MARCH 16, 2016

TIME SERIES ANALYSIS:

AUTOMOBILE SALES DATA

TAHA HAMID
SATYAN RAO

Table of Contents

NON-TECHNICAL SUMMARY

The automotive industry is one of the most prominent economic sectors in the U.S. Economy. Retail car sales occupy one-third of the total U.S. Manufacturing sales. In order to understand the U.S. Economy, it is important to see dynamics of the retail car sales over time. It is often said that the demand of automobiles has some patterns in time series. The patterns can be associated with factors such as innovations in auto technology, manufacturers' rebates, political support, growth of the economy, etc.

The data set for this project consists of monthly number of sales of automobiles in thousands of units and was extracted from St. Louis Federal Reserve. The data set consists of 480 observations from January 1976 to December 2015. The data indicates that there has been an increase in the number of units sold from 1976 till 2015. In addition, the economic condition of the country tends to influence the sales. Furthermore, the data also exhibits seasonal behavior, which means that in certain month the number of units of automobile sold is greater than the other months. For instance, May – August experiences the highest number of automobile sales every year and January is the month which is the slowest for the automobile sales.

Based on the information observed in the data we were able to create a model, which helps us to forecast the number of automobiles that will be sold in 2016. The power of the model to forecast is accurate as the observed value and the predicted values are pretty close to each other. For instance, if the model forecasts that 17.3 Million Units will be sold in the next month, the difference between the forecast and the actual value will be somewhere around half a million units, which is pretty reasonable.

TECHNICAL SUMMARY

Overview of Data

This project analyzed the given data of Automobiles Sales Data to create a model that can be used to predict the number of units of automobile sold in the United States. The data set comprises of the monthly automobiles units sold in the United States. The data extracted is from January 1976 to December 2015 and included a total of 480 observations. The source of the data was St. Louis Federal Reserve.

Exploratory Analysis

The exploratory analysis helps us to identify the anomalies within the data. The initial first step of the exploratory analysis is to check the distribution of Time Series. In order to investigate about the distribution, histogram and Normal QQ Plot are considered to be an effective indicator. The histogram displays that the distribution of Time Series is normally distribution. The same result is supported by the QQ Plot. Furthermore, Jarque-Bera test of normality was also performed to confirm the results. The results from the test shows that the null hypothesis cannot be rejected and conclude that the distribution is normal.

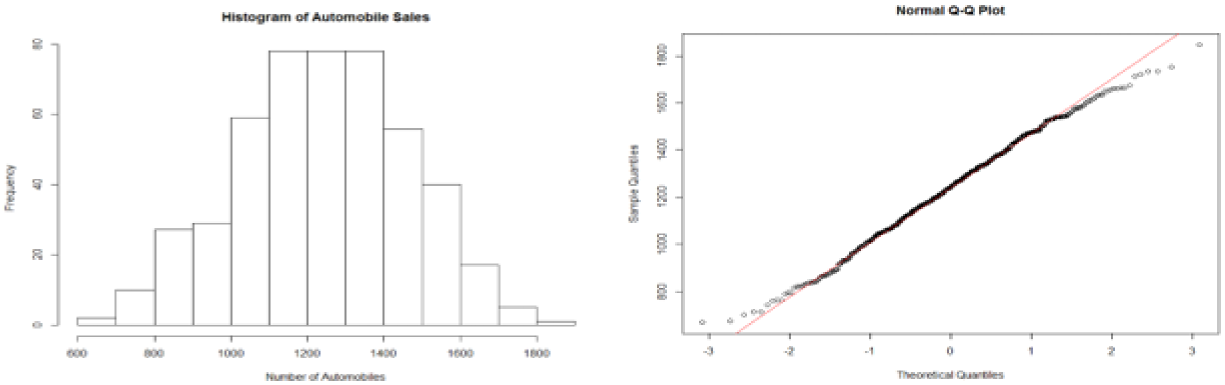


Figure 1: Histogram & Normal Quantile Plot - Original Time Series

Evaluating the time plot of units sold over time, the data shows evidence of upward trend, with few periods of downward trend during the economic slowdown. Looking at the time plot, we can see that there was a spike in the sales of automobile around 1985.

