

Chapter 2 ARIMAX Models

2.1	Autocorrelation and White Noise.....	2-3
	Demonstration: Predictability of Dice Rolls	2-5
	Demonstration: Autocorrelation and Solar Production.....	2-16
2.2	ARIMA, ARMA, and Stationarity.....	2-22
	Demonstration: Time Series Identification	2-31
	Exercises	2-34
2.3	Estimation of Autoregressive Parameters	2-35
	Demonstration: Estimation, Residual Analysis, and Goodness-of-Fit	2-42
	Exercises	2-49
2.4	ARMAX and Time Series Regression	2-50
	Demonstration: Cloud Cover and Solar Power	2-57
	Demonstration: Estimation of Cloud Cover	2-61
	Exercises	2-68
2.5	Forecasting and Accuracy Assessment.....	2-70
	Demonstration: Forecasting a Holdout Sample Using the ARIMA Model.....	2-80
	Demonstration: Forecasting a Holdout Sample Using the ARIMAX Model	2-85
	Demonstration: Comparing Models Using MAPE	2-88
	Demonstration: Forecasting Future Values Using the Champion Model.....	2-92
	Exercises	2-95
2.6	Solutions	Error! Bookmark not defined.
	Solutions to Exercises	Error! Bookmark not defined.
	Solutions to Student Activities (Polls/Quizzes)	Error! Bookmark not defined.
2.7	Chapter Summary.....	2-97

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

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?





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:

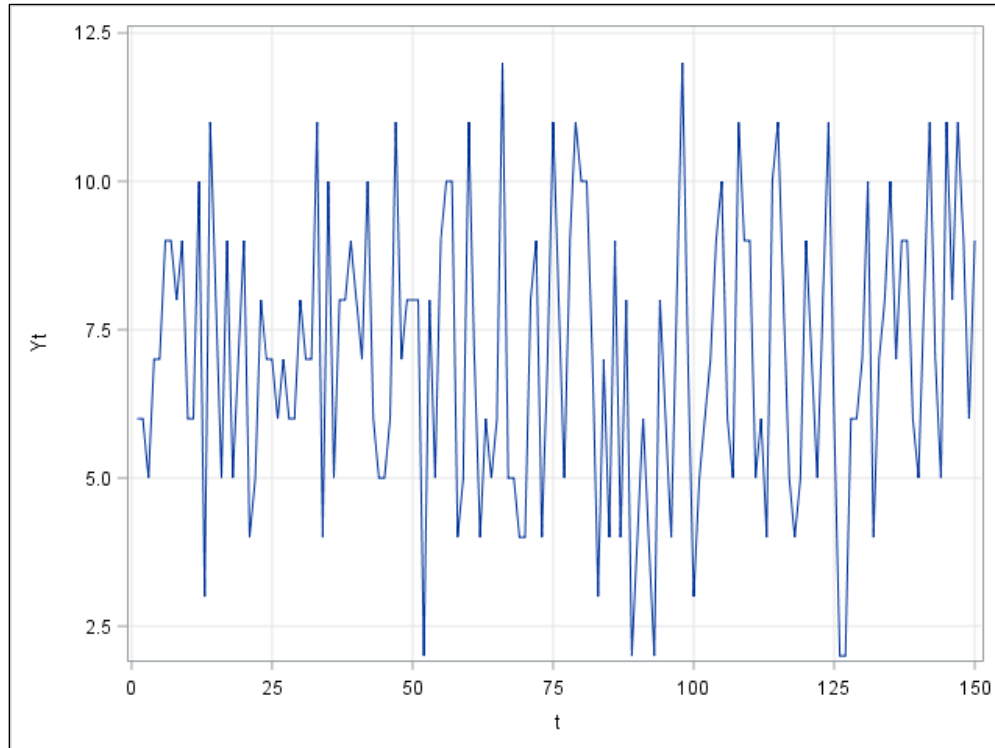
- **t**: the ordered value of the roll of the dice
- **Yt**: the sum of the two dice at roll **t**
- **Ytmin1**: the sum of the two dice on the previous roll
- **Ytmin2**: the sum of the two dice from the previous two rolls
- **Ytmin3**: the sum of the two dice from the previous three rolls
- ..., and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

The screenshot shows the SAS Studio interface with the 'Series Plot' task selected in the 'Tasks' pane on the left. The main workspace displays the configuration for the 'Series Plot' task, which is set to use the 'STSM.DICE2' data set. The 'X variable' is set to 't' and the 'Y variable' is set to 'Yt'. The 'Group variable' is set to 'Column' and the 'URL variable' is also set to 'Column'. The 'DATA' tab is active, showing the 'STSM.DICE2' data set and the 'WHERE CLAUSE FILTER' and 'ROLES' sections.

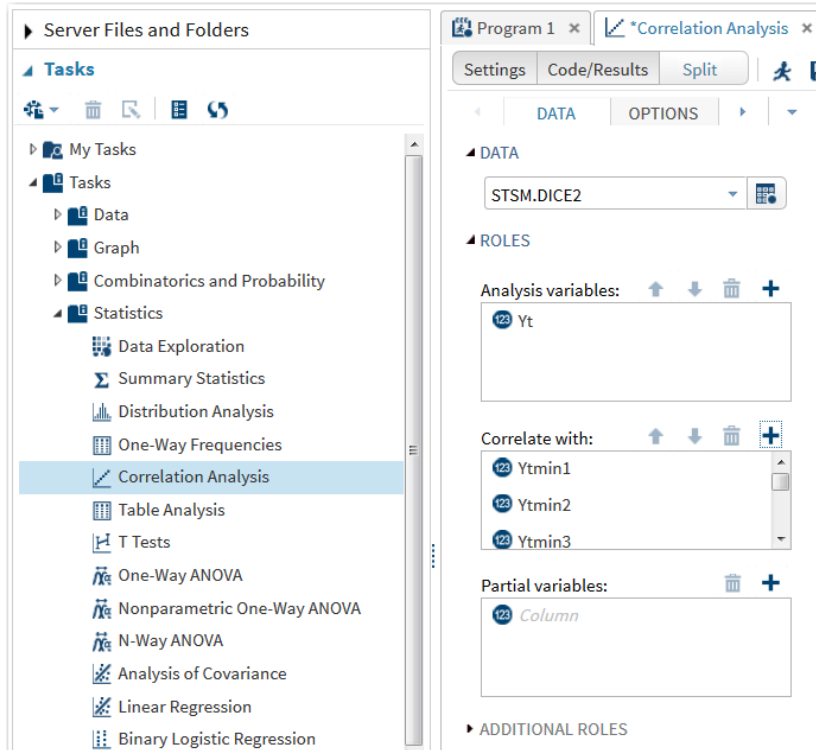
The code generated by SAS Studio is as follows:

```
/* 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 Y_t are between 2 and 12.

- Using the **stsm.dice2** data set and the Correlation Analysis task, determine the autocorrelations between Y_t and all lags of Y_t through lag 12 ($Y_{tmin1}-Y_{tmin12}$).



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

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.



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

sas THE POWER TO KNOW.

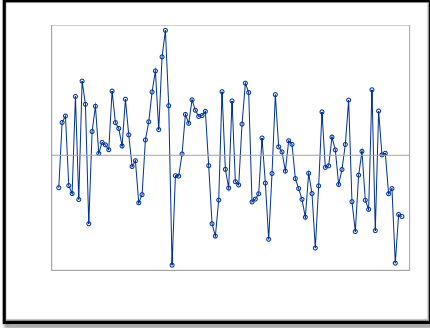
Correlation of Y with Past Y: *Autocorrelation*

Autocorrelation (Order 1):

Y_t is correlated with Y_{t-1}

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

First Lag:
 $Y_{t-1} = Y(t-1)$

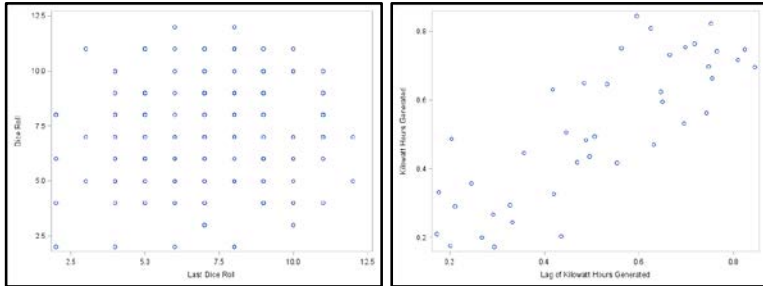


7

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.

sas THE POWER TO KNOW.

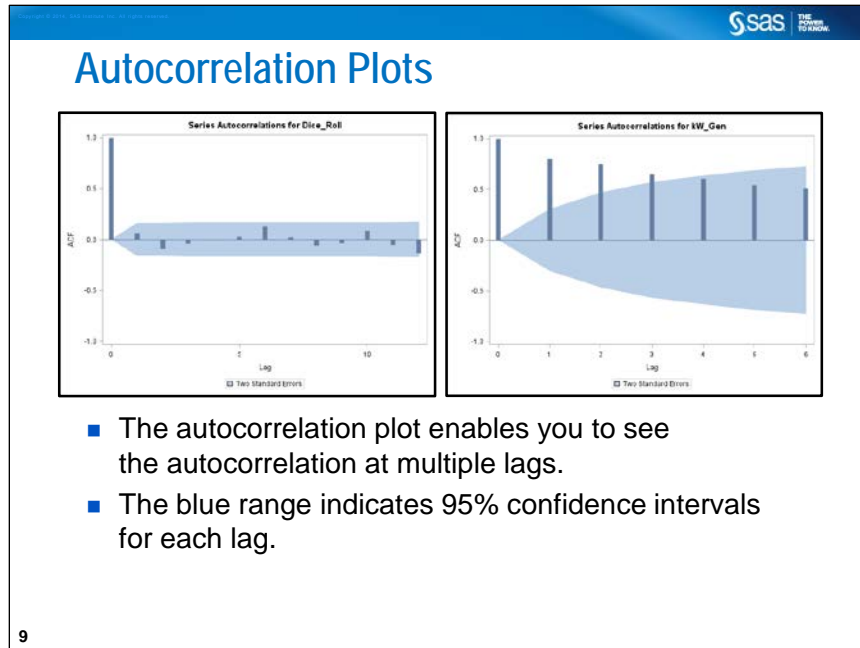
Autocorrelation Scatter Plots



- 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.

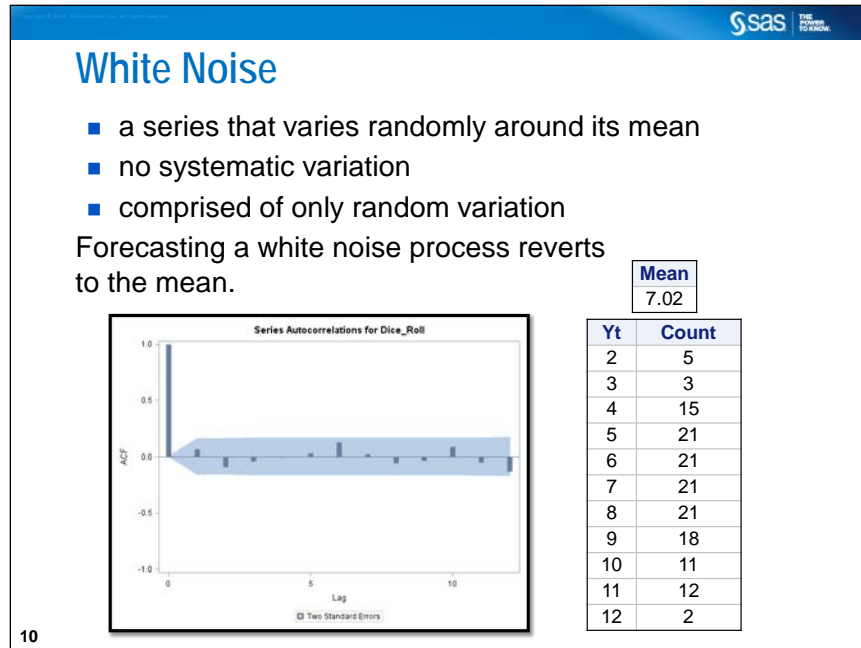


Because autocorrelation could theoretically exist with any ordered lag (Y_{t-p}), it is not reasonable to try to create scatter plots for all. Autocorrelation function (ACF) plots are useful as a first step in detecting potential autocorrelation in a time series. Each spike represents the autocorrelation 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.)



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.

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.

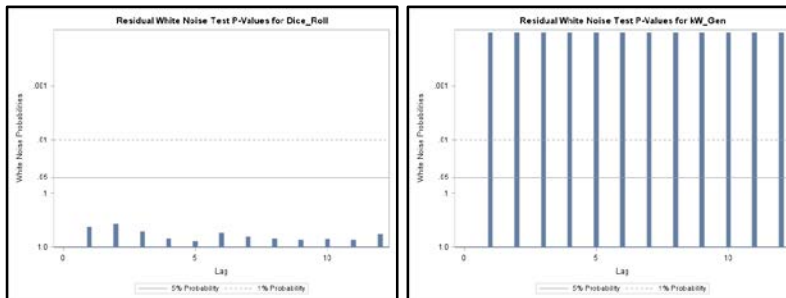
11

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

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

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.



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	

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.



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



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

The screenshot shows the SAS Studio interface with the 'Time Series Exploration' task selected in the left-hand navigation pane. The main workspace displays the configuration for the task, including the following sections:

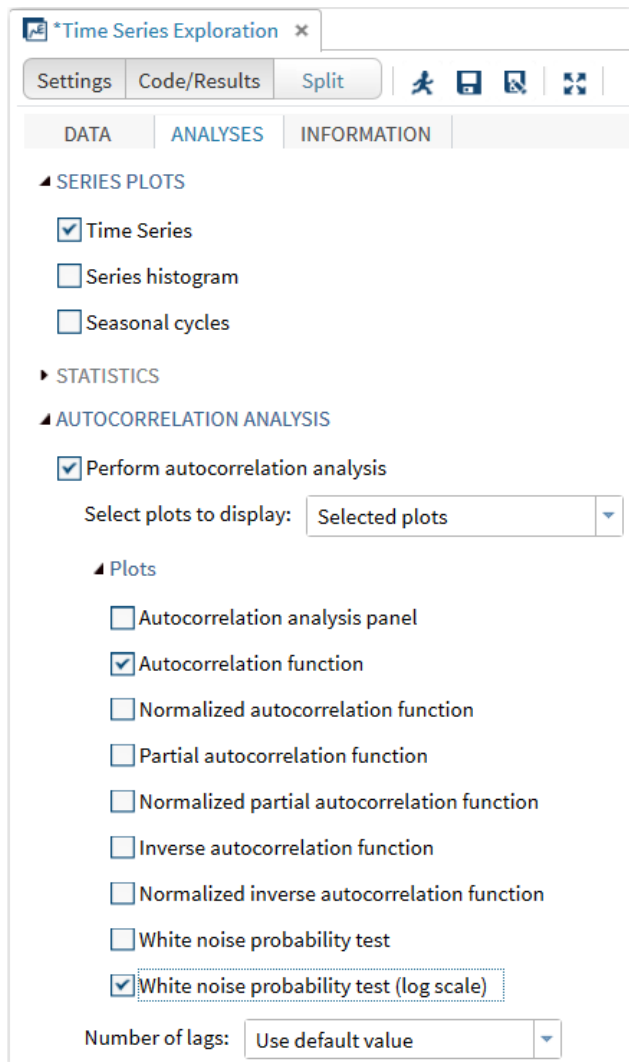
- DATA**: The data source is set to 'STSM.SOLARPV'.
- ROLES**: The dependent variable is 'kW_Gen'.
- Independent variables**: A placeholder for 'Column' is shown.
- Transformations**: A table showing transformation settings for the 'kW_Gen' variable.

Variable	Accumulation	Transformation	Simple Differ...	Seasonal Differ...
kW_Gen	None	None	0	0
- ADDITIONAL ROLES**: The time ID is set to 'EDT'.
- Properties**: The interval is set to 'Week', the multiplier is 1, the shift is 1, and the season length is 52.

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.



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;
```



