Twitter Sentiment Analysis

Gauging the Public's Sentiment on Country Leaders Using Natural Language Processing

By Alisha Peermohamed | Spring 2020

Project Overview

Project Overview and Goals:

For my term project, I decided to perform a Twitter Sentiment Analysis on the leaders of different countries. Given the current coronavirus pandemic, I wanted to know what the public's general sentiment and opinion is on each leader and how well the population thinks the leader is handling the crisis.

I selected 3 leaders of countries that seem to be handling the crisis well and 3 leaders from countries where the pandemic is still growing rapidly. I chose the following countries: United States, India, China, Germany, Japan, and Britain and searched for tweets on their leaders:

- Donald Trump
- Narendra Modi
- Xi Jinping
- Angela Merkel
- Shinzo Abe
- Boris Johnson

** Please note, this PowerPoint has only some of the R code and analysis from my project. Please refer to my R script: twitter_sent_analysis.R for my complete analysis. Thank you!

Account Setup & Retrieving Tweets

- Setup Developer Account with Twitter to access account tokens. Used my consumer keys to establish a connection to the Twitter API with the TwitteR package:

```
consumer_key <- '7TZej7KM8M6j7bFAUR4ad49Hu'
consumer_secret <- 'SDO9V566eFyXVR79OytPLBstS4f9rIrYCCoxeeCTs9aBdPoPXX'
AccessToken <- '1192863433575686144-KEru54FegpvekfV1hPFKt474gioWMh'
AccessTokenSecret <- 'ILnUus2XgYEGb3PC51hjZ7kM34odhmHe8PJD9WLF2S6Xj’

# Establishing a connection to the Twitter API
setup_twitter_oauth(consumer_key, consumer_secret, AccessToken, AccessTokenSecret)
```

- Retrieve tweets for 6 leaders:

```
# list of search terms
trump = 'Donald Trump' # American President
modi = 'Modi' # Indian Prime Minister
xi = 'Xi Jinping' # Chinese President
angela = 'Angela Merkel' # German Chancellor
shinzo = 'Shinzo Abe' # Japanese President
boris = 'Boris Johnson' #British Prime Minister

keyterms <- c(trump, modi, xi, angela, shinzo, boris)

# Fetching tweets for all keyterms- excluding retweets:
tweets_list <- list()
print(noquote('Fetching tweets. This may take a few minutes...'))
for (i in 1:6) {
  tweets_list[[i]] <- searchTwitter(paste(keyterms[i], '-filter:retweets'), n = 500,
lang = 'en')}
```

** Please note, this PowerPoint has only some of the R code and analysis from my project. Please refer to my R script: twitter_sent_analysis.R for my complete analysis. Thank you!

Cleaning & Preprocessing

- Sample Raw tweets on Donald Trump (as example):

```
# getting the tweet message from tweet for all keyterms
tweet_texts <- list()
for (i in 1:6) {tweet_texts[[i]] <- lapply(unlist(tweets_list[i]), function(t){t$getText()})}
```

Output:

```
[1] Sample tweets from " Donald Trump " search:
[1] "@realDonaldTrump i love vietnam i love donald trump.  If you can move American companies from China to Vietnam, 96… https://t.co/LZISbLpJ3P"
[2] "Donald Trump &amp;amp; CIA MUST Know US/Israeli Leaders/Intelligence Agencies \"PROUD OF TRIBAL MILITARY SUPERIORITY\", e.g.… https://t.co/jNNjo0bZZg"
[3] "Trump reportedly doesn't have time to get lunch. \n\nDishonest, deluded, dithering, Donald.\n\nYour kooky, personal, sp… https://t.co/vfdRjxSopn"
```

- Tweet Preprocessing: Removing URLs, handles, emoticons, hashtags, punctuation, numbers, and whitespaces.

```
## Tweet preprocessing
clean_up <- function(text) {
  clean <- gsub('http\\S+\\s*',"", text) # removing imbedded URLs
('https:..')
  clean <- gsub('@\\w+', "", clean) # removing twitter handles
  clean <- gsub("[^\x01-\x7F]", "", clean) # removing emoticons
  clean <- gsub('#[A-Za-z0-9]+', "", clean) # removing hashtags
  clean <- gsub('[[:punct:]]', " ", clean)
  clean <- gsub('[[:digit:]]', '', clean)
  clean <- gsub('\\d+', '', clean)
  clean <- gsub('\n', " ", clean)
  clean <- tolower(clean)
  clean <- str_squish(clean)
  return(clean)}
```

** Please note, this PowerPoint has only some of the R code and analysis from my project. Please refer to my R script: twitter_sent_analysis.R for my complete analysis. Thank you!

