504 Practical Assignment

Roll No: 17010724108

Md. Thouhidul Islam

October 26, 2022

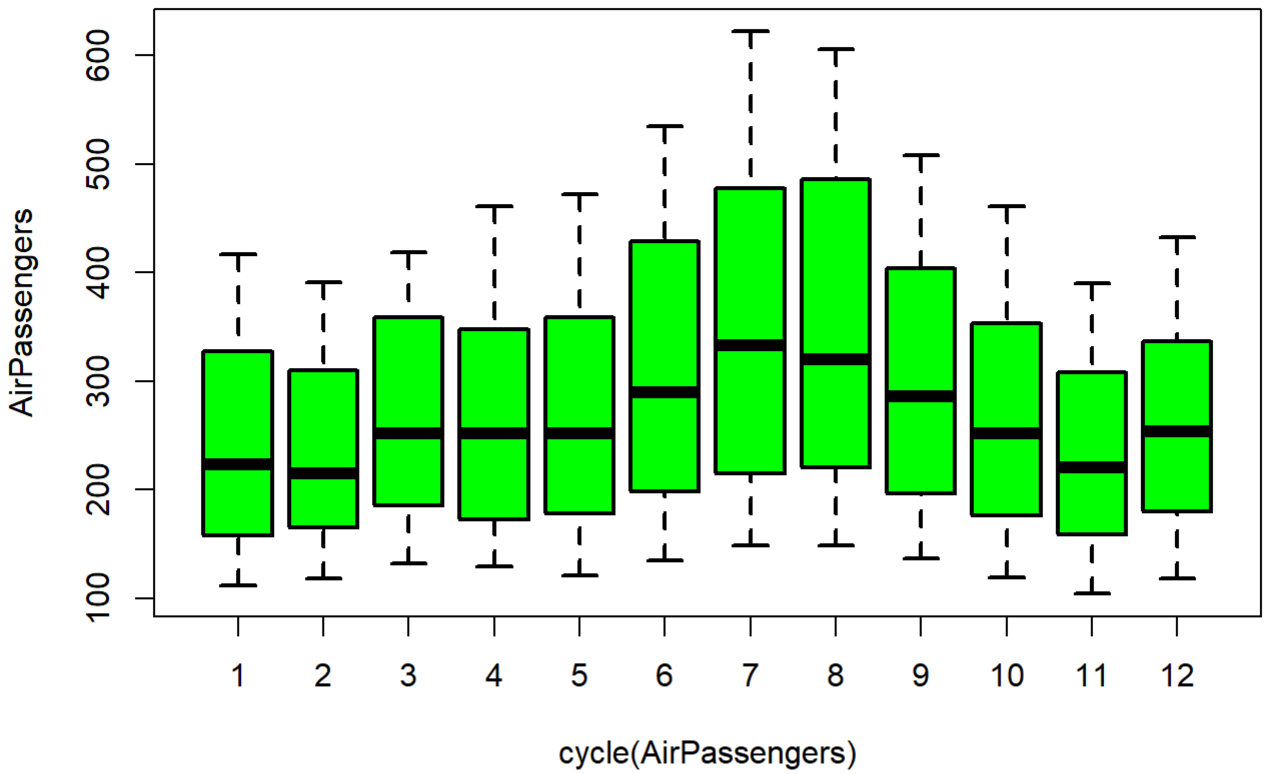
```
data("AirPassengers")
AirPassengers
```

```
##      Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec
## 1949 112 118 132 129 121 135 148 148 136 119 104 118
## 1950 115 126 141 135 125 149 170 170 158 133 114 140
## 1951 145 150 178 163 172 178 199 199 184 162 146 166
## 1952 171 180 193 181 183 218 230 242 209 191 172 194
## 1953 196 196 236 235 229 243 264 272 237 211 180 201
## 1954 204 188 235 227 234 264 302 293 259 229 203 229
## 1955 242 233 267 269 270 315 364 347 312 274 237 278
## 1956 284 277 317 313 318 374 413 405 355 306 271 306
## 1957 315 301 356 348 355 422 465 467 404 347 305 336
## 1958 340 318 362 348 363 435 491 505 404 359 310 337
## 1959 360 342 406 396 420 472 548 559 463 407 362 405
## 1960 417 391 419 461 472 535 622 606 508 461 390 432
```

We used `data` function for collecting the data from datasets of R. We may also upload file using `read.csv()` function if we need.

