DATA 630 Group Project: Trump Tweets

Group Study Report

Written by Group 2

Ed Perry, Daanish Ahmed, and Abdourahmane Bah

University of Maryland University College

DATA 630 9041 Data Mining (Summer 2017)

Dr. Edward Herranz

Introduction

Social media is a domain that contains vast amounts of untapped knowledge, especially when it comes to politics. Twitter tweets often provide valuable pieces of information; but since the data is in an unstructured format, it can be difficult and time-consuming to extract this knowledge. Fortunately, the field of text mining is making this process faster and easier—giving us greater understanding of the context of each text. One effective method for mining social media information is to use keyword-based association analysis. This method involves finding terms that frequently appear together and determining the relationships between those terms (Han, Kamber, & Pei, 2011). Our goal is to use keyword-based association analysis and data visualization methods to study a politician's successful use of social media. We will analyze the tweets sent by President (then candidate) Donald J. Trump during his 2016 presidential campaign ("The 11 Best Tweets," n.d.).

In this project, we will implement word clouds and bar graphs to study the most frequent words in Trump's tweets and how those words reflect his campaign. We will also use keyword-based association to understand Trump's attitudes towards political opponents such as Hillary Clinton and Ted Cruz, in addition to studying his views on issues such as jobs and the coal industry. We believe that by mining information from the President's Twitter posts, we may gain some insights regarding the sentiment of his campaign and the issues that many American voters find important. By doing this, we may even be able to use similar methods to predict the success of future political campaigns.

Data Exploration

The scope of this project is quite large. We will be analyzing all of the President's Twitter messages from July 16, 2015 to November 11, 2016—three days after Trump's election victory. The dataset itself consists of 7,375 tweets and 12 different variables (see Figure 1 in Appendix B). These variables include the date, time, text of the tweet, type of tweet (text, link, or video), media type (such as photo), hashtags, tweet ID, tweet URL, number of likes, and number of retweets. Our primary focus is on the tweet text variable, from which we will extract the words to use in our text mining analysis. According to Figure 1, the tweet ID is a numeric variable while the numbers of likes and retweets are integers. The variables "X" and "X1" are numeric variables, but most of their values are missing. These two variables did not exist in the original dataset and only appeared after importing the dataset into R Studio. As a result, these variables will be removed during preprocessing. The remaining variables are all factors.

For further exploration, we examined the descriptive statistics of the dataset (see Figure 2 in Appendix B). From this image, we see that the median number of likes is 5606 and the median number of retweets is 2173. We also see that the most common hashtags are #Trump2016 and #MakeAmericaGreatAgain, with 219 and 190 occurrences respectively. However, most tweets do not contain a hashtag. By examining the first 10 entries in the dataset, we notice that there are many blank values in the media type and hashtags columns (see Figure 3 in Appendix B). However, there are no missing values in any of the remaining variables (see Figure 4 in Appendix B), which means that those blank values are merely empty strings rather than NA's. The dataset will now require extensive preprocessing to prepare the data for implementation with our algorithms.

Preprocessing

There are several important preprocessing steps that we performed on our dataset. First, we installed and activated several packages such as "tm," "SnowballC," and "wordcloud." These packages are needed to perform text mining procedures such as association mining and generating word clouds. The next step is to remove unnecessary variables and convert variables into the desired format. The "time" variable is not needed for our analysis, hence we removed it. The variables "X" and "X.1" are useless as well, so we also removed them. The "date" variable was initially a factor type rather than a date type. We therefore converted this variable into a date type. And since there are no missing values in the dataset (see Figure 4 in Appendix B), we do not need to worry about handling them. After this, we converted the variable containing the tweet text into a corpus, which makes it easier to operate on textual data. From here, we can examine the first tweet in the corpus (see Figure 5 in Appendix B). It is apparent from the output that there are several issues which we must address. To handle these issues, we will use the "tm_map" method within the "tm" package to transform the text in our corpus. This method takes a corpus and applies a transformation function to the text, such as removing numbers or punctuation (Feinerer & Hornik, 2017).

First, we need to set all characters to lowercase. This is done to ensure that identical words are recognized as the same word even if their letters have different cases. Next, we need to remove numbers, punctuation, and whitespace from each tweet. These characters are unnecessary and can impact the results of our analysis if they are not removed. The last step is to remove stop words such as "and," "from," and "to" from the corpus. These words have no contextual meaning, and their removal will prevent our model from generating irrelevant or uninteresting results. The “tm” package contains a built-in list of stop words, which we used to remove all stop words from our corpus (Feinerer & Hornik, 2017). Additionally, we removed a list of unnecessary words not included in the stop words list, such as “just,” “good,” and “watch.” After removing these words, we once again examined the first tweet (see Figure 6 in Appendix B). Based on this image, we see that the remaining words have been cleaned and the data is ready for analysis.

Word Cloud

The first part of our analysis will consist of a word cloud that shows the most frequent words in Trump’s tweets. To build this model, we must first create a document term matrix (DTM) using the data in our corpus. A DTM uses the documents (or tweets) as the rows, the individual terms as the columns, and the frequency of each term as the matrix’s entries (“Basic Text Mining,” n.d.). Once the DTM has been created, we can examine its output (see Figure 7 in Appendix B). This figure reveals that the matrix has a sparsity of 100%, meaning that there are too many words with low frequencies. We will use the “removeSparseTerms” function on our DTM to address this issue. This function will remove infrequent terms—lowering the maximum sparsity to a percentage input by the user (Murphy, 2017). By using a sparsity threshold of 0.99, we find that our DTM now has a lower average sparsity of 98% (see Figure 8 in Appendix B)—meaning that the remaining words will be more relevant to our analysis.

Our next step is to find the most frequent terms in our DTM. This is done by sorting all words in descending order by their count, which results in the most common words appearing at the top of the list. These words and their counts are then stored in a variable called “freq.” We can now create a word cloud by calling the “wordcloud” method on our list. We used the names of each word and their corresponding frequencies as input values, set the maximum number of words equal to 30, set the words to appear in random order, and designated a color scheme that assigns colors according to a word’s frequency. This word cloud suggests that the most common words are “Trump” and “realdonaldtrump,” followed by the words “great” and “thank” (see Figure 9 in Appendix B). The meaning of these results will be explored in a later section of this paper. However, the issue with this word cloud is that there are numerous irrelevant words such as “many,” “night,” “last,” and “never.” We can enhance these results by including only the most relevant terms.

To improve our word cloud, we replace the term frequencies in our matrix with the term frequency-inverse document frequency (TF-IDF). The TF-IDF determines the relative importance of a certain word in a document, making it useful for removing less-frequent words (“Basic Text Mining,” n.d.). We will still create a document term matrix using our Trump corpus, but this time we will add the TF-IDF as a weight parameter. We will again remove sparse terms from our matrix, but using a sparsity threshold of 0.97 instead of 0.99 to improve the relevancy of each word. Next, we will sort the words in descending order by their frequencies and call the “wordcloud” method to build the word cloud on our data. The resulting visualization has far fewer words, but it has also eliminated many of the unnecessary terms (see Figure 10 in Appendix B). As a result, this image better reflects the important concepts of Trump’s presidential campaign when compared to our original model. And although this word cloud provides us with a glimpse of the Trump campaign's major themes, we can gain further insight by developing additional models.

Bar Graph

After building the word cloud, we began exploring other types of visualizations to represent our data. We chose to construct a bar graph to illustrate the most common words and their frequencies. Although it provides similar information as the word cloud, one advantage to using a bar graph is that it offers a more precise look at the frequency of each included word. In order to implement this visualization, the first step is to create a term document matrix (TDM) using our Trump corpus as the input parameter. A TDM is similar to the document term matrix (DTM) that we used to create our word cloud, but the difference is that a TDM uses the terms as the rows instead of the documents (Maceli, 2016). After creating the TDM, we sorted all of its words by frequency in descending order. From here, we examined the ten most common words and their frequencies (see Figure 11 in Appendix B). When compared to our word cloud, this image provides a much more specific depiction regarding how often each word is mentioned. We once again see that the two most common words are "Trump" and "realdonaldtrump," followed by words such as "great," "thank," and "Hillary." However, this figure also provides us with the specific number of times each word appeared in Trump's tweets. For example, Trump mentioned his name 1797 times on Twitter throughout his campaign.

