Analyzing Word Frequency and Sentiment of Obama and Trump Tweets

Dylan Wiwad

March 27, 2018

Word Counts and Sentiment Analysis: Barack Obama versus Donald Trump

In this notebook I did a very cursory analysis of the most frequently used words on Twitter by both Barack Obama and Donald Trump. I pulled the most recent 3,200 tweets from each account and simply counted the frequency of each word, excluding hashtags, websites, mentions, etc.

In the tweets I pulled I included all retweets and replies as well. If I look only at pure tweets made, userTimeLine function from the TwitteR package pulls only 383 tweets from Obama, and 554 tweets from Trump. I suspect this is twitter limiting the amount I can pull, or how far back - otherwise I'm not sure why the tweets are limited in this way. One interesting thing to note there - Trumps 554 tweets go back to November 17th, 2017. Obama's go back to March 14th, 2016. Trump is tweeting roughly three times a day, to Obama's one tweet every two days.

```r
# Bring in the packages I'm going to need
library(formatR)
library(ggplot2)
library(stringr)
library(twitteR)
library(ROAuth)
library(plyr)
library(httr)
library(tm)
```

Anyways, off to the word counts. First I'm just going to define a small function to append something to a list, that we need for the data cleaning.

```r
lappend <- function(lst, obj) {
  lst[[length(lst)+1]] <- obj
  return(lst)
}
```

Obama

The first thing I'm going to do is grab all of Obama's tweets using the userTimeline function from the TwitteR package. In order to do this I had to setup an account with Twitter to get an API key. You can do this on apps.twitter.com. Then I entered the keys up above, where you would enter your own to use the script.

I have hidden the code that I used to set up my api key, because I can't share it as it is linked to my twitter account. However, here is the chunk I used:

api_key <- “”
api_secret <- “”
access_token <- “”
access_token_secret <- “”
setup_twitter_oauth(api_key, api_secret, access_token, access_token_secret)

If you want to use this code, or work with twitter, you need to make an account as described above, insert your keys in the quotation marks above and run the setup_twitter_oauth command.

So with that in mind, let’s call twitter and get obama’s tweets

```r
# Grabs Obama's most recent 3200 tweets
tweets_obama <- userTimeline("BarackObama", n = 3200, includeRts = TRUE,
    excludeReplies = FALSE)
# Convert them to a DF
tweets_obama <- twListToDF(tweets_obama)
colnames(tweets_obama)  # Print the first few rows just to see whats going on.
```

```
##  [1] "text"          "favorited"     "favoriteCount" "replyToSN"
##  [5] "created"       "truncated"     "replyToSID"    "id"
##  [9] "replyToUID"    "statusSource"  "screenName"    "retweetCount"
## [13] "isRetweet"     "retweeted"     "longitude"     "latitude"
```

As you can see, the above data frame has sixteen columns, including information like the number of retweets or likes for a given tweet, the exact day and time the tweet was posted, etc. Here, I’m really onle interested in the actual content of the tweet so lets pull that out.

```r
obama_content <- tweets_obama$text
head(obama_content)
```

```
## [1] "Incredible to have a Chicago team in the Final Four. I’ll take that over an intact bracket any 
## [2] "Michelle and I are so inspired by all the young people who made today’s marches happen. Keep at
## [3] "Our most important task as a nation is to make sure all our young people can achieve their drea
## [4] "In Singapore with young people who are advocating for education, empowering young women, and ge
## [5] "41: I like the competition. And the loyalty to the home team. - 44 https://t.co/XG3ChMtWOM"
## [6] "Congrats to @LoyolaChicago and Sister Jean for a last-second upset - I had faith in my pick!"
```

Next, we’ll just do a chunk of data cleaning with our new dataframe that is only one column - the tweet itself. First thing I’m going to do is split each tweet by the space character, so each word is its own element. Then I’m going to use a small for loop to append each of those words to a new list, called obama_content. This is where we bring back in the lappend function we defined above. What we end up with is a huge list of individual words.

```r
obama_split <- strsplit(obama_content, " ")
obama_together <- c() # This makes our empty list we are going to append each word to

for (row in obama_split){
  for (word in row){
    obama_together <- lappend(obama_together, word)
  }
}

# print out the first few row of our new list
head(obama_together, n=25)
```

```
##  [1] "Incredible"              "to"
##  [3] "have"                    "a"
##  [5] "Chicago"                 "team"
##  [7] "in"                      "the"
##  [9] "Final"                   "Four."
## [11] "I’ll"                    "take"
## [13] "that"                    "over"
## [15] "an"                      "intact"
## [17] "bracket"                 "any"
## [19] "day!"                    "Congratulations..."
## [21] "https://t.co/V9IbaSlbIp" "Michelle"
## [23] "and"                     "I"
## [25] "are"
```

