**Time Series Analysis of different data.**

**Analysis done by: Sairam Chandavale**

The following Case Study explores the analysis of different types of time series including

**ARIMA, moving average method, SARIMA , Holt-Winters Exponential Smoothing, GARCH and Transfer Function Models.**

1] **ARIMA model** is fitted on daily NACH Debit Transactions and check the assumptions by residual analysis and forecasting.

2] Fitting a **SARIMA Model**, residual analysis on monthly rainfall from 2002 to 2022.

3] Fitting a **SARIMA Model** and **Holt Winters Exponential Smoothing** method on monthly closing values of IBM COMPANY. Decide which model is best among them.

4] **Transfer Function Model** is fitted on the yearly GDP Growth rate of India with

the exogenous variable as Inflation rate.

5**] GARCH model** on return data of HDFC from NSE.

Time Series Analysis on daily NACH Debit Transactions

**Introduction :**

National Automated Clearing House (NACH) is a system developed by the National Payments Corporation of India (NPCI) for banks. This system can be used to make bulk transactions towards distribution of subsidies, dividends, salaries, pension, etc. We take number of daily NACH Debit transactions for 291 days from January 1, 2023 to October 18, 2023. Number of payment transactions are in lakhs per day.

**1] Exploratory data analysis by using time- series plot.**

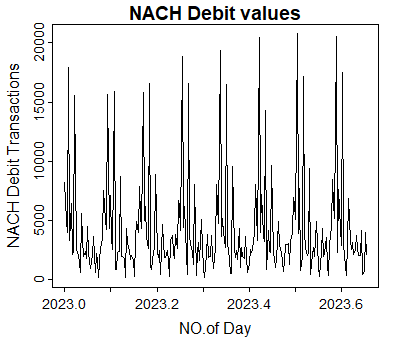
library(readxl)

d=read\_excel("C:\\Users\\admin\\Desktop\\time series.xlsx")

d=data.frame(d[,1:2])

View(d)

ts.plot(d[,2],main="NACH Debit values", xlab="NO.of Day",ylab="NACH Debit Transactions")



From the above time series plot , we observed that

1) The data does not have any trend .

3) No irregularities are seen in the plot.

**To check the stationarity of the data we perform ADF test.**

library(tseries)

adf.test(d[,2])

Augmented Dickey-Fuller Test

data: d[, 2]

Dickey-Fuller = -5.5025, Lag order = 6, p-value =0.01

alternative hypothesis: stationary

Warning message:

In adf.test(d[, 2]) : p-value smaller than printed p-value

**Conclusion :**

Null hypothesis has been rejected hence we conclude that the given time series is stationary.

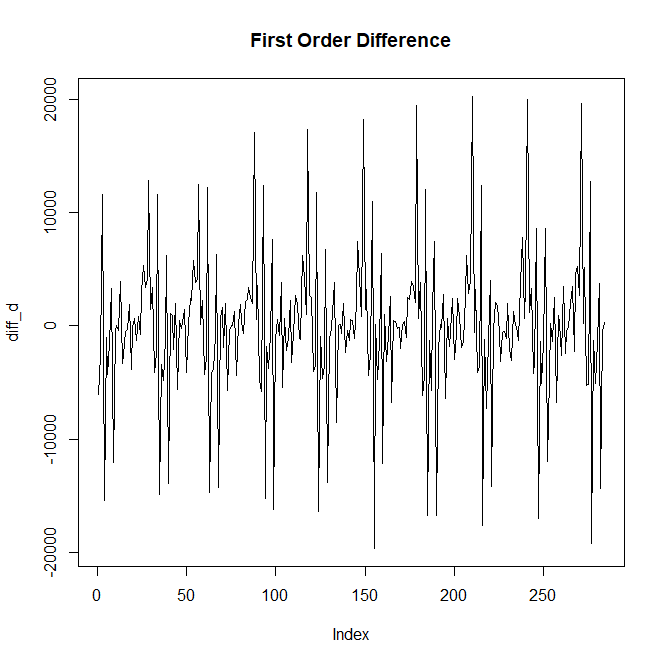
Due to observed seasonality ,to make the time series more stabilize and seasonality present Differencing of order 1 is done and the new time series is plotted.

Take the first order difference of the time series

diff\_d <- diff(d[, 2],lag=6)

# Plot the differenced time series

plot(diff\_d, type = "l", main = "First Order Difference")



To check the stationarity after differencing the time series

adf.test(diff\_d)

Augmented Dickey-Fuller Test

data: diff\_d

Dickey-Fuller = -7.6254, Lag order = 6, p-value =0.01

alternative hypothesis: stationary

Warning message:

In adf.test(diff\_d) : p-value smaller than printed p-value

**Conclusion :**

Null hypothesis has been rejected hence we conclude that the given time series is stationary.

mean(diff\_d)

[1] -88.79197

var(diff\_d)

[1] 40232874 #Here variance has been decreased after differencing

sd(diff\_d)

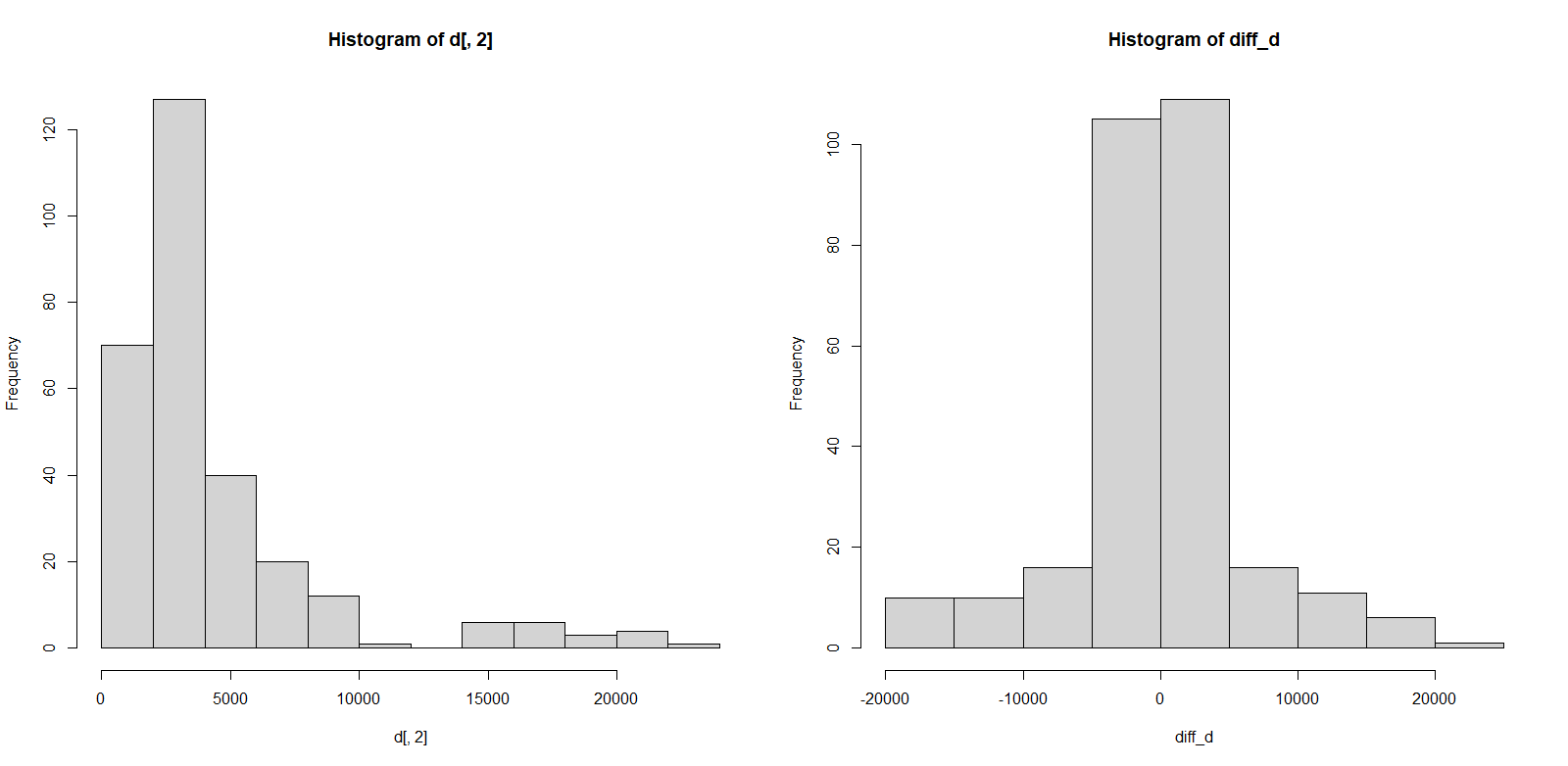
[1] 6342.939

###################

par(mfrow=c(1,2))

hist(d[,2])

hist(diff\_d)



