In [1]:
```
###################################
#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" fo
write.csv(tweetFrame,file= "tweetFrame.csv")

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

```
Attaching package: ‘plyr’

The following object is masked from ‘package:maps’:

    ozone

The following object is masked from ‘package:twitteR’:

    id

Loading required package: raster
Loading required package: sp

Attaching package: ‘dismo’

The following object is masked from ‘package:ggmap’:

    geocode


[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 [2]:
```
#-------------------------------------------------
#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-twitter2mongo

#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 [3]:
```
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)
```

```
Error in fortify(data): object 'locations' not found
Traceback:

1. geom_point(data = locations, aes(x = lon, y = lat), colour = "RED",
 .     alpha = 1/2, size = 1)
2. layer(data = data, mapping = mapping, stat = stat, geom = GeomPoint
,
 .     position = position, show.legend = show.legend, inherit.aes = i
nherit.aes,
 .     params = list(na.rm = na.rm, ...))
3. fortify(data)
```

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 [4]:
```
#https://blog.dominodatalab.com/geographic-visualization-with-rs-ggmaps/
install.packages("ggmap")
library(ggmap)
```

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

In [6]:
```
qmap(location = "universit at buffalo")
qmap(location = "universit at buffalo", zoom = 14)
qmap(location = "universit at buffalo", zoom = 14, source = "osm")
```

```
Map from URL : http://maps.googleapis.com/maps/api/staticmap?center=u
niversit+at+buffalo&zoom=10&size=640x640&scale=2&maptype=terrain&lang
uage=en-EN&sensor=false
(http://maps.googleapis.com/maps/api/staticmap?center=universit+at+bu
ffalo&zoom=10&size=640x640&scale=2&maptype=terrain&language=en-EN&sen
sor=false)
Information from URL : http://maps.googleapis.com/maps/api/geocode/js
on?address=universit%20at%20buffalo&sensor=false
(http://maps.googleapis.com/maps/api/geocode/json?address=universit%2
0at%20buffalo&sensor=false)
Warning message:
“`panel.margin` is deprecated. Please use `panel.spacing` property in
stead”

Map from URL : http://maps.googleapis.com/maps/api/staticmap?center=u
niversit+at+buffalo&zoom=14&size=640x640&scale=2&maptype=terrain&lang
uage=en-EN&sensor=false
(http://maps.googleapis.com/maps/api/staticmap?center=universit+at+bu
ffalo&zoom=14&size=640x640&scale=2&maptype=terrain&language=en-EN&sen
sor=false)
```

In [ ]:
```
mydata = read.csv("disneyTest.csv")
```

In [8]:
```
mydata$screenName1 <- as.character(mydata$screenName)
```

```
Error in eval(expr, envir, enclos): object 'mydata' not found
Traceback:
```

In [7]:
```
#mydata$MV.Number = as.numeric(mydata$MV.Number)
mydata = mydata[mydata$State != "Alaska", ]
mydata = mydata[mydata$State != "Hawaii", ]
```

```
Error in eval(expr, envir, enclos): object 'mydata' not found
Traceback:
```

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 [9]:
```
usa_center = as.numeric(geocode("United States"))
```

```
Loading required namespace: XML

Error in .geocode(xx$place, oneRecord = oneRecord, extent = extent, pr
ogress = progress): You need to install the XML package to be able use
this function
Traceback:

1. geocode("United States")
2. .geocode(xx$place, oneRecord = oneRecord, extent = extent, progress
= progress)
3. stop("You need to install the XML package to be able use this funct
ion")
```

In [10]:
```
USAMap = ggmap(get_googlemap(center=usa_center, scale=2, zoom=4), extent=
```

```
Error in get_googlemap(center = usa_center, scale = 2, zoom = 4): obje
ct 'usa_center' not found
Traceback:

1. ggmap(get_googlemap(center = usa_center, scale = 2, zoom = 4),
.     extent = "normal")
2. get_googlemap(center = usa_center, scale = 2, zoom = 4)
```

In [11]:
```
USAMap + geom_point(aes(x=lon, y=lat), data=mv_num_collisions, col="orang
```