So here are the first 25 elements in the list. You can see its just a couple tweets broken down word by word. It's not only words, though, as there is a link in there as well. This list will be full of links, hashtags, mentions, filler words, etc. Another thing to note is how some words will be capitalized and some will have punctuation attached to them, etc. This will make words like "Final" and "final," or "day!" and "day" different.

So, in a bit more cleaning we'll import a dictionary of stopwords to remove, we'll remove all the emojis, make everything lowercase, and remove all punctuation. Removing all the punctuation in this way creates a problem a bit later where a couple words that are hashtags (e.g., "#actionclimate") get ranked as highly frequently used words. I'm just going to skip over those when we visualize the data.

```
# Deletes all non-alpha-numeric characters
obama_together <- iconv(obama_together, 'ASCII', 'UTF-8', sub='')
# Makes everything lower case
obama_together <- tolower(obama_together)
# Brings in our dictionary of stopwords from the TM package
stopwords <- stopwords('en')
# Removes any element from "stopwords" from our list
obama_together <- removeWords(obama_together, stopwords)
# Removes punctuation
obama_together <- removePunctuation(obama_together)
# Turns out newly cleaned list into a single column DF
obama_together <- as.data.frame(obama_together)
```

Now, with all that done we have a newly cleaned up DF of individual words from Obama's tweets that we are ready to count and display.

```
obama_counts <- count(obama_together$obama_together)
obama_counts <- obama_counts[order(-obama_counts$freq),]
head(obama_counts, n=20)
```

```
##                  x  freq
## 1                  17861
## 6333    president  1201
## 5955        obama  1060
## 6714           rt   366
## 400  actonclimate   288
## 954        change   214
## 7641        watch   202
## 1048      climate   199
## 7690   whitehouse   199
## 5937          now   179
## 7319         time   171
## 2200       health   167
## 514     americans   159
## 7328        today   155
## 1499    doyourjob   150
## 6817       senate   149
## 5622         make   147
## 5545         live   146
## 7702         will   142
## 2012          get   140
```

So, here are the top twenty words from Obama, with a couple caveats. The top "word" is nothing - this is all leftover from our cleaning. Everything we removed is being counted here. Theres a couple other words here that we won't count when we visualize these data. Namely, in the top ten we have "rt," as well as "actionclimate", "doyourjob" and "whitehouse" which are hashtags. There are also a few filler words that came through, such as "will" and "get." We'll skip over these later.

For now, let's move on and repeat the entire process with Donald Trump. I'm going to comment this next bit less as it is a direct replication of what we did above. Afterwards, we'll visualize both Obama's

Trump

Doing the same for Trump, lumping everything from getting the tweets to cleaning them in one code chunk.

```
tweets_trump <- userTimeline("realDonaldTrump", n = 3200, includeRts = TRUE,
    excludeReplies = FALSE)
tweets_trump <- twListToDF(tweets_trump)
trump_content <- tweets_trump$text

# Split each word and then append to my new list with for
# loop
trump_split <- strsplit(trump_content, " ")
trump_together <- c()


for (row in trump_split) {
    for (word in row) {
        trump_together <- lappend(trump_together, word)
    }
}
# Get rid of Emojis, make lowercase, remove stopwords
trump_together <- iconv(trump_together, "ASCII", "UTF-8", sub = "")
trump_together <- tolower(trump_together)
trump_together <- removeWords(trump_together, stopwords)
trump_together <- removePunctuation(trump_together)

# Turn his tweets into a DF and get the frequency counts
trump_together <- as.data.frame(trump_together)
trump_counts <- count(trump_together$trump_together)
trump_counts <- trump_counts[order(-trump_counts$freq), ]
head(trump_counts, n = 20)
```

```
##                 x  freq
## 1                 24249
## 2809     great   562
## 8174      will   555
## 596         amp   453
## 6786         rt   332
## 7892         us   243
## 6024    people   222
## 5694      news   219
## 4902      just   213
## 7568     thank   206
## 7648     today   198
## 6227 president   188
## 2397      fake   187
## 7758     trump   175
## 923        big   169
## 5751       now   163
## 572    america   158
## 1589   country   155
## 7500       tax   153
## 5316      many   139
```

