STAT 231: Problem Set 6B

Mike Santos

due by 2 PM on Friday, October 9

This homework assignment is designed to help you futher 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:

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[0:1512, c(1,5,10)]
```

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 4 sources: Android, iPad, iPhone, and Computer. There are 762 tweets from Android, 628 from iPhone, 120 from a computer, and 1 from an iPad.

```
source = tweets
source$device = "Computer"
source = tweets %>% extract(col = statusSource, into = "device"
          , regex = "Twitter (.*)<"
          , remove = FALSE)
source %>% group_by(device) %>% tally()
```

```
## # A tibble: 5 x 2
##   device          n
##   <chr>       <int>
## 1 for Android   762
## 2 for iPad        1
## 3 for iPhone    628
## 4 Web Client    120
## 5 <NA>            1
```

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. (Note that "regex" stands for "regular expression".)

> ANSWER: The extract function is reading the text after the source says "Twitter for .." and reads in the device from which the tweet was written.

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

android = tweets2  %>%
  filter(source %in% "Android")

iphone = tweets2  %>%
  filter(source %in% "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: The most common phrase between both word clouds seems to be "hillary", which is certainly not what I expected. The iPhone word cloud seems to be more positive, including words such as "love", "safe", and "enjoy". The iPhone word cloud is a lot more dominated by two commonly used phrases, whereas the Android one has more less commonly used words.

```
library("tm")
```

```
## Loading required package: NLP
```

```
##
## Attaching package: 'NLP'
```

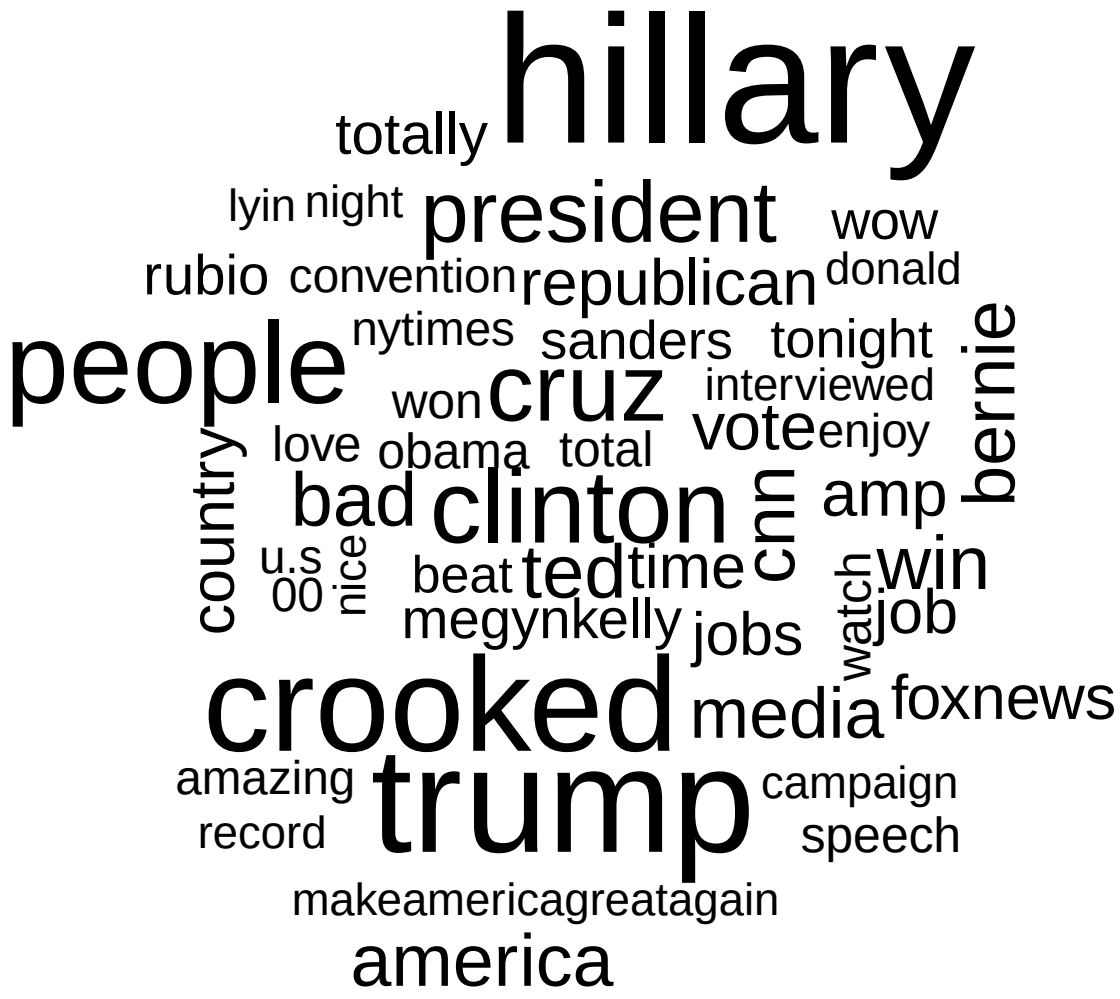
```
## The following object is masked from 'package:ggplot2':
##
##     annotate
```