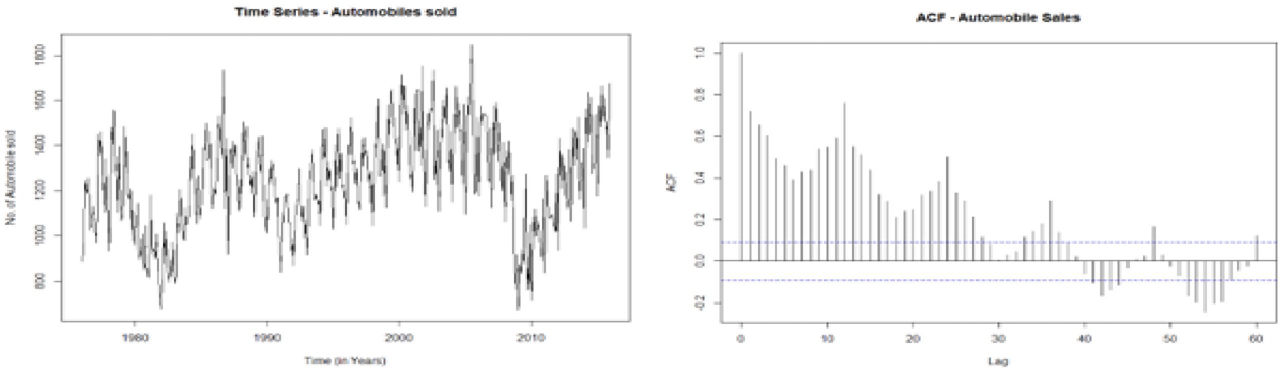


Figure 2: Time Series Plot (Left) & ACF (Right) - Original Data

The time series plot shows that the highest number of automobile sold was somewhere in 2006. We can also see that the economic recessions also affected the sales for instance, in 2009, there is a sudden decrease in the sales of automobile because of the recession.

Based on the time plot, we can see that the time series is non stationary. In addition to this, the Auto Correlation Function shows that the series has a very high serial autocorrelation and there is a very slow decay indicating that the previous sales data does effect the next sales data.

In order to remove the non-stationarity from the data, the first difference is taken. The time series plot suggests that the time series is now stationary Analyzing the first difference of a unit root time series shows that the first difference is stationary, which is further supported by the Dickey Fuller test.

