Course Guide

Introduction to Time Series Analysis Using IBM SPSS Modeler (v18.1.1)

Course code 0A028 ERC 2.0

IBM Training

January, 2018

NOTICES

This information was developed for products and services offered in the USA.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service. IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
United States of America

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you. Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

TRADEMARKS

IBM, the IBM logo, ibm.com, TM1®, Cognos®, and DB2® are trademarks of International Business Machines Corp., registered in many jurisdictions worldwide.

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Java and all Java-based trademarks and logos are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Contents

Course overview

Preface overview

This course gets you up and running with a set of procedures for analyzing time series data. Learn how to forecast using a variety of models, including Regression, Exponential Smoothing, and ARIMA, which take into account different combinations of trend and seasonality. The Expert Modeler features will be covered, which is designed to automatically select the best fitting Exponential Smoothing or ARIMA model, but you will also learn how to specify your own custom models, and also how to identify ARIMA models yourself using a variety of diagnostic tools such as sequence charts and autocorrelation plots.

Intended audience

This course is recommended for:

- Business Analyst
- Data Scientist
- Anyone who is interested in getting up to speed quickly and efficiently using the IBM SPSS Modeler forecasting capabilities

Topics covered

Topics covered in this course include:

- Introduction to time series analysis
- Automatic forecasting with the Expert Modeler
- Measuring model performance
- Time series regression
- Exponential smoothing models
- ARIMA models

Course prerequisites

Familiarity with the IBM SPSS Modeler environment (creating, editing, opening, and saving streams).

General knowledge of regression analysis is recommended but not required.

Document conventions

Conventions used in this guide follow Microsoft Windows application standards, where applicable. As well, the following conventions are observed:

- **Bold**: Bold style is used in demonstration and exercise step-by-step solutions to indicate a user interface element that is actively selected or text that must be typed by the participant.
- *Italic*: Used to reference book titles.
- CAPITALIZATION: All file names, table names, column names, and folder names appear in this guide exactly as they appear in the application.
  To keep capitalization consistent with this guide, type text exactly as shown.

Exercises

Exercise format

Exercises are designed to allow you to work according to your own pace. Content contained in an exercise is not fully scripted out to provide an additional challenge. Refer back to demonstrations if you need assistance with a particular task. The exercises are structured as follows:

The business question section

This section presents a business-type question followed by a series of tasks. These tasks provide additional information to help guide you through the exercise. Within each task, there may be numbered questions relating to the task. Complete the tasks by using the skills you learned in the unit. If you need more assistance, you can refer to the Task and Results section for more detailed instruction.

The tasks and results section

This section provides a task based set of instructions that presents the question as a series of numbered tasks to be accomplished. The information in the tasks expands on the business case, providing more details on how to accomplish a task. Screen captures are also provided at the end of some tasks and at the end of the exercise to show the expected results.

Additional training resources

Visit IBM Analytics product training and skills validation on the IBM Skills gateway at http://ibm.com/training/analytics for on:

- Instructor-led training in a classroom or online
- Self-paced training that fits your needs and schedule
- Comprehensive curricula and training paths that help you identify the courses that are right for you
- IBM Analytics Certification program
- Other resources that will enhance your success with IBM Analytics Software

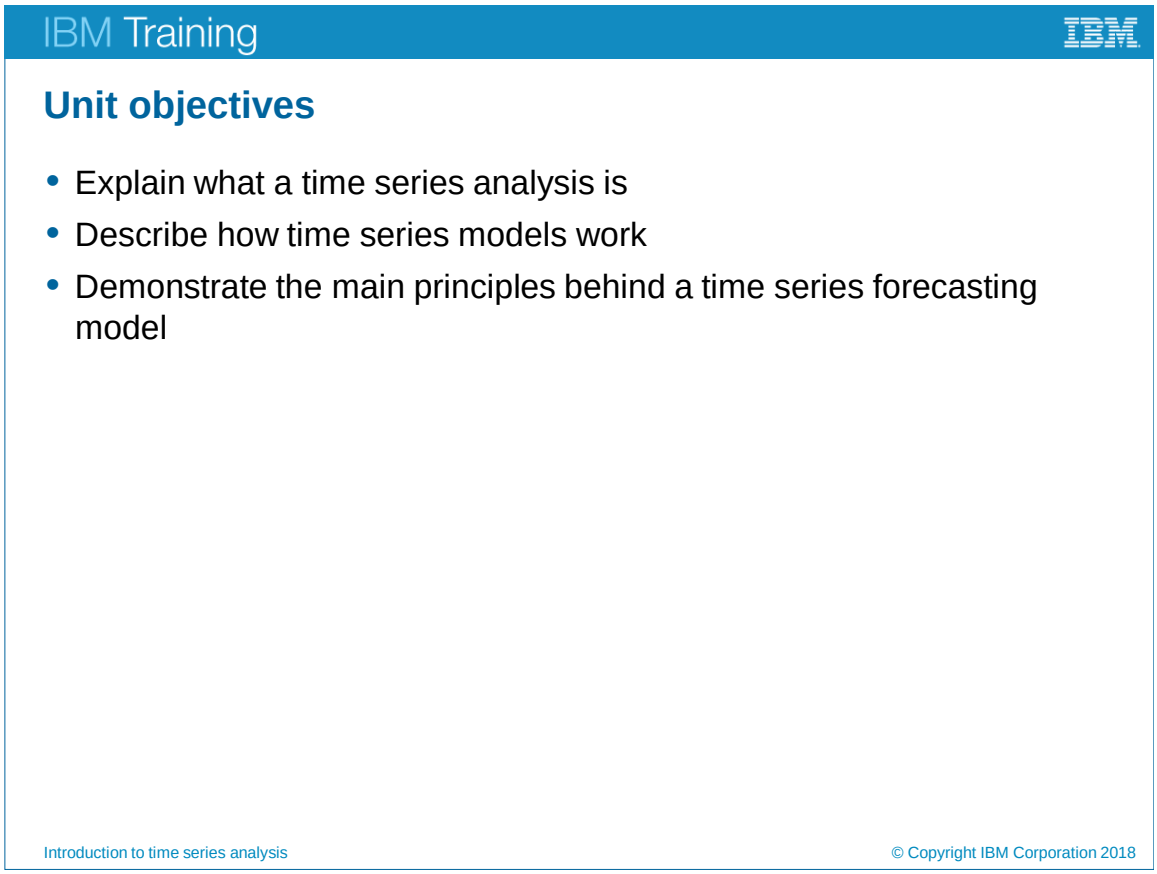
IBM product help

Help type	When to use	Location
Task-oriented	You are working in the product and you need specific task-oriented help.	*IBM Product* - Help link
Books for Printing (.pdf)	You want to use search engines to find information. You can then print out selected pages, a section, or the whole book. Use Step-by-Step online books (.pdf) if you want to know how to complete a task but prefer to read about it in a book. The Step-by-Step online books contain the same information as the online help, but the method of presentation is different.	Start/Programs/*IBM Product*/Documentation
IBM on the Web	You want to access any of the following:	
	• IBM - Training and Skills Validation	• IBM Skills Gateway at http://ibm.com/training/analytics
	• IBM Analytics Support	• https://www.ibm.com/analytics/resources/support
	• IBM Web site	• http://www.ibm.com

IBM Training

Introduction to time series analysis

IBM SPSS Modeler (v18.1.1)



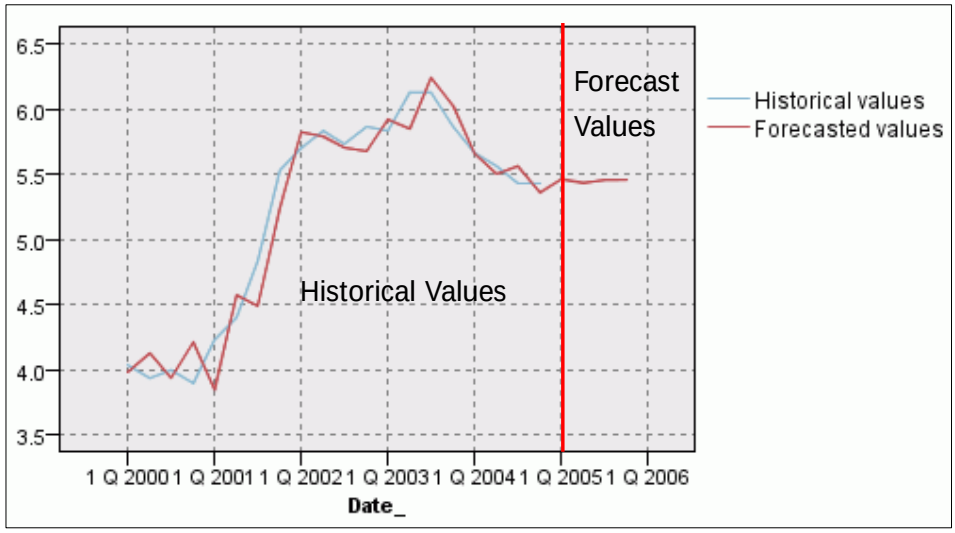
Unit objectives

Before reviewing this unit, you should be familiar with the following topics:

- Working with IBM SPSS Modeler (streams, nodes, palettes)
- Importing data (Var. File node)
- Defining measurement levels, roles, blanks, and instantiating data (Type node)
- Examining the data (Table node)

IBM Training

Forecast future values



Introduction to time series analysis

Forecast future values

It is often essential for organizations to plan ahead. In order to do this it is important to forecast events in order to ensure a smooth transition into the future. In order to minimize errors when planning for the future it is necessary to collect information on any factors which may influence plans on a regular basis over time. Once a catalogue of past and current information has been collected, patterns can be identified and these patterns help make forecasts into the future.

Even though many organizations may collect historic information relevant to the planning process, forecasts are often made on an ad-hoc basis. This often leads to large forecasting errors and costly mistakes in the planning process. Statistical techniques provide a more scientific basis upon which to base forecasts. By using these techniques, a more structured approach can be used to ensure careful planning which will reduce the chance of making costly errors. Statisticians have developed a whole area of statistical techniques, known as time series analysis, which is devoted to the area of forecasting.

In order to understand how time series analysis works it is useful to give an example. Suppose that a company wishes to forecast the growth of its sales into the future. The benefit of making the forecast is that if the company has an idea of future sales it can plan the production process for its product. In doing so, it can minimize the chances of under producing and having product shortages or, alternatively, overproducing and having excess stock which will need to be stored at additional cost.

Prior to being able to make the forecast, the company will need to collect information on its sales over time in order to gain a full picture of how sales have changed in the past. Once this information has been collected it is possible to plot how sales change over time.

One of the most common uses of time series analysis is to forecast future values of a series. There are a number of statistical time series techniques which can be used to make forecasts into the future. In the above example the forecast would be the future values of sales.

Other examples of time series analysis and forecasting include:

- Governments using time series analysis to forecast the effects of government policies on inflation, unemployment and economic growth.
- Traffic authorities analyzing the effect on traffic flows following the introduction of parking restrictions in city centers.
- The analyses of how stock market prices change over time. By being able to forecast when stock market prices rise or fall decisions can be made about the right times to buy and sell shares.
- Companies forecasting the effects of pricing policies or increased advertising expenditure on the sales of their product.
- A company wishing to predict the number of telephone calls at different times during the day, so it can arrange the appropriate level of staffing.

Time series analysis is used in many areas of business, commerce, government and academia, and its value cannot be overstated.

IBM Training

What is a time series?

sales	year_	month_	date_
589	1982	1	JAN 1982
561	1982	2	FEB 1982
640	1982	3	MAR 1982
656	1982	4	APR 1982
727	1982	5	MAY 1982
697	1982	6	JUN 1982
640	1982	7	JUL 1982
599	1982	8	AUG 1982
568	1982	9	SEP 1982
577	1982	10	OCT 1982
553	1982	11	NOV 1982
582	1982	12	DEC 1982

What is a time series?

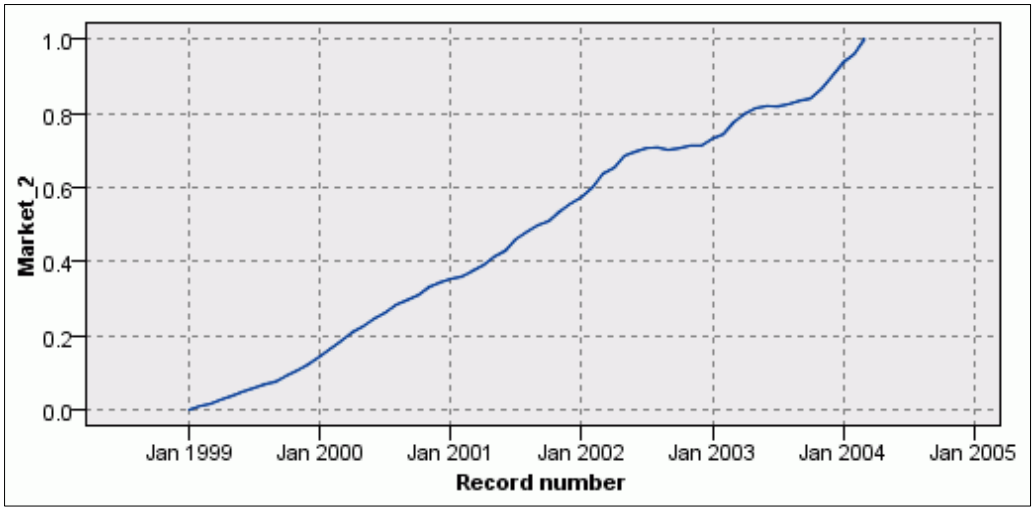
A time series is a field whose values represent equally spaced observations of a phenomenon over time. Examples of time series include quarterly interest rates, monthly unemployment rates, weekly beer sales, annual sales of cigarettes, and so on.

IBM SPSS Modeler expects each row (record) of the dataset to represent a time period, while the columns (fields) contain the time series to be forecast.

Once this information has been collected it is possible to plot how sales change over time. An example of this is shown in figure above. Here information on the sales of a product has been collected each month from January 1982 until December 1995.

IBM Training

Trend component



Introduction to time series analysis

Trend component

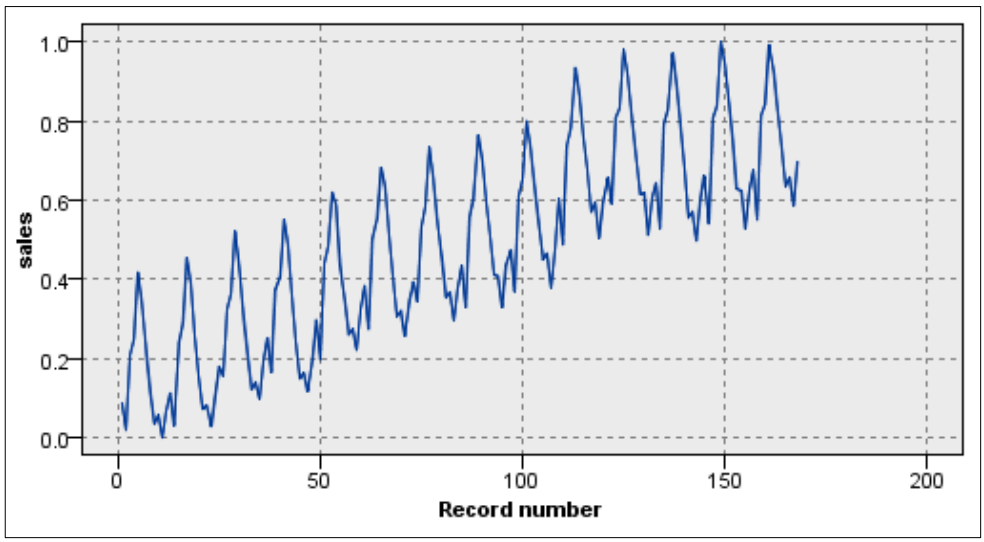
Patterns that occur in the time series can be divided up into three main categories, trend, seasonal components and cycles.

Trend refers to the smooth upward or downward movement characterizing a time series over a long period of time. This type of movement is particularly reflective of the underlying continuity of fundamental demographic and economic phenomena. Trend movements are thought of as long-term movements, usually requiring 15 or 20 years to describe (or the equivalent for series with more frequent time intervals). Trend movements might be attributable to factors such as population change, technological progress, and large-scale shifts in consumer tastes.

For example, the above graphic in Market 2 shows a steady trend upward from 1999 to 2004 in the number of broadband subscriptions.

IBM Training

Seasonal component



Introduction to time series analysis

Seasonal component

Seasonal variations are periodic patterns of movement in a time series. Such variations are considered to be a type of cycle that completes itself within the period of calendar year, and then continues in a repetition of this basic pattern. The graphic of monthly car sales shown above follows a fairly well defined seasonal (here yearly) pattern. Each year there is higher demand for cars in the middle of each year, and lower demand at the beginning and ending.

The major factors in this seasonal pattern are weather and customs, where the latter term is broadly interpreted to include patterns in social behavior as well as observance of various holidays such as Christmas and Easter. Series of monthly or quarterly data is ordinarily used to examine these seasonal patterns. Hence, regardless of trend or cyclical levels, one can observe in the United States that each year more ice cream is sold during the summer months than during the winter, whereas more fuel oil for home heating purposes is consumed in the winter than during the summer months. Both of these cases illustrate the effect of climatic factors in determining seasonal patterns. Also, department store sales generally reveal a minor peak during the months in which Easter occurs and a larger peak in December, when Christmas occurs, reflecting the shopping customs of consumers associated with these dates.

Seasonal patterns need not be linked to a calendar year. For example, if you studied the daily volume of packages delivered by a private delivery service, the periodic

pattern might well repeat weekly (heavier deliveries mid-week, lighter deliveries on the weekend). Here the period for the seasonal pattern could be seven days. Of course, if daily data were collected over several years, then there may well be a yearly pattern as well, and just which time period constitutes a season is no longer clear.

The number of time periods that occur during the completion of a seasonal pattern is referred to as the series periodicity. How often the time series data are collected usually depends on the type of seasonality that the analyst expects to find.

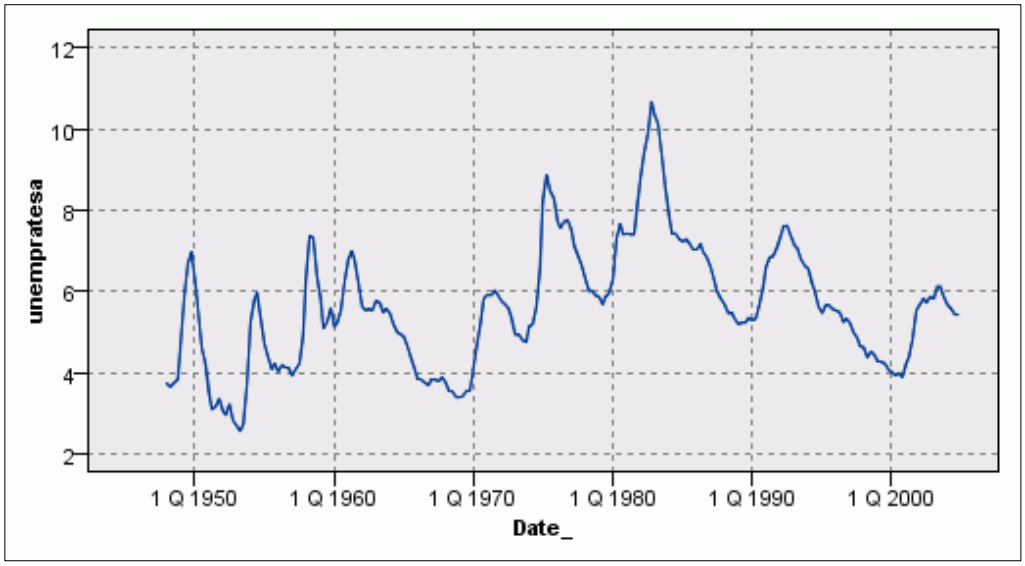
- For hourly data, where data are collected once an hour, there is usually one seasonal pattern every twenty-four hours. The periodicity is most likely to be 24.
- For monthly data, where each month a new time period of data is collected, there is usually one seasonal pattern every twelve months. The periodicity is thus likely to be 12.
- For daily data, where data are collected once every day, there is usually one seasonal pattern per week. The periodicity is therefore 7 if the data refer to a seven-day week or 5 if no data are collected on Saturdays and Sundays.
- For quarterly data, where data are collected once every three months, there is usually one seasonal pattern per year. The periodicity is therefore 4.
- For annual data, where data are collected once a year, there is no seasonal pattern. The periodicity is therefore none (undefined).

Of course, changes can occur in seasonal patterns because of changing institutional and other factors. Hence, a change in the date of the annual automobile show can change the seasonal pattern of automobile sales. Similarly, the advent of refrigeration techniques with the corresponding widespread use of home refrigerators has brought about a change of seasonal pattern of ice cream sales. The techniques of measurement of seasonal variation which we will discuss are particularly well suited to the measurement of relatively stable patterns of seasonal variation, but can be adapted to cases of changing seasonal movements as well.

In the prior graphic, there appears to be a rise in sales during the early part of the year while sales tend to fall to a low around November. Finally, there is some recovery in sales leading up to the Christmas period of each year.

IBM Training

Cyclical component



Introduction to time series analysis

Cyclical component

Cyclical patterns (or fluctuations), or business cycle movements, are recurrent up and down movements around the trend levels which have a duration of anywhere from about 2 to 15 years. The graphic shown above of U.S. unemployment between 1948 and 2003 cycles between high and low unemployment, but the duration and magnitudes are never the same.

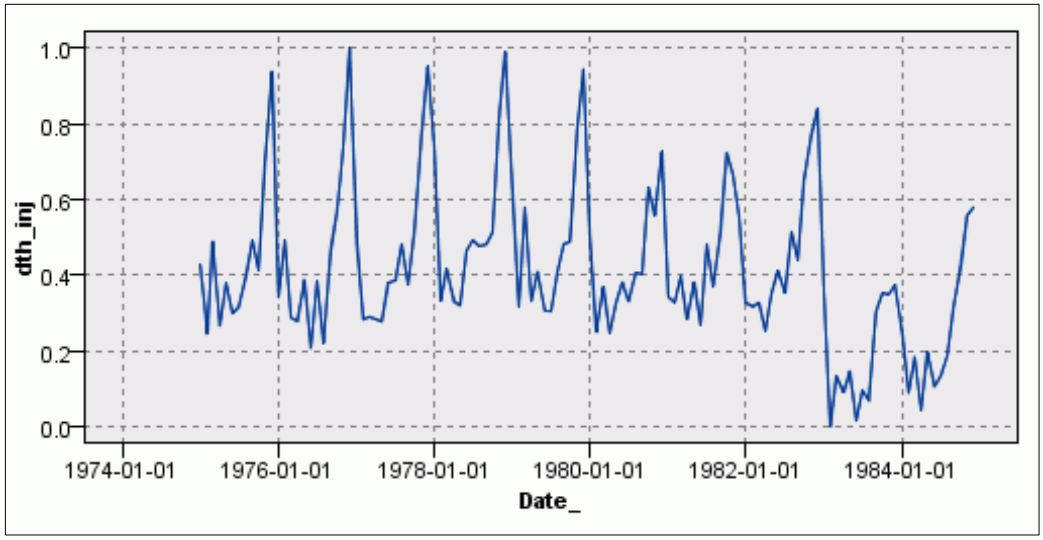
Many people confuse cyclic behavior with seasonal behavior, but they are not the same. If the fluctuations are unchanging and associated with some aspect of the calendar, then the pattern is seasonal. The pattern is cyclic if the fluctuations are not of fixed period. The duration of these cycles can be measured in terms of their turning points, or in other words, from trough to trough or peak to peak. These cycles are recurrent rather than strictly periodic. The height and length (amplitude and duration) of cyclical fluctuations in industrial series differ from those of agricultural series, and there are differences within these categories and within individual series. Hence, cycles in durable goods activity generally display greater relative fluctuations than consumer goods activity and a particular time series of, say, consumer goods activity may possess business cycles which have considerable variations in both duration and amplitude.

Economists have produced a large number of explanations of business cycle fluctuations including external theories which seek the causes outside the economic system, and internal theories in terms of factors within the economic system that lead to self-generating cycles. In theories of the internal type, periods of contracted business activity are viewed as containing within themselves the keys for the following period of expansion, which then contain the keys for the next period of contraction. In this connection, the terminology generally used is that of the late Wesley Mitchell of the National Bureau of Economic Research who distinguished various phases of the cycle. The period of expansion ends at the *peak*, or upper turning point, and moves into contraction which terminates at the lower turning point, the trough, or revival. Then these phases repeat themselves, however, with different duration and amplitude. Most economists admit that there are both internal and external components in any comprehensive explanation of business cycle activity.

Since it is clear from the foregoing discussion that there is no single simple explanation of business cycle activity and that there are different types of cycles of varying length and size, it is not surprising that no highly accurate method of forecasting this type of activity has been devised. Indeed, no generally satisfactory mathematical model has been constructed for either describing or forecasting these cycles, and perhaps never will be. Therefore, it is not surprising to find that classical time series analysis adopts a relatively rough approach to the statistical measurement of the business cycle. The approach is a residual one; that is, after trend and seasonal variations have been eliminated from a time series, by definition, the remainder or residual is treated as being attributable to cyclical and irregular factors. Since the irregular movements are by their very nature erratic and not particularly tractable to statistical analysis, no explicit attempt is usually made to separate them from cyclical movements, or vice versa. However, the cyclical fluctuations are generally large relative to these irregular movements and ordinarily no particular difficulty in description or analysis arises from this source. Therefore, unless you have data available over a long period of time, cyclic patterns are not usually fit by forecasting models.

IBM Training

Irregular movements

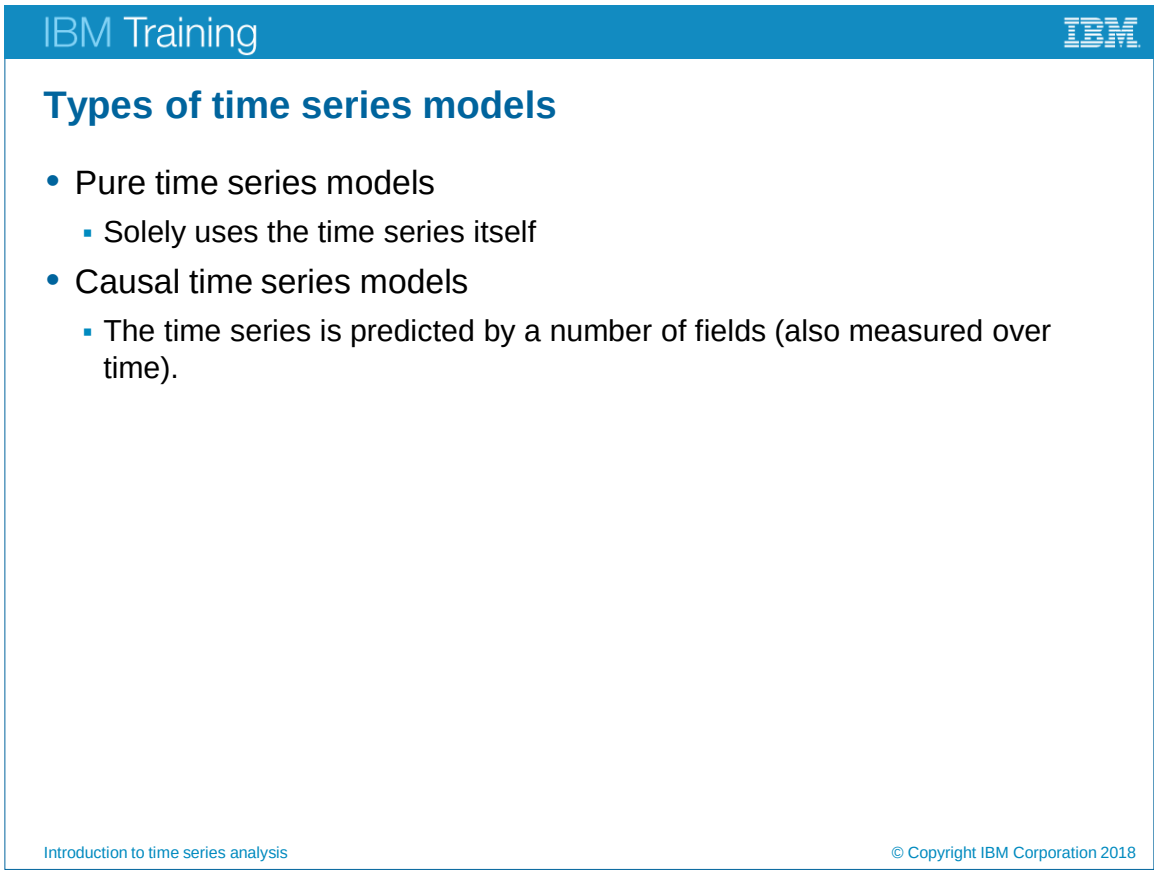


Introduction to time series analysis

Irregular movements

Irregular movements are fluctuations in time series that are erratic in nature, and follow no regularly recurrent or other discernible pattern. These movements are sometimes referred to as residual variations, since, by definition, they represent what is left over in an economic time series after trend, seasonal, and cyclical elements have been accounted for. These irregular fluctuations result from sporadic, unsystematic occurrences such as wars, earthquakes, accidents, strikes, and the like. In the above graphic of the number of fertility rate for U.S. women, 1917-2001, there is a marked drop during the 1930s which coincided with the Great Depression, and a major rise in the rate following World War II, but there is no regular pattern in the graph.

In the classical time series model, the elements of trend, cyclical, and seasonal variations are viewed as resulting from systematic influences leading to gradual growth, decline, or recurrent movements. Irregular movements, however, are considered to be so erratic that it would be fruitless to attempt to describe them in terms of a formal model. Irregular movements can result from a large number of causes of widely differing impact.



Types of time series models

A distinction can be made between pure and causal time series models.

- Pure time series models. Pure time series models utilize information solely from the series itself. In other words, pure time series forecasting makes no attempt to discover the factors affecting the behavior of a series. For example, if the aim were to forecast future sales for a product, then a pure time series model would use just the data collected on sales. Information on other explanatory forces such as advertising expenditure and economic conditions would not be used when developing a pure time series model.

  In pure time series models it is assumed that some pattern or combination of patterns in the series which is to be forecasted is recurring over time. Identifying and extrapolating that pattern can develop forecasts for subsequent time periods.

  The main advantage of pure time series modeling is that it is a quick and simple way of developing a forecast model. Also, such models rely upon little statistical theory.

  One obvious disadvantage of pure time series models is that they cannot identify important factors influencing the series. Another drawback is that it is not possible to accurately predict the impact of any decisions taken by an organization on the future values of the series.

The exponential smoothing model is an example of a pure time series method, presented later in this course. Exponential smoothing can be performed in the Time Series node.

- Causal time series models. In causal time series models, a relationship is modeled between a dependent field (the time series being predicted), time, and a set of independent fields (other associated factors also measured over time). The first task of forecasting is to find the cause-and-effect relationship. Continuing with the sales example, a causal time series technique such as regression would indicate whether advertising expenditure or the price of the product has been an important influence on sales and if it has, whether each factor has had a positive or negative influence on sales.

  The real advantage of an explanatory model is that a range of forecasts corresponding to a range of values for the different fields can be developed. For example, causal time series models can assess what the effect of a $100,000 increase in advertising expenditure will have on future sales, or alternatively a $150,000 increase in advertising expenditure.

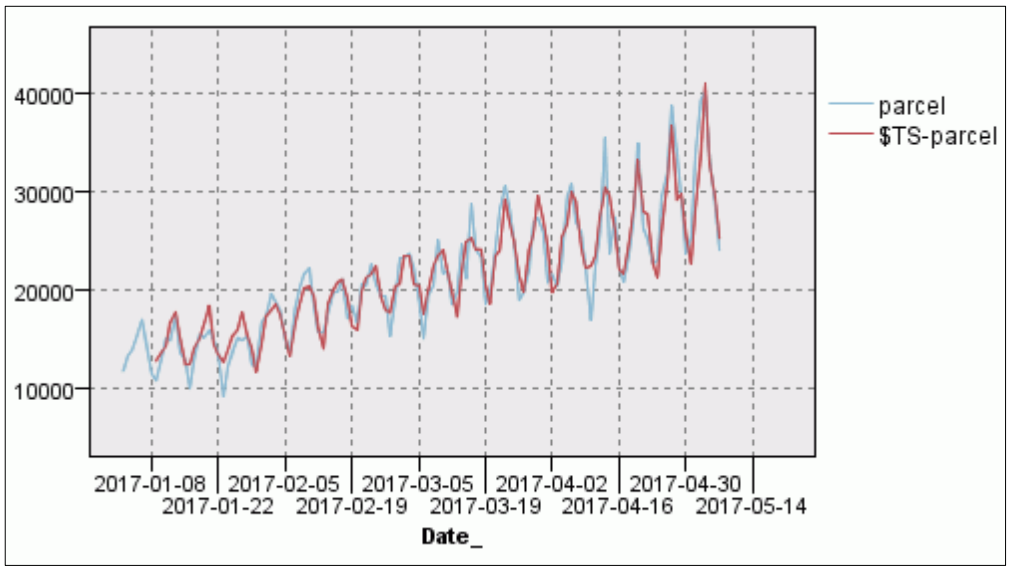
  The main drawbacks of causal time series models are that they require information on several fields in addition to the field that is being forecast and usually take longer to develop. Furthermore, the model may require estimation of the future values of the independent factors before the dependent field can be forecast.

  Regression is an example of a causal time series model. It has, though, a number of drawbacks when it is applied to time series, as will be discussed later in this course. Regression can be performed using the Regression or Linear node. For time series, the Regression node is preferred because it gives more diagnostic statistics.

  ARIMA, presented later in this course, is another example of a causal time series model. ARIMA also lets you model the time series alone, without including predictors, so is a pure time series method too. ARIMA is available in the Time Series node.

IBM Training

Exponential Smoothing



Introduction to time series analysis

Exponential Smoothing

The expert modeler in IBM SPSS Forecasting considers two classes of time series models when searching for the best forecasting model for your data: exponential smoothing and ARIMA. In this section you will be introduced to simple exponential smoothing. You will create models using exponential smoothing in a topics which will provide more detailed coverage of exponential smoothing models.

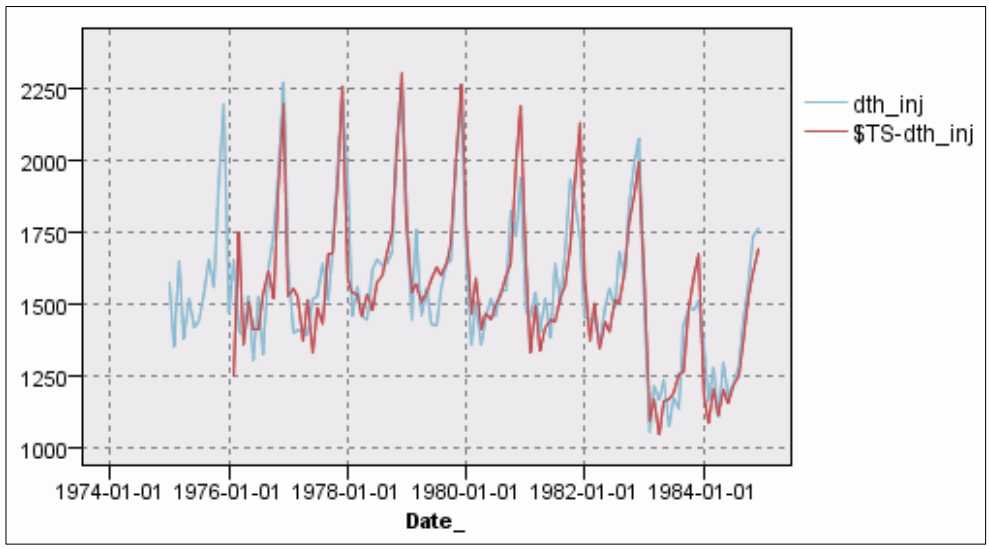
Exponential smoothing is a time series technique that can be a relatively quick way of developing forecasts. This technique is a pure time series method; this means that the technique is suitable when data has only been collected for the series that you wish to forecast. In comparison, ARIMA models can accommodate predictor variables and intervention effects.

Exponential smoothing takes the approach that recent observations should have relatively more weight in forecasting than distant observations. “Smoothing” implies predicting an observation by a weighted combination of the previous values. “Exponential” smoothing implies that the weights decrease exponentially as the observations get older. “Simple” (as in simple exponential smoothing) implies that slowly changing level is all that is being modeled. Exponential smoothing can be extended to model different combinations of trend and seasonality. Exponential smoothing implements many models in this fashion.

An analyst using custom exponential smoothing typically examines the series to make some broad characterizations (is there trend, and if so what type? Is there seasonality [a repeating pattern], and if so what type?) and fits one or more models. The best model fit is then extrapolated into the future to make forecasts. One of the main advantages of exponential smoothing is that models can be easily constructed. The type of exponential smoothing model developed will depend upon the seasonal and trend patterns inherent in the series you wish to forecast. An analyst building a model might simply observe the patterns in a time plot to decide which type of exponential smoothing model is the most promising one to generate forecasts. In IBM SPSS Forecasting, when the Expert Modeler examines the series, it considers all appropriate exponential smoothing models when searching for the most promising time series model.

IBM Training

ARIMA



Introduction to time series analysis

ARIMA

Many of the ideas that have been incorporated into ARIMA models were developed in the 1970s by George Box and Gwilym Jenkins (see Box, Jenkins and Reinsel, 1994), and for this reason ARIMA modeling is sometimes called Box-Jenkins modeling. ARIMA stands for AutoRegressive Integrated Moving Average, and the assumption of these models is that the variation accounted for in the series variable can be divided into three components:

- Autoregressive (AR)
- Integrated (I) or Difference
- Moving Average (MA)

An ARIMA model can have any component, or combination of components, at both the nonseasonal and seasonal levels. There are many different types of ARIMA models and the general form of an ARIMA model is ARIMA(p,d,q)(P,D,Q), where:

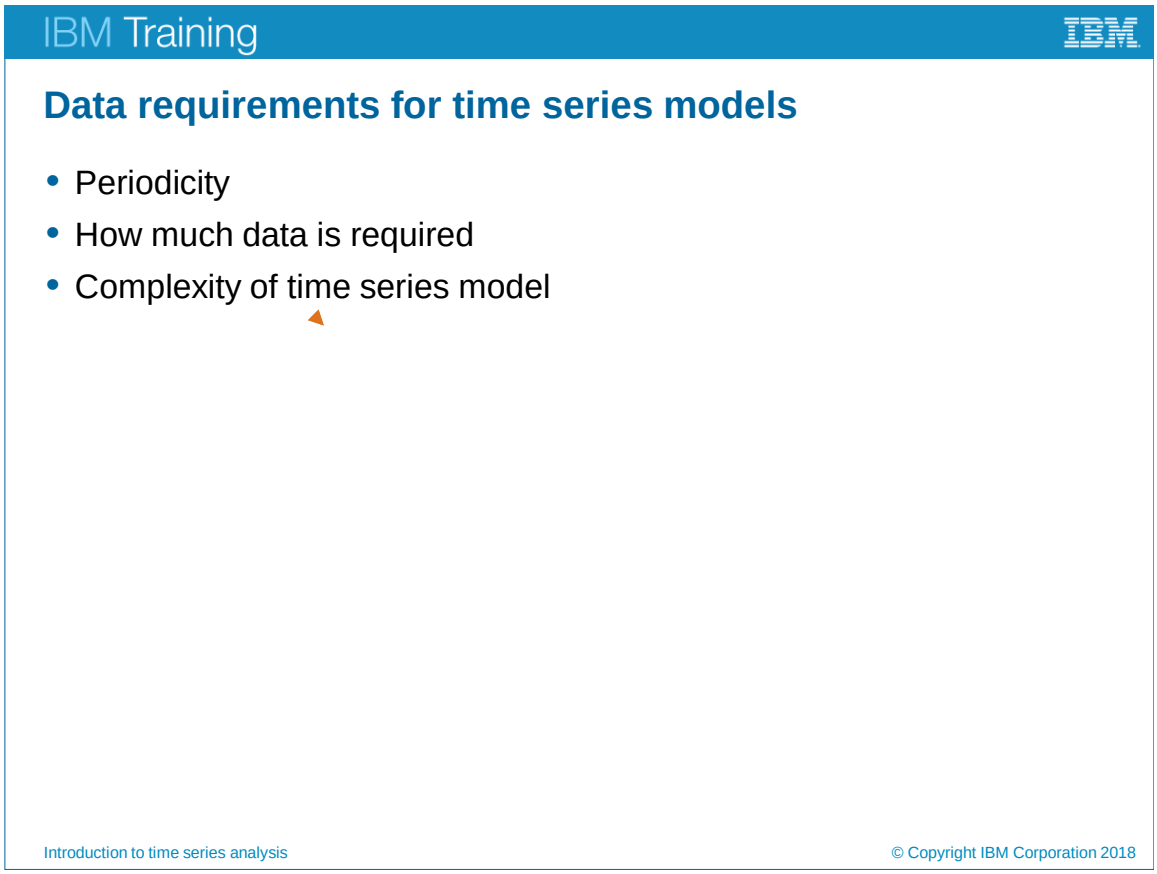
- p refers to the order of the nonseasonal autoregressive process incorporated into the ARIMA model (and P the order of the seasonal autoregressive process)
- d refers to the order of nonseasonal integration or differencing (and D the order of the seasonal integration or differencing)

- q refers to the order of the nonseasonal moving average process incorporated in the model (and Q the order of the seasonal moving average process).

So for example an ARIMA(2,1,1) would be a nonseasonal ARIMA model where the order of the autoregressive component is 2, the order of integration or differencing is 1, and the order of the moving average component is also 1. ARIMA models need not have all three components. For example, an ARIMA(1,0,0) has an autoregressive component of order 1 but no difference or moving average component. Similarly, an ARIMA(0,0,2) has only a moving average component of order 2.

ARIMA also permits a series to be predicted from values in other data series. The relations may be at the same time point (for example, a company spending on advertising this month influences the company's sales this month) or in a leading or lagging fashion (for example, the company's spending on advertising two months ago influence the company's sales this month). Multiple predictor series can be included at different time lags.

Complex ARIMA models that include other predictor series, autoregressive, moving average, and integration components can be built in IBM SPSS Modeler and are considered by the Expert Modeler.



Data requirements for time series models

In time series analysis, each time period at which the data are collected yields one sample point to the time series. The idea is that the more sample points you have, the clearer the picture of how the series behaves. It is not reasonable to collect just two months of data on the sales of a product and, on the basis of this, expect to be able to forecast two years into the future. This is because your sample size is only two (one sixth of the seasonal span) and you wish to forecast 24 data points, or months, ahead (two full seasonal spans). Therefore the way to view the collection of time series information is that the more data points you have, the greater your understanding of the past will be, and the more information you have to use to predict future values in the series.

The first important question to be answered is how many data points are required before it is possible to develop time series forecasts. Unfortunately, there is no clear-cut answer to this, but the following factors influence the minimum amount of data required:

- Periodicity
- How much data is required
- Complexity of the time series model

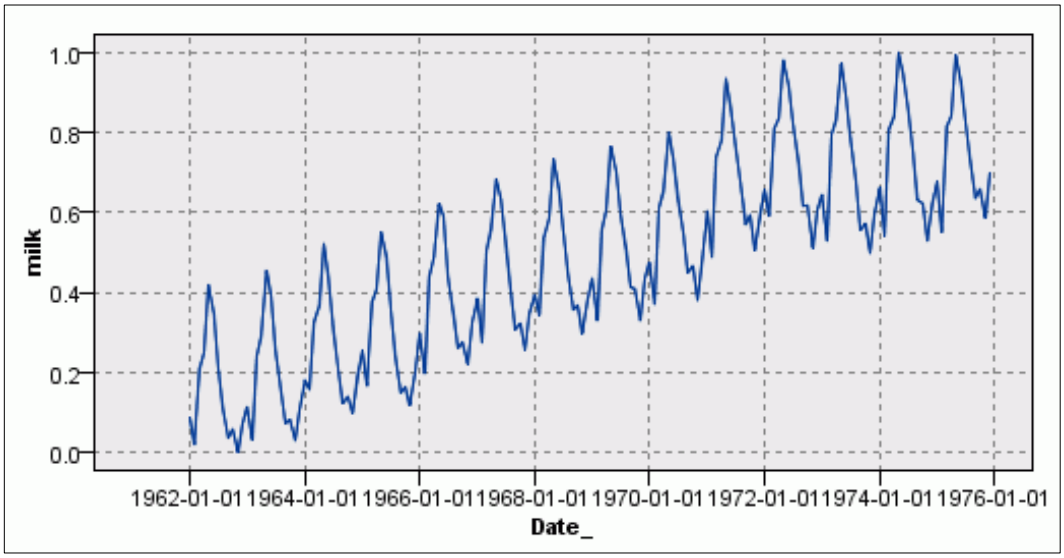
It is important to note that some time series techniques incorporating seasonal effects require several seasonal spans of time series data before it is possible to use them.

Usually four or more seasons of data observations is a good rule of thumb to use when attempting to explore seasonal modeling. For example, four years (seasonal spans) worth of quarterly or monthly data would be sufficient, as there are four replications of the time period. At the same time, four years of annual data is not enough historic data, as the sample is only four. The four year rule is not, however, a rigid rule, as time series can be developed and used for forecasting with less historic data.

Two final thoughts: first, the more complex the time series model, the larger the time series sample size should be. Secondly, time series models assume that the same patterns appear throughout the series. If you are fitting a long series in which a dramatic change occurred that might influence the fundamental relations that exist over time (for example, deregulation in the airline and telecom industries), you may obtain more accurate prediction using only the recent (after the change) data to develop the forecasts.

IBM Training

Periodicity



Introduction to time series analysis

Periodicity

The first important question to be answered concerns the periodicity of the time series. Periodicity is the number of periods that occur during the completion of a seasonal pattern is referred to as the series periodicity. In the above graphic, the monthly data seems to follow the same pattern each year.

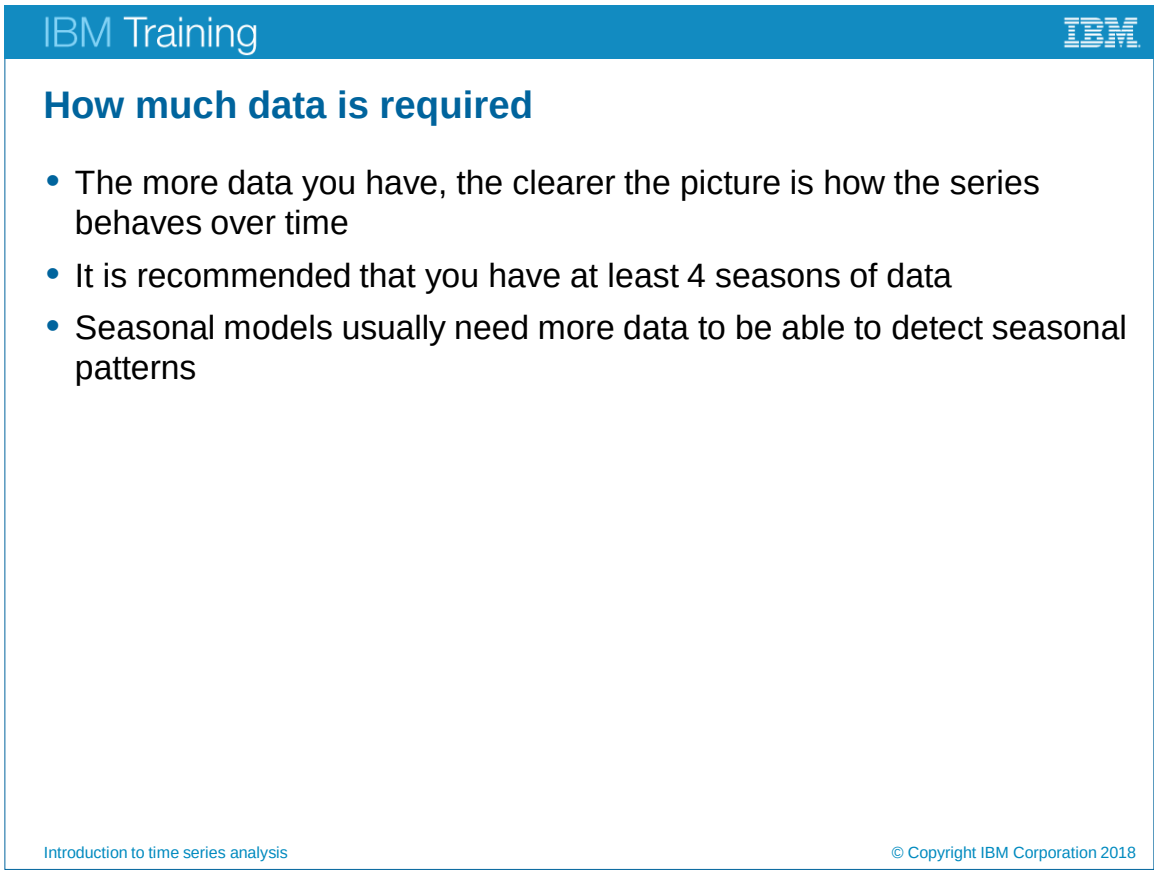
How often the time series data are collected usually depends on the type of seasonality that the researcher expects to find.

- For hourly data, where data is collected once an hour, there is usually one seasonal pattern every twenty-four hours. The periodicity is most likely to be 24.
- For monthly data, where each month a new time period of data is collected, there is usually one seasonal pattern once every twelve months. The periodicity is thus likely to be 12.
- For daily data, where data is collected once every day, there is usually one seasonal pattern per week. The periodicity is therefore 7 if the data refers to a seven-day week or 5 if, say, no data are collected on Saturdays and Sundays.
- For quarterly data, where data are collected once every three months, there is usually one seasonal pattern once every year. The periodicity is therefore 4.

- For annual data, when data are collected once a year, there is no seasonal pattern. The periodicity is therefore none (undefined).

The appropriate time period over which data should be collected will usually depend upon what is being measured. For example, stock market prices change frequently over short periods of time. Therefore, stock market data might need to be collected as often as daily, and maybe hourly or on a minute by minute basis, in order to give a satisfactory measure of movements in stock market prices. However, the series which you wish to forecast will probably not fluctuate over such short time periods. Also, you may only need forecasts at a certain level of granularity. For example, a public company may be very interested in predicting quarterly sales accurately, but not daily, weekly or even monthly values. Thus, information on the sales of a product, for example, may only need to be collected on a weekly, monthly or quarterly basis.

If you are unsure about the right time period to use in collecting the data for the series you wish to forecast, then ask yourselves the following question: Are there any seasonal patterns in the data? There is little point in only collecting data once a year on the sales of your product if you know what you wish to forecast is monthly sales, and the sales of the product are typically higher in December than in August. For time series analysis to detect monthly patterns in sales it is essential that data be collected on a monthly (or more often) basis.



How much data is required

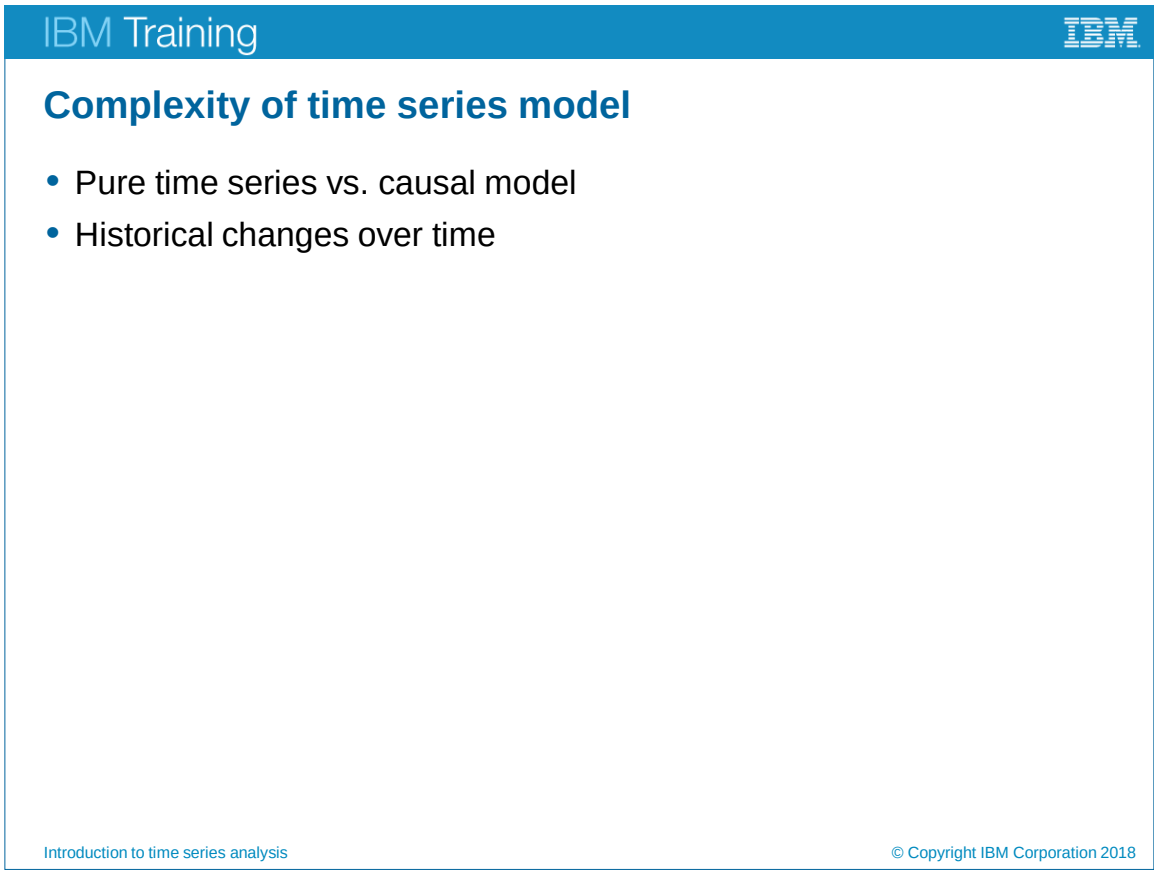
It is important to outline the basic data requirements for time series analysis. The way to view the collection of time series data is in some ways similar to other statistical studies. For example, in survey studies it is often important to collect a large sample in order to have some confidence that your statistical results accurately represent the population you are studying. In time series analysis, each time period at which the data are collected gives one sample point to the time series. The idea is that the more sample points you have, the clearer the picture of how the series behaves. It is not possible, say, to collect just two months of data on the sales of a product and then expect to be able to forecast two years into the future. This is because your sample size is only two (one sixth of the seasonal span) and you wish to forecast 24 data points or months ahead (two full seasonal spans). Therefore the way to view the collection of time series information is that the more data points you have, the greater your understanding of the past will be, and the more information you have to use to predict future values in the series.

It is important to note that some time series techniques incorporating seasonal effects require several seasonal spans of time series data before it is possible to use them. Usually four seasons of data observations or more are a good rule of thumb to use in dealing with this type of analysis. For example, four years (seasonal spans) worth of quarterly or monthly data would be sufficient, as there are four replications of the time

period. At the same time four years of annual data is not enough historic data, as the sample size is only four. The four year rule is not however a rigid rule, as some time series models can be developed and used for forecasting with less historic data, and we will examine such models in this course.

The appropriate time period over which data should be collected will usually depend upon what is being measured. For example, stock market prices change frequently over short periods of time. Therefore, stock market data might need to be collected as often as daily, and maybe hourly or on a minute by minute basis, in order to give a satisfactory measure of movements in stock market prices. However, the series which you wish to forecast will probably not fluctuate over such short time periods. Also, you may only need forecasts at a certain level of granularity. For example, a public company may be very interested in predicting quarterly sales accurately, but not daily, weekly or even monthly values. Thus, information on the sales of a product, for example, may only need to be collected on a weekly, monthly or quarterly basis.

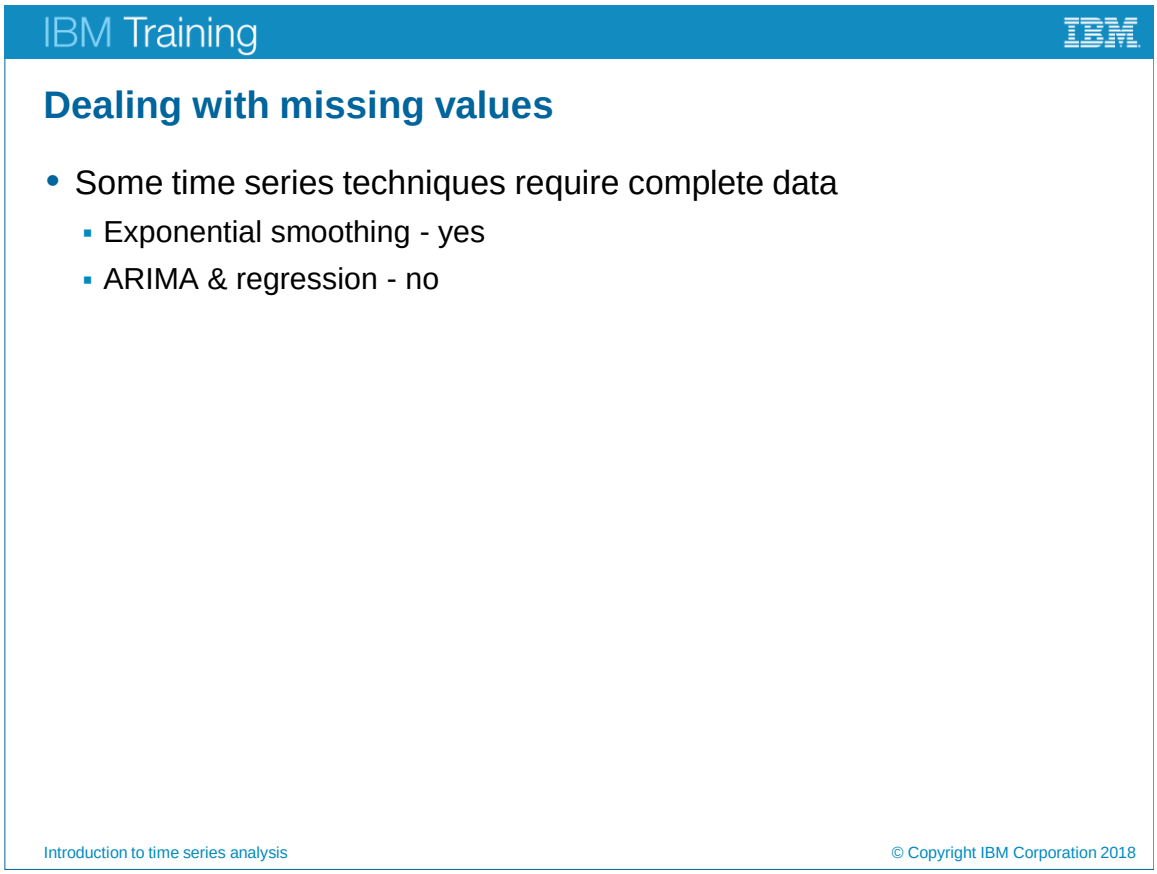
If you are unsure about the right time period to use in collecting the data for the series you wish to forecast, then ask yourselves the following question: Are there any seasonal patterns in the data? There is little point in only collecting data once a year on the sales of your product if you know what you wish to forecast is monthly sales, and the sales of the product are typically higher in December than in August. For time series analysis to detect monthly patterns in sales it is essential that data be collected on a monthly (or more often) basis.



Complexity of time series model

Pure series models requires just the series itself to create a model. They make no attempt to discover external factors that affect the series. For instance, to create a time series model for forecasting future sales, all you would need is the total sales, each day, week, month, and so on. However, if you wanted to also identify the factors that affect sales, such as advertising expenditures, you would need to create an additional time series for advertising as well. This would add complexity to the model because you need to create a separate time series for each predictor you want to include in the model.

Time series models assume that the same patterns appear throughout the series. If you are fitting a long series in which a dramatic change occurred that might influence the fundamental relations that exist over time (for example, deregulation in the airline and telecom industries), you may obtain more accurate prediction using only the recent (after the change) data to develop the forecasts. You might also consider adding an intervention variable to the model as well.



Dealing with missing values

Data collection is often full of difficulties and data may not be recorded for every time period of interest. Missing data at the ends of a time series pose no particular problem apart from shortening the length of the series. However, some time series routines will not function if there are missing data at points other than at the beginning or end of the series (in other words, even if there are cases defined for those dates, the lack of a valid value causes difficulty). Missing data embedded in a time series can therefore be a serious problem for certain time series routines. In order to overcome this problem it is often necessary to “splice” the series. Basically, splicing is a method used to replace missing data with numeric information from elsewhere in the time series. However, note that the ARIMA method in the Expert Modeler and the Regression procedure do not necessarily require complete data for a time series.

In order to overcome the problem of missing values, IBM SPSS Modeler has an option for replacing missing values that are specifically designed for splicing. For example, suppose that for the first two weeks or your study, data were not collected on Mondays. In this instance it is therefore necessary to replace the missing values with information collected on other dates. Unlike in many standard statistical analyses (for example, with survey data), estimating missing data is common in time series analysis.

Within the Missing Value Handling dialog box there are many options for replacing missing values with combinations of non-missing values. The Method drop-down list contains options to replace missing values using one of several time-series functions.

- Series mean. Replaces missing values in the series with the overall mean of the series.
- Mean of nearby points. One of the two Span of nearby points options must be selected. If the Number option is selected, the specified number of valid values above and below the missing data points will be used to compute a mean value to replace the missing value (the default is 2). Alternatively, the All option can be specified, and this is equivalent to the series-mean option
- Median of nearby points. The same as the Mean of nearby points option, except the median is used to replace the missing value.
- Linear interpolation. Useful for replacing missing values when there is more than one consecutive missing value in the series. The last valid value before, and the first valid value after the missing value provide the basis for the linear interpolation. This is useful if you believe there is local trend in the series.
- Linear trend at point. Replaces missing values by running regression on all of the valid data, where the dependent variable is the series itself and the independent variable is time. The regression equation is developed from the observed data and is then used to fit a value for missing observations.

IBM Training

Demonstration 1

Create a time plot of the data

Demonstration 1: Create a time plot of the data

Demonstration 1: Create a time plot of the data

Purpose:
You have collected data on the number of cars sold each month over a fourteen-year period, starting at January 1982. Before deciding on the appropriate time series technique you will use to model the data, you would like to create a time plot of the data in order to identify patterns in the data

Data file: **salescycle.csv**

Data folder: **C:\Training\0A028**

Task 1. Start and configure IBM SPSS Modeler 18.1.1.

The instruction to start IBM SPSS Modeler 18.1.1 will depend on the operating system. The following instruction pertains to Microsoft Windows 10.

1. From the **Start** menu, expand **IBM SPSS Modeler 18.1**, and then click **IBM SPSS Modeler 18.1**.

   Note: IBM SPSS Modeler 18.1.1 displays "IBM SPSS Modeler 18.1" in the Start menu and program name.
2. Click **Cancel** to close the **Welcome** dialog box.

   It is useful to set the working folder for IBM SPSS Modeler.
3. From the **File** menu, click **Set Directory**.
4. Beside **Look in**, navigate to the **C:\Training\0A028** folder, and then click **Set**.

Task 2. Take an initial look at the data.

1. From the **Sources** palette, add a **Var. File** node to the canvas.
2. Edit the **Var. File** node, and then select **salescycle.csv.**

3. Click **Preview**.

   The results appear as follows:

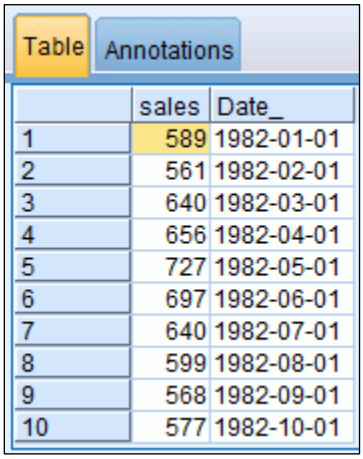


Table | Annotations

	sales	Date_
1	589	1982-01-01
2	561	1982-02-01
3	640	1982-03-01
4	656	1982-04-01
5	727	1982-05-01
6	697	1982-06-01
7	640	1982-07-01
8	599	1982-08-01
9	568	1982-09-01
10	577	1982-10-01

   The dataset contains the recorded monthly sales of a product over a fourteen year period, starting at January 1982.

   The important point to note concerning the organization of time series data is that each row (case) in the data corresponds to a particular period of time. Each row in the data must therefore represent a sequential time period.

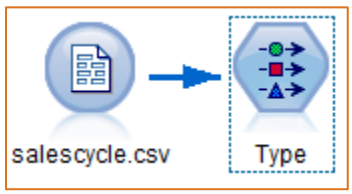
   In order to use standard time series methods it is important to collect, or at least be able to summarize, the information over equal time periods. Within a time series dataset it is essential that the rows represent equally spaced time periods. Even time periods for which no data was collected must be included as rows in the data file (with missing values for the fields).

4. Close the **Preview** output window.
5. Close the **Var. File** dialog box.

Task 3. Instantiate the data.

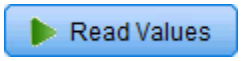
1. From the **Field Ops** palette, add a **Type** node downstream from the **Var. File** node. (In this course, node B is said to be downstream from node A, if data flows from A to B; this also implies that the nodes are connected, with a connection from A to B.)

   The results appear as follows:



2. Edit the **Type** node.

   Date_ is typed as a date field, and **sales** as a continuous field.

3. Click the **Read Values** 

button.

   Dates run from January 1982 to December 1995; minimum and maximum sales are 553 and 969 respectively. (you may need to expand the Values column)

4. Close the **Type** dialog box.

Task 4. Plot the time series.

The simplest way of identifying patterns in the series is to plot the data.

A Time Plot is used to examine time series data. This chart plots the value of the field on the vertical axis, and time on the horizontal axis. Points are joined up to give a line graph showing any important patterns in your data.

In this example, the interest might be to see how sales have changed over the fourteen-year period of interest.

1. From the **Graphs** palette, select the **Time Plot**, and add it downstream from the **Type** node.
2. Edit the **Time Plot** node.
3. Beside **Series**, select **sales**.

   By default, the x axis will show sequential numbers. In this example, you can use the Date_ field to label the x axis.

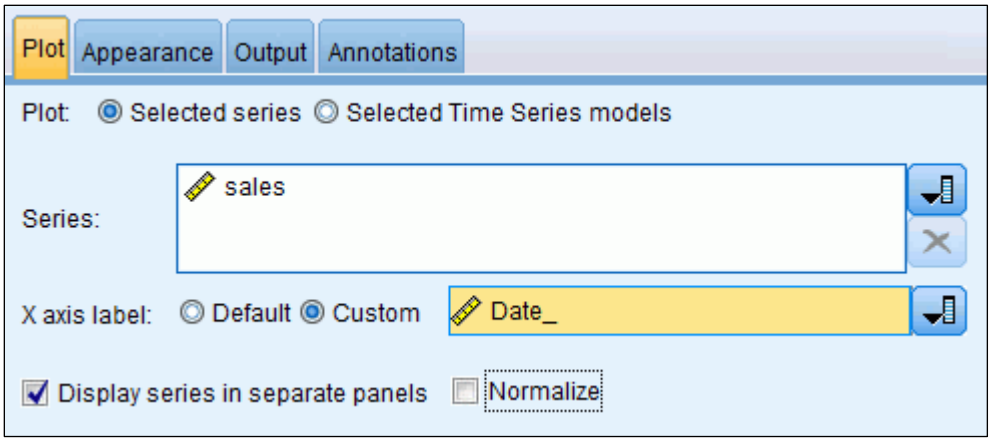
4. Beside the **X axis label**, enable the **Custom** option, and then select **Date_**.

   You could plot multiple series in a single chart, with the option to normalize the plot to ensure fields in different units of measurements can be compared.

   In this example, there is only one field, sales, so normalization is not applicable.

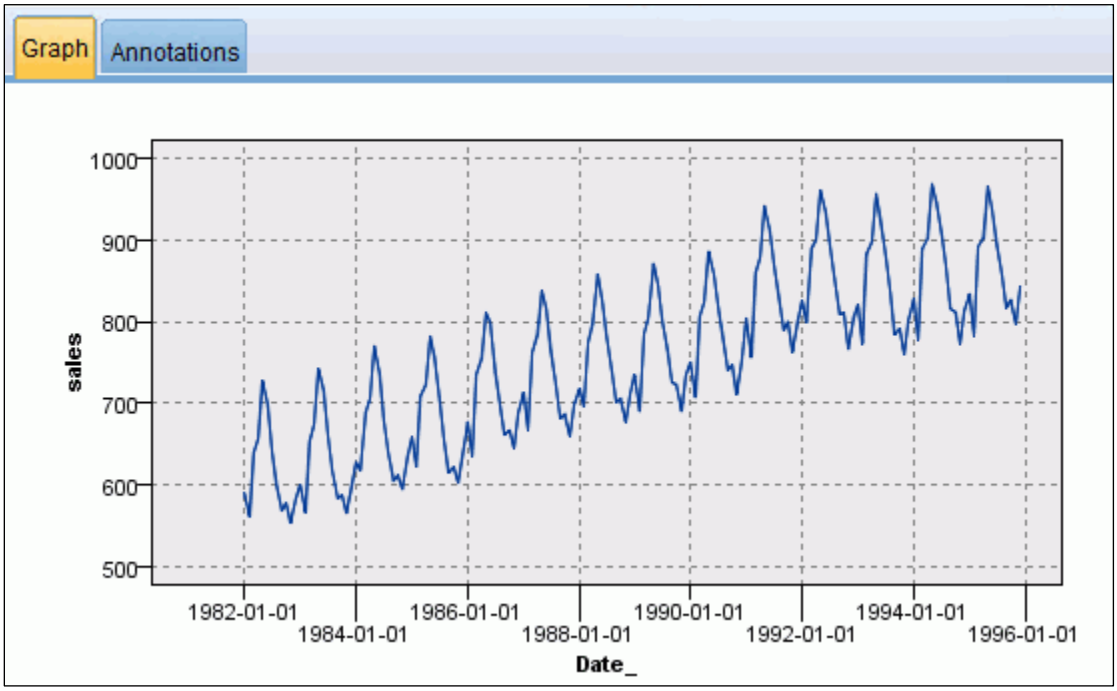
5. Disable the **Normalize** option.

   The results appear as follows:



6. Click **Run**.

   The results appear as follows:



   There is a clear upward trend in the data as sales have continued to increase from 1982 until 1995, albeit less pronounced from the beginning of 1991.

   There is also a clear pattern of seasonality in the series. Recall that a seasonal pattern occurs when there is a recurring pattern at regular intervals in the series. In this dataset, every year there appears to be a rise in sales during the early part of the year while sales tend to fall to a low around November. Finally, there is some recovery in sales leading up to the Christmas period of each year.

7. Close the **Time Plot** output window.

   This completes the demonstration. You will create a clean state for the next demonstration.

8. From the **File** menu, click **Close Stream**. Click **No** when asked to save changes.
9. From the **File** menu, click **New Stream**.

   Leave IBM SPSS Modeler open for the next demonstration.

Results:
You successfully created a time plot of your time series. Based on the results, there are clear patterns of trend and seasonality in the data. Thus, in order to model the data, you will require a modeling technique that can take into account trend and seasonality in the data.

You will find the completed stream in the following folder:
C:\Training\0A028\01-Introduction_to_Time_Series_Analysis\Solutions

IBM Training

Demonstration 2

Replace missing values

Demonstration 2: Replace missing values

Demonstration 2: Replace missing values

Purpose:
Before you create your model, you notice that data you are analyzing has some missing data. Because some time series techniques require complete data, you decide to replace

Data file: **parcelmiss.csv**

Data folder: **C:\Training\0A028**

Task 1. Start and configure IBM SPSS Modeler 18.1.1.

1. If not already open, from the **Start** menu, expand **IBM SPSS Modeler 18.1**, and then click **IBM SPSS Modeler 18.1**.

   Note: IBM SPSS Modeler 18.1.1 displays "IBM SPSS Modeler 18.1" in the Start menu and program name.
2. Click **Cancel** to close the **Welcome** dialog box.

   If you have already set the working directory in a previous demonstration or exercise, you can skip to Task 2.
3. If not already set, from the **File** menu, click **Set Directory**.
4. Beside **Look in**, navigate to the **C:\Training\0A028** folder, and then click **Set**.

Task 2. Take an initial look at the data.

1. From the **Sources** palette, add a **Var. File** node to the canvas.
2. Edit the **Var .File** node, and then select **parcelmiss.csv.**

3. Click **Preview**.

   The results appear as follows:

Table | Annotations

	parcel	Date_
1	11844.290	2017-01-02
2	$null$	2017-01-03
3	14054.290	2017-01-04
4	15680.000	2017-01-05
5	17092.860	2017-01-06
6	14610.000	2017-01-07
7	11584.290	2017-01-08
8	10865.710	2017-01-09
9	$null$	2017-01-10
10	15200.000	2017-01-11

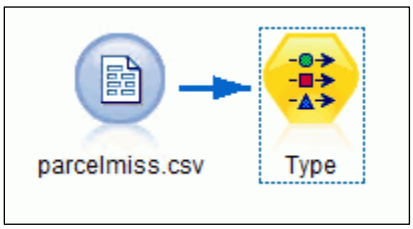
   Looking in the data it is apparent that for the first two weeks, data was not collected on Mondays. In this instance it is therefore necessary to replace the missing values with information collected on other dates. Unlike in many standard statistical analyses (for example, with survey data), estimating missing data is common in time series analysis.

4. Close the **Preview** output window.
5. Close the **Var. File** dialog box.

Task 3. Instantiate the data.

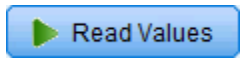
1. From the **Field Ops** palette, add a **Type** node downstream from the **Var. File** node. (In this course, node B is said to be downstream from node A, if data flows from A to B; this also implies that the nodes are connected, with a connection from A to B.)

   The results appear as follows:



2. Edit the **Type** node.

   Date_ is typed as a date field, and **parcel** as a continuous field.

