In this post, let's mine tweets and analyze their sentiment using R. We will use Tableau to visualize our results. We will see spatial-temporal distribution of tweets, cities and states with top number of tweets and we will also map the sentiment of the tweets. This will help us to see in which areas his comments are accepted as positive and where they are perceived as negative.

Load required packages:

```
library(twitteR)
library(ROAuth)
require(RCurl)
library(stringr)
library(tm)
library(ggmap)
library(dplyr)
library(plyr)
library(tm)
library(wordcloud)
```

## Get Twitter authentication

All information below is obtained from twitter developer account. We will set working directory to save our authentication.

```
key="hidden"
secret="hidden"
setwd("/text_mining_and_web_scraping")

download.file(url="http://curl.haxx.se/ca/cacert.pem",
              destfile="/text_mining_and_web_scraping/cacert.pem",
              method="auto")
authenticate <- OAuthFactory$new(consumerKey=key,
                                 consumerSecret=secret,
                                 requestURL="https://api.twitter.com/oauth/request_token",
                                 accessURL="https://api.twitter.com/oauth/access_token",
                                 authURL="https://api.twitter.com/oauth/authorize")
setup_twitter_oauth(key, secret)
save(authenticate, file="twitter authentication.Rdata")
```

## Get sample tweets from various cities

Let's scrape most recent tweets from various cities across the US. Let's request 2000 tweets from each city. We will need the latitude and longitude of each city.

```
N=2000  # tweets to request from each query
S=200  # radius in miles
lats=c(38.9,40.7,37.8,39,37.4,28,30,42.4,48,36,32.3,33.5,34.7,33.8,37.2,41.2,46.8,
       46.6,37.2,43,42.7,40.8,36.2,38.6,35.8,40.3,43.6,40.8,44.9,44.9)

lons=c(-77,-74,-122,-105.5,-122,-82.5,-98,-71,-122,-115,-86.3,-112,-92.3,-84.4,-93.3,
       -104.8,-100.8,-112,
-93.3,-89,-84.5,-111.8,-86.8,-92.2,-78.6,-76.8,-116.2,-98.7,-123,-93)

#cities=DC,New York,San Fransisco,Colorado,Mountainview,Tampa,Austin,Boston,
#       Seatle,Vegas,Montgomery,Phoenix,Little Rock,Atlanta,Springfield,
#       Cheyenne,Bisruk,Helena,Springfield,Madison,Lansing,Salt Lake City,Nashville
#       Jefferson City,Raleigh,Harrisburg,Boise,Lincoln,Salem,St. Paul

donald=do.call(rbind,lapply(1:length(lats), function(i) searchTwitter('Donald+Trump',
              lang="en",n=N,resultType="recent",
              geocode=paste(lats[i],lons[i],paste0(S,"mi"),sep=","))))
```

