STAT 231: Problem Set 6B

Lorraine Oloo

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: Lovemore, Teddy

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`

```
confirm <- subset(trump_tweets_df, screenName == "realDonaldTrump")
#has the same number of observations as the original dataset

#creating new dataset
tweets <- select(trump_tweets_df, 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 5 different sources. Instagram with 1 tweet, Twitter Web Client with 120 tweets, Twitter for iPad with 1 tweet, Twitter for Android with 762 tweets, and Twitter for iPhone with 628 tweets.

```
tweets%>%
group_by(statusSource)%>%
  summarise(n = n())
```

```
## # A tibble: 5 x 2
##   statusSource                                                          n
## * <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: Using the contents from the column statusSource, col argument creates a new column called source. The regex argument checks for the characters between the texts "Twitter for" and "<" in the statusSource, and records those characters. The remove argument removes any other character that is not in the regex.

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

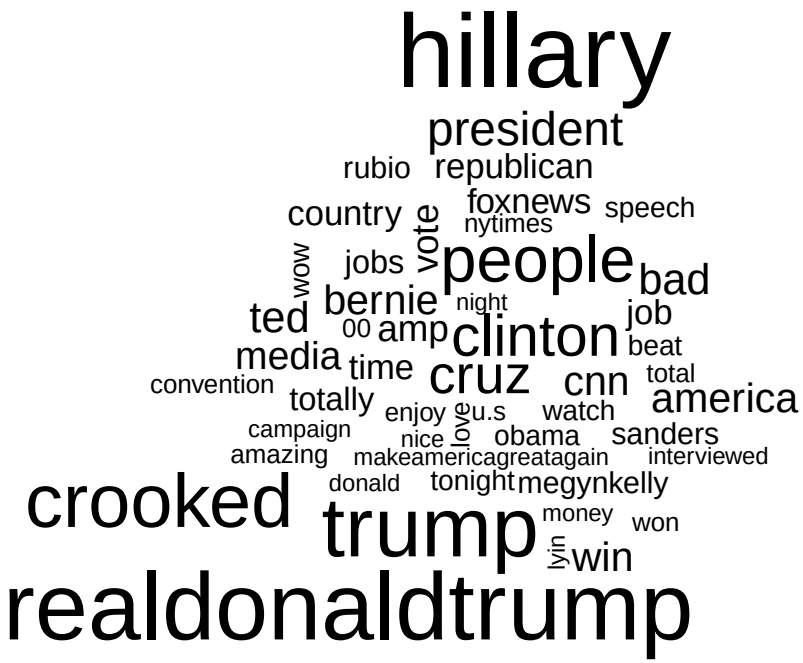
How does the language of the tweets differ by source?

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:
> For Android users, the first three words were:
> hillary 135
> realdonaldtrump 123
> trump 107
> For iPhone users, the first three words were:
> trump2016 171
> makeamericagreatagain 95
> amp 56
> There are some words that are commonly used by the two different sources. Examples are words such as clinton, foxnews, and jobs.

```
#word count for android users
tweetAndroid <- tweets2 %>%
  filter(source == "Android")

tweet_Android <- tweetAndroid %>%
  unnest_tokens(output = word, input = text) %>%
  anti_join(stop_words, by="word") %>%
  filter(word != "https" ) %>%
  filter(word != "t.co" ) %>%
  count(word, sort = TRUE)

tweet_Android %>%
  with(wordcloud(words = word, freq = n, max.words=50))
```