3. Click the **Read Values**  button.

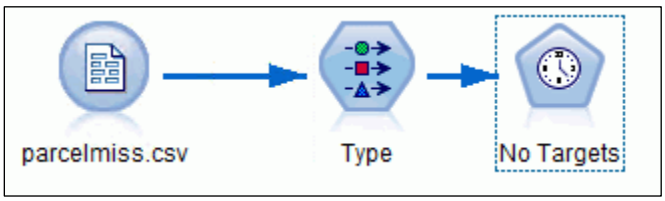
   Dates run from January 2, 2017 to May 7, 2017; minimum and maximum parcel are 9286.25 and 40234.29 respectively.

4. Close the **Type** dialog box.

Task 4. Replace missing values.

1. From the **Modeling** palette, add a **Time Series** node downstream from the **Type** node.

   The results appear as follows:



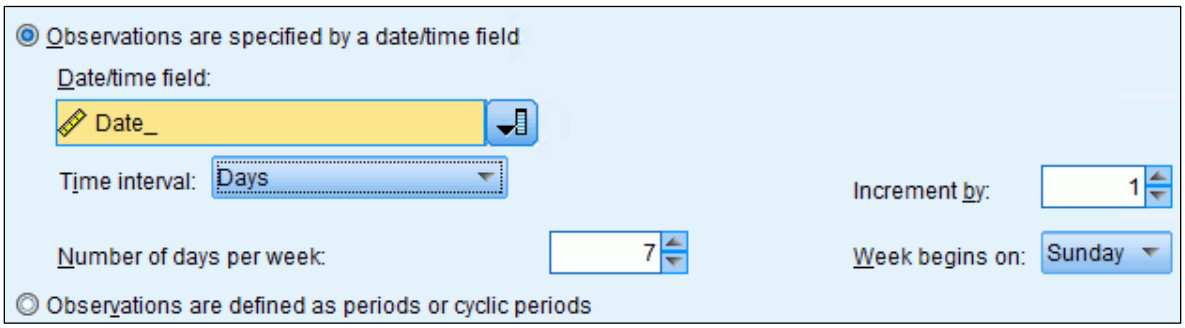
2. Edit the **Time Series** node.
3. Click the **Fields** tab, if necessary.

   You can set field roles upstream in the Type node, or you can specify the field roles here. Because you did not set field roles in the Type node yet, you will set them here.
4. Enable the **Use custom field assignments** option.

   The target is the field that you wish to forecast. In this example, **parcel** is the target field, and there are no predictors.
5. Move **parcel** into the **Targets** box.
6. Click the **Data Specifications** tab.
7. Click the **Observations** item on the left, if necessary.

   In this example, the observations are defined by a date field, named **Date_**. Because the data represented daily figures, the time interval between the observations are days.
8. Ensure that the **Observations are specified by a date/time field** option is enabled.
9. For **Date/time** field, select **Date_**.
10. Beside **Time interval**, select **Days**.

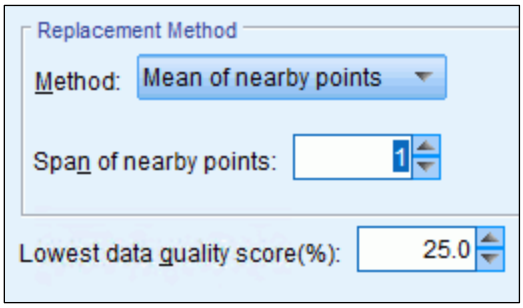
    The results appear as follows:



Note that each week is set to begin on a Sunday.

11. Click the **Missing Value Handling** item on the left side.
12. From the **Method** menu, click **Mean of nearby points**.
13. From the **Span of nearby points** menu, set the span to **1**. Span of 1 means that the mean of the valid value immediately above and below the missing value will be used to replace the missing value.

    The results appear as follows:



14. Click **Run**.

    After a short period of time execution is completed - and a nugget is created.

Task 5. Examine the results.

1. Edit the **model nugget**.
2. Click 

.

   The results appear as follows:

Table | Annotations

	Date_	$FutureFlag	parcel
1	2017-01-02	0	11844.290
2	2017-01-03	0	12949.290
3	2017-01-04	0	14054.290
4	2017-01-05	0	15680.000
5	2017-01-06	0	17092.860
6	2017-01-07	0	14610.000
7	2017-01-08	0	11584.290
8	2017-01-09	0	10865.710
9	2017-01-10	0	13032.855
10	2017-01-11	0	15200.000

   The missing values have been replaced. For example, the value for the second January 3, which was missing before is now 12949.29, which is the mean of the values for parcel on January 2 and 4.

3. Close the **Preview** window.
4. Close and **model nugget**.

   This completes the demonstration. You will create a clean state for the exercise.

5. From the **File** menu, click **Close Stream**. Click **No** when asked to save changes.
6. From the **File** menu, click **New Stream**.

   Leave IBM SPSS Modeler open for the exercise.

Results:
You successfully created a time plot of your time series. Based on the results, there are clear patterns of trend and seasonality in the data. Thus, in order to model the data, you will require a modeling technique that can take into

You will find the completed stream in the following folder:
C:\Training\0A028\01-Introduction_to_Time_Series_Analysis\Solutions

IBM Training

Unit summary

- Explain what a time series analysis is
- Describe how time series models work
- Demonstrate the main principles behind a time series forecasting model

Unit summary

IBM Training

Exercise 1

Create a time plot of the data

Exercise 1: Create a time plot of the data

Exercise 1: Create a time plot of the data

You have a data file from a private mailing company that measures the volume of mail delivered each day of the week, including weekends, over an eighteen-week period. The company wants you to develop a model that forecasts the demand for its delivery service one-week ahead so that it can arrange the appropriate staffing levels. To help you decide on the appropriate time series technique, you decide to create a time plot of the data to identify any past patterns.

Task 1. Open the data file.

- Use a **Var. File** node to import the text file **parcel.csv**, located in **C:\Training\0A028.**
- Add a **Type** node downstream from the **Var. File** node.
- Edit the **Type** node, and click **Read Values** to instantiate the data.

Task 2. Create a time plot of the data.

- Add a **Time Plot** node downstream from the **Type** node.
- In the **Series** box, select **parcel**.
- Besides **X axis label**, enable the **Custom** option, and then select **Date_**.
- Clear the **Normalize** box.
- Click **Run**.
- What patterns do you see in the data?
- From the **File** menu, click **Exit** and exit **IBM SPSS Modeler** without saving anything.

For more information about where to work and the exercise results, refer to the Tasks and results section that follows. If you need more information to complete a task, refer to earlier demonstrations for detailed steps.

Exercise 1: Tasks and results

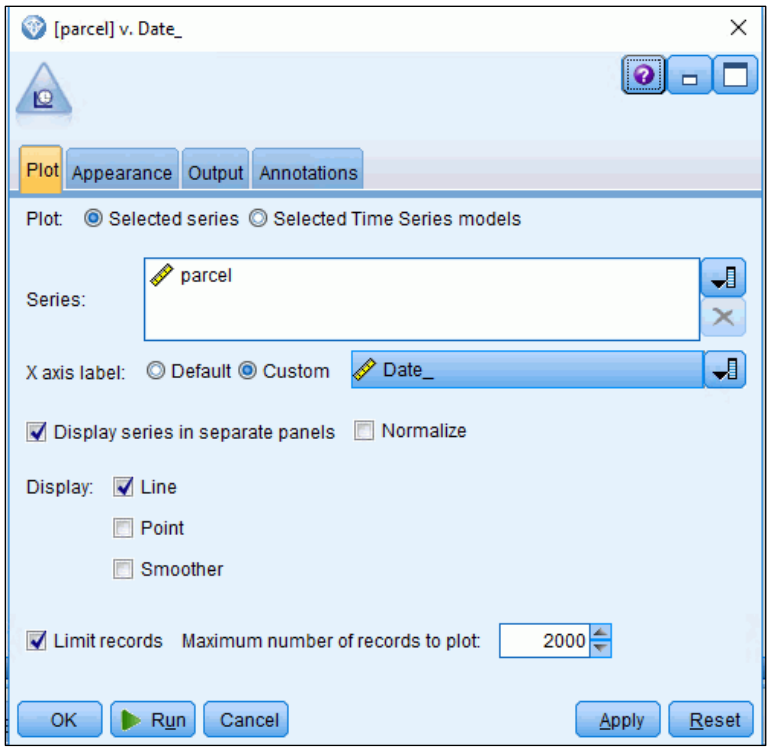
Task 1. Open the data file.

- From the **Sources** palette, drag a **Var. File** onto the stream canvas and import the data from **parcel.csv**.
- Close the **Var. File** dialog box.
- From the **Field Ops** palette, add a **Type** node downstream from the **Var. File** node.
- Edit the **Type** node, and click **Read Values** to instantiate the data.
- Click **OK**.

Task 2. Create a time plot of the data.

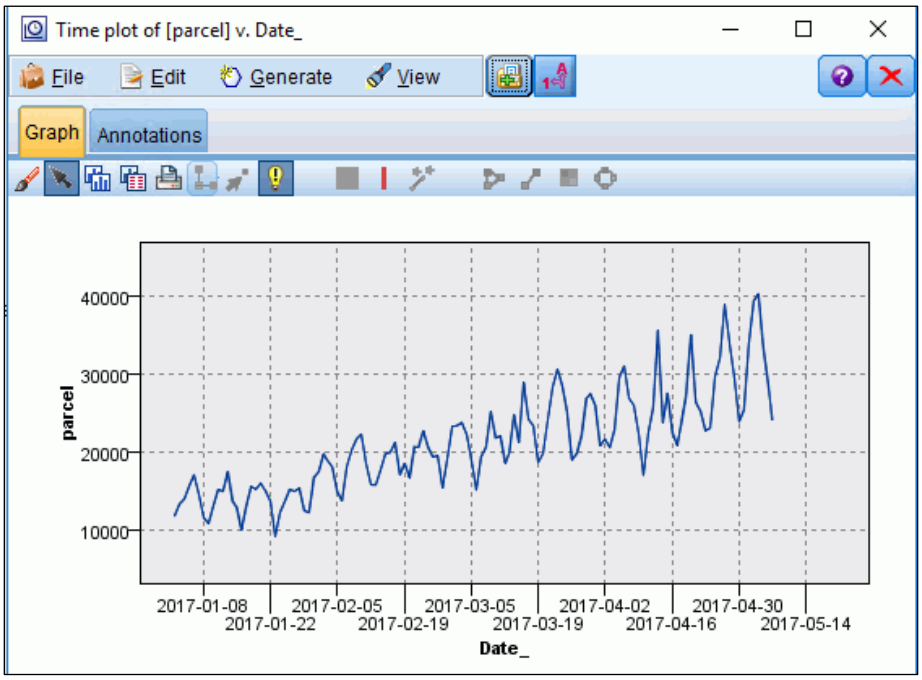
- From the **Graphs** palette, add a **Time Plot** node downstream from the **Type** node.
- In the **Series** box, select **parcel**.
- Besides **X axis label**, enable the **Custom** option, and then select **Date_**.
- Clear the **Normalize** box.

  The results appear as follows:



- Click **Run**.

  The results appear as follows:



- What patterns do you observe in the data?

  The results show two clear patterns in the data. Deliveries seem to follow a fairly well defined seasonal (here weekly) pattern in which certain days of the week have higher demand than others. Parcel delivery peaks around mid-week and reaches its low on Sunday. In addition, after a few weeks of relatively constant volume, the number of parcels has been steadily increasing for many weeks.

  These findings suggest that the technique you use to model the data needs to be able to take into account both trend and seasonality.

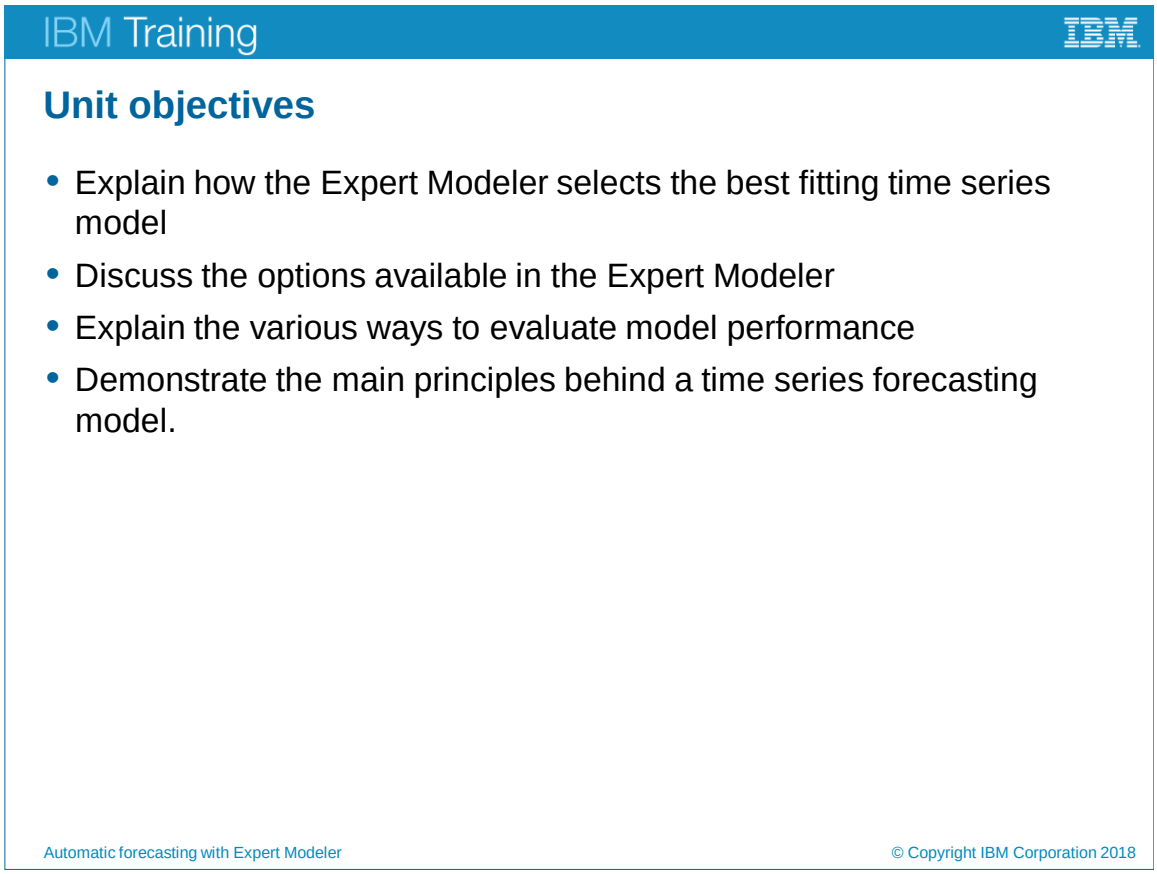
- Close the **Time plot** output window.
- From the **File** menu, click **Exit**, and exit **IBM SPSS Modeler** without saving.

You will find the completed stream in the following folder:
C:\Training\0A028\01-Introduction_to_Time_Series_Analysis\Solutions

IBM Training

Automatic forecasting with Expert Modeler

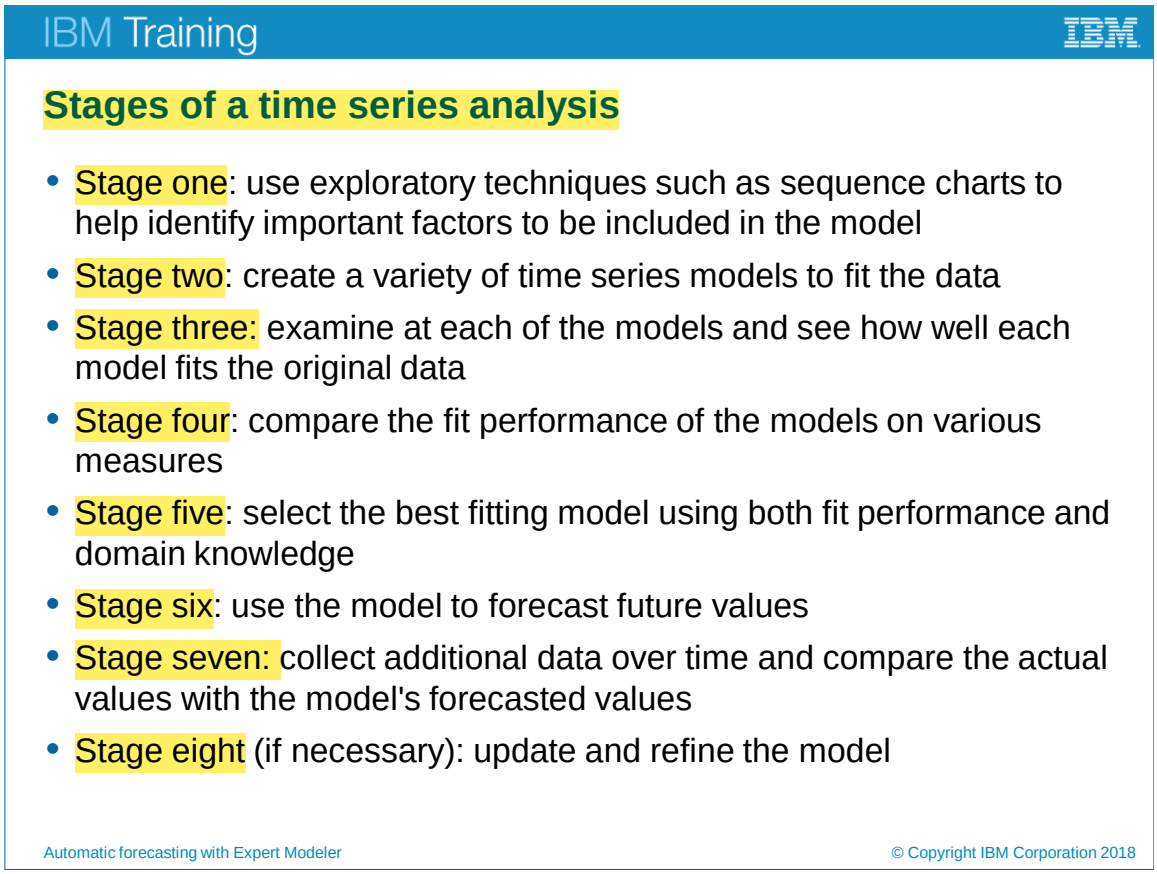
IBM SPSS Modeler (v18.1.1)



Unit objectives

Before reviewing this unit, you should be familiar with the following topics:

- Working with IBM SPSS Modeler (streams, nodes, palettes)
- Importing data (Var. File node)
- Defining measurement levels, roles, blanks, and instantiating data (Type node)
- Examining the data (Table node, Time Plot node)



Stages of a time series analysis

As you begin your first attempt at forecasting, it is important to review a general approach to developing a forecasting model. The general approach when using time series modeling is to break the process down into a number of stages. The steps below are often not followed in a linear fashion; instead, developing a model is very much a recursive approach.

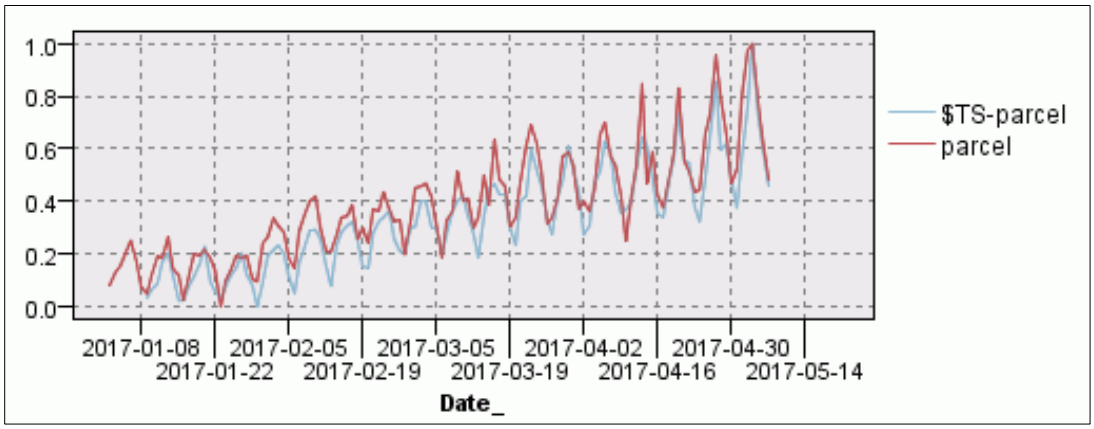
- Stage one involves using exploratory techniques such as Time plots, to help identify important factors to be included in the models. This is where the analyst and the subject matter expert use their experience and knowledge to look at the data and choose the important fields and approach.
- Stage two involves creating a variety of time series models to fit the data. Some of the models will be pure time series models and some will be causal time series models (assuming appropriate predictors are available).
- Stage three is where the analyst looks at each of the models and evaluates how well each fits the original data.
- In stage four the analyst will compare the fit performance of the models on various measures.
- In stage five the analyst will select the best model, using both fit performance and domain knowledge.

- In stage six the analyst uses the model to forecast future values.
- In stage seven, the analyst will wait and collect additional data over time to see how well the chosen model performed.
- In stage eight (if necessary), the analyst will update and refine the model, and continue the process.

These steps should not be regarded as rigid but as a guide to the way in which to develop successful forecasting models. It is important to compare several models before choosing a final model.

IBM Training

Examine fit and error



Automatic forecasting with Expert Modeler

Examine fit and error

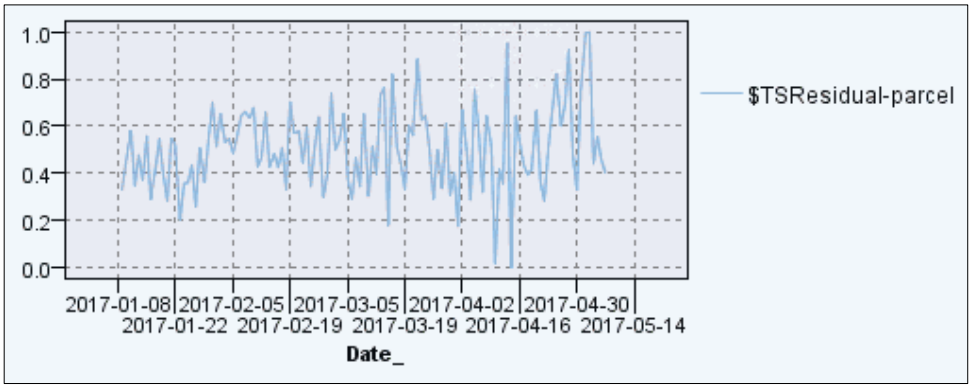
Time series models typically fit past data very well, but they will not fit perfectly. When assessing the performance of a model, you commonly compare the original series and the model predictions on a time plot.

To see how well the time series model has fit the historical series, you should create a time plot of both the model fit and the actual values. In this example, the field parcel, the historic series, is plotted together with another field called $TS-parcel, which is the fit created from a time series model.

The model fit is not perfect. Any variation not picked up by the model is unexplained variation.

IBM Training

Examine unexplained variation



Automatic forecasting with Expert Modeler

Examine unexplained variation

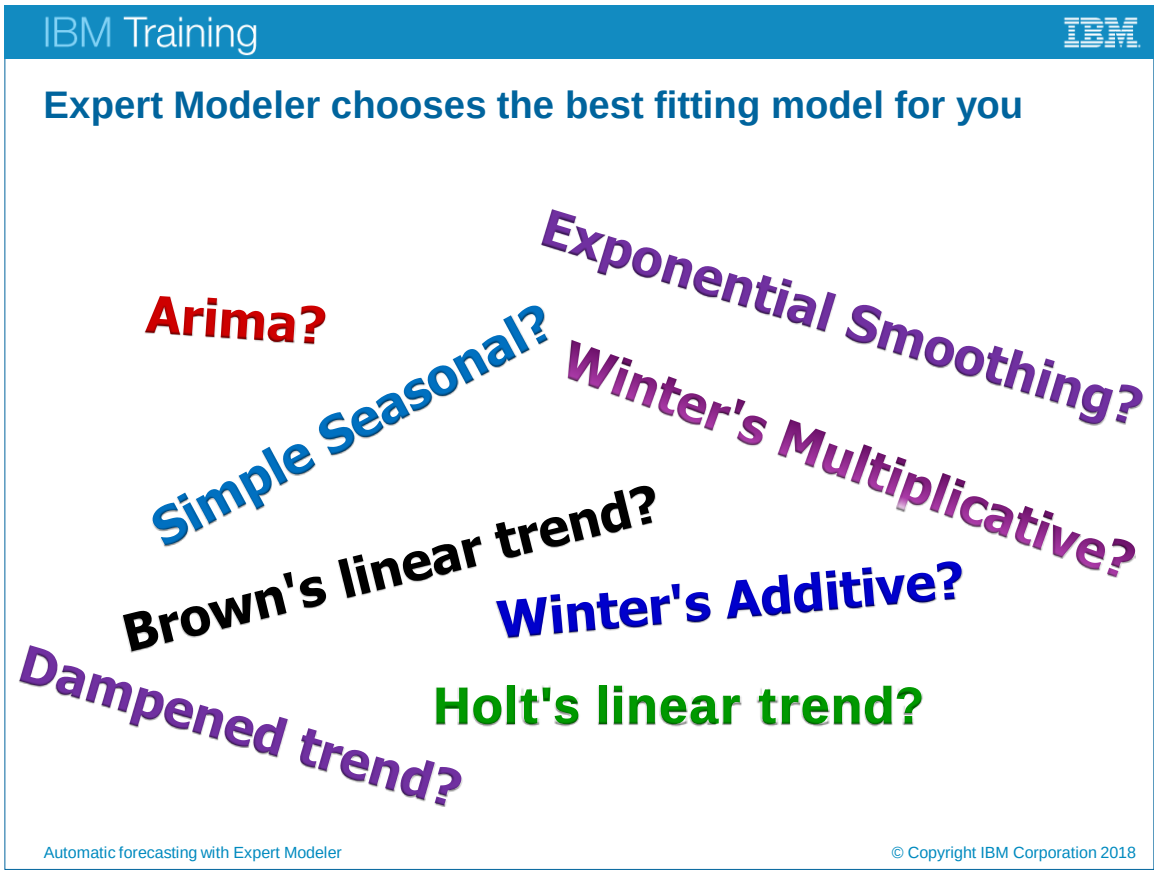
There appear to be about equal numbers of positive and negative errors, which is a good sign because it suggests that the errors are random. In order for a time series model to be successful, there should be no systematic pattern in the residuals. If the errors are random, it simply means that sometimes the model over predicts and at other times it under predicts, but the magnitude of the errors is not time dependent. Notice in the graphic that the errors seem to become more pronounced toward the end of the series. The errors are therefore heteroscedastic and this might affect the performance of the time series forecast.

The time series model will invariably not explain all of the total variation of the series. The remaining unexplained variation is known as the model error or residual. The model error will be positive if the series' value is under-predicted by the model fit. For example if the true value of parcel is 100 for a particular period and the model predicts only 80 then the error is +20. Alternatively, the error will be negative if the series' value is over-predicted. If, for example, actual sales equaled 200 and the model predicts at 250, the error is –50. The model error always makes up the difference between the actual value of the series and the model fit, as with a regression model.

You should always see how the model error changes over time by using a time plot. If a time series model has been successful in extracting all the common patterns out of the series, then the error should be random. This is a very important test of a time series model. Not all models will be successful at detecting common patterns in the data. A model may, for example, fail to use an important factor influencing the series you wish to forecast. If a model is unsuccessful in explaining common patterns in the data, then some of the common patterns will instead be picked up by the residual or error fields. If this is the case the error term will not be random, and the time series model will be misspecified. Some important variation will be omitted from the model and its fit. This will suggest that there might be alternative time series models that fit the historic series better.

If the model is a good fit then the actual series will be close to the series fit, and the error would therefore be close to zero for all points over time. In the previous example, you can see that error looks fairly random. There is no obvious tendency for a positive error to be followed by another positive error. That is, the model does not consistently under-predict (or over-predict) over a number of consecutive periods.

There appear to be about equal numbers of positive and negative errors. The highest positive error is 30. Given the values of predicted sales (somewhere between 600 and 1,000), this is a small relative (and absolute) error. Similarly, the highest negative error is about –20. The magnitude of the $TSResidual-parcel values must be evaluated by comparing them to the level and variation in the observed series. Formal methods for doing so are available in the IBM SPSS Modeler, which will be used latter in the course.



Expert Modeler chooses the best fitting model for you

The Expert Modeler will choose the best time series model for a series or a group of series. The Expert Modeler automatically identifies and estimates the best-fitting exponential smoothing or ARIMA model. This section contains a brief overview of how the Expert Modeler works.

Model Selection

The Expert Modeler chooses between the best exponential smoothing model and the best univariate ARIMA model by taking the one with the best overall fit (as measured by a certain statistic, normalized BIC, presented in unit 3, Measuring Model Performance, of this course).

By default the Expert modeler considers seasonal models check box is selected. If your data exhibits seasonal patterns, the Expert Modeler will consider them when developing a best model, and, in general, you want to model seasonal patterns if they are present in your data. For example, if your data are quarterly or monthly, then you might expect seasonality. Considering seasonal models is more computer-intensive than not doing so, but should result in better models.

No Predictors

If no predictors are selected in the Expert Modeler, it chooses between the best exponential smoothing model and the best univariate ARIMA model as follows:

- The Expert Modeler identifies the exponential smoothing model with the best fit—this will be discussed in this and later units.
- The Expert Modeler identifies a univariate ARIMA model (that is, a model with no predictors) by deciding what the amount of the lag for the series should be (for both series values and errors), and if any differencing is needed. The model is then adjusted by the significance of the parameters selected.

Models with Predictors

If predictors are selected in the Expert Modeler, it then chooses a multivariate ARIMA model (Transfer Function) by starting with a model that uses all of the predictors, identifying appropriate lags for these predictors, and also determining what lags should be included for the series and error, and what differencing is necessary.

IBM Training

Demonstration 1

Create a time series model with the Expert Modeler

Demonstration 1: Create a time series model with the Expert Modeler

Demonstration 1: Create a time series model with the Expert Modeler

Purpose:
You have examined the time plot of the car sales data, and did not see anything particularly unusual about it (outliers, unequal variance, and so on). At this point, you will proceed directly to the Expert Modeler to create a model.

Stream file: **unit_2_demonstration_1_start.str**

Folder: **C:\Training\0A028\02-Automated_Forecasting_with_Expert Modeler\Start**

Task 1. Start and configure IBM SPSS Modeler 18.1.1.

1. From the **Start** menu, expand **IBM SPSS Modeler 18.1**, and then click **IBM SPSS Modeler 18.1**.

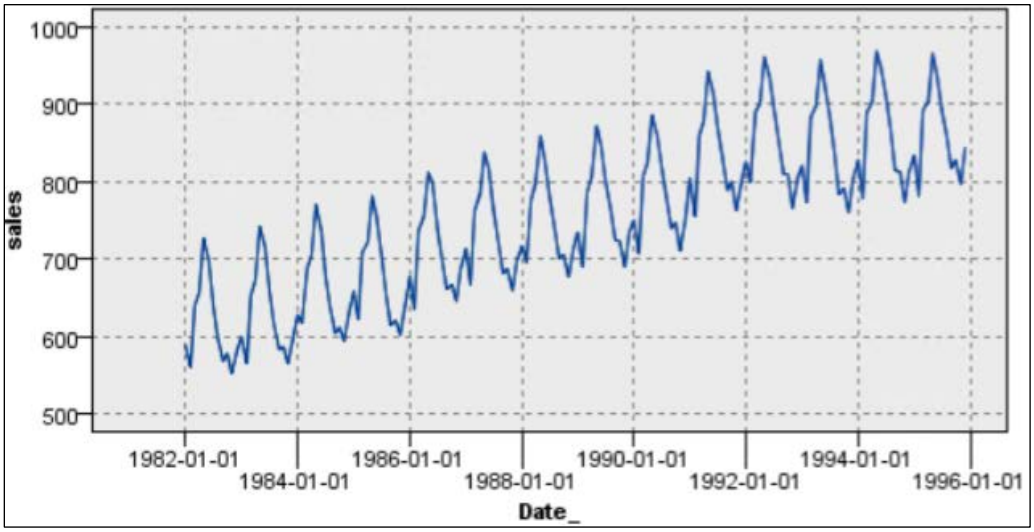
   Note: IBM SPSS Modeler 18.1.1 displays "IBM SPSS Modeler 18.1" in the Start menu and program name.
2. Click **Cancel** to close the **Welcome** dialog box.

   If you have already set the working directory in a previous demonstration or exercise, you can skip to Task 2.
3. From the **File** menu, click **Set Directory**.
4. Beside **Look in**, navigate to the **C:\Training\0A028** folder, and then click **Set**.

Task 2. Open the stream.

1. From the **File** menu, click **Open Stream**.
2. Navigate to **02-Automated_Forecasting_with_Expert_Modeler\Start**, and then double-click **unit_2_demonstration_1_start.str.**
3. Run the **Time Plot** node.

   The results appear as follows:

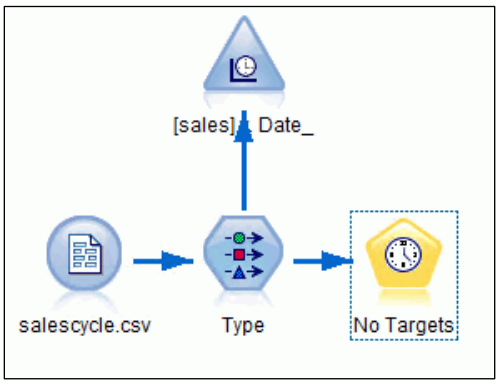


   The series has both a trend and seasonality.
4. Close the **Time Plot** output.

Task 3. Use the Expert Modeler to analyze the time series.

1. Click the **Modeling** palette, and then, at the left side, click **All**.
2. Add a **Time Series** node downstream from the **Type** node.

   The results appear similar to the following:



3. Edit the **Time Series** node.
4. Click the **Fields** tab, if necessary.

   You can set field roles upstream in the Type node, or you can specify the field roles here. Because you did not set field roles in the Type node yet, you will set them here.

5. Enable the **Use custom field assignments** option.

   The target is the field that you wish to forecast. In this example, sales is the target field, and there are no predictors.

6. Move **sales** into the **Targets** box.

   Candidate Inputs are fields used to predict the target. You do not have predictors in this dataset.

   There is also the option to add fields that flag events or interactions. Events and interactions are used to model the effects of occurrences of new advertising campaigns, strikes, changes in the law, and the like that might impact the target.

   There are no event or interaction fields in this example.

7. Click the **Data Specifications** tab.

8. Click the **Observations** item on the left, if necessary.

   In this example, the observations are defined by a date field, named **Date_**. Because the data represented monthly figures, the time interval between the observations are months.

9. Ensure that the **Observations are specified by a date/time field** option is selected.

10. For **Date/time** field, select **Date_**.

11. Beside **Time interval**, select **Months**.

12. Click the **Time Interval** item on the left.

    The dialog box tells you that the interval between observations is a month, which is taken from the previous specifications. The time interval between observations is never longer or shorter than one month, so the default option, Time interval for analysis is the interval between observations, is correct.

13. Click the **Aggregation and Distribution** item on the left.

    When the time series does not conform to the time interval you specified, you can aggregate or distribute the data to arrive at the specified interval. When you would have observations on a daily basis, and want to analyze the data on a monthly basis, the daily observations should be aggregated to months. Likewise, when your time series is comprised of quarterly data, but want to analyze monthly data, then the quarterly data needs to be distributed over the three months in the quarter in question.

    In this example, observations are on a monthly basis, and the analysis should be done on a monthly basis too, so there is no need to aggregate or distribute the data.

14. Click the **Missing Value Handling** item on the left side.

    If a time series has missing values, for instance sales is not known for some months, you can use several methods to replace them.

    In this dataset, there are no missing values, so this does not apply.

15. Click the **Estimation Period** item on the left side.

    This section defines the set of records used to construct the model. By default it includes all the records in the dataset. It is often common to estimate a model on most, but not all, of the data, and then test it in the forecast period. The forecast period begins at the first case after the estimation period. This will be demonstrated in a later example.

16. Click the **Build Options** tab.

17. Click the **General** item at the left, if necessary.

    By default, the Expert Modeler is selected. This mode will find the best model, "best" in terms of a specific statistical criterion.

    By default, the Expert Modeler will try both Exponential Smoothing models and ARIMA models. If preferred, you can search for the best model within the class of Exponential Smoothing models only, or within the class of ARIMA models only. In that case, use the Model Type buttons to restrict the models to that particular class of models.

    Rather than letting the Expert Modeler find the best model, you can choose a specific Exponential Smoothing or ARIMA model and run that model.

18. Click the **Method** dropdown, and then click **Exponential Smoothing**.

    You can choose a specific Exponential Smoothing model.

    In this example, you just will let IBM SPSS Modeler find the best model.

19. In the **Method** dropdown, click **Expert Modeler**.

20. Click the **Output** item on the left side.

    You can specify the maximum number of lags for the ACF and PACF. These functions will be presented in unit 3. You will use the default.

    If preferred, you can request predictor importances. This is only relevant in causal time series models.

21. Click the **Model Options** tab.

    The most important options in this dialog box relate to forecasting future values (the Extend records into the future option). If you use a causal model and you want to forecast the target into the future, you also will have to specify the future values of the predictors. The predictors' future values can either be computed or fixed values can be entered (the options under Future values to use in Forecasting).

    You will not make use of these options at this point.

22. Click **Run** (If the Run button does not show, make the dialog box smaller.)

    A model nugget (the yellow diamond) is generated that stores the results; it can also be used to score records.

Task 4. Examine the results.

1. Edit the **model nugget**.
2. Click the **Output** tab, if necessary.
3. In the outline on the left side, click the **Temporal Information Summary** table.

   This table lists the fields that were used to define the observations, the start and the end of the period, and the number of observations used in the analysis.

4. In the outline on the left side, click the **Model Information** table.
   The results appear as follows:

Model Information

Model Building Method		ARIMA
		Non-seasonal p=1,d=1,q=0; Seasonal p=0,d=1,q=1
Number of Predictors		0
Model Fit	MSE	54.941
	RMSE	7.412
	RMSPE	1.000
	MAE	5.681
	MAPE	0.760
	MAXAE	32.025
	MAXAPE	3.983
	AIC	622.956
	BIC	629.043
	R-Squared	0.994
	Stationary R-Squared	0.317
Ljung-Box Q(#)	Statistic	14.214
	df	16.0
	Significance	0.6

This table shows that the best model, in terms of a certain statistical criterion, is the ARIMA model, with non-seasonal components p=1, d= 1, and q= 0, and seasonal components p=0, d=1, and q= 1. ARIMA will be reviewed later, but for the moment, loosely speaking, notice that there is a non-seasonal part (representing the overall trend), and a seasonal component (representing the peaks and troughs in corresponding months).

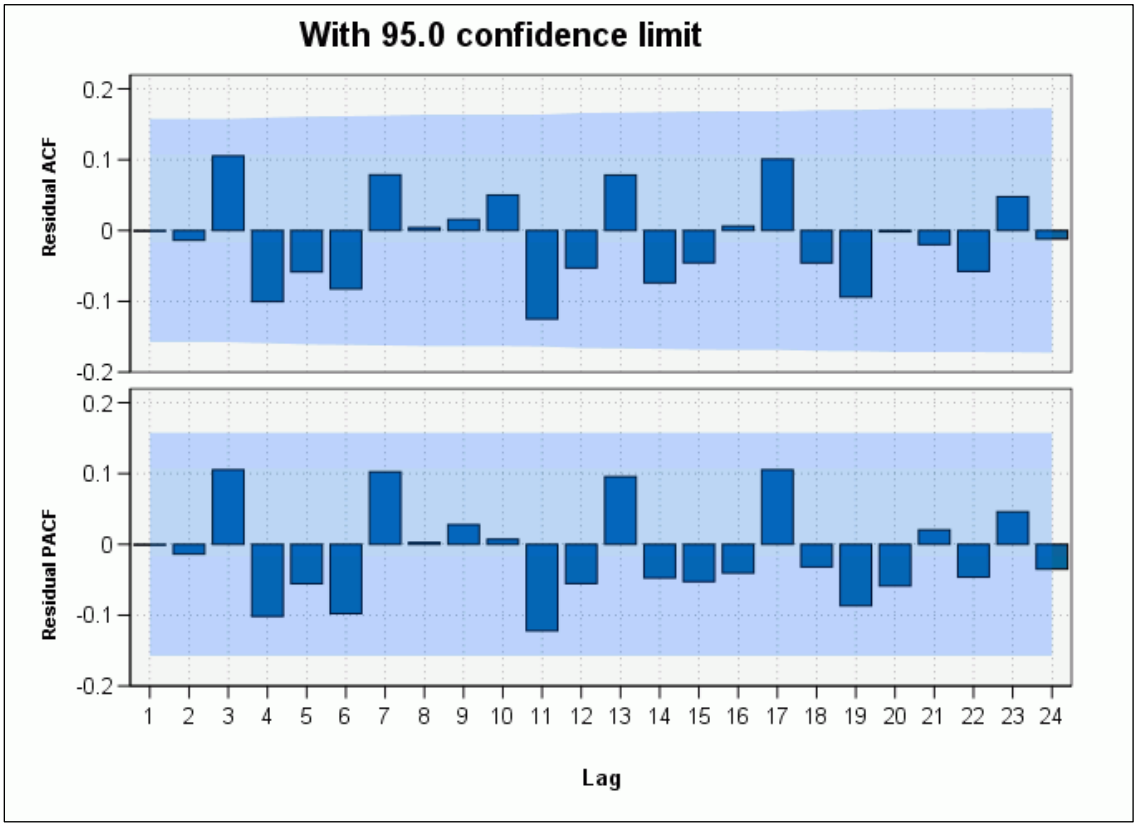
The table also includes a number of fit statistics. The most commonly used measure is the Mean Absolute Percentage Error (MAPE) which is 0.760, meaning that on average, the predictions are off by just 0.76%. The R-Squared value (0.994), points to a close fit, but the Stationary R-Squared (0.317) is generally preferred over R-Squared whenever there is trend in the series, as there is in this example. The Bayesian Information Criteria (BIC) measure is a measure of overall fit that enables you to compare different models for the same series. In general, all other things equal, you should prefer the model for a time series that has the minimum BIC.

The Ljung-Box statistic is a test of autocorrelation in the error. If the significance value is below 0.05 you would conclude that the series is autocorrelated.

The Ljung-Box statistic provides one overall test instead of examining many autocorrelation tests, each as the 0.05 level. The Ljung-Box statistic is calculated for the 18th lag for all models. Because the significance value in this example is 0.6, which is well above 0.05, you would conclude that there are no significant autocorrelations in the series (through lag 18).

5. Click **Correlogram**.

The results appear as follows:



The Residual PACF and Residual PACF charts are helpful to see whether there is any non-random pattern in the errors after creating the model. For each lag there is a bar. For example, the first bar shows the autocorrelation estimate calculated for the error term at time t and the error term at time t-1 (one time point back) throughout the series. If all the bars fall without the confidence intervals, then there are no significant autocorrelations in the error series. If any of the bars extend out of the shaded area, it would suggest that there may be non-random error which is not being captured by the model. Fortunately, in this example, there is no evidence of autocorrelation.

6. Click **Parameter Estimates**.

   The results appear as follows:

Parameter Estimates

				Coefficient	Std. Error	t	Significance
sales	No Transformation	AR	Lag 1	-0.225	0.079	-2.836	0.005
		MA, Seasonal	Lag 1	0.620	0.073	8.461	0.000

   This table reports the estimates corresponding to the ARIMA model with non-seasonal components p=1, d=1 and q=0, and seasonal components p=0, d=1, and q=1.

   The table gives the estimate for the non-seasonal AR (p) parameter, and the seasonal MA (q) parameter. The parameter estimates for the non-seasonal and seasonal I (d) parameter is missing. The reason for this will become clear when ARIMA models are presented.

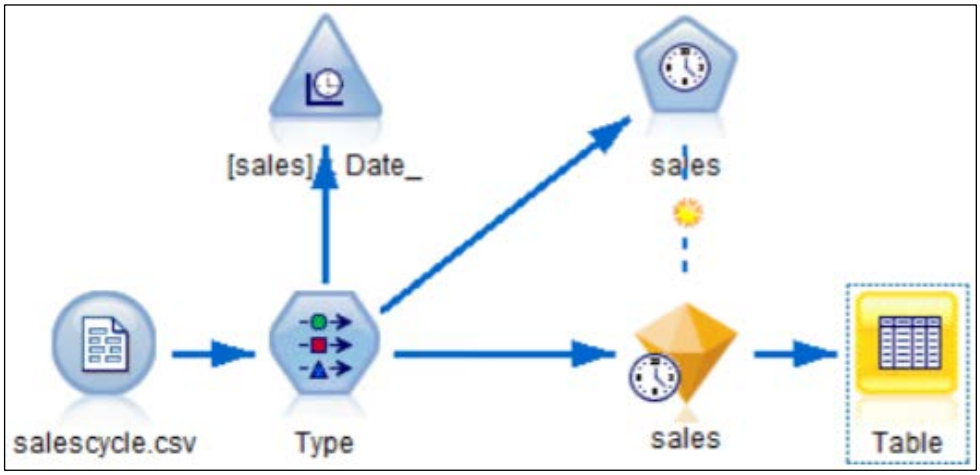
7. Close the **model nugget**.

Task 5. Score records.

The model nugget, like a model nugget from any other model, enables you to add the model's predictions to the dataset.

1. From the **Output** palette, add a **Table** node downstream from the model nugget.

   The results appear similar to the following:



2. Run the **Table** node.

The results appear as follows:

Table | Annotations

	Date_	$FutureFlag	sales	$TS-sales	$TSLCI-sales	$TSUCI-sales
1	1982-01-01	0	589....	$null$	$null$	$null$
2	1982-02-01	0	561....	$null$	$null$	$null$
3	1982-03-01	0	640....	$null$	$null$	$null$
4	1982-04-01	0	656....	$null$	$null$	$null$
5	1982-05-01	0	727....	$null$	$null$	$null$
6	1982-06-01	0	697....	$null$	$null$	$null$
7	1982-07-01	0	640....	$null$	$null$	$null$
8	1982-08-01	0	599....	$null$	$null$	$null$
9	1982-09-01	0	568....	$null$	$null$	$null$
10	1982-10-01	0	577....	$null$	$null$	$null$
11	1982-11-01	0	553....	$null$	$null$	$null$
12	1982-12-01	0	582....	$null$	$null$	$null$
13	1983-01-01	0	600....	$null$	$null$	$null$
14	1983-02-01	0	566....	572.000	554.572	589.428
15	1983-03-01	0	653....	646.353	629.374	663.332
16	1983-04-01	0	673....	667.196	650.217	684.175
17	1983-05-01	0	742....	743.098	726.119	760.077
18	1983-06-01	0	716....	712.451	695.472	729.430
19	1983-07-01	0	660....	658.098	641.119	675.077
20	1983-08-01	0	617....	618.775	601.796	635.754

A number of fields have been added to the dataset.

$FutureFlag indicates whether an observation makes part of the dataset, or whether the observation represents future data, beyond the end point of the series. $TS-SALES stores the predicted values based on the model. The fields $TSLCI-SALES and $TSUCI-SALES store the lower and upper 95% confidence limits for the predictions.

The first 13 records all have undefined values ($null$) values for the new fields. The reason is that the model uses past values for its prediction. Because of the seasonal component, the model also uses the observation from 12 months ago to compute the prediction. Since the first 12 months in the dataset do not have information from the year before, the fields added by the model are all undefined. Actually, in this ARIMA model with non-seasonal components p=1, d=1, and q=0 and seasonal components p=0, d=1, q=1 the first 13 records will have undefined values.

Although the first 13 records have undefined values for the fields added by the model, from record #14 forward you have a prediction and the confidence interval for the prediction. For example, for record #14, the predicted SALES for February 1983 is 572. This is only an estimate, so the true, unknown value might differ from this predicted value. The lower and upper limits tell you that you can be 95% confident that the interval [554.572, 589.428] contains the true value.

3. Scroll down to the end of the output.

   The predicted values for these observations are higher than those for the first observations. This reflects that there is a positive trend in the series.

4. Close the **Table** output window.

Task 6. Forecast future values.

Assuming that the model is satisfactory, you can use it to forecast future values. There are two ways to accomplish this. You can enable an option in the Time Series modeling node and then rerun the model; the model nugget that will be generated will then have the option to predict future values enabled. Alternatively, you can enable the appropriate option in the model nugget alone.

In this demonstration, you will enable the option in the Time Series modeling node and then rerun the model.

1. Edit the **Time Series** node (not the model nugget).
2. Click the **Model Options** tab.
3. Enable the **Extend records into the future** option, and specify the value **12**.

   This will predict sales for the next year, 1996.

4. Click **Run**.

   The model nugget is updated. The model will be the same as in the first run, but there will now be an option enabled to add future records. You will verify that.

5. Edit the **model nugget**.
6. Click the **Settings** tab, and then click the **Forecast** item, if necessary.

   The option to extend records into the future is enabled. If you would not have rerun the Time Series modeling node, you could have enabled this option to add future predicted values.

7. Close the **model nugget**.

8. Run the **Table** node that is downstream from the **model nugget**, and then scroll down to the end.

   The results appear as follows:

Table | Annotations

	Date_	$FutureFlag	sales	$TS-sales	$TSLCI-sales	$TSUCI-sales
161	1995-05-01	0	966....	966.984	952.551	981.418
162	1995-06-01	0	937....	939.621	925.188	954.055
163	1995-07-01	0	896....	896.006	881.572	910.439
164	1995-08-01	0	858....	855.306	840.872	869.740
165	1995-09-01	0	817....	808.522	794.088	822.955
166	1995-10-01	0	827....	817.423	802.989	831.857
167	1995-11-01	0	797....	788.727	774.293	803.160
168	1995-12-01	0	843....	835.517	821.084	849.951
169	1996-01-01	1	$null$	865.352	850.918	879.786
170	1996-02-01	1	$null$	818.398	800.142	836.655
171	1996-03-01	1	$null$	925.021	903.221	946.821
172	1996-04-01	1	$null$	938.102	913.338	962.865
173	1996-05-01	1	$null$	1001.242	973.817	1028.667
174	1996-06-01	1	$null$	973.835	943.988	1003.681
175	1996-07-01	1	$null$	932.479	900.392	964.566
176	1996-08-01	1	$null$	892.890	858.710	927.070
177	1996-09-01	1	$null$	846.996	810.844	883.149
178	1996-10-01	1	$null$	852.166	814.144	890.189
179	1996-11-01	1	$null$	818.130	778.325	857.935
180	1996-12-01	1	$null$	860.404	818.893	901.916

   Twelve records are added to the dataset, for January 1996 to December 1996. $FutureFlag equals 1 for these records, indicating that these are forecasts. You can also observe that from the fact that sales is undefined for these records.

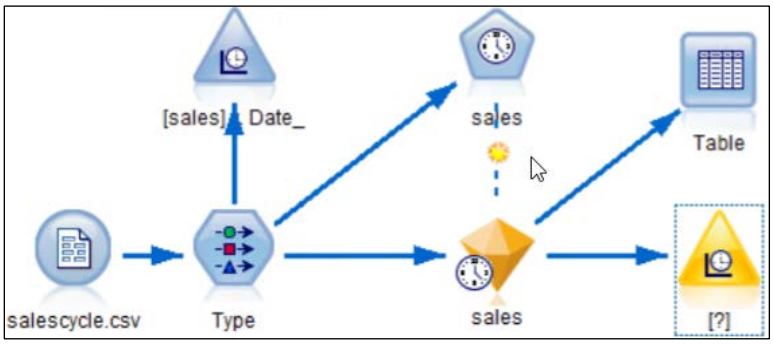
9. Close the **Table** output window.

Task 7. Plot the original series, predictions, and forecasts.

The Time Plot node lets you chart the original series, the predictions, and the forecasts. (Ensure that the option to add future values is enabled on the Settings tab in the model nugget, as demonstrated in the previous task.)

1. From the **Graphs** palette, add a **Time Plot** node downstream from the model nugget

   The results appear similar to the following:

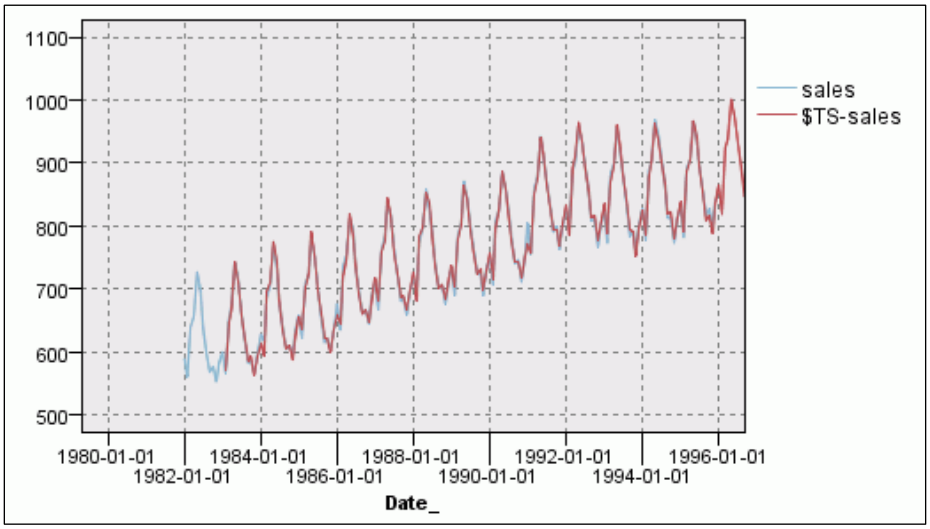


2. Edit the **Time Plot** node.

3. Beside **Series**, select **sales** and **$TS-sales**.
4. Beside **X axis label**, select the **Custom** option, and then select **Date_**.

   To compare how closely the predictions fit the observed values, you will chart both series in a single chart, without separate panels. Because both series have the same unit of measurement, there is no need to normalize the series.
5. Clear the **Display series in separate panels** option.
6. Clear the **Normalize** option.
7. Click **Run**.

   The results appear similar to the following:



   Predictions for the first observations are missing.

   It is hard to distinguish between the original series and the predictions, so there is a close fit.

   The original series runs to December 1995. Values from January 1996 forward are forecasts. The forecasts show the same pattern as the historical data, although at a somewhat higher level because of the upward trend.
8. Close the **Time Plot** output window.

   This completes the demonstration. You will create a clean state for the exercise.
9. From the **File** menu, click **Close Stream**. Click **No** when asked to save changes.
10. From the **File** menu, click **New Stream**.

    Leave IBM SPSS Modeler open for the exercise.

Results:
You successfully used the Expert Modeler to create a time series model of your data and to forecast values one year into the future.

You will find the completed stream in the following folder:
C:\Training\0A028\02-Automatic_Forecasting_with_Expert_Modeler\Solutions

IBM Training

Unit summary

- Explain how the Expert Modeler selects the best fitting time series model
- Discuss the options available in the Expert Modeler
- Explain the various ways to evaluate model performance
- Demonstrate the main principles behind a time series forecasting model.

Unit summary

IBM Training

Exercise 1

Create a time series model with the Expert Modeler

Exercise 1: Create a time series model with the Expert Modeler

Exercise 1: Create a time series model with the Expert Modeler

You have a data file from a private mailing company that measures the volume of mail delivered each day of the week, including weekends, over an eighteen-week period. The time plot you created of the data shows definite trend and seasonality patterns. Because you are unsure of which modeling technique would be best for your data, you decide to use the Expert Modeler to help you pick out the best model.

Stream file: **unit_2_exercise_1_start.str**

Folder: **C:\Training\0A028\02-Automatic_Forecasting_with_Expert_Modeler\Start**

Task 1. Open the stream.

- From the **File** menu, click **Open Stream**.
- Navigate to **C:\Training\0A028\02-Automatic_Forecasting_with_Expert_Modeler\Start**, and then double-click **unit_2_exercise_1_start.str**.
- Run the **Time Plot** node.

Task 2. Create a time series model with Expert Modeler.

- Add a **Time Series** node downstream from the **Type** node.
- Edit the **Time Series** node.
- In the **Series** box, select **parcel**.
- Select **Date_** as the **Date/time** field.
- Select **Days** as the **Time interval**.
- Forecast **7** days into the future.
- Click **Run**.

Task 3. Evaluate the model.

- Open the **model nugget** and review the output.
- Select the **Model Information** table.
- What kind of model did the Expert Modeler pick for your data?
- From looking at the fit measures, how well does the model fit the data?
- Based on Ljung-Box Q(#), is there any evidence of autocorrelation?

- Select the **Correlogram**.
- Examine the ACF and PACF plots. Is there any evidence of autocorrelation? If so, at which lag(s)?
- Close the generated model.
- Attach a **Time Plot** node downstream from the **model nugget**.
- Create a time plot of the historical and predicted series. Be sure to display them both on the same chart, not separately. How well do the predicted values match the historical values?
- Exit from **IBM SPSS Modeler** without saving anything.

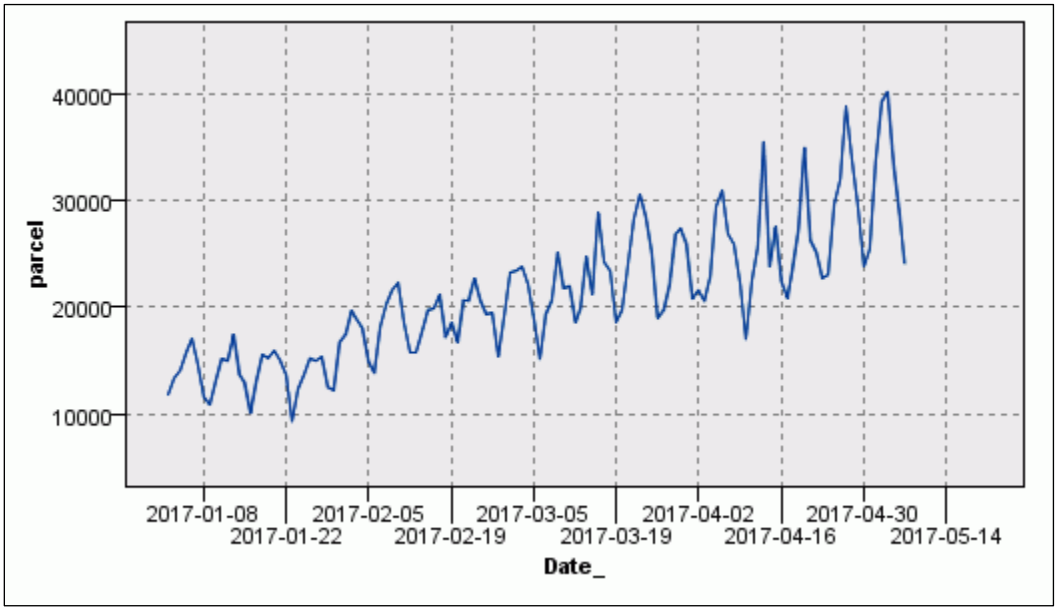
For more information about where to work and the exercise results, refer to the Tasks and results section that follows. If you need more information to complete a task, refer to earlier demonstrations for detailed steps.

Exercise 1: Tasks and results

Task 1. Open the stream.

- From the **File** menu, click **Open Stream**.
- Navigate to **C:\Training\0A028\02-Automatic_Forecasting_with_Expert_Modeler\Start**, and then double-click **unit_2_exercise_1_start.str**.
- Run the **Time Plot** node.

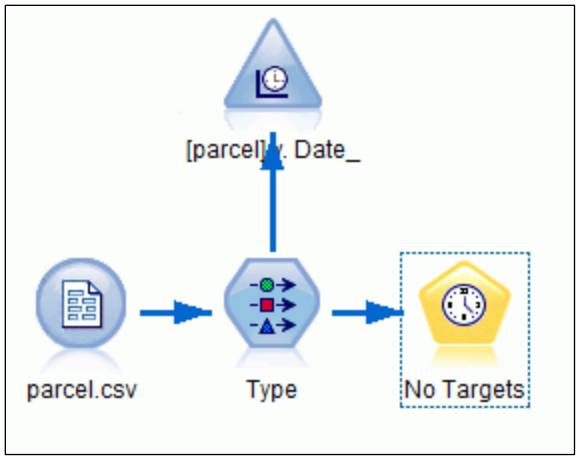
  The results appear as follows:



Task 2. Create a time series model with Expert Modeler.

- From the **Modeling** palette, **All** sub palette, add a **Time Series** node downstream from the **Type** node.

  The results appear as follows:



- Edit the **Time Series** node.
- Click **Use custom field assignments**.
- Move **parcel** to the **Targets** box.
- Click the **Data Specifications** tab.
- Select **Date**_ as the **Date/time** field.
- Select **Days** as the **Time interval**.
- Click the **Model Options** tab.
- Select **Extend records into the future**, and specify the value **7.**
- Click **Run**.

Task 3. Evaluate the model.

- Open the **model nugget** and review the output.
- Select the **Model Information**.

  The results appear as follows:

Model Information

Model Building Method		ARIMA
		Non-seasonal p=0,d=0,q=2; Seasonal p=0,d=1,q=1
Number of Predictors		0
Model Fit	MSE	4,259,485.215
	RMSE	2,063.852
	RMSPE	9.968
	MAE	1,574.977
	MAPE	7.634
	MAXAE	5,751.650
	MAXAPE	37.620
	AIC	1,820.426
	BIC	1,831.542
	R-Squared	0.896
	Stationary R-Squared	0.256
Ljung-Box Q(#)	Statistic	23.808
	df	15.0
	Significance	0.1

- What kind of model did the Expert Modeler pick for your data?

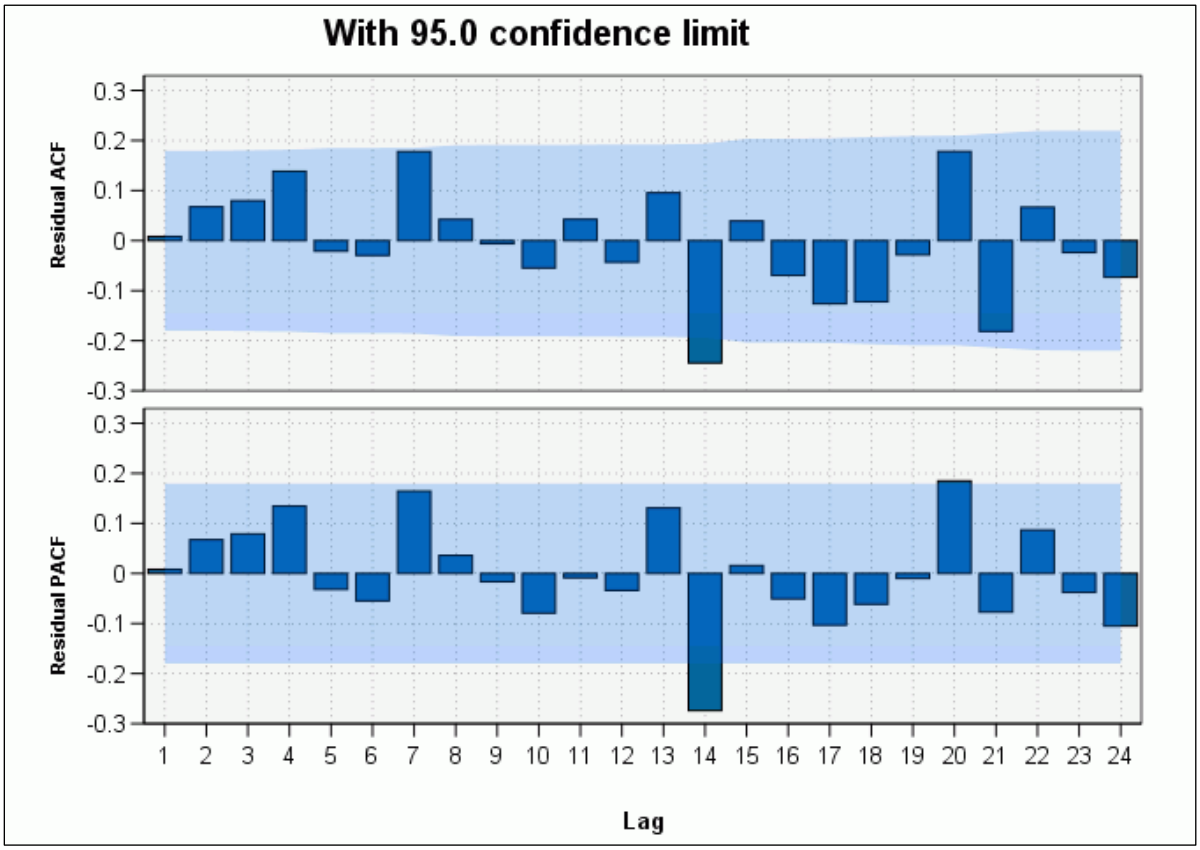
  The Expert Modeler selected an Arima (0,0,2)(0,1,1) model.
- From looking at the fit measures, how well does the model fit the data?

  Based on the MAPE measure, on average the predictions are off by 7.6%.
- Based on Ljung-Box Q(#), is there any evidence of autocorrelation?

  No. The significance value of 0.1 is well above the 0.05 threshold which indicates that there is no significant autocorrelation in the first 24 lags. It is important to remember that this is an omnibus test and does not discount the possibility that there may be individual lags that may be significant.

- Select the **Correlogram**.

  The results appear as follows:



- Examine the ACF and PACF plots. Is there any evidence of autocorrelation? If so, at which lag(s)?

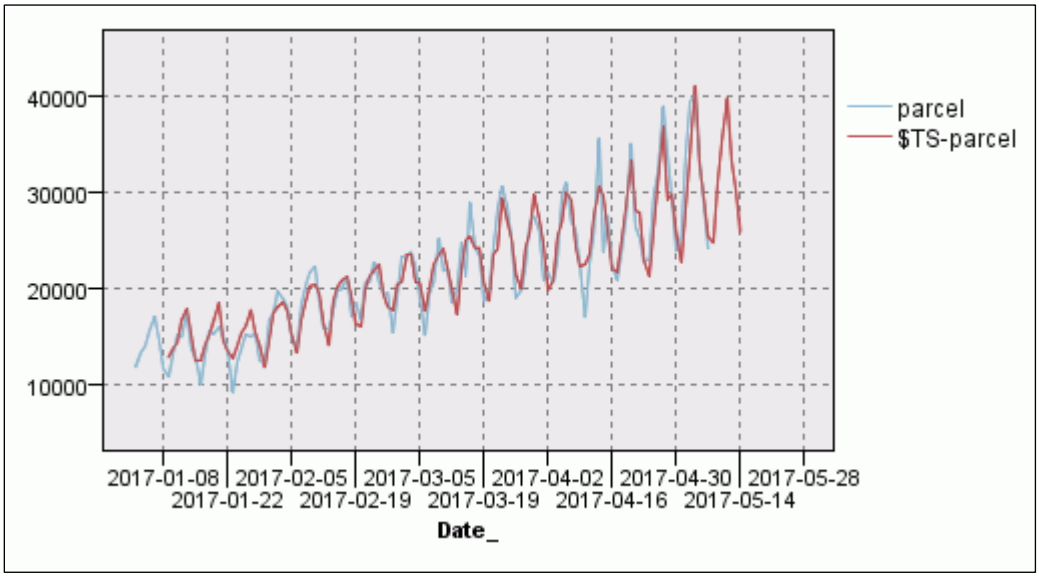
  Yes. There is significant autocorrelation in both plots at lag 14. It is not unusual to get a significant value even though the Ljung-Box Q(#) test showed no significant autocorrelation. Because you are dealing with weekly data, it is important to note that the significant value occurred at a seasonal lag. In this example, seasonal lags include 7, 14, 21, and so on. This may suggest that the model is not successfully capturing a seasonal pattern in the data.

- Close the model nugget.

- From the **Graphs** palette, attach a **Time Plot** node downstream from the **model nugget**.

- Create a time plot of the historical and predicted series'. Be sure to display them both on the same chart, not separately. How well do the predicted values match the historical values?
  - Edit the **Time Plot** node.
  - Beside **Series**, select **parcel** and **$TS-parcel**.
  - Beside **X axis label**, enable the **Custom** option, and then select **Date_**.
  - To compare how closely the predictions fit the observed values, you will chart both series in a single chart, without separate panels. Because both series have the same unit of measurement, there is no need to normalize the series.
  - Clear **the Display series in separate panels** option.
  - Clear the **Normalize** option.
  - Click **Run**.

    The results appear as follows:



    The forecasted values closely match the actual values.

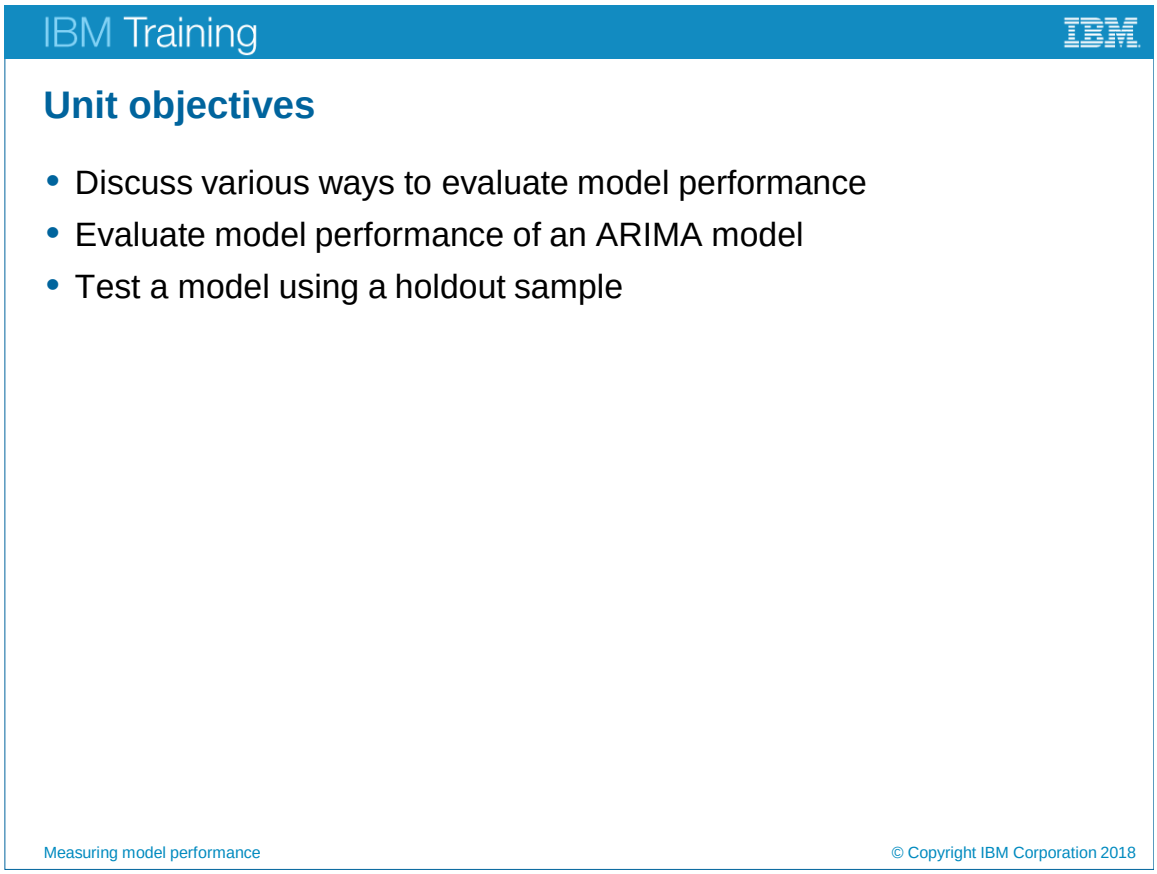
- Exit from **IBM SPSS Modeler** without saving anything.
- From the **File** menu, click **Exit** and then exit **IBM SPSS Modeler** without saving.

You will find the completed stream in the following folder:
C:\Training\0A028\02-Automatic_Forecasting_with_Expert_Modeler\Solutions

IBM Training

Measuring model performance

IBM SPSS Modeler (v18.1.1)



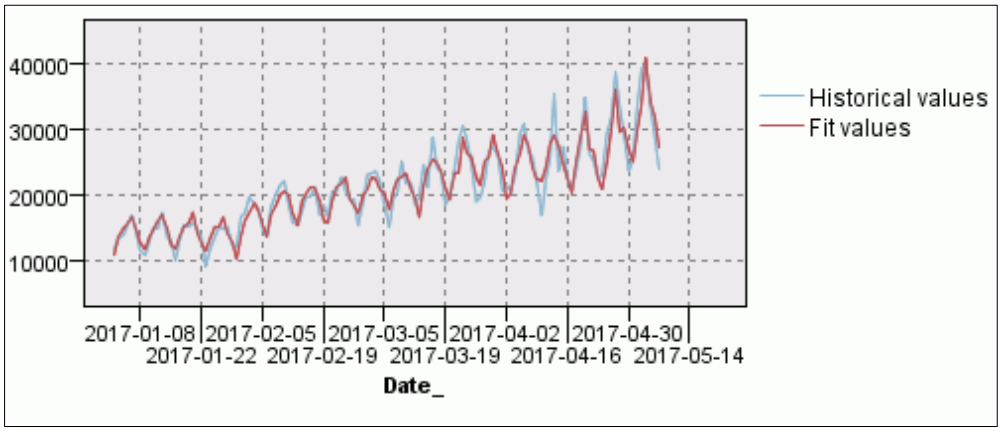
Unit objectives

Before reviewing this unit, you should be familiar with the following topics:

- Working with IBM SPSS Modeler (streams, nodes, palettes)
- Importing data (Var. File node)
- Defining measurement levels, roles, blanks, and instantiating data (Type node)
- Examining the data (Table node, Time Plot node)

IBM Training

Time series models



Measuring model performance

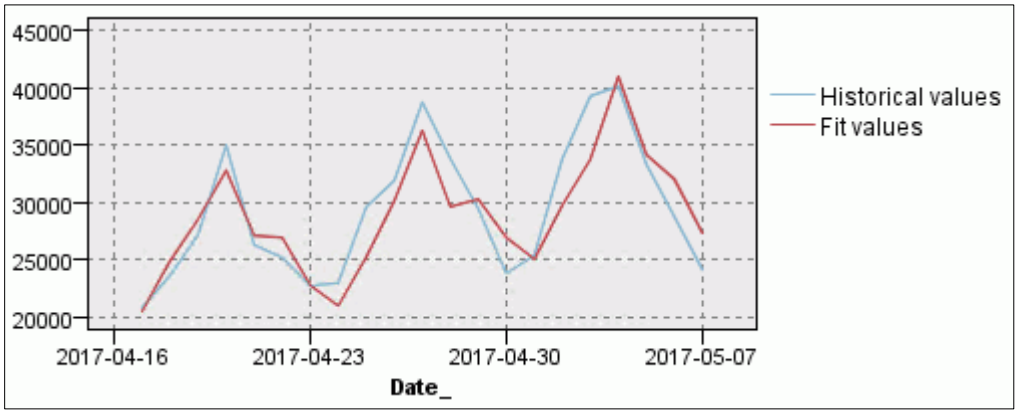
Time series models

Time series models are used to predict future values of a series by analyzing the relationship between the values observed in the historical series and the time of their occurrence. Time series models can be developed using a variety of time series statistical techniques. If there has been any trend and/or seasonal variation present in the data in the past then the time series model can detect this variation, use this information in order to fit the historical data as closely as possible, and in so doing, improve the precision of future forecasts.

