# Modeling Tesla Stock Price Using Tweets

Tilina Alzaben, Jihyo Chung, Marion Haney, Divya Rao

04-21-2024

## Executive summary

## 1 Introduction

Public opinion of a company can be beneficial or harmful to the company's image and may also affect its sales or profits. Thus, every large company hires employees to benefit their branding, marketing, and public communications. We set to investigate whether we can quantify a relationship between a company's public image and their financial health using the case study of Tesla.

Tesla, a company which specializes in electric vehicles, is currently led by Elon Musk, who is often intertwined with the company's public image. Musk can be considered a public relations nightmare, engaging in Twitter rants and having a lack of filter that often leads to controversies and speculation.

To measure public opinion, we use textual data gathered from Twitter, a text-based social media with millions of English-speaking users. Tweets mentioning Tesla or the company's stock ticker, TSLA, over one year were collected (09-30-2021 to 09-30-2022). From the tweets, we assign sentiment scores based on the positive and negative language used from the text of the tweets, giving us a measurable proxy for public opinion. To measure the financial health of Tesla, we observe its adjusted closing stock price and returns over same the year.

We chose Tesla because of our background knowledge of its popularity within the public eye, giving us a wealth of positive and negative examples for sentiment analysis. Also, Tesla's stock TSLA is a notably volatile stock, jumping between $209.38 and $409.97 several times during the year of data.

We engage in time series analysis and attempt to investigate a potential relationship between Tesla's Twitter sentiment and stock returns on a given day over the year of data.

### 1.1 Research Questions

The project aimed to answer two main questions: First, is there a relationship between the sentiment of Tesla-related tweets and the company's stock returns on any given day? Second, can the sentiment derived from tweets mentioning Tesla be used to forecast the company's stock returns on a subsequent day?

### 1.2 Data

For this analysis, we utilized two related datasets from Kaggle. The first dataset includes 80,793 tweets related to 25 different companies, while the second provided daily closing stock prices for these same companies over a period from September 30, 2021, to September 30, 2022. These companies are the top 25 "most watched" stocks from Yahoo Finance and we filtered our data to only include tweets that mention Tesla or TSLA stock. This left us with 46% of the tweets in data mentioned TSLA (37422 tweets) across the year of data collection.

