ETC3250 Lab 7

Di Cook

Week 7

Purpose

This lab will be on wrangling and plotting data.

Reading

- David Robinson's blog post
- Tidy data
- Split-apply-combine strategy: dplyr vignette, JSS paper

Trump's tweets

David Robinson wrote a post describing his analysis of the tweets coming from @realDonaldTrump. There was rumour that his staff also tweeted from this account. His analysis suggests that tweets come from two different phones and the sentiment from each is different. Let's take a look at the data:

```
library(dplyr)
library(purrr)
load(url("http://varianceexplained.org/files/trump_tweets_df.rda"))
library(tidyr)
```

Question 1

Explain what this code does.

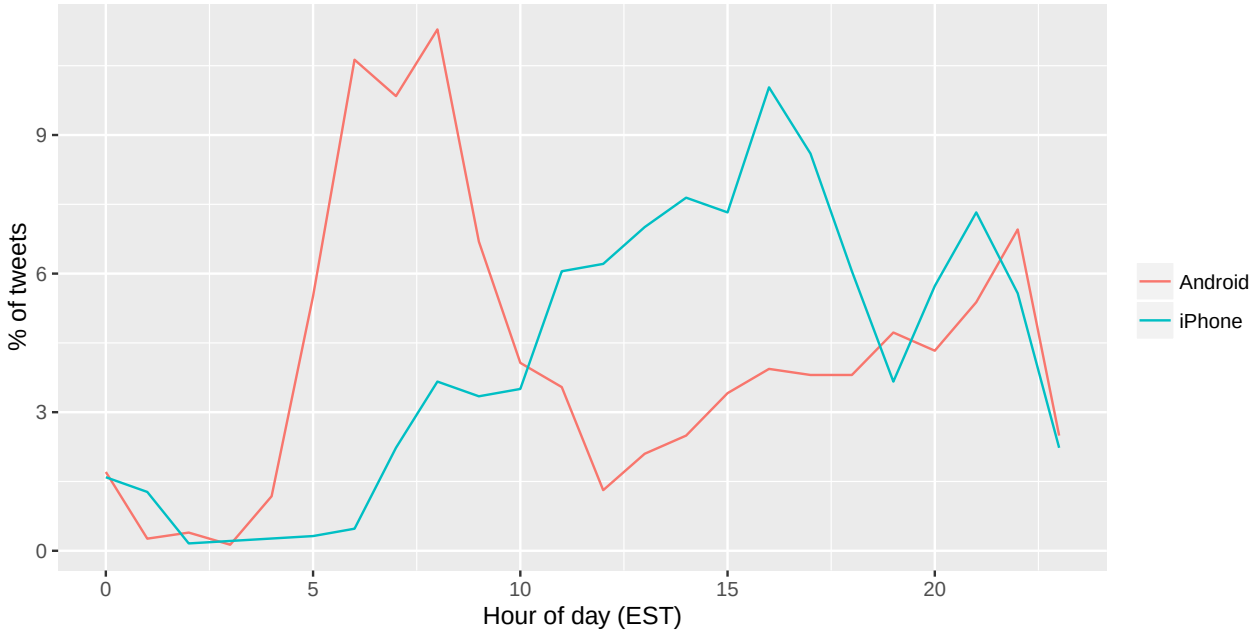
```
tweets <- trump_tweets_df %>%
  select(id, statusSource, text, created) %>%
  extract(statusSource, "source", "Twitter for (.*?)<") %>%
  filter(source %in% c("iPhone", "Android"))
```

Question 2

a. Explain what this code does.

```
library(lubridate)
library(scales)
library(ggplot2)

tweets %>%
  count(source, hour = hour(with_tz(created, "EST"))) %>%
  mutate(percent = n / sum(n)*100) %>%
  ggplot(aes(hour, percent, color = source)) +
```

```
  geom_line() +
  labs(x = "Hour of day (EST)",
       y = "% of tweets",
       color = "")
```



b. What do we learn from this plot?

Question 3

Is there a difference between the two devices in terms of including a picture?

