eWOM correlation with Apple stock price behavior

Joey Soeder

Oral Roberts University

Soederj2@oru.edu

General Terms:

Measurement, Performance

Keywords:

eWOM, Consumer Behavior, Retail Investing, Twitter, Stock, Correlation, Prediction

Abstract

It is widely known that eWOM influences the decision-making process of retail consumers. With the rise of commission-free trading platforms, it may also influence a growing amount of retail investors and become more relevant than ever. This study took Twitter data about the Apple, Inc. stock and conducted a sentiment analysis using the Loughran lexicon. This aggregate sentiment was then compared to stock price behavior over intervals of one hour. Logistic modelling was used to determine correlation between the two variables. Using the model, an algorithm was created that predicted 81.25% of values correctly. These findings were then compared to previous studies to show that there is a possibility that an increase in retail investors caused by commission-free trading platforms has led to an increase of stock price volatility in reaction to eWOM communication.

1. Introduction

Word of mouth communication has traditionally played an essential role in the marketing of products. With the emergence of the internet and subsequent emergence of web 2.0 and social media, word of mouth communication has further developed and expanded to be conducted online, resulting in electronic word of mouth (eWOM) (Hening-Thurau, Gwinner, Walsh, & Gremler, 2004). Although regularly recognized for its role in business to consumer (B2C) and business to business (B2B) sectors, it has also achieved prevalence in the financial investment market (Tang, Mehl, Eastlick, He, & Card, 2016). For example, infamous fraud architect Bernie Madoff built his entire Ponzi scheme upon WOM communication between investors (Catan & Bryan-Lowe, 2008).

In 2014, trading platform Robinhood was launched upon the premise of commission-free trading (Robinhood, 2020). The lack of commission fees allowed investors with smaller accounts to invest without the redundancy of commission-based fees, which ranged from seven to ten dollars per trade in competitive firms (Morrisey, 2017). At the end of 2020, Robinhood had achieved success with over five million users and a company valuation that has surpassed eleven billion dollars (Klebnikov, 2020). Observing the success of Robinhood, many competitors also implemented their proprietary commission-free trading platforms, such as Charles Schwab, Interactive Brokers, E-Trade, AllyInvest, Fidelity, and TradeStation (Reinkensmeyer, 2021).

Coinciding with the introduction of commission-free platforms has been a dramatic increase in retail investment in the stock market (Bloomberg News, 2020). The stock market has two main categories of investment – institutional investors and retail investors. Defined by Investopedia (2020), retail investors are individual or non-professional investors who purchase and sell securities, while institutional investors are trading professionals who invest others'

money on their behalf. The idea behind the designation is that institutional investors are more knowledgeable about the stock market, and so they will make more educated investments and manage risk better. Theories in the market today conclude that retail investors are more likely to rely on eWOM and uneducated decision-making techniques in the stock purchasing process (Beilfuss, 2019).

This study aims to look specifically at eWOM generated on Twitter in relation to Apple's stock performance and how this has been impacted by the mass access of trading offered by the recent innovation of commission-free trading platforms. Furthermore, it seeks to explore if the presence of people with an informal skill in the marketplace has coincided with a rise in volatility in the stock price behavior in relation to eWOM communication through Twitter. To better focus on these concepts, the social media platform of Twitter was chosen over alternatives such as Facebook, Instagram, and Stocktwits. Factors taken into account when making the decision included review of prior studies, the content type of posts on the platform as well as the instantaneous nature of the platform. Compared to graphic-heavy Facebook and Instagram, Twitter's content is more textual and is subsequently easier to conduct sentiment analysis on. In addition to this, Twitter is also known for its "real-time" nature, in which news items are shared almost instantaneously. Twitter also employs a cashtag feature, in which a person tweeting about Apple, for example, can use the cashtag '$AAPL' to denote that they are talking about Apple stocks (Hentschel, 2014). The cashtag feature makes it easier to find tweets that are specific to the company researched.

Apple was chosen for the analysis due to the stock's liquidity, retail investment percentage, and popularity on Twitter. In addition to being one of the top fifty most liquid stocks on the market, Apple has a much lower grade of institution investment than its peers, with

51.73% of the company's shares being held by individuals, which is significantly lower than Alphabet Inc's (GOOG) 68.3% and Facebook's (FB) 79.82% (Yahoo! Finance, 2020a ; Yahoo! Finance, 2020b; Yahoo! Finance, 2020c). In addition to this, $AAPL is the most used cashtag on Twitter (Mao et al., 2012).

