STAT240 Lab7

Jiajun Zhang

March 10, 2019

```
library(ROAuth)
library(twitteR)
```

```
setup_twitter_oauth(consumer_key, consumer_secret, access_token, access_secret)
```

```
## [1] "Using direct authentication"
```

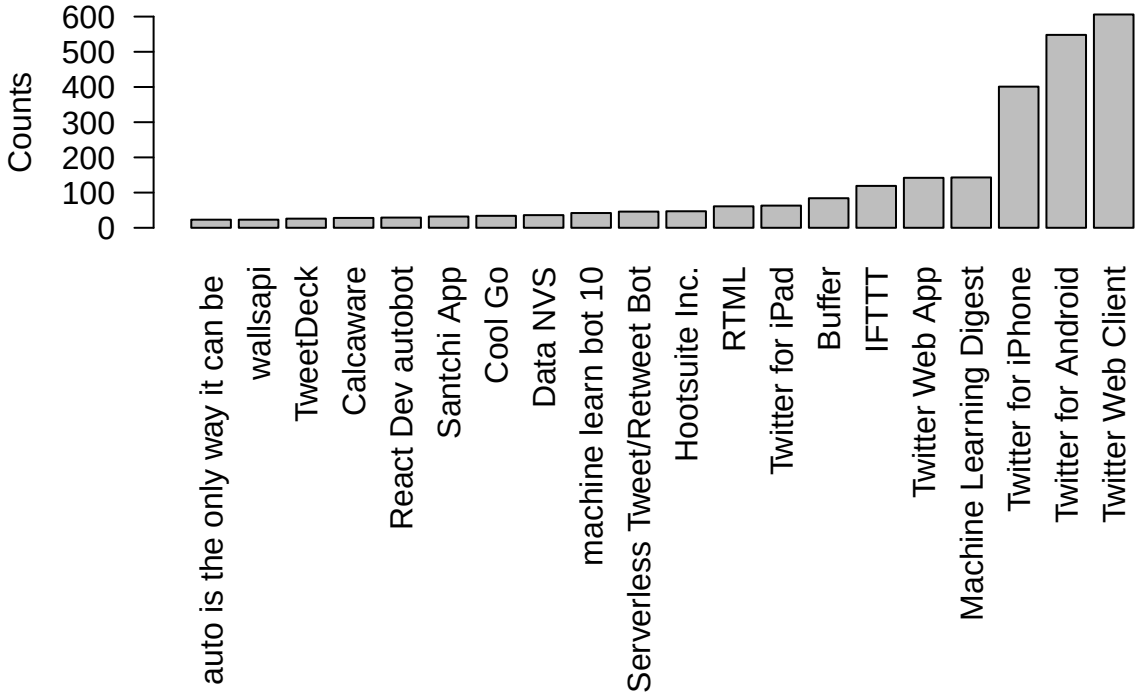
Question1

```
dsTweets = searchTwitter(searchString = "#datascience", n = 3200, lang = "en")

ds_df=twListToDF(dsTweets)
#To see the structure
head(ds_df$statusSource, 4)
```

```
## [1] "<a href=\"http://twitter.com/download/android\" rel=\"nofollow\">Twitter for Android</a>"
## [2] "<a href=\"https://dlvrit.com/\" rel=\"nofollow\">dlvr.it</a>"
## [3] "<a href=\"https://dlvrit.com/\" rel=\"nofollow\">dlvr.it</a>"
## [4] "<a href=\"http://twitter.com/download/android\" rel=\"nofollow\">Twitter for Android</a>"
```

```
Source=ds_df$statusSource
remove_http=gsub("^.*?>", "", Source)
platform=gsub("</a>", "", remove_http)
# platform
par(mar=c(13,4,4,2))
barplot(tail(sort(table(platform)),20), las=2, ylim=c(0, 600),
        main="Top 20 User Platforms of #datascience Tweets", ylab="Counts")
```


