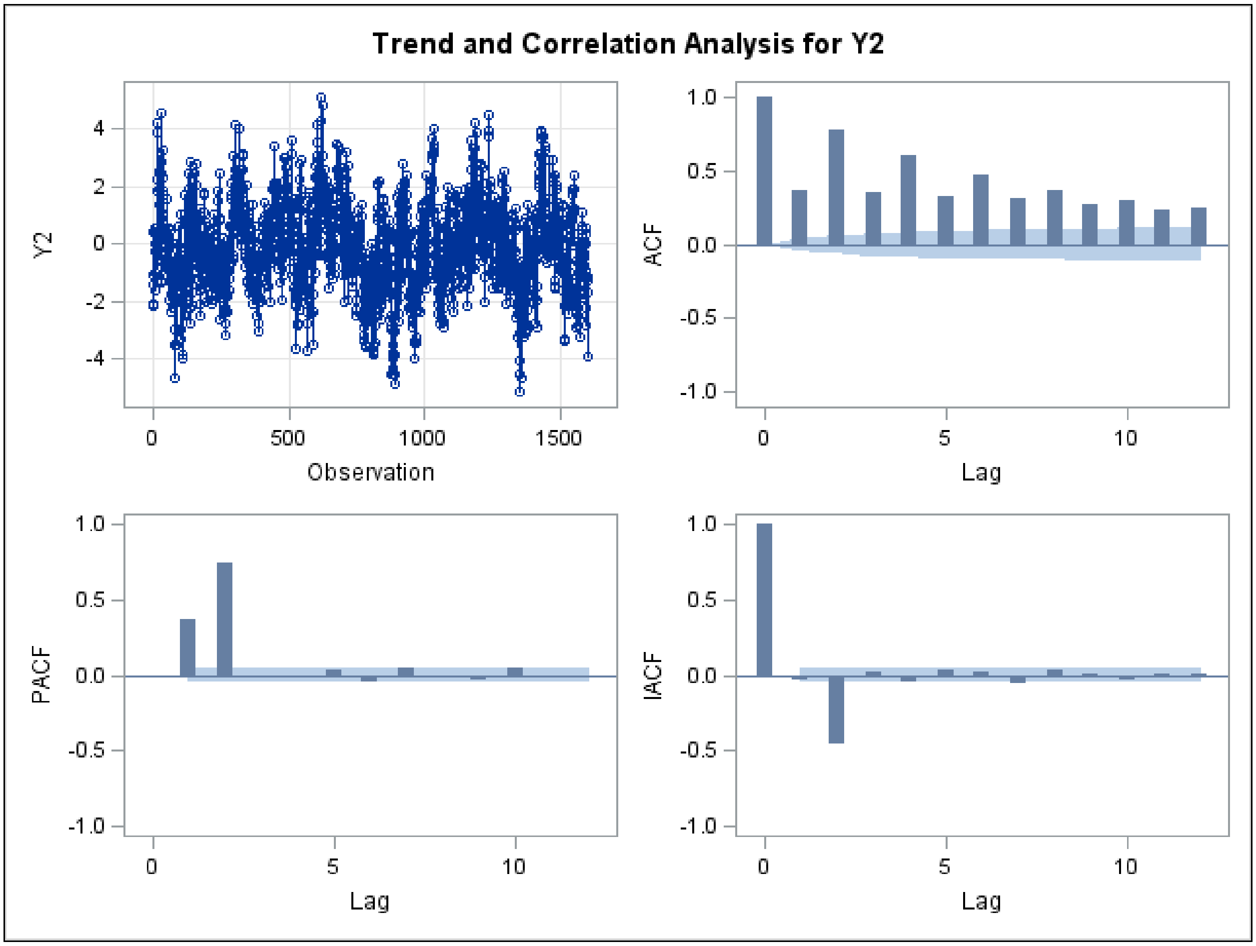
The plots are provided below with commentary.



An AR(1) model produces a single spike in the PACF and IACF at lag 1. The ACF shows decaying spikes that are decaying according to the following theoretical formula:

$$\text{ACF}(k) = \phi_1^k$$

For the simulation of the AR(1) time series, $\phi_1 = 0.8$.

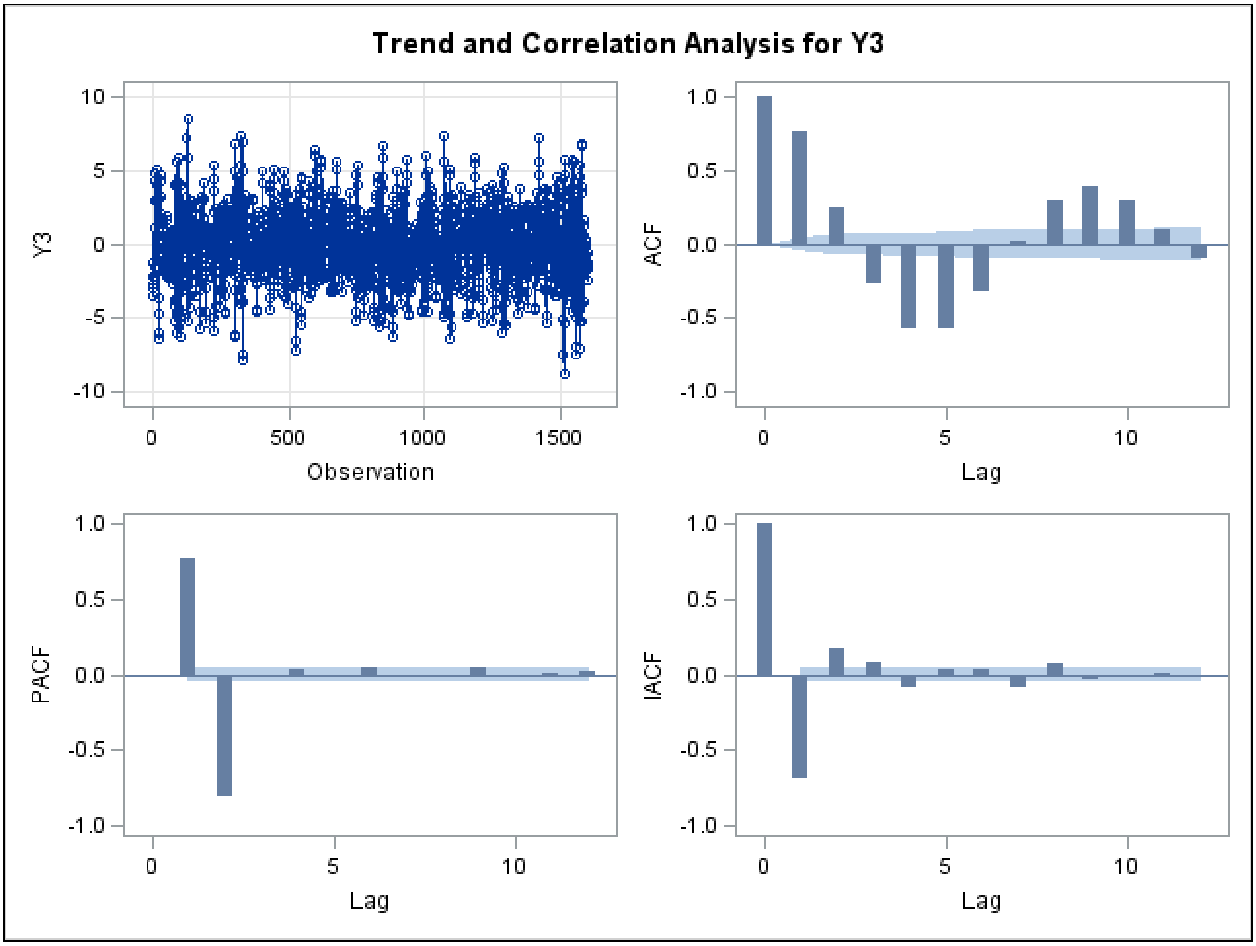


The Y2 series is generated using an AR(2) model with “roots” 0.9 and -0.8, yielding the following model:

$$(1-0.9B)(1+0.8B)Y_t = (1-0.1B-0.72B^2)Y_t = \varepsilon_t$$

The roots indicated in the label are actually the roots of the polynomial.

$$f(x) = x^2 - 0.1x - 0.72$$

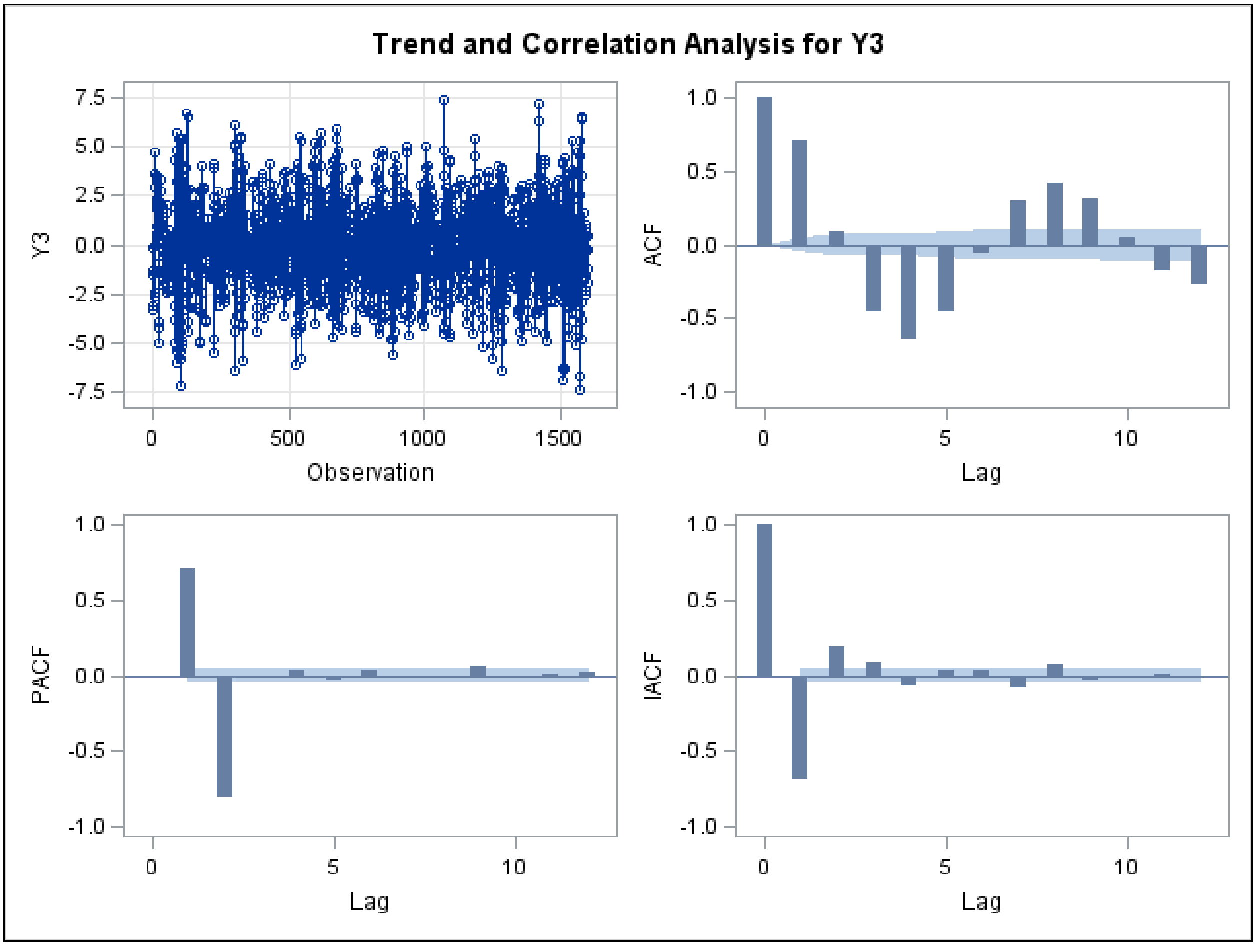


An AR(2) model has complex roots and a damped cosine with period of nine days. The ACF reveals the damped cosine behavior with a spike at lag 9 and at lags 4 and 5, corresponding to the harmonic at period 4.5. You can diagnose the AR(2) model by examining the PACF, but notice that the IACF shows a slight spike at lag 3.

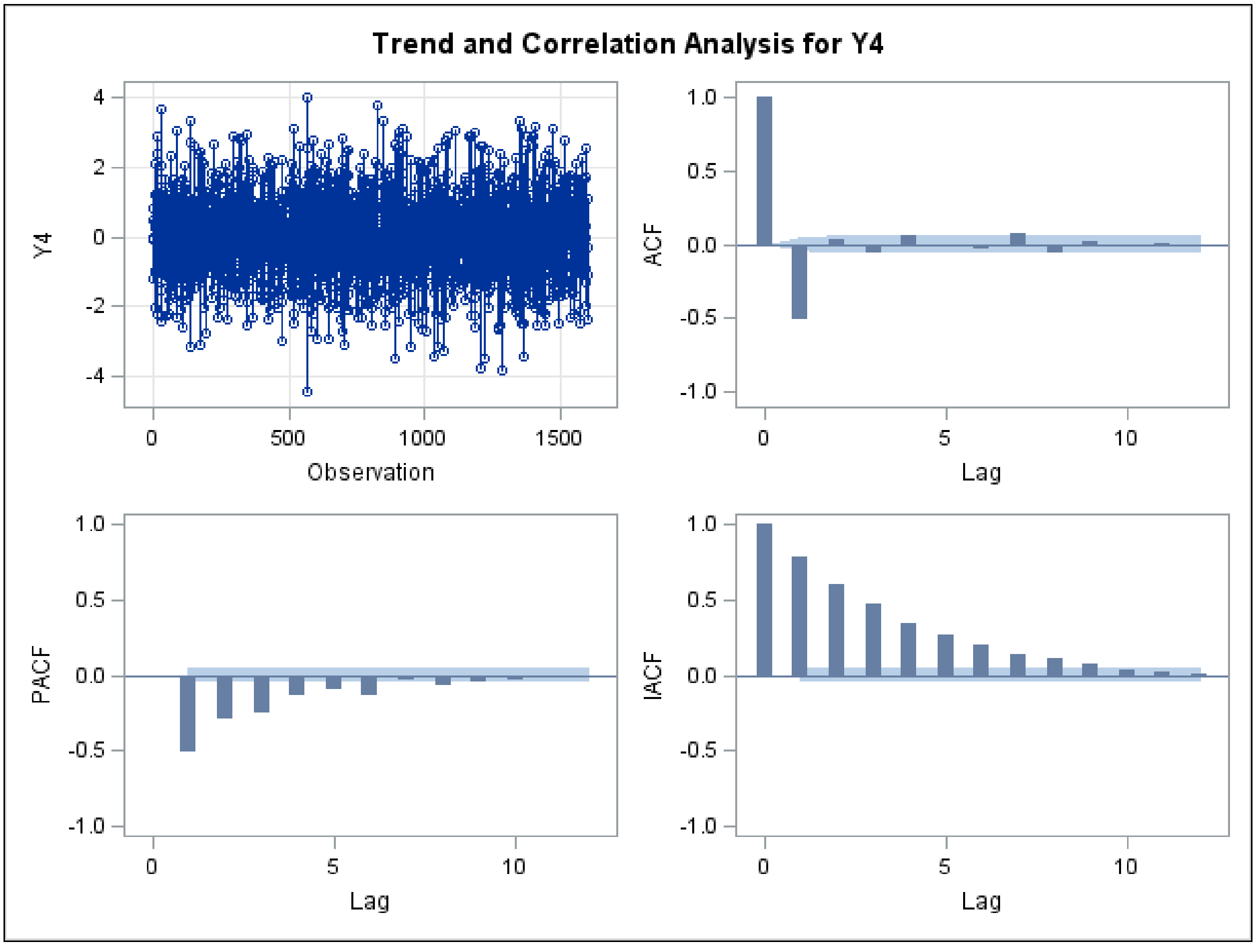
Box, Jenkins, and Reinsel (2008) observe that for an MA(2) process "[the PACF] is dominated by the sum of two exponentials if the roots … are real, and by a damped sine wave if the roots … are complex." They point out that this is the exact behavior of the ACF for an AR(2) process, again reflecting the duality of AR and MA models. Thus, the damped sine wave in the plot above is a result of the complex roots as well as the damped cosine. The damped cosine affects the period, but the shape is common for all AR(2) processes with complex roots.

To understand the effect of the damped cosine component, a different series, Y3, was simulated using a period of 8.

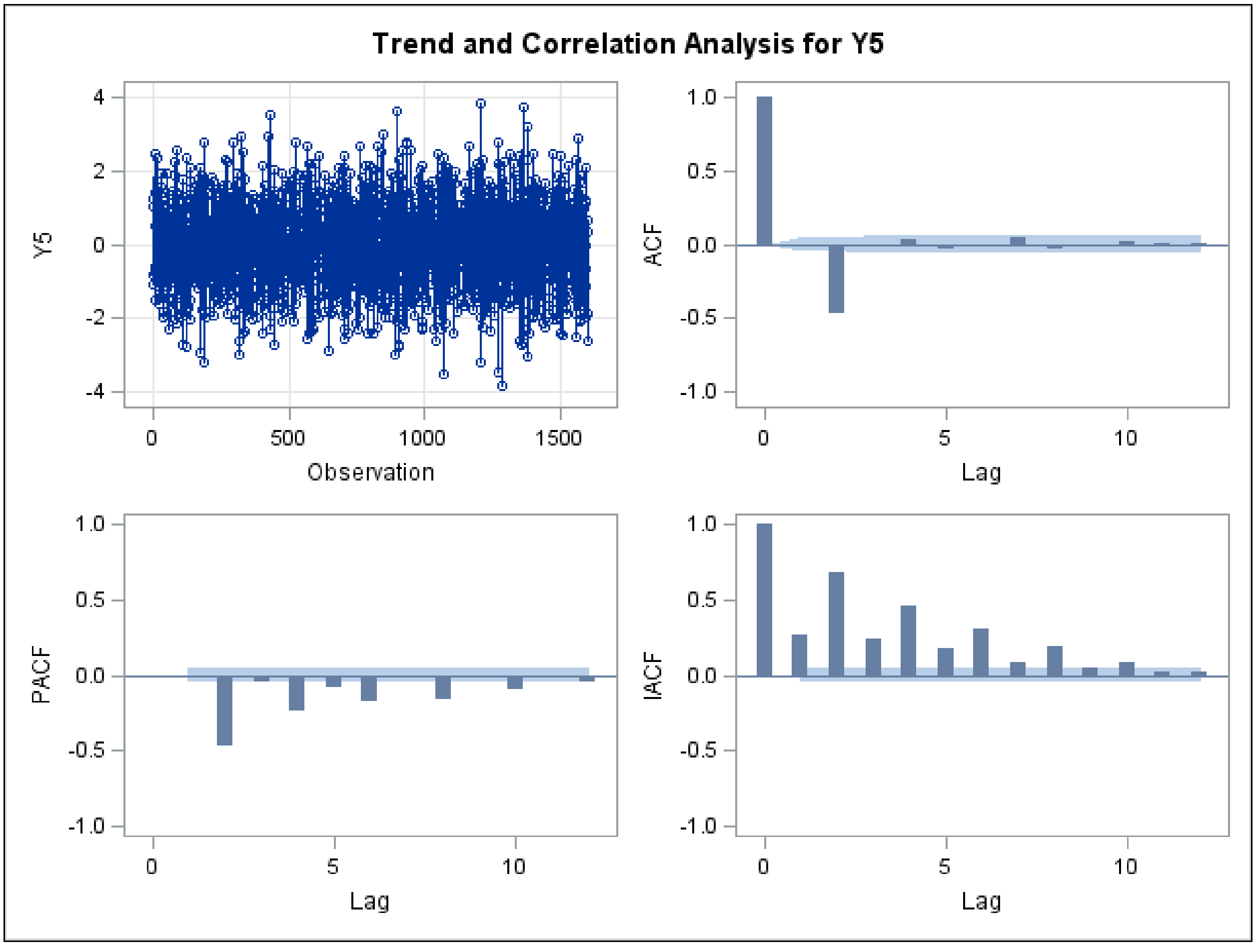
The following plots are produced by the new Y3 series.



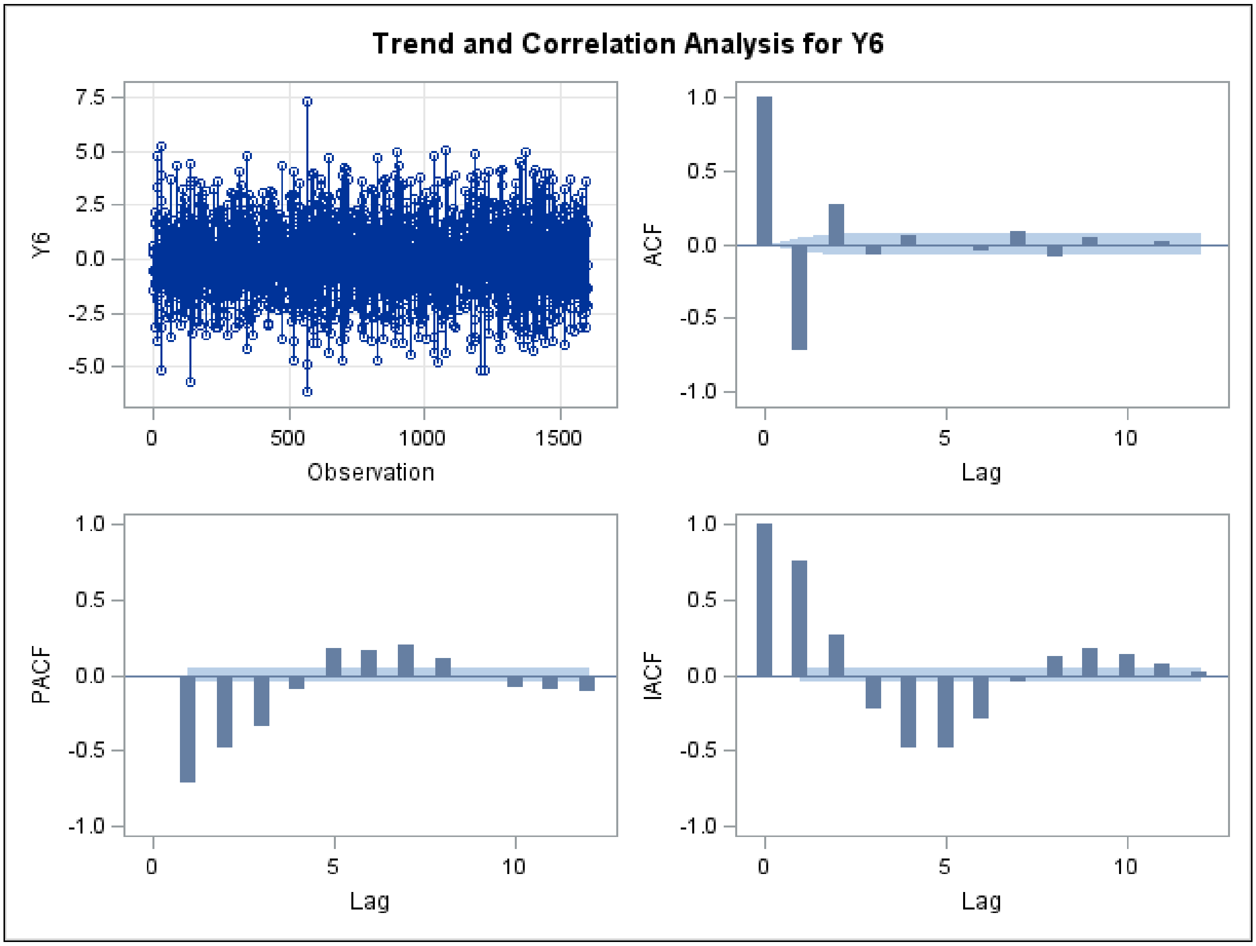
You can see that the spikes now appear in the ACF at lag 8, and at the harmonic corresponding to lag 4. (This series is not included in the course data, but can be easily produced by modifying the PERIOD macro variable in the **FETSP_CreateCourseData.sas** program.)



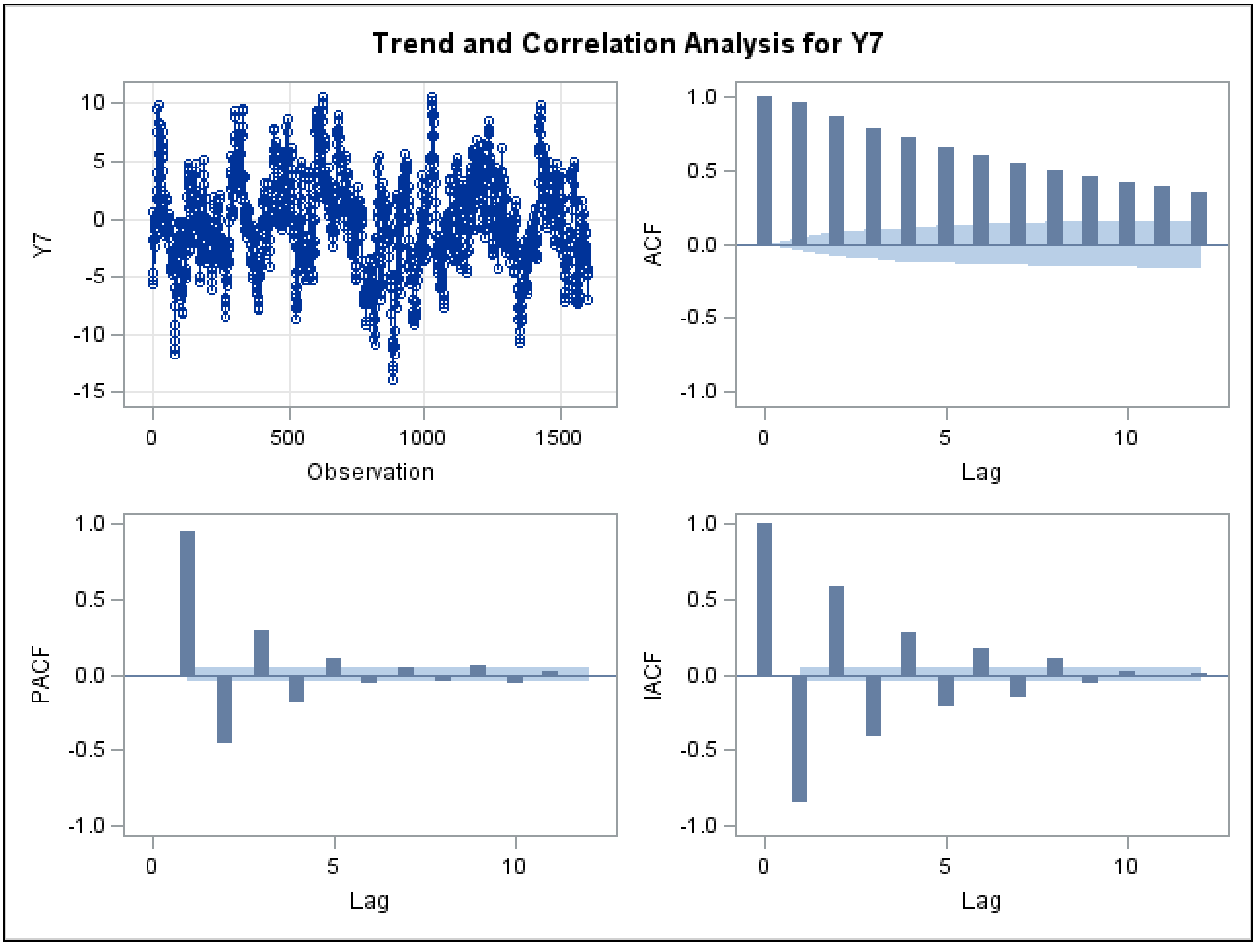
An MA(1) model generates the above plots. The ACF drops off after lag 1, and the PACF and IACF show an exponential decay pattern.



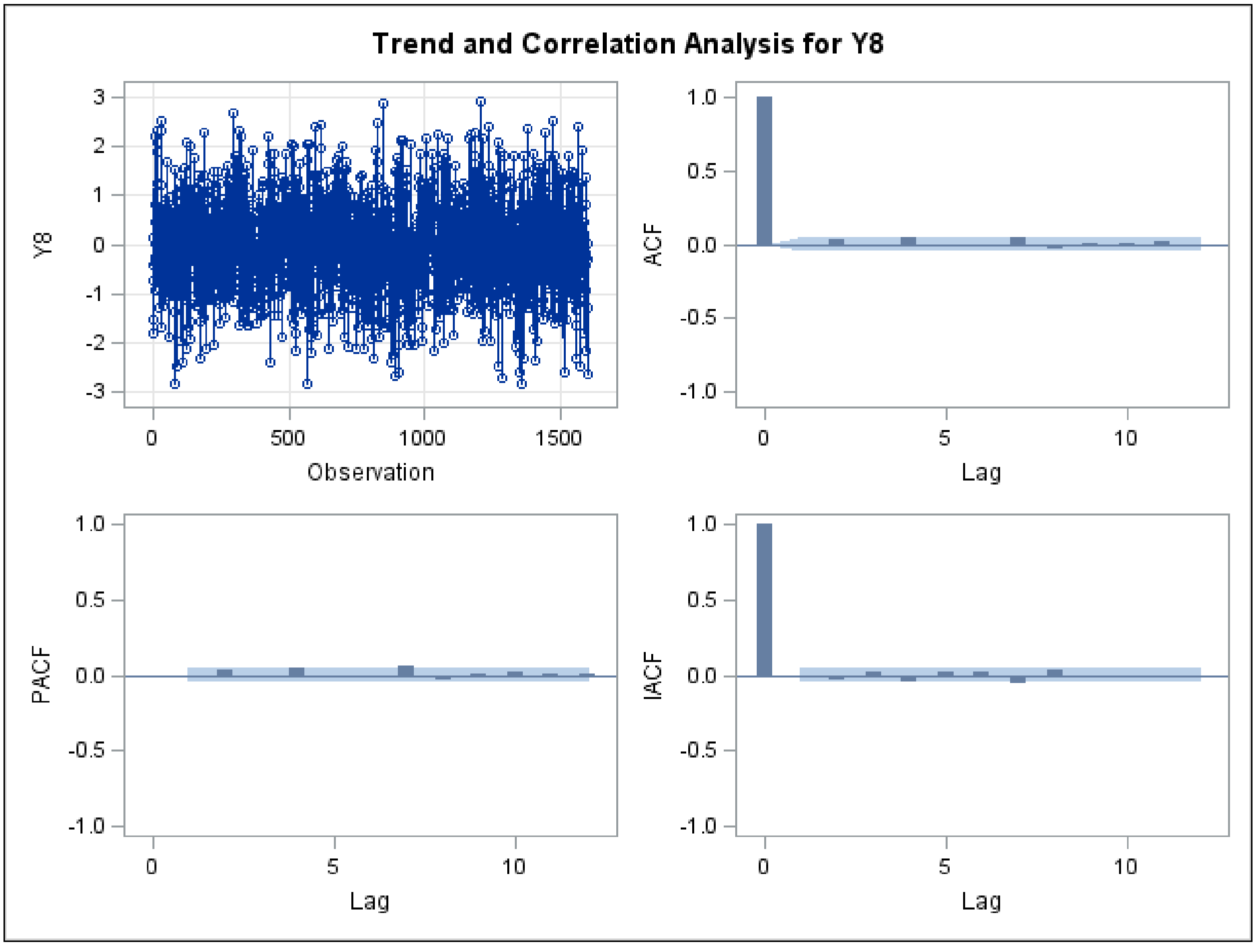
An MA(2) model that is actually the dual model for the AR(2) model generated the series Y2 that is simulated in the above series.



The model that generated the series that produced the above plots is an MA(2) model with complex roots and a damped cosine with period 9. As suggested above, when you discuss the behavior of the autocorrelation functions plots for series Y3, the complex roots produce a damped sine wave shape for the PACF and IACF.



An ARMA(1,1) model generated the data that produced the above plots. The ACF can be characterized as a hybrid of the spike at lag 1 for the MA(1) part of the model, with the remaining spikes coming from the exponential decay pattern due to the AR(1) part of the model. You quickly realize that ARMA models cannot be determined from examining the autocorrelation plots.



The final series is white noise, and no spikes appear at any lags.

Series Y3 and Y6 provide a new way to interpret the plots. Data exhibiting damped periodic behavior corresponding to sinusoidal type cycles tends to exhibit an ACF that also looks as if it has a damped sinusoidal pattern.

2.2 Automatic Model Selection Techniques for Stationary Time Series

Objectives

- Describe the three automatic order determining criteria supported by PROC ARIMA: ESACF, MINIC, and SCAN.
- Illustrate how to use the order-determining criteria.
- Describe a modification of the MINIC procedure that uses maximum likelihood estimation, and illustrate its use with a macro implementation of the technique.

25

You should be able to examine the sample autocorrelation plots and propose AR and MA models that are consistent with the plots. However, an ARMA model might exist that produces better forecasts than either the AR or MA model proposed by visual inspection. Furthermore, the concept of *parsimony* recommends choosing the model with the fewest parameters, and an appropriate ARMA(p,q) model often has p+q smaller than the p or q resulting from a pure AR or pure MA model. To help forecasters, research in the 1970s and 1980s focused on the problem of order determination for ARMA models. For example, see Tsay and Tiao (1984) and Tsay and Tiao (1985).

ARMA Order Determining Methods

- Extended Sample Autocorrelation Function (ESACF)
- Minimum Information Criterion (MINIC)
- Smallest Canonical Correlation (SCAN)

- P=(*m*:*n*) specifies that AR orders *m* through *n* (*m*<*n*) be investigated.
- Q=(*m*:*n*) specifies that MA orders *m* through *n* (*m*<*n*) be investigated.
- PERROR=(*m*:*n*) specifies the range of AR orders to investigate for the *approximating autoregressive model* that is used to generate ESACF, MINIC, or SCAN table values.

26

MINIC assumes that the time series is stationary. ESACF and SCAN permit the series to be nonstationary and provide diagnostics to indicate the nature of the nonstationarity.

ARMA Order Determining Methods

Extended Sample Autocorrelation Function (ESACF)

```
PROC ARIMA DATA=SAS-data-set;
    IDENTIFY VAR=variable
                  ESACF P=(Pmin:Pmax) Q=(Qmin:Qmax)
                  PERROR=(PEmin:PEmax);
RUN;
```

```
proc arima data=FETSP.Hurricanes;
   identify var=TropicalStorms nlags=12
            esacf p=(0:12) q=(0:12)
            perror=(3:12);
run;
```

27

continued...

ARMA Order Determining Methods

Extended Sample Autocorrelation Function (ESACF)

```
  ARMA(p+d,q)
   Tentative
     Order
   Selection
     Tests

  ---ESACF---
  p+d       q

    1       1
   10       7
    9       8
    0      10

(5% Significance Level)
```

Suggested Models*:
ARMA(1,1)
ARMA(10,7)
ARMA(9,8)
MA(10)

* Several nonstationary models are also suggested.

28

The suggested $p+d$ value implies that there might be a d^{th} difference. For example, if ESACF diagnoses $p+d=2$, then any of the following is possible:

- $p=2$ and $d=0$
- $p=1$ and $d=1$
- $p=0$ and $d=2$

Differencing provides one method for modeling nonstationary behavior. (The next chapter discusses differencing in detail.)

ARMA Order Determining Methods

Minimum Information Criterion (MINIC)

PROC ARIMA DATA=*SAS-data-set* **;**
    IDENTIFY VAR=*variable*
        MINIC P=(*Pmin:Pmax*) Q=(*Qmin:Qmax*)
        PERROR=(*PEmin:PEmax*)**;**
RUN;

```
proc arima data=FETSP.Hurricanes;
   identify var=TropicalStorms nlags=12
            minic p=(0:12) q=(0:12)
            perror=(3:12);
run;
```

29

continued...

ARMA Order Determining Methods

Minimum Information Criterion (MINIC)

```
...
Minimum Information Criterion

Lags      MA  7     MA  8     MA  9     MA 10     MA 11     MA 12

AR  0  1.562702  1.594901  1.623119  1.614338  1.630961  1.656413
AR  1  1.580082  1.592622  1.608107  1.632817  1.661636  1.688498
AR  2  1.605016   1.60998  1.638876  1.664341  1.692878  1.720179
AR  3  1.612545   1.64177   1.67027  1.691813  1.723189  1.751298
...
```

```
Error series model:  AR(3)
Minimum Table Value: BIC(1,1) = 1.428342
```

Suggested Model:
ARMA(1,1)

30

ARMA Order Determining Methods

Smallest Canonical Correlation (SCAN)

PROC ARIMA DATA=*SAS-data-set* ;
    IDENTIFY VAR=*variable*
        SCAN P=(*Pmin:Pmax*) Q=(*Qmin:Qmax*)
        PERROR=(*PEmin:PEmax*);
RUN;

```
proc arima data=FETSP.Hurricanes;
   identify var=TropicalStorms nlags=12
            scan p=(0:12) q=(0:12)
            perror=(3:12);
run;
```

31

continued...

ARMA Order Determining Methods

Smallest Canonical Correlation (SCAN)

```
   ARMA(p+d,q)
    Tentative
      Order
    Selection
      Tests

   ----SCAN---
   p+d       q

     1       1
    10       0
     0      10

(5% Significance Level)
```

Suggested Models*:
ARMA(1,1)
AR(10)
MA(10)

* Several nonstationary models are also suggested.

32

Identifying ARMA Orders

This demonstration illustrates how to use PROC ARIMA to identify orders of an ARMA model.

The program for this demonstration can be found in **Demo2_02ARMAid.sas**. Recall that series Y7 in data set **SASUSER.ARMAEXAMPLES** was generated by an ARMA(1,1) model. The following code produces candidate values for *p* and *q* for an ARMA(p,q) model:

```
proc arima data=sasuser.armaexamples;
   identify var=Y7 nlag=12 esacf scan minic;
quit;
```

The results containing the proposed orders are shown here:

Minimum Table Value: BIC(1,1) = -0.03145

ARMA(p+d,q) Tentative Order Selection Tests

SCAN			ESACF		
p+d	q	BIC	p+d	q	BIC
1	1	-0.03145	1	1	-0.03145

(5% Significance Level)

With 1,600 time points, the techniques get the model exactly correct. All three methods suggest an ARMA(1,1) model.

Tsay and Tiao (1984) and Peña, Tiao, and Tsay (2001) explain how to diagnose ARMA orders by examining the matrix of ESACF *p*-values.

The following visualization table was produced by the **VisualizeESACF** macro:

Orders	MA_0	MA_1	MA_2	MA_3	MA_4	MA_5	MA_6	MA_7	MA_8	MA_9	MA_10
AR 0	.	.	.	.	.	.	.	.	.	.	.
AR 1	.	+	+	+	+	+	+	+	+	+	+
AR 2	.	.	+	+	+	+	+	+	+	+	+
AR 3	.	.	+	+	+	+	+	+	+	+	+
AR 4	.	.	+	.	+	+	+	+	+	+	+
AR 5	.	.	.	.	.	+	+	+	+	+	+
AR 6	.	.	+	.	+	.	+	+	+	+	+
AR 7	.	.	+	.	+	.	+	+	+	+	+
AR 8	.	.	.	.	.	.	+	+	+	+	+
AR 9	.	.	.	.	+	.	.	+	.	+	+
AR 10	.	.	+	.	+	.	.	+	.	.	+

The + symbol corresponds to *p*-values greater than 0.05, and the periods, ., correspond to *p*-values smaller than 0.05. The triangle of + symbols with vertex at cell AR 1-MA_1 points to the appropriate orders. As is common with random variation, a few statistically insignificant values arise outside of the triangle, for example, at cell AR 6-MA_2. Nonetheless, you can often pick out a triangular shape in the ESACF table. Here is the original table of *p*-values:

RowName	MA__0	MA__1	MA__2	MA__3	MA__4	MA__5	MA__6	MA__7	MA__8	MA__9	MA_10
AR 0	<.0001	<.0001	<.0001	<.0001	<.0001	<.0001	<.0001	<.0001	<.0001	<.0001	<.0001
AR 1	<.0001	0.8131	0.7417	0.8129	0.5757	0.5875	0.7125	0.5370	0.6175	0.6909	0.7756
AR 2	<.0001	<.0001	0.8138	0.9475	0.3907	0.6940	0.8890	0.3147	0.7382	0.3865	0.7739
AR 3	<.0001	<.0001	0.0793	0.9965	0.6083	0.6451	0.9670	0.4594	0.8867	0.4221	0.6223
AR 4	<.0001	<.0001	0.1330	<.0001	0.9118	0.8750	0.9843	0.4234	0.8497	0.9443	0.6981
AR 5	<.0001	<.0001	<.0001	<.0001	<.0001	0.8669	0.9658	0.3286	0.8318	0.9711	0.9545
AR 6	<.0001	0.0002	0.6631	<.0001	0.4844	0.0045	0.9842	0.5064	0.5068	0.8228	0.8680
AR 7	<.0001	0.0002	0.6894	<.0001	0.7584	0.0331	0.0534	0.6113	0.8961	0.8856	0.8221
AR 8	<.0001	<.0001	<.0001	<.0001	<.0001	<.0001	0.0515	0.0815	0.9455	0.9126	0.8083
AR 9	<.0001	<.0001	<.0001	0.0220	0.7872	<.0001	0.0228	0.7937	<.0001	0.8358	0.7377
AR 10	<.0001	<.0001	0.1625	0.0040	0.8695	<.0001	0.0426	0.4901	<.0001	0.0160	0.9856

While you can develop the skill to interpret the raw table of *p*-values, it is easier to use a cutoff value such as 0.05 and produce a more visually appealing table. You can see in the above table that all of the + symbols in the visible triangle correspond to large *p*-values.

PROC ARIMA employs an algorithm to identify candidate cells for you, so you really do not need to develop skills for interpreting the table. The downside is that PROC ARIMA usually suggests more models than you might identify by visually scanning the table.

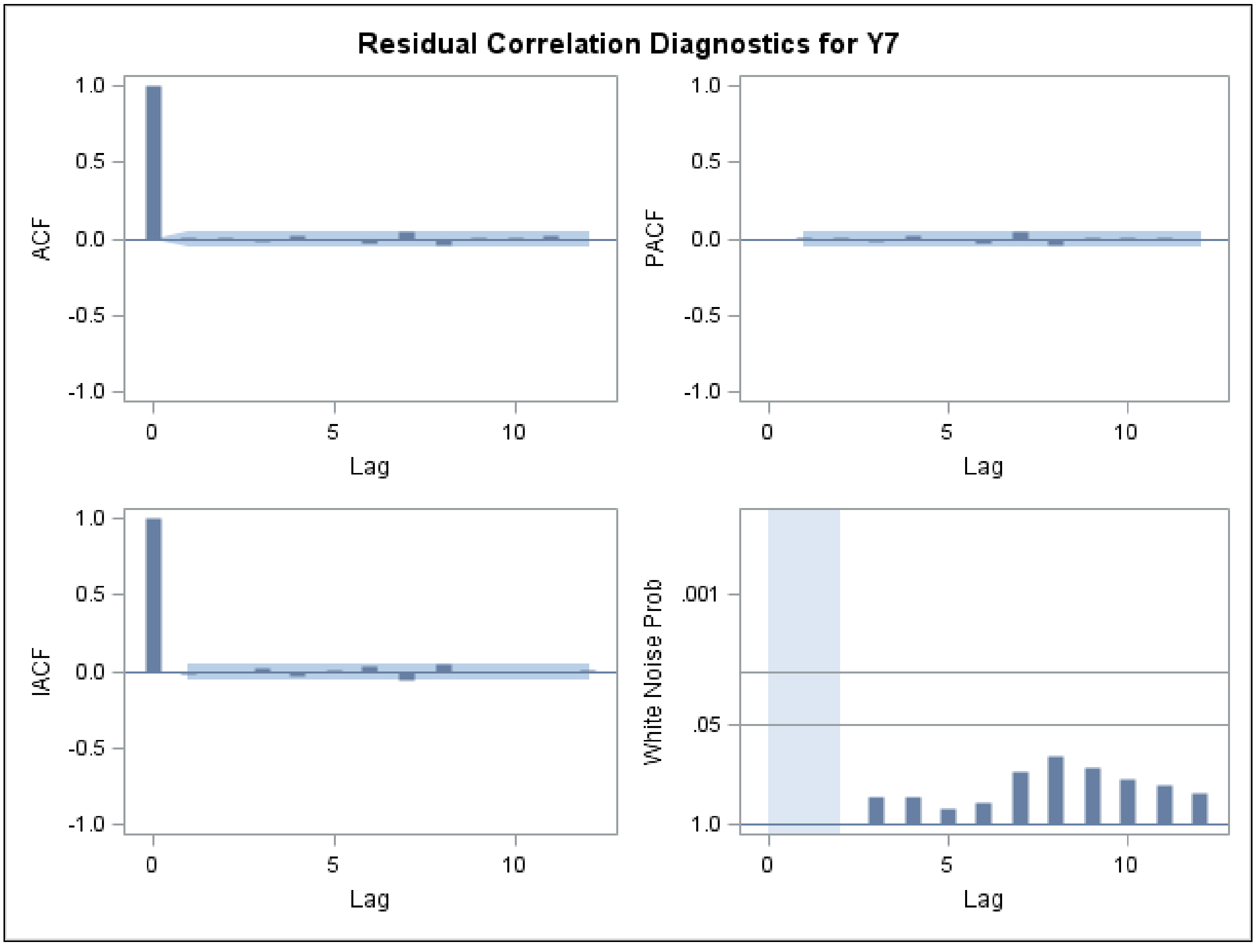
Estimation is discussed in the next section, but notice that you are now ready to propose and fit a model to the data. The following code estimates the parameters of an ARMA(1,1) model and performs diagnostic checking of the residuals:

```
proc arima data=sasuser.armaexamples;
   identify var=Y7 nlag=12 noprint;
   estimate p=1 q=1 ml;
quit;
```

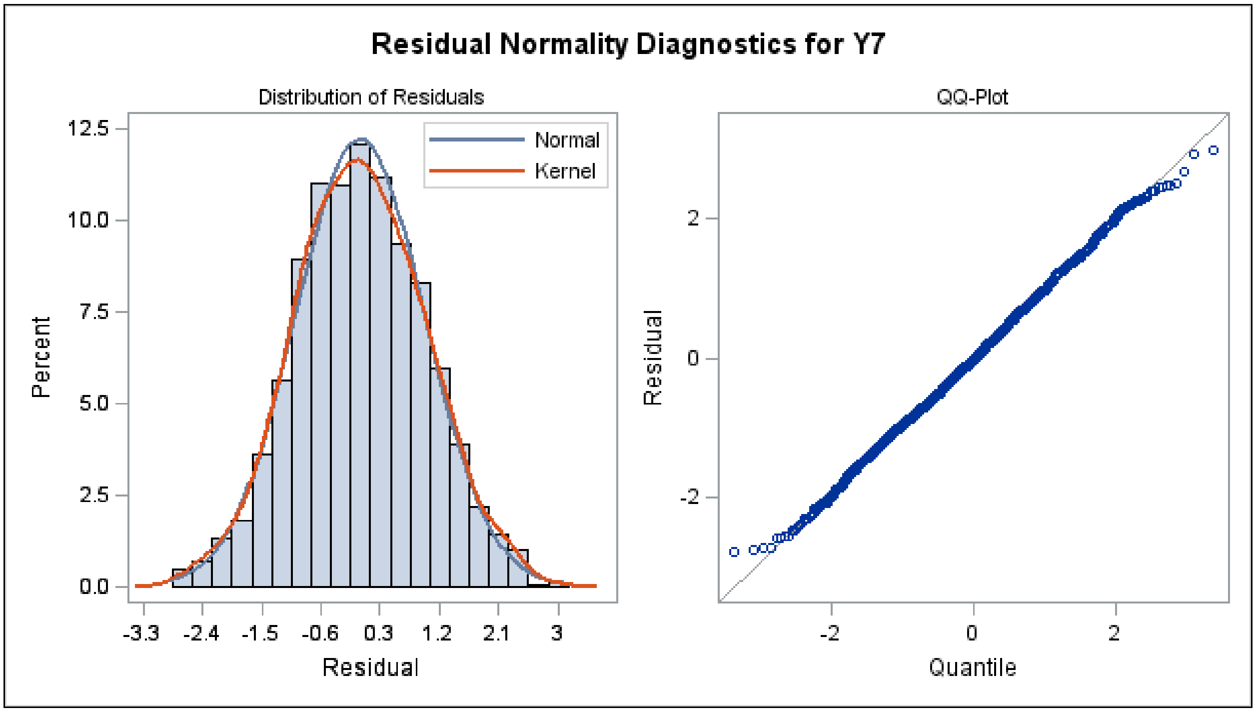
The estimated parameters are statistically significant.

Maximum Likelihood Estimation					
Parameter	Estimate	Standard Error	t Value	Approx Pr > |t|	Lag
MU	-0.43377	0.49074	-0.88	0.3767	0
MA1,1	-0.78158	0.01587	-49.26	<.0001	1
AR1,1	0.91152	0.01044	87.31	<.0001	1

The residual correlation plots suggest white noise.



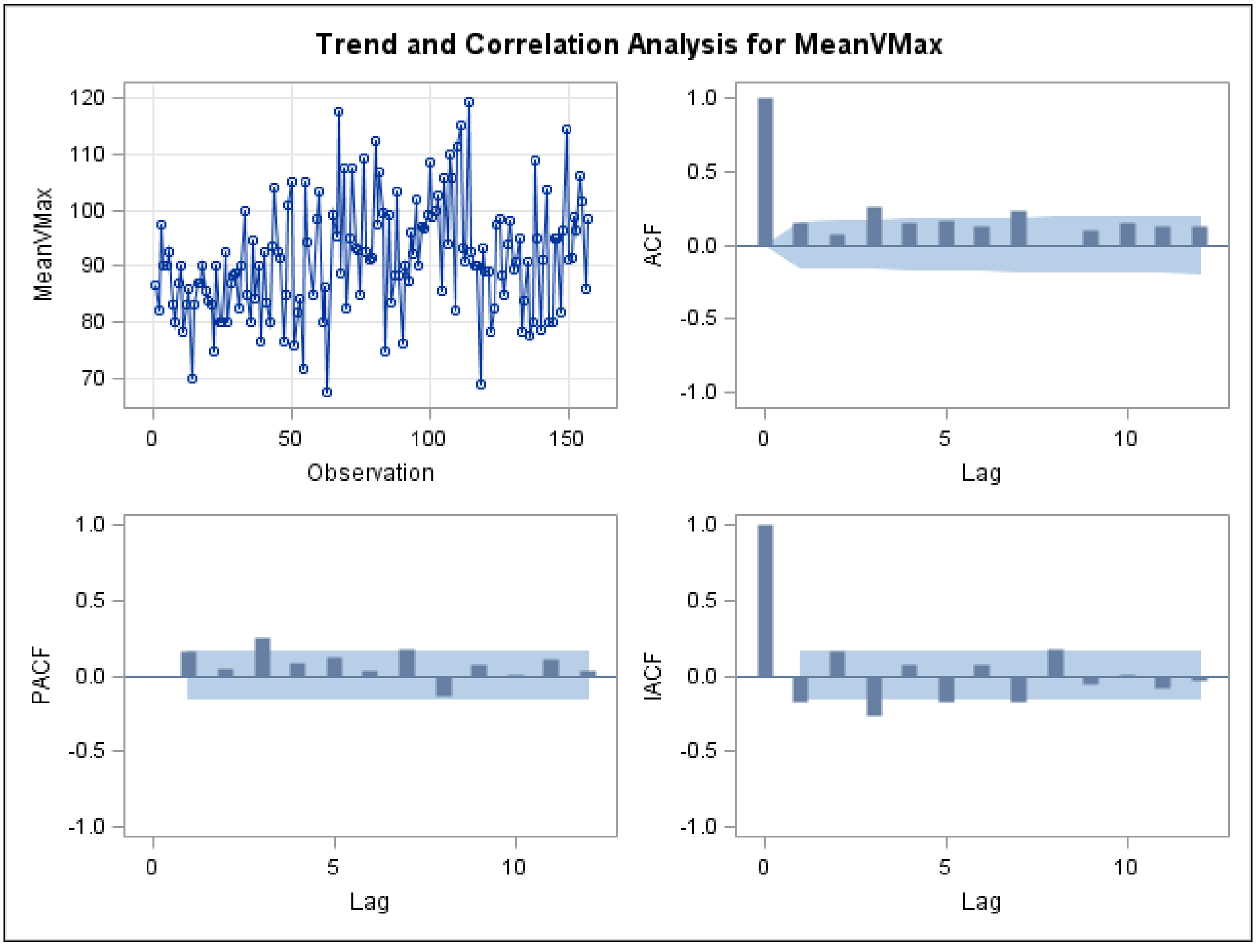
The ARMA(1,1) appears to be a promising forecast model. For such a large sample size, the residuals closely resemble data from a normal distribution.



Consider the problem of model selection for real data. The data set **SASUSER.HURRICANES** contains annual data about hurricanes in the Atlantic basin, including the Gulf of Mexico. The following code proposes models for **MEANVMAX**, the average maximum wind speed for hurricanes:

```
proc arima data=sasuser.hurricanes;
   identify var=meanvmax nlag=12
            esacf scan minic
            p=(0:12) q=(0:12)
            perror = (3:12);
quit;
```

The autocorrelation plots follow:



Spikes at higher lags for all autocorrelation plots suggest that an ARMA model might be appropriate.

Minimum Table Value: BIC(1,3) = 4.48364

ARMA(p+d,q) Tentative Order Selection Tests					
SCAN			ESACF		
p+d	q	BIC	p+d	q	BIC
1	1	4.54306	2	2	4.536147
7	0	4.561264	6	2	4.59087
0	7	4.581544	8	3	.
			9	3	.
			0	7	4.581544
			1	7	4.602286
			12	5	.

(5% Significance Level)

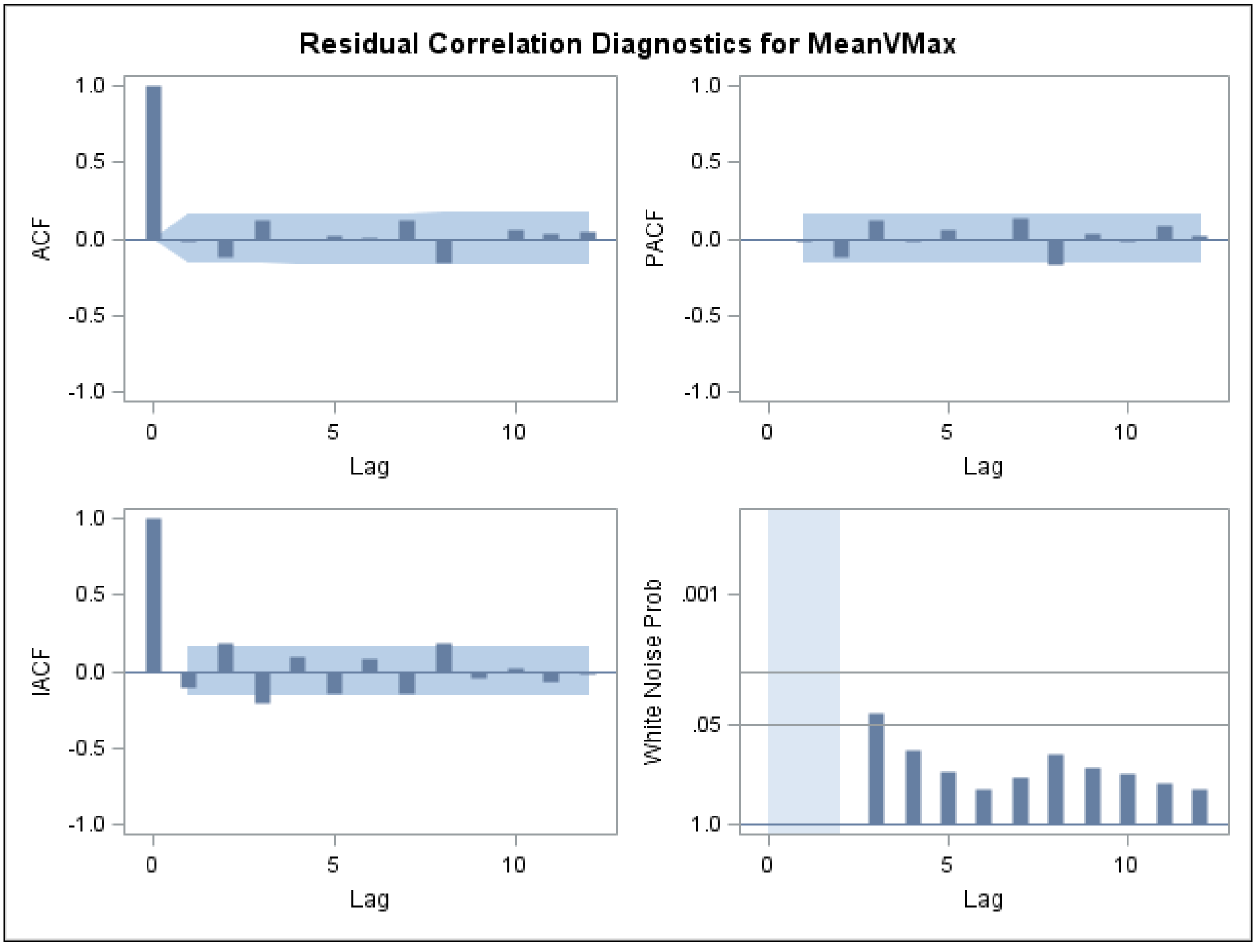
While the higher order models might be adequate, you can focus attention on the three "best" models, namely ARMA(1,2) (MINIC), ARMA(1,1) (SCAN) and ARMA(2,2) (ESACF). Consider first the ARMA(1,1) model because it is the most parsimonious.

```
proc arima data=sasuser.hurricanes;
   identify var=meanvmax nlag=12 noprint;
   estimate p=1 q=1 ml;   *  scan choice  *;
quit;
```

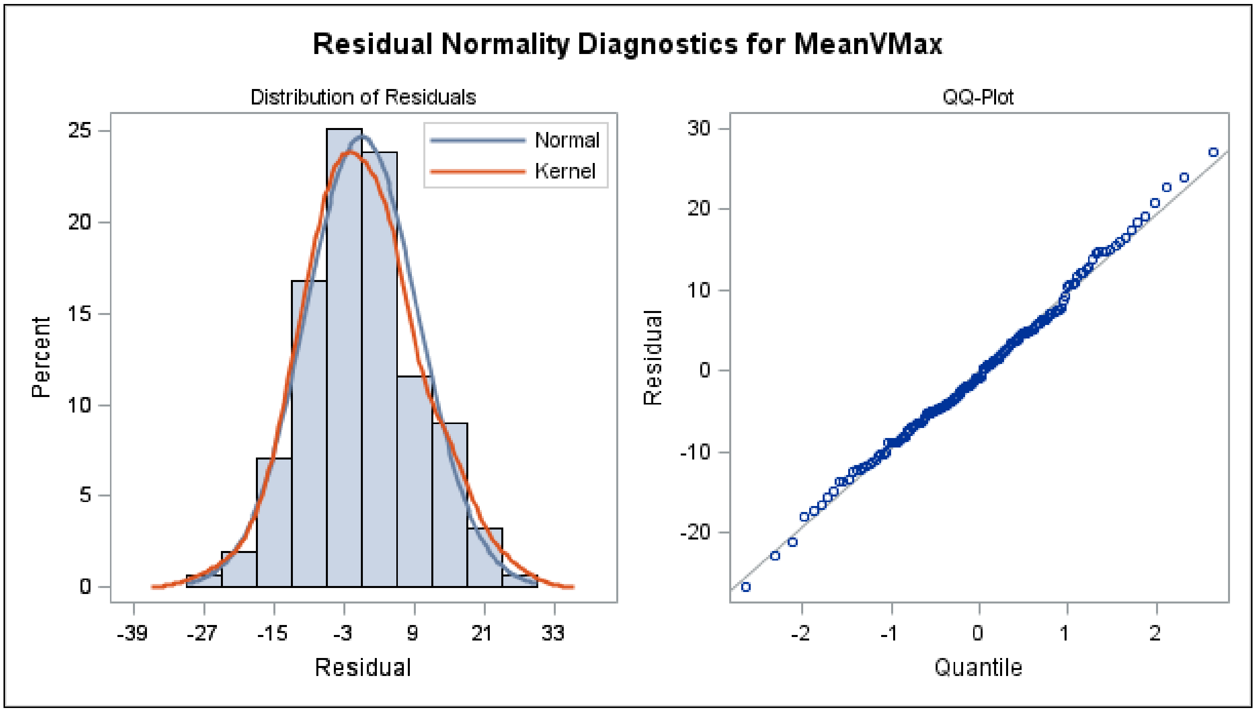
The parameter estimation and residual analysis produces the following output:

Maximum Likelihood Estimation					
Parameter	Estimate	Standard Error	t Value	Approx Pr > |t|	Lag
MU	91.26415	2.08671	43.74	<.0001	0
MA1,1	0.85466	0.08575	9.97	<.0001	1
AR1,1	0.94928	0.05190	18.29	<.0001	1

The parameter estimates are statistically significant.



The lag 3 Ljung-Box chi-square test produces a significant result, but deviations from normality do not seem to be extreme.



The table of chi-square values suggests that the model is adequate.

Autocorrelation Check of Residuals									
To Lag	Chi-Square	DF	Pr > ChiSq	Autocorrelations					
6	4.55	4	0.3363	-0.012	-0.119	0.123	0.001	0.026	0.008
12	11.09	10	0.3502	0.122	-0.154	-0.008	0.062	0.029	0.042
18	15.51	16	0.4875	-0.021	0.070	-0.051	0.020	0.062	-0.142
24	18.79	22	0.6583	-0.115	-0.068	-0.026	-0.037	0.060	-0.034
30	22.45	28	0.7599	-0.046	0.009	0.073	0.093	-0.033	-0.110

Examining the model suggested by the MINIC option is an exercise.

The ESACF criterion suggests an ARMA(2,2) model. The following code fits the model to the hurricane mean VMax series:

```
proc arima data=sasuser.hurricanes;
   identify var=meanvmax nlag=12 noprint;
   estimate p=2 q=2 ml maxiter=100;   * esacf choice *;
quit;
```

PROC ARIMA issues the following warning message:

```
Warning: The model defined by the new estimates is unstable. The iteration process has been
terminated.
```

The estimation process uses a nonlinear Gauss-Marquardt algorithm to find parameter estimates that maximize the likelihood. The algorithm detects a variety of conditions that might impact the quality of the obtained estimates. Some conditions produce error messages that imply that the model should not be used. For the ARMA(2,2) model, a warning message implies caution, but a model is produced. The model can be evaluated on its own merits, even if it is not a true maximum likelihood model.

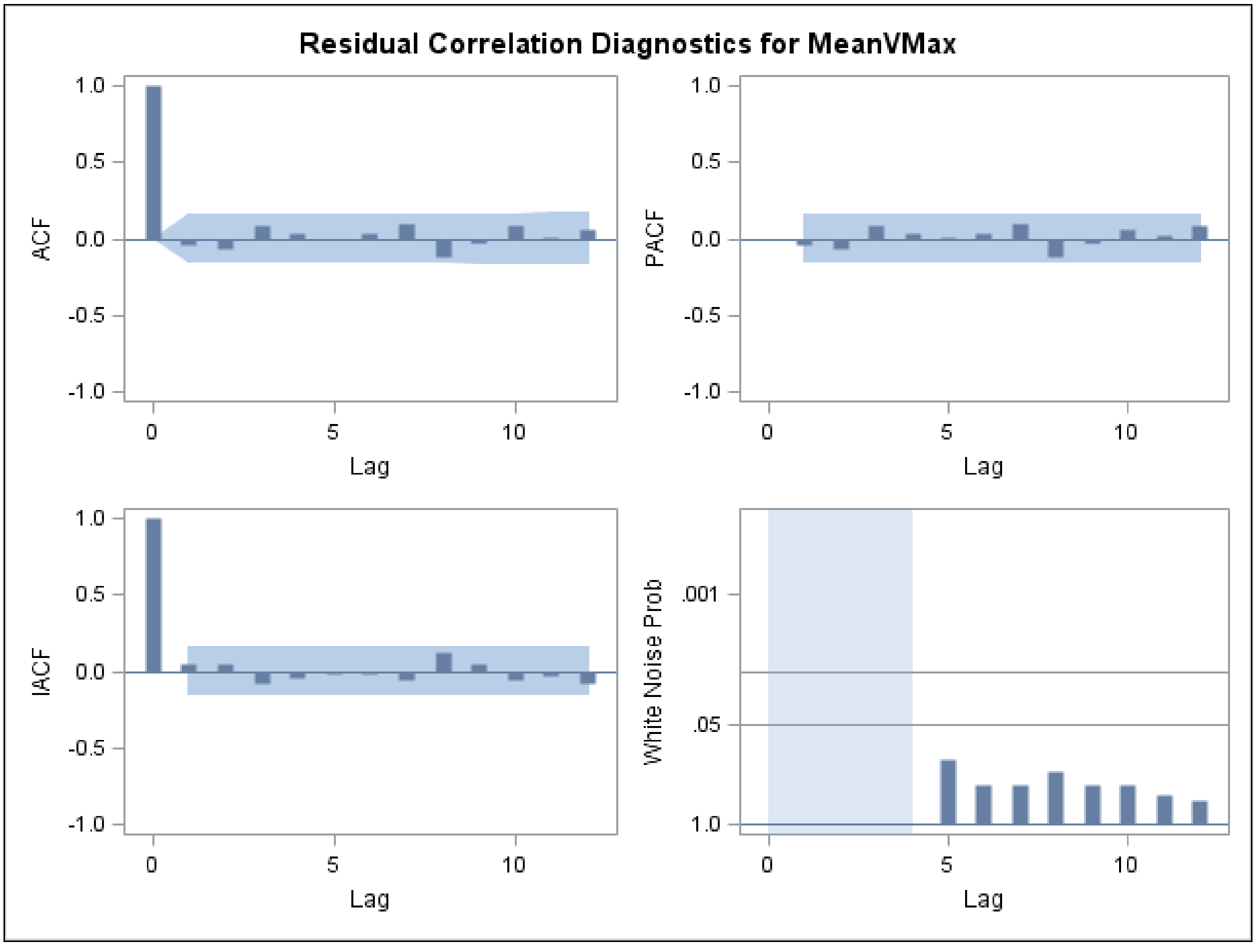
When problems are encountered during estimation, PROC ARIMA provides estimation options that might help overcome the problem. For example, the MAXITER option enables you to increase the number of allowed iterations in the search process. For unstable estimates, the MAXITER option does not help. You might need to change starting values, or you might need to change some of the parameters that control optimization. While it might be possible to achieve convergence to maximum likelihood estimates, the failure of the algorithm to converge usually means that the model is not appropriate for the data.

Despite the warning, you can examine the model that is produced. Here is the table of estimates:

Maximum Likelihood Estimation

Parameter	Estimate	Standard Error	t Value	Approx Pr > |t|	Lag
MU	91.21400	2.20080	41.45	<.0001	0
MA1,1	-0.12839	3.57387	-0.04	0.9713	1
MA1,2	0.87160	3.10909	0.28	0.7792	2
AR1,1	0.06005	0.07667	0.78	0.4335	1
AR1,2	0.85872	0.06421	13.37	<.0001	2

Even though the estimates define a predictive model, the *t* statistics and *p*-values cannot be trusted. Therefore, you cannot say that the model should not have a lag 2 MA coefficient because the *p*-value is larger than 0.1. The magnitude of the standard errors for the MA coefficients compared to the AR coefficients reveals some of the problem. It is likely that the MA estimates are too close to the AR estimates, leading to a singularity. You cannot have identical AR and MA backshift polynomials; otherwise, they “cancel out.” As you might guess, given that MA coefficients are related to unobserved error components, estimating the MA parameters can be more difficult than estimating the AR parameters.



The model produces forecasts and residuals and can be evaluated as a forecast model. The residual diagnostics imply that the model is acceptable.

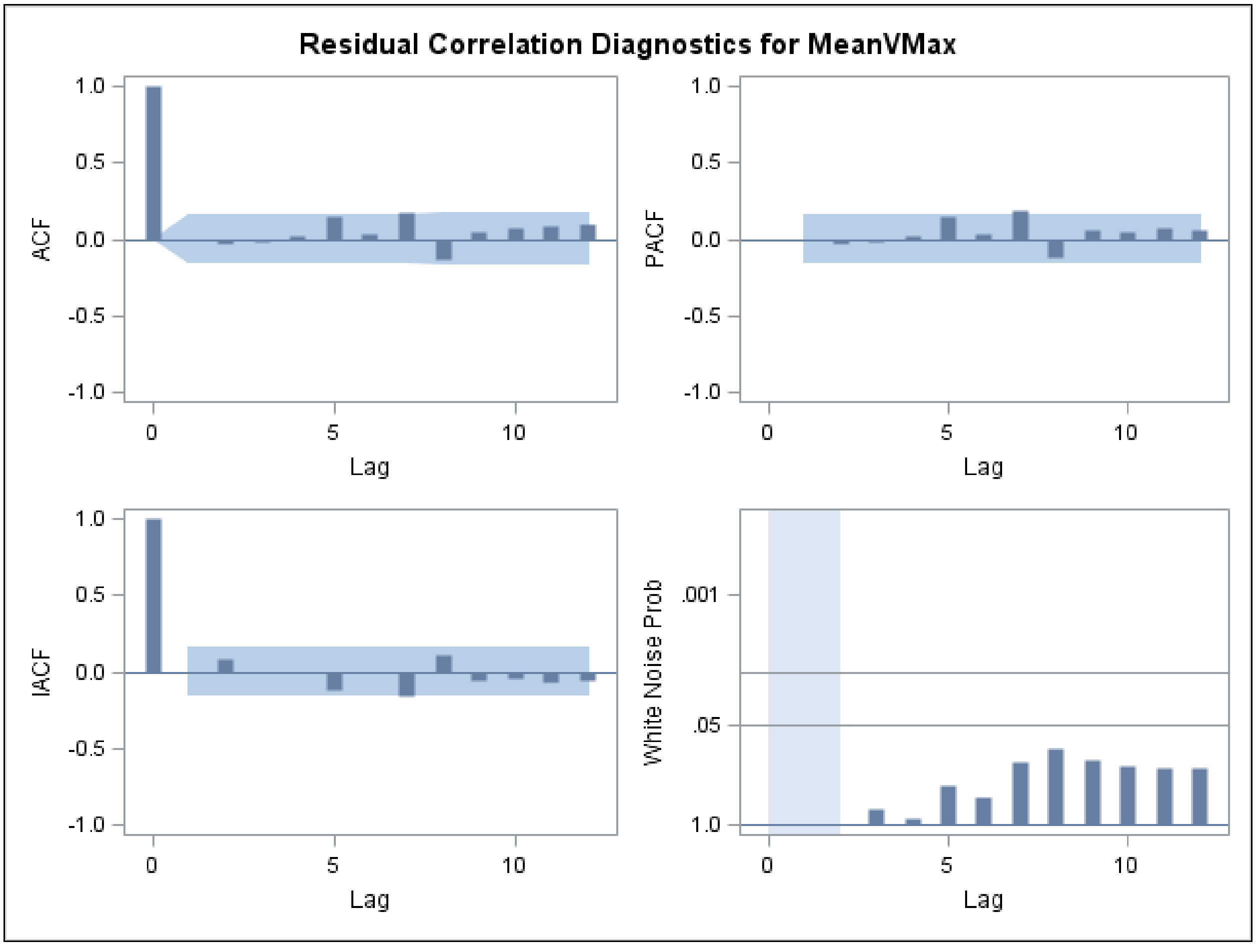
The order determining criteria are especially useful for suggesting models, but they do not restrict your analysis to these models. For example, the ARMA(1,3) model diagnosed by MINIC leads to an analysis that suggests a simpler model might be warranted. Given the orders that are proposed, you can reexamine the original autocorrelation plots for insight. The PACF and IACF plots suggest an AR(3) model, but also suggest that perhaps the coefficient for lag 2 might be zero. You should investigate thoroughly, but to simplify the analysis, proceed to an AR(3) model where the lag 2 parameter is set to zero.

```
proc arima data=sasuser.hurricanes;
   identify var=meanvmax nlag=12 noprint;
   estimate p=(1,3) ml;   * backstepping from long AR *;
quit;
```

The parameter estimates follow.

Maximum Likelihood Estimation					
Parameter	Estimate	Standard Error	t Value	Approx Pr > |t|	Lag
MU	91.27571	1.23169	74.11	<.0001	0
AR1,1	0.13733	0.07907	1.74	0.0824	1
AR1,2	0.23483	0.07925	2.96	0.0030	3

The lag 1 estimate is only marginally significant.



The residuals pass a white noise test.

For the four models that are considered (the MINIC model is left as an exercise), the following AIC values were obtained:

Model	AIC
ARMA(1,1)	1149.0
ARMA(1,3)	1148.4
AR(3) subset model	1149.4
ARMA(2,2)	1146.9

The results are very similar. While you should avoid blindly selecting a model based on a goodness-of-fit statistic, the AIC value would choose the ARMA(2,2) model.

ARMA Order Determining Methods

Limitations:

- Methods are not guaranteed to find the optimal ARMA model.
- Suggested models might be implausible.
- Methods depend on fast algorithms such as extensions of Yule-Walker estimation, but fast algorithms generally produce inferior results for small data sets.

Course macro MLMINIC implements the MINIC technique, but uses full maximum likelihood to derive the information criterion.

34

The %MLMINIC macro is in the file **ForecastUtilityMacros.sas**. The macro enables the use of AIC and SBC. Macro variables contain the top models based on AIC and SBC, and the output data set contains results for all possible models in the specified range, so you can sort by AIC or SBC to see, for example, the top 10 models.

The macro %MLMINIC is designed to be employed with sets of pairs (p,q) of AR and MA orders.

General form of the %MLMINIC macro:

%**MLMINIC**(*SAS-data-set*,*variable*,
n|*n-list*,
n|*n-list*,
OutData=*SAS-data-set*)**;**

The %MLMINIC macro is called as follows:

```
%mlminic(InputData,TargetVariable,AR_List,MA_List,OutData=DSName);
```

AR_List and **MA_List** contain a list of one or more integers separated by spaces, and both lists must have the same number of integers. If only one integer is given, the macro tries all orders from zero up to the order supplied. If values are lists of integers, then the integers are paired together to give candidate values for p and q. If the optional output data set given by the named argument OUTDATA is provided, then goodness-of-fit statistics for each model are stored in the output SAS data set. Otherwise, only the results for the selected models are written to the log file.

The following code investigates the candidate models with respect to AIC and SBC:

```
%global BestAICP BestAICQ BestSBCP BestSBCQ;
%mlminic(work.ARMA_Series05,Y,
          3 0 1 1 2 2 3 3 3,
          0 4 1 2 1 2 1 2 3,
          OutData=work.Demo2_04a);
```

Because the numeric values are lists, the macro investigates models with (p,q) pairs as follows: (3,0), (0,4), (1,1), (1,2), (2,1), (2,2), (3,1), (3,2), (and (3,3). If the macro were called with the two integers 3,3, then the (p,q) pairs up to (3,3) would be used, that is, (0,0), (0,1), (0,2), (0,3), (1,0), (1,1), (1,2), (1,3), (2,0), (2,1), (2,2), (2,3), (3,0), (3,1), (3,2), and (3,3).

A display of the output data set for the **MeanVMax** variable from the **Hurricanes** data follows:

Obs	P	Q	SBC	AIC
1	2	3	1164.70	1146.44
2	3	2	1165.15	1146.89
3	2	2	1162.13	1146.91
4	4	2	1169.38	1148.07
5	2	4	1169.38	1148.08
6	1	3	1163.66	1148.44
7	1	4	1167.16	1148.90
8	1	1	1158.17	1149.04
9	3	3	1170.42	1149.12
10	7	1	1176.88	1149.48

Several interesting models are suggested in addition to the ones proposed by MINIC, SCAN, and ESACF. The top two models are superior with respect to AIC when compared to any of the models proposed before.

2.3 Estimation and Forecasting for Stationary Time Series

Objectives

- Describe two of the estimation methods, conditional least squares and maximum likelihood, available in PROC ARIMA.
- List some of the issues related to nonlinear optimization algorithms that must be employed to obtain estimated parameters.
- Provide examples of estimation using PROC ARIMA.
- Describe the two forecast methods that are tied to the two estimation methods.
- Provide examples of identification, estimation, and forecasting of a time series.

36

Box-Jenkins Modeling Methodology

Identify	Determine ARMA orders (p,q) using ACF, PACF, and the IACF.
Estimate	Fit the ARMA(p,q) model and assess the fit of the model.
Forecast	Produce forecasts using the best ARMA model that passes assessment.

The general Box-Jenkins modeling methodology addresses trend, seasonality, and regressors as well.

37

You derived maximum likelihood estimates for stationary time series models since the first chapter. An alternative, named *conditional least squares estimation*, is also available. The two methods are discussed briefly, but the best practice is to use maximum likelihood estimation exclusively. Both methods tend to yield very similar results when a large amount of history is available to fit a model.

Estimation Methods: Conditional Least Squares

- Conditional least squares (CLS) estimators
  - are generally inferior to maximum likelihood estimators for small samples
  - are more computationally efficient than maximum likelihood estimators
  - are the default estimators used by PROC ARIMA.
- The name "conditional" least squares derives from the fact that estimation is "conditioned" on unobserved past values being exactly equal to the sample mean.
- CLS estimation is tied to infinite memory forecasting.

38

There are other estimation methods, and the study of these methods is similar to a study of the history of the development of forecasting techniques. Yule-Walker estimators were available since before World War II (Yule, 1927). As forecasting transitioned from mechanical calculators to digital computers, more sophisticated and computationally intensive methods arose. The culmination of the search for good estimators occurred when algorithms were published for maximum likelihood estimation in the 1970s. When the 1976 edition of the Box and Jenkins textbook was published, practitioners were still struggling to determine how to derive maximum likelihood estimates. The textbook came with instructions for obtaining a set of Fortran programs written by David Pack. Pack's programs implemented a crude maximum likelihood algorithm that did not produce true maximum likelihood estimates. The Pack programs included algorithms for implementing CLS estimation.

In 1980, SAS/ETS software was introduced, which included PROC ARIMA, a procedure written by David DeLong. DeLong's procedure was one of the first commercially available ARIMA programs that implemented true maximum likelihood estimation. Despite this fact, CLS was chosen as the default estimation method, because maximum likelihood estimation takes many more machine cycles than CLS. Because SAS places a high priority on upward compatibility, CLS is still the default estimation method, so programs written in 1980 can still produce the same results if they are run today with the latest version of PROC ARIMA.

Estimation Methods: Maximum Likelihood

- In general, maximum likelihood estimators are consistent, (statistically) efficient, and asymptotically normal.
- Simulation studies usually reveal that maximum likelihood estimators have superior small sample properties when compared to other estimators.
- Maximum likelihood estimation is the least efficient computationally, which is why CLS was the default method chosen for PROC ARIMA in 1980.
- Moore's law erased the computational advantage of competing methods except for extreme situations involving either many time series or unusually long time series.

39

Estimation Methods: Maximum Likelihood

- SAS software promotes upward compatibility, so METHOD=CLS remains the default setting for PROC ARIMA.
- SAS and most forecasting professionals recommend that you use METHOD=ML.
- Maximum likelihood estimation is tied to finite memory forecasting.

40

Because METHOD=ML is recommended, the syntax accepts the keyword ML to substitute for METHOD=ML in SAS®9.

Estimation Methods and Optimization Algorithms

CLS and ML algorithms are not guaranteed to find an optimal solution.

- Optimization obstacles
  - Local maxima/minima
  - Ridges (no improvement in any direction, but stopping rule not satisfied)
  - Boundary problems
  - Stability problems (scalar=dividing by zero, vector=singular matrices)
  - Others

41

continued...

Estimation Methods and Optimization Algorithms

- Constrained optimization
  - Estimates are constrained to be from stationary and invertible processes.
  - Standard errors of estimates are derived from the Hessian matrix of second-order partial derivatives. If the Jacobian vector of partial derivatives is not near zero, the Hessian matrix provides poor estimates of standard errors. This often occurs because the algorithm is stuck near the boundary of the stationary-invertible region.
  - Do ***not*** trust p-values associated with t-statistics for estimates near a boundary.

42

The last item becomes relevant when you receive a warning message from PROC ARIMA that estimates might not have converged, either due to instabilities or ridge conditions. If the warning message says that convergence did not occur after the maximum number of iterations is reached, then increase the MAXITER value.

Discussion

PROC REG will produce estimates and predictions that match most regression software and textbooks except for pathological situations. On the other hand, PROC ARIMA often produces estimates and forecasts that differ from other forecasting software products and from forecasting textbooks. Have you encountered situations like this before? What are the implications of this phenomena?

43

Forecasting Methods

- Infinite memory: Assume that the infinite past is equal to the mean of the series.
  - Used when CLS is specified
- Finite memory: Use model parameter estimates to derive a set of weights so that forecasts are a weighted average of known past values of the series.
  - Used when ML is specified

44

Woodfield and Woodward (1991) address the two forecasting methods and show why they are tied to the chosen estimation method.

Model Identification, Estimation, and Forecasting for the Groceries Data

This demonstration illustrates model identification and maximum likelihood estimation using the three time series in the **Groceries** data.

The program for this demonstration can be found in **Demo2_03Groceries.sas**. The program includes analyses for all three grocery time series, but only the grocery item **toothpaste** is described.

The data set **SASUER.Groceries** contains weekly units sold for three grocery items.

Variable	Description
Date	First day of the week defining the sales period
ToothPaste	100ml container of toothpaste, SKU suffix=010023
PeanutButter	340g jar of crunchy peanut butter, SKU suffix=020041
Biscuits	200g, 10-finger package of shortbread biscuits, SKU suffix=020153

In the United States, shortbread biscuits are referred to as shortbread cookies. Peanut butter, while common in the United States and the United Kingdom, is relatively rare in many countries. Biscuits or cookies vary in their ingredients from country to country.

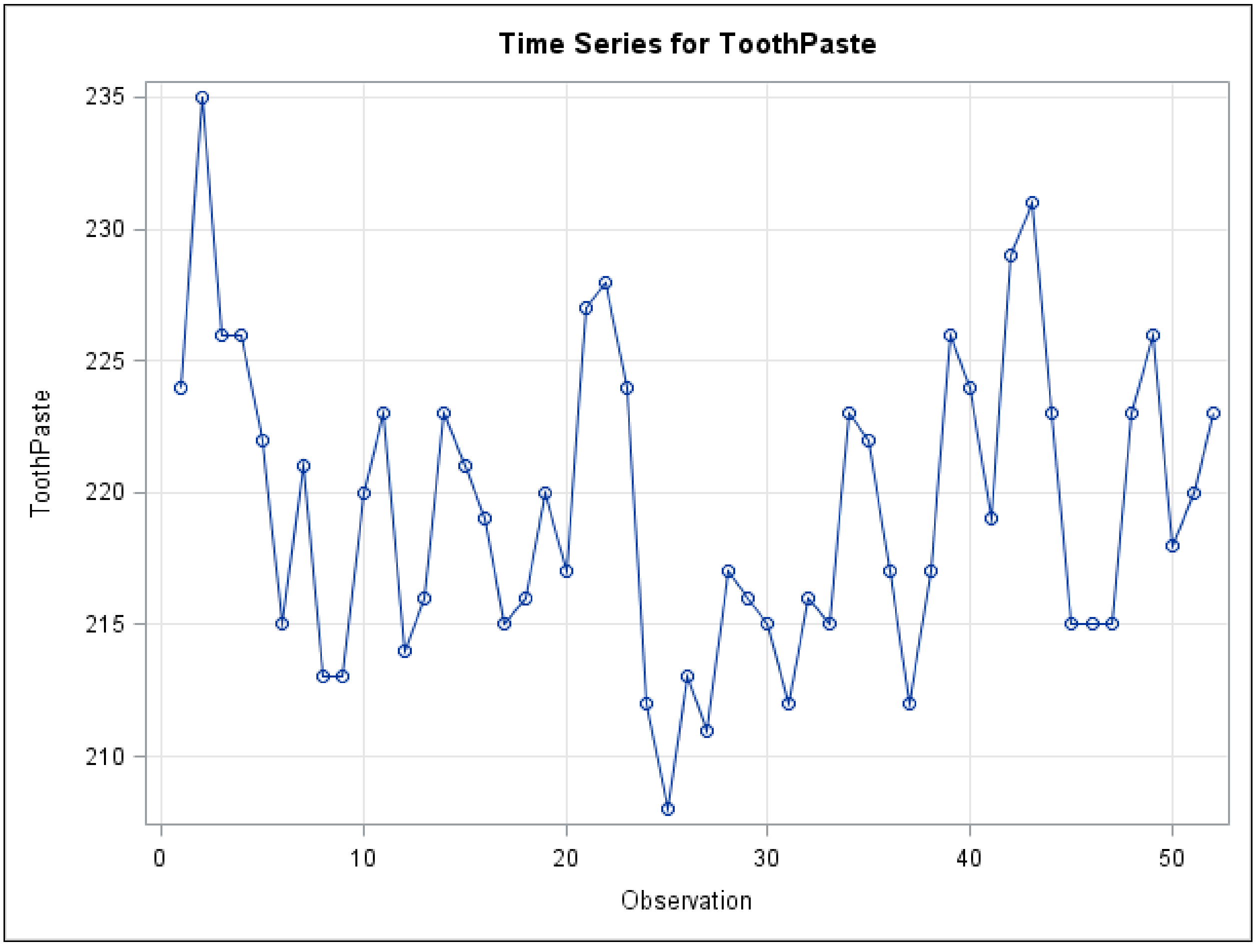
The methods for this chapter assume that the time series of interest is stationary. In general, grocery sales tend to increase until competition arises or until the retailer decides to open another store. The three items considered tend to have uniform sales throughout the year if the market is saturated, so an assumption of stationarity is valid.

The next chapter shows you how to test for stationarity. For this demonstration, visual examination of the data is sufficient.

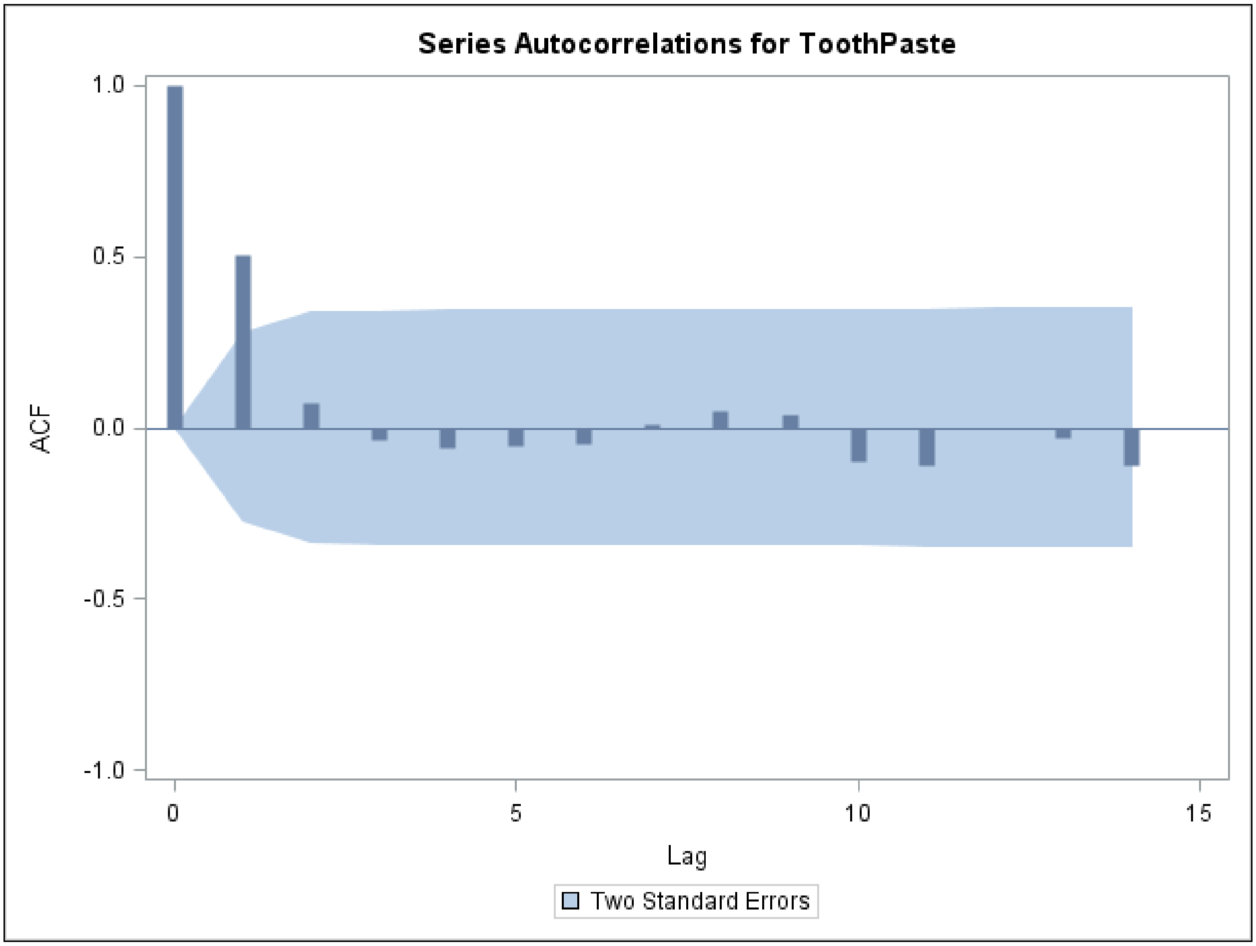
The following code generates proposed values of *p* and *q* for an ARMA(p,q) model:

```
proc arima data=sasuser.Groceries
           plots=all plots(unpack);
   identify var=ToothPaste nlags=14
            esacf minic scan
            P=(0:12) Q=(0:12)
            PERROR=(3:12);
quit;
```

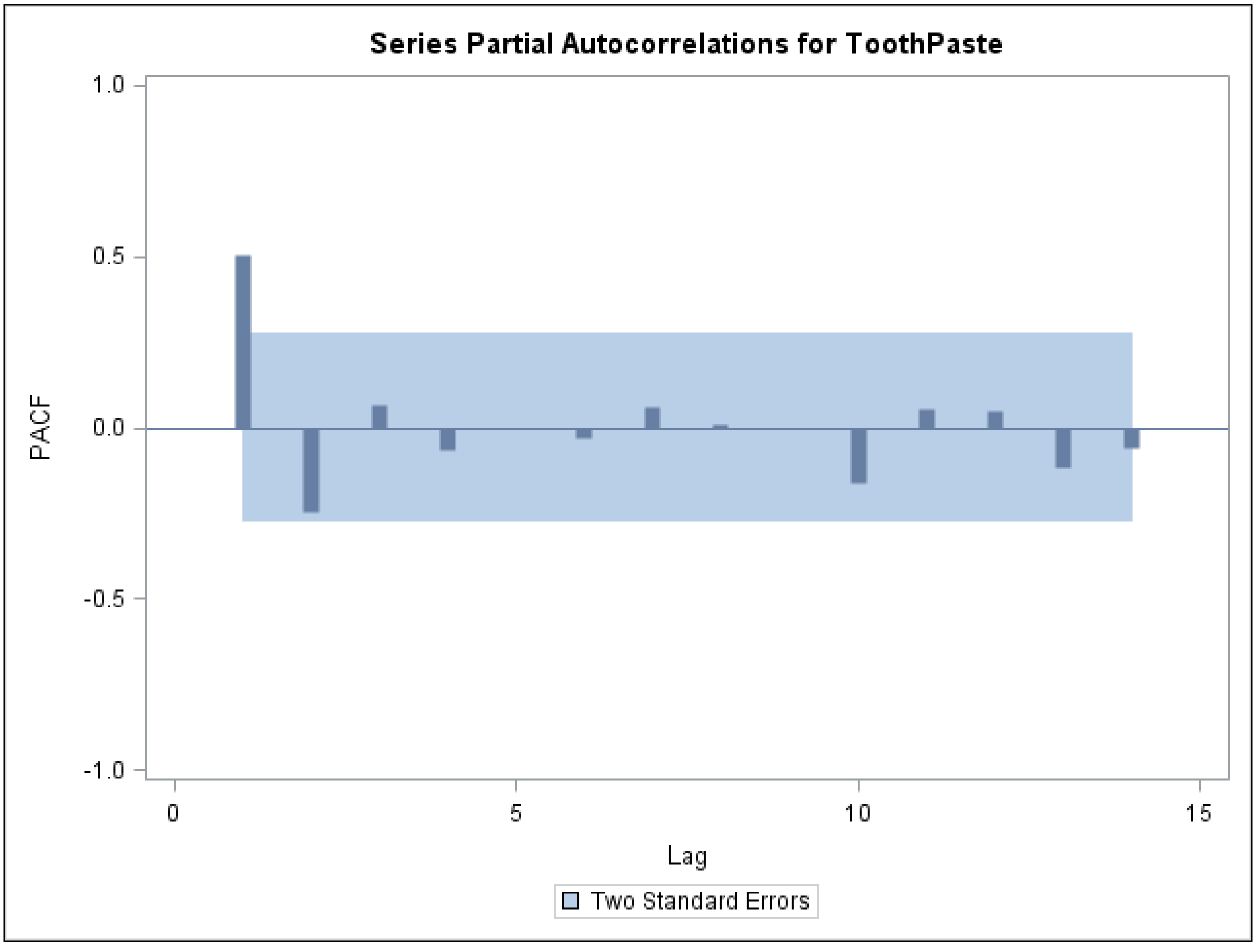
The UNPACK option produces larger plots.



