In [15]:

```
###################################
#ubitname: SAISAOKH(50168989)
#assignment:CSE487 LAB1
#instructor: BINA RAMAMURTHY
#dueDate: 2/18/2017
###################################

#mapping twitter followers in r
#https://www.r-bloggers.com/mapping-twitter-followers-in-r/

#Gathering Tweets, geocoding users, and plotting them
#https://gist.github.com/dsparks/4329876

library(twitteR)
library(maps)
library(ggplot2)
library(ggmap)
library(plyr)

#library(gdata)
#install.package("gdata")
#install.packages("maptool")#waring msg
#library(maptools)
library(dismo)
#library(TwitteR2Mongo)


#doInstall <- TRUE
#toInstall <- c("twitteR", "dismo", "maps", "ggplot2")
#if(doInstall){install.packages(toInstall, repos = "http://cran.us.r-proj
#lapply(toInstall, library, character.only = TRUE)

CONSUMER_KEY <- "imIl8tDEB6zvqAbJBqn84tU4M"
CONSUMER_SECRET <- "YycY5q4caygmOBIxERlRj9XpRPfJLCuVzUXAFQ5axHf8bKJz5I"
ACCESS_TOKEN <- "828711892495507460-TuPnRVkyCiJdYFfRWU9pUk21ktjgqMr"
ACCESS_TOKEN_SECRET <- "iDTP5bOyluzyTpYS20F39vYQ0UXtQwnztvzWEzTe5k7Ja"
setup_twitter_oauth(CONSUMER_KEY, CONSUMER_SECRET, ACCESS_TOKEN, ACCESS_T

#--------------------------------------------------
#1. Convert search result tweets into dataframe

searchTerm <- "#Disney"
#searchResult <- searchTwitter(searchTerm,n=20) #collect 20 #Disney
searchResult <- searchTwitter(searchTerm,n=20,geocode='42,-78,10000mi')#a

#Set locale to system default UTF-8
Sys.setlocale(category="LC_ALL", locale="")
tweetFrame <-twListToDF(searchResult)#str in DF named tweetFrame
```

```
#search result written at .csv, "replace with "~/Desktop/fileName.csv" fc
write.csv(tweetFrame,file= "tweetFrame.csv")

#---------------------------------------------------
```

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

'en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8'

In [16]:

```
#---------------------------------------------------
#2. Lookup screenName from tweetFrame
#3. From screenNames get user info and store to userinfoFrame

userInfo <- lookupUsers(tweetFrame$screenName)
userFrame <- twListToDF(userInfo)
#print(userFrame)
write.csv(userFrame,file= "userFrame.csv")
#---------------------------------------------------
```

In [ ]:

```
#--------------------------------------------------
#need modification
#https://www.r-bloggers.com/gathering-twitter-data-with-the-twitter2mongc

#4. Keep only users with location info
locatedUsers <- !is.na(userFrame$location)
write.csv(locatedUsers,file= "locatedUsersFrame.csv")

#locations <- geocode(userFrame$location[locatedUsers])

locations <- geocode(userFrame$location[locatedUsers])
write.csv(locations,file= "locationsFrame.csv")#lon,lat store here


with(locations, plot(lon, lat))#plot in x-y plane

#--------------------------------------------------
# Set up the map
#left <- min(geocodes$lon)
#bottom <- min(geocodes$lat)
#right <- max(geocodes$lon)
#top <- max(geocodes$lat)
#map <- get_map(location = c(left,bottom,right,top))

#not working skip for now
#locations<-geocode(userFrame$location[locatedUsers])  # Use amazing API
#locations<-geocode(locatedUsers)  # Use amazing API to guess

#Keep only users with location info
#Get the geo code of the locations from this dataframe
#group tweets ie many ways,

#collect tweets from thes people by using lookupUsers
#group by location, 26-54 log, group each long and each lat, take potions
#2) get city center for each distance, search tweets around the region, a
#to fall of log,and lat in 1 cat
```

In [ ]:

```
#--------------------------------------------------
#5 .Get the geo code of the locations from this dataframe
#6. Hints on TwitteR functions you may need: twListToDF, lookupUsers, geo




#--------------------------------------------------
# approximate lat/lon from textual location data.
##with(locations, plot(lon, lat))

#geocode
#plotting to map, find beans:=group eg 10, count tweets how many fall,
#plug it big/small,for loop in R,takes syntax, count dataFrame by taking
#cmd dim dataFrame. scanning process to code,
```