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;
```

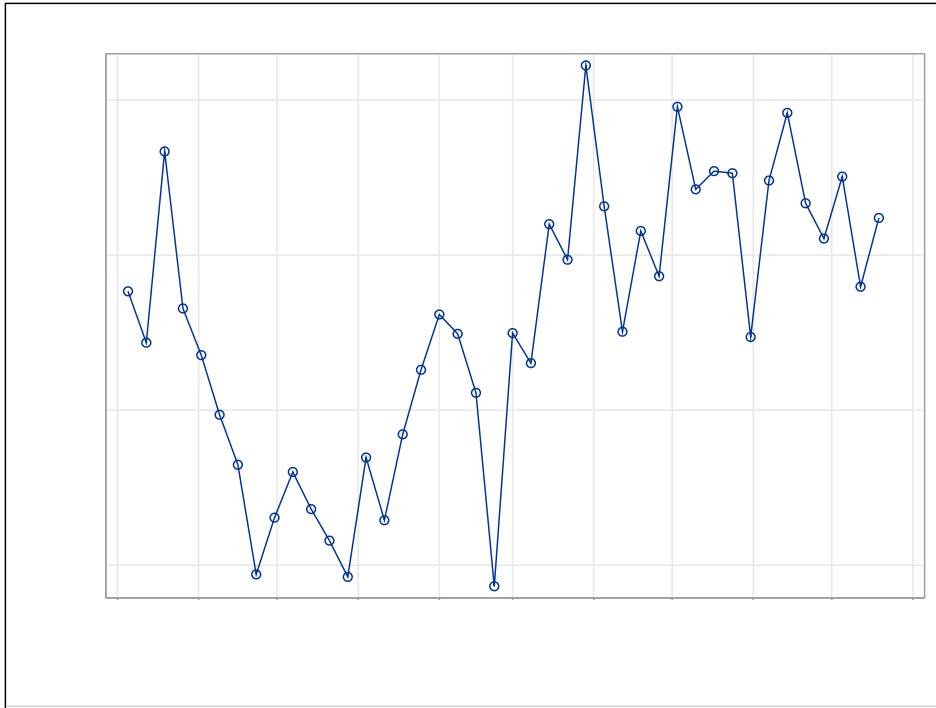
SAS Studio uses a **Work.TempSorted** data set to do the analysis instead of the original data set **stsm.solarpv**. This is true for other tasks in SAS Studio as well, and ensures that the original data set remains unchanged throughout the entire process. Throughout the remainder of Chapter 2, the displayed 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

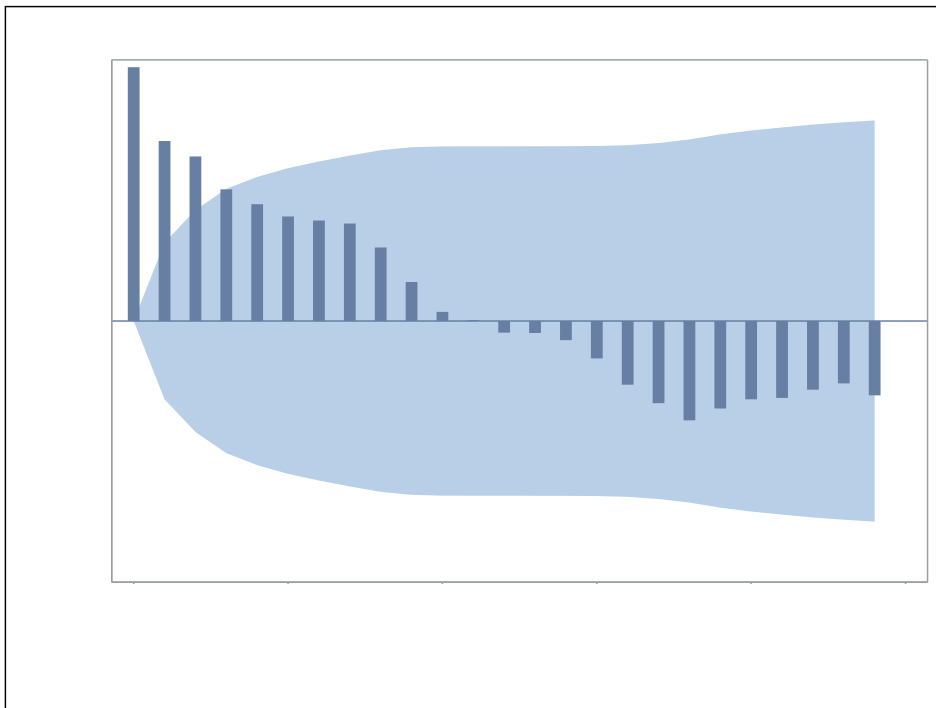
The Variable Information table provides basic information about the series, and enables confirmation that the appropriate data and time range are being used.

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

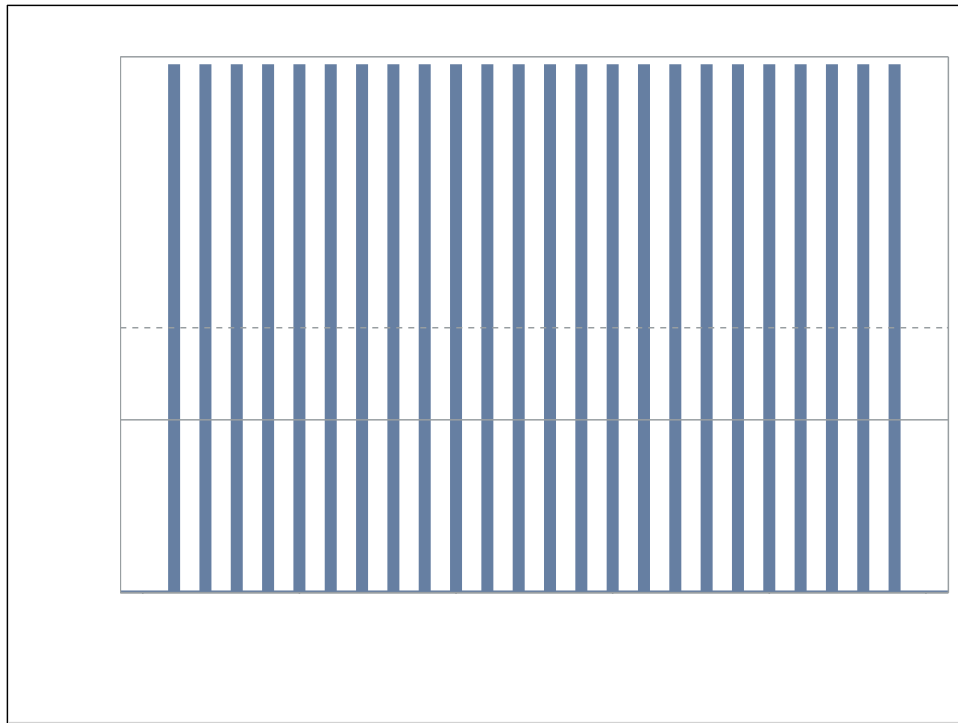
A plot of the data is generated as part of the output. Because no transformation was applied, the plot is of the raw **kW_Gen** series. Because the Time Series Exploration task creates this plot, graphing the series using the Graph task in SAS Studio is unnecessary.



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.



End of Demonstration

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.

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

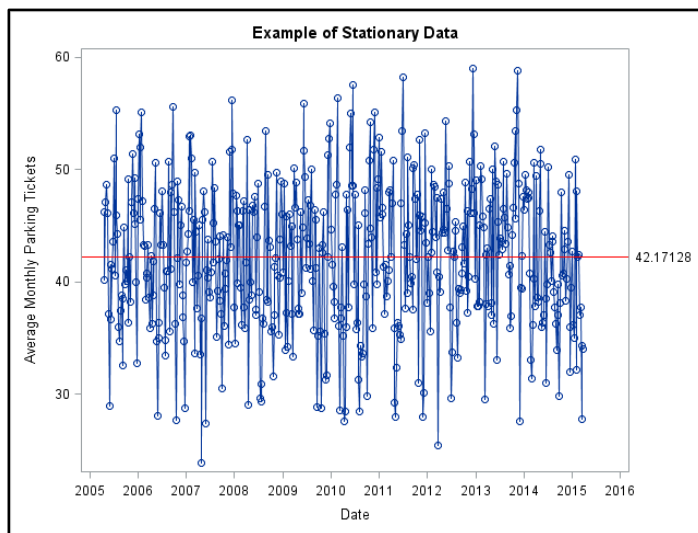
Stationarity

- 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

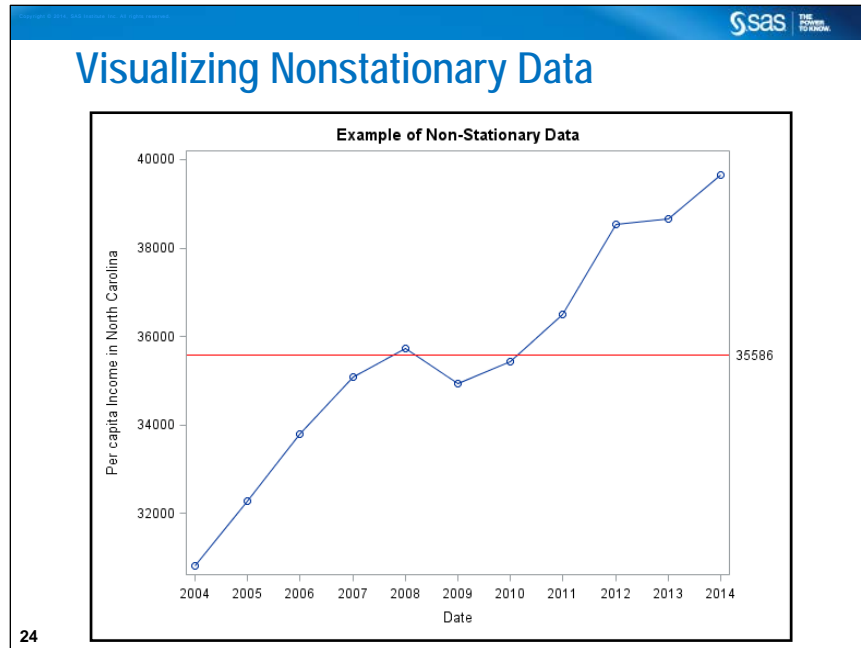
ARMA and ARIMA models follow very similar methodology. The main distinction is that ARMA models are used when the series is stationary on its original scale. If the series is not stationary on its original scale, it needs to be transformed to create a stationary series through integration, and thus ARIMA.

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.



Seasonal and trending data can be quickly identified as nonstationary.

sas THE POWER TO KNOW

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

sas THE POWER PROGRAM

First Differencing Example

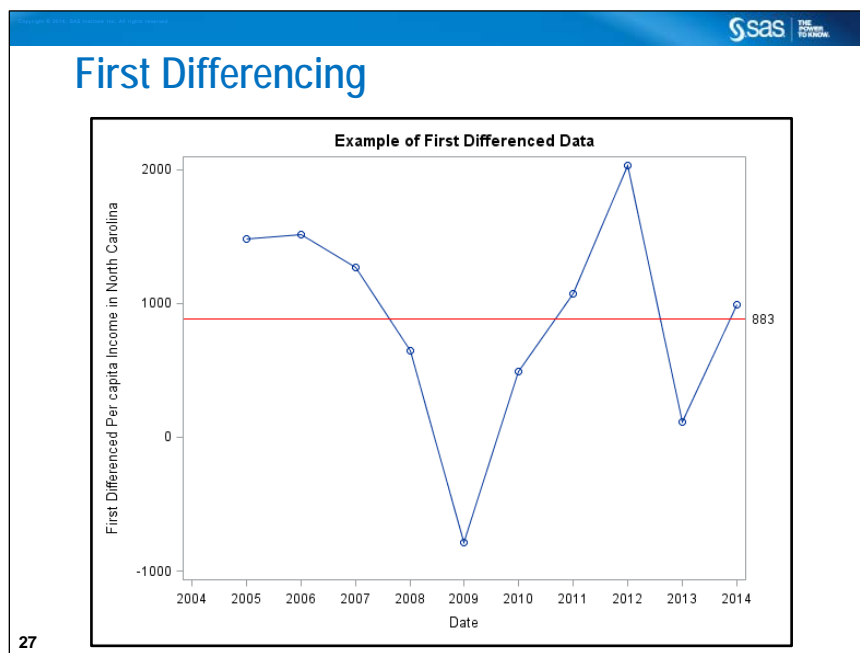
Year	Income	Lag(Income)	First Difference
2004	\$ 30,818	⚠	⚠
2005	\$ 32,296	\$ 30,818	\$ 1,478
2006	\$ 33,808	\$ 32,296	\$ 1,512
2007	\$ 35,076	\$ 33,808	\$ 1,268
2008	\$ 35,725	\$ 35,076	\$ 649
2009	\$ 34,942	\$ 35,725	\$ -783
2010	\$ 35,435	\$ 34,942	\$ 493
2011	\$ 36,508	\$ 35,435	\$ 1,073
2012	\$ 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.



ARMA versus ARIMA Models

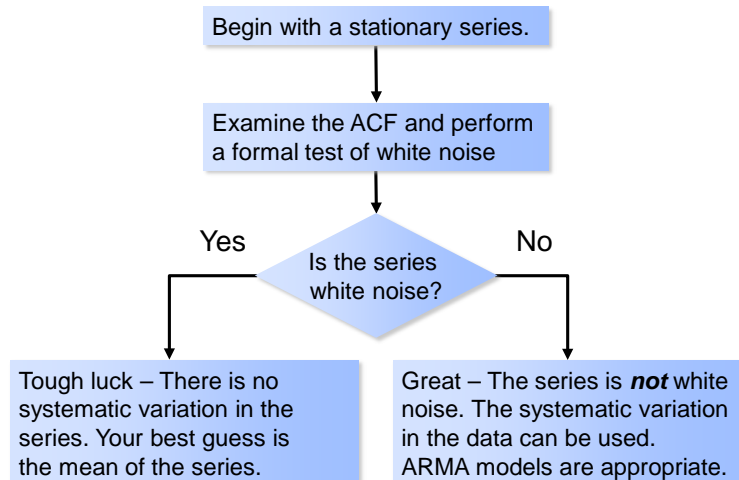
The “I” in ARIMA stands for *integrated*, and tells you in what *order* the data was differenced to convert it to a stationary process.

- 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



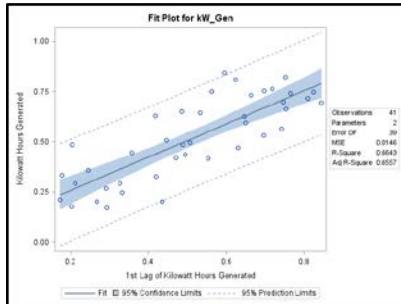
29

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

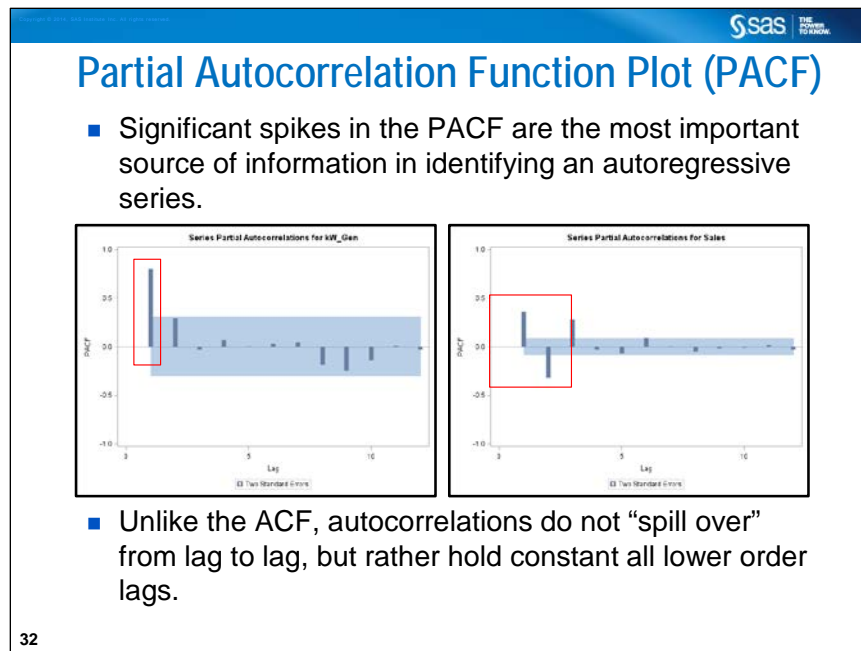
Determining Autoregressive Order

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

The partial autocorrelation function plot (PACF) helps determine this by answering the following questions:

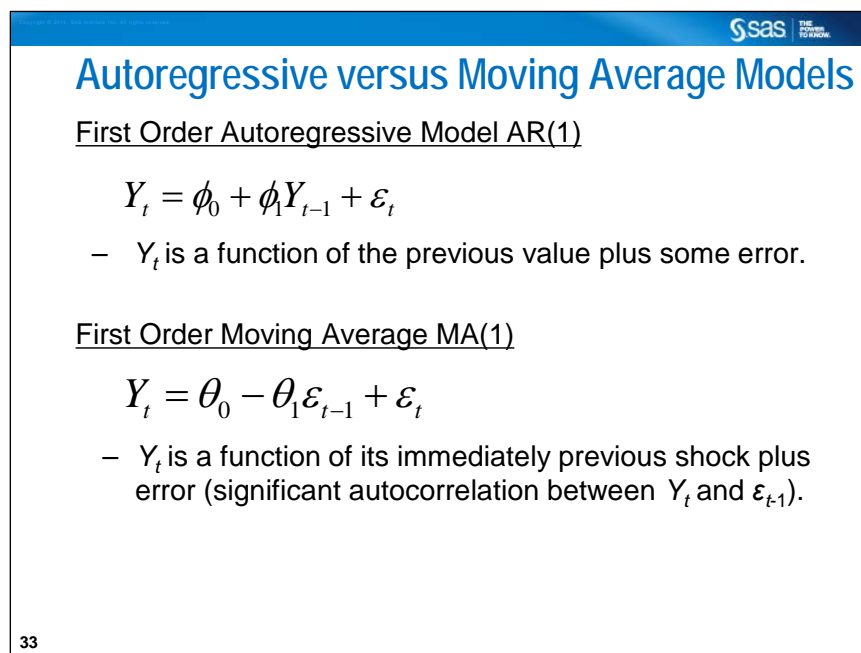
- Is there significant autocorrelation between Y_t and Y_{t-1} ?
- Is there significant autocorrelation between Y_t and Y_{t-2} , holding constant the autocorrelation between Y_t and Y_{t-1} ?

31



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.



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



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



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.

The screenshot shows the 'Time Series Exploration' task in SAS Studio. The 'DATA' tab is active, displaying the following configuration:

- DATA:** STSM.SOLARPV
- ROLES:**
 - Dependent variable:** kW_Gen
 - Independent variables:** Column
- Transformations:**

Variable	Transfo
kW_Gen	None
