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("SBUX-2.csv", head = TRUE)
weeklydata <- read.csv("SBUX-Weekly-2.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
SBUXWeekly <- ts(weeklydata$Adj.Close,start=c(2012,1,1),freq=52)
SBUXDaily <- ts(dailydata$Adj.Close,start=c(2012,1,1),freq=252)
```

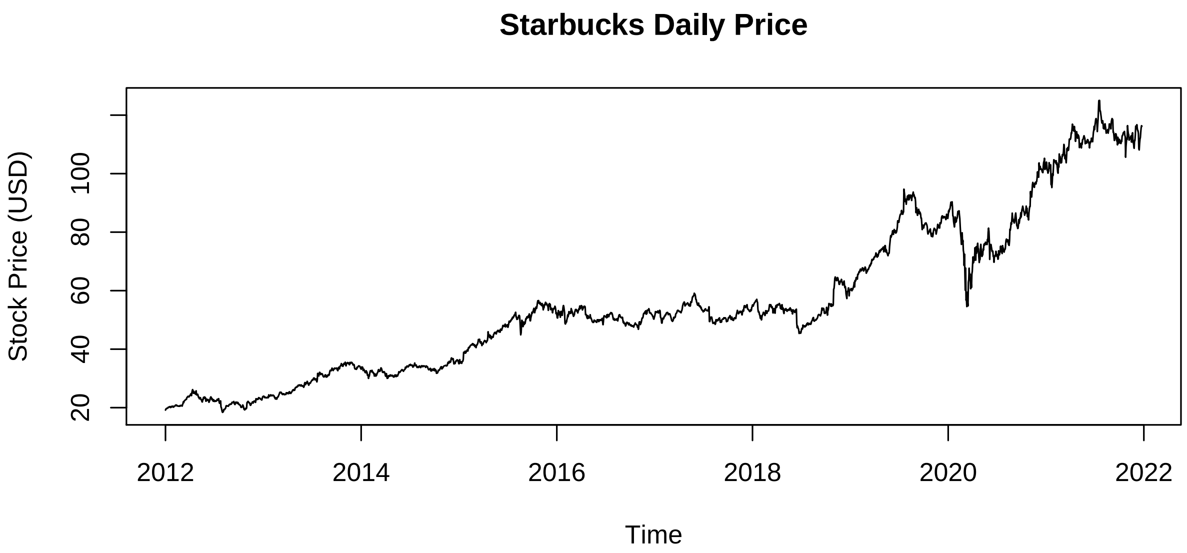
#Question 1: Exploratory Data Analysis (20 points)

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

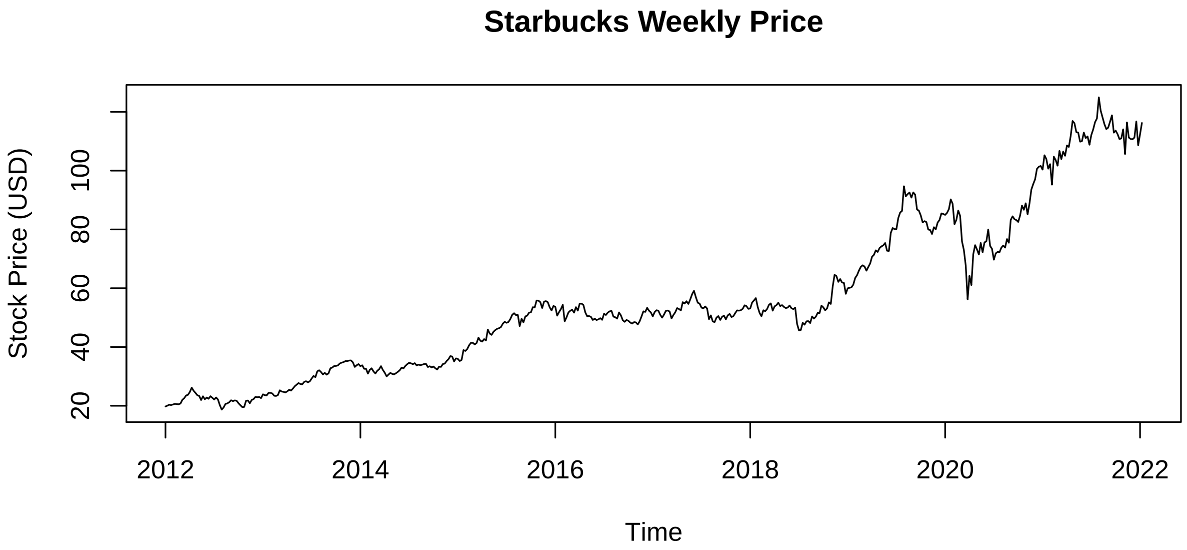
Response: Question 1a Daily price data would be more useful for swing traders, who plan to make a return after a few days. However, that data is more volatile than weekly pricing data. Weekly would be better for understanding the general trend.

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

```
plot(SBUXDaily, main="Starbucks Daily Price", ylab="Stock Price (USD)")
```



```
plot(SBUXWeekly, main="Starbucks Weekly Price", ylab="Stock Price (USD)")
```

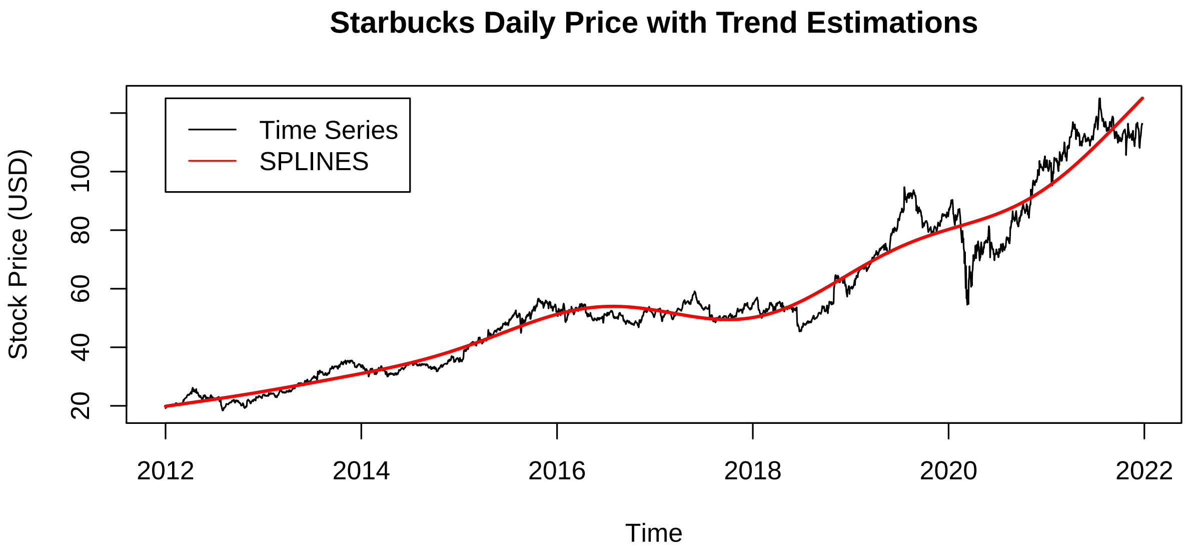


Response: Question 1b The overall trend is the same, but the daily data shows more volatility. This can be seen from the additional "spikiness" in the daily plot.

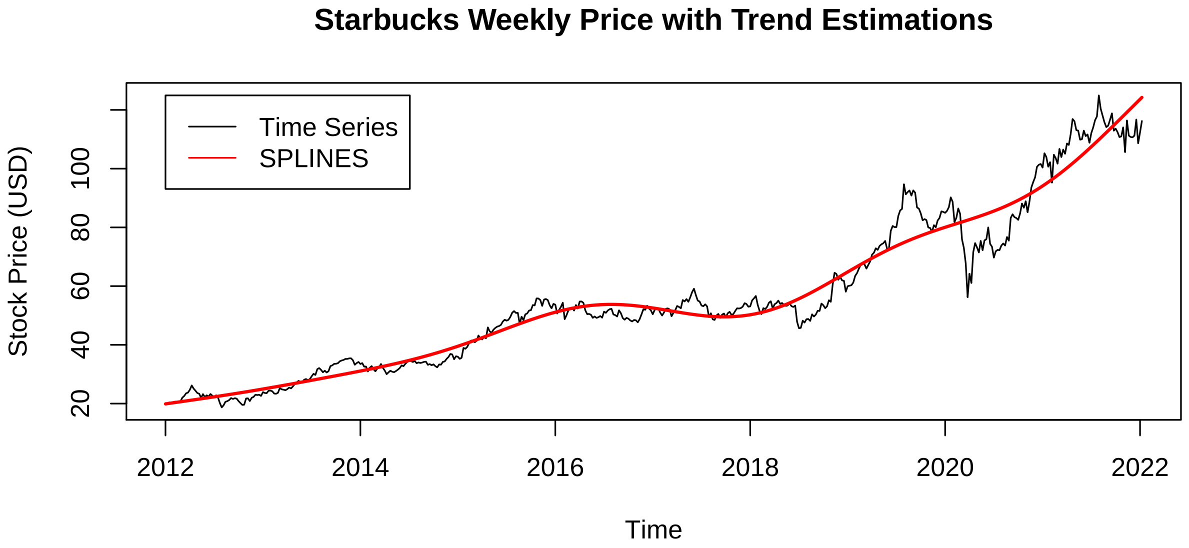
1c. Fit a non-parametric trend using splines regression to both the daily and weekly time-series data. Overlay the fitted trends. How do the trends compare?

```
daily.time.pts = c(1:length(SBUXDaily))
daily.time.pts = c(daily.time.pts - min(daily.time.pts))/max(daily.time.pts)
weekly.time.pts = c(1:length(SBUXWeekly))
weekly.time.pts = c(weekly.time.pts - min(weekly.time.pts))/max(weekly.time.pts)
```

```
spl.d <- gam(SBUXDaily ~ s(daily.time.pts))
tsspl.d <- ts(fitted(spl.d), start=c(2012,1,1), freq=252)
spl.w <- gam(SBUXWeekly ~ s(weekly.time.pts))
tsspl.w <- ts(fitted(spl.w), start=c(2012,1,1), freq=52)

