

# asishkaklab1Part3

March 8, 2019

## 0.1 Search API

```
In [61]: install.packages("twitteR")
library("twitteR")

# Keys
api_key <- "XKCvkSC3TQDBJLLfaMJGbyuyT"
api_secret <- "SPkJ1siXeBh13GmwcrCYispF8eriDx7LQoIdMYfADodsG9neUe"
token <- "4808854766-2b02ddivk2dSwqgJgtg5JLDCN741tCN0Q26ax3a"
token_secret <- "laVPlvPitjdmlhXi0KorC9ODYa83XhAq98LyUAEC6EHkr"

# Connecting to Twitter API
setup_twitter_oauth(api_key,api_secret,token,token_secret)

# Collect tweets
tweets <- searchTwitter("flu OR influenza OR cold OR fever OR #flu OR #cold OR #influenza")

Updating HTML index of packages in '.Library'
Making 'packages.html' ... done

[1] "Using direct authentication"

In [62]: # Saving tweets to a dataframe
#old_tweets <- readRDS("tweets.Rda")
new_tweets <- twListToDF(tweets)

In [63]: nrow(old_tweets)
nrow(new_tweets)

# Saving to a .csv File.
#write.table (tweets,"tweets.csv",append=TRUE,row.names = FALSE,col.names=TRUE,sep=",",as.csv = TRUE)

8400
200

In [22]: tweets <- rbind(old_tweets, new_tweets)
saveRDS(tweets, file = "tweets.Rda")
```

```

In [ ]: tweets <- readRDS("tweets.Rda")

In [64]: # Reading .csv File
# tweets <- read.csv("tweets.csv",header=TRUE, sep=",", encoding = "UCS-2LE")
tweets <- new_tweets

In [65]: nrow(tweets)

200

In [7]: doInstall <- TRUE
toInstall <- c("twitter", "dismo", "maps", "ggplot2", "ggmap")
if(doInstall){install.packages(toInstall, repos = "http://cran.us.r-project.org")}
lapply(toInstall, library, character.only = TRUE)

```

Updating HTML index of packages in '.Library'

Making 'packages.html' ... done

Google's Terms of Service: <https://cloud.google.com/maps-platform/terms/>.

Please cite ggmap if you use it! See citation("ggmap") for details.

Attaching package: ggmap

The following object is masked from package:dismo:

geocode

1. (a) 'ggplot2' (b) 'maps' (c) 'dismo' (d) 'raster' (e) 'sp' (f) 'twitter' (g) 'jsonlite' (h) 'formatR' (i) 'stats' (j) 'graphics' (k) 'grDevices' (l) 'utils' (m) 'datasets' (n) 'methods' (o) 'base'
2. (a) 'ggplot2' (b) 'maps' (c) 'dismo' (d) 'raster' (e) 'sp' (f) 'twitter' (g) 'jsonlite' (h) 'formatR' (i) 'stats' (j) 'graphics' (k) 'grDevices' (l) 'utils' (m) 'datasets' (n) 'methods' (o) 'base'
3. (a) 'ggplot2' (b) 'maps' (c) 'dismo' (d) 'raster' (e) 'sp' (f) 'twitter' (g) 'jsonlite' (h) 'formatR' (i) 'stats' (j) 'graphics' (k) 'grDevices' (l) 'utils' (m) 'datasets' (n) 'methods' (o) 'base'
4. (a) 'ggplot2' (b) 'maps' (c) 'dismo' (d) 'raster' (e) 'sp' (f) 'twitter' (g) 'jsonlite' (h) 'formatR' (i) 'stats' (j) 'graphics' (k) 'grDevices' (l) 'utils' (m) 'datasets' (n) 'methods' (o) 'base'
5. (a) 'ggmap' (b) 'ggplot2' (c) 'maps' (d) 'dismo' (e) 'raster' (f) 'sp' (g) 'twitter' (h) 'jsonlite' (i) 'formatR' (j) 'stats' (k) 'graphics' (l) 'grDevices' (m) 'utils' (n) 'datasets' (o) 'methods' (p) 'base'

## 0.2 Looking up for User Info

```

In [66]: userInfo <- lookupUsers(tweets$screenName)
userDetails <- twListToDF(userInfo)

# users with locations
userLocationCheck <- !is.na(userDetails$location)
userLocation <- userDetails$location[userLocationCheck]

```

### 0.3 Getting geocode of tweets

```
In [ ]: #install.packages("xml2")
        #install.packages("XML", repos = "http://www.omegahat.org/R")
        #devtools::install_github("r-lib/xml2")
        library(plyr)
        install.packages('XML',repos='http://cran.us.r-project.org')
        library(XML)

In [67]: install.packages("ggmap")
        library(ggmap)
        library(dismo)
        register_google(key = "AIzaSyBMEQodtdMh7x22Z9QwzUyIZnN-Sm541do")
        code <- ggmap::geocode(userDetails$location[userLocationCheck],output = "more")
```

Updating HTML index of packages in '.Library'

Making 'packages.html' ... done

Source : <https://maps.googleapis.com/maps/api/geocode/json?address=Los+Angeles,+CA&key=xxx-Sm541do>