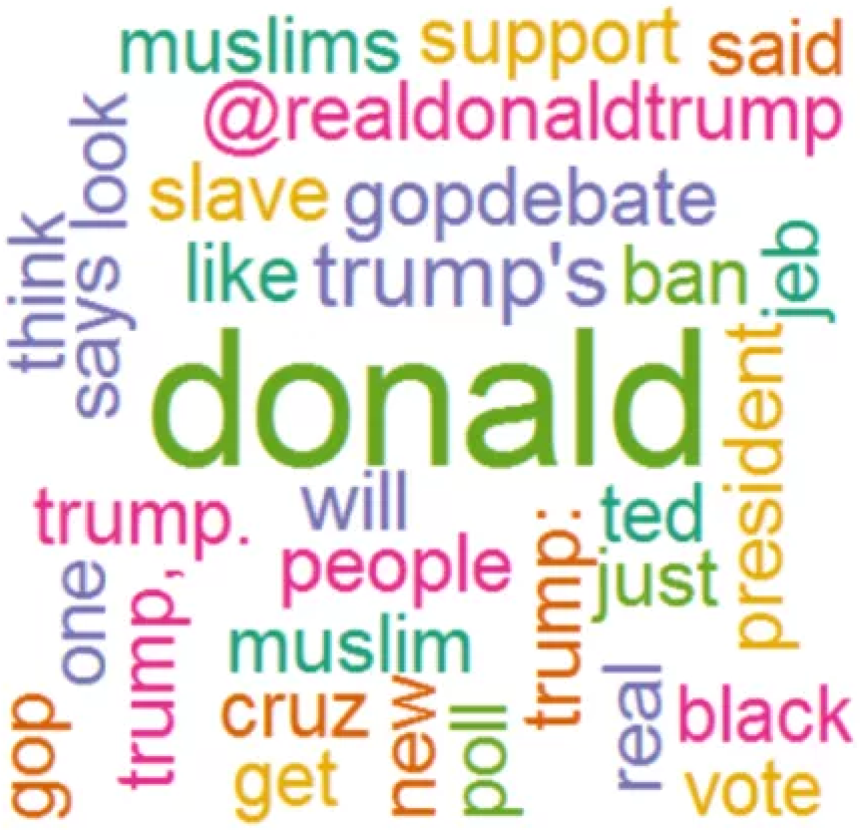
Let’s get the latitude and longitude of each tweet, the tweet itself, how many times it was re-twitted and favorited, the date and time it was twitted, etc.

```
donaldlat=sapply(donald, function(x) as.numeric(x$getLatitude()))
donaldlat=sapply(donaldlat, function(z) ifelse(length(z)==0,NA,z))

donaldlon=sapply(donald, function(x) as.numeric(x$getLongitude()))
donaldlon=sapply(donaldlon, function(z) ifelse(length(z)==0,NA,z))

donalddate=lapply(donald, function(x) x$getCreated())
donalddate=sapply(donalddate,function(x) strftime(x, format="%Y-%m-%d %H:%M:%S",tz = "UTC"))

donaldtext=sapply(donald, function(x) x$getText())
donaldtext=unlist(donaldtext)

isretweet=sapply(donald, function(x) x$getIsRetweet())
retweeted=sapply(donald, function(x) x$getRetweeted())
retweetcount=sapply(donald, function(x) x$getRetweetCount())

favoritecount=sapply(donald, function(x) x$getFavoriteCount())
favorited=sapply(donald, function(x) x$getFavorited())

data=as.data.frame(cbind(tweet=donaldtext,date=donalddate,lat=donaldlat,lon=donaldlon,
                           isretweet=isretweet,retweeted=retweeted,
retweetcount=retweetcount,favoritecount=favoritecount,favorited=favorited))
```

First, let’s create a word cloud of the tweets. A word cloud helps us to visualize the most common words in the tweets and have a general feeling of the tweets.

```
# Create corpus
corpus=Corpus(VectorSource(data$tweet))

# Convert to lower-case
corpus=tm_map(corpus,tolower)

# Remove stopwords
corpus=tm_map(corpus,function(x) removeWords(x,stopwords()))

# convert corpus to a Plain Text Document
corpus=tm_map(corpus,PlainTextDocument)

col=brewer.pal(6,"Dark2")
wordcloud(corpus, min.freq=25, scale=c(5,2),rot.per = 0.25,
          random.color=T, max.word=45, random.order=F,colors=col)
```

