CSE487/587: DATA INTENSIVE COMPUTING

Instructor: Bina Ramamurthy

DATA COLLECTION
AND
EXPLORATORY DATA ANALYSIS

Submitted By:

Aman Bhayana
50290968
Pratik Agarwal
50290970

PART 3- Twitter Application Development

Goal:

In this particular task, we are supposed to collect the tweets which are related to flu and then analyse the data. Following are the steps need to be carried out in order to complete this task.

- Collect about 2000 tweets by specifying the appropriate search query and categorize them based on the geo-location using google APIs. Following are the steps to be taken.
  - Convert search result tweets into dataframe
  - Lookup screen names from this dataframe
  - Convert these screen names into another dataframe
  - Keep only users (user names) with location info
  - Get the geocode of the locations from this dataframe
    .
- Compare: Now compare your map and with map from CDC for the same date ranges.
- Iterate: Compare with different query words related to flu.

Implementation

Following are the steps taken by us in order to achieve the goal.

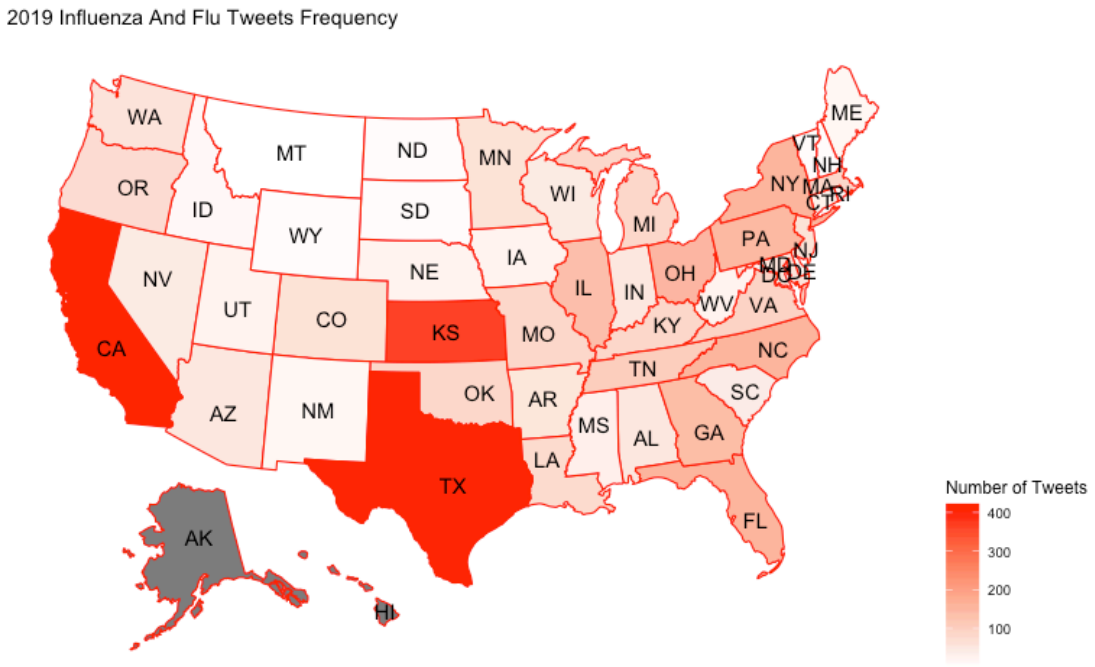
1. Installed and imported all the packages that are required to carry out the tasks.
2. Created an oauth token to connect with the twitter API
3. Registered with the google to access google API's
4. Created a function to collect the tweets using the APIs(Twitter, rtweet)
5. After extracting the tweets by using different and appropriate search query, we have created a function to clean all the tweets. Cleaning the data involves removing the tweets which don't have locations. Once the data is cleaned, we are saving the cleaned data in a csv file.

6. We are getting the geocodes in the form of longitude and latitude.
7. Once we get all the longitudes and latitudes, we are getting the state names using revgeocode function.
8. We then get the states along with their frequencies.
9. Plotting the graph.

