ISYE 6402 Homework 6 Solutions

## Background

Individual stock prices tend to exhibit high amounts of non-constant variance, and thus ARIMA models built upon that data would likely exhibit non-constant variance in residuals. In this problem we are going to analyze the Starbucks stock price data from 2011 through end of 2021. We will use the ARIMA-GARCH to model daily and weekly stock price (adjusted close price at the end of a day for daily data or at the end of the week for weekly data), with a focus on the behavior of its volatility as well as forecasting both the price and the volatility.

##Data import and cleaning

## Libraries used within this homework are uploaded here  
library(zoo,warn.conflicts=FALSE)  
library(lubridate,warn.conflicts=FALSE)  
library(mgcv,warn.conflicts=FALSE)  
library(rugarch,warn.conflicts=FALSE)

#importing the data  
dailydata <- read.csv("INTCDaily.csv", head = TRUE)  
weeklydata <- read.csv("INTCWeekly.csv", head = TRUE)  
  
#cleaning the data  
  
#dates to date format  
weeklydata$Date<-as.Date(weeklydata$Date,format='%m/%d/%y')  
dailydata$Date<-as.Date(dailydata$Date,format='%m/%d/%y')  
  
#prices to timeseries format  
INTWeekly <- ts(weeklydata$Adj.Close,start=c(2012,1,1),freq=52)  
INTDaily <- ts(dailydata$Adj.Close,start=c(2012,1,1),freq=252)

#Question 1: Exploratory Data Analysis (20 points)

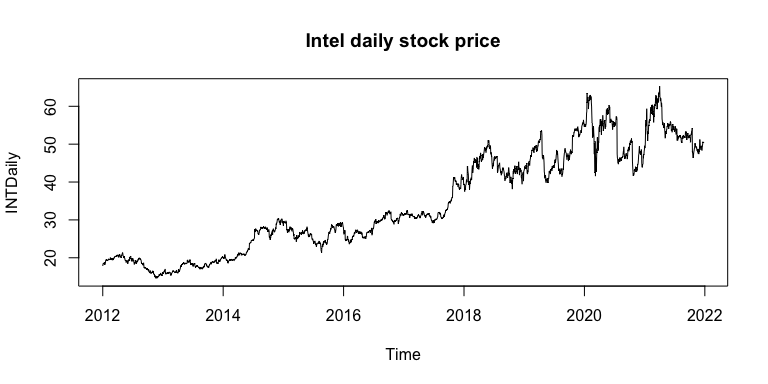
**1a.** Based on your intuition, when would you use daily vs weekly stock price data?

*Response: Question 1a*

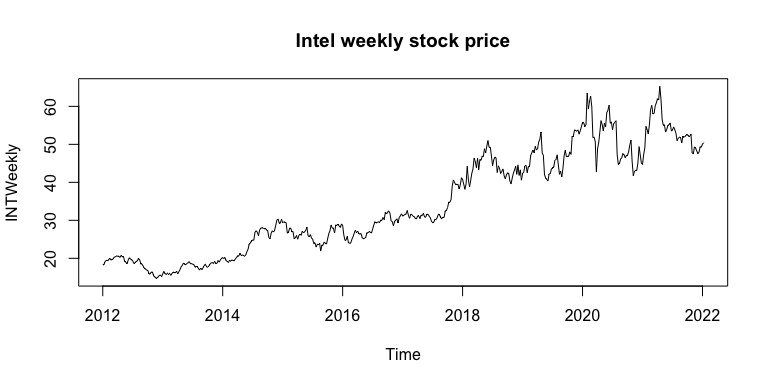
It would be a better idea to use daily data as opposed to the weekly data when the price movements seem to be displaying volatility at shorter time intervals. However, in the case of Intel, the overall change of stock price over longer intervals of time would seem large enough so as to not require analysis over shorter intervals of time.

**1b.** Plot the time series plots comparing daily vs weekly data. How do the daily vs weekly time series data compare?

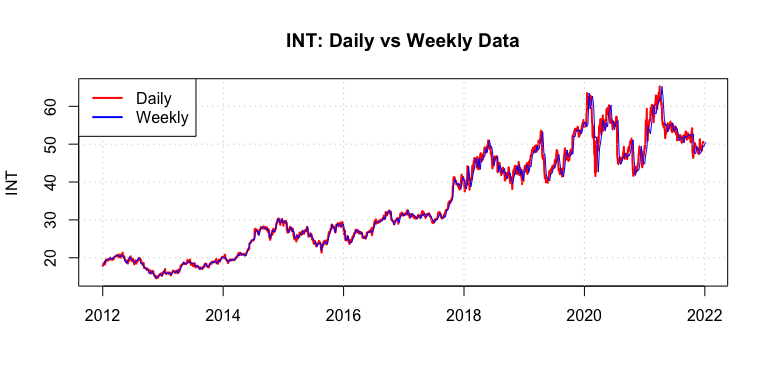
plot(INTDaily,main="Intel daily stock price",type="l")



plot(INTWeekly,main="Intel weekly stock price",type="l")



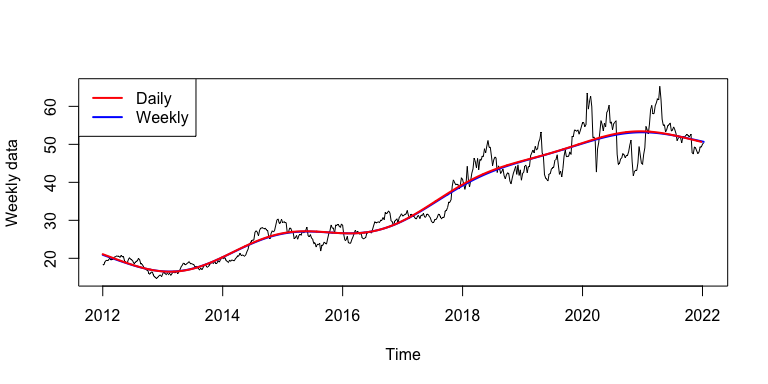
### THIS IS AN ALTERNATIVE GRAPH  
plot(INTDaily, xlab = "", ylab = "INT", col = "red", lwd = 2,type="l",  
main = "INT: Daily vs Weekly Data")  
grid()  
lines(INTWeekly, col = "blue", lwd = 1, type='l')  
legend("topleft", c("Daily", "Weekly"), col = c("red", "blue"), lwd = 2)

 *Response: Question 1b*

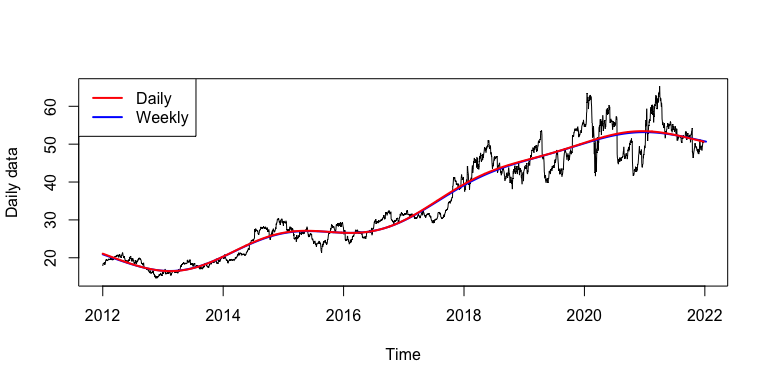
As expected, the Intel stock seems to be more volatile in recent years and this volatility in price is significant enough for us to consider the weekly data as sufficient for analyzing the movements, trends, stationarity and residual analysis. The movements in mean and variance seem to be quite significant over the whole time period; the weekly data, which is only slightly smoothed over from the daily data, should suffice for our analysis.

