Sai Abhilash Ghanta.

CS688 Final Term Project

1. **Tweeter stock market sentiment analysis term project:**
   Dataset: Your choice of 6 stocks, 3 largest gainer(loser) stocks for the day.
   (suggestions: http://finance.yahoo.com/; http://www.google.com/finance)

Use the R Twitter API, to implement sentiment analysis for the following:
Pick 2 sets of 3 stocks. For the first set select the 3 largest gainer stocks for that day, and for the second set select the 3 largest loser stocks for that day. Create an R code that will

a) (20 points) Search for the 100 tweets associated with each of these two sets.

```
> t.CAPN <- searchTwitter('$CAPN', n = 100)
Warning message:
In doRppAPICall("search/tweets", n, params = params, retryOnRateLimit = retryO
nRateLimit,  :
  100 tweets were requested but the API can only return 53
> t.XGTI <- searchTwitter('$XGTI', n = 100)
> t.EBIO <- searchTwitter('$EBIO', n = 100)
> t.BGMD <- searchTwitter('$BGMD', n = 100)
Warning message:
In doRppAPICall("search/tweets", n, params = params, retryOnRateLimit = retryO
nRateLimit,  :
  100 tweets were requested but the API can only return 22
> t.ENZN <- searchTwitter('$ENZN', n = 100)
Warning message:
In doRppAPICall("search/tweets", n, params = params, retryOnRateLimit = retryO
nRateLimit,  :
  100 tweets were requested but the API can only return 31
> t.IMDZ <- searchTwitter('$IMDZ', n = 100)
Warning message:
In doRppAPICall("search/tweets", n, params = params, retryOnRateLimit = retryO
nRateLimit,  :
  100 tweets were requested but the API can only return 70
```

   - Combine all the gainers (losers) into one set of 300 tweets.

b) (20 points) Create two separate data corpora for the above two sets of tweets.
   - Create a simple function that takes a tweet and returns a corpus.
   - Pass the 2 tweets to this function to get the corpora and name the corpora data.corpus1 and data.corpus2.

```
> get.corpus <- function (tweets)
+ {
+   tweets.text <- lapply(tweets, function(t) {t$getText()})
+   data.source <- VectorSource(tweets.text)
+   data.corpus <- Corpus(data.source)
+ }
> data.corpus.gainers <- get.corpus(gainers)
```

- Save the 2 corpora with the above names as R objects using *writeCorpus()*.

```
> writeCorpus(data.corpus.gainers,path = "Corpus.Gainers")
> data.corpus.losers <- get.corpus(losers)
> writeCorpus(data.corpus.losers,path= "Corpus.Losers")
```

c) (20 points) Use the necessary pre-processing transformations described in the lecture notes.

- Implement the pre-processing as a function that takes a corpus and returns a pre-processed corpus.

```
> trans.corpus <- function (data.corpus) {
+   data.corpus <- tm_map(data.corpus, content_transformer(removePunctuation))
+   data.corpus <- tm_map(data.corpus, content_transformer(tolower))
+   data.corpus <- tm_map(data.corpus, removeNumbers)
+   english.stopwords <- stopwords("en")
+   removeURL <- function(x) gsub("http[[:alnum:]]*", "", x)
+   data.corpus <- tm_map(data.corpus, content_transformer(removeURL))
+   data.corpus <- tm_map(data.corpus,content_transformer(removeWords),english
.stopwords)
+   data.corpus <- tm_map(data.corpus,content_transformer(stemDocument))
+   data.corpus <- tm_map(data.corpus,content_transformer(stripWhitespace))
+
+ }
> data.corpus.gainers <- trans.corpus(data.corpus.gainers)
> data.corpus.losers <- trans.corpus(data.corpus.losers)
```

d) (10 points) Create the term-document matrix for each set.

- Name them tdm1 and dtm2.
- Save the 2 term-document matrix with the above names as R objects.

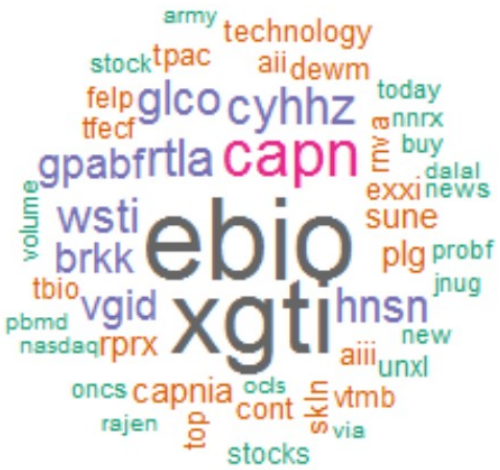
```
> tdm.Gainers <- TermDocumentMatrix(data.corpus.gainers)
> tdm.Losers <- TermDocumentMatrix(data.corpus.losers)
> save(tdm.Gainers, file = "tdm1")
> save(tdm.Losers, file = "tdm2")
```

e) (10 points) Find the most frequent terms from each set.