ts.plot(SBUXDaily, ylab="Stock Price (USD)", main="Starbucks Daily Price with Trend Estimations")
lines(tsspl.d, lwd=2,col="red")
legend(x=2012,y=max(SBUXDaily),legend=c("Time Series", "SPLINES"),
       lty = 1, col=c("black", "red"))
```



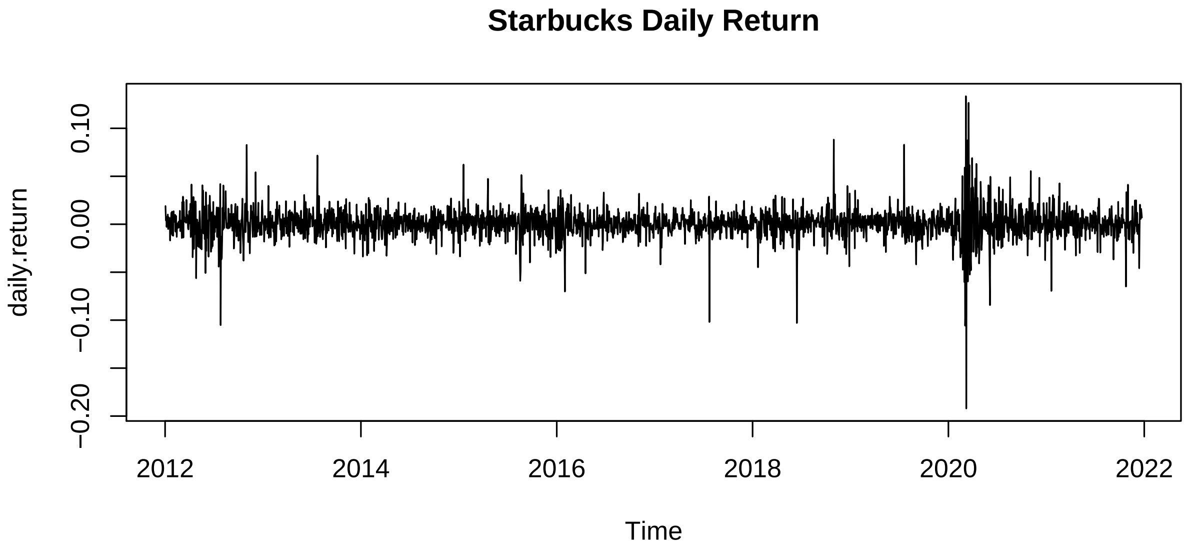
```
ts.plot(SBUXWeekly, ylab="Stock Price (USD)", main="Starbucks Weekly Price with Trend Estimations")
lines(tsspl.w, lwd=2,col="red")
legend(x=2012,y=max(SBUXWeekly),legend=c("Time Series", "SPLINES"),
       lty = 1, col=c("black", "red"))
```



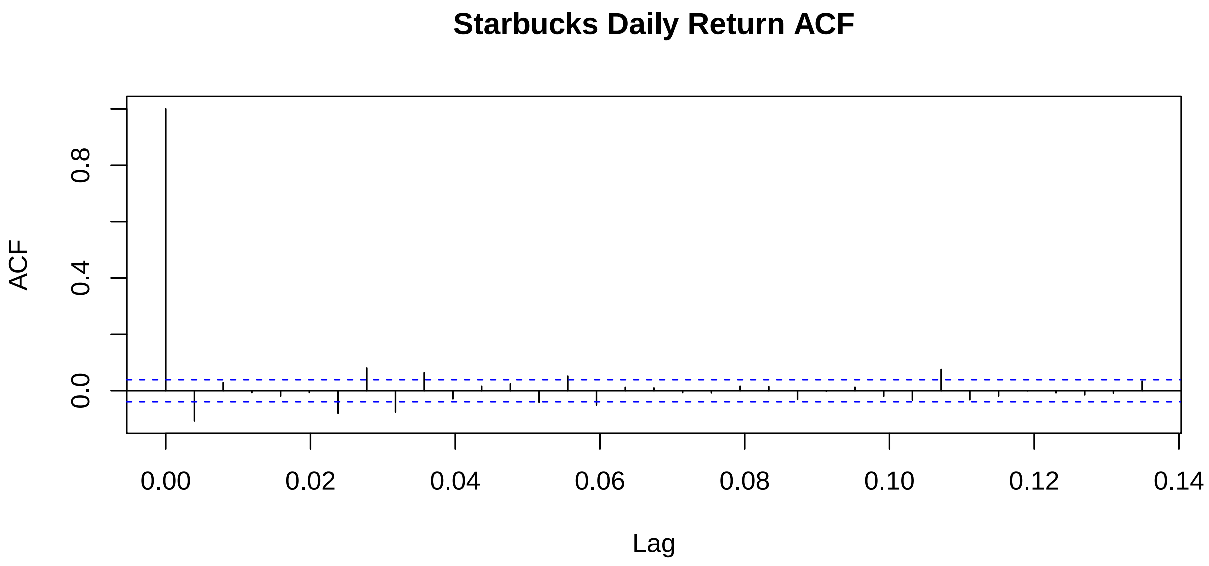
Response: Question 1c The daily vs weekly trends are similar, which isn't surprising given it's the same stock data at different intervals. The splines closely follow the time series. It doesn't fit perfectly.

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?

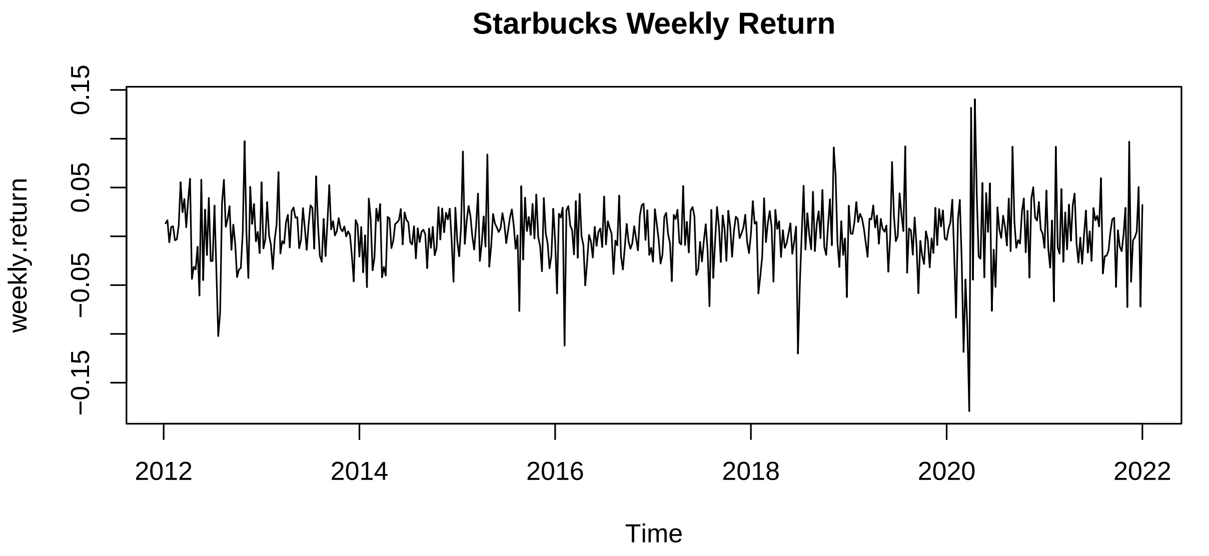
```
# Daily Return and ACF
daily.return <- diff(SBUXDaily, lag=1) / lag(SBUXDaily, 1)
plot(daily.return, main="Starbucks Daily Return")
```



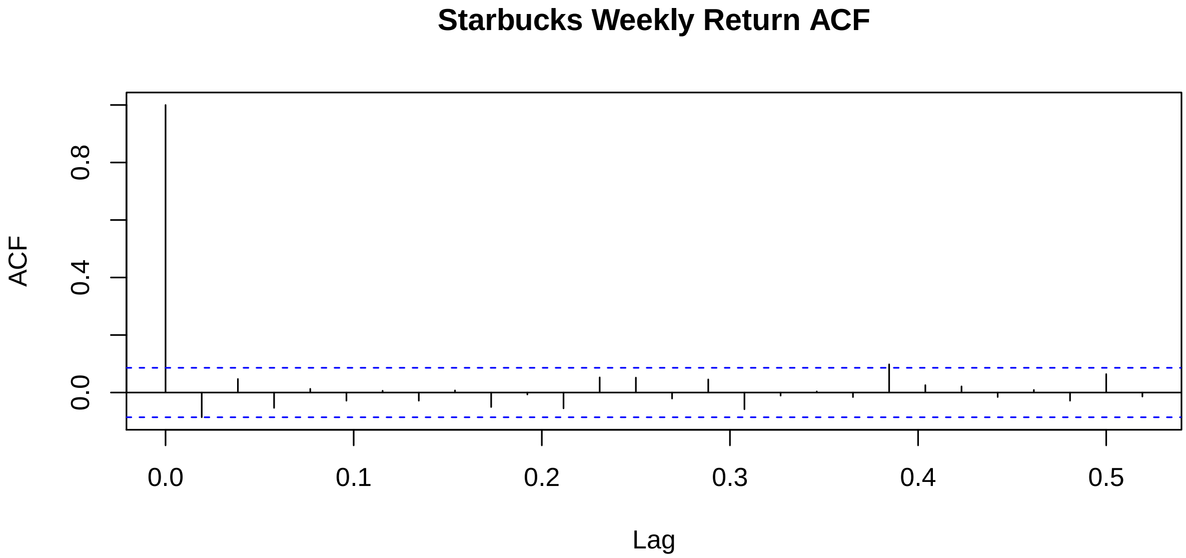
```
acf(daily.return, main="Starbucks Daily Return ACF")
```



```
# Weekly Return and ACF
weekly.return <- diff(SBUXWeekly, lag=1) / lag(SBUXWeekly, 1)
plot(weekly.return, main="Starbucks Weekly Return")
```



```
acf(weekly.return, main="Starbucks Weekly Return ACF")
```



Response: Question 1d The ACF plots indicate both time series don't have serial correlation. However, both series have non constant variance.

#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.

```r
# Daily Fits
num = 4

train.d <- ts(SBUXDaily[1:(length(SBUXDaily)-num)], freq=252)
test.d <- ts(SBUXDaily[(length(SBUXDaily)-num+1):length(SBUXDaily)],freq=252)

test_model.d <- function(p,d,q){
  mod = arima(train.d, order=c(p,d,q), method="ML")
  current.aic = AIC(mod)
  df = data.frame(p,d,q,current.aic)
  names(df) <- c("p","d","q","AIC")
  return(df)
}

# Diff of 1
orders.1d = data.frame(Inf,Inf,Inf,Inf)
names(orders.1d) <- c("p","d","q","AIC")

for (p in 0:8){
  for (q in 0:8) {
    possibleError <- tryCatch(
      orders.1d <- rbind(orders.1d,test_model.d(p,1,q)),
      error=function(e) e
    )
    if(inherits(possibleError, "error")) next

  }
}

orders.1d <- orders.1d[order(-orders.1d$AIC),]
print(tail(orders.1d,4))
```

```
##    p d q      AIC
## 61 6 1 5 6960.737
## 63 6 1 7 6960.365
## 70 7 1 5 6960.057
## 54 5 1 7 6958.434
```

```r
# Diff of 2
orders.2d = data.frame(Inf,Inf,Inf,Inf)
names(orders.2d) <- c("p","d","q","AIC")

for (p in 0:8){
  for (q in 0:8) {
    possibleError <- tryCatch(
      orders.2d <- rbind(orders.2d,test_model.d(p,2,q)),
      error=function(e) e
    )
    if(inherits(possibleError, "error")) next

  }
}

orders.2d <- orders.2d[order(-orders.2d$AIC),]
print(tail(orders.2d,4))
```

```
##    p d q      AIC
## 27 2 2 7 6964.297
## 77 8 2 3 6964.159
## 71 7 2 6 6963.677
## 62 6 2 6 6963.545
```

```r
# Weekly Fits
num = 1

train.w <- ts(SBUXWeekly[1:(length(SBUXWeekly)-num)], freq=52)
test.w <- ts(SBUXWeekly[(length(SBUXWeekly)-num+1):length(SBUXWeekly)],freq=52)

test_model.w <- function(p,d,q){
  mod = arima(train.w, order=c(p,d,q), method="ML")
  current.aic = AIC(mod)
  df = data.frame(p,d,q,current.aic)
  names(df) <- c("p","d","q","AIC")
  return(df)
}
```

```
# Diff of 1
orders.1w = data.frame(Inf,Inf,Inf,Inf)
names(orders.1w) <- c("p","d","q","AIC")

for (p in 0:8){
  for (q in 0:8) {
    possibleError <- tryCatch(
      orders.1w <- rbind(orders.1w,test_model.w(p,1,q)),
      error=function(e) e
    )
    if(inherits(possibleError, "error")) next

  }
}

orders.1w <- orders.1w[order(-orders.1w$AIC),]
print(tail(orders.1w,4))
```

```
##    p d q      AIC
## 53 5 1 6 2243.545
## 45 4 1 7 2243.005
## 81 8 1 7 2240.135
## 44 4 1 6 2237.453
```