**After comparing the histograms before and after differencing the time series data , we can observe that after differencing the data has approximately centered around mean zero .**

# Decompose the time series

#stl <- stl(d[, 2], s.window = "periodic")

# Plot the trend, seasonal, and residual components

#plot(stl$time, stl$trend, type = "l", col = "blue", main = "Trend Component")

#plot(stl$time, stl$seasonal, type = "l", col = "green", main = "Seasonal Component")

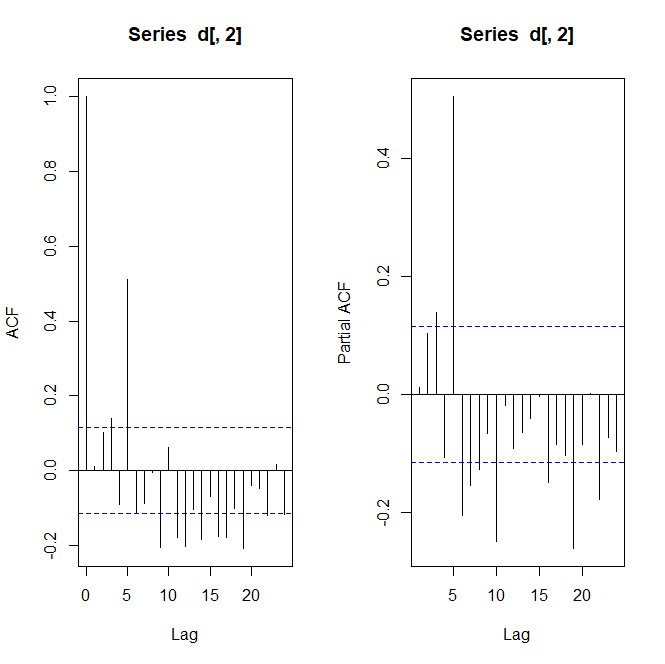
#plot(stl$time, stl$random, type = "l", col = "red", main = "Residual Component")

Now,

par(mfrow=c(1,2))

acf(d[,2])

pacf(d[,2])



**Conclusion:**

**The ACF and PACF plots should be considered together to define a process . From the above fig we observe that lags which are multiple of 6 show strong positive correlation.**

**To check the normality**shapiro.test(diff\_d)

Shapiro-Wilk normality test

data: diff\_d

W = 0.91486, p-value = 1.269e-11

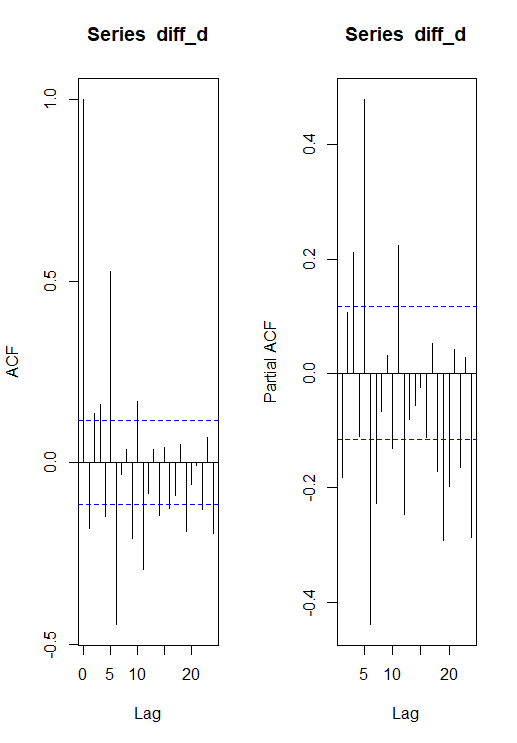
Here, p-value is less than los, hence we reject H0.

###################

par(mfrow=c(1,2))

acf(diff\_d)

pacf(diff\_d)



**Comclusion:**

**The ACF and PACF plots should be considered together to define a process . From the above figure we observed that , both the graphs show geometrical decreasing pattern hence mixed ARIMA model is considered for modelling.**

#############

par(mfrow=c(2,2))

ts.plot(d[,2],main="NACH Debit values", xlab="date",ylab=" NACH Debit ")

# 3 point MA

x\_movavg\_3=filter(d[,2],sides=2,rep(1,3)/3)

ts.plot(x\_movavg\_3,main="3 point Moving Average")

#5 point MA

x\_movavg\_5=filter(d[,2],sides=2,rep(1,5)/5)

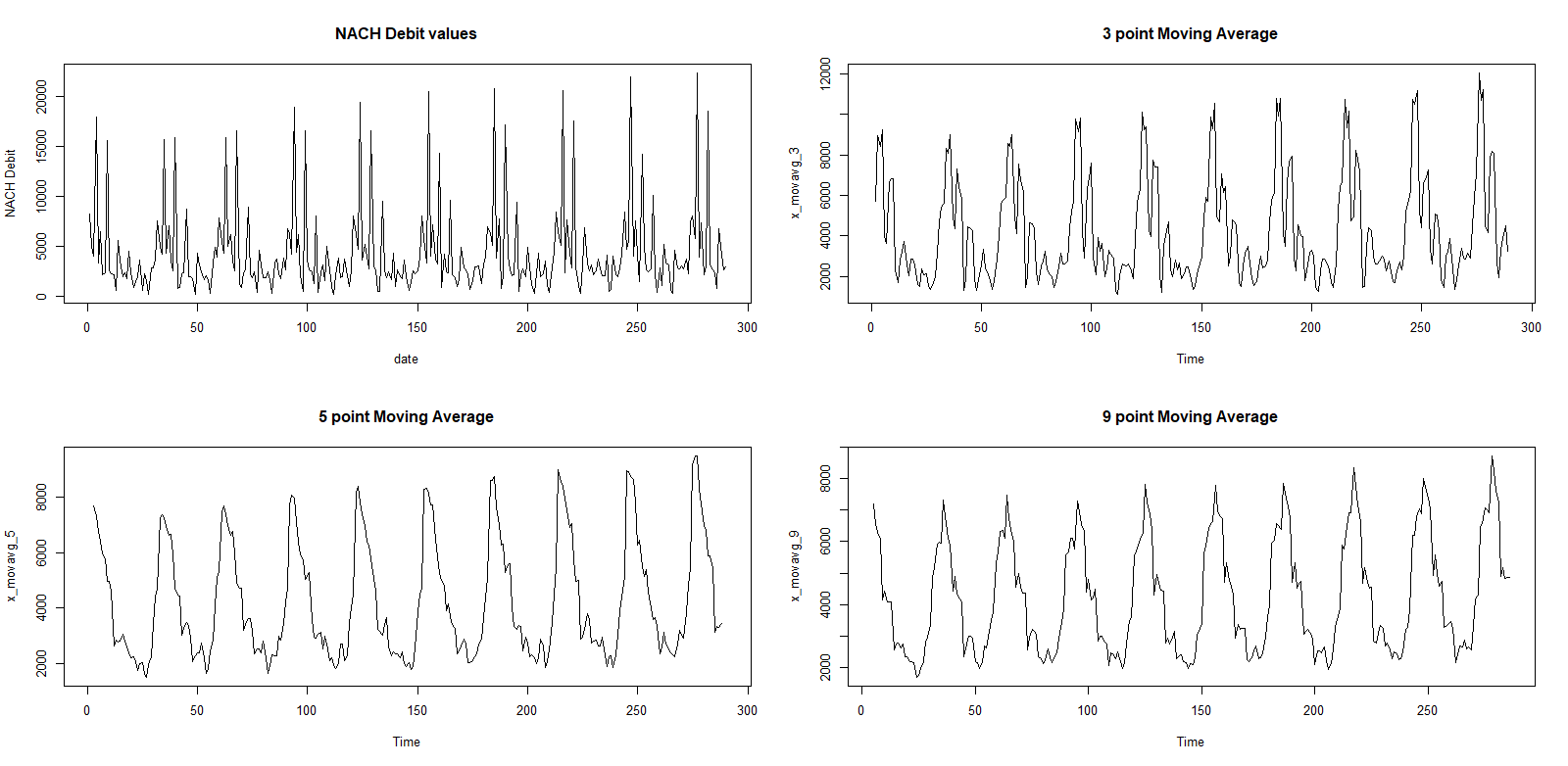
ts.plot(x\_movavg\_5,main="5 point Moving Average")

#9 point MA

x\_movavg\_9=filter(d[,2],sides=2,rep(1,9)/9)

x\_movavg\_9=na.omit(x\_movavg\_9)

ts.plot(x\_movavg\_9,main="9 point Moving Average")



adf.test(x\_movavg\_9)

Augmented Dickey-Fuller Test

data: x\_movavg\_9

Dickey-Fuller = -8.7225, Lag order = 6, p-value =

0.01

alternative hypothesis: stationary

Warning message:

In adf.test(x\_movavg\_9) : p-value smaller than printed p-value.

z=auto.arima(d[,2],seasonal=TRUE) #we fit ARIMA model with seasonality TRUE

z #it will choose the model which has min AIC and BIC.

