

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

```
library(twitterR)
library(ggplot2)
library(plotly)
```

```
##
## Attaching package: 'plotly'
```

```
## The following object is masked from 'package:ggplot2':
##
##   last_plot
```

```
## The following object is masked from 'package:stats':
##
##   filter
```

```
## The following object is masked from 'package:graphics':
##
##   layout
```

```
library(maps)
library("ggmap")
```

```
##
## Attaching package: 'ggmap'
```

```
## The following object is masked from 'package:plotly':
##
##   wind
```

```
library("maptools")
```

```
## Loading required package: sp
```

```
## Checking rgeos availability: FALSE
##   Note: when rgeos is not available, polygon geometry      computations in maptools depend
on gpclib,
##   which has a restricted licence. It is disabled by default;
##   to enable gpclib, type gpclibPermit()
```

```
library(revgeo)
library(plyr)
```

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

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

```
## The following objects are masked from 'package:plotly':  
##  
## arrange, mutate, rename, summarise
```

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

```
library(dplyr)
```

```
##  
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:plyr':  
##  
## arrange, count, desc, failwith, id, mutate, rename, summarise,  
## summarize
```

```
## The following objects are masked from 'package:twitter':  
##  
## id, location
```

```
## The following objects are masked from 'package:stats':  
##  
## filter, lag
```

```
## The following objects are masked from 'package:base':  
##  
## intersect, setdiff, setequal, union
```

```
library(RColorBrewer)
```

```
consumer_key = "8gVEzu0RXXXzTZ4XXXB5JWdhk"  
consumer_secret = "5iBIZzq2dhEsUYQbjMLEXXXWtgyYy8jOA2qERXXjtn7Ah2y8"  
access_token = "548674986-XXXXXXUp4fYPDmi73B4Ia4Qo3uMqC02F2EUft4Gi"  
access_secret = "WqpL6j7x1IXXXXQSMZBiIEml9hEHEDmRCXGzBvdVDW2E"  
  
#setup_twitter_oauth(consumer_key, consumer_secret, access_token, access_secret)
```

```
# Collected tweets for flu and stored as csv file
# 2/13/2018

#tw_data1 = searchTwitter("flu OR #flu OR influenza OR #influenza OR H1N1 OR #H1N1 OR H3N2 OR
#
# seasonalFlu OR #seasonalflu OR #fluinfection", n = 10000, since = '20
15-11-08')

#tw_data2 = searchTwitter("flu OR #flu OR influenza OR #influenza OR H1N1 OR #H1N1 OR H3N2 OR
#
# seasonalFlu OR #seasonalflu OR #fluinfection", n = 10000, since = '20
18-02-14')

#tw_data3 = searchTwitter("flu OR #flu OR influenza OR #influenza OR H1N1 OR #H1N1 OR H3N2 OR
#
# seasonalFlu OR #seasonalflu OR #fluinfection", n = 15000, since = '20
18-02-16')

#tw_data4 = searchTwitter("flu OR #flu OR influenza OR #influenza OR H1N1 OR #H1N1 OR H3N2 OR
#
# seasonalFlu OR #seasonalflu OR #fluinfection", n = 15000, since = '20
18-02-26')

