Time series analysis BASIC

Suhail AK

20 April 2018

Time series analysis on airpassenger data set

```
library(MASS)
library(tseries)
library(stats)

air <- AirPassengers
frequency(air)
```

```
## [1] 12
```

```
class(air)
```

```
## [1] "ts"
```

```
start(air)
```

```
## [1] 1949    1
```

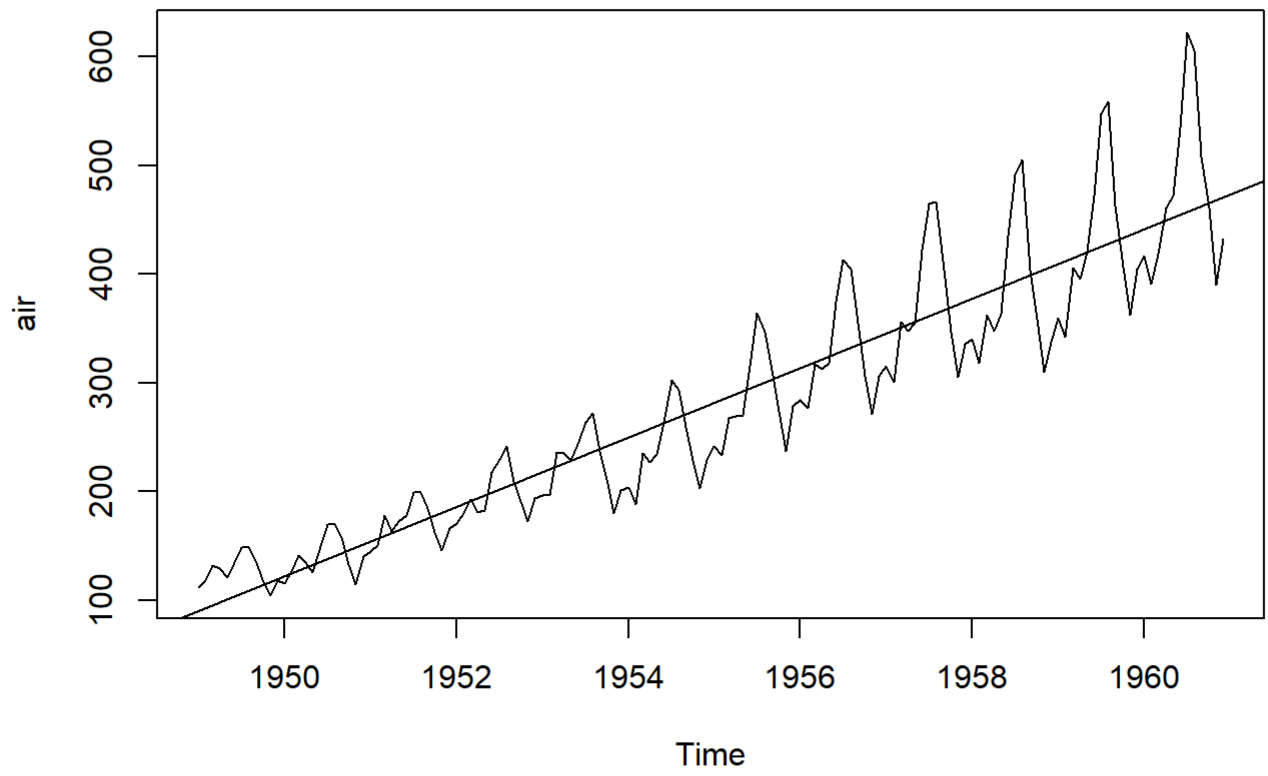
```
end(air)
```

```
## [1] 1960   12
```

