STAT 231: Problem Set 6B

Brandon Kwon

due by 10 PM on Friday, April 2

This homework assignment is designed to help you further ingest, practice, and expand upon the material covered in class over the past week(s). You are encouraged to work with other students, but all code and text must be written by you, and you must indicate below who you discussed the assignment with (if anyone).

Steps to proceed:

1. In RStudio, go to File > Open Project, navigate to the folder with the course-content repo, select the course-content project (course-content.Rproj), and click "Open"
2. Pull the course-content repo (e.g. using the blue-ish down arrow in the Git tab in upper right window)
3. Copy ps6B.Rmd from the course repo to your repo (see page 6 of the GitHub Classroom Guide for Stat231 if needed)
4. Close the course-content repo project in RStudio
5. Open YOUR repo project in RStudio
6. In the ps6B.Rmd file in YOUR repo, replace "YOUR NAME HERE" with your name
7. Add in your responses, committing and pushing to YOUR repo in appropriate places along the way
8. Run "Knit PDF"
9. Upload the pdf to Gradescope. Don't forget to select which of your pages are associated with each problem. *You will not get credit for work on unassigned pages (e.g., if you only selected the first page but your solution spans two pages, you would lose points for any part on the second page that the grader can't see).*

If you discussed this assignment with any of your peers, please list who here:

ANSWER: Clara Seo, Ayo Lewis, Alastair Poole

Trump Tweets

David Robinson, Chief Data Scientist at DataCamp, wrote a blog post "Text analysis of Trump's tweets confirms he writes only the (angrier) Android half".

He provides a dataset with over 1,500 tweets from the account realDonaldTrump between 12/14/2015 and 8/8/2016. We'll use this dataset to explore the tweeting behavior of realDonaldTrump during this time period.

First, read in the file. Note that there is a `TwitteR` package which provides an interface to the Twitter web API. We'll use this R dataset David created using that package so that you don't have to set up Twitter authentication.

```
load(url("http://varianceexplained.org/files/trump_tweets_df.rda"))
```

A little wrangling to warm-up

1a. There are a number of variables in the dataset we won't need.

- First, confirm that all the observations in the dataset are from the screen-name `realDonaldTrump`.
- Then, create a new dataset called `tweets` that only includes the following variables:
- `text`
- `created`
- `statusSource`

```
# This verifies that all observations are from "realDonaldTrump"

trump_tweets_df %>%
  filter(screenName != "realDonaldTrump") %>%
  count
```

```
## # A tibble: 1 x 1
##       n
##   <int>
## 1     0
```

```
# Creating new dataset
tweets <- trump_tweets_df %>%
  select(c(text, created, statusSource))
```

1b. Using the `statusSource` variable, compute the number of tweets from each source. How many different sources are there? How often are each used?

> ANSWER: There are five different sources. Each source was used, once, 120 times, once, 762 times, and 628 times, respectively.

```
source_number <- tweets %>%
  group_by(statusSource) %>%
  summarise(num = n())
source_number
```

```
## # A tibble: 5 x 2
##   statusSource                                                          num
## * <chr>                                                               <int>
## 1 "<a href=\"http://instagram.com\" rel=\"nofollow\">Instagram</a>"        1
## 2 "<a href=\"http://twitter.com\" rel=\"nofollow\">Twitter Web Client</a>"  120
## 3 "<a href=\"http://twitter.com/#!/download/ipad\" rel=\"nofollow\">Twitt~    1
## 4 "<a href=\"http://twitter.com/download/android\" rel=\"nofollow\">Twitt~  762
## 5 "<a href=\"http://twitter.com/download/iphone\" rel=\"nofollow\">Twitte~  628
```

1c. We're going to compare the language used between the Android and iPhone sources, so only want to keep tweets coming from those sources. Explain what the `extract` function (from the `tidyverse` package) is doing below. Include in your own words what each argument is doing. (Note that "regex" stands for "regular expression".)

