International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2013): 6.14 | Impact Factor (2013): 4.438

Simple Steps for Fitting Arima Model to Time Series Data for Forecasting Using R

Alexander Kasyoki [1]

[1]Taita Taveta University College, Department of Mathematics and Informatics

Abstract: ***Time series deals with data that has been recorded or observed over time. These data may need to be analyzed to come up with conclusions and meet the objectives intended by the researcher. A time series may be expressed as an additive model of its components which includes the seasonal, the cyclic, the trend and irregular components. When time series data is analyzed it becomes very key in forecasting or prediction of future time series values, in control of machines among others. In this study it has been noted that though most researchers may be in a position to collect time series data, it is a challenge in analyzing it since some of the steps they are aware of may be complex and not straight forward. This then implies that analysis of time series data needs a great understanding and knowledge of the procedure and the models that can be useful in meeting the researcher's objectives. This writing discusses the application of ARIMA model in analyzing time series data in a sophisticated and interactive package known as R. The procedure is vividly stated and explained with aid of some R commands and illustrations. It is expected that the researchers or students who take statistical projects in this area will greatly benefit from this work.***

Keywords: time series, ARIMA, forecasting, stationary

1. Introduction

There are several methods of forecasting. Among them is the exponential smoothing method which does not take the correlations between successive time series values into consideration. ARIMA model can make a better prediction since it takes into account the correlation between data. This aspect therefore assumes that the irregular component of time series can take non-zero autocorrelation. The ARIMA model is usually defined FOR STATIONARY time series. A stationary time series is the one whose statistical properties such as mean, variance, autocorrelation etc are all constant over time. If the time series is not stationary, then it has to be transformed to a stationary time series using any appropriate transformation technique, for instance, differencing, that is determine the differencing order d which makes it stationary. Differencing removes the trend component of a time series leaving the irregular component.

2. Background Work

Related work has been noted as discussed by several other researchers. Their work touches on time series data analysis in general with use of several models. It is really a recommendable work but its complexity in outlining the steps followed in analysis from the grass root level can be said to be a discriminating factor on fresh and young researchers who might be in search of relevant know how on modeling their time series problems. Avril Coghlan in her work "A little book of R for time series"[1], discusses about the installation of R and time series analysis using the same package. Although it is discussed on how to go about this analysis, the steps are not clearly stated and the work is too much detailed implying that incase one is interested in the procedure only, he has then to go through all the writing to really identify and extract the steps.

The engineering statistics handbook discusses the techniques used in time series modeling and analysis. Supposing one has a prior knowledge of these modeling techniques (may include: Box-Jenkin ARIMA models, Box-Jenkins multivariate models, Holt-winters exponential smoothing) and is interested in applying a specific model to analyze his data, there is indeed a need to know how to get his data ready for using such a model to accomplish his objectives which may include prediction purpose among others. Dr.Gavin Shaddick, in his discussion on using R[2] reiterates the fundamental knowledge needed in using this package though in broad terms but lacks specific procedure required for time series data analysis.

Oleg Nedadic and Walter Zucchin in their work "Statistical analysis with R"[3] ,[4] outlined on linear model fitting using R, time series analysis at some considerable length. They explained the decomposition of a time series, employment of exponential smoothing technique in time series data transformation and ARIMA-MODEL fitting. However, this can be said to be a small portion of what one expects when the interest is to quickly get an hint of time series data analysis.

The focus therefore is to fill these gaps and give a straight forward procedure which is brief without loss of the important points to note in time series analysis. This would therefore outline how to fit ARIMA model to time series data with the aim of forecasting, in a simple, self-explaining and sufficient approach, with the assumption that the researcher has some knowledge of R statistical package. It is also important to note that ARIMA MODEL has been chosen since it is one of reliable and simple models which can be used for prediction purposes. Other models could however be applied but gives results with a lower precision.