A correlation between eWOM sentiment on Twitter and Apple's stock behavior could allow for businesses to further improve stock performance by driving positive eWOM and could allow investors to predict future stock price behavior through related sentiment analysis on social media.

2. Review of Literature

eWOM is defined as any positive or negative statement made by an individual about a product or company through use of the internet (Henning-Thurau et al., 2004). It is directly comparable to WOM, with the sole distinction being the setting of communication. eWOM falls under the larger categories of Consumer-Generated Content (CGC) and the synonymous User-Generated Content (UGC), which is defined as internet-based media that is created and published by everyday consumers rather than by professionals (Wang & Rodgers, 2011). Tweets published about a public company and its stock performance would then, by definition, fall into the eWOM category.

It is known that WOM and eWOM are present in the financial investment industry (Bikhchanadi & Sharma, 2000). It is particularly evident when observing the behavioral finance principle of herd mentality, which describes the tendencies of investors to knowingly imitate others' investment decisions (Marotta, 2008). The principle of herd mentality is a byproduct of WOM communication between two or more investors, as some form of communication must take place in order for one investor to imitate another.

As stated previously, the introduction of commission-free trading platforms has made trading stocks much more attractive to the casual small-account investor. Allowing everyone access to financial markets is a novel idea, but issues arise in the fact that financial literacy is positively correlated with wealth (Lusardi & Mitchell, 2017). One of the main concerns of allowing mass-access to trade on the stock market is that it is exposing the financially illiterate to risk that they are unable to properly mitigate. Further studies have determined that financial illiteracy is prevalent worldwide, including in countries with developed economies (Lusardi & Mitchell, 2011). An influx of impressionable investors could coincide with the phenomenon of herding, which can increase volatility and decrease the stability of the market (Bikhchanadi & Sharma, 2000).

A study conducted in 2015, before the commission-free trading platforms had caught on, found a significant relationship between Twitter sentiment and abnormal returns during peaks of Twitter volume, and that sentiment polarity during said peaks implied stock price behavior (Ranco et al., 2015). Although correlation was not established, it showcased that there is a relationship between eWOM and stock price performance.

In a study on the effect that Robinhood and related applications have on the market, Roberto Stein found that Robinhood users are 5 to 7 times more likely to purchase stocks that entered the app's "Top 100 listings" list (2020). Although not technically eWOM due to it being a list published by a company rather than organic communication generated by consumers, customer responses could be similar to eWOM communication created by other investors. This provides evidence of herd mentality among the app's investors and could allude to increased app usage correlating with increasingly predictable polarity of stock price behavior in relation to eWOM.

In a study that measured the impact of Twitter sentiment on the Standard & Poor's (S&P) 500, technology industry, and Apple (individual company) stock, it was discovered that tweet sentiment can serve as a significant predictor for stock price (Mao et al., 2012). In another study, which compared price behavior to sentiment on investor-based social media platform Stocktwits, it was determined that there were similar correlations between bullish and bearish sentiment and stock price behavior.

From these findings in these previous studies, it is anticipated that correlation between tweet sentiment and stock price behavior will exist. The hypothesis of this paper is as follows: A significant correlation exists between tweet sentiment and stock price behavior; the null hypothesis is that no significant correlation exists between tweet sentiment and stock price behavior.

3. Data Collection

In order to determine if a correlation between tweet sentiment and stock price behavior exists, tweets will be pulled from Apple's Twitter feed. Stock data will pulled from Kibot, a free resource available online where over 59 years of historical stock and financial data is accessible (Oricsoft, 2021).

Twitter Data

The analysis consisted of 9,061 tweets, which were pulled from the Twitter feed using the company's API. The tweets were then read into data programming software R using the "Rtweets" package. The tweets came from a time period extending from November 16, 2020 at 1:00 am Coordinated Universal Time (UTC) to November 19, 2020 at 23:59 UTC. Using the Twitter API, search parameters were set to include the cashtag $AAPL. This allowed a look at tweets specific to information that may have impacted Apple's stock price and eliminated

information that was not as helpful. This data was held in a data frame separate from the stock data.