```
library("SnowballC")
library("wordcloud")
library("RColorBrewer")
library(tidyverse)

#cleaning android words
android_words <- android %>%
  unnest_tokens(output = word, input = text)

https = list(word = "https", lexicon = "extra")
tco = list(word = "t.co", lexicon = "extra")

data(stop_words)
stop_words = stop_words %>% rbind(tco)%>% rbind(https)
stop_words %>% count(lexicon)
```

```
## # A tibble: 4 x 2
##   lexicon      n
##   <chr>    <int>
## 1 extra        2
## 2 onix       404
## 3 SMART      571
## 4 snowball   174
```

```r
android_clean <- android_words %>%
  anti_join(stop_words, by="word")

android_frequencies <- android %>%
  unnest_tokens(output = word, input = text) %>%
  anti_join(stop_words, by="word") %>%
  count(word, sort = TRUE)

#android word cloud
android_frequencies %>%
  with(wordcloud(words = word, freq = n, max.words=50, scale=c(7,1)))
```

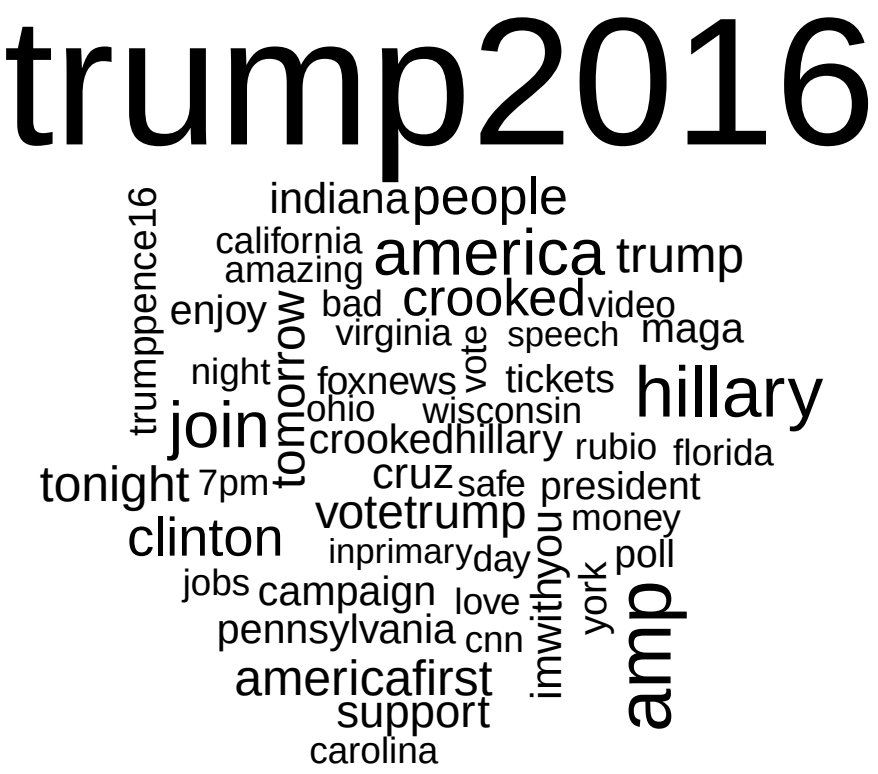
```
## Warning in wordcloud(words = word, freq = n, max.words = 50, scale = c(7, :
## realdonaldtrump could not be fit on page. It will not be plotted.
```




```
#cleaning iphone words
iphone_words <- iphone %>%
  unnest_tokens(output = word, input = text)

iphone_clean <- iphone_words %>%
  anti_join(stop_words, by="word")

iphone_frequencies <- iphone %>%
  unnest_tokens(output = word, input = text) %>%
  anti_join(stop_words, by="word") %>%
  count(word, sort = TRUE)

