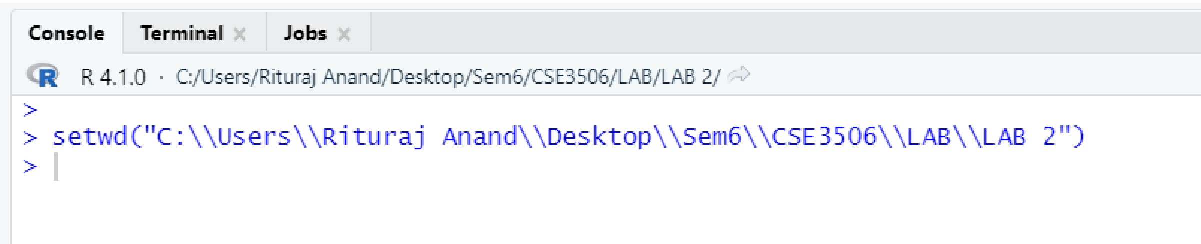
Programme	:	**B.Tech – ECE and ECM**	Semester	:	**Win 2022**
Course	:	**Essentials of Data Analytics Lab**	Code	:	**CSE3506**
Faculty	:	**Gobinath N**	Slot	:	**L51 + L52**

TimeSeries_Forecasting_Ex.02

Basic Commands:

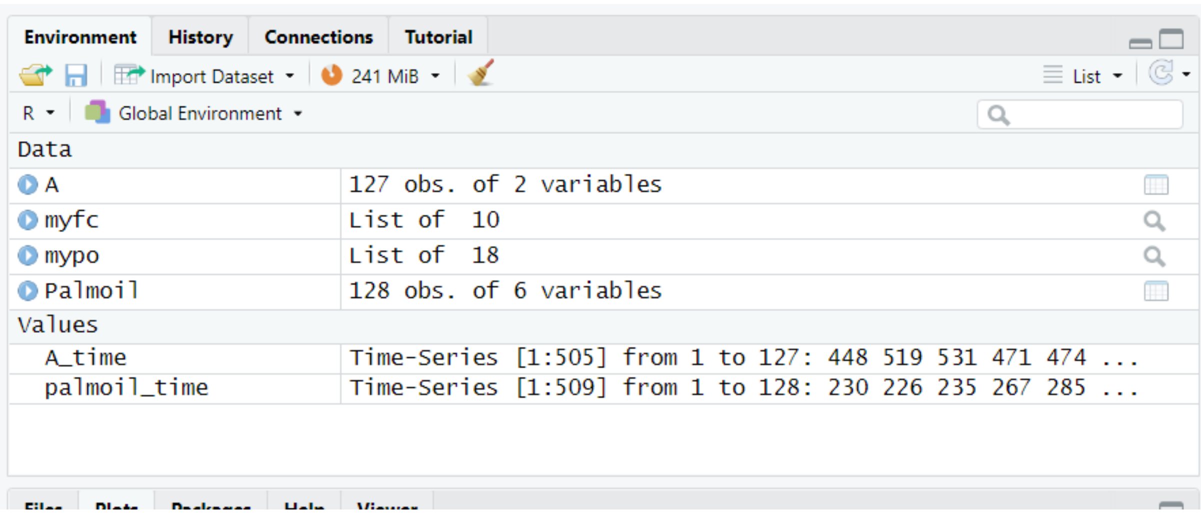
#Setting the working directory:

```
setwd("C:\\Users\\Rituraj Anand\\Desktop\\Sem6\\CSE3506\\LAB\\LAB 2")
```



#Reading the csv File:

```
A=read.csv("Soyaoil.csv")
```



#Installing the required packages for forecasting

#install.packages('forecast')

#install.packages('tseries')

#To check the class of the csv file,

class(Palmoil)

```
> #install.packages('tseries')
> class(Palmoil)
[1] "data.frame"
```

Since this is a dataframe type, we have to change it into time series by using:

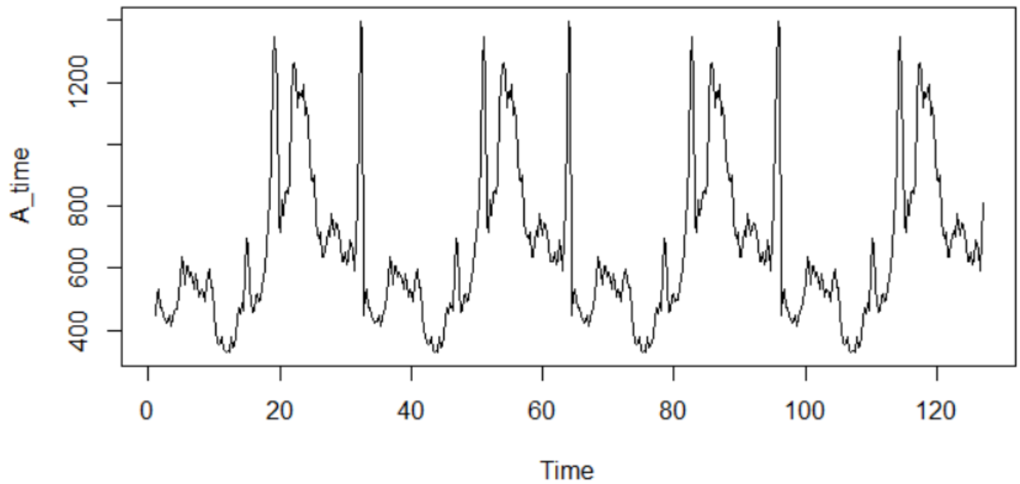
A_time=ts(A$Dollar,start=1,end=127,frequency = 4)

class(A_time)

```
> A=read.csv("Soyaoil.csv")
> class(A)
[1] "data.frame"
> A_time=ts(A$Dollar,start=1,end=127,frequency = 4)
> class(A_time)
[1] "ts"
```