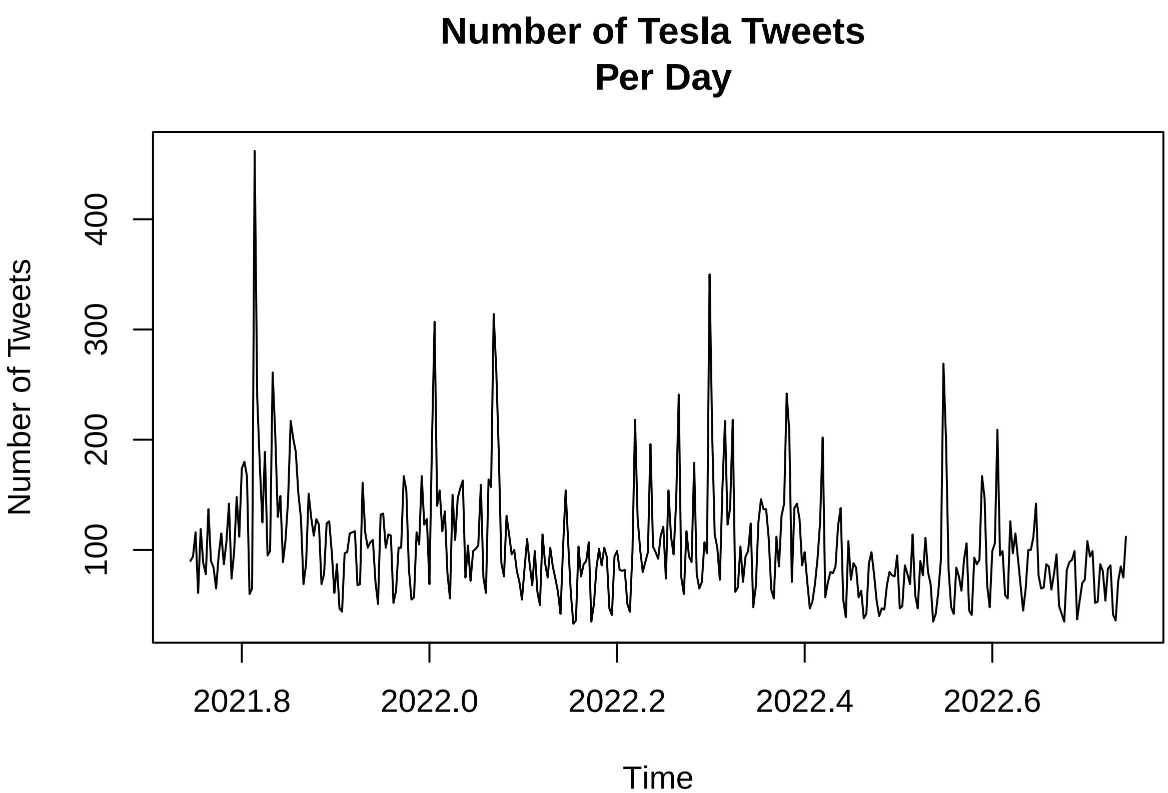
# 2 Methods

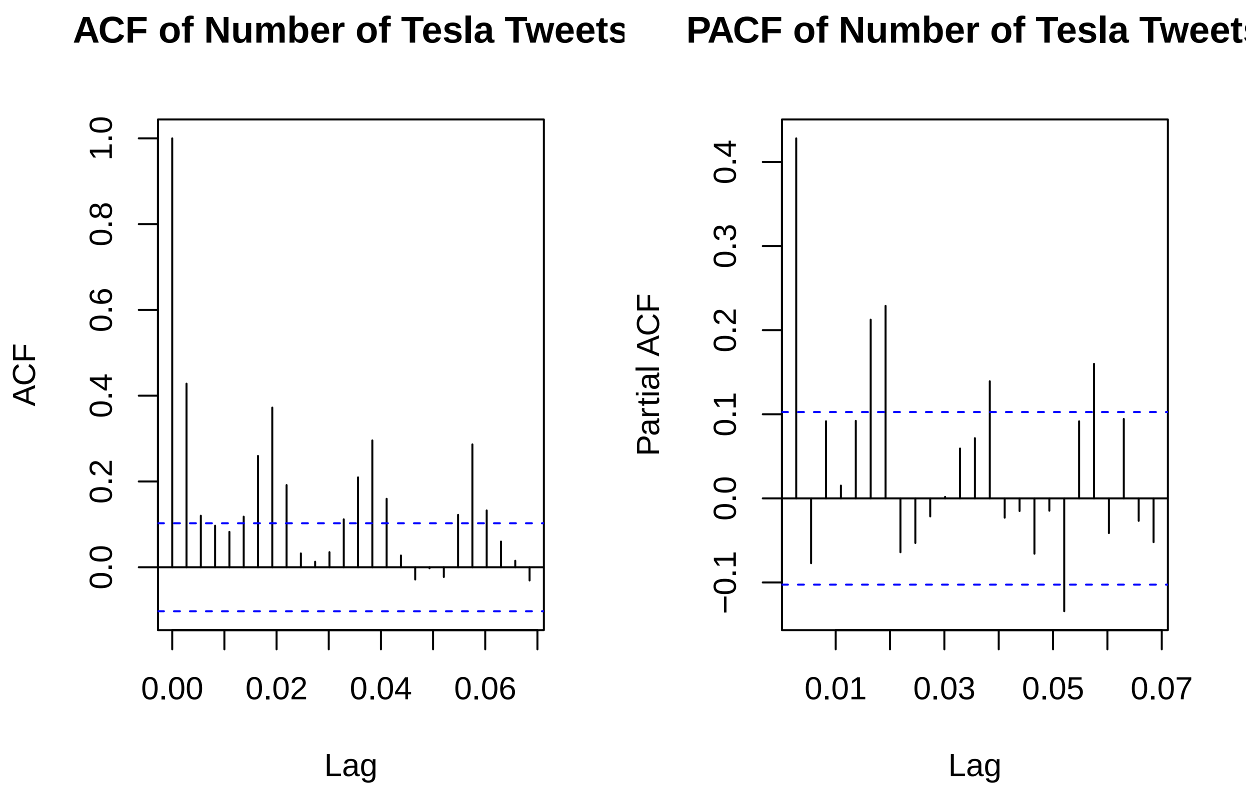
## 2.1 Exploratory Data Analysis

Exploratory Data Analysis (EDA) was conducted for deciding on reasonable data transformations and analysis, determining which type of model to use, and choosing the parameters for these models.

## 2.2 Number of Tesla Tweets Per Day

Our data included tweets from 09-30-2021 to 09-30-2022 (excluding weekends) which mentioned either Tesla's stock ticker or the company name itself. Our data included 37,422 tweets regarding Tesla over the year. Here, we view the number of tweets mentioning Tesla per day plotted over time along with the ACF and PACF plots of the time series.





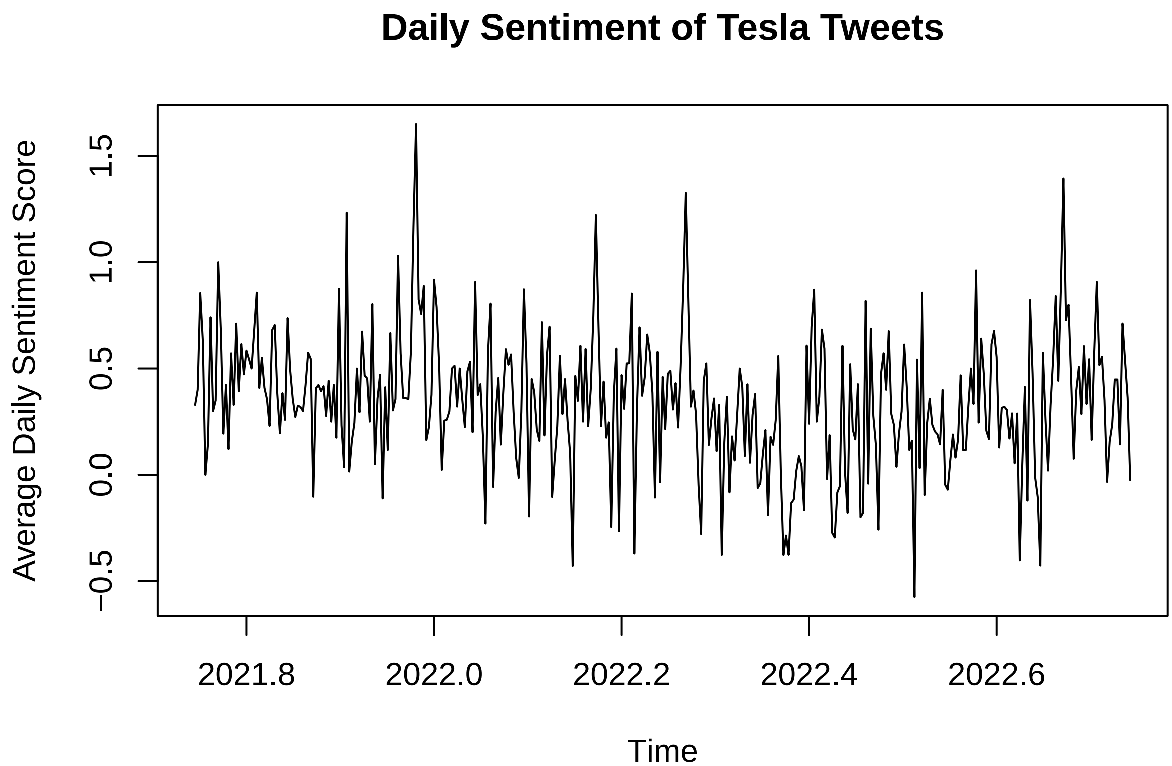

The time series of the number of Tesla tweets per day shows non-stationary characteristics, with noticeable fluctuations and periodic spikes indicating changes in the mean and variance over time. The ACF plot specifically shows possible seasonality with four clusters of ACF spikes over the time period of the data. These spikes in the number of tweets mentioning Tesla appear to occur approximately every three months.

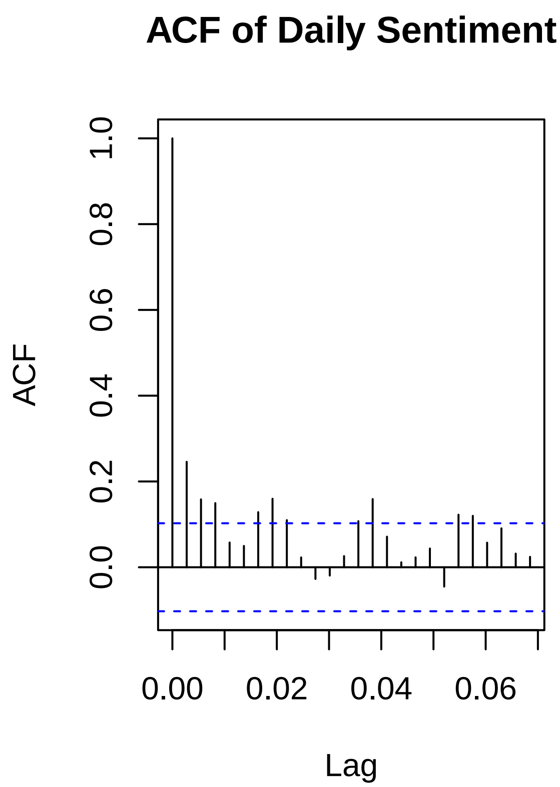
Since the number of Tesla tweets varies per day in a periodic manner, we decided to use averaging in our calculation of daily tweet sentiment. Thus, the fluctuating number of Tesla tweets is accounted for.