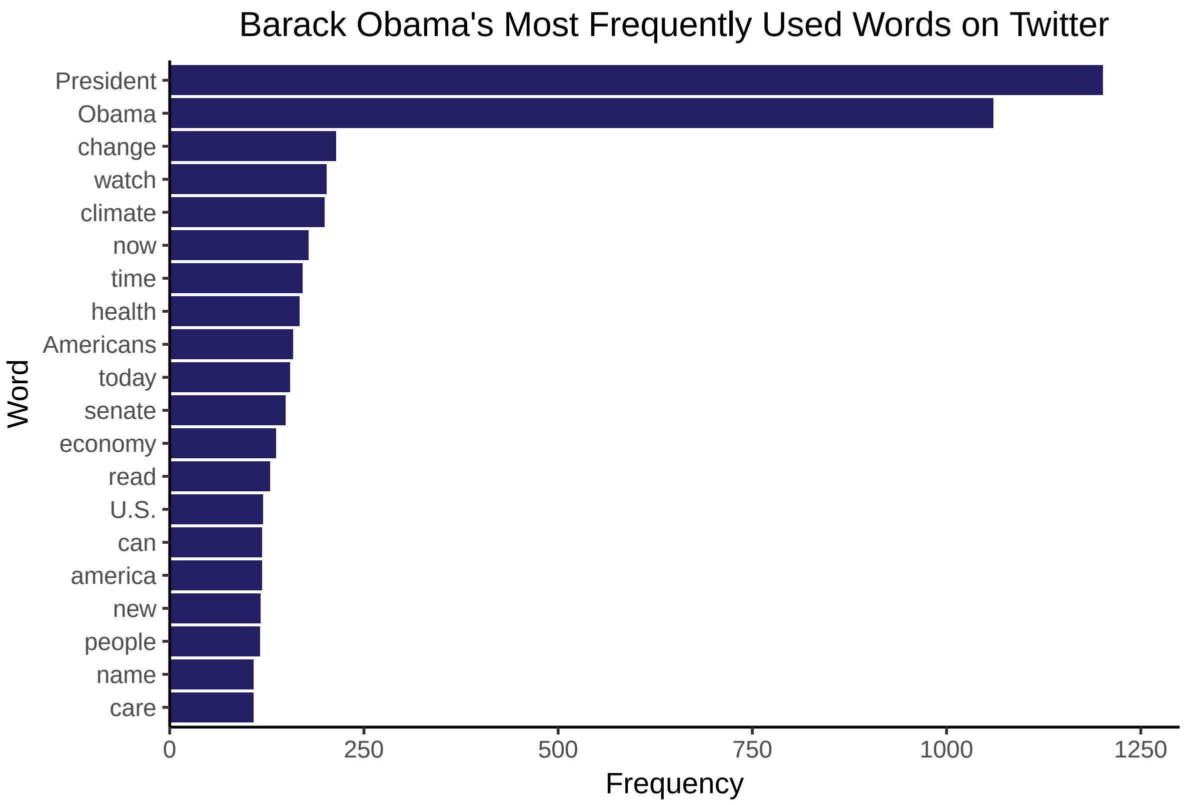
So, for Trump we get some of the same issues as Obama - for instance, "will" is a filler word, and "amp" is a distortion of the ampersand character, additionally, "us" is actually "U.S." after being stripped of punctuation and made to lower case. Again, we'll account for these things when we visualize the tweets below!

