tweet-flu

March 29, 2018

Jayant Solanki Anant Gupta

0.1 Heatmap showing occurrence of Flu in the USA by collecting and analysing the Tweet data

0.1.1 Twitter Application Development

In [2]:
```
library(maptools)
library(twitteR)
library(ggplot2)
library('ggmap')
library(maps)
library(mapdata)
library(gridExtra)
library(grid)
```

In [3]:
```
#Here we are setting up the twitter api for use
api_key <- "xxxx"
api_secret <- "xxxx"
token <- "xxxx"
token_secret <- "xxxx"
setup_twitter_oauth(api_key, api_secret, token, token_secret)
```

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

In [4]:
```
# Here we are collecting the tweet having any of these keywords "flu OR #flu OR influe
# Geocode argument is used by passing the longitude and latitude values for the centre
# 3881 miles to cover the entire reagon of US
# In this example we have taken multiple tags in the same searchTwitter query. But in
# the tweets separately for flu and influenza/fever keywords

tweets <- searchTwitter("flu OR #flu OR influenza OR #Influenza OR fever OR #fluseason"
tweets_df <-twListToDF(tweets)


# We are saving the tweets collected in a csv file
#write.csv(tweets_df, "C:/Users/anu21/Anant/Google Drive/MASTERS/Courses/Spring 2018/D
```

In [5]:
```
# Here we are getting the finding the unique screen names and then looking up for the
# After geting the users we are cleaning the result to remove any values that might cr
# Finally we are saving the locations in a csv file

screenNames <- unique(tweets_df$screenName)
screenNames <- unique(tweets_df$screenName)
userDF <- twListToDF(lookupUsers(screenNames))
screenNamedf <- data.frame(userDF$screenName,userDF$location)
colnames(screenNamedf) <- c("ScreenName","Location")
screenNamedf <- na.omit(screenNamedf)
screenNamedf <- screenNamedf[length(screenNamedf$Location) != 0,]
screenNamedf <- screenNamedf[screenNamedf$Location != " ",]
screenNamedf <- screenNamedf[screenNamedf$Location != "",]
screenNamedf <- screenNamedf[screenNamedf$Location != "   ",]


#write.csv(screenNamedf, paste("C:/Users/anu21/Anant/Google Drive/MASTERS/Courses/Spri
```

In [6]:
```
#Here we are loading the csv file saved earlier which has the Location details and use
# and longitude values for each tweets. Here we have loaded only few location to save
# Finally all the Long and Lat values were saved in a csv file and the process was rep
# geo-location per day


path <- "ScreenName_Location(Flu).csv"
plot <- read.csv(file=path,sep=",")
geocode <- geocode(as.character(plot$Location[1:20]))
plotPoints <- data.frame(geocode)
plotPoints <- na.omit(plotPoints)


#write.csv(plotPoints, "C:/Users/anu21/Anant/Google Drive/MASTERS/Courses/Spring 2018/
```

```
Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=ohio--chicago-
Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=Indianapolis&s
Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=United%20State
Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=New%20Hampshir
Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=California&sens
Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=Circle%20Pines
Warning message:
"geocode failed with status ZERO_RESULTS, location = "Circle Pines, MN USA <ed><U+00A0><U+00BC
Warning message:
"geocode failed with status OVER_QUERY_LIMIT, location = "Naperville, IL""Information from URL
Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=Maryland&senso
Warning message:
"geocode failed with status OVER_QUERY_LIMIT, location = "Maryland""Information from URL : htt
.Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=lincolnshire&s
.Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=Orlando,%20FL&
```

```
.Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=London&sensor=
.Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=New%20York&ser
Warning message:
"geocode failed with status OVER_QUERY_LIMIT, location = "New York"".Information from URL : htt
.Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=Forest%20Hill&
.Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=Alberta,%20Can
Warning message:
"geocode failed with status OVER_QUERY_LIMIT, location = "Alberta, Canada"".Information from UR
Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=Minnesota&sensor
Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=Takoma%20Park,%
Warning message:
"geocode failed with status OVER_QUERY_LIMIT, location = "Takoma Park, MD, USA""
```

