Time Series Modeling Essentials

Course Notes

Time Series Modeling Essentials Course Notes was developed by George Fernandez, Marc Huber, Jay Laramore, Danny Modlin, and Chip Wells. Editing and production support was provided by the Curriculum Development and Support Department.

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

Time Series Modeling Essentials Course Notes

Copyright © 2016 SAS Institute Inc. Cary, NC, USA. All rights reserved. Printed in the United States of America. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, or otherwise, without the prior written permission of the publisher, SAS Institute Inc.

Book code E70553, course code LWSTSM41/STSM41, prepared date 26Jan2016. LWSTSM41_001

ISBN 978-1-62960-157-1

Table of Contents

Course Description

The course covers the fundamentals of modeling time series data, and focuses on the applied use of the three main model types used to analyze univariate time series: exponential smoothing, autoregressive integrated moving average with exogenous variables (ARIMAX), and unobserved components (UCM).

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 http://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 the 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 SAS Bookstore on the web at http://support.sas.com/publishing/ for a complete list of books and a convenient order form.

Prerequisites

Before attending this course, you should have an understanding of basic statistical concepts. You can gain this experience by completing the Statistics 1: Introduction to ANOVA, Regression, and Logistic Regression course.

Chapter 1 Introduction to Time Series

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

1.1 Time Series Characteristics

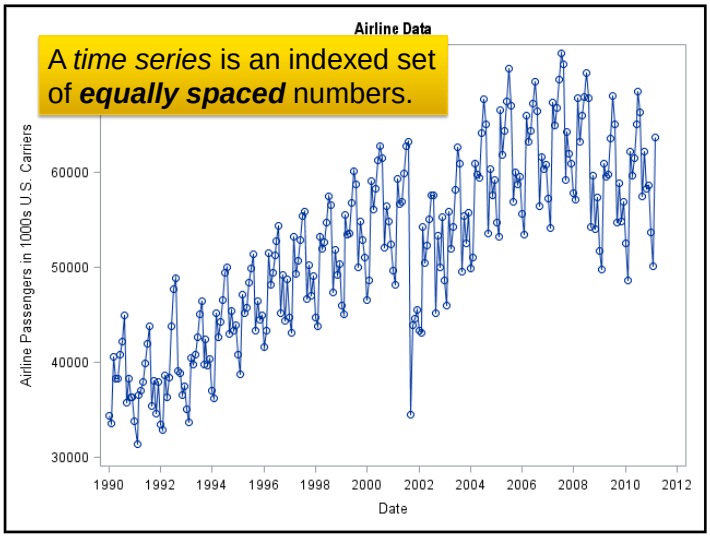
Objectives

- Define a time series.
- Describe the main ideas in time series data creation.
- Use the TIMESERIES procedure to transform transactional data into time series data (Accumulate).
- Define and explore the systematic components in a time series.

3

In business applications, time series usually start as transactional or timestamped data. An example of transactional data is a record of customer visits to a website over a period of a year. Each visit is recorded with a customer identifier and a timestamp. Transactional data are not organized with respect to a time interval. They must be made equally spaced, or indexed, before time series models can be used to quantify the systematic variation contained in it. *Accumulation* is the process of indexing or transforming transactional data into time series.

A Time Series



4

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

Time Series Creation

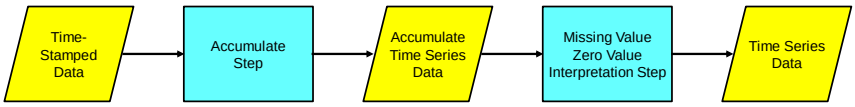
Data Accumulation:

Accumulates transactional data to a specified time interval of the data.

5

Transactional Data Preparation Steps

Before you can perform time series analysis and forecasting, the observation (raw) series must be accumulated and interpreted to form a time series. The following diagram summarizes the process of converting timestamped data to time series data:



6

The selection of the interval and the accumulation method are critical considerations in applied time series modeling. When making these choices, the modeler *creates* the data for analysis. Different accumulation choices can emphasize different systematic characteristics of the data, and impact model usefulness and precision.

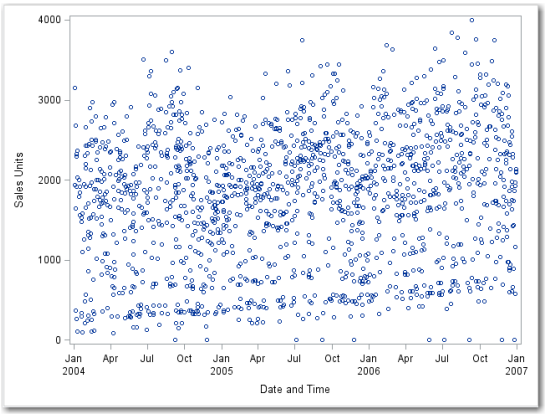
For example, consider timestamped calls into a call center. Accumulating the data to an hourly interval using a sum accumulation method gives analysts, assuming adequate volume, a good look at the hour-of-the-day cycle. It also provides direct information about what the peak and trough hours for calls are in a 24-hour cycle. This information is helpful for daily staffing decisions.

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

However, time series accumulated at an hourly interval might not be optimal for understanding and quantifying longer term trends and month-of-the-year cycles that might exist in the data. A better alternative for understanding and quantifying longer run patterns might be a monthly interval and an average accumulation method.

Transactional Data: Example

Historical Timestamped Data



No obvious patterns exist in the transactional data.

7

Transactional Analysis

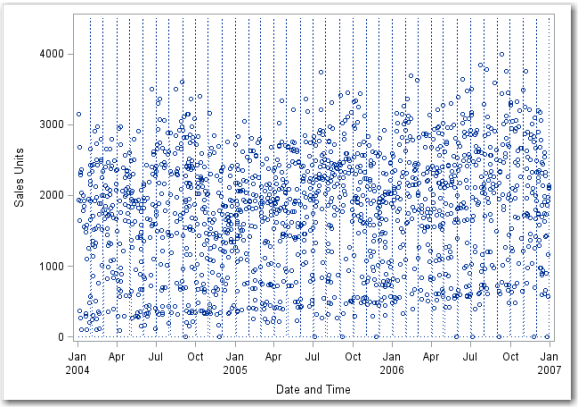
- Given a timestamped data set, each observation of the data set can be assigned an observation (raw) index, a time index, and a season index.
- Count and frequency analysis can be applied to the timestamped data set based on these indices.
- Each of these indices does not depend on the data under analysis. These indices only structure the data for subsequent analysis.

8

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

Transactional Data Time Binning

Monthly Time Bins



A monthly interval time series has one observation per interval or bin.

9

Transactional Data Time Binning

Perform time binning with the TIMESERIES procedure's INTERVAL= option. Use accumulated totals.

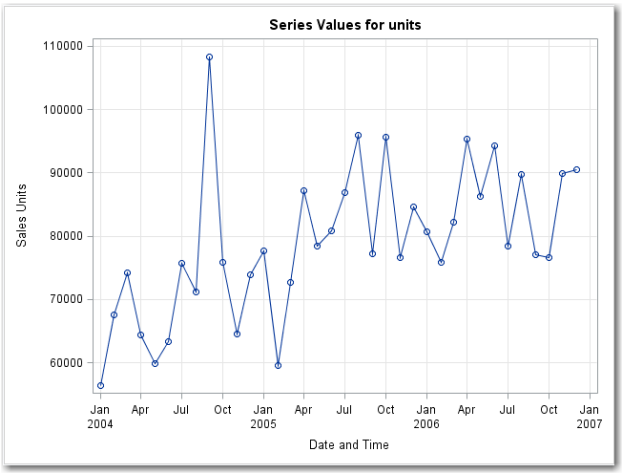
```
proc timeseries data=transactions out=outsum;
   id date interval=month
           accumulate=total;
   var units;
run;
```

10

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

Transactional Data Accumulation

Accumulated on a Monthly ***Total*** Basis



11

Transactional Data Time Binning

Perform time binning with the TIMESERIES procedure's INTERVAL= option. Use accumulated averages.

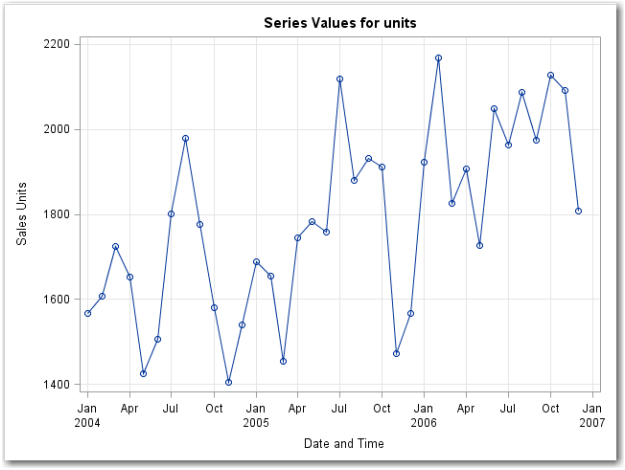
```
proc timeseries data=transactions out=outavg;
   id  interval=month
               accumulate=average;
   var units;
run;
```

12

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

Transactional Data Accumulation

Accumulated on a Monthly ***Average*** Basis



13

Transactional Data Time Binning

Perform time binning with the TIMESERIES procedure's INTERVAL= option. Use the maximum value in each interval.

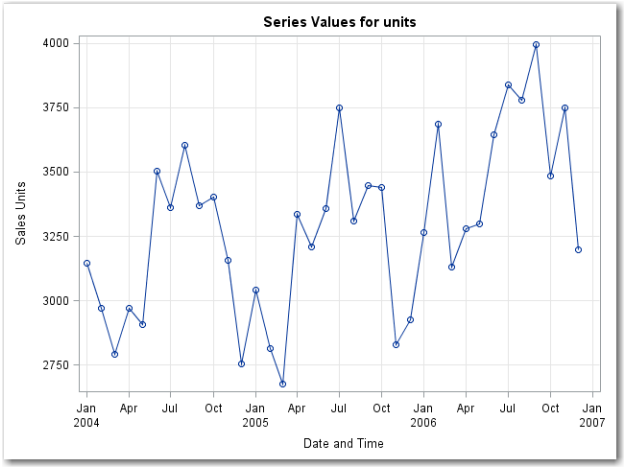
```
proc timeseries data=transactions out=outmax;
   id  interval=month
               accumulate=maximum;
   var units;
run;
```

14

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

Transactional Data Accumulation

Accumulated on a Monthly **Maximum** Basis



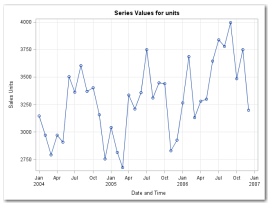
15

Transactional Data Accumulation

For this timestamped data, accumulating on a monthly ***average*** or ***maximum*** basis might be better than accumulating on a ***total*** basis.

ACCUMULATE=TOTAL

ACCUMULATE=AVERAGE



ACCUMULATE=MAXIMUM

16

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

Time Series Creation

This demonstration provides examples of creating time series through the process of accumulation. The Time Series Exploration task in SAS Studio is used to illustrate this. The transactional data set for the analysis is **STSM.CH1_DEMODAT**. The dependent variable is **units** of sales. There are two timestamps: **date** is a date variable and **dtdate** is a datetime variable. A portion of the data is shown below. The transactional observations begin on 02January2004 and end on 29December2006 inclusive.

Obs	units	dtdate	date	mon	yr
1	.	02JAN2004:08:00:00	02JAN2004	1	2004
2	1940	02JAN2004:09:00:00	02JAN2004	1	2004
3	.	02JAN2004:10:00:00	02JAN2004	1	2004
4	3147	02JAN2004:11:00:00	02JAN2004	1	2004
5	.	02JAN2004:12:00:00	02JAN2004	1	2004
6	.	02JAN2004:13:00:00	02JAN2004	1	2004
7	.	02JAN2004:14:00:00	02JAN2004	1	2004
8	.	02JAN2004:15:00:00	02JAN2004	1	2004
9	.	02JAN2004:16:00:00	02JAN2004	1	2004
10	.	02JAN2004:17:00:00	02JAN2004	1	2004

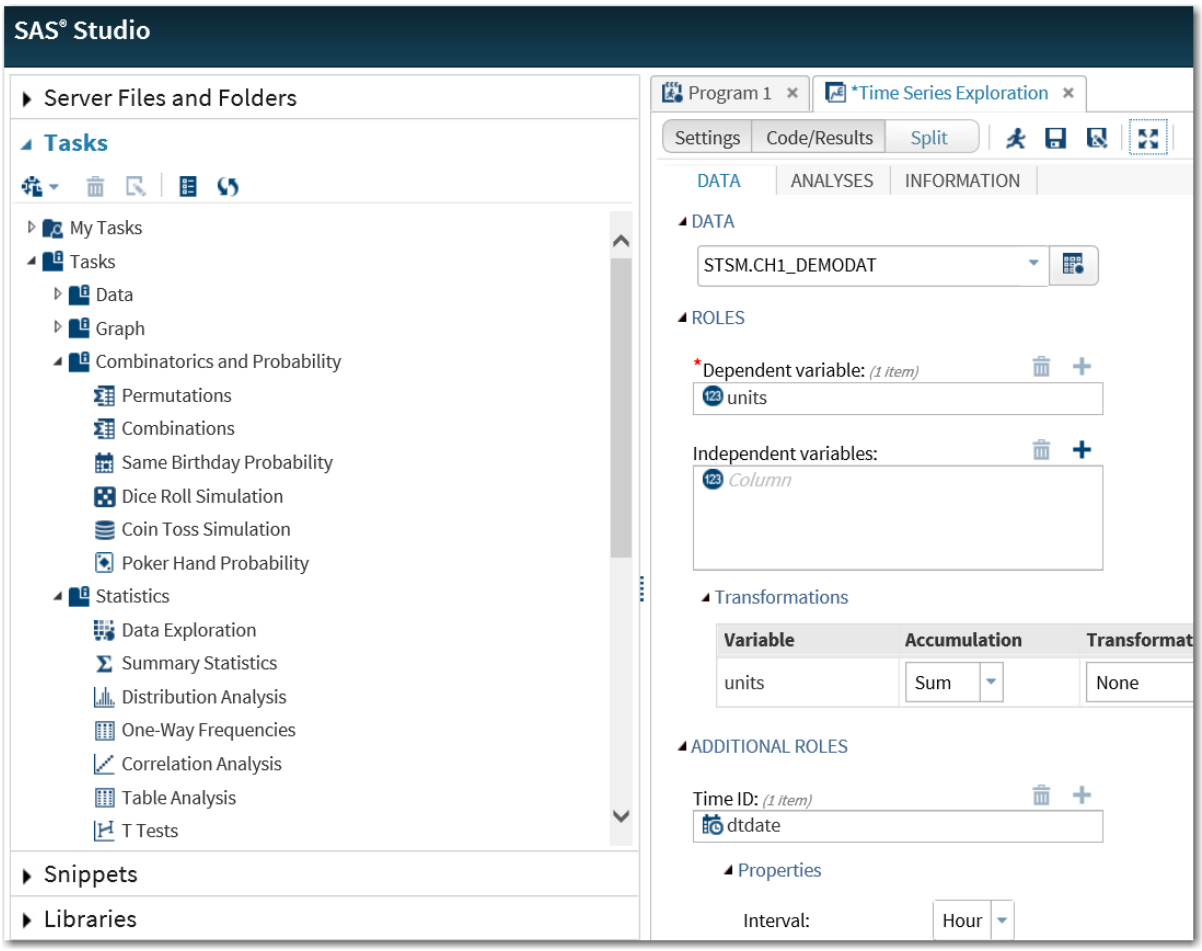
1. Log on to SAS Studio. Use the credentials that are supplied by your instructor.
2. Expand **Tasks** and then expand **Forecasting Tasks**. Double-click the **Time Series Exploration** task to initiate it.
3. Click the **Select a Table** button under the Data property to navigate to the **STSM** library. Select the **CH1_DEMODAT** data table.

These steps are summarized in the display below.

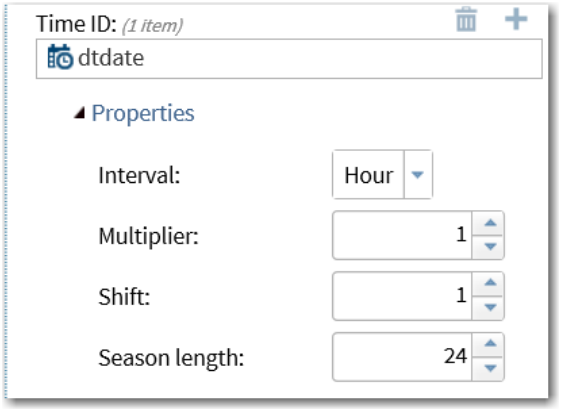
Examining the Data on an Hourly Interval

4. Assign **units** as the dependent variable. Expand the **ADDITIONAL ROLES** property, and assign the **dtdate** column as the time ID.
5. Select **Sum** for the **Accumulation** field.

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.



6. After the time ID variable is set, a one-hour interval that begins in the first hour of a 24-hour day is detected as the natural interval of the data. The options shown below can be changed to modify the detected interval.



The TIMEID procedure in SAS/ETS runs “under the hood” in SAS Studio to detect the interval of the data.

7. Click on the toolbar to maximize the view.

The TIMESERIES procedure in SAS/ETS provides the main functionality for the Time Series Exploration task. A subset of the contents of the CODE window is shown below.

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

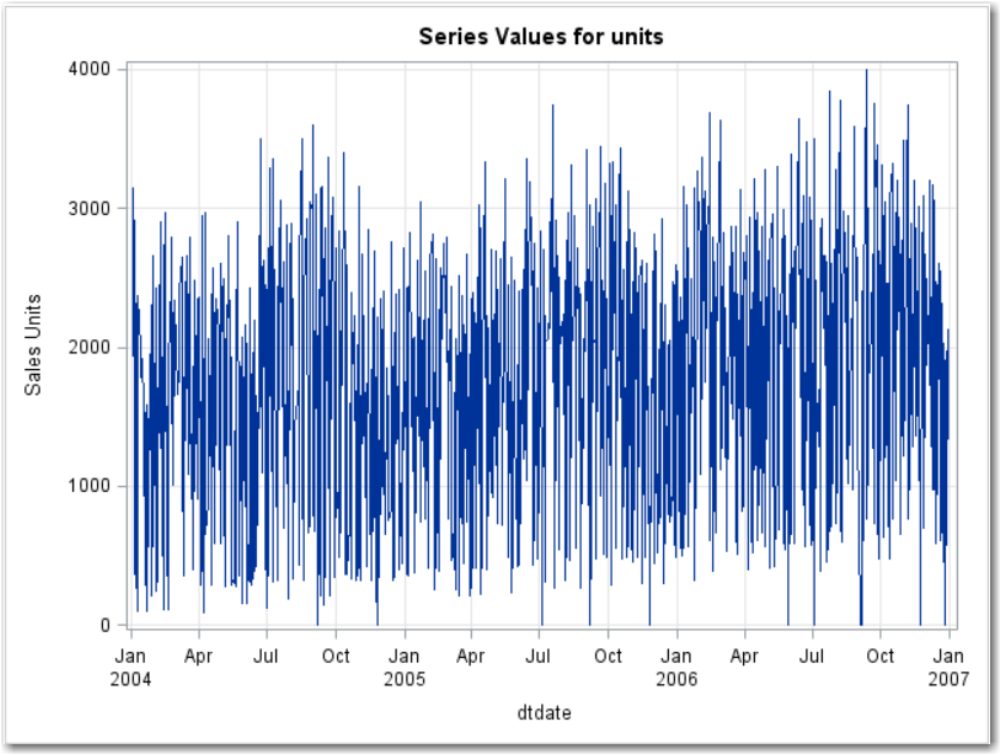
```
proc sort data=STSM.CH1_DEMODAT out=WORK.TempSorted;
   by dtdate;
run;

proc timeseries data=WORK.TempSorted seasonality=24
                plots=(series corr);
   id dtdate interval=hour;
   var units / accumulate=total transform=none dif=0 sdif=0;
run;
```

- The SORT procedure is used to create a working copy of the transactional data, and sort it from the earliest to the latest date.
- The TIMESERIES procedure is used to accumulate the transactional data to an hourly interval. The PLOTS option in the procedure statement creates the series plot and correlation panel. The ID statement defines the datetime ID variable, and the interval and accumulation options that correspond to the settings that are selected on the DATA tab.

If you produce the code by entering it directly in the editor, the code below produces the necessary output. This code assumes that the data in **stsm.ch1_demodat** are properly sorted.

```
/* accumulate to an hourly interval */
proc timeseries data=stsm.ch1_demodat out=out_totalhour
                plots=(series);
   id dtdate interval=dthour accumulate=total;
   var units;
run;
```

8. To submit the code and run the Time Series Exploration task, click the **Run** button on the toolbar. The series plot is shown below. There are many hourly intervals in a three-year time span.



Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

9. To get a better look at the hourly intervaled time series, the data is truncated and then re-plotted. Because there is currently no option to do this in the DATA properties of SAS Studio, the PROC TIMESERIES syntax is edited directly. Click the **CODE** tab and then click **Edit**.

The syntax is now contained on a new tab.

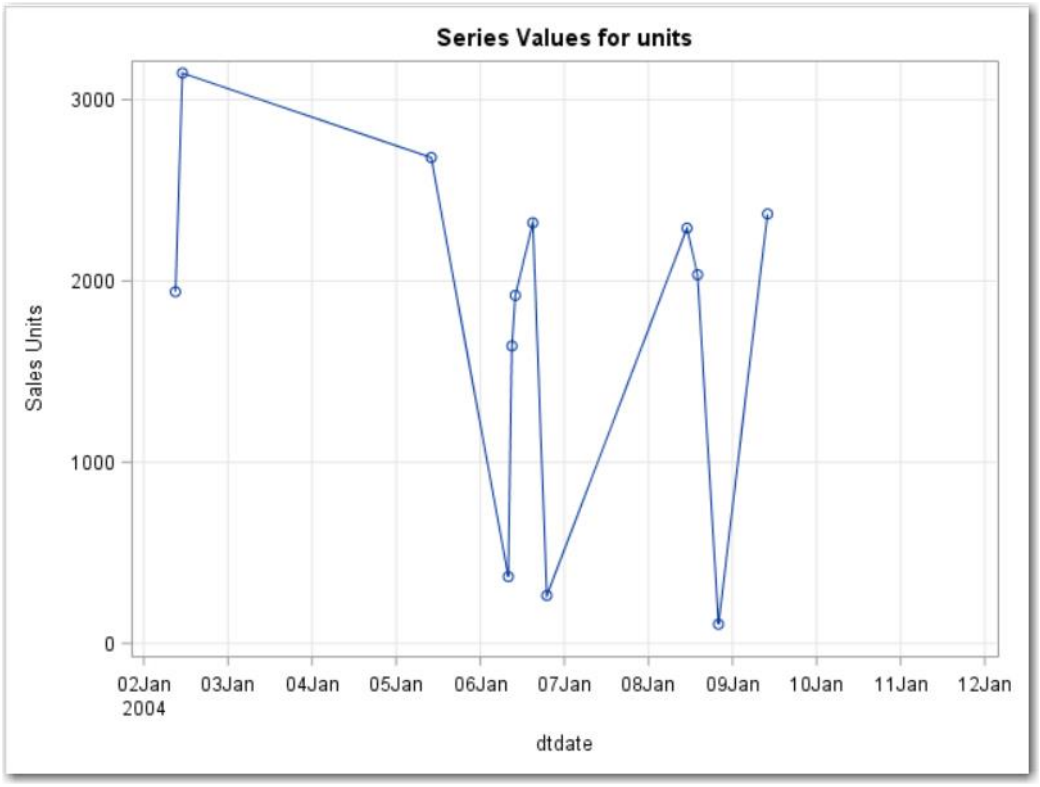
10. Add an END option to the ID statement in PROC TIMESERIES as shown below. The end value *12jan2004:00:00* truncates the data so that only the first 10 days of data are shown in the plots.

```
proc timeseries data=WORK.TempSorted seasonality=24
                plots=(series corr);
   id dtdate interval=hour end='12jan2004:00:00'dt;
   var units / accumulate=total transform=none dif=0 sdif=0;
run;
```

If you produce the code by entering it directly in the editor, the following code produces the necessary output:

```
/* subset the data using the END option */
proc timeseries data=out_totalhour plots=(series);
   id dtdate interval=dthour accumulate=total
              end='12jan2004:00:00'dt;
   var units;
run;
```

11. Click the **Run** button to submit the modified syntax.



Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

One important point, illustrated in the plot above and the table below, is that issues with missing observations cannot be discerned until after the data is accumulated. If an hourly or daily interval is to be used for forecasting, a data imputation method might be necessary.

12. Click the **OUTPUT DATA** tab to view the hourly, intervaled, time series data.

	dtdate	units
1	02JAN04:08	.
2	02JAN04:09	1940
3	02JAN04:10	.
4	02JAN04:11	3147
5	02JAN04:12	.
6	02JAN04:13	.
7	02JAN04:14	.
8	02JAN04:15	.
9	02JAN04:16	.
10	02JAN04:17	.
11	02JAN04:18	.

The SETMISSING option in the ID statement of the TIMESERIES procedure can be used to replace missing values in time series with imputed values.

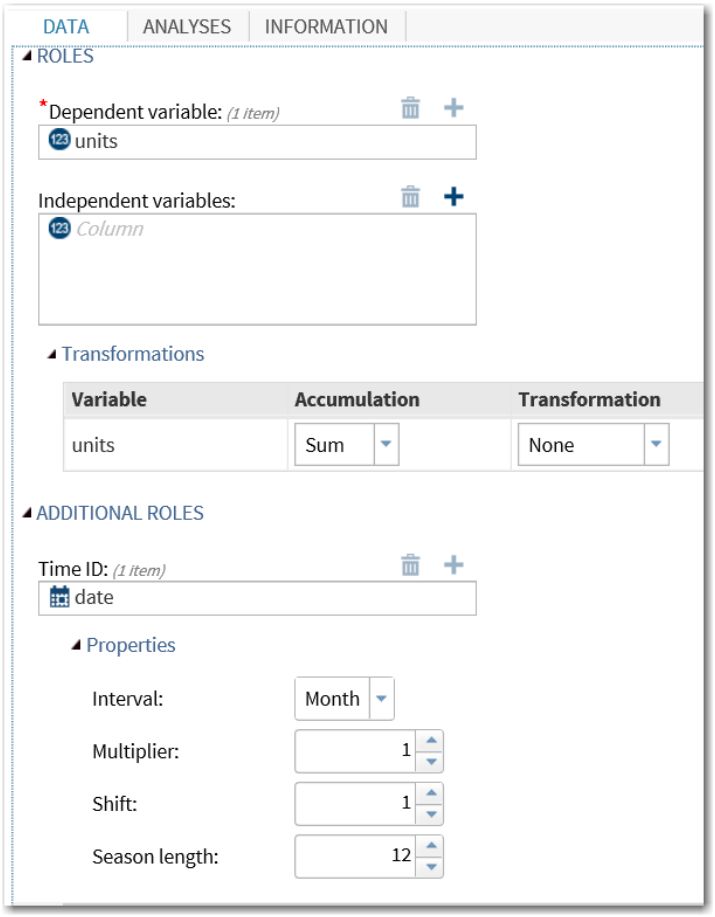
A preferred method for missing value imputation in time series is replacing missing values with their one-step-ahead forecast. The ESM procedure in SAS/ETS can be used for this. See the REPLACEMISSING option in the FORECAST statement.

Creating the Time Series on a Monthly Interval

13. Click the **Time Series Exploration 1** tab to return to the original analysis.
14. Remove the assignment of the time ID variable. Highlight **dtdate** and click the **Trash Can** icon.
15. Assign **date** as the time ID variable, and change the interval property to **Month**.
16. Expand the **Transformations** properties, and set the **Accumulation** field to **Sum**.

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

These steps are summarized in the display below.

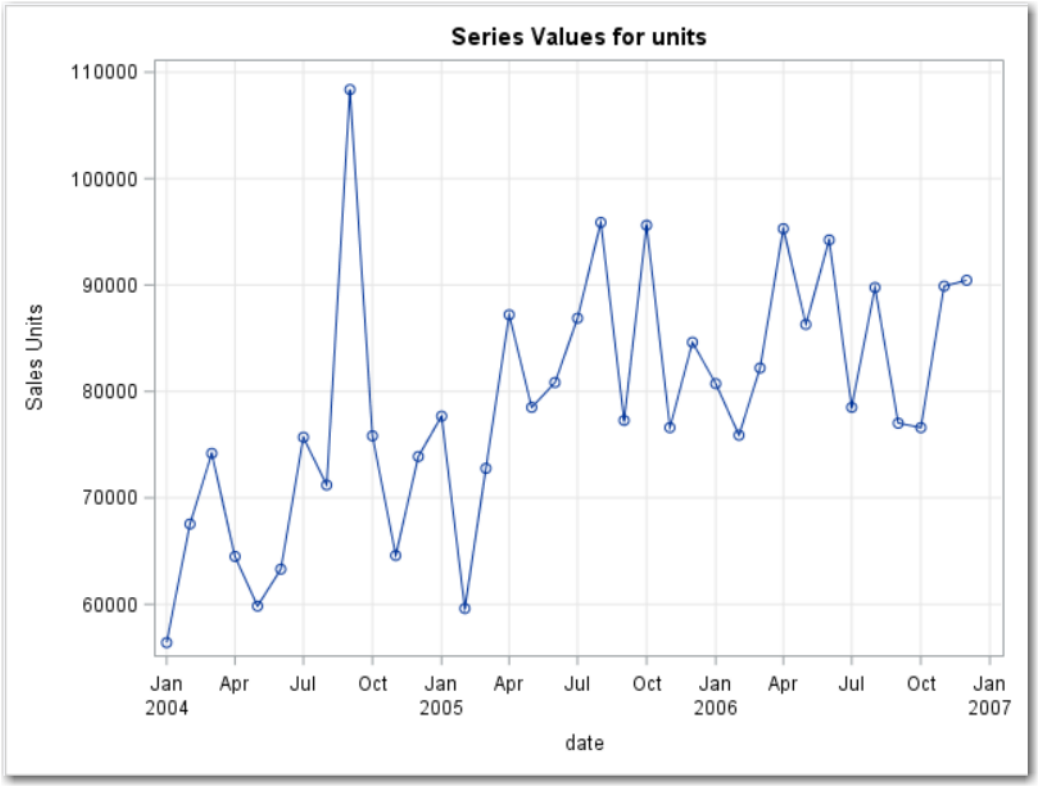


If you produce the code by entering it directly in the editor, the following code produces the necessary output:

```
/* accumulate to a monthly interval using TOTAL */
proc timeseries data=stsm.ch1_demodat out=out_totalmonth
                plots=(series);
   id date interval=month accumulate=total;
   var units;
run;
```

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

17. Click the **Run** button to submit the modified Time Series Exploration task. A plot of the monthly intervaled **units** time series is shown below.



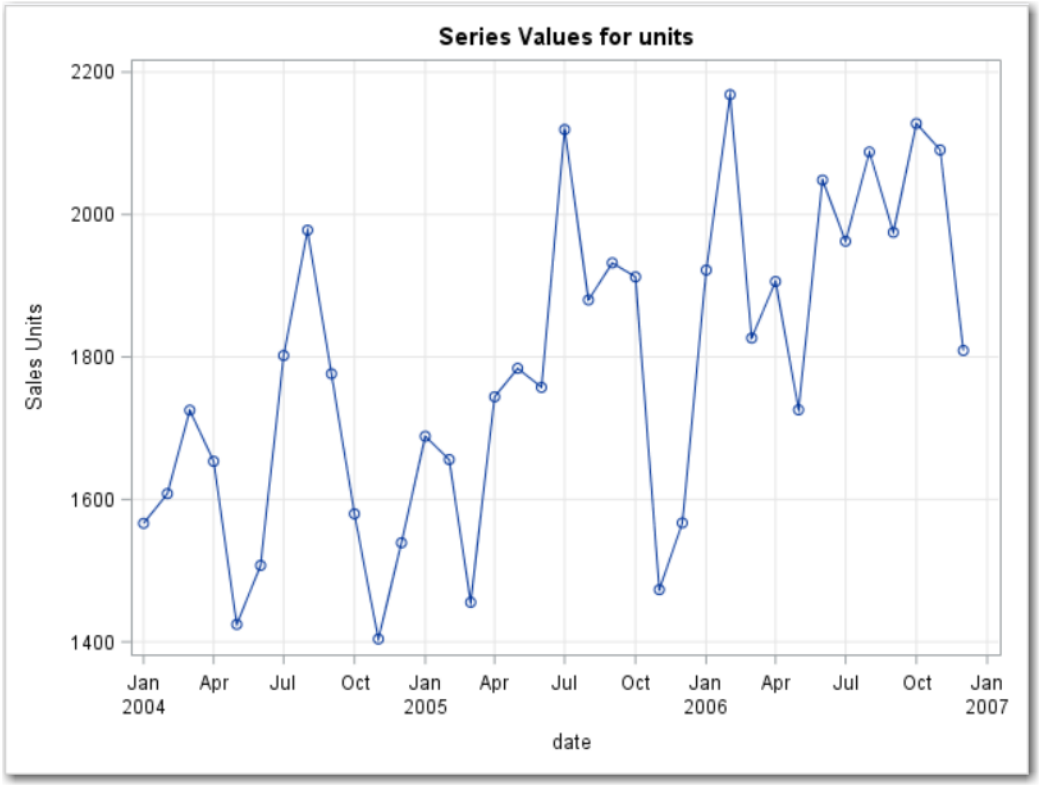
Using a Total or Sum accumulation method, the monthly intervaled data shows what might be a structural break near the middle of 2005. The spike in Sep2004 might be an anomaly that requires further investigation and consideration when you build a model to accommodate the variation in this series.

18. Change the **Accumulation** field value to **Average**.
19. If you produce the code by entering it directly in the editor, the following code produces the necessary output:

```
/* accumulate to a monthly interval using AVERAGE */
proc timeseries data=stsm.ch1_demodat out=out_totalmonth
                plots=(series);
   id date interval=month accumulate=average;
   var units;
run;
```

20. Click the **Run** button.

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.



The plot of the time series that was created using an Average accumulation method reveals a trend and a cycle in the time series data.

End of Demonstration

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

Exercises

1. Creating a New Time Series Exploration Task

The **STSM.VISITS** table is used for this exercise. It contains approximately three years of transactional data on visits to a 24-hour, emergency clinic. The dependent variable is **visits**, and the time ID variable is **date**. Create a new Time Series Exploration task and choose appropriate interval and accumulation methods to answer the questions below.

Be sure to assign an output data set. Use an option in the procedure statement.

Hint: **out=xxx**

a. What is the average number of visits per year for each of the three years in the data?

b. What interval has the highest monthly total number of visits?

c. Are there any day intervals with zero visits?

End of Exercises

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

1.2 Time Series Components

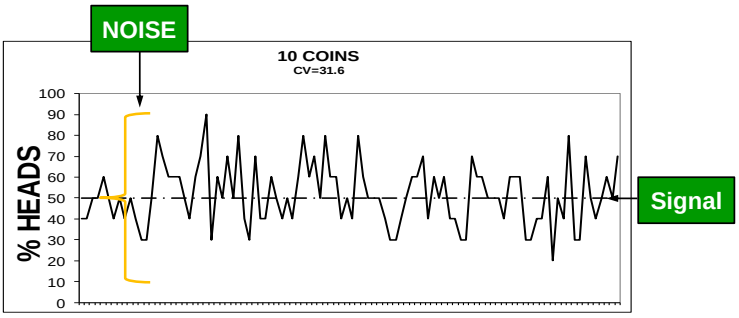
Objectives

- Describe the decomposition of time series variation.
- List the main components of systematic variation or signal that exist in time series.
- Perform a preliminary identification of a time series.

21

Variation in Time Series Data: Two Main Parts

- signal
- noise



22

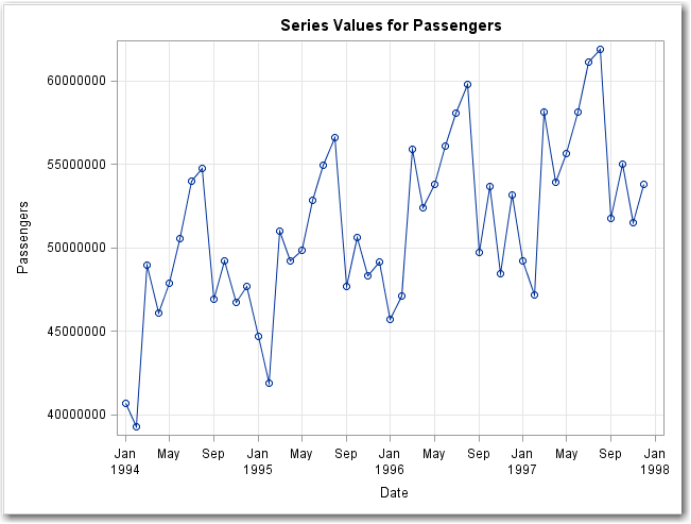
Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

Signal Components

- level
- seasonality
- trend
- irregular
- exogenous
  (also known as *explanatory variable effects*)
- cycle

23

The Airline Data

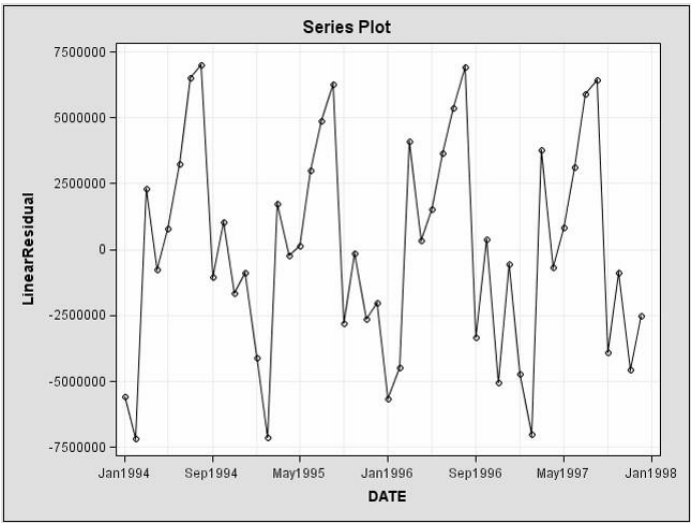


24

The airline data, **DOTAIR9497**, contain many of the signal components listed above. The data set contains a monthly intervaled time series of passengers who flew on commercial aircraft in the United States between January 1994 and December 1997. The slides below show a decomposition of the series into seasonal, trend, and irregular components.

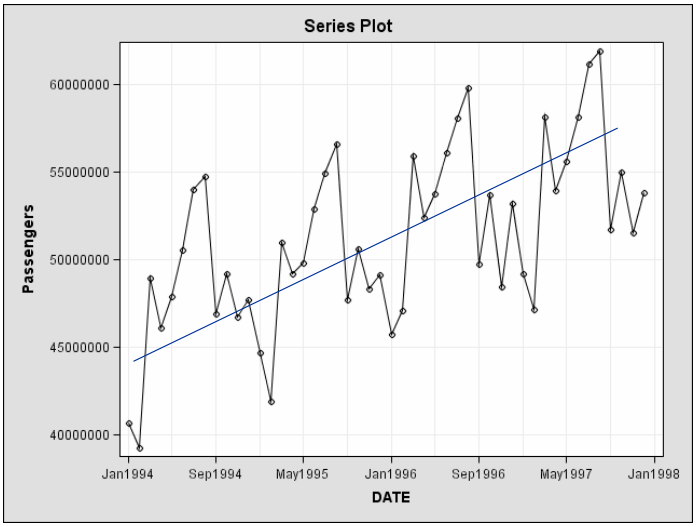
Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

Signal Components: Seasonality



25

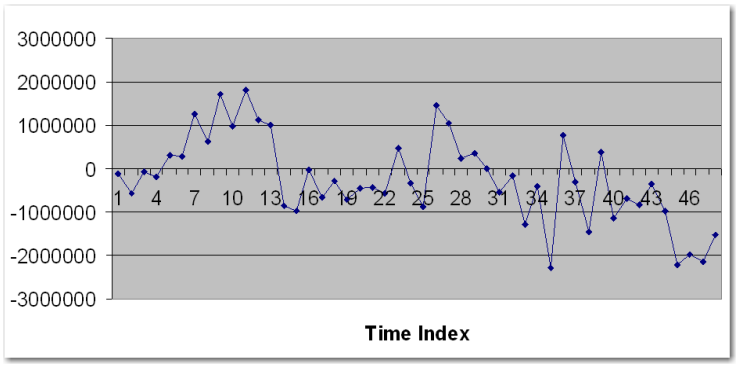
Signal Components: Trend



26

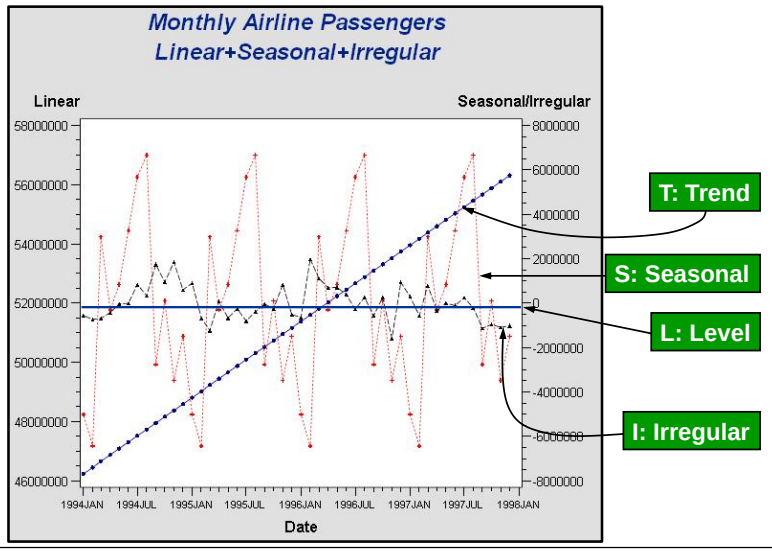
Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

Signal Components: Irregular



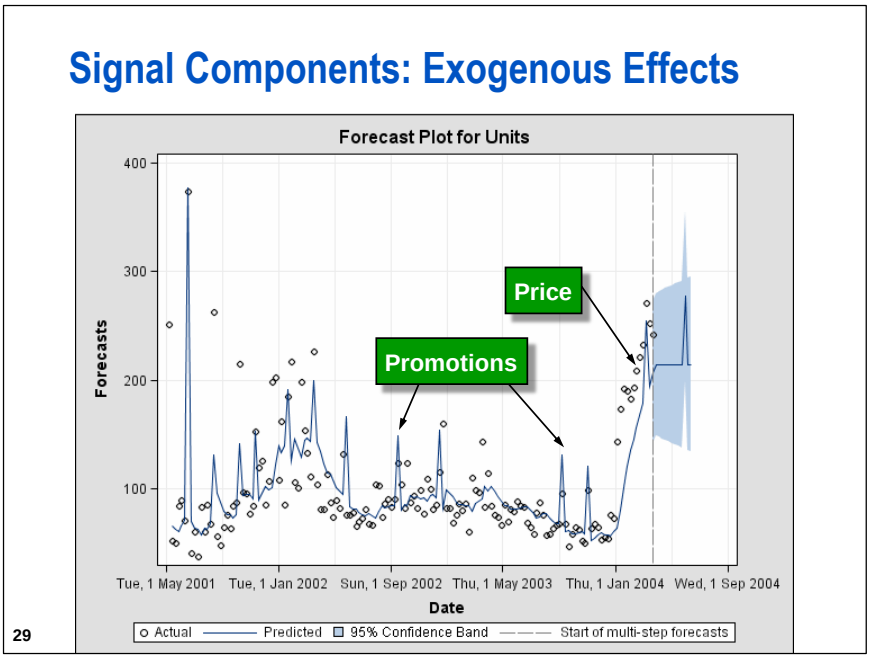
27

Component Decomposition



28

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.



The Retail time series represents weekly unit sales for an item. The series shows evidence of promotion and price effects.

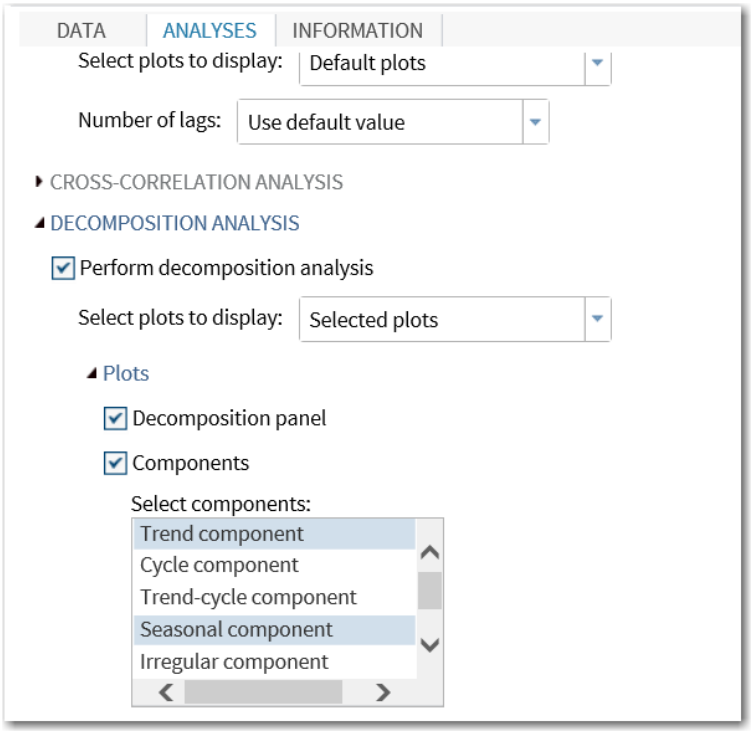
Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

Time Series Identification

This demonstration begins where the previous demonstration ended. The monthly intervaled time series that was accumulated using the Average method is decomposed to assess the presence of trend, seasonal, and other components of systematic variation.

1. Click the **ANALYSES** tab in the Time Series Exploration task that you created in the previous demonstration.
2. Expand the **DECOMPOSITION ANALYSIS** properties.
3. Select the **Perform decomposition analysis** check box, and change the **Select plots to display** field to **Selected plots**.
4. Select the **Components** check box, and highlight the **Trend** and **Seasonal** component plots.

   These steps are summarized below.



5. Use SAS/ETS program code.

   Additional syntax includes the following decomposition plot options:

   - TC = Trend Component
   - SC = Seasonal Component
   - IC = Irregular Component
   - CORR plots correlation functions

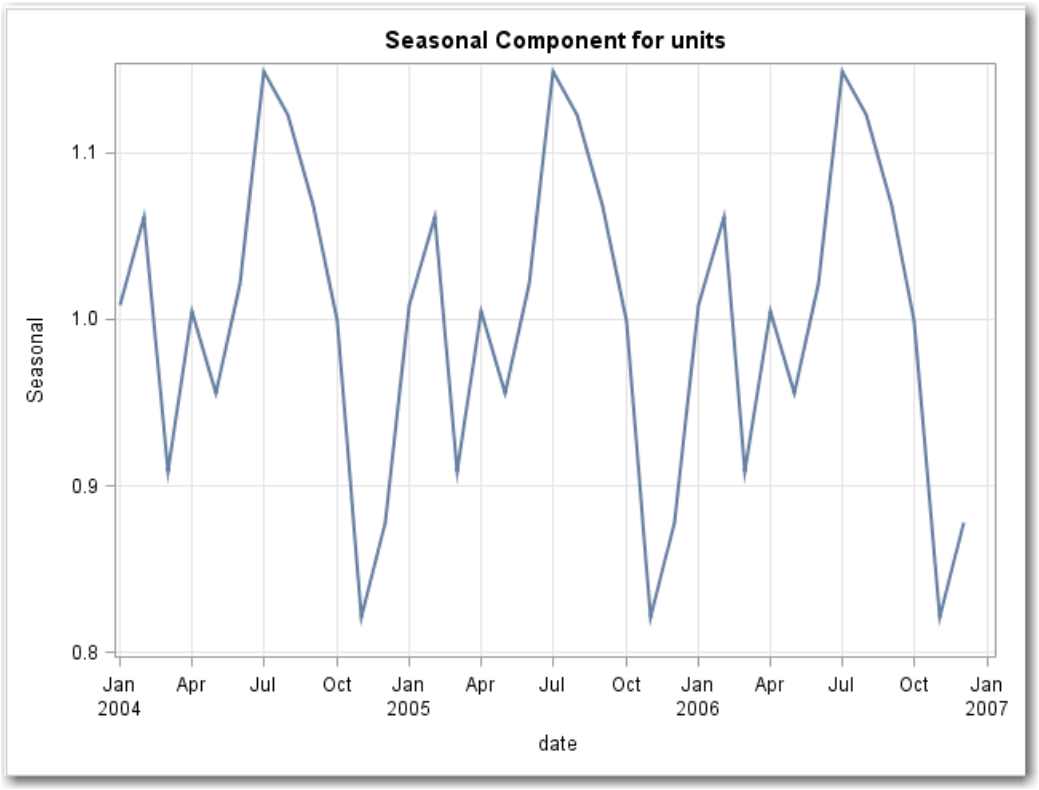
   The **decomp_total** data set contains the decomposition statistics and the *de-seasonalized series*. (The estimated seasonal component is subtracted from the original series to create the output series.)

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

```
proc timeseries data=stsm.ch1_demodat out=out_totalmonth
                outdecomp=decomp_total
                print=(descstats seasons decomp)
                plot=(series corr tc sc ic);
   id date interval=month accumulate=average;
   var units;
run;
```

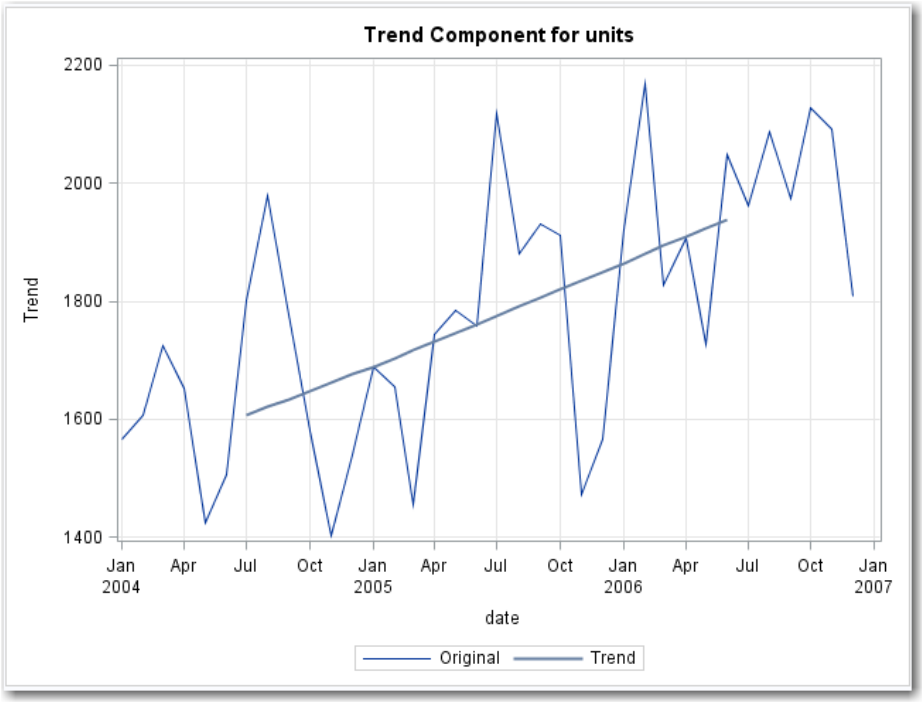
6. Click the **Run** button.

   The seasonality plot reveals a fairly strong seasonal cycle in the data that was accumulated to a monthly average. The seasonal peak month is July. Sales of **units** in July average more than 10% above average, annual sales. November is the seasonal low month. Sales in November average almost 20% below the annual average.

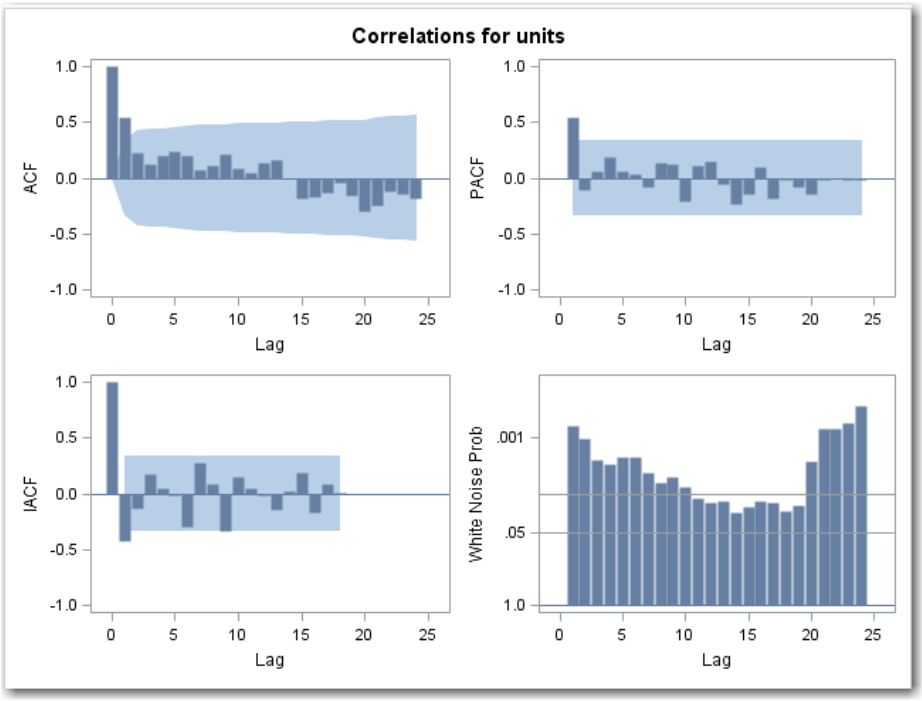


   The trend plot is based on a centered, moving-average representation of the series. A fairly strong and linear increase in average sales per month is depicted.

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.



The correlation panel shows that the average **units** series is not white noise. The additional diagnostics provide information that can be used to select autoregressive and moving average orders for ARIMA models. (More information about these is provided in the "ARIMAX Models" chapter.)



A portion of the **decomp_total** data set is shown below. There are 12 unique values of the seasonal components. The peak month, July, is approximately 15% above the annual average. The Seasonally Adjusted series is derived by dividing the **units** (Original) series by the Seasonal Component series.

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

date	Seasonal Index	Original Series	Trend-Cycle Component	Seasonal Component	Irregular Component	Seasonally Adjusted Series
JAN2004	1	1566.44	.	1.00888	.	1552.65
FEB2004	2	1608.07	.	1.06163	.	1514.72
MAR2004	3	1725.09	.	0.90875	.	1898.30
APR2004	4	1653.51	.	1.00518	.	1644.99
MAY2004	5	1424.55	.	0.95608	.	1489.98
JUN2004	6	1507.31	.	1.02105	.	1476.23
JUL2004	7	1802.05	1635.47	1.14892	0.95903	1568.47
AUG2004	8	1978.14	1642.56	1.12292	1.07247	1761.60
SEP2004	9	1776.46	1633.32	1.06845	1.01796	1662.64
OCT2004	10	1579.73	1625.86	0.99908	0.97252	1581.19
NOV2004	11	1404.00	1644.62	0.82108	1.03972	1709.95
DEC2004	12	1539.23	1670.01	0.87796	1.04980	1753.18
JAN2005	1	1688.65	1693.65	1.00888	0.98827	1673.79
FEB2005	2	1655.89	1702.78	1.06163	0.91601	1559.76
MAR2005	3	1455.50	1705.19	0.90875	0.93928	1601.64

End of Demonstration

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

Exercises

2. Using the TIMESERIES Procedure and Creating Appropriate Decomposition Plots

This exercise uses the **STSM.VIOLENTCRIME** table. The dependent variable, **MurdersTX**, is the number of murders in Texas per month between JAN1989 and DEC1997. The time ID variable is **date**.

Create appropriate decomposition plots to answer the following questions:

a. Does the data have a seasonal cycle?

b. Is there a trend component in the data? If so, is it linear?

c. Assume that you are one of the Texas governors who were elected in the years 1991 or 1995. Is it reasonable for you to claim that your progressive, yet no-nonsense policies diminished the number of homicides in Texas during your term?

End of Exercises

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

1.3 Time Series Models

Objectives

- List the three families of time series models that are illustrated in this course.
- Describe the main features of each model family.
- Give examples of using an appropriate model from each family to fit the **DOTAIR9497** data.
- Discuss which model family is preferred for different types of time series analyses.

34

Necessary Conditions for Good Forecasts

- The identified signal continues into the future.
- Forecasting model complexity should be adequate to capture signal components.
- Forecasting models should not be overly complex.
- The best forecasting model is the one that captures and extrapolates the most signal, and that also ignores the noise.

35

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

Time Series Models in This Course

- exponential smoothing (ESM)
- autoregressive integrated moving average with exogenous variables (ARIMAX)
- unobserved components (UCM)

36

Exponential Smoothing Models

Exponential Smoothing Premise

- Weighted averages of past values can produce good forecasts of the future.
- The weights should emphasize the most recent data.
- Forecasting should require only a few parameters.
- Forecast equations should be simple and easy to implement.

37

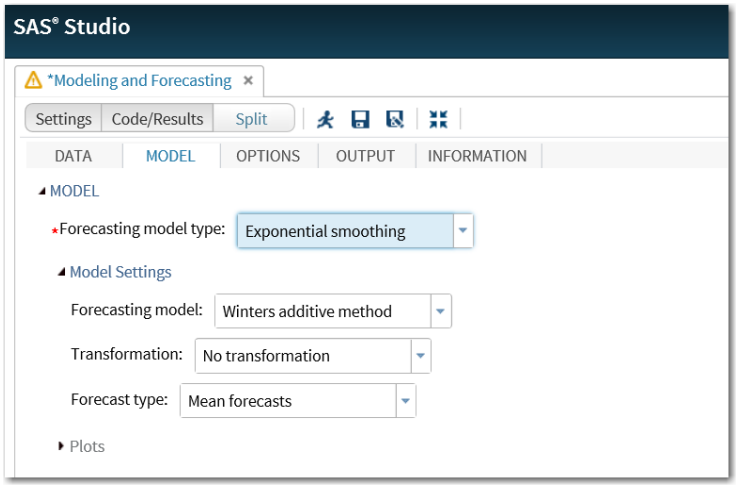
Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

ESM Parameters and Keywords

ESM	Parameters	Option Name
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`

38

Exponential Smoothing Model Implementation



39

Two of the seven exponential smoothing models accommodate trend, seasonal, and irregular variation (winters additive and winters multiplicative). The Modeling and Forecasting task in SAS Studio is used above to specify a winters additive method on the **DOTAIR9497** data.

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

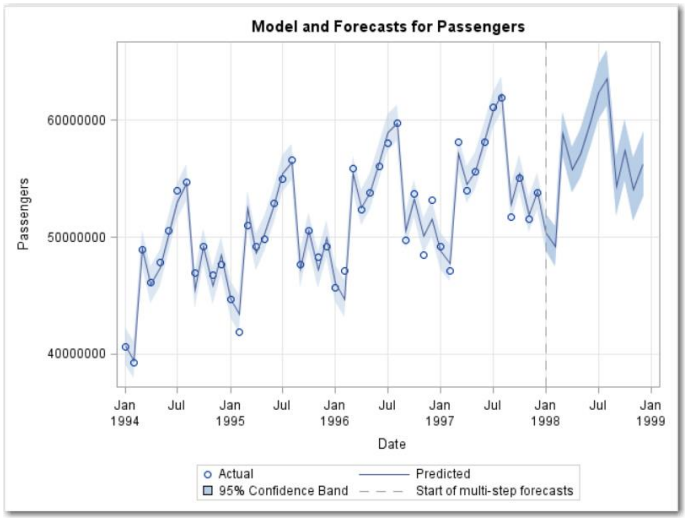
Exponential Smoothing Model Implementation

```
proc sort data=STSM.DOTAIR9497 out=WORK.TempSorted;
        by DATE;
run;

/*****************************************************************************************/
/* Exponential Smoothing                                                                 */
proc esm data=WORK.TempSorted back=0 lead=12 seasonality=12 plot=(corr errors
                modelforecasts);
        /* id statement */
        id DATE interval=month;
        forecast Passengers / alpha=0.05 model=addwinters transform=none;
run;
```

40

The ESM procedure (SAS/ETS) syntax, generated by SAS Studio, fits the winters additive (**addwinters**) specification and forecasts 12 months into the future.

Exponential Smoothing Model Forecasts



41

The selected ESM model seems to do a good job of capturing and extrapolating the trend and seasonal components of the data.

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

ARIMAX Models

Box-Jenkins ARIMAX Models

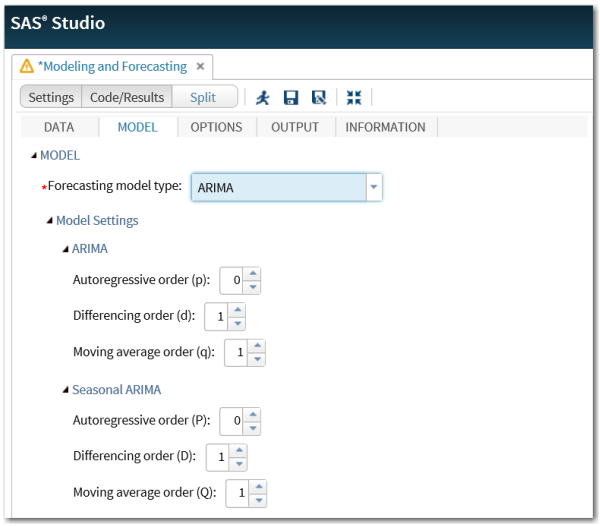
ARIMAX: AutoRegressive Integrated Moving Average with eXogenous variables

- 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).
- I: Integrated ⇒ Differencing provides stochastic trend and seasonal components, so forecasting requires *integration* (undifferencing).
- X: Exogenous ⇒ Time series is influenced by external factors.

42

Integration (the "I" in ARIMA) corresponds to the accommodation of nonstationary variation in an ARIMA model. Trend and seasonal components are examples of nonstationary variation. This is because, if a series has trend or seasonal components, its mean is a function of time. The mean of a stationary series is well defined, and is not a function of time. (A later chapter provides examples of stationary series.)

Box-Jenkins ARIMAX Implementation



43

An appropriate ARIMA model for the **DOTAIR9497** data set is the classic Box-Jenkins Airline model for series G, shown above. This model accommodates trend and seasonality with a first and seasonal (12) span differences. Differencing orders are offset by moving average orders at the same lags.

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

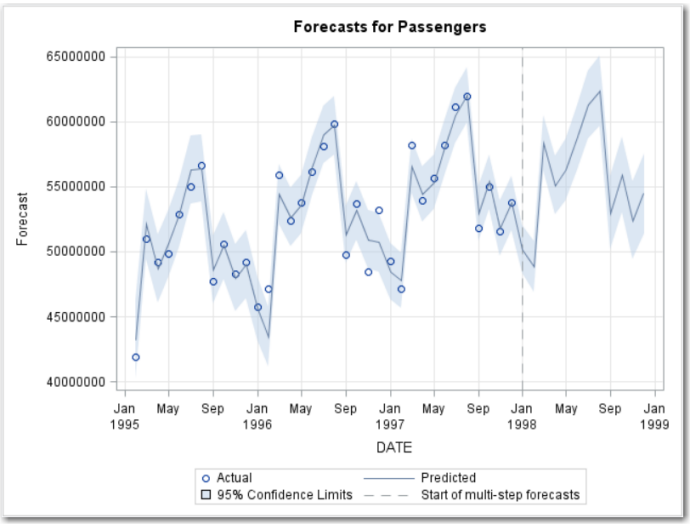
Box-Jenkins ARIMAX Implementation

```
proc sort data=STSM.DOTAIR9497 out=WORK.TempSorted;
        by DATE;
run;

/*****************************************************************************************/
/* ARIMA or ARIMAX                                                                       */
proc arima data=WORK.TempSorted plots
        (only)=(series(corr crosscorr) residual(corr normal) forecast(forecastonly));
        identify var=Passengers(1 12);
        estimate q=(1) (12) method=ML;
        forecast lead=12 back=0 alpha=0.05 id=DATE interval=month;
        outlier;
quit;
```

44

In the generated, ARIMA procedure syntax, differencing orders are specified in the IDENTIFY statement. The Q option in the ESTIMATE statement specifies moving average orders. (More details about the ARIMA procedure syntax are provided in a later chapter.)

