STAT 231: Problem Set 6B

Evan Daisy

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: Conrad Kuklinsky

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`

```
tweets <- trump_tweets_df %>%
  filter(screenName == "realDonaldTrump") %>%
  select(text, created, statusSource)
```

There are still 1512 observations, so I guess all of these tweets were created under screen name realDonaldTrump.

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 used. Instagram is used once, Twitter on a computer 120 times, Twitter on an ipad once, Twitter on an android 762 times, and Twitter on an iphone 628 times.

```
tweets_sum <- tweets %>%
  count(statusSource)
```

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 here is creating a new variable based on the statusSource column (the column the col argument is telling it to focus on). The into argument tells it to call this new variable "source", and the regex argument tells it what it should take from the input variable as the value of the output variable: in this case whatever the period is replacing in the given regular expression. Finally, the remove argument tells R not to delete the input column after creating the output one.

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

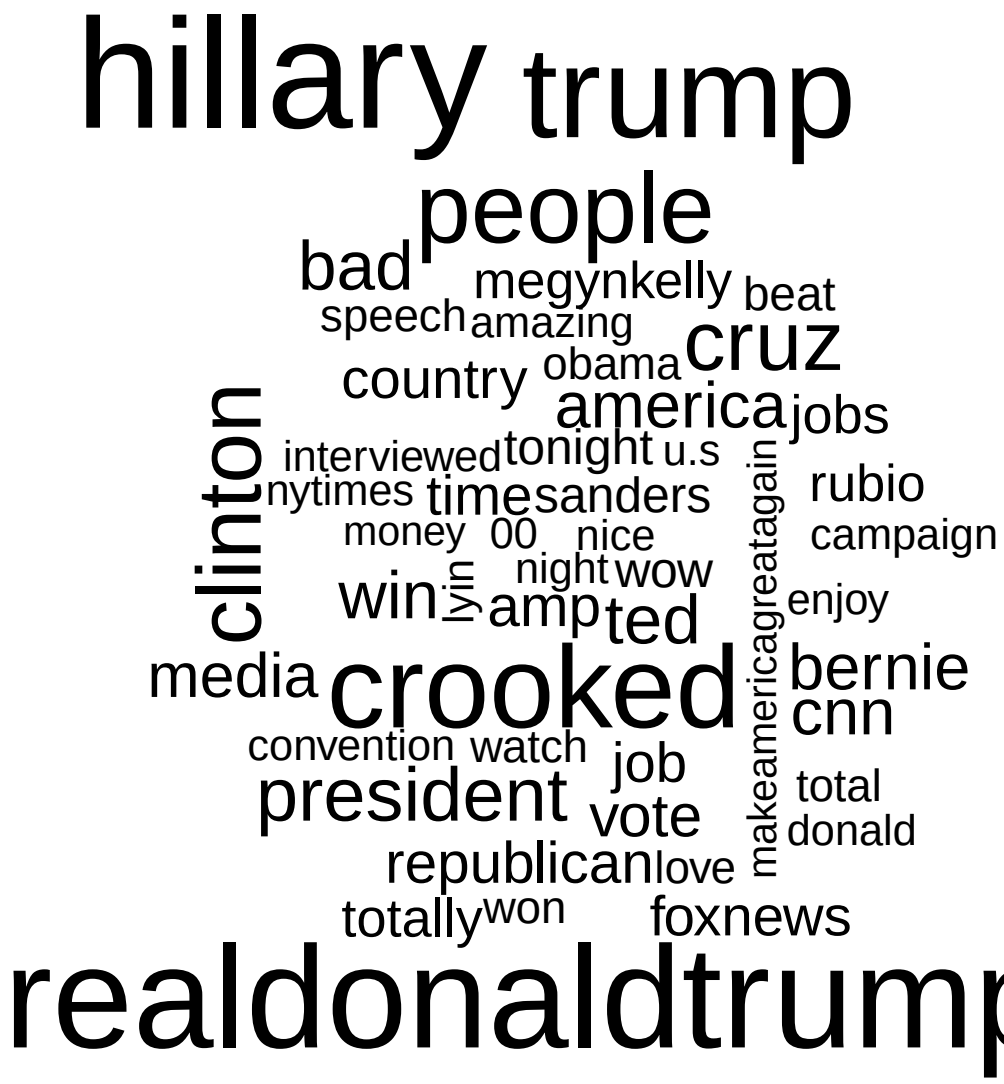
How does the language of the tweets differ by source?

