Advanced Time Series Analysis
An In-Depth Look at the auto_arima Function and its Constituents

A Whitepaper by Rogue Wave Software
Advanced Time Series Analysis

An In-Depth Look at the `auto_arima` Function and its Constituents

by Rogue Wave Software

© 2014 by Rogue Wave Software. All Rights Reserved

Trademark Information

The Rogue Wave Software name, logo, and IMSL are registered trademarks of Rogue Wave Software, Inc. or its subsidiaries in the US and other countries. All other company, product, or brand names are the property of their respective owners.

IMPORTANT NOTICE: The information contained in this document is subject to change without notice. Rogue Wave Software, Inc. makes no warranty of any kind with regards to this material, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. Rogue Wave Software, Inc. shall not be liable for errors contained herein or for incidental, consequential, or other indirect damages in connection with the furnishing, performance, or use of this material.
# TABLE OF CONTENTS

- **Introduction** ................................................................. 4
- **Background** ..................................................................... 4
- **Model Selection** ............................................................... 5
- **Missing Values** .................................................................. 6
  - Method 1 ............................................................................. 6
  - Method 2 ............................................................................. 6
  - Method 3 ............................................................................. 6
  - Method 4 ............................................................................. 7
- **Seasonal Variations** ............................................................ 7
- **Outliers** ............................................................................ 7
- **Applied Analysis to Finance** .............................................. 9
  - Missing Values ................................................................... 9
  - Seasonality ....................................................................... 10
  - Outliers ........................................................................... 11
  - Auto_ARIMA .................................................................... 12
  - Case 1 ............................................................................... 13
  - Case 2 ............................................................................... 13
  - Case 3 ............................................................................... 14
  - Case 4 ............................................................................... 15
  - Case 5 ............................................................................... 15
  - Case 6 ............................................................................... 16
- **Comparisons** ................................................................... 17
- **Conclusion** ....................................................................... 18
- **About IMSL** .................................................................... 18
- **About Rogue Wave Software** ............................................ 18
- **References** ..................................................................... 18
Introduction

Time series analysis is used by many industries in order to extract meaningful statistics, characteristics, and insights. Businesses use time series to improve business performance or mitigate risk in applications such as finance, weather prediction, cell tower capacity planning, pattern recognition, signal processing, and engineering.

This paper describes an advanced forecasting routine for time series analysis using the IMSL® routine, auto_arima. Auto_arima uses an autoregressive integrated moving average (ARIMA) model. The ARIMA model is powerful and flexible, and is appropriate for many real-world time series situations, but because of its flexibility, the ARIMA model is sometimes difficult to configure properly. IMSL’s auto_arima automates many of the configuration tasks so the ARIMA model’s effectiveness is more readily attained in practice. Auto_arima also has provisions for outliers and missing values so users do not need to pre-process the data for such issues. The auto_arima function is used for many time series problems, such as sales forecasting, commodity pricing, stock market predictions, and more.

The main application described in this paper is a routine that combines a number of techniques culminating in a function that estimates missing values, incorporates the effects of outliers, performs seasonality adjustments, selects the best input parameters to an \[\text{ARIMA}(p,d,q)\], and forecasts future values. The individual techniques of dealing with missing values, outliers, and seasonality can be used independently of the auto_arima function and are described separately after the forecasting model is illustrated.

Background

The ARIMA model was introduced by Box and Jenkins (1976)[1] and requires three different parameters: the autoregressive parameters, \(p\), the number of differencing passes, \(d\), and the moving average parameters, \(q\). The standard notation, introduced by Box and Jenkins, summarizes the models as \(\text{ARIMA}(p,d,q)\). For example, a model described as \(\{0, 1, 2\}\) means that it contains zero autoregressive parameters and two moving average parameters, which were computed for the series after it was differenced once.

The \(\text{ARIMA}(p,d,q)\) model is an extension of the Autoregressive Moving Average model \[\text{ARMA}(p,q)\], which is a combination of Autoregressive and Moving Average models \[\text{AR}(p)\] and \[\text{MA}(q)\] respectively. An \(\text{AR}(p)\) model is an autoregressive model that uses past values of a time series to predict future values. The model is referred to as “autoregressive” because future values are functions of past values of the series; i.e., “auto” meaning “self” as opposed to “automatic.” In contrast, an \(\text{MA}(q)\) model (Moving Average) assumes the series is based on a lagged white noise process from previous observations. The combination of a moving average process with a linear difference equation generates an autoregressive moving average model, the popular \(\text{ARMA}(p,q)\) model, used for general forecasting.