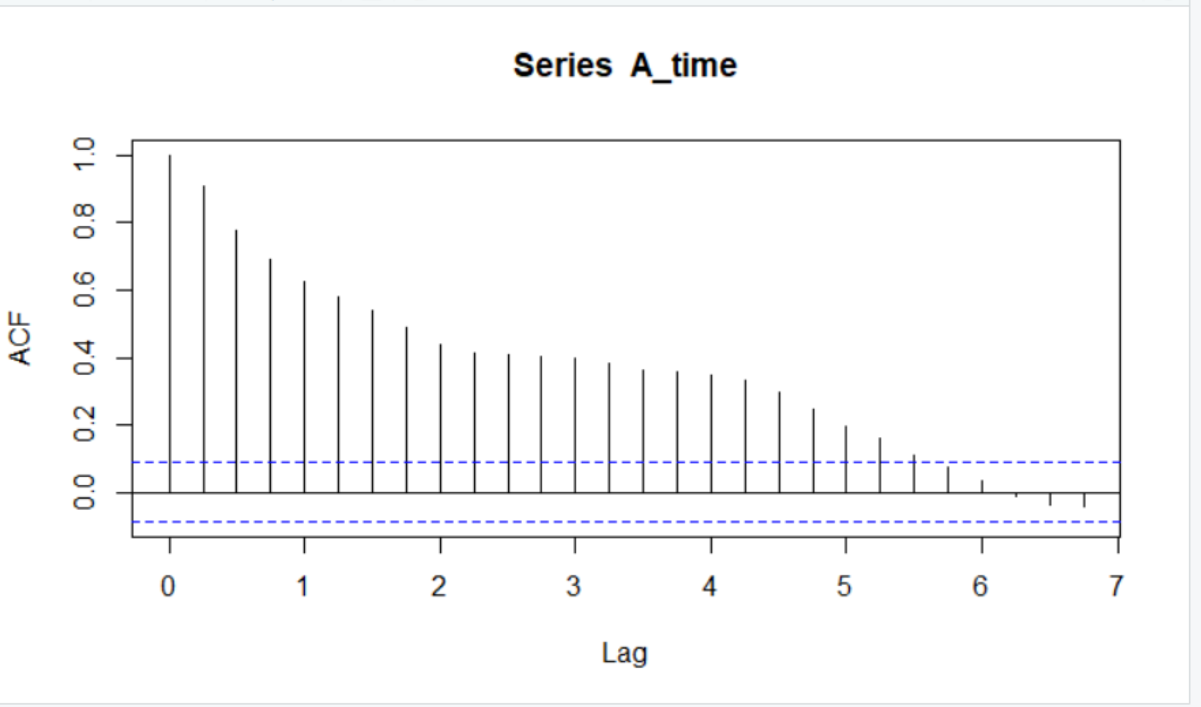
Plotting the A_time:

plot(A_time)



To find the autocorrelation function of A_time:

acf(A_time)