Stock Data

The stock data used was pulled from findata company Kibot**.** The data frame used consisted of intraday, 30-minute price intervals for the AAPL stock, dating from January 2, 1998 to November 20, 2020. Each row covered a 30-minute interval, and contained open, close, high, low, and volume data values. These values were adjusted for all stock splits and dividend payments during the time period. This data was held in a data frame separate from the tweet data.

Data cleaning

As the data was raw and relatively unstructured, a lengthy cleaning process was involved. First, both datasets were converted to the UTC time zone to ensure accurate time series forecasting before filtering the stock data to fit the time frame that was covered by the Twitter data. Because time intervals of one hour for analysis purposes were chosen, the stock data was grouped into hour intervals, keeping the adjusted open from the first 30-minute interval, the adjusted close from the second 30-minute interval, and the highest and lowest adjusted high and adjusted low values from the hour interval.

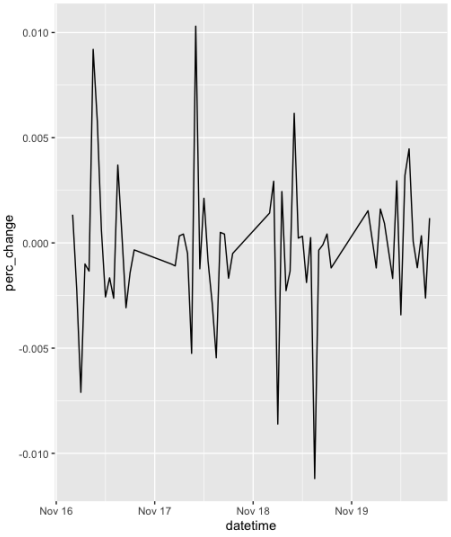
The tweet data frame also needed some cleaning. First, tweets were filtered out that had media or external URL links. This is due to external factors that could change the context of the tweet completely and a sentiment analysis would not be able to pick up on it. Using the unnest_tokens function with the Tweet token from the TidyText package, a tidy data frame was created with each individual word in its own unique row. Using the tweet token allowed for hashtags to be kept that may have positive or negative sentiment but prevented an antijoin function from being used to clean the data before conducting sentiment analysis. In place of the

antijoin, a character value of all edits was created before using mutate and filter functions to remove the data that needed to take place. This included deleting stop words, removing punctuation, and stripping words to their base. This left two tidy data sets that would allow for proper analysis.

Data Transformation

In the stock data frame, the mutate function was accessed to add a column for percentage change, which was the quotient of the difference between the adjusted open and adjusted close of the interval and the adjusted open of the interval. This provided direction as to whether the stock had made a positive or negative change throughout the interval and was expressed as a percentage to remove any possible issues from variable pricing. Two additional columns were created - a range column, which was calculated as the difference between the adjusted high and adjusted low values during the interval, and a range-as-a-percentage column, which expressed the range column as a percentage of the intervals adjusted open value and served to illustrate the stocks volatility over the interval. The final two columns were not used in the modelling portion of the data but could be interesting variables to use in future research. The percent change over time is illustrated in figure 1.

Figure 1: Percent Change in Stock Price Over Time



In order to run sentiment analysis on the tweet data frame, the get_sentiment function was used in the textdata package with the Loughran lexicon. This lexicon was developed for the sentiment analysis of company 10-k reports and purposely avoids the use of words that are vague in financial terms, such as “share” and “liability” (Loughran & McDonald, 2011). The lexicon output eight different sentiments: Negative, Positive, Uncertainty, Litigious, Strong Modal, Weak Modal, and Constraining. For the sake of the analysis, numerical values of +1 were assigned to the words with positive sentiment and -1 to words with a negative sentiment.

The inner_join function was used to assign each word a numerical value before using the group_by function and a dummy data set in order to compile the scores and attribute them to each original tweet. After a sentiment value has been assigned to each tweet, the data was grouped into 1-hour intervals to match the stock data and then an hourly sentiment value was created that was calculated by taking the sum of the numerical sentiment scores of each intervals’ tweets. The tweet sentiment calculated over the time frame of the study is illustrated in figure 2 below.

