# MA615 Final Project Presentation

*Xiaoqian Xue*

*12/18/2017*

# I Introduction

Starbucks Corporation, founded in 1971, has been long considered the main representative of “second wave coffee”, distinguishing itself from other coffee-serving venues in the US by taste, quality and customer experience.Their success and customers’ addiction with the brand have made many scholars wonder the reason. Some scholars have argued that the popularity of Starbucks is probably not due to its quality but because of customer’s buying experience and its marketing strategy.

In order to figure out what Internet users, specifically Twitter users, are talking about Starbucks and how they are feeling about Starbucks brands, I planned to collect tweets including hashtags “#starbucks” and its three major hashtags.

This project contains three parts:

1. The most frequent words twitter users mention when they talk about Starbucks and plot them to see their distributions.
2. The sentiment scores of twitter users when they talk about Starbucks tweets and the statistical relationship between the scores and the popularity of data (the popularity of data is measured by the retweet numbers) using Hypothesis Test and Anova Table. I aim to compare the difference in the user’s sentiment score across different hashtags.
3. The visualization of sentiments scores towards Starbucks among those three activities in different location (Los Angeles, Seattle, New York, Boston) integreated with shiny application to create an interactive map on tweet popularity.

# II Dataset Summary

In this project, all the data come from Twitter. The sample of this project includes 2000 tweets mentioning starbucks, 3000 tweets about starbucks’ events. (Specifically, 1000 tweets about Starbucks For Life, 1000 tweets about Starbucks At Home, 1000 tweets about Starbucks Give Good. )

Since there are few users including its brand perceptions, such as starbucks eco-friendliness, its nutrition or its taste in their tweets. I chose the searchTwitter() function to find the hashtags promoted by Starbucks to represent Starbucks’ market segmentation.

```
library(devtools)
library(twitteR)
library(ROAuth)
api_key <- "zwmo2nKieK0RGVoFMP9KfGci9"
api_secret <- "Az0OyiiUDyWJlCwnZtEBJxRKLNJIZ4FqOPyUvyDNqUM7Ko2lUr"
access_token <- "2809402781-RQuIDNFLL1dg5i3xETd1W3t8Kol9ZVqufMrgsRf"
access_token_secret <- "q5s33WockcM1JdwLyxJmQP9zGv3B0ilzYXmnYF8TMAN0q"
setup_twitter_oauth(api_key, api_secret, access_token, access_token_secret)
```

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

```
## Warning in strptime(x, fmt, tz = "GMT"): unknown timezone 'zone/tz/2017c.
## 1.0/zoneinfo/America/New_York'
```

```
# search "Starbucks" on Twitter
s <- searchTwitter("@starbucks OR #starbucks", n=2000, lang="en")
# search hashtags "StarbucksForLife" on Twitter
s2 <- searchTwitter("#starbucksforlife", n=1000, lang="en")
# search hashtags "StarbucksAtHome" on Twitter
s3 <- searchTwitter("#starbucksathome", n=1000, lang="en")
# search hashtags "StarbucksHoliday" on Twitter
s4 <- searchTwitter("#givegood", n=1000, lang="en")
```

Starbucks For Life Event is a promotion holding by Starbucks.By promoting this event, Starbucks aim to provide loyatly customers rewards and further market their new holiday products. Starbucks At Home is a hashtag promoting by Starbucks official to market their coffee-brewing products, for example, their convenient K-Cup pods, VIA Instant. Coffee at home prompts customers to buy their products so that they can make their own coffee at home and aims to provide customers a perception of convenience and warm. Starbucks Give Good is another hashtag and promotes the idea of giving. Give Good project aims to celebrate communities and local heroes with $1 million worth of Starbucks Cards throughout the month of December.It gives customers a perception of good and giving.

After searching tweets, cleaning it and obtaining locations' information, I get 3 data frames for each topic and a total data frames with all the information including screenname, text, retweet count, favorite count, score, absolute score, longitude, latitude. For convenience, I will save those frames and load those frames to conduct further analysis. (The code for all the cleaning and combining process can be found in the sweave file)

```
data <- read.csv("starbucks_info.csv",row.names = 1)
data1 <- read.csv("life_info.csv",row.names = 1)
data2 <- read.csv("home_info.csv",row.names = 1)
data3 <- read.csv("give_info.csv",row.names = 1)
total <- read.csv("total.csv",row.names = 1)
```

To have a general understanding on what are the most popular words that people use in tweets to express regard Starbucks brands and its marketing, I obtain histograms to show the frquent words and also wordclouds under each topic.

```
library(tm)
```

```
## Warning: package 'tm' was built under R version 3.4.3
```

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

```
library(wordcloud)
```

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

```
tweetFrame <- twListToDF(s)
tweetFrame1 <- twListToDF(s2)
tweetFrame2 <- twListToDF(s3)
tweetFrame3 <- twListToDF(s4)
# build a corpus, and specify the source to be character vectors
t <- Corpus(VectorSource(tweetFrame$text))
t1 <- Corpus(VectorSource(tweetFrame1$text))
t2 <- Corpus(VectorSource(tweetFrame2$text))
t3 <- Corpus(VectorSource(tweetFrame3$text))
# remove URLs
removeURL <- function(x) gsub("http[^[:space:]]*", "", x)
t <- tm_map(t, content_transformer(removeURL))
t1 <- tm_map(t1, content_transformer(removeURL))
t2 <- tm_map(t2, content_transformer(removeURL))
t3 <- tm_map(t3, content_transformer(removeURL))
# remove anything other than English letters or space remove
removeNumPunct <- function(x) gsub("[^[:alpha:][:space:]]*", "", x)
t <- tm_map(t, content_transformer(removeNumPunct))
t1 <- tm_map(t1, content_transformer(removeNumPunct))
t2 <- tm_map(t2, content_transformer(removeNumPunct))
t3 <- tm_map(t3, content_transformer(removeNumPunct))
# remove stopwords
myStopwords <- c(setdiff(stopwords('english'), c("starbucks","life","home","give","go
od")))
t <- tm_map(t, removeWords, myStopwords)
t1 <- tm_map(t1, removeWords, myStopwords)
t2 <- tm_map(t2, removeWords, myStopwords)
t3 <- tm_map(t3, removeWords, myStopwords)
# remove extra whitespace
t <- tm_map(t, stripWhitespace)
t1 <- tm_map(t1, stripWhitespace)
t2 <- tm_map(t2, stripWhitespace)
t3 <- tm_map(t3, stripWhitespace)
# convert to lower case
t <- tm_map(t, content_transformer(tolower))
t1 <- tm_map(t1, content_transformer(tolower))
t2 <- tm_map(t2, content_transformer(tolower))
t3 <- tm_map(t3, content_transformer(tolower))
# Build Term Document Matrix
tdm <- TermDocumentMatrix(t,control = list(wordLengths = c(1, Inf)))
tdm1 <- TermDocumentMatrix(t1,control = list(wordLengths = c(1, Inf)))
tdm2 <- TermDocumentMatrix(t2,control = list(wordLengths = c(1, Inf)))
tdm3 <- TermDocumentMatrix(t3,control = list(wordLengths = c(1, Inf)))
tdm
```

```
## <<TermDocumentMatrix (terms: 4320, documents: 2000)>>
## Non-/sparse entries: 21090/8618910
## Sparsity           : 100%
## Maximal term length: 37
## Weighting          : term frequency (tf)
```

```
tdm1
```

```
## <<TermDocumentMatrix (terms: 736, documents: 1000)>>
## Non-/sparse entries: 11010/724990
## Sparsity           : 99%
## Maximal term length: 26
## Weighting          : term frequency (tf)
```

```
tdm2
```

```
## <<TermDocumentMatrix (terms: 65, documents: 1000)>>
## Non-/sparse entries: 17907/47093
## Sparsity           : 72%
## Maximal term length: 17
## Weighting          : term frequency (tf)
```

```
tdm3
```

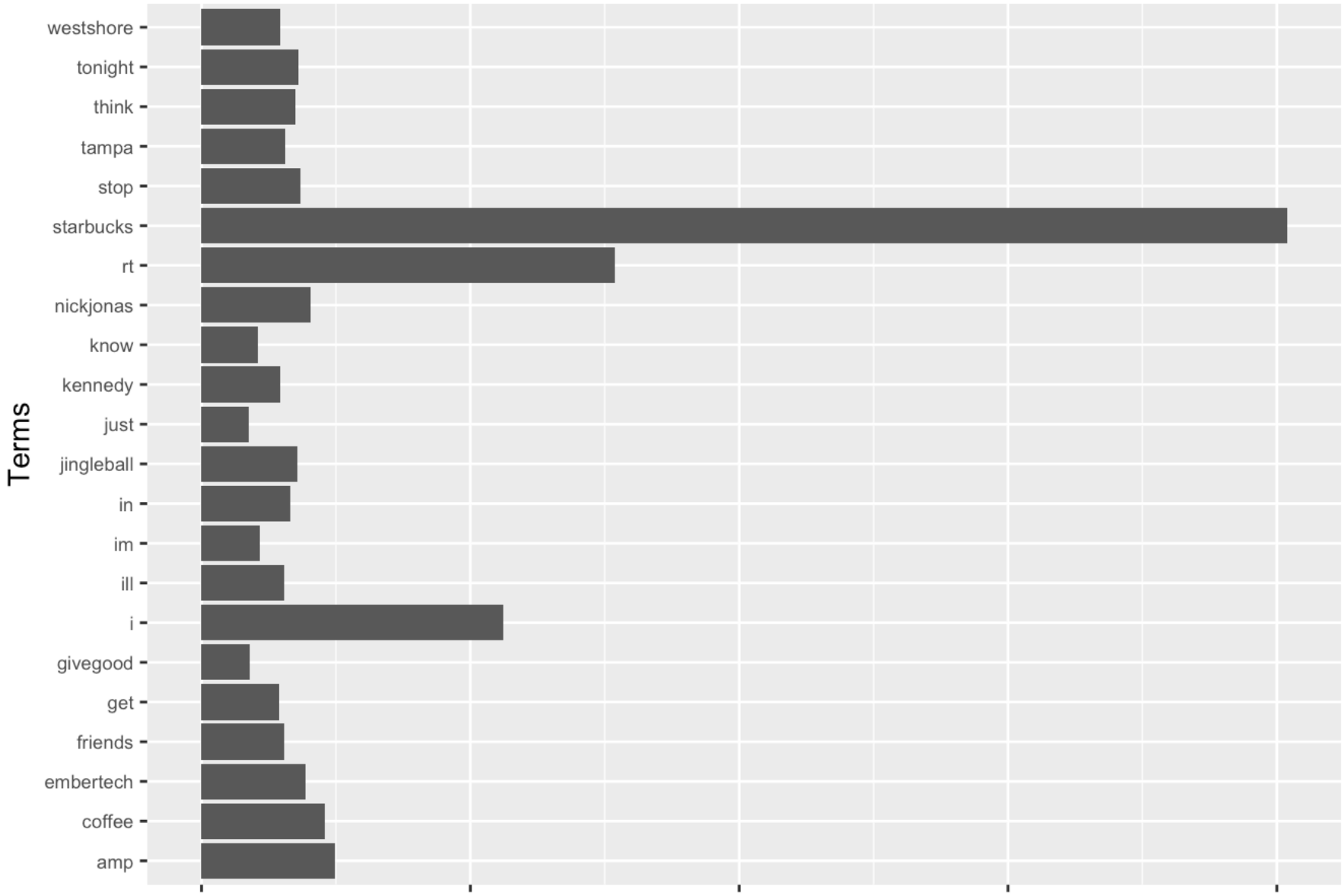
```
## <<TermDocumentMatrix (terms: 1207, documents: 1000)>>
## Non-/sparse entries: 8353/1198647
## Sparsity           : 99%
## Maximal term length: 19
## Weighting          : term frequency (tf)
```

```
# inspect frequent words
freq.terms <- findFreqTerms(tdm, lowfreq = 80)
freq.terms <- findFreqTerms(tdm1, lowfreq = 50)
freq.terms <- findFreqTerms(tdm2, lowfreq = 5)
freq.terms <- findFreqTerms(tdm3, lowfreq = 50)
term.freq <- rowSums(as.matrix(tdm))
term1.freq <- rowSums(as.matrix(tdm1))
term2.freq <- rowSums(as.matrix(tdm2))
term3.freq <- rowSums(as.matrix(tdm3))
term.freq <- subset(term.freq, term.freq >= 80)
term1.freq <- subset(term1.freq, term1.freq >= 50)
term2.freq <- subset(term2.freq, term2.freq >= 5)
term3.freq <- subset(term3.freq, term3.freq >= 50)
df <- data.frame(term = names(term.freq), freq = term.freq)
df1 <- data.frame(term = names(term1.freq), freq = term1.freq)
df2 <- data.frame(term = names(term2.freq), freq = term2.freq)
df3 <- data.frame(term = names(term3.freq), freq = term3.freq)
library(ggplot2)
```