#creating iphone word cloud
iphone_frequencies %>%
```

```
  with(wordcloud(words = word, freq = n, max.words=50, scale=c(7,1)))
```

```
## Warning in wordcloud(words = word, freq = n, max.words = 50, scale = c(7, :
## makeamericagreatagain could not be fit on page. It will not be plotted.
```

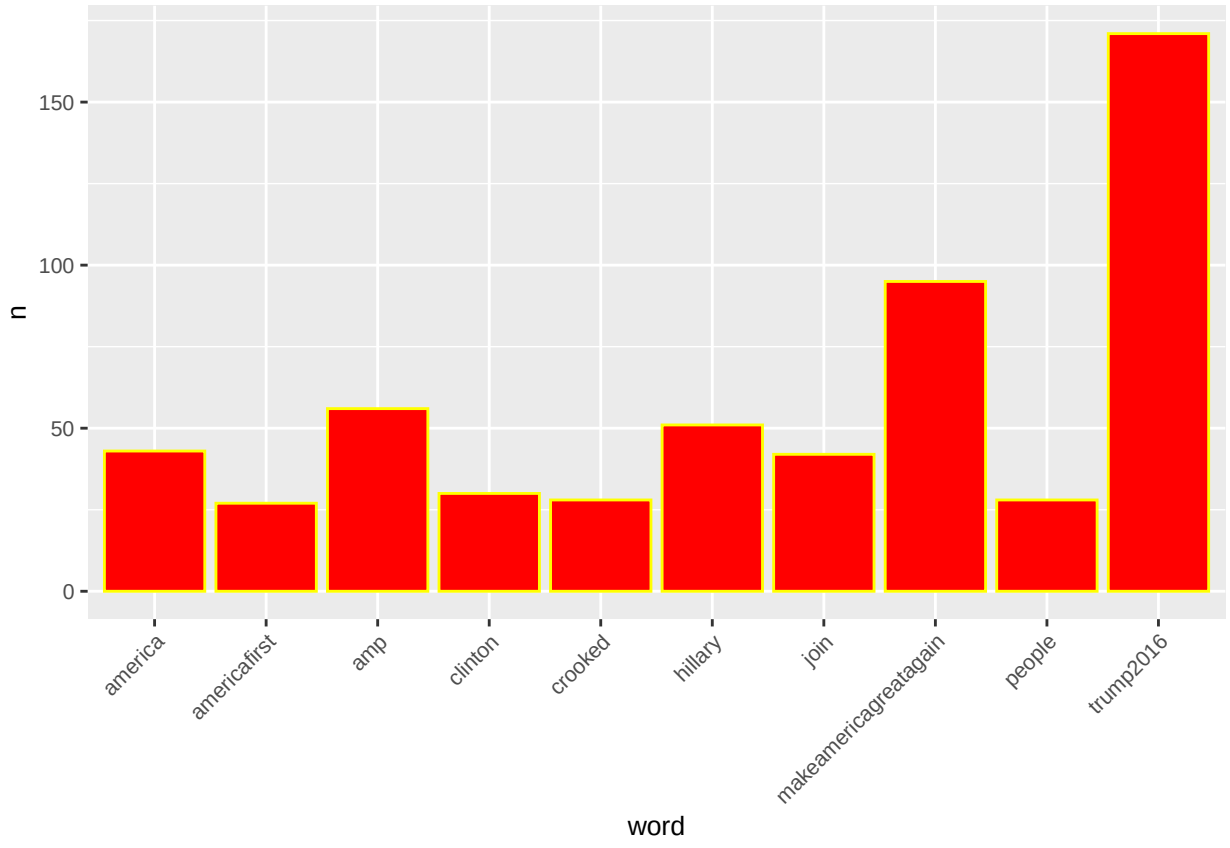


2b. Create a visualization that compares the top 10 *bigrams* appearing in tweets by each source (that is, facet by source). After creating a dataset with one row per bigram, you should remove any rows that contain a stop word within the bigram.

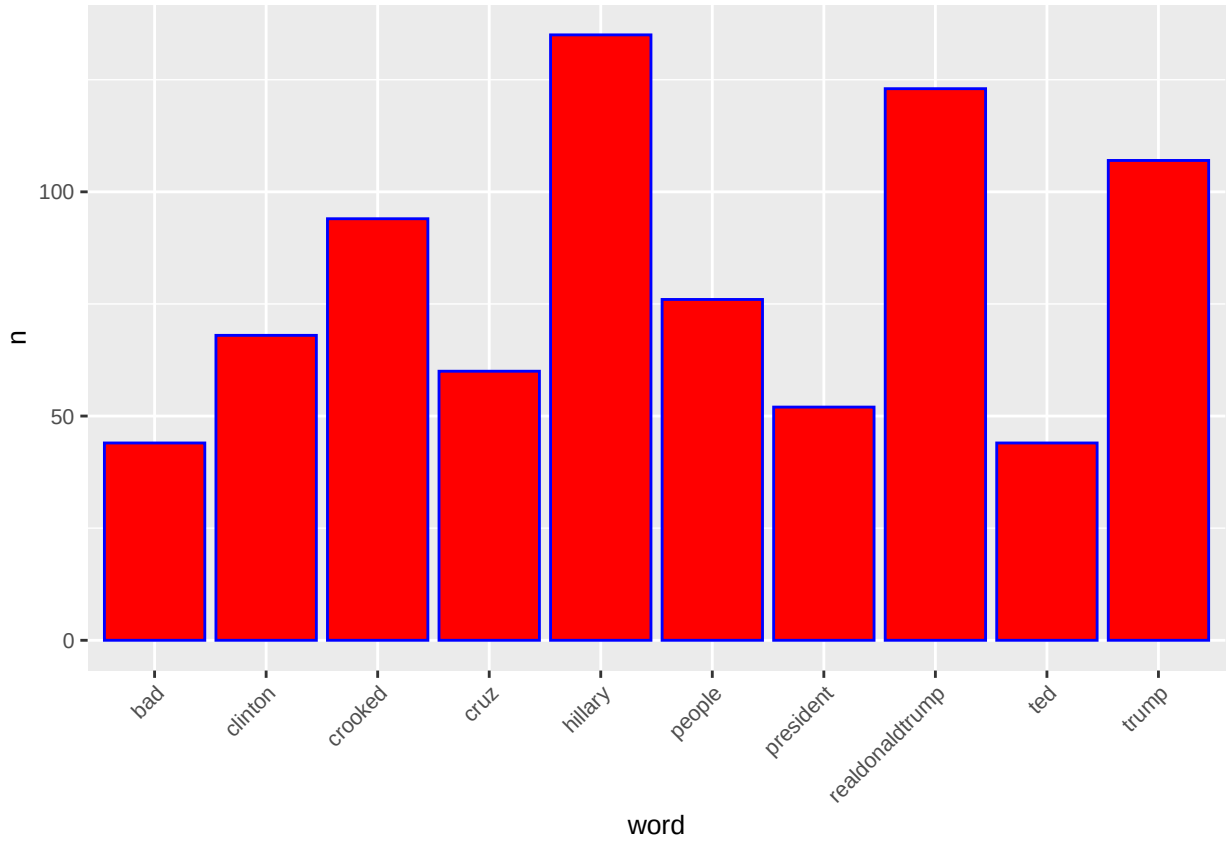
How do the top used bigrams compare between the two sources?

> ANSWER: The bigrams differ similarly to the word clouds, where the android display features words which are mentioned similar amounts of times, and the iphone display is dominated by a few words.

