VE572 Project Part 1

Yihao Liu 515370910207

August 3, 2018

Task 1

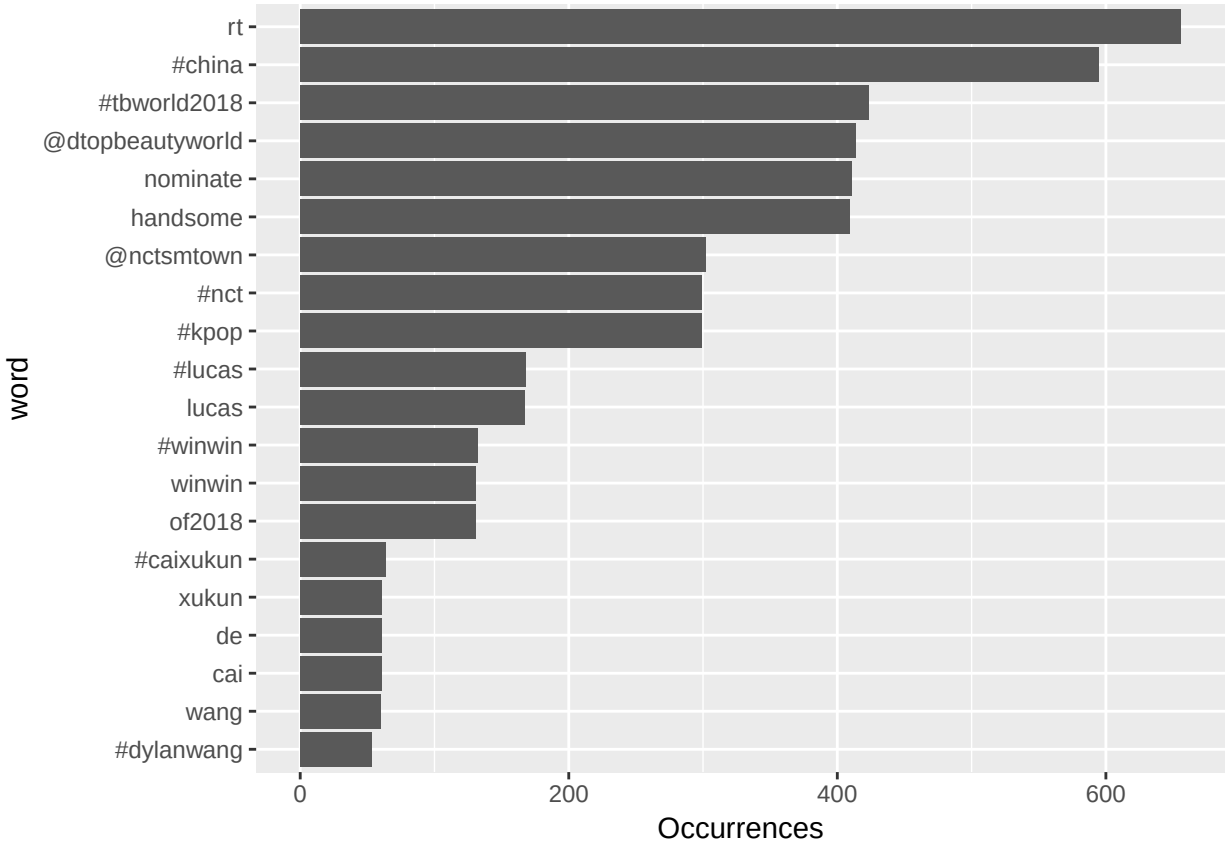
(a)

```
library(twitteR)
library(httr)
consumer_key = "bUMOqZlwm3JWqc2TJHutI7YQz"
consumer_secret = "6FORaUFkTSJP7dByIOgfVi8nLXndy9kkifN9G2DUC7loh5wnkR"
access_token = "1006882104150351872-LwKJICzZhxcEBUV4WzDTUnsKhcKABT"
access_secret = "neAA3ShJDPMfFvRdvh4nrmWLkVGm9Hl7NPyomU691stuq"
options(httr_oauth_cache = TRUE)
Sys.setenv(http_proxy="http://127.0.0.1:8123")
setup_twitter_oauth(consumer_key, consumer_secret, access_token, access_secret)
tweets = searchTwitter('#China', resultType="popular", n=1000)
save(tweets, file = 'tweets.Rdata')
```

(b)

