**FORECASTING OF U.S. AIRLINES: MONTHLY AIRCRAFT MILES FLOWN (Millions) DATA [1963 -1970]**

**INTRODUCTION**

***Problem Description:*** Here in this problem we are interested in

1. Performing various steps to identify and extract the stationary version of the data series.
2. And futher in demonstrating the simple exponential smoothing technique to find the insample forecast as well as one step ahead forecast, for the extracted version of stationary time series.
3. We are also interested in Obtaining the accuracy measures and comment about the findings.

***Objective:*** The main objective of this problem is to extract the stationary version of the given U.S. airlines monthly time series data and then for the stationary time series we want to find the insample forecast as well as one step ahead forecast using simple exponential smoothing technique. Further we are also interseted in obtaining the accuracy measure and draw a conclusion out of it.

*#Setting and getting the current working directory.*  
**setwd**("E:/M.Sc/SEM III/TIME\_SERIES\_ANALYSIS(MST371)/Practical Labs")  
**getwd**()

## [1] "E:/M.Sc/SEM III/TIME\_SERIES\_ANALYSIS(MST371)/Practical Labs"

***Data Description:***

The data set consists of U.S. airlines: monthly aircraft miles flown (Millions) 1963 -1970. The data set has total 96 records of miles flown by aircraft and and the date when the data is recorded. Also it is observed that in each year the data has been recorded in each months. There two variables in the the dataset i.e. Date and monthly aircraft miles flown (Millions).

*#Loading the package required to load the dataset.*  
**library**(readxl)  
  
*#Loading the 'U.S. airlines: monthly aircraft miles flown (Millions) 1963 -1970' dataset.*  
data<- **read\_excel**("Airlines.xlsx")  
  
*#Obtaining the first few records of the dataset.*  
**head**(data)

## # A tibble: 6 x 2  
## Month `Miles\_flown(Millions)`  
## <dttm> <dbl>  
## 1 1963-01-01 00:00:00 6827  
## 2 1963-02-01 00:00:00 6178  
## 3 1963-03-01 00:00:00 7084  
## 4 1963-04-01 00:00:00 8162  
## 5 1963-05-01 00:00:00 8462  
## 6 1963-06-01 00:00:00 9644

**ANALYSIS**

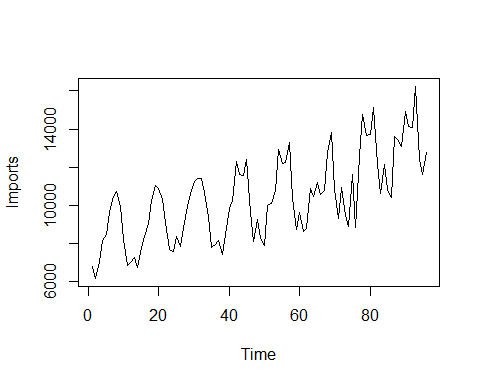
*#Extracing the data for the imports variable which we are interested in.*  
Airlines=data**$**`Miles\_flown(Millions)`  
  
*#Now converting the it into a time series data.*  
Airlines1=**ts**(Airlines)  
  
*#Here. we are checking if the dataset has been converted into a timeseries plot.*  
**class**(Airlines1)

## [1] "ts"

Hence, now the dataset we are interested in is a timeseries data.

1. Performing various steps to identify and extract the stationary version of the data series.

*#Now we will first obtain the time series plot of the data to understand the nature of the time series data.*  
**ts.plot**(Airlines1, gpars = **list**(xlab="Time",ylab="Imports",lty=**c**(1**:**20)))

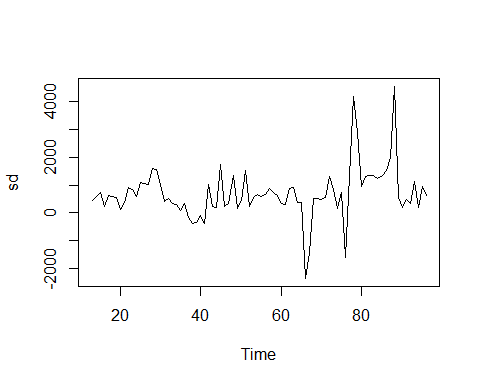




***Interpretation:*** From the above time series plot (Figure1) we observe that there exists a trend as well as a seasonal component in the dataset since there is observed a increase pattern for a longer period of time and also there’s some kind of repeatative movement in every 12 months is observed. Also we observe that there is some kind of irregularity in the dataset hence we can say that there also exists a error component in the dataset.