Question 4

The following code does text analysis of the tweets, pulling the major words used in each teet.

a. How many tweet words are produced? How many unique words are there?

```
library(tidytext)

reg <- "([^A-Za-z\\d#@']|'(?![A-Za-z\\d#@]))"
tweet_words <- tweets %>%
  filter(!str_detect(text, '^"')) %>%
  mutate(text = str_replace_all(text, "https://t.co/[A-Za-z\\d]+|&amp;", "")) %>%
  unnest_tokens(word, text, token = "regex", pattern = reg) %>%
  filter(!word %in% stop_words$word,
         str_detect(word, "[a-z]"))
tweet_words
```

b. Write the code to make the plot in Figure 1, make the plot, and explain it.

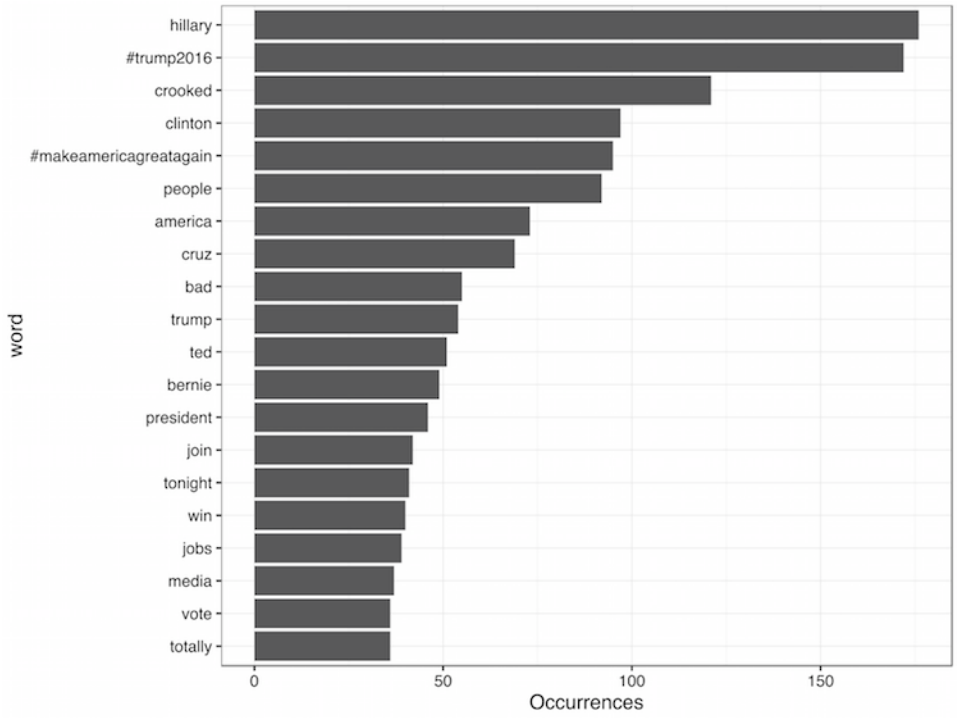


Figure 1: Top 20 words from tweets

Question 5

The following code compute the log odds ratio for the source of the word.

```
android_iphone_ratios <- tweet_words %>%
  count(word, source) %>%
  filter(sum(n) >= 5) %>%
  spread(source, n, fill = 0) %>%
  ungroup() %>%
  mutate_each(funs((. + 1) / sum(. + 1)), -word) %>%
  mutate(logratio = log2(Android / iPhone)) %>%
  arrange(desc(logratio))
```

a. Write the code to produce the plot in Figure 2.

b. What do you learn from the plot?

Question 6

The following code tags a word with a sentiment.

```
nrc <- sentiments %>%
  filter(lexicon == "nrc") %>%
  dplyr::select(word, sentiment)

nrc
```