Although your goal in time series analysis to create a model to forecast future values, such as sales, unemployment, and so on, before you can focus on the future sales, you must first evaluate how well these fit values match the values in your historical data. If they are close to being the same, you can be fairly confident that the model accurately captured past patterns in the data and can safely be used to forecast values into the future. If this is not the case, then the model probably will not be reliable for making future predictions.

IBM Training

Fit versus historical values

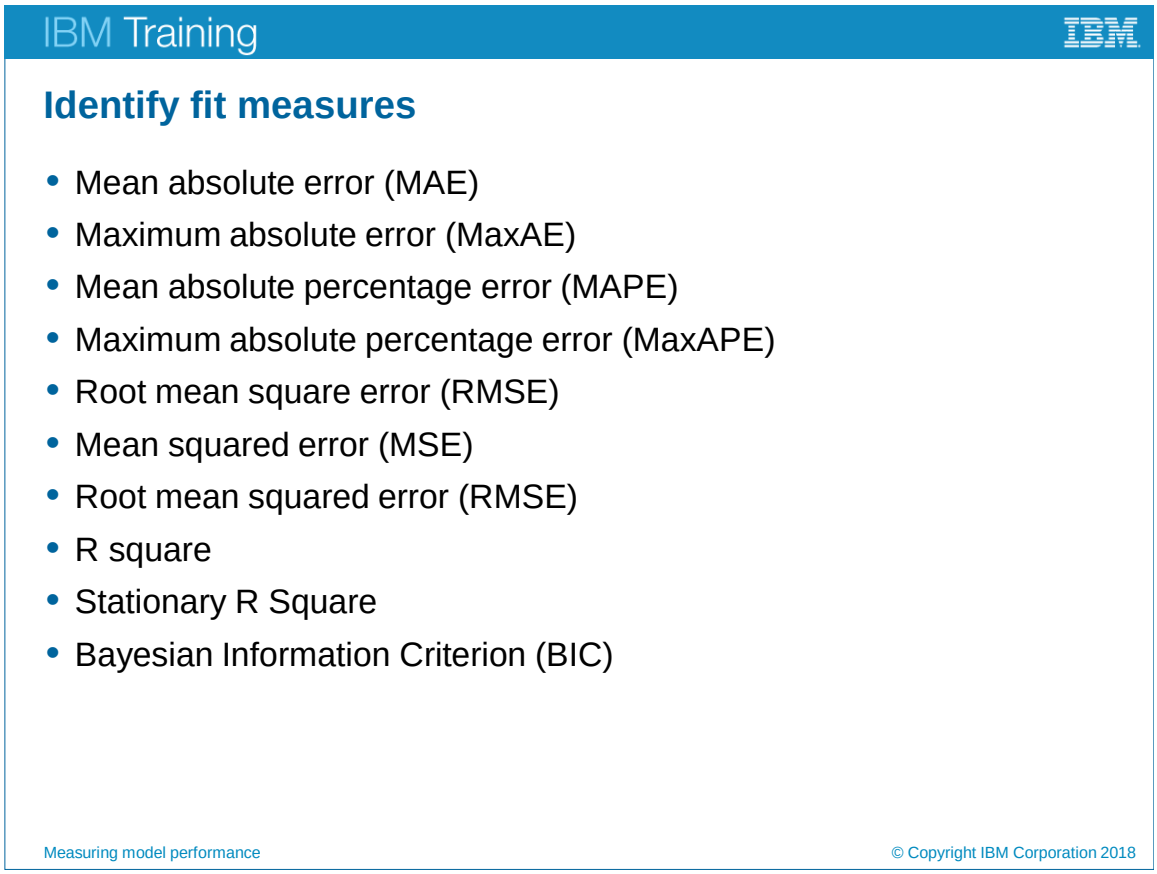


Measuring model performance

Predicted versus historical values

In this graphic, a three-week period of the actual volume of mail delivered by a private mailing company is shown along with the fit values from the model. Note that while they do not match exactly, the model did manage to capture the upward trend in mail deliveries and also the fact that they follow the same pattern each week, with a rise in deliveries in the middle of the week, and fewer deliveries at the beginning and the ending of each week.

However, while a visual look at the results is informative to a point, it is not enough evidence by itself to conclude whether this is a good or a bad model.



Identify fit measures

The results in the generated model provide a number fit measures for each estimated model. Some of these measures will be discussed again and in more detail in the next unit. Here you will be provided with a brief description or definition.

- Mean absolute error (MAE) takes the average of the absolute values of the errors and is often used as the primary measure of fit. MAE is in the same units as the dependent series.
- Maximum absolute error (MAXAE) refers to the largest forecast error, negative or positive. It is expressed in the same units as the dependent series. This measure gives a worst case scenario indication of model performance.
- Mean absolute percentage error (MAPE) is obtained by taking the absolute error for each time period, dividing it by the actual series value, averaging these ratios across all time points, and multiplying by 100. It is sometimes preferred because it is a percentage and thus a relative measure, independent of the units used.

- Maximum absolute error (MaxAE) and Maximum absolute percentage error (MaxAPE) may occur at different series points; for example, when the absolute error for a large series value is slightly larger than the absolute error for a small series value. In that case the maximum absolute error will occur at the larger series value and the maximum absolute percentage error will occur at the smaller series value.
- Mean squared error (MSE) squares the errors, sums them, and then takes the average. Compared to the measures based on absolute error values (MAE, MAPE), this measure penalizes a large error more than it does small ones. For instance, the MAE calculation counts an error of 2 as twice as much as an error of 1, where the MSE calculation squares an error of 2 and thus counts it as 4 times as much as an error of 1. Thus adopting the criterion of minimizing mean squared error implies that you would prefer to have several small deviations from the forecast value rather than one large deviation.
- Root Mean Squared Error (RMSE) is just the square root of the mean squared error. It can be thought of as the standard deviation of the error terms. A small RMSE is preferred since this signifies that the error terms do not have a large spread, that is, the errors concentrate near zero, which means a good fit. The RMSE is in the same units as the dependent series, but because the statistic is based on the MSE, large errors have a higher impact than they have on the mean absolute error (MAE).
- R square is an estimate of the proportion of the total variation in the series that is explained by the model. This measure is most useful when the series shows no trend. R-square can be negative, and has a range of negative infinity to +1. Negative values mean that the model under consideration is worse than the baseline model. Positive values mean that the model under consideration is better than the baseline model.
- Stationary R square compares the stationary part of the model to a simple mean model. This measure is preferable to the usual R square when there is a trend or seasonal pattern. Stationary R-square can be negative, and has a range of negative infinity to +1. Negative values mean that the model under consideration is worse than the baseline model. Positive values mean that the model under consideration is better than the baseline model.
- Bayesian Information Criterion (BIC) enables you to compare different models for the same series. BIC “rewards” models that fit better, that is, produce predicted values that are closer to the dependent series values, while it “penalizes” models that use more parameters. In general, all other things equal, you should prefer the model for a series that has the minimum BIC.

IBM Training

Fit measures illustrated

TIME PERIOD	SALES	MODEL PREDICTION	ERROR	ABSOLUTE ERROR	PERCENTAGE ERROR	SQUARED ERROR
1982-01-01	550	547.6	2.4	2.4	0.4	5.6
1982-02-01	575	573.1	1.9	1.9	0.3	3.8
1982-03-01	595	598.4	-3.4	3.4	0.6	11.6
1982-04-01	640	622.7	17.3	17.3	2.7	298.3
1982-05-01	652	651.0	1.0	1.0	0.2	1.0
1982-06-01	660	676.1	-16.1	16.1	2.4	260.7
1982-07-01	690	698.1	-8.1	8.1	1.2	64.8
1982-08-01	735	721.5	13.5	13.5	1.8	182.4
1982-09-01	750	749.0	1.0	1.0	0.1	0.9
1982-10-01	780	774.2	5.8	5.8	0.7	33.8
1982-11-01	805	800.3	4.7	4.7	0.6	22.4
1982-12-01	818	826.1	-8.1	8.1	1	66.2

Measuring model performance

Fit measures illustrated

The maximum absolute error (MAXAE) is 17.3. For the mean absolute error (MAE), sum the values in the ABSOLUTE ERROR column, and divide that by the number of observations: 83.3 / 12 = 6.945. Thus, on average, the prediction is about 7 off target.

The percentage error divides the absolute error by the observed value, and multiplies by 100 to have that as a percentage. The maximum absolute percentage error (MAXAPE) is 2.7, for the same observation where the maximum absolute error occurred. This does not necessarily have to be the case. For the mean absolute percentage error (MAPE), sum the values, and divide by the number of observations: 12.1 / 12 = 1.08. Thus, on average, you make a 1 percent error in the prediction.

For the mean squared error (MSE), sum the values in the SQUARED ERROR column, and divide that by the number of observations: 951.5/ 12 = 79.3.

The root mean square error (RMSE) takes the square root of the mean squared error: square root of 79.3 = 8.9. Thus, on average, you are about 9 off target. This is 2 higher than the mean absolute error (MAE), because the MSE penalizes large errors.

You can also express the squared errors as a percentage by dividing each squared error by the squared value of the series, averaging these, and then taking the square root, which gives the root mean square percentage error (RMSPE). This statistic is not shown in the table on this slide.

IBM Training

Use diagnostic statistics: Set the stage

TIME PERIOD	SALES	MODEL PREDICTION	RESIDUAL	RESIDUAL AT LAG 1	RESIDUAL AT LAG 2	RESIDUAL AT LAG 3
1982-01-01	550	547.6	2.4	.	.	.
1982-02-01	575	573.1	1.9	2.4	.	.
1982-03-01	595	598.4	-3.4	1.9	2.4	.
1982-04-01	640	622.7	17.3	-3.4	1.9	2.4
1982-05-01	652	651.0	1.0	17.3	-3.4	1.9
1982-06-01	660	676.1	-16.1	1.0	17.3	-3.4
1982-07-01	690	698.1	-8.1	-16.1	1.0	17.3
1982-08-01	735	721.5	13.5	-8.1	-16.1	1.0
1982-09-01	750	749.0	1.0	13.5	-8.1	-16.1
1982-10-01	780	774.2	5.8	1.0	13.5	-8.1
1982-11-01	805	800.3	4.7	5.8	1.0	13.5
1982-12-01	818	826.1	-8.1	4.7	5.8	1.0

Measuring model performance

Use diagnostic statistics: Set the stage

A satisfactory fit of a model is a necessary but not a sufficient condition to use a model to forecast future values.

Think of a times series that has a trend and a seasonal component, but the model used only captured the trend and nevertheless showed a close fit. For example, a time series is comprised of monthly data and has a periodicity of one year. Thus, this year's value in, say, June, resembles last year's value in June, and both observed values will differ from the predicted values, because those are based on the trend component only. Now the model's forecast for next year's June will only reflect the trend too, and miss the seasonal effect for June. Therefore, the forecast will be misleading and will appear to be erroneous once the data for next year's June are observed.

Then, how can you tell, despite a close fit, that you misspecified the model? The key to answer this question lies in the residuals of the series. (Note: ""error" and "residual" are used interchangeably in time series analysis, both refer to the same: the difference between the observed value and the predicted value.)

Suppose that the residual at time t-1 does provide information about the residual at time t, specifically that there is a positive correlation between the residual at time t and time t-a. In the table on this slide, that would be a positive correlation between the columns labeled RESIDUAL and RESIDUAL AT LAG 1 (LAG 1 refers to time t - 1). Now suppose that you underestimate the observed value at time t-1. Given the positive correlation, you know that the observed value will most likely be under estimated too at time t. And if you would use the model to forecast the series' value at time t+1, you will most likely also underestimate the true value at time t+1, making the forecast useless.

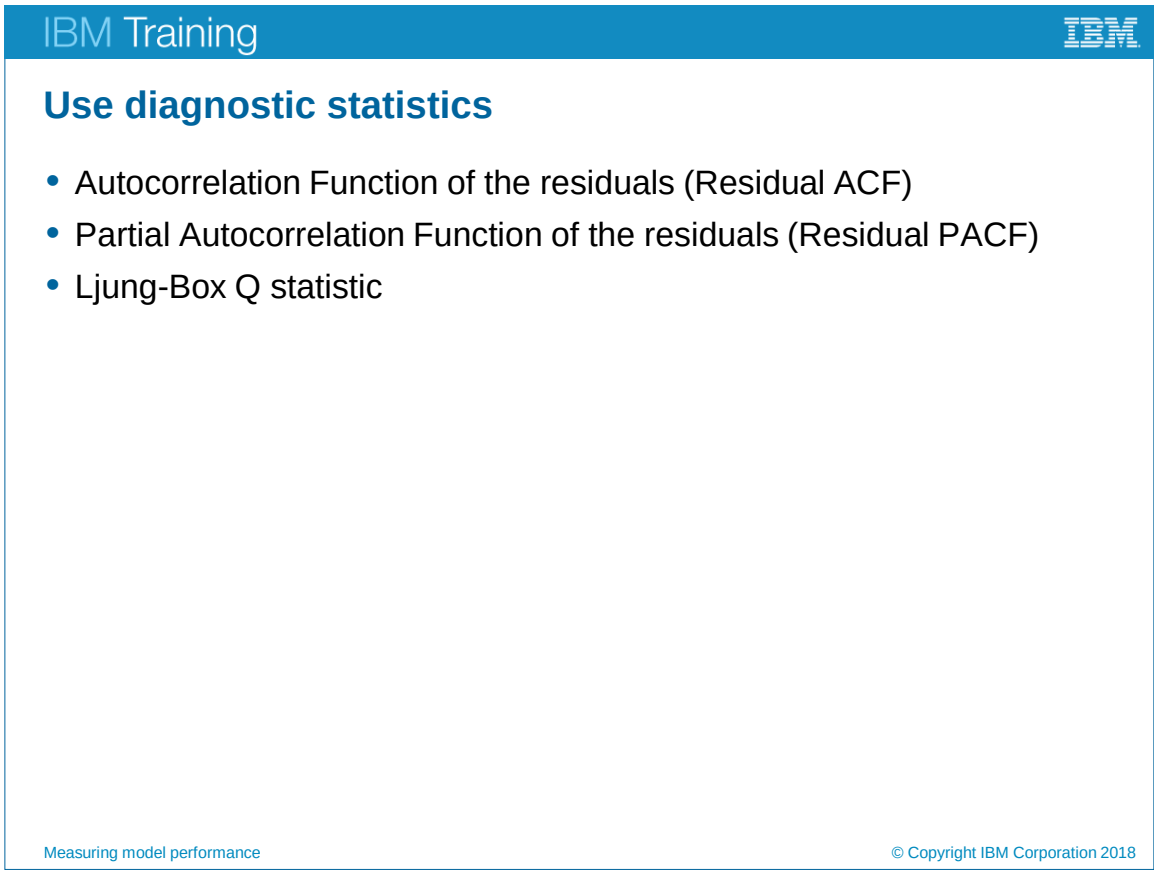
In short, a residual at time t-1 should not convey information about the residual at time t. Referring to the table, the correlation between RESIDUAL and RESIDUAL AT LAG 1 should be 0.

In the same reasoning, the correlation between RESIDUAL and RESIDUAL AT LAG 2 should be 0 too, because if there would be, say, a positive correlation, when you under estimate the value at time t-2, you expect that the value at time t will be under estimated too. And so will the forecast for time t+2.

Think of the example of a monthly series with a trend and periodicity of one year, and a model that captures trend only. The residuals at time t will then be correlated with the residuals at time t-12, indicating a seasonal component.

Thus, a time series model that is well specified and captures all of the non-random variation, such as trend and seasonality, will show residuals that are not correlated with itself over time.

The correlations of the residuals with itself are referred to as "residual autocorrelations", thus to establish whether a model is correctly specified the residual autocorrelations at different time lags are examined. For a well specified model these should all be 0.



Use diagnostic statistics

There are three statistics that test whether residuals are correlated:

First, the residual autocorrelations should all be 0. The autocorrelation function (ACF) provides all these correlations and will show whether the residual autocorrelation at a specific lag is different from 0.

Besides examining the residual autocorrelation at a specific lag, you can examine the net correlation between the residuals at that lag. For example, it could well be that the residual at time t-2 provides information for the residual at time t-1, and that the residual at time t-2 also provides information for the residual at time t, so the correlation between the residuals at time t and time t-1 could be spurious, caused by the residual at time t-2. The partial auto correlation function (PACF) will pick up such patterns. For a correctly specified model, this function should produce partial autocorrelations which are not significantly different from 0, at each lag.

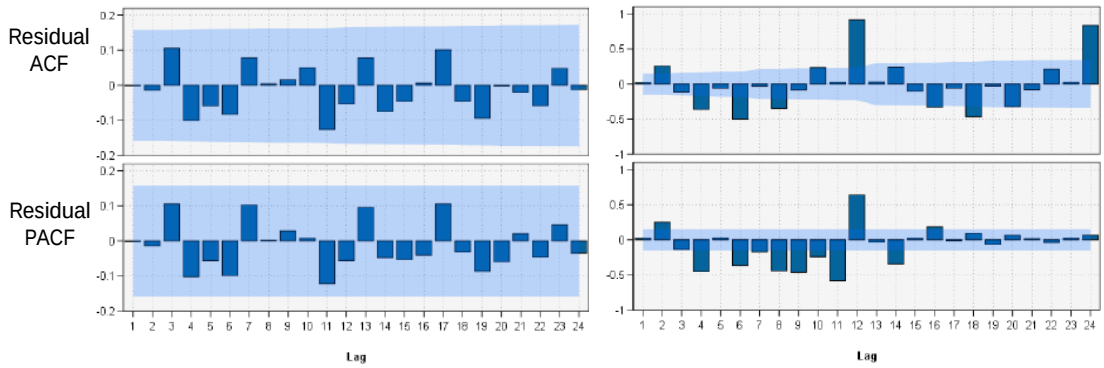
The Ljung-Box Q statistic tests whether the residuals up to a certain lag, say k, are uncorrelated. As such it provides one overall test for any significant autocorrelations at the given lag or fewer, rather than examining many autocorrelation tests, each at the 0.05 level. Unlike other statistical tests, you want the Ljung-Box Q statistic to be non-significant (have a significance value above 0.05).

In computing Box-Ljung Q statistic, IBM SPSS Modeler fixes the number of lags to be tested to 18. Technically, the Ljung-Box statistic follows a Chi-square distribution with degrees of freedom equal to the number of lags tested minus the number of parameters that was estimated in the model (thus, the degrees of freedom equals 18 - the number of estimated parameters).

The results of the ACF and PACF plots and the Ljung-Box test should be consistent and all should point to a random pattern in the residuals. When one or more of these statistics indicate that the model is misspecified, you should investigate the model further and most likely respecify it.

IBM Training

ACF and PACF illustrated



well specified model

incorrectly specified model

Measuring model performance

ACF and PACF illustrated

This slide illustrates ACF and PACF of a correctly specified model and an incorrectly specified model.

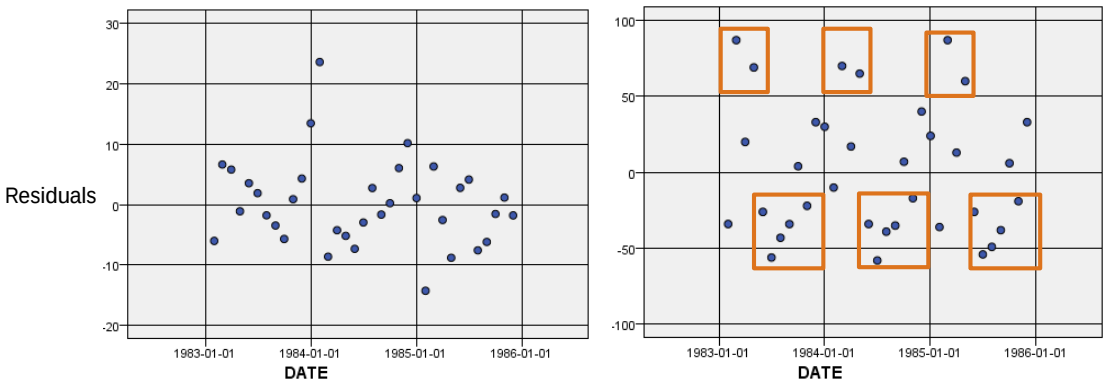
The autocorrelations are graphically represented by bars. The bars are on the negative or positive side, indicating a negative or a positive autocorrelation, respectively.

The shaded area represents the 95% confidence limit for the autocorrelations, when you expect an autocorrelation of 0. In the figure on the left, the autocorrelations stay within the band width, thus they are not significantly different from 0. In the figure on the right, the autocorrelations at lag 12 are outside the confidence bands, both in the ACF and PACF, so these autocorrelations are significantly different from 0.

Note that the spike for the incorrectly specified model occurs at lags 12 and 24 (for the residual ACF). Thus, the residual at time t-12 carries information about the residual at time t. This points to seasonality in a time series where data represents months.

IBM Training

Box-Ljung Q statistic illustrated



well specified model

Significance Lljung-Box Q statistic = 0.6

incorrectly specified model

Significance Lljung-Box Q statistic = 0.0

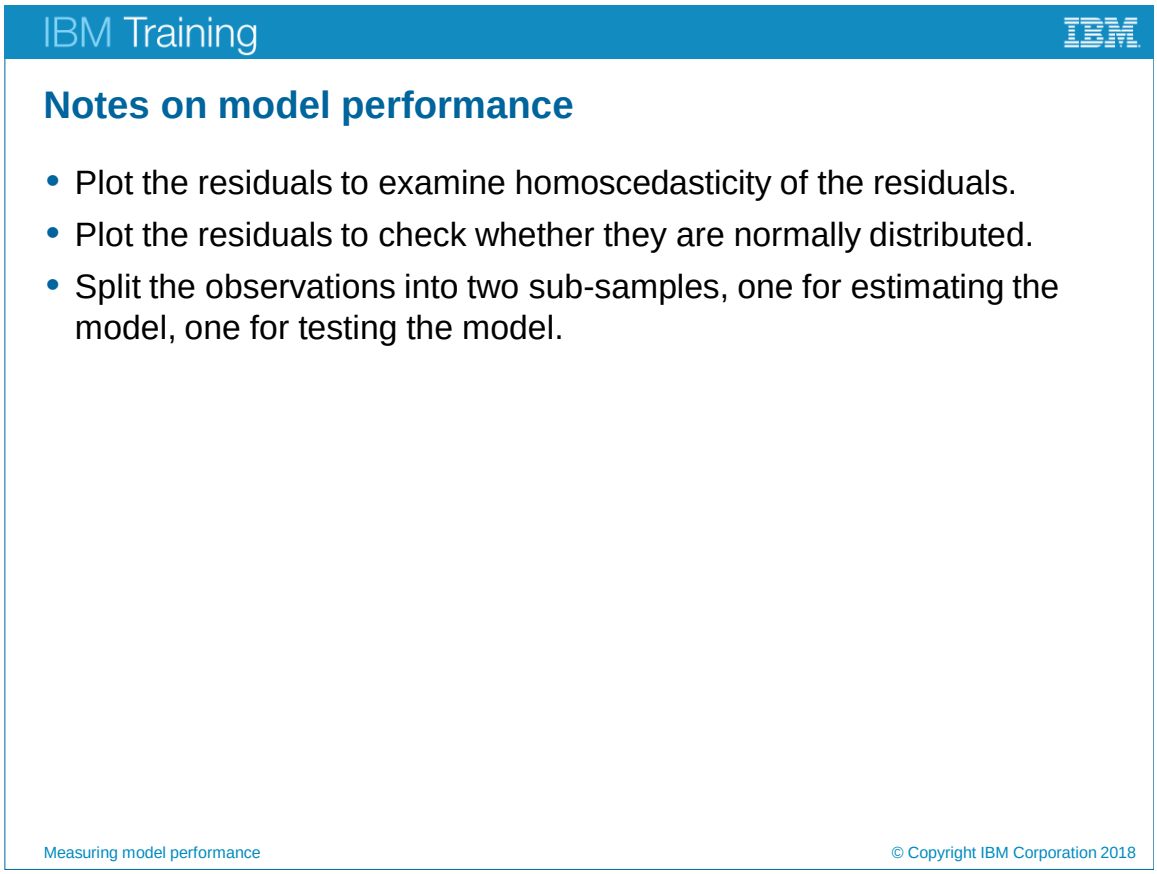
Measuring model performance

Box-Ljung Q statistic illustrated

This slide shows a plot of the residuals and Box-Ljung Q statistic for a correctly and incorrectly specified model.

Box-Ljung Q is not significant for the correctly specified model, whereas it is significant for the incorrectly specified model.

Notice that there is pattern in the residuals in the figure on the right, marked by the square boxes in the plot. If you know the residual at time t - 12, you can fairly well estimate the residual at time t.



Notes on model performance

Prior to forecasting into the future you should be sure that the model fit is satisfactory and the model is well specified. On top of fit measures, such as Mean Absolute Error or Root Mean Square Error, and diagnostic statistics such as Ljung-Box Q, you can run additional checks:

- Plot the residuals to see whether they are homoscedastic (show constant variance over time). It is desirable for the variance of the error to be fairly constant over time. If the error has a tendency to vary more or less as the series progresses, then the robustness of the time series model is diminished. For example, if the variance of the error increases towards the end of the series then this might suggest that the model is going to forecast badly beyond the historic series. If the variance of the error changes over time, this is known as heteroscedasticity.
- Another test which is sometimes applied to time series models is whether the errors are normally distributed. It is preferable for the errors to be normally distributed for some models (ARIMA, Regression). You could run a Histogram node on the residuals to see whether the residuals are normally distributed.

- A final check on model performance is to split the time observations into two sub-samples, one to derive the model fit and the second sample to test the forecasting performance of the model. For example, if you have six years of monthly data, it may be useful to divide this data into two sub-samples: the first sub-sample would be used to estimate the time series model and the second sub-sample to test the performance of forecasts. Thus, monthly data for 2010 through 2015, the first five years of data, can be used to develop and estimate the model, and the last year ((2016, months 1 to 12) can be reserved to test the model. The idea behind this is that by not using the actual 2016 data during model estimation, you are using the time series model to forecast this year. In other words you use separate data partitions to estimate and test the model. Since, in fact, you do know the actual values for 2016, you are able to immediately assess how well your model has forecasted the last year of the data. If errors for both the estimation and the forecast (holdout) sub-samples are low then this is a good indication that your model will predict well to the near future. In order to implement this approach, it is important to have a sufficient number of time periods in your data set since part of the data will not be used during the initial model estimation phase.

  Typically, you use most of the data for model estimation and just one season (if there is seasonality) for validation. This is because when the model is used for true, future forecasts, you will typically forecast only a short interval into the future, so there is no reason to test the model on many time periods.

  Splitting the dataset into an estimation period and a testing period will be discussed later in this course.

IBM Training

Demonstration 1

Evaluate the validity of a time series model

Demonstration 1: Evaluate the validity of a time series model

Demonstration 1: Evaluate the validity of a time series model

Purpose:
You have created a time series model of the car sales data you have collected since January, 1982. Before you use the model to forecast future sales, you will run several tests to make sure that it is a satisfactory model.

Stream file: **unit_3_demonstration_1_start.str**

Folder: **C:\Training\0A028\03-Measuring_Model_Performance\Start**

Task 1. Start and configure IBM SPSS Modeler 18.1.1.

1. From the **Start** menu, expand **IBM SPSS Modeler 18.1**, and then click **IBM SPSS Modeler 18.1**.

   Note: IBM SPSS Modeler 18.1.1 displays "IBM SPSS Modeler 18.1" in the Start menu and program name.
2. Click **Cancel** to close the **Welcome** dialog box.

   If you have already set the working directory in a previous demonstration or exercise, you can skip to Task 2.
3. From the **File** menu, click **Set Directory**.
4. Beside **Look in**, navigate to the **C:\Training\0A028** folder, and then click **Set**.

Task 2. Open the stream.

1. From the **File** menu, click **Open Stream**.
2. Navigate to **03-Measuring_Model_Performance\Start**, and then double-click **unit_3_demonstration_1_start.str.**

Task 3. Plot the residuals.

For a model to be satisfactory, the residuals should show constant variance over time. The simplest way of verify whether this is the case with this model is to use a Time Plot node to plot the data.

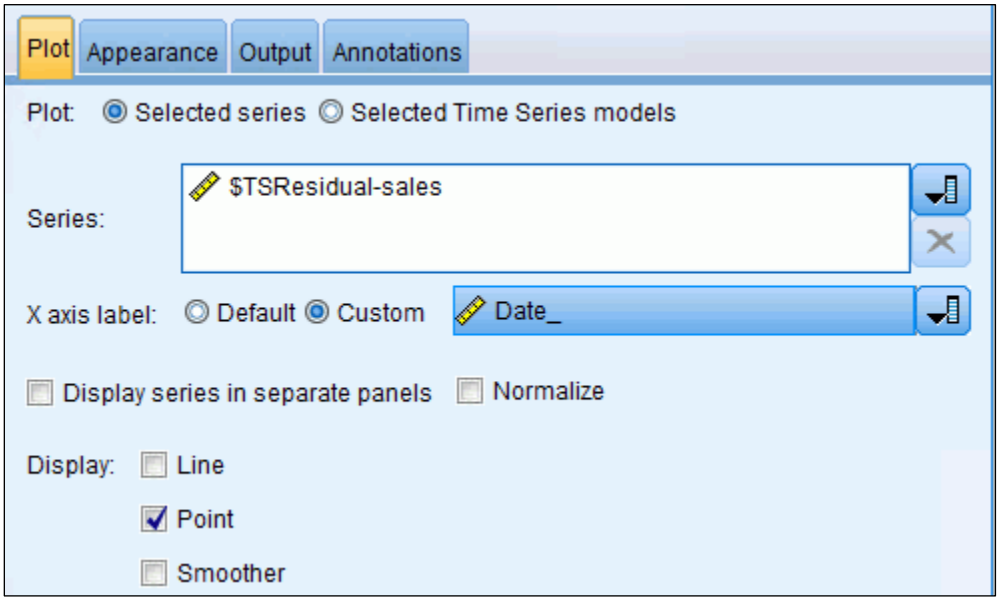
First, you will need to create a new field for the residuals in the model nugget.

1. Edit the **model nugget**.
2. Click the **Settings** tab.
3. Click the **Make Available for Scoring** item.
4. Enable the **Calculate noise residuals** option.
5. Close the **model nugget**.

You are now ready to create the graph.

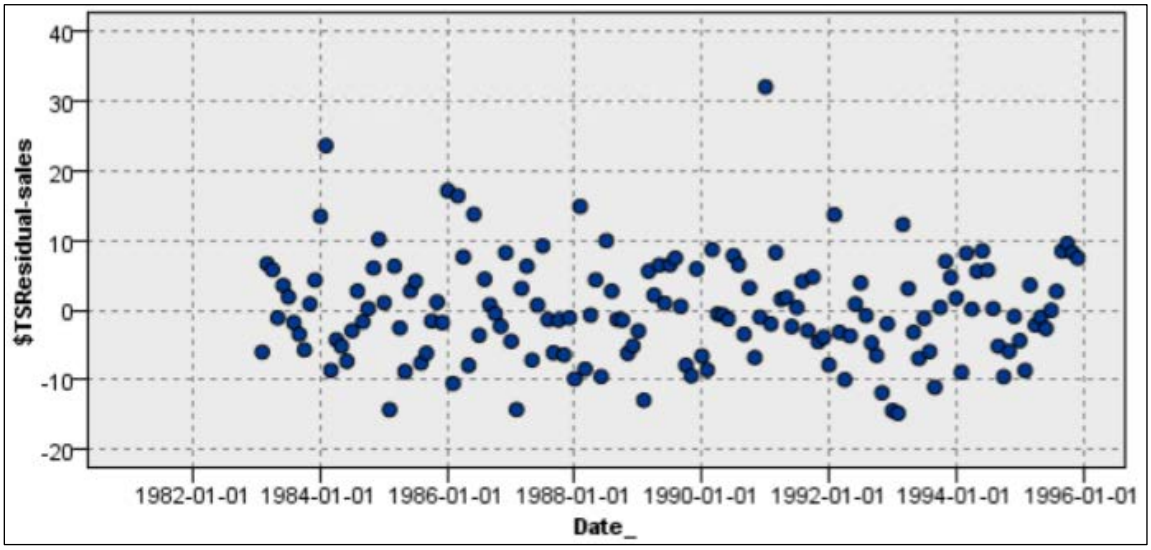
6. From the **Graphs** palette, select the **Time Plot** node, and add it downstream from the **model nugget** node.
7. Edit the **Time Plot** node.
8. Besides **Series**, select **$TSResidual-sales**.
9. Besides **X axis label**, enable the **Custom** option, and then select **Date_**.
10. Clear the **Normalize** option.
11. In the **Display** area, clear the **Line** option.
12. In the **Display** area, select the **Point** option.

The results appear as follows:



13. Click **Run**.

    The results appear as follows:



    The residuals seem to be fairly constant throughout the entire series. There would be cause for concern if the pattern was curvilinear or funnel shaped, but that is clearly not the case here.

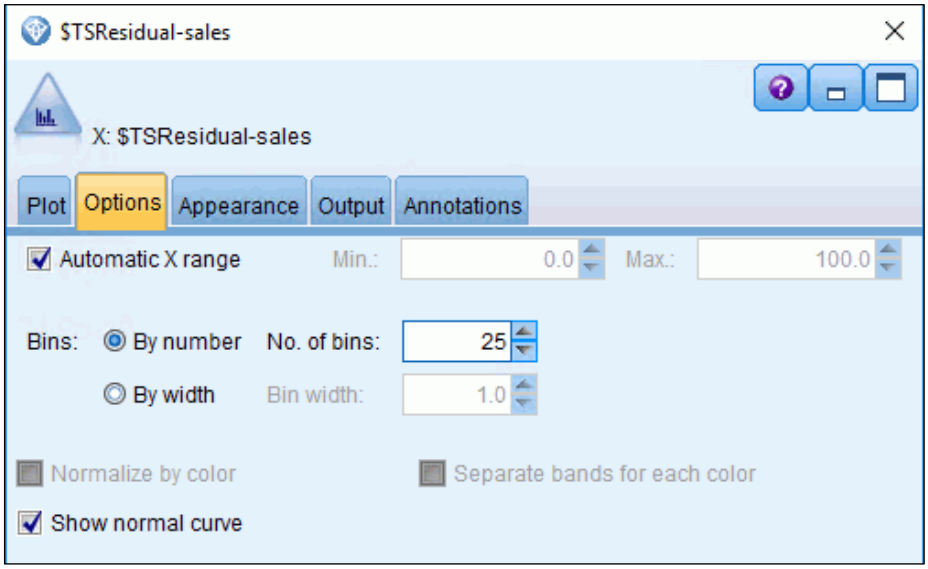
14. Close the **Time Plot** output window.

Task 4. Check if the errors are normally distributed.

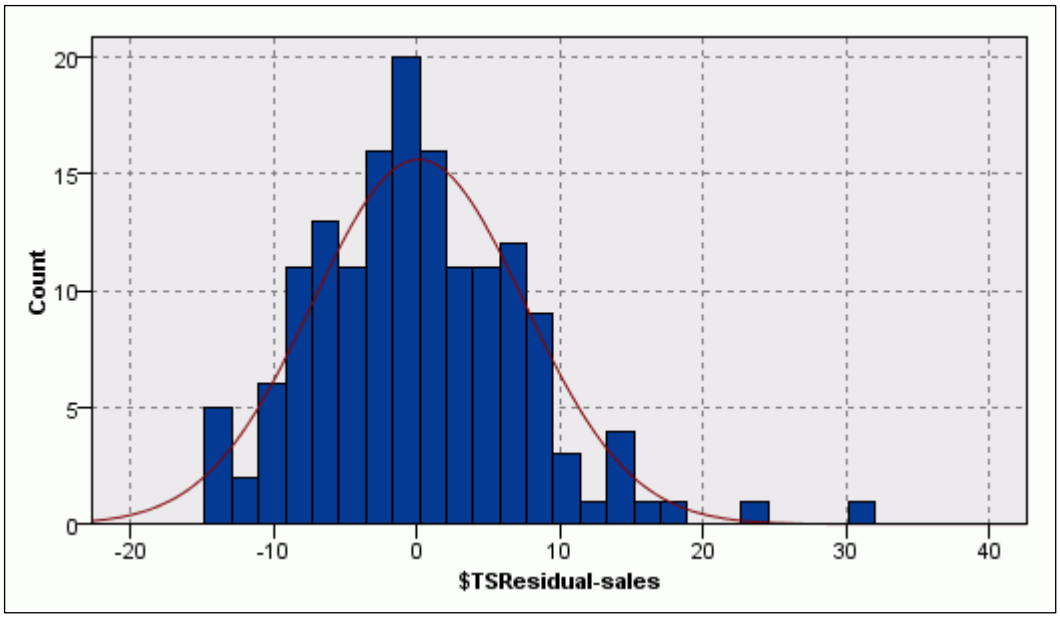
It is preferable for ARIMA and Regression models that the errors are normally distributed. You will create a histogram of the residuals from the current model to see if that is the case.

1. From the **Graphs** palette, select the **Histogram**, and add it downstream from the **model nugget** node.
2. Edit the **Histogram** node.
3. Besides **Field**, select **$TSResidual-sales**.
4. Click the **Options** tab.

5. Select the **Show normal curve** option.
   The results appear as follows:



6. Click **Run**.
   The results appear as follows:



   The distribution appears to be reasonably normal.

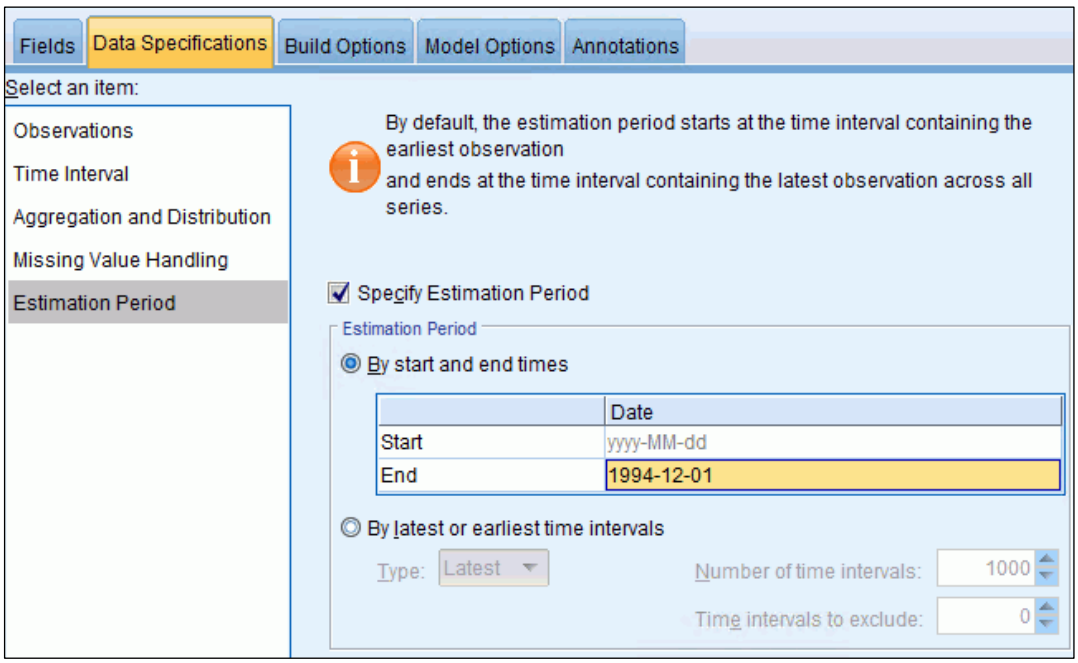
7. Close the **Histogram** output window.

Task 5. Splitting the sample into estimation and forecasting sub-samples.

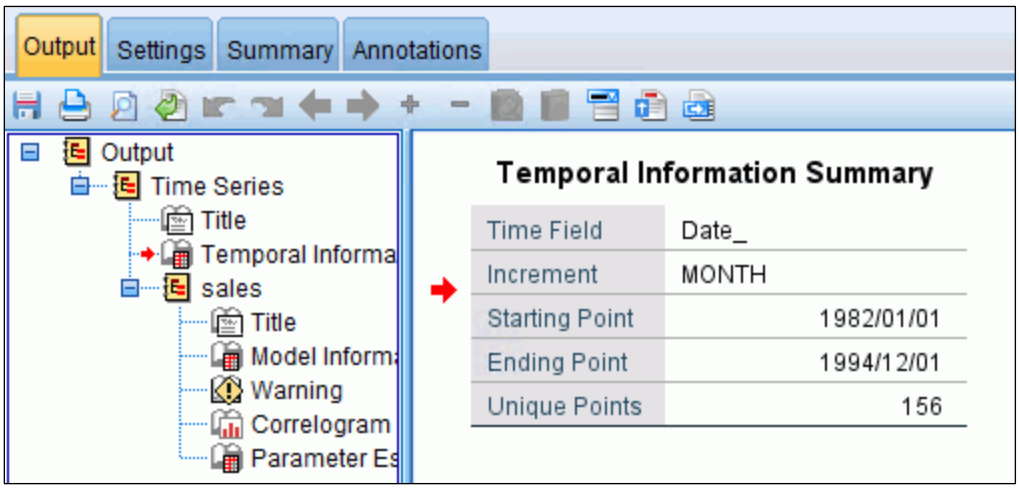
The car sales data contains monthly data on the number of cars sold from 1982 through 1995. Another useful evaluation of the model performance is to develop a model based on the first X (here you will choose 13) years of data and see how well the model forecasts the remaining (here 1) year of data. The final week will form a "hold-out" sample to instantly assess how well the model might forecast beyond the 14th year

1. Edit the **Time Series** node named **sales** (the modeling node, not the model nugget).
2. Click the **Data Specifications** tab.
3. On the left side, click the **Estimation Period** item.
4. Enable the **Specify Estimation Period** option.
5. In the **Estimation Period**, ensure that the **By start and end times** option is enabled.
6. Next to **End**, type **1994-12-01**.

The results appear as follows:



7. Click **Run**.
   The model nugget is updated.
8. Edit the **model nugget**.
9. From the outline pane in the output, select the **Temporal Information Summary** table.
   The results appear as follows:



The results verify that the model only used data through 1994.

10. Click the **Model Information** table.

    The following table presents the output extracted from the Model Information section of the previous model and the current model, combined into a single table for easy comparison.

		1982 - 1994 (13 yrs.)	1982 - 1995 (14 yrs.)
Model Building Method		ARIMA(1,1,0)(0,1,1)	ARIMA(1,1,0)(0,1,1)
No. of Predictors		0	0
Model Fit	MSE	56.501	54.941
	RMSE	7.517	7.412
	RMSPE	1.019	1.000
	MAE	5.714	5.681
	MAPE	0.769	0.760
	MAXAE	31.665	32.025
	MAXAPE	3.938	3.983
	AIC	578.811	622.956
	BIC	584.811	629.043
	R-Squared	0.994	0.994
	Stationary R-Squared	0.324	0.317
Ljung-Box Q(#)	Statistic	13.463	14.214
	df	16.0	16.0
	Significance	0.6	0.6

    The fit statistics from the 13 year data set are remarkably similar with those from the entire data set which suggests that adding an additional year of data does not denigrate the model. In fact, adding data for 1995 actually reduced the MAPE value, which measures the average percent error, from 0.769 to 0.760.

    The results of the analysis also provides you with a table of forecasted values.

11. Click the **Settings** tab, then click **Make Available for Scoring**, and then select **Calculate noise residuals**.
12. Close the **model nugget**.

13. Run the **Table** node and scroll to the 1995 results.

    The results appear as follows:

Table | Annotations

	Date_	$FutureFlag	sales	$TS-sales	$TSLCI-sales	$TSUCI-sales	$TSResidual-sales
149	1994-05-01	0	969.000	962.916	948.275	977.557	6.084
150	1994-06-01	0	947.000	938.229	923.588	952.870	8.771
151	1994-07-01	0	908.000	901.628	886.987	916.269	6.372
152	1994-08-01	0	867.000	866.033	851.392	880.674	0.967
153	1994-09-01	0	815.000	819.651	805.010	834.292	-4.651
154	1994-10-01	0	812.000	821.609	806.967	836.250	-9.609
155	1994-11-01	0	773.000	779.452	764.811	794.093	-6.452
156	1994-12-01	0	813.000	814.669	800.028	829.310	-1.669
157	1995-01-01	0	834.000	838.769	824.128	853.410	-4.769
158	1995-02-01	0	782.000	790.788	776.147	805.429	-8.788
159	1995-03-01	0	892.000	889.072	874.431	903.713	2.928
160	1995-04-01	0	903.000	905.539	890.897	920.180	-2.539
161	1995-05-01	0	966.000	966.917	952.276	981.558	-0.917
162	1995-06-01	0	937.000	939.824	925.183	954.465	-2.824
163	1995-07-01	0	896.000	896.212	881.571	910.853	-0.212
164	1995-08-01	0	858.000	855.470	840.829	870.111	2.530
165	1995-09-01	0	817.000	808.350	793.709	822.991	8.650
166	1995-10-01	0	827.000	816.919	802.278	831.560	10.081
167	1995-11-01	0	797.000	787.830	773.189	802.472	9.170
168	1995-12-01	0	843.000	834.634	819.993	849.275	8.366

    The field sales contains the actual sales, $TS-sales the predictions, and $TSResidual-sales, the error in prediction. The 1995 sales values were held out and not used during model creation. In most cases, predictions were fairly close to the actual values.

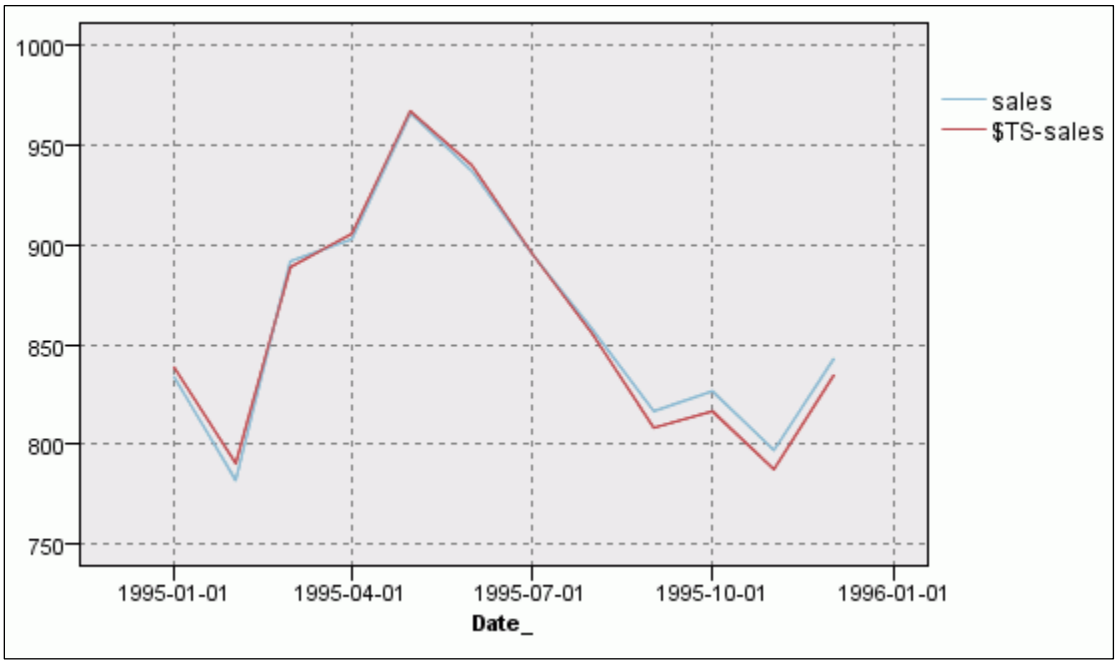
14. Close the **Table** output.

    Now you will use a time plot to compare the actual and predicted values. To make it easier to focus on the 1995 results, before creating the chart, you will select only the records with an index value greater than 156. The index values are displayed in the left column. The 1995 records start with index value 157.

15. From the **Record Ops** palette, select a **Select** node and add it downstream from the **model** nugget.
16. Edit the **Select** node, then type **@INDEX > 156** in the **Condition** box, and then click **OK**. Alternatively, you can use the expression builder to create this expression.
17. Close the **Select** node.
18. From the **Graphs** palette, select the **Time Plot** node, and add it downstream from the **Select** node.
19. Edit the **Time Plot** node.
20. Besides **Series**, select both **sales** and **$TS-sales**.
21. Besides **X axis label**, enable the **Custom** option, and then select **Date_**.
22. Clear the **Display series in separate panels** option

23. Clear the **Normalize** option.
24. Click **Run**.

    The results appear as follows:



    The predictions look very close to the actual values, although toward the end of 1995 it seems the model was consistently under predicting sales. This might not be a good sign beyond 1995 when the forecasts will begin.

25. Close the **Time Plot** output.

    To get a sense of just how accurate the model is in 1995, you will calculate a Mean Absolute Percent Error (MAPE) statistic from the actual and predicted values.

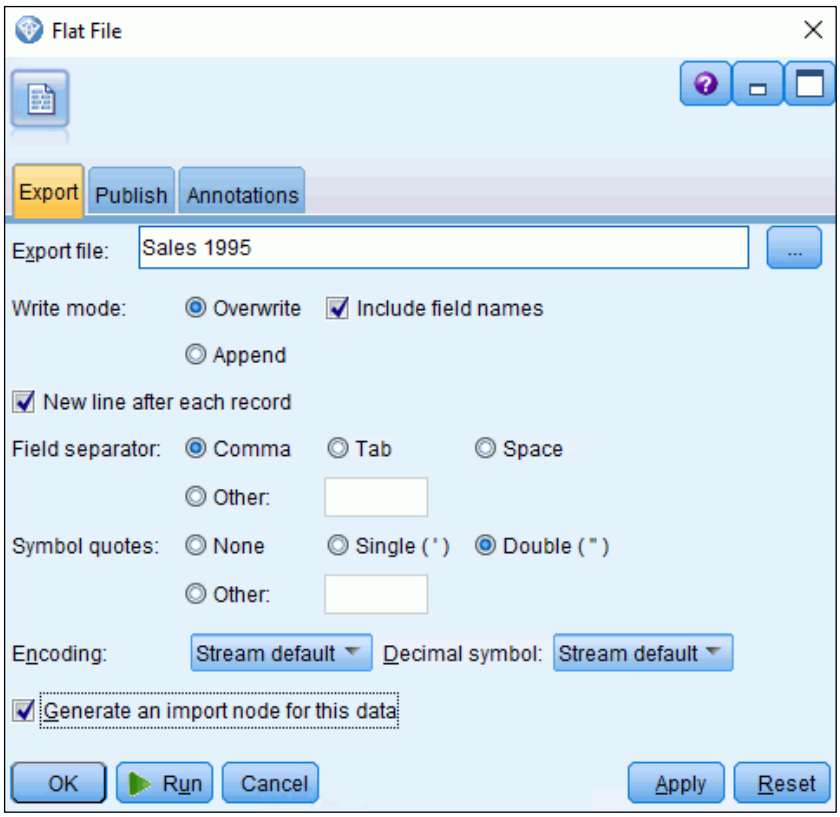
Task 6. Calculating Mean Absolute Percent Error.

The MAPE statistic is not automatically calculated for the hold-out sample, so you will need to calculate it manually.

1. From the **Export** palette, select a **Flat File** node and add it downstream from the **Select** node.
2. Edit the **Flat File** node.
3. In the **Export file** box, type **Sales 1995**.

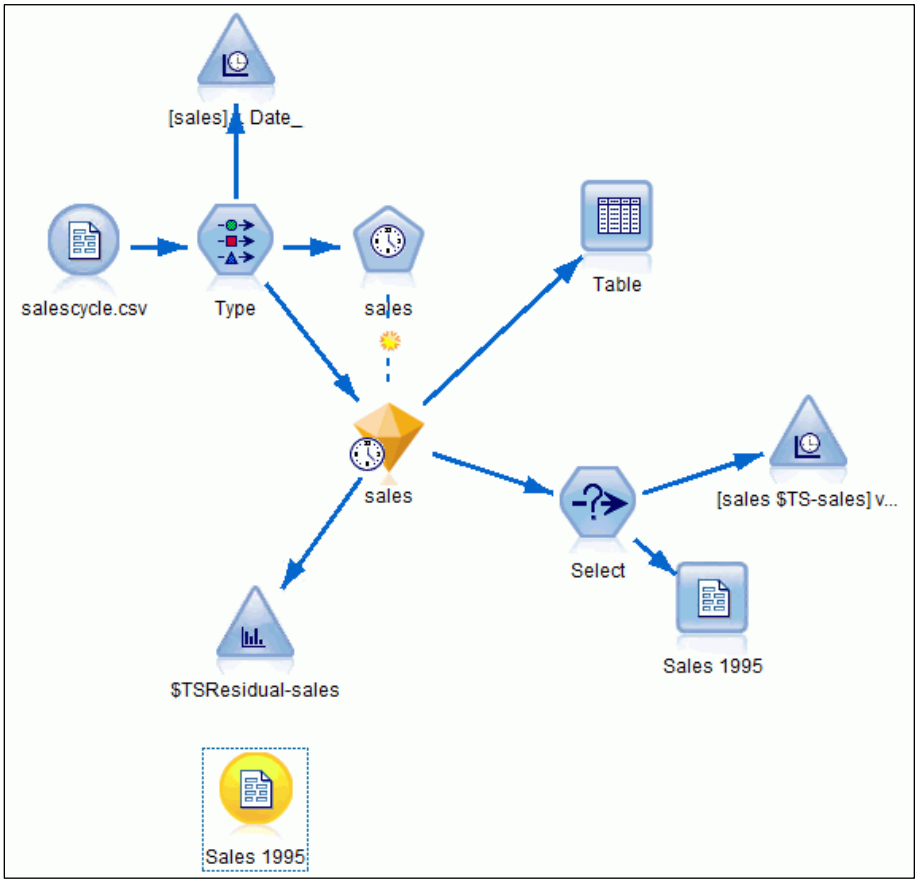
4. Enable the **Generate an import node for this data** box. This option is used to automatically generate a Variable File source node that will read the exported data file.

   The results appear as follows:



5. Click **Run**. If you get an Overwrite warning, click **OK**. The generated node named Sales 1995 will appear in the top-left corner of the canvas.

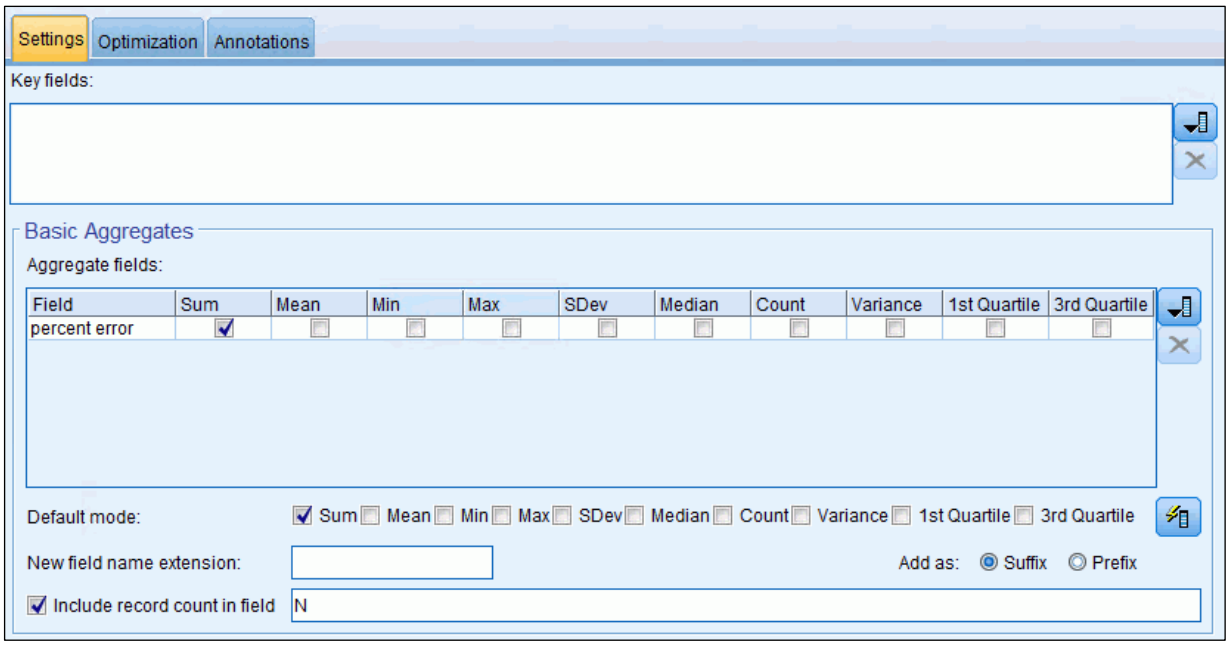
6. Drag the **Source** node named **Sales 1995** to the bottom of the canvas.
   The results appear similar to the following:



7. From the **Field Ops** palette, select a **Derive** node and add it downstream from the **Source** node.
8. Edit the **Derive** node.
   First, you will calculate the absolute value of the error field $TSResidual-sales.
9. In the **Derive field** box name the new field **absolute error**; and then in the **Formula** box type **abs('$TSResidual-sales')**. (Alternatively, use the Expression Builder.)
10. Click **OK**.
    Next, you will calculate the percent error in prediction for each month.
11. From the **Field Ops** palette, select a **Derive** node and add it downstream from the **Derive** node named **absolute error**.
12. Edit the **Derive** node.
13. In the **Derive field** box, name the new field **percent error**, and then in the **Formula** box type **('absolute error' / sales) * 100.**
14. Click **OK**.
    Next, you will use the Aggregate node to sum the values of percent error.

15. From the **Record Ops** palette, select an **Aggregate** node, and then add it downstream from the **Derive** node named **percent error**.
16. Edit the **Aggregate** node.
17. In the **Aggregate fields** area, select **percent error**, and then click **OK**.
18. Select the **Sum** box. Leave all other summary statistics cleared.
19. Enable the **Include record count in field** option and name the field **N**.

    The results appear as follows:

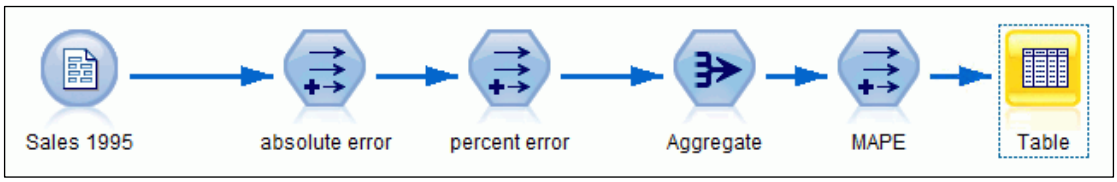


20. Click **OK**.

    Now that you have all the component fields, you will calculate the MAPE statistic for the 1995 predictions.

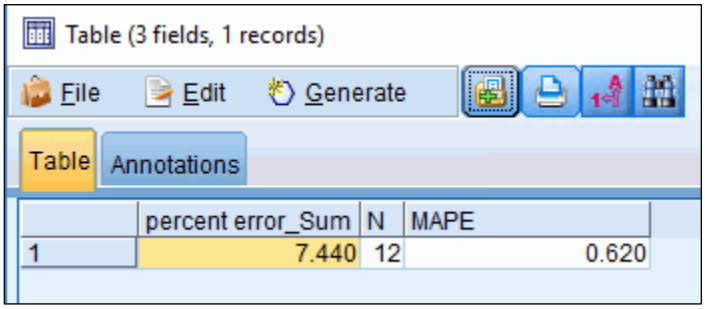
21. From the **Field Ops** palette, select a **Derive** node and add it downstream from the **Aggregate** node.
22. Edit the newest **Derive** node.
23. In the **Derive field** box, name the new field **MAPE**, and in the **Formula** box, type **'percent error_Sum' / N**.
24. Click **OK**.
25. From the **Output** palette, select a **Table** node, and then add it downstream from the last **Derive** node.

    The results appear as follows:



26. Right-click the **Table** node and click **Run**.

    The results appear as follows:



    The results show that on average the 1995 predictions of car sales numbers by the 1982 - 1994 model were only off by 0.620%, which is slightly better than the MAPE for the model based on historical data (0.769%). The fact that the model performed so well with hold-out data strongly suggests that it will produce reliable forecasts in the subsequent year.

27. Close the **Table** output.

    This completes the demonstration. You will create a clean state for the exercise.

28. From the **File** menu, click **Close Stream**. Click **No** when asked to save changes.

29. From the **File** menu, click **New Stream**.

    Leave IBM SPSS Modeler open for the exercise.

Results:
You successfully evaluated the quality of the time series model you created for car sales data from 1982 - 1995.

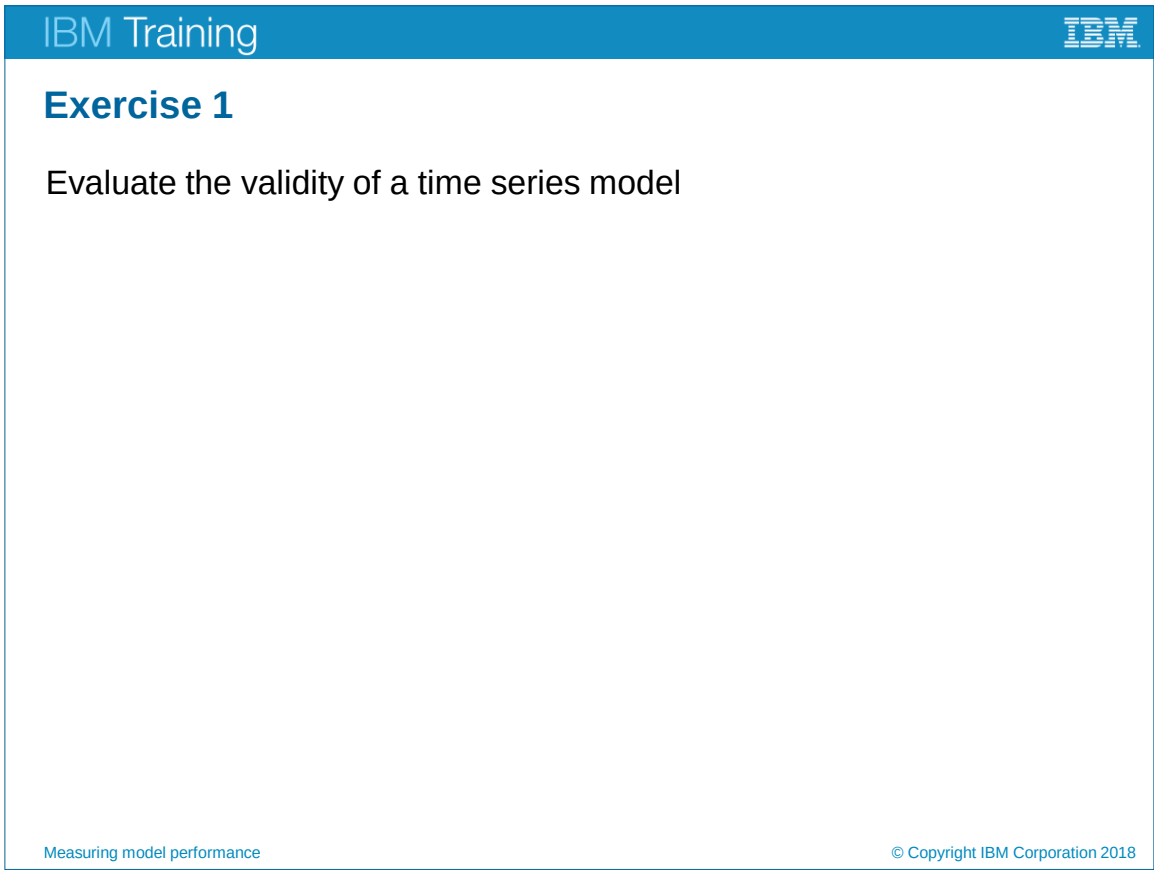
You will find the completed stream in the following folder:
C:\Training\0A028\03-Measuring Model Performance\Solutions

IBM Training

Unit summary

- Discuss various ways to evaluate model performance
- Evaluate model performance of an ARIMA model
- Test a model using a holdout sample

Unit summary



Exercise 1: Evaluate the validity of a time series model

Exercise 1: Evaluate the validity of a time series model

You created a time series model for a private mailing company, and now want to evaluate the model to make sure it is satisfactory before the company uses it to forecast the demand for its delivery service. The mailing company needs to be confident that the model works so it can arrange for the appropriate staffing levels to meet the demand each day.

Stream file: **unit_3_exercise_1_start.str**

Folder: **C:\Training\0A028\03-Measuring_Model_Performance\Start**

Task 1. Open the stream.

- From the **File** menu, click **Open Stream**.
- Navigate to the **C:\Training\0A028\03-Measuring_Model_Performance\Start** folder, and then double-click **unit_3_exercise_1_start.str**.

Task 2. Create a time plot of the residuals.

- Add a **Time Plot** node downstream from the **model nugget**.
- In the **Series** box, select **parcel**.
- Display **Date**_ as the **X axis label.** Do not normalize the graph.
- What patterns do you see in the residuals?

Task 3. Create a hold-out sample.

- In the **Time Series** modeling node, leave out week **18** from the estimation period. Hint: the last seven records to the dataset represent week 18.
- Create a time plot of which displays both the historical and forecasted series.
- Exit **IBM SPSS Modeler** without saving anything.

For more information about where to work and the exercise results, refer to the Tasks and results section that follows. If you need more information to complete a task, refer to earlier demonstrations for detailed steps.

Exercise 1: Tasks and results

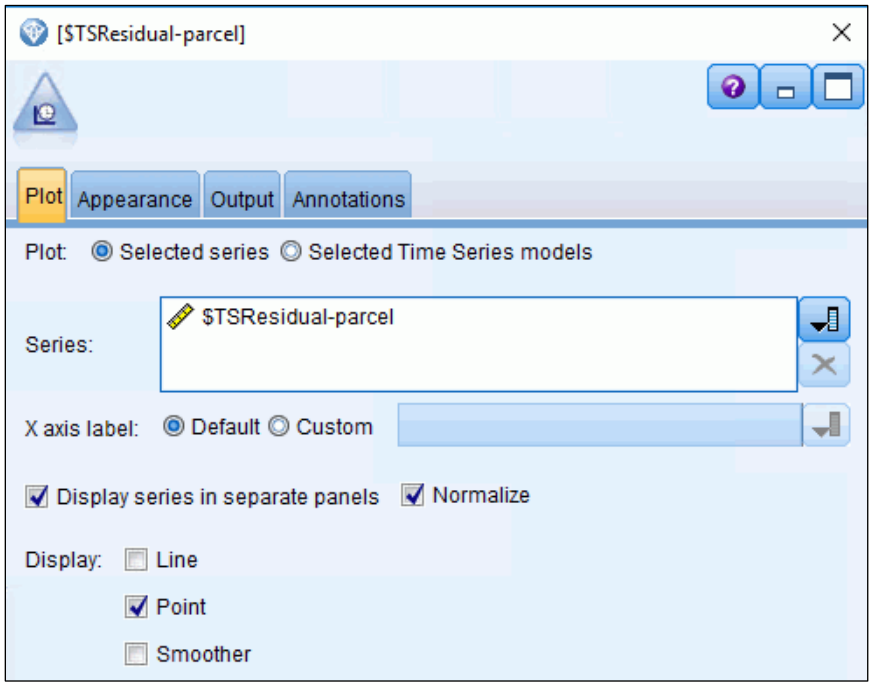
Task 1. Open the stream.

- From the **File** menu, click **Open Stream**.
- Navigate to the **C:\Training\0A028\03-Measuring_Model_Performance\Start** folder, and then double-click **unit_3_exercise_1_start.str**.

Task 2. Create a time plot of the residuals.

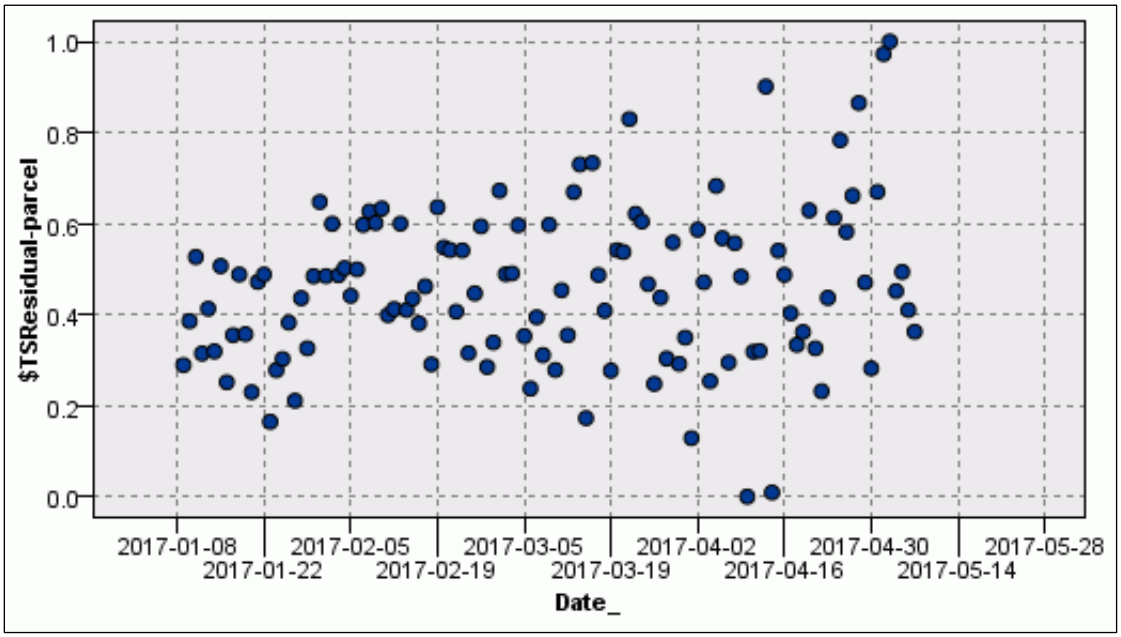
- From the **Modeling** palette, **All** sub palette, add a **Time Plot** node downstream from the **model nugget**.
- In the **Series** box, select **$TSResidual-parcel**.
- Besides **X axis label**, enable the **Custom** option, and then select **Date_**.
- Disable the **Normalize** option.
- In the **Display** area, enable **Point** and disable **Line**.

  The results appear as follows:



- Click **Run**.

  The results appear as follows:



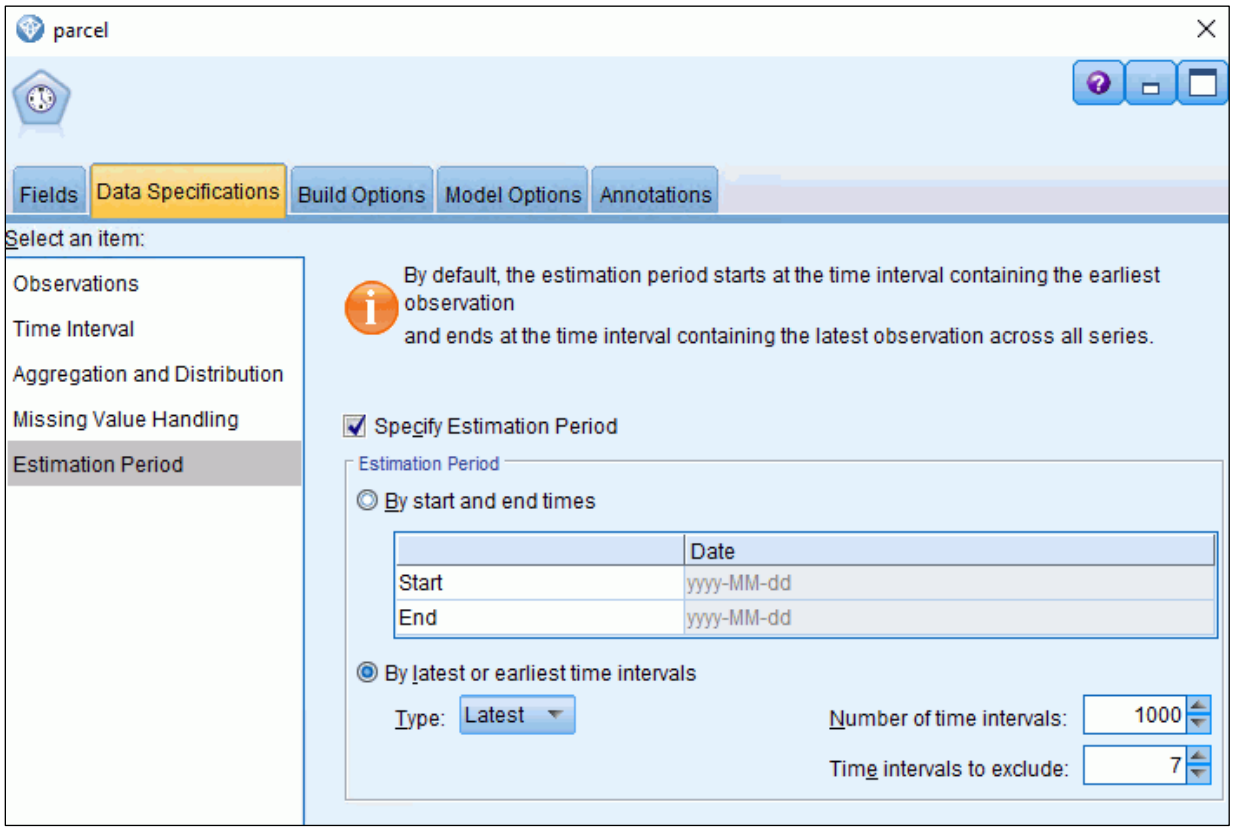
- What patterns do you see in the residuals? The errors do not appear to be random. Notice that the variance in the error increases further into the series. The errors are therefore heteroscedastic and this might affect the performance of the time series forecast.