Figure 2: Tweet Sentiment Over Time

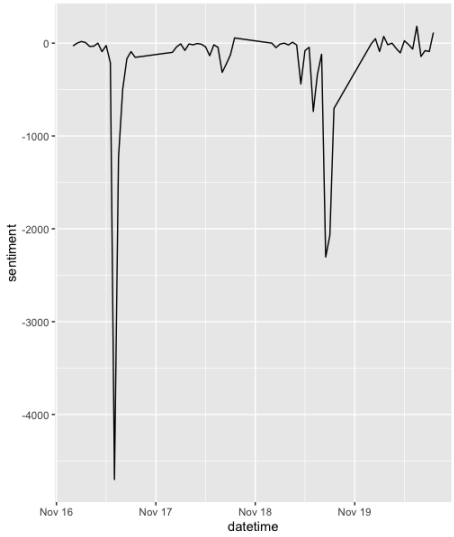
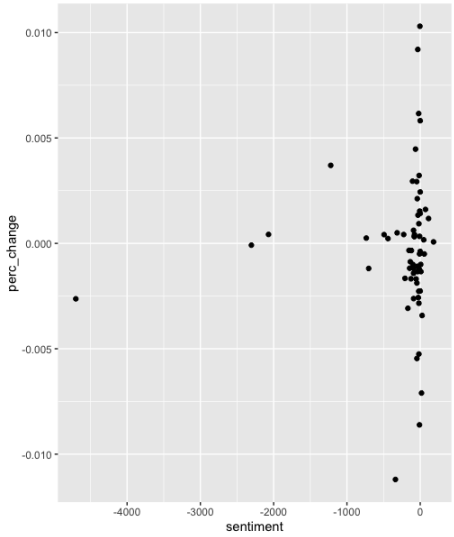


Figure 3: Percent Change in Stock Price Compared to Tweet Sentiment



To prepare the data for modelling, the ts and as.data.frame functions were used to create a dummy frame using the time series of intervals selected for the study. The inner_join function was then used to combine the stocks, tweets, and time series frame using the datetime value as the key. Next, a conditional function was utilized that gave a binary output in a new column depending on the condition of the stock and tweet data having positive correlation. For example, if sentiment was positive and percent change in the stock price was positive, then the value

printed in the new column was “1”. If sentiment was negative and the percent change in the stock price was positive, then the value printed in the new column would be a “0”.

Data Modelling

In the modelling portion of the analysis, logistic and linear models were conducted in order to determine if there was an accurate way to determine correlation. The testing process began by splitting the data into a training and testing set with a ratio of 75% to 25%, respectively. The training set was used to create the respective model, and the testing set was then used to test the models’ accuracy. For testing purposes, the null and alternative hypotheses were considered.

For the logistic model, a model that predicted correlation as positive or negative based on all of the other values in the data set was used. These values consisted of sentiment, percent change in stock price, and date and time. Through intermittent testing, it was determined that the ideal threshold for a 1-0 split would be 0.61. To test the accuracy of model, a confusion table was created. For the Linear model, a model that calculated the percent change in the stock price with tweet sentiment as the predictive variable was created. Due to poor performance indicators, this model was not tested.

4. Results

The logistic model produced a P value of 0.00, which leads to a rejection of the null hypothesis as it does provide evidence that supports that tweet sentiment and stock price behavior are correlated. When the trained model was run on the test set, it correctly predicted 13 of 16 values for an accuracy rate of 81.25%. However, it should be noted that the model has low pseudo-R squared measures of 0.37 (Cragg-Uhler) and 0.23(McFadden). This is most likely a result of the small data size and would be improved given more training data.

The Linear model was less successful; the calculated line was $y = 0.00x+0.00$. The model also resulted in an R squared value of 0.01 and a P value of 0.66. These results lead to failing to reject the null hypothesis, but due to the poor fit to the data, the model was discarded. An attempted test returned zero accurate predictions.

5. Analysis and Future Use

The Logistic model leads to a failure to reject the alternative hypothesis because there is evidence that supports that tweet sentiment and stock price behavior are correlated. This sentiment is also supported by the findings in previous studies of similar topics (e.g. Mao et al., 2012; Ranco, et al., 2015. It is important to stress that correlation does not equate to causation; there are many factors that could cause both twitter sentiment and stock price behavior to rise without the two affecting each other. Due to the instantaneous nature of both data sets, both data could be reacting to the same variable rather than influencing each other. However, creating a model with 81.25% prediction accuracy could provide a useful tool in future decision making with regards to the stock market. The fact that the results remain a positive correlation from 2012, when retail investing was at a much lower level than it is currently, disproves any notion

that retail investing would reduce the volatility of stock price in reaction to eWOM and invites further research on the subject.

