MA415 Final Project - Tweets that Mentioning President Trump

Sibo Zhu

2017/12/15

1 Introduction

President Donald John Trump, as inaugurated for an how year already, are always included in people's topic. I believe politics' work can be reflected by how people talk about them. Therefore I believe how people discuss him via their social media could express their current attitudes towards President Trump, and the work he has done.

1.1 Data Description

By utilizing the Twitter API, with all the tweets contains hashtag about Trump and the geolocation info, I captured over two 200,000 tweets in 2 hours within the United States region that mention president Trump. In the latter part of this project, I'm going to analyze and research on word frequencies of all these tweets that I've captured, comparing them among different locations, and trying to find some underlying relations of the data.

```
### setting up connections for Twitter
#requestURL <- "https://api.twitter.com/oauth/request_token"
#accessURL <- "https://api.twitter.com/oauth/access_token"
#authURL <- "https://api.twitter.com/oauth/authorize"
#api_key             <- "dSzLRlqni1W2f9CJagS19RP6b"
#api_secret          <- "pLX7gIERogxsgn7AT2eEAQlhmxGKwEx9de5pjeOtjkQDDbVVCD"
#access_token        <- "3134023545-OZ9TIdXFbEFW66HmsCUoOEWGIANJ4SgS1OgdDO7"
#access_token_secret <- "kAzcVKOEMOhCFAs13Mau2Y8q2fCYHjvA5XXl7RdOSCO1o"


###Above codes are my personal Twitter App Keys, I shouldn't have included them in this project,
###but I just did so for your convenience, so please don't use them to do bad things

### Setting up streamR

#my_oauth <- OAuthFactory$new(consumerKey = api_key, consumerSecret = api_secret,
#requestURL = requestURL, accessURL = accessURL, authURL = authURL)
#my_oauth$handshake(cainfo = system.file("CurlSSL", "cacert.pem", package = "RCurl"))
#save(my_oauth, file = "my_oauth.Rdata")      #Save my Oauth info for future convenience
```

```
#load("my_oauth.Rdata")     #Load my Oauth file save above to avoid authentication everytime running th

#  filterStream( file="D_Tweet.json", track="realDonaldTrump",  #naming the downloaded file as "D_Tweet
#      locations=c(-125, 25, -66, 50), timeout=7200, oauth=my_oauth )     #restrict tweets within Unite

###Since it would take 2 hours to complete the capturing, it's certainly meaningless to let this chunk
###If you'd like to run the code yourself, you are welcome to uncomment the above lines and go for it;
###Also, due to Github's regulation of file size restriction of 100MB maximum, I cannot upload my "D_Tw
### data in that json file will be cleaned and write into "cleaned_tweets_df.csv" later
```

In the above codes I'm capturing tweets that mentioning "realDonaldTrump" (President Trump's Twitter Account) within the United States "(-125, 25, -66, 50)" region for 2 hours (7200 seconds). Turns out I captured over 200,000 tweets, which is a very big number.

2 Data Cleaning

Later I'm doing data cleaning, including filtering for language, geolocation enabled or not and three typical states (CA, NJ, NY).

Then I'm doing plots in these three states with tweets that contain geolocations and analysis the results based on the plots.

```
#tweets_raw.df <- parseTweets("DT_Tweet.json",verbose = FALSE)


#keep <- c("text","lang","listed_count","geo_enabled","statuses_count","followers_count",
#          "favourites_count","friends_count","time_zone","country_code","full_name",
#          "place_lat","place_lon")

#tweets.df <- tweets_raw.df[,keep];


#write.csv(tweets.df,"tweets_df.csv")

#save(tweets.df, file = "tweets.df.Rdata")      #save data for future convenience
```

```
load("tweets.df.Rdata")       #load data

tweets.df <- tweets.df[tweets.df$lang=="en",]     #filtering for only English tweets
tweets.df <- tweets.df[tweets.df$country_code=="US",]     #filtering for US region tweets
tweets.df <- tweets.df[tweets.df$geo_enabled==TRUE,]     #filtering for goelocations enabled tweets

write.csv(tweets.df,"cleaned_tweets_df.csv")     #writing cleaned data into CSV
```