```
Error in eval(expr, envir, enclos): object 'USAMap' not found
Traceback:
```

In [12]:
```
worldMap <- map_data("world")  # Easiest way to grab a world map shapefil
```

In [13]:
```
zp1 <- ggplot(worldMap)
```

In [14]:
```
zp1 <- zp1 + geom_path(aes(x = long, y = lat, group = group),  # Draw map
                       colour = gray(2/3), lwd = 1/3)
```

In [15]:
```
zp1 <- zp1 + geom_point(data = locations,  # Add points indicating users
                        aes(x = lon, y = lat),
                        colour = "RED", alpha = 1/2, size = 1)
```

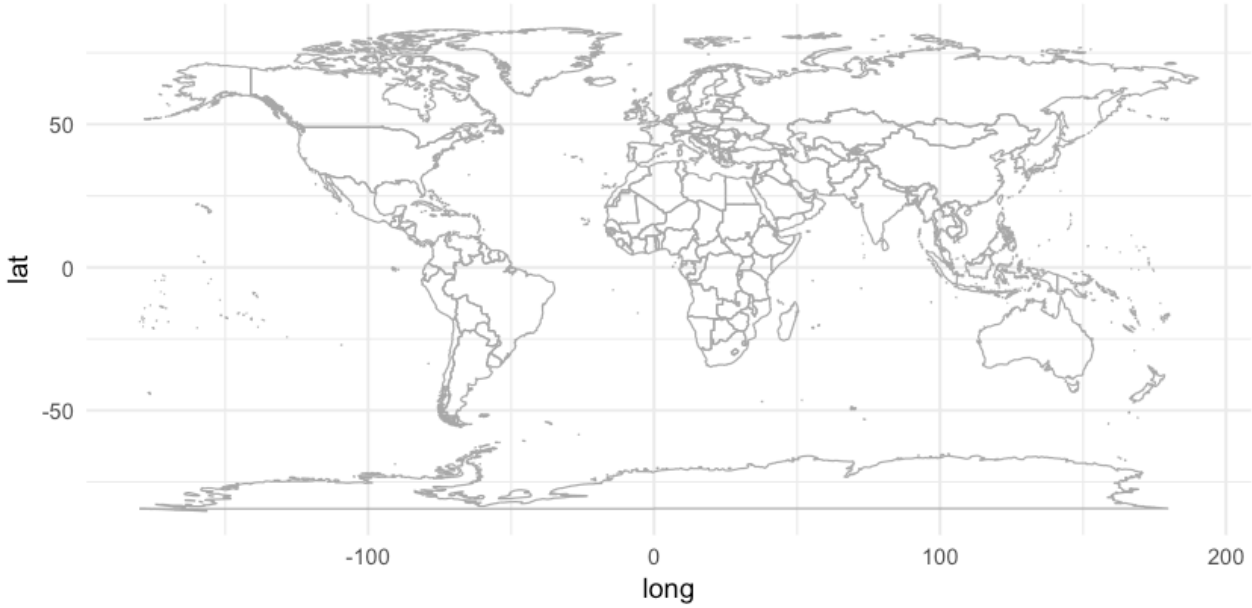
```
Error in fortify(data): object 'locations' not found
Traceback:

1. geom_point(data = locations, aes(x = lon, y = lat), colour = "RED",
 .     alpha = 1/2, size = 1)
2. layer(data = data, mapping = mapping, stat = stat, geom = GeomPoint
,
 .     position = position, show.legend = show.legend, inherit.aes = i
nherit.aes,
 .     params = list(na.rm = na.rm, ...))
3. fortify(data)
```

In [16]:
```
zp1 <- zp1 + coord_equal()  # Better projections are left for a future po
```

In [17]:
```
zp1 <- zp1 + theme_minimal()  # Drop background annotations
```

In [18]:
```
print(zp1)
#-----
```



In [19]:
```
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)
```

```
                                                                    text
1                                                                Nothing makes
me happier then seeing musical theatre! #newsiesforever #NewsiesMovie
#disney
2  RT @KaliMarcum: Disney Toys &amp;amp; Baby Lovies at Aimee's Comfy Cor
ner #eBay #Disney #Toys #ShoppingQueen #Baby #Toddlers #GiftIdeas htt
ps://t.c… (https://t.c…)
3               Check out Kohls Cares Simba The Lion King Cub 11" Plu
sh Disney Movie Stuffed Animal 2014  #Disney https://t.co/NRf982wuZs
(https://t.co/NRf982wuZs) via @eBay
4      Disney Toys &amp;amp; Baby Lovies at Aimee's Comfy Corner #eBay #D
isney #Toys #ShoppingQueen #Baby #Toddlers #GiftIdeas… https://t.co/r
AZUVJpkNY (https://t.co/rAZUVJpkNY)
5                                                    #EPSfoam creates
some #theater magic!! https://t.co/Orah7CfZaf
(https://t.co/Orah7CfZaf) #Disney https://t.co/ZXACXpWMyU
(https://t.co/ZXACXpWMyU)
6      RT @DouglasMacKrell: In 1995 #Disney released a TV specal with
```

In [20]:
```
data1 <-read.csv("disneyTest.csv")#read from disneyTest.csv-default path,
#head(data1)#display below doesn't work for now
```

In [21]:
```
summary(data1)#check to see
```