Several areas for future research can stem from this study. As this study only researched Apple stock, this research could expand to include additional stocks, for example, consider the top 50 more liquid stocks. A resulting model would provide a better understanding of the overall market and its interaction with the social media platform Twitter. Additionally, this study focused on tweets over a four-day span. Considering a longer time frame when conducing the sentiment analysis could improve upon the predictive quality of the model. Tweet volume and tweet sentiment could also be analyzed to discover whether the volatility rate can be calculated as the percent range in the stock data set. Additional research could be conducted based on the type of investor providing Twitter content (investor, potential investor, past investor, observer). Many factors can still be explored within this area that can be used to provide a clearer understanding of the impact of electronic word of mouth communication on stock price behavior.

In light of the attention that stock prices and electronic word of mouth communication have received on platforms such as Robinhood and Reddit (Lipschultz, 2021), this is an area of research that should be continued. Companies that can monitor their stock sentiment could potentially predict the behavior of their stock price. Additional research should continue in this area to better understand the stock price behavior that correlates with communication generated by investors and potential investors.

References

Argan, M. T., Yalaman, A., & Sevil, G. (2014). The effect of word-of-mouth communication on stock holdings and trades: Empirical evidence from an emerging market. *Journal of Behavioral Science*, *15*(2), 89-98.

Beilfuss, L. (2019, January 22). The latest trend in mobile gaming: Stocktrading apps; apps such as robinhood and webull are drawing in young and often inexperienced investors; 'bringing the ability to make foolish decisions to an ever-broader swath of people'. *Wall Street Journal*.

Bikhchandani, S., & Sharma, S. (2000). Herd behavior in financial markets: A review. *IMF working paper*, (48), 1-32.

Bloomberg News. (2020, August 10). *Robinhood outruns rivals in record year for retail investing*. Investment News. https://www.investmentnews.com/robinhood-dart-data-retail-investing-195941

Catan, T., & Bryan-Low, C. (2008, December 6). The Madoff case: European clients were cultivated within social networks by word of mouth. *Wall Street Journal*, p. A19.

Hennig-Thurau, T., Gwinner, K.P., Walsh, G., & Gremler, D.D. (2004). Electronic word-of-mouth via consumer-opinion platforms: What motivates consumers to articulate themselves on the Internet? *Journal of Interactive Marketing, 18*(1), 38-52. doi: 10.1002/dir.10073

Klebnikov, S. (2020). Robinhood valuation soars to $11.2 billion with new funding and record growth. *Forbes.com*

Lipschultz, B. (2021). Reddit-fueled traders trigger volatility halts across the market. *Bloomberg.com*

Loughran, T., & McDonald, B. (2011). When is a liability not a liability? Textual analysis, dictionaries, and 10-Ks. *Journal of Finance, 66*(1), 35-65.

Lusardi, A., & Mitchell, O. (2011). Financial literacy around the world: An overview. *Journal of Pension Economics and Finance, 10*(4), 497-508.

Lusardi, A., & Mitchell, O. S. (2014). The economic importance of financial literacy: Theory and evidence. *Journal for Economic Literature, 52*(1), 5-44. https://doi.org/10.1257/jel.52.1.5

Mao, Y., Wang, B., Wei, W., & Liu, B. (2012). Correlating S&P 500 stocks with twitter data. *Social Analytics*, 69-72.

Marotta, D.J. (2008). Behavioral finance: Herd mentality. *Business Journal (Central New York), 22*(33), 23.

Morrissey, J. (2017, February 19). No-frills newcomer takes on big brokerages. *New York Times*, p. BU.3

Oricsoft. (2021). Kibot: Historical intraday data. Retrieved from http://www.kibot.com/

Ranco, G., Aleksovski, D., Caldarelli, G., Grcar, M., & Mozetic, I. (2015). The effects of Twitter sentiments on stock price returns. *PLoS ONE, 10*(9). https://doi.org/10.1371/journal.pone.0138441

Reinkensmeyer, B. (2021). Best brokers for free stock trading 2021. Retrieved from https://www.stockbrokers.com/guides/free-stock-trading

Robinhood. (2020). *About Us*. https://robinhood.com/us/en/careers/.

