SentiMap

HU WEI

1 March 2018

Import Data

First of all, we can import the data.

Remove all the variables in the environment

```
rm(list=ls())
```

import csv file

```
t2<-read.csv("twitter_file_with_text.csv",fill=T, sep=",", stringsAsFactors = FALSE)
class(t2)
```

```
## [1] "data.frame"
```

```
names(t2)
```

```
##  [1] "follow_request_sent"
##  [2] "contributors"
##  [3] "truncated"
##  [4] "profile_use_background_image"
##  [5] "profile_sidebar_fill_color"
##  [6] "time_zone"
##  [7] "in_reply_to_status_id"
##  [8] "id"
##  [9] "favorite_count"
## [10] "verified"
## [11] "sentiment"
## [12] "profile_text_color"
## [13] "profile_image_url_https"
## [14] "retweeted"
## [15] "is_translator"
## [16] "source"
## [17] "followers_count"
## [18] "protected"
## [19] "in_reply_to_screen_name"
## [20] "in_reply_to_user_id"
## [21] "default_profile_image"
## [22] "retweet_count"
## [23] "id_str"
## [24] "favorited"
## [25] "utc_offset"
## [26] "statuses_count"
## [27] "profile_background_color"
## [28] "friends_count"
## [29] "profile_background_image_url_https"
## [30] "profile_link_color"
## [31] "profile_image_url"
## [32] "notifications"
## [33] "geo_enabled"
```

```
## [34] "profile_banner_url"
## [35] "in_reply_to_user_id_str"
## [36] "profile_background_image_url"
## [37] "lang"
## [38] "profile_background_tile"
## [39] "favourites_count"
## [40] "screen_name"
## [41] "url"
## [42] "created_at"
## [43] "contributors_enabled"
## [44] "location"
## [45] "filter_level"
## [46] "in_reply_to_status_id_str"
## [47] "profile_sidebar_border_color"
## [48] "place"
## [49] "default_profile"
## [50] "following"
## [51] "listed_count"
```

Attach

```
attach(t2)
Size<-dim(t2)
```

Descriptive statistics

Get the class of variables in dataset.

```
lapply(t2, class)
```

```
## $follow_request_sent
## [1] "logical"
##
## $contributors
## [1] "logical"
##
## $truncated
## [1] "character"
##
## $profile_use_background_image
## [1] "character"
##
## $profile_sidebar_fill_color
## [1] "character"
##
## $time_zone
## [1] "character"
##
## $in_reply_to_status_id
## [1] "numeric"
##
## $id
## [1] "integer"
##
## $favorite_count
```

```
## [1] "integer"
##
## $verified
## [1] "character"
##
## $sentiment
## [1] "integer"
##
## $profile_text_color
## [1] "character"
##
## $profile_image_url_https
## [1] "character"
##
## $retweeted
## [1] "character"
##
## $is_translator
## [1] "character"
##
## $source
## [1] "character"
##
## $followers_count
## [1] "integer"
##
## $protected
## [1] "character"
##
## $in_reply_to_screen_name
## [1] "character"
##
## $in_reply_to_user_id
## [1] "integer"
##
## $default_profile_image
## [1] "character"
##
## $retweet_count
## [1] "integer"
##
## $id_str
## [1] "integer"
##
## $favorited
## [1] "character"
##
## $utc_offset
## [1] "integer"
##
## $statuses_count
## [1] "integer"
##
## $profile_background_color
```

```
## [1] "character"
##
## $friends_count
## [1] "integer"
##
## $profile_background_image_url_https
## [1] "character"
##
## $profile_link_color
## [1] "character"
##
## $profile_image_url
## [1] "character"
##
## $notifications
## [1] "logical"
##
## $geo_enabled
## [1] "character"
##
## $profile_banner_url
## [1] "character"
##
## $in_reply_to_user_id_str
## [1] "integer"
##
## $profile_background_image_url
## [1] "character"
##
## $lang
## [1] "character"
##
## $profile_background_tile
## [1] "character"
##
## $favourites_count
## [1] "integer"
##
## $screen_name
## [1] "character"
##
## $url
## [1] "character"
##
## $created_at
## [1] "character"
##
## $contributors_enabled
## [1] "character"
##
## $location
## [1] "character"
##
## $filter_level
```

```
## [1] "character"
##
## $in_reply_to_status_id_str
## [1] "numeric"
##
## $profile_sidebar_border_color
## [1] "character"
##
## $place
## [1] "character"
##
## $default_profile
## [1] "character"
##
## $following
## [1] "logical"
##
## $listed_count
## [1] "integer"
```

```
summary(sentiment)
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## -12.00000   0.00000   0.00000   0.09434   0.00000   9.00000
```

```
table(sentiment)
```

```
## sentiment
##  -12  -10   -8   -7   -6   -5   -4   -3   -2   -1    0    1    2    3    4
##    1    1    1    3    3    8   21   37   71  101 1950   65   95   78   26
##    5    6    7    8    9
##   10   10    4    3    3
```

```
#==============
# Followers count
#==============
summary(followers_count)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       0     104     252    2512     610 1379617
```

```
summary(statuses_count)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       0    1239    4387   11729   13182  295091
```

```
summary(friends_count)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
##      0.0    120.0    258.0    921.0    557.5 354695.0
```

```
summary(favourites_count)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     0.0    10.0    83.0   831.8   445.0 94135.0
```

```
summary(listed_count)
```