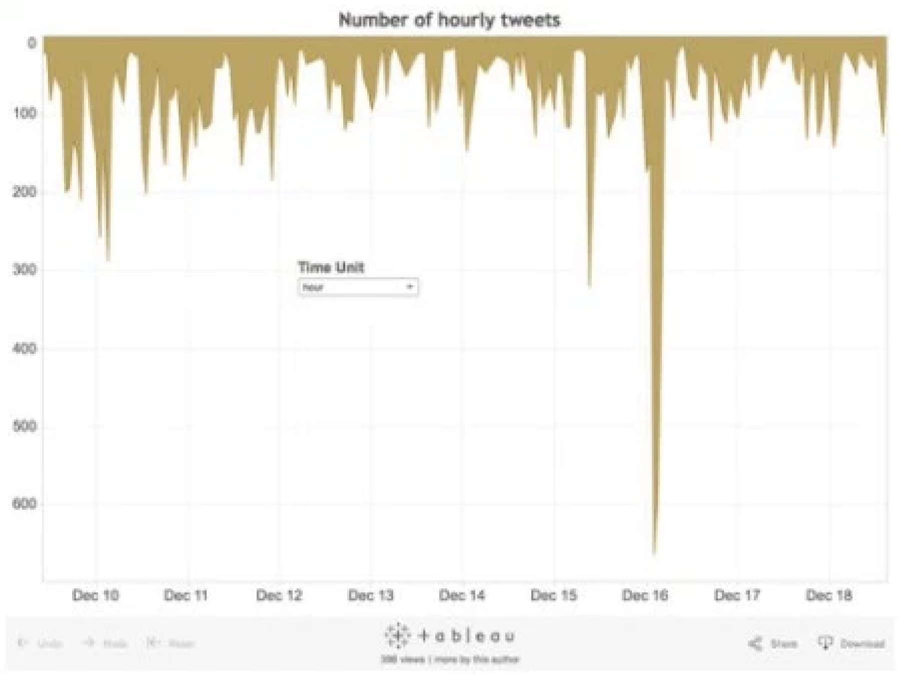
Here is the word cloud:



We see from the word cloud that among the most frequent words in the tweets are ‘muslim’, ‘muslims’, ‘ban’. This suggests that most tweets were on Trump’s recent idea of temporarily banning Muslims from entering the US.

The dashboard below shows time series of the number of tweets scraped. We can change the time unit between hour and day and the dashboard will change based on the selected time unit. Pattern of number of tweets over time helps us to drill in and see how each activities/campaigns are being perceived.

Here is the screenshot. (View it live in this link)



## Getting address of tweets

Since some tweets do not have lat/lon values, we will remove them because we want geographic information to show the tweets and their attributes by state, city and zip code.

```
data=filter(data, !is.na(lat),!is.na(lon))
lonlat=select(data,lon,lat)
```

Let’s get full address of each tweet location using the google maps API. The `ggmaps` package is what enables us to get the street address, city, zipcode and state of the tweets using the longitude and latitude of the tweets. Since the google maps API does not allow more than 2500 queries per day, I used a couple of machines to reverse geocode the latitude/longitude information in a full address. However, I was not lucky enough to reverse geocode all of the tweets I scraped. So, in the following visualizations, I am showing only some percentage of the tweets I scraped that I was able to reverse geocode.

```
result <- do.call(rbind, lapply(1:nrow(lonlat),
                     function(i) revgeocode(as.numeric(lonlat[i,1:2]))))
```

If we see some of the values of result, we see that it contains the full address of the locations where the tweets were posted.

```
result[1:5,]
     [,1]
[1,] "1778 Woodglo Dr, Asheboro, NC 27205, USA"
[2,] "1550 Missouri Valley Rd, Riverton, WY 82501, USA"
[3,] "118 S Main St, Ann Arbor, MI 48104, USA"
[4,] "322 W 101st St, New York, NY 10025, USA"
[5,] "322 W 101st St, New York, NY 10025, USA"
```

So, we will apply some regular expression and string manipulation to separate the city, zip code and state into different columns.

```
data2=lapply(result,  function(x) unlist(strsplit(x,",")))
address=sapply(data2,function(x) paste(x[1:3],collapse=''))
city=sapply(data2,function(x) x[2])
stzip=sapply(data2,function(x) x[3])
zipcode = as.numeric(str_extract(stzip,"[0-9]{5}"))
state=str_extract(stzip,"[:alpha:]{2}")
data2=as.data.frame(list(address=address,city=city,zipcode=zipcode,state=state))
```

Concatenate data2 to data:

```
data=cind(data,data2)
```

Some text cleaning:

```
tweet=data$tweet
tweet_list=lapply(tweet, function(x) iconv(x, "latin1", "ASCII", sub=""))
tweet_list=lapply(tweet, function(x) gsub("htt.*",' ',x))
tweet=unlist(tweet)
data$tweet=tweet
```