An ARIMA model extends an ARMA model an additional step by adding the ability to model non-stationary time series using differencing. An ARMA representation for a time series requires a stationary time series. An ARIMA model changes a non-stationary time series to a stationary series by using repeated seasonal differencing. The number of differences, \(d\), is input to the fitting process. Since the forecast estimates are based on the differenced time series, an integration step is required so that the forecasted values are compatible with the original data. This integration step accounts for the “I” in “ARIMA.”
Note that for each of these models, the user normally must specify the number of autoregressive and moving average parameters $p$ and $q$, respectively. The method of choosing values for $p$ and $q$ requires an expert analysis of the autocorrelations and partial autocorrelations for the series. Even then, finding the proper model is sometimes an iterative technique where values are chosen and the model fitted using a partial input set, where known future values are used to evaluate the model’s forecast. The input parameters that give the best results are then chosen. Once the input parameters have been estimated, they are used to forecast of values beyond the end of the time series.

The IMSL function `auto_arima`, uses the term “auto” as in “automatic.” Calling the function requires very little, if any, pre-processing of the time series by the user. The function automatically estimates missing values, selects the best values for $p$ and $q$, performs seasonal differencing, detects outliers and produces forecasts. Because a diligent user may be interested in the underlying time series outlier-free series as well as forecasted values the outlier-free series, both the outlier-free series and associated forecasts are available as output.

**Model Selection**

The `auto_arima` function can be invoked using one of six different cases. The case selected is specified by the user based upon their objectives and analyses of the input time series, including its autocorrelations, seasonality and non-stationary behavior. The six cases are described in **TABLE 1**. The details on how users specify which case to invoke are specified in the routine’s API description.

<table>
<thead>
<tr>
<th>Case</th>
<th>Model</th>
<th>Parameter Selection</th>
<th>Non Stationary Adjustment</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>AR($p$)</td>
<td>Automatic</td>
<td>No</td>
</tr>
<tr>
<td>2</td>
<td>AR($p,s,d$)</td>
<td>Automatic</td>
<td>Yes</td>
</tr>
<tr>
<td>3</td>
<td>ARMA($p,q$)</td>
<td>Automatic</td>
<td>No</td>
</tr>
<tr>
<td>4</td>
<td>ARMA($p,d,q$)</td>
<td>Automatic</td>
<td>Yes</td>
</tr>
<tr>
<td>5</td>
<td>ARIMA($p,d,q$)</td>
<td>Specified</td>
<td>No</td>
</tr>
<tr>
<td>6</td>
<td>ARIMA($p,d,q$)</td>
<td>$p, q$ specified, $d, s$, automatic</td>
<td>Yes</td>
</tr>
</tbody>
</table>

**TABLE 1. Six Methods for Specifying Automatic Model Selection in `auto_arima`**

The model choice is up to the user and is determined by the `auto_arima` function at runtime by users choice of input parameters. Details regarding each method become more apparent in the applied example, but can be summarized briefly. In Case 1, the method searches for the best AR($p$) using a minimum AIC (Akaike's information criterion) method for a range of $p$ up to and including an input parameter $max\_lag$, which is the largest offset used in autoregressing the time series. This method uses the fact that given a sufficiently large $p$, any stationary time series can be fitted using an AR($p$) model. It is also based upon the widespread popular use of AIC as a yardstick for measuring the quality of a time series fit. Case 2 is similar except that optimum values for $s$ and $d$ for fitting any non-stationary or seasonal trends are also determined. Thus Case 2 is an AR model that also accounts for seasonality and non-stationary time series.
Case 3 is an ARMA($p,q$) model where optimal values for $p$ and $q$ are selected from a search of all combinations of $p$ (up to $\text{max}_\text{lag}$) and input values of $q$ supplied by the user. Case 4 extends Case 3 to account for non-stationary seasonality effects, effectively then computing an ARIMA($p,d,q$) model with automatic parameter determination.