#tw_data5 = searchTwitter("flu OR #flu OR influenza OR #influenza OR H1N1 OR #H1N1 OR H3N2 OR
#
# seasonalFlu OR #seasonalflu OR #fluinfection", n = 25000, since = '20
18-03-02')

#tw_data6 = searchTwitter("flu OR #flu OR influenza OR #influenza OR H1N1 OR #H1N1 OR H3N2 OR
#
# seasonalFlu OR #seasonalflu OR #fluinfection", n = 25000, since = '20
18-03-06')

#tw_data7 = searchTwitter("flu OR #flu OR influenza OR #influenza OR H1N1 OR #H1N1 OR H3N2 OR
#
# seasonalFlu OR #seasonalflu OR #fluinfection", n = 25000, since = '20
18-03-07')

#tw_data8 = searchTwitter("flu OR #flu OR influenza OR #influenza OR H1N1 OR #H1N1 OR H3N2 OR
#
# seasonalFlu OR #seasonalflu OR #fluinfection", n = 25000, since = '20
18-03-08')

#df1 = twListToDF(tw_data1)
#df2 = twListToDF(tw_data2)
#df3 = twListToDF(tw_data3)
#df4 = twListToDF(tw_data4)
#df5 = twListToDF(tw_data5)
#df6 = twListToDF(tw_data6)
#df7 = twListToDF(tw_data7)
#df8 = twListToDF(tw_data8)

#write.csv(df1, "tweet_data1.csv")
#write.csv(df2, "tweet_data2.csv")
#write.csv(df3, "tweet_data3.csv")
#write.csv(df4, "tweet_data4.csv")
#write.csv(df5, "tweet_data5.csv")
#write.csv(df6, "tweet_data6.csv")
#write.csv(df7, "tweet_data7.csv")
#write.csv(df8, "tweet_data8.csv")
```

```
## Read tweet data from stored csv files
df1 = read.csv("C:\\Users\\Monalisa Mishra\\Desktop\\DIC\\Lab1\\Part3\\tweet_data1.csv")
df2 = read.csv("C:\\Users\\Monalisa Mishra\\Desktop\\DIC\\Lab1\\Part3\\tweet_data2.csv")
df3 = read.csv("C:\\Users\\Monalisa Mishra\\Desktop\\DIC\\Lab1\\Part3\\tweet_data3.csv")
df4 = read.csv("C:\\Users\\Monalisa Mishra\\Desktop\\DIC\\Lab1\\Part3\\tweet_data4.csv")
df5 = read.csv("C:\\Users\\Monalisa Mishra\\Desktop\\DIC\\Lab1\\Part3\\tweet_data5.csv")
df6 = read.csv("C:\\Users\\Monalisa Mishra\\Desktop\\DIC\\Lab1\\Part3\\tweet_data6.csv")
df7 = read.csv("C:\\Users\\Monalisa Mishra\\Desktop\\DIC\\Lab1\\Part3\\tweet_data7.csv")
df8 = read.csv("C:\\Users\\Monalisa Mishra\\Desktop\\DIC\\Lab1\\Part3\\tweet_data8.csv")

# To remove extra index coulmn
df1 = df1[,-c(1)]
df2 = df2[,-c(1)]
df3 = df3[,-c(1)]
df4 = df4[,-c(1)]
df5 = df5[,-c(1)]
df6 = df6[,-c(1)]
df7 = df7[,-c(1)]
df8 = df8[,-c(1)]

# Combine them to form a common data frame
tweet_data = rbind(df1,df2,df3,df4,df5,df6,df7,df8)

tweet_data$longitude = as.numeric(tweet_data$longitude)
tweet_data$latitude = as.numeric(tweet_data$latitude)
```

```
## Warning: NAs introduced by coercion
```

```
# Separate tweets which have Longitude and Latitude information
tweets = tweet_data[!is.na(tweet_data$longitude | tweet_data$latitude), ]
tweets = tweets[,c("screenName","longitude","latitude")]

dim(tweets) # total 288 tweets with locations
```

```
## [1] 288 3
```

Collected 150000 tweets using keywords related to flu and stored them as csv files. Later, read those files and combined them to one dataframe and extracted tweets which had longitude and latitude information that led to 288 tweets in total.