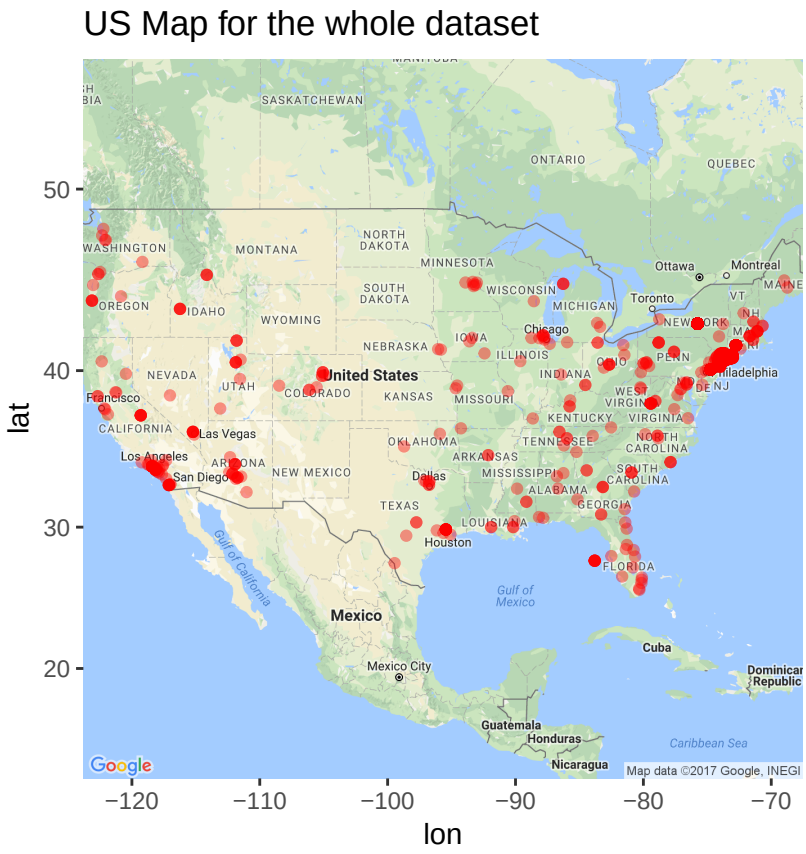
3 Geolocation Analysis

```
#filtering out three typical states that worth analyzing
ca <- data.frame(filter(tweets.df, grepl(', CA', full_name)))
nj <- data.frame(filter(tweets.df, grepl(', NJ', full_name)))
ny <- data.frame(filter(tweets.df, grepl(', NY', full_name)))

###Plotting
NJplot <- qmplot(place_lon, place_lat, data = nj, colour = I('purple'), size = I(2), alpha=I(0.4),
       mapcolor = "bw", main="New Jersey")
NYplot <- qmplot(place_lon, place_lat, data = ny, colour = I('darkred'), size = I(2), alpha=I(0.4),
       mapcolor = "bw", main="New York")
CAplot <- qmplot(place_lon, place_lat, data = ca, colour = I('blue'), size = I(2), alpha=I(0.4),
       mapcolor = "bw",main="California")

###Plotting data within whole USA region
USMap <- get_map (location = c(lon = -95.71289, lat = 37.09024), zoom=4, scale=2, maptype="roadmap", so
USAplot <- ggmap(USMap) +
```

```
  geom_point(data = tweets.df, aes(x=tweets.df$place_lon,y=tweets.df$place_lat),alpha=0.4,color="red") +
  ggtitle("US Map for the whole dataset")
```

```
USAplot
```

```
## Warning: Removed 30016 rows containing missing values (geom_point).
```

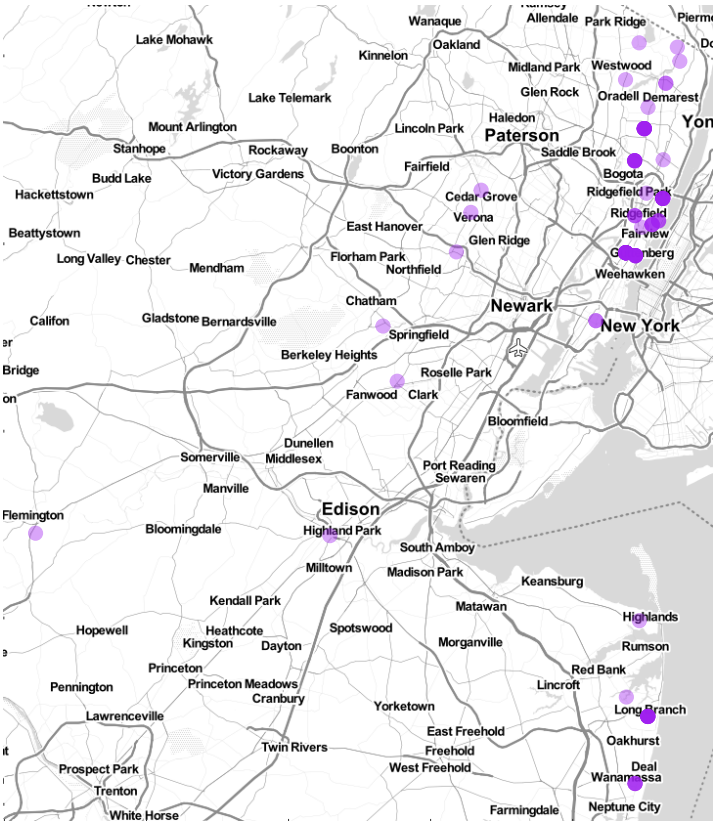



```
###From the above plot I can clearly see that most tweets about President Trump come from major cities
NYplot
```

