**CODE1**

source("http://www.rmetrics.org/Rmetrics.R") ## this connects to the file and reads the R code there

install.Rmetrics()

setwd ("C:/Users/jaswa/Desktop/Time Series/project")

sp500daily=read.csv("GSPCdaily26Feb2018.csv",header=T)

sp500daily$Date ## note that the first file contains data in inverse chronological order

sp500.price= sp500daily$Adj.Close[length(sp500daily$Adj.Close):1]

## this puts the data in proper order (first line is the oldest)

sp500.date= sp500daily$Date[length(sp500daily$Adj.Closse):1]

a = c( 1, 2, 3)

a

b = a[-length(a)]

b

sp500.dayreturn=diff(sp500.price)/ sp500.price[-length(sp500.price)] ## daily simple return

sp500.logreturn=diff(log(sp500.price)) ## Cont compounded return

hist(sp500.dayreturn)

hist(sp500.logreturn)

par(mfrow=c(1,2))

hist(sp500.dayreturn, freq=F,xlim=c(-0.25,0.1))

hist(sp500.logreturn,freq=F,xlim=c(-0.25,0.1))

par(mfrow=c(1,2))

plot(1:length(sp500.price),sp500.price,type="l")

plot(1:length(sp500.dayreturn),sp500.dayreturn,type="l")

lines(1:length(sp500.logreturn),sp500.logreturn,col="red")

lines(1:length(sp500.dayreturn),sp500.dayreturn,col="green")

hist(sp500.dayreturn, freq=F)

points(density( sp500.dayreturn),type="l",col="blue")

hist(sp500.dayreturn, freq=F,ylim=c(0,50))

points(density( sp500.dayreturn),type="l",col="blue")

points(density( sp500.dayreturn,width=0.03),type="l",col="lightblue")

points(density( sp500.dayreturn,width=0.05),type="l",col="lightgreen")

points(density( sp500.dayreturn,kernel="gaussian"),type="l",col="red")

points(density( sp500.dayreturn,kernel="epanechnikov"),type="l",col="orange")

points(density( sp500.dayreturn,kernel="cosine"),type="l",col="yellow")

qqnorm(sp500.dayreturn)

#calcultes basic stats of logreturn

mean(sp500.logreturn)

sd(sp500.logreturn)

library(fBasics) ## This loads the package fBasic

basicStats(sp500.logreturn) ## all the stats

## some stats can be accessed directly

mean(sp500.dayreturn)

sd(sp500.dayreturn)

skewness(sp500.dayreturn)

kurtosis(sp500.dayreturn)

basicStats( sp500.dayreturn)

t.test(sp500.dayreturn)

basicStats( sp500.logreturn)

t.test(sp500.logreturn)

## Normality tests

## Check documentation on normalTest

normalTest(sp500.dayreturn,method="jb")

par(mfrow=c(1,0))

acf(sp500.dayreturn,lag=15) # Obtain the ACF plot

Box.test(sp500.dayreturn,lag=10)

Box.test(sp500.dayreturn,lag=10,type="Ljung")

length(sp500.dayreturn)

par(mfcol=c(2,2)) # put 4 plots on one page

plot(sp500.dayreturn,type='l') # first plot

plot(sp500.dayreturn[1:(length(sp500.dayreturn)-1)],sp500.dayreturn[2:length(sp500.dayreturn)])

# lag 1 plot

plot(sp500.dayreturn[1:(length(sp500.dayreturn)-2)],sp500.dayreturn[3:length(sp500.dayreturn)])

# lag 2 plot

acf(sp500.dayreturn,lag=15)

par(mfcol=c(1,1))

model1=ar( sp500.dayreturn ,method="mle") # Automatic AR fitting using AIC criterion.

model1 ## AR(2) is specified

names(model1)

plot(model1$resid,type='l') ## checks residuals

Box.test(model1$resid,lag=10,type='Ljung')

model1$x.mean # Predicted overal mean value (daily)

[1] -0.000364869

## Another approach with order specified

model2= arima(sp500.dayreturn, order=c(3,0,0))

model2

par(mai=c(0.5,0.5,0.5,0.5))

tsdiag(model2)

Box.test(model2$residuals,lag=10,type='Ljung')

plot(model2$residuals,type='l')