```
# Diff of 2
orders.2w = data.frame(Inf,Inf,Inf,Inf)
names(orders.2w) <- c("p","d","q","AIC")

for (p in 0:8){
  for (q in 0:8) {
    possibleError <- tryCatch(
      orders.2w <- rbind(orders.2w,test_model.w(p,2,q)),
      error=function(e) e
    )
    if(inherits(possibleError, "error")) next

  }
}

orders.2w <- orders.2w[order(-orders.2w$AIC),]
print(tail(orders.2w,4))
```

```
##    p d q      AIC
## 44 4 2 6 2243.229
## 52 5 2 5 2243.151
## 61 6 2 5 2242.037
## 53 5 2 6 2240.607
```

```
# Final Models
modarima.d <- arima(train.d ,order=c(5,1,7), method="ML")
modarima.w <- arima(train.w ,order=c(4,1,6), method="ML")

print(modarima.d)
```

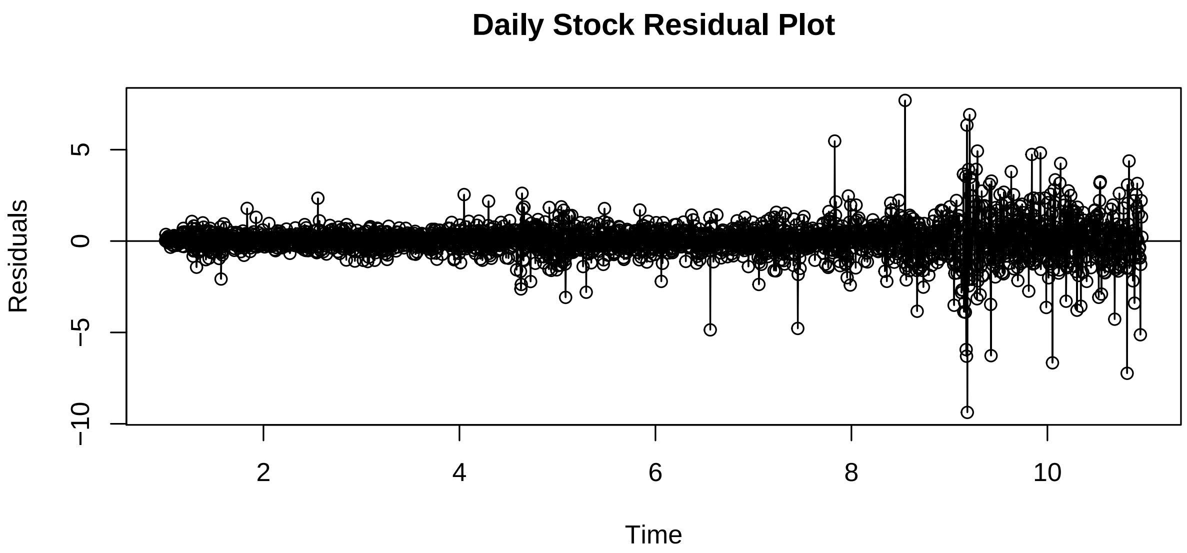
```
##
## Call:
## arima(x = train.d, order = c(5, 1, 7), method = "ML")
##
## Coefficients:
##           ar1     ar2     ar3      ar4      ar5     ma1      ma2      ma3
##       -0.4159  0.8746  0.6157  -0.5005  -0.3097  0.3232  -0.8793  -0.5326
## s.e.   0.1408  0.1380  0.1389   0.1022   0.1126  0.1403   0.1320   0.1370
##          ma4     ma5      ma6     ma7
##       0.4775  0.1830  -0.0319  0.1287
## s.e.  0.1056  0.1169   0.0262  0.0262
##
## sigma^2 estimated as 0.9257:  log likelihood = -3466.22,  aic = 6958.43
```

```
print(modarima.w)
```

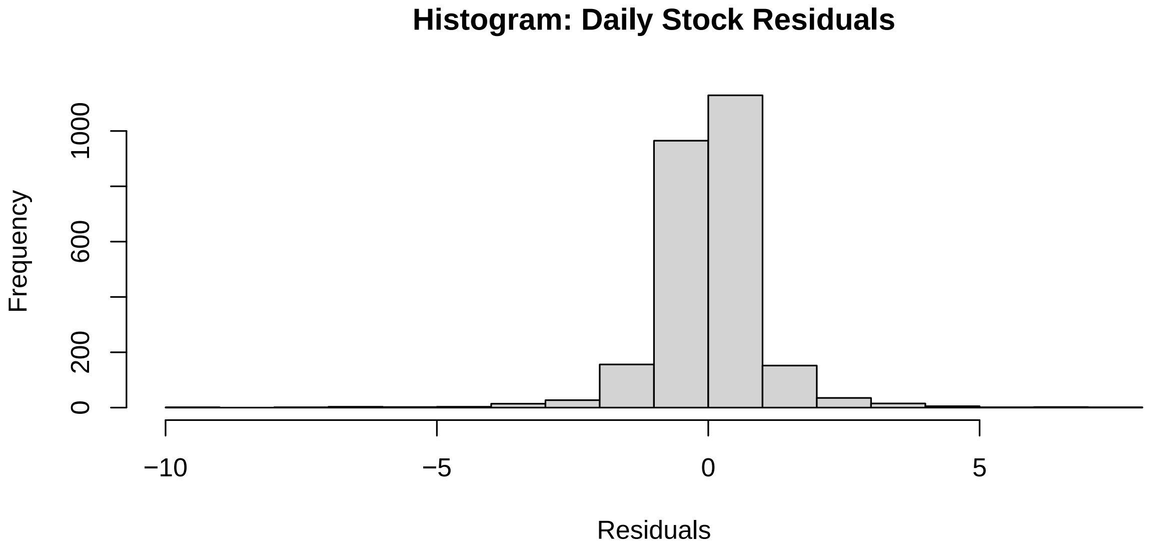
```
##
## Call:
## arima(x = train.w, order = c(4, 1, 6), method = "ML")
##
## Coefficients:
##           ar1     ar2      ar3      ar4     ma1     ma2     ma3     ma4
##       -0.1906  0.0322  -0.1706  -0.9505  0.0015  0.0563  0.1510  0.9574
## s.e.   0.0585  0.0620   0.0398   0.0225  0.0754  0.0950  0.0822  0.0537
##           ma5     ma6
##       -0.0836  0.1060
## s.e.   0.0667  0.0534
##
## sigma^2 estimated as 4.094:  log likelihood = -1107.73,  aic = 2237.45
```

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?

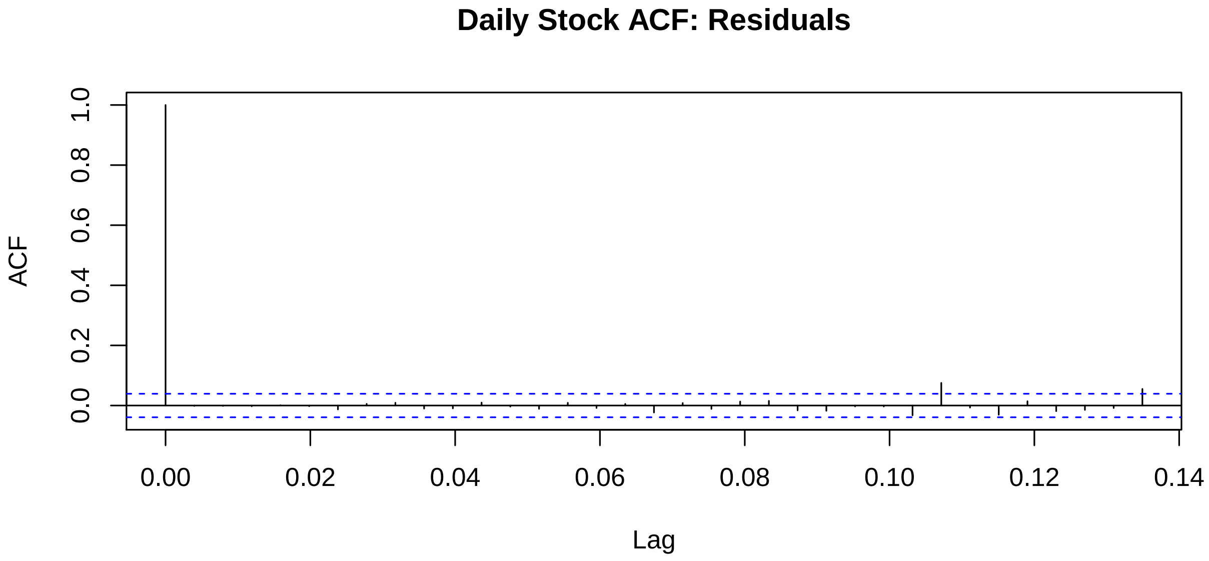
```
# Daily Residuals
res.arima.d <- resid(modarima.d)

plot(res.arima.d,
ylab = 'Residuals',
type = 'o',
main = "Daily Stock Residual Plot")
abline(h = 0)
```



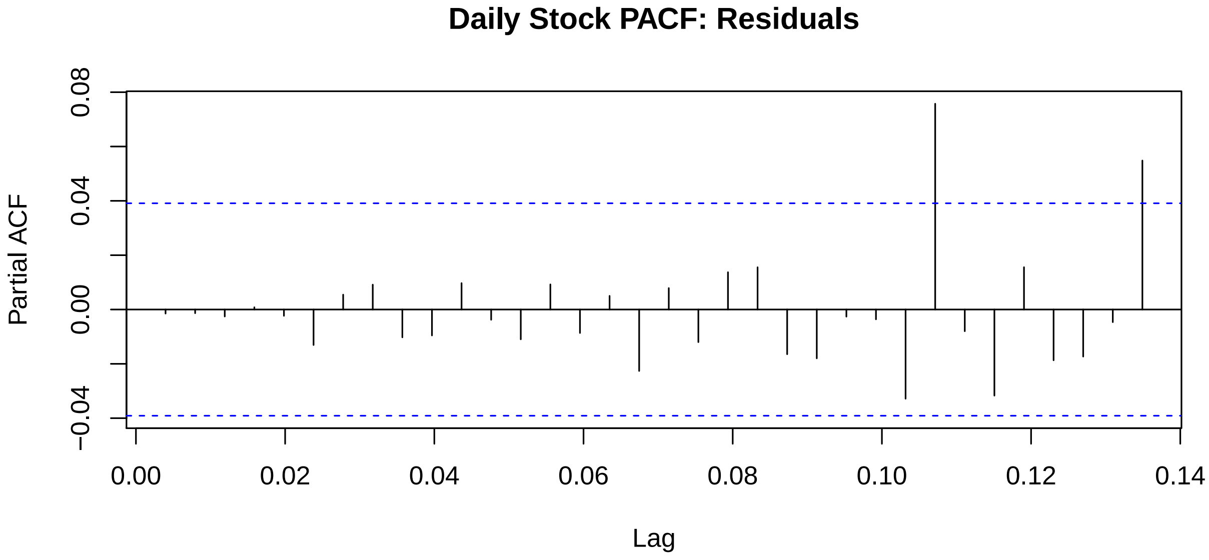
```r
hist(res.arima.d, xlab = 'Residuals', main = 'Histogram: Daily Stock Residuals')
```



