Chapter 8

Time Series Analysis

Chapter Outline

What You Will Learn in This Chapter

- How do you forecast what might happen tomorrow or in the next few months?
- Who needs to forecast data over time?
- What are the components of a time series and how can they be measured?
- What types of data can be used in time series analysis?
- What results does traditional ARIMA provide?
- How does the Microsoft Time Series estimation work?
- How can the effects of other series be incorporated into the prediction?

Aunt Bessie's

Aunt Bessie's is a British food manufacturing company part of the William Jackson Food Group. It began in 1974 making millions of frozen Yorkshire puddings for Butlin's Holiday Camps [http://www.auntbessies.co.uk/about]. By 2011, the company produced 20 million Yorkshire puddings per week at one plant in Hull. In 2011, the company installed a new QD ERP tool tightly integrated with a Preactor APS (advanced planning and scheduling) system to help the company not just monitor production but also optimize schedules and handle capacity planning. Bhris Buckle, supply planning manager, notes that "the increased visibility from Preactor has also helped us to respond quicker, especially when we have a problem on a line. Beforehand, it could take a day for us even to notice, and then additional time to work out how best to react. Now we can see much more quickly when a problem is occurring and investigate different scenarios for dealing with it." [Tinham 2011] The forecasting capabilities helped reduce the amount of inventory cluttering up manufacturing facilities. They are also useful for integrating maintenance with operations. Maintenance can be scheduled for lower production points leading to fewer issues with disrupting critical production runs. The Preactor system models the production process and matches the desired production runs with the quantity of input materials needed. The planning system can search the schedules and plant capacity valuations to find optimal schedules [Preactor Web site]. The historical data feeds the long-run forecast and planning system to identify needed stock levels and resource scheduling.

Time series statistics are used to identify patterns and make forecasts. They can identify and handle random errors, seasonality, and cyclical changes, but might not catch major structural shifts.

Brian Tinham, "Sense and Sensitivity, *Works Management*, September 14, 2011. http://www.worksmanagement.co.uk/information-technology/features/sense-and-sensitivity/36778/

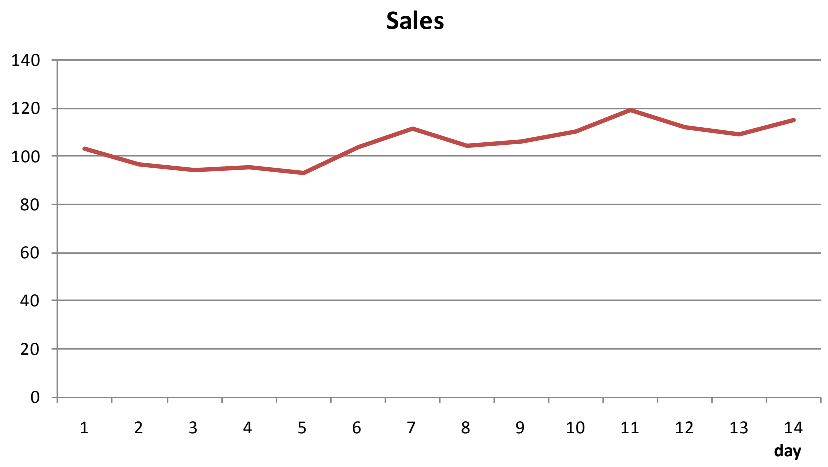
Introduction

How do you forecast what might happen tomorrow or in the next few months? It is not really possible to see into the future. However, in many cases, it is relatively easy to predict that if the same thing happened for the last three days, it will probably happen again tomorrow. This concept is called **auto regression**. To forecast the future, you can look back at the past few time periods and use those as a starting point. The information from the past will carry into the near future. Of course, as you try to predict farther into the future, an increasingly random number of events can arise that will make your forecast wrong. A **time series** is a set of outcome data that depends only on a time variable. As shown in Figure 8.1, a time series is relatively easy to understand. A single time series consist of one variable of interest (Sales) and shows how it varies over time. Time can be measured in almost any interval: hours, days, months, quarters, years, and so on. Months are a useful measure in business, but sometimes only quarterly data is available. Data might change due to external and random events, but at least to some extent, it follows some pattern based on recent values. If you can determine the relationship between the data and time, it is relatively easy to make forecasts.

Of course, a time series can be more complex. In many business settings, sales will often have a **seasonal effect**—where sales will increase or decrease in certain months of the year. For instance, as an extreme case, typical toy stores in the U.S. experience about 70 percent of their sales in the last quarter of the year—as holiday gift purchases. The essence of time series forecasting is to find these patterns to the data and use them to predict what will happen at the same point in the pattern in the future.

Figure 8.1

Simple time series. The simplest time series data consists of a single attribute plotted over a time interval. Here, total Sales are shown for each day. In many cases, a simple forecast can be made based on the information from the most recent time periods.



Almost every time series will have a random component that cannot be predicted. If the randomness is large, the prediction will almost always be inaccurate. Hence, every analysis should examine the degree of reliability based on the size of the random component compared to the predictable patterns. Some problems are relatively stable with small random elements. Others have high inherent random components which make it difficult to predict with any certainty. It is important to measure the level of randomness to provide a measure of variability—so a range forecast can be made instead of just a single point estimate.

Several tools can be used to analyze and forecast time series data. Simple tools include exponential smoothing—which averages out the random element; trend analysis—similar to regression which looks only at the basic direction over time; and auto regression—which measures the impact of prior period on trends. More complex tools combine these techniques. The most common method was developed by Box-Jenkins (see Box and Jenkins 1970) and is often called **ARIMA** analysis, which is an acronym for auto-regressive, integrated moving average. Because forecasting is so important to so many areas, several other techniques have been invented to analyze patterns over time. Some, such as ARIMA are relatively easy to automate and use for data mining. Others can be complex, such as spectral analysis, and can require considerable knowledge and supervision by the analyst. This chapter focuses on the easier methods that require less supervision.

Business Situation

Who needs to forecast data over time? Finally, a question that is easy to answer: everyone makes forecasts. Some disciplines and some people are better than others, but some processes are more stable with fewer random elements. Predicting the weather is hard, particularly with microclimates, but mostly because of enormous random effects. Predicting economic trends is relatively straightforward—up to a couple of months. But economic activity is also subject to catastrophic (sudden) changes.

Why do businesses need to make forecasts? Consider the situation of a retail store. How long does it take to get products from the manufacturer onto the shelves? A few days, weeks, months? Although American production and distribution systems have become more efficient, months is usually the correct answer. So, a store has to order products several weeks or months before they are sold. That means the managers need a good estimate of what the sales level will be several months out. Even if you are not running a retail store, the same problem applies. Manufacturers need to know how many items to produce, what raw materials to order, how many employees to hire, and what size plant will be needed. All of these decisions are based on forecasts of sales.

Sales forecasts are critical, but the sales number is not the only item that needs to be predicted. Businesses also want to know what will happen to interest rates, foreign exchange rates, stock prices, and other financial variables. Investments and borrowing play key roles in managing any firm. Small changes in rates can have huge impacts on profits. Some financial variables can be forecast, others are difficult, but everyone tries to predict what will happen. Similar issues arise in terms of costs, such as supplies of raw materials, wage rates and worker availability, rents, maintenance expenses, advertising costs, and so on.

Time series analysis is also useful for separating out the basic elements present over time: seasonality, trend, and random effects. Many times decision makers want to observe larger trends without being distracted by seasonal changes. The classic example is unemployment data. Because so many students graduate (at all levels) in the spring of each year, raw unemployment rates tend to be high. Yet, when looking at overall trends, these numbers cannot be compared directly with those for fall or winter. Consequently, time series tools are used to estimate and remove the seasonal effects to provide a more accurate estimate of the trend. The same situations can exist in production, sales, inventory levels, birth rates, and physical factors such as temperature.

Many business and manufacturing problems involve time series analysis. Almost any measurement can be treated as a time series, such as sales value, quantity of items sold, employee illness, accident rates, and quality measures. Weight measures taken every week are a classic time series. To minimize depression, you probably do not want to analyze your own weight; but growth rates and variations are critical in agriculture.

Model

What are the components of a time series and how can they be measured? A single time series consists of one variable to be predicted, with observations collected over time. The time interval has to be fixed (such as days, months, or quarters). For the most part, it cannot contain missing observations. Data of this form typically exhibits some internal pattern—sales can increase over time, some months of the year might consistently be better or worse than others. If these patterns can be identified and measured, they can be used to predict future values of the series. The challenge lies in identifying and measuring the various effects. The focus of this section is to describe the common components of time series and define some of the mathematical background to show how they can be estimated. Evaluating individual components is useful not only for prediction, but also because it provides information about the underlying process. For example, managers can form better plans if they have good measures of seasonal effects.

Time Series Components

A time series, particularly a business or economic series, is often defined in terms of four parts: (1) Trend, (2) Cycle, (3) Seasonal variation, and (4) random or irregular fluctuations. Some writers skip the cyclical components or lump them with trend changes. **Trend** represents an underlying pattern over time that is usually somewhat constant or at least independent of other time effects such as seasonality. For example, sales might increase at some base growth rate each quarter. **Seasonality** represents known changes that arise at about the same time every year. Typically they are defined in terms of years, but a process might have quarterly or monthly "seasons." However, measuring seasonal patterns requires detailed data. For example, quarterly or monthly data is needed to see annual seasonal changes. If the season is defined as a shorter time period, it could only be measured with even finer data; and it can be difficult to obtain data at those levels.

Cyclical fluctuations are changes that are influenced by other economic data. Data for a specific business process could be influenced by broader economic measures. For instance, when the economy does well, personal income increases so customers might purchase more items from a specific company. When the overall economy dips into a recession, consumers buy less from this company.

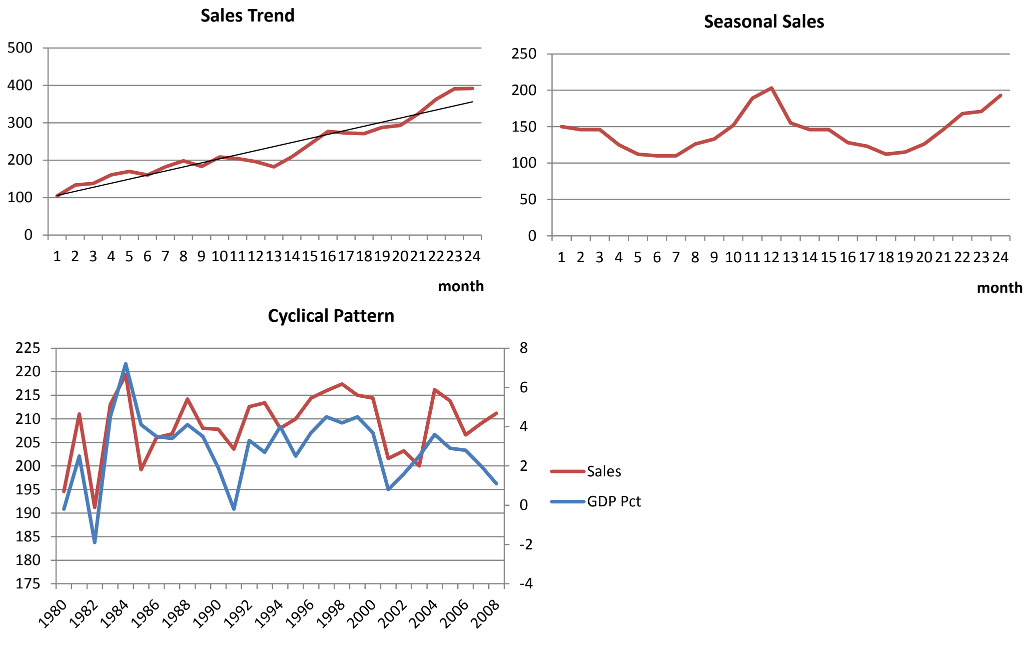


Figure 8.2

Time series components. Trends are long-term movements over time, often evaluated as linear trends. Seasonal patterns usually arise at the same point each year such as end-of-year sales increases. Cyclical patterns represent correlations between the target series and general economic data, particularly ties to recessions and growth stages.

Because the government and other organizations track business cycles, it is possible to identify and measure the timing of these effects. In many ways, cyclical effects are really cross-correlations across time series: The overall economic GDP is a time series and its values are correlated with sales of specific businesses. Cross-correlations are more difficult to estimate than the other time series effects.

Random or irregular changes are movements that have no discernible pattern. Essentially, it consists of fluctuations that exist after the other effects have been removed. In the end, any time series will contain some random effects. Almost no process is completely predictable. The point is to examine the remaining effects and see if they truly are random and to see if they are small relative to the other effects.

Figure 8.2 illustrates the three main components. Trends are long-term movements, usually representing intrinsic growth rates. Most trends are estimated to be linear because long-term nonlinear trends can be risky to forecast. Sometimes a system exhibits a true nonlinear growth (or collapse) rate, and can be estimated with a small-order polynomial or log function. Seasonal data is tied to a regular distance in the calendar. The most common interval is an annual pattern (12 months or 4 quarters). It is recognizable by external knowledge (e.g., sales always increase at the end of the year), and by the fixed number of time periods between peaks and troughs. Cyclical data is the most difficult to identify and measure.

Many books do not count it as a major component because of these difficulties. Technically, it is represented by the correlation between the time series of interest (Sales) and an economic time series that measures the business cycle (GDP or GDP percent change). Notice how the two series move together in the figure. These correlations involving multiple time series are a relatively complex problem and many data mining tools do not handle them. If cyclical measures are critical to a problem, you should seek out sophisticated tools; and probably an expert analyst. Simple versions can be handled with the Microsoft BI tool so they will be considered briefly in this chapter.

The random fluctuations can be seen in the charts as small deviations from the presented patterns. In the examples presented, the random element was deliberately kept small to highlight the trend, seasonal, and cyclical components. Now, imagine the charts with a high random element. At some point, the random element would overwhelm the underlying pattern and it would be difficult to determine if a pattern existed at all or if the series was just random noise.

It is common to write an additive component model of a time series—where the components add up to the total value. At any time period t, the observed value y_t is

$$y_t = \text{Trend}_t + \text{Cyclical}_t + \text{Seasonal}_t + \text{Random}_t$$

Sometimes the model is written as multiplicative, where the plus signs are replaced by multiplication. But, this case can be converted easily to the linear model by taking the logs of the data.

Some methods of estimating time series data attempt to estimate the components directly. If you have measures for cyclical and seasonal data, linear regression is easy to use to estimate the main components. One trick is to simply use time t as a variable, and create dummy variables for seasons with values of one or zero, so you might have a linear form:

$$y = b_0 + b_1 t + b_2\text{Summer} + b_3\text{Autumn} + b_4\text{Winter}$$

Other tricks include estimating seasonal values by computing averages for each month (all values for January, then February, and so on). However, a couple of models now dominate most discussions of time series analysis, and they are easier to automate. To understand the models, you need to understand two fundamental patterns in time series: auto regression and moving averages.

Auto Regression

Auto regression is based on the concept that the next value in a time series is likely to be correlated with the current value. Think of it as momentum—most data make relatively smooth changes from one period to the next. Even if something radical changes, it takes a few periods for everything to be impacted. Yes, some systems are more **chaotic** and it is possible that small changes in underlying factors will lead to radical changes in the output variable. However, auto regression is a useful feature to estimate, and many systems will exhibit some of its features.

In mathematical terms, auto regression can be written so that the value at time t can be influenced by any of several prior points in time, from 1 to p periods back:

$$Y_t = a_0 + a_1 Y_{t-1} + a_2 Y_{t-2} + \ldots + a_p Y_{t-p} + \varepsilon_t$$

A separate coefficient (a) is measured for each lag. The error term (ε_t) represents a simple random error with zero mean and constant variance. The model is often abbreviated to the form: AR(p), such as AR(1) for a single-period lag, or

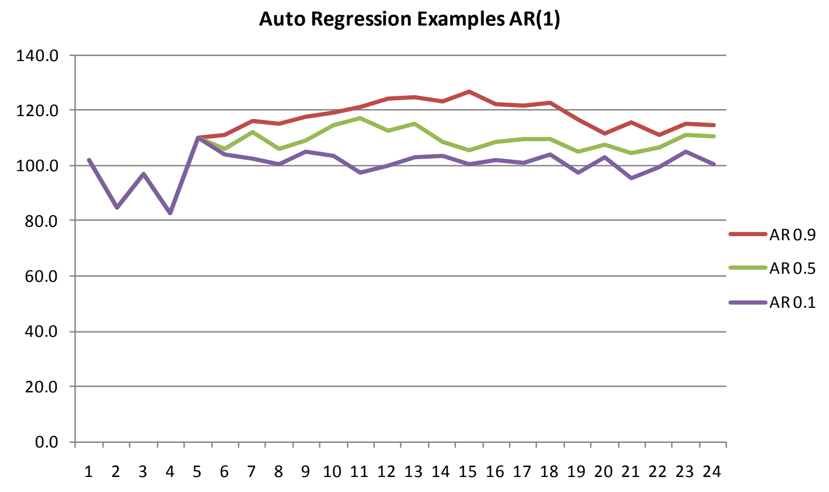
Time	AR 0.9	AR 0.5	AR 0.1
1	102.0	102.0	102.0
2	85.0	85.0	85.0
3	97.0	97.0	97.0
4	83.0	83.0	83.0
5	110.0	110.0	110.0
6	111.0	106.0	104.0
7	115.9	112.0	102.4
8	115.3	106.0	100.2
9	117.8	109.0	105.0
10	119.0	114.5	103.5
11	121.1	117.3	97.4
12	124.0	112.6	99.7

Figure 8.3

Sample auto regressive data generated with AR(1) set at three values (0.9, 0.5, 0.1). The mean was adjusted for each case to keep values within similar ranges.

Figure 8.4

Sample auto regressive data generated with AR(1) set at three values (0.9, 0.5, 0.1). Notice that the series with AR(1)=0.9 is smoother than the other two with smaller AR(1) coefficients. The high-proportion of value carried over from the prior time leads to slower changes in the data.



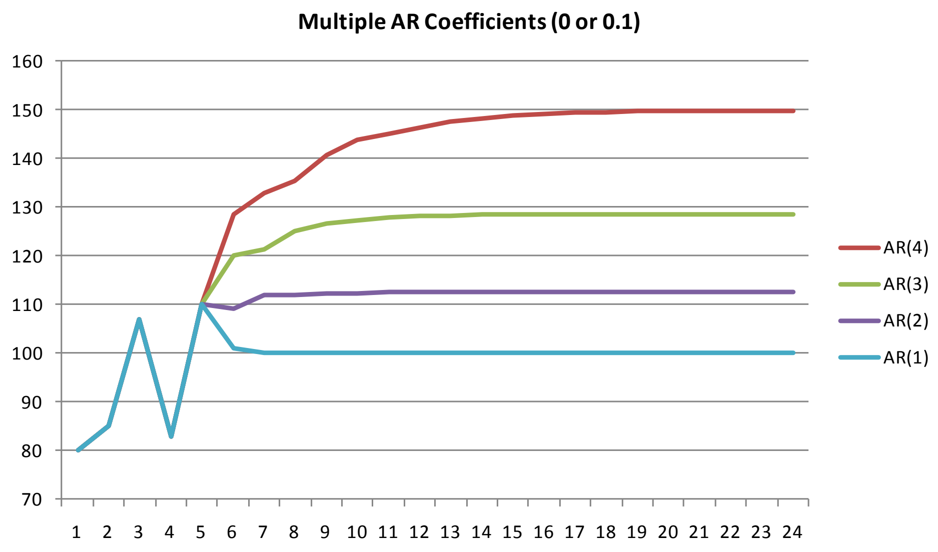


Figure 8.5

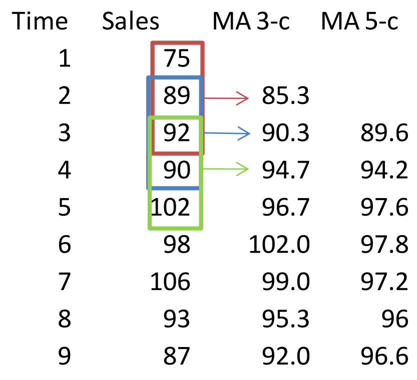
Sample auto regressive data generated with AR(1) to AR(4) coefficients set to 0.1. Without additional random elements all versions level out to a stationary state. The higher-order models pick up a more of the initial variation.