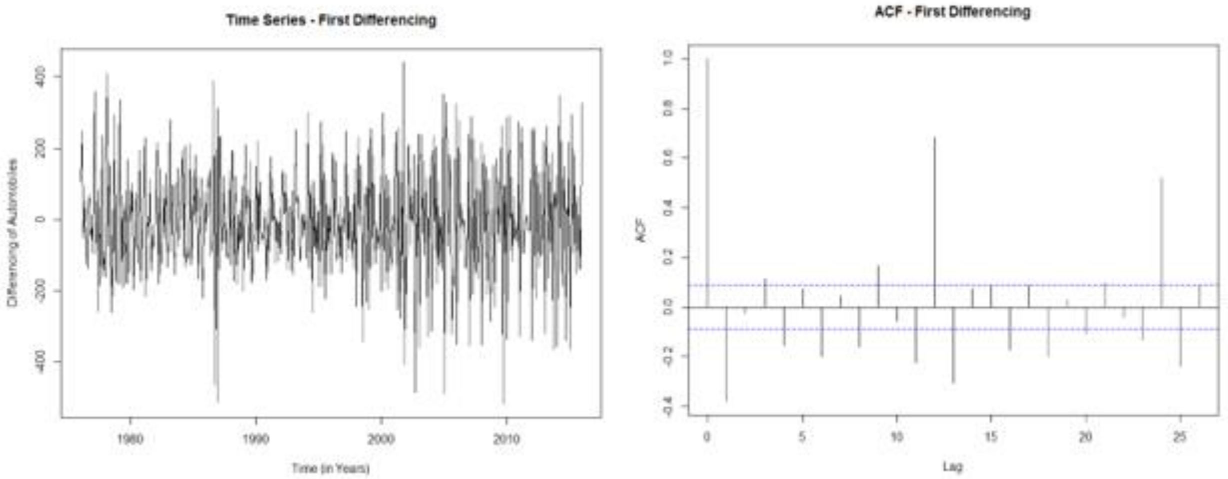
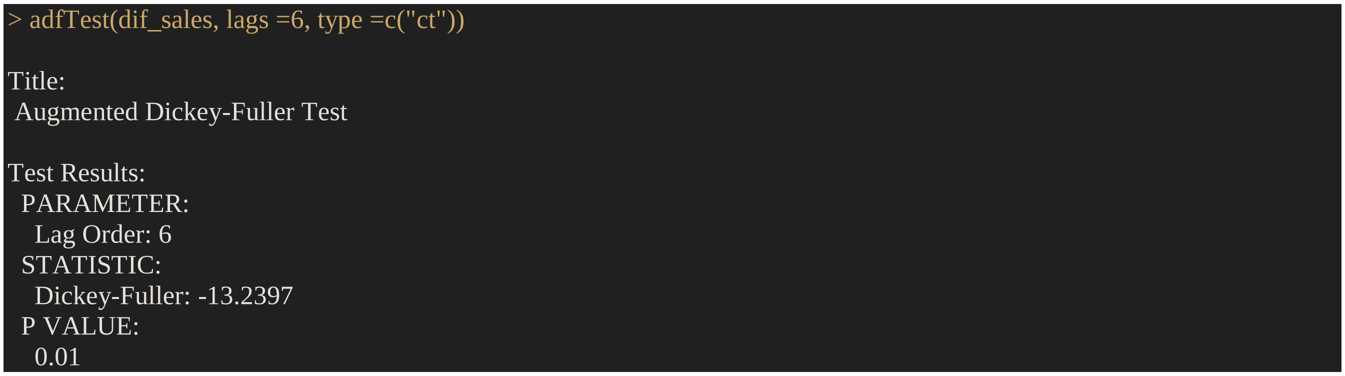


Figure 3: Time Series Plot (Left) & ACF (Right) - First Difference



```
> adfTest(dif_sales, lags =6, type =c("ct"))

Title:
 Augmented Dickey-Fuller Test

Test Results:
  PARAMETER:
    Lag Order: 6
  STATISTIC:
    Dickey-Fuller: -13.2397
  P VALUE:
    0.01
```

The next step of our exploratory analysis was to identify whether the time series displays seasonal behavior or not. In order to do, the Autocorrelation function of the first difference was analyzed and it showed large non zero values at recurrent lags. This might indicate that there is some seasonal behavior in the time series as it can be seen from the time series as well.

Model Building:

The exploratory analysis provided the information that the time series is non stationary along with this, it also has some seasonality. Keeping these points into consideration, the next stage of our analysis was to build a model.

In order to initiate the model building, the auto.arima function in R was used. This function identifies different models and based on different criteria such as AIC or BIC, it selects the model. For this analysis, Bayesian Information Criterion was used.

The model that was originally suggested by this function was ARIMA (0, 1, 1) (2, 0, 0) [12].

