Stock Price Time Series Analysis

DSC 425 – Time Series Analysis and Forecasting

June 7, 2023

Isabelle Young Choi

Lucas de Oliveira

Stephen Kim

Yun Yu

Contents

Non-Technical Report

The stock market is influenced by various factors such as politics, monetary, fiscal policy, industry conditions, the world economy, and financial crises. It has experienced significant fluctuations over the years, including the 2008 financial crisis, the rise of tech stocks due to innovations like Apple's smartphones, and the extreme volatility caused by the 2020 pandemic. Unlike the bond market, stocks carry higher uncertainty as future returns are not guaranteed. However, the higher risks are associated with the potential for higher returns. Therefore, understanding the risk-return tradeoff and analyzing company management strategies are crucial for investors in making investment decisions. In this project, we focused on analyzing stock price movements through a statistical modeling approach known as technical analysis.

Two stocks were analyzed in this project. One is Chevron, an oil company established in the United States in 1879, which provides insights into the overall U.S. market due to its long-standing presence. The other is Apple, which revolutionized the market with its innovative device, the smartphone, in 2008 and has since become a leading company in the digital era. Both stock datasets were obtained from Yahoo Finance.

The project started by analyzing daily stock prices and examining data distribution and trends. The relationship between current and past prices was analyzed over different time periods, and an average price or mean model was developed. The final statistical model was determined, and predictions were made based on the model.

However, there are limitations in this project. The final model was developed using past market data and did not take into account future expectations of investors regarding the economy and the market. Stock prices can increase or decrease based on various factors, including economic conditions and company performance. Therefore, relying solely on technical analysis to predict stock prices carries significant risks. It is recommended to incorporate economic factors, industry conditions, and other relevant variables in future studies to improve the predictive performance of the model.

Technical Summary

2-1. Introduction

As data volumes keep sharply increasing in all sorts of industry, the ability to model and drive insight from it becomes more critical by the day. Time series analysis is, by no means, a novelty as G. U Yule and J.Walker started modelling stochastic processes back in the beginning of the twentieth century. Time series analysis is, however, an incredibly powerful mathematical tool for both analysis and forecasting, thus contributing to a better understanding of high volumes of data and support of industries of all kinds, be it healthcare, manufacturing, marketing, or finance.

The stock market was first created back in the late 1400's, in Belgium, when merchants started buying goods anticipating a rise in their prices. For a long time now, people have been investing in stocks as a means of protecting their wealth and their future financial security.

Stock refers to purchasing or selling fractional ownership of equity in an organization. People might choose to invest in a company through the acquisition of its stocks in the stock market, which allows for its seamless exchange. People's economic status could be significantly affected by stock price and, as a result, effectively predicting stock trends could aim to increase the profit and reduce financial risk for an immense amount of people all over the world.

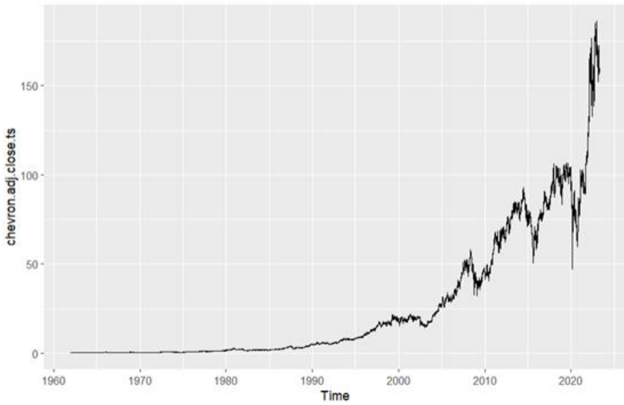
This project's goal is to apply time-series analysis on stock price which shall allow a better understanding of its behavior as well as to find predictive information in data collected over time.

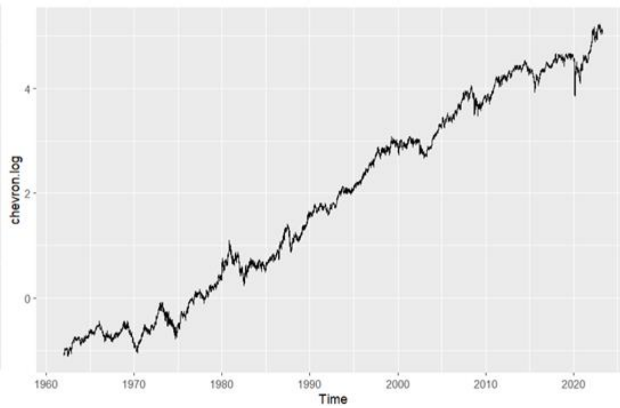
To conduct this study, stock daily prices for Chevron and Apple were extracted from Yahoo's finance's website.

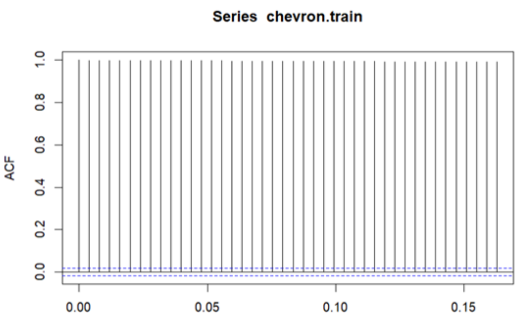
2-2. Chevron

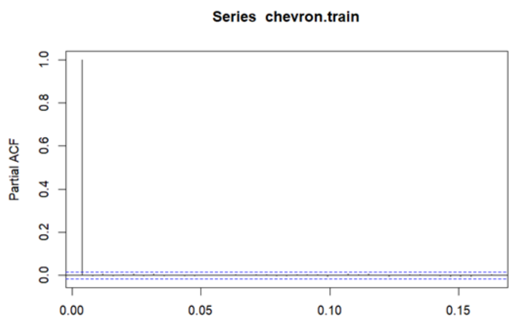
a. Exploratory Analysis

As can be seen on the image below, on the left, the time series for the Chevron stocks is multiplicative, which required the log transform resulting in the image below, on the right. The decomposition of the log transformed time series showed a clear trend, but no seasonality.





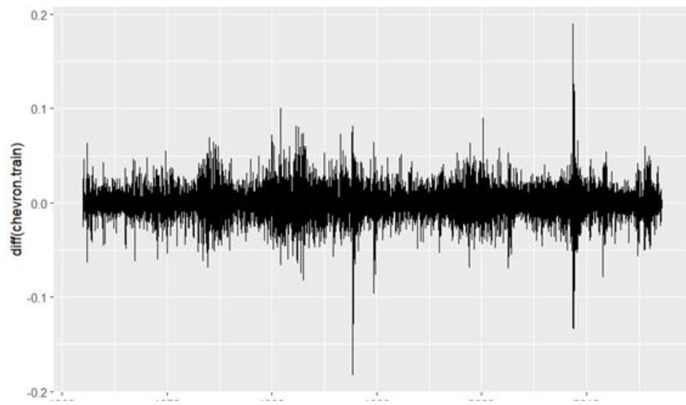





```
> adfTest(chevron.train, lags = 15, type = 'ct')

Title:
 Augmented Dickey-Fuller Test

Test Results:
  PARAMETER:
    Lag Order: 15
  STATISTIC:
    Dickey-Fuller: -3.3267
  P VALUE:
    0.06586

> kpss.test(chevron.train, null = 'Trend')

        KPSS Test for Trend Stationarity

data:  chevron.train
KPSS Trend = 8.4037, Truncation lag parameter = 13, p-value = 0.01
```

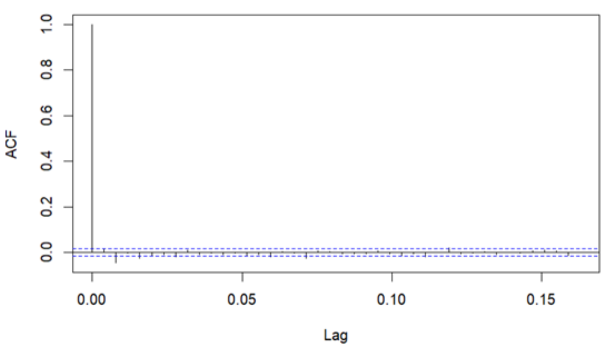
```
> eacf(chevron.train)
AR/MA
  0 1 2 3 4 5 6 7 8 9 10 11 12 13
0 x x x x x x x x x x x  x  x  x
1 o x o x o o x o o o o  o  o  o
2 x x o o o o x o o o o  o  o  o
3 x x o x o o x o o o o  o  o  o
4 x x x o o o o o o o o  o  o  o
5 x x x x o o o x o o o  o  o  o
6 x x x x x o o x o o o  o  o  o
7 x x x x x x o x o o o  o  o  o
```

The incredibly slow decay of the auto correlation on the ACF, the auto correlation of one at lag 1 on the PACF and the first row filled with x on the EACF, as well as the ADF test failing to reject the null hypothesis of unit root and the KPSS test rejecting the null hypothesis of stationarity all indicate the series is a random walk.

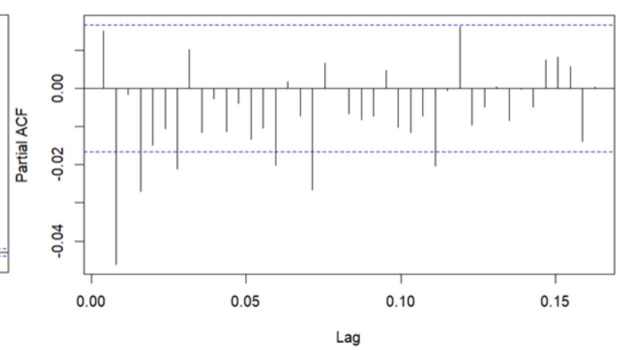



```
> t.test(diff(chevron.log))

        One Sample t-test

data:  diff(chevron.log)
t = 3.0284, df = 15443, p-value = 0.002463
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
 0.0001395019 0.0006514341
sample estimates:
  mean of x
0.000395468
```

Series diff(chevron.train)



Series diff(chevron.train)



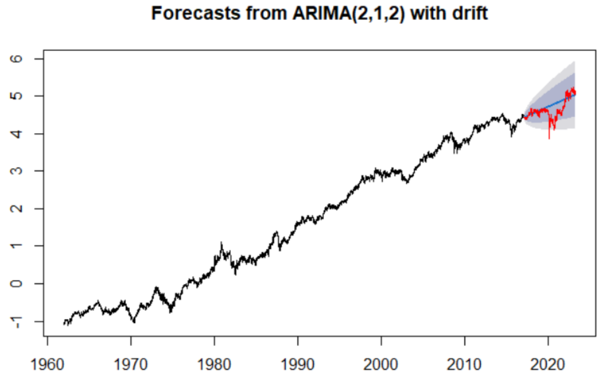

```
> adfTest(diff(chevron.train), lags = 15, type = 'nc')

Title:
 Augmented Dickey-Fuller Test

Test Results:
  PARAMETER:
    Lag Order: 15
  STATISTIC:
    Dickey-Fuller: -31.6217
  P VALUE:
    0.01
> kpss.test(diff(chevron.train), null = 'Level')

        KPSS Test for Level Stationarity

data:  diff(chevron.train)
KPSS Level = 0.037369, Truncation lag parameter = 13, p-value = 0.1
```

```
> eacf(diff(chevron.train))
AR/MA
  0 1 2 3 4 5 6 7 8 9 10 11 12 13
0 o x o x o o x o o o o  o  o  o
1 x x o o o o x o o o o  o  o  o
2 x x o x o o x o o o o  o  o  o
3 x x x o o o o o o o o  o  o  o
4 x x x x o o o x o o o  o  o  o
5 x x x x x o o x o o o  o  o  o
6 x x x x x x o x o o o  o  o  o
7 x x x x x x x x o o o  o  o  o
```

By differencing the series we can see that it becomes stationary, and the T test rejects the null hypothesis of constant mean throughout the series, indicating a random walk with a drift.