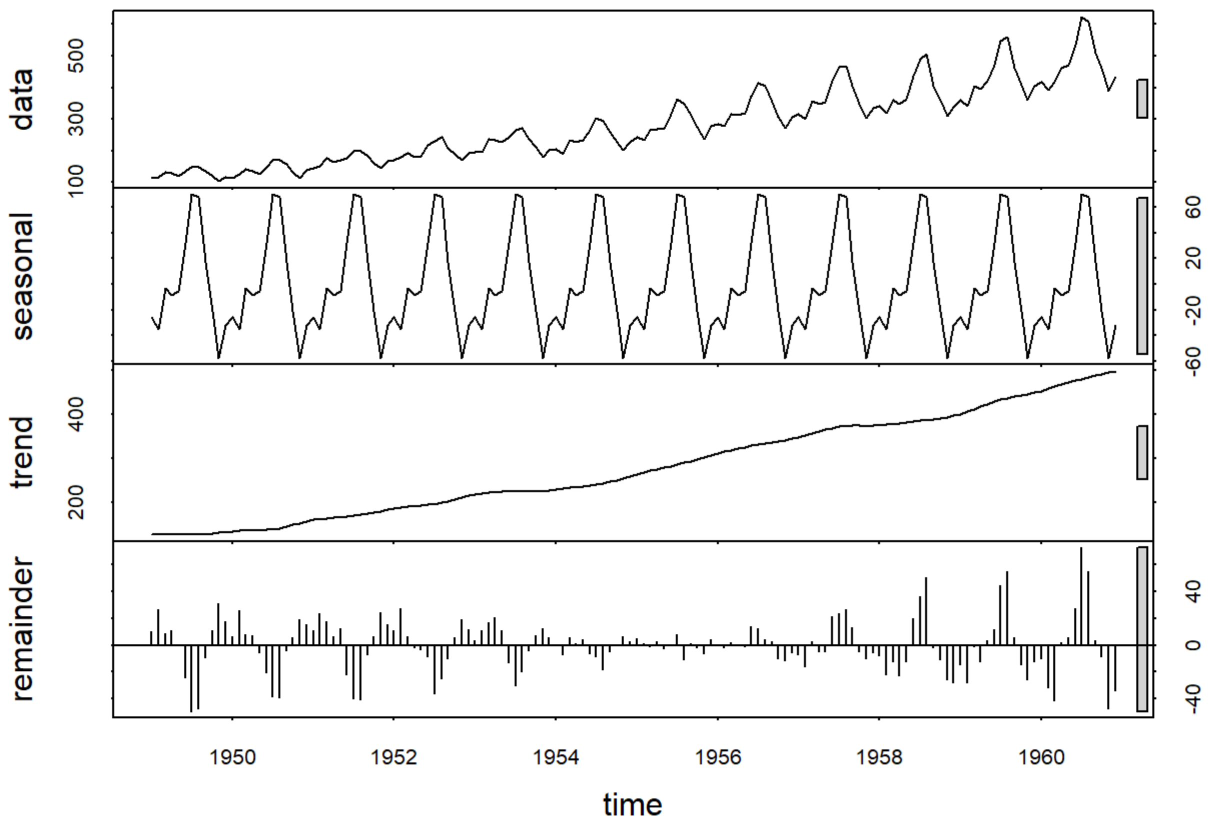
```
time(air)
```

```
##           Jan      Feb      Mar      Apr      May      Jun      Jul
## 1949 1949.000 1949.083 1949.167 1949.250 1949.333 1949.417 1949.500
## 1950 1950.000 1950.083 1950.167 1950.250 1950.333 1950.417 1950.500
## 1951 1951.000 1951.083 1951.167 1951.250 1951.333 1951.417 1951.500
## 1952 1952.000 1952.083 1952.167 1952.250 1952.333 1952.417 1952.500
## 1953 1953.000 1953.083 1953.167 1953.250 1953.333 1953.417 1953.500
## 1954 1954.000 1954.083 1954.167 1954.250 1954.333 1954.417 1954.500
## 1955 1955.000 1955.083 1955.167 1955.250 1955.333 1955.417 1955.500
## 1956 1956.000 1956.083 1956.167 1956.250 1956.333 1956.417 1956.500
## 1957 1957.000 1957.083 1957.167 1957.250 1957.333 1957.417 1957.500
## 1958 1958.000 1958.083 1958.167 1958.250 1958.333 1958.417 1958.500
## 1959 1959.000 1959.083 1959.167 1959.250 1959.333 1959.417 1959.500
## 1960 1960.000 1960.083 1960.167 1960.250 1960.333 1960.417 1960.500
##           Aug      Sep      Oct      Nov      Dec
## 1949 1949.583 1949.667 1949.750 1949.833 1949.917
## 1950 1950.583 1950.667 1950.750 1950.833 1950.917
## 1951 1951.583 1951.667 1951.750 1951.833 1951.917
## 1952 1952.583 1952.667 1952.750 1952.833 1952.917
## 1953 1953.583 1953.667 1953.750 1953.833 1953.917
## 1954 1954.583 1954.667 1954.750 1954.833 1954.917
## 1955 1955.583 1955.667 1955.750 1955.833 1955.917
## 1956 1956.583 1956.667 1956.750 1956.833 1956.917
## 1957 1957.583 1957.667 1957.750 1957.833 1957.917
## 1958 1958.583 1958.667 1958.750 1958.833 1958.917
## 1959 1959.583 1959.667 1959.750 1959.833 1959.917
## 1960 1960.583 1960.667 1960.750 1960.833 1960.917
```