Series: d[, 2]

Best model

ARIMA(5,0,3) with non-zero mean

Coefficients:

ar1 ar2 ar3 ma1 ma2 mean

-1.1152 -0.2265 0.3146 1.2861 0.4510 4251.6943

s.e. 0.0979 0.1161 0.0635 0.0926 0.0899 300.2008

sigma^2 = 14686656: log likelihood = -2801.86

AIC=5617.72 AICc=5618.12 BIC=5643.41

z1=auto.arima(d[,2],ic='aic',trace=TRUE)

Fitting models using approximations to speed things up...

ARIMA(2,0,2) with non-zero mean : 5657.396

ARIMA(0,0,0) with non-zero mean : 5672.198

ARIMA(1,0,0) with non-zero mean : 5674.301

ARIMA(0,0,1) with non-zero mean : 5674.163

ARIMA(0,0,0) with zero mean : 5872.059

ARIMA(1,0,2) with non-zero mean : 5674.559

ARIMA(2,0,1) with non-zero mean : 5670.167

ARIMA(3,0,2) with non-zero mean : 5620.07

ARIMA(3,0,1) with non-zero mean : 5629.898

ARIMA(4,0,2) with non-zero mean : 5629.524

ARIMA(3,0,3) with non-zero mean : 5621.428

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

ARIMA(5,0,3) with non-zero mean : 5570.84

Best model: ARIMA(5,0,3) with non-zero mean

> **#To forecast no of transactions for next 80 days :**

> m=arima(d[,2],order=c(5,0,3))

> m

Call:

arima(x = d[, 2], order = c(5, 0, 3))

Coefficients:

ar1 ar2 ar3 ar4 ar5 ma1

-0.1973 -0.1474 -0.0183 -0.0542 0.5420 0.3513

s.e. 0.1060 0.1002 0.0917 0.0517 0.0538 0.1252

ma2 ma3 intercept

0.2539 0.1854 4286.1893

s.e. 0.1310 0.1131 411.2132

sigma^2 estimated as 11945422: log likelihood = -2775.42, aic = 5570.84

>fcast=forecast(m,h=50)

>fcast

Point Forecast Lo 80 Hi 80 Lo 95 Hi 95

291 3106.257 -1323.0608 7535.575 -3667.8000 9880.314

292 5943.216 1461.6914 10424.740 -910.6841 12797.115

293 4072.248 -421.9496 8566.445 -2801.0339 10945.529

294 3269.683 -1260.9273 7800.293 -3659.2874 10198.653

295 3846.113 -703.4766 8395.702 -3111.8838 10804.110

296 3797.481 -1325.4595 8920.422 -4037.3804 11632.343

297 5375.705 252.3843 10499.027 -2459.7381 13211.149

298 4090.425 -1035.8990 9216.748 -3749.6108 11930.460

299 3646.059 -1491.1312 8783.249 -4210.5954 11502.714

300 4209.365 -942.7157 9361.445 -3670.0622 12088.792

301 4075.388 -1240.8589 9391.634 -4055.1099 12205.885

302 4951.900 -369.3790 10273.179 -3186.2940 13090.094

303 4115.901 -1211.4162 9443.219 -4031.5278 12263.331

304 3882.736 -1451.0383 9216.511 -4274.5681 12040.041

305 4348.491 -993.3413 9690.324 -3821.1366 12518.119

306 4186.168 -1207.6228 9579.959 -4062.9233 12435.260

307 4674.139 -724.1595 10072.438 -3581.8463 12930.125

308 4152.817 -1249.6768 9555.310 -4109.5842 12415.218

309 4035.109 -1371.8190 9442.037 -4234.0738 12304.292

310 4387.473 -1023.4241 9798.369 -3887.7798 12662.725

311 4230.431 -1197.4641 9658.326 -4070.8181 12531.680

312 4504.336 -926.1011 9934.774 -3800.8011 12809.474

313 4190.834 -1241.7845 9623.452 -4117.6389 12499.307

314 4132.300 -1303.1975 9567.797 -4180.5761 12445.176

315 4384.529 -1052.8945 9821.953 -3931.2927 12700.351

316 4249.178 -1193.8495 9692.206 -4075.2142 12573.571

317 4405.206 -1038.9809 9849.393 -3920.9592 12731.371

318 4223.021 -1222.1698 9668.213 -4104.6800 12550.723

319 4193.054 -1253.8454 9639.953 -4137.2596 12523.367

320 4367.001 -1080.8389 9814.841 -3964.7512 12698.753

321 4258.623 -1191.0593 9708.306 -4075.9469 12593.193

322 4349.345 -1100.7964 9799.487 -3985.9271 12684.617

323 4247.124 -1203.4559 9697.705 -4088.8188 12583.068

324 4230.236 -1221.2775 9681.750 -4107.1346 12567.607

325 4347.123 -1104.8530 9799.098 -3990.9546 12685.200

326 4264.768 -1187.8093 9717.346 -4074.2296 12603.767

>

plot(fcast)

checkresiduals(m)

Ljung-Box test

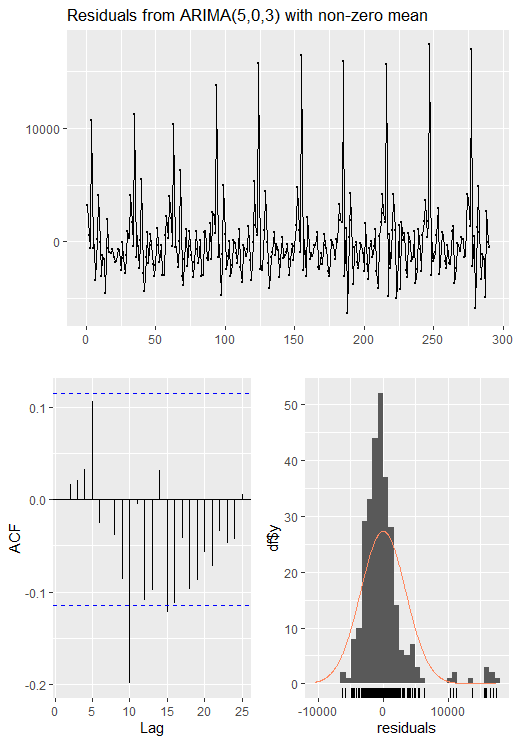
data: Residuals from ARIMA(5,0,3) with non-zero mean

Q\* = 18.796, df = 3, p-value = 0.0003013

Model df: 8. Total lags used: 11

**Conclusion**:1] there is no serial correlation as based on Ljung-Box test.(p-value<0.5)

2]residuals follow normal distribution.



**2] Time Series Analysis on monthly rainfall from 2002 to 2022.**

Introduction :  
we collect the data of monthly rainfall from 2002 to 2022 from <https://data.gov.in/> website.

We have 240 observations from 2002 to 2022. Rainfall measured in mm.

**1] Exploratory data analysis by using time- series plot.**

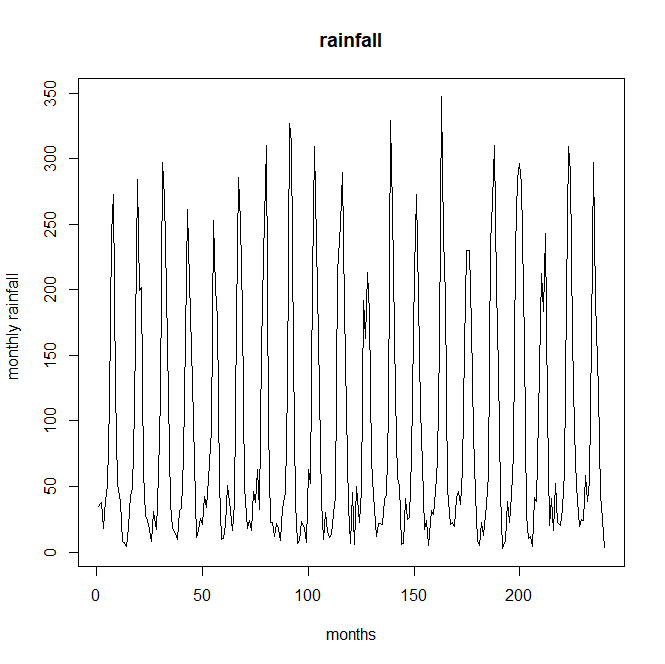
library(readxl)

d=read\_excel("C:\\Users\\admin\\Desktop\\rainfall.xlsx")

d=data.frame(d)

View(d)

ts.plot(d[,2],main="rainfall", xlab="months",ylab="monthly rainfall ")



From the above time series plot , we observed that

1) The data does not have any trend .

