

COVID-19 VS. POLITICAL TWEETS: Is there an effect of COVID-19 on the political tweets volumn during the initial days of the pandemic outbreak?

Vivian Bui

11/25/2021

Abstract

This paper attempts to study if there is an effect of COVID-19 on the number of political tweets during the date between 28 February 2020 and 27 March 2020. To answer this question, we first find the number of political tweets per day for the selected dates, then observe the general trend in the total tweet volumes during this period. Once we complete assessing the general trend, we select March 13 to be the cut-off date and a period of 2 days before and after the cut-off to further determine whether a major COVID-19 related event triggered the change in the volume of political tweets. We then conclude that the event that the Travel Ban on Non-US Citizens Travelling from Europe goes into effect and the announcement that COVID-19 is a National Emergency occurred on March 13 was correlated to the increase in the number of political posts on Twitter within two days after such announcements. However, though Twitters seemed to react more to politics within 2 days after the announcement compared to the period of two days before the event happened, the volume of political tweets during this period was declining compared to the expected volumes using the 7 days rolling average calculation. Once we have confirmed there is a correlation, we study the general political atmosphere on Twitter during the 5 days period before, during, and after the cutoff date. The computed sentiment scores despite reveal a general decrease in Twitters' positivity, no firm conclusion can be made on whether or not the event trigger such change in people reaction.

1. Introduction

Social media has empowered the approaches an individual can take to react and participate in politics. In the age of which more and more people leaning towards virtual platforms for their expression of free speech and their practice of democracy, social media and its subsidiaries become a prosperous source of information for social and behavioral studies. As a result, capturing the general atmosphere and reaction of the netizens at a selected point of time would reveal interesting and potentially important information for policy-makers and the public. 'The rapid and vast adoption of these technologies is changing [...] how we access information from the news, and how we organize to demand political change.' (Ortiz-Ospina, 2019)

The 2020 pandemic is time period that we want to study in this case. Its perpetual implication of threats to our daily life raises many questions for political, psychological, and social scientists whether if COVID-19 leaves an effect on the political reaction of the people, especially those expressed explicitly on social media platforms. With a similar curiosity mindset, we attempt to learn if a major COVID-19 event influences people's attitudes towards politics expressed on Twitter. As 'a micro-blogging system, Twitter is relevant in both private and public communication spheres' (Raamkumar et. all, 2018) and is an appropriate starting point for the our study. Our questions in this context, thus, is to determine whether a major COVID-19 event affects the number of political tweets on this social media platform and the general political sentiment of users during this period.

2. Literature Review

The majority of research papers focusing on the initial months of the coronavirus pandemic often centers to the months between December 2019 to the early of February 2020, with a lesser amount dive deep into the political behaviors of individuals, especially ones that are expressed on virtual platforms, during the initial period of the outbreak. The paper of Stern and Axt “Were Americans’ political attitudes linked to objective threats from covid-19? an examination of data from project implicit during initial months of the pandemic.”, thus, is one of a few research project using a similar time period and topic of study compared to ours.

The paper determines ‘whether objective COVID-19 threats (cases, deaths, and government restrictions) occurring over the initial months of the pandemic (February–June 2020) were associated with seven different assessments of political attitudes among Project Implicit users in the United States (N = 34,581).’ The data is derived from a separate project examining associations between symbolic and operational conservatism, in which participants come from all 50 U.S. states, among of those are 24,039 women, 10,400 men, 142 no gender specified; Mean(age) = 35.50 years, standard deviation [SD] = 15.22 years, range = 17–91 years. The paper concludes that there is lack evidence showing a meaningful connection between COVID-19 threats and people political attitudes in the initial months of the pandemic.

What Stern’s paper did not determine is the individual reaction reflected on social media during the time the event occurs. Furthermore, Stern’s data comes from surveys, while our study is observational. Our study, therefore, would add new perspective to Stern’s findings.

3. About the Data

The data is from the Department of Information Systems and Cyber Security, The University of Texas at San Antonio. Each dataset contains the information of total tweets in one day with the hashtag of COVID-19 and COVID-1 - related.

Each dataset has 11 variables:

- ID: The ID of tweet
- USER_SCREEN_NAME: The name appears on each tweet
- USER_NAME: Name of the account used to log in
- TWEET_TEXT: Content of a tweet
- FAVORITES: Number of favorites a tweet received
- IN_REPLY_TO: Number of replies
- CREATION_TIME: The date and time the tweet is created
- SOURCE: The source where the tweet is posted (i.e. Iphone, Webapp, Android, TweetDeck)
- RETWEETED_TO: The source that the tweet is retweeted to
- RETWEETS_COUNT: Number of times the tweet is retweeted
- RETWEET_FLAG: Yes or No. Flag whether the tweet is a retweeted one.

For the purpose of this study, we will only focus on two variables: ID and TWEET_TEXT.

4. Data Cleaning and Pre-processing

4.1. Load and Read Data

```
library(dplyr)
library(tidyverse)
library(fs)
library(stringr)
library(ggplot2)
library(hrbrthemes)
```

```
library(plotly)
library(viridis)
library(tm)
library(SentimentAnalysis)
library(SnowballC)
library(rtweet)
library(wordcloud2)
library(qdapRegex)
```

In the data pre-processing stage, first, we load and read all the datasets as a list of tibble. While reading each dataset into the list, we assign the date column (as “Count”) to each of the dataset as a bookmark for the date once we merge them together. When all loaded, we fully join all the datasets and randomly sample 10000 observations from each of the date.

```
#NOTE: Set this directory to where the files are located
file_paths <- fs::dir_ls("~/Desktop/Covid-data-csv")

#Initiate
file_contents <- list()
count = 0

#NOTE: Files are large. Please wait for loading.
for (i in seq_along(file_paths)){
  #ORDER column: to denote the date once datasets are merged
  count = count + 1
  file_contents[[i]] <- read_csv(file = file_paths[[i]]) %>% mutate(ORDER = count) %>% select(ID, TWEET)
}

#Rename the name of object in the list from path name to file name
file_contents <- set_names(file_contents, path_file(file_paths))
```

4.2. Preprocessing and Text Classification

‘Preprocess’ is a created function used to clean our text for the purpose of labeling data. For the classification task, we use Naive Bayes Classifier. To begin with, we use an already-labeled training dataset to calculate the frequency of words and the probability of whether a text is classified as “Politics” or “Others”. We then applied the same probability to classify our test data.