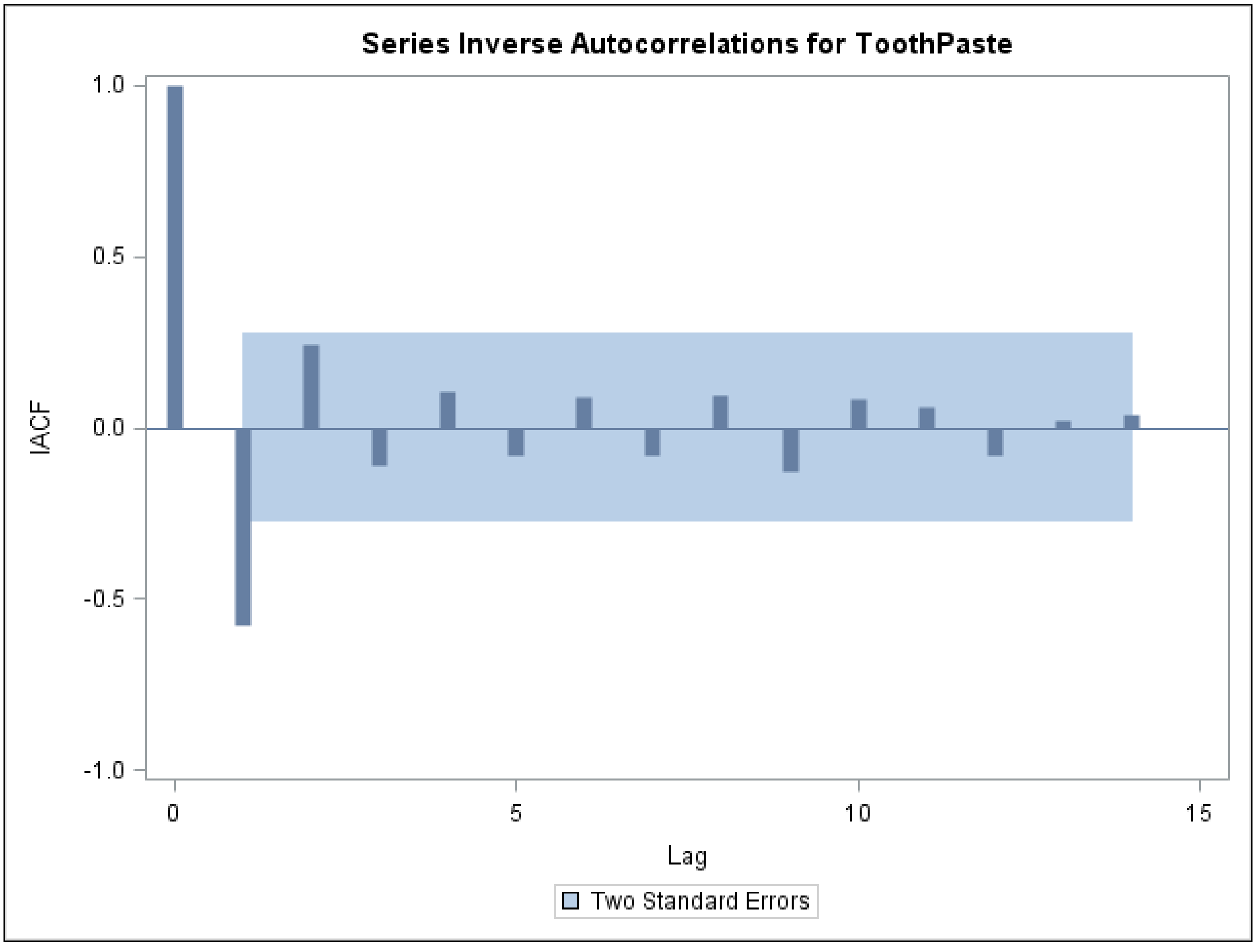
The time series plot does not exhibit any obvious trend or seasonal behavior.



The ACF has a spike at lag 1, which is consistent with an MA(1) model.



The PACF has a spike at lag 1, which is consistent with an AR(1) model.



The IACF has a spike at lag 1, which agrees with the PACF and suggests an AR(1) model. When ambiguous results suggest an AR(1) or an MA(1) model, you should also consider trying an ARMA(1,1) model.

To exploit the order-determining features of PROC ARIMA, ESACF, MINIC, and SCAN diagnostics were requested.

The results appear below:

Minimum Table Value: BIC(0,1) = 2.95113

ARMA(p+d,q) Tentative Order Selection Tests					
SCAN			ESACF		
p+d	q	BIC	p+d	q	BIC
1	0	3.055159	4	0	3.016049
0	1	2.95113	0	1	2.95113
			1	1	3.025508
			2	1	3.079199
			9	2	.
			11	3	.
			12	1	.

The MINIC procedure recommends an MA(1), which is the second choice of SCAN and ESACF. SCAN selects an AR(1), and ESACF selects an AR(4), that is, if you assume stationarity. Otherwise, for example, SCAN could be suggesting a random walk. Notice that ESACF also suggests models ARMA(1,1) and ARMA(2,1). The other suggested models are not plausible given the sample autocorrelation results.

You can use the %MLMINIC macro to perform full maximum likelihood MINIC.

```
%global BestAICP BestAICQ BestSBCP BestSBCQ;
%MLMINIC(sasuser.Groceries,ToothPaste,
          1 0 1 2,
          0 1 1 1,
          OutData=work.Toothpaste);
%AutoARMASort(work.Toothpaste,Top=4);
```

The %AutoARMASort macro displays the top four models for both AIC and SBC. That display is shown below:

Obs	P_AIC	Q_AIC	AIC	P_SBC	Q_SBC	SBC
1	0	1	313.403	0	1	317.306
2	1	1	314.942	1	0	320.208
3	1	0	316.306	1	1	320.795
4	2	1	316.878	2	1	324.683

The MA(1) model seems to be the consensus.

Similar analyses are shown for the peanut butter and the biscuits series in the demonstration program. The decision is made to use an AR(1) model for the peanut butter series and an ARMA(2,1) model for the biscuits series.

If an MA(1) model seems to be the best choice for the toothpaste time series, you should investigate how well the model fits the data. The following code fits an MA(1) model and performs a residual analysis for the toothpaste series:

```
proc arima Data=sasuser.groceries;
   identify Var=Toothpaste noprint;
   estimate q=1 ml outest=work.T_W;
   identify Var=PeanutButter noprint;
   estimate p=1 ml outest=work.P_W;
   identify Var=Biscuits noprint;
   estimate p=2 q=1 ml outest=work.B_W;
quit;
```

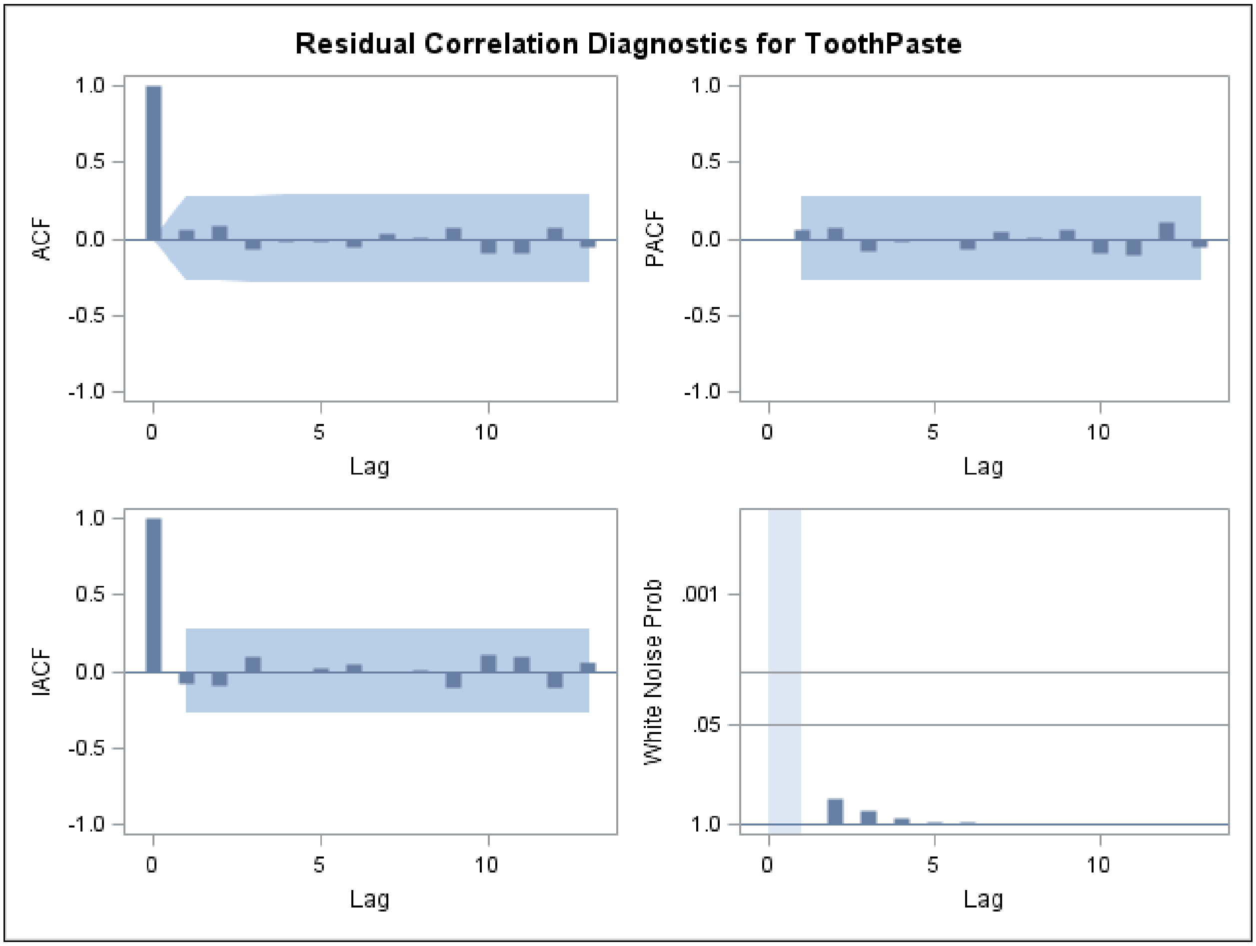
The appropriate code is also included for the other two **groceries** time series.

The estimated model and associated statistics for the **toothpaste** series follows:

Maximum Likelihood Estimation					
Parameter	Estimate	Standard Error	t Value	Approx Pr > |t|	Lag
MU	219.43124	1.05617	207.76	<.0001	0
MA1,1	-0.60003	0.11608	-5.17	<.0001	1

Constant Estimate	219.4312
Variance Estimate	23.1691
Std Error Estimate	4.813429
AIC	313.4031
SBC	317.3056
Number of Residuals	52

The estimated parameter is highly significant.



The residuals appear to be white noise.

To investigate the stability of the MA(1) model, you can exclude some of the data and see whether the parameter estimates remain stable. The following program uses the first 42 observations to refit the candidate models, dropping the last 10 weeks of data:

```
proc arima Data=sasuser.groceries(obs=42);
   identify Var=Toothpaste                 noprint;
   estimate q=1 ml outest=work.T_H         noprint;
   identify Var=PeanutButter               noprint;
   estimate p=1 ml outest=work.P_H         noprint;
   identify Var=Biscuits                   noprint;
   estimate p=2 q=1 ml outest=work.B_H     noprint;
quit;
```

You can combine the tables produced for the full sample with the tables produced by the reduced fit sample to obtain a table that is easy to examine.

```
data work.AllEsts;
   attrib Type length=$10
          Product length=$15;
   set work.T_W(in=in1) work.T_H(in=in2)
       work.P_W(in=in3) work.P_H(in=in4)
       work.b_W(in=in5) work.B_H(in=in6);
   if (_type_="EST") then do;
      if in1 or in3 or in5 then type="Full Data";
      else Type="Holdout";
      if in1 or in2 then product="Toothpaste";
      else if in3 or in4 then product="Peanut Butter";
      else Product="Biscuit";
      output;
   end;
   drop _TYPE_ _STATUS_;
run;
```

The following code prints the table for examination:

```
proc print data=work.AllEsts noobs;
run;
```

The printed table follows:

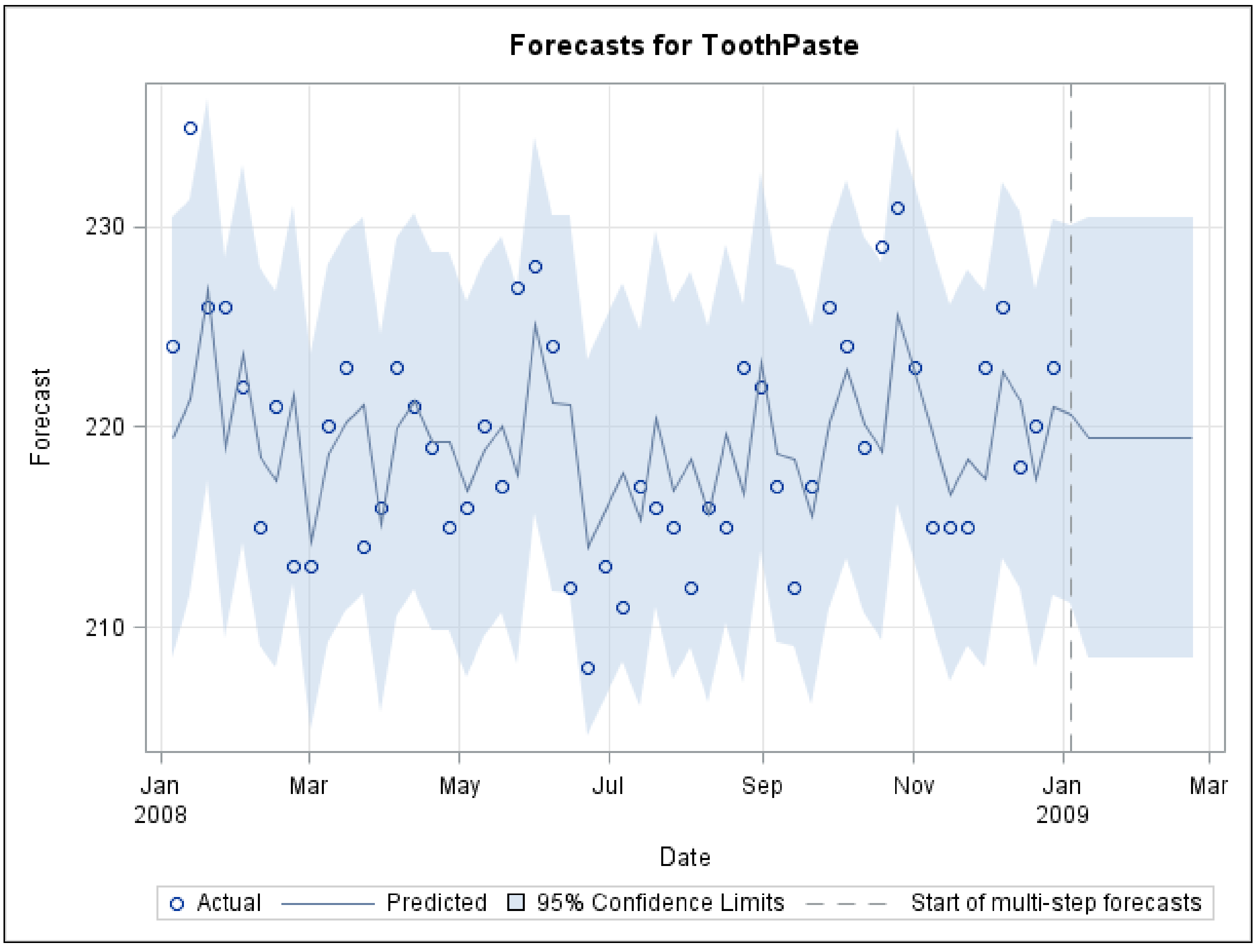
Type	Product	ERRORVAR	MU	MA1_1	AR1_1	AR1_2
Full Data	Toothpaste	23.169	219.431	-0.60003	.	.
Holdout	Toothpaste	25.679	219.200	-0.55346	.	.
Full Data	Peanut Butter	238.805	453.663	.	-0.49626	.
Holdout	Peanut Butter	271.715	453.180	.	-0.52630	.
Full Data	Biscuit	102.039	376.013	0.47363	-0.43991	-0.35883
Holdout	Biscuit	109.792	375.796	0.46969	-0.55374	-0.39682

The error variance for the **peanut butter** model suggests a possible instability, but variances calculated on data sets of different sizes should not be expected to be similar. On the other hand, the parameter estimates appear to be stable for all models.

The MA(1) model for the **toothpaste** series passes inspection. You are now ready to produce forecasts.

```
proc arima Data=sasuser.groceries plots(only)=(forecast(forecast));
   identify Var=Toothpaste noprint;
   estimate q=1 ml         noprint;
   forecast lead=8 id=date interval=week out=TPfor;

   identify Var=PeanutButter noprint;
   estimate p=1 ml           noprint;
   forecast lead=8 id=date interval=week out=PBfor;

   identify Var=Biscuits noprint;
   estimate p=2 q=1 ml   noprint;
   forecast lead=8 id=date interval=week out=Bfor;
quit;
```

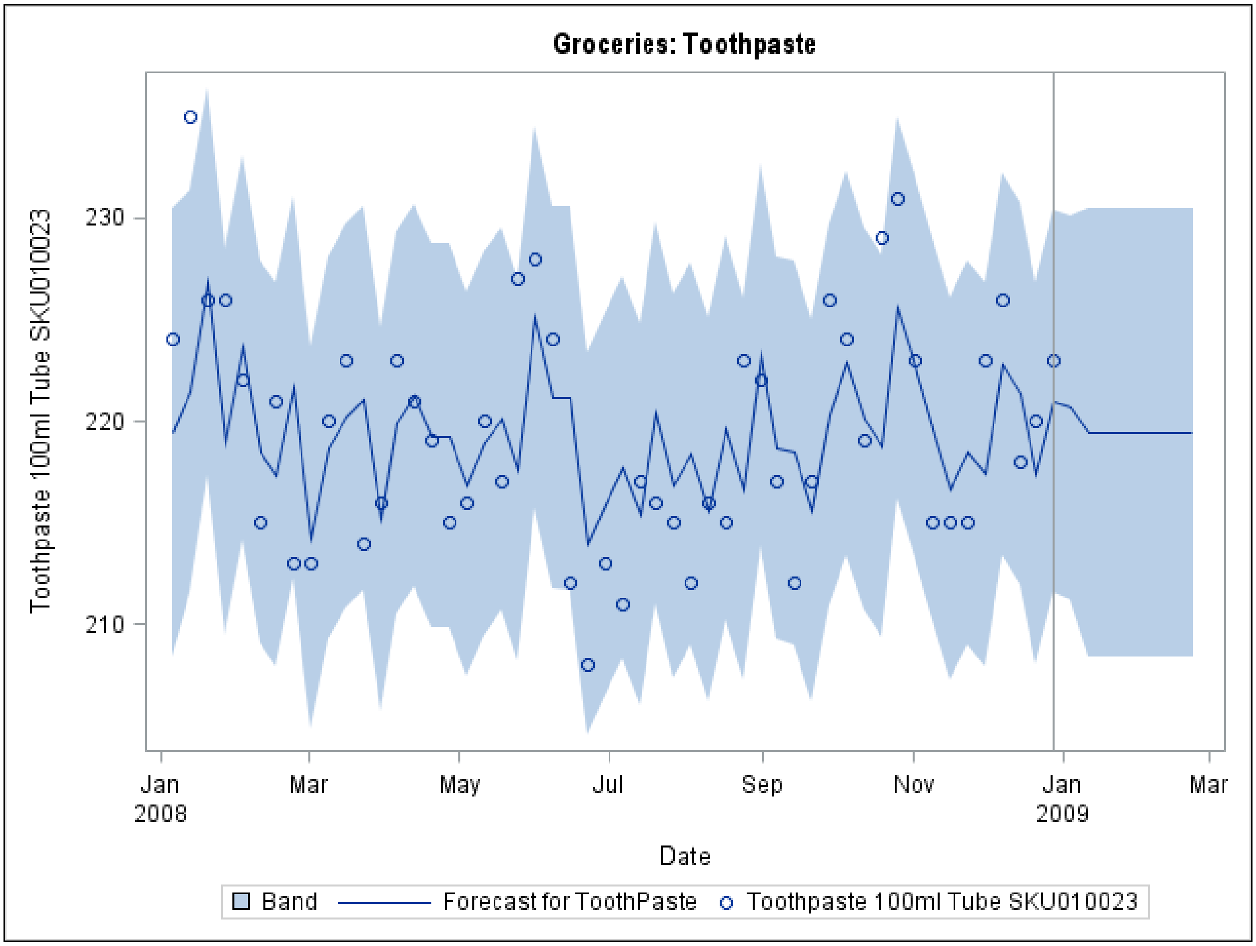
Forecasts for all three series are obtained. The forecast plot for **toothpaste** appears below:



The following code presents the forecasts using PROC SGPLOT:

```
proc sgplot data=TPfor;
   band X=date upper=U95 lower=L95;
   series X=date Y=Forecast;
   scatter X=date Y=Toothpaste;
   refline "28Dec2008"d /axis=X;
run;
```

The plot appears below:



The forecasts converge to the mean after one week.

Using a Holdout Sample

This demonstration illustrates model identification using a holdout sample.

The code for this demonstration can be found in **Demo2_04Holdout.sas**. The usual reason for using a holdout sample is to get an accurate assessment of how well a forecast model performs when deployed in the operational environment. You decide that you can accommodate a holdout sample, or you decide against a holdout sample, possibly because of a lack of data, instability, or a change or event that occurred during the holdout period. If you discover instability during the analysis, you can revert to a full sample approach.

The code uses the %AutoARMAHoldout macro to obtain MAPE and RMSE for a specified holdout period. The syntax of %AutoARMAHoldout is similar to that of %MLMINIC. However, it does not implement an established methodology other than looking at a large set of models and choosing winners based on MAPE and RMSE. The %AutoARMAHoldout macros is in the **ForecastUtilityMacros.sas** file.

```
title1 f=&coursefont "Best Models for Holdout of 8";
title2 f=&coursefont "Toothpaste";
%AutoARMAHoldout(sasuser.Groceries,ToothPaste,8,
                 1 0 1 2,
                 0 1 1 1,
                 Outdata=work.Toothpaste_H);
title2 f=&coursefont "Peanut Butter";
%AutoARMAHoldout(sasuser.Groceries,PeanutButter,8,3,4,
                 Outdata=work.PntButter_H);
title2 f=&coursefont "Biscuits";
%AutoARMAHoldout(sasuser.Groceries,Biscuits,8,
                 1 0 1 2 2 4,
                 0 1 1 0 1 0,
                 Outdata=work.Biscuits_H);
```

The following code combines the three data sets produced above:

```
data next;
   attrib item length=$15;
   set work.Toothpaste_H(in=in1)
       work.PntButter_H(in=in2)
       work.Biscuits_H(in=in3);

   if in1 then item="Toothpaste";
   else if in2 then item = "Peanut Butter";
   else if in3 then item = "Biscuits";
   else item = "Unknown";
   model="ARMA("||strip(p)||","||strip(q)||")";
   drop p q;
run;
```

The following code completes the creation of the combined table and prints the results:

```
/*----  Sort on MAPE and rank  ----*/
proc sort data=next;
   by item;
run;

proc rank data=next out=MAPE;
   by item;
   var MAPE;
   ranks rank;
run;
proc datasets library=work nolist;
   modify MAPE;
   rename Model=MAPE_Model;
quit;
/*----  Rank RMSE  ----*/
proc rank data=next out=RMSE;
   by item;
   var RMSE;
   ranks rank;
run;
proc datasets library=work nolist;
   modify RMSE;
   rename Model=RMSE_Model;
quit;
proc sort data=MAPE;
   by item rank;
run;
proc sort data=RMSE;
   by item rank;
run;
/*----  Merge by ranks and print  ----*/
data both;
   merge MAPE RMSE;
   by item rank;
   match = " ";
   if RMSE_model=MAPE_model then match = "*";
run;
title2 "Model Selections";
proc print data=both;
   where rank<5;
run;
```

The results appear below:

Obs	item	MAPE	RMSE	MAPE_Model	rank	RMSE_Model	match
1	Biscuits	1.8957	7.8789	ARMA(2,1)	1	ARMA(2,1)	*
2	Biscuits	1.9322	7.9207	ARMA(2,0)	2	ARMA(4,0)	
3	Biscuits	1.9481	8.4385	ARMA(4,0)	3	ARMA(1,1)	
4	Biscuits	1.9043	8.7088	ARMA(1,1)	4	ARMA(2,0)	
7	Peanut Butter	1.7955	10.5161	ARMA(0,1)	1	ARMA(0,1)	*
8	Peanut Butter	1.9660	10.5243	ARMA(2,0)	2	ARMA(2,0)	*
9	Peanut Butter	1.9675	10.5365	ARMA(1,1)	3	ARMA(1,1)	*
10	Peanut Butter	1.9689	10.5476	ARMA(1,0)	4	ARMA(1,0)	*
27	Toothpaste	1.5118	3.5322	ARMA(0,1)	1	ARMA(0,1)	*
28	Toothpaste	1.5626	3.6833	ARMA(1,1)	2	ARMA(1,1)	*
29	Toothpaste	1.5832	3.7760	ARMA(2,1)	3	ARMA(2,1)	*
30	Toothpaste	1.7417	4.2207	ARMA(1,0)	4	ARMA(1,0)	*

The statistics MAPE and RMSE choose the same model and ranking for nine of the 12 models. The model selected with rank 1 for each series is the same as was selected using the full sample.

The stability check for each model is included in the program, but it is identical to the one performed in the previous demonstrations.

Because the same model was selected for each series, and because you use the full data set to obtain final forecasts, the forecast results are the same as in the previous demonstration.

Discussion

Do you have to implement a forecast equation outside of SAS/ETS, for example, to permit the calculation of forecasts using a spreadsheet or other software product? If so, what are the challenges you have faced?

47

Exercises

1. Fit an ARMA(1,3) model to the **Hurricanes** data and evaluate the results. This is a continuation of the demonstration in Section 2.2, where the ARMA(1,3) model was suggested by the MINIC criterion.

2. Complete the demonstration for program **Demo2_03Groceries.sas**.

2.4 Chapter Summary

Box-Jenkins modeling methodology provides a framework for deriving good forecast models. The Box-Jenkins ARMA models for stationary time series are backed by a powerful theory that guarantees the existence of a good approximating model for any stationary series. The practical problem is how to identify the orders of an ARMA model that best approximates the data.

Diagnostic functions provide strategies for determining orders *p* and *q* for an ARMA(*p,q*) model. The autocorrelation function (ACF), partial autocorrelation function (PACF), and inverse autocorrelation function (IACF) exhibit certain properties for various models. Experienced forecasters learn how to use these functions to identify ARMA models. For more difficult diagnostic situations, the ESACF, MINIC, and SCAN methods can help to determine model orders. Additionally, a well-defined bounded trial-and-error process applied to a reduced set of models often leads to a good approximating ARMA model.

After a model is identified, the parameters are estimated, and forecast equations are derived to predict future values of the series. PROC ARIMA supports the requirements of Box-Jenkins time series methodology, although other procedures can be used to fit Box-Jenkins models.

SAS/ETS provides two simple alternatives to Box-Jenkins models: stepwise autoregression and exponential smoothing. PROC FORECAST implements stepwise autoregression methodology, and PROC ESM implements exponential smoothing methodology.

For Additional Information

Box, G.E.P., and Gwilym M. Jenkins. 1976. *Time Series Analysis: Forecasting and Control*. Oakland, California: Holden-Day.

Box, George, Gwilym M. Jenkins, and Gregory C. Reinsel. 1994. *Time Series Analysis: Forecasting and Control*, *Third Edition*. Prentice-Hall.

Hamilton, James D. 1994. *Time Series Analysis*. Princeton, New Jersey: Princeton University Press.

Peña, Daniel, George C. Tiao, and Ruey Tsay (eds.). 2001. *A Course in Time Series Analysis*. New York: John Wiley and Sons.

Tsay, R. S. and Tiao, G. C. 1984. "Consistent Estimates of Autoregressive Parameters and Extended Sample Autocorrelation Function for Stationary and Nonstationary ARMA Models." *JASA*, 79 (385), 84–96.

Tsay, R. S. and Tiao, G. C. 1985. "Use of Canonical Analysis in Time Series Model Identification." *Biometrika*. 72 (2), 299-315.

Walker, G. 1931. "On periodicity in series of related terms." *Proceedings of the Royal Society*. A131, 518.

Woodfield, Terry J. 1988. "Simulating Stationary Gaussian ARMA Time Series." *Proceedings of Computer Science and Statistics: 20th Symposium on the Interface*, pp 612-617.

Woodfield, Terry J., and Donna Woodward. 1992. "Forecasting with PROC ARIMA." *Observations: The Technical Journal for SAS Software Users*. Vol 1, No 3, 38-46.

Yule, G.U. 1927. "On a Method of investigating periodicities in a disturbed series, with a special reference to Wölfer's sunspot numbers." *Philosophical Transactions*. A226, 267.

2.5 Solutions

Solutions to Exercises

1. Fit an ARMA(1,3) model to the **Hurricanes** data and evaluate the results.

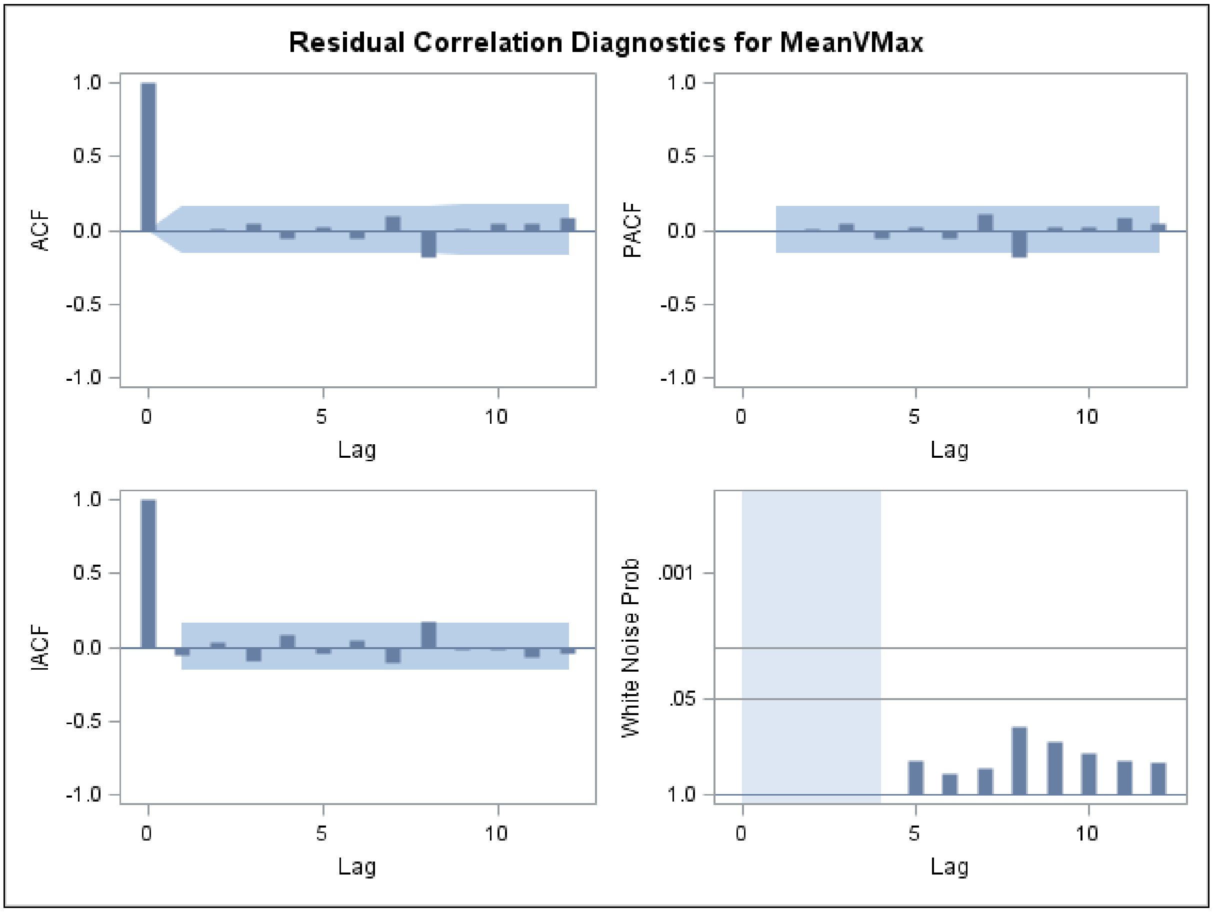
The following program fits the model and produces diagnostics for the residuals:

```
proc arima data=sasuser.hurricanes;
   identify var=meanvmax nlag=12 noprint;
   estimate p=1 q=3 ml ;  * minic model  *;
quit;
```

The parameter estimates follow:

Maximum Likelihood Estimation					
Parameter	Estimate	Standard Error	t Value	Approx Pr > |t|	Lag
MU	91.37596	1.77505	51.48	<.0001	0
MA1,1	0.77342	0.11511	6.72	<.0001	1
MA1,2	0.14567	0.10406	1.40	0.1616	2
MA1,3	-0.20736	0.08689	-2.39	0.0170	3
AR1,1	0.87834	0.08779	10.01	<.0001	1

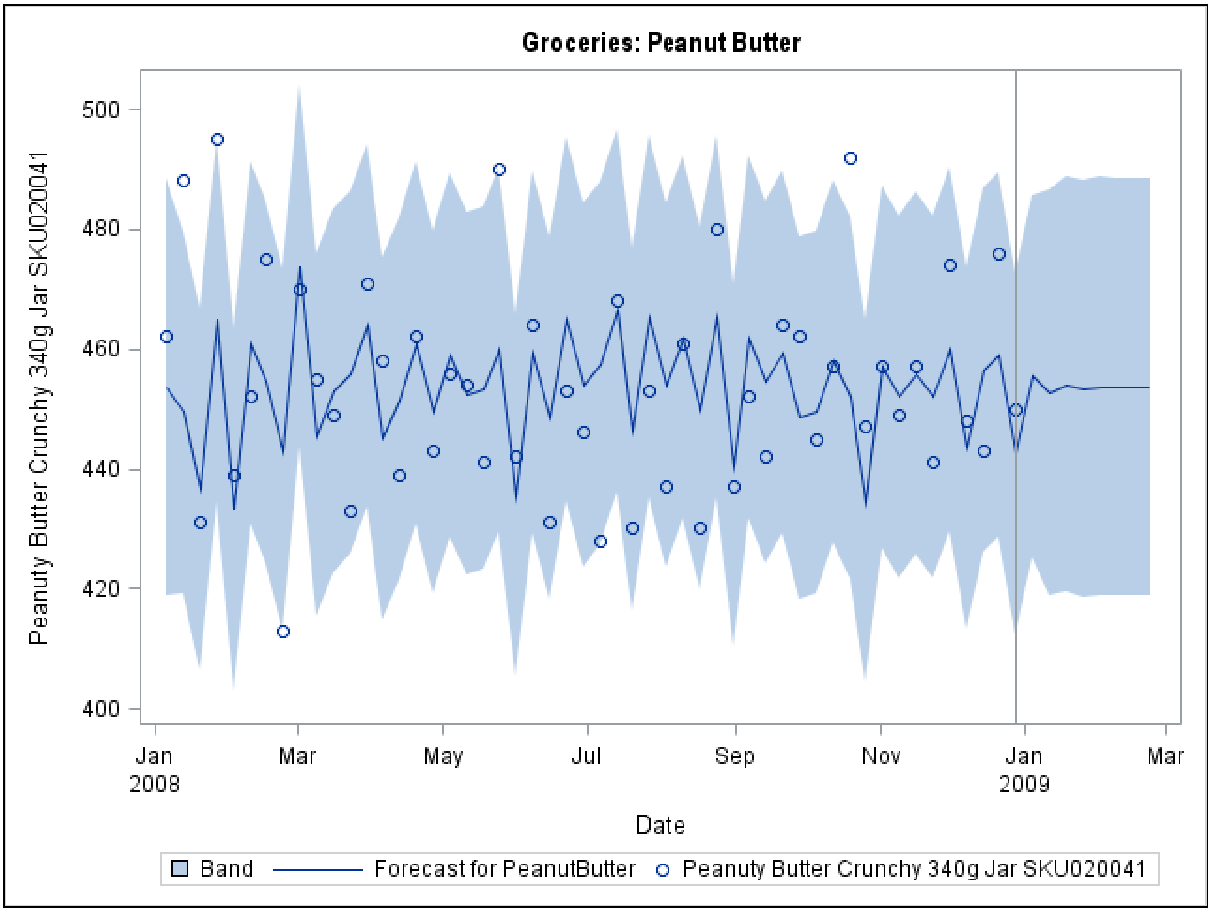
The MA1,2 coefficient (the coefficient corresponding to lag 2) is not significant at the 5% level. While this is not sufficient to disqualify a model for forecasting, it suggests that a subset model might be preferred.

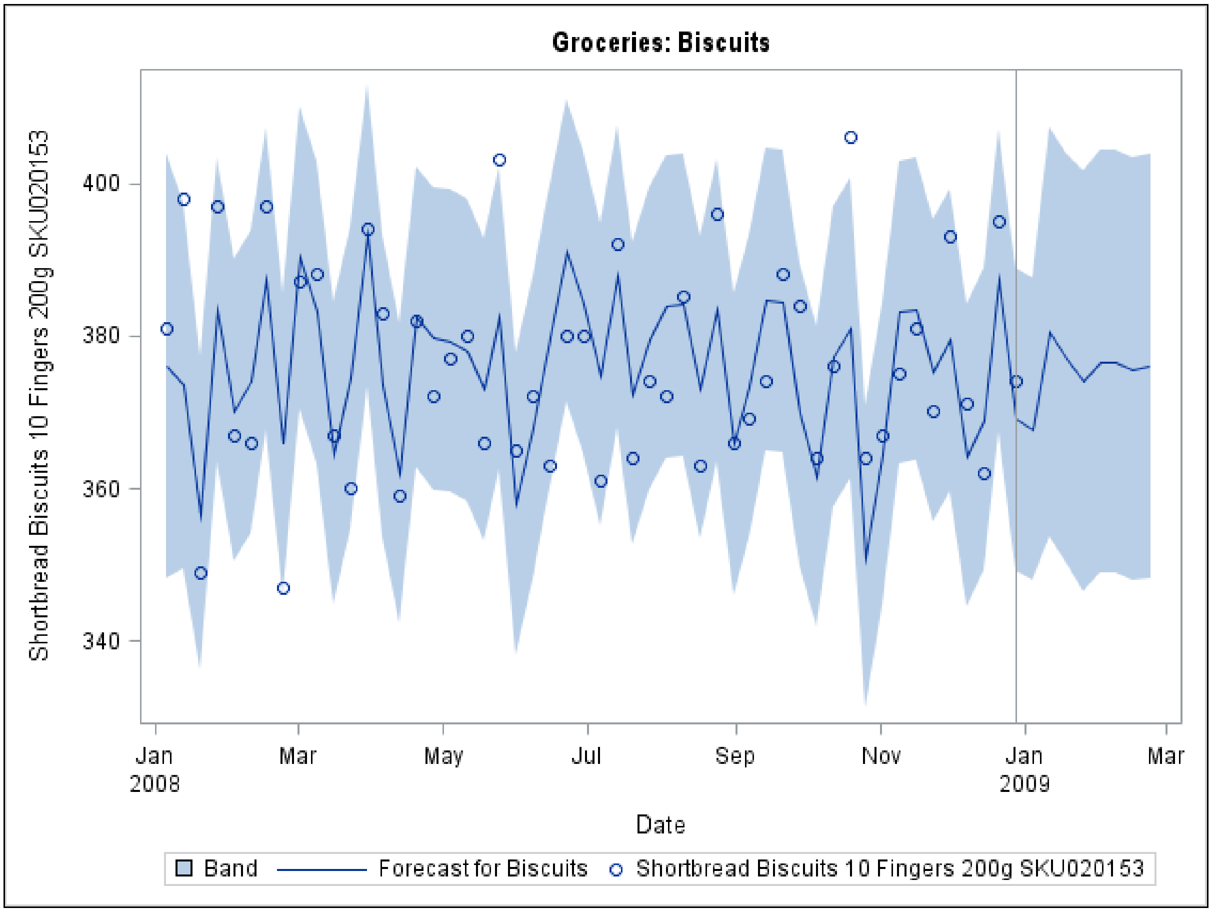


The residual resembles white noise. The model does not exhibit any obvious lack of fit problems.

2. Complete the demonstration for program **Demo2_03Groceries.sas**.

The demonstration analyzed the **toothpaste** series. Run the portions of the code for the **peanut butter** series and for the **biscuits** series. The plots appear below:




Groceries: Biscuits
Shortbread Biscuits 10 Fingers 200g SKU020153
400
380
360
340
Jan 2008
Mar
May
Jul
Sep
Nov
Jan 2009
Mar
Date
Band
Forecast for Biscuits
Shortbread Biscuits 10 Fingers 200g SKU020153

Solutions to Student Activities (Polls/Quizzes)

2.01 Multiple Choice Poll – Correct Answer

Which of the following statements is true?

a. Time series forecasting methods require that the time series of interest be stationary.
b. A trigonometric sine wave is stationary because it cycles about a constant value.
c. Forecasts for stationary time series always converge to the mean of the series.
d. Long-term forecasts are more accurate for stationary time series than for nonstationary series.
e. Politicians want to help you.

8

Chapter 3 Trend Models

3.1 Introduction to Nonstationary Time Series

Objectives

- Describe models with linear trend and autocorrelated errors.
- Explain how PROC ARIMA and other SAS procedures can model time series that have trend.
- Demonstrate trend models with an example.

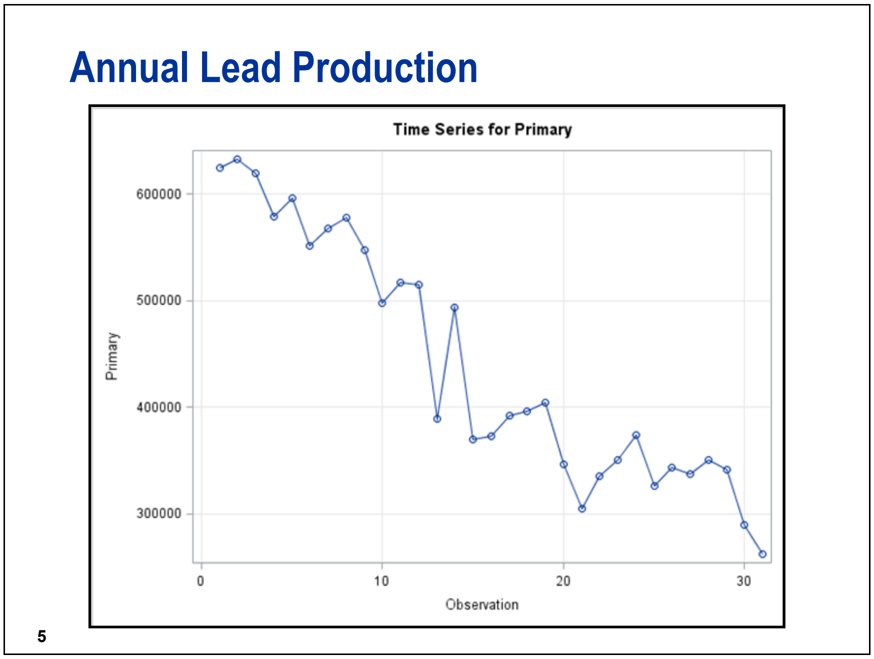
3

SAS/ETS Procedures that Can Model Trend

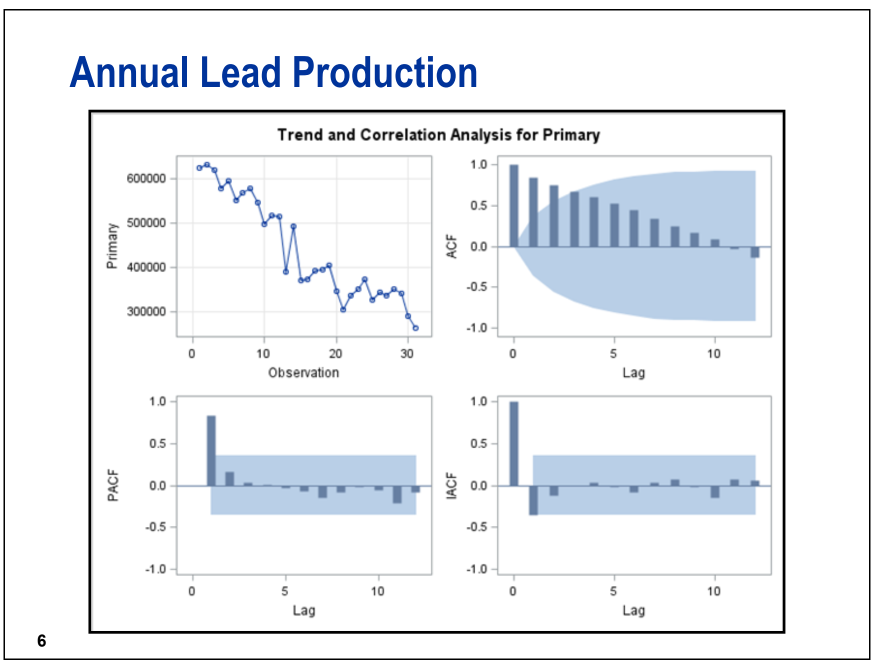
- PROC FORECAST and PROC ESM
  - Linear and quadratic trend only
- PROC AUTOREG
  - Models trend as a deterministic input
  - Accommodates AR models to deal with autocorrelation
  - Can eliminate insignificant lags with the BACKSTEP option
- PROC ARIMA
  - Can model trend as a deterministic input
  - Can model stochastic trend using differencing
  - Accommodates ARMA models to deal with autocorrelation

4

The previous chapter addressed stationary time series. The description of the above procedures in SAS/ETS implies that the supported models can accommodate a stationary error component that exhibits autocorrelation. Consequently, many of the models that you derive for forecasting use ARMA to model the residuals that remain after removing trend. Models that have trend are used to forecast nonstationary time series. The presence of trend implies that the mean of the time series depends on time through the trend function.



Two versions of lead production data are used in this course. The above plot represents primary lead production accumulated annually. Later you encounter the monthly lead production data. Primary lead production refines lead ore obtained from mining. Secondary lead production obtains lead through recycling, such as recovering lead from car batteries.



If you ignore the plot of the series (which is unwise!) and focus on the autocorrelation plots, you might conclude that annual lead production can be approximated by an AR(1) model, because of the following:

- There is a single spike in the PACF and IACF at lag 1.
- The ACF exhibits exponential decay.

Annual Lead Production—AR(1) Model

```
proc arima data=work.LeadYear
           plots=all;
   identify var=Primary;
   estimate p=1 method=ml plot;
   forecast lead=5 id=Date interval=year
            out=work.OutAR1;
quit;
```

7

Using what you learned in the last chapter, you can easily fit an AR(1) model to the primary lead production time series.

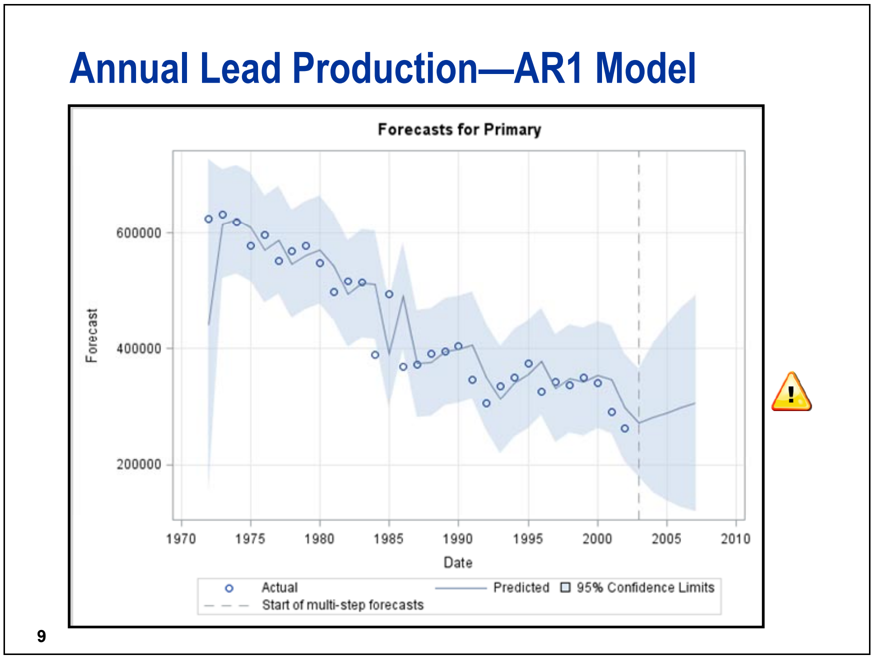
Annual Lead Production—AR1 Model

Maximum Likelihood Estimation					
Parameter	Estimate	Standard Error	t Value	Approx Pr > |t|	Lag
MU	441535.2	121278.7	3.64	0.0003	0
AR1,1	0.94588	0.06579	14.38	<.0001	1

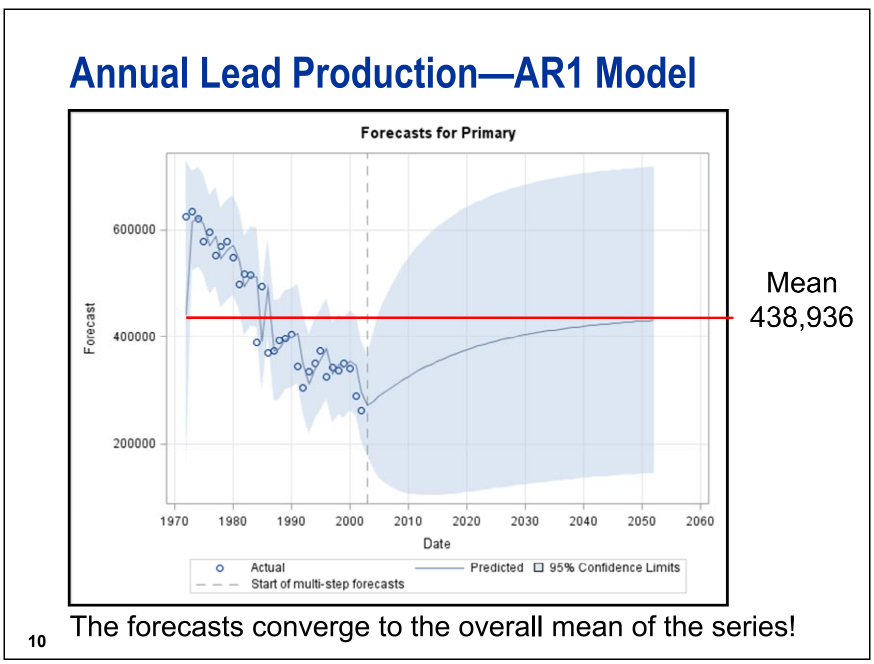
Constant Estimate	23893.85
Variance Estimate	2.2434E9
Std Error Estimate	47364.13
AIC	759.6262
SBC	762.4941
Number of Residuals	31

8

The estimated parameter is statistically significantly different from zero. The estimate is also close to the stationary boundary of 1.



One of the first indications of a problem with the AR(1) model is a forecast plot showing future values changing direction from the historic downward trend. The AR(1) model optimistically projects a change point at the very last observed time point, and propels the forecasts in a positive direction.



Extending the forecasts 50 years reminds you that a stationary model must revert to the mean. The mean reversion causes the shift to a positive trend, not any prescience emanating from the model.

This example provides the transition from the world of stationary time series to that of nonstationary time series. The methodologies presented in the last chapter are still relevant, but the model identification is tied to candidate trend components used to model the nonstationary behavior of the series.

Extensions of AR(1) Models to Model Trend

Linear Trend plus AR(1)

$$Y_t = \beta_0 + \beta_1 t + Z_t, Z_t = \phi_1 Z_{t-1} + \varepsilon_t$$

can be written as the following:

$$Y_t - \beta_0 - \beta_1 t = \phi_1 (Y_{t-1} - \beta_0 - \beta_1 (t-1)) + \varepsilon_t$$

and reduces to a random walk with drift.

$$Y_t = \beta_1 + Y_{t-1} + \varepsilon_t, \text{ when } \phi_1 = 1$$

11

The linear trend plus AR(1) model with $|\phi_1|<1$ has forecasts that revert to the line $Y_t = \beta_0 + \beta_1 t$ (where, if β_1=0, the line is only the mean β_0). The variance of the forecast error converges to a finite constant as the lead period L increases. On the other hand, if $\phi_1 = 1$, the process is a random walk with drift β_1 in which the lead one forecast is the last observation plus β_1 and each subsequent forecast adds another β_1 so that the lead L forecast is the last observation plus Lβ_1. While this describes a linear forecast, it is not the line given by a linear regression and the forecast error variance grows linearly without bound, much faster than that for a trend plus stationary AR(1) model, so the confidence intervals become quite wide as the lead period increases. Is one of these models (AR(1), linear trend plus AR(1) or RWD = random walk with drift) appropriate for the annual lead data?

Trend Models for Primary Lead Production

This demonstration illustrates how to prepare data and fit trend models to the primary lead production time series.

The program for this demonstration can be found in **Demo3_01Lead.sas**.

The following program code produces a plot of the time series and the sample autocorrelation functions for primary lead production. The plots were shown previously.

```
proc arima data=sasuser.LeadYear plots(only)=(series(corr))
           plots(unpack);
   identify var=Primary nlag=12;
quit;
```

The following code fits and AR(1) model and forecasts 20 years into the future. The forecast plot shown above uses **lead**=50:

```
proc arima data=YearlyLead plots=all;
   identify var=Primary nlag=12;
   estimate p=1 ml;
   forecast lead=20 id=Date interval=year
            out=work.outAR1;
quit;
```

When adding deterministic trend input variables to a model, you need to extrapolate the deterministic values into the future so that PROC ARIMA can use them for forecasting. Otherwise, PROC ARIMA fails to generate a forecast with a warning message that values of input variables are not available.

The following code adds the time index and increments the date variable for 20 years into the future:

```
data work.YearlyLead;
   set sasuser.LeadYear end=eof;
   Time+1;
   output;
   if (eof) then do future=1 to 20;
      Primary=.;
      Secondary=.;
      Total=.;
      Time+1;
      Date=intnx("year",Date,1);
      output;
   end;
   drop future;
run;
```

A portion of the data is shown below.

VIEWTABLE: Work.Yearlylead

	Year	Primary U.S. Lead Production Metric Tons	Secondary U.S. Lead Prod. Metric Tons	Total U.S. Lead Production Metric Tons	Time
30	2001	290,000	1,040,000	1,330,000	30
31	2002	262,000	1,070,000	1,332,000	31
32	2003	.	.	.	32
33	2004	.	.	.	33
34	2005	.	.	.	34
35	2006	.	.	.	35
36	2007	.	.	.	36
37	2008	.	.	.	37
38	2009	.	.	.	38
39	2010	.	.	.	39
40	2011	.	.	.	40
41	2012	.	.	.	41
42	2013	.	.	.	42
43	2014	.	.	.	43
44	2015	.	.	.	44
45	2016	.	.	.	45
46	2017	.	.	.	46
47	2018	.	.	.	47
48	2019	.	.	.	48
49	2020	.	.	.	49
50	2021	.	.	.	50
51	2022	.	.	.	51

The following code fits a linear trend to the data:

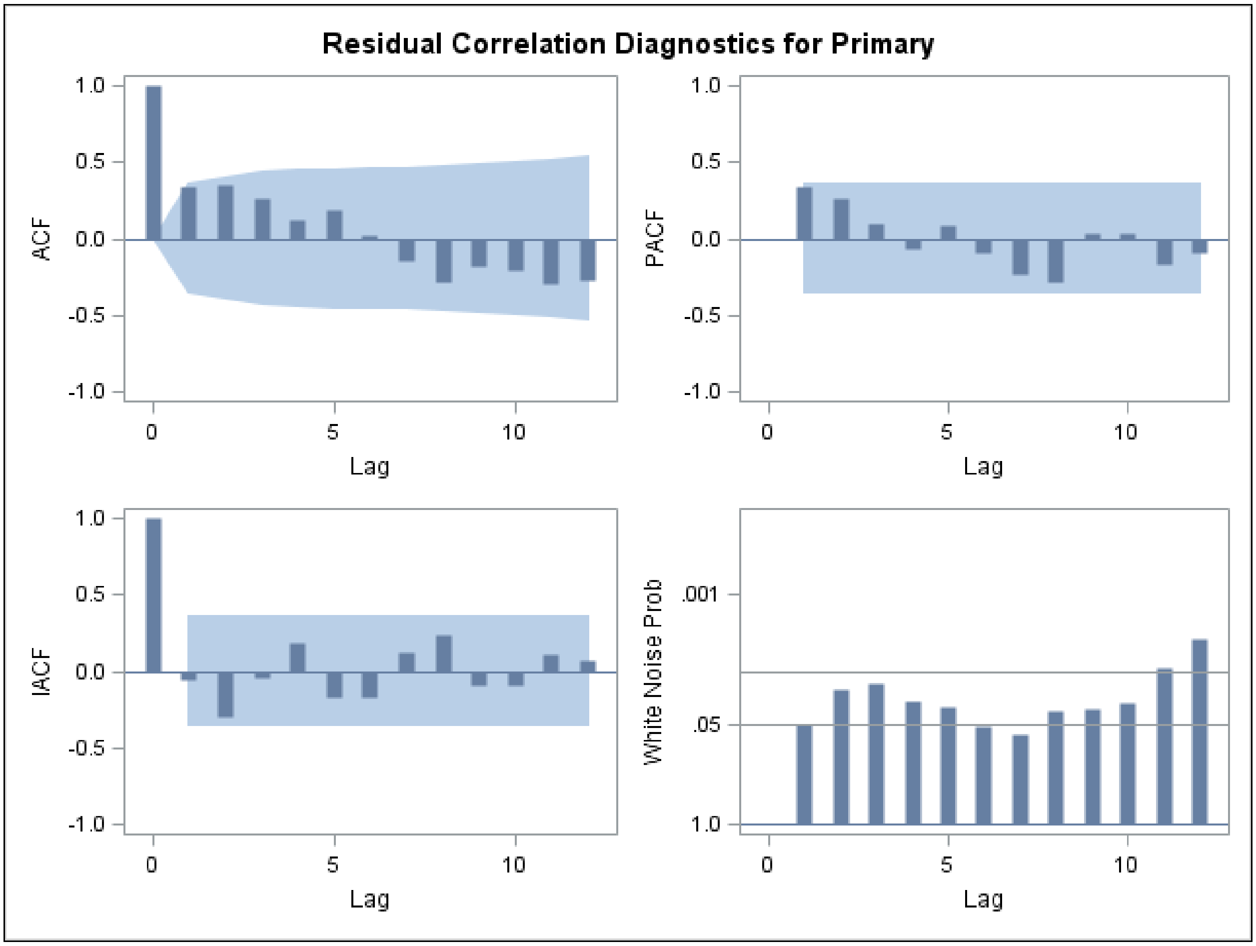
```
proc arima data=YearlyLead plots=all;
   identify var=Primary nlag=12 crosscorr=(Time);
   estimate input=(Time) plot  ml; /* White noise error */
run;
```

The **Time** variable provides the linear component. A residual analysis enables you to investigate whether the model is adequate, or whether you need to add an ARMA component for the error term.

Maximum Likelihood Estimation							
Parameter	Estimate	Standard Error	t Value	Approx Pr > |t|	Lag	Variable	Shift
MU	629858.1	14472.1	43.52	<.0001	0	Primary	0
NUM1	-11932.7	789.51895	-15.11	<.0001	0	Time	0

Constant Estimate	629858.1
Variance Estimate	1.5459E9
Std Error Estimate	39317.73
AIC	745.8315
SBC	748.6994
Number of Residuals	31

The autocorrelation plots suggest that the model is inadequate. Furthermore, they suggest possible models to use for the error.



No spikes are visible, but marginally significant bars at lag 2 of the PACF and IACF suggest an AR(2) model. You can continue the previous PROC ARIMA run by adding lines after the RUN statement. You should see the note “PROC ARIMA running” in the status window.

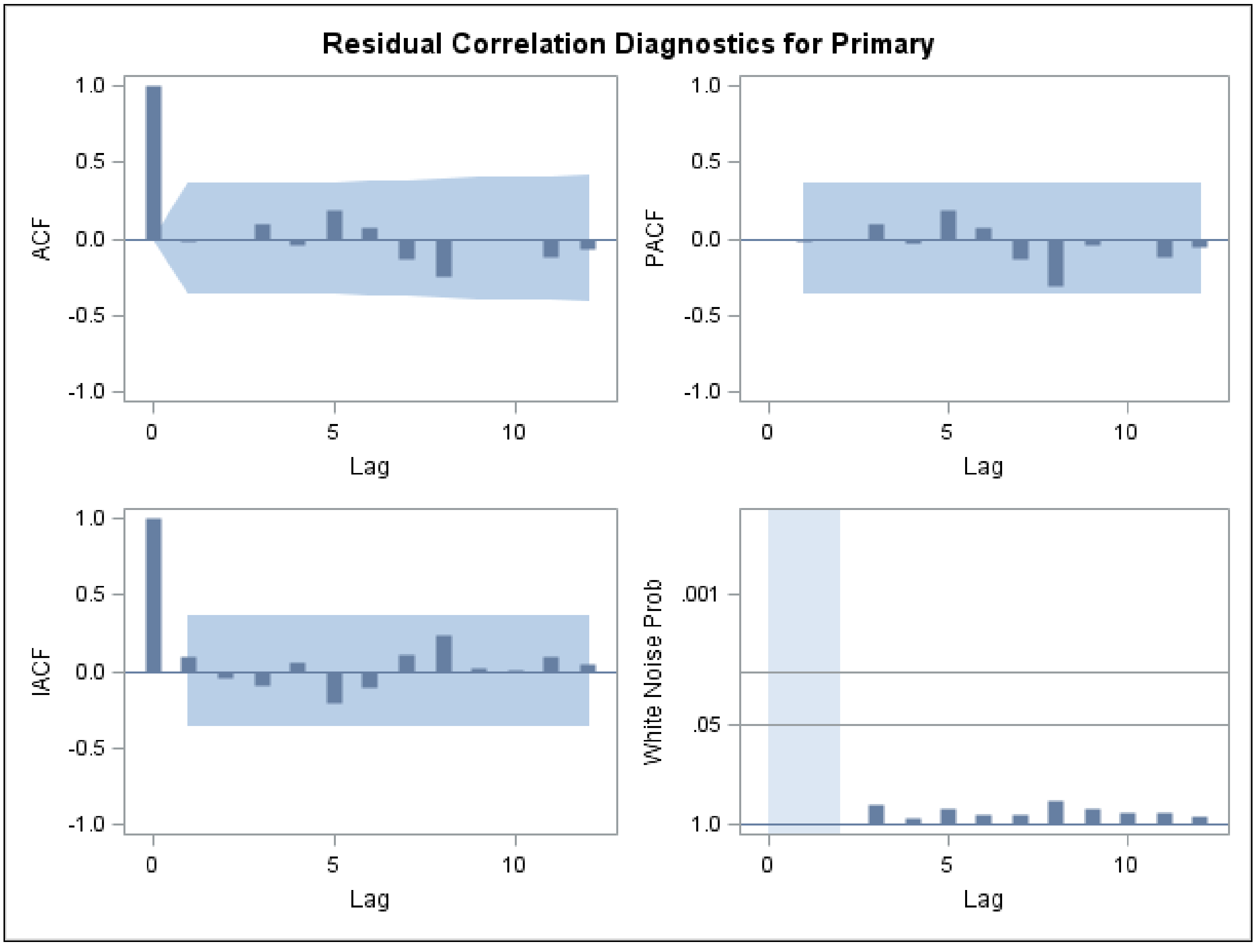
```
   estimate input=(Time) p=2   ml; /* lag 1 and 2        */
run;
```

The following table of estimates is produced for the trend plus AR(2) error model:

Maximum Likelihood Estimation							
Parameter	Estimate	Standard Error	t Value	Approx Pr > |t|	Lag	Variable	Shift
MU	631637.4	24004.1	26.31	<.0001	0	Primary	0
AR1,1	0.23686	0.18451	1.28	0.1992	1	Primary	0
AR1,2	0.25943	0.18529	1.40	0.1615	2	Primary	0
NUM1	-11984.6	1280.2	-9.36	<.0001	0	Time	0

Constant Estimate	318164
Variance Estimate	1.3651E9
Std Error Estimate	36946.84
AIC	744.0074
SBC	749.7433
Number of Residuals	31

The AR parameter estimates are not statistically significant. However, the model passes a white noise test.



You can try a subset AR(2) model by forcing the lag 1 coefficient to be zero.

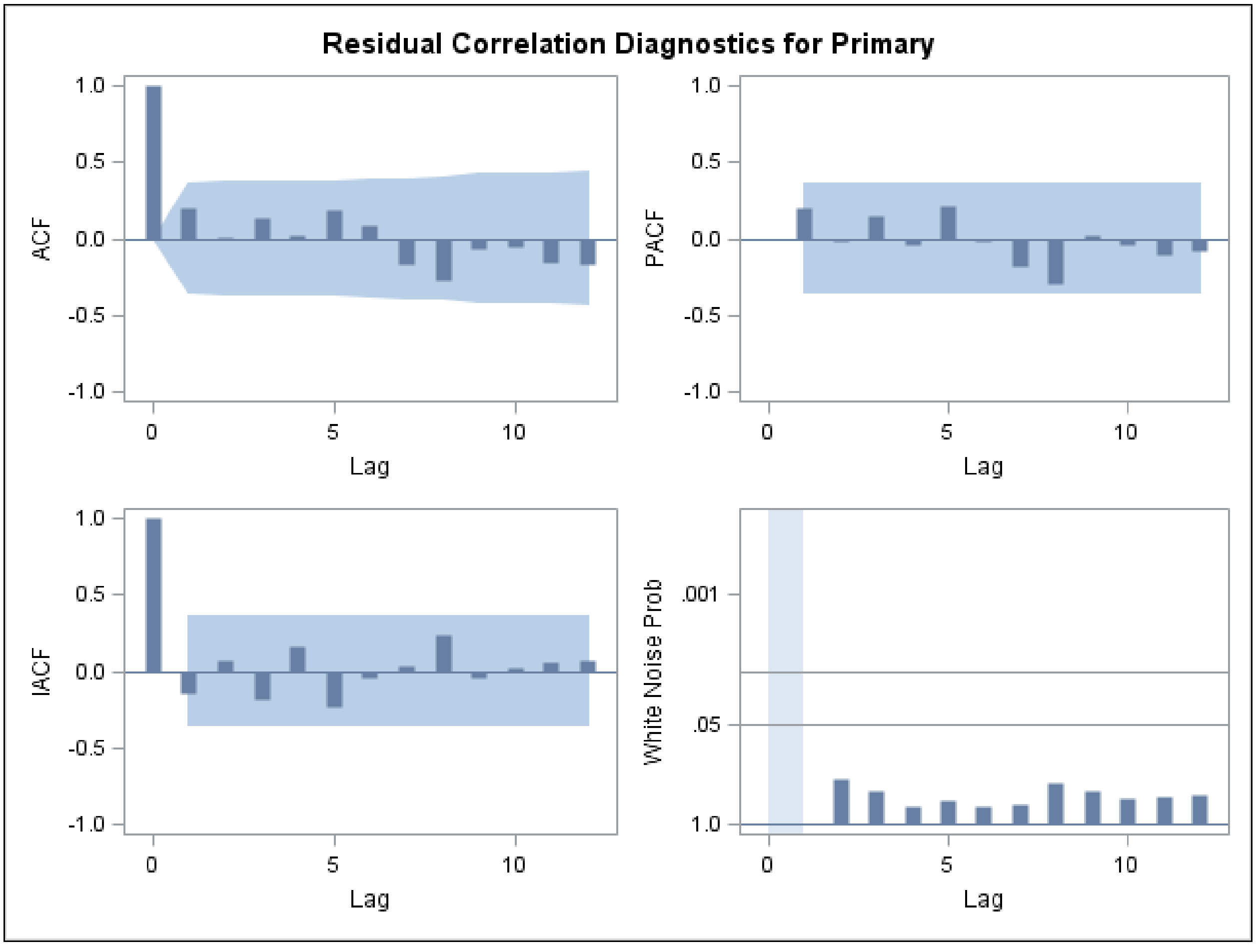
```
   estimate input=(Time) p=(2) ml; /* lag 2 only         */
run;
```

The table of estimates reveals that the lag 2 AR estimate is significant at the 10% level.

Maximum Likelihood Estimation							
Parameter	Estimate	Standard Error	t Value	Approx Pr > |t|	Lag	Variable	Shift
MU	631349.6	19137.2	32.99	<.0001	0	Primary	0
AR1,1	0.33693	0.17679	1.91	0.0567	2	Primary	0
NUM1	-11973.7	1028.5	-11.64	<.0001	0	Time	0

Constant Estimate	418630
Variance Estimate	1.401E9
Std Error Estimate	37430.58
AIC	743.935
SBC	748.237
Number of Residuals	31

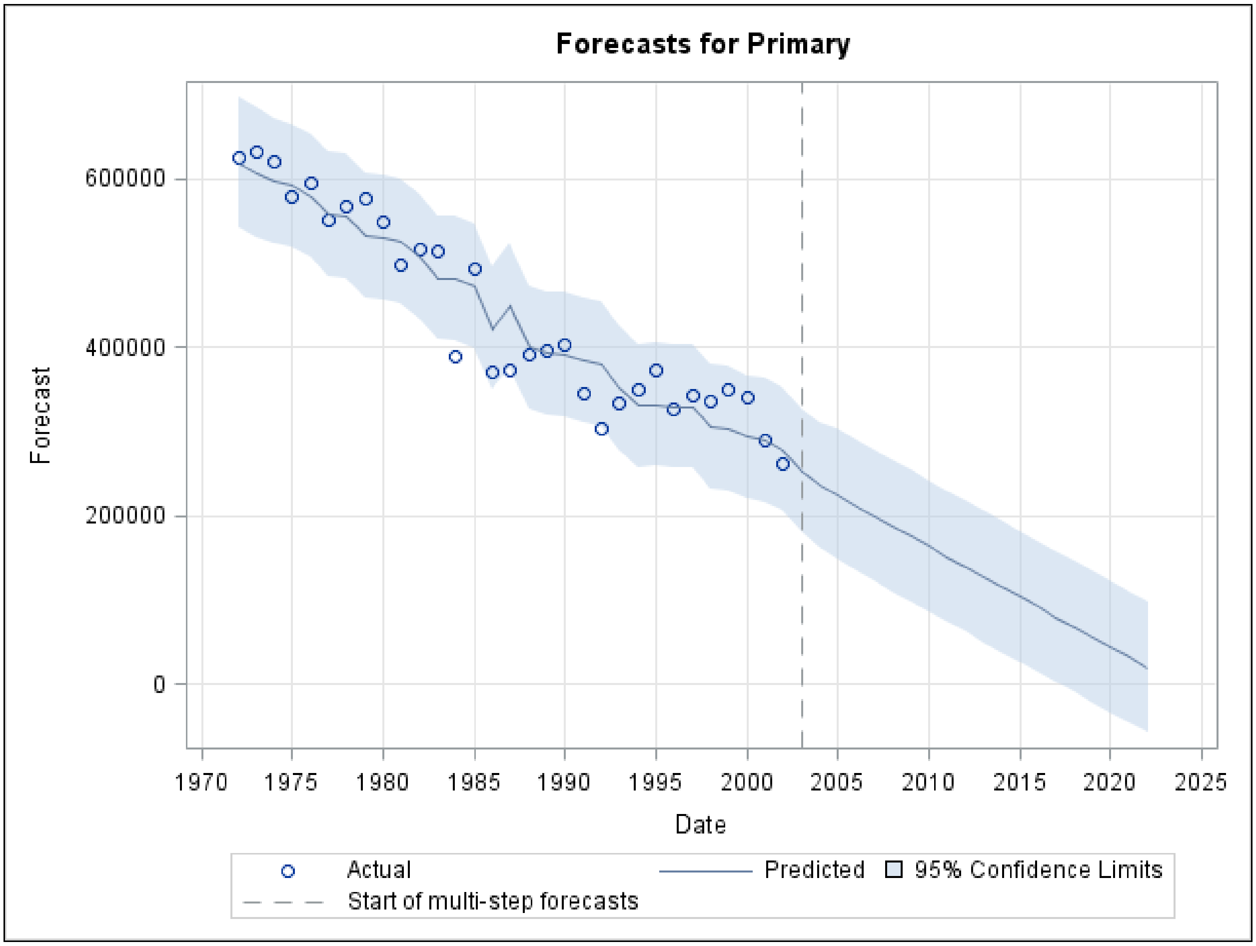
The autocorrelation plots for the residuals imply white noise.



The model cannot be disqualified. You can generate forecasts using this model by submitting a FORECAST statement. The forecast model is derived using the most recent ESTIMATE statement.

```
   forecast lead=20 id=date interval=year;
quit;
```

The QUIT statement terminates the PROC ARIMA session. The forecast plot shows that the AR portion of the forecast leads to reversion to the trend, which is the time-dependent mean.



The forecast horizon chosen with **lead**=20 is unrealistic and is used only for illustration. The long horizon shows that a linear decrease eventually results in negative primary production, which is not reasonable. One option is to replace any negative forecast with 0 and thus forecast a linear decrease to some year in the future when primary production is forecast to stop altogether. A second option is to use a linear trend model on the log transformed data. Even negative forecasts and prediction limits on the log scale transform to positive numbers on the original scale, as in Chapter 1. The forecasts were extended farther into the future (for illustration) than the length of historic data justifies. The third and best option is to argue that behavior of the forecasts that far out is irrelevant.

3.01 Multiple Choice Poll

What is a typical forecast horizon for your organization?

a. 1 to 5 time units
b. 6 to 10 time units
c. 11 to 20 time units
d. More than 20 time units

14

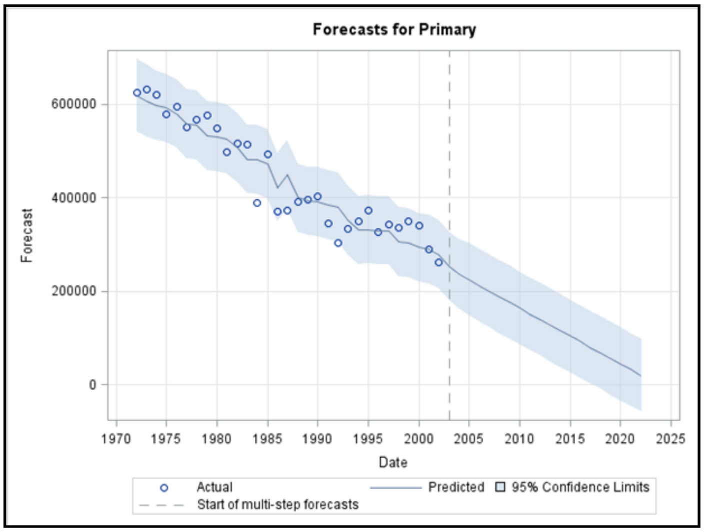
Lead Production: Linear Trend Plus AR(2)

- Trend model provides evidence of residual autocorrelation.
- Residual autocorrelation plots suggest an AR(1) or AR(2) error component.
- The subset AR(2) model with lag 2 only fits the best.
- An AR(2) error component with lags 1 and 2 might be preferred using domain knowledge or experience over a lag 2 only model.

15

From a practical perspective, it is difficult to justify that primary lead production two years ago is predictive, while lead production in the last year is not. Technically speaking, the model says that the deviations from a linear trend two years ago are predictive, but last year's deviations from linear trend are not. The linear component uses all years to calculate the slope and intercept, so in some sense, the model says that all previous years are predictive.

Lead Production: Linear Trend Plus AR(2)



16

As mentioned in the demonstration, the **lead**=20 forecast horizon was used to illustrate that linear trend forecasts can become negative.

A linear trend was fit to the yearly lead production data because of the obvious decrease seen in the graph. Another type of model that accounts for such behavior and the large coefficient when the AR(1) model was fit is the random walk with drift, which involves differencing the data. Tests for "unit roots" are available when, under the null hypothesis of unit roots, a difference needs to be taken.

Stochastic Trend Functions: Differencing

Random Walk:

$$Y_t = Y_{t-1} + \varepsilon_t$$

Random Walk with Drift:

$$Y_t = \theta + Y_{t-1} + \varepsilon_t$$

17

Stochastic Trend Functions: Differencing

Random Walk:

$$Y_t - Y_{t-1} = \varepsilon_t$$

$\Delta Y_t = Y_t - Y_{t-1}$ ⟵ First Difference

General Model with Stochastic Trend:

$$\Delta^k Y_t = SEASONAL_t + IRREGULAR_t$$

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

$\Delta^k Y_t$ ⟵ *k*-th Difference

18

Differencing Notation (Review)

$$Y_t - Y_{t-1} = \varepsilon_t$$

$\Delta Y_t = \varepsilon_t$ ⟵ First Difference

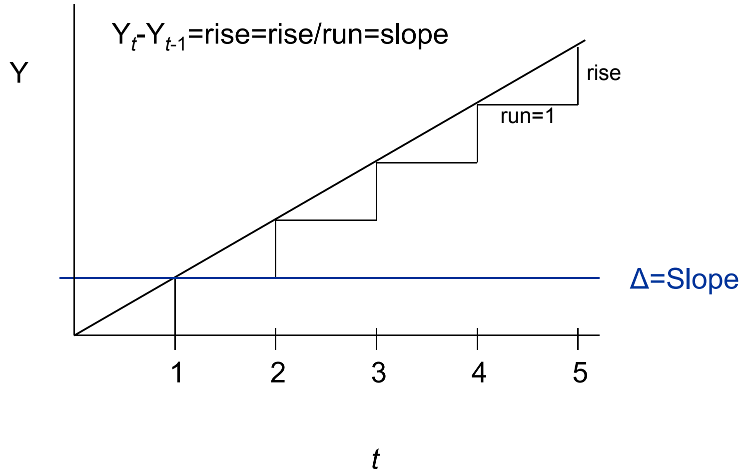
$(1-B)Y_t = \varepsilon_t$ ⟵ Backshift Notation

$\Delta^k Y_t = \varepsilon_t$ ⟵ k-th Difference

$(1-B)^k Y_t = \varepsilon_t$ ⟵ Backshift Notation

19

First Differences Turn Linear Trends into Horizontal Lines



20

Stochastic Seasonal Functions: Seasonal Differencing

- For seasonal data with period S, express the current value as a function that includes the value S time units in the past.
  - $Y_t = Y_{t-S} + TREND_t + IRREGULAR_t$
  - $\Delta_S Y_t = Y_t - Y_{t-S}$, called a *difference of order S*

Examples:

- Monthly: This January is a function of last January and so on.
- Daily: This Sunday is a function of last Sunday and so on.

21

Differencing Notation

$$Y_t - Y_{t-s} = \varepsilon_t$$

$$\Delta_s Y_t = \varepsilon_t \quad \longleftarrow \text{ Seasonal Difference}$$

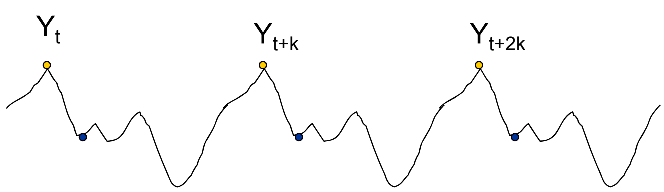
$$(1 - B^s)Y_t = \varepsilon_t \quad \longleftarrow \text{ Backshift Notation}$$

$$\Delta_s^k Y_t = \varepsilon_t \quad \longleftarrow \text{ } k\text{-th Seasonal Difference}$$

$$(1 - B^s)^k Y_t = \varepsilon_t \quad \longleftarrow \text{ Backshift Notation}$$

22

Periodic Differences Turn Periodic Data into Horizontal Lines



Δ_k=0

23

Differencing Notation for Trend and Seasonality

$(1-B^s)(1-B)Y_t = \varepsilon_t$ ← Backshift Notation

$(1-B-B^s+B^{s+1})Y_t = \varepsilon_t$ ← Polynomial Multiplication

$Y_t - Y_{t-1} - Y_{t-s} + Y_{t-s-1} = \varepsilon_t$ ← Apply Backshift Operator

$Y_t = Y_{t-1} + Y_{t-s} - Y_{t-s-1} + \varepsilon_t$ ← Model

$\hat{Y}_t = Y_{t-1} + Y_{t-s} - Y_{t-s-1}$ ← Forecast Equation

24

Seasonal models are discussed in the next chapter.

Trend and Seasonal Differencing in PROC ARIMA

General form of the IDENTIFY statement to specify differencing:

```
PROC ARIMA <options>;
      IDENTIFY VAR=variable <options>;
      ESTIMATE <options>;
      FORECAST OUT=SAS-data-set <options>;
RUN;
```

25

Trend and Seasonal Differencing in PROC ARIMA

General form for differencing in the ARIMA IDENTIFY statement:

> **IDENTIFY** VAR=*variable*<(*d1 d2 … dk*)>
> <*options*>**;**

- The list (*d1 d2 … dk*) provides differencing orders.
- VAR=Y(1 12) indicates first differencing (trend) and a seasonal difference of order 12.
- You can separate differencing lags by commas.

26

The Dickey-Fuller Unit Root Test

- This test provides a statistical test for first differencing.
- The null hypothesis is that first differencing is required.
- The alternative hypothesis has three forms:
  - Zero Mean
  - Single Mean
  - Trend
- Variations of the test exist for differences of order S so that you can test for seasonal differencing.

27

The Dickey-Fuller Zero Mean Test

Model: $$Y_t = \phi Y_{t-1} + \varepsilon_t$$

Null Hypothesis: $$H_0 : \phi = 1$$

Alternative Hypothesis: $$H_a : |\phi| < 1$$

28

The Dickey-Fuller Single Mean Test

Model: $$Y_t - \mu = \phi(Y_{t-1} - \mu) + \varepsilon_t$$

Null Hypothesis: $$H_0 : \phi = 1$$

Alternative Hypothesis: $$H_a : |\phi| < 1$$

29

The Dickey-Fuller Trend Test

Model: $Y_t - \beta_0 - \beta_1 t = \phi(Y_{t-1} - \beta_0 - \beta_1(t-1)) + \varepsilon_t$

Null Hypothesis: $H_0 : \phi = 1$

Alternative Hypothesis: $H_a : |\phi| < 1$

30

Augmented Dickey-Fuller (ADF) Tests

- The lag 0 test is equivalent to the tests described previously.
- The lag 1 test considers an AR(2) model.
- The lag 2 test considers an AR(3) model and so on.
- The null hypothesis is that the characteristic polynomial of the autoregressive model has a single unit root.
- The alternative hypothesis is that the characteristic polynomial is for a stationary AR process.

Characteristic Polynomial of an AR(p) Model

$$\phi(x) = 1 - \phi_1 x - \phi_2 x^2 - \cdots - \phi_p x^p$$

31

The augmented Dickey-Fuller lag 1 single mean test considers the following model:

$$(1 - \phi_1 B - \phi_2 B^2)(Y_t - \mu) = \varepsilon_t$$

and tests whether the characteristic polynomial has a single unit root. The model is often written with respect to the first difference, as shown below:

$$Y_t - Y_{t-1} = -(1 - \phi_1 - \phi_2)(Y_{t-1} - \mu) - \phi_2 (Y_{t-1} - Y_{t-2}) + \varepsilon_t$$

This is a model with one lagged difference. The parameter $(1 - \phi_1 - \phi_2)$ is the characteristic equation evaluated at B=1 and therefore will be 0 if 1 is a root.

The Dickey-Fuller Test in PROC ARIMA

PROC ARIMA <*options*>**;**
  IDENTIFY VAR=*variable*
    STATIONARITY=(ADF=(*AR orders*))
    <*options*>**;**
  ESTIMATE <*options*>**;**
  FORECAST OUT=*SAS-data-set* <*options*>**;**
RUN;

```
proc arima data=work.Groceries;
   identify var=Toothpaste nlags=12
            stationarity=(adf=(0 1 2 3 4 5));
```

32

The Dickey-Fuller Test in PROC ARIMA

Augmented Dickey-Fuller Unit Root Tests

Type	Lags	Rho	Pr < Rho	Tau	Pr < Tau	F	Pr > F
Zero Mean	0	-0.0214	0.6739	-0.12	0.6395		
	1	-0.0669	0.6636	-0.41	0.5309		
	2	-0.0265	0.6726	-0.22	0.6026		
	3	-0.0316	0.6713	-0.31	0.5682		
	4	-0.0152	0.6749	-0.18	0.6174		
	5	0.0005	0.6783	0.01	0.6803		
Single Mean	0	-25.0564	0.0012	-4.01	0.0027	8.03	0.0010
	1	-41.5691	0.0004	-4.97	0.0002	12.43	0.0010
	2	-34.8515	0.0004	-3.66	0.0075	6.73	0.0073
	3	-50.5816	0.0004	-3.75	0.0059	7.09	0.0010
	4	-53.8412	0.0004	-3.20	0.0260	5.13	0.0428
	5	-53.2356	0.0004	-2.71	0.0803	3.67	0.1669
Trend	0	-24.9941	0.0110	-3.95	0.0167	7.89	0.0167
	1	-40.8845	<.0001	-4.90	0.0012	12.60	0.0010
	2	-34.4841	0.0005	-3.66	0.0350	6.94	0.0452
	3	-48.3846	<.0001	-3.81	0.0247	7.83	0.0247
	4	-51.7982	<.0001	-3.31	0.0767	5.91	0.0868
	5	-54.5918	<.0001	-2.83	0.1951	4.18	0.3694

33

The Dickey-Fuller Test in PROC ARIMA

- The Rho test is the regression coefficient-based test statistic; it is also called the *normalized bias test*.
- The Tau test is the studentized test.
- The *F* test is the regression *F* test for the full model and the null hypothesis restricted reduced model, except that the distribution is not the usual *F* distribution used in ordinary regression.

Properties:

- The *F* test generally has the poorest power properties and is seldom recommended.
- The Rho test has superior power properties compared to Tau for lag 1 tests, but Tau is superior for other tests.

34

Unit Root Tests and Random Walks

This demonstration illustrates how to use PROC ARIMA to obtain augmented Dickey-Fuller test results and explains the test using a regression framework.

The code for this demonstration can be found in program **Demo3_02LeadRWD.sas**.

The following code obtains augmented Dickey-Fuller test results for specified lags 0 and 1:

```
proc arima data=work.YearlyLead plots=all;
   identify var=Primary stationarity=(adf=(0 1));
quit;
```

The ADF test was requested with 0 and 1 augmenting lags corresponding to 1 and 2 lagged levels respectively. The results appear below:

Augmented Dickey-Fuller Unit Root Tests

Type	Lags	Rho	Pr < Rho	Tau	Pr < Tau	F	Pr > F
Zero Mean	0	-0.9187	0.4802	-1.72	0.0807		
	1	-0.8463	0.4948	-2.65	0.0099		
Single Mean	0	-2.5853	0.6923	-1.16	0.6798	1.75	0.6343
	1	-1.3478	0.8410	-0.98	0.7481	3.46	0.2182
Trend	0	-19.9546	0.0273	-3.67	0.0405	6.74	0.0498
	1	-11.1161	0.2883	-2.22	0.4624	2.53	0.6813

Because there is a clear trend in the data, the only reasonable alternative to differencing is to fit a linear trend plus stationary errors so that only the trend test is of interest. The *p*-values are quite different in this small sample depending on the choice of augmenting lags. In theory, excessive lags cause a loss of power and too few invalidate the test. From the previous analyses, there is not very strong evidence of a need for lagged differences. Without lagged differences, the (lag 0) test suggests, though not strongly, that the series is stationary, that is, it rejects unit roots. That is, the *p*-value for the Rho lag 0 test is 0.0273, and for the Tau test it is 0.0405. Both show significance at the 5% level, which leads to a rejection of the null hypothesis that the series is not stationary. Unfortunately, the lag 1 tests both reveal large *p*-values, suggesting acceptance of the null hypothesis that the series is not stationary.

Because the number of lagged differences matters, it might be of interest to determine how many are needed. The theory behind the ADF test shows that when the differenced series Y_t-Y_{t-1} is regressed on the lagged level (and possibly an intercept and linear trend) and lagged differences, the usual tests, *t* and *F*, on the lagged differences have the usual distributions, for example, the *t* statistics have approximately standard normal distributions in large samples. The *t* test (Tau) on the lagged level as well as those for the intercept and trend have nonstandard distributions even in large samples. This means that in testing for the appropriate number of lagged differences, PROC REG or any other ordinary least squares procedure can be used to fit the model and test the lagged differences for significance. The following code sets up a temporary SAS data set to enable you to exploit PROC REG to determine the appropriate augmenting lag:

```
data CheckLags;
   set sasuser.LeadYear;
   Time+1;
   D=Dif(Primary);
   L=lag(Primary);
   D1=lag1(D);
   D2=lag2(D);
   D3=lag3(D);
run;
```

For the yearly **lead** data, the regression of the differences **D** on the lagged level **L** along with an intercept, linear trend, and three lagged differences (**D1**, **D2**, **D3**) shows no evidence (P=0.53) that any of them are needed. The following code implements the test:

```
proc reg data=CheckLags;
          model D=L Time D1 D2 D3/ss1;
   Augment3: test D1=0, D2=0, D3=0;
quit;
```

The Augment3 test results follow:

Test Augment3 Results for Dependent Variable D				
Source	DF	Mean Square	F Value	Pr > F
Numerator	3	1249207853	0.76	0.5311
Denominator	21	1651724512		

The *t* tests on the output for **D1**, **D2**, and **D3** have appropriate *p*-values while those on the **Intercept**, **Time**, and **L** have nonstandard distributions so computation of *p*-values for these using a *t* distribution as any ordinary least square regression does, would not be appropriate.

An X is placed beside the affected *p*-values to indicate that they are invalid.

Parameter Estimates						
Variable	**DF**	**Parameter Estimate**	**Standard Error**	**t Value**	**Pr > |t|**	**Type I SS**
Intercept	1	257340	194602	1.32	0.2003 X	3721814815
L	1	-0.44208	0.28961	-1.53	0.1418 X	3444350081
Time	1	-4811.92335	3736.16959	-1.29	0.2118 X	16466996795
D1	1	-0.36613	0.31444	-1.16	0.2573	3057245737
D2	1	-0.10869	0.31584	-0.34	0.7342	654090507
D3	1	0.03549	0.23944	0.15	0.8836	36287314

As a final check, a new PROC REG is run with only one lagged difference. The code to derive the test follows:

```
proc reg data=CheckLags;
          Model D=L Time D1;
   Augment1: test D1=0;
quit;
```

The estimated parameters appear in the following table:

Parameter Estimates					
Variable	**DF**	**Parameter Estimate**	**Standard Error**	**t Value**	**Pr > |t|**
Intercept	1	292195	143543	2.04	0.0525
L	1	-0.48798	0.22003	-2.22	0.0359
Time	1	-5535.49215	2769.58276	-2.00	0.0566
D1	1	-0.27306	0.19125	-1.43	0.1657

By running a new PROC REG rather than only adding another model to the existing PROC REG, the observations lost to differencing **D2** and **D3** are recovered, that is, only the first two observations have no **D1** (D1 = $Y_{t-1}-Y_{t-2}$). As before, **D1** is not significant. Notice that the *t* test on **L** has the value -2.22. This *t* statistic also appears in the PROC ARIMA output for the trend test with one lagged difference.

Trend	0	-19.9546	0.0273	-3.67	0.0405	6.74	0.0498
	1	-11.1161	0.2883	-2.22	0.4624	2.53	0.6813

The correct p-value 0.4624 for Tau = −2.22 is a far cry from the 0.0359 reported for the same t = −2.22 in the PROC REG output. In that output, only the p-value for **D1** is justified by statistical theory. In summary, there is no compelling reason to use the lag 1 test because there seems no need for a lagged difference in the model. The p-value 0.0405 is thus justified and you have significant (though barely) statistical evidence of stationarity around a trend.

Differencing tends to produce nice fits to data, but a price is paid in wide forecast intervals. Differencing when the data have trend plus stationary errors should not be done. Here is an example of a trend plus AR(1) error:

$$Y_t = \beta_0 + \beta_1 t + Z_t, \quad Z_t, = \phi_1 Z_{t-1} + \varepsilon_t \text{ with } |\phi_1| < 1$$

Lagging ($Y_{t-1} = \beta_0 + \beta_1(t-1) + Z_{t-1}$) and subtracting produces $Y_t – Y_{t-1} = \beta_1 + (Z_t – Z_{t-1})$, a model with error term ($Z_t – Z_{t-1}$). Using $Z_t = \phi_1 Z_{t-1} + \varepsilon_t$, you have the following by lagging ($Z_{t-1} = \phi_1 Z_{t-2} + \varepsilon_{t-1}$) and subtracting:

$$(Z_t – Z_{t-1}) = \phi_1(Z_{t-1} – Z_{t-2}) + \varepsilon_t - \theta_1\varepsilon_t \text{ where } \theta_1 = 1$$

The error term ($Z_t – Z_{t-1}$) in this differenced series is thus an ARMA(1,1) with moving average parameter $\theta_1 = 1$. The series ***should not*** be differenced. Evidence of such overdifferencing (evidence that $\theta_1 = 1$) is given when the one of the following occurs:

1. The estimate of θ_1 is near 1. (Be careful. Distribution is nonstandard.)

2. The IACF resembles the ACF of a random walk (that is, it dies off slowly).

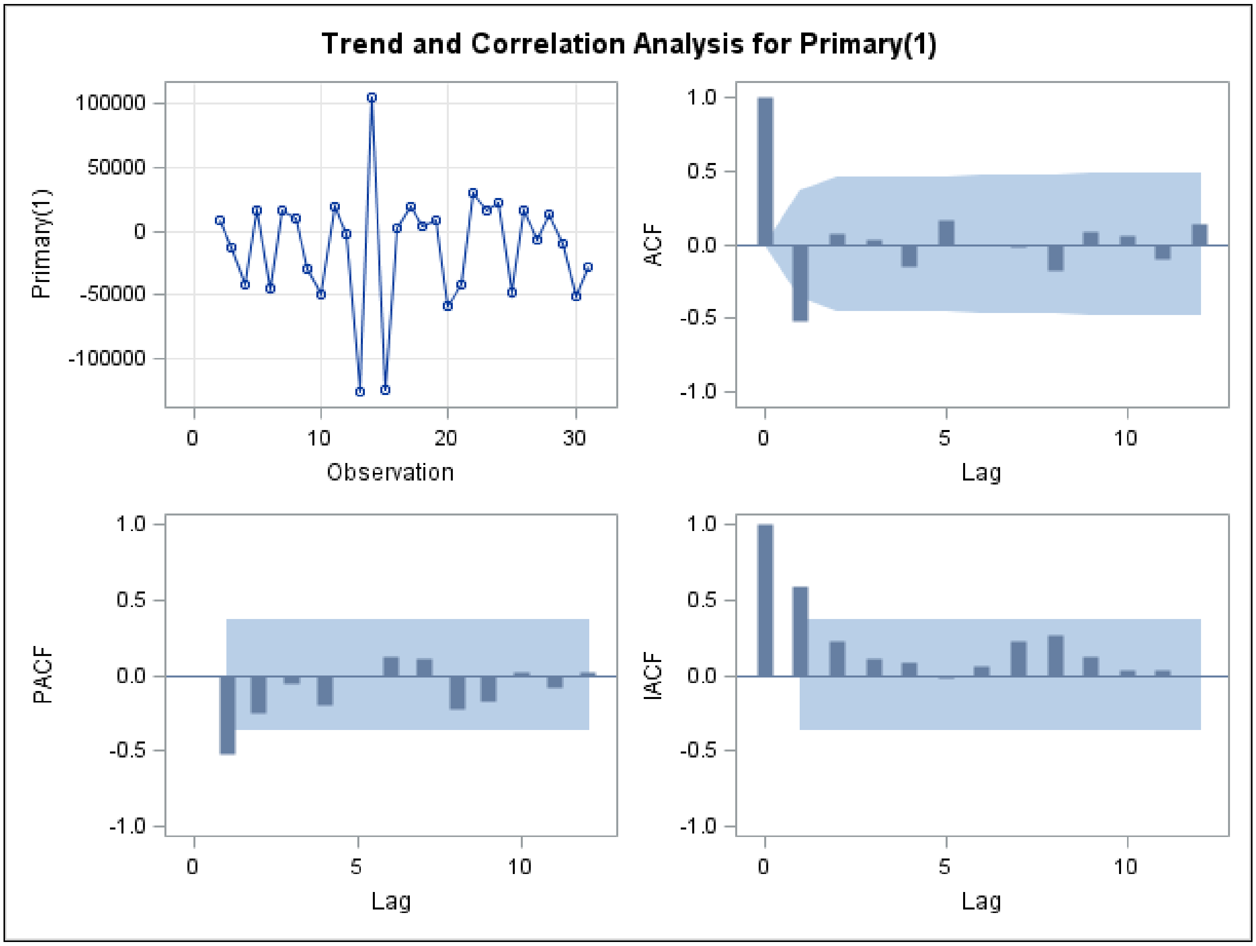
If $\phi_1 = 0$, the model is $Y_t = \beta_0 + \beta_1 t + \varepsilon_t$ and the suggestion not to difference still holds. The error term for the differences $Y_t – Y_{t-1}$ is still $(Z_t – Z_{t-1}) = \varepsilon_t - \theta_1\varepsilon_t$ where $\theta_1 = 1$.