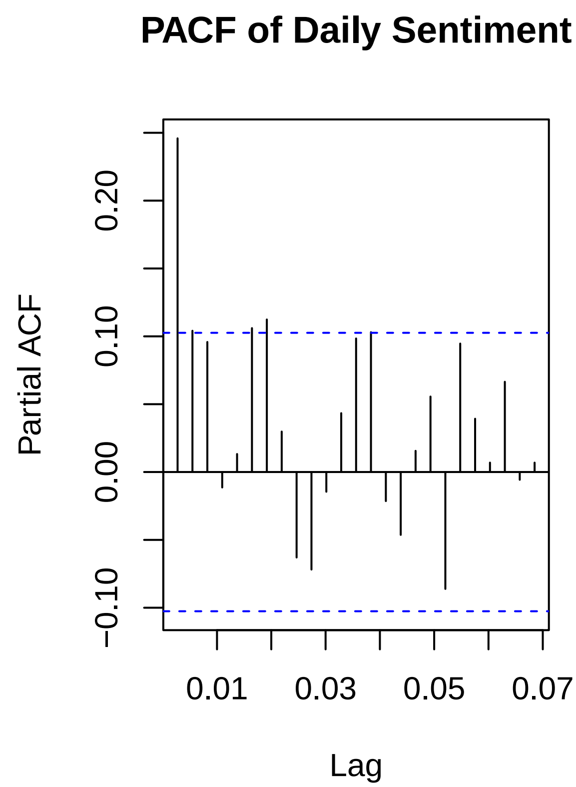
## 2.3 Daily Sentiment of Tesla Tweets

For our analysis, we calculated a daily score representing the average sentiment (positive or negative) for Tesla tweets posted on the given day. Our approach involved breaking each tweet into a bag of words representation, where each word is given a sentiment score of 1 if it is positive, -1 if it is negative, and 0 otherwise (neutral). These individual word scores were then summed into a sentiment score for each tweet. The sentiment scores per tweet ranged from -9 to 9. Finally, the individual tweet sentiment scores were averaged into a daily tweet sentiment score for the day. The daily sentiment scores ranged from -0.575 to 1.65.

Here, we can view the daily sentiment of tweets over the year of our data along with the ACF and PACF.


Daily Sentiment of Tesla Tweets
Average Daily Sentiment Score
−0.5
0.0
0.5
1.0
1.5
2021.8
2022.0
2022.2
2022.4
2022.6
Time