Source : <https://maps.googleapis.com/maps/api/geocode/json?address=Somewhere+near+chicago&key=xxx-Sm541do>

Source : <https://maps.googleapis.com/maps/api/geocode/json?address=mindin+my+bizznnaae&key=xxx-Sm541do>

Warning message:

Geocoding "mindin my bizznna..." failed with error:

Source : <https://maps.googleapis.com/maps/api/geocode/json?address=Dirty+Myrtle&key=xxx-Sm541do>

Source : <https://maps.googleapis.com/maps/api/geocode/json?address=Texas,+USA&key=xxx-Sm541do>

Source : <https://maps.googleapis.com/maps/api/geocode/json?address=California,+USA&key=xxx-Sm541do>

Source : <https://maps.googleapis.com/maps/api/geocode/json?address=University+of+Oregon&key=xxx-Sm541do>

"University of Oregon" not uniquely geocoded, using "1585 e 13th ave, eugene, or 97403, usa"

Source : <https://maps.googleapis.com/maps/api/geocode/json?address=Stu%F0%9F%8E%B6&key=xxx-Sm541do>

Warning message:

Geocoding "Stu" failed with error:

Source : <https://maps.googleapis.com/maps/api/geocode/json?address=Pennsylvania,+USA&key=xxx-Sm541do>

Source : <https://maps.googleapis.com/maps/api/geocode/json?address=w+isick%F0%9F%92%9B%F0%9F%9A&key=xxx-Sm541do>

Warning message:

Geocoding "w isick" failed with error:

Source : <https://maps.googleapis.com/maps/api/geocode/json?address=Kennesaw,+GA&key=xxx-Sm541do>

Source : <https://maps.googleapis.com/maps/api/geocode/json?address=West+Michigan&key=xxx-Sm541do>

Source : <https://maps.googleapis.com/maps/api/geocode/json?address=New+York,+USA&key=xxx-Sm541do>

Source : <https://maps.googleapis.com/maps/api/geocode/json?address=Pittsburgh&key=xxx-Sm541do>

Source : <https://maps.googleapis.com/maps/api/geocode/json?address=Letting+my+Freak+Flag+fly&key=xxx-Sm541do>

Warning message:

Geocoding "Letting my Freak ..." failed with error:

Source : <https://maps.googleapis.com/maps/api/geocode/json?address=Eastern+Time+Zone&key=xxx-Sm541do>

"Eastern Time Zone" not uniquely geocoded, using "630 old country rd, garden city, ny 11530, us"

Source : <https://maps.googleapis.com/maps/api/geocode/json?address=Seattle,+WA&key=xxx-Sm541do>

Source : <https://maps.googleapis.com/maps/api/geocode/json?address=Memphis&key=xxx-Sm541do>

Source : <https://maps.googleapis.com/maps/api/geocode/json?address=Toronto&key=xxx-Sm541do>  
Source : <https://maps.googleapis.com/maps/api/geocode/json?address=New+Jersey,+USA&key=xxx-Sm541do>  
Source : <https://maps.googleapis.com/maps/api/geocode/json?address=Los+Angeles,+CA&key=xxx-Sm541do>  
Source : <https://maps.googleapis.com/maps/api/geocode/json?address=Pewaukee,+WI&key=xxx-Sm541do>  
Source : <https://maps.googleapis.com/maps/api/geocode/json?address=704%E2%9E%A1336&key=xxx-Sm541do>  
Warning message:  
Geocoding "704336" failed with error:

Source : <https://maps.googleapis.com/maps/api/geocode/json?address=area+51&key=xxx-Sm541do>  
Source : <https://maps.googleapis.com/maps/api/geocode/json?address=CC&key=xxx-Sm541do>  
Source : <https://maps.googleapis.com/maps/api/geocode/json?address=Kentucky,+USA&key=xxx-Sm541do>  
Source : <https://maps.googleapis.com/maps/api/geocode/json?address=Butte,+MT&key=xxx-Sm541do>  
Source : <https://maps.googleapis.com/maps/api/geocode/json?address=Windham,+OH&key=xxx-Sm541do>  
Source : <https://maps.googleapis.com/maps/api/geocode/json?address=Clinton,+IL&key=xxx-Sm541do>  
Source : <https://maps.googleapis.com/maps/api/geocode/json?address=Urbana,+IL&key=xxx-Sm541do>  
Source : <https://maps.googleapis.com/maps/api/geocode/json?address=Pampa,+TX&key=xxx-Sm541do>  
Source : <https://maps.googleapis.com/maps/api/geocode/json?address=San+Francisco,+CA&key=xxx-Sm541do>  
Source : <https://maps.googleapis.com/maps/api/geocode/json?address=Long+Beach,+CA&key=xxx-Sm541do>  
Source : <https://maps.googleapis.com/maps/api/geocode/json?address=Houston+Tx&key=xxx-Sm541do>  
Source : <https://maps.googleapis.com/maps/api/geocode/json?address=Milwaukee,+WI&key=xxx-Sm541do>  
Source : <https://maps.googleapis.com/maps/api/geocode/json?address=Virginia,+USA&key=xxx-Sm541do>  
Source : <https://maps.googleapis.com/maps/api/geocode/json?address=Chino+Hills,+CA&key=xxx-Sm541do>  
Source : <https://maps.googleapis.com/maps/api/geocode/json?address=Emmett,+ID&key=xxx-Sm541do>  
Source : <https://maps.googleapis.com/maps/api/geocode/json?address=Arizona&key=xxx-Sm541do>  
Source : <https://maps.googleapis.com/maps/api/geocode/json?address=San+Jose,+CA&key=xxx-Sm541do>  
Source : <https://maps.googleapis.com/maps/api/geocode/json?address=Texas,+USA&key=xxx-Sm541do>  
Source : <https://maps.googleapis.com/maps/api/geocode/json?address=New+Jersey,+USA&key=xxx-Sm541do>  
Source : <https://maps.googleapis.com/maps/api/geocode/json?address=Moe+Bill&key=xxx-Sm541do>  
Warning message:  
Geocoding "Moe Bill" failed with error:

Source : <https://maps.googleapis.com/maps/api/geocode/json?address=Washington,+USA&key=xxx-Sm541do>  
"Washington, USA" not uniquely geocoded, using "washington, usa"  
Source : <https://maps.googleapis.com/maps/api/geocode/json?address=San+Diego,+CA&key=xxx-Sm541do>  
Source : <https://maps.googleapis.com/maps/api/geocode/json?address=United+States&key=xxx-Sm541do>  
Source : <https://maps.googleapis.com/maps/api/geocode/json?address=California,+USA&key=xxx-Sm541do>  
Source : <https://maps.googleapis.com/maps/api/geocode/json?address=Portland,+Oregon&key=xxx-Sm541do>  
Source : <https://maps.googleapis.com/maps/api/geocode/json?address=blacklivesmatter&key=xxx-Sm541do>  
Warning message:  
Geocoding "blacklivesmatter" failed with error:

Source : <https://maps.googleapis.com/maps/api/geocode/json?address=Denver&key=xxx-Sm541do>  
Source : <https://maps.googleapis.com/maps/api/geocode/json?address=Los+Angeles,+CA&key=xxx-Sm541do>  
Source : <https://maps.googleapis.com/maps/api/geocode/json?address=&key=xxx-Sm541do>  
Warning message in geocode(.x, output = output, source = source, messaging = messaging, :  
HTTP 400 Bad RequestSource : <https://maps.googleapis.com/maps/api/geocode/json?address=probably+have+dut...>  
Warning message:  
Geocoding "probably have dut..." failed with error:

Source : <https://maps.googleapis.com/maps/api/geocode/json?address=Somewhere+Near+Nowhere&key=xxx-Sm541do>  
Source : <https://maps.googleapis.com/maps/api/geocode/json?address=Australia&key=xxx-Sm541do>  
Source : <https://maps.googleapis.com/maps/api/geocode/json?address=Port+St.+Lucie,+FL&key=xxx-Sm541do>  
Source : <https://maps.googleapis.com/maps/api/geocode/json?address=Chicago,+IL&key=xxx-Sm541do>  
Source : <https://maps.googleapis.com/maps/api/geocode/json?address=Porterville,+CA&key=xxx-Sm541do>  
Source : <https://maps.googleapis.com/maps/api/geocode/json?address=Montreal&key=xxx-Sm541do>  
Source : <https://maps.googleapis.com/maps/api/geocode/json?address=Hamilton,+Ontario&key=xxx-Sm541do>  
Source : <https://maps.googleapis.com/maps/api/geocode/json?address=bay+area&key=xxx-Sm541do>  
Source : <https://maps.googleapis.com/maps/api/geocode/json?address=Fuckry&key=xxx-Sm541do>

Warning message:

Geocoding "Fuckry " failed with error:

Source : <https://maps.googleapis.com/maps/api/geocode/json?address=Garland,+TX&key=xxx-Sm541do>  
Source : <https://maps.googleapis.com/maps/api/geocode/json?address=Mesa,+AZ&key=xxx-Sm541do>  
Source : <https://maps.googleapis.com/maps/api/geocode/json?address=Philadelphia&key=xxx-Sm541do>  
Source : <https://maps.googleapis.com/maps/api/geocode/json?address=Frankfort,+IL&key=xxx-Sm541do>  
Source : <https://maps.googleapis.com/maps/api/geocode/json?address=Tennessee,+USA&key=xxx-Sm541do>  
Source : <https://maps.googleapis.com/maps/api/geocode/json?address=manchester,+ky&key=xxx-Sm541do>  
Source : <https://maps.googleapis.com/maps/api/geocode/json?address=South+Carolina,+USA&key=xxx-Sm541do>  
Source : <https://maps.googleapis.com/maps/api/geocode/json?address=Washington,+USA&key=xxx-Sm541do>  
"Washington, USA" not uniquely geocoded, using "washington, usa"  
Source : <https://maps.googleapis.com/maps/api/geocode/json?address=Wilkes-Barre,+PA&key=xxx-Sm541do>  
Source : <https://maps.googleapis.com/maps/api/geocode/json?address=Marshalltown,+IA&key=xxx-Sm541do>  
Source : <https://maps.googleapis.com/maps/api/geocode/json?address=Norfolk,+VA&key=xxx-Sm541do>  
Source : <https://maps.googleapis.com/maps/api/geocode/json?address=Boston,+MA&key=xxx-Sm541do>  
Source : <https://maps.googleapis.com/maps/api/geocode/json?address=Tijuana++San+Diego&key=xxx-Sm541do>  
Source : <https://maps.googleapis.com/maps/api/geocode/json?address=Rootstown,+Ohio&key=xxx-Sm541do>  
Source : <https://maps.googleapis.com/maps/api/geocode/json?address=Location&key=xxx-Sm541do>  
Source : <https://maps.googleapis.com/maps/api/geocode/json?address=Hollywood,+Los+Angeles&key=xxx-Sm541do>  
Source : <https://maps.googleapis.com/maps/api/geocode/json?address=Worcester,+MA//Springfield>  
Source : <https://maps.googleapis.com/maps/api/geocode/json?address=Anderson,+SC&key=xxx-Sm541do>  
Source : <https://maps.googleapis.com/maps/api/geocode/json?address=Salt+Lake+City,+UT&key=xxx-Sm541do>  
Source : <https://maps.googleapis.com/maps/api/geocode/json?address=C-U,+IL&key=xxx-Sm541do>  
Source : <https://maps.googleapis.com/maps/api/geocode/json?address=Salinas,+CA&key=xxx-Sm541do>  
Source : <https://maps.googleapis.com/maps/api/geocode/json?address=West+Des+Moines,+IA&key=xxx-Sm541do>  
Source : <https://maps.googleapis.com/maps/api/geocode/json?address=Chagrin+Falls,+OH&key=xxx-Sm541do>  
Source : <https://maps.googleapis.com/maps/api/geocode/json?address=Portland,+OR&key=xxx-Sm541do>  
Source : <https://maps.googleapis.com/maps/api/geocode/json?address=Minnesota&key=xxx-Sm541do>  
Source : <https://maps.googleapis.com/maps/api/geocode/json?address=avengers+tower&key=xxx-Sm541do>  
Source : <https://maps.googleapis.com/maps/api/geocode/json?address=United+States&key=xxx-Sm541do>  
Source : <https://maps.googleapis.com/maps/api/geocode/json?address=in+my+bed&key=xxx-Sm541do>  
Source : <https://maps.googleapis.com/maps/api/geocode/json?address=United+States&key=xxx-Sm541do>  
Source : <https://maps.googleapis.com/maps/api/geocode/json?address=New+York+City&key=xxx-Sm541do>  
Source : <https://maps.googleapis.com/maps/api/geocode/json?address=Pond+Creek,+OK&key=xxx-Sm541do>  
Source : <https://maps.googleapis.com/maps/api/geocode/json?address=New+York,+NY&key=xxx-Sm541do>  
Source : <https://maps.googleapis.com/maps/api/geocode/json?address=at+the+dog+park&key=xxx-Sm541do>  
Source : <https://maps.googleapis.com/maps/api/geocode/json?address=Durham,+NC&key=xxx-Sm541do>

Source : <https://maps.googleapis.com/maps/api/geocode/json?address=Texas,+USA&key=xxx-Sm541do>  
Source : <https://maps.googleapis.com/maps/api/geocode/json?address=Milwaukee,+WI&key=xxx-Sm541do>  
Source : <https://maps.googleapis.com/maps/api/geocode/json?address=Bay+Area,+California&key=xxx-Sm541do>  
Source : <https://maps.googleapis.com/maps/api/geocode/json?address=USA&key=xxx-Sm541do>  
Source : <https://maps.googleapis.com/maps/api/geocode/json?address=Los+Angeles,+CA,+USA,+Earth&key=xxx-Sm541do>  
Source : <https://maps.googleapis.com/maps/api/geocode/json?address=Waukegan,Illinois&key=xxx-Sm541do>  
Source : <https://maps.googleapis.com/maps/api/geocode/json?address=Braintree,+MA&key=xxx-Sm541do>  
Source : <https://maps.googleapis.com/maps/api/geocode/json?address=Houston,+TX&key=xxx-Sm541do>  
Source : <https://maps.googleapis.com/maps/api/geocode/json?address=Indianapolis&key=xxx-Sm541do>  
Source : <https://maps.googleapis.com/maps/api/geocode/json?address=la+la+land&key=xxx-Sm541do>  
Source : <https://maps.googleapis.com/maps/api/geocode/json?address=Charlotte,+NC&key=xxx-Sm541do>  
Source : <https://maps.googleapis.com/maps/api/geocode/json?address=London,+England&key=xxx-Sm541do>  
Source : <https://maps.googleapis.com/maps/api/geocode/json?address=Santiago,+Chile&key=xxx-Sm541do>  
Source : <https://maps.googleapis.com/maps/api/geocode/json?address=Leonard+Tx&key=xxx-Sm541do>  
Source : <https://maps.googleapis.com/maps/api/geocode/json?address=m+%E2%80%A2+r+%E2%80%A2+t+%E2%80%A2>

Warning message:

Geocoding "m r t c m" failed with error:

Source : <https://maps.googleapis.com/maps/api/geocode/json?address=Hell&key=xxx-Sm541do>  
Source : <https://maps.googleapis.com/maps/api/geocode/json?address=nola&key=xxx-Sm541do>  
Source : <https://maps.googleapis.com/maps/api/geocode/json?address=Western+NY&key=xxx-Sm541do>  
Source : <https://maps.googleapis.com/maps/api/geocode/json?address=Princeton,+NJ&key=xxx-Sm541do>  
Source : <https://maps.googleapis.com/maps/api/geocode/json?address=United+States&key=xxx-Sm541do>  
Source : <https://maps.googleapis.com/maps/api/geocode/json?address=Death+Row&key=xxx-Sm541do>  
Source : <https://maps.googleapis.com/maps/api/geocode/json?address=Old+Bridge,+NJ&key=xxx-Sm541do>  
Source : <https://maps.googleapis.com/maps/api/geocode/json?address=ONTARIO+CANADA&key=xxx-Sm541do>  
Source : <https://maps.googleapis.com/maps/api/geocode/json?address=Louisiana,+USA&key=xxx-Sm541do>  
Source : <https://maps.googleapis.com/maps/api/geocode/json?address=United+States&key=xxx-Sm541do>  
Source : <https://maps.googleapis.com/maps/api/geocode/json?address=Somewhere+in+Texas&key=xxx-Sm541do>  
"Somewhere in Texas " not uniquely geocoded, using "15477 markout central, forney, tx 75126, us"  
Source : <https://maps.googleapis.com/maps/api/geocode/json?address=Greensboro,+NC&key=xxx-Sm541do>  
Source : <https://maps.googleapis.com/maps/api/geocode/json?address=In+aisle+3+of+5th+&+Fashion&key=xxx-Sm541do>  
Source : <https://maps.googleapis.com/maps/api/geocode/json?address=Indiana,+USA&key=xxx-Sm541do>  
Source : <https://maps.googleapis.com/maps/api/geocode/json?address=Kalamazoo,+MI&key=xxx-Sm541do>  
Source : <https://maps.googleapis.com/maps/api/geocode/json?address=Traveling+the+USA&key=xxx-Sm541do>  
Source : <https://maps.googleapis.com/maps/api/geocode/json?address=Garden+Grove,+CA&key=xxx-Sm541do>  
Source : <https://maps.googleapis.com/maps/api/geocode/json?address=Washington,+DC&key=xxx-Sm541do>  
Source : <https://maps.googleapis.com/maps/api/geocode/json?address=334&key=xxx-Sm541do>  
Source : <https://maps.googleapis.com/maps/api/geocode/json?address=Akron,+OH&key=xxx-Sm541do>  
Source : <https://maps.googleapis.com/maps/api/geocode/json?address=in+bed&key=xxx-Sm541do>  
Source : <https://maps.googleapis.com/maps/api/geocode/json?address=Georgia&key=xxx-Sm541do>  
Source : <https://maps.googleapis.com/maps/api/geocode/json?address=Illinois,+USA&key=xxx-Sm541do>  
Source : <https://maps.googleapis.com/maps/api/geocode/json?address=North+Dakota,+USA&key=xxx-Sm541do>  
Source : <https://maps.googleapis.com/maps/api/geocode/json?address=smoking+a+phatwoods&key=xxx-Sm541do>

Warning message:

Geocoding "smoking a phatwoods" failed with error:

Source : <https://maps.googleapis.com/maps/api/geocode/json?address=Malibu,+CA&key=xxx-Sm541do>

```

Source : https://maps.googleapis.com/maps/api/geocode/json?address=ur+ma&key=xxx-Sm541do
Source : https://maps.googleapis.com/maps/api/geocode/json?address=Grandview,+WA&key=xxx-Sm541do
Source : https://maps.googleapis.com/maps/api/geocode/json?address=Ontario,+Canada&key=xxx-Sm541do
Source : https://maps.googleapis.com/maps/api/geocode/json?address=Texas,+USA&key=xxx-Sm541do
Source : https://maps.googleapis.com/maps/api/geocode/json?address=ATL&key=xxx-Sm541do
Source : https://maps.googleapis.com/maps/api/geocode/json?address=Salt+Lake+City,+UT&key=xxx-Sm541do
Source : https://maps.googleapis.com/maps/api/geocode/json?address=en+route&key=xxx-Sm541do
Source : https://maps.googleapis.com/maps/api/geocode/json?address=Brockton,+MA&key=xxx-Sm541do
Source : https://maps.googleapis.com/maps/api/geocode/json?address=Phoenix,+AZ&key=xxx-Sm541do
Source : https://maps.googleapis.com/maps/api/geocode/json?address=New+Jersey&key=xxx-Sm541do
Source : https://maps.googleapis.com/maps/api/geocode/json?address=Southern+California&key=xxx-Sm541do
Source : https://maps.googleapis.com/maps/api/geocode/json?address=Chicago,+IL&key=xxx-Sm541do
Source : https://maps.googleapis.com/maps/api/geocode/json?address=ayy+lmao&key=xxx-Sm541do
Warning message:
Geocoding "ayy lmao " failed with error:

```

```

Source : https://maps.googleapis.com/maps/api/geocode/json?address=Seattle,+WA&key=xxx-Sm541do
Source : https://maps.googleapis.com/maps/api/geocode/json?address=Canada&key=xxx-Sm541do
Source : https://maps.googleapis.com/maps/api/geocode/json?address=slytherin&key=xxx-Sm541do
Warning message:
Geocoding "slytherin" failed with error:

```

```

Source : https://maps.googleapis.com/maps/api/geocode/json?address=Lex+Corp&key=xxx-Sm541do
Source : https://maps.googleapis.com/maps/api/geocode/json?address=Georgia&key=xxx-Sm541do
Source : https://maps.googleapis.com/maps/api/geocode/json?address=New+york&key=xxx-Sm541do
Source : https://maps.googleapis.com/maps/api/geocode/json?address=Fayetteville,+AR&key=xxx-Sm541do
Source : https://maps.googleapis.com/maps/api/geocode/json?address=Odessa,+TX&key=xxx-Sm541do
Source : https://maps.googleapis.com/maps/api/geocode/json?address=Houston,+TX&key=xxx-Sm541do
Source : https://maps.googleapis.com/maps/api/geocode/json?address=Oklahoma+City,+OK&key=xxx-Sm541do
Source : https://maps.googleapis.com/maps/api/geocode/json?address=North+Carolina,+USA&key=xxx-Sm541do
Source : https://maps.googleapis.com/maps/api/geocode/json?address=Prosper,+TX&key=xxx-Sm541do
Source : https://maps.googleapis.com/maps/api/geocode/json?address=St+Louis,+MO&key=xxx-Sm541do
Source : https://maps.googleapis.com/maps/api/geocode/json?address=Palmdale,+CA&key=xxx-Sm541do
Source : https://maps.googleapis.com/maps/api/geocode/json?address=Bridgeport,+Chicago&key=xxx-Sm541do
Source : https://maps.googleapis.com/maps/api/geocode/json?address=New+York&key=xxx-Sm541do

```

```

In [68]: # Omitting NA values from the
         code <- na.omit(code)

```

```

In [69]: # Removing all the outliers from the map.
         install.packages("tidyverse")
         library(tidyverse)
         code <- code %>%
           filter(endsWith(code$address,"usa"))

```

```

Updating HTML index of packages in '.Library'
Making 'packages.html' ... done

```





[illegible]

[illegible]

```
[1] "Getting geocode data from Google: https://maps.googleapis.com/maps/api/geocode/json?latlng=
[1] "Getting geocode data from Google: https://maps.googleapis.com/maps/api/geocode/json?latlng=
[1] "Getting geocode data from Google: https://maps.googleapis.com/maps/api/geocode/json?latlng=
[1] "Getting geocode data from Google: https://maps.googleapis.com/maps/api/geocode/json?latlng=
[1] "Getting geocode data from Google: https://maps.googleapis.com/maps/api/geocode/json?latlng=
[1] "Getting geocode data from Google: https://maps.googleapis.com/maps/api/geocode/json?latlng=
[1] "Getting geocode data from Google: https://maps.googleapis.com/maps/api/geocode/json?latlng=
```

1. California 2. Illinois 3. South Carolina 4. Texas 5. California 6. Oregon 7. Pennsylvania 8. Georgia 9. Michigan 10. New York 11. Pennsylvania 12. New York 13. Washington 14. Tennessee 15. New Jersey 16. California 17. Wisconsin 18. Nevada 19. Kansas 20. Kentucky 21. Montana 22. Ohio 23. Illinois 24. Illinois 25. Texas 26. California 27. California 28. Texas 29. Wisconsin 30. Virginia 31. California 32. Idaho 33. Arizona 34. California 35. Texas 36. New Jersey 37. Washington 38. California 39. California 40. Oregon 41. Colorado 42. California 43. Kentucky 44. Florida 45. Illinois 46. California 47. California 48. Texas 49. Arizona 50. Pennsylvania 51. Illinois 52. Tennessee 53. Kentucky 54. South Carolina 55. Washington 56. Pennsylvania 57. Iowa 58. Virginia 59. Massachusetts 60. Ohio 61. Kansas 62. California 63. Massachusetts 64. South Carolina 65. Utah 66. Illinois 67. California 68. Iowa 69. Ohio 70. Oregon 71. Minnesota 72. Oregon 73. South Dakota 74. New York 75. Oklahoma 76. New York 77. Oklahoma 78. North Carolina 79. Texas 80. Wisconsin 81. California 82. California 83. Illinois 84. Massachusetts 85. Texas 86. Indiana 87. Florida 88. North Carolina 89. Texas 90. Michigan 91. Louisiana 92. New York 93. New Jersey 94. California 95. New Jersey 96. Louisiana 97. Texas 98. North Carolina 99. Georgia 100. Indiana 101. Michigan 102. New York 103. California 104. District of Columbia 105. Oklahoma 106. Ohio 107. Oklahoma 108. Georgia 109. Illinois 110. North Dakota 111. California 112. Washington 113. Texas 114. Georgia 115. Utah 116. Massachusetts 117. Arizona 118. New Jersey 119. California 120. Illinois 121. Washington 122. Georgia 123. New York 124. Arkansas 125. Texas 126. Texas 127. Oklahoma 128. North Carolina 129. Texas 130. Missouri 131. California 132. Illinois 133. New York