In contrast, if the autoregressive coefficient is $\phi_1 = 1$, then $(Z_t – Z_{t-1}) = \varepsilon_t$ and the differenced model $Y_t - Y_{t-1} = \beta_1 + (Z_t - Z_{t-1})$ becomes $Y_t – Y_{t-1} = \beta_1 + \varepsilon_t$, that is, the differences are only a mean plus white noise. Differencing ***is*** appropriate if $\phi_1 = 1$. In that case, the data satisfies a random walk with the drift model.

The code to fit the above models follows:

```
title2 font=&coursefont color=black "Models with Differencing";
proc arima data=work.YearlyLead plots=all;
   identify var=Primary(1) nlag=12;  /* (1) -> first difference */
   estimate q=1 ml;                  /* like ESM but with drift */
   estimate  ;                       /* RWD                     */
   forecast lead=20 id=date interval=year;
quit;
```

The first ESTIMATE statement fits an MA(1) model to the differenced series. This is similar to a certain exponential smoothing model that is introduced later. Notice that the IDENTIFY statement specifies a first difference, so both the IDENTIFY and the ESTIMATE statements are required to get the complete model specification. Results for this model appear below, but first the results for the differenced series appear in the output.



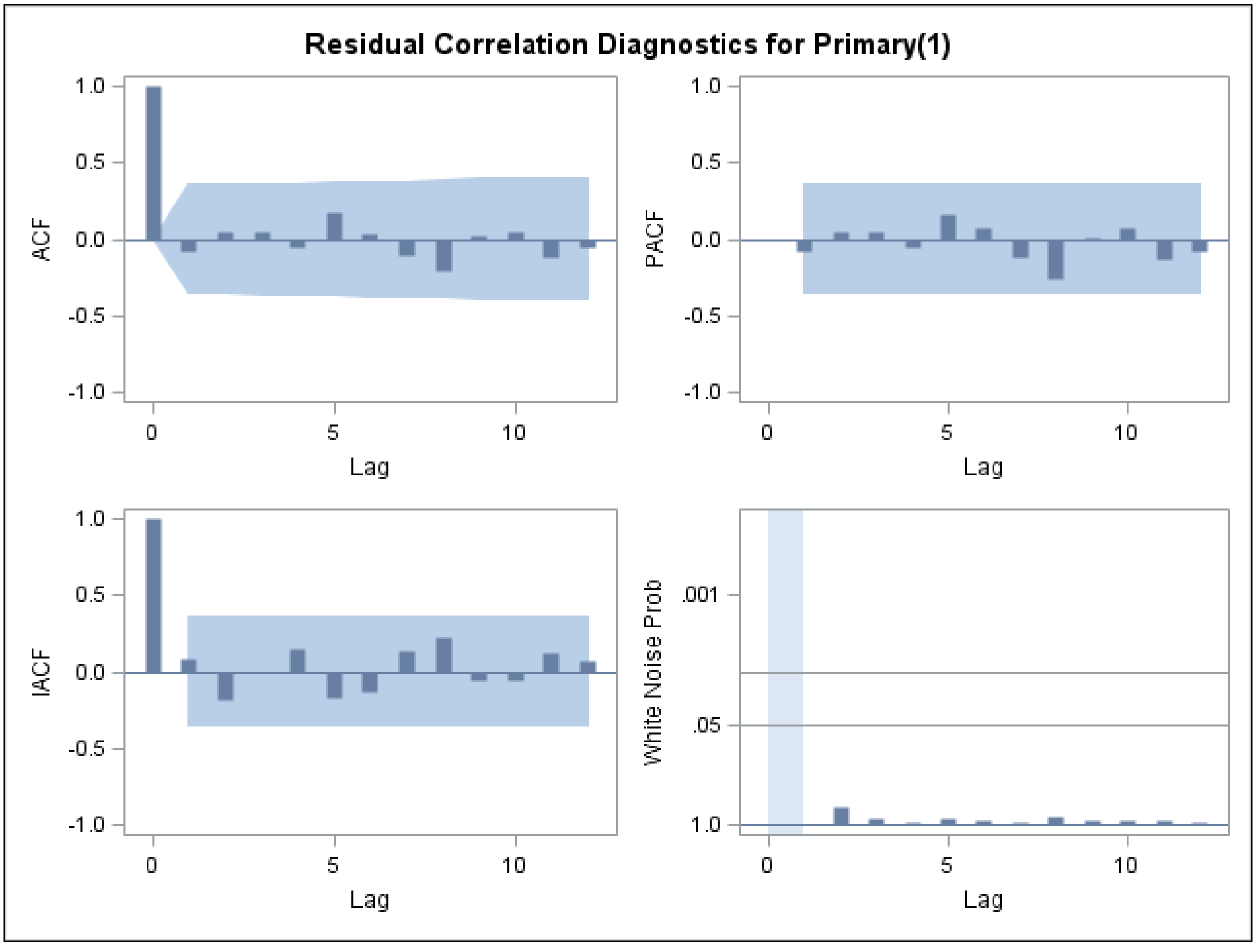
The sample autocorrelation functions suggest that the differenced series can be modeled by an MA(1) model, although other models are suggested.

The table of estimates for the MA component appears below:

Maximum Likelihood Estimation					
Parameter	Estimate	Standard Error	t Value	Approx Pr > |t|	Lag
MU	-11721.2	2700.6	-4.34	<.0001	0
MA1,1	0.62817	0.14955	4.20	<.0001	1

Constant Estimate	-11721.2
Variance Estimate	1.4292E9
Std Error Estimate	37805.19
AIC	719.9805
SBC	722.7829
Number of Residuals	30

The residual autocorrelation plots follow:



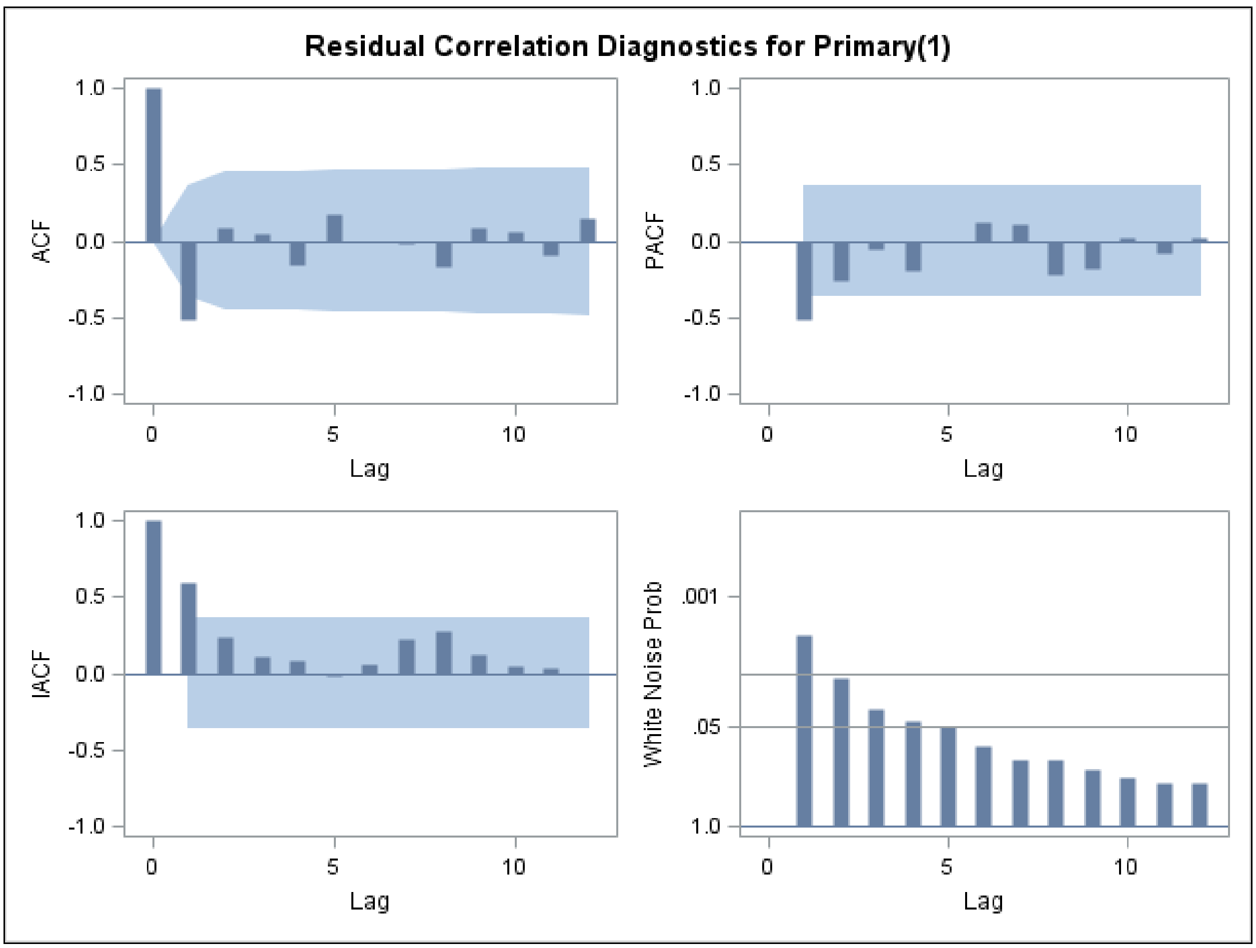
The differences of the yearly **lead** data show hints of overdifferencing, but the evidence is not strong. This might be due to the small sample size. The moving average coefficient is 0.62817, which is large but not strikingly close to 1, and the IACF of the differences has some fairly large spikes but they do not persist at long lags. This mild evidence of overdifferencing along with the ADF test that suggests the residuals are stationary adds some skepticism about the appropriateness of differencing, but a model in differences is nevertheless pursued in the interest of completeness.

The random walk with drift has a single drift parameter to estimate.

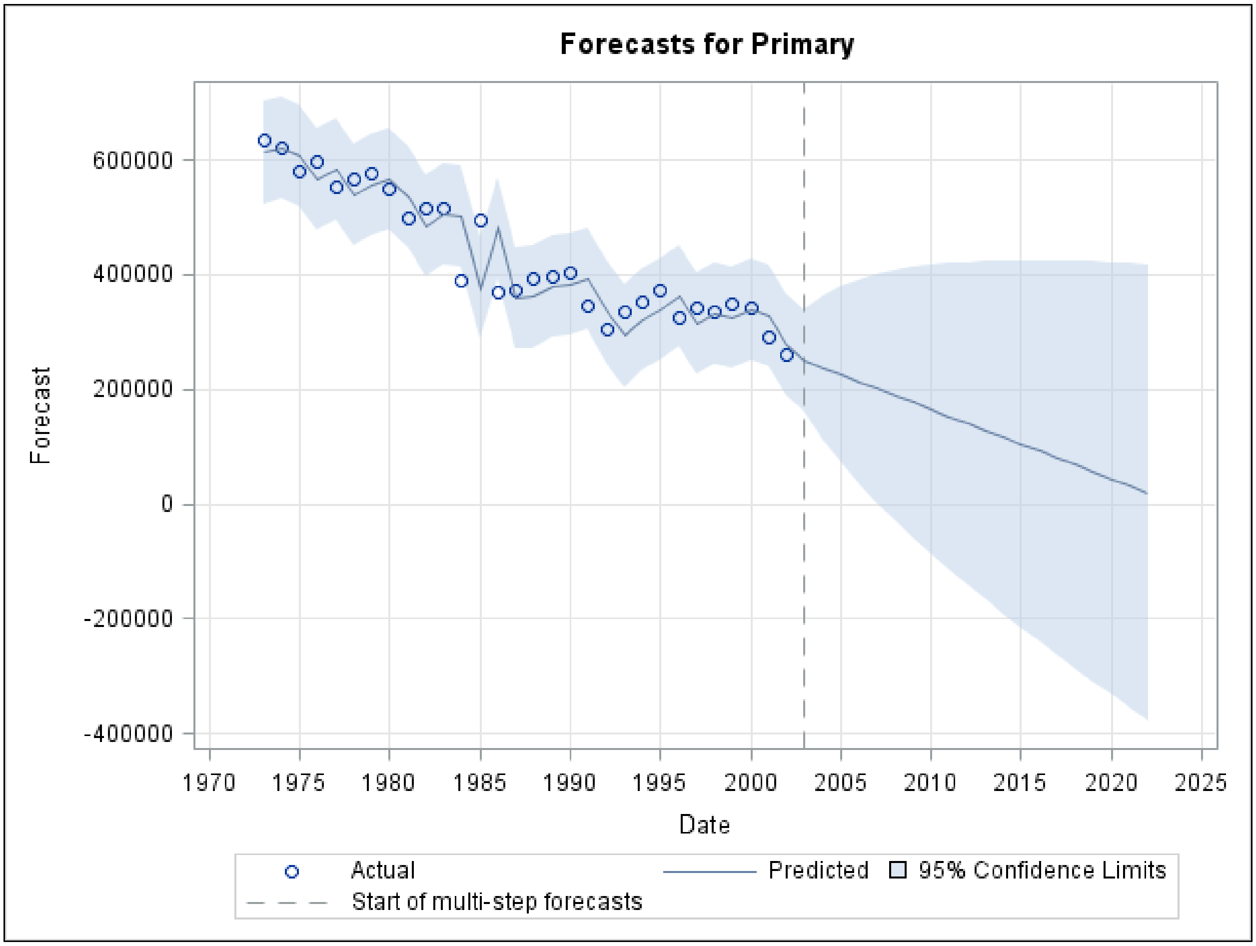
Conditional Least Squares Estimation					
Parameter	Estimate	Standard Error	t Value	Approx Pr > |t|	Lag
MU	-12100.0	8276.0	-1.46	0.1545	0

Constant Estimate	-12100
Variance Estimate	2.0548E9
Std Error Estimate	45329.71
AIC	729.4223
SBC	730.8235
Number of Residuals	30

The residual plots follow:



The random walk with drift model has significant lack of fit due to residual autocorrelation. Nonetheless, forecasts are generated for this model because it is the last ESTIMATE statement before the FORECAST statement.



As before, the long forecast horizon is employed to emphasize the drift component of the forecast. A random walk with drift produces a similar long term linear trend as a deterministic trend model. However, the confidence intervals for the predictions are much wider.

To illustrate a more dramatic example of overdifferencing as described above, use the following code:

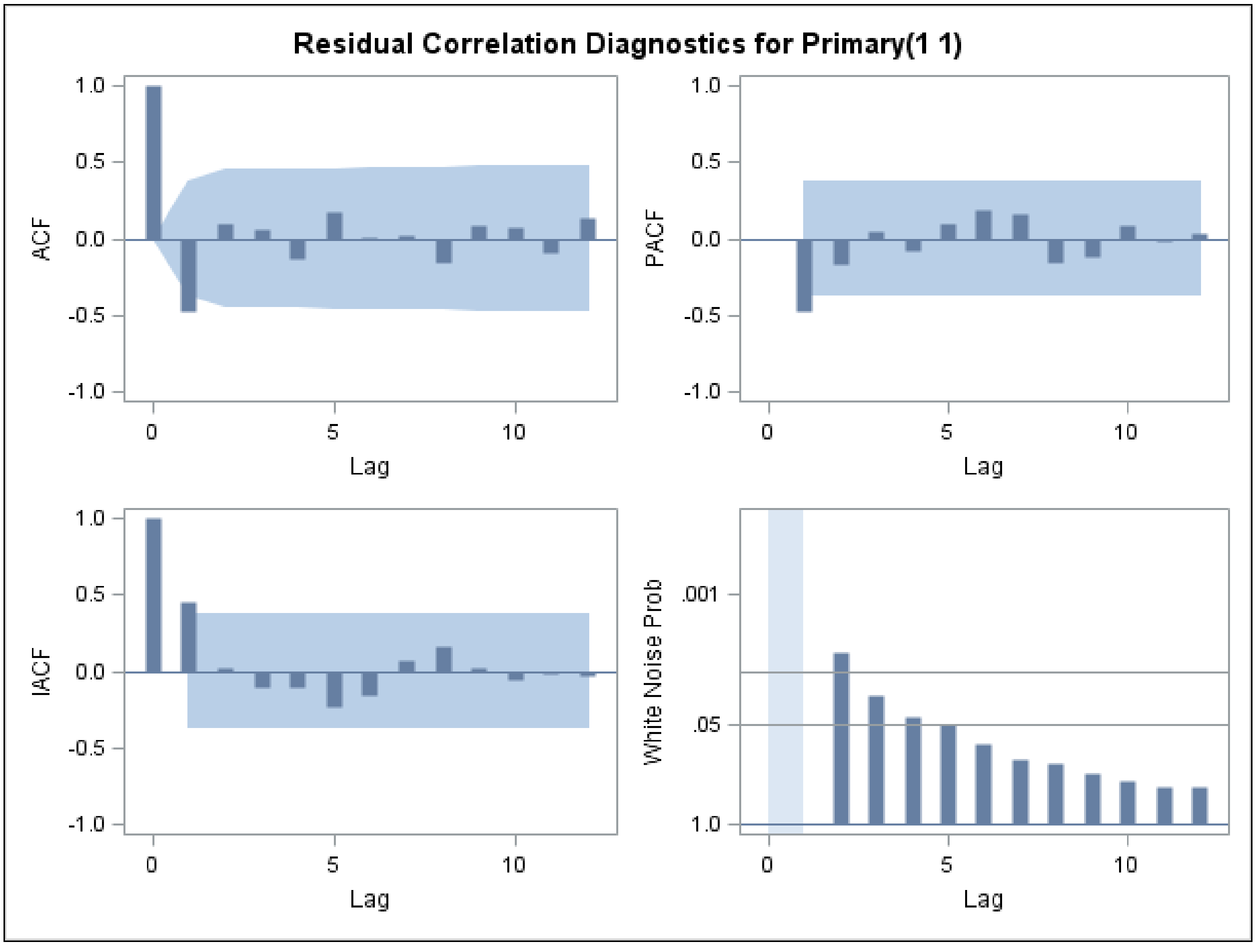
```
title2 font=&coursefont color=black "Example of Overdifferencing";
proc arima data=work.YearlyLead plots=all;
   identify var=Primary(1 1) nlag=12;
   estimate q=1 ml;
quit;
```

The estimated MA(1) parameter is very near 1. In fact, you get a convergence warning because the estimation algorithm is trying to push the parameter beyond the boundary of 1. (This boundary relates to another condition, called *invertibility*, that is not discussed here.)

Maximum Likelihood Estimation					
Parameter	Estimate	Standard Error	t Value	Approx Pr > |t|	Lag
MU	-992.53452	1649.9	-0.60	0.5474	0
MA1,1	0.99985	82.44186	0.01	0.9903	1

Constant Estimate	-992.535
Variance Estimate	2.2997E9
Std Error Estimate	47955.54
AIC	712.7488
SBC	715.4834
Number of Residuals	29

The large standard error provides strong evidence of estimation problems and suggests that the *t* statistic and *p*-value are unreliable.



The model could still be deployed, because it is a valid mathematical model that generates forecasts. However, residual autocorrelation plots suggest that the model should be disqualified. The primary source of the lack of fit is overdifferencing.

Lead Production: Differencing

- The first differences of annual primary lead production seem to be autocorrelated.
- Large IACF values might suggest overdifferencing.
- Differencing+CONSTANT+MA(1)
  - Is an exponential smoothing model if the MA parameter is positive
  - Has a trrending forecast if a constant is estimated.

The constant is the drift term.

36

You saw two approaches to modeling nonstationary time series:

- Linear trend plus ARMA error
- First difference plus ARMA error (including a drift parameter)

The next section looks at a more challenging data set and introduces some new deterministic trend functions along with outlier detection.

3.2 Modeling Trend

Objectives

- Explore deterministic trend curves.
- Illustrate how PROC ARIMA can be used to fit advanced models with trend components.
- Show how to use PROC ARIMA to detect outliers.

38

Two Types of Trend

- Deterministic
  - A mathematical function of time
    - Common functions are linear, quadratic, logarithmic, and exponential.
- Stochastic
  - Future time values depend on past values plus error.
  - A common stochastic trend model is a random walk with drift.

39

Common Deterministic Trend Functions

- Linear
- Quadratic
- Logarithmic
- Exponential — Linear for log(TARGET)

40

Deterministic Trend in PROC ARIMA

PROC ARIMA <*options*>**;**
  IDENTIFY VAR=*variable*
    CROSSCORR=(*trend-variable*)
    <*options*>**;**
  ESTIMATE INPUT=(*trend-variable*) <*options*>**;**
  FORECAST OUT=*SAS-data-set* <*options*>**;**
RUN;

```
proc arima data=work.YearlyLead;
   identify var=Primary nlags=12
            crosscorr=(_LINEAR_);
   estimate INPUT=(_LINEAR_);
```

LINEAR is a variable in work.YearlyLead.

41

Stochastic Trend Components

- Random walk
- Random walk with drift

42

Stochastic Trend in PROC ARIMA

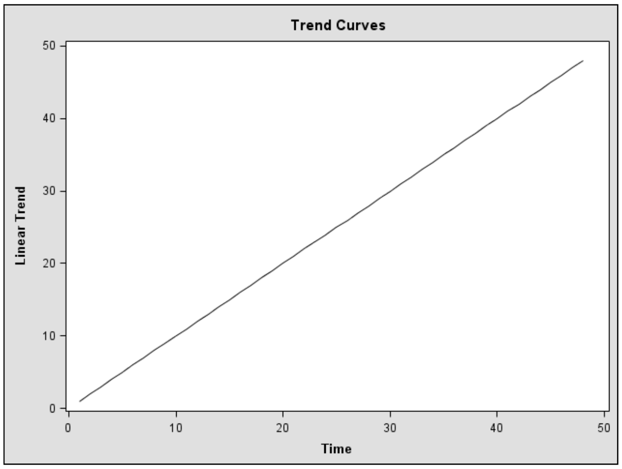
PROC ARIMA <*options*> ;
    IDENTIFY VAR=*variable*(d1 d2)
        <*options*> ;
    ESTIMATE <*options*> ;
    FORECAST OUT=*SAS-data-set* <*options*> ;
RUN;

```
proc arima data=work.Air1990_1997;
   identify var=Passengers(1 12)
            nlags=24;
   estimate q=(1)(12) method=ml;
```

43

Linear Trend

$$T_t = \beta_0 + \beta_1 t$$



44

continued...

Linear Trend

$$T_t = \beta_0 + \beta_1 t$$

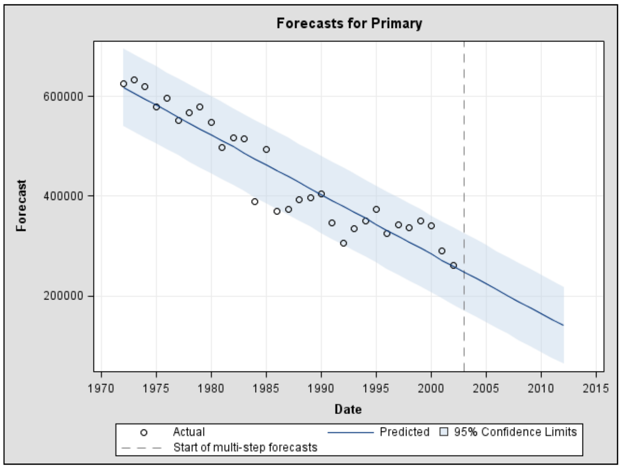
```
data work.MyData;
   set work.MyData;
   attrib _LINEAR_ label="Linear Trend";
   retain _LINEAR_ 0;
   _LINEAR_+1;
   output;
run;
proc arima data=work.MyData;
   identify var=Y cross=(_LINEAR_);
   estimate input=(_LINEAR_) method=ml;
   forecast lead=10;
run;
```

45

continued...

Linear Trend

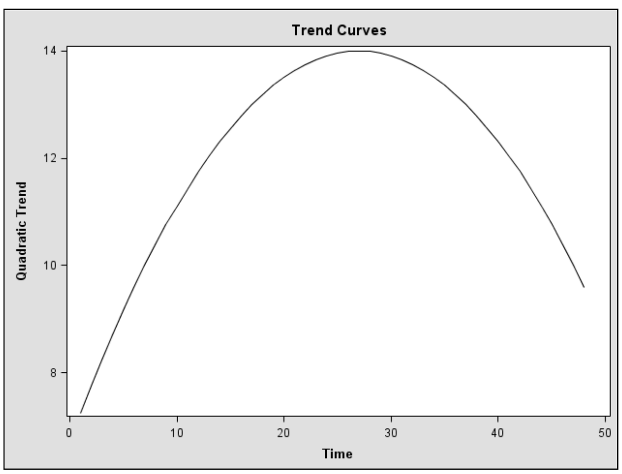
$$T_t = \beta_0 + \beta_1 t$$



46

Quadratic Trend

$$T_t = \beta_0 + \beta_1 t + \beta_2 t^2$$



47

continued...

Quadratic Trend

$$T_t = \beta_0 + \beta_1 t + \beta_2 t^2$$

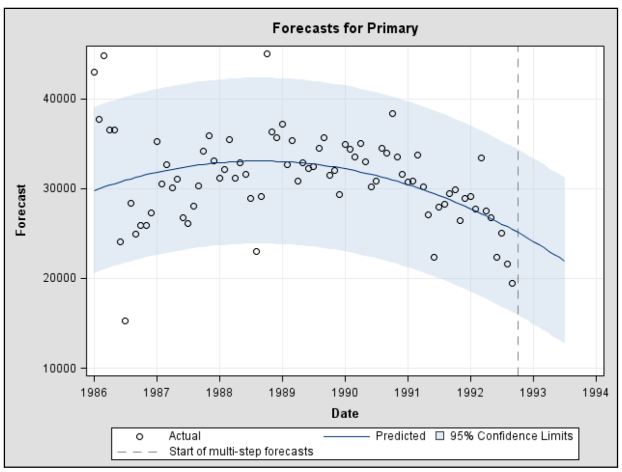
```
data work.MyData;
   set work.MyData;
   attrib _LINEAR_ label="Linear Term"
          _QUAD_   label="Quadratic Term";
   retain _LINEAR_ 0;
   _LINEAR_+1;
   _QUAD_=_LINEAR_*_LINEAR_;
   output;
run;
proc arima data=work.MyData;
   identify var=Y cross=(_LINEAR_ _QUAD_);
   estimate input=(_LINEAR_ _QUAD_) method=ml;
   forecast lead=10;
run;
```

48

continued...

Quadratic Trend

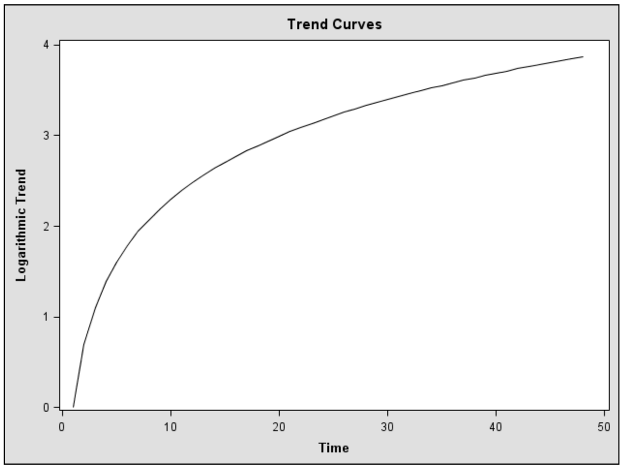
$$T_t = \beta_0 + \beta_1 t + \beta_2 t^2$$



49

Logarithmic Trend

$$T_t = \beta_0 + \beta_1 \log(t), \beta_1 > 0$$

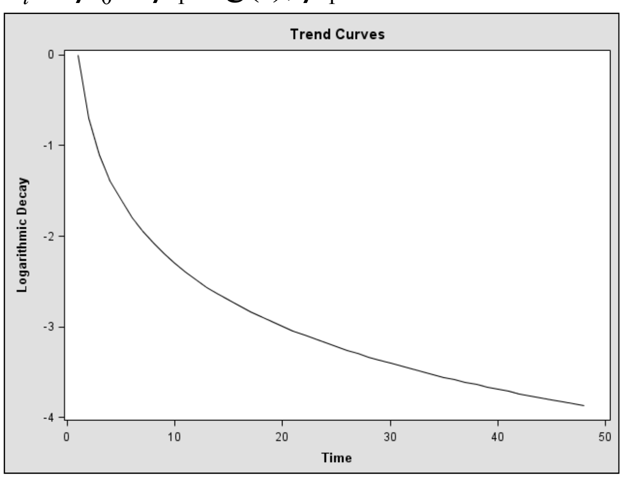


50

continued...

Logarithmic Trend

$$T_t = \beta_0 + \beta_1 \log(t), \beta_1 < 0$$



51

continued...

Logarithmic Trend

$$T_t = \beta_0 + \beta_1 \log(t)$$

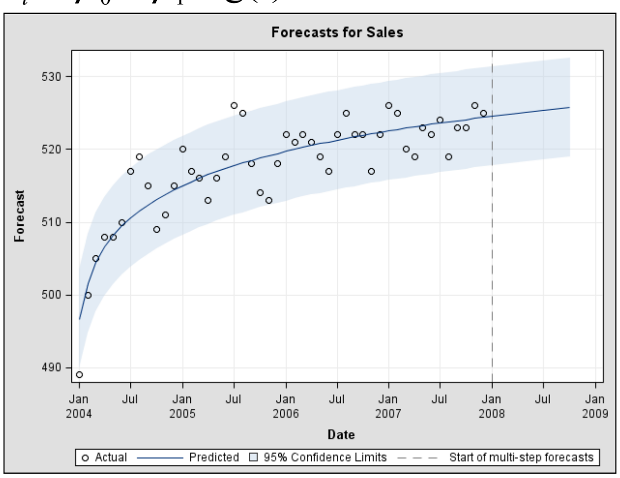
```
data work.MyData;
   set work.MyData;
   attrib _LOG_ label="Logarithmic Trend";
   retain Time 0;
   Time+1;
   _LOG_=log(Time);
   output;
run;
proc arima data=work.MyData;
   identify var=Y cross=(_LOG_);
   estimate input=(_LOG_) method=ml;
   forecast lead=10;
run;
```

52

continued...

Logarithmic Trend

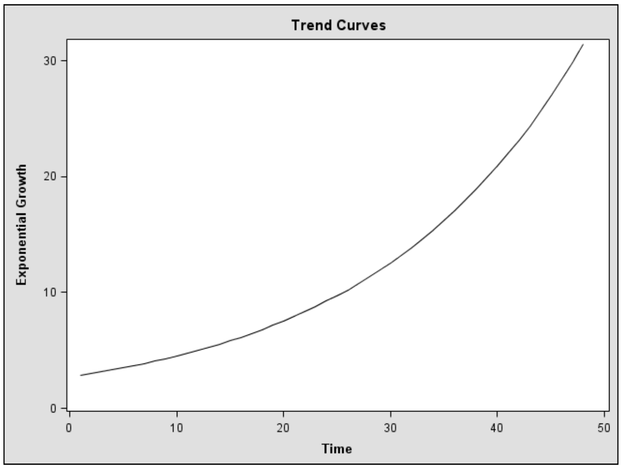
$$T_t = \beta_0 + \beta_1 \log(t)$$



53

Exponential Trend

$$T_t = \exp(\beta_0 + \beta_1 t), \beta_1 > 0$$

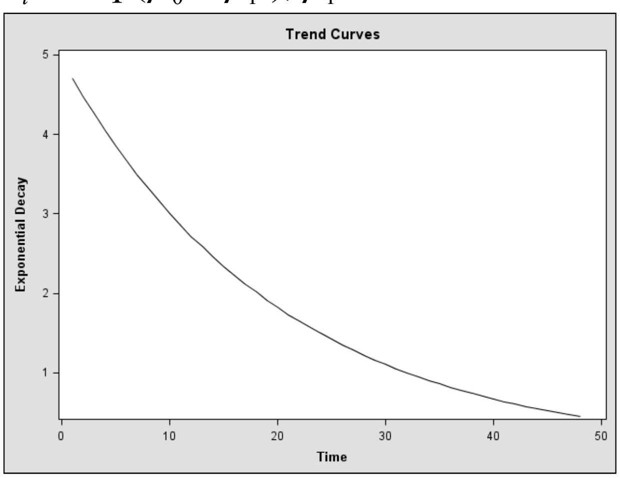


54

continued...

Exponential Trend

$$T_t = \exp(\beta_0 + \beta_1 t), \beta_1 < 0$$



55

continued...

Exponential Trend

$$T_t = \exp(\beta_0 + \beta_1 t)$$

```
data work.MyData;
   set work.MyData;
   attrib _LINEAR_ label="Linear Trend";
   retain _LINEAR_ 0;
   _LINEAR_+1;
   LogY=log(Y);
   output;
run;
```

56

continued...

Exponential Trend

$$T_t = \exp(\beta_0 + \beta_1 t)$$

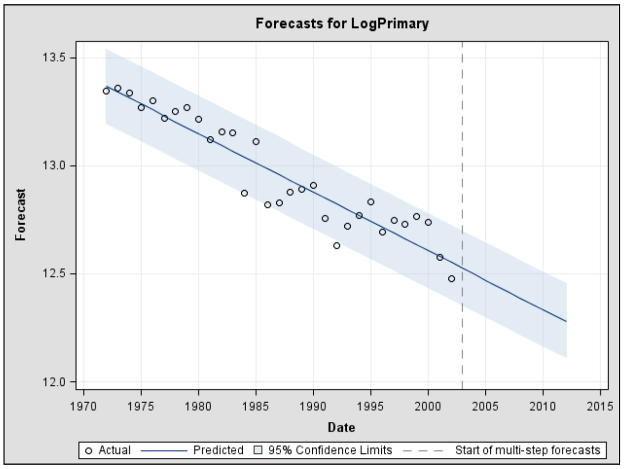
```
proc arima data=work.MyData;
   identify var=LogY cross=(_LINEAR_);
   estimate input=(_LINEAR_) method=ml;
   forecast lead=10 out=work.forecasts;
run;
data work.forecasts;
   set work.forecasts;
   Y=exp(LogY);
   L95=exp(L95);
   U95=exp(U95);
   FORECAST=exp(FORECAST+STD*STD/2);
run;
```

57

continued...

Exponential Trend

$$T_t = \exp(\beta_0 + \beta_1 t)$$

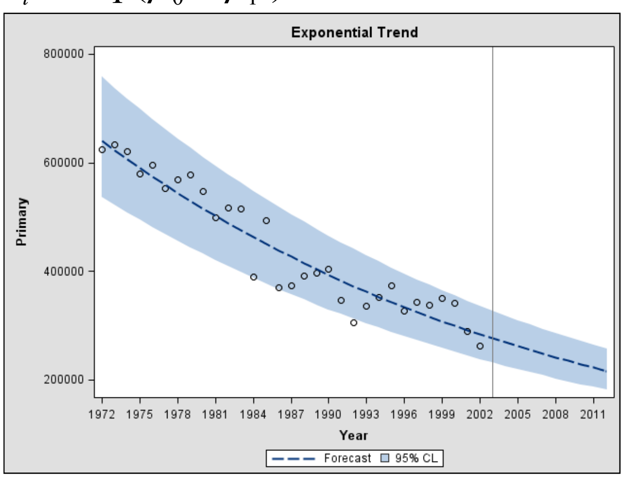


58

continued...

Exponential Trend

$$T_t = \exp(\beta_0 + \beta_1 t)$$



59

3.02 Multiple Answer Poll

What types of trend components have you used in forecast models for real (versus textbook) business problems?

a. Differencing
b. Linear
c. Quadratic
d. Higher order polynomial
e. Logarithm
f. Exponential
g. Logistic
h. Other

61

The Box-Jenkins Trend Model

- ARIMA=**A**uto**R**egressive **I**ntegrated **M**oving **A**verage
- Discrete Function: Difference
- Continuous Function: Derivative

To undo a difference or derivative, you ***integrate***, so an ARMA model for a difference must be integrated to get forecasts for the original data.

62

The Box-Jenkins Trend Model

ARIMA(*p*,*d*,*q*)

$$(1-\phi_1 B-\cdots-\phi_p B^p)(1-B)^d Y_t=\theta_0+(1-\theta_1 B-\cdots-\theta_q B^q)\varepsilon_t$$

ARIMA(2,1,1)

$$(1-\phi_1 B-\phi_2 B^2)(1-B)Y_t=\theta_0+(1-\theta_1 B)\varepsilon_t$$

ARIMA(2,1,1)

$Z_t-\phi_1 Z_{t-1}-\phi_2 Z_{t-2}=\theta_0+\varepsilon_t-\theta_1\varepsilon_{t-1}$ ← ARMA(2,1) Component

$Z_t=Y_t-Y_{t-1}$ ← Difference Component

63

Box-Jenkins Methodology for the Trend Model

Identify Example

Y_t	Stationary? Dickey-Fuller suggests d=1.
$Z_t=Y_t-Y_{t-1}$	Stationary? Dickey-Fuller rejects Null Hypothesis, differences are stationary.
$\phi(B)Z_t=\theta(B)\varepsilon_t$	Model? Diagnostics suggest p=1, q=1.

```
proc arima data=work.MyData;
   identify var=Sales nlags=12
            stationarity=(adf=(0 1 2 3 4 5));
   identify var=Sales(1) nlags=12
            stationarity=(adf=(0 1 2 3 4 5))
            esacf minic scan;
```

64

continued...

Box-Jenkins Methodology for the Trend Model

Estimate Example

$$Z_t - \phi_1 Z_{t-1} = \theta_0 + \varepsilon_t - \theta_1 \varepsilon_{t-1}$$

Estimate Using
Maximum Likelihood

Are the parameter estimates reasonable? Significant?

Do the residuals appear to be white noise?

```
proc arima data=work.MyData;
   identify var=Sales(1) nlags=12 noprint;
   estimate p=1 q=1 method=ml plot;
```

65

continued...

Box-Jenkins Methodology for the Trend Model

Forecast Example

$$\hat{Z}_t = \hat{\phi}_1 Z_{t-1} + \hat{\phi}_2 Z_{t-2} + \cdots + \hat{\phi}_{t-1} Z_1,$$
$$\hat{Y}_t = Y_{t-1} + \hat{Z}_t$$

Finite Memory
Forecast Equations

```
proc arima data=work.MyData;
   identify var=Sales(1) nlags=12 noprint;
   estimate p=1 q=1 method=ml noprint;
   forecast lead=12;
run;
```

66

Box-Jenkins Forecasting

- PROC ARIMA associates maximum likelihood estimation and ***finite memory forecasting***.
- Finite memory forecasting uses the estimated model to derive weights to be applied to the entire series, not only to the last max(p,q) values.
- Infinite memory and finite memory forecasts tend to be very similar, especially for long series.
- Infinite memory forecasts are easily coded in a spreadsheet application.
- Finite memory forecasts are difficult to code in a spreadsheet application without matrix operators.

67

Box-Jenkins Models with Deterministic Trend

$$(1-\phi_1 B-\phi_2 B^2-\cdots-\phi_p B^p)(Y_t-\mu_t)=(1-\theta_1 B-\theta_2 B^2-\cdots-\theta_q B^q)\varepsilon_t$$

$$\mu_t = f(t)$$

$f(t)$ is a deterministic function of time.

✐ The model equation above is a special case of the time series regression model presented in a later chapter.

68

Box-Jenkins Methodology for Time Series with Trend

Identify step: Determine (*p,d,q*) for an ARIMA(*p,d,q*) model.

- The value d=1 is analogous, but not equivalent to, linear trend. (The first derivative of a line is a constant.)
- The value d=2 is analogous, but not equivalent to, quadratic trend. (The second derivative of a parabola is a constant.)
- It is rare for d to be larger than 2.

69

Diagnosing Trend

- Plot the time series.
- Perform statistical tests for trend.
- Diagnose, fit, and evaluate trend models, including diagnosing ARMA models for the residual series.
- Evaluate goodness-of-fit/accuracy for the candidate models.
- Generate forecasts.

70

TIMESERIES Procedure

PROC TIMESERIES DATA=*SAS-data-set*
OUT=*SAS-data-set*
PRINT=*(options)*
PLOT=*(options)*
SEASONALITY=*n*;
ID *variable* INTERVAL=*interval*;
VAR *variable*;
DECOMP *options /* MODE=*ADD|MULT|others*;
RUN;

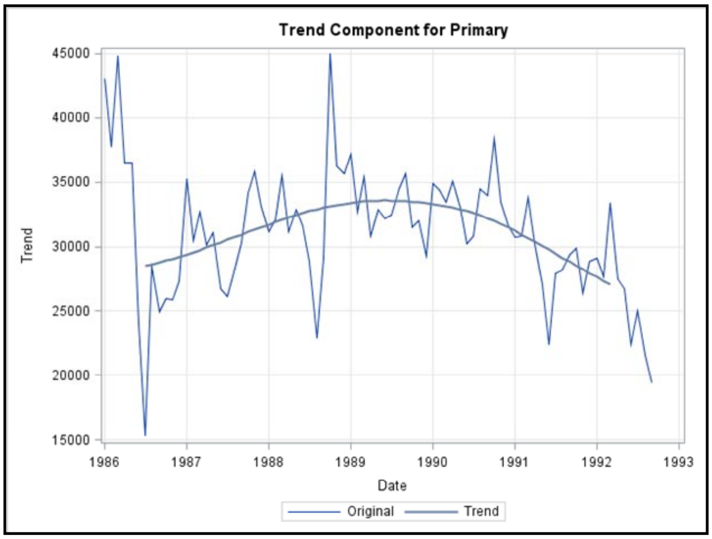
71

PROC TIMESERIES Diagnostics

```
proc timeseries
     data=sasuser.LeadMonth
     print=(descstats)
     plot=(series corr acf pacf iacf wn)
     seasonality=12;
   id Date interval=year;
   var Primary;
   decomp tcc sc / mode=mult;
run;
```

72

PROC TIMESERIES Trend



73

Detecting Outliers

- PROC ARIMA can detect outliers.
- Outliers are relative to a model.
- Three types of outliers are included in the search: ADDITIVE outlier (AO), level SHIFT (LS), and TEMPorary change (TC). TEMP is associated with durations d1, d2, … d*n*.

```
PROC ARIMA <options>;
    IDENTIFY VAR=variable <options>;
    ESTIMATE <options>;
    OUTLIER TYPE=(AO|LS|TC(d1 d2)) MAXNUM=n
            ID=variable <options>;
    FORECAST OUT=SAS-data-set <options>;
RUN;
```

74

Detecting Outliers

```
proc arima data=work.monthlead;
   identify var=Primary
            crosscorr=(_LINEAR_ _QUAD_);
   estimate p=1 input=(_LINEAR_ _QUAD_) ml;
   outlier type=(ao ls) maxnum=10 id=Date;
run;
```

Outlier Details					
Obs	**Time ID**	**Type**	**Estimate**	**Chi-Square**	**Approx Prob>ChiSq**
34	OCT1988	Additive	12235.5	15.59	<.0001
7	JUL1986	Additive	-11788.3	16.83	<.0001
3	MAR1986	Additive	8586.2	9.60	0.0019
6	JUN1986	Additive	-7767.7	8.34	0.0039
1	JAN1986	Additive	8291.3	6.87	0.0087
32	AUG1988	Additive	-6678.7	5.61	0.0178
75	MAR1992	Additive	5911.3	4.55	0.0329
13	JAN1987	Additive	5729.1	4.57	0.0326
66	JUN1991	Additive	-5325.7	4.08	0.0433
58	OCT1990	Additive	5156.1	4.38	0.0363

75

The monthly primary lead production time series poses a challenging forecasting problem that will illustrate modeling with trend.

Forecasting Monthly Lead Production

This demonstration illustrates the forecast modeling process using the monthly lead production data.

The code for this demonstration can be found in **Demo3_03LeadMonthly.sas**.

PROC TIMESERIES can be used to produce the time series plot and the autocorrelation plots. In addition, you can use the decomposition feature to diagnose trend and seasonality. The following code produces the necessary diagnostic plots:

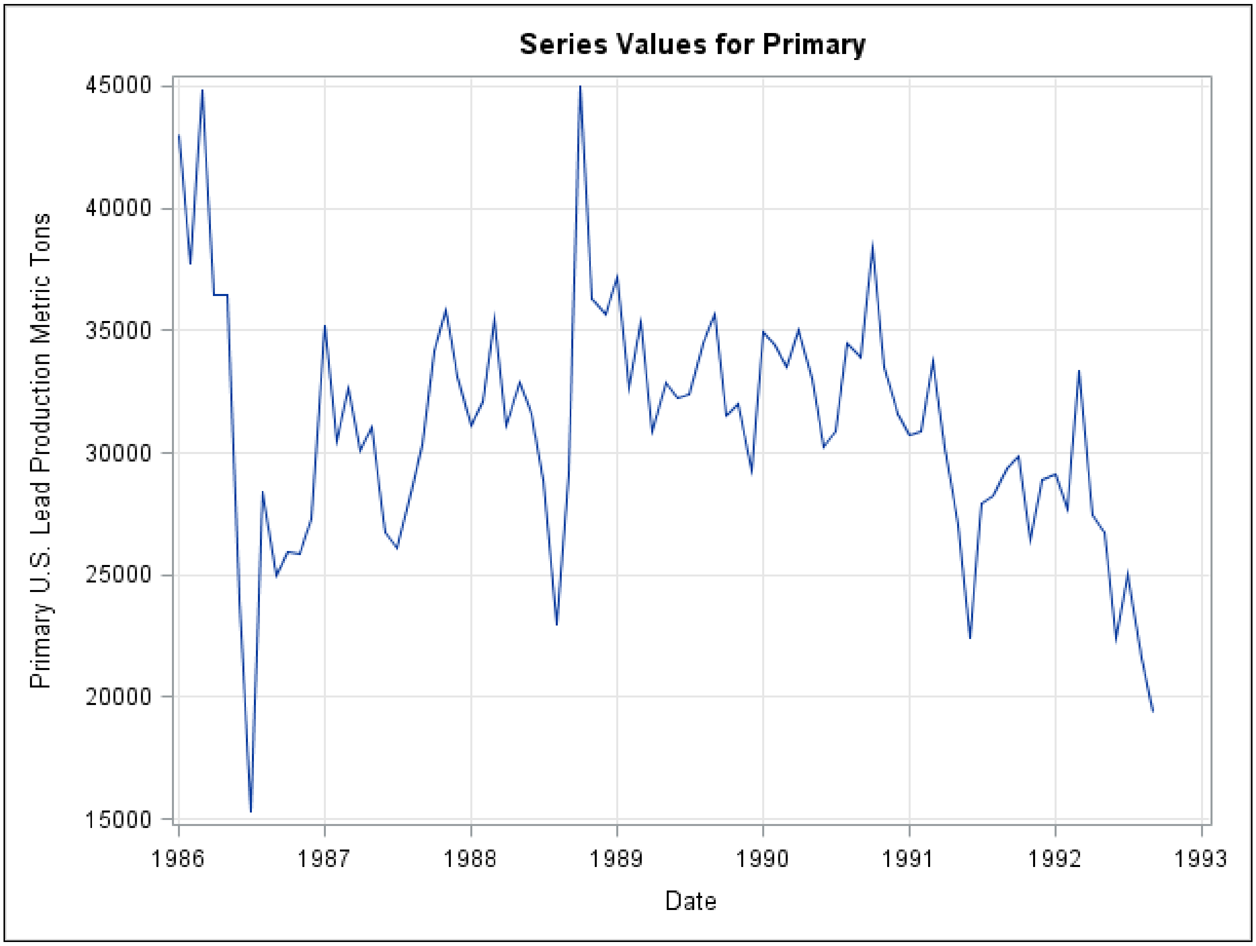
```
proc timeseries data=sasuser.monthlead
                plot=(series corr acf pacf iacf wn decomp tc sc )
                seasonality=12;
   id Date interval=month;
   var Primary;
   decomp tcc sc / mode=mult;
run;
```

Before examining the plots, a quick check of the log file reveals the following note:

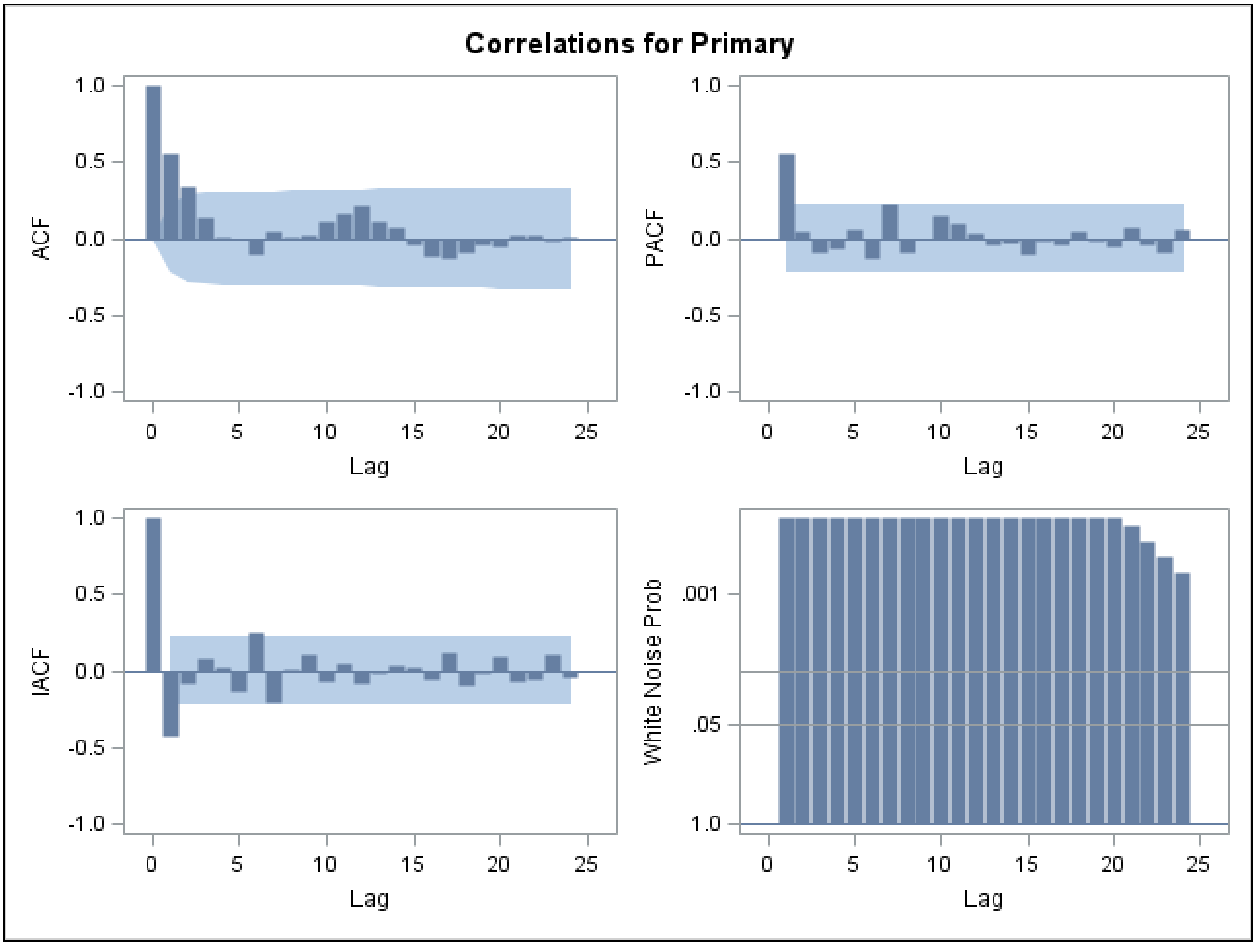
```
NOTE: The data set WORK.DATA1 has 81 observations and 2 variables.
```

If the OUT= option is not specified, PROC TIMESERIES automatically creates an output data set and uses the usual **SAS naming convention for naming unspecified data, namely WORK.DATA1, WORK.DATA2**, and so on, depending on whether previous data sets were created. You can delete this data set because it only contains a copy of the original data. Under some options, the data set can contain forecasts.

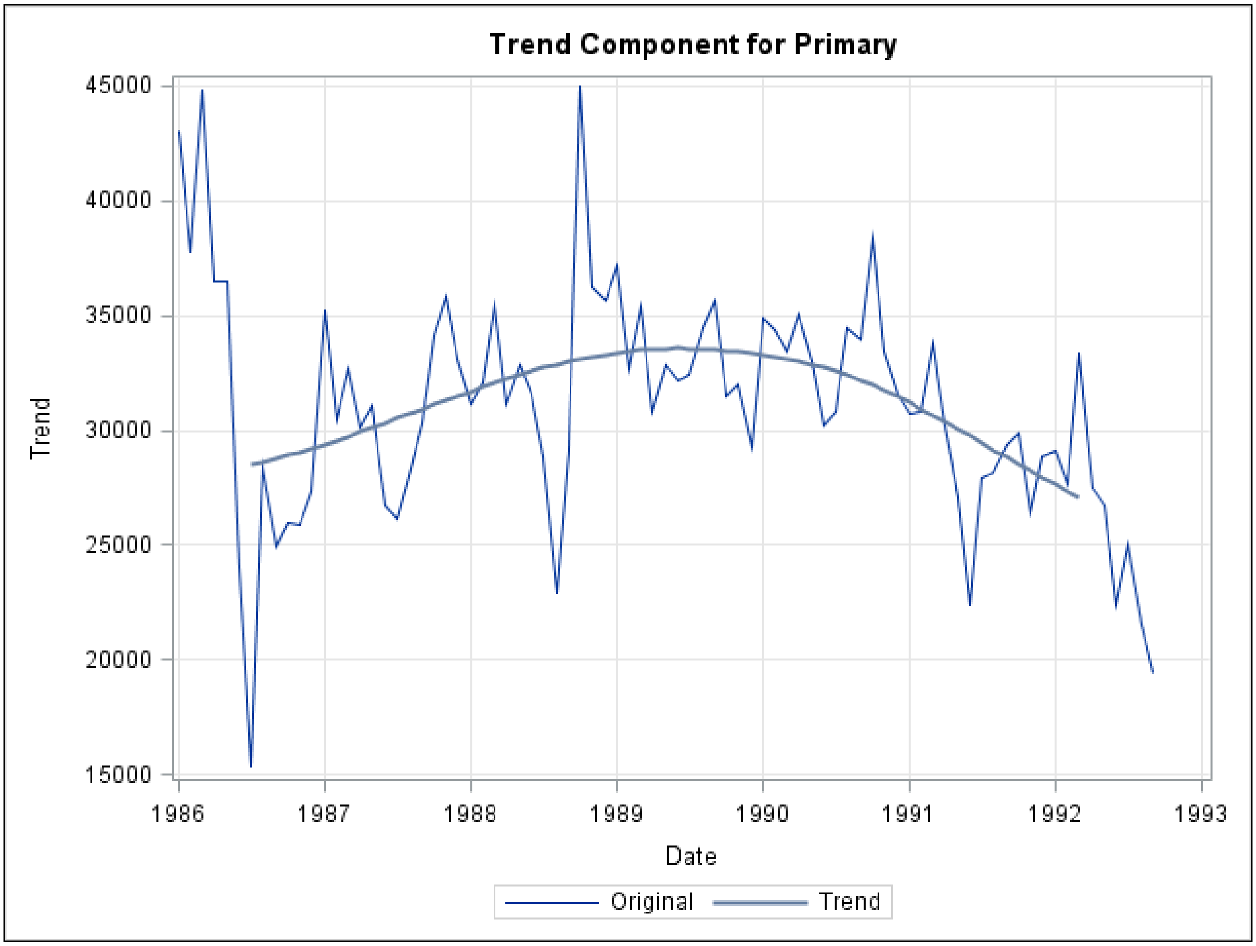
The time series plot follows:



The autocorrelation plots produced by PROC TIMESERIES are almost identical to those produced by PROC ARIMA. One difference is that the correlation panel includes a plot of the Ljung-Box white noise tests.


Correlations for Primary
ACF
PACF
IACF
White Noise Prob
1.0
0.5
0.0
-0.5
-1.0
.001
.05
1.0
0
5
10
15
20
25
Lag

All of the decomposition plots are ignored except for the trend plot that uses a moving average smoother.



This time series is highly variable with some apparent outliers, so the smoothed trend plot helps visualize a quadratic shape.

The IACF and PACF suggest that one autoregressive lag might be sufficient to model the autocorrelations of the data. Notice that no trends were removed so the analysis is only on deviations from the overall mean. This suggests zero augmenting lags in the ADF test but ask for 0, 1, and 2 for illustration. The following code generates the ADF tests:

```
proc arima data=work.MonthLead plots=all;
   identify var=primary stationarity=(ADF=(0 1 2));
quit;
```

The results follow below:

Augmented Dickey-Fuller Unit Root Tests							
Type	**Lags**	**Rho**	**Pr < Rho**	**Tau**	**Pr < Tau**	**F**	**Pr > F**
Zero Mean	0	-1.5260	0.3893	-1.22	0.2025		
	1	-1.0302	0.4652	-1.06	0.2586		
	2	-1.1196	0.4503	-1.29	0.1816		
Single Mean	0	-32.6247	0.0008	-4.48	0.0005	10.27	0.0010
	1	-27.5342	0.0009	-3.50	0.0104	6.35	0.0085
	2	-37.9461	0.0008	-3.96	0.0026	8.25	0.0010
Trend	0	-34.1689	0.0013	-4.58	0.0021	10.51	0.0010
	1	-29.2722	0.0051	-3.61	0.0354	6.53	0.0490
	2	-39.5790	0.0003	-3.98	0.0132	7.94	0.0157

The tests (single mean and trend) agree that the series is stationary, rejecting unit roots for all choices of augmenting lag differences. In that case, differencing produces noninvertible time series, a situation for which the ADF distribution is unknown, the usual outcome of overdifferencing.

Given that the series appears to be stationary based on the ADF, the interpretation of the sample autocorrelation functions leads to consideration of an AR(1) model. Other stationary models are also suggested. Notice that there is an element of suspension of disbelief at this point, because the times series plot clearly suggests that the monthly lead production data is not stationary. Furthermore, because you already examined the annual lead production data and concluded that it is not stationary, it would be surprising to discover that breaking the data into months causes a transition from nonstationarity to stationarity. The goal is to show that diagnostics and statistical tests can be ambiguous and even misleading, and to try to understand the circumstances under which such ambiguities arise. Furthermore, given that uncertainty is almost always present, methods for dealing with the ambiguities are addressed.

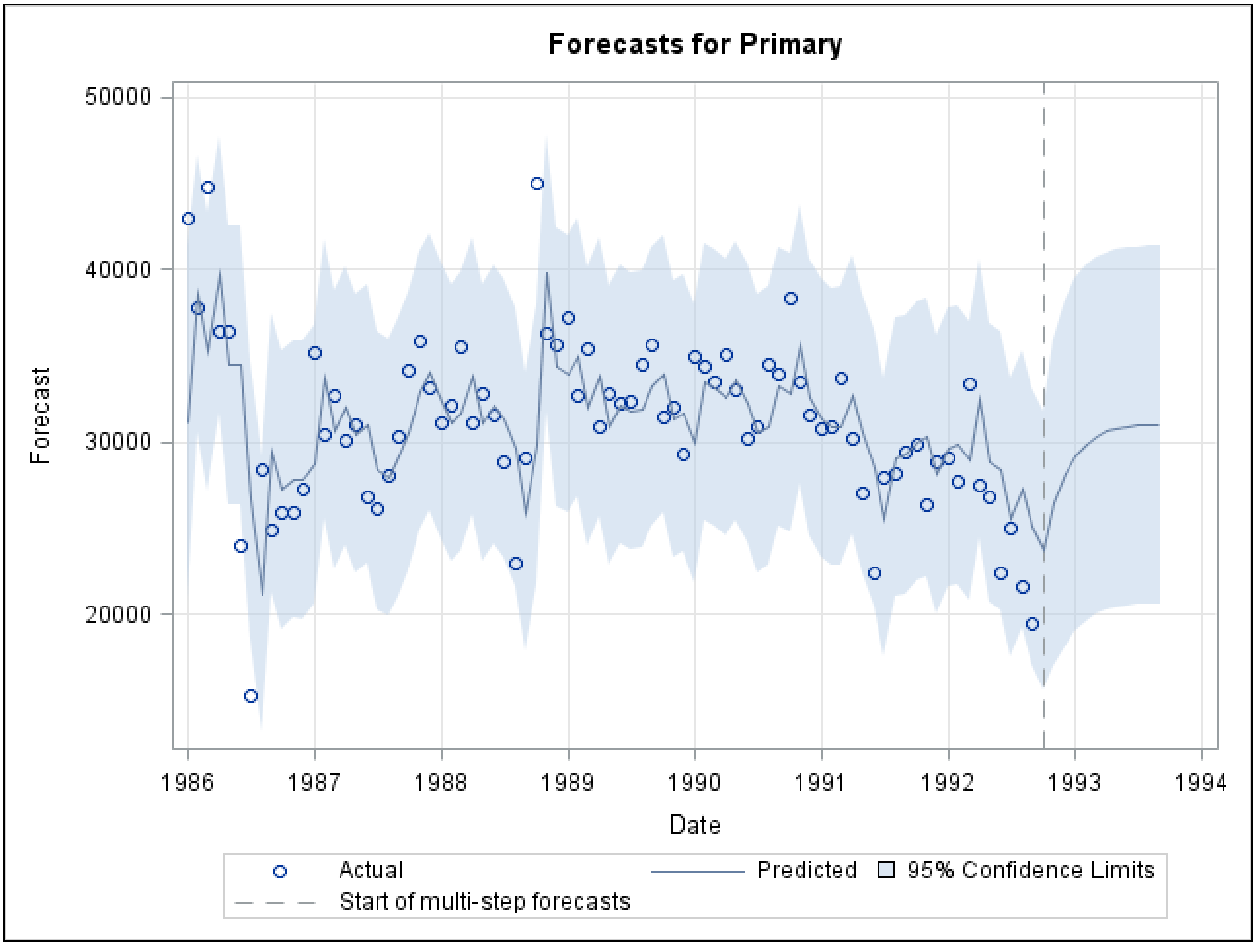
Preliminary to model building, and anticipating possible quadratic components for a model, the data must be extended to allow PROC ARIMA to produce forecasts.

```
data work.MonthLead;
   set sasuser.leadmonth end=eof;
   Time+1;
   TimeSq=Time*Time;
   output;
   if eof then do future=1 to 24;
      Time+1;
      TimeSq=Time*Time;
      Primary=.;
      Date=intnx('month',Date,1);
      output;
   end;
   drop future;
run;
```

The code to fit an AR(1) model follows. The results from the IDENTIFY step are suppressed because PROC TIMESERIES was used to generate the necessary diagnostics above.

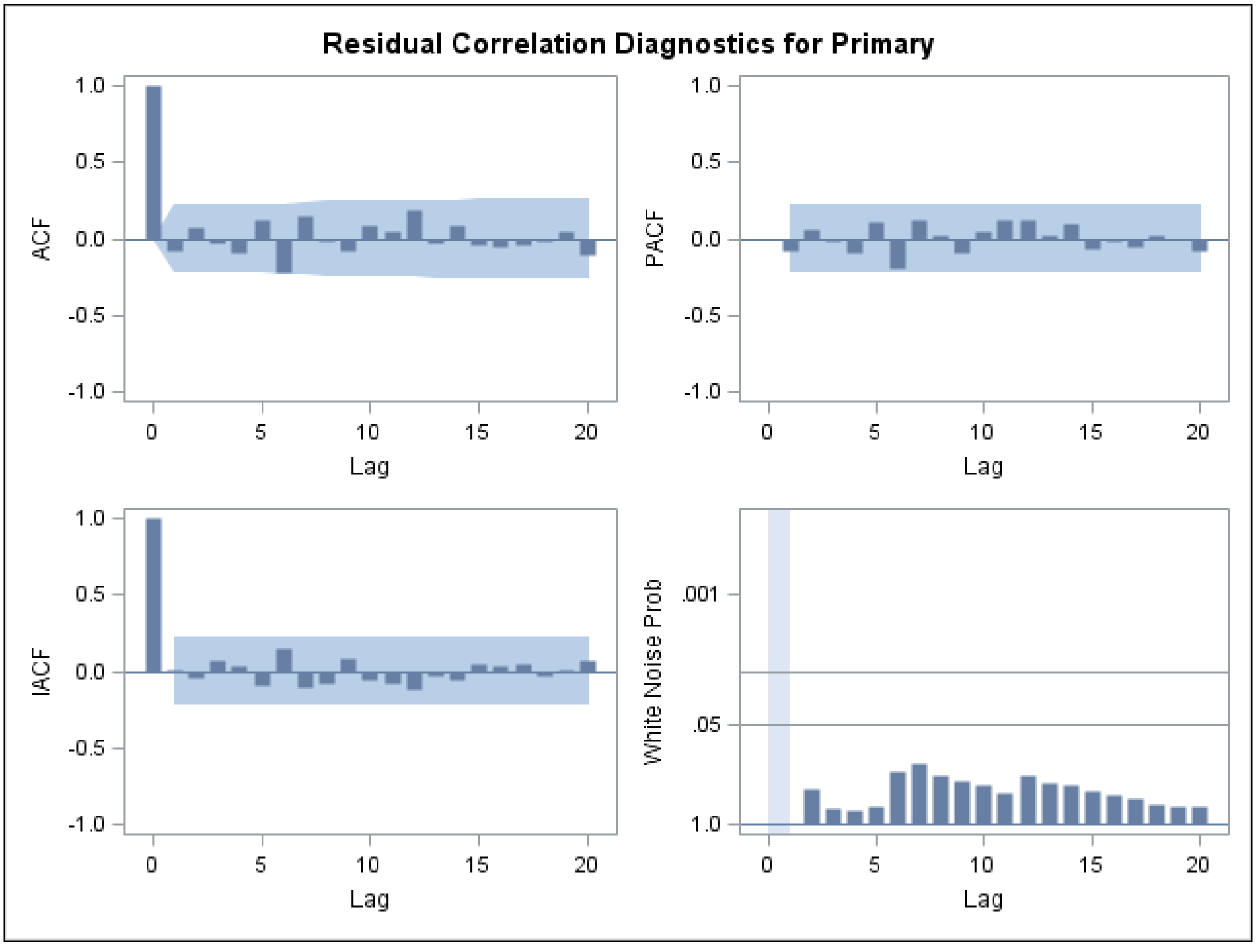
```
proc arima data=work.MonthLead plots=all;
   identify var=primary noprint;
   estimate p=1 ml;
   forecast lead=12 id=date interval=month out=work.ML_AR1;
quit;
```

While the parameter estimates are of interest, you can skip to the forecast plot. The forecasts are saved to a temporary data set so that they can be used later in a comparison.



The forecasts, of course, return to the mean as happens with any stationary model. The forecasts are implausible and cannot be defended. If you need statistical evidence of poor performance, observe that the last six forecasts within the range of the data are above the actual values. Six negative residuals at the end of the data provide evidence that the forecasts are biased too high.

This data set is used in at least one reference to suggest that the monthly lead production data is a good example of real data that is stationary. The reference was not a scientific reference, but rather a part of documentation showing how to use forecasting software. In fact, the diagnostics are convincing. For example, the residual autocorrelation plots suggest that the AR(1) model fits the data well.

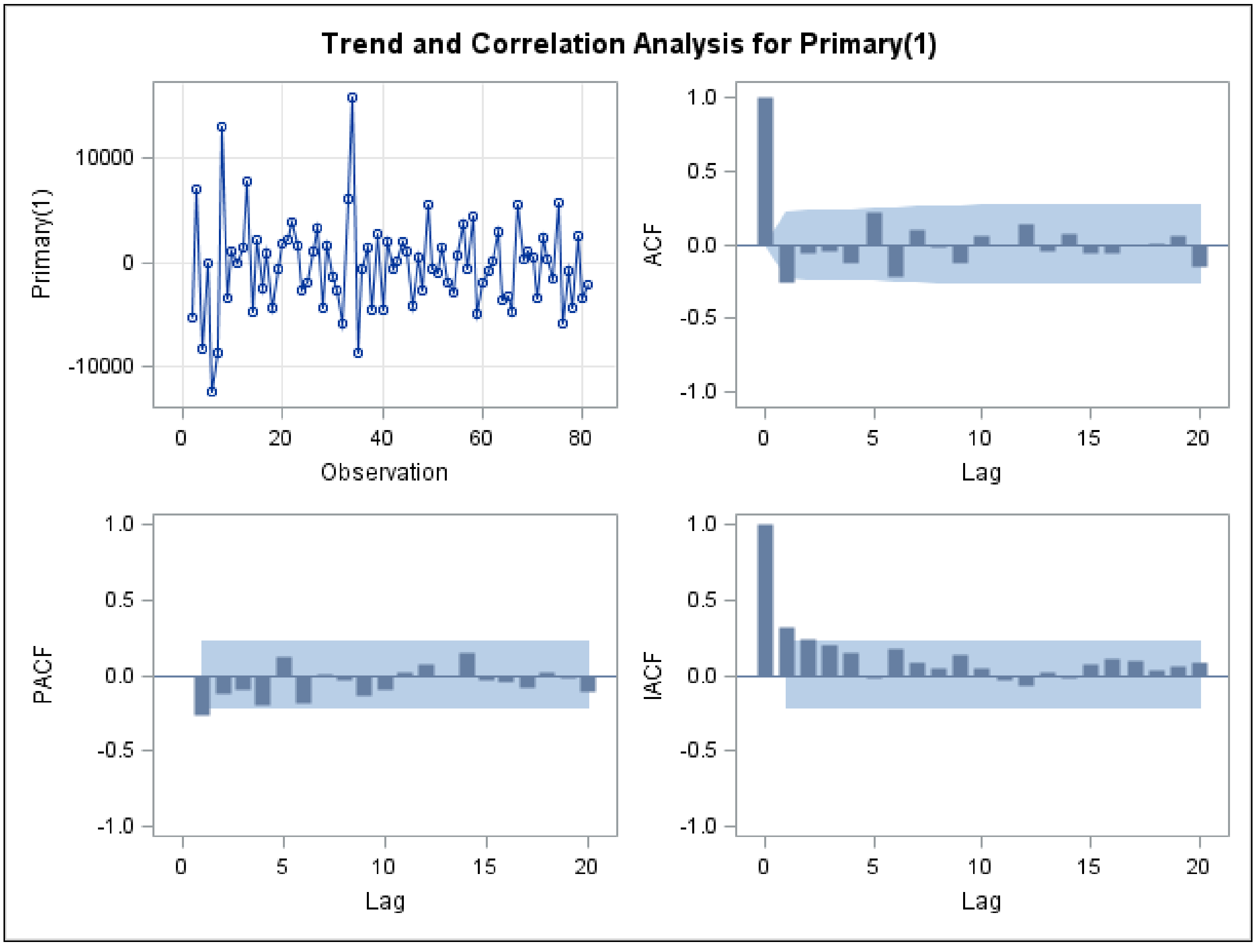


The forecast plot disqualifies the model despite what the other plots might suggest.

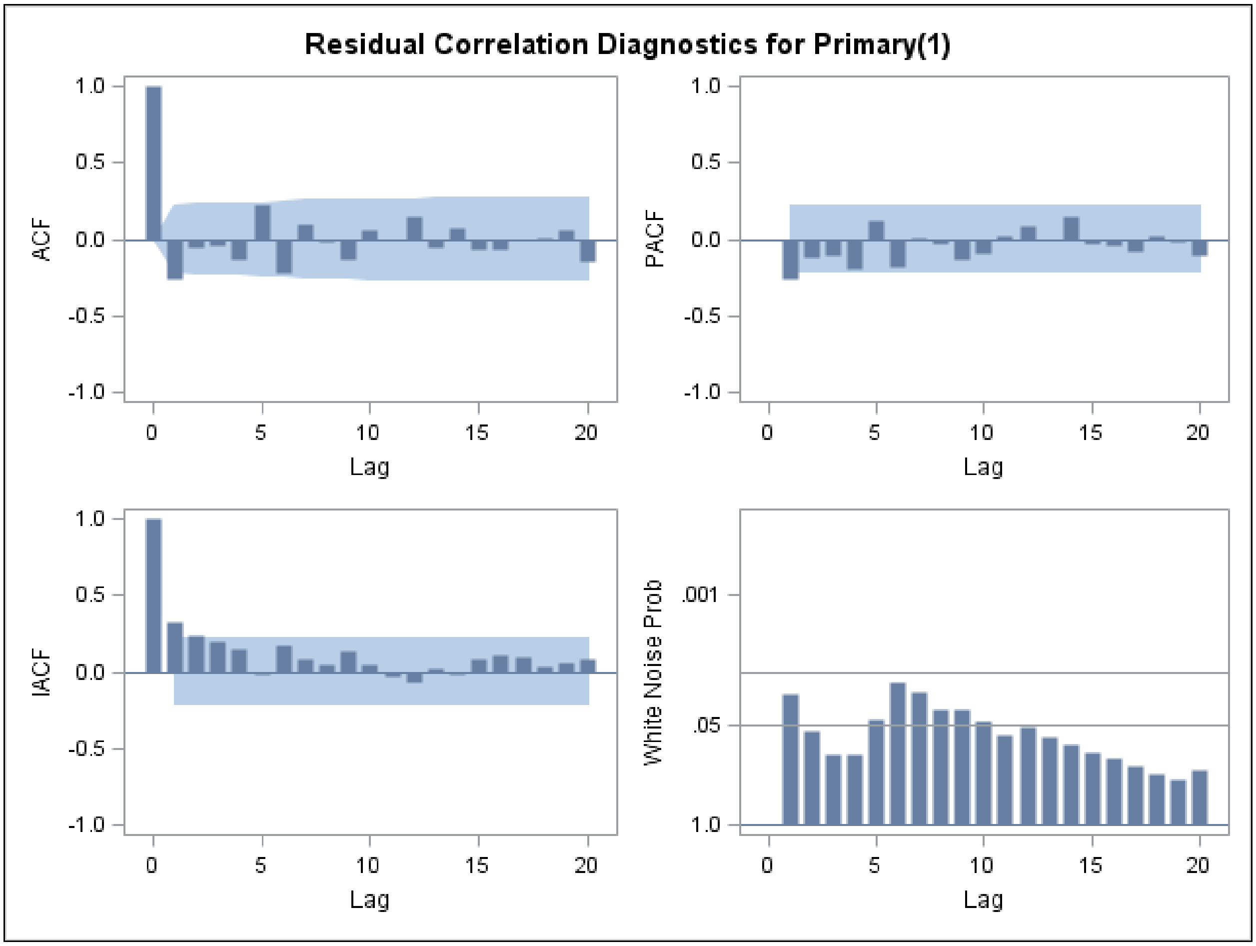
The ADF suggests stationarity, but this seems incorrect, so an obvious nonstationary model to consider is the classic random walk with drift.

```
proc arima data=work.MonthLead plots=all;
   identify var=primary(1);
   estimate ;
run;
```

A first difference specifies the random walk model. An estimate statement with no options suggests a constant only model for the first differences, which is the correct specification for a random walk with drift. A pure random walk model would require the NOCONSTANT option in the ESTIMATE statement. The sample autocorrelations follow, first for the differenced data, and then for the random walk with drift model.


Trend and Correlation Analysis for Primary(1)
Primary(1)
10000
0
-10000
0 20 40 60 80
Observation
ACF
1.0 0.5 0.0 -0.5 -1.0
0 5 10 15 20
Lag
PACF
1.0 0.5 0.0 -0.5 -1.0
0 5 10 15 20
Lag
IACF
1.0 0.5 0.0 -0.5 -1.0
0 5 10 15 20
Lag

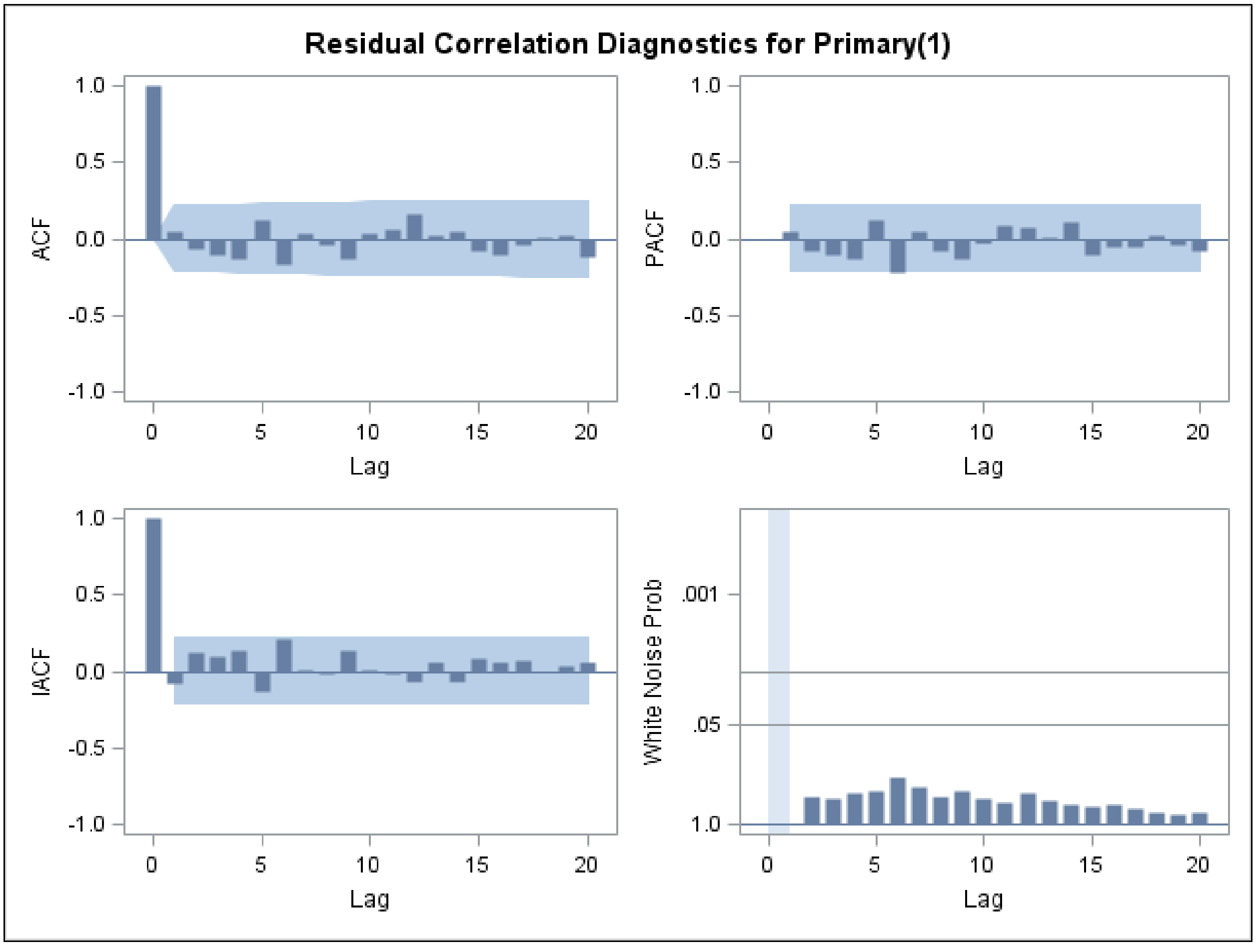
Adding the constant for the trend produces the following residual plots:



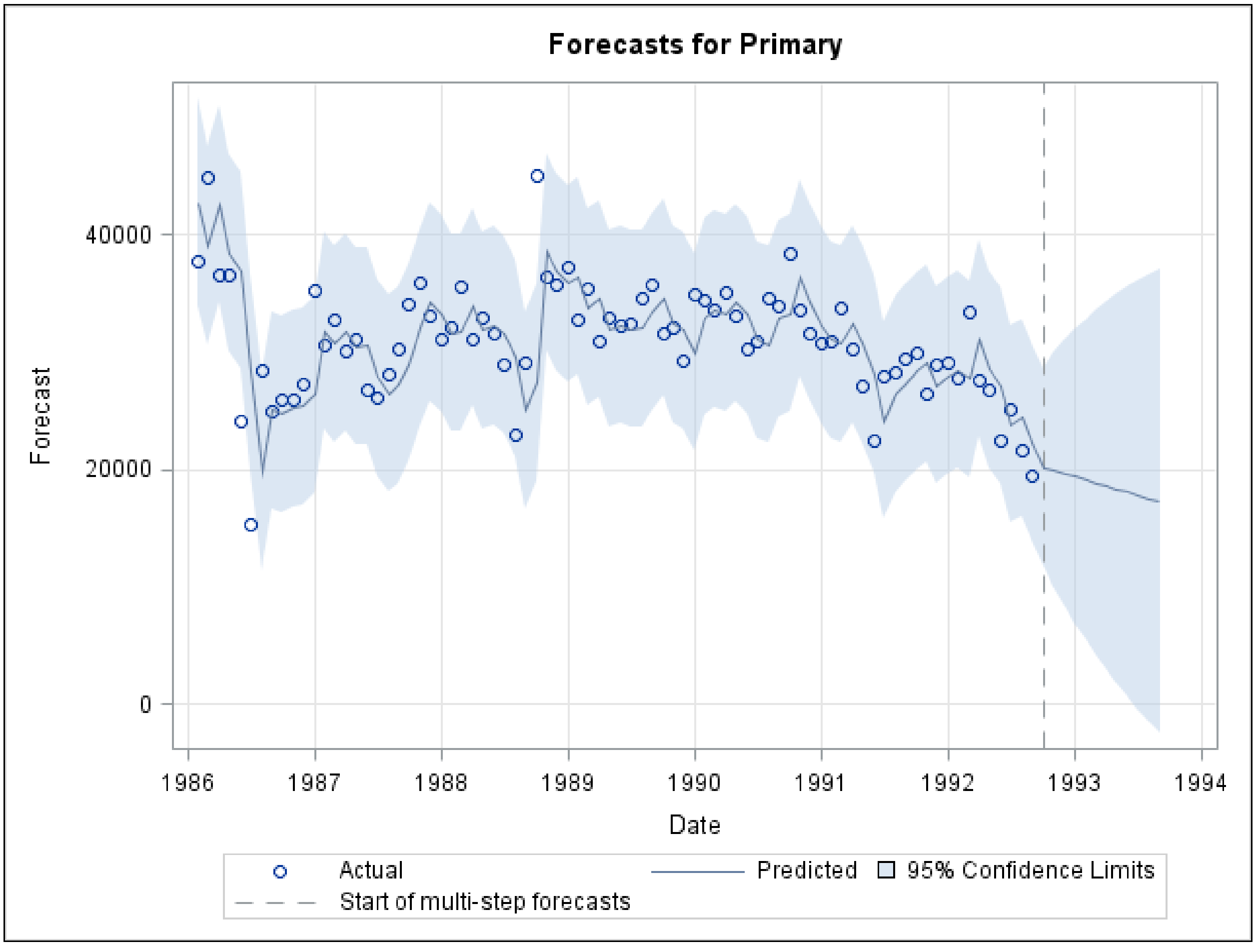
The plots suggest several nonstationary error models, but only an MA(1) is tried.

```
   estimate q=1 ml;
   forecast lead=12 id=date interval=month out=work.ML_RWD;
quit;
```

The fit diagnostics suggest a good fit for the ARIMA(0,1,1) model, also referred to as an IMA(1,1) model.


Residual Correlation Diagnostics for Primary(1)
ACF
PACF
IACF
White Noise Prob
Lag
1.0
0.5
0.0
-0.5
-1.0
.001
.05
0
5
10
15
20

The forecast plot follows:



The forecasts look reasonable and are consistent with the downward slope at the end of the historic data, although perhaps the slope of the forecasts is too shallow.

The ARIMA(0,1,1) model cannot be disqualified, but it still raises concerns, and the prediction confidence intervals are very wide. If you revisit the smoothed trend plot from PROC TIMESERIES, an obvious alternative is a quadratic trend model. The ADF has no test to suggest such a model. However, you can examine the residual plots from the fit of a quadratic trend curve to see whether any obvious nonstationary trend remains.

The data set was prepared for fitting a quadratic trend. The code to do so follows:

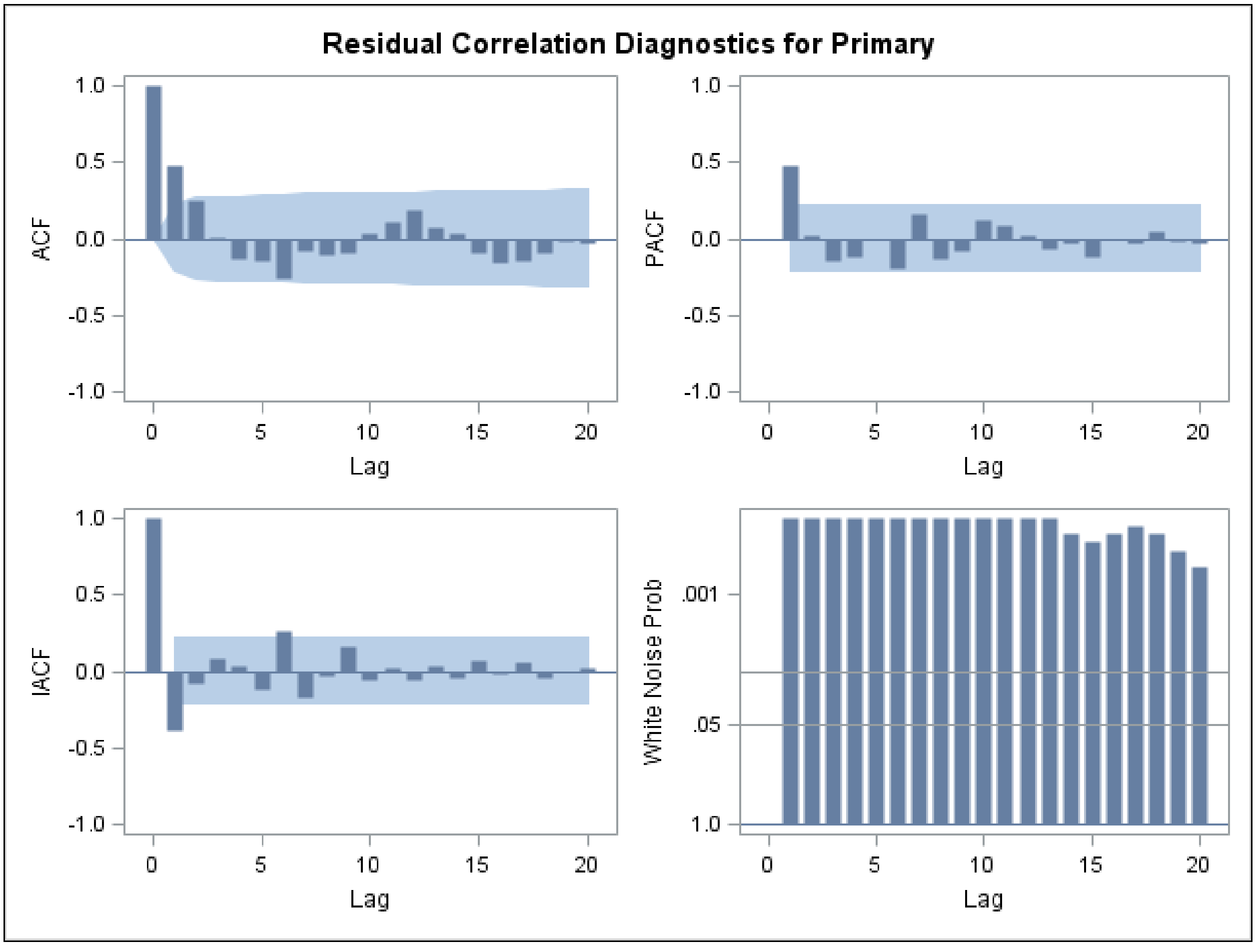
```
proc arima data=work.MonthLead plots=all;
   identify var=primary crosscor=(Time TimeSq) noprint;
   estimate input=(Time TimeSq) ml;
run;
```

The table of estimates suggests that the linear (**Time**) and quadratic (**TimeSq**) terms are valid.

Maximum Likelihood Estimation							
Parameter	Estimate	Standard Error	t Value	Approx Pr > |t|	Lag	Variable	Shift
MU	29575.6	1607.4	18.40	<.0001	0	Primary	0
NUM1	213.40470	90.47343	2.36	0.0183	0	Time	0
NUM2	-3.26730	1.06913	-3.06	0.0022	0	TimeSq	0

Constant Estimate	29575.65
Variance Estimate	22124932
Std Error Estimate	4703.715
AIC	1602.701
SBC	1609.884
Number of Residuals	81

The residual plots imply that the model is not adequate. There is still autocorrelation present in the residuals. You could perform an ADF test to see whether the residuals appear to be stationary.