Since, from the above time series plot we observed that our time series data is non stationary and there exist a seasonal and trend component in the time series data, we first try to eliminate the seasonal component using seasonal differencing and then we can try to remove trend component using method of moving average a to make it stationary.

*#Now, we proceed to perform the deseasonalising of data.*  
sd=**diff**(Airlines1,lag = 12)  
  
*#Obtaining the timeseries plot.*  
**ts.plot**(sd)

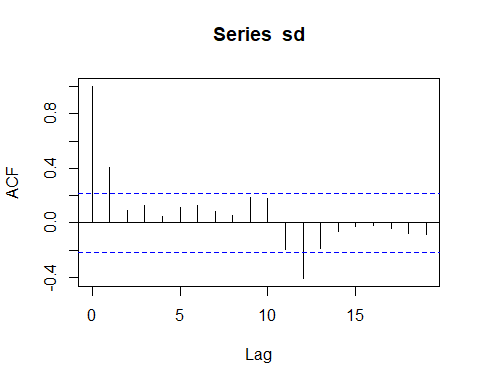




*#Obtaining the acf plot for the above time series after first differencing to check for stationarity.*  
**acf**(sd)  
  
*#loading the package 'tseries'*  
**library**(tseries)

## Warning: package 'tseries' was built under R version 4.0.5

## Registered S3 method overwritten by 'quantmod':  
## method from  
## as.zoo.data.frame zoo



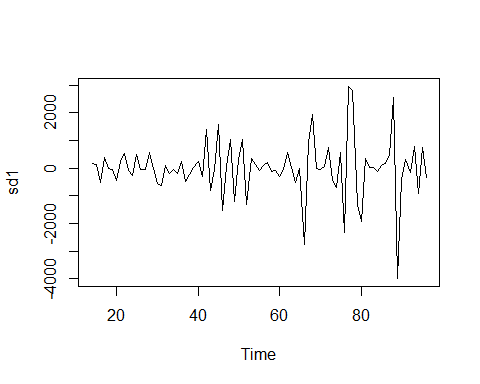


*#Now we want to validate for the stationarity of the new time series dataset using Augmented Fuller(ADF test) test. Dickey*  
**adf.test**(sd)

##   
## Augmented Dickey-Fuller Test  
##   
## data: sd  
## Dickey-Fuller = -3.0316, Lag order = 4, p-value = 0.1527  
## alternative hypothesis: stationary

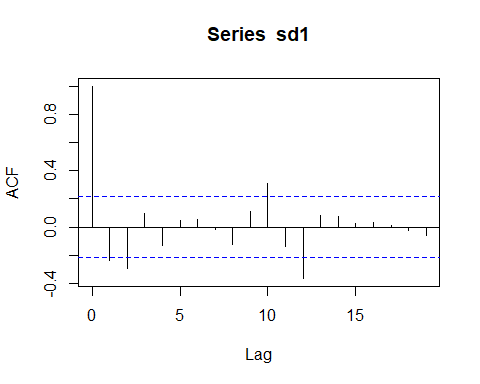
***Interpretation:*** From the acf plot (Figure3) the series seems to be non stationary because lag 1, lag 12 are beyond the threshold line and also from the above statistical test Augmented Dickey-Fuller Test it is observed tha pvalue = 0.1527 > 0.05, thus we fail to reject the null hypothesis and conclude that the deseasonalized timeseries is still non stationary series, which means there still exist a trend component in the series and we need to detrend the series and we do it by the method of differencing.

*#Now, we proceed to perform the method of differencing to remove the trend component fro the data and make it stationary..*  
sd1=**diff**(sd)  
  
*#Obtaining the timeseries plot.*  
**ts.plot**(sd1)





*#Obtaining the acf plot for the above time series after differencing to check for stationarity.*  
**acf**(sd1)



*#loading the package 'tseries'*  
**library**(tseries)  
  
*#Now we want to validate for the stationarity of the new time series dataset using Augmented Fuller(ADF test) test. Dickey*  
**adf.test**(sd1)

## Warning in adf.test(sd1): p-value smaller than printed p-value

##   
## Augmented Dickey-Fuller Test  
##   
## data: sd1  
## Dickey-Fuller = -6.1312, Lag order = 4, p-value = 0.01  
## alternative hypothesis: stationary

***Interpretation:*** From the acf plot (Figure5) the series seems to be non stationary because lag 1, lag 2,lag 10, lag 12 are beyond the threshold line but also from the above statistical test Augmented Dickey-Fuller Test it is observed tha pvalue is less than 0.01 > 0.05, thus we reject the null hypothesis and conclude that the detrended timeseries is a stationary series.

