**CSE3506 Essentials of Data Analytics**

Name : **Sparsh Raj**

Reg. No. : **19BPS1028**

Lab Exercise: 4 – Air Passengers

**Objective:** Store Airpassengers dataset (inbuilt dataset available in ‘R’) in a dataframe named “data”, install packages such as ‘forecast’, ‘tseries’. Perform the following

**Methods:**

1. Display entire dataset and check for unfilled data

(ii) Display the statistical info of the dataset such as min, max, 1st quartile, 3rd quartile, mean and median.

(iii) Plot ‘data’ (No. of Air passengers Vs Year)

(iv) Plot as timeseries ‘data’ (monthwise)

(v) Decompose the data as multiplicative and store as ‘ddata’

(vi) Plot ‘ddata’

(vii) Plot the following: trend, seasonal and random separately.

(viii) Perform ADF test for stationarity

(ix) Plot ACF and PACF

(x) Model using ARIMA

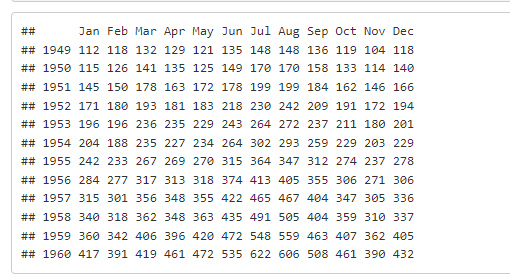
1. **Display entire dataset and check for unfilled data.**

**library**(tseries)

**library**(forecast)

data<-AirPassengers

data

****

sum(is.na(data))

## [1] 0

**(ii) Display the statistical info of the dataset such as min, max, 1st quartile, 3rd quartile, mean and median.**

dMin<-min(data)

print(paste("Min = ",dMin))

## [1] "Min = 104"

dMax<-max(data)

print(paste("Max = ",dMax))

## [1] "Max = 622"

dMean<-mean(data)

print(paste("Mean = ",dMean))

## [1] "Mean = 280.298611111111"

dMed<-median(data)

print(paste("Median = ",dMed))

## [1] "Median = 265.5"

quan<-quantile(data)

print("Quartiles are = ")

## [1] "Quartiles are = "

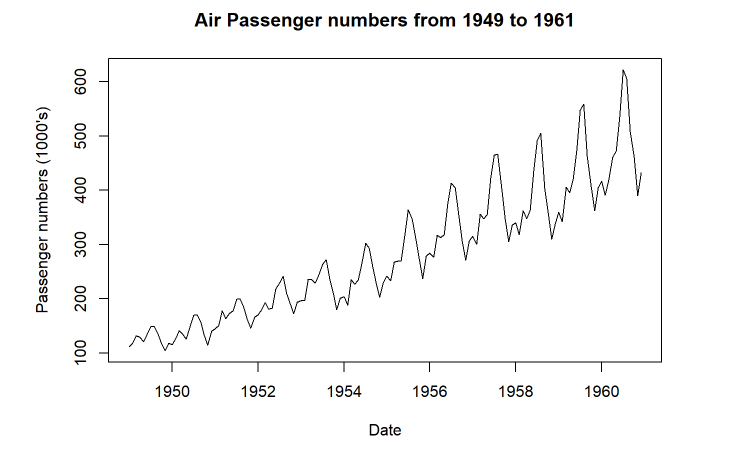
quan

## 0% 25% 50% 75% 100%

## 104.0 180.0 265.5 360.5 622.0

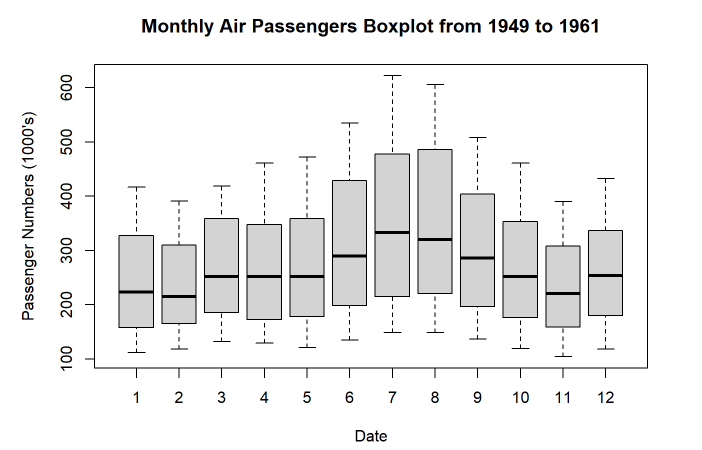
**(iii) Plot ‘data’ (No. of Air passengers Vs Year)**

plot(data,xlab="Date", ylab = "Passenger numbers (1000's)",main="Air Passenger numbers from 1949 to 1961")