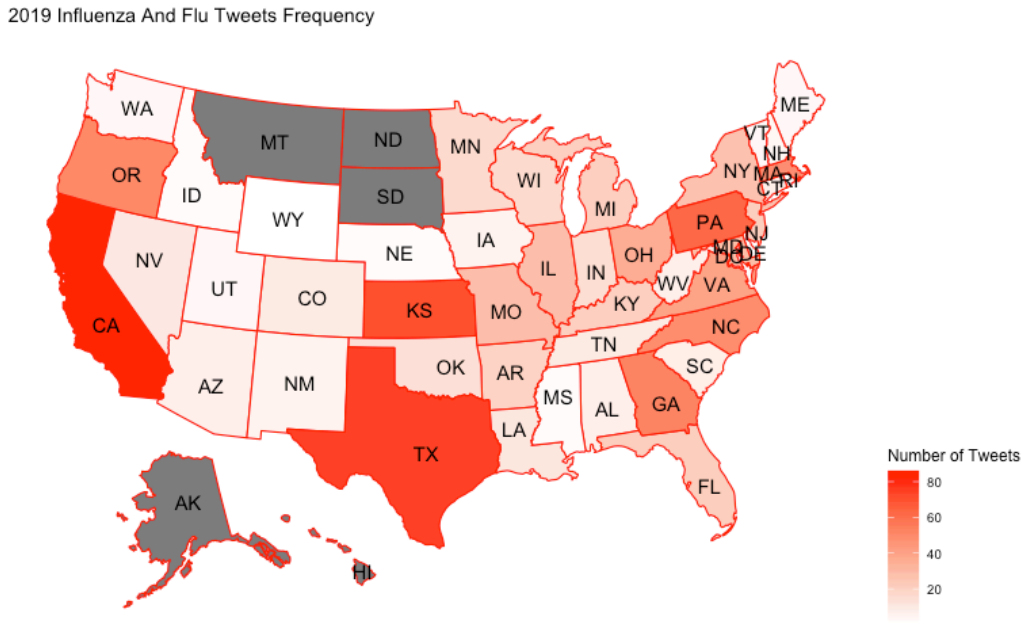
Observations:

Following are the graphs we got after the successful execution of all the above-mentioned steps.

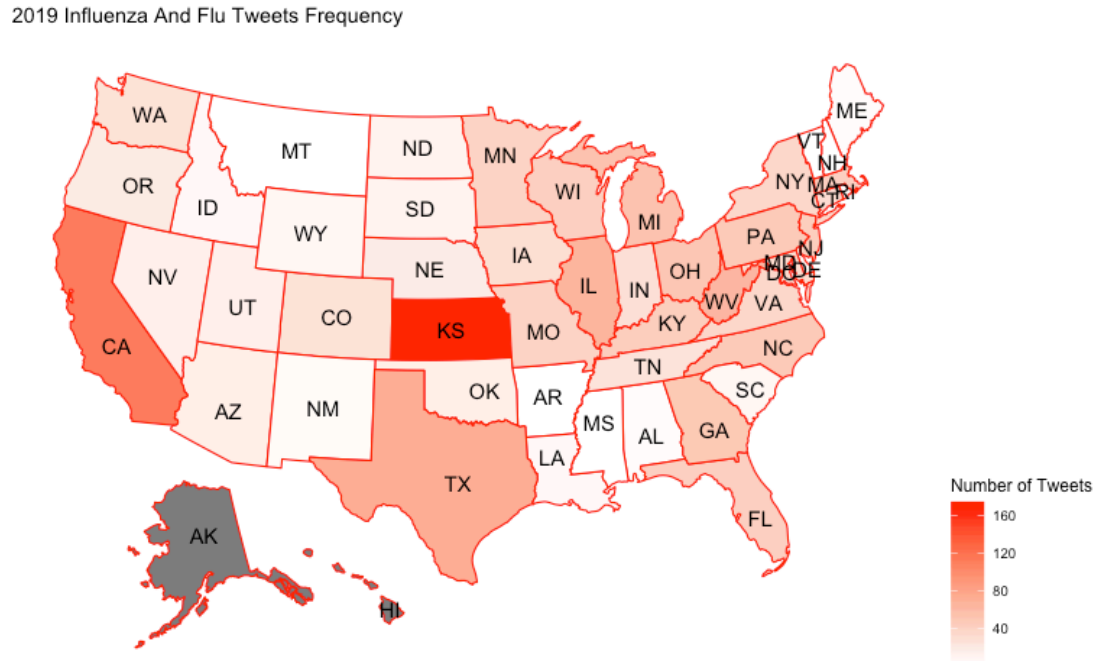
1. Map using search query – “flu”