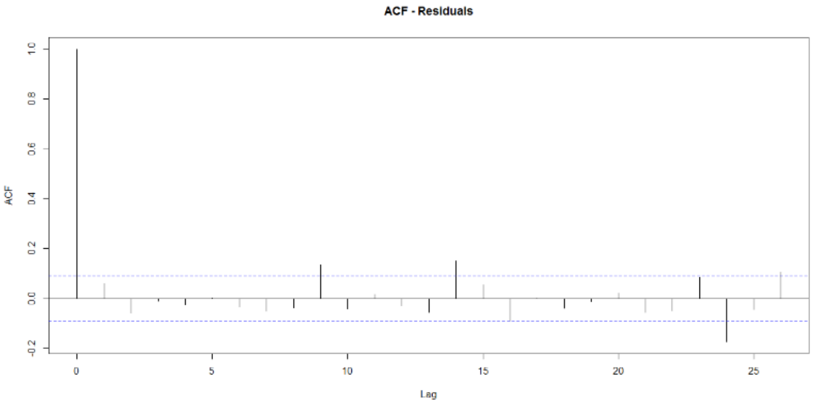


Figure 4: ACF of Residuals

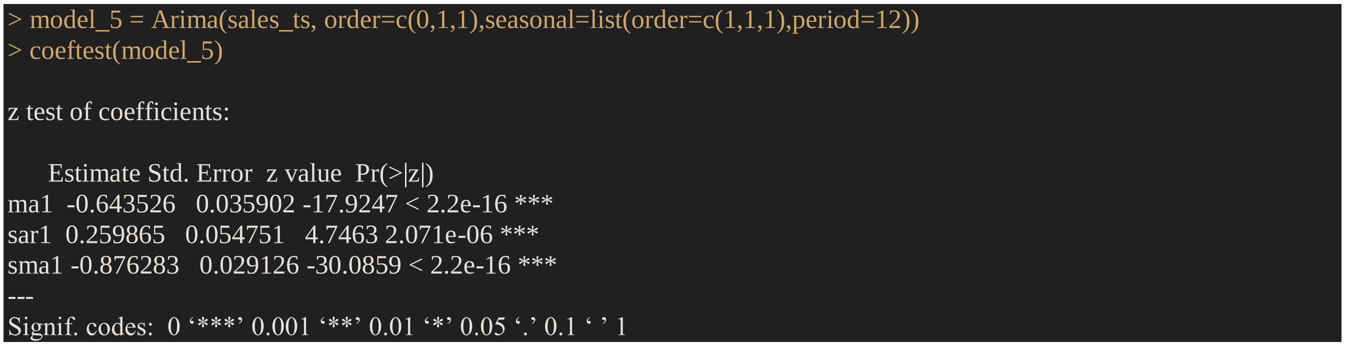
The model selected by auto.arima function was used, which showed that all the parameters were significant. However, during the residual analysis, the Auto Correlation Function displayed certain lags were non-zero at lag 9 and 14. For the purpose of further improvement in the residual analysis, we tried multiple models. These models were built by trial and error.

Based on the table displayed below, the model with the lowest Bayesian Information Criteria (BIC) was selected, which was ARIMA (0, 1, 1) (1, 1, 1) [12].

S. NO	MODELS	BIC Criteria	MAPE (95% training set)
1	ARIMA (0, 1, 1) (2, 0, 0) [12]	5811.3	3.80%
2	ARIMA (1, 1, 1) (2, 0, 0) [12]	5813.8	3.76%
3	ARIMA (0, 1, 1) (2, 0, 1) [12]	5746.7	3.14%
4	ARIMA (0, 1, 1) (0, 1, 1) [12]	5606.7	3.06%
5	ARIMA (0, 1, 1) (1, 1, 1) [12]	5592.3	3.02%

