FinalProject

Weige Guo

12/18/2017

Introduction

Disputes between Capitalsim and Communism, two of the most adopted economic systems, have long existed. Given limited time these systems have been adopted as well as different measurements, it is hard to say which is a better system that could benefit a country to the most. However, exploring social data and finding out people's perception about the two systems in a society that allows freedom of speech can shed somne lights on the topic for future studies.

This project explored the online perpection, twitter exclusively, of these two systems and generated some interesting insights. The project contains four parts: Data Extraction, Text Analysis, Mapping, Emoji Analysis and Shiny Application.

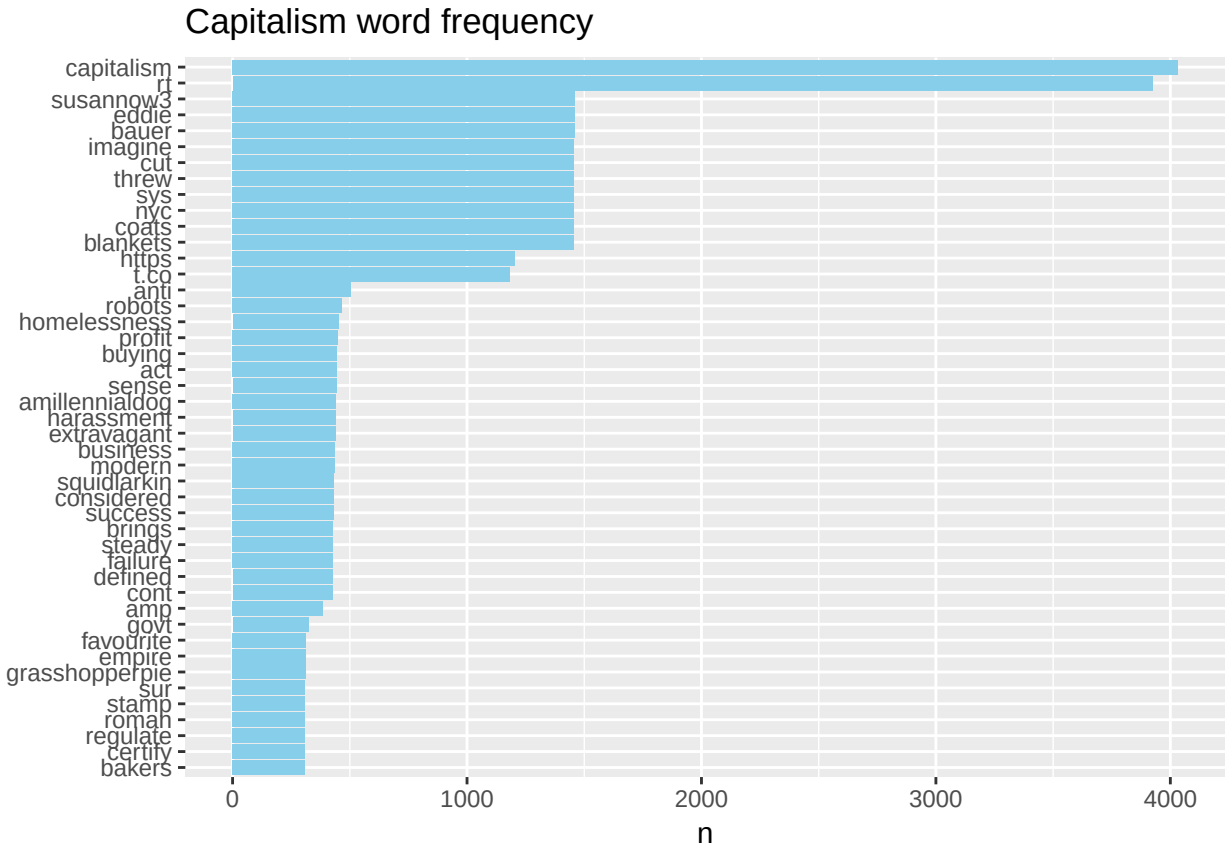
```
load("FinalShiny/veiger.rdata")
```

text analysis(capitalism) - split the text into seperate words, eliminate stop words and irrelevant words such as "rt", "tweet","http", count the occurance of words and plot the ones that occur more than 120 times in all 5000 tweets

```
tidycap <- capdata %>%
  dplyr::select(text) %>%
  unnest_tokens(word,text)
data("stop_words")
tidycap <-  tidycap %>%
  anti_join(stop_words)
```

```
## Joining, by = "word"
```

```
tidycap %>%
  count(word, sort = TRUE) %>%
  filter(n>120) %>%
  mutate(word = reorder(word, n)) %>%
  ggplot(aes(word, n)) +
  geom_col(fill = "skyblue") +
  xlab(NULL) +
  coord_flip() + ggtitle("Capitalism word frequency")
```