Most frequent words used by Obama and Trump Visualized

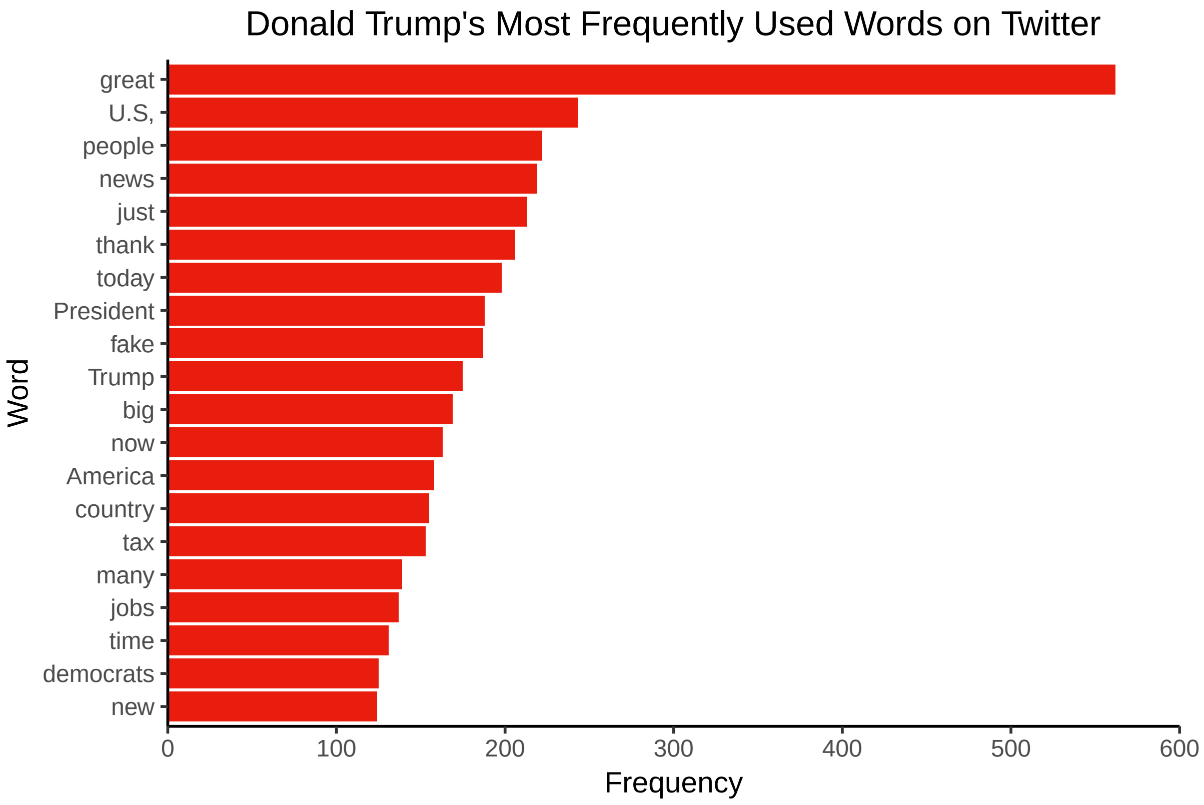
In order to make these counts a little more intuitive to interpret, let's visualize them. I'm doing this manually to get around the couple of issues I mentioned above. Namely, just working around the words that are used frequently but dont count such as 'rt' or 'actionclimate' or other filler words.

```
# Data Frame of word counts for each Obama and Trump
word_countsO <- data.frame(words = c("name", "care", "people",
    "new", "can", "america", "U.S.", "read", "economy", "senate",
    "today", "Americans", "health", "time", "now", "climate",
    "watch", "change", "Obama", "President"), counts = c(108,
    108, 116, 117, 119, 119, 120, 129, 137, 149, 155, 159, 167,
    171, 179, 199, 202, 214, 1060, 1201))

word_countsT <- data.frame(words = c("new", "democrats", "time",
    "jobs", "many", "tax", "country", "America", "now", "big",
    "Trump", "fake", "President", "today", "thank", "just", "news",
    "people", "U.S,", "great"), counts = c(124, 125, 131, 137,
    139, 153, 155, 158, 163, 169, 175, 187, 188, 198, 206, 213,
    219, 222, 243, 562))

# Make the two plots
obama_words <- ggplot(word_countsO, aes(x = reorder(words, counts),
    y = counts)) + geom_bar(stat = "identity", fill = "#232066") +
    coord_flip() + labs(x = "Word", y = "Frequency") + ggtitle("Barack Obama's Most Frequently Used Wor
    theme_bw() + theme(panel.border = element_blank(), panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(), axis.line = element_line(colour = "black"),
    plot.title = element_text(hjust = 0.5), plot.margin = unit(c(0.5,
        0.5, 0.5, 0.5), "cm")) + scale_y_continuous(expand = c(0,
    0), limits = c(0, 1300))

trump_words <- ggplot(word_countsT, aes(x = reorder(words, counts),
    y = counts)) + geom_bar(stat = "identity", fill = "#E91D0E") +
    coord_flip() + labs(x = "Word", y = "Frequency") + ggtitle("Donald Trump's Most Frequently Used Wor
    theme_bw() + theme(panel.border = element_blank(), panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(), axis.line = element_line(colour = "black"),
    plot.title = element_text(hjust = 0.5), plot.margin = unit(c(0.5,
        0.5, 0.5, 0.5), "cm")) + scale_y_continuous(expand = c(0,
    0), limits = c(0, 600))
```

```
obama_words
```



```
trump_words
```