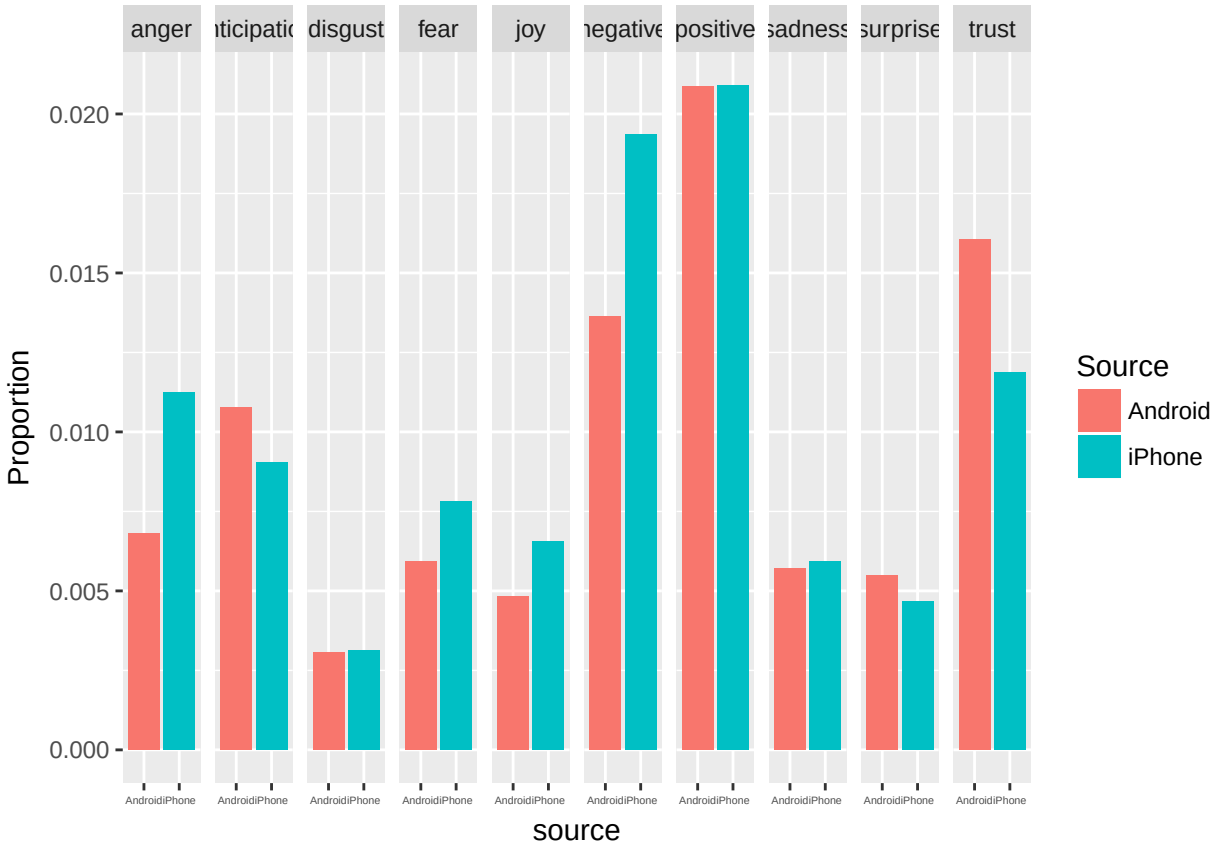
```
# i.
library(twitteR)
library(dplyr)
library(tidyr)
load(file = 'tweets.Rdata')
tweets_tb = as_tibble(purrr::map_dfr(tweets, as.data.frame)) %>%
select(id , statusSource , text, created) %>%
extract(statusSource, "source", "Twitter for (.*?)<") %>%
filter(source %in% c("iPhone", "Android"))

# ii.
library(stringr)
library(tidytext)
library(ggplot2)
reg = "([^A-Za-z\\d#@']|'(?![A-Za-z\\d#@]))"
tweets_tb = tweets_tb %>%
filter(!str_detect(text, '^"')) %>%
mutate(text = str_replace_all(text, "https://t.co/[A-Za-z\\d]+|&amp;", ""))

words = tweets_tb %>%
unnest_tokens(word, text, token = "regex", pattern = reg) %>%
filter (!word %in% stop_words$word, str_detect(word, "[a-z]"))

words %>%
count (word, sort = TRUE) %>%
head(20) %>%
mutate(word = reorder(word, n)) %>%
ggplot(aes(word, n)) +
```

```
geom_bar(stat = "identity") +
ylab("Occurrences") +
coord_flip()
```