Box-Jenkins ARIMAX Forecasts



45

The ARIMA specification fitted above does a good job of capturing and extrapolating the trend, seasonal, and irregular components in the Passengers series.

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

Unobserved Components Models

Unobserved Components Models (UCMs)

- Also known as *structural time series models*
- Decompose time series into components:
  - trend
  - season
  - cycle
  - irregular
  - regressors
- General form:

$$Y_t = \text{Trend} + \text{Season} + \text{Cycle} + \text{Regressors}$$

46

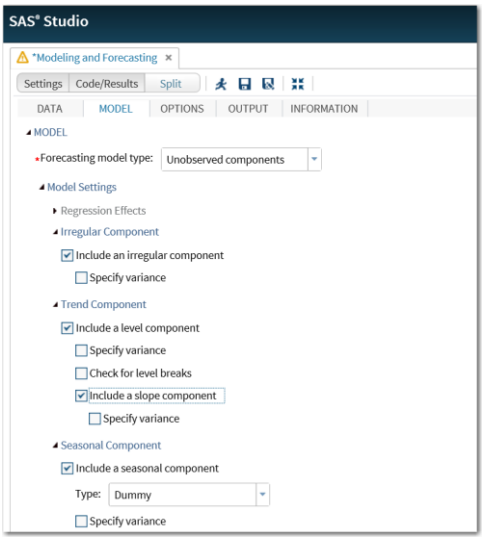
UCMs

- Each component captures some important feature of the series dynamics.
- Components in the model have their own models.
- Each component has its own source of error.
- The coefficients for trend, season, and cycle are dynamic.
- The coefficients are testable.
- Each component has its own forecasts.

47

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

UCM Implementation



48

A UCM model that accommodates the systematic components in the **DOTAIR9497** data is shown above. To create an appropriate UCM, a statement that corresponds to each component of hypothesized, systematic variation is specified.

The LEVEL and SLOPE components (statements) combine to fit the trend in the **DOTAIR9497** data.

UCM Implementation

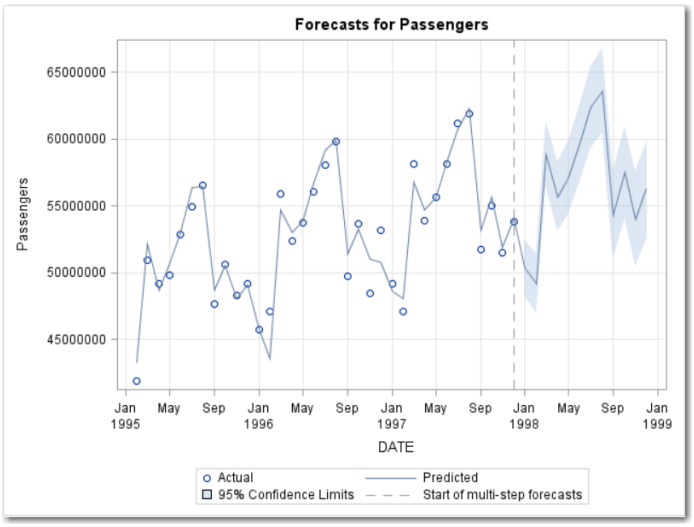
```
proc sort data=STSM.DOTAIR9497 out=WORK.TempSorted;
        by DATE;
run;

/*****************************************************************************************/
/* Unobserved Components                                                                 */
proc ucm data=WORK.TempSorted;
        id DATE interval=month;
        model Passengers;
        irregular;
        level;
        slope;
        season length=12 type=dummy;
        forecast lead=12 back=0 alpha=0.05;
        outlier;
run;
```

49

A statement for each of the components of variation in the **DOTAIR9497** data is seen in the generated syntax above.

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

UCM Forecasts



50

The UCM specification does a good job of capturing and extrapolating the systematic variation in the **DOTAIR9497** data.

Choosing the Right Model for the Job

Usability

Best to worst:

1. ESM
2. UCM
3. ARIMAX

51

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

Complexity

Least to Most:

1. ESM
2. ARIMAX
3. UCM

52

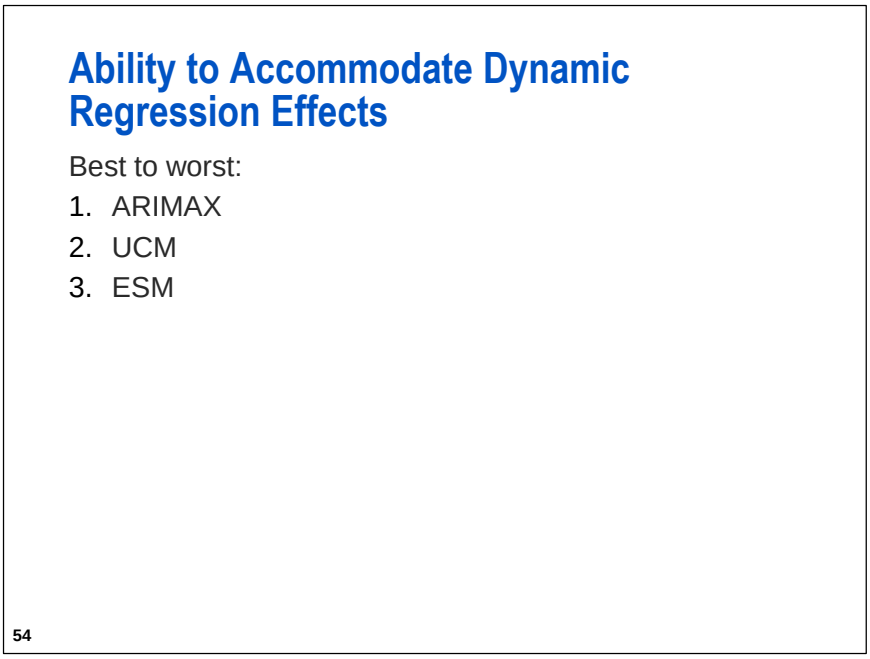
Robustness

Best to worst:

1. ESM
2. ARIMAX
3. UCM

53

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.



Idea Exchange

List the types of forecasting models that you used in your analyses.

Give an example of an analysis that would be well suited to each of the following families of models:

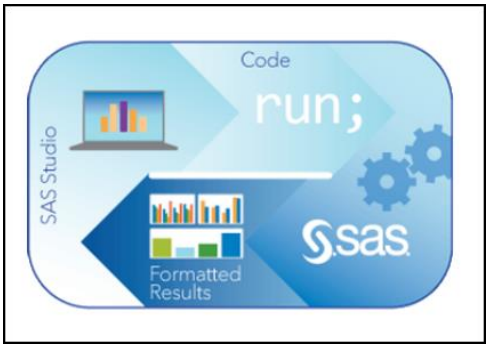
- ESM
- ARIMAX
- UCM

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

1.4 SAS Studio Introduction

SAS Studio

SAS Studio is the new browser-based SAS programming environment.



http://sww.sas.com/gobot/SASStudioTutorial

57

Interactive Mode

Some SAS procedures, such as PROC ARIMA, are interactive. That means that they remain active until you submit a QUIT statement, or until you submit a new PROC or DATA step.

In SAS Studio, you can use the Code Editor to run these procedures, as well as other SAS procedures, in interactive mode.

By default, SAS Studio does not run in interactive mode.
This icon in SAS Studio toggles interactive mode on and off.

58

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

Considerations for Running in Interactive Mode

- Interactive mode starts a new SAS session.
- Librefs and macro variables must be defined for each new SAS session.

SAS Studio Documentation:
http://sww.sas.com/gobot/SASStudioDoc

59

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

1.5 Solutions to the Exercises

1. Creating a New Time Series Exploration Task

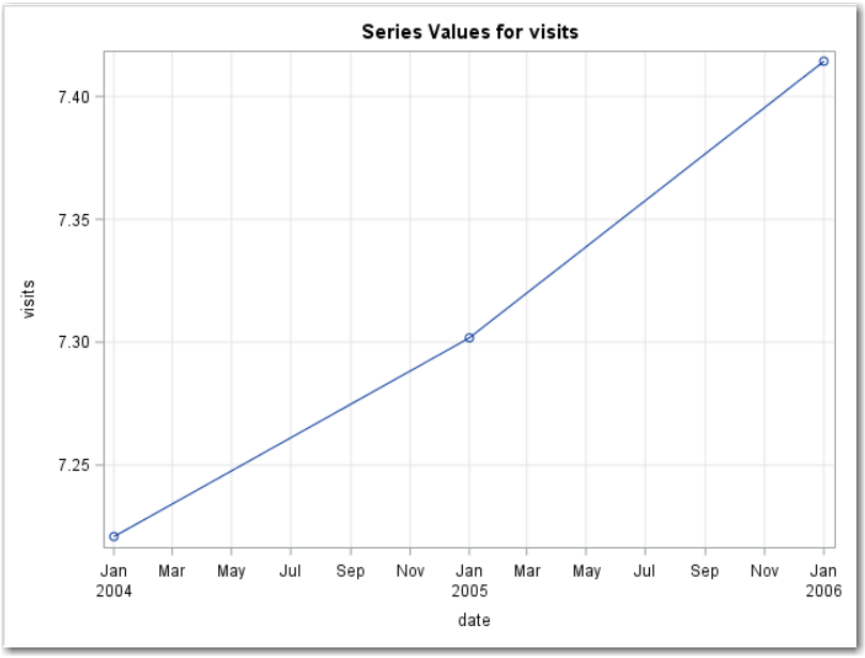
The **STSM.VISITS** table is used for this exercise. It contains approximately three years of transactional data on visits to a 24-hour, emergency clinic. The dependent variable is **visits**, and the time ID variable is **date**. Create a new Time Series Exploration task and choose appropriate interval and accumulation methods to answer the following questions:

a. What is the average number of visits per year for each of the three years in the data?

Assign **visits** as the dependent variable, **Average** as the Accumulation method, **date** as the time ID, and **Year** as the interval in the new Time Series Exploration task. The resulting plot provides the averages.

Alternatively, write the SAS/ETS code directly.

```
proc timeseries data=STSM.VISITS plots=(series) out=yr_avg;
   id date interval=year accumulate=average;
   var visits;
run;
```



The average numbers of visits are approximately 7.2 in 2004, 7.3 in 2005, and 7.4 in 2006.

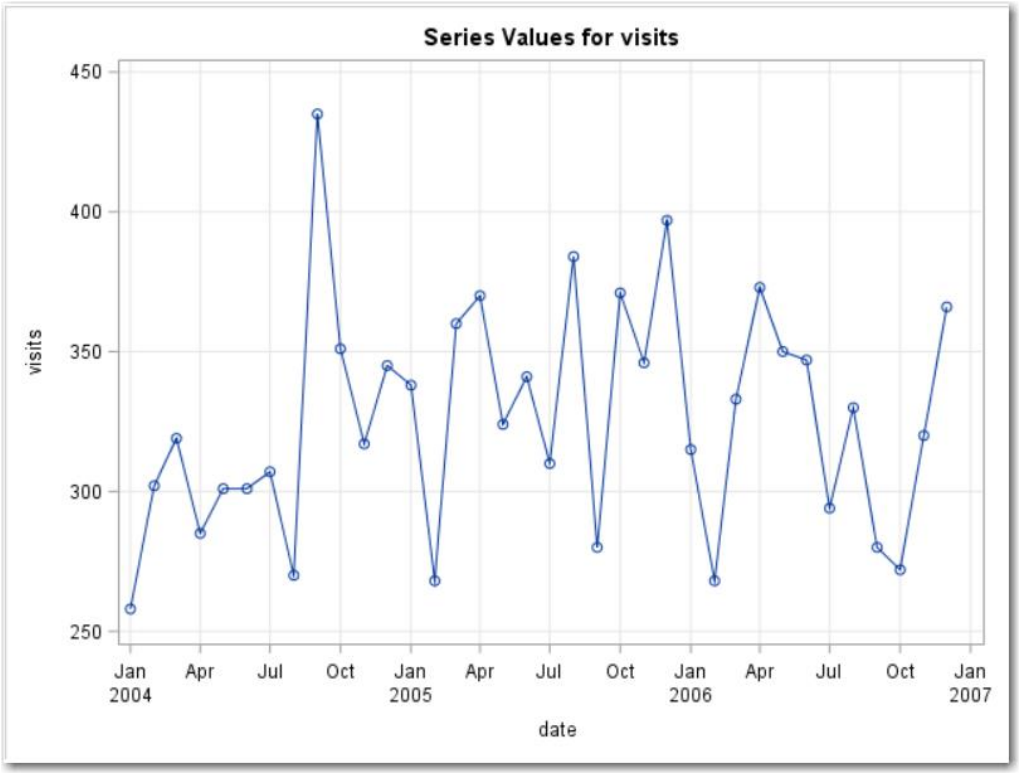
b. What interval has the highest monthly total number of visits?

In the Time Series Exploration task created above, change Accumulation to **Sum** and Interval to **Month**.

Alternatively, write the SAS/ETS code directly.

```
proc timeseries data=STSM.VISITS plots=(series) out=monthsum;
   id date interval=month accumulate=sum;
   var visits;
run;
```

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.



September 2004 has the highest total number of visits.

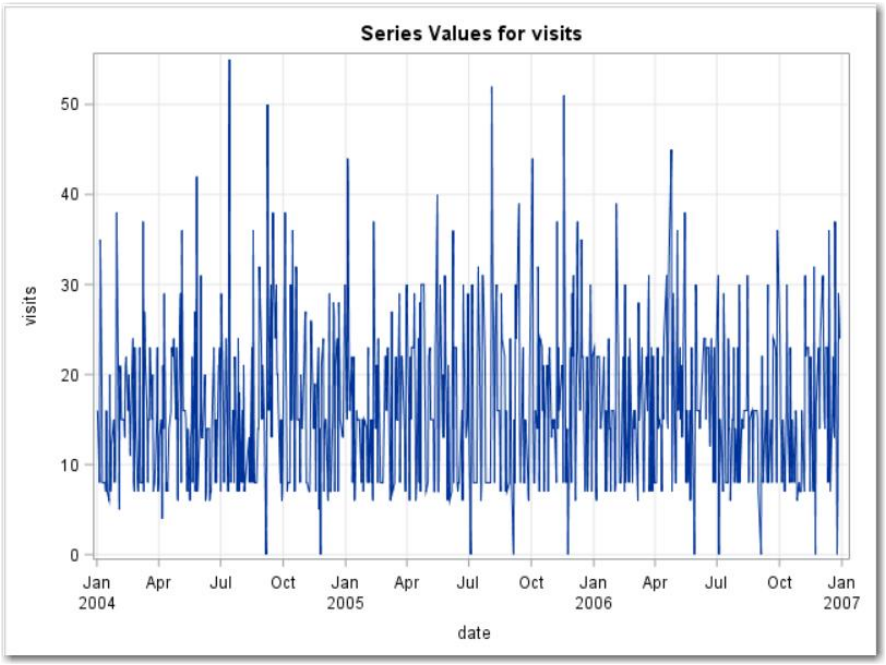
c. Are there any day intervals with zero visits?

Change the interval to **day** for the task that was modified in the exercise above.

Alternatively, write the SAS/ETS code directly.

```
proc timeseries data=STSM.VISITS plots=(series) out=daysum;
   id date interval=day accumulate=sum;
   var visits;
run;
```

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.



The answer is yes.

2. Using the TIMESERIES Procedure and Creating Appropriate Decomposition Plots

This exercise uses the **STSM.VIOLENTCRIME** table. The dependent variable, **MurdersTX**, is the number of murders in Texas per month between JAN1989 and DEC1997. The time ID variable is **date**.

Create a new Time Series Exploration task, and then create the appropriate analyses to answer the questions below.

To answer these questions, do the following:

On the DATA tab, do the following:

- Select **STSM.VIOLENTCRIME** for the data.
- Set the dependent variable to **MurdersTX**.
- Set **date** as the time ID.

The detected interval is appropriate.

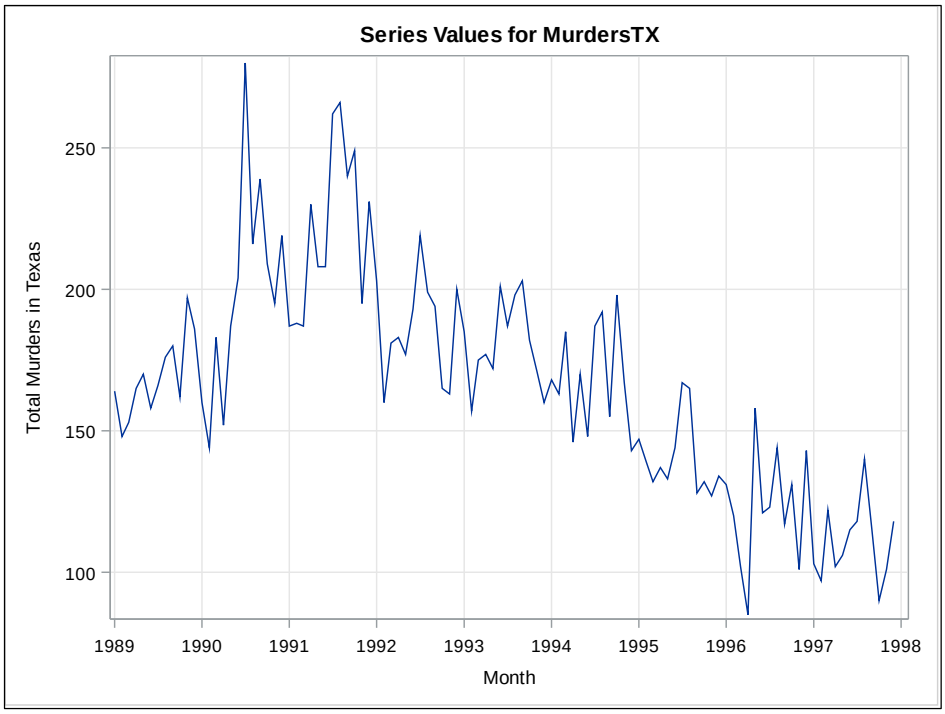
On the ANALYSES tab, do the following:

- Expand **DECOMPOSITION ANALYSES**. Select the **Perform decomposition analysis** check box.
- Change the Select Plots to Display option to **Selected plots**. Select the **Components** check box.
- Select the **Trend** and **Seasonal** components.

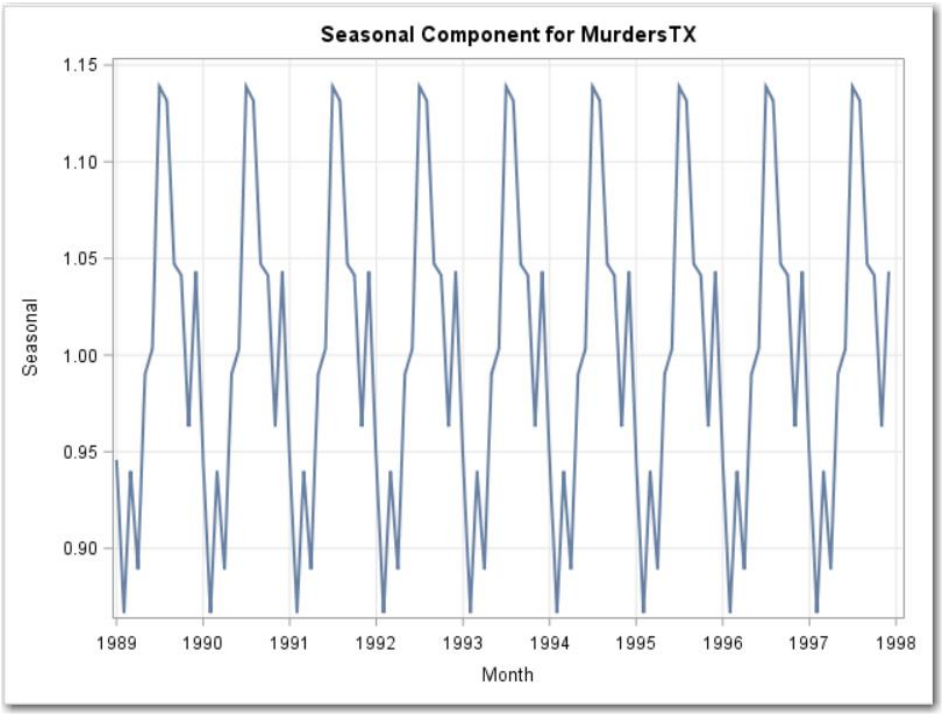
Alternatively, write the SAS/ETS code directly.

```
proc timeseries data=STSM.VIOLENTCRIME plots=(series tc sc);
   id Date interval=month;
   var MurdersTX;
   decomp tc sc / mode=multoradd;
run;
```

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

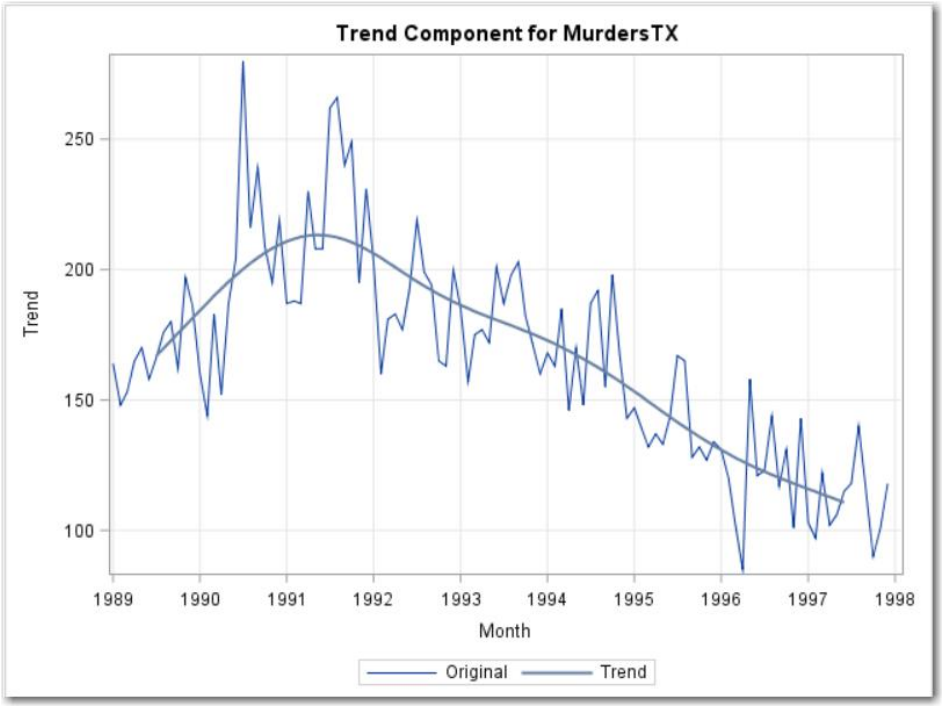


a. Does the data have a seasonal cycle? **The answer is yes.**



Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

b. Is there a trend component in the data? **The answer is yes.**
If so, is it linear? **The answer is no. The trend is better described as quadratic.**



c. Assume that you are one of the Texas governors who were elected in the years 1991 or 1995. Is it reasonable for you to claim that your progressive, yet no-nonsense policies diminished the number of homicides in Texas during your term?
Answer: The trend has a negative slope beginning in mid-1992.

End of Solutions

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

Chapter 2 ARIMAX Models

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

2.1 Autocorrelation and White Noise

Objectives

- Analyze a time series with respect to signal (systematic variation) and noise (random variation).
- Describe the autocorrelation function plot and the white noise test, and discuss their importance in ARMA modeling.

3

Forecasting?

4

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

Forecasting?

The Gambler's Fallacy

- Something that happened more frequently than normal in the past balances out and happens less frequently in the future.
  - Rolling "snake eyes"  two times in a row means that you will not roll it again for a while.
  - Landing on a red-colored number eight times in a row on the roulette wheel means that black number is more likely on the next spin.

- Can you forecast the next roll of the dice from past rolls in a dice game?

5

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

Predictability of Dice Rolls

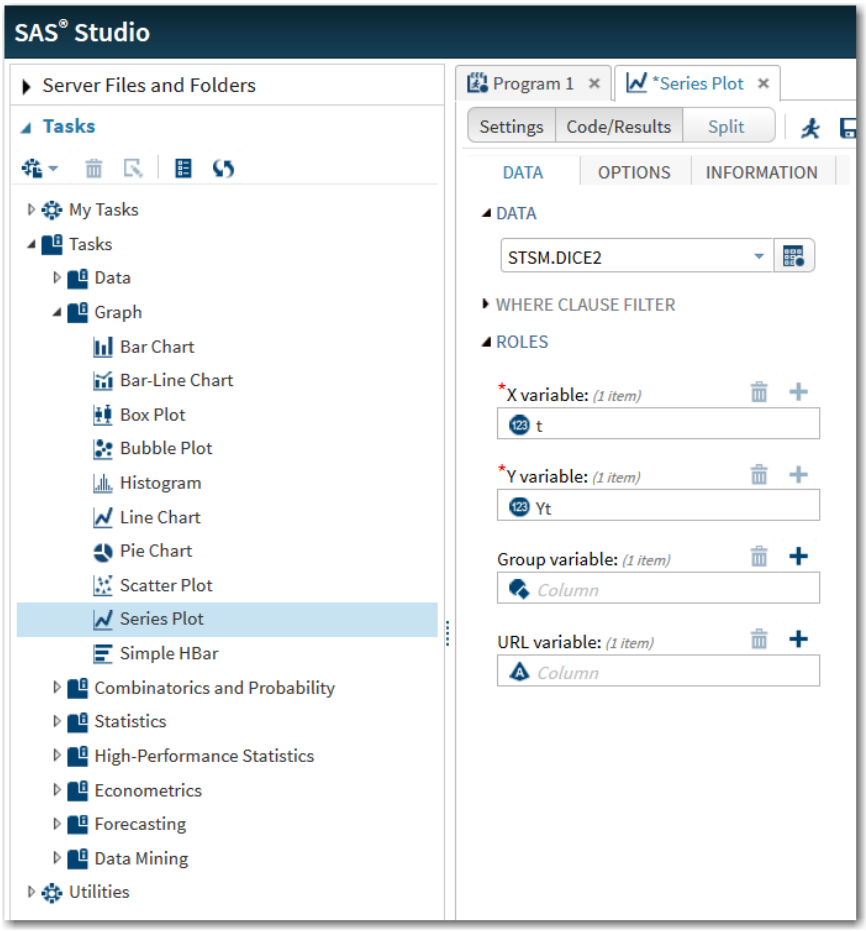
STSM02d01a

The objective of this demonstration is to determine whether you can debunk the Gambler's Fallacy by accurately forecasting future dice rolls based on the previous dice rolls. This demonstration introduces concepts that are revisited in more detail throughout this chapter, and provides the foundation for how to analyze a time series using ARMA and ARMAX models.

This demonstration uses the **stsm.dice2** data set. The **stsm.dice2** data set was created from the **stsm.dice** data set. The **stsm.dice** data set consists of the results of 100 simulated rolls of two standard six-sided dice. The **stsm.dice2** data set lists the sum of the two dice for the current roll, as well as the sum for the twelve previous rolls. The data set contains 14 variables:

- **t**: the ordered value of the roll of the dice
- **Yt**: the sum of the two dice at roll **t**
- **Ytmin1**: the sum of the two dice on the previous roll
- **Ytmin2**: the sum of the two dice from the previous two rolls
- **Ytmin3**: the sum of the two dice from the previous three rolls
- …, and so on, up to **Ytmin12**

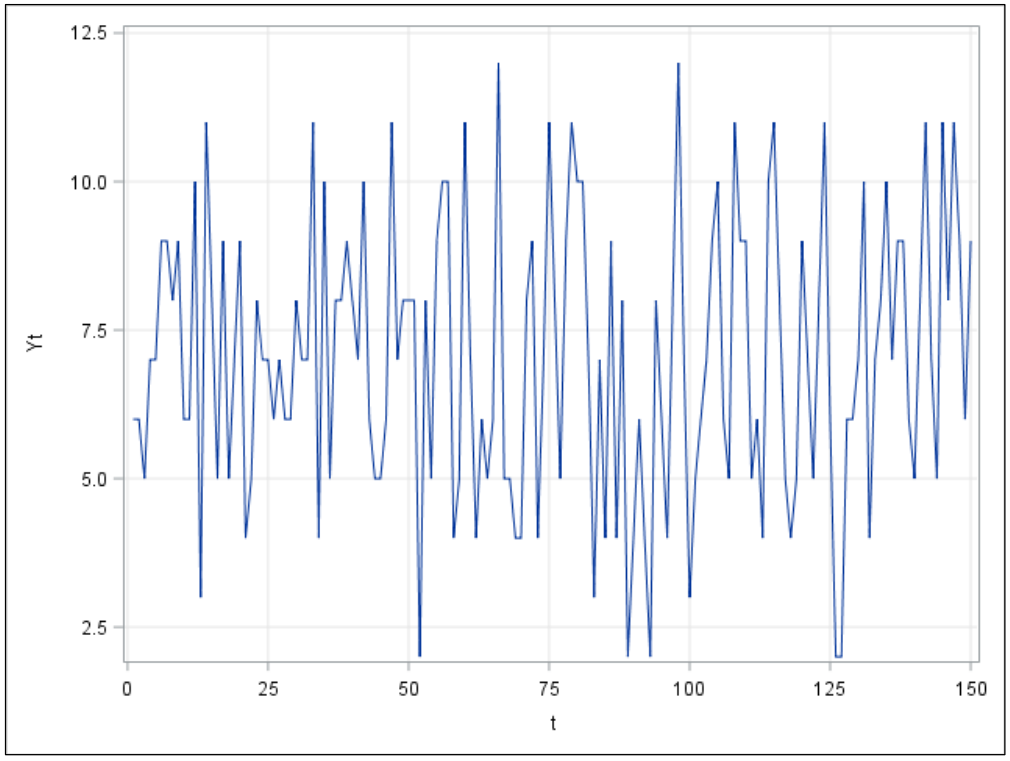
1. The Series Plot task is used to plot the time series variable **Yt**. The purpose for plotting the series is to determine whether the series is stationary. Before any analysis can be performed on the series, the series must be stationary, so perform a quick visual inspection of the plotted series.



Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

The code generated by SAS Studio is as follows:

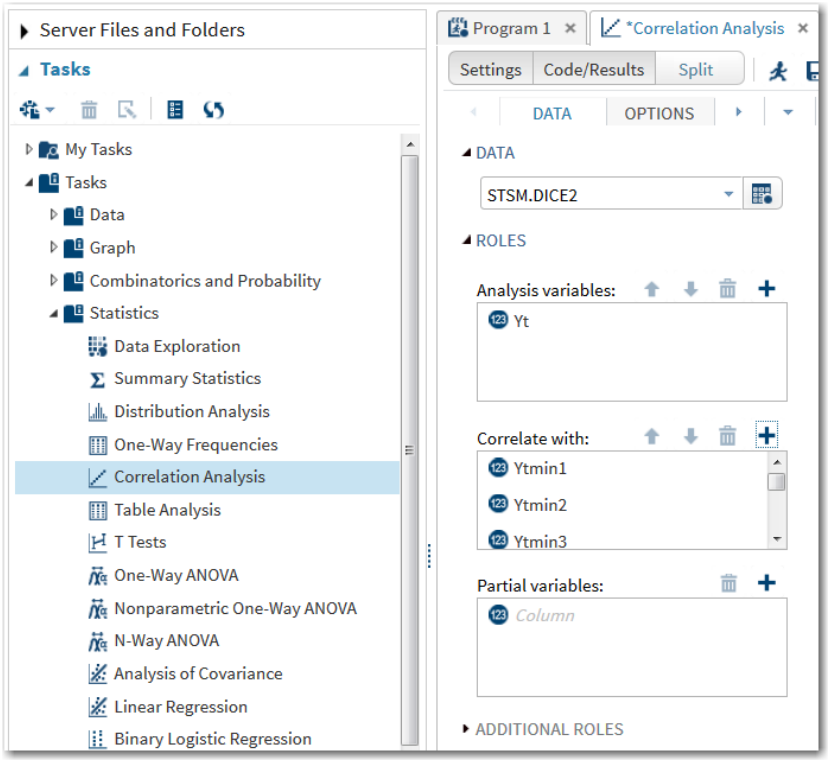
```
/* STSM02d01a.sas */
proc sgplot data=STSM.DICE2;
   /*--Scatter plot settings--*/
   series x=t y=Yt / transparency=0.0 name='Series';
   /*--X Axis--*/
   xaxis grid;
   /*--Y Axis--*/
   yaxis grid;
run;
```



A quick visual inspection concludes that the series is stationary. There are no missing values, and all values for **Yt** are between 2 and 12.

2. Using the **stsm.dice2** data set and the Correlation Analysis task, determine the autocorrelations between **Yt** and all lags of **Yt** through lag 12 (Ytmin1-Ytmin12).

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.



The code generated by SAS Studio is as follows:

```
proc corr data=STSM.DICE2 pearson nosimple noprob plots=none;
   var Yt;
   with Ytmin1 Ytmin2 Ytmin3 Ytmin4 Ytmin5 Ytmin6 Ytmin7 Ytmin8
        Ytmin9 Ytmin10 Ytmin11 Ytmin12;
run;
```

12 With Variables:	Ytmin1 Ytmin2 Ytmin3 Ytmin4 Ytmin5 Ytmin6 Ytmin7 Ytmin8 Ytmin9 Ytmin10 Ytmin11 Ytmin12
1 Variables:	Yt

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

Pearson Correlation Coefficients Number of Observations	
	Yt
Ytmin1	0.06510 149
Ytmin2	-0.09493 148
Ytmin3	-0.03813 147
Ytmin4	-0.00338 146
Ytmin5	0.02918 145
Ytmin6	0.13323 144
Ytmin7	0.02712 143
Ytmin8	-0.06237 142
Ytmin9	-0.03475 141
Ytmin10	0.09083 140
Ytmin11	-0.05716 139
Ytmin12	-0.13927 138

The output shows autocorrelations close to zero at each lag (**Ytmin1-Ytmin12**). This is the first sign that there is no systematic variation in the series. Instead, this suggests that the series might be only a random variation, and thus it is difficult to accurately forecast the next roll.

It is important to note why the **stsm.dice2** data set was used for this demonstration instead of the **stsm.dice** data set. A closer look reveals that the **stsm.dice2** data set is nothing more than a transformed version of the **stsm.dice** data set that creates additional column names for different lagged values. This transformation is necessary to create scatter plots and calculate autocorrelations within the Statistics task in SAS Studio. As this chapter continues, you learn more efficient ways to view and analyze autocorrelations that do not require this data set transformation. The Forecasting task in SAS Studio does not require this transformation, but it could prove valuable if you use scatter plots for presentation purposes.

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

The code used to create the **dice2** data set from the **dice** data set is shown below.

```
%macro lags(newdsn,olddsn,numlags);

data &newdsn;
set  &olddsn;
   %do i=1 %to &numlags. %by 1;
      Ytmin&i.=lag&i.(Yt);
   %end;
run;

proc sort data=&newdsn;
   by descending t;
run;

%mend;

%lags(STSM.dice2,STSM.dice,12);
```

End of Demonstration

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

Correlation of Y with Past Y: *Autocorrelation*

Autocorrelation (Order 1):

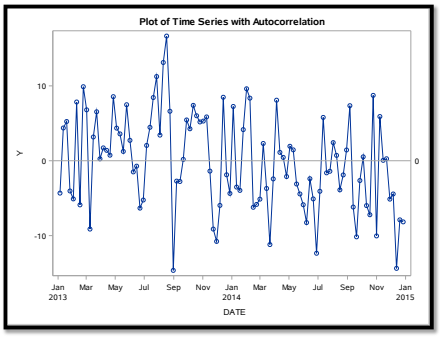
Y_t is correlated with Y_{t-1}

Time Series at Time t:

$$Y_t = Y(t)$$

First Lag:

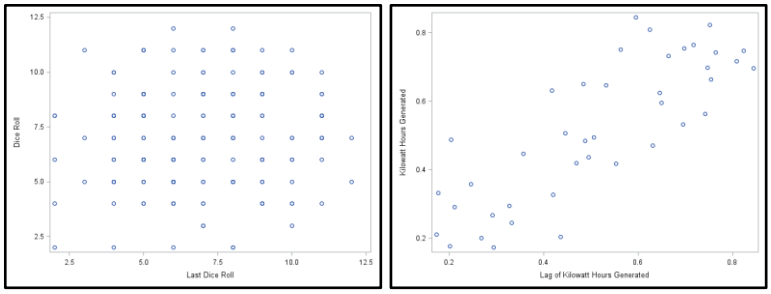
$$Y_{t-1} = Y(t-1)$$



7

Autocorrelation simply means that current values in a time series (Y_t) are related with previous values. The correlation between current values and immediately preceding or *lagged* values (Y_{t-1}) is called *first order autocorrelation*. If the correlation extends to the values two time points previous to current values (Y_{t-2}), that is called *second order autocorrelation*, and so on. Like other correlations, autocorrelations can be either positive or negative with a range between -1 and 1.

Autocorrelation Scatter Plots



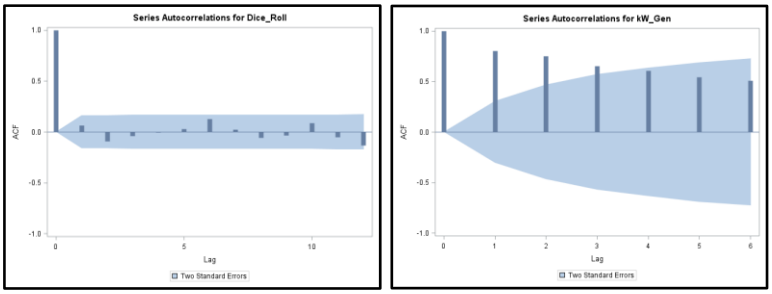
- Autocorrelation is a simple correlation of present values versus lagged values.
- Autocorrelation between the present value and the first lagged value is called *first order* autocorrelation.

8

You could see autocorrelation by creating a column of lagged values in a time series data set and creating a scatter plot. On the left is a plot of the current versus first lagged value of a time series where there is little to no autocorrelation. The right plot shows positive first order autocorrelation.

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

Autocorrelation Plots



- The autocorrelation plot enables you to see the autocorrelation at multiple lags.
- The blue range indicates 95% confidence intervals for each lag.

9

Because autocorrelation could theoretically exist with any ordered lag (Y_{t-p}), it is not reasonable to try to create scatter plots for all. Autocorrelation function (ACF) plots are useful as a first step in detecting potential autocorrelation in a time series. Each spike represents the autocorrelation at lag p. In addition, 95% confidence interval areas are represented by the blue shaded area. Where spikes extend beyond the confidence bounds, the autocorrelation is said to be statistically significant at the 0.05 level.

A spike representing the autocorrelation at lag 0 (always equal to 1) is included for comparison.

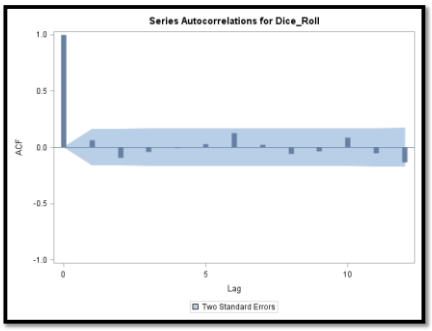
The left plot shows the ACF plot for the series represented on the left of the previous slide. There are no significant spikes. However, there are three significant spikes in the ACF on the right. Does that mean that there is autocorrelation at three lags? Perhaps not. (You learn more about detecting the order of autocorrelation in a time series in a later section.)

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

White Noise

- a series that varies randomly around its mean
- no systematic variation
- comprised of only random variation

Forecasting a white noise process reverts to the mean.



Mean
7.02

Yt	Count
2	5
3	3
4	15
5	21
6	21
7	21
8	21
9	18
10	11
11	12
12	2

10

Think about the dice roll time series example. You know that the last roll of the dice is not predictive of the next roll. Dice rolls are governed by a random process. The expected average number of dots shown in a roll of two dice is 7 and the expected standard deviation is 2.41. It does not matter whether the dice roll is the first, the seventh, or the 670,000th (so that you do not need to try this at home). The mean and variance should remain constant and each dice roll is independent of all other dice rolls. In time series terminology, this is considered a *white noise* series.

By definition, a white noise series has no autocorrelation. If you are trying to forecast the next value of a white noise series, your best guess is always the mean of the series.

Is white noise in a time series good or bad? It depends. If a series itself is simply white noise, it means that it is not forecastable. However, if the residual values (actual minus predicted values) are white noise, that indicates that the elements that you included in your model adequately explained all that is explainable (the signal) in the model. What was not explained is not explainable. In other words, white noise in the residuals is desirable.

A white noise series technically implies a mean of 0, although even a series with a nonzero mean can be considered white noise, as long as the series of deviations from the mean ($Y_t - \bar{Y}_t$) are white noise.

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

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.
  - H_0: The series is white noise.
  - H_1: The series is ***not*** white noise.
- The Ljung-Box test can be applied to the original series or to the residuals after fitting a model.

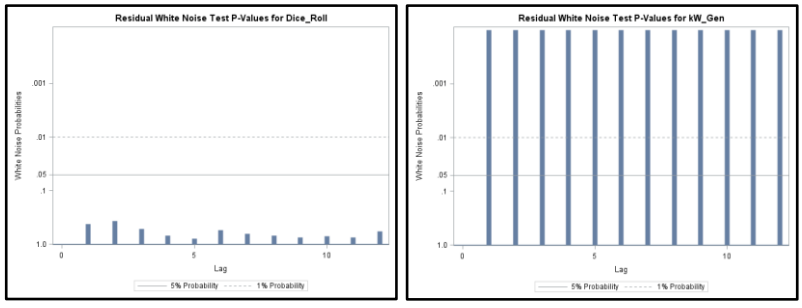
11

A popular test for white noise is the Ljung-Box test. The test statistic is calculated as

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

The statistic is cumulative, meaning that the null hypothesis is that all of the autocorrelations up to, and including lag *m*, are white noise. A rejection of the null hypothesis at level *m* does not inform which lags are causing the significant result.

The Ljung-Box Chi-Square Test for White Noise



"White means white."

12

The plot of the Ljung-Box chi-square test can be used to quickly assess whether the autocorrelation at any lag rejects the white noise assumption.

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

Notice the scale and ordering of values on the Y axis. The order is descending from bottom to top and the probability values (representing *p*-values) are not linearly scaled. This representation enables you to see statistical significance at various significance levels (0.10, 0.05, and 0.01) more easily. It also means that non-significant tests (high *p*-values) are represented by short spikes. Hence, the expression "White means white" when you glance at the plot.

The white noise plot is sometimes displayed with the Y-axis values in ascending order from bottom to top. In that representation, long bars represent high *p*-values, and therefore, white noise. You should pay attention to the Y-axis values before you draw conclusions about the white noise tests.

The Ljung-Box Chi-Square Test for White Noise

White Noise Series

Autocorrelation Check for White Noise									
To Lag	Chi-Square	DF	Pr > ChiSq	Autocorrelations					
6	5.00	6	0.5436	0.065	-0.095	-0.038	-0.004	0.028	0.129
12	10.39	12	0.5821	0.026	-0.060	-0.033	0.087	-0.054	-0.131

Autocorrelated Series

Autocorrelation Check for White Noise									
To Lag	Chi-Square	DF	Pr > ChiSq	Autocorrelations					
6	121.35	6	<.0001	0.804	0.750	0.652	0.608	0.546	0.509
12	147.33	12	<.0001	0.477	0.384	0.265	0.158	0.095	0.027

13

These tables of white-noise tests show cumulative results for six lags at a time. The top table shows non-significance up to lag 6 and also to lag 12. No individual autocorrelation value exceeded 0.131 in absolute value. This came from the dice roll data. The bottom table shows the white noise tests for a series that has autocorrelation. Notice that even though the test for white noise to lag 12 is statistically significant, none of the last three autocorrelations was greater than 0.158. Remember that these tests are cumulative.

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

Forecasting Solar Power Production

14

The **SOLARPV** data set contains the following variables:

EDT	date of Saturday ending the measurement week
kW_Gen	average daily solar electricity production in the week in kilowatt hours
Cloud_Cover	average daily estimated cloud cover in the week, scaled 0-10

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

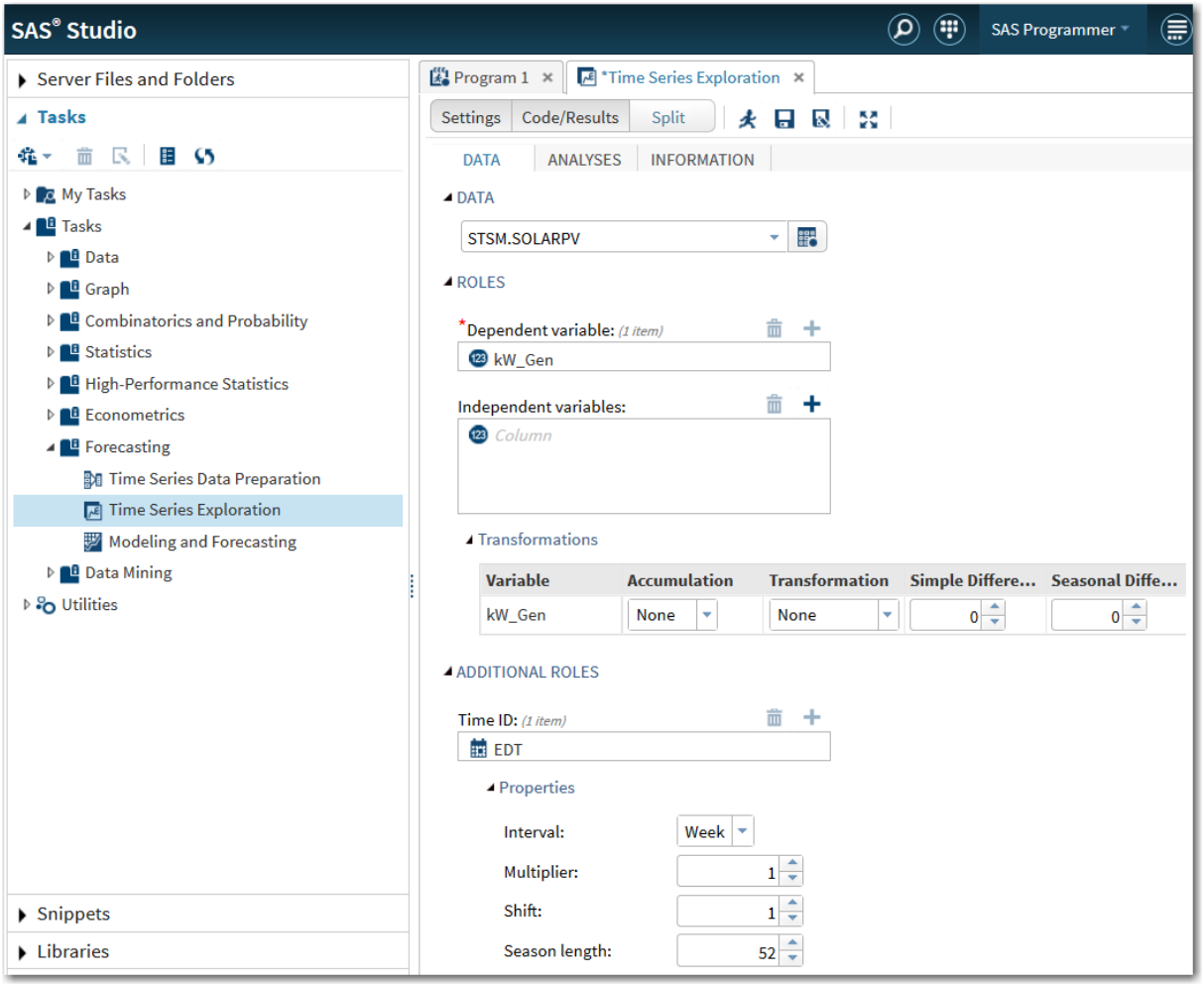
Autocorrelation and Solar Production

STSM02d01b

This demonstration uses the **stsm.solarpv** data set and the Time Series Exploration task to help visualize the **kW_Gen** series and determine whether there is a systematic variation that can be used to forecast future periods of solar power generation. By analyzing the autocorrelation function plot and the white noise probability plot, it can be determined whether the series is white noise.

The variables in the **stsm.solarpv** data set are the following:

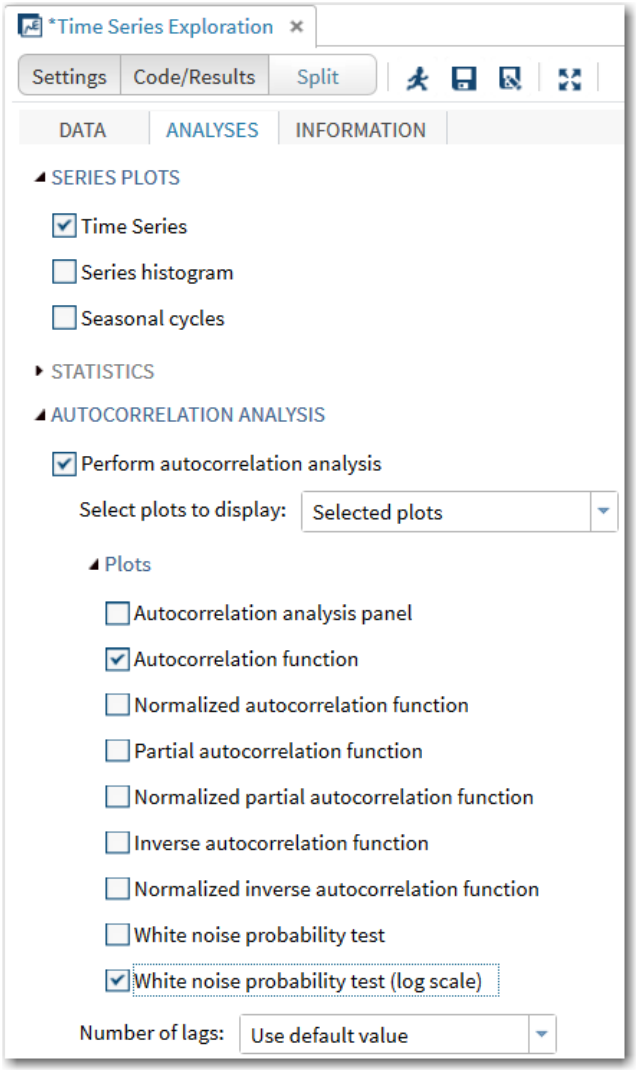
- **EDT**: time interval (weekly)
- **kW_Gen**: kilowatts of solar power generated (averaged per day)
- **Cloud_Cover**: a metric that quantifies average weekly cloud cover in the area



Under AUTOCORRELATION ANALYSES on the Split tab, only **Autocorrelation function** and **White noise probability test (log scale)** are selected. A quick visual inspection of both of these plots determines whether the series is white noise.

Why was **White noise probability test (log scale)** selected instead of *White noise probability test*? The log scale plot conforms to the "White Means white." phrase from the earlier slide. ***Be cautious of the Y axis.*** Both plots lead to the same conclusion, but can be easily misinterpreted if careful attention is not paid to the Y axis.

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.



The code generated by SAS Studio is as follows:

```
proc sort data=STSM.SOLARPV out=WORK.TempSorted;
   by EDT;
run;

proc timeseries
     data=WORK.TempSorted seasonality=52 plots=(series acf wn);
   id EDT interval=week;
   var kW_Gen / accumulate=none transform=none dif=0 sdif=0;
   ods exclude ACFNORMPlot;
   ods exclude WhiteNoiseProbabilityPlot;
run;

/* Remove the temp data set */
proc delete data=WORK.TempSorted;
run;
```

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

Alternatively, you can write the code directly in SAS/ETS.

```
/* STSM02d01b.sas */
proc timeseries data=STSM.solarpv
                seasonality=52
                plots=(series acf wn);
   id EDT interval=week;
   var kW_Gen;
   ods exclude ACFNORMPlot WhiteNoiseProbabilityPlot;
run;
```

SAS Studio uses a **Work.TempSorted** data set to do the analysis instead of the original data set **stsm.solarpv**. This is true for other tasks in SAS Studio as well, and ensures that the original data set remains unchanged throughout the entire process. Throughout the remainder of Chapter 2, the displayed code does not include the PROC SORT or PROC DELETE procedures.

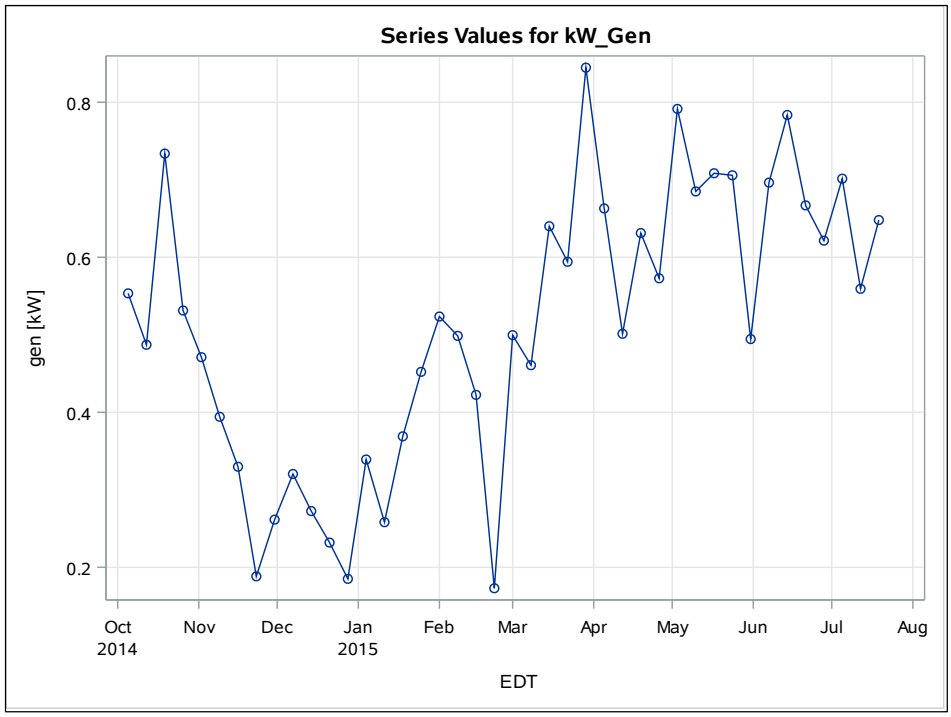
Input Data Set	
Name	WORK.TEMPSORTED
Label	
Time ID Variable	EDT
Time Interval	WEEK
Length of Seasonal Cycle	52

The Variable Information table provides basic information about the series, and enables confirmation that the appropriate data and time range are being used.

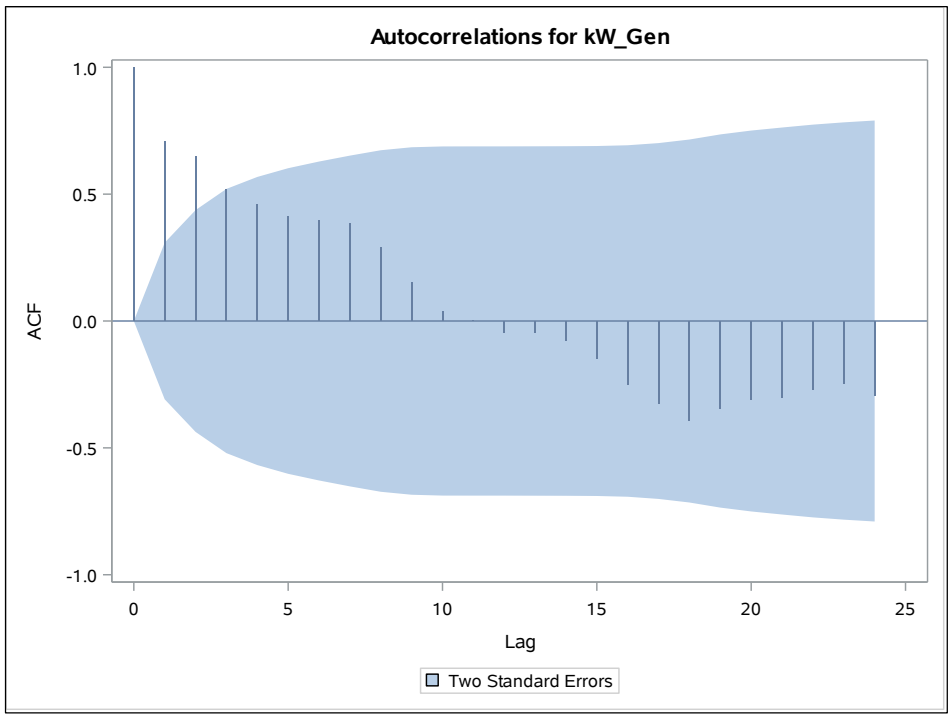
Variable Information	
Name	kW_Gen
Label	gen [kW]
First	Sun, 5 Oct 2014
Last	Sun, 19 Jul 2015
Number of Observations Read	42

A plot of the data is generated as part of the output. Because no transformation was applied, the plot is of the raw **kW_Gen** series. Because the Time Series Exploration task creates this plot, graphing the series using the Graph task in SAS Studio is unnecessary.

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

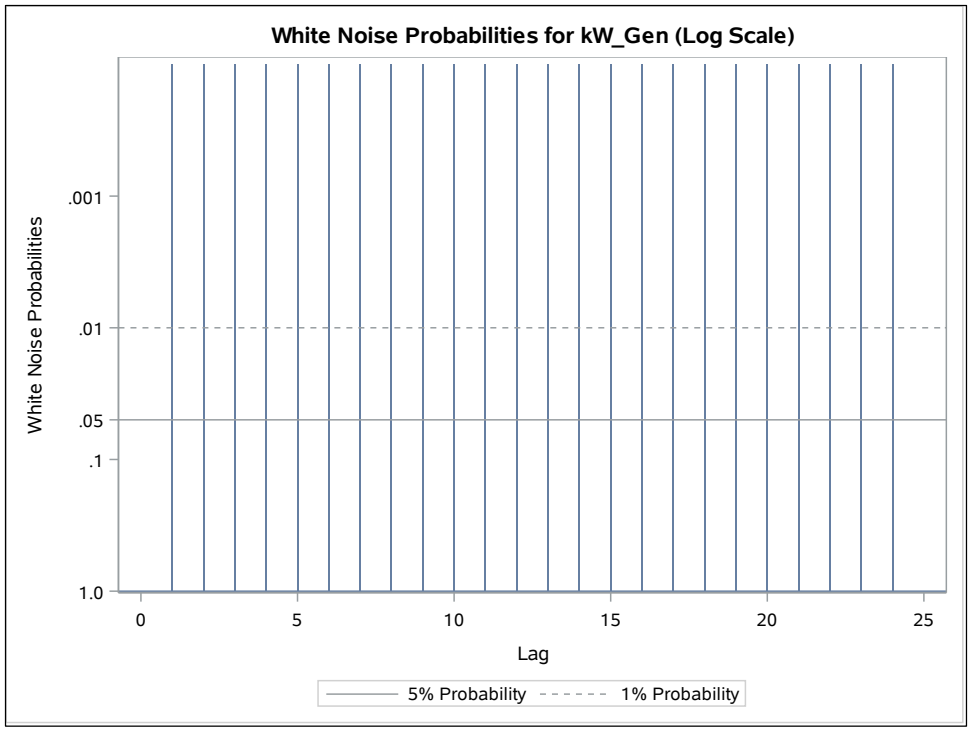


The autocorrelation function plot (ACF) shows at least one nonzero lag with a significant spike. This indicates that the series contains autocorrelation. The 95% confidence intervals are higher due to having only 42 observations, but there are clearly two nonzero lags with significant spikes.



The White Noise Probabilities plot confirms that the series is ***not*** white noise, and that the series contains a systematic variation that can be used to forecast. Recall that the null hypothesis for the White Noise test is that the series is white noise. The White Noise Probabilities plot strongly rejects the null hypothesis at all lags, concluding that the series is not white noise.

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.



End of Demonstration

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

2.01 Multiple Answer Poll

Which of the following are true?

a. Failing to reject the null hypothesis of the white noise probability test implies that the series is white noise.
b. First order autocorrelation is the correlation between the current value and the immediately preceding value.
c. A time series requires at least one measure of chronological time.
d. You can now accurately forecast future spins of the roulette wheel and share future winnings with your instructor.
e. A white noise process implies that there is no autocorrelation.

16

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

2.2 ARIMA, ARMA, and Stationarity

Objectives

- Discuss the differences between ARMA and ARIMA models.
- Define a stationary time series and discuss its importance.
- Describe and identify autoregressive and moving average processes.

20

What Is ARIMA?

AR
- AutoRegressive
  - Current values are related to past values.

I
- Integrated
  - Differenced values between successive time points can be modeled.

MA
- Moving Average
  - Current values are related to past estimation errors (that is, shocks).

✎ ARMA models are those that do ***not*** require integration.

21

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

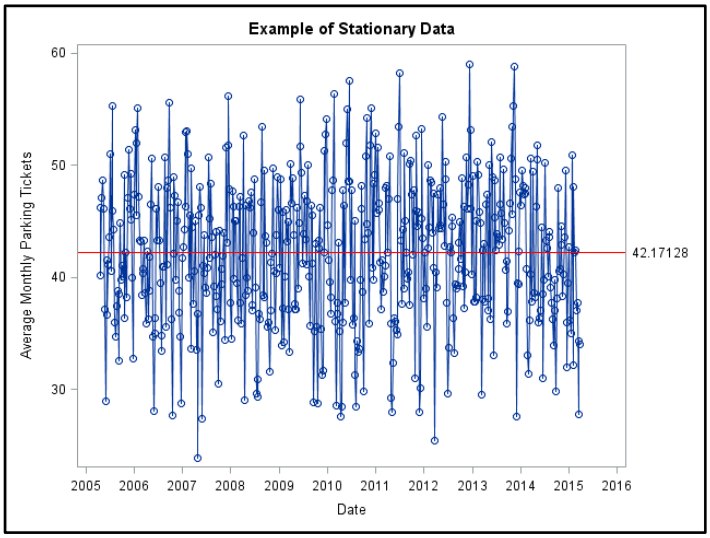
Stationarity

- A *stationary* time series is defined as having a constant mean, constant variance, and that any autocorrelation between adjacent terms is constant across all time periods.
- A *nonstationary* time series does not have a constant mean and variance, and tends to exhibit a discernable pattern in the data across time.
- 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.

22

ARMA and ARIMA models follow very similar methodology. The main distinction is that ARMA models are used when the series is stationary on its original scale. If the series is not stationary on its original scale, it needs to be transformed to create a stationary series through integration, and thus ARIMA.

Visualizing Stationary Data

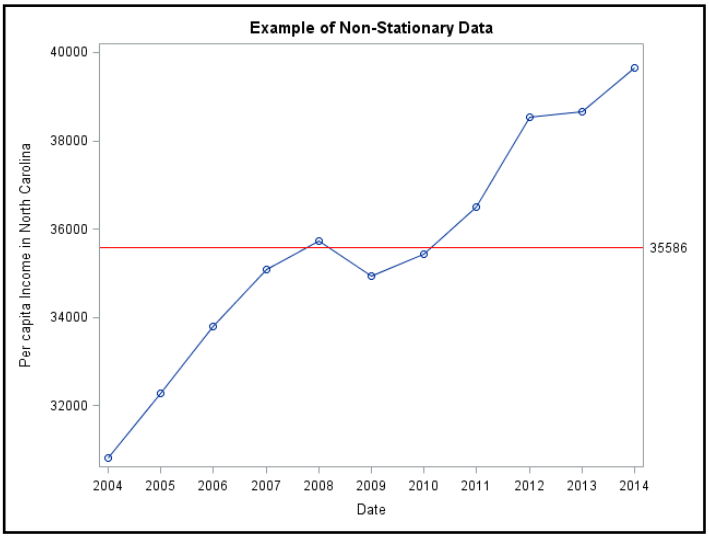


23

The data seem to hover around a constant mean and exhibit constant variance. The figure does not show any apparent trend in the data.