## Further analysis:

poly1=c(1,-model2$coef[1:3])

roots=polyroot(poly1)

roots

Mod(roots)

predict(model2,10) ## predict 10 days ahead

model3= arima(sp500.dayreturn, order=c(3,0,1))

model3

par(mai=c(0.5,0.5,0.5,0.5))

tsdiag(model3)

Box.test(model3$residuals,lag=10,type='Ljung')

plot(model3$residuals,type='l')

## Further analysis:

poly2=c(1,-model3$coef[1:3])

roots1=polyroot(poly2)

roots1

Mod(roots1)

predict(model3,10)

model4= arima(sp500.dayreturn, order=c(2,0,1))

model4

par(mai=c(0.5,0.5,0.5,0.5))

tsdiag(model4)

Box.test(model4$residuals,lag=10,type='Ljung')

plot(model4$residuals,type='l')

## Further analysis:

poly3=c(1,-model4$coef[1:3])

roots2=polyroot(poly3)

roots2

Mod(roots2)

predict(model4,10)

model5= arima(sp500.dayreturn, order=c(2,0,0))

model5

par(mai=c(0.5,0.5,0.5,0.5))

tsdiag(model5)

Box.test(model5$residuals,lag=10,type='Ljung')

plot(model5$residuals,type='l')

## Further analysis:

poly4=c(1,-model5$coef[1:3])

roots3=polyroot(poly4)

roots3

Mod(roots3)

predict(model5,10)

model6= arima(sp500.dayreturn, order=c(1,0,0))

model6

par(mai=c(0.5,0.5,0.5,0.5))

tsdiag(model6)

Box.test(model6$residuals,lag=10,type='Ljung')

plot(model6$residuals,type='l')

## Further analysis:

poly5=c(1,-model6$coef[1:2])

roots4=polyroot(poly5)

roots4

Mod(roots4)

predict(model6,10)

model7= arima(sp500.dayreturn, order=c(1,0,2))

model7

par(mai=c(0.5,0.5,0.5,0.5))

tsdiag(model7)

Box.test(model7$residuals,lag=10,type='Ljung')

plot(model7$residuals,type='l')

## Further analysis:

poly6=c(1,-model7$coef[1:3])

roots5=polyroot(poly6)

roots5

Mod(roots5)

predict(model7,10)

**CODE2**

source("http://www.rmetrics.org/Rmetrics.R") ## this connects to the file and reads the R code there

install.Rmetrics()

setwd ("C:/Users/jaswa/Desktop/Time Series/project")

MSFT=read.csv("TestMSFT20051111.csv",header=T) ## First file is daily data, second file is minute data from some random day

MSFT

MSFT.price= MSFT$close ## this puts the data in proper order (first line is the oldest)

MSFT.price

MSFT.dayreturn=diff(MSFT.price)/ MSFT.price[-length(MSFT.price)] ## daily simple return

MSFT.logreturn=diff(log(MSFT.price)) ## Cont compounded return

MSFT.dayreturn

MSFT.logreturn

hist(MSFT.dayreturn)

hist(MSFT.logreturn)

par(mfrow=c(1,2))

hist(MSFT.dayreturn, freq=F,xlim=c(-0.25,0.1))

hist(MSFT.logreturn,freq=F,xlim=c(-0.25,0.1))

par(mfrow=c(1,2))

plot(1:length(MSFT.price),MSFT.price,type="l")

plot(1:length(MSFT.dayreturn),MSFT.dayreturn,type="l")

lines(1:length(MSFT.logreturn),MSFT.logreturn,col="red")

lines(1:length(MSFT.dayreturn),MSFT.dayreturn,col="green")

hist(MSFT.dayreturn, freq=F)

points(density(MSFT.dayreturn),type="l",col="blue")

hist(MSFT.dayreturn, freq=F,ylim=c(0,2600))

points(density(MSFT.dayreturn),type="l",col="blue")

points(density(MSFT.dayreturn,width=0.03),type="l",col="lightblue")

points(density(MSFT.dayreturn,width=0.05),type="l",col="lightgreen")

points(density(MSFT.dayreturn,kernel="gaussian"),type="l",col="red")

points(density(MSFT.dayreturn,kernel="epanechnikov"),type="l",col="orange")

points(density(MSFT.dayreturn,kernel="cosine"),type="l",col="yellow")

qqnorm(MSFT.dayreturn)