In [7]:
```
# Here we are loading the csv file which contains the longitude and latitude values of
# And then we are using the latlong2state function to get the state name from the lon-
# We have referred https://gist.github.com/rweald/4720788 for lonlat2state function

path <- "Lon_Lat(Flu).csv"
plot <- read.csv(file=path,sep=",")
plot <- data.frame(plot$lon,plot$lat)

latlong2state <- function(pointsDF) {
  states <- map('state', fill=TRUE, col="transparent", plot=FALSE)
  IDs <- sapply(strsplit(states$names, ":"), function(x) x[1])
  states_sp <- map2SpatialPolygons(states, IDs=IDs, proj4string=CRS("+proj=longlat +dat
  pointsSP <- SpatialPoints(pointsDF,proj4string=CRS("+proj=longlat +datum=wgs84"))
  indices <- over(pointsSP, states_sp)
  stateNames <- sapply(states_sp@polygons, function(x) x@ID)
  stateNames[indices]
}

state_name <- latlong2state(plot)
state_name <- na.omit(state_name)

#write.csv(state_name, "C:/Users/anu21/Anant/Google Drive/MASTERS/Courses/Spring 2018/
```

In [1]:
```
# Frequency of tweets statewise for FLU DATASET
# Here we use the statenames data and then create a table which shows the frequency of
# We have saved states names differently for teweets with Flu and Influenza tags in it
# In this cell we are first loading the tweets dataset having FLu tags in it


path <- "State(Flu).csv"
state_name <- read.csv(file=path,sep=",")
state <- table(state_name$x)
flu_statedata <- as.data.frame(state)
colnames(flu_statedata) <- c("StateName","Frequency")
flu_statedata
```

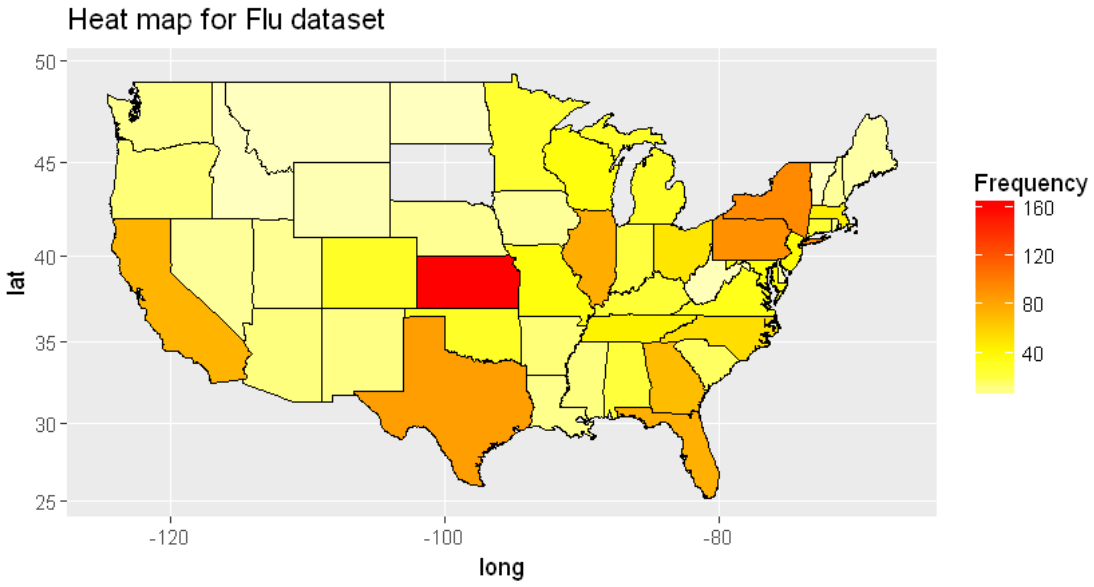
StateName	Frequency
alabama	21
arizona	11
arkansas	12
california	73
colorado	29
connecticut	23
delaware	5
district of columbia	36
florida	75
georgia	69
idaho	1
illinois	76
indiana	19
iowa	7
kansas	161
kentucky	23
louisiana	9
maine	5
maryland	36
massachusetts	44
michigan	29
minnesota	25
mississippi	11
missouri	35
montana	1
nebraska	8
nevada	7
new hampshire	7
new jersey	37
new mexico	10
new york	94
north carolina	53
north dakota	1
ohio	49
oklahoma	29
oregon	12
pennsylvania	91
rhode island	9
south carolina	15
tennessee	42
texas	84
utah	8
vermont	1
virginia	33
washington	9
west virginia	2
wisconsin	35
wyoming	3