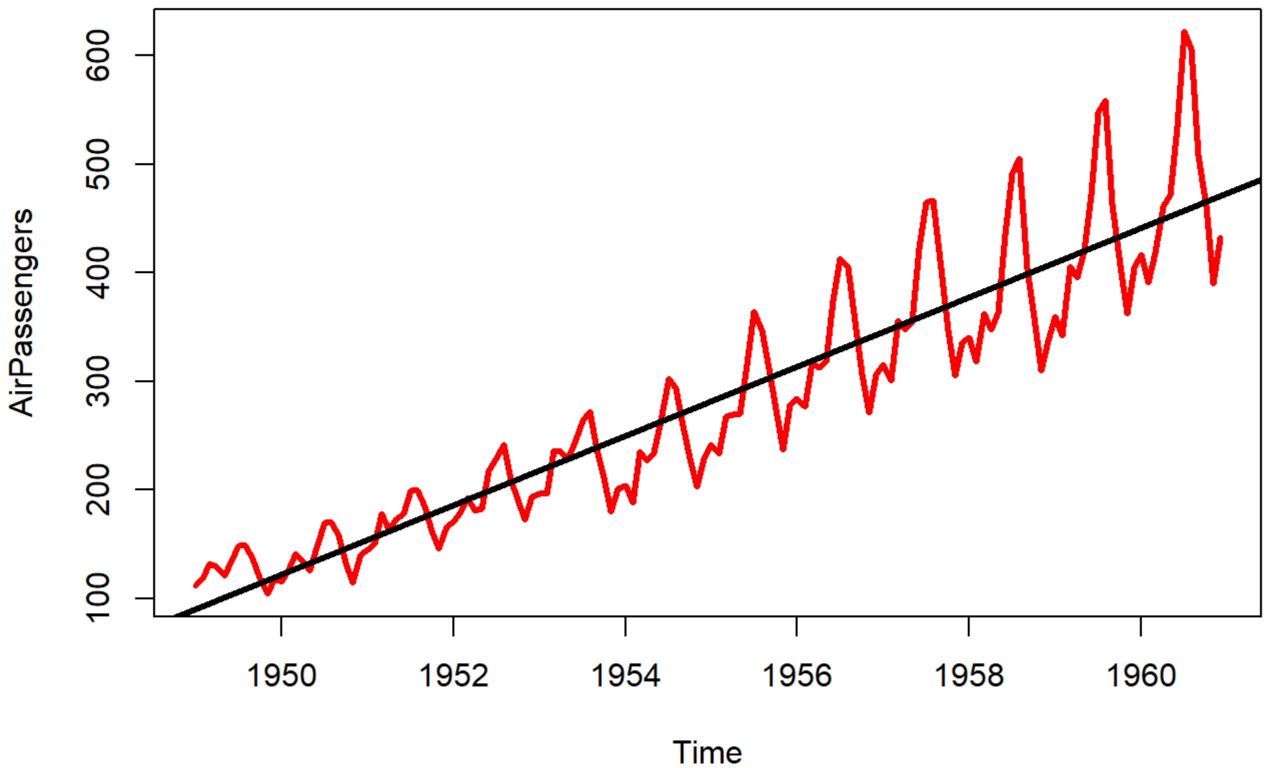
Plotting the data

```
boxplot(AirPassengers~cycle(AirPassengers),col="green",lwd=2)
```



From the **boxplot** we are seeing, Comparatively **high volume of passengers in July and August** than other months each year.