**1c.** Fit a non-parametric trend using splines regression to both the daily and weekly time-series data. Overlay the fitted trends. How do the trends compare?

time.ptsw = c(1:length(INTWeekly))  
time.ptsw = c(time.ptsw - min(time.ptsw))/max(time.ptsw)  
time.ptsd = c(1:length(INTDaily))  
time.ptsd = c(time.ptsd - min(time.ptsd))/max(time.ptsd)  
egam.fit.weekly = gam(INTWeekly~s(time.ptsw))  
eu.fit.gam.weekly = ts(fitted(egam.fit.weekly),c(2012,1,1),frequency=52)  
egam.fit.daily = gam(INTDaily~s(time.ptsd))  
eu.fit.gam.daily = ts(fitted(egam.fit.daily),start=c(2012,1,1),frequency=252)  
## Is there a trend? Which data does it fit better?  
  
ts.plot(INTWeekly,ylab="Weekly data")  
lines(eu.fit.gam.weekly,lwd=2,col="blue")  
lines(eu.fit.gam.daily,lwd=2,col="red")  
legend("topleft", c("Daily", "Weekly"), col = c("red", "blue"), lwd = 2)



ts.plot(INTDaily,ylab="Daily data")  
lines(eu.fit.gam.weekly,lwd=2,col="blue")  
lines(eu.fit.gam.daily,lwd=2,col="red")  
legend("topleft", c("Daily", "Weekly"), col = c("red", "blue"), lwd = 2)

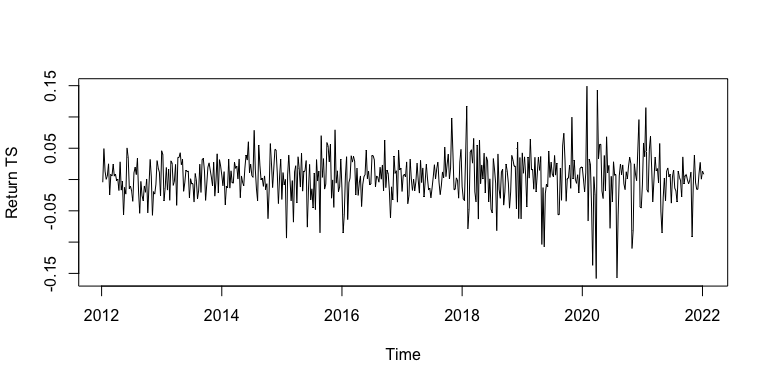


*Response: Question 1c*

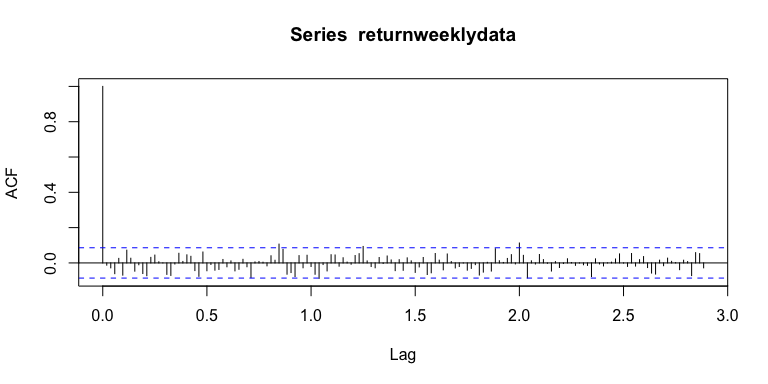
The weekly data seems to encapsulate the trend fit well, which gives us reason to not go by daily trend fitting.

**1d.** Consider the return stock price computed as provided in the canvas homework assignment. Apply this formula to compute the return price based on the daily and weekly time series data. Plot the return time series and their corresponding ACF plots. How do the return time series compare in terms of stationarity and serial dependence?

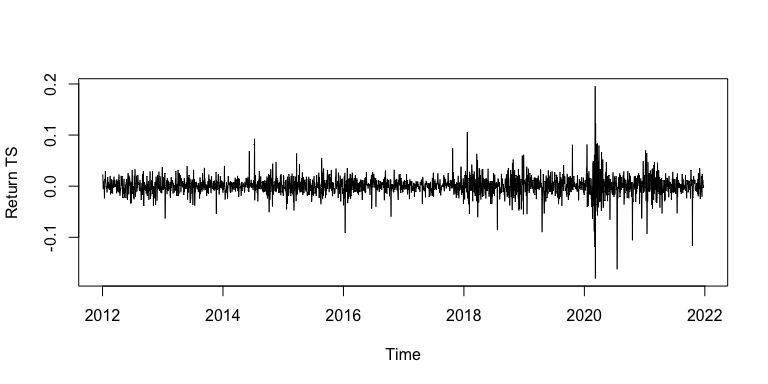
dataweeklydiff=diff(INTWeekly)  
returnweeklydata=dataweeklydiff/INTWeekly[-length(INTWeekly)]  
plot(returnweeklydata, ylab= "Return TS")



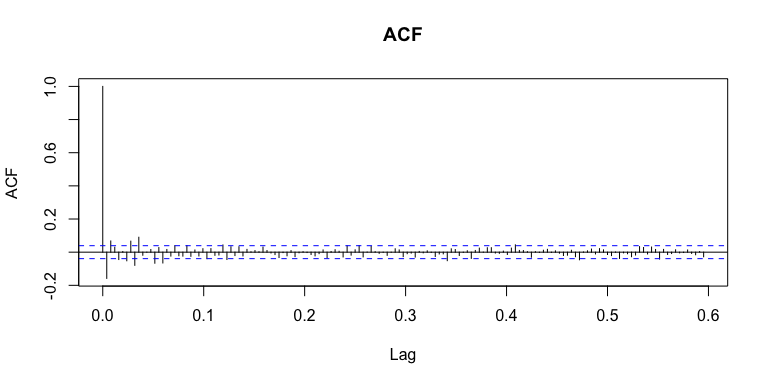
acf(returnweeklydata,lag.max=6\*25)



datadailydiff=diff(INTDaily)  
returndailydata=datadailydiff/INTDaily[-length(INTDaily)]  
plot(returndailydata,ylab= "Return TS")



acf(returndailydata,lag.max=6\*25, main='ACF')



*Response: Question 1d*

The residual acf plots for the return time series indicate stationarity; in fact, the acf plots look just like those of white noise, hence potentially indicating no serial dependence.

#Question 2: ARIMA(p,d,q) for Stock Price (20 Points)

**2a.** Divide the data into training and testing data set, where the training data exclude the last week of data (December 27th - December 30th) with the testing data including the last week of data. Apply the iterative model to fit an ARIMA(p,d,q) model with max AR and MA orders of 8 and difference orders 1 and 2 separately to the training datasets of the daily and weekly data. Display the summary of the final model fit.