There appears to be some subjective differences between the types of words that Obama uses the most versus Trump. I highlight some limitations and challenges with this at the bottom of this document. It seems odd that Obama's top words are 'President' and 'Obama,' but interesting nonetheless. Obama seems to use a lot of progressive, abstract outcome-oriented words (e.g., health, economy, read, can, new, etc). Trump, on the other hand talks a lot using descriptors and talking about specific issues (e.g., big, great, fake, news, jobs, tax, etc). Again, this is a completely subjective cursory glance at the top twenty words of either.

Can we actually glean differences in their content, though? One way of doing this might be through a sentiment analysis.

Sentiment Analysis

So, we can see from the counts and visualizations that the words used by Donald Trump and Barack Obama seem to contain different content, to some extent. Though, this is obviously subjective from a cursory glance of looking at the counts.

I wanted to take this one step further and do a small sentiment analysis - do Obama and Trump speak differently in terms of the emotions they use? My original hypothesis was simply that Trump would express much more negative sentiment than Obama. This turned out not to be true - let's see what the actual differences are.

Before I do anything, we need to bring in the dictionaries of positive and negative words and create a function that scores the sentiment of the tweets.

The lists of positive and negative words came from Hu & Liu (2004), and the function came from this medium article: https://medium.com/@rohitnair_94843/analysis-of-twitter-data-using-r-part-3-sentiment-analysis-53d0e5359cb8

```
# Get the dictionaries of pos and neg words
pos <- read.csv("pos.csv", header=FALSE)
neg <- read.csv("neg.csv", header=FALSE)

# Scan the words in
pos.words <- scan("pos.csv", what="character")
neg.words <- scan("neg.csv", what="character")

# Function to compare and get the sentiment of Obama and Trump
score.sentiment = function(sentences, pos.words, neg.words, .progress='none')
{
  require(plyr)
  require(stringr)

  # we got a vector of sentences. plyr will handle a list
  # or a vector as an "l" for us
  # we want a simple array ("a") of scores back, so we use
  # "l" + "a" + "ply" = "laply":

  scores = laply(sentences, function(sentence, pos.words, neg.words) {

    # clean up sentences with R's regex-driven global substitute, gsub():
    sentence = gsub('[[:punct:]]', '', sentence)
    sentence = gsub('[[:cntrl:]]', '', sentence)
    sentence = gsub('\\d+', '', sentence)
    # and convert to lower case:
    sentence = tolower(sentence)

    # split into words. str_split is in the stringr package
    word.list = str_split(sentence, '\\s+')
    # sometimes a list() is one level of hierarchy too much
    words = unlist(word.list)

    # compare our words to the dictionaries of positive & negative terms
    pos.matches = match(words, pos.words)
    neg.matches = match(words, neg.words)

    # match() returns the position of the matched term or NA
    # we just want a TRUE/FALSE:
    pos.matches = !is.na(pos.matches)
    neg.matches = !is.na(neg.matches)

    # and conveniently enough, TRUE/FALSE will be treated as 1/0 by sum():
    score = sum(pos.matches) - sum(neg.matches)

    return(score)
  }, pos.words, neg.words, .progress=.progress )

  scores.df = data.frame(score=scores, text=sentences)
  return(scores.df)
}
```

Now that this is set up, all we need to do is apply the function to Obama and Trumps content. I went back to the original tweet files for this - as in, the data as I pulled it from Twitter, simply working with the 'text' column containing the tweet content.