AR(3) to represent lag effects for the first three periods. Mathematically, the AR equation is a difference equation and some coefficient values will make it unstable. For example, the coefficient in an AR(1) model must be between -1 and 1, otherwise the model will predict an "explosion" where each subsequent value will increase rapidly, either positive or negative.

To understand the concept of auto regression, it helps to examine a few artificial examples. Figure 8.3 presents sample time series that were generated using three different values for AR(1). The mean coefficient was adjusted each time to hold the results to similar ranges. Random elements were added to each point. Note that the first five terms are the same in each series—to provide the same foundation for each series. Data were generated for 24 periods, but only the first few are shown in the table.

The effect of the parameters is difficult to see in the raw data. Figure 8.4 shows the charts of the three series. The most important result is that when AR(1)=0.9, which is a high value, the chart is smoother. The data shows less variation than in the other cases. The reason illustrates the role of the AR coefficient. A high value means that a large portion of any forecast consists of the prior value. Carrying such a high percentage of value each time means that small changes are smoothed out. Instead, the curve adopts longer, slower changes.

Figure 8.5 takes a slightly different approach. In this example, all of the lag coefficients have the same value or they are zero. The difference is the number of lags included in the model. AR(1) has one non-zero lag (0.1), and AR(4) has the first four lag coefficients set to 0.1. That is, AR(4) picks up effects from the four prior time periods. Notice that all four models eventually level off to a stationary state—because no additional random data was added and the model is stable.



Time	Sales	MA 3-c	MA 5-c
1	75		
2	89	85.3	
3	92	90.3	89.6
4	90	94.7	94.2
5	102	96.7	97.6
6	98	102.0	97.8
7	106	99.0	97.2
8	93	95.3	96
9	87	92.0	96.6

Figure 8.6

Simple moving averages. A three- and five-period average centered to keep the series aligned.

Also, the same mean was used in all four models, so the models with more lags will have higher average values because of the added terms. This decision was made to keep the lines separate so the patterns would be more visible.

In terms of differences, all series have time points 1 – 5 in common. Time 6 is the first predicted value. The AR(1) model drops simply because the mean stays the same and the overall values are lower. There is no variation in the line due to earlier lags. AR(2) picks up the drop at t-2 (time 4) and it picks up the gain at t-1 then levels off. AR(3) picks up those two effects plus the peak at lag t-3. AR(4) also picks up the peak, but it is mitigated by the additional drops at lag t-4. The point of the charts is that if a time series has a seasonal effect at some periodicity that lag value needs to be included in the estimation. For instance, if the data is collected at quarterly intervals and the model estimates only an AR(1) lag, then any annual changes present at AR(4) will not show up in the prediction. So a key aspect of using auto regression is identifying the specific lags that should be included in the model. It might be tempting to include all possible lags up to a year, but it takes enough data observations to estimate all of the values. Plus, many times the in-between lags are not interesting, so models are often estimated with the first two or three lags, plus a seasonal lag; skipping the ones in the middle.

Moving Average

In some ways a **moving average** is relatively easy to understand, in others, it is complex. By its basic definition, a moving average is used to smooth data. Figure 8.6 shows the calculations for a three-period and five-period moving average. Both computations are centered to keep the series aligned; otherwise the moving average series will be offset and difficult to compare to the original. As you can tell from the name, a three-period moving average computes the average of three consecutive values. Shifting up a time period, the system computes the average of the next three items shown in each box. A five-period average uses five consecutive entries. The width of the average (3 or 5) is somewhat arbitrary. If you stretch the interval to cover an entire year; such as 4 points for quarterly data or 12 for monthly data; the average will eliminate all seasonal variations.

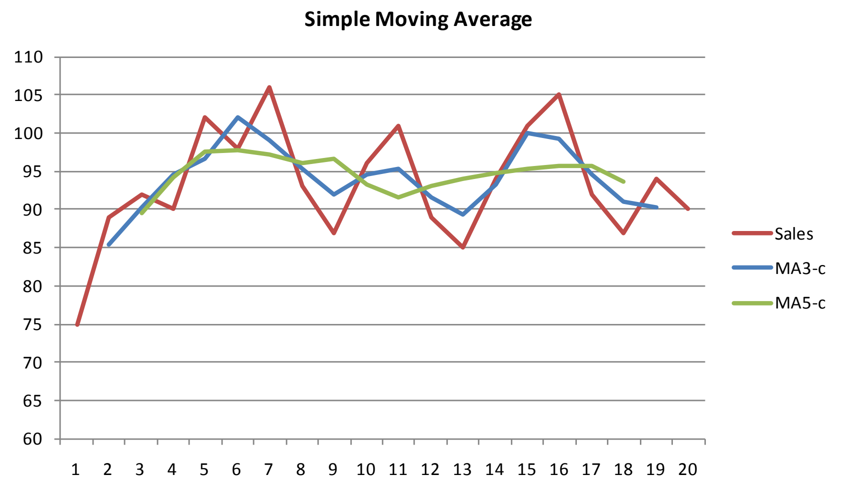


Figure 8.7

Simple moving averages. The moving average reduces the variations, and wider averages (5 versus 3) create smoother results.

The results are easiest to see in a chart. Figure 8.7 shows the original data and the two moving averages. Clearly the two averages reduce the variability in the overall series. The five-period average is also smoother than the three-period values. The technique can be used to reduce noise due to random events. It can also be used to average out seasonal effects to highlight underlying trends or other patterns.

The term moving average derives from this common method of calculating averages to reduce or eliminate seasonal effects. However, you will rarely actually perform this computation. Instead, the underlying mathematical model is more important. The moving average process defines the outcome variable (Y) at a point in time as a mean plus a weighted average of the differences from that mean in prior periods:

$$Y_t = \mu + \psi_0\varepsilon_t + \psi_1\varepsilon_{t-1} + \psi_2\varepsilon_{t-2} + \psi_3\varepsilon_{t-3} \ldots$$

Figure 8.8

Simple moving averages. Consider a simple moving average model that has four independent seasonal effects (quarters). Computing the four-period moving average yields the mean plus an average of the seasonal effects. If the seasonal effects net to zero, the moving average yields just the overall average.

$$\begin{aligned} Y1 &= \mu + S1 \\ Y2 &= \mu \quad + S2 \\ Y3 &= \mu \qquad + S3 \\ Y4 &= \mu \qquad\quad + S4 \end{aligned}$$

Compute the average: $(Y1 + Y2 + Y3 + Y4)/4 = \mu + \text{Avg}(Si)$

		S1	S2	S3	S4	S1	S2	S3
Yt	μ	+ψ0εt	+ψ1εt-1	+ψ2εt-2	+ψ3εt-3			
Yt-1	μ		+ψ0εt-1	+ψ1εt-2	+ψ2εt-3	+α3εt-4		
Yt-2	μ			+ψ0εt-2	+ψ1εt-3	+ψ2εt-4	+ψ3εt-5	
Yt-3	μ				+ψ0εt-3	+ψ1εt-4	+ψ2εt-5	+ψ3εt-6
Avg	μ + Wtd. Avg (S1) + Wtd. Avg(S2) + Wtd. Avg(S3) + Wtd. Avg(S4)							

Figure 8.9

Quarterly moving average. In the more general model, the moving average over all four seasons results in the overall mean plus weighted averages of each of the four seasonal effects. If the seasonal effects are neutral, and the weights meet some identity conditions, the weighted sums will effectively average out to zero.

Conceptually, the errors or deviations (ε) could be considered as seasonal effects, and pure random error. Figure 8.8 shows a simplified model that contains independent seasonal effects for four seasons or quarters. Computing the four-period average yields the underlying mean plus an average of the seasonal effects. If the seasonal effects are neutral in total, their average will be zero and the moving average yields just the overall mean. That is, a moving average that covers the entire season should eliminate the seasonal effects.

The general model for the moving average leads to more complex results, but the concepts are similar. Figure 8.9 shows a quarterly model with a four-period moving average. If the seasonal effects are neutral—gains in one season are offset by declines in other seasons—and the weights meet some identity constraints that are beyond this book, then the weighted sums effectively average out to zero when the moving average spans the entire year. So, again, an average across the entire year yields an estimate of the overall mean. The situation is more interesting and more valuable when the moving average spans less than a full year. For example, a problem might compute a three-month average. In this case, the moving average model shows that the predicted value will consist of the mean plus weighted errors (e.g., seasonal effects) from prior months. That is, the moving average model transfers information forward from the errors or seasonal effects. Remember this interpretation of the model. It will be needed in the ARIMA section that combines the models for auto regression and moving averages.

Charts can help show the effect of the moving average coefficients and the different lengths of the average. Figure 8.10 shows three series created with the general moving average model using only a one-period lag on the error term. The coefficient weight of the lag was tested with three different values (0.1, 0.5, and 0.9). The series were started with relatively high variation in the first five periods. Notice that all three series eventually die down to the mean with just random errors. Also, notice that the larger coefficients (particularly 0.9) carry the initial variation for a longer time. This result is clear from the equation because the next-period value is computed by adding the multiple of the coefficient and the prior error.

Figure 8.11 examines moving averages with more lags—from one to four time periods. All of the coefficients were set to the same 0.5 value. All series had the same starting values with relatively high variation. No random effects were add-

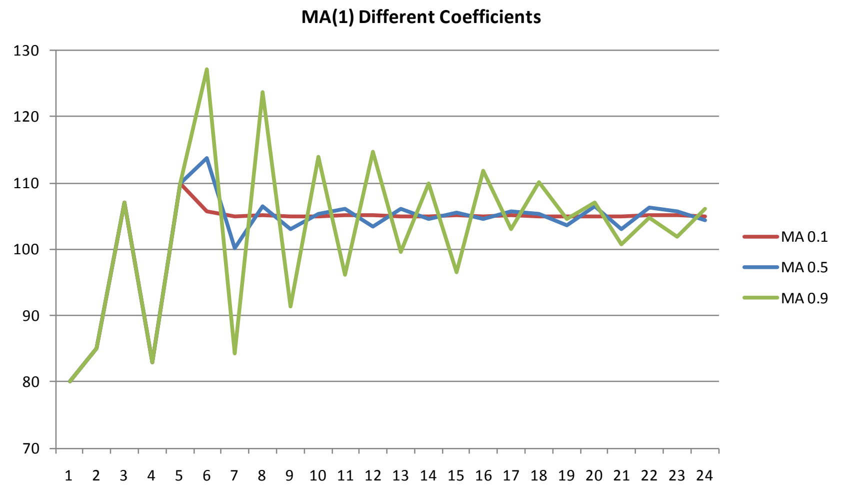


Figure 8.10

One-lag moving average with different coefficients (0.1, 0.5, 0.9). All three series eventually die down, but the higher values (0.9) carries the variation for a longer time.

ed beyond the starting values. The series that includes four lag effects is clearly more variable for a longer period of time than the single lag. The point is that using more lags in the model can capture seasonal effects for more time periods. These effects are carried into future forecast values. Eventually, you will face the question of how many lag periods should be included in a real-world model. The answer is often difficult. Using more lags can lead to a more realistic model—if the seasonal effects truly cover more time periods. Using fewer lags results in a smoother model—useful if the lagged variations are random errors instead of seasonal effects. Diagnostic tools can help you determine if variations are random or part of a seasonal pattern, but it helps if you understand the fundamental business data being examined.

Trends

As an analyst or business person, trends in time series data are important. An overall increase in sales would indicate that the average sales level is increasing each year. Likewise trends in stock prices or interest rates would signal long-term changes, which would give the wise investor the opportunity to make money. Trends can be linear or nonlinear. Linear trends are relatively easy to estimate using linear regression. Nonlinear trends are trickier because you might have to identify the degree of nonlinearity. Typical polynomial equations and log-linear equations can also be estimated with standard computations and regression techniques. Techniques for using regression to estimate trends are covered in the linear regression section of this chapter. As a hint, a simple approach is to include time (t) as a variable, where time can be a simple numerical sequence:

$$Y = b_0 + b_1 t + b_2 \text{ seasons} + b_3 \text{ others}$$

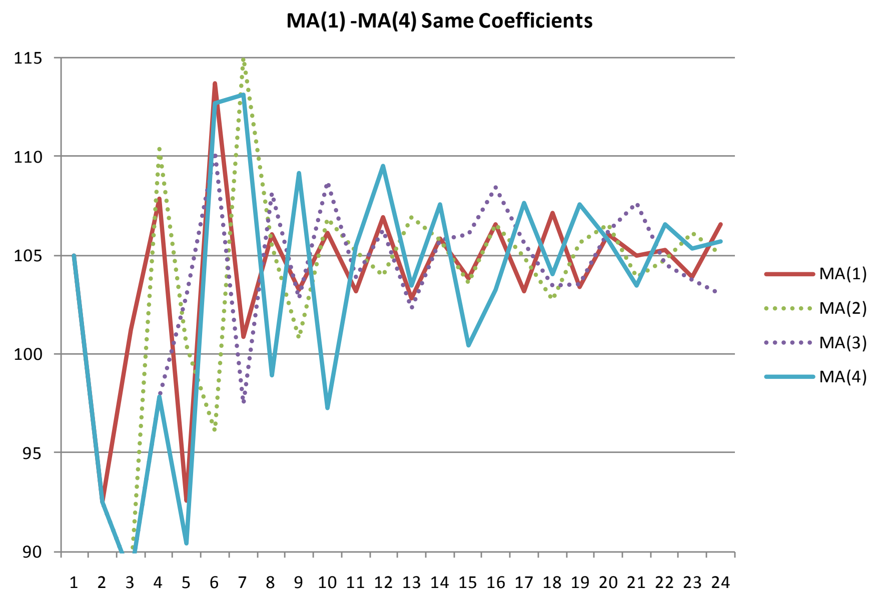


Figure 8.11

Moving average with different number of lags. All coefficients set to 0.5 and no other random error was added. The longer-term averages are more variable because they pick up and extend more of the initial variation.

Other techniques are also possible, but it is generally best to stay relatively close to linear or cubic trends when forecasting time series data. Highly nonlinear patterns, particularly exponential growth, rarely exist for long periods of time. A small company might observe exponential growth for a year or two—because the initial base is so small. But, predicting an exponential growth beyond a couple of months is rarely going to work out. It is almost always better to be conservative and forecast a linear trend, even if it seems less accurate over historical data. Figure 8.12 shows the basic problem. The initial data follow an exponential growth rate, but there just is not enough data to be comfortable with an exponential forecast. Using an exponential trend, sales would leap from 56,000 in period 8 to over 310,000 only two periods later. Sure, it could happen, but without advanced knowledge of the industry and testing, such a forecast would be risky. At the same time, a simple linear trend using all of the initial data is likely to be too low—notice that the linear trend forecasts an initial drop in sales. A polynomial (cubic) equation might be a good fit, but it is going to forecast a relatively large increase in sales as well. An alternative is to use the last few periods to generate a linear forecast from that point. This decision and the selection of the starting point would have to be made by a human, not an automated algorithm; because it requires knowledge of the industry and examination of similar patterns from other organizations if they exist.

The interesting problem at this point is that in pure time series analysis, trends cause problems in the estimation of the seasonal and random data. The ARIMA technique described in the next section only works if the data contains no trend. Fortunately, that statement does not mean ARIMA is rarely useful. Instead, it

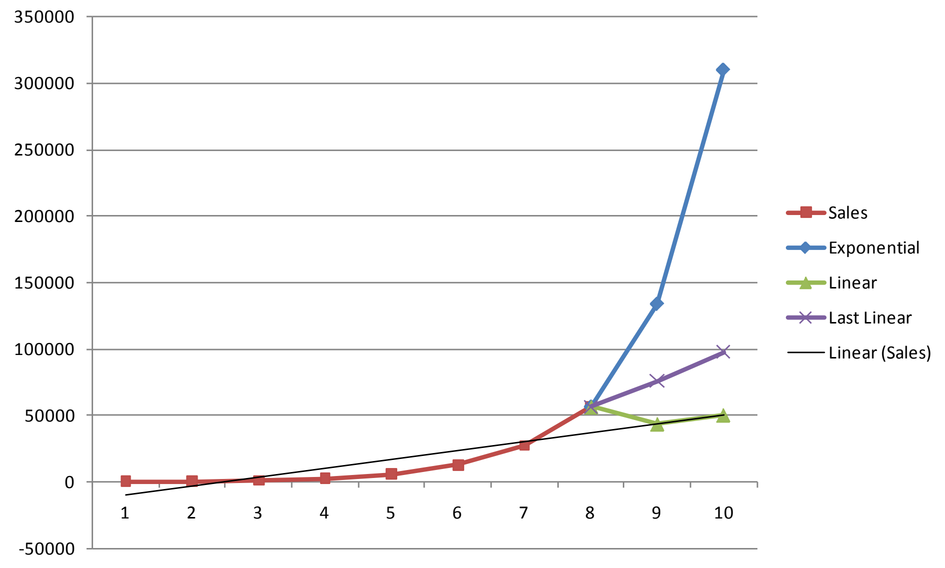


Figure 8.12

Dangers of nonlinear forecasts. An exponential trend would predict a five-fold increase in only two periods. Linear using all data is probably too conservative. Linear using the last few observations is probably a safer forecast.

means that you have to remove the trend before applying the ARIMA analysis. Most tools have an easy method to accomplish this task. If you see a trend in the data—usually by plotting it—the tools can analyze the difference instead of the raw numbers. So, if you see a linear trend, you can specify a single difference step to compute: $z_t = y_t - y_{t-1}$, and then base the analysis on the z series instead of the original. If the pattern is nonlinear, you might need to take a second difference. In extreme cases, you can convert the data using logarithms: $z_t = \log(y_t)$, and possibly difference those values if you still see a trend. The details of the differencing process are explained in the ARIMA section. The point is that you need to recognize and often test for trends in the data before trying to estimate seasonal component.

ARIMA

Auto regressive integrated moving average (ARIMA). The name makes it seem like a difficult topic, but the name also clearly defines what the tool does: combine auto regression and moving average time series estimation into one package. Besides, everyone just uses the initials ARIMA. The methodology is sometimes called Box-Jenkins after the two authors who developed the process for analyzing time series data. The clearest way to understand the combined approach is to write the mathematical model:

$$y_t = \theta_1 y_{t-1} + \theta_2 y_{t-2} + \ldots + \theta_p y_{t-p} + \varepsilon_t + \psi_1 \varepsilon_{t-1} + \psi_2 \varepsilon_{t-2} + \ldots + \psi_q \varepsilon_{t-q}$$

The equation states that a series value y at time t is generated from two parts. An auto-regressive component that pulls information from prior y values, and a moving average component that pulls information from prior error terms—which could represent seasonal variations. This version of the model is usually abbreviated ARMA(p,q) where p represents the highest lag for the auto-regressive terms and q the highest lag for the moving average terms. Note that the "I" is missing because it is not included in this model and will be explained shortly. The interpretation of the equation is straightforward. Each value in a time series is affected by the values that came before it and the variations from the average are also influenced by the variations in prior periods. The goal of ARMA is to estimate the values of the coefficients to determine the weight that each prior value plays in determining future outcomes. That is, the coefficients define the time series pattern.