In our initial approach, we attempted to use a dictionary of words to instead calculate the probability for each category. However, the dictionary was determined to be out-of-date, and proved to be redundant when the frequency of words can be directly extracted from the training dataset. As a result, we favor this approach which relied entirely on Naive Bayes over a dictionary.

```
#Merge all datasets
df_merged <- file_contents %>% map(as_tibble) %>% reduce(full_join)

#Randomly sample 10000 observations from each date
df_sampled <- df_merged %>% group_by(ORDER) %>% sample_n(10000)

#Create a function to clean text
preprocess <- function(df) {
```

```

ids_df <- df %>% select(ID)
tweets_df <- df %>% select(TWEET_TEXT)

#-----CLEAN-----
# Get the text column
text <- tweets_df$TWEET_TEXT
# Set the text to lowercase
text <- tolower(text)
# Remove mentions, urls, emojis, numbers, punctuations, etc.
text <- gsub("@\\w+", "", text)
text <- gsub("https?://.+", "", text)
text <- gsub("\\d+\\w*\\d*", "", text)
text <- gsub("#\\w+", "", text)
text <- gsub("[^\\x01-\\x7F]", "", text)
text <- gsub("[:punct:]", " ", text)
# Remove spaces and newlines
text <- gsub("\\n", " ", text)
text <- gsub("^\\s+", "", text)
text <- gsub("\\s+$", "", text)
text <- gsub("[ \\t]+", " ", text)
# Put the data to a new column
tweets_df["fix_text"] <- text

#-----EXPORT RESULTS-----
#Combine tweets and IDs dataframe
tweets_df <- cbind(ID=ids_df$ID, tweets_df)
#Final
fin <- tweets_df %>% select(ID, fix_text)
#Export file for ML
write.table(fin, file=~ /Desktop/test.txt", row.names=F, sep=" ")
}

#Apply function
preprocess(df_sampled)

```

```

#NOTE: Set directories to where the files are located

# %load twitter_classify
""" This program uses the Naive Bayes Classifier to classify the validation set
after training on the training set. """

import string

# Extract actual necessary words from the tweet
def extract_tweet_words(tweet_words):
    words = []
    alpha_lower = string.ascii_lowercase
    alpha_upper = string.ascii_uppercase
    numbers = [str(n) for n in range(10)]
    for word in tweet_words:
        cur_word = ''
        for c in word:

```

```

        if (c not in alpha_lower) and (c not in alpha_upper) and (c not in numbers):
            if len(cur_word) >= 2:
                words.append(cur_word.lower())
            cur_word = ''
            continue
        cur_word += c
    if len(cur_word) >= 2:
        words.append(cur_word.lower())
    return words

# Get Training Data from the input file
def get_tweet_training_data():
    f = open(r'/Users/vivianbui/Desktop/Tweet-Classification/training.txt')
    training_data = []
    for l in f.readlines():
        l = l.strip()
        tweet_details = l.split()
        tweet_id = tweet_details[0]
        tweet_label = tweet_details[1]
        tweet_words = extract_tweet_words(tweet_details[2:])
        training_data.append([tweet_id, tweet_label, tweet_words])

    f.close()

    return training_data

# Get Test Data from the input file
def get_tweet_test_data():
    f = open(r'/Users/vivianbui/Desktop/Tweet-Classification/test.txt')
    validation_data = []
    for l in f.readlines():
        l = l.strip()
        tweet_details = l.split(' ')
        tweet_id = tweet_details[0]
        tweet_words = extract_tweet_words(tweet_details[1:])
        validation_data.append([tweet_id, '', tweet_words])

    f.close()

    return validation_data

# Get list of words in the training data
def get_words(training_data):
    words = []
    for data in training_data:
        words.extend(data[2])
    return list(set(words))

# Get Probability of each word in the training data
# If label is specified, find the probability of each word in the corresponding labelled tweets only
def get_tweet_word_prob(training_data, label = None):
    words = get_words(training_data)

```

```

freq = {}

for word in words:
    freq[word] = 1

total_count = 0
for data in training_data:
    if data[1] == label or label == None:
        total_count += len(data[2])
        for word in data[2]:
            freq[word] += 1

prob = {}
for word in freq.keys():
    prob[word] = freq[word]*1.0/total_count

return prob

# Get Probability of given label
def get_tweet_label_count(training_data, label):
    count = 0
    total_count = 0
    for data in training_data:
        total_count += 1
        if data[1] == label:
            count += 1
    return count*1.0/total_count

# Label the test data given the trained parameters Using Naive Bayes Model
def label_data(test_data, other_word_prob, politics_word_prob, other_prob, politics_prob):
    labels = []
    for data in test_data:
        data_prob_other = other_prob
        data_prob_politics = politics_prob

        for word in data[2]:
            if word in other_word_prob:
                data_prob_other *= other_word_prob[word]
                data_prob_politics *= politics_word_prob[word]
            else:
                continue

        if data_prob_other >= data_prob_politics:
            labels.append([data[0], 'Others', data_prob_other, data_prob_politics])
        else:
            labels.append([data[0], 'Politics', data_prob_other, data_prob_politics])

    return labels

# Print the labelled test data
def print_labelled_data(labels):
    f_out = open('/Users/vivianbui/Desktop/Tweet-Classification/Output.txt', 'w')
    for [tweet_id, label, prob_other, prob_politics] in labels:

```

```

        f_out.write('%s %s\n' % (tweet_id, label))

    f_out.close()

# Get the training and test data
training_data = get_tweet_training_data()
test_data = get_tweet_test_data()

# Get the probabilities of each word overall and in the two labels
word_prob = get_tweet_word_prob(training_data)
other_word_prob = get_tweet_word_prob(training_data, 'Others')
politics_word_prob = get_tweet_word_prob(training_data, 'Politics')

# Get the probability of each label
other_prob = get_tweet_label_count(training_data, 'Others')
politics_prob = get_tweet_label_count(training_data, 'Politics')

# Normalise for stop words
for (word, prob) in word_prob.items():
    other_word_prob[word] /= prob
    politics_word_prob[word] /= prob

# Label the test data and print it
test_labels = label_data(test_data, other_word_prob, politics_word_prob, other_prob, politics_prob)
print_labelled_data(test_labels)