Task 3. Create a hold-out sample.

- Edit the **Time Series** modeling node.
- Click the **Data Specifications** tab and then click **Estimation period**.
- Enable the **Specify Estimation Period** option.

- In the **Estimation** area, enable the **By latest or earliest time intervals** option. Ensure that **Type** is set to **Latest** and set **Time intervals to exclude** to **7**.

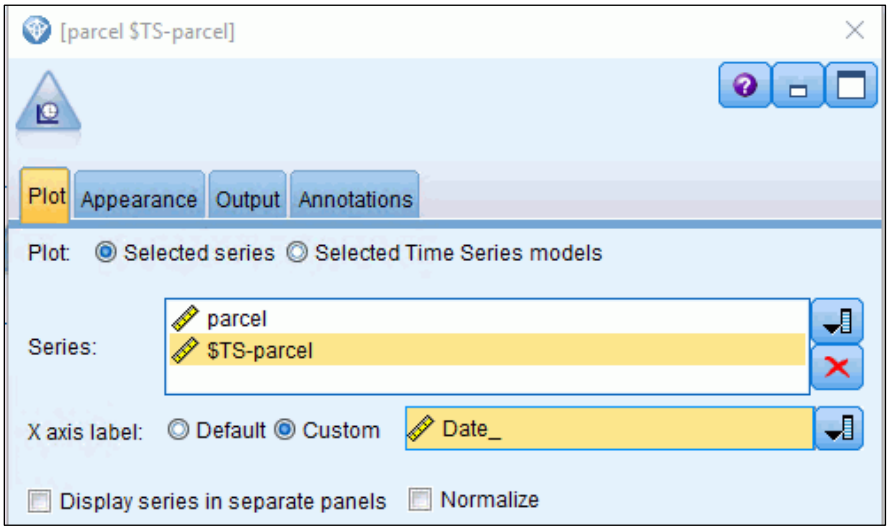
  The results appear as follows:



- Click **Run**.
- Edit the **Time Plot** node downstream from the **model nugget**.
- In the **Series** box, select **parcel** and **$TS-parcel** and then remove **$TSResidual-parcel**.
- Besides **X axis label**, enable the **Custom** option is enabled, and **Date_** selected.
- Ensure the **Normalize** option is disabled.

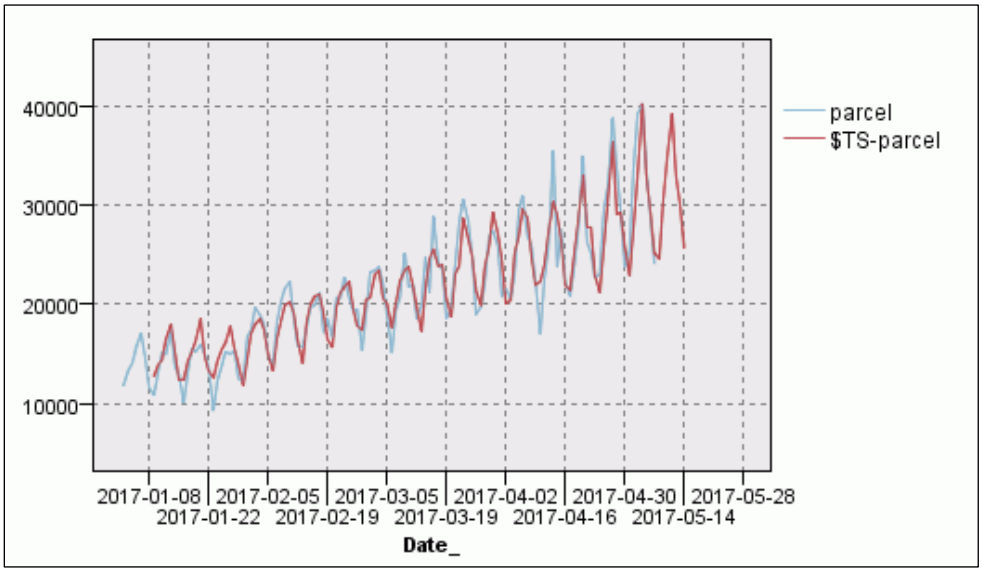
- Disable the **Display series in separate panels** option.

  The results appear as follows:



- Click **Run**.
- Create a time plot of which displays both the historical and forecasted series.

  The results appear as follows:



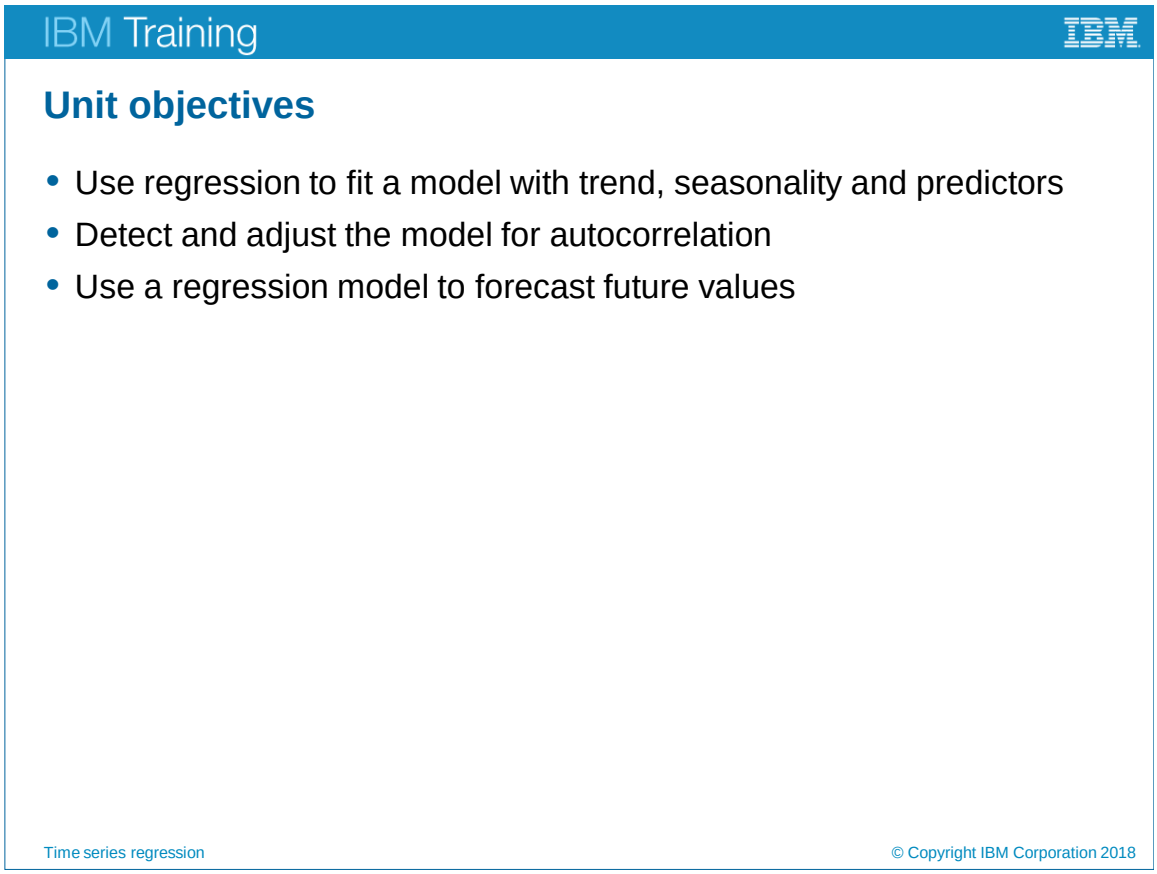
- Do the forecasts for the hold-out sample match the actual values? While in general the fitted data follows the pattern of the original data, it does not fit the peaks and troughs well.
- Close the **Time Plot** output window.
- From the **File** menu, click **Exit** and then exit **IBM SPSS Modeler** without saving.

You will find the completed stream in the following folder:
C:\Training\0A028\03-Measuring_Model_Performance\Solutions

IBM Training

Time series regression

IBM SPSS Modeler (v18.1.1)



Unit objectives

Before reviewing this unit, you should be familiar with the following topics:

- Working with IBM SPSS Modeler (streams, nodes, palettes)
- Importing data (Var. File node)
- Defining measurement levels, roles, blanks, and instantiating data (Type node)
- Examining the data (Table node, Time Plot node)
- Evaluating time series models, using time plots, autocorrelation plots, and regression model fit statistics
- Using the model nugget to score data
- The Regression model

IBM Training

Regression analysis

Time series regression

Regression analysis

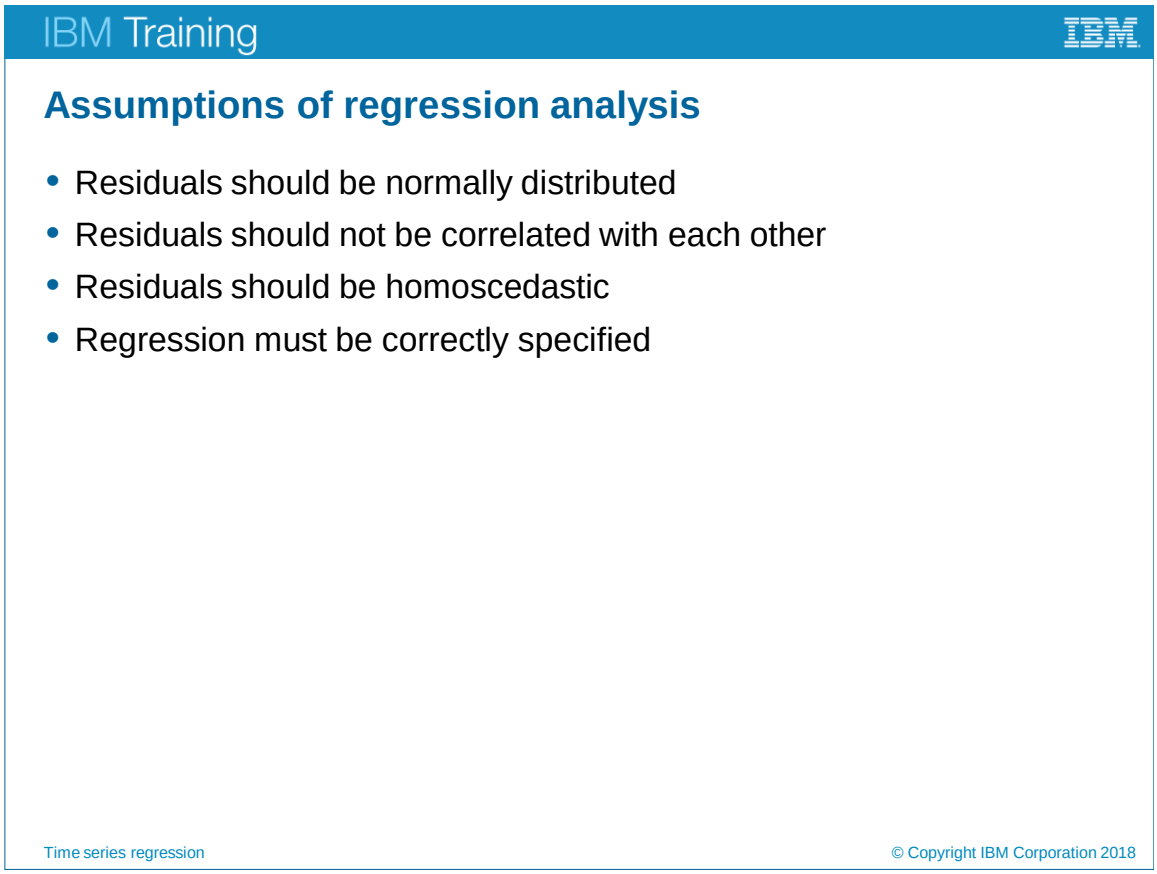
Regression analysis is a statistical tool that can produce predictions and explanations of your data. The basic principle behind regression is to use one or more fields to predict another field of interest using a linear model. The terminology used in regression is that predictor fields are known as independent fields and the field that is being predicted is known as the dependent field.

Regression is often used when the independent field(s) are something other than time. There are many applications in which regression analysis is employed to make predictions that do not involve time. For example, if you wanted to predict an individual's current wage, then a number of other factors can be used as independent fields. These could include things such as the person's age, education, region of the country in which they live, how much training they have had, whether they are male or female, etc. In this example, regression is used to predict across individuals. To develop such a model it would be necessary to collect information on individuals and analyze their wages at a particular moment in time.

A second example would be where regression is used to predict the waste produced by several industrial centers at a given point in time. The dependent field in this example would be waste levels. The possible independent predictor fields might include the amount of industrial trade, retail trade and size of center. To develop a regression model for waste it will be necessary to collect information on several centers at a specific moment in time, or assume that the time differences are not relevant to the analysis (and so use data from different times). Here regression analysis is being used to predict across industrial centers. In this form of regression, time is not taken into account.

For the purposes of time series analysis, regression is used to predict across time and the data should be collected for a number of consecutive time periods. An example would be to develop a regression model predicting a firm's sales over time. There may be important independent fields that can predict the sales performance over time, such as the introduction of new products, advertising expenditure and pricing policies. Information would need to be collected over time to develop this type of regression model.

Although regression was not developed for time series data, with appropriate methods and cautions, regression models can be used to make forecasts.



Assumptions of regression analysis

The first main assumption of regression is that the residuals/errors should be normally distributed. This assumption can be tested using the Histogram node to create a histogram of the residuals.

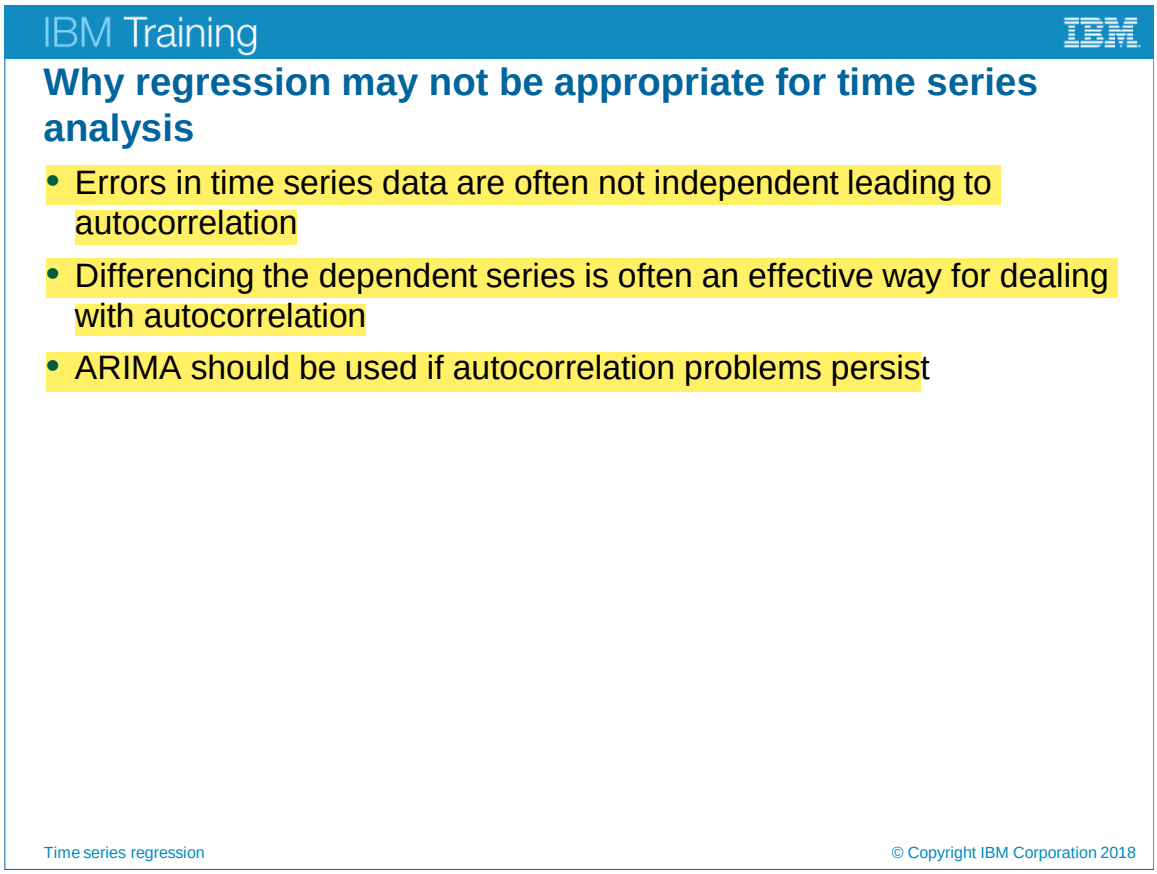
Also for valid regression results, the errors (residuals) should not be correlated with each other (they should be independent). The standard time series autocorrelation and partial autocorrelation functions can be used to test for autocorrelation of errors when regression is run on time-structured data. In general, this can be a problem for time series regression, since the value of a series at one time point will influence its value at the next point (and so the errors will also be correlated).

A third assumption of regression (not tested in this example) is that residuals should be homoscedastic. This means that the variance of the residuals should be homogeneous over time. If there is any systematic tendency for the variance to change, then this problem is known as heteroscedasticity. A scatterplot of the residual variance against time may reveal heteroscedasticity if there are changes in the variation of the residuals over time.

The regression model must also be correctly specified, which means that all the important sources of influence on the dependent field are included in the model. If not, then the model is misspecified, and there may be, for example, missing predictor fields or an important dynamic pattern in the dependent field that was ignored by the model.

This may or not be a serious problem depending on whether the omitted influences are correlated with the independent fields or the model's errors.

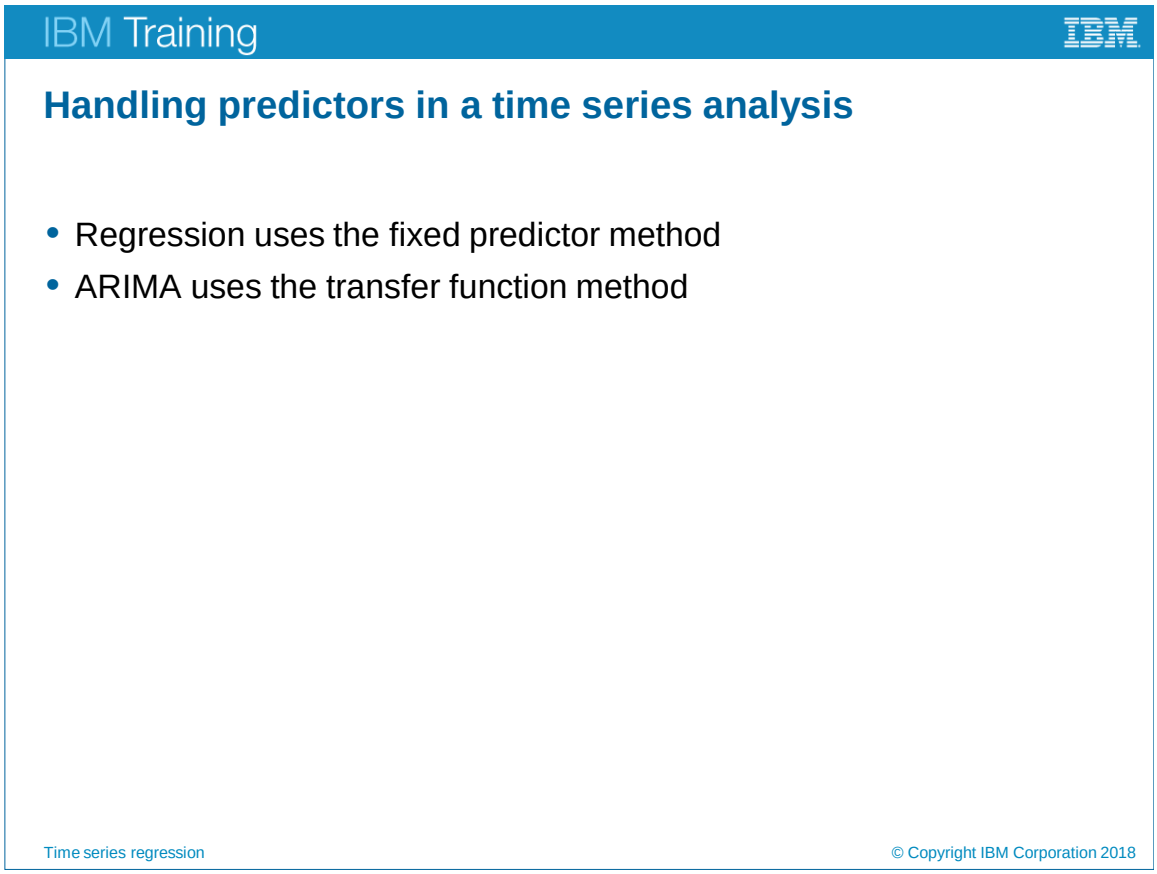
If the model passes these assumptions and tests, it is likely that the model can be used for forecasting.



Why regression may not be appropriate for time series analysis

It is important to keep in mind that although you are using time series data, the modeling technique is still regression, and so you must ensure that the model and data meet appropriate assumptions, especially autocorrelation.

In particular, regression techniques are often inadequate when developing time series models because they can fail to remove autocorrelation from the error. In many cases when regression models are developed, the error will have autocorrelated patterns. This means that the model errors are not independent (which is assumed by regression when performing significance tests) and that the time series model is misspecified. Besides model misspecification, the significance tests on individual coefficients, which are used in deciding which effects to retain in the model, may not be correct.



Handling predictors in a time series analysis

There are a number of ways to take predictors into account in a time series model. One approach, which is the method regression uses, is to treat them as fixed predictors. For example, if advertising expenditures is used as a predictor of sales, regression would report the average rate of change in sales for every unit change in advertising expenditure. A problem with this approach is that more than one past value of advertising expenditures one, or two, three, or even more time periods back may have an effect on current sales and it is difficult to know the correct number of past values to include in the model.

To get around this problem, transfer functions use a backshift operator that notes in condensed form that lags are taken into account of a field. This approach was finalized by Box and Jenkins. Rather than treating the predictor as fixed, they calculate how long it took for an advertising campaign to begin to effect sales, how long the advertising campaign continue to affect sales, and when the effect of the advertising spending began to decay.

Although, transfer functions are beyond the scope of this course, it should be noted that when the Expert Modeler creates an ARIMA model, it uses the transfer function approach. You should use Regression instead if you prefer to treat predictors as fixed.

IBM Training

Demonstration 1

Fit a regression model to time series data

Demonstration 1: Fit a regression model to time series data

Demonstration 1: Fit a regression model to time series data

Purpose:
You would like to use regression to fit a time series model. The data record the number of tourists who visited the brewery center between the first quarter of 1990 and the fourth quarter of 1996. In addition the number of visitors, the price of admission (in Euros) is recorded, along with the number of tour bus bookings, all recorded quarterly. Your aim is to develop a robust time series model which can then be used to forecast the future number of visitors and the growth of the brewery as a tourist attraction.

Stream file: **unit_4_demonstration_1_start.str**

Folder: **C:\Training\0A028\04-Time_Series_Regression\Start**

Task 1. Start and configure IBM SPSS Modeler 18.1.1.

1. From the **Start** menu, expand **IBM SPSS Modeler 18.1**, and then click **IBM SPSS Modeler 18.1**.

   Note: IBM SPSS Modeler 18.1.1 displays "IBM SPSS Modeler 18.1" in the Start menu and program name.
2. Click **Cancel** to close the **Welcome** dialog box.

   If you have already set the working directory in a previous demonstration or exercise, you can skip to Task 2.
3. From the **File** menu, click **Set Directory**.
4. Beside **Look in**, navigate to the **C:\Training\0A028** folder, and then click **Set**.

Task 2. Open the stream.

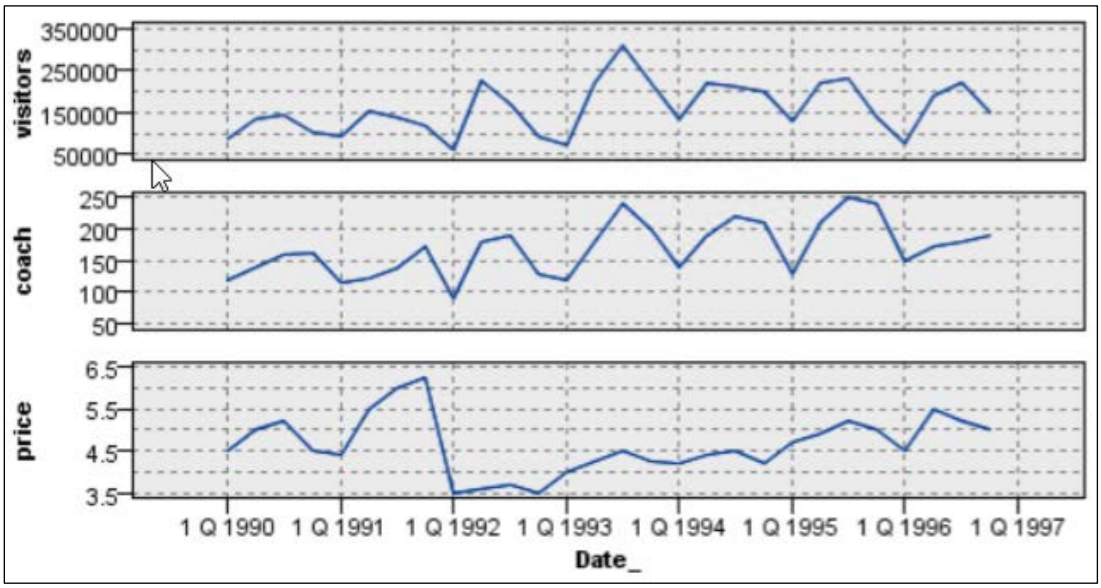
1. From the **File** menu, click **Open Stream**.
2. Navigate to the **C:\Training\0A028\04-Time_Series_Regression\Start** folder, and then double-click **unit_4_demonstration_1_start.str**.

Task 3. Create time plots for the brewery fields.

You will normally undertake an exploratory analysis in order to identify important factors that should be incorporated into the model. For this you will use time plot for the series you wish to forecast (visitors) to look for any trend or seasonality. If the dependent field (visitors) has any trend or seasonality then it will be necessary to incorporate these factors into the time series model. The time plots for the predictors may also reveal useful information.

1. From the **Graphs** palette, select the **Time Plot**, and then add it downstream from the **Type** node.
2. Edit the **Time Plot** node.
3. Beside **Series**, select **visitors**, **coach**, and **price**.
4. Besides **X axis label**, enable the **Custom** option, and then select **Date_**.
5. Ensure that the **Display series in separate panels** option is selected. This will let you review the time plot for each field separately.
6. Clear the **Normalize** option.
7. Click **Run**.

The results appear as follows:



The number of visitors to the brewery follows a clear seasonal pattern. The second and third quarters of each year appear to attract more visitors than the opening and closing quarters of each year. Given that you wish to forecast the number of brewery visitors and that this series exhibits seasonality, it may well be important to include quarterly dummy fields in the regression in order to capture the inherent seasonal pattern in the data. The idea behind dummy fields will be explained shortly. There appears to be less evidence of a trend, although you will also add a time trend to the model to be complete.

The time plot for the number of tour bookings also follows a seasonal pattern and is likely to be a good predictor of the total number of brewery visitors, especially if bookings are done well in advance (allowing it to be used for forecasts).

Ticket prices may also influence the volume of tourists visiting. The average ticket price increased above €5 (5 Euros) towards the latter part of 1991 and was reduced during the first quarter of 1992 following a ticket price reduction. Since then the average price charged has gradually risen through the end of 1996.

8. Close the **Time Plot** output window.

Task 4. Add fields to account for trend and seasonal effects.

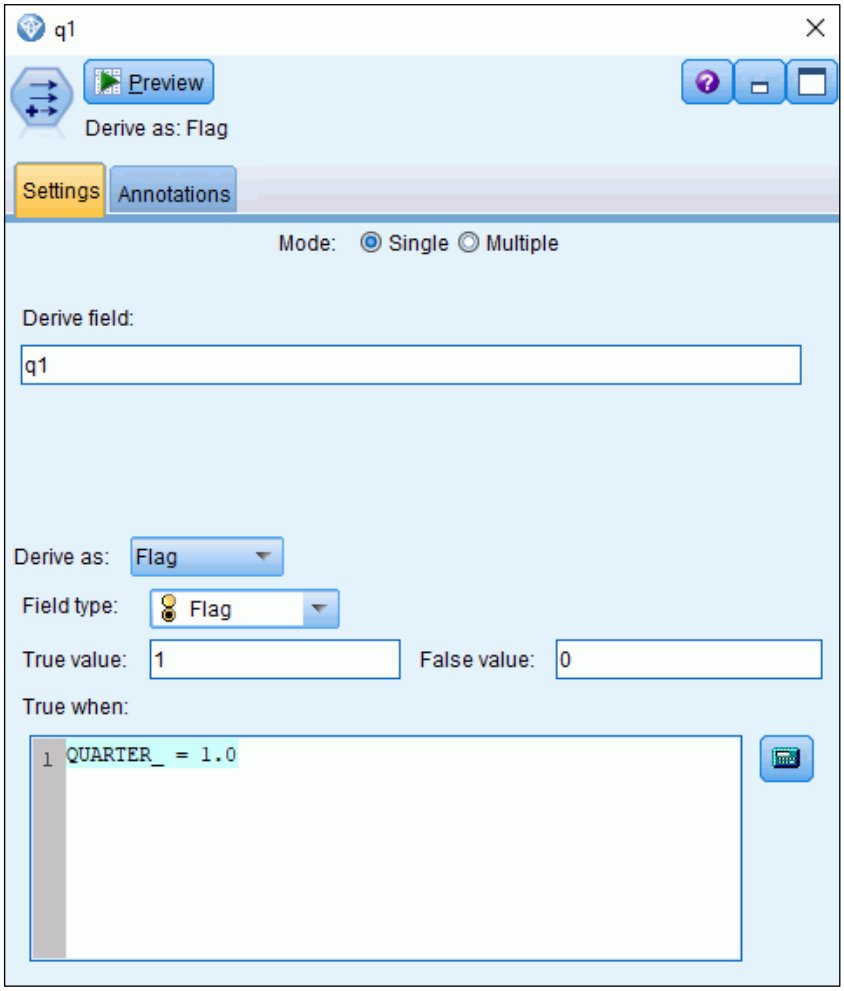
Based on your review of the time plots, you will develop additional fields that will enable you to account for the trend and seasonal patterns you observed in the charts.

First, you will create seasonal dummy fields to capture the seasonal (quarterly) variation in the number of people visiting the brewery. Although there are four quarters of data, the rule with regard to dummy fields is that the number of dummy fields should equal the number of categories minus one. The reason why one less dummy is created than the number of categories is that the complete set of dummy fields (one for each category) is redundant (linearly dependent). That is, the value of any single dummy field can be predicted from the values of the other dummies.

You create dummy fields for quarters 1, 3, and 4, and use quarter 2 as the reference category. This means that each of the three dummy fields will measure the relative impact of a quarter on the number of tourists relative to quarter 2.

1. From the **Field Ops** palette, add a **Derive** node downstream from the **Type** node.
2. Edit the **Derive** node.
3. In the **Derive field** box, name the field **q1**.
4. From the **Derive as** menu, select **Flag**.
5. In the **True value** box, change the value from **T** to **1.0** and in the **False value** box, change the value from **F** to **0.0**. This is because Regression requires numeric values for all fields.

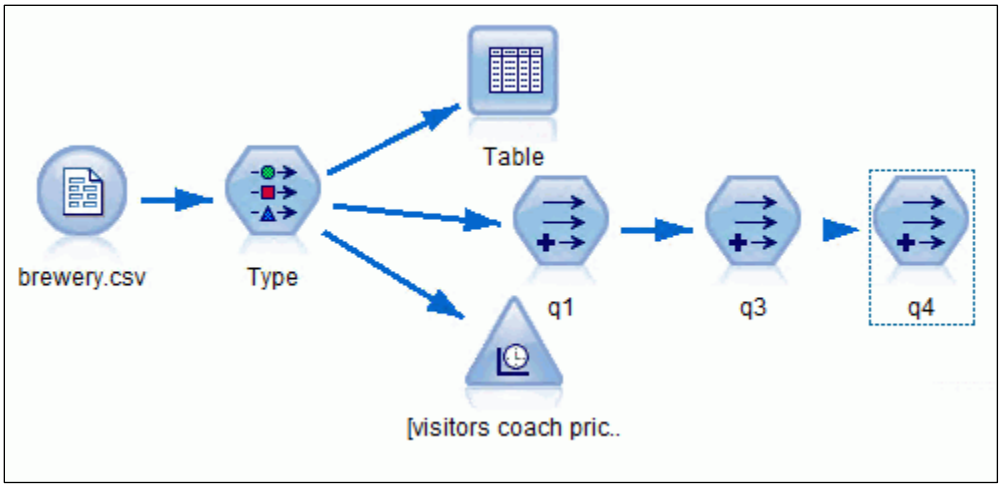
6. In the **True when** box, type **QUARTER_ = 1.0.**
   The results appear as follows:



7. Click **OK**.
8. From the **Field Ops** palette, add a **Derive** node downstream from the **Derive** node named **q1**.
9. Edit the **Derive** node.
10. In the **Derive field** box, name the field **q3**.
11. From the **Derive as** menu, select **Flag**.
12. In the **True value** box, change the value from **T** to **1.0** and in the **False value** box, change the value from **F** to **0.0**. This is because Regression requires numeric values for all fields.
13. In the **True when** box, type **QUARTER_ = 3.0**.
14. Click **OK**.
15. From the **Field Ops** palette, add a **Derive** node downstream from the **Type** node.
16. Edit the **Derive** node.

17. In the **Derive field** box, name the field **q4**.
18. From the **Derive as** menu, select **Flag**.
19. In the **True value** box, change the value from **T** to **1.0** and in the **False value** box, change the value from **F** to **0.0**. This is because Regression requires numeric values for all fields.
20. In the **True when** box, type **QUARTER_ = 4.0**.
21. Click **OK**.

    The results appear as follows:



    The dummy fields have been added to the stream. Next, you will add a time trend field to account for the potentially increasing number of people visiting the brewery over time.

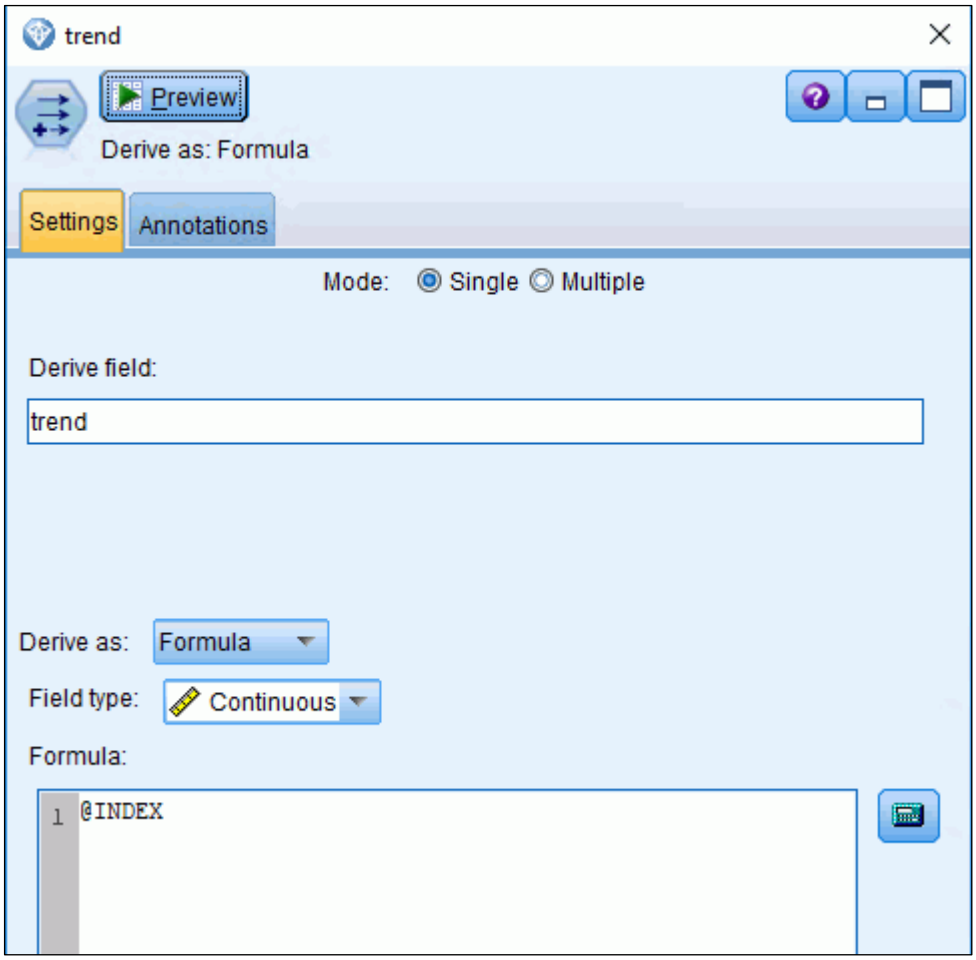
22. From the **Field Ops** palette, add a **Derive** node downstream from the **Derive** node named **q4**.

    Because time is often used to model an upward or downward linear trend in a time series, you will create a trend field that is simply a sequential count of the period or case number.

23. Edit the **Derive** node.
24. In the **Derive field** box, name the field **trend**.

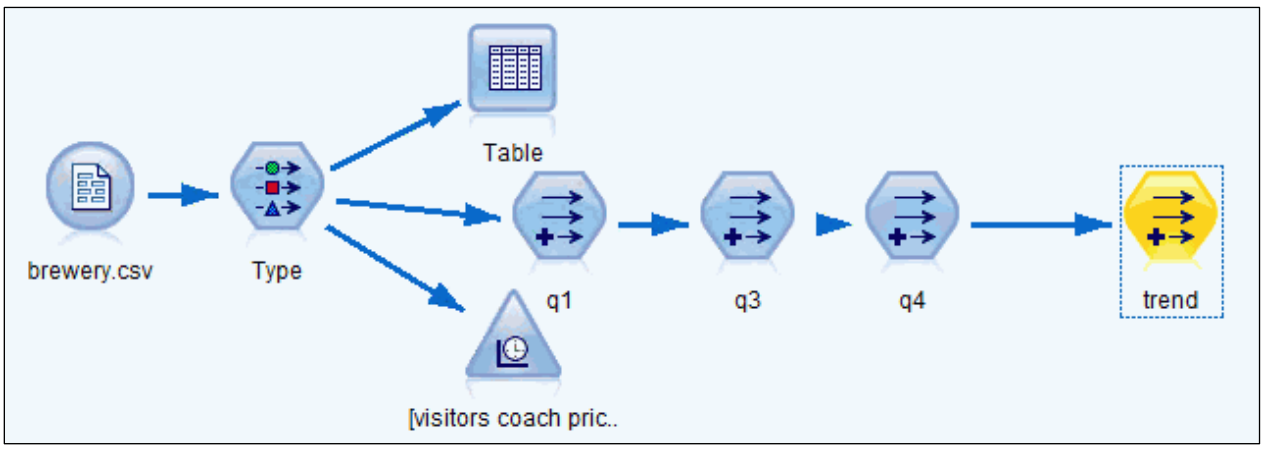
25. Change the **Field type** value to **Continuous**.
26. In the **Formula** box, type **@INDEX**. This will assign a sequential value to each case beginning with **1**.

    The results appear similar to the following:



27. Click **OK**.

    The results appear similar to the following:



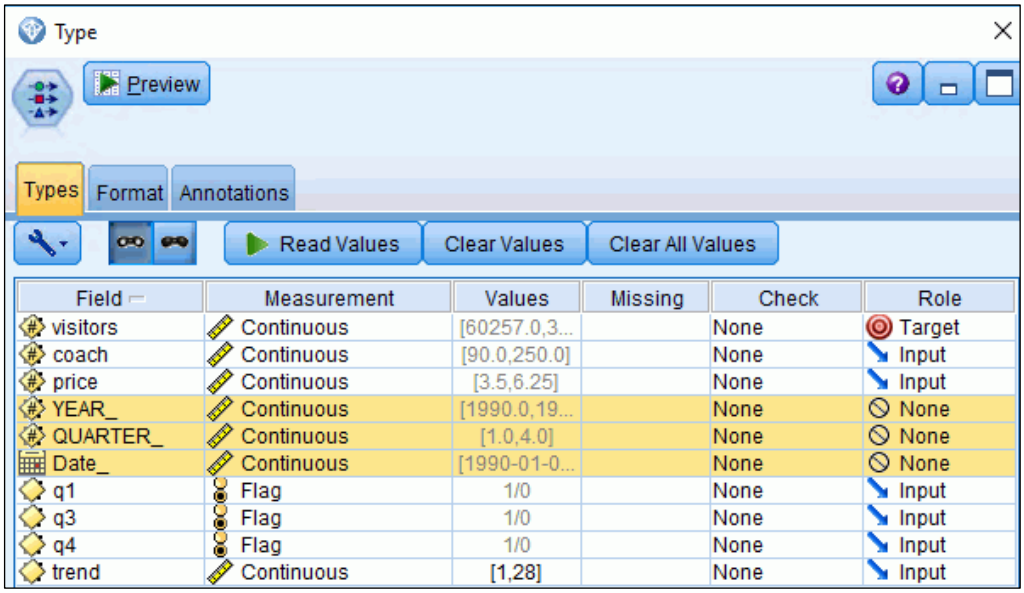
The trend field has been added to the stream.

Task 5. Specifying the regression model.

Having now identified or created the fields to incorporate into the regression model, the next stage is to build the model. This model will include the following elements: the number of visitors as the dependent field; the three quarterly dummy fields, the time trend, the number of coach bookings, and the average ticket price as independent fields.

1. From the **Field Ops** palette, add a **Type** node downstream from the **Derive** node named **trend**.
2. Edit the **Type** node.
3. Click **Read Values**.
4. In the **visitors** row, click the cell under the **Role** column, and then select **Target**.
5. Ensure the role for **coach**, **price**, **trend**, **q1**, **q3**, **q4** are set to **Input**.
6. Set the role of **YEAR_**, **QUARTER_** and **Date_** to **None**.

   The results appear as follows:



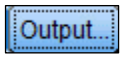
Field	Measurement	Values	Missing	Check	Role
visitors	Continuous	[60257.0,3...		None	Target
coach	Continuous	[90.0,250.0]		None	Input
price	Continuous	[3.5,6.25]		None	Input
YEAR_	Continuous	[1990.0,19...		None	None
QUARTER_	Continuous	[1.0,4.0]		None	None
Date_	Continuous	[1990-01-0...		None	None
q1	Flag	1/0		None	Input
q3	Flag	1/0		None	Input
q4	Flag	1/0		None	Input
trend	Continuous	[1,28]		None	Input

7. Close the **Type** dialog box.

   The next step will be to set up the regression analysis.
8. Click the **Modeling** palette, and then click the **Supervised** sub palette (see on the left of the Modeling palette).
9. Add the **Regression** node downstream from the most recently added **Type** node.

   Note: If you cannot locate the Regression node, click the All sub palette, and then take the Regression node from there.

   You will request diagnostic statistics.

10. Edit the **Regression** node.
11. Click the **Expert** tab.
12. Beside **Mode**, enable the **Expert** option.
13. Click **Output** 

.
14. Select the **Residuals** option.
15. Select the **Durbin-Watson** option.

    This is a test statistic used to detect the presence of autocorrelation in the residuals. The Durbin-Watson statistic ranges from 0 to 4. A value of 2 means that there is no autocorrelation in the sample. The presence of autocorrelation will invalidate the regression results.
16. Close the **Linear Regression: Advanced Output Options** dialog box.
17. Click **Run**.

    A model nugget is generated that stores the results of the analysis.
18. Edit the **model nugget**.
19. Click the **Advanced** tab.
20. Click the **Model Summary** entry.

    The results appear as follows:

Model Summary

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate	Durbin-Watson
1	.896[a]	.802	.746	31157.50576	1.009

a. Predictors: (Constant), trend, q3, price, q4, q1, coach

R Square equals 0.802, indicating a close fit. The Adjusted R Square corrects for the fact that in regression models the R square increases (increases or remains the same) with the number of independent fields. The adjusted R square estimates the explained model variation more conservatively at 0.746 or 74.6%, which still seems reasonably high. The Durbin-Watson statistic is discussed in the next task.

21. Click the **Anova** entry.

    The results appear as follows:

ANOVA

Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	8.263E+10	6	1.377E+10	14.186	.000[b]
	Residual	2.039E+10	21	970790165.1		
	Total	1.030E+11	27			

b. Predictors: (Constant), trend, q3, price, q4, q1, coach

The F test tests the null hypothesis whether R Square equals 0. This null hypothesis has to be rejected, given the significance of 0.000. In words, the model has predictive power for the target.

Note: For any test presented in this course, when the significance falls below 0.05, reject the null hypothesis.

22. Click the **Coefficients** entry.

    The results appear as follows:

Coefficients

Model		Unstandardized Coefficients		Standardized Coefficients	t	Sig.
		B	Std. Error	Beta		
1	(Constant)	80975.129	61949.195		1.307	.205
	coach	927.753	252.097	.632	3.680	.001
	price	-10611.297	9272.082	-.120	-1.144	.265
	q1	-62402.913	20985.699	-.445	-2.974	.007
	q3	-13723.054	17867.036	-.098	-.768	.451
	q4	-64310.393	16929.043	-.459	-3.799	.001
	trend	407.952	953.788	.054	.428	.673

The significance column (*Sig*.) indicates that neither linear trend (*sig*. = .673) nor ticket price (*sig*. = .265) is statistically significant (both are well above .05). Also, notice that the *q3* coefficient is not significant (*sig*. = .451), which indicates that the number of third quarter visitors does not differ from the number of second quarter visitors (the reference category). The other two dummy fields are significant predictors at the .05 level, as is the number of coach parties (*Sig.* = .001).

The full regression equation could be written as shown below, using the B, or unstandardized, coefficients.

visitors = 80975 – 62403**q1* –13723**q3* – 64310**q4* + 408**trend* + 928**coach* – 10611**price*

However, since trend and price were found to not be significant, the analysis often is rerun and the equation developed without those fields

The *q1*, *q3*, and *q4* dummy fields represent the seasonal importance of these respective quarters with respect to quarter 2. From the output it seems that after controlling for the differences in trend, ticket price, and number of coach parties, quarter two is the best quarter for visitors, since all of the other quarters have negative regression coefficients. However, recall that the q3 coefficient was not significant.

Putting this in terms that are useful to the managers of the brewery, on average:

- Quarter 1 has approximately 62,000 visitors less than Quarter 2.
- Quarter 3 has approximately the same number of visitors as Quarter 2 (not significantly different).
- Quarter 4 has approximately 64,000 visitors less than Quarter 2.

The number of coach parties was a significant predictor, and the B coefficient indicates that one extra party leads to an increase of about 928 visitors. This doesn't make sense if, indeed, a coach party is represented by a single bus (so 928 people are jammed onto a single bus), but it is possible that 1) the field is a stand-in for more drive-up and walk-up visitors, or 2) a coach party might refer to an entire tour group or a tour company that might make daily or weekly visits to the facility. In practice, the researcher would verify the source of this field to make certain it is measuring something meaningful and that the coefficient can be interpreted and used.

If it were significant (which it wasn't), the trend coefficient would suggest a positive trend where visitor numbers increase, on average by 408 visitors each quarter over the six year period. There was a suggestion of this in the time plot, but in the full model, the trend is not significant. This means that the number of visitors has not been increasing over time, once we control for the other factors. Finally, the B coefficient for ticket price seems to suggest a negative relationship with the number of visitors, which makes sense. In other words, as price increases the number of visitors would fall and vice versa (however, the price coefficient was not significant).

The standardized beta coefficients display the regression coefficients after all fields are converted to z-scores (with means of 0 and standard deviations of 1), forcing all fields onto the same scale. In doing so, the standardized beta values measure the relative importance of each factor. The standardized beta coefficient usually takes a value between –1 and +1. Using this measure, the number of coach parties is the strongest predictor, followed by the *q1* and *q4* dummy fields.

23. Close the **model nugget**.

Task 6. Testing model fit.

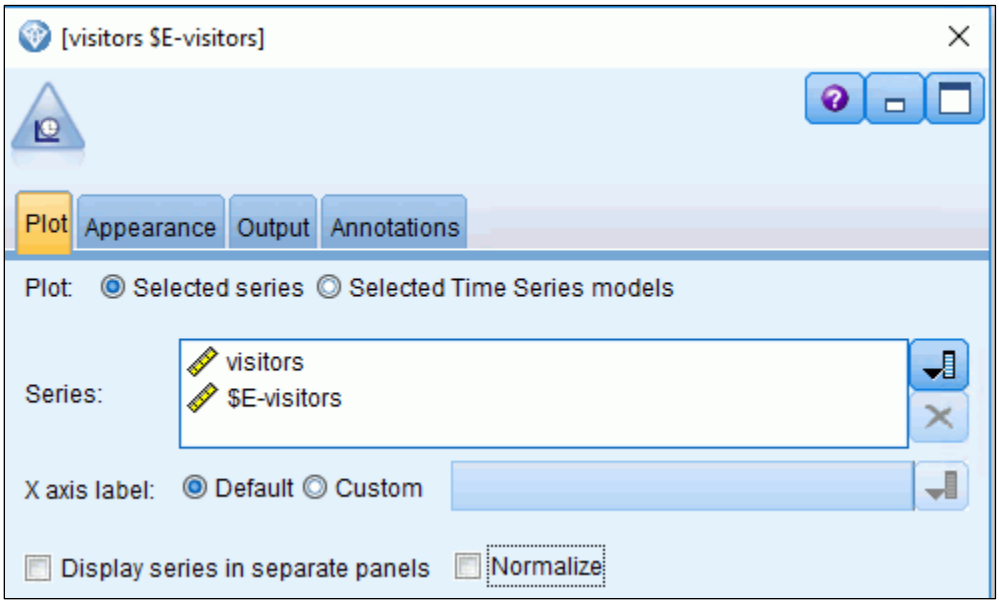
When discussing the coefficients above, we made reference to the significance values. The significance values assume that the errors are independent and normally distributed. If they are not, that is, if autocorrelation is present in the errors, then these significance tests may be compromised. Also, confidence intervals on forecasts of the number of visitors will be incorrect. Thus, rather than simply rerunning the model after dropping the nonsignificant predictors, we examine the regression residuals for the presence of autocorrelation and other problems.

You are certainly interested in how well the model fits the data. The R-square gives us a sense that the fit is reasonably good, but you can use a time plot to investigate this further.

1. From the **Graphs** palette, add a **Time Plot** node downstream from the **model nugget**.
2. Edit the **Time Plot** node.
3. Besides **Series**, select **visitors** and **$E-visitors**. The field $E-visitors is the visitors value predicted by the model.
4. Besides **X axis label**, enable the **Custom** option, and then select **Date_**.
5. Clear the **Display series in separate panels** option.

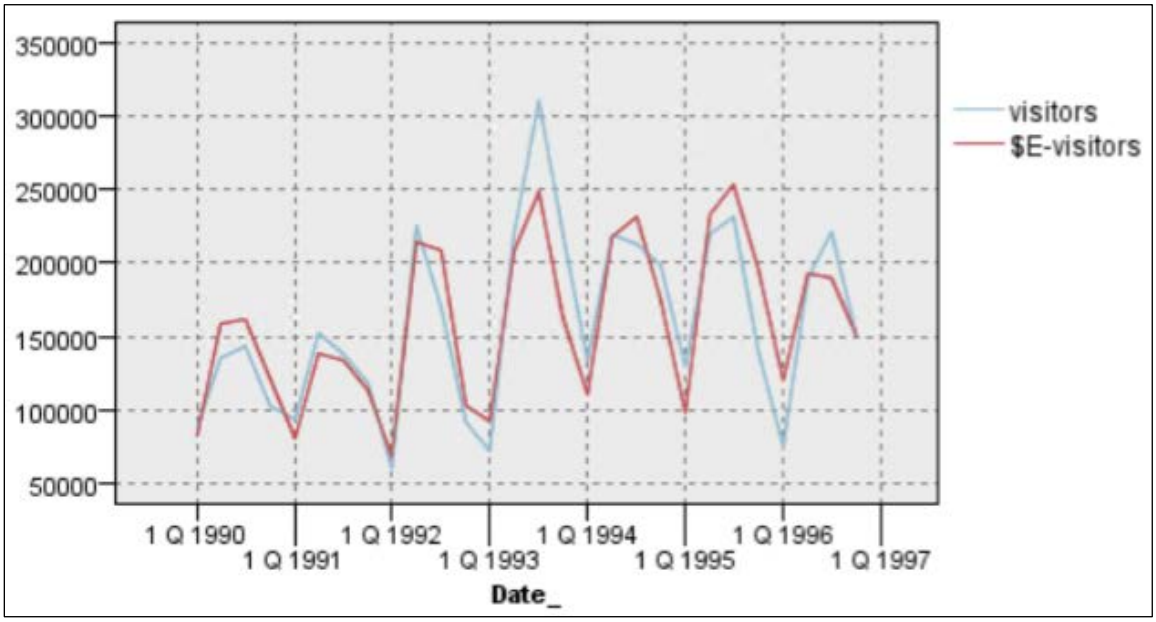
   This will allow you to review the actual and the fit values together on the same time plot.
6. Clear the **Normalize** option.

   The results appear as follows:



7. Click **Run**.

   The results appear as follows:



   The fit looks fairly impressive, although the model tends to either over, or under predict, at the peaks and troughs.

   The errors should be random. You can assess this with a time plot of the errors.

8. Close the **Time plot** output window.
9. From the **Field Ops** palette, add a **Derive** node downstream from the model nugget.
10. Edit the **Derive** node.
11. In the Derive **field** name text box, type **residual**.
12. Beside the **Formula** text box, click **Launch expression builder**  .
13. Construct the expression **visitors - '$E-visitors'** and then close the **Expression Builder** dialog box.
14. Close the **Derive** dialog box.
15. From the **Graphs** palette, add a **Time Plot** downstream from the **Derive** node named **residual**.
16. Edit the **Time Plot** node, and then besides **Series** select **residual**.
17. Beside **X axis label**, enable the **Custom** option, and then select **Date_**.

18. Clear the **Normalize** option.
19. Click **Run**.

    The results appear as follows:

    What you notice is that there appears to be a roughly equal number of positive and negative errors. However, the errors are not random because one positive error tends to be followed by another, and vice versa. So this indicates that there might be significant autocorrelation.

20. Close the **Time Plot** output window.

Task 7. Testing for Autocorrelation.

The next stage is to test whether the model error is autocorrelated. If you look at the Model Summary table shown below, the Durbin-Watson statistic is 1.009.

Model Summary

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate	Durbin-Watson
1	.896[a]	.802	.746	31157.50576	1.009

a. Predictors: (Constant), trend, q3, price, q4, q1, coach

This statistic is a test of first-order autocorrelation, and it should be near 2 if there is no autocorrelation. A value of 1.009 indicates positive first order autocorrelation. (The exact critical value of the D-W statistic depends on several factors, including the number of predictors and number of cases, but 1.009 is low).

Next, you will look at the ACF and PACF plots.

1. From the **Field Ops** palette, add a **Type** node downstream from the **Derive** node named **residual**.
2. Edit the **Type** node, and then click **Read Values** to instantiate the **residual** field.

3. Close the **Type** node.
4. From the **Modeling** palette, **All** sub palette, add a **Time Series** node downstream from the **Type** node.
5. Edit the **Time Series** node.
6. Click the **Fields** tab, if necessary.

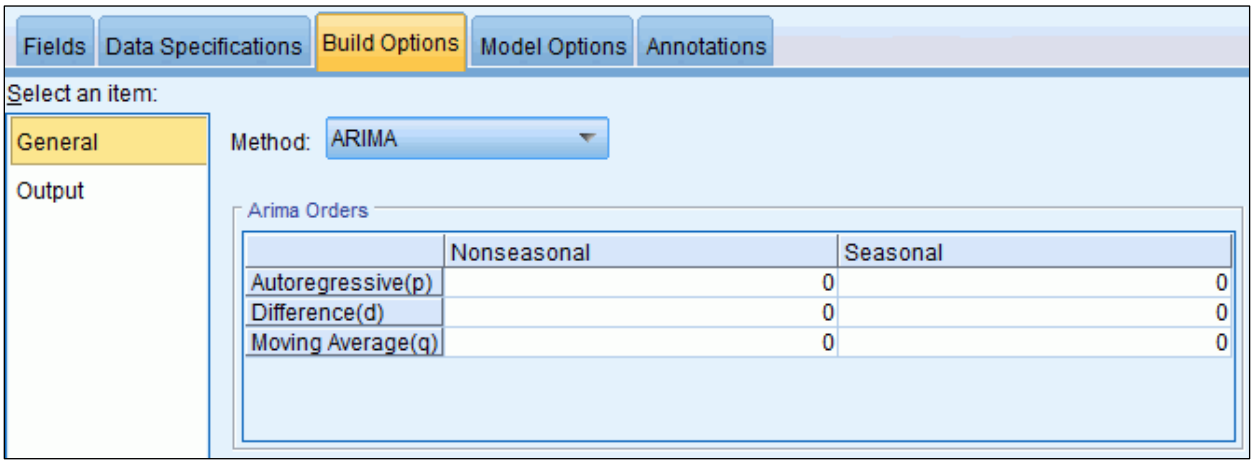
   You can set field roles upstream in the Type node, or you can specify the field roles here. Because you did not set field roles in the Type node yet, you will set them here.
7. Enable the **Use custom field assignments** option.
8. Move **residual** into the **Targets** box.

   This field stores the residuals from the Regression model.
9. Click the **Data Specifications** tab.
10. Click the **Observations** item on the left, if necessary.

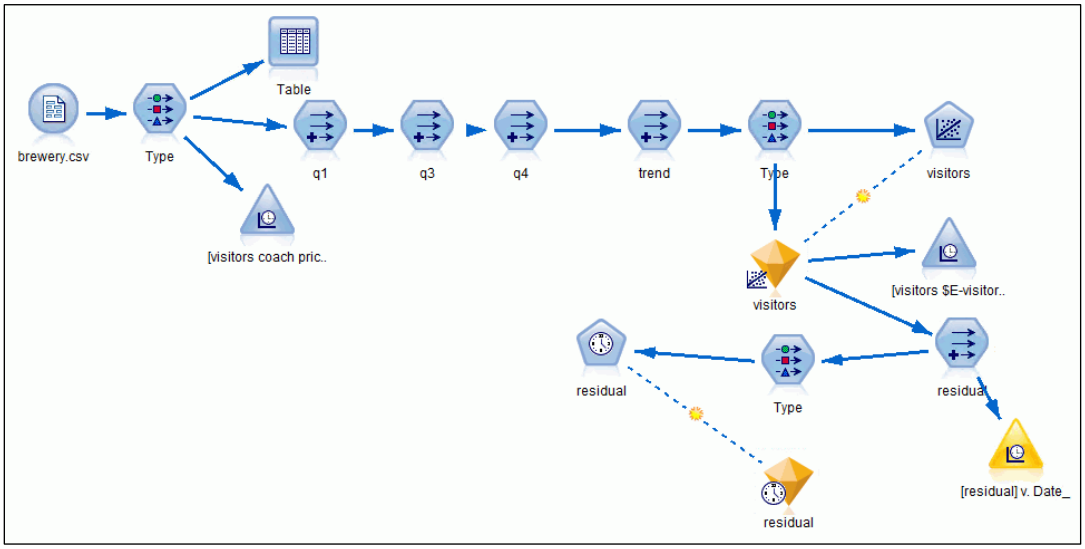
    In this example, the observations are defined by a date field, named **Date_**. Because the data represented quarterly figures, the time interval between the observations are quarters.
11. Ensure that the **Observations are specified by a date/time field** option is enabled.
12. For **Date/time** field, select **Date_**.
13. For **Time interval**, select **Quarters**.
14. Click the **Build Options** tab.
15. From the **Method** list, select **ARIMA**.

    You will create an ARIMA (0,0,0) (0,0,0) model with no seasonal or non-seasonal AR, I, or MA terms and then use ACF and PACF plots to check for correlation in the error terms.

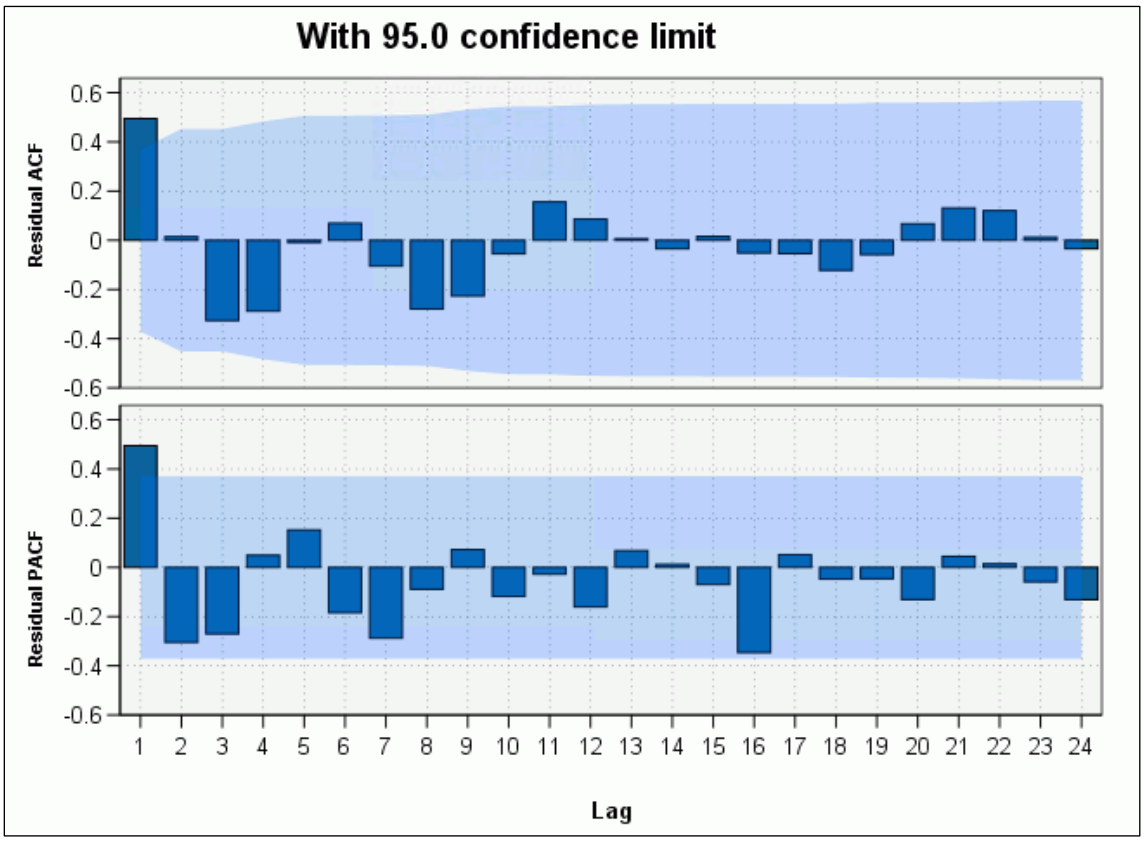
    The results appear as follows:



16. Click **Run**.
    The results appear as follows:



17. Edit the generated **model nugget**.
18. Click the **Output** tab, if necessary.
19. Click **Correlogram**.
    The results appear as follows:



There is evidence of first-order positive autocorrelation. In other words, there is a tendency for a positive error to be followed by another positive error in the next period and for a negative error to be followed by a negative error the following period (just as the time plot suggested). This seems to be the only significant source of autocorrelation in the error, though other lags have autocorrelations that almost attain significance (as at lags 3 and 4).

The PACF plot has basically the same pattern, with the lag 1 partial autocorrelation (equivalent to the ACF value) being significant, but no others.

This situation creates at least two problems. First, our original significance tests may not be correct, so fields that were not significant (price or trend) might be significant if the autocorrelation could be removed. Second, confidence intervals on forecasts will be incorrect, typically being too small, which will artificially increase our confidence in the forecast value.

This means that our model needs to be modified. We need to somehow incorporate the correlation in the error. There are sophisticated methods to adjust for autocorrelation using ARIMA models, as we will see in a later unit with these data. In this unit, you will attempt to more or less reduce the autocorrelation by creating a new version of the dependent field visitors.

20. Close the **model nugget**.

Task 8. Differencing the number of visitors.

There are various methods to smooth time series data. One of these methods is differencing, which means taking the value of a field at time t and subtracting the value of that same field at time t–1. Differencing can remove trend from a field, and because it creates a new field that combines data from two periods, it can sometimes mitigate first-order autocorrelation.

You will create a new field that differences visitors.

1. From the **Field Ops** palette, add a **Derive** node to the node named **trend**.
2. Edit the **Derive** node.
3. In the **Derive field** box, name the field **diffvisitors**.
4. Beside the **Formula** text box, click **Launch expression builder**  .
5. Construct the expression **visitors - (@OFFSET(visitors,1))** and then close the **Expression Builder** dialog box.

6. Click **Preview**.

   The results appear as follows:

Table | Annotations

	visitors	coach	price	YEAR_	QUARTER_	Date_	q1	q3	q4	trend	diffvisitors
1	87280.000	120....	4.500	1990....	1.000	1 Q 1990	1	0	0	1	$null$
2	135201.000	140....	5.000	1990....	2.000	2 Q 1990	0	0	0	2	47921.000
3	143950.000	160....	5.200	1990....	3.000	3 Q 1990	0	1	0	3	8749.000
4	102543.000	162....	4.500	1990....	4.000	4 Q 1990	0	0	1	4	-41407.000
5	93740.000	115....	4.400	1991....	1.000	1 Q 1991	1	0	0	5	-8803.000
6	152308.000	122....	5.500	1991....	2.000	2 Q 1991	0	0	0	6	58568.000
7	137840.000	137....	6.000	1991....	3.000	3 Q 1991	0	1	0	7	-14468.000
8	118901.000	172....	6.250	1991....	4.000	4 Q 1991	0	0	1	8	-18939.000
9	60257.000	90.000	3.500	1992....	1.000	1 Q 1992	1	0	0	9	-58644.000
10	225321.000	180....	3.600	1992....	2.000	2 Q 1992	0	0	0	10	165064.0...

   Each value of diffvisitors is equal to the current value of visitors minus the previous one. For example, for case 2, diffvisitors equals135201 - 87280= 47921.

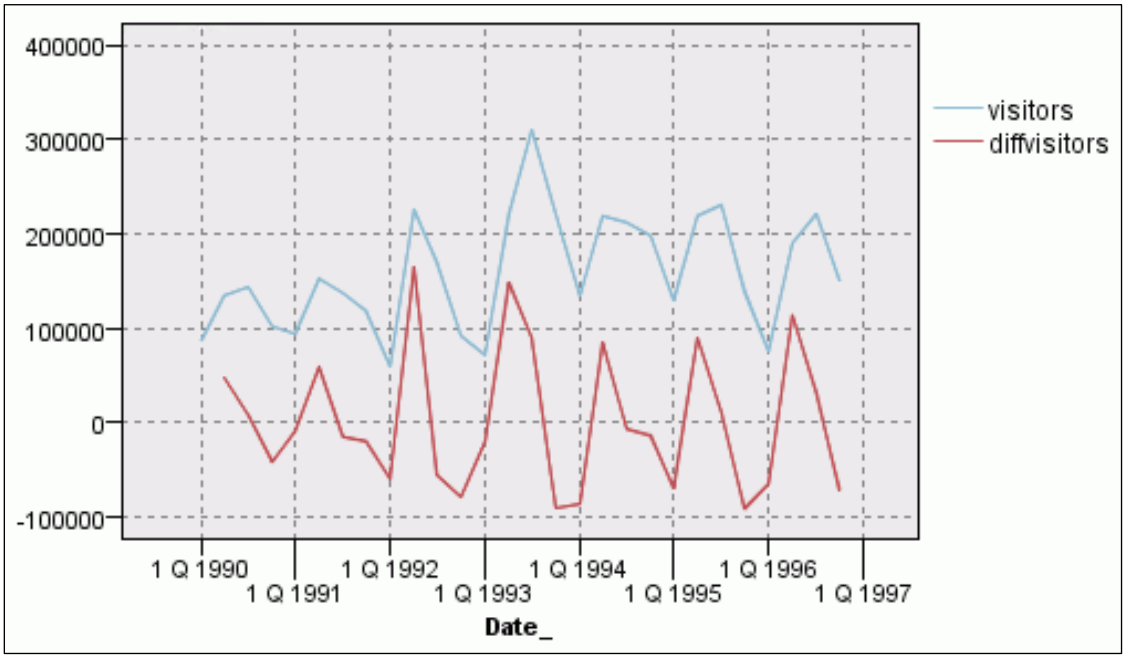
7. Close the **Preview** output window.
8. Close the **Derive** dialog box.

   Now you will verify that differencing removed the trend from the series.

9. From the **Graphs** palette, add a **Time Plot** downstream from the **Derive** node named **diffvisitors**.
10. Edit the **Time Plot** node, and then besides **Series** select **visitors** and **diffvisitors**.
11. Beside **X axis label**, enable the **Custom** option, and then select **Date_**.
12. Clear the **Normalize** option.
13. Clear the **Display series in separate panels** option.

14. Click **Run**.

    The results appear as follows:



    Notice that the series was flattened considerably. Also, the differenced values of visitors still have the same seasonal swings as the original field, so perhaps we have not gained anything. You will find out momentarily.

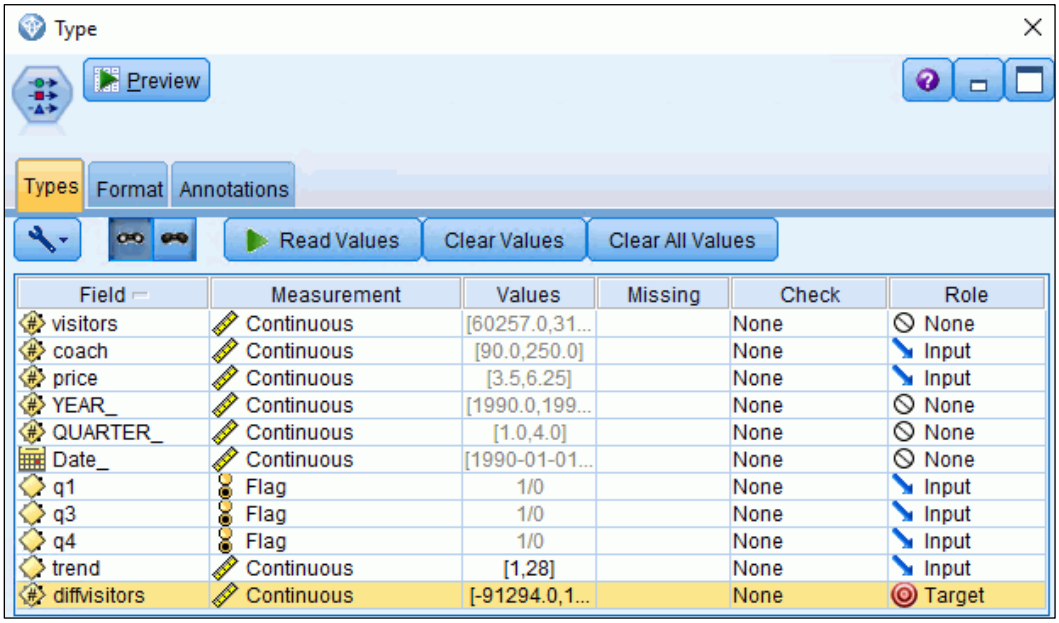
15. Close the **Time plot** output window.

Task 9. Regression model with differenced visitors.

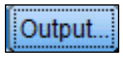
You will rerun the regression model using the field difffvisitors, instead of visitors, as the target.

1. From the **Field Ops** palette, add a **Type** node downstream from the **Derive** node named **diffvisitors**.
2. Edit the **Type** node.
3. Click **Read Values**.
4. In the **diffvisitors** row, click the cell under the **Role** column, and then select **Target**.
5. Ensure the role for **coach**, **price**, **trend**, **q1**, **q3**, and **q4** is set to **Input**.

6. Set the role of **visitors, YEAR_**, **QUARTER_** and **Date_** to **None**.
   The results appear as follows:



Field	Measurement	Values	Missing	Check	Role
visitors	Continuous	[60257.0,31...		None	None
coach	Continuous	[90.0,250.0]		None	Input
price	Continuous	[3.5,6.25]		None	Input
YEAR_	Continuous	[1990.0,199...		None	None
QUARTER_	Continuous	[1.0,4.0]		None	None
Date_	Continuous	[1990-01-01...		None	None
q1	Flag	1/0		None	Input
q3	Flag	1/0		None	Input
q4	Flag	1/0		None	Input
trend	Continuous	[1,28]		None	Input
diffvisitors	Continuous	[-91294.0,1...		None	Target

7. Close the **Type** dialog box.
   The next step will be to set up the regression analysis.
8. Click the **Modeling** palette, and then click the **Supervised** sub palette at the left side.
9. Add the **Regression** node downstream from the most recent **Type** node.
   Note: if you cannot locate the Regression node, click the All sub palette at the left side, and take the Regression node from there.
   You will request diagnostic statistics.
10. Edit the **Regression** node.
11. Click the **Expert** tab.
12. Beside **Mode**, enable the **Expert** option.
13. Click **Output** 

.
14. Select the **Residuals** option.
15. Select the **Durbin-Watson** option.