One important catch with estimating the ARMA equation is that it can be estimated only if there is no trend in the data. In mathematical terms, the time series must be stationary—where the series revolves around a fixed mean. The concepts of stationary and identifiability are covered in advanced textbooks on time series. Most tools automatically notify you if the conditions are violated, so you do not need to be able to verify them manually. Still, you do need to know that if a trend exists in the data, it needs to be removed. The most common solution is to define a new series $z_t = y_t - y_{t-1}$. If the trend is linear, this simple differencing usually eliminates the trend and the coefficients can be estimated. If the trend is nonlinear, you might need a second differencing—where the z values are subtracted. In extreme cases, you might need to convert the original y values by defining a new log(y) variable—which often needs a single differencing to be usable. Because differencing is the most common approach, it defines the integrative aspect of the model: ARIMA(p,d,q) is defined with p the number of auto-regressive lags, d as the number of differences, and q as the number of moving average lags.

Differencing to Remove Trend

Most traditional ARIMA tools require supervision and the analyst must specify the p, d, and q values. So, how do you know which values to choose? The d value is usually the easiest because it is almost always 0, 1, or 2. For a quick approach, simply plot the time series and see if a trend exists. If there is no apparent trend up or down, it will not be necessary to difference the data, so d equals 0. If there appears to be a linear trend, choose d equal to 1. If the trend appears nonlinear, perhaps as in a cycle, set d equal to 2. Avoid jumping automatically to the highest value. If you did not choose enough differencing and a strong trend remains in the data, it is unlikely that the ARMA equation can be estimated and the tool will generate an error. So increase the value of d and try again.

The lag limits p and q are trickier. Some tools use statistical methods to help estimate the appropriate values automatically. Others follow the Box-Jenkins methodology and ask you to examine two charts to help determine the appropriate values. An additional complication exists. In the equation, p and q are the highest values of the lags, and all intermediate lags are estimated as well. Often, the time series will not have enough data to estimate every possible lag value. However, you still might need some higher-order lags. For example, some time series can be estimated with p and q set to 2 or 3. However, many time series also exhibit seasonal patterns—particularly annual events such as holiday sales. Using monthly data, these patterns would arise at lag values set to 12. Because the values at lag periods 4-11 are likely to be minor, you typically do not want to estimate those lags. Consequently, most ARIMA tools allow the analyst to select potential lags by entering specific numbers, such as 1, 2, 3, 12; skipping the intermediate values that are unlikely to be relevant. Of course, the analyst needs to know or guess which values might be important.

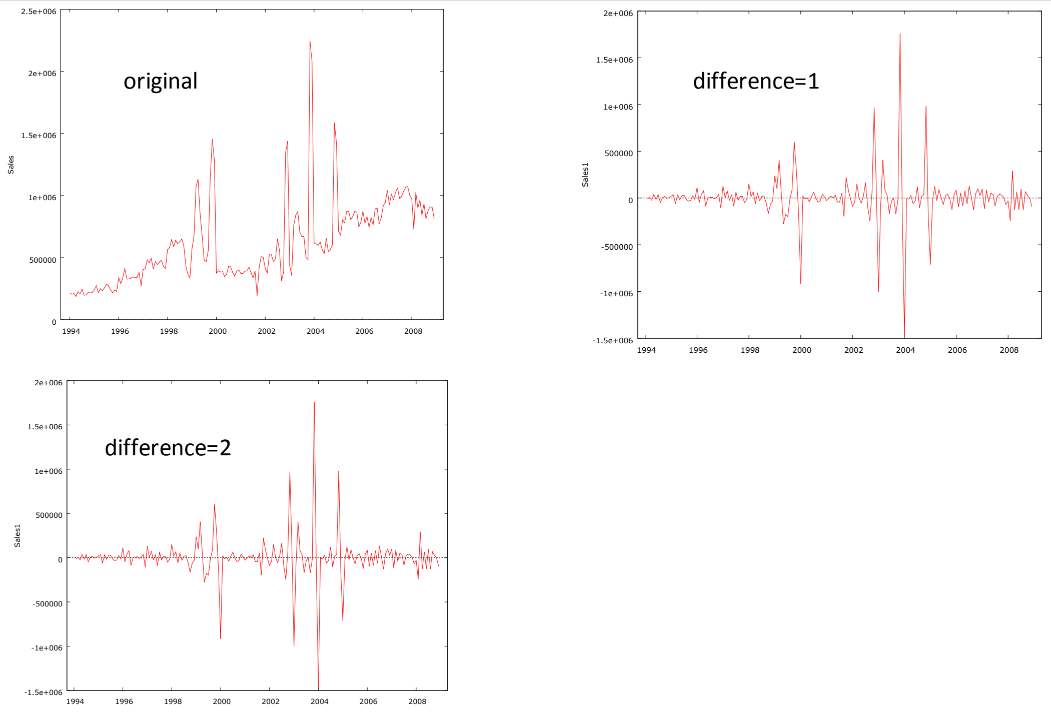


Figure 8.13

Differencing series to remove trend. The original series appears to have a linear trend. One difference has removed the trend, leaving a flat line with variations. A second difference operation does not appear to improve the pattern.

Figure 8.13 shows the time series of total sales by month for Rolling Thunder Bicycles. The original data plotted over time clearly shows a trend that needs to be removed. A series that computes the first difference appears to remove the entire trend, leaving variations along a horizontal line. There might be a slight trend remaining at the right-end of the chart. As comparison, a second difference is also plotted, and it appears to be the same as the single difference. Hence, a single difference should be sufficient

Autocorrelation and Partial Autocorrelation

The Box-Jenkins approach emphasizes the importance of the **autocorrelation function (ACF)** and the **partial autocorrelation function (PACF)**. These two functions are used to evaluate whether the model will be solvable and to help identify the appropriate values for p and q. Autocorrelation is the correlation between any two time series observations separated by a lag of k. The function is computed and plotted for several (perhaps 20) values of k. The correlation is computed as the covariance between all points separated by k time periods, and divided by the variance of y to obtain a value between -1 and +1. The values are estimated from the formula where is the mean of the y values:

$$r_k = \frac{\sum_t^{n-k}\left(y_t - \bar{y}\right)\left(y_{t+k} - \bar{y}\right)}{\sum_t^{n}\left(y_t - \bar{y}\right)^2}$$

The partial autocorrelation is loosely the correlation between two points separated by k time units without the effects of any intermediate observations. It is computed for k>1 to be:

$$r_{kk} = \frac{r_k - \sum_{j=1}^{k-1} r_{k-1,j}\, r_{k-j}}{1 - \sum_{j-1}^{k-1} r_{k-1,j}\, r_j}$$

Where $r_{kj} = r_{k-1,j} - r_{kk}\, r_{k-1,\, k-j}$ and r_k is the sample autocorrelation at lag k. Fortunately, tools exist to compute and plot both functions with respect to the lag values. The charts are called correlograms.

Figure 8.14 shows a sample ACF and PACF from a hypothetical time series. First, observe that the ACF dies down relatively quickly, while the PACF simply cuts off after 2 or 3 lags. Some systems will display confidence interval bars to help determine if the correlations at each lag are significantly different from zero. To understand the charts, first note that both functions either die down or cut off. If one of the two (or both) failed to die down, it would indicate that the series was not stationary and still had a trend that needed to be eliminated through differencing.

The most important aspect of the ACF and PACF is that they provide an indication of the lag values to be tested for the AR and MA components of the model. The key is to look for the point where either the ACF or PACF cuts off. In this example, the PACF cuts off at lag 3, with lag 2 significant, while the ACF dies down. This pattern (AR die down, PACF cut-off) is representative of an AR(2) series, so the ARIMA model could be run with p=2, d=0, and q=0. If the pattern were reversed where the PACF died down and the ACF cut off, it would be an indication of an MA(2) model, or ARIMA(0,0,2).

In many cases, the PACF or ACF will have additional spikes at later lag points. In particular, it is common to see a spike at lag 12 in monthly data to indicate an annual seasonal pattern. If these spikes are widely scattered, they can usually be specified individually in the ARIMA definition, such as p= 1, 2, 3, 12. If both the ACF and PACF exhibit cutoffs, it is likely that the model will need lags for both the AR and MA components, such as ARIMA(3, 0, 3).

The initial estimate of the AR and MA lags does not have to be perfect. You just need decent estimates to provide a starting point for the estimation. Even with tools, such as Microsoft Time Series, the estimation process works best if you can provide a reasonable starting point. The estimation results will provide additional information to evaluate the significance of each lag coefficient.

Cross Correlations

The ARIMA process is probably the most common method used to forecast time series. In addition to its long-term forecast capabilities, it provides measures of the lag and seasonal effects which provide useful information to explain the underlying patterns. However, ARIMA is designed to work only with a single series of data at a time. It estimates purely internal patterns within that series. In some business problems, particularly in finance, it is clear that changes in one series will affect the outcome of the series being investigated. The business cycle or general economy provides a classic example. If the overall economy is doing poorly, people have less income, more people are out of work, and they are less likely to buy expensive products. To analyze business sales, it might be useful to match the sales pattern to the economic pattern. These **cross correlation** models are useful

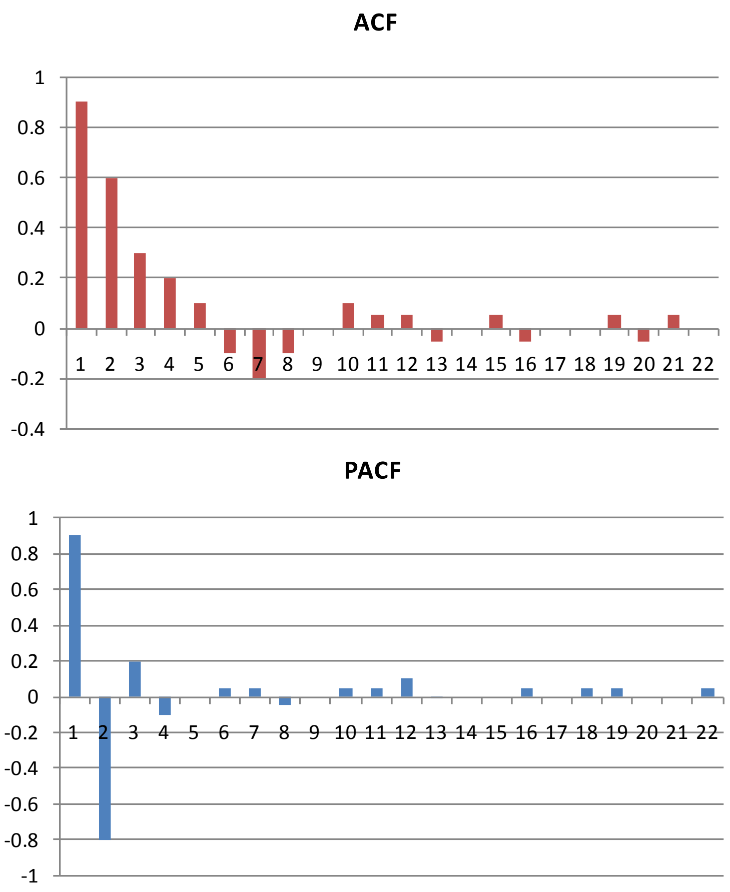


Figure 8.14

Auto correlation and partial autocorrelation functions. Both die down or cut off so the hypothetical series is stationary. The ACF dies down and the PACF cuts off after 2 or 3 lags. This pattern implies an AR(2) model with no MA terms. If the patterns were reversed, it would indicate an MA(2) model.

when a model exists that explains how some attributes affect the predictable series. Economic examples are common, and the technique is often used in financial forecasts. It can also be useful when physical or biological relationships exist. Essentially, cross-correlation combines traditional economic modeling with time series patterns. In fact, regression tools are probably the easiest method for evaluating combined cross-correlation and time series models. Regression can estimate the coefficients on the independent attributes as well as the values of the autoregressive and moving average terms.

Figure 8.15 shows a hypothetical example of Sales inversely correlated with the unemployment rate. By economic theory, sales at most firms will be correlated with the economy—as a measure of consumer income. It might be easier to understand the relationship if real GDP or income were used instead of the unemployment rate. However, the U.S. Bureau of Economic Activity (BEA) only releases quarterly and annual data for GDP accounts. Unemployment, inflation, and interest rates are available monthly which provides more data for better analyses.

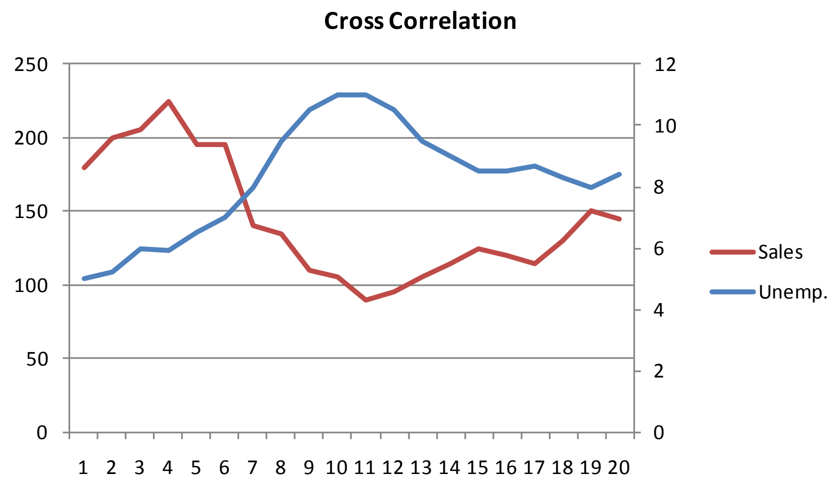


Figure 8.15

Cross correlation. Here, sales are inversely correlated with the unemployment rate, which is measured on the secondary y-axis. Adding an economic series to help forecast the primary series can boost the performance of the model. However, it requires that you obtain forecasts of the economic series to make predictions about the sales data.

One additional warning about government data is critical when working with time series. Many government time series are provided as **seasonally adjusted**. Seasonally adjusted data has been averaged to remove seasonal patterns. Fortunately, the online Web-based tools make it relatively easy today to obtain either the raw data or the seasonally adjusted series. You simply have to set a check box, but you must decide which version of data you want to use for each specific situation. If you choose the raw data, then the estimation will attempt to apply any seasonal patterns in the reference series to your data. In the example with the unemployment rate, it might be best to use seasonally adjusted data. Unemployment rates are strongly affected by school graduations and those seasons might not match typical sales. If the problem is using unemployment as a proxy for the overall state of the economy and consumer income, the seasonally adjusted values would be a better economic measure. If the problem truly needs to rely on the number of unemployed workers available, then the raw data would be better. The point is that the answer depends on the specific problem being studied and the analyst needs to make the decision to match the problem.

A second type of problem often crops up in business forecasting where cross correlation is useful. Remember that time series forecasts require a relatively long history of data—preferably at least 50-100 observations. So how do you forecast a series for a new product or a new business region? Will you have to wait four years to gather enough monthly data to make a forecast? The solution is to assume that the new product or new region will follow a pattern similar to an existing product or region and use the cross-correlation as the basis to forecast sales for the new region.

One of the drawbacks to building a cross-correlation model is that to predict the dependent variable (e.g., Sales), it is also necessary to predict the series for each of the independent variables. So, instead of making one prediction, you must first predict every other series that is influencing the outcome. If the independent series consist of common economic data, it is usually possible to find at least short-term predictions made by several organizations that track general economic activity. If the independent series consist of internal data, the series must be predicted using separate models or separate time series analysis of each series. Before embarking on building a cross-correlation model, first decide if it will be easier or harder to forecast the other series that will be used in the model. Trading one problem for a harder one is going to cause problems.

Models that use extensive numbers of cross series will probably benefit by being estimated with least squares regression. Other tools, such as Microsoft Time Series, include the ability to add external attributes to the forecast method. Microsoft BI embeds these attributes into a decision tree solution and provides different estimates of the ARIMA model for each leaf on the tree. Details are explained in the tools sections.

Evaluating Models

It should be clear by now that time series models are difficult to automate. Most tools require supervision by the analyst. Ultimately, the analyst must choose among several model variations. In the ARIMA model, the selection of p and q ultimately requires judgment. Is there a way to evaluate the resulting models?

Several overall measures have been defined for evaluating ARIMA models. Box-Jenkins proposed the Q-statistic which measures the remaining autocorrelation of the residuals after the model has been fit:

$$Q = (n-d)\sum_{i=1}^{K} r_i^2(\grave{o})$$

The value for K is somewhat arbitrary—it is chosen to see if the first K autocorrelations together account for too much variation. K is often chosen to be 12, or perhaps 24 or 36 for large models. The number of observations is n, the number of differences taken is d. The r^2 terms are the square of the autocorrelations in the residuals separate by the specified lag (i). Q follows a Chi-square distribution with $K\text{-}n_p$ degrees of freedom, where n_p is the number of parameters estimated in the model. If Q is too large the model is rejected on the grounds that too much autocorrelation remains within the residuals.

Another common measure is a variation of the RMSE:

$$s = \sqrt{\frac{\sum_{i=1}^{n}(y_t - \hat{y}_t)^2}{n - n_p}}$$

Models with smaller error values (s) are better. Check this value when comparing models and pick the model that has a substantially lower value of s. Most tools report this measure, the Chi-square statistic might not be reported automatically. Other writers have proposed numbers that have similar properties in general. For instance, the **Akaike information criterion (AIC)** becomes smaller as the AR variance decreases; so smaller values are better. Schwarz proposes a similar measure (**Schwarz criterion**) that is sometimes reported. The Schwarz measure is sometimes known as the Bayesian Information Criterion (BIC) because of the methodology he used to derive the measure. Most tools report at least some of these measures so the analyst can compare models to see which one generates the smallest errors. Judge [1985] provides the derivation and interpretations of these measures, but as long as they are computed by the software tool, the values can be compared across different models without worrying about the differences in interpretation.

The main challenge with ARIMA is to get the correct selection of AR and MA lags (p and q). Each different collection represents a different model. Whenever a model is evaluated, care must be taken to ensure that the model actually makes sense. Are there significant coefficients? Do the autoregressive and seasonal terms fit the known facts? Look at the residual plots. Do substantial trends or correlations remain in the residuals? Try a forecast to see if the model values match the actual patterns.

Data

What types of data can be used in time series analysis? The short answer is that a single attribute is evaluated at a defined time interval. For example, sales totals are computed for each day, week, month, or quarter. The key is that there can be only one attribute and the time interval has to be constant. Some tools support cross correlations where one series can affect a second series. In these cases, the series must contain exactly the same time observations—the same interval, and the same starting and ending points. The data itself should be continuous data. Typically it is a subtotal such as sales value or quantity. Almost any measurement data will work, including temperature, distance, and weights.

The time interval must remain constant for the entire series, and there must be enough observations to estimate several parameters. Common recommendations are at least 50-100 data points. The catch is that it takes time to record the observations. Fifty points is four-years of data collected monthly. It might be tempting to use more detailed data—weekly or even daily—just to collect enough data points. The drawback to detailed data is that it can be harder to interpret the results. Daily data might work if the business exhibits patterns during the week, such as low sales on Mondays and high sales on weekends. But, the daily values will not provide information about seasonal patterns (such as July versus December). The key is to decide what types of patterns are likely to be important and use that choice to select the time interval.