However, tweets are collected from all parts of the world and we require tweets from United States. Using US tweets and their location we can map them onto US map and get an overview.

```
# Get US tweets
for (i in 1:nrow(tweets)){
  tweets$city[i] = revgeo(tweets$longitude[i], tweets$latitude[i], output='hash', item='city')
  tweets$state[i] = revgeo(tweets$longitude[i], tweets$latitude[i], output='hash', item='state')
  tweets$country[i] = revgeo(tweets$longitude[i], tweets$latitude[i], output='hash', item='country')
}
```

```

11"
## [1] "Getting geocode data from Photon: http://photon.komoot.de/reverse?lon=28.38111877&lat=3
7.16932806"
## [1] "Getting geocode data from Photon: http://photon.komoot.de/reverse?lon=28.38111877&lat=3
7.16932806"
## [1] "Getting geocode data from Photon: http://photon.komoot.de/reverse?lon=28.38111877&lat=3
7.16932806"
## [1] "Getting geocode data from Photon: http://photon.komoot.de/reverse?lon=-38.97235&lat=-12.
244583"
## [1] "Getting geocode data from Photon: http://photon.komoot.de/reverse?lon=-38.97235&lat=-12.
244583"
## [1] "Getting geocode data from Photon: http://photon.komoot.de/reverse?lon=-38.97235&lat=-12.
244583"
## [1] "Getting geocode data from Photon: http://photon.komoot.de/reverse?lon=-74.0007613&lat=4
0.7207559"
## [1] "Getting geocode data from Photon: http://photon.komoot.de/reverse?lon=-74.0007613&lat=4
0.7207559"
## [1] "Getting geocode data from Photon: http://photon.komoot.de/reverse?lon=-74.0007613&lat=4
0.7207559"
## [1] "Getting geocode data from Photon: http://photon.komoot.de/reverse?lon=1.29620869&lat=52.
62843255"
## [1] "Getting geocode data from Photon: http://photon.komoot.de/reverse?lon=1.29620869&lat=52.
62843255"
## [1] "Getting geocode data from Photon: http://photon.komoot.de/reverse?lon=1.29620869&lat=52.
62843255"
## [1] "Getting geocode data from Photon: http://photon.komoot.de/reverse?lon=-0.1459&lat=51.544
7"
## [1] "Getting geocode data from Photon: http://photon.komoot.de/reverse?lon=-0.1459&lat=51.544
7"
## [1] "Getting geocode data from Photon: http://photon.komoot.de/reverse?lon=-0.1459&lat=51.544
7"
## [1] "Getting geocode data from Photon: http://photon.komoot.de/reverse?lon=-0.166667&lat=51.4
833"
## [1] "Getting geocode data from Photon: http://photon.komoot.de/reverse?lon=-0.166667&lat=51.4
833"
## [1] "Getting geocode data from Photon: http://photon.komoot.de/reverse?lon=-0.166667&lat=51.4
833"

```

```

tweets = tweets[(tweets$country == "United States of America"),]

# Remove Duplicates
tweets = tweets[!duplicated(tweets), ]

no_of_tweets = dim(tweets)
print(paste0("Total number of USA tweets related to Flu: ", no_of_tweets[1]))

```

```
## [1] "Total number of USA tweets related to Flu: 173"
```

```

# Fill the states map according to Frequency of Tweets
tweets$region = tolower(tweets$state)

tweet_states = data.frame(tweets[c(5)])
tweet_states = as.data.frame(table(unlist(tweet_states)))
tweet_states$region = tolower(tweet_states$Var1)

colnames(tweet_states) = c("States","Frequency","region")

states = map_data("state")
map.df = merge(states,tweet_states, by = "region", all.x = T)
map.df = map.df[order(map.df$order),]

# Get the code of the states to fill in the map
code = data.frame(setNames(state.abb, state.name))
code$region <- rownames(code)
colnames(code) = c("code","region")
code$region = tolower(code$region)

map.df = merge(map.df,code, by = "region", all.x = T)
map.df$hover <- with(map.df, paste(States, '<br>',"Frequency", Frequency))

# States with Frequency of tweets from its people
tweet_states[,c(1,2)]