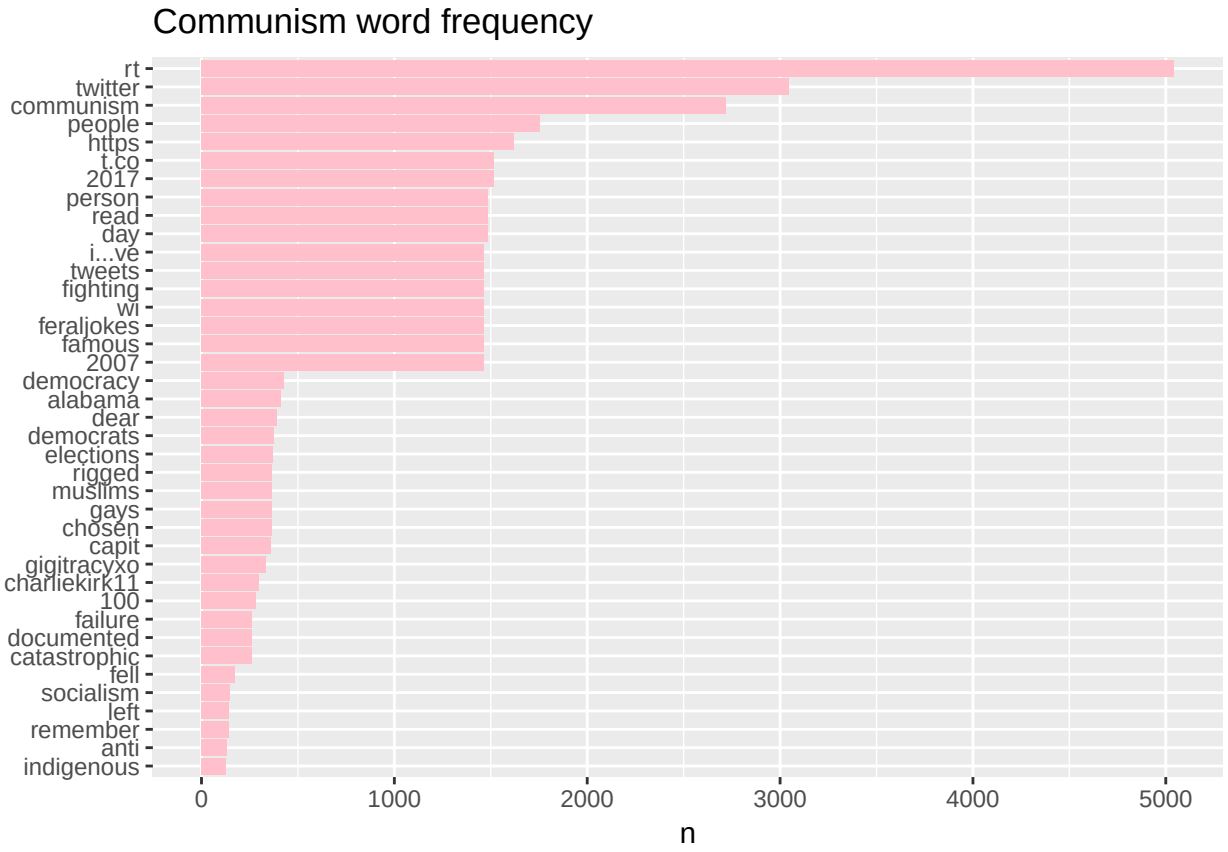


text analysis(communism) - split the text into seperate words, eliminate stop words, count the occurance of words and plot the ones that occur more than 120 times in all 5000 tweets.

```
tidycom <- comdata %>%
  dplyr::select(text) %>%
  unnest_tokens(word,text)
data("stop_words")
tidycom <-  tidycom %>%
  anti_join(stop_words)
```

```
## Joining, by = "word"
```

```
tidycom %>%
  count(word, sort = TRUE) %>%
  filter(n>120) %>%
  mutate(word = reorder(word, n)) %>%
  ggplot(aes(word, n)) +
  geom_col(fill = "pink") +
  xlab(NULL) +
  coord_flip() + ggtitle("Communism word frequency")
```



Compare the words that are frequently used in two sets of data(capitalsim and communism). The more close the word is to the top right, the more frequently it is used in both datasets.

```
frequency <- bind_rows(mutate(tidycap, ideology = "capitalism"),
                       mutate(tidycom, ideology = "communism")) %>%
  mutate(word = str_extract(word, "[a-z']+")) %>%
  count(ideology, word) %>%
  group_by(ideology) %>%
  mutate(proportion = n / sum(n)) %>%
  dplyr::select(-n) %>%
  spread(ideology, proportion) %>%
  gather(ideology, proportion, `communism`)
frequency
```

```
## # A tibble: 10,358 x 4
##         word   capitalism  ideology   proportion
##        <chr>        <dbl>     <chr>        <dbl>
##  1         a 1.428189e-04 communism 1.829278e-04
##  2     a'dam           NA communism 1.662980e-05
##  3    aagmhc           NA communism 1.662980e-05
##  4       aap           NA communism 1.662980e-05
##  5    aarhjw 1.785236e-05 communism           NA
##  6     aaron 3.570472e-05 communism           NA
##  7        ab 3.570472e-05 communism           NA
##  8   abandon 1.785236e-05 communism           NA
##  9 abandoned 3.570472e-05 communism 4.988941e-05
## 10  abbywray 1.785236e-05 communism           NA
## # ... with 10,348 more rows
```

```
library(scales)
```

```
##
## Attaching package: 'scales'
```

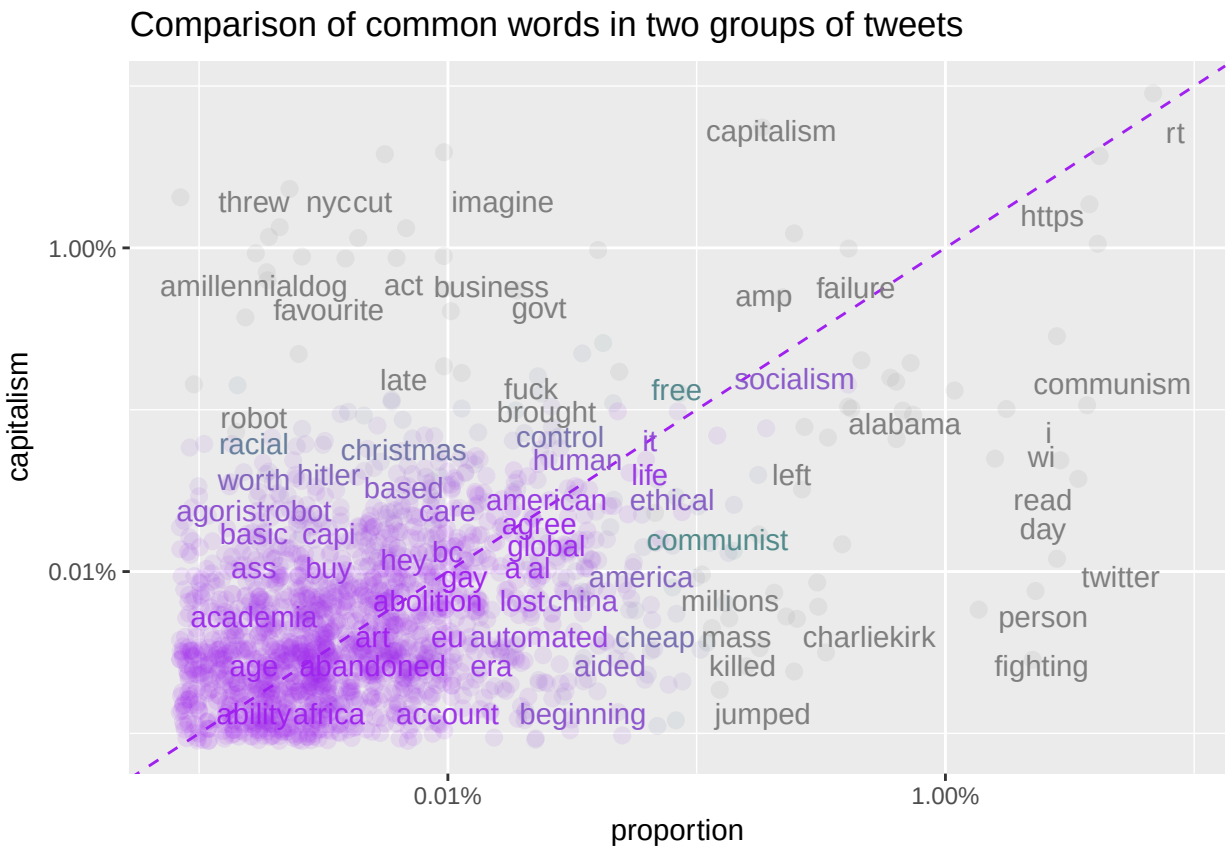
```
## The following object is masked from 'package:purrr':
##
##     discard
```

```
## The following object is masked from 'package:readr':
##
##     col_factor
```

```
ggplot(frequency, aes(x = proportion, y = `capitalism`, color = abs(`capitalism` - proportion))) +
  geom_abline(color = "purple", lty = 2) +
  geom_jitter(alpha = 0.1, size = 2.5, width = 0.3, height = 0.3) +
  geom_text(aes(label = word), check_overlap = TRUE, vjust = 1.5) +
  scale_x_log10(labels = percent_format()) +
  scale_y_log10(labels = percent_format()) +
  scale_color_gradient(limits = c(0, 0.001), low = "purple", high = "darkslategray4") +
  theme(legend.position="none") + ggtitle("Comparison of common words in two groups of tweets")
```

```
## Warning: Removed 8561 rows containing missing values (geom_point).
```

```
## Warning: Removed 8562 rows containing missing values (geom_text).
```