In [ ]:

```
worldMap <- map_data("world")  # Easiest way to grab a world map shapefil
zp1 <- ggplot(worldMap)
zp1 <- zp1 + geom_path(aes(x = long, y = lat, group = group),  # Draw map
                       colour = gray(2/3), lwd = 1/3)

zp1 <- zp1 + geom_point(data = locations,  # Add points indicating users
                        aes(x = lon, y = lat),
                        colour = "RED", alpha = 1/2, size = 1)
zp1 <- zp1 + coord_equal()  # Better projections are left for a future po
zp1 <- zp1 + theme_minimal()  # Drop background annotations
print(zp1)
```

In [ ]:

```
#--------------------------------------------------
#Summarizing trending topics about a location (place)
#When we are visiting places (say, for an interview or
#other official visits) you may want to about topics
#trending in that place. Instead of reading newspapers
#and online news, you want just a quick summary. You want
#to put use your twitter “data client” application development
#experience. You use the twitteR libraries “trends” function to
#retrieve 10 top things trending about the place and summarize
#it appropriately as a complete message (print out).

#Input: Location specified either as geo-location or by
#its name Output: A message listing the topics trending
#about the place. (Day 6)







#--------------------------------------------------
```

In [ ]:

```
#https://blog.dominodatalab.com/geographic-visualization-with-rs-ggmaps/
install.packages("ggmap")
library(ggmap)
```

In [ ]:

```
qmap(location = "boston university")
qmap(location = "boston university", zoom = 14)
qmap(location = "boston university", zoom = 14, source = "osm")
```

In [ ]:

```
mydata = read.csv("disneyTest.csv")
```

In [ ]:

```
mydata$screenName1 <- as.character(mydata$screenName)
```

In [ ]:

```
#mydata$MV.Number = as.numeric(mydata$MV.Number)
mydata = mydata[mydata$State != "Alaska", ]
mydata = mydata[mydata$State != "Hawaii", ]
```

In [ ]:
```
#for (i in 1:nrow(mydata)) {
#  latlon = geocode(mydata[i,1])
#  mydata$lon[i] = as.numeric(latlon[1])
#  mydata$lat[i] = as.numeric(latlon[2])
#}
```

In [ ]:
```
usa_center = as.numeric(geocode("United States"))
```

In [ ]:
```
USAMap = ggmap(get_googlemap(center=usa_center, scale=2, zoom=4), extent=
```

In [ ]:
```
USAMap + geom_point(aes(x=lon, y=lat), data=mv_num_collisions, col="orang
```

In [ ]:
```
worldMap <- map_data("world")  # Easiest way to grab a world map shapefil
```

In [ ]:
```
zp1 <- ggplot(worldMap)
```

In [ ]:
```
zp1 <- zp1 + geom_path(aes(x = long, y = lat, group = group),  # Draw map
                       colour = gray(2/3), lwd = 1/3)
```

In [ ]:
```
zp1 <- zp1 + geom_point(data = locations,  # Add points indicating users
                        aes(x = lon, y = lat),
                        colour = "RED", alpha = 1/2, size = 1)
```

In [ ]:
```
zp1 <- zp1 + coord_equal()  # Better projections are left for a future po
```

In [ ]:
```
zp1 <- zp1 + theme_minimal()  # Drop background annotations
```

In [ ]:
```
print(zp1)
#-----
```

In [ ]:
```
write.csv(tweetFrame,file= "disneyTest.csv")#write datas from tweetFrame
#write.csv(tweetFrame,file= "~/Desktop/disneyTest.csv")#abs path
print(tweetFrame)#print and see

#eg: read.csv(file, header = TRUE, sep = ",", quote = "\"",dec = ".", fil
#read.csv (searchResult,file= "~/Desktop/disney.csv)
```

In [ ]:
```
data1 <-read.csv("disneyTest.csv")#read from disneyTest.csv-default path,
#head(data1)#display below doesn't work for now
```

In [ ]:

```
summary(data1)#check to see
```

In [ ]:

In [ ]:

```
#Hints on TwitteR functions you may need: twListToDF, lookupUsers, geocod
#Lookup screen name from this dataframe
#Create a new variable, screenNameVar, that categorizes users as with loc


#From Screen names get user info and convert into dataframe
#to do this part, need vecotr: store screen name with log,and lat
#need to collect 20k tweets will find <10k with log,and lat


#Keep only users with location info


#Get the geo code of the locations from this dataframe

#data1$longitudeCat <-cut(data1$longitude,c(-180,180))#might not need to
#summary(data1)

#data1$latitudeCat <-cut(data1$latitude,c(-90,90))#might not need to grou
#summary(data1)

#install.packages("ggmap")

#install.packages("maps")

#install.packages("maptool")

#tweetFrame

#lookupUsers(): pass users with screenname form data frame
#users <-lookupUsers(data1$ScreenName)
#usersFrame <-twListToDF(users)

#http://stackoverflow.com/questions/40721031/twitter-package-how-to-get-u
#to store users with screenName in vector
#userWithScreenName <- c()
```