Model Diagnostics

For the final model, all the parameters in the model were significant as shown above. For the residual analysis, the normal quantile plot shows that the distribution of the residuals is close to normal distribution. In addition to this, the Ljung-Box test also indicates that there is no serial correlation among the residuals as the p-value is greater than 0.05, which means the null hypothesis that no serial correlation exists cannot be rejected. Thus, the residuals are white noise.



```
> model_5 = Arima(sales_ts, order=c(0,1,1),seasonal=list(order=c(1,1,1),period=12))
> coeftest(model_5)

z test of coefficients:

     Estimate Std. Error  z value  Pr(>|z|)
ma1  -0.643526   0.035902 -17.9247 < 2.2e-16 ***
sar1  0.259865   0.054751   4.7463 2.071e-06 ***
sma1 -0.876283   0.029126 -30.0859 < 2.2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Although the Autocorrelation function of the residual still display non-zero values at lag 9 and 14, but they are not significant. These analysis show that the model ARIMA (0, 1, 1) (1, 1, 1) [12] is acceptable and can be used to explain the process and compute the forecasts.

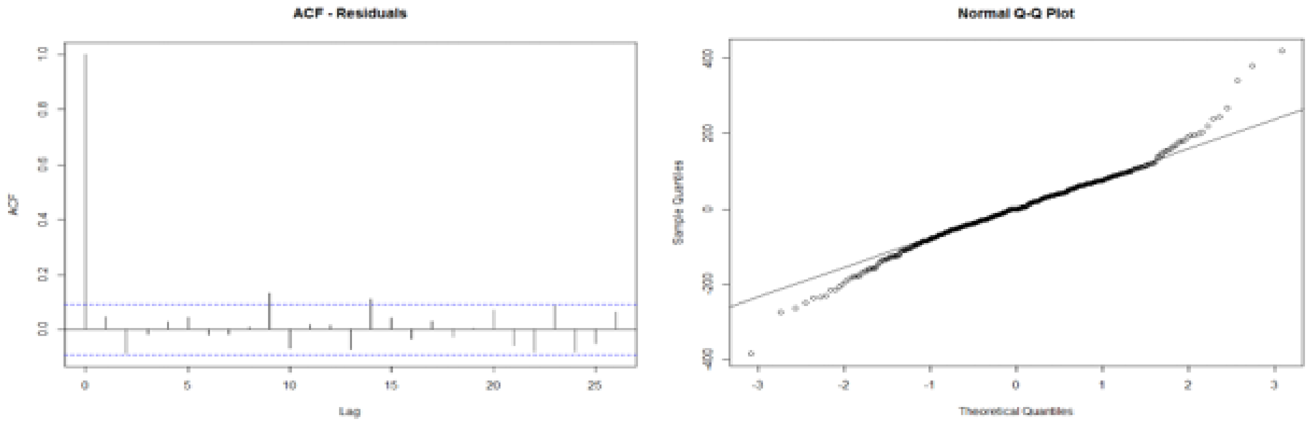


Figure 5: ACF (Left) & Normal QQ Plot (Right) - Residuals of Final Model

```
> Box.test(model_5$residuals, type = "Ljung-Box", fitdf = 3, lag = 8)

            Box-Ljung test

data:  model_5$residuals
X-squared = 6.5563, df = 5, p-value = 0.2558

> Box.test(model_5$residuals, type = "Ljung-Box", fitdf = 3, lag = 6)

            Box-Ljung test

data:  model_5$residuals
X-squared = 6.3601, df = 3, p-value = 0.09535
```

Forecasting

For the purpose of forecasting, we will use the final selected model, and compute the monthly forecast for the next year which are displayed below. The forecast table also includes the 80% Confidence interval as well as the 95% Confidence Interval. For instance, for the month of January in 2016, the model predicted the sales of 12.95 million units and we are 95% confident that the number of units will be in between 11.13 million units and 14.77 million units.