```

Once we receive the output, we will be able to calculate the total number of political tweets per day. 'Pol_tweets' will be the main dataframe we use for our analysis.

```

#NOTE: Convert your output file from txt to csv
#Load output
df_labeled <- read.csv("~/Desktop/Tweet-Classification/Output.csv")

#Add date (ORDER)
date_labels <- df_sampled %>% select(ORDER)

#Combine labeled dataframe with labels for date
df_labeled <- cbind(date_labels, df_labeled)

#Calculate political tweets
pol_tweets <- df_labeled %>% group_by(ORDER) %>% mutate(Detect=str_detect(ID.Politics, "Politics")) %>%

#Drop rows where Detect == FALSE
pol_tweets <- pol_tweets %>% filter(Detect == TRUE) %>% select(ORDER, n)

```

5. Analysis

5.1. Selection of a Cutoff Date

To determine if there is a change in political tweets volume, a cutoff date must be selected to provide a midpoint for two periods. We use the COVID-19 timeline provided by AJMC for our consideration of the cutoff:

- February 3 — US Declares Public Health Emergency
- February 10 — China’s COVID-19 Deaths Exceed Those of SARS Crisis
- February 25 — CDC Says COVID-19 Is Heading Toward Pandemic Status
- March 6 — 21 Passengers on California Cruise Ship Test Positive
- March 11 — WHO Declares COVID-19 a Pandemic
- March 13 — Trump Declares COVID-19 a National Emergency
- March 13 — Travel Ban on Non-US Citizens Traveling From Europe Goes Into Effect
- March 17 — University of Minnesota Begins Testing Hydroxychloroquine
- March 17 — CMS Temporarily Expands Use of Telehealth
- March 17 — Administration Asks Congress to Send Americans Direct Financial Relief
- March 19 — California Issues Statewide Stay-at-Home Order
- March 24 — With Clinical Trials on Hold, Innovation Stalls
- March 25 — Reports Find Extended Shutdowns Can Delay Second Wave
- March 26 — Senate Passes CARES Act
- March 27 — Trump Signs CARES Act Into Law

Two criteria are created for the selection of the cutoff. Firstly, on the selected date, there must be a major COVID-19 event. Secondly, to avoid taking into account the reaction that is escalated from other events, there should be no major COVID-19 event happened during at least 2 to 3 days before and after the selected date.

Based on these criteria, March 13 is selected to be the cutoff date. The date range which included this cutoff will be later determined based on the exploration, or the general trend, of our data.

5.2. General Trend

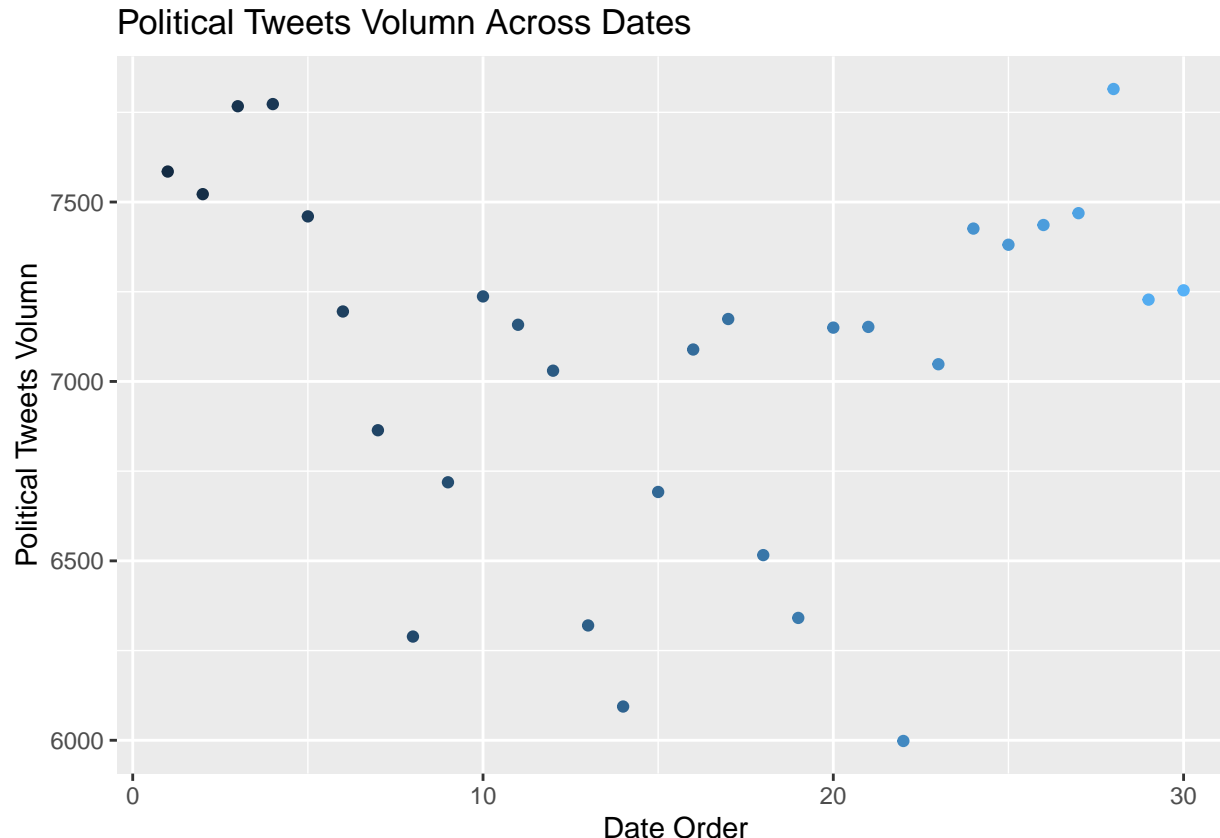
The plot of all political tweets volume shows a general downward and upward trend for the period before and after the cutoff, respectively. This downward and upward trend is even more visible for the period of two days before and after the cutoff date March 13. As a result, the period of 5 days between March 11 to March 15 will be the center of our analysis.

The plot, simultaneously, indicates a correlation between what happened on March 13 and the political tweets volume. The announcement that COVID-19 is a National Emergency and the ban on travel for non-citizens who came from Europe seems to correlate to the increase in the number of political posts on Twitter two days after. The change in the volume of political tweets, however, should be understood in a relational term. For this reason, we calculate the average number of political tweets volume of the 7 days before the considering date. This average of political tweets volume calculated from the rolling 7 days will be equal to the political tweets volume expected for each day. Then, we find the difference between the expected and the actual number of political tweets and denote the result as the “Abnormal Score”. “Abnormal score”, in essence, implies the distance, or how ‘abnormal’, our actual data is compared to what is expected to be the norm.