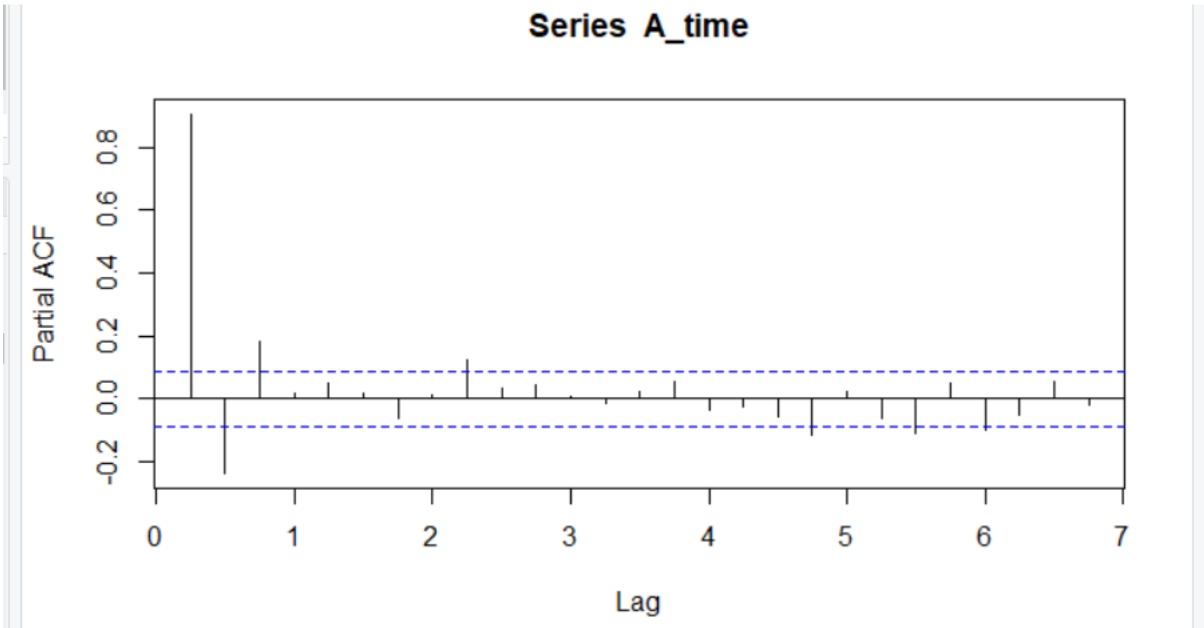


A “lag” is a fixed amount of passing time; One set of observations in a time series is plotted (lagged) against a second, later set of data. The kth lag is the time period that happened “k” time points before time i.

The blue dotted lines, the range is called set bars

<u>Partial ACF:</u>

pacf(A_time)



Relationship between y(t) and y(t-4)

What is ADF test used for?

=>Augmented Dickey Fuller test (ADF Test) is a common statistical test used to test whether a given Time series is stationary or not. It is one of the most commonly used statistical test when it comes to analysing the stationary of a series.

So, we checked if the time series is stationary or not

```
> adf.test(A_time)

        Augmented Dickey-Fuller Test

data:  A_time
Dickey-Fuller = -4.3205, Lag order = 7, p-value = 0.01
alternative hypothesis: stationary
```

If p<=0.05, it accepts the hypothesis else it rejects

In general, a p-value of less than 5% means you can reject the null hypothesis that there is a unit root. You can also compare the calculated DFT statistic with a tabulated critical value. If the DFT statistic is more negative than the table value, reject the null hypothesis of a unit root.

Now,

Since our data is stationary, we don't have to go for differencing, we can directly proceed for ARIMA test.

```
mypo=auto.arima(A_time,ic="aic",trace=TRUE)
```

here, ic = information criteria

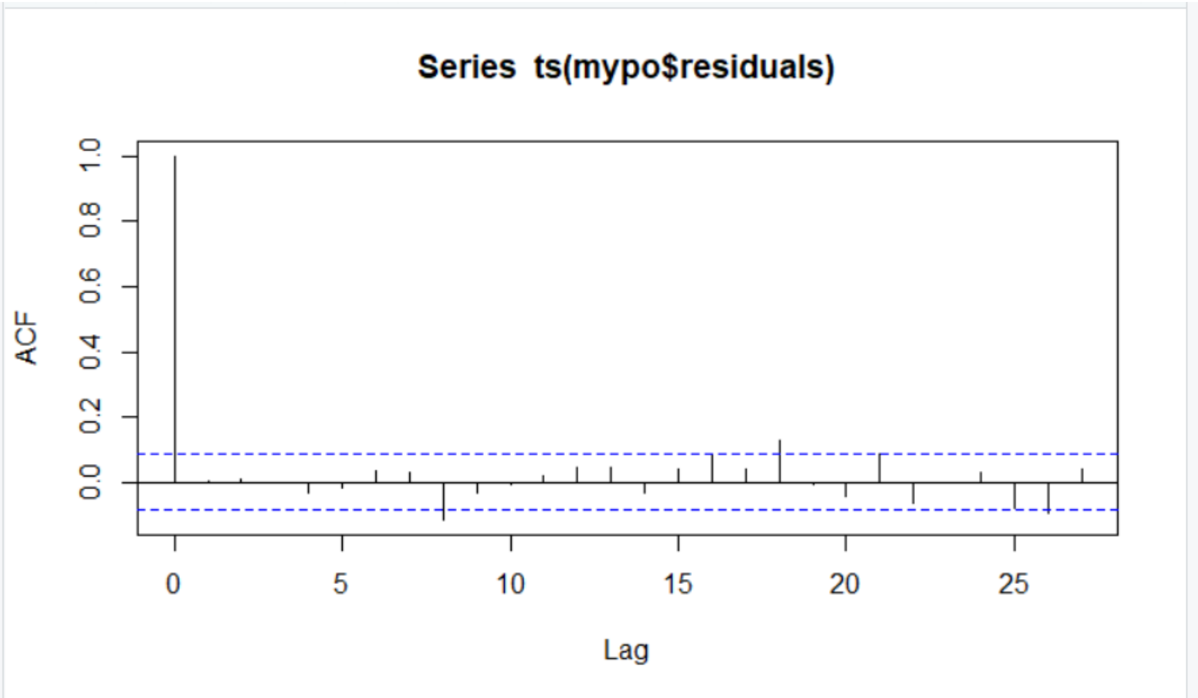
```
> #myfc=forecast(A_time,level=c(95),h=10*4)
> #myfc
> #plot(myfc)
> mypo=auto.arima(A_time,ic="aic",trace=TRUE)

 Fitting models using approximations to speed things up...

 ARIMA(2,0,2)(1,0,1)[4] with non-zero mean : 6108.624
 ARIMA(0,0,0)           with non-zero mean : 7019.745
 ARIMA(1,0,0)(1,0,0)[4] with non-zero mean : 6151.122
 ARIMA(0,0,1)(0,0,1)[4] with non-zero mean : 6380.234
 ARIMA(0,0,0)           with zero mean     : 8068.289
 ARIMA(2,0,2)(0,0,1)[4] with non-zero mean : 6105.743
 ARIMA(2,0,2)           with non-zero mean : 6103.88
 ARIMA(2,0,2)(1,0,0)[4] with non-zero mean : 6108.251
 ARIMA(1,0,2)           with non-zero mean : 6105.768
 ARIMA(2,0,1)           with non-zero mean : 6107.544
 ARIMA(3,0,2)           with non-zero mean : 6109.294
 ARIMA(2,0,3)           with non-zero mean : 6105.005
 ARIMA(1,0,1)           with non-zero mean : 6105.733
 ARIMA(1,0,3)           with non-zero mean : 6102.917
 ARIMA(1,0,3)(1,0,0)[4] with non-zero mean : 6121.404
 ARIMA(1,0,3)(0,0,1)[4] with non-zero mean : 6120.712
 ARIMA(1,0,3)(1,0,1)[4] with non-zero mean : 6122.329
 ARIMA(0,0,3)           with non-zero mean : 6235.124
 ARIMA(0,0,2)           with non-zero mean : 6309.543
 ARIMA(1,0,3)           with zero mean     : Inf

 Now re-fitting the best model(s) without approximations...

 ARIMA(1,0,3)           with non-zero mean : 6104.676

 Best model: ARIMA(1,0,3)           with non-zero mean
```