3. Procedure

i. Assume you have stored your data in a particular file, then you **need to read it to time series**. This is possible by assigning your data a variable. For instance suppose your data is in CSV file format then you may use the command;

< variableName <- read.csv("C:/fileName.csv")

ii. **Store your variable in a time series object by using ts() function, for instance**
> variableNametimeseries <- ts(variableName)

iii. If the time series data you have collected is for regular intervals that are less than a year, you may **specify the number of times that the data was collected using the frequency parameter in ts function, ts()**. For example for monthly time series data you can set the frequency = 12, for quarterly time series data set the frequency = 4 and so on

iv. **You can also specify the first year that the data was collected,** and the first interval in that year by using the 'start' parameter in the ts() function. For example, if the first data point corresponds to the second quarter of 1990, you would set start=c(1990,2).
>variableNametimeseries<-ts(variableName,frequency=12, start=c(1990,2))

v. **Now you can plot your time series data**. Investigate the random fluctuations for your time series data. If the fluctuations for your time series data are roughly constant over time you may need to use an additive model. An additive model is not appropriate for describing a time series, if the size of the seasonal fluctuations and random fluctuations seem to increase with the level of the time series. Thus, we may need to transform the time series in order to get a transformed time series that can be described using an additive model.
> plot.ts(variableNametimeseries)

vi. **Transform your time series data if need be**. In many situations, it is desirable or necessary to transform a time series data set before using the sophisticated methods for the following reasons:

- Almost all methods assume that the amount of variability in a time series is constant across time.
- Many times we would like to study what is left in a data set after having removed trends (low frequency content) or cycles in the data.

Techniques for transforming data:

- Power transformation- This involves taking the (square root, cube root, log, etc) of the time series data to stabilize its (time series) over time
- Dividing seasonal standard deviation - Sometimes with data observed periodically (hourly, daily, monthly, etc), variability may vary for different periods; for example, there may be more variability on Aprils than on September, and so on. When this happens, it is often useful to calculate the standard deviation for each of the different periods and then for example, divide each April by the standard deviation of all the Aprils, the Septembers by the standard deviation of the Septembers, and so on (notice that dividing a set of any numbers by their standard deviation results in the standard deviation of the new set of numbers being equal to one).
- Subtracting seasonal means - One way to remove cycles in data observed periodically is to calculate the sample means of each of the periods (hours or days, for example) and then subtract them from the corresponding period (subtract the mean of the Aprils from Aprils, that of September from each September and so on).
- The differencing technique –This is commonly used in fitting ARIMA models as discussed above. It is therefore our focus since we are dealing with such a model. The ARIMA model takes three parameters p, d and q that is, ARIMA (p,d,q) where p is the order of AR, d the differencing order and q is the order of MA.

You can difference a time series using the "diff()" function in R.
> variableNametimeseriesdiff1 <- diff(variableNametimeseries, differences=1)

> plot.ts(variableNametimeseriesdiff1)

This process is iterative and therefore we move on by changing the differences until you time series is stationary

vii. **Next you need to determine the values of p and q for the ARIMA model.** To do this, you usually need to examine the autocorrelogram and partial correlogram of the stationary time series. To plot a correlogram and partial correlogram, we can use the "acf()" and "pacf()" functions in R, respectively. To get the actual values of the autocorrelations and partial autocorrelations, we set "plot=FALSE" in the "acf()" and"pacf()" functions.

To plot the correlogram and partial correlogram for lags 1-20 for the once differenced time series of the variableNametimeseriesdiff1, and get the values of the correlations and partial autocorrelations, we use "acf()" and "pacf()" function, by typing:

>acf(variableNametimeseriesdiff1, lag.max=20)
>acf(variableNametimeseriesdiff1,lag.max=20,plot=FALSE)
> pacf(variableNametimeseriesdiff1, lag.max=20) # plot a partial correlogram
>pacf(variableNametimeseriesdiff1,lag.max=20, plot=FALSE)

Here we investigate the lag after which the autocorrelogram or the partial correlogram becomes zero or tends to zero (this gives us the values of q and p respectively). In simple terms this is the last lag that falls out of the significance boundary. Out of the two plots for the ACF and the PACF , you definitely come up with two ARMA models for your difference, d:
An ARMA(p, 0) ; an autoregressive model of order p, where p is determined from the plot of PACF and in this case the autocorrelogram is said to tail off to zero
An ARMA(0, q); a moving average model of order q, where q is determined from ACF plot and in this case the partial autocorrelogram is said to tail off to zero