Cases 5 and 6 allow users to specify specific values for $p$ and $q$. Case 5 accepts user-specified values for the input parameters in an ARIMA($p,d,q$) model, allowing no searching by the function. Finally, Case 6 accepts user values for $p$ and $q$ while conducting a grid search of all possible combinations for input values of $s$ and $d$. For users familiar with the ARMA($p,q$) model, this case allows the easiest extension to an ARIMA($p,d,q$) model.

Missing Values

Missing values may appear in time series for a number of reasons including recording failures or previous attempts at cleaning the data. Missing values are represented in the time series by a sequence gap in the time values. There are four methods for estimating values of missing data with the function `estimate_missing`. Note that there is error checking in the routine so that if there is not enough data or problems with input parameters for the methods three or four, the fallback is to use the first method, which can always be computed for any data. In all of these methods, all previously observed data, including previously estimated missing values, are used in computing the next estimate. No future data values are incorporated into this estimate.

**Method 1**

The first method of estimating missing values is the most straightforward. The missing observations are replaced with the median value of the time series of observed values up to the point of the missing value. This is a very simple and fast method, but is limited to use with stationary ergodic time series. Any level shifts and outlier data points may influence the median value to an extent that this method does not provide adequate estimates.

**Method 2**

Missing values may be estimated using a cubic spline interpolation method. This is the most popular of the spline interpolation methods as it is effective at creating smooth transitions across missing values. For this method, the cubic spline satisfies the “not-a-knot” endpoint condition, where the third derivative of the third order polynomial continues at the endpoints (as opposed to the “natural” condition where the second derivative is set to zero for the endpoints).

**Method 3**

Additionally, first order autoregressive model, AR(1), may be chosen to estimate the missing values. For this method, all of the data in the time series observed prior to the missing value are used to forecast the missing values. The first point of this forecast is used as the estimate for the first missing value.
Method 4

Finally, an estimate for a missing value may be computed using an AR\((p)\) model. This method begins by determining the best choice for the parameter \(p\) (using values through the \(\text{max}_\text{lag}\)) using the IMSL function \text{auto\_uni\_ar} and then computes a forecast based on that chosen value with \text{arma}. As with the third method, all of the observed and previously estimated values before the current missing observation are used to compute the forecast. This is the method used by \text{auto\_arima}.

Seasonal Variations

Financial and economic time series are often subjected to seasonal variations due to a myriad of factors from actual planetary seasonality of data to normal business cycles to higher frequency oscillations. The technique used to remove such variance is known as seasonal differencing. The number of seasons in the time series is required input and data sent to this function must be free of missing values.

The IMSL function \text{seasonal\_fit} requires two input parameters: the number of differences to use in the model, \(d\), and the number of seasons to test, \(s\). Both of these parameters can be input as an array of possible values, with the best fit values determined automatically and used in the model. Additionally, the best value for the \(p\) parameter to the AR\((p)\) model can be determined automatically from an input value for \text{max}_\text{lag}. Note that without this determination of the optimal values of \(d\) and \(s\), one could call \text{difference} directly; the novelty in \text{seasonal\_fit} is in wrapping \text{difference} without requiring explicit values for \(d\) and \(s\).

An AR\((p)\) model (identified using a minimum AIC method) is used to evaluate the best differenced output time series for the combinations of input parameters \(d\) and \(s\).

Outliers

Outliers are observations that could create spurious results that are not consistent with the rest of the time series. The source of outliers may be real disruptive events like natural disasters or war and other political upheavals or may be artificial due to data entry or recording errors. While some types of outliers are sometimes apparent through visual inspection of a time series, such visual inspections can lead to personal bias, and are simply not tractable for analysis of large sets of observations.