In [2]:
```r
# Frequency of tweets statewise for Influenza DATASET
# In this cell we are first loading the tweet dataset having Influenza tags in it and
# frequency in each state


path <- "State(Influenza).csv"
state_name <- read.csv(file=path,sep=",")
state <- table(state_name$x)
influenza_statedata <- as.data.frame(state)
colnames(influenza_statedata) <- c("StateName","Frequency")
influenza_statedata
```

StateName	Frequency
alabama	3
arizona	19
arkansas	35
california	34
colorado	48
connecticut	4
delaware	1
district of columbia	3
florida	16
georgia	8
idaho	2
illinois	11
indiana	12
iowa	41
kansas	355
kentucky	1
louisiana	18
maine	1
maryland	3
massachusetts	10
michigan	6
minnesota	39
missouri	79
nebraska	25
nevada	6
new jersey	2
new mexico	8
new york	12
north carolina	4
ohio	6
oklahoma	49
pennsylvania	9
rhode island	3
south carolina	3
south dakota	6
tennessee	25
texas	45
utah	23
vermont	2
virginia	7
washington	3
west virginia	2
wisconsin	24
wyoming	12

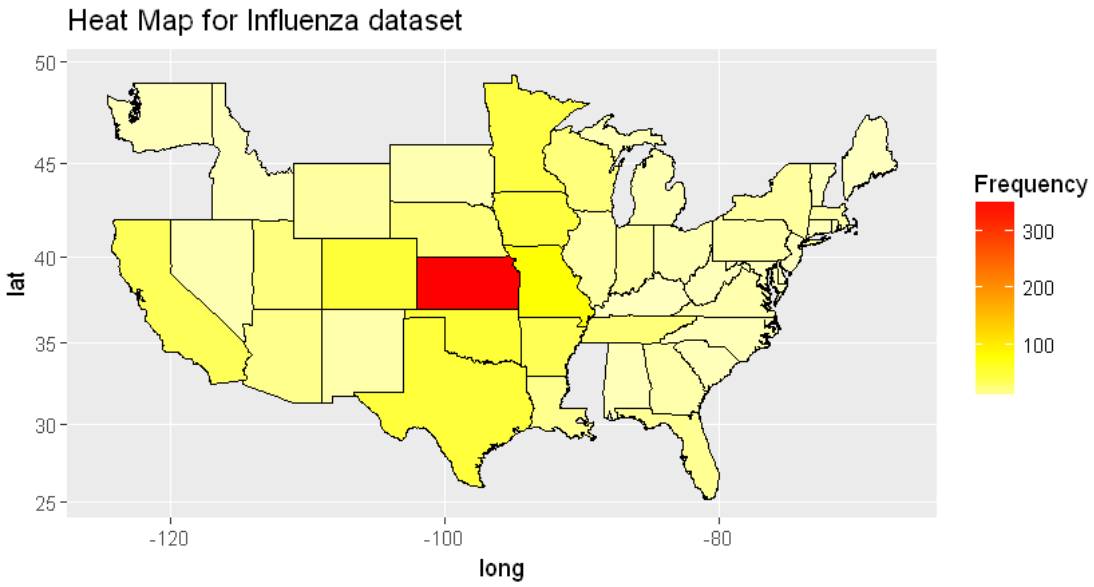
```
In [10]: # Finally we have all the count of tweets state-wise in statedata. We use this data t
         # ggplot and other map functions. We have referred https://stackoverflow.com/question
```

```
# Map for Flu dataset


flu_statedata$region <- tolower(flu_statedata$StateName)
states <- map_data("state")
map.df <- merge(states,flu_statedata, by="region", all.flu_statedata=T)
map.df <- map.df[order(map.df$order),]
ggplot(map.df, aes(x=long,y=lat,group=group))+ ggtitle("Heat map for Flu dataset")+
  geom_polygon(aes(fill=Frequency))+
  geom_path()+
  scale_fill_gradientn(colours=rev(heat.colors(10)),na.value="grey90")+
  coord_map() -> flu_plot
flu_plot
```

