Part 2: Final Project

Introduction

For my final project, I did text/sentiment analysis on twitter comments including a specific keyword as "#keyword." I chose the presidential election as a theme and picked four keywords to carry out my text/sentiment analysis. The main goal of this project was to understand methodology of text/sentiment analysis, as well as the application of this method to real world case scenario, which was to analyze tone and emotion of twitter comments on the presidential election.

Sentiment analysis is the interpretation and classification of emotions within text data using text analysis techniques. It allows to identify the emotional tone in the text. In the real word application, businesses use sentiment analysis techniques to analyze not only customer data but also internal business data to make important decisions.

In this project, there are three major parts to accomplish the goal I stated earlier. These three major parts are: 1) text mining, 2) sentiment analysis 3) visualization. Text mining would allow me to extract text data from twitter and preliminary process of texts ready for sentiment analysis. Sentiment analysis would consist of two different methods, bing (for polarity, negative and positive, analysis) and nrc (for emotion analysis). Finally, visualization would allow me to put all data together as charts and word clouds for the data presentation.

Method and Analysis

Part 1: Text mining and preliminary process

I. Extracting text from twitter
Following website (https://hackernoon.com/text-processing-and-sentiment-analysis-of-twitter-data-22ff5e51e14c) was used as a reference for the text mining method. First step, I created an account for twitter developer and applied for the data usage. This initial step took three days for the final approval from Twitter. Then, I created an application for text mining, and this application assigned me the keys that required for extracting/downloading twitter comments.

There are two required R packages to gain connection and downloading tweets. "Twitter" provides an interface to the Twitter web API. "ROAuth" allows users to authenticate via OAuth to the server of their choice.

Following is the R codes to obtained 1000 (2000 for #election2020) tweets per each keyword (#bernie, #biden, #trump)

```
library(twitteR)
library(ROAuth)

# Loading credentials
consumer_key <- 'my customer key'
consumer_secret <- 'my secret key'
access_token <- 'my token'
access_secret <- 'my access key'

# Setting up to authenticate
setup_twitter_oauth(consumer_key, consumer_secret, access_token, access_secret)

# Extracting tweets
tweets_bernie <- searchTwitter("#bernie", n=1000,lang = "en")
tweets_biden <- searchTwitter("#biden", n=1000,lang = "en")
tweets_trump <- searchTwitter("#trump", n=1000,lang = "en")
tweets_election2020 <- searchTwitter("#election2020", n=2000,lang = "en")

# Striping retweets to remove redundant tweet texts
no_rt_bernie <- strip_retweets(tweets_bernie)
no_rt_biden <- strip_retweets(tweets_biden)
no_rt_trump <- strip_retweets(tweets_trump)
no_rt_election2020 <- strip_retweets(tweets_election2020)

# Converting extracted tweets without retweet to dataframe
bernie <- twListToDF(no_rt_bernie)
biden <-twListToDF(no_rt_biden)
trump <- twListToDF(no_rt_trump)
election2020 <- twListToDF(no_rt_election2020)
```

As an outcome, codes returned four datasets, each consists of 1000 tweets (or 2000 tweets): bernie, tweets, tweets and election2020. These four datasets were now ready for the preliminary text process.

II. Preliminary text process

In order to process data, I used gsub() function to remove all the unnecessary portion of the tweets, which include: link, username, emoji, number. Also, I decided to remove word "trump" to reduce confusion to future sentiment analysis since word "trump" is considered as positive sentiment in sentiment package I used for the analysis.

Following is the R code to carry out the text clean-up process. Because I used the same codes to process each keyword, I put keyword instead of the actual keyword here to present the R code once. However, R script file contains full version of codes; thus, please refer R script file for details.

```
# Three packages required for the text clean-up process
library(dplyr)
library(tidyr)
library(tidytext)

#remove unnecessary elements include: link, username, emoji, numbers
tweets.keyword = keyword %>% select(screenName, text)
tweets.keyword$clean_text <- gsub("http\\S+", " ", tweets.keyword$text)
tweets.keyword$clean_text <- gsub("@\\w+", " ", tweets.keyword$clean_text)
tweets.keyword$clean_text <- gsub("[^\x01-\x7F]", " ", tweets.keyword$clean_text)
tweets.keyword$clean_text <- gsub("[[:digit:]]", " ", tweets.keyword$clean_text)

# Removing "trump" since trump is considered as positive sentiment during text analysis
tweets.keyword$clean_text <- gsub("Trump", " ", tweets.keyword$clean_text)
tweets.keyword$clean_text <- gsub("trump", " ", tweets.keyword$clean_text)
tweets.keyword$clean_text <- gsub("TRUMP", " ", tweets.keyword$clean_text)
```

To make each tweet as analysis ready, only text portion of the dataset was selected and transformed each word from tweets as a token. Then, I removed stop words from tokenized words from the previous step. Following is the R codes.

```
#unnext_tokens() function to convert to lowercase, remove punctuation
tweets.keyword_stem <- tweets.keyword %>%
  select(clean_text) %>%
  unnest_tokens(word, clean_text)

#remove stop words
cleaned_tweets.keyword <- tweets.keyword_stem %>%
  anti_join(stop_words)
```

As an outcome, codes returned four datasets consist of a list of words. Because number of stripped retweets was different among keywords, each final dataset was returned with the different numbers of words (or observation in other term): bernie with 2149 words, biden with 2887 words, trump with 1963 words, and election2020 with 5110 words.

