CS688 Final Term Project Option 3:
Twitter Stock Market Sentiment Analysis Term Project

Dataset

For this project, the 3 stock gainers and 3 stock losers were retrieved from http://finance.yahoo.com/ as of October, 14 2019. The gainers and losers were chosen by going to the Markets page and referring to the Stocks: Gainers and Stocks: Losers tabs. Ideally, the top 3 of each tab would have been chosen for this project; however, some stocks did not return at least 100 tweets. Therefore, the first stock gainers and losers in the tables that did return at least 100 tweets were chosen.

The gainer stocks included:

1. AECOM ($ACM)
2. Reata Pharmaceuticals, Inc. ($RETA)
3. Shopify Inc. ($SHOP)

The loser stocks included:

1. Parsley Energy, Inc. ($PE)
2. SmileDirectClub, Inc. ($SDC)
3. CrowdStrike Holdings, Inc. ($CRWD)

a) (16 points) Search for the 100 tweets associated with each of the three stocks in each set.

To ensure that the tweets were returning information regarding the company, particularly their success or loss in the stock market, the cashtag was used to pull all tweets. Although it was an option to pull tweets using the company name, some companies did not post many tweets or the company name would return tweets not in relation to the stock market.

The tweets for the gainers and losers were combined in the following manner.

Gainer stocks (gainers.all):

1. gainer1 ($ACM): 1-100
2. gainer2 ($RETA): 101-200
3. gainer3 ($SHOP): 201-300

The loser stocks included:

1. loser1 ($PE): 1-100
2. loser2 ($SDC): 101-200
3. loser3 ($CRWD): 201-300

b) (16 points) Create two separate data corpora (or tidy text objects) for the above two sets of tweets.

The gainer and loser groups were turned into tibble tables in order to apply the functions with tidytext.

To save the data for the day's tweets, each gainer and loser was mined in a separate Rdata file and loaded into the Project R file. The data for Twitter API access was also saved in a separate Rdata file. A user will need to revise the file location to where the data resides on their computer.

c) (16 points) Use the necessary pre-processing transformations described in the lecture notes.

The pre-processing function (cleaning.function) performs the following pre-processing on the tweets in this order:

1. Removes http elements
2. Transforms all words into lowercase
3. Removes punctuation
4. Removes English stop-words

Although removing English stop-words prior to removing punctuation is also suggested (to remove those with apostrophes), it would be interesting to see if these words are listed in the most frequent list and how it affects the context of the tweet.

d) (8 points) Create the document-term matrix for each set. Name them dtm1 and dtm2.

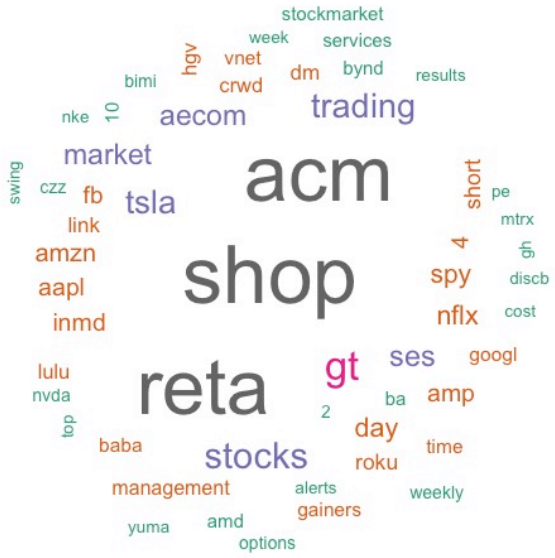
This section does not apply, as the objects are using tidytext functions.

e) (8 points) Find the most frequent terms from each set. Show a word cloud for each set.

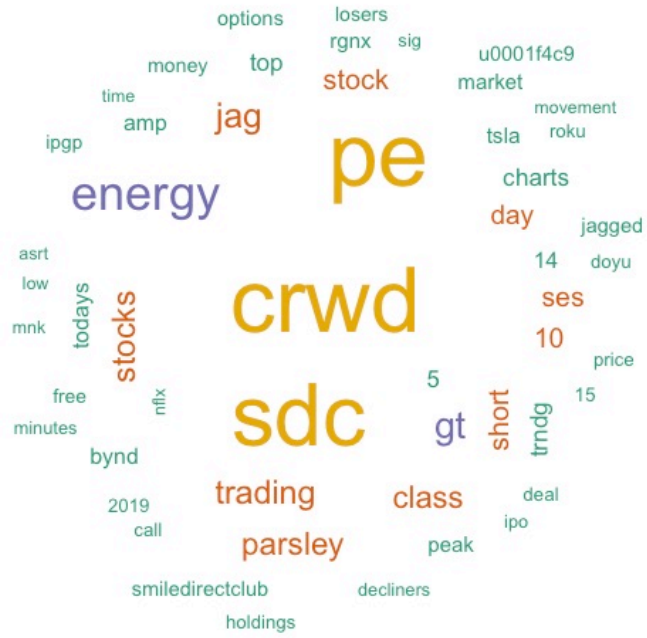
Top 10 Words for Gainer Stocks		**Top 10 Words for Loser Stocks**	
Words	*Number*	*Words*	*Number*
shop	33600	crwd	38400
acm	32700	pe	38400
reta	32100	sdc	36600
gt	14700	energy	18000
stocks	12600	gt	12900
trading	11400	jag	11400
tsla	10800	trading	10500
ses	9300	stocks	10200
aecom	9000	parsley	9900
market	9000	class	9600