2) Data has seasonality present in it.

3) No irregularities are seen in the plot.

To check the stationarity of the data we perform ADF test.

library(tseries)

adf.test(d[,2])

Augmented Dickey-Fuller Test

data: d[, 2]

Dickey-Fuller = -13.822, Lag order = 6, p-value =0.01

alternative hypothesis: stationary

Warning message:

In adf.test(d[, 2]) : p-value smaller than printed p-value.

**Conclusion :**

Null hypothesis has been rejected hence we conclude that the given time series is stationary.

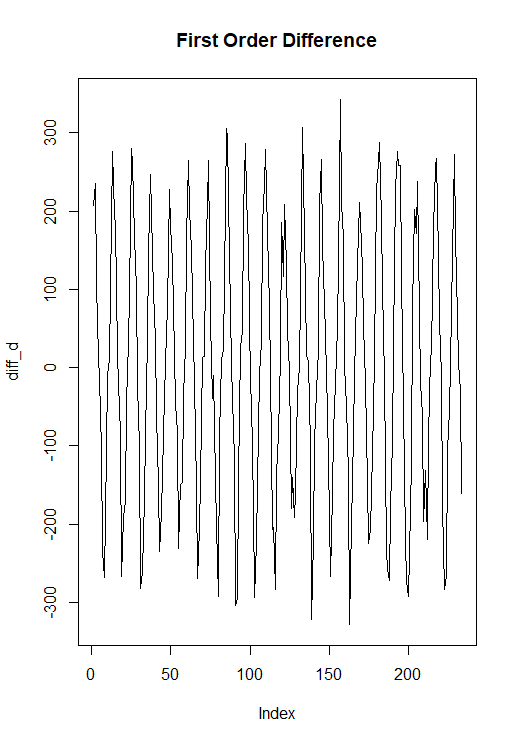
Due to observed seasonality ,to make the time series more stabilize and to eliminate trend and seasonality present Differencing of order 1 is done and the new time series is plotted.

Take the first order difference of the time series

diff\_d <- diff(d[, 2],lag=6)

# Plot the differenced time series

plot(diff\_d, type = "l", main = "First Order Difference")



To check the stationarity after differencing the time series

adf.test(diff\_d)

Augmented Dickey-Fuller Test

data: diff\_d

Dickey-Fuller = -11.657, Lag order = 6, p-value =0.01

alternative hypothesis: stationary

Warning message:

In adf.test(diff\_d) : p-value smaller than printed p-value

**Conclusion :**

**Null hypothesis has been rejected hence we conclude that the given time series is stationary.**

> mean(diff\_d)

[1] 1.667521

> var(diff\_d)

[1] 28416.52 #Here variance has been decreased after differencing

> sd(diff\_d)

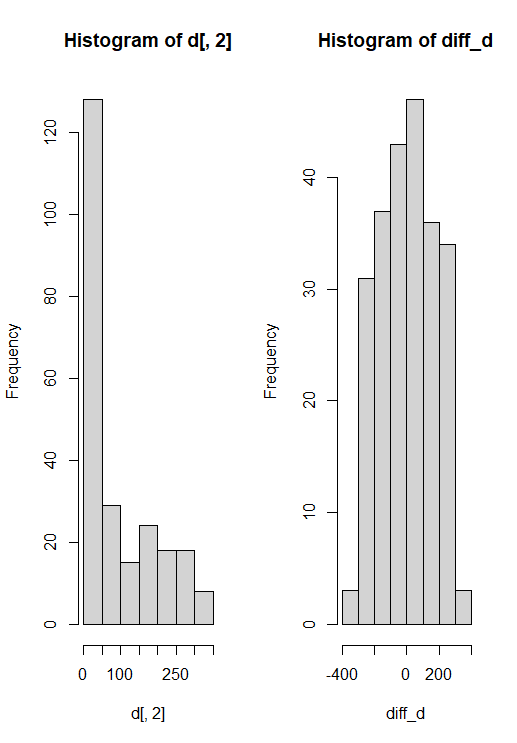
[1] 168.572

###################

par(mfrow=c(1,2))

hist(d[,2])

hist(diff\_d)



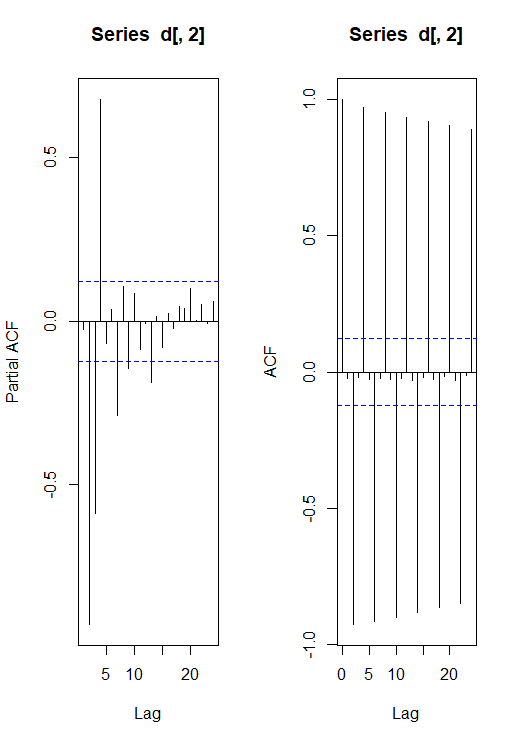
After comparing the histograms before and after differencing the time series data , we can observe that after differencing the data has approximately centered around mean zero .

To check the Collinearity

par(mfrow=c(1,2))

acf(d[,2])

pacf(d[,2])



The ACF and PACF plots should be considered together to define a process . From the above fig we observe that lags which are multiple of 5 show strong positive correlation.

> shapiro.test(diff\_d

Shapiro-Wilk normality test

data: diff\_d

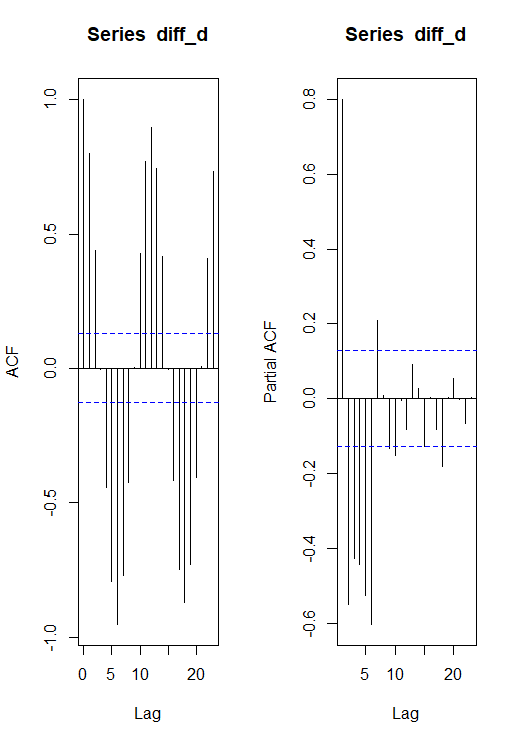
W = 0.9709, p-value = 9.839e-05

###################

par(mfrow=c(1,2))

acf(diff\_d)

pacf(diff\_d)



**Conclusion:**

**The ACF and PACF plots should be considered together to define a process . From the above figure we observed that , both the graphs show geometrical decreasing pattern hence mixed SARIMA model is considered for modelling.**

**Here, we consider seasonality is present in the model. Therefore we fit SARIMA model**

**,we take seasonality is true. We fit the model which has minimum AIC and BIC.**

library(astsa)

library(tseries)

library(fpp3)

library(forecast)

fit1=sarima(d,2,1,0,1,1,1,12)

output:

>fit1=sarima(d,2,1,0,1,1,1,12)

initial value 3.957064

iter 2 value 3.668401

iter 3 value 3.607178

iter 4 value 3.561301

iter 5 value 3.553501

iter 6 value 3.527395

iter 7 value 3.515419

iter 8 value 3.508994

iter 9 value 3.505019

iter 10 value 3.504226

iter 11 value 3.499883

iter 12 value 3.498903

iter 13 value 3.498752

iter 14 value 3.498617

iter 15 value 3.498614

iter 16 value 3.498611

iter 16 value 3.498611

iter 16 value 3.498611

final value 3.498611

converged

initial value 3.488147

iter 2 value 3.477986

iter 3 value 3.466116

iter 4 value 3.464272

iter 5 value 3.463012

iter 6 value 3.462898

iter 7 value 3.462893

iter 8 value 3.462893

iter 8 value 3.462892

iter 8 value 3.462892

final value 3.462892

converged

> fit1

$fit

Call:

arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D, Q), period = S),

include.mean = !no.constant, transform.pars = trans, fixed = fixed, optim.control = list(trace = trc,

REPORT = 1, reltol = tol))

Coefficients:

ar1 ar2 ma1 sar1 sma1

0.1444 0.0729 -1.0000 0.0334 -0.9447

s.e. 0.0665 0.0666 0.0232 0.0788 0.0981

sigma^2 estimated as 879.9: log likelihood = -1108.18, aic = 2228.35

$degrees\_of\_freedom

[1] 222

$ttable

Estimate SE t.value p.value

ar1 0.1444 0.0665 2.1697 0.0311

ar2 0.0729 0.0666 1.0958 0.2743

ma1 -1.0000 0.0232 -43.0446 0.0000

sar1 0.0334 0.0788 0.4239 0.6721

sma1 -0.9447 0.0981 -9.6326 0.0000

$AIC

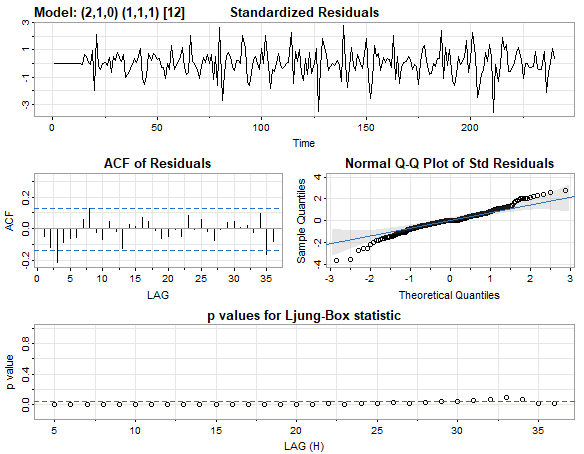
[1] 9.816525

$AICc

[1] 9.817721

$BIC

[1] 9.907053



> forecast=sarima.for(d,12,2,1,0,1,1,1,12)

>forecast #next predicted values of time series.

$pred

Time Series:

Start = 241

End = 252

Frequency = 1

[1] 23.93833 29.49204 36.84298 44.00086 65.53913 178.25595 279.39957

[8] 256.44381 169.55080 77.36384 37.25375 20.99041

$se

Time Series:

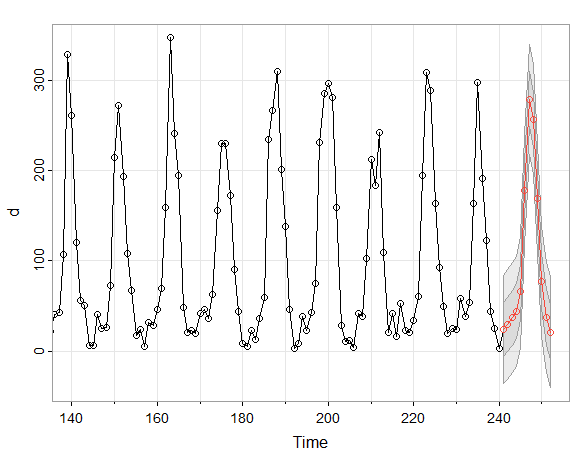
Start = 241

End = 252

Frequency = 1

[1] 30.02499 30.38461 30.55065 30.56917 30.57510 30.57665 30.57721 30.57744

[9] 30.57767 30.57807 30.57942 30.58133



**CONCLUSION:**

1] There is NO serial auto-correlation between the residuals. This is verified by conducted

the Ljung-Box Test.(i.e. p-value <0.5 for all lag)

2] The residuals of SARIMA MODEL appears to follow Normal Distribution in the histogram.

3] this model has min AIC and BIC

4] from ACF of residuals, we observe that, residuals are uncorrelated.

**3]** **Time Series Analysis on monthly closing values of IBM COMPANY from 2000 to 2023.**

The data ,we used in this analysis is ‘**monthly closing values of stocks of IBM**’ which is secondary data available on the https://www.alphavantage.co/documentation/

It has monthly values from jan-2000 to sep 2023. So in this statistical analysis, we tried to figure out, if there could be any trend or seasonality striking out from the data..

Firstly, we did some exploratory data analysis with the help of time series graph.

Then, we used some time series techniquessuch as differencing, de-trending and de-seasonalizing. Also, for forecasting purpose we used Exponential smoothing technique and SARIMA model and decide which model is best based on AIC and BIC.

Installing the required libraries for the time series analysis.

#install.packages("fpp3")

#install.packages("forecast")

library(astsa)

library(tseries)

library(fpp3)

library(forecast)

importing the data:

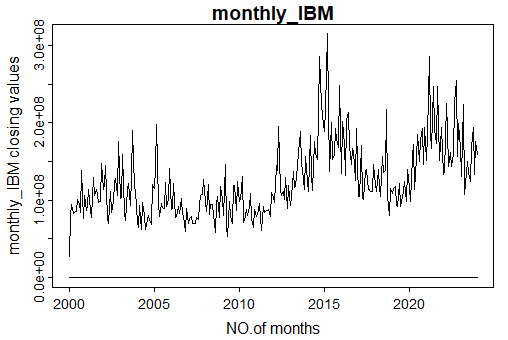
data1=read.csv("C:\\Users\\admin\\Downloads\\monthly\_IBM.csv")

data1=data.frame(data1[,6])

#View(d)

data1=ts((data1), start=2000,freq=12)

ts.plot(data1,main="monthly\_IBM", xlab="NO.of months",ylab="monthly\_IBM closing values", start=2000,freq=12)



Plotting the components of time series and checking whether trend and seasonality is Present:

decompose(data1)

plot(decompose(data1)) #This implies there is an additive model.

>$figure

[1] -12994941 -9791951 34584538 -14383950 -19654443 6251427 -7045869 -8732078

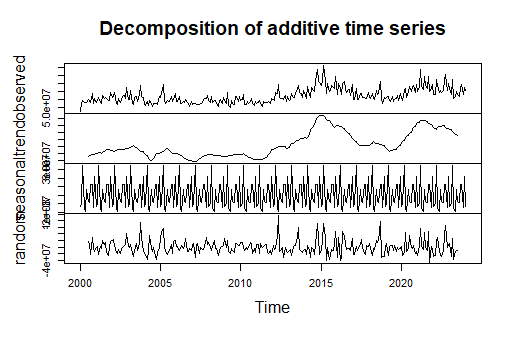
[9] 12649222 11860277 -14586045 21843813

$type

[1] "additive"

attr(,"class")

[1] "decomposed.ts"



Conclusion: # We can see that trend is present in the above data. Also seasonal effect is present.

Now to check stationarity (i.e. presence of trend)

adf.test(data1)

Augmented Dickey-Fuller Test

data: data1

Dickey-Fuller = -3.2144, Lag order = 6, p-value = 0.08585

alternative hypothesis: stationary

>

Conclusion: p-value is not less than los , this implies data is non- stationary.

Here , we use first order differencing to make the data stationary.

adf.test(diff(data1,3))

Augmented Dickey-Fuller Test

data: diff(data1, 3)

Dickey-Fuller = -7.2786, Lag order = 6, p-value = 0.01

alternative hypothesis: stationary

Warning message:

In adf.test(diff(data1, 3)) : p-value smaller than printed p-value

Conclusion: p-value is less than los , this implies data is stationary . After first order differencing ,data becomes stationary.