Part I: Text Analysis

- Sample Cleaned tweets on Donald Trump (as example):

```
Output:
[1] Sample clean tweets from " Donald Trump " search:
[1] "i love vietnam i love donald trump if you can move american companies from china to vietnam"
[2] "donald trump amp cia must know us israeli leaders intelligence agencies proud of tribal military superiority e g"
[3] "trump reportedly doesn t have time to get lunch dishonest deluded dithering donald your kooky personal sp"
```

- Text Analysis: Creating and preprocessing the text corpus, term document matrix, and retrieving top words for each leader:

```
# creating the text corpus for each leader
corpus_list <- list()
for (i in 1:6) {
  corpus_list[[i]] <- Corpus(VectorSource(unlist(clean_tweets_list[i])))}

# removing stop words and other insignificant words:
for (i in 1:6) {
  corpus.trans <- tm_map(corpus_list[[i]], removeWords, c(stopwords('english'), new_stopwords,
                  unlist(remove_words_list[i])))
  corpus_list[[i]] <- corpus.trans }
```

Note: For each leader, I removed specific words related to their country which do not provide us further insight on public sentiments. Eg. For Donald Trump, remove_words_list[[1]]<- list(c(tolower(trump), 'trump', 'donald', 'america', 'united', 'states', 'president', 'prime minister', 'trumps')

```
# finding frequent terms for each leader
for (i in 1:6) {print(noquote(paste('Frequent Terms in tweets with "', keyterms[i], ':')));
  print(findFreqTerms(tdm_list[[i]], lowfreq=15)); print(noquote(rep('_', 40)))}

# finding Associations to the term ‘economy’ for each leader
for (i in 1:6) {print(noquote(paste('Associations with "economy" in tweets with "', keyterms[i], ':')));
  print(findAssocs(tdm_list[[i]], 'economy', 0.4)); print(noquote(rep('_', 40))) }
```

** Please note, this PowerPoint has only some of the R code and analysis from my project. Please refer to my R script: twitter_sent_analysis.R for my complete analysis. Thank you!

Top *n* Most Frequent Words

- Sample output shown for Frequent Terms with Donald Trump (as example):

```
Output:
[1] Frequent Terms in tweets with " Donald Trump :
[1] "china"          "know"           "house"          "white"
[5] "coronavirus"    "disinfectant"   "responsibility" "kim"
```

- Sample output shown for Associations with 'economy' for Angela Merkel tweets (as example):

```
Output:
[1] Associations with "economy" in tweets with " Angela Merkel :
$economy
     faster     realize   crippling         guy     helping       irish  reward    varadkar     empathy
       0.41        0.41        0.41        0.41        0.41        0.41   0.41        0.41        0.41
    happens intelligent      saving     thereby         icu         pat     protect     rightly     strangle
       0.41        0.41        0.41        0.41        0.41        0.41        0.41        0.41        0.41
    fourth     largest
       0.41        0.41
```