As such, I need to again strip emojis from them. I'll do this for Trump and Obama together.

```
tweets_obama$text <- iconv(tweets_obama$text, 'ASCII', 'UTF-8', sub='')
tweets_trump$text <- iconv(tweets_trump$text, 'ASCII', 'UTF-8', sub='')
```

Now, let's just apply the above sentiment score function to both Obama and Trump.

```
obama_sent <- score.sentiment(tweets_obama$text, pos.words, neg.words)
trump_sent <- score.sentiment(tweets_trump$text, pos.words, neg.words)
```

So, lets get the counts for these. The way the function is scored is that 0 is neutral, more positive values correspond to more positive language, and negative values correspond to more negative language. First is Obama:

```
count(obama_sent$score)
```

```
##    x freq
## 1 -3    5
## 2 -2   63
## 3 -1  311
## 4  0 1465
## 5  1  947
## 6  2  322
## 7  3   69
## 8  4   13
## 9  6    1
```

then Trump:

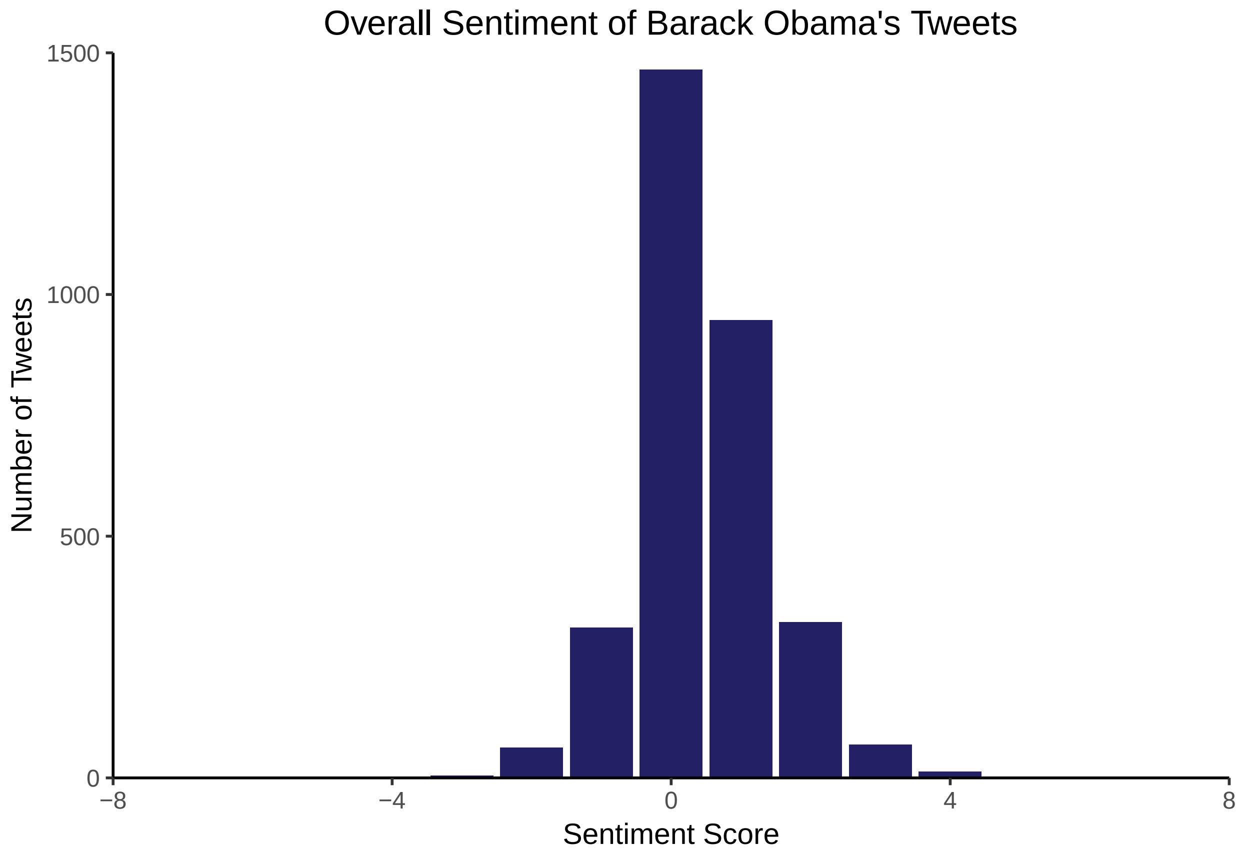
```
count(trump_sent$score)
```

```
##     x freq
## 1  -7    1
## 2  -6    1
## 3  -5   16
## 4  -4   25
## 5  -3   77
## 6  -2  225
## 7  -1  415
## 8   0  889
## 9   1  824
## 10  2  469
## 11  3  167
## 12  4   74
## 13  5   10
## 14  6    2
```