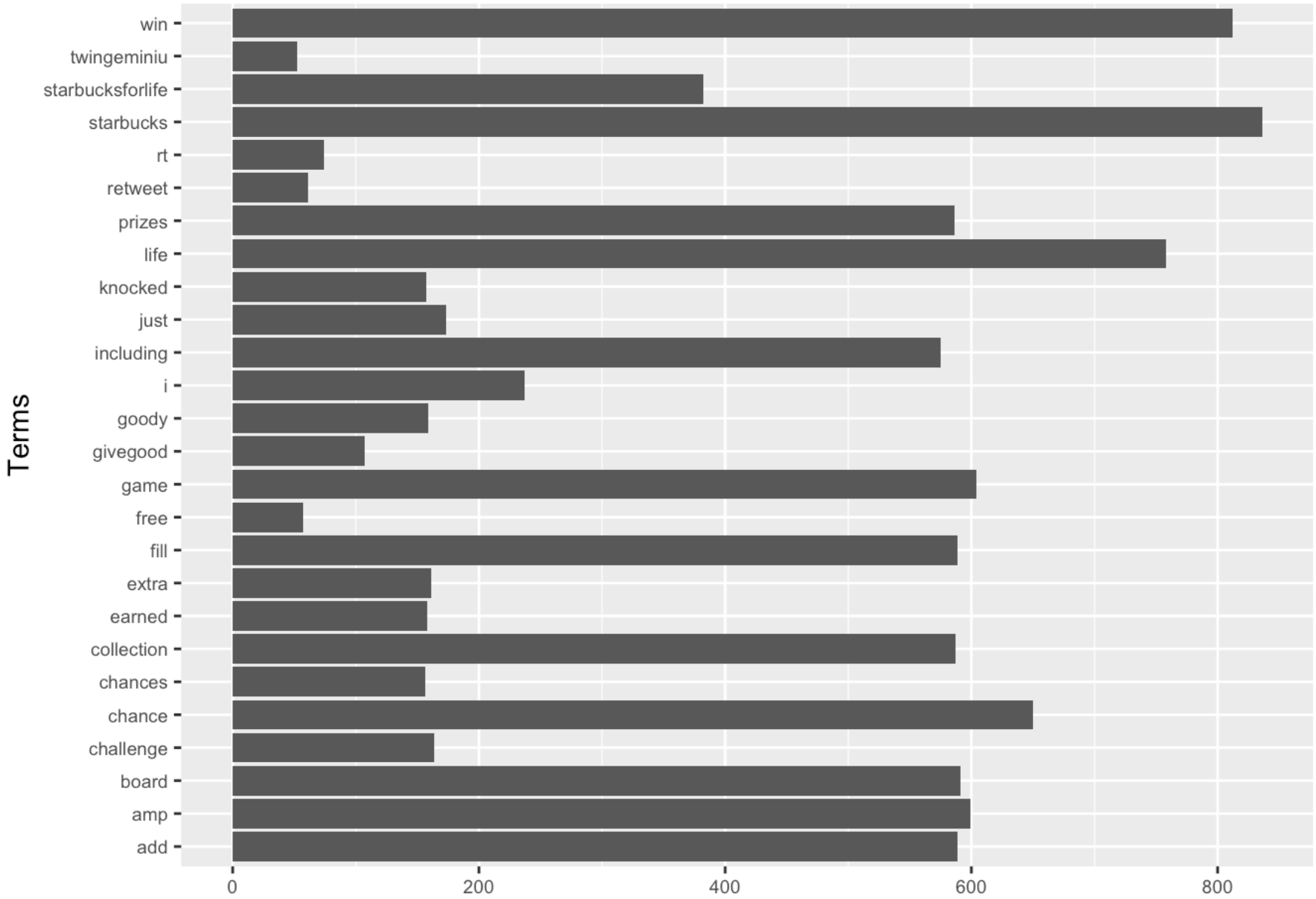
```
##
## Attaching package: 'ggplot2'
```

```
## The following object is masked from 'package:NLP':
##
##     annotate
```

```
ggplot(df, aes(x=term, y=freq)) + geom_bar(stat = "identity") + xlab ("Terms") + ylab
("Count") + coord_flip() + theme(axis.text=element_text(size=7))
```

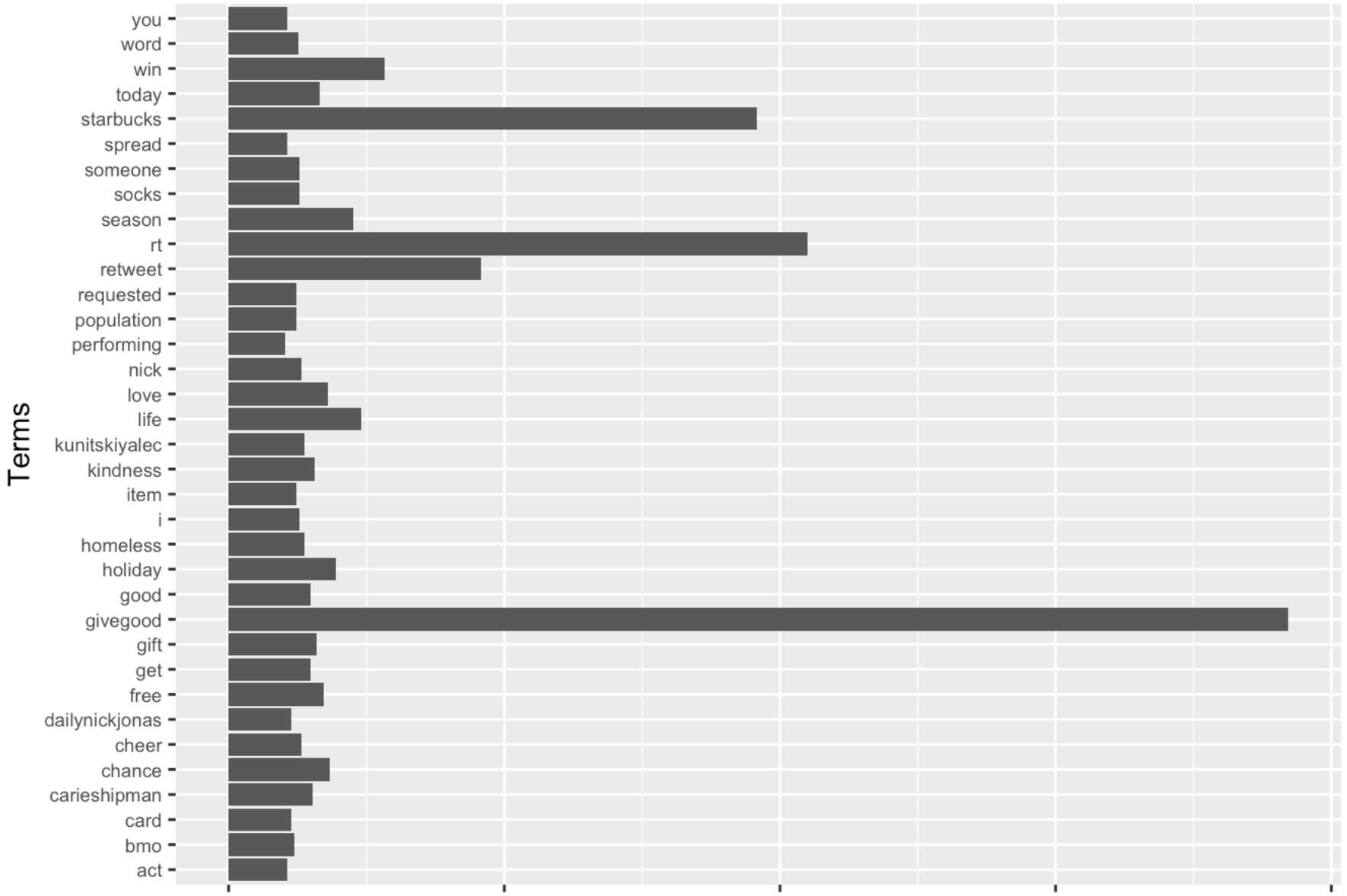



```
ggplot(df1, aes(x=term, y=freq)) + geom_bar(stat = "identity") + xlab ("Terms") + yla
b("Count") + coord_flip() + theme(axis.text=element_text(size=7))
```