After this step, we implemented our bar graph by using the "barplot" method in R. For the input parameters, we used the first ten most frequent words as our data source and set the graph’s color to red. Additionally, we used the argument “las = 2,” which positions the words vertically rather than horizontally (Gonzalez, 2017). This argument allows us to fit all of the words onto the plot. Of course, this graph shows that the words “Trump” and “realdonaldtrump” are by far the most common words in Trump’s tweets (see Figure 12 in Appendix B). However, the image emphasizes the scale of certain words’ frequencies when compared to others. For example, the third most frequent term “great” is mentioned almost 500 times less than the term “realdonaldtrump.” Also, terms such as “Hillary,” “America,” and “people” are mentioned with only a small fraction of the frequency that “Trump” is mentioned. These findings will be explored further in the “results” section of this paper.

Association Mining

One of the most interesting components of our text analysis is the association mining algorithm. This component is a major part of keyword-based association analysis, since it allows us to obtain words that frequently appear together. It is achieved by using the “findAssocs” function in R, which contains three arguments: the document term matrix, the word of interest, and the minimum percentage of instances in which the output words will appear with the input word (Maceli, 2016). We used this function seven times to find words associated with the terms “Ted,” “Marco,” “Hillary,” “Sanders,” “Paul,” “jobs,” and “coal.” The reason why we used first names such as “Ted” or “Marco” for some politicians instead of their last names is because we are seeking to determine the frequency of phrases and nicknames such as “lyin’ Ted” and “little Marco.” For each use of the “findAssocs” function, we used our Trump TDM as the input matrix. We also selected input percentages ranging between 0.1 and 0.3 and adjusted this value for each word to ensure that only the most relevant results were returned.

The output shows a series of terms that are most strongly associated with each of the input words (see Figures 13 and 14 in Appendix B). With regards to Ted Cruz, we see that the term "lyin'" is mentioned 46% of all instances where "Ted" is mentioned. We also see that the term "Canada" is frequently mentioned with regards to Ted Cruz, and the term "lightweight" is commonly associated with Marco Rubio. Likewise, the phrase "crooked" is mentioned 56% of times where Hillary Clinton's first name is mentioned. But aside from revealing the prominence of Trump's political attacks, we are also able to mine some of his major promises. For instance, we see that the word "jobs" is commonly associated with the terms "bring," "create," and "employ." Furthermore, the word "coal" is mentioned alongside words such as "decimate," "industries," and "plants." The high frequency of these words shows that Trump's message was consistent, which likely contributed to his popularity among voters. All of these findings will be examined in detail within the results section.

Word Cloud Results

In the next few sections, we will examine the output from all of our previous models. We will begin by examining our word cloud of the 30 most common words (see Figure 9 in Appendix B). As we know, the size of each word reflects its frequency—meaning that the largest words are the most commonly-occurring ones. This figure shows that "Trump" is the most frequently-occurring term, followed by "realdonaldtrump," "great," and "thank." The extensive usage of Trump's last name and his Twitter username "realdonaldtrump" are reflective of his ego and tendency to talk about his accomplishments. It is likely that this helped to convince voters that he is the strongest candidate in the race—which contributed to his support. This is consistent with his brand, which is based on putting his name "Trump" in large letters on his hotels and other properties.

We also generated a simplified word cloud that includes only the most relevant terms (see Figure 10 in Appendix B). As we see, the word "Trump" is still the most frequently used term. However, words such as "make," "America," and "great" are more prevalent in this word cloud. This is reflective of Trump's campaign slogan, "Make America Great Again," which was mentioned frequently throughout his campaign. The fact that this message was mentioned consistently suggests that it was able to resonate with voters who were discouraged by the current state of the country. Other frequent terms include "Hillary," "Clinton," "poll," "CNN," and "Fox News." These terms align with major themes in his campaign, such as criticisms of his rival Hillary Clinton and mainstream media corporations such as CNN. These terms also indicate a fixation with his own popularity, as evidenced by the frequent mention of his poll ratings and his citing of pro-Trump media companies such as Fox News.

Bar Graph Results

Next, we will analyze the bar graph of the ten most frequent words (see Figure 12 in Appendix B). As shown in the image, the words "Trump" and "realdonaldtrump" are the most common terms by far. One interesting finding is the massive difference in frequency between the word “realdonaldtrump” and the word “great” (the third most common word). According to Figure 11 in Appendix B, “realdonaldtrump” is mentioned 1,529 times while “great” is only mentioned 1,045 times. Furthermore, the terms “Hillary,” “America” and “Make America Great Again” are only mentioned about 500 times each—three times less often than the terms “Trump” and “realdonaldtrump.” This finding once again highlights Donald Trump’s narcissistic character. However, it may also improve his image by highlighting his strengths when compared to other candidates. To many voters, this may imply that only he can make America great again.

Association Mining Results

We will now examine our results from the keyword-based association analysis component. The first image in this section shows the common words associated with the terms “Ted,” “Marco,” “Hillary,” and “Sanders” (see Figure 13 in Appendix B)—referring to politicians Ted Cruz, Marco Rubio, Hillary Clinton, and Bernie Sanders respectively. Using these results, we can determine Trump’s sentiment towards his political opponents based on the words he used. We can ignore trivial results such as “Cruz” being associated with “Ted” or “Rubio” being associated with “Marco” since these refer to politicians’ last names. With Ted Cruz, we see that the word “lyin’” is used in 46% of all tweets where the term “Ted” is used. This means that Trump referred to Cruz as “lyin’ Ted” in nearly half of his tweets involving him, and it reflects how Trump often resorts to slander and name-calling towards his opponents. Trump used this technique to energize his support base and to draw attention to himself as the leading candidate and away from his opponents. The word “Canada” was mentioned in 24% of tweets involving Cruz, which reflects Trump’s claim that Cruz was born in Canada rather than in the U.S. This tactic was used to discredit Cruz from the presidency, strengthening Trump's chances of winning the Republican nomination.

When referring to Marco Rubio, common terms include "lightweight" (41%), "choker" (21%), and "little" (15%). Trump's base was comprised largely of individuals who were disassociated with the traditional Republican and Democratic parties. Therefore, he somewhat skillfully tailored his word usage to stir the pot by calling out his opponents' weaknesses. With regards to Hillary Clinton, he once again resorted to slander as evidenced by the fact that "crooked Hillary" is used in 56% of tweets mentioning her first name. Trump also mentions the terms "Bernie" (17%), "Sanders" (13%), and "rigged" (12%). Through this he highlights some of the controversies regarding Hillary Clinton, such as leaks involving the alleged rigging of the Democratic nomination against Bernie Sanders and in favor of Clinton. Furthermore, references to Bernie Sanders frequently include the term "disrespect" (20%)—indicating that Clinton and the Democratic party treated him unfairly during the election. By bringing up these issues, Trump helps to draw attention to Clinton's corruption and away from his own controversies (such as his comments towards women and minorities, and the suspected collusion involving Russia). As such, many voters were convinced that Trump was a more honest and reliable candidate than Hillary Clinton.