#dividing the data into training and testing  
INTWeekly.train=INTWeekly[1:(length(INTWeekly)-1)]  
INTWeekly.test=INTWeekly[(length(INTWeekly)):length(INTWeekly)]  
INTDaily.train=INTDaily[1:(length(INTDaily)-4)]  
INTDaily.test=INTDaily[(length(INTDaily)-3):length(INTDaily)]  
  
############# Intel Weekly Data ##########################  
# Weekly d=1  
n = length(INTWeekly)   
####### w\_n, deleted w\_  
w\_n\_fit=length(INTWeekly.train)  
w\_n\_forward=n-w\_n\_fit  
w\_n=length(INTWeekly.train)  
norder = 8  
orders = data.frame()  
p = c(0:norder); q = c(0:norder)  
aic = matrix(0,norder,norder)  
for(i in 1:norder){  
for(j in 1:norder){  
modij = stats::arima(INTWeekly.train,order = c(p[i],1,q[j]), method='ML')  
current.aic=modij$aic-2\*(p[i]+q[j]+1)+2\*(p[i]+q[j]+1)\*n/(n-p[i]-q[j]-2)  
orders<-rbind(orders,c(p[i],q[j],current.aic))  
}  
}  
names(orders) <- c("p","q","AIC")  
# Extract the "best" one according to AIC  
orders[which.min(orders$AIC),]

## p q AIC  
## 38 4 5 1907.713

# Weekly d=2  
n = length(INTWeekly) ####### w\_n, deleted w\_  
w\_n\_fit=length(INTWeekly.train)  
w\_n\_forward=n-w\_n\_fit  
w\_n=length(INTWeekly.train)  
norder = 8  
orders = data.frame()  
p = c(0:norder); q = c(0:norder)  
aic = matrix(0,norder,norder)  
for(i in 1:norder){  
for(j in 1:norder){  
modij = stats::arima(INTWeekly.train,order = c(p[i],2,q[j]), method='ML')  
current.aic=modij$aic-2\*(p[i]+q[j]+1)+2\*(p[i]+q[j]+1)\*n/(n-p[i]-q[j]-2)  
orders<-rbind(orders,c(p[i],q[j],current.aic))  
}  
}  
names(orders) <- c("p","q","AIC")  
# Extract the "best" one according to AIC  
orders[which.min(orders$AIC),]

## p q AIC  
## 39 4 6 1912.766

############# Intel Daily Data ##########################  
# Daily d=1  
n = length(INTDaily)   
#######   
n\_fit=length(INTDaily.train)  
n\_forward=n-n\_fit  
n=length(INTDaily.train)  
norder = 8  
orders = data.frame()  
p = c(0:norder); q = c(0:norder)  
aic = matrix(0,norder,norder)  
for(i in 1:norder){  
for(j in 1:norder){  
modij = stats::arima(INTDaily.train,order = c(p[i],1,q[j]), method='ML')  
current.aic=modij$aic-2\*(p[i]+q[j]+1)+2\*(p[i]+q[j]+1)\*n/(n-p[i]-q[j]-2)  
orders<-rbind(orders,c(p[i],q[j],current.aic))  
}  
}  
names(orders) <- c("p","q","AIC")  
# Extract the "best" one according to AIC  
orders[which.min(orders$AIC),]

## p q AIC  
## 39 4 6 5780.806

# Daily d=2  
n = length(INTDaily) ####### w\_n, deleted w\_  
n\_fit=length(INTDaily.train)  
norder = 8  
orders = data.frame()  
p = c(0:norder); q = c(0:norder)  
aic = matrix(0,norder,norder)  
for(i in 1:norder){  
for(j in 1:norder){  
modij = stats::arima(INTDaily.train,order = c(p[i],2,q[j]), method='ML')  
current.aic=modij$aic-2\*(p[i]+q[j]+1)+2\*(p[i]+q[j]+1)\*n/(n-p[i]-q[j]-2)  
orders<-rbind(orders,c(p[i],q[j],current.aic))  
}  
}  
names(orders) <- c("p","q","AIC")  
# Extract the "best" one according to AIC  
orders[which.min(orders$AIC),]

## p q AIC  
## 39 4 6 5798.185

#### Best model for Daily ####  
d\_final\_model=arima(INTDaily.train, order = c(4,1,6), method = "ML")  
d\_final\_model

##   
## Call:  
## arima(x = INTDaily.train, order = c(4, 1, 6), method = "ML")  
##   
## Coefficients:  
## ar1 ar2 ar3 ar4 ma1 ma2 ma3 ma4  
## -0.5362 0.3921 -0.5940 -0.9197 0.3635 -0.3951 0.7476 0.7697  
## s.e. 0.0242 0.0141 0.0115 0.0210 0.0318 0.0274 0.0262 0.0297  
## ma5 ma6  
## -0.0947 0.0663  
## s.e. 0.0227 0.0233  
##   
## sigma^2 estimated as 0.5787: log likelihood = -2879.35, aic = 5780.7

#### Best model for Weekly ####  
w\_final\_model=arima(INTWeekly.train, order = c(4,1,5), method = "ML")  
w\_final\_model

##   
## Call:  
## arima(x = INTWeekly.train, order = c(4, 1, 5), method = "ML")  
##   
## Coefficients:  
## ar1 ar2 ar3 ar4 ma1 ma2 ma3 ma4  
## -0.2414 -0.5190 -0.3792 -0.8162 0.2188 0.5079 0.3516 0.8688  
## s.e. 0.0682 0.0443 0.0452 0.0771 0.0807 0.0428 0.0426 0.0696  
## ma5  
## -0.1309  
## s.e. 0.0524  
##   
## sigma^2 estimated as 2.185: log likelihood = -943.64, aic = 1907.28

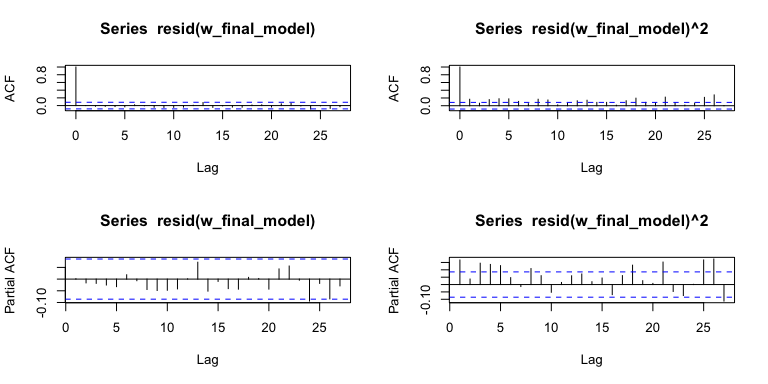
The selected models for the weekly data are as follows:

Daily Model Selection: ARIMA(4,1,6) with AICc = 5780.806 vs ARIMA(4,2,6) with AICc = 5798.185, where the less complex model ARIMA(4,1,6) model has a smaller AICc value. 10

Weekly Model Selection: ARIMA(4,1,5) with AICc = 1907.713 vs ARIMA(4,2,6) with AICc = 1912.766, with the ARIMA(4,1,5) having a lower AICc value hence selected.

**2b.** Evaluate the model residuals and squared residuals using the ACF and PACF plots as well as hypothesis testing for serial correlation. What would you conclude based on this analysis?

