Data607_Final Project Rajan

Krishna Rajan

5/9/2018

Project Outline: I work for Delphi Technologies which a leading automotive supplier of powertrain components(www.delphi.com), twitter handle (@delphitech) for this project I was tasked by my director of marketing to perform an analysis of Delphi technology on twitter. The direction given was, what was the most tweeted words about Delphi? What was the sentiments from twitter followers for our company?Also questions regarding top tweeted tweets, & followers summary?

Techniques used for this project: 1)Text Mining (extract, mine, clean & stem)

2)Topic Modelling (Frequent words, word cloud)

3)Analysis(Word association,Network of terms,sentiment analysis,followers & top tweets analysis)

4)Maping Twitter followers.

I started the project by building a map using geocode location of delphitech twitter followers.

Geo-Mapping Twitter Users

First step is to authenticate twitter data with twitter API

```
##GEO-MAP Twitter FOLLOWERS
require(twitteR)
```

```
## Loading required package: twitteR
```

```
library(ROAuth)
require(data.table)
```

```
## Loading required package: data.table
```

```
require(RJSONIO)
```

```
## Loading required package: RJSONIO
```

```
require(leaflet)
```

```
## Loading required package: leaflet
```

```
## Warning: package 'leaflet' was built under R version 3.4.4
```

```
setup_twitter_oauth("jmbwTtQL7ZoW4rxCMOmUOOmXJ", "3X5v1OUEs67lw25vg6rK7muS6VmKFTAQGMp
kVOVF2JQgHzZikI", "51183723-uKDVYied6jRISD0aJLstyJkZLW8Ykqttqos8mbMJI", "tkmsZd790aSy
6rXgDlbwfUWLtHC6Wjxm68pQE1uumO21I")
```

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

```
Delphi <- getUser("delphitech")
```

Les me pose an interesting question, where is delphitech's followers located? To answer the question, we will first create a R function called get_followers. The function can download follower information from API, remove users whose location information is blank or contains special characters. Notice that Twitter API has rate limit,

```
##organise the data
Delphi_follower_IDs <-Delphi$getFollowers(n=200)

##doubel check to see if most fo the followers have been extracted
length(Delphi_follower_IDs)
```

```
## [1] 178
```

```
##turn followers into a data frame
Delphi_followers_df <- do.call("rbind", lapply(Delphi_follower_IDs,as.data.frame))
##Take a quick peep at the data.
head(Delphi_followers_df$location, 10)
```

```
##  [1] ""                               "San Marco"
##  [3] "Cappadocia, Abruzzo"            ""
##  [5] "Rhineland-Palatinate, Germany" ""
##  [7] "Canada"                         "Ann Arbor, MI"
##  [9] "china "                         "Jaipur, India"
## Remove followers who have not listed their followers.
Delphi_followers_df<-subset(Delphi_followers_df, location!="")

##remove any instances of % since that character doesn't play well with the API.
Delphi_followers_df$location<-gsub("%", " ",Delphi_followers_df$location)
```

The location information is stored in the column named location. We can match the cities and states with exact coordinates through Google Map API. To do that, obtain a key from Google Maps Geocoding API. (https://developers.google.com/maps/documentation/geocoding/get-api-key (https://developers.google.com/maps/documentation/geocoding/get-api-key)). There is a limit of 2,500 coordinates per day if you are a standard Google Map API user.

```
#Install key package helpers:
source("https://raw.githubusercontent.com/LucasPuente/geocoding/master/geocode_helper
s.R")
#Install modified version of the geocode function
#(that now includes the api_key parameter):
source("https://raw.githubusercontent.com/LucasPuente/geocoding/master/modified_geoco
de.R")
geocode_apply<-function(x){
    geocode(x, source = "google", output = "all", api_key="AIzaSyDjTT7ysuxenX8UauY3VU
gqhjOxUBmsIKc")
}
```