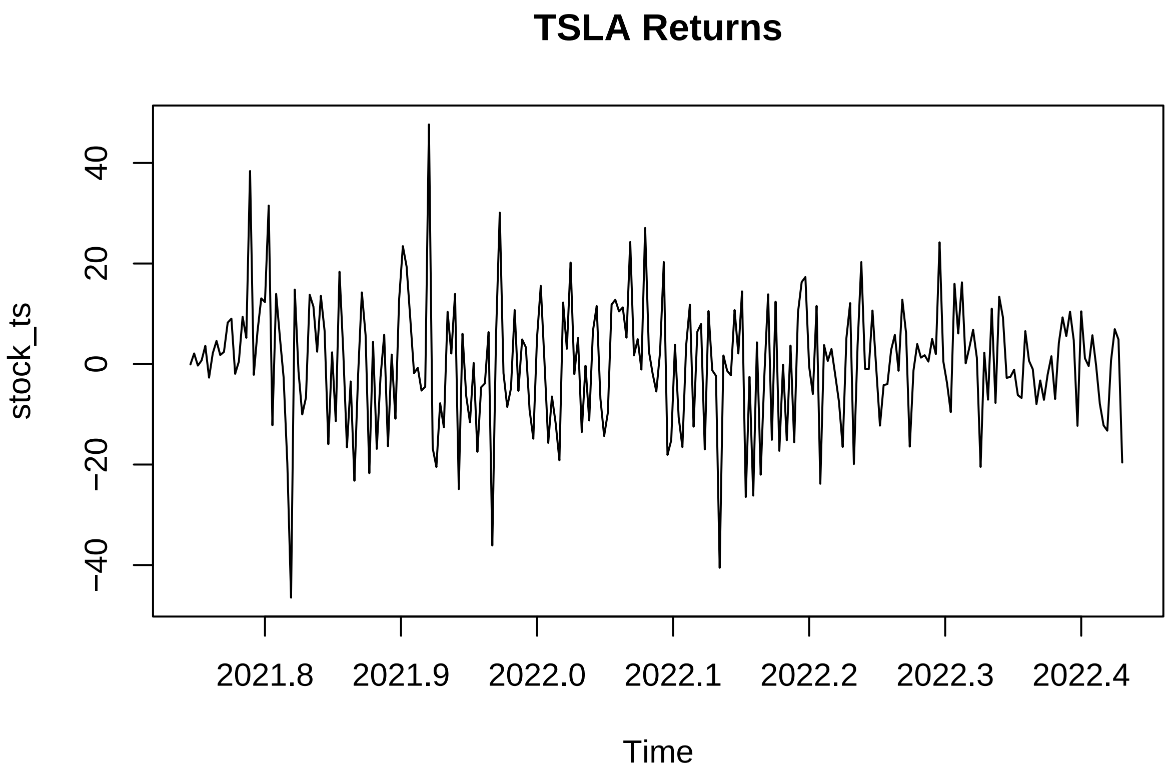


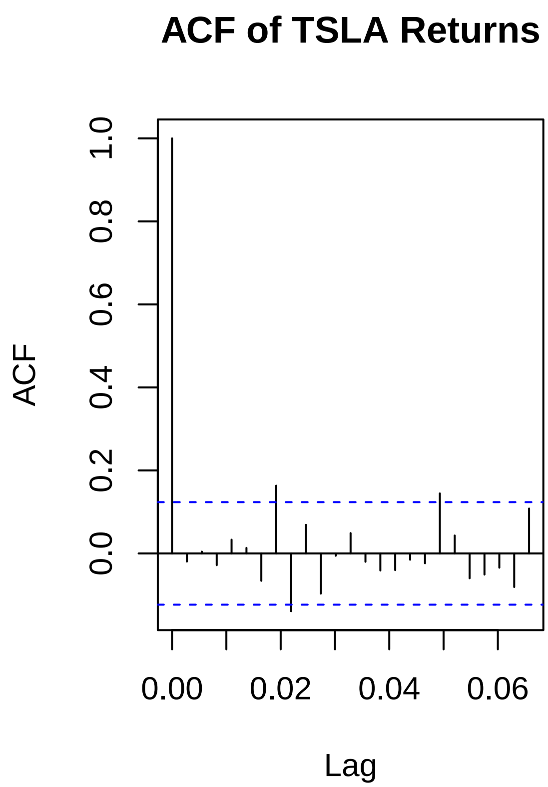



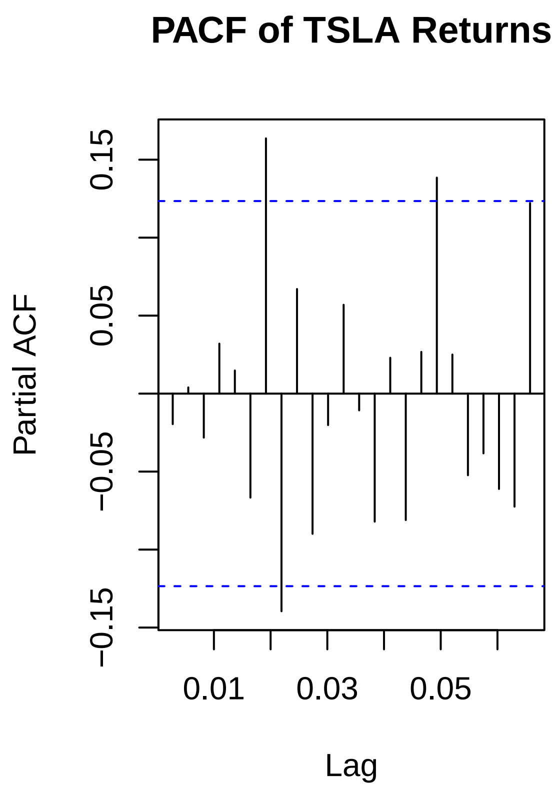

Similarly to the number of Tesla tweets per day, we observe non-stationary characteristics for the daily sentiment of Tesla tweets. we see four sections of ACF spikes approximately 3 months apart, suggesting seasonality. However, the magnitude of the spikes for daily sentiment of tweets is less severe than the magnitude of the corresponding spikes for the number of tweets per day. This is potentially due to the averaging of each individual tweet's sentiment score to produce a daily metric.

## 2.4 TSLA Stock Price and Returns

We calculated the returns of the TSLA stock by differencing the adjusted close price of TSLA to 1 degree.

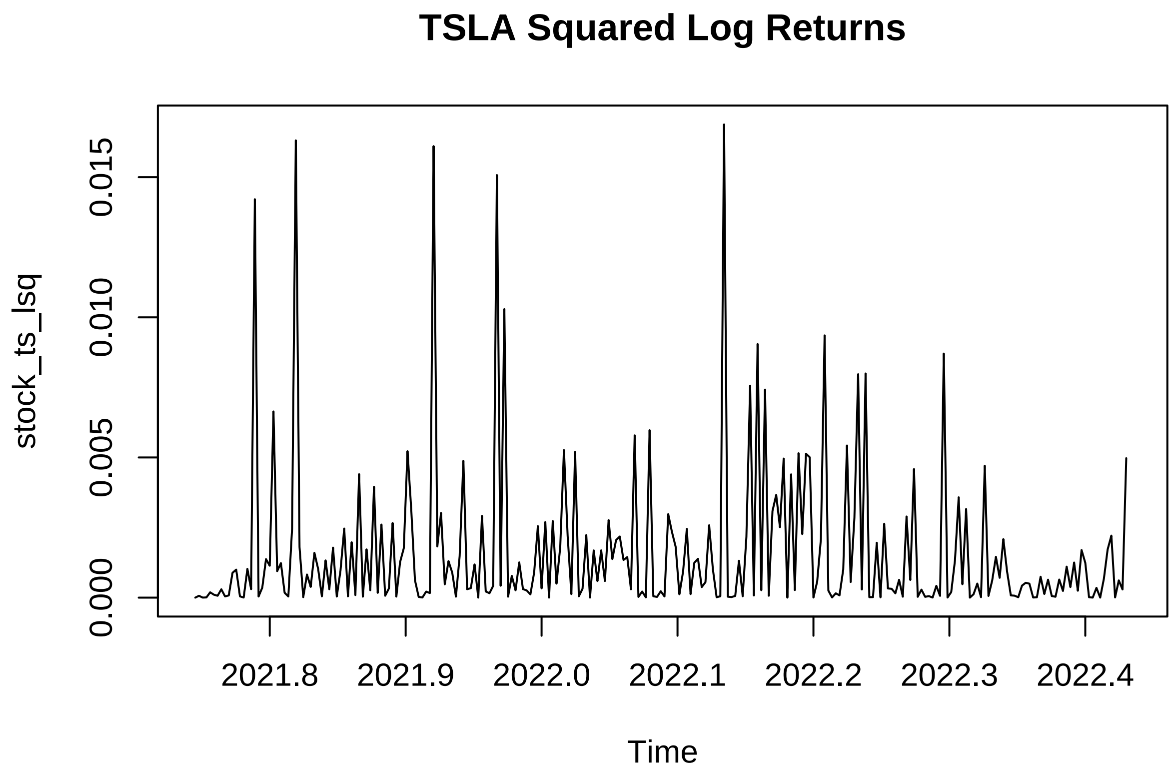

TSLA Returns
stock_ts
40
20
0
−20
−40
2021.8
2021.9
2022.0
2022.1
2022.2
2022.3
2022.4
Time




For TSLA returns, we see slight non-stationary characteristics. The ACF and PACF plots exhibit spikes outside the white noise error bounds at around lag 0.02 and 0.05. The time series plot of TSLA returns also exhibits some evidence of heteroskedasticity, with a truncated shape of the time series. The earlier returns exhibit a wider variance than the later returns.

Thus, we decided to square the logged returns to alleviate the slight non-stationary characteristics and heteroskedasticity.