After differencing it, ACF, PACF and EACF as well as ADF and KPSS tests all corroborated a stationary behavior. ACF of the differenced series indicates an MA behavior and the significant auto correlation at lag 2 indicates it might be of second order. The PACF shows more clearly some significant auto correlation at lags 2 and 3, aside from other ones at much higher orders. EACF indicates many possible models, and of the many tried, the one with the best performance and most conforming residuals was an ARIMA(2, 1, 2) with drift.

b. Model Building

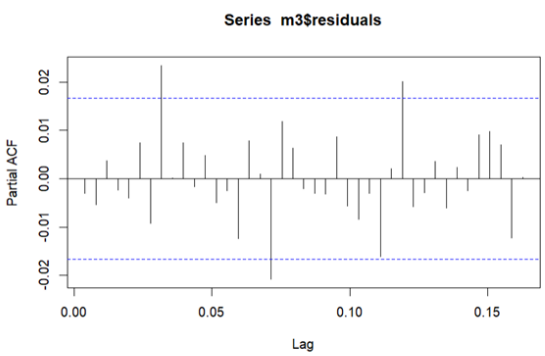



```
> m3
Series: chevron.train
ARIMA(2,1,2) with drift

Coefficients:
         ar1     ar2      ma1      ma2  drift
      0.3876  0.4802  -0.3708  -0.5300  4e-04
s.e.  0.0924  0.0889   0.0896   0.0869  1e-04

sigma^2 = 0.0002426:  log likelihood = 38128.33
AIC=-76244.66   AICc=-76244.65   BIC=-76199.42
> coeftest(m3)

z test of coefficients:

        Estimate  Std. Error z value  Pr(>|z|)
ar1    3.8758e-01  9.2424e-02  4.1935 2.747e-05 ***
ar2    4.8019e-01  8.8898e-02  5.4016 6.606e-08 ***
ma1   -3.7079e-01  8.9564e-02 -4.1400 3.473e-05 ***
ma2   -5.3003e-01  8.6914e-02 -6.0983 1.072e-09 ***
drift  3.8899e-04  9.9472e-05  3.9106 9.208e-05 ***
```

RMSFE: 0.22,
MAPE: 3.5%

The model chosen has very significant coefficients, with very low RMSFE and MAPE, at .22 and 3.5%, respectively, as well as incredibly low AIC and BIC values, at -76244.66 and -76199.42. The performance of the model can be visually analyzed on the chart on the left, where the blue line represents the predictions produced and the red line represents the true values for the period.

c. Residual Analysis



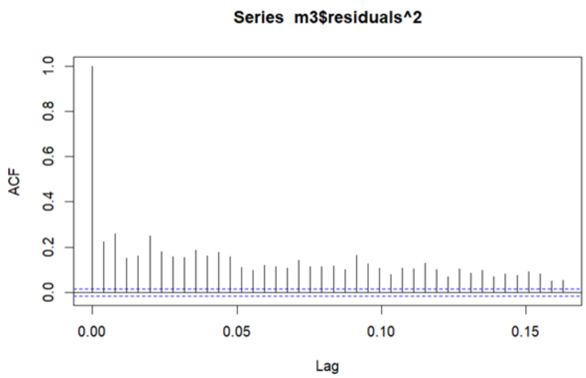

```
> Box.test(m3$residuals, lag = 10, type = 'Ljung')

	Box-Ljung test

data:  m3$residuals
X-squared = 11.313, df = 10, p-value = 0.3337
```

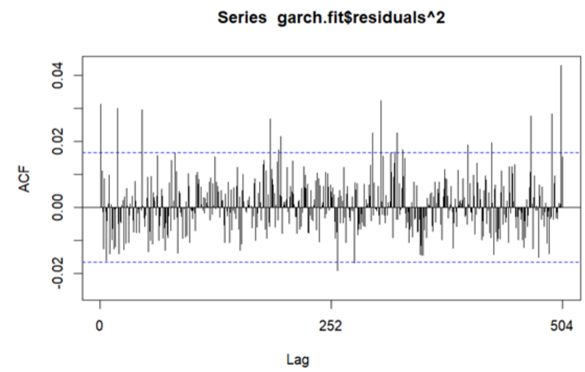
The residuals, as can be seen on the PACF on the left, look very good, with only a few auto correlations that are barely significant, and the Ljung Box test fails to reject the null hypothesis of independence.

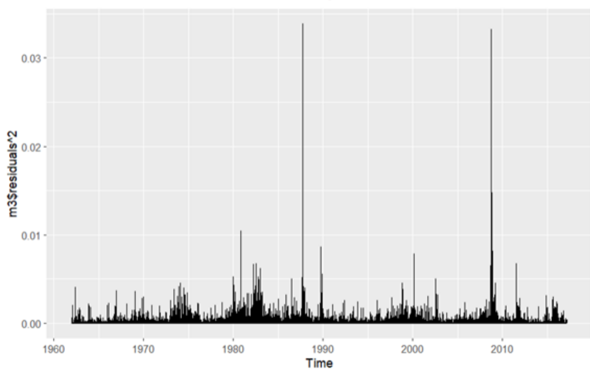
Even though the residuals for the model look independent, there is still some volatility to it, which can be seen on the ACF and time plot, on the left middle and bottom corner of the image below, respectively. By running a first GARCH (1, 1) model the volatility of the model reduced significantly, as can be seen on the ACF and time plot for the squared residuals, on the right middle and bottom corner of the image below. The coefficients for the first GARCH model were all very significant as can be seen on the left upper corner, and the Ljung box test came close to failing to reject the null hypothesis of independence at p-value of 0.0002. These results motivated the integration of the ARIMA and GARCH models into an ARMA(2, 2) GARCH(1, 1) model.

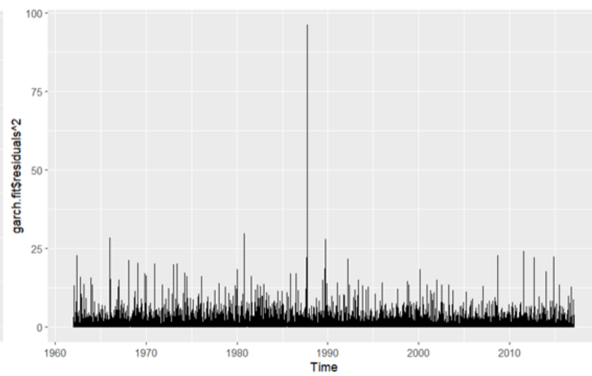
```
> coeftest(garch.fit)

z test of coefficients:

     Estimate Std. Error z value  Pr(>|z|)
a0 2.5854e-06 2.2796e-07  11.342 < 2.2e-16 ***
a1 6.0380e-02 2.4737e-03  24.408 < 2.2e-16 ***
b1 9.2934e-01 2.8817e-03 322.492 < 2.2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

```
> Box.test(garch.fit$residuals^2, type = 'Ljung')

        Box-Ljung test

data:  garch.fit$residuals^2
X-squared = 13.591, df = 1, p-value = 0.0002273
```



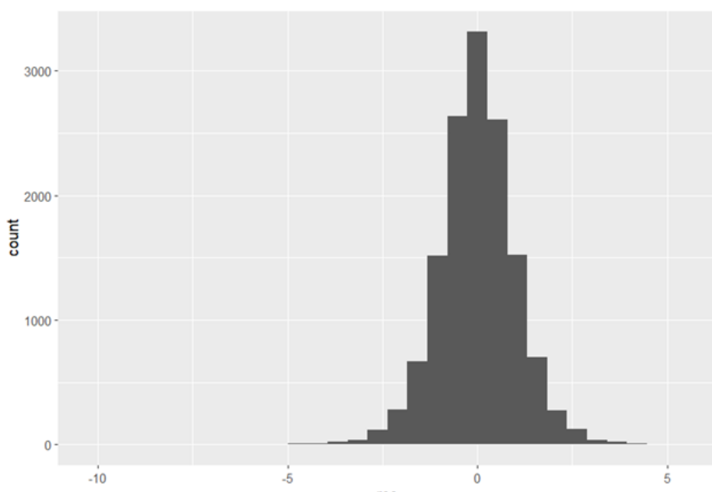







```
        Box-Ljung test

data:  gfit4res^2
X-squared = 25.625, df = 15, p-value = 0.04216
```

Skewness: -0.15
Kurtosis: 2.19

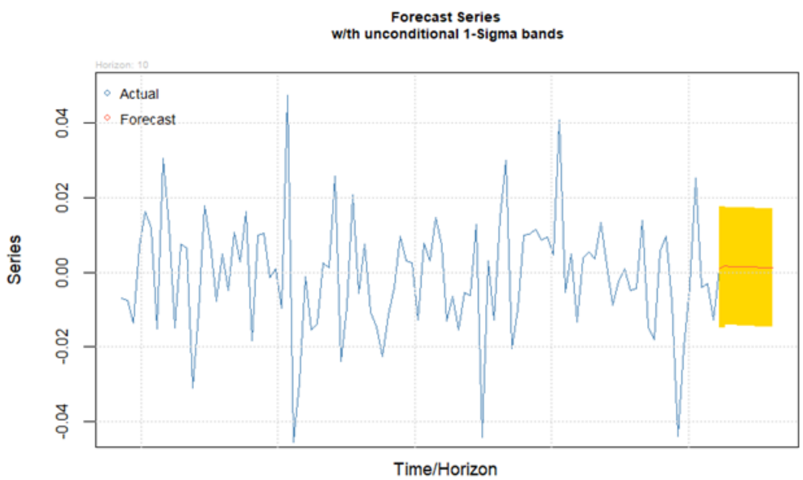



```
> jarque.bera.test(na.omit(gfit4res)) # rejects normality

        Jarque Bera Test

data:  na.omit(gfit4res)
X-squared = 3149.3, df = 2, p-value < 2.2e-16
```

As expected, all coefficients for the integrated model were significant and the volatility was kept at a very low level, just like on the first ARIMA(2, 1, 2) and GARCH(1, 1) models. Even though the differences for the squared residuals between the first GARCH model and the integrated model were visually imperceptible, the Ljung box test for the ARMA(2, 2), GARCH(1, 1) model returned a p-value of .04, on the edge of failing to reject the null hypothesis of independence, indicating the integrated model has extremely well behaved residuals. The residuals presented a somewhat normal behavior, as can be seen on the histogram to the left, with a skewness of -0.15 and a Kurtosis of 2.19, but not enough for the Jarque Bera test to fail to reject the null hypothesis of normality.

d. Forecast Analysis

The forecasts from the ARMA(2, 2) GARCH(1, 1) model can be seen below, with really low MSE and MAE. The integrated model has equal contributions from auto regressive and moving average processes, seeing that AR 1 and 2, MA 1 and 2, Alpha 1 and Beta 1 are all very significant.



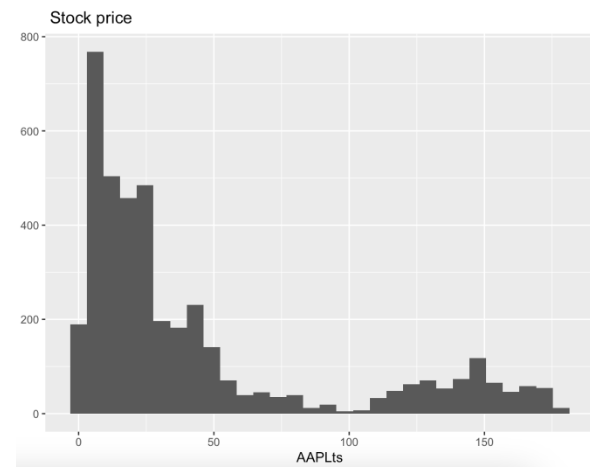

```
> report(gfit4.rolltest, type = 'fpm')

GARCH Roll Mean Forecast Performance Measures
---------------------------------------------
Model            : sGARCH
No.Refits        : 27
No.Forecasts: 13444