```r
acf(res.arima.d, main = "Daily Stock ACF: Residuals")
```



```r
pacf(res.arima.d, main = "Daily Stock PACF: Residuals")
```



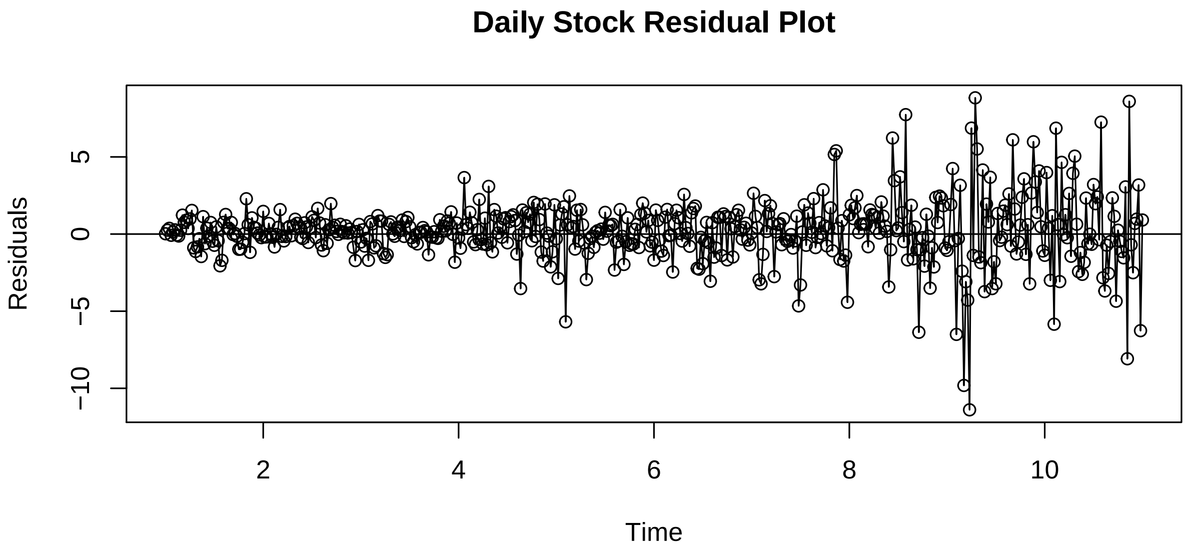
```r
Box.test(modarima.d$resid, lag = 3, type = "Box-Pierce", fitdf = 2)
```

```
##
##  Box-Pierce test
##
## data:  modarima.d$resid
## X-squared = 0.026988, df = 1, p-value = 0.8695
```

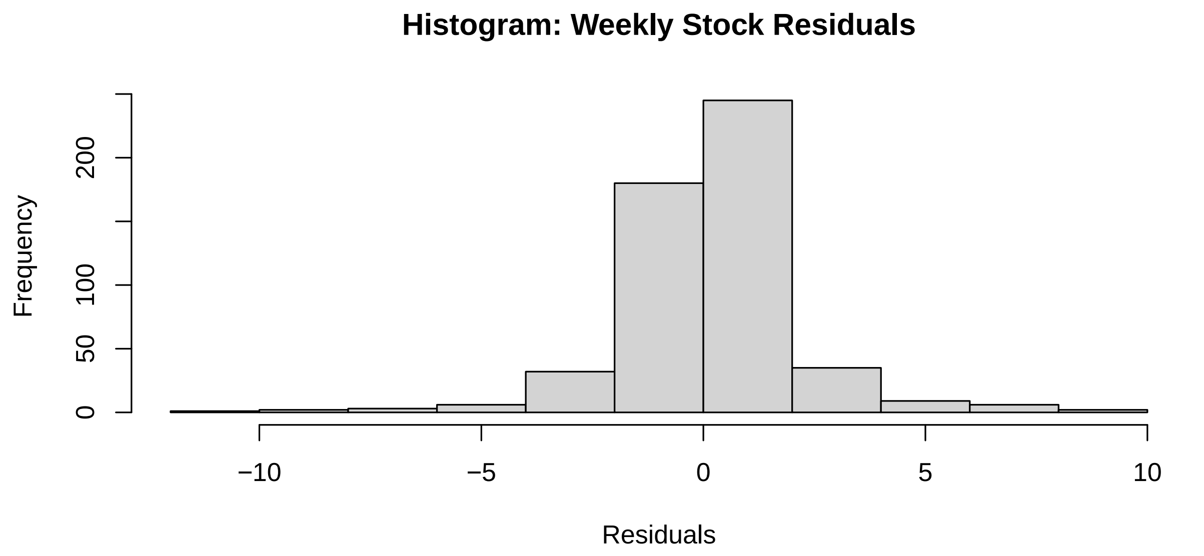
```
Box.test(modarima.d$resid, lag = 3, type = "Ljung-Box", fitdf = 2)
```

```
##
##  Box-Ljung test
##
## data:  modarima.d$resid
## X-squared = 0.027036, df = 1, p-value = 0.8694
```

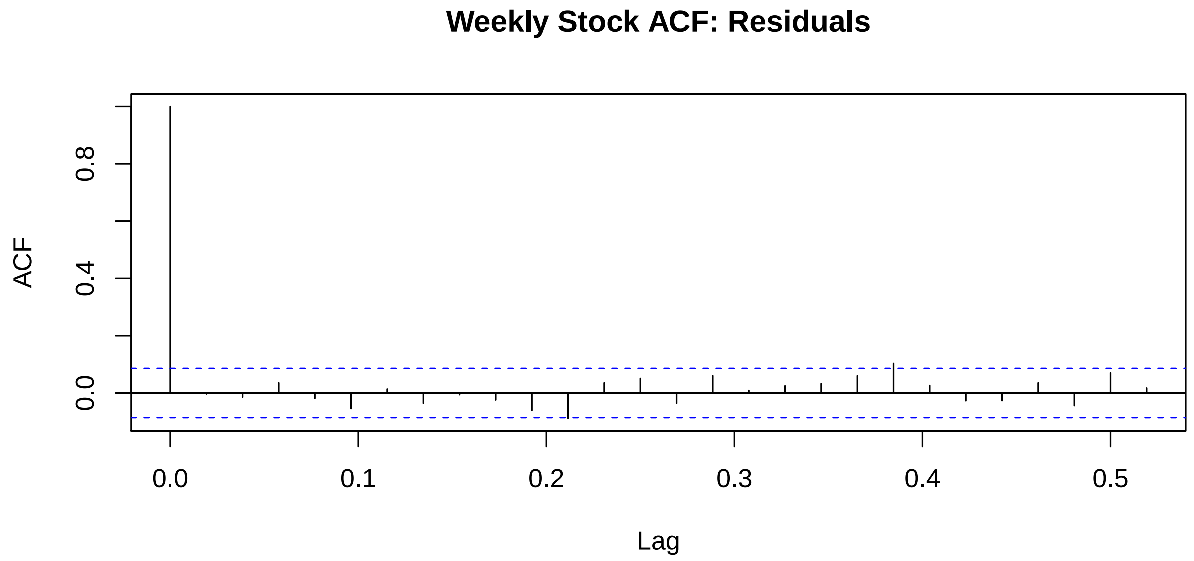
```
# Weekly Residuals
res.arima.w <- resid(modarima.w)

plot(res.arima.w,
ylab = 'Residuals',
type = 'o',
main = "Daily Stock Residual Plot")
abline(h = 0)
```



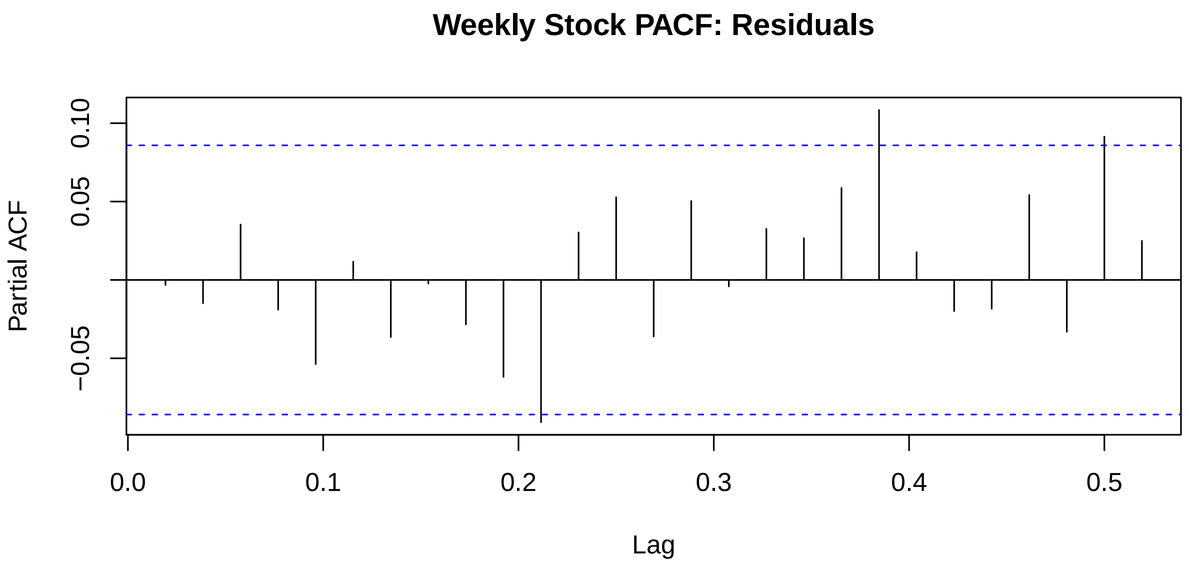
```
hist(res.arima.w, xlab = 'Residuals', main = 'Histogram: Weekly Stock Residuals')
```



```r
acf(res.arima.w, main = "Weekly Stock ACF: Residuals")
```



```r
pacf(res.arima.w, main = "Weekly Stock PACF: Residuals")
```



```
Box.test(modarima.w$resid, lag = 3, type = "Box-Pierce", fitdf = 2)
```

```
##
##  Box-Pierce test
##
## data:  modarima.w$resid
## X-squared = 0.77384, df = 1, p-value = 0.379
```