Plotting the abnormal score reveals that while the number of political tweets seems to decrease and increase in the period of 2 days before and after the cutoff respectively, an inverse trend takes place when we measure the trend in terms of abnormality. The abnormal score for the 2 days before March 13 was actually increasing, whereas the abnormal score for the 2 days after March 13 was decreasing. Such an inverse relationship between the political tweets volume and the abnormal score discovered during our selected date range points out some interesting insights for our context: the event occurred on March 13 could correlate to the underlying reason that caused the increase for an expected falling trend of the number of political tweets for the dates after the cutoff. Furthermore, even though March 13 might electrolyte the number of political tweets of Twitter users for the study of a selected 5-days period, people reaction towards politics on average was still less than what was expected reflecting upon a study for a 14-day period (7 days before and after the cutoff). This indicates that there might be other events or COVID-19 related events before the cutoff that caused even more intense political reactions of Twitter users than the events happened on March 13.

```
#Plot to view general trend
```

```
pol_tweets %>% ggplot(aes(x = ORDER, y = n, color = ORDER)) + geom_point() + ggtitle("Political Tweets Volumn Across Dates")
```



```

#Create column to denote cutoff date
Date_Order <- c()
Date_Order <- as.data.frame(seq(-14,15,1))

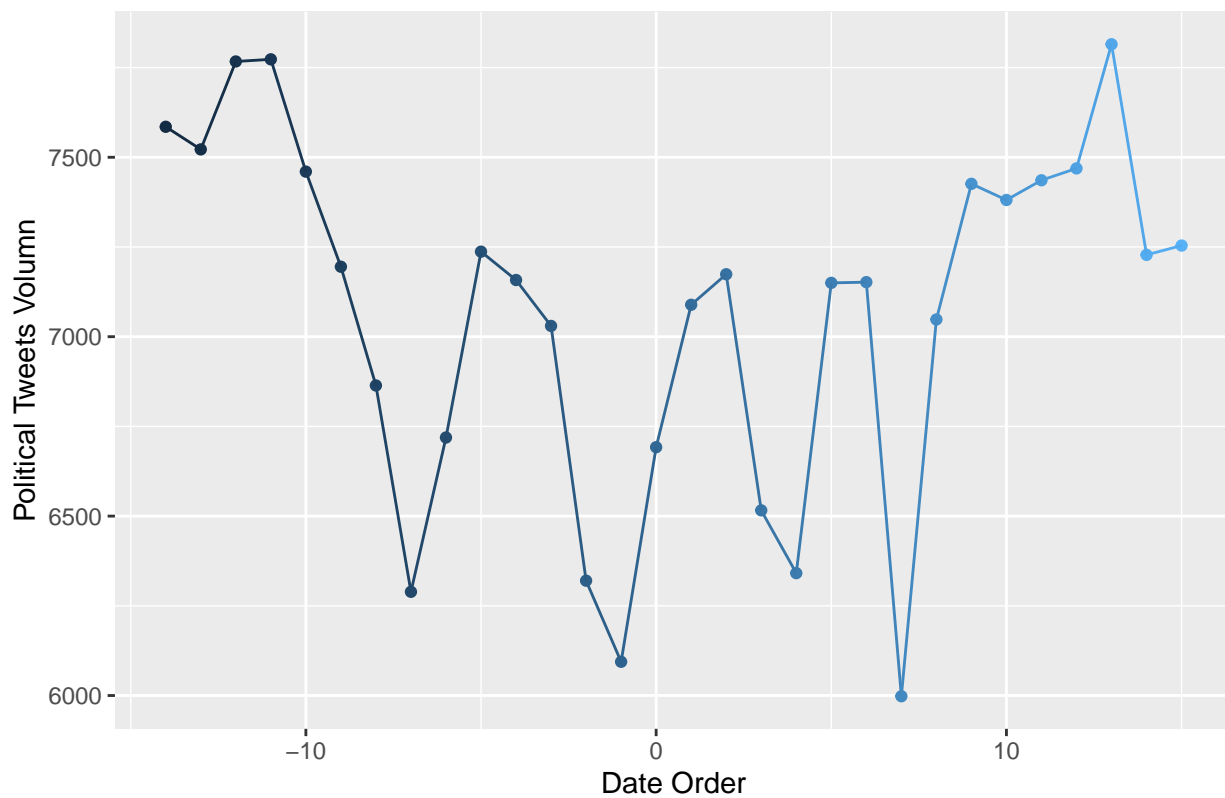
#Rename column
names(Date_Order)[1] <- 'Date_Order'

#Combine
pol_tweets <- cbind(Date_Order, pol_tweets)

#Re-plot data after choosing cut-off day
pol_tweets %>% ggplot(aes(x = Date_Order, y = n, color = Date_Order)) + geom_point() + geom_line() + gg

```

Political Tweets Volumns Across Dates (Dates adjusted for cutoff)