        Stats
MSE 0.0002834
MAE 0.0119000
DAC 0.4953000
Optimal Parameters
------------------------------------
        Estimate  Std. Error   t value Pr(>|t|)
mu      0.000571    0.000076    7.5404  0.00000
ar1     0.877407    0.008244  106.4320  0.00000
ar2     0.050740    0.003890   13.0446  0.00000
ma1    -0.843608    0.002508 -336.3150  0.00000
ma2    -0.103779    0.008259  -12.5660  0.00000
omega   0.000003    0.000001    2.7511  0.00594
alpha1  0.064495    0.008041    8.0210  0.00000
beta1   0.924647    0.009772   94.6232  0.00000
```

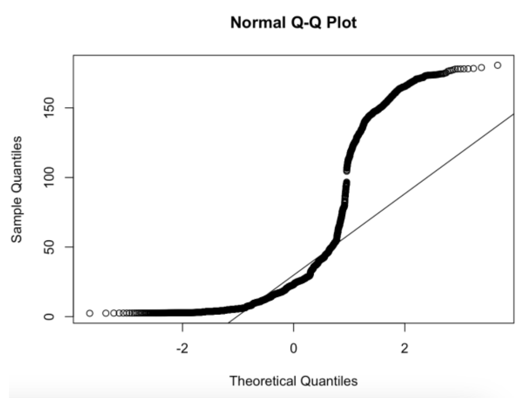
2-3. Apple

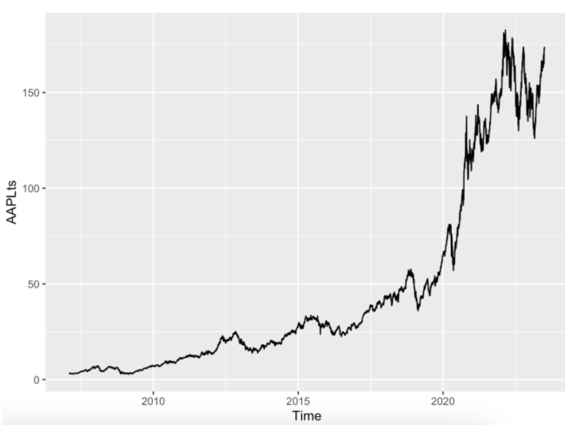
a. Exploratory Analysis

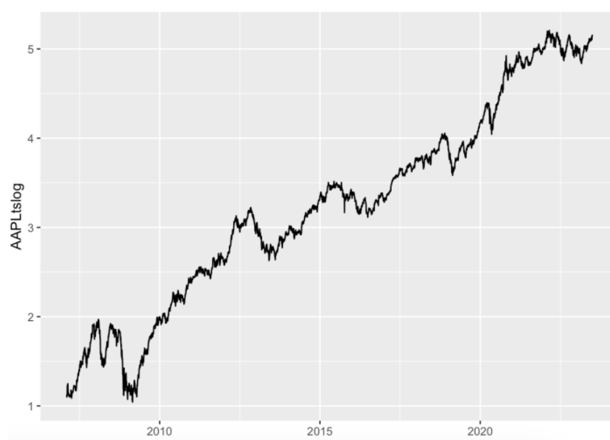
```
> head(AAPL)
# A tibble: 6 × 7
  Date        Open  High   Low Close `Adj Close`     Volume
  <date>     <dbl> <dbl> <dbl> <dbl>       <dbl>      <dbl>
1 2007-01-03  3.08  3.09  2.92  2.99        2.55 1238319600
2 2007-01-04  3.00  3.07  2.99  3.06        2.60  847260400
3 2007-01-05  3.06  3.08  3.01  3.04        2.59  834741600
4 2007-01-08  3.07  3.09  3.05  3.05        2.60  797106800
5 2007-01-09  3.09  3.32  3.04  3.31        2.81 3349298400
6 2007-01-10  3.38  3.49  3.34  3.46        2.95 2952880000
```

Apple's IPO was launched in 1980, but we decided to focus on the data from 2007 to now since it is the year that the iPhone was released, which is the key factor that affects the fluctuation of the Appl's stock prices. From the histogram below, we can tell this is not a normal distribution and it is heavily left skewed. From the qq-plot, we can see that it has a heavy tail.

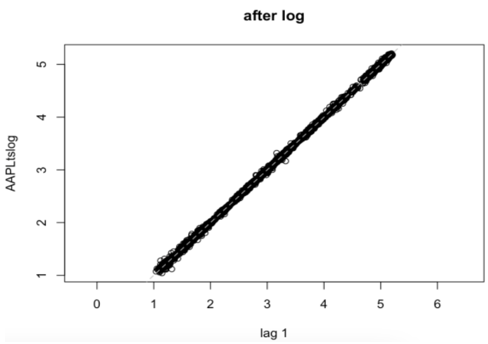




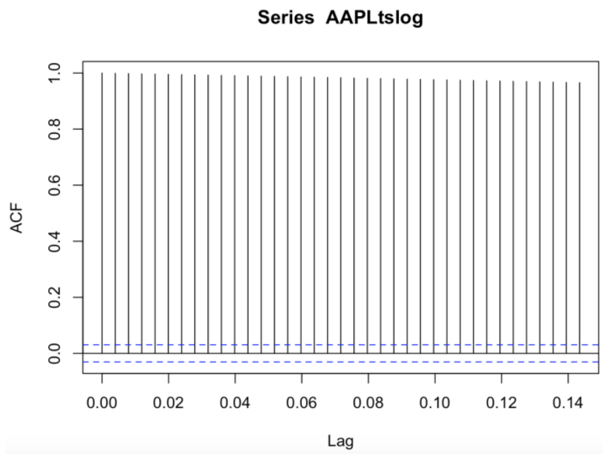


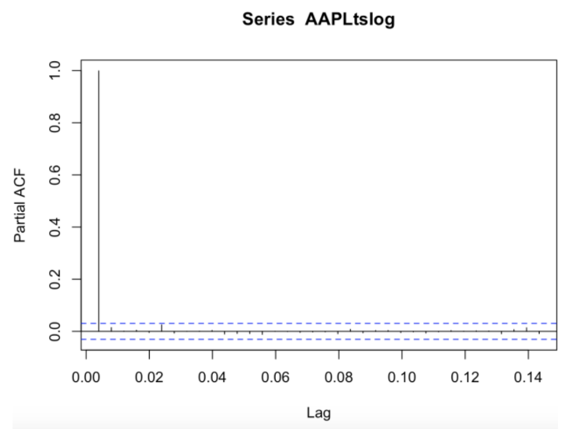


From the graph before logging, we can see an exponentially upward trend, which indicates the stock price emotionally going up. This is a multiplicative series, because the bigger the value is, the bigger the swing is going to be, so we need to take the log to help to stabilize the variance of the time series. After logging the data, we can see a plot likely to be a random walk. On the lag plot below, we can see a strong correlation between the series of time t and the series of time t-1, which indicates there is a serial dependence.



We can see that the ACF decays slowly and, therefore, the time series should be non-stationary, which is corroborated by an auto correlation of one at the PACF's lag one.





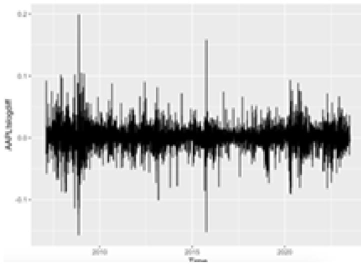

```
> adfTest(AAPLtslog,type="ct")

Title:
 Augmented Dickey-Fuller Test

Test Results:
  PARAMETER:
    Lag Order: 1
  STATISTIC:
    Dickey-Fuller: -2.7463
  P VALUE:
    0.2623

Description:
 Mon May 29 17:05:48 2023 by user:
```

```
> kpss.test(AAPLtslog)

        KPSS Test for Level Stationarity

data:  AAPLtslog
KPSS Level = 35.323, Truncation lag parameter = 10, p-value = 0.01
```

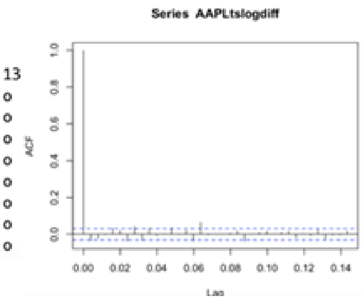
Differencing can help stabilize the mean of a time series by removing changes in its level, thus eliminating or reducing trend and seasonality. After differencing we could see that the autoplot is stable, although there is something left at two points.

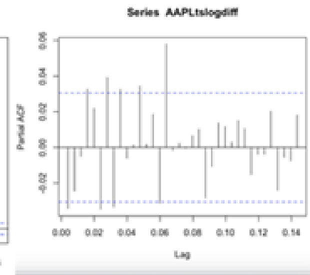
We can then check the Acf and Pacf of the series, to see that there are some autocorrelations but it is close to white noise. A lot of stock time series will have little autocorrelation because of the efficient market hypothesis.

From the ACF we can see that maybe the series is an AR(2) process. From the PACF we can see there is an observed correlation at lags four, seven and seventeen, indicating a possible MA(4) or MA(7) process.



```
> eacf(AAPLtslogdiff)
AR/MA
  0 1 2 3 4 5 6 7 8 9 10 11 12 13
0 x o o x o x x x x o o x  o  o
1 x o o o o o o o o o o x  o  o
2 x x o o o o o o o o o o  o  o
3 x x o o o o o o o o o x  o  o
4 x x o x o o o o o o o o  o  o
5 x o x x x o o o o o o o  o  o
6 x x x x x x o o o o o o  o  o
7 x x x o x x o o o o o o  o  o
```





Next, we can run an adfTest, which indicates that we can reject the null hypothesis of unit root. At the same time, we can run a kpss.test, which fails to reject the null hypothesis of stationarity.

```
> adfTest(AAPLtslogdiff,type="nc")

Title:
 Augmented Dickey-Fuller Test

Test Results:
  PARAMETER:
    Lag Order: 1
  STATISTIC:
    Dickey-Fuller: -47.0962
  P VALUE:
    0.01

Description:
 Wed May 24 14:15:34 2023 by user:
```

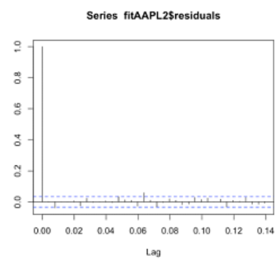
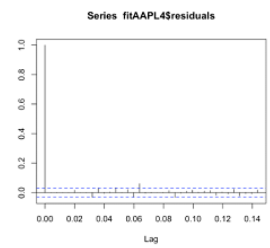
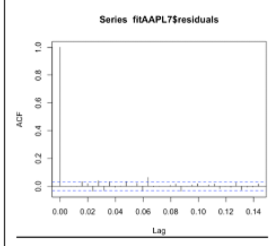
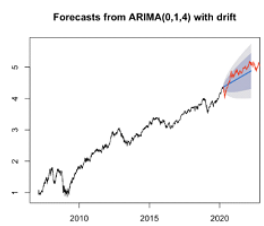
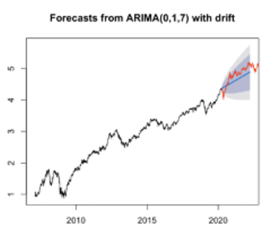
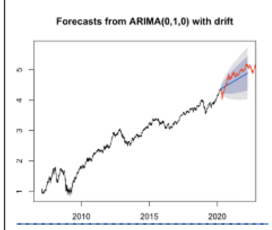
```
> adfTest(AAPLtslogdiff,type="ct")

Title:
 Augmented Dickey-Fuller Test

Test Results:
  PARAMETER:
    Lag Order: 1
  STATISTIC:
    Dickey-Fuller: -47.2752
  P VALUE:
    0.01
```

```
> adfTest(AAPLtslogdiff,type="c")

Title:
 Augmented Dickey-Fuller Test

Test Results:
  PARAMETER:
    Lag Order: 1
  STATISTIC:
    Dickey-Fuller: -47.2799
  P VALUE:
    0.01
```

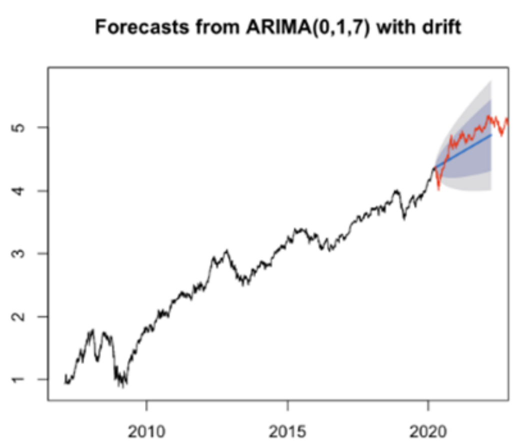
```
> kpss.test(AAPLtslogdiff)
```

```
> kpss.test(AAPLtslogdiff)