Apply the function geocode_apply to get cordinates

```
geocode_results<-sapply(Delphi_followers_df$location, geocode_apply, simplify = F)
```

```
## Information from URL : https://maps.googleapis.com/maps/api/geocode/json?address=S
an%20Marco&sensor=false&key=AIzaSyDjTT7ysuxenX8UauY3VUgqhjOxUBmsIKc
```

```
## Information from URL : https://maps.googleapis.com/maps/api/geocode/json?address=C
appadocia,%20Abruzzo&sensor=false&key=AIzaSyDjTT7ysuxenX8UauY3VUgqhjOxUBmsIKc
```

```
## Information from URL : https://maps.googleapis.com/maps/api/geocode/json?address=R
hineland-Palatinate,%20Germany&sensor=false&key=AIzaSyDjTT7ysuxenX8UauY3VUgqhjOxUBmsI
Kc
```

```
## Information from URL : https://maps.googleapis.com/maps/api/geocode/json?address=C
anada&sensor=false&key=AIzaSyDjTT7ysuxenX8UauY3VUgqhjOxUBmsIKc
```

```
## Information from URL : https://maps.googleapis.com/maps/api/geocode/json?address=A
nn%20Arbor,%20MI&sensor=false&key=AIzaSyDjTT7ysuxenX8UauY3VUgqhjOxUBmsIKc
```

```
## Information from URL : https://maps.googleapis.com/maps/api/geocode/json?address=c
hina%20&sensor=false&key=AIzaSyDjTT7ysuxenX8UauY3VUgqhjOxUBmsIKc
```

```
## Information from URL : https://maps.googleapis.com/maps/api/geocode/json?address=J
aipur,%20India&sensor=false&key=AIzaSyDjTT7ysuxenX8UauY3VUgqhjOxUBmsIKc
```

```
## Information from URL : https://maps.googleapis.com/maps/api/geocode/json?address=G
lobal&sensor=false&key=AIzaSyDjTT7ysuxenX8UauY3VUgqhjOxUBmsIKc
```

```
## Information from URL : https://maps.googleapis.com/maps/api/geocode/json?address=H
ong%20Kong&sensor=false&key=AIzaSyDjTT7ysuxenX8UauY3VUgqhjOxUBmsIKc
```

```
## Information from URL : https://maps.googleapis.com/maps/api/geocode/json?address=W
estpark%20Shannon%20Co.Clare&sensor=false&key=AIzaSyDjTT7ysuxenX8UauY3VUgqhjOxUBmsIKc
```

```
## .Information from URL : https://maps.googleapis.com/maps/api/geocode/json?address=
Gothenburg,%20Sweden&sensor=false&key=AIzaSyDjTT7ysuxenX8UauY3VUgqhjOxUBmsIKc
## .Information from URL : https://maps.googleapis.com/maps/api/geocode/json?address=
Berkeley,%20CA&sensor=false&key=AIzaSyDjTT7ysuxenX8UauY3VUgqhjOxUBmsIKc
## .Information from URL : https://maps.googleapis.com/maps/api/geocode/json?address=
Headquartered%20in%20Beijing&sensor=false&key=AIzaSyDjTT7ysuxenX8UauY3VUgqhjOxUBmsIKc
## Information from URL : https://maps.googleapis.com/maps/api/geocode/json?address=M
umbai&sensor=false&key=AIzaSyDjTT7ysuxenX8UauY3VUgqhjOxUBmsIKc
## Information from URL : https://maps.googleapis.com/maps/api/geocode/json?address=Z
apopan,%20Jalisco&sensor=false&key=AIzaSyDjTT7ysuxenX8UauY3VUgqhjOxUBmsIKc
## Information from URL : https://maps.googleapis.com/maps/api/geocode/json?address=A
ntibes,%20France&sensor=false&key=AIzaSyDjTT7ysuxenX8UauY3VUgqhjOxUBmsIKc
## Information from URL : https://maps.googleapis.com/maps/api/geocode/json?address=D
en%20Haag%20NL&sensor=false&key=AIzaSyDjTT7ysuxenX8UauY3VUgqhjOxUBmsIKc
## Information from URL : https://maps.googleapis.com/maps/api/geocode/json?address=R
io%20de%20Janeiro,%20Brazil&sensor=false&key=AIzaSyDjTT7ysuxenX8UauY3VUgqhjOxUBmsIKc
## Information from URL : https://maps.googleapis.com/maps/api/geocode/json?address=F
rance&sensor=false&key=AIzaSyDjTT7ysuxenX8UauY3VUgqhjOxUBmsIKc
## Information from URL : https://maps.googleapis.com/maps/api/geocode/json?address=R
io%20de%20Janeiro,%20Brazil&sensor=false&key=AIzaSyDjTT7ysuxenX8UauY3VUgqhjOxUBmsIKc
## .Information from URL : https://maps.googleapis.com/maps/api/geocode/json?address=
United%20States&sensor=false&key=AIzaSyDjTT7ysuxenX8UauY3VUgqhjOxUBmsIKc
## .Information from URL : https://maps.googleapis.com/maps/api/geocode/json?address=
Addison%20Township,%20MI&sensor=false&key=AIzaSyDjTT7ysuxenX8UauY3VUgqhjOxUBmsIKc
```

```
## .Information from URL : https://maps.googleapis.com/maps/api/geocode/json?address=
Ju%C3%A1rez,%20Chihuahua&sensor=false&key=AIzaSyDjTT7ysuxenX8UauY3VUgqhjOxUBmsIKc
## .Information from URL : https://maps.googleapis.com/maps/api/geocode/json?address=
Chicago,%20IL&sensor=false&key=AIzaSyDjTT7ysuxenX8UauY3VUgqhjOxUBmsIKc
## Information from URL : https://maps.googleapis.com/maps/api/geocode/json?address=S
unny%20Okanagan%20%20&sensor=false&key=AIzaSyDjTT7ysuxenX8UauY3VUgqhjOxUBmsIKc
## .Information from URL : https://maps.googleapis.com/maps/api/geocode/json?address=
Alton,%20England&sensor=false&key=AIzaSyDjTT7ysuxenX8UauY3VUgqhjOxUBmsIKc
## Information from URL : https://maps.googleapis.com/maps/api/geocode/json?address=W
orldwide&sensor=false&key=AIzaSyDjTT7ysuxenX8UauY3VUgqhjOxUBmsIKc
## Information from URL : https://maps.googleapis.com/maps/api/geocode/json?address=E
urope&sensor=false&key=AIzaSyDjTT7ysuxenX8UauY3VUgqhjOxUBmsIKc
## Information from URL : https://maps.googleapis.com/maps/api/geocode/json?address=S
pain&sensor=false&key=AIzaSyDjTT7ysuxenX8UauY3VUgqhjOxUBmsIKc
## Information from URL : https://maps.googleapis.com/maps/api/geocode/json?address=N
ew%20Jersey&sensor=false&key=AIzaSyDjTT7ysuxenX8UauY3VUgqhjOxUBmsIKc
## Information from URL : https://maps.googleapis.com/maps/api/geocode/json?address=B
arcelona&sensor=false&key=AIzaSyDjTT7ysuxenX8UauY3VUgqhjOxUBmsIKc
## .Information from URL : https://maps.googleapis.com/maps/api/geocode/json?address=
Johannesburg&sensor=false&key=AIzaSyDjTT7ysuxenX8UauY3VUgqhjOxUBmsIKc
## .Information from URL : https://maps.googleapis.com/maps/api/geocode/json?address=
Bangalore&sensor=false&key=AIzaSyDjTT7ysuxenX8UauY3VUgqhjOxUBmsIKc
## .Information from URL : https://maps.googleapis.com/maps/api/geocode/json?address=
Jeddah,%20Kingdom%20of%20Saudi%20Arabia&sensor=false&key=AIzaSyDjTT7ysuxenX8UauY3VUgq
hjOxUBmsIKc
## Information from URL : https://maps.googleapis.com/maps/api/geocode/json?address=E
stherville,%20IA&sensor=false&key=AIzaSyDjTT7ysuxenX8UauY3VUgqhjOxUBmsIKc
## Information from URL : https://maps.googleapis.com/maps/api/geocode/json?address=O
porto,%20Portugal&sensor=false&key=AIzaSyDjTT7ysuxenX8UauY3VUgqhjOxUBmsIKc
## .Information from URL : https://maps.googleapis.com/maps/api/geocode/json?address=
Philadelphia&sensor=false&key=AIzaSyDjTT7ysuxenX8UauY3VUgqhjOxUBmsIKc
## Information from URL : https://maps.googleapis.com/maps/api/geocode/json?address=S
henzhen,China&sensor=false&key=AIzaSyDjTT7ysuxenX8UauY3VUgqhjOxUBmsIKc
## .Information from URL : https://maps.googleapis.com/maps/api/geocode/json?address=
United%20States&sensor=false&key=AIzaSyDjTT7ysuxenX8UauY3VUgqhjOxUBmsIKc
## Information from URL : https://maps.googleapis.com/maps/api/geocode/json?address=K
aiserslautern,%20Germany&sensor=false&key=AIzaSyDjTT7ysuxenX8UauY3VUgqhjOxUBmsIKc
## Information from URL : https://maps.googleapis.com/maps/api/geocode/json?address=M
ichigan,%20USA&sensor=false&key=AIzaSyDjTT7ysuxenX8UauY3VUgqhjOxUBmsIKc
## .Information from URL : https://maps.googleapis.com/maps/api/geocode/json?address=
DC%20Metro&sensor=false&key=AIzaSyDjTT7ysuxenX8UauY3VUgqhjOxUBmsIKc
## .Information from URL : https://maps.googleapis.com/maps/api/geocode/json?address=
Las%20Vegas,%20NV&sensor=false&key=AIzaSyDjTT7ysuxenX8UauY3VUgqhjOxUBmsIKc
## .Information from URL : https://maps.googleapis.com/maps/api/geocode/json?address=
Germany&sensor=false&key=AIzaSyDjTT7ysuxenX8UauY3VUgqhjOxUBmsIKc
## .Information from URL : https://maps.googleapis.com/maps/api/geocode/json?address=
NewEngland,Usa&sensor=false&key=AIzaSyDjTT7ysuxenX8UauY3VUgqhjOxUBmsIKc
```

```
## Information from URL : https://maps.googleapis.com/maps/api/geocode/json?address=C
olumbus,%20OH&sensor=false&key=AIzaSyDjTT7ysuxenX8UauY3VUgqhjOxUBmsIKc
## .Information from URL : https://maps.googleapis.com/maps/api/geocode/json?address=
S%C3%A9tif,%20Alg%C3%A9rie&sensor=false&key=AIzaSyDjTT7ysuxenX8UauY3VUgqhjOxUBmsIKc
## Information from URL : https://maps.googleapis.com/maps/api/geocode/json?address=Z
hejiang&sensor=false&key=AIzaSyDjTT7ysuxenX8UauY3VUgqhjOxUBmsIKc
## .Information from URL : https://maps.googleapis.com/maps/api/geocode/json?address=
Coventry,%20England&sensor=false&key=AIzaSyDjTT7ysuxenX8UauY3VUgqhjOxUBmsIKc
## Information from URL : https://maps.googleapis.com/maps/api/geocode/json?address=O
ntario&sensor=false&key=AIzaSyDjTT7ysuxenX8UauY3VUgqhjOxUBmsIKc
## Information from URL : https://maps.googleapis.com/maps/api/geocode/json?address=C
asablanca,%20Maroc&sensor=false&key=AIzaSyDjTT7ysuxenX8UauY3VUgqhjOxUBmsIKc
## Information from URL : https://maps.googleapis.com/maps/api/geocode/json?address=J
akarta,%20Indonesia&sensor=false&key=AIzaSyDjTT7ysuxenX8UauY3VUgqhjOxUBmsIKc
## Information from URL : https://maps.googleapis.com/maps/api/geocode/json?address=C
hicago%20&%20Global&sensor=false&key=AIzaSyDjTT7ysuxenX8UauY3VUgqhjOxUBmsIKc
## .Information from URL : https://maps.googleapis.com/maps/api/geocode/json?address=
Northeast,%20USA&sensor=false&key=AIzaSyDjTT7ysuxenX8UauY3VUgqhjOxUBmsIKc
## Information from URL : https://maps.googleapis.com/maps/api/geocode/json?address=C
oimbra,%20Portugal&sensor=false&key=AIzaSyDjTT7ysuxenX8UauY3VUgqhjOxUBmsIKc
## Information from URL : https://maps.googleapis.com/maps/api/geocode/json?address=B
erlin,%20Germany&sensor=false&key=AIzaSyDjTT7ysuxenX8UauY3VUgqhjOxUBmsIKc
## .Information from URL : https://maps.googleapis.com/maps/api/geocode/json?address=
Malta&sensor=false&key=AIzaSyDjTT7ysuxenX8UauY3VUgqhjOxUBmsIKc
## Information from URL : https://maps.googleapis.com/maps/api/geocode/json?address=K
okomo,%20IN&sensor=false&key=AIzaSyDjTT7ysuxenX8UauY3VUgqhjOxUBmsIKc