#plotting the acfs and pacfs of residuals and squared residuals for weekly Data  
  
par(mfrow=c(2,2))  
acf(resid(w\_final\_model))  
acf(resid(w\_final\_model)^2)  
pacf(resid(w\_final\_model))  
pacf(resid(w\_final\_model)^2)



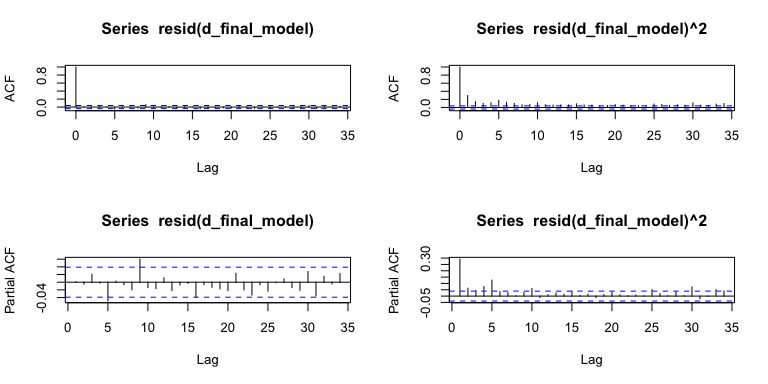
Box.test(w\_final\_model$resid, lag = (4+5+1), type = "Ljung-Box", fitdf = (4+5))

##   
## Box-Ljung test  
##   
## data: w\_final\_model$resid  
## X-squared = 4.6857, df = 1, p-value = 0.03041

Box.test(w\_final\_model$resid^2, lag = (4+5+1), type = "Ljung-Box", fitdf = (4+5))

##   
## Box-Ljung test  
##   
## data: w\_final\_model$resid^2  
## X-squared = 97.792, df = 1, p-value < 2.2e-16

#plotting the acfs and pacfs of residuals and squared residuals for daily Data  
  
par(mfrow=c(2,2))  
acf(resid(d\_final\_model))  
acf(resid(d\_final\_model)^2)  
pacf(resid(d\_final\_model))  
pacf(resid(d\_final\_model)^2)



Box.test(d\_final\_model$resid, lag = (4+6+1), type = "Ljung-Box", fitdf = (4+6))

##   
## Box-Ljung test  
##   
## data: d\_final\_model$resid  
## X-squared = 18.998, df = 1, p-value = 1.309e-05

Box.test(d\_final\_model$resid^2, lag = (4+6+1), type = "Ljung-Box", fitdf = (4+6))

##   
## Box-Ljung test  
##   
## data: d\_final\_model$resid^2  
## X-squared = 532.96, df = 1, p-value < 2.2e-16

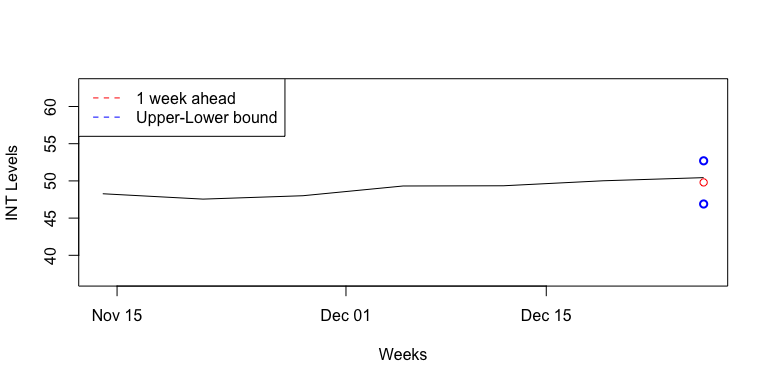
*Response: Question 2b*

The fit for the weekly residuals data seems to be well fitting based on the residual analysis. The null hypothesis is that the residual process consists of uncorrelated variables, which, with a 99% is not rejected since the the p-value is higher than 1%. There appears to be arch effect present based on a p-value that is lower than 0.05 for the square residuals.

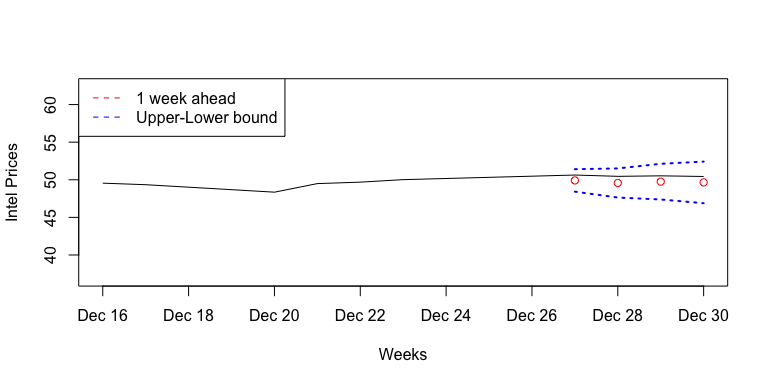
For the daily data, we fail to reject both correlated data for residulas and squared residuals.

**2c.** Apply the models identified in (2a) and forecast the last week of data. Plot the predicted data to compare the predicted values to the actual observed ones. Include 95% confidence intervals for the (mean) forecasts in the corresponding plots.

#predicting last week data using weekly data model  
outpred = predict(w\_final\_model,n.ahead=w\_n\_forward)  
# 95% confidence interval  
ubound = outpred$pred+1.96\*outpred$se  
lbound = outpred$pred-1.96\*outpred$se  
ymin = min(lbound)-10  
ymax = max(ubound)+10  
dates.diff = weeklydata$Date  
#nfit = n-n\_forward  
par(mfrow=c(1,1))  
n = length(INTWeekly)  
plot((dates.diff)[(n-w\_n\_forward-5):n],INTWeekly[(n-w\_n\_forward-5):n],type="l", ylim=c(ymin,ymax), xlab="Weeks", ylab="INT Levels")  
points((dates.diff)[(w\_n\_fit+1):n],outpred$pred,col="red")  
points((dates.diff)[(w\_n\_fit+1):n],ubound,lty=3,lwd= 2, col="blue")  
points((dates.diff)[(w\_n\_fit+1):n],lbound,lty=3,lwd= 2, col="blue")  
legend('topleft', legend=c("1 week ahead ","Upper-Lower bound"),lty = 2, col=c("red","blue"))



#predicting last week data using daily data model  
outpred = predict(d\_final\_model,n.ahead=n\_forward)  
# 95% confidence interval  
ubound = outpred$pred+1.96\*outpred$se  
lbound = outpred$pred-1.96\*outpred$se  
ymin = min(lbound)-10  
ymax = max(ubound)+10  
dates.diff = dailydata$Date  
#nfit = n-n\_forward  
par(mfrow=c(1,1))  
n = length(INTDaily)  
plot((dates.diff)[(n-n\_forward-5):n],INTDaily[(n-n\_forward-5):n],type="l", ylim=c(ymin,ymax), xlab="Weeks", ylab="Intel Prices")  
points((dates.diff)[(n\_fit+1):n],outpred$pred,col="red")  
lines((dates.diff)[(n\_fit+1):n],ubound,lty=3,lwd= 2, col="blue")  
lines((dates.diff)[(n\_fit+1):n],lbound,lty=3,lwd= 2, col="blue")  
legend('topleft', legend=c("1 week ahead ","Upper-Lower bound"),lty = 2, col=c("red","blue"))



The model for the daily data seems to fit the data better hence resulting in generally better predictions. The variability of the daily data helps the model predict values with a lower variance than predicting a single weekly value with a large probability of being away from the actual value. This is also due to the volatile nature of the stock price data.

**2d.** Calculate Mean Absolute Percentage Error (MAPE) and Precision Measure (PM) (PM only for daily data). How many observations are within the prediction bands? Compare the accuracy of the predictions for the daily and weekly time series using these two measures.