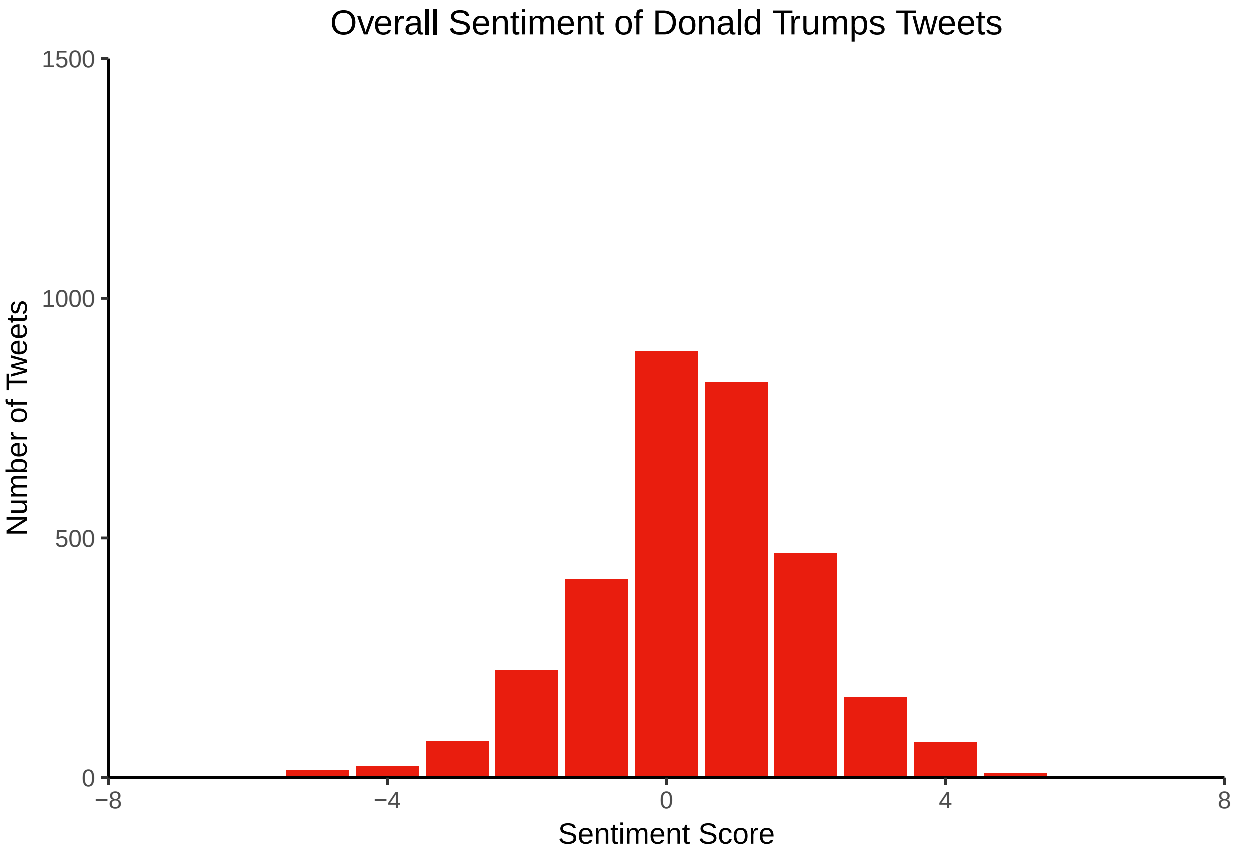
And a couple quick visualizations of the distribution of their sentiment scores:

```
obama_sent_plot <- qplot(obama_sent$score, xlab = "Sentiment Score",
    ylab = "Number of Tweets", bins = 9, binwidth = 0.5) + ggtitle("Overall Sentiment of Barack Obama's
    geom_bar(fill = "#232066") + theme_bw() + theme(panel.border = element_blank(),
    panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
    axis.line = element_line(colour = "black"), plot.title = element_text(hjust = 0.5)) +
    scale_x_continuous(expand = c(0, 0), limits = c(-8, 8)) +
    scale_y_continuous(expand = c(0, 0), limits = c(0, 1500))

trump_sent_plot <- qplot(trump_sent$score, xlab = "Sentiment Score",
    ylab = "Number of Tweets", bins = 13, binwidth = 0.5) + ggtitle("Overall Sentiment of Donald Trumps
    geom_bar(fill = "#E91D0E") + theme_bw() + theme(panel.border = element_blank(),
    panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
    axis.line = element_line(colour = "black"), plot.title = element_text(hjust = 0.5)) +
    scale_x_continuous(expand = c(0, 0), limits = c(-8, 8)) +
    scale_y_continuous(expand = c(0, 0), limits = c(0, 1500))
```

```
obama_sent_plot
```



```
trump_sent_plot
```