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: Hillary, Cruz, and people are relatively big in both word clouds, but the biggest Android words are trump, realdonaldtrump, crooked, and Hillary, whereas the biggest Iphone words are his campaign slogans (trump2016 and makeamericagreatagain).

```
tweet_words <- tweets2 %>%
  unnest_tokens(output = word, input = text) %>%
  anti_join(stop_words, by="word") %>%
  filter(!(word %in% c("https", "t.co")))
android_words <- tweet_words %>%
  filter(source == "Android") %>%
  count(word, sort = TRUE)
iphone_words <- tweet_words %>%
  filter(source == "iPhone") %>%
  count(word, sort = TRUE)
android_words %>%
  with(wordcloud(words = word, freq = n, max.words=50, scale = c(6,1)))
```




```
iphone_words %>%
  with(wordcloud(words = word, freq = n, max.words=50, scale = c(6,1)))
```

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

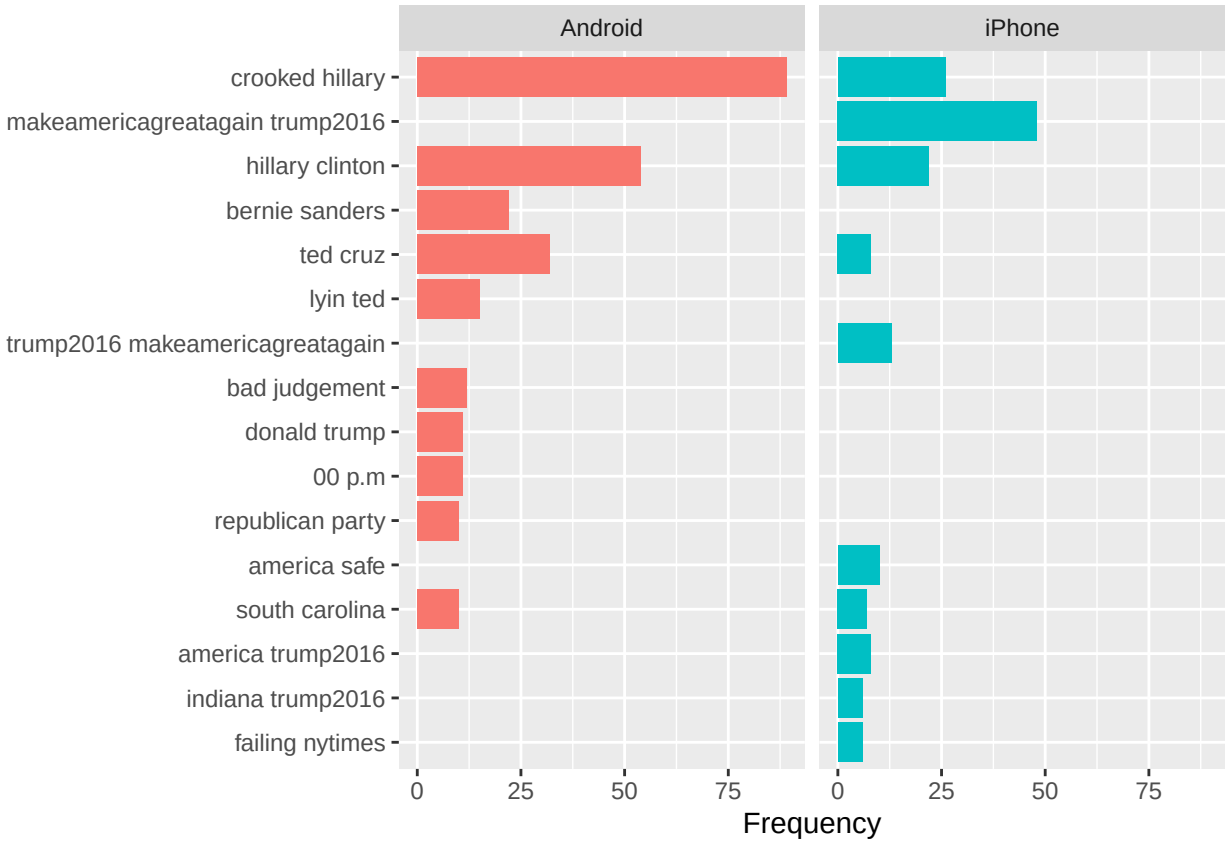
> ANSWER: Bigrams about political opponents ("crooked Hillary", "Ted Cruz", "Bernie Sanders", etc.) seem to come mostly from the Android, although Hillary makes quite an appearance in the iPhone tweets as well. Self-promotion in the form of versions of Trump's campaign slogan (trump2016 makeamericagreatagain) seem to come mostly from the iPhone

```
tweet_bigrams <- tweets2 %>%
  unnest_tokens(output = bigram, input = text, token = "ngrams", n = 2) %>%
  separate(bigram, into = c("word", "word_2"), sep = " ", remove = FALSE) %>%
  anti_join(stop_words, by = "word") %>%
  filter(!(word %in% c("https", "t.co"))) %>%
  separate(bigram, into = c("word_1", "word"), sep = " ", remove = FALSE) %>%
  anti_join(stop_words, by = "word") %>%
  filter(!(word %in% c("https", "t.co")))
top_bigrams <- tweet_bigrams %>%
  group_by(bigram, source) %>%
  summarise(frequency = n()) %>%
  arrange(desc(frequency)) %>%
  group_by(source) %>%
  slice(1:10) %>%
  ungroup()
```

```
## `summarise()` has grouped output by 'bigram'. You can override using the `.groups` argument.
```

```
#android_bigrams <- tweet_bigrams %>%
  #filter(source == "Android") %>%
  #group_by(bigram) %>%
  #summarise(frequency = n()) %>%
  #arrange(desc(frequency))
#iphone_bigrams <- tweet_bigrams %>%
  #filter(source == "iPhone") %>%
  #group_by(bigram) %>%
  #summarise(frequency = n()) %>%
  #arrange(desc(frequency))
```

```
ggplot(top_bigrams, aes(x = reorder(bigram, frequency),
                        y = frequency,
                        fill = source)) +
  geom_col(show.legend = FALSE) +
  labs(x = NULL, y = "Frequency") +
  facet_wrap(~source) +
  coord_flip()
```