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

```
##     0.00     0.00     0.00    17.72     1.00 12319.00
```

Percentage of followers number that exceeds 1000

```
sum(followers_count>1000)/Size[1]
```

```
## [1] 0.1633882
```

Percentage of followers number that exceeds 5000

```
sum(followers_count>5000)/Size[1]
```

```
## [1] 0.0337214
```

```
sum(lang=="en")/Size[1]
```

```
## [1] 0.505821
```

```
sum(geo_enabled == "True")/Size[1]
```

```
## [1] 0.3753513
```

```
sum(location[lang=="en"]=="")/Size[1]
```

```
## [1] 0.190285
```

Time Zone

Get geographical data

```
# Check version of R, becasue ggmap require R version higher than 3.4.3
#R.Version()
#install.packages("ggmap")
library(ggmap)
```

```
## Loading required package: ggplot2
```

```
#install.packages("tidyverse")
#library(tidyverse)
# Check the version info of ggmap
#sessionInfo()
```

Set the number of points that we want to get geoinfo.

```
Num = 10
```

```
# Initialize the data frame
lon <- vector(mode="numeric", length=Num)
lat <- vector(mode="numeric", length=Num)
geoAddress <- vector(mode="character", length=Num)
# Loop through the addresses to get the latitude and longitude of each address and add it to the
# origAddress data frame in new columns lat and lon
#for(i in 1:Num)
for(i in 1:Num)
{
  result <- tryCatch(geocode(time_zone[i], output = "latlona", source = "google"),
                     warning = function(w) data.frame(lon = NA, lat = NA, address = NA))
  lon[i] <- as.numeric(result[1])
  lat[i] <- as.numeric(result[2])
```

```
  geoAddress[i] <- as.character(result[3])
}
```

```
## Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=Monterrey&sensor=fals
## Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=Paris&sensor=false
## Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=London&sensor=false
## Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=Athens&sensor=false
## Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=Amsterdam&sensor=fals
## Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=Baghdad&sensor=false
## Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=Pacific%20Time%20(US
```

```
geocoded <- data.frame(lon, lat, geoAddress)
```

Save geographical data

```
# Write a CSV file containing origAddress to the working directory
write.csv(geocoded, "geocoded.csv", row.names=FALSE)
```

Plot Map

```
#install.packages("rworldmap")
library(rworldmap)
```

```
## Loading required package: sp
## ### Welcome to rworldmap ###
## For a short introduction type :   vignette('rworldmap')
```

```
newmap <- getMap(resolution = "low")
plot(newmap, xlim = c(-20, 59), ylim = c(35, 71), asp = 1)
points(geocoded$lon, geocoded$lat, col = "red", cex = .6)
```