****

**(iv) Plot as timeseries ‘data’ (monthwise)**

boxplot(data~cycle(data),xlab="Date", ylab = "Passenger Numbers (1000's)" ,main ="Monthly Air Passengers Boxplot from 1949 to 1961")

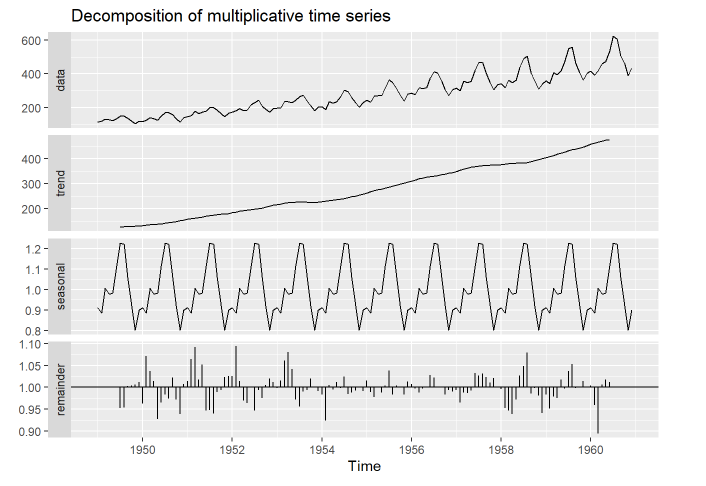
****

**(v) Decompose the data as multiplicative and store as ‘ddata’**

ddata <- decompose(data,"multiplicative")

**(vi) Plot ‘ddata’ & (vii) Plot the following: trend, seasonal and random separately.**

autoplot(ddata)

****

**(viii) Perform ADF test for stationarity**

adf.test(data)

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

##

## Augmented Dickey-Fuller Test

##

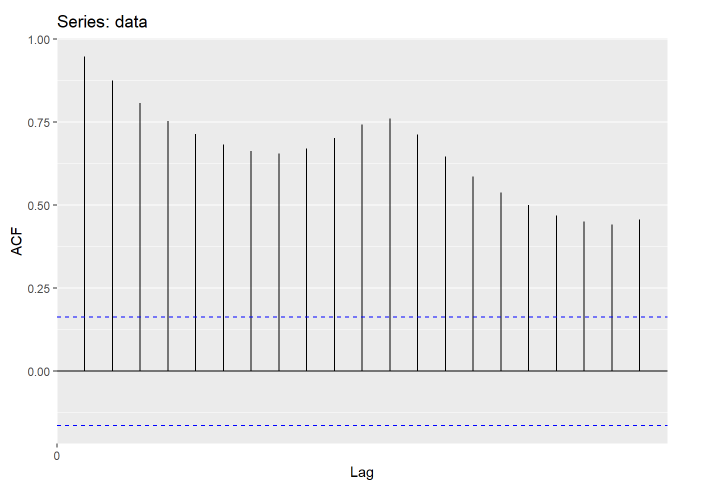
## data: data

## Dickey-Fuller = -7.3186, Lag order = 5, p-value = 0.01

## alternative hypothesis: stationary

**(ix) Plot ACF and PACF**

autoplot(acf(data,plot=FALSE))