```
Box.test(modarima.w$resid, lag = 3, type = "Ljung-Box", fitdf = 2)
```

```
##
##  Box-Ljung test
##
## data:  modarima.w$resid
## X-squared = 0.78107, df = 1, p-value = 0.3768
```

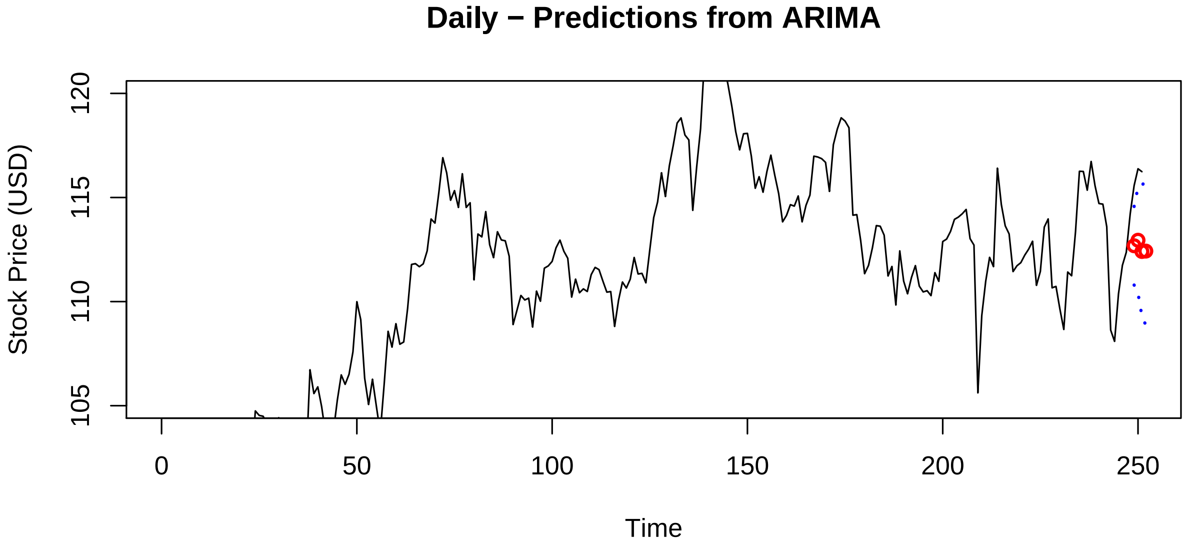
Response: Question 2b The residual ACF plots don't show autocorrelation.

Serial correlation testing results: The null hypothesis is that the residual process consists of uncorrelated variables. This is unsupported at the 90% confidence level for both series since the p-values are greater than 0.1.

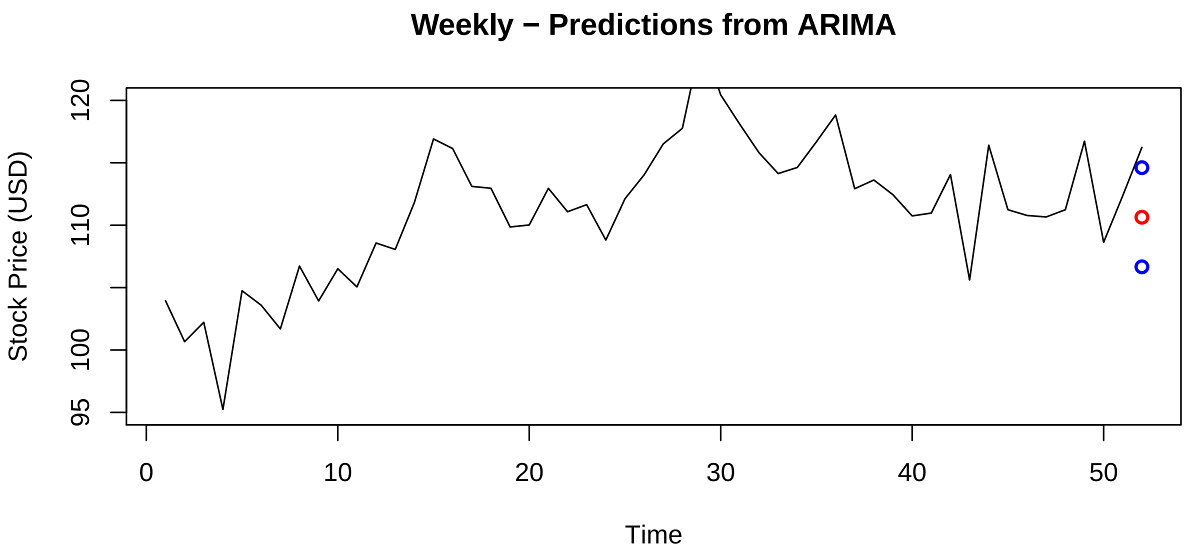
2c. Apply the model identified in (2a) and forecast the last week of data. Plot the predicted data to compare the predicted values to the actual observed ones. Include 95% confidence intervals for the forecasts in the corresponding plots.

```
# Daily Prices
predarima.d <- predict(modarima.d, n.ahead=4)
pred.d = predarima.d$pred

ubound = pred.d+1.96*predarima.d$se
lbound = pred.d-1.96*predarima.d$se
```

```
plot(SBUXDaily[2266:2516],type="l",ylab="Stock Price (USD)",xlab="Time",ylim=c(105,120), main="Daily - 
points(ts(pred.d, start=249),lwd=2,col="red")
lines(ts(ubound,start=249),lty=3,lwd= 2, col="blue")
lines(ts(lbound,start=249),lty=3,lwd= 2, col="blue")
```



```
# Weekly Prices
predarima.w <- predict(modarima.w, n.ahead=1)
pred.w = predarima.w$pred

ubound = pred.w+1.96*predarima.w$se
lbound = pred.w-1.96*predarima.w$se
plot(SBUXWeekly[471:522],type="l",ylab="Stock Price (USD)",xlab="Time",ylim=c(95,120), main="Weekly - P
points(ts(pred.w, start=52),lwd=2,col="red")
points(ts(ubound,start=52),lwd= 2, col="blue")
points(ts(lbound,start=52),lwd= 2, col="blue")
```



2d. Calculate Mean Absolute Percentage Error (MAPE) and Precision Measure (PM). 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.

```
# Daily
test.d = as.vector(test.d)
cat("Daily Model Fit Measures","",
    "\nMAPE:",mean(abs(pred.d-test.d)/test.d),
    "\nPM:",sum((pred.d-test.d)^2)/sum((test.d-mean(test.d))^2))
```

```
## Daily Model Fit Measures
## MAPE: 0.02577881
## PM: 13.53883
```

```
cat("\n\n")
```

```
# Weekly
test.w = as.vector(test.w)
cat("Weekly Model Fit Measures","",
    "\nMAPE:",mean(abs(pred.w-test.w)/test.w),
    "\nPM:",sum((pred.w-test.w)^2)/sum((test.w-mean(test.w))^2))
```

```
## Weekly Model Fit Measures
## MAPE: 0.04820606
## PM: Inf
```

Response: Question 2d All observations are outside the prediction bands. The MAPE scores are low, but the PM scores indicate there is more variance in the models vs the real world.

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 3 and orders for m & n up to 2. Display the summary of the final model fit. Write up the equation of the estimated model.

```
# Daily Fits
num = 4

train.d <- ts(daily.return[1:(length(daily.return)-num)], freq=252)
test.d <- ts(daily.return[(length(daily.return)-num+1):length(daily.return)],freq=252)

test_model.d <- function(p,d,q){
  mod = arima(train.d, order=c(p,d,q), method="ML")
  current.aic = AIC(mod)
  df = data.frame(p,d,q,current.aic)
  names(df) <- c("p","d","q","AIC")
  return(df)
}

orders.d = data.frame(Inf,Inf,Inf,Inf)
names(orders.d) <- c("p","d","q","AIC")

for (p in 0:3){
  for (q in 0:3) {
    possibleError <- tryCatch(
      orders.d <- rbind(orders.d,test_model.d(p,0,q)),
      error=function(e) e
    )
    if(inherits(possibleError, "error")) next

  }
}

orders.d <- orders.d[order(-orders.d$AIC),]
print(tail(orders.d,4))
```

```
##    p d q       AIC
## 10 2 0 0 -13699.46
## 3  0 0 1 -13699.56
## 6  1 0 0 -13700.77
## 17 3 0 3 -13729.79
```

```
# ARMA [3,3]
```

```
# Iteratively modified below to search for ARMA-GARCH parameters, using ARMA [3,3] as the starting poin
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(0, 0), include.mean=T),
    distribution.model="std")
    fit = ugarchfit(spec, train.d, solver = ’hybrid’)
    current.bic = infocriteria(fit)[2]
    if (current.bic < final.bic)
    {
        final.bic = current.bic
        final.order = c(p, q)
    }
}
final.order
```

```
## [1] 1 1
```

```
# ARMA: [3,3], GARCH: [1,1] (ARMA given)
# ARMA: [0,0], GARCH: [1,1] (GARCH given)
# ARMA: [0,0], GARCH: [1,1] (ARMA given) ### FINAL

spec.d = ugarchspec(variance.model=list(garchOrder=c(1,1)),
            mean.model=list(armaOrder=c(0,0),
            include.mean=T), distribution.model="std")
final.model.d = ugarchfit(spec.d, train.d, solver = ’hybrid’)

infocriteria(final.model.d)
```

```
##
## Akaike       -5.868842
## Bayes        -5.857232
## Shibata      -5.868850
## Hannan-Quinn -5.864628
```

```
# Weekly Fits
num = 1

train.w <- ts(weekly.return[1:(length(weekly.return)-num)], freq=52)
test.w <- ts(weekly.return[(length(weekly.return)-num+1):length(weekly.return)],freq=52)

test_model.w <- function(p,d,q){
  mod = arima(train.w, order=c(p,d,q), method="ML")
  current.aic = AIC(mod)
  df = data.frame(p,d,q,current.aic)
  names(df) <- c("p","d","q","AIC")
  return(df)
}

orders.w = data.frame(Inf,Inf,Inf,Inf)
names(orders.w) <- c("p","d","q","AIC")

for (p in 0:3){
  for (q in 0:3) {
    possibleError <- tryCatch(
      orders.w <- rbind(orders.w,test_model.w(p,0,q)),
      error=function(e) e
    )
    if(inherits(possibleError, "error")) next

  }
}

orders.w <- orders.w[order(-orders.w$AIC),]
print(tail(orders.w,4))
```

```
##    p d q       AIC
## 10 2 0 0 -2076.770
## 3  0 0 1 -2077.733
## 6  1 0 0 -2077.986
## 7  1 0 1 -2078.126
```

```
# ARMA [1,1]
```

```
# Iteratively modified below to search for ARMA-GARCH parameters, using ARMA [1,1] as the starting poin
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(0, 0), include.mean=T),
    distribution.model="std")
    fit = ugarchfit(spec, train.w, solver = 'hybrid')
    current.bic = infocriteria(fit)[2]
    if (current.bic < final.bic)
    {
        final.bic = current.bic
        final.order = c(p, q)
    }
}
final.order
```

```
## [1] 1 1
```

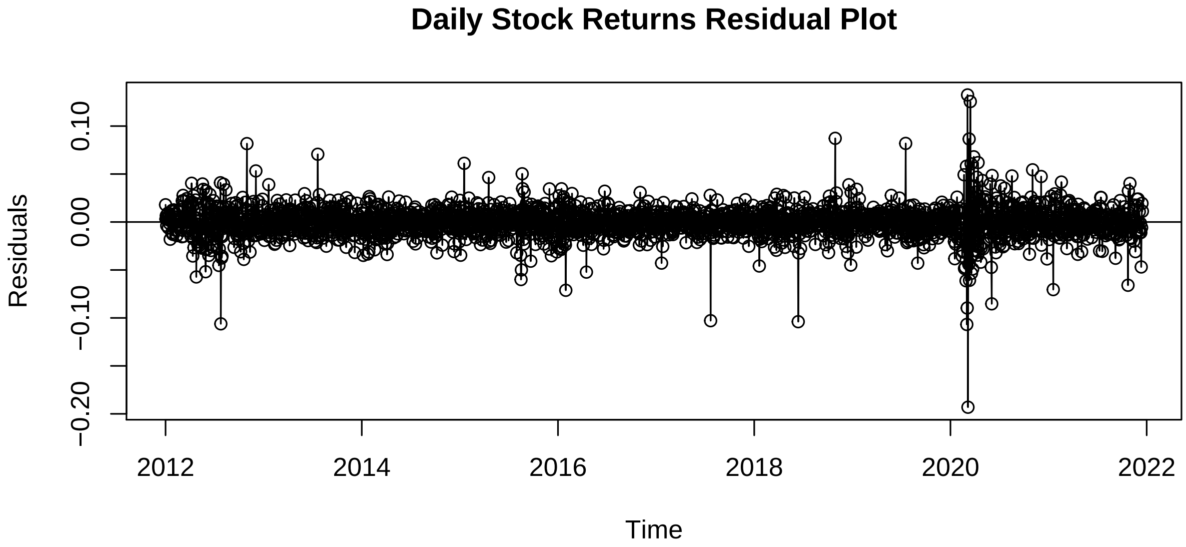
```
# ARMA: [1,1], GARCH: [1,1] (ARMA given)
# ARMA: [0,0], GARCH: [1,1] (GARCH given)
# ARMA: [0,0], GARCH: [1,1] (ARMA given) ### FINAL

spec.w = ugarchspec(variance.model=list(garchOrder=c(1,1)),
            mean.model=list(armaOrder=c(0,0),
            include.mean=T), distribution.model="std")
final.model.w = ugarchfit(spec.w, train.w, solver = 'hybrid')