        KPSS Test for Level Stationarity

data:  AAPLtslogdiff
KPSS Level = 0.040289, Truncation lag parameter = 10, p-value = 0.1
```

b. Model Building

Model	ARIMA(0,1,4) with drift	ARIMA(0,1,7) with drift	ARIMA(0,1,2) with drift
Sigma^2	0.0003866	0.0004103	0.0004116
AIC	-16537.47	-20430.04	-20419.89
BIC	-16519.17	-20385.78	-20394.59
log likelihood	8271.7	10222.02	10213.94
Box.test p-value	0.5005	0.5192	0.03028
ACF Residual			
MAPE	0.0387	0.0386	0.0384
Forecast Plot			

ARIMA stands for Autoregressive Integrated Moving Average, and is capable of capturing a suite of different standard temporal structures in time-series data. The model is a good fit if the residuals show white noise behavior. From ACF and PACF we can try ARIMA(0,1,4) with drift, ARIMA(0,1,7) with drift and auto.arima. We can check the performance of a given model based a series of metrics shown on the table to the left. The ARIMA(0,1,7) model performs best among the three models.

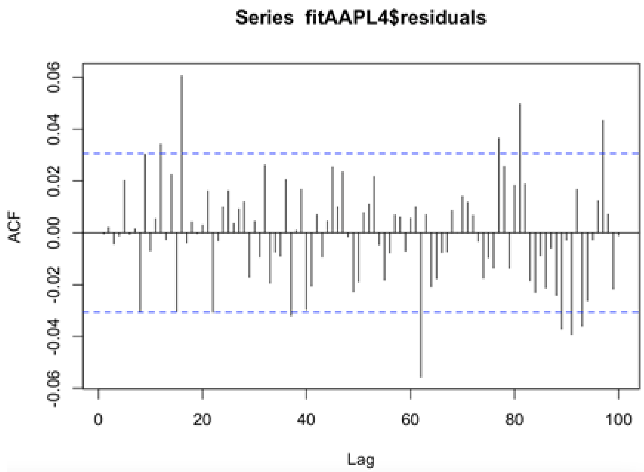


We separate 80% of the observation from the dataset as training data and 20% of the observations as test data to verify the performance of the mode. We can see that the MAPE from back test of three models are around 3.8%. We built the forecasting model based on a test dataset. As we can see, we have a blue line that represents our predictions and the red line that represents the real values. The darker and lighter regions represent 80% and 95% confidence intervals, respectively.

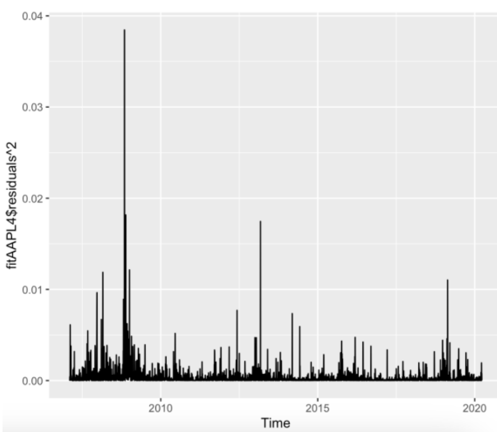
c. Residual Analysis

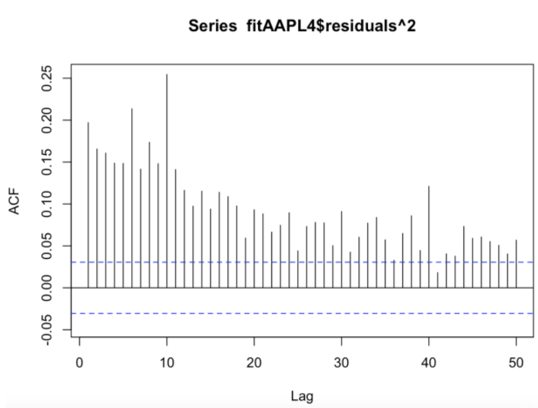
Residuals show much less auto correlation, even though the Ljung Box test still rejects the null hypothesis of independence.

```
> Box.test(fitAAPL4$residuals,lag=10,type = "Ljung")

	Box-Ljung test

data:  fitAAPL4$residuals
X-squared = 9.6536, df = 10, p-value = 0.4714
```



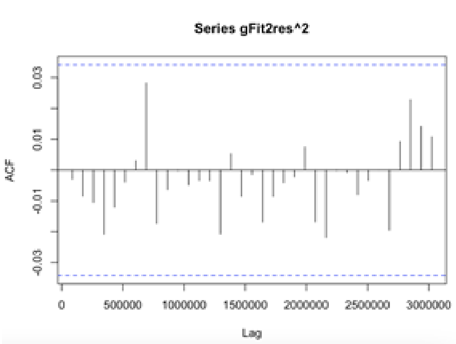
Even though the chosen ARIMA model performed well and presented residuals very close to white noise, there is still some volatility to be modeled as can be seen on the ACF of the squared residuals.





Seeing that GARCH modeling has the capability to reduce volatility, it becomes a strong candidate to address the situation.

A GARCH (1, 1) model was integrated to the ARIMA(0,1,7), to determine whether residual volatility can be brought down to a minimum.

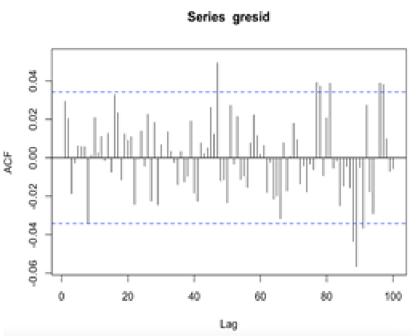


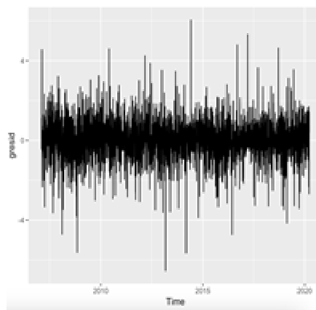

```
> Box.test(gFit2res^2, lag = 15, type = 'Ljung')

	Box-Ljung test

data:  gFit2res^2
X-squared = 7.9799, df = 15, p-value = 0.9246
```

As can be seen on the image below, the coefficients of the model are all significant. It can also be seen on the ACF of the squared residuals that they are now close to white noise. Finally, the Ljung box test fails to reject the null hypothesis of independence.

```
> coeftest(gFit)

z test of coefficients:

     Estimate Std. Error z value  Pr(>|z|)
a0 1.2299e-05 1.3145e-06   9.356 < 2.2e-16 ***
a1 9.4219e-02 8.5995e-03  10.956 < 2.2e-16 ***
b1 8.7282e-01 1.0698e-02  81.584 < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
> Box.test(gresid^2,type = "Ljung")

        Box-Ljung test

data:  gresid^2
X-squared = 0.019766, df = 1, p-value = 0.8882
```



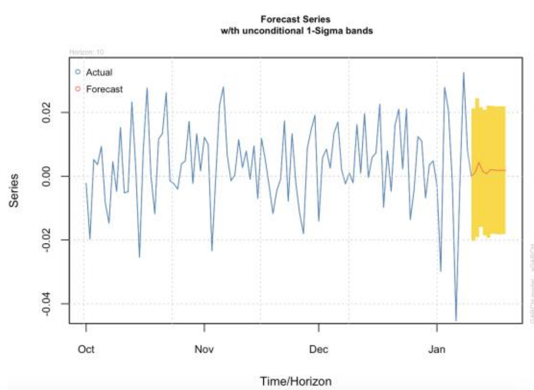


Even though the GARCH (1,1) model performs well, there is still some autocorrelation at high order lags. By integrating the GARCH(1,1) model with our ARMA(0,7) model, we can see on the ACF of the squared residuals that it is finally white noise. The Ljung box test confirms that, by failing to reject the null hypothesis with high confidence.

d.Forecast Analysis

The model had a greater contribution from a moving average process, seeing that it is an MA 7, and performed well, at an MSE of 0.0002489 and MAE of 0.0111900.

```
> report(gFit2.rolltest, type = 'fpm')

GARCH Roll Mean Forecast Performance Measures
---------------------------------------------
Model           : sGARCH
No.Refits       : 3
No.Forecasts: 1295