The last image shows the word associations involving the terms "Paul," "jobs," and "coal" (see Figure 14 in Appendix B). The term "Paul" refers primarily to Paul Ryan (65%), but it may also refer to Rand Paul as well (53%). Terms such as "brat," "spoiled," "entitlement," and "disloyalty" are frequently mentioned with regards to either of these two men. As always, Trump is using derogatory terms to respond to some negative comments made by politicians in his own party. When it comes to jobs on the other hand, frequent terms include “bring” (33%), “create” (19%), and “promised” (17%). Here, Trump is proposing a positive message by promising to bring millions of disenfranchised Americans back to work. And with regards to coal, key terms include “steel” (50%), “industry” (38%), “miners” (38%), and “decimate” (33%). Almost as provocative as his political attacks are Trump’s boasts about ending the war on clean beautiful coal, which he claims will put miners back to work. However, the fact that he wants to bring back the coal and steel industries has appealed to many Americans who have felt neglected by the previous administrations that allowed these industries to decay. Since these words were mentioned frequently throughout his campaign, they provided a sense of optimism to many voters that contrasts with the harsh negativity of the entire election.

Conclusion

Text mining is very effective for finding insights in unstructured data. Unstructured data consists of text without column labels. Unstructured data has exploded in volume with ever increasing internet usage. Although, we found that using R and text mining of tweets has limitations. In this section, we will highlight some limitations. In addition, we will offer some of our ideas for suggested improvements.

The bar plot showing the top 10 most frequent words indicated that the top six words were “Trump,” “realdonaldtrump,” “great,” “thank,” “Hillary,” and “MakeAmericaGreatAgain” (see Figure 12 in Appendix B). The first four words are not very unique. “Hillary” is interesting as her campaign was the number one target and he was running as an alternative to electing her. “MakeAmericaGreatAgain” is interesting as Trump’s campaign was focused on a solution to the suffering from job loss due to the Great Recession, as well as towards public emotions on immigration. However, the most frequent words from our text mining methods provide only limited insights by themselves. To fully understand Trump’s campaign, we will need more insights than these.

In order to provide more insights, we used keyword-based association analysis to provide some in-depth analysis of President Trump’s tweets. Keyword association is a technique used to find which words appear together frequently. Using R, we could produce hundreds of keyword associations. Rather than relying strictly on the R program output, we needed to use our intuition about the election to analyze the output. We focused on associations involving key candidates and politicians such as Hillary Clinton, Ted Cruz, Marco Rubio, Bernie Sanders, and Paul Ryan (see Figures 13 and 14 in Appendix B). We also looked at key concerns such as jobs and coal (see Figure 14 in Appendix B). From this, we were able to gather insights about Trump’s tweets—namely the condescending, negative tone and sentiment towards the other candidates and the emphasis on protecting jobs and freeing regulation on coal.

The last presidential election was unprecedented in the importance of those tweets. The media coverage of these tweets was extensive. The derogatory phrases and negative tone stood out in our frequent words and word associations. This election may not have been the first time an underrated candidate won, but it seems that Trump’s tweets garnered far more media attention in a crowded campaign field. This most frequent words in these tweets were memorable even if not well liked.

One suggestion for further analysis would be to build a time series graph mapping popular phrases and key terms over time. We could analyze the initial timing of these key words and observe how the usage of these words changed throughout the campaign. Then we could analyze the correlation of these terms towards the candidate polling. Trump’s approval rating rose quickly in the lead up to the Republican nomination on the basis that he was an alternative to Hillary Clinton and traditional Washington politics, and that he was focused on reversing the job losses suffered during the Great Recession. For example, it would be interesting to determine when “crooked Hillary,” “Make America Great Again,” “jobs,” and “coal” first appeared and how they affected Trump’s popularity. A time series analysis will require extensive and time-consuming programming in R, so we may consider implementing such a model in the future.

In conclusion, text mining in R is very useful for finding insights from large volumes of unstructured data. However, there are limitations and a learning curve for analysts to provide useful information. Knowledge of the data is required to determine keyword associations from the hundreds of code-generated associations. Time series analysis of the keywords and associations may be difficult to program, but it can be very useful for additional insight.

References

Basic Text Mining with R. (n.d.). Retrieved July 8, 2017, from https://rstudio-pubs-static.s3.amazonaws.com/132792_864e3813b0ec47cb95c7e1e2e2ad83e7.html

Han, J., Kamber, M., & Pei, J. (2011). *Data mining: concepts and techniques* (3rd ed.). Retrieved June 23, 2017.

Feinerer, I., & Hornik, K. (2017, March 2). Package ‘tm’. Retrieved August 7, 2017, from https://cran.r-project.org/web/packages/tm/tm.pdf

Gonzalez, B. (2017, March 6). Text Mining in R using the wordcloud and tm packages. Retrieved July 17, 2017, from https://rpubs.com/bgonzo/textmining

Maceli, M. (2016, July 19). Introduction to Text Mining with R for Information Professionals. Retrieved July 18, 2017, from http://journal.code4lib.org/articles/11626

Murphy, P. (2017, April 4). Basic Text Mining in R. Retrieved July 5, 2017, from https://rstudio-pubs-static.s3.amazonaws.com

The 11 Best Tweets of All Time (by Donald Trump). (n.d.). Retrieved June 24, 2017, from https://www.crowdbabble.com/blog/2016/11/14/the-11-best-tweets-of-all-time-by-donald-trump/

Appendix A

R Script for Group Project

```
# DATA 630 Group Project

# Written by Group 2

# Ed Perry, Daanish Ahmed, and Abdourahmane Bah

# Semester Summer 2017

# August 10, 2017

# Professor Edward Herranz


# This R script involves performing text mining analysis on a dataset containing

# President (then candidate) Donald J. Trump's Twitter tweets during the 2016

# presidential election.  The text mining processes include creation of a word

# cloud showing his most frequently-used words as well as a bar graph with the

# ten most common words and their frequencies.  The last algorithm is keyword-

# based association analysis, which will involve finding the terms that frequently

# appear with the input words and determining the relationship between these words.




# Loading the data and initializing packages.
# Directory (Change this to the directory you are using.)

setwd("~/Class Documents/2016-17 Summer/DATA 630/R/Group Project")


# Install the packages for text mining (remove #'s if you haven't installed them yet).


# Needed <- c("tm", "SnowballCC", "RColorBrewer", "ggplot2", "wordcloud", "biclust",

#          "cluster", "igraph", "fpc")

# install.packages(Needed, dependencies = TRUE)

# install.packages("Rcampdf", repos = "http://datacube.wu.ac.at/", type = "source")


# Loads the packages we need.

library(tm)

library(SnowballC)

library(wordcloud)


# Loads the data.

trump <- read.csv("Donald-Trump_7375-Tweets-Excel.csv", head=TRUE, sep=",")


# Views the dataset.

View(trump)


# Shows the initial structure of the dataset.
```

```
str(trump)

# End of loading the data.



# Preprocessing

# Converts "Date" variable to date type.
trump$Date <- as.Date(trump$Date, format = "%m/%d/%Y")
trump$Time <- NULL

# Removal of other unnecessary variables.
trump$X <- NULL
trump$X.1 <- NULL

# Shows the descriptive statistics of the dataset.
summary(trump)

# Shows the first 10 entries, excluding tweet text and URL.
head(trump[, -c(2, 7)], 10)
```

```
# This function reveals there are no missing values (NAs).
apply(trump, 2, function(trump) sum(is.na(trump)))

# Converts tweet text into corpus type.
trump_corpus <- Corpus(VectorSource(trump$Tweet_Text))

# Examines the first tweet.
inspect(trump_corpus[1])

# Additional stopwords and unnecessary words to remove.
stop = c("just", "good", "watch", "time", "join", "get", "big", "going", "much", "said",
         "like", "will", "now", "new", "can", "amp", "doesnt", "gave", "means", "one")

# Changes all letters to lowercase, removes numbers and punctuation, removes all
# stopwords, and strips whitespace.
trump_corpus = tm_map(trump_corpus, content_transformer(tolower))
trump_corpus = tm_map(trump_corpus, removeNumbers)
trump_corpus = tm_map(trump_corpus, removePunctuation)
trump_corpus = tm_map(trump_corpus, removeWords, c("the", "and", stop,
stopwords("english")))
trump_corpus =  tm_map(trump_corpus, stripWhitespace)
```