infocriteria(final.model.w)
```

```
##
## Akaike       -4.177753
## Bayes        -4.136790
## Shibata      -4.177936
## Hannan-Quinn -4.161705
```

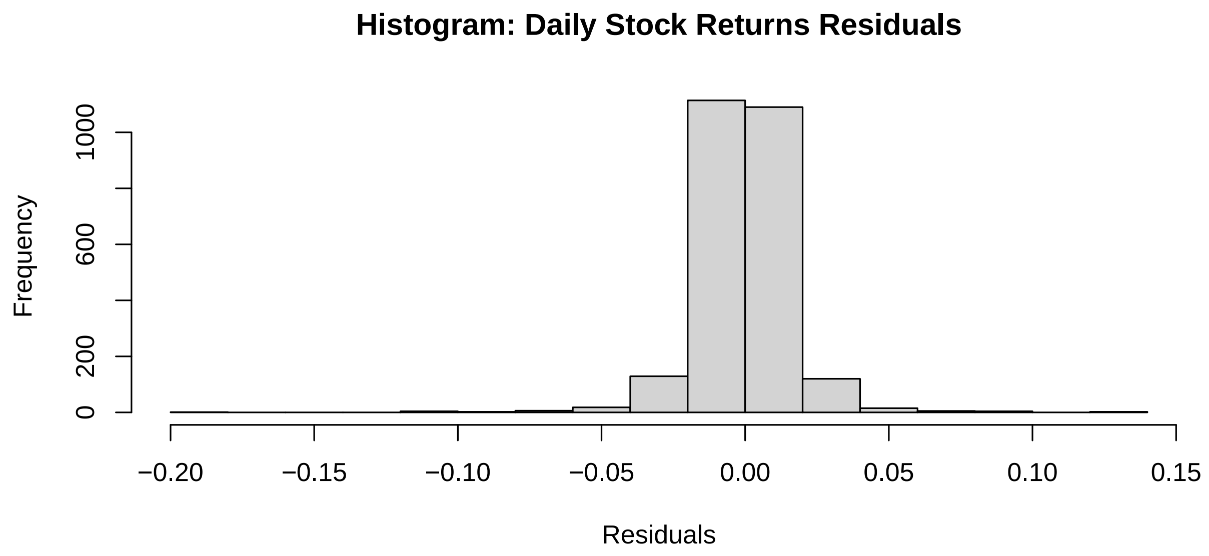
Response: Question 3a For the equations, please refer to the model orders labeled “FINAL” listed above. An important note is that both models are effectively GARCH models.

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?

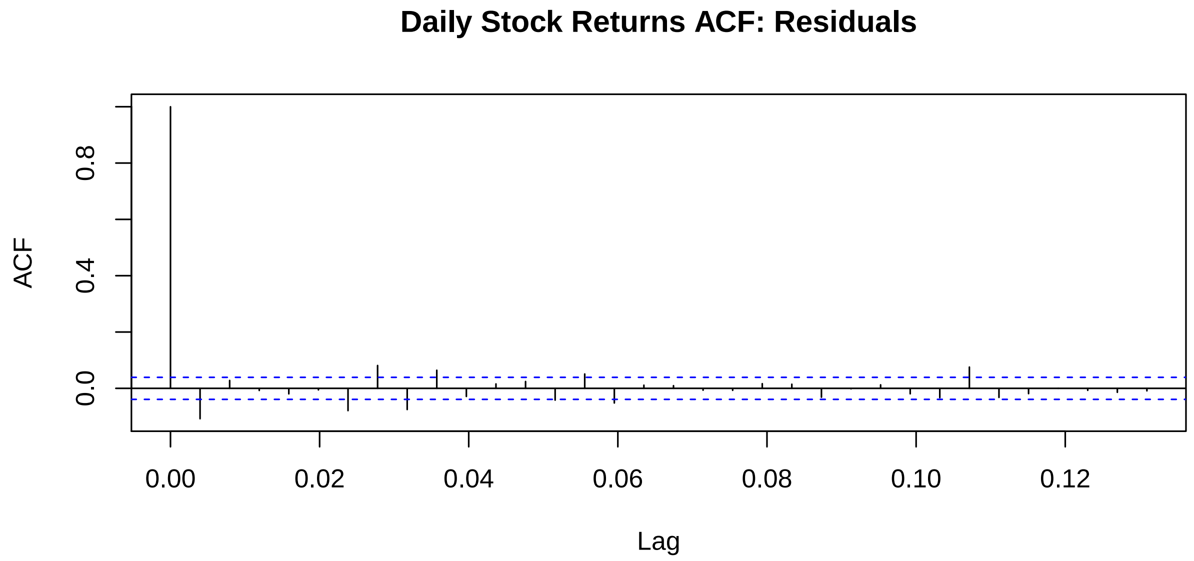
```
# Daily Residuals
res.d <- residuals(final.model.d)
res.d <- ts(res.d, start=c(2012,1,1), freq=252)

plot(res.d,
ylab = 'Residuals',
type = 'o',
main = "Daily Stock Returns Residual Plot")
abline(h = 0)
```



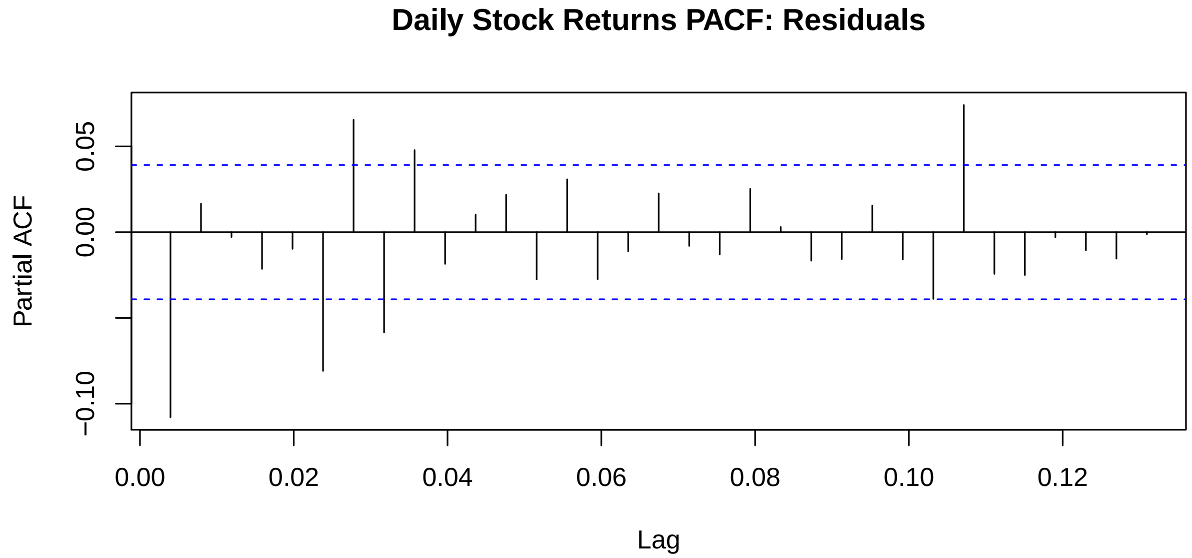
```
hist(res.d, xlab = 'Residuals', main = 'Histogram: Daily Stock Returns Residuals')
```



```r
acf(res.d, main = "Daily Stock Returns ACF: Residuals")
```



```r
pacf(res.d, main = "Daily Stock Returns PACF: Residuals")
```



```r
cat("Residuals:\n")
```

```
## Residuals:
```

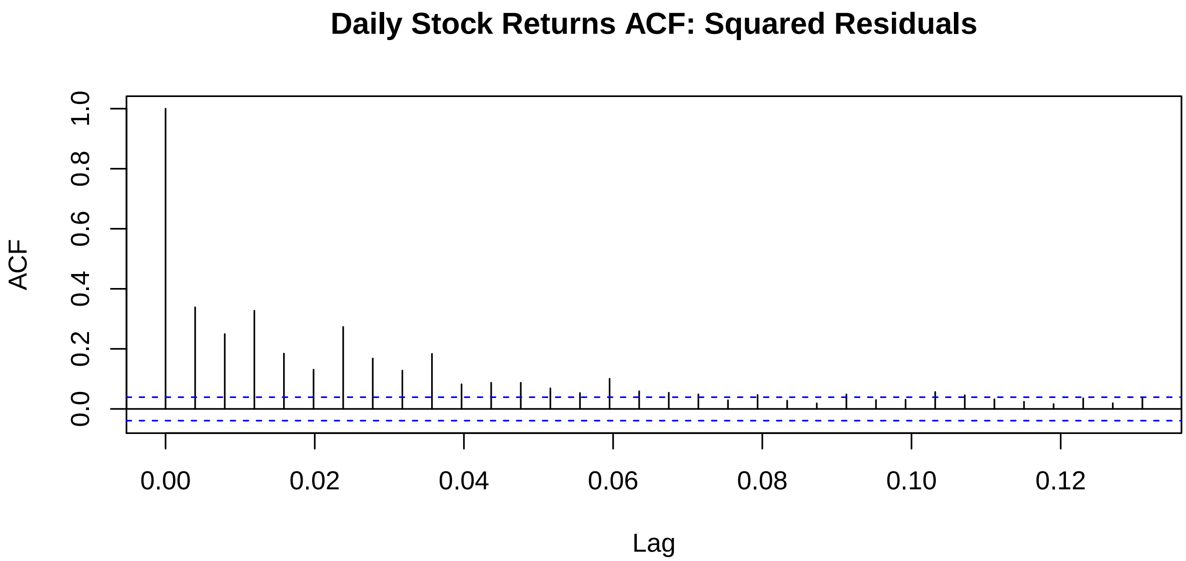
```r
Box.test(res.d, lag = 2, type = "Box-Pierce", fitdf = 2)
```

```
##
##  Box-Pierce test
##
## data:  res.d
## X-squared = 31.203, df = 0, p-value < 2.2e-16
```

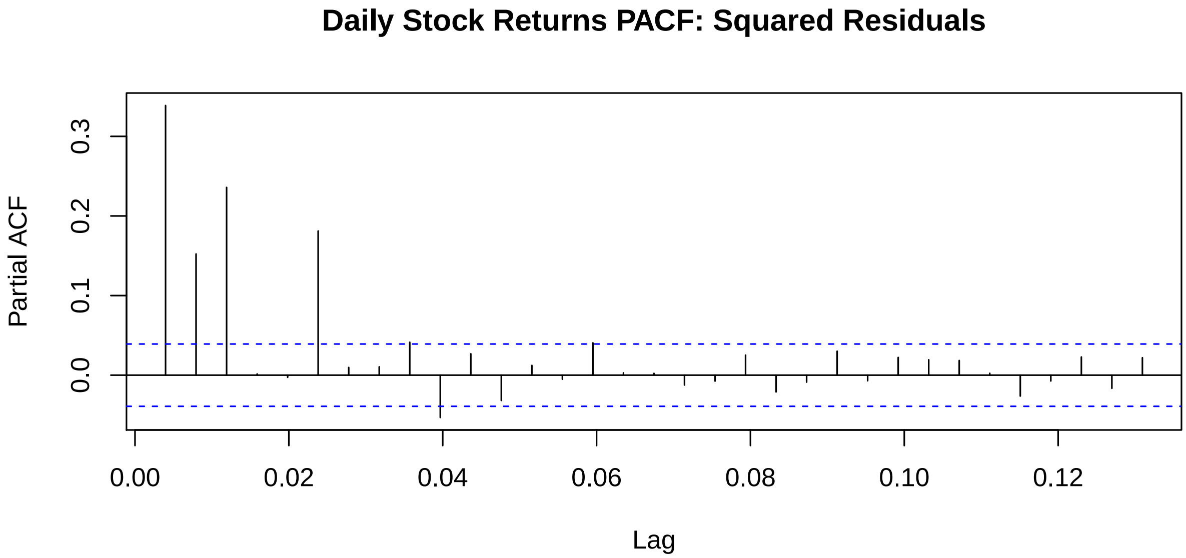
```r
Box.test(res.d, lag = 2, type = "Ljung-Box", fitdf = 2)
```

```
##
##  Box-Ljung test
##
## data:  res.d
## X-squared = 31.241, df = 0, p-value < 2.2e-16
```