```
       X
 Min.   : 1.00
 1st Qu.: 5.75
 Median :10.50
 Mean   :10.50
 3rd Qu.:15.25
 Max.   :20.00
```

```
                                                                    text
 RT @mjsmith0523: Going back means I'll have to face my past. I've bee
n running from it for so long. —Simba (The Lion King) #Disney #disneyq
…: 3
 @ClayTravis did you go to Country Bears to see your kinfolk yet, you
dumb hillbilly? #disney
```

```
: 1
 #EPSfoam creates some #theater magic!! https://t.co/Orah7CfZaf
(https://t.co/Orah7CfZaf) #Disney https://t.co/ZXACXpWMyU
(https://t.co/ZXACXpWMyU)
: 1
 Check out Kohls Cares Simba The Lion King Cub 11" Plush Disney Movie
Stuffed Animal 2014  #Disney https://t.co/NRf982wuZs
(https://t.co/NRf982wuZs) via @eBay         : 1
 Disney Toys &amp;amp; Baby Lovies at Aimee's Comfy Corner #eBay #Disney #
Toys #ShoppingQueen #Baby #Toddlers #GiftIdeas… https://t.co/rAZUVJpkN
Y: (https://t.co/rAZUVJpkNY:) 1
 Enter to win a #Disneyland Cars Land Prize pack! #disney #contests\nh
ttps://t.co/34lA1GKqJA https://t.co/gNmLobvciP
(https://t.co/gNmLobvciP)                          : 1
 (Other)
:12
 favorited       favoriteCount       replyToSN                 created
 Mode :logical   Min.   :0.00   ClayTravis: 1   2017-02-18 20:42:21: 2
 FALSE:20        1st Qu.:0.00   NA's      :19   2017-02-18 20:27:56: 1
 NA's :0         Median :0.00                   2017-02-18 20:28:19: 1
                 Mean   :0.05                   2017-02-18 20:33:26: 1
                 3rd Qu.:0.00                   2017-02-18 20:34:17: 1
                 Max.   :1.00                   2017-02-18 20:34:52: 1
                                                (Other)            :13
 truncated         replyToSID              id              replyToUID
 Mode :logical   Min.   :8.33e+17   Min.   :8.331e+17   Min.   :507729
18
 FALSE:19        1st Qu.:8.33e+17   1st Qu.:8.331e+17   1st Qu.:507729
18
 TRUE :1         Median :8.33e+17   Median :8.331e+17   Median :507729
18
 NA's :0         Mean   :8.33e+17   Mean   :8.331e+17   Mean   :507729
18
                 3rd Qu.:8.33e+17   3rd Qu.:8.331e+17   3rd Qu.:507729
18
                 Max.   :8.33e+17   Max.   :8.331e+17   Max.   :507729
18
                 NA's   :19                             NA's   :19

statusSource
 <a href="http://twitter.com/download/android" rel="nofollow">Twitter
for Android</a>:7
 <a href="http://twitter.com" rel="nofollow">Twitter Web Client</a>
:5
 <a href="http://twitter.com/download/iphone" rel="nofollow">Twitter f
or iPhone</a>  :2
 <a href="http://instagram.com" rel="nofollow">Instagram</a>
:1
 <a href="http://paper.li" rel="nofollow">Paper.li</a>
:1
```

```
 <a href="http://sproutsocial.com" rel="nofollow">Sprout Social</a>
:1
 (Other)
:3
          screenName  retweetCount   isRetweet       retweeted
 srimes1cor    : 2   Min.   : 0.00   Mode :logical   Mode :logical
 3rdrockhome   : 1   1st Qu.: 0.00   FALSE:12        FALSE:20
 aaron3000     : 1   Median : 1.00   TRUE :8         NA's :0
 ChicagoToyShop: 1   Mean   : 1.80   NA's :0
 DebWein_Morgan: 1   3rd Qu.: 2.25
 DestiniLozano : 1   Max.   :14.00
 (Other)       :13
 longitude      latitude
 Mode:logical   Mode:logical
 NA's:20        NA's:20
```

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()
```