Top 20 User Platforms of #datascience Tweets
Counts
600
500
400
300
200
100
0
auto is the only way it can be
wallsapi
TweetDeck
Calcaware
React Dev autobot
Santchi App
Cool Go
Data NVS
machine learn bot 10
Serverless Tweet/Retweet Bot
Hootsuite Inc.
RTML
Twitter for iPad
Buffer
IFTTT
Twitter Web App
Machine Learning Digest
Twitter for iPhone
Twitter for Android
Twitter Web Client

Question2

```
BestBuytweet=userTimeline("BestBuy", n=500)
BestBuy_df=twListToDF(BestBuytweet)
nrow(BestBuy_df)
```

```
## [1] 500
```

```
BestBuy.text=BestBuy_df$text
BestBuy.text = gsub("(RT|via)((?:\\b\\W*@\\w+)+)", "", BestBuy.text)
BestBuy.text = gsub("@\\w+", "", BestBuy.text)
BestBuy.text = gsub("(?!')[[:punct:]]", "", BestBuy.text, perl = T)
BestBuy.text = gsub("[[:cntrl:]]", "", BestBuy.text)
BestBuy.text = gsub("[[:digit:]]", "", BestBuy.text)
BestBuy.text = gsub("http\\w+", "", BestBuy.text)
BestBuy.text = gsub("^\\s+|\\s+$", "", BestBuy.text)
BestBuy.text = tolower(BestBuy.text)
BestBuy.text = gsub("http\\w+", "", BestBuy.text)
BestBuy.text = gsub("[ \t]{2,}", " ", BestBuy.text)
BestBuy.text = gsub("^\\s+|\\s+$", "", BestBuy.text)
word.list = strsplit(BestBuy.text, " ")
words = unlist(word.list)
library(tm)
```

```
## Loading required package: NLP
```

```
words = words[!words %in% tm::stopwords(kind = "english")]
library(wordcloud)
```

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

```
wordcloud(names(table(words)), table(words), min.freq=10,
          colors=rainbow(8), random.order = FALSE)
```


help
going
new
always
happy
love
right
good
like
can
blue
buy
day
best
amazing
get
us
way
now
great
thanks
will
just
one
got
looking
say
gaming
enjoy
perfect

Question3

a)

```
#Selected 3 Oscar nominees are Black Panther, Green Book, and A Star is Born
BP=searchTwitter(searchString = "#BlackPanther", n = 2000, lang = "en")
GB=searchTwitter(searchString = "#GreenBook", n = 2000, lang = "en")
STAR=searchTwitter(searchString = "#AStarIsBorn", n = 2000, lang = "en")
c( Black_Panther=nrow(twListToDF(BP)), Green_Book=nrow(twListToDF(GB)),
   A_Star_Is_Born=nrow(twListToDF(STAR)) )
```

```
##  Black_Panther     Green_Book A_Star_Is_Born
##           2000           2000           2000
```

```
BP_df=twListToDF(BP)
GB_df=twListToDF(GB)
STAR_df=twListToDF(STAR)

#There are many duplicated retweets, and we want only the unique tweets
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:twitteR':
##
##     id, location
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
BP_df1=BP_df%>%filter(isRetweet==FALSE)
GB_df1=GB_df%>%filter(isRetweet==FALSE)
STAR_df1=STAR_df%>%filter(isRetweet==FALSE)
#The number of unique tweets for each movie
c( Black_Panther=nrow(BP_df1), Green_Book=nrow(GB_df1),
   A_Star_Is_Born=nrow(STAR_df1) )
```

```
##  Black_Panther     Green_Book A_Star_Is_Born
##            752            895            696
```