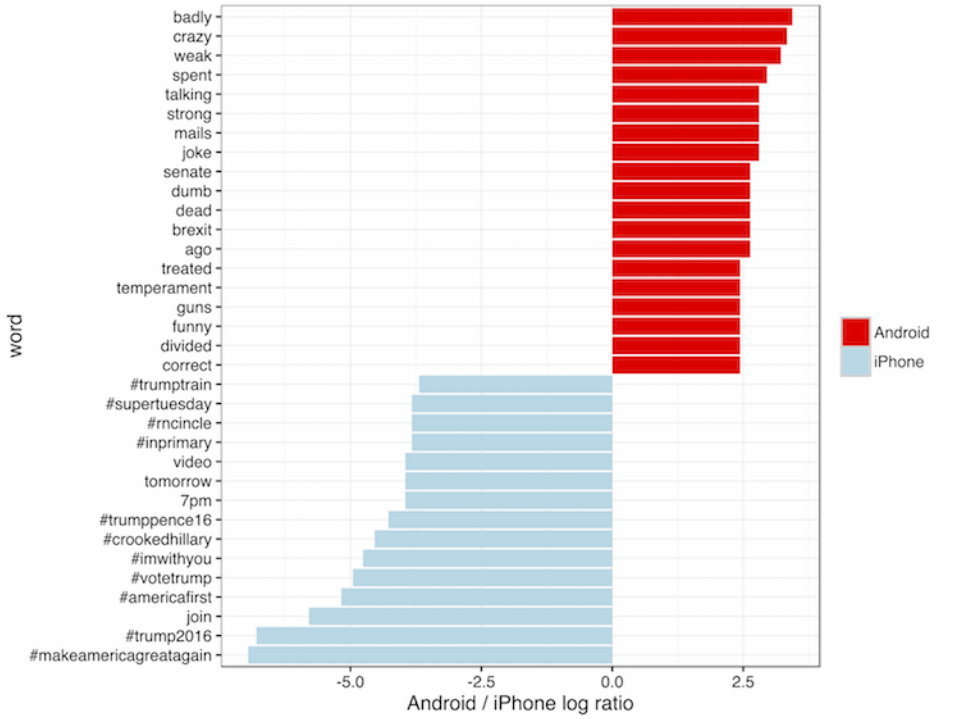


Figure 2: Top 20 words from each device

How many words fall into the `fear` sentiment? What proportion of the total number of words is this?

Question 7

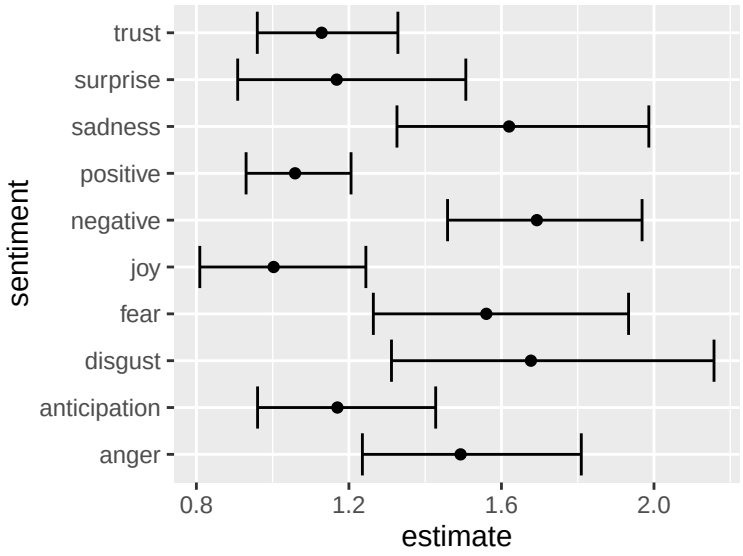
Here is the final critical analysis of David Robinson's analysis, for each sentiment a Poisson test of the differences between whether it is mode likely to emerge from the Android or iphone is conducted. This is the code.

```
sources <- tweet_words %>%
  group_by(source) %>%
  mutate(total_words = n()) %>%
  ungroup() %>%
  distinct(id, source, total_words)

by_source_sentiment <- tweet_words %>%
  inner_join(nrc, by = "word") %>%
  count(sentiment, id) %>%
  ungroup() %>%
  complete(sentiment, id, fill = list(n = 0)) %>%
  inner_join(sources) %>%
  group_by(source, sentiment, total_words) %>%
  summarize(words = sum(n)) %>%
  ungroup()

library(broom)
sentiment_differences <- by_source_sentiment %>%
```

```
  group_by(sentiment) %>%
  do(tidy(poisson.test(.$words, .$total_words)))

sentiment_differences
```

The following code produces the plot of the confidence intervals, from David's blog post, almost:

```
ggplot(sentiment_differences, aes(x=sentiment, y=estimate)) +
  geom_point() +
  geom_errorbar(aes(x=sentiment, ymin=conf.low, ymax=conf.high)) +
  coord_flip()
```



Fix it so that the sentiments are ordered from highest estimated rate to lowest. Explain why ordering makes the plot easier to read.