#calcultes basic stats of logreturn

mean(MSFT.logreturn)

sd(MSFT.logreturn)

library(fBasics) ## This loads the package fBasic

basicStats(MSFT.logreturn) ## all the stats

## some stats can be accessed directly

mean(MSFT.dayreturn)

sd(MSFT.dayreturn)

skewness(MSFT.dayreturn)

kurtosis(MSFT.dayreturn)

basicStats(MSFT.dayreturn)

t.test(MSFT.dayreturn)

basicStats(MSFT.logreturn)

t.test(MSFT.logreturn)

normalTest(MSFT.dayreturn,method="jb")

par(mfrow=c(1,0))

acf(MSFT.dayreturn,lag=15) # Obtain the ACF plot

Box.test(MSFT.dayreturn,lag=10)

Box.test(MSFT.dayreturn,lag=10,type="Ljung")

length(MSFT.dayreturn)

par(mfcol=c(2,2)) # put 4 plots on one page

plot(MSFT.dayreturn,type='l') # first plot

plot(MSFT.dayreturn[1:(length(MSFT.dayreturn)-1)],MSFT.dayreturn[2:length(MSFT.dayreturn)])

# lag 1 plot

plot(MSFT.dayreturn[1:(length(MSFT.dayreturn)-2)],MSFT.dayreturn[3:length(MSFT.dayreturn)])

# lag 2 plot

acf(MSFT.dayreturn,lag=15)

par(mfcol=c(1,1))

model1=ar(MSFT.dayreturn ,method="mle") # Automatic AR fitting using AIC criterion.

model1 ## AR(2) is specified

names(model1)

plot(model1$resid,type='l') ## checks residuals

Box.test(model1$resid,lag=10,type='Ljung')

model1$x.mean # Predicted overal mean value (daily)

[1] -1.03936e-05

## Another approach with order specified

model2= arima(MSFT.dayreturn, order=c(3,0,0))

model2

par(mai=c(0.5,0.5,0.5,0.5))

tsdiag(model2)

Box.test(model2$residuals,lag=10,type='Ljung')

plot(model2$residuals,type='l')

## Further analysis:

poly1=c(1,-model2$coef[1:3])

roots=polyroot(poly1)

roots

Mod(roots)

predict(model2,10) ## predict 10 days ahead

model3= arima(MSFT.dayreturn, order=c(3,0,1))

model3

par(mai=c(0.5,0.5,0.5,0.5))

tsdiag(model3)

Box.test(model3$residuals,lag=10,type='Ljung')

plot(model3$residuals,type='l')

## Further analysis:

poly2=c(1,-model3$coef[1:3])

roots1=polyroot(poly2)

roots1

Mod(roots1)

predict(model3,10)

model4= arima(MSFT.dayreturn, order=c(2,0,1))

model4

par(mai=c(0.5,0.5,0.5,0.5))

tsdiag(model4)

Box.test(model4$residuals,lag=10,type='Ljung')

plot(model4$residuals,type='l')

## Further analysis:

poly3=c(1,-model4$coef[1:3])

roots2=polyroot(poly3)

roots2

Mod(roots2)

predict(model4,10)

model5= arima(MSFT.dayreturn, order=c(2,0,0))

model5

par(mai=c(0.5,0.5,0.5,0.5))

tsdiag(model5)

Box.test(model5$residuals,lag=10,type='Ljung')

plot(model5$residuals,type='l')

## Further analysis:

poly4=c(1,-model5$coef[1:3])

roots3=polyroot(poly4)

roots3

Mod(roots3)

predict(model5,10)

model6= arima(MSFT.dayreturn, order=c(1,0,0))

model6

par(mai=c(0.5,0.5,0.5,0.5))

tsdiag(model6)

Box.test(model6$residuals,lag=10,type='Ljung')

plot(model6$residuals,type='l')

## Further analysis:

poly5=c(1,-model6$coef[1:2])

roots4=polyroot(poly5)

roots4

Mod(roots4)

predict(model6,10)

model7= arima(MSFT.dayreturn, order=c(1,0,1))

model7

par(mai=c(0.5,0.5,0.5,0.5))

tsdiag(model7)

Box.test(model7$residuals,lag=10,type='Ljung')

plot(model7$residuals,type='l')

## Further analysis:

poly6=c(1,-model7$coef[1:3])

roots5=polyroot(poly6)

roots5

Mod(roots5)

predict(model7,10)