split the text into lines, set line number as row number, count sentiment scores for each row, and compare the result of two data sets with plots.

```
tidycap2 <- capdata %>%
  dplyr::select(text) %>%
  mutate(linenumber = row_number()) %>%
  ungroup() %>%
  unnest_tokens(word, text)

tidycom2 <- comdata %>%
  dplyr::select(text) %>%
  mutate(linenumber = row_number()) %>%
  ungroup() %>%
  unnest_tokens(word, text)


tidycap2 <- tidycap2%>%
  mutate(ideology = "capitalsim")
tidycom2 <- tidycom2%>%
  mutate(ideology = "communism")

com2cap2 <- rbind(tidycap2, tidycom2)

nrcjoy <- get_sentiments("nrc") %>%
  filter(sentiment == "joy")

nrcjoy
```

```
## # A tibble: 689 x 2
##              word sentiment
##             <chr>     <chr>
##  1    absolution       joy
##  2     abundance       joy
##  3      abundant       joy
##  4      accolade       joy
##  5 accompaniment       joy
##  6    accomplish       joy
##  7  accomplished       joy
##  8       achieve       joy
##  9   achievement       joy
## 10       acrobat       joy
## # ... with 679 more rows
```

```
joycap <- com2cap2 %>%
  inner_join(nrcjoy) %>%
  count(word, sort = TRUE)
```

```
## Joining, by = "word"
```

```
com2cap2 %>%
  filter(ideology == "capitalism") %>%
  inner_join(nrcjoy) %>%
  count(word, sort = TRUE)
```

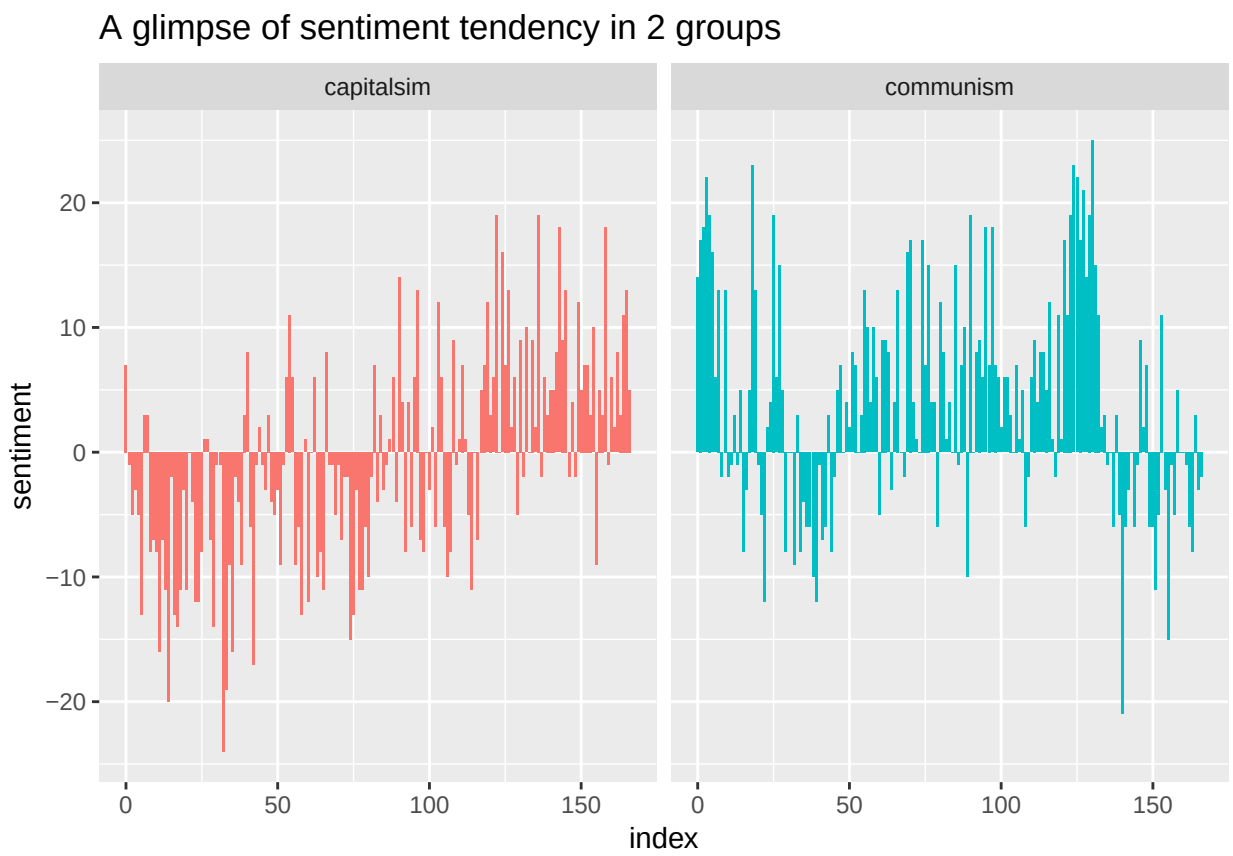
```
## Joining, by = "word"
```

```
## # A tibble: 0 x 2
## # ... with 2 variables: word <chr>, n <int>
```

```
com2cap2 <- com2cap2 %>%
  inner_join(get_sentiments("bing")) %>%
  count(ideology, index = linenumber %/% 30, sentiment) %>%
  spread(sentiment, n, fill = 0) %>%
  mutate(sentiment = positive - negative)
```

```
## Joining, by = "word"
```

```
ggplot(com2cap2, aes(index, sentiment, fill = ideology)) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~ideology, ncol = 2, scales = "free_x") + ggtitle("A glimpse of sentiment tendency in 2 gr
```



Select sentimental words in 5000 tweets mentioning "capitalism",plot the top 10 most used positive words and negative words respectively.