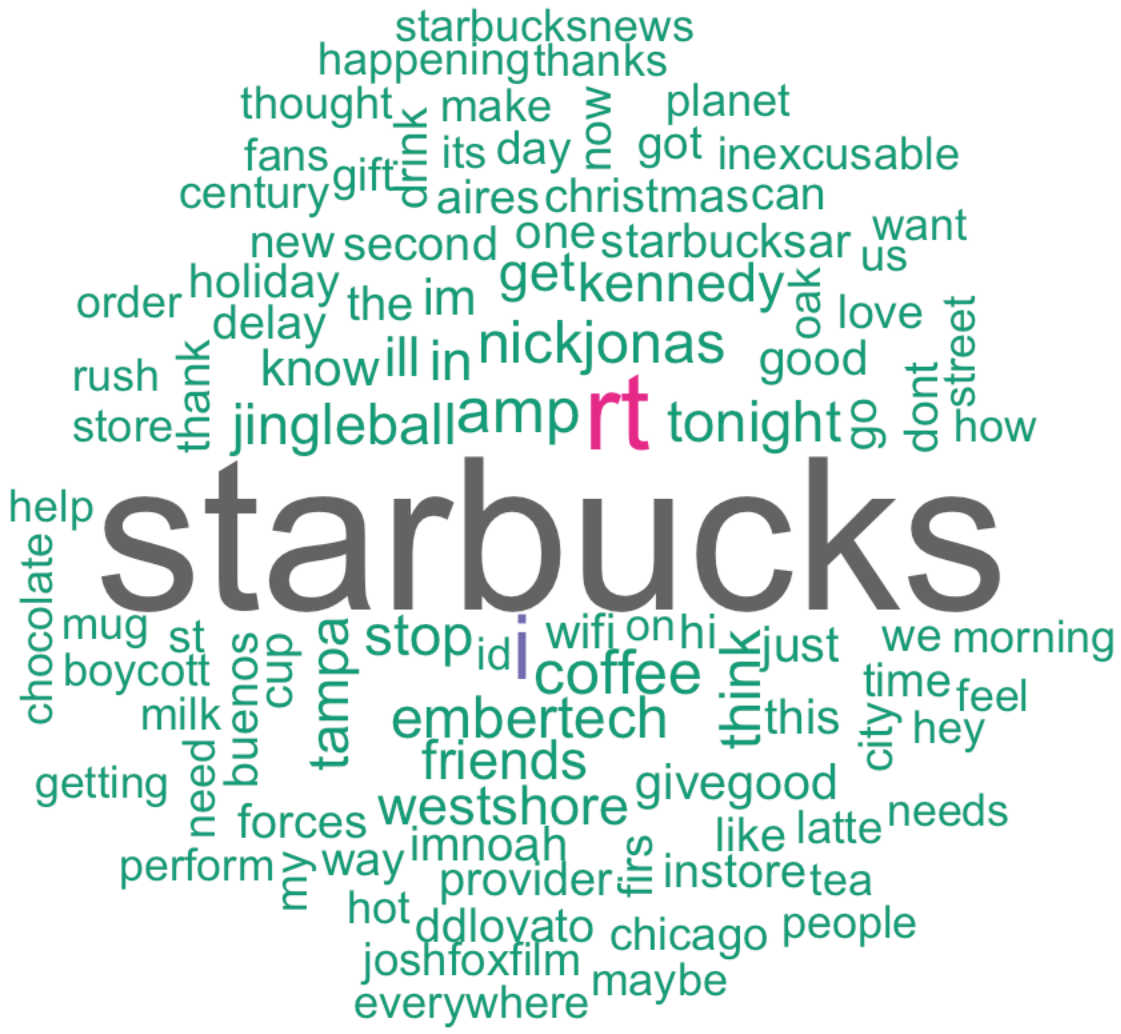



```
ggplot(df2, aes(x=term, y=freq)) + geom_bar(stat = "identity") + xlab ("Terms") + yla
b("Count") + coord_flip() + theme(axis.text=element_text(size=7))
```


```
ggplot(df3, aes(x=term, y=freq)) + geom_bar(stat = "identity") + xlab ("Terms") + yla
b("Count") + coord_flip() + theme(axis.text=element_text(size=7))
```




```
# word cloud
library(RColorBrewer)
library(wordcloud)
m <- as.matrix(tdm)
m1 <- as.matrix(tdm1)
m2 <- as.matrix(tdm2)
m3 <- as.matrix(tdm3)
word_freqs <- sort(rowSums(m), decreasing = T)
word_freqs1 <- sort(rowSums(m1), decreasing = T)
word_freqs2 <- sort(rowSums(m2), decreasing = T)
word_freqs3 <- sort(rowSums(m3), decreasing = T)
dm <- data.frame(word=names(word_freqs),freq=word_freqs)
dm1 <- data.frame(word=names(word_freqs1),freq=word_freqs1)
dm2 <- data.frame(word=names(word_freqs2),freq=word_freqs2)
dm3 <- data.frame(word=names(word_freqs3),freq=word_freqs3)
w1 <- wordcloud(dm$word,dm$freq,scale=c(5,1),max.words=100,min=20,random.order = FALS
E, colors= brewer.pal(8,"Dark2"))
```




```
w2 <- wordcloud(dm1$word,dm1$freq,scale=c(5,1),max.words=100,min=20,random.order = FA
LSE, colors= brewer.pal(8,"Dark2"))
```


```
w3 <- wordcloud(dm2$word,dm2$freq,scale=c(5,1),max.words=100,min=2,random.order = FAL
SE, colors= brewer.pal(8,"Dark2"))
```

```
## Warning in wordcloud(dm2$word, dm2$freq, scale = c(5, 1), max.words =
## 100, : starbucksathome could not be fit on page. It will not be plotted.
```

```
## Warning in wordcloud(dm2$word, dm2$freq, scale = c(5, 1), max.words =
## 100, : morning could not be fit on page. It will not be plotted.
```

```
## Warning in wordcloud(dm2$word, dm2$freq, scale = c(5, 1), max.words =
## 100, : viva could not be fit on page. It will not be plotted.
```

```
## Warning in wordcloud(dm2$word, dm2$freq, scale = c(5, 1), max.words =
## 100, : whimsy could not be fit on page. It will not be plotted.
```

```
## Warning in wordcloud(dm2$word, dm2$freq, scale = c(5, 1), max.words =
## 100, : glace could not be fit on page. It will not be plotted.
```


```
w4 <- wordcloud(dm3$word,dm3$freq,scale=c(5,1),max.words=100,min=20,random.order = FA
LSE, colors= brewer.pal(8,"Dark2"))
```


# II Statistical Analysis

I will explore some questions below using Statistical modelling:

1. Whether people hold a negative attitude toward Starbucks three marketing hashtags
2. What is the relationship between users' sentiment of tweets and the tweets' popularity
3. Whether there is a statistical difference between sentiment scores among those three different marketing hashtags and between retweet numbers

Before answering those questions, I am going to get each tweets' sentiment scores:

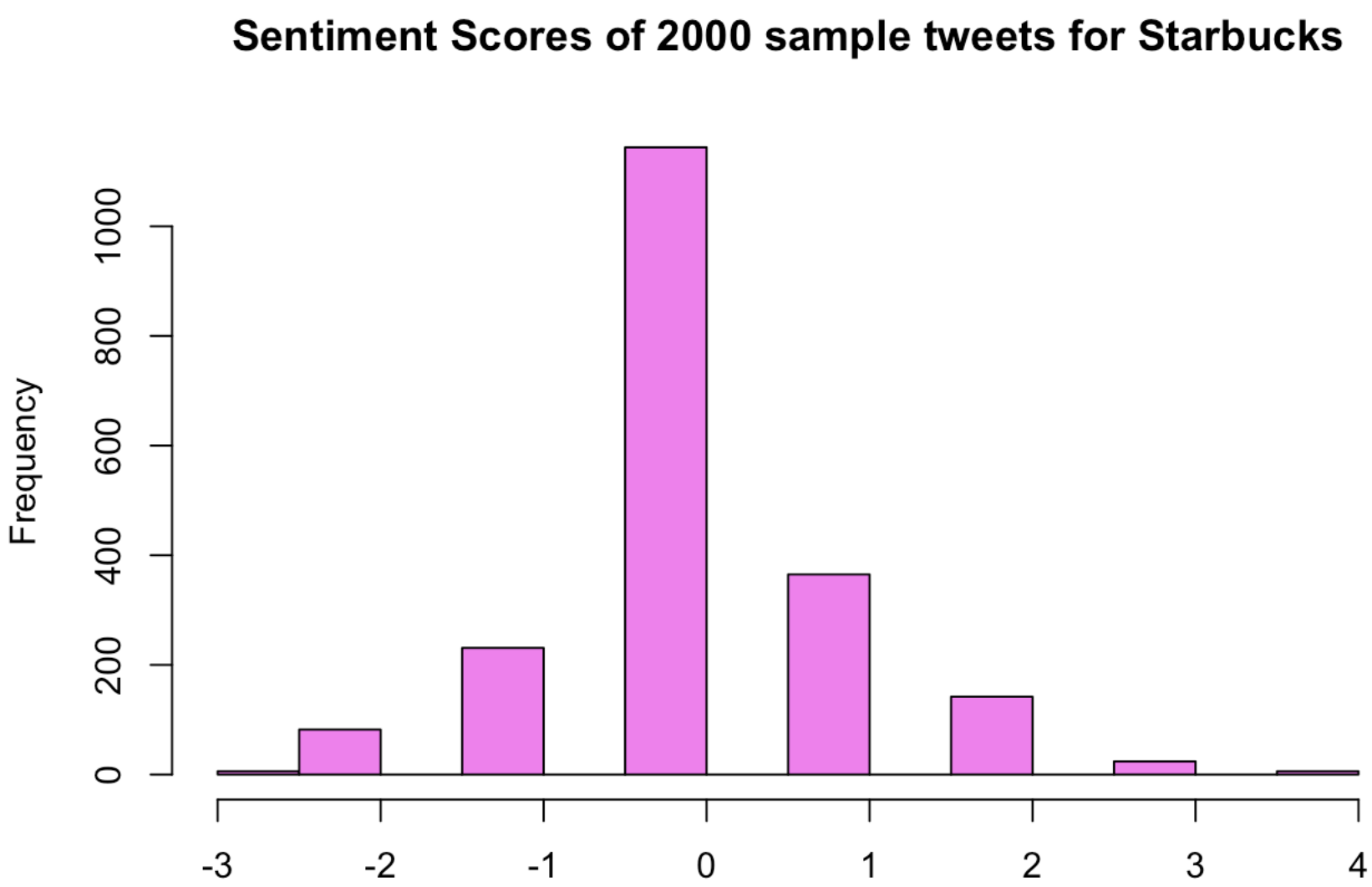
```
positives = readLines("positive-words.txt")
negatives = readLines("negative-words.txt")
library(plyr)
```

```
##
## Attaching package: 'plyr'
```

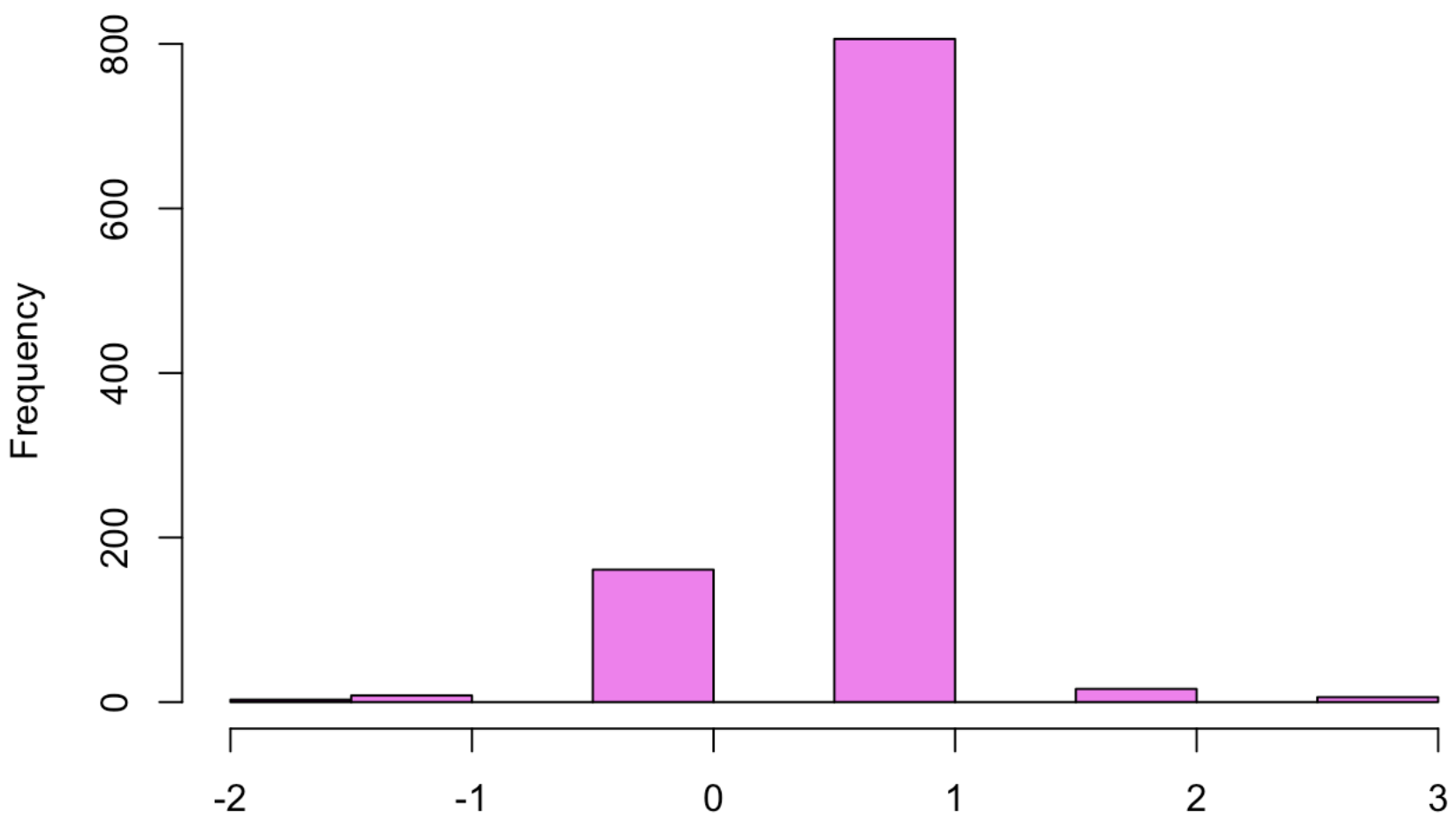
```
## The following object is masked from 'package:twitteR':
##
##     id
```

```
library(stringr)
require(plyr)
require(stringr)
sentiment_scores = function(tweets, positive_words, negative_words, .progress='none')
{
  scores = laply(tweets,
                 function(tweets, positive_words, negative_words){
                   tweets = gsub("[[:punct:]]", "", tweets)    # remove punctuation
                   tweets = gsub("[[:cntrl:]]", "", tweets)   # remove control charac
ters
                   tweets = gsub('\\+', '', tweets)          # remove digits

                   # Let's have error handling function when trying tolower
                   tryTolower = function(x){
                     # create missing value
                     y = NA
                     # tryCatch error
                     try_error = tryCatch(tolower(x), error=function(e) e)
                     # if not an error
                     if (!inherits(try_error, "error"))
                       y = tolower(x)
                     # result
                     return(y)
                   }
                   # use tryTolower with sapply
                   tweets = sapply(tweets, tryTolower)
                   # split sentence into words with str_split function from stringr p
ackage
                   word_list = str_split(tweets, "\\s+")
                   words = unlist(word_list)

                   # compare words to the dictionaries of positive & negative terms
                   positive.matches = match(words, positive_words)
                   negative.matches = match(words, negative_words)
                   # get the position of the matched term or NA
                   # we just want a TRUE/FALSE
                   positive_matches <- !is.na(positive.matches)
                   negative_matches <- !is.na(negative.matches)
                   # final score
                   score = sum(positive_matches) - sum(negative_matches)
                   return(score)
                 }, positive_words, negative_words, .progress=.progress)
  return(scores)
}
score = sentiment_scores(tweetFrame$text, positives, negatives)
```