```
> forecast.model5
         Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
Jan 2016       1295.528 1176.817 1414.239 1113.975 1477.081
Feb 2016       1428.688 1302.660 1554.716 1235.945 1621.431
Mar 2016       1701.537 1568.594 1834.480 1498.219 1904.856
Apr 2016       1591.102 1451.587 1730.618 1377.732 1804.473
May 2016       1737.057 1591.264 1882.849 1514.087 1960.026
Jun 2016       1640.334 1488.524 1792.143 1408.161 1872.506
Jul 2016       1630.111 1472.514 1787.708 1389.087 1871.135
Aug 2016       1701.599 1538.420 1864.779 1452.038 1951.161
Sep 2016       1527.926 1359.348 1696.503 1270.108 1785.743
Oct 2016       1526.697 1352.889 1700.505 1260.881 1792.513
Nov 2016       1443.918 1265.033 1622.803 1170.337 1717.499
Dec 2016       1688.706 1504.883 1872.528 1407.574 1969.838
```

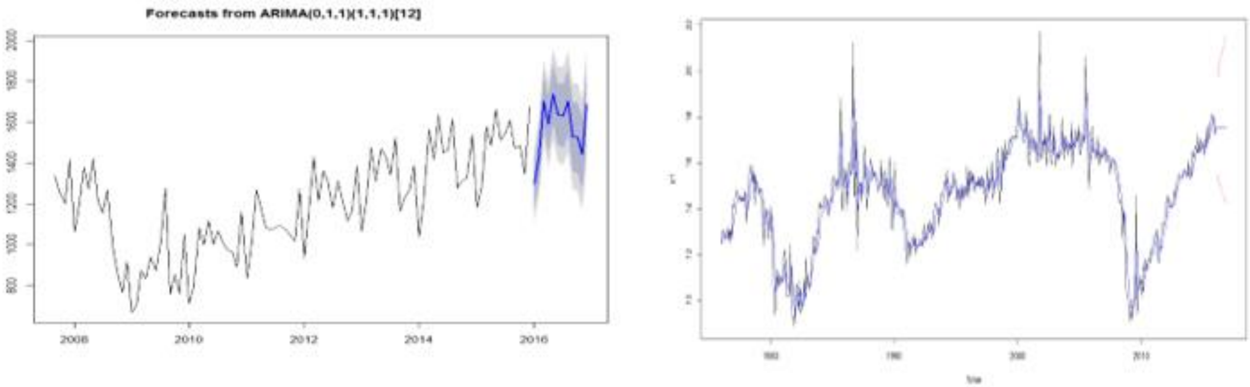
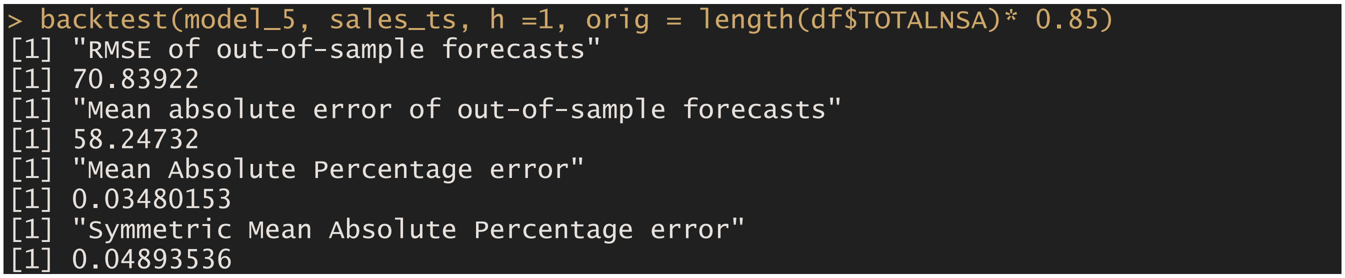


Figure 6: Forecast Plot (Left) & Forecasted and Observed Time Series (Right)

Furthermore, the forecast plot indicates that the forecasts follow similar trend as in the previous years, this means that it takes into account the seasonal trends of the time series. One thing to take into consideration is that there was a lot of variation in 2009 economic crisis and after that the economy is trying to recover from it.

MODEL VALIDATION:

The final part of our analysis includes the process of model validation. The technique of back testing is used to validate the model and analyze the prediction accuracy of the final model. In order to do this, the training set comprises of 85% of the data whereas the testing set consists of 15%. The procedure shows that the MAPE for one-step ahead forecast is 0.0348. This means that the forecasts for the number of automobile sold using the current model are on an average 3.5% off the observed values.