The \text{auto\_arima} function uses the Chen-Liu algorithm (Chen & Liu, 1993\cite{2}), a joint estimation method, to identify outliers. Outputs include the number of outliers identified, the time of the observation, and the classification of the outlier. Outliers are classified into one of five categories based on the standardized statistic for each outlier type. The five classifications are outlined in TABLE 2.
### Outlier Classifications

<table>
<thead>
<tr>
<th>Outlier Class</th>
<th>General Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IO – Innovational Outlier</td>
<td>Innovational outliers persist. That is, there is an initial impact at the time the outlier occurs. This effect continues in a lagged fashion with all future observations.</td>
</tr>
<tr>
<td>AO – Addictive Outlier</td>
<td>Additive outliers do not persist. As the name implies, an additive outlier affects only the value observed at the time of the outlier. Hence additive outliers have no effect on future forecasts.</td>
</tr>
<tr>
<td>LS – Level Shift</td>
<td>Level shift outliers persist. They have the effect of either increasing or lowering the mean of the series starting at the time the outlier occurs. This shift in the mean is abrupt and permanent.</td>
</tr>
<tr>
<td>TC – Time Change</td>
<td>Time change outliers persist and are similar to level shift outliers with one major exception. Like level shift outliers, there is an abrupt change in the mean of the series at the time this outlier occurs. However, unlike level shift outliers, this shift is not permanent. The change outlier gradually decays, eventually bringing the mean of the series back to its original value. The rate of this decay is modeled using an input parameter delta. The default value of delta=0.7 is recommended for general use by Chen &amp; Liu (1993).</td>
</tr>
<tr>
<td>UI – Unable to Identify</td>
<td>If an outlier is identified as the last observation, then the algorithm is unable to determine the outlier’s classification. For forecasting, a UI outlier is treated as an IO outlier. That is, its effect is lagged into forecasts after the outlier occurs.</td>
</tr>
</tbody>
</table>

**TABLE 2. Outlier Classifications**

Except for additive outliers, the effect of an outlier persists to following observations. Forecasts produced based on a time series containing such events must take this persistence into account.

A number of inputs are required including a critical value used to identify a point as a possible outlier and a convergence parameter used to determine when adjustment of outliers is sufficient. The critical point, if chosen too low, could lead to spurious outliers or too many outliers being detected. As a point of reference, Chen & Liu (1993) recommend a value of 3.0 for data series with between 100 and 200 observations. The value should be considered carefully and verified through analysis of the residual pattern.

Any model can be selected to identify outliers. Data to be analyzed for outliers must be seasonally adjusted prior to outlier analysis and also must not contain any missing values.
Applied Analysis to Finance

Time series are ubiquitous in the finance industry. From historical data of market indices to daily interest rate fluctuations, the amount of data is often overwhelming. One major issue with such data is the individual time series’ suitability to a specific method of analysis. Many techniques require or assume the input time series to be “well behaved”. That is, the series is stationary, ergodic and free of outliers and missing values. Due to the dynamics of financial markets, such clean data is not likely to be found except for in the smallest subsets, and some officially reported, data adjusted for inflation and seasonality.

The analysis of financial time series often revolves around a singular goal: to make a confident prediction of future behavior based on historical performance. Such forecasts are typically derived from models such as ARCH, GARCH, and ARMA. While many analysts will treat such numerical models as black boxes that accept only their well-conditioned time series as input, the reality is that the models take some specific knowledge of the dynamics of the input to be used effectively. For example, the common GARCH and ARMA($p,q$) models require two parameters $p$ and $q$, the number of autoregressive parameters and the number of moving average parameters, respectively. Finding the proper values is often an iterative manual technique: one re-runs the model until the output is deemed the best (often subjectively). Automatic determination of parameters should be a requirement for models, especially those like ARIMA($p,d,q$), which requires four different input parameters (including $s$, a parameter related to identifying seasonality).

To adapt real financial data to proper time series analysis, it must often be pre-processed. Such pre-processing would involve estimating missing values, removing outliers, and accounting for seasonal variations. Simplistic methods exist for analysts to overcome these obstacles; however, simplistic methods, while easy to program, are rarely the most effective. This document presents methods available in the IMSL Numerical Libraries to meet these demands. Finally, a forecasting ARIMA model is presented. This model is able to automatically determine the best set of values of input parameters by finding the maximal likelihood estimates of each possible combination of input parameters.

Although the data used in the subsequent examples are not strictly financial data, one can easily see the correlation between the referenced data sets and true financial data (stock prices, currencies, etc.).