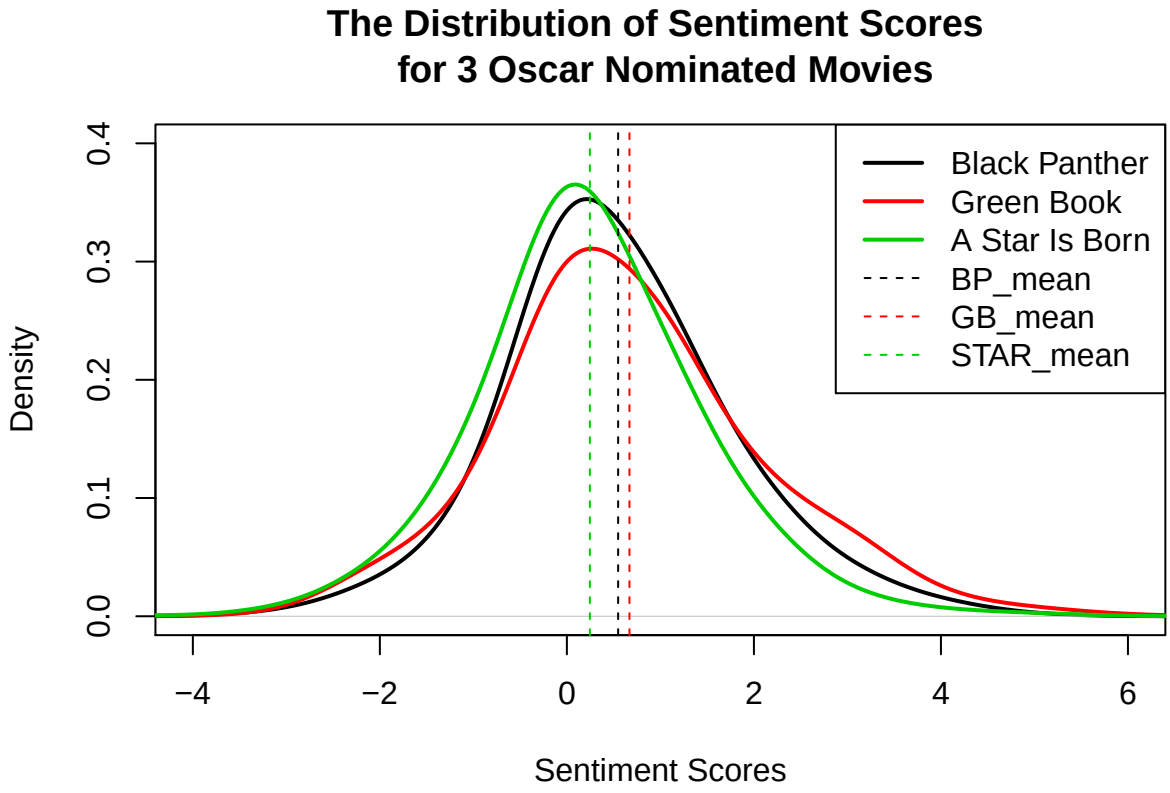
```
pos = scan("positive-words.txt", what = "character", comment.char = ";")
neg = scan("negative-words.txt", what = "character", comment.char = ";")
getSentimentScore = function(tweet_text, pos, neg) {
  sentence = gsub("(RT|via)((?:\\b\\W*@\\w+)+)", "", tweet_text)
  sentence = gsub("@\\w+", "", sentence)
  sentence = gsub("[[:punct:]]", "", sentence)
  sentence = gsub("[[:cntrl:]]", "", sentence)
  sentence = gsub("[[:digit:]]", "", sentence)
  sentence = gsub("http\\w+", "", sentence)
  sentence = gsub("^\\s+|\\s+$", "", sentence)
```

```
  # sentence = iconv(sentence, "ASCII", "UTF-8", sub = "")
  sentence = tolower(sentence)
  word.list = strsplit(sentence, " ")
  score = numeric(length(word.list))
  for (i in 1:length(word.list)) {
    pos.matches = match(word.list[[i]], pos)
    neg.matches = match(word.list[[i]], neg)
    pos.matches = !is.na(pos.matches)
    neg.matches = !is.na(neg.matches)
    score[i] = sum(pos.matches) - sum(neg.matches)
  }
  return(score)
}
```