- ADDITIONAL ROLES:** (collapsed)

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.

The screenshot shows the 'ADDITIONAL ROLES' tab in the 'Time Series Exploration' task. The configuration is as follows:

- Time ID:** EDT
- Properties:**
 - Interval:** Week
 - Multiplier:** 1
 - Shift:** 1
 - Season length:** 52
- Group analysis by:** Column

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:

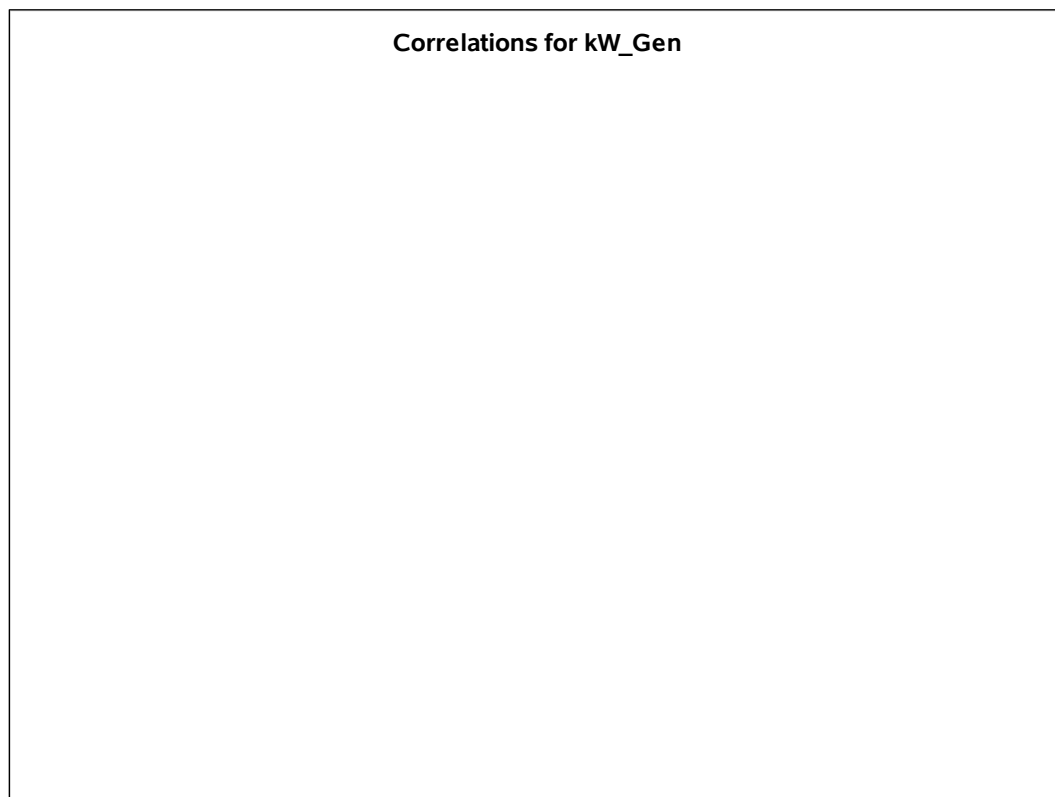
```
proc timeseries data=STSM.SOLARPV plots=(corr);
  id EDT interval=week;
  var kW_Gen;
run;
```

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.



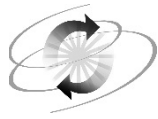
The IACF is the inverse autocorrelation function. If the model is a pure autoregressive model, then the IACF is an ACF that corresponds to a pure moving average model. It cuts off sharply when the lag is greater than p . This behavior is similar to the behavior of the partial autocorrelation function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

End of Demonstration

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



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

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

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.

sas THE POWER OF DATA

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.

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

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.

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

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 $\mu(1-\phi_1)$.

sas THE POWER OF DATA

Model Goodness-of-Fit Statistics

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

Akaike's A Information Criteria:

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

Schwarz's Bayesian Information Criteria

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

- 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 $-2\log(L)$, 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 $-2\log(L)$ is affected by the number of parameters. Values of $-2\log(L)$ 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.

sas THE POWER TO KNOW

Check of Residuals

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

Residual Normality Diagnostics for Sales

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.

sas THE POWER TO KNOW


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

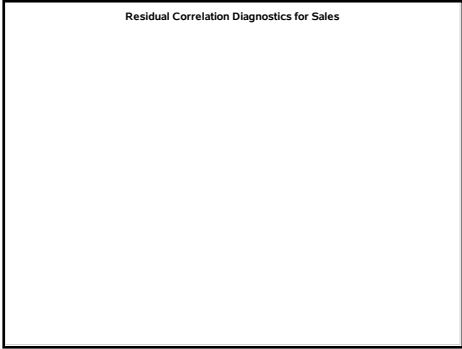
You can check autocorrelations of the residuals using the Ljung-Box test for white noise. In addition to checking p -values, you might also want to check the individual autocorrelation estimates to make sure that there is not one suspiciously high value.



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**

Residual Correlation Diagnostics for Sales



52

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



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.

*Modeling and Forecasting x

Settings Code/Results Split

DATA MODEL OPTIONS OUTPUT INFORMATION

DATA

MARC.SOLARPV

NOTE

This task requires data in a valid time series format. To prepare your data, run the Time Series Data Preparation task before starting this task.

ROLES

Dependent variable (1 item)

kW_Gen

ADDITIONAL ROLES

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.

ADDITIONAL ROLES

Time ID (1 item)

EDT

Properties

Interval: Week

Multiplier: 1

Shift: 1

Season length: 52

Group analysis by

Column

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.