1. Demonstrating the simple exponential smoothing technique to find the in-sample forecast as well as one step ahead forecast, for the extracted version of stationary time series.

Now, since we have extracted the stationary series we proceed to buld a forecasting model using simple exponential smoothing.

*#Forecasting using simple exponential smoothing.*  
Airlines\_forecast<-**HoltWinters**(sd1, beta = FALSE, gamma = FALSE)  
Airlines\_forecast

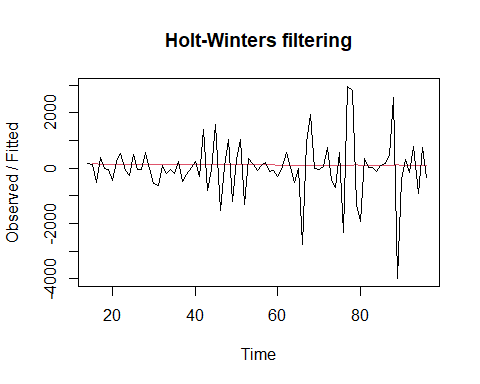
## Holt-Winters exponential smoothing without trend and without seasonal component.  
##   
## Call:  
## HoltWinters(x = sd1, beta = FALSE, gamma = FALSE)  
##   
## Smoothing parameters:  
## alpha: 0.006258201  
## beta : FALSE  
## gamma: FALSE  
##   
## Coefficients:  
## [,1]  
## a 92.40181

***Interpretation:*** Thus, we observe from the above model that the alpha value = 0.006258201 which is farther from 1 which indicates that the weightage is given to all the observations that means forecast is not just on the basis of recent past observation it is based on all the observations.

*#Performing in-sample forecast.*  
Airlines\_forecast**$**fitted

## Time Series:  
## Start = 15   
## End = 96   
## Frequency = 1   
## xhat level  
## 15 155.00000 155.00000  
## 16 154.89361 154.89361  
## 17 150.63244 150.63244  
## 18 152.18052 152.18052  
## 19 151.20937 151.20937  
## 20 150.01274 150.01274  
## 21 146.38290 146.38290  
## 22 146.94375 146.94375  
## 23 149.43486 149.43486  
## 24 148.01153 148.01153  
## 25 145.44559 145.44559  
## 26 147.70202 147.70202  
## 27 146.60870 146.60870  
## 28 145.41583 145.41583  
## 29 148.05419 148.05419  
## 30 147.07757 147.07757  
## 31 142.62750 142.62750  
## 32 137.91115 137.91115  
## 33 137.76777 137.76777  
## 34 135.79789 135.79789  
## 35 134.66642 134.66642  
## 36 132.59078 132.59078  
## 37 133.26923 133.26923  
## 38 129.33739 129.33739  
## 39 127.14491 127.14491  
## 40 126.59954 126.59954  
## 41 127.44065 127.44065  
## 42 124.75312 124.75312  
## 43 132.82775 132.82775  
## 44 126.88979 126.88979  
## 45 125.92672 125.92672  
## 46 134.93899 134.93899  
## 47 124.63837 124.63837  
## 48 124.34650 124.34650  
## 49 130.00174 130.00174  
## 50 121.68458 121.68458  
## 51 122.84432 122.84432  
## 52 128.67794 128.67794  
## 53 119.76202 119.76202  
## 54 121.10902 121.10902  
## 55 121.09582 121.09582  
## 56 119.81855 119.81855  
## 57 119.59439 119.59439  
## 58 120.06630 120.06630  
## 59 118.47004 118.47004  
## 60 117.14662 117.14662  
## 61 114.52351 114.52351  
## 62 113.62531 113.62531  
## 63 116.46888 116.46888  
## 64 115.90897 115.90897  
## 65 111.91055 111.91055  
## 66 111.21645 111.21645  
## 67 93.37296 93.37296  
## 68 98.63378 98.63378  
## 69 110.23251 110.23251  
## 70 109.58646 109.58646  
## 71 108.65032 108.65032  
## 72 108.40218 108.40218  
## 73 112.42369 112.42369  
## 74 108.84760 108.84760  
## 75 103.75438 103.75438  
## 76 106.64721 106.64721  
## 77 91.48580 91.48580  
## 78 109.44379 109.44379  
## 79 126.36319 126.36319  
## 80 117.44924 117.44924  
## 81 104.74854 104.74854  
## 82 106.20201 106.20201  
## 83 105.72513 105.72513  
## 84 105.23245 105.23245  
## 85 103.82916 103.82916  
## 86 103.88655 103.88655  
## 87 104.28778 104.28778  
## 88 106.54519 106.54519  
## 89 121.94947 121.94947  
## 90 96.25362 96.25362  
## 91 93.19803 93.19803  
## 92 94.52978 94.52978  
## 93 92.97444 92.97444  
## 94 97.38663 97.38663  
## 95 91.01962 91.01962  
## 96 95.11862 95.11862