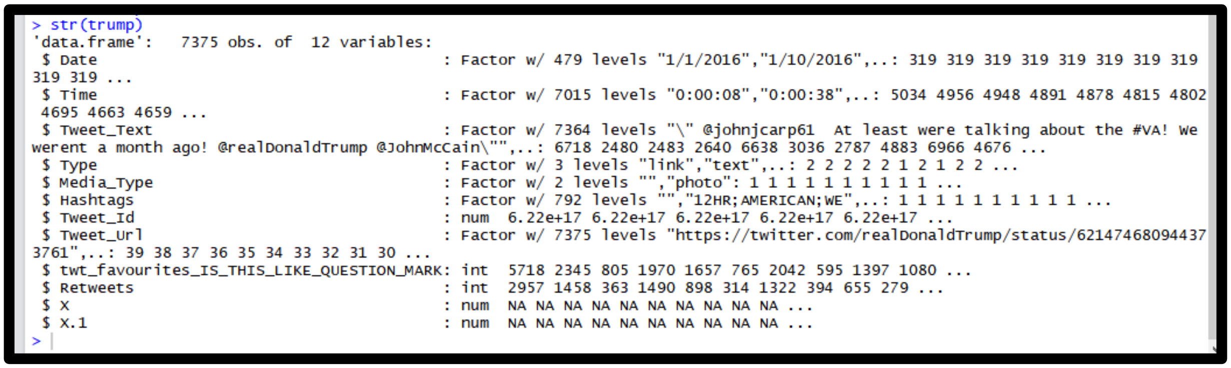
```
# Verifies that the first tweet has been preprocessed.
inspect(trump_corpus[1])

# End of preprocessing section.



# This section covers the creation of word clouds to visualize the most frequent
# words found in Trump's tweets.

# Analyze the textual data using a document-term matrix.
trump_dtm <- DocumentTermMatrix(trump_corpus)
trump_dtm

# Examine the document.
inspect(trump_dtm [20:25, 20:25])

# Reduce the dimension of the Document-Term Matrix (DTM).
trump_dtm <- removeSparseTerms(trump_dtm, 0.99)
trump_dtm

# Verify that the document's sparsity has decreased.
```

```
inspect(trump_dtm[1,1:20])

# Obtains the most frequent words, sorted in descending order by count.
findFreqTerms(trump_dtm, 5)
freq = data.frame(sort(colSums(as.matrix(trump_dtm)), decreasing = TRUE))

# Creates a word cloud using Trump's most frequent words, with a 30 word maximum.
wordcloud(rownames(freq), freq[,1], max.words = 30, random.order=TRUE, colors =
brewer.pal(1, "Dark2"))

# Use term frequency-inverse document frequency (TD-IDF) for more relevant results.
trump_dtm_tfidf <- DocumentTermMatrix(trump_corpus, control = list(weighting =
weightTfIdf))
trump_dtm_tfidf = removeSparseTerms(trump_dtm_tfidf, 0.97)

# Obtains the most frequent words, sorted in descending order by count.
freq = data.frame(sort(colSums(as.matrix(trump_dtm_tfidf)), decreasing=TRUE))

# Creates a second word cloud that includes fewer irrelevant words.
wordcloud(rownames(freq), freq[,1], max.words=30, colors=brewer.pal(3, "Dark2"))

# End of generating word clouds.
# This section covers the creation of a barplot showing the most common words

# and how often they are used.


# Creates a term-document matrix (TDM).

trump_tdm <- TermDocumentMatrix(trump_corpus)


# Sorts the word frequency in descending order.

freq <- sort(rowSums(as.matrix(trump_tdm)), decreasing=TRUE)


# Shows the 10 most common words and their frequencies.

freq[1:10]


# Creates a bar plot of the 10 most common words.

barplot(freq[1:10], col = "red", las = 2)


# End of generating bar plot.
# This section of code covers association mining, which involves finding words that
# commonly appear together.  It is useful for finding phrases such as "lyin' Ted,"
# "little Marco," etc.  It is also useful for understanding Trump's attitudes
# towards issues such as jobs and coal mining.

# It was obtained from "Introduction to Text Mining with R for Information Professionals"
# (https://rpubs.com/bgonzo/textmining).

# Finds common words associated with "Ted"
findAssocs(trump_tdm, term = "ted", 0.2)

# Finds common words associated with "Marco"
findAssocs(trump_tdm, term="marco", 0.15)

# Finds common words associated with "Hillary"
findAssocs(trump_tdm, term="hillary", 0.1)

# Finds common words associated with "Sanders"
findAssocs(trump_tdm, term="sanders", 0.15)

# Finds common words associated with "Paul"
findAssocs(trump_tdm, term="paul", 0.2)
```

```
# Finds common words associated with "jobs"

findAssocs(trump_tdm, term="jobs", 0.15)


# Finds common words associated with "coal"

findAssocs(trump_tdm, term="coal", 0.3)


# End of association mining section.


# End of script.
```