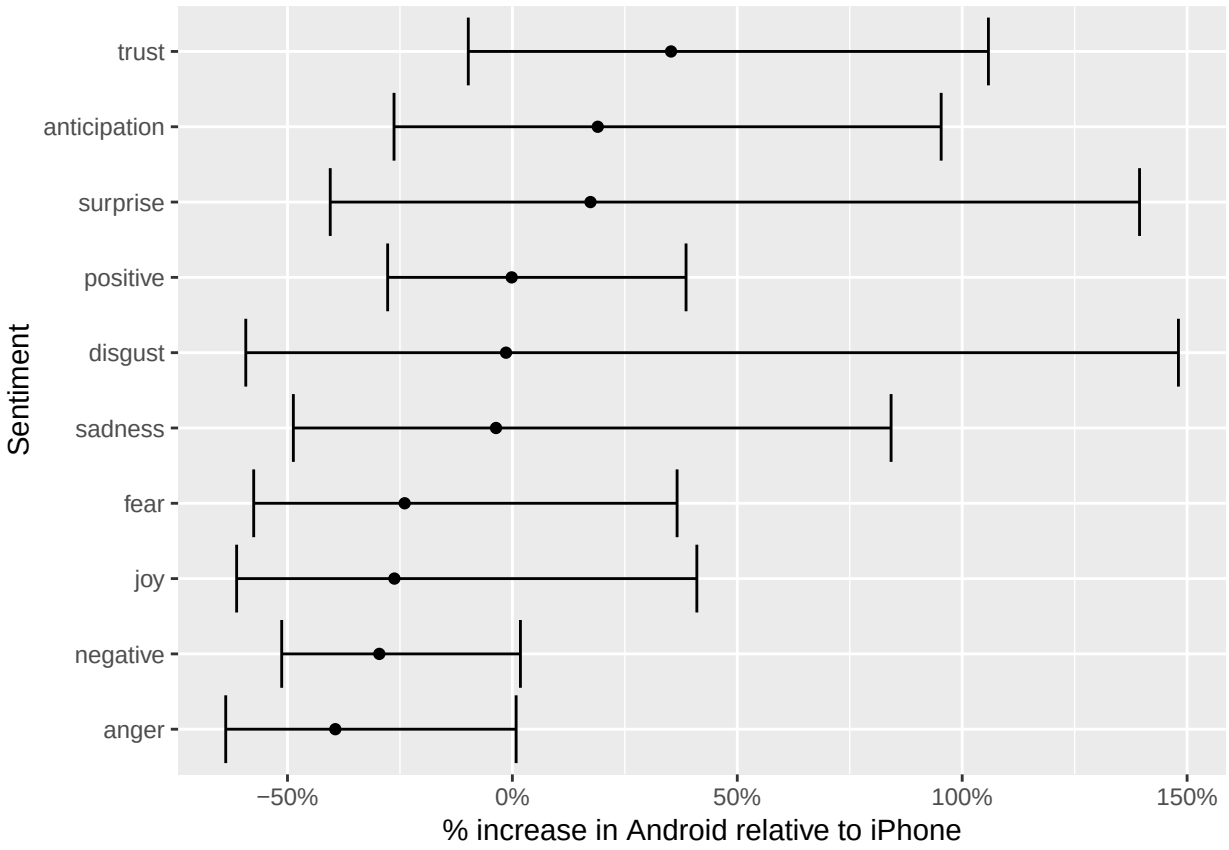



```
# iii.
nrc = sentiments %>%
filter (lexicon == "nrc") %>%
select (word , sentiment)

sources = words %>%
group_by (source) %>%
mutate (total = n ()) %>%
ungroup () %>%
distinct (id , source , total)

words_by_source_sentiment = 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) %>%
summarize(counts = sum (n)) %>%
ungroup()

words_by_source_sentiment %>%
ggplot (aes (source , counts / total , fill = source)) +
```

```
geom_bar (stat = "identity" , position = "dodge") +
labs (y = "Proportion" , fill = "Source") +
facet_grid ( ~ sentiment) +
theme(axis.text.x = element_text(size = 4))
```