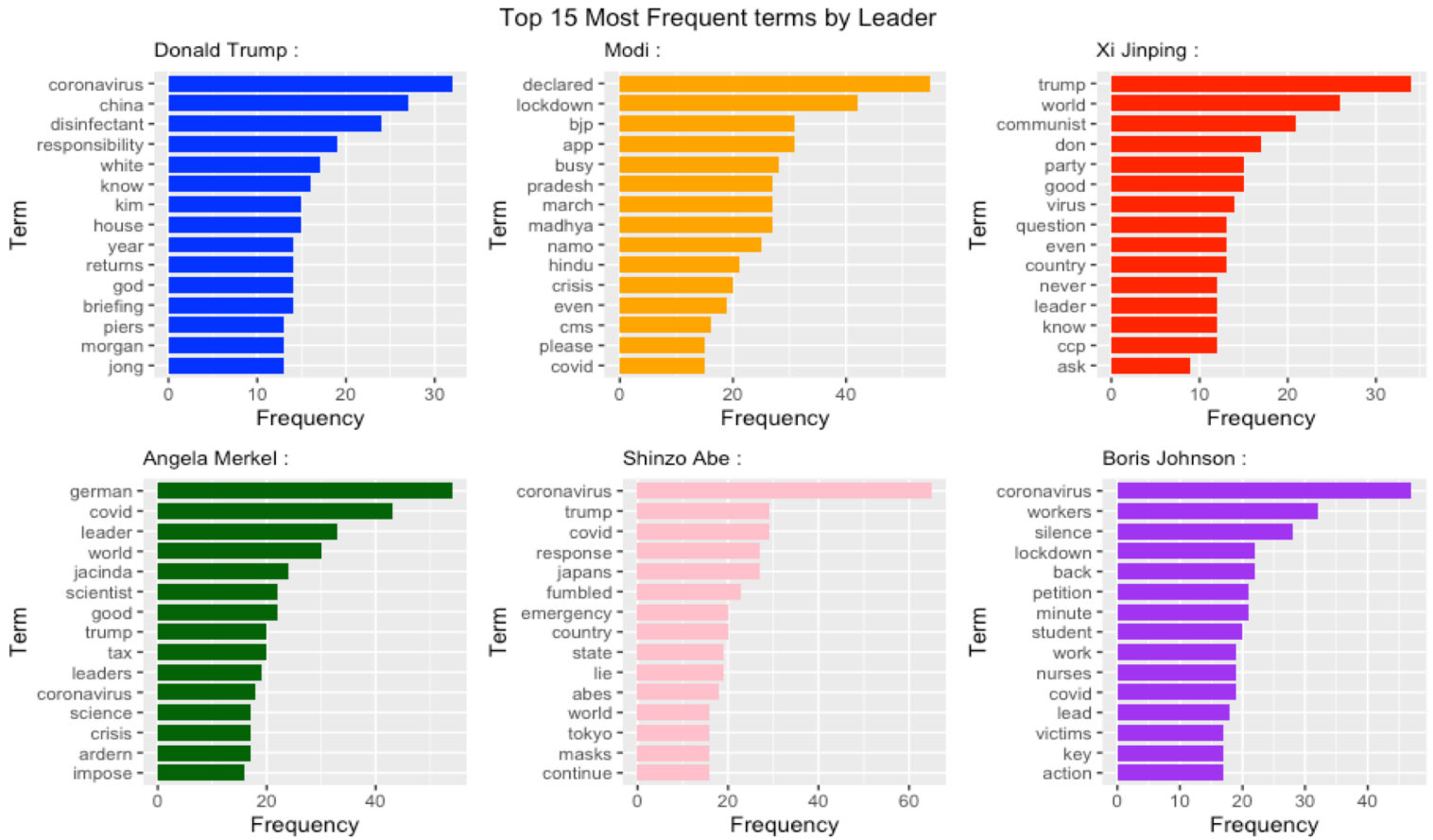
- Computed Top_n() most frequent terms for each leader

```
# Sorting the words by descending order of frequency
for (i in 1:6) {term_freq_list[[i]] <- sort(term_freq_list[[i]], decreasing=TRUE)}

# Function to return the top n most frequent words:
top_n <- function(i, n) {
  term_freq = unlist(term_freq_list[i])
  x <- data.frame(names(term_freq[1:n]), term_freq[1:n])
  rownames(x) <- NULL
  colnames(x) <- c('Term', 'Frequency')
  a <- noquote(paste('Top', n, 'terms occuring in tweets with', '"', keyterms[i], '"', ':'))
  print(a); print (x)}

# top 10 words:
for (i in 1:6) {top_n(i,10); print(noquote(rep('_', 25)))}
```

** Please note, this PowerPoint has only some of the R code and analysis from my project. Please refer to my R script: twitter_sent_analysis.R for my complete analysis. Thank you!

Data Visualization

```
# plotting the most frequent words by leader:
plot_list <- list()
colors <- c('blue', 'orange', 'red', '#046307', 'pink', 'purple')
for (i in 1:6) {
  plot_list[[i]] <- ggplot(top_n(i, 15), aes(x = reorder(Term, Frequency), y = Frequency)) +
    geom_bar(stat = 'identity', width = 0.8, fill = colors[i]) + coord_flip() +
    ggtitle(paste(keyterms[i], ':')) +
    xlab('Term') + theme(plot.title = element_text(size=10))}

grided <- gridExtra::grid.arrange(grobs = plot_list, top = 'Top 15 Most Frequent terms by
Leader' , ncol = 3, nrow = 2)
```