**CODE1**

source("http://www.rmetrics.org/Rmetrics.R") ## this connects to the file and reads the R code there

install.Rmetrics()

setwd ("C:/Users/jaswa/Desktop/Time Series/project")

sp500daily=read.csv("GSPCdaily26Feb2018.csv",header=T)

sp500daily$Date ## note that the first file contains data in inverse chronological order

sp500.price= sp500daily$Adj.Close[length(sp500daily$Adj.Close):1]

## this puts the data in proper order (first line is the oldest)

sp500.date= sp500daily$Date[length(sp500daily$Adj.Close):1]

a = c( 1, 2, 3)

a

b = a[-length(a)]

b

sp500.dayreturn=diff(sp500.price)/ sp500.price[-length(sp500.price)] ## daily simple return

sp500.logreturn=diff(log(sp500.price)) ## Cont compounded return

hist(sp500.dayreturn)

hist(sp500.logreturn)

par(mfrow=c(1,2))

hist(sp500.dayreturn, freq=F,xlim=c(-0.25,0.1))

hist(sp500.logreturn,freq=F,xlim=c(-0.25,0.1))

par(mfrow=c(1,2))

plot(1:length(sp500.price),sp500.price,type="l")

plot(1:length(sp500.dayreturn),sp500.dayreturn,type="l")

lines(1:length(sp500.logreturn),sp500.logreturn,col="red")

lines(1:length(sp500.dayreturn),sp500.dayreturn,col="green")

hist(sp500.dayreturn, freq=F)

points(density( sp500.dayreturn),type="l",col="blue")

hist(sp500.dayreturn, freq=F,ylim=c(0,50))

points(density( sp500.dayreturn),type="l",col="blue")

points(density( sp500.dayreturn,width=0.03),type="l",col="lightblue")

points(density( sp500.dayreturn,width=0.05),type="l",col="lightgreen")

points(density( sp500.dayreturn,kernel="gaussian"),type="l",col="red")

points(density( sp500.dayreturn,kernel="epanechnikov"),type="l",col="orange")

points(density( sp500.dayreturn,kernel="cosine"),type="l",col="yellow")

qqnorm(sp500.dayreturn)

#calcultes basic stats of logreturn

mean(sp500.logreturn)

sd(sp500.logreturn)

library(fBasics) ## This loads the package fBasic

basicStats(sp500.logreturn) ## all the stats

## some stats can be accessed directly

mean(sp500.dayreturn)

sd(sp500.dayreturn)

skewness(sp500.dayreturn)

kurtosis(sp500.dayreturn)

basicStats( sp500.dayreturn)

t.test(sp500.dayreturn)

basicStats( sp500.logreturn)

t.test(sp500.logreturn)

## Normality tests

## Check documentation on normalTest

normalTest(sp500.dayreturn,method="jb")

par(mfrow=c(1,0))

acf(sp500.dayreturn,lag=15) # Obtain the ACF plot

Box.test(sp500.dayreturn,lag=10)

Box.test(sp500.dayreturn,lag=10,type="Ljung")

length(sp500.dayreturn)

par(mfcol=c(2,2)) # put 4 plots on one page

plot(sp500.dayreturn,type='l') # first plot

plot(sp500.dayreturn[1:(length(sp500.dayreturn)-1)],sp500.dayreturn[2:length(sp500.dayreturn)])

# lag 1 plot

plot(sp500.dayreturn[1:(length(sp500.dayreturn)-2)],sp500.dayreturn[3:length(sp500.dayreturn)])

# lag 2 plot

acf(sp500.dayreturn,lag=15)

par(mfcol=c(1,1))

model1=ar( sp500.dayreturn ,method="mle") # Automatic AR fitting using AIC criterion.

model1 ## AR(2) is specified

names(model1)

plot(model1$resid,type='l') ## checks residuals

Box.test(model1$resid,lag=10,type='Ljung')

model1$x.mean # Predicted overal mean value (daily)

[1] -0.000364869

## Another approach with order specified

model2= arima(sp500.dayreturn, order=c(3,0,0))

model2

par(mai=c(0.5,0.5,0.5,0.5))

tsdiag(model2)

Box.test(model2$residuals,lag=10,type='Ljung')

plot(model2$residuals,type='l')