Part 2: Sentiment Analysis

I. Polarity analysis: negative or positive sentiment

A variety of methods and dictionaries exist for evaluating the emotion in text. R package, tidytext, provides access to several sentiment lexicon. For polarity analysis, I decided to use bing lexicon (from Bing Liu and collaborators, https://www.cs.uic.edu/~liub/FBS/sentiment-analysis.html). Bing lexicon categorizes words in a binary fashion into positive and negative categories. Following is the R codes.

```
#bing sentiment analysis
bing_keyword = cleaned_tweets.keyword %>%
 inner_join(get_sentiments("bing")) %>%
 count(word, sentiment, sort=TRUE) %>%
 ungroup()
bing_keyword
```

```
# A tibble: 358 x 3
   word      sentiment     n
   <chr>     <chr>     <int>
 1 breaking  negative     10
 2 win       positive     10
 3 wins      positive     10
 4 wrong     negative      9
 5 supported positive      7
```

Codes returned a tibble (right box above) containing three variables: word, sentiment (negative or positive) and n (number of words counted). For further analysis, I used the below function.

```
# function for calculating sentiment score for each tweet
# https://towardsdatascience.com/twitter-sentiment-analysis-and-visualization-using-r-22e1f70f6967
sentiment_bing = function(twt){
 twt_tbl = tibble(text=twt) %>%              #text cleaning
  mutate(
   stripped_text=gsub("http\\S+"," ",text),
   stripped_text=gsub("TRUMP", " ",stripped_text),
   stripped_text=gsub("Trump", " ",stripped_text),
   stripped_text=gsub("trump", " ",stripped_text)) %>%
  unnest_tokens(word,stripped_text) %>%
  anti_join(stop_words) %>%
  inner_join(get_sentiments("bing")) %>%
  count(word,sentiment, sort= TRUE) %>%
  ungroup() %>%
  mutate(
   score= case_when(               #create a column score
    sentiment == 'negative'~n*(-1),         #assigns -1 when negative word
    sentiment == 'positive'~n*1) #assigns 1 when positive word
 sent.score=case_when(
  nrow(twt_tbl)==0~0,    #if there are no words, score is 0
  nrow(twt_tbl)>0~sum(twt_tbl$score)) #otherwise, sum the positive and negatives
 zero.type=case_when(
  nrow(twt_tbl)==0~"Type1",      #no words at all, zero=no
  nrow(twt_tbl)>0~"Type2")               #zero means sum of words=0
 list(score= sent.score, type=zero.type, twt_tbl=twt_tbl)
}
```

Above function allows me to calculate the sentiment score for each tweet. To apply function, I used the codes below, and codes returned a list of each tweet with its sentiment score, zero type and a tibble. For the later visualization, the returned candidate tibbles were put together as tweets_sentiment.

```
#apply function: retuns a list of all the sentiment scores, types and tables of the tweets
bernie_sent = lapply(bernie$text, function(x){sentiment_bing(x)})
biden_sent = lapply(biden$text, function(x){sentiment_bing(x)})
trump_sent = lapply(trump$text, function(x){sentiment_bing(x)})
election_sent = lapply(election2020$text, function(x){sentiment_bing(x)})

#create a tibble specifying the #keywords, sentiment scores and types
tweets_sentiment = bind_rows(
 tibble(
 keyword='#bernie',
 score=unlist(map(bernie_sent, 'score')),
 type=unlist(map(bernie_sent, 'type'))
),
tibble(
 keyword='#biden',
 score=unlist(map(biden_sent, 'score')),
 type=unlist(map(biden_sent, 'type'))
),
tibble(
 keyword='#trump',
 score=unlist(map(trump_sent, 'score')),
 type=unlist(map(trump_sent, 'type'))
)
)
election_sentiment= tibble(
 keyword='#election2020',
 score=unlist(map(election_sent, 'score')),
 type=unlist(map(election_sent, 'type'))
)
```

```
OUTCOME:
> tweets_sentiment
# A tibble: 1,020 x 3
   keyword score type 
   <chr>   <dbl> <chr>
 1 #bernie     0 Type1
 2 #bernie     1 Type2
 3 #bernie    -1 Type2
 4 #bernie     0 Type1
 5 #bernie     0 Type1
 6 #bernie    -1 Type2
 7 #bernie     0 Type1
 8 #bernie     0 Type1
 9 #bernie     0 Type1
10 #bernie     1 Type2
# ... with 1,010 more rows
```

II. Emotion analysis: 8 different emotions

For emotion analysis, I used nrc lexicon (from Saif Mohammad and Peter Turney, http://saifmohammad.com/WebPages/NRC-Emotion-Lexicon.htm). The nrc lexicon categorizes words in a binary fashion (yes or no) into categories of positive, negative, anger, anticipation, disgust, fear, joy, sadness, surprise and trust. From ten categories, I only used eight emotion sentiments without positive and negative sentiment.

Following R codes returned a tibble (right box above) containing three variables: word, sentiment (one of eight emotions) and n (number of words counted).

```
# NRC emotion sentiment analysis
nrc_keyword = cleaned_tweets.keyword %>%
  inner_join(get_sentiments("nrc")) %>%
  filter(!sentiment %in% c("positive","negative")) %>%
  count(word, sentiment, sort=TRUE) %>%
  ungroup()
nrc_keyword
```

```
# A tibble: 399 x 3
   word  sentiment        n
   <chr> <chr>        <int>
 1 vote  anger           20
 2 vote  anticipation    20
 3 vote  joy             20
 4 vote  sadness         20
 5 vote  surprise        20
 6 vote  trust           20
 7 time  anticipation    11
 8 trade trust            7
```