We will use lexicon based sentiment analysis. A list of positive and negative opinion words or sentiment words for English was downloaded from here.

```
positives= readLines("positivewords.txt")
negatives = readLines("negativewords.txt")
```

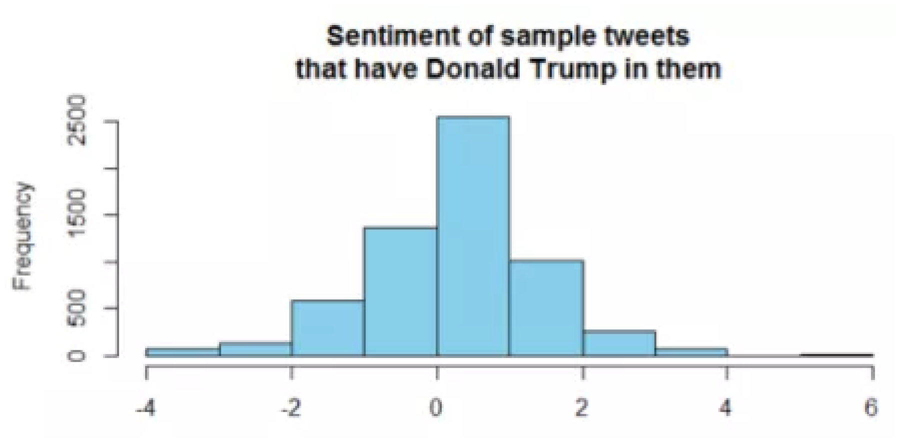
First, let’s have a wrapper function that calculates sentiment scores.

```
sentiment_scores = function(tweets, positive_words, negative_words, .progress='none'){
  scores = laply(tweets,
                 function(tweet, positive_words, negative_words){
                 tweet = gsub("[[:punct:]]", "", tweet)    # remove punctuation
                 tweet = gsub("[[:cntrl:]]", "", tweet)   # remove control characters
                 tweet = gsub('\d+', '', tweet)          # 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
                   tweet = sapply(tweet, tryTolower)
                   # split sentence into words with str_split function from stringr package
                   word_list = str_split(tweet, "\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_matches, negative_matches, .progress=.progress )
  return(scores)
}

score = sentiment_scores(tweet, positives, negatives, .progress='text')
data$score=score
```

Let’s plot a histogram of the sentiment score:

```
hist(score,xlab=" ",main="Sentiment of sample tweetsn that have Donald Trump in them ",
     border="black",col="skyblue")
```

Here is the plot:

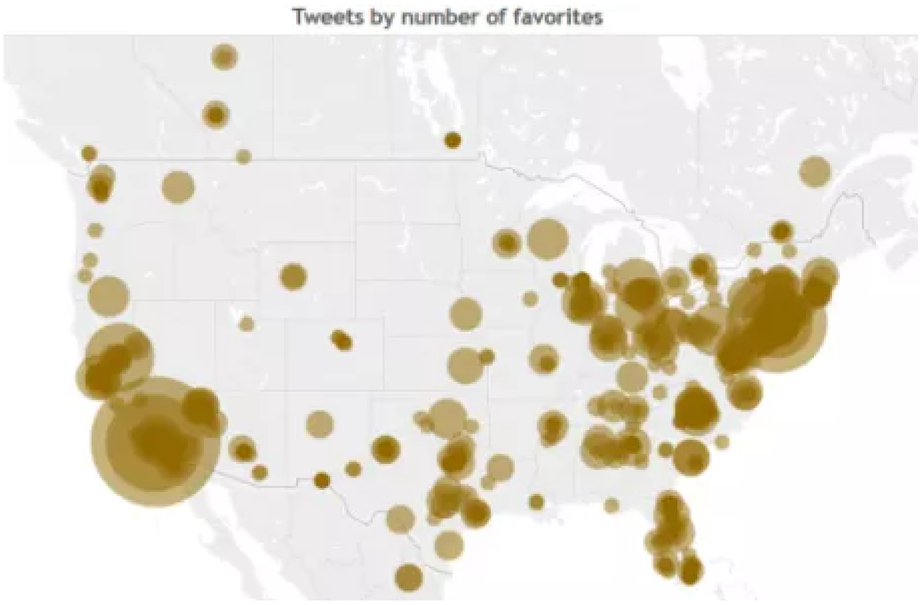


We see from the histogram that the sentiment is slightly positive. Using Tableau, we will see the spatial distribution of the sentiment scores.