The plots suggest several stationary models. An AR(1) seems to be the best choice based on the spike at lag 1 for the PACF and IACF, but the ACF bars do not necessarily have the expected exponential decay arising from an AR(1) model.

```
   estimate input=(Time TimeSq) p=1 ml;
   forecast lead=12 id=date interval=month out=work.ML_Quad;
quit;
```

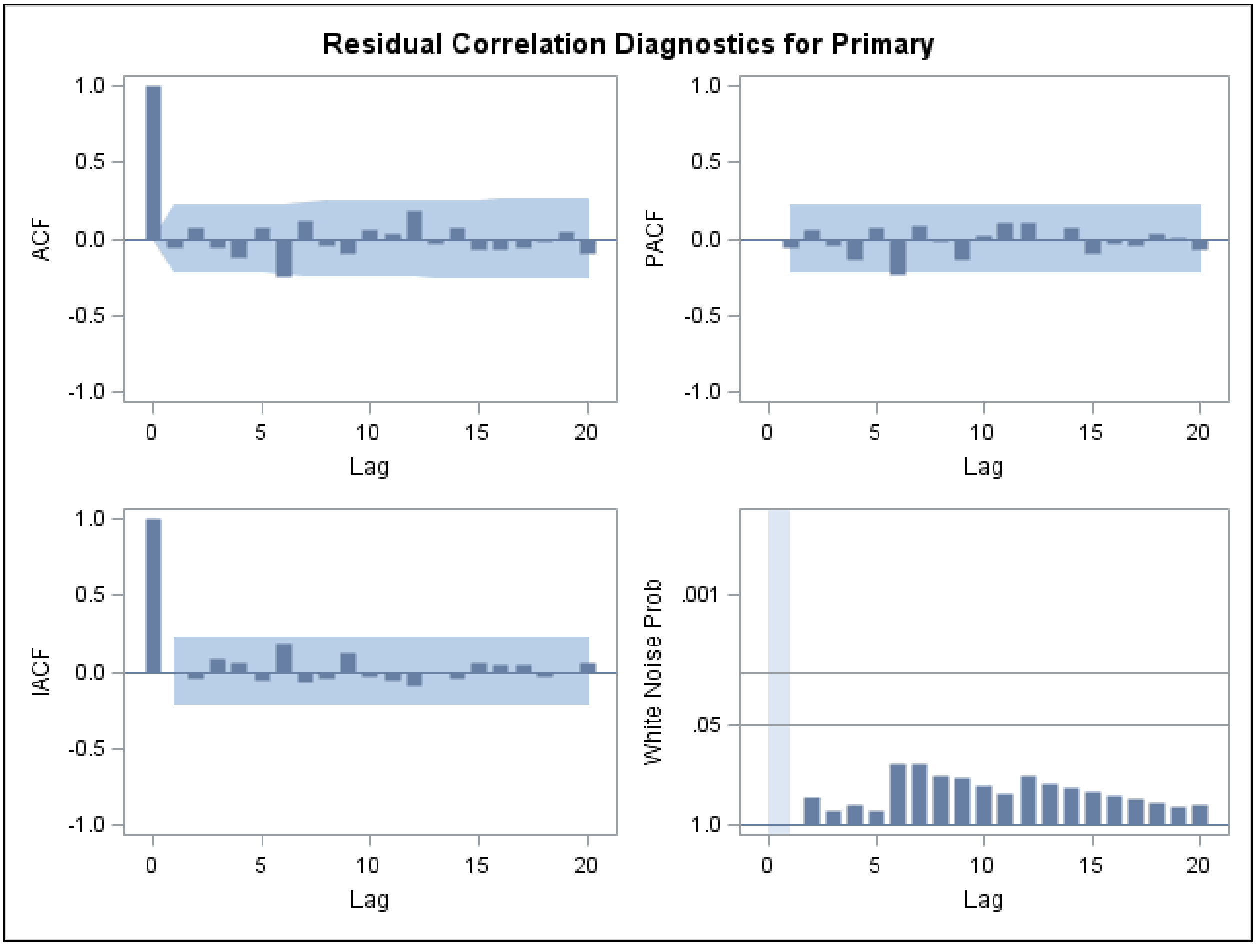
The table of estimates follows:

Maximum Likelihood Estimation							
Parameter	Estimate	Standard Error	t Value	Approx Pr > |t|	Lag	Variable	Shift
MU	31040.7	2850.5	10.89	<.0001	0	Primary	0
AR1,1	0.54937	0.09634	5.70	<.0001	1	Primary	0
NUM1	154.26172	160.76875	0.96	0.3373	0	Time	0
NUM2	-2.79561	1.89888	-1.47	0.1410	0	TimeSq	0

Constant Estimate	13987.88
Variance Estimate	16365696
Std Error Estimate	4045.454
AIC	1579.592
SBC	1589.169
Number of Residuals	81

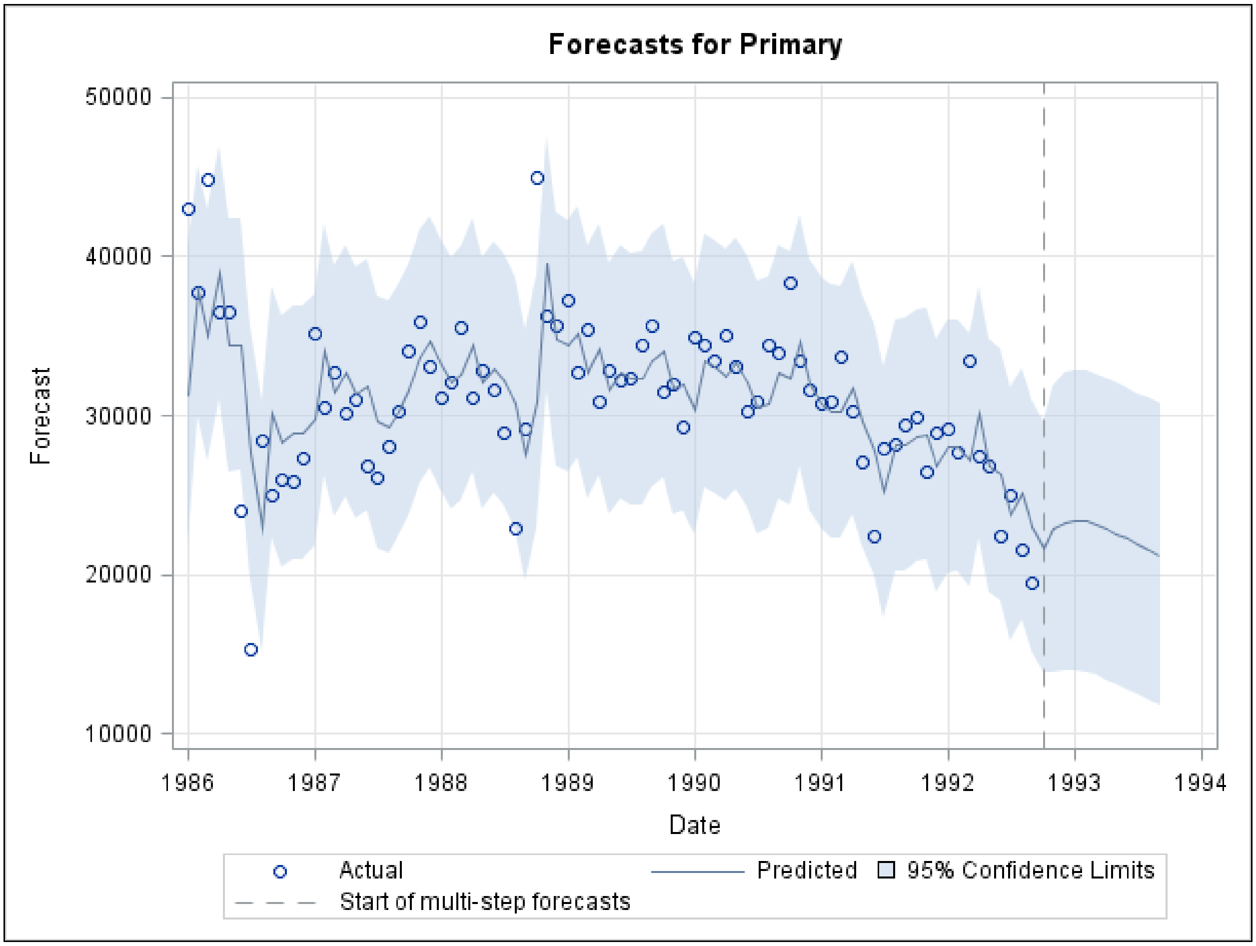
The addition of the AR(1) parameter seems to cause the linear and quadratic terms to lose statistical significance. The model is used nonetheless.

Here are the residual autocorrelation plots:



The residuals pass a white noise test.

The forecasts appear below:



The forecasts are disappointing. With any deterministic trend, the stationary error model generates forecasts that eventually converge to the trend mean, in this case, a quadratic trend. Unfortunately, the quadratic trend does not appear to be flexible enough to address the steeper downturn at the end of the historic data. You would have difficulty defending the positive bump in production for the next few months if there is no external economic or business indicator. At least the forecasts show a long term decline consistent with the end of the data.

The forecasts are required, so you have to decide what forecasts you will supply to management. You could examine goodness-of-fit statistics. First, notice that use of a holdout sample is ***not*** recommended because of the downturn in the data at the end of the series. Thus, AIC or SBC are good candidates for use in model comparison.

Model	AIC	SBC
AR(1)	1580.5	1585.3
Quadratic+AR(1)	1579.6	1589.2
ARIMA(0,1,1)	1566.8	1571.6

You should never blindly choose a model because it has the best goodness-of-fit value. However, for these three models, the ARIMA(0,1,1) produces the best AIC and SBC statistics.

An overlay plot can help you compare the forecasts. You must combine all three forecast data sets.

```
data work.forecasts;
merge work.ML_AR1 (rename=(forecast=F_AR1  U95=U95_AR1  L95=L95_AR1))
      work.ML_RWD (rename=(forecast=F_RWD  U95=U95_RWD  L95=L95_RWD))
      work.ML_Quad(rename=(forecast=F_Quad U95=U95_Quad L95=L95_Quad));
   by date;
   if (Date>"01sep1992"d) then output;
run;
```

Notice that the future forecasts are retained. You need to combine these forecasts with the original data.

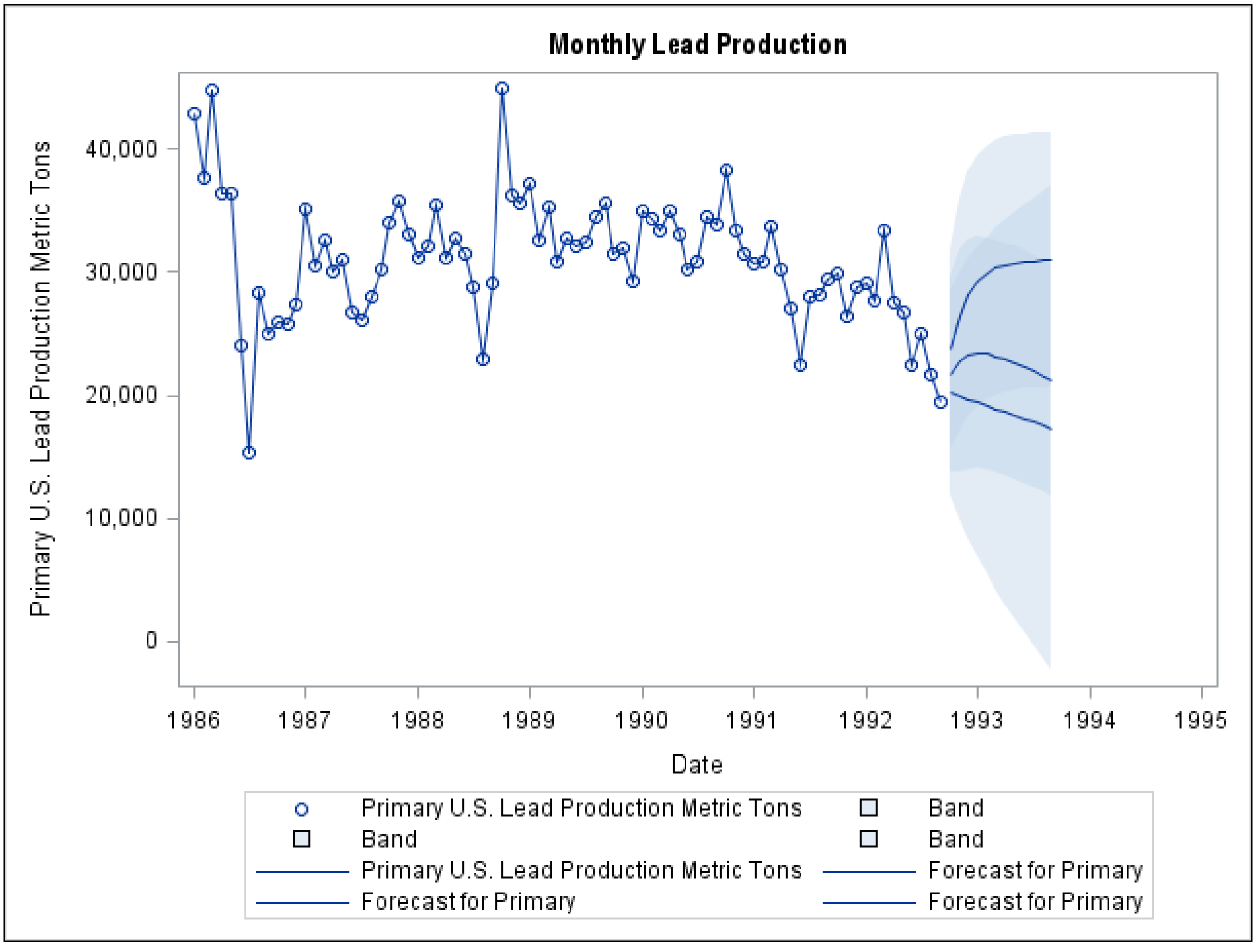
```
data work.all;
   merge work.MonthLead work.forecasts;
   by Date;
run;
```

The combined data can be used to generate overlay plots.

```
proc sgplot data=work.all;
   scatter x=Date y=Primary;
   band    x=Date upper=U95_AR1  lower=L95_AR1   / transparency=0.6;
   band    x=Date upper=U95_RWD  lower=L95_RWD   / transparency=0.6;
   band    x=Date upper=U95_Tuba lower=L95_Drums / transparency=0.6;
   band    x=Date upper=U95_Quad lower=L95_Quad  / transparency=0.6;
   series  x=Date y=Primary;
   series  x=Date y=F_AR1;
   series  x=Date y=F_RWD;
   series  x=Date y=F_Quad;
run;
```

The TRANSPARENCY option makes it easier to see the different confidence bands.

The plot appears below:



The legend can be customized along with the line choices to help distinguish the forecasts, but for this example, such embellishments are not needed. The top forecast line comes from the stationary AR(1) model, the middle forecast line from the quadratic trend model, and the bottom forecast line comes form the ARIMA(0,1,1) model.

Overlaying the forecasts and confidence bands gives a nice comparison. As might be expected, the degree of differencing and the nature of deterministic predictor functions plays the major role in determining forecasts more than a few steps into the future while the nature of the ARMA error term hold less importance, though it does have an impact on the hypothesis tests. You get to choose which forecasts to use. Perhaps the ARIMA(0,1,1) would be the easiest to defend.

Another option is to take the best forecasts, and rather than pick one set of forecasts, use forecast averaging to arrive at a unique set of forecasts. For example, the final forecasts could be the average of the quadratic trend forecasts and the ARIMA(0,1,1) forecasts. Another approach is to use “optimal weighting.” For example, you could take the forecasts of the top two models as input variables to a regression model with the actual time series as the target variable. The least squares coefficients define the weights for a weighted average.

Outlier Detection

This demonstration illustrates how to use PROC ARIMA to detect outliers and how to use outlier indicators to improve forecasts.

The code for this demonstration can be found in **Demo3_04Outliers.sas**.

It is typically possible to get a good fitting model with enough effort. Skepticism of the results, proportional to the effort required, is typically justified.

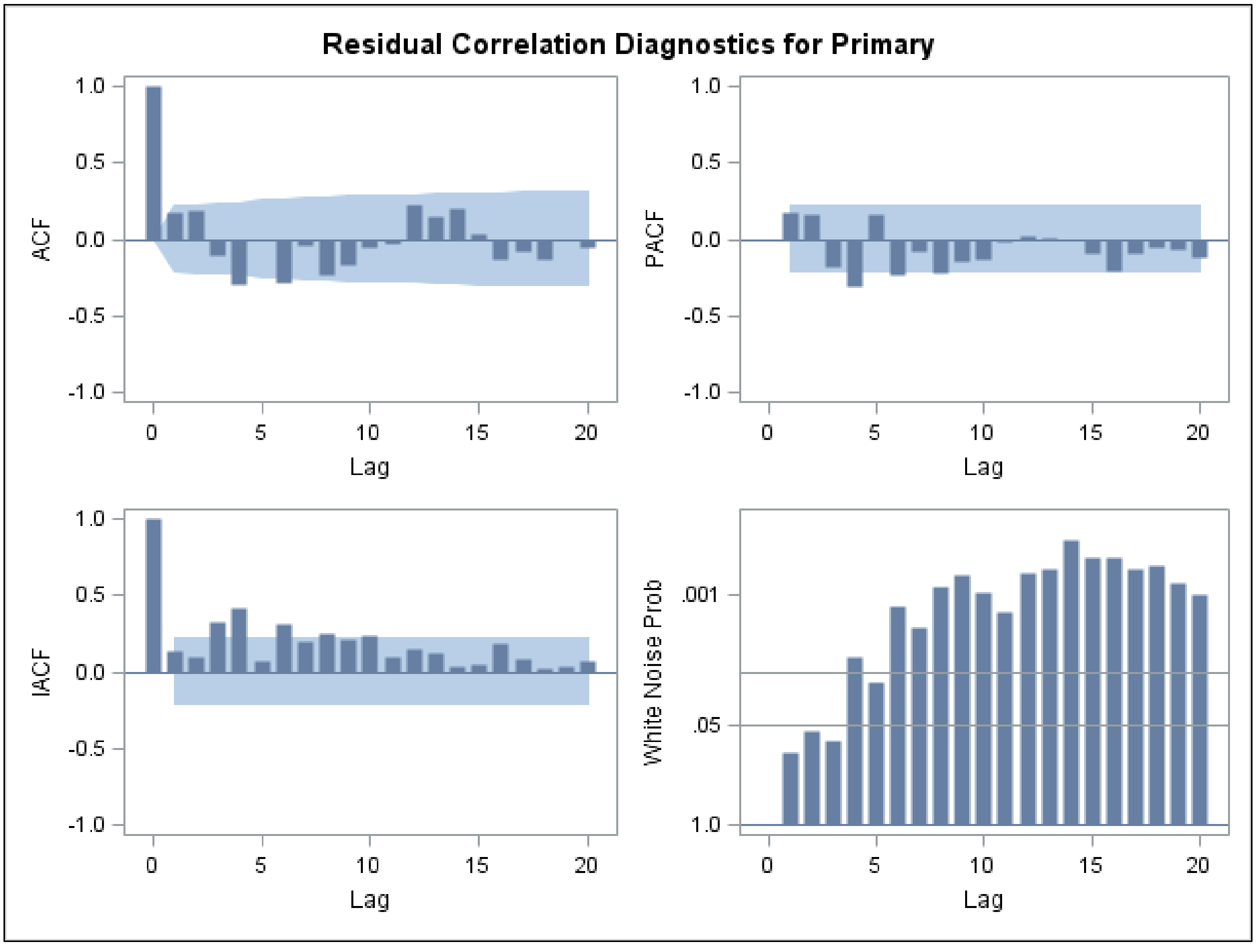
Notice the initial five large values, previously ignored, in the graph of the primary lead production time series. Use of any test in a misspecified model can be misleading. This previously ignored level shift is one possible explanation for the initial conclusion of stationarity. A variable **X** that is `1` for those variables and `0` for the others was added to the data to account for the level shift. A variable **X2** that is `-1` in August of 1988 to model the drop that month and `1` two months later to model the almost equal magnitude jump that month and `0` elsewhere was added to the data as well. The code to generate the modeling data follows:

```
data work.MonthLead;
   set sasuser.leadmonth end=eof;
   Time+1;
   TimeSq=Time*Time;
   X=(Date<"01jun1986"d);
   X2=(Date="01oct1988"d)-(Date="01Aug1988"d); /* =0 1 or -1 */
   output;
   if eof then do future=1 to 24;
      Time+1;
      TimeSq=Time*Time;
      Primary=.;
      Date=intnx('month',Date,1);
      output;
   end;
   drop future;
run;
```

These variables had a rather large effect on the fit but correlation remained in the residuals. The following code generates the residual autocorrelation plots:

```
proc arima data=work.MonthLead plots=all;
   identify var=primary crosscor=(Time TimeSq X X2) noprint;
   estimate input=(Time TimeSq X X2) ml;
quit;
```

The plots follow:



Interpreting the plots is challenging. For help, go to PROC AUTOREG. The BACKSTEP option in the MODEL statement implements backward elimination variable selection. You specify a maximum AR order using the NLAGS= option, and then backward elimination is used to pick a subset of the orders.

```
proc autoreg data=work.MonthLead;
   model Primary=Time TimeSq X X2/nlag=13 backstep;
run;
```

The BACKSTEP option suggests the use of an autoregressive error model with lags at 4 and 8. The following code fits the appropriate model:

```
proc arima data=work.MonthLead plots=all;
   identify var=Primary crosscorr=(Time TimeSq X X2) noprint;
   estimate input=(Time TimeSq X X2) p=(4 8) ml plot;
   forecast lead=12 id=Date interval=month out=work.ML_Quad_X;
quit;
```

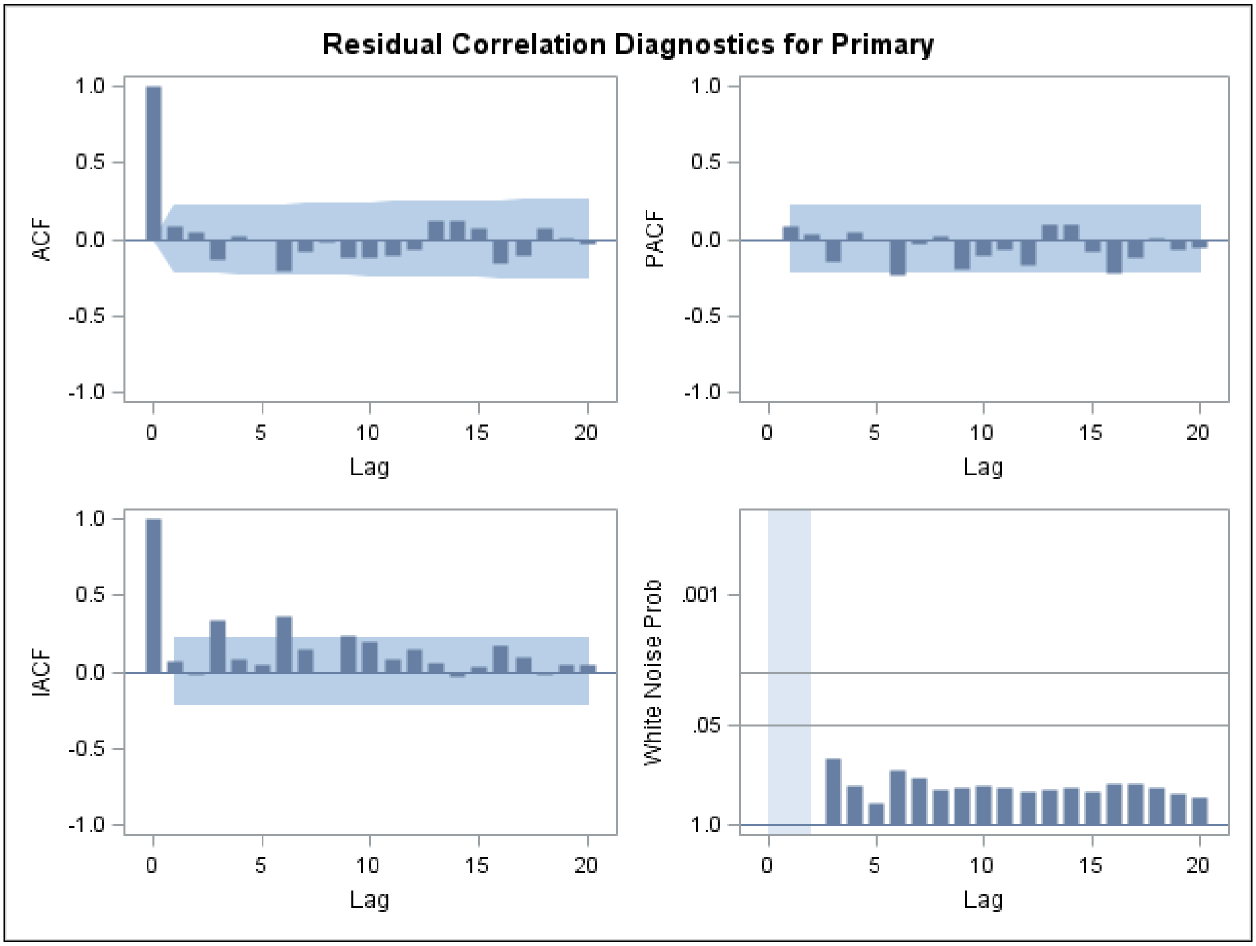
The estimates table follows:

Maximum Likelihood Estimation							
Parameter	**Estimate**	**Standard Error**	**t Value**	**Approx Pr > |t|**	**Lag**	**Variable**	**Shift**
MU	21814.1	759.03090	28.74	<.0001	0	Primary	0
AR1,1	-0.44143	0.11012	-4.01	<.0001	4	Primary	0
AR1,2	-0.44297	0.10882	-4.07	<.0001	8	Primary	0
NUM1	571.58483	39.56893	14.45	<.0001	0	Time	0
NUM2	-6.79987	0.44599	-15.25	<.0001	0	TimeSq	0
NUM3	15545.1	1433.8	10.84	<.0001	0	X	0
NUM4	10184.8	1562.4	6.52	<.0001	0	X2	0

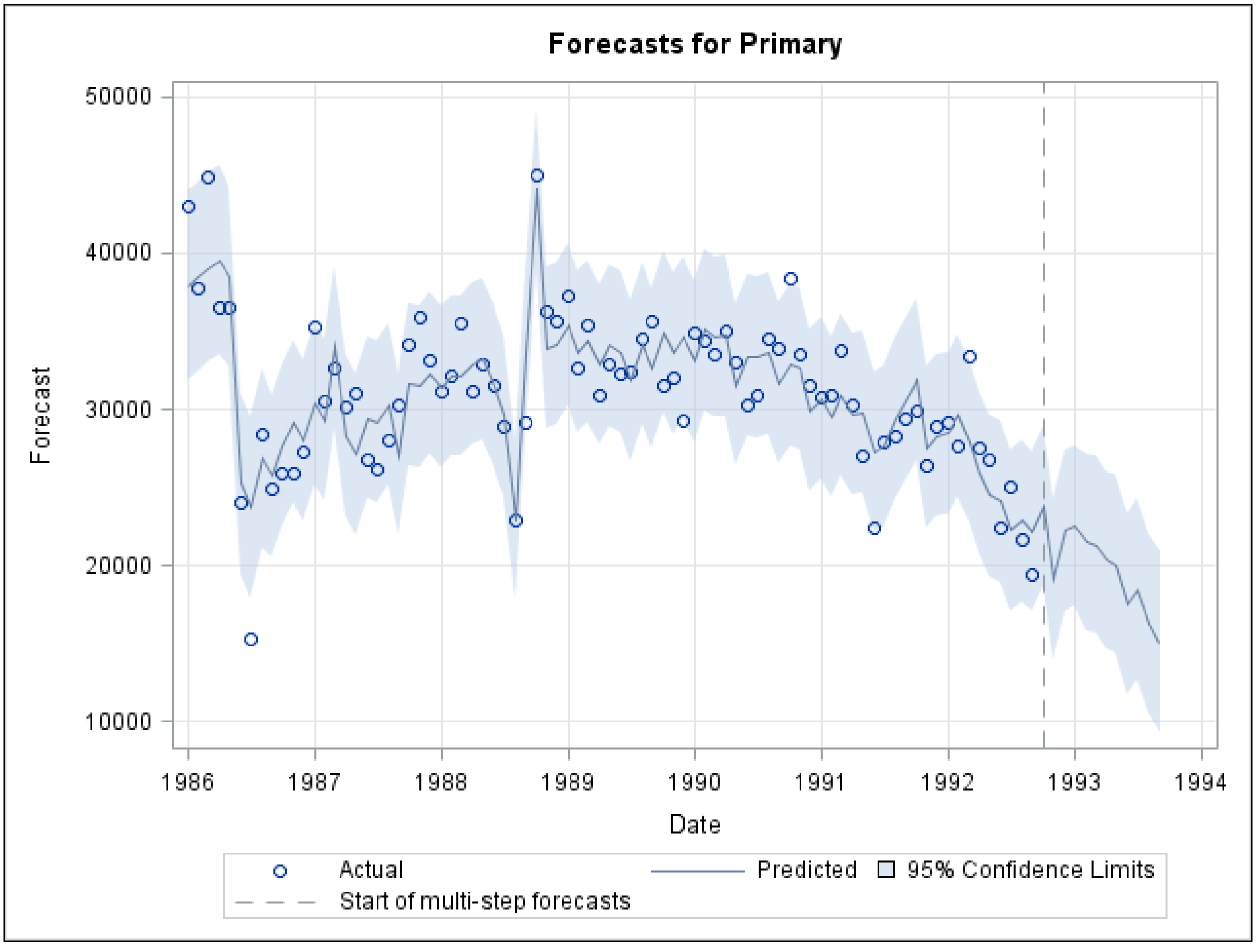
Constant Estimate	41106.75
Variance Estimate	6963620
Std Error Estimate	2638.867
AIC	1514.94
SBC	1531.702
Number of Residuals	81

The addition of the shift variables and the rather unusual AR component caused the estimates for the quadratic to become significant.

The residual analysis plots follow:



The forecasts appear reasonable. Furthermore, the above table reveals that AIC=1514.9 and SBC=1531.7, which are the best results that you have seen so far.



The upward bump for the few months into the future looks more plausible, but might be difficult to explain. Especially difficult might be explaining the memory of four and eight months for the AR component.

The downside of this is that the added variables **X** and **X2** were created, after looking at the data, to take out points that visually did not fit with the quadratic. This is a weak justification, and this sort of tweaking of a model to get a better fit is often the cause of overfitting, which results in poor extrapolated forecasts. Another concern is that the error term skips over some lags while retaining lags that were not justified as being reasonable. If there were evidence of some intervention, for example, the closing of a major mining company or a law eliminating lead from some major product such as gasoline or paint, the use of the level shift variable would be on a much firmer foundation. Nevertheless, the forecasts look reasonable and have some interesting features associated with the nontrivial lag structure. The width of the forecast intervals suggests that the nuances in the forecast be taken with the proverbial grain of salt. The use of **X** and **X2** clearly has major consequences for the fit. The left side is lifted by the coefficient 15545, and the drop and then increase of 10184 account for the capture of the associated two data points immediately left of the plot’s center. The variables **X** and **X2** essentially remove the influence of these sets of points from the overall fit and the forecast.

When spikes at lag 3 and 6 (close to significant) are eliminated by added moving average terms, namely specifying P=(4,8) Q=(3,6), the further improvement is considered worthwhile by AIC and SBC which then become AIC=1502.6 and SBC=1524.1.

You are in perilous territory when you so aggressively pursue goodness-of-fit at the possible expense of forecast accuracy. However, for the monthly lead production data, there are clearly legitimate concerns about influential observations. These were addressed by visual inspection of the data and some creativity. PROC ARIMA provides a feature that can supply more objectivity to this endeavor. The OUTLIER statement identifies potentially influential observations. The following code looks for outliers in a model with quadratic trend:

```
proc arima data=work.MonthLead;
   identify var=Primary crosscorr=(Time TimeSq);
   estimate input=(Time TimeSq) ml;
   outlier type=(ao ls tc(5)) maxnum=10 id=Date;
quit;
```

The option TC(5) looks for temporary level shifts in the data lasting for five months. Notice that MAXNUM=10 produces at most 10 outliers. There might not be 10 outliers in the data. The lead production data has a surplus of outliers. The following 10 results are found:

Outlier Details					
Obs	**Time ID**	**Type**	**Estimate**	**Chi-Square**	**Approx Prob>ChiSq**
1	**JAN1986**	Temp(5)	9518.5	33.96	<.0001
6	**JUN1986**	Temp(5)	-7335.6	22.59	<.0001
34	**OCT1988**	Additive	11941.6	15.50	<.0001
32	**AUG1988**	Additive	-10106.9	12.78	0.0004
17	**MAY1987**	Temp(5)	-3994.9	10.24	0.0014
7	**JUL1986**	Additive	-8242.7	9.11	0.0025
56	**AUG1990**	Temp(5)	3408.8	8.36	0.0038
78	**JUN1992**	Shift	-3774.6	9.11	0.0025
66	**JUN1991**	Additive	-7021.0	8.66	0.0033
75	**MAR1992**	Additive	6181.6	7.51	0.0061

The top outlier corresponds to the variable **X**, which represents a temporary level shift for the first five months in the data. The third and fourth outliers correspond to the variable **X2**. You can include them as separate indicator variables, which enables the estimation of different magnitudes for the outlier effect.

The following code illustrates how you could define additional input variables to correspond to all of the outliers identified above:

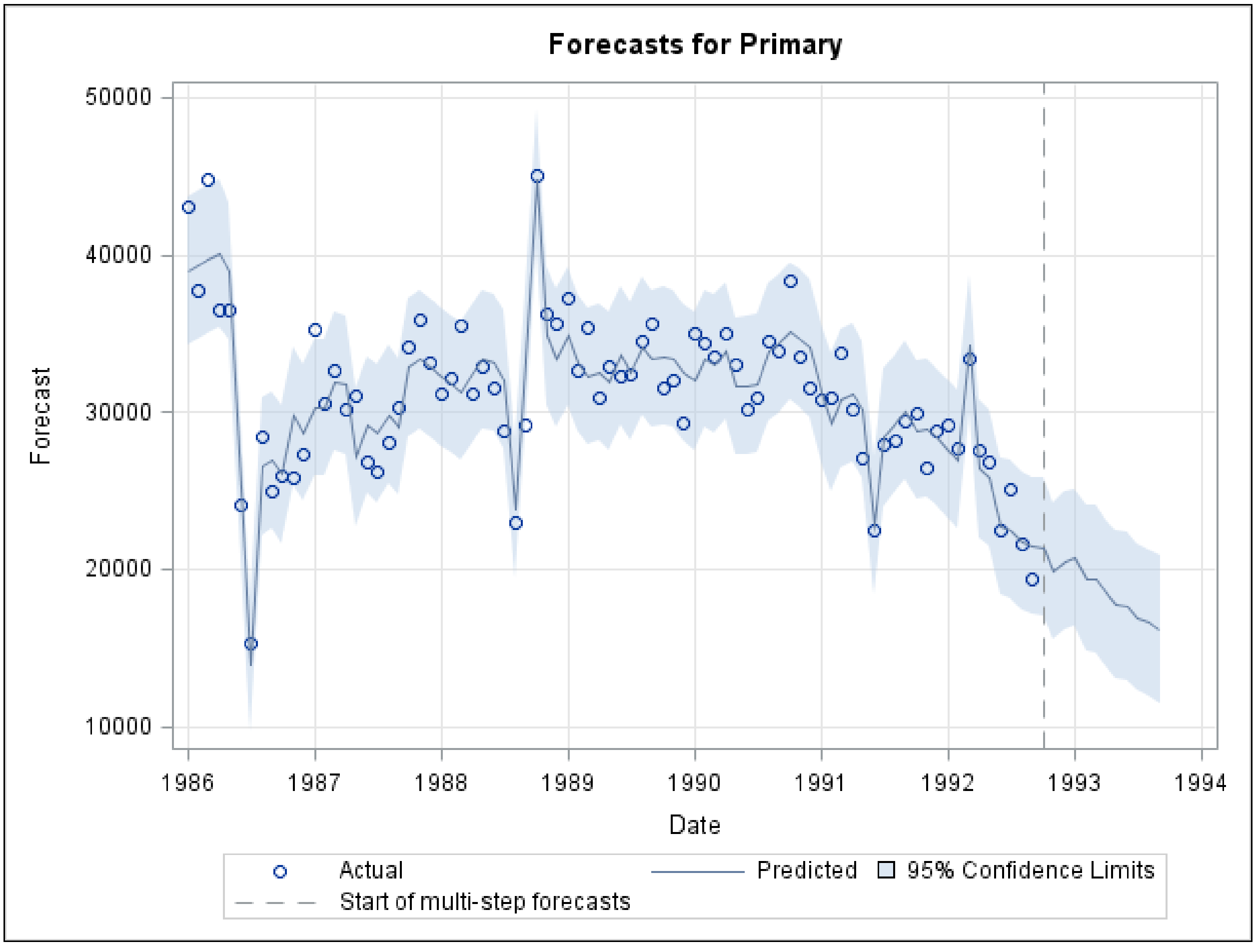
```
data work.monthlead;
   set work.monthlead;
   TC5_JAN1986=('01JAN1986'd<=Date<='31MAY1986'd);
   TC5_JUN1986=('01JUN1986'd<=Date<='31OCT1986'd);
   AO_OCT1988 =('01OCT1988'd<=Date<='31OCT1988'd);
   AO_AUG1988 =('01AUG1988'd<=Date<='31AUG1988'd);
   TC5_MAY1987=('01MAY1987'd<=Date<='30SEP1987'd);
   AO_JUL1986 =('01JUL1986'd<=Date<='31JUL1986'd);
   TC5_AUG1990=('01AUG1990'd<=Date<='31DEC1990'd);
   LS_JUN1992 =(Date>='01JUN1992'd);
   AO_JUN1991 =('01JUN1991'd<=Date<='30JUN1991'd);
   AO_MAR1992 =('01MAR1992'd<=Date<='31MAR1992'd);
   drop X X2;
run;
```

The variables **X** and **X2** are dropped because they were replaced. The following code fits a model that attempts to account for all of the outliers:

```
proc arima data=work.monthlead plots=all;
   identify var=Primary
            crosscorr=(Time TimeSq TC5_JAN1986
            TC5_JUN1986 AO_OCT1988 AO_AUG1988
            TC5_MAY1987 AO_JUL1986 TC5_AUG1990
            LS_JUN1992 AO_JUN1991 AO_MAR1992) noprint;
   estimate p=(4) input=(Time TimeSq TC5_JAN1986
            TC5_JUN1986 AO_OCT1988 AO_AUG1988
            TC5_MAY1987 AO_JUL1986 TC5_AUG1990
            LS_JUN1992 AO_JUN1991 AO_MAR1992) ml;
   forecast lead=0 id=Date interval=month printall;
run;
```

The mysterious **P=(4)** appears for the new model based on examination of the residuals from a model fit with no ARMA error structure.

The forecast plot follows:



The forecasts look reasonable. However, the same concerns about overfitting arise, and the fact that an OUTLIER statement suggested the new indicator variables does not make them any more legitimate than if they were discovered by visual inspection.

3.03 Poll

Have outliers been a problem in your organization?

❍ Yes
❍ No

3.3 Alternatives to PROC ARIMA for Modeling Trend

Objectives

- Describe the use of PROC FORECAST and stepwise autoregression for forecasting time series with trend.
- Illustrate how to use PROC AUTOREG for forecasting.
- Describe exponential smoothing models having trend components.
- Use PROC ESM to fit exponential smoothing models for nonstationary time series.

81

Stepwise Autoregression

Full Autoregressive Model

$$Y_t = \theta_0 + \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + \cdots + \phi_p Y_{t-p} + \varepsilon_t$$

First Order Models

$$Y_t = \theta_0 + \phi_1 Y_{t-1} + \varepsilon_t$$

$$Y_t = \theta_0 + \phi_2 Y_{t-2} + \varepsilon_t$$

$$\cdots$$

$$Y_t = \theta_0 + \phi_p Y_{t-p} + \varepsilon_t$$

82

continued...

Stepwise Autoregression

Second Order Models

$$Y_t = \theta_0 + \phi_1 Y_{t-1} + \phi_k Y_{t-k} + \varepsilon_t$$

$$Y_t = \theta_0 + \phi_2 Y_{t-2} + \phi_k Y_{t-k} + \varepsilon_t$$

$$\ldots$$

$$Y_t = \theta_0 + \phi_k Y_{t-k} + \phi_p Y_{t-p} + \varepsilon_t$$

...

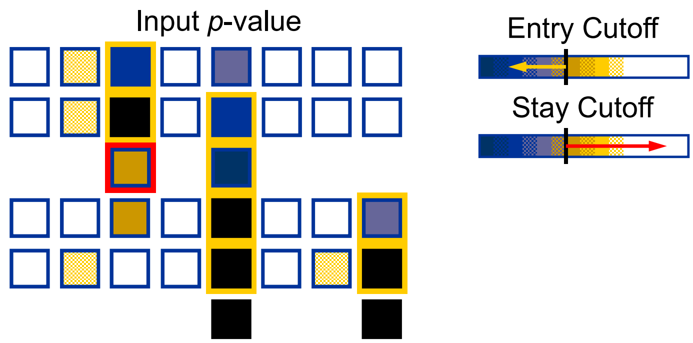
83

Stepwise Autoregression

- Stepwise autoregression is the same as stepwise variable selection in regression, except regressors are replaced by lagged values of the time series.
- The SLENTRY= option specifies the significance level required for a lag to enter the model. The default is 0.2.
- The SLSTAY= option specifies the significance level required for a lag to stay in the model. The default is 0.05.
- Specifying SLENTRY=1.0 and SLSTAY=1.0 produces Yule-Walker estimates for the specified AR order.

84

Stepwise Variable Selection



91

Stepwise Autoregression

Full Autoregressive Model with Polynomial Trend

$$Y_t = \beta_0 + \beta_1 t + \beta_2 t^2 + Z_t$$

$$Z_t = \phi_1 Z_{t-1} + \phi_2 Z_{t-2} + \cdots + \phi_p Z_{t-p} + \varepsilon_t$$

$$\varepsilon_t = \text{white noise}$$

92

PROC FORECAST Syntax

PROC FORECAST DATA=*SAS-data-set* OUT=*SAS-data-set*
    OUTEST=*SAS-data-set*
    TREND=*1*|*2*|*3*
    METHOD=*STEPAR*|*method-name*
    AR=*n*
    SLENTRY=*value* SLSTAY=*value*
    INTERVAL=*interval*
    LEAD=*n*
    <options>;
    BY *variables*;
    ID *variables*;
    VAR *variables*;
RUN;

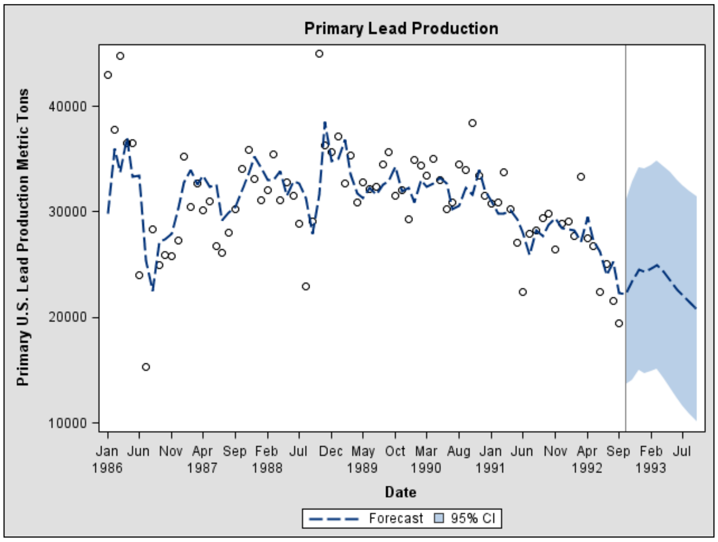
93

Forecasting with PROC FORECAST

```
proc forecast data=work.LeadMonth
              out=work.forecast
              outall
              outest=work.estimates
              outestall
              method=stepar
              ar=25
              slentry=0.20
              slstay=0.1
              trend=3
              interval=MONTH
              lead=12;
   var Primary;
   id Date;
run;
```

94

Stepwise Autoregressive Forecasts, Trend=3



95

Using Stepwise Autoregression to Forecast Annual Lead Production

This demonstration illustrates how to use PROC FORECAST to derive forecasts for the annual lead production time series using stepwise autoregression.

The code for this demonstration can be found in **Demo3_05StepAR.sas**.

Because PROC FORECAST supports linear trend internally, you do not need to pre-process the data to add a trend column or to extend the data into the future. The following code fits a linear trend model with a possible AR error term:

```
proc forecast data=sasuser.LeadYear
              out=work.LeadARfor
              outall
              outest=work.parameters
              method=stepar
              ar=6
              trend=2
              interval=year
              lead=5;
   var Primary;
   id Date;
run;
```

PROC FORECAST produces no printed output. The procedure was designed to forecast many time series at once, so any printing could possibly overwhelm output page limits. The output data set is in a format called *interleaved*. The forecasts, actual values, standard errors, and confidence limits are interleaved in the data so that a single data item has multiple observations.

A sample of the data is shown in the following display:

VIEWTABLE: Work.Leadarfor

	Date	_TYPE_	_LEAD_	Primary
85	2000	ACTUAL	0	341,000
86	2000	FORECAST	0	283,811
87	2000	RESIDUAL	0	57,189
88	2001	ACTUAL	0	290,000
89	2001	FORECAST	0	271,878
90	2001	RESIDUAL	0	18,122
91	2002	ACTUAL	0	262,000
92	2002	FORECAST	0	259,946
93	2002	RESIDUAL	0	2,054
94	2003	FORECAST	1	248,013
95	2003	L95	1	165,897
96	2003	STD	1	41,897
97	2003	U95	1	330,129
98	2004	FORECAST	2	236,080
99	2004	L95	2	153,485
100	2004	STD	2	42,141
101	2004	U95	2	318,676
102	2005	FORECAST	3	224,148
103	2005	L95	3	141,046
104	2005	STD	3	42,399
105	2005	U95	3	307,249
106	2006	FORECAST	4	212,215
107	2006	L95	4	128,582
108	2006	STD	4	42,671
109	2006	U95	4	295,848
110	2007	FORECAST	5	200,282
111	2007	L95	5	116,093
112	2007	STD	5	42,954
113	2007	U95	5	284,471

The three types (indicated by the _TYPE_ variable) for historic data are ACTUAL, FORECAST, and RESIDUAL. The four types for future forecasts (where **_LEAD_** is greater than 0) are FORECAST, L95, STD, and U95. Thus, a single date corresponds to three or four observations. You can plot interleaved data by using special options in PROC SGPLOT.

The following code prints the **OUTEST** data set, which contains parameter estimates and other statistics.

```
proc print data=parameters noobs;
run;
```

The table appears below:

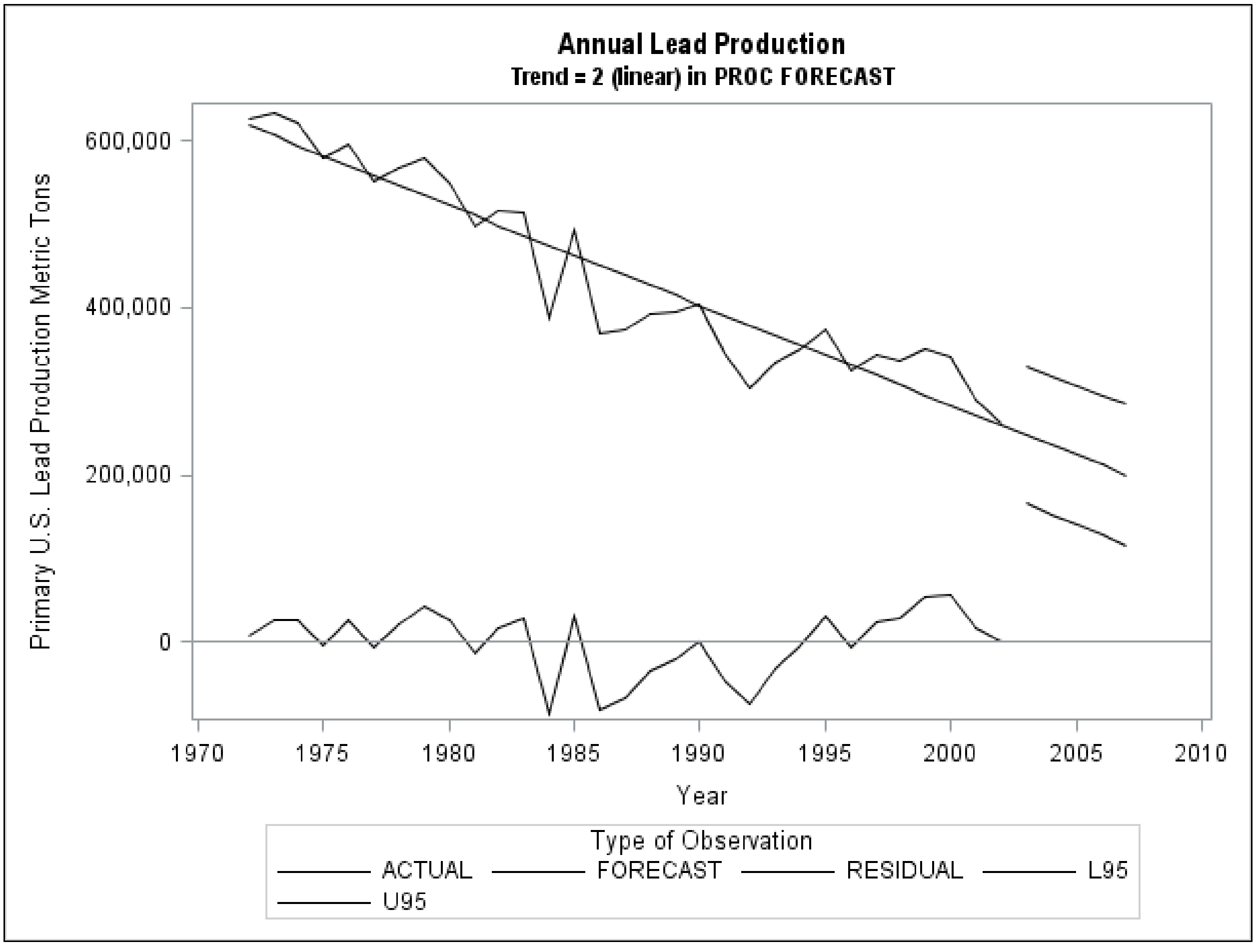
TYPE	Date	Primary
N	2002	31
NRESID	2002	31
DF	2002	29
SIGMA	2002	39317.727
CONSTANT	2002	629858.06
LINEAR	2002	-11932.66
AR1	2002	.
AR2	2002	.
AR3	2002	.
AR4	2002	.
AR5	2002	.
AR6	2002	.
SST	2002	3.9795E11
SSE	2002	4.4831E10
MSE	2002	1.54588E9
RMSE	2002	39317.727
MAPE	2002	7.6788404
MPE	2002	-0.678203
MAE	2002	30341.675
ME	2002	-4.51E-11
RSQUARE	2002	0.8873472

If you use the OUTESTALL options, you get many more goodness-of-fit statistics. You can see that PROC FORECAST chose no AR terms, so the model is a pure linear trend model that could be fit using PROC REG or PROC AUTOREG, except you would have to supply a linear trend variable and extend the data.

The following code produces a forecast plot with overlaid residuals using the interleaved data:

```
proc sgplot data=work.LeadARfor;
   where _TYPE_ ne "STD";
   series x=Date Y=primary/
          group=_TYPE_ lineattrs=(color=black pattern=solid);
   refline 0/axis=Y;
run;
```

The plot follows:



The default settings and ODS style make the legend useless, but you can ascertain the plotted curves easily. A more reasonable forecast horizon of five years was chosen.

Because a composite data set seems better suited for analysis, you can convert the interleaved data to composite data.

```
data work.composite;
   attrib Date label="Date"
          Actual length=8 label="Primary"
          Forecast length=8 label="Forecast"
          Residual length=8 label="Residual"
          STD length=8 label="Standard Error of Forecast"
          L95 length=8 label="Lower 95% Prediction Limit"
          U95 length=8 label="Upper 95% Prediction Limit";
   set work.LeadARfor;
   by Date;
   retain Forecast Actual Residual L95 U95 STD .;
   if (first.Date) then do;
      Forecast=.;
      Actual=.;
      Residual=.;
      L95=.;
      U95=.;
      STD=.;
   end;
   if (upcase(_TYPE_)='FORECAST') then
      Forecast=Primary;
   else if (upcase(_TYPE_)='ACTUAL') then
      Actual=Primary;
   else if (upcase(_TYPE_)='RESIDUAL') then
      Residual=Primary;
   else if (upcase(_TYPE_)='L95') then
      L95=Primary;
   else if (upcase(_TYPE_)='U95') then
      U95=Primary;
   else if (upcase(_TYPE_)='STD') then
      STD=Primary;
   if (last.Date) then output;
   keep Date Forecast Actual Residual STD L95 U95;
run;
```

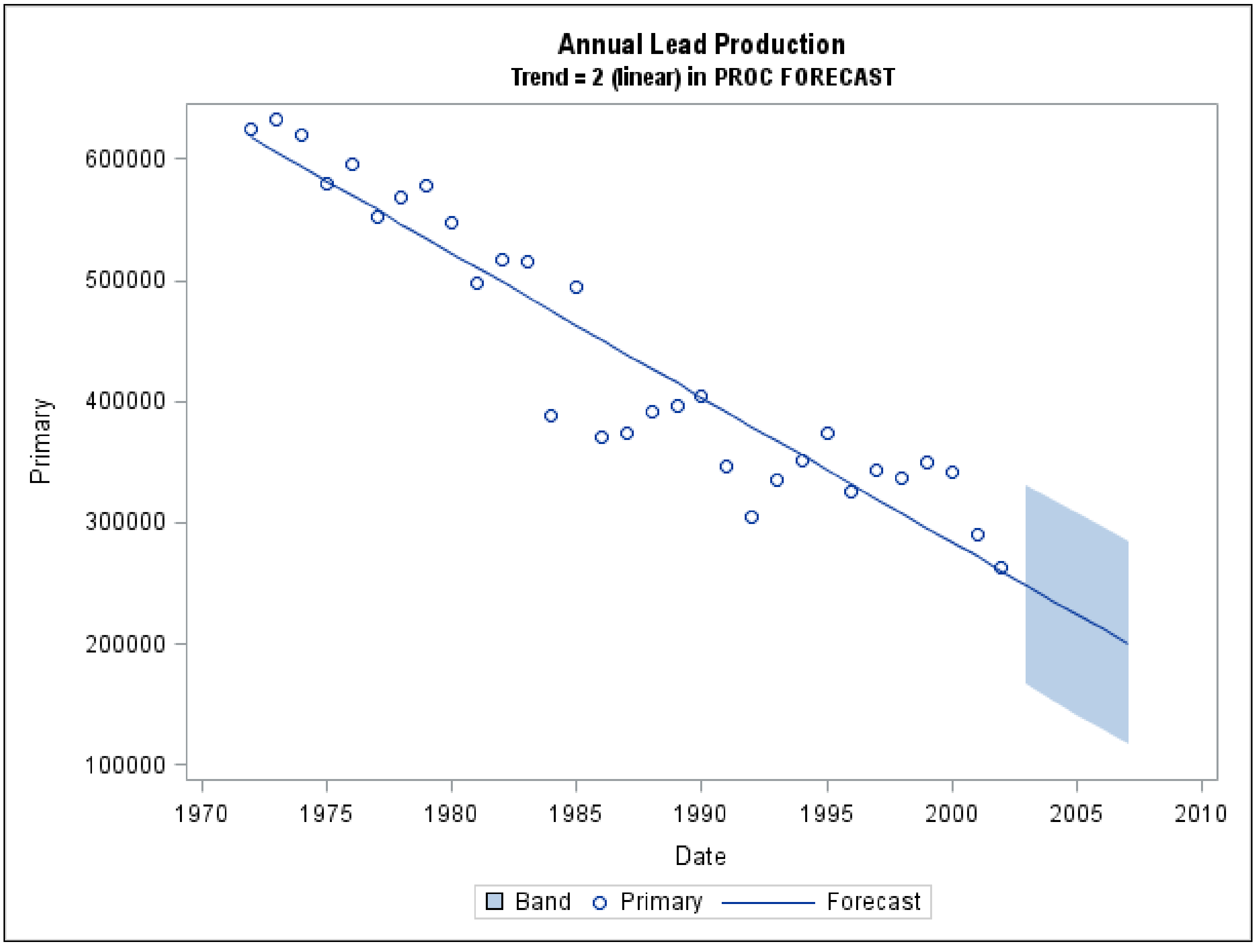
The task of converting interleaved data to composite data can become tiresome, so a macro was created for that purpose. The following macro call will produce a data set identical to the one created above. The macro converts the data in place, meaning that it overwrites the existing interleaved data with the new composite data.

```
%ILeave2CompositeAll(work.LeadARfor,Primary,Date);
```

The composite data makes use of PROC SGPLOT easier.

```
proc sgplot data=work.composite;
   band  x=Date upper=U95 lower=L95;
   scatter x=Date y=Actual;
   series x=Date y=Forecast;
run;
```

The plot appears below:



You can manipulate model selection in PROC FORECAST by using the SLENTRY and SLSTAY options. If the *p*-value of an AR coefficient is larger than SLENTRY, it is not allowed into a model. If the *p*-value for an AR term already in the model exceeds SLSTAY, it is removed from the model. Using SLSTAY=1 and SLENTRY=1 is equivalent to fitting a full model using Yule-Walker estimates for the AR coefficients. For example, if you want to emulate earlier results and fit a linear trend with an AR(2) error model, use the following code:

```
proc forecast data=sasuser.LeadYear
              out=work.LeadAR2for
              outall
              outest=work.parmar2
              method=stepar
              slentry=1.
              slstay=1.
              ar=2
              trend=2
              interval=year
              lead=5;
   var Primary;
   id Date;
run;
```

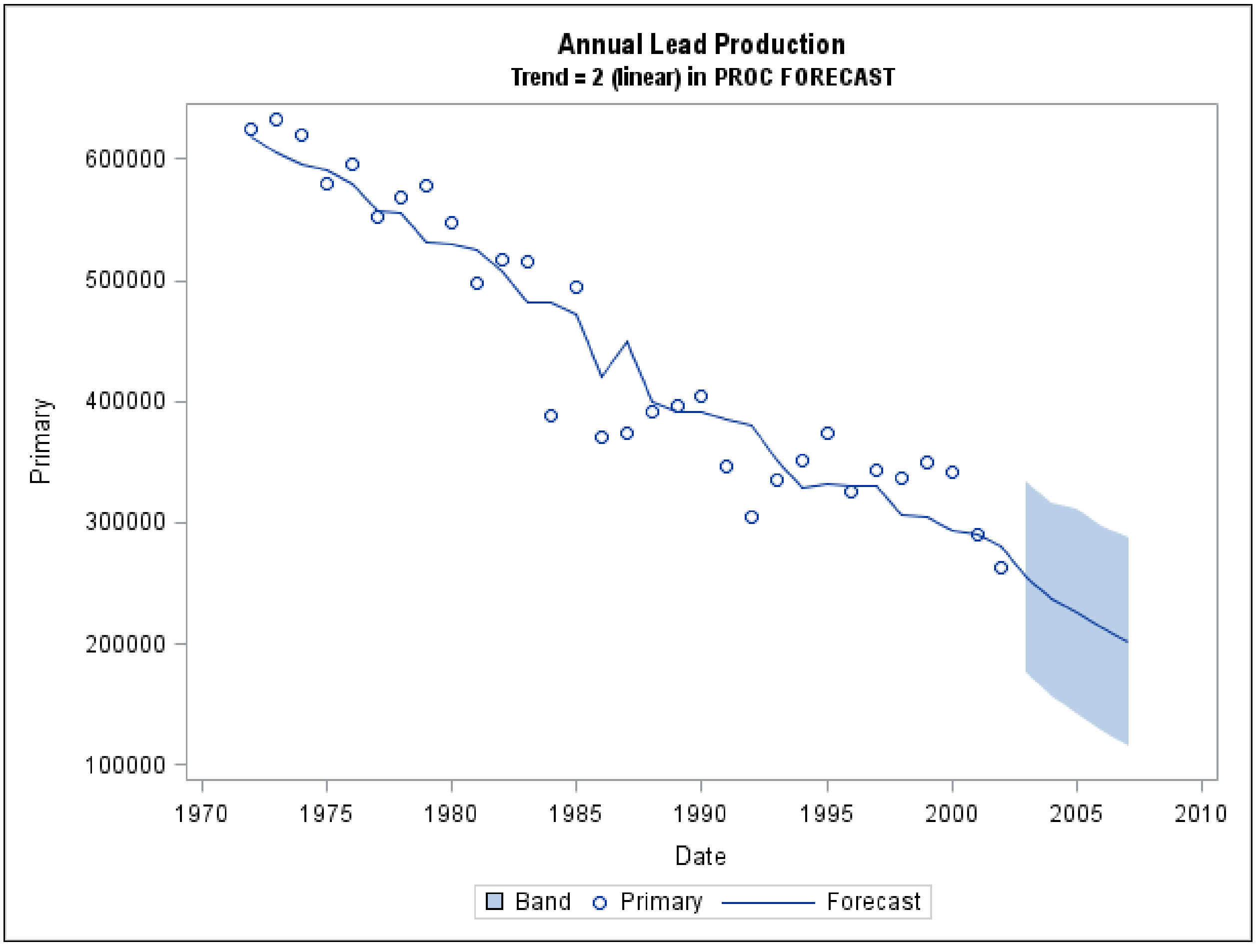
Reproducing the AR(2) model with only the lag 2 coefficient is more difficult, and might be impossible to fit using PROC FORECAST. For the annual lead production data, the *p*-value for the lag 1 term is larger than the *p*-value for the lag 2 term in the full AR(2) model, so you can manipulate SLSTAY to remove the lag 1 term. The following code does the trick:

```
proc forecast data=sasuser.LeadYear
              out=work.LeadAR2for2
              outall
              outest=work.parmar2lag2
              method=stepar
              slentry=1.
              slstay=0.1
              ar=2
              trend=2
              interval=year
              lead=5;
   var Primary;
   id Date;
run;
```

The forecast plot is obtained using the following code:

```
%ILeave2CompositeAll(work.LeadAR2for2,Primary,Date);
proc sgplot data=work.LeadAR2for2;
   band  x=Date upper=U95 lower=L95;
   scatter x=Date y=Actual;
   series x=Date y=Forecast;
run;
```

The plot appears below:



You can use PROC COMPARE to compare the forecasts for this model to the forecasts using a pure linear trend with no AR error component.

```
proc compare data=work.LeadARfor compare=work.LeadAR2for2;
   id Date;
   var Forecast;
run;
```

The following results used a WHERE clause to restrict results to 2000-2007:

Value Comparison Results for Variables

Date	Forecast Base Forecast	Compare Forecast	Diff.	% Diff
2000	283811	294053	10243	3.6089
2001	271878	290830	18951	6.9705
2002	259946	279921	19976	7.6846
2003	248013	254343	6330	2.5522
2004	236080	236798	717.5982	0.3040
2005	224148	226359	2211	0.9864
2006	212215	212466	250.6514	0.1181
2007	200282	201055	772.2664	0.3856

The first future forecast for 2003 differs by approximately 2.6%. The remaining future forecasts differ by less than 1%.

PROC FORECAST provides an alternative to PROC ARIMA, but you must decide which procedure produces the best forecasts for your data.

PROC AUTOREG

PROC AUTOREG DATA=*SAS-data-set* <*options*>;
MODEL *variable* = *variables*
</ NLAG=*n* BACKSTEP *options*>;
RUN;

```
proc autoreg data=work.MonthLead;
   model Primary=Time TimeSq X X2
                  /nlag=13 backstep;
run;
```

97

PROC AUTOREG was introduced in the last section to investigate results from the augmented Dickey-Fuller test and to illustrate the use of indicator variables in a forecast model.

Using PROC AUTOREG to Forecast Lead Production

This demonstration illustrates how to use PROC AUTOREG to forecast annual and monthly lead production.

The code for this demonstration can be found in **Demo3_06Autoreg.sas**.

The data must be preprocessed as before.

```
data work.YearlyLead;
   set sasuser.LeadYear end=eof;
   Time+1;
   output;
   if (eof) then do future=1 to 5;
      Primary=.;
      Secondary=.;
      Total=.;
      Time+1;
      Date=intnx("year",Date,1);
      output;
   end;
   drop future;
run;
```

The following code uses PROC AUTOREG to fit linear trend with an AR error component. Backward elimination is used to select AR lags.

```
proc autoreg data=YearlyLead;
   model Primary=Time / nlag=6 backstep;
   output out=OutAReg_Y UCL=U95 LCL=L95 Predicted=Forecast;
run;
```

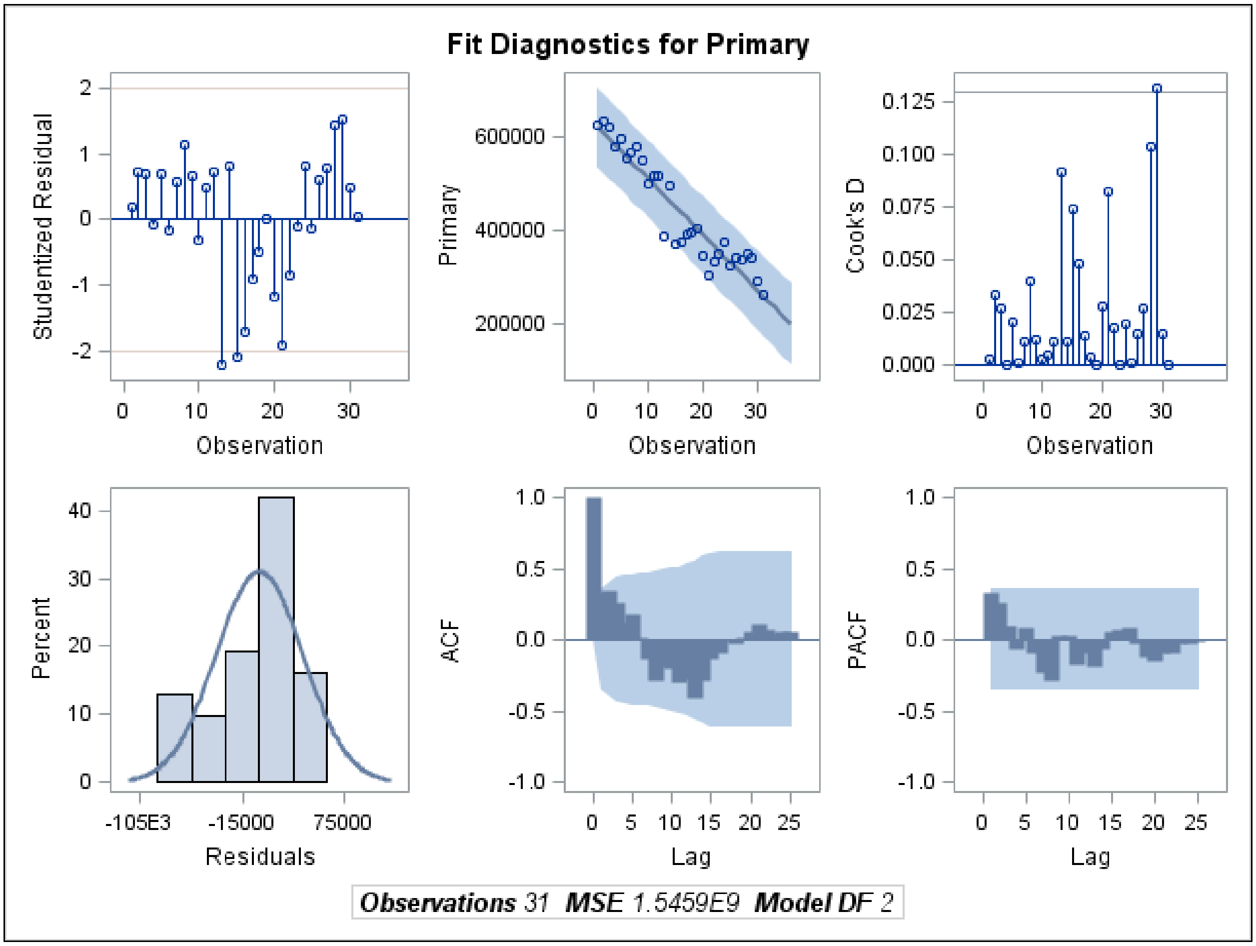
As PROC FORECAST did, PROC AUTOREG eliminated all candidate lags and settled on a simple linear regression model for predictions of the yearly lead production data.

Backward Elimination of Autoregressive Terms			
Lag	**Estimate**	**t Value**	**Pr > |t|**
4	0.064338	0.30	0.7704
3	-0.095666	-0.45	0.6559
6	0.094507	0.49	0.6290
5	-0.088296	-0.48	0.6348
1	-0.245348	-1.32	0.1969
2	-0.349292	-1.97	0.0585

The final model is a simple linear trend model.

Parameter Estimates					
Variable	**DF**	**Estimate**	**Standard Error**	**t Value**	**Approx Pr > |t|**
Intercept	1	629858	14472	43.52	<.0001
Time	1	-11933	789.5190	-15.11	<.0001

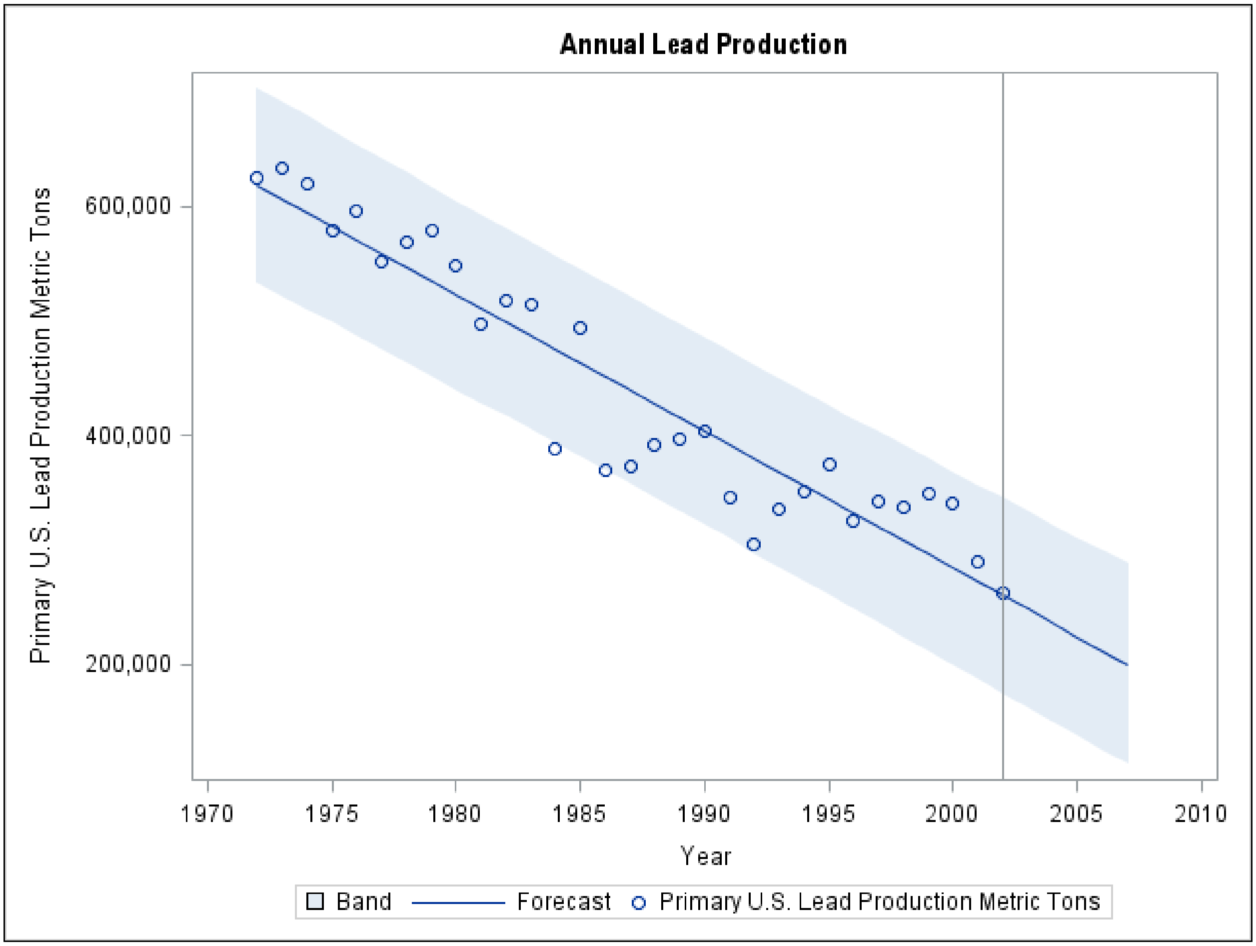
PROC AUTOREG provides comprehensive model diagnostics.



The following code produces a plot of the forecasts:

```
proc sgplot Data=work.outAReg_Y;
   band  X=Date upper=U95 lower=L95 / transparency=0.60;
   series X=Date Y=Forecast;
   scatter X=Date Y=Primary;
   refline "01jan2002"d /axis=X;
run;
```

The plot appears below:



The procedures in this chapter have various limitations. Typically the simpler procedures to use are those with the greater limitations. PROC ARIMA is the most general and powerful but requires the most user manipulation and thought.

Procedure	General Deterministic Inputs?	Automatic Difference Handling?	Autoregressive and Moving Average Errors?
ARIMA	Yes	Yes	Yes
AUTOREG	Yes	No	No: AR only
FORECAST	No (*)	No	No: AR only
ESM	No (*)	No	No

* Constant, linear, or quadratic trends only

For the monthly lead production, preparing the data includes the addition of variables **X** and **X2** that were explained previously.

```
data work.MonthLead;
   set sasuser.leadmonth end=eof;
   Time+1;
   TimeSq=Time*Time;
   X=(Date<"01jun1986"d);
   X2=(Date="01oct1988"d)-(Date="01Aug1988"d);
   output;
   if eof then do future=1 to 24;
      Time+1;
      TimeSq=Time*Time;
      Primary=.;
      Date=intnx('month',Date,1);
      output;
   end;
   drop future;
run;
```

The quadratic trend model is fit with the outlier variables and an AR error component.

```
proc autoreg data=work.MonthLead;
   model primary=Time TimeSq X X2 / nlag=13 backstep;
   output out=work.outAREG_M PM=P_mean P=P_indiv;
quit;
```

For the monthly lead data, PROC AUTOREG, unlike PROC FORECAST (and PROC ESM as you see later), can accommodate predictors other than low order polynomials. Dummy variables such as those introduced in the outlier demonstration can be used. When input variables are used, the model consists of a predicted mean function with parameters estimated with generalized least squares (that is, incorporating autocorrelation) and an autoregressive error term around that function. For the monthly lead production data with regressors **Time**, **TimeSQ**, **X**, and **X2**, the parameter estimates of the regression part of the model are tabulated as below:

Parameter Estimates

Variable	DF	Estimate	Standard Error	t Value	Approx Pr > |t|
Intercept	1	21747	819.8097	26.53	<.0001
Time	1	574.7867	42.7303	13.45	<.0001
TimeSq	1	-6.8328	0.4809	-14.21	<.0001
X	1	15681	1475	10.63	<.0001
X2	1	10246	1666	6.15	<.0001

The autoregressive part of the model produces a separate table of estimates.

Estimates of Autoregressive Parameters			
Lag	Coefficient	Standard Error	t Value
4	0.396520	0.108979	3.64
8	0.348052	0.108979	3.19

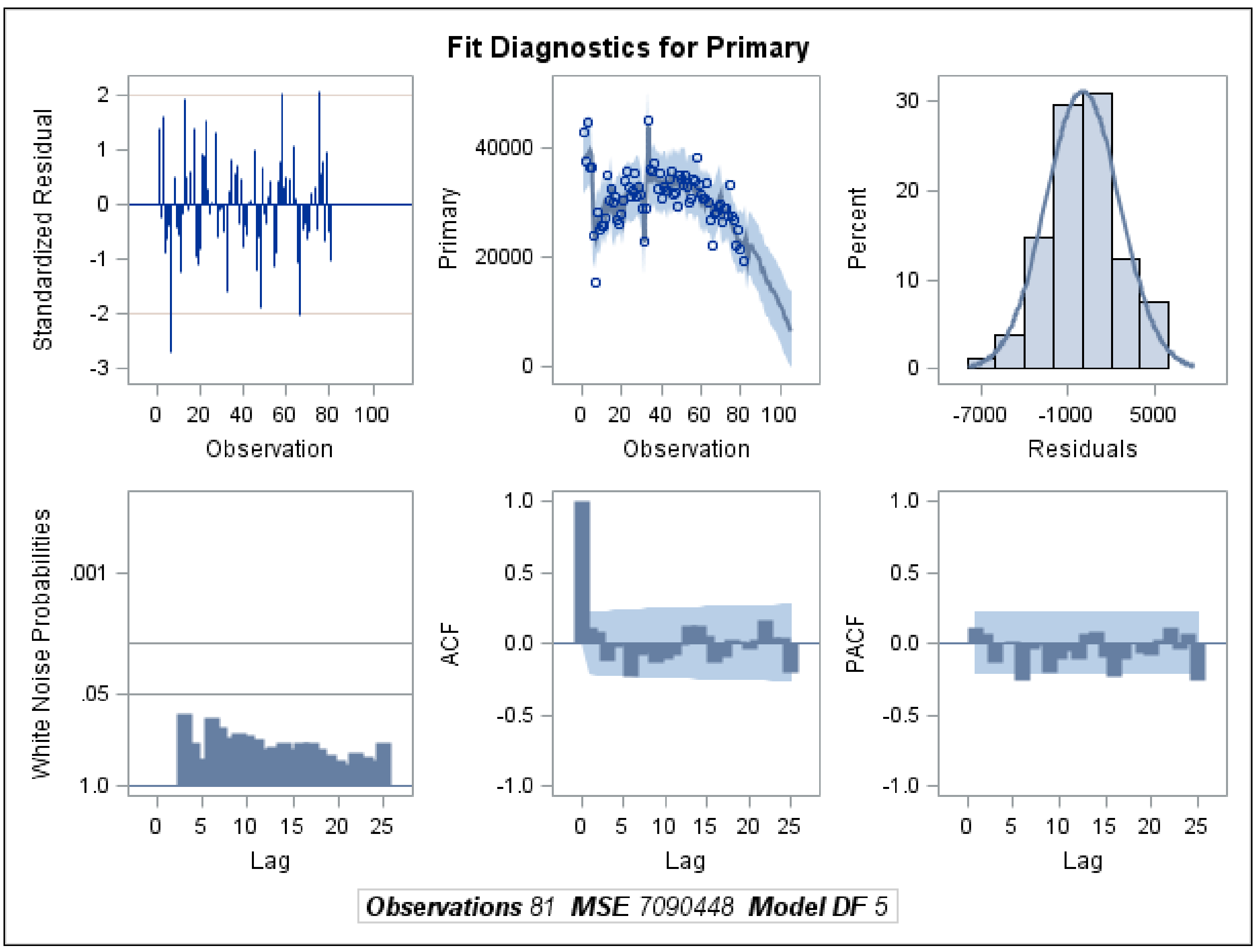
Because of the notation used in the article upon which the original version of PROC AUTOREG was based, PROC AUTOREG parameterizes the autoregressive error model as follows:

$$Z_t + \phi_1 Z_{t-1} + \phi_2 Z_{t-2} + \cdots + \phi_p Z_{t-p} = \varepsilon_t$$

Thus, the signs of the lagged coefficients are opposite of the sign employed in the PROC ARIMA notation. The estimated error model derived for this data is given below:

$$Z_t + 0.39652 Z_{t-4} + 0.348052 Z_{t-8} = \varepsilon_t$$

The ODS GRAPHICS statement for PROC AUTOREG gives some diagnostics reminiscent of those from PROC ARIMA.



The Results

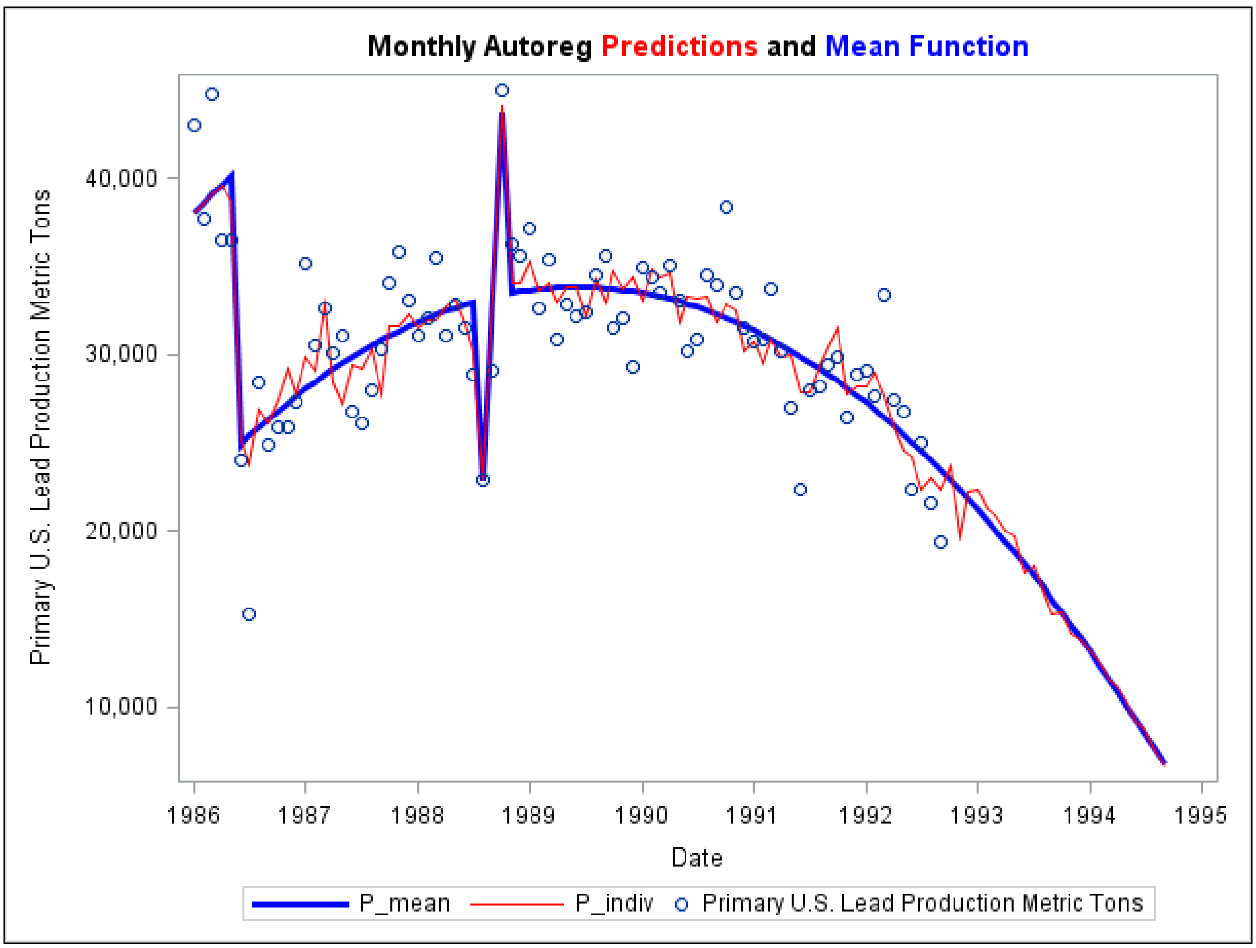
The mean function adds the predictions from the error model to get the final forecasts. The error model is as follows:

$$Z_t = -0.39652 Z_{t-4} - 0.348052 Z_{t-8} + \varepsilon_t$$

An overlay of the mean function and the predictions shows the effect of **X** and **X2** as well as the predictive capability of the combined model. The following code plots the forecasts overlaid with the regression mean function:

```
title1 "Monthly Autoreg" c=red " Predictions " c=black "and "
                          c=blue "Mean Function";
proc sgplot data=work.outAREG_M;
   series X=date Y=P_mean / lineattrs=(thickness=3 color=blue
          pattern=solid);
   series X=date Y=P_indiv / lineattrs=(thickness=1 color=red
          pattern=solid);
   scatter X=date Y=Primary;
run;
```

The plot follows:



Because the data set was extended 24 months, PROC AUTOREG forecasts to the end of the data. PROC AUTOREG has no LEAD= option to specify a forecast horizon because it anticipates input variables, and it has no internal capability to forecast input variables as PROC ARIMA does. This distinction makes it clear that PROC AUTOREG is more similar to PROC REG than PROC ARIMA. It is a regression tool intended primarily for interpolative prediction and inference.

A summary of distinctive features of PROC AUTOREG that contrast it with PROC ARIMA is warranted.

PROC AUTOREG Features

- Stationarity tests, including augmented Dickey-Fuller tests
- Multiple estimation options, including Yule-Walker (default), conditional least squares, and maximum likelihood
- Models that address non-constant variance of the error (heteroskedasticity)

99

PROC AUTOREG is also a convenient learning tool that can help a student transition from the world of ordinary least squares regression to the world of ARIMA models and autocorrelated errors.

PROC ESM implements exponential smoothing models and provides another alternative to PROC ARIMA for forecasting.

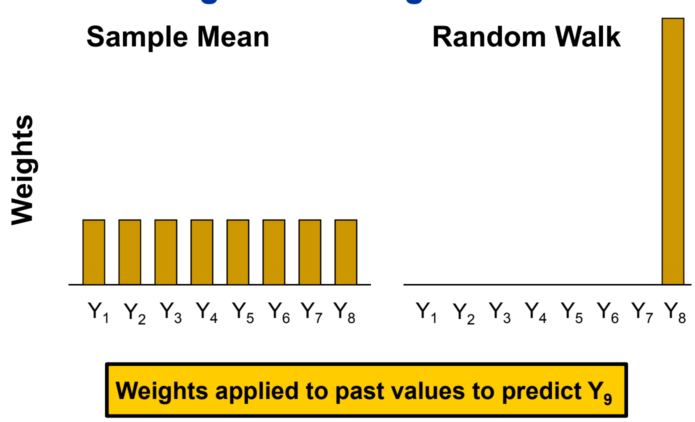
Exponential Smoothing Models: Premise

- Weighted averages of past values can produce good forecasts of the future.
- The weights should emphasize the most recent data.
- Forecasting should only require a few parameters.
- Forecast equations should be simple and easy to implement.

100

There are seven common exponential smoothing models, and these are supported by PROC ESM. In addition, a few exponential smoothing models are not as common and are not supported. For example, triple exponential smoothing models use third differencing, that is, three first differences. Such models are addressed in textbooks but rarely provide good forecasts for real data, and when they do, the application is usually highly specialized.

ESM as Weighted Averages



101

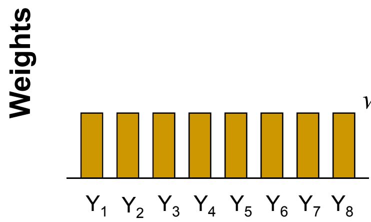
ESM as Weighted Averages

Sample Mean

$$\hat{Y}_{n+1} = \sum_{t=1}^{n} w_t Y_t = w_1 Y_1 + w_2 Y_2 + \cdots + w_n Y_n$$

$$= \sum_{t=1}^{n} \frac{1}{n} Y_t = \frac{1}{n} \sum_{t=1}^{n} Y_t = \overline{Y}$$

$$w_t = \frac{1}{n}$$



$$\hat{Y}_9 = \frac{1}{8} \sum_{t=1}^{8} Y_t$$

The mean is a weighted average where all weights are the same.

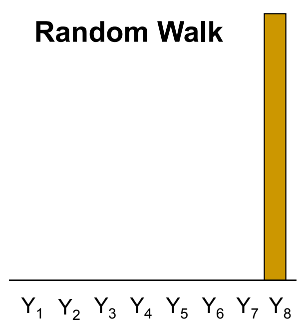
102

ESM as Weighted Averages

$$\hat{Y}_{n+1} = \sum_{t=1}^{n} w_t Y_t = Y_n$$

$$w_n = 1, w_t = 0 \text{ for } t = 1,2,\ldots,n-1$$

Random Walk



$$\hat{Y}_9 = Y_8$$

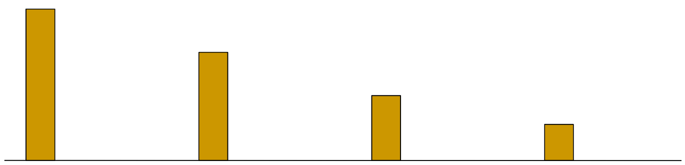
A random walk forecast is a weighted average where all weights are 0 except the most recent, which is 1.

103

The Exponential Smoothing Coefficient

Forecast Equation

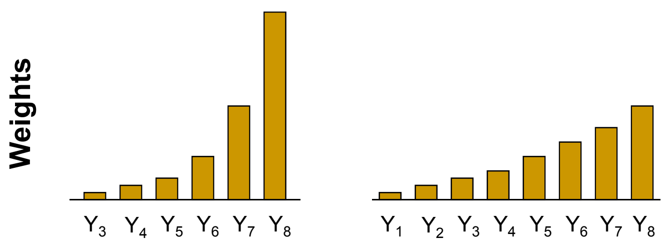
$$
\begin{aligned}
\hat{Y}_{t+1} &= \omega Y_t + (1-\omega)\hat{Y}_t \\
&= \omega Y_t + (1-\omega)[\omega Y_{t-1} + (1-\omega)\hat{Y}_{t-1}] \\
&= \omega Y_t + \omega(1-\omega)Y_{t-1} + (1-\omega)^2\hat{Y}_{t-1} \\
&= \omega Y_t + \omega(1-\omega)Y_{t-1} + (1-\omega)^2[\omega Y_{t-2} + (1-\omega)\hat{Y}_{t-2}] \\
&= \omega Y_t + \omega(1-\omega)Y_{t-1} + \omega(1-\omega)^2 Y_{t-2} + \omega(1-\omega)^3 Y_{t-3} + \cdots
\end{aligned}
$$



104

Simple Exponential Smoothing

$\omega = 0.5$ $\omega = 0.25$



Weights applied to past values to predict Y_9

As the parameter increases, the emphasis on the most recent values increases.

105

PROC ESM Syntax

PROC ESM DATA=*SAS-data-set* OUT=*SAS-data-set*
OUTEST=*SAS-data-set*
OUTFOR=*SAS-data-set*
OUTSTAT=*SAS-data-set*
OUTSUM=*SAS-data-set*
SEASONALITY=*n*
PLOT=*option*|(*options*)
PRINT=*option*|(*options*)
LEAD=*n*
<*options*>**;**
BY *variables***;**
ID *variable* INTERVAL=*interval***;**
FORECAST *variables* / MODEL=*model* <*options*>**;**
RUN;

106

Forecasting with PROC ESM

```
proc esm data=work.WidgetSales
     out=work.out
     outfor=work.outfor
     outest=work.outest
     outstat=work.outstat
     outsum=work.outsum
     print=(ESTIMATES STATISTICS SUMMARY);
   id Date interval=month;
   forecast UnitsSold / model=simple;
run;
```

107

Exponential Smoothing Models

- Models for time series with trend:
  - Simple exponential smoothing
  - Double (Brown) exponential smoothing
  - Linear (Holt) exponential smoothing
  - Damped-trend exponential smoothing
- Models for time series with seasonality:
  - Seasonal exponential smoothing
- Models for time series with trend and seasonality
  - Winters additive exponential smoothing
  - Winters multiplicative exponential smoothing

108

Because there are exactly seven models, trial-and-error becomes an effective strategy for model selection in the age of high speed computers. Consequently, a macro is included in the course material. The macro can fit all seven models and record various goodness-of-fit statistics for each model. An example of a call to the macro follows:

```
%AutoESM(sasuser.LeadYear,work.ESMstats,Primary,Date);
```

Simple Exponential Smoothing Predictions

$$\hat{Y}_1 = S_0 = \text{starting value},$$

$$\hat{Y}_2 = \omega Y_1 + (1-\omega)\hat{Y}_1 = \omega Y_1 + (1-\omega)S_0,$$

$$\hat{Y}_3 = \omega Y_2 + (1-\omega)\hat{Y}_2,$$

$$\cdots$$

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

The forecast equation only requires knowledge of the previous value, the previous forecast, and the smoothing weight ω.
The starting value is often taken to be the mean of the first *n* observations. PROC ESM uses *backcasting* to calculate the initial forecast.

109

Smoothing Weights

ω **Level smoothing weight**

γ **Trend smoothing weight**

ϕ **Trend damping weight**

The choice of Greek letter is arbitrary. The software uses names rather than Greek symbols.

110

ESM Parameters

ESM	Parameters	Component
Simple	ω	Level
Double	ω	Level/Trend
Linear (Holt)	ω, γ	Level,Trend
Damped-Trend	ω, γ, ϕ	Level,Trend,Damping

111

PROC ESM Syntax

PROC ESM DATA=*SAS-data-set* OUT=*SAS-data-set*
OUTEST=*SAS-data-set*
OUTFOR=*SAS-data-set*
OUTSTAT=*SAS-data-set*
OUTSUM=*SAS-data-set*
SEASONALITY=*n*
PLOT=*option*|(*options)*
PRINT=*option*|(*options)*
LEAD=*n*
<options>;
BY *variables*;
ID *variable* INTERVAL=*interval*;
FORECAST *variables* / MODEL=*model <options>*;
RUN;

112

ESM Parameters and Keywords

ESM	Parameters	Model= Keyword
Simple	ω	SIMPLE
Double	ω	DOUBLE
Linear (Holt)	ω, γ	LINEAR
Dampled-Trend	ω, γ, ϕ	DAMPTREND
Seasonal	ω, δ	SEASONAL
Additive Winters	ω, γ, δ	ADDWINTERS
Multiplicative Winters	ω, γ, δ	WINTERS

113

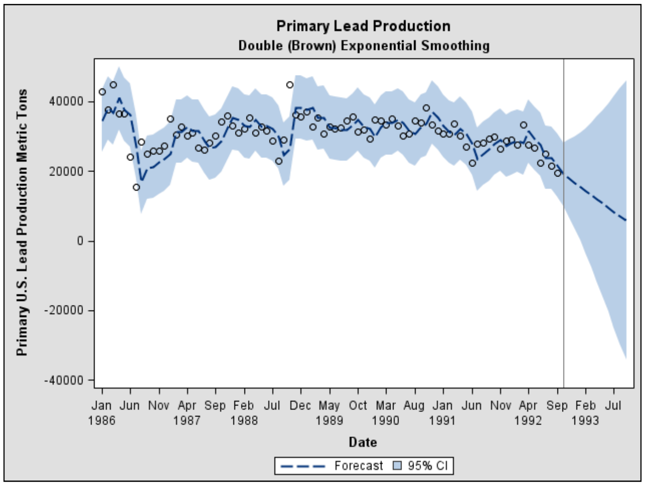
Exponential Smoothing Models for Lead Production

```
proc esm data=work.LeadMonth
     out=work.out
     outfor=work.forecast
     outest=work.estimates
     outstat=work.stats
     outsum=work.summary
     lead=12
     print=(ESTIMATES STATISTICS SUMMARY);
   id Date interval=month;
   forecast Primary / model=double;
run;
```

114

continued...

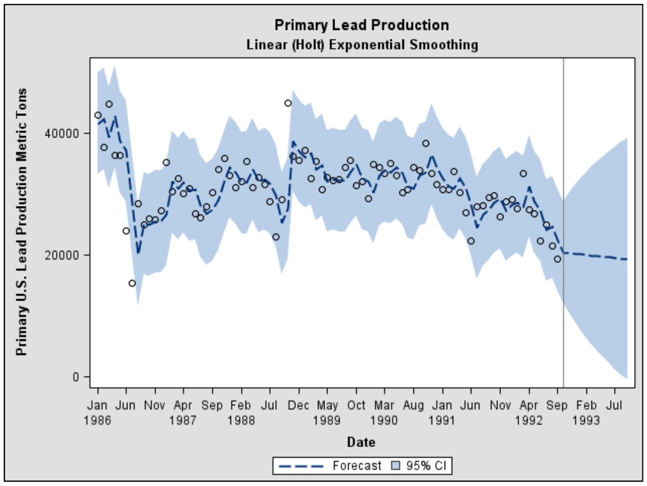
Exponential Smoothing Models for Lead Production



115

continued...

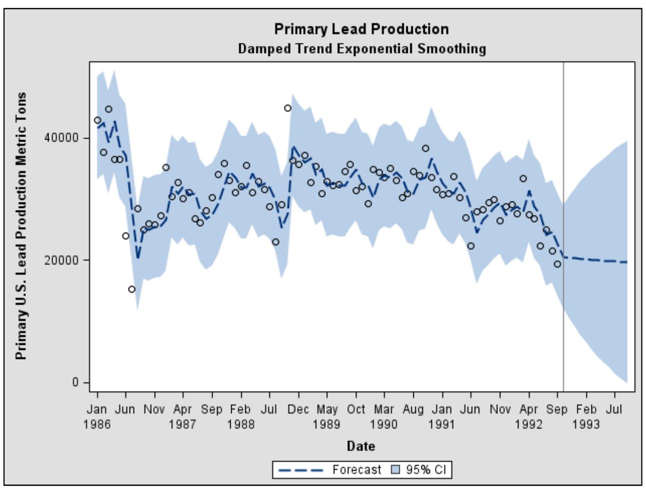
Exponential Smoothing Models for Lead Production



116

continued...

Exponential Smoothing Models for Lead Production



117

continued...