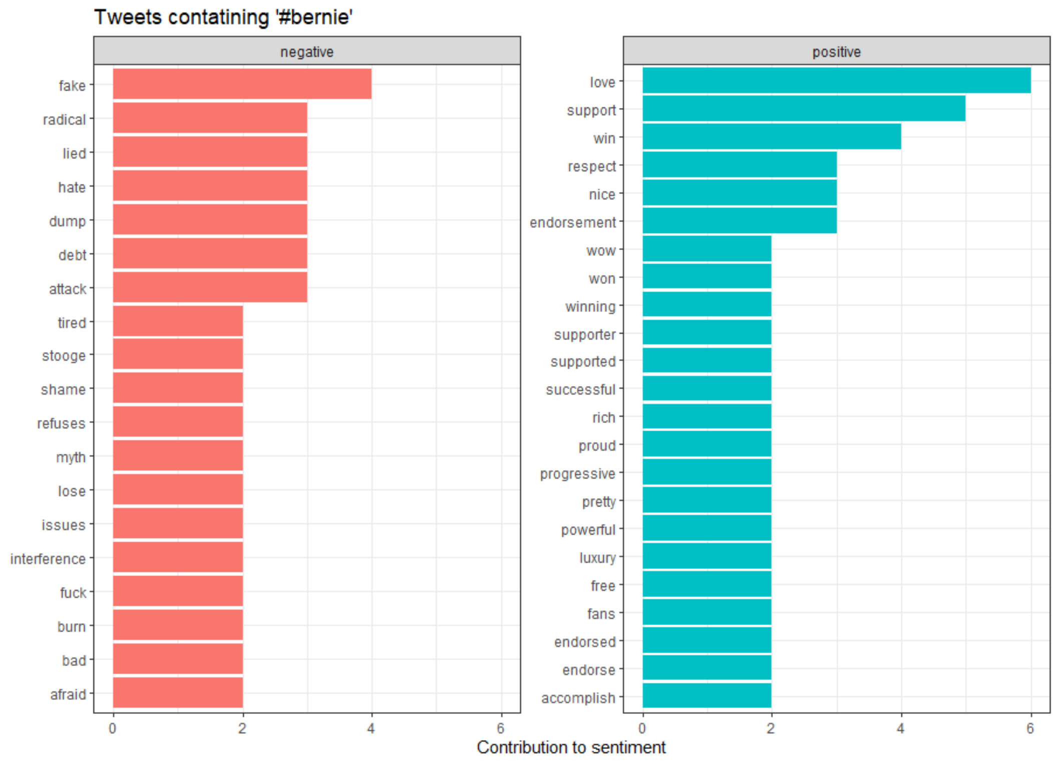
Visualization and Discussion

I. Top 10 commonly used words in the set of tweets for each keyword

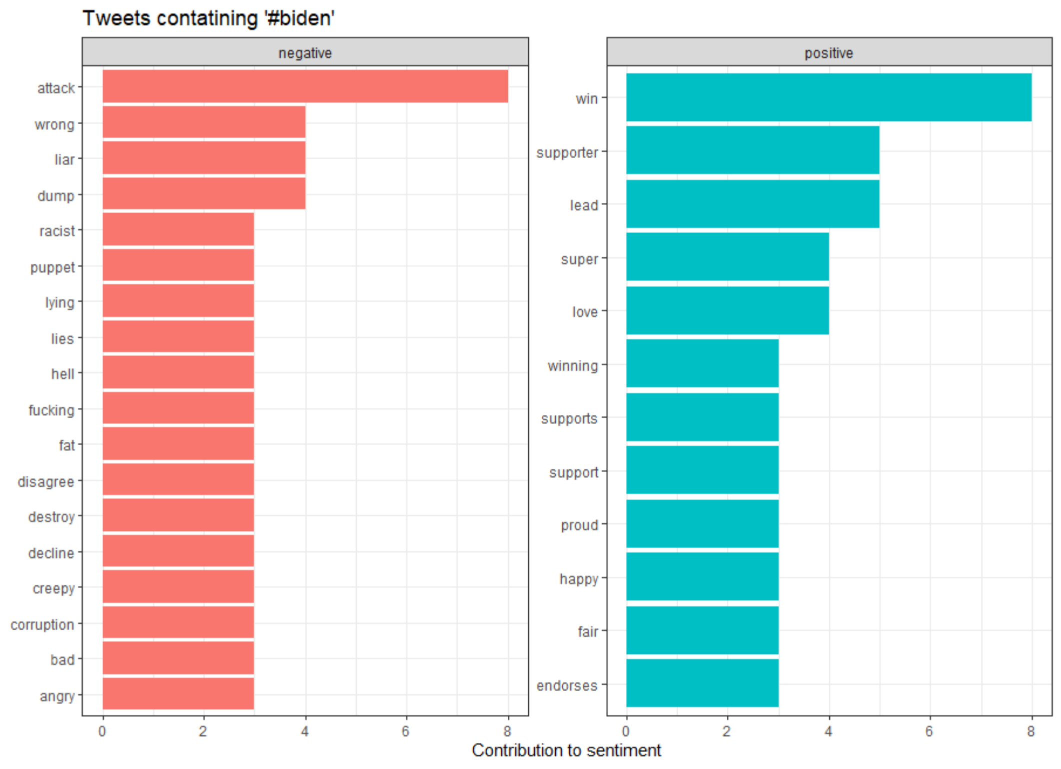
```
library(ggplot2)
bing_keyword %>%
  group_by(sentiment) %>%
  top_n(10) %>%
  ungroup() %>%
  mutate(word=reorder(word,n)) %>%
  ggplot(aes(word,n,fill=sentiment))+
  geom_col(show.legend=FALSE)+
  facet_wrap(~sentiment,scale="free_y")+
  labs(title="Tweets contatining '#trump'", y="Contribution to sentiment", x=NULL)+
  coord_flip()+theme_bw()
```

Above R codes returned bar graphs showing top 10 commonly used words in the set of tweets for each keyword. These commonly used words provide idea that what positive words and negative words are associated to each presidential candidate and presidential election. From these association, overall image (or portrait) of each candidate could be found.