## .Information from URL : https://maps.googleapis.com/maps/api/geocode/json?address=
Paris,%20France&sensor=false&key=AIzaSyDjTT7ysuxenX8UauY3VUgqhjOxUBmsIKc
## .Information from URL : https://maps.googleapis.com/maps/api/geocode/json?address=
Stockholm,%20Sweden&sensor=false&key=AIzaSyDjTT7ysuxenX8UauY3VUgqhjOxUBmsIKc
## Information from URL : https://maps.googleapis.com/maps/api/geocode/json?address=L
os%20Angeles&sensor=false&key=AIzaSyDjTT7ysuxenX8UauY3VUgqhjOxUBmsIKc
## .Information from URL : https://maps.googleapis.com/maps/api/geocode/json?address=
New%20York%20&%20London&sensor=false&key=AIzaSyDjTT7ysuxenX8UauY3VUgqhjOxUBmsIKc
## Information from URL : https://maps.googleapis.com/maps/api/geocode/json?address=S
tockholm%20/%20San%20Francisco&sensor=false&key=AIzaSyDjTT7ysuxenX8UauY3VUgqhjOxUBmsI
Kc
## Information from URL : https://maps.googleapis.com/maps/api/geocode/json?address=m
israta,libya&sensor=false&key=AIzaSyDjTT7ysuxenX8UauY3VUgqhjOxUBmsIKc
## .Information from URL : https://maps.googleapis.com/maps/api/geocode/json?address=
Detroit,%20MI&sensor=false&key=AIzaSyDjTT7ysuxenX8UauY3VUgqhjOxUBmsIKc
## Information from URL : https://maps.googleapis.com/maps/api/geocode/json?address=L
ower%20Silesia,%20Poland&sensor=false&key=AIzaSyDjTT7ysuxenX8UauY3VUgqhjOxUBmsIKc
## .Information from URL : https://maps.googleapis.com/maps/api/geocode/json?address=
Tamil%20nadu%20India&sensor=false&key=AIzaSyDjTT7ysuxenX8UauY3VUgqhjOxUBmsIKc
## Information from URL : https://maps.googleapis.com/maps/api/geocode/json?address=N
ew%20York%20City&sensor=false&key=AIzaSyDjTT7ysuxenX8UauY3VUgqhjOxUBmsIKc
```

```
## Information from URL : https://maps.googleapis.com/maps/api/geocode/json?address=T
he%20Near%20Future&sensor=false&key=AIzaSyDjTT7ysuxenX8UauY3VUgqhjOxUBmsIKc
## Information from URL : https://maps.googleapis.com/maps/api/geocode/json?address=K
rakow,%20Poland&sensor=false&key=AIzaSyDjTT7ysuxenX8UauY3VUgqhjOxUBmsIKc
## .Information from URL : https://maps.googleapis.com/maps/api/geocode/json?address=
Santiago,%20Rep.%20Dominicana&sensor=false&key=AIzaSyDjTT7ysuxenX8UauY3VUgqhjOxUBmsIK
c
## Information from URL : https://maps.googleapis.com/maps/api/geocode/json?address=A
mpang,%20Malaysia&sensor=false&key=AIzaSyDjTT7ysuxenX8UauY3VUgqhjOxUBmsIKc
## Information from URL : https://maps.googleapis.com/maps/api/geocode/json?address=S
an%20Antonio,%20TX&sensor=false&key=AIzaSyDjTT7ysuxenX8UauY3VUgqhjOxUBmsIKc
## Information from URL : https://maps.googleapis.com/maps/api/geocode/json?address=U
nited%20Kingdom%20&sensor=false&key=AIzaSyDjTT7ysuxenX8UauY3VUgqhjOxUBmsIKc
## .Information from URL : https://maps.googleapis.com/maps/api/geocode/json?address=
M%C3%BCnchen&sensor=false&key=AIzaSyDjTT7ysuxenX8UauY3VUgqhjOxUBmsIKc
## Information from URL : https://maps.googleapis.com/maps/api/geocode/json?address=P
eople's%20Republic%20of%20China&sensor=false&key=AIzaSyDjTT7ysuxenX8UauY3VUgqhjOxUBms
IKc
## .Information from URL : https://maps.googleapis.com/maps/api/geocode/json?address=
North%20Carolina,%20USA&sensor=false&key=AIzaSyDjTT7ysuxenX8UauY3VUgqhjOxUBmsIKc
## Information from URL : https://maps.googleapis.com/maps/api/geocode/json?address=N
r%20Crewe,%20Cheshire&sensor=false&key=AIzaSyDjTT7ysuxenX8UauY3VUgqhjOxUBmsIKc
## Information from URL : https://maps.googleapis.com/maps/api/geocode/json?address=N
ew%20York,%20USA&sensor=false&key=AIzaSyDjTT7ysuxenX8UauY3VUgqhjOxUBmsIKc
## Information from URL : https://maps.googleapis.com/maps/api/geocode/json?address=K
ansas%20City%20Metro%20Area&sensor=false&key=AIzaSyDjTT7ysuxenX8UauY3VUgqhjOxUBmsIKc
## Information from URL : https://maps.googleapis.com/maps/api/geocode/json?address=R
atingen,%20Deutschland&sensor=false&key=AIzaSyDjTT7ysuxenX8UauY3VUgqhjOxUBmsIKc
## .Information from URL : https://maps.googleapis.com/maps/api/geocode/json?address=
Southeast%20Michigan&sensor=false&key=AIzaSyDjTT7ysuxenX8UauY3VUgqhjOxUBmsIKc
## Information from URL : https://maps.googleapis.com/maps/api/geocode/json?address=W
ashington,%20DC&sensor=false&key=AIzaSyDjTT7ysuxenX8UauY3VUgqhjOxUBmsIKc
## Information from URL : https://maps.googleapis.com/maps/api/geocode/json?address=L
as%20Vegas,%20NV&sensor=false&key=AIzaSyDjTT7ysuxenX8UauY3VUgqhjOxUBmsIKc
## .Information from URL : https://maps.googleapis.com/maps/api/geocode/json?address=
Global&sensor=false&key=AIzaSyDjTT7ysuxenX8UauY3VUgqhjOxUBmsIKc
## Information from URL : https://maps.googleapis.com/maps/api/geocode/json?address=C
hicago,%20IL&sensor=false&key=AIzaSyDjTT7ysuxenX8UauY3VUgqhjOxUBmsIKc
## .Information from URL : https://maps.googleapis.com/maps/api/geocode/json?address=
Hoboken,%20NJ&sensor=false&key=AIzaSyDjTT7ysuxenX8UauY3VUgqhjOxUBmsIKc
## Information from URL : https://maps.googleapis.com/maps/api/geocode/json?address=T
ianjin&sensor=false&key=AIzaSyDjTT7ysuxenX8UauY3VUgqhjOxUBmsIKc
## Information from URL : https://maps.googleapis.com/maps/api/geocode/json?address=U
K&sensor=false&key=AIzaSyDjTT7ysuxenX8UauY3VUgqhjOxUBmsIKc
## Information from URL : https://maps.googleapis.com/maps/api/geocode/json?address=M
exico%20D.F.%20&sensor=false&key=AIzaSyDjTT7ysuxenX8UauY3VUgqhjOxUBmsIKc
## .Information from URL : https://maps.googleapis.com/maps/api/geocode/json?address=
Birmingham,%20MI&sensor=false&key=AIzaSyDjTT7ysuxenX8UauY3VUgqhjOxUBmsIKc
## Information from URL : https://maps.googleapis.com/maps/api/geocode/json?address=D
etroit,%20MI&sensor=false&key=AIzaSyDjTT7ysuxenX8UauY3VUgqhjOxUBmsIKc
## Information from URL : https://maps.googleapis.com/maps/api/geocode/json?address=I
ndia&sensor=false&key=AIzaSyDjTT7ysuxenX8UauY3VUgqhjOxUBmsIKc
## Information from URL : https://maps.googleapis.com/maps/api/geocode/json?address=D
etroit,%20MI&sensor=false&key=AIzaSyDjTT7ysuxenX8UauY3VUgqhjOxUBmsIKc
## .Information from URL : https://maps.googleapis.com/maps/api/geocode/json?address=
Potsdam,%20Germany&sensor=false&key=AIzaSyDjTT7ysuxenX8UauY3VUgqhjOxUBmsIKc
## Information from URL : https://maps.googleapis.com/maps/api/geocode/json?address=A
ustralia&sensor=false&key=AIzaSyDjTT7ysuxenX8UauY3VUgqhjOxUBmsIKc
## .Information from URL : https://maps.googleapis.com/maps/api/geocode/json?address=
Pensacola,%20FL&sensor=false&key=AIzaSyDjTT7ysuxenX8UauY3VUgqhjOxUBmsIKc
## Information from URL : https://maps.googleapis.com/maps/api/geocode/json?address=G
urgaon,%20India&sensor=false&key=AIzaSyDjTT7ysuxenX8UauY3VUgqhjOxUBmsIKc
## Information from URL : https://maps.googleapis.com/maps/api/geocode/json?address=S
an%20Jose,%20CA&sensor=false&key=AIzaSyDjTT7ysuxenX8UauY3VUgqhjOxUBmsIKc
## Information from URL : https://maps.googleapis.com/maps/api/geocode/json?address=U
nited%20Kingdom&sensor=false&key=AIzaSyDjTT7ysuxenX8UauY3VUgqhjOxUBmsIKc
## .Information from URL : https://maps.googleapis.com/maps/api/geocode/json?address=
Mumbai,%20India&sensor=false&key=AIzaSyDjTT7ysuxenX8UauY3VUgqhjOxUBmsIKc
## Information from URL : https://maps.googleapis.com/maps/api/geocode/json?address=S
outh%20Africa&sensor=false&key=AIzaSyDjTT7ysuxenX8UauY3VUgqhjOxUBmsIKc
## Information from URL : https://maps.googleapis.com/maps/api/geocode/json?address=C
leveland%20&sensor=false&key=AIzaSyDjTT7ysuxenX8UauY3VUgqhjOxUBmsIKc
## Information from URL : https://maps.googleapis.com/maps/api/geocode/json?address=L
os%20Angeles,%20CA%20&sensor=false&key=AIzaSyDjTT7ysuxenX8UauY3VUgqhjOxUBmsIKc
## .Information from URL : https://maps.googleapis.com/maps/api/geocode/json?address=
United%20Kingdom&sensor=false&key=AIzaSyDjTT7ysuxenX8UauY3VUgqhjOxUBmsIKc
## Information from URL : https://maps.googleapis.com/maps/api/geocode/json?address=N
ova%20Scotia&sensor=false&key=AIzaSyDjTT7ysuxenX8UauY3VUgqhjOxUBmsIKc
## .Information from URL : https://maps.googleapis.com/maps/api/geocode/json?address=
Dayton,%20Ohio&sensor=false&key=AIzaSyDjTT7ysuxenX8UauY3VUgqhjOxUBmsIKc
## .Information from URL : https://maps.googleapis.com/maps/api/geocode/json?address=
Contagem&sensor=false&key=AIzaSyDjTT7ysuxenX8UauY3VUgqhjOxUBmsIKc
## Information from URL : https://maps.googleapis.com/maps/api/geocode/json?address=L
as%20Vegas,%20NV&sensor=false&key=AIzaSyDjTT7ysuxenX8UauY3VUgqhjOxUBmsIKc
## Information from URL : https://maps.googleapis.com/maps/api/geocode/json?address=C
hennai,%20India&sensor=false&key=AIzaSyDjTT7ysuxenX8UauY3VUgqhjOxUBmsIKc
## Information from URL : https://maps.googleapis.com/maps/api/geocode/json?address=H
otAtlanta.%20from%20Madrid,%20Spain&sensor=false&key=AIzaSyDjTT7ysuxenX8UauY3VUgqhjOx
UBmsIKc
## Information from URL : https://maps.googleapis.com/maps/api/geocode/json?address=%
D8%A7%D9%84%D9%85%D8%AF%D9%8A%D9%86%D8%A9%20%D8%A7%D9%84%D9%85%D9%86%D9%88%D8%B1%D8%A
9%20-%20%D8%A7%D9%84%D8%B1%D9%8A%D8%A7%D8%B6&sensor=false&key=AIzaSyDjTT7ysuxenX8UauY
3VUgqhjOxUBmsIKc
## Information from URL : https://maps.googleapis.com/maps/api/geocode/json?address=H
```

```
arrogate,%20UK&sensor=false&key=AIzaSyDjTT7ysuxenX8UauY3VUgqhjOxUBmsIKc
## .Information from URL : https://maps.googleapis.com/maps/api/geocode/json?address=
Sheffield,%20England&sensor=false&key=AIzaSyDjTT7ysuxenX8UauY3VUgqhjOxUBmsIKc
## .Information from URL : https://maps.googleapis.com/maps/api/geocode/json?address=
United%20Kingdom&sensor=false&key=AIzaSyDjTT7ysuxenX8UauY3VUgqhjOxUBmsIKc
## Information from URL : https://maps.googleapis.com/maps/api/geocode/json?address=L
oir-et-Cher,%20Centre&sensor=false&key=AIzaSyDjTT7ysuxenX8UauY3VUgqhjOxUBmsIKc
```

```
length(geocode_results)
```

```
## [1] 116
```