Decision on the model to choose
The model with the fewest parameters is always the best

If p<q , the model becomes ARMA(P,0)
If q<p, the model becomes ARMA(0,q)
The original time series model is then modeled using the ARMA model chosen for example suppose we choose ARMA(P,0) then the ARIMA model is stated as ARIMA(p,d,0), otherwise, ARIMA(0,d,q)

viii. Estimating the parameters

Once you have selected the ARIMA model, the next step is to estimate the parameters which can then be used for forecasting. You can estimate these parameters using "arima()" function in R, for instance

```
fit<-arima(data,order =c(p,d,q))
```

ix. Forecasting

After determining the parameters you can then predict future values for your time series using predict() function in R Package. Specify the period for which you want the forecast using the n.ahead parameter in predict function. For instance to predict values for our data for the next 5 years, we may use the command.

```
data.pred<-predict(fit, n.ahead=5)
```

x. Plot the forecasted values

One may use "plot. forecast"function , for example;

```
> library("forecast") # load the "forecast" R library
 > plot.forecast(data.predict)
```

xi. Investigate whether the forecast errors of an arima model are normally distributed with mean zero and constant variance, and whether there are correlations between successive forecast errors.

You make the correlogram of the forecast errors for your ARIMA model and perform the Ljung-Box test for lags you specify. For illustration purpose, **suppose** we want to carry out investigation for **data.pred** for 20 lags and then we get results as follows;

```
> acf(data.pred$residuals, lag.max=20)
>Box.test(data.pred$residuals, lag=20, type="Ljung-Box")
Box-Ljung test
data: data.pred$residuals
X-squared = 13.5844, df = 20, p-value = 0.851
```

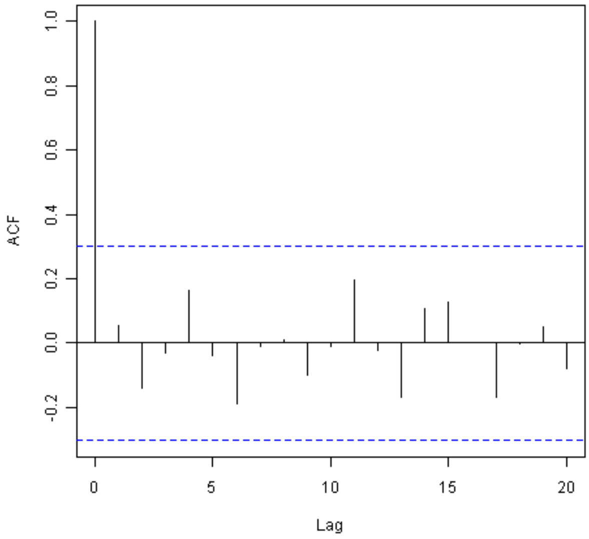
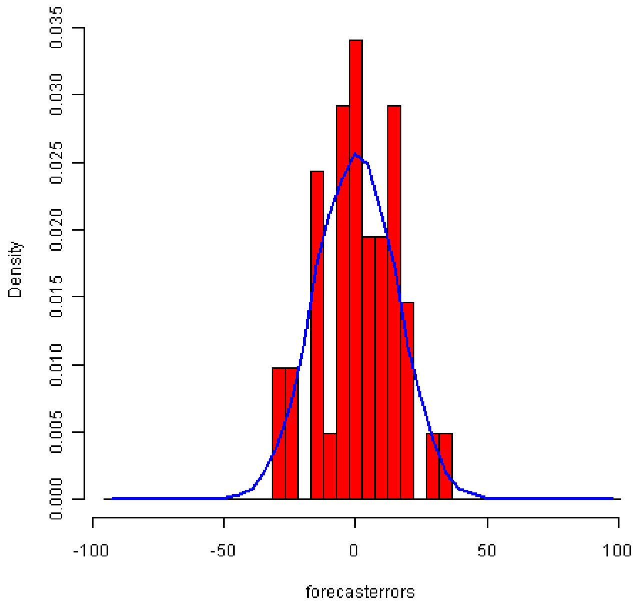