Now we get the stationary data. Now for seasonality we use Forecasting using Holt Winters Exponential Smoothing method. #Double Exp smoothing

data.hw=hw(data1,damped=T,seasonal="additive",h=12)

summary(data.hw)

autoplot(data1)+autolayer(data.hw,PI=T)

output:

>Forecast method: Damped Holt-Winters' additive method

Model Information:

Damped Holt-Winters' additive method

Call:

hw(y = data1, h = 12, seasonal = "additive", damped = T)

Smoothing parameters:

alpha = 0.4108

beta = 2e-04

gamma = 2e-04

phi = 0.952

Initial states:

l = 83328689.0375

b = 1636563.7625

s = 21843812 -14586046 11860277 12649221 -8732076 -7045871

6251427 -19654443 -14383950 34584539 -9791950 -12994941

sigma: 26280025

AIC AICc BIC

1153.81 1153.35 1159.81

Error measures:

ME RMSE MAE MPE MAPE MASE ACF1

Training set 3837.5 254953 193502 -2.584528 16.28346 0.6029004 0.07618272

Forecasts:

Point Forecast Lo 80 Hi 80 Lo 95 Hi 95

Feb 2024 151956230 118277023 185635437 100448328 203464132

Mar 2024 196325255 159913018 232737492 140637544 252012966

Apr 2024 147364750 108408909 186320592 87786933 206942568

May 2024 142092780 100747867 183437693 78861192 205324367

Jun 2024 167991920 124387136 211596704 101304158 234679682

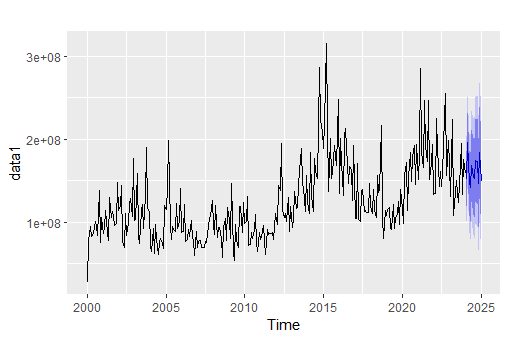
Jul 2024 154697171 108942637 200451706 84721649 224672693

Aug 2024 153006965 105198005 200815924 79889471 226124458

Sep 2024 174393953 124614148 224173758 98262311 250525595

Oct 2024 173608492 121931909 225285075 94575978 252641005

……………



Conclusion: Holt Winters Exponential Smoothing give predictions having high AIC and BIC ,

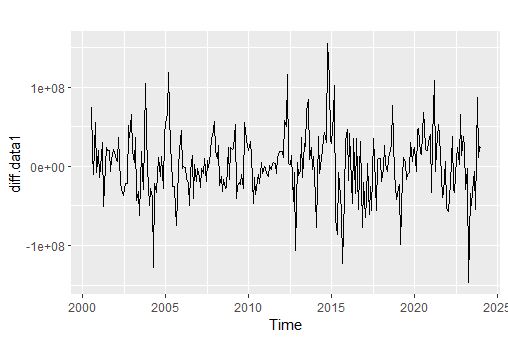
Removing the seasonal effect using differencing. By differencing already we make the data stationaty. Now we remove seasonal effect.

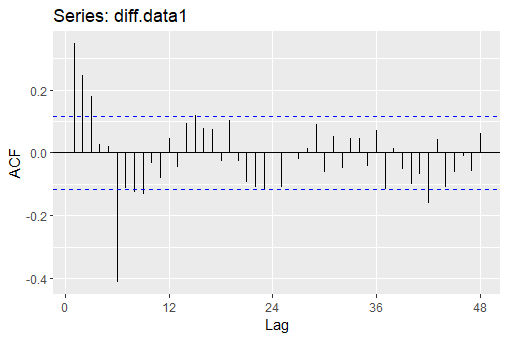
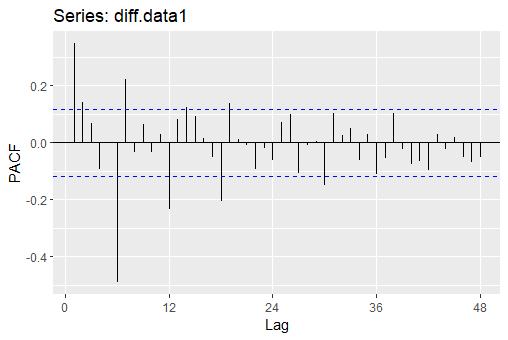
diff.data1=diff(data1,lag=6)

autoplot(diff.data1)

ggAcf(diff.data1,lag=48)

ggPacf(diff.data1,lag=48)





**Conclusion:**

As the seasonal effect is present in the dataset and trend is present and after seeing the ACF and

PACF of the given data, we can see that SARIMA model is best for the given data

(due to seasonal component) as it gives less AIC and BIC than other models.

fit1=sarima(data1,2,1,1,1,1,1,12)

fit1

> fit1

$fit

Call:

arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D, Q), period = S),

include.mean = !no.constant, transform.pars = trans, fixed = fixed, optim.control = list(trace = trc,

REPORT = 1, reltol = tol))

Coefficients:

ar1 ar2 ma1 sar1 sma1

0.1721 -0.0686 -0.6738 -0.0303 -0.9671

s.e. 0.1793 0.1142 0.1753 0.0684 0.1081

sigma^2 estimated as 6.814e+14: log likelihood = -5120.71, aic = 10253.42

$degrees\_of\_freedom

[1] 271

$ttable

Estimate SE t.value p.value

ar1 0.1721 0.1793 0.9600 0.3379

ar2 -0.0686 0.1142 -0.6009 0.5484

ma1 -0.6738 0.1753 -3.8429 0.0002

sar1 -0.0303 0.0684 -0.4422 0.6587

sma1 -0.9671 0.1081 -8.9436 0.0000

$AIC

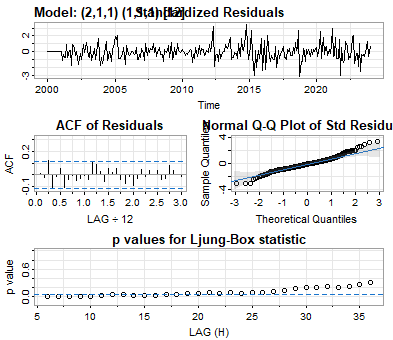
[1] 37.15008

$AICc

[1] 37.15088

$BIC

[1] 37.22878



**Conclusion :**

**all assumptions are hold . 1]residuals are uncorrelated .2] normality assumptions is hold . This SARIMA model is better as compared to Holt Winters Exponential Smoothing as SARIMA has low value of AIC and BIC. SARIMA model performs well on this data.**

Now, Forecasting using SARIMA(2,1,1,1,1,1,12) model.

>forecast=sarima.for(data1,12,2,1,1,1,1,1,12)

> forecast

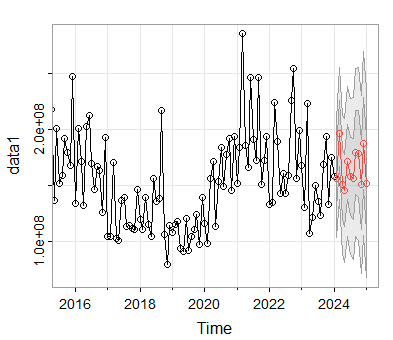
$pred

Jan Feb Mar Apr May Jun Jul Aug

2024 156891898 198798578 152662989 147346716 173035255 160414002 158443854

Sep Oct Nov Dec

2024 181517210 181355653 153992629 190775357



4] **Transfer function model on GDP and Inflation rates of india**

We take data of GDP growth rate and inflation rates of india from 1961 to 2022 from

<https://www.macrotrends.net/countries/IND/india/gdp-growth-rate> .

Transfer Function Model is fitted on the yearly GDP Growth rate of India with the exogenous

variable as Inflation rate. We consider the exogenous variable is Yearly inflation rate while GDP

growth rate is the response for that.

Importing the data:

#install.packages("tfarima")

library(tfarima)

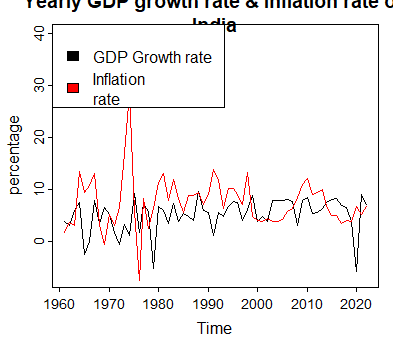
data7=read.csv("C:\\Users\\admin\\Downloads\\india-gdp-growth-rate.csv")

plot(gdp , main="Yearly GDP growth rate & Inflation rate of India",ylab="percentage",ylim=c(7,40))

lines(inflation,col="red")

legend("topleft",legend=c("GDP Growth rate", "Inflation rate"),fill=c("black","red"))

**Graph of GDP growth rate and inflation rates :**



we observe that whenever GDP growth rate decreases, the inflation rate increases.

**Model Fitting–**

The data is split into train data and validation data. The data till the year 2010 is considered

as the train set, and the data after 2010 upto 2022 is considered in the validation set.

The Transfer Function Arima Model is fitted on GDP growth rate, with inflation rate as the

response.