## Further analysis:

poly1=c(1,-model2$coef[1:3])

roots=polyroot(poly1)

roots

Mod(roots)

predict(model2,10) ## predict 10 days ahead

model3= arima(sp500.dayreturn, order=c(3,0,1))

model3

par(mai=c(0.5,0.5,0.5,0.5))

tsdiag(model3)

Box.test(model3$residuals,lag=10,type='Ljung')

plot(model3$residuals,type='l')

## Further analysis:

poly2=c(1,-model3$coef[1:3])

roots1=polyroot(poly2)

roots1

Mod(roots1)

predict(model3,10)

model4= arima(sp500.dayreturn, order=c(2,0,1))

model4

par(mai=c(0.5,0.5,0.5,0.5))

tsdiag(model4)

Box.test(model4$residuals,lag=10,type='Ljung')

plot(model4$residuals,type='l')

## Further analysis:

poly3=c(1,-model4$coef[1:3])

roots2=polyroot(poly3)

roots2

Mod(roots2)

predict(model4,10)

model5= arima(sp500.dayreturn, order=c(2,0,0))

model5

par(mai=c(0.5,0.5,0.5,0.5))

tsdiag(model5)

Box.test(model5$residuals,lag=10,type='Ljung')

plot(model5$residuals,type='l')

## Further analysis:

poly4=c(1,-model5$coef[1:3])

roots3=polyroot(poly4)

roots3

Mod(roots3)

predict(model5,10)

model6= arima(sp500.dayreturn, order=c(1,0,0))

model6

par(mai=c(0.5,0.5,0.5,0.5))

tsdiag(model6)

Box.test(model6$residuals,lag=10,type='Ljung')

plot(model6$residuals,type='l')

## Further analysis:

poly5=c(1,-model6$coef[1:2])

roots4=polyroot(poly5)

roots4

Mod(roots4)

predict(model6,10)

model7= arima(sp500.dayreturn, order=c(1,0,2))

model7

par(mai=c(0.5,0.5,0.5,0.5))

tsdiag(model7)

Box.test(model7$residuals,lag=10,type='Ljung')

plot(model7$residuals,type='l')

## Further analysis:

poly6=c(1,-model7$coef[1:3])

roots5=polyroot(poly6)

roots5

Mod(roots5)

predict(model7,10)

**CODE2**

source("http://www.rmetrics.org/Rmetrics.R") ## this connects to the file and reads the R code there

install.Rmetrics()

setwd ("C:/Users/jaswa/Desktop/Time Series/project")

MSFT=read.csv("TestMSFT20051111.csv",header=T) ## First file is daily data, second file is minute data from some random day

MSFT

MSFT.price= MSFT$close ## this puts the data in proper order (first line is the oldest)

MSFT.price

MSFT.dayreturn=diff(MSFT.price)/ MSFT.price[-length(MSFT.price)] ## daily simple return

MSFT.logreturn=diff(log(MSFT.price)) ## Cont compounded return

MSFT.dayreturn

MSFT.logreturn

hist(MSFT.dayreturn)

hist(MSFT.logreturn)

par(mfrow=c(1,2))

hist(MSFT.dayreturn, freq=F,xlim=c(-0.25,0.1))

hist(MSFT.logreturn,freq=F,xlim=c(-0.25,0.1))

par(mfrow=c(1,2))

plot(1:length(MSFT.price),MSFT.price,type="l")

plot(1:length(MSFT.dayreturn),MSFT.dayreturn,type="l")

lines(1:length(MSFT.logreturn),MSFT.logreturn,col="red")

lines(1:length(MSFT.dayreturn),MSFT.dayreturn,col="green")

hist(MSFT.dayreturn, freq=F)

points(density(MSFT.dayreturn),type="l",col="blue")

hist(MSFT.dayreturn, freq=F,ylim=c(0,2600))

points(density(MSFT.dayreturn),type="l",col="blue")

points(density(MSFT.dayreturn,width=0.03),type="l",col="lightblue")

points(density(MSFT.dayreturn,width=0.05),type="l",col="lightgreen")

points(density(MSFT.dayreturn,kernel="gaussian"),type="l",col="red")

points(density(MSFT.dayreturn,kernel="epanechnikov"),type="l",col="orange")

points(density(MSFT.dayreturn,kernel="cosine"),type="l",col="yellow")

qqnorm(MSFT.dayreturn)