I put both graphs on the same x-axis scale, despite Obama's range being more restricted than Trumps. This makes it a lot easier to compare and see the differences between the kurtosis of the two distributions.

The immediately interesting thing to notice is that I was wrong in my initial hypothesis that Trump would express more negative sentiment overall than Obama. Trump is simply more emotional than Obama. In fact, 46% of Obama's tweets contain a completely neutral sentiment - no emotions at all. By Comparison, only 28% of Trumps tweets contain neutral sentiment - the rest contain some degree of positive or negative sentiment.

So, Trump is just more emotional than Obama on twitter - both positive and negative. Additionally, while they may only be outliers, Trump sometimes expresses more extremely emotional sentiment, with his tweets ranging from -7 to 6, and Obama's ranging only from -3 to 6.

Looking quickly at the descriptives, we can see that Obama's sentiment distribution is more strongly peaked than Trumps:

```
psych::describe(obama_sent$score)
```

```
##    vars    n mean   sd median trimmed  mad min max range skew kurtosis
## X1    1 3196 0.44 0.98      0    0.41 1.48  -3   6     9 0.37     1.12
##      se
## X1 0.02
```

```
psych::describe(trump_sent$score)
```

```
##    vars    n mean   sd median trimmed  mad min max range  skew kurtosis
## X1    1 3195 0.42 1.59      0    0.45 1.48  -7   6    13 -0.25     0.79
##      se
## X1 0.03
```

So, they both Obama and Trump are centered on being overall slightly positive in their tweets, but Trump's distribution is a bit flatter - he expresses more emotion overall.

Final Thoughts

So this was just a quick frequency count and sentiment analysis of Barack Obama's and Donald Trump's most recent 3,200 tweets. It seems like the content of what they talk about most often is subjectively different, just based on the types of words they each use often. The sentiment analysis shows that Donald Trump is overall more emotional than Barack Obama (regardless of the valence of the emotions).

One other tidbit not reported in the above analyses is the number of likes and retweets, as well as frequency of their tweeting. For one, Trump gets substantially more retweets and likes - I'm presuming this is because he gets a lot of people on the opposing side who "hate-retweet" him; it is also possible there are bots that frequently like and retweet trump. Donald Trump tweets on average three times a day, while Obama tweets on average once every two days.

Lastly, I recognize this is not the most fair comparison in terms of timeline etc. The analyses are confounded with the fact that Trump is currently the sitting President, while Obama is not. This could likely change the content of both of their tweets. Ideally, I would have liked to compare Trump's first year in office with Obama's first year in office, but I could not specify the data from twitter in this way. Additionally, that analysis would bring on other confounds (e.g., history) - so, six in one.

One open question that remains is regarding the extension of these differences; is this a Trump versus Obama thing, or is this a Republican versus Democrat thing? That remains to be seen, so stay tuned. . . .