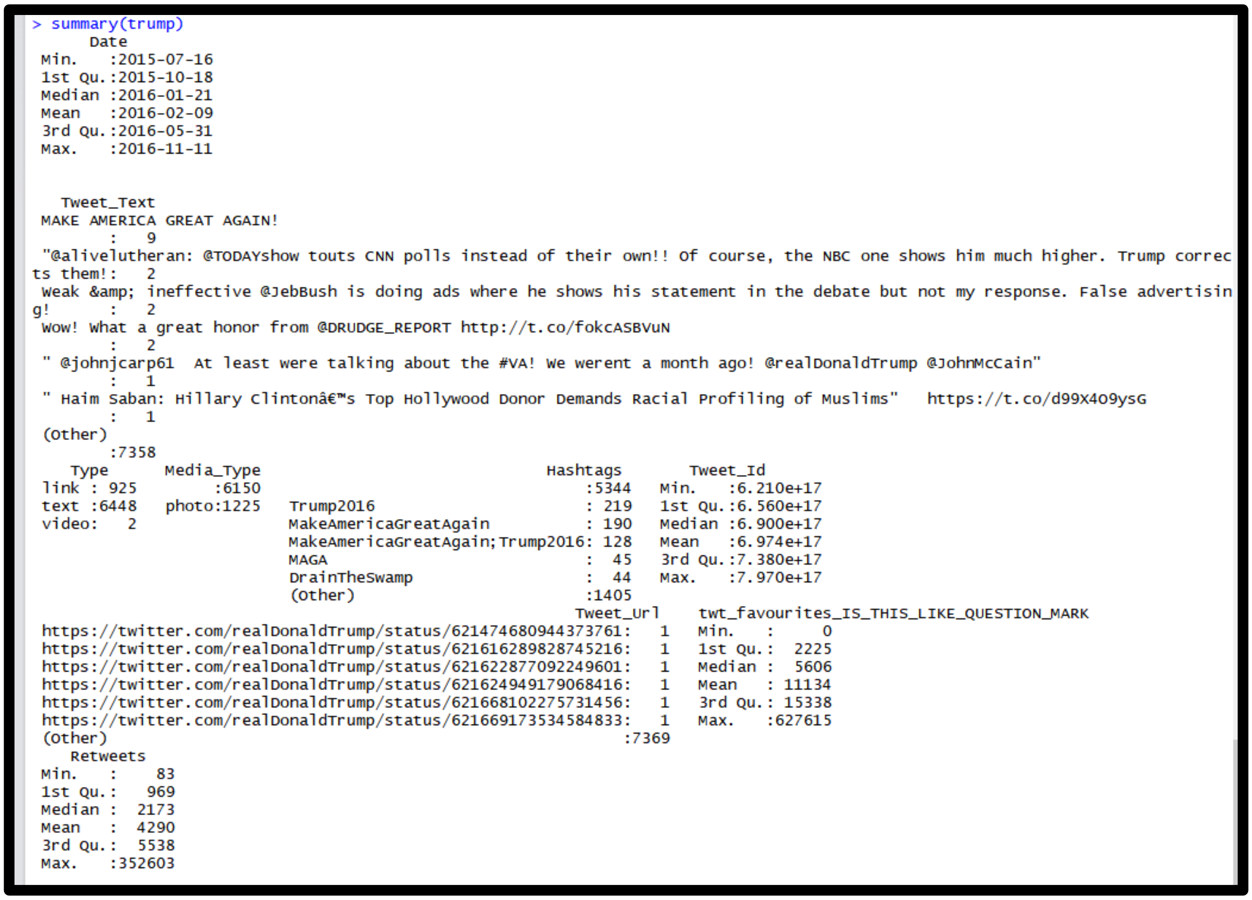
Appendix B

Relevant R Output Images



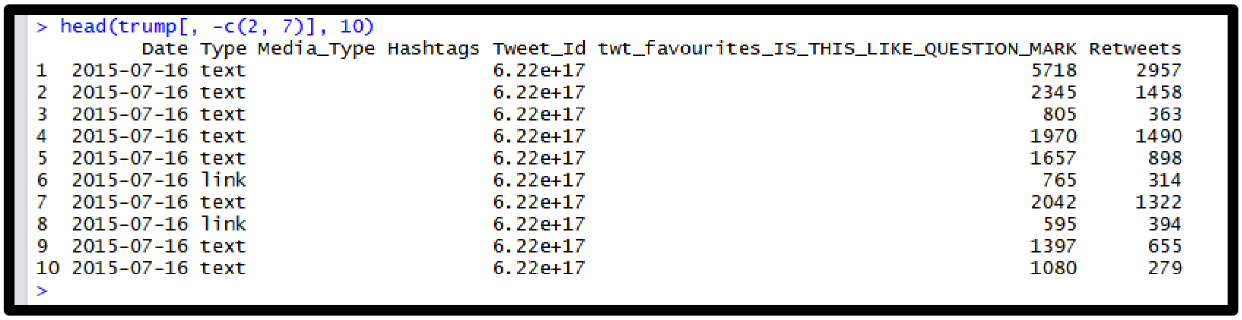
```
> str(trump)
'data.frame':	7375 obs. of  12 variables:
 $ Date                                  : Factor w/ 479 levels "1/1/2016","1/10/2016",..: 319 319 319 319 319 319 319 319
319 319 ...
 $ Time                                  : Factor w/ 7015 levels "0:00:08","0:00:38",..: 5034 4956 4948 4891 4878 4815 4802
 4695 4663 4659 ...
 $ Tweet_Text                            : Factor w/ 7364 levels "\" @johnjcarp61  At least were talking about the #VA! We
werent a month ago! @realDonaldTrump @JohnMcCain\"",..: 6718 2480 2483 2640 6638 3036 2787 4883 6966 4676 ...
 $ Type                                  : Factor w/ 3 levels "link","text",..: 2 2 2 2 2 1 2 1 2 2 ...
 $ Media_Type                            : Factor w/ 2 levels "","photo": 1 1 1 1 1 1 1 1 1 1 ...
 $ Hashtags                              : Factor w/ 792 levels "","12HR;AMERICAN;WE",..: 1 1 1 1 1 1 1 1 1 1 ...
 $ Tweet_Id                              : num  6.22e+17 6.22e+17 6.22e+17 6.22e+17 6.22e+17 ...
 $ Tweet_Url                             : Factor w/ 7375 levels "https://twitter.com/realDonaldTrump/status/62147468094437
3761",..: 39 38 37 36 35 34 33 32 31 30 ...
 $ twt_favourites_IS_THIS_LIKE_QUESTION_MARK: int  5718 2345 805 1970 1657 765 2042 595 1397 1080 ...
 $ Retweets                              : int  2957 1458 363 1490 898 314 1322 394 655 279 ...
 $ X                                     : num  NA NA NA NA NA NA NA NA NA NA ...
 $ X.1                                   : num  NA NA NA NA NA NA NA NA NA NA ...
> 
```

Figure 1. Initial Data Structure of Trump Dataset.



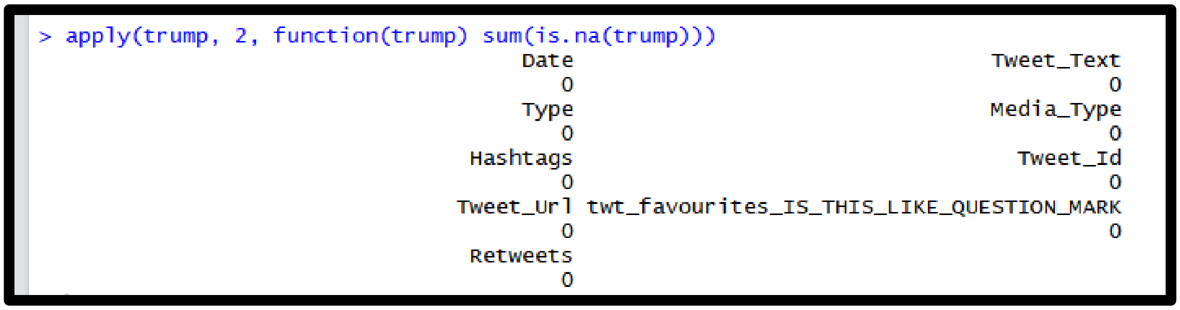
```
> summary(trump)
      Date           
 Min.   :2015-07-16  
 1st Qu.:2015-10-18  
 Median :2016-01-21  
 Mean   :2016-02-09  
 3rd Qu.:2016-05-31  
 Max.   :2016-11-11  


   Tweet_Text
 MAKE AMERICA GREAT AGAIN!
        :   9
 "@alivelutheran: @TODAYshow touts CNN polls instead of their own!! Of course, the NBC one shows him much higher. Trump correc
ts them!:   2
 Weak &amp; ineffective @JebBush is doing ads where he shows his statement in the debate but not my response. False advertisin
g!       :   2
 Wow! What a great honor from @DRUDGE_REPORT http://t.co/fokcASBVuN
        :   2
 " @johnjcarp61  At least were talking about the #VA! We werent a month ago! @realDonaldTrump @JohnMcCain"
        :   1
 " Haim Saban: Hillary Clintonâ€™s Top Hollywood Donor Demands Racial Profiling of Muslims"   https://t.co/d99X4O9ysG
        :   1
 (Other)
        :7358
    Type      Media_Type                              Hashtags       Tweet_Id        
 link : 925        :6150                                  :5344   Min.   :6.210e+17  
 text :6448   photo:1225   Trump2016                      : 219   1st Qu.:6.560e+17  
 video:   2                MakeAmericaGreatAgain          : 190   Median :6.900e+17  
                           MakeAmericaGreatAgain;Trump2016: 128   Mean   :6.974e+17  
                           MAGA                           :  45   3rd Qu.:7.380e+17  
                           DrainTheSwamp                  :  44   Max.   :7.970e+17  
                           (Other)                        :1405                      
                                                         Tweet_Url    twt_favourites_IS_THIS_LIKE_QUESTION_MARK
 https://twitter.com/realDonaldTrump/status/621474680944373761:   1   Min.   :     0                           
 https://twitter.com/realDonaldTrump/status/621616289828745216:   1   1st Qu.:  2225                           
 https://twitter.com/realDonaldTrump/status/621622877092249601:   1   Median :  5606                           
 https://twitter.com/realDonaldTrump/status/621624949179068416:   1   Mean   : 11134                           
 https://twitter.com/realDonaldTrump/status/621668102275731456:   1   3rd Qu.: 15338                           
 https://twitter.com/realDonaldTrump/status/621669173534584833:   1   Max.   :627615                           
 (Other)                                                      :7369                                            
    Retweets     
 Min.   :    83  
 1st Qu.:   969  
 Median :  2173  
 Mean   :  4290  
 3rd Qu.:  5538  
 Max.   :352603  
```

Figure 2. Descriptive Statistics of Dataset After Preprocessing.