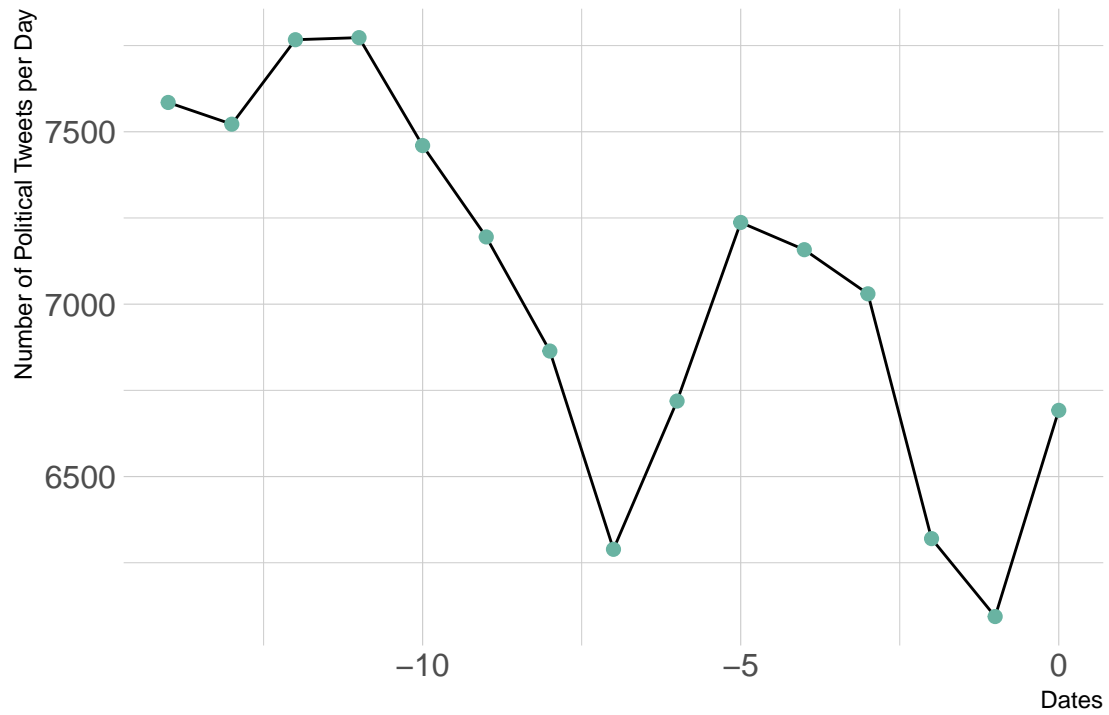


```

#Subset before and after cutoff date
before <- pol_tweets[1:15,]
after <- pol_tweets[16:30,]

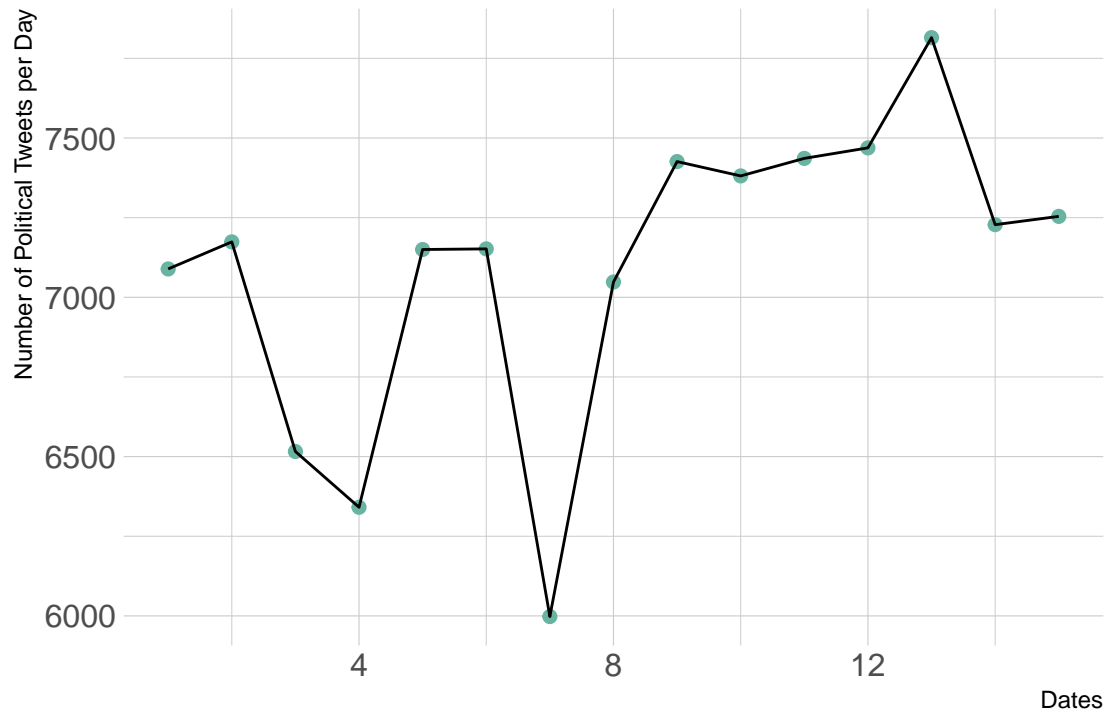
#Plot before
ggplot(data=before, aes(x=Date_Order, y=n)) + geom_line() + geom_point(size=2,color="#69b3a2") + labs(

```



#Plot after

```
ggplot(data=after, aes(x=Date_Order, y=n)) + geom_point(size=2, color="#69b3a2") + geom_line() + labs()
```

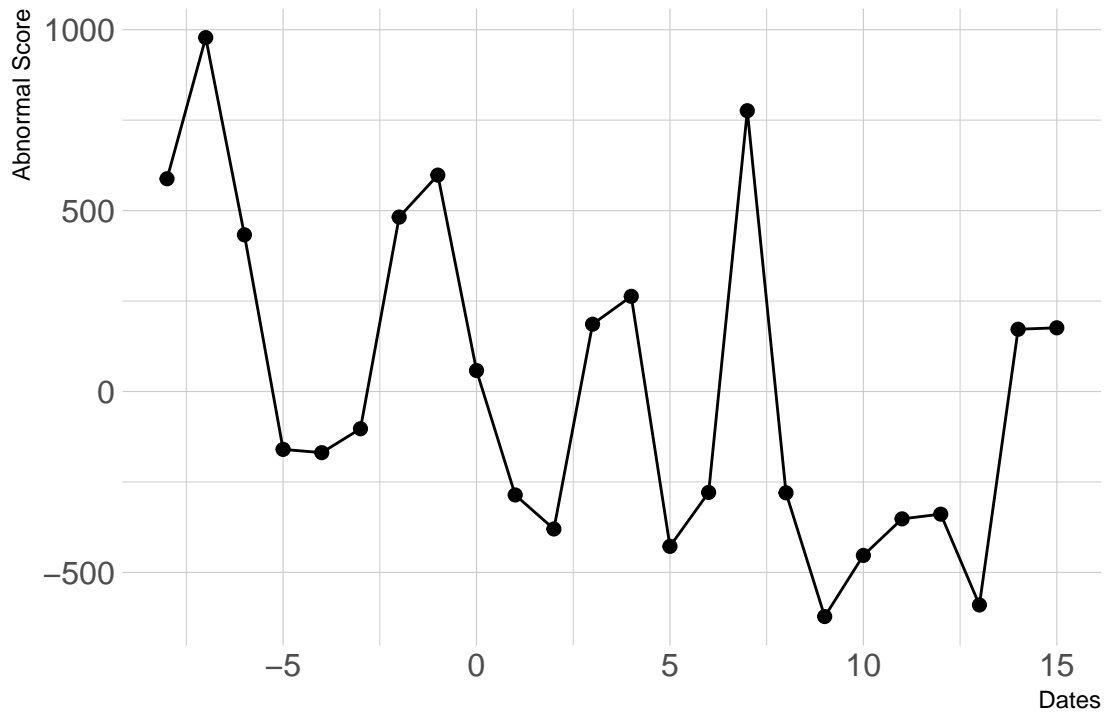


```
library(RcppRoll)
#Calculate rolling 7 days
pol_tweets$rolling_7_avg<- roll_mean(pol_tweets$n, n = 7, align = "right", fill = NA)
pol_tweets$rolling_7_avg <- format(pol_tweets$rolling_7_avg, digits = 2)

#Abnormal score
pol_tweets$Abnormal_score <- as.numeric(pol_tweets$rolling_7_avg) - pol_tweets$n

#Subset and drop NA
df_abnormal <- pol_tweets %>% select(Date_Order, Abnormal_score) %>% na.omit()

#Plot Abnormal score
ggplot(data=df_abnormal, aes(x=Date_Order, y=Abnormal_score)) + geom_point(size=2) + geom_line() + labs
```