```r
acf(res.d^2, main = "Daily Stock Returns ACF: Squared Residuals")
```



```r
pacf(res.d^2, main = "Daily Stock Returns PACF: Squared Residuals")
```



```r
cat("\n\n Squared Residuals:\n")
```

```
##
##
##  Squared Residuals:
```

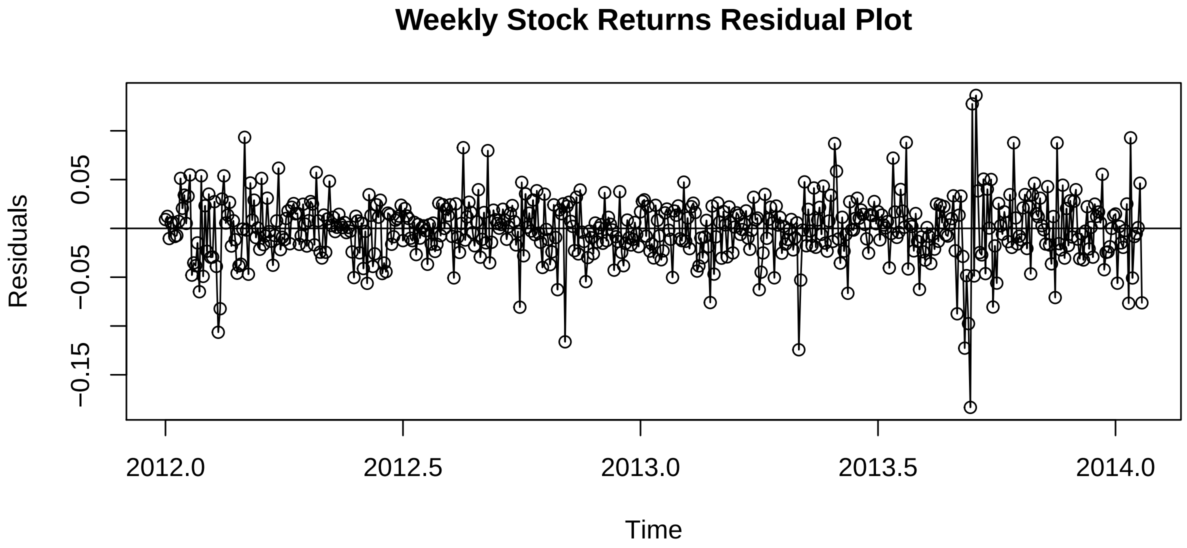
```
Box.test(res.d^2, lag = 2, type = "Box-Pierce", fitdf = 2)
```

```
##
##  Box-Pierce test
##
## data:  res.d^2
## X-squared = 444.37, df = 0, p-value < 2.2e-16
```

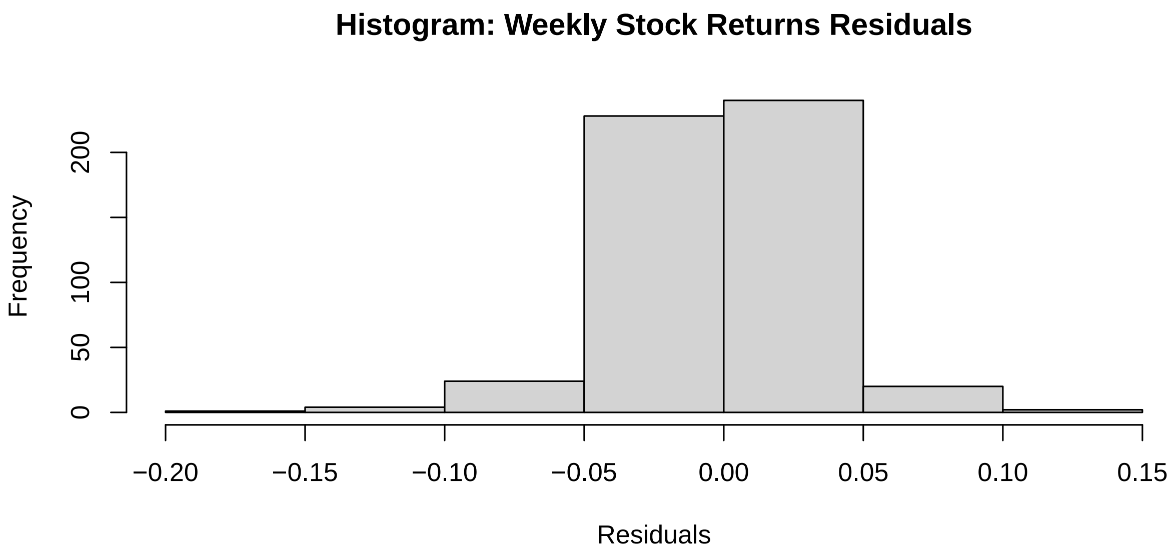
```
Box.test(res.d^2, lag = 2, type = "Ljung-Box", fitdf = 2)
```

```
##
##  Box-Ljung test
##
## data:  res.d^2
## X-squared = 444.96, df = 0, p-value < 2.2e-16
```

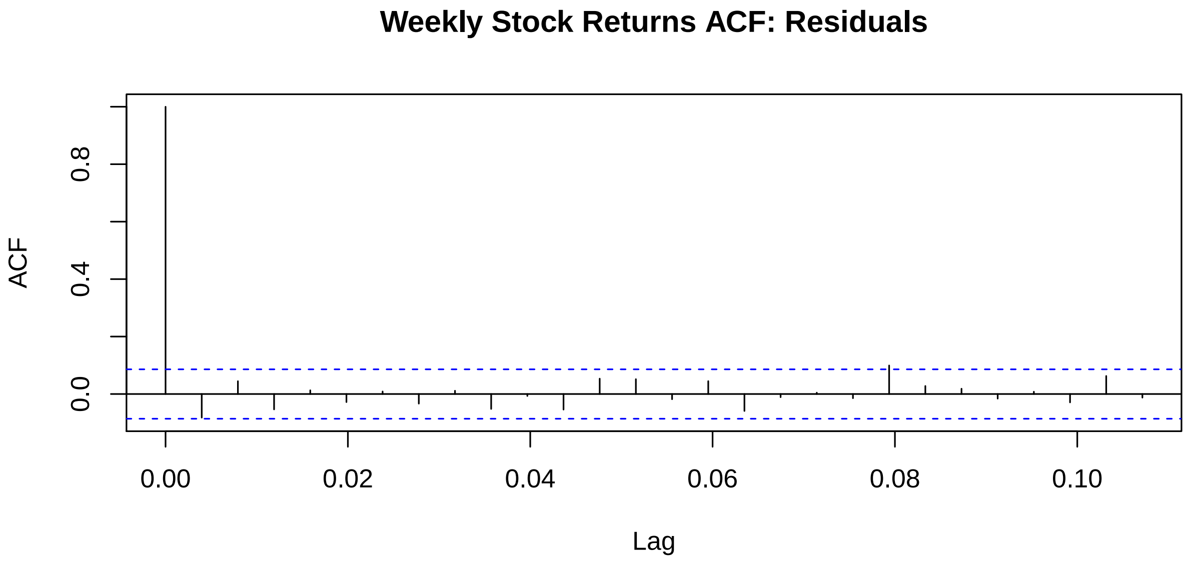
```
# Weekly Residuals
res.w <- residuals(final.model.w)
res.w <- ts(res.w, start=c(2012,1,1), freq=252)

plot(res.w,
ylab = 'Residuals',
type = 'o',
main = "Weekly Stock Returns Residual Plot")
abline(h = 0)
```



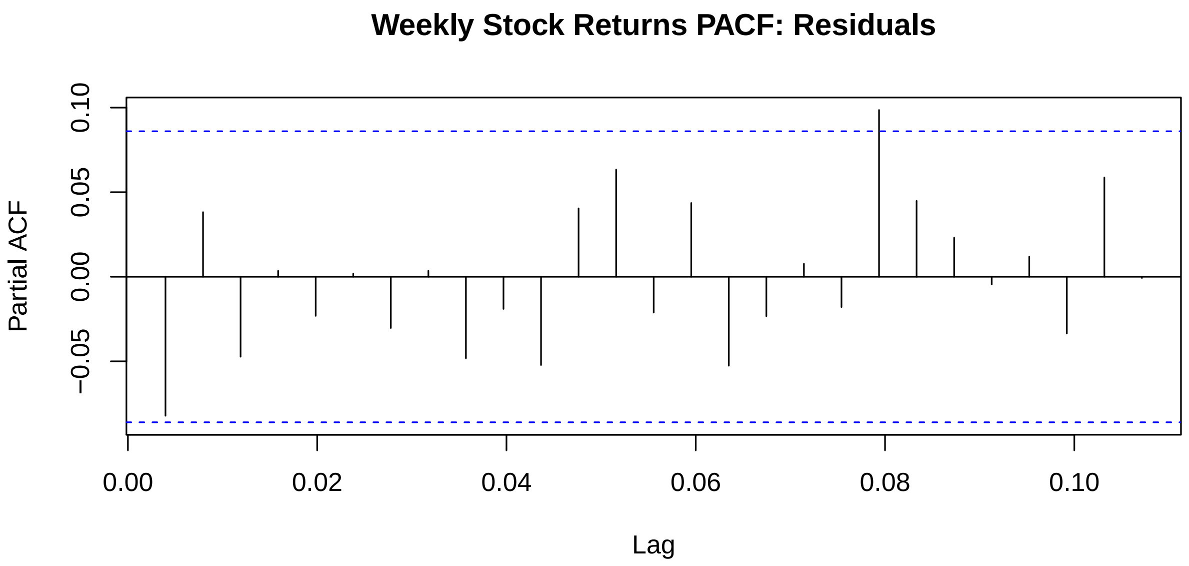
```
hist(res.w, xlab = 'Residuals', main = 'Histogram: Weekly Stock Returns Residuals')
```



```
acf(res.w, main = "Weekly Stock Returns ACF: Residuals")
```



```
pacf(res.w, main = "Weekly Stock Returns PACF: Residuals")
```



```
cat("Residuals:\n")
```

```
## Residuals:
```

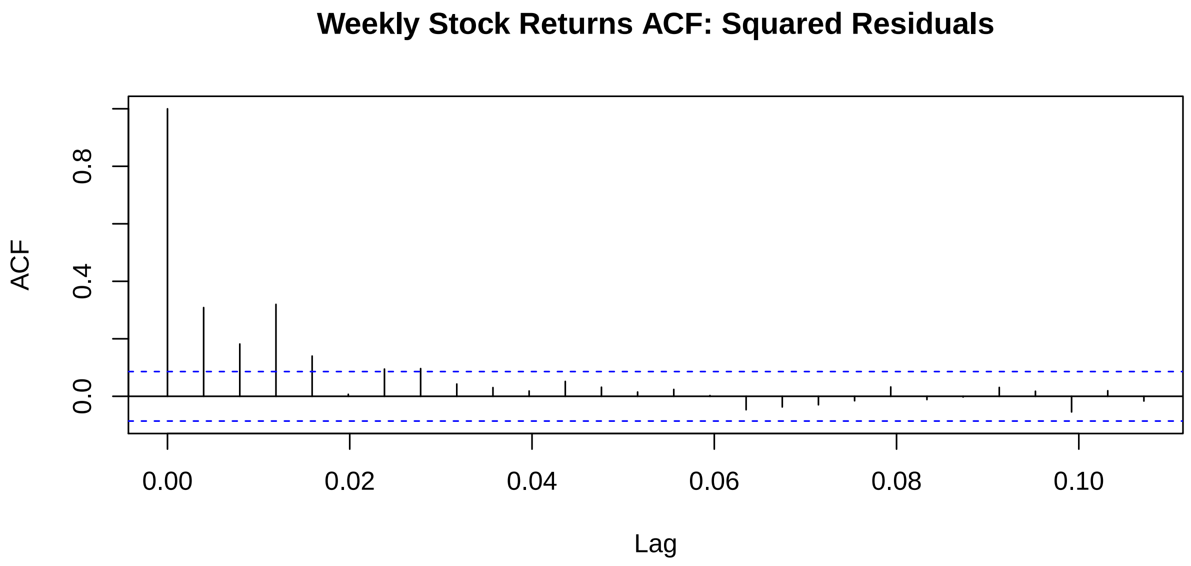
```
Box.test(res.w, lag = 2, type = "Box-Pierce", fitdf = 2)
```