New York


```
NJplot
```

New Jersey



```
CAplot
```

California

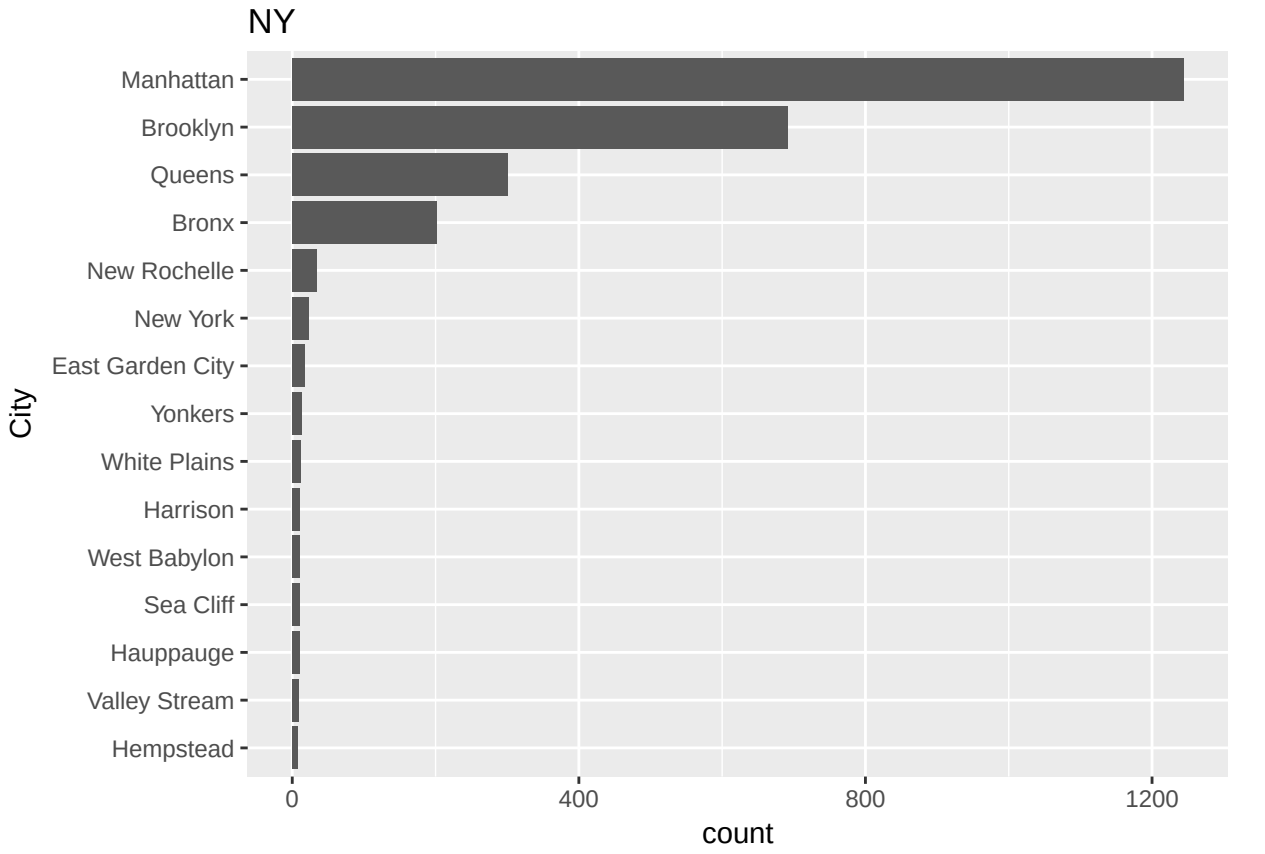

```
###Adding a new variable named "state" for future usage
ny[,"state"] <- rep("NY",nrow(ny));
nj[,"state"] <- rep("NJ",nrow(nj));
ca[,"state"] <- rep("CA",nrow(ca));

###getting city's names
ny$full_name <- as.character(ny$full_name);
nj$full_name <- as.character(nj$full_name);
ca$full_name <- as.character(ca$full_name);
tweets.df$full_name <- as.character(tweets.df$full_name);

### helping function that used for capturing cities
city_get <- function(x){
  city_name <- c()
  name <- strsplit(x$full_name,",")
  for(i in 1:nrow(x)){
    city_name <- c(city_name,name[[i]][1])
  }
  return(city_name)
}

ny[,"city"] <- factor(city_get(ny));
nj[,"city"] <- factor(city_get(nj));
ca[,"city"] <- factor(city_get(ca));
```

```
tweets.df[,"city"] <- factor(city_get(tweets.df))
total_us <- filter(tweets.df,lang=="en")
```

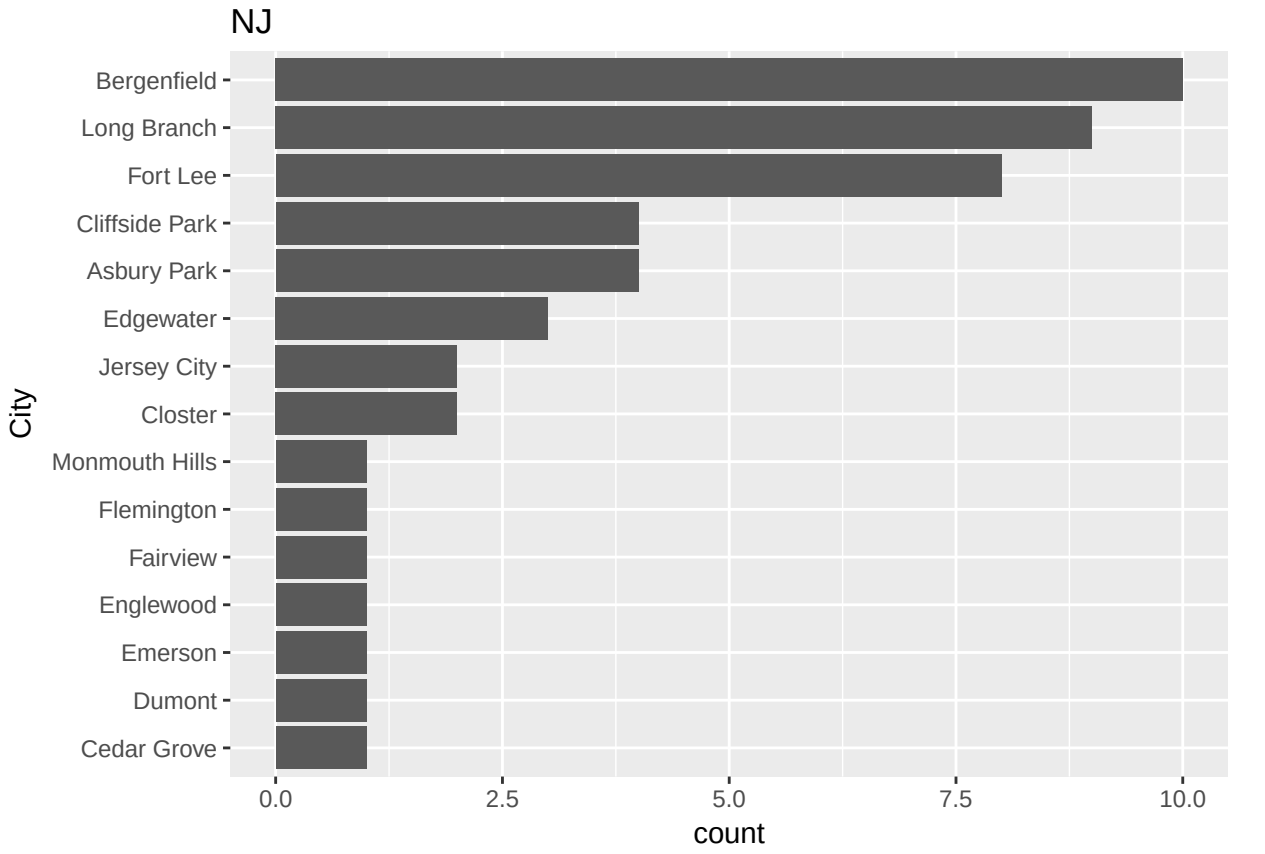
```
###Counting tweets for every city in each states
count <- summary(ny$city)[1:15];ny_city_count <- as.data.frame(count)
count <- summary(nj$city)[1:15];nj_city_count <- as.data.frame(count)
count <- summary(ca$city)[1:15];ca_city_count <- as.data.frame(count)
###Do the same thing towards whole United States region
count <- summary(tweets.df$city)[1:15];us_city_count <- as.data.frame(count)


NYCplot <- ggplot(ny_city_count,aes(reorder(rownames(ny_city_count),count),count))+
  geom_bar(stat = "identity")+coord_flip()+xlab("City")+ggtitle("NY")
#nycityplot
NJCyplot <- ggplot(nj_city_count,aes(reorder(rownames(nj_city_count),count),count))+
  geom_bar(stat = "identity")+coord_flip()+xlab("City")+ggtitle("NJ")
#njcityplot
CACplot <- ggplot(ca_city_count,aes(reorder(rownames(ca_city_count),count),count))+
  geom_bar(stat = "identity")+coord_flip()+xlab("City")+ggtitle("CA")
#cacityplot
#grid.arrange(nycityplot, njcityplot, cacityplot, ncol=3)

USCplot <- ggplot(us_city_count,aes(reorder(rownames(us_city_count),count),count))+
  geom_bar(stat = "identity")+coord_flip()+xlab("City")+ggtitle("United States")

#uscityplot
```