> ANSWER: The extract function basically looks at the observations from the statusSource column and from that, creates a new column called "source." Then it scans the observations from the column statusSource for the expression "Twitter for" (This could include N/A if nothing follows the expression.). Then, the "function looks for rows that only contain the strings"Android" or "iPhone" following "Twitter for." The "col = statusSource" argument states that the extract function will be scanning observations in the column "statusSource." The "into = 'source' " argument indicates that the desired observations are put into a new column called "source." The "regex = 'Twitter for (.*)<'" argument indiciates that the extract function will not get rid of the original statusSource column after extraction. Finally, the "filter" argument indicates that we are only selecting rows that contain "Android" or "iPhone" and placing these into the new column.

```
tweets2 <- tweets %>%
  extract(col = statusSource, into = "source"
          , regex = "Twitter for (.*)<"
          , remove = FALSE) %>%
  filter(source %in% c("Android", "iPhone"))
```

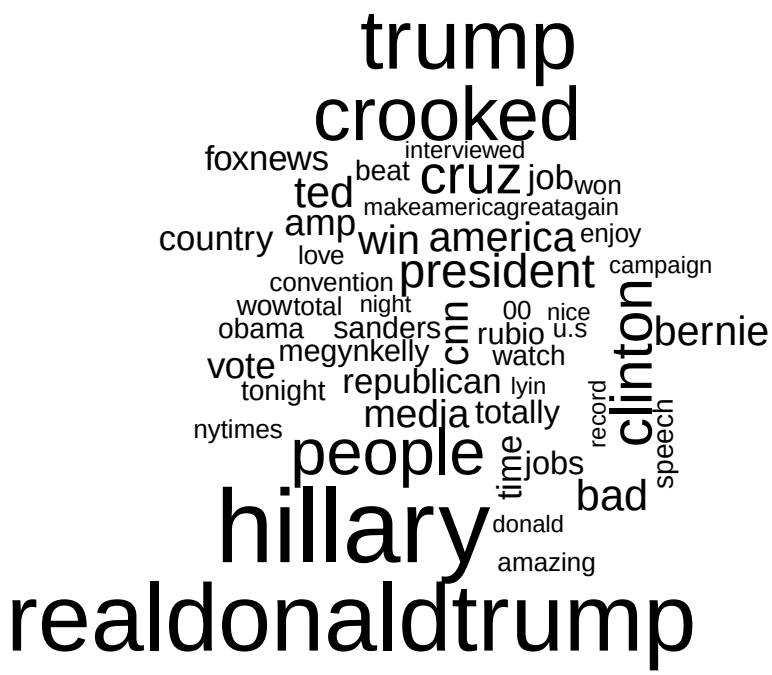
How does the language of the tweets differ by source?

2a. Create a word cloud for the top 50 words used in tweets sent from the Android. Create a second word cloud for the top 50 words used in tweets sent from the iPhone. How do these word clouds compare? (Are there some common words frequently used from both sources? Are the most common words different between the sources?)

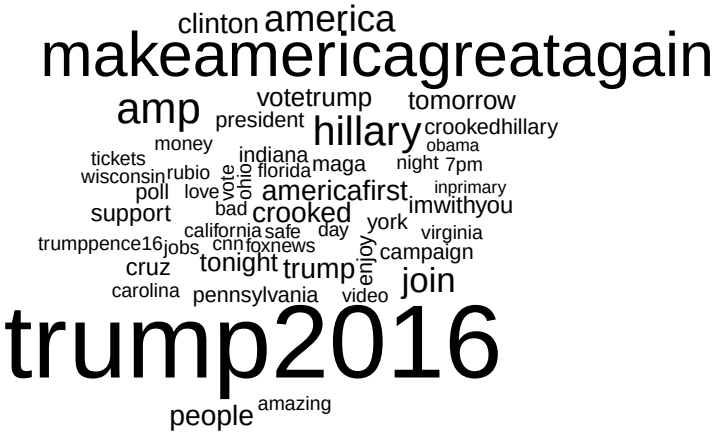
Don't forget to remove stop words before creating the word cloud. Also remove the terms "https" and "t.co".

> ANSWER: In the Android word cloud, I see that Trump uses words that more or less put down others/his opponents. Therefore, the words are placed in a more negative connotation. However, in the iPhone word cloud, I see that Trump uses words that have a more positive connotation and thus can infer that he is writing tweets that are about himself more than his opponents/adversaries. Some commond words used from sources include "Hillary," "vote," and "America," however.

```
#First create dataset where each word results in a distinct observation
tweets_words <- tweets2 %>%
  unnest_tokens(output = word, input = text)

#Removes stop words and "https" and "t.co"
tweets_words2 <- tweets_words %>%
  anti_join(stop_words, by="word") %>%
  filter(word != "https") %>%
  filter(word != "t.co")

#Creates two datasets just for iPhone and Android tweets respectively
#Use count function to count the occurrence of each word
tweets_iPhone <- tweets_words2 %>%
  filter(source == "iPhone") %>%
  count(word, sort = TRUE)
tweets_Android <- tweets_words2 %>%
  filter(source == "Android") %>%
  count(word, sort = TRUE)

#Wordcloud for Android
tweets_Android %>%
  with(wordcloud(words = word, freq = n, max.words=50))
```




```
#Wordcloud for iPhone
tweets_iPhone %>%
  with(wordcloud(words = word, freq = n, max.words=50))
```