Clean the cordinate data with the following code,

```
condition_a <- sapply(geocode_results, function(x) x["status"]=="OK")
geocode_results<-geocode_results[condition_a]
condition_b <- lapply(geocode_results, lapply, length)
condition_b2<-sapply(condition_b, function(x) x["results"]=="1")
geocode_results<-geocode_results[condition_b2]
source("https://raw.githubusercontent.com/LucasPuente/geocoding/master/cleaning_geoco
ded_results.R")
results_b<-lapply(geocode_results, as.data.frame)
results_c<-lapply(results_b,function(x) subset(x, select=c("results.formatted_address
",
                                                              "results.geometry.location"))
)
results_d<-lapply(results_c,function(x) data.frame(Location=x[1,"results.formatted_ad
dress"],
                                                        lat=x[1,"results.geometry.location"
],
                                                      lng=x[2,"results.geometry.location"])
)
results_e<-rbindlist(results_d)
```

Now, with this dataframe of Twitter followers with the coordinates matching their self-reported location on Twitter bio. its time to plot their location on a map.

```
library(maps)
```

```
## Warning: package 'maps' was built under R version 3.4.4
```

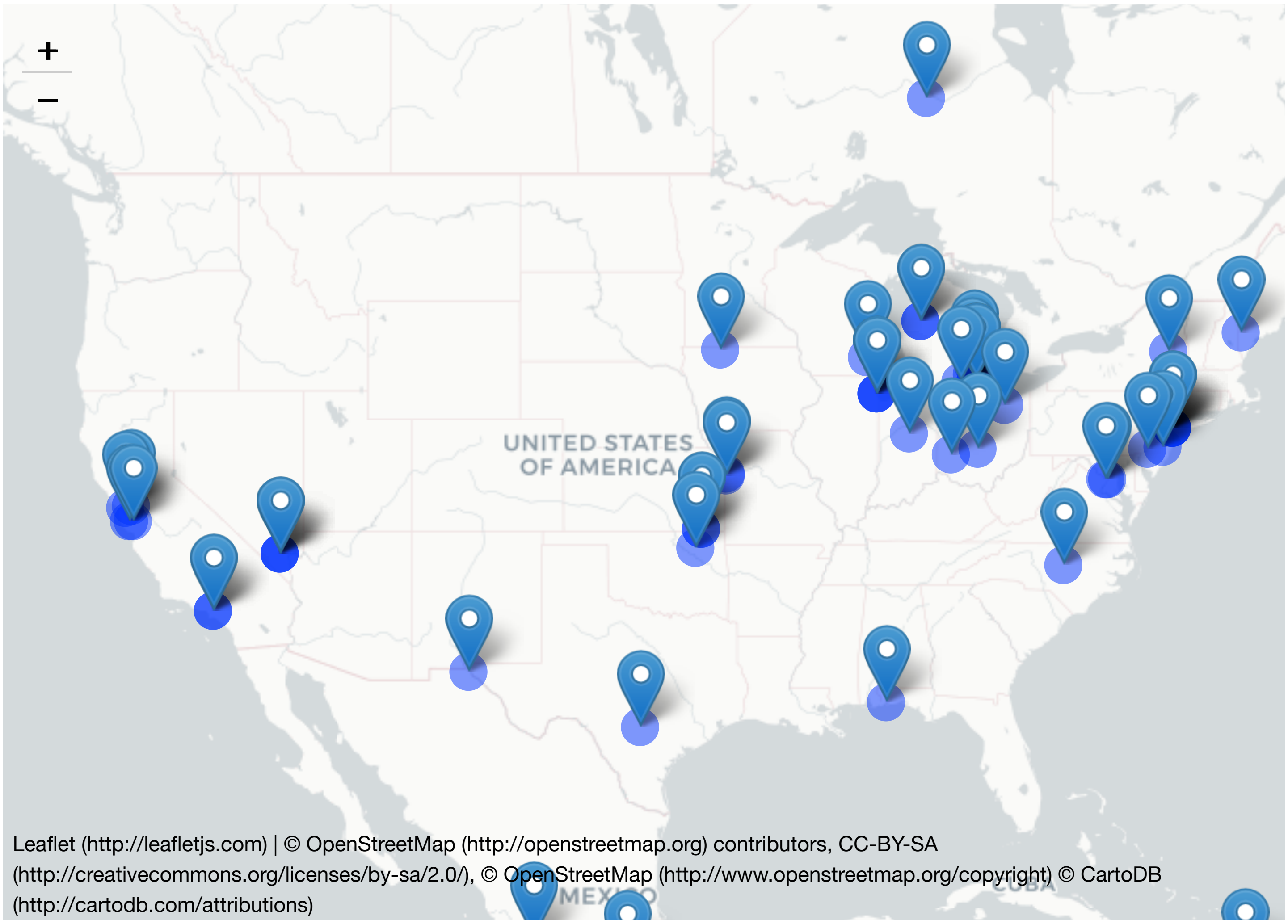
```
library(mapproj)
```

```
## Warning: package 'mapproj' was built under R version 3.4.4
```

```
map1 <- leaflet(data = results_e) %>%
  addTiles() %>%
  setView(lng = -98.35, lat = 39.50, zoom = 4) %>%
  addMarkers(lng = ~lng, lat = ~lat, popup = ~ as.character(Location)) %>%
  addProviderTiles("CartoDB.Positron") %>%
  addCircleMarkers(
    stroke = FALSE, fillOpacity = 0.5
  )
```