16. Close the **Linear Regression: Advanced Output Options** dialog box.
17. Click **Run**.

    A model nugget is generated that stores the results of the analysis.
18. Edit the new **model nugget**.
19. Click the **Advanced** tab.
20. Click the **Model Summary** entry.

    The results appear as follows:

Model Summary

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate	Durbin-Watson
1	.885[a]	.783	.718	39768.24428	1.838

a. Predictors: (Constant), trend, q3, price, q4, q1, coach

Differencing the target field seems to have solved the autocorrelation problem. This time the Durbin-Watson statistic is much closer to the 2.0 threshold level so it appears there no longer is first-order autocorrelation.

21. Click the **Coefficients** entry.

    The results appear as follows:

Coefficients

Model		Unstandardized Coefficients		Standardized Coefficients	t	Sig.
		B	Std. Error	Beta		
1	(Constant)	45656.291	80381.421		.568	.576
	coach	434.066	332.035	.242	1.307	.206
	price	-514.222	11944.951	-.005	-.043	.966
	q1	-131344.935	28878.232	-.743	-4.548	.000
	q3	-102231.876	22883.875	-.610	-4.467	.000
	q4	-163552.737	21607.992	-.975	-7.569	.000
	trend	-1137.387	1321.190	-.121	-.861	.400

The only significant fields are the quarterly dummy fields. The fields coach, price and trend and not significant and can be dropped from the model.

You will rerun the regression using only significant predictors.

22. Close the **model nugget**.
23. Edit the **Type** node that is connected to the **Regression** node named **diffvisitors**.
24. Change the roles of **coach**, **price**, and **trend** to **None**.
25. Close the **Type** node.

26. Rerun the **Regression** node named **diffvisitors**.
27. Edit the **model nugget**.
28. Click the **Advanced** tab.
29. Click the **Model Summary** entry.
30. The results appear as follows:

Model Summary

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate	Durbin-Watson
1	.874[a]	.764	.733	38717.09756	2.010

a. Predictors: (Constant), q4, q1, q3

The Durbin-Watson statistic is now almost exactly 2.0, which indicates no autocorrelation.

31. Click the **Coefficients** entry.

The results appear as follows:

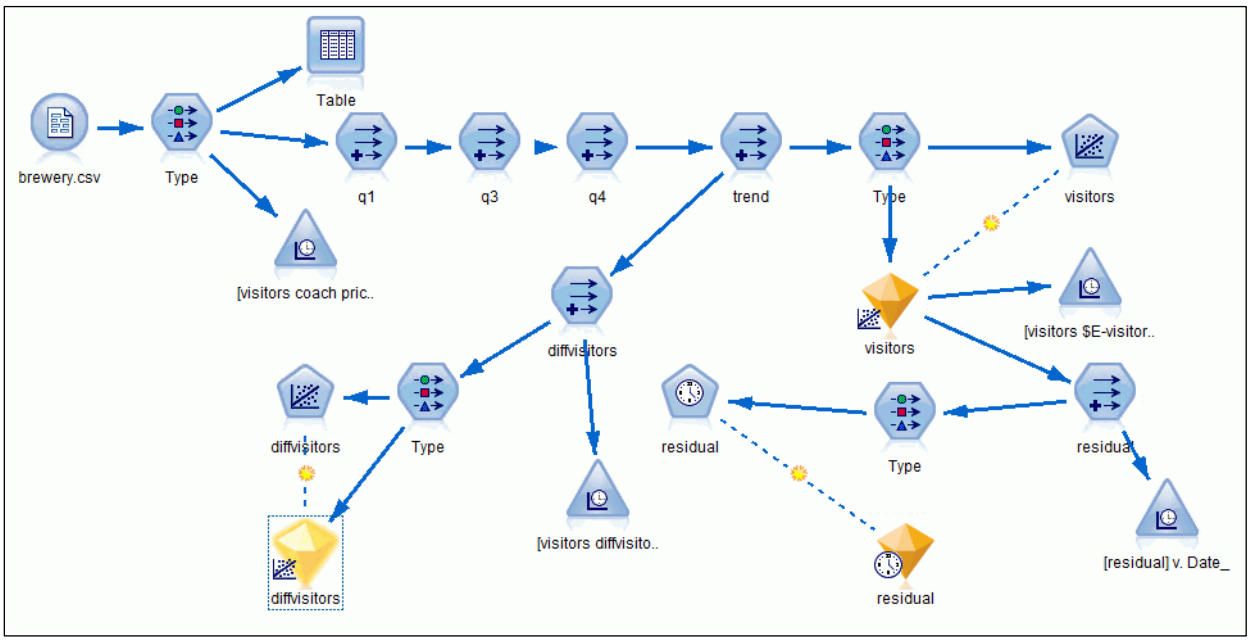
Coefficients

Model		Unstandardized Coefficients		Standardized Coefficients	t	Sig.
		B	Std. Error	Beta		
1	(Constant)	101337.000	14633.687		6.925	.000
	q1	-152358.167	21540.205	-.862	-7.073	.000
	q3	-92075.000	20695.159	-.549	-4.449	.000
	q4	-159097.429	20695.159	-.949	-7.688	.000

Now all the predictor fields are statistically significant.

32. Close the **model nugget**.

    At this point you could use time plots to examine the model fit and to check the residuals for randomness, like you did earlier, but you will not perform those tasks at this time. The final stream is shown below.



    This completes the demonstration. You will create a clean state for the next demonstration.

33. From the **File** menu, click **Close Stream**. Click **No** when asked to save changes.
34. From the **File** menu, click **New Stream**.

    Leave IBM SPSS Modeler open for the next demonstration.

Results:
You have successfully used Regression to create a time series model.

You will find the completed stream in the following folder:
C:\Training\0A028\04-Time_Series_Regression\Solutions

IBM Training

Demonstration 2

Forecasting future values with a regression model

Demonstration 2: Forecasting future values with a regression model

Demonstration 2: Forecasting future values with a Regression model

Purpose:
You created a time series model with regression using your data on the number of visitors each quarter to a brewery from 1990 to 1996. Now you want to use the model to forecast the number of visitors for each quarter in 1997.

Stream file: **unit_4_demonstration_2_start.str**

Folder: **C:\Training\0A028\04-Time_Series_Regression\Start**

Task 1. Forecasting with Regression.

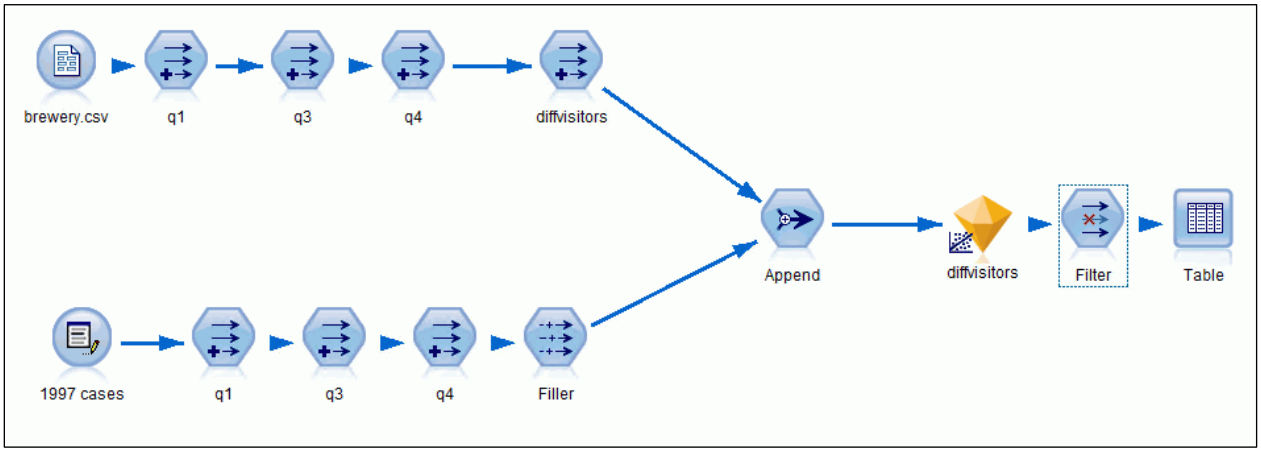
Creating forecasts with a model created with the Regression procedure is less straight forward than ARIMA or Exponential Smoothing because Regression has no built-in functionality to create forecasts automatically. Second, if you have independent fields such as price and coach here, they would need to be have values for future dates to apply the regression equation.

Inputting values for new fields is literally a matter of entering the new values into a spreadsheet or using a User Input node to generate the new values on the fly. Since you typically only use a few predictors, and only forecast for a few time periods, this is not a very difficult task.

For your model, the job is much easier. The only significant predictors are the quarterly dummy fields, whose values are fixed into the future. The officials at the brewery will be satisfied if the number of visitors can be predicted one year (season) into the future, so you will create a forecast for 1997.

The first step is to add four new cases to the file.

1. Open the stream **unit_4_demonstration_2_start.str**.
   The results appear as follows:



   The User Input node on the bottom-left named 1997 cases is used to create the 1997 cases from scratch.

   The Append node on the right adds these cases to the bottom of the brewery.csv file. Normally, in IBM SPSS Modeler you can simply apply a model to the new data itself without combining it with the historical data. However, because the Regression model was created with the differenced version of visitors instead of visitors itself, you need to have all the data together in the same file. There will be more explanation later.

   The generated model named diffvisitors contains the Regression model that will be applied to the combined file.

   The Filter node downstream from the diffvisitors model nugget renamed the Regression predictions from $E-diffvisitors to something more intuitive, PredDiffVisitors

2. Run the **Table** node and scroll to the bottom.

   The results appear as follows:

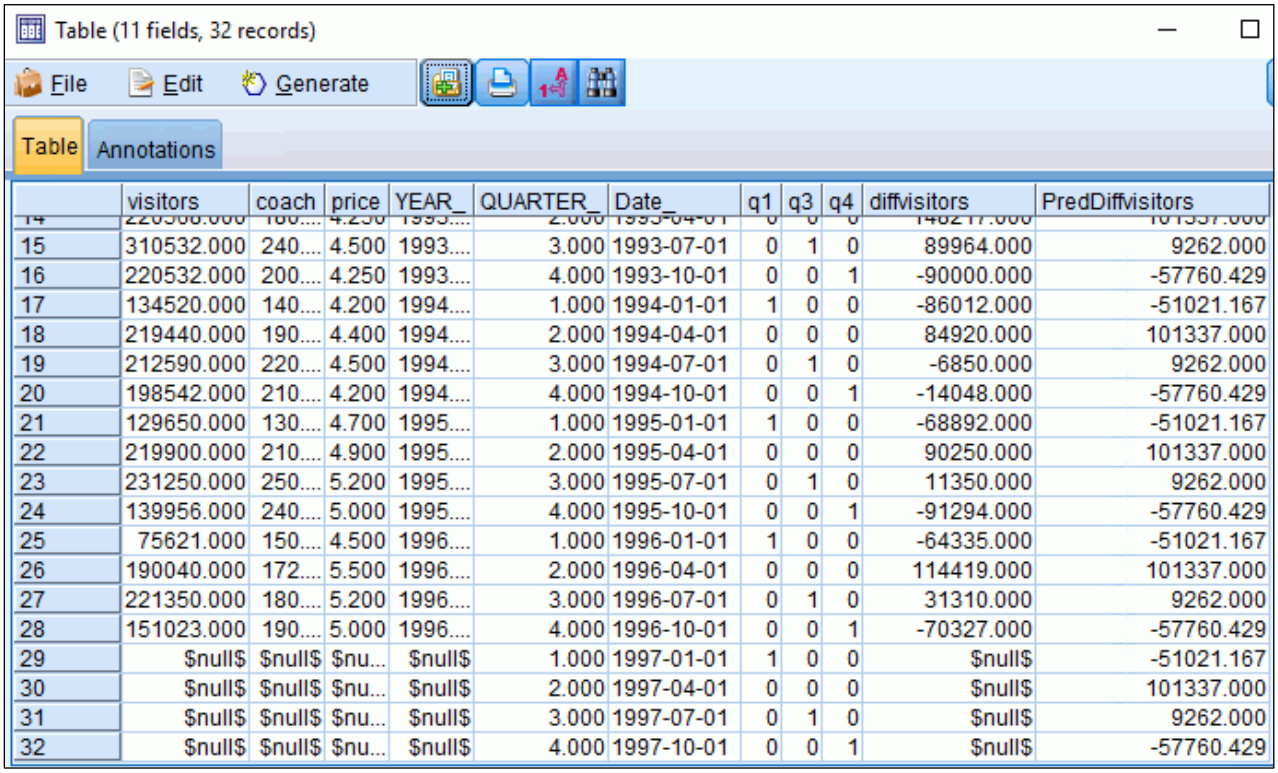


Table (11 fields, 32 records)

File Edit Generate

Table Annotations

	visitors	coach	price	YEAR_	QUARTER_	Date_	q1	q3	q4	diffvisitors	PredDiffvisitors
15	310532.000	240....	4.500	1993....	3.000	1993-07-01	0	1	0	89964.000	9262.000
16	220532.000	200....	4.250	1993....	4.000	1993-10-01	0	0	1	-90000.000	-57760.429
17	134520.000	140....	4.200	1994....	1.000	1994-01-01	1	0	0	-86012.000	-51021.167
18	219440.000	190....	4.400	1994....	2.000	1994-04-01	0	0	0	84920.000	101337.000
19	212590.000	220....	4.500	1994....	3.000	1994-07-01	0	1	0	-6850.000	9262.000
20	198542.000	210....	4.200	1994....	4.000	1994-10-01	0	0	1	-14048.000	-57760.429
21	129650.000	130....	4.700	1995....	1.000	1995-01-01	1	0	0	-68892.000	-51021.167
22	219900.000	210....	4.900	1995....	2.000	1995-04-01	0	0	0	90250.000	101337.000
23	231250.000	250....	5.200	1995....	3.000	1995-07-01	0	1	0	11350.000	9262.000
24	139956.000	240....	5.000	1995....	4.000	1995-10-01	0	0	1	-91294.000	-57760.429
25	75621.000	150....	4.500	1996....	1.000	1996-01-01	1	0	0	-64335.000	-51021.167
26	190040.000	172....	5.500	1996....	2.000	1996-04-01	0	0	0	114419.000	101337.000
27	221350.000	180....	5.200	1996....	3.000	1996-07-01	0	1	0	31310.000	9262.000
28	151023.000	190....	5.000	1996....	4.000	1996-10-01	0	0	1	-70327.000	-57760.429
29	$null$	$null$	$nu...	$null$	1.000	1997-01-01	1	0	0	$null$	-51021.167
30	$null$	$null$	$nu...	$null$	2.000	1997-04-01	0	0	0	$null$	101337.000
31	$null$	$null$	$nu...	$null$	3.000	1997-07-01	0	1	0	$null$	9262.000
32	$null$	$null$	$nu...	$null$	4.000	1997-10-01	0	0	1	$null$	-57760.429

   The predictions have been added to the file for all years including 1997.

3. Close the **Table** output window.

   The next step is to transform the values of differenced visitors to visitors. When we differenced visitors originally, this simple equation was used:

   Differenced Visitors = Current Value visitors - Previous Value visitors

   Therefore, to calculate the current value (for a case) for visitors, we have the equation:

   Current Value visitors = Differenced Visitors + Previous Value visitors

   For the cases in the historical period, you can make calculations with the actual values of visitors and the predicted value for differenced visitor to arrive at the predicted value for visitors. But in the forecast period, that field does not exist, so there you will use the new field PredDiffVisitors. Because you will use two different equations, depending on whether a specific condition is true, you will use a Derive node to create a new field based on conditional logic to accomplish the task.

4. From the **Field Ops** palette, add a **Derive** node downstream from the **Filter** node.
5. Edit the **Derive** node.
6. In the **Derive field** box, name the field **PredVisitors**.

7. From the **Derive as** list, select **Conditional.**

   The next steps are to define the conditions in the provided boxes. You can either type the expressions directly into the boxes or if you prefer, you can create them with the Expression Builder.

   First, you will specify how handle cases which have a valid value for the visitors field. This will include all of the records through 1996.

8. In the **If** box, type **not(@NULL(visitors))**.

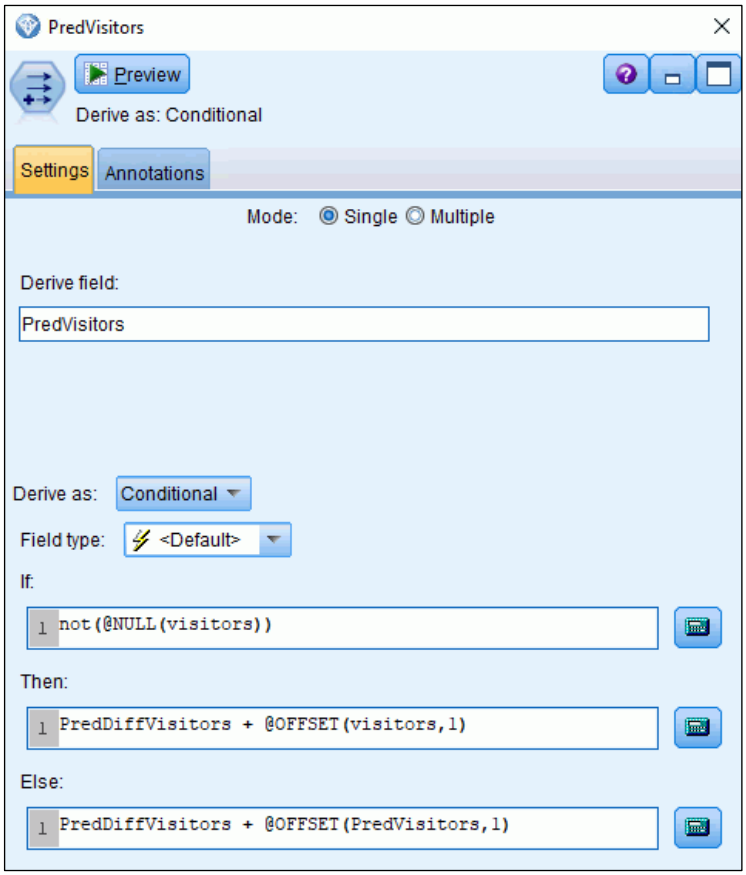
   Next you will create an expression that will calculate PredVisitors for cases that have a non-missing value for visitors.

9. In the **Then** box, type **PredDiffVisitors + @OFFSET(visitors,1)**. For these cases (those from the beginning of the file through Q1 of 1997), PredVisitors is calculated by adding the current value of PredDiffvisitors with the value of visitors on the previous case.

   Because the last three records do not have a valid value of visitors on the previous case, the calculation will need to be done differently.

10. In the **Else** box, type **PredDiffVisitors + @OFFSET(PredVisitors,1)**. For these cases, PredVisitors is calculated by adding the predicted value of the differenced visitors field (PredDiffVisitors) to the predicted value of visitors (PredVisitors) of the previous record.

    The results appear as follows:



11. Close the **Derive** node.
12. From the **Output** palette, add a **Table** node downstream from the **Derive** node named PredVisitors.
13. Run the **Table** node and scroll to the bottom.

    The results appear similar to the following:

Table | Annotations

	e_	q1	q3	q4	diffvisitors	PredDiffVisitors	PredVisitors
13	3-01-01	1	0	0	-19441.0...	-51021.167	40770.833
14	3-04-01	0	0	0	148217....	101337.000	173688.000
15	3-07-01	0	1	0	89964.000	9262.000	229830.000
16	3-10-01	0	0	1	-90000.0...	-57760.429	252771.571
17	4-01-01	1	0	0	-86012.0...	-51021.167	169510.833
18	4-04-01	0	0	0	84920.000	101337.000	235857.000
19	4-07-01	0	1	0	-6850.000	9262.000	228702.000
20	4-10-01	0	0	1	-14048.0...	-57760.429	154829.571
21	5-01-01	1	0	0	-68892.0...	-51021.167	147520.833
22	5-04-01	0	0	0	90250.000	101337.000	230987.000
23	5-07-01	0	1	0	11350.000	9262.000	229162.000
24	5-10-01	0	0	1	-91294.0...	-57760.429	173489.571
25	6-01-01	1	0	0	-64335.0...	-51021.167	88934.833
26	6-04-01	0	0	0	114419....	101337.000	176958.000
27	6-07-01	0	1	0	31310.000	9262.000	199302.000
28	6-10-01	0	0	1	-70327.0...	-57760.429	163589.571
29	7-01-01	1	0	0	$null$	-51021.167	112568.405
30	7-04-01	0	0	0	$null$	101337.000	213905.405
31	7-07-01	0	1	0	$null$	9262.000	223167.405
32	7-10-01	0	0	1	$null$	-57760.429	165406.976

    The predictions have been added to the file for all cases, including the last 4 cases which are the ones you wanted to forecast. The forecasted cases are highlighted in the Table output displayed above.
14. Close the **Table** output window.

    This completes the demonstration. You will create a clean state for the exercise.
15. From the **File** menu, click **Close Stream**. Click **No** when asked to save changes.
16. From the **File** menu, click **New Stream**.

    Leave IBM SPSS Modeler open for the exercise.

Results:
You have successfully used Regression to forecast future values.

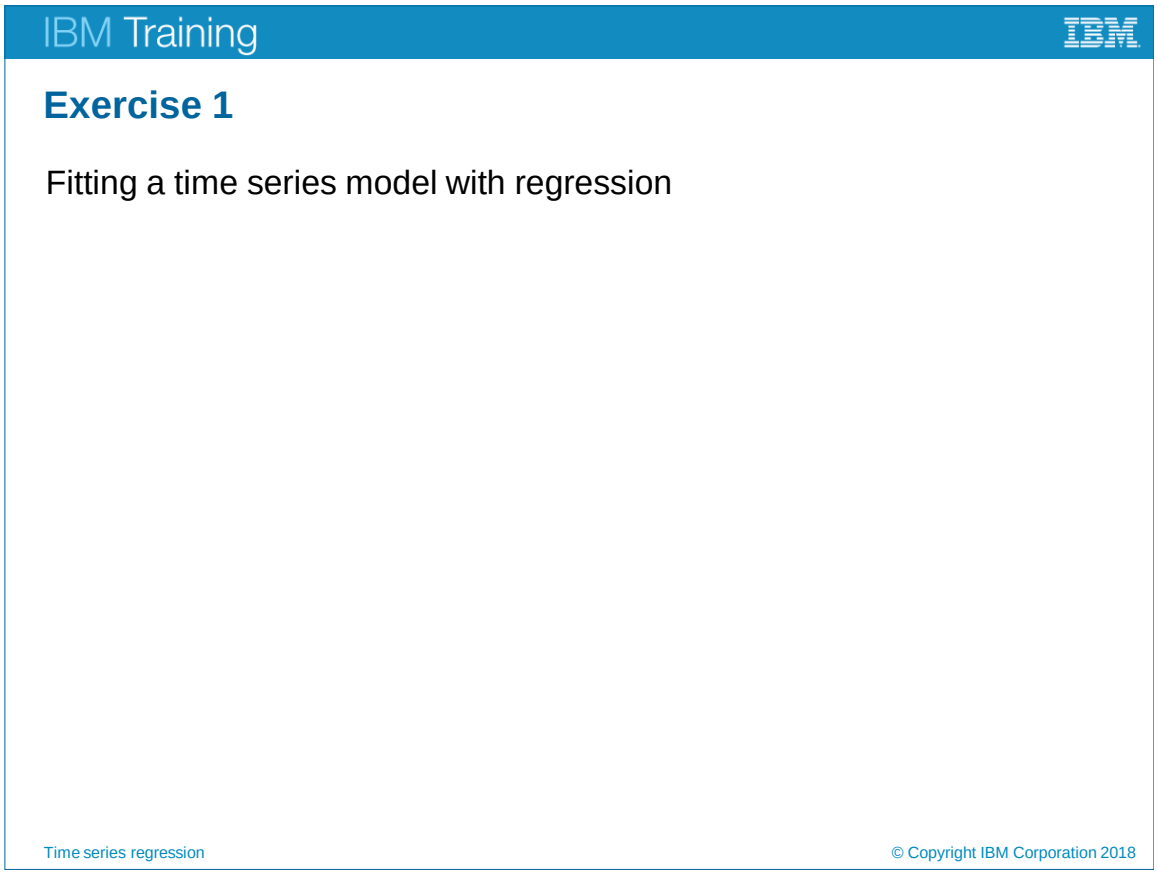
You will find the completed stream in the following folder:
C:\Training\0A028\04-Time_Series_Regression\Solutions

IBM Training

Unit summary

- Use regression to fit a model with trend, seasonality and predictors
- Detect and adjust the model for autocorrelation
- Use a regression model to forecast future values

Unit summary



Exercise 1: Fitting a time series model with regression

Exercise 1: Fitting a time series model with regression

You have a data file on urban crime from a specific region in the United States. There are two fields in the data set: the first measures the rate of reported property crime and the second measures the rate of regional urban unemployment. Both were collected yearly over the period from 1968 to 1992. You would like to use regression to forecast property crime beyond 1992 using urban unemployment as a predictor.

Stream file: **unit_4_exercise_1_start.str**

Folder: **C:\Training\0A028\04-Time_Series_Regression\Start**

Task 1. Open the stream.

- From the **File** menu, click **Open Stream**.
- Navigate to the **C:\Training\0A028\04-Time_Series_Regression\Start** folder, and then double-click **unit_4_exercise_1_start.str**.

Task 2. Examine the data.

- From the **Graphs** palette, select the **Time Plot**, and add it downstream from the **Type** node.
- Create a time plot that displays **property, unemp** on the same chart.
- Use **Year_** to label the **X axis.**
- Disable the **Display series in separate panels** option.
- Disable the **Normalize** option.

  How would you describe the relationship between the property crime rate and unemployment? Does it look like the unemployment rate will be a good predictor of the property crime rate?

Task 3. Fit a simple regression model.

A regression model with only one predictor is referred to as simple regression. Here, the unemployment rate is used as a single predictor for rate of property crime.

- Edit the **Type** node.
- Set the role for **property** to **Target**, **unemp** to **Input**, and **Year_** to **None**.
- Close the **Type** dialog box.
- Click the **Modeling** palette, and then click the **Supervised** sub palette at the left side.

- From the **Modeling** palette, add the **Regression** node downstream from the **Type** node. You will request diagnostic statistics.

  Note: If you cannot locate the Regression node, click the All sub palette at the left side, and then take the Regression node from there.
- Edit the **Regression** node.
- Click the **Expert** tab.
- Beside **Mode**, enable the **Expert** option.
- Click the **Output** button.
- Enable the **Residuals** option.
- Enable the **Durbin-Watson** option.
- Close the **Linear Regression: Advanced Output Options** dialog box.
- Click **Run**.

  A model nugget is generated that stores the results of the analysis.

Task 4. Examine the results.

- Edit the **model nugget**.
- Click the **Advanced** tab.
- Click the **Model Summary** entry.

  How well does the model seem to fit the data?

  Does the Durbin-Watson statistic indicate auto-correlation?
- Click the **Anova** entry.

  Based on the result, does the unemployment rate have a statistically significant effect on the property crime rate?
- Click the **Coefficients** entry.

  Based on the results, by how much can you expect property crime rate to change for every one unit change in the unemployment rate?
- Close the **model nugget**.

Task 5. Forecast property crime rate through 1995.

Because unemployment rate is unknown beyond 1992, to forecast with the Regression model, you first need to provide values for the unemployment rate for the years you want to forecast. One approach would be to use create an Exponential Smoothing model to forecast the unemployment rate after 1992 and use the forecasted values as your inputs. Another approach would be to manually input values or your own choosing. The second approach would allow you to perform a What If? type of analysis in which you could test how a steady reduction in the unemployment rate below the 20% rate in 1992 might affect the property crime rate in future years. You will use the second approach in this exercise.

Now you will use the model to predict property crime rate for three years beyond the historical series.

- Right-click the **model nugget** and click **Copy Node**.
- Right-click the stream canvas and click **Paste**.
- Connect the **model nugget** you pasted to the source node named **crime_thru_1995.csv**. The following values for unemployment rate have already been added to the data file: 19% for 1993, 17% for 1994, and 15% for 1995. Thus, you will be testing whether the property crime will decline if the unemployment rate is brought down.
- From the **Graphs** palette, select the **Time Plot**, and add it downstream from the **model nugget.**
- Edit the **Time Plot** node.
- Besides **Series**, select **$E-property.** These are the Regression model's predictions.
- Besides **X axis label**, enable the **Custom** option, and then select **Year_**.
- Disable the **Normalize** option.
- Click **Run**.

  Based on these results, how will a reduction in the unemployment rate affect the property crime rate in 1992-1995?

For more information about where to work and the exercise results, refer to the Tasks and results section that follows. If you need more information to complete a task, refer to earlier demonstrations for detailed steps

Exercise 1: Tasks and results

Task 1. Open the stream.

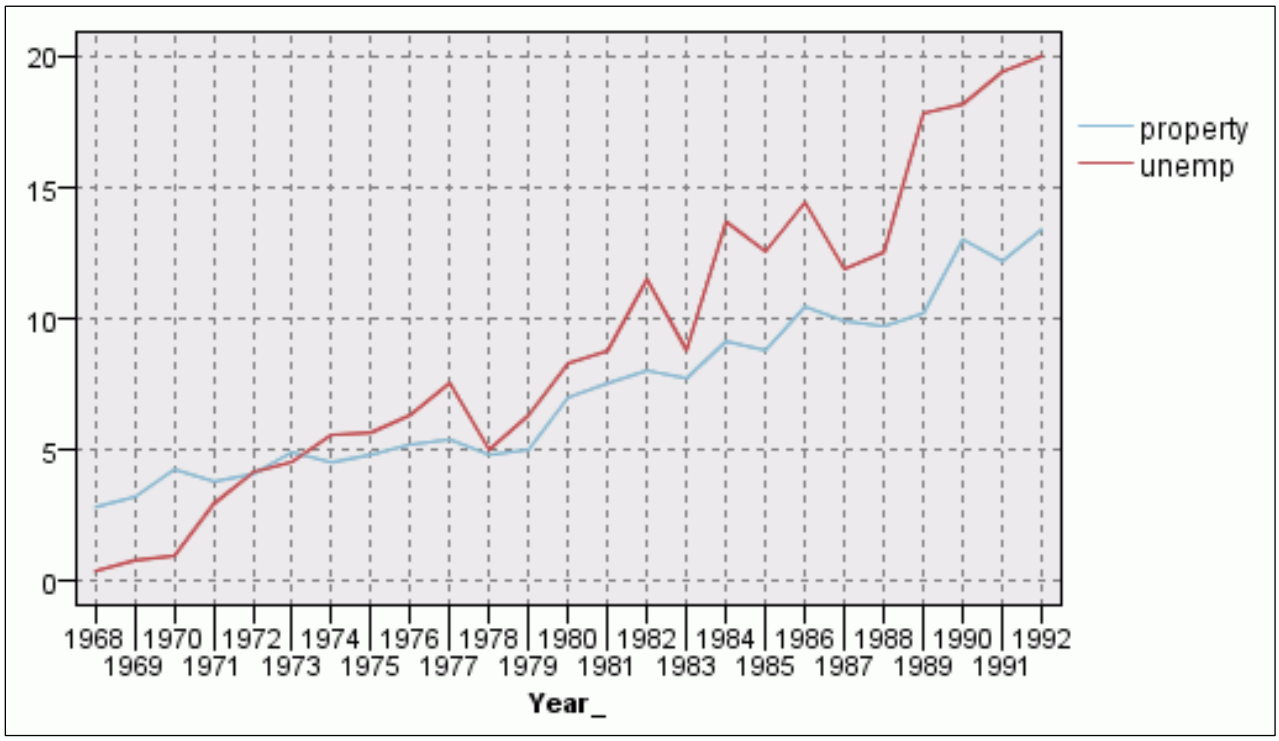
- From the **File** menu, click **Open Stream**.
- Navigate to **C:\Training\0A028\04-Time_Series_Regression\Start**, and then double-click **unit_4_exercise_1_start.str**.

Task 2. Examine the data.

- From the **Graphs** palette, select the **Time Plot**, and add it downstream from the **Type** node.
- Edit the **Time Plot** node.
- Besides **Series**, select **property, unemp**.
- Besides **X axis label**, enable the **Custom** option, and then select **Year_**.
- Clear the **Display series in separate panels** option. This will allow you to review the relationship between each field together on the same chart.
- Clear the **Normalize** option.

- Click **Run**.

  The results appear as follows:



  Examining the plot, the rate of crime has increased over the twenty-five year period and there are two notable sharp rises during the 1979-82 and 1989-92 recessions. The regional urban unemployment rate also increased during most of the period. Given that both fields follow a similar pattern it is likely that one will be a good predictor of the other. So unemployment will be used to predict crime rate.

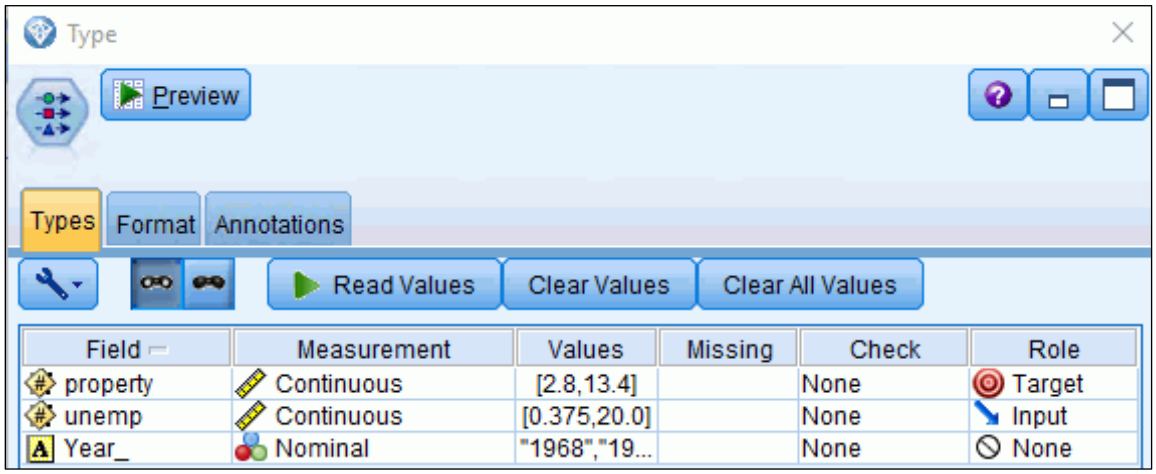
- Close the **Time Plot** output window.

Task 3. Fit a simple regression model.

A regression model with only one predictor is referred to as simple regression. Here, the unemployment rate is used as a single predictor for rate of property crime.

- Edit the **Type** node.
- Set the role for **property** to **Target**, **unemp** to **Input**, and **Year_** to **None**.

  The results appear as follows:



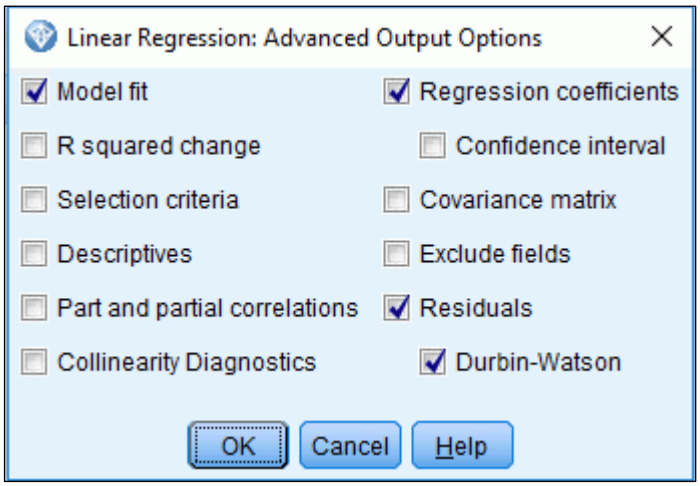
Field	Measurement	Values	Missing	Check	Role
property	Continuous	[2.8,13.4]		None	Target
unemp	Continuous	[0.375,20.0]		None	Input
Year_	Nominal	"1968","19...		None	None

- Close the **Type** dialog box.
- Click the **Modeling** palette, and then click the **Supervised** sub palette at the left side.
- Add the **Regression** node downstream from the **Type** node. You will request diagnostic statistics.

  Note: If you cannot locate the Regression node, click the All sub palette at the left side, and then take the Regression node from there.
- Edit the **Regression** node.
- Click the **Expert** tab.
- Beside **Mode**, enable the **Expert** option.
- Click **Output**.
- Select the **Residuals** option.

- Select the **Durbin-Watson** option.

  The results appear as follows:



- Close the **Linear Regression: Advanced Output Options** dialog box.
- Click **Run**.

  A model nugget is generated that stores the results of the analysis.

Task 4. Examine the results.

- Edit the **model nugget**.
- Click the **Advanced** tab.
- Click the **Model Summary** entry.

  The results appear as follows:

Model Summary

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate	Durbin-Watson
1	.973[a]	.948	.945	.737463	2.115

a. Predictors: (Constant), unemp

R Square equals 0.948, indicating a close fit. The Durbin-Watson statistic equals 2.142. It falls between 1.5 and 2.5, indicating that there is no auto correlation.

- Click the **ANOVA** entry.

  The results appear as follows:

ANOVA

Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	226.101	1	226.101	415.740	.000[b]
	Residual	12.509	23	.544		
	Total	238.609	24			

b. Predictors: (Constant), unemp

  The F test tests the null hypothesis whether R Square equals 0. This null hypothesis has to be rejected, given the significance of 0.000. In words, the model has predictive power for the target.

- Click the **Coefficients** entry.

  The results appear as follows:

Coefficients

Model		Unstandardized Coefficients		Standardized Coefficients	t	Sig.
		B	Std. Error	Beta		
1	(Constant)	2.409	.277		8.700	.000
	unemp	.524	.026	.973	20.390	.000

  Thus, the relationship between unemployment rate and crime rate is as follows:

  Expected property crime rate (property) = 2.409 + 0.524 * unemployment rate (unemp).

  The interest lies in the coefficient for the predictor, 0.524 here. In words, you expect that a one-unit increase in the unemployment rate will be associated with a 0.524 unit increase in the property crime rate.

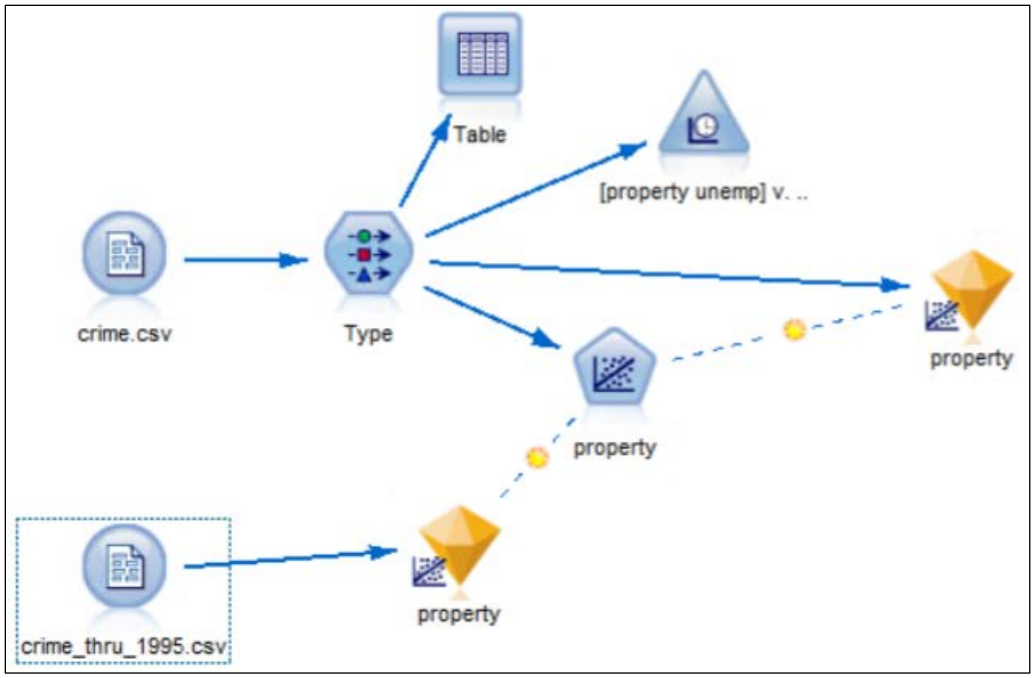
- Close the **model nugget**.

Task 5. Forecast property crime rate through 1995.

Because unemployment rate is unknown beyond 1992, to forecast with the Regression model, you first need to provide values for the unemployment rate for the years you want to forecast. One approach would be to use create an Exponential Smoothing model to forecast the unemployment rate after 1992 and use the forecasted values as your inputs. Another approach would be to manually input some hypothetical values of your own to test how a reduction in the future unemployment rate might affect property crime. For instance, what would happen to property crime if the unemployment rate dipped from 20% in 1992 down to 19% in 1993, 17% in 1994, and 15% in 1995? Would property crime also go down? You will use the second approach in this exercise.

Now you will use the model to predict property crime rate for three years beyond the historical series.

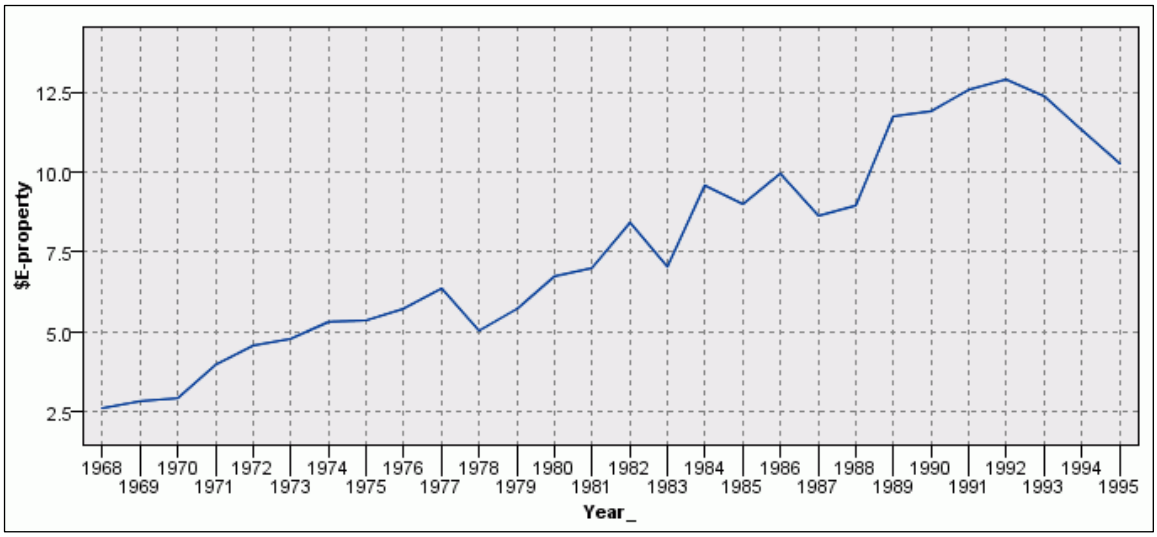
- Right-click the **model nugget**, and then click **Copy Node**.
- Right-click the stream canvas, and then click **Paste**.
- Connect the **model nugget** you pasted to the source node named **crime_thru_1995.csv**. The following hypothetical values for unemployment rate have already been added to the data file: 19% for 1993, 17% for 1994, and 15% for 1995. Thus, you will be testing whether the property crime will decline if the unemployment rate is brought down.



- From the **Graphs** palette, select the **Time Plot**, and add it downstream from the **model nugget.**
- Edit the **Time Plot** node.

- Besides **Series**, select **$E-property.** These are the Regression model's predictions.
- Besides **X axis label**, enable the **Custom** option, and then select **Year_**.
- Clear the **Normalize** option.
- Click **Run**.

  The results appear as follows:



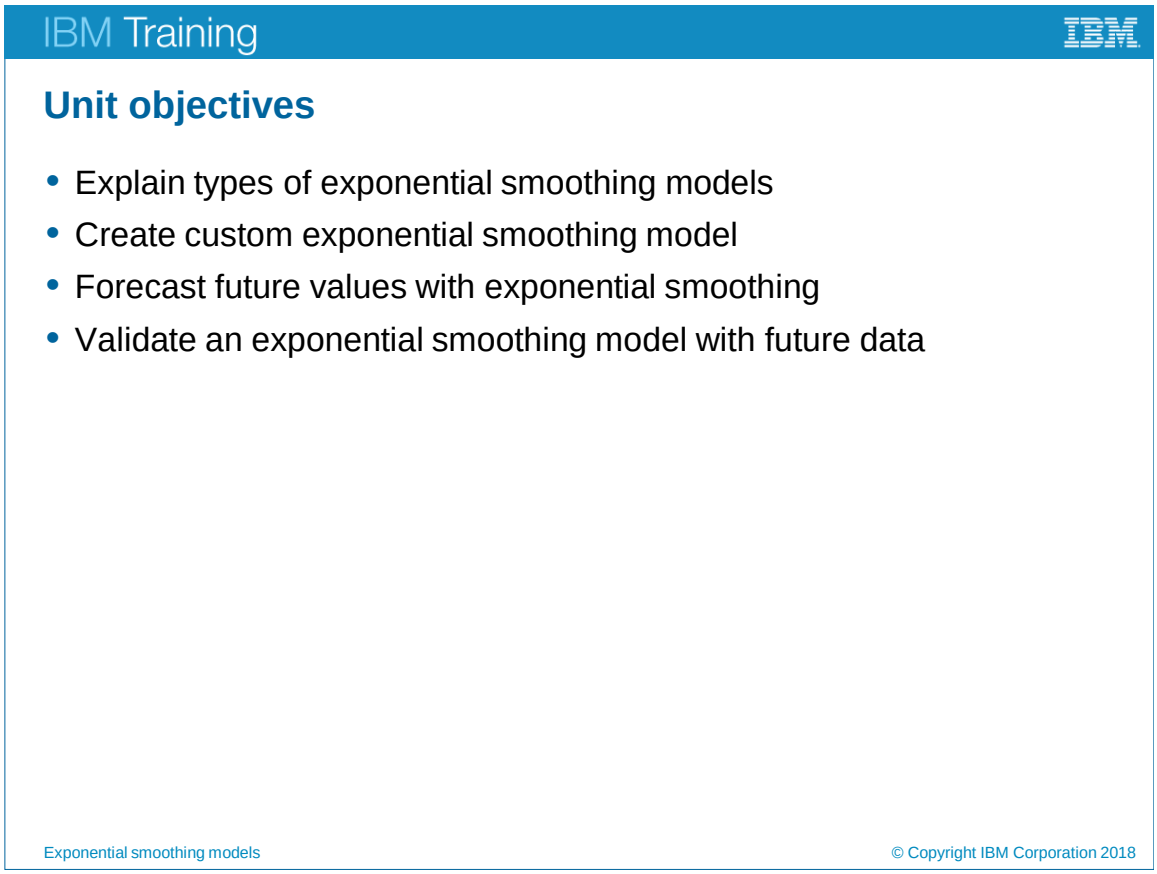
  These results strongly suggest that bringing the unemployment rate down to more reasonable levels will decrease the property crime rate as well.

You will find the completed stream in the following folder:
C:\Training\0A028\04-Time_Series_Regression\Solutions

IBM Training

Exponential smoothing models

IBM SPSS Modeler (v18.1.1)



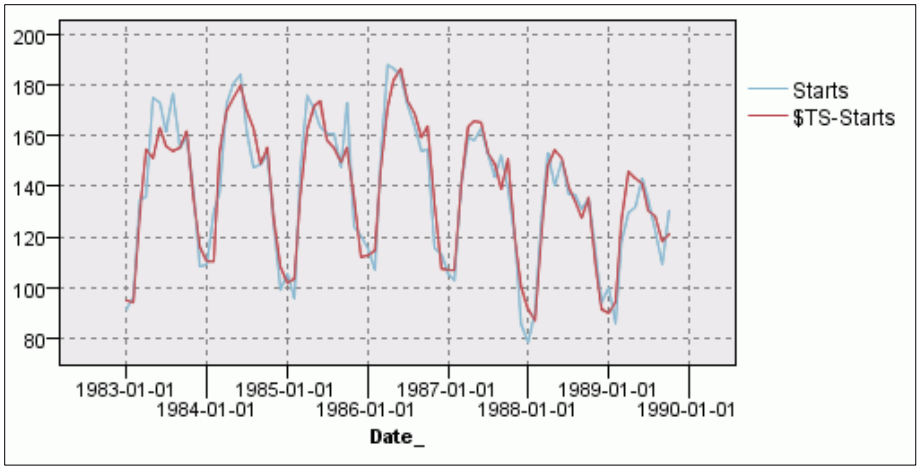
Unit objectives

Before reviewing this unit, you should be familiar with the following topics:

- Working with IBM SPSS Modeler (streams, nodes, palettes)
- Importing data (Var. File node)
- Defining measurement levels, roles, blanks, and instantiating data (Type node)
- Examining the data (Table node, Time Plot node)
- Evaluating time series models, using time plots, autocorrelation plots, and time series model fit statistics
- Using the model nugget to score data

IBM Training

Exponential smoothing modeling



Exponential smoothing models

Exponential smoothing modeling technique

The Time Series node considers two classes of time series models when searching for the best time series model for your data: exponential smoothing and ARIMA. This topic presents exponential smoothing.

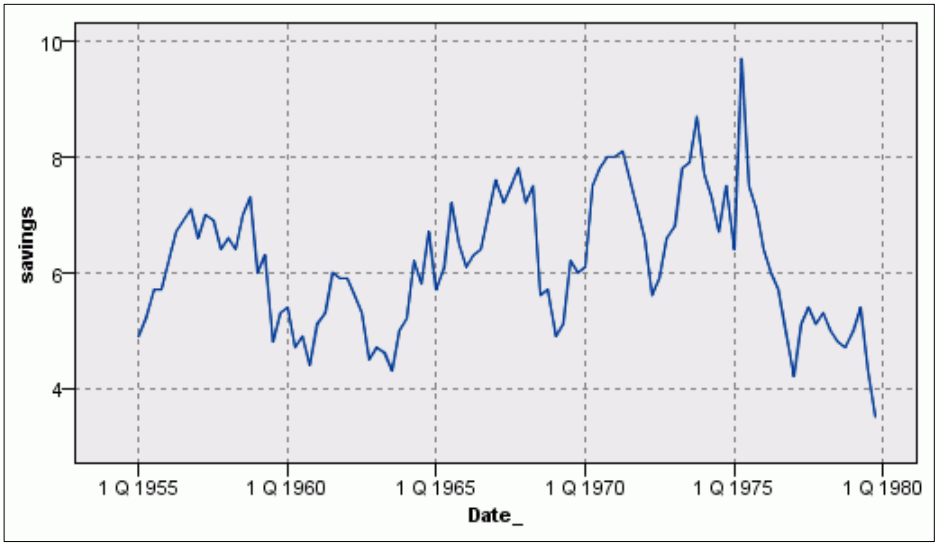
Exponential smoothing is a time series technique that can be a relatively quick way of developing forecasts. The technique is a pure time series method; this means that the technique is suitable when data has only been collected for the series that you wish to forecast, but you have no independent fields. Exponential smoothing can therefore be applied in instances when there are not enough fields to construct good causal time series models, or when the quality of the data is such that causal time series models give poor forecasts

Exponential smoothing models account for three patterns in a series, level, trend, and seasonality. The simplest model is a model without trend and seasonality, so the series just moves around a certain level. This level can remain the same throughout the series, or it may change from time to time.

Typically, you will run the Time Plot first to make some broad characterizations, such as whether there is a trend or seasonality. You can then decide which type of exponential smoothing model to use. Alternatively, you can let the Expert Modeler execute all exponential smoothing models and let it determine the model with the best fit. Either way, you can select the most promising model to generate forecasts.

IBM Training

Simple exponential smoothing



Exponential smoothing models

Simple Exponential Smoothing

The model - recurrence form

Simple exponential smoothing assumes that there is no trend or seasonality present. The following formula, known as the recurrence form, describes the model:

$S_t = \alpha * y_t + (1- \alpha) * S_{t-1}$

Where:

y_t is the observed value of the time series at time t

S_{t-1} is the smoothed level of the series at time t-1

$S_{(t)}$ is the smoothed level of the series at time t

α (alpha) is the smoothing parameter for the level of the series

The formula states that the current smoothed value is obtained by combining information from two sources: the current point and the history embodied in the series.

The smoothed value for the current case becomes the forecast value at all future time points.

To see why the model is called "exponential" smoothing, write out the formula for the recurrence form:

$S_t = \alpha y_t + (1- \alpha) * S_{t-1}$

Substitute the equation for simple exponential smoothing back into itself:

$S_t = \alpha y_t + (1- \alpha) * S_{t-1}$

$= \alpha y_t + \alpha(1- \alpha)y_{t-1} + (1- \alpha)^2 * S_{t-2}$

$= \alpha[y_t + (1- \alpha)y_{t-1} + (1- \alpha)^2 y_{t-2} + (1- \alpha)^3 y_{t-3} + \ldots]$

In computing the smoothed value at time t, observation y_t has a weight of α, y_{t-1} has a weight of $\alpha * (1- \alpha)$, and y_{t-2} has a weight of $\alpha * (1- \alpha)^2$ in computing the smoothed value for S_t. Substituting for S_{t-3} would show that y_{t-3} has a weight of $\alpha * (1- \alpha)^3$, and so forth. Thus, the weights decrease exponentially in terms of (1- α) as observations get older.

How high should the alpha value be? The closer alpha is to 1, the faster exponential smoothing adjusts forecast to patterns in the time series and the less smoothing that occurs. The closer alpha is to 0, the more slowly your forecast adjusts to patterns in the time series, and the more smoothing that occurs. In general, low alpha values are usually appropriate as long as the series is stable. Otherwise, a higher alpha is preferable.

In the simple exponential smoothing model, all that is being modeled is the level of the series. In words, an alpha near 1 means that the most recent observations are weighted more heavily to adjust to the quickly-changing level of the series. An alpha value near 0 implies a slowly-changing level of the series, so more history is used to estimate that level. When alpha would be 0, there is only variation around the series' mean.

The model - error-correction form

The previous section presented the recurrence form of the model:

$S_t = \alpha * y_t + (1- \alpha) * S_{t-1}$

You can rewrite the formula as follows:

$S_t = \alpha * y_t + S_{t-1} - \alpha * S_{t-1}$

$S_t = S_{t-1} + \alpha * (y_t - S_{t-1})$

The term $(y_t - S_{t-1})$ is the difference between the observation y at time t and the smoothed value from time t-1, and is called the one-step-ahead forecast error, denoted as e_t. Using this notation, the formula then becomes:

$S_t = S_{t-1} + \alpha * e_t$

In words, the smoothed level of the series at time t equals the smoothed level of the series at time t-1 plus an adjustment for the difference between the smoothed value at time t-1 and the actual value at the time t. The closer alpha is to 1, the more exponential smoothing adjusts for the difference between the previous smoothed value and the current value, so that quickly changing levels can be modeled.

Forecasting

The simple exponential model states that the series will vary around a certain level, whether slowly or quickly changing. Therefore, the best prediction for future values $\hat{y}_{(t+1)}$, $\hat{y}_{(t+2)}$ and so forth, is the level at which the series sits at time t.

The smoothed value for time t becomes the forecast value for the next times, or:

$\hat{y}_{(t+i)} = S_t$

$\hat{y}_{(t+i)} = S_{(t)}$

Where i = 1, …, m, with m the number of forecasts to make. Notice that the forecasted value will remain the same from time t+1 forward, in agreement with the level of the series at point t. When the level is slowly changing (alpha close to 0), the forecasted value will be an adequate estimate of future values, even for larger values of m. However, when the level is quickly changing (alpha close 1), you know for sure that the series will turn to another level in the near future, so it is risky to forecast too many periods ahead.

IBM Training

Demonstration 1

Simple exponential smoothing

Demonstration 1: Simple Exponential Smoothing

Demonstration 1: Simple exponential smoothing

Purpose:

You have monthly data on two of the products your company sells from 1982 through 1995. After examining a time plot of the data, you decided that based on the patterns you observe in the chart, a simple exponential smoothing model might be the appropriate time series model. You decide to test out your theory to see if you are correct or not.

Stream file: **unit_5_demonstration_1_start.str**

Folder: **C:\Training\0A028\05-Exponential_Smoothing\Start**

Task 1. Start and configure IBM SPSS Modeler 18.1.1.

1. From the **Start** menu, expand **IBM SPSS Modeler 18.1**, and then click **IBM SPSS Modeler 18.1**.

   Note: IBM SPSS Modeler 18.1.1 displays "IBM SPSS Modeler 18.1" in the Start menu and program name.
2. Click **Cancel** to close the **Welcome** dialog box.

   If you have already set the working directory in a previous demonstration or exercise, you can skip to Task 2.
3. From the **File** menu, click **Set Directory**.
4. Beside **Look in**, navigate to the **C:\Training\0A028** folder, and then click **Set**.

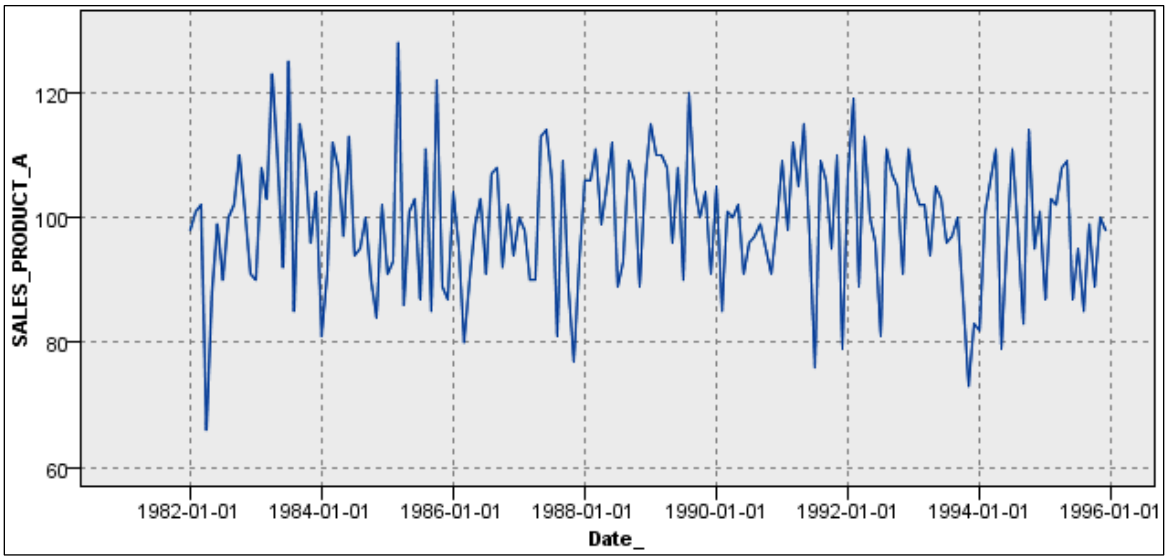
Task 2. Examine the data.

1. From the **File** menu, click **Open Stream**, and then select the file, **unit_5_demonstration_1_start.str**.
2. Run the **Table** node.

   The dataset stores the sales for two products in the period January 1982 to December 1995.
3. Close the **Table** output window.
4. From the **Graphs** palette, select the **Time Plot**, and add it downstream from the **Type** node.
5. Edit the **Time Plot** node.
6. Besides **Series**, select **SALES_PRODUCT_A.**
7. Besides **X axis label**, enable the **Custom** option, and then select **Date_**.

8. Clear the **Normalize** option.
9. Click **Run**.

   The results appear as follows:



   The series seems to move around an overall level; there is no trend and no seasonality. Therefore, a simple exponential model applies, with an alpha level close to 0 to model the level of the series in the entire period.

10. Close the **Time Plot** output window.

Task 3. Create a model.

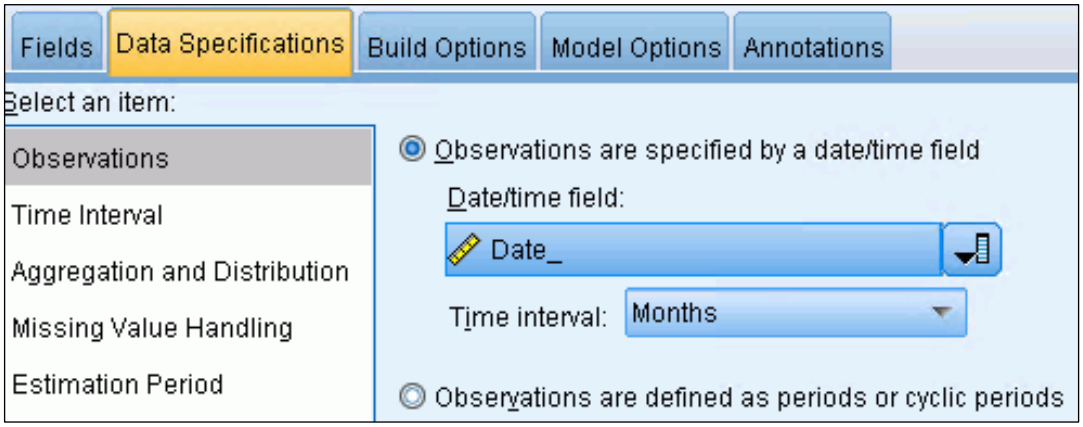
1. Click the **Modeling** palette, and then, at the left side, click **All**.
2. Add a **Time Series** node downstream from the **Type** node.
3. Edit the **Time Series** node.
4. Click the **Fields** tab, if necessary.
5. Enable the **Use custom field assignments** option.
6. Move **SALES_PRODUCT_A** into the **Targets** box.
7. Click the **Data Specifications** tab.
8. Click the **Observations** item on the left, if necessary.

   The observations are defined by the field Date_ and represent months.

9. Ensure that the **Observations are specified by a date/time field** option is enabled.
10. For **Date/time** field, select **Date_**.

11. Beside **Time intervals**, select **Months**.

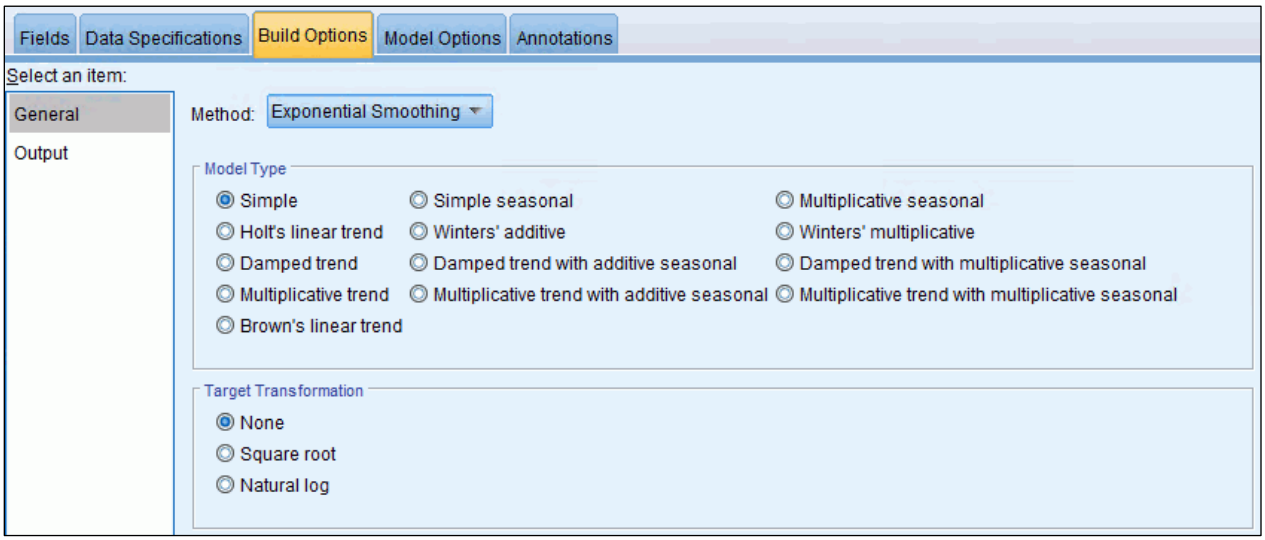
    The results appear as follows:



12. Click the **Build Options** tab.
13. Click the **General** item at the left, if necessary.

    Only exponential smoothing models need to be executed, and a simple exponential smoothing model in particular.
14. Click the **Method** dropdown, and then select **Exponential Smoothing**.
15. Under **Model Type**, select **Simple.**

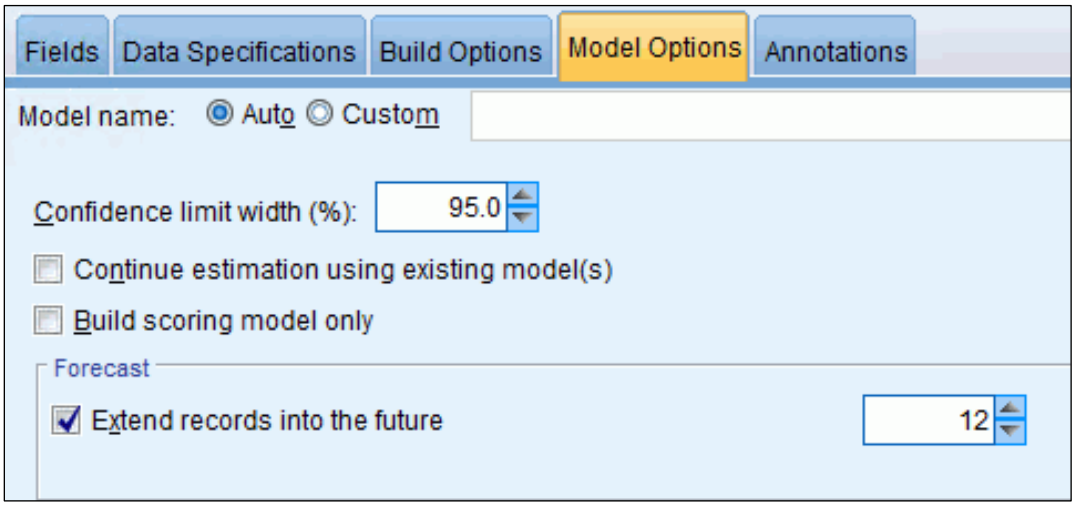
    The results appear as follows:



You will add forecasts for the next year.

16. Click the **Model Options** tab.
17. Under **Forecast**, select the **Extend records into the future** option, and then update the default value to reflect **12**.

    The results appear as follows:



18. Click **Run**.

    A model nugget is generated that stores the results and which can be used to score records.

Task 4. Examine the results.

1. Edit the **model nugget**.

   This table lists the fields that were used to define the observations, the start and the end of the period, and the number of observations used in the analysis.
2. At the left side, click the **Parameter Estimates** entry.

   The results appear as follows:

Parameter Estimates

			Coefficient	Std. Error	t	Significance
SALES_PRODUCT_A	No Transformation	Alpha (Level)	0.004	0.006	0.615	0.539

Alpha is estimated as 0.004, very close to 0. A t test is performed to test the hypothesis that alpha = 0. The significance for that test is 0.539, which means that the hypothesis alpha = 0 cannot be rejected.

3. At the left side, click the **Model Information** entry.
   The results appear as follows:

Model Information

Model Building Method		Exponential Smoothing Simple
Number of Predictors		1
Model Fit	MSE	115.665
	RMSE	10.755
	RMSPE	11.532
	MAE	8.648
	MAPE	8.960
	MAXAE	32.579
	MAXAPE	49.362
	AIC	799.115
	BIC	802.239
	R-Squared	-0.005
	Stationary R-Squared	0.525
Ljung-Box Q(#)	Statistic	15.824
	df	17.0
	Significance	0.5

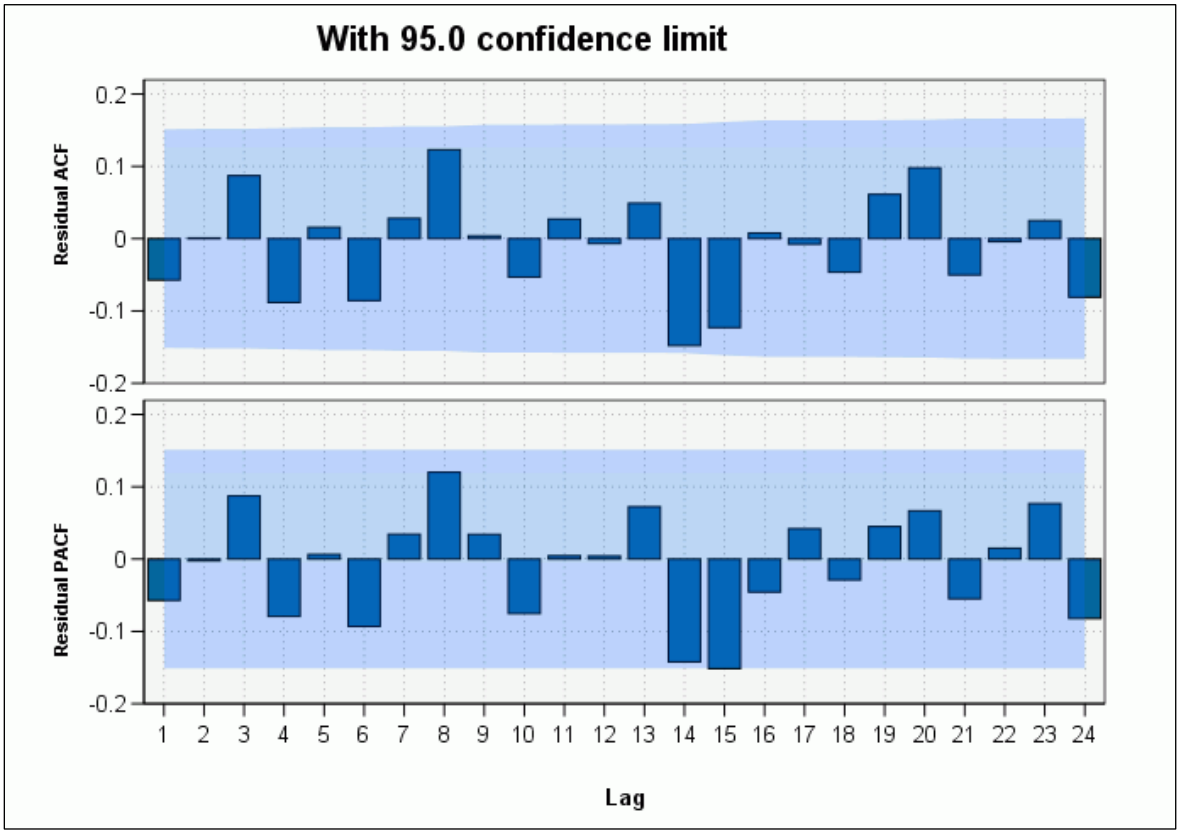
The MAPE (mean absolute percent error) indicates that on average the predictions are off by 8.96%.

The MAE (mean absolute error) is 8.648, the root mean square error 10.755, which means that there are some outliers.

The R-Squared is negative, -0.005. This means that the fit of the simple exponential smoothing model is worse than fitting the series by the series' mean. Stationary R-Squared is 0.525, which represents a moderate fit.

The significance of the Ljung-Box Q statistic is 0.5, meaning that the model is correctly specified.

4. Click the **Correlogram** entry.
   The results appear as follows:



   All bars are within the confidence limits, which means that observations at lags 1 to 24 do not convey information for the observation at time t.

5. Close the **model nugget**.

Task 5. Examine the forecasts.

1. From the **Output** palette, add a **Table** node downstream from the **model nugget**, and then run the **Table** node.

   Scroll down to the end of the output.

   The results appear similar to the following:

Table | Annotations

	Date_	$FutureFlag	SALES_PRODUCT_A	$TS-SALES_PRODUCT_A
165	1995-09-01	0	99.000	98.835
166	1995-10-01	0	89.000	98.836
167	1995-11-01	0	100.000	98.799
168	1995-12-01	0	98.000	98.803
169	1996-01-01	1	$null$	98.800
170	1996-02-01	1	$null$	98.800
171	1996-03-01	1	$null$	98.800
172	1996-04-01	1	$null$	98.800
173	1996-05-01	1	$null$	98.800
174	1996-06-01	1	$null$	98.800
175	1996-07-01	1	$null$	98.800
176	1996-08-01	1	$null$	98.800
177	1996-09-01	1	$null$	98.800
178	1996-10-01	1	$null$	98.800
179	1996-11-01	1	$null$	98.800
180	1996-12-01	1	$null$	98.800

   12 records are added to the dataset, for January 1996 to December 1996. $FutureFlag equals 1 for predictions and 0 for historical data. You can also observe that from the fact that SALES_PRODUCT_A is undefined for these records.

   It may seem odd that the forecasts for each month in 1996 are all the same, 98.800. The forecasts are all the same because of the simple exponential model that is in use. This model states that the series will stay at a certain level. When alpha is near 0, the level does not change throughout the historical period, and thus the forecasts, basically remaining at that level, will be not too far off. When alpha is near 1, as with the next example, you know that the series will move around a certain level, but that that level changes quickly, so forecasting too many future moments is risky.

   In general, it is usually not a good idea to forecast too far ahead with time series models. Future predictions are based on past patterns in the data. When you get too far into the future, the predictions are based on predictions themselves rather than actual historical values.

