FORECASTING

Assignment 1

BP: Forecast the Airlines Passengers data set. Prepare a document for each model explaining how many dummy variables you have created and RMSE value for each model. Finally which model you will use for Forecasting.

**PROCEDURE:**

STEP 1: First we have to Exploratory Data Analysis which can be done by plotting scattered plot, box plots and summary.

summary(Airlines\_Data\_1\_)

Month Passengers

Min. :1995-01-01 00:00:00 Min. :104.0

1st Qu.:1996-12-24 06:00:00 1st Qu.:156.0

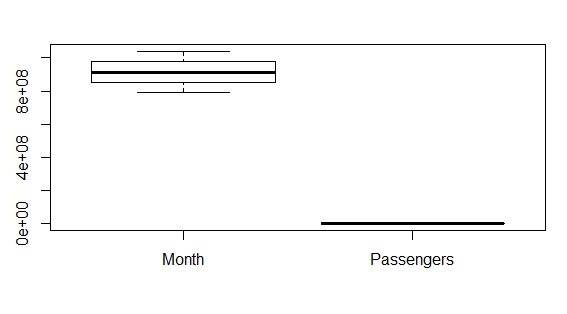
Median :1998-12-16 12:00:00 Median :200.0

Mean :1998-12-16 05:00:00 Mean :213.7

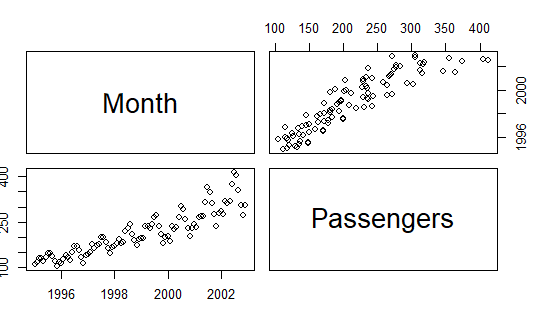
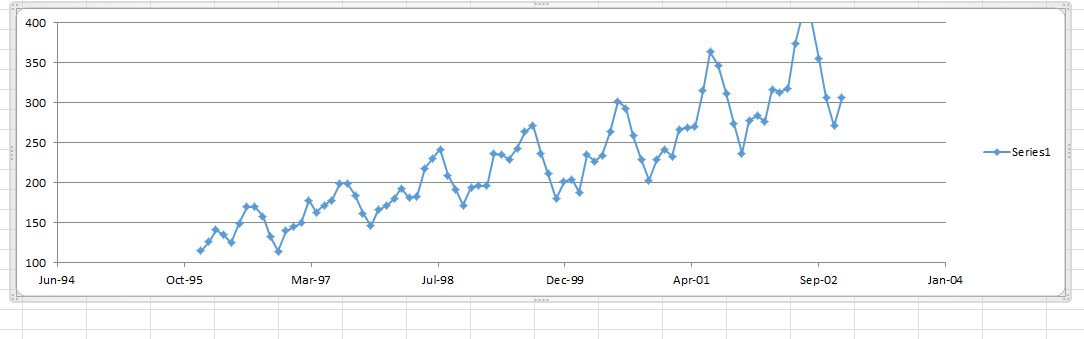
3rd Qu.:2000-12-08 18:00:00 3rd Qu.:264.8

Max. :2002-12-01 00:00:00 Max. :413.0

From the summary we can see that in Month & Passengers the difference between mean and max is not so large so it is centre skewed can my confirmed by using Box Plot .



Scatter Plot for the Following data frame is:

  
  
STEP 2: Forecasting the data   
  
  
first from the data we have to analyse the seasonality, trend.  
  
Now on drawing the plots we can say that the data is seasonal, has some trend.  
  


Now, pre-processing the data. After pre-processing uploading the data in excel miner then we can partition the data and then apply forecasting methods.

Now using r Studio we can pre-process data by following code:

Dividing the dataset into 12 columns and assigning names to columns  
x <- data.frame(outer(rep(month.abb,length=96),month.abb,"==") +0)

> colnames(x) <- month.abb

> View(x)  
  
Combing the columns of original data set to the newly formed dataset :

> airlinespassengers <- cbind(Airlines\_Data\_original,x)

> View(airlinespassengers)

Adding the columns as a part of preprocessing data :   
adding time index:

airlinespassengers["t"] <- c(1:96)

adding the column log :airlinespassengers["log\_Passengers"] <- log(airlinespassengers["Passengers"])

adding the column t Square : airlinespassengers["T\_Square"] <- airlinespassengers["t"] \* airlinespassengers["t"]

> View(airlinespassengers)

End of pre-processing the data. Now we have to partition data :

train <- airlinespassengers[1:84,]

test <- airlinespassengers[85:96,]

Now preparing the models to find out the value of RMSE :

Linear Model:  
linearmodel <- lm(Passengers~t , data = train)

> linear\_pred <- data.frame(predict(linearmodel, interval = 'predict', newdata=test))

> rmselinear <- sqrt(mean((test$Passengers-linear\_pred$fit)^2 , na.rm = T))

exponential model :

expomodel <- lm(log\_Passengers~t , data= train)

> expopred <- data.frame(predict(expomodel,interval = 'predict',newdata = test))

> rmseexpo <- sqrt(mean((test$Passengers- exp(expopred$fit))^2 , na.rm = T))

> rmseexpo

Quadratic:

quamodel <- lm(Passengers~t+ T\_Square , data= train)