TSLA Squared Log Returns
stock_ts_lsq
0.000
0.005
0.010
0.015
2021.8
2021.9
2022.0
2022.1
2022.2
2022.3
2022.4
Time


The square of logged TSLA returns exhibits only one ACF and PACF spike at around lag 0.03 and the plot of the square of logged TSLA returns no longer has a truncated shape.

So, we continued with using the square of logged TSLA returns to represent the measure of TSLA stock price in our model.

## 2.5 GARCH Model

Based on the non-constant variance and non-stationarity seen from the EDA for daily sentiment, we decided to use a GARCH model. A GARCH (generalized autoregressive conditionally heteroskedastic) model uses values of the past observations and variances to model the variance at time t. GARCH is used commonly with financiald ata because of the high volatility. Since we calculated the order of the model from running auto.arima(), we decided to fit a GARCH(1, 1) model to returns, the difference of the log of the adjusted closing price.

### 2.5.1 auto.arima()

The auto.arima() function was applied on the squared logged returns, which is the difference of the log of the adjusted closing price squared. This helped identify if there was autoregressive or moving average pattern in the votatility of the returns and helps determine the order of the GARCH model.

The best ARIMA model returned by the function was ARIMA(1,0,1) with a non-zero mean. The output of the function can be found in the Appendix.

### 2.6 VAR

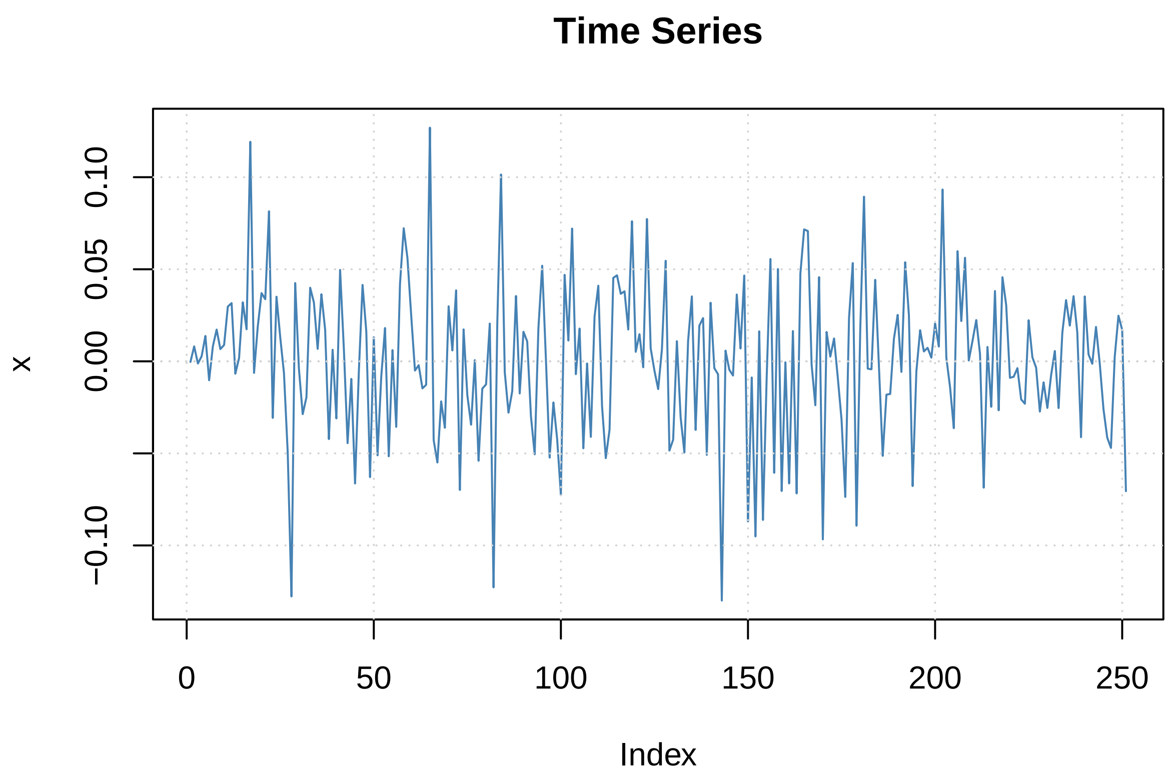
We used a VAR (Vector Autoregression) model to perform feature selection for linear modeling. After running the VARselect() function, we found that the lagged values of daily sentiment and lagged values of returns were the selected variables. Thus, we use these two variables at specific lags in two linear models: one linear model to predict TSLA stock returns and one to predict the average daily sentiment of Tesla tweets. The function output is included in the Appendix.