The screenshot shows the 'MODEL' tab of a forecasting model settings window. The 'Forecasting model type' is set to 'ARIMA'. Under 'Model Settings', the 'ARIMA' section is expanded, showing 'Autoregressive order (p)' set to 1, 'Differencing order (d)' set to 0, and 'Moving average order (q)' set to 0. The 'Seasonal ARIMA' section is also expanded, showing 'Autoregressive order (P)' set to 0, 'Differencing order (D)' set to 0, and 'Moving average order (Q)' set to 0. The 'Include intercept in model' checkbox is checked. The 'Plots' section is collapsed.

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

The screenshot shows the 'Plots' section expanded in the 'MODEL' tab. A dropdown menu is open, showing the following options: 'Default plots', 'Selected plots', 'All plots', and 'No plots'. A mouse cursor is pointing at the 'Selected plots' option.

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.

The screenshot shows the SAS software interface with the **MODEL** tab selected. The **Plots** section is expanded, showing the following options:

- Series Plots**
 - ☐ Autocorrelations plot
 - ☒ Panels of correlation plots
 - ☐ Panels of cross-correlation plots
 - ☐ Inverse-autocorrelations plot
 - ☐ Partial-autocorrelations plot
- Residual Plots**
 - ☐ Residual autocorrelations plot
 - ☒ Panel of the residual correlation diagnostics
 - ☐ Histogram of the residuals
 - ☐ Residual inverse-autocorrelations plot
 - ☒ Panel of the residual normality diagnostics
 - ☐ Residual partial-autocorrelations
 - ☐ Normal quantile plot of the residuals
 - ☐ Scatter plot of the residuals against time
 - ☐ Ljung-Box white-noise test p-values at different lags
- Forecast Plots**
 - ☐ One-step-ahead and multistep-ahead forecasts
 - ☐ Multistep-ahead forecasts in the forecast region

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**.

The screenshot shows the SAS software interface with the **OPTIONS** tab selected. The **FORECAST SETTINGS** section is expanded, showing the following options:


- Number of periods to forecast:** 0
- Forecast confidence level:** 95%
- Number of periods to hold back:** 0

The **OUTLIER DETECTION** section is also expanded, showing the following option:

- ☐ Perform outlier detection

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.

Trend and Correlation Analysis for kW_Gen

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

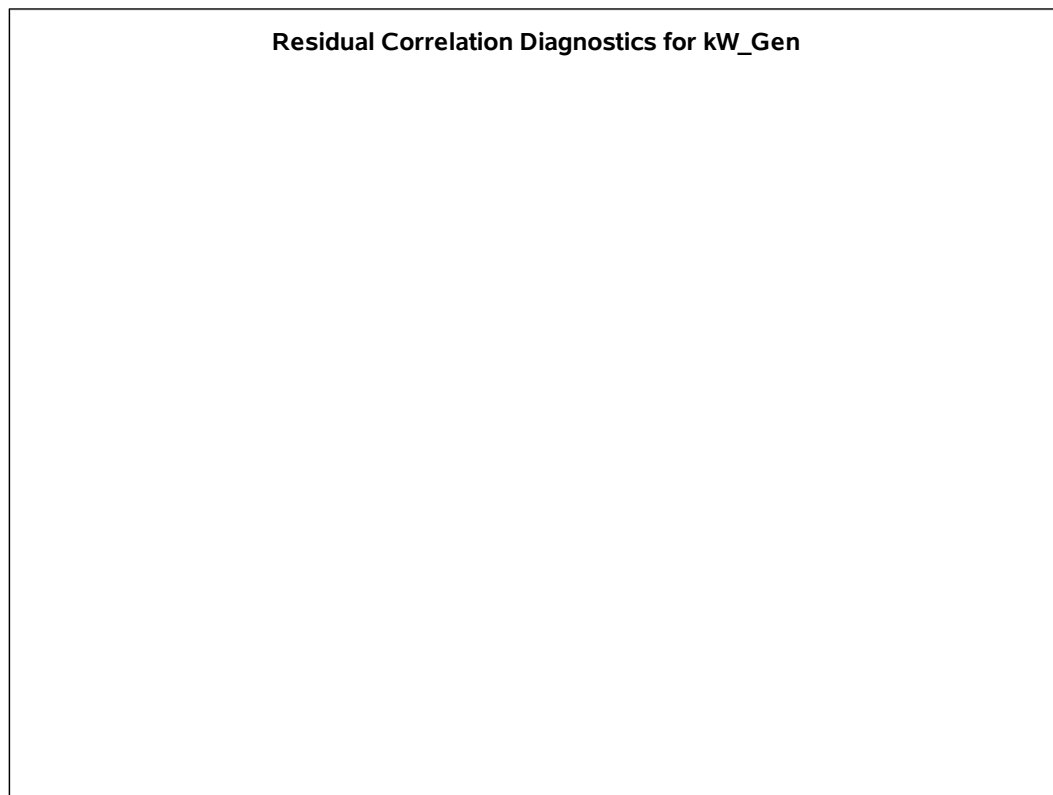
The correlations of parameter estimates can be used to check the multicollinearity of parameters. In this case, the correlation, as expected, is very low at 0.055.

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

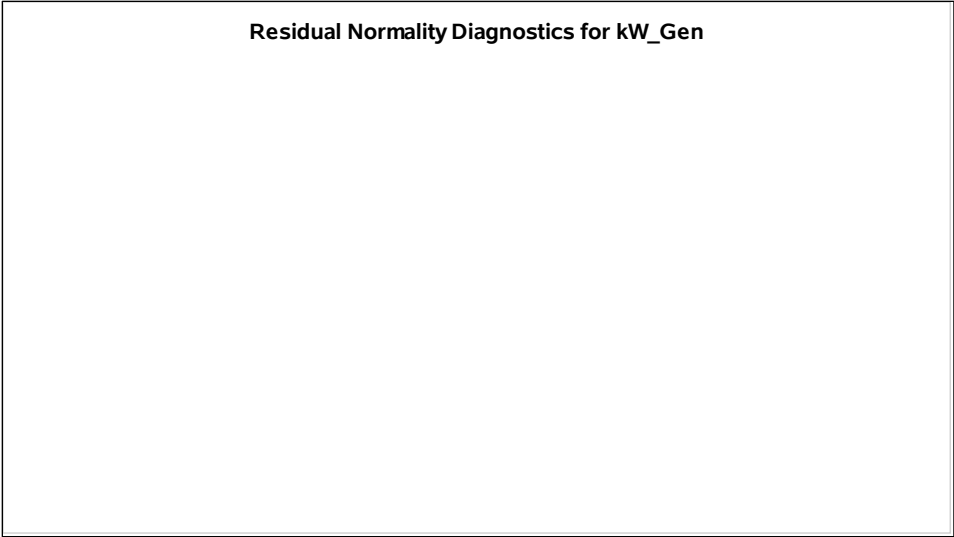
If this model is correct, the residuals from it should be white noise. The autocorrelation check of the first 24 lags indicates no particularly strong autocorrelation, although one value is above 0.2 in absolute value.

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).

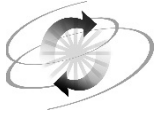


The estimated mean of the model is reiterated and the estimate for the autoregressive parameter is expressed in a particular model formulation.

Model for variable kW_Gen	
Estimated Mean	0.520193

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

End of Demonstration



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

2.4 ARMAX and Time Series Regression

Objectives

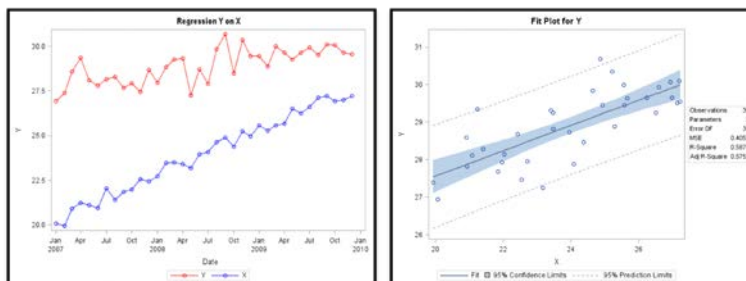
- Explain the X in ARMAX.
- Relate linear regression with time series regression models.
- Examine linear regression assumptions.
- Explain the relationship between ordinary multiple linear regression models and time series regression models.

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.

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 $t, t-1, t-2, \dots$, influences Y at time t .

Transfer Function

- a function that provides the mathematical relationship between a dynamic regressor and the target variable