2. Close the **Table** output window.

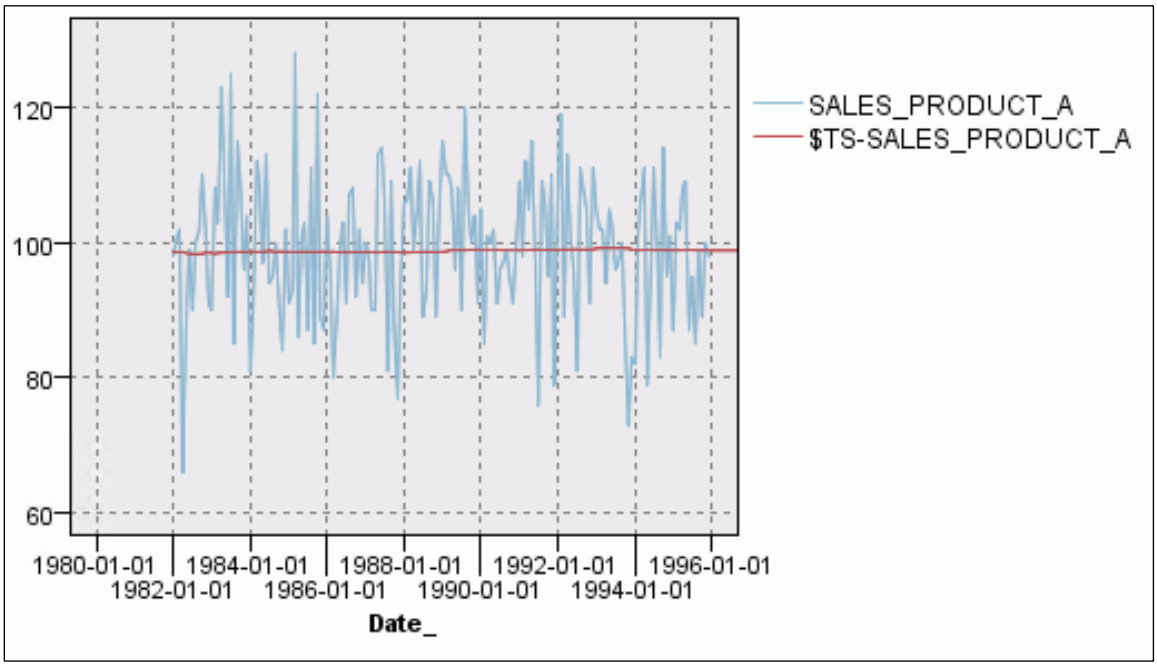
   Now you will create a time plot to see how well the fit values match the historical values in the series.

3. From the **Graphs** palette, add a **Time Plot** node downstream from the **model nugget** (ensure that the option to add future values is enabled on the Settings tab in the model nugget).

   The Time Plot node let you chart the original series, the predictions, and the forecasts.

4. Edit the **Time Plot** node.
5. Beside **Series**, select **SALES_PRODUCT_A** and **$TS-SALES_PRODUCT_A**.
6. Beside **X axis label**, enable the **Custom option**, and then select **Date_**.
7. Clear the **Display series in separate panels** option.
8. Clear the **Normalize** option.
9. Click **Run**.

   The results appear similar to the following:



   The fit values form almost a straight line in the historical part of the series, reflecting the low alpha value of 0.004. This is not entirely unexpected, because, low alpha values tend to dampen out trends and fluctuations in a series. It also helps explain why the forecasts are all the same in this simple exponential smoothing model.

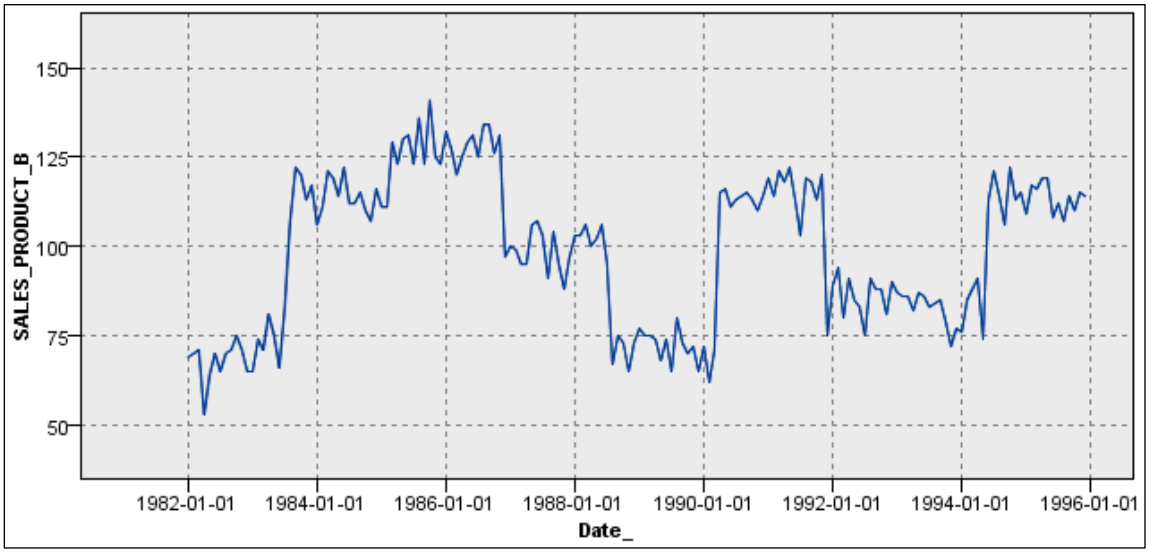
10. Close the **Time Plot** window.

Task 6. Simple exponential smoothing, high value for alpha.

You will go through the same steps, for another field, SALES_PRODUCT_B.

1. Edit the **Time Plot** node that is downstream from the **Type** node, and then replace **SALES_PRODUCT_A** by **SALES_PRODUCT_B**.
2. Click **Run**.

   The results appear similar to the following:



   The series seems to arrive at a certain level for a few periods, and then moves on to the next level; there is no trend and no seasonality. Therefore, a simple exponential model applies, with an alpha level close to 1 to model the quickly-changing levels.

3. Close the **Time Plot** output window.
4. Edit the **Time Series** node.
5. Click the **Fields** tab, if necessary.
6. In the **Targets** box, replace **SALES_PRODUCT_A** by **SALES_PRODUCT_B**.
7. Click the **Build Options** tab.
8. Click the **General** item at the left, if necessary.

   Only exponential smoothing models need to be executed, and a simple exponential smoothing model in particular.
9. Beside **Method**, ensure **Exponential Smoothing** is selected.
10. Under **Model Type**, ensure **Simple** is selected**.**
11. Click **Run**.

    The model nugget is updated with the results for SALES_PRODUCT_B.

Task 7. Examine the results.

1. Edit the **model nugget**.
2. At the left side, click the **Parameter Estimates** entry.

   The results appear as follows:

Parameter Estimates

			Coefficient	Std. Error	t	Significance
SALES_PRODUCT_B	No Transformation	Alpha (Level)	0.712	0.074	9.610	0.000

   The Alpha of 0.712 is fairly large, and is also statistically significant.

   The closer the Alpha is to 1, the more exponential smoothing weights the most recent observation in determining the forecast, and the rapidly the forecasts adjust to the data.

3. At the left side, click the **Model Information** entry.

   The results appear as follows:

Model Information

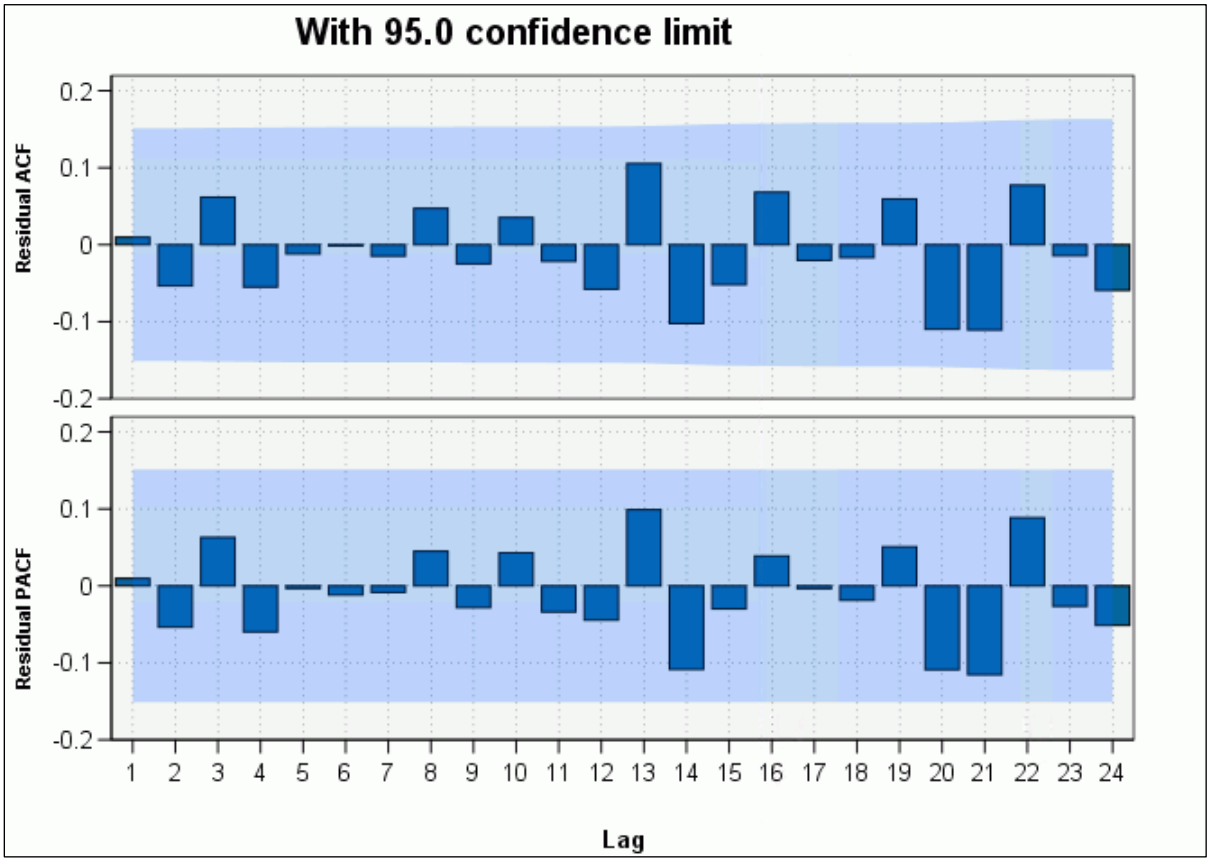
Model Building Method		Exponential Smoothing Simple
Number of Predictors		1
Model Fit	MSE	94.028
	RMSE	9.697
	RMSPE	10.627
	MAE	6.501
	MAPE	6.907
	MAXAE	45.883
	MAXAPE	57.783
	AIC	764.321
	BIC	767.445
	R-Squared	0.789
	Stationary R-Squared	0.070
Ljung-Box Q(#)	Statistic	8.696
	df	17.0
	Significance	0.9

   The MAPE (mean absolute percent error) indicates that on average the predictions are off by only 6.907%.

   The R-Squared is 0.789, so based on this statistic, simple exponential smoothing model seems to fit the series well. Stationary R-Squared is 0.070, which indicates a poor fit, but that may be because this statistic is more appropriate when there is trend in the series, which there is not in this case.

   The significance of the Ljung-Box Q statistic is 0.9, meaning that the model is no autocorrelation.

4. Click the **Correlogram** entry.
   The results appear as follows:



   All bars are within the confidence limits, which means that observations at lags 1 to 24 do not convey information for the observation at time t.

5. Close the **model nugget**.

Task 8. Examine the forecasts.

1. Downstream from the model nugget, re-run the existing **Table** node.

   Scroll down to the end of the output.

   The results appear as follows:

Table | Annotations

	Date_	$FutureFlag	SALES_PRODUCT_B	$TS-SALES_PRODUCT_B
164	1995-08-01	0	107.000	111.738
165	1995-09-01	0	114.000	108.364
166	1995-10-01	0	110.000	112.377
167	1995-11-01	0	115.000	110.684
168	1995-12-01	0	114.000	113.757
169	1996-01-01	1	$null$	113.930
170	1996-02-01	1	$null$	113.930
171	1996-03-01	1	$null$	113.930
172	1996-04-01	1	$null$	113.930
173	1996-05-01	1	$null$	113.930
174	1996-06-01	1	$null$	113.930
175	1996-07-01	1	$null$	113.930
176	1996-08-01	1	$null$	113.930
177	1996-09-01	1	$null$	113.930
178	1996-10-01	1	$null$	113.930
179	1996-11-01	1	$null$	113.930
180	1996-12-01	1	$null$	113.930

   12 records are added to the dataset, for January 1996 to December 1996. $FutureFlag equals 1 for predictions and 0 for historical data. You can also observe that from the fact that SALES_PRODUCT_A is undefined for these records.

   All the predictions for 1996 are the same again, even though this model has a high alpha value. It is important to remember that simple exponential smoothing only models the level, so the level at the last known time will be the forecast for future values. However, with this alpha near 1 you know that the level will change quickly, so it is risky to forecast too far into the future.

2. Close the **Table** output window.

   The Time Plot node let you chart the original series, the predictions, and the forecasts.

3. Edit the **Time Plot** node that is downstream from the model nugget.
4. Beside **Series**, replace the current specifications by **SALES_PRODUCT_B** and **$TS-SALES_PRODUCT_B**.
5. Beside **X axis label**, ensure **Date_** is selected.
6. Ensure the **Display series in separate panels** option is not selected.
7. Ensure the **Normalize** option is not selected.

8. Click **Run**.

   The results appear similar to the following:

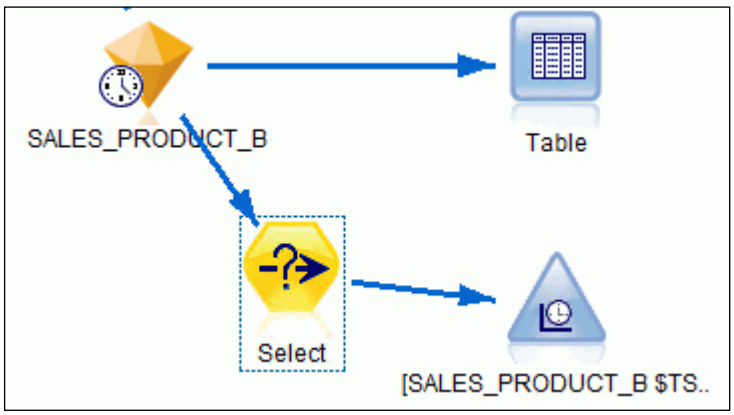
   This time, the predicted value closely matches the historical part of the series, reflecting the high alpha value of 0.712. This is because the closer the alpha value is to 1, the faster it adjusts for fluctuations in the series.

9. Close the **Time Plot** output.

   At this point, you will take a closer look at the forecasts for 1996, this time by using a time plot.

10. From the **Record Ops** palate, take a **Select** node and insert it between the **model nugget** and the **Time Plot** node.

    The results appear as follows:



11. Edit the **Select** node.

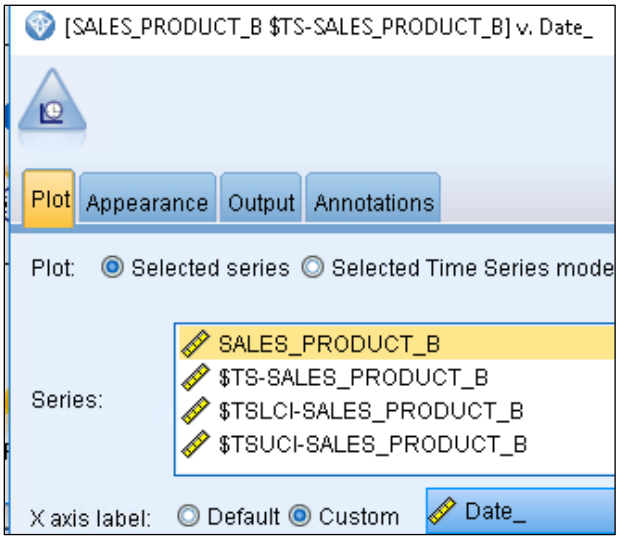
12. In the **Condition** box, type **Date_ >= datetime_date(1995,1,1)**, and then click **OK**.

    This will allow you to focus easier on the forecasted period, rather than having to look at the whole series.

13. Edit the **Time Plot** node.

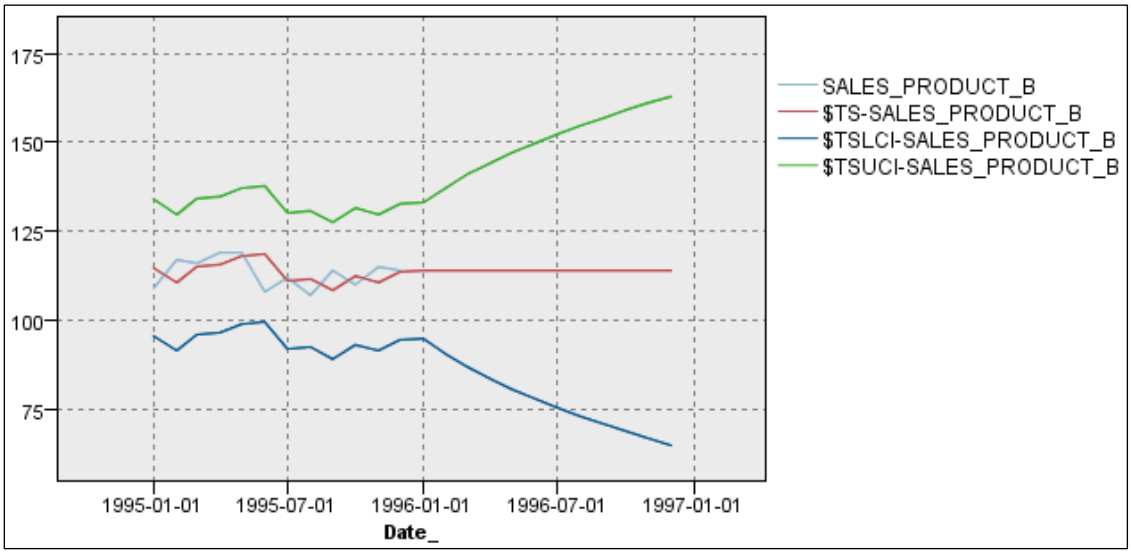
14. In the Series box, select **SALES_PRODUCT_B, $TS-Sales_PRODUCT_B, $TSLCI-SALES_PRODUCT_B, $TSUCI_SALES_PRODUCT_B.**

    The results appear as follows:



15. Click **Run**.

    The results appear as follows:



This time plot not only displays the fit and actual values, but also the upper and lower confidence intervals. Notice how the confidence intervals widen considerably throughout the year, making you progressively less confident in the predictions. This is pretty persuasive evidence how risky it can be predict too far beyond the historical data.

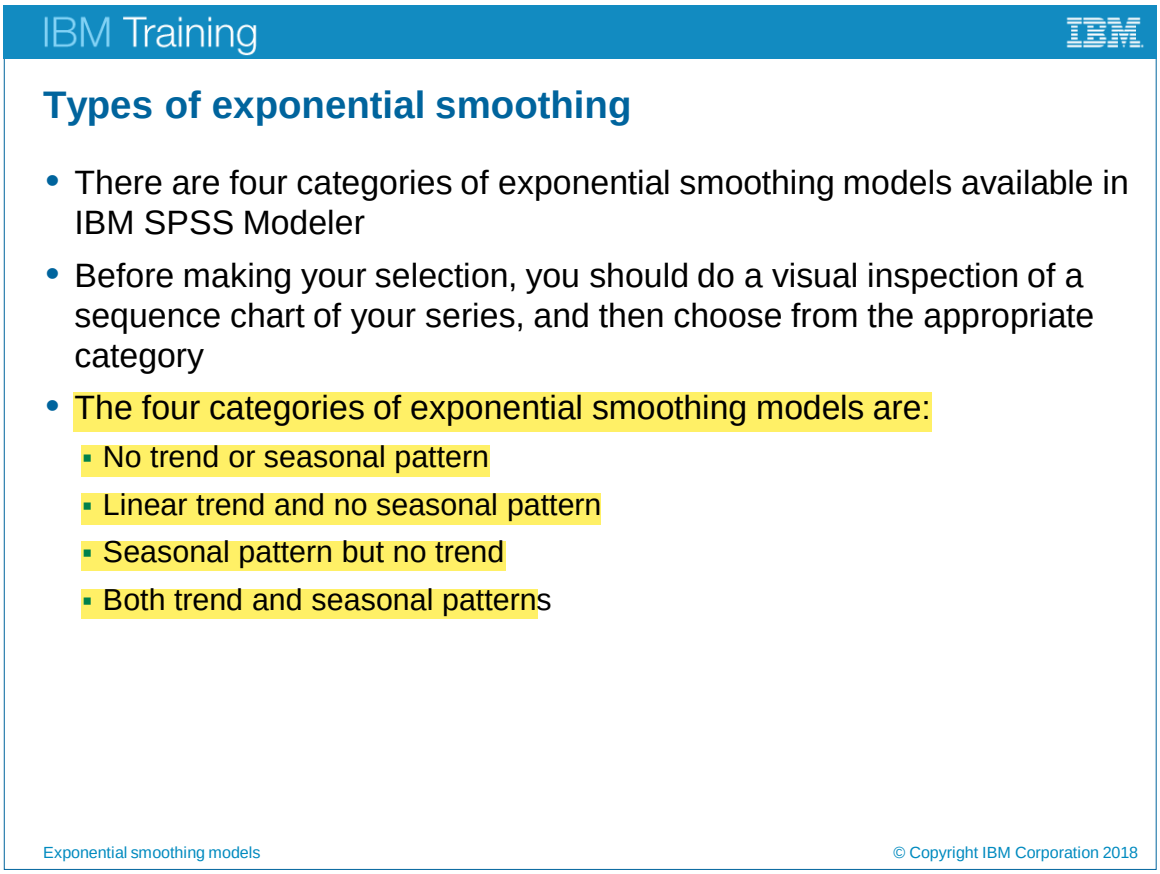
16. Close the **Time Plot** output.

    This completes the demonstration. You will create a clean state for the next demonstration.
17. From the **File** menu, click **Close Stream**. Click **No** when asked to save changes.
18. From the **File** menu, click **New Stream**.

    Leave IBM SPSS Modeler open for the next demonstration.

Results:
You have successfully performed Simple Exponential Smoothing on two different time series

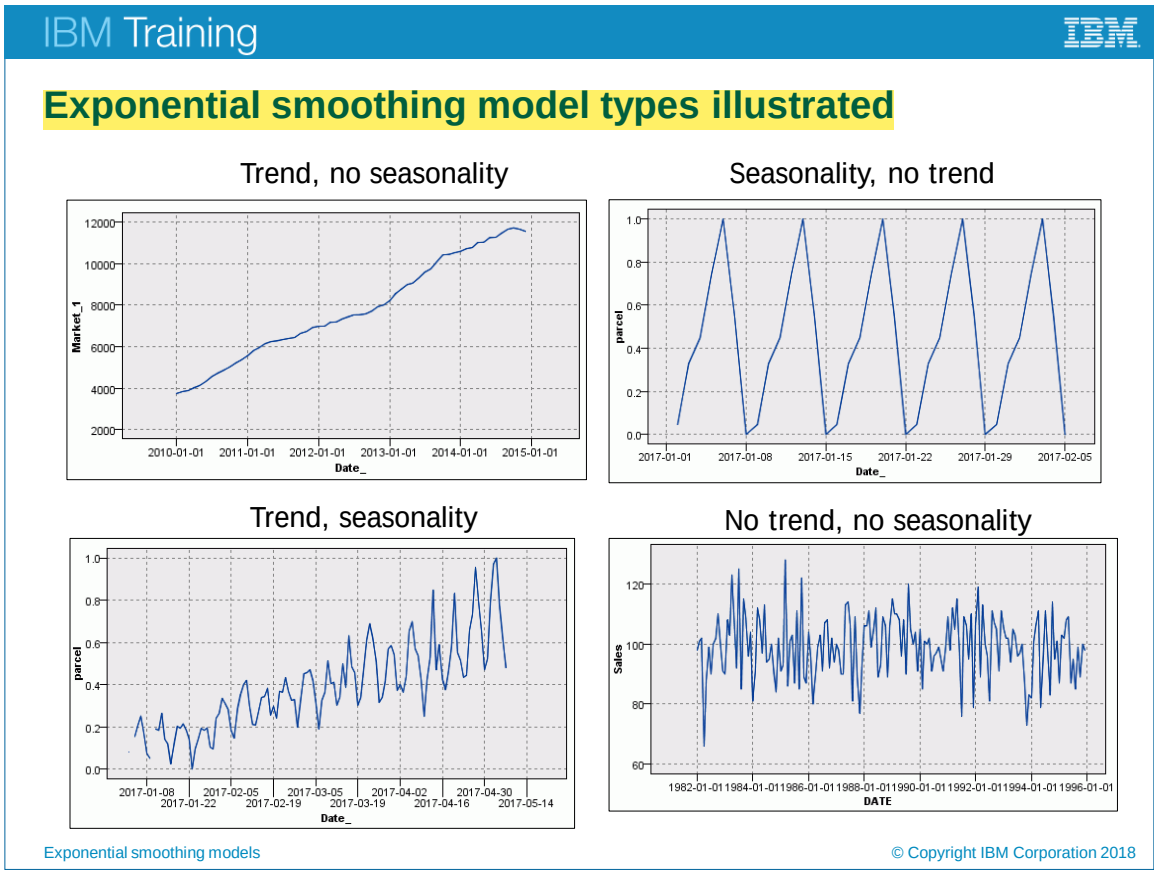
You will find the completed stream in the following folder:
C:\Training\0A028\05-Exponential_Smoothing\Solutions



Types of exponential smoothing

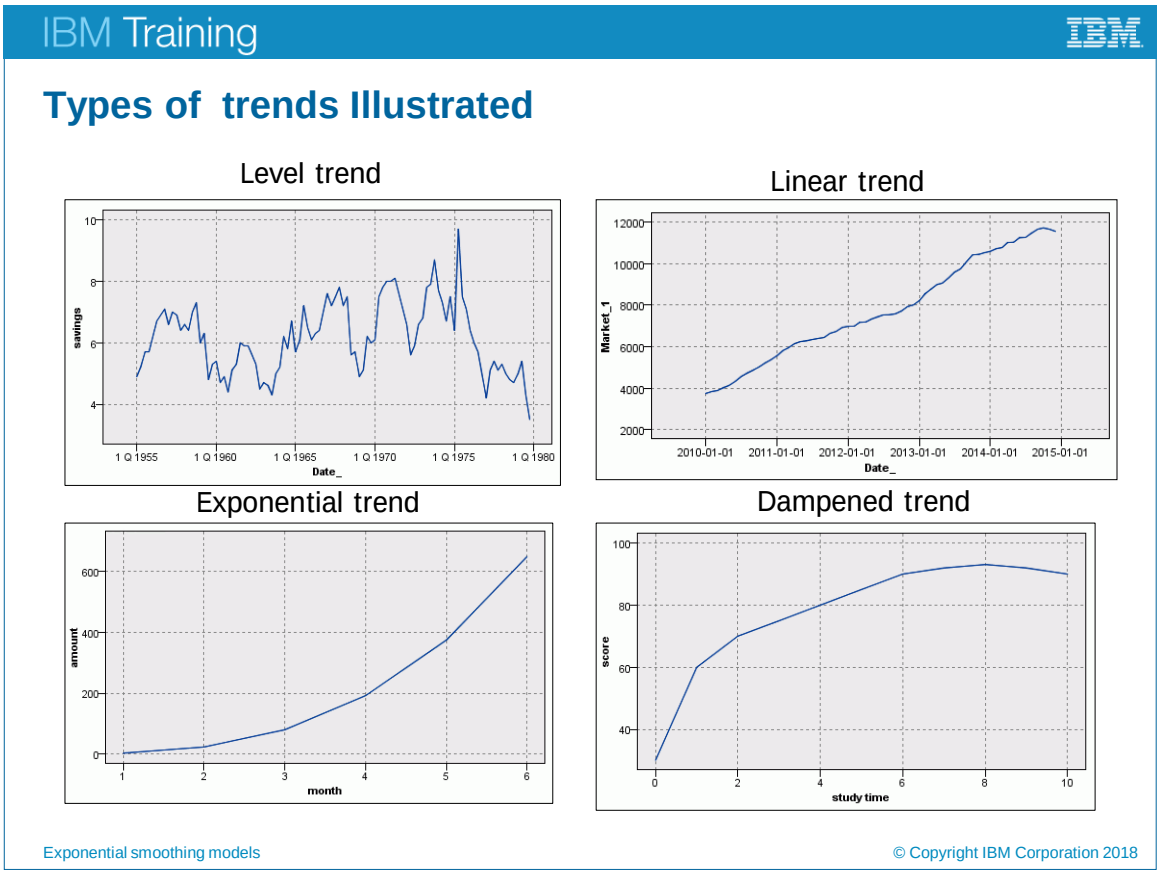
Thus far in this unit, you have taken a look at Simple Exponential Smoothing, which is just one of the many modeling techniques available in the Expert Modeler. You should use that type of model if your series has no trend or seasonal patterns. If your series shows trend, or seasonal patterns, or both, you should make a selection from one of the other categories.

The easiest way to identify the patterns in a series is to create a time plot. Often you can tell from looking at the chart whether or not there are trend or seasonal patterns. If there is no obvious pattern, you can always try several different models and see which one fits best, or use the Expert Modeler to pick the best exponential smoothing model for you.



Exponential Smoothing model types illustrated

To illustrate, the time plot in the upper-left is from data collected over several months on broadband subscriptions. Notice, there is a definite trend upward, but there is no seasonal pattern. This might lead you to select a technique from the class of models that allows trend but not seasonality into the model. In the upper-right is a chart from data collected on weekly sales at a major department store. The chart shows sales peaking toward the middle of each week, but there does not appear to be any trend in overall sales either upward or downward. For that type of data, you should consider a technique that allows seasonality but not trend into the model. The chart in the lower-left is from a private parcel delivery company. Because there is a definite trend upward and also seasonal fluctuation, you should choose a technique that allows both seasonal and trend patterns. And finally, the graph in the lower-right, based on sales of a particular product, shows no trend or seasonal pattern. For this, you should consider a model that does not allow for either trend or seasonality.



Types of trends illustrated

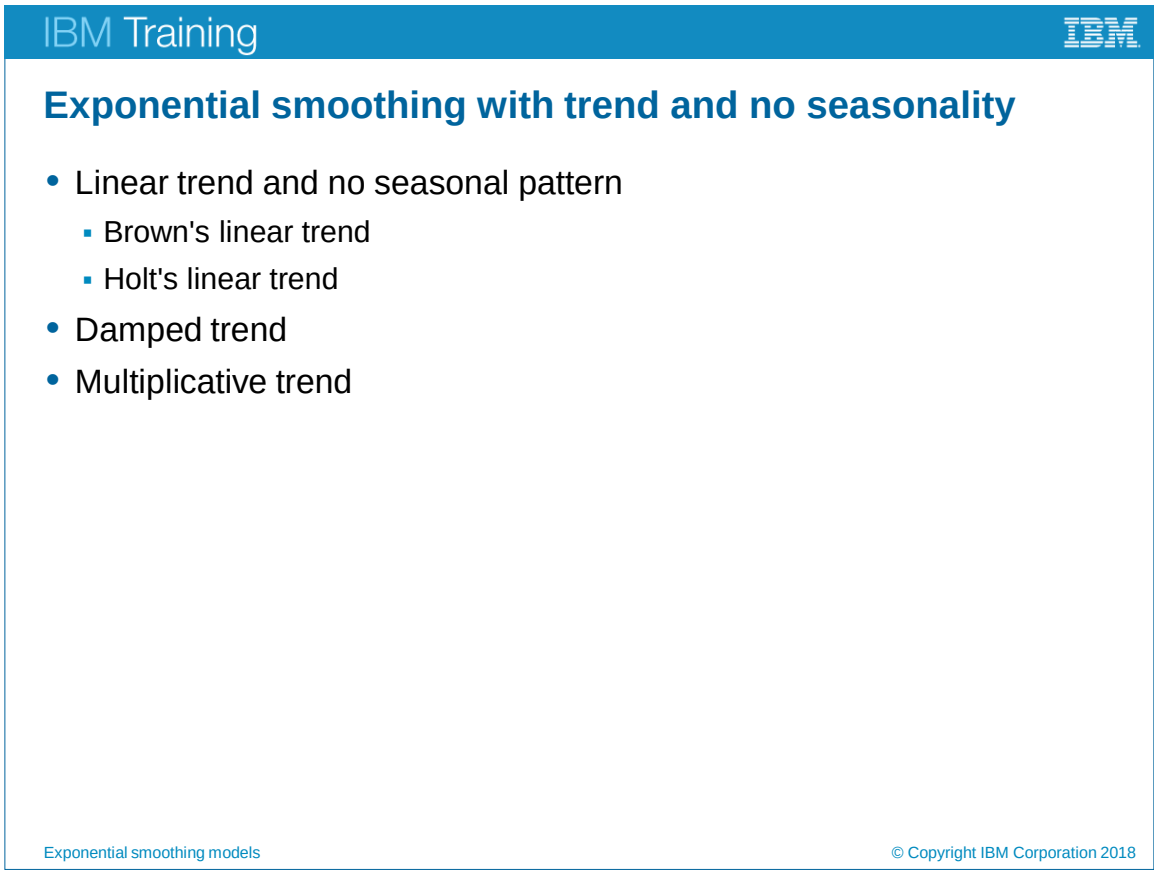
There are four types of trend that can be incorporated within an exponential smoothing model.

- Level trend: The simplest of the exponentially smoothing methods is called "simple exponential smoothing" (SES). This method is suitable for forecasting data with no trend or seasonal pattern.
- Linear trend**:** A linear trend model assumes a constant increase or decrease in the series over time. This has been referred to as a globally constant linear trend model, that is, the trend or slope is constant throughout the series. Exponential smoothing generalizes this to allow for a slowly changing slope. In other words, it allows a locally (but not necessarily globally) constant linear trend model. This means that the increase or decrease is steady over some number of time periods, and the slope needs not to be the same throughout the entire series.

  There are two exponential models supporting linear trend. Brown's model uses the same weight coefficient (α) when updating both the smoothed level and trend effects, while Holt's model is more general in that it permits different weight coefficients to trend and level effects.

  In cases where a linear trend is present, both models might be applied and the goodness-of-fit measures compared.

- Exponential trend: The series has an exponential trend where the series level tends to increase or decrease in value, but at an increasing rate the further the time series progresses.
- Damped Trend: The series increases or decreases in value but at a decreasing rate as the time series proceeds.



Exponential Smoothing with trend and no seasonality

Brown's linear trend

Brown's exponential smoothing model with linear trend can be algebraically represented as follows:

$S_t = S_{t-1} + T_{t-1} - \alpha * e_t$

$T_t = T_{t-1} + \alpha^2 * e_t$

In words, the smoothed level of the series at time t is given by the sum of (1) the smoothed level of the series at time (t-1), (2) the smoothed trend of the series at time (t-1), and (3) a fraction (determined by the value of alpha (α)) of the one-step-ahead forecast error.

The smoothed trend at the end of time t is given by the sum of (1) the smoothed trend at the end of time (t-1) and (2) a fraction (determined by alpha(α) squared) of the one-step-ahead forecast error.

The second equation allows the estimate of the trend to change. An alpha value of 0 implies that no update of the trend and level components are necessary, and a global, constant slope and level will be fit. An alpha value near one suggests the level shifts and the trend involves a linear slope that changes over time. Since the slope is a function of α^2, a given error will influence the slope (by α^2) less than the level (by α).

In Brown's model, shifts in both level and trend are linked to the same weight coefficient (α); this is relaxed in Holt's model, so Brown's model can be treated as a special case of Holt's model.

The forecast for m periods ahead from the time t is given by the following equation:

$\hat{y}_{(t+i)} = S_t + i * T_t$

Where i = 1, …, m, with m the number of forecasts to make.

Holt's linear trend

Holt's exponential smoothing model with linear trend can be algebraically represented as follows:

$S_t = S_{t-1} + T_{t-1} - \alpha * e_t$

$T_t = T_{t-1} + \alpha * \gamma * e_t$

In words, the smoothed level of the series at time t is given by the sum of (1) the smoothed level of the series at time (t-1), (2) the smoothed trend of the series at time (t-1), and (3) a fraction (determined by the value of alpha) of the one-step-ahead forecast error.

The smoothed trend at the end of time t is given by the sum of (1) the smoothed trend at the end of time (t-1) and (2) a fraction (determined by the product of alpha(α) and gamma (γ)) of the one-step-ahead forecast error.

The second equation allows the estimate of the trend to change:

- A gamma value of 0 implies that no update of the trend component is necessary and a constant slope will be fit. Notice that a gamma value of 0 does not state that there is no trend, as one might conclude. A gamma value of 0 states that there is global trend.
- A gamma value near one suggests the trend involves a linear slope that changes over time. Since the weight coefficient for the trend is separate from that for the level, Holt's model is more general than Brown's.

The forecast for m periods ahead from the time t is given by the following equation:

$\hat{y}_{(t+i)} = S_t + i * T_t$

Where i = 1, …, m, with m the number of forecasts to make.

Damped trend

A linear trend model assumes a constant increase or decrease in the series over time, while Brown's and Holt's models permit the trend to change slowly over time.

Damped exponential smoothing applies when there is a linear trend that is dying out over time. In sales this might correspond to a late period of a product cycle in which the rate of increase of sales is declining. In this way, exponential smoothing allows for a specific form of flattening trend.

Damped exponential smoothing can be algebraically represented as follows:

$S_t = S_{t-1} + \varphi * T_{t-1} + \alpha * e_t$

$T_t = \varphi * T_{t-1} + \alpha * \gamma * e_t$

In words, the smoothed level of the series at time t is given by the sum of (1) the smoothed level of the series at time (t-1), (2) the damped trend (damped by the value of phi (φ)) of the series at time (t-1), and (3) a fraction (determined by the value of alpha (α)) of the one-step-ahead forecast error.

The smoothed trend at the end of time t is given be the sum of (1) the damped trend (damped by the value of phi) at the end of time (t-1) and (2) a fraction (determined by the product of alpha and gamma (γ)) of the one-step-ahead forecast error.

The coefficient for damping, phi, is separate from the weight coefficient for trend, gamma, allowing the estimate of the trend to change and explicitly taking damping into account.

A gamma value of zero implies that no update of the trend component is necessary, and the slope will change only due to damping. A gamma value near one suggests the trend involves a linear slope that changes over time. The closer phi is to one, the more gradual is the effect on damping (a phi value of one indicates no damping occurs).

The forecasts ahead from the time t is given by the following equation:

$\hat{y}_{(t+i)} = S_{(t)} + \sum_{(k = 1 \text{ to } i)} \varphi^k * T_{(t)}.$

Where i = 1, …, m, with m the number of forecasts to make.

Multiplicative trend

This model is appropriate for a series in which there is a trend that changes with the magnitude of the series and no seasonality. Its relevant smoothing parameters are level and trend.

IBM Training

Demonstration 2

Exponential smoothing with trend

Demonstration 2: Exponential smoothing with trend

Demonstration 2: Exponential smoothing with trend

Purpose:
You have monthly data on user subscriptions to broadband services for a national provider from January 1999 to December 2003. You intend to create an exponential smoothing model and use it to forecast 12 months into the future.

Stream file: **unit_5_demonstration_2_start.str**

Folder: **C:\Training\0A028\05-Exponential_Smoothing\Start**

Task 1. Start and configure IBM SPSS Modeler 18.1.1.

1. From the **Start** menu, expand **IBM SPSS Modeler 18.1**, and then click **IBM SPSS Modeler 18.1**.

   Note: IBM SPSS Modeler 18.1.1 displays "IBM SPSS Modeler 18.1" in the Start menu and program name.
2. Click **Cancel** to close the **Welcome** dialog box.

   If you have already set the working directory in a previous demonstration or exercise, you can skip to Task 2.
3. From the **File** menu, click **Set Directory**.
4. Beside **Look in**, navigate to the **C:\Training\0A028** folder, and then click **Set**.

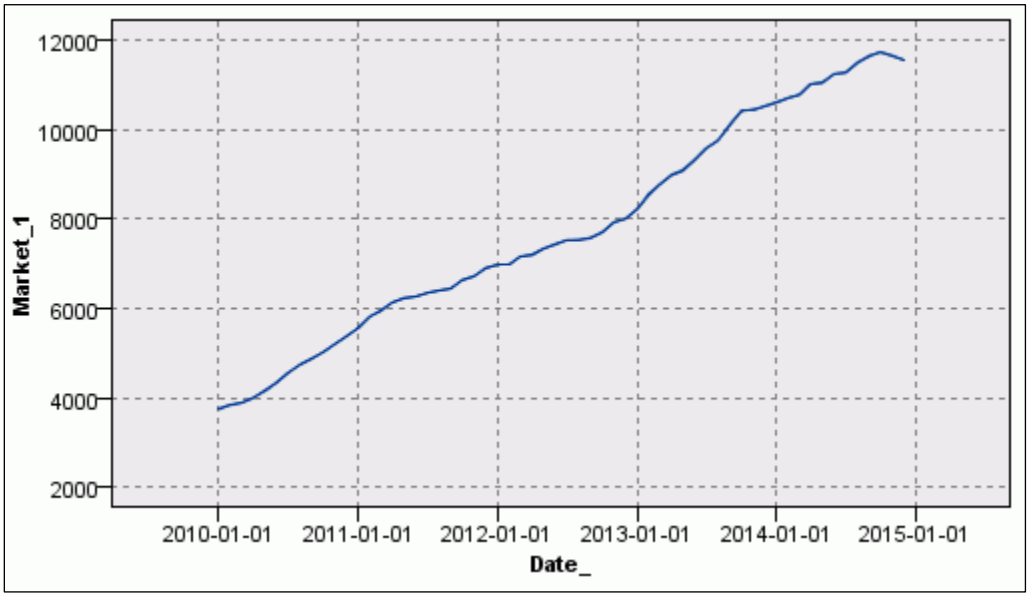
Task 2. Examine the data.

1. From the **File** menu, click **Open Stream**, and then select the file, **unit_5_demonstration_2_start.str**.
2. Run the **Table** node.

   The dataset stores the broadband subscriptions for several markets from the period January 2010 to December 2014.
3. Close the **Table** output window.
4. From the **Graphs** palette, select the **Time Plot**, and add it downstream from the **Type** node.
5. Edit the **Time Plot** node.
6. Besides **Series**, select **Market_1.**
7. Besides **X axis label**, enable the **Custom** option, and then select **Date_**.
8. Clear the **Normalize** option.

9. Click **Run**.

   The results appear as follows:



   The series is a straight line, without seasonality. Therefore, either a Holt's linear trend or a Brown's linear trend model would seem appropriate. You will test your theory by using the Expert Modeler select the best exponential smoothing model for you.

10. Close the **Time Plot** output window.

Task 3. Create a model.

1. Click the **Modeling** palette, and then, at the left side, click **All**.
2. Add a **Time Series** node downstream from the **Type** node.
3. Edit the **Time Series** node.
4. Click the **Fields** tab, if necessary.
5. Enable the **Use custom field assignments** option.
6. Move **Market_1** into the **Targets** box.
7. Click the **Data Specifications** tab.
8. Click the **Observations** item on the left, if necessary.

   The observations are defined by the field Date_ and represent months.

9. Ensure that the **Observations are specified by a date/time field** option is enabled.
10. For **Date/time field**, select **Date_**.
11. Beside **Time interval**, select **Months**.
12. Click the **Build Options** tab.

13. Click the **General** item at the left, if necessary.
    Only exponential smoothing models need to be executed.
14. Under **Model Type**, select **Exponential smoothing models only**.
    You will add forecasts for the next 12 months.
15. Click the **Model Options** tab.
16. Under **Forecast**, select the **Extend records into the future** option, and then replace the default value with a value of **12**.
17. Click **Run**.
    A model nugget is generated that stores the results and which can be used to score records.

Task 4. Examine the results.

1. Edit the **model nugget**.
2. At the left side, click the **Model Information** entry.
   The results appear as follows:

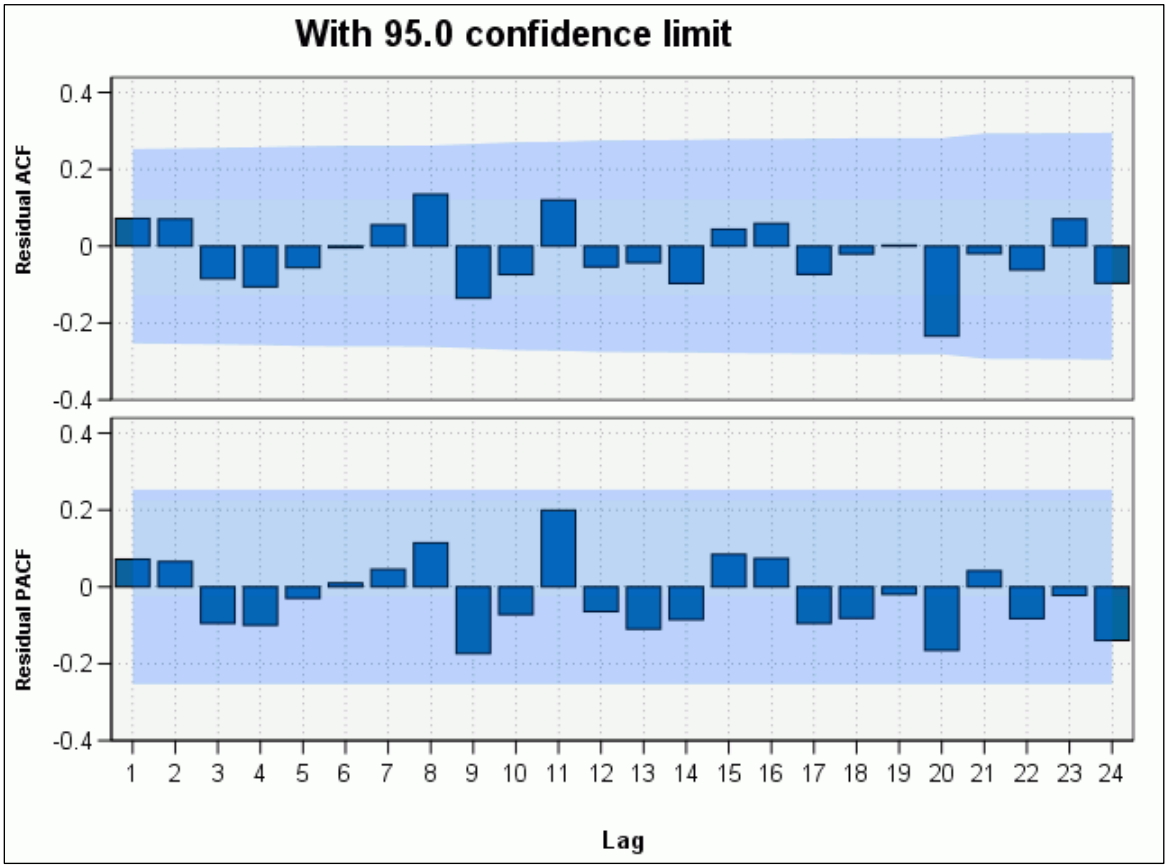
Model Information

Model Building Method		Exponential Smoothing
		Holts linear trend
Number of Predictors		1
Model Fit	MSE	8,188.446
	RMSE	90.490
	RMSPE	1.086
	MAE	73.765
	MAPE	0.939
	MAXAE	223.839
	MAXAPE	2.141
	AIC	542.595
	BIC	546.783
	R-Squared	0.999
	Stationary R-Squared	0.262
Ljung-Box Q(#)	Statistic	8.517
	df	16.0
	Significance	0.9

The Expert Modeler picked the Holts linear model because it had the best fit. R-Square is almost 1.0, but a better fit measure for data that shows a trend is Stationary R-Square, which equals 0.262. This indicates that the model explains more than 26% of the variance in the series.

The significance of the Ljung-Box statistic is 0.9, meaning that you may assume that the model is specified correctly.

3. At the left side, click the **Correlogram** entry.
   The results appear as follows:



   ACF and PACF are all within their confidence bands.

4. At the left side, click the **Parameter Estimates** entry.
   The results appear as follows:

Parameter Estimates

			Coefficient	Std. Error	t	Significance
Market_1	No Transformation	Alpha (Level)	1.000	0.138	7.239	0.000
		Gamma (Trend)	0.300	0.135	2.223	0.030

   The Alpha is estimated at 1.000 which means that more recent periods have a great influence on the current series value. Gamma, which measures the change in trend is significant which means that the trend is not constant over the length of the series.

5. Close the **model nugget**.

6. From the **Output** palette, add a **Table** node downstream from the model nugget, and then run the **Table** node.

   $FutureFlag indicates whether a record is part of the historical data or represents a future point in time. The field is 0 for the historical data.

7. Scroll down to the end of the output.

   The results appear as follows:

Table | Annotations

	Date_	$FutureFlag	Market_1	$TS-Market_1	$TSLCI-Market_1	$TSUCI-Market_1
53	2014-05-01	0	11040.350	11163.345	10982.209	11344.480
54	2014-06-01	0	11244.051	11151.686	10970.550	11332.821
55	2014-07-01	0	11270.266	11383.106	11201.971	11564.242
56	2014-08-01	0	11488.729	11375.453	11194.318	11556.589
57	2014-09-01	0	11656.669	11627.914	11446.779	11809.050
58	2014-10-01	0	11731.133	11804.494	11623.358	11985.629
59	2014-11-01	0	11656.308	11856.937	11675.802	12038.073
60	2014-12-01	0	11549.200	11721.877	11540.741	11903.012
61	2015-01-01	1	$null$	11562.916	11381.780	11744.051
62	2015-02-01	1	$null$	11576.619	11279.505	11873.733
63	2015-03-01	1	$null$	11590.322	11175.206	12005.438
64	2015-04-01	1	$null$	11604.025	11064.709	12143.340
65	2015-05-01	1	$null$	11617.728	10947.047	12288.409
66	2015-06-01	1	$null$	11631.431	10822.053	12440.809
67	2015-07-01	1	$null$	11645.134	10689.824	12600.444
68	2015-08-01	1	$null$	11658.837	10550.548	12767.126
69	2015-09-01	1	$null$	11672.540	10404.442	12940.638
70	2015-10-01	1	$null$	11686.243	10251.728	13120.758
71	2015-11-01	1	$null$	11699.946	10092.616	13307.275
72	2015-12-01	1	$null$	11713.649	9927.309	13499.989

   12 records are added to the dataset, for January 1996 to December 1996. $FutureFlag equals 1 for these 12 records, indicating that these are predictions into the future. You can also observe that from the fact that Market_1 is undefined for these records.

8. Close the **Table** output window.

   The Time Plot node let you chart the original series, the predictions, and the forecasts.

9. From the **Graphs** palette, add a **Time Plot** node downstream from the **model nugget** (ensure that the option to add future values is enabled on the Settings tab in the model nugget).

10. Edit the **Time Plot** node.

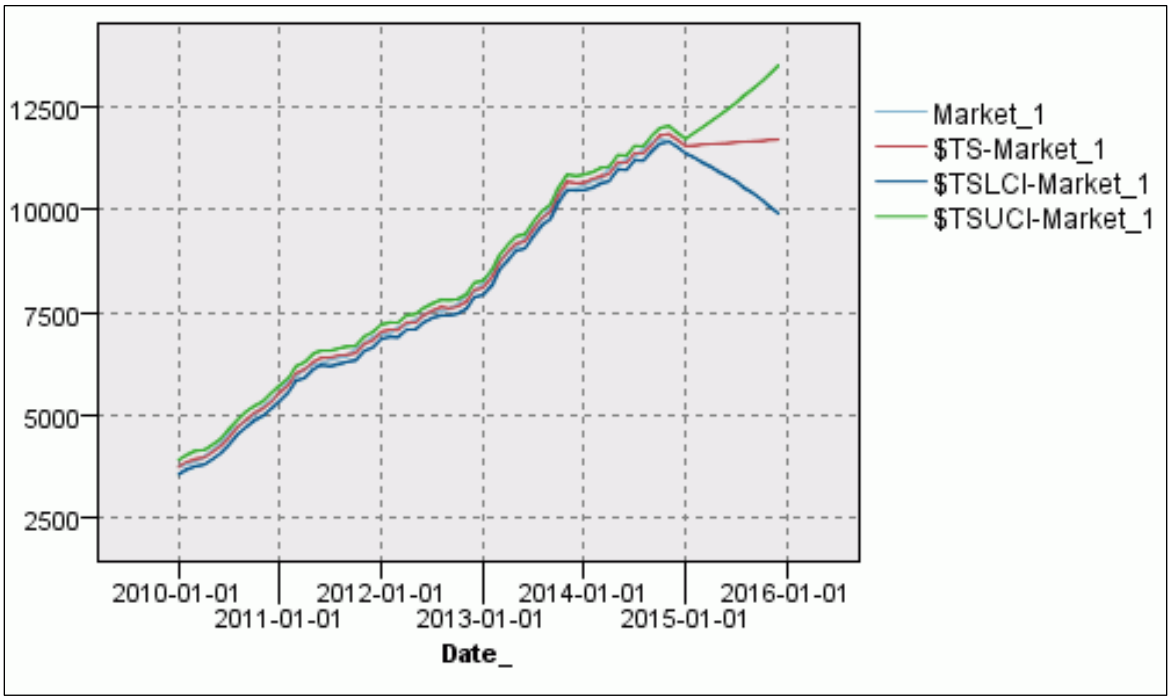
11. Beside Series, select **Market_1**, **$TS-Market_1**, **$TSLCI-Market_1**, and **$TSUCI-Market_1**.

12. Beside **X axis label**, enable the **Custom** option, and then select **Date_**.

13. Clear the **Display series in separate panels** option.

14. Clear the **Normalize** option.
15. Click **Run**.

    The results appear as follows:



    The fit values follow the data very well. However, the result makes is crystal clear that exponential smoothing models are best for short-term forecasts, and certainly not ones that go beyond two or three time periods. Note that the confidence limits grow larger with each month..

16. Close the **Time Plot** output window.

    This completes the demonstration. You will create a clean state for the next demonstration.

17. From the **File** menu, click **Close Stream**. Click **No** when asked to save changes.
18. From the **File** menu, click **New Stream**.

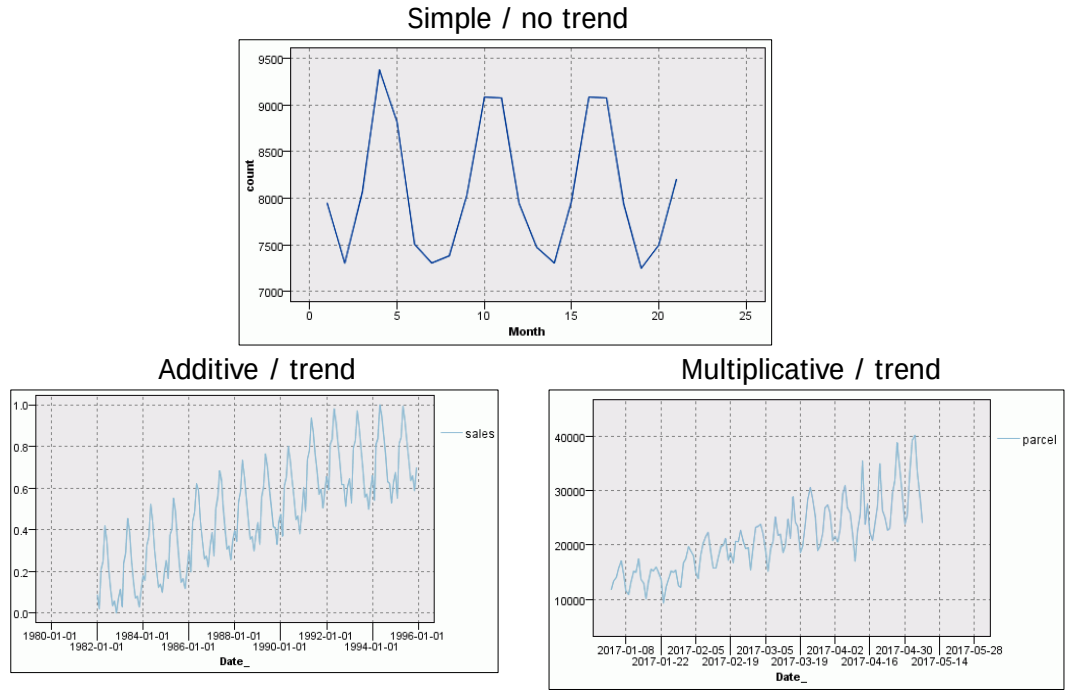
    Leave IBM SPSS Modeler open for the next demonstration.

Results:
You have successfully performed Exponential Smoothing on a series with trend.

You will find the completed stream in the following folder:
C:\Training\0A028\05-Exponential_Smoothing\Solutions

IBM Training

Exponential smoothing with seasonality illustrated



Exponential smoothing models

Exponential smoothing with seasonality illustrated

There are three types of seasonality that are allowed for in exponential smoothing models.

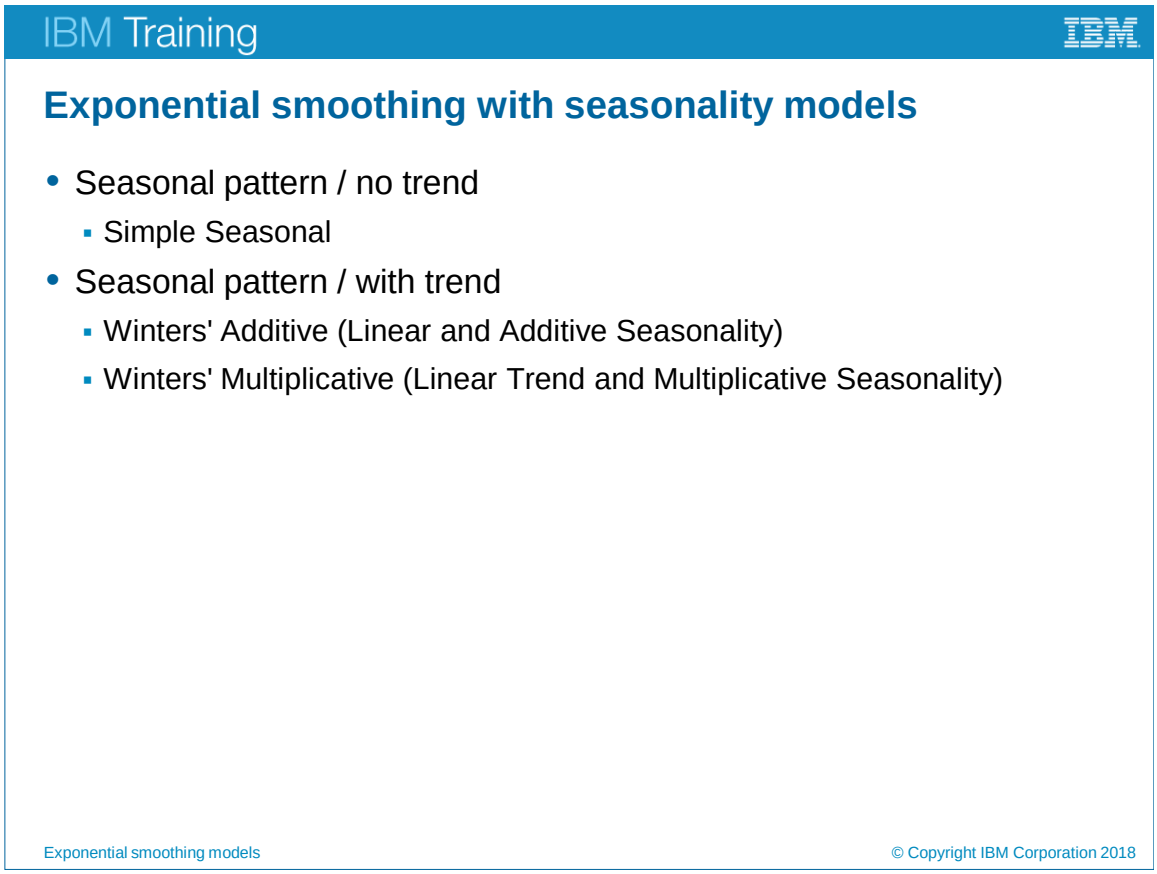
The chart on top displays a seasonal pattern but no trend:

- Simple seasonality. A seasonal pattern exists in the series which is influenced by seasonal factors (such as, the quarter of the year, the month, or day of the week) but there is no trend.

The next two charts show both trend and seasonality:

- Additive seasonality. Additive seasonality describes a series with a seasonal pattern that maintains the same magnitude when the series level increases or decreases. Notice that in the graph above labelled Additive, the magnitude of the seasonal bumps remain constant over the length of the series.
- Multiplicative seasonality. If the seasonal model patterns become more (less) pronounced when the series values increase (decrease), then the seasonal pattern is multiplicative. In the chart labelled Multiplicative, the size of the seasonal bumps become progressively larger as time goes on.

All these models include a parameter named delta, which controls the relative weight given to recent observations in estimating seasonality. It ranges from 0 to 1, with values near 1 giving higher weight to recent values. As with gamma, a delta of 0 does not imply there is no seasonality, but that the seasonality is constant over time.



Exponential smoothing with seasonality models

Simple Seasonal (Additive)

Additive seasonality describes a series with a repeating, seasonal pattern that maintains the same magnitude if the series level increases or decreases.

Seasonal exponential smoothing incorporates both local level shifts and seasonality into the model. Consistent with how trend is handled in exponential smoothing, seasonal exponential smoothing allows for slowly changing seasonal effects. In other words, it allows for a locally (but not necessarily globally) constant seasonal model.

The following equations represent the error-correction form of the exponential smoothing model with additive seasonality:

$$S_{(t)} = S_{(t-1)} + \alpha * e_{(t)}$$

$$I_{(t)} = I_{(t-p)} + \delta * (1-\alpha) * e_{(t)}$$

And predicted $y_{(t+m)} = S_{(t)} + I_{(t-p+m)}$, where α and δ represent the alpha and delta weight coefficients. $S_{(t)}$ is the smoothed level of the series computed after $y_{(t)}$ is observed, $I_{(t)}$ is the smoothed seasonal factor at the end of period t, m is the number of periods in the forecast lead time, $e_{(t)}$ is the one-step-ahead forecast error from the previous period, and the predicted $y_{(m)}$ is the forecast for m periods ahead of the period t.

Winters' additive (Linear Trend and Additive Seasonality)

If there are signs that the series increases (decreases) over time and there is a repeating seasonal pattern that maintains the same magnitude when the series level increase (decreases), the Winters' additive exponential smoothing model should be specified. It accommodates linear trend and additive seasonality.

Recall that simple exponential smoothing relies on a single parameter; alpha. We expressed the model in two forms. In the recurrence form, the prediction for the next observation is a weighted combination of the most recent observation and the "history" of the series. In the error-correction form, the prediction for the next observation is a weighted combination of the prediction for the current case and the error in predicting the current case from the last case. Winters' exponential smoothing for seasonal series with linear trend extends the above approach to incorporate three smoothing parameters: alpha for level, gamma for trend, and delta for seasonality. The advantage of the seasonal exponential smoothing model over trend regression is that exponential smoothing models can model series with changing trend seasonality (stochastic trend and seasonality) while regression assumes that trend and seasonal effects are constant (deterministic or globally constant trend and seasonality) over the observed span of the series.

The following equations are the algebraic representations of the error-correction form of the exponential smoothing model with linear trend and additive seasonality:

$$S_{(t)} = S_{(t-1)} + T_{(t-1)} + \alpha * e_{(t)}$$

$$T_{(t)} = T_{(t-1)} + \alpha * \gamma * e_{(t)}$$

$$I_{(t)} = I_{(t-p)} + \delta * (1-\alpha) * e_{(t)}$$

And predicted $y_{(t+m)} = S_{(t)} + m * T_{(t)} + I_{(t-p+m)}$

Here α, γ, and δ represent the alpha, gamma, and delta weight coefficients, $S_{(t)}$ is the smoothed level of the series computed after $y_{(t)}$ is observed, $T_{(t)}$ is the smoothed trend at the end of period t, $I_{(t)}$ is the smoothed seasonal factor at the end of period t, m is the number of periods in the forecast lead time, p is the number of time periods in one season, $e_{(t)}$ is the one-step-ahead error from the previous period, and the predicted $y_{(m)}$ is the forecast for m periods ahead of the current period t.

Winters' multiplicative (Linear Trend and Multiplicative Seasonality)

If there are signs that the series increases (decreases) over time and there is a repeating seasonal pattern that increases (decreases) in magnitude when the series level increases (decreases), then Winters' multiplicative exponential smoothing model should be specified. Winters' multiplicative model accommodates linear trend and multiplicative seasonality; the seasonal pattern becomes more (less) pronounced when the series values increase (decrease).

Exponential smoothing can also accommodate multiplicative seasonality, which can, in the fashion discussed with trend and additive seasonality, vary over time. Recall that multiplicative seasonality implies that seasonal effects increase with increasing mean level of the series. That is, multiplicative effects are proportional to the local mean level of the series.

The following equations are the algebraic representations of the error-correction form of Winters' exponential smoothing model with linear trend and multiplicative seasonality:

$$S_{(t)} = S_{(t-1)} + T_{(t-1)} + \alpha * e_{(t)}/I_{(t-p)}$$

$$T_{(t)} = T_{(t-1)} + \alpha * \gamma * e_{(t)}/I_{(t-p)}$$

$$I_{(t)} = I_{(t-p)} + \delta * (1-\alpha) * e_{(t)}/S_{(t)}$$

And predicted $y_{(t+m)} = (S_{(t)} + m * T_{(t)}) * I_{(t-p+m)}$

Here α, γ, and δ represent the alpha, gamma, and delta weight coefficients, $S_{(t)}$ is the smoothed level of the series computed after $y_{(t)}$ is observed, $T_{(t)}$ is the smoothed trend at the end of period t, $I_{(t)}$ is the smoothed seasonal factor at the end of period t, m is the number of periods in the forecast lead time, p is the number of time periods in one season, $e_{(t)}$ is the one-step-ahead forecast error from the previous period, and the predicted $y_{(m)}$ is the forecast for m periods ahead from the period t.

If you examine the formulas for $S_{(t)}$ or $T_{(t)}$ under additive and multiplicative seasonal models, you see that in the former case $e_{(t)}$ is multiplied by the smoothing coefficients in order to update $S_{(t)}$ or $T_{(t)}$. However, in the latter case $e_{(t)}/I_{(t-p)}$ is multiplied by the smoothing coefficients. So the influence of an error is relative to the seasonal effect one season in the past. Also, the predicted series value $y_{(m)}$ involves a multiplicative seasonal factor $((S_{(t)} + m * T_{(t)}) * I_{(t-p+m)})$.

Thus exponential smoothing can accommodate a variety of pure time series models involving trend and seasonality.

One important point to note about exponential smoothing with seasonal components is that it is recommended that you have at least four cycles worth of data before fitting exponential smoothing models with seasonal effects. For example, if the data that you collect are quarterly and you wish to use exponential smoothing to apply a seasonal model, then it is suggested that you have four or more years worth of data. Exponential smoothing is better able to detect shifts in seasonal effects when there are more instances of the seasons.

IBM Training

Demonstration 3

Exponential smoothing with trend and seasonality

Demonstration 3: Exponential smoothing with trend and seasonality

Demonstration 3: Exponential smoothing with trend and seasonality

Purpose:
You have data on new monthly housing permits issued over the period January 1990 to September 2005. From looking at the data, you notice a trend upwards in permits, and seasonal patterns, with more permits issued in warmer months and fewer in colder months. You would like to use Exponential Smoothing to forecast the number of permits for 2006 and need a model that allows for trend and seasonality.

Stream file: **unit_5_demonstration_3_start.str**

Folder: **C:\Training\0A028\05-Exponential_Smoothing\Start**

Task 1. Start and configure IBM SPSS Modeler 18.1.1.

1. From the **Start** menu, expand **IBM SPSS Modeler 18.1**, and then click **IBM SPSS Modeler 18.1**.

   Note: IBM SPSS Modeler 18.1.1 displays "IBM SPSS Modeler 18.1" in the Start menu and program name.
2. Click **Cancel** to close the **Welcome** dialog box.

   If you have already set the working directory in a previous demonstration or exercise, you can skip to Task 2.
3. From the **File** menu, click **Set Directory**.
4. Beside **Look in**, navigate to the **C:\Training\0A028** folder, and then click **Set**.

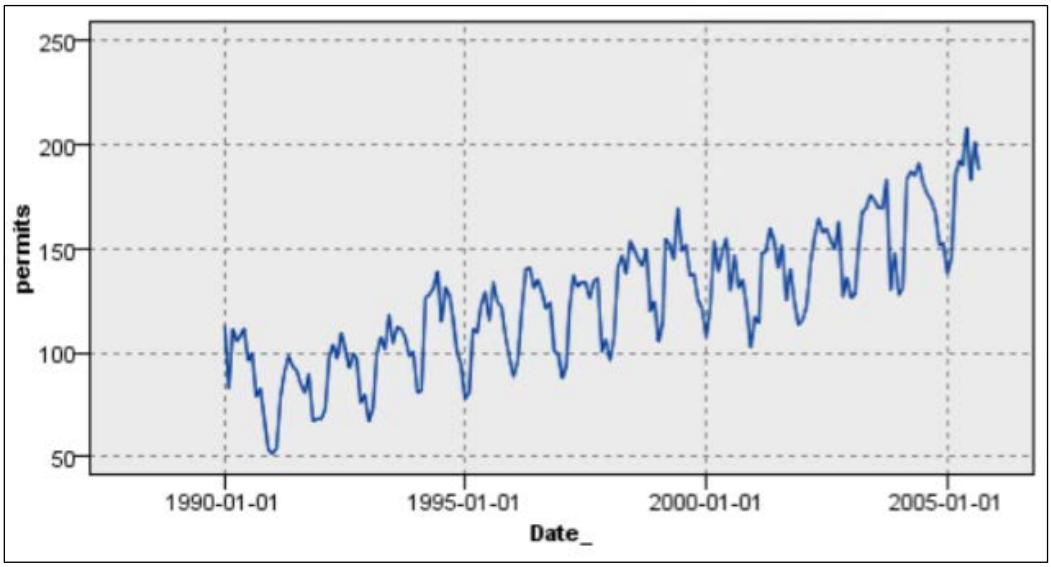
Task 2. Examine the data.

1. From the **File** menu, click **Open Stream**, and then select **unit_5_demonstration_3_start.str**.
2. Run the **Table** node.

   The dataset stores the housing permits issued monthly from January 1990 through September 2005.
3. Close the **Table** results window.
4. From the **Graphs** palette, select the **Time Plot**, and add it downstream from the **Type** node.
5. Edit the **Time Plot** node.
6. Besides **Series**, select **permits.**

7. Besides **X axis label**, enable the **Custom** option, and then select **Date_**.
8. Clear the **Normalize** option.
9. Click **Run**.

   The results appear as follows:



   The plot is highly regular, exhibiting strong seasonality. It is true, if you examine the data closely, that the peak month for housing permits varies from year to year, but more housing permits are issued during warmer months, and fewer during colder months, every year. Non-constant level and trend also characterize the series, as the number of permits issues continues to climb from 1990 to 2005, although it isn't clear that the increase is linear.

   The time plot suggests that the appropriate model is either the Winters' additive or the Winters' multiplicative model. Both models allow for trend and seasonality, but because it is not clear from the time plot whether the seasonality is additive or multiplicative, you will need to try both models to determine which one fits better.

10. Close the results window.

Task 3. Create a Winters' additive model.

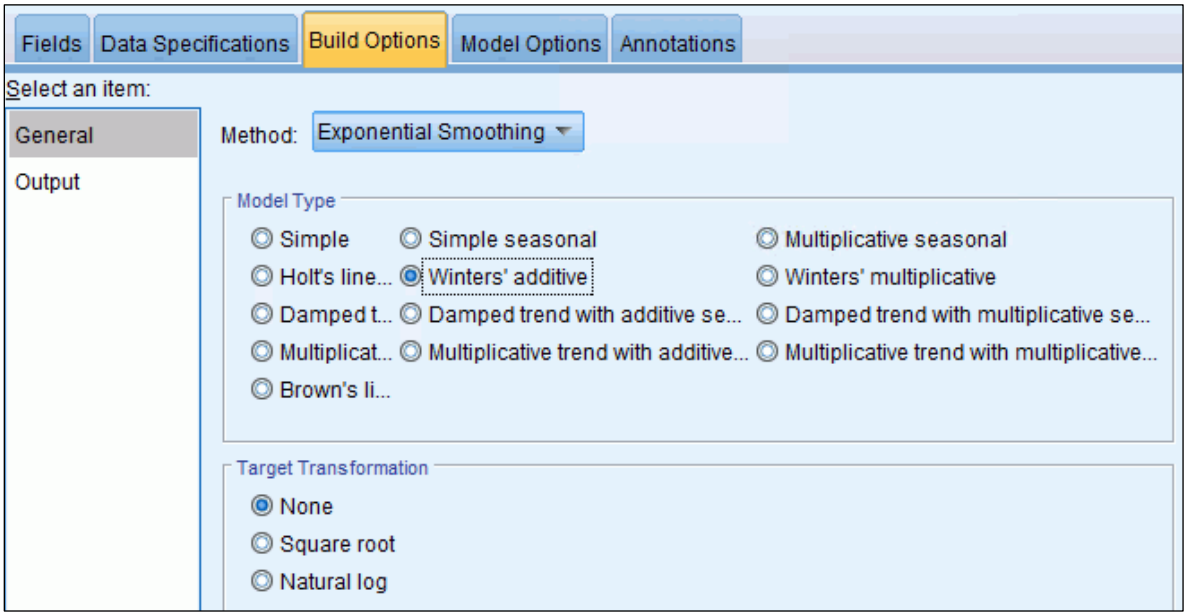
1. Click the **Modeling** palette, and then, at the left side, click **All**.
2. Add a **Time Series** node downstream from the **Type** node.
3. Edit the **Time Series** node.
4. Click the **Fields** tab, if necessary.
5. Enable the **Use custom field assignments** option.
6. Move **permits** into the **Targets** box.

7. Click the **Data Specifications** tab.
8. Click the **Observations** item on the left, if necessary.

   The observations are defined by the field Date_ and represent months.
9. Ensure that the **Observations are specified by a date/time field** option is enabled.
10. For **Date/time** field, select **Date_**.
11. Beside **Time intervals**, select **Months**.
12. Click the **Build Options** tab.
13. Click the **General** item at the left, if necessary.
14. Click the **Method** dropdown, and then select **Exponential Smoothing**.
15. Under **Model Type**, select **Winters' additive.**

    This model is appropriate for a series in which there is a linear trend and a seasonal effect that is constant over time.