Figure 1: ACF for data .pred

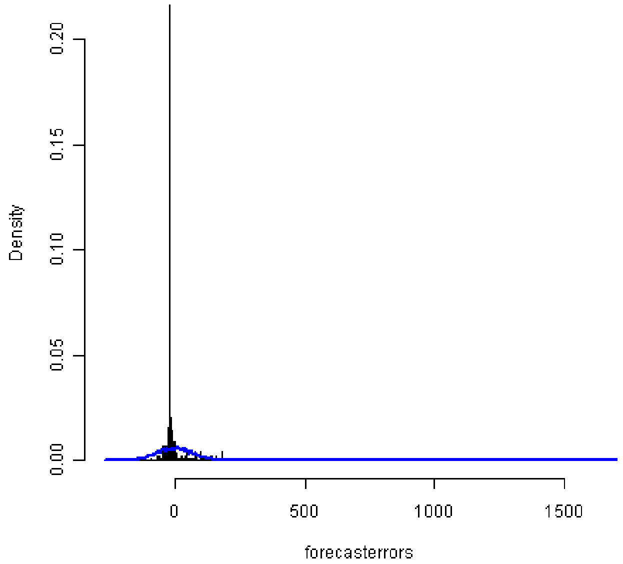
Since the correlogram shows that none of the sample autocorrelations for lags 1-20 exceed the significance bounds, and the p-value for the Ljung-Box test is 0.9, we can conclude that there is very little evidence for non-zero autocorrelations in the forecast errors at lags 1-20.

To investigate whether the forecast errors are normally distributed with mean zero and constant variance, we can make a time plot and histogram (with overlaid normal curve) of the forecast errors:

```
> plot.ts(data.pred$residuals) # make time plot of forecast errors
> plotForecastErrors(data.pred$residuals)# make a histogram
```



Case 1: Forecast errors normally distributed with mean zero



Case 2: Forecast errors skewed to right with negative mean

3.1 Explanation

Case 1 gives the kind of results we would expect if indeed our forecast errors are normally distributed with mean zero.

In Case 2 it is clear that the forecast errors have non-zero mean, contrary to our expectation. This may lead to a conclusion that even if the forecast errors may have a constant variance, some of the predicted values are negative. This therefore means that the ARIMA model stated is not sufficient for our prediction and therefore it can be improved upon to give a better prediction.

4. Conclusion

The proposed algorithm/procedure is simple and time saving since it is straight forward.

5. Recommendations

A variety of R commands are available for performing a particular task and therefore one should not only depend on the commands used in this paper but also make effort to be conversant with other commands which are equally applicable.

References

[1] Avril Coghlan, “A little Book of R For Time Series” release 0.2, January 2014.
[2] Dr.Gavin Shaddick, “using R (with applications in time series analysis)”, January 2004.
[3] Oleg Nedadic,Walter Zucchin, “Statistical analysis with R”, September 2004.
[4] Oleg Nedadic,Walter Zucchin, “Time Series Analysis with R-Part I”, 2004.
[5] Cheng-Der Fur “Financial Time Series,Topic: ARMA and Time series Modelling”, institute of Statistical Science, Academia Sinica,2003.
[6] Robert H. Shumway, David S. Stoffer, “Time Series Analysis and Its Applications”, EZ-Third edition, version: 20140526170200,2003.
[7] Akaike,H.“Fitting autoregressive models for prediction”, Ann. Inst. Stat.Math., 21, 243-247, 1969.
[8] Box, G.E.P. and G.M. Jenkins, “Time Series Analysis, Forecasting, and Control”, Oakland, CA: Holden-Day.1970

Author Profile

Alexander Kasyoki received B.S. degree in Mathematics and Computer Science from Taita Taveta University College in 2013. He is currently pursuing a M.S degree in Applied Statistics at JKUAT, Kenya.