Question 8

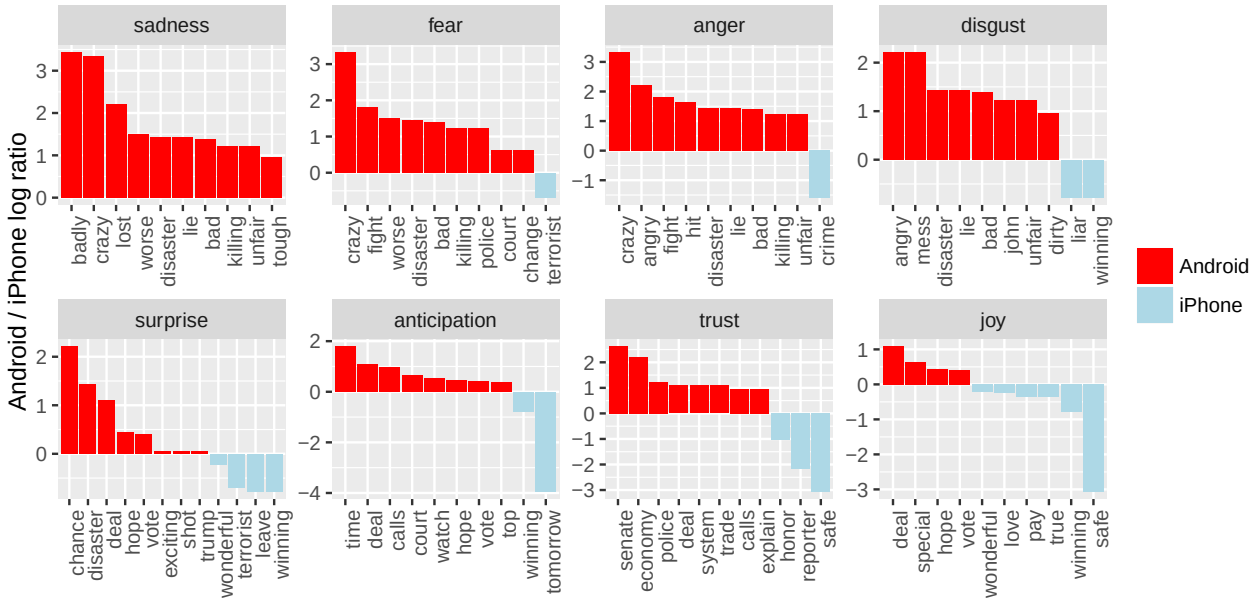
The following code makes David's final plot, and he makes this claim:

`This confirms that lots of words annotated as negative sentiments (with a few exceptions like "crime" and "terrorist") are more common in Trump's Android tweets than the campaign's iPhone tweets.`

Explain what is plotted. How does the plot support this claim?

```
android_iphone_ratios %>%
  inner_join(nrc, by = "word") %>%
  filter(!sentiment %in% c("positive", "negative")) %>%
  mutate(sentiment = reorder(sentiment, -logratio),
         word = reorder(word, -logratio)) %>%
  group_by(sentiment) %>%
  top_n(10, abs(logratio)) %>%
  ungroup() %>%
  ggplot(aes(word, logratio, fill = logratio < 0)) +
```

```
  facet_wrap(~ sentiment, scales = "free", nrow = 2) +
  geom_bar(stat = "identity") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
  labs(x = "", y = "Android / iPhone log ratio") +
  scale_fill_manual(name = "", labels = c("Android", "iPhone"),
                    values = c("red", "lightblue"))
```



Question 9

Take a look at the tweet data. Find something else to examine. Make a plot to communicate what you have learned.

WHAT TO TURN IN

Turn in two items: a `.Rmd` document, and the output `.pdf` or `.docx` from running it. Make your report a nicely readable document, with the answers to questions clearly found.

Resources

- RStudio cheat sheets