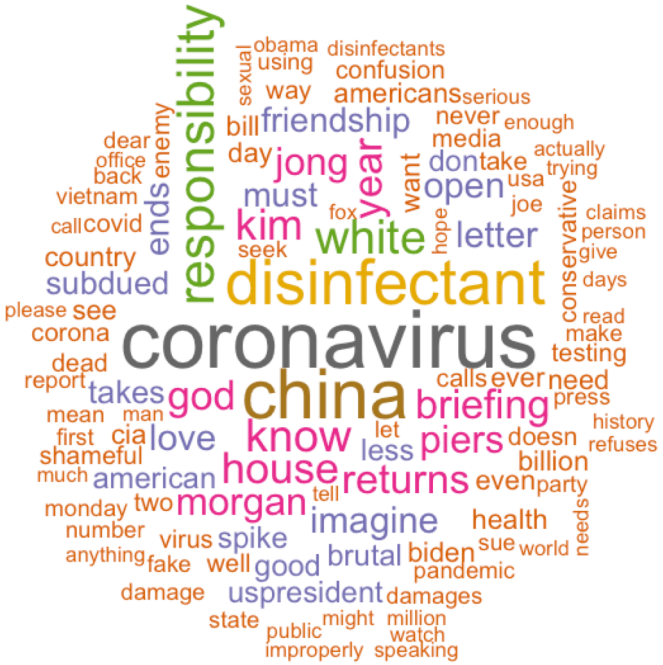


Data Visualization: WordClouds

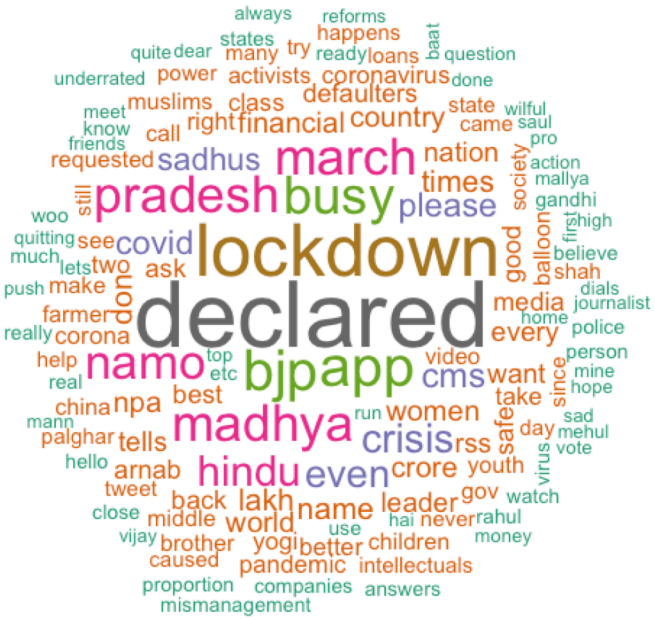
```
# Word Cloud Analysis
par(mfrow = c(1,1))
palette <- brewer.pal(8,"Dark2")

set.seed(137)
noquote(paste('WordCloud of Top terms in', trump, 'tweets --->'))
wordcloud(words=names(term_freq_list[[1]]), freq=term_freq_list[[1]],
          min.freq=5, random.order=F, colors=palette)
```

WordCloud by Leader:

Donald Trump



Modi

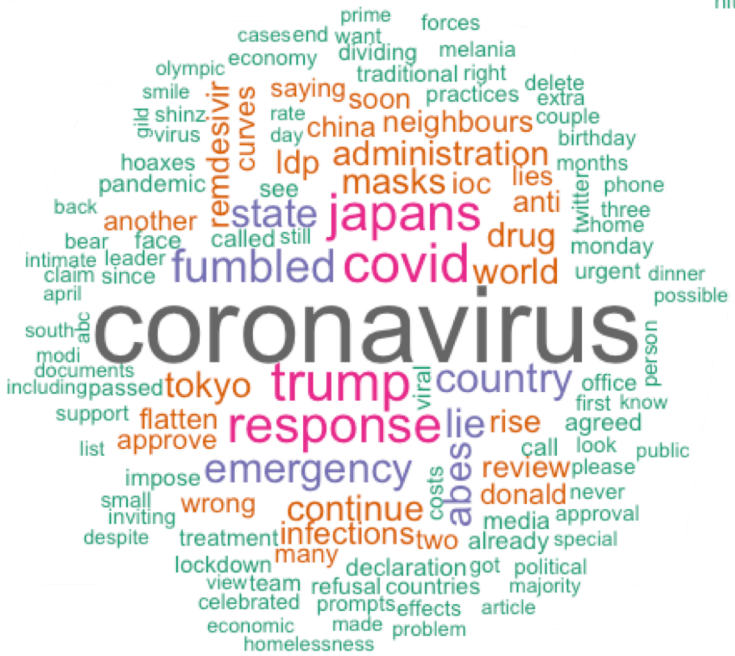


** Please note, this PowerPoint has only some of the R code and analysis from my project. Please refer to my R script: twitter_sent_analysis.R for my complete analysis. Thank you!

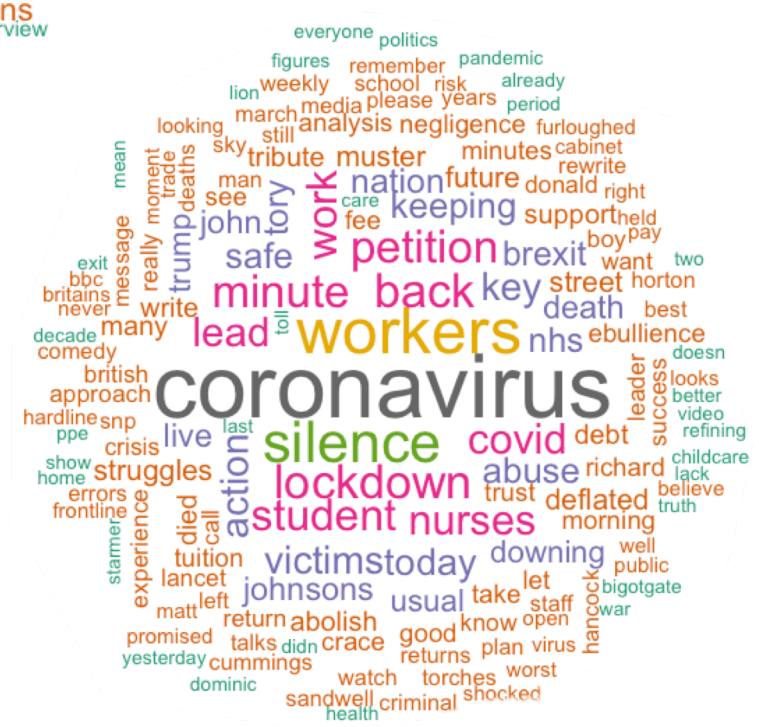
Xi Jingping

Angela Merkel

Shinzo Abe



Boris Johnson



Part II: Sentiment Analysis

- Determining Sentiment Score for each tweet for each Leader:

```
# Calculating the sentiment score for a given text
sentiment <- function(text, i) {
  # split the text into a vector of words
  words <- strsplit(text, '\\s+')
  words <- unlist(words)
  words <- words[!words %in% stopwords('en')]
  words <- words[!words %in% c(new_stopwords, remove_words_list[[i]])]
  # find which words are positive
  pos.matches <- match(words, pos.words)
  pos.matches <- !is.na(pos.matches)
  # find which words are negative
  neg.matches <- match(words, neg.words)
  neg.matches <- !is.na(neg.matches)
  # calculate the sentiment score
  p <- sum(pos.matches)
  n <- sum(neg.matches)
  if (p==0 & n==0)
    return(NA)
  else
    return(p-n)}

# calculating sentiment analysis scores for individual tweets
twitter_sentiments_list <- list()
for (i in 1:6) {
  sent_scores <- sapply(clean_tweets_list[[i]], sentiment, i = i)
  vector <- sapply(clean_tweets_list[[i]], function (t) {(t)})
  twitter_sentiments_list[[i]] <- data.frame(Score=sent_scores, Tweet= vector)}
```

Sample Sentiment Score for tweets on Donald Trump:

Output:

```
[1] Sample Sentiment Score for Tweets on " Donald Trump ":
  Score                                                                                  Tweet
1     2                       i love vietnam i love donald trump if you can move american companies from china to vietnam
2     3 donald trump amp cia must know us israeli leaders intelligence agencies proud of tribal military superiority e g
3    -3        trump reportedly doesn t have time to get lunch dishonest deluded dithering donald your kooky personal sp
4    -2      because americans sick and dying is all about donald trump the guardian trump returns to white house briefi
5    NA                      wet haunted doll cosmetic surgery conundrum is it ok to speculate about jared kushner and botox
```