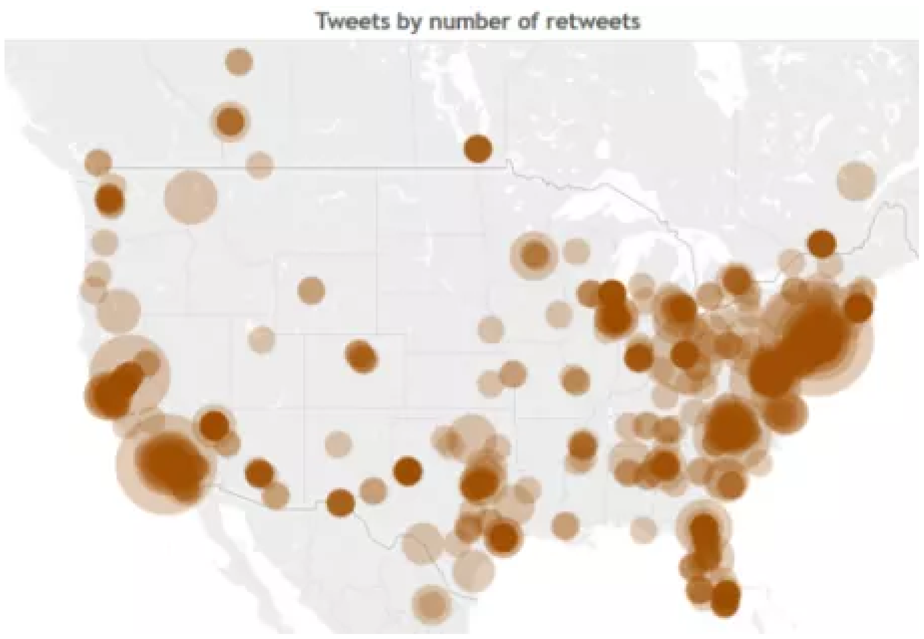
## Save the data as csv file and import it to Tableau

The map below shows the tweets that I was able to reverse geocode. The size is proportional to the number of favorites each tweet got. In the interactive map, we can hover each circle and read the tweet, the address it was tweeted from, and the date and time it was posted.

Here is the screenshot (View it live in this link)