58

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 \Rightarrow 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



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

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

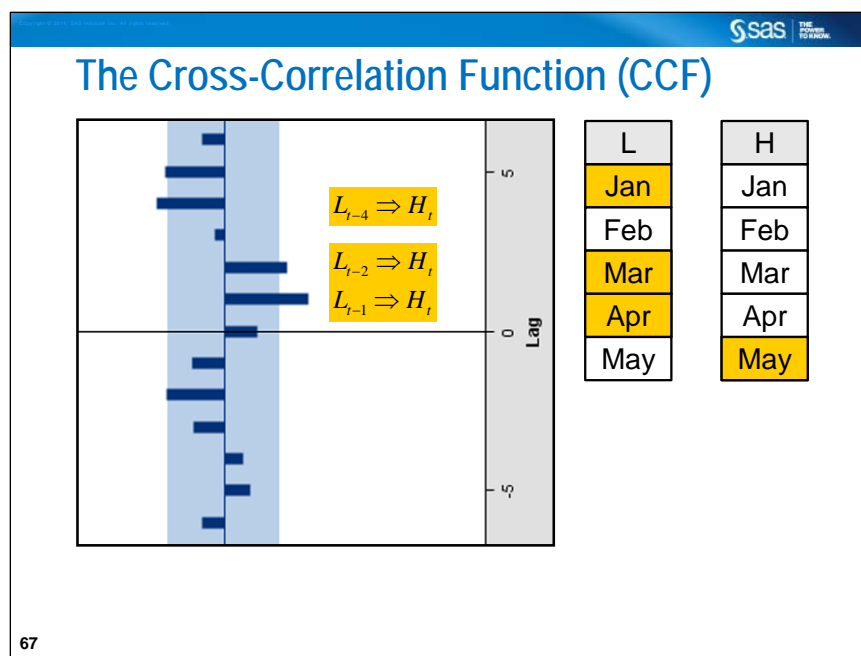
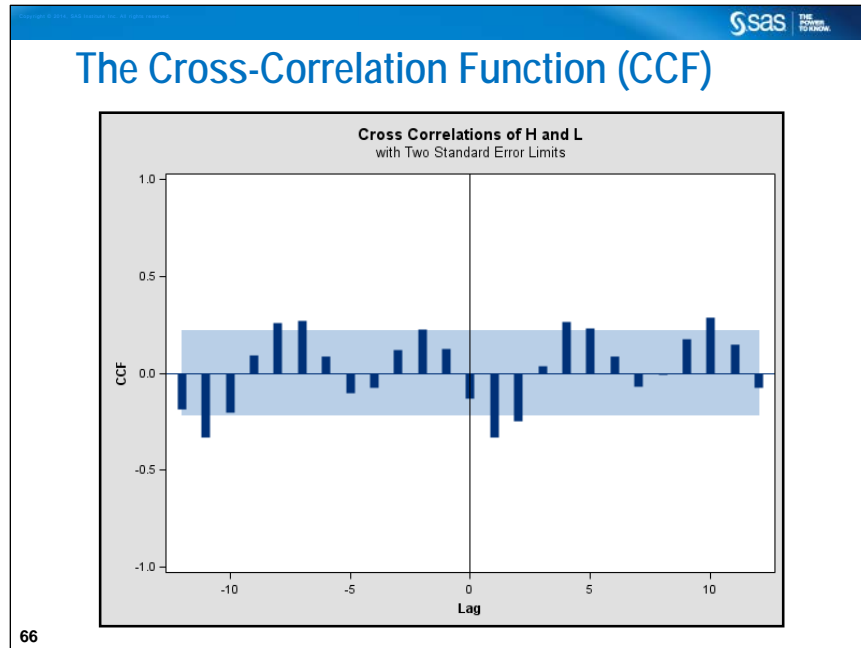
The Cross-Correlation Function (CCF)

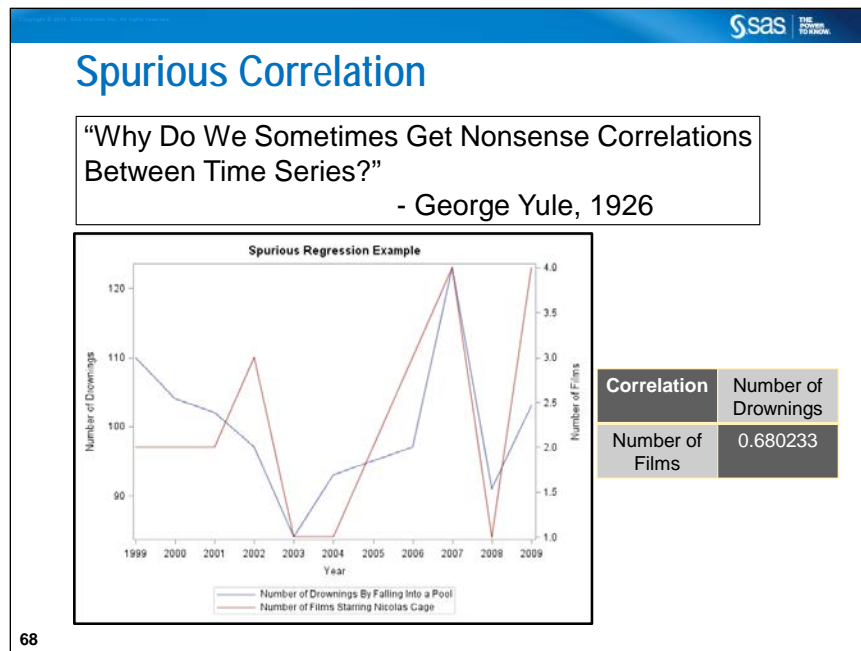
$CCF(k)$ is the cross-correlation of target Y with input X at lag k .

- 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...





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.



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.

The screenshot shows the 'DATA' tab of a software interface. Under 'DATA', the dataset 'MARC.SOLARPV' is selected. Under 'ROLES', 'kW_Gen' is assigned as the dependent variable and 'Cloud_Cover' as the independent variable. Under 'ADDITIONAL ROLES', 'EDT' is selected as the Time ID. The 'Transformations' table shows 'None' for both variables. The 'Properties' section shows 'Interval' set to 'Week', 'Multiplier' at 1, 'Shift' at 1, and 'Season length' at 52.

Variable	Accumulation	Transformation	Simple Diff
kW_Gen	None	None	0
Cloud_Cover	None	None	0

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

The screenshot shows the 'ANALYSES' tab. Under 'SERIES PLOTS', the 'Time Series' checkbox is unchecked. Under 'AUTOCORRELATION ANALYSIS', the 'Perform autocorrelation analysis' checkbox is also unchecked. Other analysis options like 'STATISTICS', 'CROSS-CORRELATION ANALYSIS', 'DECOMPOSITION ANALYSIS', 'SPECTRAL DENSITY ANALYSIS', and 'UNIT ROOT TEST ANALYSIS' are listed but not expanded.

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 screenshot shows the SAS ETS software interface with the **ANALYSES** tab selected. The **CROSS-CORRELATION ANALYSIS** section is expanded, and the **Perform cross-correlation analysis** checkbox is checked. Under the **Plots** sub-section, the **Cross-series** and **Cross-correlation function** checkboxes are also checked. The **Number of lags** is set to **Use default value**.

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.

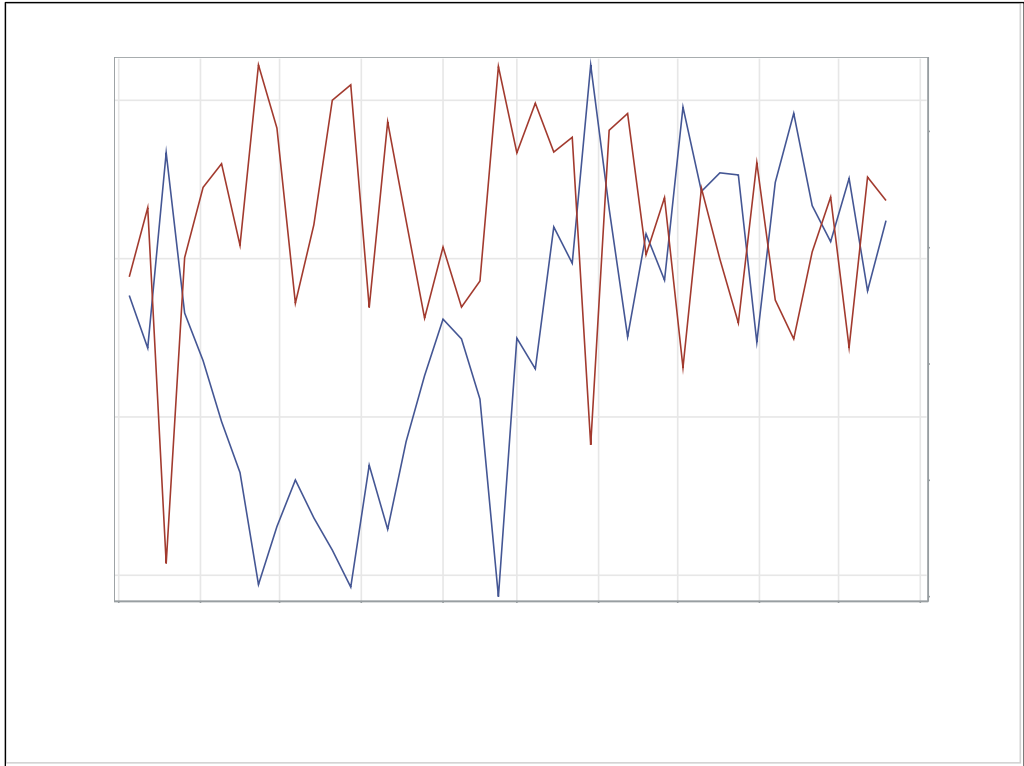
```
/* STSM02d04a.sas */
proc timeseries data=STSM.SOLARPV
                crossplots=(series ccf);
  id EDT interval=week;
  var kW_Gen;
  crossvar Cloud_Cover;
  ods exclude CCFNORMPlot;
run;
```

5. Submit the code.

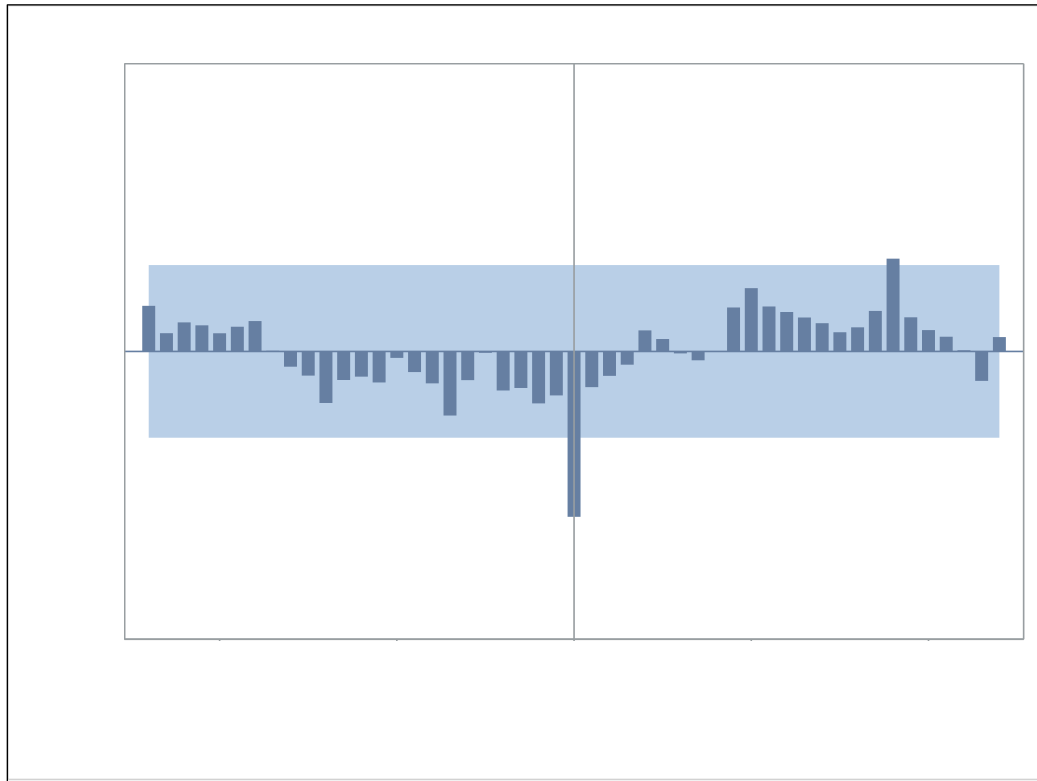
Input Data Set	
Name	WORK.TEMPSorted
Label	
Time ID Variable	EDT
Time Interval	WEEK
Length of Seasonal Cycle	52

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.



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



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 ($p < .0001$).

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

Residual Correlation Diagnostics for kW_Gen

The residuals are relatively normally distributed.

Residual Normality Diagnostics for kW_Gen

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

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.

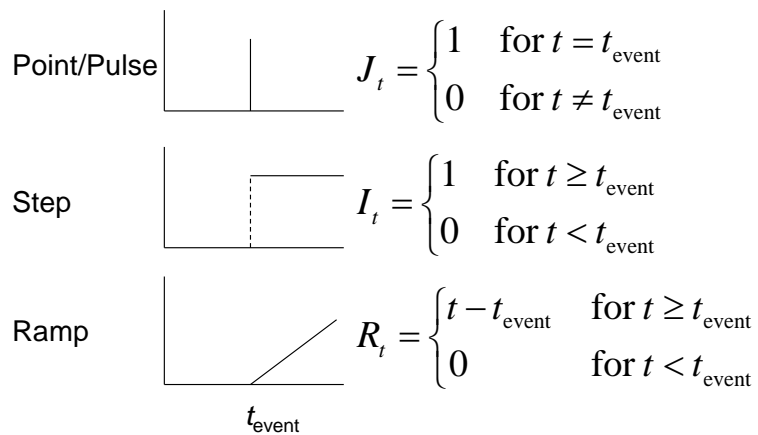
72

Event and Intervention Analysis Practices

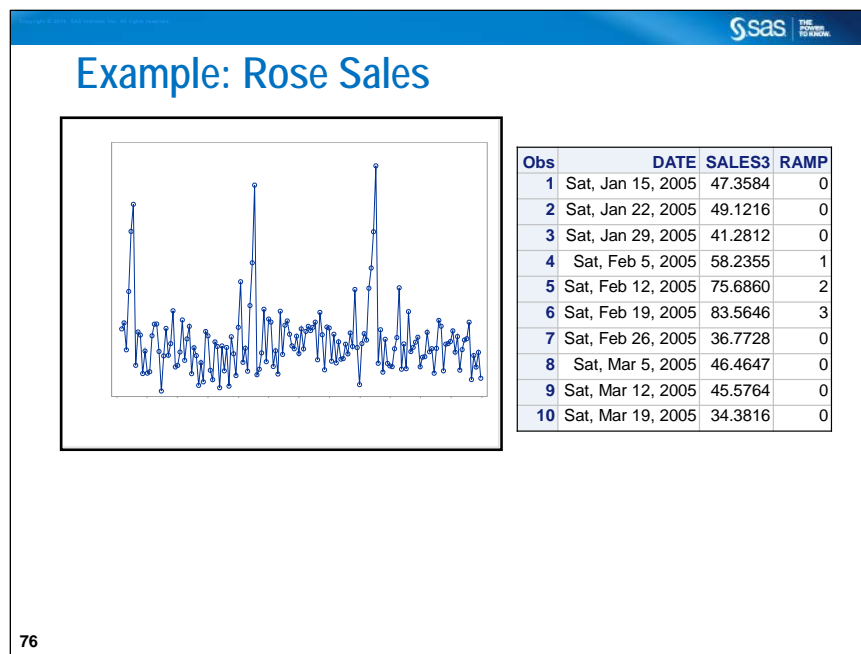
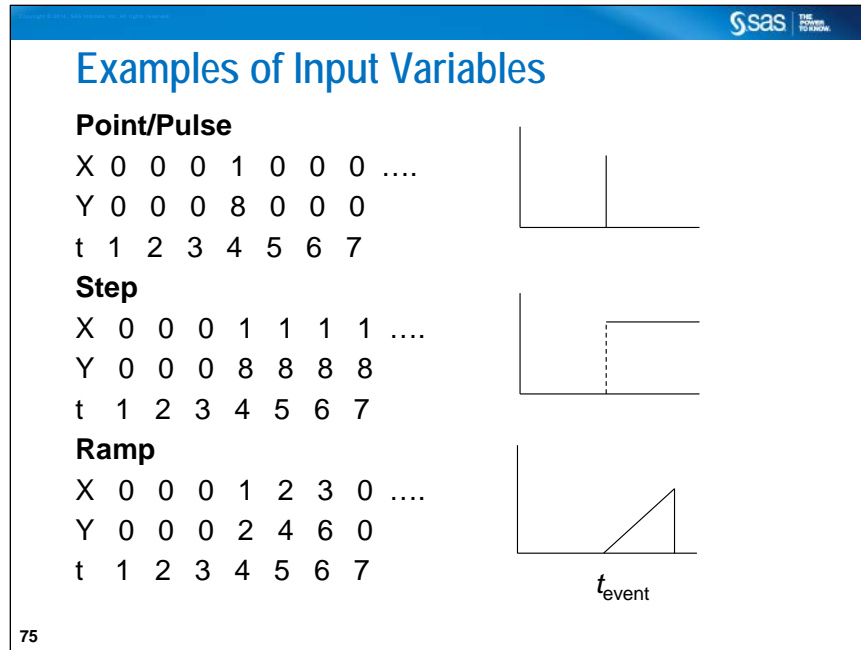
- In retail sales, the term *event* is often used and includes the following:
 - promotional events: discounts, sales, featured displays, and so on
 - advertising events: broadcast, Internet, and print media advertising campaigns, sponsored events, celebrity spokespersons, and so on
- In economics and 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

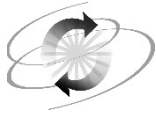
73

Primary Event Variables



74