Stein, R. (2020). *The top 5 predictable effects of new entries in Robinhood's '100 most popular' list*. SSRN. http://dx.doi.org/10.2139/ssrn.3694588

Sundaram, D. S., Mitra, K., & Webster, C. (1998). Word of mouth communications: A motivational analysis. *NA- Advances in Consumer Research*, *25*, 527-531.

Tauni, M. Z., Fang, H. X., & Iqbal, A. (2017). The role of financial advice and word-of-mouth communication on the association between investor personality and stock trading behavior: Evidence from Chinese stock market. *Personality and Individual Differences*, *108*, 55-65. https://doi.org/10.1016/j.paid.2016.11.048

Tang, C., Mehl, M.R., Eastlick, M.A., He, W., & Card, N.A. (2016). A longitudinal exploration of the relations between electronic word-of-mouth indicators and firms' profitability: Findings from the banking industry. *International Journal of Information Management*, *36*(6), 1124-1132. https://doi.org/10.1016/j.ijifomgt.2016.03.015

Wang, Y., & Rodgers, S. (2010). Electronic word of mouth and consumer generated content: From concept to application. In *Handbook of Research on Digital Media and Advertising* (pp. 212-231). Information Science Reference.

Yahoo! Finance. (2020a, November 22). *Apple, inc. (AAPL)*. Yahoo!. https://finance.yahoo.com/quote/AAPL/holders?p=AAPL

Yahoo! Finance. (2020b, November 22). *Alphabet, inc. (GOOG)*. Yahoo!. https://finance.yahoo.com/quote/GOOG/holders?p=GOOG

Yahoo! Finance. (2020c, November 22). *Facebook, inc. (FB)*. Yahoo!. https://finance.yahoo.com/quote/FB/holders?p=FB