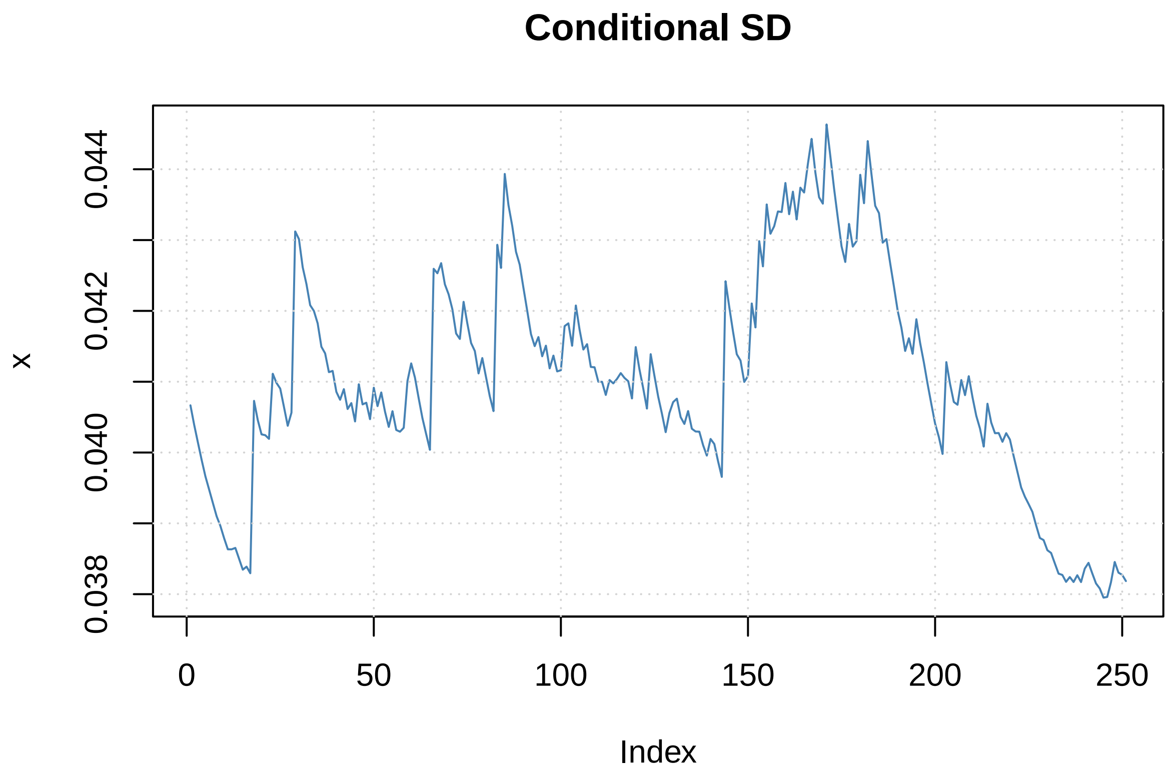
## 3 Results

The summary of our GARCH(1,1) model is below.

```
##
## Title:
##  GARCH Modelling
##
## Call:
##  garchFit(formula = ~garch(1, 1), data = stock_ts, trace = FALSE)
##
## Mean and Variance Equation:
##  data ~ garch(1, 1)
## <environment: 0x128a8fc78>
##  [data = stock_ts]
##
## Conditional Distribution:
##  norm
##
## Coefficient(s):
##         mu       omega      alpha1       beta1
## 0.00034657  0.00007856  0.01438639  0.93899717
##
## Std. Errors:
##  based on Hessian
##
## Error Analysis:
##         Estimate  Std. Error  t value Pr(>|t|)
## mu     3.466e-04   2.557e-03    0.136    0.892
## omega  7.856e-05   8.856e-05    0.887    0.375
## alpha1 1.439e-02   1.997e-02    0.720    0.471
## beta1  9.390e-01   5.494e-02   17.090   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Log Likelihood:
##  448.0723    normalized:  1.785149
##
## Description:
##  Sun Apr 21 21:54:20 2024 by user:
##
##
## Standardised Residuals Tests:
##                                    Statistic     p-Value
```

```
##  Jarque-Bera Test   R    Chi^2  11.8733436 0.002640804
##  Shapiro-Wilk Test  R    W       0.9837818 0.005887743
##  Ljung-Box Test     R    Q(10)  17.7126029 0.060009507
##  Ljung-Box Test     R    Q(15)  19.0600447 0.211025838
##  Ljung-Box Test     R    Q(20)  27.7444493 0.115588155
##  Ljung-Box Test     R^2  Q(10)  11.4358694 0.324582474
##  Ljung-Box Test     R^2  Q(15)  30.6866295 0.009672734
##  Ljung-Box Test     R^2  Q(20)  33.2456537 0.031704058
##  LM Arch Test       R    TR^2   21.5516296 0.042863209
##
## Information Criterion Statistics:
##       AIC       BIC       SIC      HQIC
## -3.538425 -3.482243 -3.538922 -3.515816
```






Based on the GARCH(1,1) fit, we see that only the GARCH coefficient of 0.939 is significant at $alpha = 0.05$ level($p \approx 0$). This model suggests that the volatility of the Tesla's stock returns is best predicted by the lagged conditional variance since other coefficients are not statistically significant. If we take a look at the conditional volatility plot, we indeed see that the conditional volatility is in different clusters of high and low volatility, meaning that it exhibits volatility clustering.

Going further into modeling the Tesla's stock price, we used the lag 1 and lag 2 terms of the returns, and also the lag 1 and lag 2 of the daily sentiment scores of the tweets. Our model is:

$$\text{Return} = \beta_0 + \beta_1 \text{Return}_1 + \beta_2 \text{Return}_2 + \beta_3 \text{sentiment} + \beta_4 \text{sentiment}_1 + \beta_5 \text{sentiment}_2$$

The linear model output is below.

| | Estimate | Std. Error | t value | Pr(>\|t\|) |
|---|---|---|---|---|
| **(Intercept)** | -0.001044 | 0.004789 | -0.2179 | 0.8277 |
| **Returns_l1** | -0.0182 | 0.06541 | -0.2783 | 0.781 |
| **Returns_l2** | 0.02776 | 0.06567 | 0.4228 | 0.6728 |
| **daily_sentiment** | -0.002767 | 0.01051 | -0.2634 | 0.7925 |
| **daily_sentiment_l1** | -0.01416 | 0.01044 | -1.356 | 0.1763 |
| **daily_sentiment_l2** | 0.02117 | 0.01028 | 2.06 | 0.04045 |

Table 2: Fitting linear model: Returns ~ Returns_l1 + Returns_l2 + daily_sentiment + daily_sentiment_l1 + daily_sentiment_l2

| Observations | Residual Std. Error | $R^2$ | Adjusted $R^2$ |
|---|---|---|---|
| 249 | 0.04085 | 0.02186 | 0.001734 |

Based on the output, we see that only the L2 term for daily sentiment score is significant with estimate of 0.02($p$=0.0405). The adjusted R squared value is 0.001734, which means that we can only explain the 0.1734% of the variability in the response value. Thus, this model is not very helpful in predicting the stock returns.

For predicting the daily sentiment score based on its previous lagged terms and the returns, the model is:

$$\text{sentiment} = \beta_0 + \beta_1\text{sentiment}_1 + \beta_2\text{sentiment}_2 + \beta_3\text{Return} + \beta_4\text{Return}_1 + \beta_5\text{Return}_2$$

The second linear model output is below:

| | Estimate | Std. Error | t value | Pr(>\|t\|) |
|---|---|---|---|---|
| **(Intercept)** | 0.1871 | 0.02666 | 7.019 | 2.223e-11 |
| **Returns** | -0.1031 | 0.3916 | -0.2634 | 0.7925 |
| **Returns_l1** | 1.311 | 0.3905 | 3.358 | 0.0009101 |
| **Returns_l2** | 0.7894 | 0.3979 | 1.984 | 0.04837 |
| **daily_sentiment_l1** | 0.1618 | 0.06313 | 2.564 | 0.01096 |
| **daily_sentiment_l2** | 0.1585 | 0.06247 | 2.537 | 0.01182 |

Table 4: Fitting linear model: daily_sentiment ~ Returns + Returns_l1 + Returns_l2 + daily_sentiment_l1 + daily_sentiment_l2

| Observations | Residual Std. Error | $R^2$ | Adjusted $R^2$ |
|---|---|---|---|
| 249 | 0.2494 | 0.12 | 0.1019 |

The lagged 1 and 2 terms of both returns and daily sentiment score are significant, which intuitively makes sense. However, this model is not as useful since we cannot make profit off of predicting the sentiment as opposed to predicting the stock returns.