*Levels:* 1. 'Arizona' 2. 'Arkansas' 3. 'California' 4. 'Colorado' 5. 'District of Columbia' 6. 'Florida' 7. 'Georgia' 8. 'Idaho' 9. 'Illinois' 10. 'Indiana' 11. 'Iowa' 12. 'Kansas' 13. 'Kentucky' 14. 'Louisiana' 15. 'Massachusetts' 16. 'Michigan' 17. 'Minnesota' 18. 'Missouri' 19. 'Montana' 20. 'Nevada' 21. 'New Jersey' 22. 'New York' 23. 'North Carolina' 24. 'North Dakota' 25. 'Ohio' 26. 'Oklahoma' 27. 'Oregon' 28. 'Pennsylvania' 29. 'South Carolina' 30. 'South Dakota' 31. 'Tennessee' 32. 'Texas' 33. 'Utah' 34. 'Virginia' 35. 'Washington' 36. 'Wisconsin'

```
In [71]: Freq_frame <- as.data.frame(table(reved$state))
```

```
In [72]: Freq_frame
```

Var1	Freq
Arizona	3
Arkansas	1
California	21
Colorado	1
District of Columbia	1
Florida	2
Georgia	5
Idaho	1
Illinois	10
Indiana	2
Iowa	2
Kansas	2
Kentucky	3
Louisiana	2
Massachusetts	4
Michigan	3
Minnesota	1
Missouri	1
Montana	1
Nevada	1
New Jersey	5
New York	8
North Carolina	4
North Dakota	1
Ohio	4
Oklahoma	5
Oregon	4
Pennsylvania	4
South Carolina	3
South Dakota	1
Tennessee	2
Texas	13
Utah	2
Virginia	2
Washington	5
Wisconsin	3

```
In [73]: library("ggmap")
library("maptools")
library("maps")
library("ggplot2")

referencedata = map_data("state")
head(referencedata)
```

Checking rgeos availability: FALSE

Note: when rgeos is not available, polygon geometry

computations in maptools c

which has a restricted licence. It is disabled by default;  
to enable gpclib, type gpclibPermit()

long	lat	group	order	region	subregion
-87.46201	30.38968	1	1	alabama	NA
-87.48493	30.37249	1	2	alabama	NA
-87.52503	30.37249	1	3	alabama	NA
-87.53076	30.33239	1	4	alabama	NA
-87.57087	30.32665	1	5	alabama	NA
-87.58806	30.32665	1	6	alabama	NA

```
In [74]: state_and_levels = data.frame(Freq_frame$Var1, Freq_frame$Freq)
```

```
In [75]: state_and_levels
```

Freq_frame.Var1	Freq_frame.Freq
Arizona	3
Arkansas	1
California	21
Colorado	1
District of Columbia	1
Florida	2
Georgia	5
Idaho	1
Illinois	10
Indiana	2
Iowa	2
Kansas	2
Kentucky	3
Louisiana	2
Massachusetts	4
Michigan	3
Minnesota	1
Missouri	1
Montana	1
Nevada	1
New Jersey	5
New York	8
North Carolina	4
North Dakota	1
Ohio	4
Oklahoma	5
Oregon	4
Pennsylvania	4
South Carolina	3
South Dakota	1
Tennessee	2
Texas	13
Utah	2
Virginia	2
Washington	5
Wisconsin	3

```

In [76]: region_level = c()
  for(i in 1:nrow(referencedata)){
    region = referencedata$region[i]
    index = grep(region, tolower(state_and_levels$Freq_frame.Var1))
    region_level = append(region_level, state_and_levels$Freq_frame.Freq[index[1]])
  }

referencedata$region_level = region_level

In [77]: table(referencedata$region_level)
referencedata$final_regionlevel = sapply(referencedata$region_level, function(x) ifel.

```

1	2	3	4	5	8	10	13	21
2014	3117	2068	1714	1415	495	329	1088	516

```
In [78]: new <- data.frame(table(referencedata$region_level))
```

```
In [79]: referencedata
```