In below I plot TOP15 cities of the number of tweets in three states and in the aspect of whole the United States. Not surprisingly, in NY and CA, most tweets are posted in Manhattan and Los Angeles, and the sum of remaining nine cities is just equal to this super city. In NJ, most cities the similar amount of tweets. By that analysis, I can tell it's still the biggest cities care about President Trump more.

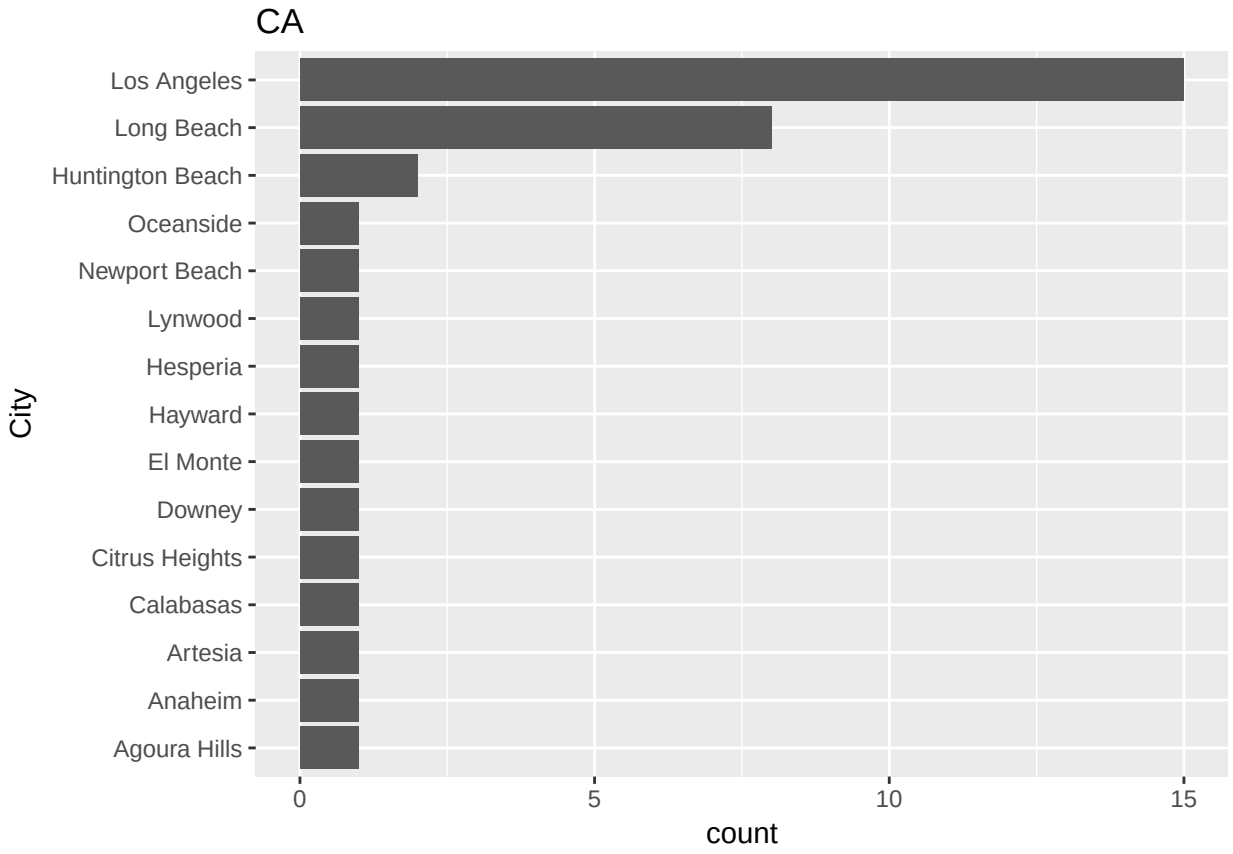
```
NYCplot
```


NY
Manhattan
Brooklyn
Queens
Bronx
New Rochelle
New York
East Garden City
Yonkers
White Plains
Harrison
West Babylon
Sea Cliff
Hauppauge
Valley Stream
Hempstead
City
0
400
800
1200
count