The following word clouds show the top 50 most frequent terms for the gainer and loser stocks.

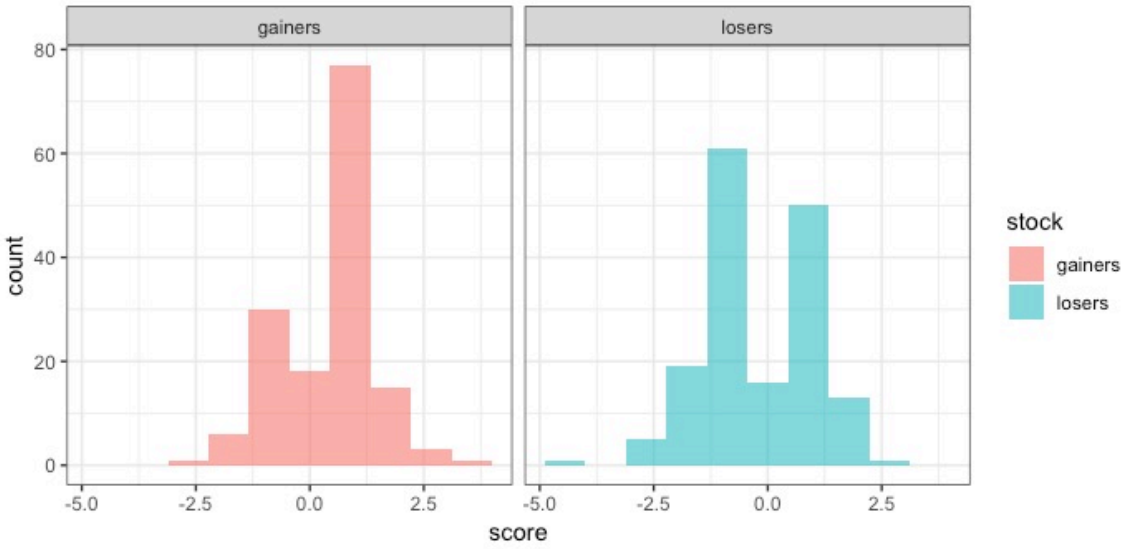
Gainers Word Cloud



Losers Word Cloud



f) (16 points) Using the positive and negative word lists, compute the sentiment score (as described in the lecture) for all the tweets for each gainers (losers) set. Were the tweets about the 3 largest gainer stocks for that day characterized by a positive sentiment, and the tweets about the 3 largest loser stocks for that day characterized by a negative sentiment?



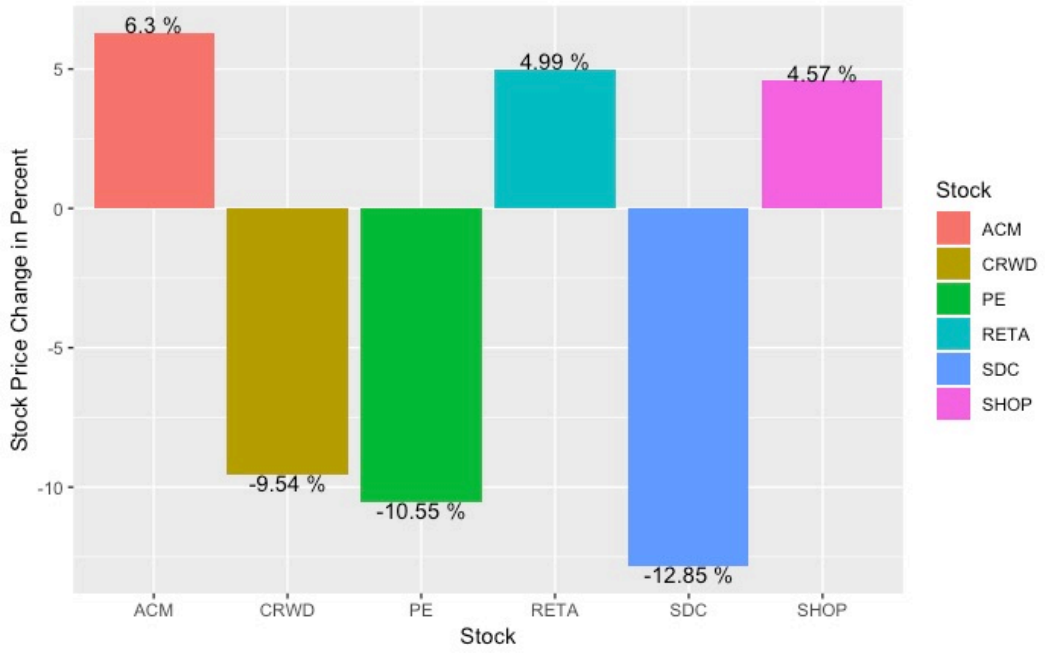
Stock	Count	Median	SD	Max	Min
gainers	300	1	1.17	4	-3
losers	300	-1	1.35	3	-4