Attributes and Observations

Only a single attribute will be used for most analyses. The data is stored with the time values in rows. Traditional tools work well with just the single column of data for the series. Microsoft Time Series requires a second column to provide an identifier or key value. Each observation row must be assigned a unique key value. For time series data, this value is typically either a simple integer or it is a combination value for the date interval. Microsoft tools only use the value to sort the data and display cutoff points, so the intervals do not have to be continuous.

Most time series data is aggregated, and the values can be computed using a GROUP BY query or an OLAP cube. The requirement in Microsoft BI of using a single key column often requires some work to define a column that concatenates date parts. For example, assume the database contains a SaleDate column that lists sales by day. Analyzing the data by month will require the creation of a column that combines year and month and then computing the subtotal by that column. The catch with Microsoft BI is that the time identifier column should be numeric. However, that objective is somewhat easier to achieve than it first sounds. The simplest way to create the time key column is to define a new column:

```
Year(SaleDate)*100+Month(SaleDate) As YearMonth
```

The Year and Month functions return numeric values. Multiplying the year by 100 shifts it two places to the left to leave room for a two-digit month. The resulting column contains values such as 200801, 200802, 200901, and so on. These values are sorted correctly by year first and then month. However, note that values cannot be subtracted from each other to obtain a time distance. Subtraction will work within a given year (200803 - 200801 = 2 months from January to March), but not across years (200901-200812 is way more than one month). Because Microsoft BI uses the column only for sorting, subtraction is not an issue; just remember not to use the column for any other purpose.

Microsoft BI can use other attributes to determine how they influence the predictable variable. To include other attributes in the problem, they will have to be treated as additional columns in the query. Most importantly, the time identifiers (rows) must exactly match those of the predictable series. If the data comes from other internal tables, it can usually be computed as subtotals the same as the primary series. If the data is retrieved from external (e.g., government) sources, it most likely will need to be assigned the same time key value used for the primary series. At that point, a JOIN query can match the rows correctly between the imported comparison series and the predictable series.

Missing Data

A time series should not contain missing observations. For instance, with monthly data, it cannot skip some months. If a series has many missing points, it is better to aggregate up to a wider time interval. For example, if daily data has many holes, aggregate and perform the analysis with weekly or monthly data. If a series has a small number of missing values, they can be imputed through a variety of methods. The most common are: (1) average the two nearest values, (2) copy the nearest older neighboring value into the missing value, (3) compute an average of all data for that time slot (e.g., quarter 1 or specific month). All of the methods have some drawbacks, which is why series with many missing values should not be used. Microsoft BI provides options similar to these to automatically replace missing observations. However, a query should be used to count the number of missing values first to decide if the number is small enough to reduce the impact on the results.

Traditional ARIMA Estimation

What results does traditional ARIMA provide? The model section of this chapter provides a detailed description of how ARIMA works. Most tools that implement the traditional version follow the Box-Jenkins process. Microsoft Time Series has a version of ARIMA but it is mixed with a proprietary algorithm so it is a different process described in the next section. The traditional Box-Jenkins process essentially provides tools to plot the data, create ACF and PACF charts, and estimate the ARIMA model according to lags and differencing specified by the analyst. The analyst needs to organize the data into a series, check the model

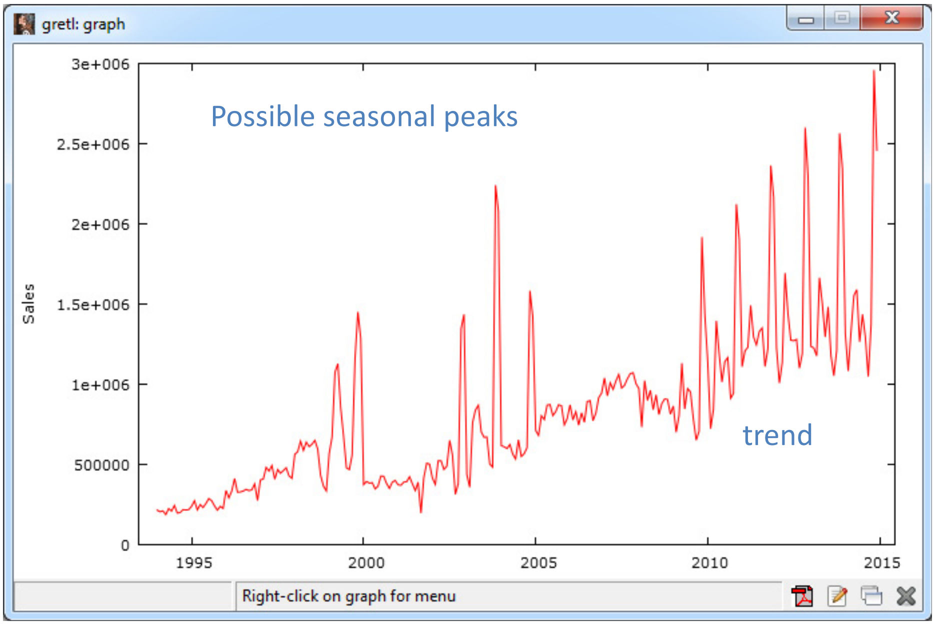


Figure 8.16

Rolling Thunder Bicycle sales by month. Notice the upward trend and the potential seasonal peaks. Some of the stronger peaks might be difficult to predict.

for trends, then try to find the best combination of auto-regressive and moving average lags for the model. The lag structure provides the key information about seasonality. The model itself and most tools make it easy to forecast future values for the series.

Goals

The main goal of traditional ARIMA is to identify the patterns inherent within a time series set of data. The patterns are defined in terms of auto-regressive terms that transfer information from prior values of the series and moving average coefficients that transfer data from prior values of variations—particularly seasonal variations.

The data consists of a simple series that is organized by time. Each row represents one time period and the time periods must be uniform (days, weeks, months, quarters, years, and so on). Most tools allow the addition of a column that holds the time definitions, making it possible to sort the series correctly (early to later). Typically, the data is created from a database query or OLAP cube because it represents subtotals by time period.

Tools

To illustrate the process, create a SQL query for the Rolling Thunder Bicycle company that computes total sales by month:

```
SELECT YEAR(OrderDate)*100+MONTH(OrderDate) As YearMonth,
SUM(SalePrice) As Sales
FROM Bicycle
GROUP BY YEAR(OrderDate)*100+MONTH(OrderDate)
ORDER BY 1
```

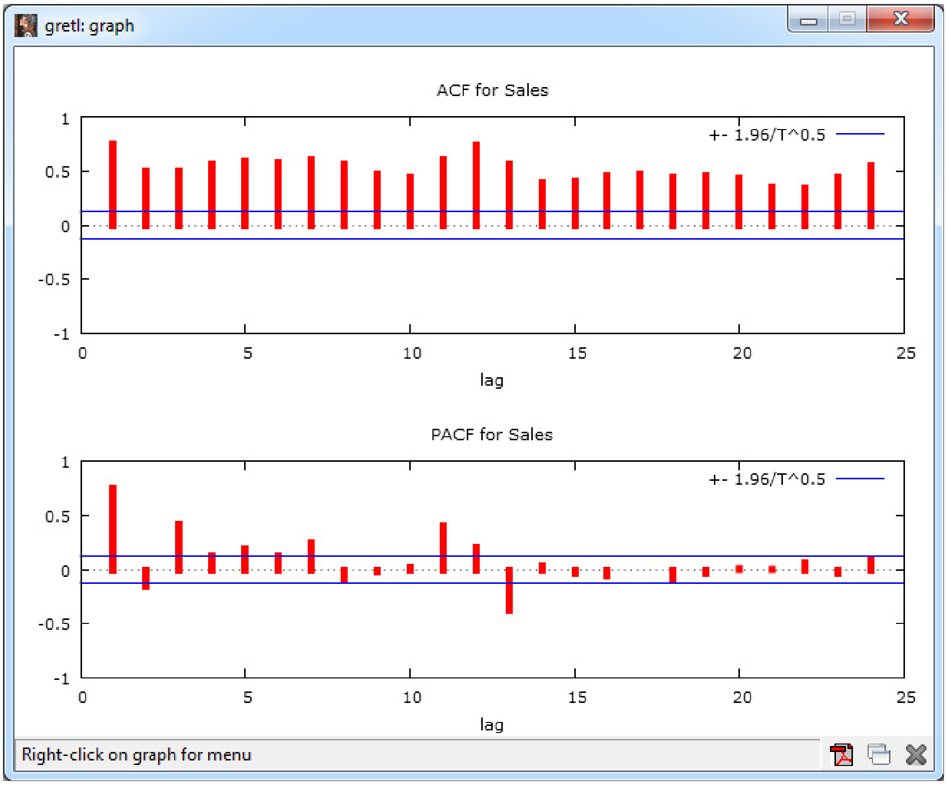


Figure 8.17

Initial correlogram. Notice the strong peaks in the PACF suggesting some important auto-regressive terms. But the ACF does not die down or cut off so the series is not stationary and needs to be differenced before trying to estimate the ARIMA model.

In SQL Server Management Studio, the results can be saved as a CSV file by right-clicking the query results and choosing the Save As option. You might want to use WordPad to add the column names to the top of the file. A CSV file can be imported into a standard tool that supports ARIMA such as gretl, which is free and relatively easy to use. The gretl package does not use the YearMonth column directly, so you need to specify that the file contains time series data that is monthly and begins in 1994:01. All of these options are specified when the file is opened.

One of the first steps to take with any times series analysis is to plot the data against time. It is critical to look for trends, but just exploring the data visually helps identify possible seasonal patterns. Figure 8.16 shows the plot for Rolling Thunder. First, notice the upward trend. This trend is good for business, but will need to be removed to analyze with ARMA. The trend appears to be linear, so a single differencing should solve the problem. Second, notice the potential seasonal peaks. Exploring the detailed data should show relatively higher sales in the last couple months of the year (holiday sales), plus more sales in the spring months. The five high peaks shown in the data will likely be hard to predict. They are probably related to economic or bicycle issues (such as the introduction of full suspension mountain bikes).

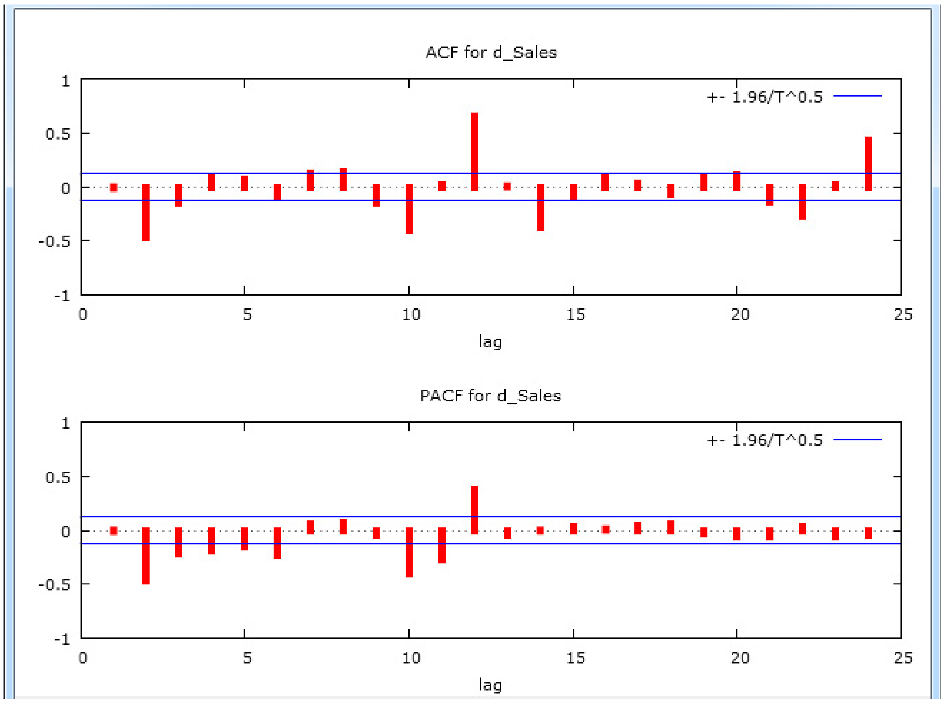


Figure 8.18

ACF and PACF for differenced data. Series appears to be stationary. The PACF dies down with additional peaks at 10 and 12. The ACF has peaks at lags 2, 8, 10, 12, and 14. The model leans towards MA, but has a complex structure.

The correlogram shown in Figure 8.17 reveals similar information to the raw data plot. Because the ACF does not die down or cut off, the series is not stationary and cannot be estimated with the raw data. At least one difference will be needed to remove the linear trend. The differencing can be handled directly within the ARIMA tool. However, it would be useful to look at the new correlogram first to see if a single difference is going to be sufficient. More importantly, the ACF and PACF based on the differenced data are needed to provide initial guesses for the AR and MA lag structure.

Gretl makes it easy to define a new series, so use Add/Series difference or create: Sales2=diff(Sales). Figure 8.18 shows the ACF and PACF for the new, differenced data. The data for the differenced series should also be plotted to ensure the trend has been removed. The PACF appears to die down after a few lags, but has some additional peaks at lags 10 and 12. The ACF has interesting peaks at lags 2, 8, 10, 12, and 14. The overall structure seems to be relatively complicated. It seems to lean towards an MA structure, with important lags specified by the ACF, but there are likely to be a couple of important AR terms as well—indicated by the PACF peaks. The peaks at 12 indicate important annual/seasonal elements.

Results

Some of the lag values are difficult to explain, but the first pass model should probably be ARIMA(4, 1, 3) with an additional MA coefficient estimated for a

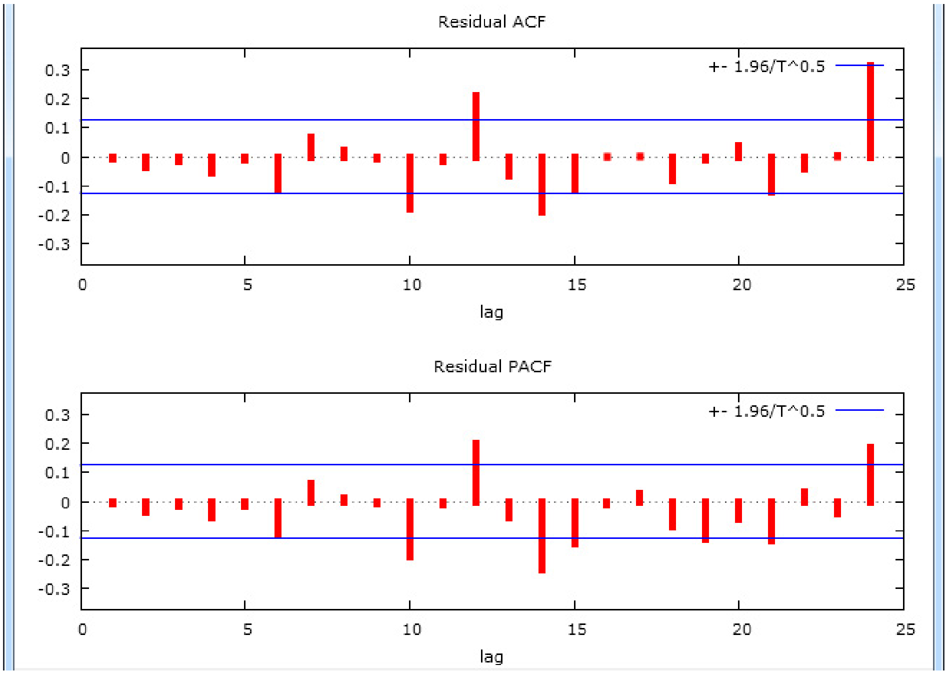
```
             Coeff.     Std. Err.    Z        P-value
const        6607.7     7364.5       0.8972   0.3696
phi_1        -0.1559    0.0991       -1.5730  0.1157
phi_2        -0.6170    0.0981       -6.2910  0.0000   ***
phi_3        -0.0234    0.0777       -0.3009  0.7635
phi_4        -0.2196    0.0764       -2.8750  0.0040   ***
theta_1      -0.0594    0.0794       -0.7484  0.4542
theta_2      0.0363     0.0734       0.0494   0.6213
theta_3      -0.4407    0.0659       -6.6850  0.0000   ***
theta_12     0.5127     0.0517       9.9160   0.0000   ***

Akaike criterion:       6930.493
Schwarz criterion:      6965.747
Hannan-Quinn:           6944.680
```

Figure 8.19

Results from initial model ARIMA(4, 1, 3) and MA 12. Notice the significance of the AR 2 and 4 lags and the MA 3 and 12 lags.

Figure 8.20

ACF and PACF for residuals. Major peaks have been reduced, but both the ACF and PACF show statistically significant effects at lags 10, 12, and 14.



```
              Coeff       std. error   t-ratio    p-value
---------     ---------   ----------   ---------- ---------
const         5589.04     757.496      7.378      0.0000   ***
phi_1         0.0255      0.0795       0.3208     0.7484
phi_2         -0.2282     0.0779       -2.9290    0.0034   ***
phi_3         -0.0000     0.0626       -0.0005    0.9996
phi_4         -0.0917     0.0504       -1.8180    0.0690
phi_12        0.6707      0.0492       13.640     0.0000   ***
theta_1       -0.3340     0.1031       -3.2410    0.0012   ***
theta_2       -0.2690     0.1139       -2.3610    0.0182   **
theta_3       -0.2348     0.0963       -2.4370    0.0148   **
theta_10      -0.1499     0.0567       -2.6410    0.0083   ***
theta_14      -0.0124     0.0611       -0.2032    0.8390

Akaike criterion:         6863.546
Schwarz criterion:        6905.851
Hannan-Quinn:             6880.571
```

Figure 8.21

Results from new model ARIMA(1 2 3 4 12, 1, 1 2 3 10 14). Notice the new terms are significant and the diagnostic measures have improved.

12-period lag. Figure 8.19 shows the results of that estimation. Notice the significance of the AR 2 and 4 lags along with the MA 3 and 12 lags. Some basic model evaluation numbers are also reported. They are useful when comparing this base model to other variations.

Is this model good enough now? The easiest way to answer that important question is to look at the correlogram of the residuals. Figure 8.20 shows the ACF and PACF for those residuals. First, notice that the major peaks have at least been reduced. However, both the ACF and PACF have effects at lags 10 and 14 that are significantly different from zero. Both are also negative, so they are likely to be secondary effects from the seasonal data. That is, people appear to buy more bicycles at a couple of times in the year (holidays and spring), and consequently, they pull back even further than expected just before and just after these two big events. Because the events are appearing in both diagrams, it is not clear if they should be estimate as AR or MA terms.

As a first pass you can try estimating the additional lag effects as MA terms but the estimation might not succeed. Also, the lag at 12 should probably be modeled as an AR term, bringing the full seasonal information forward. So the new model is ARIMA(1 2 3 4 12, 1, 1 2 3 10 14). Figure 8.21 shows the estimated coefficients. Notice the new lag effects are significant (except for MA 14) and the diagnostic values have improved from the original model. Consequently, this model is better than the initial one. Examine the residual correlogram to double check—no significant effects remain in the residual ACF or PACF. You could try running the model with the 14-lag in the AR side instead of MA to see if it makes a difference. (It does not so it could just be dropped, which slightly improves the model evaluation statistics.)