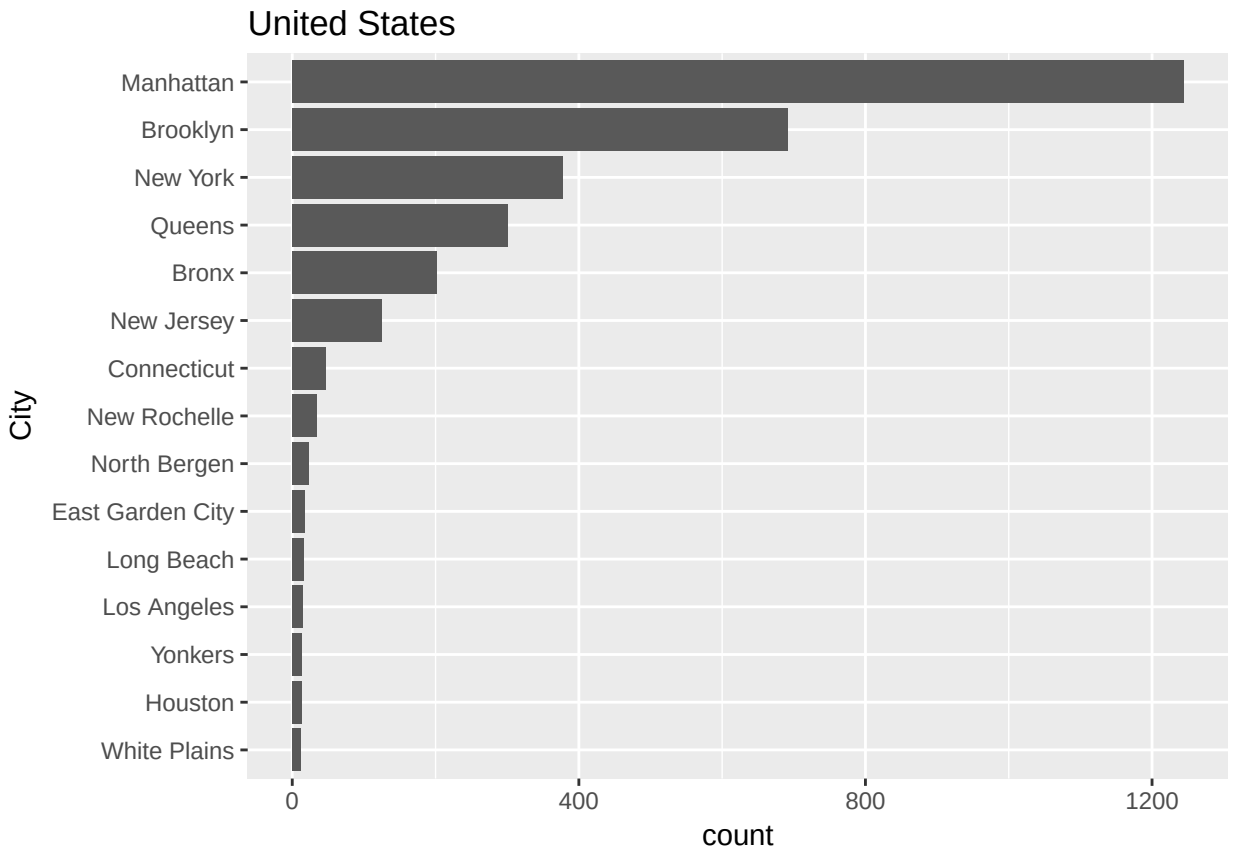
```
NJCyplot
```


NJ
Bergenfield
Long Branch
Fort Lee
Cliffside Park
Asbury Park
Edgewater
Jersey City
Closter
Monmouth Hills
Flemington
Fairview
Englewood
Emerson
Dumont
Cedar Grove
City
0.0
2.5
5.0
7.5
10.0
count

```
CACplot
```


CA
Los Angeles
Long Beach
Huntington Beach
Oceanside
Newport Beach
Lynwood
Hesperia
Hayward
El Monte
Downey
Citrus Heights
Calabasas
Artesia
Anaheim
Agoura Hills
City
0
5
10
15
count

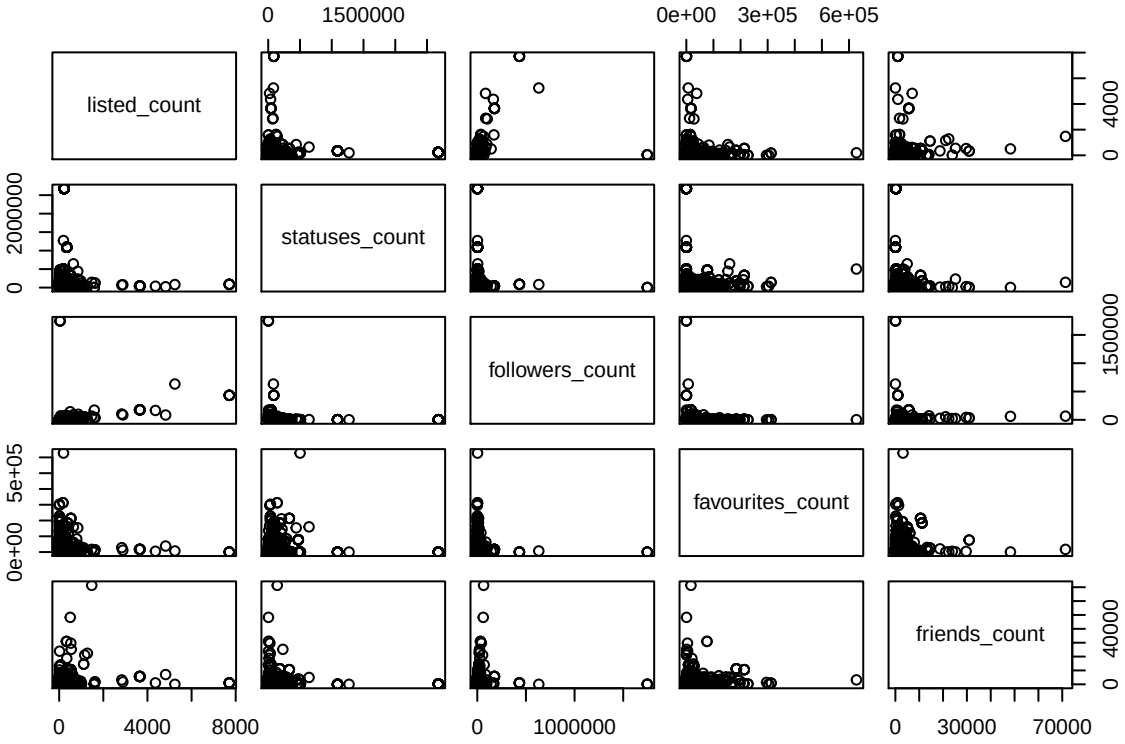
```
USCplot
```




```
#Analysis about cities and USA tweets count goes here
```