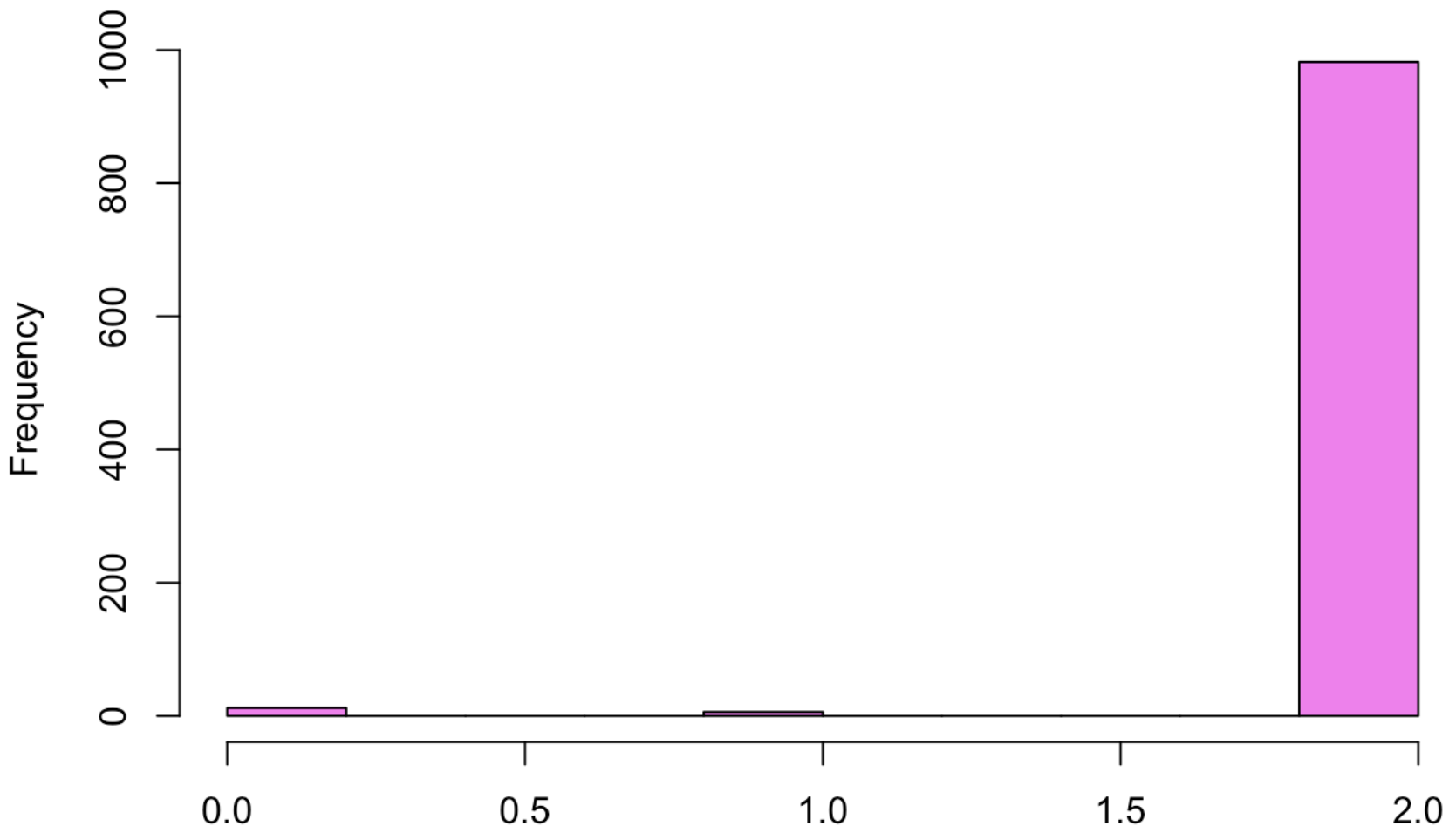
```
score1=sentiment_scores(tweetFrame1$text, positives, negatives)
score2=sentiment_scores(tweetFrame2$text, positives, negatives)
score3=sentiment_scores(tweetFrame3$text, positives, negatives)
hist(score,xlab=" ",main="Sentiment Scores of 2000 sample tweets for Starbucks" ,
     border="black",col="violet")
```



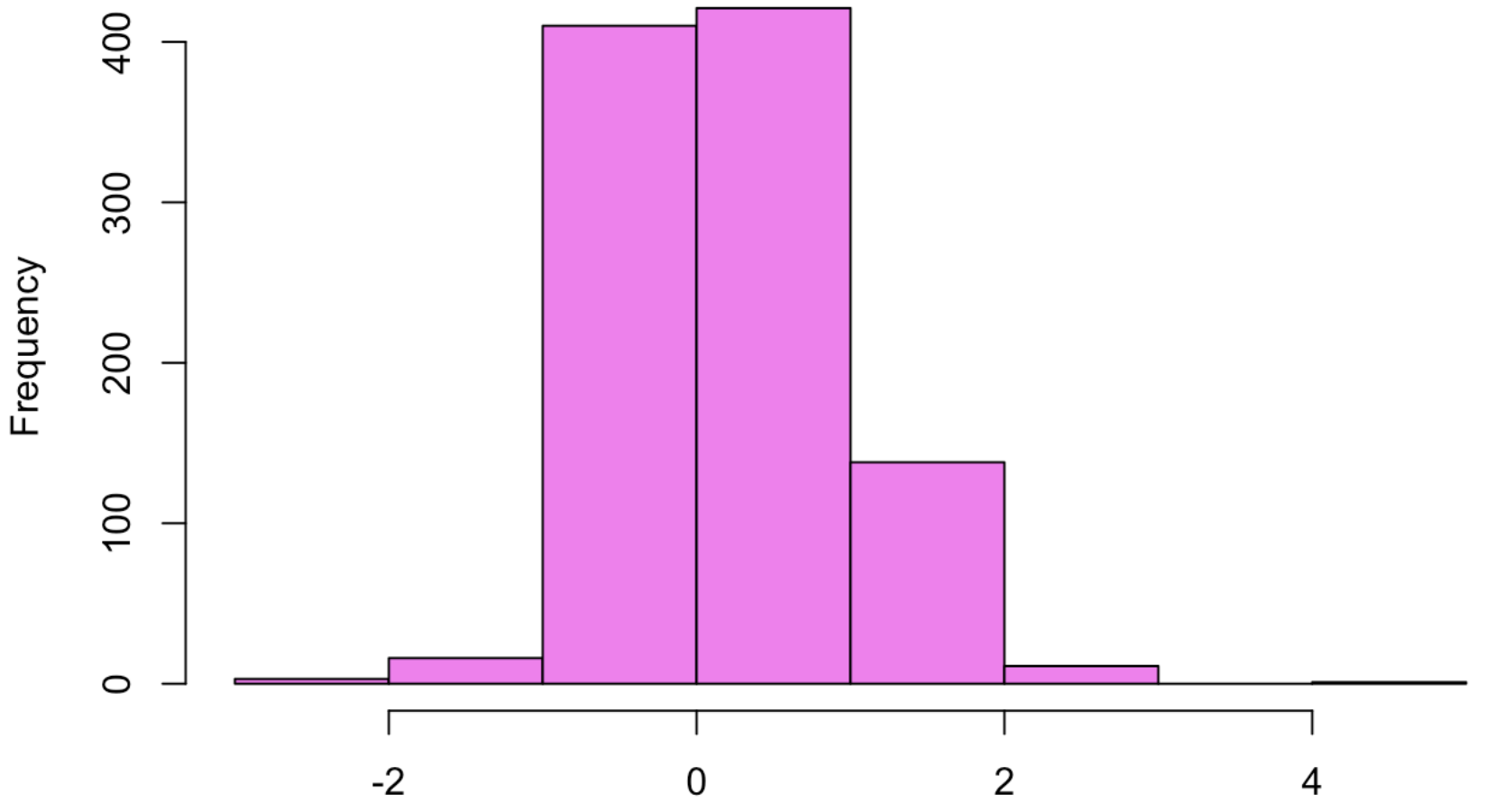

```
hist(score1,xlab=" ",main="Sentiment Scores of sample tweets for Starbucksforlife",
     border="black",col="violet")
```




```
hist(score2,xlab=" ",main="Sentiment Scores of sample tweets for Starbucksathome",
     border="black",col="violet")
```




```
hist(score3,xlab=" ",main="Sentiment Scores of sample tweets for Starbucksgivegood",
     border="black",col="violet")
```




I notice that the sentiment scores in the tweets mentioning Starbucks and the starbucks' perceptions are mostly neutral around 0. The averages of sentiment scores for Starbucks for life and Starbucks holiday deals are both very high around 1.