The tweets about the 3 largest gainers and 3 largest losers are characterized by positive and negative sentiments, respectively. In the comparison graph, there are more positive scores, where a score of 1 is almost 80 words for the gainer group. The loser group still has some positive words, but a score of 1 is only reached by about 50 words. The loser group does have more -1 words, which totals to more than 60 words and twice as much as the gainer group. To better see the comparison between positive and negative sentiments, Type 1 words are removed from the graph.

The summary table also demonstrates how gainers have a more positive sentiment and losers have a more negative sentiment. The table data also has the Type 1 scores removed. Using the median, gainers have a positive sentiment of 1 and losers have a positive sentiment of -1.

g) (For up to 10 points extra credit) Create ONE appropriate data visualization, using the principles of Module 6, that shows the stock prices and/or the change in stock prices for the stocks and day you selected for this project.

The graph below charts the change in stock price by percent. This visual allows the viewer to easily compare the percentage amounts between both the gainer and loser stocks.



Appendix A: Project R Code

```
library(rtweet)
library(dplyr)
library(tidyr)
library(ggplot2)
library(tidytext)
library(tidyverse)
library(SnowballC)
library(purrr)
library(wordcloud)

load(file = '~/Desktop/BU MET/688/Project/
 twitterAuthentication.Rdata')
## Execute load file code above to retreive API information.

load(file = '~/Desktop/BU MET/688/Project/tweets.Rdata')
## Execute load file code above to retreive mined tweets.

load(file = '~/Desktop/BU MET/688/Project/Project data.Rdata')
## Execute load file code above to retreive all data for the
 following project

create_token(app = app_name,
             consumer_key = consumer_key,
             consumer_secret = consumer_secret,
             access_token = access_token,
             access_secret = access_secret)

## Gainers as of 10/14/2019 ---------------------------------------
# gainer1 <- search_tweets(
#   "$ACM", n = 150, include_rts = FALSE
# )
#
# gainer1 <- gainer1[1:100,]
#
# gainer2 <- search_tweets(
#   "$RETA", n = 150, include_rts = FALSE
# )
# gainer2 <- gainer2[1:100,]
#
# gainer3 <- search_tweets(
#   "$SHOP", n = 150, include_rts = FALSE
# )
# gainer3 <- gainer3[1:100,]
#
```

```
## Losers as of 10/14/2019 -------------------------------------
#
# loser1 <- search_tweets(
#   "$PE", n = 150, include_rts = FALSE
# )
#
# loser1 <- loser1[1:100,]
#
# loser2 <- search_tweets(
#   "$SDC", n = 150, include_rts = FALSE
# )
# loser2 <- loser2[1:100,]
#
# loser3 <- search_tweets(
#   "$CRWD", n = 150, include_rts = FALSE
# )
# loser3 <- loser3[1:100,]

## Create 2 tidy text objects ----------------------------------
# gainers.all <- c(gainer1$text,gainer2$text,gainer3$text)
# gainers.tib <- tibble(gainers.all)
#
# losers.all <- c(loser1$text,loser2$text,loser3$text)
# losers.tib <- tibble(losers.all)


## Pre-processing Function -------------------------------------
cleaning.function <- function(tweets){
  # Remove http elements manually
  tweets$stripped_text <- gsub("http\\S+","",tweets)
  tweets$stripped_text <- gsub("[^\u0020-\u007F]
 +","",tweets$stripped_text)
  tweets$stripped_text <- gsub("'|’","",tweets$stripped_text)

  # Lowercase, remove punctuation, remove English stop_words
  data("stop_words")
  tweets.clean <- tweets %>%
    select(stripped_text) %>%
    unnest_tokens(word, stripped_text) %>%
    anti_join(stop_words)
}

gainers.clean <- cleaning.function(gainers.tib)
losers.clean <- cleaning.function(losers.tib)
```

```r