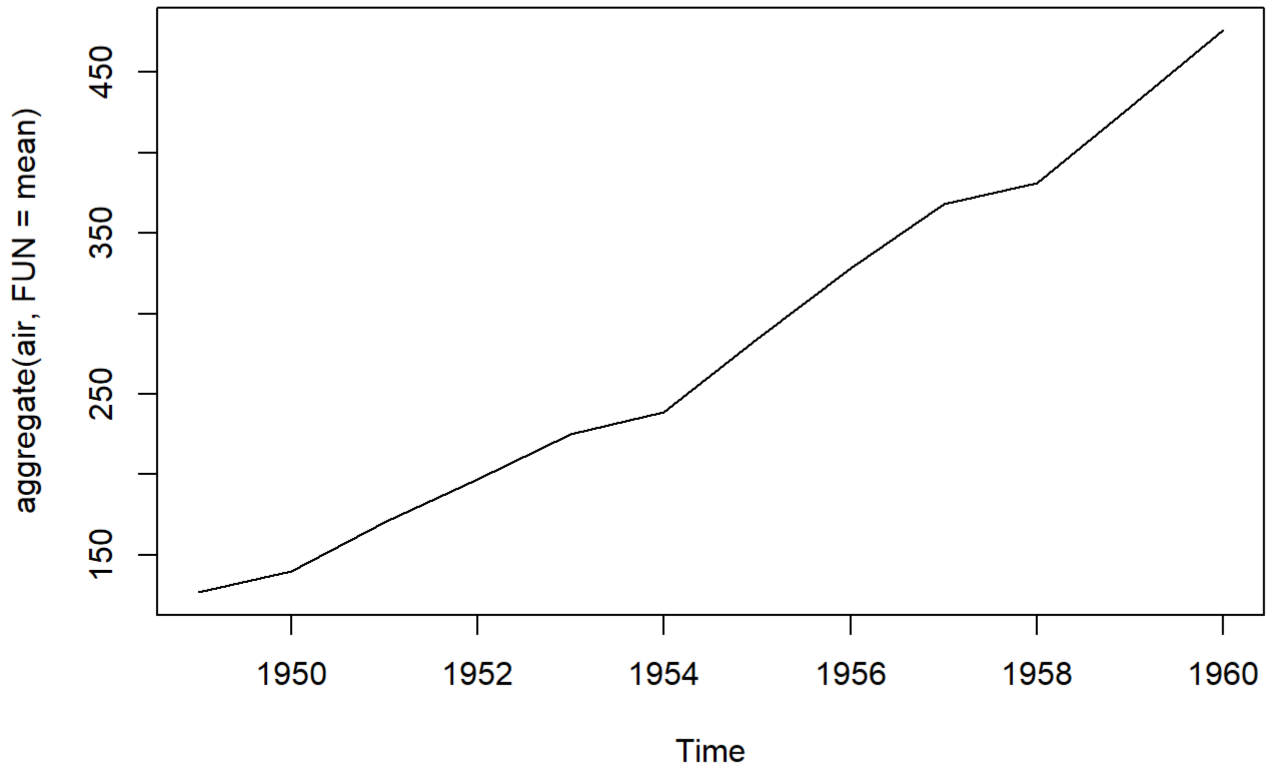
```
View(AirPassengers)

plot(air)
abline(reg=lm(air ~time(air)))
```




```
stlval <- stl(air,s.window = "periodic")
plot(stlval)
```