Before exploring those statistical questions, I will select all the necessary information and get the summary of data.

```
# add "hashtag"
data1$hashtag =rep("starbucksforlife",454)
data2$hashtag = rep("starbucksathome",220)
data3$hashtag=rep("starbucksgivegood",313)
# combining those data into one for comparison
d4 = rbind(data1,data2,data3)
scoretotal = na.omit(d4$score)
summary(scoretotal)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    -2.0     1.0     1.0     1.1     2.0     6.0
```

```
retweet = total$retweet_count
favorite = total$favorite_count
summary(retweet)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     0.0     0.0     0.0   411.7   600.0  5233.0
```

```
summary(favorite)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.0000  0.0000  0.0000  0.3597  0.0000 83.0000
```

The average of total sentiment score equals to 1.1. The minimum for the sentiment score is -2 and the maximum is 6.0. Thus, we can conclude that on average tweet users have a very positive attitude toward all three starbucks' marketing strategies.

The average of retweet count equals to 411.7. The maximum number of retweet is 5233.0. The average of favorite count equals to 0.3597. The maximum number of favorite is 83.0. On average, retweet number is larger than the favorite number. Therefore, for the further exploration, I am only going to focus on the sentiment score and the number of retweet count.

1. Hypothesis Test

1. Null Hypothesis: People have a negative attitude toward Starbucks For Life. (i.e. Score = -1)

```
alpha = 0.05
m0 = 1
t1 = (mean(data1$score-m0))/(sd(data1$score)/sqrt(length(data1$score)))
t1
```

```
## [1] -4.121518
```

```
2* pt(-abs(t1),df=length(data1$score)-1)
```

```
## [1] 4.47653e-05
```

2. Null Hypothesis: People have a negative attitude toward Starbucks At Home.

```
t2 = (mean(data2$score-m0))/(sd(data2$score)/sqrt(length(data2$score)))
t2
```

```
## [1] 57.12711
```

```
2* pt(-abs(t2),df=length(data2$score)-1)
```

```
## [1] 1.536937e-133
```

3. Null Hypothesis: People have a negative attitude toward Starbucks Give Good.

```
t3 = (mean(data3$score-m0))/(sd(data3$score)/sqrt(length(data3$score)))
t3
```

```
## [1] -5.229486
```

```
2* pt(-abs(t3),df=length(data3$score)-1)
```

```
## [1] 3.121508e-07
```

Since all of the p-value is less than 0.05, we should reject all three null hypothesis and conclude that on average twitter users do not have a negative attitude toward starbucks all three marketing, i.e. Starbucks For Life, Starbucks At Home, Starbucks Give Good.

3. Single Factor Studies

In this single factor analysis, I would explore the relationship between sentiment scores and retweet numbers. Whether there is a linear relationship between those variables and what is the causal relationship are going to be analyzed by the ANOVA Table.

```
# Anova Table
# analyze the relationship between sentiment and retweet number
aov.out = aov(total$retweet_count~total$score,data=total)
summary(aov.out)
```

```
##              Df    Sum Sq   Mean Sq F value Pr(>F)
## total$score   1 150486101 150486101   510.6 <2e-16 ***
## Residuals   985 290295938    294717
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
model = lm(total$retweet_count~total$score)
summary(model)
```

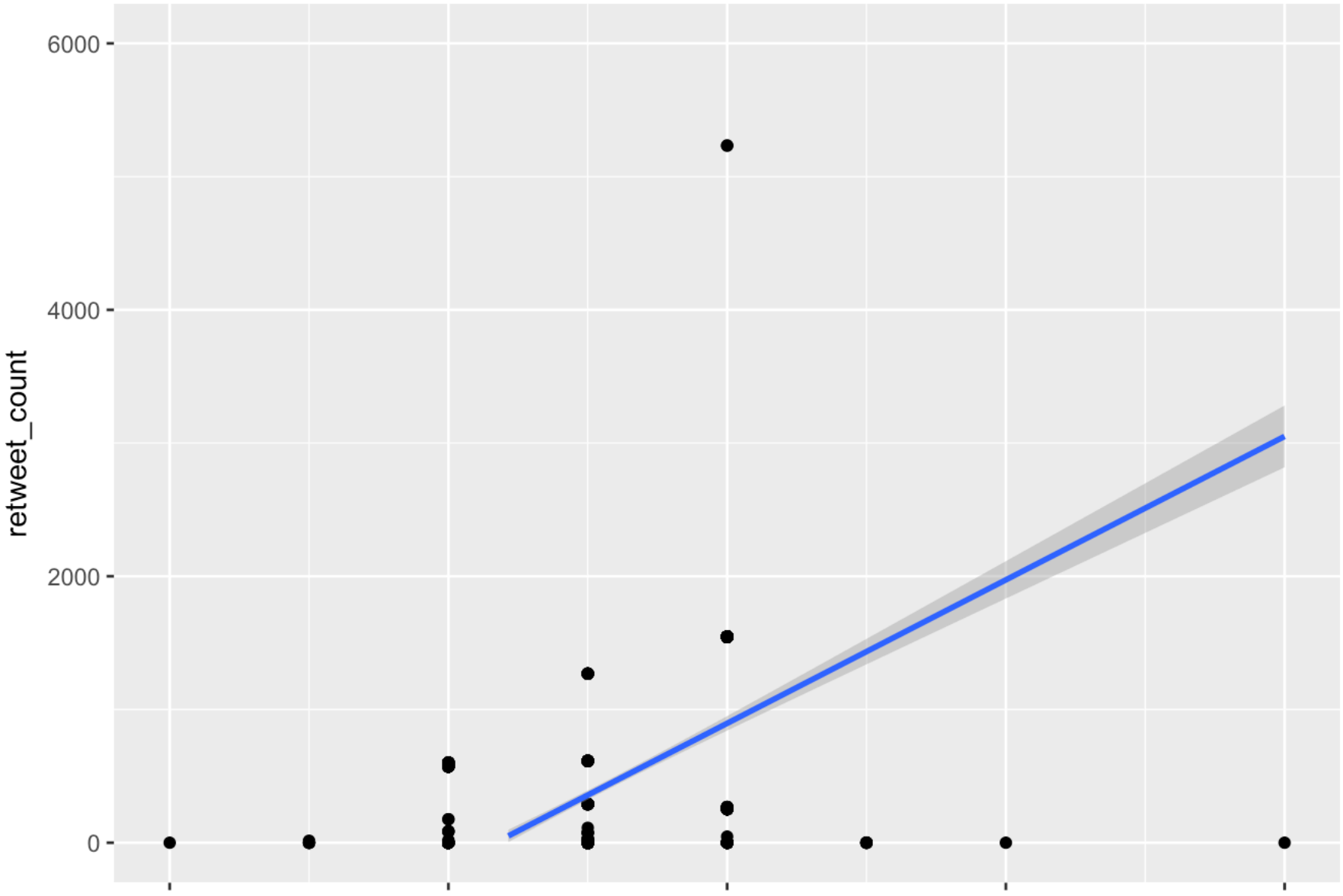
```
##
## Call:
## lm(formula = total$retweet_count ~ total$score)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3049.9  -357.7  -357.7   649.8  4336.8
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -180.72      31.40  -5.755 1.15e-08 ***
## total$score   538.44      23.83  22.597  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 542.9 on 985 degrees of freedom
## Multiple R-squared:  0.3414, Adjusted R-squared:  0.3407
## F-statistic: 510.6 on 1 and 985 DF,  p-value: < 2.2e-16
```

Since the p < 2e-16 < 0.05, we are confident to say that retweet number is related to sentiments.

In the linear regression model, the coefficient in front of the score is 538.44. Thus, we can conclude that there is a postivie relationship between the sentiment score and retweet number. Specifically, when the sentiment score increases by 1, the retweet number would increase by 538.44 on average. The R-squared in our model equals to 0.3414, which means that 34.14% of the total variaion in the sentiment scores can be explained by our model.

```
ggplot(data=total,aes(x=score,y=retweet_count)) +
  geom_point() +
  geom_smooth(method="lm") +
  scale_y_continuous(limits=c(0,6000))
```

```
## Warning: Removed 24 rows containing missing values (geom_smooth).
```




From the plot, we notice that only tweets having netural or positive get retweeted. Tweets having negative sentiment scores are not retweeted by others. However, the data set are still too small to make this predication strong. Therefore, we may need a larger data set for further analysis. In order to find the relationship between

4. Multiple Factors Analysis

In this multiple factors analysis, I would explore whether there is a mean difference between three treatment groups, i.e. three different hashtags:“Starbucks For Life”,“Starbucks At Home”, “Starbucks Give Good”.

```
# Compare the retweet mean difference across the three hashtag groups
results <- aov(total$retweet_count~factor(d4$hashtag), data =total)
summary(results)
```

```
##                      Df    Sum Sq   Mean Sq F value Pr(>F)
## factor(d4$hashtag)    2 349183958 174591979    1876 <2e-16 ***
## Residuals           984  91598081     93087
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary.lm(results)
```

```
##
## Call:
## aov(formula = total$retweet_count ~ factor(d4$hashtag), data = total)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1511.0   -16.5   -16.5    35.1  5020.6
##
## Coefficients:
##                                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)                         1510.95      20.57   73.45   <2e-16
## factor(d4$hashtag)starbucksforlife  -1494.46      25.06  -59.63   <2e-16
## factor(d4$hashtag)starbucksgivegood -1298.58      26.84  -48.38   <2e-16
##
## (Intercept)                         ***
## factor(d4$hashtag)starbucksforlife  ***
## factor(d4$hashtag)starbucksgivegood ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 305.1 on 984 degrees of freedom
## Multiple R-squared:  0.7922, Adjusted R-squared:  0.7918
## F-statistic:  1876 on 2 and 984 DF,  p-value: < 2.2e-16
```

Since the p-value < 2e-16, we can conclude that there are significant differences between those three groups.Tweets with hashtags starbucks at home get the largest retweeted. “Starbucks give good” gets the second largest retweeted. “Starbucks for life” gets the least retweeted.

```
# Compare the score mean difference between the three hashtag groups
# (i.e. "Starbucks For Life","Starbucks At Home", "Starbucks Give Good")
results <- aov(total$score ~ factor(d4$hashtag), data=total)
summary(results)
```

```
##                      Df Sum Sq Mean Sq F value Pr(>F)
## factor(d4$hashtag)    2  215.5  107.76   349.3 <2e-16 ***
## Residuals           984  303.6    0.31
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary.lm(results)
```

```
##
## Call:
## aov(formula = total$score ~ factor(d4$hashtag), data = total)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.9163  0.0364  0.0837  0.0837  5.2396
##
## Coefficients:
##                                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)                         1.96364    0.03745   52.44   <2e-16
## factor(d4$hashtag)starbucksforlife -1.04734    0.04563  -22.95   <2e-16
## factor(d4$hashtag)starbucksgivegood -1.20325    0.04887  -24.62   <2e-16
##
## (Intercept)                        ***
## factor(d4$hashtag)starbucksforlife ***
## factor(d4$hashtag)starbucksgivegood ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5554 on 984 degrees of freedom
## Multiple R-squared:  0.4152, Adjusted R-squared:  0.414
## F-statistic: 349.3 on 2 and 984 DF,  p-value: < 2.2e-16
```

Because p < 2e-16 < 0.05, there are significant differences between three hashtags groups. Tweets with hashtag #starbucks at home have the strongest emotions.

# III. Mapping

In order to visualize the sentiment scores in the context of location distributions of twitter users who tweet on those three topics. I am going to map based on the sentiment scores. The positive scores would be indicated by the red points and the negative scores would be indicated by the blue points. The absolute value of the sentiment scores would be shown by the size of the points on the map.