5.3. The Political Atmosphere on Twitter during the Selected Date Range

a) The Frequency of Words Now the correlation between the COVID-19 event of March 13 and the change in the political tweets volume 2 days followed the event is confirmed, the next question is how the event affected the general political sentiment of Twitter users during the selected studying period?

To answer this question, we generate a wordcloud for each of the selected date range before and after our cutoff date. Wordcloud summarizes the most frequently repeated words across tweets, which might help deepen our understanding of what are the keywords that related to people's concern in the topic of politics and COVID-19 and potentially.

It is noticeable that across the selected date, the political-related word that was mentioned most often is "Trump", with the frequency ranged from 839, 1014, 1342, 892, to 982 for the selected date period. Other than that, political words do not appear to be the highlights of the wordclouds. The wordclouds do not give sufficient information to capture the political concerns Twitter users have during the selected period.

b) Sentiment Analysis of the Selected Date Range The computed sentiment of each selected date range reveals that the overall sentiment score was, in effect, positive across dates. When counting the number of negative versus positive sentiment score of all tweets for each day, the majority of tweets shows a positive sentiment with more than 50% of tweets demonstrates positive scores for all selected dates. For the study period of 5 selected days, the average highest score is observed on March 13, and there was a steep decrease in the average score 2 days after the announcement of national emergency and the travel ban. There is no adequate evidence to conclude if such positive or decrease in the overall political sentiment score in these days correspond to the events on March 13 or not, however, one can observe a correlation between the COVID-19 announcement and the Twitter users' decrease of positive sentiment during this selected period of study.

```

#-----GET SENTIMENT SCORE FOR MARCH 11-----
#Subset March 11 [Date Order = -2, ORDER = 13]
mar11 <- df_sampled %>% filter(ORDER==13)
#Clean data
mar11$TWEET_TEXT <- gsub("@[[:alpha:]]*", "", mar11$TWEET_TEXT)
text_corpus11 <- Corpus(VectorSource(mar11$TWEET_TEXT))
text_corpus11 <- tm_map(text_corpus11, tolower)
text_corpus11 <- tm_map(text_corpus11, removeWords,
                        stopwords("english"))
text_corpus11 <- tm_map(text_corpus11, removePunctuation)
text_df11 <- data.frame(text_clean11 = get("content", text_corpus11),
                      stringsAsFactors = FALSE)
mar11 <- cbind.data.frame(mar11, text_df11)

#Sentiment
mar11_sentiment <- analyzeSentiment(mar11$text_clean11)
mar11_sentiment <- dplyr::select(mar11_sentiment,
                                SentimentGI, SentimentHE,
                                SentimentLM, SentimentQDAP,
                                WordCount)
mar11_sentiment <- dplyr::mutate(mar11_sentiment,
                                mean_sentiment = rowMeans(mar11_sentiment[, -5]))
mar11_sentiment <- dplyr::select(mar11_sentiment,
                                WordCount,
                                mean_sentiment)
mar11 <- cbind.data.frame(mar11, mar11_sentiment)

#Text Corpus for word cloud
text_df11 <-
  text_df11 %>%
  str_remove("\\n") %>% # remove linebreaks
  rm_twitter_url() %>% # Remove URLs
  rm_url() %>%
  str_remove_all("#\\S+") %>% # Remove any hashtags
  str_remove_all("@\\S+") %>% # Remove any @ mentions
  removeWords(stopwords("english")) %>% # Remove common words (a, the, it etc.)
  removeNumbers() %>%
  stripWhitespace() %>%
  removeWords(c("amp"))
textCorpus11 <- Corpus(VectorSource(text_df11)) %>% TermDocumentMatrix() %>% as.matrix()
textCorpus11 <- sort(rowSums(textCorpus11), decreasing=TRUE)
textCorpus11 <- data.frame(word = names(textCorpus11), freq=textCorpus11, row.names = NULL)

#Create a word cloud
wordcloud11 <- wordcloud2(data = textCorpus11, minRotation = 0, maxRotation = 0, ellipticity = 0.6)
wordcloud11