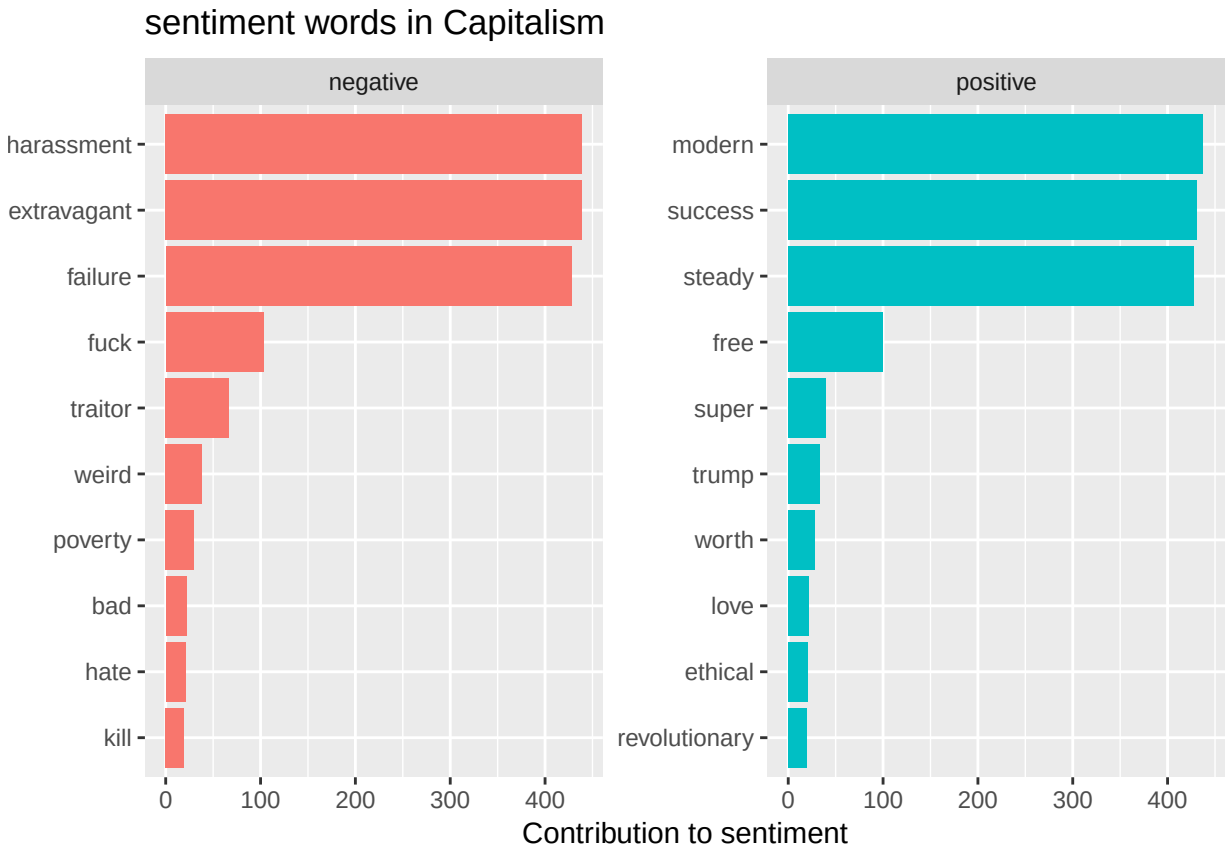
```
capwords <- tidycap %>%
  inner_join(get_sentiments("bing")) %>%
  count(word, sentiment, sort = TRUE) %>%
  ungroup()
```

```
## Joining, by = "word"
```

```
capwords %>%
  group_by(sentiment) %>%
  top_n(10) %>%
  ungroup() %>%
  mutate(word = reorder(word, n)) %>%
  ggplot(aes(word, n, fill = sentiment)) +
  geom_col(show.legend = FALSE) +
```

```
  facet_wrap(~sentiment, scales = "free_y") +
  labs(y = "Contribution to sentiment",
       x = NULL) +
  coord_flip() + ggtitle("sentiment words in Capitalism")
```

```
## Selecting by n
```



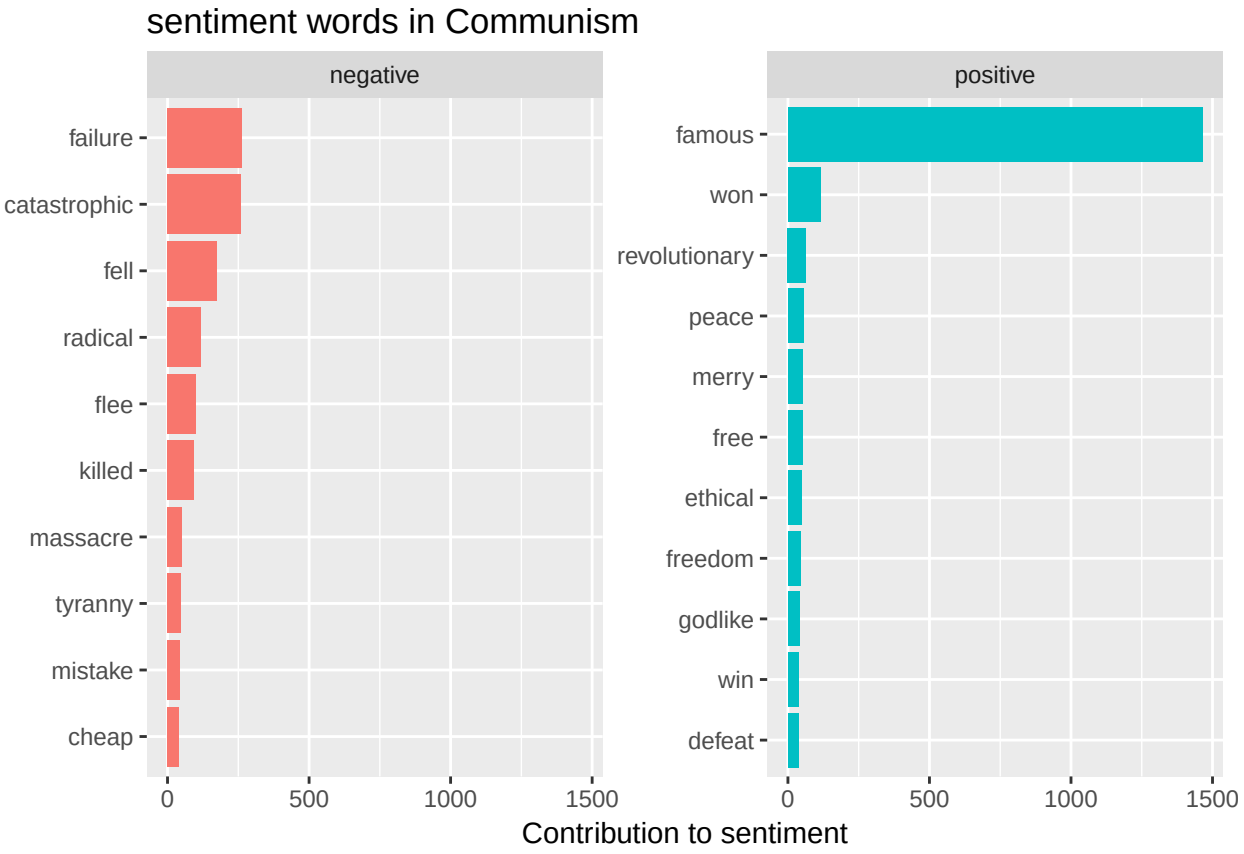
Select sentimental words in 5000 tweets mentioning "communism", plot the top 10 most used positive words and negative words respectively.

```
comwords <- tidycom %>%
  inner_join(get_sentiments("bing")) %>%
  count(word, sentiment, sort = TRUE) %>%
  ungroup()
```

```
## Joining, by = "word"
```

```
comwords %>%
  group_by(sentiment) %>%
  top_n(10) %>%
  ungroup() %>%
  mutate(word = reorder(word, n)) %>%
  ggplot(aes(word, n, fill = sentiment)) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~sentiment, scales = "free_y") +
  labs(y = "Contribution to sentiment",
       x = NULL) +
  coord_flip() + ggtitle("sentiment words in Communism")
```

```
## Selecting by n
```