    The results appear as follows:



16. Click **Run**.

    A model nugget is generated that stores the results.

Task 4. Examine the results of Winters' additive model.

1. Edit the **model nugget**.
2. At the left side, click the **Model Information** entry.

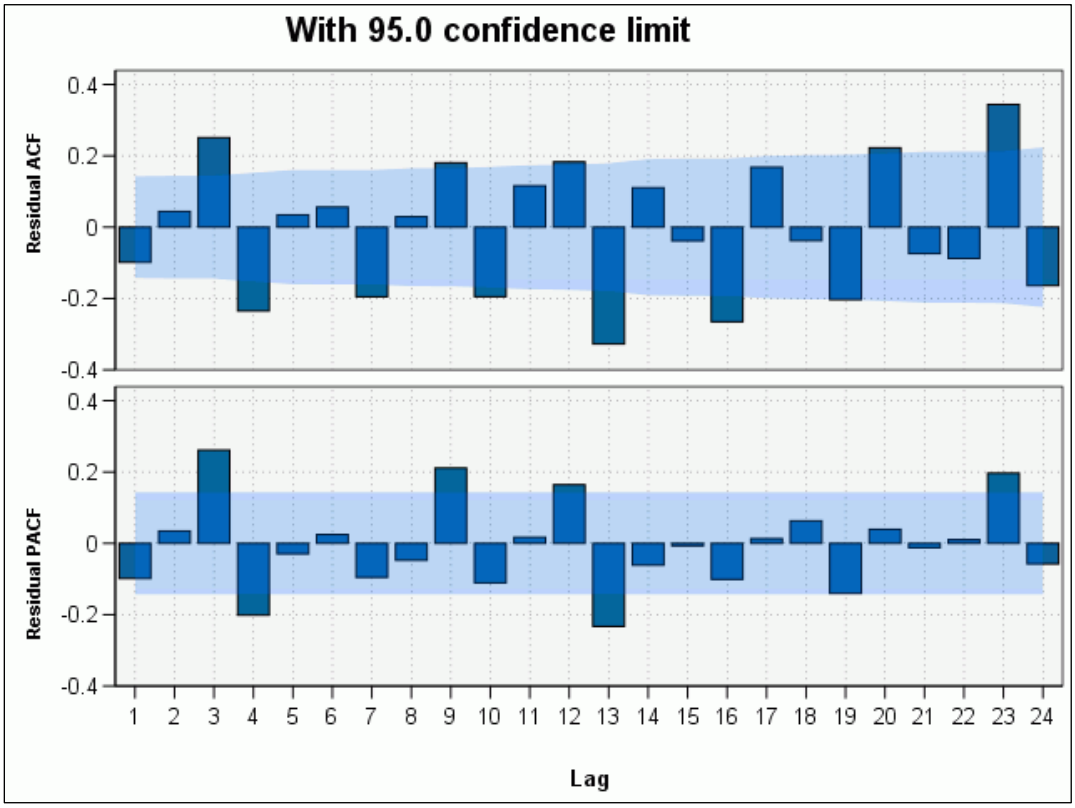
   The results appear as follows:

Model Information

Model Building Method		Exponential Smoothing Winters additive
Number of Predictors		1
Model Fit	MSE	59.936
	RMSE	7.742
	RMSPE	6.891
	MAE	6.203
	MAPE	5.345
	MAXAE	23.424
	MAXAPE	20.596
	AIC	776.606
	BIC	786.331
	R-Squared	0.943
	Stationary R-Squared	0.527
Ljung-Box Q(#)	Statistic	103.364
	df	15.0
	Significance	0.0

The Ljung-Box Q significance value of 0.0 indicates that there is significant autocorrelation in the series, which means the model was not correctly specified. A look at the ACF and PACF plots may shed some light on this problem.

3. At the left side, click on the **Correlogram** entry.
   The results appear as follows:



   Several of the bars are outside the confidence limits, which means that observations at these lags convey information for the observation at time t. These results are in agreement with the outcome of the Ljung-Box Q test that the Winters' additive exponential smoothing is not the correct model for this series.

   You will try the Winters' multiplicative model to see if it fits the series any better.

4. Close the **model nugget**.

Task 5. Create a Winters' multiplicative model.

1. Edit the **Time Series** node.
2. Click the **Build Options** tab.
3. Click the **General** item at the left, if necessary.
4. Click the **Method** dropdown, and then click **Exponential Smoothing**.
5. Under **Model Type**, select **Winters' multiplicative.**
6. Click **Run**.
   A model nugget is generated that overwrites the previous model.

Task 6. Examine the Winters' multiplicative results.

1. Edit the **model nugget**.
2. At the left side, click the **Model Information** entry.

   The results appear as follows:

Model Information

Model Building Method		Exponential Smoothing
		Winters multiplicative
Number of Predictors		1
Model Fit	MSE	66.930
	RMSE	8.181
	RMSPE	7.298
	MAE	6.625
	MAPE	5.682
	MAXAE	24.927
	MAXAPE	27.830
	AIC	797.465
	BIC	807.191
	R-Squared	0.936
	Stationary R-Squared	0.511
Ljung-Box Q(#)	Statistic	100.550
	df	15.0
	Significance	0.0

   The Normalized Bayesian Information Criterion (BIC), which can be used to compare the fit of different models is higher (807.191 vs 786.331) indicating that the Winters' additive model fit the series better than the Winters' multiplicative model. In general you should prefer the model for the dependent series that has the minimum BIC. However, the Ljung-Box Q significance value of 0.0 for both models indicates that both models are correctly specified. There, you would conclude that neither the Winters' additive or the Winters' multiplicative models are correct choices for this series.
3. Close the **model nugget**.

Task 7. Finding the best exponential smoothing model.

What is the best model for this series? Clearly, there is trial and error involved when you take the manual approach to modeling the data, even when you use diagnostics like time plots to narrow down the possible choices. If may well be that exponential smoothing is not appropriate for this series anyway, but if your goal is the find the best exponential smoothing model, you should let the Expert Modeler find it for you.

1. Edit the **Time Series** node.
2. Click the **Build Options** tab.
3. Click the **General** item at the left, if necessary.
4. Click the **Method** dropdown, and then click **Expert Modeler**.
5. In the **Model Type** area, click **Exponential smoothing models only.**
6. Click **Run**.

   A model nugget is generated that overwrites the previous model.

Task 8. Examining the results.

1. Edit the **model nugget**.
2. At the left side, click the **Model Information** entry.

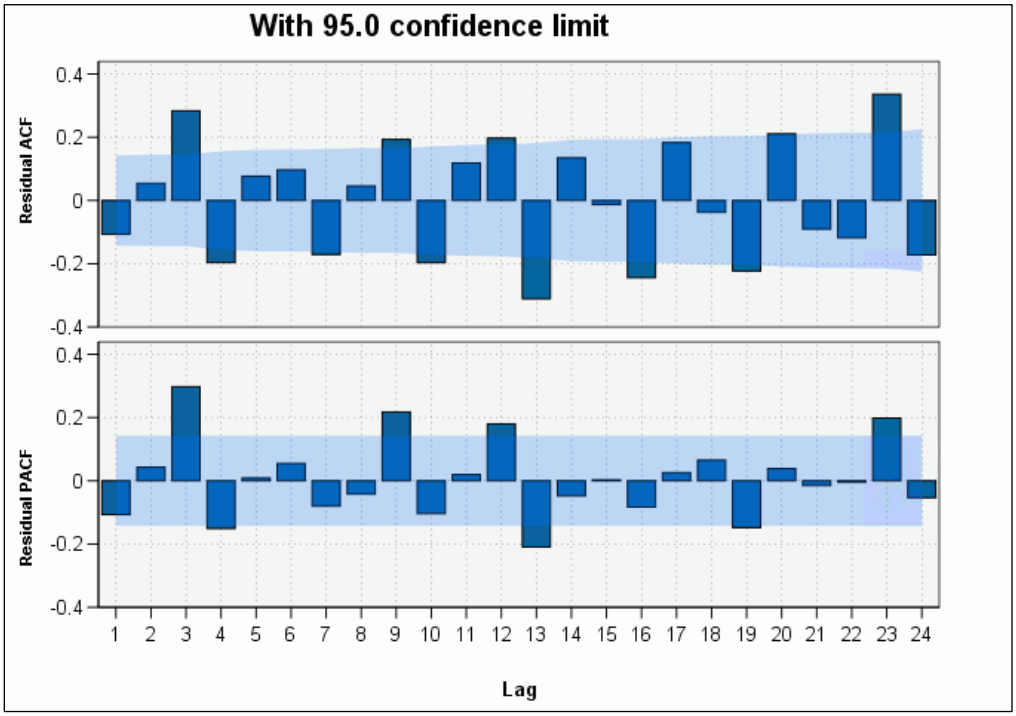
   The results appear as follows:

Model Information

Model Building Method		Exponential Smoothing
		Simple seasonal
Number of Predictors		1
Model Fit	MSE	60.870
	RMSE	7.802
	RMSPE	6.791
	MAE	6.303
	MAPE	5.331
	MAXAE	21.560
	MAXAPE	25.422
	AIC	778.542
	BIC	785.025
	R-Squared	0.941
	Stationary R-Squared	0.556
Ljung-Box Q(#)	Statistic	104.598
	df	16.0
	Significance	0.0

Although the Mean Absolute Percent Error (MAPE) is slightly better than the other two models (5.331% vs 5.345% & 5.682%), and the Normalized Bayesian Information Criterion (BIC) indicates a overall better fit (785.025 vs 786.331 & 807.191), like the other two models, the Ljung-Box Q test indicates that the model is also flawed by significant autocorrelation. This suggests that Exponential Smoothing is not the correct modeling technique for this series.

3. On the left, click the **Correlogram** entry.
   The results appear as follows:



   The ACF and PACF plots both show several instances of autocorrelations outside the confidence bands, which means these autocorrelations are significantly different from 0. Again this suggests that Exponential Smoothing may not be the correct time series technique for this series. As an exercise, rather than focusing only on Exponential Smoothing models, you may want to try letting the Expert Modeler select the best model for you. This exercise will left for you to do on your own.

4. Close the **model nugget**
   This completes the demonstration. You will create a clean state for the exercise.
5. From the **File** menu, click **Close Stream**. Click **No** when asked to save changes.
6. From the **File** menu, click **New Stream**.
   Leave IBM SPSS Modeler open for the exercise.

Results:
You have successfully determined that exponential smoothing is not the correct model for this series.

You will find the completed stream in the following folder:
C:\Training\0A028\05-Exponential_Smoothing\Solutions

IBM Training

Unit summary

- Explain types of exponential smoothing models
- Create custom exponential smoothing model
- Forecast future values with exponential smoothing
- Validate an exponential smoothing model with future data

Unit summary

IBM Training

Exercise 1

Exponential smoothing

Exercise 1: Exponential smoothing

Exercise 1: Exponential smoothing

You have a data file from a private parcel delivery company. According to the Expert Modeler, the best fitting model is ARIMA(1,1,0)(0,1,1). However, there are some people in your company who wondered why you did not use Exponential Smoothing instead. In order to answer their questions, you decide to use the Expert Modeler to find the best exponential smoothing model so you can compare it to the ARIMA model.

The results from the ARIMA model are shown below:

Model Information

Model Building Method		ARIMA
		Non-seasonal p=1,d=1,q=0; Seasonal p=0,d=1,q=1
Number of Predictors		0
Model Fit	MSE	54.941
	RMSE	7.412
	RMSPE	1.000
	MAE	5.681
	MAPE	0.760
	MAXAE	32.025
	MAXAPE	3.983
	AIC	622.956
	BIC	629.043
	R-Squared	0.994
	Stationary R-Squared	0.317
Ljung-Box Q(#)	Statistic	14.214
	df	16.0
	Significance	0.6

Stream file: **unit_5_exercise_1_start.str**

Folder: **C:\Training\0A028\05-Exponential_Smoothing\Start**

Task 1. Open the stream.

- From the **File** menu, click **Open Stream**.
- Navigate to the **C:\Training\0A028\05-Exponential_Smoothing\Start** folder, and then double-click **unit_5_exercise_1_start.str**.

Task 2. Examine a time plot of the series.

- From the **Graphs** palette, select the **Time Plot**, and add it downstream from the **Type** node.
- Create a time plot that displays **parcel** on the chart. Be sure to display the dates on the X axis and to not normalize the graph.

  How would you describe the parcel series? Is there a trend? Is there seasonality? If there is seasonality, does it look like additive or multiplicative seasonality?

Task 3. Find the best exponential smoothing model.

- Add a **Time Series** node downstream from the **Type** node.
- Edit the **Time Series** node.
- Click the **Fields** tab, if necessary.
- Move **parcel** into the **Targets** box.
- Click the **Data Specification** tab.
- For **Date/time** field, select **Date_**.
- Beside **Time intervals**, select **Days**.
- Click the **Build Options** tab.
- Click the **General** item at the left, if necessary.
- Click the **Method** dropdown, and then click **Expert Modeler**.
- Under **Model Type**, select **Exponential smoothing models only.**
- Click **Run**.

Task 4. Examine the results.

- Edit the **model nugget**.
- From the left, select **Model Information**.

  Which model fits the best? Why? Which model would you prefer?
- Close the **model nugget**.
- Exit **IBM SPSS Modeler** without saving anything.

Exercise 1: Tasks and results

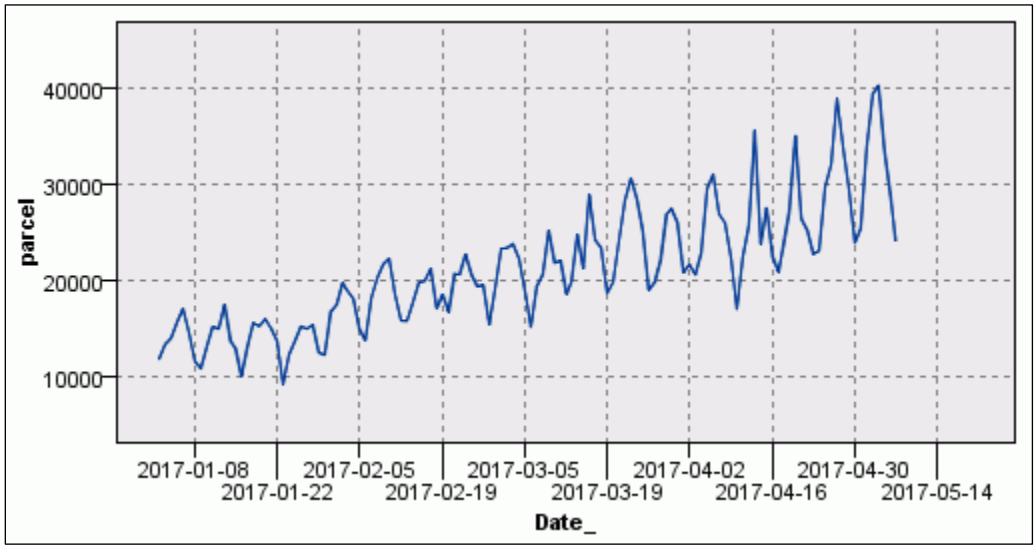
Task 1. Open the stream.

- From the **File** menu, click **Open Stream**.
- Navigate to the **C:\Training\0A028\05-Exponential_Smoothing\Start** folder, and then double-click **unit_5_exercise_1_start.str**.

Task 2. Examine a time plot of the series.

- From the **Graphs** palette, select the **Time Plot**, and add it downstream from the **Type** node.
- Edit the **Time Plot** node.
- Besides **Series**, select **parcel**.
- Besides **X axis label**, enable the **Custom** option, and then select **Date_**.
- Clear the **Normalize** option.
- Click **Run**

  The results appear as follows:



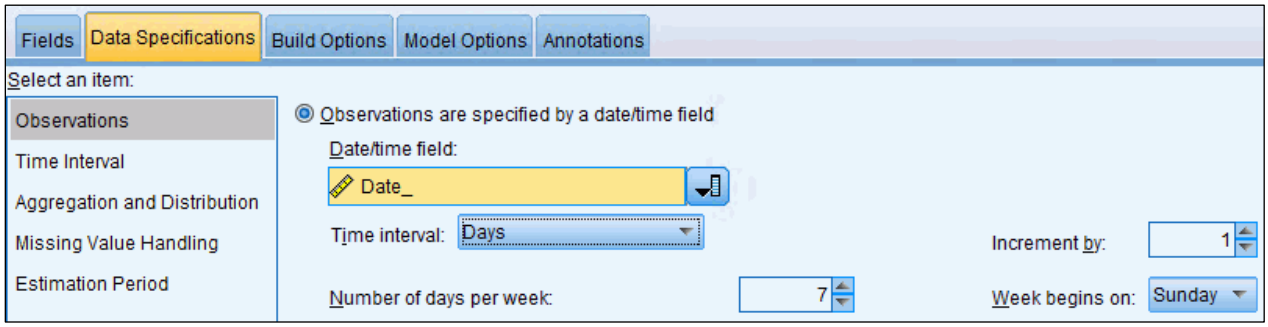
  There is definitely both trend and seasonality in the series. The magnitude of the bumps seem to become more pronounced toward the end of the series so in all likelihood there is multiplicative seasonality.

Task 3. Find the best exponential smoothing model.

- From the **Modeling** palette, add a **Time Series** node downstream from the **Type** node.
- Edit the **Time Series** node.
- Click the **Fields** tab, if necessary.
- Enable the **Use custom field assignments** option.
- Move **parcel** into the **Targets** box.
- Click the **Data Specification** tab.
- Click the **Observations** item on the left, if necessary.

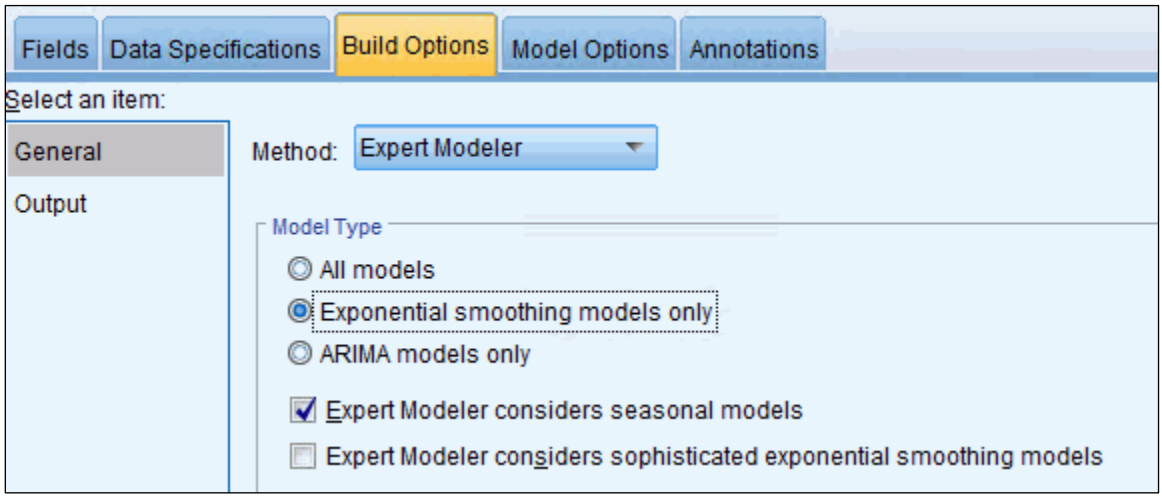
  The observations are defined by the field Date_ and represent days.
- Ensure that the **Observations are specified by a date/time field** option is enabled.
- For **Date/time** field, select **Date_**.
- Beside **Time intervals**, select **Days**.

  The results appear as follows:



- Click the **Build Options** tab.
- Click the **General** item at the left, if necessary.
- Click the **Method** dropdown, and then click **Expert Modeler**.
- Under **Model Type**, select **Exponential smoothing models only.**

  The results appear as follows:



- Click **Run**.

Task 4. Examine the results.

- Edit the **model nugget**.
- From the left, select **Model Information**.

  The results appear as follows:

Model Information

Model Building Method		Exponential Smoothing
		Winters multiplicative
Number of Predictors		1
Model Fit	MSE	3,827,515.520
	RMSE	1,956.404
	RMSPE	8.872
	MAE	1,547.282
	MAPE	7.347
	MAXAE	6,547.171
	MAXAPE	26.959
	AIC	1,912.837
	BIC	1,921.346
	R-Squared	0.908
	Stationary R-Squared	0.470
Ljung-Box Q(#)	Statistic	25.874
	df	15.0
	Significance	0.0

  Based on the results, the Ljung-Box Q test is significant for the exponential smoothing model but not for ARIMA. Therefore, you would conclude that the exponential smoothing model is misspecified and that you will prefer ARIMA.

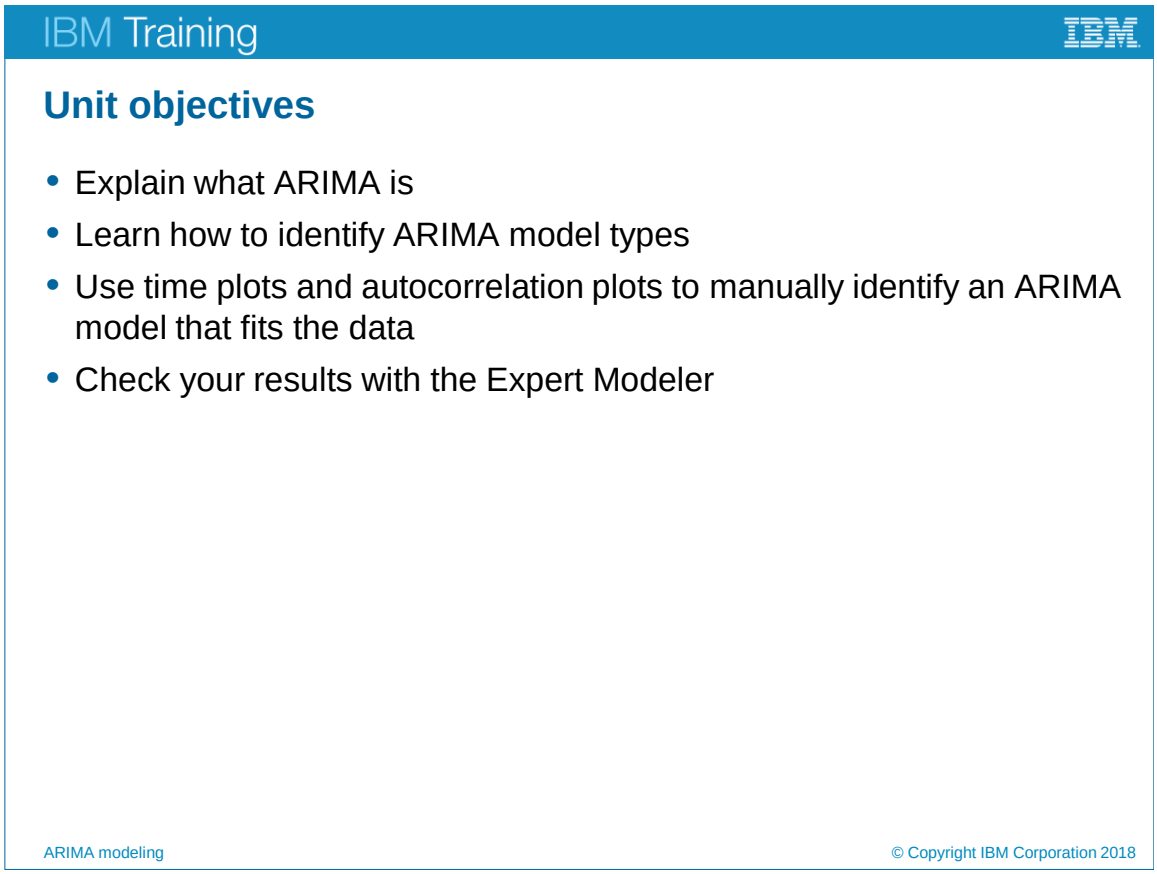
- Close the **model nugget**.
- Exit **Modeler** without saving anything.

You will find the completed stream in the following folder:
C:\Training\0A028\05-Exponential_Smoothing\Solutions

IBM Training

ARIMA modeling

IBM SPSS Modeler (v18.1.1)



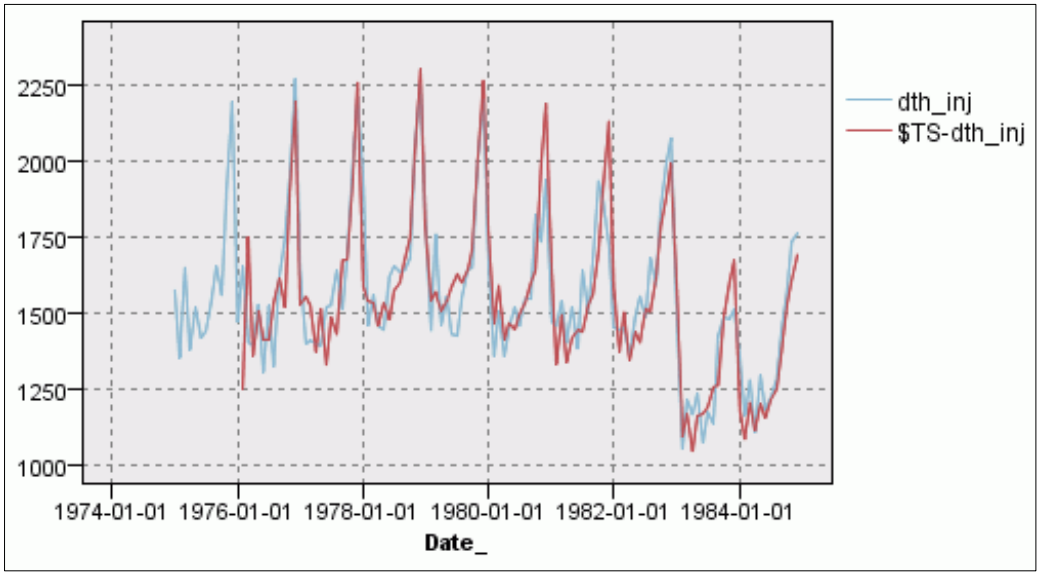
Unit objectives

Before reviewing this unit, you should be familiar with the following topics:

- Working with IBM SPSS Modeler (streams, nodes, palettes)
- Importing data (Var. File node)
- Defining measurement levels, roles, blanks, and instantiating data (Type node)
- Examining the data (Table node, Time Plot node)
- Evaluating time series models, using time plots, autocorrelation plots, and time series model fit statistics
- Using the model nugget to score data

IBM Training

ARIMA modeling



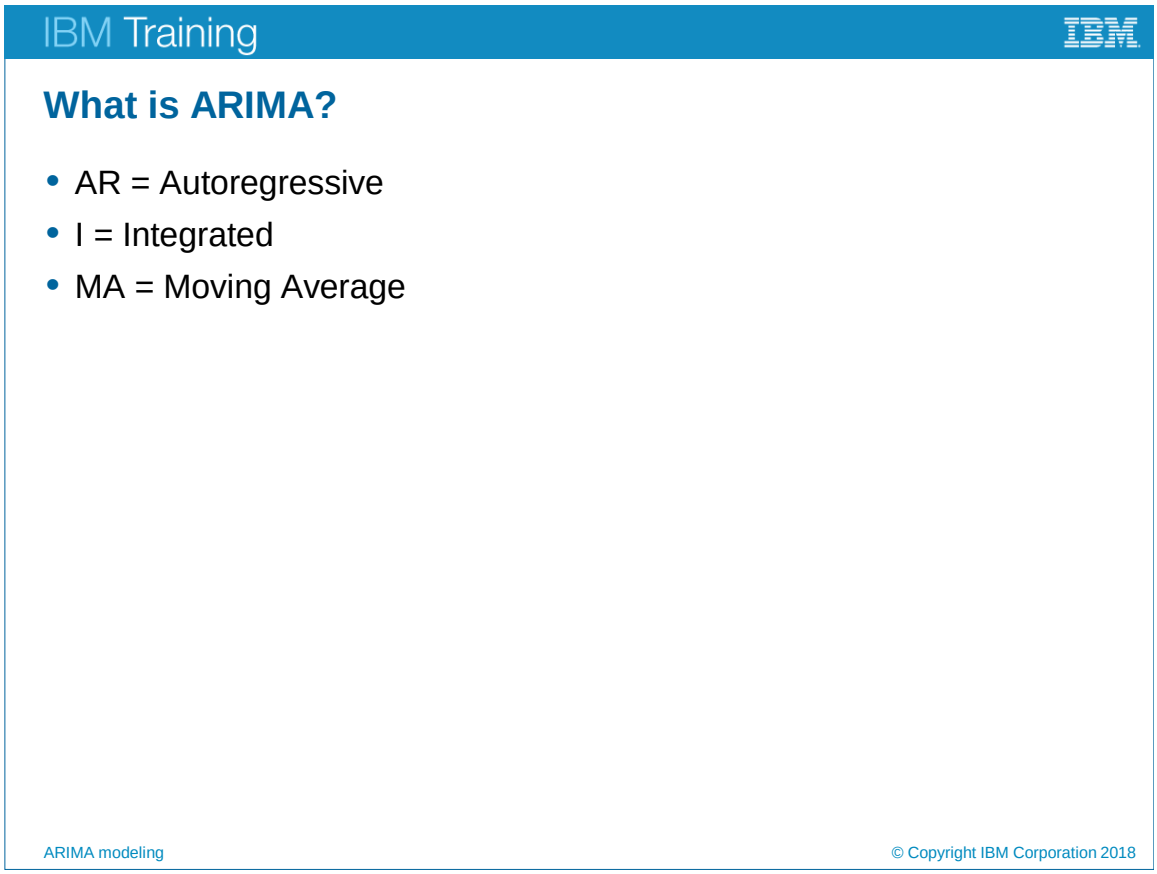
ARIMA modeling

ARIMA modeling

In this unit, you will consider time series models in which future values of the series are predicted from previous series values at specified lags. Such models are called autoregressive (regression of lags of a time series field on the field itself).

Since autoregressive models represent one component of the more general ARIMA model, this unit serves as an introduction to ARIMA modeling. We will apply autoregressive models to a series containing unemployment data. First you will try to create an ARIMA model from information gained in exploratory analysis, including autocorrelation plots. Then you will let the Expert Modeler create an ARIMA model and compare the two.

In this unit, you will be introduced to ARIMA models, learn the process used to tentatively identify an ARIMA model (assuming you, the analyst, are doing it, not the Expert Modeler), and how to build custom ARIMA models. It is important to realize that this topic is only a starting point for constructing ARIMA models and does not provide all the information and exposure you would need to master the art of identifying, estimating, and diagnosing ARIMA models, which can be a complicated subject. However, even if you intend to have the Expert Modeler fit the time series models, you should be familiar with the components of ARIMA models in order to understand the model created by the Expert Modeler and to be able to evaluate whether it makes sense in light of what you know of the series.

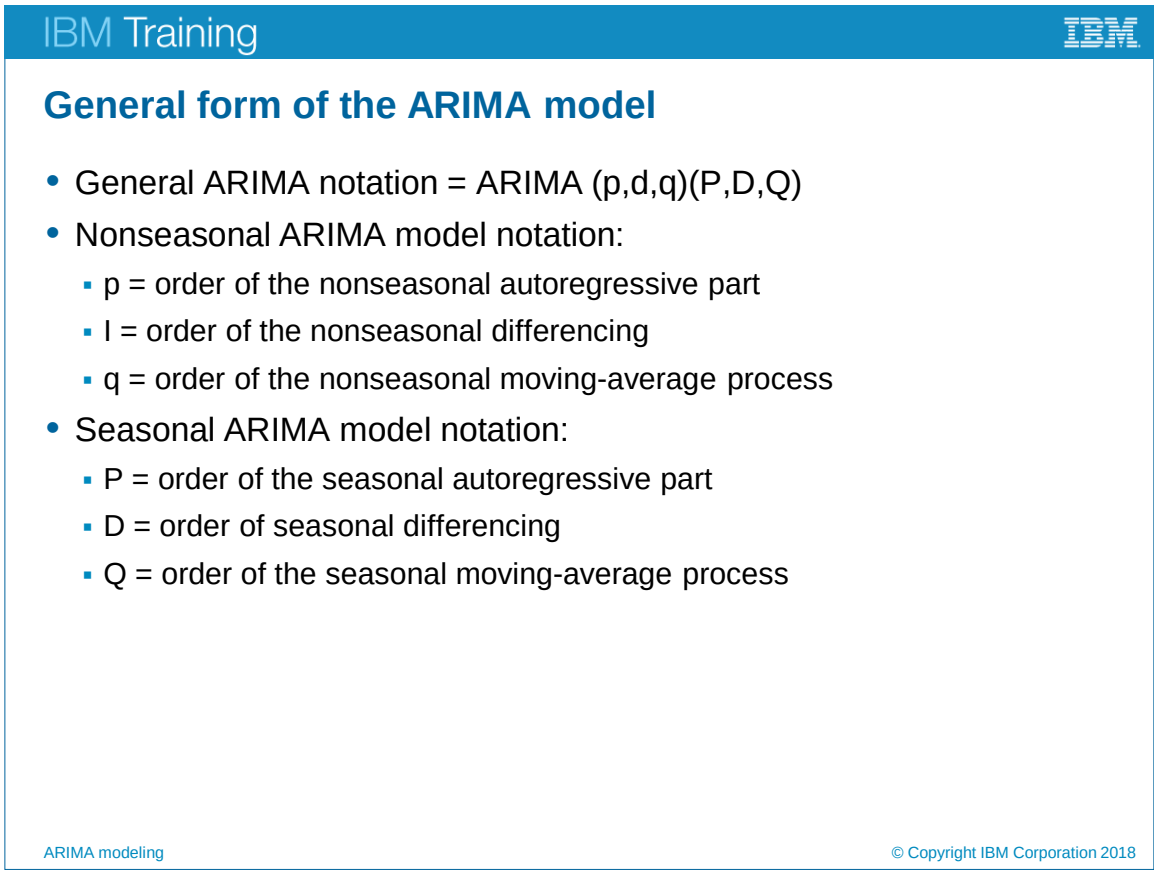


What is ARIMA?

ARIMA is a rich modeling technique, so we begin by outlining the different components contained in ARIMA models. George Box and Gwilym Jenkins developed many of the ideas incorporated into ARIMA models in the 1970s, and for this reason ARIMA modeling is sometimes called “Box-Jenkins” modeling. ARIMA stands for AutoRegressive Integrated Moving Average, and the assumption of this approach is that the common variation in the series to be forecast can be divided into three components:

- Autoregressive (AR)
- Integrated (I) or Difference
- Moving Average (MA)

An ARIMA model can have one, two, or all three of these components. In addition, these components can operate at both the seasonal and nonseasonal level. For example, sales this month may be related to sales last month (nonseasonal) as well as sales one year ago (seasonal).



General form of the ARIMA model

There are many different types of ARIMA models but the general form of the model is ARIMA(p,d,q)(P,D,Q) where:

- p refers to the order of the nonseasonal autoregressive process incorporated into the ARIMA model (and P is the order of the seasonal autoregressive process).
- d refers to the order of integration or differencing (and D is the order of the seasonal integration or differencing).
- q refers to the order of the moving average process incorporated in the model (and Q is the order of the seasonal moving average process).

So for example an ARIMA(2,1,1) would be a nonseasonal ARIMA model where the order of the autoregressive component is 2, the order of integration or differencing is 1, and the order of the moving average component is also 1. Not all ARIMA models have all three components. For example, an ARIMA(1,0,0) has an autoregressive component of order 1 but no difference or moving average component. Similarly, an ARIMA(0,0,2) will have just two moving average components of order 1 and 2.

Normally, a model will include lower order components along with the highest order term. Thus, in the ARIMA(0,0,2), moving average terms of order 1 and 2 will be included. But models can instead include only higher order components. Thus an ARIMA(0,0,3) model might include moving average terms of order 2 and 3, but not 1.

Autoregressive

In regression analysis we assume that one field influences another field, for example, advertising spending affects sales of a product. Alternatively in regression it might also be the case that lagged values of one field are a good predictor of another field, for example, that the amount of advertising two periods ago is a good indicator of what sales are now.

In an analogous manner to regression, ARIMA models use lagged values to predict the dependent field. The term autoregressive implies that a field based on the actual dependent field is used in some way to predict the dependent field. More specifically, the autoregressive component of an ARIMA model uses the lagged values of the dependent field as predictors of the current value of the dependent field. For example, a good predictor of sales could be the number of sales in the previous time period. The lagged values of the dependent field account for part of the model's fit.

There can be different orders of autoregression, and the order of autoregression refers to the time difference between the dependent field and the lagged dependent field, which is being used as a predictor. If the dependent field is influenced by the dependent field lagged one time period, then this is an autoregressive model of order one and is sometimes called an AR(1) process. The AR(1) component of the ARIMA model is saying that the value of the dependent field in the previous period (t–1) is a good indicator and predictor of what the dependent field will be now (t). If on the other hand, a good predictor of a dependent field is the dependent field two periods previous, then the autoregressive process is of order two, an AR(2) process. This pattern continues for higher order processes.

Integrated

The "I" part of the ARIMA model refers to Integrated. This term relates to whether a time series requires differencing in order to become stationary. The idea of stationarity is very important in ARIMA modeling. The dependent field in an ARIMA model should be stationary to meet the assumptions of this technique.

Stationarity

In time series analysis the term stationarity is often used to describe how a particular time series field changes over time. Stationarity has three components. First, the series has a constant mean, which implies that there is no tendency for the mean of the series to increase or decrease over time. Second, the variance of the series is assumed constant over time. So, for example, if the magnitude of the seasonal swings increase over time, then a series is not stationary. Finally, any autocorrelation pattern is assumed constant throughout the series. For example, if there is an AR(2) pattern in the series, it is assumed to be present throughout the entire series.

Any violation of stationarity creates estimation problems for ARIMA models. It is difficult to detect the true variations in the dependent field if it is non-stationary. Because the mean of the series is changing over time, correlations and relationships between the fields in the ARIMA model will be exaggerated or distorted. Only if the mean of the dependent field is stationary will true relationships and correlations be identified.

The Integration component of ARIMA is typically associated with removing trend from the series, which would violate the constant mean component of stationarity. It is often the case in time series analysis that the mean of a field increases or decreases over time. For example, if your aim is to study how car ownership has changed over time, we know that ownership has increased greatly. Therefore, the mean of the series will also have increased over time. If the mean of the series changes over time then the field is non-stationary.

In order to make a series containing trend stationary, you can create a new series that is the difference of the original series. We have already encountered this concept in a previous topic where you predicted the number of visitors to a brewery using linear regression.

A first order difference creates a value for the new series which is the difference between the series value in the current period minus the series value in the previous period. The idea of differencing the series is shown below. For example, the first value of the differenced series is calculated by taking the value of the series in the current period (9) minus the value of the series in the previous period (7).

Original Series	Differenced Series
7	--
9	2
12	3
14	2
15	1
14	-1
9	-5

Often the differenced series will have a stationary mean. If the differenced series does not have a stationary mean then it might be necessary to take first differences of the differenced series. This transformation is known as second order differencing as the original series has now been differenced twice. This idea is shown in the following table.

Original Series	First Order Differenced Series	Second Order Differenced Series
7	--	--
9	2	--
12	3	1
14	2	-1
15	1	-1
14	-1	-2
9	-5	-4

The number of times a series needs to be differenced is known as the order of integration. Differencing can be performed at the seasonal (current time period value minus the value from one season ago) or non-seasonal (current time period minus the value from the previous time period).

In short, differencing can remove trend from a series in order to create the stationary series that forms the basis of ARIMA, and integration later builds the trend back into the series when predictions (forecasts) are produced.

- If a series is stationary then there is no need to difference the series and the order of integration is zero. In all ARIMA models the dependent field should be left in its original values. ARIMA models will be of the form ARIMA (p,0,q) where p is the order of the autoregressive process and q is the order of the moving average process in the model.
- If a series is non-stationary then usually first differencing the series will make it stationary. If first differencing makes the series stationary then the order of integration is one. ARIMA models will be in the form of ARIMA (p,1,q).
- Very occasionally it might be necessary to difference a series twice to make it stationary in which case the order of integration is two. It is however nearly always the case that first differences will make a non-stationary series stationary.

In IBM SPSS Modeler, the dependent field will automatically be transformed if necessary if you use the Expert Modeler and an ARIMA model is selected. If you wish to check on whether differencing will be necessary, you can specify a particular order of differencing within time series charts (time plots, autocorrelation plots). Also the Time Series Model node and the time series charts permit you to request log transforms (often used to stabilize the variance of a series). Thus you may not need to make your own transformations outside the Time Series node before using a field in a model.

Note

The term “integration” comes from calculus, where it is the opposite operation of “differentiation.” In ARIMA it is the opposite of differencing. If requested by the user, ARIMA will difference the dependent field before estimating the rest of the model parameters. Then, when creating fitted values, it integrates them so that they actually fit the original, rather than the differenced, dependent field. Thus, it is best to do differencing and other transformations in the Time Series Model node.

Moving Average

The autoregressive component of an ARIMA model uses lagged values of the dependent field as independent fields. In contrast to this, the moving average component of the model uses lagged values of the model error as independent fields. It can be successful at extracting autocorrelated patterns in the series that would have otherwise been included in the model error.

The order of moving average refers to the lag length between the error and the dependent field. For example, if the dependent field is influenced by the model’s error lagged one period then this is a moving average process of order one and is sometimes called an MA(1) process. The MA(1) component of the ARIMA model is saying that the model’s error in the previous period is related to what the dependent field will be now.

Some analysts interpret moving average components as outside events or shocks to the system. That is, an unpredicted change in the environment occurs now, which influences the current value in the series as well as future values. Thus the error component for the current time period relates to series’ values in the future.

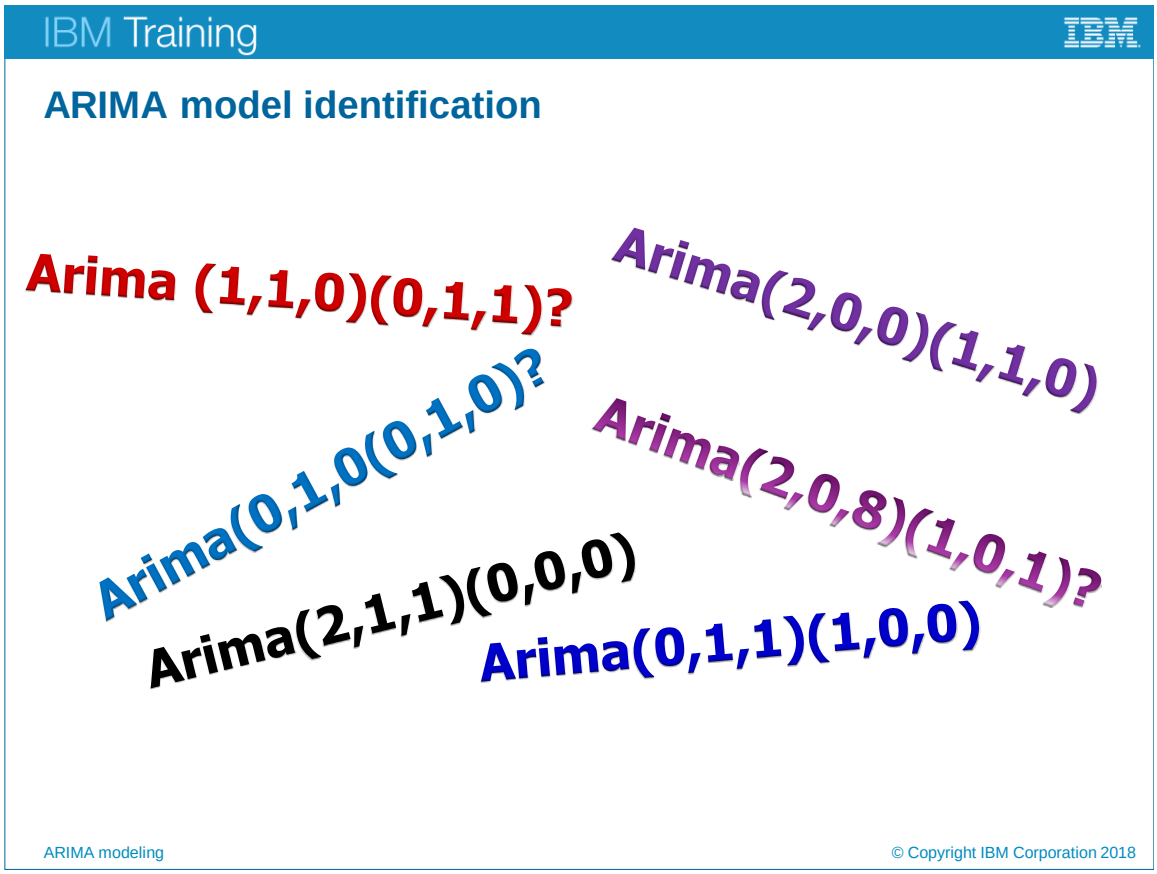
Consider a time series model with a tendency for a positive error to be followed two periods later by an increase in the series value. A moving average of order two would use the positive error two periods previous in order to better predict this pattern. Techniques such as regression cannot extract this variation as part of its model fit and as a result the model error in regression would be autocorrelated.

The Basic ARIMA Equation

Let Y* be the dependent series transformed to stationarity. Then the general form of the model is:

$$Y^*_t = \phi_1 Y^*_{t-1} + \phi_2 Y^*_{t-2} + \ldots + \phi_p Y^*_{t-p} + \varepsilon_t - \theta_1 * \varepsilon_{t-1} - \theta_2 * \varepsilon_{t-2} - \ldots - \theta_q * \varepsilon_{t-q}$$

That is, Y* is predicted by its own past values along with current and past errors. The challenge with any dependent series is identifying the order of differencing and seasonal differencing, the order of autoregressive parameters, both nonseasonal and seasonal, and the order of moving average parameters, both nonseasonal and seasonal.



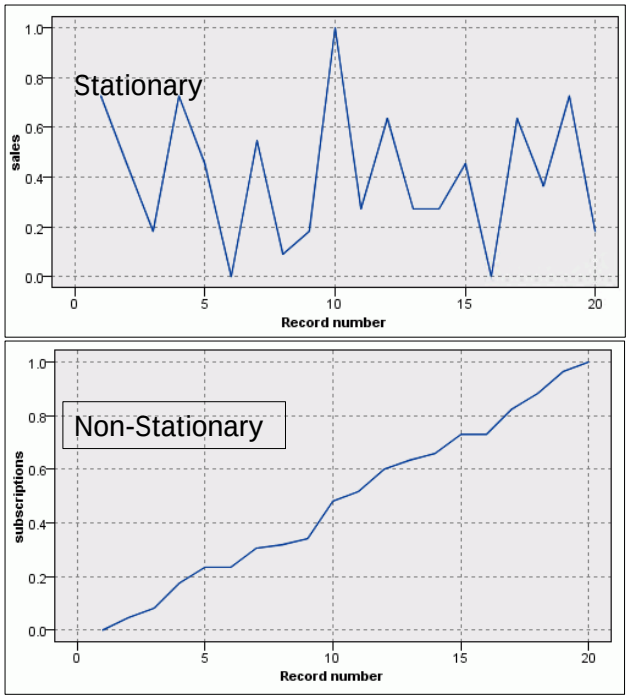
ARIMA model identification

There are many different ARIMA models which could be fit to a particular time series of interest. The type of ARIMA model depends upon the selected orders of autoregression (p), integration (d), and moving average (q).

As with all times series analysis, you wish to find an ARIMA model that fits the historic data the closest and will perform the best when used for forecasting. In order to do this, it is necessary to select the optimal combination of p, d, and q. In other words, it is important to select the “right” combination of autoregressive, integration, and moving average orders. Unfortunately, it is often the case that identifying the p, d, and q combination to give the “best-fit or forecasting” ARIMA model is a process of trial and error. In many ways the identification stage is by far the most subjective in the entire ARIMA modeling process. As we noted above, the Expert Modeler will, if you wish, do the identification for you.

IBM Training

Identifying the order of integration (d) with time plots



ARIMA modeling

*Identifying **the order of integration (d) with time plots***

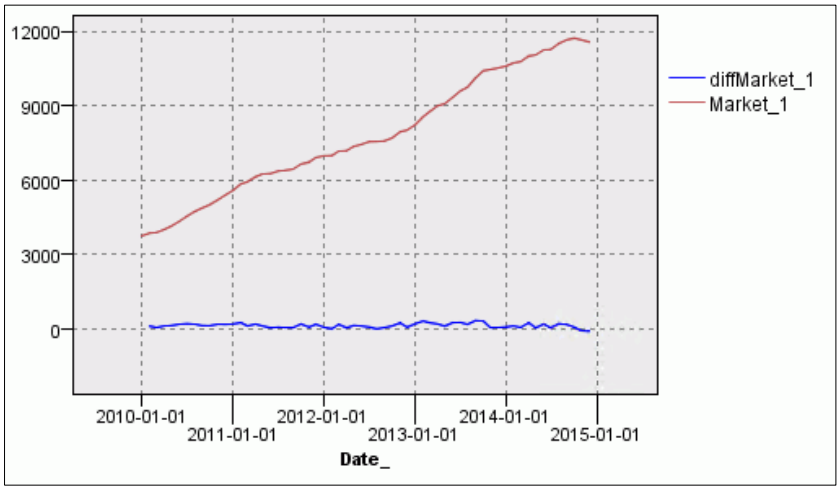
The identification stage involves using exploratory techniques (time plots and autocorrelation and partial autocorrelation plots) in order to determine the most likely combination of p, d, and q that will give the closest fit to the historic data.

- The order of integration (d) can usually be identified by looking at time plots for the dependent field (or the dependent field after differencing).
- Autocorrelation and partial autocorrelation plots of the dependent field are used to suggest plausible values for p and q, the orders of autoregression and moving average.

Remember that in ARIMA modeling it is important for the dependent field to be stationary. The general procedure for correctly identifying the order of integration is to begin by reviewing a time plot of the series to see whether the dependent field is stationary. Recall that a series is stationary when there is no tendency for the mean of a field to increase or decrease over time, the variance is homogeneous, and the autocorrelation pattern is constant throughout the series (although it is difficult to assess this last requirement). In this slide, the series in the top graph is stationary because it rises and falls around a constant mean. In contrast, the bottom graph is non-stationary because the mean changes as the series progresses to the right.

IBM Training

Differencing illustrated



ARIMA modeling

Differencing illustrated

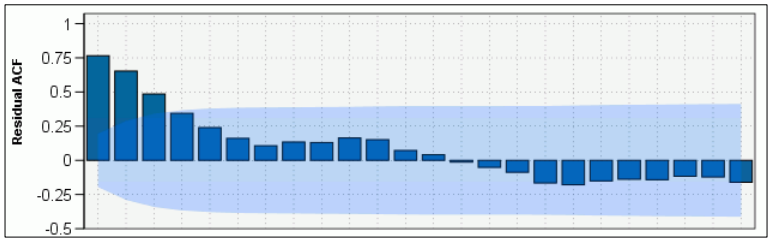
This slide illustrates what a series looks like before and after it is differenced. The original series, Market_1, is not stationary because it trends upward and the overall mean changes over time. In contrast, the differenced version of the series, diffMarket_1, is stationary because it has no trend and the mean remains constant for the length of the series.

Here you focus on problems with stationary violations due to mean shifts.

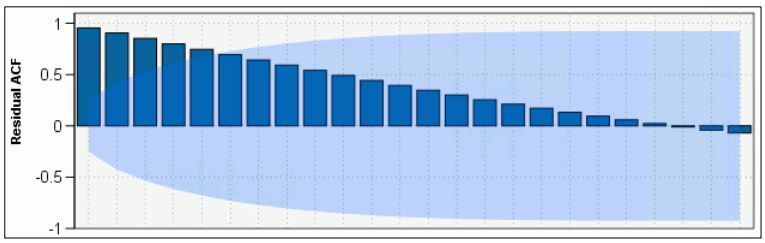
- If the series is stationary then the order of integration (d) is zero and there is no need to difference the series. The type of ARIMA model will be an ARIMA(p,0,q).
- Alternatively if the series is non-stationary due to trend, then the next stage is to run a time plot of the first differenced values of the dependent field. Usually, first differencing the series will make it stationary. If the time plot of the first differenced field shows that the first differences are stationary, then the order of integration is one. The type of ARIMA model will be an ARIMA(p,1,q).
- Finally, occasionally it will be necessary to second difference the series in order to make it stationary. If this is the case the type of ARIMA model is an ARIMA(p,2,q).

IBM Training

Identifying the order of integration (d) with ACF Plots



Stationary



Non-Stationary

ARIMA modeling

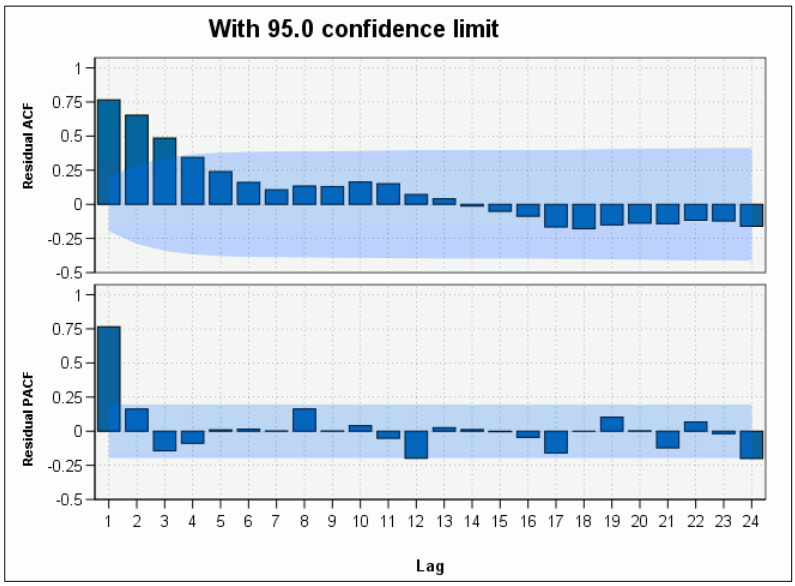
Identifying the order of integration (d) with ACF plots

While usually you can tell in a time plot if a series is stationary or not, at other times it is not so obvious. If that is the case, you should try using an autocorrelation plot (ACF) to figure it out. In an ACF plot, the series is stationary if the spikes decline rapidly or in an exponential fashion. Notice that in the top ACF plot of the United States savings rate from 1955 to 1979, the spikes seem to decline rapidly. This suggests that the series is stationary. In contrast, the spikes in the bottom ACF plot decline very slowly. This is the type of pattern you see when a series is non-Stationary.

Of course, in practice, it is often not this easy to determine whether or not a series is stationary, but usually you can figure it out by using the time plots, ACF plots, or both. If you are still in doubt, the Expert Modeler will determine for you whether or not the series is stationary.

IBM Training

Identifying order of AR (p) & MA (q) terms



ARIMA modeling

Identifying order of AR (p) & MA (q) terms

The exploratory process for identifying the orders of autoregression and moving average is the subjective part of model identification. Identification of possible autoregressive (p) and moving average (q) orders requires examination of the autocorrelation (ACF) and partial autocorrelation (PACF) functions for the dependent field. There are some theoretical guidelines of how autocorrelation and partial autocorrelation functions behave for different orders of autoregressive and moving average processes. The plots shown in the slide above are examples of ACF and PACF plots.

General guidelines for identifying AR and MA orders

Here are some general guidelines for identifying the number of AR and MA terms that will be needed for your ARIMA model:

- Nonstationary series have an ACF characterized by significant bars (spikes) on the low lags that remains significant for half a dozen or more lags, rather than rapidly declining to zero. You must difference such a series until it is stationary before you can identify the process.

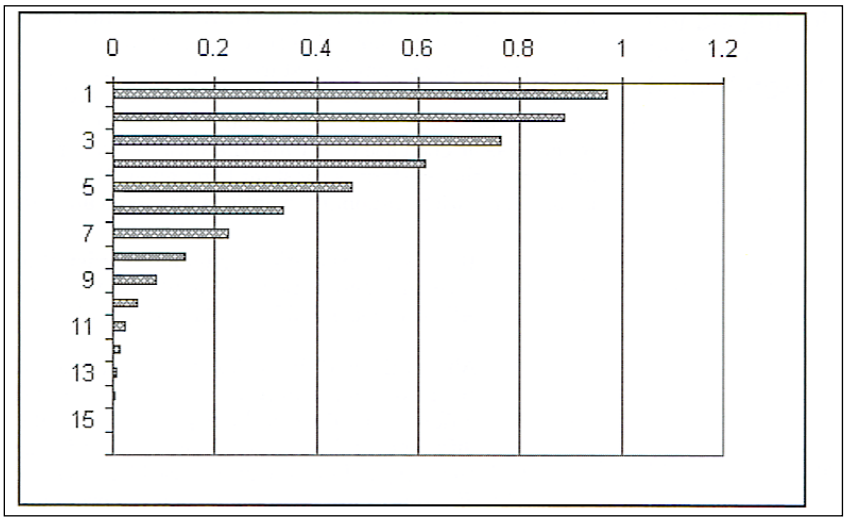
- Autoregressive processes have an exponentially declining ACF and significant spikes in the first one or more lags of the PACF. The number of significant spikes in the PACF plot indicates the number of AR terms that need to be included in the model.
- Moving average processes have spikes in the first one or more lags of the ACF and an exponentially declining spikes in the PACF. The number of spikes indicates the number of MA terms that are needed in the model.

Identifying an Autoregressive Process

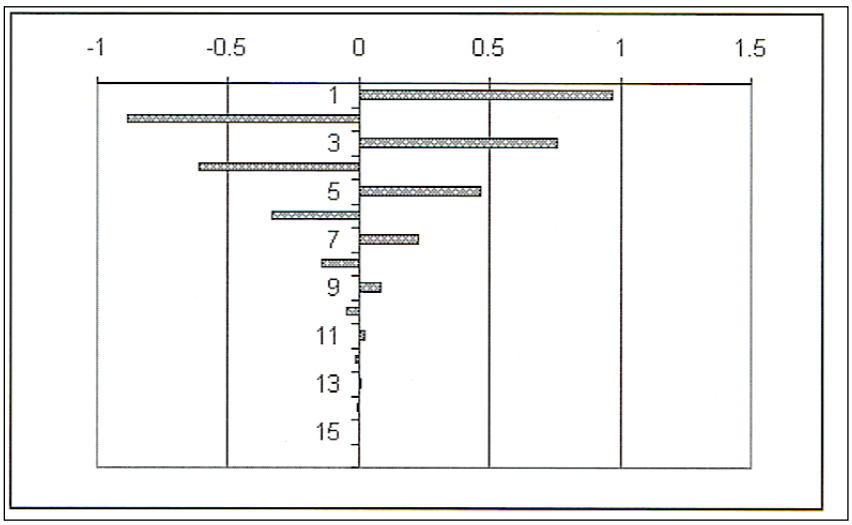
ARIMA models which have an autoregressive component but no moving average component, for example, ARIMA(p,d,0) models, have exponentially or sine wave declining autocorrelation functions. The first graphic shows an exponentially declining autocorrelation function while the second graphic shows a declining sine wave pattern in the autocorrelation function.

The plots below assume any necessary differencing has already been performed.

ACF Plot for an Autoregressive ARIMA (p,d,0) Process:



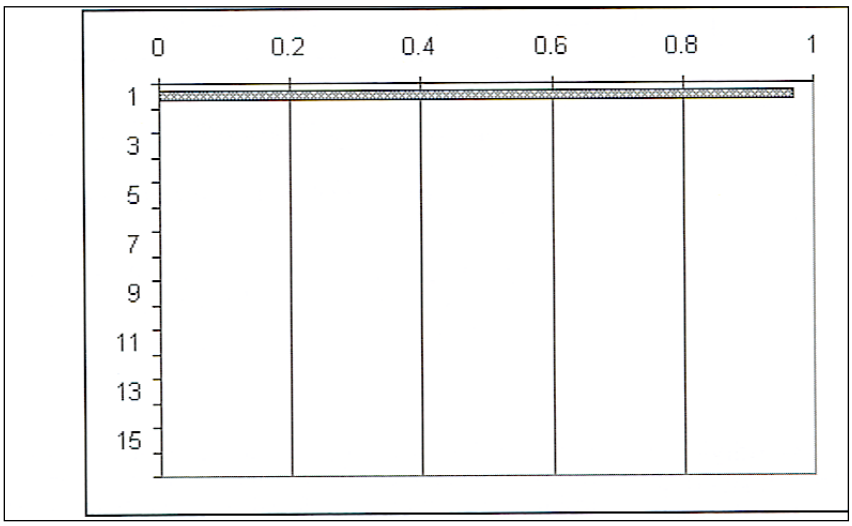
ACF Plot for an Autoregressive ARIMA (p,d,0) Process:



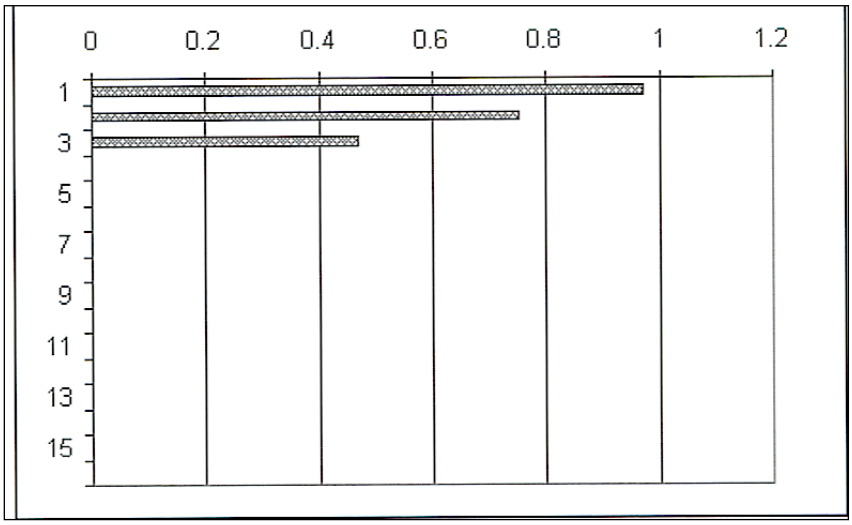
A pure autoregressive process with no moving average component is also characterized by significant bars (spikes) on the low lags of the partial autocorrelation functions followed by a sudden stop in the significance of the bars for all subsequent lags. The number of significant bars (spikes) indicates the order of the autoregressive process.

- If the ARIMA model followed a first order autoregressive process—an ARIMA(1,d,0)—the partial autocorrelation function will have one spike followed by a sudden decline to zero for all subsequent lagged bars. This is shown in exaggerated form in the first graphic.
- Similarly, if the ARIMA model followed a third order autoregressive process, an ARIMA(3,d,0), the partial autocorrelation function will have three spikes followed by a sudden decline to zero for all subsequent lagged bars. This is shown in the second graphic.

PACF Plot for an Autoregressive Model of Order p = 1



PACF Plot for an Autoregressive Model of Order p =3



To summarize, a pure autoregressive process is characterized by

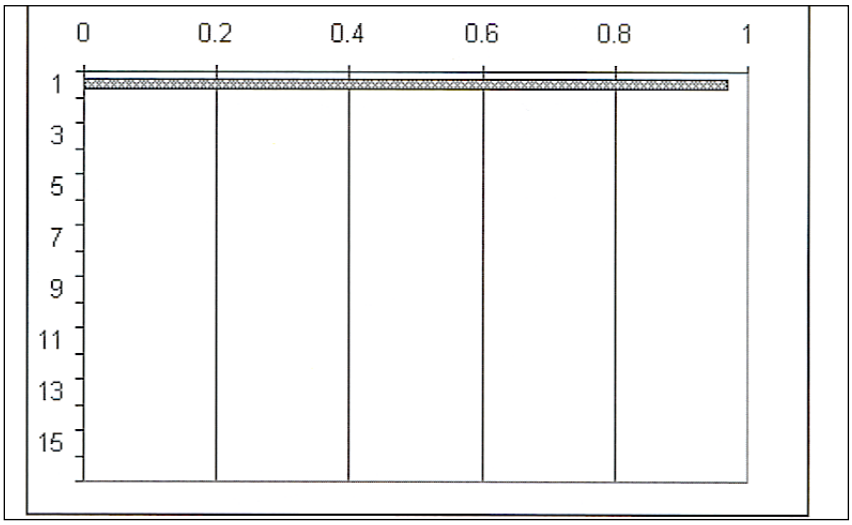
- An exponentially or sine wave declining autocorrelation function
- A number of spikes on the partial autocorrelation function equal to the order of the autoregressive process

Identifying a Moving Average Process

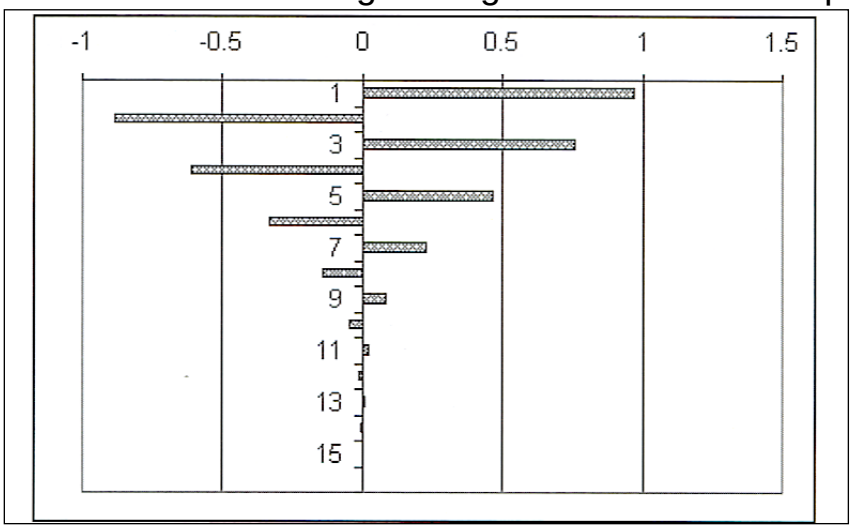
For an ARIMA model that has a moving average component only, an ARIMA(0,d,q), there are also well-defined patterns in the autocorrelation and partial autocorrelation functions. The patterns described below assume that differencing, if needed, has already been performed.

As just shown, for an autoregressive model there is an exponentially or sine wave declining autocorrelation plot and spikes in the partial autocorrelation function. In contrast to this, a moving average process has the spikes in the autocorrelation function and the declining exponential or sine wave in the partial autocorrelation function. This is the opposite of what was true for the autoregressive process. The following two graphics show the respective plots for a moving average process of order one, for example, an ARIMA(0,d,1). In the autocorrelation plot, the number of spikes indicates the order of the moving average process. In this case, there is one spike for the autocorrelation function along with an exponentially declining or sine wave declining partial autocorrelation plot.

ACF Plot for a Moving Average Process of Order q = 1



PACF Plot for a Moving Average Process of Order q=1



To summarize, a pure moving average process is characterized by

- A number of spikes on the autocorrelation function equal to the order of the moving average process
- An exponentially or sine wave declining

Identifying an ARIMA Model in Practice

These guidelines can help in many instances to identify the orders of autoregression and moving average which best characterize the series. Very often, however, this part of the identification process is more of an art than a scientific process. The art is to learn how to pick out the relevant information from autocorrelation and partial autocorrelation plots. Plots are almost always less clear-cut than those shown in the theoretical guidelines, particularly when the best model has a combination of autoregressive and moving average components.

If the dependent field is characterized by both autoregressive and moving average components, then the patterns in the autocorrelation and partial autocorrelation functions are more complex, although patterns for low order combinations of autocorrelation and moving average components are found in most time series books. The book by Box, Jenkins, and Reinsel (1994) is a comprehensive reference concerning this issue.

Identifying the Best Model

After a model has been estimated, you then need to decide whether that particular ARIMA(p,d,q) model is acceptable. Normally, when you do custom ARIMA modeling, several models are fit, and you will need to decide between them. There are many potential tests you can use:

- Parameter estimates. After estimation, all parameter estimates should be statistically significant. You might remove non-significant parameter estimates from the model and estimate a simpler model. You can tentatively add parameters to the model, but the added parameter estimates should be statistically significant.
- Residual ACF and PACF. The residual ACF and PACF correlations should be small and nonsignificant. By chance, you will occasionally observe significant autocorrelations associated with a random series, but you should pay special attention to significant autocorrelations at the first few lags as well as seasonal lags.
- Ljung-Box Q statistic. The Ljung-Box Q statistic is a statistic associated with residual autocorrelations up to and including a given lag. IBM SPSS Modeler fixes the number of lags to be tested to 18. The Ljung-Box Q statistic tests the null hypothesis that the autocorrelations from lag 1 to the given lag are collectively associated with a white noise process.
- Model fit statistics. Fit statistics such as the mean absolute error are measured in the same metric as the dependent series. Small values of these statistics are associated with better-fitting models. But, take care that such better fit is not attained through adding unnecessary parameters to the model.
- Information criteria. There are a number of information criteria with names such as the Akaike Information Criterion (AIC), and the Bayesian Information Criterion (BIC), and the like. These criteria combine the sum of squared residuals and the number of parameter estimates in various fashions to produce a criterion value. The rule for any of these criteria is to select the model with the minimum value. The criteria will not always agree on the “best” model. BIC is used by the Expert Modeler to select between the “best” ARIMA model and the “best” exponential smoothing model for a dependent series.

IBM Training

Demonstration 1

Identify an ARIMA model without using the Expert Modeler

Demonstration 1: Identify an ARIMA model without using the Expert Modeler

Demonstration 1: Identify an ARIMA model without using the Expert Modeler

Purpose:
You have monthly data on milk production in pounds per cow from January 1962 through December 1975. You would like to create an ARIMA model that you will use to forecast production. Initially, you decide to identify the model on your own without using the Expert Modeler. As a follow-up, you plan see how well the Expert Modeler the model you identified on your own.

Stream file: **unit_6_demonstration_1_start.str**

Folder: **C:\Training\0A028\06-Arima_Modeling\Start**

Task 1. Start and configure IBM SPSS Modeler 18.1.1.

1. From the **Start** menu, expand **IBM SPSS Modeler 18.1**, and then click **IBM SPSS Modeler 18.1**.

   Note: IBM SPSS Modeler 18.1.1 displays "IBM SPSS Modeler 18.1" in the Start menu and program name.
2. Click **Cancel** to close the **Welcome** dialog box.

   If you have already set the working directory in a previous demonstration or exercise, you can skip to Task 2.
3. From the **File** menu, click **Set Directory**.
4. Beside **Look in**, navigate to the **C:\Training\0A028** folder, and then click **Set**.

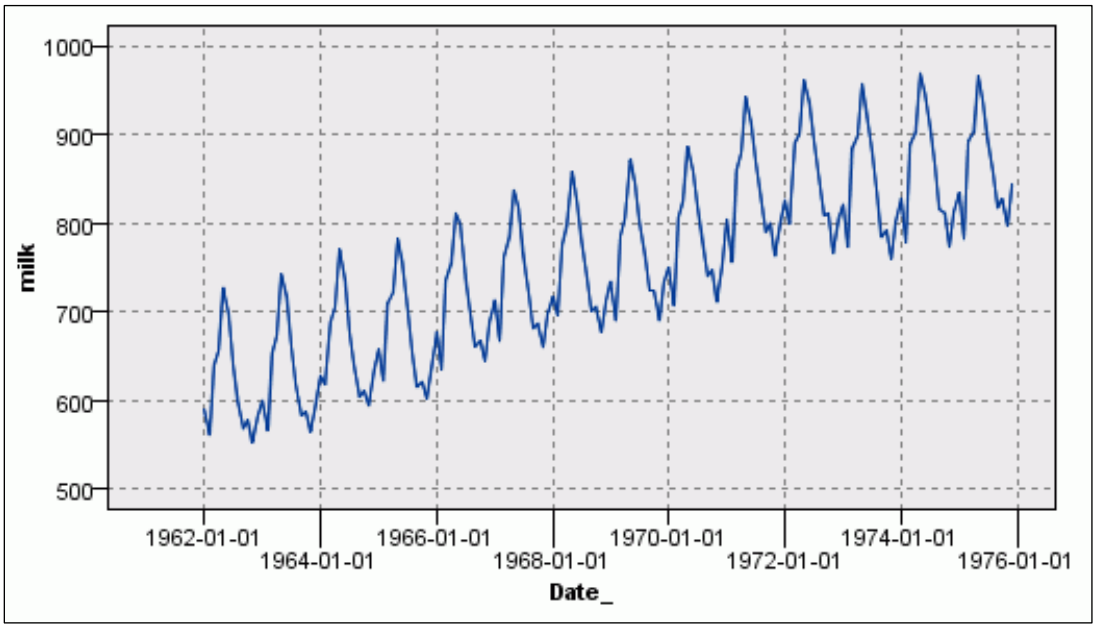
Task 2. Examine a time plot of the series.

1. From the **File** menu, click **Open Stream**, and then select **unit_6_demonstration_1_start.str**, located in the .**C:\Training\0A028\06-Arima_Modeling\Start** folder.
2. Run the **Table** node.

   The dataset stores the monthly milk production data from January 1962 through December 1975.
3. Close the **Table** results window.
4. From the **Graphs** palette, select the **Time Plot** node, and then add it downstream from the **Type** node.
5. Edit the **Time Plot** node.
6. Besides **Series**, select **milk.**

7. Besides **X axis label**, select the **Custom** option, and then select **Date_**.
8. Clear the **Normalize** option.
9. Click **Run**.