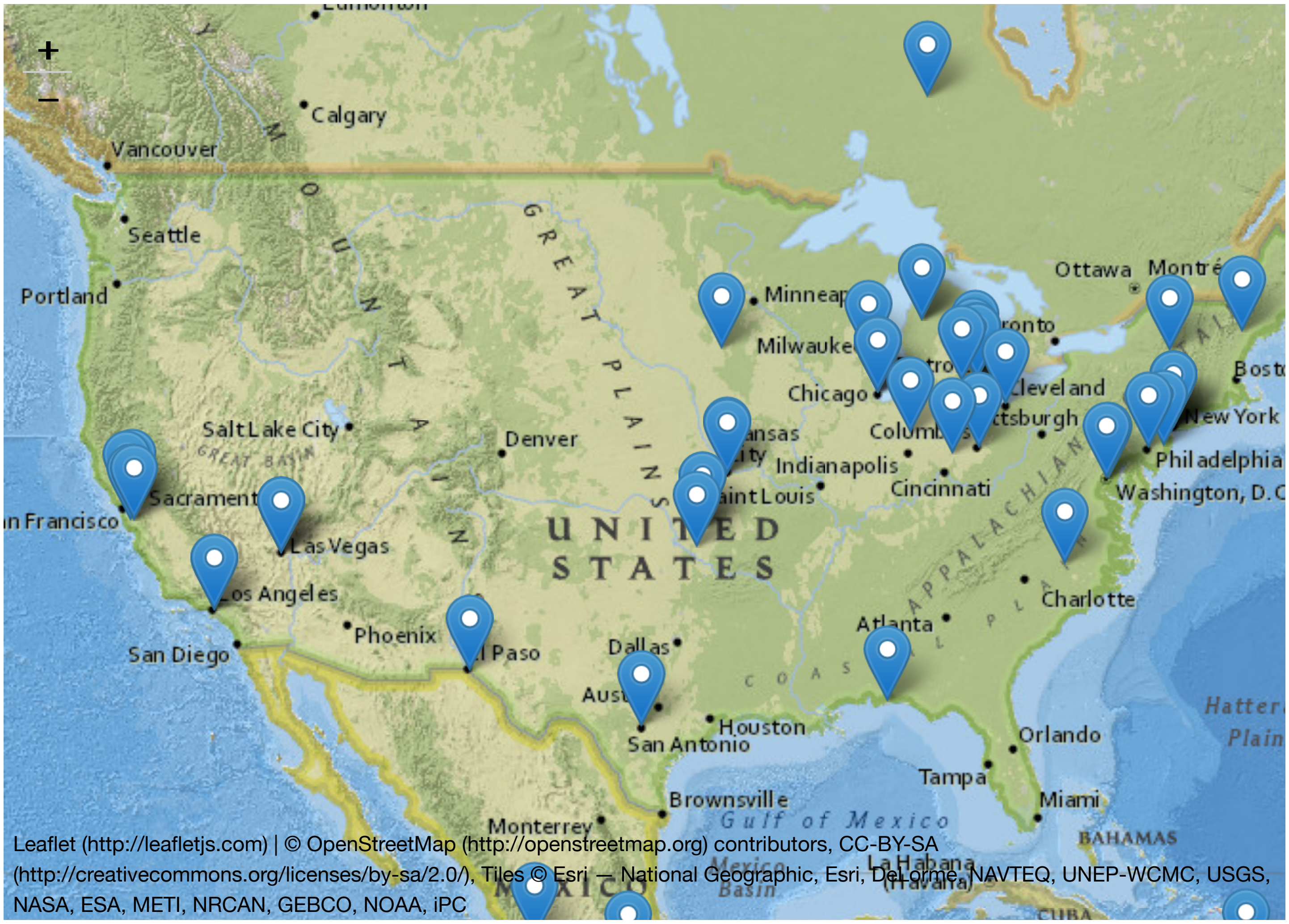
```
## Assuming "lng" and "lat" are longitude and latitude, respectively
```

```
map1
```



Leaflet (http://leafletjs.com) | © OpenStreetMap (http://openstreetmap.org) contributors, CC-BY-SA (http://creativecommons.org/licenses/by-sa/2.0/), © OpenStreetMap (http://www.openstreetmap.org/copyright) © CartoDB (http://cartodb.com/attributions)

```
usericon <- makeIcon(
  iconUrl = results_e$profile_image,
  iconWidth = 15, iconHeight = 15
)
map2 <- leaflet(data = results_e) %>%
  addTiles() %>%
  setView(lng = -98.35, lat = 39.50, zoom = 4) %>%
  addMarkers(lng = ~lng, lat = ~lat, popup = ~ as.character(Location),icon = usericon
,data = results_e) %>%
  addProviderTiles(providers$Esri.NatGeoWorldMap)
map2
```



1)Text Mining (extract, mine, clean & stem) The first step was to setup an API on twitter to extract the tweets directly in R.Using “twitteR”" package the api is authenticated for us to start loading tweets into R.

After authenticating it was time to bring in the tweets related to “Delphi Technologies”using “#delphitech”"

Process 1. Extract tweets and followers from the Twitter website with R and the twitteR package

2. With the tm package, clean text by removing punctuations, numbers, hyperlinks and stop words, followed by stemming and stem completion
3. Build a term-document matrix

```
##RETRIEVE TWEETS

library(twitteR)
library(ROAuth)

setup_twitter_oauth("jmbwTtQL7ZoW4rxCMOmUOOmXJ", "3X5v1OUEs67lw25vg6rK7muS6VmKFTAQGMp
kVOVF2JQgHzZikI", "51183723-uKDVYied6jRISD0aJLstyJkZLW8Ykqttqos8mbMJI", "tkmsZd790aSy
6rXgDlbwfUWLtHC6Wjxm68pQE1uumO21I")
```

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

Text Cleaning: To clean the tweets data “tm” package was used.The main structure for managing documents in tm is a so-called Corpus, representing a collection of text documents. Therefore below code served us as a pre-cleaning in order to prepare clean text mining data. Each of the tweets will be considered a separate document. Useful source: https://cran.r-project.org/web/packages/tm/vignettes/tm.pdf (https://cran.r-project.org/web/packages/tm/vignettes/tm.pdf)

```
## TEXT CLEANING
library(tm)
```

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

```
DelTech <- userTimeline("delphitech", n = 3200)
head(DelTech)
```

```
## [[1]]
## [1] "delphitech: We are proud to again partner with @Rebuild2gthrOKC to restore a
home in need. https://t.co/cSneU2tZrJ"
##
## [[2]]
## [1] "delphitech: We are starting the first quarter strong, raising our full-year o
utlook as growth momentum in key technologies cont… https://t.co/gvGKlgssXK"
##
## [[3]]
## [1] "delphitech: Start your day with a steering &amp; suspension hot topics podcas
t hosted by ASE certified master technician Dave Hobbs… https://t.co/YOvyZperZm"
##
## [[4]]
## [1] "delphitech: Going to the Vienna Symposium? If so, check us out and come get y
our free global emissions guide book!… https://t.co/Mxnr4wjuT7"
##
## [[5]]
## [1] "delphitech: Good luck to all of the talented kids competing in the #FIRSTCham
p in Detroit. We are rooting for you! https://t.co/wm1GkQzhpG"
##
## [[6]]
## [1] "delphitech: Because we know every gallon of fuel saved counts. Driven to make
a difference.  #EarthDay2018 https://t.co/5rWqbunaWF"
```

```
DelTech_df <- twListToDF(DelTech)
head(DelTech_df)
```

	▶
1	
2	
3	
4	
5	
6	

6 rows | 1-1 of 17 columns

```
DelTech_df[190, c("id", "created", "screenName", "replyToSN",
  "favoriteCount", "retweetCount", "longitude", "latitude", "text")]
```

	id	created	screenNa...	replyToSN	favoriteCount	ret
	<chr>	<S3: POSIXct>	<chr>	<chr>	<dbl>	
190	848981933673504768	2017-04-03 19:34:00	delphitech	NA	3	