clinton
america
makeamericagreatagain
amp
votetrump
tomorrow
president
hillary
crookedhillary
money
obama
tickets
indiana
maga
night
7pm
wisconsin
rubio
ohio
vote
florida
inprimary
poll
love
americafirst
imwithyou
support
bad
crooked
york
california
safe
day
virginia
trumppence16
jobs
cnn
foxnews
enjoy
campaign
cruz
tonight
trump
join
carolina
pennsylvania
video
trump2016
people
amazing

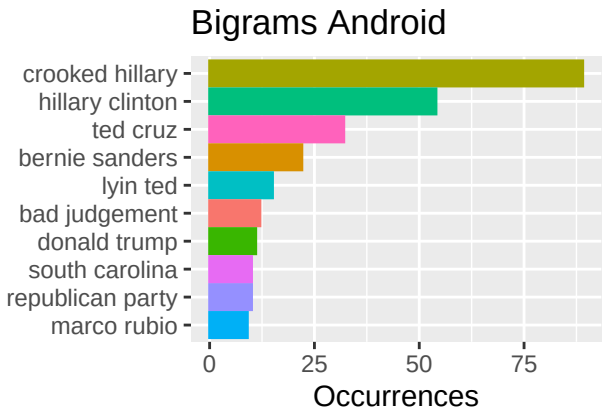
2b. Create a visualization that compares the top 10 *bigrams* appearing in tweets by each source (that is, facet by source). After creating a dataset with one row per bigram, you should remove any rows that contain a stop word within the bigram.

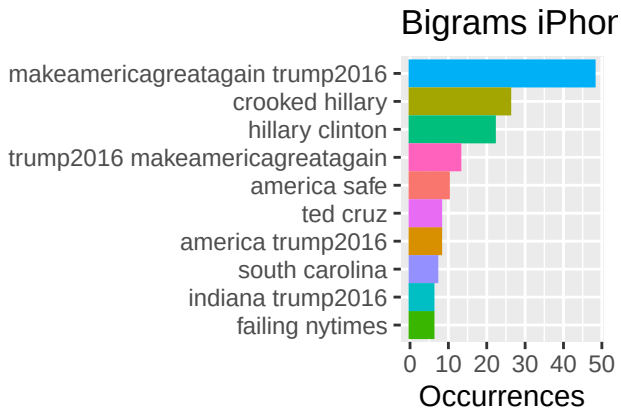
How do the top used bigrams compare between the two sources?