Exponential Smoothing Models for Lead Production

Primary Lead Production

Exponential Smoothing Models for Trend

Model	MAPE	RMSE	AIC	SBC
Linear	10.381	4207.0	1355.8	1360.6
DampTrend	10.390	4207.8	1357.8	1365.0
Simple	10.429	4214.8	1354.1	1356.5
Double	11.592	4593.0	1368.0	1370.4

118

Forecasting Using PROC ESM

This demonstration illustrates how to use PROC ESM to derive forecasts.

The code for this demonstration can be found in **Demo3_07ESM.sas**.

No data preparation is required by PROC ESM unless you need to use PROC TIMESERIES or PROC EXPAND to accumulate data or replace missing values. The following code produces forecasts for both primary and secondary lead production. The results for secondary lead production are included to show that PROC ESM is intended for use on wide data sets with many series to be forecast.

```
proc esm data=sasuser.LeadYear
      outfor=work.outfor(rename=(LOWER=L95 UPPER=U95 PREDICT=Forecast))
      print=(estimates statistics summary)
      lead=5; /* Default is LEAD=12 !!! */
   id Date interval=year;
   forecast Primary Secondary / model=Linear;
run;
```

MODEL=LINEAR was judged to have the greatest chance for success. Method LINEAR is Holt's method that uses two smoothing parameters to derive forecasts, making it more flexible than SIMPLE or DOUBLE, which use only one parameter.

The printed output includes goodness-of-fit statistics, of which there is no shortage.

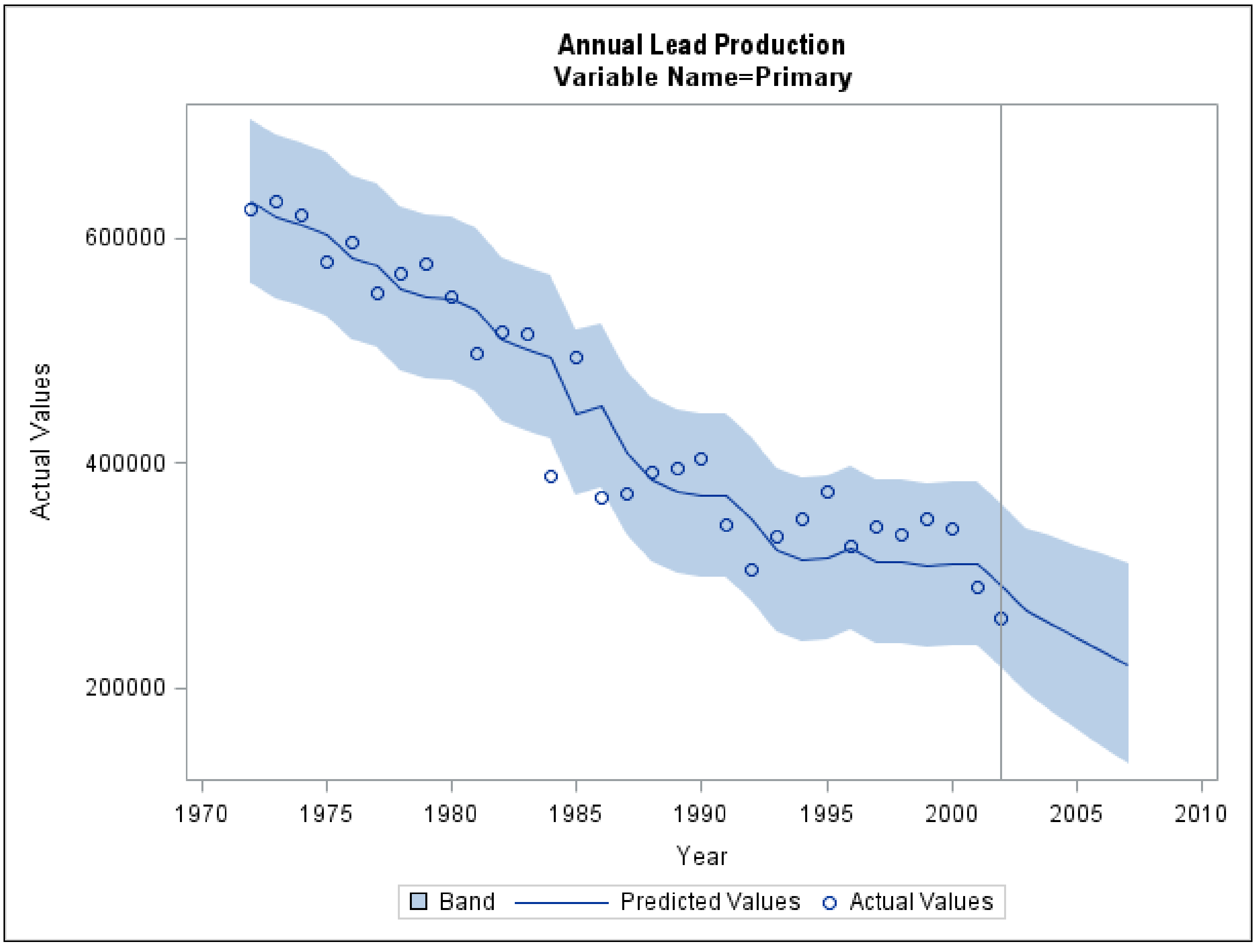
A subset of the output is shown below:

Statistics of Fit for Variable Secondary	
Statistic	**Value**
Degrees of Freedom Error	29
Number of Observations	31
Number of Observations Used	31
Number of Missing Actuals	0
Number of Missing Predicted Values	0
Number of Model Parameters	2
Total Sum of Squares	1.8743E13
Corrected Total Sum of Squares	1.38196E12
Sum of Square Error	8.84698E10
Mean Square Error	2853863975
Root Mean Square Error	53421.5684
Unbiased Mean Square Error	3050682180
Unbiased Root Mean Square Error	55232.9809
Mean Absolute Percent Error	6.06247658
Mean Absolute Error	39896.6543
R-Square	0.93598253
Adjusted R-Square	0.93377503
Amemiya's Adjusted R-Square	0.92715253
Random Walk R-Square	0.03093724
Akaike Information Criterion	678.930131
Schwarz Bayesian Information Criterion	681.798105

You can plot the forecasts for all series using BY-group processing. PROC ESM stacks the forecasts based on the name of the variable being forecast. The name of the variable is placed in the **_NAME_** column. Sorting by **_NAME_** and **Date** ensures that the data can be plotted by PROC SGPLOT.

```
proc sort Data=work.outfor;
   by _NAME_ Date;
run;
proc sgplot data=work.outfor; by _NAME_;
   band  X=Date upper=U95 lower=L95;
   series X=Date Y=Forecast;
   scatter X=Date Y=Actual;
   refline "01jan2002"d /axis=X;
run;
```

The plot for **Primary** is shown below:



The forecasts generated by PROC ESM using MODEL=LINEAR are competitive with those produced by other procedures. Notice that the values AIC and SBC are based on the SSE formula, not the likelihood formula. The SSE formula is used because one of the models, the Winters multiplicative model, does not have a likelihood function. Because PROC ARIMA uses the likelihood formula for AIC and SBC, you cannot compare the PROC ESM results directly to the PROC ARIMA results. However, it is easy to calculate the SSE version for any set of forecasts.

As mentioned previously, because there are only seven exponential smoothing models, a macro was written to fit all seven models (if appropriate) and produce goodness-of-fit statistics for each model. You can do this for the primary lead production series. The code for doing this follows:

```
%AutoESM(sasuser.LeadYear,work.ESMstats,Primary,Date);
proc print data=work.ESMstats noobs;
   var Model AIC SBC MAPE RMSE;
run;
```

The data set that is created, given by the second argument to the macro, contains all goodness-of-fit statistics produced by PROC ESM. PROC PRINT enables you to select the statistics of interest. The printed results follow:

Model	AIC	SBC	MAPE	RMSE
Simple	661.882	663.316	7.48419	41909.17
Double	659.528	660.962	8.20304	40347.93
Linear	654.354	657.222	7.27891	35939.19
DampTrend	656.286	660.588	7.18125	35899.56

The Holt linear exponential smoothing model appears to provide the most accurate forecasts. The damped trend model should be disqualified unless you actually want to dampen the forecasts. Choice of a damped trend model is made before the analysis and would be appropriate if you only had trend in the data and bounding the forecasts is preferred. Goodness-of-fit statistics for damped trend can be misleading.

Exercises

1. Reproduce the results from the **Demo3_03LeadMonthly** program. For extra credit, write a program to use PROC REG to combine the quadratic trend model and the ARIMA(0,1,1) model forecasts.

2. Verify that a simple exponential smoothing model is similar to an ARIMA(0,1,1) model using the monthly lead production data. (In fact, the simple model is equivalent to an ARIMA(0,1,1) model subject to parameter constraints, but results are also slightly different because of different estimation techniques even if the parameter constraints are satisfied.)

   Hint: You need to use the NOCONSTANT option in the PROC ARIMA ESTIMATE statement.

3.4 Chapter Summary

Box-Jenkins time series analysis promotes the use of differencing to model trend. The general Box-Jenkins ARIMA(*p,d,q*) model can be employed to forecast nonstationary time series using stochastic trend and seasonal components. The augmented Dickey-Fuller test provides methodology for determining whether a time series is stationary or not.

The Box-Jenkins methodology developed in a previous chapter is expanded to include the identification of difference order *d*. The estimation and forecasting steps include a step to convert a series to stationarity, estimate the stationary component, and then integrate with the original series.

Despite the emphasis on differencing, the methodology accommodates deterministic trend components as well as stochastic ones. PROC ARIMA provides a full implementation of Box-Jenkins models.

PROC AUTOREG, PROC FORECAST, and PROC ESM provide alternatives to PROC ARIMA for forecasting models with trend. PROC ESM supports four exponential smoothing models for time series with trend.

For Additional Information

Bartlett, M. S. 1966. *An Introduction to Stochastic Processes*. Second Edition, Cambridge: Cambridge University Press.

Bowerman, Bruce L., Richard T. O'Connell, and Anne B. Koehler. 2005. *Forecasting, Time Series, and Regression: An Applied Approach.* Belmont, California: Thomson-Brooks/Cole.

Box, G.E.P., and G.M. Jenkins. 1976. *Time Series Analysis: Forecasting and Control*. Oakland, California: Holden-Day.

Dickey, David A., and Wayne A. Fuller. 1979. "Distribution of the estimates for autoregressive time series with a unit root," *Journal of the American Statistical Association*, **74**, 427-431.

3.5 Solutions

Solutions to Exercises

1. Reproduce the results from the **Demo3_03LeadMonthly** program. For extra credit, write a program to use PROC REG to combine the quadratic trend model and the ARIMA(0,1,1) model forecasts.

   The demonstration results are in the text. For the extra credit portion, consider the following program:

```
title1 font=&COURSEFONT color=black "Monthly Lead Production";

/*---- Make sure you have run Demo3_03LeadMonth.sas ----*/
/*---- and have access to datasets: work.ML_RWD   ----*/
/*----                 work.ML_Quad   ----*/

data work.regdata;
   merge work.ML_RWD(keep=Date Primary Forecast Std
             rename=(Forecast=ForeIMA11 Std=StdIMA11))
         work.ML_Quad(keep=Date Primary Forecast Std
             rename=(Forecast=ForeQuad Std=StdQuad));
   by Date;
run;

proc reg data=work.regdata outest=work.regest;
   model Primary=ForeIMA11 ForeQuad / noint;
   restrict ForeIMA11+ForeQuad=1;
run;

/*---- Unfortunately, ForeQuad dominates and the ----*/
/*---- estimate for ForeIMA11 is negative.    ----*/

%let W1=1.01877;
%let W2=-0.01877;
data work.newfore;
   set work.regdata;
   Forecast=(&W1)*ForeQuad+(&W2)*ForeIMA11;
   Std=sqrt(((&W1)**2)*StdQuad**2+((&W2)**2)*StdIMA11**2);
   L95=Forecast-2*Std;
   U95=Forecast+2*Std;
run;

proc sgplot data=work.newfore;
   band  x=Date upper=U95 lower=L95;
   scatter x=Date y=Primary;
   series x=Date y=Forecast;
   refline "01SEP1992"d /axis=X;
run;
```

(Continued on the next page.)

```
/*---- Simple Average ----*/
%let W1=0.5;
%let W2=0.5;
data work.newfore;
   set work.regdata;
   Forecast=(&W1)*ForeQuad+(&W2)*ForeIMA11;
   Std=sqrt(((&W1)**2)*StdQuad**2+((&W2)**2)*StdIMA11**2);
   L95=Forecast-2*Std;
   U95=Forecast+2*Std;
run;

proc sgplot data=work.newfore;
   band  x=Date upper=U95 lower=L95;
   scatter x=Date y=Primary;
   series x=Date y=Forecast;
   refline "01SEP1992"d /axis=X;
run;
```

The estimated parameters from PROC REG are disappointing as indicated in the comment lines. Nonetheless, you can use them to construct forecasts and confidence limits. The confidence limits assume that the forecast random variables follow a normal distribution. The program approximates a 95% confidence limit by using two standard errors. In general, the standard errors are obtained using the following general formula, which is valid for any distribution:

$$Var(aX + bY) = a^2Var(X) + b^2Var(Y)$$

The program ends by using a simple average of the two forecasts.

2. Verify that a simple exponential smoothing model is similar to an ARIMA(0,1,1) model using the monthly lead production data.

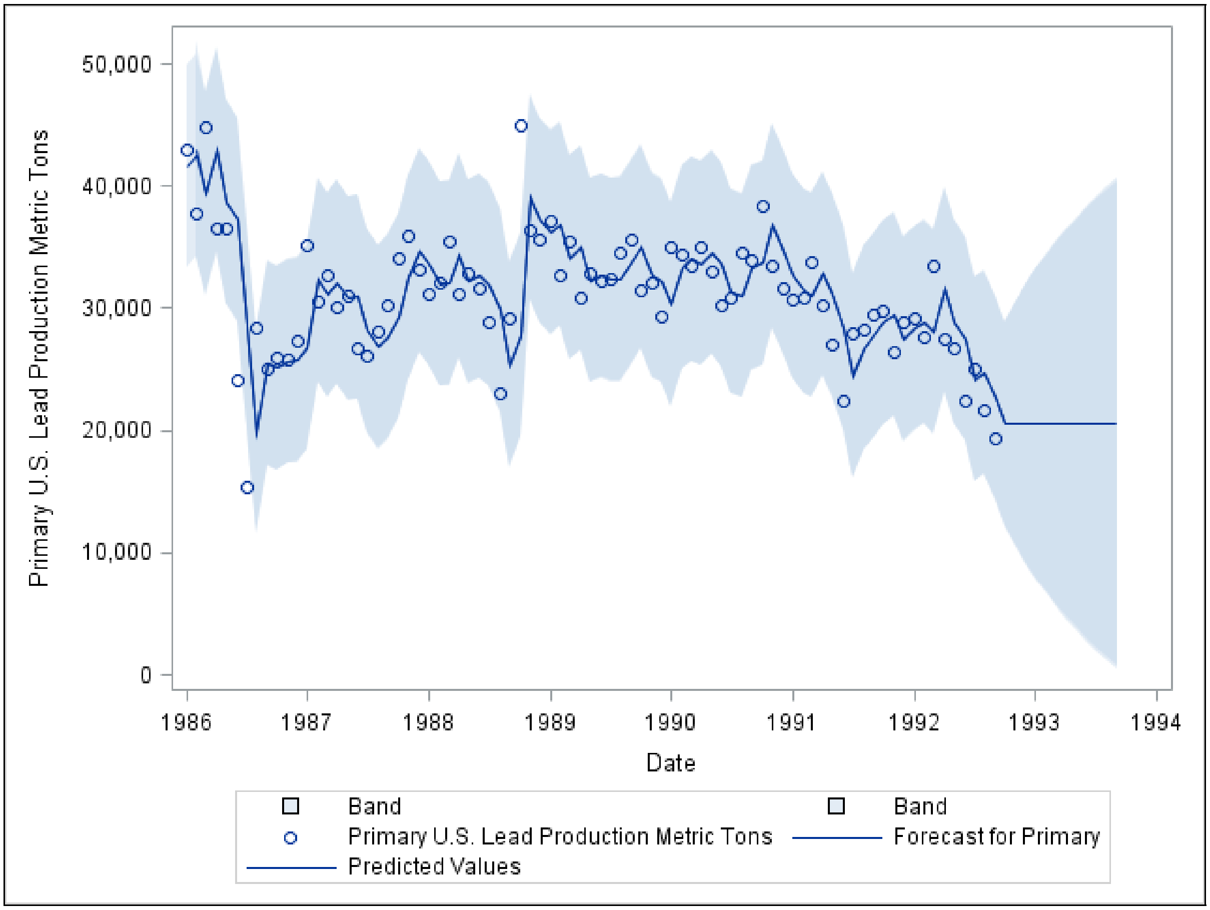
The ARIMA(0,1,1) model is fit using PROC ARIMA and the NOCONSTANT option. The PROC ESM is used with MODEL=SIMPLE. The forecasts are compared in an overlay plot and by using PROC COMPARE. The solution code follows:

```
proc arima data=sasuser.LeadMonth;
   identify var=Primary(1) noprint;
   estimate q=1 noconstant ml;
   forecast lead=12 id=Date interval=month out=ML_IMA11;
quit;

proc esm data=sasuser.LeadMonth
         outfor=work.esmfore(rename=(PREDICT=FORECAST))
         print=(estimates statistics summary)
         lead=12;
   id Date interval=month;
   forecast Primary / model=Simple;
run;
```

(Continued on the next page.)

```
data work.ML_Expo;
   set work.esmfore;
   rename LOWER=L95_Expo
          UPPER=U95_Expo
          FORECAST=ForeExpo;
run;

data work.both;
   merge work.ML_IMA11(keep=Date Primary Forecast L95 U95
           rename=(Forecast=ForeIMA11 L95=L95_IMA11 U95=U95_IMA11))
         work.ML_Expo(keep=Date ForeExpo L95_Expo U95_Expo);
   by Date;
run;

proc sgplot data=work.both;
   band  x=Date upper=U95_IMA11 lower=L95_IMA11 / transparency=0.6;
   band  x=Date upper=U95_Expo lower=L95_Expo / transparency=0.6;
   scatter x=Date y=Primary;
   series x=Date y=ForeIMA11;
   series x=Date y=ForeExpo;
run;

proc compare data=work.ML_IMA11 compare=work.esmfore;
   var FORECAST;
run;
```

The overlay plot makes the forecasts look identical, but PROC COMPARE shows small differences.



Primary U.S. Lead Production Metric Tons
50,000
40,000
30,000
20,000
10,000
0
1986
1987
1988
1989
1990
1991
1992
1993
1994
Date
Band
Band
Primary U.S. Lead Production Metric Tons
Forecast for Primary
Predicted Values

Chapter 4 Seasonal Models

4.1 Seasonal ARIMA Models

Objectives

- Review deterministic and stochastic seasonal components.
- Show how differencing and ARMA stationary models can be combined to create ARIMA models for time series with seasonality.
- Show how an ARMA stationary component can be combined with deterministic seasonal components.
- Describe how to combine the trend and seasonal components with an ARMA stationary component to model data with trend and seasonality.
- Illustrate how PROC ARIMA can be used to fit advanced models with trend and seasonal components.

3

Common Seasonal Components

- Deterministic
  - Seasonal dummy variables
  - Sinusoids at varying frequencies
- Stochastic
  - Differences of order S, S=seasonal period

4

Dummy Variables

- A dummy variable is an indicator variable.
- To indicate a specific time point, a dummy variable takes the value one for that time point.
- At all other time points, it takes the value zero.

Dummy variable for Hurricane Katrina event:

$$I_t = \begin{cases} 1 \text{ when } t = \text{Aug } 2005 \\ 0 \text{ otherwise} \end{cases}$$

5

Seasonal Dummy Variables

- For a time series with S seasons, there will be S dummy variables, one for each season.
- If the model has a constant term, only S-1 dummy variables are required.

6

continued...

Seasonal Dummy Variables

Monthly Data: I_{JAN} , I_{FEB} , ..., I_{DEC}

Daily Data: I_{SUN} , I_{MON} , ..., I_{SAT}

Quarterly Data: I_{Q1} , I_{Q2} , I_{Q3} , I_{Q4}

Dummy variable for June:

$$I_{JUN} = \begin{cases} 1 \text{ when } \ t = \text{June} \\ 0 \text{ otherwise} \end{cases}$$

7

continued...

Seasonal Dummy Variables

Model with No Constant Term

$$\boxed{\begin{aligned} &Y_t = \beta_{JAN} I_{JAN} + \beta_{FEB} I_{FEB} + \cdots + \beta_{DEC} I_{DEC} \\ &\beta_M = \text{effect of month } M \end{aligned}}$$

Model with Constant Term

$$\boxed{\begin{aligned} &Y_t = CONSTANT + \beta_{JAN} I_{JAN} + \cdots \beta_{NOV} I_{NOV} \\ &CONSTANT + \beta_{JAN} = \text{effect of } JAN \\ &\vdots \\ &CONSTANT + \beta_{NOV} = \text{effect of } NOV \\ &CONSTANT = \text{effect of } DEC \end{aligned}}$$

8

continued...

Seasonal Dummy Variables

```
data work.Air1990_2000;
   set LWFETSP.Air9001_0803
       (where=(Date<='31DEC2000'd));
   array Seas{*} MON1-MON11;
   retain MON1-MON11 .;
   if (MON1=.) then do index=1 to 11;
      Seas[index]=0;
   end;
   if (month(Date)<12) then do;
      Seas[month(Date)]=1;
      output;
      Seas[month(Date)]=0;
   end;
   else output;
   drop index;
run;
```

9

continued...

Seasonal Dummy Variables

Date	MON1	MON2	MON3	MON4	MON5	MON6	MON7	MON8	MON9	MON10	MON11
JAN1990	1	0	0	0	0	0	0	0	0	0	0
FEB1990	0	1	0	0	0	0	0	0	0	0	0
MAR1990	0	0	1	0	0	0	0	0	0	0	0
APR1990	0	0	0	1	0	0	0	0	0	0	0
MAY1990	0	0	0	0	1	0	0	0	0	0	0
JUN1990	0	0	0	0	0	1	0	0	0	0	0
JUL1990	0	0	0	0	0	0	1	0	0	0	0
AUG1990	0	0	0	0	0	0	0	1	0	0	0
SEP1990	0	0	0	0	0	0	0	0	1	0	0
OCT1990	0	0	0	0	0	0	0	0	0	1	0
NOV1990	0	0	0	0	0	0	0	0	0	0	1
DEC1990	0	0	0	0	0	0	0	0	0	0	0

10

continued...

Seasonal Dummy Variables

```
proc arima data=work.Air1990_2000;
   identify var=Passengers
            cross=(MON1 MON2 MON3 MON4
                   MON5 MON6 MON7 MON8
                   MON9 MON10 MON11)
            noprint;
   estimate q=(1)(12)
            input=(MON1 MON2 MON3 MON4
                   MON5 MON6 MON7 MON8
                   MON9 MON10 MON11)
            method=ml;
quit;
```

⚠ PROC ARIMA does not recognize the shorthand notation (MON1-MON11) for numbered variables.

11

Trigonometric Functions

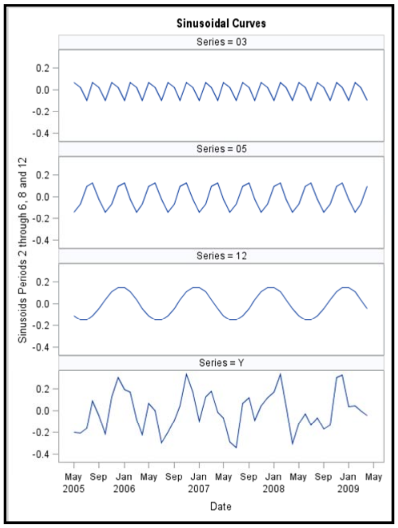
- Trigonometric functions can be used as ordinary regressors. The raw periodogram used in spectral analysis is derived by regressing the target with sinusoids.
- Trigonometric functions can be accommodated by creating trigonometric variables in the modeling data set defined into the future for forecasting purposes. The variables are then treated as ordinary regressors using any least squares regression procedure.

Example: $X_{1t} = \sin(2\pi\, t / S), X_{2t} = \cos(2\pi\, t / S)$

12

continued...

Trigonometric Functions



Target=Sum of sinusoids

13

continued...

Trigonometric Functions

```
data work.Air1990_2000;
   set work.Air1990_2000;
   retain TwoPi . Time 0;
   if (TwoPi=.) then TwoPi=2*constant("pi");
   Time+1;
   S4=sin(TwoPi*Time/4);
   C4=cos(TwoPi*Time/4);
   S12=sin(TwoPi*Time/12);
   C12=cos(TwoPi*Time/12);
run;
```

14

continued...

Trigonometric Functions

Date	Time	S4	C4	S12	C12
JAN1990	1	1.0000	0.0000	0.5000	0.8660
FEB1990	2	0.0000	-1.0000	0.8660	0.5000
MAR1990	3	-1.0000	-0.0000	1.0000	0.0000
APR1990	4	-0.0000	1.0000	0.8660	-0.5000
MAY1990	5	1.0000	0.0000	0.5000	-0.8660
JUN1990	6	0.0000	-1.0000	0.0000	-1.0000
JUL1990	7	-1.0000	-0.0000	-0.5000	-0.8660
AUG1990	8	-0.0000	1.0000	-0.8660	-0.5000
SEP1990	9	1.0000	0.0000	-1.0000	-0.0000
OCT1990	10	0.0000	-1.0000	-0.8660	0.5000
NOV1990	11	-1.0000	-0.0000	-0.5000	0.8660
DEC1990	12	-0.0000	1.0000	-0.0000	1.0000

15

continued...

Trigonometric Functions

```
proc arima data=work.Airline;
   identify var=Passengers
            cross=( Time S4 C4 S12 C12 );
   estimate p=(1)(12) q=(12)
            input=( Time S4 C4 S12 C12 )
            method=ml;
quit;
```

16

The Periodogram

The periodogram:

- Plots period against the amplitude of the sinusoid at that period. Frequencies can be used instead of periods.
- Is a sample estimate of a population *spectral density function*.
- Can be smoothed to provide a better estimate of the spectral density function.
- Helps identify sinusoids that can be used to approximate the series when the series is seasonal.
- Helps identify the dominate seasonal period of the data.
- Is obtained using PROC SPECTRA.

17

continued...

The Periodogram

```
PROC SPECTRA DATA=SAS-data-set
             OUT=SAS-data-set
             COEF P S <options>;
    VAR variables;
    WEIGHTS PARZEN|BART|TUKEY|TRUNCAT|QS
            <number number>;
RUN;
```

18

continued...

The option P requests that the periodogram be output. The option S requests that the smoothed periodogram be output. The option COEF requests the Fourier coefficients that are used to calculate the amplitudes in the periodogram. The WEIGHTS statement determines how to smooth the periodogram. The named smoothers represent specific methods for smoothing the periodogram to obtain sample spectral density functions. Parzen, Bartlett, and Tukey pioneered much of the research in estimating the spectral density function and have methods named after them.

The Periodogram

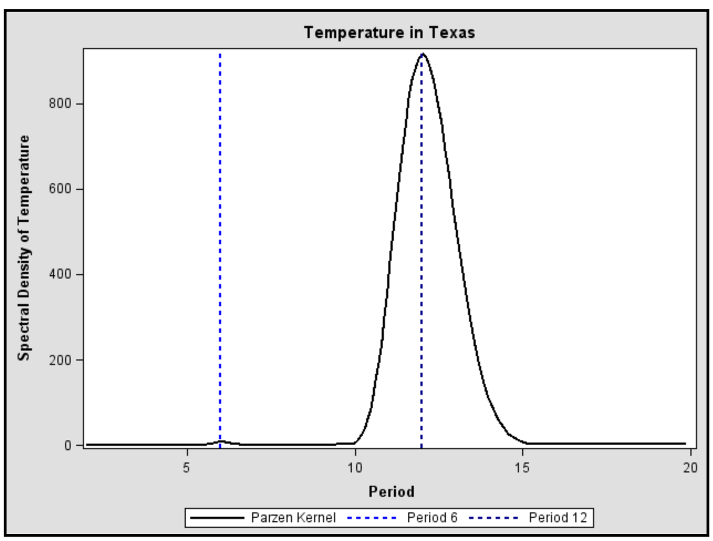
```
proc spectra data=work.USA_TX_NOAA
             out=work.Periodogram
             p s;
   var Temperature;
   weights Parzen 1 0.5;
run;
```

19

continued...

The above code produces an estimate of the spectral density function using a Parzen kernel estimator with the specified smoothing constants.

The Periodogram



20

The above plot of the smoothed periodogram shows that the series is dominated by seasonality at period 12, but there is also a very slight effect due to period 6.

The Box-Jenkins Seasonal Model

ARIMA(*p,d,q*)(*P,D,Q*)$_S$

- The *p*, *d*, and *q* are the orders of the nonseasonal terms of the model.
- *P* is the order of the seasonal autoregressive terms.
- *Q* is the order of the seasonal moving average terms.
- *D* is the order of the seasonal difference.
- *S* the length of the seasonal period.

ARIMA(*0,0,0*)(*1,1,1*)$_{12}$

$$(1-\Phi_1 B^{12})(1-B^{12})Y_t = \theta_0 + (1-\Theta_1 B^{12})\varepsilon_t$$

21

continued...

The Box-Jenkins Seasonal Model

ARIMA(*1,1,1*)(*1,1,1*)$_{12}$

$$(1-\phi_1 B^1)(1-\Phi_1 B^{12})(1-B^{12})(1-B^1)Y_t$$
$$= \theta_0 + (1-\theta_1 B^1)(1-\Theta_1 B^{12})\varepsilon_t$$

or equivalently

$$Z_t - \phi_1 Z_{t-1} - \Phi_1 Z_{t-12} + \phi_1 \Phi_1 Z_{t-13}$$
$$= \theta_0 + \varepsilon_t - \theta_1 \varepsilon_{t-1} - \Theta_1 \varepsilon_{t-12} + \theta_1 \Theta_1 \varepsilon_{t-13}$$

where

$$Z_t = Y_t - Y_{t-1} - Y_{t-12} + Y_{t-13}$$

22

Box-Jenkins Methodology for the Seasonal Model

Identify

Determine (*p*,*d*,*q*)(*P*,*D*,*Q*)

Estimate

Estimate Using Maximum Likelihood

Forecast

Forecast Using Finite Memory Forecasting

23

The General Box-Jenkins Seasonal Model

ARIMA(*p*,*d*,*q*)(*P*,*D*,*Q*)$_S$

$$(1-\phi_1 B-\cdots-\phi_p B^p)(1-\Phi_1 B-\cdots-\Phi_P B^P)(1-B)^d(1-B^S)^D(Y_t-\mu)$$
$$=(1-\theta_1 B-\cdots-\theta_q B^q)(1-\Theta_1 B-\cdots-\Theta_Q B^Q)\varepsilon_t$$

ARIMA Model with Deterministic Trend and Seasonal Terms

$$(1-\phi_1 B-\cdots-\phi_p B^p)(1-\Phi_1 B-\cdots-\Phi_P B^P)(Y_t-\mu_t)$$
$$=(1-\theta_1 B-\cdots-\theta_q B^q)(1-\Theta_1 B-\cdots-\Theta_Q B^Q)\varepsilon_t$$
$$\mu_t=f(t)+g(t)$$
$$f(t)=\text{Trend},\ g(t)=\text{Seasonal}$$

24

Differencing for Trend and Seasonality

$(1-B^s)(1-B)Y_t = \varepsilon_t$ ← Backshift Notation

$(1-B-B^s+B^{s+1})Y_t = \varepsilon_t$ ← Polynomial Multiplication

$Y_t - Y_{t-1} - Y_{t-s} + Y_{t-s-1} = \varepsilon_t$ ← Apply Backshift Operator

$Y_t = Y_{t-1} + Y_{t-s} - Y_{t-s-1} + \varepsilon_t$ ← Model

$\hat{Y}_t = Y_{t-1} + Y_{t-s} - Y_{t-s-1}$ ← Forecast Equation

Forecast depends on value s+1 time points in the past.

25

The General Box-Jenkins Seasonal Model

ARIMA(*0,1,1*)(*0,1,1*)$_{12}$

```
proc arima data=work.Airline;
   identify var=Passengers(1 12);
   estimate q=(1)(12) method=ml;
   forecast id=Date interval=month
            lead=24;
quit;
```

26

continued...

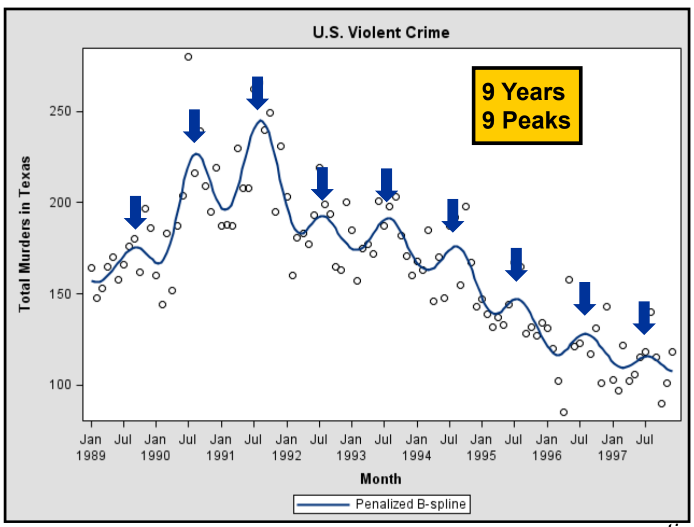
The General Box-Jenkins Seasonal Model

ARIMA with Linear Trend and Sinusoids

```
data work.Airline;
   set work.Airline end=lastobs;
<...omitted code...>
   C4=cos(TwoPi*Time/4);
   S12=sin(TwoPi*Time/12);
<...omitted code...>
proc arima data=work.Airline;
   identify var=Passengers
            cross=( Time S4 C4 S12 C12 );
   estimate p=(1)(12) q=(12)
            input=( Time S4 C4 S12 C12 )
            method=ml;
   forecast id=Date interval=month lead=24;
quit;
```

27

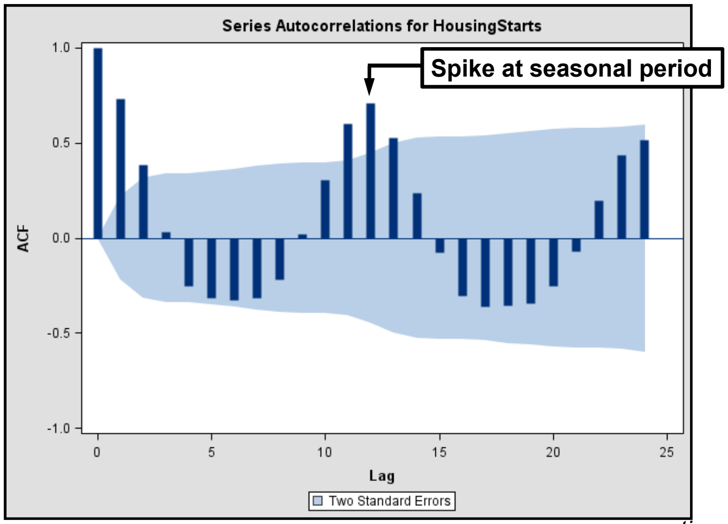
Diagnosing Seasonality



28

continued...

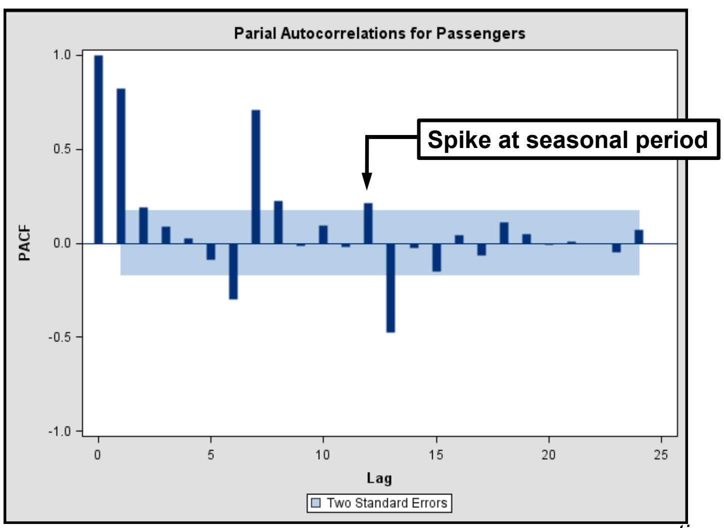
Diagnosing Seasonality



29

continued...

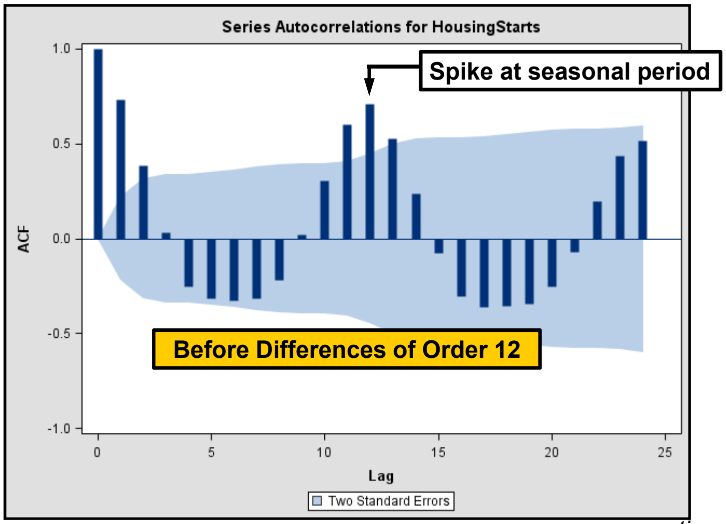
Diagnosing Seasonality



30

continued...

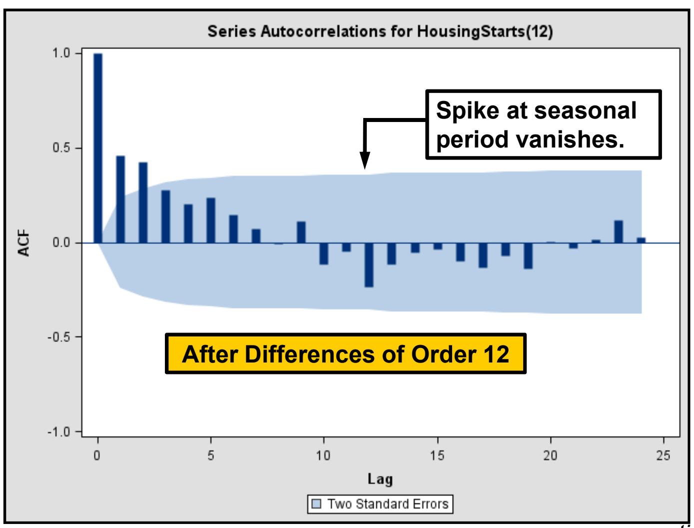
Diagnosing Seasonality



31

continued...

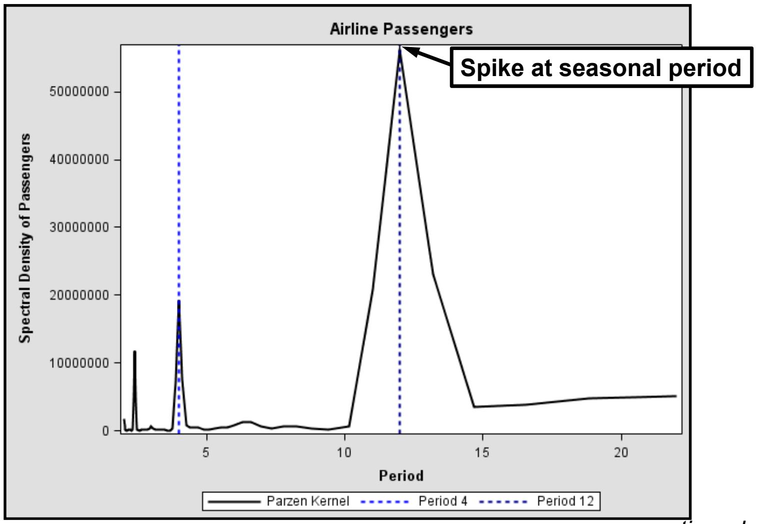
Diagnosing Seasonality



32

continued...

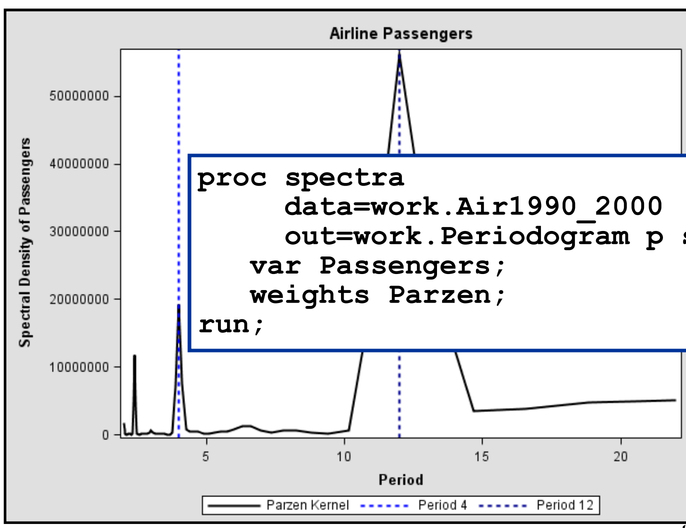
Diagnosing Seasonality



33

continued...

Diagnosing Seasonality




```
proc spectra
     data=work.Air1990_2000
     out=work.Periodogram p s;
   var Passengers;
   weights Parzen;
run;
```

34

continued...

Diagnosing Seasonality

Seasonal Augmented Dickey-Fuller Unit Root Tests

Type	Lags	Rho	Pr < Rho	Tau	Pr < Tau
Zero Mean	0	1.5233	0.6227	9.17	0.9999
	1	3.5033	0.7389	3.19	0.9987
	2	2.8359	0.6983	1.60	0.9455
	3	1.4959	0.6147	0.75	0.7833
	4	1.4304	0.6083	0.71	0.7709
	5	1.4008	0.6040	0.67	0.7585
Single Mean	0	-1.8825	0.4743	-1.06	0.2127
	1	0.5239	0.6185	0.26	0.6852
	2	0.5423	0.6181	0.26	0.6824
	3	0.4431	0.6096	0.21	0.6652
			0.5986	0.15	0.6409
			0.6075	0.22	0.6650

p>0.05 implies failure to reject the null hypothesis of a seasonal unit root.

continued...

Exploring Average Monthly Temperature in Texas

This demonstration illustrates the modeling of trend and seasonality using the NOAA temperature data for the state of Texas.

The code for this demonstration can be found in **Dem4_01Temperature.sas**. The data was obtained from the National Climatic Data Center Web site of the NOAA.

http://www.ncdc.noaa.gov/oa/ncdc.html

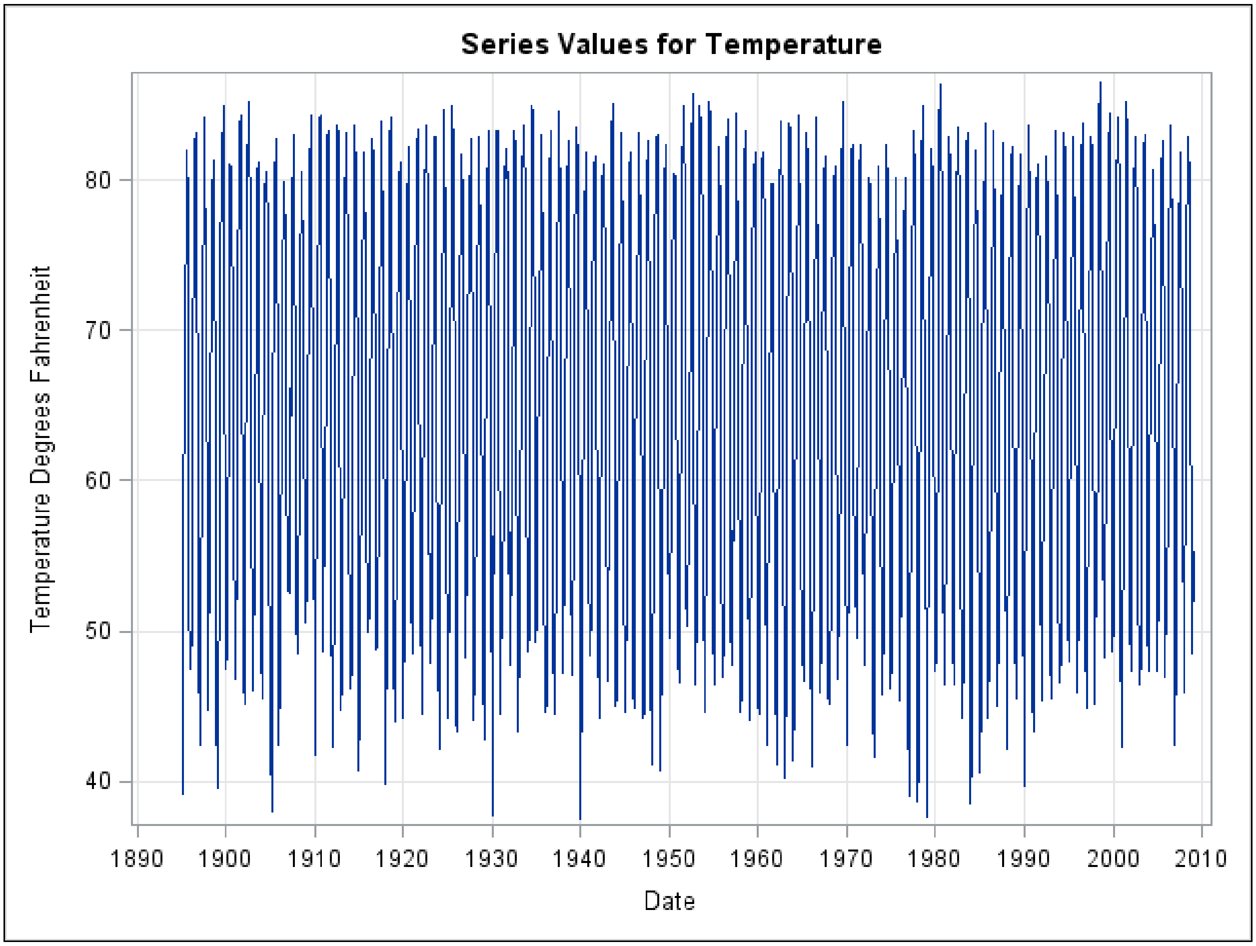
The SAS data set **SASUSER.USA_TX_NOAA** contains several climate variables for the state of Texas in the United States. This demonstration focuses on temperature values in Fahrenheit from January 1895, to February 2009. Temperature data sets gathered over long periods can help determine the effects of global climate change.

The primary purpose of this demonstration is to illustrate strategies for problems that arise when modeling very long time series.

The following program provides the primary diagnostic plots for model determination:

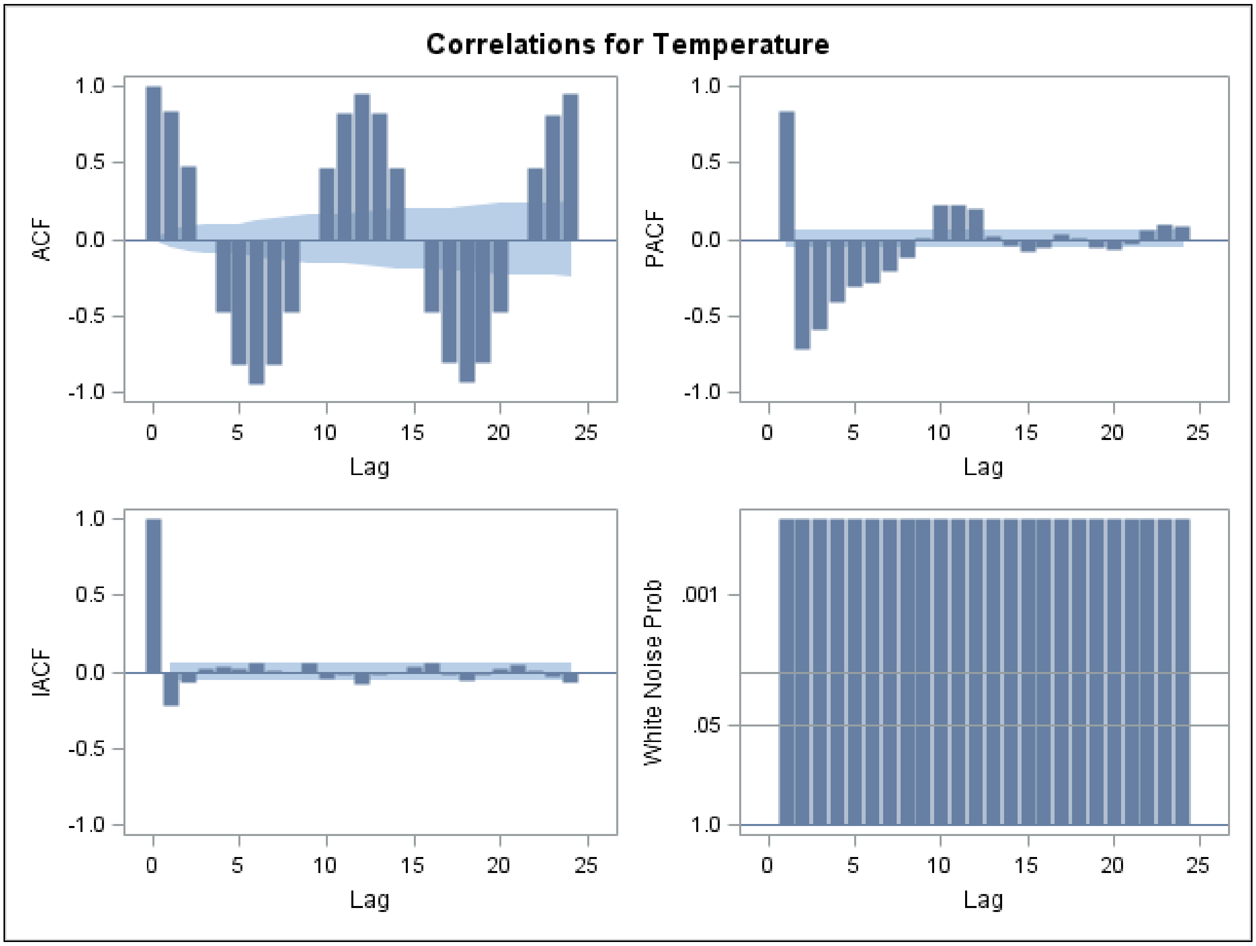
```
proc timeseries data=sasuser.usa_tx_noaa
                out=work.temp
                outdecomp=work.decomp
                plot=(series corr acf pacf iacf wn decomp tc sc)
                seasonality=12;
   id Date interval=month;
   var Temperature;
   decomp tcc sc / mode=mult;
run;
```

The diagnostic plots follow:



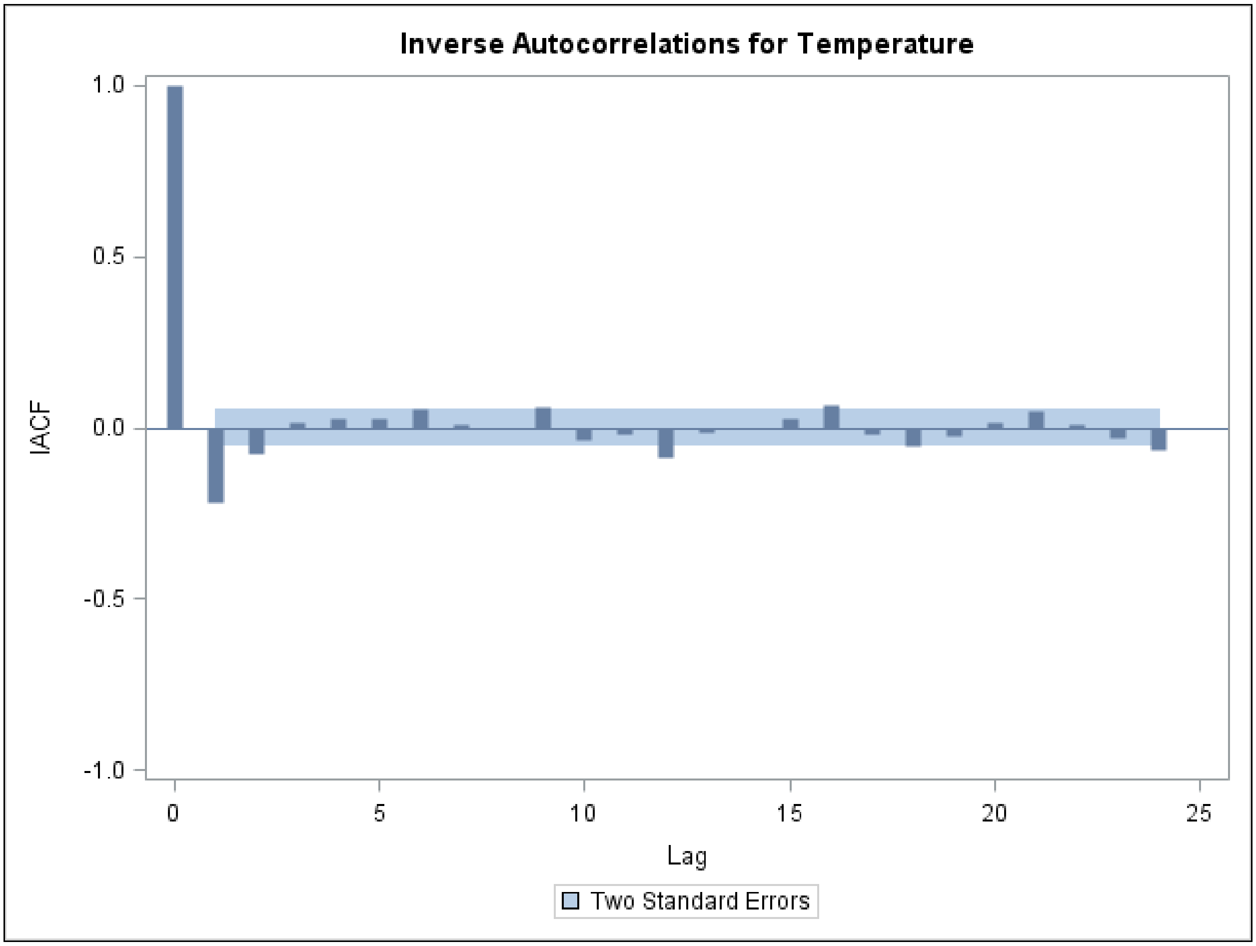
The series plot illustrates the problem with plotting long time series.

A proposed solution to this “inkblot” problem appears below.

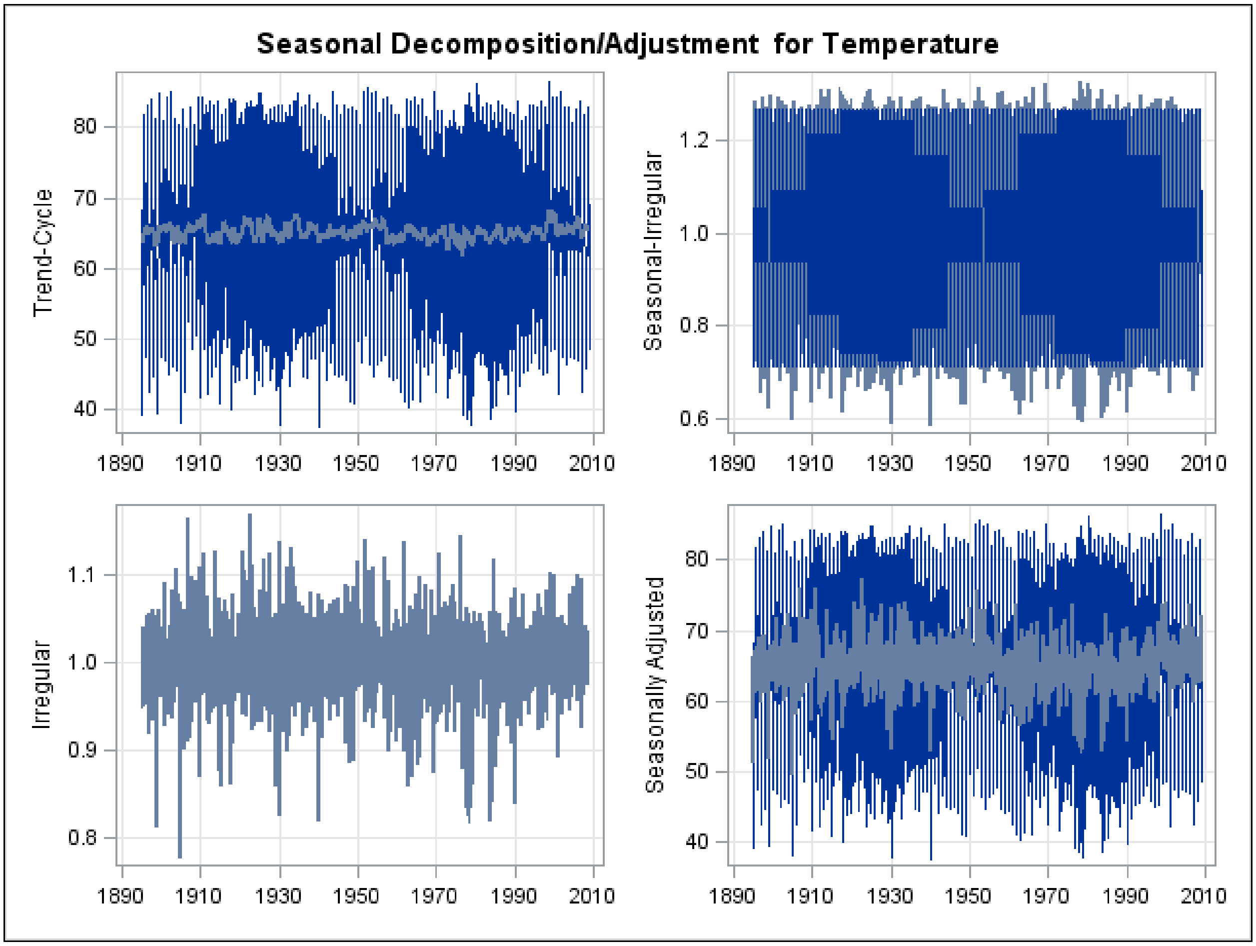


Spikes at lag 12 emphasize the seasonal nature of temperature data.

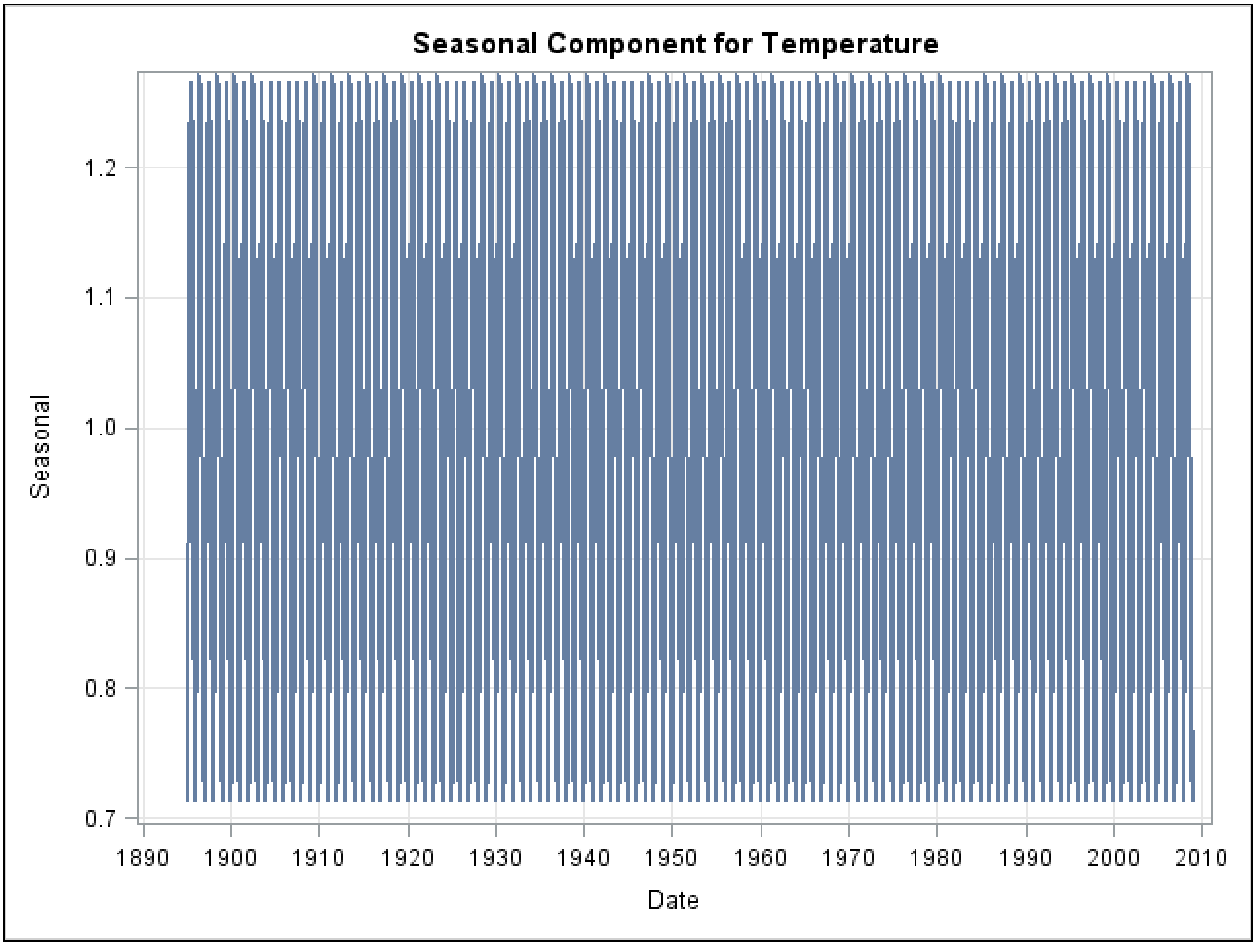
The enlarged plots are available, but only the enlarged IACF need be viewed. Spikes in the other plots are clearly visible in the correlation panel.



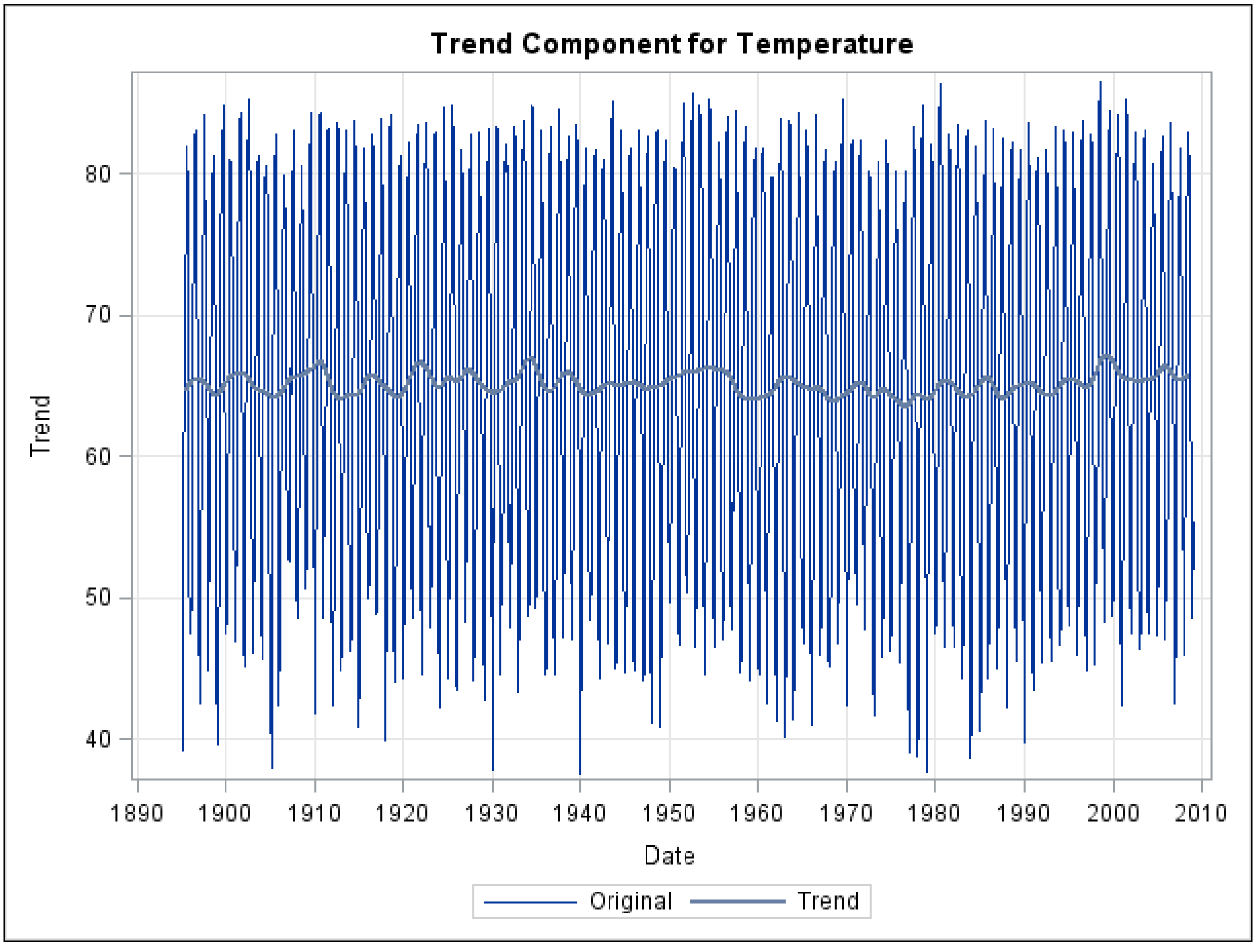
There is a small spike at lag 12 and at lag 16, in addition to the spikes at lags 1 and 2.



The decomposition plots suffer from the high density of time points.



The seasonal plot makes it clear that the data are seasonal, with monthly effects ranging from about 0.7 to about 1.25. On the other hand, the plot is too dense to get a picture of the nature of the seasonality. This problem is resolved in a later demonstration that examines the airline passenger data where the nature of seasonal fluctuations is less obvious.

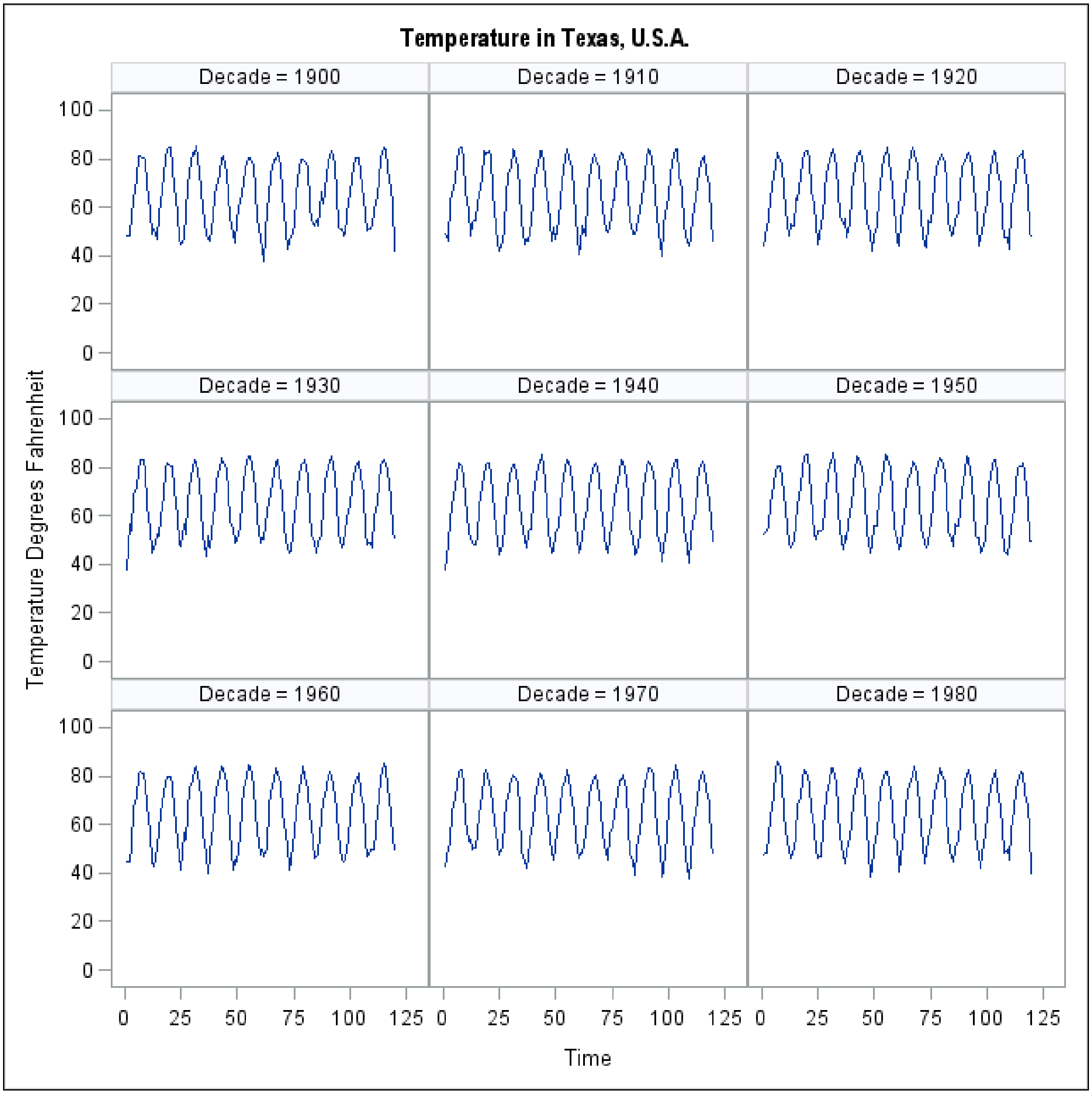


The moving average smoother shows a flat but erratic pattern for the trend.

One way to visualize a long series is to break the series into pieces, and also to use a better aspect ratio. The following code creates time series plots for each decade:

```
data work.temppanel;
   set sasuser.USA_TX_NOAA;
   Decade=10*floor(year(Date)/10);
   if (year(Date)<1900) then delete;
   keep Date Temperature Decade;
run;

data work.temppanel;
   set work.temppanel;
   by Decade;
   retain Time 0;
   if (first.Decade) then Time=0;
   Time+1;
run;

proc sgpanel data=work.temppanel;
   panelby Decade / rows=3 columns=3;
   series x=Time y=Temperature;
   rowaxis min=0 max=100;
run;
```

The first panel is shown below. The last panel only has two series and can be viewed by running the program.

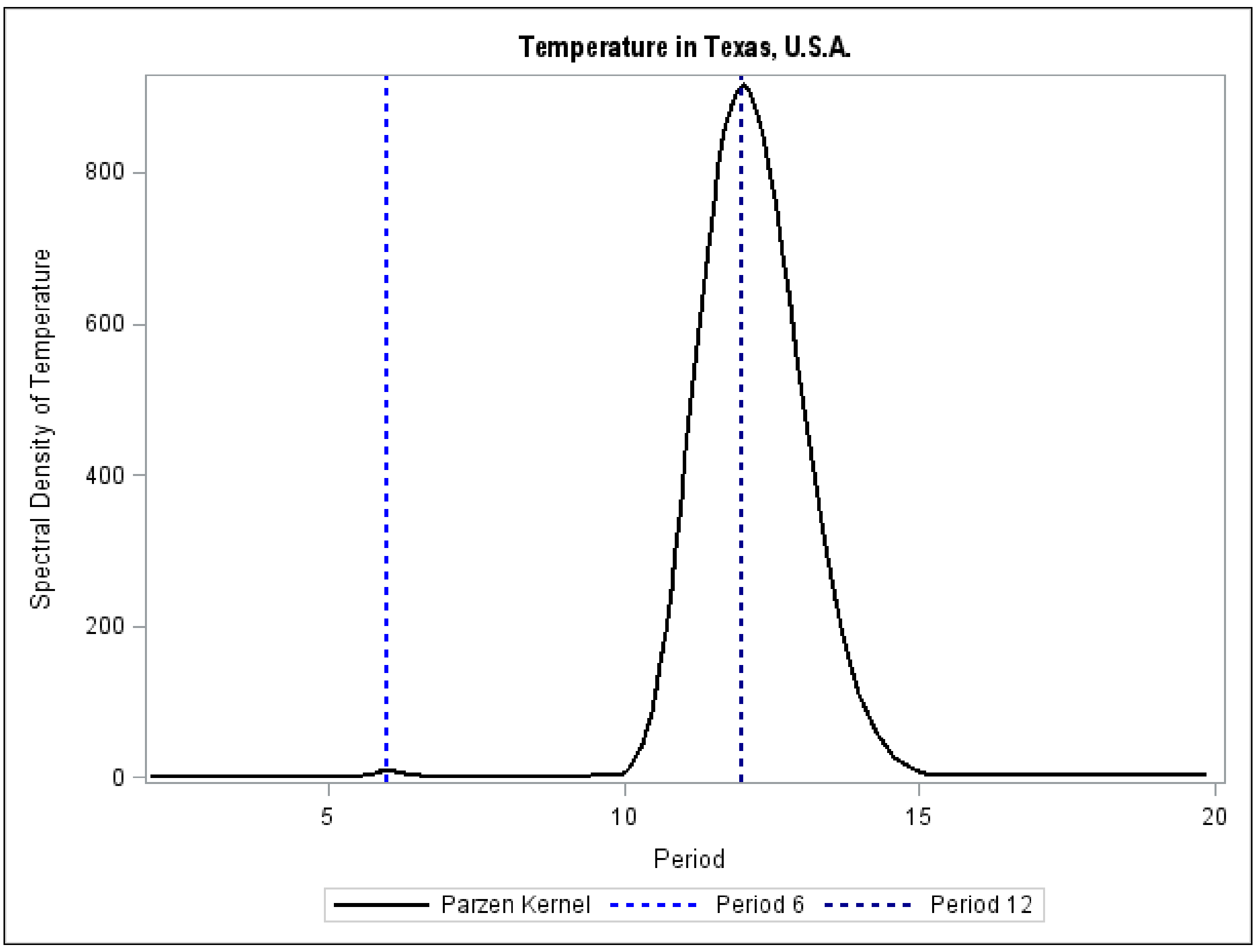


The series is strongly seasonal and very stable across decades. By using the same axes, it is also clear that no upward or downward trend is present in the series.

PROC SPECTRA can be used to check for multiple sinusoidal factors.

```
proc spectra data=sasuser.USA_TX_NOAA
             out=work.Periodogram p s;
   var Temperature;
   weights Parzen 1 0.5;
run;

proc sgplot data=work.Periodogram(where=(2<=PERIOD<=20));
   series x=PERIOD y=S_01 /
          lineattrs=GraphPrediction(pattern=1 color=black)
          legendlabel="Parzen Kernel" name="series1";
   refline 6 / axis=x
           lineattrs=GraphPrediction(pattern=2 color=blue)
           legendlabel="Period 6" name="series2";
   refline 12 / axis=x
           lineattrs=GraphPrediction(pattern=2 color=darkblue)
           legendlabel="Period 12" name="series3";
   keylegend "series1" "series2" "series3" /
             location=outside position=bottom;
run;
```

The plot follows:



The evidence is compelling and overwhelming, but one purpose of this demonstration is to reveal problems that arise with long time series. The following code requests an augmented Dickey-Fuller test:

```
title2 font=&COURSEFONT color=black "Augmented Dickey Fuller Test";
proc arima data=sasuser.USA_TX_NOAA;
   identify var=Temperature stationarity=(adf=(0 1 2));
   identify var=Temperature stationarity=(adf=(0 1 2) dlag=12);
quit;
```

The trend test implies a stationary trend, but the seasonal test also implies stationarity.

Seasonal Augmented Dickey-Fuller Unit Root Tests					
Type	Lags	Rho	Pr < Rho	Tau	Pr < Tau
Zero Mean	0	-1.7834	0.4178	-0.87	0.2348
	1	-2.5931	0.3619	-1.08	0.1757
	2	-3.0963	0.3296	-1.21	0.1428
Single Mean	0	-55.6396	0.0013	-5.44	<.0001
	1	-70.5150	0.0013	-6.10	<.0001
	2	-70.9479	0.0013	-6.06	<.0001

The apparent flaw in the ADF test is revealed by considering estimates of the lag 12 coefficient, which under the null hypothesis, is exactly one.

```
data work.check;
   set sasuser.USA_TX_NOAA;
   LagTemp12=lag12(Temperature);
   keep Date Temperature LagTemp12;
run;

proc reg data=work.check;
   model Temperature=LagTemp12;
run;
```

The estimated lag 12 coefficient under ordinary least squares is given in the following table:

Parameter Estimates						
Variable	Label	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|
Intercept	Intercept	1	2.68611	0.50016	5.37	<.0001
LagTemp12		1	0.95903	0.00754	127.27	<.0001

The estimate is very close to the null hypothesis value of 1, but with such a large sample size, it apparently is not close enough!

You can also investigate the maximum likelihood estimate from PROC ARIMA.

```
proc arima data=work.check;
   identify var=Temperature noprint;
   estimate p=(12) ml;
quit;
```

The estimate table follows:

Maximum Likelihood Estimation					
Parameter	**Estimate**	**Standard Error**	**t Value**	**Approx Pr > |t|**	**Lag**
MU	64.93911	2.09526	30.99	<.0001	0
AR1,1	0.96065	0.0070223	136.80	<.0001	12

The estimate is again close to the null hypothesis value. While it is tempting to examine the standard error of the estimate in the context of the usual t-test, recall that the ADF test is necessary because the t-distribution is not valid for testing deviations from unity.

Is it possible that the sample size is ***too*** big? Consider the following code that looks at two different subsets of the original data that are much smaller, covering only a few years rather than a century:

```
proc arima data=work.check(where=('01JAN1950'd<=Date<='31DEC1953'd));
   identify var=Temperature stationarity=(adf=(0 1 2) dlag=12);
quit;

proc arima data=work.check(where=('01JAN1972'd<=Date<='31DEC1976'd));
   identify var=Temperature stationarity=(adf=(0 1 2) dlag=12);
quit;
```

You can verify that for both subsets, the seasonal ADF test concludes that a unit root exists with respect to DLAG=12, that is, evidence suggests that the series is seasonal. The first subset yields the following ADF table:

Seasonal Augmented Dickey-Fuller Unit Root Tests					
Type	**Lags**	**Rho**	**Pr < Rho**	**Tau**	**Pr < Tau**
Zero Mean	0	0.0871	0.5296	0.31	0.6401
	1	0.1550	0.5327	0.45	0.6894
	2	0.1860	0.5333	0.67	0.7606
Single Mean	0	0.5854	0.6240	0.39	0.7281
	1	-0.2425	0.5707	-0.14	0.5365
	2	-0.2418	0.5689	-0.15	0.5299

The results look more reasonable with consistent evidence that the series is seasonal.

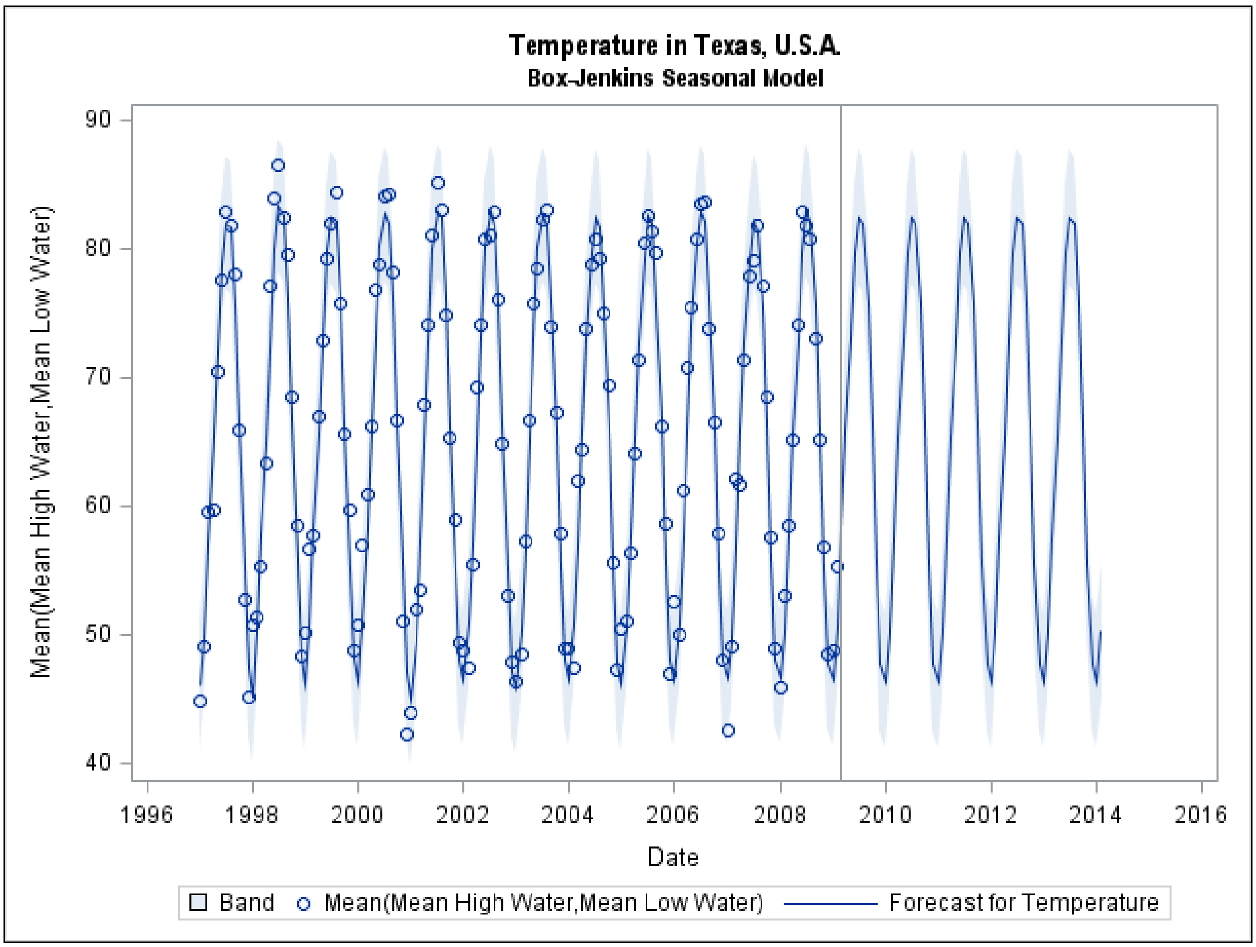
Often the most challenging problem in forecasting is having too little data, so practitioners sometimes forget the problems that can be encountered when there is a wealth of data. In data mining, large data sets are the norm, and predictive modeling professionals know that techniques such as *Bayes correction* might be warranted due to the high precision of estimators. Strategies like Bayes correction highlight the problem with blindly applying significance levels, like 5%.

While sample size might be the cause of the false inference, another factor is that the ADF implemented by PROC ARIMA uses the model

$$Y_t - Y_{t-12} = \rho Y_{t-12} + \varepsilon_t .$$

When $\rho = 0$, the model is nonstationary with a seasonal unit root. SAS PROC ARIMA does not provide an alternative formulation with seasonal dummy variables, analogous to the ADF trend test, which accomodates linear trend. Therefore, you might have to resort to heuristics to evaluate seasonality. Obviously, you can fit a model with seasonal components and evaluate the individual tests for each parameter to deduce the seasonal nature of the data. This is the approach that is usually followed in this course when seasonal differencing does not seem to be adequate.

The source code for this demonstration continues with details related to forecasting. The forecast plot for the final model follows:



The airline data examined at the end of the chapter provides a better instructional example for forecasting with trend and seasonality.

4.01 Multiple Choice Poll

A time series recorded every month has a seasonal component. Which of the following diagnostics indicate seasonality?

a. A significant PACF value at lag 12.
b. A sample spectral density function that exhibits a peak at period 12.
c. An augmented Dickey-Fuller seasonal test exhibits p-values larger than 0.1.
d. All of the above.

38

4.2 Alternatives to PROC ARIMA for Fitting Seasonal Models

Objectives

- Describe the use of PROC FORECAST and stepwise autoregression for time series with trend and seasonality.
- Introduce exponential smoothing models having seasonal components.
- Describe how to use PROC ESM to fit exponential smoothing models for seasonal time series.

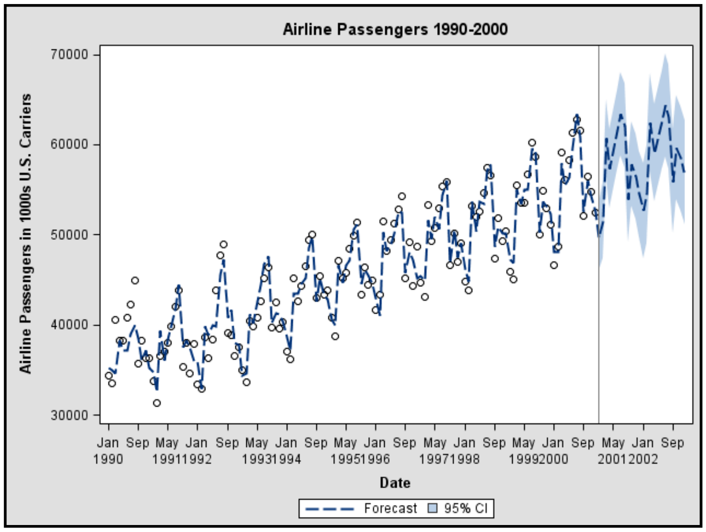
41

PROC FORECAST and Seasonal Data

```
proc forecast data=work.Air1990_2000
              out=work.forecast
              outall
              outest=work.estimates
              outestall
              method=stepar
              ar=25
              slentry=0.20
              slstay=0.10
              trend=2
              interval=MONTH
              lead=24;
   var Passengers;
   id Date;
run;
```

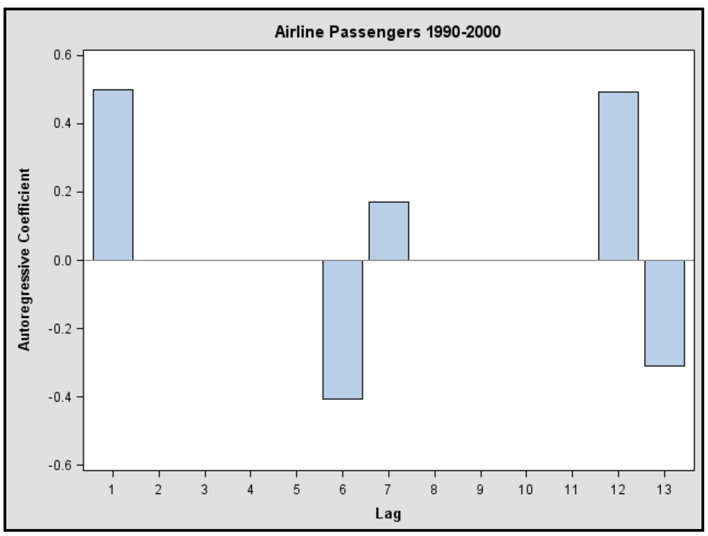
42

Stepwise Autoregressive Forecasts, Trend=2



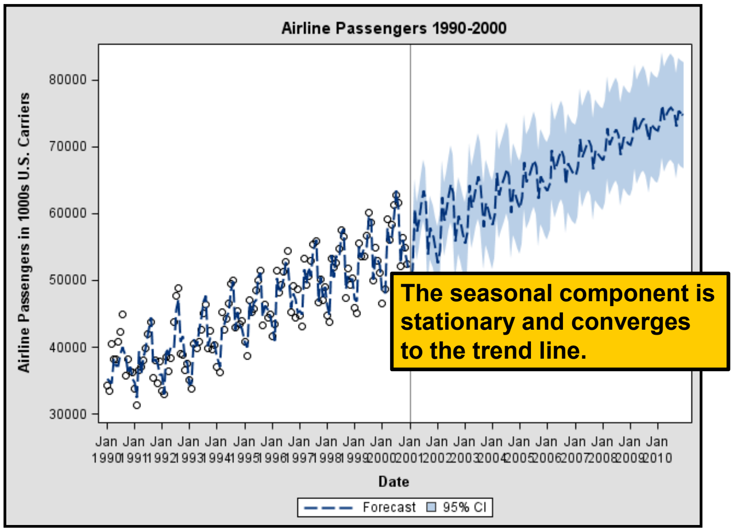
43

Autoregressive Coefficients



44

Stepwise Autoregressive Forecasts, Trend=2



45

Exponential Smoothing Premise (Review)

- Weighted averages of past values can produce good forecasts of the future.
- The weights should emphasize the most recent data.
- Forecasting should only require a few parameters.
- Forecast equations should be simple and easy to implement.

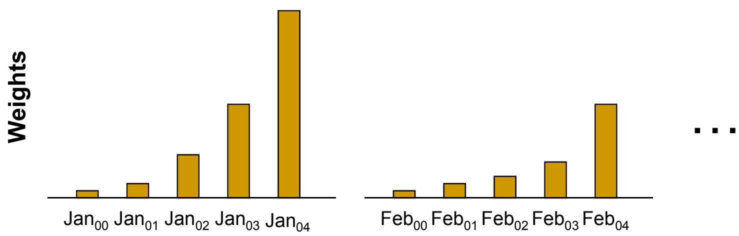
46

Exponential Smoothing Models (Review)

- Models for time series with trend:
  - Simple exponential smoothing
  - Double (Brown) exponential smoothing
  - Linear (Holt) exponential smoothing
  - Damped-trend exponential smoothing
- Models for time series with seasonality:
  - Seasonal exponential smoothing
- Models for time series with trend and seasonality
  - Winters additive exponential smoothing
  - Winters multiplicative exponential smoothing

47

Exponential Smoothing for Seasonal Data



Weights decay with respect to the seasonal factor.

48

Smoothing Weights

ω	**Level smoothing weight**
γ	**Trend smoothing weight**
δ	**Seasonal smoothing weight**
ϕ	**Trend damping weight**

The choice of Greek letter is arbitrary. The software uses names rather than Greek symbols.

49

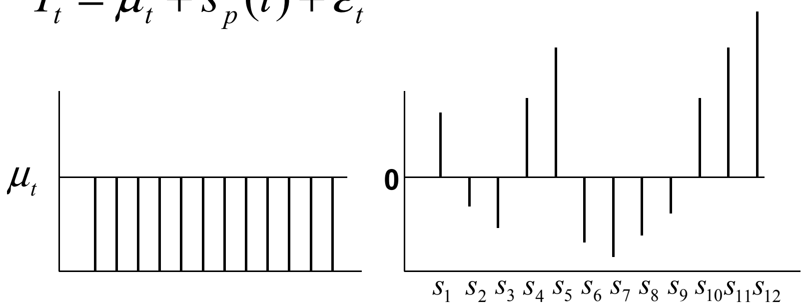
ESM Parameters and Keywords

ESM	Parameters	Model = Keyword
Simple	ω	`SIMPLE`
Double	ω	`DOUBLE`
Linear (Holt)	ω, γ	`LINEAR`
Damped-Trend	ω, γ, ϕ	`DAMPTREND`
Seasonal	ω, δ	`SEASONAL`
Additive Winters	ω, γ, δ	`ADDWINTERS`
Multiplicative Winters	ω, γ, δ	`WINTERS`

50

Seasonal Exponential Smoothing

$$Y_t = \mu_t + s_p(t) + \varepsilon_t$$

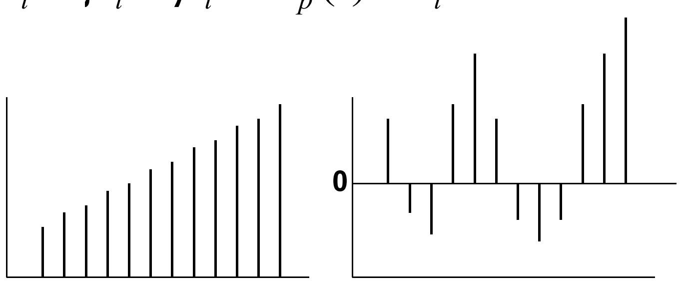


Level + Seasonal

51

Winters Method — Additive

$$Y_t = \mu_t + \beta_t t + s_p(t) + \varepsilon_t$$

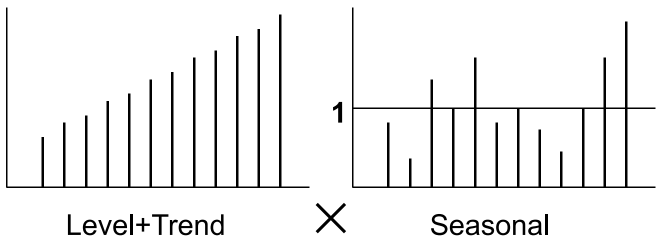


Level+Trend + Seasonal

52

Winters Method — Multiplicative

$$Y_t = (\mu_t + \beta_t t)s_p(t) + \varepsilon_t$$



53

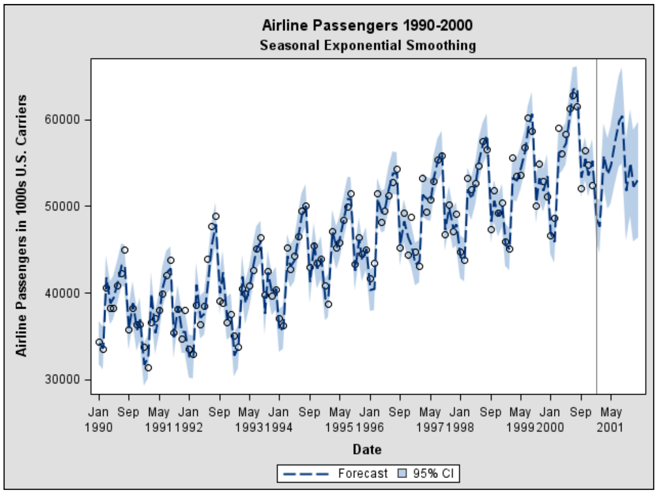
Exponential Smoothing Models for Airline Passengers

```
proc esm data=work.Air1990_2000
     out=work.out
     outfor=work.forecast
     outest=work.estimates
     outstat=work.stats
     outsum=work.summary
     lead=12
     print=(ESTIMATES STATISTICS SUMMARY);
   id Date interval=month;
   forecast Passengers / model=seasonal;
run;
```

54

continued...