Once the model has captured the main effects, it should be compared to the actual data. Figure 8.22 shows a basic time series plot of the actual values versus

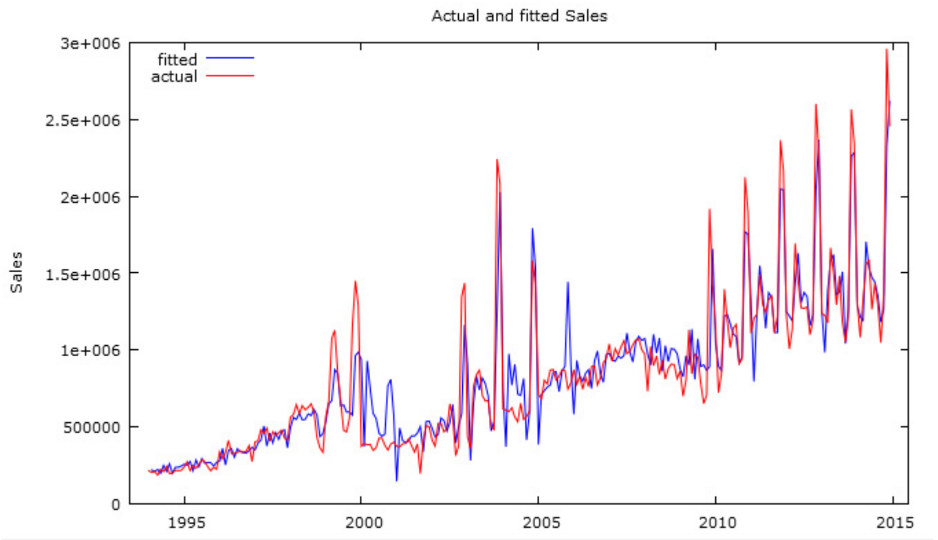
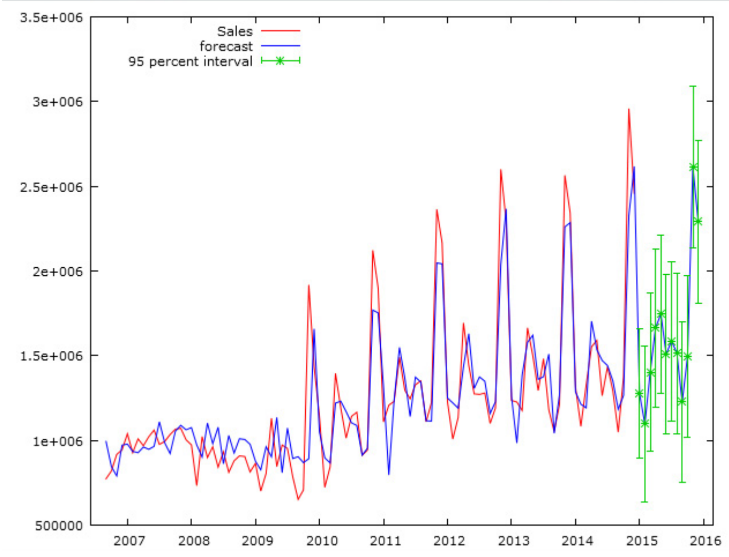


Figure 8.22

Plot of actual(red) versus values predicted by the model (blue). Overall the model appears close, except for an over estimation in 2001 and 2009. These drops are probably due to external factors.

Figure 8.23

Forecast into the future (12 months). The blue line is hard to see but it is the central wavy line. The green bars indicate the 95 percent confidence interval for the prediction.



```
2015:01      1276797
2015:02      1097255
2015:03      1402733
2015:04      1663312
2015:05      1746264
2015:06       1510111
2015:07      1587235
2015:08      1515373
2015:09      1228551
2015:10      1494301
2015:11      2617069
2015:12      2290510
```

Figure 8.24

Forecast values for 2015.

the values predicted by the current model. Overall, the model prediction appears to match the actual values fairly closely—surprisingly even for the major peaks. However, the predictions for 2001 and 2009 seem to be substantially higher than the actual sales for the months in that year. These drops in actual sales is most likely due to external factors. Any guesses? Do you know history? Plausibly the attacks of 9/11 played a role, but the other major effect that year was the dot-com e-commerce crash, which also caused a slight economic recession. Similarly, the housing crash of 2008/2009 probably reduced sales for a couple of years. Remember that the ARIMA method relies solely on internal patterns. Deviations similar to this one provide interesting points to look for external events that might have influenced the data.

Forecasts

The ARIMA model is well-designed for forecasting patterns into the future. Most ARIMA tools have options to generate forecasts and gretl is no exception. From the main model results, gretl has a menu option to generate a forecast using that model. Sticking with the default options leads to a standard forecast. Figure 8.23 shows the forecast for 12 months past the original data. The wavy blue line appears to copy the pattern over the prior three or four years with variations for seasonal effects. The green bars indicate the 95 percent confidence interval for each prediction. Notice that the range is relatively large—indicating that actual sales could be quite different from the expected forecast. Also notice that the confidence interval increases as the forecast moves further out in time. This effect is rational because more random events can affect patterns over longer periods of time. It might be tempting to discount the wide range of possibilities, but remember that the model does not evaluate external economic effects, and remember the changes missed in 2001 and 2009.

Figure 8.24 shows the actual data forecast by the model for 2015. The numbers could be used by managers to make purchasing decisions, change marketing plans, or develop long-term plans to expand or contract the business. The seasonal information is particularly useful for staffing decisions at various times through the year. The ARIMA tools also provide estimates of the standard errors; however, these numbers are quite high in this situation because of the high variability in data in the past years. These numbers will be useful to compare with forecasts created by other techniques in later sections of this chapter.

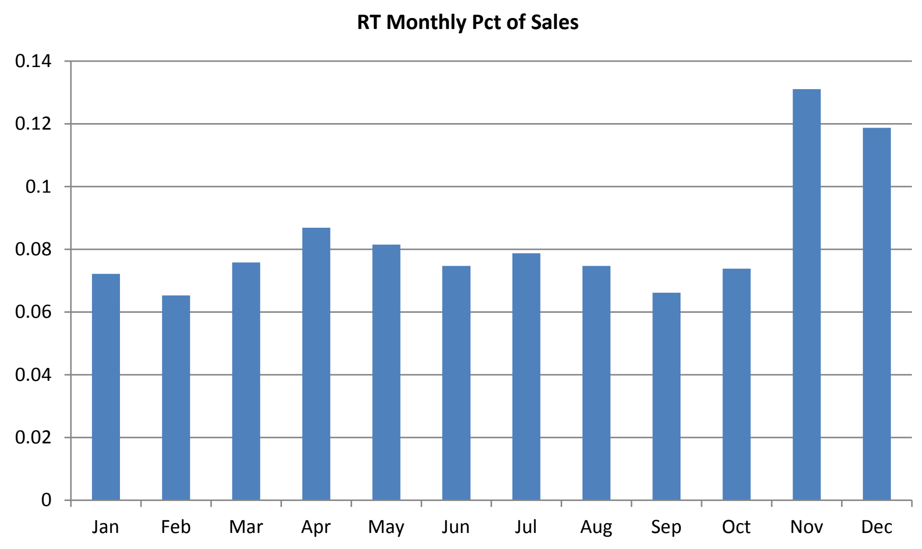


Figure 8.25

Average percent of sales by month. Computed from query of total sales by month, with percent computed in Excel.

Seasonality Evaluation

Another of the key strengths of traditional ARIMA is the visibility of the seasonal effects. These are easiest to see from the coefficients on the moving average terms. Specifically, the value of 0.6707 for the AR(12) term is positive and quite high. It indicates a strong seasonal effect—whatever happened 12 months ago is likely to repeat each year. To see the effects by month, create a query that computes the total sales by month for all of the years. Transfer the data to Excel to compute the overall total and divide to get the rough percent of sales by month. A more time-consuming approach is to compute the percentage of sales by month within each year and then compute the average of the percentages. This latter approach is more accurate and is a good exercise for the reader. Anyway, Figure 8.25 shows the primary sales months of November and December. Spring months of March, April, and May also appear to have slight gains.

Look again at the values for the MA lag coefficients. The MA coefficient are all negative. And only the AR(1) and AR(12) coefficients are positive. With monthly data, this value means strong seasonal effects exist. Now realize that the coefficients for MA(1), MA(2), and MA(3) are all negative. Yet, these are not auto-regressive coefficients, so the negative value needs to be interpreted carefully. Look again at the percentage sales values by month. Notice that months with big jumps (November and March) are preceded by months with low sales. The moving average coefficient applies to the variation within the month, not to the overall value. Hence, when the preceding three months are lower than average, the MA coefficients reveal that the sales in the target month (e.g., November) are even higher (negative coefficient times negative variation). This pattern means that consumers are essentially holding back on purchases during certain months and then buying at specific times—particularly for November sales. The negative values also mean that sales fall off after the big months.

The negative values for MA(10) and MA(14) are a little more difficult to explain. On face value, they would imply purchase decisions are made 10 and 14 months in advance. For example, customers who plan to purchase a bicycle in November 2010 have already made some decisions about that purchase in October 2009 and January 2010. The January effect of 10 months might be understandable, but the October effect takes a stretch of imagination. It could be an artifact of the data; or it could be a statistical result of the extremely low sales in October and January—note that it is small and not significantly different from zero. In any case, the important point is that the ARIMA coefficients clearly picked up these variations, but the annual percentage chart makes it easier to explain to managers.

Microsoft Time Series Estimation

How does the Microsoft Time Series estimation work? With SQL Server 2008 BI, Microsoft modified its time series estimation method by adding a version of ARIMA. The tool attempts to reduce reliance on analysts by using statistical tools to guess the lag structure. More importantly, the default estimation method (MIXED) also relies on a proprietary model called **auto-regressive tree with cross prediction (ARTxp)**. ARTxp is used for immediate (1-5 period forecasts). It uses a decision tree approach to try and determine which time periods affect the outcome. It also emphasizes cross correlation, so it encourages the addition of other series that might provide information to predict the original series.

With an emphasis on prediction instead of evaluation, the Microsoft results can be challenging to interpret. However, it is easy to incorporate other attributes into the analysis. Still, as shown in the Results section, the forecast results can be highly variable depending on the model and any parameter hints.

Goals

The Microsoft approach focuses on the prediction of the time series, hopefully with minimal intervention by the analyst. Microsoft uses two models estimated independently to forecast data. It is possible to restrict the process to a single model, and users of the enterprise version of SQL Server can control the mixture rate. However, the default is to use both models (MIXED), so it is important to understand both methods.

The ARTxp method is a decision tree approach that can be thought of as a piecewise linear regression model. The tool searches for change points that indicate a different effect on the outcome variable. Figure 8.26 shows that these change points can be created from various points within the time series or from external, cross-correlated series. Within the time series, the argument is that a time series might exhibit a different pattern while the series is declining compared to when the series is increasing. If so, the ARTxp method creates a node and estimates new model coefficients for each segment. When mixed with the ARIMA process, each leaf node has a separate ARIMA model. Forecasting entails finding the matching decision tree leaf node and applying its model. In the simple example, if the data to be forecast is earlier than 2003, and the Income level is greater than 50,000, ARIMA model 2 would be used to make one forecast. The node also contains ARTxp parameters to make a separate forecast. Depending on the mixture parameters, the two values are averaged to obtain the final forecast. By default, short-term forecasts within about 5 periods of the ending data are predominantly made with the ARTxp model. Longer-term forecasts lean more heavily on the ARIMA

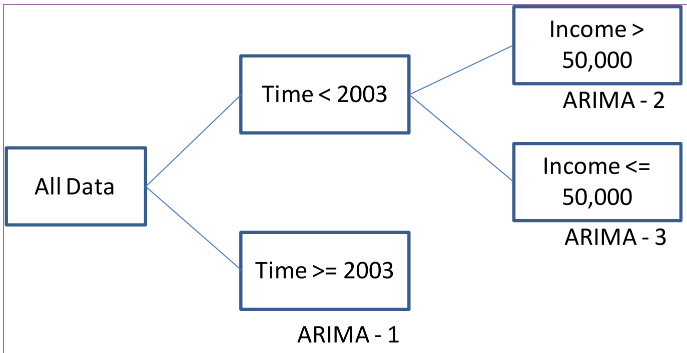


Figure 8.26

Microsoft ARTxp goal. Find split points where a model change can improve the prediction. Mixed with ARIMA, each leaf node has a separate ARIMA estimate.

model based on a parameter that sets the weighted average. Predictions that are farther out in time rely more on the ARIMA model because it is less volatile.

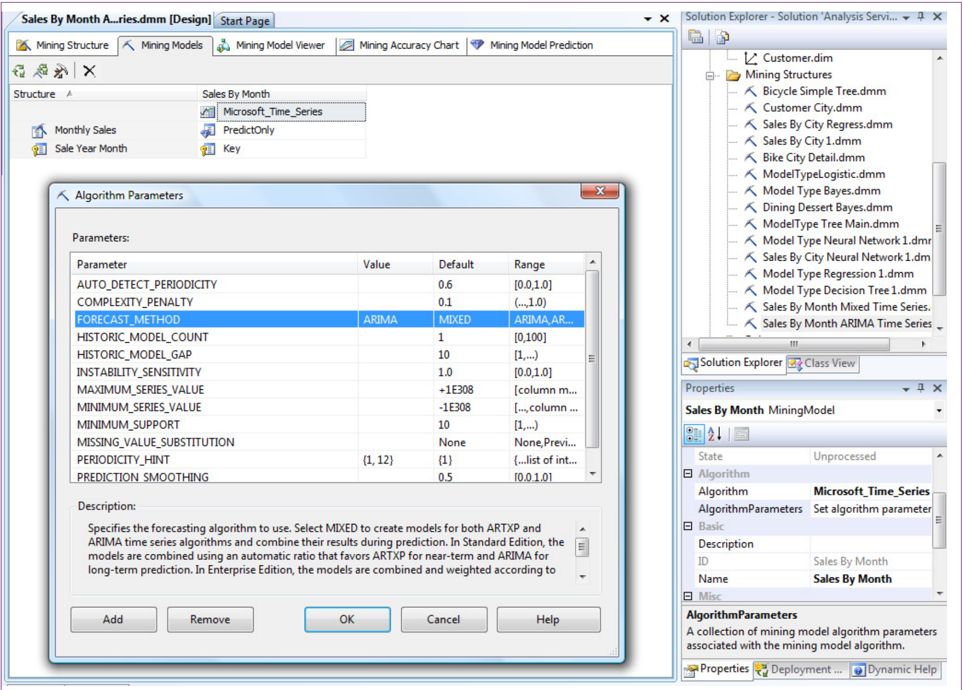
The ARIMA method used by Microsoft is also slightly different from the traditional model. As proposed by Box-Jenkins, seasonal models can be made easier to estimate with a multiplicative seasonal structure. The seasonality (s) first needs to be defined. For example, it would be 12 in a monthly model and 4 in a quarterly model to estimate annual effects. Then, a separate **seasonal ARIMA (SARIMA)** model is estimated based on lags for the seasonal data, specified separately as P, D, Q. The seasonal lags (P and Q) are defined as multiples of the seasonality. Typically, they are small values (1 or 2 at most). For example, a value of P=1 would generate a **seasonal auto-regressive (SAR)** estimate of one season (or 12 months). A value of Q=2 would estimate a **seasonal moving average SMA**(2) lag equal to 24 months. Differences are made at the seasonal level, so D=1 leads to subtracting all values 12-months apart. It should never be necessary to go above 1 difference for seasonal components. The model is typically written as:

ARIMA (p, d, q) x (P, D, Q)

The model is multiplicative, where the seasonal terms are multiplied by the unit terms. Consequently, the seasonal terms should be kept small or the model ends up with a huge number of coefficients to be estimated. A common seasonal model is based on two levels of exponential smoothing:

ARIMA (0, 1, 1) x (0, 1, 1)

Microsoft also complicates the results somewhat by estimating up to 8 times the number of lags specified. The high-end value of 8 is often used for the unit ARIMA, and rarely for the higher-order seasonal component. That is, if periodicity is specified as MA (1) or ARIMA (0, 1, 1), the tool will actually estimate coefficients for MA(1), MA(2), … MA(8). Consequently, it is seldom necessary to specify a base ARIMA with a parameter greater than 1 because the other values will be estimated anyway. Most commonly, if you know that the data is monthly, you would specify the periodicity parameter as: {1, 12}.



Structure	Sales By Month
	Microsoft_Time_Series
Monthly Sales	PredictOnly
Sale Year Month	Key

Algorithm Parameters

Parameters:

Parameter	Value	Default	Range
AUTO_DETECT_PERIODICITY		0.6	[0.0,1.0]
COMPLEXITY_PENALTY		0.1	(...,1.0)
FORECAST_METHOD	ARIMA	MIXED	ARIMA,AR...
HISTORIC_MODEL_COUNT		1	[0,100]
HISTORIC_MODEL_GAP		10	[1,...)
INSTABILITY_SENSITIVITY		1.0	[0.0,1.0]
MAXIMUM_SERIES_VALUE		+1E308	[column m...
MINIMUM_SERIES_VALUE		-1E308	[...,column ...
MINIMUM_SUPPORT		10	[1,...)
MISSING_VALUE_SUBSTITUTION		None	None,Previ...
PERIODICITY_HINT	{1, 12}	{1}	{...list of int...
PREDICTION_SMOOTHING		0.5	[0.0,1.0]

Description:

Specifies the forecasting algorithm to use. Select MIXED to create models for both ARTXP and ARIMA time series algorithms and combine their results during prediction. In Standard Edition, the models are combined using an automatic ratio that favors ARTXP for near-term and ARIMA for long-term prediction. In Enterprise Edition, the models are combined and weighted according to

Customer.dim
Mining Structures
Bicycle Simple Tree.dmm
Customer City.dmm
Sales By City Regress.dmm
Sales By City 1.dmm
Bike City Detail.dmm
ModelTypeLogistic.dmm
Model Type Bayes.dmm
Dining Dessert Bayes.dmm
ModelType Tree Main.dmm
Model Type Neural Network 1.dmr
Sales By City Neural Network 1.dm
Model Type Regression 1.dmm
Model Type Decision Tree 1.dmm
Sales By Month Mixed Time Series.
Sales By Month ARIMA Time Series

Properties

Sales By Month MiningModel

State	Unprocessed
Algorithm	
Algorithm	Microsoft_Time_Series
AlgorithmParameters	Set algorithm parameter
Basic	
Description	
ID	Sales By Month
Name	Sales By Month
Misc	

AlgorithmParameters
A collection of mining model algorithm parameters associated with the mining model algorithm.

Figure 8.27

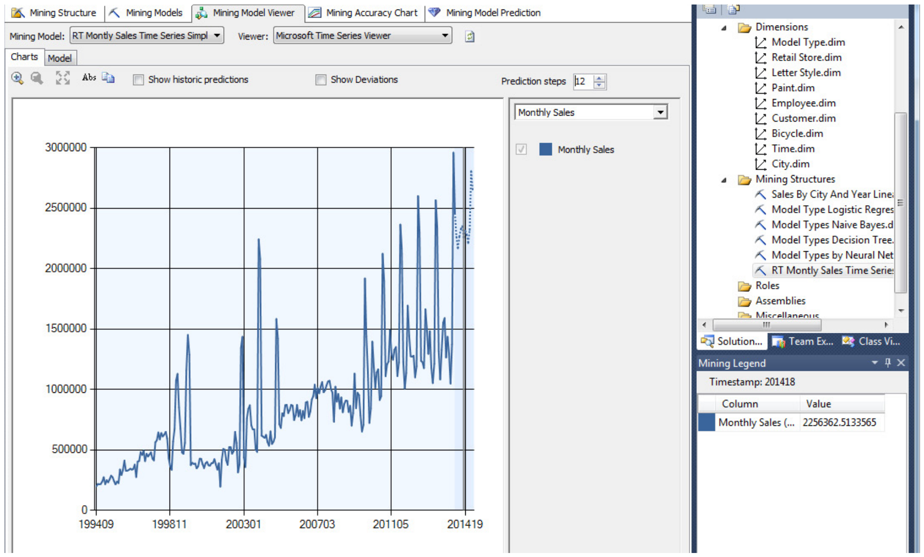
Microsoft time series set to ARIMA. Select Microsoft Time Series in the Mining Model tab. In the properties window, click the button under the Algorithm Parameters option. Enter ARIMA for the Forecast_Method and {1, 12} as the Periodicity_Hint.