**Missing Values**

The quarterly average price of gasoline$^{[3]}$ from 1978 through the second quarter of 2003 is used as a test data set for the estimation of missing values. While the data set actually contained all values, three of the 101 observations were removed. Each method described above was used to estimate the missing values. The estimated values are shown in TABLE 3 along with the actual observed value of the original time series; percent difference is shown in TABLE 4.
TABLE 3. Estimated Missing Values for Each Method

<table>
<thead>
<tr>
<th>Point</th>
<th>Method 1</th>
<th>Method 2</th>
<th>Method 3</th>
<th>Method 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>28</td>
<td>1.2250000</td>
<td>0.99687300</td>
<td>1.1816310</td>
<td>1.1575510</td>
</tr>
<tr>
<td>84</td>
<td>1.1870000</td>
<td>0.82449000</td>
<td>1.1136950</td>
<td>1.1136580</td>
</tr>
<tr>
<td>85</td>
<td>1.1870000</td>
<td>0.02395900</td>
<td>1.1160210</td>
<td>1.1159510</td>
</tr>
</tbody>
</table>

TABLE 4. Percent Difference of Estimated Values

<table>
<thead>
<tr>
<th>Point</th>
<th>Method 1</th>
<th>Method 2</th>
<th>Method 3</th>
<th>Method 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>28</td>
<td>3.0</td>
<td>-16.2</td>
<td>-0.6</td>
<td>-2.6</td>
</tr>
<tr>
<td>84</td>
<td>1.10.2</td>
<td>-23.4</td>
<td>3.4</td>
<td>3.4</td>
</tr>
<tr>
<td>85</td>
<td>15.1</td>
<td>-97.7</td>
<td>8.2</td>
<td>8.2</td>
</tr>
</tbody>
</table>

For this particular data, method 3 performs the best overall, with method 4 providing very similar results since method 3 is a special case (p=1) of method 4. Method 2 performs very poorly, underestimating values considerably, while the nearly trivial method of median replacement of method 1 works adequately.

It should be noted that the estimated values might be identified as outliers, depending on the data used to evaluate the estimates. Once missing values have been filled in, there is no special consideration given to them with respect to seasonal differencing, outlier detection, or forecasting. That is, there is no internal information kept by the routine to give less weight or importance to these estimated values.

**Seasonality**

A lot of sales data is seasonal, and automobile sales are a prime example, traditionally peaking in the fall when new models arrive on the showroom floor. The data set for this example is monthly vehicle sales from the United States from January 1971 to December 1991.

For this example, we allow the `seasonal_fit` function to determine the best values for the necessary seasonality parameters `s` and `d`, which were found to be 2 and 1 respectively. The forecast was made with an ARIMA(2,2,2) with a non-stationary component.

The original time series is shown in FIGURE 1 as the black line, with the next eight predicted values in red. The modified time series with the seasonal trend removed appears as the blue line, with the prediction in green. Notice the low frequency trend in the data has been removed in the seasonally adjusted time series, so that the prediction is influenced more dominantly by the higher frequency oscillations.
Outliers

The same gasoline price data[^3] used in the Missing Values section are used in this showcase of outlier identification. In this example, the original time series (thin black line in FIGURE 2) is used to generate a forecast (dashed black line) using an AR(6) model. (This parameter selection was chosen by auto_arima to be the best for this particular data.) The ts_outlier_identification function is then employed to identify any outliers. The input parameters for this function were delta = 0.07, critical = 3.5, epsilon = 0.01 where the delta parameter is the dynamic dampening effect adjustment parameter used in identifying temporary change (TC) outliers; critical is the critical value for identifying outliers in the Chen & Liu algorithm; and epsilon is the convergence parameter used to determine when adjustment of outliers is sufficient.

Each of these is the normal recommended value. The delta parameter may be adjusted if the effects of temporary change outliers are desired to have short or longer effects. Depending on the length of the data set and its variance, one may need to adjust critical to ensure that neither too many nor too few outliers are detected. As one gains familiarity with the use of this algorithm, it is recommended to try different values for critical to get a feel for the different results higher and lower values give. The value for epsilon should always remain near 0.01; slight adjustments may be required if different values of critical are not providing the expected effect for a given time series.

To analyze the effects of outlier removal, the original data set is taken without modification and eight future values are forecasted. Then, with the input parameters described, the time series is passed through the outlier identification algorithm to find and identify any outliers. Another forecast is computed using the same AR(6) model.
For this data set, four points were identified. Two time change (TC) outliers were found at April 1989 and October 1990. An additive outlier (AO) was identified at April 2001. Finally and innovative outlier (IO) was determined to exist at April 2002.