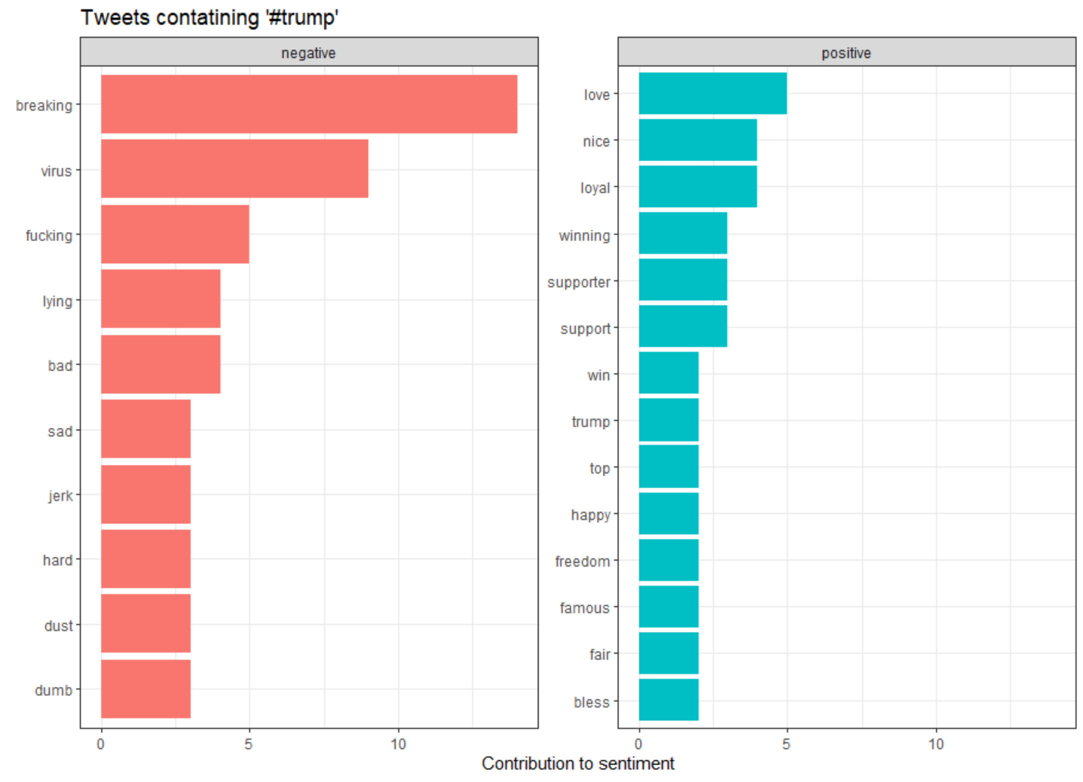
a. #bernie



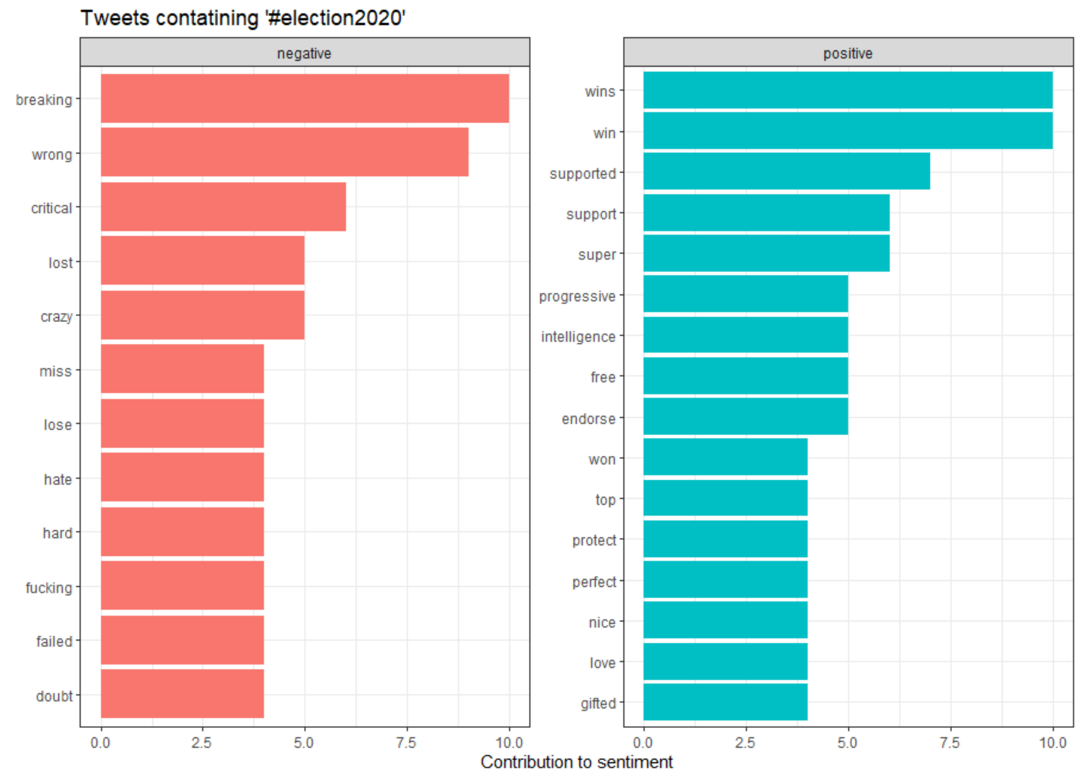
b. #biden



c. #trump



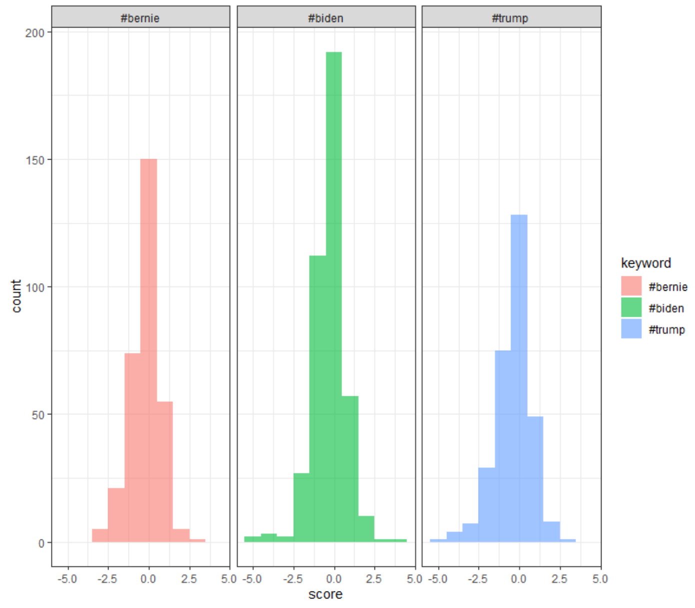
d. #election2020

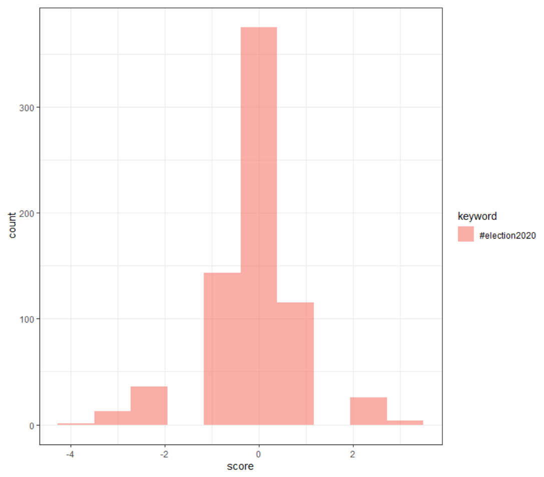


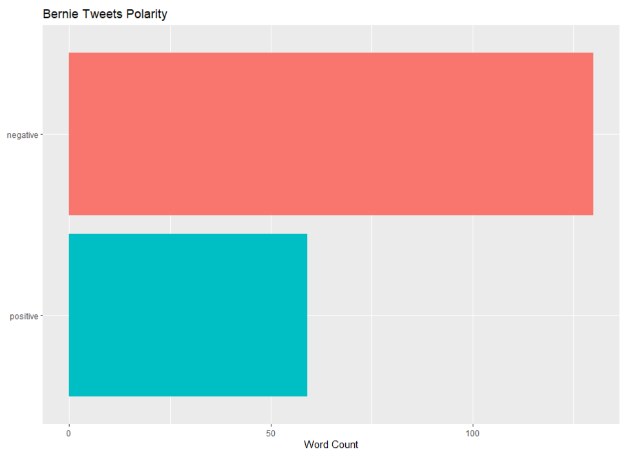
II. Polarity histogram and Negative vs Positive charts

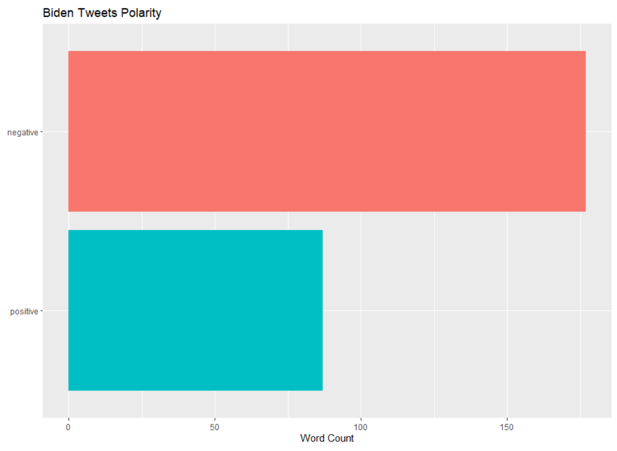
```
#plot histograms of tweets sentiment for three candidates
ggplot(tweets_sentiment,aes(x=score, fill=keyword)) +
  geom_histogram(bins=10, alpha=0.6) +
  facet_grid(~keyword) + theme_bw()

#plot histogram of tweets sentiment for election 2020
ggplot(election_sentiment,aes(x=score, fill=keyword)) +