        Stats
MSE 0.0002489
MAE 0.0111900
DAC 0.5189000
```



2-4. Analysis of Results

Chevron's and Apple's stocks were the two time series chosen for the final report. After conducting a rigorous exploratory data analysis and attempting many different models, the best performing model for each time series was kept. Despite performing well, both ARIMA models still showed a considerable amount of volatility for their residuals, which was addressed by integrating the ARIMA models with GARCH models. The ARMA(2, 2) GARCH(1, 1) for Chevron, and the ARMA(0, 7) GARCH(1, 1) for Apple, ended up with conforming residuals and very low MSE and MAE. Both models successfully normalized the residuals and achieved low MAE values, indicating good predictive performance.

However, it is important to note that relying solely on these models has its limitations. Since the analysis only included technical analysis, it did not incorporate various factors that can influence stock prices, which can have a significant impact on the prediction outcomes. Therefore, more complex and advanced models that capture the influence of multiple factors on prices are needed.

2-5. Appendix A

a. Isabelle Choi

When I first started this project, I conducted an analysis on Walmart stocks. I performed exploratory data analysis of the data of Walmart stock. Firstly, I noticed a multiplicative time series pattern in the plot and took the logarithm of the data. I confirmed the absence of significant seasonality through analysis and tests such as the Dickey Fuller and KPSS tests, which also indicated that the time series had a unit root. Additionally, I observed the presence of drift after differencing the series.

I applied various ARIMA models and obtained ARIMA(0,1,3) and ARIMA(0,1,5) as potential models. Both models passed the L-Jung Box test for independence, but ARIMA(0,1,5) showed better results in the GARCH analysis. Finally, I integrated the ARIMA(0,1,5) and GARCH(1,1) models, generated forecasts, and evaluated the performance using rolling mean forecast measures.

We initially started with four different stocks for the project, but it became too extensive. Therefore, we decided to focus only on Chevron and Apple for the presentation. In the report, my role involved writing the non-technical summary, analyzing the results, and making professional edits to the report. Additionally, for the presentation, I collected and edited members' videos and PowerPoint slides to create a cohesive presentation.

Through this time series course, I had an opportunity to work on a financial project involving stock data and modeling for predictions. It was fascinating to explore the process of modeling and forecasting using stock data. While I acknowledge the limitations of relying solely on technical analysis for stock price prediction due to the various factors affecting stock prices, I believe that such analysis is still essential. Predicting the future is an intriguing task, but it also requires caution and responsibility, as even a small factor can significantly impact the results. This project served as a reminder of the importance of careful analysis and accountability in forecasting.

b. Lucas de Oliveira

I started by exploring the Chevron time series and trying to understand how it behaved. On a first plot I noticed it was, unsurprisingly, a multiplicative time series and took the log of it. I moved on to decomposing the series, which showed me there was no significant seasonality. The series then looked like a random walk with a drift and, to confirm it, I investigated ACF, PACF, EACF plots as well as Dickey Fuller, KPSS tests, which confirmed the series was a unit root. I finally differenced the series used a T test to confirm there was a drift. ACF, PACF, EACF plots, Dickey Fuller and KPSS tests on the differenced series confirmed stationarity and indicated possible models.

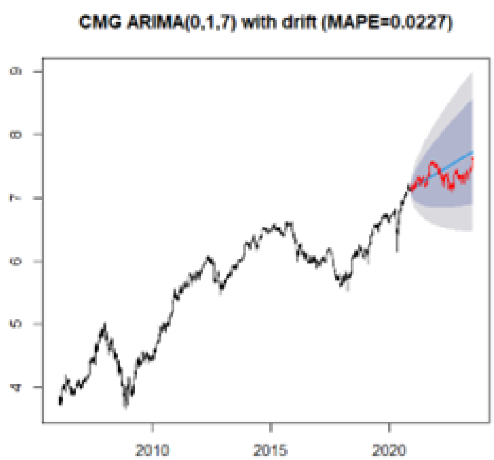
I started with an ARIMA(0, 1, 4) with a drift, checked for coefficient significance and removed the insignificant ones. I then plotted the predictions versus the actual values for the series, followed by performance metrics such as RMSFE and MAPE. I managed to use the back test function for the first model, but it returned errors for all the other models attempted. I then moved on to residual analysis and looked at the plot of the residuals, ACF, PACF and L-Jung box test results. Not only did the L-Jung box test rejected independence of the residuals but the plot and ACF of the squared residuals indicated volatility, which is why I decided to move on to a GARCH model. The GARCH model improved the residuals significantly, but L-Jung box test still rejected independence and some auto correlation could still be seen on the plot and ACF of the squared residuals. I then investigated a few other models by repeating the process described above.

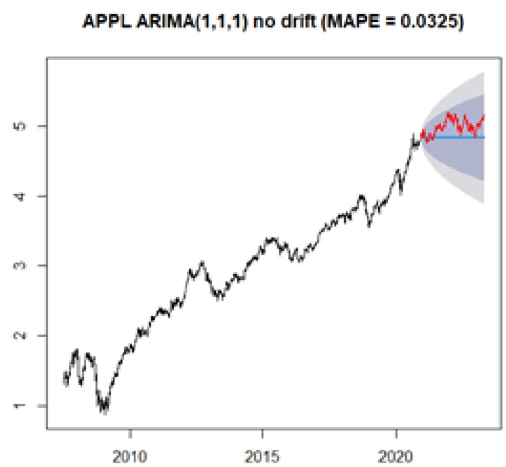
The next one was an AIC based auto ARIMA which resulted in an ARIMA(4, 1, 0) with a drift model. The third model was a BIC based auto ARIMA, which produced no significant coefficients. The fourth model ended up being the best one as it was a parsimonious ARIMA(2, 1, 2) with a drift for whose residuals the L-Jung box test, for the first and only time, failed to reject independence. GARCH produced better results for this model's squared residuals as well, which is why this was the model investigated further. I then kept investigating and tried an ARIMA(0, 1, 2) with a drift and an ARIMA(2, 1, 6) with a drift, to see whether lower or higher order models would produce better results, but none of them outperformed the fourth model built. I then finally integrated the ARIMA(2, 1, 2) and GARCH(1, 1) models using the rugarch function, created a forecast and evaluated the results of roll mean forecast performance measures. Lastly, I also helped coordinate the group's efforts, combined and edited reports, built the presentation material and recorded my own presentation.

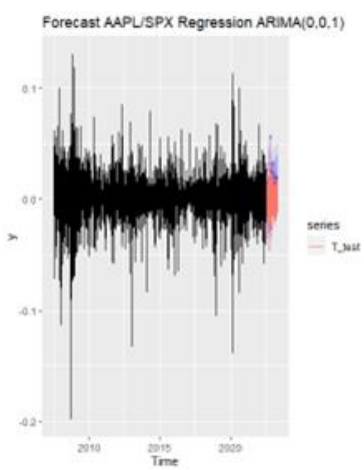
I have had an interest in time series since I tried predicting product demand using decision trees, which turned out to be a huge fiasco. I knew nothing about the topic prior to starting the course, and I really enjoyed learning about it. The fact that adding seasonality to a model can make it so much more accurate, especially when using harmonic regression, was really exciting at first as a big part of why I got into data science is that I like solving problems, coming up with answers, and predicting outcomes. It was also really interesting to see that, sometimes, the residuals of an ARIMA model can behave as an auto regressive process and that, in those cases, regression can be used with ARIMA. GARCH modelling is a powerful technique to reduce volatility and got me more interested in learning about finance.

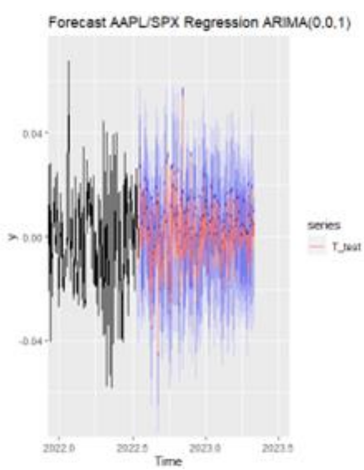
c. Stephen Kim

The three forecasts show the major areas of what I worked on during this project. The first analysis is of Chipotle, and the plot shows a forecast using an ARIMA(0,1,7) model with drift using only the MA(4) and MA(7) terms, along with a 15% test set held out. The second plot is Apple, starting on June29, 2007 when the iPhone debuted, and the best model turned out to be an ARIMA(1,1,1) with no drift. The third plot used the S&P500 (SPX) time series to create a regression analysis on Apple log returns, forecasting on the final 5% of the dataset. I tried to zoom in as much as I could, but overall it looks like the log returns are generally within the 90% confidence interval.









I found it interesting to analyze stocks and have similar practice to that in the classes. I think being able to find real world data that could correlate with stocks would make it more useful, but I had a hard time finding specific datasets like number of computer chips made per month, or things like that. I spent quite a bit of time working on visualizations and trying to format cleaner representations for the audience as well. I think that the time series analysis itself does not necessarily take a long time, but making it useful for others can be a much different feat.

d. Yun Yu

Stock refers to purchasing or selling fractional ownership of equity in an organization, people could invest in a company by buying or selling its stock in the stock market which allows for the seamless exchange. People's economic status could be significantly affected by stock price, as a result, effectively predicting stock trends could aim to increase the profit and reduce the risk. We'll apply time-series analysis on stock price which allows us to find predictive information in data collected over time. I am in charge of the analysis of AAPL stock, first, I'll do the Exploratory Data Analysis to see the data distribution, if it is the additive ts or multiplicative ts, checking the ACF and PACF to see the correlation of the time series, applying differencing to make the time series to be stable and then chose the order for ARIMA model. After choosing the order for the model, I'll separate the model into a training dataset and a testing dataset and apply backtest to see the performance of the model, then I'll check the residual to make sure it is white noise. If we detect something spurious in the residual we infer that it could be a cause of the volatile observations of the financial volatile observation, so we can use the GARCH model to minimize the volatility effect. In the next step, we'll try to integrate the volume variable into our dataset so we can do regression with arima error to see if we can forecast the changes in price based on changes in volumes.

2-6. Appendix B

a. Isabelle Choi

```
wmt <- read.csv("C:/Users/isabe/Downloads/WMT.csv")
head(wmt)
wmt$Date= as.Date(wmt$Date, '%y-%m-%d')

tsWmt = ts(wmt$Adj.Close, start = c(1972,8),  frequency = 250)
autoplot(tsWmt)
logwmt = log(tsWmt)
autoplot(logwmt)
plot(decompose(logwmt))
acf(logwmt)
pacf(logwmt)
Acf(logwmt, lag.max = 30)
eacf(logwmt)
t.test(logwmt)
rewmt = diff(log(tsWmt))
autoplot(rewmt)
acf(rewmt)
pacf(rewmt)
acf(rewmt, lag.max = 20)
pacf(rewmt, lag.max = 20)
plot(decompose(rewmt))
eacf(rewmt)
t.test(rewmt)

adfTest(logwmt, type = 'ct')
# cannot reject the null hypothesis
adfTest(rewmt, type = 'ct')
# can reject the null hypothesis
```

```
adfTest(rewmt, type = 'nc')

kpss.test(logwmt, null = 'Trend')
kpss.test(rewmt, null = 'Trend')
mean(logwmt)
mean(rewmt)
auto.arima(logwmt, ic='aic')
auto.arima(rewmt)
fit1 = Arima(logwmt, order = c(2,0,3), include.drift = TRUE)
coeftest(fit1)

fit2 = Arima(logwmt, order = c(3,0,3), include.drift = FALSE)
coeftest(fit2)
acf(fit2$residuals)
pacf(fit2$residuals)
Box.test(fit2$residuals, type = 'Ljung')

fit3 = Arima(logwmt, order = c(1,1,0), include.drift = TRUE)
coeftest(fit3)
acf(fit3$residuals)
Box.test(fit3$residuals, type = 'Ljung')

fit4 = Arima(logwmt, order = c(1,0,1), include.drift = TRUE)
coeftest(fit4)
acf(fit4$residuals)
Box.test(fit4$residuals, type = 'Ljung')

fit5 = Arima(logwmt, order = c(0,1,3), include.drift = TRUE)
fit5 = Arima(logwmt, order = c(0,1,3), include.drift = TRUE, fixed = C(NA,0,NA,NA))
coeftest(fit5)
```

```
acf(fit5$residuals)
pacf(fit7$residuals)
Box.test(fit5$residuals, type = 'Ljung')
forecast(fit5) %>% autoplot()

fit7 = Arima(logwmt, order = c(0,1,5), include.drift = TRUE)
fit7 = Arima(logwmt, order = c(0,1,5), include.drift = TRUE, fixed = c(NA,0,NA,0,NA,NA))
coeftest(fit7)
acf(fit7$residuals)
pacf(fit7$residuals)
Box.test(fit3$residuals, type = 'Ljung')
forecast(fit5) %>% autoplot()

ntest = length(logwmt) * .8
backtest(fitb, logwmt, ntest, 1)

gFit <- garch(logwmt, order = c(1,1))
gFit
coeftest(gFit)
autoplot(gFit$residuals)
Acf(gFit$residuals)
Box.test(gFit$residuals, type = 'Ljung')
```

b. Lucas de Oliveira

```
# Subset by decade and check how coefficients compare for model m3
# Try to have GARCH model capture seasonality in the residuals
# Ljung box will always reject independence on garch residuals
# Try garch extension for leverage effects.
# Use garchfit to have it all in one model.
# Test normality of residuals
```

```
chevron <- read.csv('CVX.csv')
head(chevron)
chevron.adj.close.ts <- ts(chevron$Adj.Close,
            start = c(1962, 1),
            frequency = 252)
# Log transformation is required to make it additive
autoplot(chevron.adj.close.ts)
chevron.log <- log(chevron.adj.close.ts)
# Trend stationary or random walk with a drift?
autoplot(chevron.log)
# Checking length of series
length(chevron.log)

# Splitting into train and test datasets
chevron.train <- subset(chevron.log, end = 13900)
chevron.test <- subset(chevron.log, start = 13901)

# Decomposing
autoplot(decompose(chevron.train))

# These look like random walk, but that might be just because of the trend
acf(chevron.train)
pacf(chevron.train)

# Unit root
eacf(chevron.train)

# Dickey fuller barely rejects the null hypothesis of unit root with a trend
# (both at lag 10 and 15), but kpss rejects the null hypothesis of stationarity
```

```
# with a trend. This could be a random walk with drift, and I will start
# exploring it as such.
adfTest(chevron.train, lags = 10, type = 'ct')
adfTest(chevron.train, lags = 15, type = 'ct')
kpss.test(chevron.train, null = 'Trend')

# Now it definitely looks stationary
autoplot(diff(chevron.train))

# T test rejects the null hypothesis, meaning the sample mean and the mean of the
# population differ, indicating a drift.
t.test(diff(chevron.log))

# Dickey fuller  now definitely rejects the null hypothesis of unit root and
# KPSS fails to reject the null hypothesis of stationarity.
adfTest(diff(chevron.train), lags = 10, type = 'nc')
adfTest(diff(chevron.train), lags = 15, type = 'nc')
kpss.test(diff(chevron.train), null = 'Level')

# No apparent AR behavior and some auto correlation after lag 0.
acf(diff(chevron.train))

# Looks like a high order MA considering that there does not seem to be seasonality.
# Looking at it up close it is much clearer that there is auto correlation after lag 0.
pacf(diff(chevron.train))

# This is unclear. Maybe an MA(4)?
eacf(diff(chevron.train))

# Trying a first model
```

```
m0 <- Arima(chevron.train, order = c(0,1,4), include.drift = TRUE)

# MA 4 and drift are significant, as expected, and MA2, surprisingly, is also
# significant
coeftest(m0)

# Removing insignificant terms from first model
m0 <- Arima(chevron.train, order = c(0,1,4), include.drift = TRUE,
        fixed = c(0, NA, 0, NA, NA))

coeftest(m0)

# Visually it seems ok.
fc1 <- forecast(m0, h = length(chevron.test))
plot(fc1)
lines(chevron.test, col = 'red')

# RMSFE of .22
rmsfe.m0 <- sqrt(mean((fc1$mean - chevron.test)^2))
rmsfe.m0

# About 3.5%, not bad.
mape.m0 <- mean(abs((fc1$mean - chevron.test)/chevron.test))
mape.m0
bt0 <- backtest(m0, chevron.log, orig = .8* length(chevron.log), h = 1)
# Residuals are definitely stationary, but there's some volatility.
autoplot(m0$residuals)