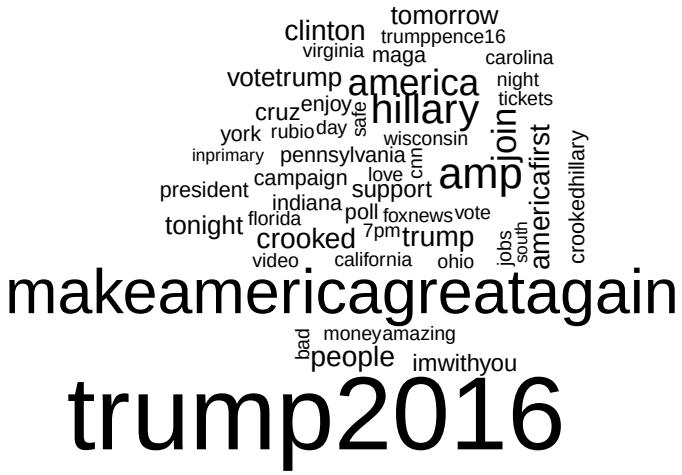



```
head(tweet_Android, 50)
```

```
## # A tibble: 50 x 2
##    word               n
##    <chr>          <int>
##  1 hillary          135
##  2 realdonaldtrump  123
##  3 trump            107
##  4 crooked           94
##  5 people            76
##  6 clinton           68
##  7 cruz              60
##  8 president         52
```

```
##  9 bad                44
## 10 ted                44
## # ... with 40 more rows
```

```
# word count for iPhone users
tweetiPhone <- tweets2 %>%
  filter(source == "iPhone")

tweet_iPhone <- tweetiPhone %>%
  unnest_tokens(output = word, input = text) %>%
  anti_join(stop_words, by="word") %>%
  filter(word != "https" ) %>%
  filter(word != "t.co" ) %>%
  count(word, sort = TRUE)

tweet_iPhone %>%
  with(wordcloud(words = word, freq = n, max.words=50))
```




```
head(tweet_iPhone, 50)
```

```
## # A tibble: 50 x 2
##    word                    n
##    <chr>               <int>
##  1 trump2016             171
##  2 makeamericagreatagain  95
##  3 amp                    56
##  4 hillary                51
##  5 america                43
##  6 join                   42
##  7 clinton                30
##  8 crooked                28
```

```
##  9 people                   28
## 10 americafirst             27
## # ... with 40 more rows
```

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.

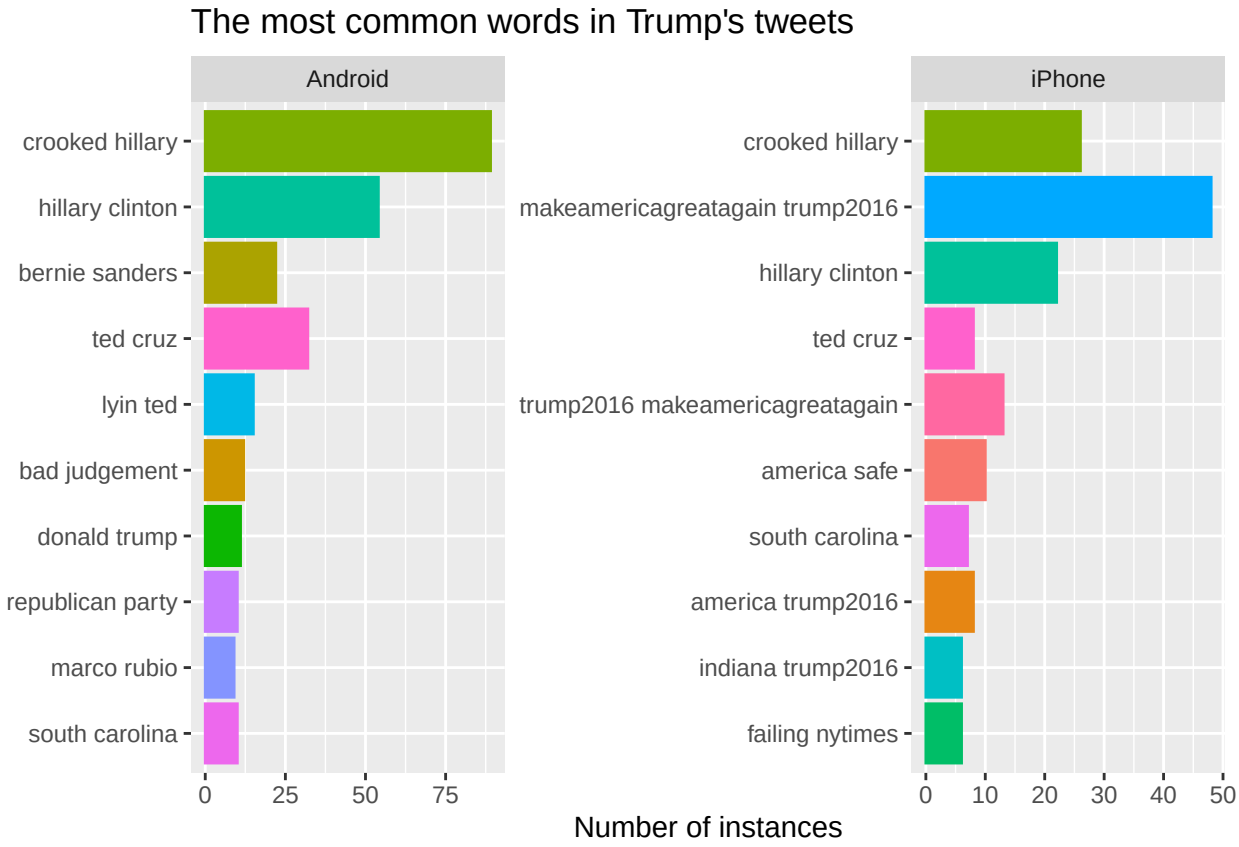
How do the top used bigrams compare between the two sources?