  geom_histogram(bins=10, alpha=0.6) +
  theme_bw()

#Negative vs Positive
polar_bar_keyword <- bing_keyword %>%
  group_by(sentiment) %>%
  summarise(word_count = n()) %>%
  ungroup() %>%
  mutate(sentiment = reorder(sentiment, word_count)) %>%
  #Use `fill = -word_count` to make the larger bars darker
  ggplot(aes(sentiment, word_count, fill = c("blue","red"))) +
  geom_col() +
  guides(fill = FALSE) +
  labs(x = NULL, y = "Word Count") +
  ggtitle("Keyword Tweets Polarity") +
  coord_flip()
```

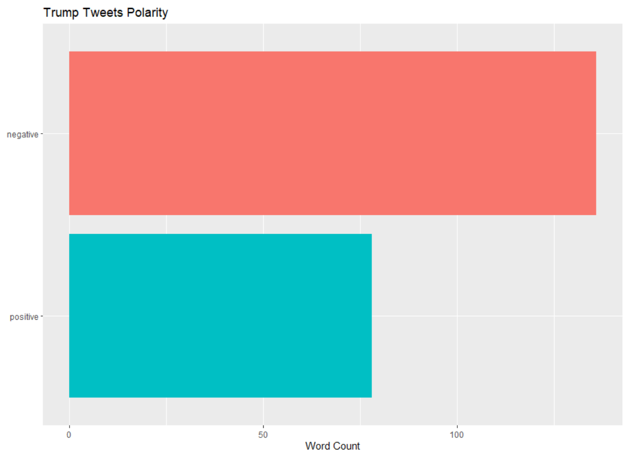
First two code groups returned histograms, and last code group returned negative vs positive bar graph.