wordcloud of mosted used sentimental words in 5000 tweets mentioning "capitalsim", distinguish the negative ones from the positive ones by using different colors.

```
library(reshape2)
```

```
##
## Attaching package: 'reshape2'
```

```
## The following object is masked from 'package:tidyr':
##
##     smiths
```

```
library(wordcloud)
```

```
## Loading required package: RColorBrewer
```

```
  tidycap %>%
  inner_join(get_sentiments("bing")) %>%
  count(word, sentiment, sort = TRUE) %>%
  acast(word ~ sentiment, value.var = "n", fill = 0) %>%
  comparison.cloud(colors = c("darkgrey","red"),
                   max.words = 100)
```

```
## Joining, by = "word"
```

wordcloud of mosted used sentimental words in 5000 tweets mentioning"communism", distinguish the negative ones from the positive ones by using different colors.

```
library(reshape2)


  tidycom %>%
  inner_join(get_sentiments("bing")) %>%
  count(word, sentiment, sort = TRUE) %>%
  acast(word ~ sentiment, value.var = "n", fill = 0) %>%
  comparison.cloud(colors = c("darkgrey","red"),
                   max.words = 100)
```

```
## Joining, by = "word"
```

negative

positive

Aqcuire tweets sent from the United States that contain “capitalsim” for 30 seconds, plot the map of the United States, and locate these tweets in the map.

```
library(streamR)
```

```
## Loading required package: RCurl
```

```
## Loading required package: bitops
```

```
##
## Attaching package: 'RCurl'
```

```
## The following object is masked from 'package:tidyr':
##
##     complete
```

```
## Loading required package: rjson
```

```
# filterStream("tweets_cap.json", track="capitalism",
#              locations = c(-125, 25, -66,50),  timeout = 30,
#              oauth = my_oauth)

tweets.cap <- parseTweets("FinalShiny/tweets_cap.json")
```

```
## 3596 tweets have been parsed.
```

```
library(ggplot2)
library(grid)

map.data <- map_data("state")
```

```
##
```

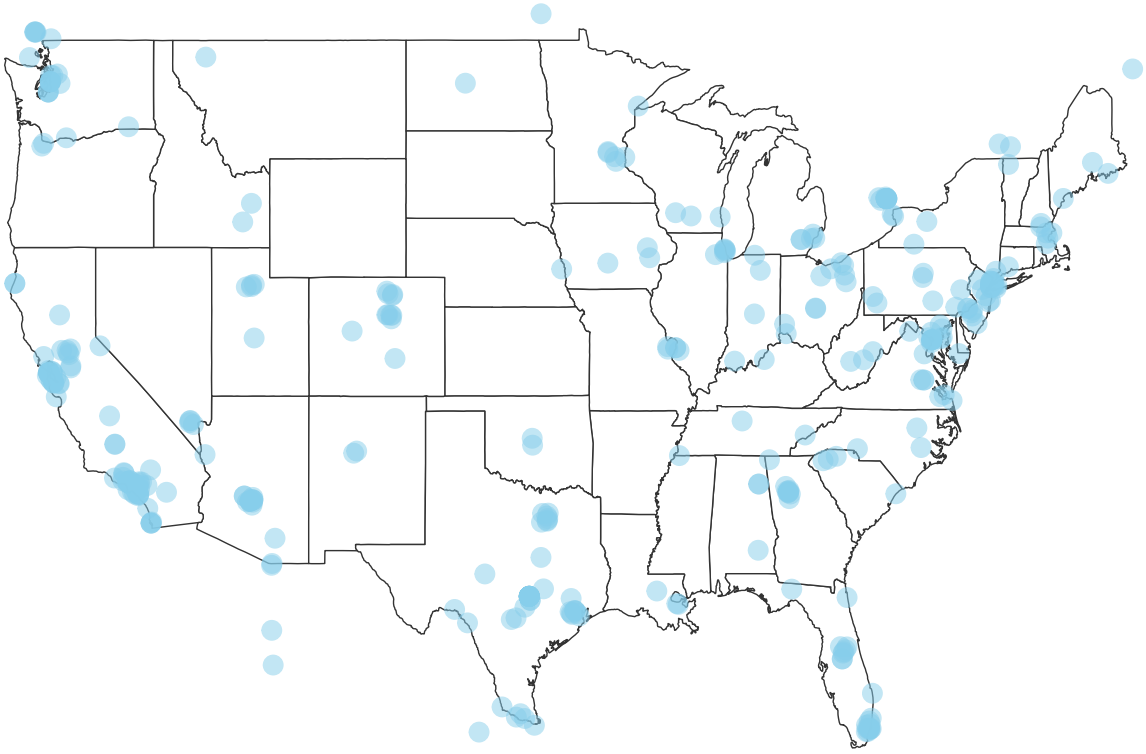
```
## Attaching package: 'maps'
```

```
## The following object is masked from 'package:purrr':
##
##     map
```

```
cappoints <- data.frame(x = as.numeric(tweets.cap$lon),
                        y = as.numeric(tweets.cap$lat))

cappoints <- cappoints[cappoints$y > 25, ]
ggplot(map.data) +
  geom_map(aes(map_id = region),
           map = map.data,
           fill = "white",
           color = "grey20", size = 0.25) +
  expand_limits(x = map.data$long, y = map.data$lat) +
  theme(axis.line = element_blank(),
        axis.text = element_blank(),
        axis.ticks = element_blank(),
        axis.title = element_blank(),
        panel.background = element_blank(),
        panel.border = element_blank(),
        panel.grid.major = element_blank(),
        plot.background = element_blank(),
        plot.margin = unit(0 * c( -1.5, -1.5, -1.5, -1.5), "lines")) +
        geom_point(data = cappoints,
        aes(x = x, y = y), size = 3,
        alpha = 1/2, color = "skyblue") + ggtitle("Where did people talk about Capitalism")
```

```
## Warning: Removed 3215 rows containing missing values (geom_point).
```