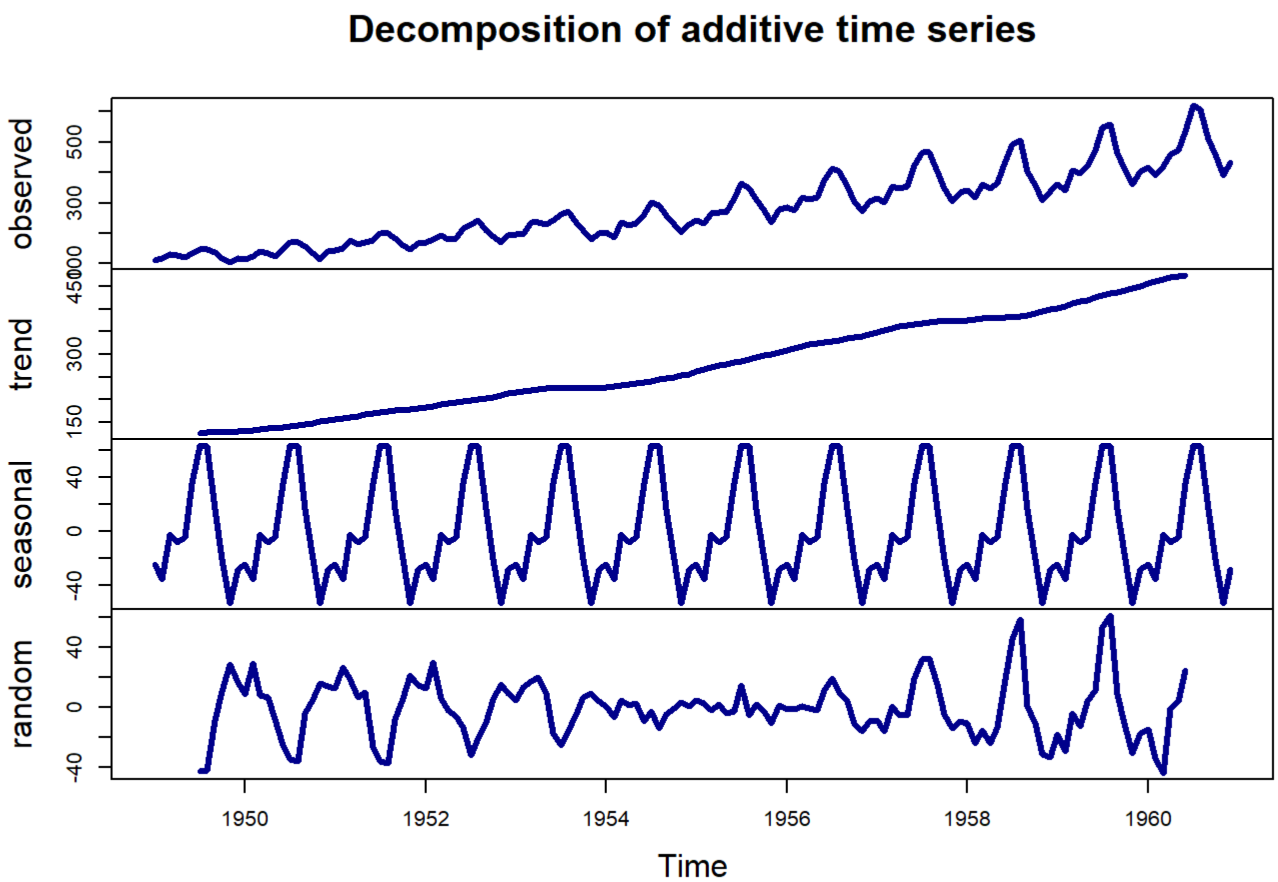
```
plot(AirPassengers,col="red",lwd=3)
abline(lm(AirPassengers~time(AirPassengers)),lwd=3)
```



This is a basic line plot from where we can see there is **upward trend** and **seasonality** on the data, we can confirm that and decompose that in next plot. Black line is the **regression line**.

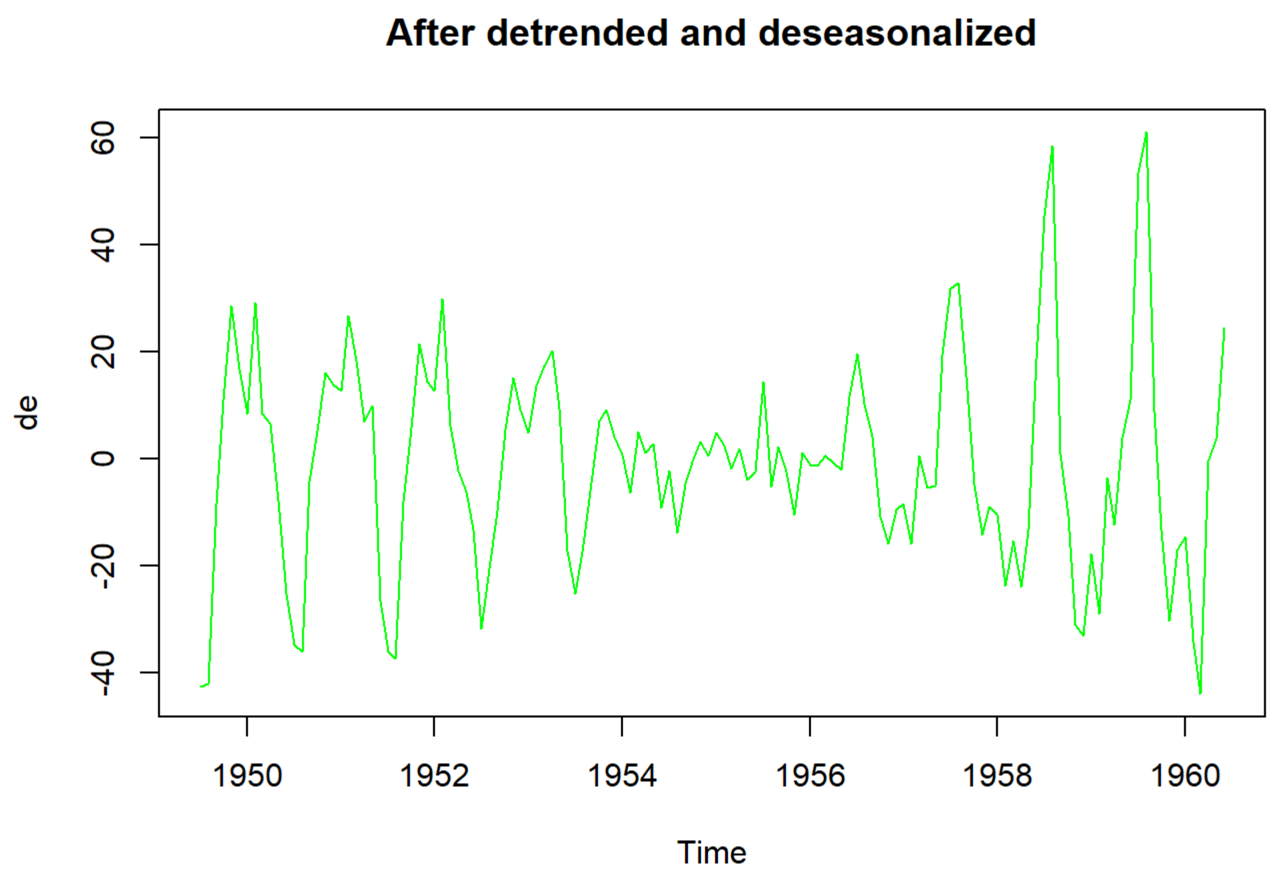
Time series decomposition into trend, seasonal and random (noise).

```
b=decompose(AirPassengers)
plot(b,lwd=3,col="darkblue")
```