> quapred <- data.frame(predict(quamodel, interval = 'predict' , newdata = test))

> rmsequa <- sqrt(mean((test$Passengers-quapred$fit)^2 , na.rm=T))

Additive seasonality :

addmodel <- lm(Passengers ~ Jan+Feb+Mar+Apr+May+Jun +Jul +Aug +Sep +Oct +Nov , data=train)

> addpred <- data.frame(predict(addmodel, interval = 'predict', newdata = test))

> rmsadd <- sqrt(mean((test$Passengers-addpred$fit)^2, na.ram=T))

Additive seasonality with quadratic trend :

addquadpred <- data.frame(predict(addquadmodel, interval='predict' , newdata=test))

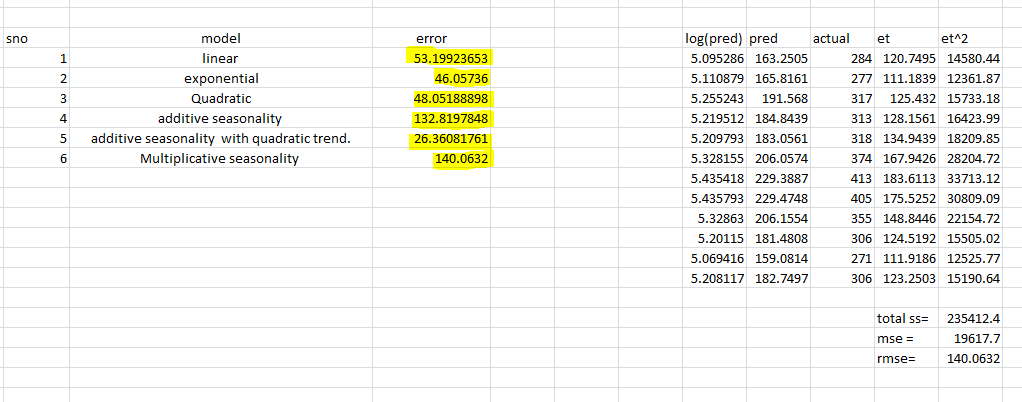
> addquadadd <- sqrt(mean((test$Passengers-addquadpred$fit)^2, na.rm = T))

Multiplication Seasonality :

mulmodel <- lm(log\_Passengers~Jan+Feb+Mar+Apr+May+Jun +Jul +Aug +Sep +Oct +Nov , data=train)

> mullpred <- data.frame(predict(mulmodel , interval = 'predict' , newdata = test))

> rmsemul <- sqrt(mean((test$Passengers-exp(mullpred$fit))^2, na.rm = T))

  
 From the above table we can that the Best model is Additive seasonality with quadratic trend.  
  
Now prepare a final model with complete data ( not with partitioning )

finalmodel <- lm(Passengers~t+T\_Square+Jan+Feb+Mar+Apr+May+Jun +Jul +Aug +Sep +Oct +Nov , data=airlinespassengers)

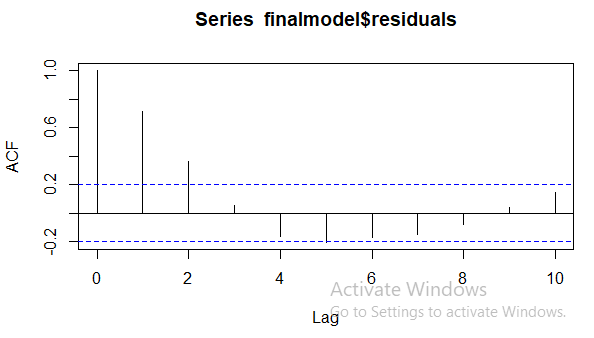
Now a dataset of pre-processed set is uploaded into r studio:

predictnew <- predict(finalmodel , newdata = test\_data , interval = 'predict')

> predictnew <- as.data.frame(predictnew)

Now using these we can predict the values and plotting the acf we can find any significance is existing :

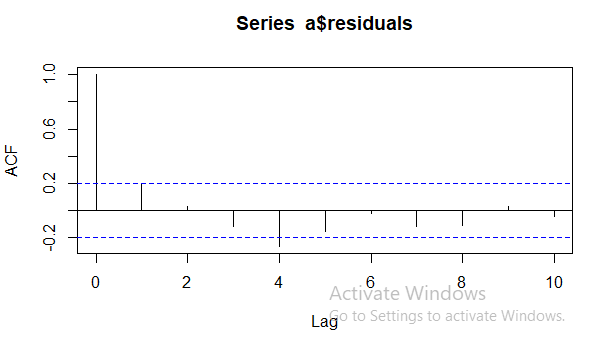
acf(finalmodel$residuals , lag.max = 10)



From the plot we can see that there are lot of significant values , acc to law of parsimony we use arima model i.e., autoregressive model with 1 (as 1st significant error) :

a <- arima(finalmodel$residuals , order = c(1,0,0))

Now we do error or errors to reduce the significant values:



Now we can observe that there are no significant values ( as lag 0 we dint consider it ).

Forecasting the errors :  
errors12 <- forecast(a , h=12)

futureerrors <- data.frame(errors12)

> class(futureerrors)

[1] "data.frame"

> futureerrors <- futureerrors$Point.Forecast  
  
  
Now the forecasted data for the next 12 months along with errors I given by :   
  
prednewvalues <- predictnew + futureerrors

We can save the file by using the following code:

write.csv(prednewvalues , file="prednewvalues.csv" )