Android
iPhone
crooked hillary
makeamericagreatagain trump2016
hillary clinton
bernie sanders
ted cruz
lyin ted
trump2016 makeamericagreatagain
bad judgement
donald trump
00 p.m
republican party
america safe
south carolina
america trump2016
indiana trump2016
failing nytimes
0
25
50
75
0
25
50
75
Frequency

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: Of the 539 Android words categorized by the NRC lexicon, about 20% of them were angry whereas about 14% of them were joyful. Of the 356 iPhone words categorized by the NRC lexicon, about 23% of them were angry whereas about 17% of them were joyful. 42% of the Android words were positive whereas 44% were negative, and 41% of the iPhone words were positive while 47% of them were negative.

```
nrc_lexicon <- get_sentiments("nrc")
nrc_missed <- android_words %>%
  anti_join(nrc_lexicon, by = "word")
2445-1906
```

```
## [1] 539
```

```
nrc_missed_i <- iphone_words %>%
  anti_join(nrc_lexicon, by = "word")
2163-1807
```

```
## [1] 356
```

```
android_moods <- android_words %>%
  inner_join(nrc_lexicon, by = "word") %>%
  filter(sentiment %in% c("anger", "joy", "positive", "negative")) %>%
  count(sentiment)
iphone_moods <- iphone_words %>%
  inner_join(nrc_lexicon, by = "word") %>%
  filter(sentiment %in% c("anger", "joy", "positive", "negative")) %>%
  count(sentiment)
106/539
```

```
## [1] 0.1966605
```

```
77/539
```

```
## [1] 0.1428571
```

```
82/356
```

```
## [1] 0.2303371
```

```
59/356
```

```
## [1] 0.1657303
```

```
227/539
```

```
## [1] 0.4211503
```

```
238/539
```

```
## [1] 0.4415584
```

```
147/356
```

```
## [1] 0.4129213
```

```
166/356
```

```
## [1] 0.4662921
```

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: If we simply look at word and bigram frequencies, there is certainly evidence to support Robinson's claim. The most frequently used words and bigrams from the Android mostly involve "crooked" political opponents, whereas those from the iPhone are more self-promoting, with the most frequent bigram being Trump's campaign slogan. If we look at the sentiments however, the nrc lexicon actually attributes negative or angry sentiments to a greater proportion of iPhone words than Android words. Nonetheless, there are distinct differences in word and bigram usage that could suggest that a different person was tweeting from the Android than was tweeting from the iPhone.