1 row | 1-7 of 10 columns

```
myCorpus <- Corpus(VectorSource(DelTech_df$text))
# 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 anything other than English letters or space removeNumPunct <- function(x) g
sub("[^[:alpha:][:space:]]*", "", x) myCorpus <- tm_map(myCorpus, content_transformer
(removeNumPunct))
# remove stopwords
myStopwords <- c(setdiff(stopwords('english'), c("r", "big")),
                     "use", "see", "used", "via", "amp")
    myCorpus <- tm_map(myCorpus, removeWords, myStopwords)
    # remove extra whitespace
    myCorpus <- tm_map(myCorpus, stripWhitespace)
    # keep a copy for stem completion later
    myCorpusCopy <- myCorpus
```

Term Document Matrix A common approach in text mining is to create a term-document matrix from a corpus. In the tm package the classes TermDocumentMatrix and DocumentTermMatrix (depending on whether you want terms as rows and documents as columns, or vice versa) employ sparse matrices for corpora. Inspecting a term-document matrix displays a sample, whereas as.matrix() yields the full matrix in dense format.

```
##BUILD TERM DOCUMENT MATRIX
(tdm <- TermDocumentMatrix(myCorpus, control = list(wordLengths = c(1, Inf))))
```

```
## <<TermDocumentMatrix (terms: 1389, documents: 369)>>
## Non-/sparse entries: 3781/508760
## Sparsity           : 99%
## Maximal term length: 20
## Weighting          : term frequency (tf)
```

```
tdm
```

```
## <<TermDocumentMatrix (terms: 1389, documents: 369)>>
## Non-/sparse entries: 3781/508760
## Sparsity           : 99%
## Maximal term length: 20
## Weighting          : term frequency (tf)
```

```
idx <- which(dimnames(tdm)$Terms %in% c("clean", "after market", "fuel"))
as.matrix(tdm[idx, 21:30])
```

```
##        Docs
## Terms   21 22 23 24 25 26 27 28 29 30
##   fuel   0  0  0  0  0  1  0  0  0  0
##   clean  0  0  0  0  0  0  0  0  0  0
```

The top frequent terms used for delphitech tweets

```
## TOP FREQUENT TERMS.
(freq.terms <- findFreqTerms(tdm, lowfreq =20))
```

```
##  [1] "check"           "us"              "…"
##  [4] "fuel"            "north"           "can"
##  [7] "delphi"          "injector"        "new"
## [10] "learn"           "visit"           "performance"
## [13] "pump"            "delphiautoparts" "sundayfunday"
```

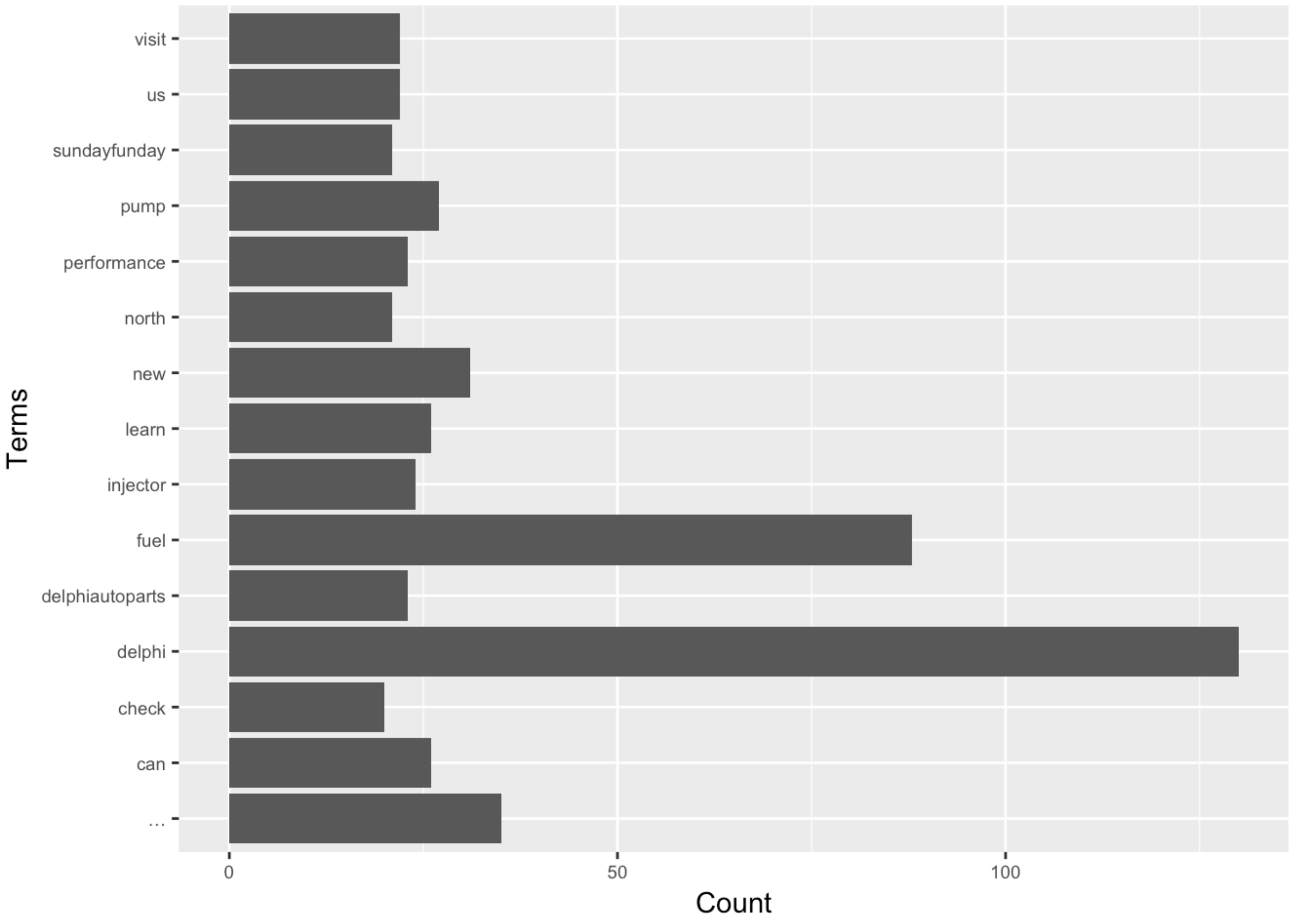
```
term.freq <- rowSums(as.matrix(tdm))
term.freq <- subset(term.freq, term.freq >= 20)
df <- data.frame (term = names(term.freq), freq = term.freq)

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))
```



```
##below you can see how the occurances behave, Delphi & fuel show up in a lot of twee
ts and it makes sense as Fuel products is one of the biggest drivers for our company.
```

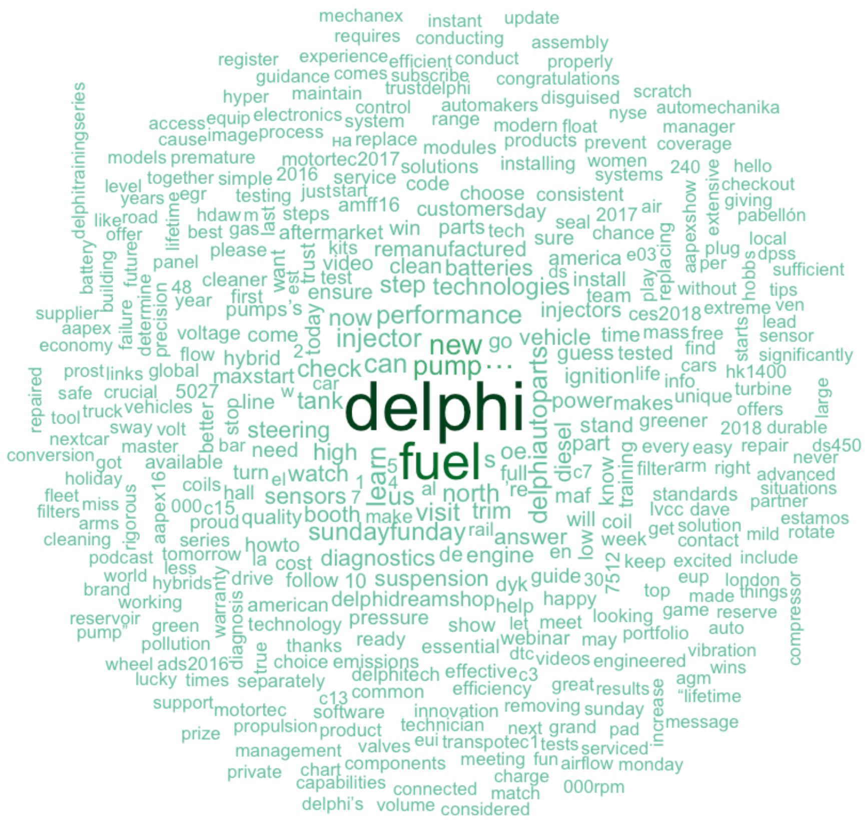
Wordcloud I used wordcloud to present the words in Delphi Technology tweets in which size of each word indicates its frequency or importance.