# Looks very close to white noise, except for a few instances of auto correlation
acf(m0$residuals)
```

```
# Looking at it closer there is auto correlation at four different lags
pacf(m0$residuals)

# L-Jung box test barely fails to reject the null hypothesis of independence
Box.test(m0$residuals, lag = 10, type = 'Ljung')

# Square residuals show a lot of volatility
autoplot(m0$residuals^2)
acf(m0$residuals^2)

res <- m0$residuals

# Trying a garch model instead as, judging by the pacf of the squared residuals
# it would take a really high order arch model to get rid of auto correlation.
# The coefficients are all very significant
garch.fit <- garch(res, order = c(1,1))
garch.fit
coeftest(garch.fit)

# These residuals are looking better but there is still some auto correlation
autoplot(garch.fit$residuals^2)
Acf(garch.fit$residuals^2)

# Ljung box test definitely rejects the null hypothesis
Box.test(garch.fit$residuals, type = 'Ljung')

# If garch didn't manage to make the residuals white noise, let's increase the
# order of the arima/try different models.
```

```
#=========================================================================
=========
# Checking on what auto.arima comes up with
m1 <- auto.arima(chevron.train)

# AR(2) and drift are very significant, AR(1) not so much.
coeftest(m1)

# Keeping only significant terms
m1 <- Arima(chevron.train, order = c(2, 1, 0), include.drift = TRUE,
        fixed = c(0, NA, NA))

coeftest(m1)

# Visually it seems ok.
fc2 <- forecast(m1, h = length(chevron.test))
plot(fc2)
lines(chevron.test, col = 'red')

# RMSFE of .22, same as m0
rmsfe.m1 <- sqrt(mean((fc2$mean - chevron.test)^2))
rmsfe.m1

# About 3.5%, same as m0.
mape.m1 <- mean(abs((fc2$mean - chevron.test)/chevron.test))
mape.m1

# "Error in if (orig > T) orig = T : the condition has length > 1"
bt1 <- backtest(m1, chevron.log, orig = .8* length(chevron.log), h = 1)
```

```
# Residuals are definitely stationary, but there's some volatility.
autoplot(m1$residuals)

# Looks very close to white noise, except for a few instances of auto correlation
acf(m1$residuals)

# Looking at it closer there is auto correlation at four different lags
pacf(m1$residuals)

# L-Jung box test rejects the null hypothesis of independence
Box.test(m1$residuals, lag = 10, type = 'Ljung')

# Square residuals show a lot of volatility
autoplot(m1$residuals^2)
acf(m1$residuals^2)

res <- m1$residuals

# Trying a garch model instead as, judging by the pacf of the squared residuals
# it would take a really high order arch model to get rid of auto correlation.
# The coefficients are all very significant
garch.fit <- garch(res, order = c(1,1))
garch.fit
coeftest(garch.fit)

# These residuals are looking better but there is still some auto correlation
autoplot(garch.fit$residuals^2)
Acf(garch.fit$residuals^2)

# Ljung box test definitely rejects the null hypothesis
```

```
Box.test(garch.fit$residuals, type = 'Ljung')
#===========================================================================
=========
# Checking auto.arima using bic.
m2 <- auto.arima(chevron.train, ic = 'bic')
m2

# It seems to not return any coefficients!
coeftest(m2)
#===========================================================================
=========
# This is the best model. RMSFE and MAPE are about the same as all other ones,
# but residuals are much better behaved. Not to mention this one is more parsimonious
# than many other ones I tried.
# Trying a higher order model

# SUBSET BY DECADE AND CHECK ON COEFFICIENTS.
m3 <- Arima(chevron.train, order = c(2,1,2), include.drift = TRUE)

# All very significant!
coeftest(m3)

# Visually it seems ok.
fc3 <- forecast(m3, h = length(chevron.test))
plot(fc3)
lines(chevron.test, col = 'red')

# RMSFE of .22
rmsfe.m3 <- sqrt(mean((fc3$mean - chevron.test)^2))
rmsfe.m3
```

```
# About 3.5%, not bad.
mape.m3 <- mean(abs((fc3$mean - chevron.test)/chevron.test))
mape.m3

# "Error in solve.default(res$hessian * n.used, A) :
# Lapack routine dgesv: system is exactly singular: U[1,1] = 0"
bt3 <- backtest(m3, chevron.log, orig = .8* length(chevron.log), h = 1)

# Residuals are definitely stationary, but there's some volatility.
autoplot(m3$residuals)

# Looks very close to white noise, except for a few instances of auto correlation
acf(m3$residuals)

# Residuals seem better than the ones for the previous models!
pacf(m3$residuals)

# L-Jung box test fails to reject the null hypothesis of independence by quite
# a lot! Finally!
Box.test(m3$residuals, lag = 10, type = 'Ljung')

# TRY TO HAVE GARCH MODEL SEASONALITY
# Square residuals show a lot of volatility
autoplot(m3$residuals^2)
acf(m3$residuals^2)

res <- m3$residuals

# Trying a garch model instead as, judging by the pacf of the squared residuals
# it would take a really high order arch model to get rid of auto correlation.
```

```
# The coefficients are all very significant
garch.fit <- garch(res, order = c(1,1))
garch.fit
coeftest(garch.fit)

# These residuals are looking better but there is still some auto correlation
autoplot(garch.fit$residuals^2)
Acf(garch.fit$residuals^2)

# This is fine! Try extension for leverage effects.
# Use garchfit to have it all in one model.
# Ljung box test still definitely rejects the null hypothesis, but again, closer
# to failing to reject than other models
Box.test(garch.fit$residuals^2, type = 'Ljung')

# Evaluate normality of residuals here
res <- na.omit(garch.fit$residuals)
ggplot(res, aes(x=res)) + geom_histogram()

ggplot(res, aes(sample=res)) +
  stat_qq() +
  stat_qq_line()

jarque.bera.test(na.omit(res)) # rejects normality

# GARCH modelling
# garchfit produces weird results, stick with ugarch
gfit3 = garchFit( ~ arma(2, 2) + garch(1, 1), data = diff(chevron.train), trace = F)
gfit3 # AR2 and MA2 are insignificant now.
```

```
resgfit3 <- gfit3@residuals/gfit3@sigma.t
autoplot(ts(resgfit3^2))
Acf(resgfit3^2)
Box.test(resgfit3^2, lag = 15, type = 'Ljung')

# This looks better
s = ugarchspec(variance.model=list(garchOrder=c(1, 1)),
               mean.model=list(armaOrder=c(2, 2)))

gfit4 <- ugarchfit(s, diff(chevron.log))

# Everything is significant. Robust AR2, MA2, omega and alpha are not significant.
gfit4@fit
gfit4res <- residuals(gfit4, standardize=T)
autoplot(gfit4res^2)
acf(gfit4res^2)

# Almost fails to reject independence at a 95% confidence interval! .04
Box.test(gfit4res^2, lag = 15, type = 'Ljung')

ggplot(gfit4res, aes(x=gfit4res)) + geom_histogram()

ggplot(gfit4res, aes(sample=gfit4res)) +
  stat_qq() +
  stat_qq_line()

skewness(gfit4res) # Very small skewness, -0.15
kurtosis(gfit4res) # Not bad, 2.19

jarque.bera.test(na.omit(gfit4res)) # rejects normality
```

```
gfit4.rolltest <- ugarchroll(s, data = diff(chevron.log), n.start = 2000,
                             refit.window = 'moving',
                             refit.every = 500)

report(gfit4.rolltest, type = 'fpm')
plot(ugarchforecast(gfit4, n.ahead=10))
#======================================================================
=========

# Trying a lower order model
m4 <- Arima(chevron.train, order = c(0,1,2), include.drift = TRUE)

# MA(2) and drift are significant
coeftest(m4)

# Visually it seems ok.
fc4 <- forecast(m4, h = length(chevron.test))
plot(fc4)
lines(chevron.test, col = 'red')

# RMSFE of .22
rmsfe.m4 <- sqrt(mean((fc4$mean - chevron.test)^2))
rmsfe.m4

# About 3.5%, not bad.
mape.m4 <- mean(abs((fc4$mean - chevron.test)/chevron.test))
mape.m4

# "Error in if (orig > T) orig = T : the condition has length > 1"
```

```
bt4 <- backtest(m4, chevron.log, orig = .8* length(chevron.log), h = 1)

# Residuals are definitely stationary, but there's some volatility.
autoplot(m4$residuals)

# Looks very close to white noise, except for a few instances of auto correlation
acf(m4$residuals)

# hmm
pacf(m4$residuals)

# L-Jung box test rejects the null hypothesis of independence
Box.test(m4$residuals, lag = 10, type = 'Ljung')

# Square residuals show a lot of volatility
autoplot(m4$residuals^2)
acf(m4$residuals^2)

res <- m4$residuals

# Trying a garch model instead as, judging by the pacf of the squared residuals
# it would take a really high order arch model to get rid of auto correlation.
# The coefficients are all very significant
garch.fit <- garch(res, order = c(1,1))
garch.fit
coeftest(garch.fit)

# These residuals are looking better but there is still some auto correlation
autoplot(garch.fit$residuals^2)
Acf(garch.fit$residuals^2)
```

```
# Ljung box test still definitely rejects the null hypothesis
Box.test(garch.fit$residuals, type = 'Ljung')

#=========================================================================
=========
# Trying a lower order model
m5 <- Arima(chevron.train, order = c(2,1,6), include.drift = TRUE)

# AR(2), MA(2), MA(6) and drift are significant
coeftest(m5)

# Removing insignificant terms from first model
m5 <- Arima(chevron.train, order = c(2,1,6), include.drift = TRUE,
            fixed = c(0, NA, 0, NA, 0, 0, 0, NA, NA))

# Visually it seems ok.
fc5 <- forecast(m5, h = length(chevron.test))
plot(fc5)
lines(chevron.test, col = 'red')

# RMSFE of .22
rmsfe.m5 <- sqrt(mean((fc5$mean - chevron.test)^2))
rmsfe.m5

# About 3.5%, not bad.
mape.m5 <- mean(abs((fc5$mean - chevron.test)/chevron.test))
mape.m5

# "Error in if (orig > T) orig = T : the condition has length > 1"
```

```
bt5 <- backtest(m5, chevron.log, orig = .8* length(chevron.log), h = 1)

# Residuals are definitely stationary, but there's some volatility.
autoplot(m5$residuals)

# Looks very close to white noise, except for a few instances of auto correlation
acf(m5$residuals)

# Looks a little better than most models
pacf(m5$residuals)

# L-Jung box test fails to rejects the null hypothesis of independence, but barely
Box.test(m5$residuals, lag = 10, type = 'Ljung')

# Square residuals show a lot of volatility
autoplot(m5$residuals^2)
acf(m5$residuals^2)
res <- m5$residuals

# Trying a garch model instead as, judging by the pacf of the squared residuals
# it would take a really high order arch model to get rid of auto correlation.
# The coefficients are all very significant
garch.fit <- garch(res, order = c(1,1))
garch.fit
coeftest(garch.fit)

# These residuals are looking better but there is still some auto correlation
autoplot(garch.fit$residuals^2)
Acf(garch.fit$residuals^2)
```

```
# Ljung box test still definitely rejects the null hypothesis
Box.test(garch.fit$residuals, type = 'Ljung')
```

==
========

```
# As nothing seems to work with an ARIMA model, I'll try an ARMA model instead.
# Maybe it is trend stationary after all.
m6 <- Arima(chevron.train, order = c(0,0,4), include.drift = TRUE)

# All very significant
coeftest(m6)

# Visually not as appropriate as all previous models.
fc6 <- forecast(m6, h = length(chevron.test))
plot(fc6)
lines(chevron.test, col = 'red')

# RMSFE of .35
rmsfe.m6 <- sqrt(mean((fc6$mean - chevron.test)^2))
rmsfe.m6

# About 6.4%, almost twice the error of all other models.
mape.m6 <- mean(abs((fc6$mean - chevron.test)/chevron.test))
mape.m6

# "Error in if (orig > T) orig = T : the condition has length > 1"
bt6 <- backtest(m6, chevron.log, orig = .8* length(chevron.log), h = 1)