When we `decompose` the data, we get *trend, seaonality and random* component from the data. Here we can confirmed that the data has *seasonality* and *upward trend*.

```
de=na.omit(b$random) #na.omit for deleting missing value
plot(de,col="green",main="After detrended and deseasonalized")
```



We could **de trended and deseaonalized** the original data by keeping only **random component**. Now we will check whether *de trended and deseaonalized* is **stationary** or not.

Stationary checking(de trended and deseaonalized data)

```
library(tseries) # for stationary checking
```

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

```
adf.test(de,k=12) #k= how many lags will we consider
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  de
## Dickey-Fuller = -4.5382, Lag order = 12, p-value = 0.01
## alternative hypothesis: stationary
```

From adf test we can say that **de trended and deseaonalized** is **stationary** as p<0.05 and out *alnernative hypothesis* is Stationary.

Stationary checking for original data

```
adf.test(AirPassengers, k=12)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  AirPassengers
## Dickey-Fuller = -1.5094, Lag order = 12, p-value = 0.7807
## alternative hypothesis: stationary
```

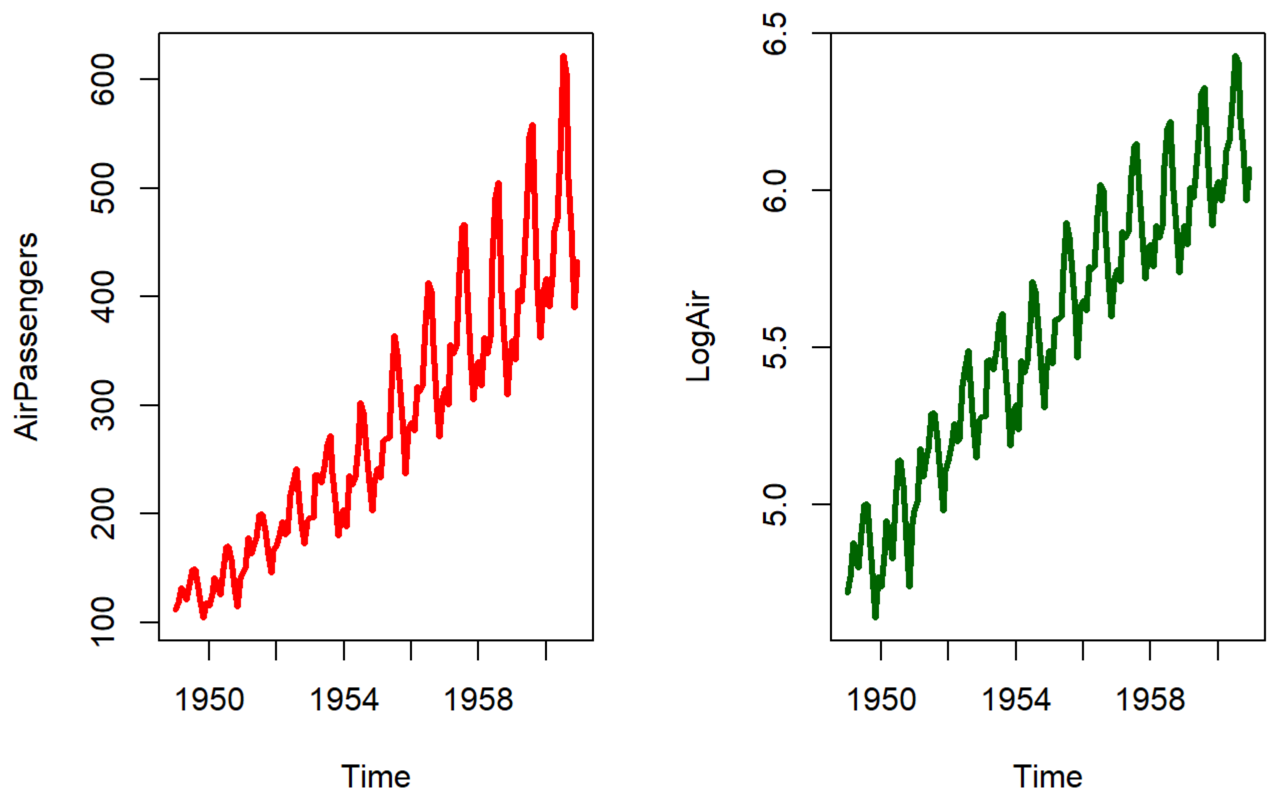
```
pp.test(AirPassengers,lshort = T)
```

```
##
##  Phillips-Perron Unit Root Test
##
## data:  AirPassengers
## Dickey-Fuller Z(alpha) = -46.406, Truncation lag parameter = 4, p-value
## = 0.01
## alternative hypothesis: stationary
```

