# Time Series ARIMA Models in R

# Copyright 2020 by Tolga Omay

#install.packages("tseries")

#install.packages("readxl")

library(tseries)

library(readxl)

library(forecast)

mydata <- read\_excel("TimeSeriesR.xlsx")

mydata <- read.xlxs("C:/Users/Tolga OMAY/Desktop/R ile Ekonometri/ARIMA süreçleri/TimeSeriesR.xlsx")

View(TimeSeriesR)

mydata <- TimeSeriesR

View(mydata)

attach(mydata)

str(mydata)

View(mydata)

# Defining variables

y <-ts(mydata[,2],start=c(2000,1),frequency = 12)

plot(y)

TT = dim(mydata)

TT

d.y <- diff(y)

# cumulative series integral

cy <-ts(mydata[,3],start=c(2000,1),frequency = 12)

par(mfrow=c(3,1))

plot(y)

plot(d.y)

plot(dy)

y11 <- mydata[,2]

par(mfrow=c(1,1))

plot(t,y)

trend <- ts(mydata[,1],start=c(2000,1),frequency = 12)

# Descriptive statistics and plotting the data

summary(y)

summary(d.y)

par(mfrow=c(1,1))

hist(y)

plot(y,d.y)

# Dickey-Fuller test for variable

adf.test(y, alternative="stationary", k=3)

#adf.test(y, alternative="explosive", k=0)

#summary(lm(dppi ~ lppi, na.action=na.omit))

#summary(lm(dppi ~ lppi + trend, na.action=na.omit))

# Augmented Dickey-Fuller test

adf.test(y, alternative="stationary")

# DF and ADF tests for differenced variable

adf.test(d.y, k=0)

adf.test(d.y)

y1 <-arima.sim(model=list(ar=c(1.5)),n=100)

par(mfrow=c(1,1))

plot(y1)

#y1

#View(y1)

par(mfrow=c(1,2))

acf(y1)

pacf(y1)

ma.sim<-arima.sim(model=list(ma=c(-0.9)),n=100)

ma.sim

par(mfrow=c(1,1))

plot(y1)

par(mfrow=c(1,2))

acf(ma.sim)

pacf(ma.sim)

arma.sim<-arima.sim(model=list(ar=c(0.9),ma=c(0.7)),n=100)

arma.sim

par(mfrow=c(1,1))

plot(arma.sim)

par(mfrow=c(1,2))

acf(arma.sim)

pacf(arma.sim)

# ACF and PACF

acf(y)

pacf(y)

par(mfrow=c(1,2))

acf(y)

pacf(y)

acf(d.y)

pacf(d.y)

yx <-arima.sim(model=list(ar=c(0.7),ma=c(0.7)),n=100)

plot(yx)

?arima

y1 <-arima.sim(model=list(ar=c(0.7)),n=100)

# ARIMA(1,0,0) or AR(1)

arima(yx, order = c(1,0,0), include.mean = FALSE, optim.method = "CG")

# Teshis Testleri

fit1 <- arima(yx, order = c(1,0,1))

jarque.bera.test(residuals(fit1))

res <- fit1$residuals

Box.test(res, lag=15)

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

#install.packages("aTSA")

library(aTSA)

arch.test(fit1)

par(mfrow=c(1,2))

acf(res^2)

pacf(res^2)

# ARIMA(2,0,0) or AR(2)

arima(yx, order = c(2,0,0), include.mean = FALSE)

# ARIMA(0,0,1) or MA(1)

arima(yx, order = c(0,0,1), include.mean = FALSE)

# ARIMA(1,0,1) or AR(1) MA(1)

arima(yx, order = c(1,0,1), include.mean = FALSE)

arima(y1, order = c(2,0,2))

# ARIMA on differenced variable

# ARIMA(1,1,0)

arima(d.y, order = c(1,0,0))

# ARIMA(0,1,1)

arima(d.y, order = c(0,0,1))

# ARIMA(1,1,1)

arima(d.y, order = c(1,0,1))

# ARIMA(1,1,3)

arima(d.y, order = c(1,0,3))

# ARIMA(2,1,3)

arima(d.y, order = c(2,0,3))

# ARIMA(1,0,1) forecasting

mydata.arima101 <- arima(y, order = c(1,0,0))

mydata.pred1 <- predict(mydata.arima101, n.ahead=10)

plot (y)

lines(mydata.pred1$pred, col="blue")

lines(mydata.pred1$pred+2\*mydata.pred1$se, col="red")

lines(mydata.pred1$pred-2\*mydata.pred1$se, col="red")

# ARIMA(1,1,1) forecasting

mydata.arima111 <- arima(d.y, order = c(1,0,1))

mydata.pred1 <- predict(mydata.arima111, n.ahead=100)

plot (d.y)

lines(mydata.pred1$pred, col="blue")

lines(mydata.pred1$pred+2\*mydata.pred1$se, col="red")

lines(mydata.pred1$pred-2\*mydata.pred1$se, col="red")

###### kütüphane kullanmadan AR(1) modelinin tahmini

l1y<-lag(y, -1)

data1 <-cbind(y,l1y)

data1 <- data1[-dimy,]

AR1\_model <- lm(data1[,1]~data1[,2] -1)

summary(AR1\_model)

b1 <- summary(AR1\_model)$coefficients[1,1]

se <- summary(AR1\_model)$coefficients[1,2]

Rsq <- summary(AR1\_model)$r.squared

ee <-ts(AR1\_model$resid)

b1

se

Rsq

plot(ee)

#####################¦ADF test by using basic commands ###########

y <- ts(y)

dim(y)

dimy<-length(y)

#make lag lag variable and a difference variable

l1y<-lag(y, -1)

dy<-diff(y)

#also make lagged difference

l1dy<-lag(dy, -1)

#put all of them in to new data frame data2

data2 <-cbind(dy,l1y,l1dy)

head(data2)

tail(data2)

#remove the extra observation

data2 <- data2[-dimy,]

tail(data2)

#do regression on adf

ADF<-lm(data2[,1]~data2[,2]+data2[,3] -1)

summary(ADF)

SSE1 <- sum( ADF$resid^2 )

T<-length(y)

BIC1 <- log(SSE1/T)+4\*log(T)/T

BIC1

#####################¦AR(1) model by using basic commands ###########

data3 <-cbind(y,l1y)

ADF<-lm(data3[,1]~data3[,2] -1)

summary(ADF)

ploty <- matplot(cbind(data3[,1],data3[,2]),type="o",col=c("black","red"),lty=c(1,3))

par(mfrow=c(1,1))

abline(reg = lm(data3[,1] ~ data3[,2]))

#cor(data3[,1],data3[,2])

##########################NonLin Model estimation###########

#x.nls <- nls(y ~ a1+a2\*(1/(1+exp(-gamma\*(trend-thr\*dimy)))), start = list(a1=1.0, a2=2.0, gamma = 1.0 , thr= 0.5))

#summary(x.nls)$parameters

x.nls <- nls(y ~ exp(alp0 + alp1\* trend), start = list(alp0 = 1.0 , alp1 = 0.05))

summary(x.nls)$parameters

############################Filtering and decomposition#############

par(mfrow=c(1,1))

hw <- HoltWinters(y)

plot(hw)

dc <- decompose(y)

plot(dc)

Trend1 <- dc$trend

seas <- dc$seasonal

rand <- dc$random

plot(rand)

ploty <- matplot(cbind(y,d.y),type="o",col=c("black","red"),lty=c(1,3))