**Code:**

gdp1=data7$GDP[1:50]

gdp1=ts(gdp1,start=1961,freq=1)

inflation1=data7$Inflation[1:50]

inflation1=ts(inflation1,start=1961,freq=1)

gdp=data7$GDP

gdp=ts(gdp,start=1961,freq=1)

inflation=data7$Inflation

inflation=ts(inflation,start=1961,freq=1)

#Transfer Function Models

M7 = auto.arima(gdp1,xreg=inflation1)

summary(M7)

**output:**

>summary(M7)

Series: gdp1

Regression with ARIMA(0,1,1) errors

Coefficients:

ma1 xreg

-0.8977 -0.0044

s.e. 0.0520 0.0790

sigma^2 = 9.514: log likelihood = -124.52

AIC=255.04 AICc=255.57 BIC=260.71

Training set error measures:

ME RMSE MAE MPE MAPE MASE

Training set 0.5297693 2.990442 2.338511 141.2841 185.7994 0.7277008

ACF1

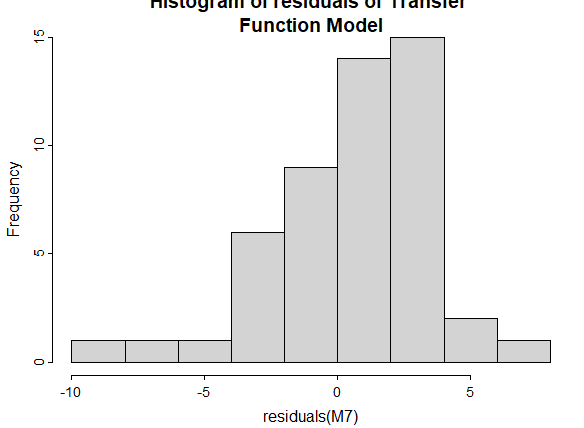
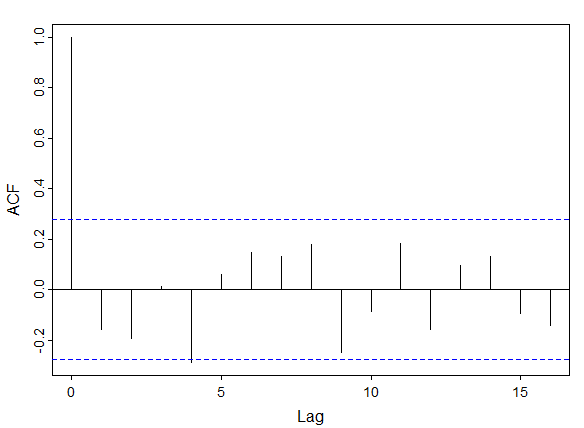
Training set -0.1582956

**Residual Analysis**

The residuals are analysed of the fitted Model to find the Model adequacy. We will first use

exploratory data analysis to check for Normality and serial uncorrelatedness of residuals. We

will verify these assumptions by using Shapiro test and Ljung-Box test.



**Assumptions checking:**

>shapiro.test(residuals(M7))

Shapiro-Wilk normality test

data: residuals(M7)

W = 0.94352, p-value = 0.0186

> Box.test(residuals(M7),type="Ljung-Box")

Box-Ljung test

data: residuals(M7)

X-squared = 1.3296, df = 1, p-value = 0.2489

**Conclusion:**

1) The Ljung-Box test suggests that the residuals are serially uncorrelated.

2) The Shapiro-Wilik test suggests that at 1% LOS, we can accept the hypothesis that the

residuals are normal.

**Model Predictions –**

The Transfer Function Model has been fitted, and even though the residual analysis is not

Very well. We will generate predictions for the test data. We obtain the predictions, and after

plotting them on the graph, we note the following things:

**code:**

M7.pred=forecast(M7,h=12,xreg = inflation)$mean[1:12]

M7.pred=ts(M7.pred,start=2011,freq=1)

plot(gdp,main="Yearly GDP growth rate & Inflation rate of

India",ylab="percentage",ylim=c(-7,40))

lines(fitted(M7),col="red",lwd=2)

lines(M7.pred,col="blue",lwd=2)

legend("topleft",legend=c("Predictions on Train

Data","Predictions on validation Data", "Actual

data"),fill=c("red","blue","black"))

RMSE.tfm=sum((gdp[51:62]-M7.pred)^2)

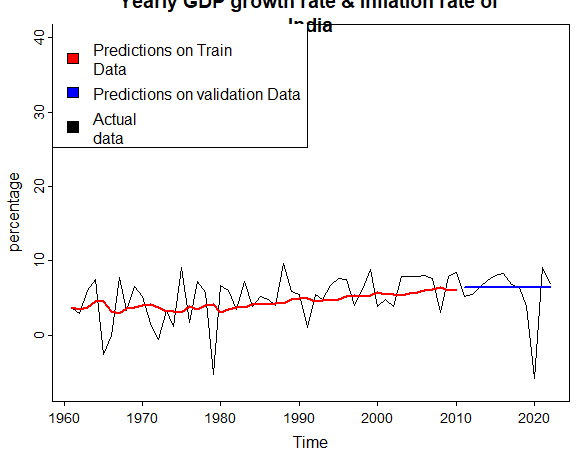
> RMSE.tfm=sum((gdp[51:62]-M7.pred)^2)

> RMSE.tfm

[1] 174.2585

#we get the RMSE value is 174.2585.

**Output:**



**Conclusions :**

1) We observe that ARIMA(0,1,1) is used by transfer function with Y=GDP and X=Inflation rate.

2) There is a relationship between the GDP growth rate and Inflation rate.

3) We fit the model and perform residual analysis.

**5] GARCH model on return data of HDFC from NSE**.

This data is taken as the returns of the NSE. The data is taken from the

[https://finance.yahoo.com/quote/HDFC/history?p=HDFC](https://finance.yahoo.com/quote/HDFC/history?p=HDFCB)

The data is taken from Jan 2010 to Dec 2023.**Exploratory Data Analysis –**

The exploratory data analysis enables us to find the presence of heteroskedasticity in the

data. The returns of a stock have drift parameter 0, so there is no need to fit an ARMA Model

prior to performing the ARCH-GARCH Model.

We observe the following from the graphs –

1) The ACF and PACF plots of squared residuals show the presence of GARCH structure.

2) We verify this by using Ljung-Box test, and get the p-value 0.2084

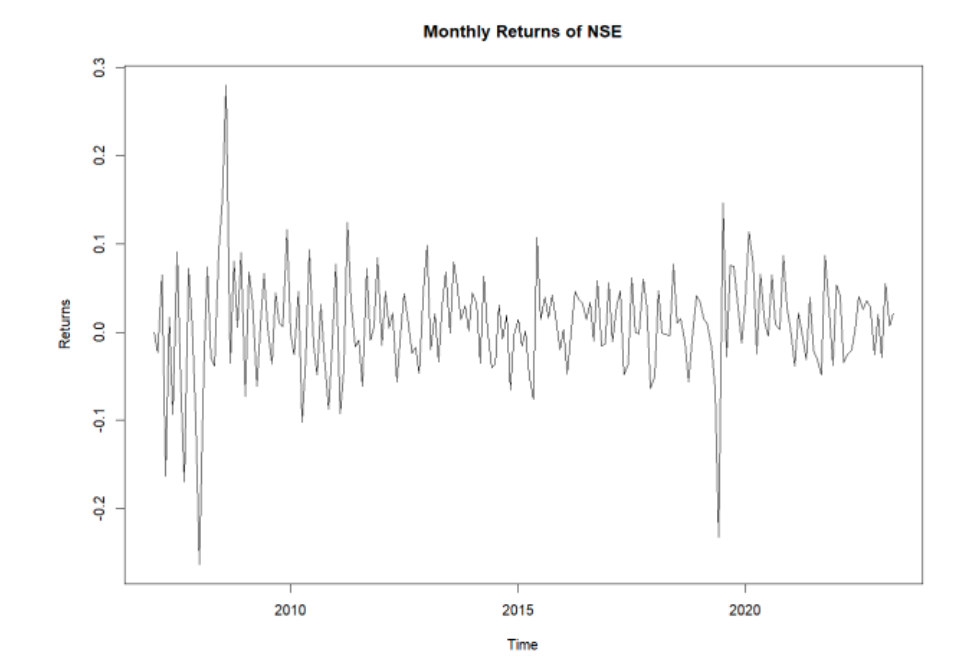
3) The Monthly returns of HDFC has drift parameter 0, so there is no need to fit an ARMA

model before conducting the analysis.