Exercises

3. Intervention Analysis of the Rose Series

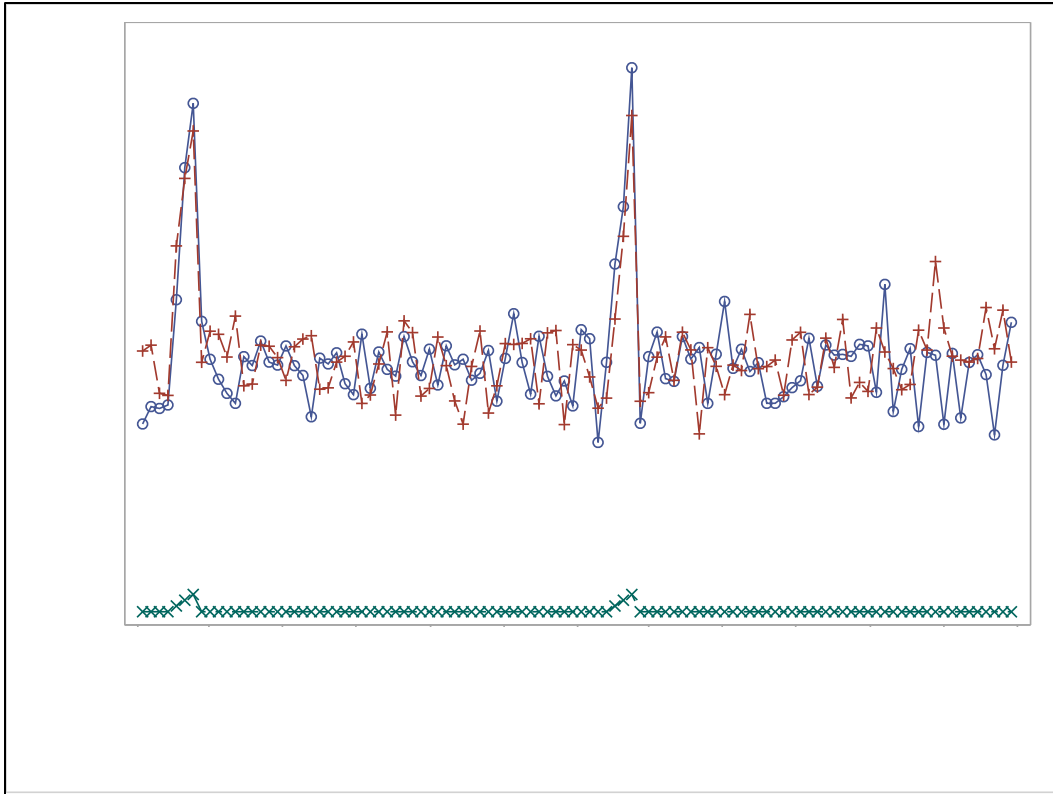
For each rose series (where it is appropriate), use the **Ramp** variable to model the effect of the impending Valentine's Day on weekly rose sales.

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



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

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

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

Forecasting Before You Forecast			
		Quarter	t
Ultimate Goal: Forecast the next four quarters.		4Q2015	Y_{t+4}
		3Q2015	Y_{t+3}
		2Q2015	Y_{t+2}
		1Q2015	Y_{t+1}
How well can you forecast these four most recent observed quarters?		4Q2014	Y_t
		3Q2014	Y_{t-1}
		2Q2014	Y_{t-2}
		1Q2014	Y_{t-3}
Forecasting observed values with the remaining observed series		4Q2013	Y_{t-4}
	
		Holdout Sample	
		Fit Sample	

83

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.

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

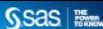


Summary of Data Used for Forecast Model Building

Fit Sample	Holdout Sample
<ul style="list-style-type: none">■ Used to estimate model parameters for accuracy evaluation■ Used to forecast values in holdout sample	<ul style="list-style-type: none">■ 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



Model Fit Statistics

Mean Absolute Percent Error:

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

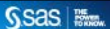
Mean Absolute Error:

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

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

Root Mean Square Error:

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

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

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

89

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



Examples of This Conundrum

To accurately forecast future _____ for the next year, you need to first accurately forecast _____.

- crop yields : rainfall or precipitation amount
- gasoline prices : the price of crude oil per barrel
- solar power generation : cloud cover

Can you accurately forecast rainfall or precipitation, the price of oil per barrel, or cloud cover over the next T time periods?

92

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 .


sas THE POWER OF DATA

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 $t+2$

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.

Forecasting and Accuracy Assessment
 THE POWER OF DATA

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



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.

The screenshot displays the SAS Enterprise Miner interface with the 'Modeling and Forecasting' task selected in the left-hand navigation pane. The main workspace is divided into tabs: DATA, MODEL, OPTIONS, OUTPUT, and INFORMATION. The 'DATA' tab is active, showing the following configuration:

- DATA:** A dropdown menu shows 'STSM.SOLARPV'.
- NOTE:** A message states: 'This task requires data in a valid time series format. To prepare your data, run the Time Series Data Preparation task before starting this task.'
- ROLES:**
 - Dependent variable (1 item):** A dropdown menu shows '123 kW_Gen'.
 - ADDITIONAL ROLES:**
 - Time ID (1 item):** A dropdown menu shows 'EDT'.
 - Properties:**
 - Interval:** A dropdown menu shows 'Week'.
 - Multiplier:** A numeric input field shows '1'.
 - Shift:** A numeric input field shows '1'.
 - Season length:** A numeric input field shows '52'.
 - Group analysis by:** A dropdown menu shows 'Column'.

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.)

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;
```