ARIMA(p,d,q) here, the model chose ARIMA(1,0,1) as best model without approximations,

As data is stationary, d=0, lag and error part is 1

```
> mypo
Series: A_time
ARIMA(1,0,3) with non-zero mean

Coefficients:
         ar1     ma1      ma2      ma3      mean
      0.9128  0.2548  -0.1603  -0.1188  663.9200
s.e.  0.0268  0.0537   0.0612   0.0541   49.1242

sigma^2 = 10228:  log likelihood = -3046.34
AIC=6104.68   AICc=6104.84   BIC=6130.02
```

acf(ts(mypo$residuals))

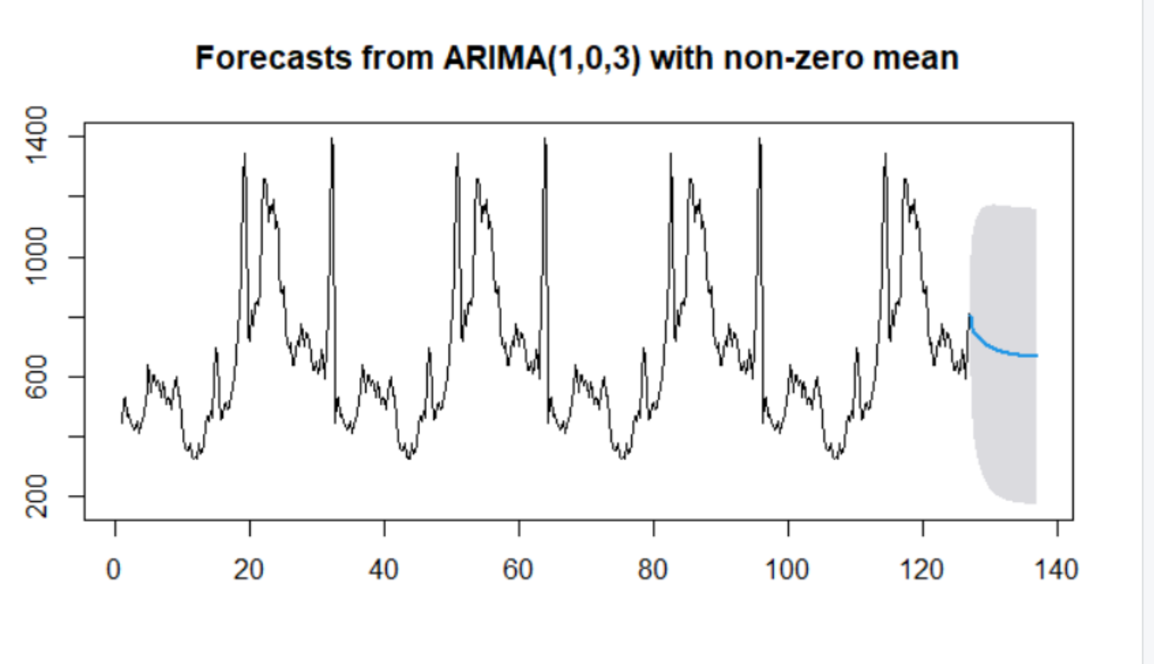


myfc=forecast(mypo,level=c(95),h=10*4)