	long	lat	group	order	region	subregion	region_level	final_regionlevel
1	-87.46201	30.38968	1	1	alabama	NA	NA	High
2	-87.48493	30.37249	1	2	alabama	NA	NA	High
3	-87.52503	30.37249	1	3	alabama	NA	NA	High
4	-87.53076	30.33239	1	4	alabama	NA	NA	High
5	-87.57087	30.32665	1	5	alabama	NA	NA	High
6	-87.58806	30.32665	1	6	alabama	NA	NA	High
7	-87.59379	30.30947	1	7	alabama	NA	NA	High
8	-87.59379	30.28655	1	8	alabama	NA	NA	High
9	-87.67400	30.27509	1	9	alabama	NA	NA	High
10	-87.81152	30.25790	1	10	alabama	NA	NA	High
11	-87.88026	30.24644	1	11	alabama	NA	NA	High
12	-87.92037	30.24644	1	12	alabama	NA	NA	High
13	-87.95475	30.24644	1	13	alabama	NA	NA	High
14	-88.00632	30.24071	1	14	alabama	NA	NA	High
15	-88.01778	30.25217	1	15	alabama	NA	NA	High
16	-88.01205	30.26936	1	16	alabama	NA	NA	High
17	-87.99486	30.27509	1	17	alabama	NA	NA	High
18	-87.95475	30.27509	1	18	alabama	NA	NA	High
19	-87.90318	30.28082	1	19	alabama	NA	NA	High
20	-87.82870	30.28655	1	20	alabama	NA	NA	High
21	-87.80006	30.28655	1	21	alabama	NA	NA	High
22	-87.80006	30.32665	1	22	alabama	NA	NA	High
23	-87.81724	30.34385	1	23	alabama	NA	NA	High
24	-87.84016	30.38395	1	24	alabama	NA	NA	High
25	-87.85162	30.40114	1	25	alabama	NA	NA	High
26	-87.87453	30.41260	1	26	alabama	NA	NA	High
27	-87.90318	30.42406	1	27	alabama	NA	NA	High
28	-87.92610	30.44698	1	28	alabama	NA	NA	High
29	-87.93183	30.49281	1	29	alabama	NA	NA	High
30	-87.94329	30.52719	1	30	alabama	NA	NA	High
15570	-105.0289	45.00583	63	15570	wyoming	NA	NA	High
15571	-104.9258	45.00583	63	15571	wyoming	NA	NA	High
15572	-104.7825	45.00583	63	15572	wyoming	NA	NA	High
15573	-104.5820	45.00583	63	15573	wyoming	NA	NA	High
15574	-104.3413	45.00583	63	15574	wyoming	NA	NA	High
15575	-104.1580	45.00583	63	15575	wyoming	NA	NA	High
15576	-104.0549	45.00583	63	15576	wyoming	NA	NA	High
15577	-104.0549	44.58185	63	15577	wyoming	NA	NA	High
15578	-104.0549	44.18077	63	15578	wyoming	NA	NA	High
15579	-104.0606	44.13494	63	15579	wyoming	NA	NA	High
15580	-104.0549	43.84846	63	15580	wyoming	NA	NA	High
15581	-104.0606	43.49895	63	15581	wyoming	NA	NA	High
15582	-104.0663	43.47604	63	15582	wyoming	NA	NA	High
15583	-104.0606	43.00621	63	15583	wyoming	NA	NA	High
15584	-104.0606	42.61087	63	15584	wyoming	NA	NA	High
15585	-104.0549	41.99781	63	15585	wyoming	NA	NA	High
15586	-104.0606	41.69987	63	15586	wyoming	NA	NA	High
15587	-104.0606	41.56236	63	15587	wyoming	NA	NA	High
15588	-104.0606	41.39620	63	15588	wyoming	NA	NA	High
15589	-104.0606	41.00659	63	15589	wyoming	NA	NA	High
15590	-104.9429	41.01232	63	15590	wyoming	NA	NA	High



```
In [117]: options(repr.plot.width=4, repr.plot.height=2)
plot <- ggplot(referencedata, aes(referencedata$long, referencedata$lat)) +
  geom_polygon(aes(group = group, fill = referencedata$final_regionlevel), color="white") +
  scale_fill_manual(name="ILI Activity Level", values = c("red", "#43CD80", "#76EE00", "#FFA500"),
    theme(panel.background = element_blank()) +
    theme(axis.ticks = element_blank()) +
    theme(axis.text = element_blank()) + labs(x = NULL, y = NULL) + ggtitle("Influenza Twitter Analysis, 2019")

In [118]: plot
```

## Influenza Twitter Analysis, 2019



### 0.5 Alternative :: Stream API

```
In [ ]: install.packages("ROAuth")
library(ROAuth)
requestURL <- "https://api.twitter.com/oauth/request_token"
accessURL <- "https://api.twitter.com/oauth/access_token"
authURL <- "https://api.twitter.com/oauth/authorize"
consumerKey <- "XKCvkSC3TQDBJlLfaMJGbyuyT"
consumerSecret <- "SPkJ1siXeBh13GmwcrCYispF8eriDx7LQoIdMYfADodsG9neUe"

my_oauth <- OAuthFactory$new(consumerKey=consumerKey,
  consumerSecret=consumerSecret, requestURL=requestURL,
  accessURL=accessURL, authURL=authURL)

In [ ]: my_oauth$handshake(cainfo = system.file("CurlSSL", "cacert.pem", package = "RCurl"))
save(my_oauth, file="twitter-token.Rdata")

In [ ]: install.packages("streamR")
library(streamR)
```

```

load("twitter-token.Rdata")

filterStream(file.name = "tweets1.json", # Save tweets in a json file
             track = c("flu", "Influenza"), # Collect tweets mentioning either Affordab
             language = "en",
             location = c(-125, 25, -66, 50), # latitude/longitude pairs providing sou
             timeout = 120, # Keep connection alive for 60 seconds
             oauth = my_oauth) # Use my_oauth file as the OAuth credentials

In [ ]: tweets.df <- parseTweets("tweets1.json", simplify = FALSE)

In [ ]: tweets.df

```