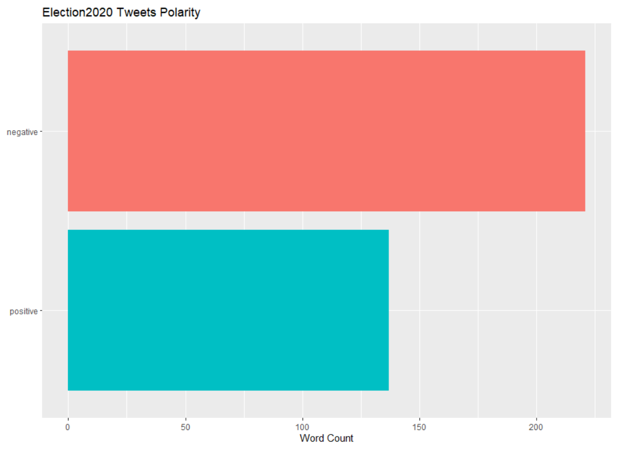










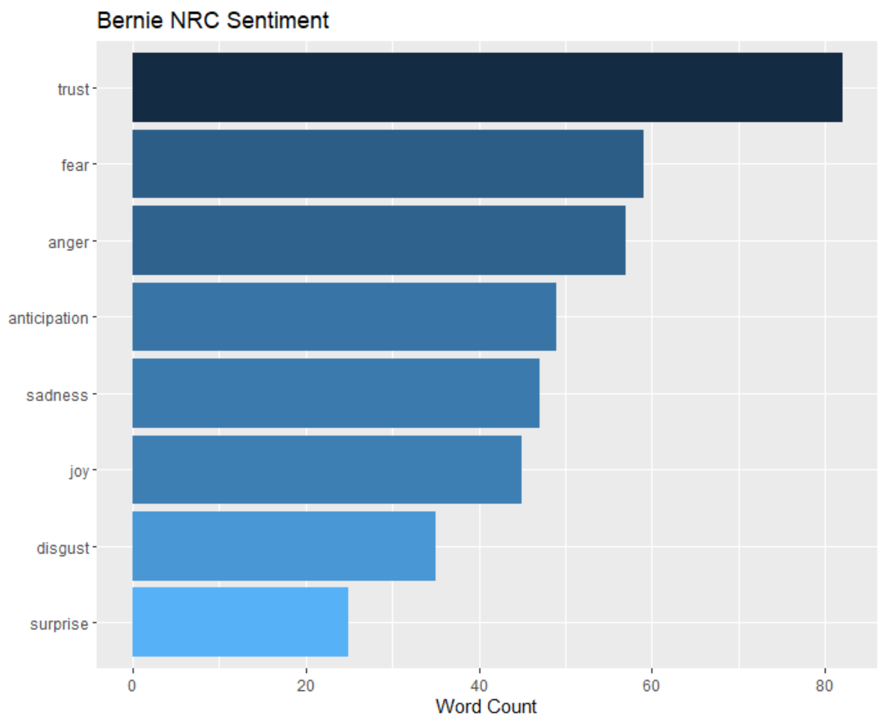


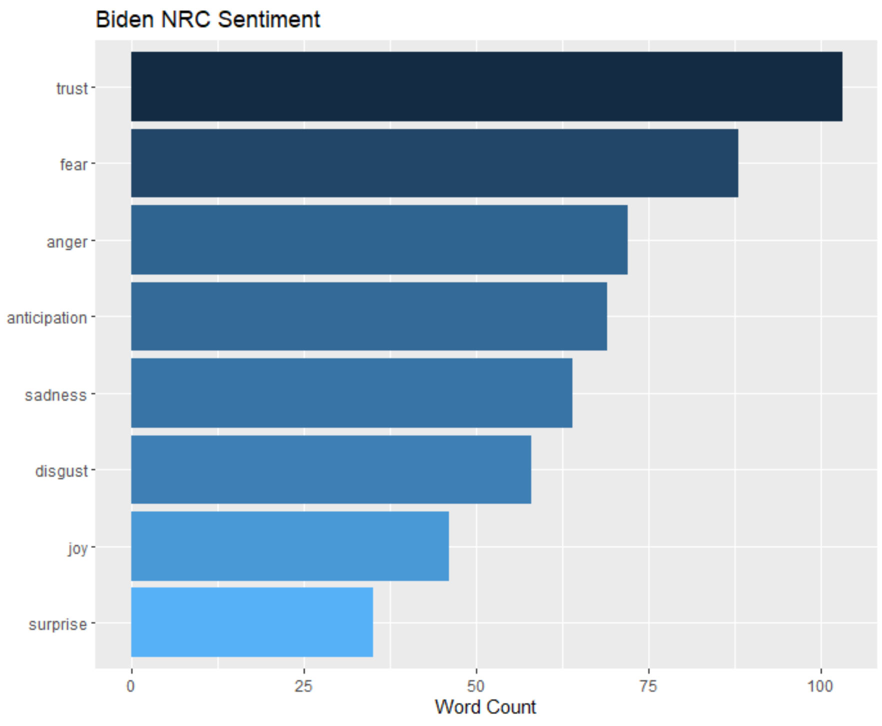
According to histograms and negative vs positive charts, tweets for all presidential candidates and presidential election tend to be negative. Histograms have the distribution with a right skewed. It implies that tweets tend to contain more negative sentiment. Negative vs positive charts clearly show that there is more negative sentiment in all candidates and election. It tells that people have more negative opinions toward the candidates and upcoming election. Based on these results, it is difficult to tell which candidate is most likely win for the upcoming presidential election.