#Computing Accuracy measures  
#Weekly Data  
outpred = predict(w\_final\_model,n.ahead=w\_n\_forward)  
ubound = outpred$pred+1.96\*outpred$se #confidence interval  
lbound = outpred$pred-1.96\*outpred$se  
n = length(INTWeekly)  
INT\_true = as.vector(INTWeekly[(w\_n\_fit+1):n])  
INT\_pred = outpred$pred  
#MAPE  
mean(abs(INT\_pred-INT\_true)/INT\_true)

## [1] 0.01266207

#Daily Data  
outpred = predict(d\_final\_model,n.ahead=n\_forward)  
ubound = outpred$pred+1.96\*outpred$se #confidence interval  
lbound = outpred$pred-1.96\*outpred$se  
n = length(INTDaily)  
INT\_true = as.vector(INTDaily[(n\_fit+1):n])  
INT\_pred = outpred$pred  
#MAPE  
mean(abs(INT\_pred-INT\_true)/INT\_true)

## [1] 0.0156435

#PM  
sum((INT\_pred-INT\_true)^2)/sum((INT\_true-mean(INT\_true))^2)

## [1] 107.9916

*Response: Question 2d*

For the weekly data, we cannot compute the PM as there is only one value. But we see that by the MAPE, the weekly data fitting seems to be a better option.

# Question 3: ARMA(p,q)-GARCH(m,n) for Return Stock Price (20 Points)

**3a.** Divide the data into training and testing data set, where the training data exclude the last week of data (December 27th - December 30th) with the testing data including the last week of data. Apply the iterative model to fit an ARMA(p,q)-GARCH(m,n) model by selecting the orders for p & q up to 5 and orders for m & n up to 2. Display the summary of the final model fit. Write up the equation of the estimated model.

#dividing the data into training and testing  
INTWeekly.train=returnweeklydata[1:(length(returnweeklydata)-1)]  
INTWeekly.test=returnweeklydata[(length(returnweeklydata)):length(returnweeklydata)]  
INTDaily.train=returndailydata[1:(length(returndailydata)-4)]  
INTDaily.test=returndailydata[(length(returndailydata)-3):length(returndailydata)]  
  
#applying the ARIMA-GARCH model to the weekly data  
  
w.final.aic=Inf  
w.final.order=c(0,0,0)  
for (p in 0:5) for (d in 0:1) for (q in 0:5)  
{  
w.current.aic=AIC(arima(INTWeekly.train,order=c(p, d, q)))  
if(w.current.aic<w.final.aic)  
{  
w.final.aic=w.current.aic  
w.final.order=c(p,d,q)  
w.final.arima=arima(INTWeekly.train, order=w.final.order)  
}  
}  
# What is the selected order?  
w.final.order

## [1] 3 0 3

#Select model with smallest BIC  
final.bic = Inf  
final.order = c(0,0)  
for (p in 0:2) for (q in 0:2)  
{  
spec = ugarchspec(variance.model=list(garchOrder=c(p,q)),  
mean.model=list(armaOrder=c(w.final.order[1], w.final.order[3]), include.mean=T),  
distribution.model="std")  
fit = ugarchfit(spec, INTWeekly.train, solver = 'nloptr')  
current.bic = infocriteria(fit)[2]  
if (current.bic < final.bic)  
{  
final.bic = current.bic  
final.order = c(p, q)  
}  
}  
final.order

## [1] 1 1

#Refine the ARMA order  
final.bic = Inf  
final.order.arma = c(0,0)  
for (p in 0:5) for (q in 0:5)  
{  
spec = ugarchspec(variance.model=list(garchOrder=c(final.order[1],final.order[2])),  
mean.model=list(armaOrder=c(p, q), include.mean=T),  
distribution.model="std")  
fit = ugarchfit(spec, INTWeekly.train, solver = 'hybrid')  
current.bic = infocriteria(fit)[2]  
if (current.bic < final.bic)  
{  
final.bic = current.bic  
final.order.arma = c(p, q)  
}  
}  
final.order.arma

## [1] 0 0

#Refine the GARCH order  
final.bic = Inf  
final.order.garch = c(0,0)  
for (p in 0:2) for (q in 0:2)  
{  
spec = ugarchspec(variance.model=list(garchOrder=c(p,q)),  
mean.model=list(armaOrder=c(final.order.arma[1], final.order.arma[2]),  
include.mean=T), distribution.model="std")  
fit = ugarchfit(spec, INTWeekly.train, solver = 'hybrid')  
current.bic = infocriteria(fit)[2]  
if (current.bic < final.bic)  
{  
final.bic = current.bic  
final.order.garch = c(p, q)  
}  
}  
final.order.garch

## [1] 1 1

final.order.garch.weekly=final.order.garch  
#Goodness of Fit  
w.spec.1 = ugarchspec(variance.model=list(garchOrder=c(1,1)),  
mean.model=list(armaOrder=c(0,0),  
include.mean=T), distribution.model="std")  
w.final.model.1 = ugarchfit(w.spec.1, INTWeekly.train, solver = 'hybrid')  
w.final.model.1

##   
## \*---------------------------------\*  
## \* GARCH Model Fit \*  
## \*---------------------------------\*  
##   
## Conditional Variance Dynamics   
## -----------------------------------  
## GARCH Model : sGARCH(1,1)  
## Mean Model : ARFIMA(0,0,0)  
## Distribution : std   
##   
## Optimal Parameters  
## ------------------------------------  
## Estimate Std. Error t value Pr(>|t|)  
## mu 0.003340 0.001344 2.4855 0.012937  
## omega 0.000041 0.000026 1.5971 0.110234  
## alpha1 0.059608 0.022887 2.6044 0.009203  
## beta1 0.911402 0.032045 28.4414 0.000000  
## shape 5.388766 1.283756 4.1977 0.000027  
##   
## Robust Standard Errors:  
## Estimate Std. Error t value Pr(>|t|)  
## mu 0.003340 0.001294 2.5814 0.009840  
## omega 0.000041 0.000023 1.8071 0.070753  
## alpha1 0.059608 0.021233 2.8074 0.004995  
## beta1 0.911402 0.027729 32.8682 0.000000  
## shape 5.388766 1.375455 3.9178 0.000089  
##   
## LogLikelihood : 1027.174   
##   
## Information Criteria  
## ------------------------------------  
##   
## Akaike -3.9314  
## Bayes -3.8905  
## Shibata -3.9316  
## Hannan-Quinn -3.9154  
##   
## Weighted Ljung-Box Test on Standardized Residuals  
## ------------------------------------  
## statistic p-value  
## Lag[1] 0.02675 0.8701  
## Lag[2\*(p+q)+(p+q)-1][2] 0.13541 0.8955  
## Lag[4\*(p+q)+(p+q)-1][5] 2.04803 0.6071  
## d.o.f=0  
## H0 : No serial correlation  
##   
## Weighted Ljung-Box Test on Standardized Squared Residuals  
## ------------------------------------  
## statistic p-value  
## Lag[1] 0.01195 0.9129  
## Lag[2\*(p+q)+(p+q)-1][5] 0.96139 0.8684  
## Lag[4\*(p+q)+(p+q)-1][9] 1.77049 0.9306  
## d.o.f=2  
##   
## Weighted ARCH LM Tests  
## ------------------------------------  
## Statistic Shape Scale P-Value  
## ARCH Lag[3] 0.3617 0.500 2.000 0.5475  
## ARCH Lag[5] 0.9988 1.440 1.667 0.7333  
## ARCH Lag[7] 1.2499 2.315 1.543 0.8698  
##   
## Nyblom stability test  
## ------------------------------------  
## Joint Statistic: 1.2687  
## Individual Statistics:   
## mu 0.05013  
## omega 0.06782  
## alpha1 0.07477  
## beta1 0.07258  
## shape 0.54772  
##   
## Asymptotic Critical Values (10% 5% 1%)  
## Joint Statistic: 1.28 1.47 1.88  
## Individual Statistic: 0.35 0.47 0.75  
##   
## Sign Bias Test  
## ------------------------------------  
## t-value prob sig  
## Sign Bias 0.5795 0.5625   
## Negative Sign Bias 0.2168 0.8284   
## Positive Sign Bias 0.4285 0.6685   
## Joint Effect 0.3575 0.9489   
##   
##   
## Adjusted Pearson Goodness-of-Fit Test:  
## ------------------------------------  
## group statistic p-value(g-1)  
## 1 20 26.92 0.1065  
## 2 30 32.00 0.3199  
## 3 40 50.46 0.1034  
## 4 50 53.85 0.2942  
##   
##   
## Elapsed time : 0.07157993