b)

```
BP_Score=getSentimentScore(BP_df1$text, pos, neg)
GB_Score=getSentimentScore(GB_df1$text, pos, neg)
STAR_Score=getSentimentScore(STAR_df1$text, pos, neg)

plot(density(BP_Score, bw=0.6), xlim=c(-4,6), col=1, ylim=c(0,0.4), lwd=2,
     main="The Distribution of Sentiment Scores \n for 3 Oscar Nominated Movies",
     xlab="Sentiment Scores")
lines(density(GB_Score, bw=0.6), col=2, lwd=2)
lines(density(STAR_Score, bw=0.6), col=3, lwd=2)
abline(v=mean(BP_Score), col=1, lty=2)
abline(v=mean(GB_Score), col=2, lty=2)
abline(v=mean(STAR_Score), col=3, lty=2)
legend("topright", col=c(1,2,3,1,2,3),
       lwd=c(2,2,2,1,1,1), lty=c(1,1,1,2,2,2),
       c("Black Panther", "Green Book", "A Star Is Born",
         "BP_mean", "GB_mean", "STAR_mean"))
```



c)

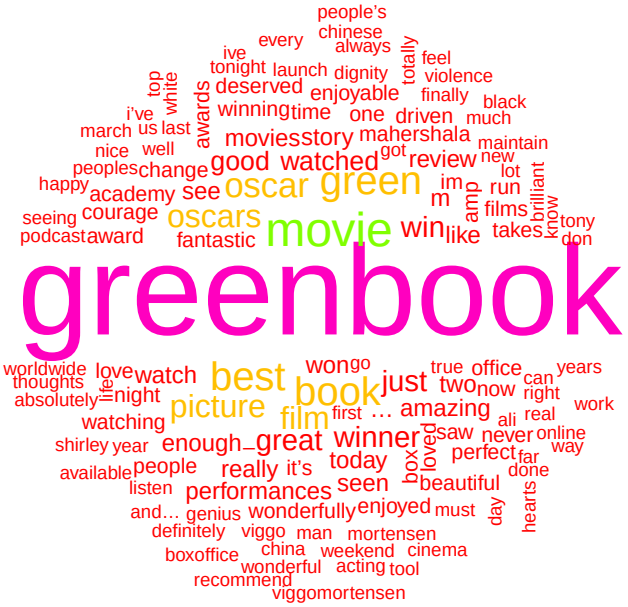
```
c( BP_positive=sum(BP_Score>0)/length(BP_Score),
   GB_positive=sum(GB_Score>0)/length(GB_Score),
   STAR_positive=sum(STAR_Score>0)/length(STAR_Score) )
```

```
##   BP_positive   GB_positive STAR_positive
##     0.4587766     0.4994413     0.3534483
```

- Green Book has the highest proportion of positive tweets.

```
GB_pos=GB_df1$text[GB_Score>0]
GB_pos = gsub("(RT|via)((?:\\b\\W*@\\w+)+)", "", GB_pos)
GB_pos = gsub("@\\w+", "", GB_pos)
GB_pos = gsub("[[:punct:]]", "", GB_pos)
GB_pos = gsub("[[:cntrl:]]", "", GB_pos)
GB_pos = gsub("[[:digit:]]", "", GB_pos)
GB_pos = gsub("http\\w+", "", GB_pos)
GB_pos = gsub("^\\s+|\\s+$", "", GB_pos)
GB_pos = tolower(GB_pos)
GB_pos = gsub("http\\w+", "", GB_pos)
GB_pos = gsub("[ \t]{2,}", " ", GB_pos)
GB_pos = gsub("^\\s+|\\s+$", "", GB_pos)
```

```
word.list = strsplit(GB_pos, " ")
words = unlist(word.list)
library(tm)
words = words[!words %in% tm::stopwords(kind = "english")]
library(wordcloud)
wordcloud(names(table(words)), table(words), min.freq=6,
          colors=rainbow(8), random.order = FALSE)
```