Part II: Sentiment Analysis

- Analyzing the Distribution of Sentiment Scores for each Leader:

```
# distribution of sentiment scores
sentiment_dist <- function(i) {
  a <- noquote(paste('Distribution of Sentiment Scores for Tweets on', '"', keyterms[i],'":'))
  b <- twitter_sentiments_list[[i]]$Score
  print(a); print(table(b))}

for (i in 1:6) {sentiment_dist(i)}
```

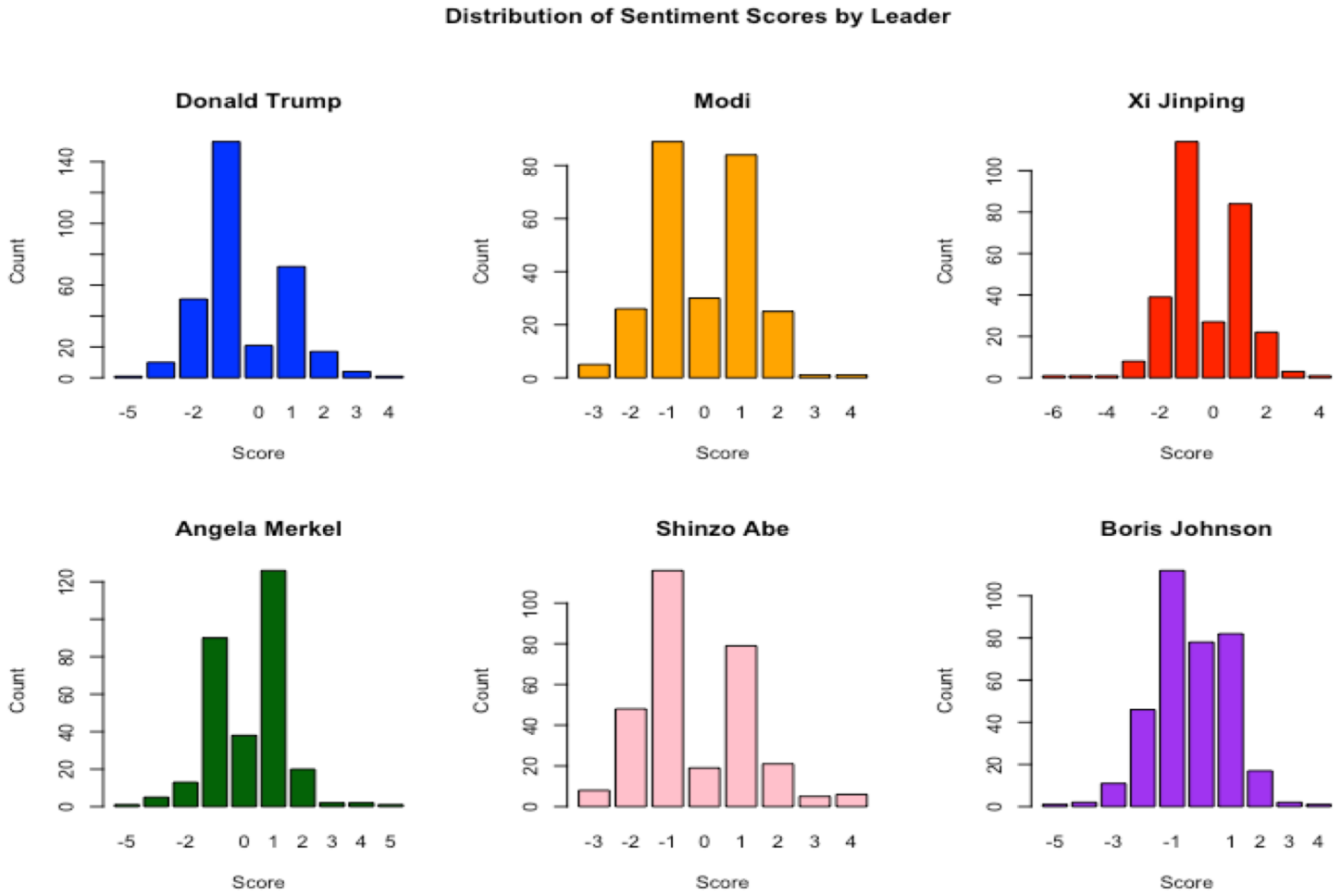
Output:

```
[1] Distribution of Sentiment Scores for Tweets on " Donald Trump ":
b
 -5  -3  -2  -1   0   1   2   3   4
  1  10  51 153  21  72  17   4   1
[1] Distribution of Sentiment Scores for Tweets on " Modi ":
b
-3 -2 -1  0  1  2  3  4
 5 26 89 30 84 25  1  1
[1] Distribution of Sentiment Scores for Tweets on " Xi Jinping ":
b
 -6  -5  -4  -3  -2  -1   0   1   2   3   4
  1   1   1   8  39 114  27  84  22   3   1
[1] Distribution of Sentiment Scores for Tweets on " Angela Merkel ":
b
 -5  -3  -2  -1   0   1   2   3   4   5
  1   5  13  90  38 126  20   2   2   1
[1] Distribution of Sentiment Scores for Tweets on " Shinzo Abe ":
b
 -3  -2  -1   0   1   2   3   4
  8  48 116  19  79  21   5   6
[1] Distribution of Sentiment Scores for Tweets on " Boris Johnson ":
b
 -5  -4  -3  -2  -1   0   1   2   3   4
  1   2  11  46 112  78  82  17   2   1
```