```
> backtest(model_5, sales_ts, h =1, orig = length(df$TOTALNSA)* 0.85)
[1] "RMSE of out-of-sample forecasts"
[1] 70.83922
[1] "Mean absolute error of out-of-sample forecasts"
[1] 58.24732
[1] "Mean Absolute Percentage error"
[1] 0.03480153
[1] "Symmetric Mean Absolute Percentage error"
[1] 0.04893536
```

CONCLUSION:

The final model ARIMA (0, 1, 1) (1, 1, 1) [12] takes into account the fact that the original time series was non stationary along with the seasonal behavior. Different tests for the residual analysis supported our claim that the residuals are white noise and there is no particular pattern that is followed by them. Hence, we can say that the selected model is acceptable.

Finally, the forecasts also indicate that the seasonal factor is captured by the model and the model validation indicates that the MAPE is around 3.5% which is considered to be reasonable. Thus, we can say that our model predicts the number of automobile units sold with an accuracy. However, one thing to keep in mind is that the model is not so accurate in case of a sudden shock in the economy, but overall the model is a good predictive model and can be used for forecasting.

APPENDIX

```
library (forecast)
library (zoo)
library (fBasics)
library (tseries)
library (lmtest)
library (stats)
library (dplyr)
library (ggplot2)
library (openxlsx)
library (fUnitRoots)

#reading the datafile
df = read.csv("rawcar.csv")
#Creating a time series for automobile sales
sales_ts = ts(df$TOTALNSA, start=c(1976, 1), freq = 12)

################## Exploratory Analysis ##########################
basicStats(sales_ts)
#Plot the time series
plot(sales_ts, ylab = 'No. of Automobile sold', xlab = 'Time (in Years)', main = 'Time Series - Automobiles sold')
# Perform Jarque-Bera normality test.
normalTest(sales_ts,method=c("jb"))
#Histogram
hist(sales_ts, main = "Histogram of Automobile Sales", xlab = 'Number of Automobiles')
#QQ Plot
qqnorm(sales_ts)
qqline(sales_ts, col = 'red')
# Dickey Fuller
adfTest(sales_ts, lags =8, type =c("c"))
adfTest(sales_ts, lags =10, type =c("c"))

'''
library(dygraphs)
dygraph(sales_ts,ylab="Number of Automobile Sold (In Thousands)",        ) %>%
  dyOptions(colors = "red") %>%
  dyRangeSelector()
'''

#Acf plot
#Acf indicates highly correlated data
acf(coredata(sales_ts), main = 'ACF - Automobile Sales', lag = 60)
#to look for seasonality
sales.stl = stl(sales_ts, s.window = 'per')
plot(sales.stl)

#Differencing of the ts
dif_sales = diff(sales_ts)
#plotting the time series
#Plot shows that the series is not stationary, no seasonality
plot(dif_sales, main = 'Time Series - First Differencing', xlab = 'Time (in Years)', ylab = 'Differencing of Automobiles')
#Acf & PACF with first differencing to analyze seasonality, no seasonality indicated
acf(coredata(dif_sales), main = 'ACF - First Differencing')
pacf(coredata(dif_sales), main = 'PACF - First Differencing')

# Dickey Fuller
adfTest(dif_sales, lags =6, type =c("ct"))
adfTest(sales_ts, lags =3, type =c("ct"))


########################### Dicky Fuller ###########################


adfTest(sales_ts,lag=3,type='ct')
adfTest(sales_ts,lag=5,type='ct')
#We fail to reject the null hypothesis of unit root non stationary

########################## Model Creation #######################
sample_fit = auto.arima(sales_ts, trace = T, max.p = 8, max.q = 8, stationary = FALSE, seasonal = T,
               ic = c('bic'))