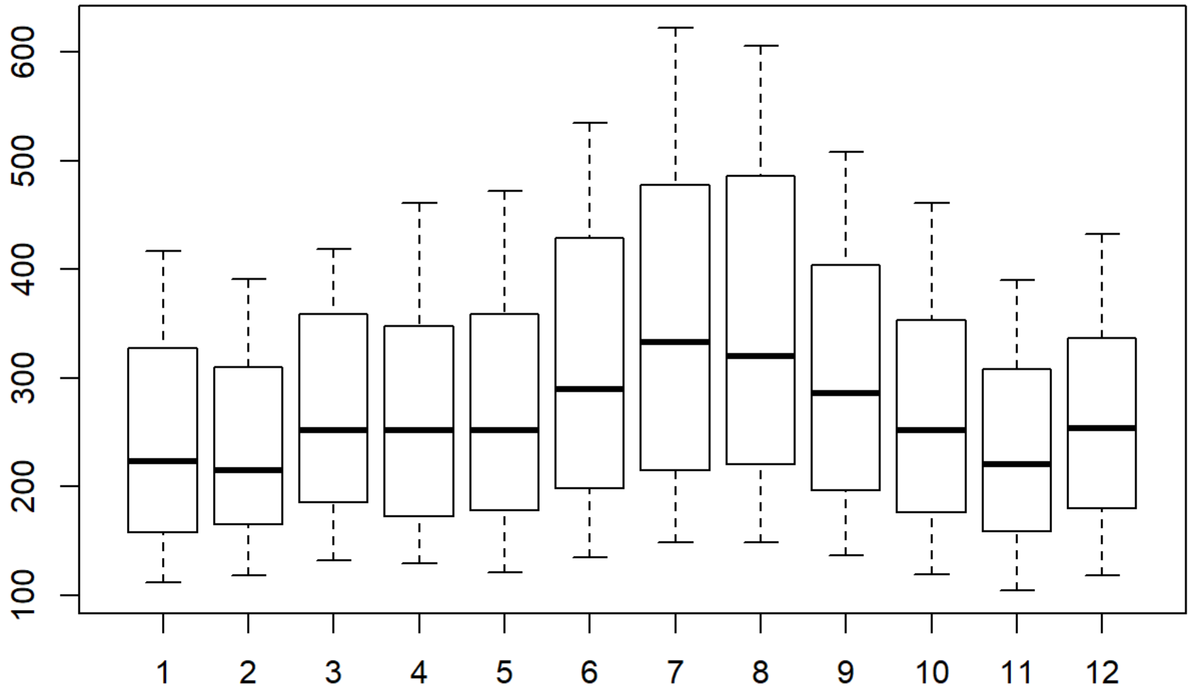
```
#to plot trendline
plot(aggregate(air,FUN = mean))
```