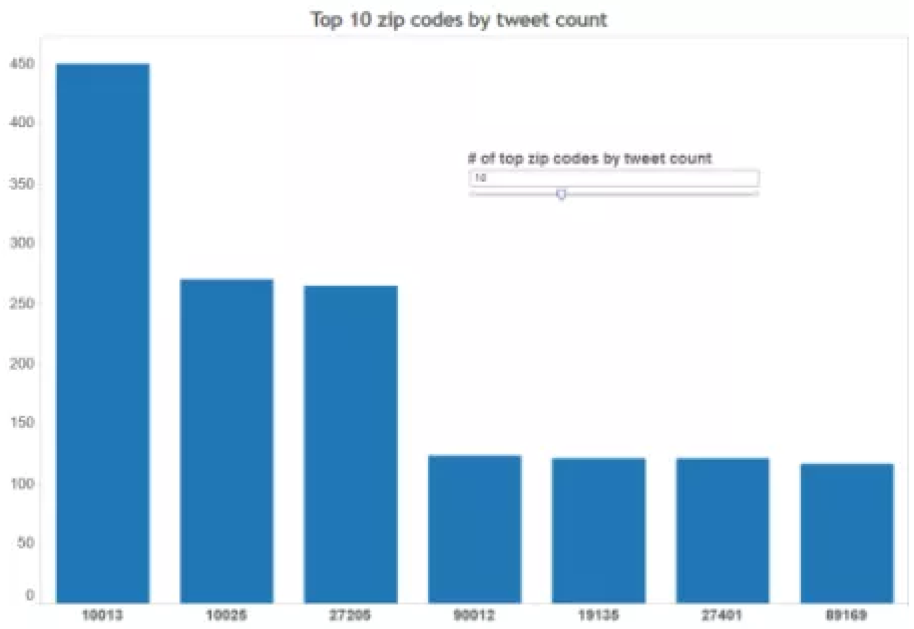


Similarly, the dashboard below shows the tweets and the size is proportional to the number of times each tweet was retweeted. Here is the screenshot (View it live in this link)

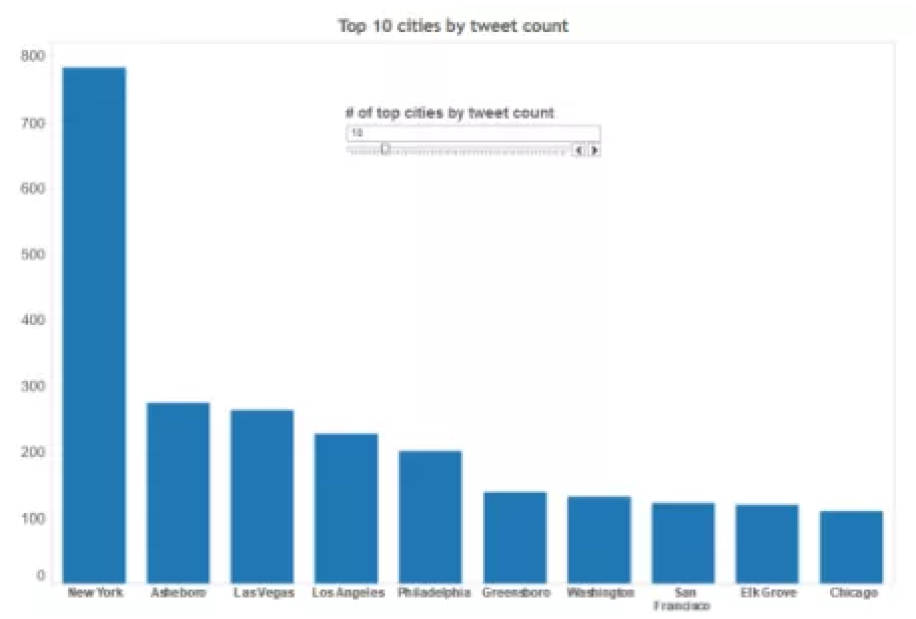


In the following three visualizations, top zip codes, cities and states by the number of tweets are shown. In the interactive map, we can change the number of zip codes, cities and states to display by using the scrollbars shown in each viz. These visualizations help us to see the distribution of the tweets by zip code, city and state.

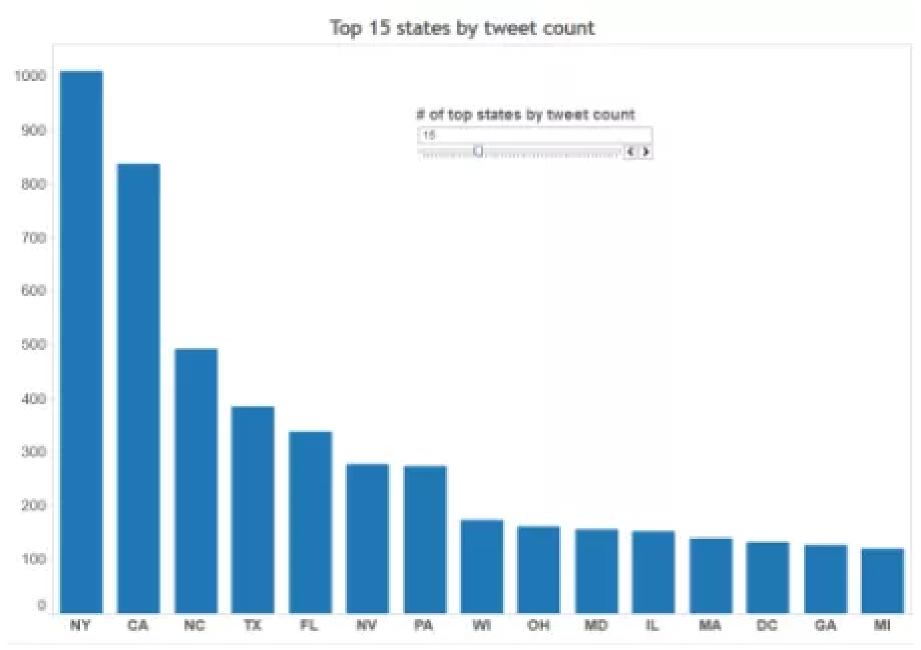
**By zip code**
Here is the screenshot (View it live in this link)