4 Popularity Study

```
pairs(~listed_count+statuses_count+followers_count+favourites_count+friends_count, data=total_us)     #
```




```
cor(total_us[,c(3,5,6,7,8)])      #analyzing correlations
```

```
##                 listed_count statuses_count followers_count
## listed_count       1.00000000    0.076124932     0.357339575
## statuses_count     0.07612493    1.000000000    -0.001358509
## followers_count    0.35733957   -0.001358509     1.000000000
## favourites_count   0.03683789    0.028239280    -0.002667248
## friends_count      0.15308667   -0.013025551     0.043717263
##                 favourites_count friends_count
## listed_count          0.036837887    0.15308667
## statuses_count        0.028239280   -0.01302555
## followers_count      -0.002667248    0.04371726
## favourites_count      1.000000000    0.17700534
## friends_count         0.177005339    1.00000000
```

The number of followers has always been considered as a decent indication of a Twitter account's popularity. Though it's not so hard to have a big amount of followers, it shows people's attitude towards their social media's reputation. I believe there could have some correlations between accounts with large number of followers and the number of their tweets about President Trump.

In above the matrix plot shows the relationship among listed count, statuses count, followers count, favourites count and friends count. From the output, I can see there's slight linear relation between followers count and listed count and statuses count also have slight linear relation with friends count.

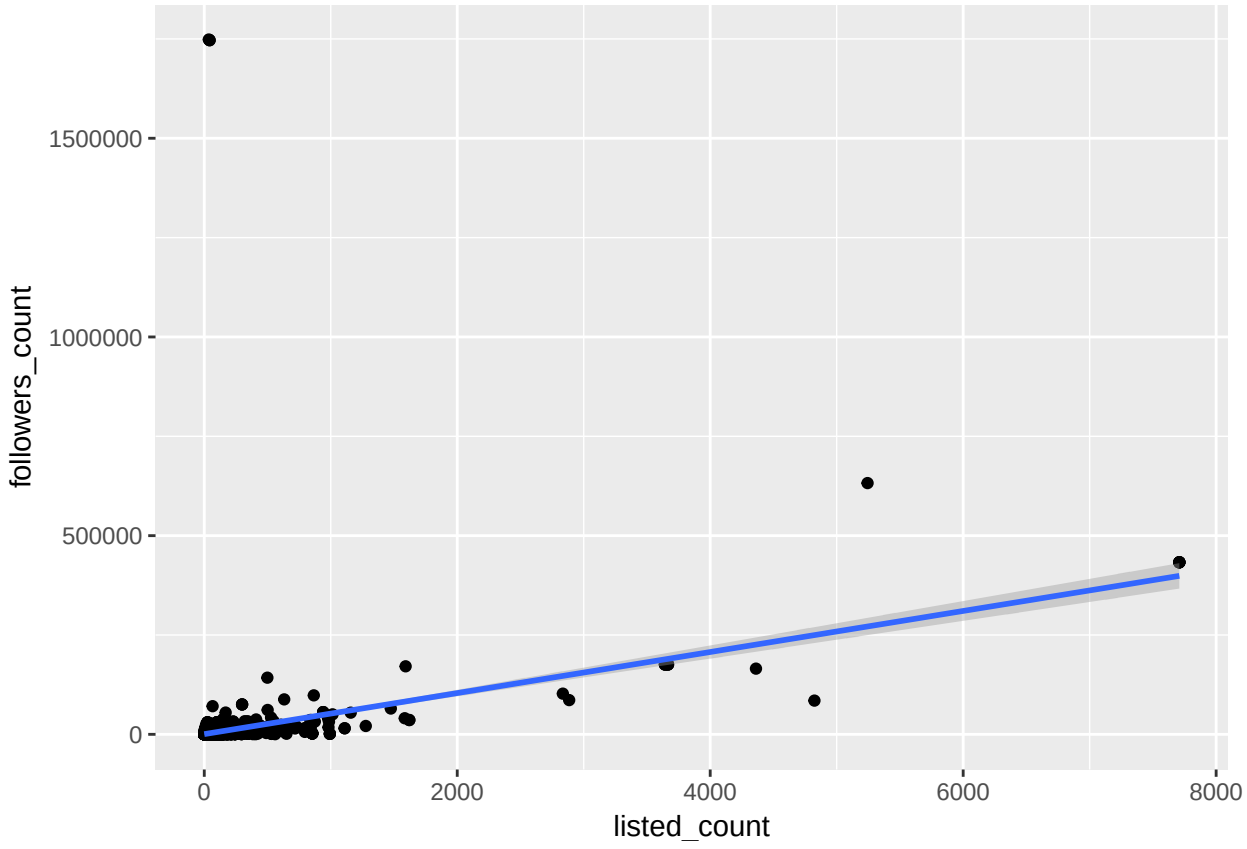
Therefore I believe a linear regression between followers count and listed count would be useful and good looking at my analysis.

```
###plotting linear regression for the dataset
model_fl <- lm(followers_count~listed_count,data=total_us)
summary(model_fl)
```