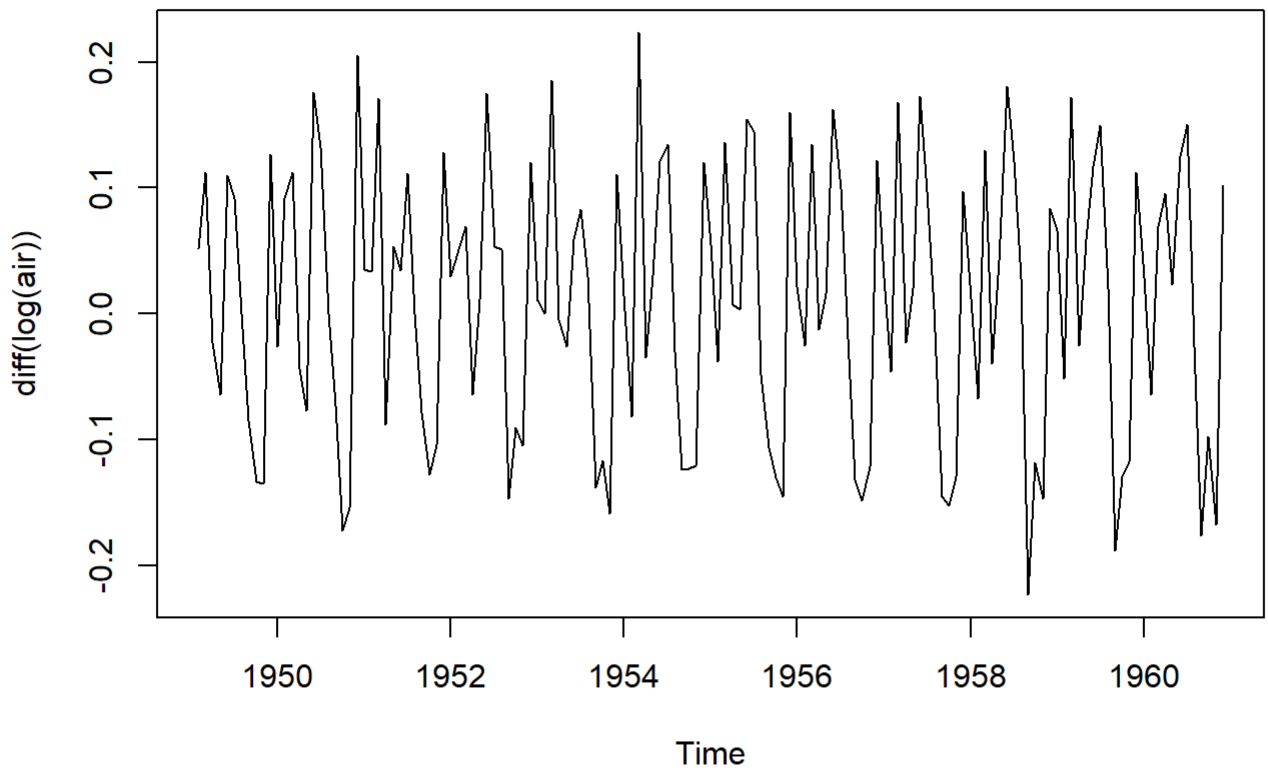

```
class(air)
```

```
## [1] "ts"
```

```
#for seasonality check use boxplot
boxplot(air~cycle(air))
```




```
#now to make varience equal we use log function

station <- plot(diff(log(air)))
```




```
stationair <- diff(log(air))

#now to check if the data is Stationary or not (use adf testing)
adf.test(stationair,alternative = "stationary")
```

```
## Warning in adf.test(stationair, alternative = "stationary"): p-value
## smaller than printed p-value
```