ANSWER:

```
nrc0<- get_sentiments("nrc")

sentiments  <- c("anger", "joy", "positive", "negative")


tweet_ngrams <- tweets2 %>%
  unnest_tokens(output = word, input = text
                , token = "ngrams", n = 2) %>%

separate(word, into = c("one", "two"), remove = FALSE)
```

```
## Warning: Expected 2 pieces. Additional pieces discarded in 2030 rows [11, 12,
## 27, 28, 35, 36, 44, 45, 113, 114, 115, 124, 125, 159, 160, 239, 240, 246, 247,
## 254, ...].
```

```
tweet_ngrams2 <- tweet_ngrams %>%
  anti_join(stop_words, by = c("one" = "word")) %>%
  anti_join(stop_words, by = c("two" = "word")) %>%
  filter(one != "https" & two != "https" ) %>%
  filter(one != "t.co" & two != "t.co" ) %>%
  group_by(source) %>%
  count(word, sort = TRUE)

#Creating the actual visualization by source
tweet_ngrams2 %>%
  slice(1:10) %>%
  ggplot(aes(x = reorder(word,n), y = n, color = word, fill=word)) +
  geom_col() +
  xlab(NULL) +
  coord_flip() +
  labs(y = "Number of instances"
       , title="The most common words in Trump's tweets") +
  guides(color = "none", fill = "none") +
  facet_wrap(~source, scales = "free")
```


The most common words in Trump's tweets
Android
crooked hillary
hillary clinton
bernie sanders
ted cruz
lyin ted
bad judgement
donald trump
republican party
marco rubio
south carolina
0
25
50
75
iPhone
crooked hillary
makeamericagreatagain trump2016
hillary clinton
ted cruz
trump2016 makeamericagreatagain
america safe
south carolina
america trump2016
indiana trump2016
failing nytimes
0
10
20
30
40
50
Number of instances

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 proportion of anger for android users is 0.01523426 and for joy is 0.01106640. The proportion of anger for iPhone users is 0.01780673 and for joy is 0.01281216. There is a higher proportion for both anger and joy sentiments in iPhone users.

The proportion of positive for android users is 0.03420523 and 0.03262432 for negative. The proportion of positive for iPhone users is 0.03604777 and 0.03192182 for negative. iPhone users had a higher proportions of positive, and the android users had a higher proportions for negative sentiments.

```
#Android sentiment
Android_Totalwords <- sum(tweet_Android$n)

Android_sentiment <- get_sentiments("nrc") %>%
  filter(sentiment %in% c("anger", "joy", "positive", "negative")) %>%
  inner_join(tweet_Android, by = "word") %>%
  anti_join(stop_words, by=c("sentiment" = "word"))%>%
  group_by(sentiment)%>%
  summarise(n=n())%>%
  mutate(proportions = n/Android_Totalwords)

Android_sentiment
```

```
## # A tibble: 4 x 3
##   sentiment     n proportions
## * <chr>     <int>       <dbl>
## 1 anger       106      0.0152
## 2 joy          77      0.0111
## 3 negative    227      0.0326
## 4 positive    238      0.0342
```

```
#iPhone sentiment
iPhone_Totalwords <- sum(tweet_iPhone$n)

iPhone_sentiment <- get_sentiments("nrc") %>%
  filter(sentiment %in% c("anger", "joy", "positive", "negative")) %>%
  inner_join(tweet_iPhone, by = "word") %>%
  anti_join(stop_words, by=c("sentiment" = "word"))%>%
  group_by(sentiment)%>%
  summarise(n=n())%>%
  mutate(proportions = n/iPhone_Totalwords)

iPhone_sentiment
```

```
## # A tibble: 4 x 3
##   sentiment     n proportions
## * <chr>     <int>       <dbl>
## 1 anger        82      0.0178
```

```
## 2 joy          59      0.0128
## 3 negative    147      0.0319
## 4 positive    166      0.0360
```

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: The number of angry tweets from android is 106, and the one from iPhone was 82. He wrote more of the angrier tweets from android.