The results of this analysis are shown in FIGURE 2. The original time series is plotted as a thin black line, with its computed forecast a black dashed line. Outliers are marked with an asterisk and labeled with the type of outlier. The time series with the effects of these four points removed is overlaid as a thick blue line in the figure. The forecasted values using the adjusted time series appear as a dotted blue line. Eight points (two years) are forecasted; error estimates have been left out for clarity.

Notice that while the forecasted trends are similar, their magnitude differs by an average of 13% for the eight points. By accounting for outliers in input time series, more accurate forecasts can be obtained. Using the auto_arima function, such outlier identification comes at virtually no cost to the user.

**Auto_ARIMA**

We will now put the auto_arima function itself through the paces, examining the results of each method of usage as outlined in TABLE 1. The data used in this analysis is the price of Western Texas crude oil\(^5\). Data from 2/1986 through 3/2001 are used to forecast values through 5/2003, for which data values are known, but not input into the model. For each case, results are presented graphically, with the original time series as a black line (including the “future” values not input into the model). The twenty-six forecasted values are shown in red with the 95% confidence interval designated by the surrounding blue lines.
Case 1

The first case uses an AR($p$) model with the automatic selection of the number of autoregressive parameters. The function determined the value of $p = 6$ for the input data, and the results are presented in FIGURE 3. The first few forecasted values are quite close to the actual observed values. However, the overall trend for the next two years of data is not very well predicted.

This illustrates the simple fact that long-range forecasts for any stationary series will trend towards the overall mean of the series. In this case, the mean is a little above 20.0 and as shown, the long range forecast is moving towards that value.

For gasoline prices, this is counter-intuitive. That is popular expectation is for gasoline prices to increase over the long term. Historically there have been only a few periods when prices varied randomly around some mean.

If the expectations are for the series to drift upwards or downwards, then the series needs to be differenced to remove this non-stationary trend.

![Figure 3: Case 1, an AR(6) model.](image)

Case 2

Case 2 is another autoregressive model for which seasonal effects can be taken into account. The best parameters for this case are $p=4$, $s=1$ and $d=1$, or an AR(4,0,1) model; see FIGURE 4. The key difference between the fit from this case and the previous is that the series was differenced before fitting. This removes the non-stationary effect of upward trending gasoline prices.

The forecasted values of this model do capture most of the overall trend for the two years of predicted data. However, the initial decrease is not predicted and the large fluctuations fall considerably outside the 95% confidence band.
Case 3

This case is a standard ARMA($p,q$) model where the parameter values are determined automatically. Interestingly, the best fit for the input data is $p=q=0$, which results in a linear prediction as seen in FIGURE 5. Intuitively this prediction is not as good as the other cases, but it does draw a compromise between those who predict prices to increase and those than think they will fall.
Case 4

The fourth case is the first ARIMA model. All the parameters are determined by auto_arima and an adjustment for non-stationary series is possible. The parameters determined for this case is \( p=0, q=2, d=1 \) with a non-stationary component. This ARIMA(0,2,1) model is shown in FIGURE 6. The general trend of the future values is captured. All but the most extreme downward fluctuation falls within the 95% confidence interval, indicating that this model works quite well for this data set.

![FIGURE 6. Case 4, an ARIMA(0,2,1) Model.](image)

Case 5

All of the input parameters must be set by the user for this case. Arbitrarily, input values of \( p=2, q=0, d=2 \) with \( s=1 \) were used to compute the ARIMA(2,2,0) model. Obviously, these were not the best values for representing this series. The results are shown in FIGURE 7. The predicted values greatly overestimate those observed and the confidence interval is so large as to be nearly useless. These are not good values to use for this time series.
Case 6

Finally, this case is a partially specified model. We have set $p=2$, $q=0$, and the function determined that $d=1$ and $s=1$ are the best parameters, resulting in an ARIMA$(2,1,0)$ model with a non-stationary component. The results, shown in FIGURE 8, are quite similar to Case 4, where the model is also very similar except that the $p$ and $q$ values are reversed, providing slightly different weighting to the autoregressive and moving average components of the model. Overall the forecast is quite good.
Comparisons

The predicted values for each of the six models are shown below in TABLE 5 for comparison with the observed data. Total mean square error (MSE) is also provided as a means to quantify model effectiveness for prediction. Note that MSE is a convenient way to evaluate model performance, but certainly not the only way.