```



15


```

#Mean score of the day
mar12_mean <- mean(mar12$mean_sentiment, na.rm=TRUE)

#-----GET SENTIMENT SCORE FOR MARCH 13-----
#Subset March 13 [Date Order: 0, ORDER = 15]
mar13 <- df_sampled %>% filter(ORDER==15)
#Clean data
mar13$TWEET_TEXT <- gsub("@[:alpha:]*", "", mar13$TWEET_TEXT)
text_corpus13 <- Corpus(VectorSource(mar13$TWEET_TEXT))
text_corpus13 <- tm_map(text_corpus13, tolower)
text_corpus13 <- tm_map(text_corpus13, removeWords,
                        stopwords("english"))
text_corpus13 <- tm_map(text_corpus13, removePunctuation)
text_df13 <- data.frame(text_clean13 = get("content", text_corpus13),
                      stringsAsFactors = FALSE)
mar13 <- cbind.data.frame(mar13, text_df13)

#Sentiment
mar13_sentiment <- analyzeSentiment(mar13$text_clean13)
mar13_sentiment <- dplyr::select(mar13_sentiment,
                                SentimentGI, SentimentHE,
                                SentimentLM, SentimentQDAP,
                                WordCount)
mar13_sentiment <- dplyr::mutate(mar13_sentiment,
                                mean_sentiment = rowMeans(mar13_sentiment[, -5]))
mar13_sentiment <- dplyr::select(mar13_sentiment,
                                WordCount,
                                mean_sentiment)
mar13 <- cbind.data.frame(mar13, mar13_sentiment)

#Text Corpus for word cloud
text_df13 <-
  text_df13 %>%
  str_remove("\\n") %>%                # remove linebreaks
  rm_twitter_url() %>%                # Remove URLs
  rm_url() %>%
  str_remove_all("#\\S+") %>%          # Remove any hashtags
  str_remove_all("@\\S+") %>%          # Remove any @ mentions
  removeWords(stopwords("english")) %>% # Remove common words (a, the, it etc.)
  removeNumbers() %>%
  stripWhitespace() %>%
  removeWords(c("amp"))
textCorpus13 <- Corpus(VectorSource(text_df13)) %>% TermDocumentMatrix() %>% as.matrix()
textCorpus13 <- sort(rowSums(textCorpus13), decreasing=TRUE)
textCorpus13 <- data.frame(word = names(textCorpus13), freq=textCorpus13, row.names = NULL)

#Create a word cloud
wordcloud13 <- wordcloud2(data = textCorpus13, minRotation = 0, maxRotation = 0, ellipticity = 0.6)
wordcloud13

```



```
#Mean score of the day
mar13_mean <- mean(mar13$mean_sentiment, na.rm=TRUE)

#-----GET SENTIMENT SCORE FOR MARCH 14-----
#Subset March 14 [Date Order: 1, ORDER = 16]
mar14 <- df_sampled %>% filter(ORDER==16)
#Clean data
mar14$TWEET_TEXT <- gsub("@[[:alpha:]]*", "", mar14$TWEET_TEXT)
text_corpus14 <- Corpus(VectorSource(mar14$TWEET_TEXT))
text_corpus14 <- tm_map(text_corpus14, tolower)
text_corpus14 <- tm_map(text_corpus14, removeWords,
                        stopwords("english"))
text_corpus14 <- tm_map(text_corpus14, removePunctuation)
text_df14 <- data.frame(text_clean14 = get("content", text_corpus14),
                      stringsAsFactors = FALSE)
mar14 <- cbind.data.frame(mar14, text_df14)

#Sentiment
mar14_sentiment <- analyzeSentiment(mar14$text_clean14)
mar14_sentiment <- dplyr::select(mar14_sentiment,
                                SentimentGI, SentimentHE,
                                SentimentLM, SentimentQDAP,
                                WordCount)
mar14_sentiment <- dplyr::mutate(mar14_sentiment,
                                mean_sentiment = rowMeans(mar14_sentiment[, -5]))
```



```

#Mean score of the day
mar14_mean <- mean(mar14$mean_sentiment, na.rm=TRUE)

#-----GET SENTIMENT SCORE FOR MARCH 15-----
#Subset March 15 [Date Order: 2, ORDER = 17]
mar15 <- df_sampled %>% filter(ORDER==17)
#Clean data
mar15$TWEET_TEXT <- gsub("@[:alpha:]]*", "", mar15$TWEET_TEXT)
text_corpus15 <- Corpus(VectorSource(mar15$TWEET_TEXT))
text_corpus15 <- tm_map(text_corpus15, tolower)
text_corpus15 <- tm_map(text_corpus15, removeWords,
                        stopwords("english"))
text_corpus15 <- tm_map(text_corpus15, removePunctuation)
text_df15 <- data.frame(text_clean15 = get("content", text_corpus15),
                      stringsAsFactors = FALSE)
mar15 <- cbind.data.frame(mar15, text_df15)

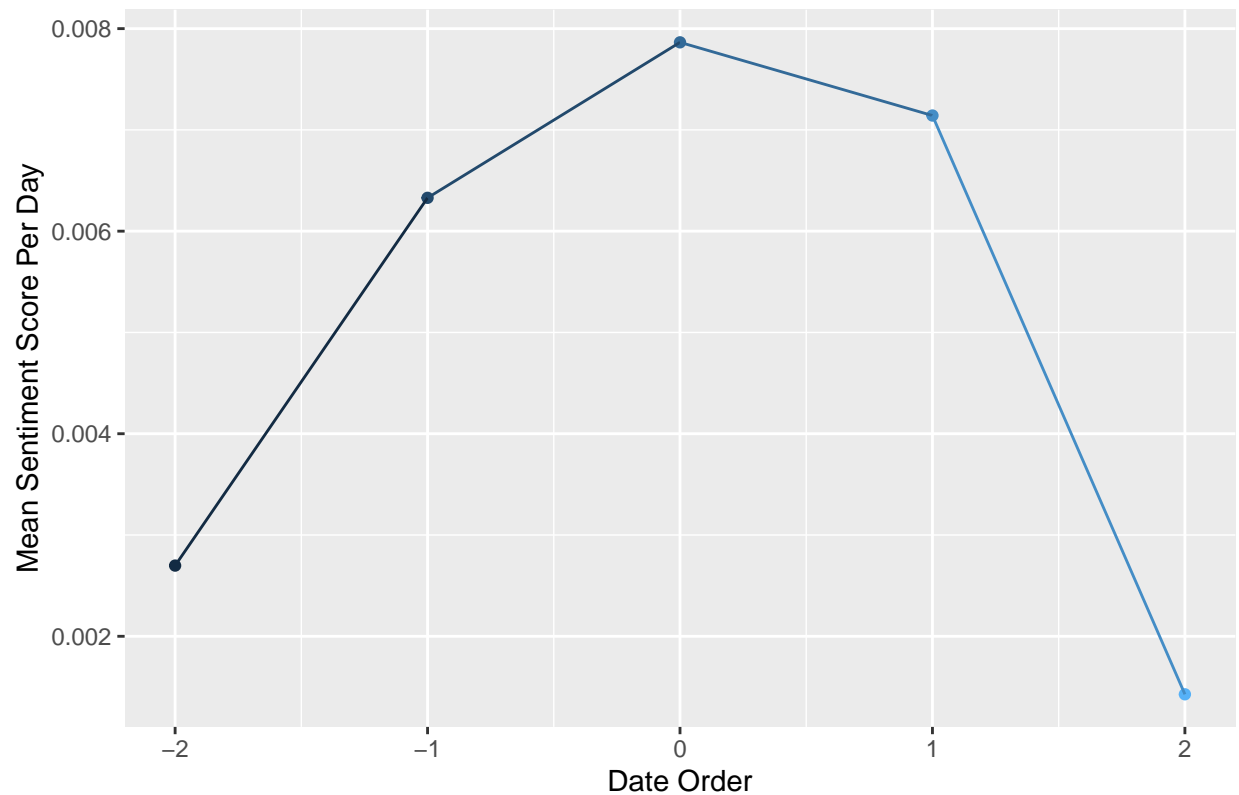
#Sentiment
mar15_sentiment <- analyzeSentiment(mar15$text_clean15)
mar15_sentiment <- dplyr::select(mar15_sentiment,
                                SentimentGI, SentimentHE,
                                SentimentLM, SentimentQDAP,
                                WordCount)
mar15_sentiment <- dplyr::mutate(mar15_sentiment,
                                mean_sentiment = rowMeans(mar15_sentiment[, -5]))
mar15_sentiment <- dplyr::select(mar15_sentiment,
                                WordCount,
                                mean_sentiment)
mar15 <- cbind.data.frame(mar15, mar15_sentiment)

#Text Corpus for word cloud
text_df15 <-
  text_df15 %>%
  str_remove("\\n") %>%                # remove linebreaks
  rm_twitter_url() %>%                 # Remove URLs
  rm_url() %>%
  str_remove_all("#\\S+") %>%          # Remove any hashtags
  str_remove_all("@\\S+") %>%          # Remove any @ mentions
  removeWords(stopwords("english")) %>% # Remove common words (a, the, it etc.)
  removeNumbers() %>%
  stripWhitespace() %>%
  removeWords(c("amp"))
textCorpus15 <- Corpus(VectorSource(text_df15)) %>% TermDocumentMatrix() %>% as.matrix()
textCorpus15 <- sort(rowSums(textCorpus15), decreasing=TRUE)
textCorpus15 <- data.frame(word = names(textCorpus15), freq=textCorpus15, row.names = NULL)

#Create a word cloud
wordcloud15 <- wordcloud2(data = textCorpus15, minRotation = 0, maxRotation = 0, ellipticity = 0.6)
wordcloud15