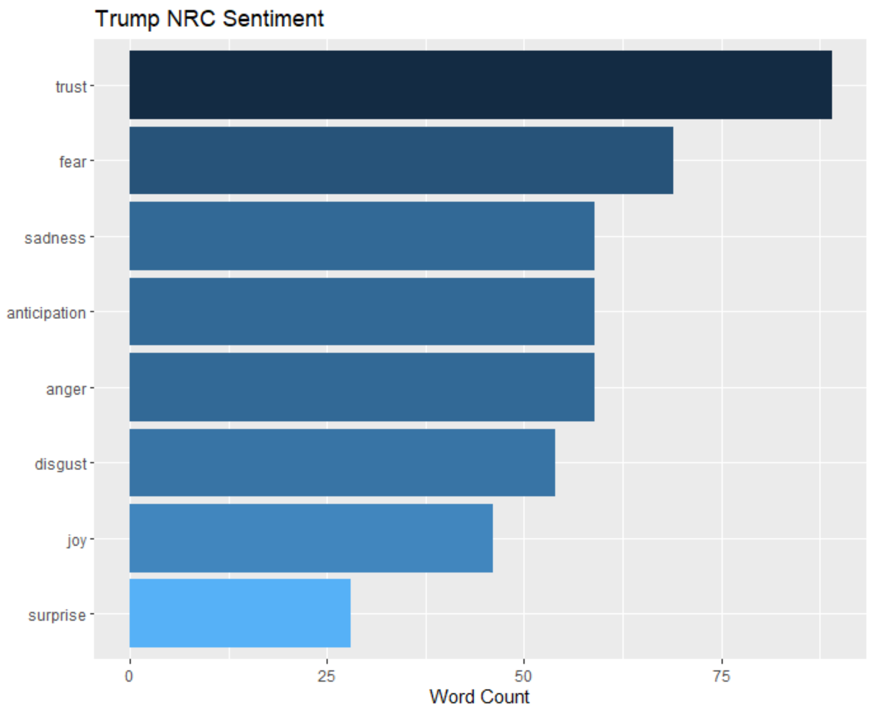
III. Emotions plot

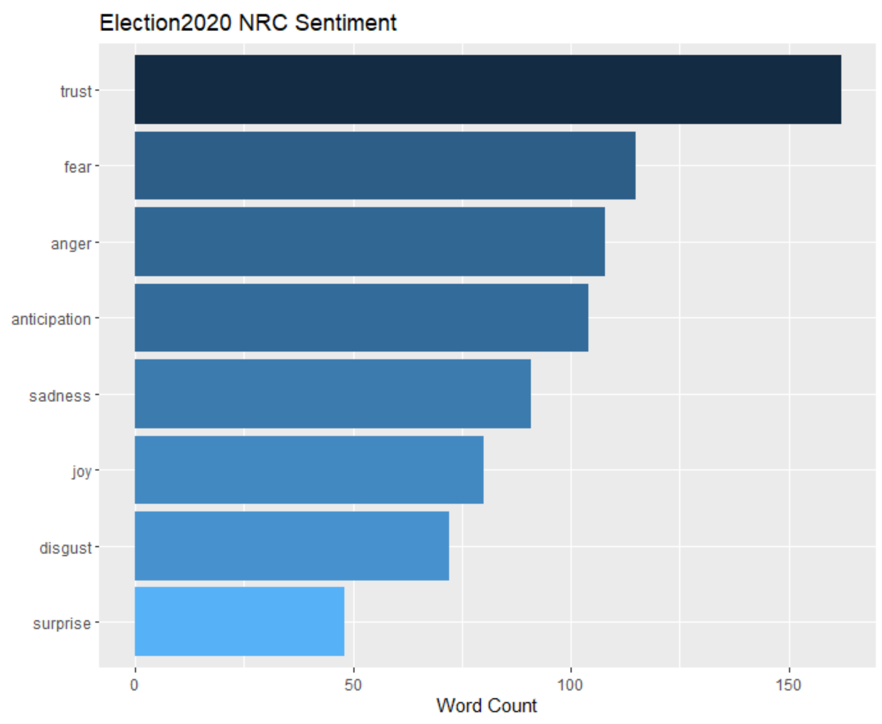
```
# Emotion plot
keyword_plot <- nrc_keyword %>%
  group_by(sentiment) %>%
  summarise(word_count = n()) %>%
  ungroup() %>%
  mutate(sentiment = reorder(sentiment, word_count)) %>%
  ggplot(aes(sentiment, word_count, fill = -word_count)) +
  geom_col() +
  guides(fill = FALSE) +
  labs(x = NULL, y = "Word Count") +
  ggtitle("Bernie NRC Sentiment") +
  coord_flip()
```

Above codes returned a graph of nrc sentiment analysis from the previous section of this paper. This graph shows which counts for each emotional category, descending order.




Biden NRC Sentiment
trust
fear
anger
anticipation
sadness
disgust
joy
surprise
0
25
50
75
100
Word Count


Trump NRC Sentiment
trust
fear
sadness
anticipation
anger
disgust
joy
surprise
0
25
50
75
Word Count

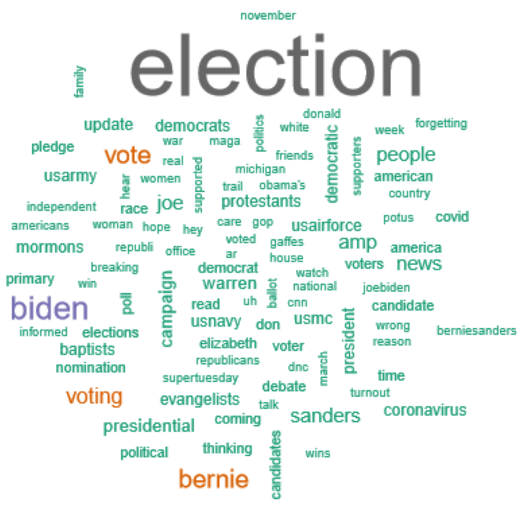


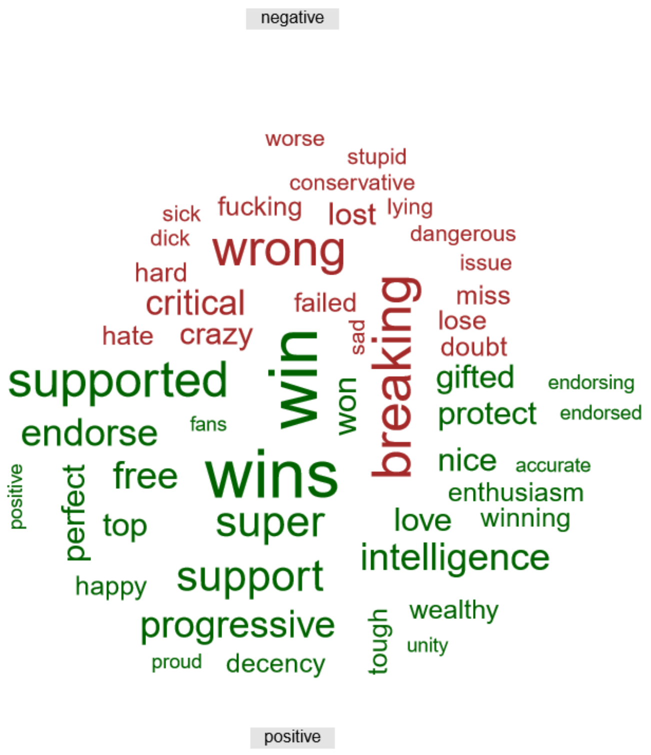
Trust, fear and anger were most prevalent emotions observed from all four keywords. It is difficult to interpret how exactly each emotion describe people's overall opinion on each keyword. However, trust, fear and anger would be the key emotions when people decide which candidate they would like to support and vote for.