```
##
##  Box-Pierce test
##
## data:  res.w
## X-squared = 4.5445, df = 0, p-value < 2.2e-16
```

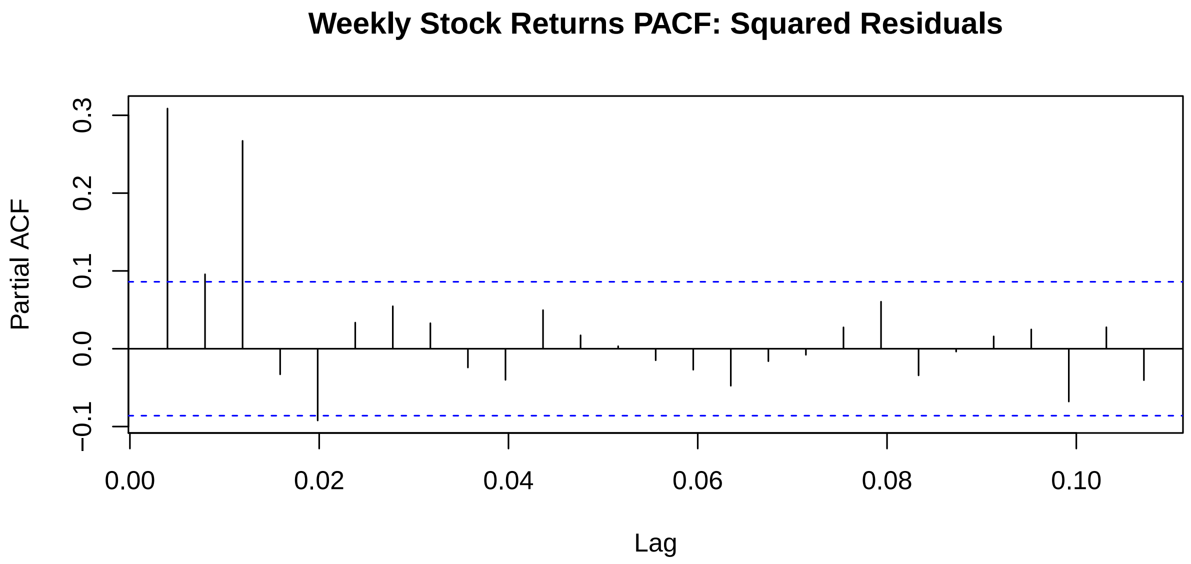
```
Box.test(res.w, lag = 2, type = "Ljung-Box", fitdf = 2)
```

```
##
##  Box-Ljung test
##
## data:  res.w
## X-squared = 4.5728, df = 0, p-value < 2.2e-16
```

```
acf(res.w^2, main = "Weekly Stock Returns ACF: Squared Residuals")
```



```
pacf(res.w^2, main = "Weekly Stock Returns PACF: Squared Residuals")
```



```
cat("\n\n Squared Residuals:\n")
```

```
##
##
##  Squared Residuals:
```

```
Box.test(res.w^2, lag = 2, type = "Box-Pierce", fitdf = 2)
```

```
##
##  Box-Pierce test
##
## data:  res.w^2
## X-squared = 66.635, df = 0, p-value < 2.2e-16
```

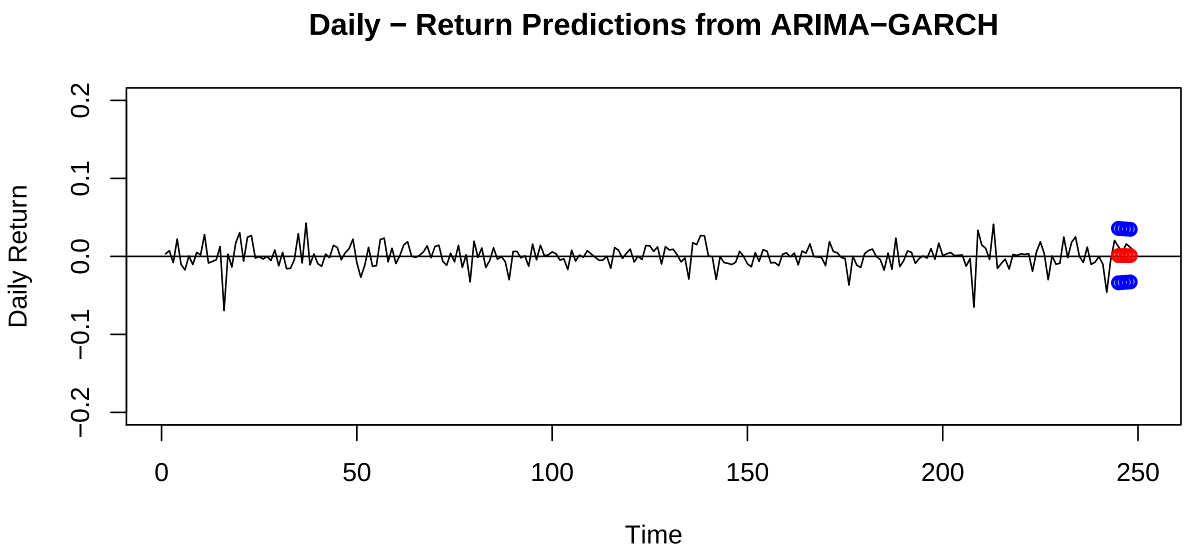
```
Box.test(res.w^2, lag = 2, type = "Ljung-Box", fitdf = 2)
```

```
##
##  Box-Ljung test
##
## data:  res.w^2
## X-squared = 67.054, df = 0, p-value < 2.2e-16
```

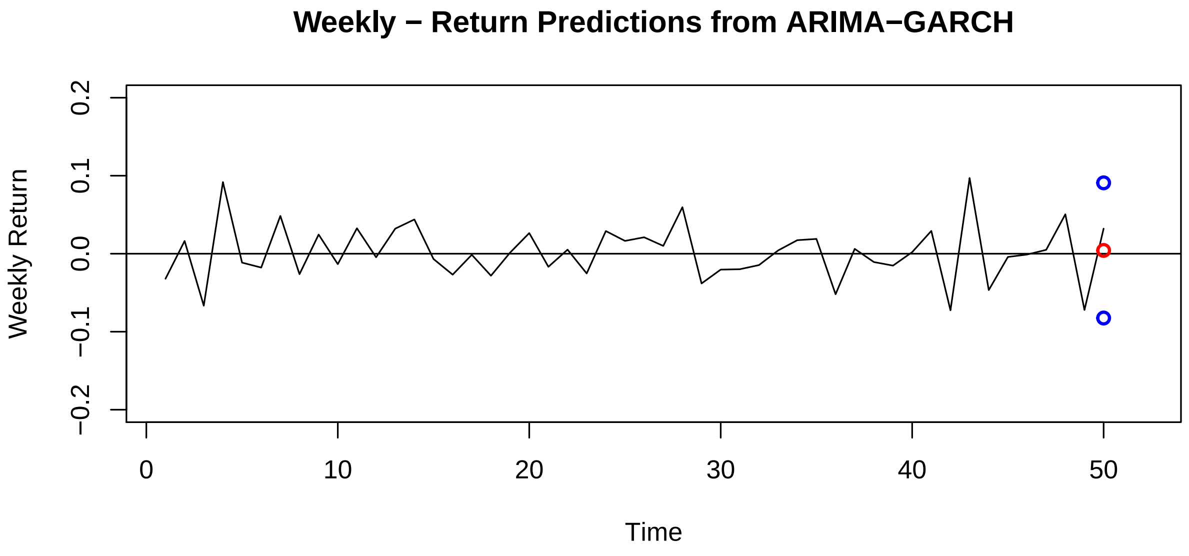
Response: Question 3b The ACF plot of the residuals indicate no serial correlation, but the ACF of the squared residuals show some serial correlation in the variance. The Box tests indicate otherwise since the p-values are less than 0.05.

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. Include 95% confidence intervals for the forecasts in the corresponding plots. Interpret the results, particularly comparing forecast using daily versus weekly data.

```
# Daily
daily.forecast = ugarchforecast(final.model.d, n.ahead=4)
pred.d = daily.forecast@forecast$seriesFor
lbound = pred.d - 1.96*daily.forecast@forecast$sigmaFor
ubound = pred.d + 1.96*daily.forecast@forecast$sigmaFor

plot(daily.return[2266:2516],type="l",ylab="Daily Return",xlab="Time",ylim=c(-.20,0.20), main="Daily - 
abline(h = 0)
points(ts(pred.d, start=245),lwd=2,col="red")
points(ts(ubound,start=245),lwd= 2, col="blue")
points(ts(lbound,start=245),lwd= 2, col="blue")
```



```r
# Weekly
weekly.forecast = ugarchforecast(final.model.w, n.ahead=1)
pred.w = weekly.forecast@forecast$seriesFor
lbound = pred.w - 1.96*weekly.forecast@forecast$sigmaFor
ubound = pred.w + 1.96*weekly.forecast@forecast$sigmaFor

plot(weekly.return[471:522],type="l",ylab="Weekly Return",xlab="Time",ylim=c(-.20,0.20), main="Weekly -
abline(h = 0)
points(ts(pred.w, start=50),lwd=2,col="red")
points(ts(ubound,start=50),lwd= 2, col="blue")
points(ts(lbound,start=50),lwd= 2, col="blue")
```



Response: Question 3c All actual values fall within prediction intervals. The mean forecast for the week is higher than the average daily forecast. This makes sense as the weekly return is dependent on the daily returns. Because the daily returns were all positive, the weekly return would be expected to be greater due to compounding.

3d. Calculate Mean Absolute Percentage Error (MAPE) and Precision Measure (PM) for the mean forecasts. Compare the accuracy of the predictions for the daily and weekly time series using these two measures. Compare the accuracy of the forecasts with those obtained in (2d). Interpret the results.

```r
# Daily
test.d = as.vector(test.d)
cat("Daily Returns Model Fit Measures","",
    "\nMAPE:",mean(abs(pred.d-test.d)/abs(test.d)),
    "\nPM:",sum((pred.d-test.d)^2)/sum((test.d-mean(test.d))^2))
```

```
## Daily Returns Model Fit Measures
## MAPE: 0.8811377
## PM: 5.788173
```

```r
cat("\n\n")
```

```r
# Weekly
test.w = as.vector(test.w)
cat("Weekly Returns Model Fit Measures","",
    "\nMAPE:",mean(abs(pred.w-test.w)/abs(test.w)),
    "\nPM:",sum((pred.w-test.w)^2)/sum((test.w-mean(test.w))^2))
```

```
## Weekly Returns Model Fit Measures
## MAPE: 0.8696371
## PM: Inf
```

Response: Question 3d The PM score for the weekly returns model is the same as what was found in 2d. This is because we are only forecasting for 1 period ahead, causing the PM to tend toward infinity. The PM score for the daily returns model is better than what was found in 2d. However, the MAPE scores for both returns models are higher.

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 The ARIMA models were not good fits for the pricing data. All actual values fell outside the prediction ranges. The returns forecasts, despite their worse MAPE scores compared to the ARIMA models, were closer to the real values (all actual values fell within prediction range). Both sets of models are comparable for weekly vs daily.

GARCH models are better suited for time series with constant mean but non-constant variance (ie returns). ARIMA is a better fit when the time series has a trend and seasonality.