   The results appear as follows:



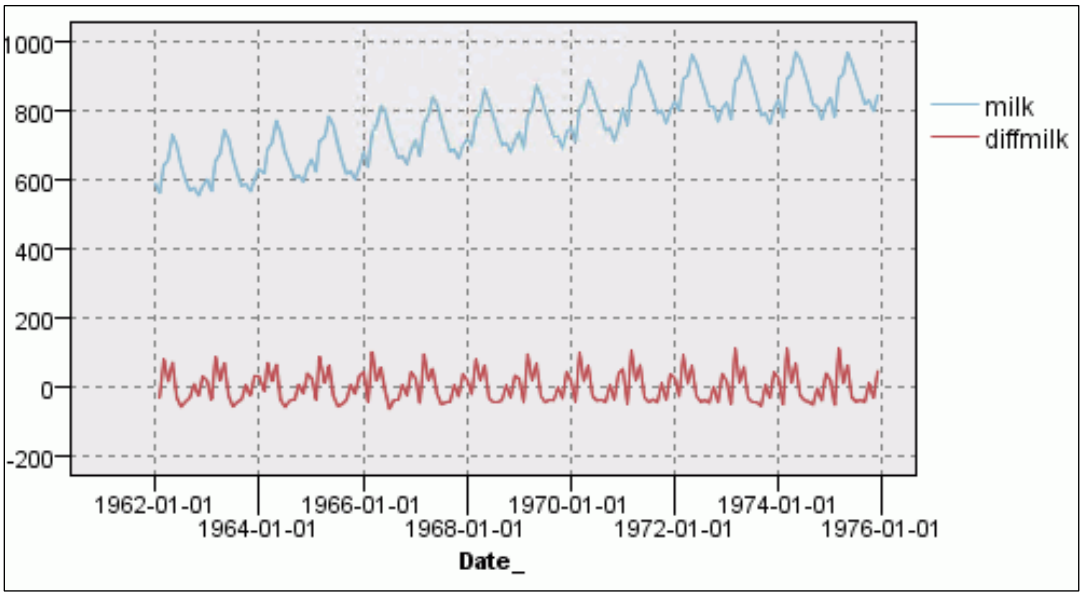
   The plot shows both clear seasonality and a trend, although the series levels off in the last few years. We know that a series must be stationary to be used in ARIMA, so the series will need to be differenced to create stationarity. The seasonality (variation within the year) seems to be relatively constant throughout.

   You will create a differenced version of the series and then display it in a time plot to see if differencing will remove the trend.

10. Close the window.
11. From the **Field Ops** palette, attach a **Derive** node to the **Type** node.
12. Edit the **Derive** node.
13. In the **Derive field** box, type **diffmilk**.
14. In the **Formula** box, type **milk - (@OFFSET(milk,1))**.
15. Close the **Derive node** dialog box.
16. From the **Graph** palette, attach a **Time Plot** node to the Derive node.
17. Edit the **Time Plot** node.
18. Besides **Series**, select **milk** and **diffmilk.**
19. Besides **X axis label**, enable the **Custom** option, and then select **Date_**.
20. Clear the **Normalize** option.

21. Clear the **Display series in separate panels** option.
22. Click **Run**.

    The results appear as follows:



    The series has been made stationary by differencing.

23. Close the **Time Plot** output.

    Now you will look at the autocorrelation plots.

Task 3. Perform further diagnostics with ACF & PACF charts.

Now you will that you have confirmed that the series is not stationary and needs to be differenced, the next step will be to determine the type of differencing you need to apply, either regular, seasonal or both. Seasonal differencing in this series is defined as a difference between the value of milk production value and each lag that is a multiple of the seasonality, which in this case is 12.

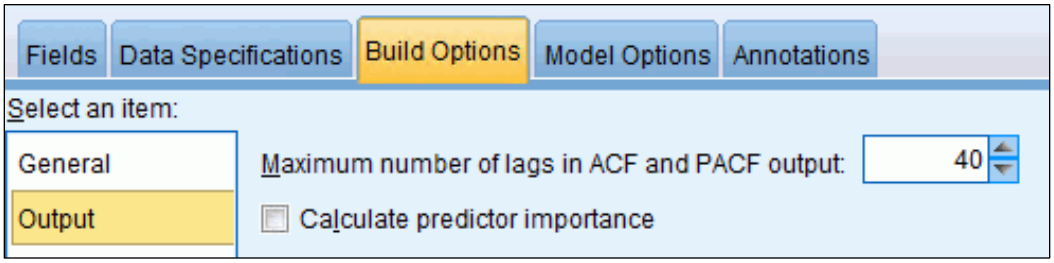
1. From the **Modeling** palette, add a **Time Series** node downstream from the **Type** node.
2. Edit the **Time Series** node.
3. Click the **Fields** tab, if necessary.
4. Enable the **Use custom field assignments** option.
5. Move **milk** into the **Targets** box.
6. Click the **Data Specifications** tab.

7. Click the **Observations** item on the left, if necessary.

   In this example, the observations are defined by a date field, named Date_. Because the data represented monthly figures, the time intervals between the observations are months.
8. Ensure that the **Observations are specified by a date/time field** option is enabled.
9. For **Date/time** field, select **Date_**.
10. For **Time interval**, select **Months**.
11. Click the **Build Options** tab.
12. From the **Method** menu, click **ARIMA**.

    You will create an ARIMA (0,0,0) (0,0,0) model with no seasonal or non-seasonal AR, I, or MA terms and then use ACF and PACF plots to check for correlation in the error terms.
13. From the **Select an item** area, click **Output**.
14. Change the value of **Maximum number of lags in ADF and PACF output** to **40**. That will give you a little over three seasons to examine.

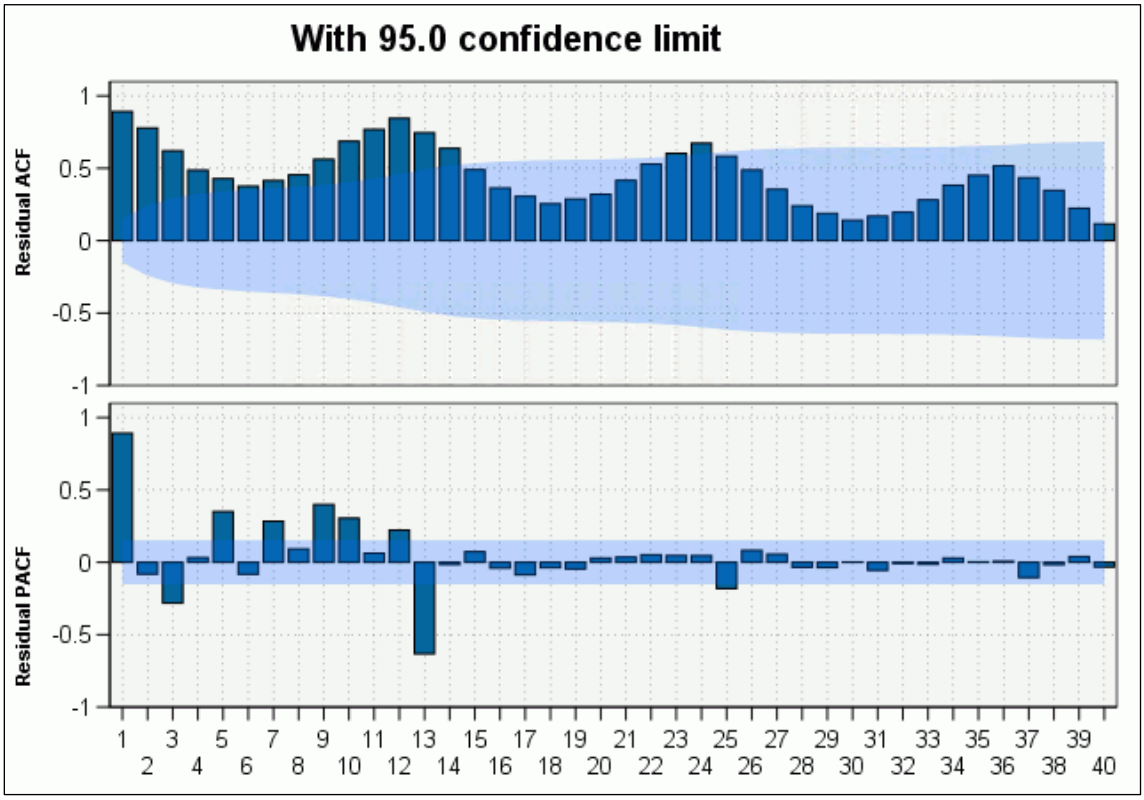
    The results appear as follows:



    That will give you a little over three seasons to examine.
15. Click **Run**.
16. Edit the **model nugget**.

17. On the left, click **Correlogram**.

    The results appear as follows:



    The seasonality is immediately apparent in the autocorrelations. The amount of milk production at dates 12 months apart is highly correlated, and this continues at 24 and 36 months. Also, there is nonstationarity, as lower order lags are large and significant.

    Because the ACF plot shows such obvious nonstationarity and seasonality, you need not spend much time with the PACF plot. The PACF plot is only useful after adjusting for these other effects.

18. Close the **model nugget**.

    It is clear that you will want to include some seasonal components in the ARIMA model, but not obvious how to decide on the specific parameters. The process proceeds in two steps:

    First, difference the series field until the ACF plot no longer shows seasonality and the series is stationary. This will tell you how many orders of differencing should be included.

    Secondly, use the ACF and PACF charts to determine the autoregressive and moving average terms, as you would with any ARIMA model, but also apply this to the seasonal factors.

    You will now attempt to identify an ARIMA model for the milk production series.

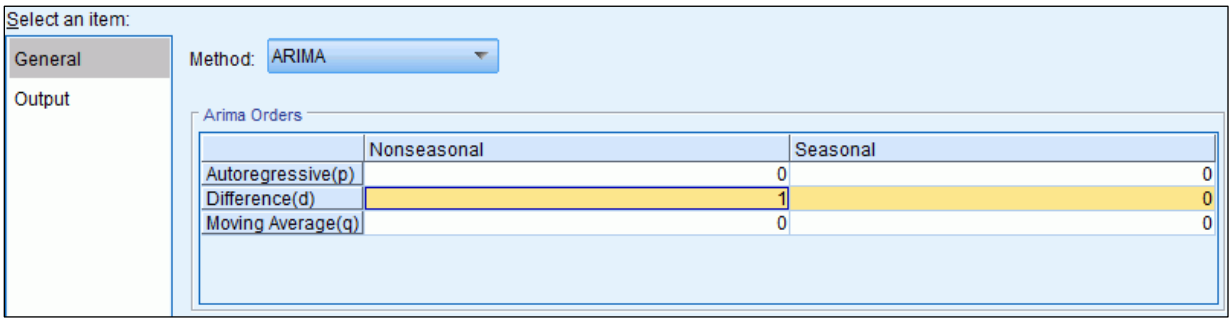
Task 4. Identify the ARIMA model.

You know that some differencing will be required, and we can do this directly in the dialog boxes.

1. Edit the **Time Series** node.
2. Click the **Build Options** tab.
3. Under **Select an item**, click **General**.
4. From the **Method** menu, click **ARIMA**.

   You will create an ARIMA (0,1,0) (0,0,0) model and then use ACF and PACF plots to check to see if further differencing needs to be done.
5. In the **Arima Orders** area, type **1** in the **Difference(d)** row, **Nonseasonal** column.

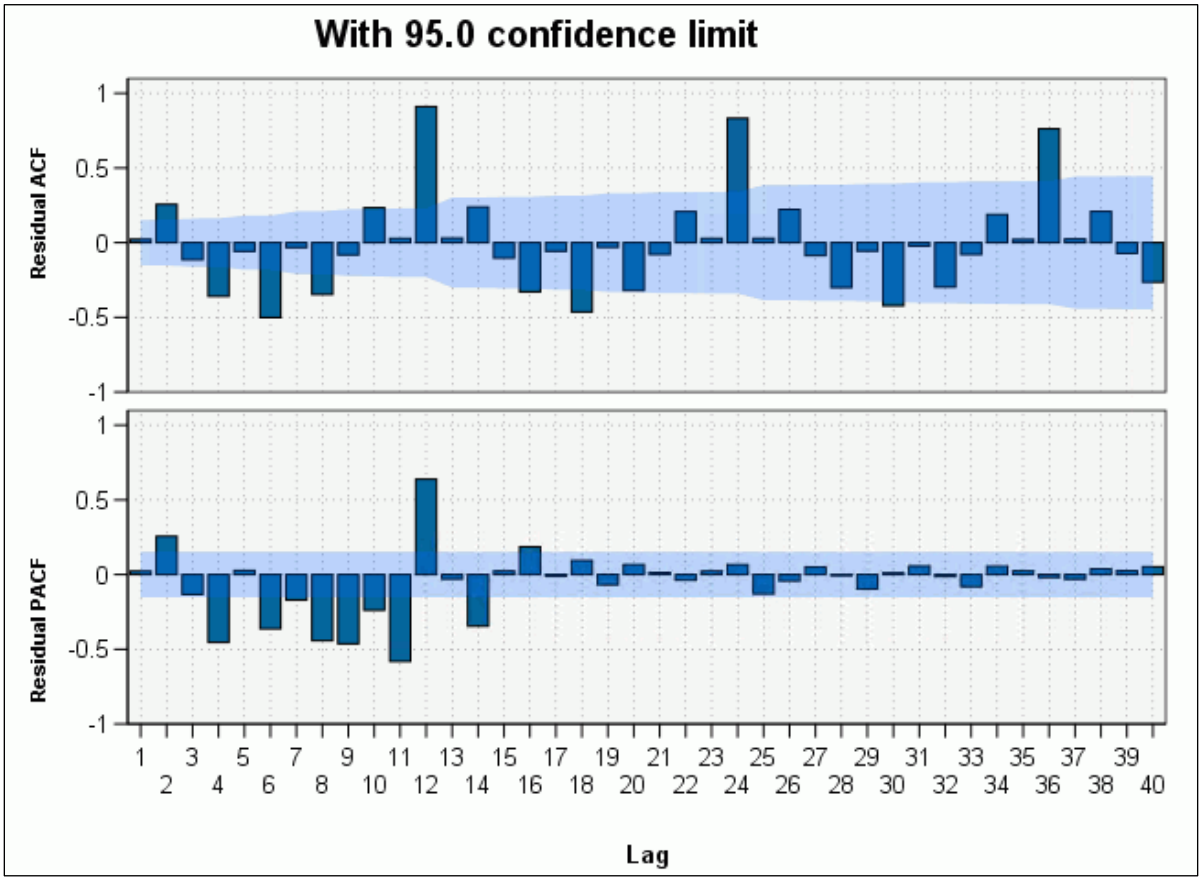
   The results appear as follows:



6. Click **Run**.
7. Edit the **model nugget**.

8. On the left, click **Correlogram**.

   The results appear as follows:



   The autocorrelations at lags 12, 24, and 36 are large and dampen very slowly, which indicates seasonal nonstationarity, which should be remedied by seasonal differencing.

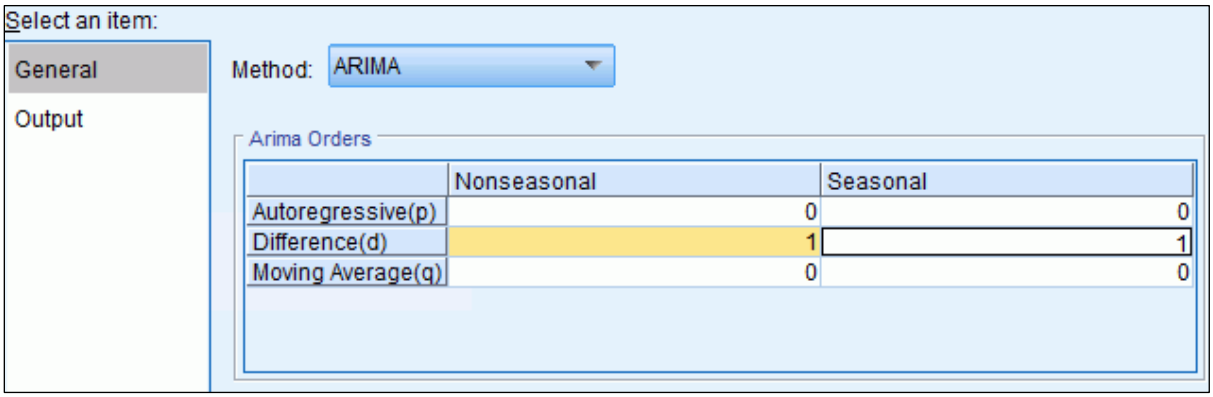
9. Close the **model nugget**.

10. Edit the **Time Series** node, and click the **Build Options** tab.

    This time, you will create an ARIMA (0,1,0) (0,1,0) model and then use ACF and PACF plots to check the results again.

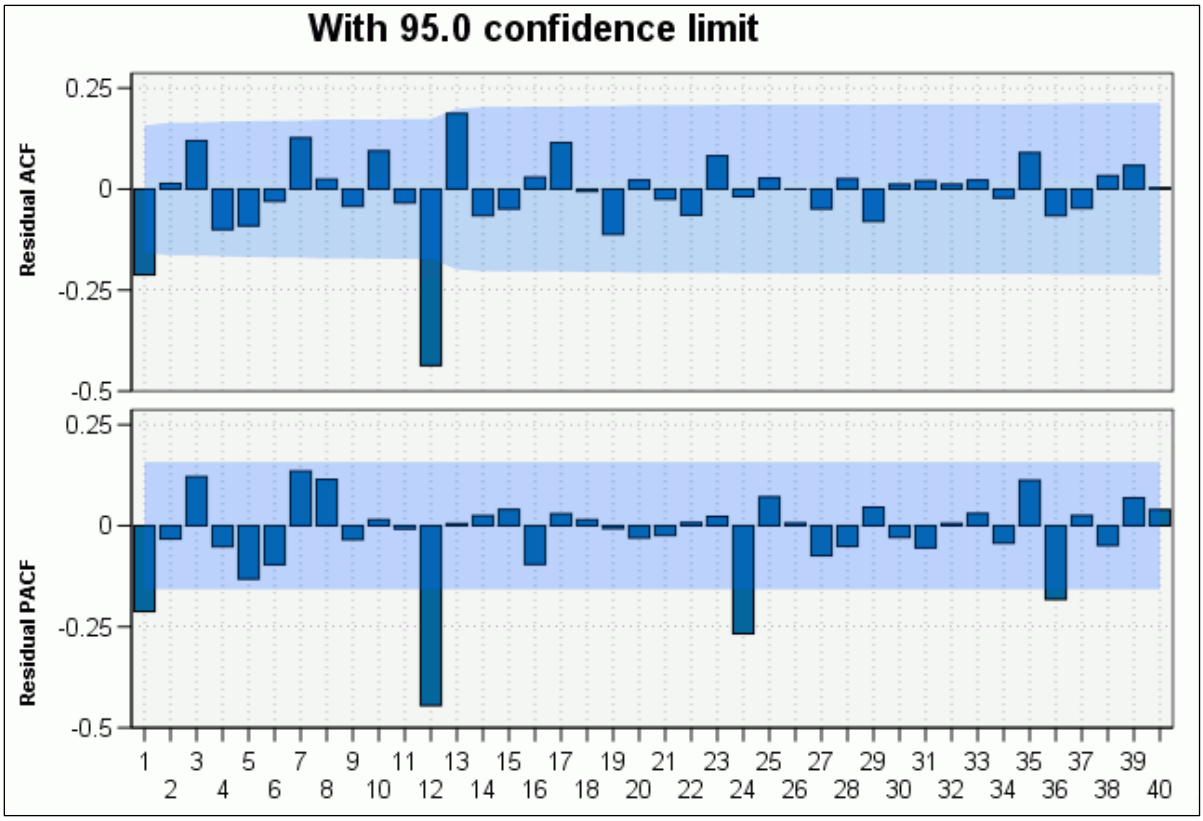
11. In the **Arima Orders** area, type **1** in the Seasonal **Difference(d)** row, **Seasonal** column.

    The results appear as follows:



12. Click **Run**.
13. Edit the **model nugget**.
14. On the left, click **Correlogram**.

    The results appear as follows:

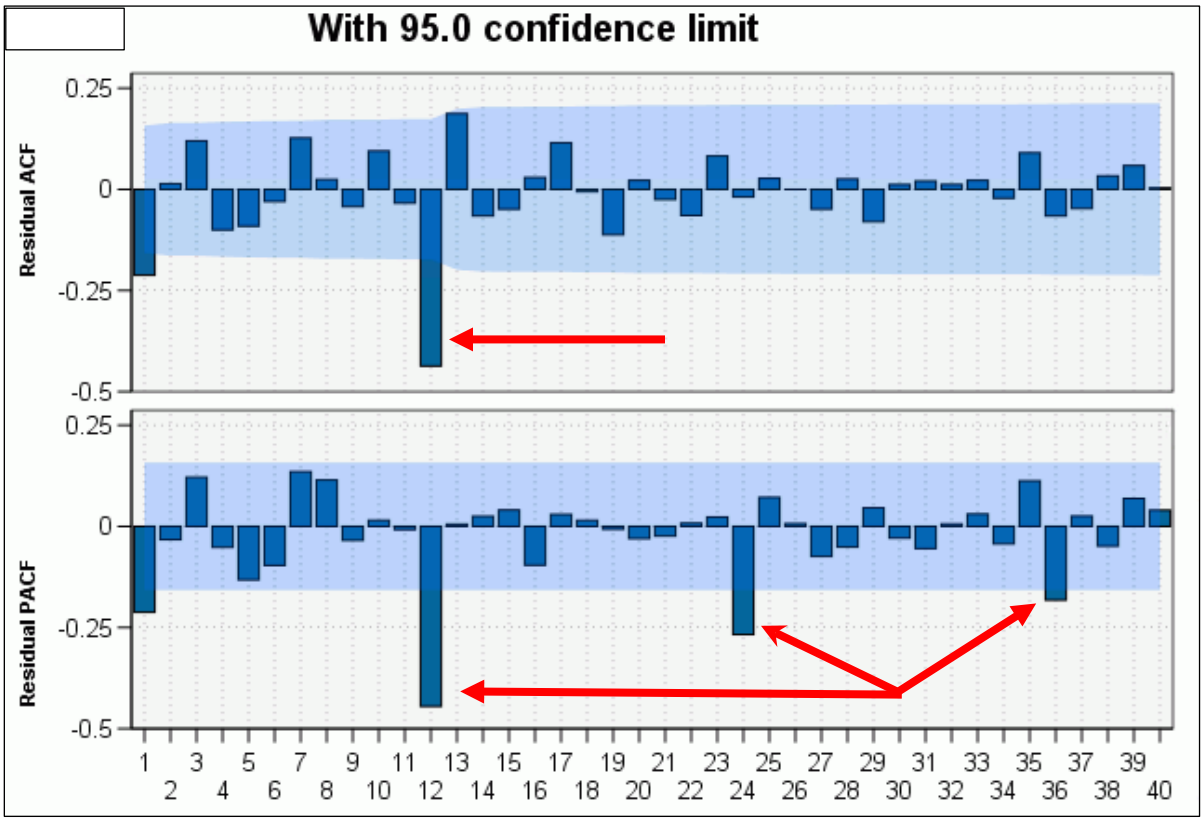


As before when you examined the ACF and PACF plots, you are looking for both seasonal and nonseasonal effects, using the same rules you learned before about ARIMA identification.

In the ACF plot, the autocorrelations are significant at lags 1, 12, and 13, suggesting the need for both seasonal and nonseasonal parameters in the ARIMA model. It is important to note that this is in addition to the seasonal differencing.

The PACF plot autocorrelations are significant at lags 1, 12, 24, and 36, again suggesting both seasonal and nonseasonal effects.

To help sort this out, focus on the ACF and PACF functions at only seasonal lags (12, 24, and 36) on the nonseasonally and seasonally differenced series.



You can see a classic pattern here of exponentially declining PACF values and one significant spike in the ACF. But since these are only at the seasonal lags, this suggests including a seasonal MA(1) term in the model.

You do not need an AR seasonal term because there is no evidence for that in the pattern of ACF and PACF values.

What about the nonseasonal terms? If you review again the ACF and PACF plots of the first differenced, seasonally differenced series, it is unclear whether you should add an AR or an MA term. The lower order lags do not fit the classic pattern. There is one significant ACF value at lag 1, and one significant PACF value at lag 1 as well. This suggests including an AR(1) nonseasonal term, or possibly an MA(1) nonseasonal term as well. You would often begin simply, with only the AR(1) term.

Therefore, in summary, you have identified an ARIMA(1,1,0)(0,1,1) model. There is both seasonal and nonseasonal differencing, both MA(1) seasonal and nonseasonal terms, and an AR(1) nonseasonal term. This means that the series is a function of past random shocks (moving average) and recent values (autoregressive).

Note

Some authorities suggest not including both AR and MA nonseasonal terms together (or AR and MA seasonal terms) because it can lead to overfitting.

15. Close the **model nugget**.

Task 5. Estimate the ARIMA model.

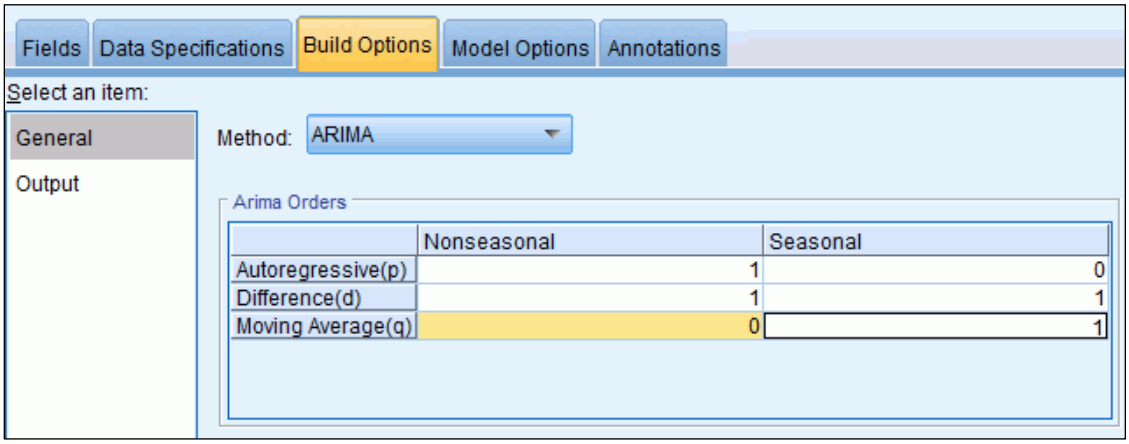
The next step will be to the estimate the model based on the diagnostic steps you have taken so far.

1. Edit the **Time Series** node, and click the **Build Options** tab.

   This time, you will create an ARIMA (1,1,0) (0,1,1) model and then use ACF and PACF plots to check the results again.

2. In the **Arima Orders** area:
   - Type **1** in the **Autoregressive(p)** row, **Nonseasonal** column.
   - Ensure you have **1** in the **Difference(d)** row, **Nonseasonal** column.
   - Ensure you have **1** in the **Difference(d)** row, **Seasonal** column.
   - Type **1** in the **Moving Average(q)** row, **Seasonal** column.

The results appear as follows:



3. Click **Run**.
4. Edit the **model nugget**.
5. On the left, click **Model Information**.

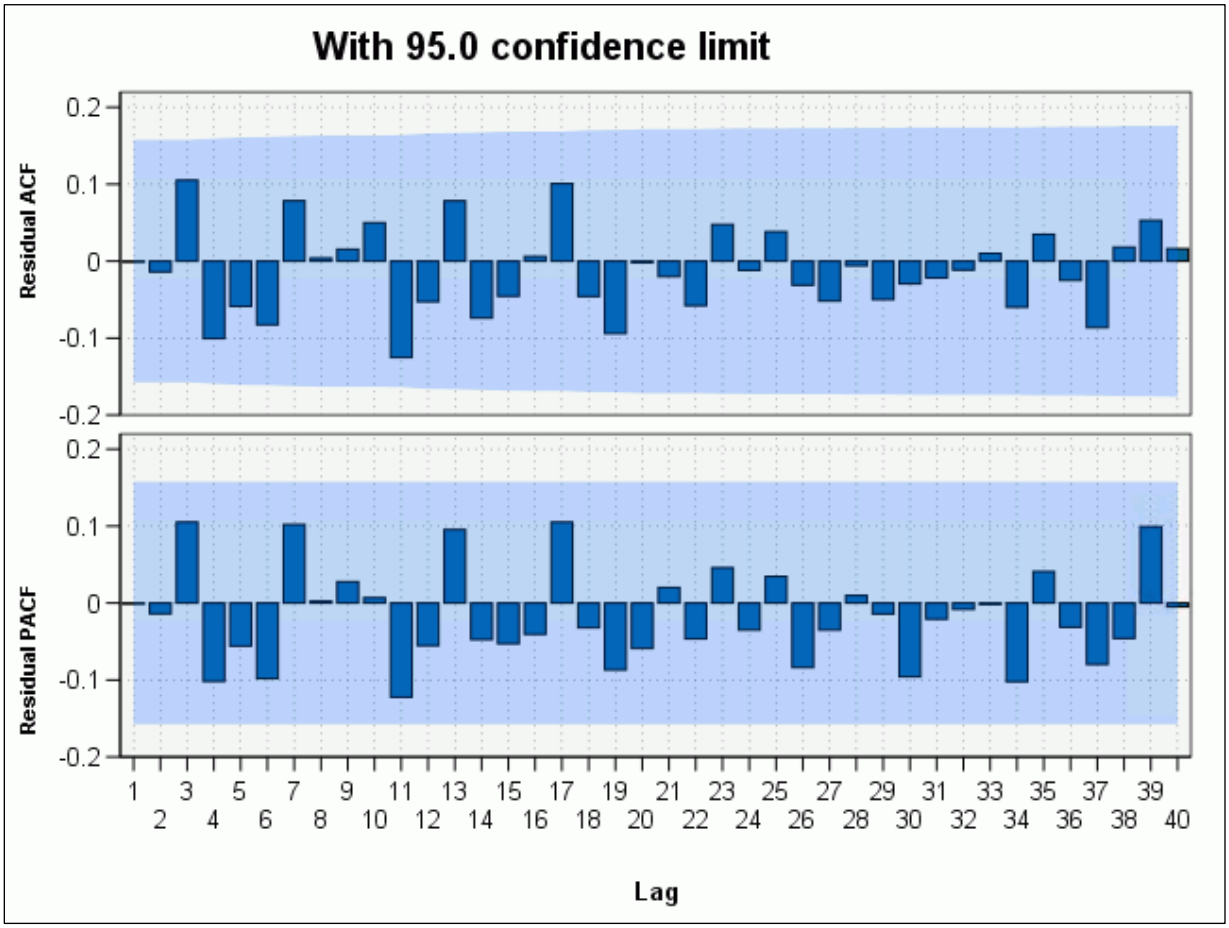
   The results appear as follows:

Model Information

Model Building Method		ARIMA
		Non-seasonal p=1,d=1,q=0; Seasonal p=0,d=1,q=1
Number of Predictors		0
Model Fit	MSE	55.301
	RMSE	7.436
	RMSPE	1.000
	MAE	5.678
	MAPE	0.759
	MAXAE	32.075
	MAXAPE	3.989
	AIC	624.954
	BIC	634.084
	R-Squared	0.994
	Stationary R-Squared	0.317
Ljung-Box Q(#)	Statistic	14.223
	df	16.0
	Significance	0.6

Just as you requested, the table shows that the model is ARIMA (1,1,0)(0,1,1), but does it fit? Based on the significance value of the Ljung-Box (Q), which is 0.6, the model does seem to be correctly specified. Thus, it appears that you eliminated the autocorrelation problems you identified in the diagnostic tests you performed earlier.

6. On the left, click on **Correlogram**.
   The results appear as follows:



   As you see in these plots, there is no autocorrelation, at least through 40 lags). This is an excellent result and indicates that the model has passed this test.

7. Click **Parameter Estimates**.
   The results appear as follows:

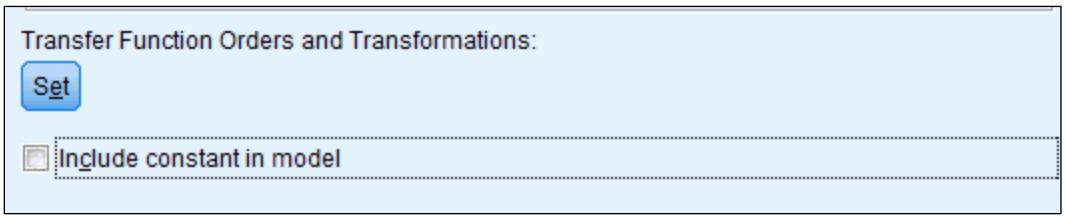
Parameter Estimates

				Coefficient	Std. Error	t	Significance
milk	No Transformation	Constant		-0.017	0.212	-0.081	0.935
		AR	Lag 1	-0.226	0.080	-2.827	0.005
		MA, Seasonal	Lag 1	0.620	0.074	8.430	0.000

   The parameter estimates table included only the AR(1) and MA, Seasonal(1) terms, both of which are significant. The constant term is also included in the table, even though it is not significant. This is because by default, when you manually specify an ARIMA model, the constant is automatically included in the model. You will rerun the model without the constant term.

8. Close the **model nugget**.
9. Edit the **Time Series** node.
10. Click the **Build Options** tab.
11. At the bottom of the dialog, in the **Transfer Function Orders and Transformations** area, clear the **Include constant in model** option.

    The results appear as follows:



    The step will be to rerun the model.
12. Click **Run**.
13. Edit the **model nugget**.
14. On the left, click **Model Information**.

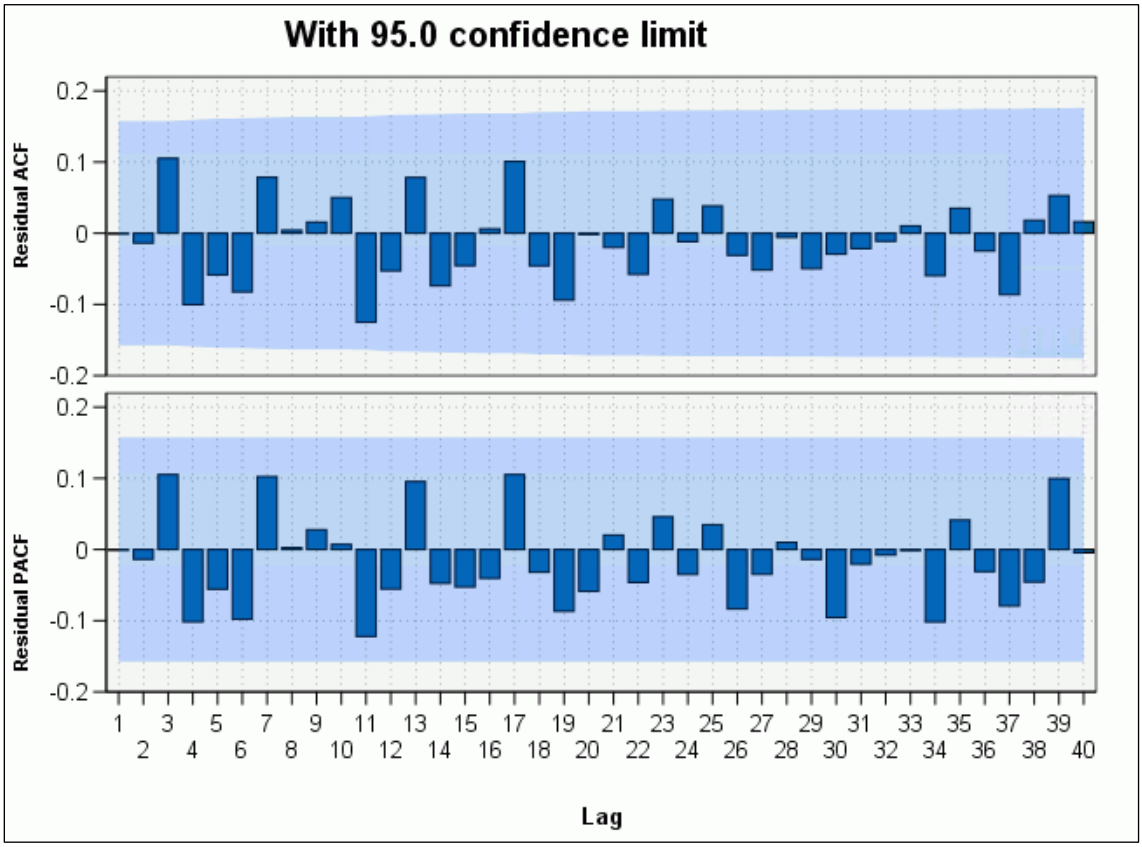
    The results appear as follows:

Model Information

Model Building Method		ARIMA
		Non-seasonal p=1,d=1,q=0; Seasonal p=0,d=1,q=1
Number of Predictors		0
Model Fit	MSE	54.941
	RMSE	7.412
	RMSPE	1.000
	MAE	5.681
	MAPE	0.760
	MAXAE	32.025
	MAXAPE	3.983
	AIC	622.956
	BIC	629.043
	R-Squared	0.994
	Stationary R-Squared	0.317
Ljung-Box Q(#)	Statistic	14.214
	df	16.0
	Significance	0.6

    The model is still the same, but the results have changed slightly. For example, when the constant was in the model, the MAPE statistic was 0.759 and now it is 0.760. Similarly, the MSE was 55.301 and now it is 54.941.

15. Click the **Correlogram** entry.
    The results appear as follows:



    The ACF and PACF plots still show no significant autocorrelations in the first 40 lags.

16. Click on **Parameter Estimates**.
    The results appear as follows:

Parameter Estimates

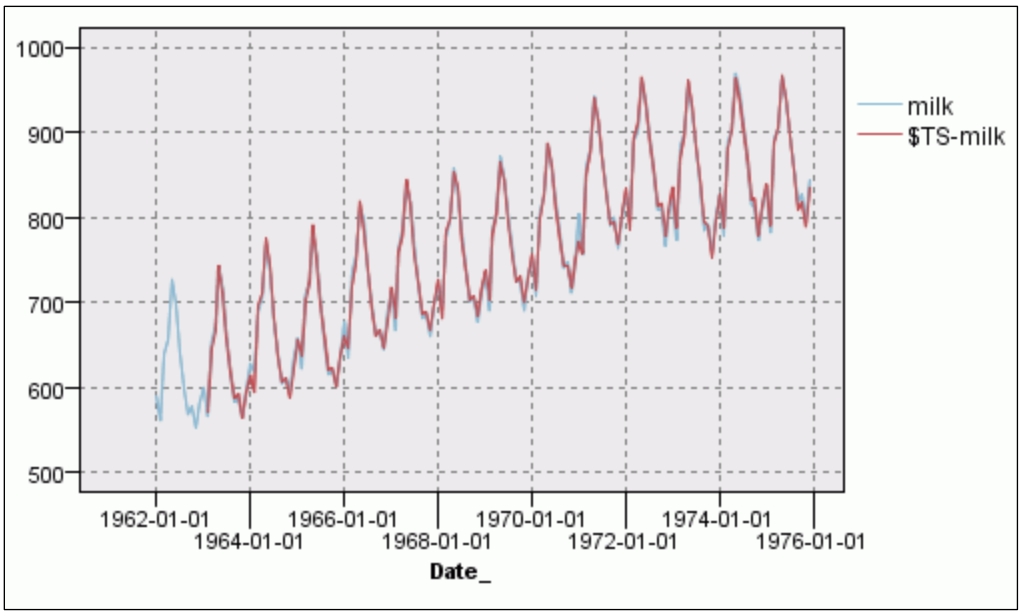
				Coefficient	Std. Error	t	Significance
milk	No Transformation	AR	Lag 1	-0.225	0.079	-2.836	0.005
		MA, Seasonal	Lag 1	0.620	0.073	8.461	0.000

    Now only the AR(1) and MA, Seasonal(1) terms are included in the model.
    The next step will be to look at the chart showing the observed and fit values to see how well the model fits the original series.

17. Close the **model nugget**.
18. From the **Graphs** palette, select the **Time Plot**, and add it downstream from the **model nugget**.

19. Edit the **Time Plot** node.
20. Beside **Series**, select **milk** and **$TS-milk .**
21. Besides **X axis label**, enable the **Custom** option, and then select **Date_**.
22. Clear the **Display series in separate panels** option.
23. Clear the **Normalize** option.
24. Click **Run**.

    The results appear as follows:



    It is clear that the fit is very good, but is it the best fitting model? You will check to see by letting the Expert Modeler select the best fitting model.

25. Close the **Time Plot** output window.

Task 6. Estimate the ARIMA model.

1. Edit the **Time Series** node, and click the **Build Options** tab.
2. From the **Method** menu, select **Expert Modeler**.
3. Click **Run**.
4. Edit the **model nugget**.

5. On the left, click **Model Information**.

   The results appear as follows:

Model Information

Model Building Method		ARIMA
		Non-seasonal p=1,d=1,q=0; Seasonal p=0,d=1,q=1
Number of Predictors		0
Model Fit	MSE	54.941
	RMSE	7.412
	RMSPE	1.000
	MAE	5.681
	MAPE	0.760
	MAXAE	32.025
	MAXAPE	3.983
	AIC	622.956
	BIC	629.043
	R-Squared	0.994
	Stationary R-Squared	0.317
Ljung-Box Q(#)	Statistic	14.214
	df	16.0
	Significance	0.6

   The Expert Modeler selected an ARIMA(1,1,0)(0,1,1) model. This means that you were able to obtain exactly the correct model on your own, although with much more preliminary work than simply invoking the Expert Modeler.

6. Click **Parameter Estimates**.

   The results appear as follows:

Parameter Estimates

				Coefficient	Std. Error	t	Significance
milk	No Transformation	AR	Lag 1	-0.225	0.079	-2.836	0.005
		MA, Seasonal	Lag 1	0.620	0.073	8.461	0.000

   The Expert Modeler automatically dropped the Constant term from the model because it was not significant.

   This completes the demonstration. You will create a clean state for the exercises.

7. Close the **model nugget**.
8. From the **File** menu, click **Close Stream**. Click **No** when asked to save.
9. From the **File** menu, click **New Stream**.

   Leave IBM SPSS Modeler open for the exercise.

Results:
You have successfully created an ARIMA model on your own without using the Expert Modeler.

You will find the completed stream in the following folder:
C:\Training\0A028\06-Arima_Modeling\Solutions

IBM Training

Unit summary

- Explain what ARIMA is
- Learn how to identify ARIMA model types
- Use time plots and autocorrelation plots to manually identify an ARIMA model that fits your data
- Check your results with the Expert Modeler

Unit summary

IBM Training

Exercise 1

Identify an ARIMA model without using the Expert Modeler

Exercise 1: Identify an ARIMA model without using the Expert Modeler

Exercise 1: Identify an ARIMA model without using the Expert Modeler

You have a data file which contains quarterly data on the U.S. savings rate for the years 1955 to 1979. The data have been seasonally adjusted. You would like to identify the best fitting ARIMA model without using the Expert Modeler in order to test your knowledge about how ARIMA works.

Stream file: **unit_6_exercise_1_start.str**

Folder: **C:\Training\0A028\06-Arima_Modeling\Start**

Task 1. Open the stream.

- From the **File** menu, click **Open Stream**.
- Navigate to the **C:\Training\0A028\06-Arima_Modeling\Start** folder, and then double-click **unit_6_exercise_1_start.str**.

Task 2. Identify the ARIMA model on your own.

- Run time plots and autocorrelation plots and look for patterns in the data. Does it appear to be stationary? Is there seasonality? If there is seasonality, does it look like additive or multiplicative seasonality?
- Use the results of the time plots and autocorrelation plots to identify the best fitting ARIMA model.

  What is the best ARIMA model?
- Evaluate the model using Fit statistics.

Task 3. Use the Expert Modeling to identify the best ARIMA model.

- Use the Expert modeler to identify the best fitting ARIMA model. What model did the Expert Modeler choose? Hint: the Expert Modeler will select Simple Seasonal Exponential Smoothing as the best fitting model, so you will need to select ARIMA Models Only to get the best fitting ARIMA model.
- Is the model the same as the one you created on your own? If not, which one do you prefer and why?
- Close the **model nugget**.
- Exit **IBM SPSS Modeler** without saving anything.

Exercise 1: Tasks and results

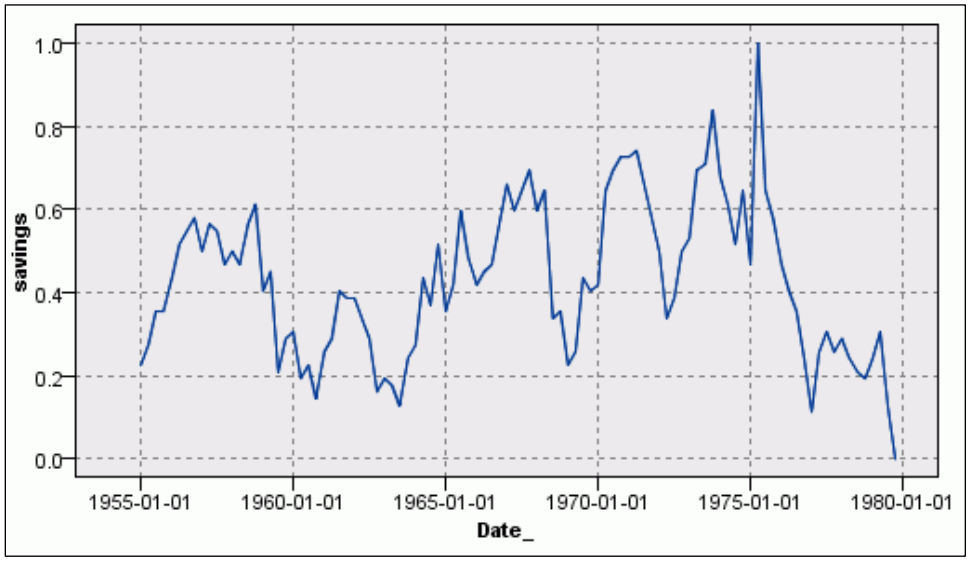
Task 1. Open the stream.

- From the **File** menu, click **Open Stream**.
- Navigate to the **C:\Training\0A028\06-Arima_Modeling\Start** folder, and then double-click **unit_6_exercise_1_start.str**.

Task 2. Identify the ARIMA model on your own.

- From the **Graphs** palette, select the **Time Plot**, and add it downstream from the **Type** node.
- Edit the **Time Plot** node.
- Besides **Series**, select **savings**.
- Besides **X axis label**, enable the **Custom** option, and then select **Date_**.
- Clear the **Normalize** option.
- Click **Run**.

  The results appear as follows:

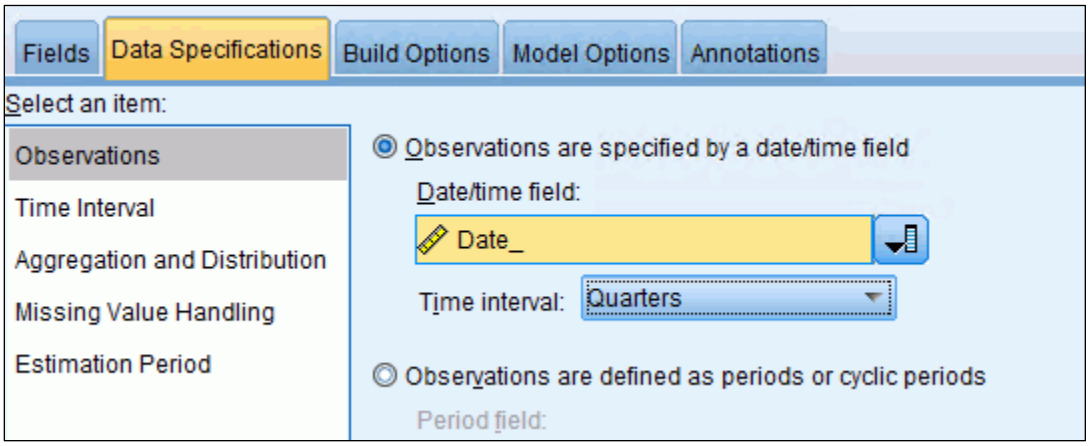


The series has several peaks and valleys, but there does not appear to be any seasonality in the series. There seems to be a slight upward trend in the series until it suddenly reaches its historic high in mid-1975 due pulse intervention. In April, 1975 a one-time tax rebate occurred meant to stimulate the economy, which was then in recession. Interventions can be included as predictors in an ARIMA model, though that is out of the scope of this course. At the end of the series, the savings rate declined rather rapidly down to 1960s levels.

- Close the **Time Plot** output.

  Now you will create autocorrelation plots to further understand the patterns.
- From the **Modeling** palette, add a **Time Series** node downstream from the **Type** node.
- Edit the **Time Series** node.
- Click the **Fields** tab, if necessary.
- Enable the **Use custom field assignments** option.
- Move **savings** into the **Targets** box.
- Click the **Data Specifications** tab.
- Click the **Observations** item on the left, if necessary.
- Ensure that the **Observations are specified by a date/time field** option is enabled.
- For **Date/time** field, select **Date_**.
- Beside **Time interval**, select **Quarters**.

  The results appear as follows:

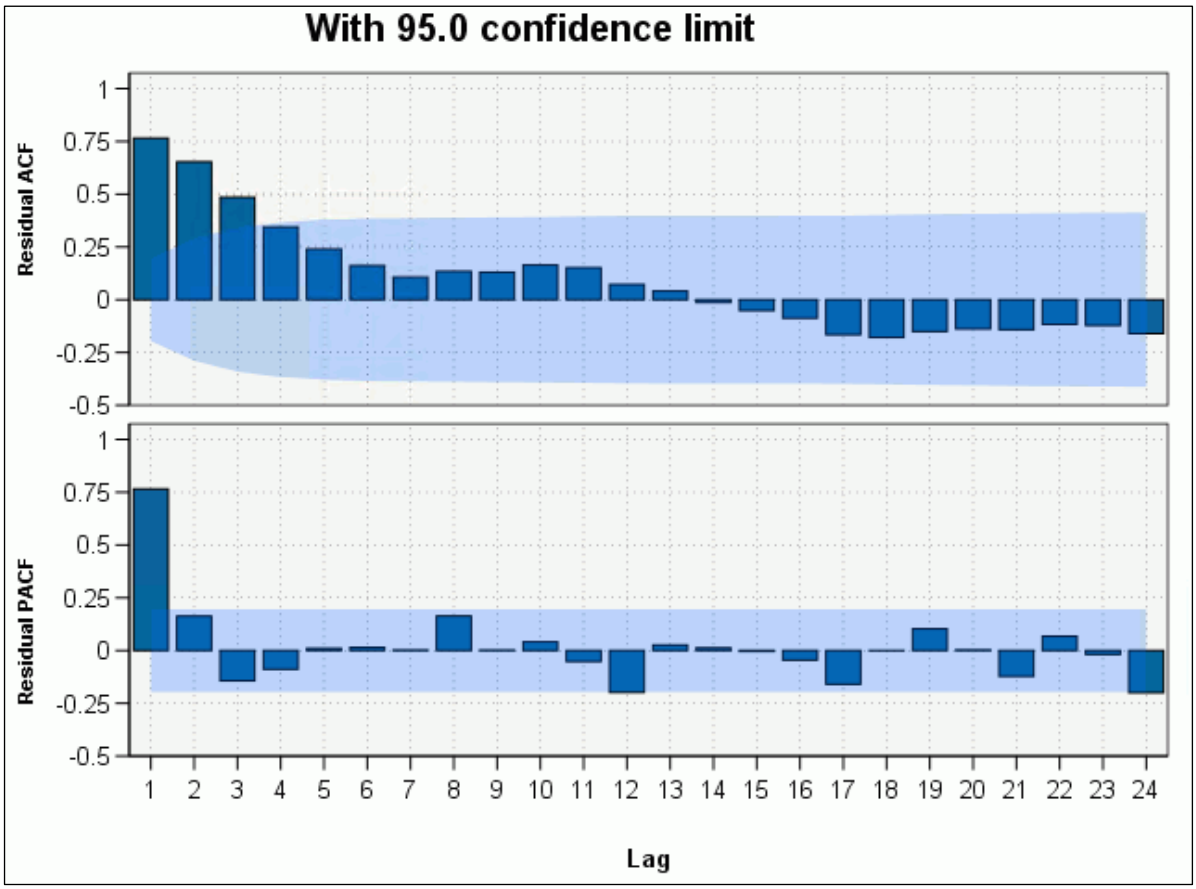


- Click the **Build Options** tab.
- Click the **General** item at the left, if necessary.
- Click the **Method** dropdown, and then click **Arima**.

  You will run an ARIMA(0,0,0)(0,0,0) model to see if you detect any patterns
- Click **Run**.
- Edit the **model nugget**.

- On the left, click **Correlogram**.

  The results appear as follows:



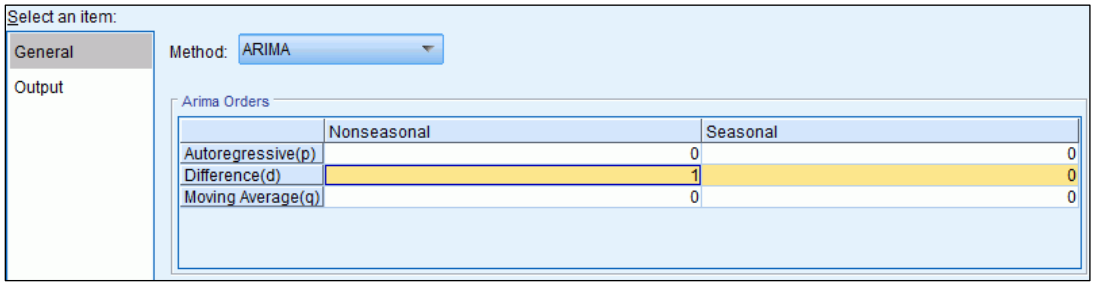
  While it may not be obvious the rate of decline is gradual or rapid, you will assume that because there was a slight upward trend in time plot, that the rate of decline is gradual. This means the series is not stationary and you will need to difference it before continuing.

- Close the **model nugget**.
- Edit the **Time Series** node.
- Click the **Build Options** tab.
- From the **Method** menu, click **ARIMA**.

  You will create an ARIMA (0,1,0) (0,0,0) model and then use ACF and PACF plots to check to see if further differencing needs to be done.

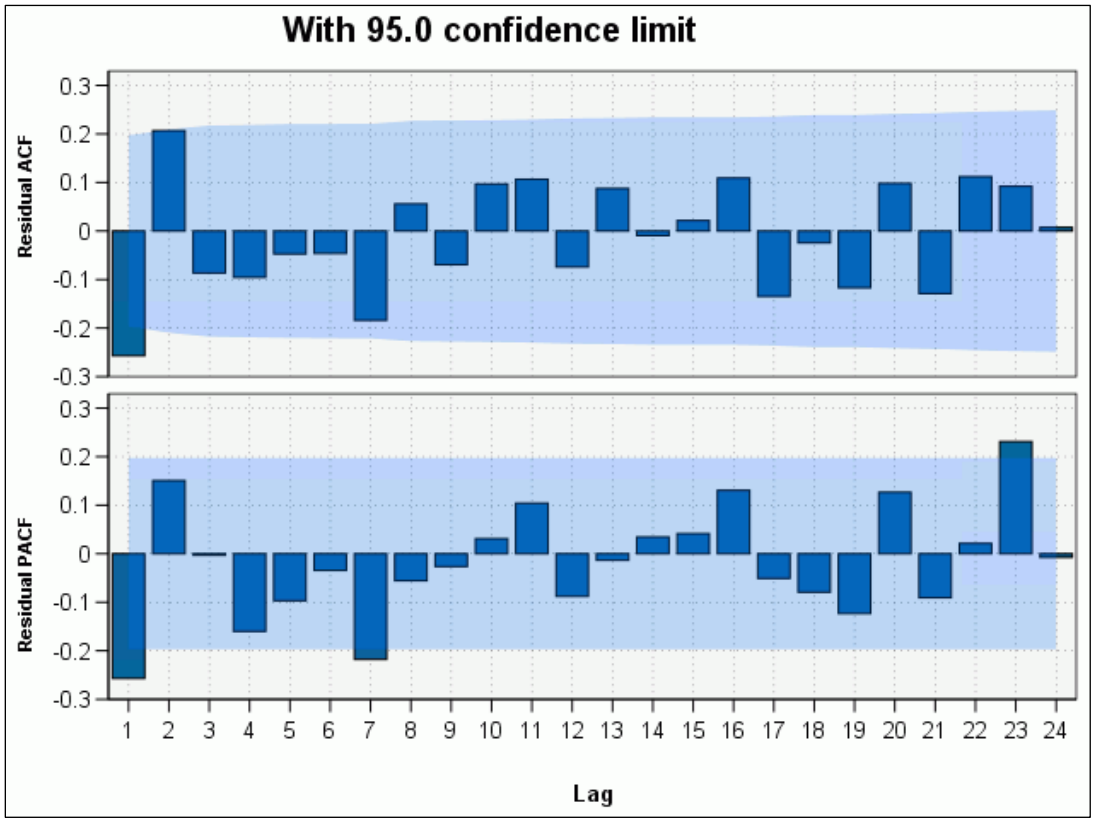
- In the **Arima Orders** area, type **1** in the **Nonseasonal** column**, Difference(d)** row.

  The results appear as follows:



- Click **Run**.
- Edit the **model nugget**.
- On the left, click **Correlogram**.

  The results appear as follows:

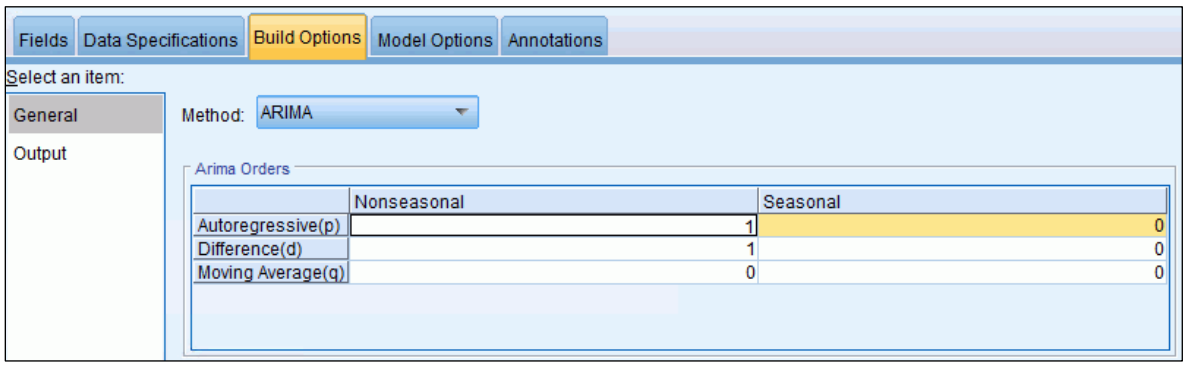


There is a significant spike at lag 1 in both the ACF and PACF plots. This suggests including an AR(1) nonseasonal term or possibly an MA(1) nonseasonal term as well. You will start by adding a nonseasonal AR(1) term.

- Close the **model nugget**.
- Edit the **Time Series** node.
- Click the **Build Options** tab.
- From the **Method** menu, ensure that **ARIMA** is selected.

  You will create an ARIMA (1,1,0) (0,0,0) model and then use ACF and PACF plots to check to see if further differencing needs to be done.
- In the **Arima Orders** area, type **1** in the **Autoregressive(p)** row, **Nonseasonal** column.

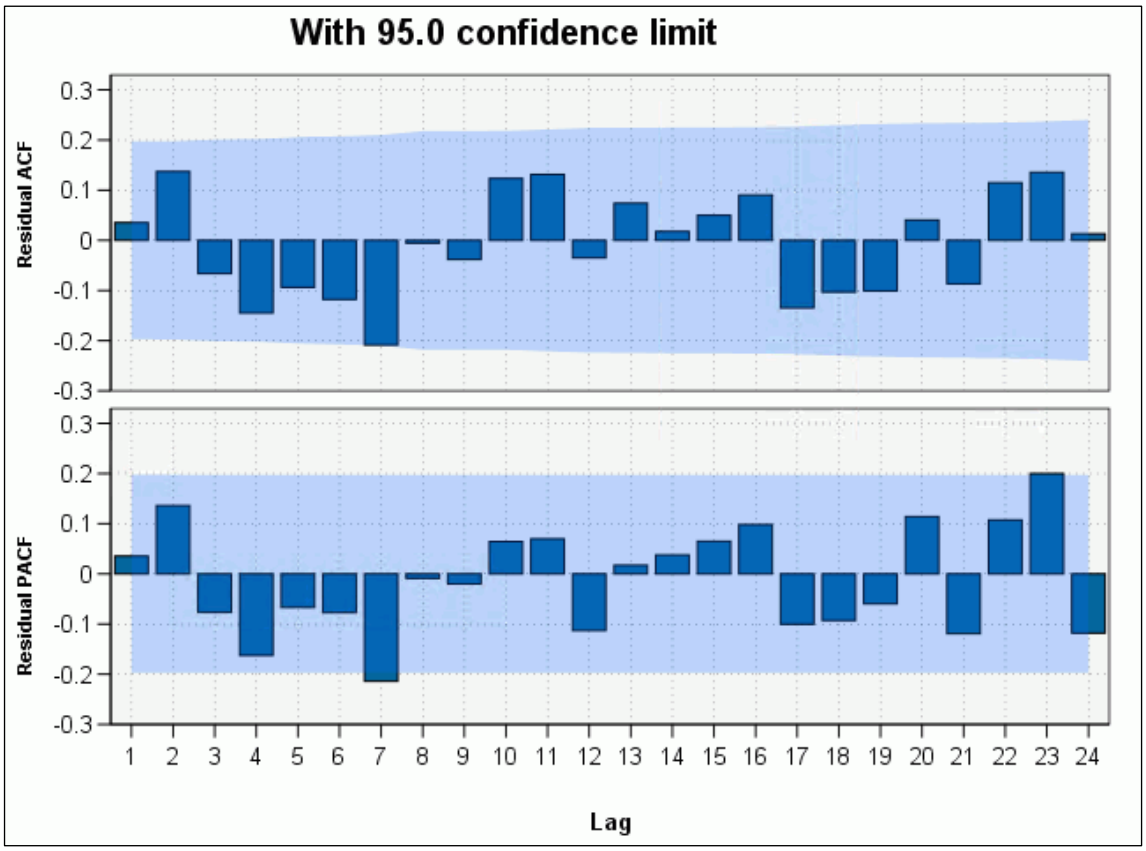
  The results appear as follows:



- Click **Run**.
- Edit the **model nugget**.

- On the left, click **Correlogram**.

  The results appear as follows:



There is a significant correlation at lag 7. You will examine the Ljung-Box Q results in the Model Information section of the output to see if this is cause for concern. The Ljung-Box Q statistic tests whether the residuals up to lag k are uncorrelated. As such it provides one overall test for any significant autocorrelations at the given lag or fewer, rather than examining many autocorrelation tests, each at the 0.05 level.

- On the left, click **Model Information**.

  The results appear as follows:

Model Information

Model Building Method		ARIMA
		Non-seasonal p=1,d=1,q=0; Seasonal p=0,d=0,q=0
Number of Predictors		0
Model Fit	MSE	0.503
	RMSE	0.709
	RMSPE	11.372
	MAE	0.537
	MAPE	8.828
	MAXAE	3.033
	MAXAPE	32.251
	AIC	-65.981
	BIC	-60.791
	R-Squared	0.620
	Stationary R-Squared	0.067
Ljung-Box Q(#)	Statistic	21.326
	df	17.0
	Significance	0.2

  The Ljung-Box Q significance value is 0.2, which indicates that there is no significant autocorrelation up to lag 24. Therefore, the fact that one individual lag was slightly significant in the ACF and PACF plots is probably not cause for much concern.

- Click **Parameter Estimates**.

  The results appear as follows:

Parameter Estimates

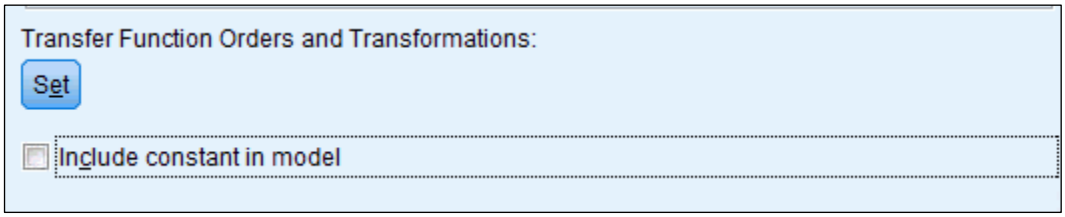
				Coefficient	Std. Error	t	Significance
savings	No Transformation	Constant		-0.013	0.057	-0.232	0.817
		AR	Lag 1	-0.258	0.099	-2.614	0.010

  Because the Constant is not significant, you will need to rerun the analysis without including the constant.

- Close the **model nugget**.
- Edit the **Time Series** node.
- Click the **Build Options** tab.

- At the bottom of the dialog, in the **Transfer Function Orders and Transformations** area, clear the **Include constant in model** option.

  The results appear as follows:



  The next step will be to rerun the model.

- Click **Run**.
- Edit the **model nugget**.
- On the left, click **Model Information**.

  The results appear as follows:

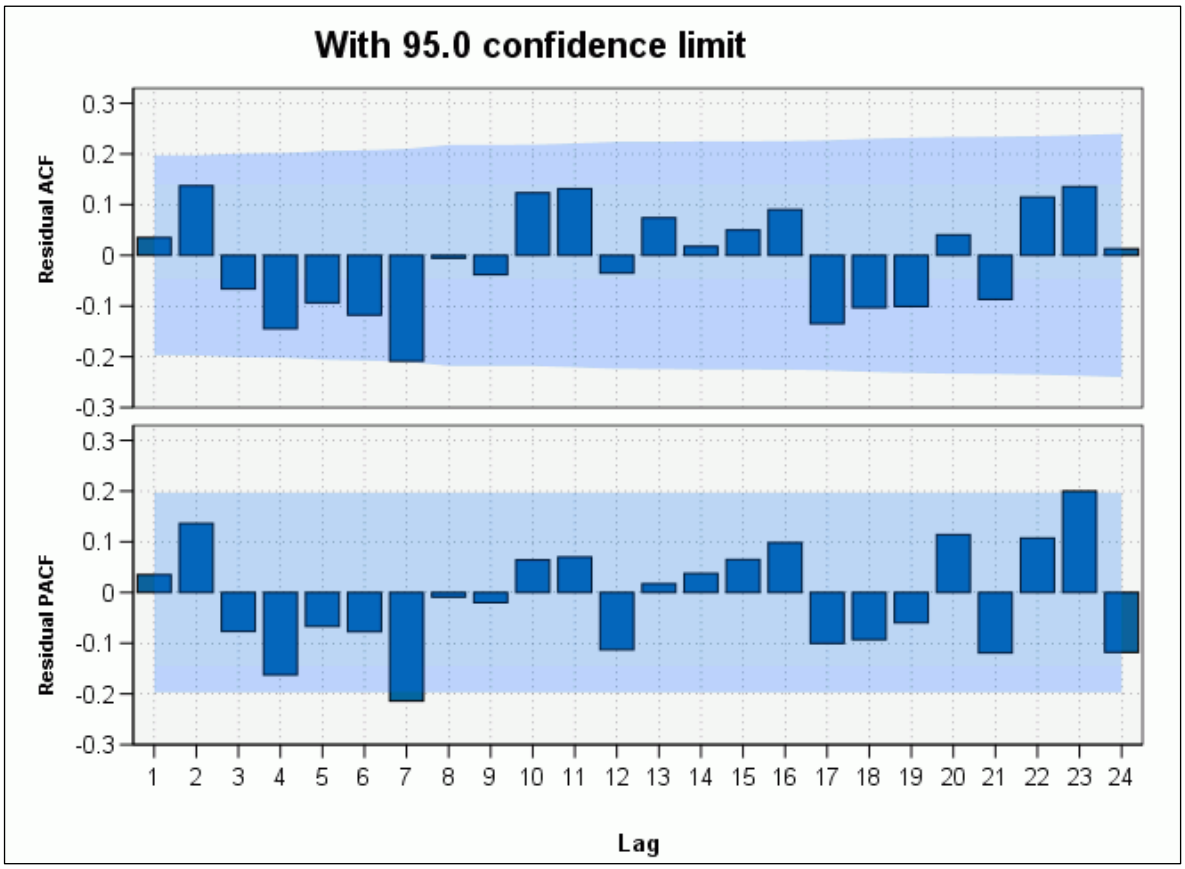
Model Information

Model Building Method		ARIMA
		Non-seasonal p=1,d=1,q=0; Seasonal p=0,d=0,q=0
Number of Predictors		0
Model Fit	MSE	0.498
	RMSE	0.706
	RMSPE	11.419
	MAE	0.536
	MAPE	8.819
	MAXAE	3.017
	MAXAPE	32.549
	AIC	-67.927
	BIC	-65.332
	R-Squared	0.620
	Stationary R-Squared	0.066
Ljung-Box Q(#)	Statistic	21.316
	df	17.0
	Significance	0.2

  The table includes a number of fit statistics. The Mean Absolute Error (MAE) is 0.536, meaning that on average, the prediction is about .5 off the target. Putting this in percentage terms, the Mean Absolute Percent Error (MAPE) is 8.819, which means that on average the predictions are off by 8.819%. The Ljung-Box Q significance value of 0.2 indicates that because there is no pattern in the residuals the model is correctly specified.

- Click **Correlogram**.

  The results appear as follows:



  The ACF and PACF plots still indicate no significant autocorrelation, except at lag 7, even without the constant included in the model.

- Click **Parameter Estimates**.

  The results appear as follows:

Parameter Estimates

				Coefficient	Std. Error	t	Significance
savings	No Transformation	AR	Lag 1	-0.258	0.098	-2.624	0.010

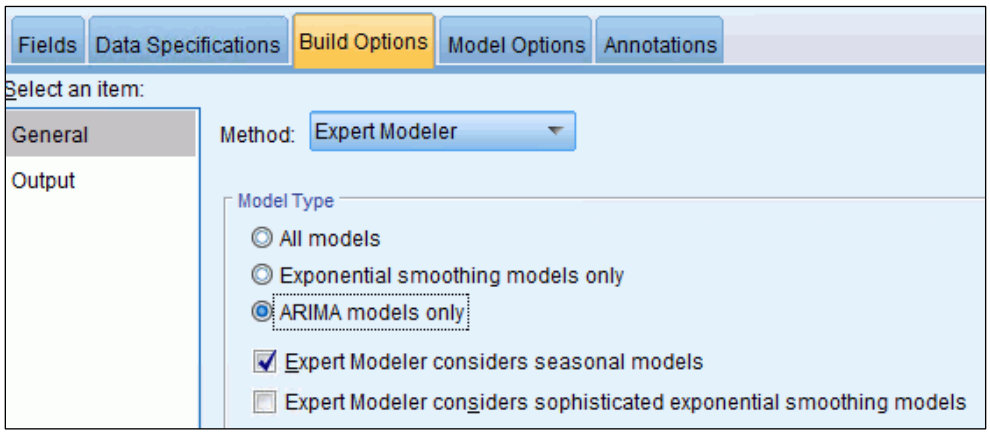
  The constant is no longer included in the model. The AR(1) term is significant at the 0.05 level. The value of -0.258 for the AR(1) term means that the value of savings at t-1 is multiplied by that coefficient to predict savings at time t.

- Close the **model nugget**.

Task 3. Use the Expert Modeling to find the best ARIMA model.

- Edit the **Time Series** node.
- From the **Method** menu, select **Expert Modeler**.
- In the **Model Type** area, click **ARIMA models only**.

  The results appear as follows:



- Click **Run**.
- Edit the **model nugget**.
- On the left, click **Model Information**.

  The results appear as follows:

Model Information

Model Building Method		ARIMA
		Non-seasonal p=0,d=1,q=1; Seasonal p=0,d=0,q=0
Number of Predictors		0
Model Fit	MSE	0.507
	RMSE	0.712
	RMSPE	11.453
	MAE	0.538
	MAPE	8.845
	MAXAE	3.105
	MAXAPE	33.233
	AIC	-66.247
	BIC	-63.652
	R-Squared	0.613
	Stationary R-Squared	0.050
Ljung-Box Q(#)	Statistic	21.400
	df	17.0
	Significance	0.2

The Expert Modeler selected an ARIMA(0,1,1) model instead of the ARIMA(1,1,0) you specified. When you were developing your model, at one point, you had to choose between adding an AR(1) term or a MA(1) term and you chose the former. Because you were unsure, you could have tried creating an ARIMA(1,1,0) and an ARIMA(0,1,1) model and seen which one fit better, but you did not. The Expert Modeler automatically tries all possible alternatives to come up with the best model.

Which is the better model? For comparison purposes both Model Information tables are displayed below:

Model Information

Model Building Method		ARIMA
		Non-seasonal p=1,d=1,q=0; Seasonal p=0,d=0,q=0
Number of Predictors		0
Model Fit	MSE	0.498
	RMSE	0.706
	RMSPE	11.419
	MAE	0.536
	MAPE	8.819
	MAXAE	3.017
	MAXAPE	32.549
	AIC	-67.927
	BIC	-65.332
	R-Squared	0.620
	Stationary R-Squared	0.066
Ljung-Box Q(#)	Statistic	21.316
	df	17.0
	Significance	0.2

Your Model

Model Information

Model Building Method		ARIMA
		Non-seasonal p=0,d=1,q=1; Seasonal p=0,d=0,q=0
Number of Predictors		0
Model Fit	MSE	0.507
	RMSE	0.712
	RMSPE	11.453
	MAE	0.538
	MAPE	8.845
	MAXAE	3.105
	MAXAPE	33.233
	AIC	-66.247
	BIC	-63.652
	R-Squared	0.613
	Stationary R-Squared	0.050
Ljung-Box Q(#)	Statistic	21.400
	df	17.0
	Significance	0.2

Expert Modeler

The Expert Modeler chose the ARIMA (0,1,1) based on having the lowest BIC value. The BIC value of the model with an MA(1) term is -63.652 versus -65.332 for your model. The model with the lower BIC is preferred. However, you may still prefer your model over the one the Expert Modeler selected because your model has a lower MAPE (8.819% vs. 8.845%). Although these two values are very close, the predictions by your model are slightly more accurate than those by the Expert Modeler, even though it does not fit quite as well.

- Close the **model nugget**.
- Exit **IBM SPSS Modeler** without saving anything.

You will find the completed stream in the following folder:
C:\Training\0A028\06-Arima_Modeling\Solutions

IBM Training