here, h is the amount of time we are going to forecast, we are going to forecast for 10 years and each year has 4 quarters., so h=10*4;

```
> acf(ts(mypo$residuals))
> myfc=forecast(mypo,level=c(95),h=10*4)
> myfc
        Point Forecast    Lo 95    Hi 95
127 Q2        805.2846 607.0660 1003.503
127 Q3        767.6930 462.9725 1072.413
127 Q4        749.4673 395.8245 1103.110
128 Q1        742.0042 361.5589 1122.450
128 Q2        735.1922 333.7812 1136.603
128 Q3        728.9745 310.8984 1147.051
128 Q4        723.2992 291.8301 1154.768
129 Q1        718.1190 275.8013 1160.437
129 Q2        713.3907 262.2339 1164.548
129 Q3        709.0749 250.6840 1167.466
129 Q4        705.1356 240.8039 1169.467
130 Q1        701.5400 232.3162 1170.764
130 Q2        698.2581 224.9971 1171.519
130 Q3        695.2624 218.6641 1171.861
130 Q4        692.5282 213.1671 1171.889
131 Q1        690.0324 208.3818 1171.683
131 Q2        687.7544 204.2045 1171.304
131 Q3        685.6751 200.5485 1170.802
131 Q4        683.7772 197.3410 1170.213
132 Q1        682.0449 194.5202 1169.570
132 Q2        680.4637 192.0341 1168.893
132 Q3        679.0204 189.8381 1168.203
132 Q4        677.7031 187.8946 1167.512
133 Q1        676.5006 186.1711 1166.830
133 Q2        675.4031 184.6398 1166.166
133 Q3        674.4013 183.2770 1165.526
133 Q4        673.4869 182.0620 1164.912
134 Q1        672.6523 180.9771 1164.328
134 Q2        671.8905 180.0069 1163.774
134 Q3        671.1952 179.1379 1163.252
134 Q4        670.5605 178.3587 1162.762
135 Q1        669.9812 177.6590 1162.303
135 Q2        669.4524 177.0299 1161.875
135 Q3        668.9698 176.4637 1161.476
135 Q4        668.5292 175.9536 1161.105
136 Q1        668.1271 175.4935 1160.761
136 Q2        667.7601 175.0782 1160.442
136 Q3        667.4251 174.7029 1160.147
136 Q4        667.1193 174.3636 1159.875
```

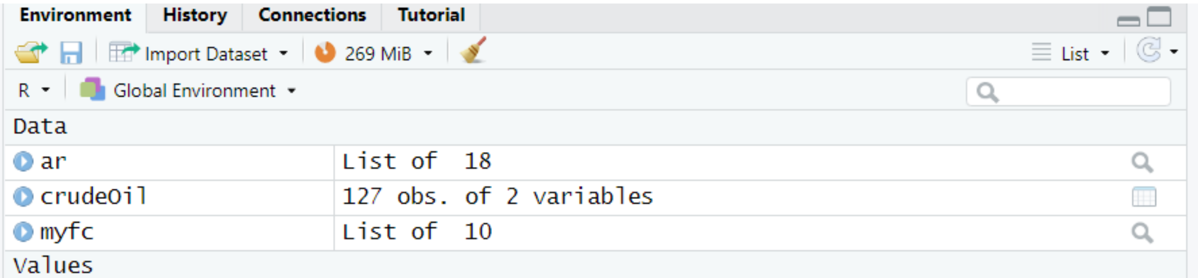
plot(myfc)



#Non Stationary Dataset:

Reading the csv file:

crudeOil=read.csv("Crudeoil_NS.csv")



#to check the class of the file

class(crudeOil)

```
> class(crudeOil)
[1] "data.frame"
```

Since this is a dataframe type, we have to change it into time series by using:

co=ts(crudeOil$POILBREUSDQ,start=1,frequency = 12)

```
> co=ts(crudeOil$POILBREUSDQ,start=1,frequency = 12)
> class(co)
[1] "ts"
```

```
> adf.test(co)

        Augmented Dickey-Fuller Test

data:  co
Dickey-Fuller = -1.9024, Lag order = 5, p-value = 0.6171
alternative hypothesis: stationary
```

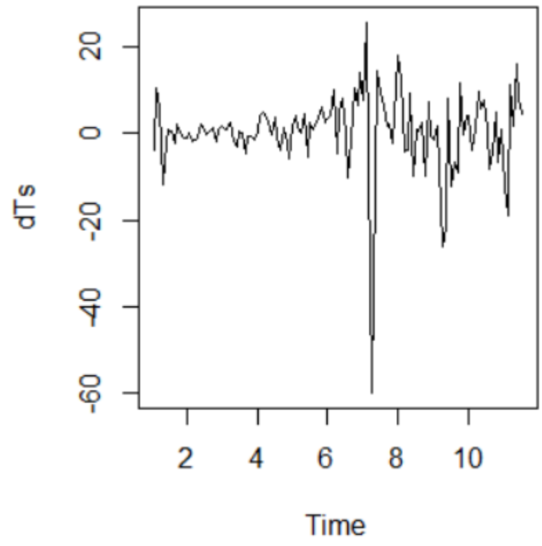
In general, a p-value of less than 5% means you can reject the null hypothesis that there is a unit root. You can also compare the calculated DFT statistic with a tabulated critical value. If the DFT statistic is more negative than the table value, reject the null hypothesis of a unit root.

As our p-value>0.5, we need to accept the Null Hypothesis and we need to do differencing in order to get stationary data out of this dataset.