*#Visualizing the in-sample forecast.*  
**plot**(Airlines\_forecast)





*#Loading the library 'forecast'.*  
**library**(forecast)

## Warning: package 'forecast' was built under R version 4.0.5

*#Forecasting the last observation to compare with actual out-sample forecast result if the model obtained is good or not.*  
Airlines\_forecast1<-**HoltWinters**(sd1[1**:**82], beta=FALSE, gamma = FALSE)  
Airlines\_forecast1

## Holt-Winters exponential smoothing without trend and without seasonal component.  
##   
## Call:  
## HoltWinters(x = sd1[1:82], beta = FALSE, gamma = FALSE)  
##   
## Smoothing parameters:  
## alpha: 0.005473258  
## beta : FALSE  
## gamma: FALSE  
##   
## Coefficients:  
## [,1]  
## a 101.0708

forecast1<-**forecast**(Airlines\_forecast1,h=1)  
forecast1

## Point Forecast Lo 80 Hi 80 Lo 95 Hi 95  
## 83 101.0708 -1207.767 1409.909 -1900.624 2102.766

*#Loading the package 'Metrics'.*  
**library**(Metrics)

## Warning: package 'Metrics' was built under R version 4.0.5

##   
## Attaching package: 'Metrics'

## The following object is masked from 'package:forecast':  
##   
## accuracy

*#Obtaining the accuracy measure for the forecasting model obtained by removing the last value.*  
ma<-**rmse**(sd1,Airlines\_forecast1**$**fitted)  
ma

## [1] 924.3185

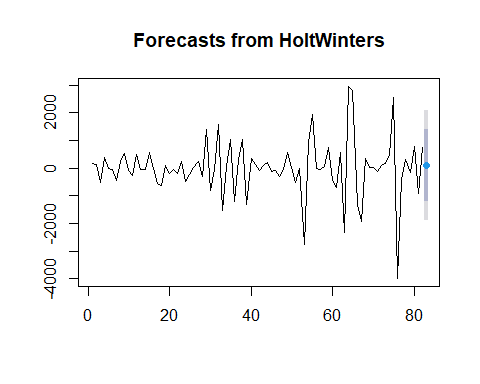
***Interpretation:*** It is observed from the above calculations that the the 83rd value obtained using forecasting model is 101.07 which is way too far from the actual value also the root mean square for the above model is evident to the fact that the above model is not that good.

*#Performing out-sample one step ahead forecast for Jan 1971.*  
forecast2<-**forecast**(Airlines\_forecast,h=1)  
forecast2

## Point Forecast Lo 80 Hi 80 Lo 95 Hi 95  
## 97 92.40181 -1209.651 1394.455 -1898.917 2083.72

***Interpretation:*** The forcated value for the airlines dataset on jan, 1971 is obtained as 92.40181 from the model obtained. Thus we will get an idea about the accuracy of the model from the following accuracy measure.

*#Plotting the forecasted value.*  
**plot**(forecast1)





Thus, the forecasted value is plotted in the graph.

1. Obtaining the accuracy measures and comment about the findings.

***3.1 Root mean square value***

*#Obtaining the root mean square error value to check the accuracy of the model.*  
ma<-**rmse**(sd1,Airlines\_forecast**$**fitted)  
ma

## [1] 1017.124

***3.2 Mean Absolute Error***

*#Obtaining the mean absolute error value to check the accuracy of the model.*  
**mae**(sd1,Airlines\_forecast**$**fitted)

## [1] 642.1991

***3.3 Mean Square Error***

*#Obtaining the mean square error value to check the accuracy of the model.*  
**mse**(sd1,Airlines\_forecast**$**fitted)

## [1] 1034542

***Interpretation:*** As metioned above it is also evident from the above root mean square error, mean absolute error and mean square error that the model obtained is not accurate since the root mean square error, mean absolute error and mean square error value for the model are too large. Thus, it is observed that there is 0% accuracy in the model

**CONCLUSION**

Thus we conclude from the above analysis we interpret that the forecasting model obtained for the airline data is not good that is the model is less accurate. Thus, the forecasted value for airline flown in Jan, 1971 is also not correct.