**By city**
Here is the screenshot (View it live in this link)



**By state**
Here is the screenshot (View it live in this link)

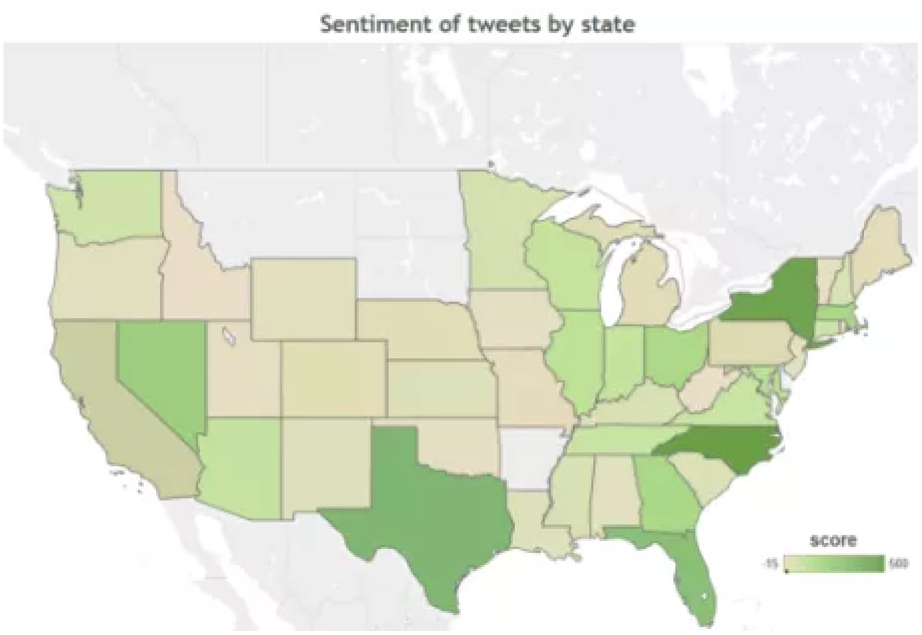


## Sentiment of tweets

Sentiment analysis has myriads of uses. For example, a company may investigate what customers like most about the company's product, and what are the issues the customers are not satisfied with? When a company releases a new product, has the product been perceived positively or negatively? How does the sentiment of the customers vary across space and time? In this post, we are evaluating, the sentiment of tweets that we scraped on Donald Trump.

The viz below shows the sentiment score of the reverse geocoded tweets by state. We see that the tweets have highest positive sentiment in NY, NC and Tx.

Here is the screenshot (View it live in this link)



## Summary

In this post, we saw how to integrate R and Tableau for text mining, sentiment analysis and visualization. Using these tools together enables us to answer detailed questions.

We used a sample from the most recent tweets that contain Donald Trump and since I was not able to reverse geocode all the tweets I scraped because of the constraint imposed by google maps API, we just used about 6000 tweets. The average sentiment is slightly above zero. Some states show strong positive sentiment. However, statistically speaking, to make robust conclusions, mining ample size sample data is important.

The accuracy of our sentiment analysis depends on how fully the words in the the tweets are included in the lexicon. More over, since tweets may contain slang, jargon and collequial words which may not be included in the lexicon, sentiment analysis needs careful evaluation.