- Show word cloud for each set.

```
> m.Gainers <-as.matrix(tdm.Gainers)
> m.Losers <- as.matrix(tdm.Losers)
> frq.Gainers <- rowSums(m.Gainers)
> frq.Gainers <- sort(frq.Gainers,decreasing = T)
> frq.Losers <- rowSums(m.Losers)
> frq.Losers <- sort(frq.Losers,decreasing = T)
> cbind(frq.Gainers[1:10])
      [,1]
ebio   112
xgti   104
capn    55
cyhhz   41
rtla    41
hnsn    37
glco    36
wsti    36
brkk    32
gpabf   32
```

```
> cbind(frq.Losers[1:10])
        [,1]
imdz      76
biotech   35
enzn      31
bgmd      22
new       19
winners   19
pick      17
todays    17
corp      16
design    15
> palette <- brewer.pal(8,"Dark2")
> set.seed(123)
> wordcloud(words=names(frq.Gainers),
+           scale = c(3, 0.4),
+           freq = frq.Gainers,
+           min.freq = 10,
+           random.order = F,
+           colors = palette)
> wordcloud(words=names(frq.Losers),
+           scale = c(4, 0.4),
+           freq = frq.Losers,
+           min.freq = 10,
+           random.order = F,
+           colors = palette)
```