<table>
<thead>
<tr>
<th>Date</th>
<th>Observed</th>
<th>Case 1</th>
<th>Case 2</th>
<th>Case 3</th>
<th>Case 4</th>
<th>Case 5</th>
<th>Case 6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Apr-01</td>
<td>27.41</td>
<td>26.64</td>
<td>26.96</td>
<td>29.42</td>
<td>27.31</td>
<td>31.55</td>
<td>27.11</td>
</tr>
<tr>
<td>May-01</td>
<td>28.64</td>
<td>26.59</td>
<td>27.32</td>
<td>29.42</td>
<td>27.42</td>
<td>33.53</td>
<td>27.60</td>
</tr>
<tr>
<td>Jun-01</td>
<td>27.6</td>
<td>25.85</td>
<td>27.10</td>
<td>29.42</td>
<td>27.53</td>
<td>34.87</td>
<td>27.79</td>
</tr>
<tr>
<td>Jul-01</td>
<td>26.45</td>
<td>25.61</td>
<td>27.40</td>
<td>29.42</td>
<td>27.64</td>
<td>36.65</td>
<td>27.86</td>
</tr>
<tr>
<td>Aug-01</td>
<td>27.47</td>
<td>25.49</td>
<td>27.73</td>
<td>29.42</td>
<td>27.75</td>
<td>38.56</td>
<td>27.97</td>
</tr>
<tr>
<td>Sep-01</td>
<td>25.88</td>
<td>25.10</td>
<td>27.79</td>
<td>29.42</td>
<td>27.86</td>
<td>40.34</td>
<td>28.10</td>
</tr>
<tr>
<td>Oct-01</td>
<td>22.21</td>
<td>24.84</td>
<td>27.97</td>
<td>29.42</td>
<td>27.97</td>
<td>42.25</td>
<td>28.22</td>
</tr>
<tr>
<td>Nov-01</td>
<td>19.67</td>
<td>24.61</td>
<td>28.15</td>
<td>29.42</td>
<td>28.08</td>
<td>44.25</td>
<td>28.34</td>
</tr>
<tr>
<td>Dec-01</td>
<td>19.33</td>
<td>24.31</td>
<td>28.27</td>
<td>29.42</td>
<td>28.19</td>
<td>46.26</td>
<td>28.47</td>
</tr>
<tr>
<td>Jan-02</td>
<td>19.67</td>
<td>24.06</td>
<td>28.42</td>
<td>29.42</td>
<td>28.30</td>
<td>48.33</td>
<td>28.59</td>
</tr>
<tr>
<td>Feb-02</td>
<td>20.74</td>
<td>23.83</td>
<td>28.58</td>
<td>29.42</td>
<td>28.41</td>
<td>50.48</td>
<td>28.71</td>
</tr>
<tr>
<td>Mar-02</td>
<td>24.42</td>
<td>23.58</td>
<td>28.72</td>
<td>29.42</td>
<td>28.52</td>
<td>52.68</td>
<td>28.83</td>
</tr>
<tr>
<td>Apr-02</td>
<td>26.27</td>
<td>23.36</td>
<td>28.87</td>
<td>29.42</td>
<td>28.62</td>
<td>54.93</td>
<td>28.96</td>
</tr>
<tr>
<td>May-02</td>
<td>27.02</td>
<td>23.16</td>
<td>29.02</td>
<td>29.42</td>
<td>28.73</td>
<td>57.24</td>
<td>29.08</td>
</tr>
<tr>
<td>Jun-02</td>
<td>25.52</td>
<td>22.97</td>
<td>29.17</td>
<td>29.42</td>
<td>28.84</td>
<td>59.61</td>
<td>29.20</td>
</tr>
<tr>
<td>Jul-02</td>
<td>26.94</td>
<td>22.79</td>
<td>29.32</td>
<td>29.42</td>
<td>28.95</td>
<td>62.03</td>
<td>29.32</td>
</tr>
<tr>
<td>Aug-02</td>
<td>28.38</td>
<td>22.62</td>
<td>29.47</td>
<td>29.42</td>
<td>29.06</td>
<td>64.51</td>
<td>29.45</td>
</tr>
<tr>
<td>Sep-02</td>
<td>29.67</td>
<td>22.46</td>
<td>29.62</td>
<td>29.42</td>
<td>29.17</td>
<td>67.05</td>
<td>29.57</td>
</tr>
<tr>
<td>Oct-02</td>
<td>28.85</td>
<td>22.32</td>
<td>29.77</td>
<td>29.42</td>
<td>29.28</td>
<td>69.64</td>
<td>29.69</td>
</tr>
<tr>
<td>Nov-02</td>
<td>26.27</td>
<td>22.18</td>
<td>29.92</td>
<td>29.42</td>
<td>29.39</td>
<td>72.29</td>
<td>29.81</td>
</tr>
<tr>
<td>Dec-02</td>
<td>29.42</td>
<td>22.06</td>
<td>30.07</td>
<td>29.42</td>
<td>29.50</td>
<td>75.00</td>
<td>29.94</td>
</tr>
<tr>
<td>Jan-03</td>
<td>32.94</td>
<td>21.94</td>
<td>30.22</td>
<td>29.42</td>
<td>29.61</td>
<td>77.77</td>
<td>30.06</td>
</tr>
<tr>
<td>Feb-03</td>
<td>35.87</td>
<td>21.83</td>
<td>30.37</td>
<td>29.42</td>
<td>29.72</td>
<td>80.59</td>
<td>30.18</td>
</tr>
<tr>
<td>Mar-03</td>
<td>33.55</td>
<td>21.72</td>
<td>30.52</td>
<td>29.42</td>
<td>29.83</td>
<td>83.47</td>
<td>30.30</td>
</tr>
<tr>
<td>Apr-03</td>
<td>28.25</td>
<td>21.63</td>
<td>30.67</td>
<td>29.42</td>
<td>29.94</td>
<td>86.41</td>
<td>30.43</td>
</tr>
<tr>
<td>May-03</td>
<td>28.14</td>
<td>21.54</td>
<td>30.82</td>
<td>29.42</td>
<td>30.05</td>
<td>89.40</td>
<td>30.55</td>
</tr>
<tr>
<td>MSE</td>
<td>1.04</td>
<td>0.97</td>
<td>1.27</td>
<td>0.94</td>
<td>41.78</td>
<td>1.01</td>
<td></td>
</tr>
</tbody>
</table>