# Residuals are definitely NOT stationary. I'll stop here.
autoplot(m6$residuals)
```

c. Stephen Kim

```
# Load file
apple = read.csv('AAPL.csv')
head(apple)
class(apple$Date)
#Format Date
apple$Date = ymd(apple$Date)
head(apple)
tail(apple)
class(apple$Date)

# Subset for when iphone debuted June 29, 2007
apple = apple[6699:10692, ]
head(apple)
tail(apple)

# Plot as Additive Series
length(apple$Adj.Close) # 3994 trading days
appleTS = ts(apple$Adj.Close, start = c(2007, 124), frequency = 252)
autoplot(appleTS, ylab = 'Price', main = 'Apple Inc. (AAPL) - Additive')

# Multiplicative series
appleLogTS = ts(log(apple$Adj.Close), start = c(2007, 124), frequency = 252)

autoplot(appleLogTS, ylab = 'Price', main = 'Apple Inc. (AAPL) - Multiplicative')
# Use loess smoothing to see trend
# Create data frame for adjusted close prices and dates
price = log(apple$Adj.Close)
date = apple$Date
appleDS = data.frame(price, date)
```

```
# Create loess object with 10% weight
loess10 = loess(price ~ as.numeric(date), data = appleDS, span=0.10)
smoothed10 = predict(loess10)
plot(price, type="l")
lines(smoothed10, col="red", lwd=2)

# Plot the LOESS smooth line over the price lines
ggplot(data=appleDS, aes(x=date, y=price)) + geom_line() + geom_smooth(col="red") +
  ggtitle('Apple Inc. (AAPL) - Multiplicative w/ LOESS Smoothing')
# Create decomposition plot
# Extremely tiny seasonality, residuals can get pretty big, and possible trend and maybe drift
autoplot(decompose(appleLogTS))
# Plot the acf, pacf, and eacf
Acf(appleLogTS, lag.max = 50)    # non-stationary
pacf(appleLogTS, lag.max = 50)   # only lag-1
eacf(appleLogTS)                 # Looks like ARMA(1,1) or ARMA(2,1)

# Ljung Box test
Box.test(appleLogTS,lag=6,type='Ljung')   # not white noise

# Test for unit root stationarity with Dickey-Fuller and KPSS tests
adfTest(appleLogTS, lags = 15, type = 'ct') # 0.1984 non-stationary
adfTest(appleLogTS, lags = 15, type = 'c')  # 0.9206
adfTest(appleLogTS, lags = 15, type = 'nc') # 0.99
kpss.test(appleLogTS, null = 'Trend')   # 0.01 non-stationary
kpss.test(appleLogTS, null = 'Level')   # 0.01
# Take differences to get the log-returns
appRet = diff(appleLogTS)
# Analyze the mean
mean(appRet)         # 0.0009630812
```

```
t.test(appRet)       # p-value 0.01171 mean is not 0
# Plot the returns
autoplot(appRet, main = 'AAPL Log-Returns')  #heteroschedastic
# Analyze the autocorrelation structure
Acf(appRet, lag.max = 50)     # Autocorrelation at Lags 1,4,6,7,8,9,12,15,16
pacf(appRet, lag.max = 50)    # biggest autocorrelation at lag-16?
eacf(appRet)                  # MA(1)? ARMA(1,1)
Box.test(appRet, lag=15, type="Ljung")          # reject white noise
# Check stationarity with Dickey-Fuller and KPSS tests
adfTest(appRet, type="ct")          # 0.01 stationary
adfTest(appRet, type="nc")          # 0.01
adfTest(appRet, type="c")           # 0.01
kpss.test(appRet, null = 'Trend')      # 0.1 stationary

# Indicating stationary
kpss.test(appRet, null = 'Level')      # 0.1
#Build ARIMA model (1,1) no drift
appleFit1 = Arima(appRet, order=c(1, 1, 1), include.drift = F)
appleFit1
coeftest(appleFit1)          # AR1 and MA1 significant
#Build ARIMA model (1,1) with drift
appleFit1 = Arima(appRet, order=c(1, 1, 1), include.drift = T)
appleFit1
coeftest(appleFit1)          # drift not significant
#Build MA model (1) with drift
appleFit2 = Arima(appRet, order=c(0, 0, 1), include.drift = T)
appleFit2
coeftest(appleFit2)          # drift and intercept not significant

#Build MA model (1) no drift
```

```
appleFit2 = Arima(appRet, order=c(0, 0, 1), include.drift = F)
appleFit2
coeftest(appleFit2)          # intercept and MA(1) term significant
#Build best ARIMA model (1,1,1) no drift
appleFitX = Arima(appRet, order=c(1, 1, 1), include.drift = F)
appleFitX
coeftest(appleFitX)

# Build Arima model with backtesting
# Try backtest function on log returns
ntest = 0.80*length(appRet)
appleFitX
Acf(appRet, lag.max = 50)
Acf(appleFitX$residuals, lag.max = 50)    # autocorrelation at lags 4, 6, 7, 12, 15, 16
bt1 = backtest(appleFitX, appRet, orig = ntest, h=1)   # error
bt1
# Backtest with train85/test15 datasets
length(appleLogTS)   # 3994 entries
3994 * 0.85
3994 * 0.15
appTrain = subset(appleLogTS, end = 3394)
autoplot(appTrain)
appTest = subset(appleLogTS, start = 3395)
autoplot(appTest)
Acf(appTrain, lag.max = 50)     # slow decay
pacf(appTrain, lag.max = 50)    # lag-1 correlation
appTrain
# Fit ARIMA(1,1,1) model no drift
length(appTest)
fit1BT = Arima(appTrain, order=c(1, 1, 1), include.drift = F)
```

```
#fixed = c(NA, 0, NA, NA, 0))
fit1Future = forecast(fit1BT, h = 599)
plot(fit1Future, main = 'AAPL ARIMA(1,1,1) no drift (MAPE = 0.0325)')
lines(appTest, col = 'red')
# Get MAPE
mape_app1 = mean(abs((fit1Future$mean - appTest)/appTest))
mape_app1
# Check with auto.arima and backtesting
fitAIC = auto.arima(appTrain, ic = 'aic')
fitAIC          # ARIMA (5,1,2) with drift
coeftest(fitAIC)
fitAICFuture = forecast(fitAIC, h = 599)

plot(fitAICFuture, main = 'AAPL auto.arima(AIC) - ARIMA(5,1,2) w/ drift (MAPE = 0.0332)')
lines(appTest, col = 'red')
# Get MAPE
mape_app2 = mean(abs((fitAICFuture$mean - appTest)/appTest))
mape_app2
# Check with auto.arima and backtesting
fitBIC = auto.arima(appTrain, ic = 'bic')
fitBIC          # ARIMA (0,1,0) with drift
coeftest(fitBIC)
fitBICFuture = forecast(fitBIC, h = 599)
plot(fitBICFuture, main = 'AAPL auto.arima(BIC) - ARIMA(0,1,0) w/ drift (MAPE = 0.0327)')
lines(appTest, col = 'red')
# Get MAPE
mape_app3 = mean(abs((fitBICFuture$mean - appTest)/appTest))
mape_app3
# Load data
```

```
spx = read.csv('SPX.csv')
head(spx)
length(spx$Adj.Close)
spx = spx[19961:23954,]
length(spx$Adj.Close)
head(spx)
 # Create time series for spx matching apple timeframe
spxTS = ts(log(spx$Adj.Close), start=c(2007, 124), frequency=252)
autoplot(spxTS, main = 'SPX')
# Plot time series in one plot
stocks <- cbind(appleLogTS, spxTS) %>%
  ts(start = c(2007, 124), frequency = 252)
autoplot(stocks, main = 'Apple/S&P 500')
# Create lag plots of apple and spx
lag2.plot(appleLogTS, spxTS, 8, main = 'AAPL/SPX')
# Create cross-correlation plots for each
ccf(diff(appleLogTS), diff(spxTS), main = 'Apple/SPX')
# Compute regression of apple on spx
predictors <- cbind(
  stats::lag(biscTS, -1),
  stats::lag(pbTS,0))
spxRet = diff(spxTS)
Acf(appleLogTS, lag.max = 20)
Acf(diff(appleLogTS), lag.max = 20)
Acf(appRet, lag.max = 20)
Acf(spxRet, lag.max = 20)
appZoo = as.zoo(ts.intersect(appRet, spx=stats::lag(spxRet, -1)))
Acf(appZoo)
fitX1 = Arima(appZoo$appRet, xreg=(tpZoo$spxRet),
```

```
            order=c(0, 0, 1))
summary(fitX1)
coeftest(fitX1)
plot(fitX1)
# Check autocorrelation in the residuals
acf(fitX1$residuals, lag.max = 50)
Box.test(fitX1$residuals, type = 'Ljung-Box')
# Check auto arima
fitX2 = auto.arima(appZoo$appRet, xreg = (appZoo$spxRet))
fitX2
summary(fitX2)
coeftest(fitX2)
# Check autocorrelation in the residuals
acf(fitX2$residuals)
Box.test(fitX2$residuals, type = 'Ljung-Box')
length(appRet)
length(spxRet)
# Time series regression
fitR = lm(appRet ~ spxRet)
summary(fitR)
plot(fitR$residuals, type="l")
Acf(fitR$residuals)
pacf(fitR$residuals)
eacf(fitR$residuals)
# Now, let's do this again on the copies with holdouts and compare
# using models that have ARMA errors
f1 = Arima(appRet, xreg=spxRet, order=c(2, 0, 2))   # same as "lm"
f1   # Sigma^2 = 0.000221      AIC - -22273.9
coeftest(f1)
f2 = Arima(appRet, xreg=spxRet, order=c(3, 0, 3))
```

```
f2   # Sigma=^2 = 0.000221     AIC - -22272.11
coeftest(f2)
f3 = Arima(appRet, xreg=spxRet, order=c(0, 0, 4))
f3   # Sigma=^2 = 0.000221     AIC - -22274.45
coeftest(f3)
plot(f3$residuals)
acf(f3$residuals)   # looks like white noise
Box.test(f3$residuals, lag=10, type="Ljung")   # Cannot reject WN
f4 = auto.arima(appRet, xreg=spxRet)  # auto.arima finds a simpler model
f4  # ARIMA(2,0,0)   AIC - -22268.15
coeftest(f4)
# Let's try backtesting these
n = length(series2)
b1 = backtest(f1, appRet, xre=spxRet, h=1, orig=.8*n)
b2 = backtest(f2, appRet, xre=spxRet, h=1, orig=.8*n)
b3 = backtest(f3, appRet, xre=spxRet, h=1, orig=.8*n)
b4 = backtest(f4, appRet, xre=spxRet, h=1, orig=.8*n)
b1$rmse   # 0.01544898
b2$rmse   # 0.0157939
b3$rmse   # 0.01552617
b4$rmse   # 0.01502045
autoplot(spxRet)
autoplot(appRet)
length(spxRet)-399
plot(forecast(f1, xreg=appRet[3594:3993]))
lines(3594:3993, as.numeric(spxRet[3594:3993]), col="red")
##################################################################
qplot(appRet, spxRet)   # somewhat linear relationship
```

```
# 10% test set
n = length(appRet)
n * .95
T_train = subset(spxRet, end=3793)
D_train = subset(appRet, end=3793)
T_test = subset(spxRet, start=3794)
D_test = subset(appRet, start=3794)
# Try a naive regression
x1 = Arima(D_train, xreg=T_train, order=c(0, 0, 0))
x1   # AIC - -19852.2
coeftest(x1)   # significant xreg
autoplot(x1$residuals)   #
Acf(x1$residuals, lag.max = 100)      # Lag-1
pacf(x1$residuals)      # lag-1
eacf(x1$residuals)      # Perhaps ARMA(0, 1) or ARMA(1,1)
frequency(T_train)
x2 = Arima(D_train, xreg=T_train, order=c(0, 0, 1))
x2      # AIC - -19854.97
coeftest(x2)  # All are significant
plot(x2$residuals)  #
Acf(x2$residuals, lag.max = 100)   # lag-2, 4
# Try an AIC optimized model
xa = auto.arima(D_train, xreg=T_train)
xa   #  ARIMA(1,0,0) AIC - -19854.59
# Try a BIC optimized model
xb = auto.arima(D_train, xreg=T_train, ic="bic")
xb   # ARiMA(0,0,0) AIC - -19846.54
acf(xb$residuals)
# Let's compare with backtesting
n = length(D_train)
```