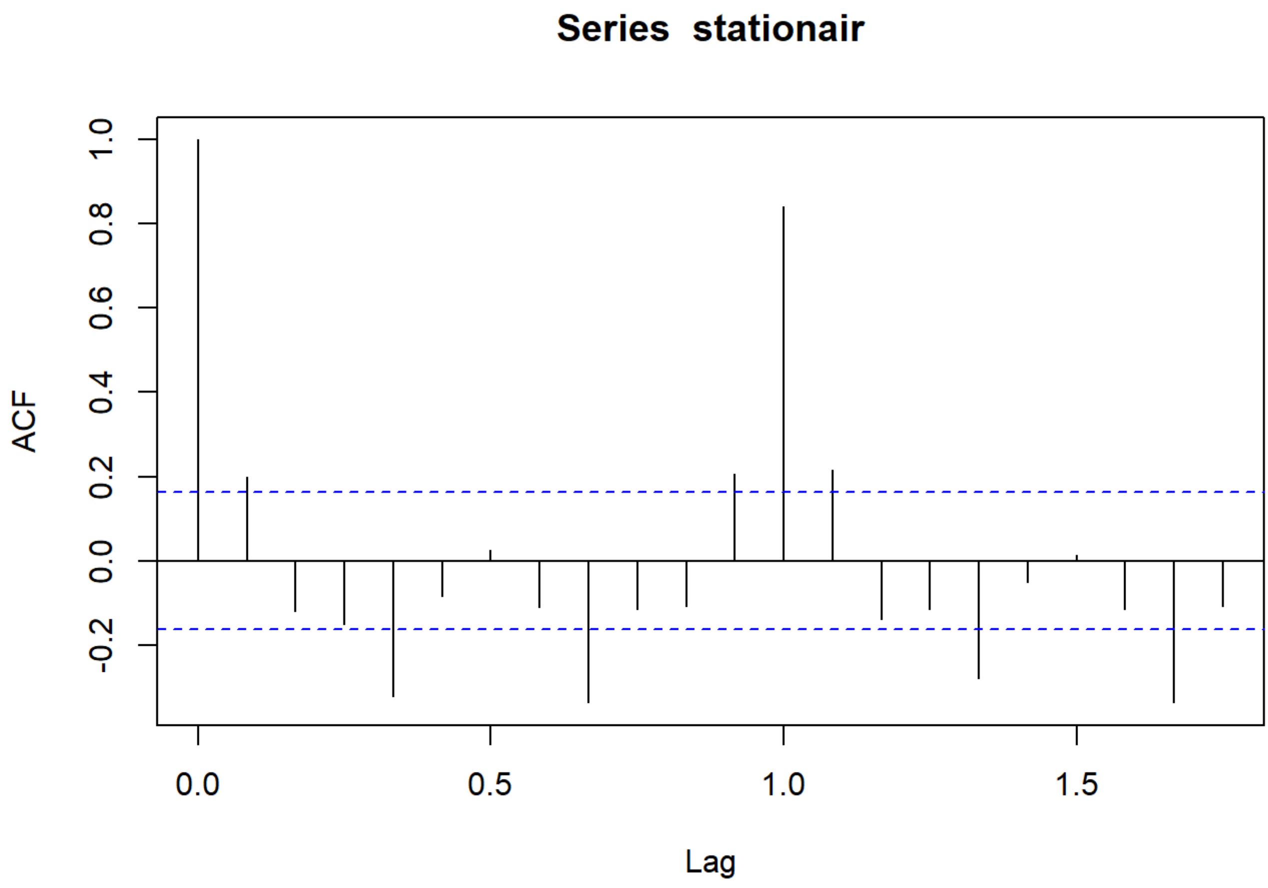
```
##
##  Augmented Dickey-Fuller Test
##
## data:  stationair
## Dickey-Fuller = -6.4313, Lag order = 5, p-value = 0.01
## alternative hypothesis: stationary
```

Once the data is stationary then apply time series model like eg: ARIMA model - ARIMA Model AR I MA AR-auto regression (p-value coeffecient) I- integration (d value) MA- moving average (q value)

to find q value using autocorrelation function

ie find the line before the inverted line index in this case from the graph the value of q is 1