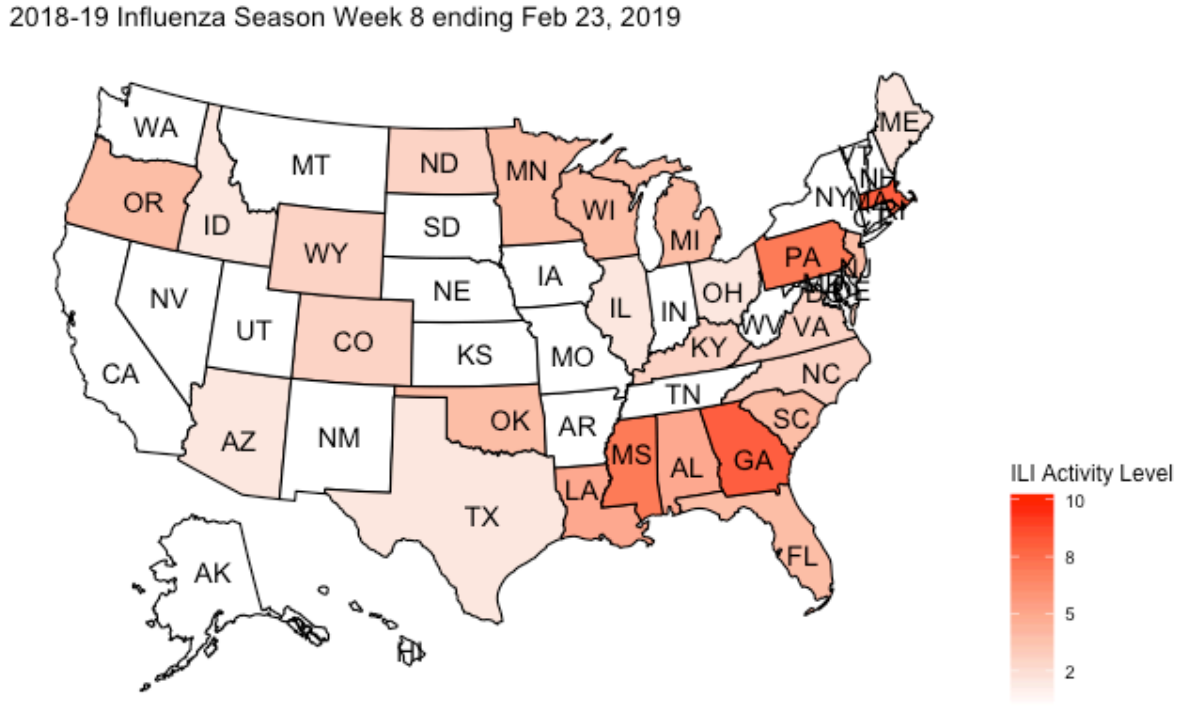
2. Map created by search query- “#flu”



3. Map created by using search query-“influenza”



4. CDC Heat map after averaging ILI activity level



As we could see that there are similarities in the graphs created by collecting tweets and the one that was created using the data available online.

Web Hosting:

We have hosted our shiny app here-https://agarwalpratik.shinyapps.io/fluanalysis/

Note: Since the free version of shiny server allows running time of only 25 hours at a stretch, this link might be down.