#applying the ARIMA-GARCH model to the daily data  
  
d.final.aic=Inf  
d.final.order=c(0,0,0)  
for (p in 0:3) for (d in 0:1) for (q in 0:3)  
{  
d.current.aic=AIC(arima(INTDaily.train,order=c(p, d, q)))  
if(d.current.aic<d.final.aic)  
{  
d.final.aic=d.current.aic  
d.final.order=c(p,d,q)  
d.final.arima=arima(INTDaily.train, order=d.final.order)  
}  
}  
# What is the selected order?  
d.final.order

## [1] 3 0 1

#Select model with smallest BIC  
final.bic = Inf  
final.order = c(0,0)  
for (p in 0:2) for (q in 0:2)  
{  
spec = ugarchspec(variance.model=list(garchOrder=c(p,q)),  
mean.model=list(armaOrder=c(d.final.order[1], d.final.order[3]), include.mean=T),  
distribution.model="std")  
fit = ugarchfit(spec, INTDaily.train, solver = 'nloptr')  
current.bic = infocriteria(fit)[2]  
if (current.bic < final.bic)  
{  
final.bic = current.bic  
final.order = c(p, q)  
}  
}  
final.order

## [1] 1 1

#Refine the ARMA order  
final.bic = Inf  
final.order.arma = c(0,0)  
for (p in 0:3) for (q in 0:3)  
{  
spec = ugarchspec(variance.model=list(garchOrder=c(final.order[1],final.order[2])),  
mean.model=list(armaOrder=c(p, q), include.mean=T),  
distribution.model="std")  
fit = ugarchfit(spec, INTDaily.train, solver = 'hybrid')  
current.bic = infocriteria(fit)[2]  
if (current.bic < final.bic)  
{  
final.bic = current.bic  
final.order.arma = c(p, q)  
}  
}  
final.order.arma

## [1] 0 0

#Refine the GARCH order  
final.bic = Inf  
final.order.garch = c(0,0)  
for (p in 0:2) for (q in 0:2)  
{  
spec = ugarchspec(variance.model=list(garchOrder=c(p,q)),  
mean.model=list(armaOrder=c(final.order.arma[1], final.order.arma[2]),  
include.mean=T), distribution.model="std")  
fit = ugarchfit(spec, INTDaily.train, solver = 'hybrid')  
current.bic = infocriteria(fit)[2]  
if (current.bic < final.bic)  
{  
final.bic = current.bic  
final.order.garch = c(p, q)  
}  
}  
final.order.garch

## [1] 1 1

final.order.garch.daily=final.order.garch  
#Goodness of Fit  
d.spec.1 = ugarchspec(variance.model=list(garchOrder=c(1,1)),  
mean.model=list(armaOrder=c(0,0),  
include.mean=T), distribution.model="std")  
d.final.model.1 = ugarchfit(d.spec.1, INTDaily.train, solver = 'hybrid')  
d.final.model.1

##   
## \*---------------------------------\*  
## \* GARCH Model Fit \*  
## \*---------------------------------\*  
##   
## Conditional Variance Dynamics   
## -----------------------------------  
## GARCH Model : sGARCH(1,1)  
## Mean Model : ARFIMA(0,0,0)  
## Distribution : std   
##   
## Optimal Parameters  
## ------------------------------------  
## Estimate Std. Error t value Pr(>|t|)  
## mu 0.000864 0.000249 3.4668 0.000527  
## omega 0.000006 0.000004 1.3855 0.165895  
## alpha1 0.059339 0.005843 10.1558 0.000000  
## beta1 0.922901 0.005244 175.9943 0.000000  
## shape 3.874024 0.163613 23.6779 0.000000  
##   
## Robust Standard Errors:  
## Estimate Std. Error t value Pr(>|t|)  
## mu 0.000864 0.000246 3.51722 0.000436  
## omega 0.000006 0.000021 0.29439 0.768457  
## alpha1 0.059339 0.047657 1.24511 0.213092  
## beta1 0.922901 0.048211 19.14302 0.000000  
## shape 3.874024 1.206159 3.21187 0.001319  
##   
## LogLikelihood : 7002.76   
##   
## Information Criteria  
## ------------------------------------  
##   
## Akaike -5.5737  
## Bayes -5.5621  
## Shibata -5.5737  
## Hannan-Quinn -5.5695  
##   
## Weighted Ljung-Box Test on Standardized Residuals  
## ------------------------------------  
## statistic p-value  
## Lag[1] 2.667 0.1025  
## Lag[2\*(p+q)+(p+q)-1][2] 2.837 0.1553  
## Lag[4\*(p+q)+(p+q)-1][5] 4.635 0.1848  
## d.o.f=0  
## H0 : No serial correlation  
##   
## Weighted Ljung-Box Test on Standardized Squared Residuals  
## ------------------------------------  
## statistic p-value  
## Lag[1] 0.1352 0.7131  
## Lag[2\*(p+q)+(p+q)-1][5] 0.4856 0.9599  
## Lag[4\*(p+q)+(p+q)-1][9] 1.0995 0.9818  
## d.o.f=2  
##   
## Weighted ARCH LM Tests  
## ------------------------------------  
## Statistic Shape Scale P-Value  
## ARCH Lag[3] 0.1622 0.500 2.000 0.6872  
## ARCH Lag[5] 0.5298 1.440 1.667 0.8748  
## ARCH Lag[7] 0.9400 2.315 1.543 0.9231  
##   
## Nyblom stability test  
## ------------------------------------  
## Joint Statistic: 3.3028  
## Individual Statistics:   
## mu 0.06057  
## omega 0.24997  
## alpha1 0.26218  
## beta1 0.19870  
## shape 0.13391  
##   
## Asymptotic Critical Values (10% 5% 1%)  
## Joint Statistic: 1.28 1.47 1.88  
## Individual Statistic: 0.35 0.47 0.75  
##   
## Sign Bias Test  
## ------------------------------------  
## t-value prob sig  
## Sign Bias 0.02508 0.9800   
## Negative Sign Bias 1.44564 0.1484   
## Positive Sign Bias 1.20228 0.2294   
## Joint Effect 3.54104 0.3155   
##   
##   
## Adjusted Pearson Goodness-of-Fit Test:  
## ------------------------------------  
## group statistic p-value(g-1)  
## 1 20 13.60 0.8067  
## 2 30 19.72 0.9015  
## 3 40 37.83 0.5230  
## 4 50 39.92 0.8193  
##   
##   
## Elapsed time : 0.2150428