```

##	States	Frequency
## 1	Alabama	2
## 2	Arizona	1
## 3	Arkansas	1
## 4	California	31
## 5	Colorado	2
## 6	District of Columbia	1
## 7	Florida	5
## 8	Georgia	5
## 9	Illinois	3
## 10	Indiana	1
## 11	Iowa	1
## 12	Kentucky	2
## 13	Louisiana	2
## 14	Maryland	3
## 15	Massachusetts	8
## 16	Minnesota	2
## 17	Missouri	3
## 18	Nevada	1
## 19	New York	15
## 20	North Carolina	1
## 21	Ohio	2
## 22	Oklahoma	1
## 23	Oregon	1
## 24	Pennsylvania	5
## 25	Puerto Rico	1
## 26	South Carolina	1
## 27	South Dakota	1
## 28	Tennessee	4
## 29	Texas	58
## 30	Utah	1
## 31	Virginia	2
## 32	Washington	4
## 33	Wisconsin	2

```

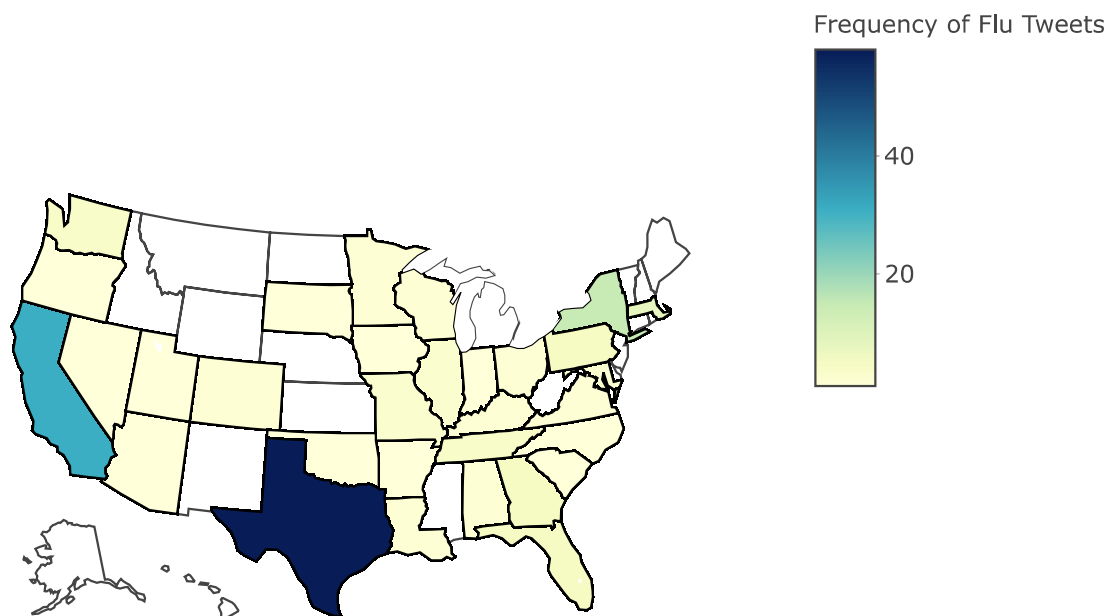
# give state boundaries a white border
l <- list(color = toRGB("white"), width = 2)
# specify some map projection/options
g <- list(
  scope = 'usa',
  projection = list(type = 'albers usa'),
  showlakes = TRUE,
  lakecolor = toRGB('white')
)

plot_geo(map.df, locationmode = 'USA-states') %>%
  add_trace(
    z = ~Frequency, text = ~hover, locations = ~code,
    color = ~Frequency, colors = brewer.pal(9, "YlGnBu")
  ) %>%
  colorbar(title = "Frequency of Flu Tweets") %>%
  layout(
    title = 'Frequency of Influenza in USA based on Tweets about Flu',
    geo = g
  )

```

```
## Warning: Ignoring 3737 observations
```

Frequency of Influenza in USA based on Tweets about Flu



So, we see that people from Texas and California tweet about flu, which indicates that they are the most affected. Even health officials say that California has seen unusually high numbers of flu-related deaths and hospitalizations, which bolsters our findings. Texas had seen 6,832 combined adult flu and pneumonia deaths and

12 flu-related pediatric deaths as of the state's most recent report March 29. After that New York also can be taken into consideration for flu cases.