** Please note, this PowerPoint has only some of the R code and analysis from my project. Please refer to my R script: twitter_sent_analysis.R for my complete analysis. Thank you!

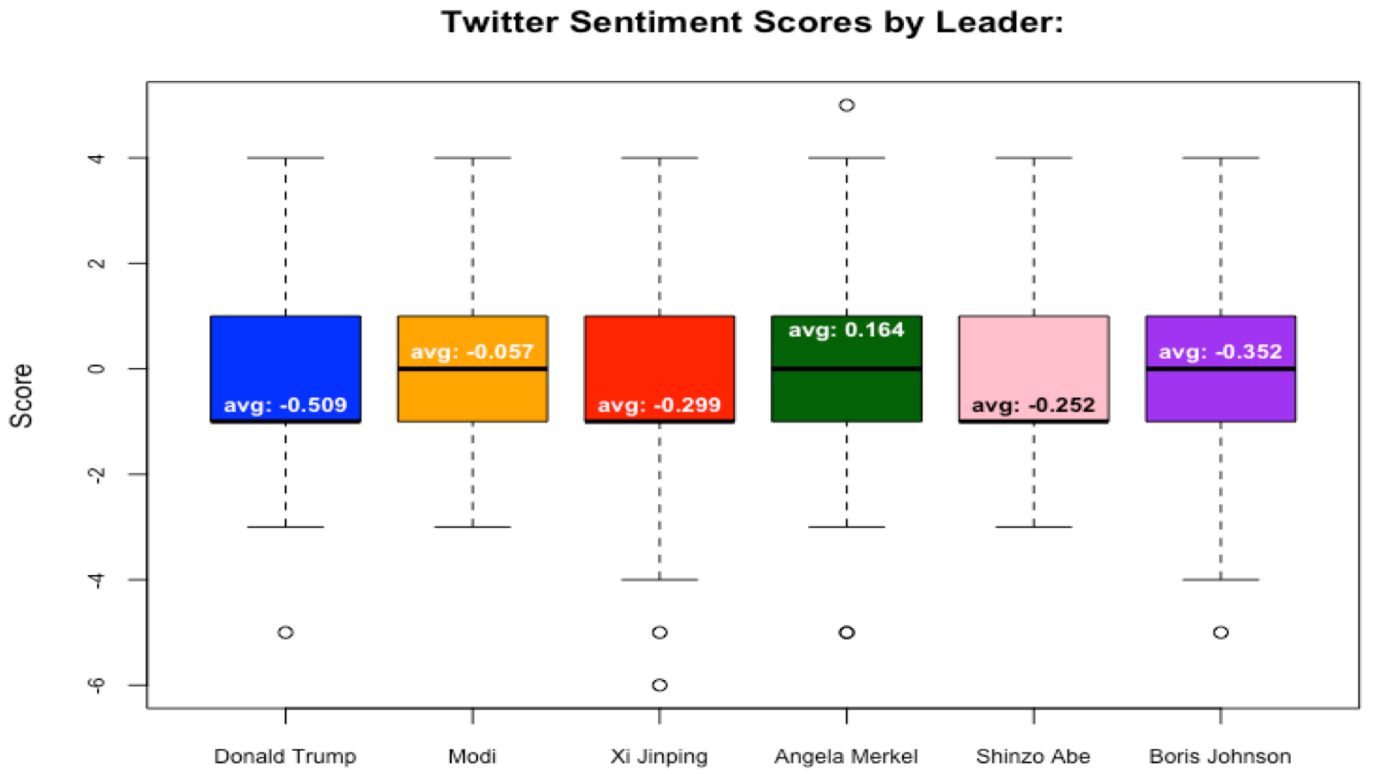
Data Visualizations: Sentiment Analysis

```
# barplots of sentiment distributions:
par(mfrow = c(2,3), oma = c(0, 0, 4, 0))

barplot_list <- list()
for (i in 1:6) {
  barplot_list[[i]] <- barplot(table(twitter_sentiments_list[[i]]$Score), xlab = 'Score',
                               ylab = 'Count', col = colors[i], main = keyterms[i])}
title(main=print(("Distribution of Sentiment Scores by Leader")),outer=T, cex = 1.5)
```