For this,

dTs=diff(co)

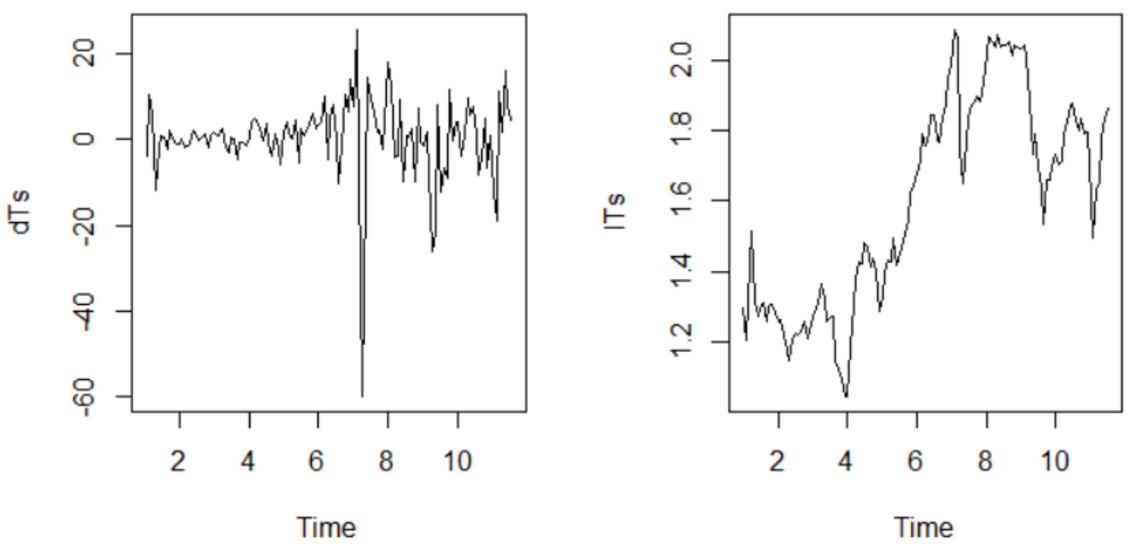
plot(dTs)



#Since, this dosent have a constant variance, we will again go for differencing,

lTs=log10(co)

plot(lTs)



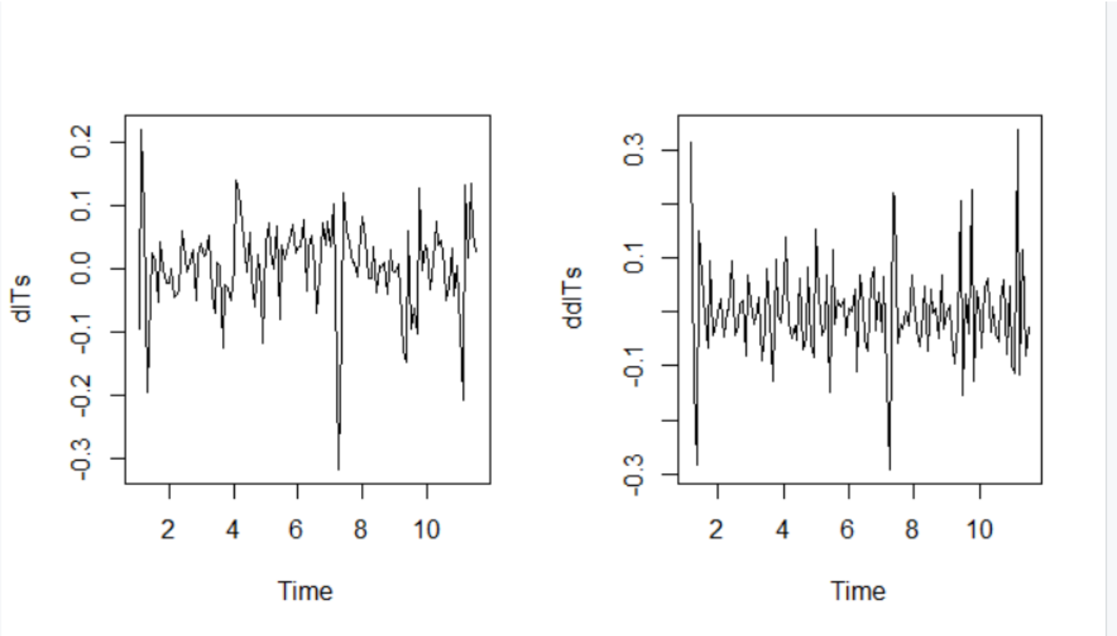
#This is of same nature, so again differencing,

dlTs=diff(lTs)

plot(dlTs)