# 4 Discussion

Our goal for this project was to predict Tesla's stock price based on the sentiment of the tweets. For the first linear model, we are predicting returns based on past returns, current daily sentiment, past daily sentiment. We see only one statistically significant coefficient, which means none of the other features have a statistically significant coefficient other than L2 daily sentiment. However, L2 daily sentiment has a small magnitude of 0.02, reflecting a weak correlation between returns and L2 daily sentiment. In the second linear model, we are predicting daily sentiment using returns, past returns, past daily sentiment. Almost all of the features are statistically significant value other than current returns. The adjusted $R^2$ of 10% suggests that these features account for 10% of the variability of daily sentiment. Past returns and past daily sentiment Granger causes current daily sentiment, i.e. past returns and past daily sentiment can be used to predict the daily sentiment for the current day. However, our original hypothesis was that we would be able to predict returns and stock price using past returns and daily sentiment, which is the opposite direction of Granger causality.

The findings from our model indicate a limited capacity for using tweet sentiment and frequency to predict Tesla's stock price. The significant but minimal correlation found in L2 daily sentiment suggests that while some predictive power exists, it is not strong enough to influence financial decisions effectively. The ability to predict daily sentiment from past stock prices and sentiment highlights potential areas for deeper exploration into the dynamics between social media and financial markets.

## 4.1 Conclusion

TSLA Stock returns cannot be predicted based on the current or lagged 1 values of daily sentiment or lagged 1 or 2 values for returns. The coefficient for lagged 2 daily sentiment is statistically significant but too small to be useful. Daily sentiment cannot be predicted based on current stock returns but can be explained by the lagged 1 or 2 values of the stock price or daily sentiment.

## 4.2 Future Work

As we move forward, our project will concentrate on enhancing the predictive models and broadening the analytical scope to better understand the relationship between public opinion and stock market behavior. Future efforts will include incorporating broader market performance indicators to understand Tesla's stock movements within the larger financial ecosystem. We will also explore longer temporal analyses to determine whether the effects of negative social media events on stock prices are only temporary or if they have a lasting impact. Additionally, we plan to incorporate tweet engagement metrics such as likes, retweets, and views, which may offer deeper insights into how social media dynamics can influence stock price movements. Through these improvements, we aim to build stronger predictive tools for financial market analysis, refining our approach to use the predictive power of social media sentiment more effectively.

...

# Appendix

## 4.3 Automatic ARIMA output

The output from auto.arima() is below.

```
## Series: stock$Returns^2
## ARIMA(1,0,1) with non-zero mean
##
## Coefficients:
##           ar1     ma1    mean
##       -0.6196  0.5105  0.0017
## s.e.   0.1958  0.2102  0.0002
##
## sigma^2 = 7.706e-06:  log likelihood = 1122.91
## AIC=-2237.83   AICc=-2237.66   BIC=-2223.72
##
## Training set error measures:
##                         ME        RMSE         MAE       MPE     MAPE      MASE
## Training set -4.434692e-07 0.002759389 0.001733483 -17226.44 17257.83 0.7116065
##                    ACF1
## Training set 0.01881578
```

## 4.4 VAR function output

The output from VARselect() is below.

- **names**: *Returns* and *daily_sentiment*
- **varresult**:
  - **Returns**:

| | Estimate | Std. Error | t value | Pr(>\|t\|) |
|---|---|---|---|---|
| **Returns.l1** | -0.02184 | 0.06382 | -0.3422 | 0.7325 |
| **daily_sentiment.l1** | -0.01461 | 0.01028 | -1.421 | 0.1565 |
| **Returns.l2** | 0.02559 | 0.06502 | 0.3935 | 0.6943 |
| **daily_sentiment.l2** | 0.02074 | 0.01013 | 2.048 | 0.04161 |
| **const** | -0.001562 | 0.004358 | -0.3584 | 0.7204 |

Table 6: Fitting linear model: y ~ -1 + .

| Observations | Residual Std. Error | $R^2$ | Adjusted $R^2$ |
|---|---|---|---|
| 249 | 0.04078 | 0.02158 | 0.005541 |

  - **daily_sentiment**:

| | Estimate | Std. Error | t value | Pr(>\|t\|) |
|---|---|---|---|---|
| **Returns.l1** | 1.314 | 0.3897 | 3.371 | 0.0008695 |

| | Estimate | Std. Error | t value | Pr(>\|t\|) |
|---|---|---|---|---|
| **daily_sentiment.l1** | 0.1634 | 0.06275 | 2.603 | 0.009799 |
| **Returns.l2** | 0.7868 | 0.397 | 1.982 | 0.04862 |
| **daily_sentiment.l2** | 0.1563 | 0.06183 | 2.529 | 0.01208 |
| **const** | 0.1873 | 0.02661 | 7.04 | 1.942e-11 |

Table 8: Fitting linear model: y ~ -1 + .

| Observations | Residual Std. Error | $R^2$ | Adjusted $R^2$ |
|---|---|---|---|
| 249 | 0.249 | 0.1198 | 0.1054 |

- **covres**:

| | Returns | daily_sentiment |
|---|---|---|
| **Returns** | 0.001663 | -0.0001715 |
| **daily_sentiment** | -0.0001715 | 0.06198 |

- **corres**:

| | Returns | daily_sentiment |
|---|---|---|
| **Returns** | 1 | -0.01689 |
| **daily_sentiment** | -0.01689 | 1 |