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

Visualizing Nonstationary Data



24

Seasonal and trending data can be quickly identified as nonstationary.

ARMA and Stationarity

- ARMA models require a stationary time series to produce reliable forecasts.
- If your data is not stationary, you must transform your series to make it stationary.
- This is typically done by transforming the series (for example, by *differencing* (the change between current values and previous values) your series) or taking the square root of the series values, and then modeling your transformed series instead of the actual values themselves.

25

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

First Differencing Example

Year	Income	Lag(Income)	First Difference
2004	$ 30,818	⚠	⚠
2005	$ 32,296	$ 30,818	$ 1,478
2006	$ 33,808	$ 32,296	$ 1,512
2007	$ 35,076	$ 33,808	$ 1,268
2008	$ 35,725	$ 35,076	$ 649
2009	$ 34,942	$ 35,725	$ -783
2010	$ 35,435	$ 34,942	$ 493
2011	$ 36,508	$ 35,435	$ 1,073
2012	$ 38,538	$ 36,508	$ 2,030
2013	$ 38,653	$ 38,538	$ 115
2014	$ 39,646	$ 38,653	$ 993

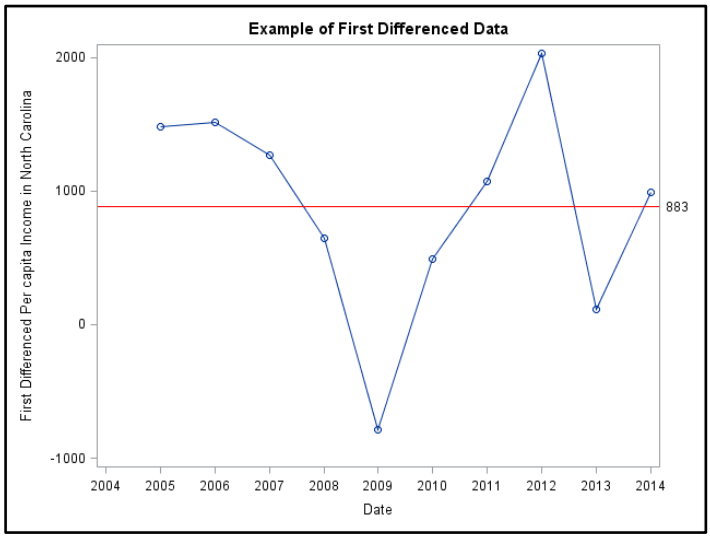
26

Notice how the first difference is calculated. It is only the change between the current row and the prior row.

Remember that applying a first difference eliminates one observation from your data set. The year 2004 is the first row in the series. Because there is no data prior to 2004, a first difference calculation cannot be calculated. Therefore, you use one less observation when identifying, estimating, and forecasting.

Losing one data point is often manageable, but consider a sixth difference or a twelfth difference. Taking the current row and subtracting six previous rows eliminates six observations. A twelfth difference eliminates 12. Be aware of how many data points you have when you determine the appropriate difference, if your series is not stationary.

First Differencing



27

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

ARMA versus ARIMA Models

The "I" in ARIMA stands for *integrated*, and tells you in what *order* the data was differenced to convert it to a stationary process.

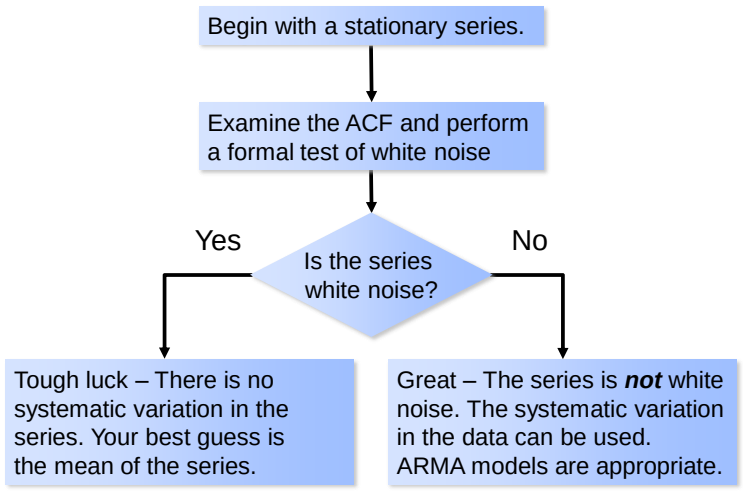
- starting with a stationary process: ARMA model
- starting without a stationary process: Transforming the data in order to create a stationary process warrants using an ARIMA model.

✎ This class works with stationary time series and thus uses ARMA models.

28

Every example in this class uses a stationary series on the original scale. No differencing is applied.

ARMA Models: Initial Process Flow



29

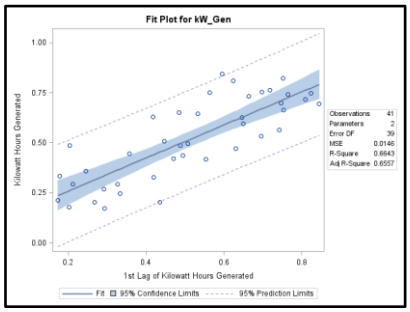
Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

Regression of Y on Past Y: *Autoregression*

Reminder:

OLS Regression Model: $Y = \beta_0 + \beta_1 X + \varepsilon$

Autoregressive (Order 1) Model:



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

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

30

Determining Autoregressive Order

- Confirm that the series is stationary and not white noise.
- Determine which lagged values (Yt-1, Yt-2, Yt-3, and so on) are correlated with the current value (Yt), adjusting for the autocorrelation of all lower order lags.

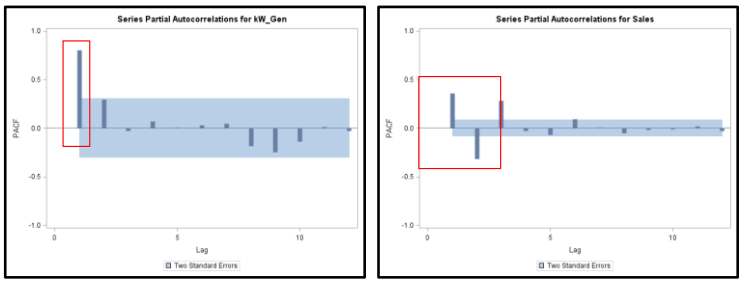
  The partial autocorrelation function plot (PACF) helps determine this by answering the following questions:
  - Is there significant autocorrelation between Y_t and Y_{t-1}?
  - Is there significant autocorrelation between Y_t and Y_{t-2}, holding constant the autocorrelation between Y_t and Y_{t-1}?

31

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

Partial Autocorrelation Function Plot (PACF)

- Significant spikes in the PACF are the most important source of information in identifying an autoregressive series.



- Unlike the ACF, autocorrelations do not "spill over" from lag to lag, but rather hold constant all lower order lags.

32

The ACF does not hold autocorrelations between lower order lags constant. This results in the "spill over" effect, also referred to as the *proximity effect*. This makes it difficult, if not impossible, to determine which lags are truly influencing the current value when the systematic variation is autoregressive.

Partial autocorrelations work similar to first derivatives. They isolate the autocorrelation between Y_t and Y_{t-k}, and hold all lower order autocorrelations between Y_t and Y_{t-k+1}, Y_{t-k+2}, …, Y_{t-1} constant. This enables the analyst to quickly determine the autoregressive order when the systematic variation is purely autoregressive. Thus, the PACF does not fall victim to the proximity effect.

Autoregressive versus Moving Average Models

First Order Autoregressive Model AR(1)

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

– Y_t is a function of the previous value plus some error.

First Order Moving Average MA(1)

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

– Y_t is a function of its immediately previous shock plus error (significant autocorrelation between Y_t and ε_{t-1}).

33

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

Moving Average

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

- A moving average is generated by a weighted average of random disturbances going back (q) periods.
  - Error terms are assumed to be white noise, normally distributed with a mean of zero.
- Unlike autoregressive processes, moving average processes have short-term, finite memories.
  - used to model short-lived or more abrupt patterns in the data

34

Moving Average: Temporary Shock Scenarios

- Shocks are exogenous.
- Example: Demand forecasting
  - a competitor's fixed-time-period sales promotion
    - 30% off sale, buy-one-get-one-free, and so on
    - After the promotion ends, the series instantly reverts to the mean.
  - an advertising campaign
    - the ALS "Ice Bucket Challenge"
  - positive or negative media coverage
    - The effect diminishes as the shock moves further into the past.

35

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

ARIMA Ordering - ARIMA(p,d,q)

AR	• Autoregressive order = *p*
I	• Differencing order = *d*
MA	• Moving average order = q

- ARMA models are ARIMA models with *d*=0 and are denoted ARMA(*p*,*q*).

36

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

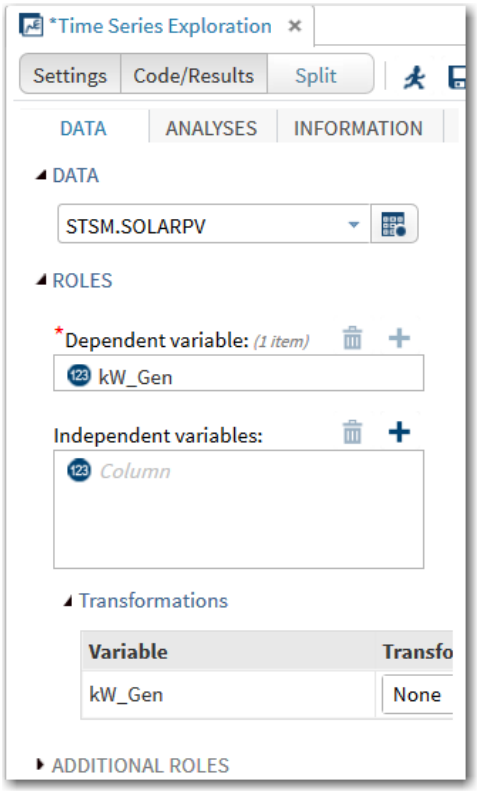
Time Series Identification

STSM02d02

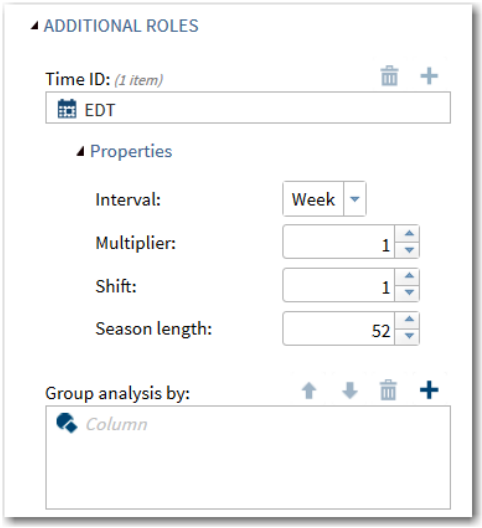
The series **stsm.solarpv** is analyzed to determine whether there is an autoregressive process in **kW_Gen** and, if so, in what order.

Create a new Time Series Exploration task in SAS Studio.

1. On the DATA tab, select the data set **SolarPV**. Then, select **kW_Gen** as the dependent variable.



2. Click the triangle next to ADDITIONAL ROLES and then select **EDT** as the time ID and accept the properties that are populated. SAS recognizes **EDT** as weekly.



Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

The generated SAS syntax is shown below.

```
proc timeseries data=WORK.TempSorted
                seasonality=52
                plots=(series corr);
   id EDT interval=week;
   var kW_Gen / accumulate=none transform=none dif=0 sdif=0;
run;
```

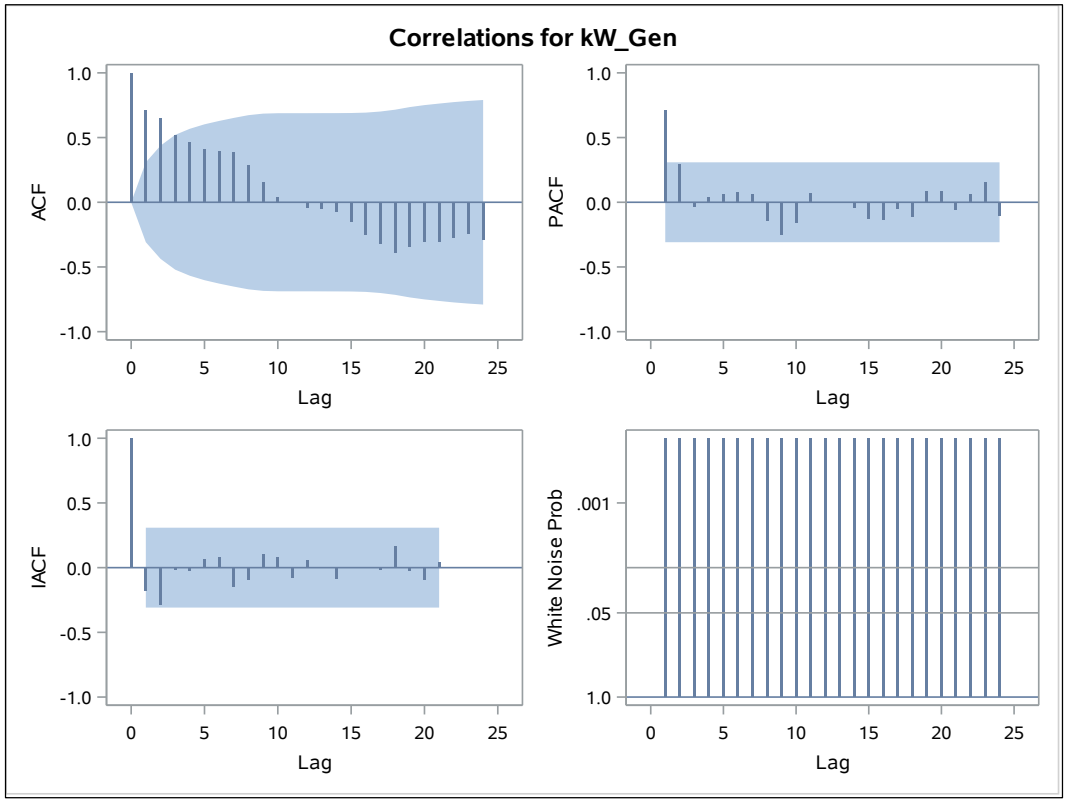
Alternatively, you can write the code directly in SAS/ETS:

```
proc timeseries data=STSM.SOLARPV plots=(corr);
   id EDT interval=week;
   var kW_Gen;
run;
```

3. Submit the code.

Some of the output is the same as in the previous demonstration and is not displayed here.

The ACF plot shows a pattern of gradually declining autocorrelation as lags increase. The PACF plot gives a clearer picture of the true autoregressive order, because it removes the proximity effect. The first lag is the only clearly significant lag, which implies an autoregressive order of 1.



The IACF is the inverse autocorrelation function. If the model is a pure autoregressive model, then the IACF is an ACF that corresponds to a pure moving average model. It cuts off sharply when the lag is greater than *p*. This behavior is similar to the behavior of the partial autocorrelation function (PACF).

End of Demonstration

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

2.02 Multiple Answer Poll

Which of the following is a stationary process?

a. a series that, when graphed, appears to exhibit a constant mean and variance across all time periods
b. a necessary component needed before ARMA modeling can occur
c. often the result after differencing a nonstationary series
d. the paper and envelopes used for writing correspondence

38

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

Exercises

1. Analyzing a Rose Sales Series

STSM.ROSESERIES contains four series, named **SALES1** through **SALES4**. These data represent average weekly sales of roses over a 10-year period for four different stores. The data are simulated.

In this exercise, either use SAS Studio tasks or code SAS programs directly using SAS/ETS procedures to determine whether there is any apparent autocorrelation in any of the series.

Which series show autocorrelation?

End of Exercises

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

2.3 Estimation of Autoregressive Parameters

Objectives

- Estimate an order 1 autoregressive model.
- Assess the fit of the model.
- Analyze the residuals and check error assumptions.

42

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.

43

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

Box and Jenkins built on the work of others, such as Yule (developer of AR models), Slutsky (the developer of MA models), and Wold (who brought them all together). Box and Jenkins not only added the I to ARMA models, creating ARIMA models, but they also formulated a methodology for analyzing time series data from initial investigation to implementation of models for real world use. Statistical software written to perform ARIMA modeling reflects this methodology of *Identify, Estimate, and Forecast*.

In the previous section, you learned about the identification process. This section and the next describe the estimation stage, where the ARIMA model parameter estimates are calculated and models are assessed for goodness of fit.

Estimation of an AR(1) Model

- Recall that for a first order autoregressive model:

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

- For series with mean, $\mu = 0$, $\phi_0 = 0$.
- In general, $\phi_0 = \mu(1-\phi_1)$, so the following is true:

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

44

The AR(1) formula can be written in terms of deviations from the series mean.

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

Because $\mu(1-\phi_1)$ does not depend on time (there is no time subscript), it is often referred to as a constant, μ_0. You often see the general formula written as $Y_t = \phi_0 + \phi_1 Y_{t-1} + \varepsilon_t$, where $\phi_0 = \mu(1-\phi_1)$.

Parameter estimation can be done using the methods of unconditional least squares, conditional least squares, and maximum likelihood. Maximum likelihood is generally the preferred method, although it is not always the default method in all forecasting software.

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

ML Estimation of an AR(1) Model

Maximum Likelihood Estimation					
Parameter	Estimate	Standard Error	*t* Value	Approx Pr > |t|	Lag
MU	42.59964	0.46518	91.58	<.0001	0
AR1,1	0.45529	0.03909	11.65	<.0001	1

- **MU** is the estimated mean of the series, μ.
- **AR1,1** at **Lag=1** is the estimated first order autoregression parameter, ϕ_1.
- *P*-values test H_0: parameter=0.
- If the series is white noise, then all parameter estimates, other than that for MU, should be non-significant.

45

"AR1,1" does not necessarily refer to the first order autoregressive parameter. You must look at the lag value to determine the order of the parameter being estimated and tested.

Also, notice that the *p*-value column is titled "Approx Pr > |t|." The *p*-value is considered approximate because the standard errors estimates approximate, based on large sample theory.

Accuracy versus Goodness of Fit

- A diagnostic statistic that is calculated using a holdout sample that was not used in modeling is an *accuracy* statistic.
- Assessing a predictive model using accuracy statistics that are calculated for a holdout sample is called *honest assessment*.

46

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

Accuracy versus Goodness of Fit

- 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.

47

Model Goodness-of-Fit Statistics

A diagnostic statistic calculated using the same sample that was used to fit the model is a *goodness-of-fit* statistic.

Constant Estimate	23.20426
Variance Estimate	33.5095
Std Error Estimate	5.788739
AIC	3304.076
SBC	3312.583
Number of Residuals	520

48

In this table, also notice that the constant estimate is the estimate of $\mu(1-\phi_1)$.

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

Model Goodness-of-Fit Statistics

Information Criterion Formula ***(Smaller is Better!)***:

Akaike's A Information Criteria:

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

Schwarz's Bayesian Information Criteria

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

- Series Length: *n*
- Number of Model Parameters to Estimate: *k*
- Model Likelihood Function Evaluated at Maximum

49

The AIC and SBC general formulas are **IC=Accuracy + Penalty**. Where the estimation is maximum likelihood, accuracy is estimated by -2log(L), where log is the (base *e*) natural log and L represents the estimate of the likelihood. If another method is used, the value of **Accuracy** is approximated differently.

These accuracy measures are used to assess the relative fit of the model. There are no standards of AIC or SBC for concluding that any model fits well. The values can be used to compare one candidate model to another. The model with the smaller value (or more negative value, in some cases) is the better fitting model.

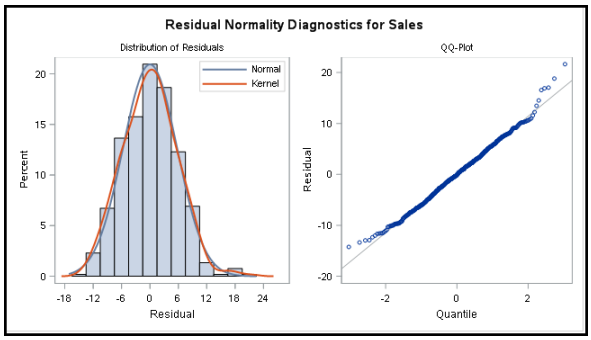
The value of -2log(L) is affected by the number of parameters. Values of -2log(L) are always reduced by the addition of other parameters (even random ones). That is why this "accuracy" measure is not used in practice to compare models. It suffers most from the "optimism principle."

The penalty for AIC is based on the number of parameters only, whereas the penalty for SBC is also affected by sample size (size of the series, in this case). The SBC carries the more severe penalty for adding additional parameters, so it is a more conservative accuracy measure.

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

Check of Residuals

- The residuals are $(Y_t - Y_t)$.
- White noise assumption
  - **normal distribution with a mean of 0 and constant variance σ^2**
  - independence of observed values at different times



50

Remember that there are statistical assumptions for ARIMA models. They are concerned with the errors of the model. The residuals of the model (the difference between the actual value and the predicted value at each time point *t*) can be used to assess those assumptions. In particular, there is a white noise assumption of the error. It is common to assess normality using both histograms and quantile-quantile plots of the residuals.

Check of Residuals

- White noise assumption
  - normal distribution with a mean of 0 and constant variance
  - **independence of observed values at different times**

Autocorrelation Check of Residuals

To Lag	Chi-Square	DF	Pr > ChiSq	Autocorrelations					
6	1.09	6	0.9820	0.000	-0.005	0.006	-0.042	-0.013	0.006
12	8.99	12	0.7034	-0.024	0.111	-0.010	0.005	-0.011	0.043
18	14.31	18	0.7085	0.028	-0.033	-0.022	-0.068	0.047	0.028
24	17.86	24	0.8100	0.027	0.046	0.019	0.009	-0.018	0.054
30	24.93	30	0.7284	0.089	-0.031	-0.053	-0.007	-0.023	0.024
36	32.30	36	0.6451	-0.028	0.070	-0.038	-0.072	-0.030	0.001
42	39.59	42	0.5775	-0.010	0.086	-0.035	-0.055	-0.034	-0.008
48	44.56	48	0.6147	-0.008	-0.006	-0.069	-0.054	-0.026	0.011

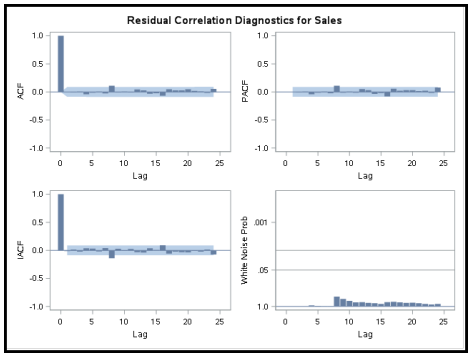
51

You can check autocorrelations of the residuals using the Ljung-Box test for white noise. In addition to checking *p*-values, you might also want to check the individual autocorrelation estimates to make sure that there is not one suspiciously high value.

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

Assumptions for Residuals

- White noise assumption
  - normal distribution with a mean of 0 and constant variance σ^2
  - **independence of observed values at different times**



52

Finally, checks of the ACF and the PACF enable you to inspect a graphical presentation of the white noise analysis.

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

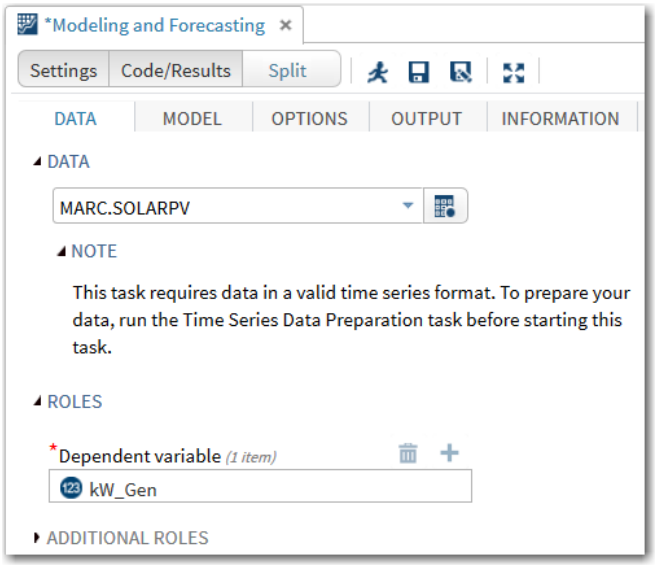
Estimation, Residual Analysis, and Goodness-of-Fit

STSM02d03

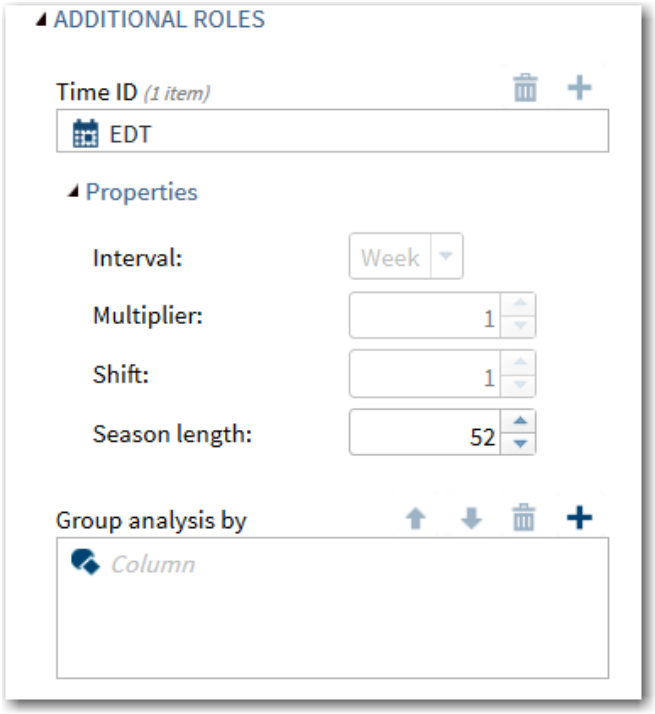
Estimate an AR(1) model for the **SolarPV** data set. Check the residual series to see whether it is white noise. Display the goodness-of-fit statistics for comparison with future models.

Use the kilowatts generated (**kW_Gen**) time series.

1. Create a new Modeling and Forecasting task in SAS Studio.
2. On the DATA tab, select **SolarPV** as the data set. Select **kW_Gen** as the dependent variable.

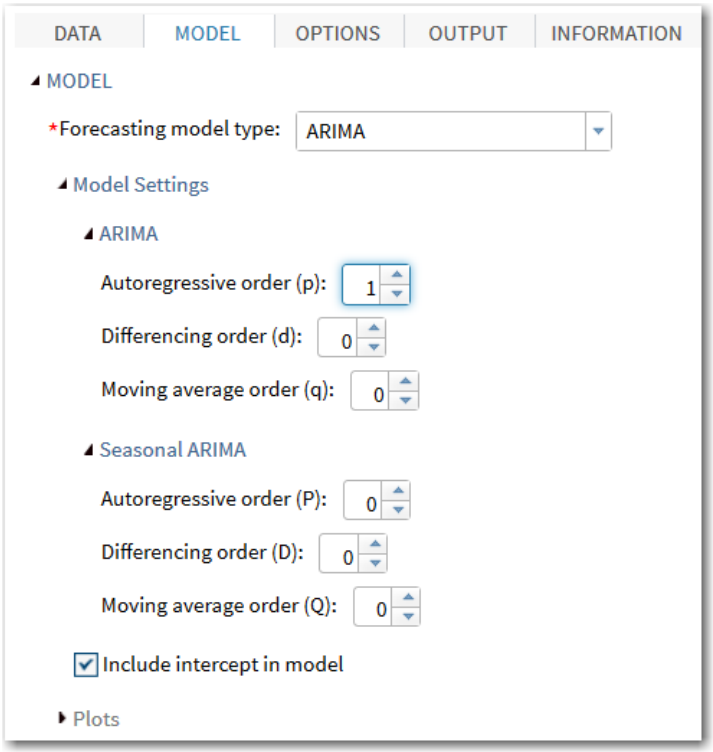


3. Click the triangle next to ADDITIONAL ROLES and then select **EDT** as the time ID and accept the properties that are populated. SAS recognizes **EDT** as weekly.

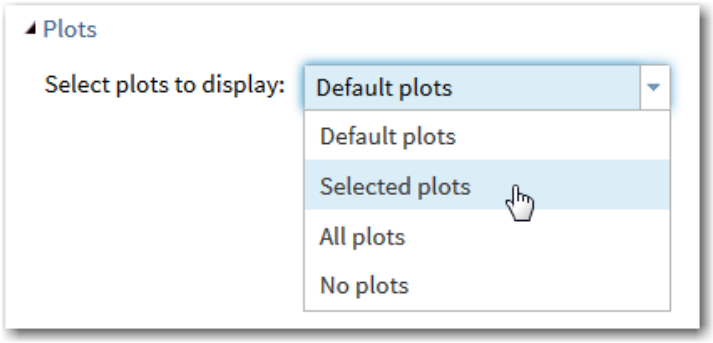


Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

4. On the MODEL tab, select **ARIMA** as the forecasting model type. Model settings appear. Select **1** in the **Autoregressive order (p)** field under ARIMA.

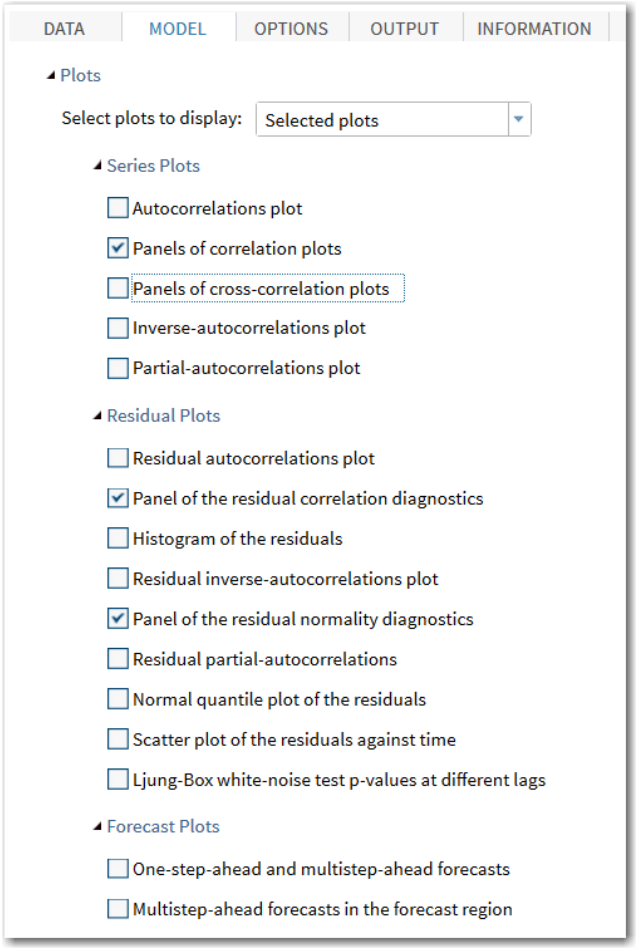


5. Expand **Plots** and click **Selected plots**.

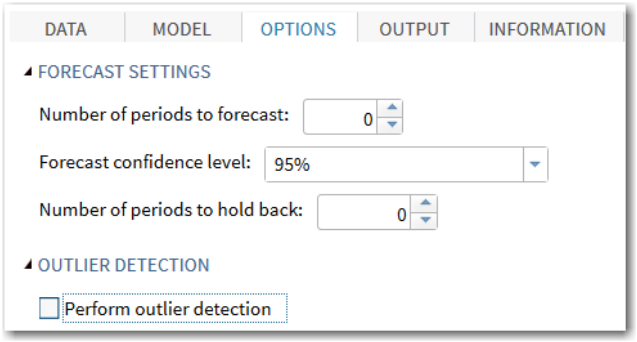


Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

6. Clear the **Panels of cross-correlations plots** check box under Series Plots and the **One-step-ahead and multistep-ahead forecasts** check box under Forecast Plots.



7. On the OPTIONS tab, set the **Number of periods to forecast** field to **0** under FORECAST SETTINGS. Clear the **Perform outlier detection** check box under OUTLIER DETECTION.



Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

The generated SAS syntax is shown below.

```
/* ARIMA or ARIMAX                                                    */
proc arima data=WORK.TempSorted
           plots(only)=(series(corr) residual(corr normal) );
   identify var=kW_Gen;
   estimate p=(1) method=ML;
   forecast lead=0 back=0 alpha=0.05 id=EDT interval=week;
quit;
run;
```

Alternatively, you can write the SAS/ETS code directly:

```
/* STSM02d03.sas */
ods noproctitle;
ods graphics / imagemap=on;

/* Identify the SOLARPV series and estimate AR(1) parameters */

proc arima data=STSM.SOLARPV
           plots(only)=(series(corr)
                        residual(corr normal));
   identify var=kW_Gen;
   estimate p=(1) method=ML;
quit;
```

8. Submit the code.

The output starts with basic statistics for the time series.

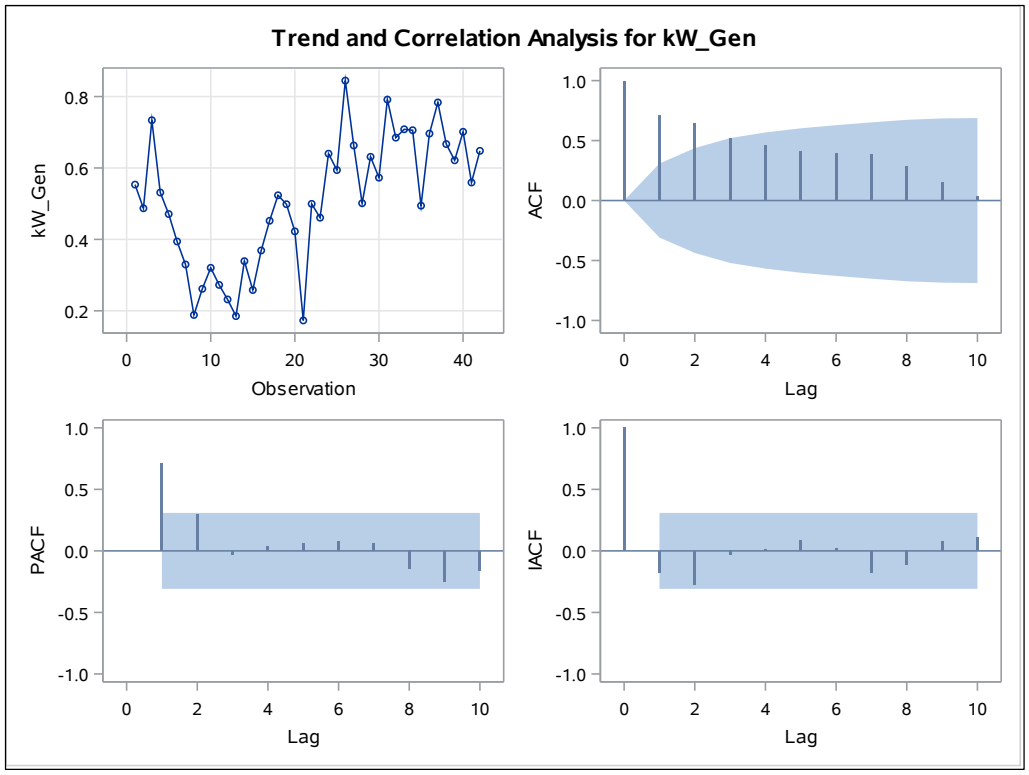
Name of Variable = kW_Gen	
Mean of Working Series	0.511078
Standard Deviation	0.179364
Number of Observations	42

The autocorrelations test for the first six lags shows statistical significance, which means a rejection of the white noise null hypothesis. Notice that the autocorrelation at the first lag is 0.709. That value slowly declines over sequential lags.

Autocorrelation Check for White Noise									
To Lag	Chi-Square	DF	Pr > ChiSq	Autocorrelations					
6	81.65	6	<.0001	0.709	0.648	0.519	0.460	0.412	0.396

These plots were seen in the previous section. The order 1 autoregressive model seems appropriate for these data.

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.



Here the maximum likelihood parameters for the AR(1) model are displayed. The Parameter AR1,1 at Lag 1 is estimated as 0.70389 and that is statistically significant, with a *p*-value less than .0001.

Maximum Likelihood Estimation					
Parameter	Estimate	Standard Error	t Value	Approx Pr > |t|	Lag
MU	0.52019	0.06309	8.25	<.0001	0
AR1,1	0.70389	0.11038	6.38	<.0001	1

The AIC and SBC values are displayed, but recall that these values are useful only when you compare two models of the same data.

Constant Estimate	0.154036
Variance Estimate	0.016529
Std Error Estimate	0.128564
AIC	-50.4857
SBC	-47.0103
Number of Residuals	42

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

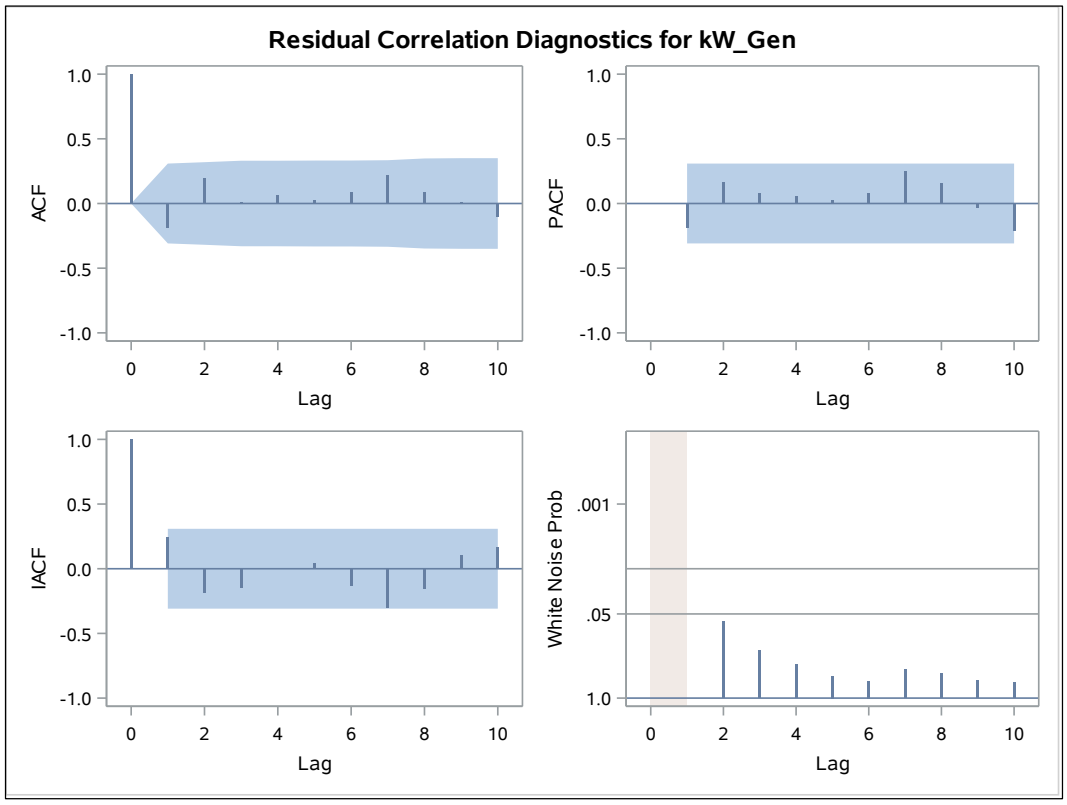
The correlations of parameter estimates can be used to check the multicollinearity of parameters. In this case, the correlation, as expected, is very low at 0.055.

Correlations of Parameter Estimates		
Parameter	MU	AR1,1
MU	1.000	0.055
AR1,1	0.055	1.000

If this model is correct, the residuals from it should be white noise. The autocorrelation check of the first 24 lags indicates no particularly strong autocorrelation, although one value is above 0.2 in absolute value.

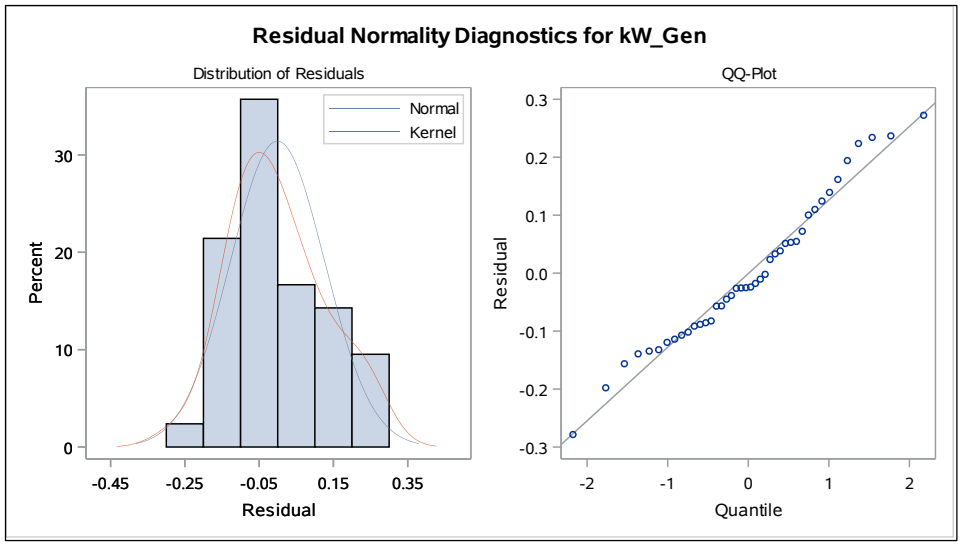
Autocorrelation Check of Residuals									
To Lag	Chi-Square	DF	Pr > ChiSq	Autocorrelations					
6	4.04	5	0.5430	-0.187	0.197	0.011	0.070	0.024	0.089
12	7.81	11	0.7301	0.221	0.090	0.009	-0.103	0.023	-0.039
18	13.25	17	0.7193	0.024	0.071	-0.000	-0.103	-0.027	-0.235
24	15.41	23	0.8792	-0.020	-0.010	-0.066	-0.077	0.043	-0.099

The residual auto-correlation plots, along with the white noise plot of the residuals, confirm that the residual autocorrelations are not statistically significant.



Another white noise assumption is that the residuals are normally distributed. The histogram and QQ plot show the residuals to be reasonably normal (Gaussian normal).

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.



The estimated mean of the model is reiterated and the estimate for the autoregressive parameter is expressed in a particular model formulation.

Model for variable kW_Gen	
Estimated Mean	0.520193

Autoregressive Factors	
Factor 1:	1 - 0.70389 B**(1)

End of Demonstration

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

Exercises

2. Rose Series Estimation

For each rose sales series that showed any autocorrelation, estimate the autoregression parameters of an AR(1) model and look at the residuals.

a. Is the autoregression parameter estimate statistically significant?

b. Do the residuals indicate that the model is sufficient for the series?

End of Exercises

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

2.4 ARMAX and Time Series Regression

Objectives

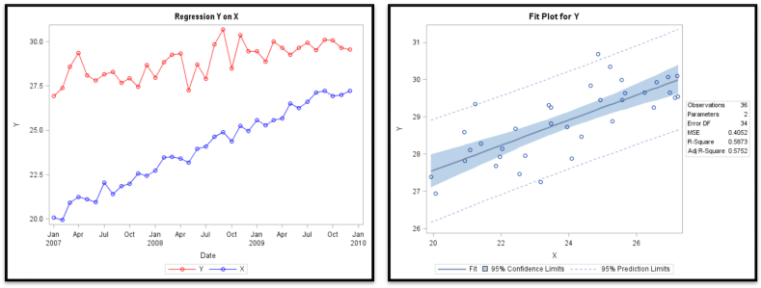
- Explain the X in ARMAX.
- Relate linear regression with time series regression models.
- Examine linear regression assumptions.
- Explain the relationship between ordinary multiple linear regression models and time series regression models.

55

Regression of Y on X

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

* X_t is an external or *exogenous* predictor of Y_t.



56

For a time series, the simplest comparison to ordinary least squares regression can be made with an exogenous (input) variable and a target variable, where the effects are contemporaneous. In other words, a change in the input, X, at time *t* is associated with a change in Y at time *t*. An example of this type of relationship is between the amount of sunshine in a day and the maximum temperature during that day.

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

Multiple Linear Regression

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \varepsilon$$

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.

57

Time Series Regression Terminology

Ordinary Regressor

- an input variable that has only 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

58

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

Types of Regressors: Measurement Scale

Binary (dummy) variables

- takes 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

59

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

60

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

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 by your company (These can be treated as stochastic when decisions are influenced by stochastic factors, such as market share, promotions by competitors, and so on.)

61

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

62

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

2.03 Multiple Answer Poll

Which would be an example of a stochastic regressor?

a. ambient indoor air temperature
b. number of people at a beach
c. occurrence of a full moon
d. occurrence of a solar flare
e. United States prime lending rate
f. your company's mortgage rate for prime customers

63

The Cross-Correlation Function (CCF)

CCF(k) is the cross-correlation of target Y with input X at lag k.

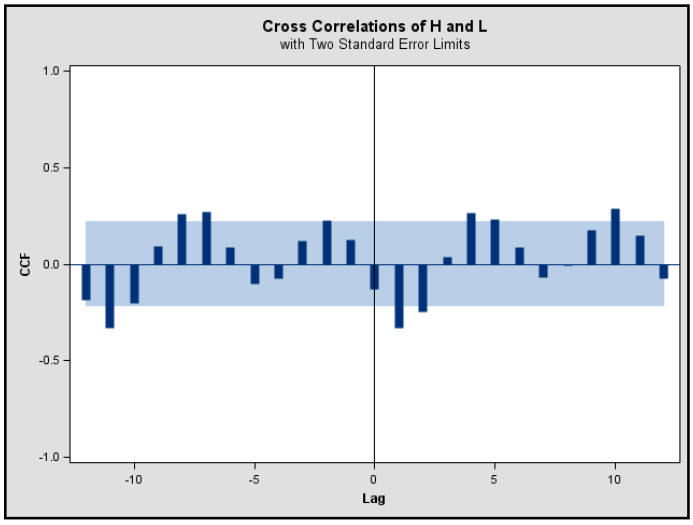
- A significant value at lag k implies that Y_t and X_{t-k} are correlated.
- Spikes and decay patterns in the cross-correlation function can help determine the form of the transfer function.
- The sample CCF estimates an unknown population CCF.

65

continued...

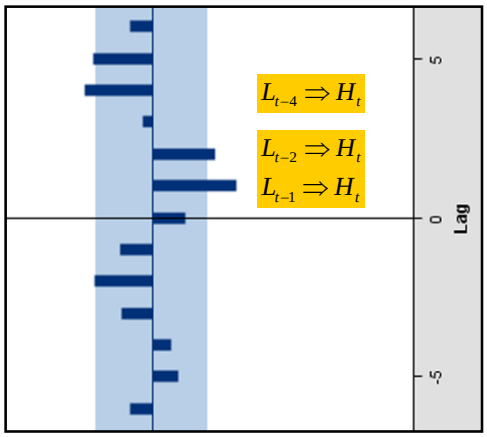
Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

The Cross-Correlation Function (CCF)



66

The Cross-Correlation Function (CCF)

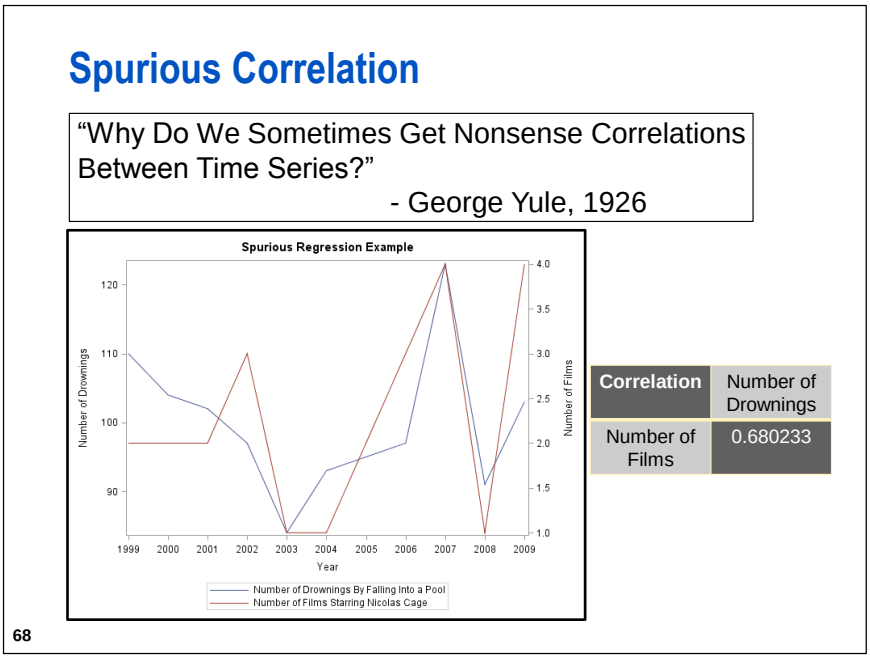


L
Jan
Feb
Mar
Apr
May

H
Jan
Feb
Mar
Apr
May

67

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.



Before you spend too much time with a model looking at the relationship between two variables, think about whether the relationship makes sense. Remember that correlation does not imply causation. A third variable, Z, might be the cause of both X and Y. X and Y might appear correlated only through their relationship with Z. A classic example is of the apparent relationship between the crime rate and purchases of ice cream cones. Perhaps some criminal organization controls the ice cream industry, which causes high crime when there are higher sales and higher profits. More likely, crime is more probable in the warmer months, as is ice cream eating.

If two series follow the same seasonal pattern, they are likely to appear correlated. Unless the trend part of the series is first removed, it is impossible to see what the direct relationship is between such variables.

In the years from 1999 through 2009, the number of deaths in the U.S. by drowning in a pool was correlated with the number of films starring the actor Nicholas Cage. In this example, it would be difficult to draw a conclusion of causation in either direction.

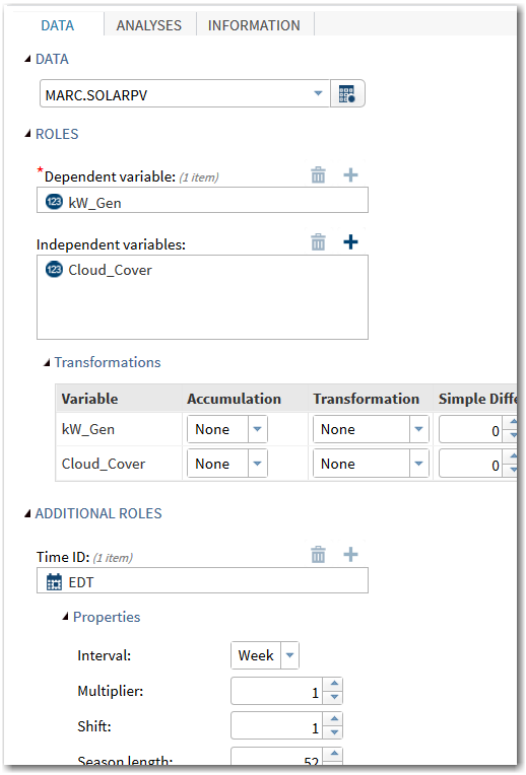
Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

Cloud Cover and Solar Power

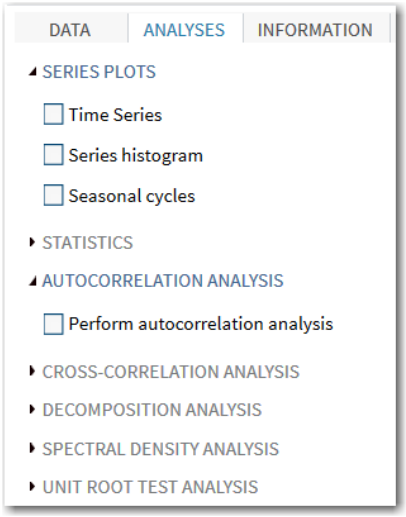
STSM02d04a

Look at the relationship between **kW_Gen** and **Cloud_Cover**.

1. Use the Time Series Exploration task.
2. On the DATA tab, select the data set **SolarPV**. Then, select **kW_Gen** as the dependent variable and **Cloud_Cover** as the independent variable. Expand **ADDITIONAL ROLES** so that you can select **EDT** as the time ID variable.

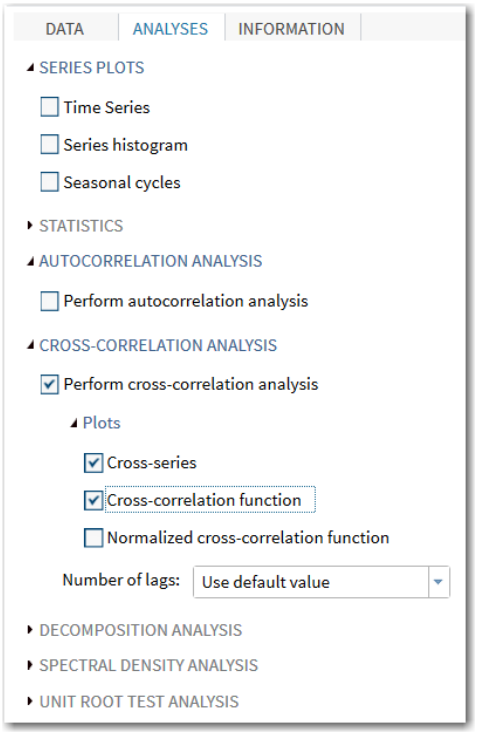


3. On the ANALYSES tab, clear the **Time Series** check box under SERIES PLOTS. Also, clear the **Perform autocorrelation analysis** check box.



Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

4. Expand the **CROSS-CORRELATION ANALYSIS** section and select the **Perform cross-correlation analysis** check box. Plot suboptions appear and the **Cross-series** check box is already selected. Also select the **Cross-correlation function plot** check box.



The generated SAS syntax is shown below.

```
proc timeseries data=WORK.TempSorted seasonality=52
                crossplots=(series ccf);
   id EDT interval=week;
   var kW_Gen / accumulate=none transform=none dif=0 sdif=0;
   crossvar Cloud_Cover / accumulate=none transform=none dif=0 sdif=0;
   ods exclude CCFNORMPlot;
run;
```

Alternatively, you can write the SAS/ETS code directly.

```
/* STSM02d04a.sas */
proc timeseries data=STSM.SOLARPV
                crossplots=(series ccf);
   id EDT interval=week;
   var kW_Gen;
   crossvar Cloud_Cover;
   ods exclude CCFNORMPlot;
run;
```

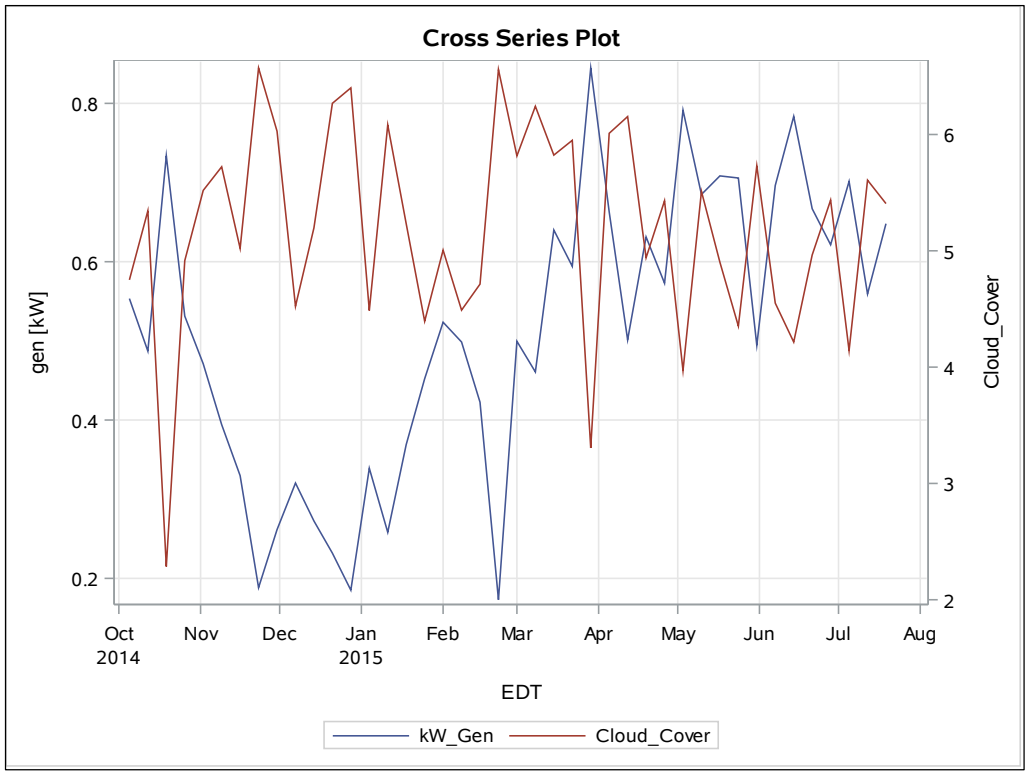
Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

5. Submit the code.

Input Data Set	
Name	WORK.TEMPSORTED
Label	
Time ID Variable	EDT
Time Interval	WEEK
Length of Seasonal Cycle	52

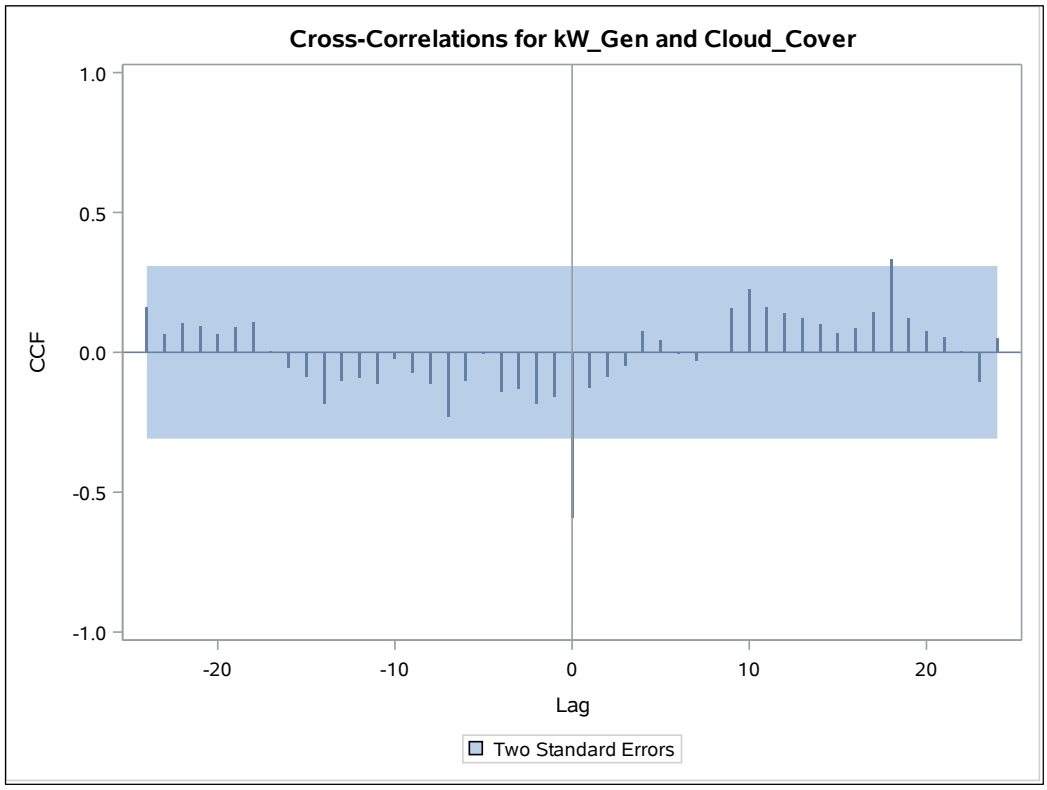
Variable Information	
Name	kW_Gen
Label	gen [kW]
First	Sun, 5 Oct 2014
Last	Sun, 19 Jul 2015
Number of Observations Read	42

The cross series plot seems to show an inverse relationship between cloud cover and generated solar power. Where cloud cover increases, there appears to be a concurrent decrease in the generated power.



Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

The cross-correlation plot shows modest correlations at many positive lags and a large and statistically significant one at the 0 lag, which is the concurrent time. This indicates that **Cloud_Cover** might make an important contribution to modeling and forecasting the kilowatts that are generated.



End of Demonstration

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

Estimation of Cloud Cover

STSM02d04b.sas

Estimate an ARMAX(1,0) model for the **SolarPV** data set with **Cloud_Cover** as an exogenous effect. Check the residual series to see whether it is white noise. Display the goodness-of-fit statistics for comparison with the AR(1) model that was previously estimated.

Use the kilowatts generated (kW_Gen) time series.

1. Create a new Modeling and Forecasting task in SAS Studio.
2. On the DATA tab, select **SolarPV** as the data set. Select **kW_Gen** as the dependent variable
3. Click the triangle next to ADDITIONAL ROLES and then select **EDT** as the time ID and accept the properties that are populated. SAS recognizes **EDT** as weekly.
4. On the MODEL tab, select **ARIMAX** as the forecasting model type. Model settings appear. Select **1** in the **Autoregressive order (p)** field under ARIMA.
5. Under Independent variables, add **Cloud_Cover**.
6. Expand **Plots** and click **Selected plots**.
7. Clear the **Panels of cross-correlations plots** and **Panels of correlation plots** check boxes under Series Plots and the **One-step-ahead and multistep-ahead forecasts** check box under Forecast Plots.
8. On the OPTIONS tab, set the **Number of periods to forecast** field to **0** under FORECAST SETTINGS and clear the **Perform outlier detection** check box next to under OUTLIER DETECTION.

The generated SAS syntax is shown below.

```
proc arima data=WORK.TempSorted plots
           (only)=(residual(corr normal) );
   identify var=kW_Gen crosscorr=(Cloud_Cover);
   estimate p=(1) input=(Cloud_Cover) method=ML;
   forecast lead=0 back=0 alpha=0.05 id=EDT interval=week;
quit;
```

Alternatively, you can write the SAS/ETS code directly.

```
/* STSM02d04b.sas */
proc arima data=STSM.SOLARPV
           plots(only)=(series(corr crosscorr)
                        residual(corr normal));
   identify var=kW_Gen crosscorr=(Cloud_Cover);
   estimate p=(1) input=(Cloud_Cover) method=ML;
quit;
```

9. Submit the code.

   The series identification portion of the output is not shown.

   Both the autoregressive component and the component for **Cloud_Cover** are statistically significant ($p<.0001$).

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

The parameter estimate for **Cloud_Cover** indicates that for each unit increase in cloud cover for a week, the average daily production of solar power decreases by .0096 kilowatts. This value is statistically significant with a *p*-value less than .0001. Cloud cover seems to have a negative effect on solar production, as you likely guessed, and as the overlaid series plots from the previous demonstration imply.

Maximum Likelihood Estimation

Parameter	Estimate	Standard Error	t Value	Approx Pr > |t|	Lag	Variable	Shift
MU	1.00001	0.08901	11.23	<.0001	0	kW_Gen	0
AR1,1	0.86587	0.07766	11.15	<.0001	1	kW_Gen	0
NUM1	-0.09061	0.0096050	-9.43	<.0001	0	Cloud_Cover	0

The AIC for this model is -95, which is lower (more negative) than the AR(1) model (AIC=-50). This model fits better than the AR(1) model.