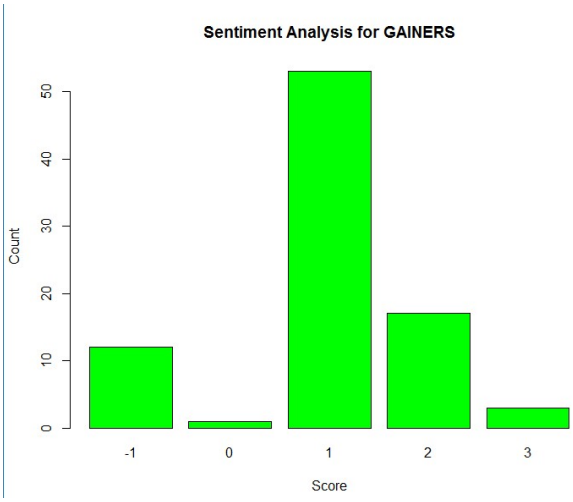


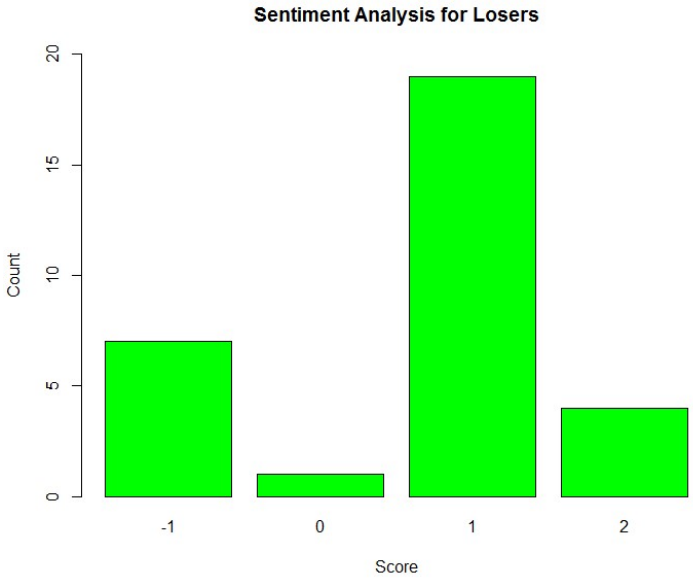
f) (20 points) Using the positive and negative word lists, compute the sentiment score (as described in the lecture) for all the tweets for each gainers (losers) set. Were the tweets about the 3 largest gainer stocks for that day characterized by a positive sentiment, and the tweets about the 3 largest loser stocks for that day characterized by a negative sentiment?

```
> sentiment <- function(text, pos.words, neg.words) {
+    text <- gsub('[[:punct:]]', '', text)
+    text <- gsub('[[:cntrl:]]', '', text)
+    text <- gsub('\\d+', '', text)
+    text <- tolower(text)
+    # split the text into a vector of words
+    words <- strsplit(text, '\\s+')
+    words <- unlist(words)
+    # find which words are positive
+    pos.matches <- match(words, pos.words)
+    pos.matches <- !is.na(pos.matches)
+    # find which words are negative
+    neg.matches <- match(words, neg.words)
+    neg.matches <- !is.na(neg.matches)
+    # calculate the sentiment score
+    p <- sum(pos.matches)
+    n <- sum(neg.matches)
+    if (p == 0 & n == 0)
+      return (NA)
+    else
+      return (p - n)
+  }
> pos.words <- scan('positive-words.txt',
+ what='character',
+ comment.char = ';')
Read 2006 items
> neg.words <- scan('negative-words.txt',
+ what='character',
+ comment.char = ';')
Read 4783 items
```

```
> # Fetch texts
> Gainers.texts <-
+    lapply(gainers,
+           function(t) {
+             iconv(t$getText(),
+                   "latin1", "ASCII", sub="")
+           })
> Losers.texts <-
+    lapply(losers,
+           function(t) {
+             iconv(t$getText(),
+                   "latin1", "ASCII", sub="")
+           })
> Gainers.scores <- sapply(Gainers.texts,
+                          sentiment,
+                          pos.words, neg.words)
> Losers.scores <- sapply(Losers.texts,
+                         sentiment,
+                         pos.words, neg.words)
```

```
> table(Gainers.scores)
Gainers.scores
-1  0  1  2  3
12  1 53 17  3
> table(Losers.scores)
Losers.scores
-1  0  1  2
 7  1 19  4
> barplot(table(Gainers.scores), main="Sentiment Analysis for GAINERS",
+         xlab="Score", ylab="Count", ylim=c(0,54), col="green")
> barplot(table(Losers.scores), main="Sentiment Analysis for Losers",
+         xlab="Score", ylab="Count", ylim=c(0,20), col="green")
```

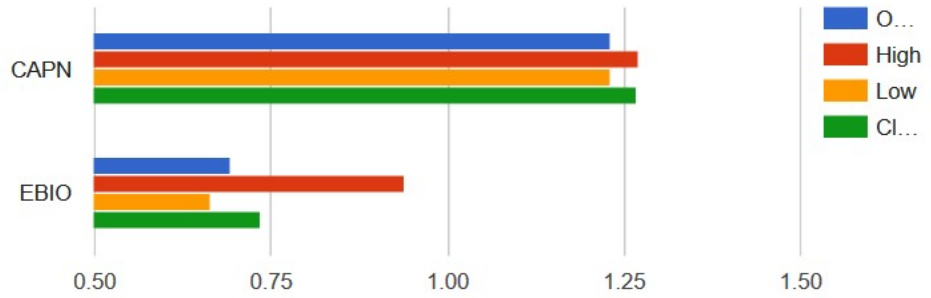


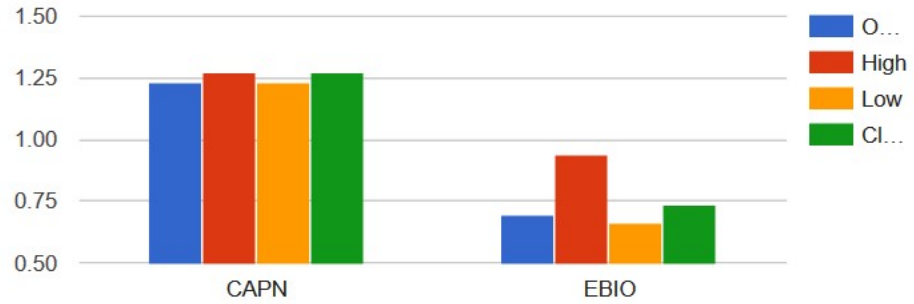


Yes, the tweets about the 3 largest gainer stocks for that day characterized by a positive sentiment, and the tweets about the 3 largest loser stocks for that day characterized by a negative sentiment. The negative sentiment for Gainers is Just 13.95% of total score and for losers it is 22.58% of total score indicating that negative sentiment is more for Losers on that day and vice-versa.