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

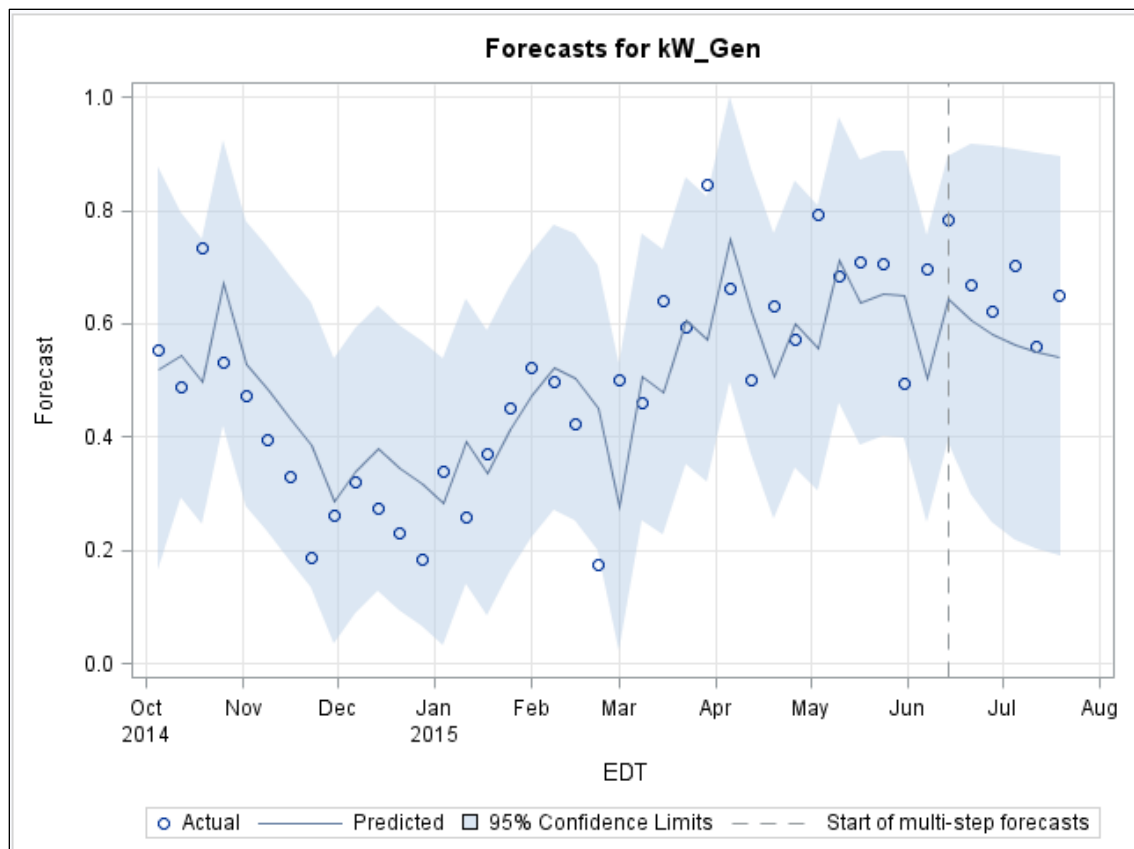
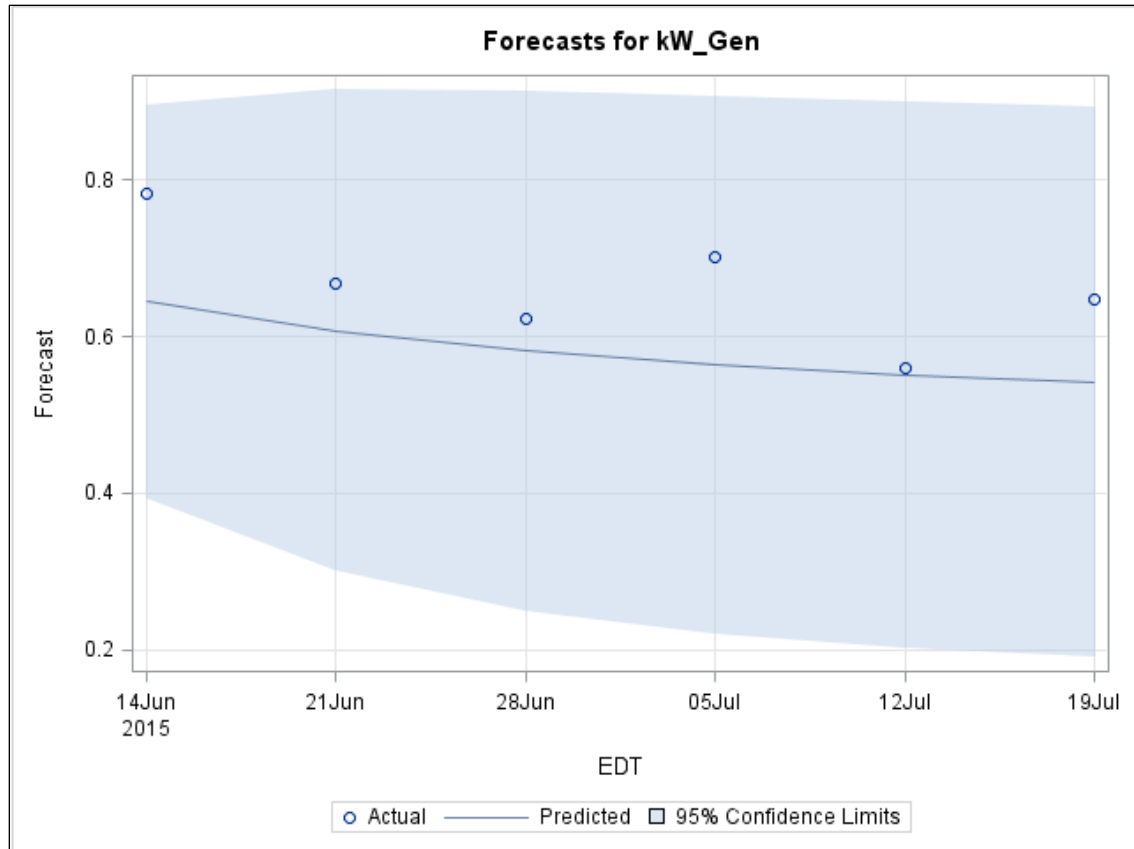
Autoregressive Factors	
Factor 1:	1 - 0.70389 B**(1)

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.