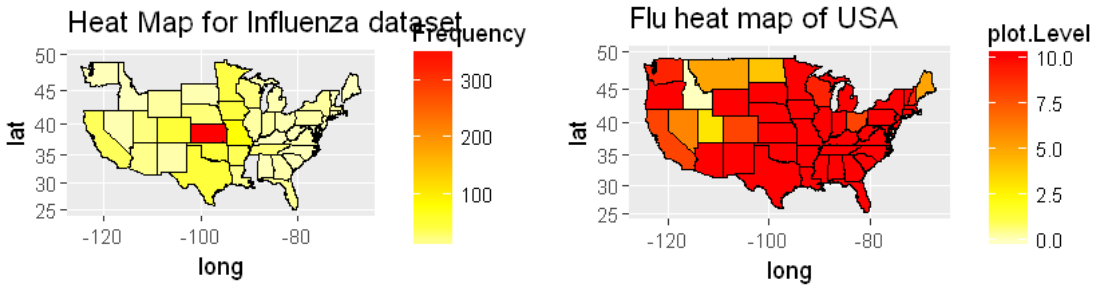


In [11]: *# Map for Influenza dataset*

```
influenza_statedata$region <- tolower(influenza_statedata$StateName)
states <- map_data("state")
map.df <- merge(states,influenza_statedata, by="region", all.influenza_statedata=T)
map.df <- map.df[order(map.df$order),]
ggplot(map.df, aes(x=long,y=lat,group=group))+
  geom_polygon(aes(fill=Frequency))+
  geom_path()+ ggtitle("Heat Map for Influenza dataset") +
  scale_fill_gradientn(colours=rev(heat.colors(10)),na.value="grey90")+
  coord_map() -> influenza_plot
influenza_plot
```

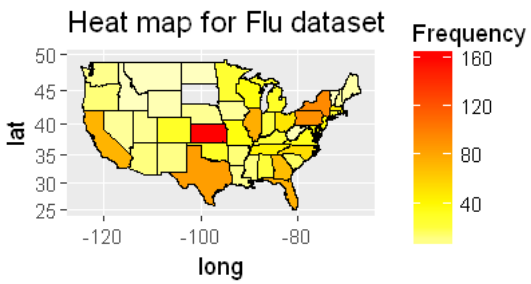


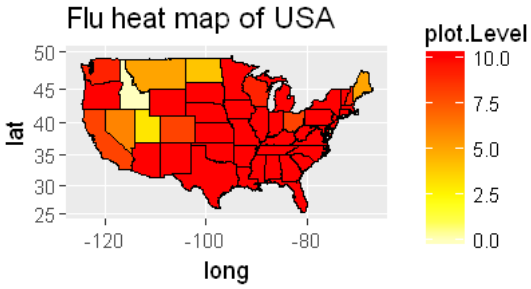
In [14]:
```
# # Comparision of our influenza dataset with the CDC's Flu Heat map of USA

path <- "StateDatabyWeekforMap_2017-18week4-4.csv"
plot <- read.csv(file=path,sep=",")
plot <- data.frame(plot$STATENAME,plot$ACTIVITY.LEVEL,plot$Level)
plot$region <- tolower(plot$plot.STATENAME)
states <- map_data("state")
map.df <- merge(states,plot, by="region", all.plot=T)
map.df <- map.df[order(map.df$order),]

ggplot(map.df, aes(x=long,y=lat,group=group))+ ggtitle("Flu heat map of USA ")+
     geom_polygon(aes(fill=plot.Level))+
     geom_path()+
     scale_fill_gradientn(colours=rev(heat.colors(10)),na.value="grey90")+
     coord_map() -> cdc_plot

grid.arrange(influenza_plot, cdc_plot, ncol = 2)
```



In [15]: *# Comparision of our flu dataset with the CDC's Flu Heat map of USA*

```
grid.arrange(flu_plot, cdc_plot, ncol = 2)
```


Heat map for Flu dataset
Frequency
160
120
80
40
lat
50
45
40
35
30
25
-120
-100
-80
long


Flu heat map of USA
plot.Level
10.0
7.5
5.0
2.5
0.0
lat
50
45
40
35
30
25
-120
-100
-80
long