#calcultes basic stats of logreturn

mean(MSFT.logreturn)

sd(MSFT.logreturn)

library(fBasics) ## This loads the package fBasic

basicStats(MSFT.logreturn) ## all the stats

## some stats can be accessed directly

mean(MSFT.dayreturn)

sd(MSFT.dayreturn)

skewness(MSFT.dayreturn)

kurtosis(MSFT.dayreturn)

basicStats(MSFT.dayreturn)

t.test(MSFT.dayreturn)

basicStats(MSFT.logreturn)

t.test(MSFT.logreturn)

normalTest(MSFT.dayreturn,method="jb")

par(mfrow=c(1,0))

acf(MSFT.dayreturn,lag=15) # Obtain the ACF plot

Box.test(MSFT.dayreturn,lag=10)

Box.test(MSFT.dayreturn,lag=10,type="Ljung")

length(MSFT.dayreturn)

par(mfcol=c(2,2)) # put 4 plots on one page

plot(MSFT.dayreturn,type='l') # first plot

plot(MSFT.dayreturn[1:(length(MSFT.dayreturn)-1)],MSFT.dayreturn[2:length(MSFT.dayreturn)])

# lag 1 plot

plot(MSFT.dayreturn[1:(length(MSFT.dayreturn)-2)],MSFT.dayreturn[3:length(MSFT.dayreturn)])

# lag 2 plot

acf(MSFT.dayreturn,lag=15)

par(mfcol=c(1,1))

model1=ar(MSFT.dayreturn ,method="mle") # Automatic AR fitting using AIC criterion.

model1 ## AR(2) is specified

names(model1)

plot(model1$resid,type='l') ## checks residuals

Box.test(model1$resid,lag=10,type='Ljung')

model1$x.mean # Predicted overal mean value (daily)

[1] -1.03936e-05

## Another approach with order specified

model2= arima(MSFT.dayreturn, order=c(3,0,0))

model2

par(mai=c(0.5,0.5,0.5,0.5))

tsdiag(model2)

Box.test(model2$residuals,lag=10,type='Ljung')

plot(model2$residuals,type='l')

## Further analysis:

poly1=c(1,-model2$coef[1:3])

roots=polyroot(poly1)

roots

Mod(roots)

predict(model2,10) ## predict 10 days ahead

model3= arima(MSFT.dayreturn, order=c(3,0,1))

model3

par(mai=c(0.5,0.5,0.5,0.5))

tsdiag(model3)

Box.test(model3$residuals,lag=10,type='Ljung')

plot(model3$residuals,type='l')

## Further analysis:

poly2=c(1,-model3$coef[1:3])

roots1=polyroot(poly2)

roots1

Mod(roots1)

predict(model3,10)

model4= arima(MSFT.dayreturn, order=c(2,0,1))

model4

par(mai=c(0.5,0.5,0.5,0.5))

tsdiag(model4)

Box.test(model4$residuals,lag=10,type='Ljung')

plot(model4$residuals,type='l')

## Further analysis:

poly3=c(1,-model4$coef[1:3])

roots2=polyroot(poly3)

roots2

Mod(roots2)

predict(model4,10)

model5= arima(MSFT.dayreturn, order=c(2,0,0))

model5

par(mai=c(0.5,0.5,0.5,0.5))

tsdiag(model5)

Box.test(model5$residuals,lag=10,type='Ljung')

plot(model5$residuals,type='l')

## Further analysis:

poly4=c(1,-model5$coef[1:3])

roots3=polyroot(poly4)

roots3

Mod(roots3)

predict(model5,10)

model6= arima(MSFT.dayreturn, order=c(1,0,0))

model6

par(mai=c(0.5,0.5,0.5,0.5))

tsdiag(model6)

Box.test(model6$residuals,lag=10,type='Ljung')

plot(model6$residuals,type='l')

## Further analysis:

poly5=c(1,-model6$coef[1:2])

roots4=polyroot(poly5)

roots4

Mod(roots4)

predict(model6,10)

model7= arima(MSFT.dayreturn, order=c(1,0,1))

model7

par(mai=c(0.5,0.5,0.5,0.5))

tsdiag(model7)

Box.test(model7$residuals,lag=10,type='Ljung')

plot(model7$residuals,type='l')

## Further analysis:

poly6=c(1,-model7$coef[1:3])

roots5=polyroot(poly6)

roots5

Mod(roots5)

predict(model7,10)