APPENDIX

Appendix 1: R Code

```
################################################################################
    ####
### PACKAGES ###
################################################################################
    ####
library(rtweet)
library(dplyr)
library(tidyr)
library (tidytext)
library(httpuv)
library(ggplot2)
library(lubridate)
library(chron)
library (stringr)
library(textdata)
library(randomForest)
library(caTools)
library(e1071)
library(jtools)
```

```
################################################################################
    ####
###TWEET ACCESS ###
################################################################################
    ####

## label key and secret key
#api_key <- "XXXXXXXXXXXX"
#api_secret_key <- "XXXXXXXXXXXXXXXXXXX"

### Create Token for API access ###
#token <- create_token(
#  app = "Soederstocks",
#  consumer_key = api_key,
 # consumer_secret = api_secret_key)

################################################################################
    ####
###uploading and cleaning TWEET data ###
################################################################################
    ####

## upload aapl tweets
```

```
#AAPLtweets <- rtweet::search_tweets(q = '$AAPL',
#                                    n = 10000,
#                                    since = "2020-11-10",
#                                    until = "2020-11-19")


## read in data that I had uploaded^^
AAPLtweetsread <- readr::read_csv("/Users/joeysoeder/Downloads/GCB
        FINAL/AAPLtweets.csv")
AAPLstock <- readr::read_csv('/Users/joeysoeder/Downloads/GCB FINAL/AAPL.txt')
AAPLstocka <- readr::read_csv('/Users/joeysoeder/Downloads/GCB FINAL/AAPL.txt')


##checking to make sure that time zones were the same
as_datetime(AAPLtweetsread$created_at)
as_datetime(AAPLstock$time)



## convert to data frame
AAPLtweets <- as.data.frame(AAPLtweetsread)



##split Created at into date and time- keep datetime for later
AAPLtweets$date <- as.Date(AAPLtweets$created_at, format = "%Y/%m/%d")
AAPLtweets$time <- format(AAPLtweets$created_at, format = "%H:%M:%S")
```

```
AAPLtweets$datetime <- as_datetime(AAPLtweets$created_at)

## split dates into individual for grouping purposes
AAPLtweets <- mutate(AAPLtweets,
                     year = as.numeric(format(AAPLtweets$date, format = "%Y")),
                     month = as.numeric(format(AAPLtweets$date, format = "%m")),
                     day = as.numeric(format(AAPLtweets$date, format = "%d")))

## subset with only necessary data
AAPLtweets <- select(AAPLtweets, c(time, year, month, day, text, urls_url, media_url, lang,
          datetime))

## filter out non-english tweets, Media,URL's
AAPLtweets <- filter(AAPLtweets, lang == "en")
AAPLtweets <- filter(AAPLtweets, is.na(AAPLtweets$media_url) == TRUE)
AAPLtweets <- filter(AAPLtweets, is.na(AAPLtweets$urls_url) == TRUE)

## give each tweet unique ID for sentiment grouping
AAPLtweets$ID <- seq.int(nrow(AAPLtweets))

##reorder & delete
AAPLtweets <- select(AAPLtweets, c(ID, time, year, month, day, text, datetime))
```

```
################################################################################
    ####
###TEXT MINING ###
################################################################################
    ####


## remove stop words and such but keep hashtags and usernames, also tokenize
remove_reg <- "&amp;|&lt;|&gt;"

AAPLtext <- AAPLtweets %>%
  filter(!str_detect(text, "^RT")) %>%
  mutate(text = str_remove_all(text, remove_reg)) %>%
  unnest_tokens(word, text) %>%
  filter(!word %in% stop_words$word,
       !word %in% str_remove_all(stop_words$word, "'"),
       str_detect(word, "[a-z]"))

##get sentiment dataset and add numerical values to neg and pos sentiments

possentiment = get_sentiments(lexicon = "loughran") %>%
  filter(sentiment == "positive") %>%
  mutate(sentiment = 1)
```

```
negsentiment = get_sentiments(lexicon = "loughran") %>%
  filter(sentiment == "negative") %>%
  mutate(sentiment = -1)

sentiment <- rbind(possentiment,negsentiment)

## join sentiment df and AAPLtweets df

AAPLsentiment <- AAPLtext %>%
  inner_join(sentiment)

AAPLsentiment <- AAPLsentiment %>%
  group_by(ID) %>%
summarise(sentiment = sum(sentiment))

## join sentiment sums to original tweet data
AAPLtweets <- inner_join(AAPLtweets, AAPLsentiment, by = 'ID')


##Group in 30 min intervals
```

```
AAPLtweets$hour <- as.POSIXlt(AAPLtweets$datetime)$hour


AAPLtweets <-  AAPLtweets%>%
  group_by(date=floor_date(datetime, "1 hour")) %>%
  summarize(ID = ID, year = year, month = month, day = day, datetime = datetime, hour = hour,
            sentiment = sum(sentiment))



### ready to go - add a dataframe based on the time series studied and then inner join based on
            time/hour

##############################################################################
            ####
###STOCKS DATA ###
##############################################################################
            ####


## import data
##AAPLstock <- readr::read_csv('/Users/joeysoeder/Downloads/GCB FINAL/AAPL.txt')


## define COL names
names(AAPLstock) <- c("date", "time", "adj_open" , "adj_high", "adj_low", "adj_close" ,
            "volume")