- **logLik**: *441.4*
- **obs**: *249*
- **roots**: *0.5583*, *0.3913*, *0.2375* and *0.2375*
- **type**: const
- **call**: `VAR(y = df, p = optimal.lags)`

```
###########################
# STYLE EDITS: IGNORE THIS
###########################

# normally you'll want to include this with the libraries at the beginning of your document
knitr::opts_chunk$set(message = FALSE) # include this if you don't want markdown to knit messages
knitr::opts_chunk$set(warning = FALSE) # include this if you don't want markdown to knit warnings
knitr::opts_chunk$set(echo = FALSE) # set echo = FALSE to hide code from output
suppressMessages(library(tidyverse))
suppressMessages(library(lubridate))
suppressMessages(library(tidytext))
suppressMessages(library(textdata))
suppressMessages(library(dplyr))
suppressMessages(library(quantmod))
suppressMessages(library(fGarch))
suppressMessages(library(pander))
# Gather stock tweets data
```

```r
#tweets <- read.csv("/Users/divyarao/time-series-tweets-stocks/data/stock_tweets.csv")
tweets <- read.csv("/Users/marionhaney/Library/Mobile Documents/com~apple~CloudDocs/CMU/36671 Time Seri
# Retaining the year, month, day
tweets$day <- as.Date(tweets$Date,"%Y-%m-%d %H:%M:%S")
# Filter to just Tesla tweets
tesla <- filter(tweets, tweets$Company.Name == "Tesla, Inc.")
num_tweets_tesla <- data.frame(table(tesla$day))
names(num_tweets_tesla) <- c("day", "num_tweets")
# Time series for number of Tesla tweets per day
tesla_ts <- ts(num_tweets_tesla$num_tweets,
                    start = c(2021, 273),
                    frequency = 365)
plot(tesla_ts, main = "Number of Tesla Tweets \n Per Day",
     ylab = "Number of Tweets")
par(mfrow=c(1,2))
acf(tesla_ts, main = "ACF of Number of Tesla Tweets")
pacf(tesla_ts, main = "PACF of Number of Tesla Tweets")
#df <- read.csv("/Users/divyarao/time-series-tweets-stocks/data/stock_tweets.csv")
df <- read.csv("/Users/marionhaney/Library/Mobile Documents/com~apple~CloudDocs/CMU/36671 Time Series/t

df <- df %>%
  filter(Stock.Name == 'TSLA') %>%
  mutate(Tweet.ID = row_number()) %>%
  dplyr::select(Tweet.ID, Date, Tweet)
#dim(df)
#names(df)

df$Date <-  ymd(substr(df$Date, 1, 10))
tweets <- data.frame(df)

# Sentiment analysis
map_bing_sentiment <- function(sentiment) {
  ifelse(sentiment %in% c("positive"), 1, ifelse(sentiment %in% c("negative"), -1, 0))
}

map_nrc_sentiment <- function(sentiment) {
  nrc_positive_sentiments <- c("positive", "anticipation", "surprise", "trust", "joy")
  nrc_negative_sentiments <- c("negative", "anger", "disgust", "fear", "sadness")
  ifelse(sentiment %in% nrc_positive_sentiments, 1,
         ifelse(sentiment %in% nrc_negative_sentiments, -1, 0))
}

tweet_tokens <- tweets %>%
  unnest_tokens(word, Tweet)

sentiments <- get_sentiments("bing") %>% mutate(sentiment_score = map_bing_sentiment(sentiment))

tweets_sentiment <- tweet_tokens %>%
  inner_join(sentiments, by = "word", relationship = "many-to-many")  %>%
  distinct(Tweet.ID, Date, word, .keep_all = TRUE)

tweets_sentiment_summary <- tweets_sentiment %>%
  group_by(Tweet.ID, Date) %>%
```

```r
  summarise(sentiment_score = sum(sentiment_score, na.rm = TRUE), .groups = "drop")

daily_sentiment <- tweets_sentiment_summary  %>%
  group_by(Date) %>%
  summarise(daily_sentiment = mean(sentiment_score))
sentiment_ts <- ts(daily_sentiment$daily_sentiment,
                    start = c(2021, 273),
                    frequency = 365)
plot(sentiment_ts, main = "Daily Sentiment of Tesla Tweets",
     ylab = "Average Daily Sentiment Score")
par(mfrow=c(1,2))
acf(sentiment_ts, main = "ACF of Daily Sentiment")
pacf(sentiment_ts, main = "PACF of Daily Sentiment")
#df <- read.csv("/Users/divyarao/time-series-tweets-stocks/data/stock_yfinance_data.csv")
df <- read.csv("/Users/marionhaney/Library/Mobile Documents/com~apple~CloudDocs/CMU/36671 Time Series/t

df <- df %>%
  filter(Stock.Name == 'TSLA') %>%
  dplyr::select(Date, Adj.Close)
df$Date <- as.Date(df$Date)
df$Returns <- c(diff(df$Adj.Close), NA)
stock_ts <- ts(df$Returns,
                    start = c(2021, 273),
                    frequency = 365)
df$Returns <- c(diff(log(df$Adj.Close)), NA)
stock_ts_lsq <- ts(df$Returns^2,
                    start = c(2021, 273),
                    frequency = 365)
stock <- data.frame(df)
stock <- stock %>% na.omit()
# EDA TSLA returns
plot(stock_ts, main="TSLA Returns")
par(mfrow=c(1,2))
acf(stock_ts, main="ACF of TSLA Returns", na.action = na.pass)
pacf(stock_ts, main="PACF of TSLA Returns", na.action = na.pass)
plot(stock_ts_lsq, main="TSLA Squared Log Returns")
par(mfrow=c(1,2))
acf(stock_ts_lsq, main="ACF of TSLA \nSquared Log Returns", na.action = na.pass)
pacf(stock_ts_lsq, main="PACF of TSLA \nSquared Log Returns", na.action = na.pass)
stock_ts <- ts(stock$Returns,
               start=c(2021,9),
               frequency=365)
garch_model <- garchFit(~ garch(1,1), data=stock_ts, trace=FALSE)
summary(garch_model)

par(mfrow=c(1,1))
plot(garch_model, which = 1)
plot(garch_model, which = 2)

par(mfrow=c(1,1))
combined_data <- left_join(stock, daily_sentiment, by = "Date")
#cor(combined_data$daily_sentiment, combined_data$Returns, use = "complete.obs")
model <- lm(Returns ~ daily_sentiment, data = combined_data)
```

```r
#pander(summary(model))
suppressMessages(library(dplyr))

combined_data <- combined_data %>%
  arrange(Date) %>%
  mutate(
    Returns_l1 = lag(Returns, 1),
    Returns_l2 = lag(Returns, 2),
    daily_sentiment_l1 = lag(daily_sentiment, 1),
    daily_sentiment_l2 = lag(daily_sentiment, 2)
  )

model <- lm(Returns ~ Returns_l1 + Returns_l2 + daily_sentiment +
              daily_sentiment_l1 + daily_sentiment_l2, data = combined_data)

pander(summary(model))
model <- lm(daily_sentiment ~ Returns + Returns_l1 +
              Returns_l2 + daily_sentiment_l1 +
              daily_sentiment_l2, data = combined_data)

pander(summary(model))
suppressWarnings(library(forecast))

arma_rt_squared <- auto.arima(stock$Returns^2, max.p = 5, max.q = 5, max.order = 10,
                              stationary = T, seasonal = F, trace = F,
                              stepwise = F, approximation = F)
summary(arma_rt_squared)
suppressMessages(library(vars))
df <- combined_data[, c("Returns", "daily_sentiment")]
# VARselect
lag.select <- VARselect(df,
                        lag.max = 30,
                        type = "both")
optimal.lags <- lag.select$selection['AIC(n)']

# Fit the VAR model
var.model <- VAR(df, p = optimal.lags)

pander(summary(var.model))
```