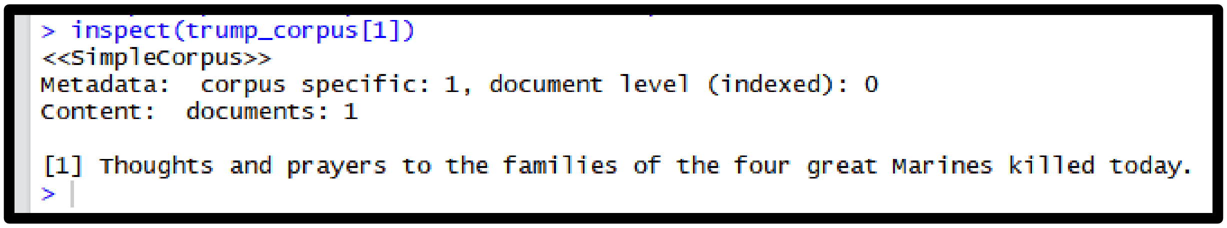
```
> head(trump[, -c(2, 7)], 10)
         Date Type Media_Type Hashtags Tweet_Id twt_favourites_IS_THIS_LIKE_QUESTION_MARK Retweets
1  2015-07-16 text                     6.22e+17                                      5718     2957
2  2015-07-16 text                     6.22e+17                                      2345     1458
3  2015-07-16 text                     6.22e+17                                       805      363
4  2015-07-16 text                     6.22e+17                                      1970     1490
5  2015-07-16 text                     6.22e+17                                      1657      898
6  2015-07-16 link                     6.22e+17                                       765      314
7  2015-07-16 text                     6.22e+17                                      2042     1322
8  2015-07-16 link                     6.22e+17                                       595      394
9  2015-07-16 text                     6.22e+17                                      1397      655
10 2015-07-16 text                     6.22e+17                                      1080      279
> 
```

Figure 3. First 10 Entries of the Dataset, Excluding Tweet Text and URL.



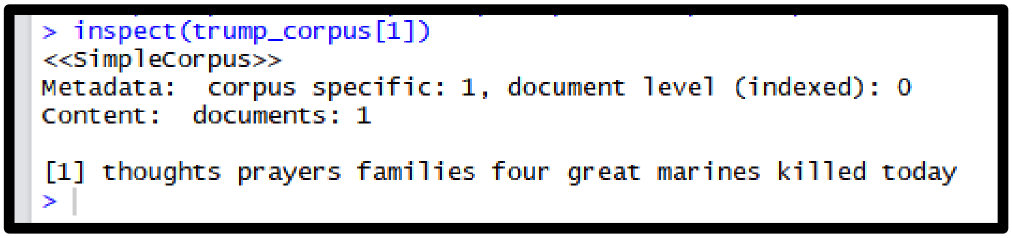
```
> apply(trump, 2, function(trump) sum(is.na(trump)))
                                Date                           Tweet_Text 
                                   0                                    0 
                                Type                           Media_Type 
                                   0                                    0 
                            Hashtags                             Tweet_Id 
                                   0                                    0 
                           Tweet_Url twt_favourites_IS_THIS_LIKE_QUESTION_MARK 
                                   0                                    0 
                            Retweets 
                                   0 
```

Figure 4. Number of Missing Values for All Variables in the Dataset.



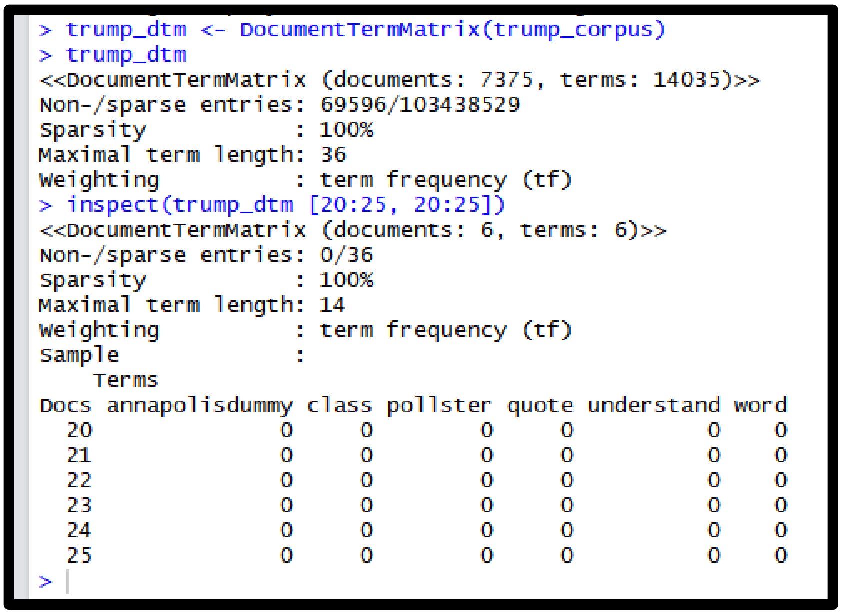
```
> inspect(trump_corpus[1])
<<SimpleCorpus>>
Metadata:  corpus specific: 1, document level (indexed): 0
Content:  documents: 1

[1] Thoughts and prayers to the families of the four great Marines killed today.
> 
```

Figure 5. First Tweet of the Trump Dataset Before Preprocessing.



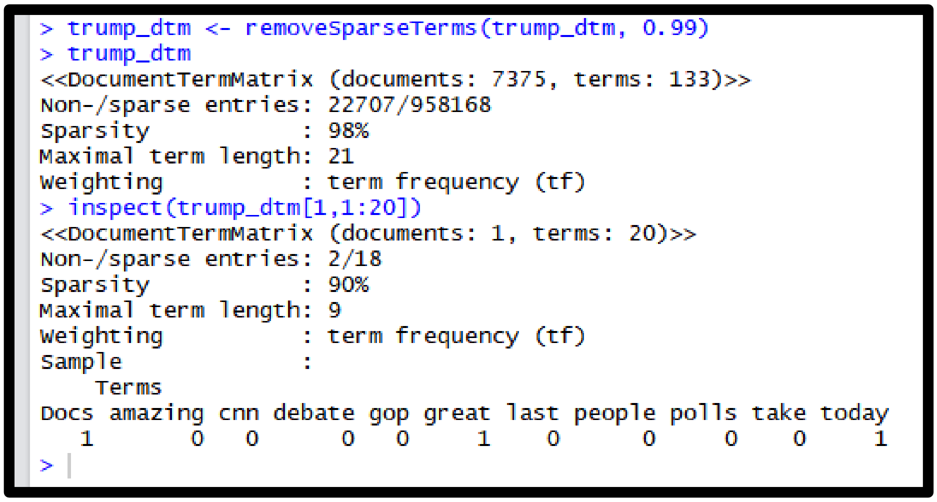
```
> inspect(trump_corpus[1])
<<SimpleCorpus>>
Metadata:  corpus specific: 1, document level (indexed): 0
Content:  documents: 1

[1] thoughts prayers families four great marines killed today
> 
```

Figure 6. First Tweet of the Trump Dataset After Preprocessing.



```
> trump_dtm <- DocumentTermMatrix(trump_corpus)
> trump_dtm
<<DocumentTermMatrix (documents: 7375, terms: 14035)>>
Non-/sparse entries: 69596/103438529
Sparsity           : 100%
Maximal term length: 36
Weighting          : term frequency (tf)
> inspect(trump_dtm [20:25, 20:25])
<<DocumentTermMatrix (documents: 6, terms: 6)>>
Non-/sparse entries: 0/36
Sparsity           : 100%
Maximal term length: 14
Weighting          : term frequency (tf)
Sample             :
    Terms
Docs annapolisdummy class pollster quote understand word
  20              0     0        0     0          0    0
  21              0     0        0     0          0    0
  22              0     0        0     0          0    0
  23              0     0        0     0          0    0
  24              0     0        0     0          0    0
  25              0     0        0     0          0    0
> 
```

Figure 7. Document Term Matrix Before Reducing Dimension.



```
> trump_dtm <- removeSparseTerms(trump_dtm, 0.99)
> trump_dtm
<<DocumentTermMatrix (documents: 7375, terms: 133)>>
Non-/sparse entries: 22707/958168
Sparsity           : 98%
Maximal term length: 21
Weighting          : term frequency (tf)
> inspect(trump_dtm[1,1:20])
<<DocumentTermMatrix (documents: 1, terms: 20)>>
Non-/sparse entries: 2/18
Sparsity           : 90%
Maximal term length: 9
Weighting          : term frequency (tf)
Sample             :
    Terms
Docs amazing cnn debate gop great last people polls take today
   1       0   0      0   0     1    0      0     0    0     1
> 
```

Figure 8. Document Term Matrix After Reducing Dimension.