From both test we can say that **Original data** is **non stationary** as p>0.05. So, the time series has a unit root.

Transform the data to stabilize the variance

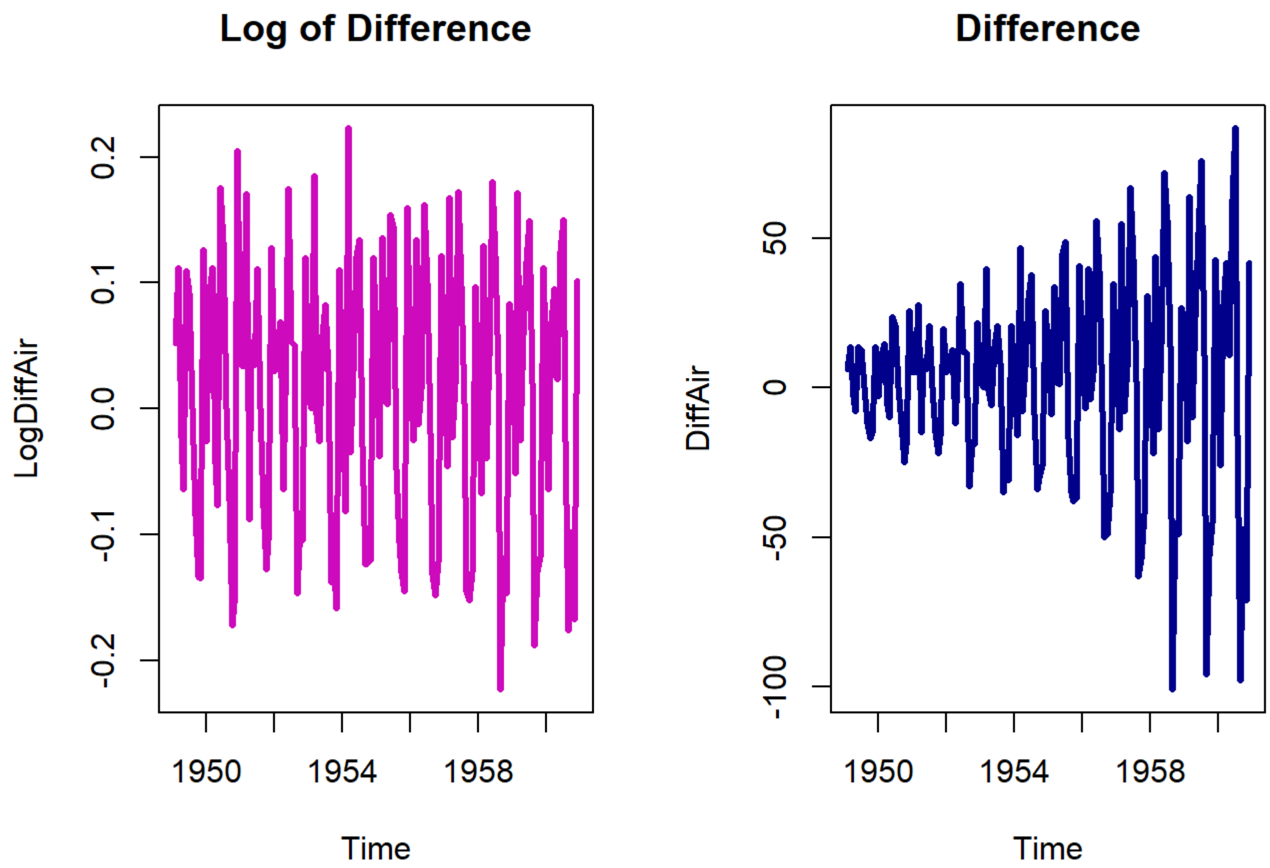
```
par(mfrow=c(1,2))
LogAir <- round(log(AirPassengers),2)
plot(AirPassengers,col="red",lwd=3)
plot(LogAir,lwd=3, col="darkgreen")
```