Constant Estimate	0.134134
Variance Estimate	0.005503
Std Error Estimate	0.074179
AIC	-95.0433
SBC	-89.8303
Number of Residuals	42

Correlations of Parameter Estimates

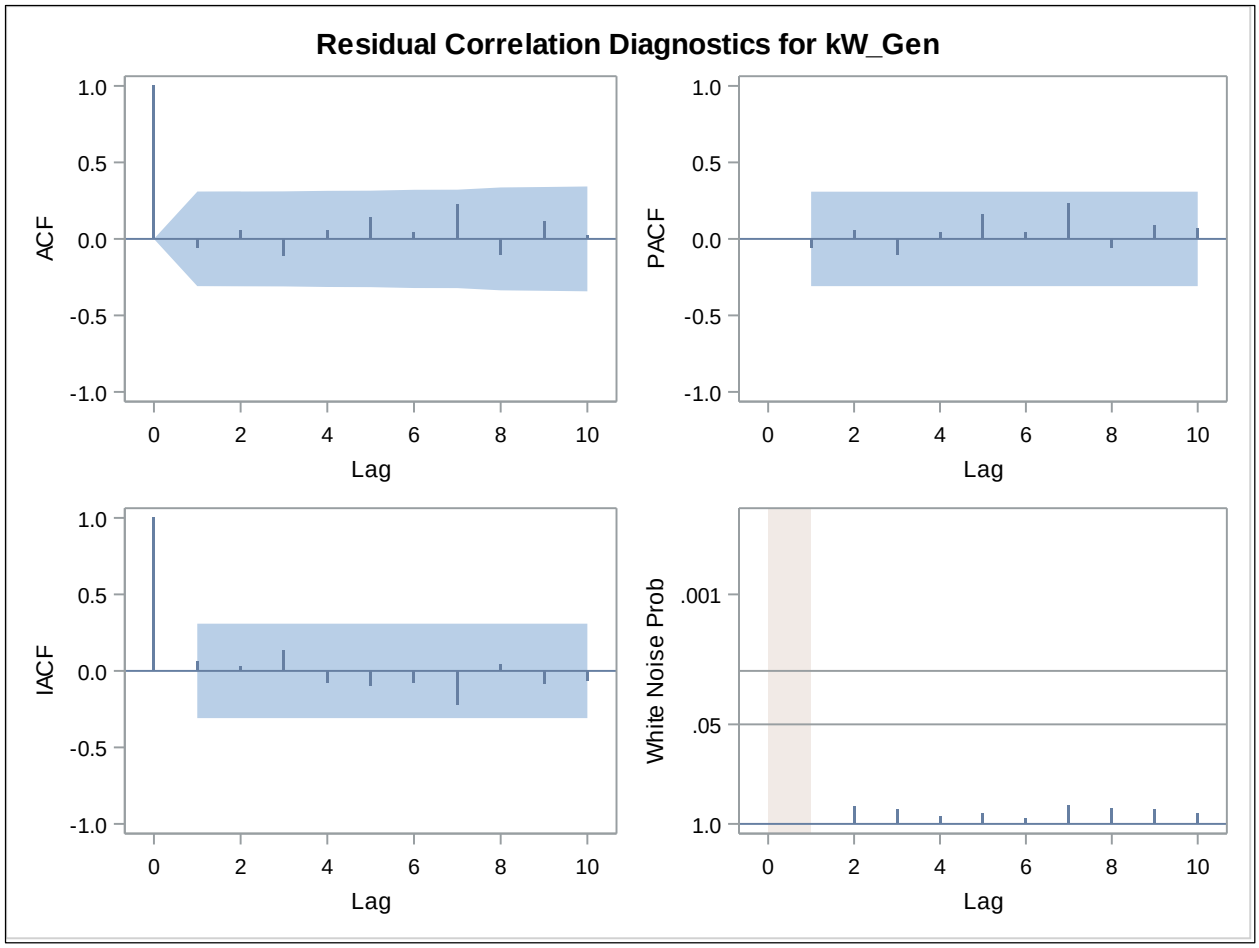
Variable	Parameter	kW_Gen MU	kW_Gen AR1,1	Cloud_Cover NUM1
kW_Gen	MU	1.000	0.103	-0.553
kW_Gen	AR1,1	0.103	1.000	0.033
Cloud_Cover	NUM1	-0.553	0.033	1.000

The residuals seem to be a white noise series.

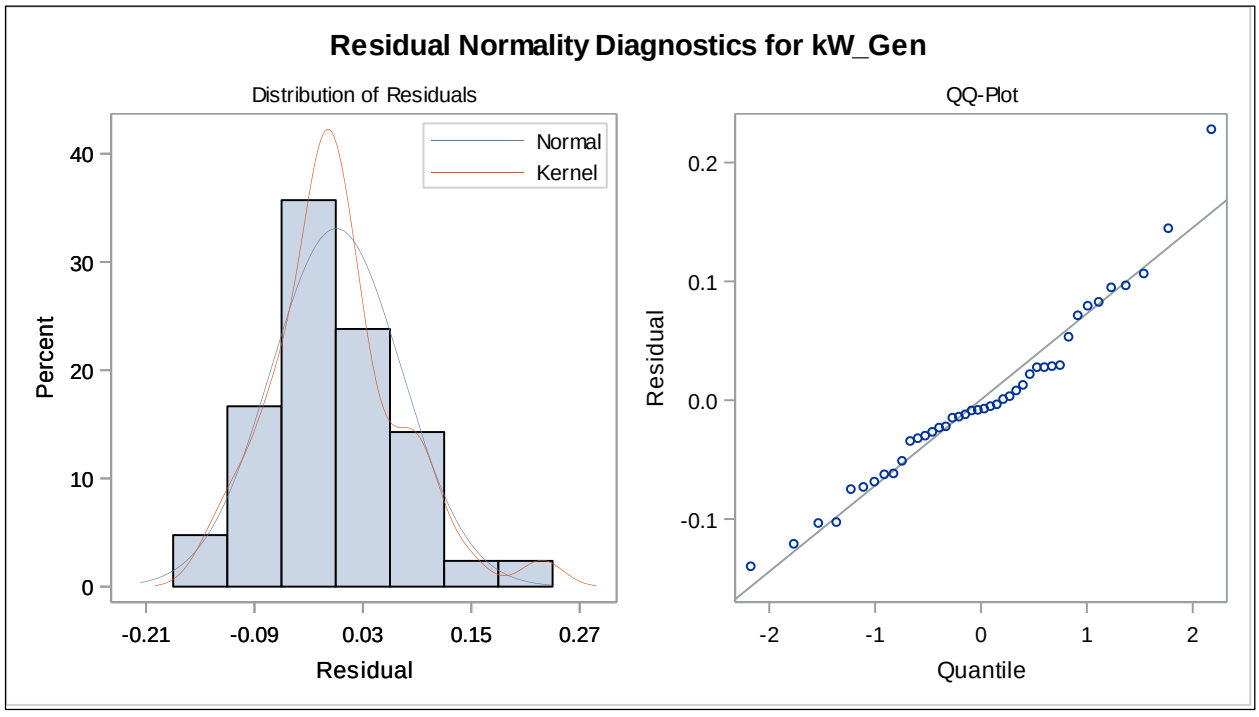
Autocorrelation Check of Residuals

To Lag	Chi-Square	DF	Pr > ChiSq	Autocorrelations					
6	2.08	5	0.8379	-0.058	0.056	-0.109	0.057	0.139	0.043
12	7.16	11	0.7860	0.225	-0.101	0.112	0.024	-0.116	-0.071
18	14.98	17	0.5970	0.029	0.121	-0.006	-0.202	-0.212	0.086
24	16.33	23	0.8406	-0.033	0.030	-0.016	-0.051	-0.088	-0.040

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.



The residuals are relatively normally distributed.



Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

Model for variable kW_Gen	
Estimated Intercept	1.000009

Autoregressive Factors	
Factor 1:	1 - 0.86587 B**(1)

Input Number 1	
Input Variable	Cloud_Cover
Overall Regression Factor	-0.09061

End of Demonstration

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

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

71

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.

72

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

Event and Intervention Analysis Practices

- In retail sales, the term *event* is often used and includes the following:
  - promotional events: discounts, sales, featured displays, and so on
  - advertising events: broadcast, Internet, and print media advertising campaigns, sponsored events, celebrity spokespersons, and so on
- In economics and the social sciences, the term *intervention* is often used and includes these:
  - catastrophic events
  - events related to a key player (CEO, spokesperson): imprisonment, scandal, illness, injury, or death
  - public policy changes

73

Primary Event Variables

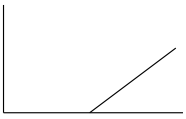
Point/Pulse

$$J_t = \begin{cases} 1 & \text{for } t = t_{\text{event}} \\ 0 & \text{for } t \neq t_{\text{event}} \end{cases}$$

Step

$$I_t = \begin{cases} 1 & \text{for } t \geq t_{\text{event}} \\ 0 & \text{for } t < t_{\text{event}} \end{cases}$$

Ramp



$$R_t = \begin{cases} t - t_{\text{event}} & \text{for } t \geq t_{\text{event}} \\ 0 & \text{for } t < t_{\text{event}} \end{cases}$$

74

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

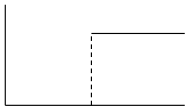
Examples of Input Variables

Point/Pulse

X 0 0 0 1 0 0 0 ....
Y 0 0 0 8 0 0 0
t 1 2 3 4 5 6 7

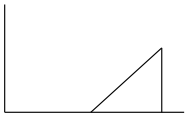
Step

X 0 0 0 1 1 1 1 ....
Y 0 0 0 8 8 8 8
t 1 2 3 4 5 6 7



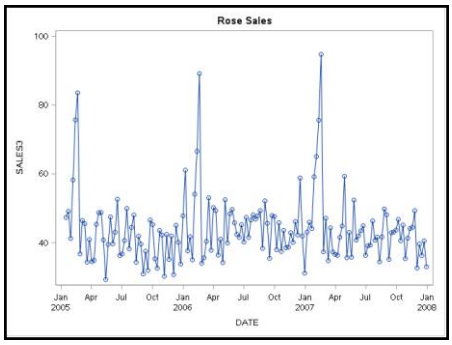
Ramp

X 0 0 0 1 2 3 0 ....
Y 0 0 0 2 4 6 0
t 1 2 3 4 5 6 7



75

Example: Rose Sales



Obs	DATE	SALES3	RAMP
1	Sat, Jan 15, 2005	47.3584	0
2	Sat, Jan 22, 2005	49.1216	0
3	Sat, Jan 29, 2005	41.2812	0
4	Sat, Feb 5, 2005	58.2355	1
5	Sat, Feb 12, 2005	75.6860	2
6	Sat, Feb 19, 2005	83.5646	3
7	Sat, Feb 26, 2005	36.7728	0
8	Sat, Mar 5, 2005	46.4647	0
9	Sat, Mar 12, 2005	45.5764	0
10	Sat, Mar 19, 2005	34.3816	0

76

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

Exercises

3. Intervention Analysis of the Rose Series

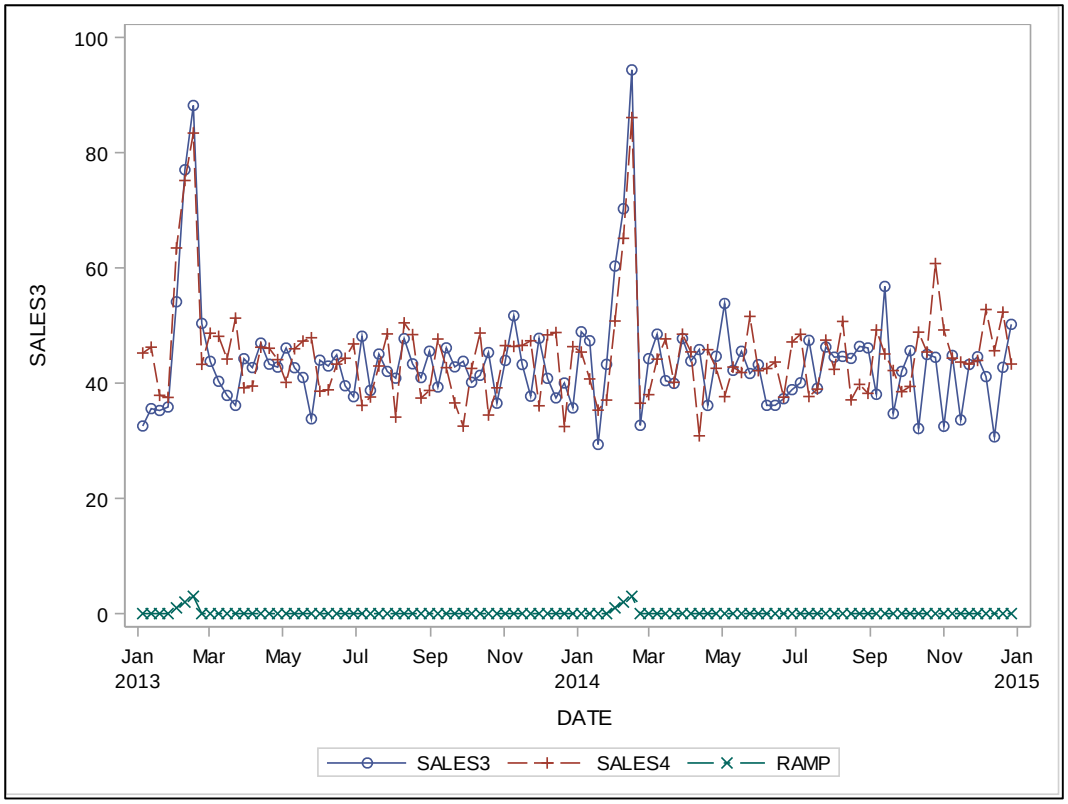
For each rose series (where it is appropriate), use the **Ramp** variable to model the effect of the impending Valentine's Day on weekly rose sales.

a. Open and submit the code in **STSM02e04.sas**.

```
proc print data=stsm.roseseries(obs=12);
   where date >= '01JAN2013'd;
   id date;
   var sales3 sales4 Ramp;
run;

proc sgplot data=stsm.roseseries;
   where date >= '01JAN2013'd;
   series x=date y=sales3 / markers;
   series x=date y=sales4 / markers;
   series x=date y=ramp / markers;
run;
```

DATE	SALES3	SALES4	RAMP
Sat, Jan 5, 2013	32.5574	45.2230	0
Sat, Jan 12, 2013	35.5735	46.2396	0
Sat, Jan 19, 2013	35.2491	37.8820	0
Sat, Jan 26, 2013	35.8502	37.5207	0
Sat, Feb 2, 2013	54.1156	63.4573	1
Sat, Feb 9, 2013	77.0294	75.1663	2
Sat, Feb 16, 2013	88.1882	83.3964	3
Sat, Feb 23, 2013	50.3606	43.2661	0
Sat, Mar 2, 2013	43.7793	48.6604	0
Sat, Mar 9, 2013	40.3168	48.1204	0
Sat, Mar 16, 2013	37.8662	44.1769	0
Sat, Mar 23, 2013	36.1352	51.2801	0

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.



The **Ramp** dummy variable was created to model the seemingly linear increase in sales in the past leading to Valentine's Day. There is no reason to restrict yourself to only a linearly increasing dummy variable. Various shapes of regular impulses can be modeled using dummy variables.

b. For each rose sales series that was not white noise and was not adequately modeled as AR(1) alone, look at the cross-correlation plot with the **RAMP** dummy code series.

Do the series seem to show significant cross-correlation with the RAMP series?

c. For each series that showed any cross-correlation with **Ramp**, estimate the autoregression parameters of an appropriate ARMAX model and look at the residuals.

1) Is the autoregression parameter estimate statistically significant?
2) Is the cross-correlation parameter estimate statistically significant?
3) Do the residuals indicate that the model is sufficient for the series?

End of Exercises

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

2.5 Forecasting and Accuracy Assessment

Objectives

- Use a holdout sample to validate a model.
- Use error measures to evaluate forecast accuracy.
- Use sample time series data to exemplify forecasting concepts.

79

Forecasting

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

If someone asks you whether you can forecast something ***accurately***, you cannot answer until you establish what accuracy means and until you perform preliminary modeling of the data.

80

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

Liability

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

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

- You might have no control over data accuracy and validity.
- Is modeling the volume sold a true reflection of real demand?
  - Were there supply shortages that you were not aware of that could hurt forecasts?

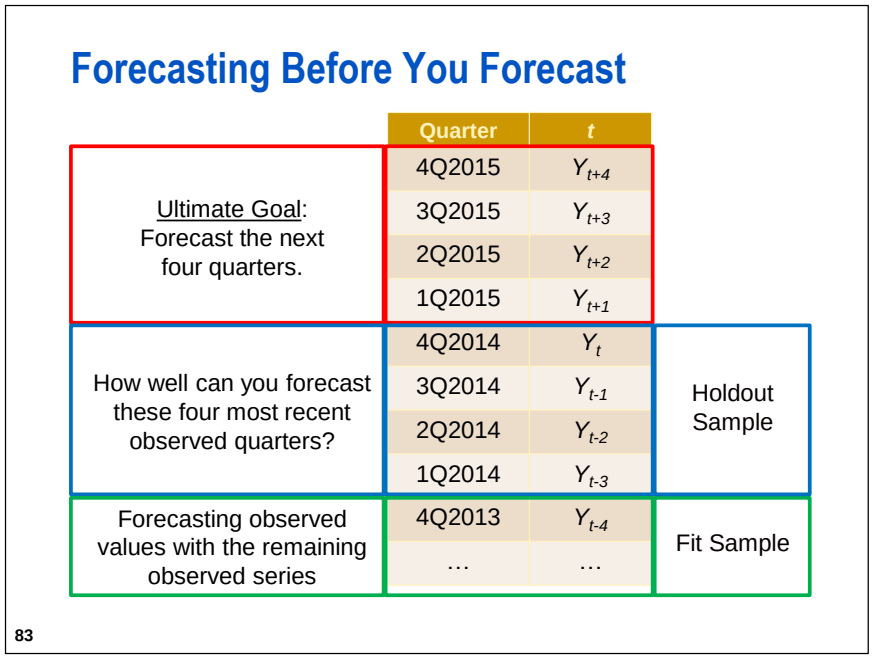
81

Liability

- You need to assume that the underlying future behavior remains consistent with past behavior.
- However, you have no control over future events that might affect future behavior, such as catastrophes, economic downturns, war, the integrity of key players, the survival of key players, and so on.

82

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.



Time series analysis, similar to other branches in statistics, can be grouped into two broad segments: inference-based analysis and prediction analysis. For those who are intent on forecasting future, unobserved periods, it is a best practice to split the data set into a fit sample and a holdout sample. However, unlike other forms of predictive modeling, where the holdout sample is a random sub-sample from the original sample, the holdout sample in time series forecasting is the final k values of the series. You are simulating a scenario, k time periods before the last measurement, when you could be trying to forecast the next k values in the series. However, now you know what those last k values are and you can see how accurately you forecasted them.

The fit sample is used to derive the forecast model, and the holdout sample is used to evaluate how well the forecast model predicts the most recent *n* observations. The following slides discuss the process in more detail and provide rules of thumb for selecting the holdout sample.

For an overview of the entire ARMA and ARMAX modeling process, including how the fit and holdout samples are used, refer to the process flow chart at the end of this chapter.

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

Honest Assessment: 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 for each model by forecasting the holdout sample.
4. Choose the model with the best accuracy statistic.

84

Choosing the Holdout Sample

- 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.

85

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

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.

86

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 more than 25% of the series.

87

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

Model Fit Statistics

Mean Absolute Percent Error:

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

Mean Absolute Error:

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

88

continued...

Both MAPE and MAE are very common measurements of model fit. When you choose between candidate models, the model with the ***lowest*** value for MAPE or for MAE is the model that fits the holdout data the best.

Notice that both statistics take the absolute value of the observed values minus the predicted values. Mathematically, this is done by necessity, but might omit important information about the fit of the model. For example, one model might have a very low MAPE or MAE, but it constantly under fits. (That is, the predicted values always fall just short of the observed values in the holdout data set.) Looking at MAPE or MAE alone does not always paint the whole picture. Instead, look at the value of MAPE in conjunction with a plot of observed and predicted values in the holdout sample to assess the model fit.

Model Fit 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 Square Error:

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

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

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

89

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

Model Selection and Deployment

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 for each model by forecasting the holdout sample.
4. Choose the model with the best accuracy statistic.
5. Using the best model, generate forecasts for *n* future periods.

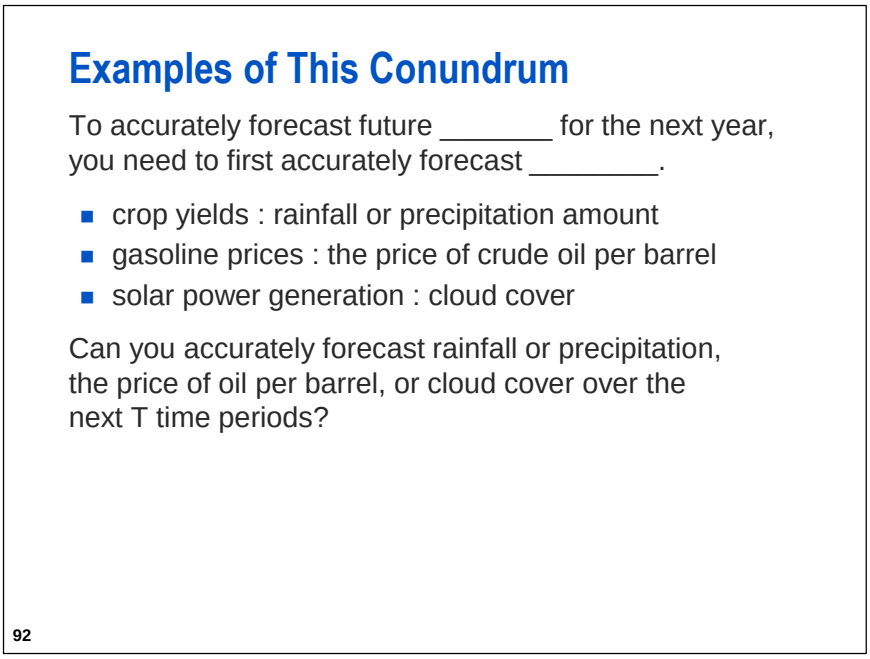
90

The Stochastic Input Variable Conundrum

- Future values of the input variable are either of the following:
  - deterministic (known)
  - stochastic (unknown, and therefore, estimated)
- A stochastic input, X_t, must be forecast for *T* periods so that Y_t can be forecast for *T* periods.
- The forecast accuracy of Y_t depends, in part, on the forecast accuracy of the stochastic input variable.

91

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.



Inaccurate forecasts of future periods of a stochastic input variable produce unreliable forecasts of the analysis series. The question is how many periods into the future you can accurately forecast the stochastic input variable. That answer varies significantly, depending on the data with which you are working.

What do you do when you are required to generate forecasts for Y_t for a longer time horizon than X_t can be accurately forecast? Suppose you are required to forecast crop yields for the next four weeks, but your rainfall or precipitation forecasts (used to forecast crop yields) are accurate for only two weeks. This might be an instance where scenario analysis can add value to the overall forecasting process.

The upcoming slide discusses *scenario analysis*, also called *what-if analysis*. This analysis enables the analyst to produce a number of different forecasts for Y_t, conditioned on different values of the input variable X_t.

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

Scenario Analysis / What-if Analysis

- Choose future values of the stochastic input variable to generate different forecasts for Y_t.
- Run the same model, and replace the chosen future values each time.

  This reduces a complex process into a series of simple Boolean conditional statements.
  - For example, for period t+2:
    - if $X_{t+2} = X_1$, then $Y_{t+2} = Y_1$
    - if $X_{t+2} = X_2$, then $Y_{t+2} = Y_2$
    - …
    - if $X_{t+2} = X_k$, then $Y_{t+2} = Y_k$
  - For k chosen future values of X_{t+2}

93

Example: Forecasting Retail Fuel Prices

Suppose you are tasked with forecasting the average retail price of gasoline (Y_t) given the cost of oil per barrel (X_t). You went through the appropriate steps to build a model, and are ready to forecast future, unknown periods using an ARMAX(1,0).

You are asked to deliver a forecast for the retail price of gasoline for the next two periods (Y_{t+1}, Y_{t+2}). In order to deliver reliable forecasts for Y_{t+1} and Y_{t+2}, you need reliable forecasts for X_{t+1} and X_{t+2}. Suppose your model for the cost of oil per barrel is deemed accurate and reliable for only one future time period (X_{t+1}). You need a value, or values, for X_{t+2} now so that you can produce a forecast for Y_{t+2}. This is where scenario analysis can be used effectively.

Scenario Analysis

Suppose the cost of oil per barrel (X_t) is \$85 at period t. The forecast for the cost of oil per barrel in one future time period (X_{t+1}) is \$88. Running the ARMAX(1,0) model and forecasting one period ahead produces a forecast for Y_{t+1}. Because X_{t+2} cannot be accurately forecast, different scenarios can be run in its place.

The chart below runs five different scenarios for X_{t+2}. Based on the forecast from X_{t+1}, the scenarios for X_{t+2} were whether the cost of oil per barrel does the following:

- drops by \$4 from t+1 to t+2
- drops by \$2 from t+1 to t+2
- stays the same from t+1 to t+2
- rises by \$2 from t+1 to t+2
- rises by \$4 from t+1 to t+2

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

Period	X	Y
t+1	$ 88	Y_{t+1} forecast
t+2	$ 84	Y_{t+2} forecast if X drops $4 from t+1 to t+2
	$ 86	Y_{t+2} forecast if X drops $2 from t+1 to t+2
	$ 88	Y_{t+2} forecast if X remains the same from t+1 to t+2
	$ 90	Y_{t+2} forecast if X increases $2 from t+1 to t+2
	$ 92	Y_{t+2} forecast if X increases $4 from t+1 to t+2

Choosing the number of scenarios and the values for each scenario is dependent on the variability of the specific stochastic input variable across time periods. The analyst must use his or her industry expertise and judgment to make those decisions. The example above is for illustrative purposes. It could be altered to include more or fewer values, but the underlying process remains consistent.

Choosing a Winning Set of Forecasts

Good forecasts should

- be highly correlated with the actual series values
- exhibit small forecast errors
- capture the prominent features of the original time series.

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

94

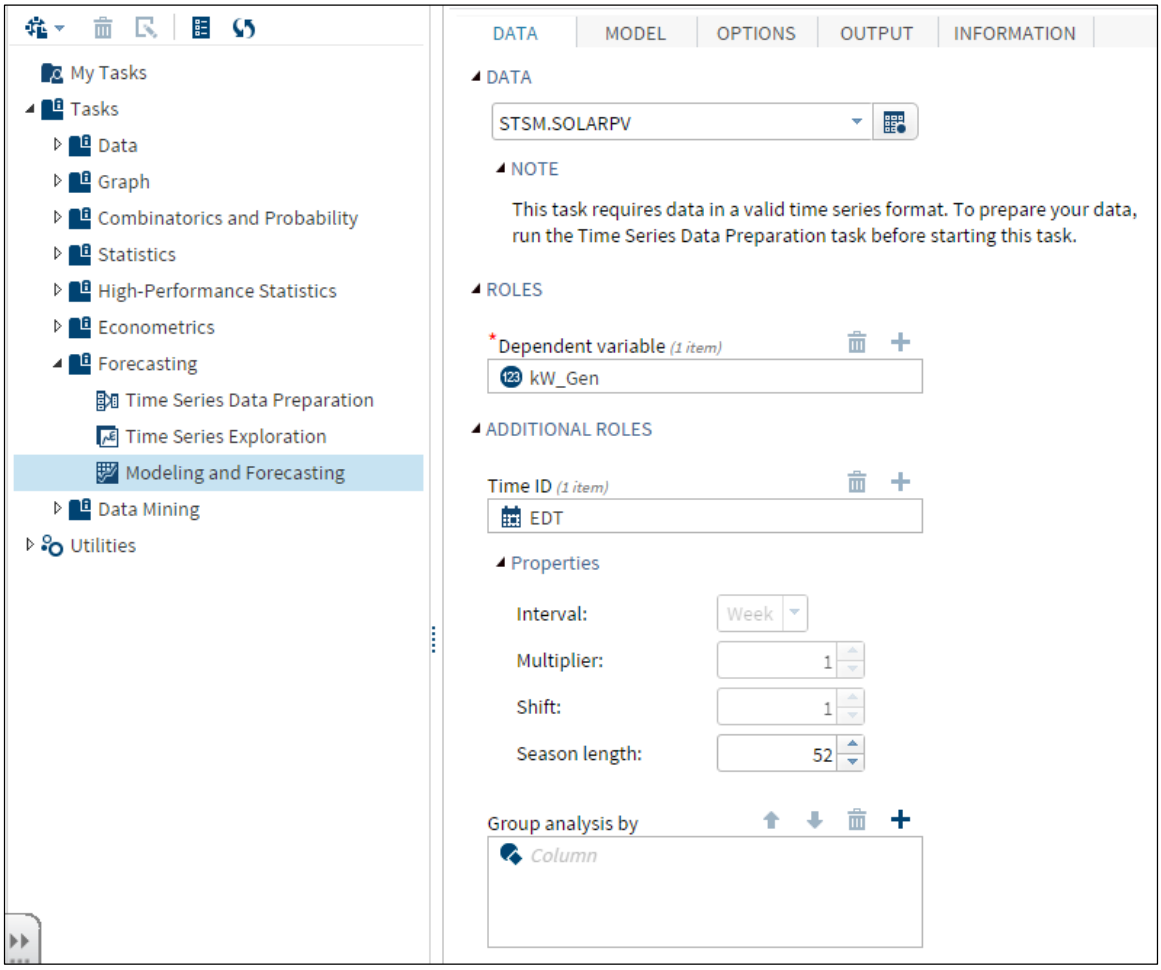
Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

Forecasting a Holdout Sample Using the ARIMA Model

STSM02d05a

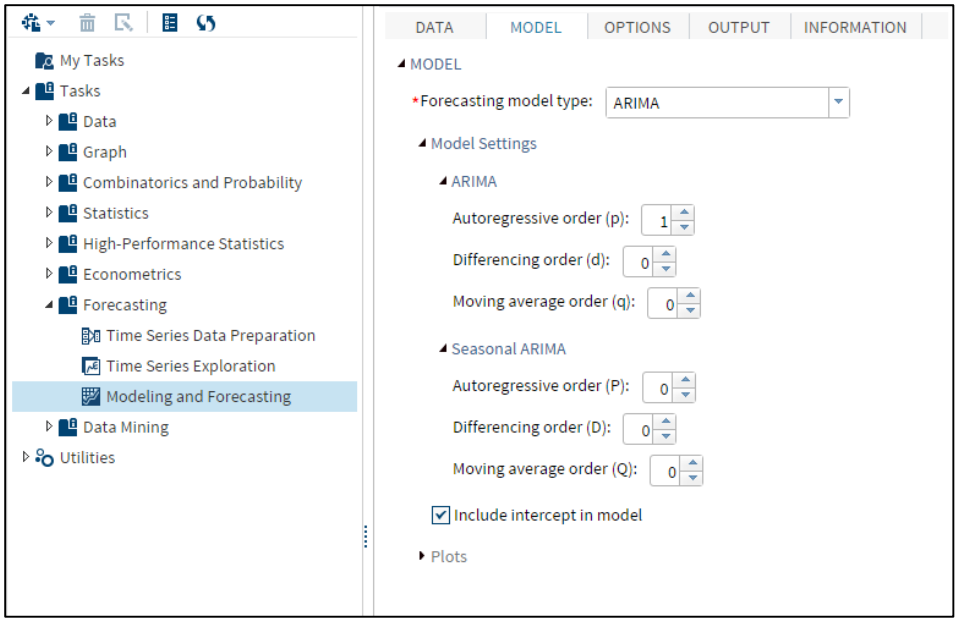
The first of two models to be built is the ARMA(1,0) model excluding the **Cloud_Cover** input variable.

1. Under the Modeling and Forecasting task on the DATA tab, specify the **STSM.SOLARPV** data set.
2. Specify **kW_Gen** as the dependent variable and **EDT** as the time ID variable.



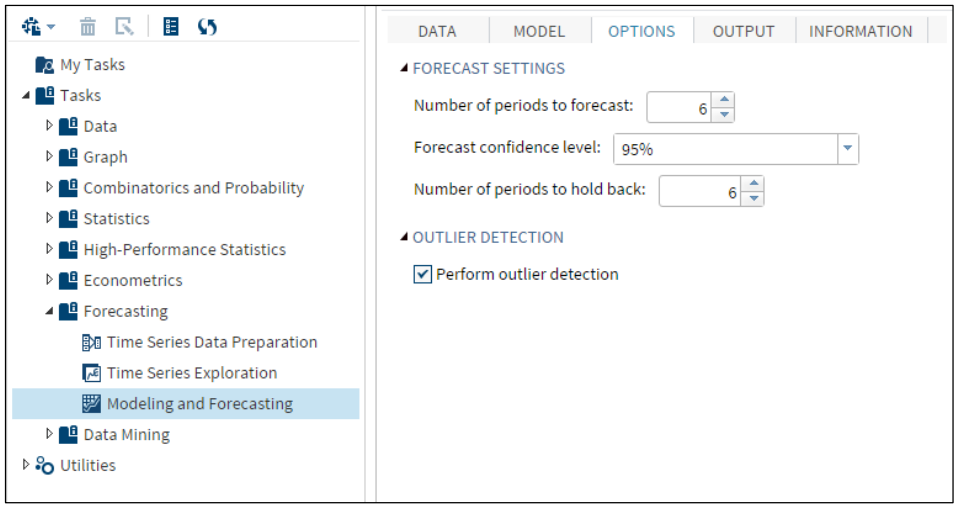
3. On the MODEL tab, specify an **ARIMA** model of Autoregressive order **1**.
4. Change Default plots to **Selected plots**.
5. At the bottom under Forecast Plots, make sure that both check boxes are selected. This provides two different forecast plots for the ARMA(1,0) model. They are important when you compare them with the ARMAX(1,0) model.
6. Clear the check boxes for all other plots. (You saw them previously.)

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.



7. On the OPTIONS tab, request to forecast six periods as well as to hold back six periods. This builds the ARMA(1,0) model on the fit sample (that is, all observations except the most recent six periods) and forecasts the holdout sample (that is, the most recent six periods).

In practice, the holdout sample should not be used to build the model if the goal is predicting future, unobserved periods. The fit sample should be used to build the model and then tested on the holdout sample. Refer to the ARMA and ARMAX process flow chart for details.



The SAS Studio generated code is shown below.

```
proc arima data=WORK.TempSorted plots
           (only)=(series(corr crosscorr) residual(corr normal)
                   forecast(forecast forecastonly));
   identify var=kW_Gen;
   estimate p=(1) method=ML;
   forecast lead=6 back=6 alpha=0.05 id=EDT interval=week;
   outlier;
quit;
```

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

Alternatively, you can write the program directly as shown below.

```
/* STSM02d05.sas */
/* Part a: ARMA(1,0) Forecasting the holdout sample */
proc arima data=STSM.SOLARPV
           plots(only)=forecast(forecast forecastonly);
   identify var=kW_Gen;
   estimate p=(1) method=ML;
   forecast lead=6 back=6 id=EDT;
   outlier;
quit;
```

8. Run the program.

After the results are generated, much of it should look familiar from earlier in the chapter.

To minimize redundancy, the output generated in earlier parts of this chapter is not printed. It is already confirmed that an ARMA(1,0) is a good candidate model for modeling **kW_Gen**. The point of focus now is shifted onto how well the ARMA(1,0) model forecasts the holdout sample, so only that output is printed here.

The first two tables provide the estimated mean and autoregressive factor parameter estimates. Given these two pieces of information, the intercept can be calculated and the model can be written.

Model for variable kW_Gen	
Estimated Mean	0.520193

Autoregressive Factors	
Factor 1:	1 - 0.70389 B**(1)

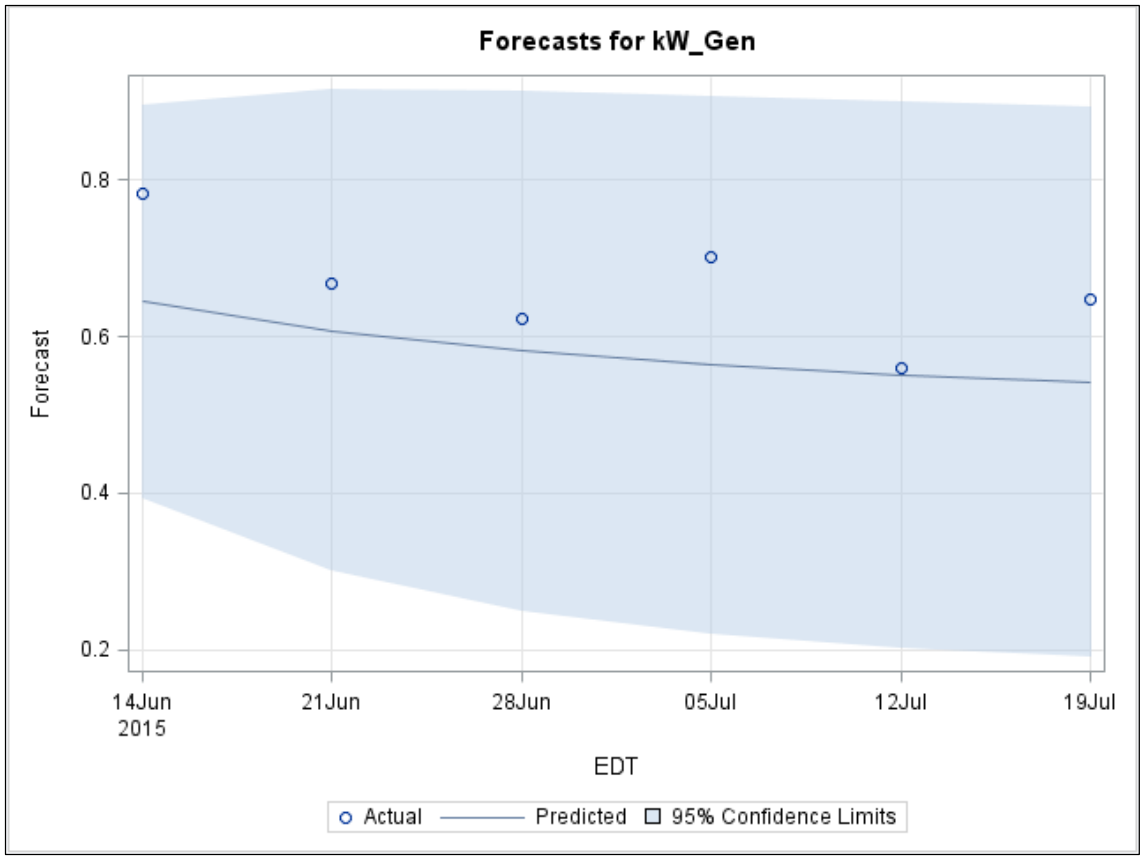
The Forecasts table provides forecasts, standard errors, 95% confidence limits, actual values, and residual values for the six variables in the holdout data set.

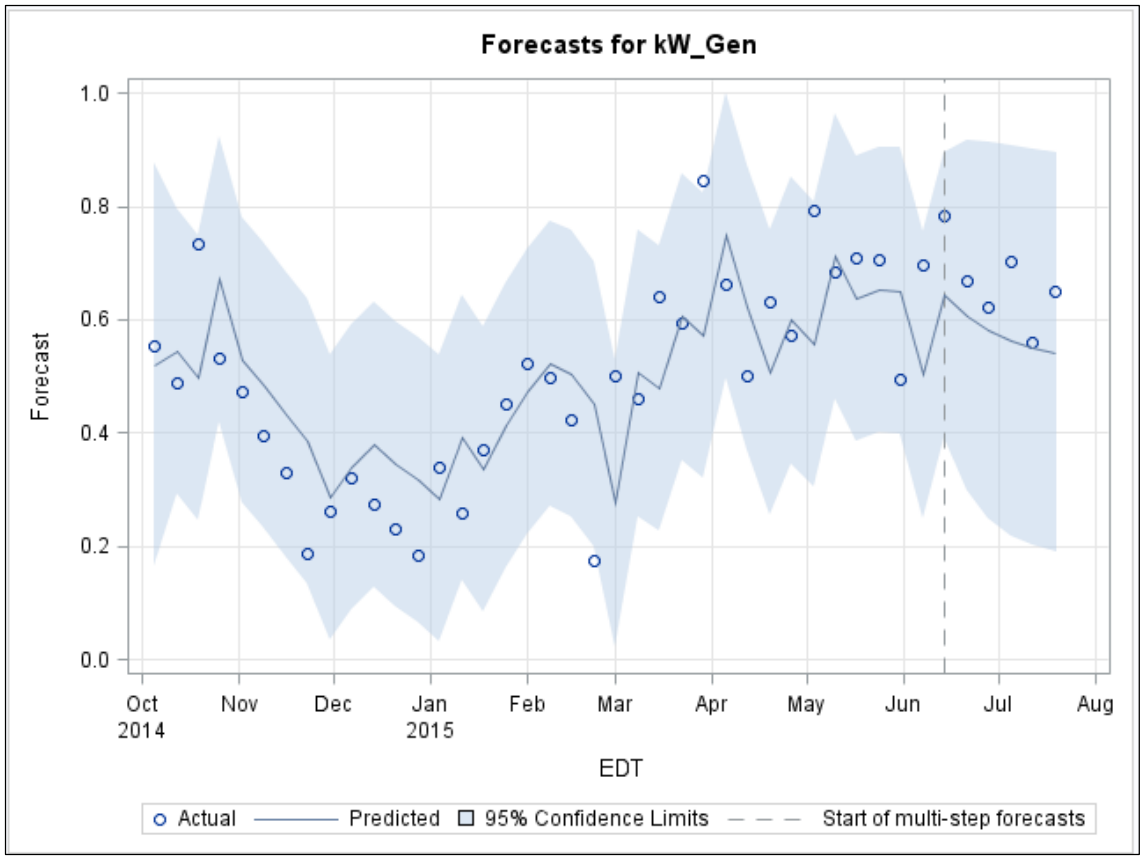
Only the holdout sample (observations 1 through 36) is displayed in the Forecasts table.

Forecasts for variable kW_Gen						
Obs	Forecast	Std Error	95% Confidence Limits		Actual	Residual
37	0.6442	0.1286	0.3922	0.8961	0.7835	0.1394
38	0.6075	0.1572	0.2993	0.9156	0.6669	0.0595
39	0.5816	0.1696	0.2491	0.9141	0.6214	0.0398
40	0.5634	0.1755	0.2195	0.9073	0.7014	0.1379
41	0.5506	0.1783	0.2012	0.9000	0.5593	0.0087
42	0.5416	0.1797	0.1895	0.8937	0.6480	0.1064

The two plots show the forecasts plotted alone as well as plotted with the rest of the series. A quick glance at both plots shows that the forecasts tend to move in the general direction of the observed values, but are under forecasting each time. The model did not seem to capture the increase from June 28 to July 5, the decrease from July 5 to July 12, or the increase from July 12 to July 19.

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.





Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

The Outlier Detection Summary and Outlier Details tables suggest that observation 21 is an outlier.

Outlier Detection Summary	
Maximum number searched	1
Number found	1
Significance used	0.05

Outlier Details				
Obs	Type	Estimate	Chi-Square	Approx Prob>ChiSq
21	Additive	-0.29146	6.85	0.0089

Now that the ARMA(1,0) model is built, it is time to build the ARMAX(1,0) model and compare which model best forecasts the holdout sample. Recall that the ARMAX(1,0) model includes **Cloud_Cover** as the input variable, which was deemed a significant predictor in a previous section.

End of Demonstration

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

Forecasting a Holdout Sample Using the ARIMAX Model

STSM02d05b

The second of two models to be built is the ARMAX(1,0) model including the **Cloud_Cover** input variable.

1. On the DATA tab in the Modeling and Forecasting task, choose the same options as before. The data set, dependent variable, and time ID variable are the same from the ARMA(1,0) model.
2. On the MODEL tab, change the model type to **ARIMAX**, autoregressive order to **1**, and include **Cloud_Cover** as the independent variable.
3. As before, under **Plots**, click **Selected Plots** and select both **Forecast Plots** check boxes and clear all other plot check boxes.
4. As before, forecast six periods ahead while holding back six periods. This uses the ARMAX(1,0) model to forecast the six-period holdout sample.
5. You can choose whether to perform outlier detection. By default, outlier detection is performed.

The code generated by SAS Studio is as follows:

```
proc arima data=WORK.TempSorted plots
           (only)=(forecast(forecast forecastonly));
   identify var=kW_Gen crosscorr=(Cloud_Cover);
   estimate p=(1) input=(Cloud_Cover) method=ML;
   forecast lead=6 back=6 alpha=0.05 id=EDT interval=week printall;
   outlier;
quit;
```

Alternatively, you can write the SAS code directly as shown below.

```
/* STSM02d05.sas */
/* Part b: ARMAX(1,0) Forecasting the holdout sample */
proc arima data=STSM.SOLARPV
           plots(only)=forecast(forecast forecastonly);
   identify var=kW_Gen crosscorr=(Cloud_Cover);
   estimate p=(1) input=(Cloud_Cover) method=ML;
   forecast lead=6 back=6 id=EDT;
   outlier;
quit;
```

6. Submit the program.

   Moving to the bottom of the Results tab, the estimated intercept, autoregressive parameter, and input variable parameter estimate are given. Given these, the model can be derived.

Model for variable kW_Gen	
Estimated Intercept	1.000009

Autoregressive Factors	
Factor 1:	1 - 0.86587 B**(1)

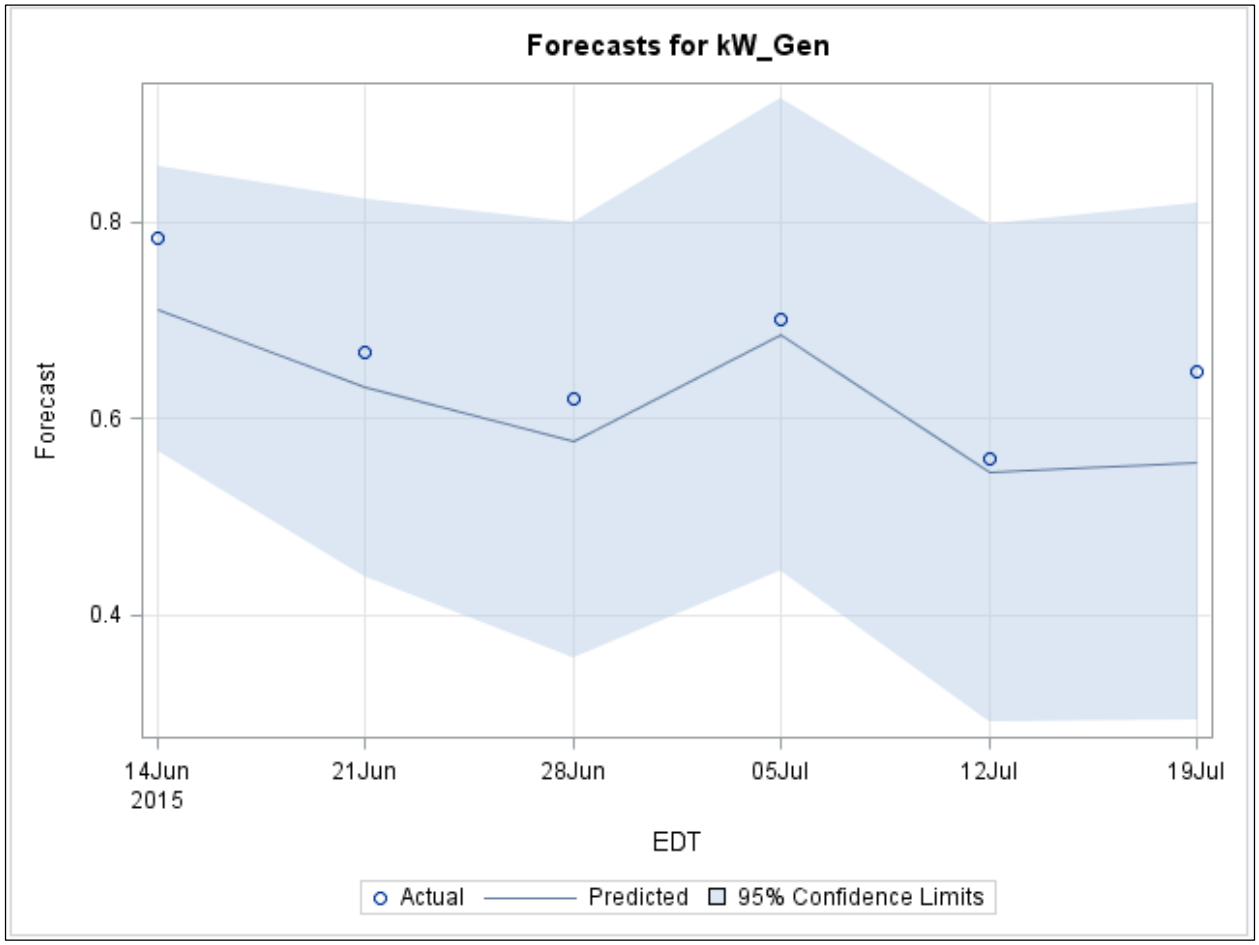
Input Number 1	
Input Variable	Cloud_Cover
Overall Regression Factor	-0.09061

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

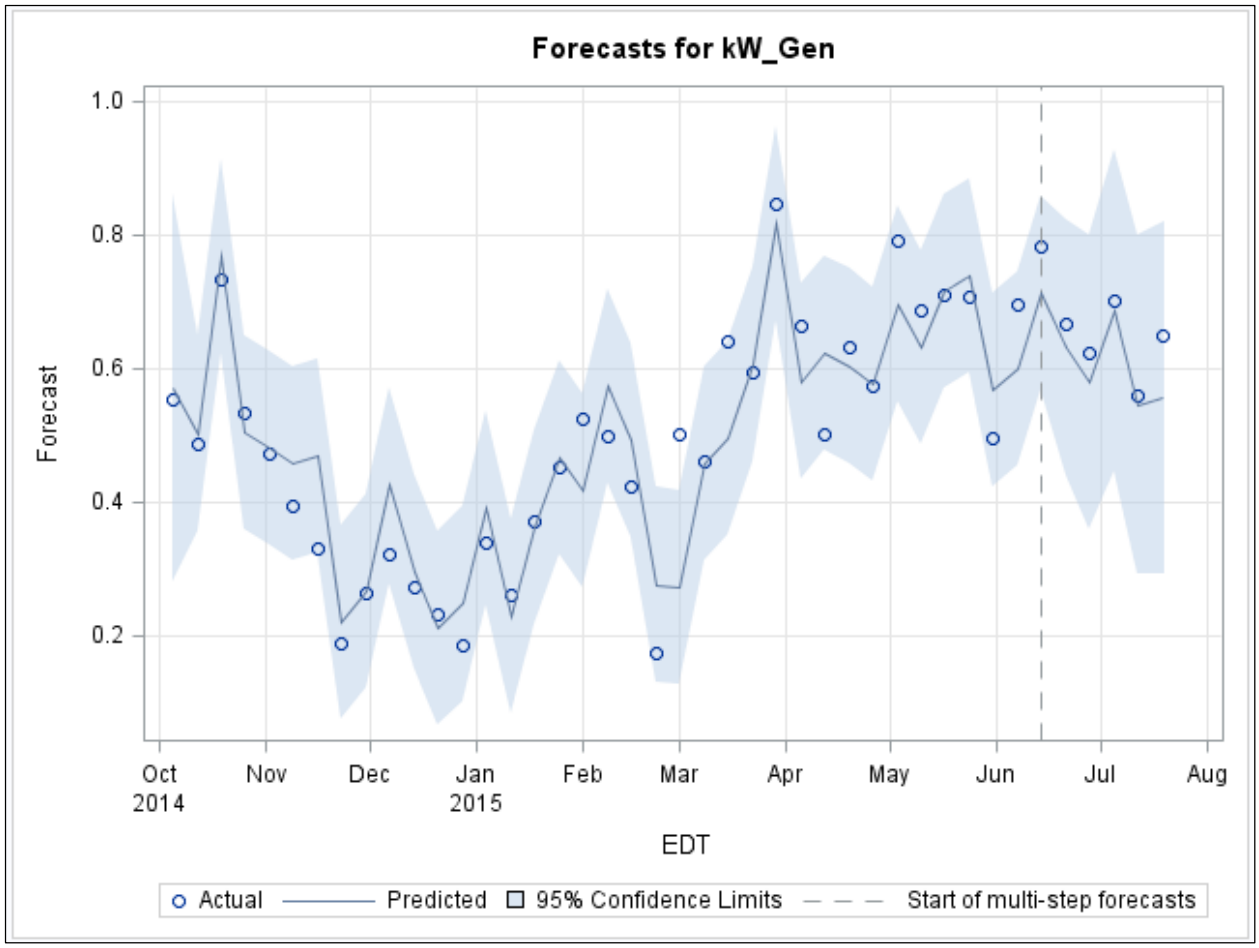
Forecasts for all 42 observations are given in the next table. The table also lists standard errors, 95% confidence limits, actual values, and residuals.

Forecasts for variable kW_Gen						
Obs	Forecast	Std Error	95% Confidence Limits		Actual	Residual
37	0.7121	0.0742	0.5667	0.8575	0.7835	0.0715
38	0.6316	0.0981	0.4393	0.8240	0.6669	0.0353
39	0.5780	0.1128	0.3569	0.7991	0.6214	0.0434
40	0.6857	0.1226	0.4454	0.9261	0.7014	0.0156
41	0.5448	0.1295	0.2909	0.7987	0.5593	0.0145
42	0.5559	0.1345	0.2924	0.8195	0.6480	0.0921

The two forecast plots are listed next. A quick glance seems to favor the ARMAX(1,0) model, because it seems to forecast the holdout sample more accurately. However, looking at the two MAPE calculations can help determine which model is most accurate.



Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.



End of Demonstration

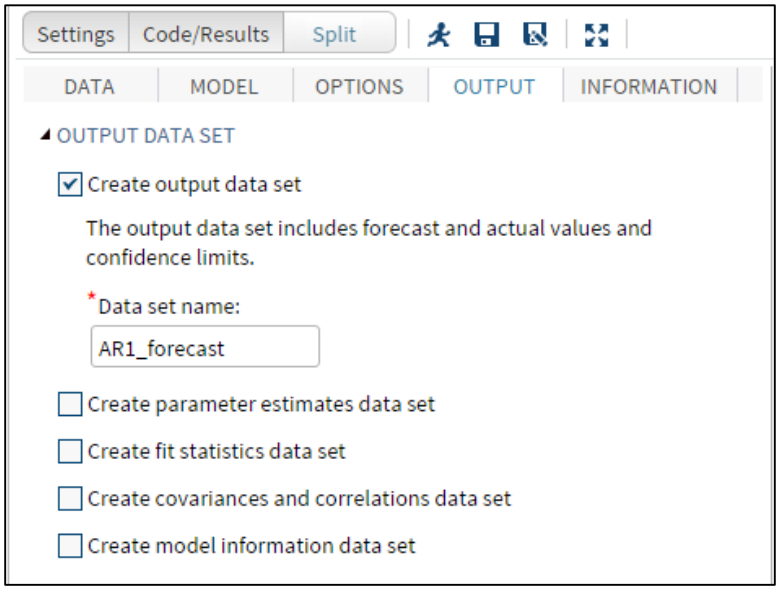
Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

Comparing Models Using MAPE

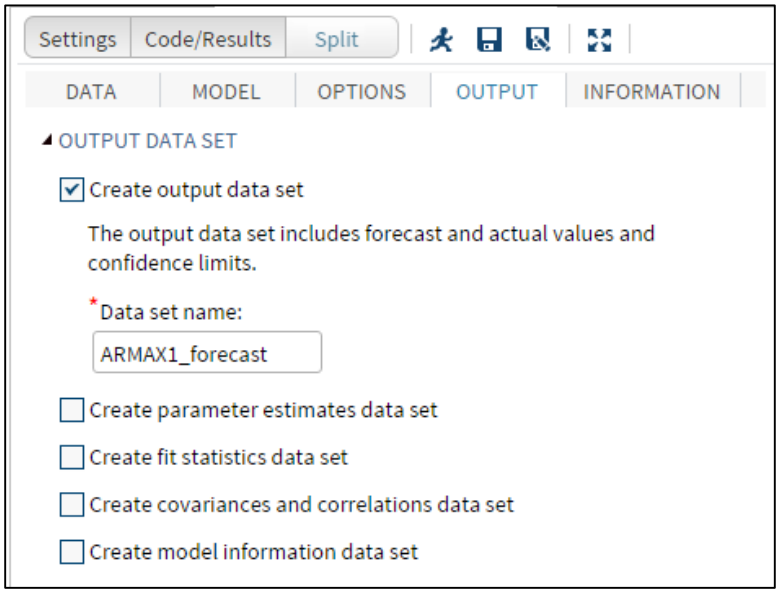
STSM02d05c

To determine with which candidate model to move forward, MAPE is chosen as the statistic of choice to assess the models. It is at the discretion of the analyst to choose which statistic to use to assess candidate models. MAPE was chosen for this demonstration.

1. When you build the ARMA(1,0) model, on the Output tab, create an output data set called **AR1_forecast**. This writes an output data set with the aforementioned name to the **Work** library.



2. When you build the ARMAX(1,0) model, on the Output tab, create an output data set called **ARMAX1_forecast**. Like the **AR1_forecast** data set, **ARMAX1_forecast** is also written to the **Work** library.



Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

Alternatively, write the following SAS code:

```
/* STSM02d05.sas */
/* Part c: Calculating MAPE for each of the above models */
proc arima data=STSM.SOLARPV
           plots(only)=forecast(forecast forecastonly)
           out=AR1_forecast;
   identify var=kW_Gen;
   estimate p=(1) method=ML;
   forecast lead=6 back=6 id=EDT;
   outlier;
quit;
```

```
proc arima data=STSM.SOLARPV
           plots(only)=forecast(forecast forecastonly)
           out=ARMAX1_forecast;
   identify var=kW_Gen crosscorr=(Cloud_Cover);
   estimate p=(1) input=(Cloud_Cover) method=ML;
   forecast lead=6 back=6 id=EDT;
   outlier;
quit;
```

3. Open the file, **STSM02d05a.sas**.
4. Find and submit the %INCLUDE statement.

The INCLUDE statement is used to run the macro code in the file, **%MAPEMacros.sas**.

```
%include "&programloc\MAPEMacros.sas";
```

5. Then, use the macros with either **%MAPE** (a macro using PROC SQL code) or **%MAPE_D** (a macro using DATA step code). The output is presented differently with each macro.

The **%MAPE** macro requires the following four arguments:

OUTPUTDSN the name of the data set output from the SAS/ETS forecasting procedure

TIMEID the name of the time variable in the series

SERIES the name of the target series

NUMHOLDOUT the number of time points for the holdout sample

```
/* MAPE macro using PROC SQL */

%macro mape(outputdsn,timeid,series,numholdout);
proc sql noprint;

   select &timeid format=DATE9. into:cutoffdate
      from &outputdsn.
      having monotonic(&timeid)=max(monotonic(&timeid))-&numholdout;

   create table work.%scan(&outputdsn,1,'_')_mape as
      select sum((abs((&series.-forecast)/&series.))/&numholdout)
      as %upcase(%scan(&outputdsn,1,'_'))_MAPE
      from &outputdsn.
      where &timeid.>"&cutoffdate"d
      order by &timeid.;
```

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

```
proc sort data=work.%scan(&outputdsn,1,'_')_mape nodupkey;
   by %upcase(%scan(&outputdsn,1,'_'))_MAPE;
run;

proc print data=work.%scan(&outputdsn,1,'_')_mape;
run;

quit;

%mend;
```

The macro is run on the output data sets for both the ARMA(1,0) and the ARMAX(1,0) models.

```
/* Using the MAPE macro */
%mape(ar1_forecast,EDT,kW_Gen,6);
%mape(armax1_forecast,EDT,kW_Gen,6);
```

Output

Obs	AR1_MAPE
1	0.11792

Obs	ARMAX1_MAPE
1	0.067398

Alternatively, see below for using DATA step coding.

The **%MAPE_D** macro requires the following three arguments:

INDSN= the name of the data set output from the SAS/ETS forecasting procedure

SERIES= the name of the target series

NUMHOLDOUT= the number of time points for the holdout sample

```
/* MAPE macro using DATA step */

%macro mape_d(indsn=,series=,holdback=);
```

The DATA step creates a macro variable, **&FIRSTN**, that contains the k^{th} to the last observation value.

```
   data _null_;
      set &indsn end=eof;
      if eof then call symputx('firstn',(_n_-%eval(&holdback-1)));
   run;
```

The output data set is named the same as **&indsn**, but with a **_MAPE** suffix.

```
   %let outputdsn=&indsn._MAPE;
```

The DATA step creates the variable **Model**, whose value is the name of **&indsn**. The value of the variable **Series** is the value of **&series**. **MAPE** is calculated. A single observation is written to the SAS data set named the value of **&outputdsn**.

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

```
   data &outputdsn(keep=Series Model MAPE);
      length Model $200;
      set &indsn(firstobs=&firstn)
         end=eof;
      retain MAPE 0;
      MAPE+abs((&series-Forecast)/&series)/&holdback;
      if eof then do;
         Model="&indsn";
         Series="&series";
         output;
      end;
   run;
```

The data set is printed.

```
   proc print data=&outputdsn;
      id Series;
   run;

%mend mape_d;
```

The macro is run on the output data sets for both the ARMA(1,0) and the ARMAX(1,0) models.

```
/* Using the MAPE_D macro */
%mape_d(indsn=work.ar1_forecast,series=kW_Gen,holdback=6);
%mape_d(indsn=work.armax1_forecast,series=kW_Gen,holdback=6);
```

Output

Series	Model	MAPE
kW_Gen	work.ar1_forecast	0.11792

Series	Model	MAPE
kW_Gen	work.armax1_forecast	0.067398

The MAPE statistic is lower for the ARMAX(1,0) model (MAPE=0.067398) than for the ARMA(1,0) model (MAPE=0.11792). Because this model predicted the holdout sample better than the ARMA(1,0) model, it is used to forecast **kW_Gen** for future, unobserved periods.

In order to forecast **kW_Gen** one period into the future (t+1) using the ARMAX(1,0) model, **Cloud_Cover** for one period into the future must be provided in the data set. **Cloud_Cover** is forecasted for the next week and is listed in the **STSM.SOLARPV_F** data set.

The only difference between the **SOLARPV** and **SOLARPV_F** data sets is the 43rd observation corresponding to t+1. **Cloud_Cover** is provided, and **kW_Gen** is missing. Using the ARMAX(1,0) model, a forecast for **kW_Gen** can be generated.

The additional observation in the **STSM.SOLARPV_F** data set is as follows:

Obs	EDT	kW_Gen	Cloud_Cover
43	Sun, 26 Jul 2015	.	4.7869485829

End of Demonstration

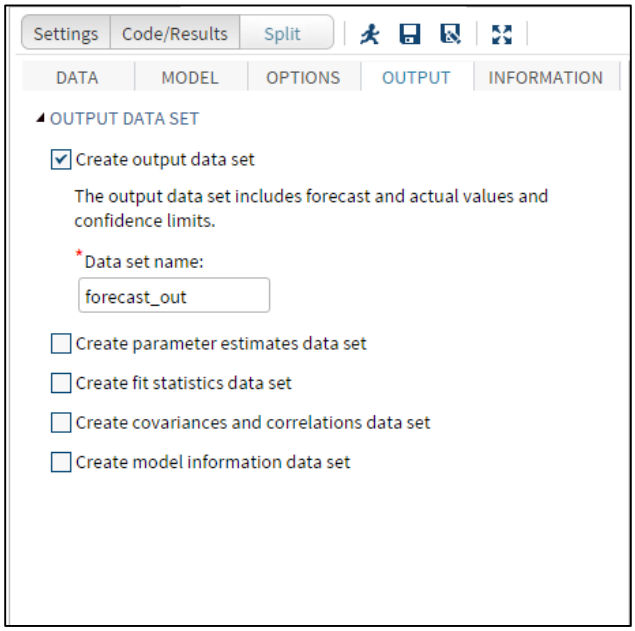
Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

Forecasting Future Values Using the Champion Model

STSM02d05d

Use the model with the smaller MAPE from the previous demonstration to forecast future values of **kW_Gen**.