Where did people talk about Capitalism



Aqcuire tweets sent from the United States that contain "communism" for 30 seconds, plot the map of the United States, and locate these tweets in the map.

```
# filterStream("tweets_com.json", track="communism",
#              locations = c(-125, 25, -66,50),  timeout = 30,
#              oauth = my_oauth)

tweets.com <- parseTweets("FinalShiny/tweets_com.json")
```

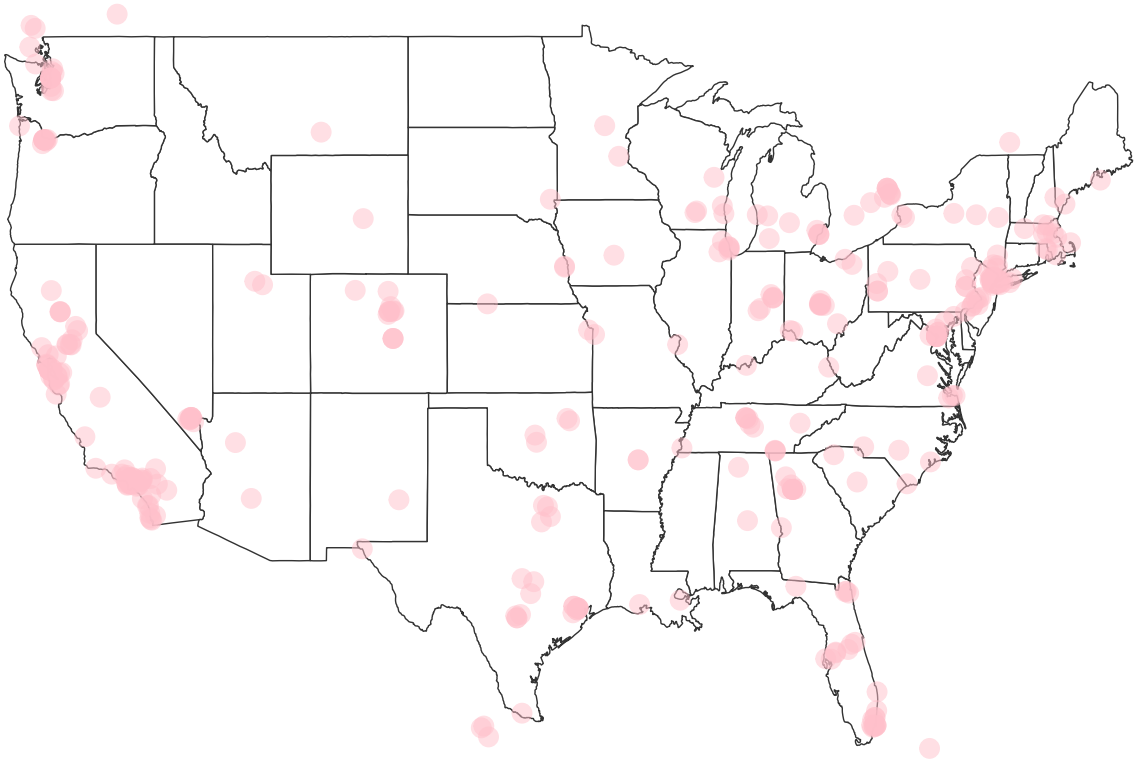
```
## 3492 tweets have been parsed.
```

```
library(ggplot2)
library(grid)

map.data <- map_data("state")
compoints <- data.frame(x = as.numeric(tweets.com$lon),
                        y = as.numeric(tweets.com$lat))

compoints <- compoints[compoints$y > 25, ]
ggplot(map.data) +
  geom_map(aes(map_id = region),
           map = map.data,
           fill = "white",
           color = "grey20", size = 0.25) +
  expand_limits(x = map.data$long, y = map.data$lat) +
  theme(axis.line = element_blank(),
        axis.text = element_blank(),
        axis.ticks = element_blank(),
        axis.title = element_blank(),
```

```
        panel.background = element_blank(),
        panel.border = element_blank(),
        panel.grid.major = element_blank(),
        plot.background = element_blank(),
        plot.margin = unit(0 * c( -1.5, -1.5, -1.5, -1.5), "lines")) +
        geom_point(data = compoints,
        aes(x = x, y = y), size = 3,
        alpha = 1/2, color = "pink")  + ggtitle("Where did people talk about Communism")
```

```
## Warning: Removed 3107 rows containing missing values (geom_point).
```