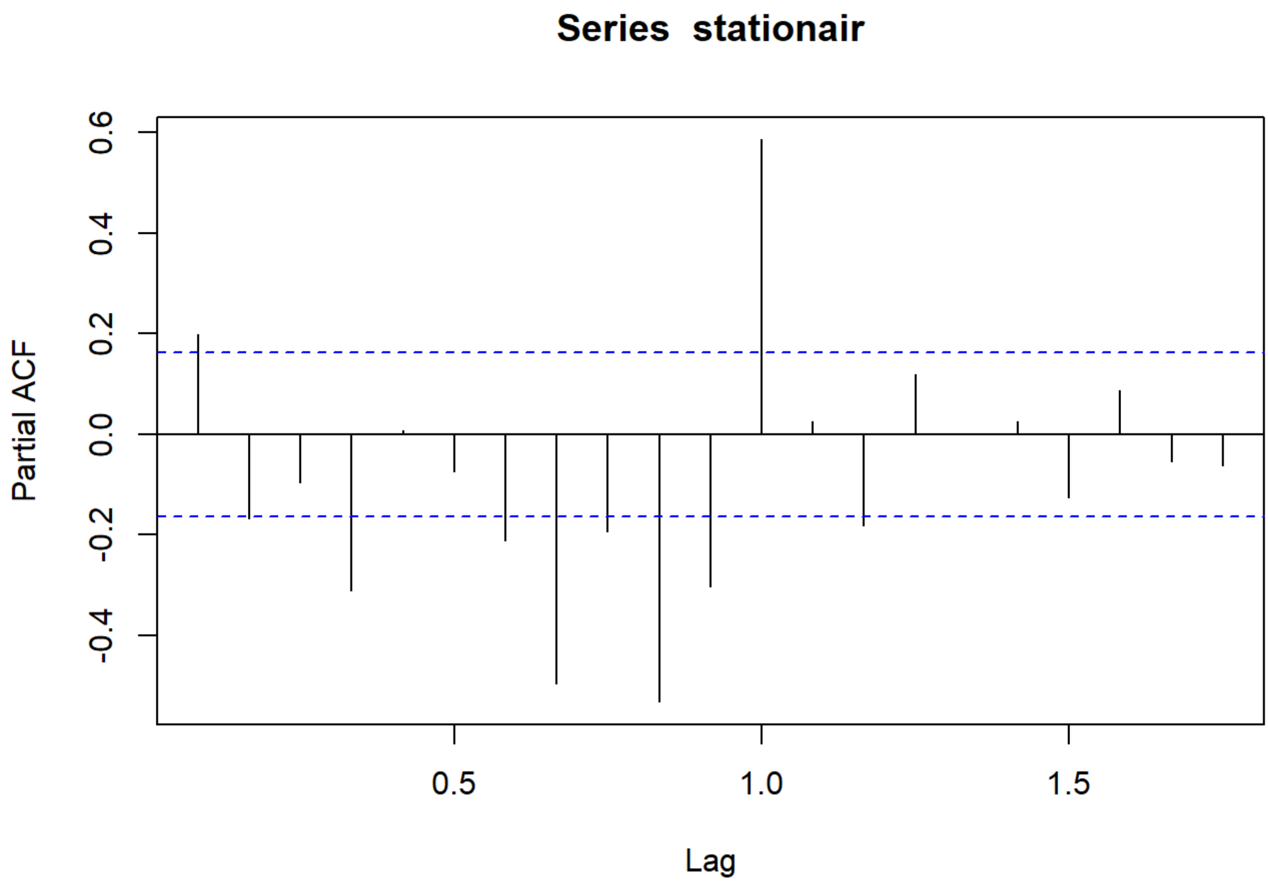
```
#using autocorrelation function to predict q value
acf(stationair)
```



similarly to find P coeffecient using partial autocorrealtion function ie. from the graph the value of P coeffecient is 0