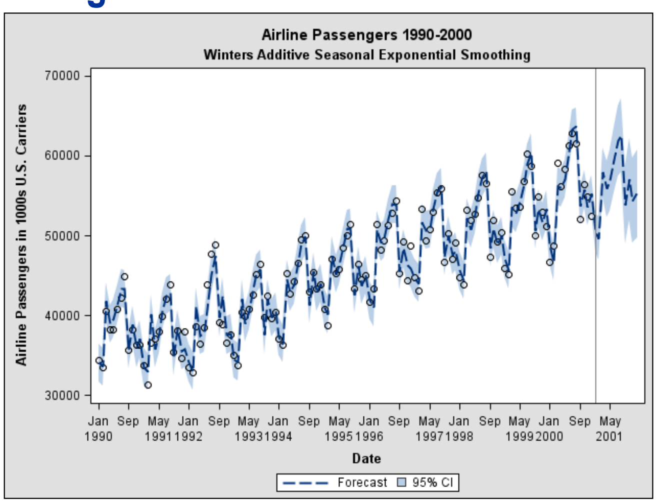
Exponential Smoothing Models for Airline Passengers



55

continued...

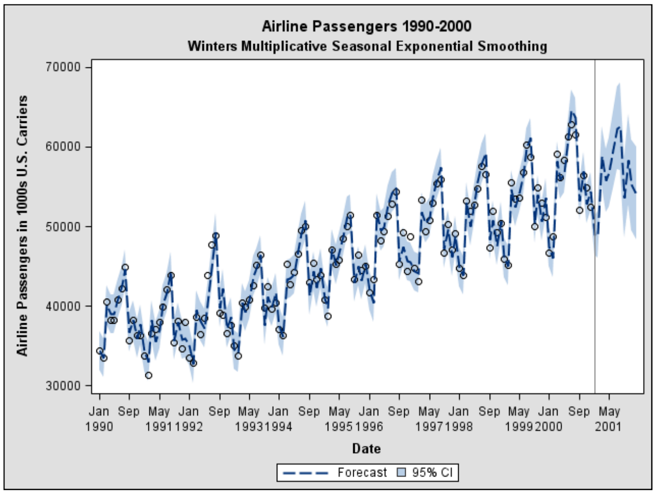
Exponential Smoothing Models for Airline Passengers



56

continued...

Exponential Smoothing Models for Airline Passengers



57

continued...

Exponential Smoothing Models for Airline Passengers

Airline Passengers 1990-2000

Exponential Smoothing Models Goodness-of-fit

Model	MAPE	RMSE	AIC	SBC
AddWinters	8.543	3449.6	1325.7	1332.8
Seasonal	8.641	3467.9	1324.5	1329.3
Winters	9.927	3796.1	1341.2	1348.3
Linear	10.381	4207.0	1355.8	1360.6
DampTrend	10.390	4207.8	1357.8	1365.0
Simple	10.429	4214.8	1354.1	1356.5
Double	11.592	4593.0	1368.0	1370.4

58

4.3 Forecasting the Airline Passengers Data

Objectives

- Diagnose trend and seasonality for the airline data.
- Illustrate how PROC ARIMA can be used to fit advanced models with trend and seasonal components.

60

Forecasting the Airline Passengers Time Series, 1990–2000

This demonstration illustrates how to use PROC ARIMA, PROC FORECAST, and PROC ESM to derive a forecast model for the U.S. Department of Transportation airline data, 1990–2000. The time series before the events of September 11, 2001, is used to illustrate the concepts of this chapter. The full time series will be examined in the next chapter.

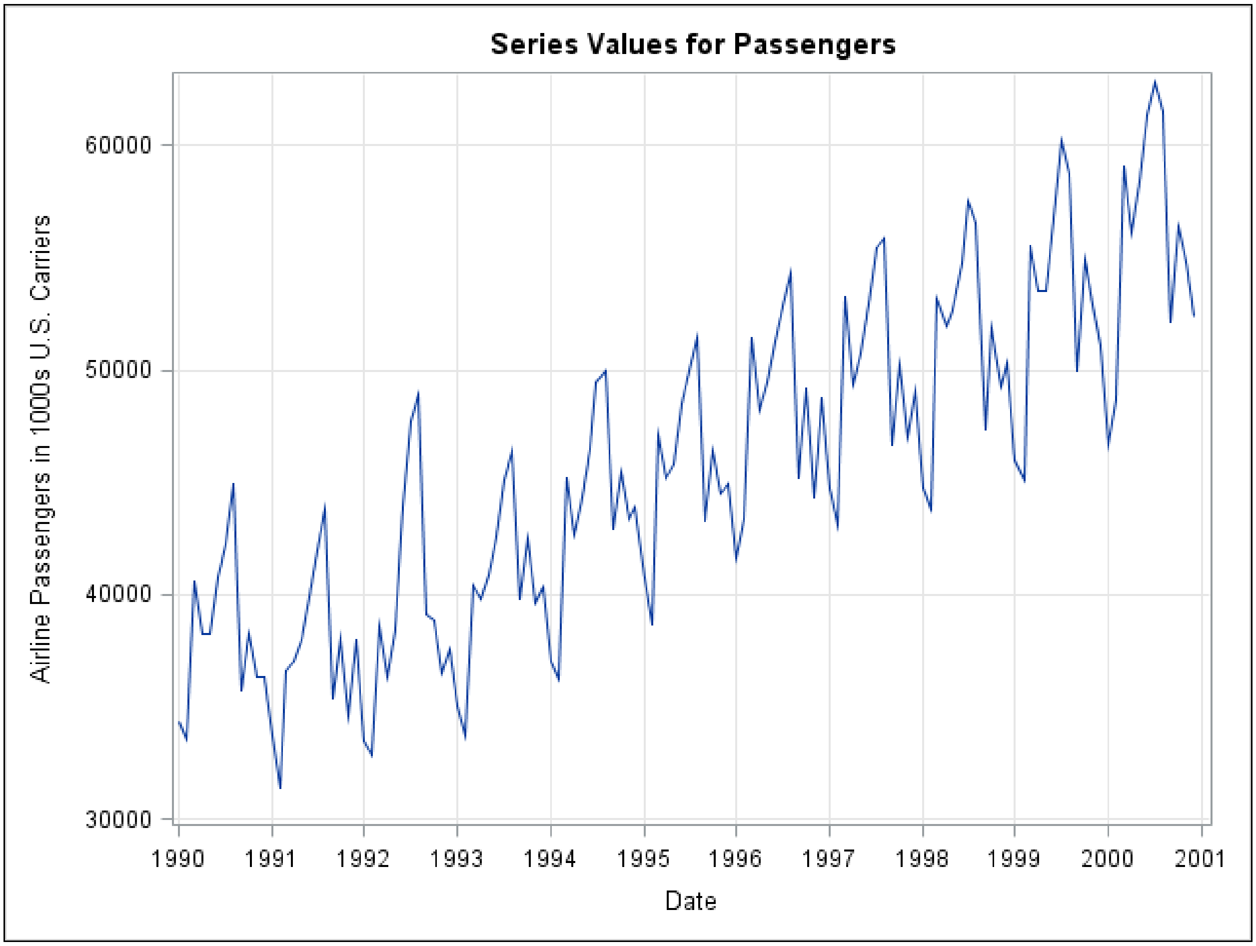
The code for this demonstration can be found in **Demo4_02Airline.sas**. The following code creates a working copy of the time series restricted to 1990–2000:

```
data work.Air1990_2000;
   set sasuser.usairlines
       (where=(Date<='31DEC2000'd));
run;
```

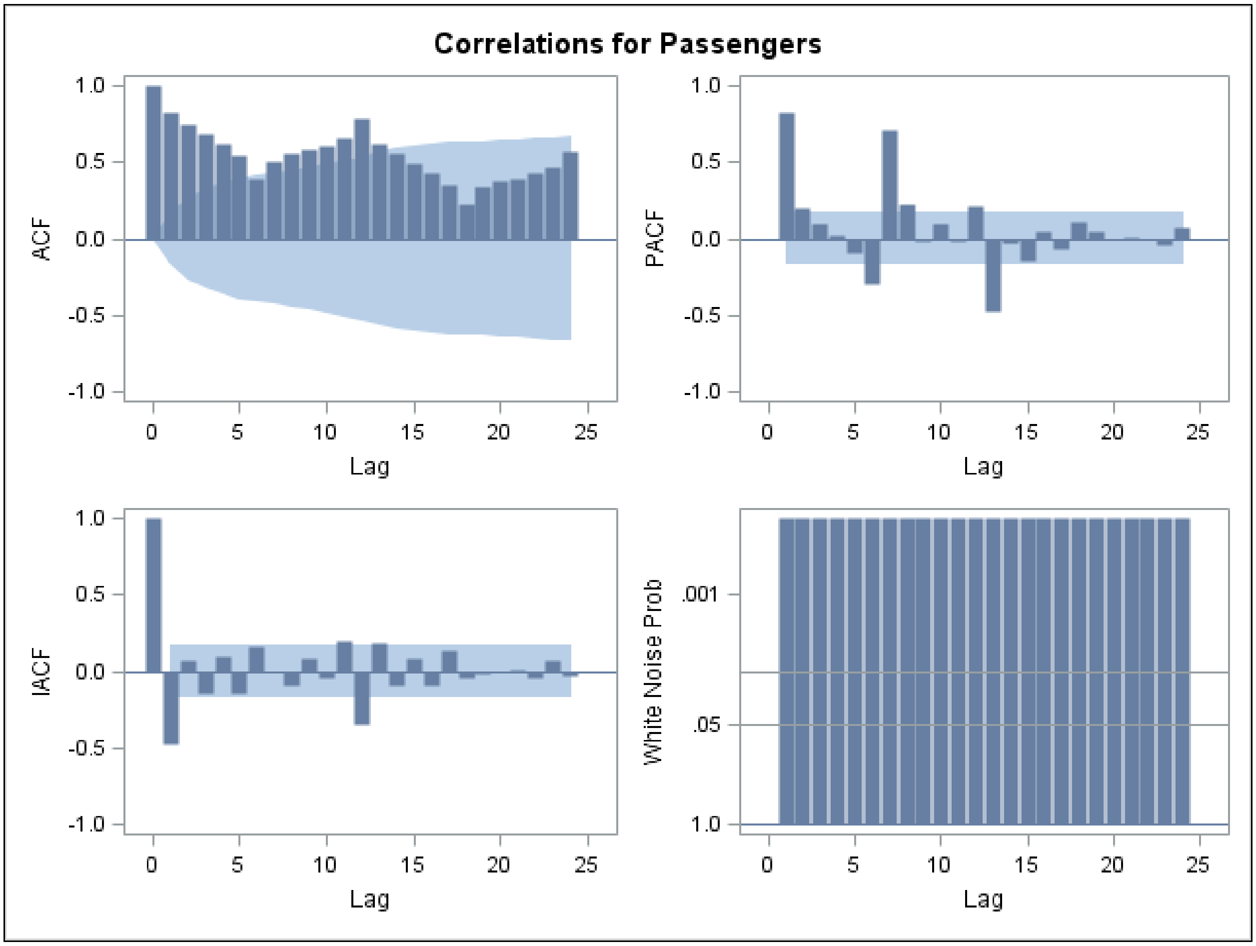
You can obtain diagnostic plots from PROC TIMESERIES.

```
proc timeseries data=work.Air1990_2000
                out=work.temp
                outdecomp=work.decomp
                plot=(series corr acf pacf iacf wn decomp tc sc )
                seasonality=12;
   id Date interval=month;
   var Passengers;
   decomp tcc sc / mode=mult;
run;
```

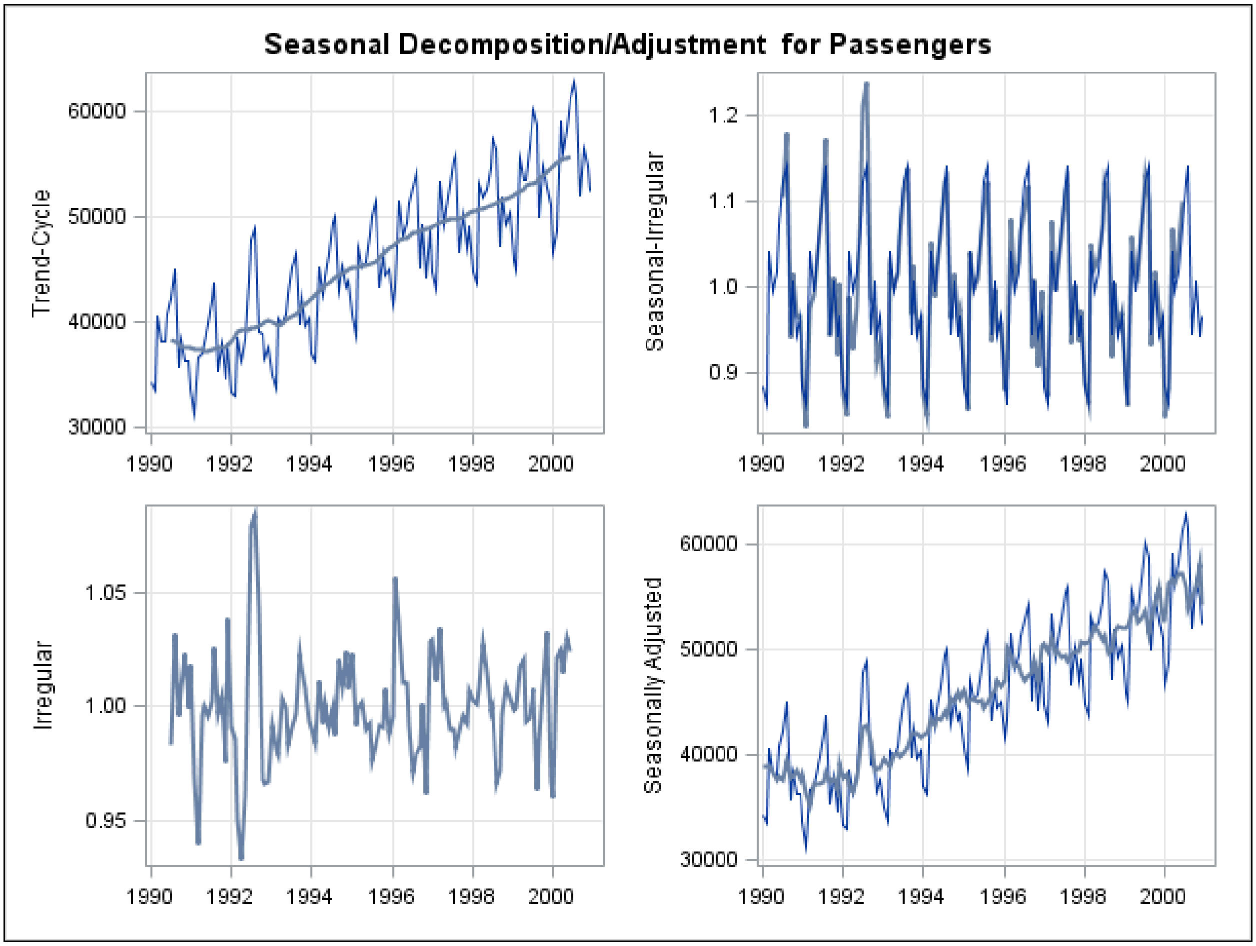
The plots follow:



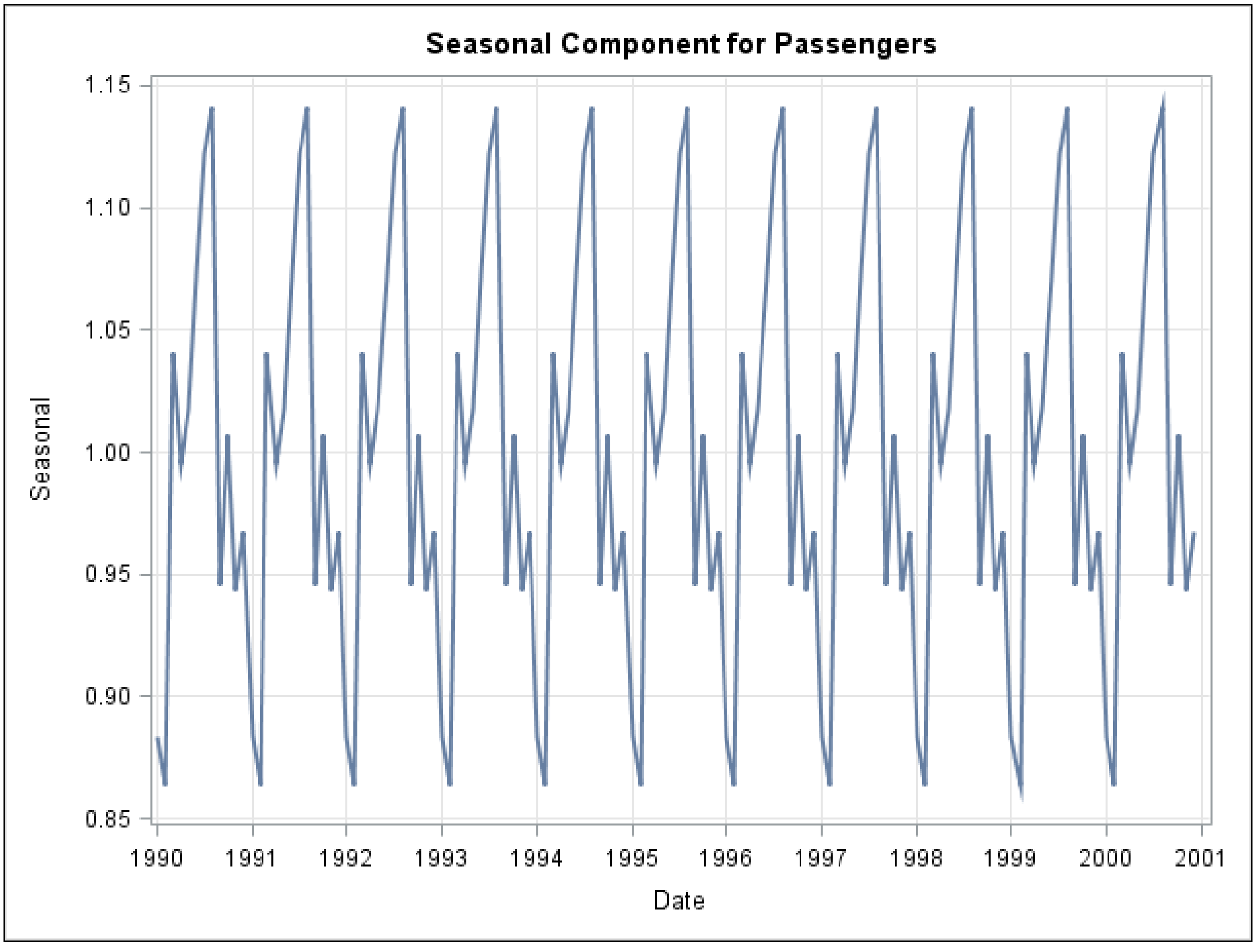
The airline passengers' time series clearly has strong trend and seasonal components.


Correlations for Passengers
ACF
PACF
IACF
White Noise Prob
Lag
1.0
0.5
0.0
-0.5
-1.0
.001
.05
0
5
10
15
20
25

The autocorrelation plots support the finding of trend and seasonality.

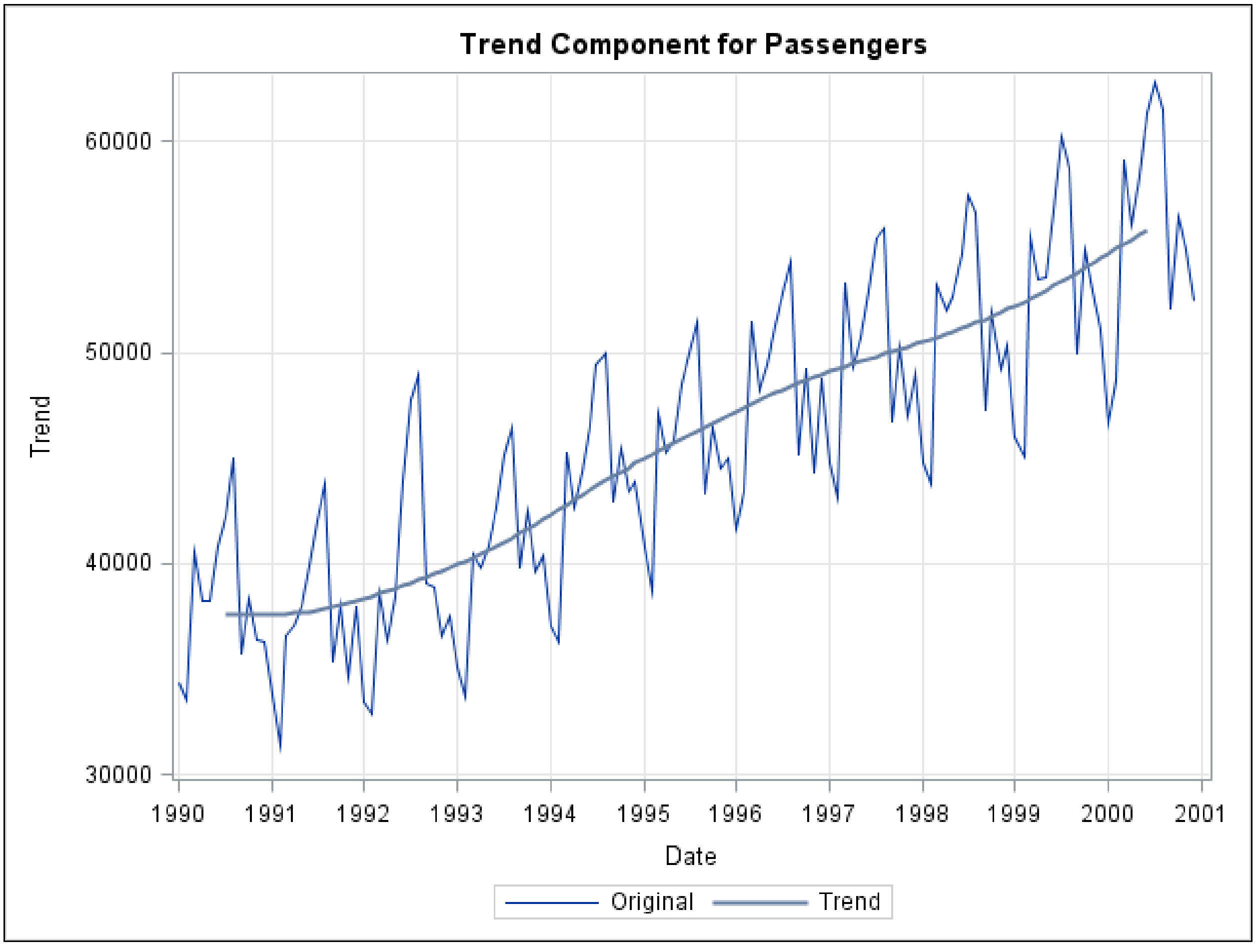


The decomposition plots help propose model components for handling trend and seasonality.



The seasonal components plot is difficult to interpret because it spans all years in the series. One strategy for overcoming this high-density plot problem is to restrict the seasonal values to one full year. Code is presented below the trend plot to accomplish this task.

The moving average trend plot follows:

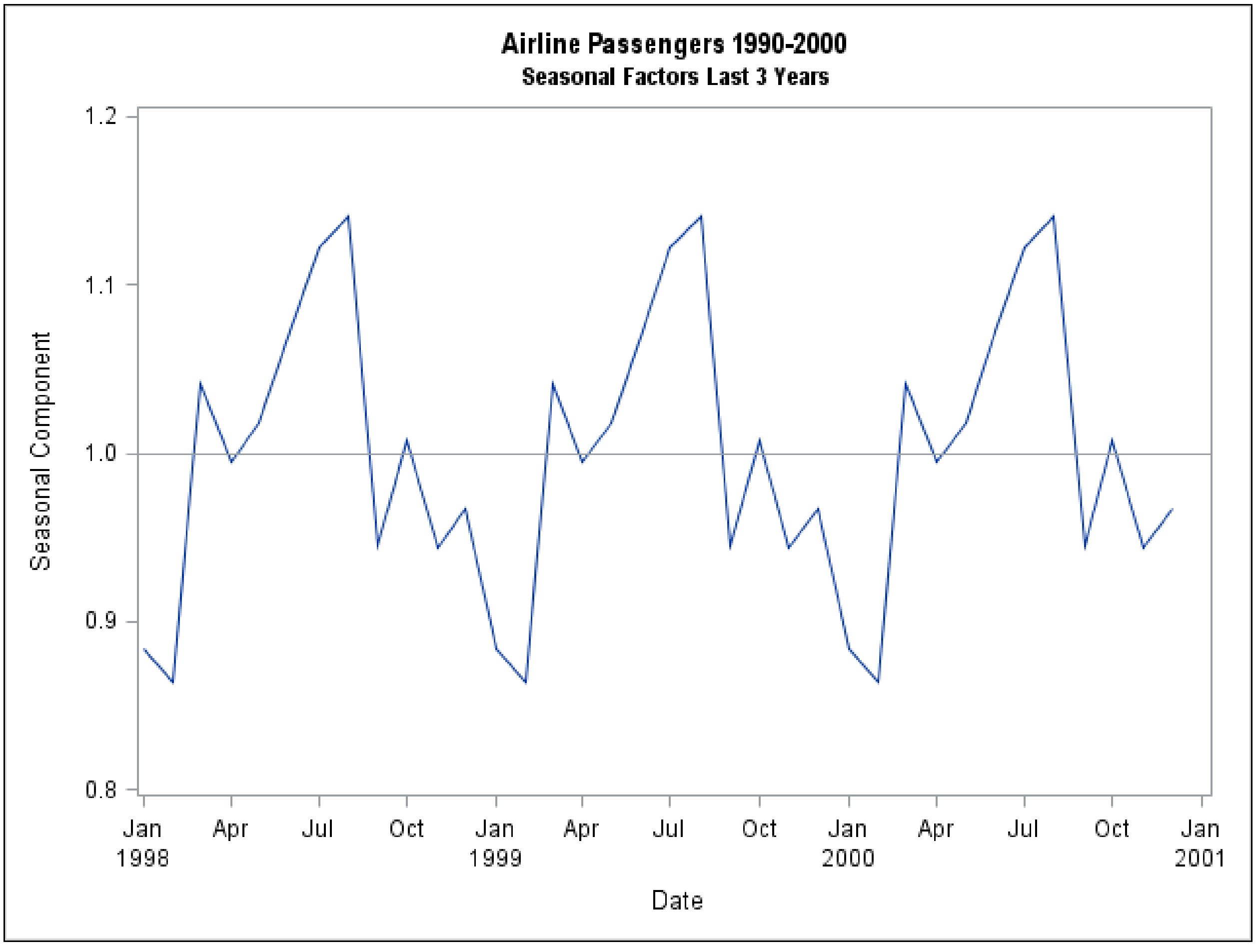


The trend fluctuates slightly from pure linear trend.

You can use the following code to improve the seasonal decomposition plot:

```
data work.decomp;
   set work.decomp(where=(year(Date)>=1998));
   keep Date _SEASON_ SC;
run;

title2 font=&COURSEFONT color=black "Seasonal Factors Last 3 Years";
/*----  Less data and slightly better aspect ratio  ----*/
proc sgplot data=work.decomp;
   series x=Date y=SC;
   refline 1 / axis=y;
   yaxis min=0.8 max=1.2;
run;
```

The plot of the multiplicative seasonal components for the past three years follows. Of course, the components are constant for each month and thus are identical across all years.



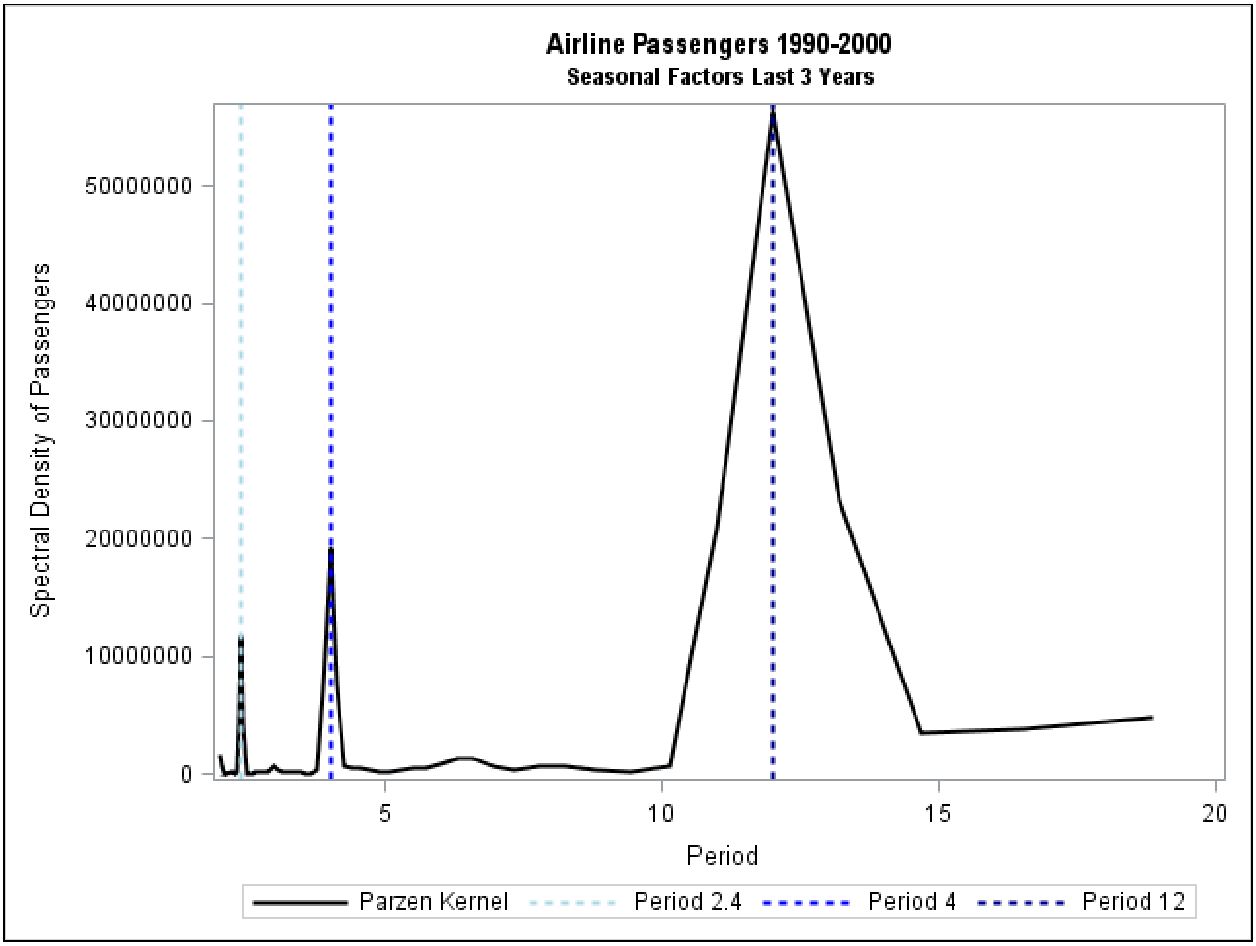
The above plot makes it very clear that peak travel occurs in August, and February is the lowest travel month.

The sample spectral density will help provide additional information about the nature of the seasonality.

```
proc spectra data=work.Air1990_2000
             out=work.Periodogram s;
   var Passengers;
   weights Parzen;
run;

proc sgplot data=work.Periodogram(where=(2<=PERIOD<=20));
   series x=PERIOD y=S_01 /
          lineattrs=GraphPrediction(pattern=1 color=black)
          legendlabel="Parzen Kernel" name="series1";
   refline 2.4 / axis=x
           lineattrs=GraphPrediction(pattern=2 color=lightblue)
           legendlabel="Period 2.4" name="series2";
   refline 4 / axis=x
           lineattrs=GraphPrediction(pattern=2 color=blue)
           legendlabel="Period 4" name="series3";
   refline 12 / axis=x
           lineattrs=GraphPrediction(pattern=2 color=darkblue)
           legendlabel="Period 12" name="series4";
   keylegend "series1" "series2" "series3" "series4" /
             location=outside position=bottom;
run;
```

The reference lines were added after the preliminary plot revealed dominant seasonal periods.

The plot follows:



The data reflects sinusoidal behavior with periods 12, 4, and 2.4.

The evidence for trend and seasonality is strong enough to forego the Dickey-Fuller tests. The following code adds candidate trend and seasonal terms to the data:

```
data work.Air1990_2000;
   set work.Air1990_2000 end=lastobs;
   array Seas{*} MON1-MON11;
   retain TwoPi . Time 0 MON1-MON11 .;
   if (TwoPi=.) then TwoPi=2*constant("pi");
   if (MON1=.) then do index=1 to 11;
      Seas[index]=0;
   end;
   Time+1;
   S2p4=sin(TwoPi*Time/2.4);
   C2p4=cos(TwoPi*Time/2.4);
   S4=sin(TwoPi*Time/4);
   C4=cos(TwoPi*Time/4);
   S12=sin(TwoPi*Time/12);
   C12=cos(TwoPi*Time/12);
   if (month(Date)<12) then do;
      Seas[month(Date)]=1;
      output;
      Seas[month(Date)]=0;
   end;
   else output;
   if (lastobs) then do;
      Passengers=.;
      do index=1 to 24;
         Time+1;
         Date=intnx("month",Date,1);
         S2p4=sin(TwoPi*Time/2.4);
         C2p4=cos(TwoPi*Time/2.4);
         S4=sin(TwoPi*Time/4);
         C4=cos(TwoPi*Time/4);
         S12=sin(TwoPi*Time/12);
         C12=cos(TwoPi*Time/12);
         if (month(Date)<12) then do;
            Seas[month(Date)]=1;
            output;
            Seas[month(Date)]=0;
         end;
         else output;
      end;
   end;
   drop index TwoPi;
run;
```

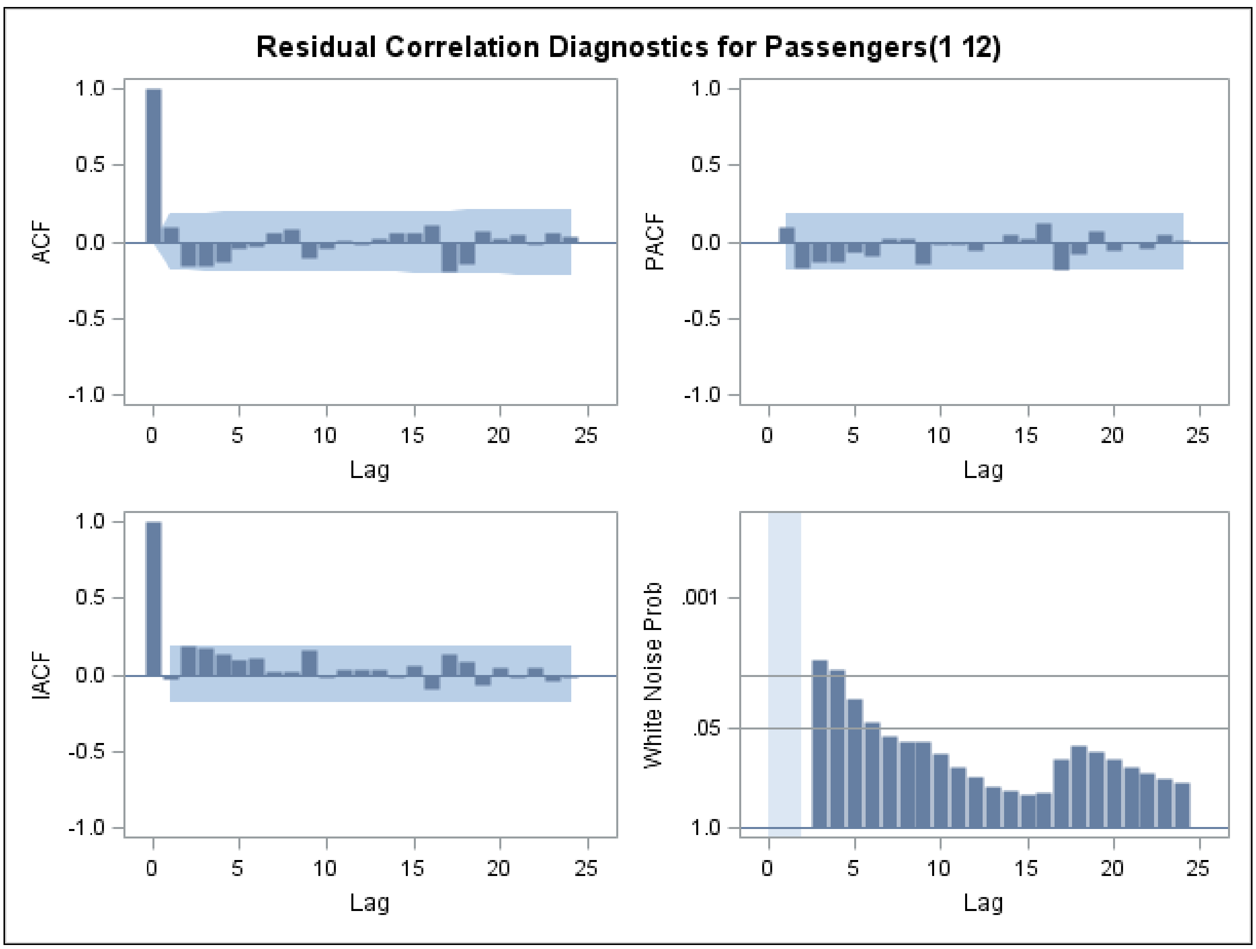
Inputs were extrapolated for forecasting the next 24 months. While you might want to investigate the use of sinusoids for this problem, they are excluded from further analysis to promote instructional goals.

The classic airline model proposed by Box and Jenkins for the international airline passengers data from 1948 to 1960 uses a log transformation and fits an ARIMA(0,1,1)(0,1,1)$_{12}$ model to the transformed series. The log transformation was primarily used as a variance stabilizing transformation. The modern airline data does not exhibit increasing variance as a function of time, so the log transformation will not be considered. The following code fits the classic airline model without the log transformation to the current domestic U.S. airline passengers data:

```
proc arima data=work.Air1990_2000 plots=all;
   identify var=Passengers(1 12) noprint;
   estimate q=(1)(12)
            method=ml
            outstat=statclassic;
   forecast id=Date interval=month lead=24
            out=work.fclassic;
quit;
```

The fit is acceptable, although the Ljung-Box chi-square test for white noise fails for lags 3, 4, 5, and 6. The residual autocorrelations suggest a more complex error component.

The residual autocorrelation plots follow:

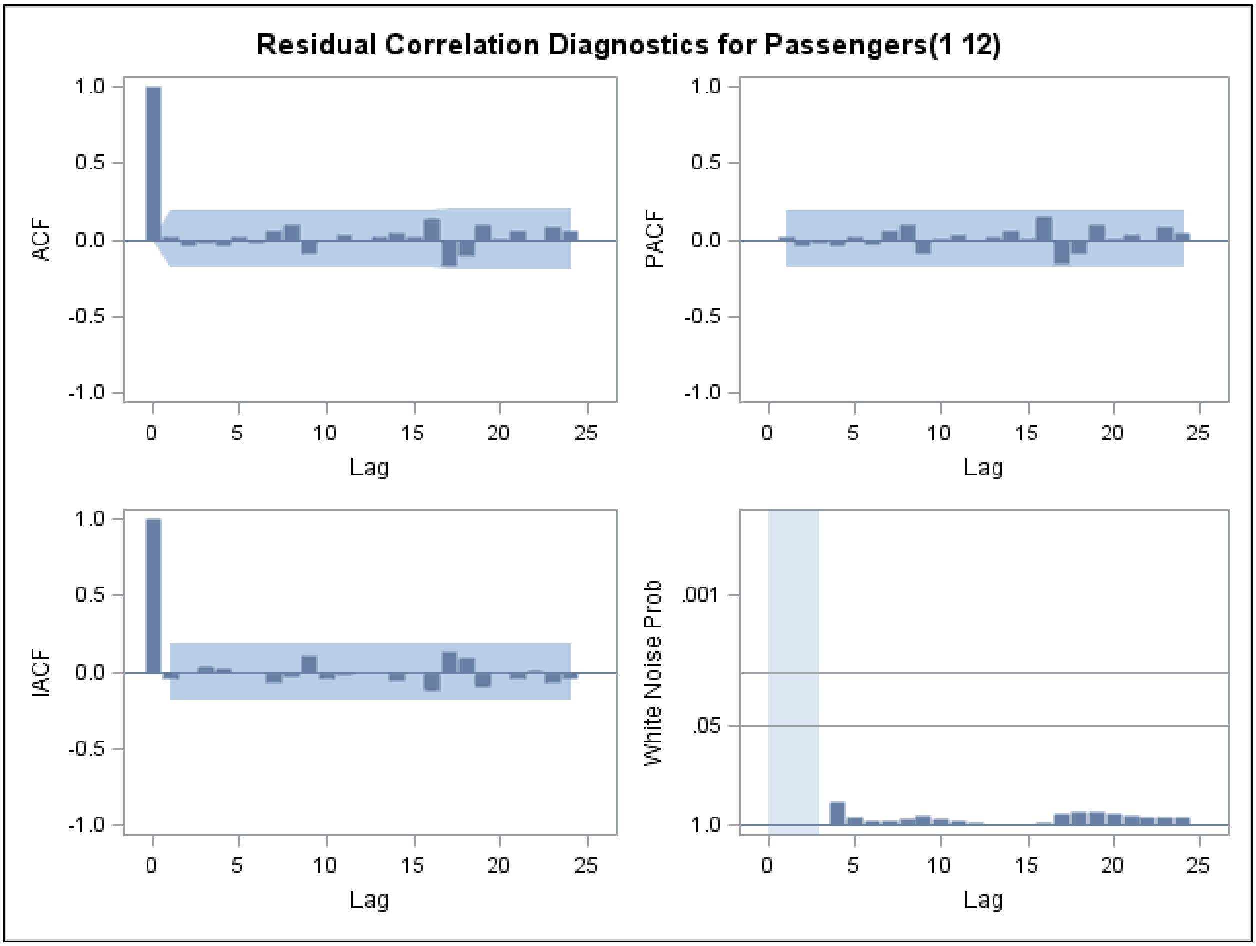


Box and Jenkins recommend that models with difference orders 1 and 12 use corresponding MA subset factors. This is consistent with the classic airline model.

The following code tries the recommended MA orders with the addition of P=1. The choice of P=1 was made after trying different values.

```
proc arima data=work.Air1990_2000 plots=all;
   identify var=Passengers(1 12) noprint;
   estimate p=1 q=(1)(12)
            method=ml
            outstat=statclassicp1;
   forecast id=Date interval=month lead=24
            out=work.fclassicp1;
quit;
```

The residual plots follow:



Adding a single parameter seems to have resolved white noise issues.

The above model is an ARIMA(1,1,1)(0,1,1)$_{12}$ model.

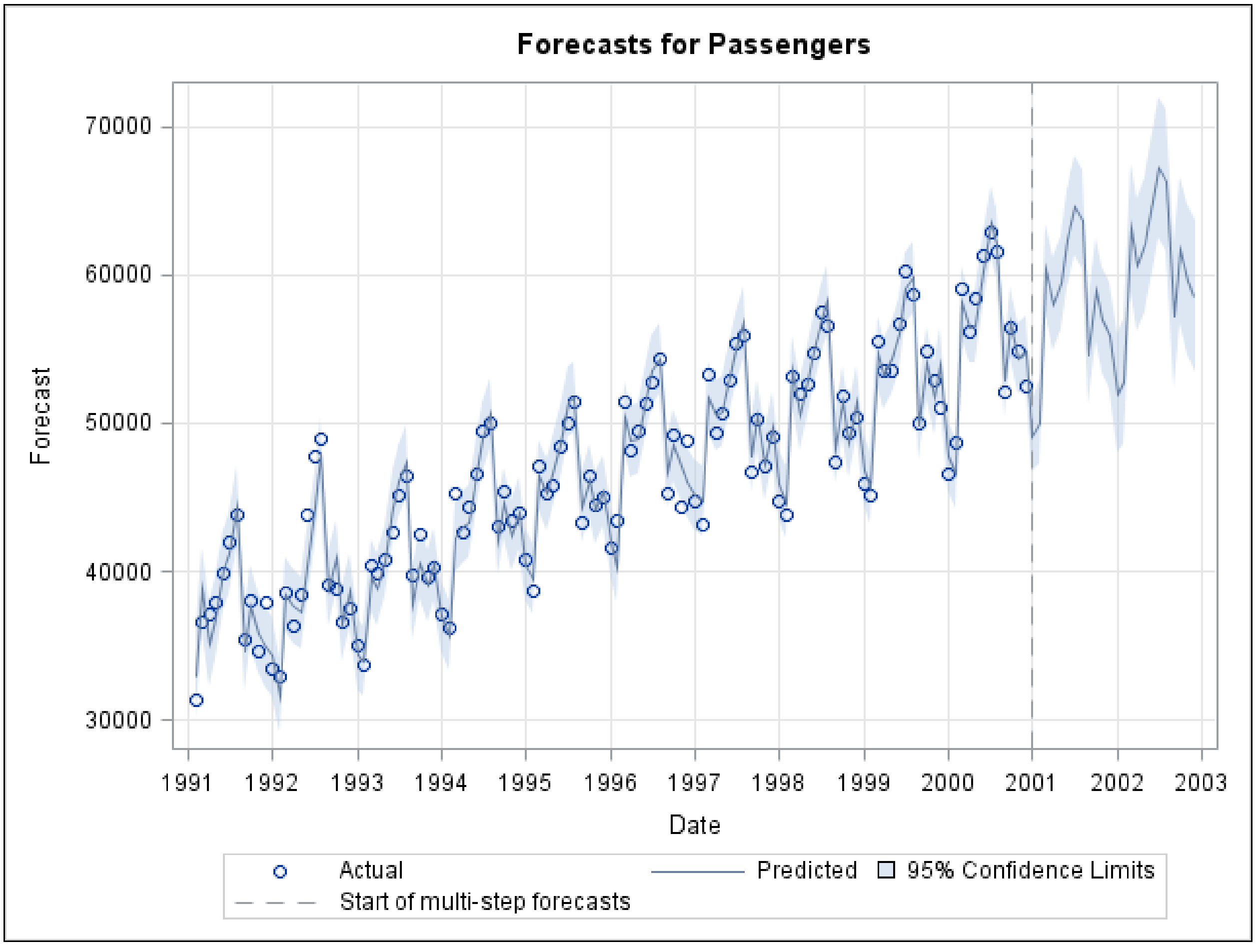
The estimates and related statistics for this model appear below:

Maximum Likelihood Estimation					
Parameter	Estimate	Standard Error	t Value	Approx Pr > |t|	Lag
MU	16.65439	15.79306	1.05	0.2916	0
MA1,1	0.87853	0.07665	11.46	<.0001	1
MA2,1	0.52034	0.09722	5.35	<.0001	12
AR1,1	0.51025	0.12386	4.12	<.0001	1

Constant Estimate	8.156536
Variance Estimate	1476186
Std Error Estimate	1214.984
AIC	2036.642
SBC	2047.758
Number of Residuals	119

The autocorrelation plots suggest white noise, but the Ljung-Box test for lags 3 through 6 rejects white noise.

The forecasts from this model appear below:



One additional ARIMA model is considered, one with linear trend and seasonal dummies.

```
proc arima data=work.Air1990_2000 plots=all;
   identify var=Passengers
            cross=(Time
                   MON1 MON2 MON3 MON4 MON5 MON6
                   MON7 MON8 MON9 MON10 MON11) noprint;
   estimate input=(Time
                   MON1 MON2 MON3 MON4 MON5 MON6
                   MON7 MON8 MON9 MON10 MON11)
            method=ml;
run;
   estimate input=(Time
                   MON1 MON2 MON3 MON4 MON5 MON6
                   MON7 MON8 MON9 MON10 MON11)
            p=1
            method=ml
            outstat=work.stattsd;
   forecast id=Date interval=month lead=24
            out=work.foretsd;
quit;
```

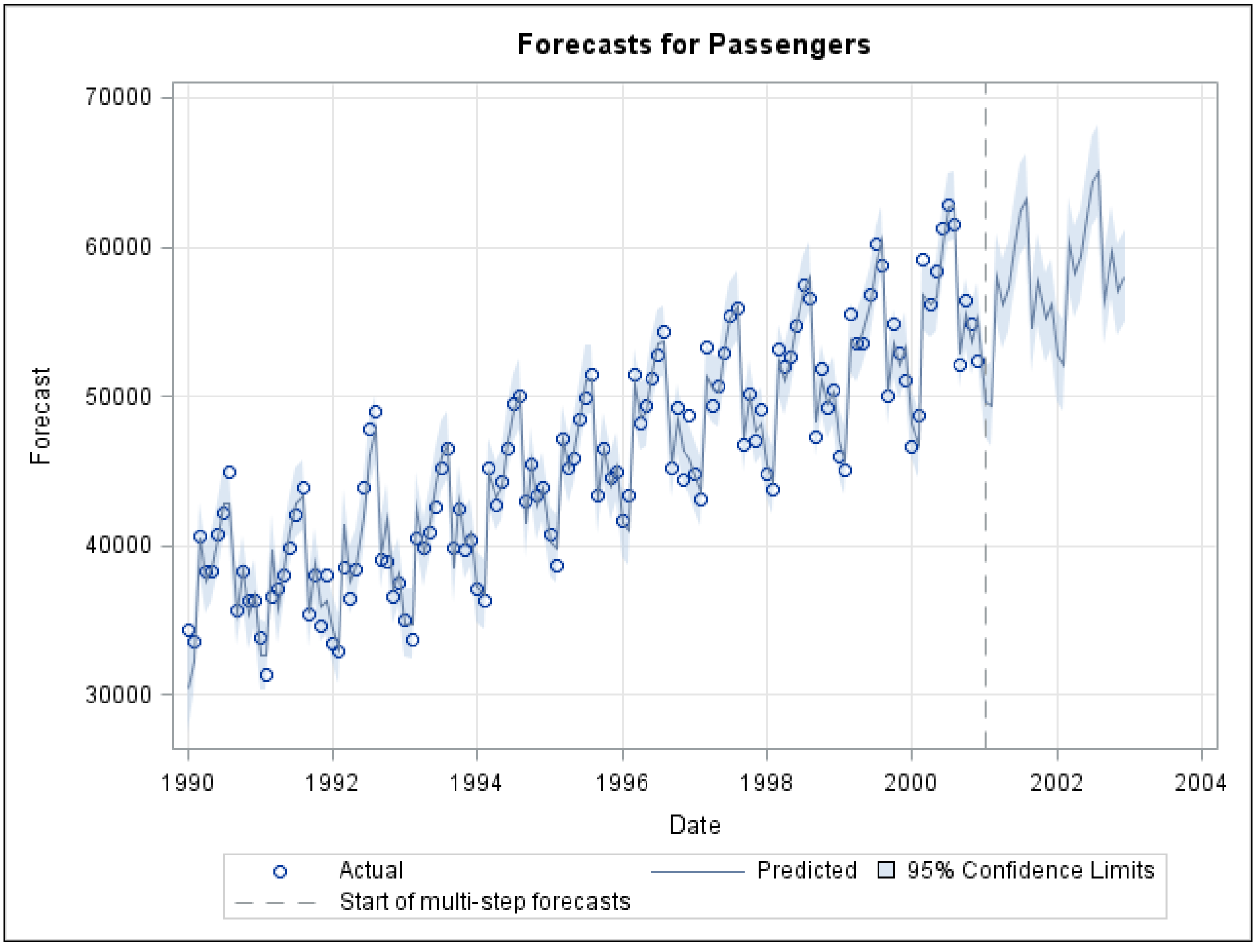
Estimates for the above model follow:

Maximum Likelihood Estimation							
Parameter	Estimate	Standard Error	t Value	Approx Pr > |t|	Lag	Variable	Shift
MU	33888.0	684.48738	49.51	<.0001	0	Passengers	0
AR1,1	0.64930	0.07040	9.22	<.0001	1	Passengers	0
NUM1	154.67995	7.40392	20.89	<.0001	0	Time	0
NUM2	-3591.9	404.02096	-8.89	<.0001	0	MON1	0
NUM3	-4401.7	507.16425	-8.68	<.0001	0	MON2	0
NUM4	3703.6	561.46875	6.60	<.0001	0	MON3	0
NUM5	1519.8	591.27309	2.57	0.0102	0	MON4	0
NUM6	2436.2	606.15438	4.02	<.0001	0	MON5	0
NUM7	4915.5	610.23633	8.06	<.0001	0	MON6	0
NUM8	7151.1	604.63477	11.83	<.0001	0	MON7	0
NUM9	7685.2	588.01314	13.07	<.0001	0	MON8	0
NUM10	-1078.3	555.97386	-1.94	0.0524	0	MON9	0
NUM11	1941.9	498.48056	3.90	<.0001	0	MON10	0
NUM12	-765.03902	389.76551	-1.96	0.0497	0	MON11	0

Constant Estimate	11884.5
Variance Estimate	1396483
Std Error Estimate	1181.729
AIC	2256.077
SBC	2296.437
Number of Residuals	132

Further inspection of this model is left as an exercise.

The forecasts from this model appear below:



To compete with PROC ARIMA, consider fitting a seasonal exponential smoothing model by using PROC ESM. The macro %AutoESM fits all seven supported models and compiles statistics on each model.

```
%AutoESM(work.Air1990_2000,work.esm_stats,Passengers,Date);

proc sort data=work.esm_stats;
   by MAPE;
run;

proc print data=work.esm_stats noobs;
   var Model MAPE RMSE AIC SBC SMAPE;
run;
```

The additive Winters model has the smallest goodness-of-fit statistic no matter which one you select, so it is chosen to generate forecasts.

Model	MAPE	RMSE	AIC	SBC	SMAPE
AddWinters	1.98741	1191.92	1876.00	1884.65	1.98320
Winters	2.06413	1229.69	1884.23	1892.88	2.06758
Seasonal	2.10521	1249.12	1886.37	1892.14	2.11226
Linear	6.68991	3931.81	2189.09	2194.86	6.63333
DampTrend	6.69401	3931.67	2191.08	2199.73	6.64336
Simple	6.70936	3936.22	2187.39	2190.27	6.69014
Double	7.85912	4138.01	2200.58	2203.47	7.82523

The following code fits the additive Winters model and produces forecasts:

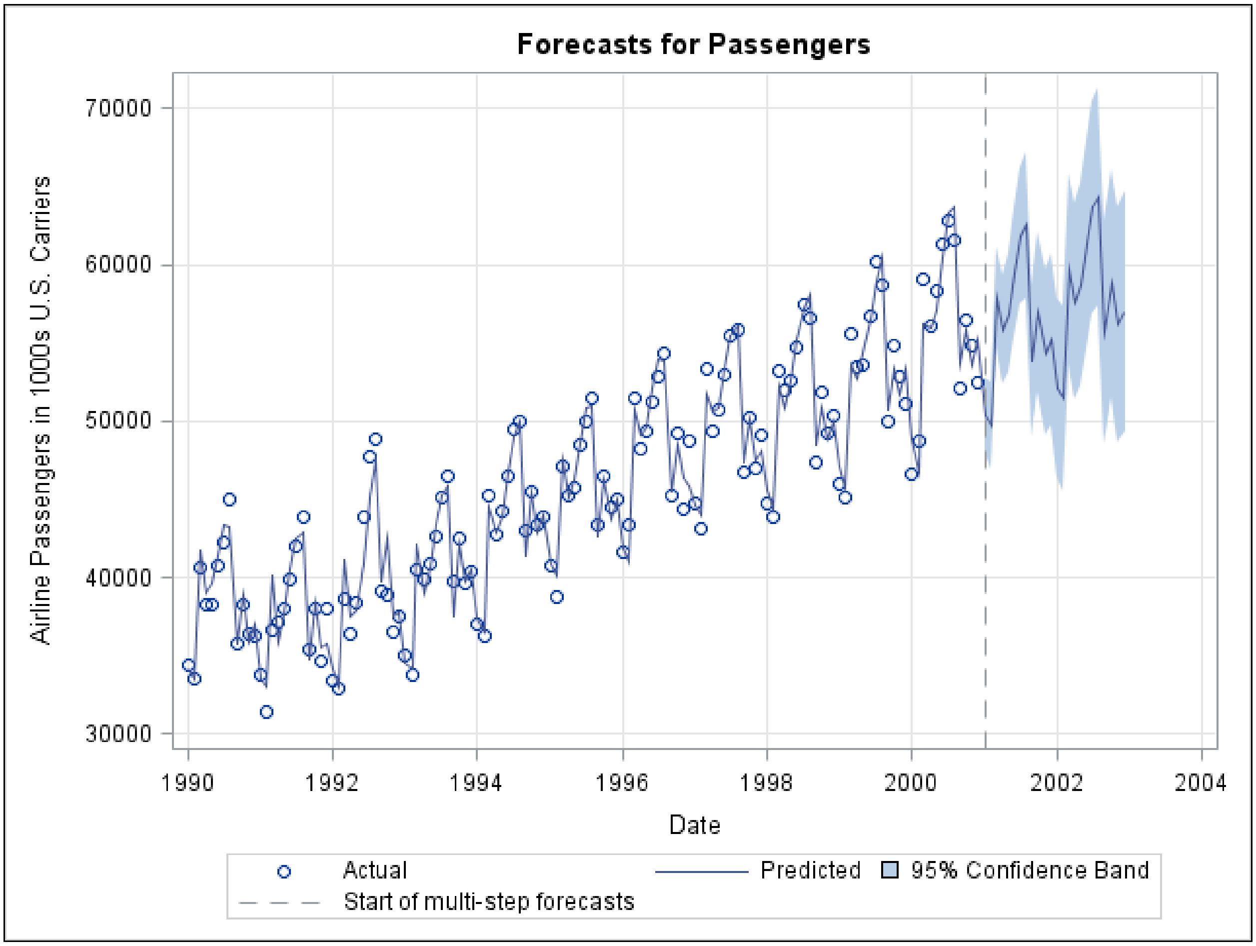
```
ods output SmoothedStates=work.SmoothedStates
           SeasonStatePlot=work.SeasonStatePlot;
proc esm data=work.Air1990_2000(where=(Date<='31DEC2000'd))
     out=work.out
     outfor=work.foreesm
     outstat=work.StatESM
     lead=24
     print=(ESTIMATES STATISTICS SUMMARY STATES)
     plonk=(WINTER ISCOLD)
     plot=(MODELS FORECASTS LEVELS TRENDS SEASONS);
   id Date interval=month;
   forecast Passengers / model=addwinters;
run;
ods output close;
```

Two ODS tables are captured to provide scrutiny of the model. Examination of these two tables is left as an exercise. Notice that the WHERE clause removes the data added to the end. Otherwise, PROC ESM would forecast 24 months beyond the end of the extended data, which would be 48 months into the future. The table of estimated smoothing weights follows:

Winters Method (Additive) Parameter Estimates				
Parameter	Estimate	Standard Error	t Value	Approx Pr > |t|
Level Weight	0.63498	0.05880	10.80	<.0001
Trend Weight	0.0010000	0.0050719	0.20	0.8440
Seasonal Weight	0.0010000	0.04573	0.02	0.9826

The trend and seasonal weights hit the lower bound imposed by PROC ESM. When this occurs, the *t* statistics and *p*-value are based on poor standard error estimates and cannot be trusted. You should ***not*** interpret the estimates to imply that trend and seasonal components are not needed.

The forecasts for this model appear below:



The remainder of the code in the demonstration uses several goodness-of-fit macros to compile statistics for comparing the models. The final table of comparison statistics appears below:

Model	SBC_SSE	AIC_SSE	MAPE	RMSE
ARIMA(1,1,1)(0,1,1)12	1711.96	1700.84	2.10803	1248.72
ARIMA(0,1,1)(0,1,1)12	1717.11	1708.77	2.19770	1296.29
Winters Additive	1884.65	1876.00	1.98741	1205.70
Linear+SeasDummies	1926.32	1885.96	1.93282	1204.47

As stated elsewhere, the SSE versions of AIC and SBC must be used. While the last two models in the table are competitive based on MAPE, the MAPE values are too close to provide compelling evidence for choosing any model. On the other hand, the information criteria favor the ARIMA models. Again, you should consider more than just a single goodness-of-fit statistic when selecting a model. For the purposes of this demonstration, the ARIMA(0,1,1)(0,1,1)$_{12}$ model would appear to be a good choice. This model will be used later as the base model when the effects of the events of September 11, 2001, are considered for this data.

4.02 Multiple Choice Poll

Which procedure has no specific options for dealing with long term seasonal effects?

a. PROC ARIMA
b. PROC AUTOREG
c. PROC ESM
d. PROC FORECAST

63

Exercises

1. Reproduce the demonstration for program **Demo4_02Airline.sas**. Do all models suggested in the demonstration seem adequate?

2. For the additive Winters model fit in program **Demo4_02Airline.sas**, run the part of the code that plots information in the two ODS tables that are created. What is the purpose of examining these two tables?

4.4 Chapter Summary

Box-Jenkins time series analysis promotes the use of differencing to model trend and seasonality. The general Box-Jenkins ARIMA(p,d,q)(P,D,Q)$_s$ model can be used to forecast nonstationary time series using stochastic trend and seasonal components.

The Box-Jenkins methodology developed in previous chapters is expanded to include identification of seasonal orders (P,D,Q). The estimation and forecasting steps include a step to convert a series to stationarity, estimate the stationary component, and then integrate back to the original series.

Despite the emphasis on differencing, the methodology accommodates deterministic trend and seasonal components as well as stochastic ones. PROC ARIMA provides a full implementation of Box-Jenkins models. In addition, PROC SPECTRA provides diagnostic functions to help diagnose seasonality.

In addition to models for trend, PROC ESM provides one model for time series with seasonal effects only, and two models for time series with trend and seasonal factors.

PROC FORECAST can be used to forecast seasonal time series, but the seasonal component is stationary. The seasonal effects might hold up for short forecast horizons but should not be used for long horizons.

For Additional Information

Bartlett, M. S. 1966. *An Introduction to Stochastic Processes*. Second Edition, Cambridge: Cambridge University Press.

Box, G.E.P., and G.M. Jenkins. 1976. *Time Series Analysis: Forecasting and Control*. Oakland, California: Holden-Day.

Parzen, E. 1957. "On Consistent Estimates of the Spectrum of a Stationary Time Series," *Annals of Mathematical Statistics*, 28, 329-348.

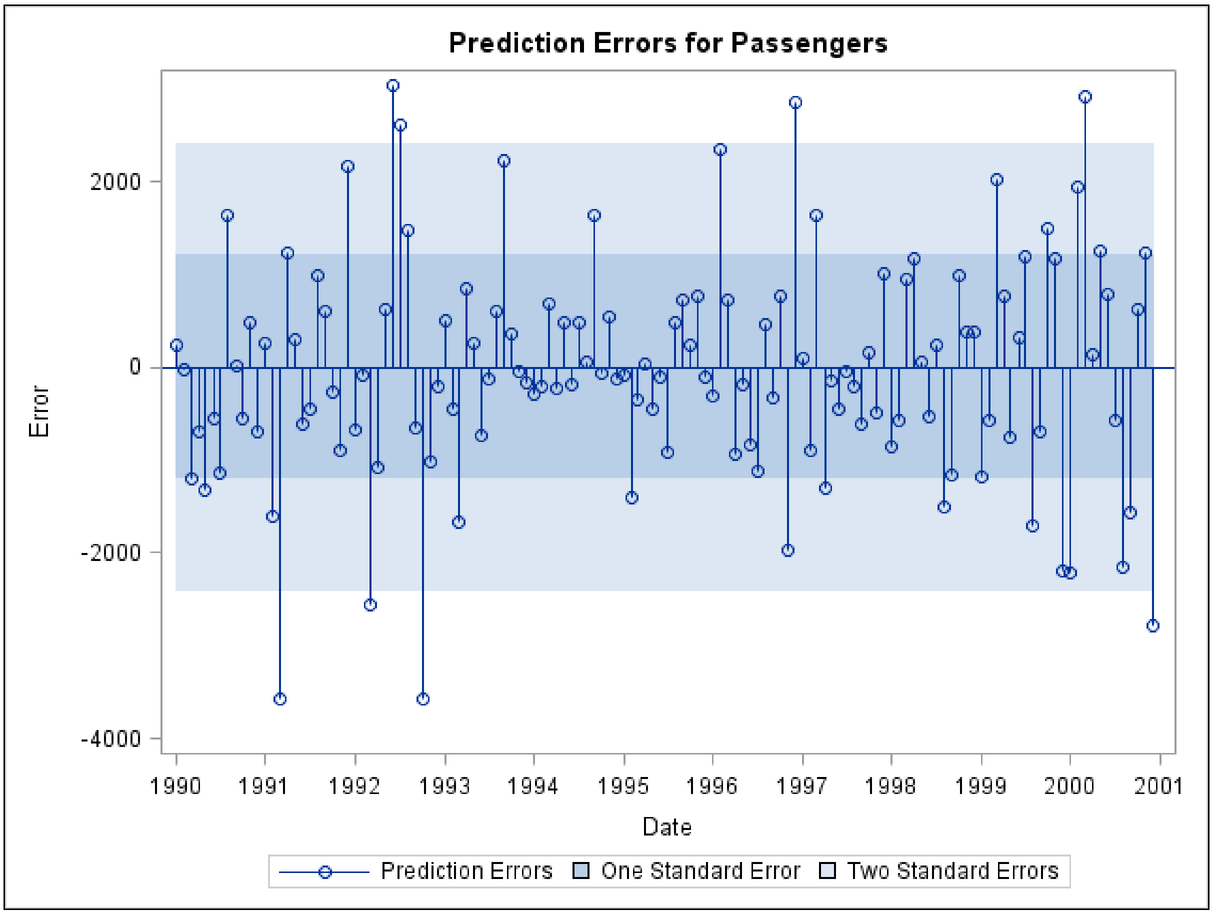
Priestly, M. B. 1981. *Spectral Analysis and Time Series*. New York: Academic Press, Inc.

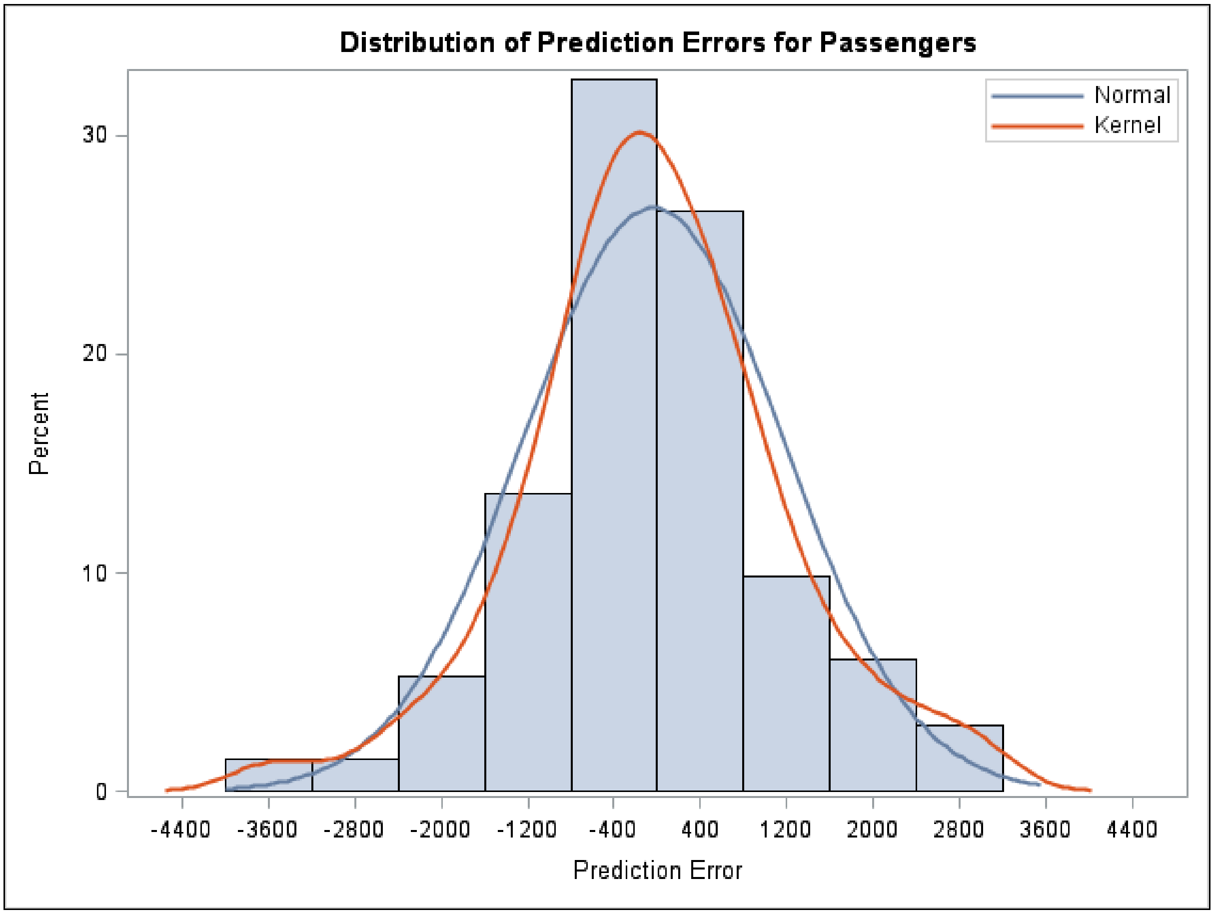
4.5 Solutions

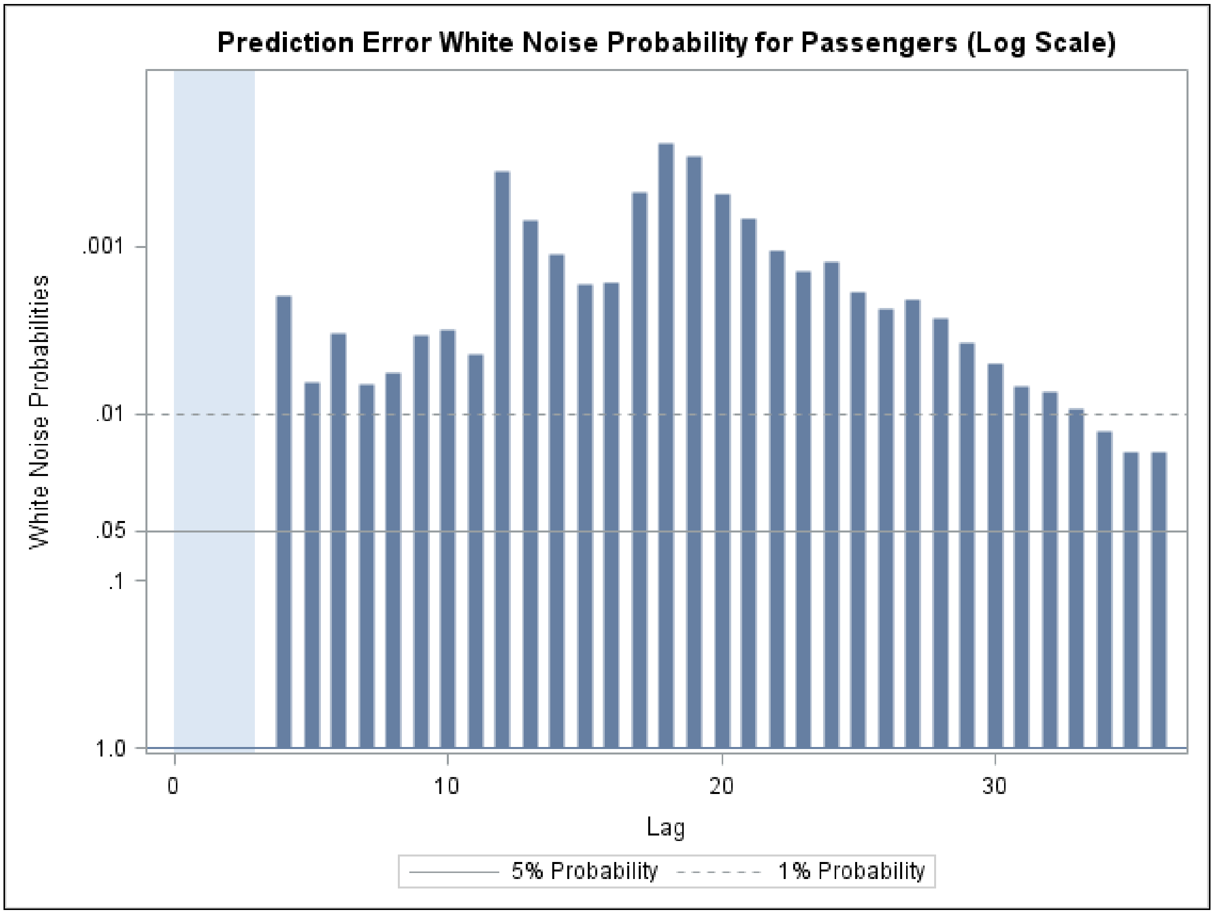
Solutions to Exercises

1. Reproduce the demonstration for program **Demo4_02Airline.sas**. Do all models suggested in the demonstration seem adequate?

The code includes diagnostic plots. The course notes imply that the classic airline model might not be adequate due to Ljung-Box test results. The additive Winters model produces a large set of goodness-of-fit statistics, but PROC ESM options were not selected for analysis of residuals. Replace the PLOT= option as written with PLOT=ALL. This will generate plots like the following:




Distribution of Prediction Errors for Passengers
Normal
Kernel
Percent
0
10
20
30
-4400
-3600
-2800
-2000
-1200
-400
400
1200
2000
2800
3600
4400
Prediction Error



Evidence suggests that the additive Winters model might yield residuals with some autocorrelation present.

2. For the additive Winters model fit in program **Demo4_02Airline.sas**, run the part of the code that plots information in the two ODS tables that are created. What is the purpose of examining these two tables?

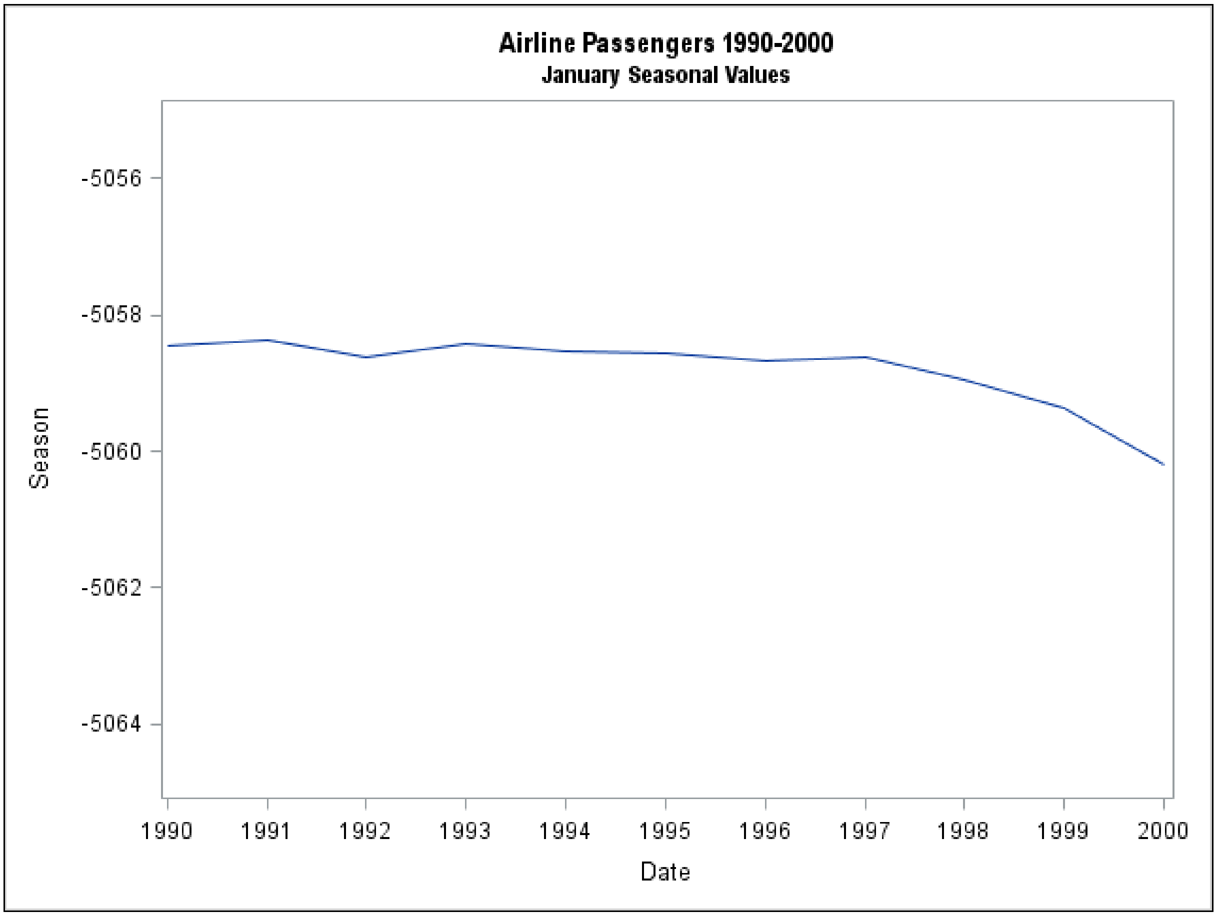
The relevant code that was excluded from the discussion above is included below. The PROC ESM portion of the code has been modified to restrict output to that which is related to this exercise.

```
ods output SmoothedStates=work.SmoothedStates
           SeasonStatePlot=work.SeasonStatePlot;
proc esm data=work.Air1990_2000(where=(Date<='31DEC2000'd))
     out=work.out
     outfor=work.foreesm
     outstat=work.StatESM
     lead=24
     print=(STATES)
     plot=(SEASONS);;
   id Date interval=month;
   forecast Passengers / model=addwinters;
run;
ods output close;
```

(Continued on the next page.)

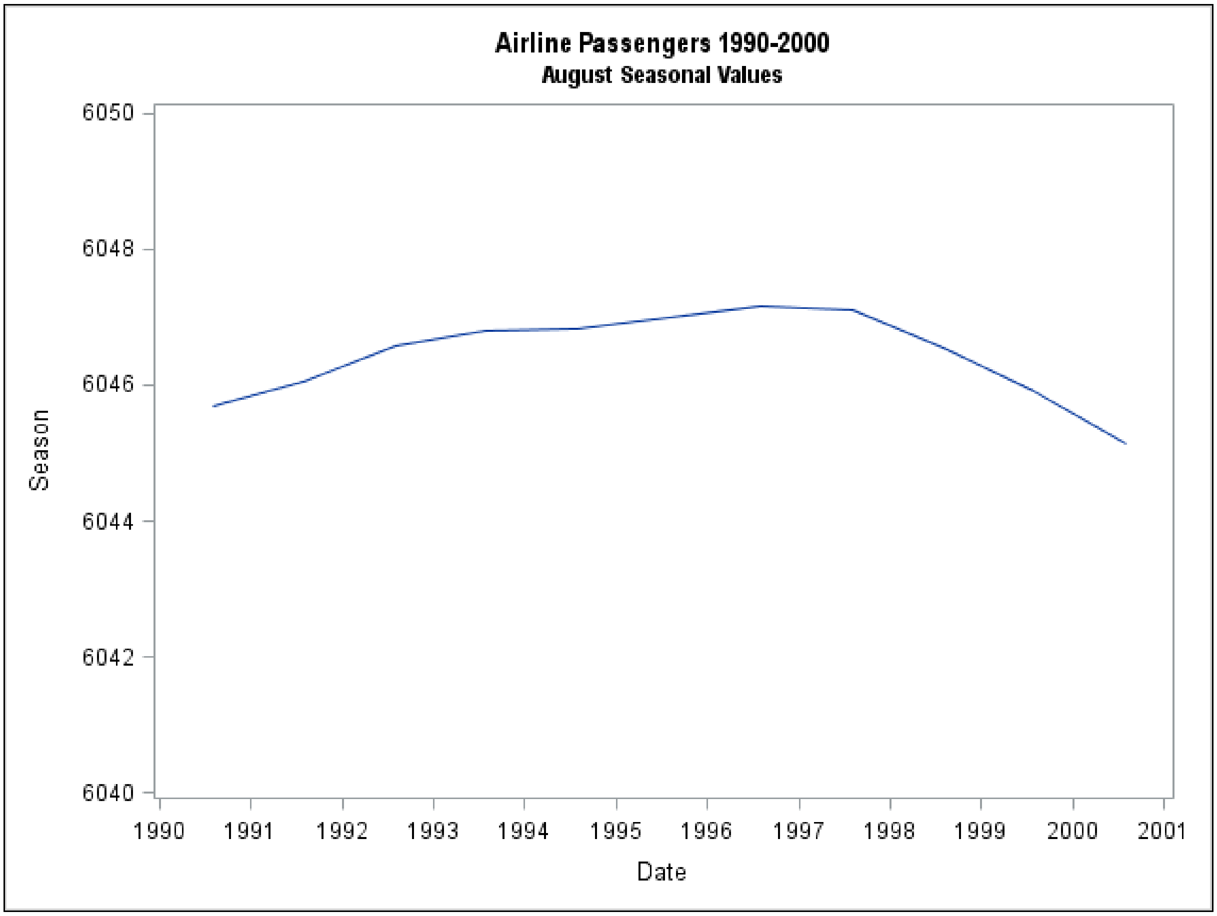
```
title2 font=&COURSEFONT color=black "January Seasonal Values";

proc sgplot data=SeasonStatePlot(where=(month(Time)=1));
   series x=Time y=Season;
   yaxis min=-5065 max=-5055;
run;

title2 font=&COURSEFONT color=black "August Seasonal Values";

proc sgplot data=SeasonStatePlot(where=(month(Time)=8));
   series x=Time y=Season;
   yaxis min=6040 max=6050;
run;

title2 font=&COURSEFONT color=black "Year 2000 Seasonal Values";

proc sgplot data=SeasonStatePlot(where=(year(Time)=2000));
   series x=Time y=Season;
   yaxis min=-8000 max=8000;
run;

proc sgplot data=SeasonStatePlot(where=(year(Time)=2000));
   vbar Time / response=Season;
run;
```

This January states plot appears below:



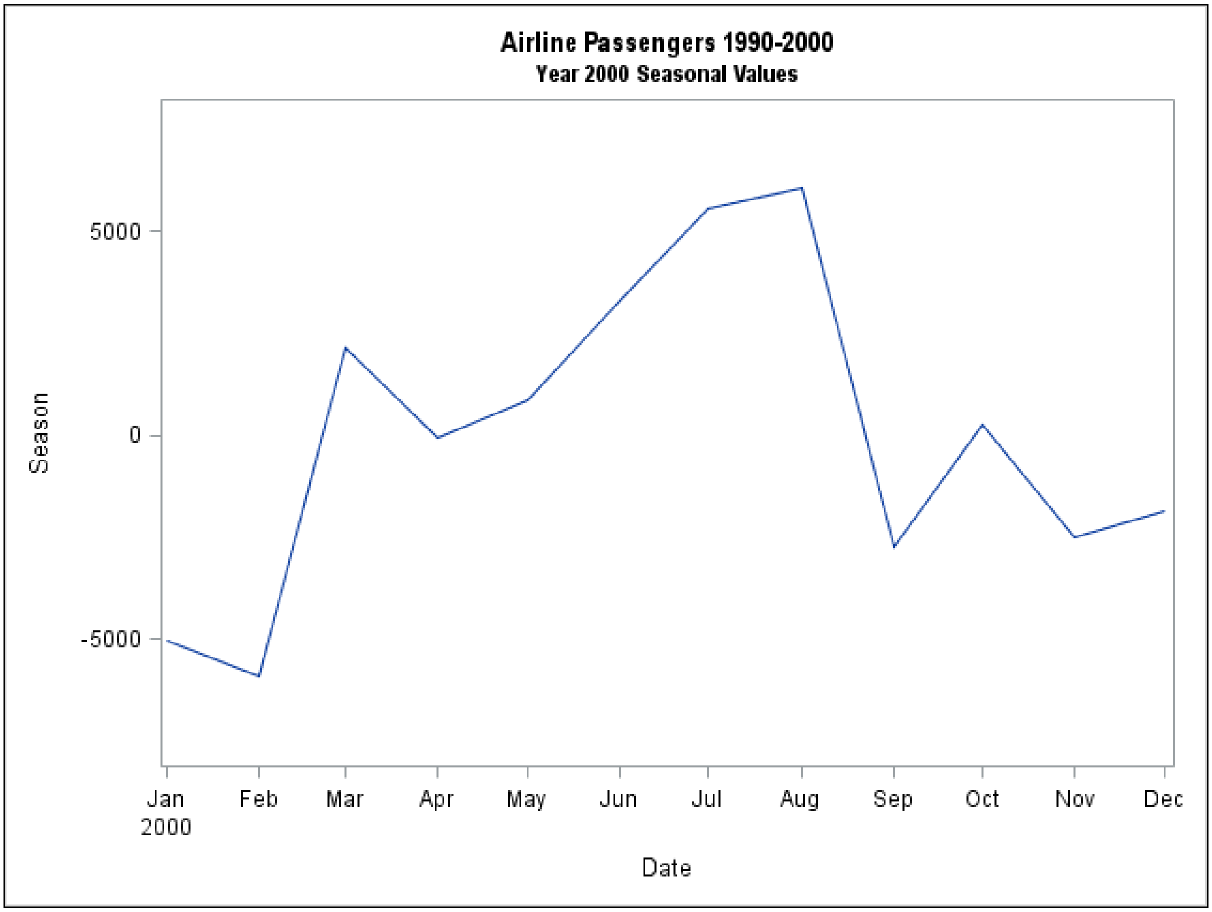
The January effect is negative, meaning that fewer passengers travel in January than the average travel month. The effect is relatively stable, but min and max values must be specified. Otherwise, the default settings of PROC SGPLOT would make the values appear to be more erratic.

The August plot follows:



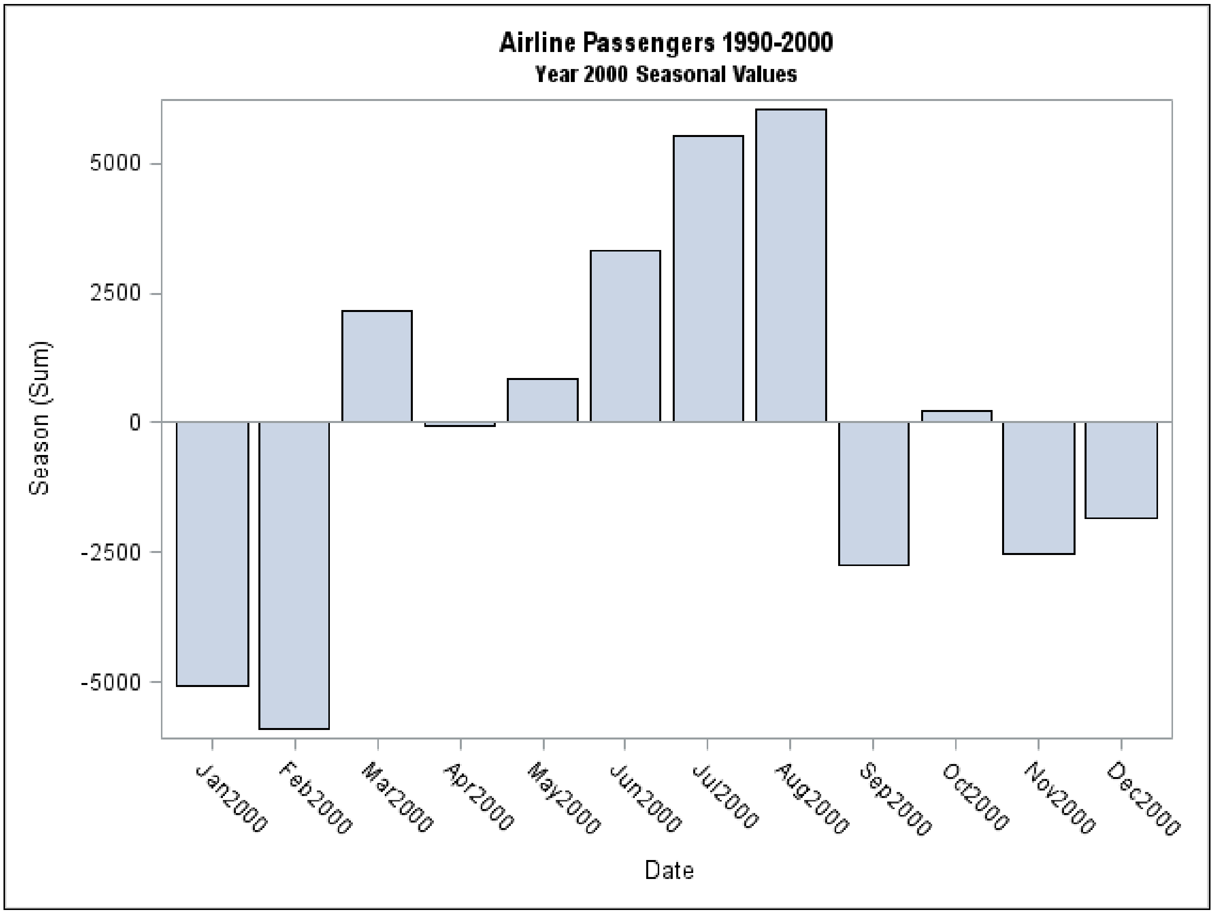
August has an overall positive effect, meaning more passengers than on average travel in August. In fact, August is the peak travel month. However, notice that the August effect appears to be more volatile than the January effect. Perhaps as a summer vacation month, higher variability results from passengers taking vacations earlier in the summer, or perhaps postponing trips until the U.S. Labor Day holiday in September. However, before you spend too much time trying to explain the volatility, note that the most extreme shift is only around 2,000 passengers.

Year 2000 represent the most recent year in the data. Hence, the 2000 plot contains the latest estimate of monthly effects.



The interpretation is clear—more people travel by air in the summer than in the winter.

The last plot simply presents the annual cycle in a different way.



The SmoothedStates data set contains the starting and final seasonal states. If you want the most up-to-date estimate of the August state, examine the final state estimate for August in the SmoothedStates table.

Examining the tables should have revealed the purpose in examining the tables. You are looking for any unusual variation that might influence how you model the data to generate forecasts.

Solutions to Student Activities (Polls/Quizzes)

4.01 Multiple Choice Poll – Correct Answer

A time series recorded every month has a seasonal component. Which of the following diagnostics indicate seasonality?

a. A significant PACF value at lag 12.
b. A sample spectral density function that exhibits a peak at period 12.
c. An augmented Dickey-Fuller seasonal test exhibits p-values larger than 0.1.
d. All of the above.

39

4.02 Multiple Choice Poll – Correct Answer

Which procedure has no specific options for dealing with long term seasonal effects?

a. PROC ARIMA
b. PROC AUTOREG
c. PROC ESM
d. PROC FORECAST

64

Chapter 5 Models with Explanatory Variables

5.1 Ordinary Regression Models

Objectives

- Explain multiple linear regression models.
- Examine linear regression assumptions.
- Explain the relationship between ordinary multiple linear regression models and time series regression models.

3

Multiple Linear Regression with Two Variables

$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \varepsilon$

Y is the target (response/dependent) variable.

X_1 and X_2 are input (predictor/independent) variables.

ε is the error term.

β_0, β_1, and β_2 are parameters.

β_0 is the intercept or constant term.

β_1 and β_2 are partial regression coefficients.

4

Multiple Linear Regression

Assumptions

- The predictor variables are known and measured without error.
- The functional relationship between inputs and target is linear.
- The error term represents a set of random variables that are independent and identically distributed with a Gaussian normal distribution having a mean of 0 and variance σ^2.

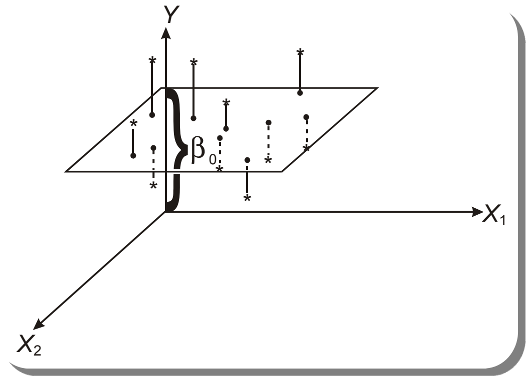
5

Violation of Model Assumptions

- *Normality* affects standard errors, and might or might not affect the parameter estimates.
- *Constant variance* does not affect the parameter estimates, but the standard errors are compromised.
- *Independent observations* do not affect the parameter estimates in the limit, but the standard errors are compromised, and estimates might be affected for small sample sizes.
- *Linearity* indicates a misspecified model, and therefore the results are not meaningful.

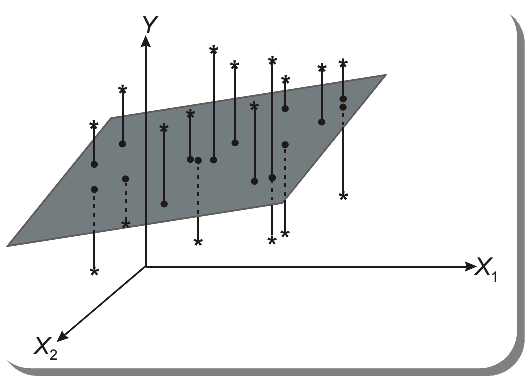
6

Picturing the Model — No Relationship



7

Picturing the Model — A Linear Relationship



8

The Multiple Linear Regression Model

In general, you model the dependent variable Y as a linear function of *k* independent variables (the Xs) as the following:

$$Y = \beta_0 + \beta_1 X_1 + \ldots + \beta_k X_k + \varepsilon$$

The relationship is linear in the parameters.

Transformations of target and inputs are supported by the model. For example:

$$\log(Y) = \beta_0 + \beta_1 X_1 + \beta_2 X_1^2 + \beta_3 \log(X_2) + \varepsilon$$

9

continued...

The Multiple Linear Regression Model

Partial *t* tests

Null Hypothesis: $\beta_i = 0$

Alternative Hypothesis: $\beta_i \neq 0$

Test Statistic: $t = \hat{\beta}_i \,/\, s(\hat{\beta}_i)$

10

continued...

The Multiple Linear Regression Model

Identification

- Determine what inputs are correlated with the target.
  - An input can be correlated with the target and be redundant.
  - An input can be correlated with the target through a relationship with one or more other inputs.
- Determine an appropriate relationship between inputs and target.