```

```
##set date as date
AAPLstock$date <- as.Date(AAPLstock$date, format = "%m / %d / %Y")


##split date to 3 seperate col
AAPLstock <- mutate(AAPLstock,
            year = as.numeric(format(AAPLstock$date, format = "%Y")),
            month = as.numeric(format(AAPLstock$date, format = "%m")),
            day = as.numeric(format(AAPLstock$date, format = "%d")))

## filter to selected time period
AAPLstock <- filter(AAPLstock, year == 2020, month == 11, day >= 10)


## create new columns <- %change, range (variation), and %range (volatility?)
AAPLstock <- mutate(AAPLstock, perc_change = ((adj_close - adj_open)/adj_open),
              range = (adj_high - adj_low),
              perc_range = ((adj_high - adj_low)/ adj_open))


AAPLstock$datetime <- paste(AAPLstock$date, AAPLstock$time)
```

```
AAPLstock$datetime <- as_datetime(AAPLstock$datetime)



AAPLstock$hour <- as.POSIXlt(AAPLstock$datetime)$hour


AAPLstocks <-  AAPLstock%>%
  group_by(date=floor_date(datetime, "1 hour")) %>%
  summarize(time = time, volume = sum(volume), perc_change = sum(perc_change), perc_range
            = sum(perc_range), hour = hour)



#########################################################


AAPL <- seq(as_datetime('2020-11-16 01:00:00'), as_datetime('2020-11-20 12:30:00'), by =
            'hours')
AAPL <- as.data.frame(AAPL)
names(AAPL) <- c('datetime')


AAPLt <- select(AAPLtweets, datetime, sentiment)
AAPLs <- select(AAPLstocks, date, perc_change, perc_range)


names(AAPLs) <- c("datetime", "perc_change", "perc_range")
AAPLs$datetime <- format(AAPLs$datetime, "%Y/%m/%d %H")
```

```
AAPL$datetime <- format(AAPL$datetime, "%Y/%m/%d %H")

AAPLt$datetime <- format(AAPLt$datetime, "%Y/%m/%d %H")


AAPL <- inner_join(AAPL,AAPLs, by = "datetime")

AAPL <- inner_join(AAPL, AAPLt, by = 'datetime')


AAPL <- select(AAPL, datetime, sentiment, perc_change)


AAPL <- AAPL[!duplicated(AAPL), ]


################################################################################
    ####
###MODELLING ###
################################################################################
    ####


## simple plot
ggplot(data = AAPL)+
  geom_point(mapping = aes(x = sentiment, y = perc_change))


### I found a way around it <- exported to Excel and then went through manually and introduced
    correlation values.
```

```
AAPL <- readr::read_csv('/Users/joeysoeder/Downloads/AAPLcor.csv')

AAPL <- select (AAPL, !c("ID"))

AAPL$datetime <- as_datetime(AAPL$datetime,format = "%Y/%m/%d %H")

AAPL <- AAPL %>%

  filter(!is.na(datetime))


ggplot(data = AAPL)+

geom_line(mapping = aes(x = datetime, y = perc_change))


ggplot(data = AAPL)+

  geom_line(mapping = aes(x = datetime, y = sentiment))


## split data

sample = sample.split(AAPL$correlation, SplitRatio = 0.75)

train = subset(AAPL, sample == TRUE)

test  = subset(AAPL, sample == FALSE)

dim(train)

dim(test)




## LOGMOD
```

```
logistic_model<- glm(correlation == '1' ~ ., data = train, family = "binomial")

summ(logistic_model)

# make predictions

Logistic_predicted <- predict(logistic_model, test, type = "response")


############################################################

##MODEL INFO:

#  Observations: 48

#Dependent Variable: correlation == "1"

#Type: Generalized linear model

#Family: binomial

#Link function: logit


#MODEL FIT:

#  χ²(3) = 15.41, p = 0.00

#Pseudo-R² (Cragg-Uhler) = 0.37

#Pseudo-R² (McFadden) = 0.23

#AIC = 59.05, BIC = 66.53


#Standard errors: MLE

#-------------------------------------------------

#  Est.      S.E.   z val.      p

#----------------- ---------- --------- -------- ------
```

```
# (Intercept)        -7114.37   6194.76    -1.15   0.25
#datetime              0.00      0.00     1.15   0.25
#sentiment            -0.00      0.00    -0.56   0.57
#perc_change         -604.12    231.98    -2.60   0.01
#---------------------------------------------------
#############################################################################
# set threshold for 1 - 0 split
# ###
threshold <- .61


# # This is the code block to run after changing your threshold value to see how the accuracy
        changes.
predicted <- predict(logistic_model, test, type="response")
# # convert predictions to binary
predicted <- ifelse(predicted > threshold, 1, 0)
if(!1 %in% test$correlation){
  test$correlation <- ifelse(as.character(test$correlation) == '1', 0, 1)
}
table(test$correlation, predicted)
(accuracy = sum(test$correlation == predicted) / length(predicted))


#0.8125
```

```
#==============================

# Linear Regression Model

#==============================


linear_model <- lm(perc_change ~ sentiment, train)


summ(linear_model)


# # Use the model to make predictions on the test set

linear_predicted <- predict(linear_model, test, type="response")


##MODEL FIT:

##  F(1,46) = 0.24, p = 0.63

##R² = 0.01

##Adj. R² = -0.02


##Standard errors: OLS

#------------------------------------------------

#  Est.   S.E.   t val.      p

#----------------- ------- ------ -------- ------

#  (Intercept)        -0.00   0.00    -0.15   0.88

#sentiment           0.00   0.00     0.49   0.63

#------------------------------------------------
```