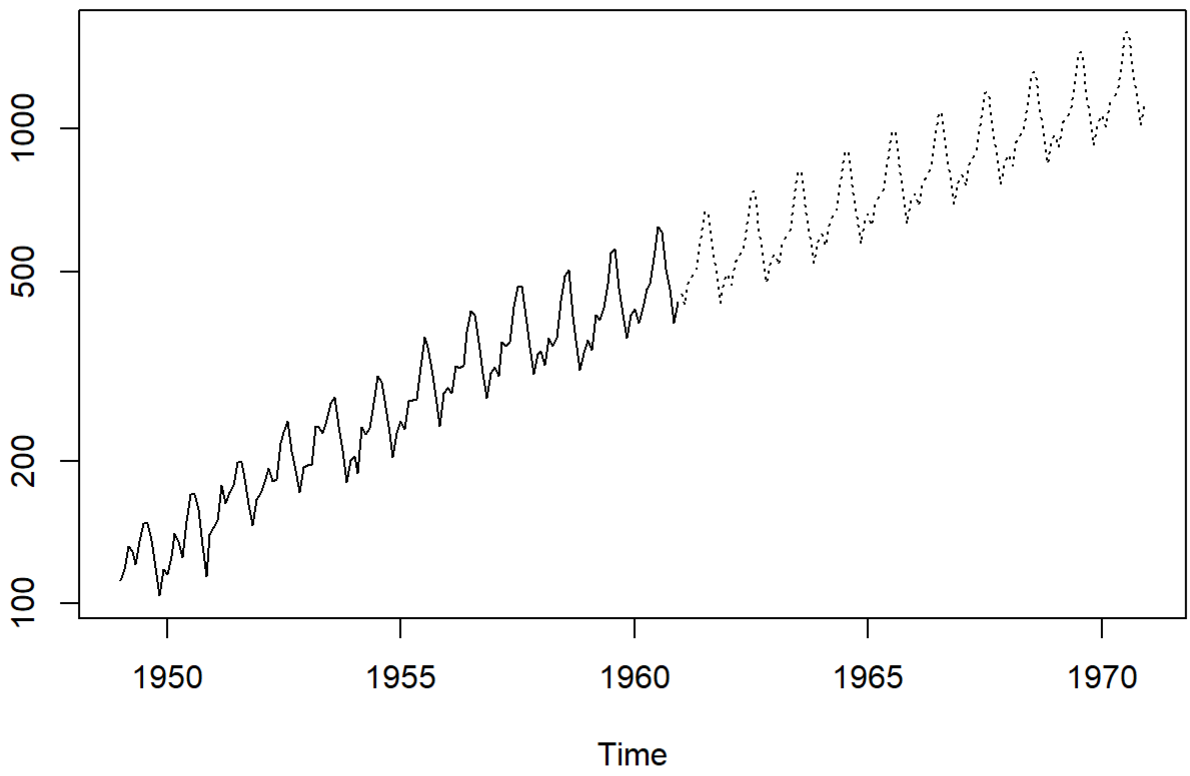
also to find coeffecient d we have to check how many times we used differentiation(diff()) to make mean constant. hence in this case we differentiated once hence d=1

```
pacf(stationair)
```



Now applying ARIMA model ie c(p,d,q) in this case (0,1,1) in predict function fit is a model and n.ahead =(no of years)*frequency/period and in pred1 converting the values from log to decimal form using e^

```
fit <- arima(log(air),c(0,1,1),list(order=c(0,1,1),period=12))
pred <- predict(fit,n.ahead = 10*12)

pred1 <- 2.718^pred$pred

ts.plot(air,pred1,log="y",lty=c(1,3))
```



Now to test the model for the year 1960

```
air1=ts(air,frequency = 12,start = c(1949,1),end = c(1959,12))
air1
```

```
##      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
```

```
fit1 <- arima(log(air1),c(0,1,1),seasonal=list(order=c(0,1,1),period=12))
pred2 <- predict(fit1,n.ahead=10*12)
pred2deci <- 2.718^pred2$pred
data1 <- head(pred2deci,12)
data1
```

```
##  [1] 419.0628 398.6732 466.2820 454.1188 472.9611 546.7614 621.8017
##  [8] 629.7291 526.4044 461.9958 406.3747 452.0098
```

```
pred_data1960 <- round(data1,0)
actual_data1960 <- tail(air,12)
df <- data.frame(pred_data1960,actual_data1960)
df
```

```
##    pred_data1960 actual_data1960
## 1            419             417
## 2            399             391
## 3            466             419
## 4            454             461
## 5            473             472
## 6            547             535
## 7            622             622
## 8            630             606
## 9            526             508
## 10           462             461
## 11           406             390
## 12           452             432
```