g) (For extra credit) Use *googleVis* R Package described in Module 6 to create a plot of the stock prices for the highest and lowest gainers for the day you were using for this project.

```
> getSymbols("CAPN")  #highest gainer
[1] "CAPN"
Warning message:
In download.file(paste(yahoo.URL, "s=", Symbols.name, "&a=", from.m,  :
  downloaded length 15399 != reported length 200
> getSymbols("EBIO")  #lowest gainer
[1] "EBIO"
Warning message:
In download.file(paste(yahoo.URL, "s=", Symbols.name, "&a=", from.m,  :
  downloaded length 25484 != reported length 200
> View(CAPN)
> CAPN.dframe = data.frame(CAPN['2016-04-20'])
> EBIO.dframe = data.frame(EBIO['2016-04-20'])
> Open = c(as.numeric(CAPN.dframe$CAPN.Open),as.numeric(EBIO.dframe$EBIO.Open)
)
> High = c(as.numeric(CAPN.dframe$CAPN.High),as.numeric(EBIO.dframe$EBIO.High)
)
> Low = c(as.numeric(CAPN.dframe$CAPN.Low),as.numeric(EBIO.dframe$EBIO.Low))
> Close = c(as.numeric(CAPN.dframe$CAPN.Close),as.numeric(EBIO.dframe$EBIO.Clo
se))
> Volume = c(as.numeric(CAPN.dframe$CAPN.Volume),as.numeric(EBIO.dframe$EBIO.V
olume))
> Adjusted = c(as.numeric(CAPN.dframe$CAPN.Adjusted),as.numeric(EBIO.dframe$EB
IO.Adjusted))
> stocks.data = data.frame(Stocks=c("CAPN","EBIO"),Open,High,Low,Close)
> chart1 <- gvisBarChart(stocks.data)
> plot(chart1)
> print(chart1, tag="chart", file="chart1.html")
> chart2 <- gvisColumnChart(stocks.data)
> plot(chart2)
> print(chart2, tag="chart", file="chart2.html")
```