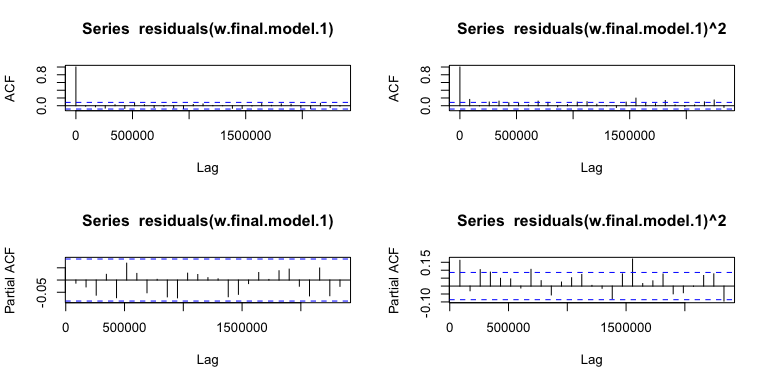
*Response: Question 3a* The equation for the daily data is: and

The equation for the weekly data is: and

*Please also consider as correct solution those who used ARMA order (0,1), instead of ARMA (0,0).*

**3b.** Evaluate the model residuals and squared residuals using the ACF and PACF plots as well as hypothesis testing for serial correlation. What would you conclude based on this analysis?

#plotting the acfs and pacfs of residuals and squared residuals  
par(mfrow=c(2,2))  
acf(residuals(w.final.model.1))  
acf(residuals(w.final.model.1)^2)  
pacf(residuals(w.final.model.1))  
pacf(residuals(w.final.model.1)^2)



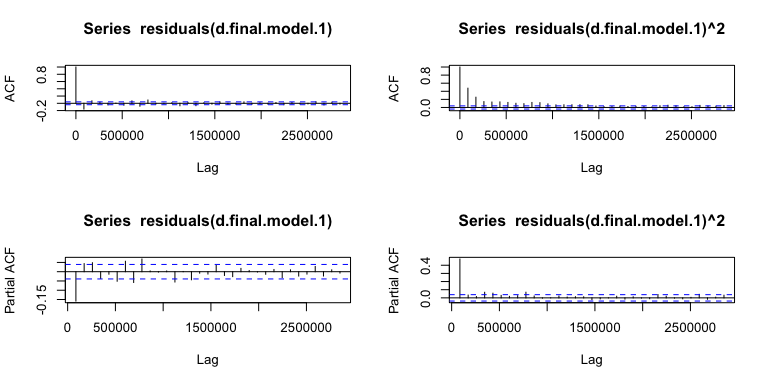
Box.test(residuals(w.final.model.1),lag=,type='Ljung',fitdf=)

##   
## Box-Ljung test  
##   
## data: residuals(w.final.model.1)  
## X-squared = 0.084368, df = 1, p-value = 0.7715

Box.test(residuals(w.final.model.1)^2,lag=,type='Ljung',fitdf=)

##   
## Box-Ljung test  
##   
## data: residuals(w.final.model.1)^2  
## X-squared = 13.773, df = 1, p-value = 0.0002063

par(mfrow=c(2,2))  
acf(residuals(d.final.model.1))  
acf(residuals(d.final.model.1)^2)  
pacf(residuals(d.final.model.1))  
pacf(residuals(d.final.model.1)^2)



Box.test(residuals(d.final.model.1),lag=,type='Ljung',fitdf=)

##   
## Box-Ljung test  
##   
## data: residuals(d.final.model.1)  
## X-squared = 63, df = 1, p-value = 2.109e-15

Box.test(residuals(d.final.model.1)^2,lag=,type='Ljung',fitdf=)

##   
## Box-Ljung test  
##   
## data: residuals(d.final.model.1)^2  
## X-squared = 570.36, df = 1, p-value < 2.2e-16

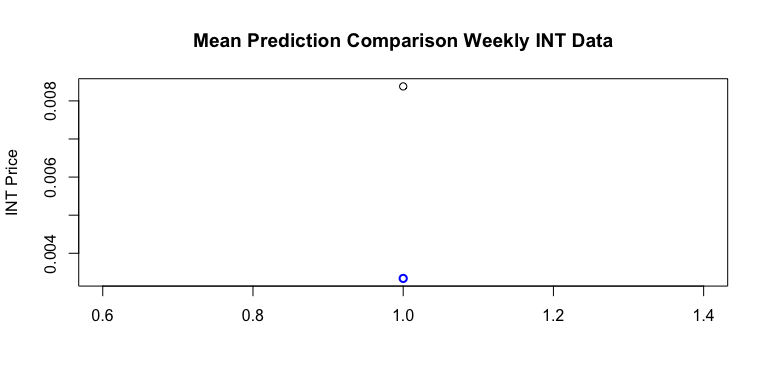
*Response: Question 3b*

From the residuals’ plots for the model weekly, we can see that there is no serial correlation, also confirmed by the p\_value of the Box test, as it is large enough to fail to reject the null hypothesis. Regarding the plots of the squared residuals, we can see that the Arch effect was somewhat controlled by the model, still showing a cyclical pattern. Based on the Box test we reject the null hypothesis that there is no serial correlation.

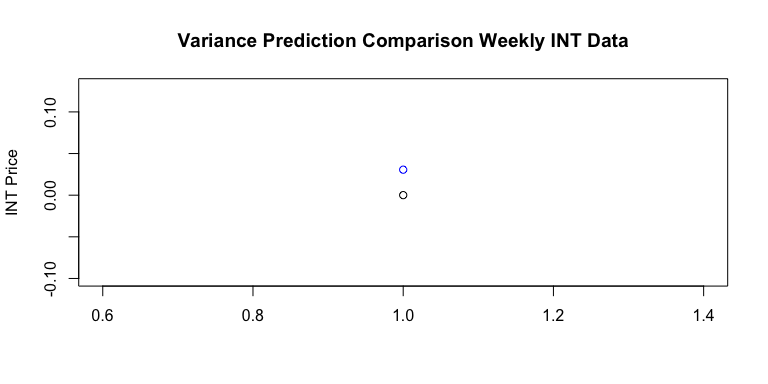
From the residuals’ plots for the daily model, we can see that there is some serial correlation, also confirmed by the p\_value of the Box test. Regarding the plots of the squared residuals, we can see that the Arch effect was somewhat controlled by the model, still showing a cyclical pattern. Based on the Box test we reject the null hypothesis that there is no serial correlation.

**3c.** Apply the model identified in (3a) and forecast the mean and the variance of the last week of data. Plot the predicted data to compare the predicted values to the actual observed ones. Interpret the results, particularly comparing forecast using daily versus weekly data.