Transformation Stabilized the variance compared to orginial

Difference and Transformation

```
DiffAir<- diff(AirPassengers) #difference of main data
LogDiffAir <- diff(log(AirPassengers)) # log and difference of main data
par(mfrow=c(1,2))
plot(LogDiffAir,col=6,lwd=3,main="Log of Difference")
plot(DiffAir,col="darkblue",lwd=3,main="Difference")
```



From the plot we can now say that, Mean and Variance are approxiamtely constant So, Data is now may be Stationary . We will furthe check.

Stationary checking for Log Difference data

```
adf.test(LogDiffAir, k=12)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  LogDiffAir
## Dickey-Fuller = -3.3656, Lag order = 12, p-value = 0.06313
## alternative hypothesis: stationary
```

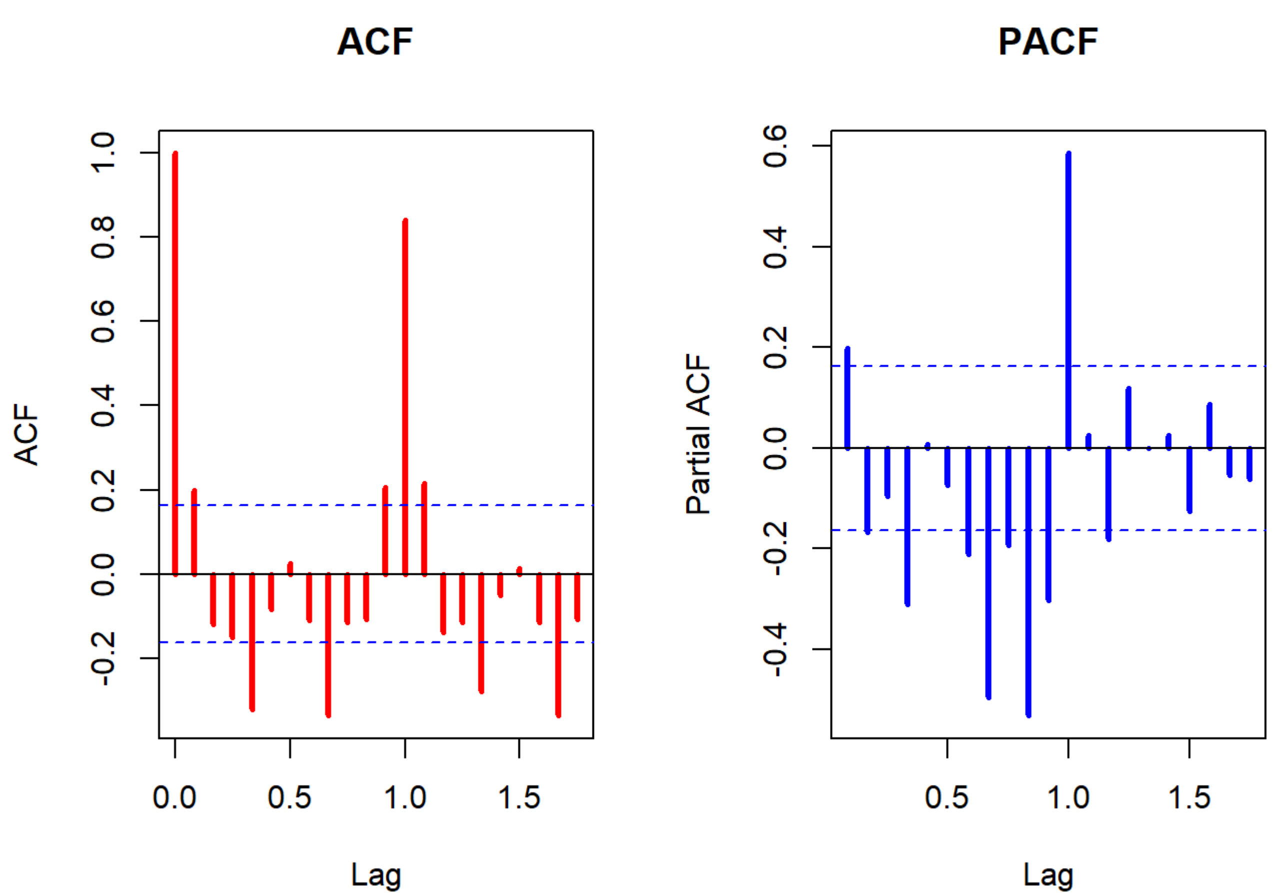
```
pp.test(LogDiffAir,lshort = T)
```

```
##
##  Phillips-Perron Unit Root Test
##
## data:  LogDiffAir
## Dickey-Fuller Z(alpha) = -93.215, Truncation lag parameter = 4, p-value
## = 0.01
## alternative hypothesis: stationary
```