Data Visualizations: Sentiment Analysis

```
# boxplot of scores
a <- cbind(unlist(scores_list[1]), unlist(scores_list[2]),
           unlist(scores_list[3]), unlist(scores_list[4]),
           unlist(scores_list[5]), unlist(scores_list[6]))
colnames(a) <- c(trump, modi, xi, angela, shinzo, boris)

#Graphical depiction of State vs. Points received data
par(mfrow = c(1,1), oma = c(0, 0, 0, 0))

boxplot(a, main = 'Twitter Sentiment Scores by Leader:',
        col = colors, cex.axis = 0.75, ylab = 'Score')
```



** Please note, this PowerPoint has only some of the R code and analysis from my project. Please refer to my R script: twitter_sent_analysis.R for my complete analysis. Thank you!

Interpreting Sentiment Scores

Average Sentiment Score for each Leader:

```
# removing NA scores
scores_list <- list()
for (i in 1:6) {
  scores_list[[i]] <-
twitter_sentiments_list[[i]]$Score[!is.na(twitter_sentiments_list[[i]]$Score)]}

# Average sentiment score by leader
avg.score_list <- c()
for (i in 1:6) {avg.score_list[i] <- round((sum(scores_list[[i]])/length(scores_list[[i]])),3)}

for (i in 1:1){
a <- noquote('Average Sentiment Score by Leader (Sorted in decreasing order):')
b <- data.frame(Leader = keyterms, Average.Score = avg.score_list)
b <- b[order(b$Average.Score, decreasing = TRUE),]
print(a); print(b)}
```

```
Output:
[1] Average Sentiment Score by Leader:
         Leader Average.Score
4 Angela Merkel         0.164
2          Modi        -0.057
5    Shinzo Abe        -0.252
3    Xi Jinping        -0.299
6 Boris Johnson        -0.352
1  Donald Trump        -0.509
```

- Angela Merkel (Chancellor of Germany) has the highest Sentiment Score, indicating the public is speaking positive things about her and is possibly content with how she has handled the coronavirus pandemic.

- Donald Trump (President of the United Stated) has the lowest Sentiment Score, indicating the public is generally tweeting negative things about him and possibly unhappy with how he has handled the coronavirus pandemic.

Final Statements

While my results indicate peoples' general sentiment on the selected leaders, it is important to take note of the following when interpreting this twitter sentiment analysis:

- People typically only tweet when they feel strongly about a topic and more often than not, when they feel cheated or unhappy with a situation (as is similar to yelp reviews). The twitter sentiment scores we see may be skewed to reflect more negative sentiments than what the people may actually be feeling.

- By the nature of twitter, tweet language is very casual and often abbreviated to shorten characters. For example, a user may tweet 'hppy' instead of 'happy' – only the latter would be found in the lexicon of positive and negative words. Some sentiment based words may have been unaccounted for in the analysis.

- The twitter analyses we see now is only based on the tweets at this current time. Twitter sentiment scores on search terms frequently evolve as new events and information develops and people tweet on those new topics. For example, previously Donald Trump had one of the top sentiment scores, however based on the tweets used for this powerpoint, Donald Trump fared poorly since new events may have come up that changed the public's sentiments. Next time, it would be interesting to analyze how sentiments for a leader change over a span of time.

Overall, this project was a great exercise to understand the top words and topics associated with each leader and how people are generally perceiving each leader currently! I hope you enjoyed learning with me.

Thank You!