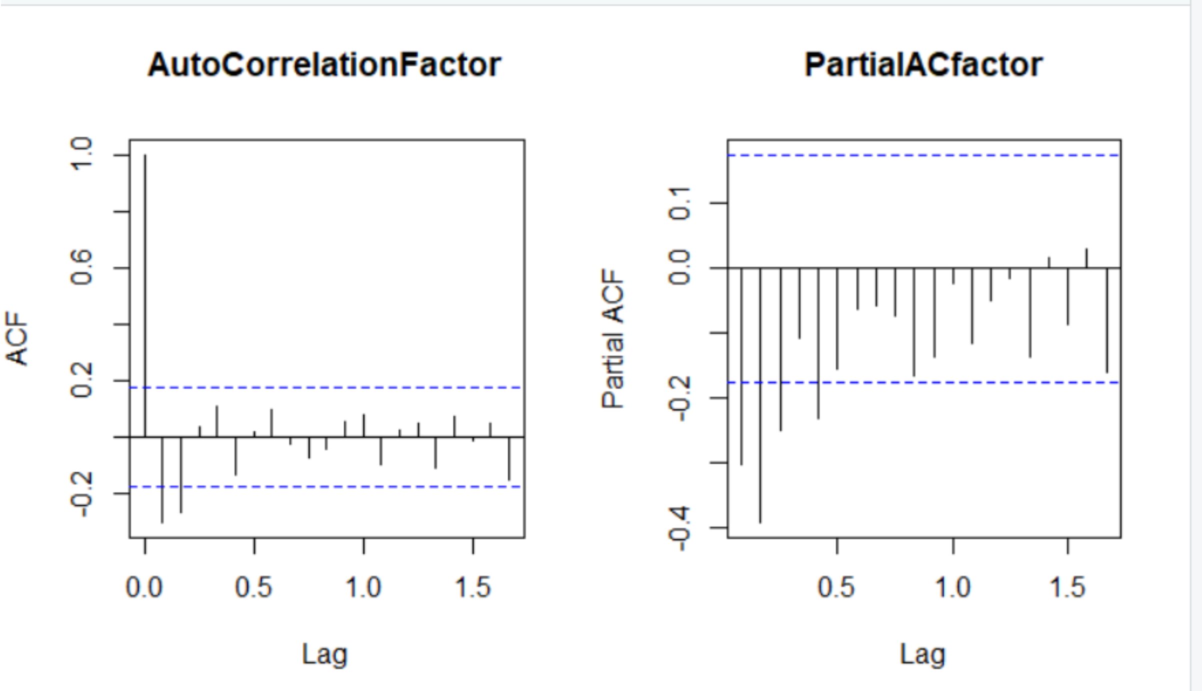
ddlTs=diff(dlTs) #incase,the variance dsnt seem almost constant, we go for another time differentiating the data

plot(ddlTs)



Now,

ACF and PACF:



Since our data is stationary now, we don’t have to go for differencing, we can directly proceed for ARIMA test.

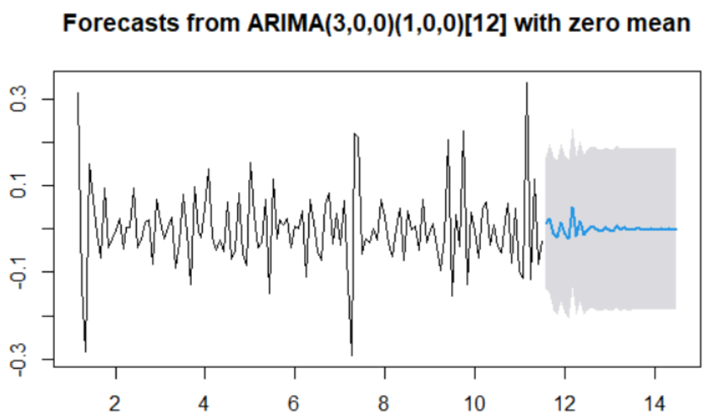
```
pacf(ddlTs,main="PartialACfactor")

library(forecast)

library(tseries)

ar=auto.arima(ddlTs,ic="aic",trace=TRUE)
```

```
> library(forecast)
> library(tseries)
> ar=auto.arima(ddlTs,ic="aic",trace=TRUE)

 ARIMA(2,0,2)(1,0,1)[12] with non-zero mean : Inf
 ARIMA(0,0,0)            with non-zero mean : -236.911
 ARIMA(1,0,0)(1,0,0)[12] with non-zero mean : -247.1034
 ARIMA(0,0,1)(0,0,1)[12] with non-zero mean : Inf
 ARIMA(0,0,0)            with zero mean     : -238.8971
 ARIMA(1,0,0)            with non-zero mean : -247.9639
 ARIMA(1,0,0)(0,0,1)[12] with non-zero mean : -246.8419
 ARIMA(1,0,0)(1,0,1)[12] with non-zero mean : -245.8547
 ARIMA(2,0,0)            with non-zero mean : -269.1623
 ARIMA(2,0,0)(1,0,0)[12] with non-zero mean : -268.9366
 ARIMA(2,0,0)(0,0,1)[12] with non-zero mean : -268.8181
 ARIMA(2,0,0)(1,0,1)[12] with non-zero mean : -266.9613
 ARIMA(3,0,0)            with non-zero mean : -275.2505
 ARIMA(3,0,0)(1,0,0)[12] with non-zero mean : -275.4293
 ARIMA(3,0,0)(2,0,0)[12] with non-zero mean : -273.4476
 ARIMA(3,0,0)(1,0,1)[12] with non-zero mean : -273.4367
 ARIMA(3,0,0)(0,0,1)[12] with non-zero mean : -275.3826
 ARIMA(3,0,0)(2,0,1)[12] with non-zero mean : Inf
 ARIMA(4,0,0)(1,0,0)[12] with non-zero mean : -273.9149
 ARIMA(3,0,1)(1,0,0)[12] with non-zero mean : Inf
 ARIMA(2,0,1)(1,0,0)[12] with non-zero mean : Inf
 ARIMA(4,0,1)(1,0,0)[12] with non-zero mean : Inf
 ARIMA(3,0,0)(1,0,0)[12] with zero mean     : -277.4172
 ARIMA(3,0,0)            with zero mean     : -277.233
 ARIMA(3,0,0)(2,0,0)[12] with zero mean     : -275.4363
 ARIMA(3,0,0)(1,0,1)[12] with zero mean     : -275.425
 ARIMA(3,0,0)(0,0,1)[12] with zero mean     : -277.3713
 ARIMA(3,0,0)(2,0,1)[12] with zero mean     : Inf
 ARIMA(2,0,0)(1,0,0)[12] with zero mean     : -270.9329
 ARIMA(4,0,0)(1,0,0)[12] with zero mean     : -275.897
 ARIMA(3,0,1)(1,0,0)[12] with zero mean     : Inf
 ARIMA(2,0,1)(1,0,0)[12] with zero mean     : Inf
 ARIMA(4,0,1)(1,0,0)[12] with zero mean     : Inf

 Best model: ARIMA(3,0,0)(1,0,0)[12] with zero mean
```

