#moving average(MA) model does not uses the past forecasts to predict the future values whereas it uses the errors from the past forecasts.

#While, the autoregressive model(AR) uses the past forecasts to predict future values.

# Taken Sample Datasets.

ruler=read.csv("rulers.csv")

attach(ruler)

age=ts(Age)

plot(age,type = "b")

#ADF Test- null hypothesis is that that a unit root is present in a time series sample.

# A unit root (also called a unit root process or a difference stationary process) is a stochastic trend

# in a time series, sometimes called a "random walk with drift"'

library(tseries)

adf.test(age)

# The Kwiatkowski-Phillips-Schmidt-Shin (KPSS) test figures out if a time series is stationary

# around a mean or linear trend.

kpss.test(age)

library(forecast)

ndiffs(age)

diff\_age= diff(age,differences = 1)

plot(diff\_age,type = "b")

adf.test(diff\_age)

kpss.test(diff\_age)

# we get data as stationary

# so my model is ARIMA(p,1,q) since differencing is 1

# ARIMA(p,d,q)=ARIMA(p,1,q)

# Now calculating the Auto Correlation Function for diff\_age

# Remember if acf cuts at any lag it will be q at that lag and p=0

# Remember if pacf cuts at any lag it will be p at that lag and q=0

acf(diff\_age)

#above we see lags only upto 15, if we want to increase the lag use below.

acf(diff\_age,lag.max =48)

# acf is cutting at lag 1 , (we ignore 0 lag) so q=1

# so p=0

#now computing PACF-Partial Autocorrelation Function

pacf(diff\_age)

#pacf is dying down at exponential decay w

model1=arima(age,c(0,1,1))

#ARIMA (0,1,1)

summary(model1)

# now to check which is best fit model we will check accuracy metrices

#like MAPE,MAD,RMSE, AIC, BIC

BIC(model1)

# Considering 2nd assumption as p=3,d=1,q=1 ARIMA(3,1,1)

model2=arima(age,c(3,1,1))

summary(model2)

BIC(model2)

# Considering 3rd assumption as p=3,d=1,q=0 ARIMA(3,1,0)

model3=arima(age,c(3,1,0))

summary(model3)

BIC(model3)

#first preference should be given to lowest BIC for considering best fit model.

# We check auto arima function also to identify automatically p,d,q

model4= auto.arima(age)

summary(model4)

# Here it comes out to be 0,1,1

# Now we are predicting values of the model.

pred=predict(model4)

res=residuals(model4)

#using shapiro test to test normality of residuals.

shapiro.test(res)

#use library portes for box test

# Box test is done to check the white noise

Box.test(res,lag = 15,type = "Ljung-Box")

library(portes) # required for portest package

# Both box test and portest are used for checking white noise.

portest(res)

fcast=forecast(model4,h=3)

plot(fcast,type="b")

# residual - Check formula in copy. we use it to frame final model.

library(readxl)

airline=read\_excel("Airline\_Passenger.xlsx")

attach(airline)

View(airline)

numberts=ts(airline$Number,frequency = 12,start = c(2000,1))

plot(numberts,type="b")

mymodel=decompose(numberts,type="multiplicative")

plot(mymodel,type="b")

# Seasonality Index

SI=mymodel$seasonal

# adjusting seasonality

adjusted= airline$Number/SI

plot(adjusted)

sr=1:144

mydata2= cbind(sr,Number,adjusted)

View(mydata2)

trend= lm(adjusted~sr)

summary(trend)

# actual forecast= trend \* SI ( Seasonality INdex)

#lets see what autoarima gives

automodel=auto.arima(numberts)

summary(automodel)

airpass=read\_excel("Airline\_Passenger.xlsx")

number=ts(airpass$Number,frequency = 12, start = c(2000,1))

diff1=diff(number,lag=12,differences = 1)

plot(diff1,type = "b")

adf.test(diff1)

kpss.test(diff1)

# we see still the data is not stationary

library(tseries)

ndiffs(diff1)

diff2=diff(diff1,lag = 1,differences = 1)

adf.test(diff2)

kpss.test(diff2)

plot(diff2,type = "b")

acf(diff2)

#Observation: ACF plot is cutting at lag 1

Pacf(diff2)

#Observation: Cutting off at lag 1 & exponentially dying.

sarimamodel=arima(number,order = c(0,1,1),seasonal = list(order=c(2,1,0)))

summary(sarimamodel)

BIC(sarimamodel)

pred=predict(sarimamodel)

res=residuals(sarimamodel)

shapiro.test(res)

# even though shapiro test failed it does not carry more weigtage, because Box Test and Portes passed

Box.test(res,lag = 1,type = "Ljung-Box")

portest(res)

pred1= predict(automodel)

res1=residuals(automodel)

shapiro.test(res1)

Box.test(res1,lag = 1,type = "Ljung-Box")

portest(res1)

# forecast from our period

fcast1=forecast(sarimamodel,h=12)

# forecast from auto arima

fcast2=forecast(automodel,h=12)