data= read.csv("C:\\Users\\admin\\Downloads\\^NSEI.csv")

data=data$returns

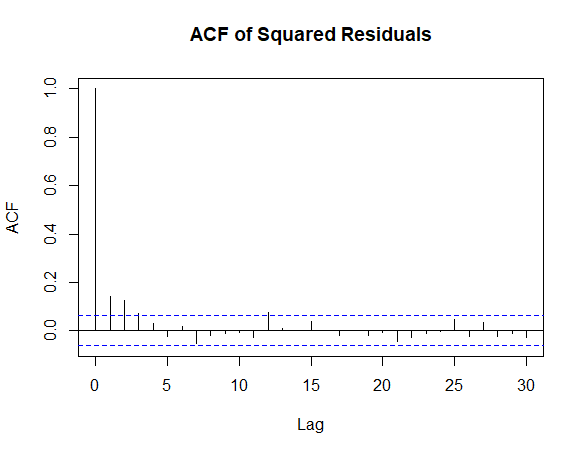
data <- ts(data)

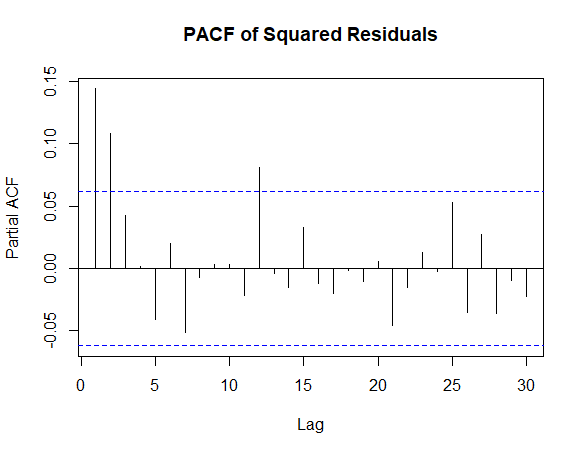
plot(data, type = 'l', ylab = 'Returns', main = 'Monthly Returns of NSE' )

acf\_result <- acf(squared\_residuals, main = 'ACF of Squared Residuals')

pacf\_result <- pacf(squared\_residuals, main = 'PACF of Squared Residuals')

output:





**Model Fitting –**

The GARCH(1,1) Model is fitted on this data, and obtain the coefficients of the model. We

also obtain the error variance table, which provides the significance of the coefficients.

We choose GARCH(1,1) model since it is the model with the smallest AIC and BIC.

>print(garch\_model)

\*---------------------------------\*

\* GARCH Model Fit \*

\*---------------------------------\*

Conditional Variance Dynamics

-----------------------------------

GARCH Model : sGARCH(1,1)

Mean Model : ARFIMA(0,0,0)

Distribution : norm

Optimal Parameters

------------------------------------

Estimate Std. Error t value Pr(>|t|)

omega 0.001046 0.001323 7.9065e-01 0.42915

alpha1 0.000000 0.001342 3.6000e-05 0.99997

beta1 0.999000 0.000040 2.5059e+04 0.00000

Robust Standard Errors:

Estimate Std. Error t value Pr(>|t|)

omega 0.001046 0.001267 8.2549e-01 0.40910

alpha1 0.000000 0.001249 3.9000e-05 0.99997

beta1 0.999000 0.000042 2.3555e+04 0.00000

LogLikelihood : -1390.893

Information Criteria

------------------------------------

Akaike 2.7878

Bayes 2.8025

Shibata 2.7878

Hannan-Quinn 2.7934

Weighted Ljung-Box Test on Standardized Residuals

------------------------------------

statistic p-value

Lag[1] 1.582 0.2084

Lag[2\*(p+q)+(p+q)-1][2] 1.631 0.3320

Lag[4\*(p+q)+(p+q)-1][5] 2.453 0.5160

d.o.f=0

H0 : No serial correlation

Weighted Ljung-Box Test on Standardized Squared Residuals

------------------------------------

statistic p-value

Lag[1] 20.63 5.566e-06

Lag[2\*(p+q)+(p+q)-1][5] 36.62 4.246e-10

Lag[4\*(p+q)+(p+q)-1][9] 40.44 5.444e-10

d.o.f=2

Weighted ARCH LM Tests

------------------------------------

Statistic Shape Scale P-Value

ARCH Lag[3] 5.070 0.500 2.000 0.02434

ARCH Lag[5] 5.967 1.440 1.667 0.06115

ARCH Lag[7] 7.377 2.315 1.543 0.07196

Nyblom stability test

------------------------------------

Joint Statistic: 1.996

Individual Statistics:

omega 0.1301

alpha1 0.1212

beta1 0.1273

Asymptotic Critical Values (10% 5% 1%)

Joint Statistic: 0.846 1.01 1.35

Individual Statistic: 0.35 0.47 0.75

Sign Bias Test

------------------------------------

t-value prob sig

Sign Bias 0.0952 9.242e-01

Negative Sign Bias 4.0549 5.407e-05 \*\*\*

Positive Sign Bias 3.1968 1.433e-03 \*\*\*

Joint Effect 27.2890 5.121e-06 \*\*\*

Adjusted Pearson Goodness-of-Fit Test:

------------------------------------

group statistic p-value(g-1)

1 20 18.88 0.4646

2 30 32.24 0.3094

3 40 41.20 0.3746

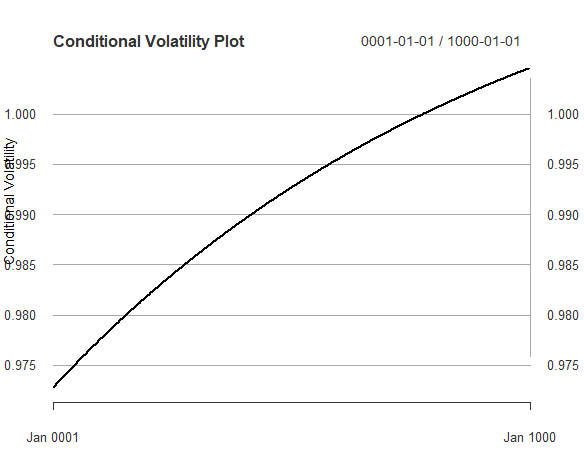
4 50 51.70 0.3688

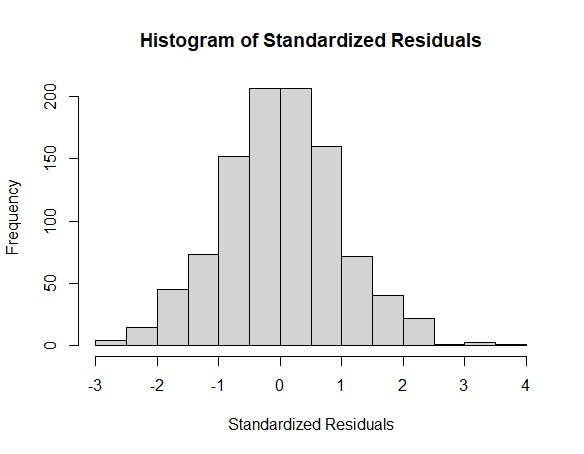
Residual Analysis –

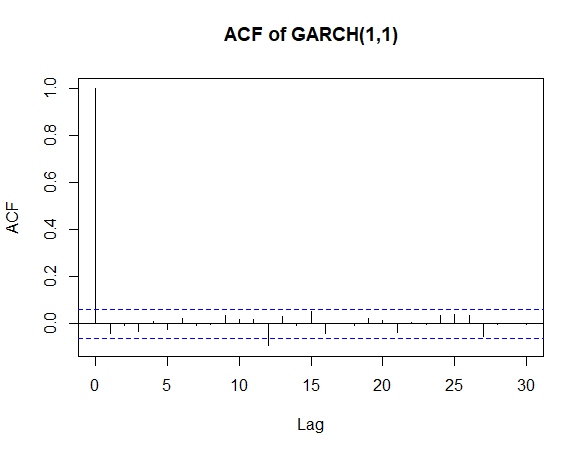
The residuals generated by the ARCH Model are as follows

>plot(conditional\_volatility, type = 'l', ylab = 'Conditional Volatility', main = 'Conditional Volatility Plot')

> hist(standardized\_residuals, main = 'Histogram of Standardized Residuals', xlab = 'Standardized Residuals')







From the ACF and Histogram of ACFs, we observe that

1) The residuals of GARCH(1,1) appear to follow Normal Distribution in the histogram.

We verify this by conducting Shapiro-Wilik normality test.

2)Serial Correlation: The weighted Ljung-Box tests on both standardized residuals and squared residuals suggest no serial correlation at the specified lags.

3)ARCH Effect: The weighted ARCH LM tests indicate significance for the ARCH effect at lag 3.

4)Nyblom Stability Test: The Nyblom stability test suggests stability in the parameters.

5)Sign Bias Test: The test reveals a significant negative sign bias and a significant positive sign bias.

6)Goodness-of-Fit: The adjusted Pearson goodness-of-fit test provides p-values for assessing the fit of the model at different lag lengths.