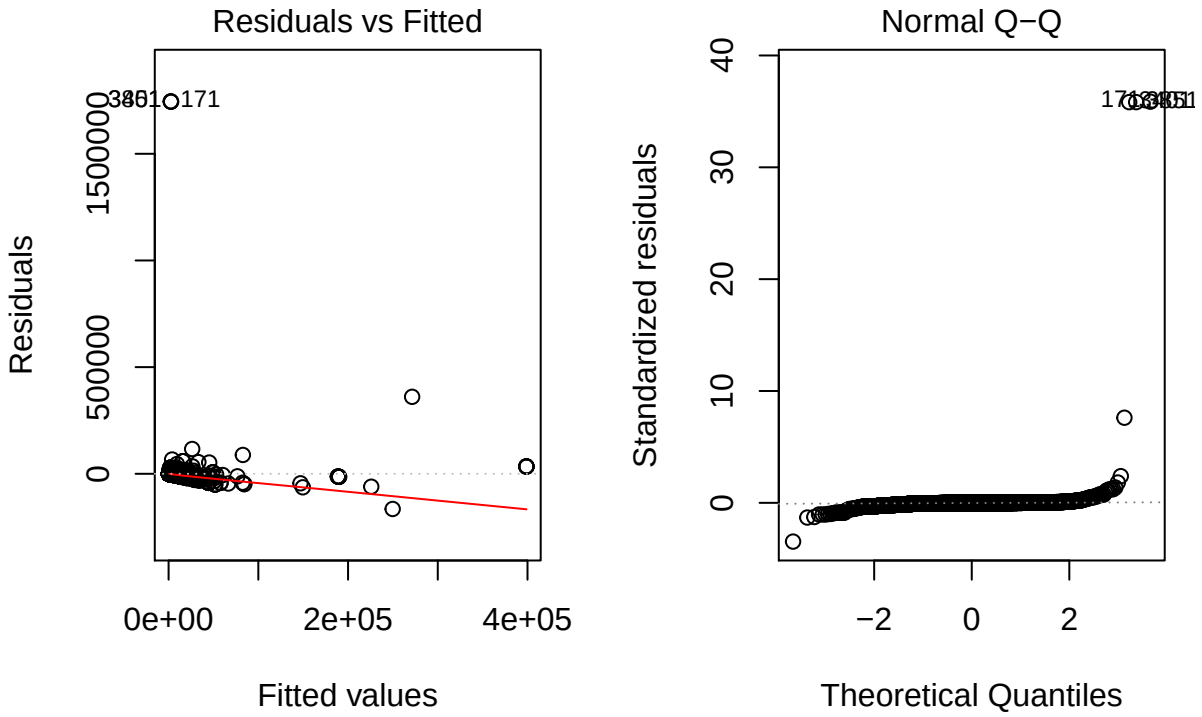
```
##
## Call:
```

```
## lm(formula = followers_count ~ listed_count, data = total_us)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -164852   -1362    -551    -254 1746106
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   532.863    787.149   0.677    0.498
## listed_count   51.660      2.137  24.171   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 48690 on 3991 degrees of freedom
## Multiple R-squared:  0.1277, Adjusted R-squared:  0.1275
## F-statistic: 584.2 on 1 and 3991 DF,  p-value: < 2.2e-16
```

```
regplot <- ggplot(total_us,aes(x=listed_count,y=followers_count))+geom_point()+geom_smooth(method = "lm
regplot
```




```
#analysis of regression goes here
par(mfrow=c(1,2))
plot(model_fl,1)
plot(model_fl,2)
```



The model is not very good with two significant variables and the R-squared is 0.1275, which means there are not so much connection between listed count and the number of followers. Also Residuals vs Fitted plot and Normal Q-Q plot are not good, indicating that there are heteroskedasticity and unlinear problems in the model.

5 Text Study

By utilizing the word cloud interface, I'm introducing all my tweets toward word clouds and conduct text mining on them.

First of all, since there are too many meaningless words in my data, I need to clean these distracting words by building a corpus and specify the certain filters to get the best-cleaned data for word clouding.

```
#wordcloud

total_us_word <- total_us     #import data
total_us_word <- total_us_word[total_us_word$lang=="en",]     #filter out only the english data
total_us_word <- total_us_word[total_us_word$geo_enabled==TRUE,]     #filter out data with geocode enab
total_us_word <- total_us_word[total_us_word$place_lat >25 &
                    total_us_word$place_lat <50 & total_us_word$place_lon > -125 &
                    total_us_word$place_lon< -66,]     #filter out extreme value
total_us_word$geo_enabled <- NULL

# build a corpus, and specify the source to be character vectors
myCorpus <- Corpus(VectorSource(total_us_word$text))
# remove anything other than English letters or space(!!!)
removeNumPunct <- function(x) gsub("[^[:alpha:][:space:]]*", "", x)
myCorpus <- tm_map(myCorpus, content_transformer(removeNumPunct))
# convert to lower case
myCorpus <- tm_map(myCorpus, content_transformer(tolower))
# remove URLs
```

```
removeURL <- function(x) gsub("http[^[:space:]]*", "", x)
myCorpus <- tm_map(myCorpus, content_transformer(removeURL))
# remove stopwords
myStopwords <- c(stopwords('english'),"use", "see", "used", "via", "amp","im")
myCorpus <- tm_map(myCorpus, removeWords, myStopwords)
# remove extra whitespace
myCorpus <- tm_map(myCorpus, stripWhitespace)
# remove punctuation
myCorpus <- tm_map(myCorpus, removePunctuation)

# Build Term Document Matrix
tdm <- TermDocumentMatrix(myCorpus, control = list(wordLengths = c(1, Inf)))
term.freq <- rowSums(as.matrix(tdm))
term.freq2 <- subset(term.freq, term.freq >= 5)
df <- data.frame(term = names(term.freq2), freq = term.freq2)
m <- as.matrix(tdm)
```

```
# calculate the frequency of words and sort it by frequency
word.freq <- sort(rowSums(m), decreasing = T)

# plot word cloud
wordcloud(words = names(word.freq),
          freq = word.freq,
          max.words = 50,
          min.freq = 3,
          scale = c(4.5, 1),
          colors = brewer.pal(8, "Dark2"),
          random.color = T,
          random.order = F)
```

```
## Warning in wordcloud(words = names(word.freq), freq = word.freq, max.words
## = 50, : realdonaldtrump could not be fit on page. It will not be plotted.
```

```
## Warning in wordcloud(words = names(word.freq), freq = word.freq, max.words
## = 50, : never could not be fit on page. It will not be plotted.
```


```
#analysis of wordcloud goes here
```