First, I created a U.S. map with the sentiment scores of all three hashtags. The map shows that the data are mostly scattered around California and East Coast. The map are generally covered by the red point, which means that customers hold a positive attitude toward Starbucks when they talk about those three topics. There are few blue point show on Inidana and Massachusetts. Thus, I will plot each topic's sentiment scores on the second U.S. map to see which topic has the largest number of negative scores.

```
library(ggplot2)
library(ggmap)
```

```
## Google Maps API Terms of Service: http://developers.google.com/maps/terms.
```

```
## Please cite ggmap if you use it: see citation("ggmap") for details.
```

```
library(RgoogleMaps)
library(devtools)
library(maps)
```

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

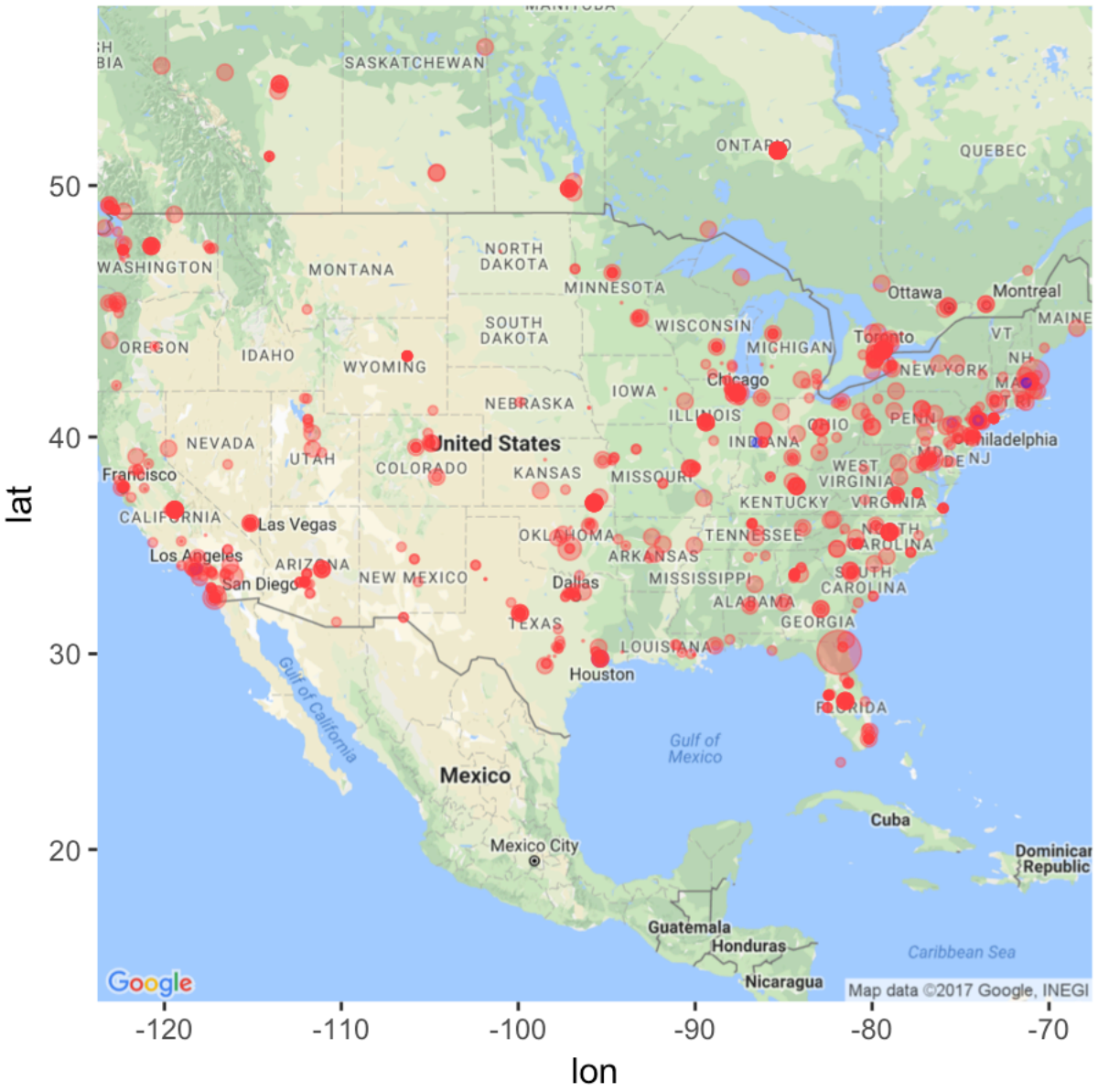
```
## The following object is masked from 'package:plyr':
##
##     ozone
```

```
library(mapdata)
USMap <- get_map (location = c(lon = -95.71289, lat = 37.09024), zoom=4, scale=2, map
type="roadmap", source="google",crop=TRUE)
```

```
## Source : https://maps.googleapis.com/maps/api/staticmap?center=37.09024,-95.71289&
zoom=4&size=640x640&scale=2&maptype=roadmap&language=en-EN
```

```
Map1 <- ggmap(USMap) +
  geom_point(data = total, aes(x=total$lon,y=total$lat),col=ifelse(((total$score>=0))
,"brown1","blue"),alpha=0.4,size=total$absolute_score) +
  scale_size_continuous(range=total$score)+
  ggtitle("US Map for the Total Dataset")
Map1
```

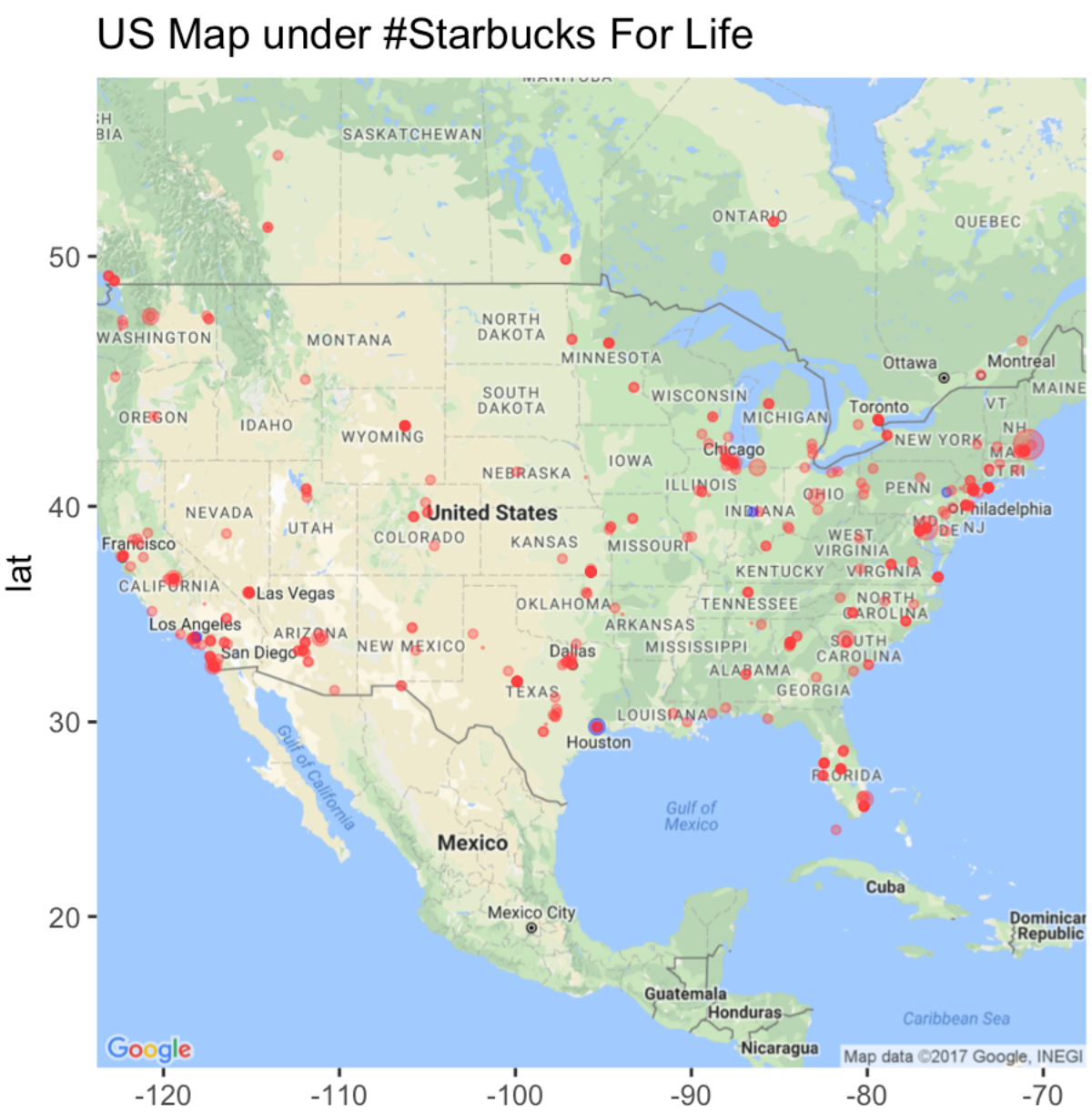
```
## Warning: Removed 4 rows containing missing values (geom_point).
```




From the individual map, I notice that the blue point occurs when people in Indiana, Penn, and Houston talk about starbucks for life event. People in Massachusetts have negative perception toward Starbucks Give Good program. There are no blue point shows under the topic Starbucks At Home. To get a better sence of sentiment scores in different region, I then plot all those sentiments scores in Los Angels, Massachusetts, Inidana, and New York to visualize what local users' perceptions toward Starbucks.

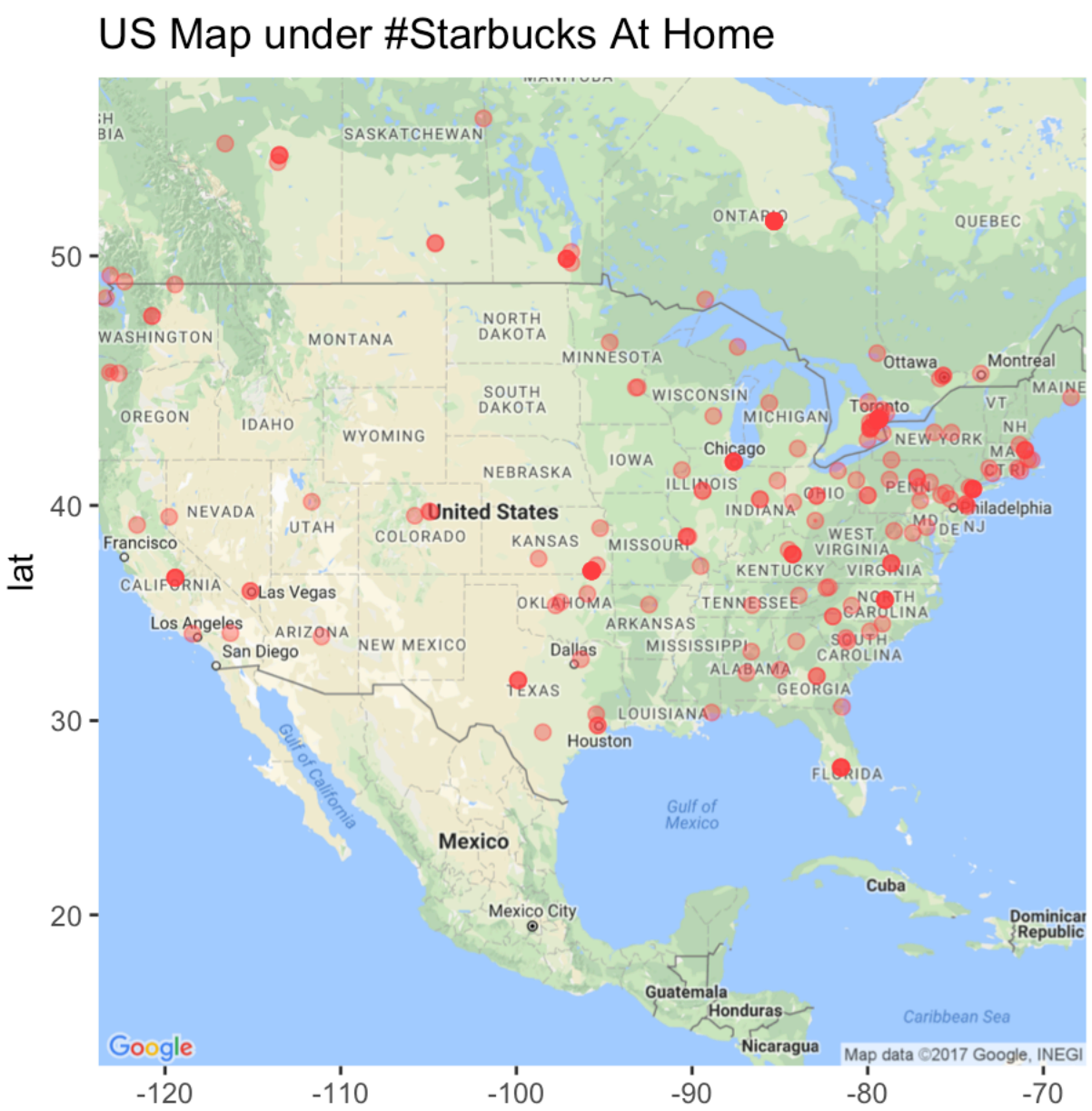
```
# mapping under topic "starbucks for life"
Map2 <- ggmap(USMap) +
  geom_point(data = data1, aes(x=data1$lon,y=data1$lat),col=ifelse(((data1$score>=0))
,"brown1","blue"),alpha=0.4,size=data1$absolute_score) +
  scale_size_continuous(range=data1$score)+
  ggtitle("US Map under #Starbucks For Life")
Map2
```

```
## Warning: Removed 3 rows containing missing values (geom_point).
```