1. On the DATA tab under the Modeling and Forecasting task, select the **STSM.SOLARPV_F** data set.
2. Specify **kW_Gen** as the dependent variable, and **EDT** as the time ID.
3. On the MODEL tab, specify the **ARIMAX** model type and the autoregressive order of **1**.
4. Click the plus sign (+) to the right of Independent variables. Add **Cloud_Cover** as the input variable.
5. Click the drop-down arrow next to **Plots**.
6. Under Selected Plots, scroll to the bottom and select both check boxes under Forecast Plots and clear all other plot check boxes.
7. On the OPTIONS tab, set the number of periods to forecast equal to **1**. No periods are being held back, so make sure that value is set to **0**. Clear the **Perform outlier detection** check box.
8. On the OUTPUT tab, create an output data set called **forecast_out**. This data set is written to the **Work** library and includes the forecast of **kW_Gen** for the unobserved time period.



The code generated by SAS Studio is shown below.

```
proc arima data=WORK.TempSorted plots
           (only)=(forecast(forecast forecastonly))
           out=WORK.forecast_out;
   identify var=kW_Gen crosscorr=(Cloud_Cover);
   estimate p=(1) input=(Cloud_Cover) method=ML;
   forecast lead=1 back=0 alpha=0.05 id=EDT interval=week printall;
   outlier;
quit;

run;
```

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

Alternatively, you can write the SAS code directly as shown.

```
/* STSM02d05.sas */
/* Part d: Forecasting the next period */
proc arima data=STSM.SOLARPV_F
           plots(only)=forecast(forecast forecastonly)
           out=WORK.forecast_out;
   identify var=kW_Gen crosscorr=(Cloud_Cover);
   estimate p=(1) input=(Cloud_Cover) method=ML;
   forecast lead=1 back=0 id=EDT;
quit;
```

9. Submit the program.

Scrolling to the bottom on the Results tab, notice that the Estimated Intercept, Autoregressive Factors, and Input Number tables are the same from the prior run of the ARMAX(1,0) model.

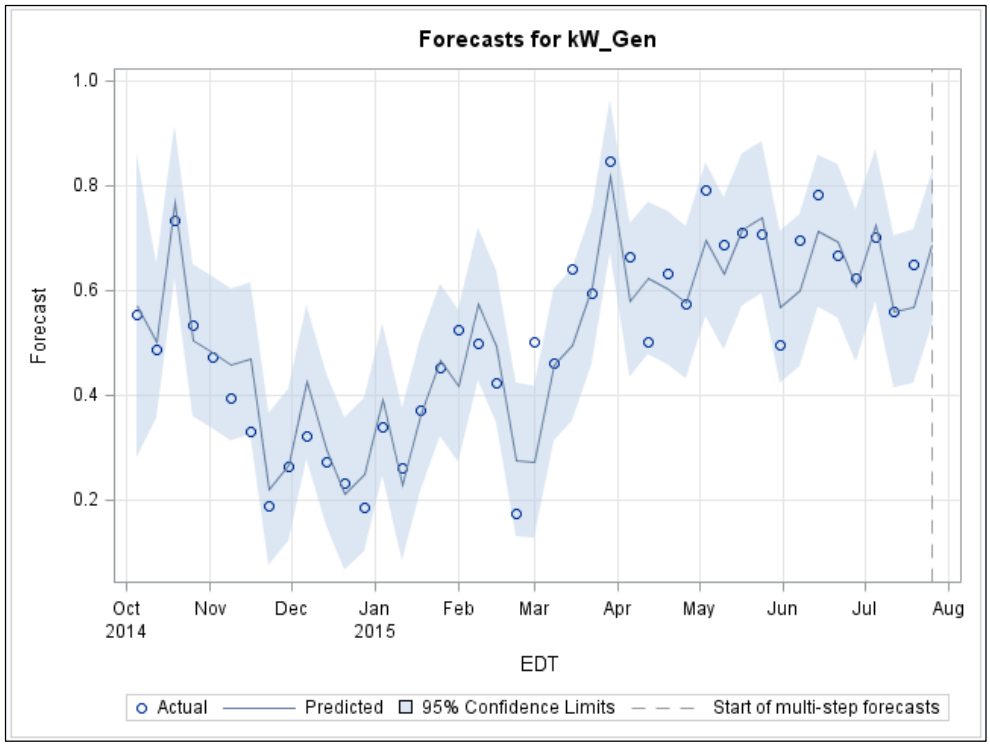
The Forecast table lists the forecasts, standard errors, 95% confidence limits, actual values, and residual values for all observations. The output below shows only the forecast for the future, unobserved period. The forecast for **kW_Gen** for the next period is 0.6856. This forecast for **kW_Gen**, along with the accompanying information is included in the **forecast_out** data set.

Forecasts for variable kW_Gen

Obs	Forecast	Std Error	95% Confidence Limits		Actual	Residual
…						
43	0.6856	0.0742	0.5402	0.8310	.	.

Because only one unmeasured period was forecast, the plots are not as informative as they otherwise might be if more periods were forecast. Nevertheless, the forecast for **kW_Gen** can still be seen in conjunction with observed periods in the forecast plot.

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.



The data set created on the Output tab can be used for a variety of different purposes, such as creating custom graphs, capturing forecasted periods, and so on. Also, because **Cloud_Cover** is a stochastic input variable, scenario analysis can be used to forecast future periods of **kW_Gen** (> t+1) given different values of **Cloud_Cover**.

End of Demonstration

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

Exercises

4. Validation and Forecasting of Rose Series 4

Use validation statistics on the **STSM.ROSESERIES** series with the most recent year (52 weeks) as a holdout sample. Find a champion model and use it to forecast 11 future time periods.

a. Using the **SALES4** series and the **STSM.ROSESERIES** data set, build an ARMA(1,0) model and forecast a holdout sample of the most recent 52 observations. Create an output data set titled **AR1_FORECAST** that is written to the **Work** library.

Analyze the forecast plots. Visually, how does the ARMA(1,0) model appear to fit the holdout sample?

b. Using the **SALES4** series, the **RAMP** input variable, and the **STSM.ROSESERIES** data set, build an ARMAX(1,0) model and forecast a holdout sample of the most recent 52 observations. Create an output data set titled **ARMAX1_FORECAST** that is written to the **Work** library.

How well does the ARMAX(1,0) model seem to fit the holdout sample? Does this or the previous model appear to fit the holdout sample better?

c. Use the two output data sets that you created, **AR1_forecast** and **ARMAX1_forecast**, to calculate MAPE for both models. Use one of the macros in **MAPE_Macro.sas** to calculate MAPE for both the ARMA(1,0) model and the ARMAX(1,0) model.

Which candidate model for **SALES4** provides the lower MAPE?

(Use the better model for step **d**.)

d. Use the **STSM.ROSESERIES_F** data set and the best candidate model (ARMAX(1,0)) to forecast the next 11 future, unknown time periods of rose sales.

End of Exercises

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

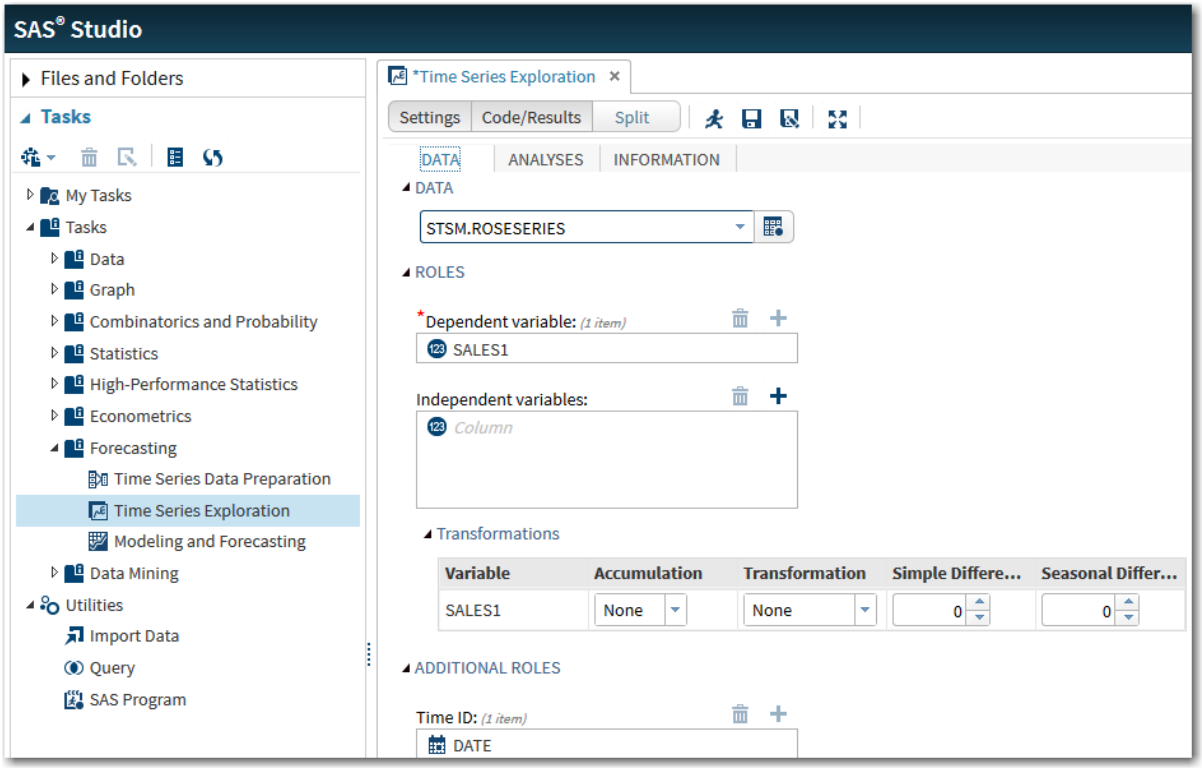
2.6 Solutions

Solutions to Exercises

1. Analyzing a Rose Sales Series

Which series show autocorrelation?

Use the pattern below in the Time Series Exploration task to analyze the four series. Start with **SALES1**.

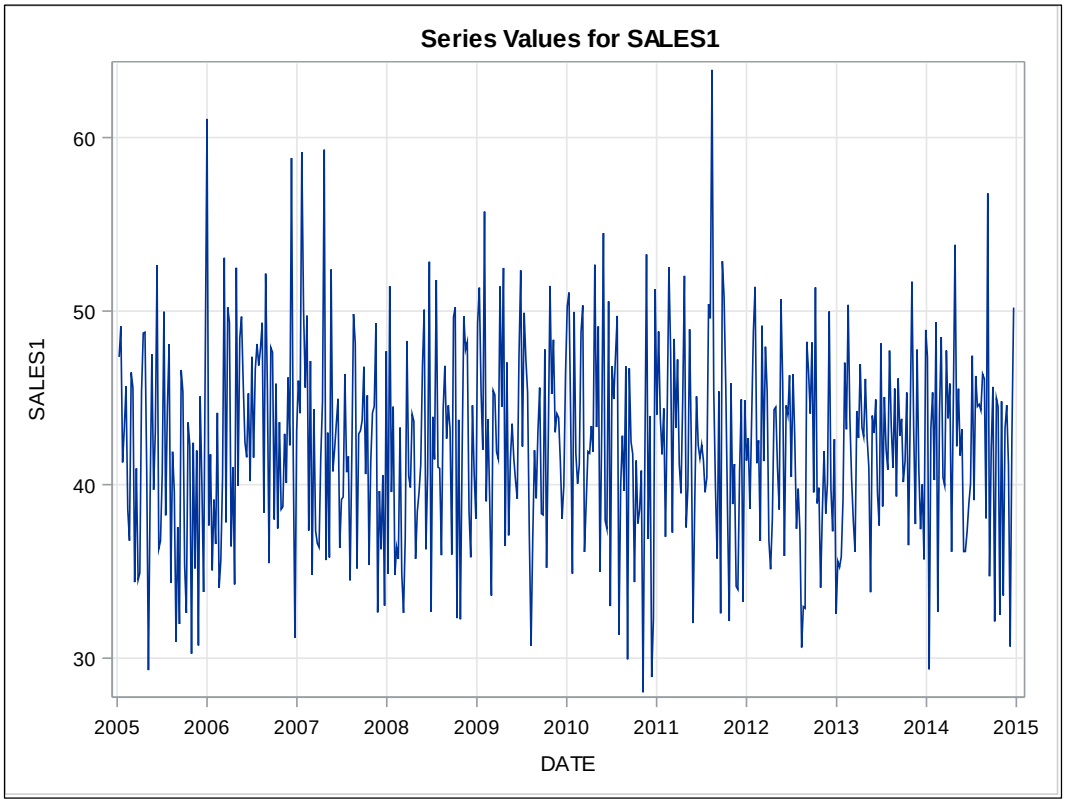


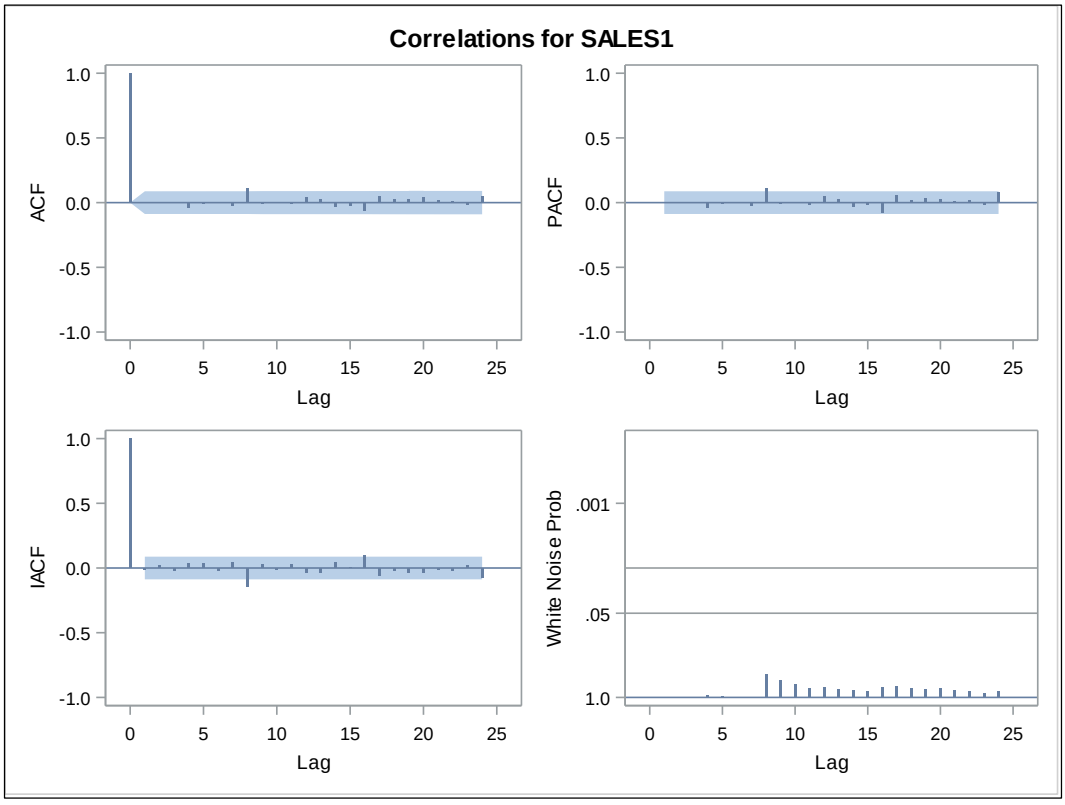
You can write the SAS code to look at all four series simultaneously.

```
/* STSM02s02.sas */
ods graphics / imagemap=on;

/* Look at the autocorrelation panel plot */
proc timeseries data=STSM.ROSESERIES plots=(series corr);
   id DATE interval=week;
   var SALES1 SALES2 SALES3 SALES4;
run;
```

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

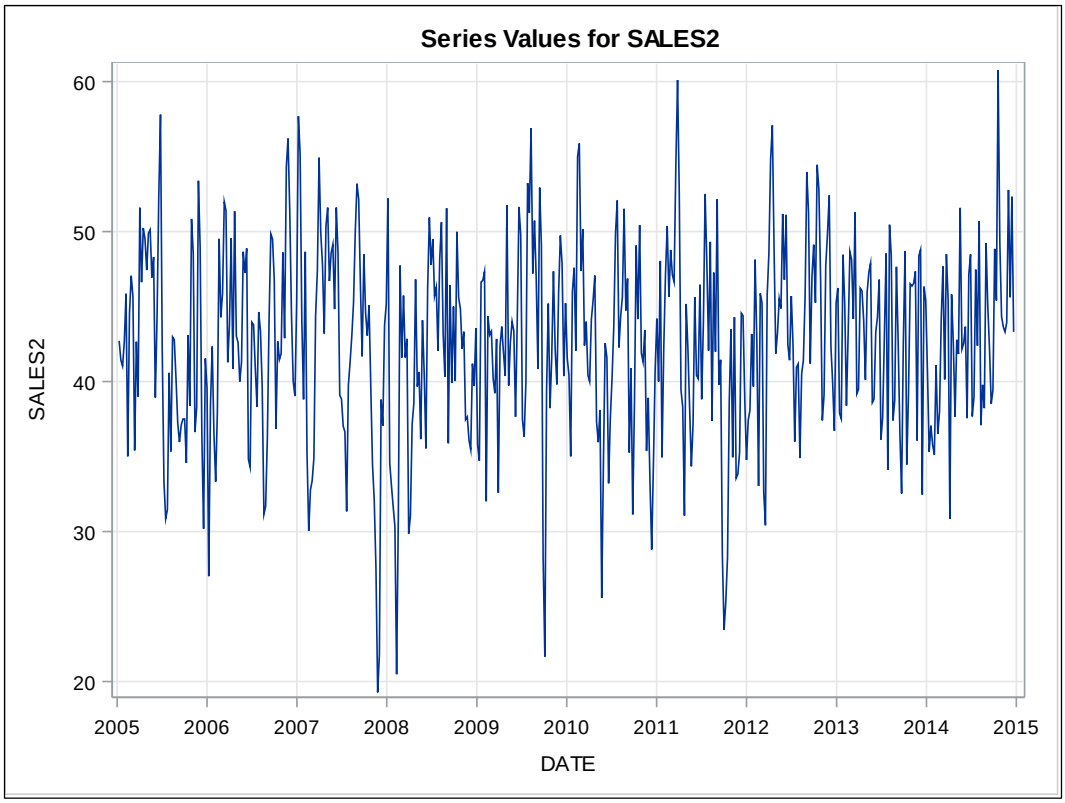
Selected Output

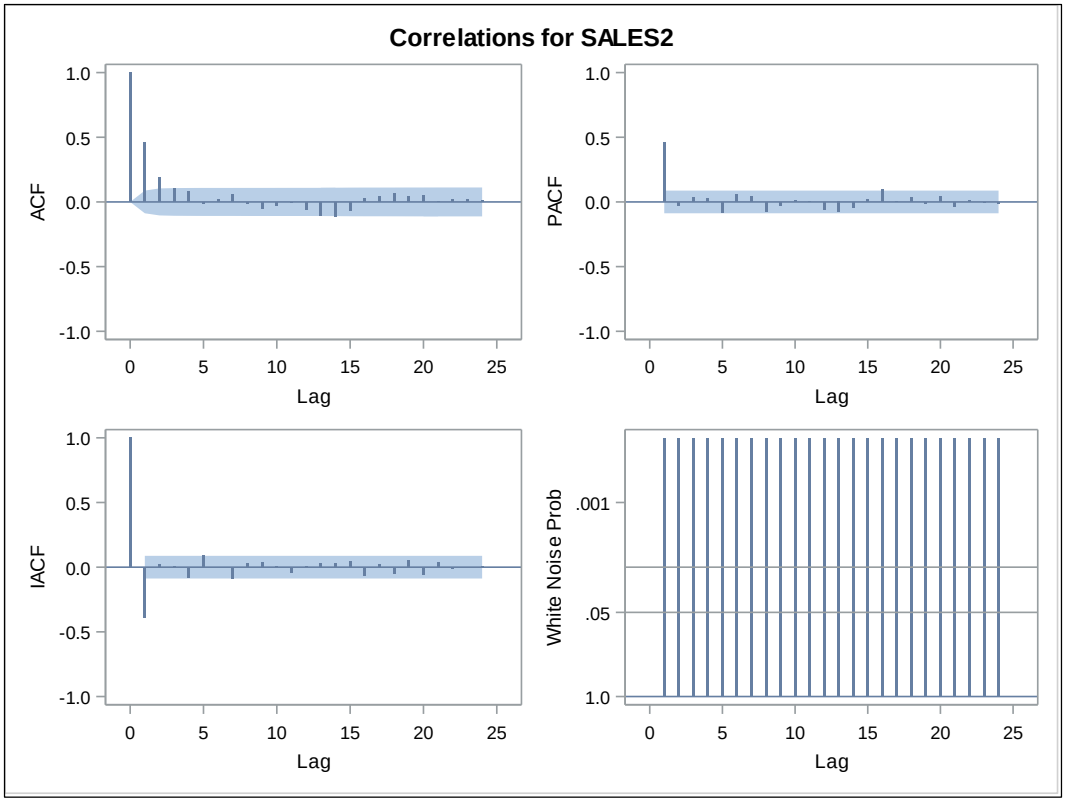




SALES1 shows no evidence of dependency across the first 24 lags. It looks like a white noise series.

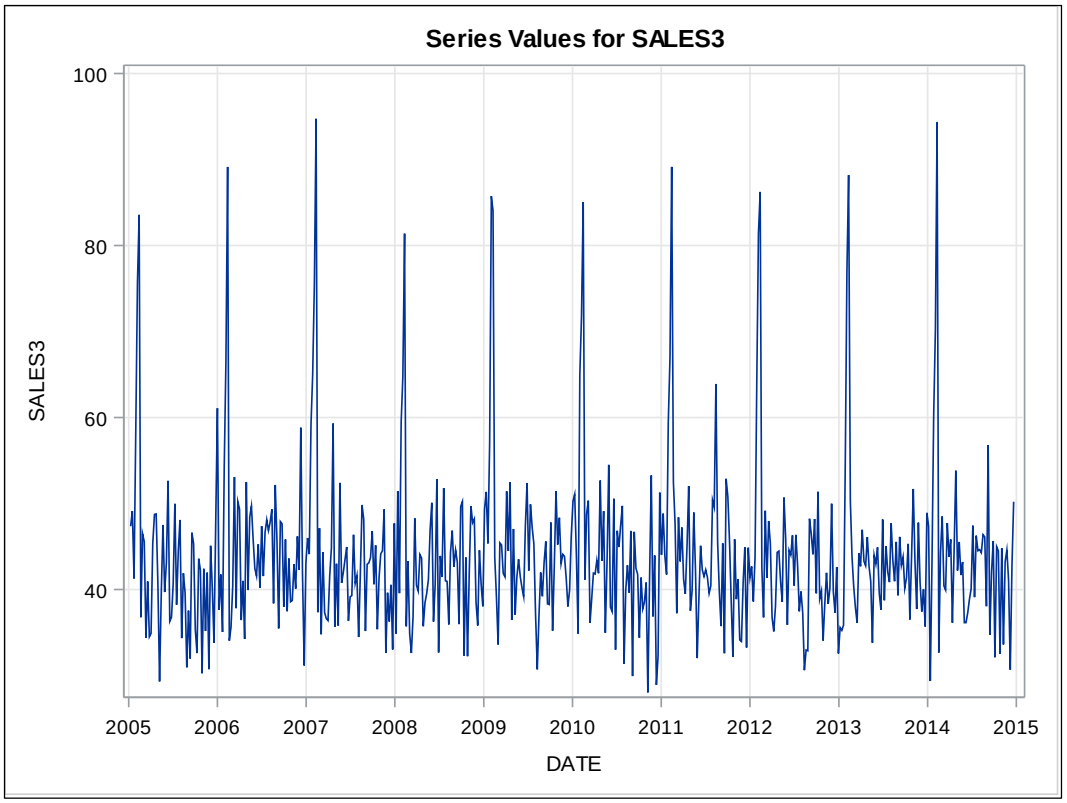
Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

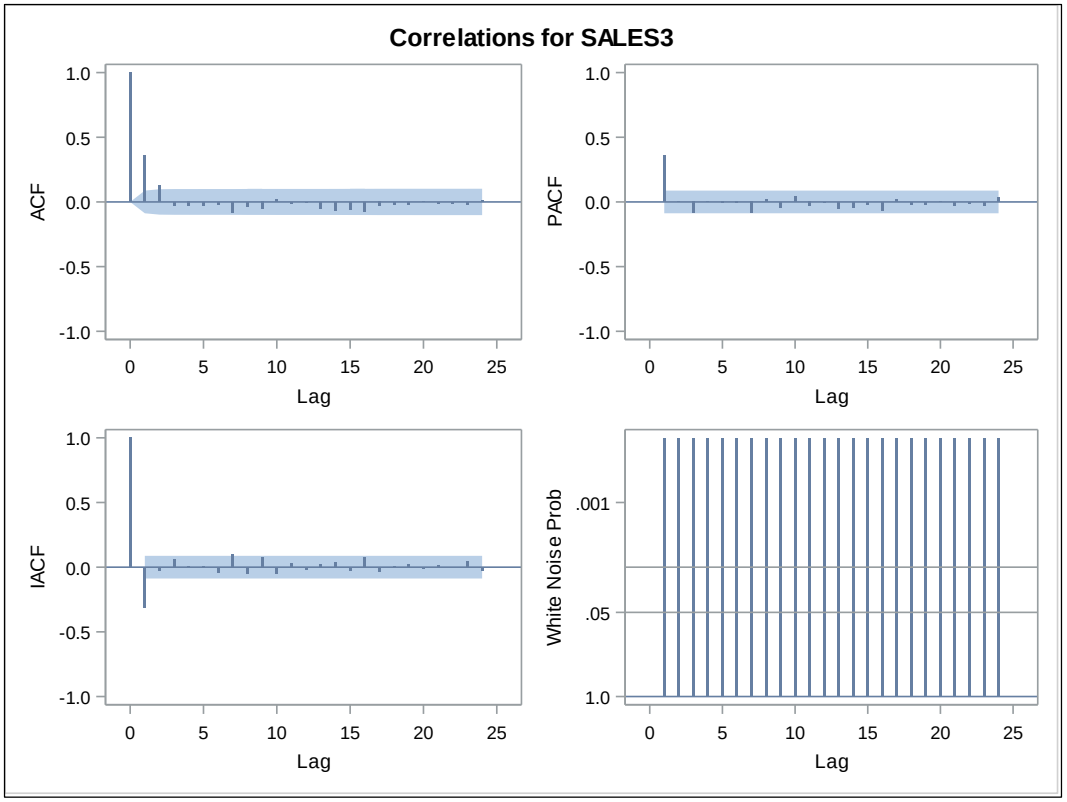




There is one spike at 0 lag in the **SALES2** PACF plot. This indicates a potential order 1 autocorrelation.

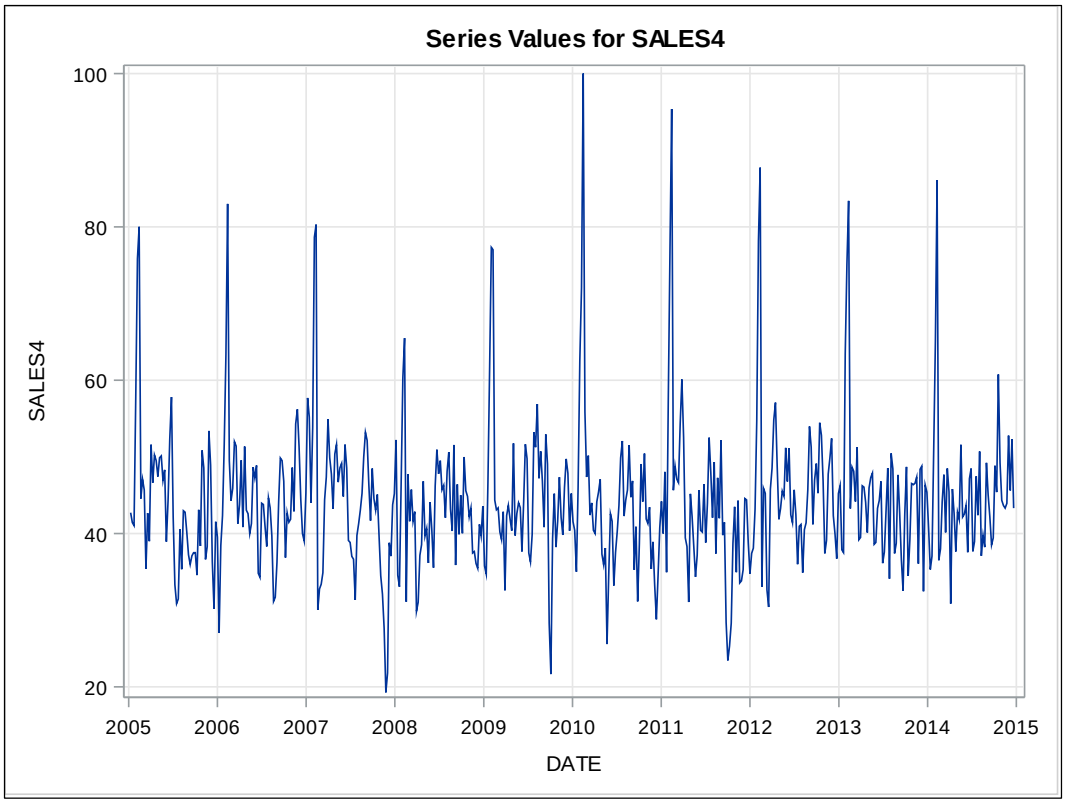
Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

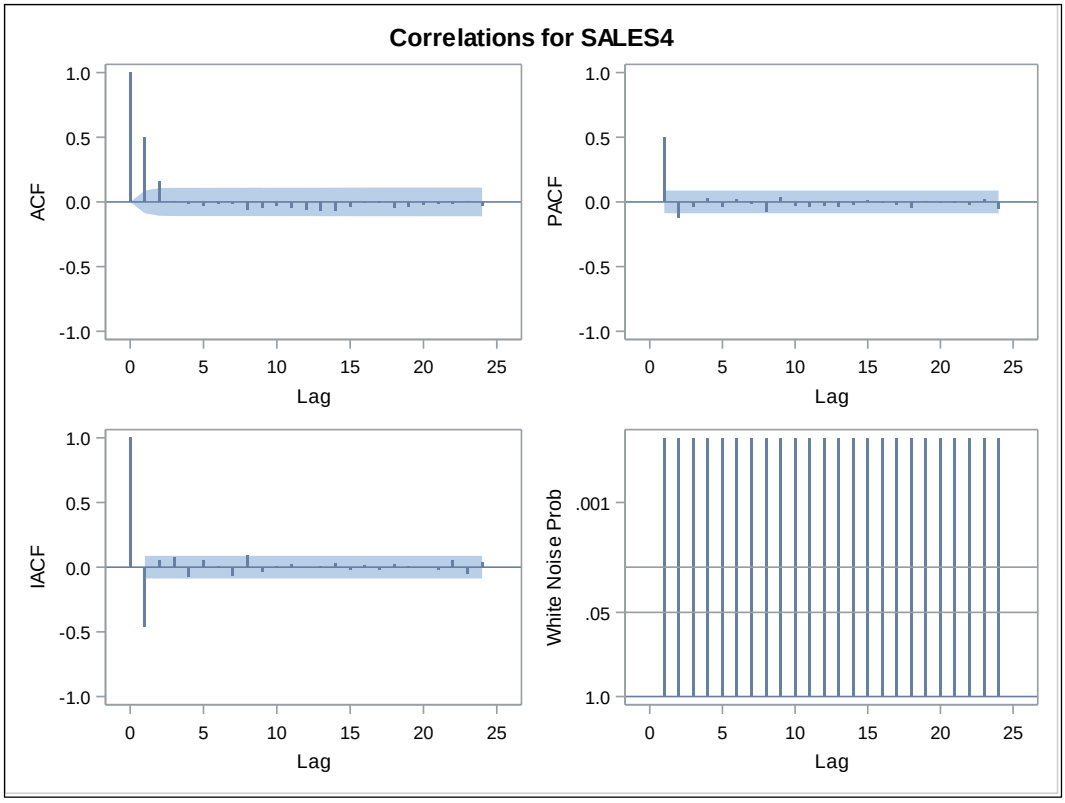




There is also one statistically significant spike at lag 1 for the **SALES3** PACF. There are also some spikes at regular intervals throughout the series.

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.





The PACF of **SALES4** also indicates first order autocorrelation, but with spikes, it is similar to **SALES3**.

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

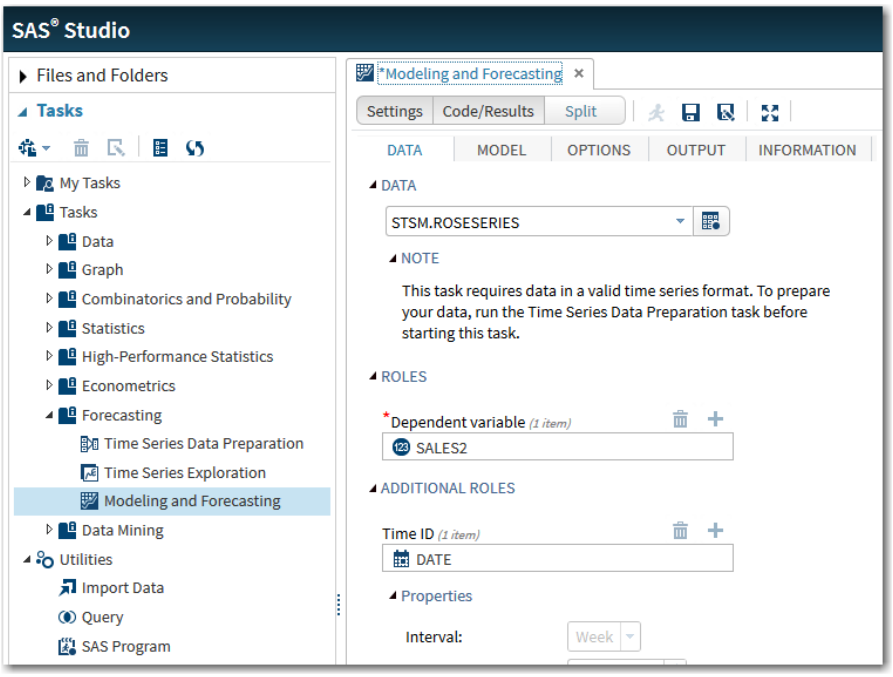
2. Rose Series Estimation

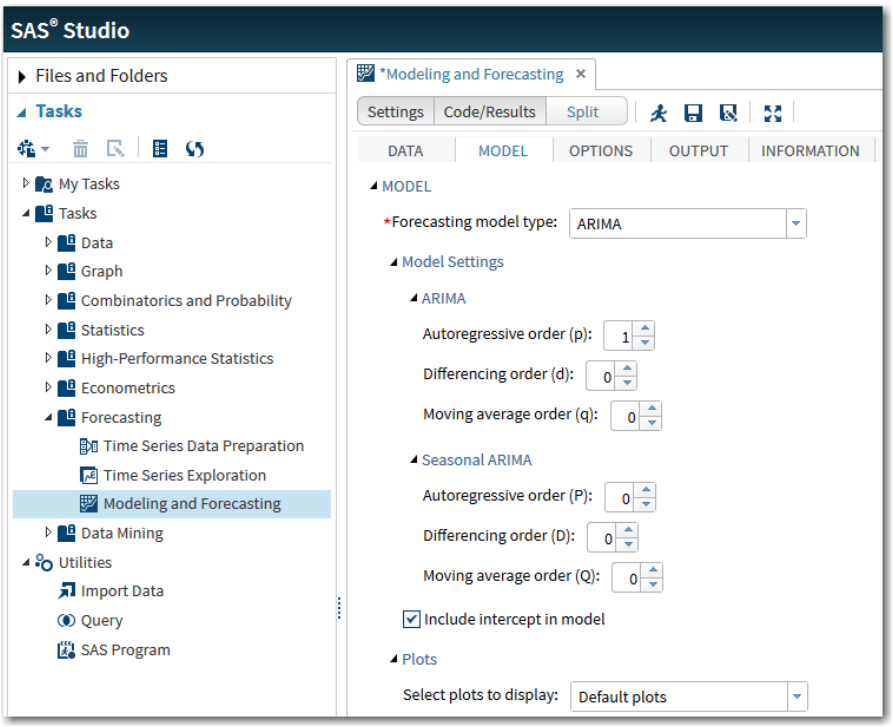
For each rose sales series that showed any autocorrelation, estimate the autoregression parameters of an AR(1) model and look at the residuals?

a. Is the autoregression parameter estimate statistically significant?

b. Do the residuals indicate that the model is sufficient for the series?

Use the pattern below in the Modeling and Forecasting task to analyze the series. Start with **SALES2**.





Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

You can also write PROC ARIMA code directly to model both series by using an IDENTIFY statement, an ESTIMATE statement, and then a RUN statement (not a QUIT statement, until all models are estimated).

If you are running the code in SAS Studio and want to submit the program in parts, as in this example, be sure to use an interactive mode first by clicking this button:  .

```
/* STSM02s03.sas */
proc arima data=STSM.ROSESERIES
           plots(only)=(series(corr)
                        residual(corr normal));
   identify var=SALES2;
   estimate p=(1) method=ML;
run;
```

Partial Output

Name of Variable = SALES2	
Mean of Working Series	42.5983
Standard Deviation	6.492443
Number of Observations	520

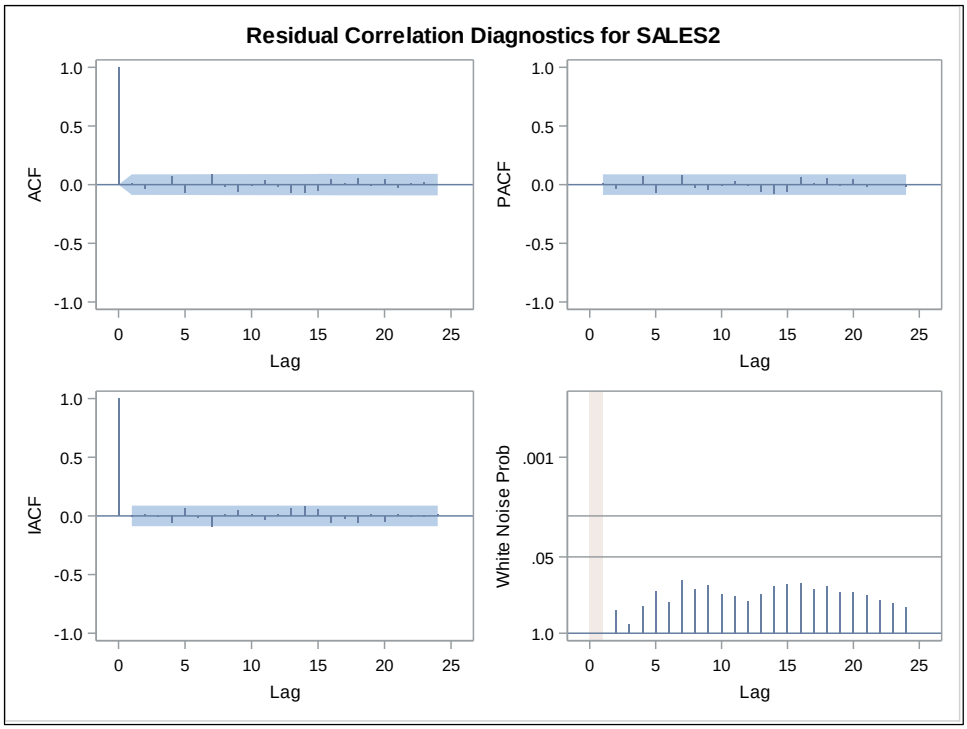
Maximum Likelihood Estimation					
Parameter	Estimate	Standard Error	t Value	Approx Pr > |t|	Lag
MU	42.59964	0.46518	91.58	<.0001	0
AR1,1	0.45529	0.03909	11.65	<.0001	1

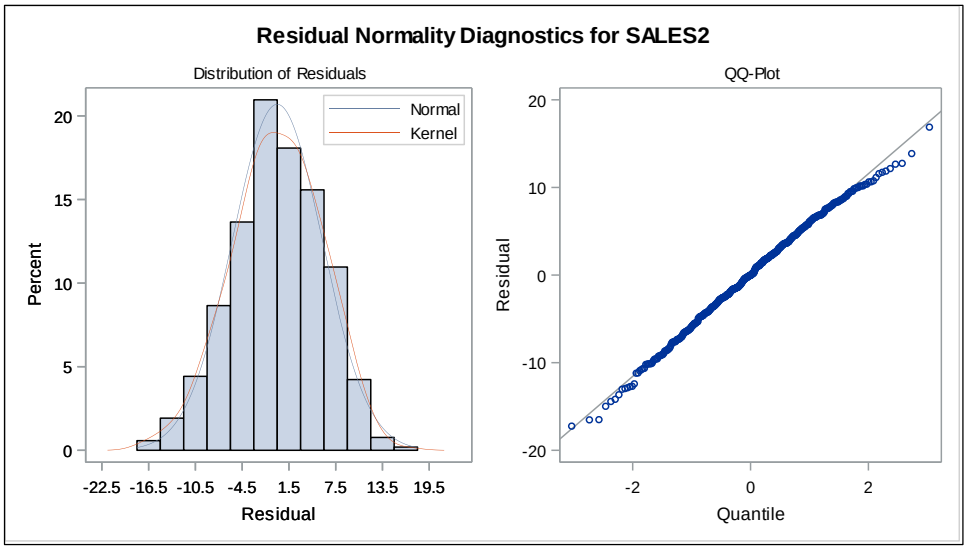
The first order autoregressive term is statistically significant.

Constant Estimate	23.20426
Variance Estimate	33.5095
Std Error Estimate	5.788739
AIC	3304.076
SBC	3312.583
Number of Residuals	520

Autocorrelation Check of Residuals									
To Lag	Chi-Square	DF	Pr > ChiSq	Autocorrelations					
6	6.08	5	0.2985	0.012	-0.035	0.005	0.070	-0.072	-0.002
12	13.08	11	0.2882	0.086	-0.019	-0.058	-0.009	0.040	-0.021
18	22.74	17	0.1577	-0.068	-0.066	-0.055	0.051	0.014	0.057
24	24.68	23	0.3670	-0.007	0.047	-0.028	0.010	0.019	-0.003
30	30.96	29	0.3674	0.012	-0.035	0.064	-0.069	-0.020	0.028
36	34.52	35	0.4912	-0.034	-0.036	-0.035	-0.028	-0.038	-0.024
42	42.07	41	0.4243	-0.059	0.071	-0.047	-0.038	-0.033	-0.014
48	49.50	47	0.3738	0.020	0.010	0.018	-0.104	0.004	0.038

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.





Model for variable SALES2	
Estimated Mean	42.59964

Autoregressive Factors	
Factor 1:	1 - 0.45529 B**(1)

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

For SALES2, the residuals do not appear to be autocorrelated and they are relatively normally distributed. The AR(1) model seems appropriate.

```
   identify var=SALES3;
   estimate p=(1) method=ML;
run;
```

Partial Output

Name of Variable = SALES3	
Mean of Working Series	43.99515
Standard Deviation	9.676915
Number of Observations	520

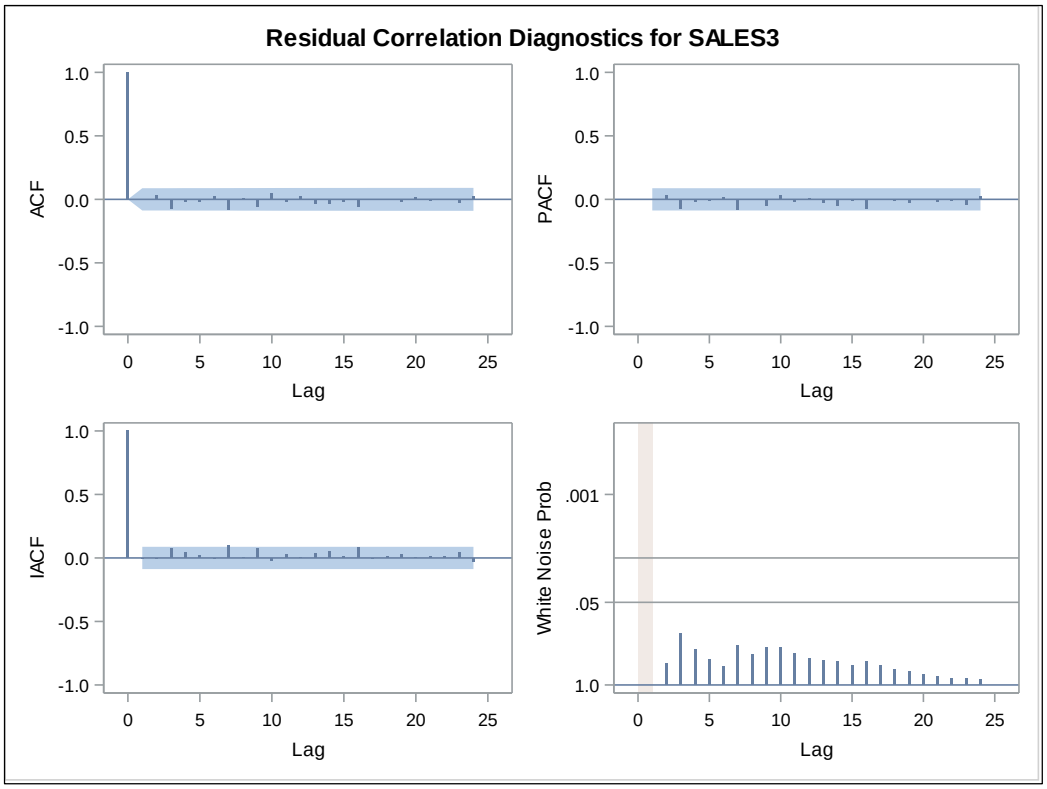
Maximum Likelihood Estimation					
Parameter	Estimate	Standard Error	t Value	Approx Pr > |t|	Lag
MU	44.00528	0.61573	71.47	<.0001	0
AR1,1	0.35547	0.04107	8.66	<.0001	1

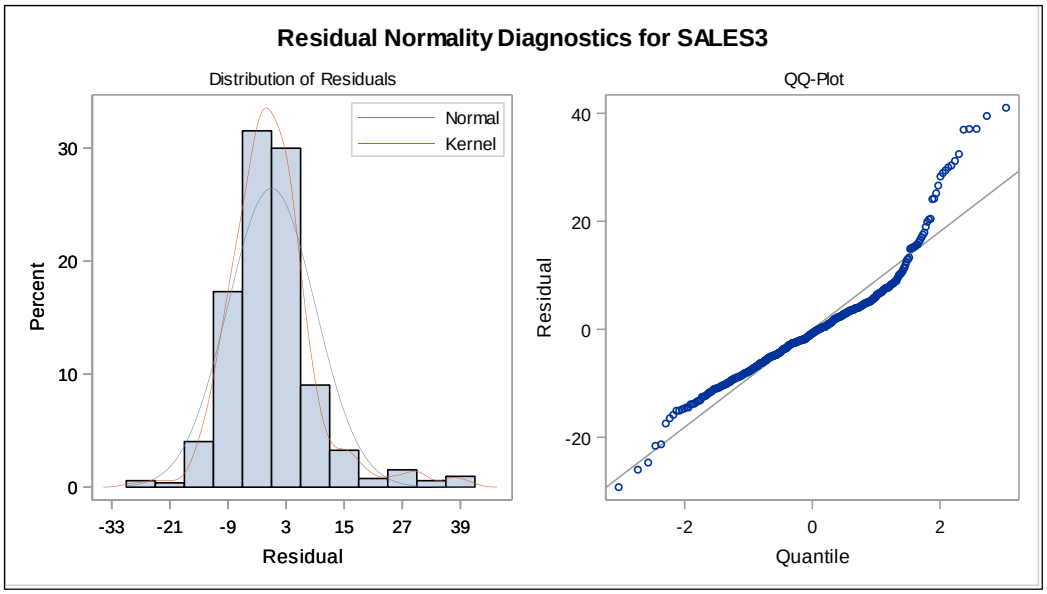
The first order autoregressive term is statistically significant.

Constant Estimate	28.36282
Variance Estimate	82.09258
Std Error Estimate	9.060496
AIC	3769.908
SBC	3778.416
Number of Residuals	520

Autocorrelation Check of Residuals										
To Lag	Chi-Square	DF	Pr > ChiSq	Autocorrelations						
6	4.30	5	0.5065	-0.000	0.033	-0.078	-0.017	-0.017	0.023
12	11.69	11	0.3877	-0.084	0.007	-0.062	0.046	-0.018	0.022
18	15.34	17	0.5707	-0.037	-0.038	-0.020	-0.060	-0.003	-0.000
24	16.75	23	0.8211	-0.021	0.017	-0.014	-0.002	-0.029	0.029
30	20.94	29	0.8615	0.010	-0.033	-0.064	0.021	-0.036	0.023
36	24.62	35	0.9047	-0.040	0.043	-0.033	-0.012	-0.035	-0.025
42	30.45	41	0.8866	-0.050	0.024	-0.073	-0.026	-0.014	0.032
48	36.37	47	0.8690	-0.031	-0.059	-0.072	-0.006	-0.024	0.010

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.





Model for variable SALES3	
Estimated Mean	44.00528

Autoregressive Factors	
Factor 1:	1 - 0.35547 B**(1)

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

For SALES3, although the residuals are not significantly autocorrelated, they are not normally distributed, but rather positively skewed.

```
   identify var=SALES4;
   estimate p=(1) method=ML;
   run;
quit;
```

Partial Output

Name of Variable = SALES4	
Mean of Working Series	44.32907
Standard Deviation	9.591172
Number of Observations	520

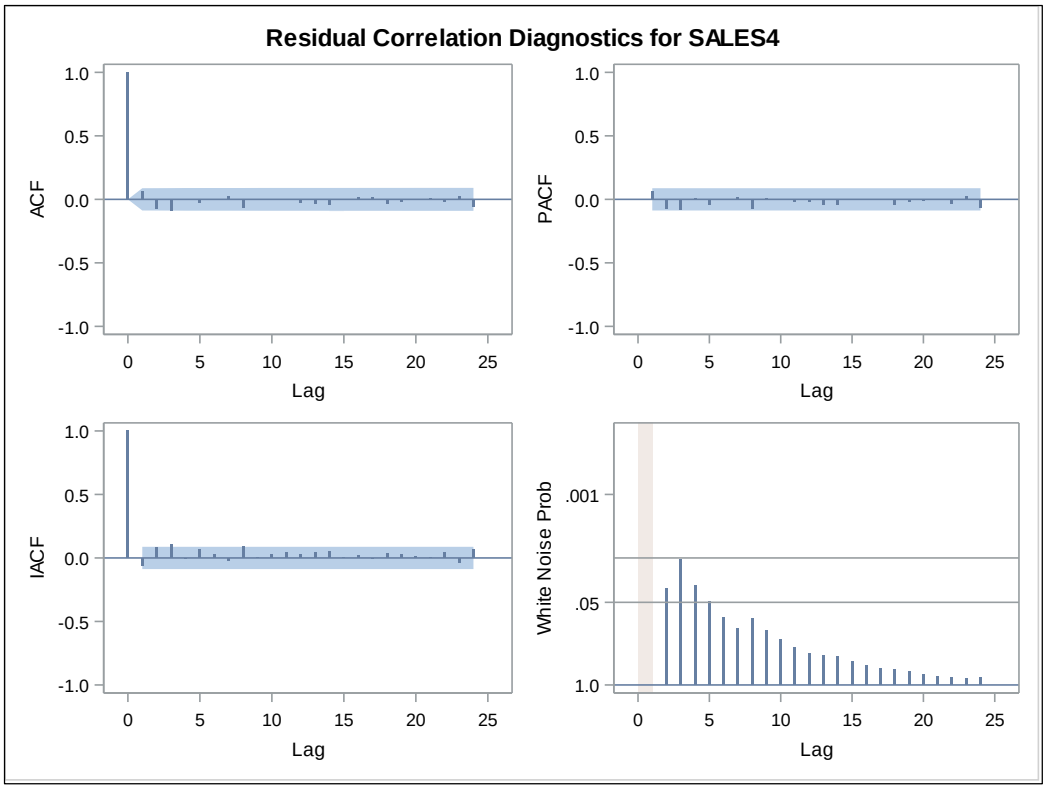
Maximum Likelihood Estimation					
Parameter	Estimate	Standard Error	t Value	Approx Pr > |t|	Lag
MU	44.32410	0.72503	61.13	<.0001	0
AR1,1	0.49707	0.03809	13.05	<.0001	1

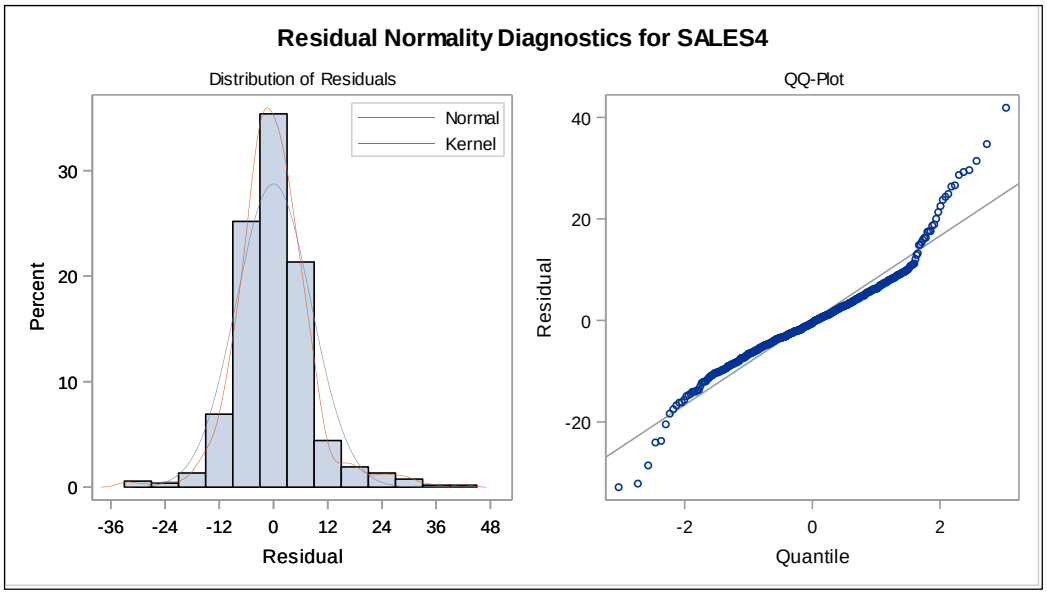
The autoregressive term is statistically significant.

Constant Estimate	22.29208
Variance Estimate	69.44338
Std Error Estimate	8.333269
AIC	3683.042
SBC	3691.55
Number of Residuals	520

Autocorrelation Check of Residuals									
To Lag	Chi-Square	DF	Pr > ChiSq	Autocorrelations					
6	9.62	5	0.0868	0.061	-0.072	-0.092	0.000	-0.030	0.006
12	12.69	11	0.3141	0.025	-0.067	-0.006	0.003	-0.008	-0.024
18	15.42	17	0.5650	-0.033	-0.044	-0.007	0.019	0.016	-0.037
24	18.07	23	0.7536	-0.020	-0.006	0.009	-0.020	0.029	-0.055
30	26.50	29	0.5987	0.032	0.008	0.052	-0.096	0.046	-0.015
36	31.19	35	0.6528	-0.006	-0.054	-0.005	-0.066	-0.032	-0.003
42	32.79	41	0.8161	0.009	0.006	-0.017	-0.038	-0.029	-0.011
48	35.95	47	0.8796	0.030	-0.011	-0.014	-0.062	0.020	-0.009

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.





Model for variable SALES4	
Estimated Mean	44.3241

Autoregressive Factors	
Factor 1:	1 - 0.49707 B**(1)

The residuals do not appear to be white noise. They also appear to be non-normal, exhibiting high kurtosis.

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

3. Intervention Analysis of the Rose Series

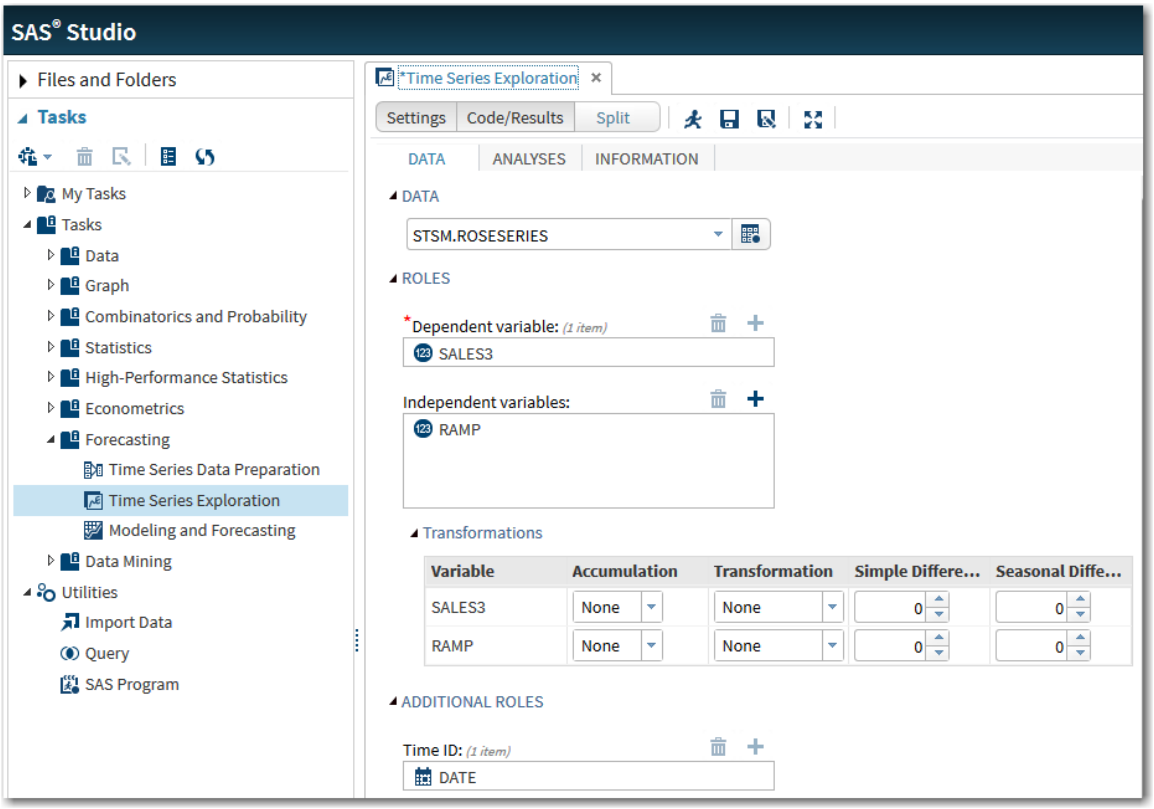
For each rose series (where it is appropriate), use the **Ramp** variable to model the effect of the impending Valentine's Day on weekly rose sales.

a. Open and submit the code in **STSM02e04.sas**.

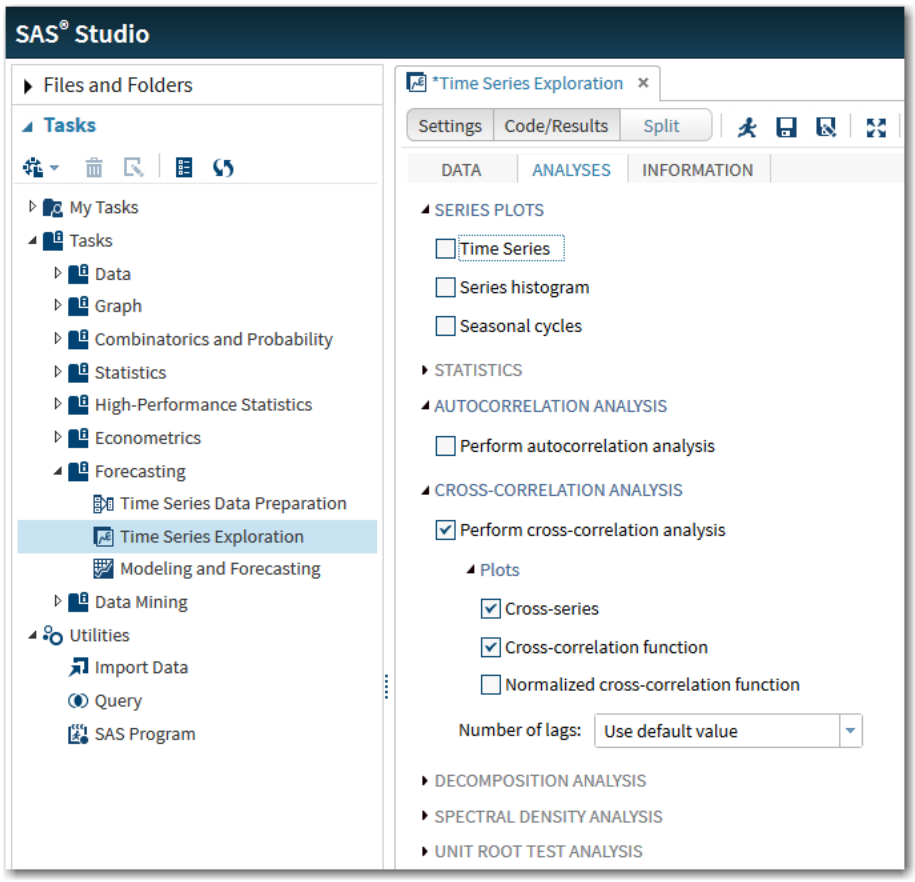
b. For each rose sales series that was not white noise and was not adequately modeled as AR(1) alone, look at the cross-correlation plot with the **RAMP** dummy code series.

Do the series seem to show significant cross-correlation with the RAMP series?

- Use the pattern below in the Time Series Exploration task to analyze the potential cross-correlation with **Ramp** of the last two series. Start with **SALES3**.



Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.




```
/* STSM02s04a.sas */
proc timeseries data=STSM.ROSESERIES
                crossplots=(series ccf);
   id DATE interval=week;
   var SALES3 SALES4;
   crossvar RAMP;
   ods exclude CCFNORMPlot;
run;
```

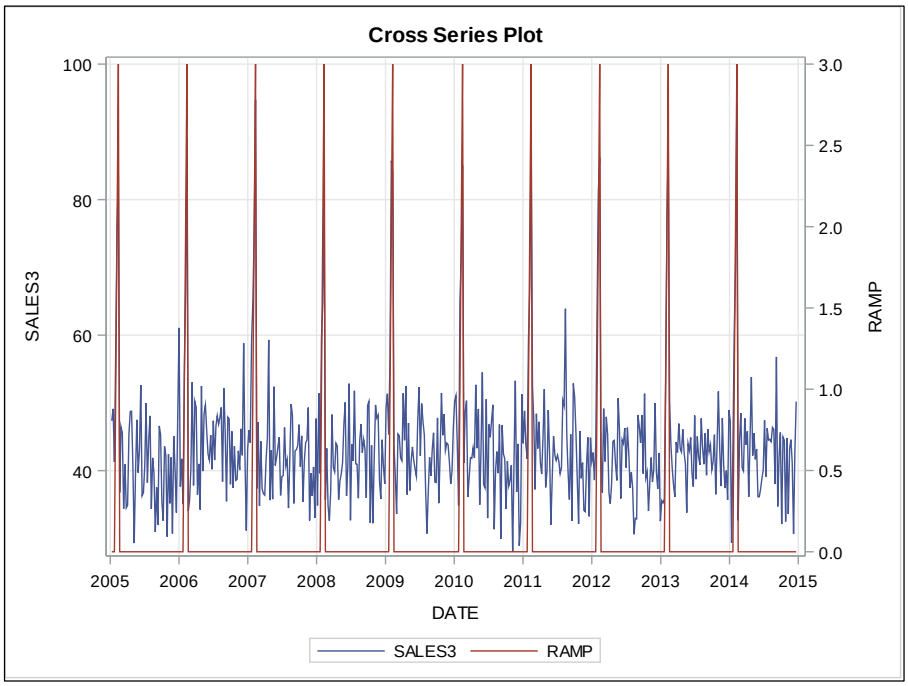
Output

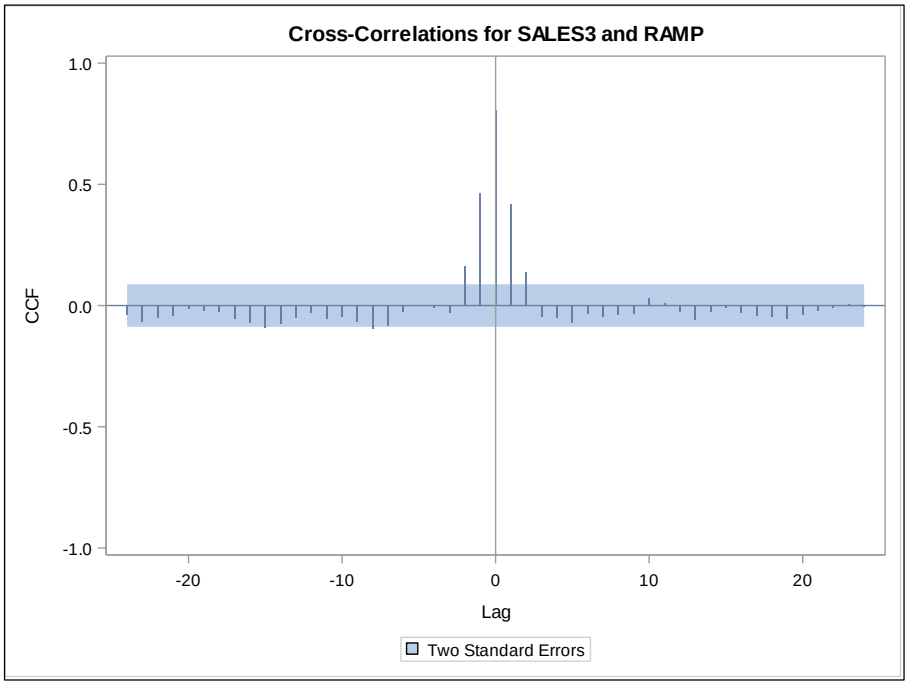
Input Data Set	
Name	STSM.ROSESERIES
Label	
Time ID Variable	DATE
Time Interval	WEEK
Length of Seasonal Cycle	52

Variable Information	
Name	SALES3
Label	
First	Sat, 15 Jan 2005

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

Variable Information	
Last	Sat, 27 Dec 2014
Number of Observations Read	520

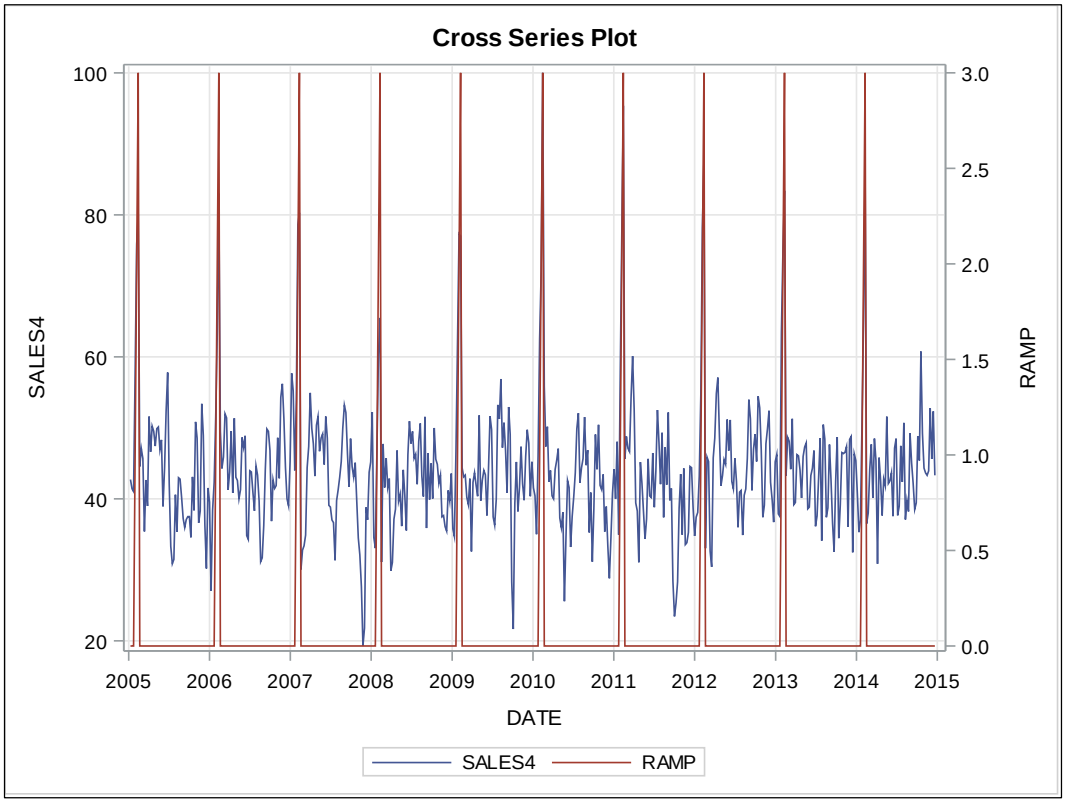




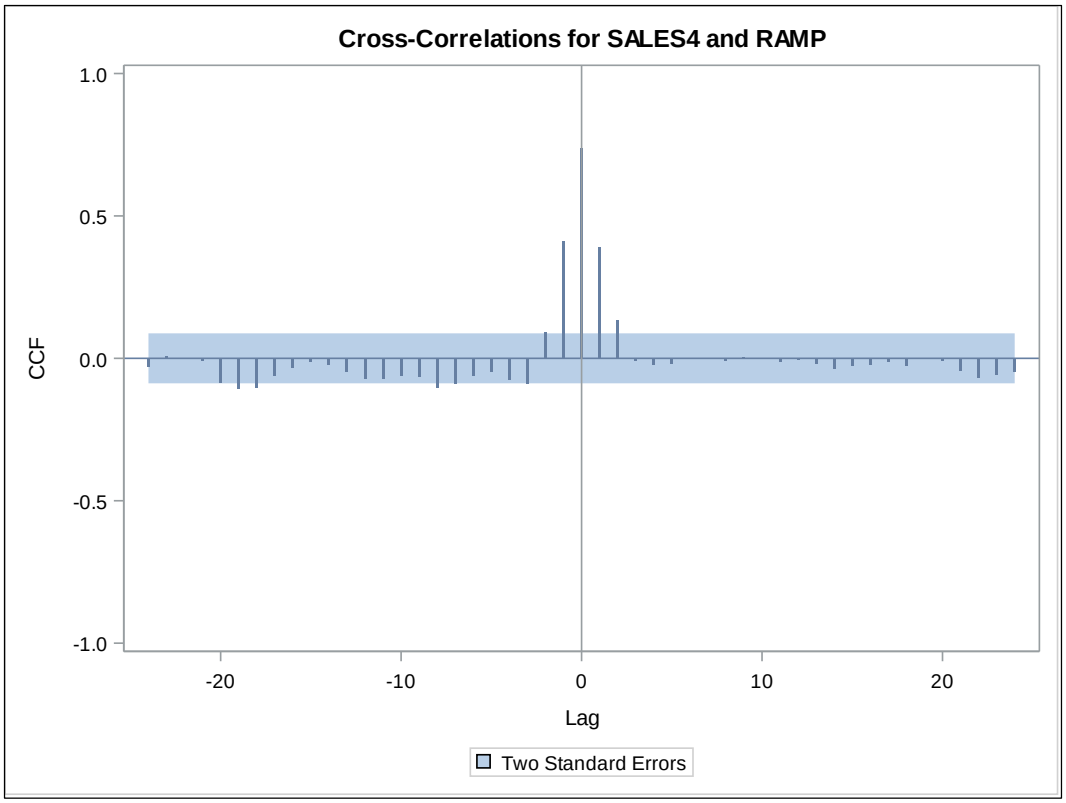
For SALES3, there seems to be concurrent (lag 0) cross-correlation with the Ramp dummy variable.

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

Variable Information	
Name	SALES4
Label	
First	Sat, 15 Jan 2005
Last	Sat, 27 Dec 2014
Number of Observations Read	520



Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.



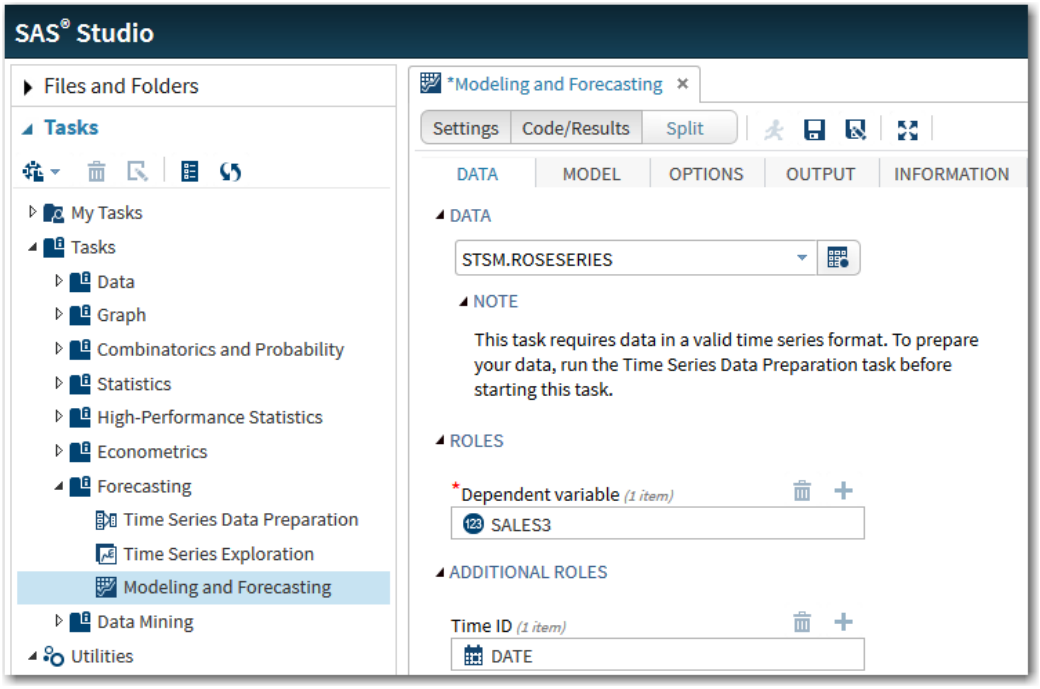
For SALES4, there also seems to be a significant cross-correlation with Ramp.

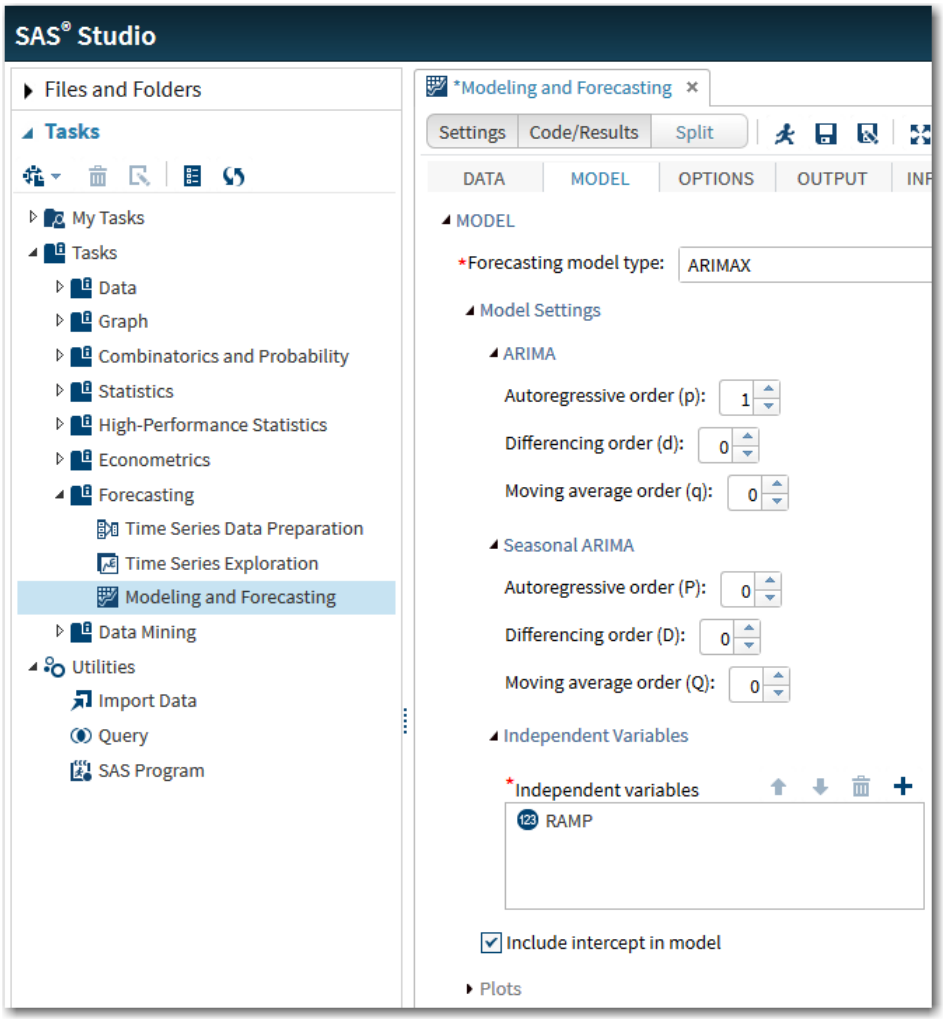
c. For each series that showed any cross-correlation with **Ramp**, estimate the autoregression parameters of an appropriate ARMAX model and look at the residuals.

1) Is the autoregression parameter estimate statistically significant?
2) Is the cross-correlation parameter estimate statistically significant?
3) Do the residuals indicate that the model is sufficient for the series?

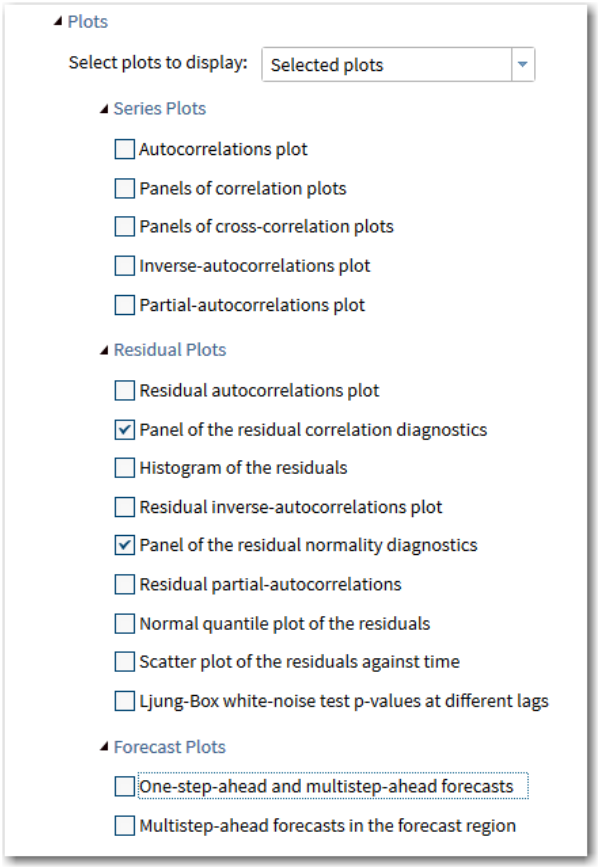
Use the pattern below in the Modeling and Forecasting task to analyze the final two series. Start with **SALES3**.

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.





Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.



The PROC ARIMA code is as follows:

```
/* STSM02s04b.sas */
proc arima data=STSM.ROSESERIES
           plots(only)=(residual(corr normal));
   identify var=SALES3 crosscorr=(RAMP);
   estimate p=(1) input=(RAMP) method=ML;
run;
```

Partial Output

Correlation of SALES3 and RAMP	
Variance of input =	0.255917
Number of Observations	520

Maximum Likelihood Estimation							
Parameter	Estimate	Standard Error	t Value	Approx Pr > |t|	Lag	Variable	Shift
MU	42.21243	0.25728	164.07	<.0001	0	SALES3	0
AR1,1	-0.0004305	0.04409	-0.01	0.9922	1	SALES3	0
NUM1	15.45016	0.49621	31.14	<.0001	0	RAMP	0

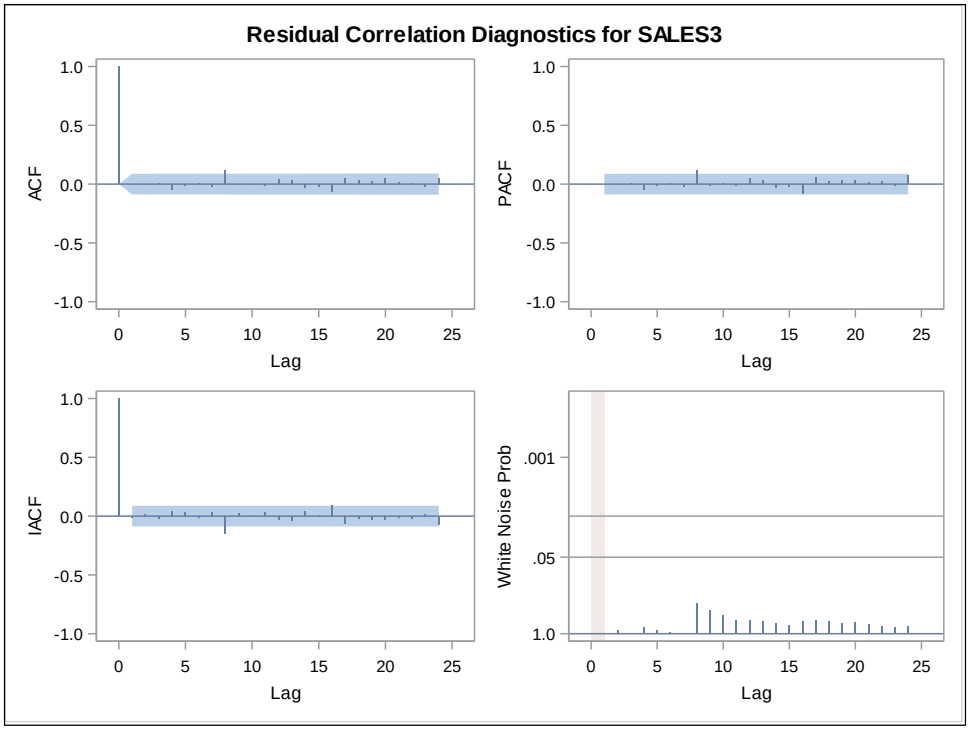
The parameter estimate for the first-order autoregressive parameter is not statistically significant, but the parameter estimate for the transfer function for RAMP is significant. The autoregression parameter can possibly be removed and the model can be re-estimated.

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

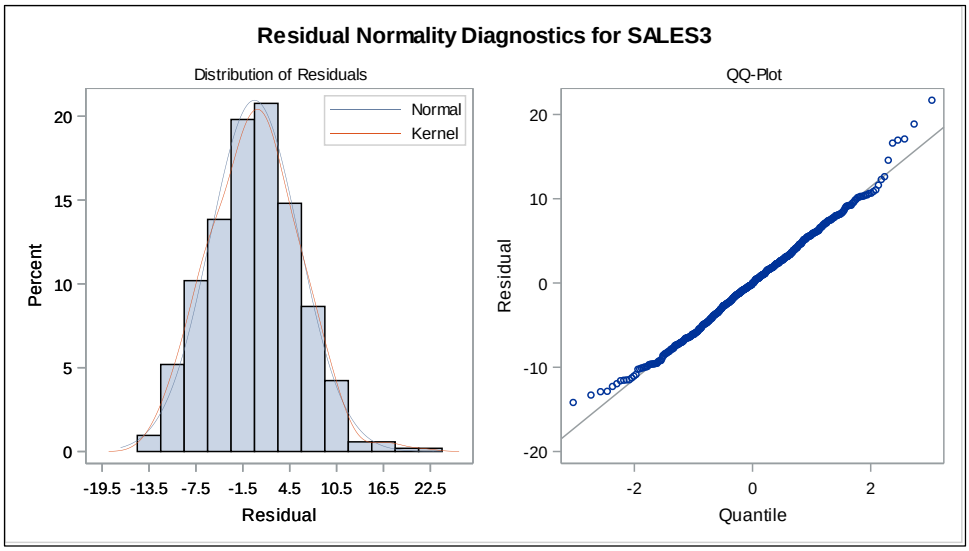
Constant Estimate	42.2306
Variance Estimate	32.74205
Std Error Estimate	5.722067
AIC	3292.791
SBC	3305.552
Number of Residuals	520

The AIC for this model is 3292.791. This can be compared to the AIC in a model without the autoregression parameter.

Autocorrelation Check of Residuals									
To Lag	Chi-Square	DF	Pr > ChiSq	Autocorrelations					
6	1.17	5	0.9476	-0.000	-0.007	0.006	-0.044	-0.014	0.005
12	9.43	11	0.5823	-0.020	0.115	-0.008	0.001	-0.013	0.042
18	14.67	17	0.6192	0.030	-0.031	-0.020	-0.066	0.048	0.028
24	18.02	23	0.7565	0.027	0.044	0.018	0.008	-0.019	0.052
30	25.00	29	0.6783	0.089	-0.031	-0.052	-0.009	-0.024	0.022
36	32.43	35	0.5930	-0.030	0.069	-0.039	-0.073	-0.029	0.003
42	39.87	41	0.5208	-0.007	0.088	-0.031	-0.054	-0.036	-0.011
48	44.51	47	0.5761	-0.006	-0.001	-0.066	-0.054	-0.026	0.010



Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.



The residuals appear to be white noise.

Model for variable SALES3	
Estimated Intercept	42.21243

Autoregressive Factors	
Factor 1:	1 + 0.00043 B**(1)

Input Number 1	
Input Variable	RAMP
Overall Regression Factor	15.45016

Remove the autoregressive parameter.

```
   estimate input=(RAMP) method=ML;
run;
```

Partial Output

Maximum Likelihood Estimation							
Parameter	Estimate	Standard Error	t Value	Approx Pr > |t|	Lag	Variable	Shift
MU	42.21243	0.25713	164.17	<.0001	0	SALES3	0
NUM1	15.45018	0.49554	31.18	<.0001	0	RAMP	0

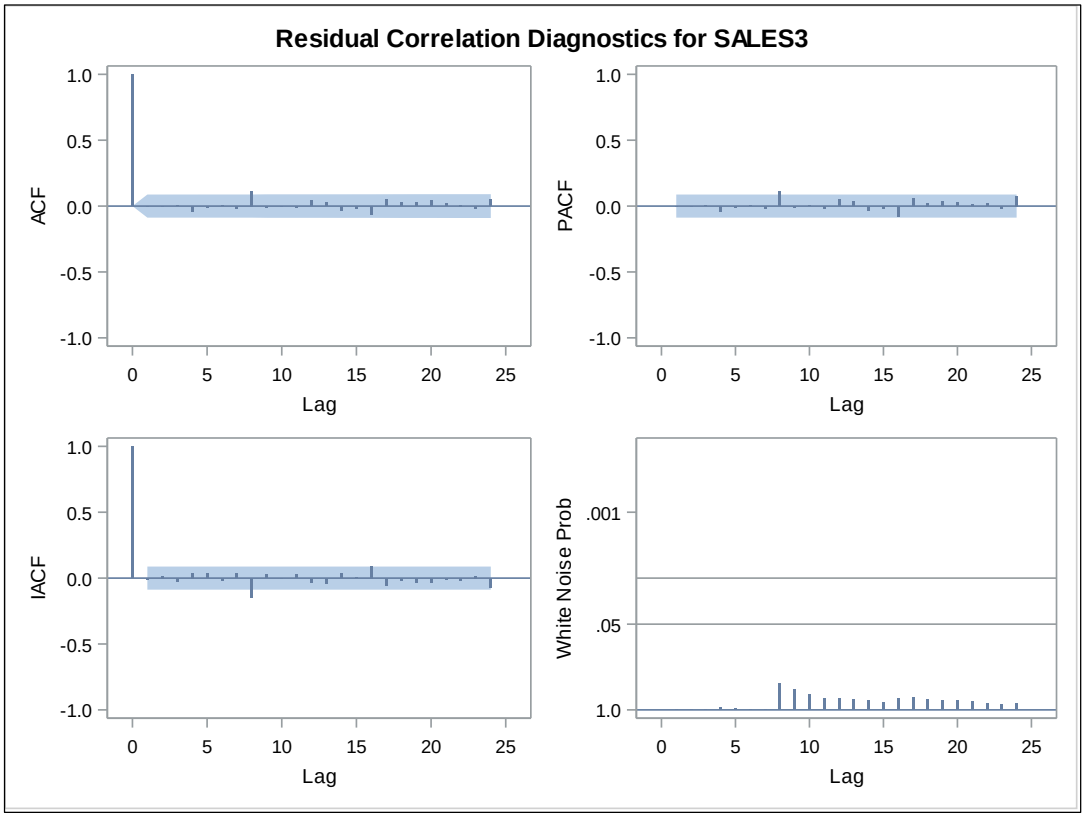
The RAMP parameter is still statistically significant.

Constant Estimate	42.21243
Variance Estimate	32.67885
Std Error Estimate	5.716542
AIC	3290.791
SBC	3299.298
Number of Residuals	520

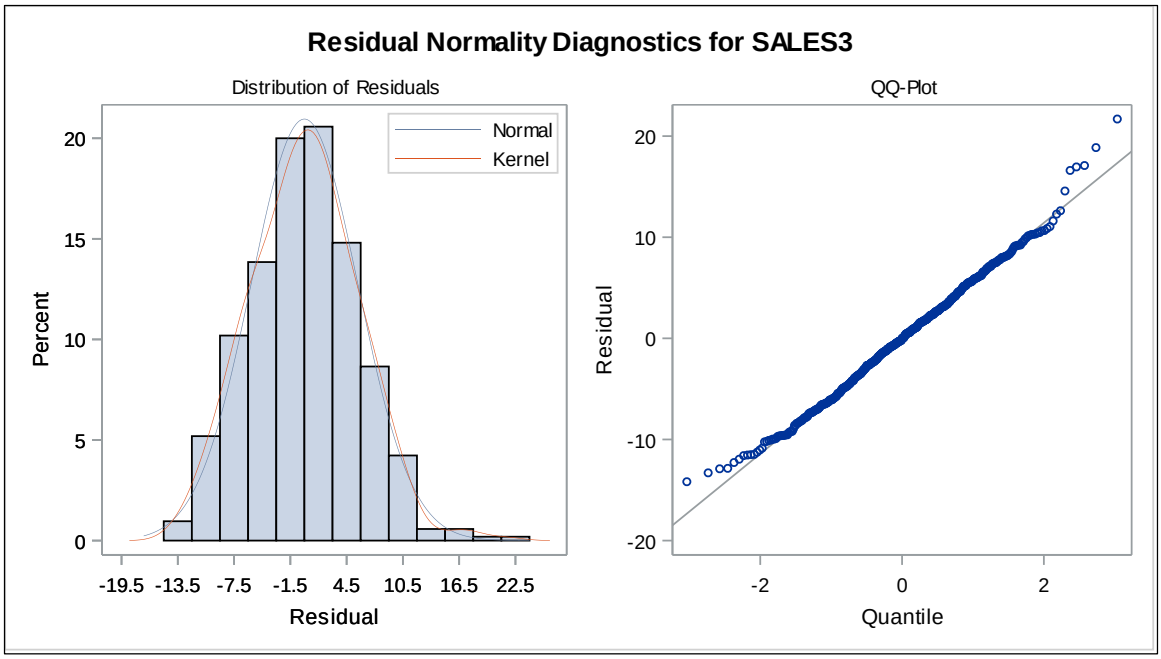
Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

The AIC is now 3290.791, which is slightly lower and smaller than that from the model including the autoregressive term. This indicates a slightly better fitting model.

Autocorrelation Check of Residuals									
To Lag	Chi-Square	DF	Pr > ChiSq	Autocorrelations					
6	1.17	6	0.9783	-0.000	-0.007	0.006	-0.044	-0.014	0.005
12	9.43	12	0.6656	-0.021	0.115	-0.008	0.001	-0.013	0.042
18	14.67	18	0.6843	0.030	-0.031	-0.020	-0.066	0.048	0.028
24	18.02	24	0.8022	0.027	0.044	0.018	0.008	-0.019	0.052
30	25.00	30	0.7251	0.089	-0.031	-0.052	-0.009	-0.024	0.022
36	32.42	36	0.6394	-0.030	0.069	-0.039	-0.073	-0.029	0.003
42	39.87	42	0.5650	-0.008	0.088	-0.031	-0.054	-0.036	-0.011
48	44.51	48	0.6168	-0.006	-0.001	-0.066	-0.054	-0.026	0.010



Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.



Model for variable SALES3	
Estimated Intercept	42.21243

Input Number 1	
Input Variable	RAMP
Overall Regression Factor	15.45018

The residuals from this model look like white noise.

Now run the model for the **SALES4** series, with a first order autoregressive parameter and a parameter for the input, **RAMP**.

```
   identify var=SALES4 crosscorr=(RAMP);
   estimate p=(1) input=(RAMP) method=ML;
run;
quit;
```

Partial Output

Correlation of SALES4 and RAMP	
Variance of input =	0.255917
Number of Observations	520

Maximum Likelihood Estimation							
Parameter	Estimate	Standard Error	t Value	Approx Pr > |t|	Lag	Variable	Shift
MU	42.70060	0.46830	91.18	<.0001	0	SALES4	0
AR1,1	0.45364	0.03917	11.58	<.0001	1	SALES4	0
NUM1	14.12213	0.59575	23.70	<.0001	0	RAMP	0

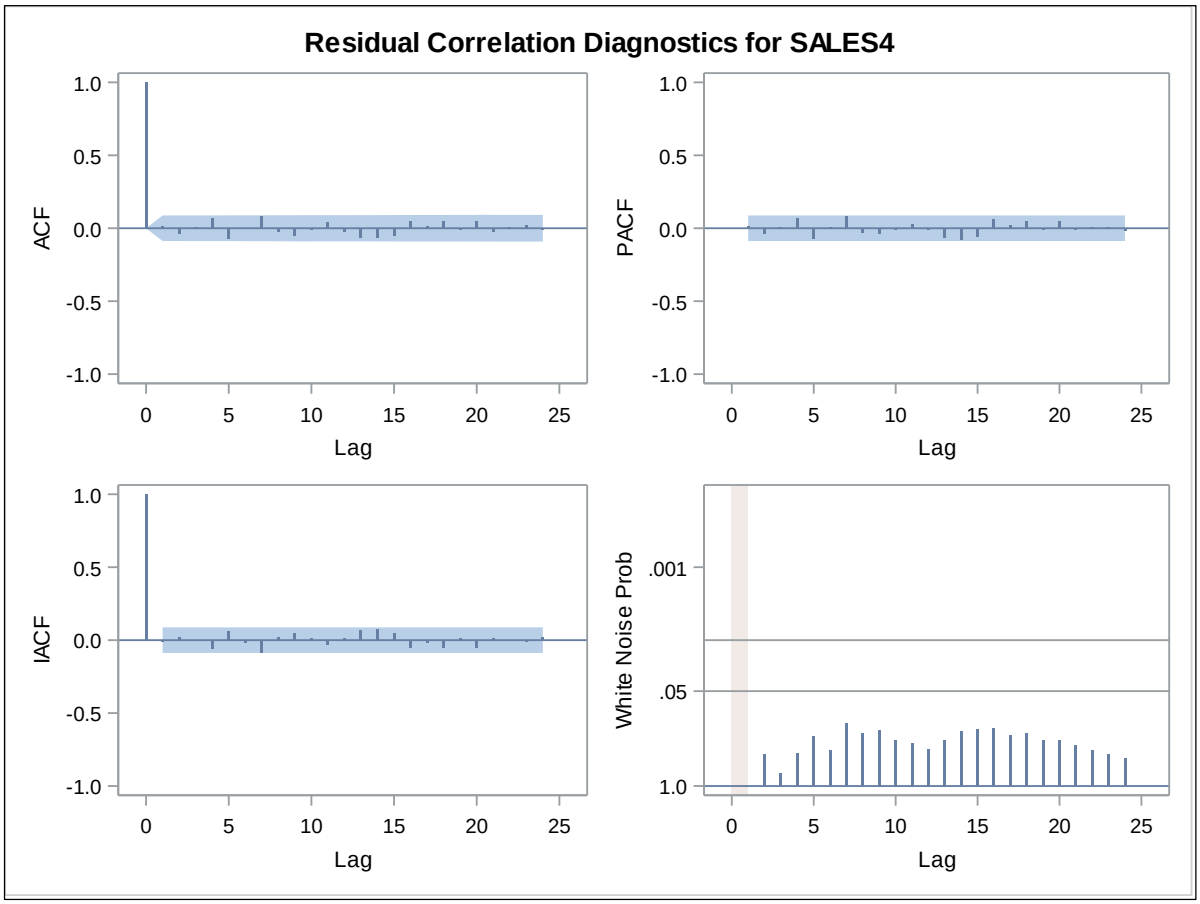
Both the autoregression parameter and the RAMP parameter are statistically significant.

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

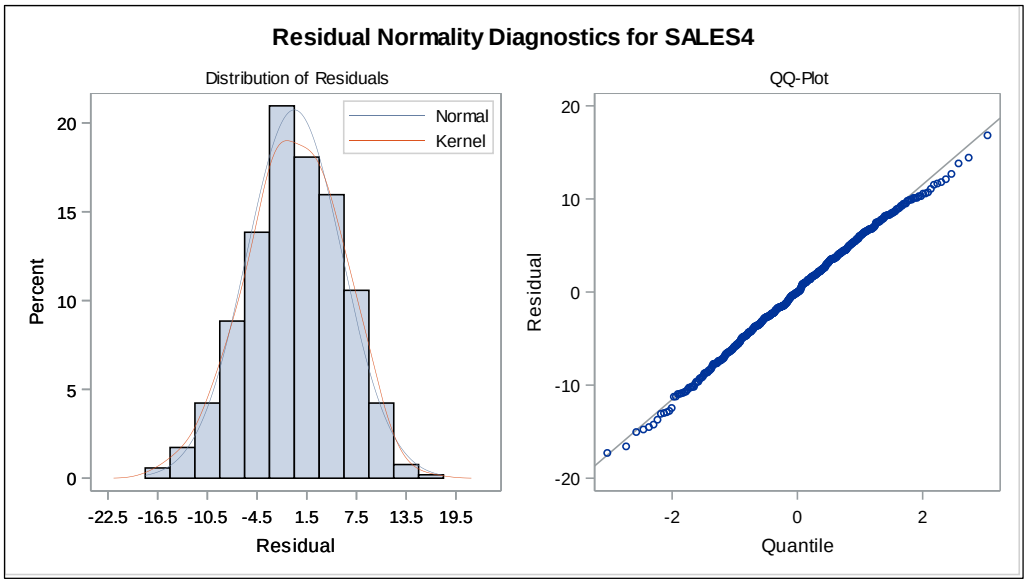
Constant Estimate	23.33003
Variance Estimate	33.43418
Std Error Estimate	5.78223
AIC	3303.899
SBC	3316.66
Number of Residuals	520

Autocorrelation Check of Residuals

To Lag	Chi-Square	DF	Pr > ChiSq	Autocorrelations					
6	5.85	5	0.3213	0.013	-0.037	0.006	0.068	-0.070	-0.000
12	12.61	11	0.3195	0.085	-0.024	-0.055	-0.007	0.038	-0.022
18	21.83	17	0.1915	-0.068	-0.067	-0.052	0.051	0.017	0.050
24	23.65	23	0.4235	-0.008	0.045	-0.024	0.008	0.023	-0.008
30	30.38	29	0.3955	0.017	-0.031	0.068	-0.075	-0.013	0.026
36	34.14	35	0.5093	-0.032	-0.039	-0.033	-0.034	-0.039	-0.022
42	41.18	41	0.4627	-0.053	0.068	-0.046	-0.039	-0.034	-0.013
48	48.64	47	0.4069	0.023	0.009	0.016	-0.104	0.006	0.036



Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.



The residuals appear to be white noise. The model seems to fit the series.

Model for variable SALES4	
Estimated Intercept	42.7006

Autoregressive Factors	
Factor 1:	1 - 0.45364 B**(1)

Input Number 1	
Input Variable	RAMP
Overall Regression Factor	14.12213

4. Validation and Forecasting of Rose Series 4

a. Using the **SALES4** series and the **STSM.ROSESERIES** data set, build an ARMA(1,0) model and forecast a holdout sample of the most recent 52 observations. Create an output data set titled **AR1_FORECAST** that is written to the **Work** library.

Recall that you saw this model in an earlier exercise. The residuals do not appear to be white noise. This indicates that there is still systematic variation that the model is not capturing. The residuals also indicate longer tails than would be expected if they were normally distributed. Forecast accuracy might suffer as a result of these model assumption violations.

Analyze the forecast plots. Visually, how does the ARMA(1,0) model appear to fit the holdout sample?

1) On the DATA tab, specify the appropriate data set, dependent variable, and time ID.
2) On the MODEL tab, specify the appropriate forecasting model and autoregressive order.
3) Also, on the MODEL tab, expand the **Plots** section and select **Selected plots**. Then clear all plot check boxes and make sure that both Forecast Plots options are checked.
4) On the OPTIONS tab, select **52** as both the number of periods to forecast and the number of periods to hold back. This forecasts the 52-period holdout sample.
5) Clear the check box for outlier detection.

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

6) Create an output data set called **AR1_forecast** on the OUTPUT tab. (This is used later for the MAPE calculations.)

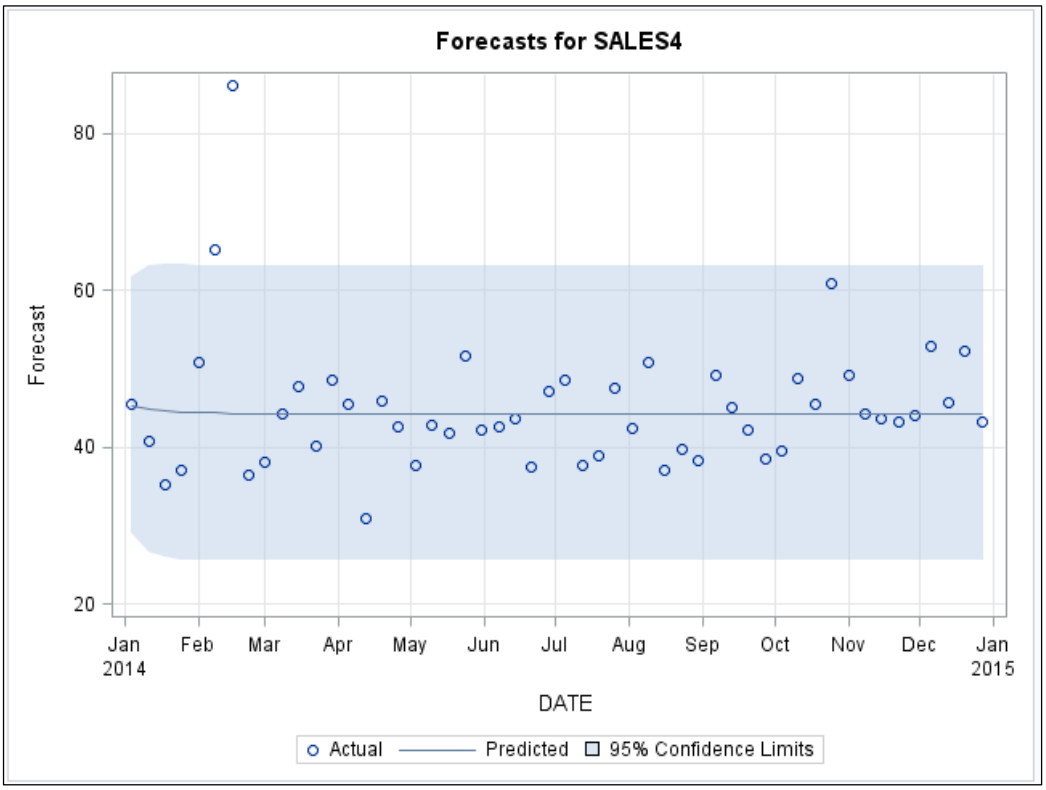
The code generated by SAS Studio is as follows:

```
proc arima data=WORK.TempSorted plots (only)=( forecast(forecast
forecastonly))
          out=WORK.AR1_forecast;
   identify var=SALES4;
   estimate p=(1) method=ML;
   forecast lead=52 back=52 alpha=0.05 id=DATE interval=week.7;
quit;
```

You can enter the SAS code directly.

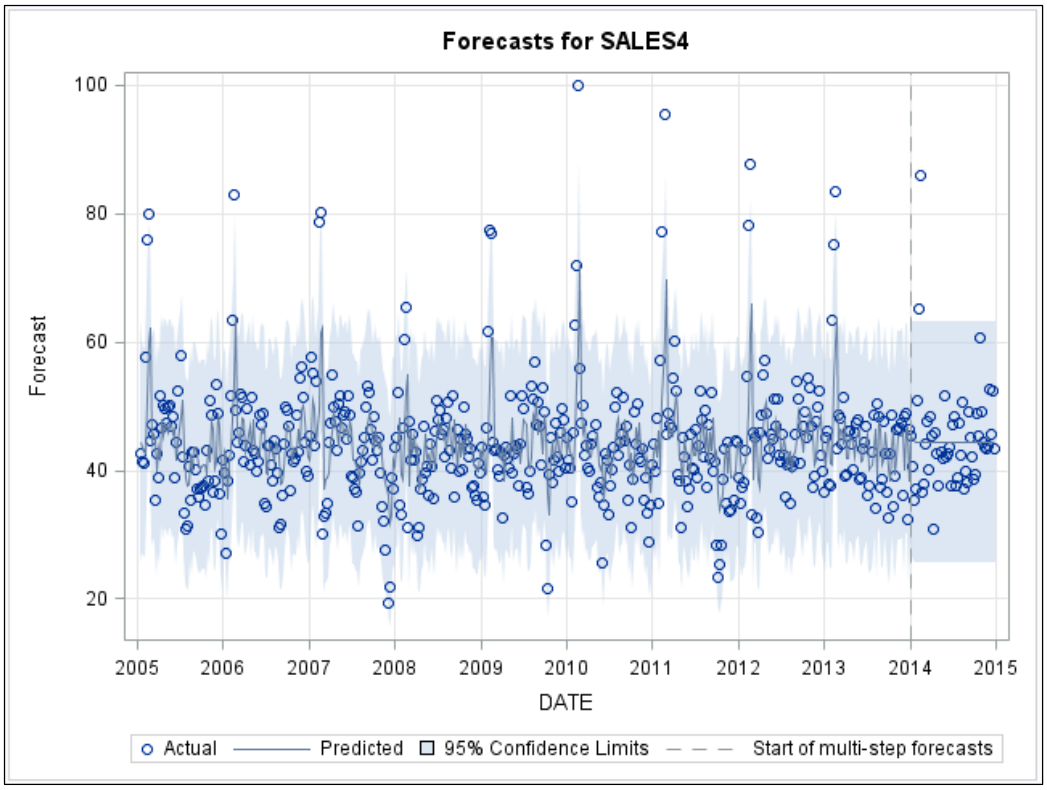
```
/* STSM02s05.sas */
/* Part A */
proc arima data=STSM.ROSESERIES
          plots(only)=forecast(forecast forecastonly))
          out=WORK.AR1_forecast;
   identify var=SALES4;
   estimate p=(1) method=ML;
   forecast lead=52 back=52 id=DATE;
quit;
```

Partial Output



The forecast plots show two weeks around Valentine's Day exceeding the 95% confidence intervals. The ARMA(1,0) model does not appear to account for the increase in sales leading up to Valentine's Day.

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.



The entire series is displayed, including the holdout sample, which starts at the dashed, vertical line.

b. Using the **SALES4** series, the **RAMP** input variable, and the **STSM.ROSESERIES** data set, build an ARMAX(1,0) model and forecast a holdout sample of the most recent 52 observations. Create an output data set titled **ARMAX1_FORECAST** that is written to the **Work** library.

As you saw in a previous exercise, the residuals appear to be white noise for all lags. This is a good sign that no systematic variation remains in the residuals. The Q-Q plot shows that the residuals are approximately normally distributed.

How well does the ARMAX(1,0) model seem to fit the holdout sample? Does this or the previous model appear to fit the holdout sample better?

Repeat the steps for the previous exercise, with the following exceptions:

1) On the MODEL tab, select the **ARIMAX** model type with autoregressive order **1**, and the input (Independent) variable, **RAMP**.

2) On the OUTPUT tab, create an output data set titled **ARMAX1_forecast**. (This is used later to calculate MAPE.)

Code generated by SAS Studio:

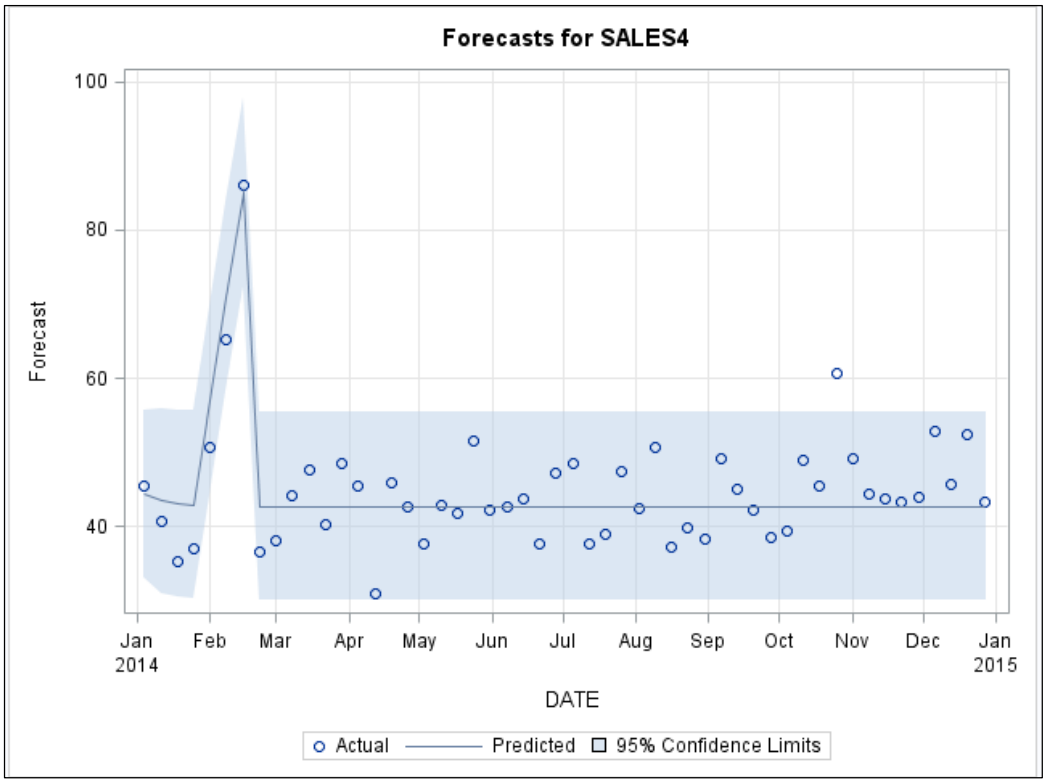
```
proc arima data=WORK.TempSorted plots
           (only)=(forecast( forecast forecastonly))
           out=WORK.ARMAX1_forecast;
   identify var=SALES4 crosscorr=(RAMP);
   estimate p=(1) input=(RAMP) method=ML;
   forecast lead=52 back=52 alpha=0.05 id=DATE interval=week.7;
quit;
```

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

You can write the SAS code directly as follows:

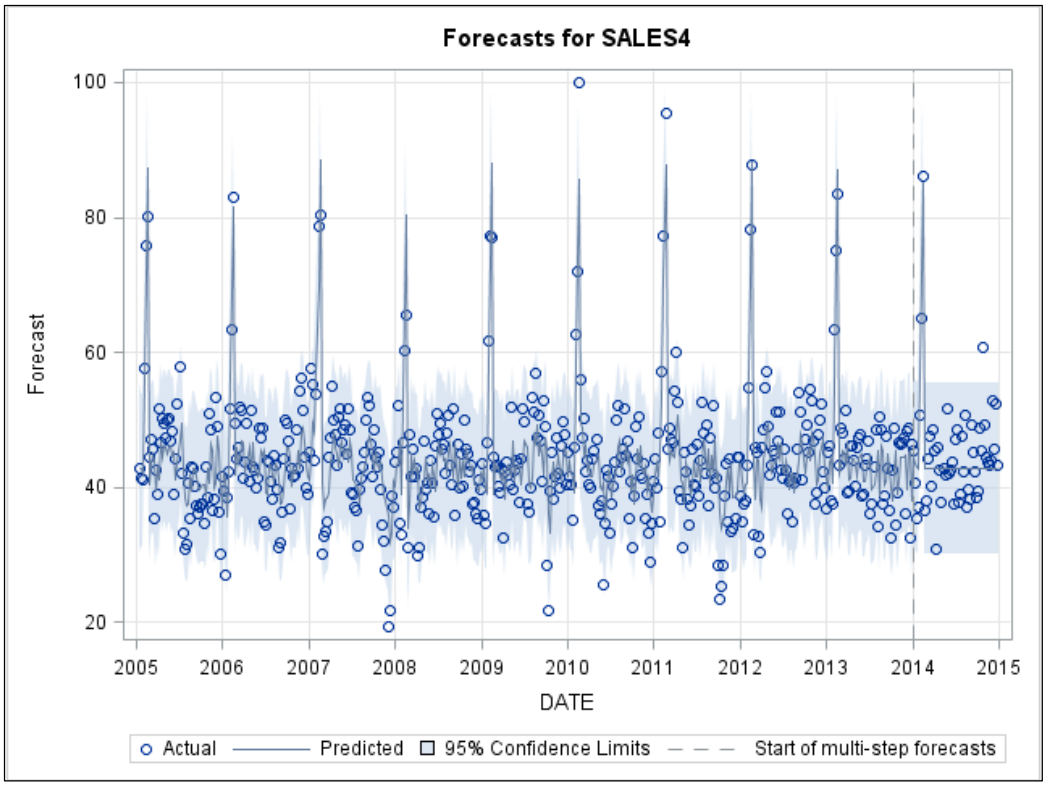
```
/* Part B */
proc arima data=STSM.ROSESERIES
           plots (only)=forecast(forecastonly)
           out=WORK.ARMAX1_forecast;
   identify var=SALES4 crosscorr=(RAMP);
   estimate p=(1) input=(RAMP) method=ML;
   forecast lead=52 back=52 id=DATE;
quit;
```

Partial Output



The first forecast graph displays only the 52 forecasted periods of the holdout sample. The Ramp input variable seems to account for the increase in SALES4 leading up to Valentine's Day. It appears to fit the SALES4 series better than the ARMA(1,0) model.

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.



The second graph displays the fit and holdout samples together in one graph.

c. Use the two output data sets you created, **AR1_forecast** and **ARMAX1_forecast**, to calculate MAPE for both models. Use one of the macros in **MAPE_Macro.sas** to calculate MAPE for both the ARMA(1,0) model and the ARMAX(1,0) model.

Which candidate model for **SALES4** provides the lower MAPE?

Use the %INCLUDE statement to first run the macro code. Then, use the macros. Use either **%MAPE** (the macro using PROC SQL code) or **%MAPE_D** (the macro using DATA step code).

```
/* Part C */
%include "&programloc\MAPEMacros.sas";

/* Using the MAPE macro */
%mape(ar1_forecast,Date,SALES4,52);
%mape(armax1_forecast,Date,SALES4,52);

/* Using the MAPE_D macro */
%mape_d(indsn=work.ar1_forecast,series=sales4,holdback=52);
%mape_d(indsn=work.armax1_forecast,series=sales4,holdback=52);
```

Output Using the **%MAPE** Macro

Obs	AR1_MAPE
1	0.11308

Obs	ARMAX1_MAPE
1	0.094768

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

Output Using the **%MAPE_D** Macro

Series	Model	MAPE
sales4	work.ar1_forecast	0.11308

Series	Model	MAPE
sales4	work.armax1_forecast	0.094768

Running either macro above shows that the ARMAX(1,0) model produces a lower MAPE than the ARMA(1,0) model. Therefore, when you forecast future, unobserved periods in the next section of the exercise, the ARMAX(1,0) model is used.

d. Use the **STSM.ROSESERIES_F** data set and the best candidate model (ARMAX(1,0)) to forecast the next 11 future, unknown time periods of rose sales.

1) On the DATA tab, specify the appropriate data set, dependent variable, and time ID.
2) On the MODEL tab, specify the winning candidate model. The ARMAX(1,0) model with the RAMP input variable produced the lowest MAPE, so it is used to forecast the future 11 unknown periods.
3) Be sure to select both Forecast Plots check boxes and clear all other plot check boxes. Unlike the demonstration, you are forecasting the next 11 periods, not only one period, so the forecast plots are more informative.
4) On the OPTIONS tab, choose to forecast the next 11 periods while holding back 0 periods. This ensures that both the fit and holdout samples are used.

For this exercise, an output data set is not necessary and therefore not created. In practice, however, you might find it helpful to output a data set with the forecasted periods. As shown earlier, this can be done on the OUTPUT tab.

SAS Studio Generated Code

```
proc arima data=WORK.TempSorted plots
           (only)=(forecast(forecast forecastonly))
           out=WORK.forecast_out;
   identify var=SALES4 crosscorr=(RAMP);
   estimate p=(1) input=(RAMP) method=ML;
   forecast lead=11 back=0 alpha=0.05 id=DATE
            interval=week.7 printall;
quit;
```

Alternatively, you can write the SAS code directly as follows:

```
/* Part D */
proc arima data=STSM.ROSESERIES_F
           plots(only)=forecast(forecast forecastonly);
   identify var=SALES4 crosscorr=(RAMP);
   estimate p=(1) input=(RAMP) method=ML;
   forecast lead=11 back=0 id=DATE interval=week.7;
quit;
```

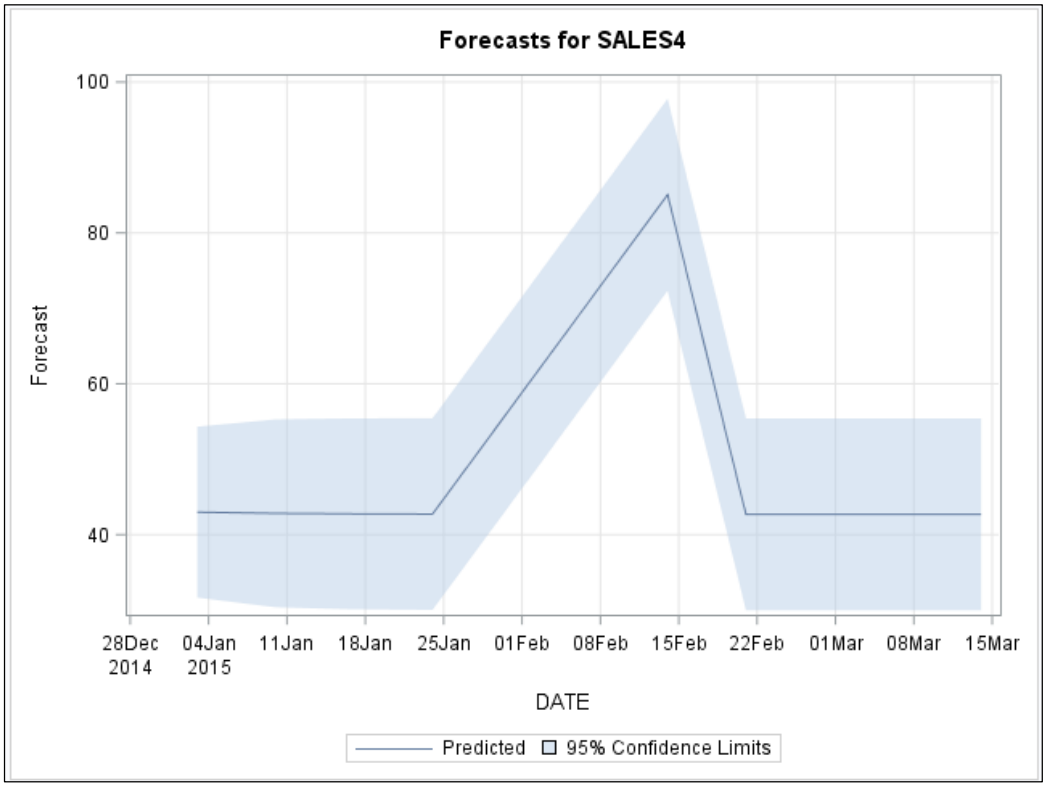
Running the program produces forecasts for the next 11 periods. The table below lists partial output from the RESULTS tab. For space considerations, observations 1 through 520 were not included.

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

When the PRINTALL option is removed from the code, the forecast table displays only the forecasted values from the forecasted period.

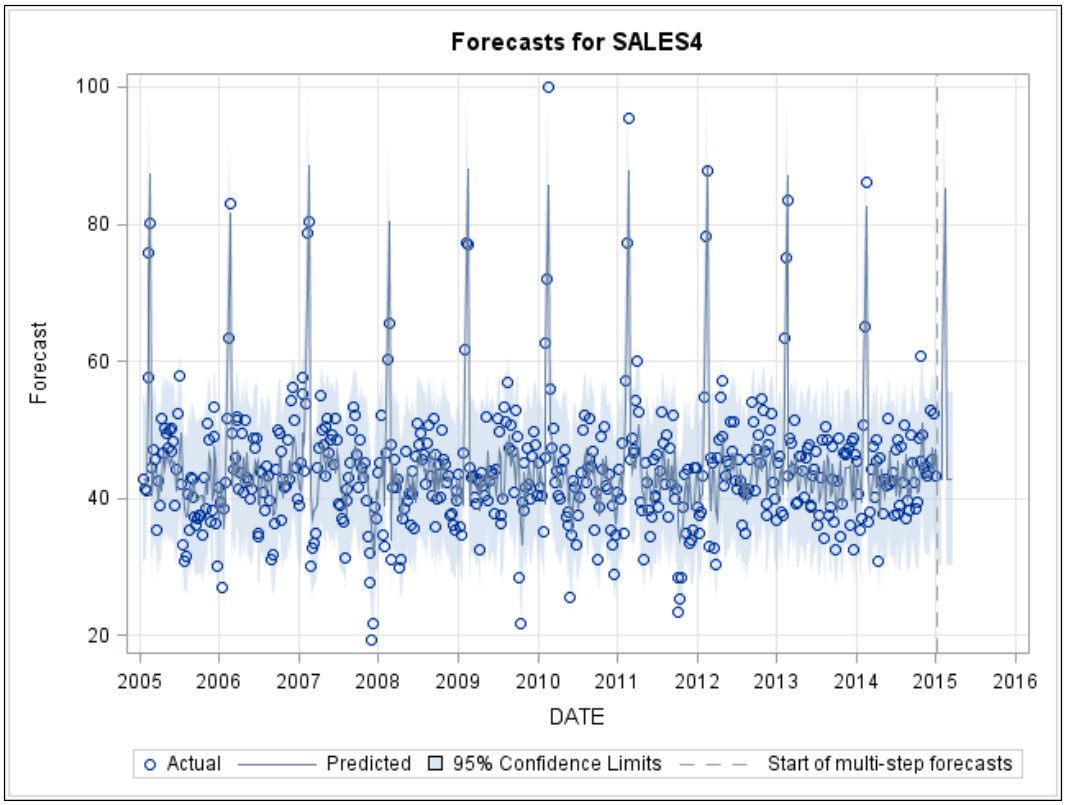
Forecasts for variable SALES4						
Obs	Forecast	Std Error	95% Confidence Limits		Actual	Residual
...						
521	42.9785	5.7822	31.6455	54.3114	.	.
522	42.8267	6.3494	30.3821	55.2712	.	.
523	42.7578	6.4599	30.0966	55.4190	.	.
524	42.7265	6.4824	30.0212	55.4318	.	.
525	56.8345	6.4870	44.1201	69.5489	.	.
526	70.9502	6.4880	58.2340	83.6664	.	.
527	85.0694	6.4882	72.3528	97.7860	.	.
528	42.7017	6.4882	29.9850	55.4184	.	.
529	42.7011	6.4882	29.9844	55.4178	.	.
530	42.7008	6.4882	29.9841	55.4175	.	.
531	42.7007	6.4882	29.9840	55.4174	.	.

The first graph displays only the 11 forecasted periods.



The second graph displays all observations including the predicted 11 periods to the right of the vertical dashed line.

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.



End of Solutions

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

Solutions to Student Activities (Polls/Quizzes)

2.01 Multiple Answer Poll – Correct Answers

Which of the following are true?

a. Failing to reject the null hypothesis of the white noise probability test implies that the series is white noise.
b. First order autocorrelation is the correlation between the current value and the immediately preceding value.
c. A time series requires at least one measure of chronological time.
d. You can now accurately forecast future spins of the roulette wheel and share future winnings with your instructor.
e. A white noise process implies that there is no autocorrelation.

17

2.02 Multiple Answer Poll – Correct Answers

Which of the following is a stationary process?

a. a series that, when graphed, appears to exhibit a constant mean and variance across all time periods
b. a necessary component needed before ARMA modeling can occur
c. often the result after differencing a nonstationary series
d. the paper and envelopes used for writing correspondence

39

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

2.03 Multiple Answer Poll – Correct Answers

Which would be an example of a stochastic regressor?

a. ambient indoor air temperature
b. number of people at a beach
c. occurrence of a full moon
d. occurrence of a solar flare
e. United States prime lending rate
f. your company's mortgage rate for prime customers

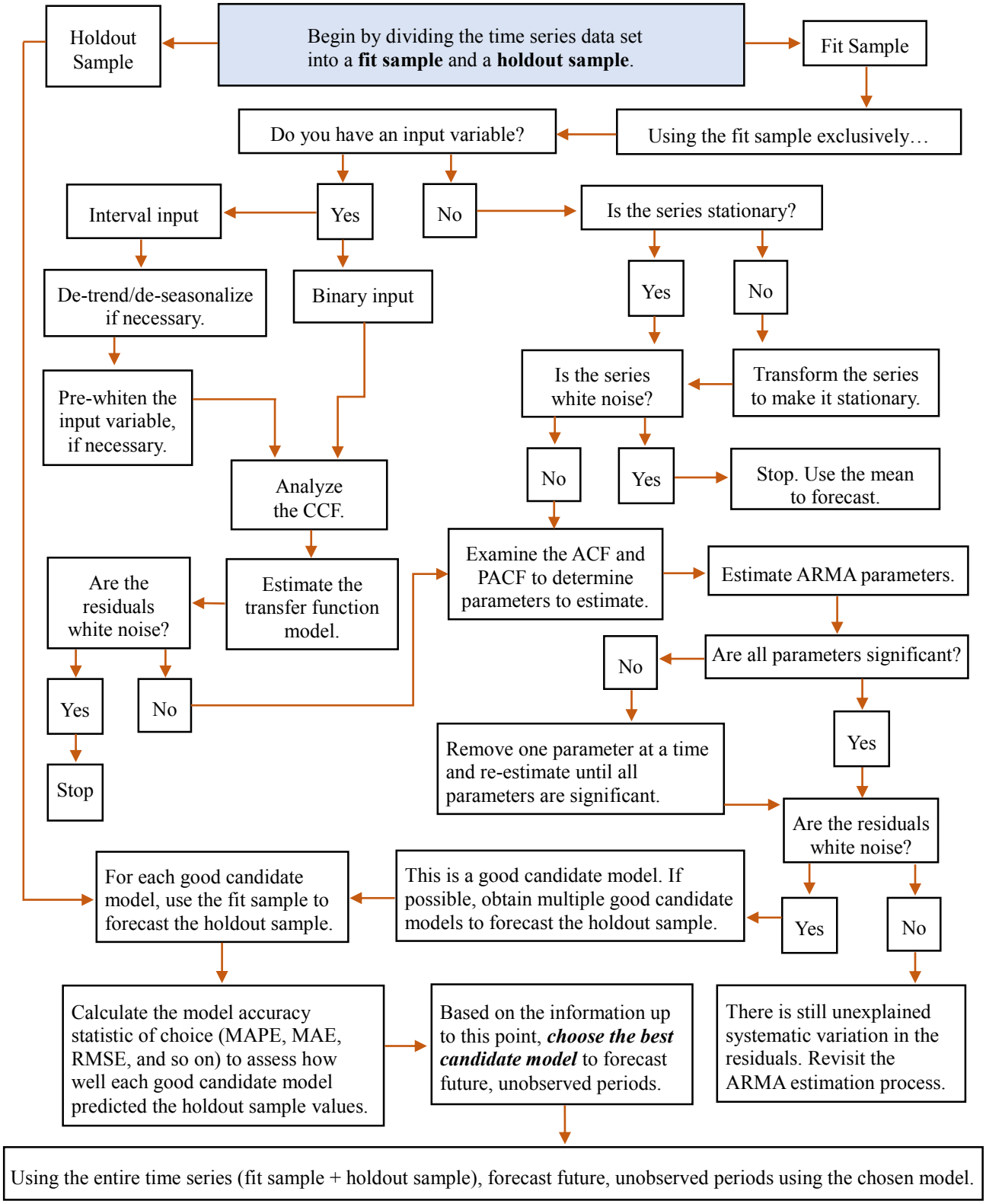
64

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

2.7 Chapter Summary

Below is a flow chart showing the necessary steps that are needed to fit an ARMA or ARMAX model to a time series. These topics were discussed in detail throughout the chapter, and should be used as a reference when you revisit this material.