Estimation

- Least Squares ≡ Maximum Likelihood

Forecasting

- Infinite memory ≡ finite memory because the model has no memory of the past

11

continued...

The Multiple Linear Regression Model

Problems and Diagnostics

Identification

- Variable selection
  - Stepwise selection
  - Best subsets selection
  - Others

Estimation

- Multicollinearity: eigenvalues of X'X, VIF
- Influential observations: studentized residuals, DFFITS, DFBETAS, Cook's D, leave-one-out statistics
- Autocorrelation: Durbin-Watson
- Nonconstant variance: diagnostic plots

12

From Ordinary Regression to Time Series Regression

The time series regression model is an extension of the ordinary regression model in which the following conditions exist:

- Variables are observed in time.
- Autocorrelation is allowed.
- The target variable can be influenced by past values of inputs.

13

Time Series Regression Terminology

Ordinary Regressor

- An input variable that only has a concurrent influence on the target variable
  - X at time *t* is correlated with Y at time *t*, X at times before *t* is uncorrelated with Y at time *t*.

Dynamic Regressor

- An input variable that influences the target variable at current and past values:
  - X at times *t*, *t*-1, *t*-2, …, influences Y at time *t*.

Transfer Function

- A function that provides the mathematical relationship between a dynamic regressor and the target variable

14

Time Series Regression

Multiple Regression

$$Y = \beta_0 + \beta_1 X_1 + \ldots + \beta_k X_k + \varepsilon$$

Time Series Regression with Ordinary Regressors

$$Y_t = \beta_0 + \beta_1 X_{1t} + \ldots + \beta_k X_{kt} + \varepsilon_t$$

Time Series Regression with Dynamic Regressors

$$\begin{aligned} Y_t = \beta_0 &+ \omega_{10} X_{1,t} + \omega_{11} X_{1,t-1} + \cdots + \omega_{1m_1} X_{1,t-m_1} \\ &+ \omega_{20} X_{2,t} + \omega_{2,t} X_{2,t-1} + \cdots + \omega_{2m_2} X_{2,t-m_2} \\ &+ \cdots \\ &+ \omega_{k0} X_{k,t} + \omega_{k1} X_{k,t-1} + \cdots + \omega_{km_k} X_{k,t-m_k} + \varepsilon_t \end{aligned}$$

15

The Time Series Regression Error Term

In the ordinary regression model, the error term ε_t is white noise. For any time series regression extensions of the ordinary regression model, the error term is a stationary process that can be approximated by an ARMA model,

$$(1 - \phi_1 B - \cdots - \phi_p B^p)\varepsilon_t = (1 - \theta_1 B - \cdots - \theta_q B^q)\eta_t$$

where η_t is a white noise process. Because, for example,

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

the time series regression model is often written as

$$(1 - \phi_1 B - \cdots - \phi_p B^p)(Y_t - \beta_0 - \beta_1 X_{1t} - \beta_2 X_{2t}) = (1 - \theta_1 B - \cdots - \theta_q B^q)\eta_t$$

16

The use of notation ε_t for the stationary component and η_t for the white noise component is nonstandard for time series regression models and is only used in the above slide to show the transition from ordinary regression to time series regression. The stationary error or irregular component is usually expressed as a time series Z_t, and the white noise error is usually expressed as ε_t.

The Time Series Regression Error Term

- The error term is a stationary Gaussian process.
- The distribution of the error term is approximated by an ARMA(*p,q*) model.
- Proper specification of the error model is necessary to obtain the most accurate forecasts.
- Classical tests used in regression analysis for serial autocorrelation of the errors, for example, the Durbin-Watson test, are replaced by more general tests, such as the Ljung-Box test.

17

Coefficients ⇨ Transfer Functions

Ordinary Regression

$$X_{1,t} \Rightarrow \beta_1 \Rightarrow Y_t$$

Time Series Regression

$$X_{1,t} \Rightarrow \frac{\omega(B)}{\delta(B)} \Rightarrow Y_t$$

Transfer Functions

18

The General Rational Polynomial Transfer Function

$$X_{i,t} \Rightarrow \frac{\omega_i(B)}{\delta_i(B)} \Rightarrow Y_t,$$

$$\omega_i(B) = \omega_{i0} - \omega_{i1}B - \cdots - \omega_{im_i}B^{m_i},$$

$$\delta_i(B) = 1 - \delta_{i1}B - \cdots - \delta_{ih_i}B^{h_i}$$

$$\frac{\omega_i(B)}{\delta_i(B)} = \eta_{i0} - \eta_{i1}B - \eta_{i2}B^2 - \cdots - \eta_{ik}B^k - \cdots$$

Truncation Point ⇨ $\eta_{i,k+1}$ < **Tolerance**

19

continued...

The General Rational Polynomial Transfer Function

Motivation

- The ratio of the numerator and denominator polynomials is an infinite order polynomial that accommodates a finite order approximation.
- A ratio of polynomials requires fewer parameters to model the input than a pure numerator polynomial alone ($k >> h+m+1$).
- The length of time series in the dynamic regression problem prevents estimating many parameters.
- The rational transfer function helps solve the problem of limited data.

20

Types of Regressors—Measurement Scale

Binary (dummy) variables

- Take the value zero or one
- Can be used to quantify nominal data

Categorical variables

- Nominal scaled ⇨ nonquantitative categories
- Ordinal scaled variables can be treated as categorical.
- They must be coded into a quantitative input, usually using a form of dummy coding for each level (less one if a constant term is used in the model).

Quantitative variables

- Interval or ratio scaled
- Can be transformed

21

Types of Regressors—Randomness

Deterministic

- Controlled by experimenter
- Alternatively, can be perfectly predicted without error

Stochastic

- Governed by unknown probability distributions
- Cannot be perfectly predicted

22

Types of Regressors

Deterministic examples

- Dummy coding for holiday events
- Settings on a machine, for example, electric current, temperature, and pressure on production equipment
- Intervention weights, for example, saturation for legislation that is phased in uniformly by month over a year: 1/12, 2/12, 3/12,…,12/12
- Advertising expenditures
  (These can be treated as stochastic when decisions are influenced by stochastic factors, such as market share, promotions by competitors, and so on.)

23

continued...

Types of Regressors

Stochastic examples

- Ambient outside air temperature
- Competitor sales
- Interest rates
- Consumer price index
- Unemployment rate
- Rate per 1000 households of television viewership
- Stock market indices

24

SAS/ETS Time Series Regression Procedures

AUTOREG

- Ordinary regressors
- AR errors
- No differencing options
- Heteroscedastic error
- Estimation methods ML, ULS, YW, ITYW

ARIMA

- Ordinary and dynamic inputs
- ARMA errors
- Differencing of any orders
- Homogeneous error
- Estimation methods CLS, ML, ULS

25

ARIMA Syntax for Time Series Regression

```
PROC ARIMA DATA=SAS-data-set <options>;
     IDENTIFY VAR=target-variable<(d1 d2…)>
                  CROSS=(input-1<(d11 d12…)>
                         input-2<(d21 d22…)>
                           …
                         input-k<(dk1 dk2…)>
                  <options>;
     ESTIMATE
                P=n|(n11 n12…)<(n21 n22…)(n31 n32…)…>
                Q=n|(n11 n12…)<(n21 n22…)(n31 n32…)…>
                INPUT=(<n$><(n11 n12…)(n21 n22…)…>
                       </(n11 n12…)(n21 n22…)…>input-1…)
                <options>;
```

26

The rational transfer function involves a shift, a numerator polynomial, and a denominator polynomial. For a shift of three time units, a numerator polynomial with two parameters and a denominator polynomial with one parameter, the specification is **3$(1)/(1)**. A parameter for the numerator lag zero term is always implied.

ARIMA Syntax for Time Series Regression

```
proc arima data=work.GulfOilGas;
   identify var=oil(1 12)
                cross=(PRICE(1 12)
                       ISIDORE(1 12)
                       MITCH(1 12)
                       RITA(1 12))
                nlags=24;
   estimate q=(1)(12)
            input=(1$PRICE
                   /(1)ISIDORE
                   MITCH
                   /(1)RITA)
            method=ml plot;
   forecast id=Date interval=month
            lead=12
            out=work.TempOilF;
```

27

The syntax /(1) is equivalent to (0)/(1) and implies a single numerator parameter at lag zero and a single denominator parameter at lag one.

The Cross-Correlation Function (CCF)

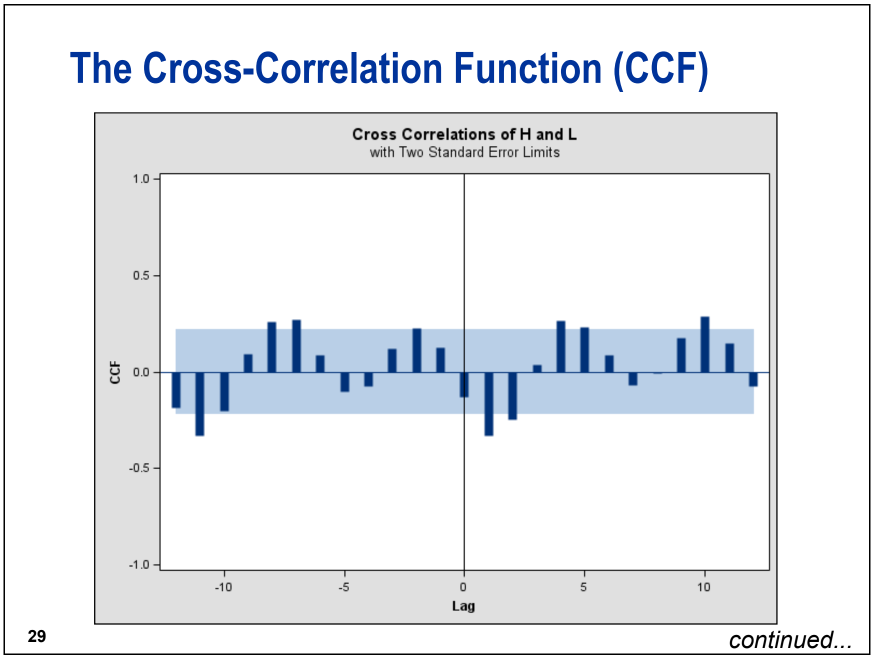
CCF(k) is the cross-correlation of target Y with input X at lag k.

- The response variable is always listed first.
- A significant value at lag k implies that Y_t and X_{t-k} are correlated.
- Spikes and decay patterns in the cross-correlation function can help determine the form of the transfer function.
- The sample CCF estimates an unknown population CCF.

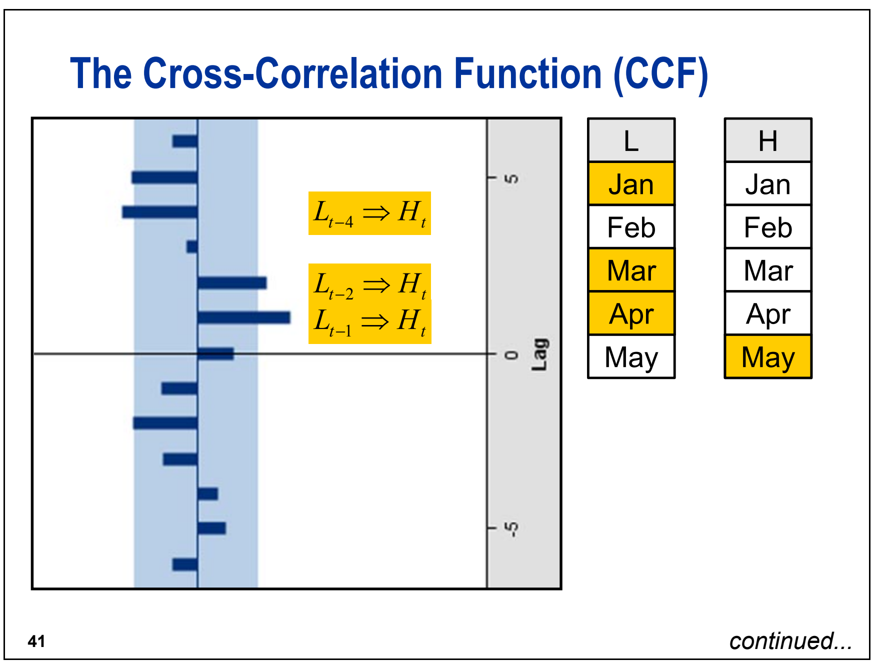
28

continued...

The cross-correlation function works best when the two time series are *prewhitened*. Otherwise, trend, seasonality, and cross-correlations with other input variables can contaminate the CCF and make it almost impossible to identify an appropriate transfer function without some trial and error. (Prewhitening is discussed in a later section.)



The above CCF exhibits spikes at lags 1, 2, 4, 5, and 9. Spikes also occur for negative lags 2, 7, 8, and 11. A spike at a negative lag implies that the target variable depends on future values of the input variable. While this might be useful to know, it usually cannot be used in a model without adding too much variability in the forecast. If the lead-lag/lag-lead relationships are real, procedures such as VARMAX and STATESPACE accommodate a more general multivariate approach to the problem.



Spikes at lags 1, 2, and 4 imply that input L affects target H through past values at 1, 2, and 4 lags. Hence, the value of H in May is influenced by the past values of L in April, March, and January.

The Cross-Correlation Function (CCF)

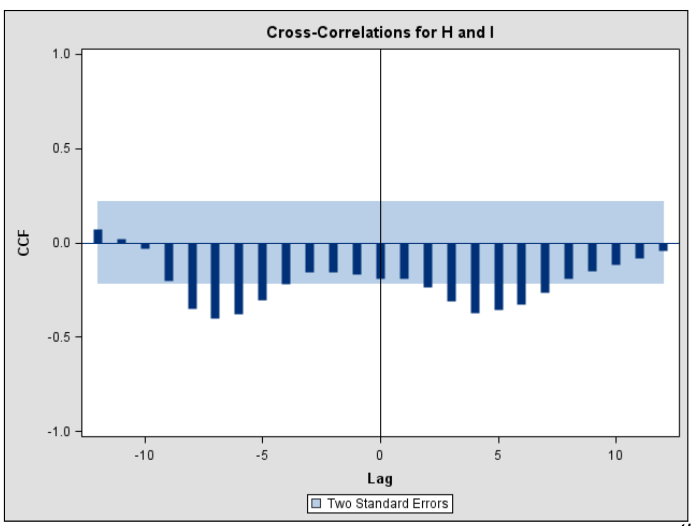
```
ods graphics on;
proc timeseries data=work.CCFexample
                out=work.temp
                plot=(series)
                crossplots=(ccf)
                seasonality=12;
   id Date interval=month;
   var H;
   crossvar I L;
   crosscorr ccf / nlag=12;
run;
ods graphics off;
```

42

continued...

Cross-correlation plots and values can be obtained from PROC TIMESERIES and PROC ARIMA. Only PROC ARIMA can apply *filtering* operations, such as differencing and prewhitening, to the two time series of interest.

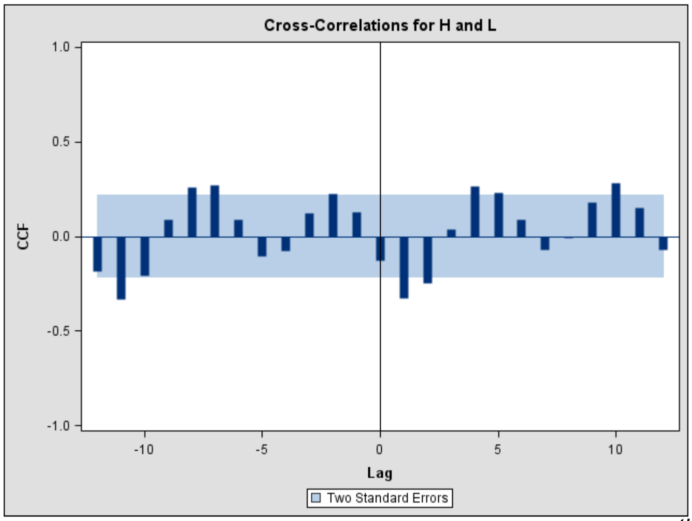
The Cross-Correlation Function (CCF)



43

continued...

The Cross-Correlation Function (CCF)



44

continued...

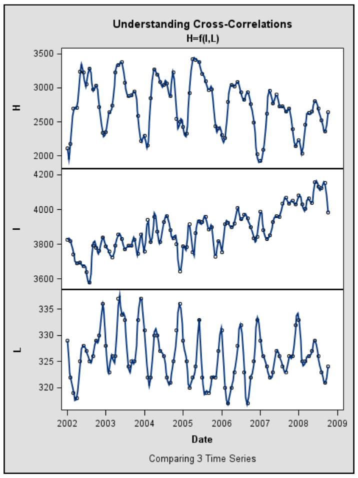
The Cross-Correlation Function (CCF)

```
proc sgscatter data=work.CCFexample;
   compare y=(H I L) x=Date /
           pbspline=(smooth=0.0001);
run;
```

45

continued...

The Cross-Correlation Function (CCF)

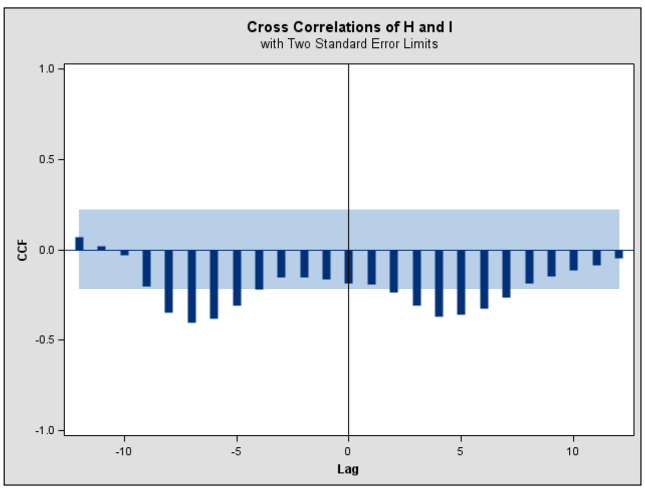


- Two time series with trend will usually appear to be correlated.
- Two time series with seasonal fluctuations will usually appear to be correlated.
- Trend and seasonal components should be removed before calculating the CCF.

46

continued...

The Cross-Correlation Function (CCF)

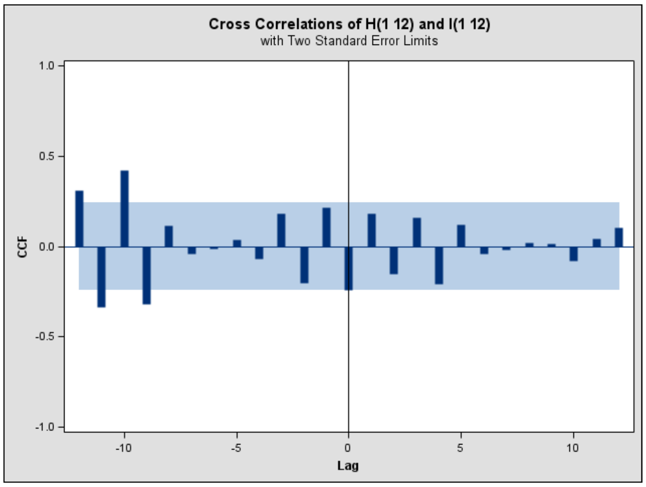


Before removing trend and seasonal components

47

continued...

The Cross-Correlation Function (CCF)



After removing trend and seasonal components

48

continued...

The Cross-Correlation Function (CCF)

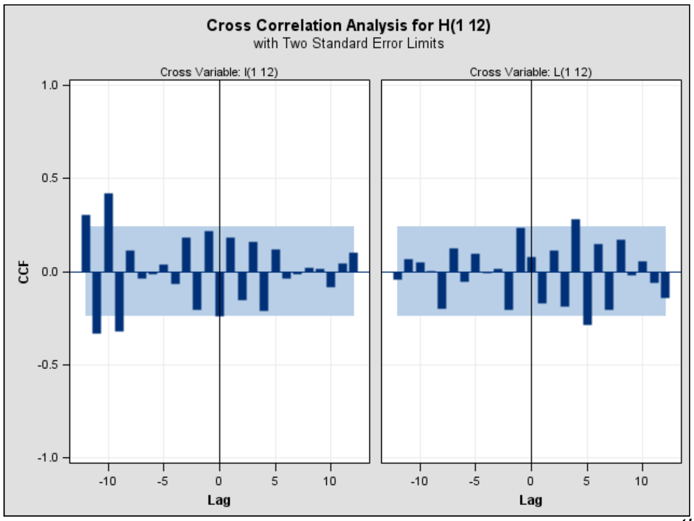
```
ods graphics on;
proc arima data=work.CCFexample plots=all;
   identify var=H(1 12)
            cross=( I(1 12) L(1 12) )
            nlags=12;
run;
quit;
ods graphics off;
```

PROC ARIMA enables you to difference to remove trend and seasonality before calculating the CCF.

49

continued...

The Cross-Correlation Function (CCF)



50

continued...

The Cross-Correlation Function (CCF)

Reality — The three series are these:

- Housing Starts (H) for the U.S.
- Motor Vehicle Injuries (I) occurring in a large U.S. metropolitan area
- Lowest Tide Gauge Mark (L) for a San Francisco monitoring station

For this data, any significant correlation remaining after de-trending and de-seasonalizing is likely to be spurious. Tide gauge height is unlikely to be predictive of housing starts!

(Series were shifted and scaled.)

51

The cross-correlation function is the primary tool for identifying a transfer function that relates an input variable to the target. The diagnostic ability of the CCF is best when the two time series are prewhitened, but this only helps when all inputs that influence the target are uncorrelated with each other. When inputs are correlated, the CCF exhibits the influence of the given input on the target, but the CCF also exhibits the influence of the correlated variables simultaneously on the target. In such circumstances, the CCF at best can identify possible lead-lag relationships but usually provides no help in diagnosing the form of the transfer function.

From Ordinary Regression to Dynamic Regression

This demonstration illustrates how to fit a standard multiple regression model to time series data using PROC REG from SAS/STAT, and how to fit a dynamic regression model to time series data using PROC ARIMA.

The code for this demonstration can be found in **Demo5_01WorldOil.sas**.

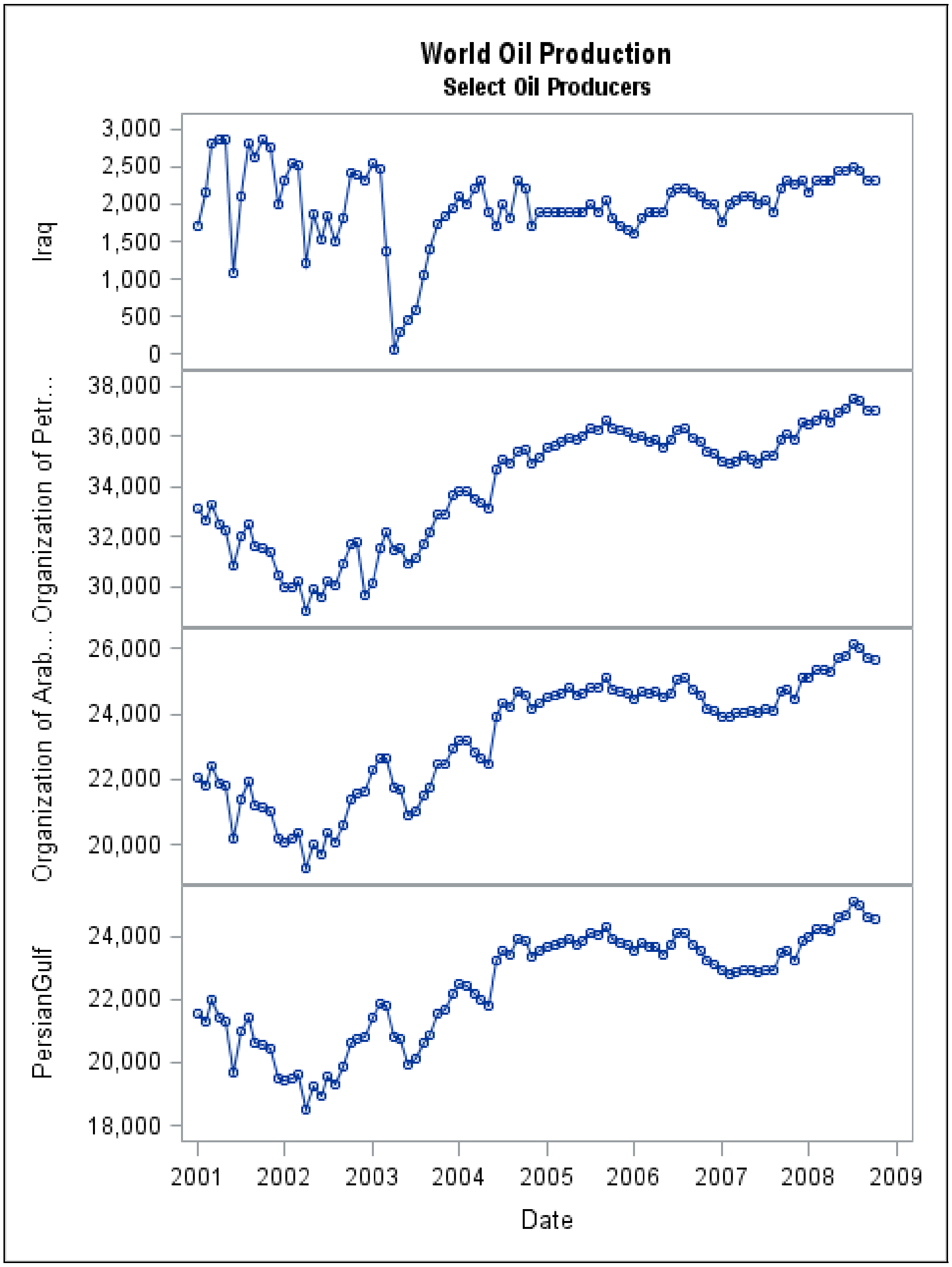
World oil production data can be obtained from the United States Department of Energy, Energy Information Administration Web site, **http://www.eia.gov/**.

To limit the size of this demonstration, only eight oil production time series are considered: Canada, Iraq, Mexico, OAPEC, OPEC, Persian Gulf, USA, and Venezuela. The scenario considers a business analyst of a U.S.-based organization that wants to incorporate energy issues into planning for growth and expansion. The U.S. production of oil can have a significant effect on business operations, especially if oil production on the international scene is limited by political action or conflict.

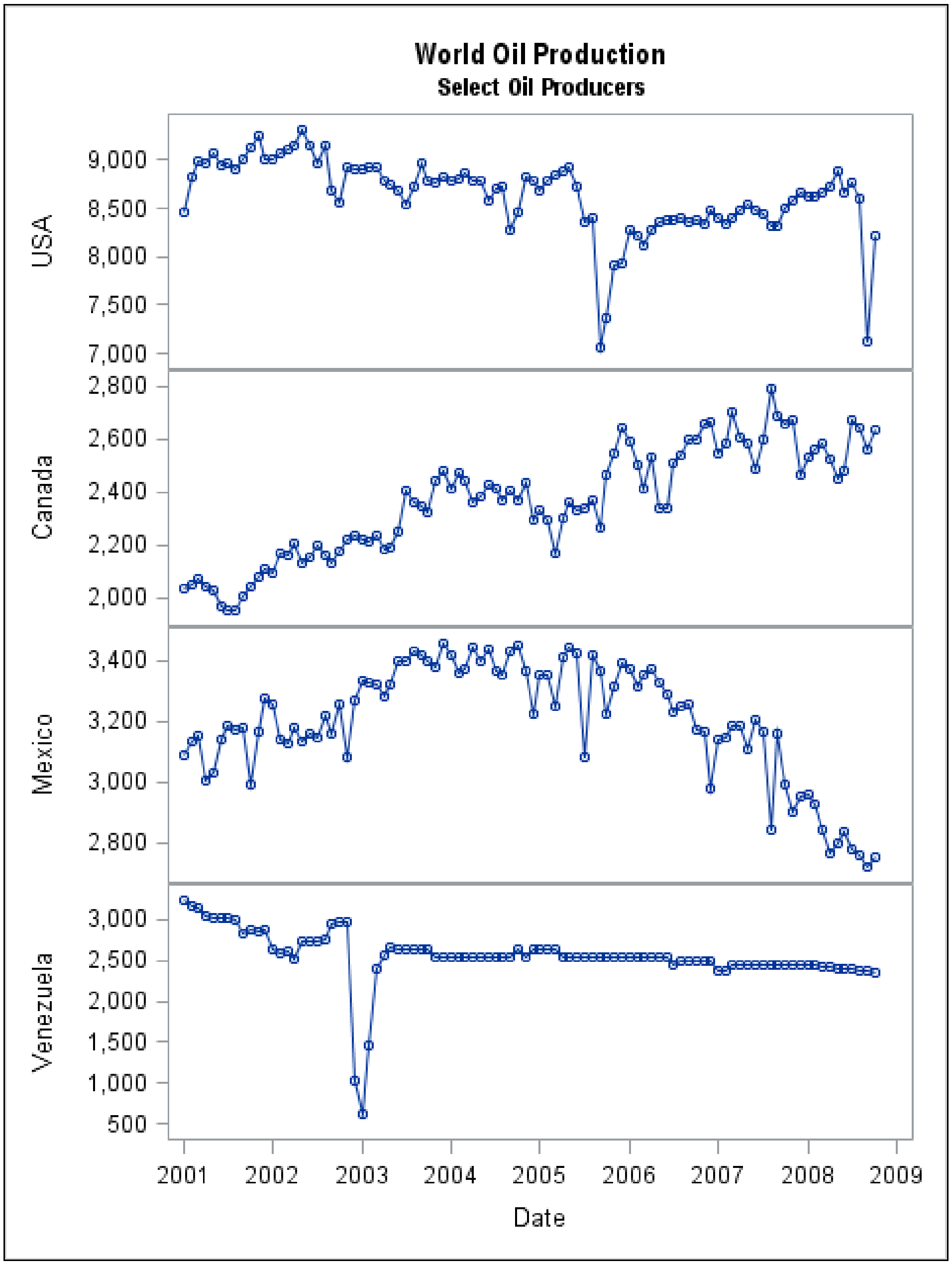
Oil production values are mean daily oil production in thousands of barrels accumulated monthly. The following code provides two sets of comparison plots, four per page:

```
proc sgscatter data=sasuser.WorldOil;
   compare y=(Iraq OPEC OAPEC PersianGulf)
           x=Date / join;
run;
proc sgscatter data=sasuser.WorldOil;
   compare y=(USA Canada Mexico Venezuela)
           x=Date / join;
run;
```

Graphs of world oil production show very similar patterns for three petroleum exporting organizations so that a high degree of collinearity is expected. These organizations contain many of the same countries.



The western hemisphere countries show less collinearity and a few severe drops in production.



Multicollinearity, or simply collinearity, is a property of the predictor variables, and as such, does not depend on error assumptions such as independence. It refers to correlation among the predictor variables. Using PROC REG to diagnose multicollinearity is a reasonable approach. One undesirable effect of multicollinearity is an increase in the variance of parameter estimates. PROC REG delivers a *variance inflation factor*, VIF, that compares an overall measure of variance in the parameter estimates to what it would be if there were no collinearity in the predictors. Marquardt (1970) suggests that VIF>10 is an indication of multicollinearity problems. The VIF is calculated as follows:

For each predictor variable, regress that variable on the other predictors obtaining R square. The VIF is $1/(1-R^2)$. An R square near 1 indicates that the predictor being studied is almost a linear function of the others. That is the definition of multicollinearity. Here the U.S. oil production is taken to be the dependent variable. Not surprisingly, the OPEC, OAPEC, and Persian Gulf organizations, which contain some of the same countries, show large VIF statistics much larger than 10.

The following code uses PROC REG in SAS/STAT to fit an ordinary regression model using **USA** oil production as the target variable, and using the remaining seven oil production variables as predictor variables:

```
proc reg data=sasuser.WorldOil
         outest=work.EstFullModel;
   model USA=Canada Iraq Mexico OAPEC
             OPEC PersianGulf Venezuela /
             vif aic sbc;
quit;
```

The parameter estimates table with the high VIF values is shown below:

Parameter Estimates							
Variable	**Label**	**DF**	**Parameter Estimate**	**Standard Error**	**t Value**	**Pr > |t|**	**Variance Inflation**
Intercept	Intercept	1	12225	1022.97654	11.95	<.0001	0
Canada	Canada	1	-0.02571	0.33926	-0.08	0.9398	5.17275
Iraq	Iraq	1	0.12506	0.07723	1.62	0.1090	1.63011
Mexico	Mexico	1	-0.13724	0.21495	-0.64	0.5249	1.77198
OAPEC	Organization of Arab Petroleum Exporting Countries	1	0.32652	0.34570	0.94	0.3476	404.50623
OPEC	Organization of Petroleum Exporting Countries	1	-0.35791	0.15085	-2.37	0.0199	133.11933
PersianGulf	PersianGulf	1	0.01204	0.30509	0.04	0.9686	277.11510
Venezuela	Venezuela	1	0.37970	0.22075	1.72	0.0890	6.17486

One approach is to fit three models, each having only one of the three collinear variables. Using ODS to output the fit statistics, you see that the model using OPEC is the best on all diagnostics.

The following code fits the three models. (Notice that the last TEST statement relates to the most recent previous MODEL statement.)

```
ods output ParameterEstimates=work.ParmEst5V
           FitStatistics=work.FitStat5V;
proc reg data=sasuser.WorldOil
         outest=work.Est5VModel;
   OAPEC: model USA=Canada Iraq Mexico
                    OAPEC Venezuela /
                    aic sbc edf;
   PGulf: model USA=Canada Iraq Mexico
                    PersianGulf Venezuela /
                    aic sbc edf;
   OPEC:  model USA=Canada Iraq Mexico
                    OPEC Venezuela /
                    VIF aic sbc edf dwprob;
   Omit_Can_Ven_Mex: test Canada=0, Mexico=0, Venezuela=0;
   title3 f=&Coursefont  "Warning: Tests Assume Independent Errors";
quit;
ods output close;
```

The following code prints the fit statistics:

```
proc print data=work.Est5VModel noobs;
   var _MODEL_ _RMSE_ _RSQ_ _AIC_ _SBC_;
   format _RMSE_ 6.1 _RSQ_ 6.3 _AIC_ _SBC_ 7.1;
run;
```

The table is shown below:

MODEL	_RMSE_	_RSQ_	_AIC_	_SBC_
OAPEC	303.4	0.435	1080.2	1095.5
PGulf	303.4	0.435	1080.2	1095.5
OPEC	299.0	0.452	1077.5	1092.7

When you investigate the OPEC model, three points emerge:

1. The remaining VIF values are quite reasonable.

Parameter Estimates							
Variable	Label	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|	Variance Inflation
Intercept	Intercept	1	12526	977.10948	12.82	<.0001	0
Canada	Canada	1	-0.20649	0.25721	-0.80	0.4242	2.93980
Iraq	Iraq	1	0.12916	0.07139	1.81	0.0738	1.37721
Mexico	Mexico	1	-0.11673	0.18553	-0.63	0.5309	1.30520
OPEC	Organization of Petroleum Exporting Countries	1	-0.09968	0.02100	-4.75	<.0001	2.55167
Venezuela	Venezuela	1	0.04247	0.10236	0.41	0.6792	1.31277

2. The Durbin-Watson statistic, DW= $\sum_2^n (r_t - r_{t-1})^2 / \sum_1^n r_t^2$, is significantly less than 2, its theoretical value for white noise, which indicates a significantly greater than 0 lag 1 autocorrelation. The lag 1 autocorrelation is estimated as 0.495.

Durbin-Watson D	0.992
Pr < DW	<.0001
Pr > DW	1.0000
Number of Observations	94
1st Order Autocorrelation	0.495

3. Not all variables appear to be significant based on the *p*-values of their *t* statistics. The joint test of all three suggests that they can all be omitted without significant loss in estimation.

Test Omit_Can_Ven_Mex Results for Dependent Variable USA				
Source	DF	Mean Square	F Value	Pr > F
Numerator	3	51968	0.58	0.6287
Denominator	88	89375		

Such inference depends on error assumptions and is technically incorrect when autocorrelation is present. Because the Durbin-Watson test indicates that autocorrelation is a problem, you should be cautious in interpreting test statistics and *p*-values. From a practical perspective, forecasting projects are often constrained by time, and you might be forced to rely on questionable results to obtain good forecasts in a reasonable time. The saying, “A good model today is better than a great model tomorrow,” reflects the practical side of forecasting under time constraints. However, you can see that PROC AUTOREG matches much of the functionality of PROC REG while enabling you to adjust for autocorrelation. When you transition from ordinary regression to time series regression, PROC AUTOREG can improve inference without costing extra project time. Unfortunately, PROC REG currently has a richer set of variable selection tools than PROC AUTOREG, so some sacrifices might have to be made.

```
proc autoreg data=sasuser.WorldOil;
   model USA=Canada Iraq Mexico OPEC Venezuela
             / nlag=10 backstep;
   Test Canada=0, Mexico=0, Venezuela=0;
run;
```

The table of estimates provides reliable test statistics and *p*-values if the derived AR(1) error component successfully models the autocorrelation.

Parameter Estimates						
Variable	**DF**	**Estimate**	**Standard Error**	**t Value**	**Approx Pr > |t|**	**Variable Label**
Intercept	1	11503	1253	9.18	<.0001	
Canada	1	0.2434	0.3188	0.76	0.4473	Canada
Iraq	1	0.1492	0.0793	1.88	0.0632	Iraq
Mexico	1	0.0402	0.2332	0.17	0.8634	Mexico
OPEC	1	-0.1239	0.0298	-4.16	<.0001	Organization of Petroleum Exporting Countries
Venezuela	1	0.1349	0.1180	1.14	0.2562	Venezuela

The *p*-values for Iraq and Venezuela become smaller, while the *p*-values for Canada and Mexico become larger. The three variable test *p*-value changes, but the conclusion is the same.

Test 1				
Source	**DF**	**Mean Square**	**F Value**	**Pr > F**
Numerator	3	31173	0.47	0.7030
Denominator	87	66121		

A careful elimination of these terms one at a time, accounting for autocorrelation by using PROC AUTOREG, confirms that they can all be omitted. A lag 1 autoregressive coefficient near 0.5, as expected from the Durbin Watson results, is observed throughout this process. Next the model with Iraq and OPEC is fitted in PROC AUTOREG. The initial least squares fit is the same thing that would be obtained in PROC REG. As with PROC REG, *p*-values are unjustified because the error correlation was not addressed. The initial least squares parameter estimates table is shown in the PROC AUTOREG output for comparison to the final estimation results that include autocorrelation. You must be careful to use the final parameter estimates table to get parameter estimates adjusted for autocorrelation.

The following code fits the two-variable model:

```
proc autoreg data=sasuser.WorldOil;
   model USA=Iraq OPEC / nlag=10 backstep;
   output out=work.OutAR_Oil residual=Resid;
run;
```

The initial ordinary least squares parameter estimates table appears below:

Parameter Estimates						
Variable	DF	Estimate	Standard Error	t Value	Approx Pr > |t|	Variable Label
Intercept	1	12151	446.8247	27.19	<.0001	
Iraq	1	0.1569	0.0617	2.54	0.0127	Iraq
OPEC	1	-0.1125	0.0133	-8.43	<.0001	Organization of Petroleum Exporting Countries

The final estimates table appears below:

Parameter Estimates						
Variable	DF	Estimate	Standard Error	t Value	Approx Pr > |t|	Variable Label
Intercept	1	12039	745.6423	16.15	<.0001	
Iraq	1	0.1201	0.0731	1.64	0.1038	Iraq
OPEC	1	-0.1071	0.0226	-4.75	<.0001	Organization of Petroleum Exporting Countries

The estimated AR(1) parameter is shown below:

Estimates of Autoregressive Parameters			
Lag	Coefficient	Standard Error	t Value
1	-0.517101	0.090222	-5.73

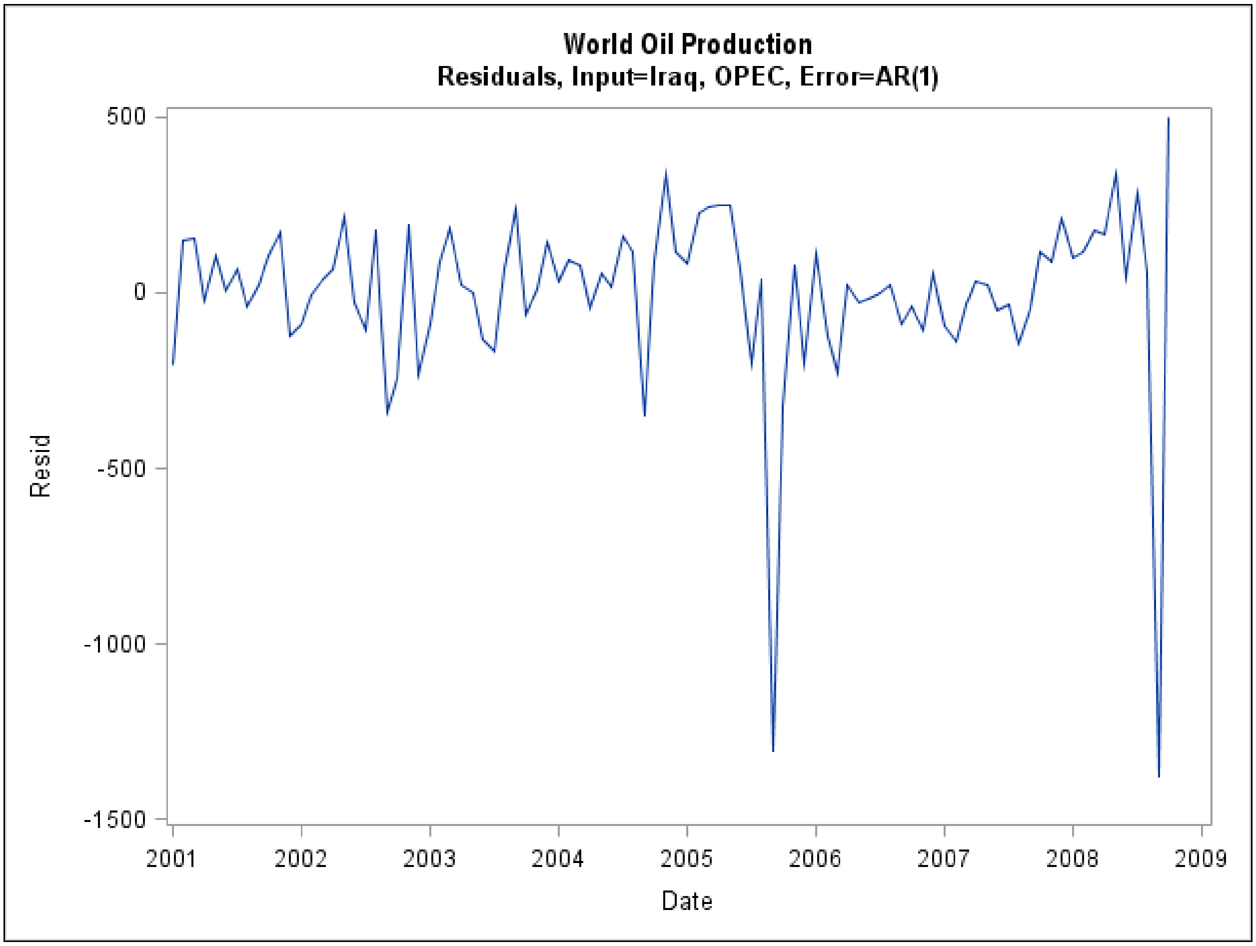
The lag 1 autocorrelation for the least squares residuals is significantly different from 0 and not close to 1. Ten lags were presented to the procedure but only lag 1 was chosen. While not essential, the finding of only one or a few contiguous autoregressive coefficients is somewhat reassuring in terms of model specification. Recall that the parameterization of PROC AUTOREG produces AR values that are opposite in sign to those produced by PROC ARIMA.

The lag 1 autocorrelation is used to produce the final generalized least squares estimates. In this case, specification of a correlated error structure renders the Iraq effect insignificant. This shows the importance of accounting for autocorrelation before hypothesis testing. This is especially interesting in that the autoregressive model is not particularly close to a unit root case where the effects on test statistics are more dramatic. These models can, of course, be fit in PROC ARIMA as well but with no moving averages, lagged input effects, or differencing, PROC AUTOREG is an option as illustrated here.

An output data set that captured the model residuals was created. A plot of the residuals can be obtained using the following code:

```
proc sgplot Data=work.OutAR_Oil;
   series x=Date Y=Resid;
run;
```

The plot is shown here:



The question arises as to whether this is a proper model. A graph of the residuals shows two extremely low residuals. The following code enables you to identify the dates when the extreme residuals occur:

```
proc print data=work.OutAR_Oil;
   where Resid<-500;
   var Resid Date;
run;
```

The first occurs in September 2005 and the second in September 2008. Hurricanes commonly occur in September. These residuals do not appear to come from the same distribution as the rest, which is a violation of assumptions. Proper inference about Iraq requires that this issue be dealt with. The next section addresses dealing with events.

Events and Outliers

- If an event is not identified, PROC ARIMA can detect outliers that might represent events.
- Three types of outliers are included in the search: **ADDITIVE** outlier (**AO**), level **SHIFT** (**LS**), and **TEMP**orary change (**TC**). (ADDITIVE, SHIFT, and TEMP are the primary keywords, and AO, LS, and TC are accepted variants.)

```
PROC ARIMA <options>;
      IDENTIFY VAR=variable <options>;
      ESTIMATE <options>;
      OUTLIER TYPE=(AO|LS|TC) <options>;
      FORECAST OUT=SAS-data-set <options>;
RUN;
```

53

Detecting outliers can lead to the discovery of previously unknown events. This was the premise of the previous demonstration, that two unusual residuals might correspond to known omitted from the model.

Events and Outliers in the World Oil Time Series

This demonstration illustrates how to use the OUTLIER statement in PROC ARIMA to identify unusual observations, and how to specify event variables for forecast models to improve forecasts for the **WorldOil** data.

The code for this demonstration can be found in **Demo5_02WorldOil.sas**.

The following code formalizes the search for unusual observations in USA oil production:

```
proc arima data=sasuser.WorldOil plots=all;
   identify var=USA crosscorr=(OPEC Iraq) noprint;
   estimate p=1 input=(OPEC Iraq) ML noprint;
   outlier type=(ls ao) maxnum=10 id=Date;
quit;
```

The following table reveals four outliers:

Outlier Details

Obs	Time ID	Type	Estimate	Chi-Square	Approx Prob>ChiSq
93	SEP2008	Additive	-1291.7	90.46	<.0001
57	SEP2005	Additive	-892.55530	46.05	<.0001
58	OCT2005	Additive	-653.29801	24.67	<.0001
82	OCT2007	Shift	198.63469	6.28	0.0122

The outliers can be explained as follows: **SEP2005** is due to hurricanes Katrina and Rita, **OCT2005** is the prolonged effect of Katrina and Rita, and **SEP2008** is due to hurricane Ike. The positive level shift in October 2007 cannot be explained. To avoid possible overfitting, the outlier corresponding to **OCT2007** is ignored for now.

Indicator (dummy) variables for hurricanes Katrina (Aug. 2005), Rita (Sept. 2005), and Ike (Sept. 2008) are presented to PROC AUTOREG as additional inputs. The code for creating the event variables for these three hurricanes follows:

```
data work.WorldOil;
   set sasuser.WorldOil;
   attrib Katrina length=3 label="Hurricane Katrina"
          Rita    length=3 label="Hurricane Rita"
          Ike     length=3 label="Hurricane Ike";
   Katrina=("01AUG2005"d<=Date<="31AUG2005"d);
   Rita=("01SEP2005"d<=Date<="30SEP2005"d);
   Ike=("01SEP2008"d<=Date<="30SEP2008"d);
run;
```

A regression model can be fit using PROC AUTOREG.

```
proc autoreg data=work.WorldOil;
   model USA=OPEC Iraq Katrina Rita Ike / nlag=1;
run;
```

Surprisingly Katrina, one of the most devastating storms to hit the U.S., is only marginally significant, and the estimate for Katrina is positive!

Parameter Estimates						
Variable	DF	Estimate	Standard Error	t Value	Approx Pr > |t|	Variable Label
Intercept	1	11662	590.6709	19.74	<.0001	
OPEC	1	-0.0955	0.0179	-5.33	<.0001	Organization of Petroleum Exporting Countries
Iraq	1	0.1186	0.0500	2.37	0.0200	Iraq
Katrina	1	277.5604	160.8780	1.73	0.0880	Hurricane Katrina
Rita	1	-744.3300	161.3044	-4.61	<.0001	Hurricane Rita
Ike	1	-1300	143.7862	-9.04	<.0001	Hurricane Ike

Could this be a multicollinearity problem? Notice also that with the addition of these hurricane indicator variables, Iraq again becomes significant. Also, the AR(1) parameter is estimated to be 0.61. The same model could be fit using PROC ARIMA.

```
proc arima data=WorldOil;
   identify var=USA
            cross=(OPEC Iraq Katrina Rita Ike);
   estimate p=1 input=(OPEC Iraq Katrina Rita Ike) ml;
quit;
```

The parameter estimates table looks different and includes the AR coefficient.

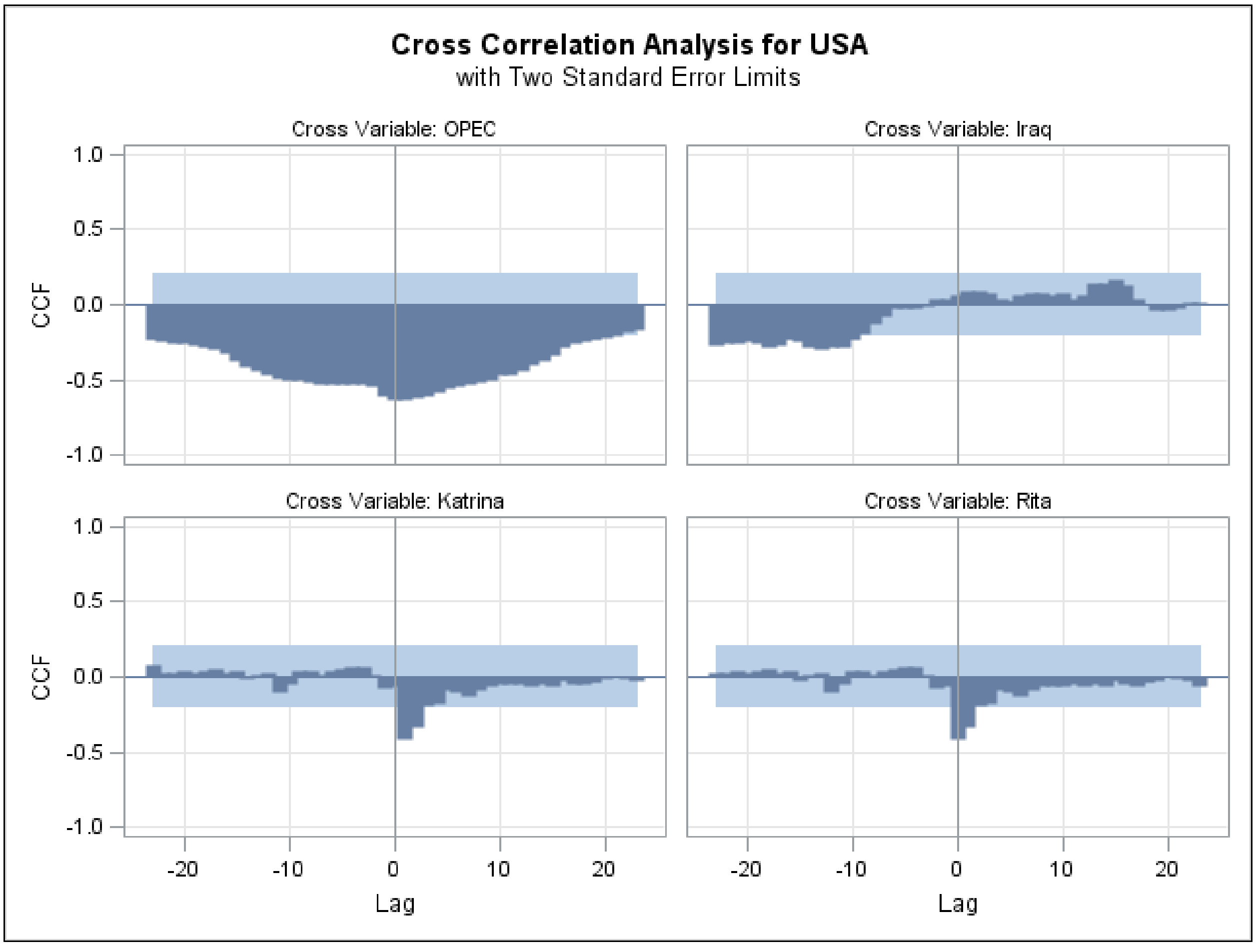
Maximum Likelihood Estimation							
Parameter	Estimate	Standard Error	t Value	Approx Pr > |t|	Lag	Variable	Shift
MU	11360.9	800.96417	14.18	<.0001	0	USA	0
AR1,1	0.75372	0.07075	10.65	<.0001	1	USA	0
NUM1	-0.08579	0.02433	-3.53	0.0004	0	OPEC	0
NUM2	0.09932	0.05060	1.96	0.0497	0	Iraq	0
NUM3	337.74187	150.28812	2.25	0.0246	0	Katrina	0
NUM4	-667.67976	150.67947	-4.43	<.0001	0	Rita	0
NUM5	-1301.3	131.32155	-9.91	<.0001	0	Ike	0

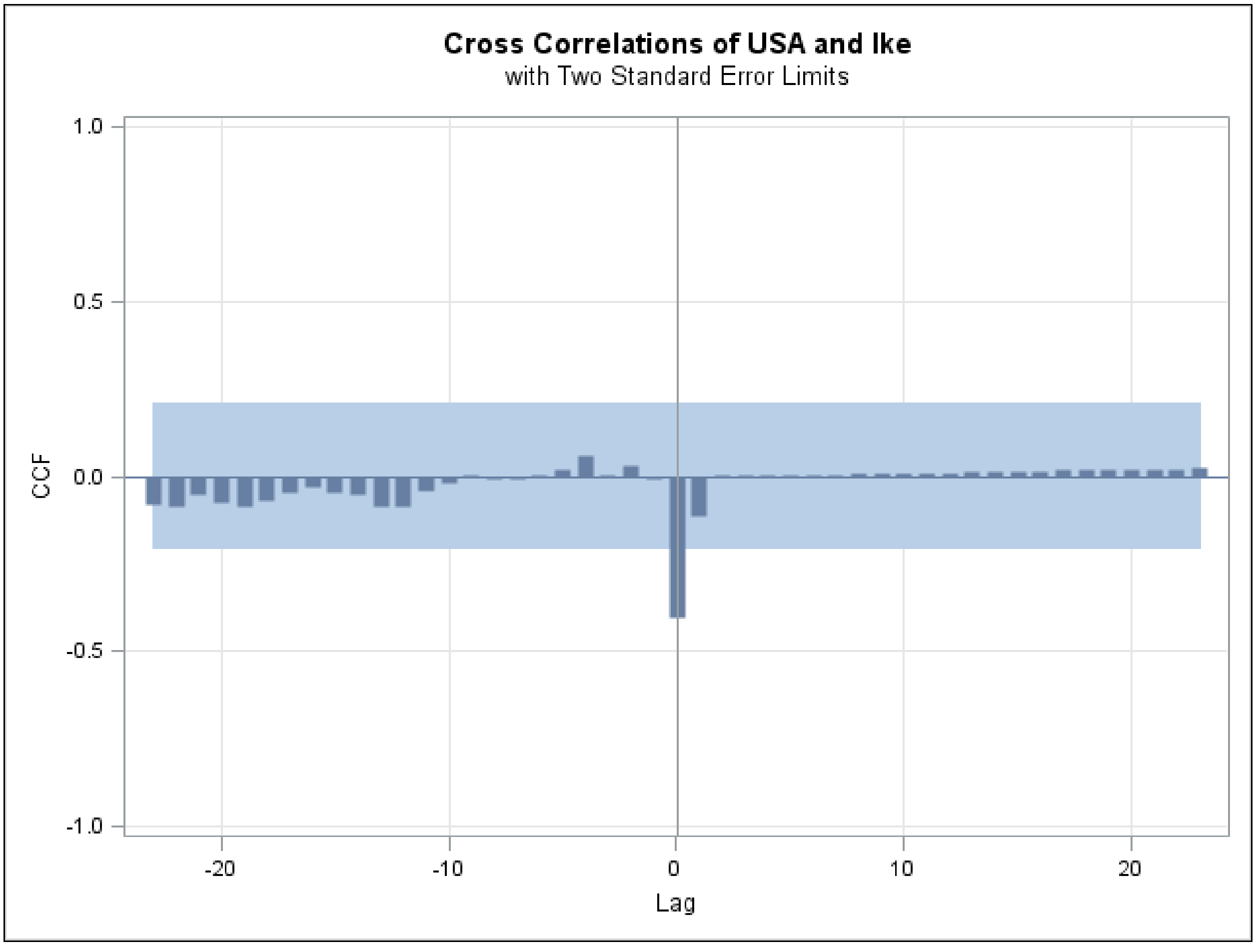
The maximum likelihood estimates are different than the Yule-Walker/Least Squares estimates coming from PROC AUTOREG, but the results are similar otherwise, except that PROC ARIMA thinks that Katrina had a significant ***positive*** impact on USA oil production! Furthermore, residual analysis from PROC ARIMA leads to a conclusion that this model fits the data well!

A little research provides guidance for building a model. Katrina had its most devastating effect on the Gulf of Mexico region in the last week of August 2005. By that time, approximately 75% of the gulf oil production was completed. Rita struck three weeks after Katrina in September. This provides a behavioral explanation for the multicollinearity between Katrina and Rita.

The CROSSCORR option (abbreviated CROSS) in the IDENTIFY statement produces the sample cross-correlation function between the target variable and each input variable named in the CROSSCORR list. (The cross-correlation function was introduced previously.)

Here is each cross-correlation function for each input:





Ike does not fit in the panel of four cross-correlation functions, so it is presented in a single larger plot. Notice that there is correlation at lag 1 between U.S. oil production at time t (for example, September) and the Katrina indicator variable at time t-1 (August is t-1 when t is September) and additional smaller correlations at lags 2 and 3 at least. In fact, the correlation seems to decline exponentially. The same correlation pattern occurs for Rita, but it starts at lag 0, that is, it correlates the oil production (time t = September) with the indicator variable for Rita at time t–0 = September. This is because the lagged indicator variable for Katrina ***is*** the Rita indicator variable, a case of perfect collinearity. The cross correlation could represent an immediate effect of Rita or a lagged effect of Katrina (or both). There is no statistical way to separate the two with this data. Because the Katrina dummy variable is not lagged, the unusual positive effect simply means that there was no negative Katrina effect in the month that Katrina struck. The Ike effect seems to be immediate, although a small (insignificant) lag 1 correlation appears as well. It could be tested for significance in a careful analysis.

As indicated above, the Rita effect is the same as lag 1 of the Katrina effect so a model cannot include both. You do not have to specify the cause of the negative drop as either Katrina or Rita, and in fact, the effect is probably the cumulative effect of both hurricanes. The critical aspect of the analysis is to correct for a known cause in the drop in oil production, and not to infer whether that cause came from Katrina or Rita. Because of the combined effect of both hurricanes (and multicollinearity!), you can drop the reference to Rita and focus on the Katrina event variable. Expected responses for various transfer functions can be compared to data near an intervention point for identification of the transfer function form.

The Katrina cross correlations suggest a delay of one period and exponential decay typical of a denominator backshift on the Katrina indicator variable. The transfer function model component is the following:

$$\frac{\omega}{1-\delta B} X_{t-1}$$

This transfer function can be specified with 1$ for the pure delay of 1 and /(1) for the one denominator lag. In the ESTIMATE statement, it would be specified as follows:

```
INPUT=(1$/(1)Katrina)
```

Of course, other terms also appear in the INPUT option. A closer look at the Katrina cross-correlations shows an initial spike followed by another spike about 90% of the height of the initial one. The next spike is only about 50% of its predecessor but then the approximately 90% decay rate continues. A numerator term at lag 2 adjusts for the drop after the first two nonzero crosscorrelations. Along with the pure delay of one month and the denominator term for exponential decay, the code for this is as follows:

```
INPUT=(1$(2)/(1)Katrina)
```

The transfer function became somewhat confusing. Is there now a three-month delay (shift of one month plus numerator lag of two months) in how Katrina impacts oil production? Does this skip any two-month delayed effects? The best way to understand the nature of the transfer function is to do the math. If you have access to symbolic Algebra software, you can have the software expand the polynomials for you. If you have SAS/IML, you can derive the transfer function in expanded form given the estimated coefficients. This is the purpose of the %PlotPsi macro, which is demonstrated after estimation is performed.

The transfer function for Ike could be only an ordinary regressor type, meaning that you simply multiply the Ike indicator variable by a single coefficient. The cross-correlation plot for Ike only shows a significant result at lag 0, but the small bar at lag 1 suggests adding a delay effect. This means that Ike is a dynamic regressor with a coefficient at lag 0 and at lag 1.

The following code fits the described transfer functions:

```
proc arima data=work.WorldOil;
   identify var=USA
            cross=(Iraq OPEC Katrina Ike)
            noprint;
   estimate input=(Iraq OPEC 1$(2)/(1)Katrina (1)Ike)
            method=ml plot;
quit;
```

The table of estimates follows:

Maximum Likelihood Estimation							
Parameter	Estimate	Standard Error	t Value	Approx Pr > |t|	Lag	Variable	Shift
MU	10453.7	286.76206	36.45	<.0001	0	USA	0
NUM1	0.10845	0.03199	3.39	0.0007	0	Iraq	0
NUM2	-0.05545	0.0088052	-6.30	<.0001	0	OPEC	0
NUM3	-1477.0	113.29745	-13.04	<.0001	0	Katrina	1
NUM1,1	-664.74251	136.20144	-4.88	<.0001	2	Katrina	1
DEN1,1	0.93535	0.01424	65.69	<.0001	1	Katrina	1
NUM4	-1468.2	150.71878	-9.74	<.0001	0	Ike	0
NUM1,1	378.32061	150.74675	2.51	0.0121	1	Ike	0

The transfer function for Katrina becomes this formula:

$$\frac{\omega_0 - \omega_1 B^2}{1 - \delta B} X_{t-1} .$$

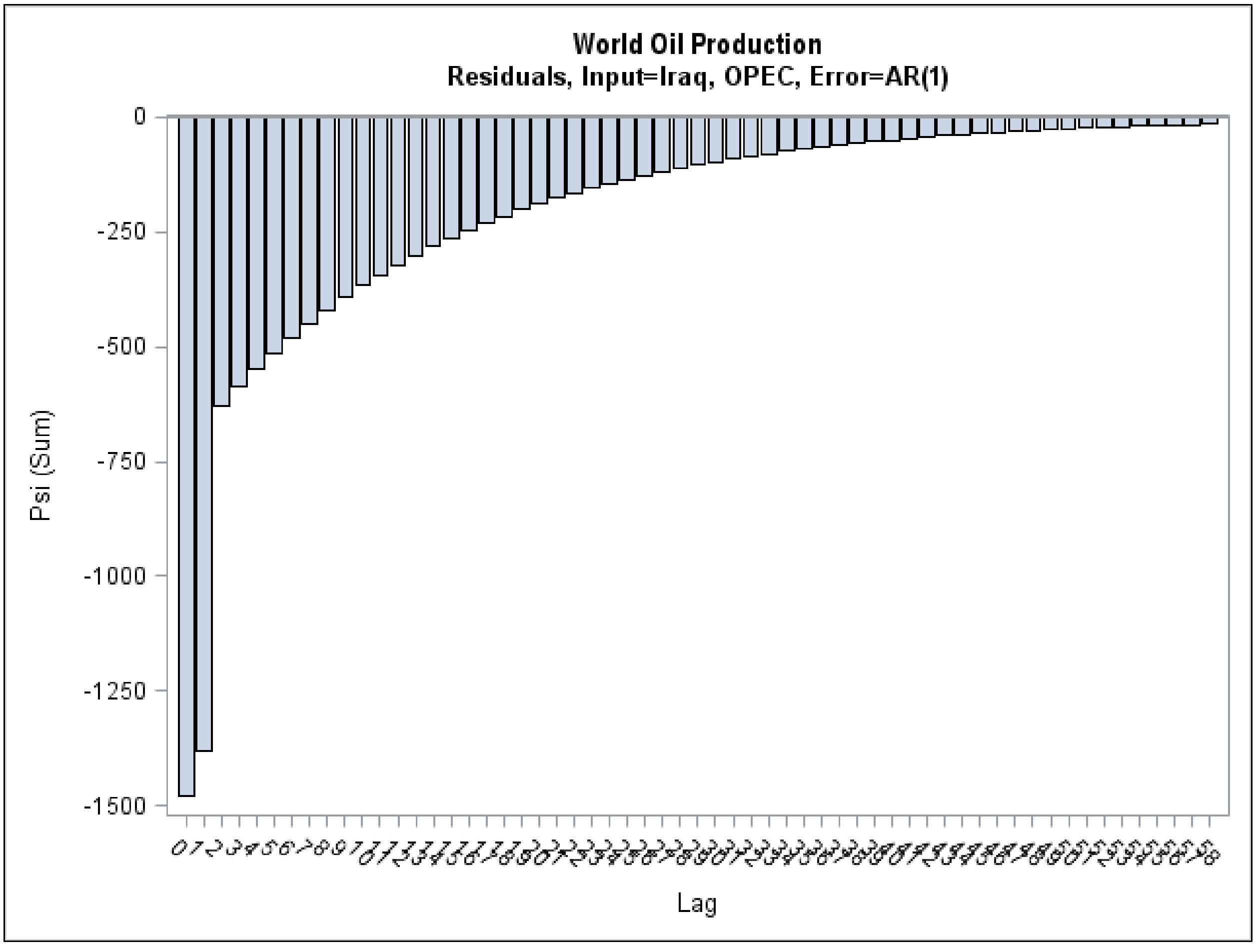
The estimates given in the table create the estimate transfer function:

$$\frac{-1477 + 664.74B^2}{1 - 0.935B} X_{t-1} .$$

All subsequent parameters after the first parameter have a negative sign in front of them, so you must interpret the estimate in the table with the implied negative sign. Thus, the shift 1 lag 2 Katrina effect is a positive increase of 664.7 thousand barrels. The %PlotPsi macro can be used to visualize the transfer function.

```
%PlotPsi(-1477 0 664.743,-0.93535,NLAGS=99,OutDS=work.temp);
```

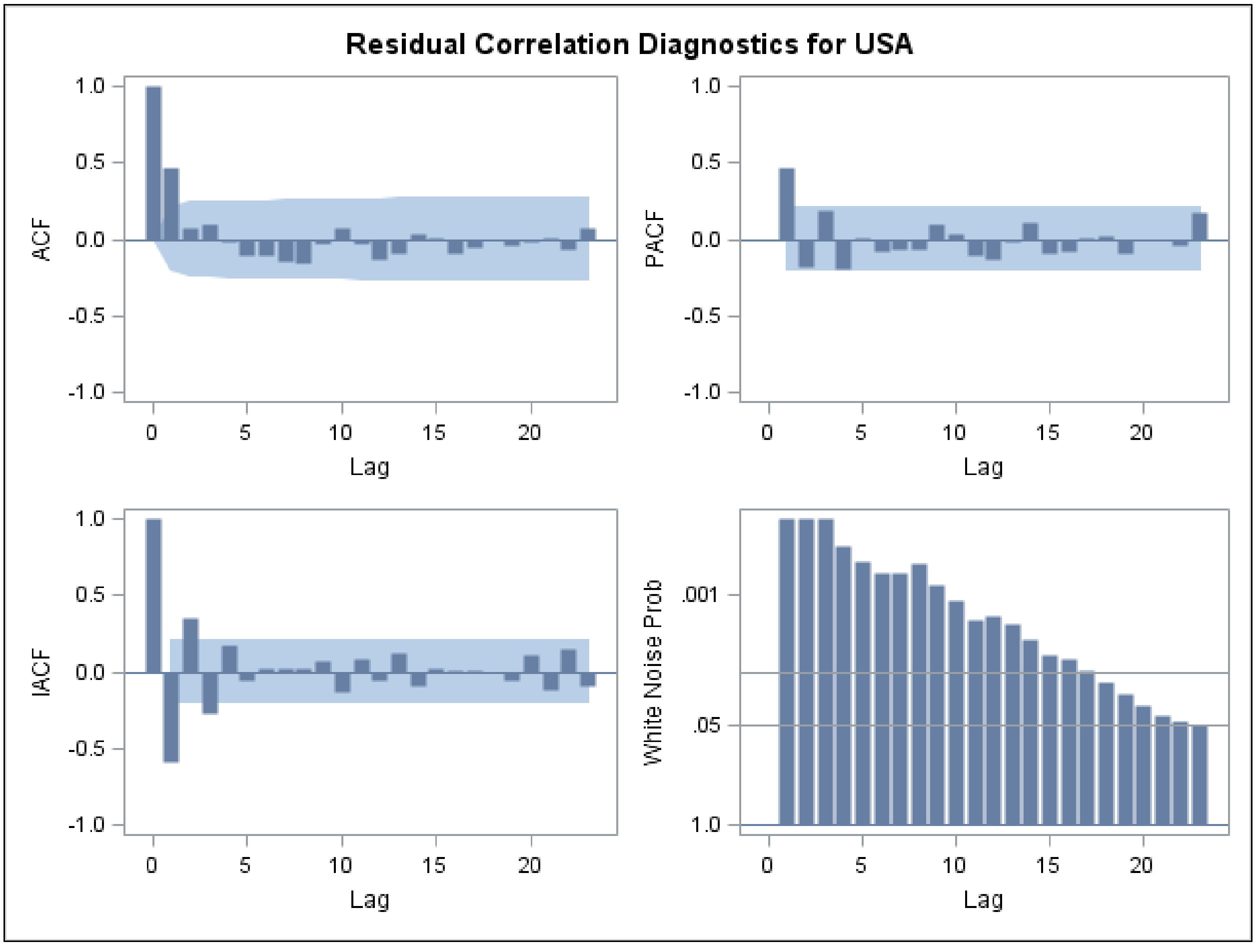
The plotted transfer function weights appear below:



The lag 2 coefficient cause a more rapid return to the production levels before Katrina and Rita hit the Gulf Coast.

The above plot is actually premature, but was intended to help you understand the nature of the transfer function. The estimates, and especially the standard errors of the estimates, cannot be trusted because of autocorrelation.

The residual autocorrelation panel is shown here:



The diagnostics suggest several models. There is enough ambiguity that you might want to use one of the automatic order determining criteria. The demonstration assumes the necessary analysis occurred and arrives at the model specified by the following code.

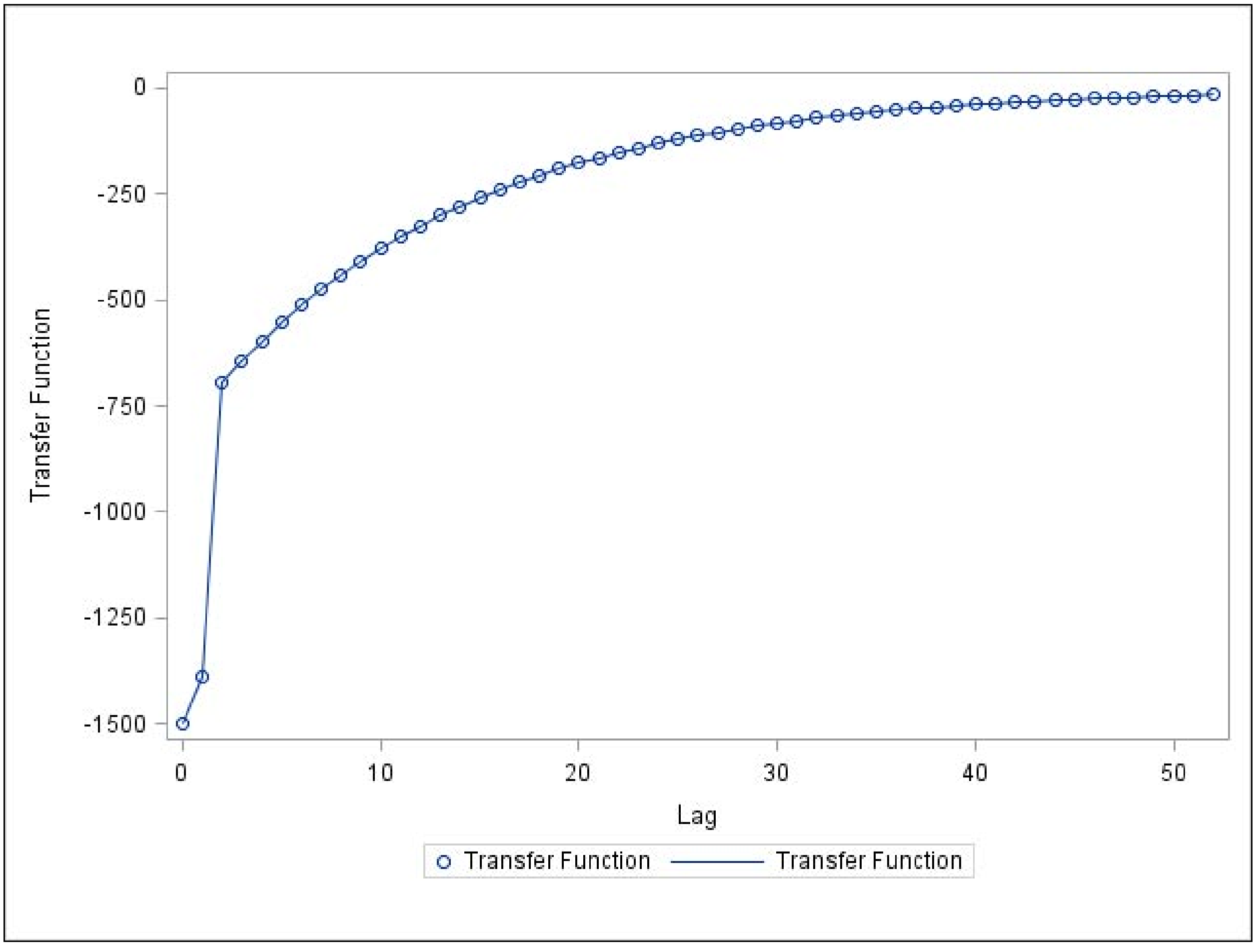
```
proc arima data=work.WorldOil;
   identify var=USA
            cross=(Iraq OPEC Katrina Ike)
            noprint;
   estimate p=1 q=(2)
            input=(Iraq OPEC 1$(2)/(1)Katrina (1)Ike)
            method=ml plot;
quit;
```

The table of parameter estimates follows:

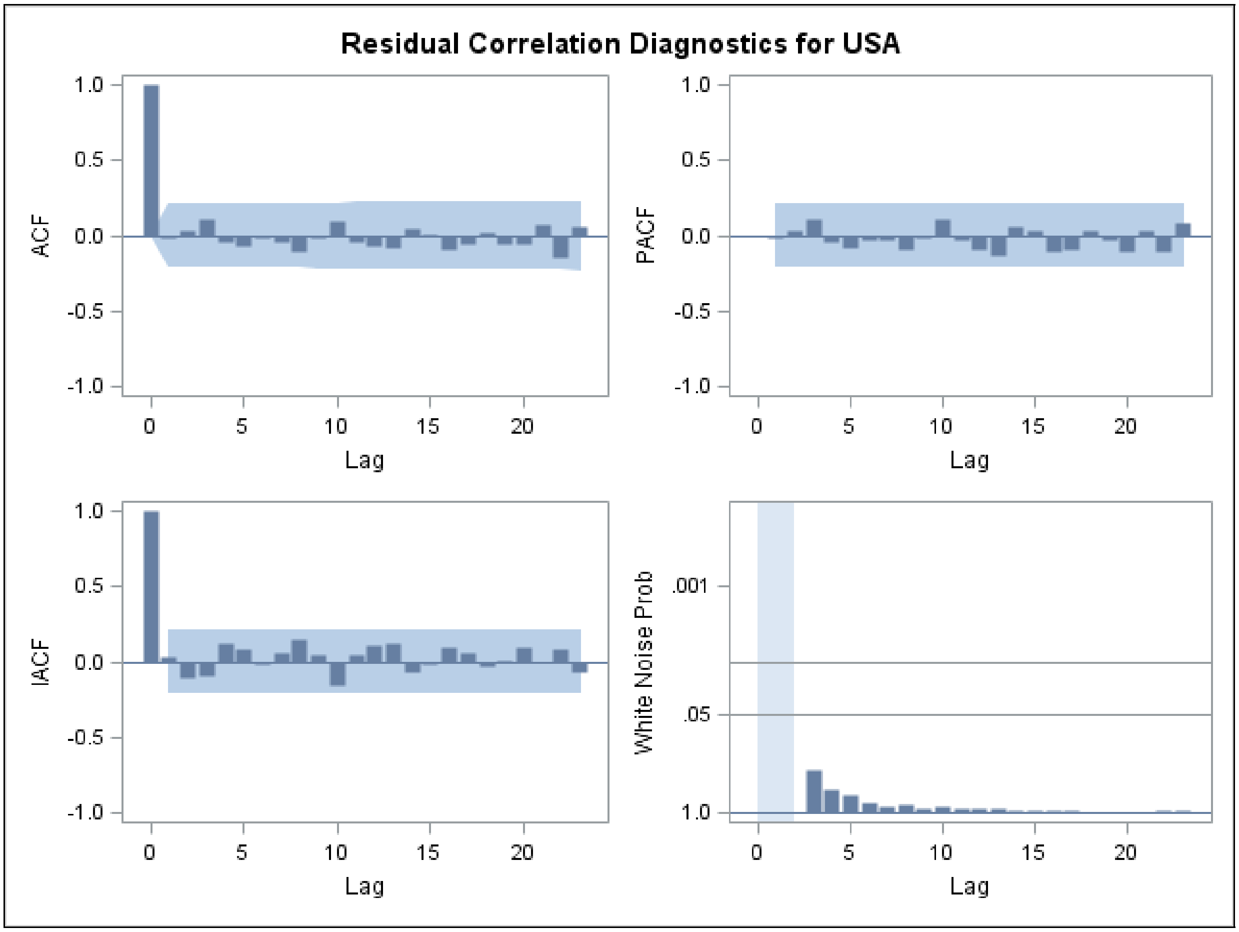
Maximum Likelihood Estimation							
Parameter	**Estimate**	**Standard Error**	**t Value**	**Approx Pr > |t|**	**Lag**	**Variable**	**Shift**
MU	10415.4	389.95412	26.71	<.0001	0	USA	0
MA1,1	0.46368	0.13750	3.37	0.0007	2	USA	0
AR1,1	0.69144	0.10302	6.71	<.0001	1	USA	0
NUM1	0.09330	0.03585	2.60	0.0093	0	Iraq	0
NUM2	-0.05344	0.01224	-4.37	<.0001	0	OPEC	0
NUM3	-1498.2	125.23678	-11.96	<.0001	0	Katrina	1
NUM1,1	-591.61883	143.78031	-4.11	<.0001	2	Katrina	1
DEN1,1	0.92677	0.01858	49.87	<.0001	1	Katrina	1
NUM4	-1384.9	133.45351	-10.38	<.0001	0	Ike	0
NUM1,1	392.00418	157.31596	2.49	0.0127	1	Ike	0

The transfer functions changed slightly. You can run %PlotPsi for this model.

The following plot is produced with PROC SGPLOT applied to the output from %PlotPsi:



The residual autocorrelation plots appear below:



The model cannot be disqualified.

Details

After the pure delay of one month (that is, in September), Katrina produced a drop of 1498 in oil production. The next month (October) production was approximately 0.92677(1498) units below normal, and the next month (November) it was about $-(0.92677)^2(1498)+592=-695$ above normal (which is 695 below normal). Then it was 0.92677(695) below, $(0.92677)^2(695)$ below, and so on.

The coefficient 0.0933 for Iraq and the coefficient -0.05344 for OPEC have opposite signs and are reasonably close in magnitude, given their standard errors. Perhaps there is a single coefficient between .09 and .05 that is closer to 0.05 given its smaller standard error, which applies to the difference (OPEC-IRAQ). A follow-up analysis might consider replacing the two variables by the difference (OPEC-Iraq).

An immediate drop of 1385 is associated with the month that Ike occurred. The next month production was down by an estimated 392 because the estimate of the transfer function is (-1385-392B). Events with major impacts such as this are often called *interventions*. You can better understand why the term *transfer function* is used by considering the hurricane effects. The name suggests that the effects of the hurricanes were ***transferred*** over time to oil production. You have no way to know when a future hurricane similar to Katrina or Ike will occur, but ignoring their effects in the modeling stage gives biased parameter estimates (a too low level, for example) and causes the forecasts to be too low, at least for this example.

The dummy variables for Ike and Katrina are examples of pulse dummy variables. The cross-correlation function between a response Y and an input X has a pattern proportional to $\sum (Y_t - \overline{Y}) X_{t-j}$ where Y is the response, X is the input, and j can be any integer for which the sum can be computed. If X is a pulse indicator having the value 1 at time t_0 and 0 elsewhere, the sum becomes $\sum (Y_t - \overline{Y}) X_{t-j} = Y_{t_0+j} - \overline{Y}$. The cross-correlation function at lags j = 0, 1, 2, 3, … thus shows the response of Y to the intervention over time as does the graph of Y near the time of the intervention. This is the key to identifying the nature of the transfer function. When an exponential decay was seen for Katrina, this was reminiscent of the ACF decay pattern of an AR(1), so one denominator lag was specified. The unusual cross-correlation drop in magnitude at lag 2 was reminiscent of an MA lag 2 effect. When spikes at lag 0 and 1 were seen for Ike, this was reminiscent of the spikes at lag 0 and 1 in the ACF with a moving average of order 1. In the case of a pulse intervention, the transfer function model can be identified this way in general. Simply interpret the cross-correlation pattern using the ACF patterns that you already know. Make a diagnosis of the AR and MA lags as if you were looking at an ACF pattern. Then put your autoregressive lags in the denominator and your MA lags in the numerator of the transfer function. For example, if you saw a pure delay 1 followed by a damped sinusoidal pattern (reminiscent of an AR(2)), you would specify the following:

```
INPUT=(1$(1 2)X)
```

If the input is not a pulse dummy variable, a form of preprocessing known as *prewhitening* is required in order to produce interpretable cross-correlations similar to those for pulse interventions. (This additional complication will be addressed later.)

5.01 Multiple Choice Poll

An outlier can be related to which of the following?

a. A miscoded data value
b. A source of variation left out of the model
c. A suspicious or unusual data value that can bias forecasts
d. All of the above

56

5.2 Event Models

Objectives

- Discuss events in time series analysis.
- List the three basic variables for modeling time series data with event effects.
- Show how models are specified using PROC ARIMA.
- Fit models to time series data that might be impacted by events.
- Make inferences and forecasts using models with event terms.

60

Events

- An *event* is anything that changes the underlying process that generates time series data.
- The analysis of events includes two activities:
  - Exploration to identify the functional form of the effect of the event
  - Inference to determine whether the event has a statistically significant effect
- Other names for the analysis of events are the following:
  - ***Intervention analysis***
  - Interrupted time series analysis

61

Intervention Analysis

- Special case of *transfer function modeling* in which the predictor variable is a deterministic categorical variable
- Derived from the concept of a public policy *intervention* having an effect on a socio-economic variable
  - Example: Raising the minimum wage increases the unemployment rate.
  - Example: Implementing a severe drunk-driving law reduces automobile fatalities.

62

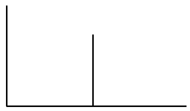
Event/Intervention Analysis Practices

- In retail sales, the term *event* is often employed and includes the following:
  - Promotional events—Discounts, sales, featured displays, and so forth
  - Advertising events—Broadcast, Internet, and print media advertising campaigns, sponsored events, celebrity spokespersons, and so forth
- In economics and the social sciences, the term *intervention* is often employed and includes these:
  - Catastrophic events
  - Events related to a key player (CEO, spokesperson)—imprisonment, scandal, illness or injury, death
  - Public policy changes

63

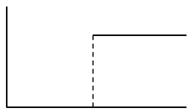
Primary Event Variables

Point/Pulse



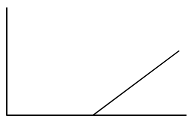
$$J_t = \begin{cases} 1 & \text{for } t = t_{\text{event}} \\ 0 & \text{for } t \neq t_{\text{event}} \end{cases}$$

Step



$$I_t = \begin{cases} 1 & \text{for } t \geq t_{\text{event}} \\ 0 & \text{for } t < t_{\text{event}} \end{cases}$$

Ramp



t_{event}

$$R_t = \begin{cases} t - t_{\text{event}} & \text{for } t \geq t_{\text{event}} \\ 0 & \text{for } t < t_{\text{event}} \end{cases}$$

64

Example Transfer Function

$$Y_t = \frac{8X_t}{(1-0.5B)} \rightarrow \text{which simplifies to} \rightarrow Y_t = 0.5Y_{t-1} + 8X_t$$

Pulse

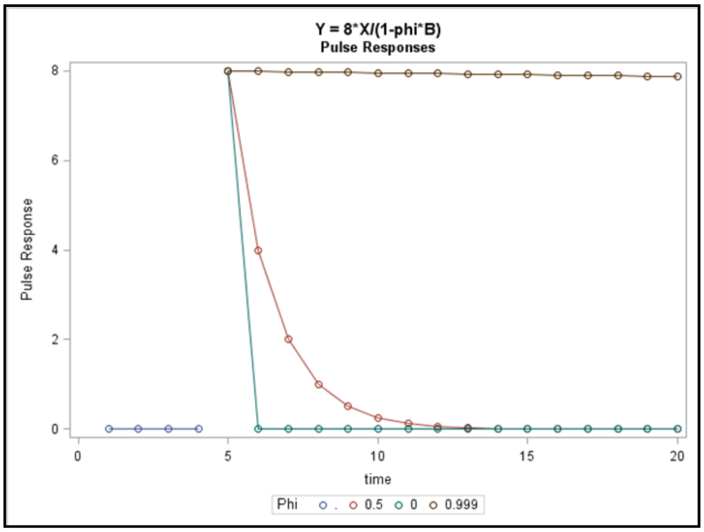
X	0	0	1	0	0	0	0	….
Y	0	0	8	4	2	1	0.5	→ 0

Step

X	0	0	1	1	1	1	1	….
Y	0	0	8	12	14	15	15.5	→ 16

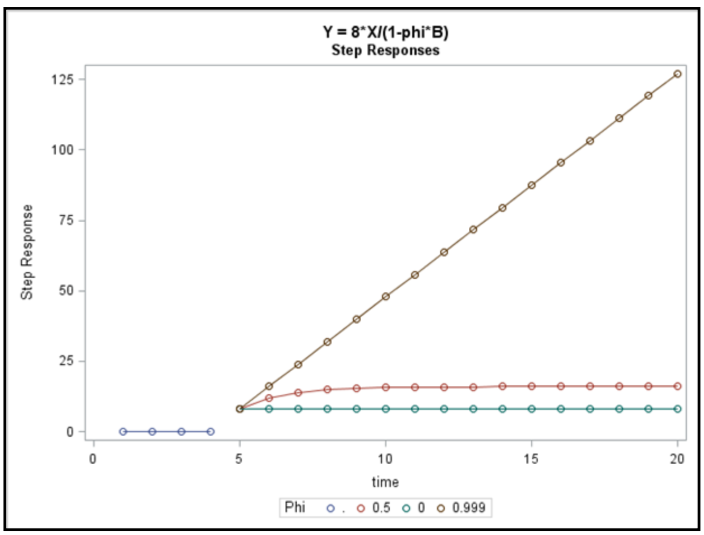
65

Pulse Responses



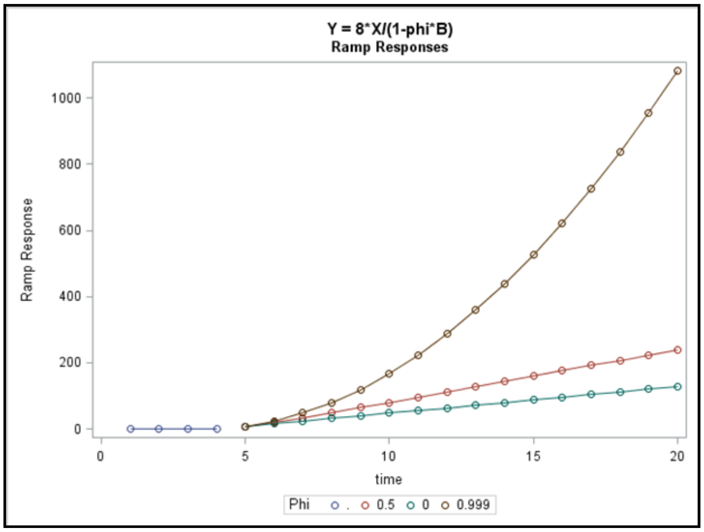
66

Step Responses



67

Ramp Responses



68

Each Phi=0 case is eight times the corresponding intervention variable (X) and as such shows the shape of that intervention input. The Phi=0.5 cases show the effect of the denominator term when Phi has a value not too close to 0 or 1. The phi=.999 cases deserve discussion. Notice that $Y_t = 0.999Y_{t\text{-}1} + 8X_t$ is ***almost*** $Y_t = Y_{t\text{-}1} + 8X_t$. Looking at this extreme Phi=1 case gives insight. When X is a pulse, Y jumps to 8 and each subsequent Y is almost 8 + 0. A phi coefficient near 1 for a pulse input suggests that a step function might be a better choice. Similarly, a step function with a denominator coefficient near 1 acts similar to a ramp, and a ramp function with a coefficient near 1 acts similar to a quadratic response. Mathematically, when the X sequence is 0 0 0 1 2 3 4 … and $Y_t = Y_{t\text{-}1}+X_t$, the Y sequence becomes 0 0 0 1 3 6 10 … , that is, $Y = .5(X+X^2)$, a quadratic response. Put another way, the first differences of the quadratic sequence form a ramp sequence. The first differences of a ramp form a step and the first differences of a step form a pulse sequence. This is seen to be related to the ideas of unit roots and differencing as discussed earlier.

To identify the appropriate transfer function form, simply look at the series near the intervention point and compare it to the graphs above. Remember the previous discussion of the additional features associated with numerator lags and additional denominator lags.

The General Rational Polynomial Transfer Function

$$Y_t = \frac{\omega_i(B)}{\delta_i(B)} X_{i,t} + \cdots$$

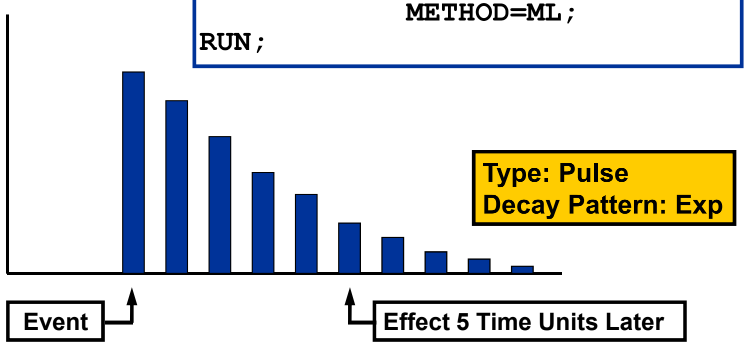
$$\frac{\omega_i(B)}{\delta_i(B)} = \eta_{i0} - \eta_{i1}B - \eta_{i2}B^2 - \cdots - \eta_{ik}B^k - \cdots$$

$$Y_t = \eta_{i0}X_{i,t} - \eta_{i1}X_{i,t-1} - \eta_{i2}X_{i,t-2} - \cdots - \eta_{ik}X_{i,t-k} + \cdots$$

69

Abrupt, Temporary Effect

```
PROC ARIMA DATA=<data-set>;
   IDENTIFY VAR=Y CROSS=(PULSE)
            NLAGS=16;
   ESTIMATE INPUT=(/(1)PULSE)
            METHOD=ML;
RUN;
```



70

continued...