Data

Time series data for the Microsoft model is usually computed with a GROUP BY query or the subtotals are pulled directly from an OLAP cube. The data must have a time key column that uniquely identifies the rows so they can be sorted in the proper order. The column should be numeric but the data values are not important. A common approach is to combine year and month numbers, or quarter numbers for quarterly data). A named query can be created quickly. The query for Rolling Thunder Bicycles is:

```
SELECT YEAR(OrderDate)*100 + MONTH(OrderDate) AS
SaleYearMonth, SUM(SalePrice) AS MonthlySales
FROM dbo.Bicycle
GROUP BY YEAR(OrderDate) * 100 + MONTH(OrderDate)
```

Any additional attributes that will be used for cross correlation must match the time key data. If the data is imported, it should be imported with a similar key column so the named query can use a JOIN to match the values correctly. Also, remember that there should be no missing data. Microsoft Time Series does offer a parameter with several options to control how missing data values will be replaced. However, even this option should not be used if there are too many missing observations. Instead, aggregate the data to a higher level.



Mining Model: RT Montly Sales Time Series Simpl

Viewer: Microsoft Time Series Viewer

Prediction steps 12

Mining Legend

Timestamp: 201418

	Column	Value
🟦	Monthly Sales (...	2256362.5133565

Figure 8.28

Microsoft ARIMA forecast. The forecast chart is similar to that from the traditional ARIMA forecast.

Tools

For comparison to traditional ARIMA, try running a model with just the ARIMA component first. The ARTxp or MIXED version that blends in Microsoft's proprietary tree method can be run second—it complicates the interpretation of the results.

Create a new mining structure using Microsoft Time Series and select the data source with the named query containing the sales totals by month. Follow the wizard to create the model. Choose the MonthlySales as the predictable column. It does not need to be selected as an Input column. The SaleYearMonth column should automatically be selected as a key time column. As shown in Figure 8.27, the ARIMA specification is made as a parameter change. On the Mining Models tab, select the Microsoft_Time_Series entry. Switch to the parameters window in the lower-right corner. Highlight the Algorithm Parameters entry and click the ellipses button to open the parameter edit window. Find the Forecast_Method row and enter ARIMA as the Value. Find the periodicity hint and enter {1, 12} to indicate the series contains monthly data. Process and Browse the model.

Figure 8.28 shows the results can be displayed as a chart. The default forecast is 5 periods, but it can be increased by setting the number of prediction steps to 12. At first glance, the chart forecast appears similar to that created with traditional ARIMA. Switch to the Model view to see the estimated coefficients.

Because the model was restricted to simple ARIMA, only one tree node is displayed in the model view. Select that node and examine the Mining Legend. The ARIMA model is displayed in the legend, but it is difficult to read. Figure 8.29 shows the model reformatted and labeled by hand. The multiplicative seasonal component (P, D, Q) is listed after the "X" symbol. It is labeled with (12) to indicate the seasonal length and the algorithm determined that one was estimated

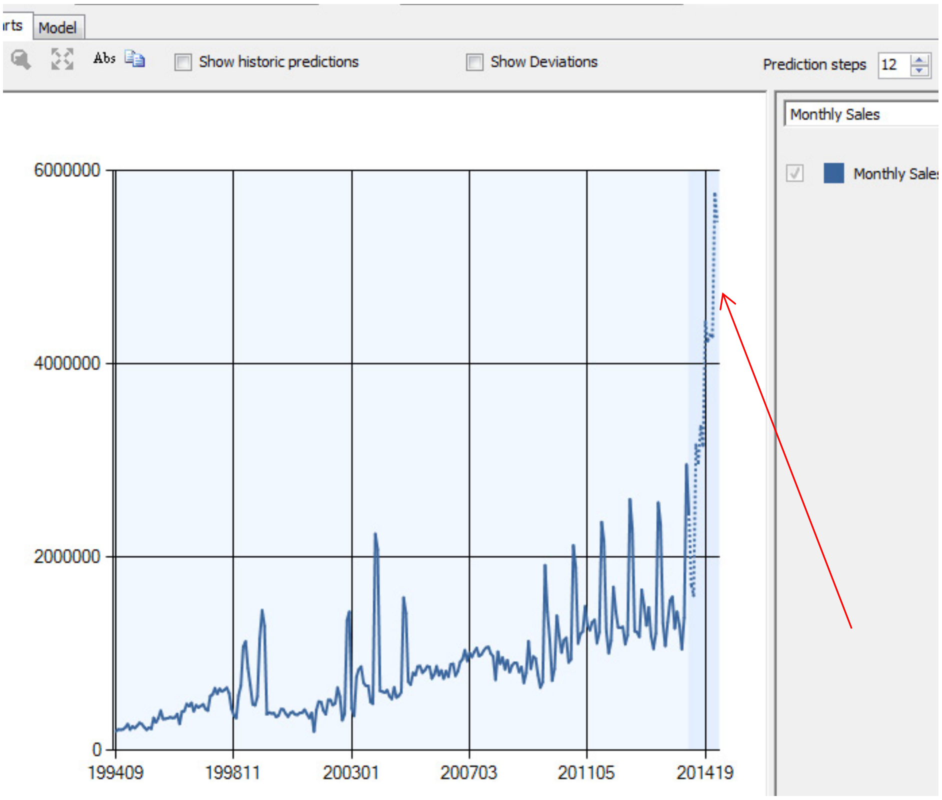
```
ARIMA (
          {1,0.23311},                    AR(1)
             1,                           difference
         {1,-0.14026})                    MA(1)
X (
         {1,-0.39775,5.34085E-02}, SAR(12)
             0,                           S difference
         {1,-6.72065E-02})                SMA(12)
Intercept:7225.834
```

Figure 8.29

Microsoft ARIMA coefficients. The legend display is hard to read. The coefficients here are displayed on separate lines and labeled by hand.

Figure 8.30

Microsoft MIXED forecast. The model predicts a large drop in sales for the next year—both immediately and throughout the year. Based solely on internal sales, this forecast seems unrealistic.



Prediction steps 12

```
ARIMA (
         {1,-1.45534E-02},                    AR(1)
             1,                               Diff.
         {1,-0.74364}) X                      MA(1)
         ({1,0.27151,                         SAR (6)
             4.86931661E-02},                 SAR(12)
             0,                               Diff.
          {1,0.767121})(6) X                  SMA(6)
         ({1,0.10754,                         SAR(4)
             0.09174323},                     SAR(8)
             1,                               Diff.
          {1,2.832889E-02})(4)                SMA(4)
Intercept:7293.6173
```

Figure 8.31

ARIMA coefficients for root node Note the seasonal auto-regressive terms. The structure of the model was selected automatically to use a 4 and 6 month seasonality lag.

at 12 months. The coefficients are different from those computed with the traditional ARIMA model. Note that gretl can estimate a seasonal ARIMA model, and when it is specified similar to this version, the coefficient values are closer to these results.

The goal is not to see if Microsoft ARIMA can produce identical results to traditional ARIMA tools. The point of the example is to ensure that the model can be run and the results retrieved and interpreted. The more common method of using Microsoft Time Series is to stick with the default model: Mixed and let the tool choose the lag structure. In other words, the least amount of supervision by the analyst.

Create a new mining structure, choose the same data source and follow the default prompts to create a new Time Series model based on the mixed ARTxp/ARIMA process. Figure 8.30 shows the result as a chart. Even before checking the coefficients and estimates look at the forecast for the next 12 months. The model predicts that sales are going to jump precipitously—even continuing through the year. Clearly, an outcome that predicts a three-fold increase in sales in a cople of months is unlikely to be realistic—based only on the data available from this internal list. Something is wrong with the model. It is possible to dig into the model in more detail, but the simplest solution is to remember that all of the default model parameters were used. So the biggest problem is that the system does not know that the data values are monthly.

To see what is happening, look at the Model structure which is displayed as a tree diagram. Select the root node (or one of the end nodes) to see the model for that slice of data. Although cut off in the figure, the ARIMA coefficients can be found by scrolling the Mining Legend. Figure 8.31 shows the values for the same future node. First, check the seasonal terms and recognize that the algorithm guessed at SAR(6) and SAR(4) which covers half-year, and slightly more than quarterly effects; but it did not estimate any annual effects. Remember that the algorithm does not know the series contains monthly data.

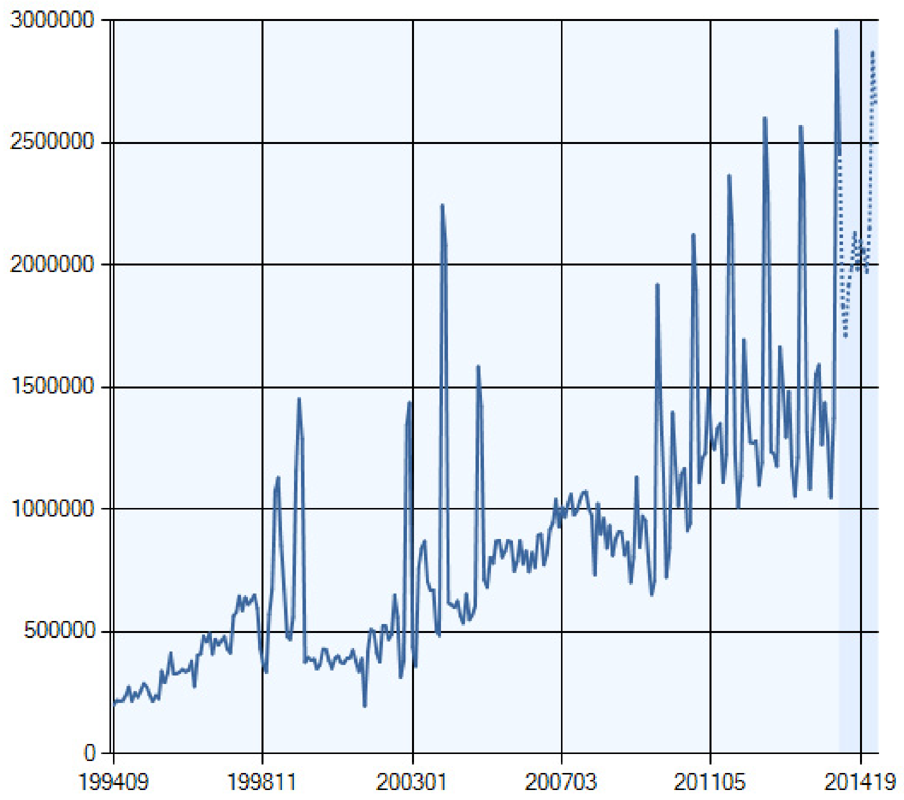


Figure 8.32

Microsoft MIXED forecast with monthly periodicity hint of 12. The forecast is considerably more realistic and appears to follow the trend and seasonal variations now.

The answer is to tell the tool that the data has 12 months in the year. Build a new time series model. After the wizard has finished, open the Mining Models tab and select the Microsoft_Time_Series entry. Open the Properties tab and click the ellipses button on the entry for Algorithm Parameters. Scroll to find the Periodicity_Hint and enter a value of {1, 12} to specify monthly data. This parameter is the only thing that will be changed from the default specification. Process and Browse the model to see the results.

Results

Figure 8.32 shows the charted forecast results for the slightly modified model. Compare the forecast to the one in Figure 8.30 computed with the default periodicity selected automatically. Telling the algorithm that the main series holds monthly data made a huge improvement in the forecast. So you should always plan on specifying the overall periodicity of the series. Of course, if you enter values that do not match the data, the results are likely to be even worse.

```
ARIMA (
          {1, 0.23311},      AR(1)
          1,        diff.
          {1,-0.14026})      MA(1)
X (
          {1,-0.39775,       SAR(12)
          0.0534085},        SAR(24)
          0,        diff.
          {1,-0.06721})(12)  SMA(12)
Intercept:7225.8342
```

Figure 8.33

ARIMA coefficients for the root node with the periodicity hint. Note the new seasonal terms for 12 months and 24 months.

The difference lies in the ARIMA model—which did use the periodicity hint. Figure 8.33 shows the new coefficients for the root node displayed from the default model. Because of the periodicity hint of monthly, the new ARIMA model contains seasonal moving average terms for 12 and 24 months. The misleading auto-regressive values for lags 4 and 6 months are gone. Why is this model better—in the sense that the forecast does not go crazy? First, notice that the SMA term is negative in this model, compared to the positive values for the SMA terms in the initial model. This negative value provides a damping effect. The same holds true for the SAR terms. The main AR(1) term is positive and strong in this model, which would lead to higher forecast values, but the other terms tend to dampen the effects.

ARTxp Model

The ARIMA results are actually just a small portion of the results from the Microsoft time series model. They are a useful starting point because they are similar to the traditional ARIMA model. However, the Microsoft approach downplays the role of the ARIMA model and focuses on decision-tree approach. As shown in Figure 8.34, the tool creates a tree structure by splitting nodes where it is possible to improve the results. At this point, the model contains only the key column (YearMonth) and the data column (Sales). Still, the tool found several points that justified splitting the model. However, the tree is stronger when additional data is added. This mechanism makes it relatively easy to add correlated series including economic data.

The ARIMA model coefficients remain the same for each node, but the forecast value is modified within each node based on the other terms. Figure 8.35 shows that each node has a separate equation. For example, for the node on the right side (Sales-7 < 710039.8) the additional equation is:

Monthly Sales = 395688.159 + 0.499 * Monthly Sales(-12)

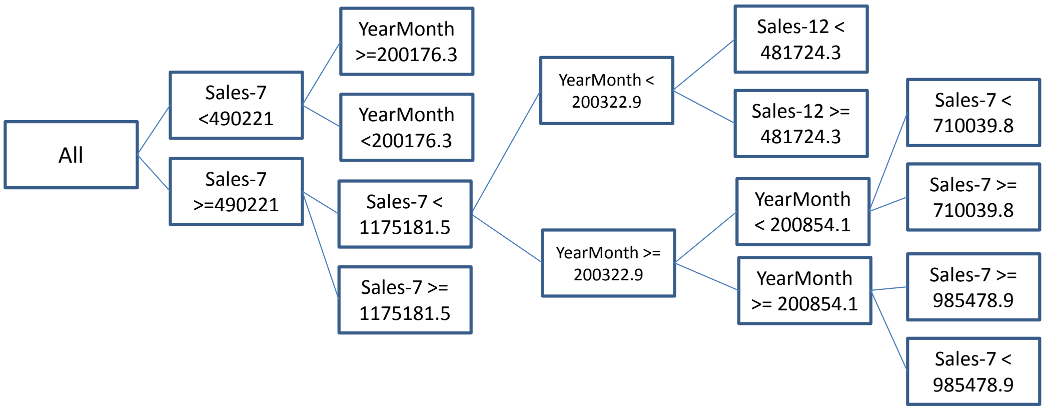
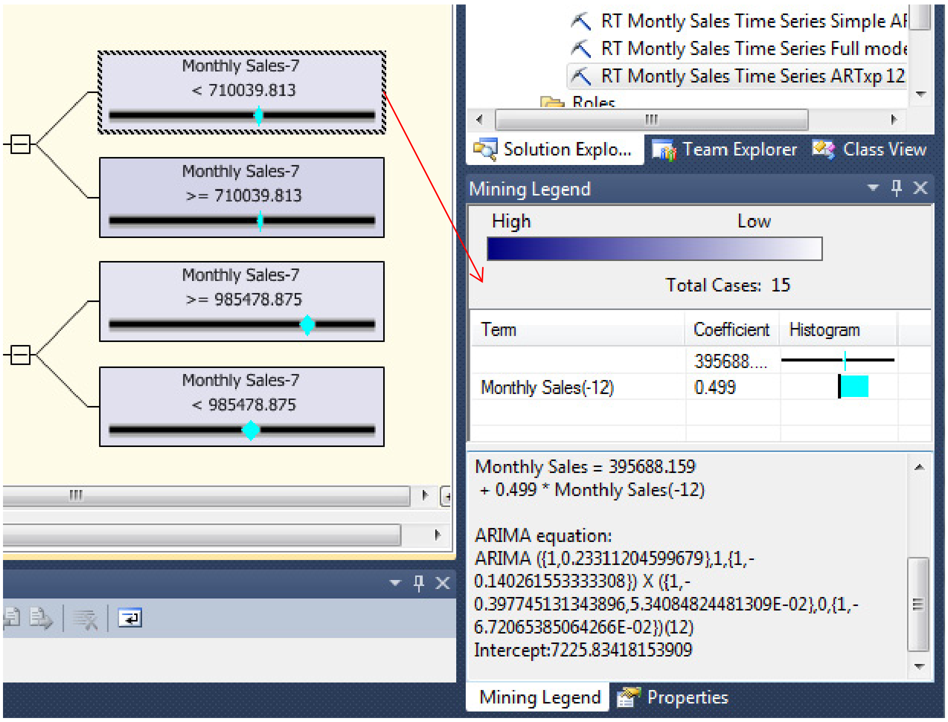


Figure 8.34

ARTxp decision tree results. The tool searches the associated data for breakpoints where the model results can be improved by adding a node to a tree branch. This model has only the key column (YearMonth) and the sales data itself.

Figure 8.35

Node differences. Select a node to see the equation for that item on the tree. The ARIMA remains unchanged but each node has an additional equation.


Monthly Sales-7 < 710039.813
Monthly Sales-7 >= 710039.813
Monthly Sales-7 >= 985478.875
Monthly Sales-7 < 985478.875
RT Montly Sales Time Series Simple AI
RT Montly Sales Time Series Full mode
RT Montly Sales Time Series ARTxp 12
Roles
Solution Explo...
Team Explorer
Class View
Mining Legend
High
Low
Mining Legend
Properties

Total Cases: 15

Term	Coefficient	Histogram
	395688....	
Monthly Sales(-12)	0.499	

```
Monthly Sales = 395688.159
 + 0.499 * Monthly Sales(-12)

ARIMA equation:
ARIMA ({1,0.23311204599679},1,{1,-
0.140261553333308}) X ({1,-
0.397745131343896,5.34084824481309E-02},0,{1,-
6.72065385064266E-02})(12)
Intercept:7225.83418153909
```

Time	Sales	StDev
201413	1831835	115390
201414	1709642	113679
201415	1914923	110646
201416	1991393	77833
201417	2130782	105124
201418	1980396	102778
201419	2098216	100685
201420	2056840	102464
201421	1971231	106954
201422	2148194	91530
201423	2868532	96244
201424	2647986	93432

Figure 8.36

Forecast values formatted with Excel. Note the Time data is useful only for sorting.

Forecasts

The forecasting process with Microsoft Time Series is different than in the other tools. In particular, a time series forecast does not need source data, so there is no need to set up a table of values, or to enter data for a singleton query. If only one or two values are needed, the forecast chart displays the graph and provides a way to obtain one value at a time. Simply select a point on the chart with the mouse and the corresponding predicted value is displayed in a table on the right. However, this process is cumbersome for multiple values.