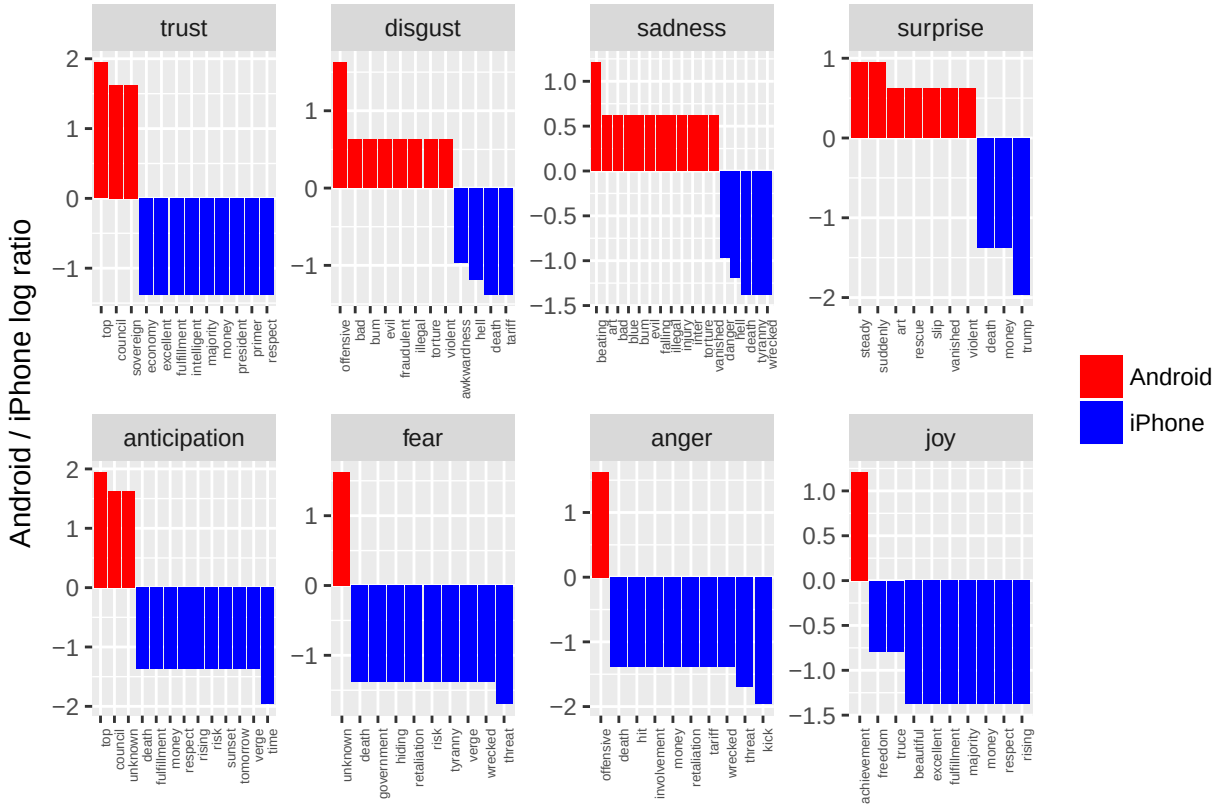



```
sentiment_differences =
words_by_source_sentiment %>%
group_by (sentiment) %>%
do (broom::tidy (poisson.test (.$counts , .$total)))

sentiment_differences %>%
ungroup () %>%
mutate (sentiment = reorder (sentiment , estimate)) %>%
mutate_at (c ("estimate" , "conf.low" , "conf.high") , funs (. - 1)) %>%
ggplot (aes (estimate , sentiment)) +
geom_point () +
geom_errorbarh (aes (xmin = conf.low , xmax = conf.high)) +
scale_x_continuous (labels = scales::percent_format ()) +
labs(x = "% increase in Android relative to iPhone", y = "Sentiment")
```




```
android_iphone_ratios = words %>%
count (word , source) %>%
spread (source, n , fill = 0) %>%
mutate_at (c ("Android" , "iPhone") , funs ((. + 1) / sum (. + 1))) %>%
mutate (logratio = log2 (Android / iPhone)) %>%
arrange (desc (logratio))

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 (
size = 5,
angle = 90 ,
hjust = 1
)) +
labs (x = "" , y = "Android / iPhone log ratio") +
scale_fill_manual (
name = " " ,
values = c ("red" , "blue"),
```

```
labels = c ("Android" , "iPhone")
)
```



 # iv. (visulization has been implemented in the above parts)