Code:

```
# A. Get Tweets for Gainers
t.CAPN <- searchTwitter('$CAPN', n = 100)
t.XGTI <- searchTwitter('$XGTI', n = 100)
t.EBIO <- searchTwitter('$EBIO', n = 100)
save(t.CAPN, file="t.CAPN")
save(t.XGTI, file="t.XGTI")
save(t.EBIO, file="t.EBIO")
gainers <- c(t.CAPN,t.XGTI,t.EBIO)

#Get Tweets for Losers
t.BGMD <- searchTwitter('$BGMD', n = 100)
t.ENZN <- searchTwitter('$ENZN', n = 100)
t.IMDZ <- searchTwitter('$IMDZ', n = 100)
save(t.BGMD, file="t.BGMD")
save(t.ENZN, file="t.ENZN")
save(t.IMDZ, file="t.IMDZ")
losers <- c(t.BGMD,t.ENZN,t.IMDZ)

#B) Create corpora for tweets
get.corpus <- function (tweets)
{
 tweets.text <- lapply(tweets, function(t) {t$getText()})
 data.source <- VectorSource(tweets.text)
 data.corpus <- Corpus(data.source)
}

data.corpus.gainers <- get.corpus(gainers)
writeCorpus(data.corpus.gainers,path = "data.corpus1")

data.corpus.losers <- get.corpus(losers)
writeCorpus(data.corpus.losers,path= "data.corpus2")

#C Pre-Processing
trans.corpus <- function (data.corpus) {
 data.corpus <- tm_map(data.corpus,
content_transformer(removePunctuation))
 data.corpus <- tm_map(data.corpus, content_transformer(tolower))
 data.corpus <- tm_map(data.corpus, removeNumbers)
 english.stopwords <- stopwords("en")
 removeURL <- function(x) gsub("http[[:alnum:]]*", "", x)
 data.corpus <- tm_map(data.corpus, content_transformer(removeURL))
```

```
  data.corpus <-
tm_map(data.corpus,content_transformer(removeWords),english.stopwords)
  data.corpus <- tm_map(data.corpus,content_transformer(stemDocument))
  data.corpus <- tm_map(data.corpus,content_transformer(stripWhitespace))

}

data.corpus.gainers <- trans.corpus(data.corpus.gainers)
data.corpus.losers <- trans.corpus(data.corpus.losers)

#D TDM for each corpus
tdm.Gainers <- TermDocumentMatrix(data.corpus.gainers)
tdm.Losers <- TermDocumentMatrix(data.corpus.losers)

save(tdm.Gainers, file = "tdm1")
save(tdm.Losers, file = "tdm2")

#E Frequent terms
findFreqTerms(tdm.Gainers, lowfreq=15)
findFreqTerms(tdm.Losers, lowfreq=15)
m.Gainers <-as.matrix(tdm.Gainers)
m.Losers <- as.matrix(tdm.Losers)
frq.Gainers <- rowSums(m.Gainers)
frq.Gainers <- sort(frq.Gainers,decreasing = T)
frq.Losers <- rowSums(m.Losers)
frq.Losers <- sort(frq.Losers,decreasing = T)
cbind(frq.Gainers[1:10])
cbind(frq.Losers[1:10])

palette <- brewer.pal(8,"Dark2")
set.seed(123)
wordcloud(words=names(frq.Gainers),
        scale = c(3, 0.4),
        freq = frq.Gainers,
        min.freq = 10,
        random.order = F,
        colors = palette)
wordcloud(words=names(frq.Losers),
        scale = c(4, 0.4),
        freq = frq.Losers,
        min.freq = 10,
        random.order = F,
        colors = palette)
```

```
#F Sentiment Analysis
sentiment <- function(text, pos.words, neg.words) {
  text <- gsub('[[:punct:]]', '', text)
  text <- gsub('[[:cntrl:]]', '', text)
  text <- gsub('\\d+', '', text)
  text <- tolower(text)
  # split the text into a vector of words
  words <- strsplit(text, '\\s+')
  words <- unlist(words)
  # find which words are positive
  pos.matches <- match(words, pos.words)
  pos.matches <- !is.na(pos.matches)
  # find which words are negative
  neg.matches <- match(words, neg.words)
  neg.matches <- !is.na(neg.matches)
  # calculate the sentiment score
  p <- sum(pos.matches)
  n <- sum(neg.matches)
  if (p == 0 & n == 0)
   return (NA)
  else
   return (p - n)
 }
pos.words <- scan('positive-words.txt',
                  what='character',
                  comment.char = ';')
neg.words <- scan('negative-words.txt',
                  what='character',
                  comment.char = ';')
# Fetch texts
Gainers.texts <-
  lapply(gainers,
         function(t) {
           iconv(t$getText(),
                 "latin1", "ASCII", sub="")
         })

Losers.texts <-
  lapply(losers,
         function(t) {
           iconv(t$getText(),
                 "latin1", "ASCII", sub="")
         })
```

```
# Scores
Gainers.scores <- sapply(Gainers.texts,
                         sentiment,
                         pos.words, neg.words)

Losers.scores <- sapply(Losers.texts,
                        sentiment,
                        pos.words, neg.words)


table(Gainers.scores)

table(Losers.scores)

barplot(table(Gainers.scores), main="Sentiment Analysis for GAINERS",
      xlab="Score", ylab="Count", ylim=c(0,54), col="green")
barplot(table(Losers.scores), main="Sentiment Analysis for Losers",
      xlab="Score", ylab="Count", ylim=c(0,20), col="green")

# G GogleVis
library(quantmod)
library(googleVis)

getSymbols("CAPN")
getSymbols("EBIO")

CAPN.dframe = data.frame(CAPN['2016-04-20'])
EBIO.dframe = data.frame(EBIO['2016-04-20'])

Open =
c(as.numeric(CAPN.dframe$CAPN.Open),as.numeric(EBIO.dframe$EBIO.Open))
High =
c(as.numeric(CAPN.dframe$CAPN.High),as.numeric(EBIO.dframe$EBIO.High))
Low =
c(as.numeric(CAPN.dframe$CAPN.Low),as.numeric(EBIO.dframe$EBIO.Low))
Close =
c(as.numeric(CAPN.dframe$CAPN.Close),as.numeric(EBIO.dframe$EBIO.Close))
Volume =
c(as.numeric(CAPN.dframe$CAPN.Volume),as.numeric(EBIO.dframe$EBIO.Volum
e))
```

```
Adjusted =
c(as.numeric(CAPN.dframe$CAPN.Adjusted),as.numeric(EBIO.dframe$EBIO.Adju
sted))

stocks.data = data.frame(Stocks=c("CAPN","EBIO"),Open,High,Low,Close)


chart1 <- gvisBarChart(stocks.data)
plot(chart1)
print(chart1, tag="chart", file="chart1.html")
chart2 <- gvisColumnChart(stocks.data)
plot(chart2)
print(chart2,tag = "chart",file="chart2.html")
```