Where did people talk about Communism



Emoji Analysis

data

```
load("FinalShiny/veiger1.rdata")
```

```
capdata = capdata %>% select(text, created, screenName)
capdata$text = iconv(capdata$text, from = "latin1", to = "ascii", sub = "byte")
capdata$created <- as.POSIXlt(capdata$created)
capdata$tweetid = 1:nrow(capdata)

comdata = comdata %>% select(text, created, screenName)
comdata$text = iconv(comdata$text, from = "latin1", to = "ascii", sub = "byte")
```

```
comdata$created <- as.POSIXlt(comdata$created)
comdata$tweetid = 1:nrow(comdata)
```

```
emdict.la <- read.csv('emoticon_conversion_noGraphic.csv')
row.names(emdict.la) <- NULL
names(emdict.la) <- c('unicode', 'bytes', 'name')
emdict.la$emojiid <- row.names(emdict.la)

emdict.jpb <- read.csv('emDict.csv')
row.names(emdict.jpb) <- NULL
names(emdict.jpb) <- c('name', 'bytes', 'rencoding')
emdict.jpb$name <- tolower(emdict.jpb$name)
emdict.jpb$bytes <- NULL

emojis <- merge(emdict.la, emdict.jpb, by = 'name')
emojis$emojiid <- as.numeric(emojis$emojiid)

emojis <- arrange(emojis, emojiid)

rm(emdict.jpb,emdict.la)
```

comdata

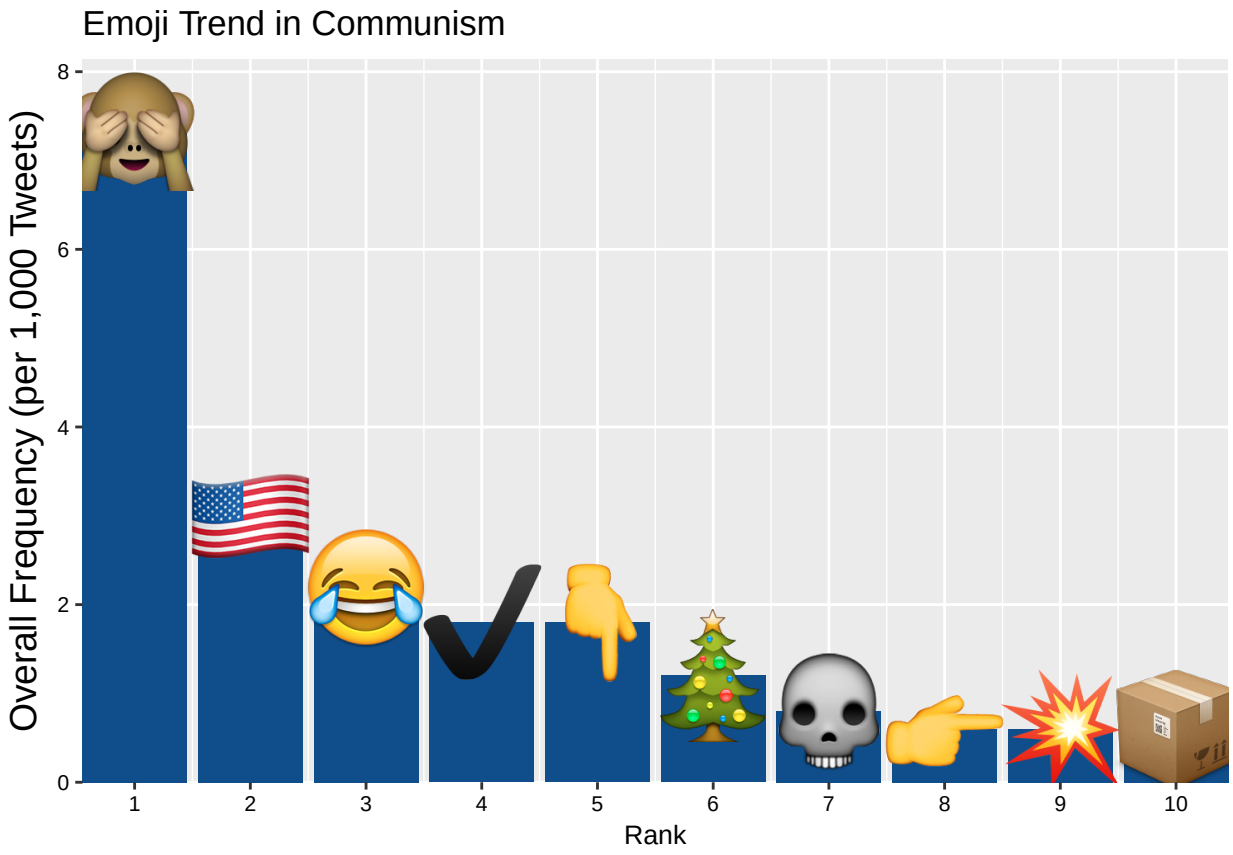
```
df.s <- matrix(NA, nrow = nrow(comdata), ncol = ncol(emojis))

system.time(df.s <- sapply(emojis$rencoding, regexpr, comdata$text, ignore.case = T, useBytes = T))
```

```
##    user  system elapsed
##  11.278   0.028  11.315
```

```
rownames(df.s) <- 1:nrow(df.s)
colnames(df.s) <- 1:ncol(df.s)
df.t <- data.frame(df.s)
df.t$tweetid <- comdata$tweetid
df = df.t[,1:842]
count = colSums(df > -1)

emojis.m <- cbind(count, emojis)
emojis.m <- arrange(emojis.m, desc(count))

emojis.count <- subset(emojis.m, count > 0)

emojis.count$dens <- round(1000 * (emojis.count$count / nrow(comdata)), 1)
emojis.count$dens.sm <- (emojis.count$count + 1) / (nrow(comdata) + 1)

emojis.count$rank <- as.numeric(row.names(emojis.count))
emojis.count.p <- subset(emojis.count, select = c(name, dens, count, rank))
```

```
df.plot <- subset(emojis.count.p, rank <= 10); xlab <- 'Rank'; ylab <- 'Overall Frequency (per 1,000 Tw
setwd('ios_9_3_emoji_files/');
df.plot <- arrange(df.plot, name);
imgs <- lapply(paste0(df.plot$name, '.png'), png::readPNG); g <- lapply(imgs, grid::rasterGrob);
```

```
k <- 0.20 * (10/nrow(df.plot)) * max(df.plot$dens); df.plot$xsize <- k; df.plot$ysize <- k; #df.plot$xs
df.plot <- arrange(df.plot, name);
g1 <- ggplot(data = df.plot, aes(x = rank, y = dens)) +
  geom_bar(stat = 'identity', fill = 'dodgerblue4') +
  xlab(xlab) + ylab(ylab) +
  mapply(function(x, y, i) {
    annotation_custom(g[[i]], xmin = x-0.5*df.plot$xsize[i], xmax = x+0.5*df.plot$xsize[i],
                      ymin = y-0.5*df.plot$ysize[i], ymax = y+0.5*df.plot$ysize[i])},
    df.plot$rank, df.plot$dens, seq_len(nrow(df.plot))) +
  scale_x_continuous(expand = c(0, 0), breaks = seq(1, nrow(df.plot), 1), labels = seq(1, nrow(df.plot)
  scale_y_continuous(expand = c(0, 0), limits = c(0, 1.10 * max(df.plot$dens))) +
  theme(panel.grid.minor.y = element_blank(),
        axis.title.x = element_text(size = 10), axis.title.y = element_text(size = 14),
        axis.text.x  = element_text(size = 8, colour = 'black'), axis.text.y  = element_text(size = 8,
g1
```




```
# png(paste0('emoji_barchart_', as.Date(min(comdata$created)), '_', as.Date(max(comdata$created)), '_',
#     width = 6600, height = 4000, units = 'px', res = 1000);
# g1
# dev.off()
```