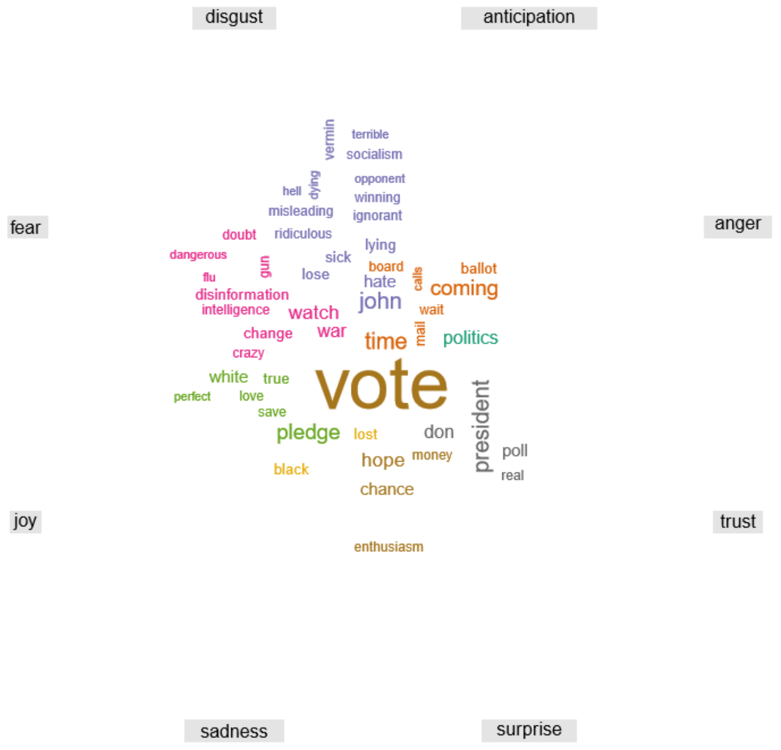
IV. Bonus: word clouds

I used following R packages and codes to generate three different word clouds: common word cloud, polarity word cloud and nrc emotion word cloud.

```
library(wordcloud)
library(RColorBrewer)

#common wordcloud
cleaned_tweets.election %>%
  anti_join(stop_words) %>%
  count(word) %>%
  with(wordcloud(word, n, max.words = 100, colors=brewer.pal(8, "Dark2")))

#polarity wordcloud
cleaned_tweets.election %>%
  inner_join(get_sentiments("bing")) %>%
  count(word, sentiment, sort = TRUE) %>%
  acast(word ~ sentiment, value.var = "n", fill = 0) %>%
  comparison.cloud(colors = c("brown", "dark green"),
                   title.size=1, max.words = 50)

#nrc emotion wordcloud
cleaned_tweets.election %>%
  inner_join(get_sentiments("nrc")) %>%
  filter(!sentiment %in% c("positive","negative")) %>%
  count(word, sentiment, sort = TRUE) %>%
  acast(word ~ sentiment, value.var = "n", fill = 0) %>%
  comparison.cloud(colors = brewer.pal(8, "Dark2"),
                   title.size=1, max.words=50, random.order = FALSE)
```


november
election
family
update
democrats
politics
donald
white
week
forgetting
pledge
vote
war
maga
democratic
supporters
people
usarmy
real
women
friends
michigan
american
country
independent
hear
race
joe
supported
trail
obama's
protestants
potus
covid
americans
woman
hope
hey
care
gop
usairforce
mormons
republi
office
voted
ar
gaffes
house
amp
america
primary
breaking
win
democrat
warren
watch
national
voters
news
joebiden
biden
poll
campaign
read
uh
ballot
cnn
candidate
informed
elections
usnavy
don
usmc
president
wrong
reason
berniesanders
baptists
nomination
elizabeth
republicans
voter
dnc
march
time
voting
supertuesday
evangelists
debate
turnout
talk
sanders
coronavirus
presidential
coming
political
thinking
candidates
wins
bernie


negative
worse
stupid
conservative
sick
fucking
lost
lying
dick
wrong
dangerous
hard
issue
critical
failed
miss
hate
crazy
sad
breaking
lose
doubt
supported
win
won
gifted
endorsing
endorse
fans
protect
endorsed
positive
perfect
free
wins
nice
accurate
enthusiasm
top
super
love
winning
intelligence
happy
support
wealthy
progressive
tough
unity
proud
decency
positive


disgust
anticipation
vermin
terrible
socialism
hell
dying
opponent
winning
misleading
ignorant
fear
anger
doubt
ridiculous
lying
dangerous
sick
gun
board
ballot
flu
lose
hate
calls
coming
disinformation
intelligence
watch
john
wait
change
war
time
mail
politics
crazy
white
true
vote
perfect
love
save
president
pledge
lost
don
poll
hope
money
real
black
chance
joy
trust
enthusiasm
sadness
surprise

Conclusion and Limitation

By completing this project, I was able to understand the concept and methodology of sentiment analysis. Also, I learned how to choose and apply the appropriate R packages for the analysis. In this project, I explored the available text mining and sentiment tools and had chance to use them in real world case scenario. However, there are some limitation I found. First, tweets that I collected was from recent date, so it is possible that tweets were biased toward one specific event or topic. Second, because words can be interpreted different depending on their context, sentiment tools may not fully catch the real sentiment of the words. Last, because I still was on the learning process of the R program and packages, I may not fully utilize the useful functions of the R and packages.

Reference

1. https://hackernoon.com/text-processing-and-sentiment-analysis-of-twitter-data-22ff5e51e14c

2. https://tabvizexplorer.com/sentiment-analysis-using-r-and-twitter/

3. https://rpubs.com/SulmanKhan/437587

4. https://www.r-bloggers.com/thrice-sentiment-analysis-emotions-in-lyrics/

5. https://www.tidytextmining.com/sentiment.html

6. https://cran.r-project.org/web/packages/syuzhet/vignettes/syuzhet-vignette.html

7. https://www.datacamp.com/community/tutorials/sentiment-analysis-R

8. https://towardsdatascience.com/twitter-sentiment-analysis-and-visualization-using-r-22e1f70f6967

9. https://dataaspirant.com/2018/03/22/twitter-sentiment-analysis-using-r/

10. https://cran.r-project.org/web/packages/saotd/vignettes/saotd.html

11. https://github.com/Twitter-Sentiment-Analysis/R

12. https://www.r-bloggers.com/twitter-data-analysis-in-r/

13. https://www.r-bloggers.com/twitter-sentiment-analysis-with-r/

14. https://scholar.afit.edu/cgi/viewcontent.cgi?article=2853&context=etd (Scholarly source)