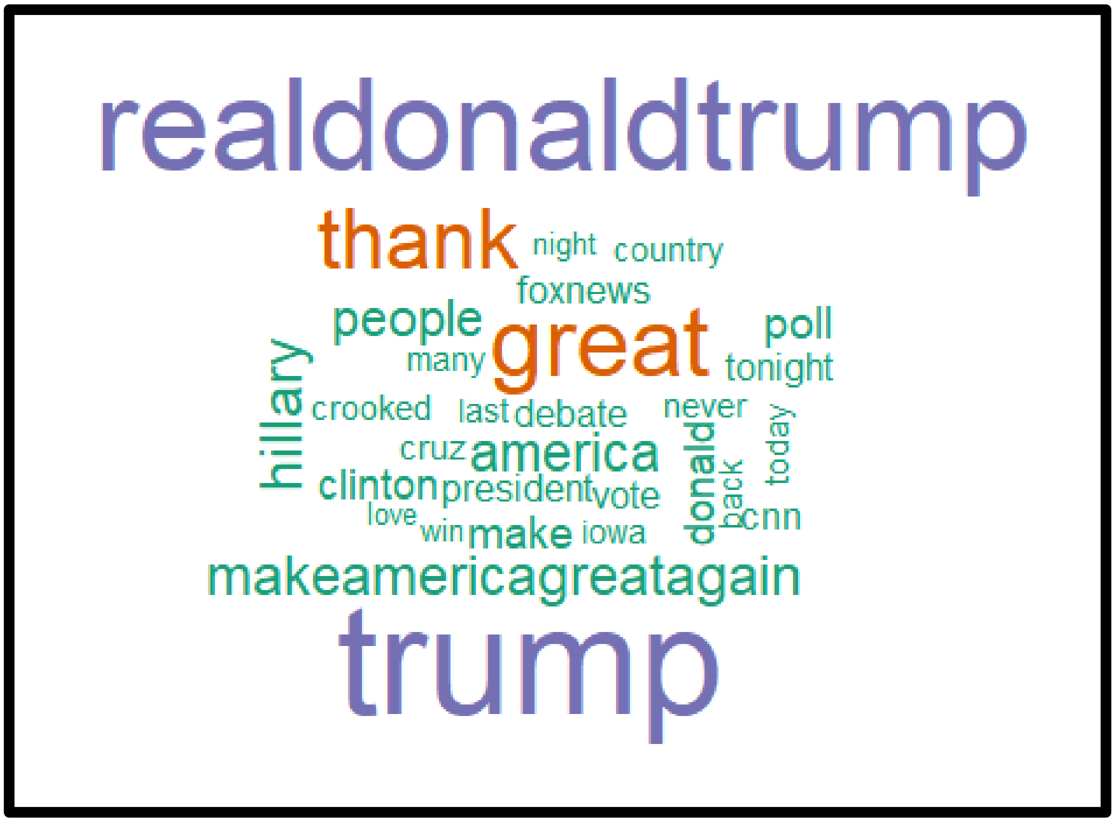


Figure 9. Word Cloud of the 30 Most Common Words.

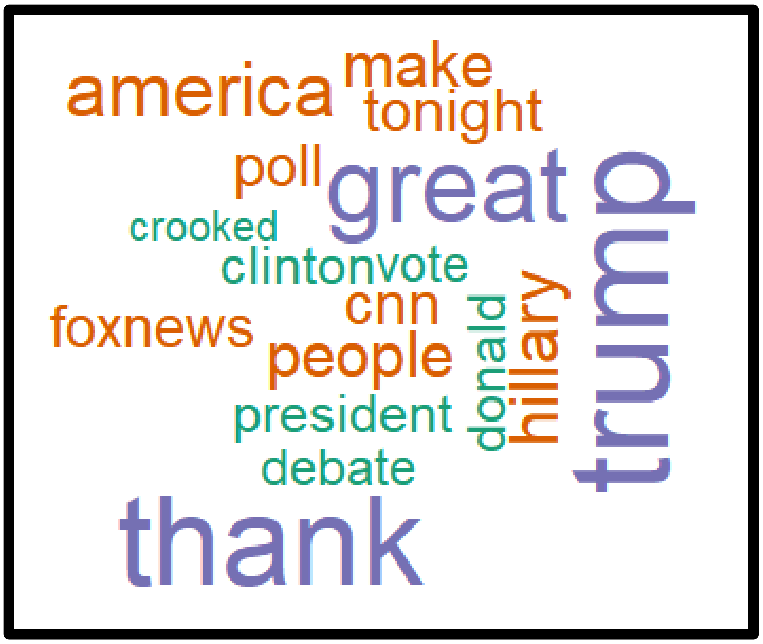
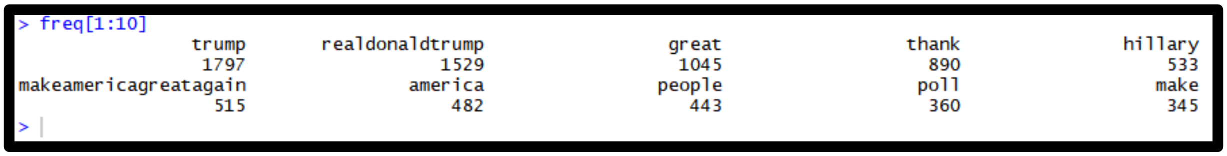


Figure 10. Simplified Word Cloud Showing the Most Relevant Words.



```
> freq[1:10]
               trump          realdonaldtrump                great                thank              hillary 
                1797                     1529                 1045                  890                  533 
makeamericagreatagain                  america               people                 poll                 make 
                 515                      482                  443                  360                  345 
> |
```

Figure 11. Top 10 Most Frequent Words and Their Counts.