Also, on the other hand, let's see President Trump his word cloud analysis. Since I don't have President Trump's previous tweets, I decide to do use Twitter API again and then capture his past 3000 tweets as my new dataset.

```
###Here I used a new Twitter Application Api keys for different purpose, I also shouldn't have included

#api_key             <- "RE3jcprdcB3YwIXwcCG2rHPmj"
#api_secret          <- "rPeMG3nqOwDVKtXOuIsd4czCZjJd1eBh9BHqPWuUf8xBDaIY4j"
#access_token        <- "3134023545-cJbCbhXSeklrnVtEKU2yvv3bTI7APn93As93WiS"
#access_token_secret <- "glaUQunnz9e2QQT2gpKNgJHWtqKGRBK12JP9HtN44rCdT"
#setup_twitter_oauth(api_key, api_secret, access_token, access_token_secret)

###I comment this chunk out for avoid unneccessary time when kniting the whole project, but you are wel
###uncomment them for running the code throughly. However, since I will store Trump's tweets later and
###There won't be any problem you just leave this chunk commented out.

#user_id <- "@realDonaldTrump"
#trump_tweets_raw <- userTimeline(user_id, n = 3200)

#save(trump_tweets_raw, file = "trump_tweets_raw.Rdata")
```

```
load("trump_tweets_raw.Rdata")     #loading captured data that stored before

trump_tweets_df <- twListToDF(strip_retweets(trump_tweets_raw, strip_manual = TRUE, strip_mt = TRUE))

trump_tweets_df$text <- iconv(trump_tweets_df$text, 'latin1', 'ASCII', 'byte')      # normalize data
```

```r
Trump_Corpus <- Corpus(VectorSource(trump_tweets_df$text))      # build a corpus, and specify the sourc

removeNumPunct <- function(x) gsub("[^[:alpha:][:space:]]*", "", x)     # remove anything other than En
Trump_Corpus <- tm_map(Trump_Corpus, content_transformer(removeNumPunct))

Trump_Corpus <- tm_map(Trump_Corpus, stripWhitespace)     # remove extra whitespace

Trump_Corpus <- tm_map(Trump_Corpus, content_transformer(tolower))      # convert to lower case

myStopwords <- c(stopwords('english'),"use", "see", "used", "via", "amp","im")      # remove stopwords
Trump_Corpus <- tm_map(Trump_Corpus, removeWords, myStopwords)

rm_url   <- function(x) gsub("http[^[:space:]]*", "", x)      # remove urls
Trump_Corpus <- tm_map(Trump_Corpus, content_transformer(rm_url))

Trump_Corpus <- tm_map(Trump_Corpus, removePunctuation)     # remove punctuation

# build term document matrix
Trump_tdm <- TermDocumentMatrix(Trump_Corpus, control = list(wordLengths = c(1, Inf)))
Trump_term.freq <- rowSums(as.matrix(Trump_tdm))
Trump_term.freq <- subset(Trump_term.freq, Trump_term.freq >= 5)
df <- data.frame(term = names(Trump_term.freq), freq = Trump_term.freq)
Trump_m  <- as.matrix(Trump_tdm)

# calculate the frequency of words and sort it by frequency
Trump_word_freq <- sort(rowSums(Trump_m), decreasing = T)

wordcloud(words = names(Trump_word_freq),
          freq  = Trump_word_freq,
          scale = c(4.5, 1),
          max.words = 50,
          min.freq = 3,
          colors = brewer.pal(8, "Dark2"),
          random.color = T,
          random.order = F)
```

Surprisingly, we can easily find some similar words and attitudes in both tweets mentioning President Trump and President Trump's own tweets, such as "job","hiring","working" those words that are related employment. I believe this somehow reflects that President Trump's new policy and work towards jobs are causing people talking about it and there might still have problem with the rate of unemployment.

6 Shiny Application

For the Shiny application, I created a navbar with all the graphs that I created earlier in different tabs, including Maps, statistic plots, and word cloud comparison. Descriptions of each graph are also included

All shiny codes are included in the folder called "shiny_app".

Shiny application are available through here: https://sibozhu.shinyapps.io/shiny_app/

7 Conclusion

In this analysis of Tweets mentioning President Trump, I captured a large number of tweets and researched in the density of those tweets that are posted in different states and cities, making plots to compared in different situations. Also, I tried to find the relations among followers count, listed count and friends count, which could represent its popularity and reputation among social media, to check their words influence toward topics about President Trump. Finally, I plotted the most frequent words in my tweets data with their word cloud and comparing it with President Trump's own word cloud.