```
iphone_ten = head(iphone_frequencies, 10)
android_ten = head(android_frequencies, 10)

library(ggplot2)
iphone <-ggplot(data=iphone_ten, aes(x=word, y=n)) + geom_bar(color = "yellow", fill = "red", stat="ide
theme(text = element_text(size=10),
        axis.text.x = element_text(angle=45, hjust=1))
iphone
```




```
android <-ggplot(data=android_ten, aes(x=word, y=n)) +
geom_bar(color = "blue", fill = "red", stat="identity")+
theme(text = element_text(size=10),
        axis.text.x = element_text(angle=45, hjust=1))
android
```


100
50
0
n
bad
clinton
crooked
cruz
hillary
people
president
realdonaldtrump
ted
trump
word

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: Both devices have a roughly 1:1 positive to negative tweet ratio, and both have a roughly 2:1 positive to angry tweet ratio. It seems as if the proportions of angry to joy words are fairly similar. Proportionally the devices seem to tweet the same ratio of sentiments.

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

#adding nrc lexicon to android words
android_missed_words <- android_frequencies %>%
  anti_join(nrc_lexicon, by="word")

android_freq2 = android_frequencies %>% anti_join(android_missed_words, by="word")
android_freq2 = merge(android_freq2, nrc_lexicon, by="word")
android_count = android_freq2 %>% count(sentiment)
head(android_count, 10)
```

```
##       sentiment   n
## 1         anger 106
## 2  anticipation  94
## 3       disgust  78
## 4          fear 115
## 5           joy  77
## 6      negative 227
## 7      positive 238
## 8       sadness 108
## 9      surprise  54
## 10        trust 140
```

```
#adding nrc lexicon to iphone words
iphone_missed_words <- iphone_frequencies %>%
  anti_join(nrc_lexicon, by="word")

iphone_freq2 = iphone_frequencies %>% anti_join(iphone_missed_words, by="word")
iphone_freq2 = merge(iphone_freq2, nrc_lexicon, by="word")
iphone_count = iphone_freq2 %>% count(sentiment)
head(iphone_count, 10)
```

```
##       sentiment   n
## 1         anger  82
## 2  anticipation  67
## 3       disgust  47
## 4          fear  80
## 5           joy  59
## 6      negative 147
## 7      positive 166
## 8       sadness  77
## 9      surprise  40
## 10        trust  99
```

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: I would not say that there is enough evidence to support this claim. While at a glance, there are more negative tweets from the android half, there are not proportionally many more negative tweets. Both devices have a roughly 1:1 positive to negative tweet ratio, and both have a roughly 2:1 positive to angry tweet ratio. Proportionally the devices seem to tweet the same ratio of sentiments.