First step is to create matrix of a TermDocumentMatrix; Second Step is to calculate frequency of a given word and sort it descending; The last step is to create wordcloud with the most frequent word in the center.

```
##WORDCLOUD
library(wordcloud)
```

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

```
mat <- as.matrix(tdm)
# Frequency#
word.freq <- sort(rowSums(mat), decreasing = T)
# Colors#
pal <- brewer.pal(9, "BuGn")[-(1:4)]
# Generate wordcloud#
wordcloud(word = names(word.freq), freq = word.freq, min.freq = 3, random.order = F,
    colors = pal, scale = c(2, 0.5))
```



Analyses Associations For any given word, findAssocs() calculates its correlation with every other word in a TDM or DTM. Scores range from 0 to 1. A score of 1 means that two words always appear together, while a score of 0 means that they never appear together.

```
##Find Associations.
findAssocs(tdm, "fuel", 0.2)
```

```
## $fuel
##      large  reservoir situations sufficient     volume   maintain
##       0.49       0.49       0.49       0.49       0.49       0.46
##       pump       tank     vehic…    modules        low      howto
##       0.43       0.40       0.40       0.36       0.34       0.34
##  replacing   requires      video  cornering      clean   removing
##       0.32       0.31       0.29       0.28       0.27       0.26
##    process       step      pumps     simple        dyk
##       0.26       0.23       0.23       0.22       0.22
```

```
findAssocs(tdm, "delphi", 0.2)
```

```
## $delphi
##          com       device      suspect      warwick           ds
##         0.31         0.27         0.27         0.27         0.24
##  counterfeit        trust technologies  aftermarket
##         0.24         0.21         0.20         0.20
```

Network of terms We often want to know connection between words just like between humans. With network analysis, not only can we determine which terms appear together frequently, we can visualize how keywords and tweets are connected as a network of terms. This way, we can resolve the number of connections keywords have with one another, and how many connections a specific keyword has with other keywords. We have chosen to show the network of the 15 most frequent terms.

```
##NETWORK OF TERMS.
library(graph)
```

```
## Loading required package: BiocGenerics
```

```
## Loading required package: parallel
```

```
##
## Attaching package: 'BiocGenerics'
```

```
## The following objects are masked from 'package:parallel':
##
##     clusterApply, clusterApplyLB, clusterCall, clusterEvalQ,
##     clusterExport, clusterMap, parApply, parCapply, parLapply,
##     parLapplyLB, parRapply, parSapply, parSapplyLB
## The following object is masked from 'package:twitteR':
##
##     as.data.frame
```

```
## The following objects are masked from 'package:stats':
##
##     IQR, mad, sd, var, xtabs
```

```
## The following objects are masked from 'package:base':
##
##     anyDuplicated, append, as.data.frame, cbind, colMeans,
##     colnames, colSums, do.call, duplicated, eval, evalq, Filter,
##     Find, get, grep, grepl, intersect, is.unsorted, lapply,
##     lengths, Map, mapply, match, mget, order, paste, pmax,
##     pmax.int, pmin, pmin.int, Position, rank, rbind, Reduce,
##     rowMeans, rownames, rowSums, sapply, setdiff, sort, table,
##     tapply, union, unique, unsplit, which, which.max, which.min
```

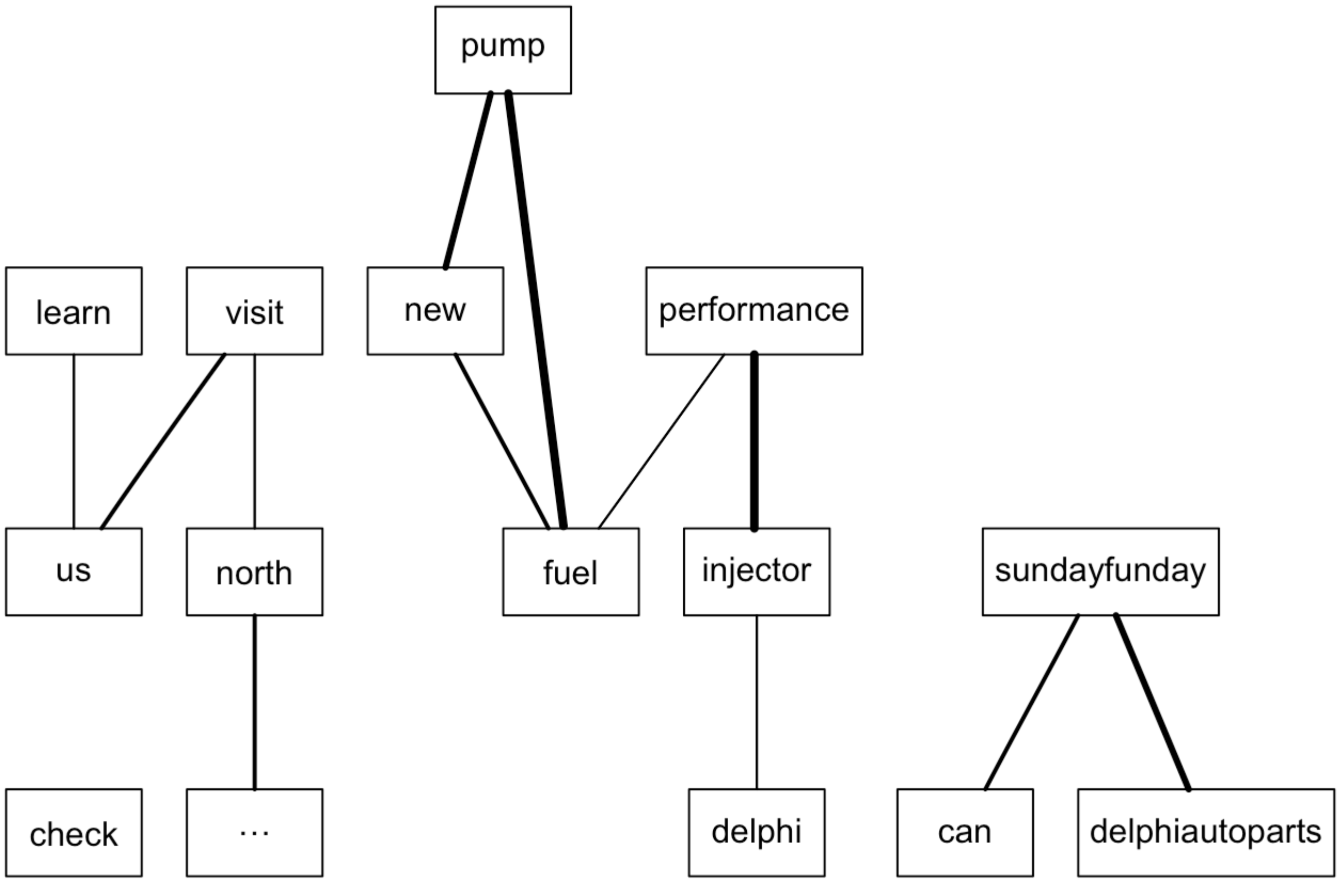
```
library(Rgraphviz)
```

```
## Loading required package: grid
```

```
##
## Attaching package: 'Rgraphviz'
```

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

```
plot(tdm, term = freq.terms, corThreshold = 0.1, weighting = T)
```



As you can see the beiggest associations are with fuel & pump along with injector and performance.

Sentiment Analysis The term sentiment is for a way to judge how our brand and company is perceived by people following us.It helps us to understand the positive,negative & neutral preference to our brand and can be a key tool for markting to convert the negative and neutral followers to positive followers which increases brand awerness.

```
## SENTIMENT ANALYSIS
library(sentiment)
```

```
## Loading required package: RCurl
```

```
## Loading required package: bitops
```

```
## Loading required package: rjson
```

```
##
## Attaching package: 'rjson'
```

```
## The following objects are masked from 'package:RJSONIO':
##
##     fromJSON, toJSON
```

```
## Loading required package: plyr
```

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

```
## The following object is masked from 'package:graph':
##
##     join
```