From Adf test we are seeing data is **stationary at 7% level of signicance.** From PP test we can say that data is **stationary at 5% level of signicance.**

Model Selection using ACF and PACF function [ARIMA(p,d,q)]

```
par(mfrow=c(1,2))
acf(diff(log(AirPassengers)),col="red",lwd=3,main="ACF")
pacf(diff(log(AirPassengers)),col="blue",lwd=3,main="PACF")
```



As, we differentiate the data 1 time to make stationary so d=1

Determine the value **q (Order of MA)** from **ACF plot** Ignore lag=0, we get q=1 Ignore lag=1, we get Q=1

Determine the value **p (Order of AR)** from **PACF plot** Ignore lag=0, we get p=0 Ignore lag=1, we get P=1 ARIMA(0,1,1)

```
library(lmtest) #for autocorrelation
```

```
## Loading required package: zoo
```

```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
original = lm(AirPassengers~time(AirPassengers))
dwtest(original)
```

```
##
##  Durbin-Watson test
##
## data:  original
## DW = 0.53719, p-value < 2.2e-16
## alternative hypothesis: true autocorrelation is greater than 0
```

Original data have high **high positive autocorelation** as DW1<2.

```
transformed = lm(LogDiffAir~time(LogDiffAir))
dwtest(transformed)
```

```
##
##  Durbin-Watson test
##
## data:  transformed
## DW = 1.5959, p-value = 0.005883
## alternative hypothesis: true autocorrelation is greater than 0
```

Transformed data have comparatively lower **positive autocorelation** than original data as DW2 tends to 2.

Appropriate Model Selection and Parameter Estimation

```
model1 = arima(log(AirPassengers),c(0,1,1))
model2 = arima(log(AirPassengers),c(1,1,2))
model3 = arima(log(AirPassengers),c(2,1,2))

model4 = arima(log(AirPassengers),c(0,1,1),
                seasonal = list(order=c(0,1,1),period=12))
model5 = arima(log(AirPassengers),c(0,1,2),
                seasonal = list(order=c(0,1,2),period=12))
model6 = arima(log(AirPassengers),c(1,1,2),seasonal = list(order=c(1,1,2),period=12))

model = c("model1","model2","model3","model4","model5","model6")
AIC = round(c(model1$aic,model2$aic,model3$aic,model4$aic,model5$aic,model6$aic),2)
LL = c(model1$loglik,model2$loglik,model3$loglik,model4$loglik,model5$loglik,model6$loglik)

cbind(model,AIC, LL)
```

```
##      model    AIC       LL
## [1,] "model1" "-238.73" "121.362699551405"
## [2,] "model2" "-248.86" "128.432400496633"
## [3,] "model3" "-247.78" "128.889833336688"
## [4,] "model4" "-483.4"  "244.699530596796"
## [5,] "model5" "-480.16" "245.077935698224"
## [6,] "model6" "-478.16" "246.081707637908"
```

For model selection,lowest AIC or BIC is preferred.

Usually, AIC is positive, negative AIC indicates a lower degree of information loss than does a positive. If Likelihood is large then AIC will be likely negative.