 #First model based on Auto Arima function (0,1,1)(2,0,0)
model_1 = Arima(sales_ts, order=c(0,1,1),seasonal=list(order=c(2,0,0),period=12))
coeftest(model_1)

#ACF for Residuals
acf(coredata(model_1$residuals), main = 'ACF - Residuals')
pacf(coredata(model_1$residuals), main = 'ACF - Residuals')
#Ljung Box Test
Box.test(model_1$residuals, type = "Ljung-Box", fitdf = 3, lag = 9)
Box.test(model_1$residuals, type = "Ljung-Box", fitdf = 3, lag = 12)
```

```
##################### Second Attempt
model_2 = Arima(sales_ts, order=c(0,1,2),seasonal=list(order=c(2,0,0),period=12))
coeftest(model_2)

#ACF for Residuals
acf(coredata(model_2$residuals), main = 'ACF - Residuals')
pacf(coredata(model_2$residuals), main = 'ACF - Residuals')
#Ljung Box Test
Box.test(model_2$residuals, type = "Ljung-Box", fitdf = 4, lag = 12)
Box.test(model_2$residuals, type = "Ljung-Box", fitdf = 4, lag = 13)


##################### Third Attempt ############################
model_3 = Arima(sales_ts, order=c(0,1,1),seasonal=list(order=c(2,0,1),period=12))
coeftest(model_3)

#ACF for Residuals
acf(coredata(model_3$residuals), main = 'ACF - Residuals')
pacf(coredata(model_3$residuals), main = 'ACF - Residuals')
#Ljung Box Test
Box.test(model_3$residuals, type = "Ljung-Box", fitdf = 4, lag = 7)
Box.test(model_3$residuals, type = "Ljung-Box", fitdf = 4, lag = 8)

##################### Fourth Attempt ##############################

model_4 = Arima(sales_ts, order=c(0,1,1),seasonal=list(order=c(0,1,1),period=12))
coeftest(model_4)

#ACF for Residuals
acf(coredata(model_4$residuals), main = 'ACF - Residuals')
pacf(coredata(model_4$residuals), main = 'ACF - Residuals')
#Ljung Box Test
Box.test(model_4$residuals, type = "Ljung-Box", fitdf = 3, lag = 9)
Box.test(model_4$residuals, type = "Ljung-Box", fitdf = 3, lag = 12)


####################### Fifth Attempt ##########################

model_5 = Arima(sales_ts, order=c(0,1,1),seasonal=list(order=c(1,1,1),period=12))
coeftest(model_5)

#ACF for Residuals
acf(coredata(model_5$residuals), main = 'ACF - Residuals')
pacf(coredata(model_5$residuals), main = 'ACF - Residuals')
#Ljung Box Test
Box.test(model_5$residuals, type = "Ljung-Box", fitdf = 3, lag = 8)
Box.test(model_5$residuals, type = "Ljung-Box", fitdf = 3, lag = 6)
#QQplot
qqnorm(model_5$residuals)
qqline(model_5$residuals)

########################### Prediction ######################

#Compute Prediction
forecast.model5 = forecast.Arima(model_5, h=12)
#display the forecast
forecast.model5
#plot the forecast
plot(forecast.model5, include  = 100)
lines(ts(c(forecast.model5$fitted, forecast.model5$mean), frequency=12,start=c(1976,1)), col="blue")


source("F:/MSPA/CSC 425/Homework 4/backtest.R")
backtest(model_1, sales_ts, h =1, orig = length(df$TOTALNSA)* 0.95)
backtest(model_2, sales_ts, h =1, orig = length(df$TOTALNSA)* 0.95)
backtest(model_3, sales_ts, h =1, orig = length(df$TOTALNSA)* 0.95)
backtest(model_4, sales_ts, h =1, orig = length(df$TOTALNSA)* 0.95)
backtest(model_5, sales_ts, h =1, orig = length(df$TOTALNSA)* 0.85)
```