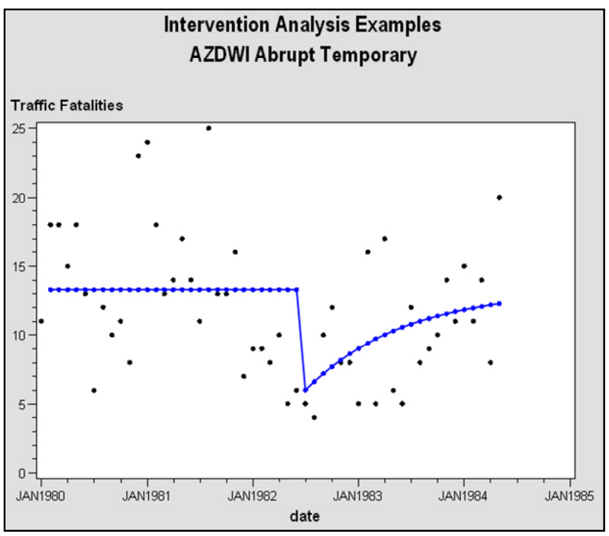
Regression Model for an Abrupt, Temporary Effect

$$Y_t = \beta_0 + \frac{\omega(B)}{\delta(B)} J_t + Z_t$$ Dynamic Regression Model

$$\phi(B) Z_t = \theta(B) \varepsilon_t$$ ARMA(p,q) Error Term

71

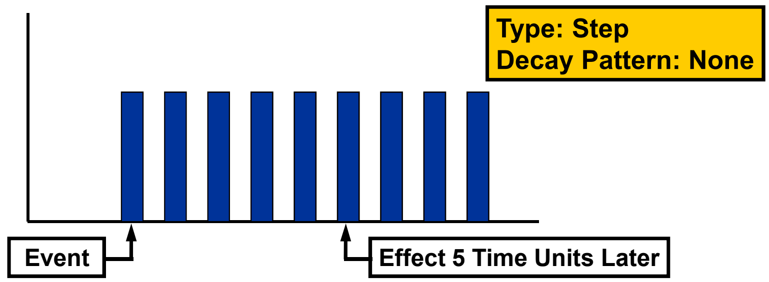
Abrupt, Temporary Effect



72

Abrupt, Permanent Effect

```
PROC ARIMA DATA=<data-set>;
   IDENTIFY VAR=Y CROSS=(STEP)
            NLAGS=16;
   ESTIMATE INPUT=(STEP)
            METHOD=ML;
RUN;
```



73

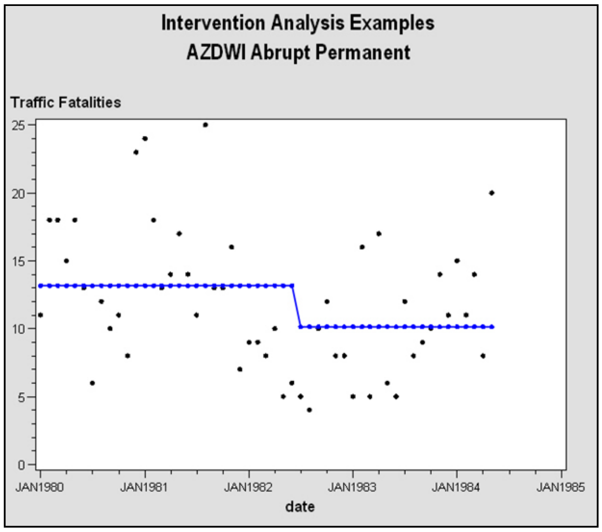
Regression Model for an Abrupt, Permanent Effect

$$Y_t = \beta_0 + \omega_0 I_t + Z_t$$ Ordinary Regression Model

$$\phi(B)Z_t = \theta(B)\varepsilon_t$$ ARMA(*p*,*q*) Error Term

74

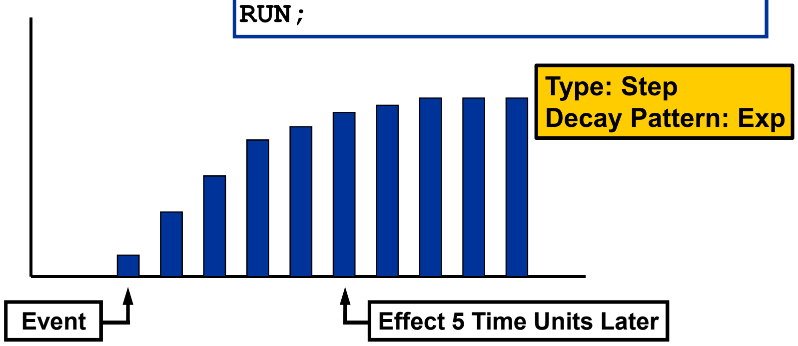
Abrupt, Permanent Effect



75

Gradual, Permanent Effect

```
PROC ARIMA DATA=<data-set>;
   IDENTIFY VAR=Y CROSS=(STEP)
            NLAGS=16;
   ESTIMATE INPUT=(/(1)STEP)
            METHOD=ML;
RUN;
```



76

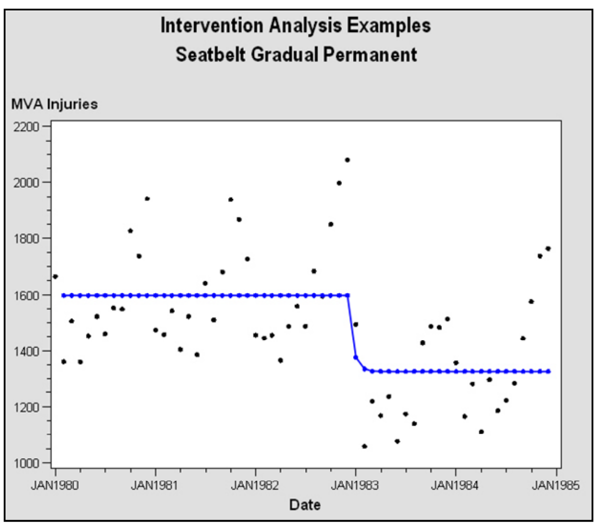
Regression Model for a Gradual, Permanent Effect

$$Y_t = \beta_0 + \frac{\omega(B)}{\delta(B)} I_t + Z_t$$ Dynamic Regression Model

$$\phi(B)Z_t = \theta(B)\varepsilon_t$$ ARMA(*p*,*q*) Error Term

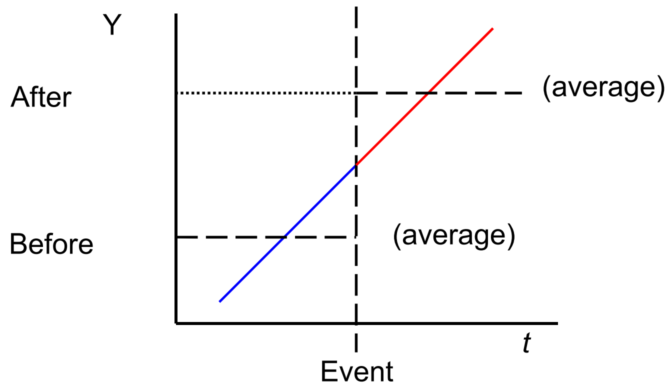
77

Gradual, Permanent Effect



78

Changes in Level and Trend for Events

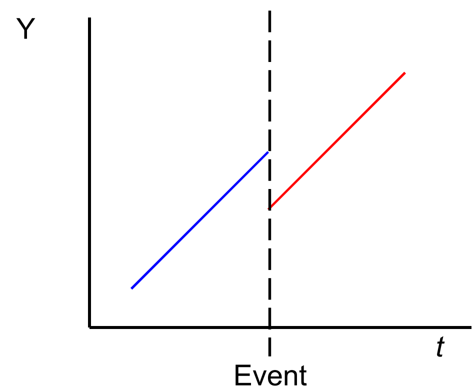


False Inference: The event causes the result to increase because AVERAGE(after) > AVERAGE(before).

Valid Inference: The event has no effect on the results.

79

Changes in Level and Trend for Events

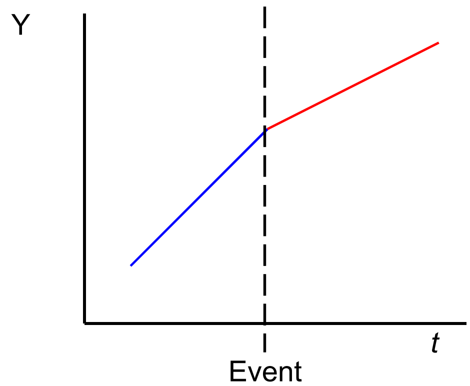


Valid Inference: The event causes a change in level.

80

continued...

Changes in Level and Trend for Events

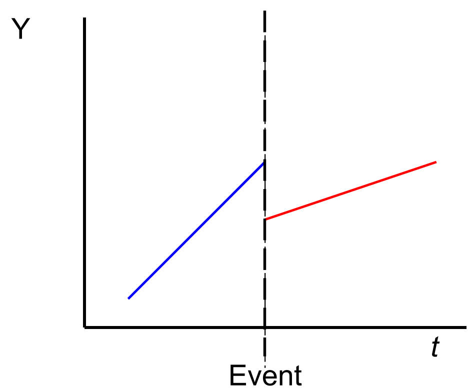


Valid Inference: The event causes a change in the slope of the trend line.

81

continued...

Changes in Level and Trend for Events

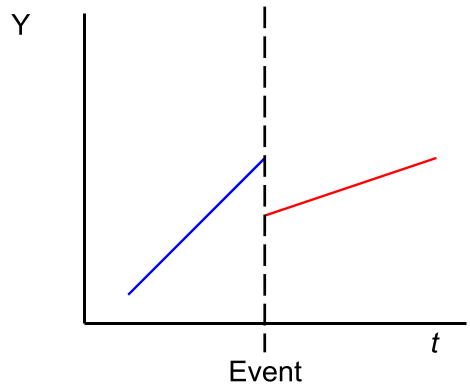


Valid Inference: The event causes a change in the level and the slope.

82

continued...

Changes in Level and Trend for Events



Step Function: Changes in level.
Ramp Function: Changes in slope.

83

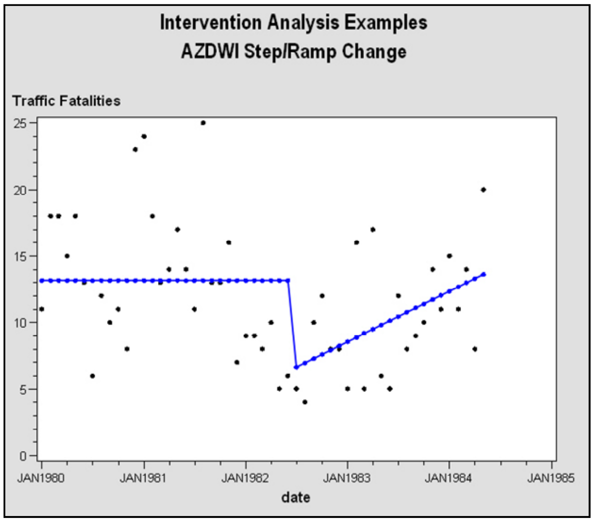
Changes in Level and Trend for Events

```
proc arima data=SASDataSet;
   identify var=TargetVariable
            crosscor=(Step Ramp)
            nlags=24 noprint;
   estimate input=(Step Ramp)
            method=ml;
quit;
```

84

continued...

Changes in Level and Trend for Events



85

continued...

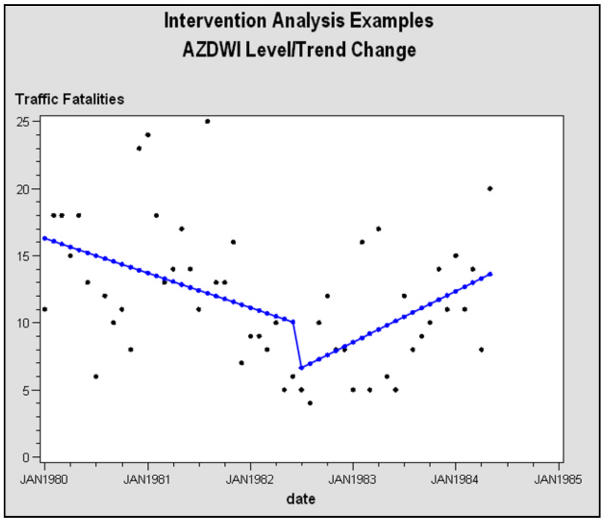
Changes in Level and Trend for Events

```
proc arima data=SASDataSet;
   identify var=TargetVariable
            crosscor=(TimeIndex Step Ramp)
            nlags=24 noprint;
   estimate input=(TimeIndex Step Ramp)
            method=ml;
quit;
```

86

continued...

Changes in Level and Trend for Events



87

Popular Intervention Effects

```
/*---- Abrupt Temporary: Pulse         ----*/
estimate input=(    Pulse) method=ml;
/*---- Abrupt Temporary:               ----*/
/*---- Pulse with exponential decay    ----*/
estimate input=(/(1)Pulse) method=ml;
/*---- Abrupt Temporary:               ----*/
/*---- Pulse with oscillating decay    ----*/
estimate input=(/(1 2)Pulse) method=ml;
/*---- Abrupt Permanent: Step          ----*/
estimate input=(    Step) method=ml;
```

88

continued...

Popular Intervention Effects

```
/*--- Gradual Permanent:                  ----*/
/*--- Gradual Step asymptoting to new level--*/
estimate input=(/(1)Step) method=ml;
/*--- Gradual Permanent:                  ----*/
/*--- Gradual Step oscilating to new level --*/
estimate input=(/(1 2)Step) method=ml;
/*--- Change in trend                     ----*/
estimate input=(TimeIndex Ramp) method=ml;
/*--- Change in level and trend           ----*/
estimate input=(TimeIndex Step Ramp) method=ml;
/*--- Change in level and trend           ----*/
estimate input=(Step Ramp) method=ml;
```

89

Combined Effect of Two Events

01JAN1982: Media Coverage Intensifies

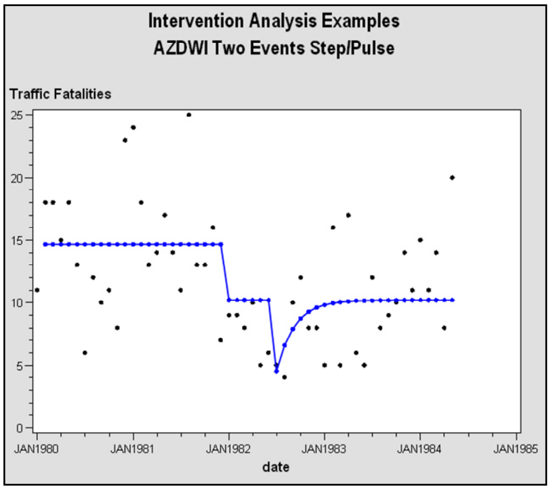
```
estimate input=(
                MediaStep

                /(1)Pulse
               ) method=ml;
```

01JUL1982: DWI Law Takes Effect

90

continued...

Combined Effect of Two Events



RMSE=4.341

91

Effect of Two Events: Alternate Model

01JAN1982: Step Function

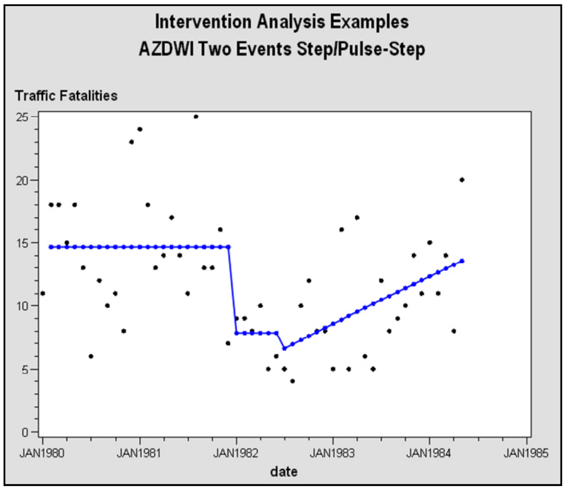
```
estimate input=(
                MediaStep

                /(1)Pulse Step
               ) method=ml;
```

01JUL1982: Point-Exp and Step Functions

92

continued...

Effect of Two Events: Alternate Model



RMSE=4.159

93

Two Approaches to Event Analysis

1. Identify the underlying model structure first, and then identify the nature of the event transfer function.
   a. Select the pre-intervention series.
   b. Apply the usual modeling strategy.
   c. Forecast the entire series with the pre-intervention model.
   d. Examine the residuals to identify the form of the transfer function.
2. Identify the nature of the event transfer function first, and then identify the underlying model structure.
   a. Propose event transfer functions based on domain knowledge, plots of the series, or trial and error.
   b. Apply the chosen transfer function, and identify the form of the underlying model from the residuals.

94

Intervention Analysis of the Airline Data

This demonstration illustrates how to use PROC ARIMA to identify the underlying model structure and identify candidate intervention transfer functions to evaluate the effect of the September 11, 2001, terrorist attacks in the United States on scheduled flights of domestic U.S. carriers.

The SAS code for this demonstration can be found in **Demo5_03Airline.sas**.

In a previous demonstration, a model was fit to the pre-intervention airline data for years 1990 through 2000. The selected model is an ARIMA(1,1,1)(0,1,1)$_{12}$ model. This model will be used as the underlying model for the airline time series.

Visual inspection of the plot suggests possible intervention components. The following code adds appropriate event variables to the original data:

```
data work.usairlines;
   set sasuser.usairlines end=lastobs;
   attrib Pulse     length=3 label='Pulse Function'
          Step      length=3 label='Step Function'
          Ramp2001  length=3 label='Ramp Function Sep2001'
          Ramp2005  length=3 label='Ramp Function Jan2005'
          Time      length=8 label='Time Index';
   retain Pulse Step Ramp2001 Ramp2005 Time 0;
   Time+1;
   if ('01SEP2001'd<=Date<='30SEP2001'd) then do;
      Ramp2001+1;
      Step=1;
      Pulse=1;
   end;
   else if (Date > '30SEP2001'd) then do;
      Ramp2001+1;
      Pulse=0;
   end;
   if ('01JAN2005'd<=Date<='31Jan2005'd) then Ramp2005+1;
   else if (Date > '31JAN2005'd) then Ramp2005+1;
   output;
   if (lastobs) then do;
      Passengers=.;
      do Future=1 to 24;
         Date=intnx('month',Date,1);
         Year=year(Date);
         Month=month(Date);
         Ramp2001+1;
         Ramp2005+1;
         output;
      end;
   end;
   drop Future;
run;
```

Two intervention types are investigated as suggested by the following code. (Other types were tried, but only two are included to illustrate the process.)

```
proc arima data=work.usairlines;
   identify var=Passengers(1 12)
            crosscorr=(Pulse(1 12) Ramp2005(1 12))
            noprint;
   /*----  Abrupt Temporary  ----*/
   estimate p=1 q=(1)(12)
            input=(/(1)Pulse)
            method=ml
            outest=work.AT_Est
            noprint;
   forecast id=Date interval=month lead=0
            out=work.AbruptTemp noprint;
   /*----  Abrupt Temporary w/Ramp  ----*/
   estimate p=1 q=(1)(12)
            input=(/(1)Pulse Ramp2005)
            method=ml
            outest=work.ATR_Est

            noprint;
   forecast id=Date interval=month lead=0
            out=work.AbruptTempRamp noprint;
quit;
```

General diagnostics are excluded from this analysis. The models passed inspection, except the Ramp2005 estimate is not statistically significant.

The following code generates goodness-of-fit tables, plots of the intervention effect, and plots of the in-sample forecasts:

```
%GOFstats(ModelName=%str(Abrupt Temporary),
          DSName=work.AbruptTemp,
          OutDS=work.AbruptTemp_m,
          ActualVar=Passengers,
          ForecastVar=FORECAST,
          NumParms=6);
%GOFstats(ModelName=%str(Abrupt Temp w/Ramp),
          DSName=work.AbruptTempRamp,
          OutDS=work.AbruptTempRamp_m,
          ActualVar=Passengers,
          ForecastVar=FORECAST,
          NumParms=7);

data work.all;
   set work.AbruptTemp_m
       work.AbruptTempRamp_m;
run;

proc print data=work.all noobs;
run;
```

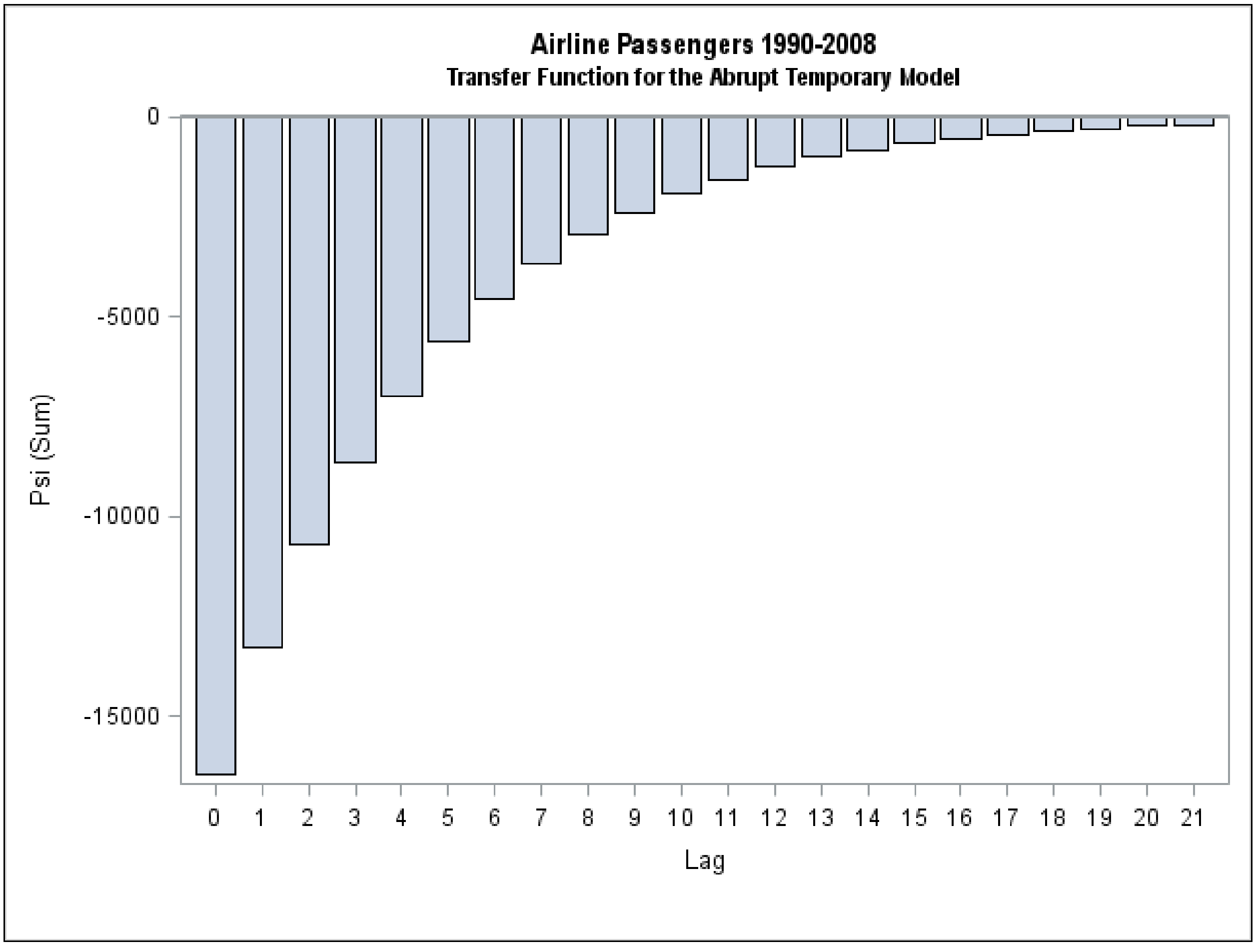
The generated table follows:

Model	MAPE	RMSE	NumParm	AIC_SSE	SBC_SSE
Abrupt Temporary	2.04446	1351.32	6	2961.53	2981.47
Abrupt Temp w/Ramp	2.05630	1351.87	7	2962.67	2985.93

The model without the **Ramp2005** event variable is slightly superior. Furthermore, the lack of significance of the **Ramp2005** variable leads to the selection of the simple abrupt temporary intervention effect. The following code uses %PlotPsi to get a visual image of the intervention effect:

```
data _null_;
   set work.AT_Est;
   if (_TYPE_="EST") then do;
      call symput("NumFactor",put(I1_1,20.10));
      call symput("DenFactor",put(-I1_2,20.11));
   end;
run;
%put &NumFactor &DenFactor;

title2 font=&COURSEFONT 'Transfer Function for the Abrupt Temporary Model';
%PlotPsi(&NumFactor,&DenFactor,NLAGS=37);
```

The effects plot follows:



The resulting chart reveals that the effect of the events of September 11, 2001, was an abrupt, temporary change in passenger counts that returned to original levels approximately 23 months after September 2001, which means that July 2003, experienced passenger counts that would be expected even if the events of September 11, 2001, did not occur. Of course, the recovery of passenger counts does not imply recovery from the economic losses.

The following code regenerates all of the statistics and plots for the final model:

```
proc arima data=work.usairlines plots=all;
   identify var=Passengers(1 12)
            crosscorr=(Pulse(1 12) Ramp2005(1 12))
            noprint;
   estimate p=1 q=(1)(12)
            input=(/(1)Pulse)
            method=ml;
   forecast id=Date interval=month lead=24;
quit;
```

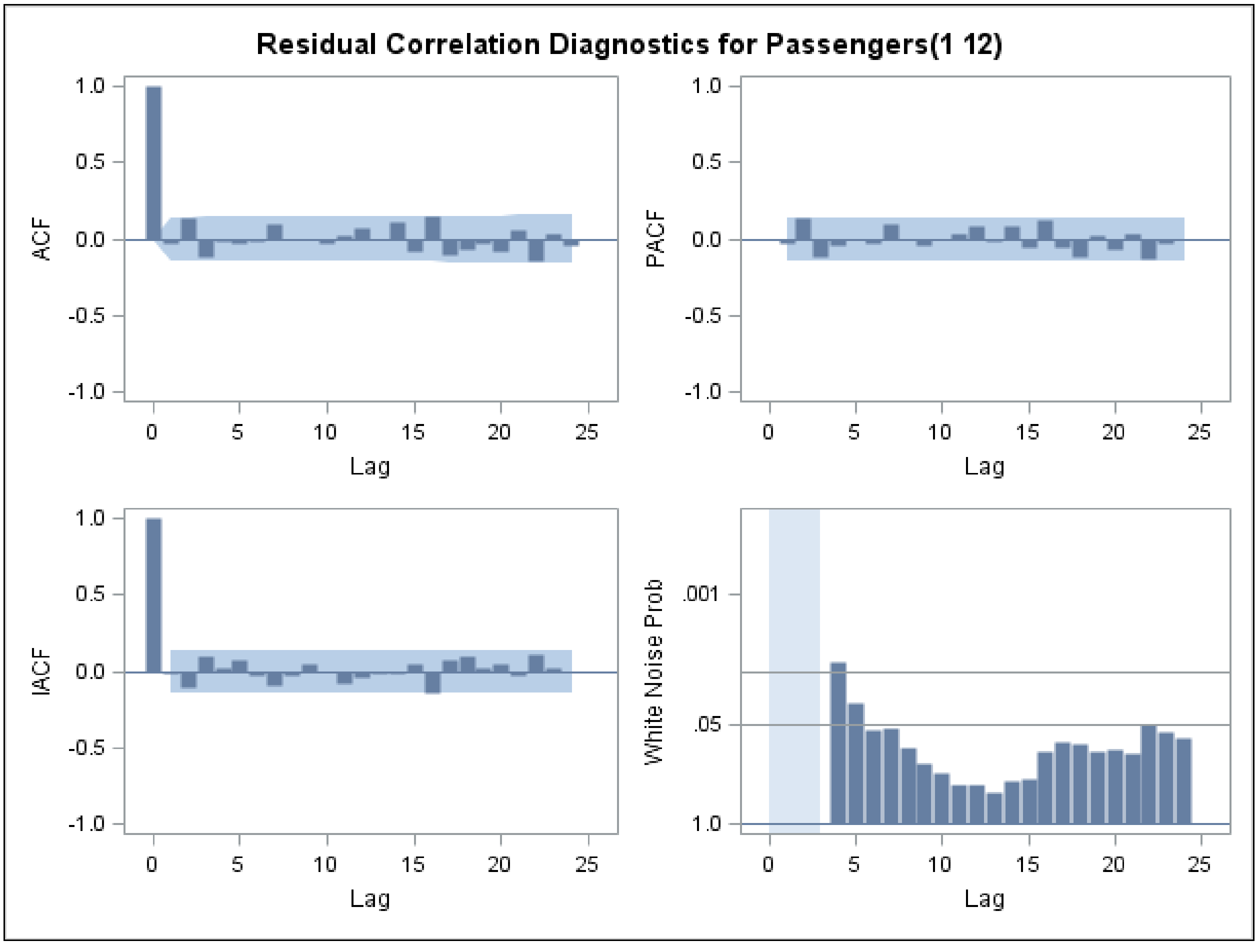
The results for the model appear below:

Maximum Likelihood Estimation							
Parameter	Estimate	Standard Error	t Value	Approx Pr > |t|	Lag	Variable	Shift
MU	6.03747	19.15102	0.32	0.7526	0	Passengers	0
MA1,1	0.66230	0.11395	5.81	<.0001	1	Passengers	0
MA2,1	0.59660	0.06121	9.75	<.0001	12	Passengers	0
AR1,1	0.27641	0.14468	1.91	0.0561	1	Passengers	0
NUM1	-16465.4	1107.3	-14.87	<.0001	0	Pulse	0
DEN1,1	0.80650	0.03452	23.37	<.0001	1	Pulse	0

Constant Estimate	4.368655
Variance Estimate	1768236
Std Error Estimate	1329.751
AIC	3542.242
SBC	3562.18
Number of Residuals	205

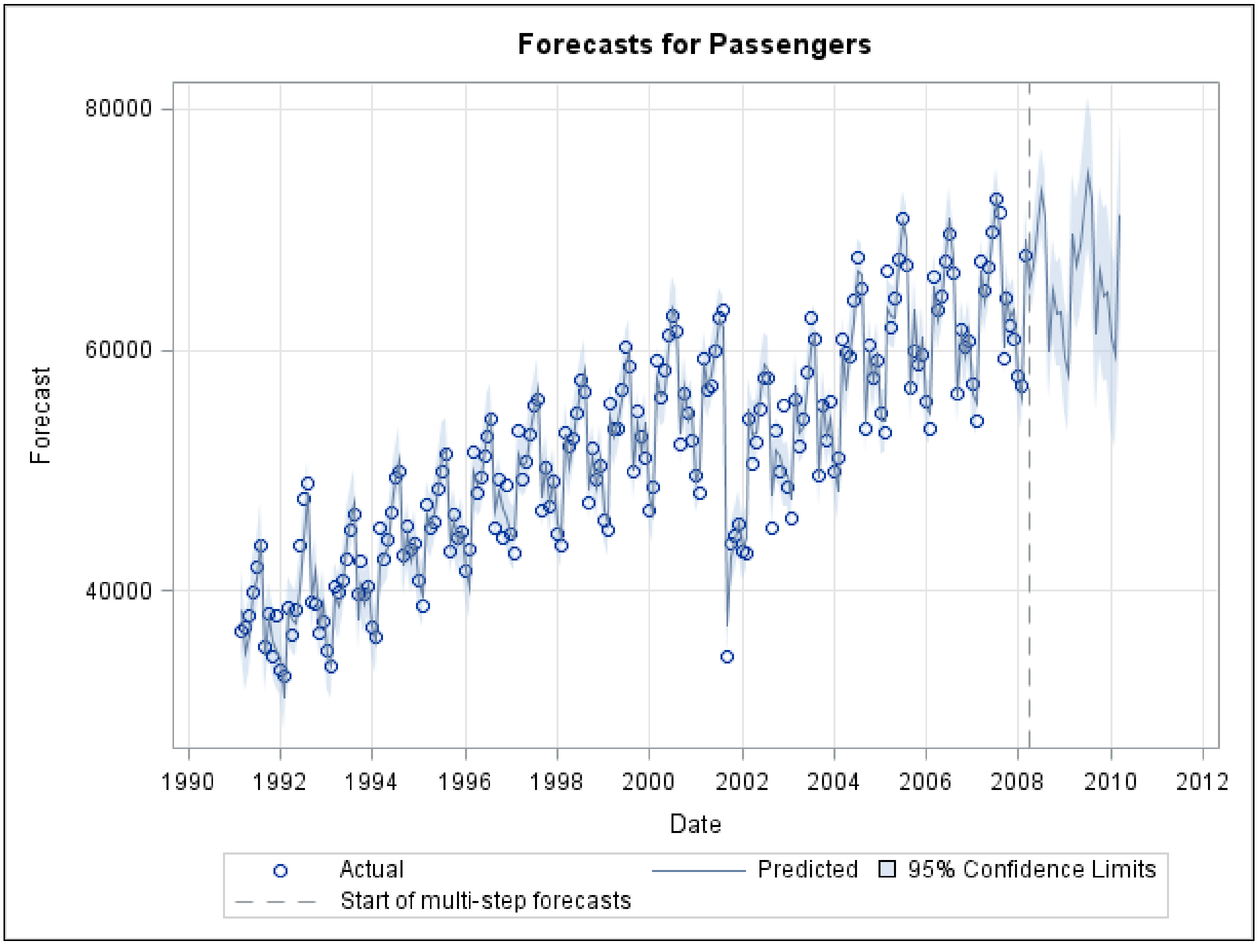
The model suggests that the event caused an immediate loss of 16.5 million passengers per month, with a recovery reducing this loss by about 80% each month. For example, the loss for October 2001, is estimated to be 16,514,300 times 0.8065, which is approximately 13.3 million passengers. The *p*-value for the AR(1) coefficient is larger than 0.05, but the AR(1) term is kept in the model to ensure that the residuals pass a white noise test.

The residual diagnostic plots follow:



White noise *p*-values are low for cumulative lags 4 and 5, but overall the residuals appear to be similar to white noise.

The overall forecast plot follows:



With additional data, the model with a ramp might prove to be superior.

5.02 Multiple Choice Poll

Which of the following represents the correct specification of an abrupt temporary intervention effect in PROC ARIMA?

a. INPUT=(/(1)PULSE)
b. INPUT=(/(1)STEP)
c. INPUT=(/(1)RAMP)
d. INPUT=(STEP RAMP)
e. INPUT=(PULSE RAMP)

97

5.3 Time Series Regression Models

Objectives

- Describe the general dynamic regression model.
- Give common simple examples of dynamic regression models.
- Use PROC ARIMA to diagnose and fit time series regression models.

101

Box-Jenkins Time Series Regression Model

$$(1-B)^d(1-B^S)^D g(Y_t)$$
$$= \mu + \sum_{i=1}^{k} B^{m_i}(1-B)^{d_i}(1-B^S)^{D_i}\frac{\omega_i(B)\omega_{S,i}(B^S)}{\delta_i(B)\delta_{S,i}(B^S)} f_i(X_{i,t}) + \frac{\theta(B)\Theta(B^S)}{\phi(B)\Phi(B^S)}\varepsilon_t$$

Theoretical ARIMAX Model

102 ...

Box-Jenkins Model Components

Trend

$$\boxed{(1-B)^{d}}(1-B^{S})^{D}g(Y_t)$$
$$= \mu + \sum_{i=1}^{k} B^{m_i}\boxed{(1-B)^{d_i}}(1-B^{S})^{D_i}\frac{\omega_i(B)\omega_{S,i}(B^{S})}{\delta_i(B)\delta_{S,i}(B^{S})}f_i(X_{i,t}) + \frac{\theta(B)\Theta(B^{S})}{\phi(B)\Phi(B^{S})}\varepsilon_t$$

Theoretical ARIMAX Model

103 ...

Box-Jenkins Model Components

Seasonal

$$(1-B)^{d}\boxed{(1-B^{S})^{D}}g(Y_t)$$
$$= \mu + \sum_{i=1}^{k} B^{m_i}(1-B)^{d_i}\boxed{(1-B^{S})^{D_i}}\frac{\omega_i(B)\boxed{\omega_{S,i}(B^{S})}}{\delta_i(B)\boxed{\delta_{S,i}(B^{S})}}f_i(X_{i,t}) + \frac{\theta(B)\boxed{\Theta(B^{S})}}{\phi(B)\boxed{\Phi(B^{S})}}\varepsilon_t$$

Theoretical ARIMAX Model

104 ...

Box-Jenkins Model Components

Stationary

$$(1-B)^{d}(1-B^{S})^{D}g(Y_t)$$
$$=\mu+\sum_{i=1}^{k}B^{m_i}(1-B)^{d_i}(1-B^{S})^{D_i}\frac{\omega_i(B)\omega_{S,i}(B^{S})}{\delta_i(B)\delta_{S,i}(B^{S})}f_i(X_{i,t})+\frac{\theta(B)\Theta(B^{S})}{\phi(B)\Phi(B^{S})}\varepsilon_t$$

Theoretical ARIMAX Model

105 ...

Box-Jenkins Model Components

Input Variables

$$(1-B)^{d}(1-B^{S})^{D}g(Y_t)$$
$$=\mu+\boxed{\sum_{i=1}^{k}B^{m_i}(1-B)^{d_i}(1-B^{S})^{D_i}\frac{\omega_i(B)\omega_{S,i}(B^{S})}{\delta_i(B)\delta_{S,i}(B^{S})}f_i(X_{i,t})}+\frac{\theta(B)\Theta(B^{S})}{\phi(B)\Phi(B^{S})}\varepsilon_t$$

Theoretical ARIMAX Model

106

The general theoretical model can be quite intimidating. Examining simple versions of the general model might help you understand how the model is formulated.

Common Transfer Functions

Ordinary Regression

$$\omega(B) = \omega_0, \delta(B) = 1$$

Model

$$Y_t = \theta_0 + \omega_0 X_t + Z_t$$
$$\phi(B) Z_t = \theta(B) \varepsilon_t$$

107

Common Transfer Functions

Defining an Ordinary Regression

```
proc arima data=<data>;
   identify var=Y cross=(X) nlags=12;
   estimate p=1 q=1
            input=(X)
            method=ml;
run;
```

108

Common Transfer Functions

Ordinary Regression with One Lag Term

$$\omega(B) = \omega_0 + \omega_1 B, \delta(B) = 1$$

Model

$$Y_t = \theta_0 + \omega_0 X_t + \omega_1 X_{t-1} + Z_t$$
$$\phi(B) Z_t = \theta(B) \varepsilon_t$$

109

Common Transfer Functions

Defining an Ordinary Regression with One Lag Term

```
proc arima data=<data>;
   identify var=Y cross=(X) nlags=12;
   estimate p=1 q=1
            input=((1) X)
            method=ml;
run;
```

110

Common Transfer Functions

Ordinary Regression with One Shifted Term

$$\omega(B) = \omega_k B^k, \delta(B) = 1$$

Model

$$Y_t = \theta_0 + \omega_k X_{t-k} + Z_t$$
$$\phi(B) Z_t = \theta(B) \varepsilon_t$$

111

Common Transfer Functions

Defining an Ordinary Regression with One Shifted Term (k=3)

```
proc arima data=<data>;
   identify var=Y cross=(X) nlags=12;
   estimate p=1 q=1
            input=(3$X)
            method=ml;
run;
```

112

Common Transfer Functions

Ordinary Regression with One Shifted and One Lag Term

$$\omega(B) = \omega_1 B + \omega_2 B^2, \delta(B) = 1$$

Model

$$Y_t = \theta_0 + \omega_1 X_{t-1} + \omega_2 X_{t-2} + Z_t$$
$$\phi(B) Z_t = \theta(B) \varepsilon_t$$

113

Common Transfer Functions

Defining an Ordinary Regression with One Shifted and One Lag Term

```
proc arima data=<data>;
   identify var=Y cross=(X) nlags=12;
   estimate p=1 q=1
            input=(1$(1) X)
            method=ml;
run;
```

114

Common Transfer Functions

Infinite Past Regression

$$\omega(B)=\omega_0,\delta(B)=1-\delta_1 B$$

Model

$$Y_t=\theta_0+\frac{\omega_0}{1-\delta_1 B}X_t+Z_t$$

$$\phi(B)Z_t=\theta(B)\varepsilon_t$$

115

continued...

Common Transfer Functions

Defining an Infinite Past Regression

```
proc arima data=<DATA>;
   identify var=Y cross=(X) nlags=12;
   estimate p=1 q=1
            input=(/(1) X)
            method=ml;
run;
```

116

Box-Jenkins Methodology for Dynamic Regression

Identify Step

- Determine which input variables to include in the model.
- Determine the numerator and denominator order of the transfer function for each input variable.
- Determine trend and seasonality.
- Determine the ARMA(p,q) error model.

117

5.03 Multiple Choice Poll

How many candidate input variables do you typically encounter in a forecasting project?

a. 0
b. 1-10
c. 11-50
d. 51-100
e. More than 100

119

Special Considerations for Stochastic Inputs

- Trend and seasonal components are removed before you investigate the dynamic relationship between the input and target.
- After you remove trend and seasonal effects, each input can be evaluated with respect to the target variable using a *cross-correlation* plot.
- Additionally, a *prewhitening filter* should be applied to the input and target before calculating cross-correlations. The cross-correlation function with optional prewhitening is provided by PROC ARIMA.

120

Dynamic Regression Using PROC ARIMA

- Derive a univariate forecast model for each stochastic input variable, with differencing in the IDENTIFY step reducing the series to stationarity. The ARMA model specified in the ESTIMATE statement should produce white noise residuals.
- Apply differencing to reduce the target to a stationary time series.
- Prewhiten the stationary residuals of the target series by applying the ARMA model specified in the ESTIMATE statement of the input series.
- Cross-correlate the white noise input residuals with the prewhitened target residuals, and use the cross-correlations to identify the transfer function for the input.
- Specify the full dynamic regression model for all inputs, and then diagnose an ARMA(p,q) model for the stationary error component.
- Forecast the inputs, and use these forecasts to derive a forecast for the target.

121

Cross-Correlations and Prewhitening

- Prewhitening only works when all input variables to be used in a model are uncorrelated with each other.
- Filtering using only trend and seasonal components produces a CCF that will be difficult to interpret with respect to fully specifying a transfer function.
- Significant values of the CCF for negative lags imply that the target depends on future values of the input, so spikes for negative lags do not influence selection of a transfer function.
- A CCF obtained after prewhitening can suggest an appropriate transfer function.

122

Interpreting the Prewhitened CCF

If the CCF for input **X** depicts the following:

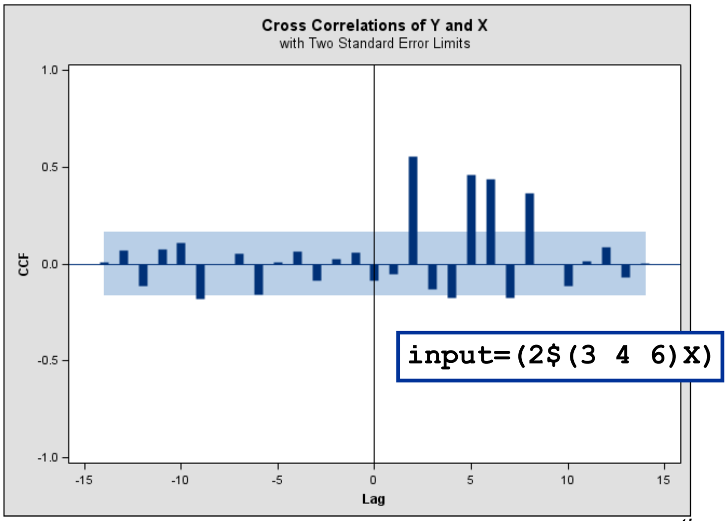
- A set of *m* spikes occurring at lags k_1, k_2, …, k_m with no significant pattern, then the transfer function specification should be **INPUT=($k_1$$(k_2-k_1 ,…, k_m- k_1)X)**
- A set of decaying spikes beginning at lag *k*, then the transfer function specification should be **INPUT=(*k*$/(1)X)**
- A set of *m*+1 spikes beginning at lag *k* followed by decaying spikes, then the specification should be **INPUT=(*k*$(1 2 … *m*)/(1)X)**

123

continued...

The values, such as k_2-k_1, are calculated and supplied. Thus, if you see spikes at lags 4, 7, and 10, you specify **INPUT=(4$(3 6)X)**. The next slide illustrates this with spikes at lags 2, 5, 6, and 8.

Interpreting the Prewhitened CCF

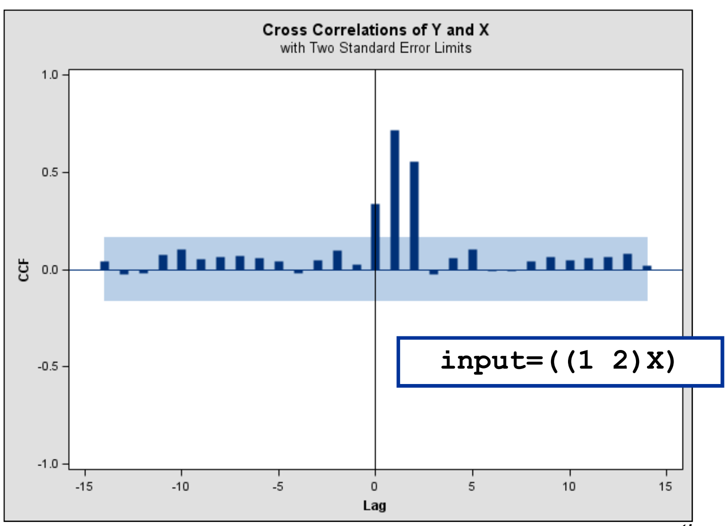


124

continued...

The spikes at lags 4 and 7 are small relative to the other spikes and are treated as spurious.

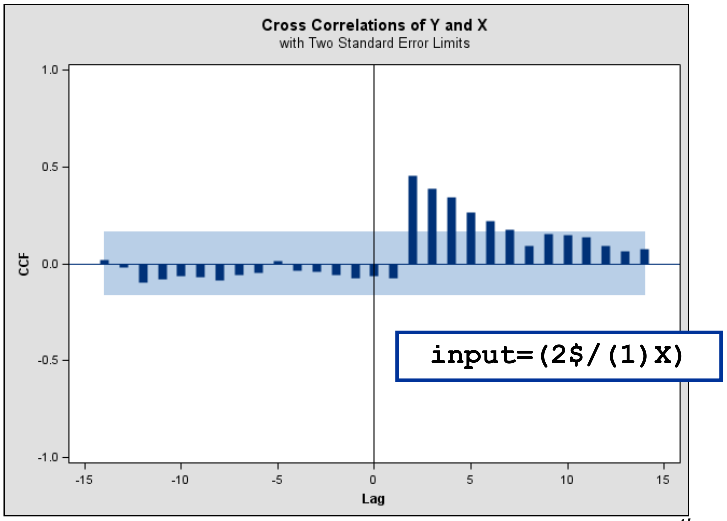
Interpreting the Prewhitened CCF



125

continued...

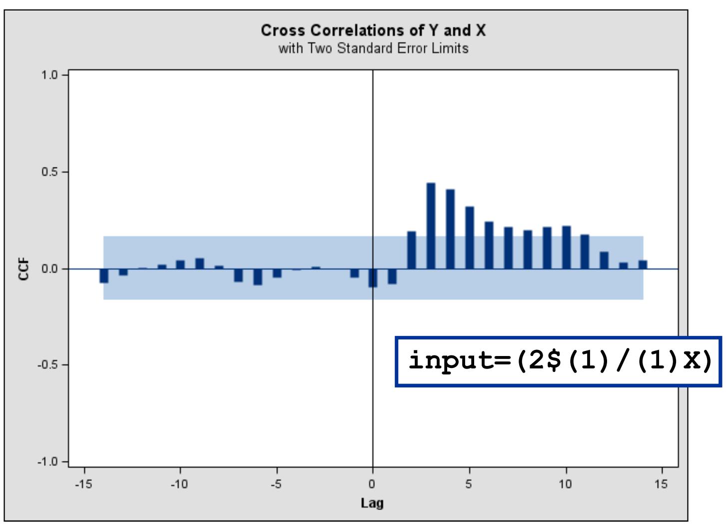
Interpreting the Prewhitened CCF



126

continued...

Interpreting the Prewhitened CCF



127

Because the exponential decay begins at the second spike, two numerator parameters are required.

Pre-Whitening

This demonstration illustrates how pre-whitening is used for transfer function identification.

The SAS code for this demonstration can be found in **Demo5_04TransferFunction.sas**.

A useful way to illustrate pre-whitening is to use simulated data where the model is known. The following code uses macro SIMARMA to simulate stationary time series for use in constructing a transfer function model. The variable X is simulated to be an AR(1) process, and X affects Y through a transfer function with a two-month delay and an exponential decay with a decay rate of approximately 0.6.

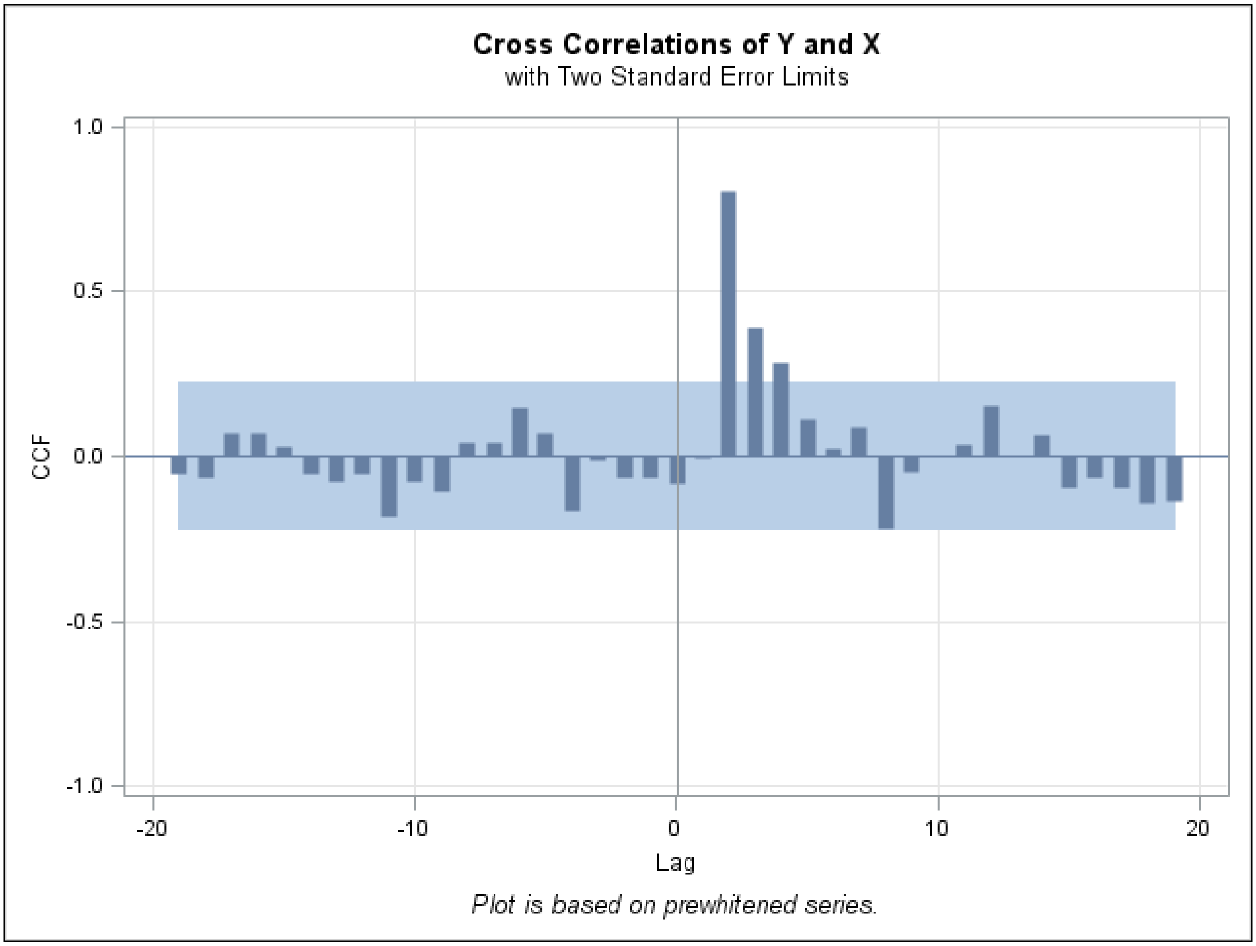
```
/*----  Simulate X  ----*/
%SimARMA(work.tempX,1 -0.65,1,50,2.1,108,
         '01JUL2001'd,month,274307);
/*----  Simulate Error  ----*/
%SimARMA(work.tempE,1 -0.4,1,0.0,0.6,108,
         '01JUL2001'd,month,4202301);
data work.tempall;
   merge work.tempX(rename=(Y=X)) work.tempE(rename=(Y=E));
   by Date;
   retain X1 X2 X3 X4 X5 X6 X7 50;
   Y=200+1.2*X2+0.6*X3+0.36*X4+0.216*X5+0.129*X6+0.06*X7+E;
   if (Date>='01JAN2004'd) then output;
   X7=X6;
   X6=X5;
   X5=X4;
   X4=X3;
   X3=X2;
   X2=X1;
   X1=X;
   keep Y X Date;
run;
```

In PROC ARIMA, the first IDENTIFY and ESTIMATE statements build a model for X. This model is then used to pre-whiten X and Y to calculate the cross-correlation function.

```
proc arima data=work.tempall;
   identify var=X;
   estimate p=1 ml;
   identify var=y crosscorr=(x);
   estimate p=1 input=(1$/(1)X) ml;
   identify var=y crosscorr=(x) clear;
quit;
```

The cross-correlation function suggests the form of the transfer function. The second ESTIMATE statement fits the full transfer function model. The steps to identify the error component have been omitted.

Here is the cross-correlation plot:

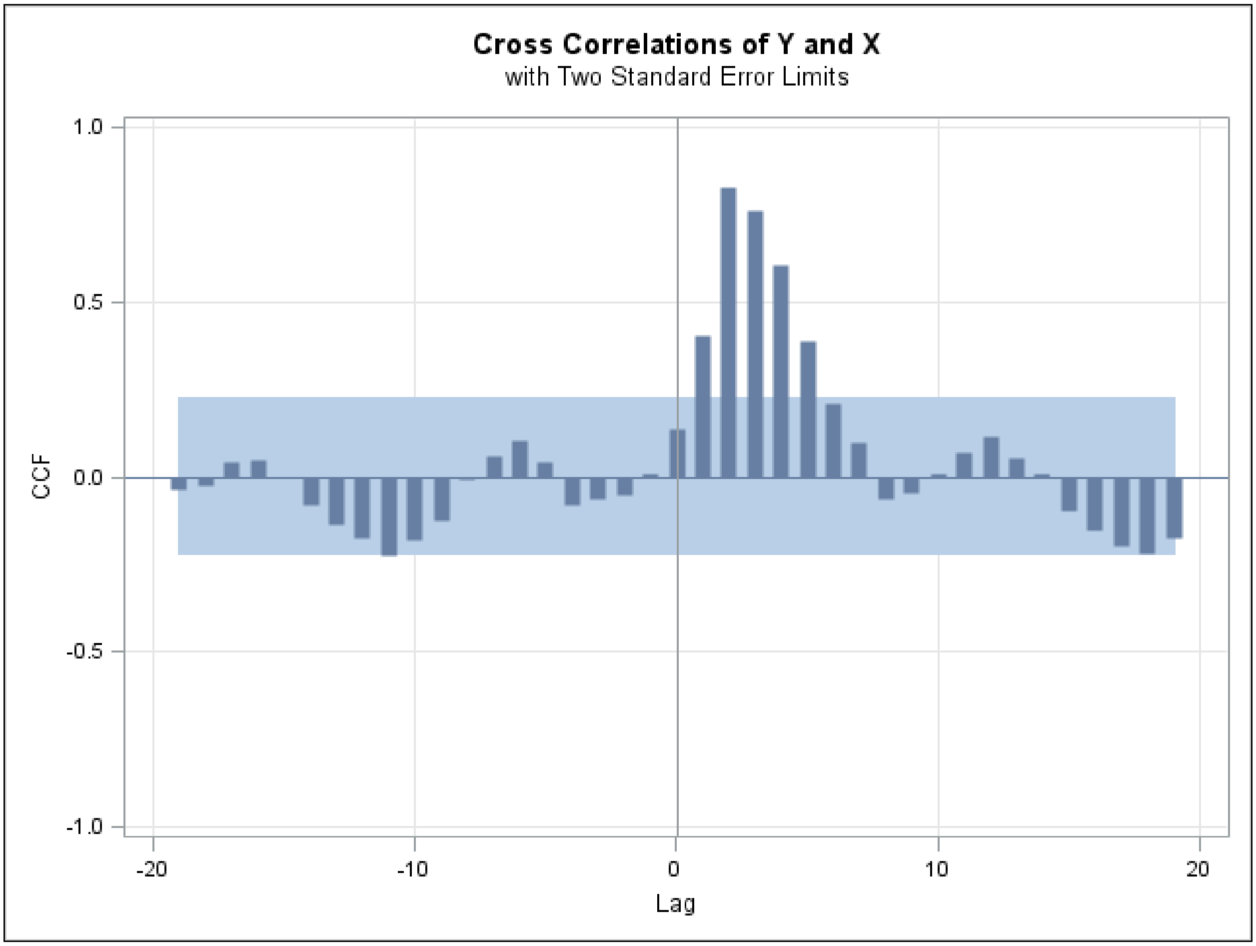


The spike at lag 2 followed by exponential decay suggests the transfer function specified by

INPUT=(1$/(1)X)

The last IDENTIFY statement uses the CLEAR option, which removes any pre-whitening filtering.

The cross-correlation plot without pre-whitening follows.



Without pre-whitening, spurious results appear for lag 1.

The next demonstration illustrates dynamic regression with stochastic inputs and transfer function specification using domain knowledge.

Evaluating Advertising Effectiveness

This demonstration illustrates how to use PROC ARIMA to evaluate the effect of advertising expenditures on sales.

The SAS code for this demonstration can be found in **Demo5_05Advertising.sas**.

An Internet start-up company provides financial and brokerage services to customers. A customer sets up an account with an initial deposit of $1,000, and then the company provides services via the Internet or by phone. Services include a variety of banking and investment options such as automatic debits for paying bills, on-demand transaction and tax information, automatic money market investing for positive balances, and online brokerage services for investing in stocks and mutual funds.

The full data contains extensive breakdowns of advertising dollars. For example, for print media advertising, dollar amounts are recorded for each individual newspaper or magazine in which ads were placed. This illustrates a common situation where there are almost as many variables as there are observations. Another problem results from some accounts being tabulated monthly, while others are tabulated weekly. Many advertising accounts are dominated by zeros, meaning that no advertising dollars were spent on the account for several months in a row.

The response variable is the total sales amount for all customers. This amount reflects revenue generated by account activity such as deposits, investments, withdrawals, and so on. Finally, data is obtained from a data vendor that tabulates total sales amounts for a subset of the companies engaged in this type of Internet business.

Many of the time-consuming and tedious steps of the analysis are excluded. Some actions include aggregating data to the same time unit and deriving candidate predictor variables. The demonstration starts after the data is processed into a form suitable for final modeling.

The predictor variables are shown here:

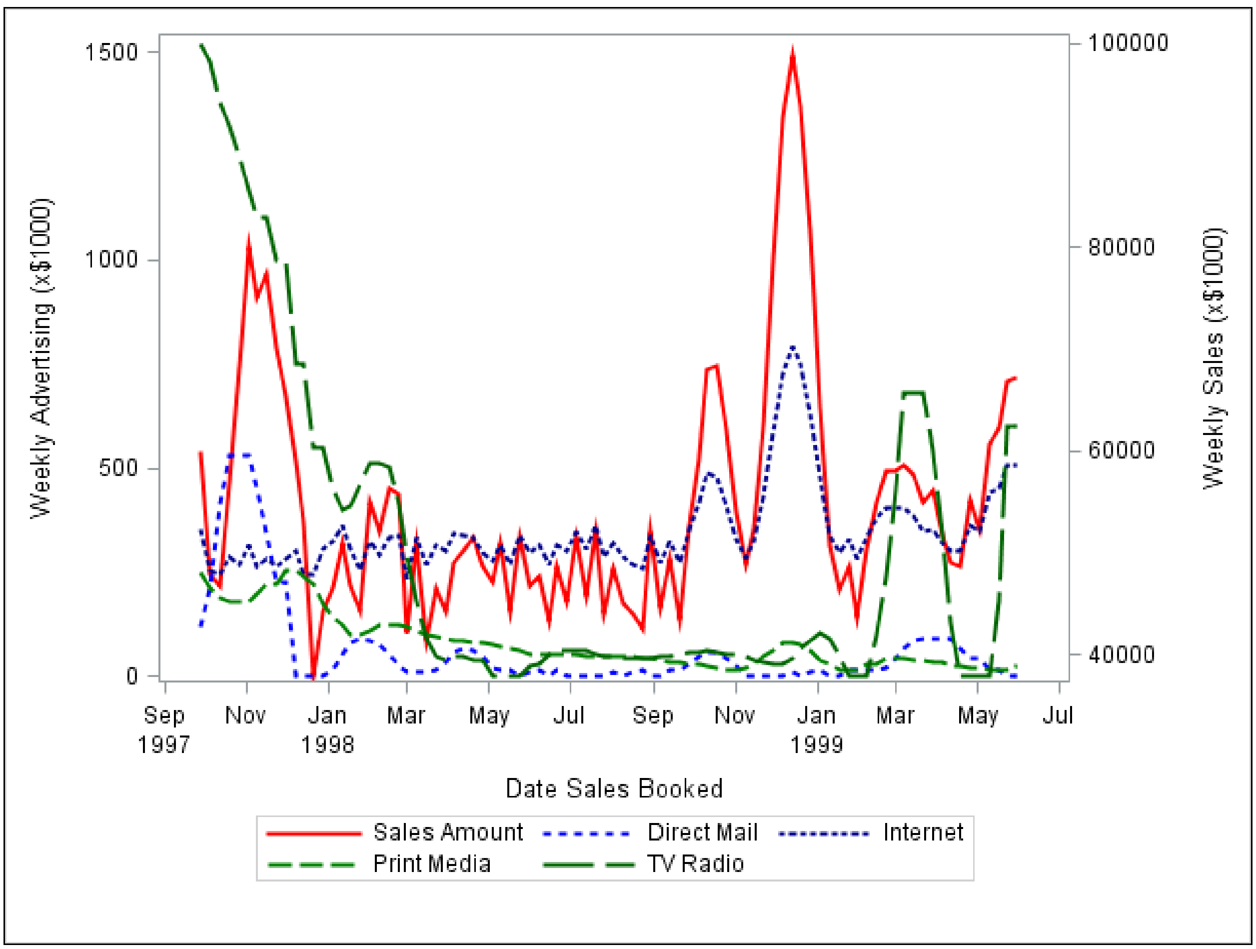
Predictor Variable	Description
DirectMail	Weekly direct mail advertising (×$1000)
Internet	Weekly Internet advertising (×$1000)
PrintMedia	Weekly print media advertising (×$1000)
SalesRatio	Ratio of competitor sales to total known sales
TVRadio	Weekly TV/radio advertising (×$1000)

The target variable is **SalesAmount**, which is total sales in thousands of dollars. An additional variable used in deriving **SalesRatio** is **CompeteSales**, which is the vendor-supplied value of sales for competing companies in the Internet domain. The data set that contains these variables is **SASUSER.SALEDATA**, which has 88 weekly observations from the week of 28 September 1997 to the week of 30 May 1999.

The following code produces a plot for the data:

```
proc sgplot data=work.temp;
   series x=Date y=SalesAmount / y2axis
          lineattrs=GraphPrediction(pattern=1 color=red)
          legendlabel="Sales Amount" name="series0";;
   series x=Date y=DirectMail /
          lineattrs=GraphPrediction(pattern=2 color=blue)
          legendlabel="Direct Mail" name="series1";
   series x=Date y=Internet /
          lineattrs=GraphPrediction(pattern=3 color=darkblue)
          legendlabel="Internet" name="series2";
   series x=Date y=PrintMedia /
          lineattrs=GraphPrediction(pattern=4 color=green)
          legendlabel="Print Media" name="series3";
   series x=Date y=TVRadio /
          lineattrs=GraphPrediction(pattern=5 color=darkgreen)
          legendlabel="TV Radio" name="series4";
   keylegend "series0" "series1" "series2" "series3" "series4"/
          location=outside position=bottom;
run;
```

The plot is shown below:



The plot suggests that the influence of all input variables is aligned with the target series, except **DirectMail**, which appears to be a leading indicator of **SalesAmount** by two weeks. A time series regression model with the regressors **Internet**, **PrintMedia**, **TVRadio**, and **SalesRatio**, and the dynamic regressor **DirectMail** shifted two weeks is fit to the data.

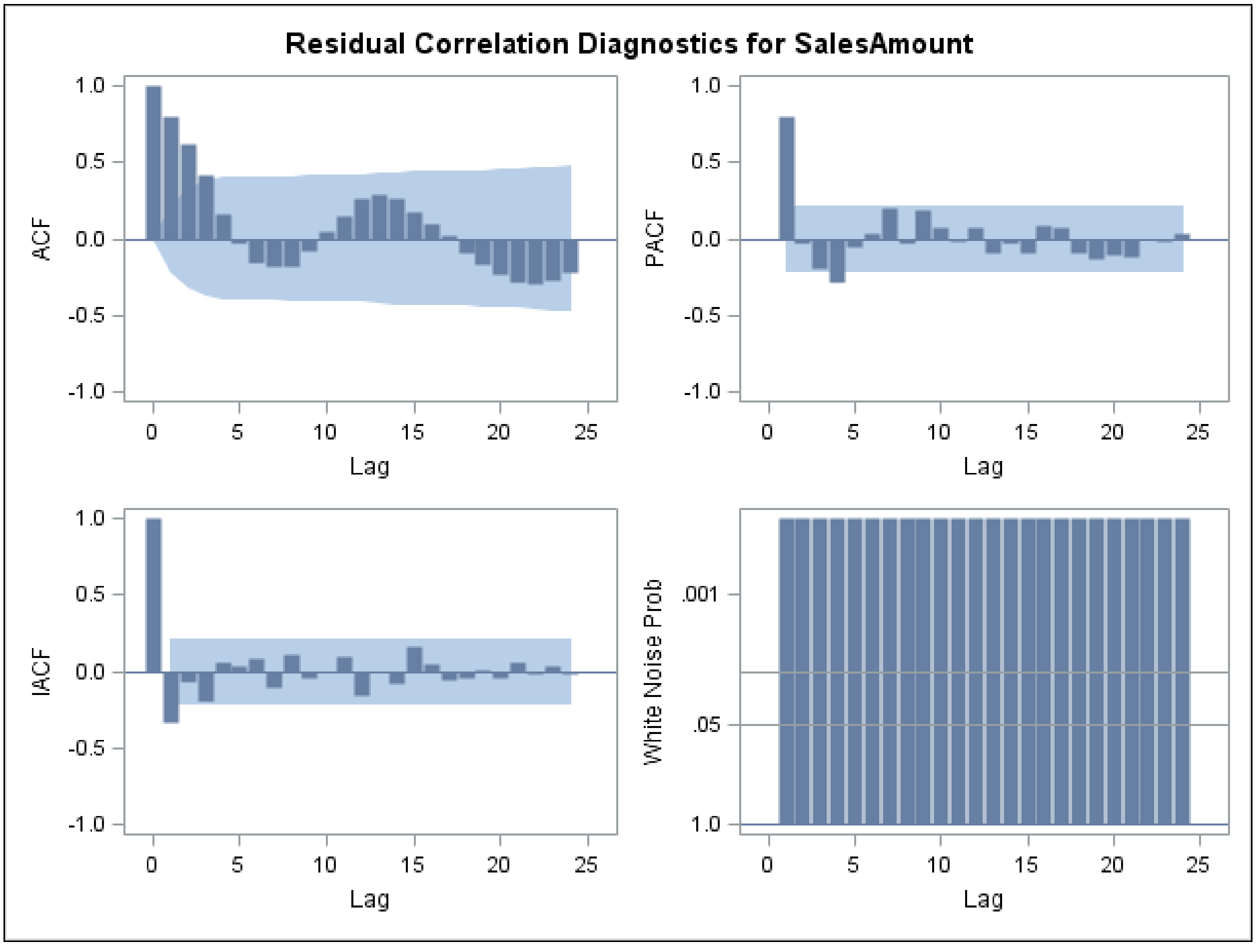
```
proc arima data=work.SaleData;
   identify var=SalesAmount
            crosscorr=(PrintMedia
                       TVRadio
                       Internet
                       SalesRatio
                       DirectMail)
            nlag=24;
   estimate input=(PrintMedia
                   TVRadio
                   Internet
                   SalesRatio
                   2 $ DirectMail)
            method=ML
            outest=work.SDest1
            plot;
   forecast id=DATE interval=WEEK align=BEGINNING
            lead=0
            out=work.SDfore1;
run;
```

The table of estimates follows:

Maximum Likelihood Estimation							
Parameter	Estimate	Standard Error	t Value	Approx Pr > |t|	Lag	Variable	Shift
MU	45934.1	6950.4	6.61	<.0001	0	SalesAmount	0
NUM1	-12.89303	3.98145	-3.24	0.0012	0	PrintMedia	0
NUM2	0.61620	0.63200	0.97	0.3296	0	TVRadio	0
NUM3	92.74618	3.34100	27.76	<.0001	0	Internet	0
NUM4	-41441.4	8835.7	-4.69	<.0001	0	SalesRatio	0
NUM5	51.89434	2.98076	17.41	<.0001	0	DirectMail	2

The estimation results imply that TV and radio advertising might not have a significant impact on sales. You want to keep **TVRadio** in the model to estimate its effect, even if the effect is not statistically significant.

The Ljung-Box test and the residual plots produced by the above code suggest that the residuals are not white noise.



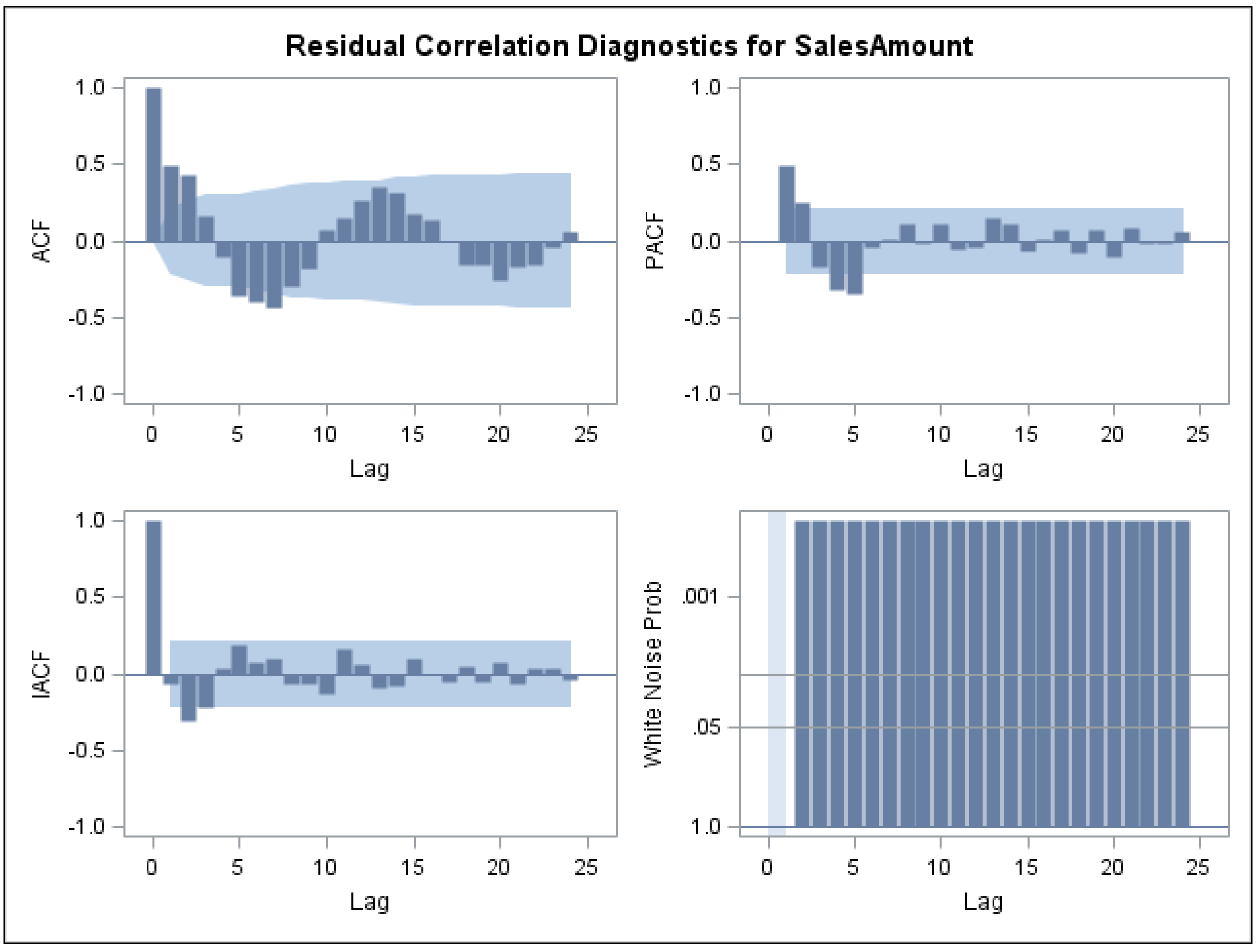
The following code adds an ARMA(1,0) error component:

```
proc arima data=work.SaleData plots=all;
   identify var=SalesAmount
            crosscorr=(PrintMedia
                       TVRadio
                       Internet
                       SalesRatio
                       DirectMail)
            nlag=24 noprint;
   estimate p=1
            input=(PrintMedia
                   TVRadio
                   Internet
                   SalesRatio
                   2 $ DirectMail)
            method=ML
            outest=work.SDest2
            plot;
   forecast id=DATE interval=WEEK align=BEGINNING
            lead=0
            out=work.SDfore2;
quit;
```

The table of estimates follows:

Maximum Likelihood Estimation							
Parameter	Estimate	Standard Error	t Value	Approx Pr > |t|	Lag	Variable	Shift
MU	19019.8	2657.1	7.16	<.0001	0	SalesAmount	0
AR1,1	0.97200	0.02293	42.39	<.0001	1	SalesAmount	0
NUM1	-1.01109	4.18735	-0.24	0.8092	0	PrintMedia	0
NUM2	1.82350	0.52136	3.50	0.0005	0	TVRadio	0
NUM3	103.37059	1.50945	68.48	<.0001	0	Internet	0
NUM4	-9091.5	2870.0	-3.17	0.0015	0	SalesRatio	0
NUM5	62.81360	1.35755	46.27	<.0001	0	DirectMail	2

The Ljung-Box test and residual diagnostics follow:



The residuals are clearly not white noise. The ARMA(1,0) error component is clearly not adequate.

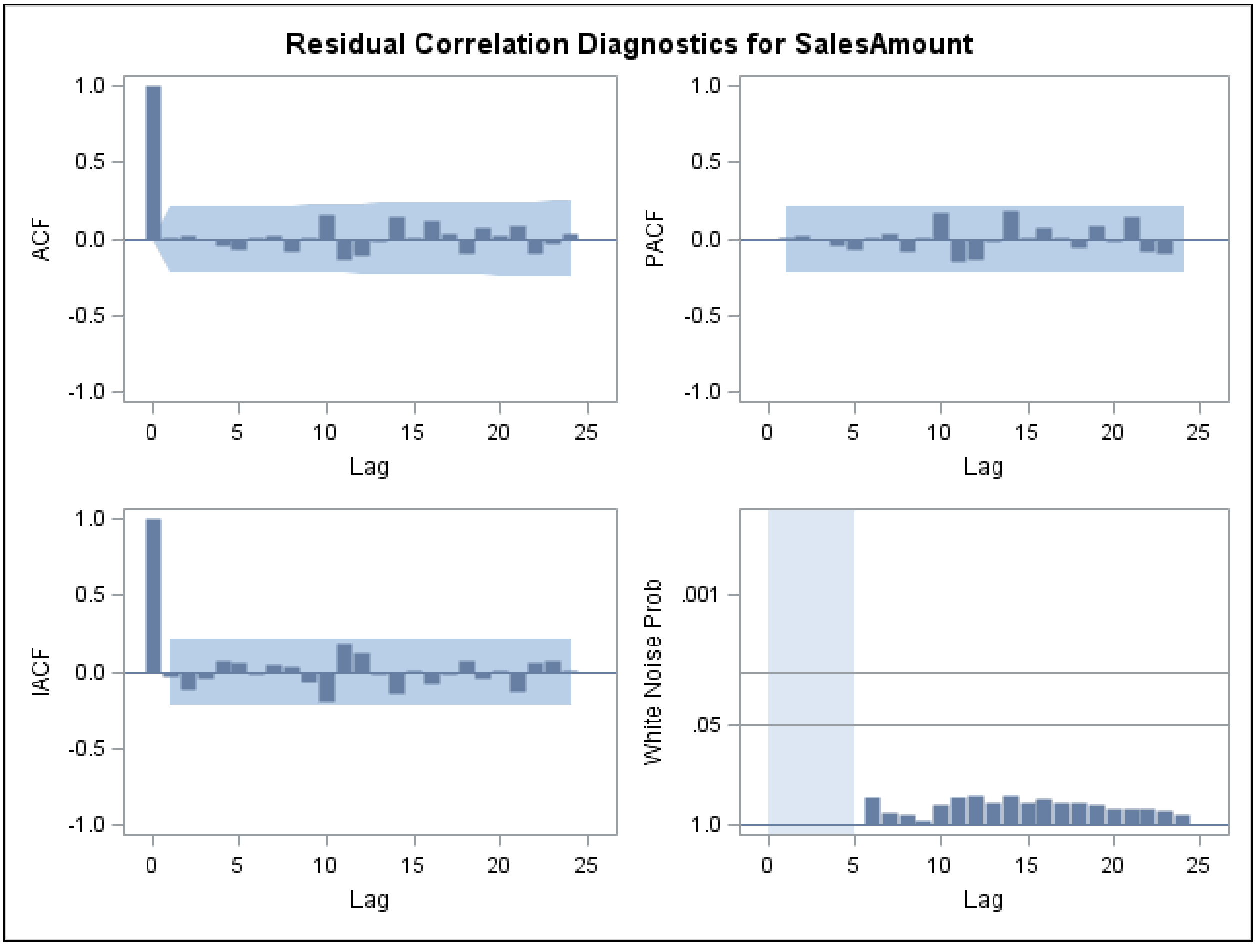
You can investigate the implications of changing the P=1 option to P=5 to see whether things improve.

```
proc arima data=work.SaleData plots=all;
   identify var=SalesAmount
            crosscorr=(PrintMedia
                       TVRadio
                       Internet
                       SalesRatio
                       DirectMail)
            nlag=24 noprint;
   estimate p=5
            input=(PrintMedia
                   TVRadio
                   Internet
                   SalesRatio
                   2 $ DirectMail)
            method=ML
            outest=work.SDest3
            plot;
   forecast id=DATE interval=WEEK align=BEGINNING
            lead=0
            out=work.SDfore3;
quit;
```

The table of estimates follows:

Maximum Likelihood Estimation							
Parameter	Estimate	Standard Error	t Value	Approx Pr > |t|	Lag	Variable	Shift
MU	15989.9	1536.4	10.41	<.0001	0	SalesAmount	0
AR1,1	1.70930	0.11277	15.16	<.0001	1	SalesAmount	0
AR1,2	-0.77469	0.23155	-3.35	0.0008	2	SalesAmount	0
AR1,3	0.08970	0.24270	0.37	0.7117	3	SalesAmount	0
AR1,4	-0.36429	0.22365	-1.63	0.1033	4	SalesAmount	0
AR1,5	0.30738	0.11002	2.79	0.0052	5	SalesAmount	0
NUM1	2.65600	4.09752	0.65	0.5169	0	PrintMedia	0
NUM2	0.99756	0.43875	2.27	0.0230	0	TVRadio	0
NUM3	107.80637	0.90194	119.53	<.0001	0	Internet	0
NUM4	-6289.7	1426.7	-4.41	<.0001	0	SalesRatio	0
NUM5	64.98103	0.95352	68.15	<.0001	0	DirectMail	2

The Ljung-Box test and autocorrelation plots imply white noise residuals.



Estimation results pass face validity. However, **PrintMedia** becomes insignificant and **TVRadio** becomes significant. However, the **PrintMedia** coefficient implies that $2.66 is returned for every $1 invested in print media advertising. The **TVRadio** coefficient implies a return of $.99 for every $1 spent, which is close to breaking even. Because monetary variables are expressed in thousands of dollars, you could interpret a coefficient of 2.66 as implying that a one unit ($1,000) increase in **PrintMedia** expenditures produces 2.66 units ($2,660) increase in **SalesAmount**.

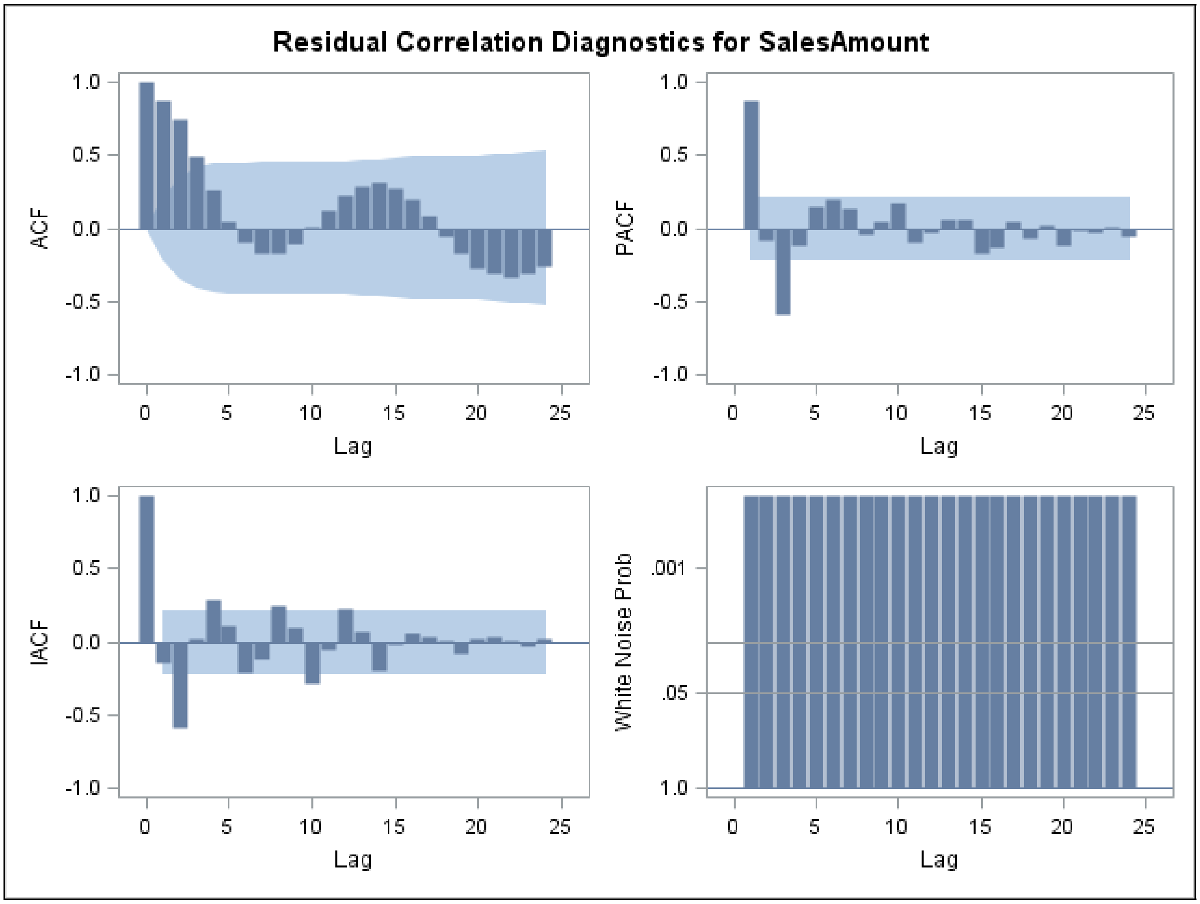
Before the model is accepted, experts gather to discuss the implications. One executive suggests that advertising expenditures are impacted by the results on competitor sales, and thus **SalesRatio** should be a leading indicator of sales by one week.

This conjecture is tested by fitting a model using the following code:

```
proc arima data=work.SaleData plots=all;
   identify var=SalesAmount
            crosscorr=(PrintMedia
                       TVRadio
                       Internet
                       SalesRatio
                       DirectMail)
            nlag=24 noprint;
   estimate /* p=5 */
            input=(PrintMedia
                   TVRadio
                   Internet
                   1 $ SalesRatio
                   2 $ DirectMail)
            method=ML
            outest=work.SDest4
            plot;
   forecast id=DATE interval=WEEK align=BEGINNING
            lead=0
            out=work.SDfore4;
run;
quit;
```

Notice that P=5 is commented out, because the structural change in the model forces reassessment of the error model.

A quick look at the residual diagnostics reveals a similar problem to what you initially encountered.



The autocorrelation plots are examined to suggest an ARMA error component. While the IACF suggests an AR(12) model, the PACF suggests an AR(3) model. You can compromise with an AR(5) error term that resembles the one used before.

```
proc arima data=work.SaleData plots=all;
   identify var=SalesAmount
            crosscorr=(PrintMedia
                       TVRadio
                       Internet
                       SalesRatio
                       DirectMail)
            nlag=24 noprint;
   estimate p=5
            input=(PrintMedia
                   TVRadio
                   Internet
                   1 $ SalesRatio
                   2 $ DirectMail)
            method=ML
            outest=work.SDest5
            plot;
   forecast id=DATE interval=WEEK align=BEGINNING
            lead=0
            out=work.SDfore5;
run;
quit;
```

The estimates follow:

Maximum Likelihood Estimation							
Parameter	Estimate	Standard Error	t Value	Approx Pr > |t|	Lag	Variable	Shift
MU	6387.7	682.63596	9.36	<.0001	0	SalesAmount	0
AR1,1	3.72826	0.09663	38.58	<.0001	1	SalesAmount	0
AR1,2	-5.84263	0.34052	-17.16	<.0001	2	SalesAmount	0
AR1,3	4.80599	0.49113	9.79	<.0001	3	SalesAmount	0
AR1,4	-2.08383	0.34320	-6.07	<.0001	4	SalesAmount	0
AR1,5	0.38819	0.09823	3.95	<.0001	5	SalesAmount	0
NUM1	11.03545	0.11822	93.35	<.0001	0	PrintMedia	0
NUM2	0.07122	0.02040	3.49	0.0005	0	TVRadio	0
NUM3	107.17046	0.02183	4908.99	<.0001	0	Internet	0
NUM4	8938.6	32.89295	271.75	<.0001	0	SalesRatio	1
NUM5	66.63791	0.03426	1945.05	<.0001	0	DirectMail	2

All parameter estimates are significant at the 1% level. However, the assessment of **TVRadio** fell to seven cents on the dollar.

While the executive's conjecture might be correct, you still have a competing model. The difference in estimates should influence you to closely scrutinize each model. Because neither model can be disqualified, review the accuracy statistics for each model.

The last model produces SBC=898.8, whereas the prior competing model produces SBC=1263.0. SBC favors the executive's model.

The following code calculates goodness-of-fit statistics:

```
%GOFstats(ModelName=%str(Model 1 with no ARMA Error),
          DSName=work.SDfore1,
          OutDS=work.SDfore1_m,
          ActualVar=SalesAmount,
          ForecastVar=FORECAST,
          NumParms=6);
%GOFstats(ModelName=%str(Model 2 with AR(1) Error),
          DSName=work.SDfore2,
          OutDS=work.SDfore2_m,
          ActualVar=SalesAmount,
          ForecastVar=FORECAST,
          NumParms=7);
%GOFstats(ModelName=%str(Model 3 with AR(5) Error),
          DSName=work.SDfore3,
          OutDS=work.SDfore3_m,
          ActualVar=SalesAmount,
          ForecastVar=FORECAST,
          NumParms=11);
%GOFstats(ModelName=%str(Model 4 with no ARMA Error),
          DSName=work.SDfore4,
          OutDS=work.SDfore4_m,
          ActualVar=SalesAmount,
          ForecastVar=FORECAST,
          NumParms=6);
%GOFstats(ModelName=%str(Model 5 with ARMA Error),
          DSName=work.SDfore5,
          OutDS=work.SDfore5_m,
          ActualVar=SalesAmount,
          ForecastVar=FORECAST,
          NumParms=11);

data work.all;
   set work.SDfore1_m
       work.SDfore2_m
       work.SDfore3_m
       work.SDfore4_m
       work.SDfore5_m;
run;

proc print data=work.all noobs;
run;
```

Calculation of MAPE and RMSE for all models produces the following table:

Model	MAPE	RMSE	NumParm	AIC_SSE	SBC_SSE
Model 1 with no ARMA Error	1.58647	1148.12	6	1217.67	1232.40
Model 2 with AR(1) Error	0.62627	454.67	7	1059.26	1076.44
Model 3 with AR(5) Error	0.44726	332.10	11	1008.76	1035.76
Model 4 with no ARMA Error	1.80425	1286.09	6	1237.19	1251.92
Model 5 with ARMA Error	0.09412	207.94	11	928.24	955.24

A properly specified error component can have a substantial impact on accuracy and inference.

For this problem, the business intelligence is that Internet and direct-mail advertising have the biggest impact on sales. The business professional can perform a what-if analysis using the forecast models with various values for advertising expenses. However, remember that extrapolation beyond the data can be dangerous. For example, if $500,000 is the most that was spent for Internet advertising, it would be foolish to expect that spending $50,000,000 on Internet advertising will increase sales by $5 billion.

Exercises

The SAS data set **SASUSER.MVAInjuries** contains time series data related to injuries sustained in motor vehicle accidents in a desert vacation destination. Public policy analysts are interested in the effects of two legislative actions:

- In April 1994, a new law took effect that changed the number of months that a driver was seizure free from 12 to three months in order for them to operate a motor vehicle.
- In June 1999, a law restricted the driving privileges of teenage drivers.

Analysts are also aware that the events of September 11, 2001, could impact tourism, and thus indirectly influence motor vehicle accident frequency.

Step functions exist in the data for these three events.

1. Add three pulse functions for the events to the data, and add a variable that is the ratio of licensed drivers to estimated population.

2. Use a difference of order 1 and order 12 to model trend and seasonality.

3. Find an appropriate intervention model to determine the effects of the three events on motor vehicle accident injuries.

5.4 Chapter Summary

Time series regression analysis is an extension of ordinary regression analysis. Time series regression is known as dynamic regression when one or more of the input variables are correlated with the target variable at non-concurrent time lags.

Time series regression inherits the usual regression challenges: input variable selection, multicollinearity, and influential observations. However, time series regression solves the problem of autocorrelated errors.

The partial correlation coefficient that expresses how an input variable influences the target in ordinary regression becomes a set of coefficients that comprise the transfer function in time series regression. The rational transfer function devised by Box, Jenkins, and Tiao permits estimation of complex relationships with relatively few parameters to accommodate the limited data that often restricts time series regression model fitting.

Event models use transfer functions applied to point or step function variables. The use of point, step, and ramp variables combined with rational transfer functions provides a rich set of event effects.

Dynamic regression adds additional challenges to variable selection in regression. An input can be uncorrelated with the target variable concurrently in time, but the input can be strongly correlated with the target when it is shifted by one or more time units. Consequently, variable selection includes additional steps because of the addition of the time dimension.

For Additional Information

Box, G.E.P., and Jenkins, G.M. 1976. *Time Series Analysis: Forecasting and Control, Revised Edition*. San Francisco: Holden-Day.

Box, G.E.P., and Tiao, G.C. 1975. "Intervention Analysis with Applications to Economic and Environmental Problems." *JASA*. **70**, 70-79.

Broadbent, S. 1998. *Accountable Advertising: Handbook for Managers and Analysts*. Washington, DC: Admap.

Brocklebank, J.C., and Dickey, D.A. 2003. *SAS System for Forecasting Time Series, Second Edition*. Cary, North Carolina: SAS Institute Inc.

Cook, R.D., and Weisberg, S. 1999. *Applied Regression Including Computing and Graphics*. New York: John Wiley & Sons, Inc.

Diggle, P.J. 1990. *Time Series: A Biostatistical Introduction*. Oxford: Clarendon Press.

Judge, G.G., Griffiths, W.E., Hill, R.C., Lütkepohl, and Lee, T.C. 1985. *The Theory and Practice of Econometrics, Second Edition*. New York: John Wiley & Sons, Inc.

Marquardt, D.W. 1970. "Generalized Inverses, Ridge Regression, Biased Linear Estimation, and Nonlinear Estimation," *Technometrics*. 12(3), 591, 605–07.

Pankratz, A. 1991. *Forecasting with Dynamic Regression Models*. New York: John Wiley & Sons, Inc.

West, S.G., McCall, M.A., Hepworth, J.T., and Reich, J.W. 1985. "An Evaluation of Arizona's 1982 Drunk Driving Law: Effects on the City of Phoenix." Final report to the Morrison Institute.

Woodfield, T.J. 1987. "Time Series Intervention Analysis Using SAS Software." *Proceedings of the Twelfth Annual SAS Users Group International Conference*, 331-339. Cary, North Carolina: SAS Institute Inc.

5.5 Solutions

Solutions to Exercises

The SAS data set **SASUSER.MVAInjuries** contains time series data related to injuries sustained in motor vehicle accidents in a desert vacation destination. Public policy analysts are interested in the effects of two legislative actions:

- In April 1994, a new law took effect that changed the number of months that a driver was seizure free from 12 to three months in order for them to operate a motor vehicle.
- In June 1999, a law restricted the driving privileges of teenage drivers.

Analysts are also aware that the events of September 11, 2001, could impact tourism, and thus indirectly influence motor vehicle accident frequency.

Step functions exist in the data for these three events.

1. Add three pulse functions for the events to the data, and add a variable that is the ratio of licensed drivers to estimated population.

2. Use a difference of order 1 and order 12 to model trend and seasonality.

3. Find an appropriate intervention model to determine the effects of the three events on motor vehicle accident injuries.

The program **Exercises_Ch5.sas** contains code to for this exercise. Only a small portion of the code is presented here. A candidate model is given by the following PROC ARIMA code:

```
proc arima data=work.MVAInjuries plots=all;
   identify var=Injuries(1 12)
            cross=(Seizure(1 12)
                   SEP2001(1 12)
                   TouristIndex(1 12))
            nlags=24 noprint;
   estimate p=0 q=(1)(12) noconstant
            input=(Seizure
                   SEP2001
                   TouristIndex)
            method=ml maxiter=150 plot;
run;
quit;
```

The table of estimates follows:

Maximum Likelihood Estimation							
Parameter	Estimate	Standard Error	t Value	Approx Pr > |t|	Lag	Variable	Shift
MA1,1	0.74905	0.04873	15.37	<.0001	1	Injuries	0
MA2,1	0.85301	0.06964	12.25	<.0001	12	Injuries	0
NUM1	66.27232	41.80861	1.59	0.1129	0	Seizure	0
NUM2	222.23353	55.86925	3.98	<.0001	0	Sep2001	0
NUM3	1.47262	0.15608	9.43	<.0001	0	TouristIndex	0

Variance Estimate	4462.958
Std Error Estimate	66.80538
AIC	2145.99
SBC	2162.199
Number of Residuals	189

The loosening of the seizure law is estimated to increase the number of injuries by about 66 each month. Tourism adds about 1.5 injuries for every one unit increase in the tourist index. The interpretation of the September 11, 2001, effect is counter-intuitive. There could be a problem with how the SEP2001 step function is correlated with the tourist index. The removal of the teenage driver intervention effect resulted from a lack of statistical significance.

Solutions to Student Activities (Polls/Quizzes)

5.01 Multiple Choice Poll – Correct Answer

An outlier can be related to which of the following?

a. A miscoded data value
b. A source of variation left out of the model
c. A suspicious or unusual data value that can bias forecasts
d. All of the above

57

5.02 Multiple Choice Poll – Correct Answer

Which of the following represents the correct specification of an abrupt temporary intervention effect in PROC ARIMA?

a. INPUT=(/(1)PULSE)
b. INPUT=(/(1)STEP)
c. INPUT=(/(1)RAMP)
d. INPUT=(STEP RAMP)
e. INPUT=(PULSE RAMP)

98