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.



Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

Chapter 3 Exponential Smoothing Models

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

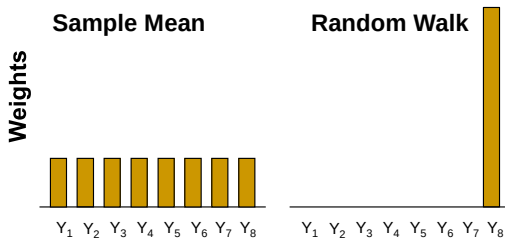
3.1 Exponential Smoothing Models

Objectives

- Explore weighted average models and exponential smoothing.
- Compare and contrast simple mean, random walk, and exponential smoothing models.

2

Weighted Average Examples



Weights applied to past values to predict Y_9

3

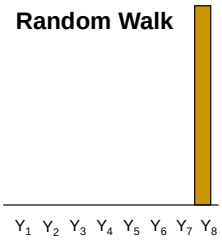
Weighted averaging is a simple and intuitive method for smoothing a time series and forecasting future values from your past observations. In weighted averaging, weights are applied to past values in such a way that they predict the value of the next time point in the series. The values of these weights are selected based on the determination of the importance of these past observations in determining the future. Look at two, very commonly used styles of weighted averages: sample mean and random walk.

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

Weighted Average Example: Random Walk

$$\hat{Y}_{n+1} = \sum_{t=1}^{n} w_t Y_t = Y_n$$

$$w_n = 1, w_t = 0 \text{ for } t = 1,2,\ldots,n-1$$



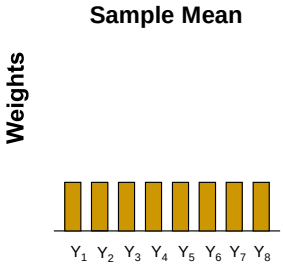
$$\hat{Y}_9 = Y_8$$

A random walk forecast is a weighted average where all weights are 0 except the most recent, which is 1.

4

A common example of a weighted average is the random walk. In this setup, all weights for previous observations are 0 except for the most recent, which receives a weight of 1. In the random walk, where you will be at the next time point is related only to where you are immediately prior.

Weighted Average Example: Simple Moving Average



$$\hat{Y}_9 = \frac{1}{8}\sum_{t=1}^{8} Y_t$$

$$\hat{Y}_{n+1} = \sum_{t=1}^{n} w_t Y_t = w_1 Y_1 + w_2 Y_2 + \cdots + w_n Y_n$$

$$= \sum_{t=1}^{n} \frac{1}{n} Y_t = \frac{1}{n}\sum_{t=1}^{n} Y_t = \bar{Y}$$

$$w_t = \frac{1}{n}$$

The mean is a weighted average where all weights are the same.

5

In the sample mean example, equal weight is placed on each of the previous n observations in the calculation of the prediction of the next series value. In this setup, the weight of $(1/n)$ is applied to each of the previous n observations and zero to all remaining prior observations.

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

Simple Moving Average

Disadvantages

- cannot be used on the first *n*-1 terms of the time series without adding other terms by some other means
- can be influenced by extreme values within the window
- requires the retaining of the most recent *n* observations to produce forecasted value

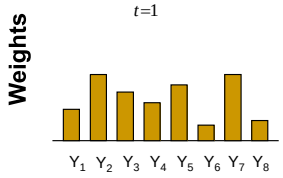
6

Despite the simplicity of the simple moving average, there are disadvantages to its usage. With the requirement of the *n* terms to produce the weighted average, this cannot be applied to the first (*n*-1) terms of the time series unless extrapolation beyond the data is applied. Another disadvantage is that a simple moving average shares the same issues with extreme values as a typical mean. Just as the mean is not robust to outliers, extreme values can make their presence known within a simple moving average. Finally, in the production of forecasts that use a simple moving average, the most recent *n* observations must be retained.

Weighted Average Example: Weighted Moving Average

$$\hat{Y}_{n+1} = \sum_{t=1}^{n} w_t Y_t = w_1 Y_1 + w_2 Y_2 + \cdots + w_n Y_n$$

$$\sum_{t=1}^{n} w_t = 1$$

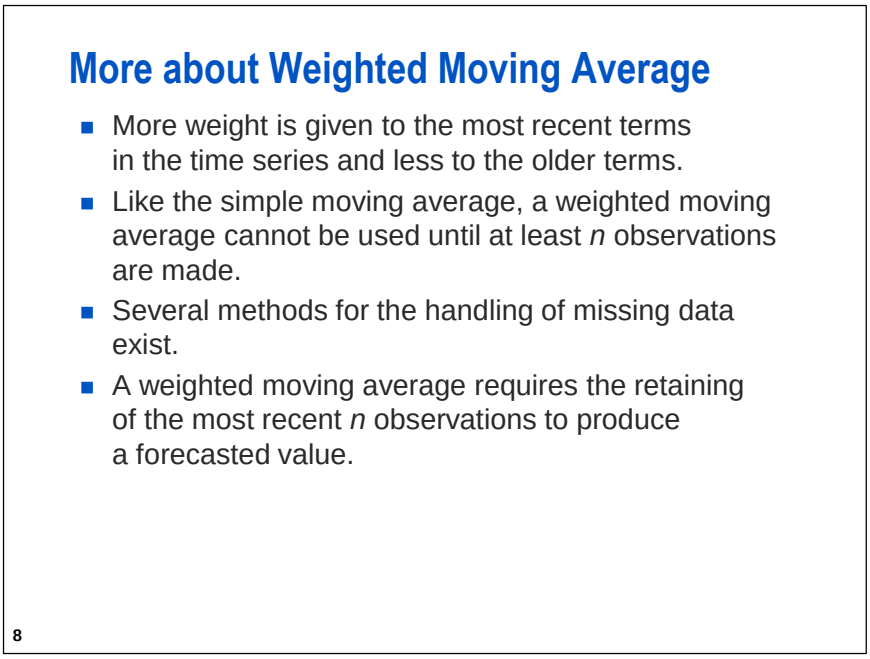


The mean is a weighted average where not all weights are the same.

7

When you use a weighted moving average, it is not a requirement that the weights of the prior *n* observations be equal. In this example, the weights can vary across the previous observations in the creation of the forecasted value.

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

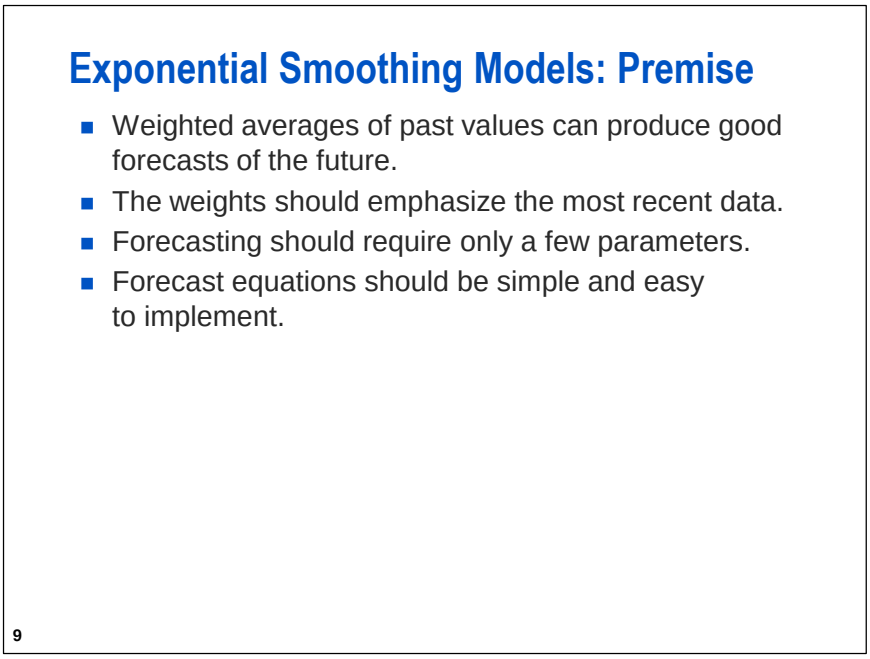


In many weighted moving average setups, where the weights are not equal, more weight is given to the more recent values in the series and less to the older observations. The premise is that the forecasted value is more like the observations immediately prior to it and less like those farther in the past. Even including this alteration to the weights, this setup cannot be used until the necessary *n* observations occur in the series, unless extrapolation is allowed.

If data is missing from a series, there are several methods to accommodate the data. These range from using the overall mean of the series to the one-step-ahead forecasting that is typically used in exponential smoothing (Yaffee and McGee 2000).

Like the simple moving average, the weight moving average requires the retaining of the most recent *n* observations to generate the forecasts.

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.



An exponential smoothing model (ESM) was first suggested by Robert Goodell Brown (1959, 1962) and added to by Charles C. Holt (1960). This model is used primarily for the creation of forecasting models for inventory control systems. ESMs formulate forecasts using a "smoothing" method of weighted averages. In the construction of the forecasts, more recent observations are given more weight than observations in the more distant past. The "exponential" is derived from the fact that weights not only diminish over time, but they do so exponentially (Fomby 2008). ESMs have the added bonus that only a few parameters are required in the forecasting model and these equations are simple to implement.

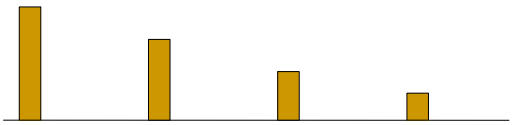
The seven common exponential smoothing models are supported by the ESM procedure. In addition, a few exponential smoothing models are not as common and are not supported. For example, triple exponential smoothing models use third differencing, that is, three differences. Such models are addressed in textbooks, but rarely provide good forecasts for real data. When they do offer acceptable results, the application is usually highly specialized.

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

The Exponential Smoothing Coefficient

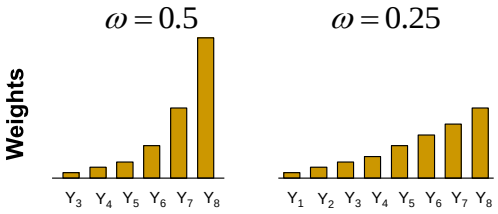
Forecast Equation

$$\hat{Y}_{t+1} = \omega Y_t + (1-\omega)\hat{Y}_t$$
$$= \omega Y_t + (1-\omega)[\omega Y_{t-1} + (1-\omega)\hat{Y}_{t-1}]$$
$$= \omega Y_t + \omega(1-\omega)Y_{t-1} + (1-\omega)^2\hat{Y}_{t-1}$$
$$= \omega Y_t + \omega(1-\omega)Y_{t-1} + (1-\omega)^2[\omega Y_{t-2} + (1-\omega)\hat{Y}_{t-2}]$$
$$= \omega Y_t + \omega(1-\omega)Y_{t-1} + \omega(1-\omega)^2 Y_{t-2} + \omega(1-\omega)^3 Y_{t-3} + \cdots$$



10

Simple Exponential Smoothing



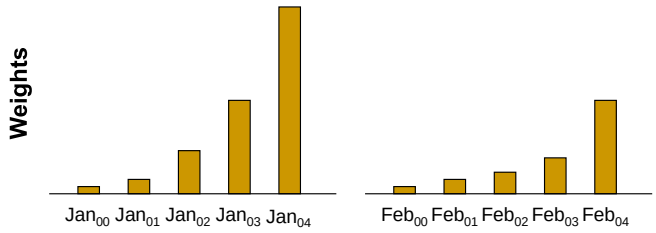
Weights applied to past values to predict Y_9

As the parameter increases, the emphasis on the most recent values increases.

11

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

Exponential Smoothing for Seasonal Data



Weights decay with respect to the seasonal factor.

12

To obtain the exponential decay, the absolute values of the smoothing parameter must be less than one ($|\omega| < 1$). The SAS/ETS documentation gives an explanation for weights near zero or one.

Exponential Smoothing Models (ESM)

- 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

13

Because there are exactly seven models, a trial-and-error method becomes an effective strategy for model selection in the age of high-speed computers.

Simple exponential smoothing should be used when the time series data has no trend and no seasonality. The ARIMA model that is equivalent to the simple exponential smoothing model is the ARIMA(0,1,1).

Double (Brown) exponential smoothing should be used when the time series data has trend but no seasonality. The ARIMA model that is equivalent to the linear exponential smoothing model is ARIMA(0,2,2).

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

Seasonal exponential smoothing should be used when the time series data has no trend but has seasonality.

Winters additive or multiplicative exponential smoothing models should be used when the time series data has trend ***and*** seasonality (Fomby 2008).

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

14

ODS Graphics

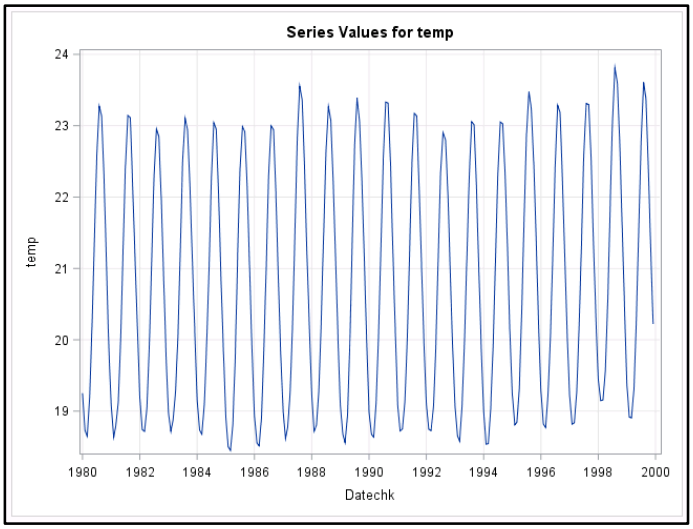
ODS Graph Name	Plot Description	PLOT= Option
ErrorACFNORMPlot	standardized autocorrelation of prediction errors	ACF
ErrorACFPlot	autocorrelation of prediction errors	ACF
ErrorHistogram	prediction error histogram	ERRORS
ErrorCorrelationPlots	prediction error plot panel	CORR
ErrorIACFNORMPlot	standardized inverse autocorrelation of prediction errors	IACF
ErrorIACFPlot	inverse autocorrelation of prediction errors	IACF
ErrorPACFNORMPlot	standardized partial autocorrelation of prediction errors	PACF
ErrorPACFPlot	partial autocorrelation of prediction errors	PACF
ErrorPeriodogramPlot	periodogram of prediction errors	PERIODOGRAM
ErrorPlot	plot of prediction errors	ERRORS
ErrorSpectralDensityPlot	combined periodogram and spectral density estimate plot	SPECTRUM
ErrorWhiteNoiseLogProbPlot	white noise log probability plot of prediction errors	WN
ErrorWhiteNoiseProbPlot	white noise probability plot of prediction errors	WN
ForecastsOnlyPlot	forecasts only plot	FORECASTSONLY
ForecastsPlot	forecasts plot	FORECASTS
LevelStatePlot	smoothed level state plot	LEVELS
ModelForecastsPlot	model and forecasts plot	MODELFORECASTS
ModelPlot	model plot	MODELS
SeasonStatePlot	smoothed season state plot	SEASONS
TrendStatePlot	smoothed trend state plot	TRENDS

15

Because of ODS Graphics capabilities, the exponential smoothing procedure is capable of producing many ODS graphs during the analysis. This table displays the ODS graph name, a brief plot description, and the PLOT= option that generates the image. The ODS graph name can be used in conjunction with ODS SELECT, ODS EXCLUDE, or ODS OUTPUT statements to restrict displayed output or to produce an output data set for further use.

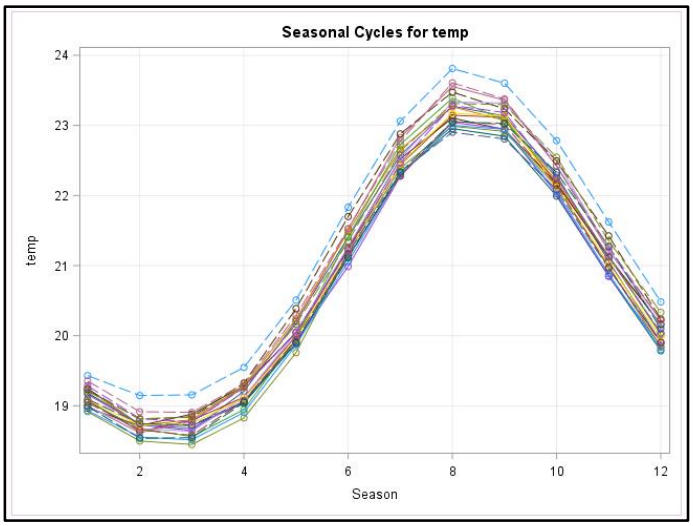
Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

ESM ODS Output



16

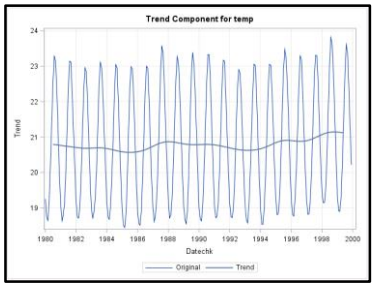
ESM ODS Output

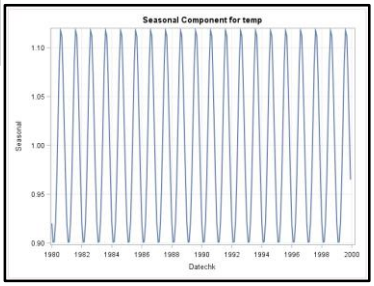


17

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

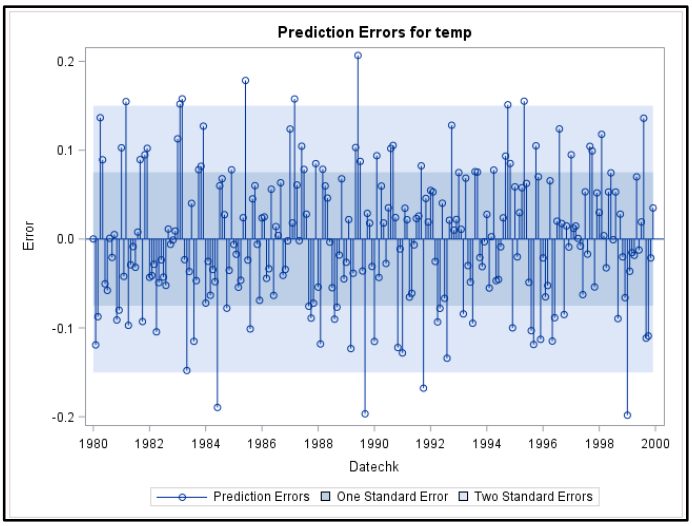
ESM ODS Output





18

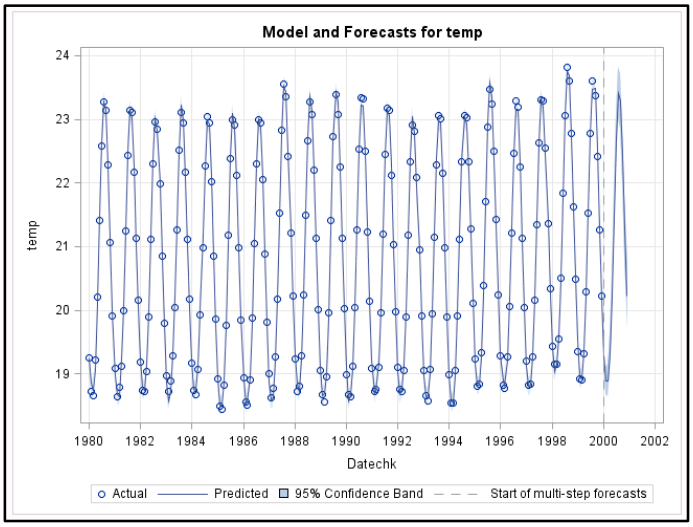
ESM ODS Output



19

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

ESM ODS Output



20

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;
```

21

Selected ESM procedure options and statements:

OUTEST= names the output data set to contain the model parameter estimates and the associated test statistics and probability values. This data set is useful for evaluating the significance of the model parameters and understanding the model dynamics.

OUTFOR= names the output data set to contain the forecast time series components (actual, predicted, lower confidence limit, upper confidence limit, prediction error, prediction standard error). This is useful for displaying the forecasts in tabular or graphical form.

OUTSTAT= names the output data set to contain the statistics of fit (goodness of fit). This is useful for evaluating how well the model fits the series.

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

OUTSUM= names the output data set to contain the summary statistics and the forecast summation. This is useful when forecasting large numbers of series and a summary of the results is needed.

SEASONALITY= specifies the length of the seasonal cycle.

PLOT= specifies the graphical output that is desired.

PRINT= specifies the printed output that is desired.

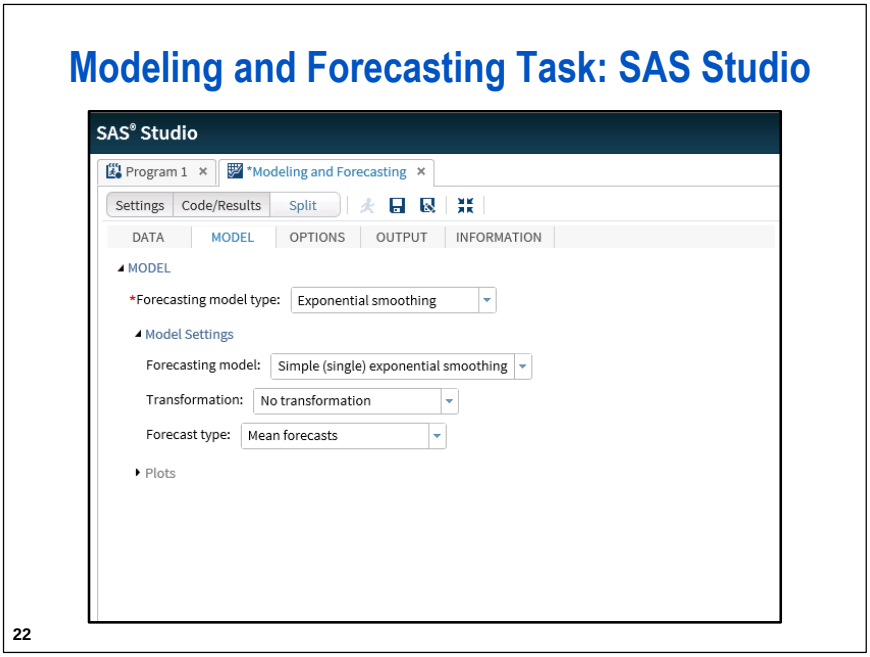
LEAD= specifies the number of periods ahead to forecast. The default is LEAD=12.

ID names a numeric variable, assumed to be SAS data or time data valued, that identifies observations in the input and output data sets.

INTERVAL= specifies the frequency of the input time series such as quarterly, monthly, weekly, and so on.

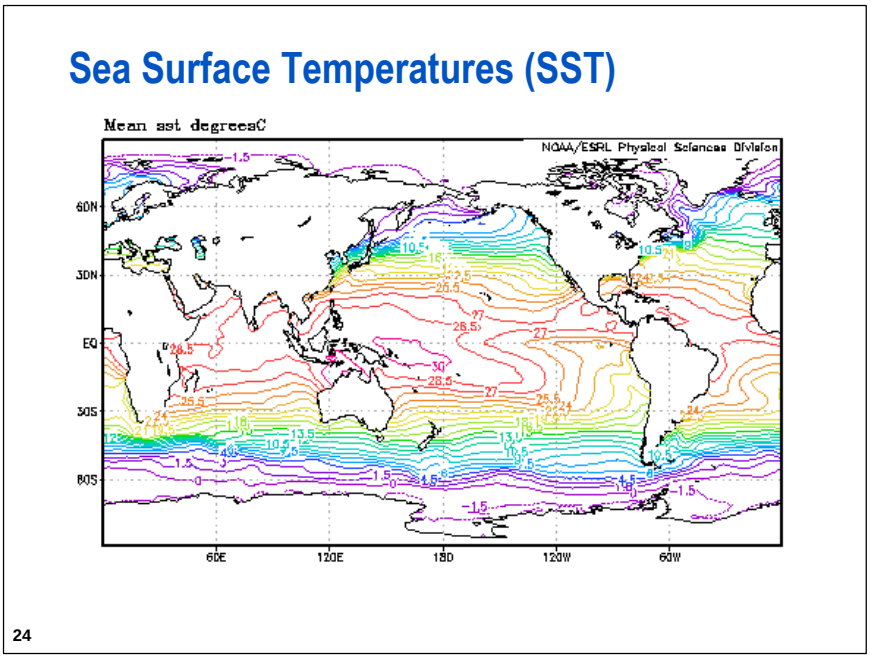
FORECAST lists the numeric variables in the DATA= data set whose accumulated values are the time series to be modeled and forecast.

MODEL= specifies the forecasting model to be used to forecast the time series. The default is MODEL=SIMPLE, which performs simple exponential smoothing.



Exponential smoothing models can be performed using the Modeling and Forecasting task in SAS Studio. After you choose the data set to analyze and set the necessary roles for variables on the DATA tab, exponential smoothing can be selected as the forecasting model type on the MODEL tab. Within the model settings, options that include the type of forecasting model to use can be selected.

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.



The demonstration examines North Atlantic sea surface temperatures (SST). The data in **STSM.SST** and the image were provided by NOAA/OAR/ESRL PSD, Boulder, Colorado, USA (**http://www.esrl.noaa.gov/psd/**). The data provide monthly average sea surface temperatures for the North Atlantic.

The data file was augmented into the proper time series formatting. It contains the following variables:

Datechk SAS date variable providing the month and year of the observation

Temp numeric average sea surface temperature for that respective time point

The data spans from the mid-1800s until the present. The class focus is reduced to the 1980s and 1990s.

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

Analyzing Sea Surface Temperatures Using SAS Studio

In this demonstration, you use both the Time Series Exploration task and the Modeling and Forecasting task in SAS Studio to explore the data set and generate an exponential smoothing model for sea surface temperature. The **SST** data set can be found in the **STSM** library. Use the Exploration task to investigate the possible inclusion of a trend or seasonality component. Then use the Modeling and Forecasting task to generate the appropriate exponential smoothing models. Include forecasts 12 months into the future.

1. Expand the **Tasks** area in the left navigation pane as well as the **Forecasting** subsection. Double-click the **Time Series Exploration** task. Click the **Maximized View** button.
2. On the DATA tab, select the **SST** data set within the **STSM** library.
3. Click the plus sign (+) next to **Dependent variable**. Select **temp**. Click **OK**. For this model, there is no independent variable.
4. Click and expand the **Additional Roles** section. Click the plus sign (+) next to **Time ID**. Select **Datechk**. Click **OK**. Notice that the Properties section is updated to reflect the interval of monthly data.
5. Click the **Analyses** tab. Select the **Seasonal cycles** check box. Do ***not*** clear the **Time Series** check box. Under Autocorrelation Analysis, make sure that the **Perform autocorrelation analysis** check box is selected. Change the drop-down box beside **Select plots** to display to read **Selected plots**.
6. Under Plots, select the **Autocorrelation analysis panel** and **White noise probability test (log scale)** check boxes.
7. Under Decomposition Analysis, make sure that the **Perform decomposition analysis** check box is selected. Change the drop-down box beside **Select plots** to display to read **Selected plots**.
8. Select all four check boxes (**Decomposition panel**, **Components**, **Seasonally adjusted series**, and **Seasonally adjusted series (percent change)**) under Plots.
9. When **Components** is selected, select **Trend component** and **Seasonal component** only.

   Alternatively, you can write the SAS code directly as follows:

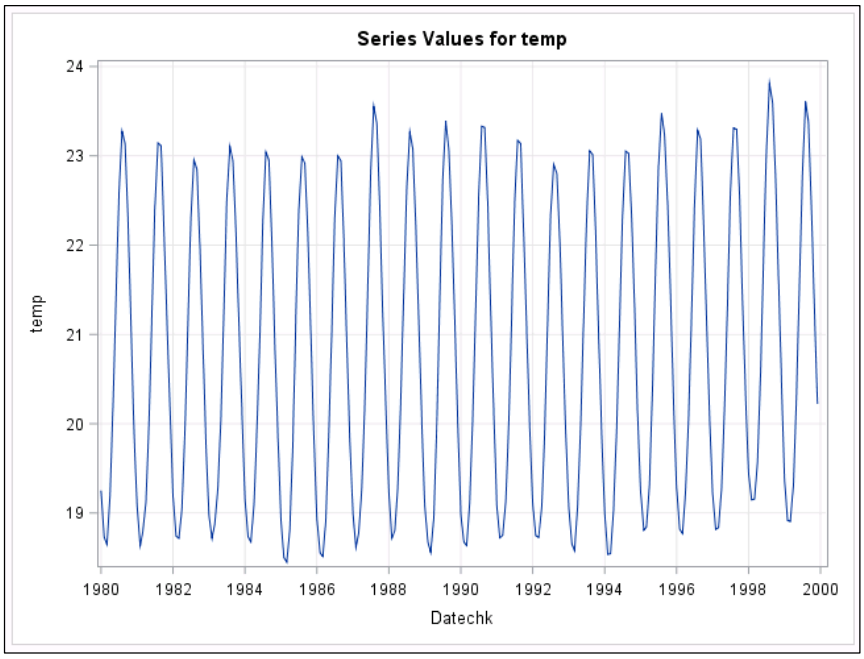
```
/* STSM03d01.sas */
proc timeseries data=STSM.SST
                seasonality=12
                plots=(series cycles corr wn
                       decomp sa pcsa tc sc);
   id Datechk interval=month;
   var temp;
   decomp sa pcsa tc sc / mode=multoradd;
   ods exclude WhiteNoiseProbabilityPlot;
run;
```

10. Click the running person icon to execute the task.

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

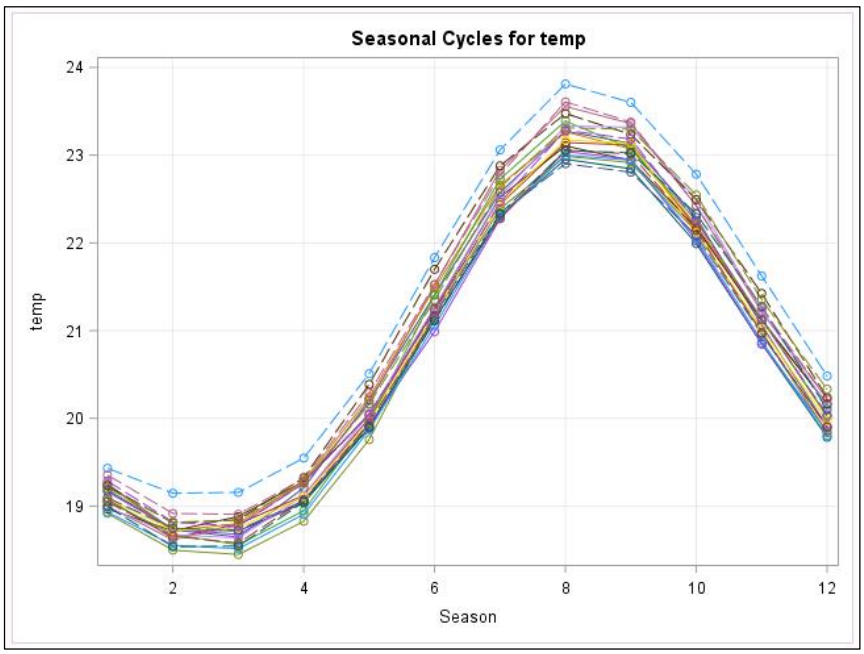
Selected Output

The output begins with a few tables that display information about the data set as well as the time series variable.



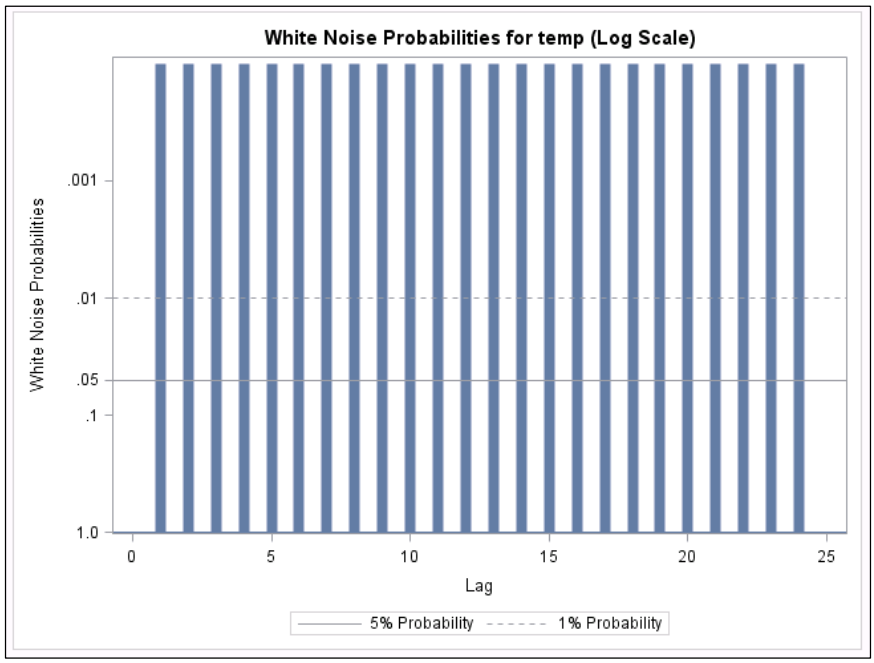
The time series plot provides a visual perspective of the dependent variable **temp** across the time ID variable **Datechk**. From this plot, you can see a clear seasonality to the data. However, it is difficult to determine whether there is a trend.

The series plot also gives an impression of whether the seasonality is additive or multiplicative. In the plot, notice that the peaks and valleys of the seasonal fluctuation appear consistent. This indicates an additive seasonality. If this plot showed peaks and valleys of the fluctuations becoming closer together or farther apart as time passed, this would indicate a multiplicative seasonality.

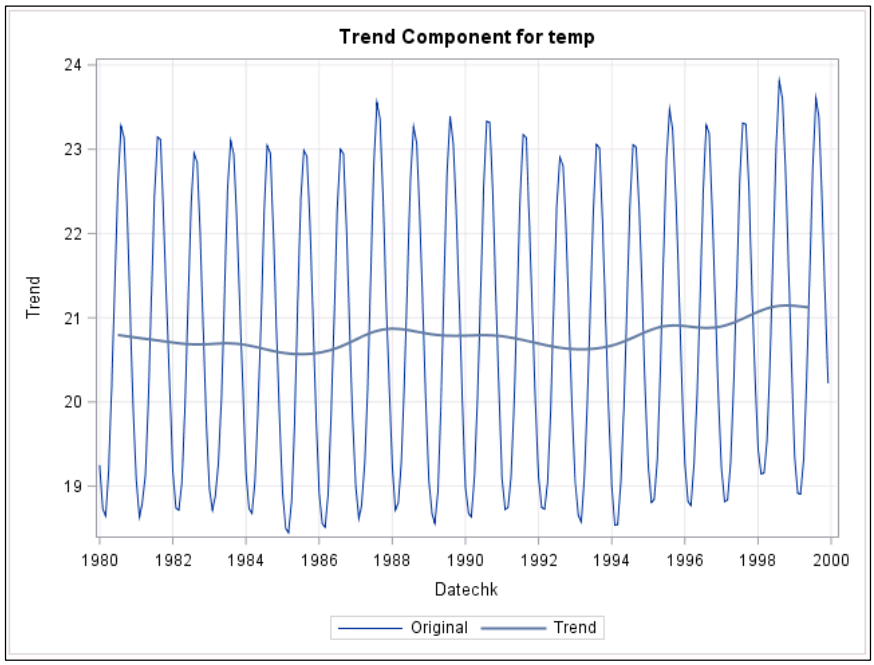


Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

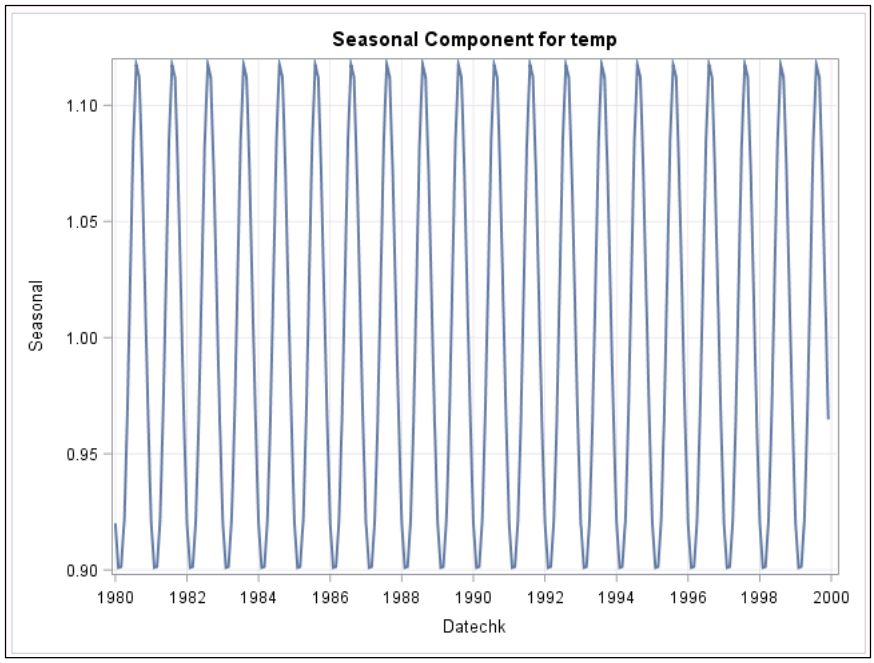
The seasonal cycles image displays a line for each year of the data. From this output, you can determine that each year followed a similar seasonal cycle. If there were lines that did not follow the pattern of the others, this could represent years when potentially some departure from the typical seasonal pattern existed. In this case, you see that the SST reached its lowest near March and its highest near August.



When the white noise probabilities are very small (p-values < 0.0001), it might be more advantageous to look at them using the log scale image. In either case, you can deduce that the series is not composed of only white noise. There is some attribute that can be modeled. What are these attributes? What components can you use in the modeling for this series?



Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.



Output from the decomposition analysis assists in the determination of the presence of a trend component, seasonal component, or both. From the trend component plot, you see that the trend line is rather flat. This suggests the lack of a trend in this series. From the seasonal component plot, you see a definitive seasonality that ranges from 10% below the mean at its lowest to slightly more than 10% above the mean at its highest. From these plots, you can narrow the focus to an exponential smoothing model that would incorporate seasonality but not trend.

In this case, from what you saw in the series plot earlier, this is either an additive seasonal ESM or the additive Winters model.

With the exploration complete, you can proceed to the Modeling and Forecasting task.

11. Exit the Maximized View. This causes the left side navigation panel to reappear.
12. Double-click the **Modeling and Forecasting** task to open it. Click the **Maximized View** button.
13. On the DATA tab, select the **SST** data set as the data set of interest. Click the plus sign (+) next to **Dependent variables**. Select **temp**. Click **OK**. Click and expand **Additional Roles**.
14. Click the plus sign (+) next to **Time ID**. Select **Datechk**. Click **OK**. The Interval area in the properties is automatically updated.
15. Click the **MODEL** tab. Next to Forecasting model type, select **Exponential smoothing** in the drop-down box. Under Model Settings, select **Additive seasonal exponential smoothing** for the Forecasting model. Do not make changes to the transformation or to the forecast type.
16. Click and expand **Plots**. Leave Select plots to display as Default plots.
17. Click the **OPTIONS** tab. Select **12** as the number of periods to forecast. (That is one year in this case.)
18. Click the **OUTPUT** tab. Select the **Create fit statistics data set** check box. Leave the name of this data set as **outstat**.

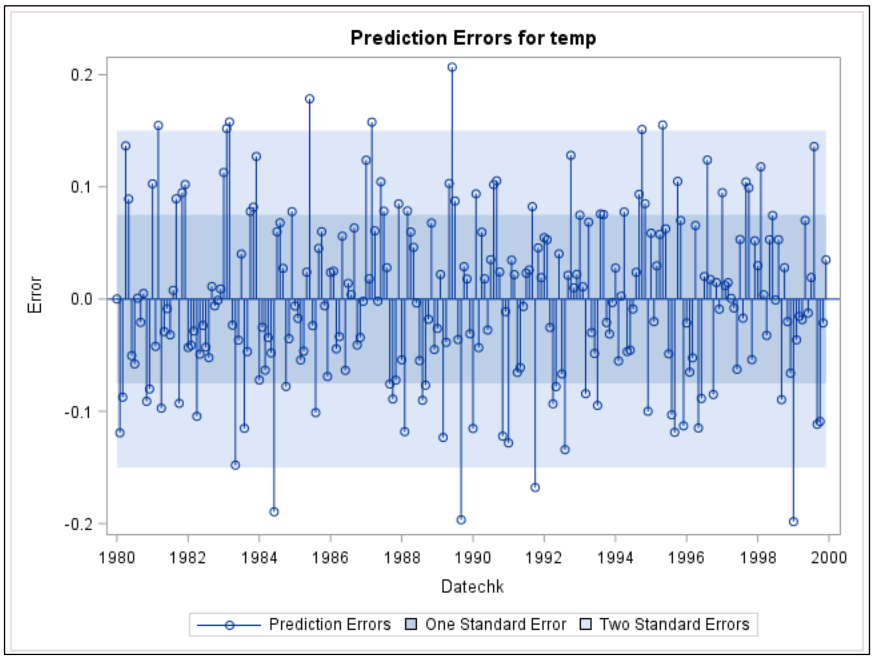
Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

Alternatively, you can write the SAS code directly as follows:

```
*Additive Seasonal Model;
proc esm data=STSM.SST
         back=0 lead=12
         seasonality=12
         plot=(corr errors modelforecasts)
         outstat=WORK.outstat;
   id Datechk interval=month;
   forecast temp / alpha=0.05 model=addseasonal;
run;
```

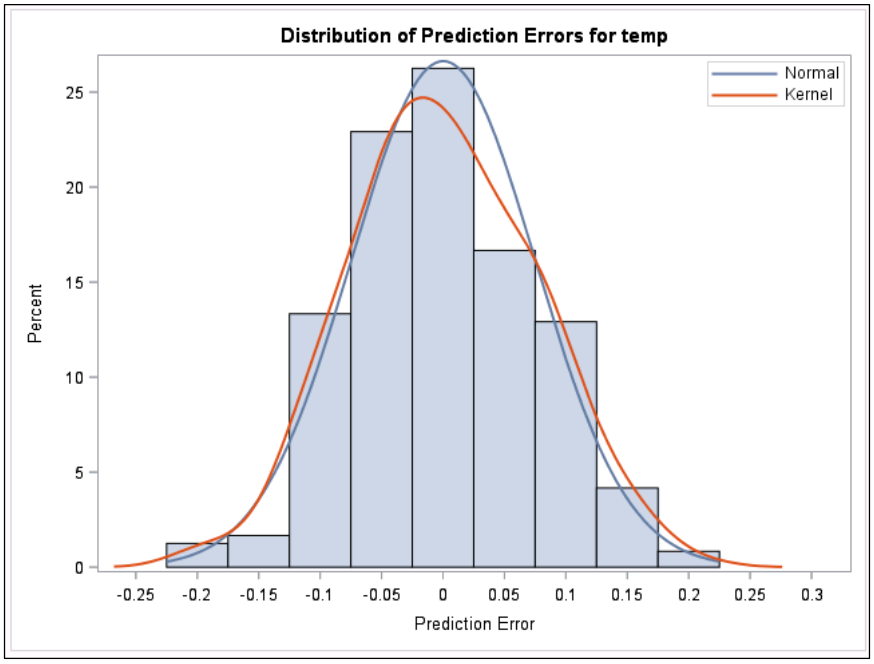
19. Click the running person icon to run this task.

Selected Output

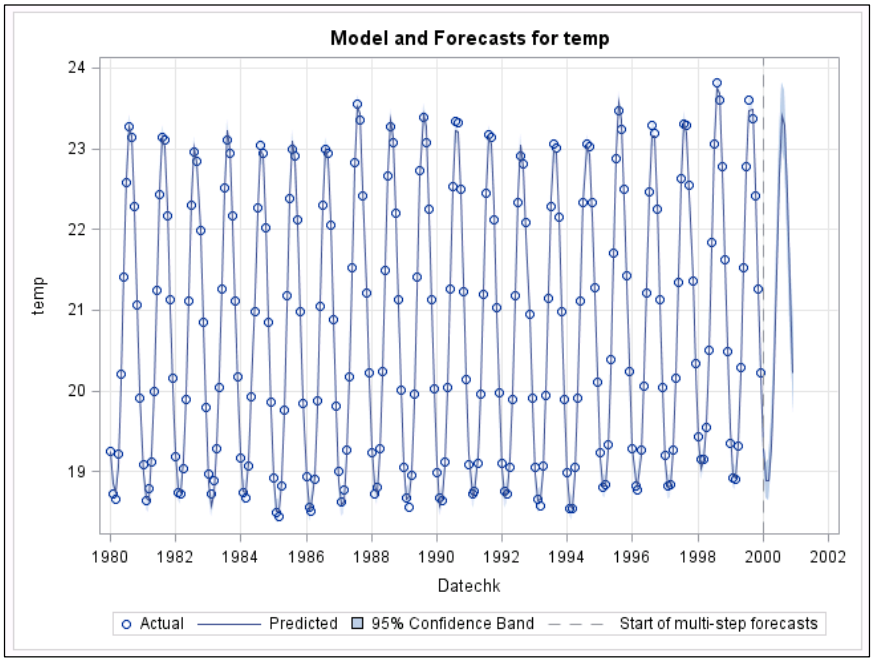


Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

From the prediction errors image, you can see that the prediction error for the **temp** series generally falls within one or two standard errors from zero. A few points escape this region but not enough to cause concern.



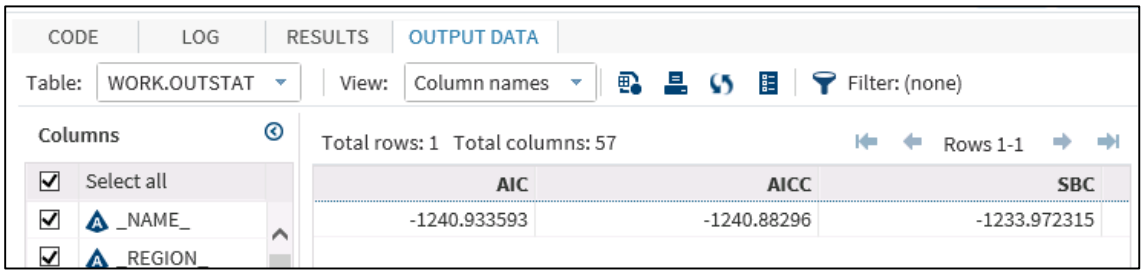
The distribution of prediction errors for the **temp** series is shown in a histogram with a superimposed normal and kernel curve. You can see that the errors do appear to be approximately normal.



You are also provided with a model and forecasts series plot for **temp**. You can see that the additive seasonal ESM does a good job of picking up the seasonality of the time series and carries this into the future forecasts.

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

To check the fit statistics of this additive seasonal model, check the **outstat** data set that you asked SAS to generate. Do this by clicking the **OUTPUT DATA** tab next to the RESULTS tab.



Scrolling over, you see many statistics that were generated from the model. You can choose to look at any of them but focus on the AIC (-1240.934) and SBC (-1233.972). There is another additive model that could be chosen for the time series. This is the additive Winters model. Run an additive Winters model and compare the AIC and SBC to this model.

Alternatively, you can run PROC PRINT.

```
proc print data=WORK.outstat;
    var AIC SBC;
run;
```

Obs	AIC	SBC
1	-1240.93	-1233.97

20. Click the **Model** tab and look under **Model Settings**. Change the **Forecasting model** drop-down selection to the **Winters additive** method.

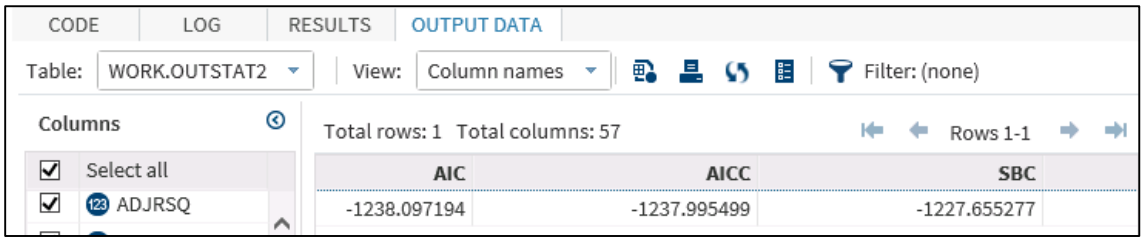
21. Under the OUTPUT tab, change the name of the fit statistics data set to **outstat2**. This enables you to retain both output data sets.

Alternatively, you can write the SAS code directly as:

```
*Additive Winters model;
proc esm data=STSM.SST
         back=0 lead=12
         seasonality=12
         plot=(corr errors modelforecasts)
         outstat=WORK.outstat2;
   id Datechk interval=month;
   forecast temp / alpha=0.05 model=addwinters;
run;
```

22. Rerun the model by clicking the running person icon.

23. Similar to the additive seasonal model, you can look at the provided images. However, click the **OUTPUT DATA** tab and look at the **outstat2** data set.



Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

If you compare the AIC and SBC from this model, -1238.097 and -1227.655 respectively, to the additive seasonal model, the additive seasonal model has slightly smaller values and smaller is better.

Alternatively, you can run PROC PRINT.

```
proc print data=WORK.outstat;
   var AIC SBC;
run;
```

Obs	AIC	SBC
1	-1238.10	-1227.66

In both the additive seasonal model and the additive Winters model, the white noise image shows that there might still be something beyond white noise that could be incorporated into the model. This might suggest the inclusion of a predictor variable. For this, you leave the ESM methodology and incorporate another method.

End of Demonstration

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

Exercises

1. Determining a Trend or Seasonal Component

The data set **STSM.ConcertSales** contains weekly sales data for a new ticket provider company. The data is weekly starting in January 2005 and collected until December 2014. It is organized by the variable **Date**. The variable **sales** is already a time series with the correct accumulation for the collected time periods.

Explore this to determine the presence of a trend or seasonal component. Based on your interpretation, run the one model of the seven ESM models that you think would be useful for this 12-week forecast.

a. Are there attributes that can be modeled within this time series?

b. Is there a trend or seasonal component (or both)?

c. After you ran your selected ESM models, what were the AIC and SBC of your selections?

End of Exercises

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

3.01 Multiple Answer Poll

Which of the following plots can assist in determining that there are components capable of being modeled for the time series?

a. white noise probability
b. trend component
c. seasonal component

27

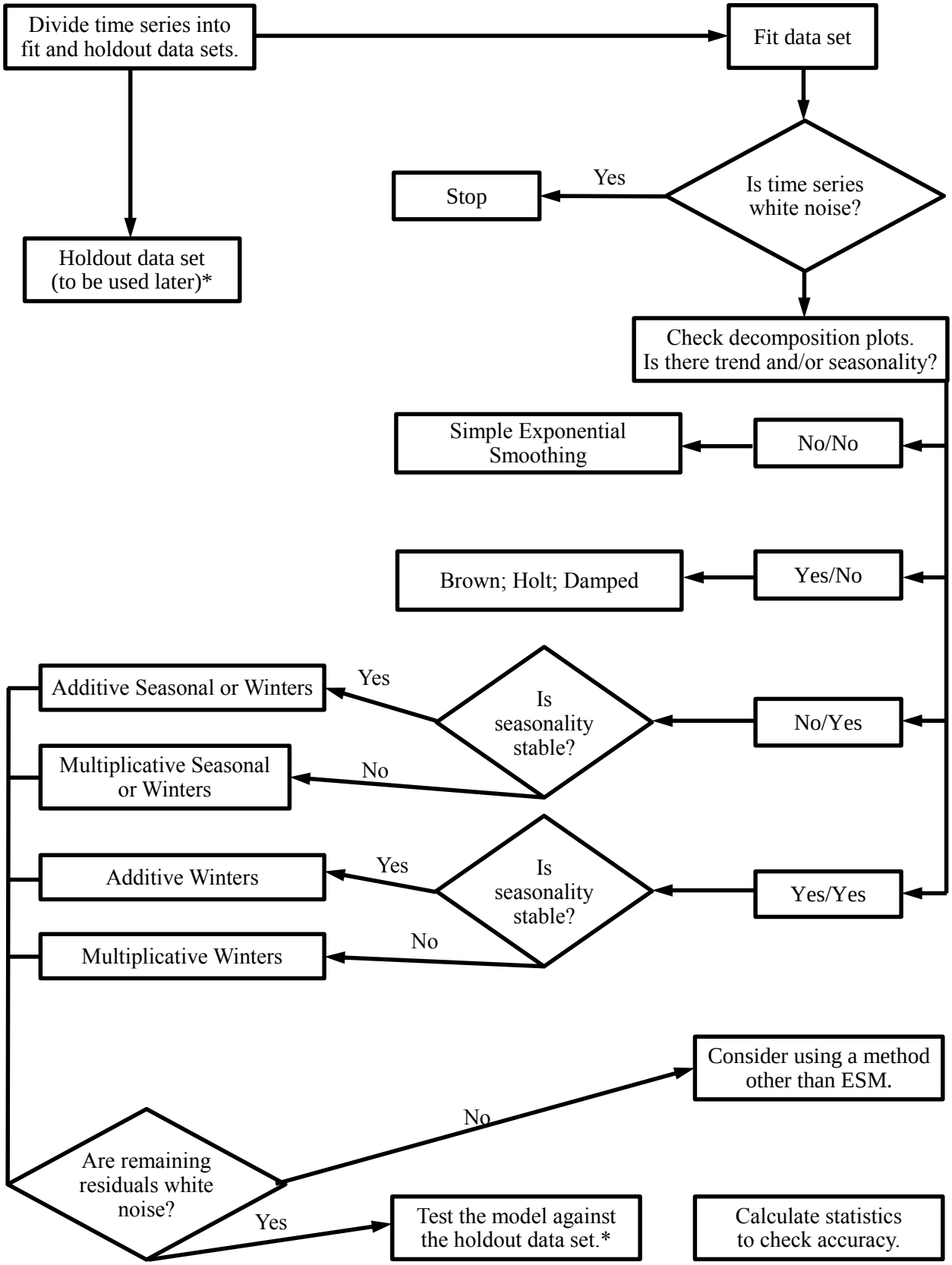
Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

3.2 Chapter Summary

ESMs formulate forecasts using a *smoothing* method of weighted averages. In the construction of the forecasts, more recent observations are given more weight than observations in the more distant past. The *exponential* is derived from the fact that weights not only diminish over time but they do so exponentially (Fomby 2008). ESMs have the added bonus that only a few parameters are required in the forecasting model and these equations are simple to implement.

The flowchart can serve as a guide for performing exponential smoothing analysis.

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.



Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

3.3 Solutions

Solutions to Exercises

1. Determining a Trend or Seasonal Component (SAS Studio Exercise)

1) Expand the **Tasks** area in the left navigation panel as well as the **Forecasting** subsection. Double-click the **Time Series Exploration** task. Click the **Maximized View** button.
2) On the Data tab, select the **ConcertSales** data set within the **STSM** library.
3) Click the plus sign (+) next to Dependent variable. Select **sales**. Click **OK**. For this model, there is no independent variable.
4) Click and expand the **Additional Roles** section. Click the plus sign (+) next to Time ID. Select **Date**. Click **OK**. Notice that the Properties section is updated to reflect the interval of weekly data.
5) Click the **Analyses** tab. Select the **Seasonal cycles** check box. Do *not* clear the Time Series check box. Under Autocorrelation Analysis, make sure that the **Perform autocorrelation analysis** check box is active and change the drop-down box beside Select plots to display to read selected plots.
6) Under Plots, select the **Autocorrelation analysis panel** and **White noise probability test (log scale)** check boxes.
7) Under Decomposition Analysis, make sure that the **Perform decomposition analysis** check box is selected and change the drop-down box beside Select plots to display to read selected plots.
8) Select all four check boxes (**Decomposition panel**, **Components**, **Seasonally adjusted series**, and **Seasonally adjusted series (percent change)**) under Plots.
9) When **Components** is selected, select **Trend component** and **Seasonal component** only.

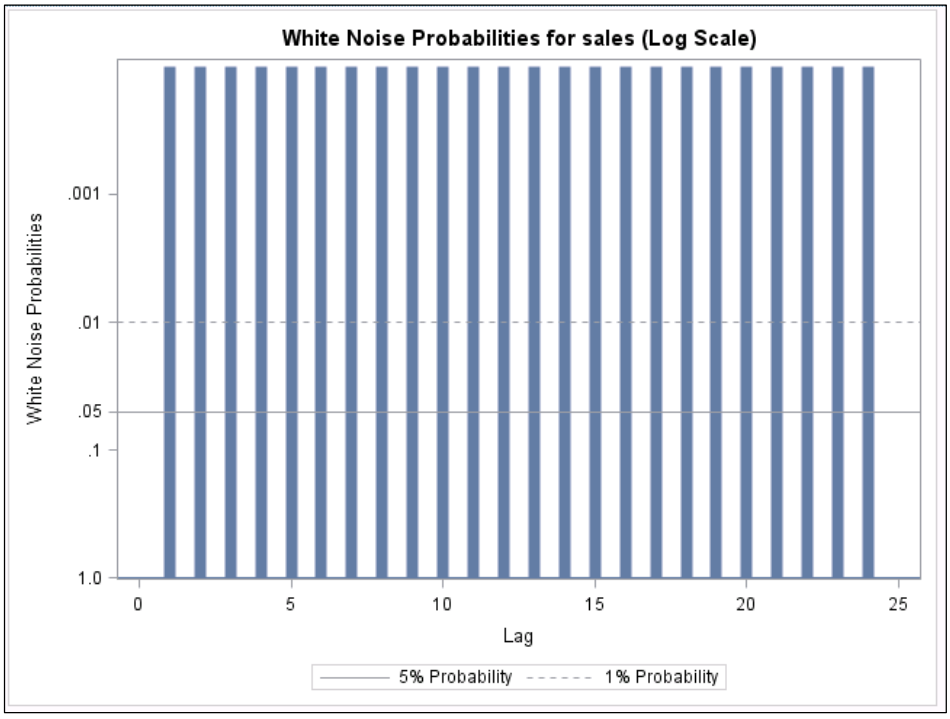
Alternatively, you can write the SAS code directly as follows:

```
/* STSM03s01.sas */
proc timeseries data=STSM.ConcertSales
                seasonality=52
                plots=(series cycles corr wn
                       decomp sa pcsa tc sc);
   id DATE interval=week.7;
   var sales;
   decomp sa pcsa tc sc / mode=multoradd;
   ods exclude WhiteNoiseProbabilityPlot;
run;
```

10) Click the running person icon to execute the task.

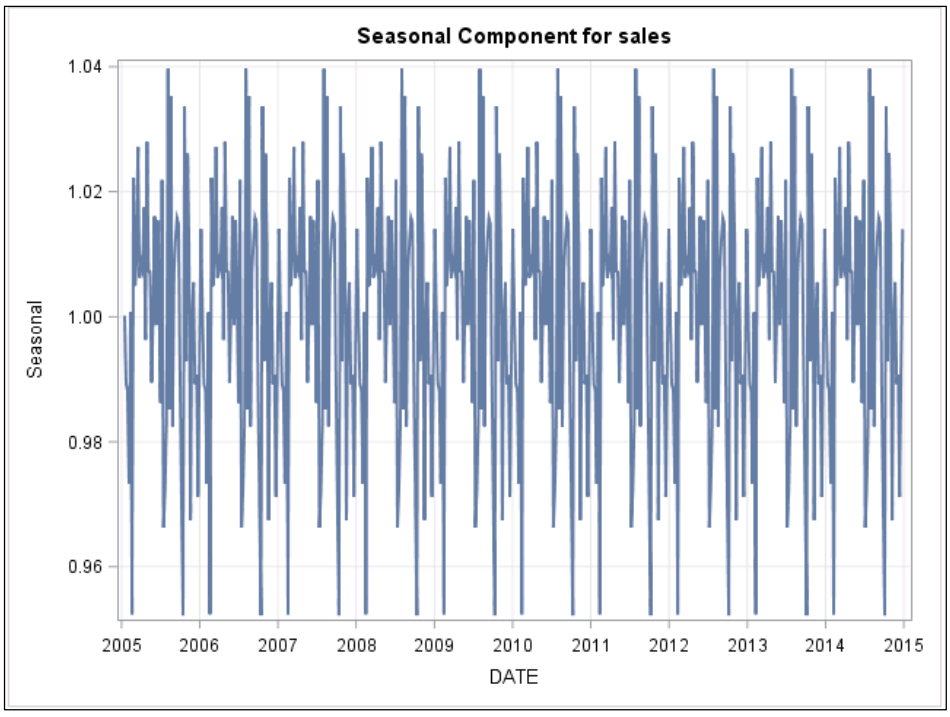
Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

a. Are there attributes that can be modeled within this time series?



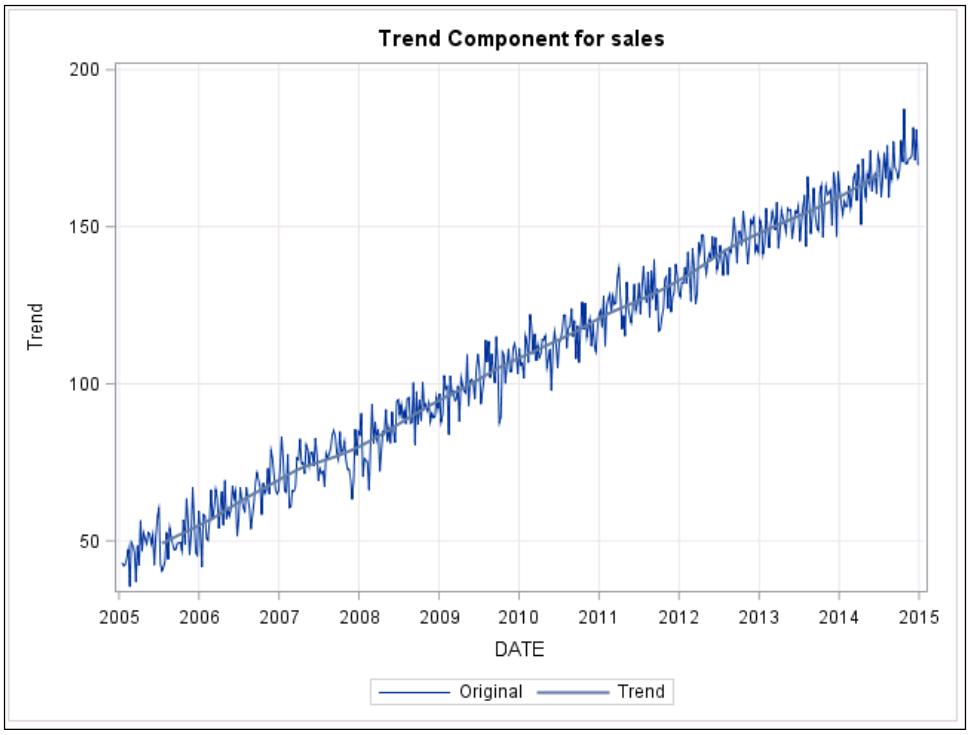
According to the White Noise Probabilities for sales plot, there do seem to be attributes that can be modeled within this time series.

b. Is there a trend or seasonal component (or both)?



The Seasonal Component for sales plot does not seem to suggest a seasonal component that should be modeled in the time series.

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.



On the other hand, the Trend Component for sales plot definitely shows a trend component that should be modeled in the time series.

c. After you ran your selected ESM models, what were the AIC and SBC of your selections?

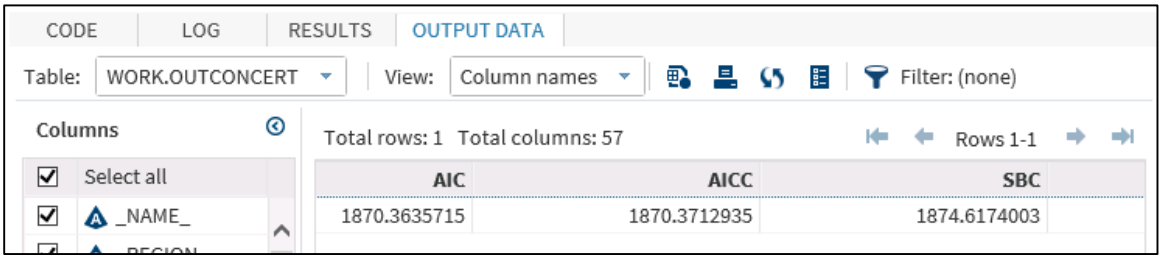
1) Exit the Maximized View. This causes the left side of the navigation panel to reappear.
2) Double-click the **Modeling and Forecasting** task to open it. Click the **Maximized View** button.
3) On the DATA tab, select the **ConcertSales** data set as the data set of interest. Click the plus sign (+) next to Dependent variables. Select **sales**. Click **OK**. Click and expand **Additional Roles**.
4) Click the plus sign (+) next to Time ID. Select **Date**. Click **OK**. Again, the Interval area in properties is automatically updated.
5) Click the **MODEL** tab. Next to Forecasting model type, select **Exponential smoothing** in the drop-down box. Under Model Settings, select **Double (Brown) exponential smoothing** for the forecasting model. Do not make changes to Transformation or to Forecast type.
6) Click and expand **Plots**. Under Select plots, keep the Default plots setting.
7) Click the **OPTIONS** tab. Select **12** as the number of periods to forecast. (That is three months in this case.)
8) Click the **OUTPUT** tab. Select the **Create fit statistics data set** check box. The name of this data set is **outconcert**.

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

Alternatively, you can write the SAS code directly as follows:

```
*Double (Brown) Model;
proc esm data=STSM.ConcertSales
         back=0 lead=12
         plot=(corr errors modelforecasts)
         outstat=WORK.outconcert;
   id DATE interval=week.7;
   forecast sales / alpha=0.05 model=double;
run;

proc print data=WORK.outconcert;
    var AIC SBC;
run;
```

9) Click the running person icon to run this task.



Obs	AIC	SBC
1	1870.36	1874.62

Using Double (Brown) exponential smoothing, the AIC is 1870.364 and the SBC is 1874.617.

10) Click the **MODEL** tab and look under **Model Settings**. Change the Forecasting model drop-down list to be **Linear (Holt) exponential smoothing method**.

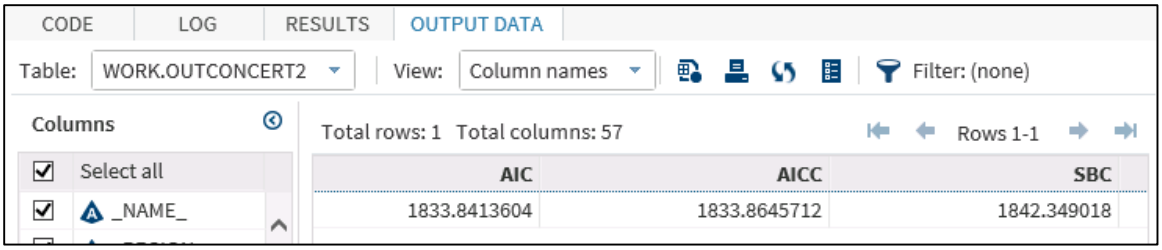
11) On the OUTPUT tab, change the name of the **fit statistics** data set to **outconcert2**. This enables you to retain both output data sets.

Alternatively, you can write the SAS code directly as:

```
*Linear (Holt) Model;
proc esm data=STSM.ConcertSales
         back=0 lead=12
         plot=(corr errors modelforecasts)
         outstat=WORK.outconcert2;
   /* id statement */
   id DATE interval=week.7;
   forecast sales / alpha=0.05 model=linear;
run;

proc print data=WORK.outconcert2;
   var AIC SBC;
run;
```

12) Rerun the model by clicking the running person icon.

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.



Obs	AIC	SBC
1	1833.84	1842.35

Using Linear (Holt) exponential smoothing, the AIC is 1833.841 and the SBC is 1842.349.

If you compare these two model approaches, it appears that the Holt model is better.

End of Solutions

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

Solutions to Student Activities (Polls/Quizzes)

3.01 Multiple Answer Poll – Correct Answers

Which of the following plots can assist in determining that there are components capable of being modeled for the time series?

a. white noise probability
b. trend component
c. seasonal component

28

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

Chapter 4 Unobserved Components Models

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

4.1 Introduction to Using Unobserved Components Models

Unobserved components models (UCMs) are substantially different from other models that are discussed in this course. They can accommodate and extrapolate more general features of the data (for example, seasonal patterns that change as a function of time). However, they are relatively easy to specify and refine. This chapter focuses on their ease of use and builds some intuition about the UCM framework.

Time Series Analysis: UCM Goals

Tasks:

- Forecast future Y values. (Interpolate past missing Ys.)
- Determine the nature of the relationship between Y and X1, X2, and so on.
- Decompose Y into some interpretable sub-components such as trend, cycles, seasons, and regression effects. Extrapolate these subcomponents into the future.

3

Time Series Analysis: UCM Goals

Answer questions such as the following:

- Did the behavior of Y qualitatively change at some past time *t*? Do some of the observed Y values look odd?
- Is the seasonal pattern changing over time?
- Did the nature of the relationship between Y and X1 remain stable through the life of the series?
- Is Y increasing or decreasing at a steady rate? If so, at what rate?

4

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

Unobserved Components Models

Response Time Series = Superposition of components such as Trend, Seasons, Cycles, and Regression effects

- Each component in the model captures some important feature of the series dynamics.
- Components in the model have their own probabilistic models.
- The probabilistic component models include meaningful deterministic patterns as special cases.

5

4.01 Multiple Choice Poll

Which time series analysis tool do you use most often?

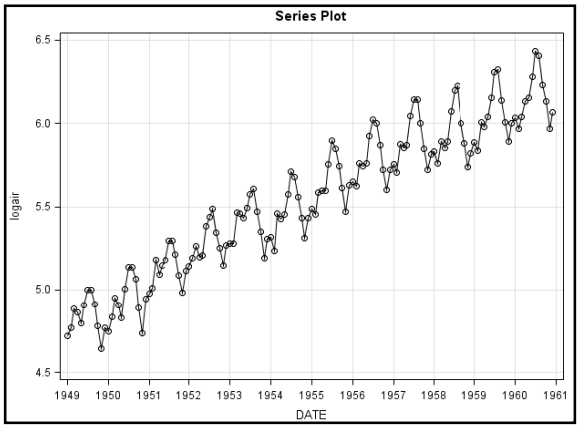
a. ARIMA modeling
b. exponential smoothing
c. X12 Census Bureau seasonal decomposition
d. spectral analysis
e. state space modeling
f. nonlinear time series modeling
g. other

6

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

Airline Data Example: *Hello, World!* of the Time Series Modeling

- Log-transformed monthly airline passenger series
- Source: Series G in Box and Jenkins (1976)



7

UCM Model for the Airline Series

Basic UCM Model:

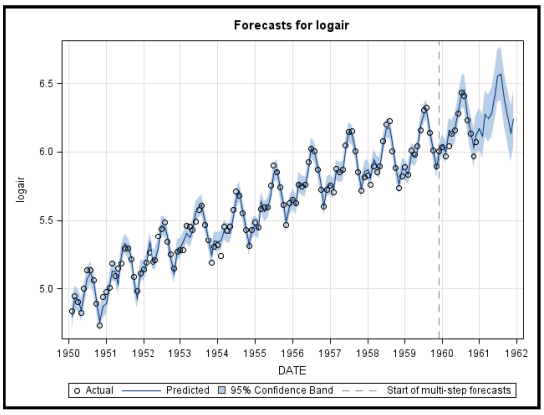
Logair ~ trend + season + noise

```
proc ucm data=airline;
   model logair;
   irregular;
   level;
   slope var=0 noest;
   season length=12 type=trig;
   estimate back=12;
   forecast back=12 lead=24;
run;
```

8

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

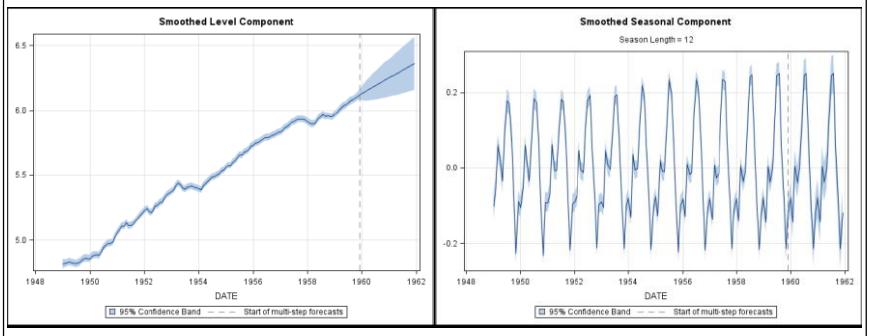
Airline Data: UCM Forecasts



UCM

9

Airline Data: Component Estimates

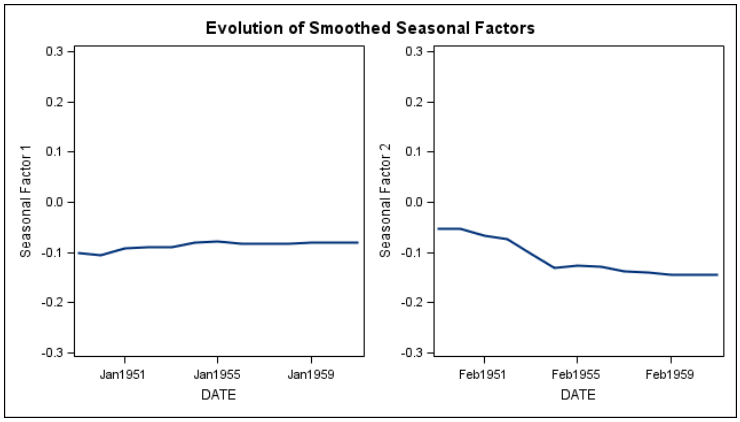


Estimated slope of the trend curve ~ 0.01/month
(in the log scale)

10

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

Annual Variation in the Seasonal Effects



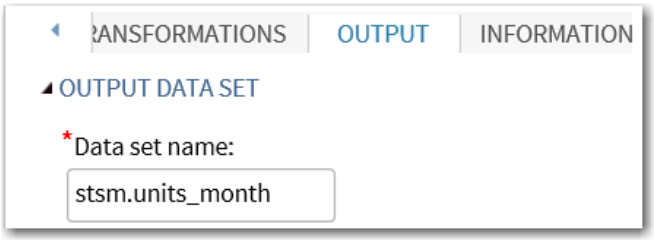
11

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

Creating the Unit Series on a Monthly Interval Using an Average Accumulation Method

Recall that in Chapter 1, you explored a monthly, intervaled time series that showed evidence of trend and seasonal component variation. This series was created from time-stamped observations on **units** in the **CH1_DEMODAT** table. The first part of this demonstration creates a new table, **stsm.units_month**, that aggregates the **units** time series to a monthly interval using the average aggregation method. A UCM model is then specified and fit to **units**.

1. Create a new Time Series Data Preparation task in SAS Studio.
2. On the DATA tab, set the data option to **STSM.CH1_DEMODAT**. Set the time series variable to **units**. Set the time ID to **date**, and set the interval of the time ID variable to **Month**.
3. Click the **TRANSFORMATIONS** tab, and set the Accumulation method to **Average**.
4. Click the **OUTPUT** tab, and change the output data set name to **units_month** in the **STSM** library.



Alternatively, write the SAS code directly as follows:

```
/* STSM04d01a.sas */
proc timeseries data=stsm.CH1_DEMODAT seasonality=12
                out=stsm.UNITS_MONTH;
   id date interval=month;
   var units / accumulate=average transform=none dif=0 sdif=0;
run;
```

5. Run the task.

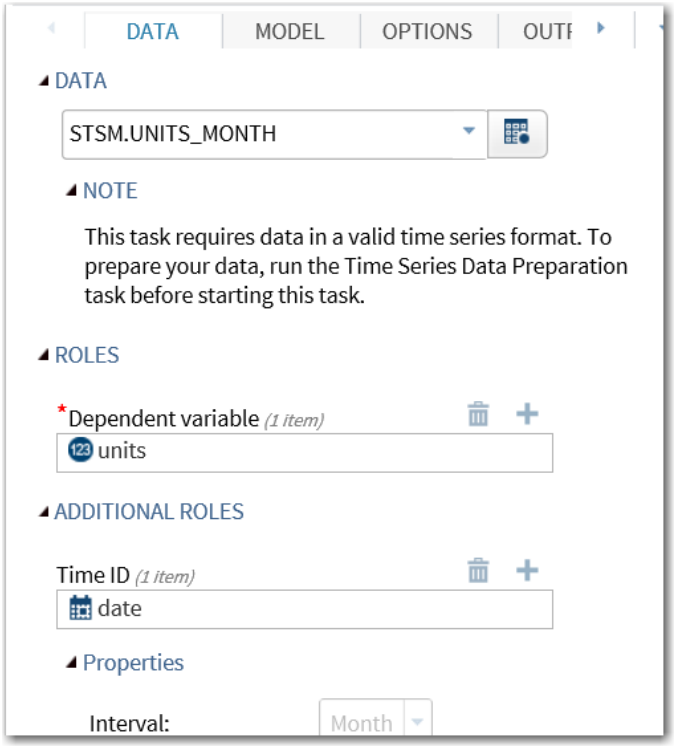
End of Demonstration

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

Specifying an Unobserved Components Model

Build a UCM and explore the results.

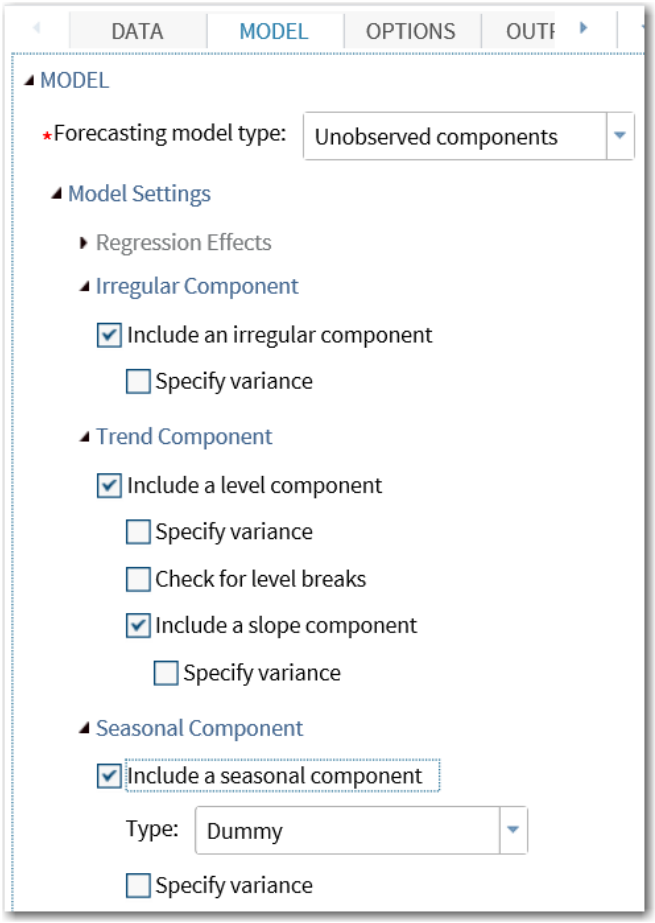
1. Expand the **Forecasting** tasks, and create a new Modeling and Forecasting task.
2. On the DATA tab, assign table and variable roles as shown below.



3. On the MODEL tab, select **Unobserved Components** as the forecasting model type.
4. Recall that the component analysis from Chapter 1 suggested that the monthly, intervaled **units** data has trend, seasonal, and irregular (ARMA type) patterns. To accommodate these, modify the default model settings by selecting the **Include a slope component** check box. Also, expand the **Seasonal Component** dialog box, and select the **Include a seasonal component** check box. These settings are summarized below.

   The level and slope components combine to accommodate a trend in the model. More information about UCM components is provided later in this chapter.

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.



5. Expand the **Plots** dialog box at the bottom of the MODEL tab, and select **Selected Plots** in the Selected Plots to Display box.

6. In addition to the default plots, select the following: **One Step Ahead Forecasts**, as well as smoothed **Irregular Component**, **Season Component**, **Level Component**, and **Slope Component**.

Alternatively, you can write the SAS code directly as follows:

```
/* STSM04d01b.sas */
/* Build and Explore a UCM */
proc ucm data=stsm.UNITS_MONTH;
   id date interval=month;
   model units;
   irregular plot=smooth;
   level plot=(smooth);
   slope plot=(smooth);
   season length=12 type=dummy plot=(smooth);
   estimate plot=(panel model loess);
   forecast lead=12 back=0 alpha=0.05 plot=(forecasts);
   outlier;
run;
```

Recall that the component analysis from Chapter 1 suggested that the monthly intervaled **units** data has trend, seasonal, and irregular (ARMA type) patterns. The UCM procedure syntax below accommodates these components.

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

The level and slope components combine to accommodate a trend in the model. More information about UCM components is provided later in this chapter.

Additional plot options were added to the component statements:

- The PLOT=(SMOOTH) options produce smoothed representations of the level, slope, and season components.
- The PLOT=(PANEL, MODEL, and LOESS) options in the ESTIMATE statement produce residuals diagnostics plots.
- The PLOT=(FORECASTS) option in the FORECAST statement produces a plot of historical and lead forecasted values.

7. Select **Run** to submit the generated UCM syntax.

   The Significance Analysis of Components table indicates that the data contain a significant level, slope, and seasonal component. However, the irregular component seems to be negligible in the presence of other components in the model. The slope estimate suggests that the data is increasing by approximately 14 units per month.

Significance Analysis of Components (Based on the Final State)

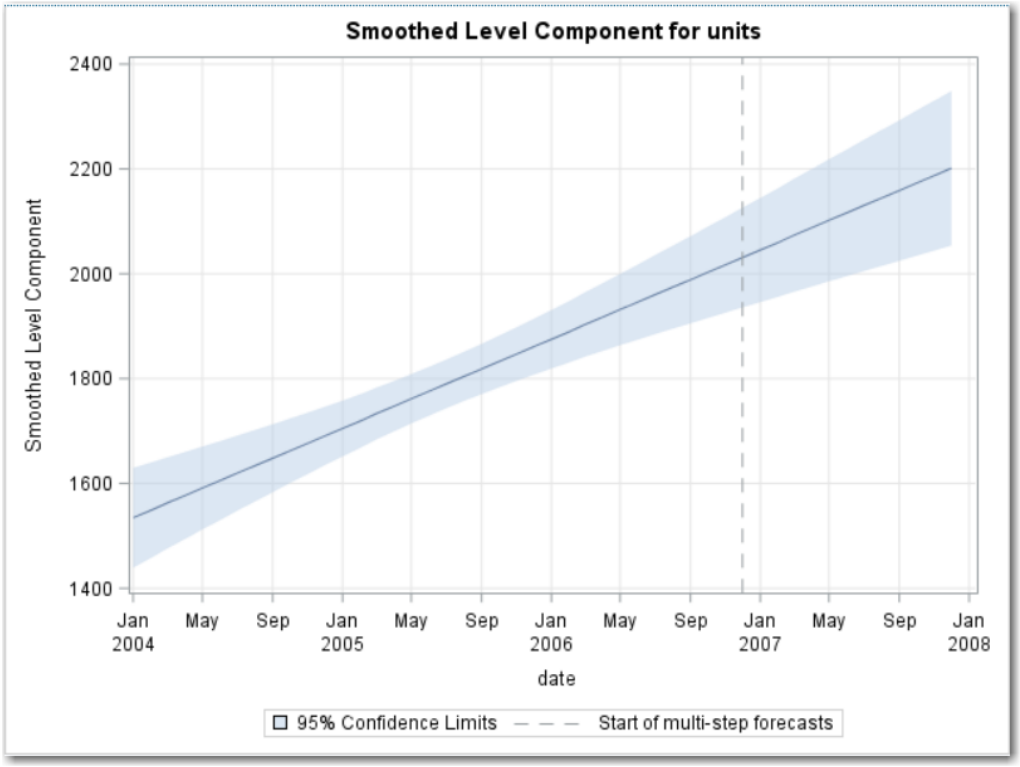
Component	DF	Chi-Square	Pr > ChiSq
Irregular	1	0.00	0.9967
Level	1	1748.27	<.0001
Slope	1	34.28	<.0001
Season	11	26.55	0.0054

Trend Information (Based on the Final State)

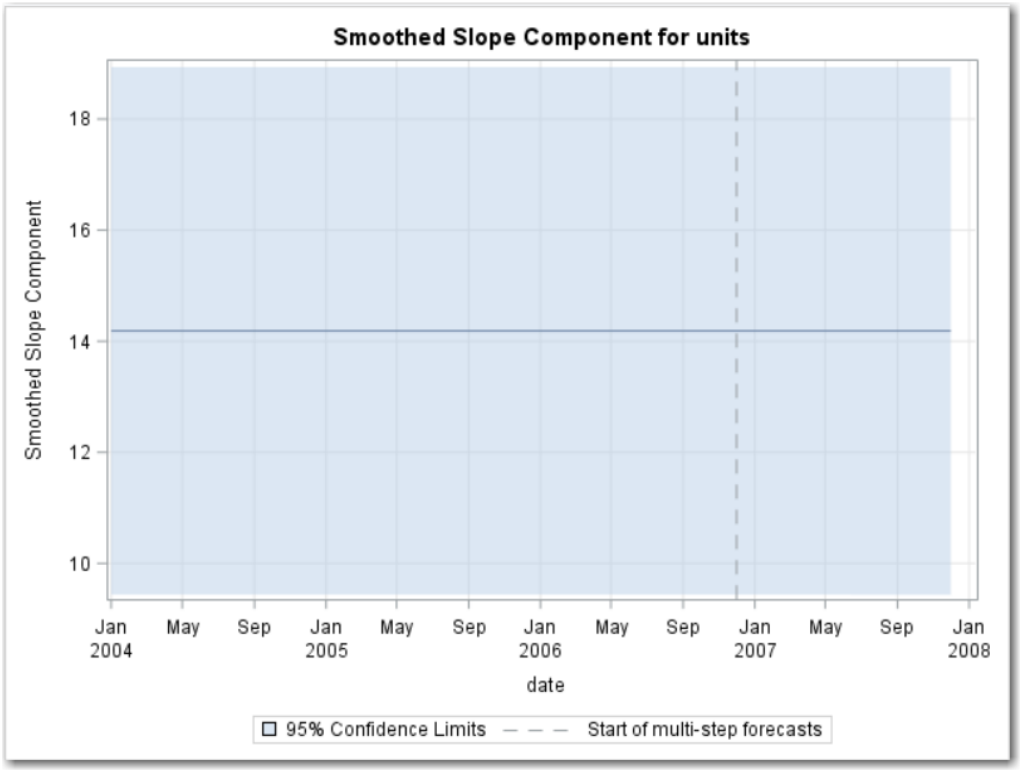
Name	Estimate	Standard Error
Level	2031.166251	48.578223
Slope	14.18644584	2.4228224

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

The Smoothed Level Component plot shows how the level of the data is estimated to evolve over the history and future forecast horizon.

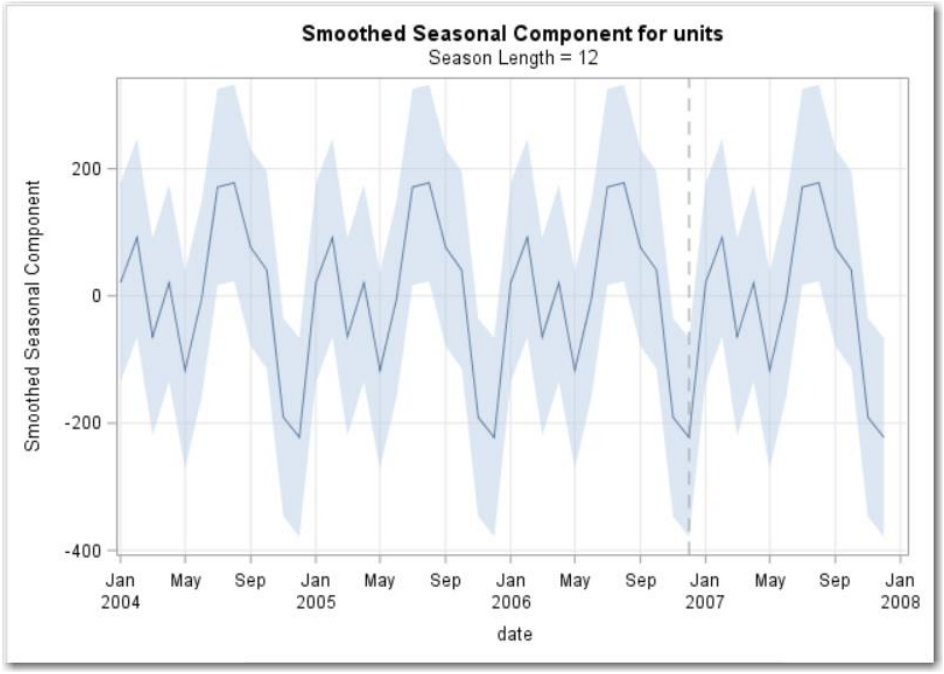


The Smoothed Slope Component plot indicates that the slope in the data is a constant, and that the trend component is basically a deterministic linear trend.



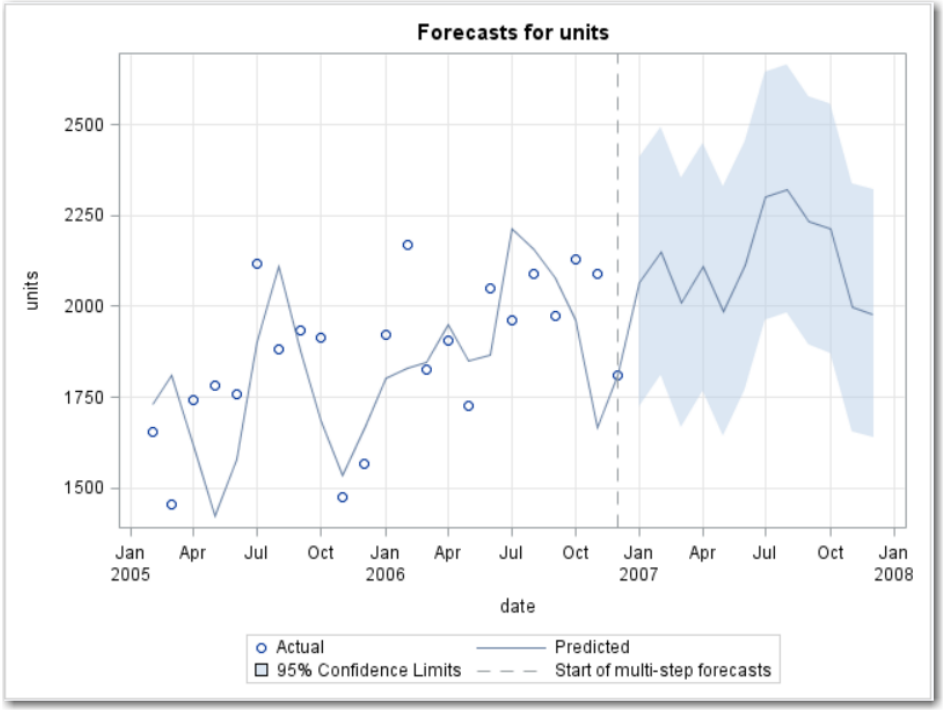
Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

The Smoothed Seasonal Component plot shows in the sample and lead forecasts from the seasonal component model.



Smoothed component plots are based on parameter estimates derived from all of the observations in the data.

The forecast plot shows the extrapolated trend, level, and seasonal components in the lead forecast. The forecast is the sum of forecasts from the estimated components listed in the table above.



End of Demonstration

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

4.2 Unobserved Components Models

The Nature of Components

The examples shown previously share some commonly observed time series data qualities.

- If you call trend the long-term, slowly varying pattern of the series, it rarely has a definite shape, such as linear or quadratic or some other simple parametric curve.
- The periodic patterns exhibited by the series also rarely preserve their properties over the life of the series.
- Therefore, if the observed series is to be modeled as a sum of components, then these patterns must be flexible and adaptive.

15

Trend Component Example

Two models for trend:

- The *random walk trend* (RW) represents a slowly varying level without a drift in any particular direction.
- The *local linear trend* (LL) represents a locally linear pattern with slowly varying intercept and slope.

16

Additional trend specifications, such as trend specified using differencing, can also be considered.

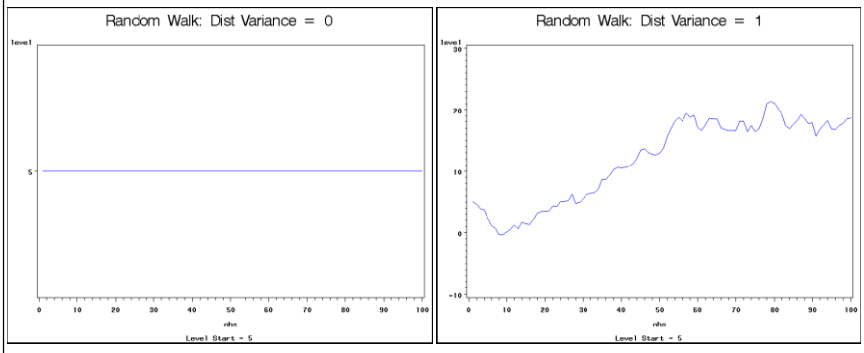
Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

Random Walk Trend

$$\mu_t = \mu_{t-1} + \eta_t \qquad \eta_t \sim N(0, \sigma^2_\mu)$$

17

Random Walk Simulation



18

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

Local Linear Trend

Deterministic linear trend:

$$\mu_t = \mu_0 + \beta_0 * t$$

Recursive form:

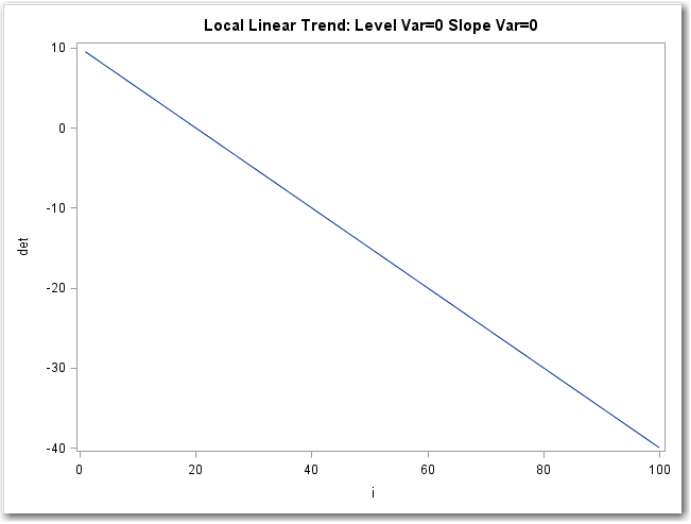
$$\mu_t = \mu_{t-1} + \beta_{t-1}$$
$$\beta_t = \beta_{t-1}$$

Local linear trend:

$$\mu_t = \mu_{t-1} + \beta_{t-1} + \eta_t \qquad \eta_t \sim N(0, \sigma^2_\mu)$$
$$\beta_t = \beta_{t-1} + \xi_t \qquad \xi_t \sim N(0, \sigma^2_\beta)$$

19

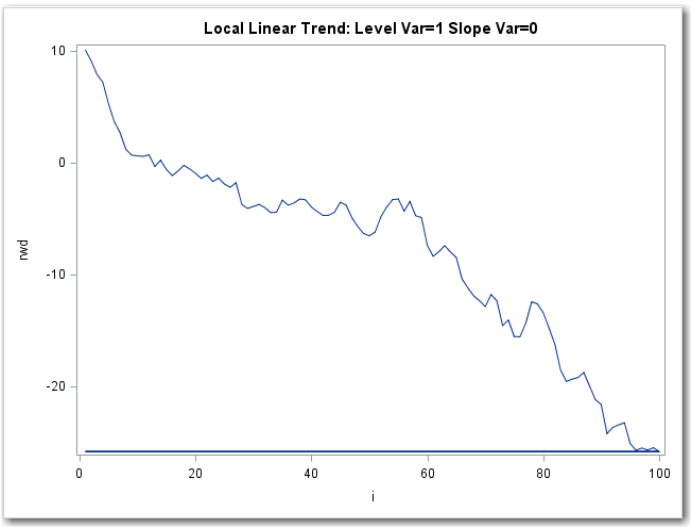
Local Linear Trend Simulation



20

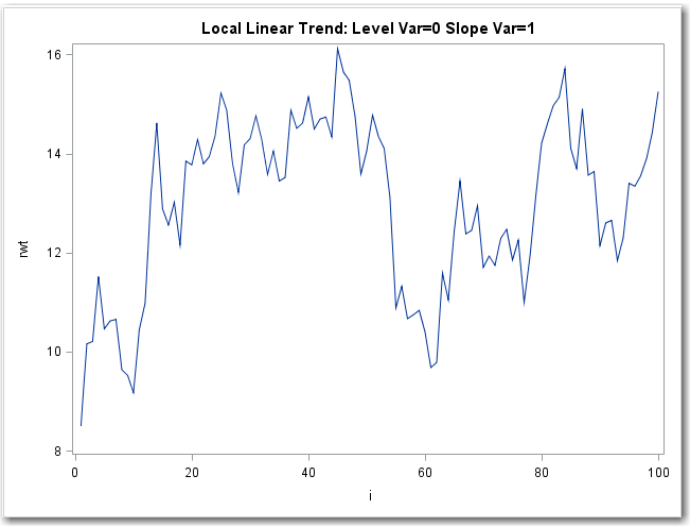
Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

Local Linear Trend Simulation



21

Local Linear Trend Simulation



22

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

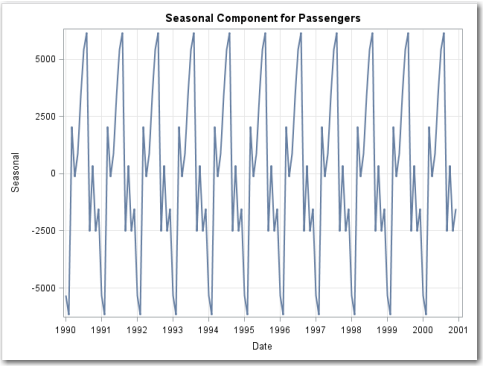
Season Component Example

1. The seasonal fluctuations are a common source of variation in the time series data.
2. The seasonal effects are regarded as corrections to the general trend of the series due to seasonal variations, and these effects sum to zero when summed over the full season cycle.
3. Therefore, a (deterministic) seasonal component γ_t is modeled as a periodic pattern of an integer period s so that the sum is as follows:

$$\sum_{i=0}^{s-1} \gamma_{t-i} = 0$$

23

Example of a (Deterministic) Seasonal Pattern (Period=12)



Seasonal Index	Seasonal Component
1	-5338.903472
2	-6177.449306
3	2037.0590278
4	-128.6951389
5	882.51736111
6	3367.1923611
7	5416.2756944
8	6168.7340278
9	-2525.282639
10	344.73819444
11	-2519.265972
12	-1526.920139

24

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

Stochastic Seasonal: Dummy Type

$$\sum_{i=0}^{s-1} \gamma_{t-i} = \omega_t, \qquad \omega_t \sim i.i.d.\, N\left(0, \sigma_{\varpi}^2\right)$$

1. The periodic pattern sums to zero **in the mean**.

2. The disturbance variance controls the variation in the seasons. If it is zero, the model reduces to a deterministic seasonal. This is equivalent to having (*s*-1) dummy regressors.

25

A General UCM

A general UCM can be described as follows:

$$y_t = \mu_t + \gamma_t + \psi_t + r_t + \sum_{i=1}^{p} \phi_i y_{t-i} + \sum_{j=1}^{m} \beta_j x_{jt} + \varepsilon_t$$

$$\varepsilon_t \sim i.i.d.\, N\left(0, \sigma_{\varepsilon}^2\right)$$

- $\varepsilon_t, \mu_t, \gamma_t, \psi_t$, and r_t represent different stochastic components.
- The model can contain multiple seasons and cycles.
- The term $\sum_{j=1}^{m} \beta_j x_{jt}$ represents the effects of predictors.
- The term $\sum_{i=1}^{p} \phi_i y_{t-i}$ is a regression term involving the lags of the dependent variable.

26

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

Model Specification Syntax

A UCM is specified by describing the components in the model. For example, consider the following model:

$$y_t = \mu_t + \gamma_t + \varepsilon_t$$

It consists of the LL trend μ_t, monthly trigonometric season γ_t, and an irregular component ε_t.
The corresponding syntax is as follows:

```
MODEL y;
      IRREGULAR;
      LEVEL;
      SLOPE;
      SEASON LENGTH=12 TYPE=TRIG;
```

27

continued...

A General Model Building Approach

A general modeling approach can be described as follows:

- Identify systematic components of variation in the data.
- Specify a general UCM that accommodates these components.
- Identify components that are non-stochastic. The variance of these components can be fixed at 0.
- Identify components that are not significant in explaining variation in the target. These components are candidates for removal from the model.

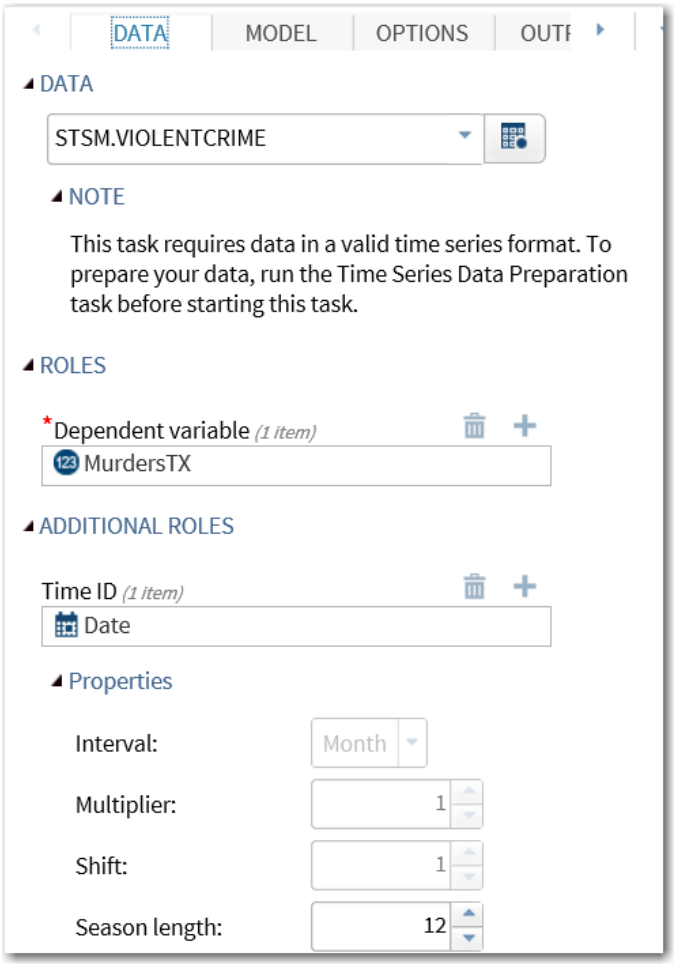
28

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

Refining an Unobserved Components Model

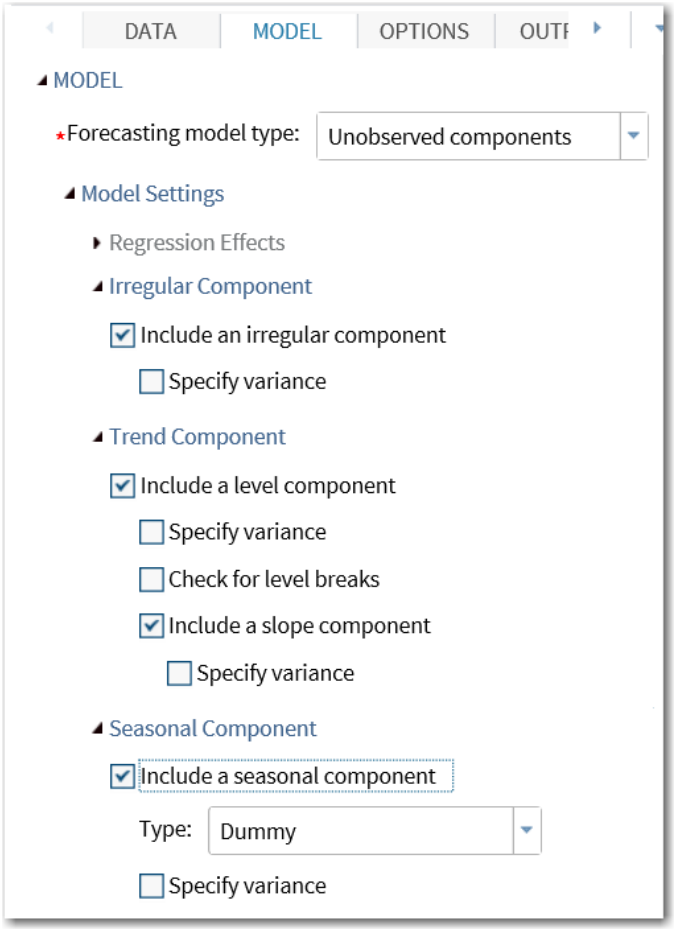
The time series **MurdersTX** in the **STSM.VIOLENTCRIME** data set was explored in an exercise in Chapter 1. A reasonable starting hypothesis is that the data contain trend (level + slope), seasonal, and irregular components.

1. Specify a baseline UCM that accommodates the hypothesized components.
   a. Create a new Modeling and Forecasting task, and assign table and variable roles as shown below.



   b. Click the **MODEL** tab, and select **Unobserved components** as the forecasting model type.
   c. Specify that the UCM should contain *irregular*, *level*, *slope*, and *seasonal* components.

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.



Alternatively, you can write the SAS code directly as follows:

```
/* STSM04d02.sas */
/* Specify the Baseline model */
proc ucm data=stsm.VIOLENTCRIME;
   id Date interval=month;
   model MurdersTX;
   irregular;
   level;
   slope;
   season length=12 type=dummy;
   forecast lead=12 back=0 alpha=0.05;
   outlier;
run;
```

d. Run the task to submit the generated code.

This model can be considered the baseline to judge model refinements. The initial fit statistics are shown below.

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

Likelihood Based Fit Statistics	
Statistic	**Value**
Full Log Likelihood	-431.5
Diffuse Part of Log Likelihood	-4.97
Non-Missing Observations Used	108
Estimated Parameters	4
Initialized Diffuse State Elements	13
Normalized Residual Sum of Squares	95
AIC (smaller is better)	870.96
BIC (smaller is better)	881.17
AICC (smaller is better)	871.4
HQIC (smaller is better)	875.08
CAIC (smaller is better)	885.17

The final estimates of the variances associated with each component indicate that only the Irregular component is stochastic.

Final Estimates of the Free Parameters					
Component	**Parameter**	**Estimate**	**Approx Std Error**	**t Value**	**Approx Pr > |t|**
Irregular	**Error Variance**	239.38162	48.55214	4.93	<.0001
Level	**Error Variance**	25.44934	18.72870	1.36	0.1742
Slope	**Error Variance**	0.08117	0.13315	0.61	0.5421
Season	**Error Variance**	1.87519	5.64980	0.33	0.7400

The Significance Analysis of Components table indicates that only the level and season components explain a substantial proportion of the variation in **MurdersTX**, in the presence of other components in the model.

Significance Analysis of Components (Based on the Final State)			
Component	**DF**	**Chi-Square**	**Pr > ChiSq**
Irregular	1	0.55	0.4590
Level	1	133.11	<.0001
Slope	1	0.99	0.3202
Season	11	56.83	<.0001

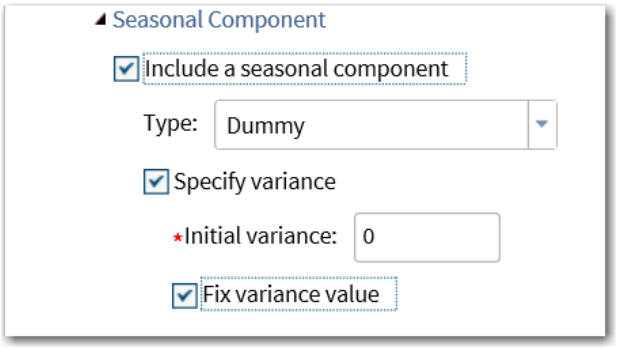
2. Refine the baseline UCM. Fix the seasonal variance at 0.

   Based on the estimates of component variance, the season component is the most deterministic. Model refinement begins by fixing the variance of the component at 0.

   a. Select the **Specify variance** and **Fix variance value** check boxes in the Seasonal Component options area.

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

b. Verify that 0 is in the **Initial variance** field.



Alternatively, you can write the SAS code directly as follows:

```
/* Fix the Season component variance at 0 */
proc ucm data=stsm.VIOLENTCRIME;
   id Date interval=month;
   model MurdersTX;
   irregular;
   level;
   slope;
   season length=12 type=dummy variance=0 noest;
   forecast lead=12 back=0 alpha=0.05;
   outlier;
run;
```

Model refinement begins by fixing the variance of the season component at 0 using the VARIANCE and NOEST options as shown.

c. Select **Run** to fit the re-specified model.

The Fit Statistics table indicates that the penalized, overall fit of the model is better than the baseline.

Likelihood Based Fit Statistics	
Statistic	**Value**
Full Log Likelihood	-431.5
Diffuse Part of Log Likelihood	-4.97
Non-Missing Observations Used	108
Estimated Parameters	3
Initialized Diffuse State Elements	13
Normalized Residual Sum of Squares	95
AIC (smaller is better)	869.09
BIC (smaller is better)	876.75
AICC (smaller is better)	869.35
HQIC (smaller is better)	872.18
CAIC (smaller is better)	879.75

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

The slope and level components are still deterministic in the re-specified model.

Final Estimates of the Free Parameters					
Component	Parameter	Estimate	Approx Std Error	t Value	Approx Pr > |t|
Irregular	Error Variance	247.85482	43.80894	5.66	<.0001
Level	Error Variance	24.50078	18.36962	1.33	0.1823
Slope	Error Variance	0.08448	0.13614	0.62	0.5349

The relative importance of the components in explaining variation in **MurdersTX** did not change from the baseline.

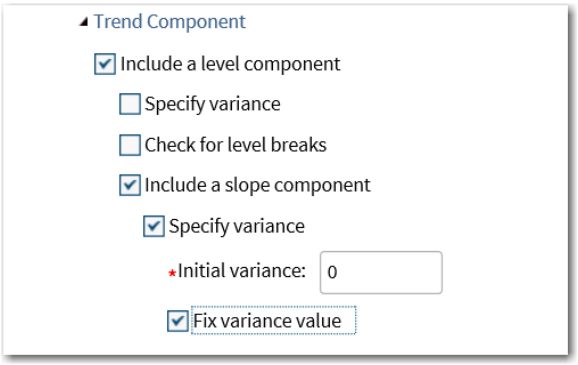
Significance Analysis of Components (Based on the Final State)			
Component	DF	Chi-Square	Pr > ChiSq
Irregular	1	0.62	0.4313
Level	1	132.54	<.0001
Slope	1	0.98	0.3227
Season	11	62.62	<.0001

3. Refine the UCM. Fix the slope variance at 0.

   Because the slope is indicated to be deterministic, the next step fixes the slope component variance at 0.

   a. In the Trend Component section, beneath the Include a slope component check box, select the **Specify variance** and **Fix variance value** check boxes in the Trend Component options area.

   b. Verify that 0 is in the **Initial variance** field.



Alternatively, you can write the code directly as follows:

```
/* Fix the Slope component variance at 0 */
proc ucm data=stsm.VIOLENTCRIME;
   id Date interval=month;
   model MurdersTX;
   irregular;
   level;
   slope variance=0 noest;
   season length=12 type=dummy variance=0 noest;
   forecast lead=12 back=0 alpha=0.05;
   outlier;
run;
```

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

c. Select **Run** to fit the re-specified model.

The Fit Statistics table indicates that the penalized, overall fit of the model is slightly better.

Likelihood Based Fit Statistics	
Statistic	**Value**
Full Log Likelihood	-432.1
Diffuse Part of Log Likelihood	-4.97
Non-Missing Observations Used	108
Estimated Parameters	2
Initialized Diffuse State Elements	13
Normalized Residual Sum of Squares	95
AIC (smaller is better)	868.26
BIC (smaller is better)	873.37
AICC (smaller is better)	868.39
HQIC (smaller is better)	870.32
CAIC (smaller is better)	875.37

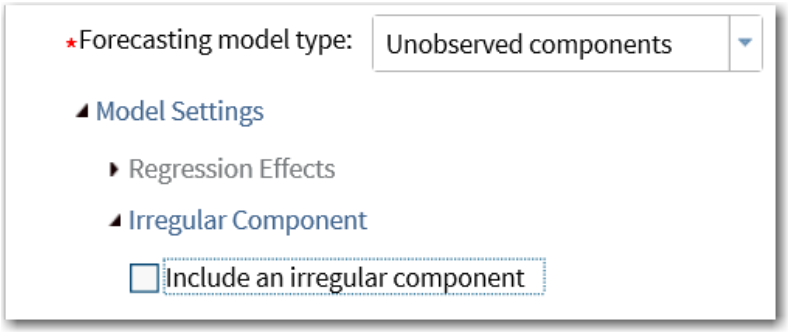
The Significance Analysis of Components table indicates that the irregular component is the least useful in terms of accounting for variation in **MurdersTX**.

Significance Analysis of Components (Based on the Final State)			
Component	**DF**	**Chi-Square**	**Pr > ChiSq**
Irregular	1	0.37	0.5442
Level	1	128.47	<.0001
Slope	1	1.01	0.3140
Season	11	59.03	<.0001

4. Refine the UCM. Remove the irregular component.

The irregular component is dropped from the model.

a. Clear the **Include an irregular component** check box.



Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

Alternatively, you can write the code directly as follows:

```
/* Remove the Irregular component */
proc ucm data=stsm.VIOLENTCRIME;
   id Date interval=month;
   model MurdersTX;
   level;
   slope variance=0 noest;
   season length=12 type=dummy variance=0 noest;
   forecast lead=12 back=0 alpha=0.05;
   outlier;
run;
```

The IRREGULAR statement was removed.

b. Select **Run** to fit the re-specified model.

The Fit Statistics table indicates that the penalized, overall fit of the model is substantially worse after removing the irregular component.

Likelihood Based Fit Statistics	
Statistic	**Value**
Full Log Likelihood	-452.4
Diffuse Part of Log Likelihood	-4.97
Non-Missing Observations Used	108
Estimated Parameters	1
Initialized Diffuse State Elements	13
Normalized Residual Sum of Squares	95
AIC (smaller is better)	906.87
BIC (smaller is better)	909.42
AICC (smaller is better)	906.91
HQIC (smaller is better)	907.9
CAIC (smaller is better)	910.42

5. The final model, assessed below, is the one fit in the next-to-the-last step.

a. Select the **Irregular** component on the MODEL tab.

b. In the Plots options on the bottom of the MODEL tab, select **One-step-ahead Forecasts** and all of the **Smoothed Component Estimates** plots.

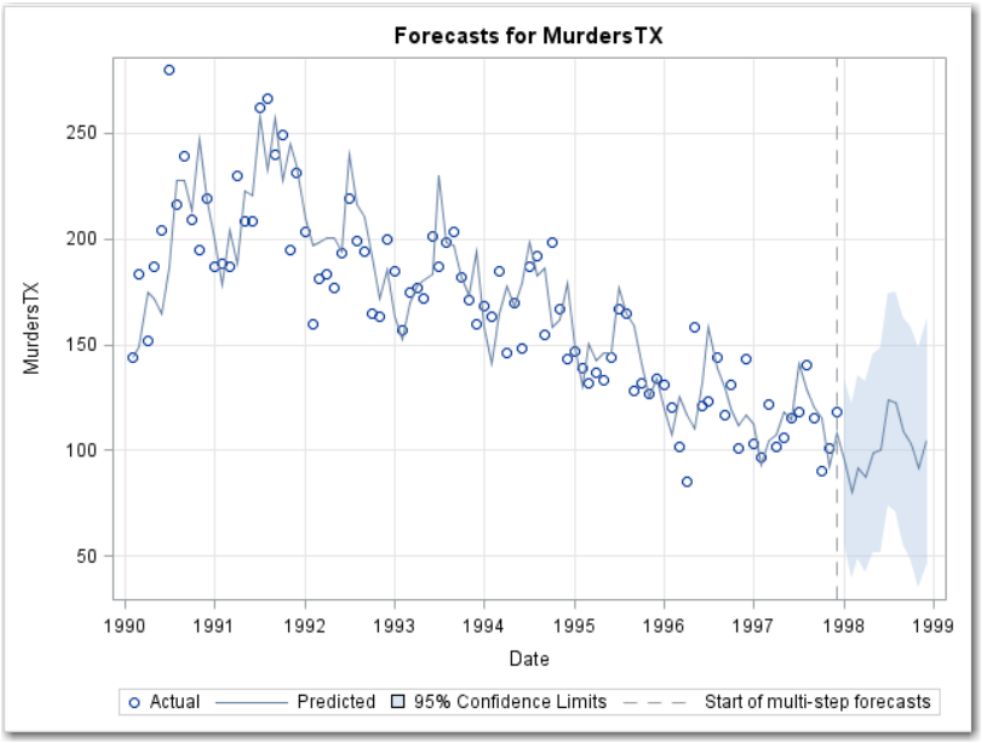
The final model syntax is shown below.

```
/* Add the Irregular component back in to get the Final model */
proc ucm data=stsm.VIOLENTCRIME;
   id Date interval=month;
   model MurdersTX;
   irregular plot=smooth;
   level plot=(smooth);
   slope variance=0 noest;
   season length=12 type=dummy variance=0 noest;
   estimate plot=(panel model loess);
   forecast lead=12 back=0 alpha=0.05 plot=(forecasts);
   outlier;
run;
```

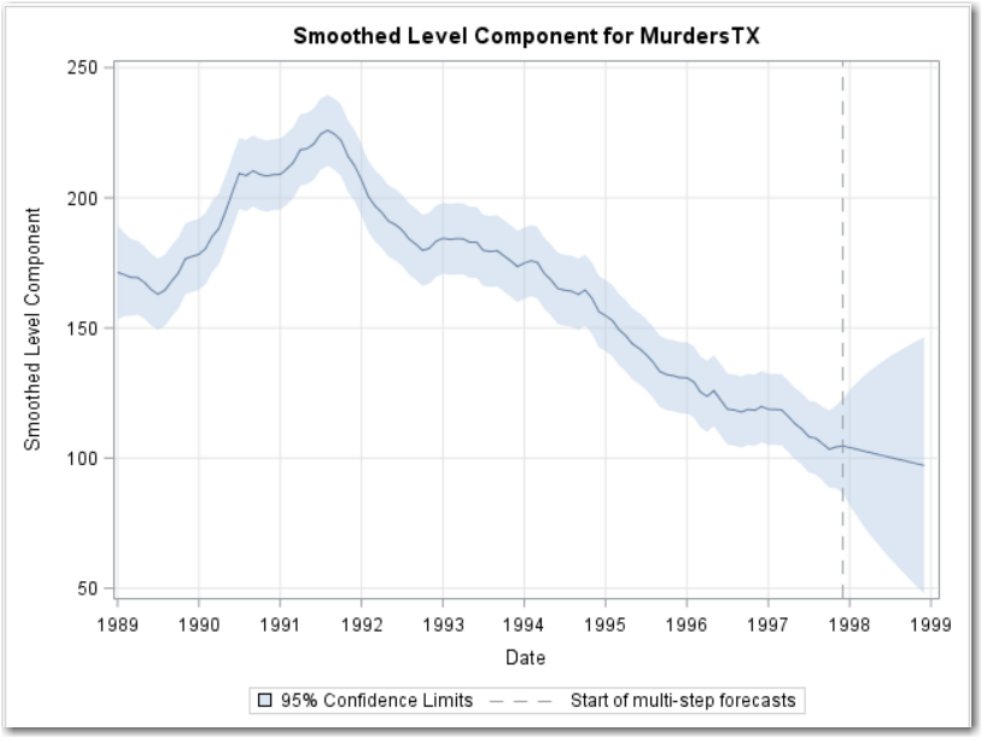
Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

c. Select **Run** to submit the final model.

The overall model does a good job of accommodating and extrapolating the salient features of the data.

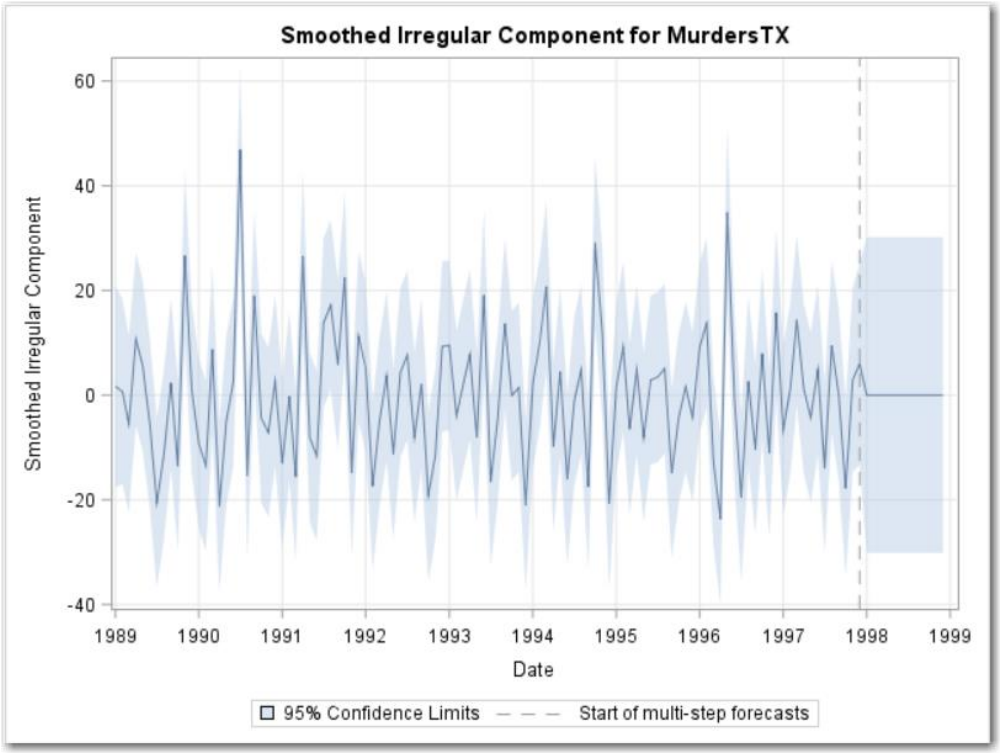


The Level Component plot illustrates how the level of the series changes as a function of time.

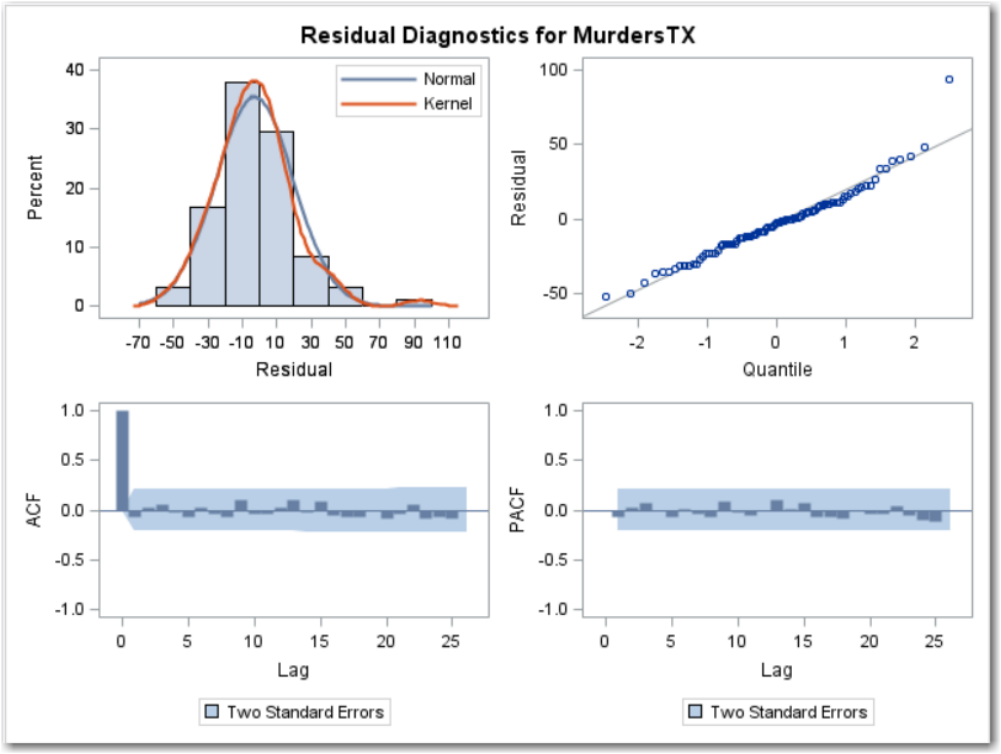


Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

The Irregular Component plot represents the estimated, stationary variation in the series.



The Residual Diagnostics panel indicates that the model is adequately specified.



End of Demonstration

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

Appendix A References

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

A.1 References

Brown, R.G. 1959. *Statistical Forecasting for Inventory Control*. McGraw-Hill: New York, NY.

Brown, R.G. 1962. *Smoothing, Forecasting, and Prediction of Discrete Time Series*. Prentice-Hall: Englewood Cliffs, NJ.

Fomby, Thomas B. June 2008. *Exponential Smoothing Models*. Southern Methodist University. Dallas, TX.

Holt, C.C. et al. 1960. *Planning Production, Inventories, and Work Force*. Prentice-Hall: Englewood Cliffs, NJ. (Chapter 14).

SAS Institute Inc. 2015. *SAS/ETS 14.1 User's Guide*. Cary, NC: SAS Institute Inc.

Yaffee, R. and M. McGee. 2000. *Introduction to Time Series Analysis and Forecasting with Applications of SAS and SPSS*. Academic Press: San Diego, CA.

Yule, G.U. *Journal of the Royal Statistical Society*. Vol. 89. No. 1. (Jan. 1926) pp. 1-63.

ARIMA:

Box, G.E.P., G.M. Jenkins, and G.C. Reinsel. 1994. *Time Series Analysis: Forecasting and Control*. Upper Saddle River, NJ: Prentice Hall.

Brockwell, P.J. and R.A. Davis. 1992. *Time Series: Theory and Methods*. New York, NY: Springer-Verlag Inc.

UCM:

Harvey, Andrew C. 1989. *Forecasting, Structural Time Series Models and the Kalman Filter*. Cambridge, UK: Cambridge University Press.

Koopman, Siem Jan, and James Durbin. 2001. *Time Series Analysis by State Space Methods*. Oxford, UK: Oxford University Press.

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.