## Find most frequent terms ----------------------------------
gainers.frequency <- gainers.clean %>%
  count(word, sort = TRUE)
gainers.frequency %>% top_n(10)
gainers.top <- gainers.frequency %>% top_n(50)

losers.frequency <- losers.clean %>%
  count(word, sort = TRUE)
losers.frequency %>% top_n(10)
losers.top <- losers.frequency %>% top_n(50)

## Create word cloud for each set ----------------------------
wordcloud(words = gainers.top$word, freq = gainers.top$n,
          random.order=FALSE,
          colors=brewer.pal(8, "Dark2"))

wordcloud(words = losers.top$word, freq = losers.top$n,
          random.order=FALSE,
          colors=brewer.pal(6, "Dark2"))

## Compute sentiment score ----------------------------------
get_sentiments('bing')

sentiment_bing = function(twt){
  #Step 1;  perform basic text cleaning (on the tweet), as seen
 earlier
  twt_tbl = tibble(text = twt) %>%
    mutate(
      # Remove http elements manually
      stripped_text = gsub("http\\S+","",text)
    ) %>%
    unnest_tokens(word,stripped_text) %>%
    anti_join(stop_words, by="word") %>%  #remove stop words
    inner_join(get_sentiments("bing"), by="word") %>% # merge with
 bing sentiment
    count(word, sentiment, sort = TRUE) %>%
    ungroup() %>%
    ## Create a column "score", that assigns a -1 one to all
 negative words, and 1 to positive words.
    mutate(
      score = case_when(
        sentiment == 'negative'~ n*(-1),
        sentiment == 'positive'~ n*1)
```

```
    )
  ## Calculate total score
  sent.score <- case_when(
    nrow(twt_tbl)==0~0, # if there are no words, score is 0
    nrow(twt_tbl)>0~sum(twt_tbl$score) #otherwise, sum the positive
 and negatives
  )
  ## This is to keep track of which tweets contained no words at
 all from the bing list
  zero.type <- case_when(
    nrow(twt_tbl)==0~"Type 1", # Type 1: no words at all, zero = no
    nrow(twt_tbl)>0~"Type 2" # Type 2: zero means sum of words = 0
  )
  list(score = sent.score, type = zero.type, twt_tbl = twt_tbl)
}

gainers_sent <- lapply(gainers.all,function(x){sentiment_bing(x)})
losers_sent <- lapply(losers.all,function(x){sentiment_bing(x)})

stock_sentiment <- bind_rows(
  tibble(
    stock = 'gainers',
    score = unlist(map(gainers_sent,'score')),
    type = unlist(map(gainers_sent,'type'))
  ),
  tibble(
    stock = 'losers',
    score = unlist(map(losers_sent,'score')),
    type = unlist(map(losers_sent,'type'))
  )
)

ggplot(stock_sentiment %>% filter(type != "Type 1"), aes(x=score,
 fill = stock)) + geom_histogram(bins = 10, alpha = .6) +
  facet_grid(~stock) + theme_bw()

stock_sentiment %>% group_by(stock) %>%
  filter(type != "Type 1") %>%
  summarise(
    Count = n(),
    Median = median(score),
    SD = sd(score),
    max = max(score),
    min = min(score)
```

```r
  )

## Create one appropriate data visualization
 --------------------------

stock_info <- data.frame(
  Type = c("Gainer","Gainer","Gainer","Loser","Loser","Loser"),
  Stock = c("ACM","RETA","SHOP","PE","SDC","CRWD"),
  ChangePercent = c(6.3,4.99,4.57,-10.55,-12.85,-9.54)
)

ggplot(stock_info, aes(Stock, ChangePercent)) +
  geom_bar(stat = "identity", aes(fill = Stock), legend = FALSE) +
  geom_text(aes(label = paste(ChangePercent, "%"),
                vjust = ifelse(ChangePercent >= 0, 0, 1))) +
  scale_y_continuous("Stock Price Change in Percent")
```

```
app_name = 'Darigo\'s App to connect to R'
consumer_key = 'Qjqx9txvno6JmXgGbgYQnmJ83'
consumer_secret = 'KChTvThSQb9w9uTaqH1pCvtEe6vlWkP9KLQFge4SfDsHjUjE3i'
access_token = '1120057260368068609-VQhFlSn66O4ytSgJbDB7Dd4O5Z2EiP'
access_secret = 'JpQabNjLHz7N6YpsDwHSf0uGGSA9hsqM9ZbcglBy45TUM'

save(app_name,
 consumer_key,consumer_secret,access_token,access_secret,
     file = '~/Desktop/BU MET/688/Module 5/
 twitterAuthentication.Rdata')
```