library(forecast)

netflix <- read.csv("dailystockpricesfor1year.csv") #daily stock values for year 2021 (251 obs)

#using adjusted average just as the paper

library(tidyverse)

netflix\_1 <-netflix%>%select(Date, Adj.Close)

head(netflix\_1)

#changeing date to actual date and not just numeric

netflix\_1$Date <- as.Date(netflix\_1$Date, "%m/%d/%Y")

head(netflix\_1)

# from Jan 4,2021 to Dec 30, 2021 as a time series object

TS <- ts(netflix\_1, start=c(2021-01-04, 1), end=c(2021-12-30, 251))

plot(TS)

# Libraries

library(ggplot2)

library(dplyr)

# plot of time series

p <- ggplot(netflix\_1, aes(x=Date, y=Adj.Close)) +

geom\_line() +

xlab("")

p

acf(netflix\_1$Adj.Close, main='Autocorrelations', ylab='', ylim=c(-1, 1), xlim=c(0, 15), ci.col = "black")

pacf(netflix\_1$Adj.Close, main='Partial Autocorrelations', ylab='', ylim=c(-1, 1), ci.col = "black")

Diff1 <- diff(netflix\_1$Adj.Close, differences = 1) # Another way to do differencing

plot(Diff1)

acf(Diff1, main='Autocorrelations', ylab='', ylim=c(-1, 1), xlim=c(0, 15), ci.col = "black")

pacf(Diff1, main='Partial Autocorrelations', ylab='', ylim=c(-1, 1), ci.col = "black")

# Fit Model1 could be ARIMA(0,1,1)X(0,1,1)12 to first few years of AirPassengers data

Model1 <- Arima(netflix\_1$Adj.Close,order=c(0,1,0),lambda=NULL)

# Lambda is Box-Cox transformation parameter. If lambda="auto", then a transformation is automatically selected using BoxCox.lambda. The transformation is ignored if NULL. Otherwise, data transformed before model is estimated.

# Note that Lambda=0 means log transformation under Box-Cox transformations.

Model1

modelARIMA.1 <- auto.arima(netflix\_1$Adj.Close)

modelARIMA.1

install.packages("quantmod")

install.packages("xts")

install.packages("PerformanceAnalytics")

install.packages("rugarch")

# Libraries

library(quantmod)

library(xts)

library(PerformanceAnalytics)

library(rugarch)

library(forecast)

library(tidyverse)

# Daily returns

return <- CalculateReturns(netflix\_1$Adj.Close)

return <- diff(netflix\_1$Adj.Close, differences = 1) # Another way to do differencing

return <- return[-1]

hist(return)

plot(return)

# 1. sGARCH model with contant mean

s1 <- ugarchspec(mean.model = list(armaOrder = c(0,0)),

variance.model = list(model = "sGARCH"),

distribution.model = 'norm')

m1 <- ugarchfit(data = return, spec = s1)

m1

plot(m1)

#Remark all parameters sig, the q-q plot shows heavy tails

# 2. sGARCH model with contant mean with t-distribution for error terms

s1t <- ugarchspec(mean.model = list(armaOrder = c(0,0)),

variance.model = list(model = "sGARCH"),

distribution.model = 'sstd')

m1t <- ugarchfit(data = return, spec = s1t)

m1t

plot(m1t)

# 3. ARCH(1) model is the sGARCH(0,1) model

s2 <- ugarchspec(mean.model = list(armaOrder = c(0,0)),

variance.model = list(model = "sGARCH", garchOrder = c(0, 1)),

distribution.model = 'norm')

m2 <- ugarchfit(data = return, spec = s2)

m2

plot(m2)

#4 compute the ARMA(1,1)-GARCH(1,1) model as an example:

s4 <- ugarchspec(mean.model = list(armaOrder = c(1,1)),

variance.model = list(model = "sGARCH", garchOrder = c(0, 1)),

distribution.model = 'norm')

m4 <- ugarchfit(data = return, spec = s4)

m4

plot(m4)

help(ugarchspec)

f2 <- ugarchforecast(fitORspec = m5, n.ahead = 20)

plot(fitted(f5))

plot(sigma(f5))