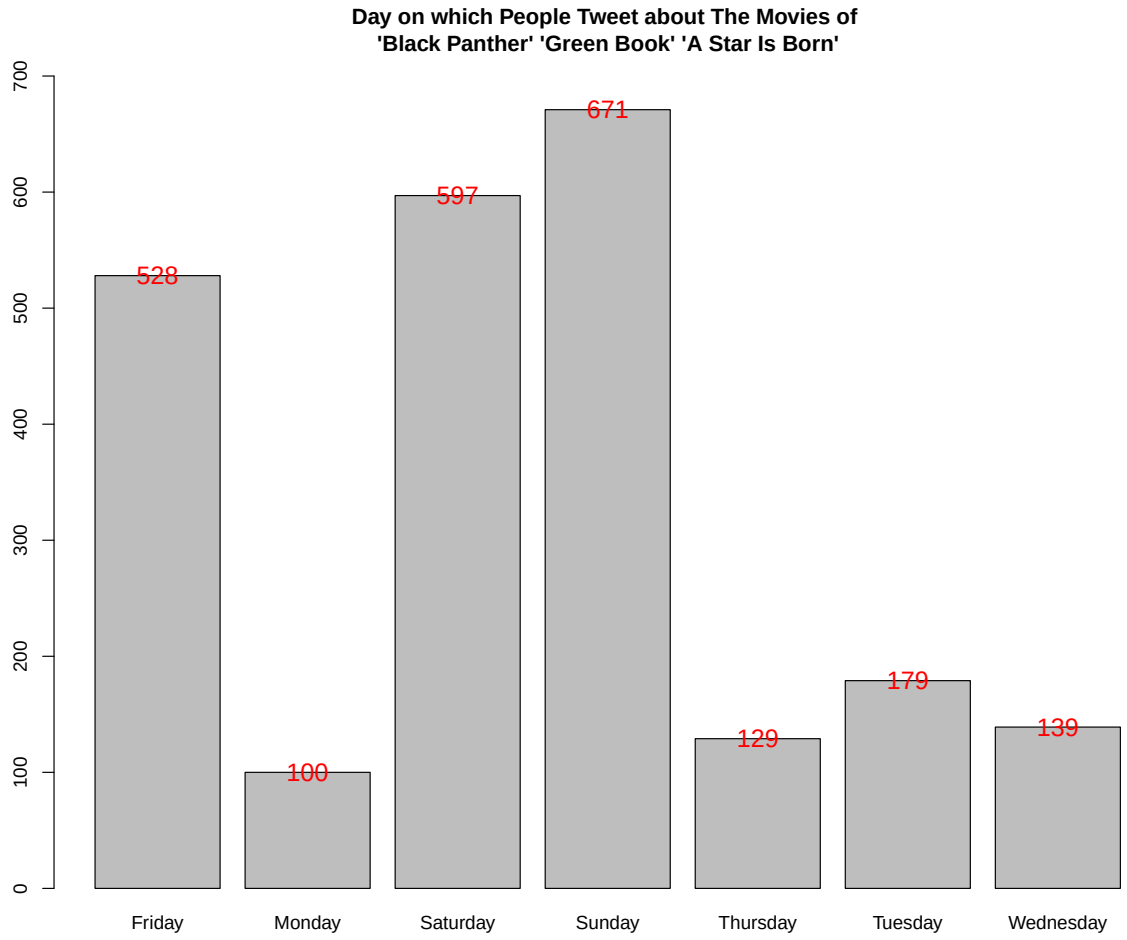


d)

```
all_df=rbind( cbind(BP_df1, Score=BP_Score),
              cbind(GB_df1, Score=GB_Score),
              cbind(STAR_df1, Score=STAR_Score) )
table(weekdays(all_df[, "created"]))
```

```
##
##    Friday    Monday  Saturday    Sunday  Thursday   Tuesday Wednesday
##       528       100       597       671       129       179       139
```

```
all_df1=as.data.frame(table(weekdays(all_df[, "created"])))
x_df=barplot(table((weekdays(all_df[, "created"]))), ylim=c(0,700),
        main="Day on which People Tweet about The Movies of \n 'Black Panther' 'Green Book' 'A Star Is
text(x=x_df, y=all_df1$Freq, label=all_df1$Freq,
     cex = 1.3, col = "red")
```



- Using a plot can be misleading since we want the actual number of Tweets on specific day of week. Providing a plot can be regarded as a time-series timeplot; however, we are focusing on the actual number distribution but not the trends.