TABLE 5. Comparison of Models

Case 4 performs the best using the MSE criteria. Examining the results graphically by comparing Figures 3 through 8, one comes to a similar conclusion. Case 3, where no real prediction was made, and Case 5, where the user-supplied parameters were not optimal, show poor performance. Any of the other cases could be considered a reasonable forecasting model, with Case 4 performing the best.
Conclusion

A detailed summary of the `auto_arima` function and an example application of the function and all of its constituent pieces have been presented. Using example data sets, the estimation of missing values, adjustment for seasonality, and identification of outliers are discussed individually. Finally, the `auto_arima` function is applied to sample data for each usage model; not surprisingly, the fully automatic method (Case 4) performs the best as determined graphically (see FIGURE 6) and quantitatively using a mean square error estimate (see TABLE 5). Utilizing the IMSL Numerical Library function and the expertise of Rogue Wave Software’s Services Team, an organization can integrate Auto_ARIMA to create an optimal and custom forecasting solution for their situation. IMSL features time series modeling algorithms including Automatic ARIMA, Regression ARIMA, and AUTO-PARM to detect structural breaks in time series.

About IMSL

The IMSL Numerical Libraries are available for a wide range of computing platforms, and offer robust, scalable, portable, and high-performing analytics, allowing developers to focus on their domain of expertise and reduce development time. With IMSL, businesses and organizations realize a lower total cost of ownership, and improve quality and maintainability.

About Rogue Wave Software

Rogue Wave Software, Inc. is the largest independent provider of cross-platform software development tools and embedded components. Rogue Wave’s proven technical solutions simplify the growing complexity of building and testing quality software code. Rogue Wave customers improve software quality and ensure code integrity, while shortening development cycle times, and include industry leaders in the Global 2000 as well as leading government institutions and universities. For more information, visit http://www.roguewave.com.

References