To obtain a set of predicted values, and to get their standard deviations, you need to enter a special MDX query. This query needs to be entered in the BI processor, not in the SQL Server query processor. Switch to the Mining Model Prediction tab. Click the icon to switch to SQL, which opens an edit window in the bottom half of the screen. The simple query to get the predictions uses the **PredictTimeSeries** function:

```
SELECT FLATTENED PredictTimeSeries([Monthly Sales], 12) As
Forecast
FROM [RT Monthly Sales Time Series ARTxp 12]
```

The query is straightforward, with the only drawback of having to type the table name and column name by hand. Be sure to specify the FROM section with the name of the model you are using in brackets. The PredictTimeSeries function needs only two parameters: The column name and the number of periods (e.g., 12) to be forecast. Forecasting a single value is always risky—it is always good to include the standard deviation of the forecast value to obtain the accuracy of the forecast. This query is slightly more complex because the standard deviation needs to be computed for each forecast value:

```
SELECT FLATTENED
(SELECT *, PredictStdev([Monthly Sales]) As [StDev]
FROM PredictTimeSeries([Monthly Sales], 12))
As Forecast
FROM [RT Monthly Sales Time Series ARTxp 12]
```

Figure 8.36 shows the output from the query. The actual output is a little messier. These values were formatted in Excel to make them easier to read. Notice that the Time variable is useful only for sorting. To restore the business meaning, the numbers would have to be manually changed to 201501, 201502, and so on. The standard deviation values can be used to compute 95 percent confidence intervals to indicate the accuracy of the forecast.

The MDX language includes additional functions to return the variance (PredictVariance) and the decision tree node that matches the conditions (PredictNodeID). The language also has powerful tools to create forecasts by changing some of the input data. An even more interesting tool is to predict a new short set of data based on the patterns established from the longer series. For instance, what if RT managers have introduced a new model type and have sales data for only a few months. This data would not be sufficient to establish a useful pattern on its own. Instead, the sales patterns from the entire company can be applied to this new data. That is, take the trend and seasonal information from the overall sales and apply it to the data from the new model. The syntax is a little tricky, but it is such a useful trick it is worth examining:

```
SELECT FLATTENED
  PredictTimeSeries([Monthly Sales], 24, REPLACE_MODEL_
CASES)
FROM [RT Monthly Sales Time Series ARTxp 12]
PREDICTION JOIN
(SELECT 201407 AS [Sale Year Month], 25621 As [New Model]
  UNION SELECT 201408 AS [Sale Year Month], 25621 As [New
Model]
  UNION SELECT 201409 AS [Sale Year Month], 11592 As [New
Model]
  UNION SELECT 201410 AS [Sale Year Month], 32119 As [New
Model]
  UNION SELECT 201411 AS [Sale Year Month], 27559 As [New
Model]
  UNION SELECT 201412 AS [Sale Year Month], 18225 As [New
Model]
) AS t
ON [RT Monthly Sales Time Series ARTxp 12].[Sale Year
Month] = t.[Sale Year Month]
 AND [RT Monthly Sales Time Series ARTxp 12].[Monthly
Sales] = t.[New Model]
```

In this query, the PredictTimeSeries function specifies a prediction of 24 periods and it uses the REPLACE_MODEL_CASES parameter to ensure new data is used as the starting point of the prediction. This new data is provided manually by the PREDICTION JOIN subquery. Notice that the new data includes the SaleYearMonth entries that match up to the existing values for the end of 2014.

Figure 8.37 shows the results of the forecast. The data was copied and pasted into Excel to generate the chart. Notice that the new model sales data form the starting point. The forecast applies the overall seasonal pattern to that initial data and generates a forecast specifically for the new model. Of course, it is possible that the new model sales will be radically different—particularly if a special advertising campaign is created. Nonetheless, the ability to apply the overall seasonal pattern to the small set of data is a useful technique.

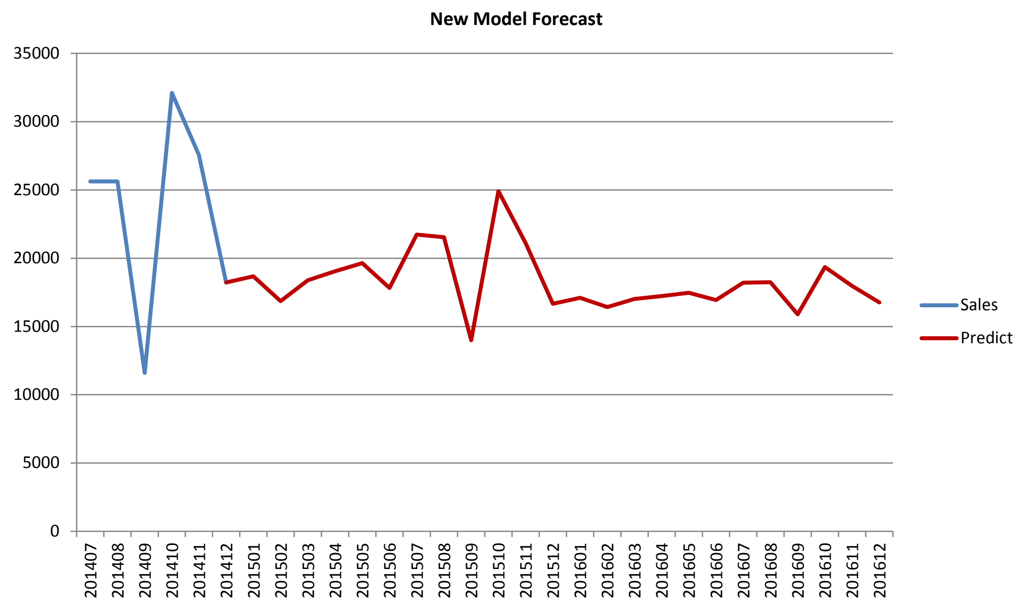


Figure 8.37

Existing sales pattern applied to a new model. The new model starting values were entered manually but then the existing pattern is applied to yield a forecast that follows the seasonal patterns.

Seasonality Evaluation

The ARTxp model provides only a limited amount of information about seasonality. Recall that the node results included an AR term for a 12-month lag, indicating that a portion of sales from 12 months back will automatically be included in a prediction for the current month.

The ARIMA model is more powerful at identifying seasonal patterns—much the same way it is used in the traditional regression approach. The objective is to evaluate the coefficients on the various seasonal terms. Because Microsoft time series relies on the seasonal model, the analyst can focus on the seasonal P and Q terms. In the final monthly model estimated here, the results returned two seasonal terms: SAR(12) = -0.398, SAR(24) = 0.053, and SMA(12) = -0.067. The 12-month term represents a one-year lag and indicates that a seasonal pattern does exist. Each year, sales in a specific month will be determined somewhat by the sales that occurred one year ago in that month. The interesting twist is that the sales will be negatively impacted by the sales from one year ago in that month. But the positive SMA(24) coefficient buffers that value by adding sales from two years ago.

Cross Correlation and Linear Regression

How can the effects of other series be incorporated into the prediction? The question of multiple time series falls into the subject heading of dynamic systems. It is a complex topic that can involve detailed mathematical models. Few data mining packages handle the complex tools by default, and if you really need these high-end tools, it would be safest to consult with an expert. For example, the ARMA process has been extended to vectors. Data sets with multiple time series can also run into issues involving simultaneous equations, and problems with fitting multiple models to the data. So, at least read all of the literature on a specific tool before attempting to use it. On the other hand, it is possible to get basic estimates for models that incorporate time series and multiple input variables. Just remember that these tools only scratch the surface, and this section is merely an introduction.

Goals

The basic goal is to predict future values for a time series. The time series can exhibit trend, auto-regressive, and seasonal effects. It can also be affected by other attributes. Economic models are fairly common examples. From a business perspective, sales can be heavily dependent on overall economic data. For example, in economic terms, bicycle sales are most likely a normal good—which means that as consumer income increases, consumers will potentially buy more high-end bicycles. Certainly, if an economy dips into a recession, sales of high-end bicycles are likely to decline. If it is possible to predict overall economic trends, this information could be used to improve forecasts for specific products, such as custom bicycles. Of course, any external attributes that are added to the model must be predictable by some process. Fortunately, most broad economic measures are important to many people, so it is possible to find short-run forecasts by experts. These forecasts are not always perfect, but they are available and are probably as good as possible given the random nature of the world.

As usual, a secondary goal of including additional attributes is simply to learn more about the data. If some series do affect sales, just that knowledge can help improve decision making. Similarly, if some attributes do not significantly affect the outcome, many decision problems become simplified.

Data

The predictable variable should remain the same as for basic time series analysis: A single column of observations at regular time intervals. It will usually be necessary to include a key column that defines the sequence of the time data. As with simpler models, this column is often a composite number consisting of the year and month for monthly data. Quarter or day numbers can be used for different intervals. If linear regression is going to be used to analyze the data, the data set should also contain simple columns containing the year and the month number. These columns will become separate attributes to measure trend and seasonal effects. Any data for cross correlation should be held in separate columns—being careful to match the time increments of the original series.

Economic data is often used to estimate effects on internal data. Standard economic series are readily available from government Web sites. Several U.S. government agencies provide tools to find and download economic data. A few international organizations provide data on other nations, but the U.S. data is available free of charge—some organizations charge fees to download data. The U.S. government maintains a Web site that serves as an index to the various agencies. To find data, start at www.fedstats.gov. One catch to federal data is that it is often seasonally adjusted. It can be difficult to find raw data for some government series. In many cases, seasonally adjusted data will be acceptable for cross correlation analysis. For example, comparing national income to sales will work even when the economic income data are seasonally adjusted. The sales data will already contain any seasonal variation which can be measured, so any correlation with the economic income data would represent basic income effects.

For the Rolling Thunder Bicycle sales example, it is possible to download disposable personal income and a price index (inflation) measure. Both of these measures are available as monthly data, and both are seasonally adjusted. To be used with Microsoft BI, the data series need to be downloaded and imported in to SQL Server. It is often easier to copy-and-paste the data into Excel first and then save the two series as a CSV file which can be imported into a new table in SQL Server. While the data sits in an Excel worksheet, you can add the time stamp information for year and month. Later, this column makes it possible to join the economic data to a query that computes total sales by month. The U.S. Bureau of Economic Analysis reports standard national income and product account (NIPA) data. For the bicycle sales case, Table 2.6 reports Disposable Personal Income as monthly data. Table 2.8.4 reports the personal consumption expenditures (PCE) price indexes on a monthly basis. Both series can be found at the www.bea.gov/national/nipaweb site. Find each data set separately, set the range for 1994/01 through 2014/12 to get the data that matches the sales for Rolling Thunder Bicycles. The data can be downloaded and saved as a CSV file or copy-and-paste the data into Excel and save the data as a CSV file. SQL Server can import a CSV file and create a new table to hold the data. Be sure to include a YearMonth column that is in the same format as the SaleYearMonth column used to compute the subtotals of sales by month (such as 201201).

Tools

Ignoring high-end complex tools, the most common method of incorporating cross correlation into time series analysis involves some version of linear regression, or a modified version of ARIMA. Microsoft Time Series uses the decision tree approach to emulate a regression effect. Simply including another data series as an Input attribute leads the decision tree model to look for node points where the values produce a different effect on the results. The decision tree essentially estimates a piecewise-linear model to find the node values that are significant factors.

Box and Jenkins, who defined the ARIMA approach, also defined a method to incorporate external data into the prediction. Some tools, including gretl, support this approach. With this approach, as with Microsoft ARTxp, the analyst simply needs to import the new data series and add the attribute as an Input variable to the model. The tools do most of the work automatically.

It is also possible to use linear regression directly to build a linear model of trend, seasonality, and cross correlation effects. The key lies in creating the correct set of X-attribute values. These attributes would be easier to create if SQL Server supported the Lag function, because a common solution involves using y_{t-1} as an input variable. Linear regression requires the most amount of supervision and setup, but it also provides detailed control over the model.

Microsoft ARTxp

Adding cross correlation to a Microsoft time series model is relatively easy—once the data is imported and available for analysis. The ARTxp process was specifically designed to handle multiple input attributes. The model simply factors them into decisions about creating tree nodes as split points and the results are available in the Mining Legend.

By now, you should have downloaded and imported the Income and price index from the BEA database into SQL Server. Assume the columns are in a table called DPI with YearMonth as a primary key. The challenge now is to build a named query that (1) computes the subtotal of sales by month and (2) joins the result to the imported DPI data. The process is somewhat tricky because data sources do not handle one-to-one joins, and named queries apparently cannot reference other named queries. The complication arises because the SalesByMonth calculation must be performed first to create the SaleYearMonth column which is then joined to the YearMonth data in the DPI table. One solution to the problems would be to build the SalesByMonth as a saved view within SQL Server itself, then join the DPI table to the result either within SQL Server or in the data source. However, the entire calculation and join can be handled as a named query with the little trick of using a temporary table:

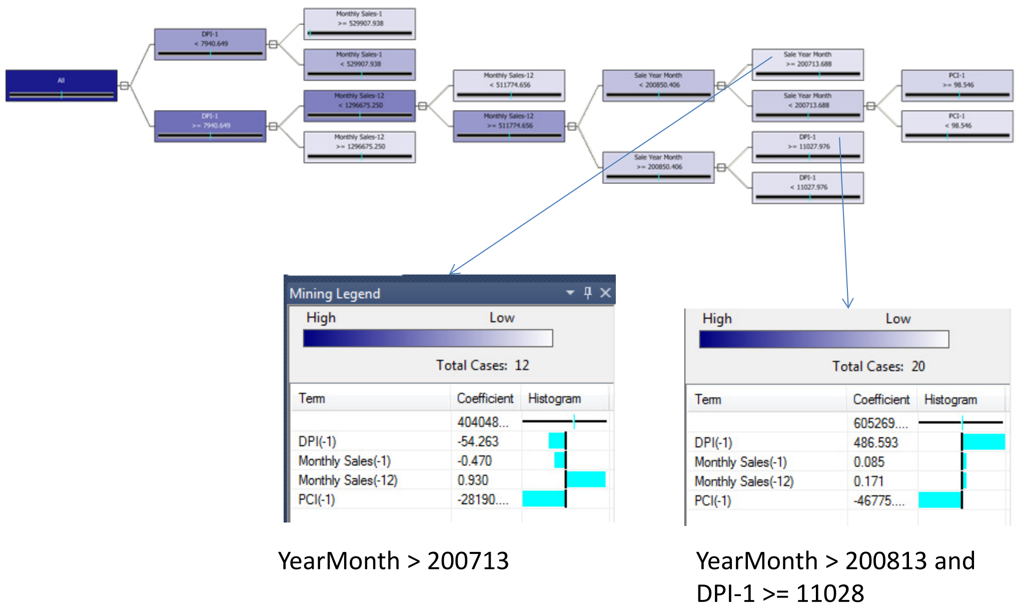
```
SELECT t.SaleYearMonth, t.MonthlySales, dbo.DPI.DPI, dbo.
DPI.PCI
FROM dbo.DPI INNER JOIN
(SELECT YEAR(OrderDate) * 100 + MONTH(OrderDate) AS
SaleYearMonth, SUM(SalePrice) AS MonthlySales
  FROM dbo.Bicycle
  GROUP BY YEAR(OrderDate) * 100 + MONTH(OrderDate)) AS t
ON dbo.DPI.YearMonth = t.SaleYearMonth
```

The inner SELECT statement handles the computations to get monthly sales and this output is assigned to an internal temporary table named t. The outer select joins those rows to the DPI table based on the year/month columns and displays the results. Examine a couple rows of the named query to see that it works correctly.

After the data is defined, building the mining model is straightforward. Create a new mining model using Microsoft Time Series, based on the new named query. Work through the wizard, selecting Monthly Sales as the predictable column, [Sale Year Month] as the time key, and add DPI and PCI as Input columns. After the wizard finishes, remember to return to the Mining Models tab and specify the Periodicity_Hint as {1, 12} to indicate monthly data. Select the Microsoft_Time_Series entry, and click the ellipses button in the properties for Algorithm Parameters. Process the model and Browse the results. The ARIMA model results should be the same as before—they are based solely on the predictable series. Although, the dataset might be slightly different because real-world data might not exist yet for the 2013 and 2014 years.

Figure 8.38 shows the results for the node that handles all months after 2007. Check the Mining Legend for details about the node. The fact that the change point is 2008 is important alone—that was a key year in the housing market crash. The coefficients indicate the declining trend with the negative coefficient on sales for the prior month. That shift altered the effect of both income (DPI) and inflation (PCI). In essence, the values measure a structural shift in the underlying economic impacts.

At this point, an analyst for the company would look at the values for all of the other nodes. Note there is another sale month change after 2008, which is further split based on values of the DPI. Fortunately, that node (not shown here) reveals that the coefficient on DPI turns positive after 2008 and once DPI exceeds a value of 11028. The point is that these nodes providing the starting point for deeper


All
DPI-1 < 7940.649
DPI-1 >= 7940.649
Monthly Sales-1 >= 529907.938
Monthly Sales-1 < 529907.938
Monthly Sales-12 < 1296675.250
Monthly Sales-12 >= 1296675.250
Monthly Sales-12 < 511774.656
Monthly Sales-12 >= 511774.656
Sale Year Month < 200850.406
Sale Year Month >= 200850.406
Sale Year Month >= 200713.688
Sale Year Month < 200713.688
DPI-1 >= 11027.976
DPI-1 < 11027.976
PCI-1 >= 98.546
PCI-1 < 98.546
Mining Legend
High
Low
High
Low

Total Cases: 12

Term	Coefficient	Histogram
	404048...	
DPI(-1)	-54.263	
Monthly Sales(-1)	-0.470	
Monthly Sales(-12)	0.930	
PCI(-1)	-28190....	

YearMonth > 200713

Total Cases: 20

Term	Coefficient	Histogram
	605269....	
DPI(-1)	486.593	
Monthly Sales(-1)	0.085	
Monthly Sales(-12)	0.171	
PCI(-1)	-46775....	

YearMonth > 200813 and DPI-1 >= 11028

Figure 8.38

Results of the ARTxp model with economic data. Results for the node: Sale Year Month >= 200713. First note that 2008 caused a shift point in sales. Second, the coefficients indicate the downward trend from the prior month; and the role of DPI and PCI have changed.

understanding of the data. Analysts who lived and worked through 2008 might remember the impact of the U.S. housing market crash. The key is that it does show up in the model results.

ARMAX Addition of Series to ARIMA

Many tools that support traditional ARIMA also include the option to analyze the impacts of other data on the main time series. Gretl has a simple option to specify other series as inputs, and then estimates an ARMAX model based on the approach defined by Box-Jenkins. The new columns are evaluated as single points. The model does not automatically test these columns at different lag points.

The data from the U.S. BEA Web site needs to be integrated into the existing sales data. Because the sales data is stored in a simple CSV file, the easiest approach is to edit that file in Excel, download the two new columns and paste them into the main CSV file. Estimating the model is almost identical to the traditional ARIMA model. The trend needs to be differenced out and the lag structure needs to be identified from the ACF and PACF charts. These lags should be similar to the values found in the initial model with the single series, but they can be tweaked in the new model if necessary. Set up the model as before, using AR terms of 1 and 12, a single difference (D=1), and MA terms of 1, 2, 3, and 10. Then in the gretl model definition window, move the DPI and price index variables to the Input box. Run the model.

```
           coefficient    std. error     t-ratio    p-value
const        -8710.8        8019.9       -1.086     0.2774
phi_1         0.0286        0.0812       0.3529     0.7242
phi_12        0.5701        0.0591       9.6520     0.0000         ***
theta_1      -0.2700        0.0950      -2.8420     0.0045         ***
theta_2      -0.4881        0.0802      -6.0870     0.0000         ***
theta_3      -0.0854        0.0892      -0.9572     0.3885
theta_10     -0.1565        0.0552      -2.8350     0.0046         ***
DPI           220.61         187.1       1.1790     0.2385
PCI          37251.3         42794       0.8705     0.3840
```

Figure 8.39

ARMAX (gretl) results with the DPI and price index series. The ARIMA structure is similar to the base model. The DPI coefficient is negative but not significant, and the price index (PCI) is positive but not significant.