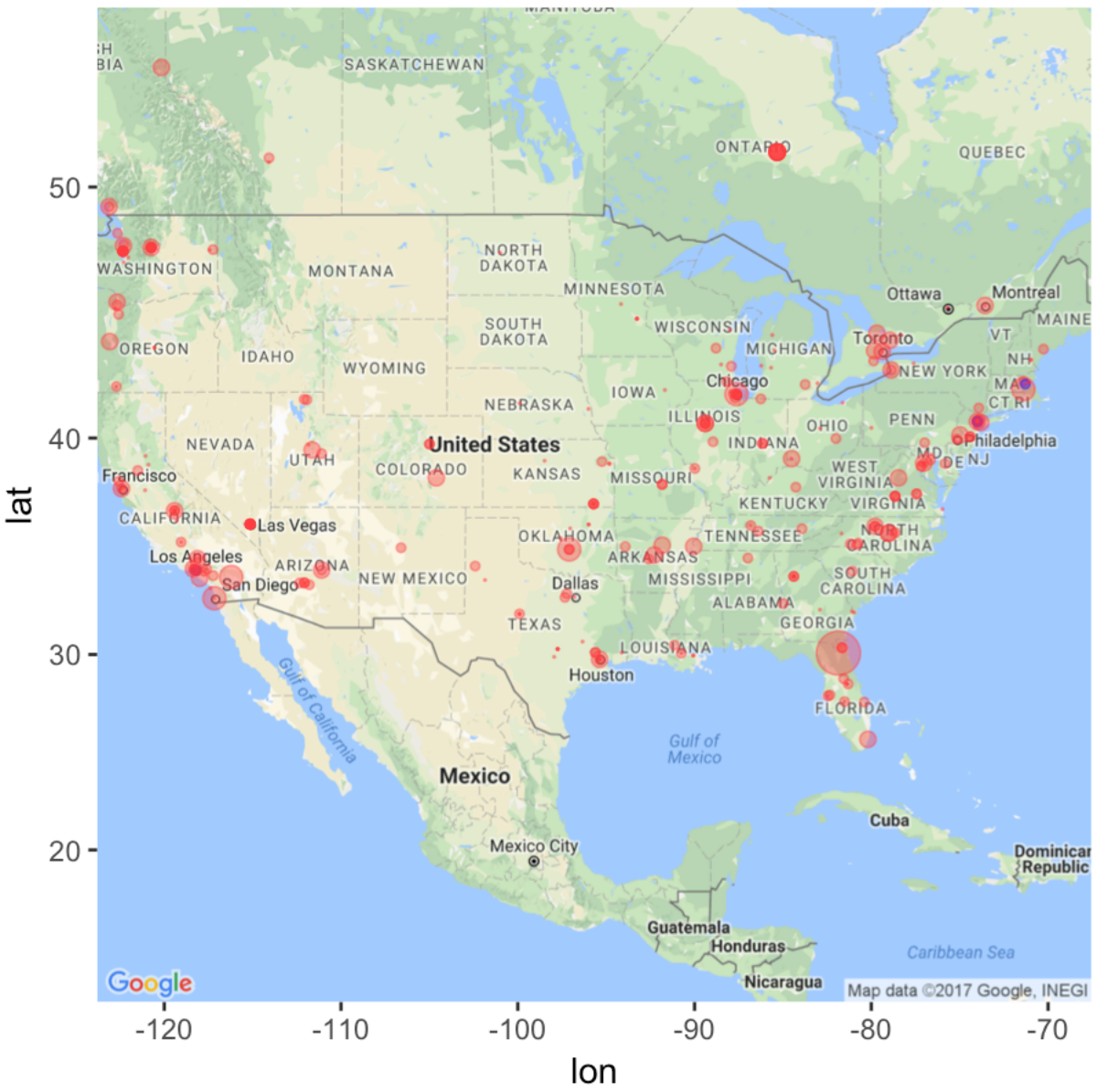



```
# mapping under topic "starbucks at home"
Map3 <- ggmap(USMap) +
  geom_point(data = data2, aes(x=data2$lon,y=data2$lat),col=ifelse(((data2$score>=0))
,"brown1","blue"),alpha=0.4,size=data2$absolute_score) +
  scale_size_continuous(range=data2$score)+
  ggtitle("US Map under #Starbucks At Home")
Map3
```

```
## Warning: Removed 1 rows containing missing values (geom_point).
```




```
# mapping under topic "starbucks give good"
Map4 <- ggmap(USMap) +
  geom_point(data = data3, aes(x=data3$lon,y=data3$lat),col=ifelse(((data3$score>=0))
,"brown1","blue"),alpha=0.4,size=data3$absolute_score) +
  scale_size_continuous(range=data3$score)+
  ggtitle("US Map under #Starbucks Give Good")
Map4
```



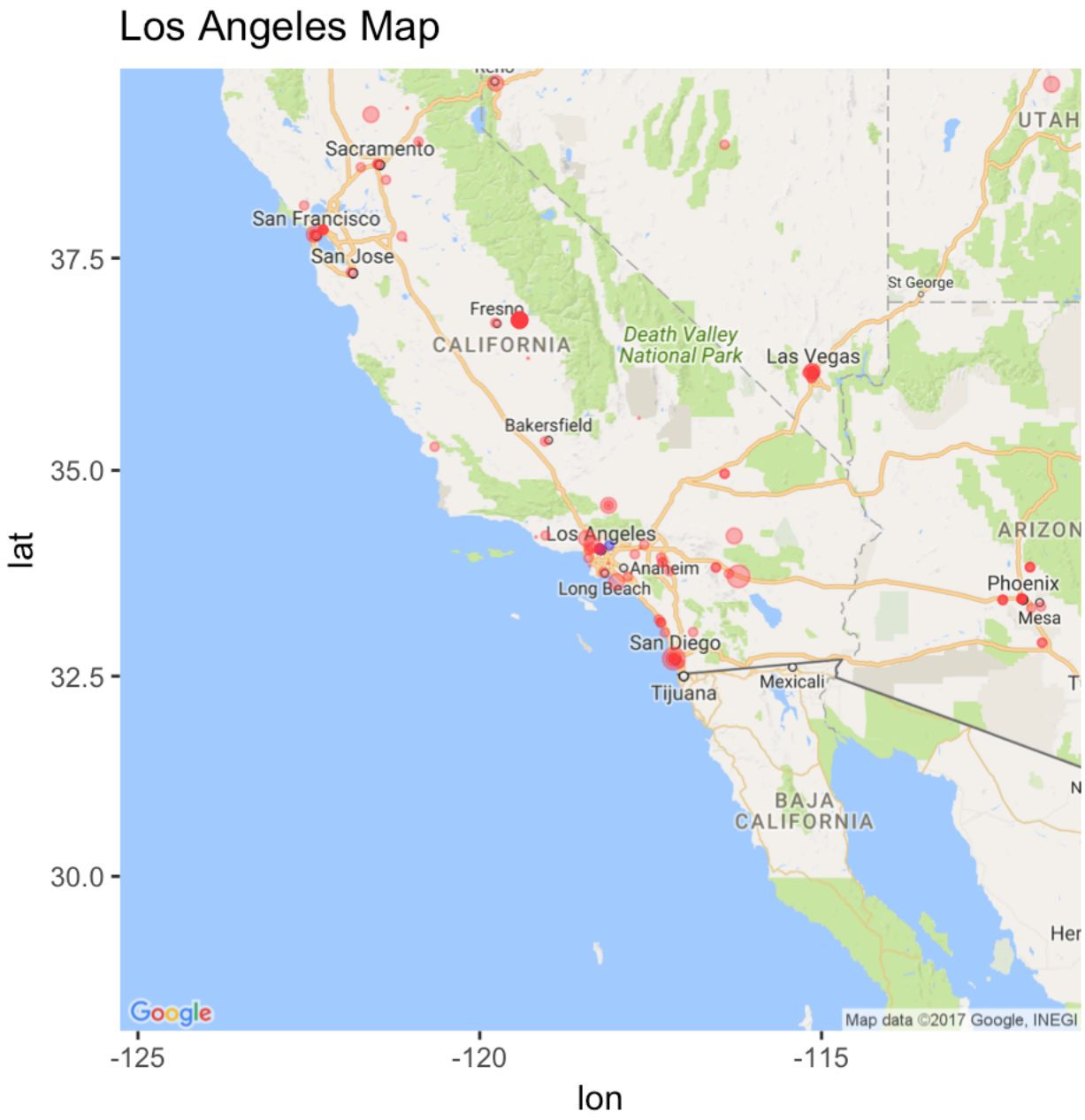

I find out that there are still blue point shows on Los Angeles Map and on the map of Peen and Massachussetes. The New York sentiment scores are much higher in that there is almost no blue point. However, the sample size of data with local information is so small that there are too little points plotted on the map to reach any firm conclusion.

```
# sentiment across different locations
# Los Angeles
LAMap <- get_map (location = c(lon = -118.2437, lat = 34.05223), zoom=6, scale=2, map
type="roadmap", source="google",crop=TRUE)
```

```
## Source : https://maps.googleapis.com/maps/api/staticmap?center=34.05223,-118.2437&
zoom=6&size=640x640&scale=2&maptype=roadmap&language=en-EN
```

```
Map5 <- ggmap(LAMap) +
  geom_point(data = total, aes(x=total$lon,y=total$lat),col=ifelse(((total$score>=0))
,"brown1","blue"),alpha=0.4,size=total$absolute_score) +
  scale_size_continuous(range=total$score)+
  ggtitle("Los Angeles Map")
Map5
```

```
## Warning: Removed 831 rows containing missing values (geom_point).
```