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



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

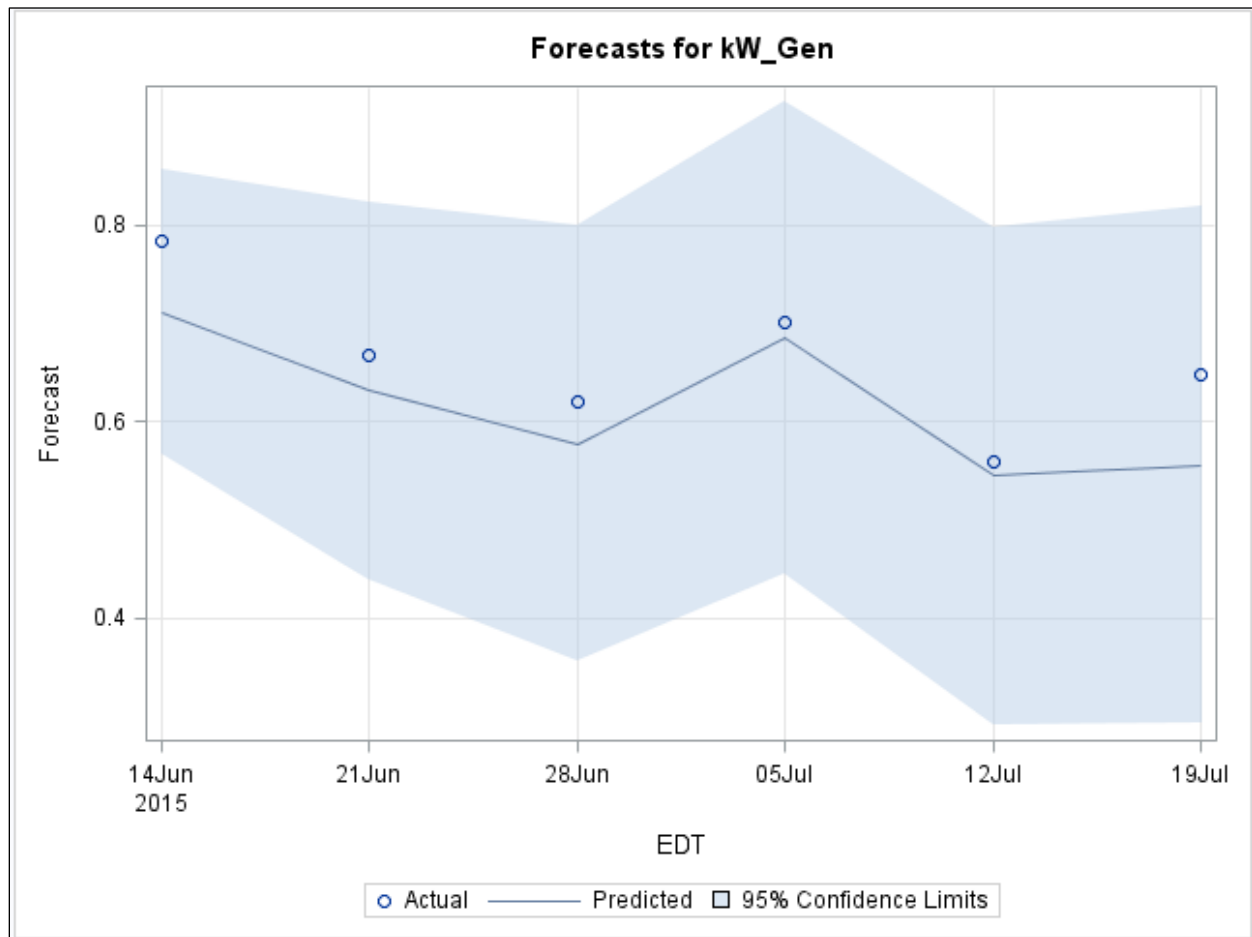
Autoregressive Factors	
Factor 1:	1 - 0.86587 B**(1)

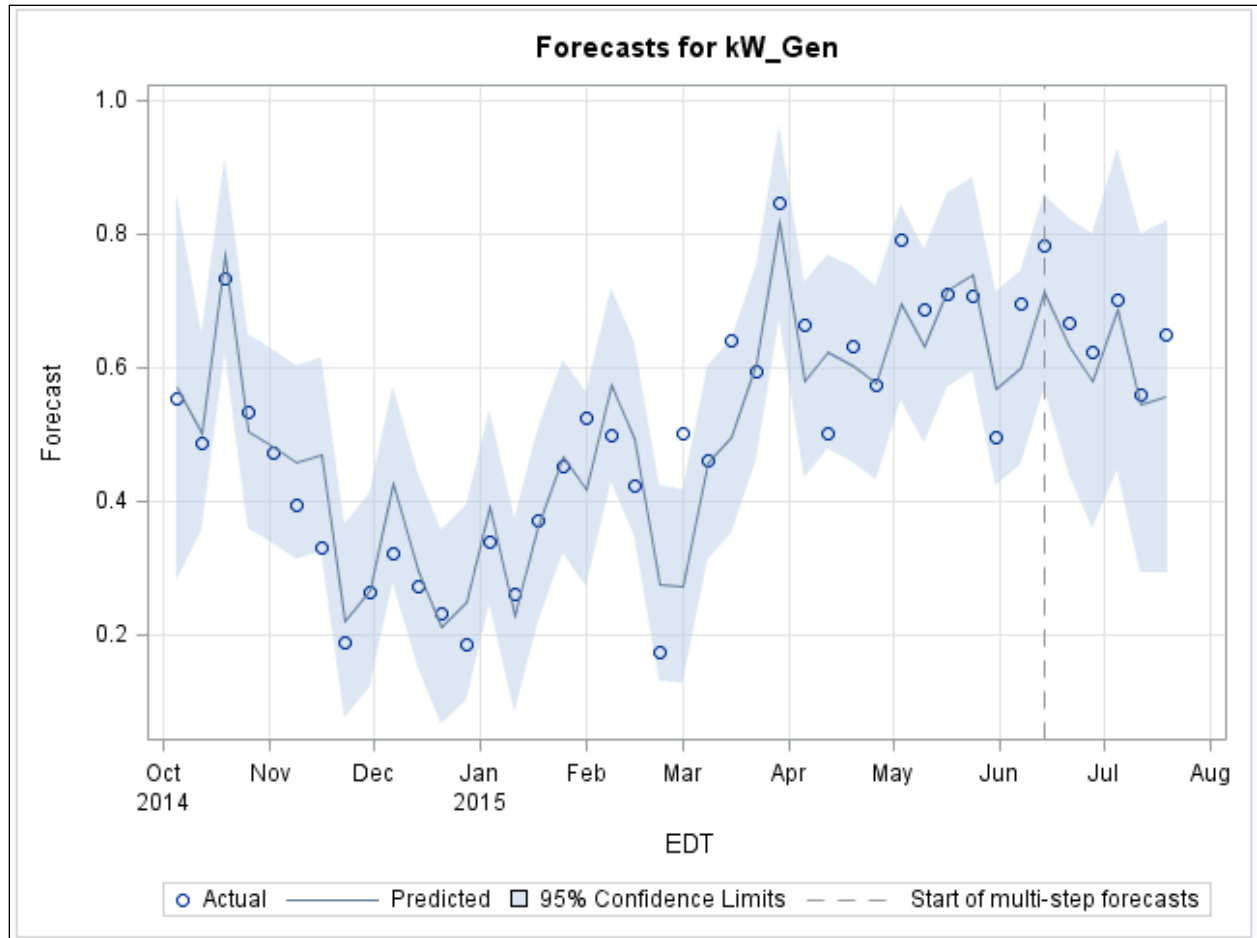
Input Number 1	
Input Variable	Cloud_Cover
Overall Regression Factor	-0.09061

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.





End of Demonstration



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.

Settings Code/Results Split |

DATA MODEL OPTIONS **OUTPUT** INFORMATION

▲ OUTPUT DATA SET

☒ Create output data set

The output data set includes forecast and actual values and confidence limits.

* Data set name:

AR1_forecast

☐ Create parameter estimates data set

☐ Create fit statistics data set

☐ Create covariances and correlations data set

☐ Create model information data set

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.

Settings Code/Results Split |

DATA MODEL OPTIONS **OUTPUT** INFORMATION

▲ OUTPUT DATA SET

☒ Create output data set

The output data set includes forecast and actual values and confidence limits.

* Data set name:

ARMAX1_forecast

☐ Create parameter estimates data set

☐ Create fit statistics data set

☐ Create covariances and correlations data set

☐ Create model information data set



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.;
```

```

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**.

```

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



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;
```



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;
```

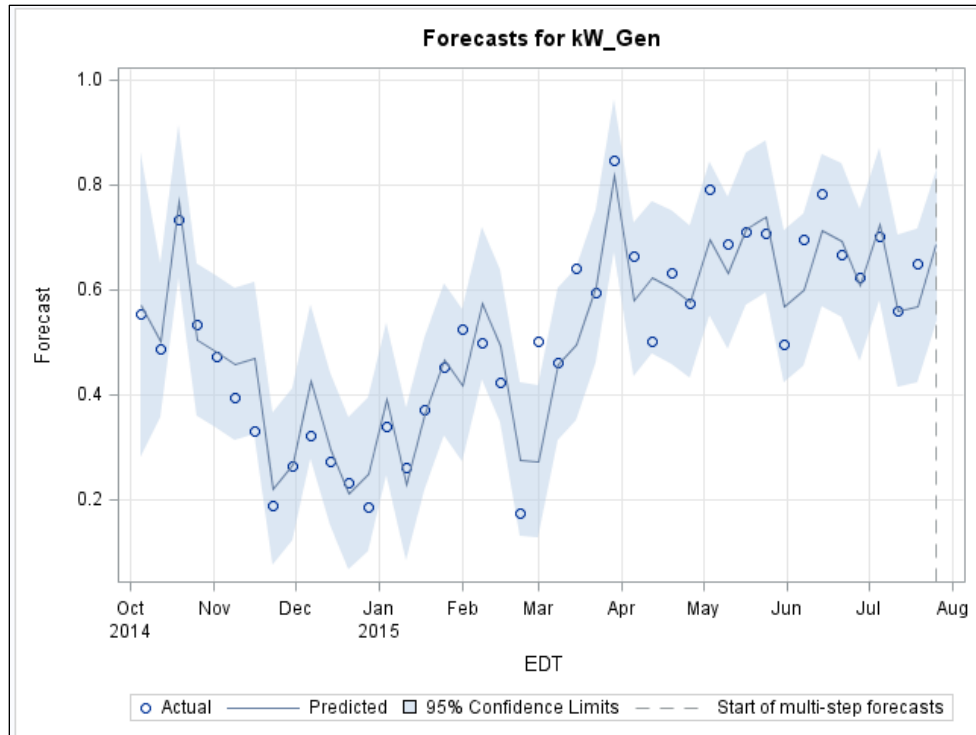
9. Submit the program.

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.

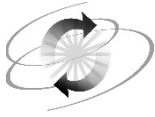
Forecasts for variable kW_Gen					
Obs	Forecast	Std Error	95% Confidence Limits		Actual Residual
...					
43	0.6856	0.0742	0.5402	0.8310	. .

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.



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



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

2.6 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.