> ANSWER: When looking at the top used bigrams between the two sources, we see that while both sources contain negative bigrams that condescend Trump's opponents (including Hillary and Ted), we see that in the iPhone top used bigrams, there are more bigrams that elevate himself rather than knock down his opponents, including "makeamericagreatagain2016" and "trump2016."

```
# Create Bigram for Androids
Android_bigrams <- tweets2 %>%
  #Creates dataset with top-10
  filter(source == "Android") %>%
  unnest_tokens(output = bigram, input = text, token = "ngrams", n = 2) %>%
  separate(bigram, into = c("first", "second"), sep = " ", remove = FALSE) %>%
  anti_join(stop_words, by = c("first" = "word")) %>%
  anti_join(stop_words, by = c("second" = "word")) %>%
  filter(str_detect(first, "[a-z]") & str_detect(second, "[a-z]")) %>%
  filter(!str_detect(first, 'https') & !str_detect(second, 'https')) %>%
   filter(!str_detect(first, 't.co') & !str_detect(second, 't.co')) %>%
  count(bigram, sort = TRUE) %>%
  slice(1:10)

# Create Bigram for iPhones
iPhone_bigrams <- tweets2 %>%
  #Creates dataset with Top 10
  filter(source == "iPhone") %>%
  unnest_tokens(output = bigram, input = text, token = "ngrams", n = 2) %>%
  separate(bigram, into = c("first", "second"), sep = " ", remove = FALSE) %>%
  anti_join(stop_words, by = c("first" = "word")) %>%
  anti_join(stop_words, by = c("second" = "word")) %>%
  filter(str_detect(first, "[a-z]") & str_detect(second, "[a-z]")) %>%
  filter(!str_detect(first, 'https') & !str_detect(second, 'https')) %>%
   filter(!str_detect(first, 't.co') & !str_detect(second, 't.co')) %>%
  count(bigram, sort = TRUE) %>%
  slice(1:10)


visual1 <- Android_bigrams %>%
  # Creates Visualization
  ggplot(aes(x = reorder(bigram,n), y = n, color = bigram, fill=bigram)) +
  geom_col() +
  xlab(NULL) +
  coord_flip() +
  labs(y = "Occurrences"
       , title="Bigrams Android") +
  guides(color = "none", fill = "none")

visual2 <- iPhone_bigrams %>%
  # Creates Visualization
  ggplot(aes(x = reorder(bigram,n), y = n, color = bigram, fill=bigram)) +
  geom_col() +
```

```
  xlab(NULL) +
  coord_flip() +
  labs(y = "Occurrences"
       , title="Bigrams iPhone") +
  guides(color = "none", fill = "none")

#Places Plots Next to Each Other
ggarrange(visual1, visual2, ncol = 2, nrow = 2)
```





2c. Consider the sentiment. Compute the proportion of words among the tweets within each source classified as "angry" and the proportion of words classified as "joy" based on the NRC lexicon. How does the proportion of "angry" and "joy" words compare between the two sources? What about "positive" and "negative" words?

> ANSWER: The proportions of "angry" and "joy" words in iPhone are generally less than those of "angry" and "joy" words in Android. Moreover, the proportions of "positive" and "negative" words in iPhone are also less than those of "positive" and "negative" words in Android.

```
nrc_lexicon <- get_sentiments("nrc")

n_android <- sum(tweets_Android$n)

n_iPhone <- sum(tweets_iPhone$n)


#Creates Android dataset
Android_sentiment <-
  merge(tweets_Android, nrc_lexicon, by.x = "word", by.y = "word") %>%
  filter(sentiment == "joy" | sentiment == "anger" |
           sentiment == "positive" | sentiment == "negative") %>%
  #Finds total number of words associated with particular sentiment
  group_by(sentiment) %>%
  summarize(total = sum(n)) %>%
  mutate(proportion = total / n_android)

print(Android_sentiment)
```

```
## # A tibble: 4 x 3
##   sentiment total proportion
## * <chr>     <int>      <dbl>
## 1 anger       363     0.0522
## 2 joy         267     0.0384
## 3 negative    647     0.0930
## 4 positive    734     0.105
```

```
#Creates iPhone dataset
iPhone_sentiment <-
  merge(tweets_iPhone, nrc_lexicon, by.x = "word", by.y = "word") %>%
  filter(sentiment == "joy" | sentiment == "anger" |
           sentiment == "positive" | sentiment == "negative") %>%
  #Finds total number of words associated with particular sentiment
  group_by(sentiment) %>%
  summarize(total = sum(n)) %>%
  mutate(proportion = total / n_android)

print(Android_sentiment)
```

```
## # A tibble: 4 x 3
##   sentiment total proportion
## * <chr>     <int>      <dbl>
## 1 anger       363     0.0522
## 2 joy         267     0.0384
## 3 negative    647     0.0930
## 4 positive    734     0.105
```

2d. Lastly, based on your responses above, do you think there is evidence to support Robinson's claim that Trump only writes the (angrier) Android half of the tweets from realDonaldTrump? In 2-4 sentences, please explain.

ANSWER: There is evidence to support Robinson's claim that Trump only writes the angrier Android half of the tweets from realDonaldTrump. We see that the proportions of "angry" and "joy" words in Android are generally more than those of "angry" and "joy" words in Android. When looking at the top used bigrams between the two sources, we see that while both sources contain negative bigrams that condescend Trump's opponents (including Hillary and Ted), we see that in the iPhone top used bigrams, there are more bigrams that elevate himself rather than knock down his opponents, including "makeamericagreatagain2016" and "trump2016." According to wordclouds, we also see that words like "crooked" and "bad" make up a good majority of most of the words in the Android wordcloud.