Figure 8.39 shows the coefficient estimates for the new model. The lag structure is similar to the base model. The new series (DPI and PCI) do not appear to have significant impacts on the monthly sales data. Both are positive but not significant. The inflation measure is also not significantly different from zero. If the PCI coefficient really were positive, it would likely indicate that the value of sales increases during times of inflation. Because the sales value is measured in nominal prices, such a result would simply indicate that managers raise prices during inflationary times, so total sales increase. For this reason, any model that examines price changes should probably look at quantity of sales instead of just value.

Linear Regression

Linear regression has been used for many years and several variations have been developed to handle time series and dynamic data. The theoretical details are beyond the scope of this book, but issues involving lags, auto-regression, and other complications can be found in almost any econometrics book. The techniques used in this section are relatively common methods to handle basic estimation of trends, seasonal effects, and evaluate the impact of other data on the main series. In some ways, bringing in multiple series can cause the most problems. Consider the basic problem of trying to estimate a demand curve by evaluating data collected over time that compares quantity sold to sale price. Economic theory reveals that this relationship should almost always be negative. Yet, in many cases, if this data is estimated with a simple regression, the coefficient will be positive. The problem is that too many other variables will affect the quantity sold, and at least some of these need to be added to the estimation. In econometric terms, it is a problem with simultaneous equations. The observed data consists of changes in the equilibrium point or intersection of the demand and supply curves over time. The model needs enough data to measure changes in both the demand curve as well as the supply curve. Then special estimation techniques are needed to separate out the impacts on both curves. Again, these topics are beyond the scope of this book. Just remember that when you try to measure effects from multiple time series, you should seek the assistance of an expert.

With time series data, it is highly likely that values in one period affect values in the next. This auto-regressive model is common in economics and several methods exist to estimate the effect. One of the easiest approaches is to define lagged variables of the dependent variable. That is, create new X-variables that are lags of the Sales variable and estimate the coefficients directly:

$$Sales = b_0 + b_1 Sales(-1) + b_2 Sales(-2) + b_3 Sales(-3) + b_4 Sales(-12)$$

The analyst can choose the lag variables, similar to the way AR terms are selected in the ARIMA model. The regression results will return a test of the significance of each coefficient.

Trend patterns are even easier to estimate. Simply include a measure of the time variable—often the year is used to measure annual changes. These values are easily generated in SQL with the Year function when the sales totals are created. Using a generic (-i) to represent all lag values, the model becomes:

$$Sales = b_0 + b_i Sales(-i) + b_y Year$$

Seasonal data is also relatively easy. It is tempting to use a Month column similar to the concept of the Year column. This approach will work, but the interpretation is different from a true seasonal measure. It would indicate whether any month affected the results, with no way to identify which month or season made a difference. To determine which specific months made a difference, it is necessary to create dummy variables—essentially one for each month. A **dummy variable** has a value of zero or one. The value is one if the desired attribute (month) is true, otherwise it is zero. So, define 12 new variables. In gretl, an example is: M01=(Month==1). This statement defines a new variable (M01) and sets each observation to 0 except when the Month value is 1 (January). However, when using dummy variables in a linear regression, a specific problem arises. If all 12 dummy variables are included in the regression, it is not possible to include the constant term. Because only 12 months exist, all of the data is covered and defined by those 12 dummy variables. The constant term is then a linear combination of the 12 variables and linear regression does not work if some variables are perfectly correlated. The solution is to drop either the constant term or to drop one of the 12 dummy variables. In most cases, it makes sense to keep the constant term and drop the first dummy variable (M01 for January). Nothing is lost. The impact results for January can be obtained as a linear combination of the other months and the constant term. The regression model now looks something like:

$$Sales = b_0 + b_i Sales(-i) + b_y Year + b_{m2} M02 + \ldots b_{m12} M12$$

Finally, it is possible to add terms for the other series. These variables will hold the same interpretation as they would in a traditional linear regression model. The model can now be written as:

$$Sales = b_0 + b_i Sales(-i) + b_y Year + b_{m2} M02 + \ldots b_{m12} M12 + b_d DPI + b_p PCI$$

Of course, it is also possible that the exogenous variables (DPI and PCI) do not have an immediate effect on sales. Particularly for income, it is possible that the impact is delayed. People might choose to make a purchase a few months in advance (particularly for the year-end holidays). Even if income drops in the month of the sale, consumers might complete the purchase because the decision has already been made. So, perhaps the real effect of income occurs a month or so before the sale. This dynamic situation can be evaluated by including lagged values of the DPI and PCI data. It should be clear that the regression approach is flexible enough to accommodate relatively complex models.

	coefficient	std. error	t-ratio	p-value	
const	-173894000	64787700	-2.684	0.0079	***
DPI_1	-105.5	181.089	-0.582	0.5610	
DPI_3	-137.9	186.6	-0.739	0.4608	
PCI_1	7500.1	45595.8	0.1645	0.8695	
PCI_3	7349.5	44302.2	0.1659	0.8684	
M02	51363.2	78077.0	0.6579	0.5114	
M03	105546	81738.7	1.291	0.1982	
M04	136553	78115.6	1.748	0.0821	*
M05	78413.0	73504.1	1.067	0.2874	
M06	98425.6	73524.4	1.339	0.1823	
M07	123598	76786.4	1.610	0.1092	
M08	98079.3	78434.4	1.250	0.2127	
M09	68710.6	77535.8	0.8862	0.3767	
M10	146903	78651.6	1.868	0.0634	*
M11	438455	82203.4	5.334	0.0000	***
M12	130437	83798.5	1.557	0.1213	
Year	87203.4	32756.3	2.662	0.0084	***
Sales_1	0.6043	0.0709	8.527	0.0000	***
Sales_2	-0.3251	0.0829	-3.925	0.0001	***
Sales_3	0.1356	0.0826	1.642	0.1023	
Sales_4	-0.02148	0.0734	-0.3051	0.7606	
Sales_12	0.2702	0.0615	4.395	0.0000	***

Figure 8.40

Regression results. Notice the significant seasonal (month) terms. Notice the positive trend (Year). The AR lags on the dependent variable loosely match the results from the other tools. The external series (DPI and PCI) do not have a significant impact on sales. $R^2 = 0.7799$.

The process of obtaining all of the lag and dummy variables depends on the specific tool used. Most, such as gretl, SAS, and SPSS have functions to define lags and new variables with one command. The interactive version of gretl even has a form that makes it easy to specify the number of periods to use for each lag.

Figure 8.40 shows the results of the regression. The R^2 value indicates the regression results are reasonable—however, time series data with auto-regressive terms often have high R^2 values, and 78 percent is not exceptional. Check the significance on the seasonal (monthly) terms. The results loosely match those from the ARIMA model. April, and November are important months. However, July and October show strong coefficients here, but not in the ARIMA model. The catch with the dummy variable approach is that the January results are incorporated into the values. July and October usually have lower sales than other months, but they are higher on average than January. The coefficient on the Year variable is significant and positive, so there is an upward trend to the data in the

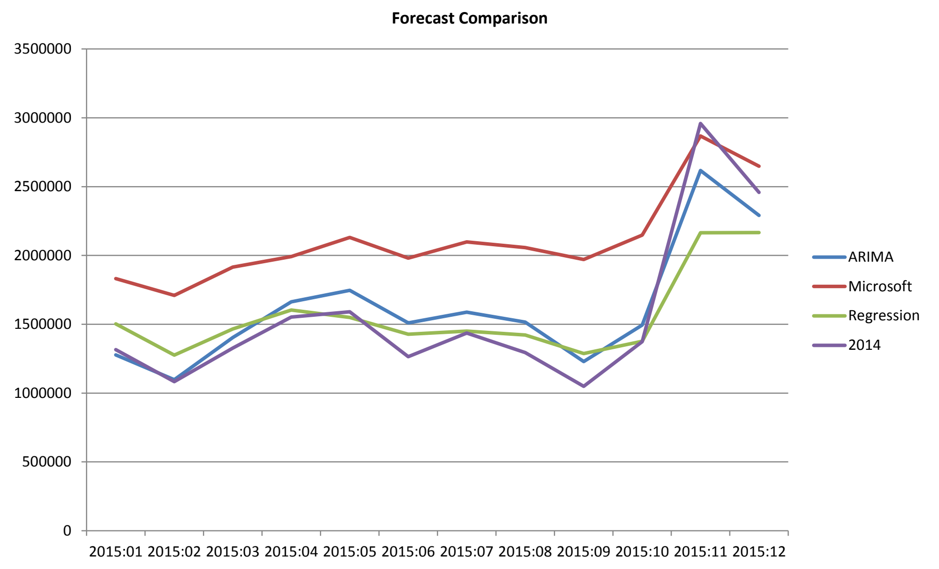


Figure 8.41

Forecast comparison. The forecasts are relatively similar, with the Microsoft ARTxp values slightly higher than the others. On the other hand, the forecasts are not much different from the "actual" 2014 values.

amount of about $87,000 a year. The AR lags on Sales indicate that prior sales are important—particularly for the past two months. One month back has a strong positive influence on sales for the current month, but this boost is mitigated slightly by the negative effect from two-months prior. The significant coefficient on the 12-month lag also indicates the presence of a seasonal effect.

Finally, notice that the DPI (income) and PCI (inflation) terms do not significantly affect the sales results. Even the lagged values are not significantly different from zero. A couple of the lagged coefficients appear strong, so perhaps if more data is acquired, more variables are added, or variability declines, these variables might influence total sales. The key point is that it is relatively easy to incorporate external data and still measure time series effects with linear regression.

Comparison

This chapter covers several tools that try to accomplish the same task: Identify patterns in time series and use them to forecast future values. Many data mining tools for time series rely on ARIMA as the foundation to estimate trend and seasonal effects. Still, variations in the methods lead to differences in the coefficients and in the forecasts. Figure 8.41 shows the forecasts for the first four months of 2015 using the three basic tools covered in the chapter. The values are all relatively close, except the Microsoft ARTxp forecast numbers appear higher than the others. The ARIMA forecast has a bit more variation than the Microsoft forecast. Note the slight increase over the "actual" 2014 values for the regression and ARIMA forecasts. Overall, they are showing minimal trend effects. And although the ARTxp forecast seems too high for most of the year, it is the only one that seems to pick up the full forecast for the months of October and November. Regression

Lag	ARIMA/MA	Microsoft/MA	Regression/AR
1	-0.3340	-0.1403	0.6043
2	-0.2690		-0.3251
3	-0.2348		0.1356
4	-0.0917		-0.0215
10	-0.1499		
12	0.0124	-0.0672	0.2702

Figure 8.42

Seasonality comparison. Because of the dissimilarities the results are difficult to summarize. However, the primary 12-month lag values are except for Microsoft's ARTxp.

and ARIMA are both predicting that those months for 2015 would be lower than the observed values in 2014. The bottom line is that no model is "perfect."

Figure 8.42 is more confusing. It compares the seasonality results from the three models. The values are not completely comparable because the techniques are somewhat different. Specifically, the linear regression approach estimates AR terms instead of MA values. Still, it would be nice if the seasonal effects were similar across the models. The big difference is that the 12-month lag coefficient is positive except in Microsoft's ARTxp model. Even including the SAR(12) and SAR(24) terms does not resolve this difference. Although not shown in the table, the ARTxp AR(1) term of 0.233 is larger than the AR(1) term of 0.026 from the ARIMA model. So the ARTxp model is responding more strongly to recent events. The negative coefficients for the 2-period lag are interesting—they are also somewhat incorporated into the Microsoft 1-period lag. They imply a consistent damping effect against the other attributes. In essence, when RT has a good sales month, the next month is almost always going to be lower. But that effect represents the seasonality—the data indicates that consumers tend to have a couple of targeted purchasing months.

Summary

Time series data is common in business. Managers commonly make forecasts based on patterns over time. Basic questions include the total level of sales, peaks and troughs in weekly and seasonal patterns so that adequate staffing can be provided and forecasting consumer needs for the next holiday period. Time series data often exhibit patterns—including long-term trends or growth over time, seasonal patterns such as holiday sales or weekly peaks, and cyclical changes that follow the general economy. The essence of time series forecasting is to evaluate a series and identify its patterns. These patterns can then be used to forecast the series into the future. Auto-regressive moving average tools are a common method for evaluating patterns—particularly seasonal patterns. These AR and MA terms are defined as coefficients that weight the effect of lags in the original data and the variations. Traditional Box-Jenkins/ARIMA tools provide auto-correlation and partial auto-correlation functions to help determine the lag structure of time series data. Trends are removed from ARIMA models by differencing the data—usually once for linear growth, twice for non-linear trends.

The Microsoft Time Series tool mixes ARIMA forecasts with a proprietary ARTxp tool that uses a decision-tree approach to find significant split points in the data. ARTxp focuses on auto-regressive lags to determine a piecewise linear model. Microsoft claims ARTxp is best at forecasting out a few time periods. For further forecasts, the tool mixes and eventually switches to a prediction based on an ARIMA model. The Microsoft forecasting tool has a useful trick to apply patterns in a larger series to a new series that is assumed will follow the same pattern—such as when introducing a new model or product.

Incorporating effects from business cycles (national income) or from other time series events requires tools that can handle dynamic models. Complex tools have been developed to analyze specialized data—particularly in finance—but this chapter covers only basic tools. Some ARIMA tools, such as gretl, support the addition of other series and can estimate coefficients to measure their correlation with the original series. Microsoft ARTxp evaluates the effects of other series when building a decision tree, but does not change the ARIMA model. Linear regression can be used to estimate trend, seasonal, and cross-correlation effects. Trend is estimated by including a time variable in the model. Seasonal effects are identified by adding a binary dummy variable for each season, less one if the constant term is included. Time series models generally include lagged dependent variables as well to capture dynamic effects and autocorrelation. Cross correlation or cyclical effects are evaluated by including other variables, where the resulting coefficients provide an estimate of the correlation.

When time series data is highly variable, the results from the different tools can vary. When multiple models are created that have different results it can be difficult to choose among the models to determine the most likely forecast. Various diagnostic measures can be used to compare models, but ultimately the final prediction will simply have a high variation. The bottom line is that forecasting the future is often difficult.

Key Words

Akaike information criterion (AIC)
auto regression
auto regressive integrated moving average (ARIMA)
autocorrelation function (ACF)
auto-regressive tree with cross prediction (ARTxp)
chaotic
cross correlation
cyclical
dummy variable
moving average
partial autocorrelation function (PACF)
PredictTimeSeries Microsoft function
Schwarz criterion or Bayesian information criterion (BIC)
seasonal ARIMA (SARIMA)
seasonal auto-regressive (SAR)
seasonal effect
seasonal moving average (SMA)
seasonality
seasonally adjusted
time series
trend

Review Questions

1. What are the main components of a time series?
2. What is the difference between auto-regressive and moving average terms in a time series?
3. How does the ARIMA model handle long-term trend in a time series?
4. How are the ACF and PACF used to identify the preliminary lag structure of a times series?
5. What measures are often used to compare the performance of different time series models?
6. How is missing data handled in time series analysis?
7. How are forecasts made with the traditional ARIMA model?
8. How is seasonality identified with the ARIMA model?
9. How is Microsoft ARIMA different from the traditional ARIMA (ignoring ARTxp for now)?
10. What is the most important parameter to specify in a Microsoft Time Series model?
11. What does Microsoft's ARTxp provide that is not available in traditional ARIMA?
12. What is the purpose of cross correlation models and what problems might they create?

Exercises

Book

1. Set up and run the traditional ARIMA example from the chapter. Summarize the results and interpretation.
2. Set up and run the Microsoft Time Series analysis example from the chapter. Summarize the results and interpretation.
3. Set up and run the cross-correlation example from the chapter using Microsoft Time Series analysis. Summarize the results and interpretation.
4. Compute the average percent sales by month for Rolling Thunder Bicycles. As outlined in the chapter, compute the percentage of sales by month within each year and then average those values by month. Compare the results to the simpler method used in the chapter.

Rolling Thunder Database

5. Rerun the time series analysis using quantity of bicycles sold instead of sale price as the primary series. Compare the results to those using prices.

6. Rerun the time series analysis using sale prices, but use quarterly instead of monthly subtotals. Compare the results to those using monthly totals.

7. Examine the time series of purchases of component parts from vendors. Is there a seasonal pattern?

8. Assume the company introduces a new model type (downhill or free ride) in 2014. Given the sales values listed in the following table, and assuming that the sales will follow the pattern of all mountain bikes, use Microsoft Time Series tools to estimate the sales pattern for 2015.

July	August	September	October	November	December
55,612	38,291	36,289	25,423	67,500	71,300

Diner

9. Examine the total sales by week and identify any seasonal patterns that exist.

10. Examine the total sales by day of week and identify any patterns that exist.

Corner Med

11. Examine total visits by month and identify any seasonal patterns and trends that exist.

12. Examine total visits by day of week and identify any patterns that exist.

13. Examine total revenue (charges) from visits by month and identify any seasonal patterns and trends that exist.

Basketball

14. Choose one team and one season. Examine average points per game scored by that team by week and identify any time series patterns.

15. Choose one team and season and examine total fouls by that team by week and identify any time series patterns.

16. Choose one team and season and compute the average free throw percentage by week (total free throws made / total free throws attempted per week). Identify any patterns found.

17. Choose one player. Examine total points scored by that player per game by week in one season and identify any time series patterns.

Bakery

18. Examine total sales revenue by month and identify any trends and patterns.

19. Examine total sales revenue by day and identify any trends and patterns.

20. Examine total sales revenue by hour and identify any daily patterns.

Cars

21. Use the monthly car sales data to create a new database table. Identify any trends and seasonal patterns in the total sales. Source: Bureau of Economic Analysis: http://www.bea.gov/national/nipaweb/nipa_underlying/TableView.asp?SelectedTable=55&FirstYear=2009&LastYear=2009&Freq=Month&ViewSeries=N

22. Use the data from the previous question and identify patterns and trends for the domestic and import manufacturers and comment on the two results.

Teamwork

23. Assign one model type of bicycle to each team member. Use time series analysis to examine monthly sales of bicycles by model type. Are there differences in seasonal models (lags) for the different model types? Note, skip track and hybrid models because they have limited sales.

24. Assign one basketball team to each person in the group. Compute the average weekly field goal percentage for the team (total field goals made/total field goals attempted that week). Do a time series analysis by team and compare results obtained by all members in the group.

25. Using the Corner Med database create subgroups for each of the physicians. Examine total number of weekly visits handled by each physician separately and identify any patterns or trends. Compare the results for each physician and against the total.

26. Using the Bakery database, assign one product category to each person. Analyze total daily sales for the chosen category and identify any patterns and trends. Combine the results from each category and comment on the results.

Additional Reading

Bowerman, Bruce L. and Richard T. O'Connell, 1979, *Time Series and Forecasting*, Duxbury: North Scituate Massachusetts. [A good introduction to the statistics of Time Series, particularly ARIMA. Undergraduate level.]

Box, G.E.P., and G.M. Jenkins, 1976, *Time Series Analysis: Forecasting and Control*, Holden-Day, San Francisco. [Classic description of ARIMA. Mathematical.]

Judge, George G., W.E. Griffiths, R. Carter Hill, Helmut Lütkepohl, and Tsoung-Chao Lee, 1985, *The Theory and Practice of Econometrics, second edition*, Wiley: New York. [A classic complete work on econometric theory for those who want to know how to handle problems that arise with regressions and time series introduction. Graduate level.]