```


Average Sentiment Score Two Days Before and After Cutoff



```
#Count negative vs. positive score
mar11 %>% count(mean_sentiment <0)
```

```
##   mean_sentiment < 0    n
## 1                FALSE 5824
## 2                 TRUE 4173
## 3                  NA    3
```

```
mar12 %>% count(mean_sentiment <0)
```

```
##   mean_sentiment < 0    n
## 1                FALSE 6076
## 2                 TRUE 3917
## 3                  NA    7
```

```
mar13 %>% count(mean_sentiment <0)
```

```
##   mean_sentiment < 0    n
## 1                FALSE 6176
## 2                 TRUE 3824
```

```
mar14 %>% count(mean_sentiment <0)
```

```
## mean_sentiment < 0      n
## 1                     FALSE 6014
## 2                      TRUE 3986
```

```
mar15 %>% count(mean_sentiment < 0)
```

```
## mean_sentiment < 0      n
## 1                     FALSE 5682
## 2                      TRUE 4318
```

6. Conclusion

We conclude that there is a correlation between the COVID-19 related event occurred on March 13 and the people reaction on Twitters, either in terms of the number of political tweets or their general sentiment. Even though there was an upward trend in the volume of tweets two days after the announcement and the travel ban took place, the overall volume of tweets when being observed in an expanding period reveals a general decreasing volume. The sentiment analysis then provides some insights into how such correlated effect influence the general sentiment of tweets during the selected date range. The calculated results indicate that there is a correlation between the event happened on the cutoff and the decrease in the general positive sentiment score of Twitter users. In general, the results of this study provide insights into the study of COVID-19 versus political behaviors on Twitter during the initial months of the outbreak, which potentially contribute to the study of political behaviors of people across social media platforms and the virtual political behaviors during the break of COVID-19. When being accompanied with other findings in the related topics of political behaviors on Twitter and other social media platforms during the time of COVID-19, our findings can provide insights for policy makers in how a political decision drives public reaction and behavioral responses. Finally, our technique of data manipulation, text analysis, cleaning and classifying big data will contribute to the field of text mining of big data for data scientists.

7. Acknowledgement

We thank Dr. H. Raghav Rao, AT&T Distinguished Chair in Infrastructure Assurance and Security at the Department of Information Systems and Cyber Security, The University of Texas at San Antonio; and Dr. Naga Vemprala, Assistant Professor of Information Systems at Pamplin School of Business, University of Portland for providing the Twitter dataset for this research.

8. Reference

- A timeline of covid-19 developments in 2020. AJMC. (2021, January 1). Retrieved from <https://www.ajmc.com/view/a-timeline-of-covid19-developments-in-2020>.
- Feinerer, Ingo, and Kurt Hornik. 2018. Tm: Text Mining Package. <https://CRAN.R-project.org/package=tm>.
- Kearney, Michael W. 2018. Rtweet: Collecting Twitter Data. <https://CRAN.R-project.org/package=rtweet>.
- Lang, Dawei. 2018. Wordcloud2: Create Word Cloud by htmlWidget. <https://github.com/lchiffon/wordcloud2>.
- Ortiz-Ospina, E. (2019). The rise of Social Media. Our World in Data. Retrieved from <https://ourworldindata.org/rise-of-social-media>.
- Raamkumar et al. (2018). Understanding the Twitter Usage of Humanities and Social Sciences Academic Journals.

Rinker, Tyler. 2017. QdapRegex: Regular Expression Removal, Extraction, and Replacement Tools. <https://CRAN.R-project.org/package=qdapRegex>.

Sipra, V. (2020, January 21). Twitter sentiment analysis and visualization using R. Medium. Retrieved from <https://towardsdatascience.com/twitter-sentiment-analysis-and-visualization-using-r-22e1f70f6967>.

Silge, J. and Robinson, D. Text Mining with R: A Tidy Approach (2017). O'Reilly Media, Inc.

Stern C, Axt J. Were Americans' Political Attitudes Linked to Objective Threats From COVID-19? An Examination of Data From Project Implicit During Initial Months of the Pandemic. *Pers Soc Psychol Bull*. 2021 Oct 20:1461672211052121. doi: 10.1177/01461672211052121. Epub ahead of print. PMID: 34668457.