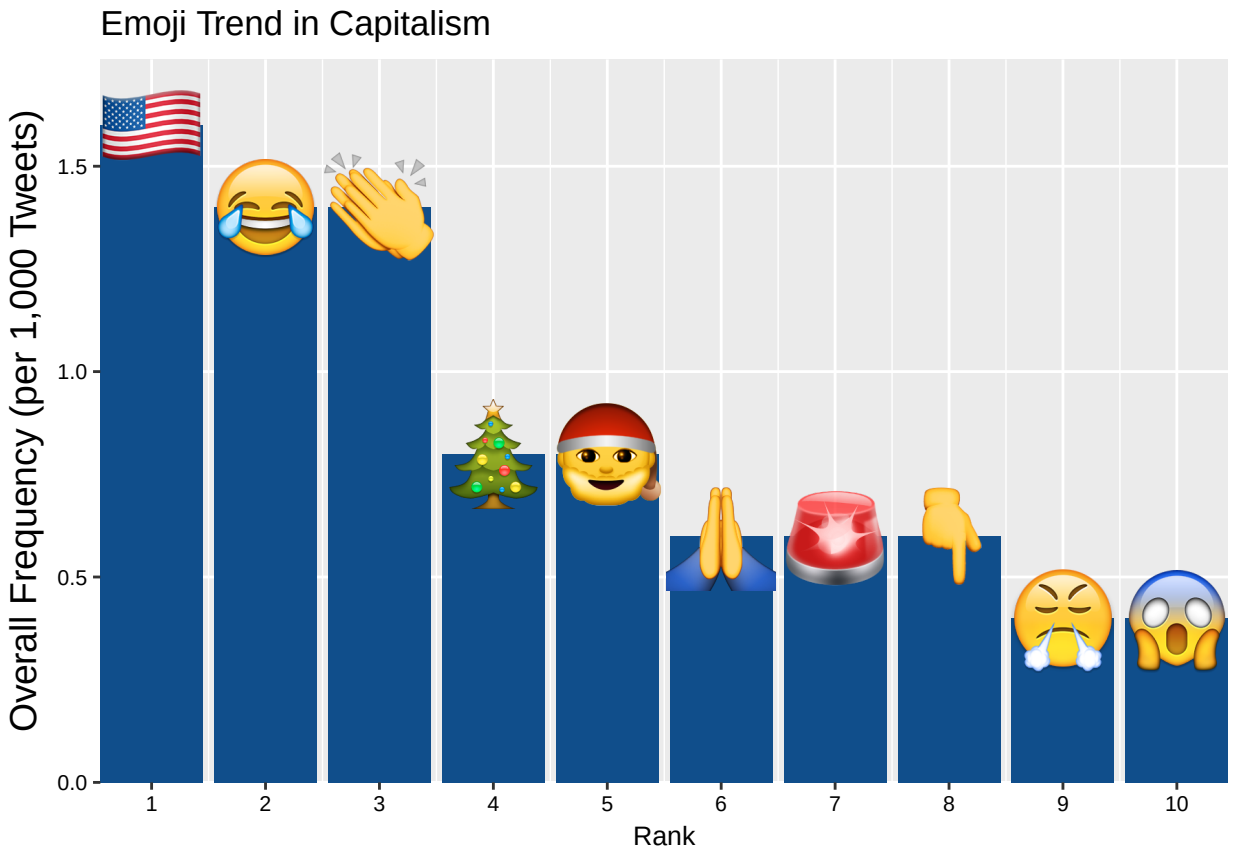
campdata

```
df.s <- matrix(NA, nrow = nrow(capdata), ncol = ncol(emojis))

system.time(df.s <- sapply(emojis$rencoding, regexpr, capdata$text, ignore.case = T, useBytes = T))
```

```
##    user  system elapsed
##  11.435   0.027  11.471
```

```
rownames(df.s) <- 1:nrow(df.s)
colnames(df.s) <- 1:ncol(df.s)
df.t <- data.frame(df.s)
df.t$tweetid <- capdata$tweetid
df = df.t[,1:842]
count = colSums(df > -1)

emojis.m <- cbind(count, emojis)
emojis.m <- arrange(emojis.m, desc(count))

emojis.count <- subset(emojis.m, count > 0)

emojis.count$dens <- round(1000 * (emojis.count$count / nrow(capdata)), 1)
emojis.count$dens.sm <- (emojis.count$count + 1) / (nrow(capdata) + 1)

emojis.count$rank <- as.numeric(row.names(emojis.count))
emojis.count.p <- subset(emojis.count, select = c(name, dens, count, rank))
```

```
df.plot <- subset(emojis.count.p, rank <= 10); xlab <- 'Rank'; ylab <- 'Overall Frequency (per 1,000 Tw
setwd('ios_9_3_emoji_files/');
df.plot <- arrange(df.plot, name);
imgs <- lapply(paste0(df.plot$name, '.png'), png::readPNG); g <- lapply(imgs, grid::rasterGrob);
k <- 0.60 * (10/nrow(df.plot)) * max(df.plot$dens); df.plot$xsize <- k; df.plot$ysize <- k; #df.plot$xs
df.plot <- arrange(df.plot, name);
g1 <- ggplot(data = df.plot, aes(x = rank, y = dens)) +
  geom_bar(stat = 'identity', fill = 'dodgerblue4') +
  xlab(xlab) + ylab(ylab) +
  mapply(function(x, y, i) {
    annotation_custom(g[[i]], xmin = x-0.5*df.plot$xsize[i], xmax = x+0.5*df.plot$xsize[i],
                      ymin = y-0.5*df.plot$ysize[i], ymax = y+0.5*df.plot$ysize[i])},
    df.plot$rank, df.plot$dens, seq_len(nrow(df.plot))) +
  scale_x_continuous(expand = c(0, 0), breaks = seq(1, nrow(df.plot), 1), labels = seq(1, nrow(df.plot)
  scale_y_continuous(expand = c(0, 0), limits = c(0, 1.10 * max(df.plot$dens))) +
  theme(panel.grid.minor.y = element_blank(),
        axis.title.x = element_text(size = 10), axis.title.y = element_text(size = 14),
        axis.text.x  = element_text(size = 8, colour = 'black'), axis.text.y  = element_text(size = 8,
g1
```


Emoji Trend in Capitalism
Overall Frequency (per 1,000 Tweets)
0.0
0.5
1.0
1.5
1
2
3
4
5
6
7
8
9
10
Rank