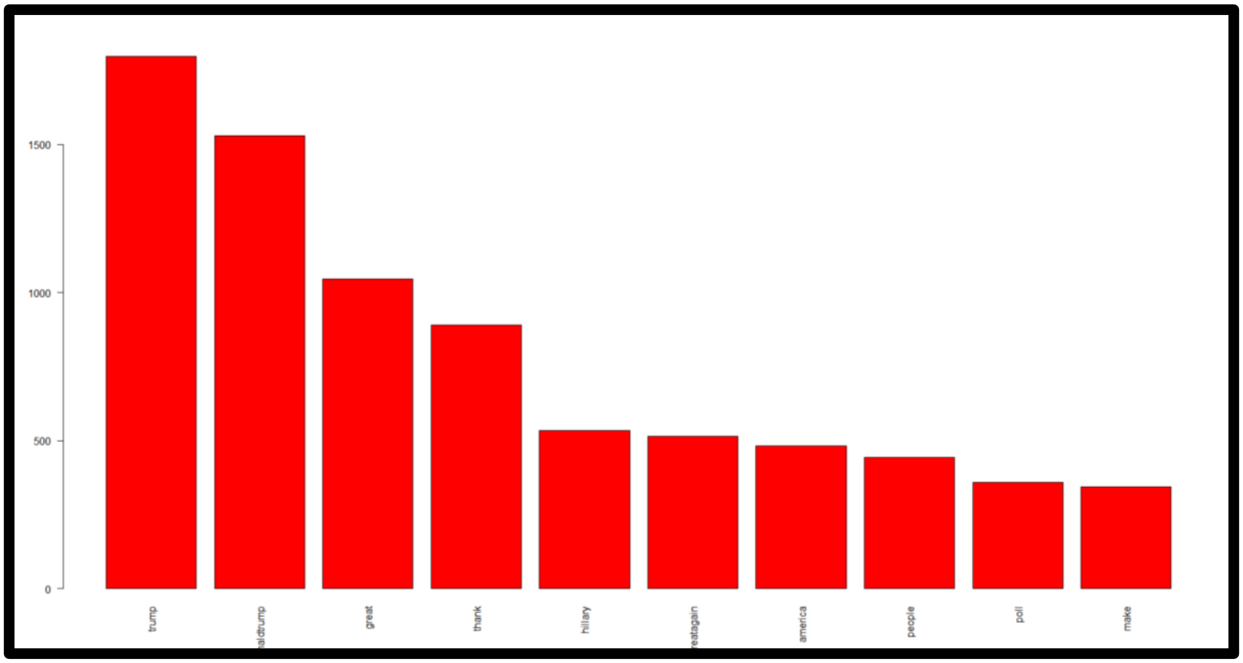
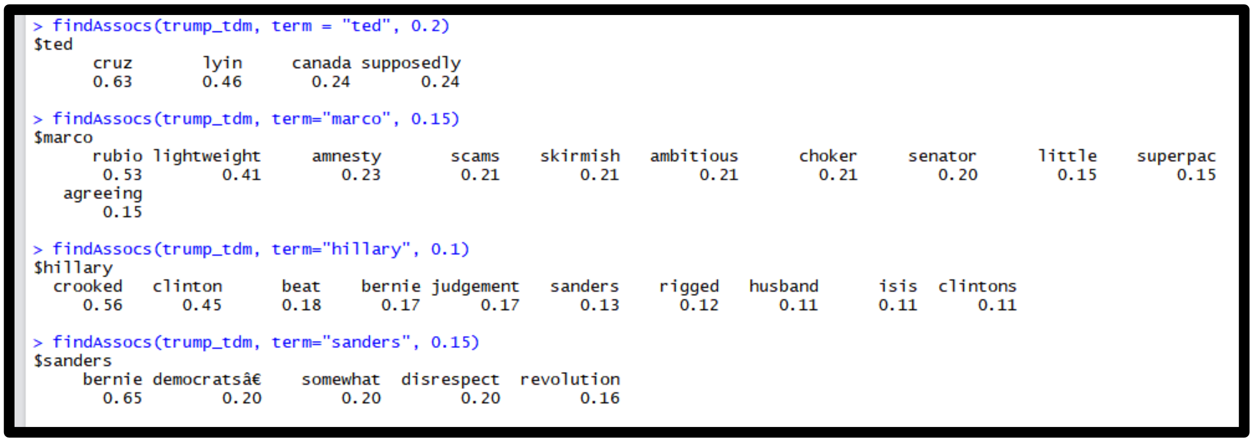
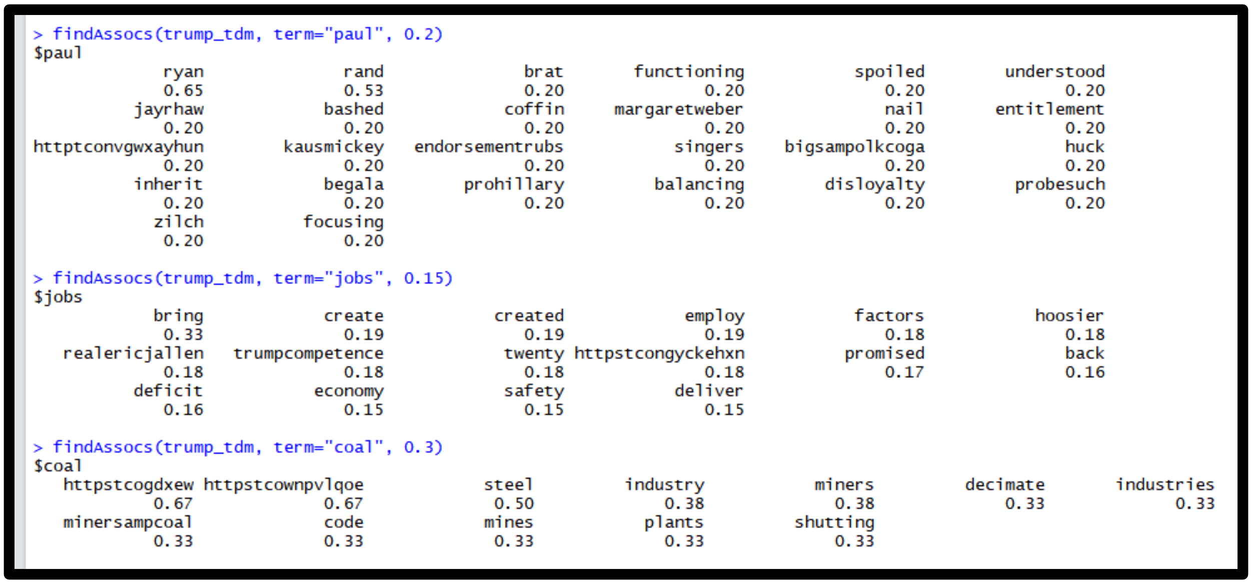


Figure 12. Bar Plot of the Top 10 Most Frequent Words.



```
> findAssocs(trump_tdm, term = "ted", 0.2)
$ted
      cruz       lyin     canada supposedly 
      0.63       0.46       0.24       0.24 

> findAssocs(trump_tdm, term="marco", 0.15)
$marco
      rubio lightweight     amnesty       scams    skirmish   ambitious      choker     senator      little    superpac 
       0.53        0.41        0.23        0.21        0.21        0.21        0.21        0.20        0.15        0.15 
   agreeing 
       0.15 

> findAssocs(trump_tdm, term="hillary", 0.1)
$hillary
  crooked   clinton      beat    bernie judgement   sanders    rigged   husband      isis  clintons 
     0.56      0.45      0.18      0.17      0.17      0.13      0.12      0.11      0.11      0.11 

> findAssocs(trump_tdm, term="sanders", 0.15)
$sanders
     bernie democratsâ€     somewhat  disrespect  revolution 
       0.65        0.20        0.20        0.20        0.16 
```

Figure 13. Common Words Associated With “Ted,” “Marco,” “Hillary,” and “Sanders.”



```
> findAssocs(trump_tdm, term="paul", 0.2)
$paul
             ryan              rand              brat       functioning           spoiled        understood 
             0.65              0.53              0.20              0.20              0.20              0.20 
          jayrhaw            bashed            coffin     margaretweber              nail       entitlement 
             0.20              0.20              0.20              0.20              0.20              0.20 
httptconvgwxayhun        kausmickey   endorsementrubs           singers    bigsampolkcoga              huck 
             0.20              0.20              0.20              0.20              0.20              0.20 
          inherit            begala        prohillary         balancing        disloyalty         probesuch 
             0.20              0.20              0.20              0.20              0.20              0.20 
            zilch          focusing 
             0.20              0.20 

> findAssocs(trump_tdm, term="jobs", 0.15)
$jobs
            bring            create           created            employ           factors           hoosier 
             0.33              0.19              0.19              0.19              0.18              0.18 
   realericjallen   trumpcompetence            twenty httpstcongyckehxn          promised              back 
             0.18              0.18              0.18              0.18              0.17              0.16 
          deficit           economy            safety           deliver 
             0.16              0.15              0.15              0.15 

> findAssocs(trump_tdm, term="coal", 0.3)
$coal
   httpstcogdxew httpstcownpvlqoe            steel         industry           miners         decimate       industries 
            0.67             0.67             0.50             0.38             0.38             0.33             0.33 
   minersampcoal             code            mines           plants         shutting 
            0.33             0.33             0.33             0.33             0.33 
```

Figure 14. Common Words Associated With “Paul,” “Jobs,” and “Coal.”