****

ddata$random

## Jan Feb Mar Apr May Jun Jul

## 1949 NA NA NA NA NA NA 0.9516643

## 1950 0.9626030 1.0714668 1.0374474 1.0140476 0.9269030 0.9650406 0.9835566

## 1951 1.0138446 1.0640180 1.0918541 1.0176651 1.0515825 0.9460444 0.9474041

## 1952 1.0258814 1.0939696 1.0134734 0.9695596 0.9632673 1.0003735 0.9468562

## 1953 0.9976684 1.0151646 1.0604644 1.0802327 1.0413329 0.9718056 0.9551933

## 1954 0.9829785 0.9232032 1.0044417 0.9943899 1.0119479 0.9978740 1.0237753

## 1955 1.0154046 0.9888241 0.9775844 1.0015732 0.9878755 1.0039635 1.0385512

## 1956 1.0066157 0.9970250 0.9876248 0.9968224 0.9985644 1.0275560 1.0217685

## 1957 0.9937293 0.9649918 0.9881769 0.9867637 0.9924177 1.0328601 1.0261250

## 1958 0.9954212 0.9522762 0.9469115 0.9383993 0.9715785 1.0261340 1.0483841

## 1959 0.9825176 0.9505736 0.9785278 0.9746440 1.0177637 0.9968613 1.0373136

## 1960 1.0039279 0.9590794 0.8940857 1.0064948 1.0173588 1.0120790 NA

## Aug Sep Oct Nov Dec

## 1949 0.9534014 1.0022198 1.0040278 1.0062701 1.0118119

## 1950 0.9733720 1.0225047 0.9721928 0.9389527 1.0067914

## 1951 0.9397599 0.9888637 0.9938809 1.0235337 1.0250824

## 1952 0.9931171 0.9746302 1.0046687 1.0202797 1.0115407

## 1953 0.9894989 0.9934337 1.0192680 1.0009392 0.9915039

## 1954 0.9845184 0.9881036 0.9927613 0.9995143 0.9908692

## 1955 0.9831117 1.0032501 1.0003084 0.9827720 1.0125535

## 1956 1.0004765 1.0008730 0.9835071 0.9932761 0.9894251

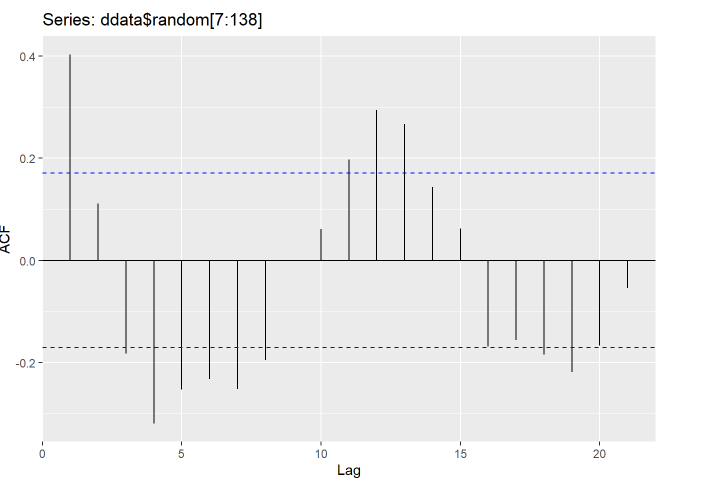
## 1957 1.0312668 1.0236147 1.0108432 1.0212995 1.0005263

## 1958 1.0789695 0.9856540 0.9977971 0.9802940 0.9405687

## 1959 1.0531001 0.9974447 1.0013371 1.0134608 0.9999192

## 1960 NA NA NA NA NA

autoplot(acf(ddata$random[7:138],plot=FALSE))

****

**(x) Model using ARIMA**

fitData <- auto.arima(data)

fitData

## Series: data

## ARIMA(2,1,1)(0,1,0)[12]

##

## Coefficients:

## ar1 ar2 ma1

## 0.5960 0.2143 -0.9819

## s.e. 0.0888 0.0880 0.0292

##

## sigma^2 = 132.3: log likelihood = -504.92

## AIC=1017.85 AICc=1018.17 BIC=1029.35

**library**(ggfortify)

## Warning: package 'ggfortify' was built under R version 4.1.2

## Loading required package: ggplot2

## Registered S3 methods overwritten by 'ggfortify':

## method from

## autoplot.Arima forecast

## autoplot.acf forecast

## autoplot.ar forecast

## autoplot.bats forecast

## autoplot.decomposed.ts forecast

## autoplot.ets forecast

## autoplot.forecast forecast

## autoplot.stl forecast

## autoplot.ts forecast

## fitted.ar forecast

## fortify.ts forecast

## residuals.ar forecast

ggtsdiag(fitData)