```
n
b1 = backtest(x1, D_train, xre=T_train, h=1, orig=.8*n)
b2 = backtest(x2, D_train, xre=T_train, h=1, orig=.8*n)
b3 = backtest(xa, D_train, xre=T_train, h=1, orig=.8*n)  # Takes a while
b4 = backtest(xb, D_train, xre=T_train, h=1, orig=.8*n)  # ditto
print("MAPE model comparison")
print(paste("OLS:", round(b1$mape, 3), sep=" "))
print(paste("Manual:", round(b2$mape, 3), sep=" "))
print(paste("AIC:", round(b3$mape, 3), sep=" "))
print(paste("BIC:", round(b1$mape, 3), sep=" "))
#Forecast
xf = forecast(x2, xreg=T_test)
subset(xf, start=2000)
autoplot(xf) + autolayer(T_test)
lines(as.numeric(time(D_test)), as.numeric(D_test), col="red")
autoplot(appRet)
# zoom in with ggplot
attr(xf$x, "tsp")
# [1] 1949.000 1960.917   12.000
# 2007.492 2021.750  252.000
g1 <- autoplot(xf) +
  ggtitle("Forecast AAPL/SPX Regression ARIMA(0,0,1)") +
  ylab("y") + autolayer(T_test)

g2 <- autoplot(xf) +
  ggtitle("Forecast AAPL/SPX Regression ARIMA(0,0,1)") +
  ylab("y") +
  coord_cartesian(xlim = c(2022, 2023.5), ylim = c(-0.07, 0.07)) +
  autolayer(T_test)
grid.arrange(g1, g2, ncol=2)
```

d. Yun Yu

```
#readdata
str(AAPL)
AAPL<- read_csv("Downloads/HistoricalPrices.csv")

 # transpose of dataframe
transpose <- t(AAPL)
# converting the result to dataframe
transpose <- as.data.frame(transpose)
# calculating reverse of dataframe
rev_data_frame <- rev(transpose)
# transpose of reverse dataframe
rev_data_frame <- t(rev_data_frame)
# converting the result to dataframe
rev_data_frame <- as.data.frame(rev_data_frame)
AAPLts <- ts(rev_data_frame$Close,start = 2007.1, frequency = 251)

AAPLts1 <- ts(rev_data_frame$Volume,start = 2007.1, frequency = 251)
plot(decompose(AAPLtslog))
autoplot(AAPLts1)
rev_data_frame$Close<- as.numeric(rev_data_frame$Close)
rev_data_frame$Volume<- as.numeric(rev_data_frame$Volume)
v0 <- ts(log(rev_data_frame$Close),start = 2007.1, frequency = 251)
v1 <- ts(log(rev_data_frame$Volume),start = 2007.1, frequency = 251)
s = as.zoo(ts.intersect(v0, v1lag=stats::lag(v1, -822)))
autoplot(s)    # At the beginning of the plot, we can now see the correlation
# When soi is high, rec is low and vice-versa
0.4*length(v1) - 1645 + 1  # The last 6 elements of soi are not used in the training
# set, so we can use them to forecast "rec" out beyond
v1Test = subset(v1, start = length(v1) - 411 + 1)
```

```
# Notice that a zoo is a little more convenient because we can use $!!
fit2 = Arima(s$v0, xreg=s$v1, order=c(0, 0, 0))
fit2    # Same slope and intercept
# Now, let's run the forecast.
plot(forecast::forecast(fit2, xreg=v1Test), xlim=c(2007, 2023))
v0 <- ts(log(rev_data_frame$Close),start = 2007.1, frequency = 251)
v1 <- ts(log(rev_data_frame$Volume),start = 2007.1, frequency = 251)
v = ts.intersect(v0, v1)
v0 = v[, "v0"]
v1 = v[, "v1"]
v1_validation = subset(v1, start=3289)
v1_validation
fit3 = Arima(subset(v1, start=412), xreg=subset(v0, end=3702), order=c(0, 0, 0))
plot(forecast::forecast(fit3, xreg=subset(v0, start=3703)))
lines(4114:4526, v1_validation, col="red")
length(v1)
length(v1_validation)
############################################################################
#AAPL <- read_csv("Desktop/AAPL.csv")
AAPL$Date=as.Date(AAPL$Date,"%y/%m/%d")
head(AAPL)
AAPLts <- ts(AAPL$Close,start = 2007.1, frequency = 251)
#data normality
qplot(AAPLts,geom="histogram",main=" Stock price")
qqnorm(AAPLts)
qqline(AAPLts)
#We check the adfTest, we see there is a unit root that we cannot reject.
autoplot(AAPLts)
adfTest(AAPLts,type="ct")
adfTest(AAPLts,type="nc")
```

```
adfTest(AAPLts,type="c")
kpss.test(AAPLts)
#log
AAPLtslog <- ts(log(AAPL$`Adj Close`),start = 2007.1, frequency = 251)

#check adf test if there is any unit root

adfTest(AAPLtslog,type="ct")

adfTest(AAPLtslog,type="nc")

adfTest(AAPLtslog,type="c")

kpss.test(AAPLtslog)
autoplot(AAPLtslog )

acf(AAPLtslog )

pacf(AAPLtslog)

eacf(AAPLtslog)
#try differencing the series to de-trend

AAPLtslogdiff<- diff(AAPLtslog)
autoplot(AAPLtslogdiff)
t.test(diff(AAPLtslog))
adfTest(AAPLtslogdiff,type="ct")
adfTest(AAPLtslogdiff,type="nc")
adfTest(AAPLtslogdiff,type="c")
kpss.test(AAPLtslogdiff)
```

```
# Difference the model
autoplot(AAPLtslogdiff )
acf(diff(AAPLtslog))
pacf(diff(AAPLtslog))
eacf(AAPLtslogdiff)
# fit model according to the results above.

#1. seperate model as training dataset and testing dataset.
#Ttry (0,1,4)
fitAAPL1<-Arima(AAPLtslog, order = c(0,1,4),include.drift = TRUE)
coeftest(fitAAPL1)
fitAAPL2<-Arima(AAPLtslog, order = c(0,1,4),include.drift = TRUE,fixed = c(0,0,0,NA,NA))
coeftest(fitAAPL2)
summary(fitAAPL2)
#Check the residual
acf(fitAAPL2$residuals)
Box.test(fitAAPL2$residuals,lag=10,type = "Ljung")
#Ttry (0,1,7)
fitAAPL3<-Arima(AAPLtslog, order = c(0,1,7),include.drift = TRUE)
coeftest(fitAAPL3)
fitAAPL4<-Arima(AAPLtslog, order = c(0,1,7),include.drift = TRUE,fixed = c(NA,NA,0,NA,0,NA,NA,NA))
coeftest(fitAAPL4)
summary(fitAAPL4)
#Check the residual(0,1,7)
Acf(fitAAPL4$residuals,lag=100)
Box.test(fitAAPL4$residuals,lag=10,type = "Ljung")
#Ttry (2,1,4)
fitAAPL5<-Arima(AAPLtslog, order = c(2,1,4),include.drift = TRUE)
```

```
coeftest(fitAAPL5)

fitAAPL6<-Arima(AAPLtslog, order = c(2,1,4),include.drift = TRUE,fixed = c(0,NA,0,NA,0,0,NA))

coeftest(fitAAPL6)

summary(fitAAPL6)


#Check the residual(2,1,4)

# we can reject the null hypothesis here which means there could be correlation in the time series

acf(fitAAPL6$residuals)


Box.test(fitAAPL6$residuals,lag=10,type = "Ljung")

#Check the autocorrelation


fitAAPL7<-auto.arima(AAPLtslog)


coeftest(fitAAPL7)


summary(fitAAPL7)


#Check the residual


acf(fitAAPL7$residuals)


Box.test(fitAAPL7$residuals,lag=10,type = "Ljung")

#see how model performs using back test


# hold out the last 20% for prediction


length(AAPLtslog)*0.8
```

```
AAPLTrain = subset(AAPLtslog, end=3296)

AAPLTest = subset(AAPLtslog, start=3297)

#try (0,1,4)

fitAAPL2<-Arima(AAPLTrain, order = c(0,1,4),include.drift = TRUE,fixed = c(0,NA,0,NA,NA))
coeftest(fitAAPL2)
forcastm1 = forecast::forecast(fitAAPL2)
plot(forcastm1)
lines(AAPLTest, col="red")
RMSFE1 = sqrt(mean((forcastm1$mean - AAPLTest)^2))
RMSFE1
MAFE1 = mean(abs(forcastm1$mean - AAPLTest))
MAFE1
MAPE1 = mean(abs((forcastm1$mean - AAPLTest) / AAPLTest))
MAPE1   # About 3% off on average
#try (0,1,7)
fitAAPL4<-Arima(AAPLTrain, order = c(0,1,7),include.drift = TRUE,fixed = c(NA,NA,0,NA,0,NA,NA,NA))
summary(fitAAPL4)
forcastm2 = forecast::forecast(fitAAPL4)
plot(forcastm2)
lines(AAPLTest, col="red")
RMSFE2 = sqrt(mean((forcastm2$mean - AAPLTest)^2))
RMSFE2
MAFE2 = mean(abs(forcastm2$mean - AAPLTest))
MAFE2
MAPE2 = mean(abs((forcastm2$mean - AAPLTest) / AAPLTest))
```

```
MAPE2   # About 3% off on average
#try (0,1,2)
fitAAPL7<-auto.arima(AAPLTrain)

forcastm3 = forecast::forecast(fitAAPL7)

plot(forcastm3)

lines(AAPLTest, col="red")

RMSFE3 = sqrt(mean((forcastm3$mean - AAPLTest)^2))

RMSFE3
MAFE3 = mean(abs(forcastm3$mean - AAPLTest))

MAFE3
MAPE3 = mean(abs((forcastm3$mean - AAPLTest) / AAPLTest))

MAPE3   #
#Garch Analysis
# check the box.text of the return

r<- diff(AAPLts)

Box.test(r,type="Ljung")

# we ar going to check residual square

autoplot(fitAAPL4$residuals^2)
```

```
Acf(fitAAPL4$residuals^2,lag=50)

res = fitAAPL4$residuals

gFit = garch(res, order =c(1,1))

coeftest(gFit)

gresid= na.omit(gFit$residuals)

autoplot(gresid)

Acf(gresid,lag=100)

Box.test(gresid^2,type = "Ljung")
###################################################################

#Include the garch model and arma model together, t

install.packages("fGarch")

library(fGarch)

AAPLdiff<- diff(AAPLTrain)
gFit1 = garchFit( ~ arma(0, 7) + garch(1, 1), data = AAPLdiff, trace = F)

gFit1
# in garchFit, the "residuals" are the result of the Arima
```

```
# and the input to the garch.  To get the residuals after

# the garch, we need to divide by the computed "volatility"

# i.e. the standard deviation

autoplot(ts(gFit1@residuals / gFit1@sigma.t))
# Compare normality measures

skewness(gFit1@residuals / gFit1@sigma.t)

kurtosis(gFit1@residuals / gFit1@sigma.t)

normalTest(gFit1@residuals / gFit1@sigma.t)
# Now, the computed standard deviations are in sigma.t

# the computed variances are in "h.t"

autoplot(ts(gFit1@residuals)) +

  autolayer(ts(1.96 * gFit1@sigma.t), color="red") +

  autolayer(ts(-1.96 * gFit1@sigma.t), color="red")
ugarch.spec = ugarchspec(variance.model=list(garchOrder=c(1, 1)),

               mean.model=list(armaOrder=c(0, 7)))
gFit2 <- ugarchfit(ugarch.spec, diff(AAPLTrain))
gFit2@fit

gFit2res <- residuals(gFit2, standardize=T)
```

```
autoplot(gFit2res^2)

acf(gFit2res^2)
Box.test(gFit2res^2, lag = 15, type = 'Ljung')
gFit2.rolltest <- ugarchroll(ugarch.spec, data = diff(AAPLTrain), n.start = 2000,

                        refit.window = 'moving',

                        refit.every = 500)
report(gFit2.rolltest, type = 'fpm')
```