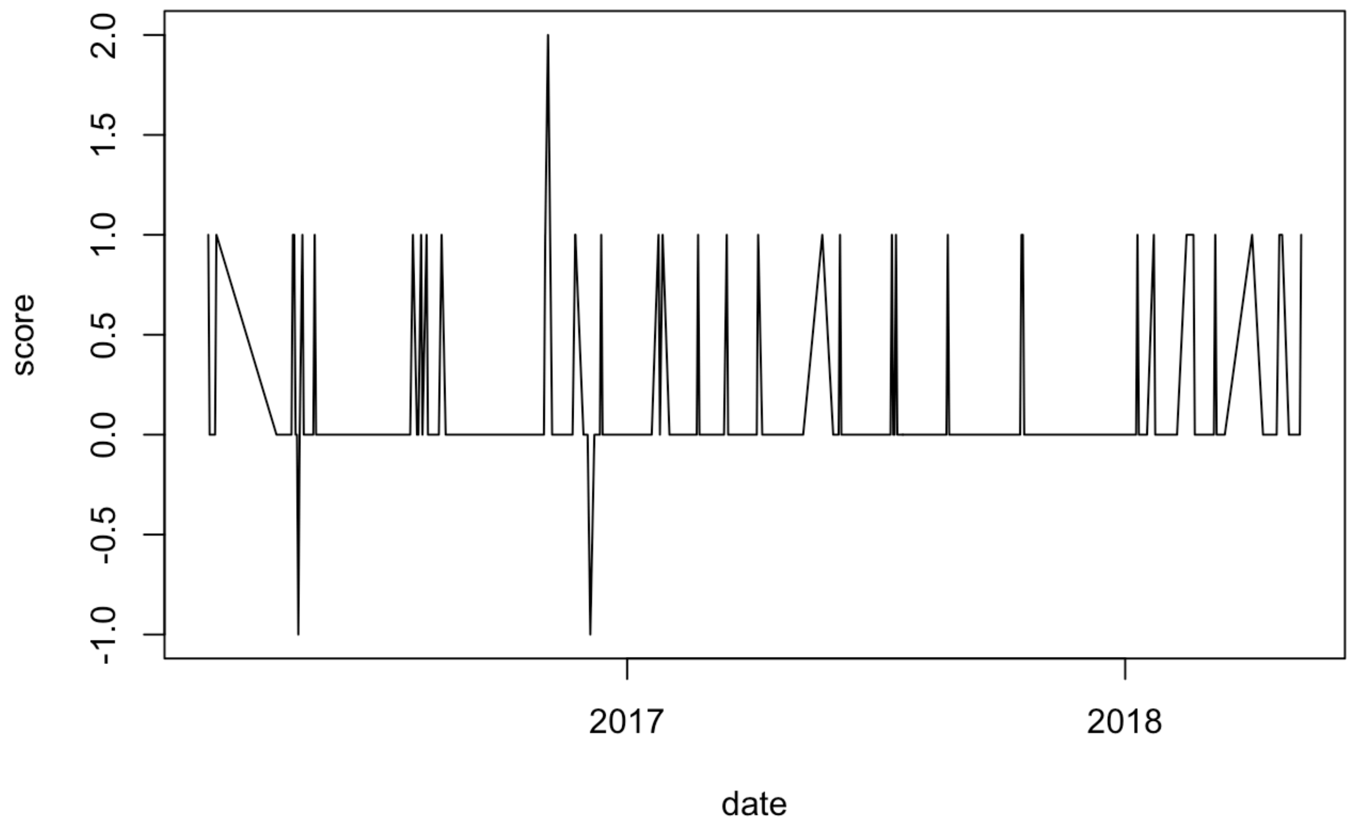
```
## The following object is masked from 'package:maps':
##
##     ozone
```

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

```
sentiments <- sentiment(DelTech_df$text)
table(sentiments$polarity)
```

```
##
## negative  neutral positive
##        2      331       36
```

```
sentiments$score <- 0
sentiments$score[sentiments$polarity == "positive"] <- 1
sentiments$score[sentiments$polarity == "negative"] <- -1
sentiments$date <- as.Date(DelTech_df$created)
result <- aggregate(score ~ date, data = sentiments, sum)
plot(result, type = "l")
```



Looking at the sentiment analysis certain actions are required to convert the negative & neutral followers to positive followers.

Followers analysis: Retrieve User Info Used the twitteR package to retrieve user info. The appopriate code is shown above. only friends and followers were retrieved and There is an option to retireve followers of our followers, but this is very time consuming.

```
##FOLLOWER ANALYSIS
user <- getUser("Delphitech")
user$toDataFrame()
```

description <chr>
We are a leading global automotive emissions, fuel economy and aftermarket solutions provider ready to cre Get there with us.

1 row | 1-1 of 17 columns

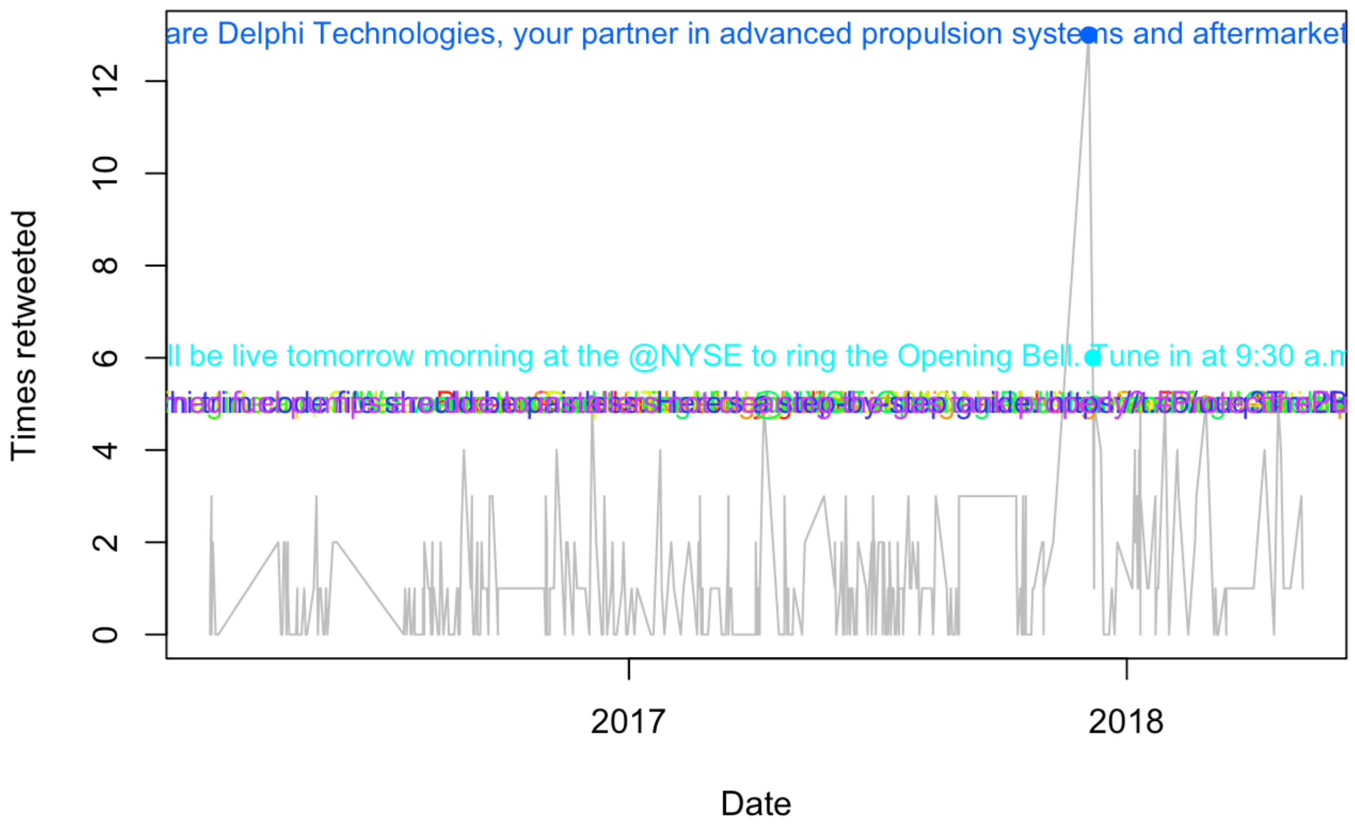
```
friends <- user$getFriends() # who this user follows
followers <- user$getFollowers() # this user's followers
```

Top Retweeted Tweets Plotted the top retweeted tweets, the limit at 5, to understand what the the most retweeted tweet.

```
## RETWEETS
table(DelTech_df$retweetCount)
```

```
##
##   0   1   2   3   4   5   6  13
## 155 118  49  29   9   7   1   1
```

```
selected <- which(DelTech_df$retweetCount >= 5)
# plot them
dates <- strptime(DelTech_df$created, format="%Y-%m-%d")
plot(x=dates, y=DelTech_df$retweetCount, type="l", col="grey",
     xlab="Date", ylab="Times retweeted")
colors <- rainbow(10)[1:length(selected)]
points(dates[selected], DelTech_df$retweetCount[selected],
       pch=19, col=colors)
text(dates[selected], DelTech_df$retweetCount[selected],
     DelTech_df$text[selected], col=colors, cex=.9)
```



Conclusion: Based on the project above I can say I learnt a lot and used the material thought in class to answer a business question. Social media is now part our daily lives and the data from it is valuable and can highlight strenghts and weakness of a person/brand and with tools and projects like these certain value added insights can be gleamed which can help teams to convert the weakenss into strenghts while keeping the strenghts intact.