Forecasting :

myfc=forecast(ar,level=c(95),h=3*12)

plot(myfc)



Attributes of arima test:

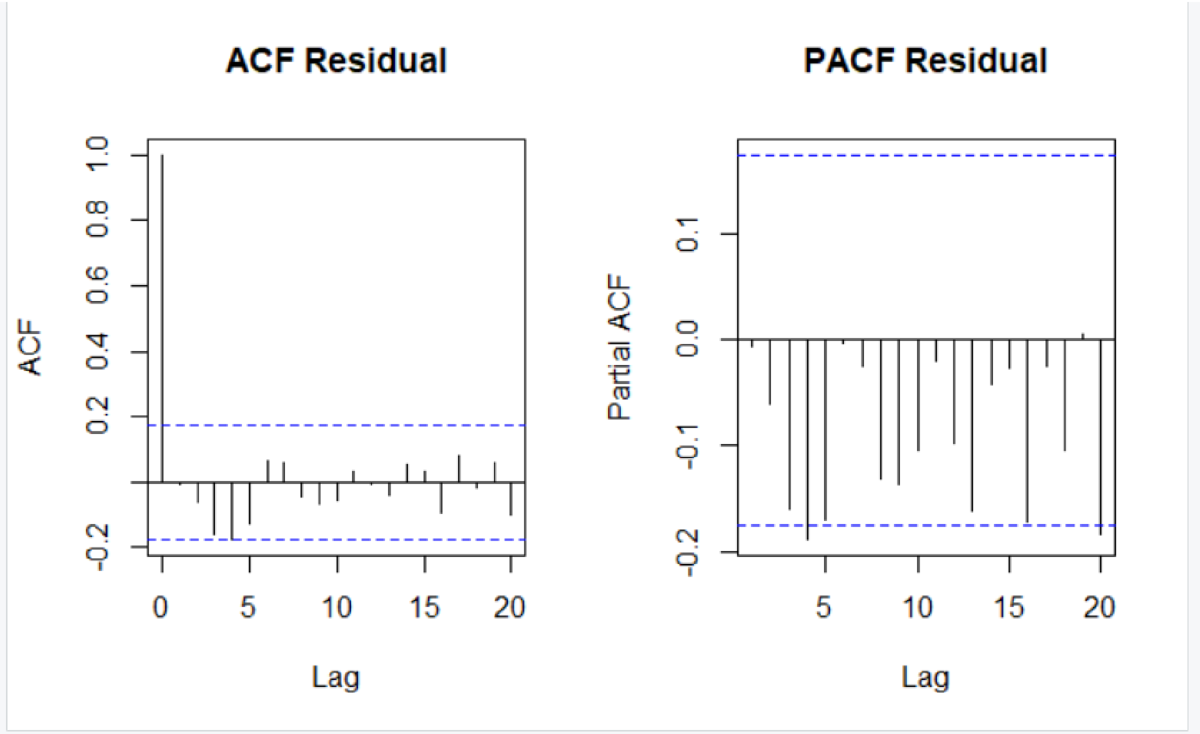
```
> attributes(ar)
$names
 [1] "coef"      "sigma2"    "var.coef"  "mask"      "loglik"    "aic"       "arma"
 [8] "residuals" "call"      "series"    "code"      "n.cond"    "nobs"      "model"
[15] "bic"       "aicc"      "x"         "fitted"

$class
[1] "forecast_ARIMA" "ARIMA"          "Arima"

>
```

#Plotting the ACF and PACF of residual:

par(mfrow=c(1,2))

acf(ts(ar$residuals),main='ACF Residual')

pacf(ts(ar$residuals),main='PACF Residual')



Result:

Hence, the Auto Arima test has been performed on both stationary and non stationary data.