Sanple Code:

```
#----- All the utility functions ------------


getAndSaveTweets<- function(searchQuery,tweetsCount, oauthToken, outputFileName){...}
cleanAndSaveGeocodedTweets<- function (rawTweets, outputFileName ){...}
mergeGeocode<-function(...){...}
latlong2state <- function(pointsDF) {...}
getStateFrequency<-function(geocodes){...}

getStateFrequencyRevGeo<- function(geocode){...}
plotHeatMap <- function(data,location, frequency){...}

plotHeatMap_USMmap <- function(data, searchString)
{...}


head(cleanTweetsFlu)
```

```
head(state_freq)
plotHeatMap_USMmap(data = state_freq, "flu")

tweetsFlu <- getAndSaveTweets("flu",15000,twitter_token,"Twitter_data/raw_15k_flu_2mar.csv" )
tweetsHashFlu <- getAndSaveTweets("#flu",15000,twitter_token,"Twitter_data/raw_15k_HashFlu_2mar.csv" )
tweetsInfluenza <- getAndSaveTweets("influenza",15000,twitter_token,"Twitter_data/raw_15k_influenza_2mar.csv" )

tweetsFlu <-read.csv("Twitter_data/raw_15k_flu_2mar.csv")

#---------------------Cleaning the data and getting the geocodes. Saving it in a file------------------
cleanTweetsFlu <- cleanAndSaveGeocodedTweets(tweetsFlu,"Twitter_data/clean_15k_flu_2mar.csv")
cleanTweetsHashFlu <- cleanAndSaveGeocodedTweets(tweetsHashFlu,"Twitter_data/clean_15k_hash_flu_2mar.csv")
cleanTweetsInfluenza <- cleanAndSaveGeocodedTweets(tweetsInfluenza,"Twitter_data/clean_15k_influenza_2mar.csv")

cleanTweetsFlu <-read.csv("Twitter_data/clean_15k_flu_2mar.csv")
cleanTweetsHashFlu <- read.csv("Twitter_data/clean_15k_hash_flu_2mar.csv")
cleanTweetsInfluenza <-read.csv("Twitter_data/clean_15k_influenza_2mar.csv")

#---------------------Plotting geocodes of  all the tweets collected using search query as flu----------
#dfGeodataTweetsFluStateFreq<-getStateFrequency(cleanTweetsFlu)
dfGeodataTweetsFluStateFreq<-getStateFrequencyRevGeo(cleanTweetsFlu)
plotHeatMap_USMmap(dfGeodataTweetsFluStateFreq)
#plotHeatMap(dfGeodataTweetsFluStateFreq,dfGeodataTweetsFluStateFreq$usStatelocation,dfGeodataTweetsFluStateFreq$Freq


#---------------------Plotting geocodes of  all the tweets collected using search query as #flu----------

#dfGeodataTweetsHashFluStateFreq<-getStateFrequency(cleanTweetsHashFlu)
```

User Defined Functions:

Following are the functions created.

1. getAndSaveTweets – To retrieve and save it in csv files.
2. cleanAndSaveGeocodedTweets- To clean the data based on various parameters such as location.
3. mergeGeocode- To merge all the geocodes retrieved using the data collected by specifying various search query.
4. latlong2state- Not used any more as this utility uses spatial points.
5. getStateFrequencyRevGeo- Getting the states and their respective frequencies using revgeocode.

6. plotHeatMap- To plot the heat map using ggplot.
7. plotHeatMap_USMmap- Plot heat map using the library called USMap

Code Features:

1. **Abstraction**: We have tried to achieve abstraction so that users do not have to deal with the entire code.
2. **Reusability**: We have divided the steps into different components and each component is coded as a function. So, users can input files dynamically and can create the chart.
3. **Comments**: We have added comments where ever required.

Obstacles Encountered:

1. We were not able to retrieve the sufficient tweets using twitter API. Thus, we started collecting tweets using rtweet package.
2. After filtering out tweets, there were only a handful tweets left which had the proper location. Filtering out was a tedious task
3. Converting longitudes to latitudes using library other than sp.
4. While plotting the heat map, there were a few states that were not showing up. Thus, we changed the library to Usmap.