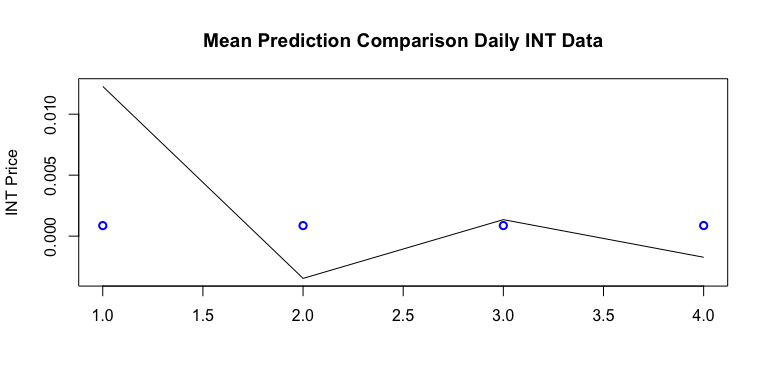
#predicting last week data using weekly data model  
#Prediction of the return time series and the volatility sigma  
wnfore = length(INTWeekly.test)  
w.fore.series.1 = NULL  
w.fore.sigma.1 = NULL  
  
for(f in 1: wnfore)  
{  
#Fit models  
wdata = INTWeekly.train  
if(f>2)  
wdata = c(INTWeekly.train,INTWeekly.test[1:(f-1)])  
w.final.model.1 = ugarchfit(w.spec.1, wdata, solver = 'hybrid')  
  
#Forecast  
  
wfore = ugarchforecast(w.final.model.1, n.ahead=1)  
w.fore.series.1 = c(w.fore.series.1, wfore@forecast$seriesFor)  
w.fore.sigma.1 = c(w.fore.sigma.1, wfore@forecast$sigmaFor)  
  
}  
w.fore.series.1[is.nan(w.fore.series.1)]=0  
#predicting last week data using daily data model  
#Prediction of the return time series and the volatility sigma  
dnfore = length(INTDaily.test)  
d.fore.series.1 = NULL  
d.fore.sigma.1 = NULL  
  
for(f in 1: dnfore)  
{  
#Fit models  
ddata = INTDaily.train  
if(f>2)  
ddata = c(INTDaily.train,INTDaily.test[1:(f-1)])  
d.final.model.1 = ugarchfit(d.spec.1, ddata, solver = 'hybrid')  
  
#Forecast  
dfore = ugarchforecast(d.final.model.1, n.ahead=1)  
d.fore.series.1 = c(d.fore.series.1, dfore@forecast$seriesFor)  
d.fore.sigma.1 = c(d.fore.sigma.1, dfore@forecast$sigmaFor)  
  
}  
d.fore.series.1[is.nan(d.fore.series.1)]=0  
  
#plotting forecasts and comparing with train data  
#Weekly  
#Mean Prediction Comparison Plot  
n=length(INTWeekly)  
ymin = min(c(as.vector(INTWeekly.test),w.fore.series.1))  
ymax = max(c(as.vector(INTWeekly.test),w.fore.series.1))  
data.plot = INTWeekly.test  
names(data.plot)="Fore"  
plot(INTWeekly.test,type="o", ylim=c(ymin,ymax), xlab=" ",  
ylab="INT Price",main="Mean Prediction Comparison Weekly INT Data")  
data.plot$Fore=w.fore.series.1  
points(w.fore.series.1,lwd= 2, col="blue")



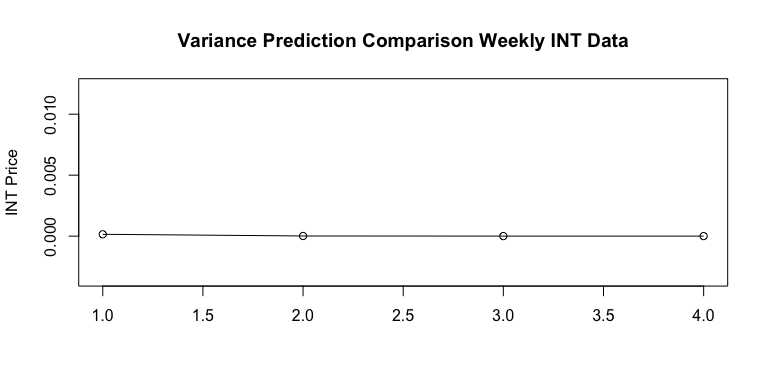
plot(INTWeekly.test^2,type="o", xlab=" ", ylim=c(INTWeekly.test^2-0.1,w.fore.sigma.1+0.1),  
ylab="INT Price",main="Variance Prediction Comparison Weekly INT Data")  
data.plot$Fore=w.fore.sigma.1  
points(w.fore.sigma.1, col="blue")



#Daily  
#Mean Prediction Comparison Plot  
n=length(INTDaily)  
ymin = min(c(as.vector(INTDaily.test),d.fore.series.1))  
ymax = max(c(as.vector(INTDaily.test),d.fore.series.1))  
data.plot = INTDaily.test  
names(data.plot)="Fore"  
plot(INTDaily.test,type="l", ylim=c(ymin,ymax), xlab=" ",  
ylab="INT Price",main="Mean Prediction Comparison Daily INT Data")  
points(d.fore.series.1,lwd= 2, col="blue")



plot(INTDaily.test^2,type="o", ylim=c(ymin,ymax), xlab=" ",  
ylab="INT Price",main="Variance Prediction Comparison Weekly INT Data")  
data.plot$Fore=d.fore.sigma.1  
points(d.fore.sigma.1,lwd= 2, col="blue")



*Response: Question 3c*

Graphically, we can see that the model using daily returns, better predict both, mean and variance.

**3d.** Calculate Mean Absolute Percentage Error (MAPE) and Precision Measure (PM) for the mean forecasts (PM should net be calculated for weekly data). Compare the accuracy of the predictions for the daily and weekly time series using these two measures.

#Compute Accuracy Measures for weekly data  
#Mean Absolute Percentage Error (MAPE)  
mean(abs(w.fore.series.1 - INTWeekly.test)/abs(INTWeekly.test+0.000001))

## [1] 0.6013278

#Compute Accuracy Measures for daily data  
#Mean Absolute Percentage Error (MAPE)  
mean(abs(d.fore.series.1 - INTDaily.test)/abs(INTDaily.test+0.000001))

## [1] 1.009619

#Precision Measure (PM)  
sum((d.fore.series.1 - INTDaily.test)^2)/sum((INTDaily.test-mean(INTDaily.test))^2)

## [1] 1.041334

*Response: Question 3c*

We can see that the MAPE for weekly data is lower, indicating that the mean is better predicted with this data.

# Question 4: Reflection on the Modeling and Forecasting (10 points)

Based on the analysis above, discuss the application of ARIMA on the stock price versus the application of ARMA-GARCH on the stock return. How do the models fit the data? How well do the models predict? How do the models perform when using daily versus weekly data? Would you use one approach over another for different settings? What are some specific points of caution one would need to consider when applying those models?

*Response: Question 4*

Based on the analysis above, discuss the application of ARIMA on the stock price versus the application of ARMA-GARCH on the stock return. How do the model fit the data? How well do the model predict? How do the models perform when using daily versus weekly data? Would you use one approach over another for different settings? What are some specific points of caution one would need to consider when applying those models?

For a stock like INT, where the volatility is quite high and the variance can have sudden jumps in the local means, it would be difficult to forecast values with a high precision. The models we have implemented also give us a similar experience. However, it seems better to use daily data in this case for the simple reason of capturing more fluctuations rather than less. This adds a numerical predictability to the forecast data and also reduces some variance we see in a single forecast value (in case of using weekly data).

It would be advisable to prefer ARMA-GARCH as a tool over ARIMA modeling in cases where the data fluctuations are a rare event, or can be captured in less accurate forecasting. It is also a generally better idea to use return data for better precision and reducing trend and stationarity issues.

First and foremost, we need to be careful with the data cleaning and allocation with training and testing data that is polished. We need to ensure uniformity in order to compare models. Precision measure calculations are not wary of the inputs given to them, hence, it can become quite easy to miscompare numbers. Finally, the theoretical attributions that underly numerical analysis might not be the case. Causality is an important consideration while applying models that can give us results contrary to the actual scenarios.