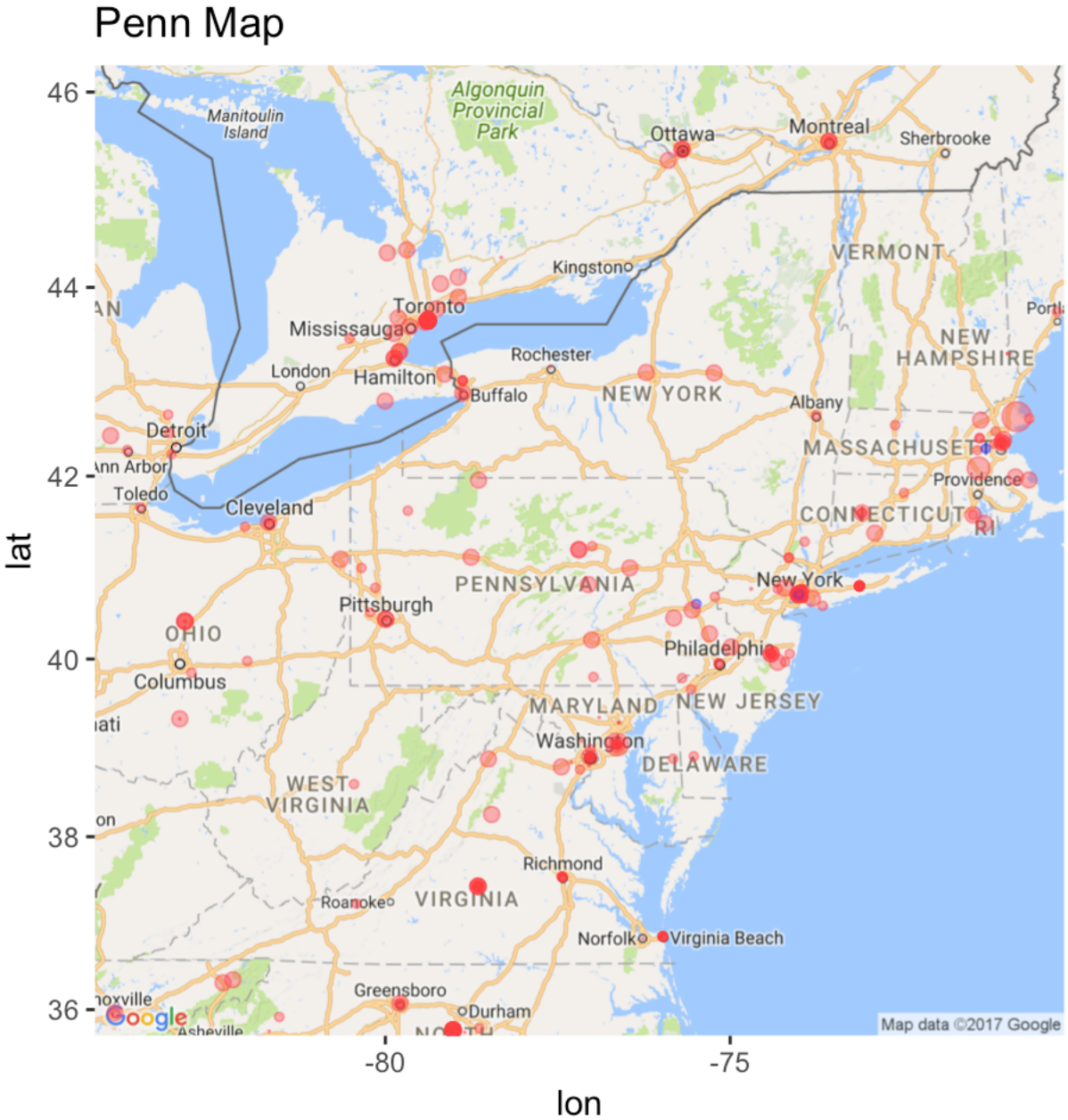



```
# Penn
PMap <- get_map (location = c(lon = -77.19452, lat = 41.20332), zoom=6, scale=2, mapt
ype="roadmap", source="google",crop=TRUE)
```

```
## Source : https://maps.googleapis.com/maps/api/staticmap?center=41.20332,-77.19452&
zoom=6&size=640x640&scale=2&maptype=roadmap&language=en-EN
```

```
Map6 <- ggmap(PMap) +
  geom_point(data = total, aes(x=total$lon,y=total$lat),col=ifelse(((total$score>=0))
,"brown1","blue"),alpha=0.4,size=total$absolute_score) +
  scale_size_continuous(range=total$score)+
  ggtitle("Penn Map")
Map6
```

```
## Warning: Removed 700 rows containing missing values (geom_point).
```

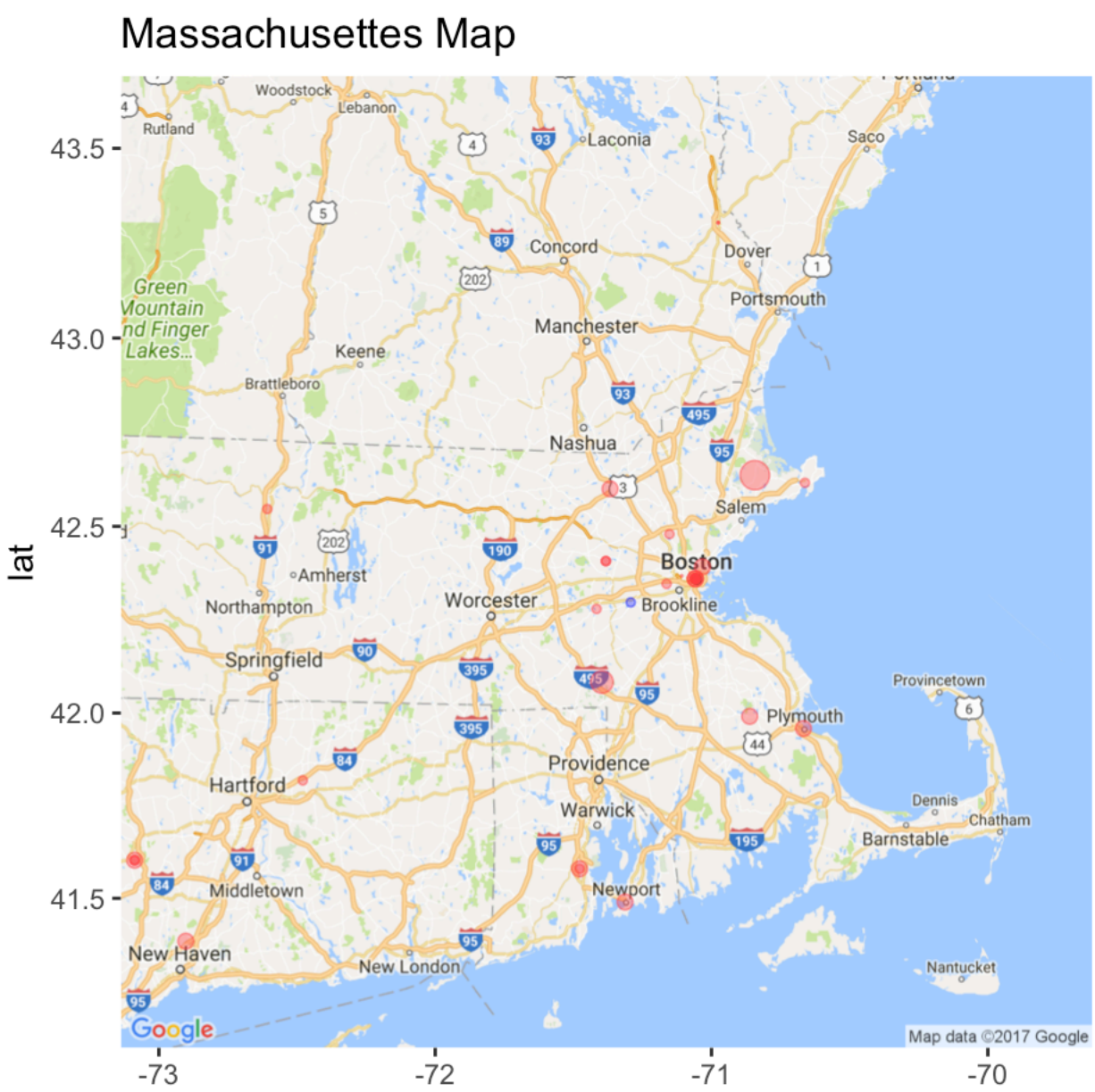



```
# Massachusettes
MaMap <- get_map (location = c(lon = -71.38244, lat = 42.40721), zoom=8, scale=2, map
type="roadmap", source="google",crop=TRUE)
```

```
## Source : https://maps.googleapis.com/maps/api/staticmap?center=42.40721,-71.38244&
zoom=8&size=640x640&scale=2&maptype=roadmap&language=en-EN
```

```
Map7 <- ggmap(MaMap) +
  geom_point(data = total, aes(x=total$lon,y=total$lat),col=ifelse(((total$score>=0))
,"brown1","blue"),alpha=0.4,size=total$absolute_score) +
  scale_size_continuous(range=total$score)+
  ggtitle("Massachusettes Map")
Map7
```

```
## Warning: Removed 953 rows containing missing values (geom_point).
```




```
# New York
NMap <- get_map (location = c(lon = -74.00597, lat = 40.71278), zoom=9, scale=2, mapt
ype="roadmap", source="google",crop=TRUE)
```

```
## Source : https://maps.googleapis.com/maps/api/staticmap?center=40.71278,-74.00597&
zoom=9&size=640x640&scale=2&maptype=roadmap&language=en-EN
```

```
# mapping under topic "starbucks for life"
Map8 <- ggmap(NMap) +
  geom_point(data = total, aes(x=total$lon,y=total$lat),col=ifelse(((total$score>=0))
,"brown1","blue"),alpha=0.4,size=total$absolute_score) +
  scale_size_continuous(range=total$score)+
  ggtitle("New York Map")
Map8
```

```
## Warning: Removed 936 rows containing missing values (geom_point).
```


# IV. Shiny

For the Shiny application, I created a navbar with all the graphs included, an interactive map to compare the retweet tweets in different regions, and an interactive word cloud app. I include the code for shiny on the sweave file and will inclde the website link to visualize on the shiny app.

For the navbar, I put all the plots and maps I created earlier in different tab, including Map, Distribution of Sentiment scores vs. Frequency and Frequent Words Distribution.The are five subtabs in the map section. Four subtabs are included in both two distributions sections. The descriptions for each graph can also be founded above each graph. Users can easily find the graph they want to see by selecting different bar.

Follow this link to access navbar: https://xuexiaoqian.shinyapps.io/app2/ (https://xuexiaoqian.shinyapps.io/app2/)

The interactive map to compare the retweet tweets is designed to explore the popularity of the tweets in different regions. Since we have focused on the sentiment scores for each tweets on different regions, in this part, I would like to compare the popularity. I used the raw data from number of retweets counts for each tweets. In the map, the deeper the color of the popup points, the more popular the tweet is. User can zoom in to compare different regions' tweets' popularity and also click on each individual points to find out more detail about the tweet.

Follow this link to access interactive map: https://xuexiaoqian.shinyapps.io/app1/ (https://xuexiaoqian.shinyapps.io/app1/)

# V. Future Improvements

The initial sample I have chosen is 5000 tweets data. However, after selecting all the useful tweets with valid locations and english as the language, there are only around 2000 tweets available. Therefore, the sample size is too small to reach firm conclusions. In order to reach a firm conclusion about the Starbucks marketing perceptions, we should explore more data sets and not be limitted only by twitter and Google. Also, there are definitely more marketing strategies used by Starbucks, for example, sustainability, international brand images, working environment and so on. Hence, I look forward to exploring more about customers' perceptions on those strategies in different platform and area. Please feel free to email me (xuexq@bu.edu (mailto:xuexq@bu.edu)) if you have any questions. Thank you for going through this exploration with me!