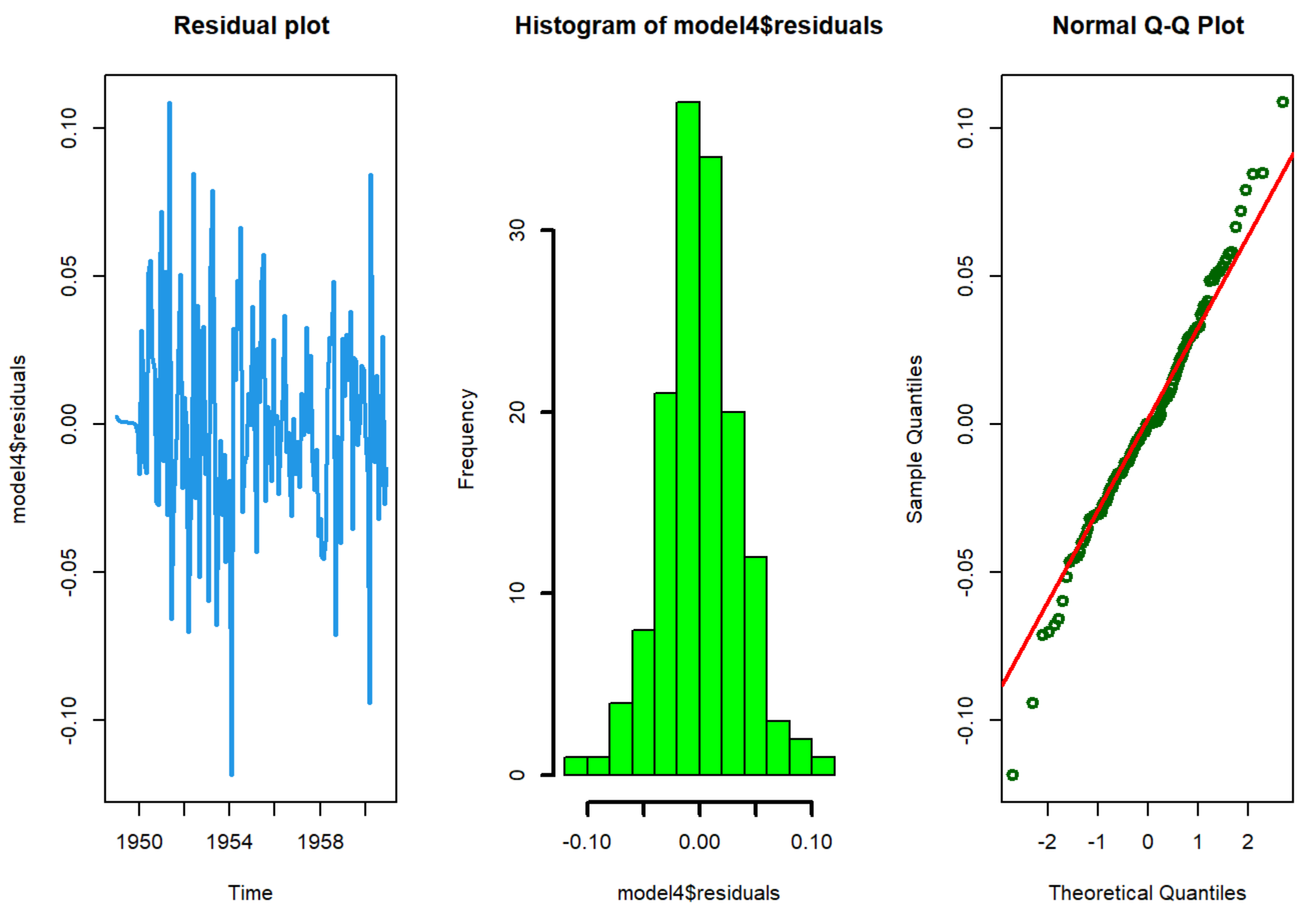
So, **model 4 is best**

```
model4
```

```
##
## Call:
## arima(x = log(AirPassengers), order = c(0, 1, 1), seasonal = list(order = c(0,
##     1, 1), period = 12))
##
## Coefficients:
##           ma1     sma1
##       -0.4018  -0.5569
## s.e.   0.0896   0.0731
##
## sigma^2 estimated as 0.001348:  log likelihood = 244.7,  aic = -483.4
```

Basic, Histogram and Q-Q plot of Residuals for checking Normality

```
par(mfrow=c(1,3))
plot(model4$residuals,col=4,lwd=2,main="Residual plot")
hist(model4$residuals, lwd=2,col="green")
qqnorm(residuals(model4), lwd=2,col="darkgreen")
qqline(residuals(model4), lwd=2,col="red")
```



From Histogram and QQ plot we can say that residual is approximately normal

Check the model of residuals

acf(model4$residuals, lwd = 3, col =” red ”)pacf(model4$residuals,lwd=3,col=“darkgreen”) # There is no significant spike of p and q # ARIMA(0,0,0)

ADF test of residuals

adf.test(model4$residuals,alternative =“stationary”,12)