Geolocation.R

install.packages("twitteR")

install.packages("RCurl")

install.packages("base64enc")

install.packages("devtools")

install.packages("tm")

install.packages("googleVis")

install.packages("streamR")

install.packages("RJSONIO")

install.packages("stringr")

install.packages("ROAuth")

install.packages("googleVis")

install.packages("rworldmap")

install.packages("rworldxtra")

library(rworldmap)

library(rworldxtra)

library(streamR)

library(RCurl)

library(RJSONIO)

library(stringr)

library(ROAuth)

library(streamR)

library(googleVis)

tweets\_trump.df <- parseTweets("tweets\_trump.json", simplify = FALSE)

tweets.df <- parseTweets("tweets.json", simplify = FALSE)

combined.df <- rbind(tweets.df, tweets\_trump.df)

newdf.df <- combined.df[tweets.df$place\_lat != "NaN",]

keeps <- c("place\_lat", "place\_lon")

latlondata.df <- newdf.df[keeps]

newmap <- getMap(resolution = "low")

plot(newmap)

points(latlondata.df$place\_lon, latlondata.df$place\_lat, col = "#ff6666", cex = 1.3)

createdat\_vs\_retweet.R

install.packages("ggplot2")

require(ggplot2)

theme\_set(theme\_bw())

ggplot(aes(x=combined.df$created\_at, y=combined.df$retweet\_count), data= combined.df, breaks=20) + geom\_point()+ ylab("Number of retweets") + xlab("Time")

abline(h=45)

loldf <- tweets.df[tweets.df$retweet\_count != "0",]

Sentiment analysis code

#connect all libraries

library(twitteR)

library(ROAuth)

library(plyr)

library(dplyr)

library(stringr)

library(ggplot2)

#connect to API

download.file(url='http://curl.haxx.se/ca/cacert.pem', destfile='cacert.pem')

reqURL <- 'https://api.twitter.com/oauth/request\_token'

accessURL <- 'https://api.twitter.com/oauth/access\_token'

authURL <- 'https://api.twitter.com/oauth/authorize'

consumerKey <- 'rI1mXmZRL0rETGBK12yaIijjt' #put the Consumer Key from Twitter Application

consumerSecret <- 'cqPQFDqyTnwY1mnW7tXgbMTUrdUP1O9gr8nkAHfH8JYxwgK4BG'  #put the Consumer Secret from Twitter Application

Cred <- OAuthFactory$new(consumerKey=consumerKey,

                        consumerSecret=consumerSecret,

                        requestURL=reqURL,

                        accessURL=accessURL,

                        authURL=authURL)

Cred$handshake(cainfo = system.file('CurlSSL', 'cacert.pem', package = 'RCurl')) #There is URL in Console. You need to go to, get code and enter it in Console

save(Cred, file='twitter authentication.Rdata')

load('twitter authentication.Rdata') #Once you launched the code first time, you can start from this line in the future (libraries should be connected)

setup\_twitter\_oauth(consumer\_key=consumerKey,  consumer\_secret=consumerSecret, access\_token=NULL , access\_secret=NULL)

#the function for extracting and analyzing tweets

search <- function(searchterm)

{

 #extact tweets and create storage file

 list <- searchTwitter(searchterm, n=15, lang="en")

 df <- twListToDF(list)

 df <- df[, order(names(df))]

 df$created <- strftime(df$created, '%Y-%m-%d')

 if (file.exists(paste(searchterm, '\_stack.csv'))==FALSE) write.csv(df, file=paste(searchterm, '\_stack.csv'), row.names=F)

 #merge the last extraction with storage file and remove duplicates

 stack <- read.csv(file=paste(searchterm, '\_stack.csv'))

 stack <- rbind(stack, df)

 stack <- subset(stack, !duplicated(stack$text))

 write.csv(stack, file=paste(searchterm, '\_stack.csv'), row.names=F)

 #tweets evaluation function

 score.sentiment <- function(sentences, pos.words, neg.words, .progress='none')

 {

   require(plyr)

   require(stringr)

   scores <- laply(sentences, function(sentence, pos.words, neg.words){

     sentence <- iconv(sentence, 'UTF-8', 'ASCII')

     sentence <- gsub('[[:punct:]]', "", sentence)

     sentence <- gsub('[[:cntrl:]]', "", sentence)

     sentence <- gsub('\\d+', "", sentence)

     sentence <- tolower(sentence)

     word.list <- str\_split(sentence, '\\s+')

     words <- unlist(word.list)

     pos.matches <- match(words, pos.words)

     neg.matches <- match(words, neg.words)

     pos.matches <- !is.na(pos.matches)

     neg.matches <- !is.na(neg.matches)

     score <- sum(pos.matches) - sum(neg.matches)

     return(score)

   }, pos.words, neg.words, .progress=.progress)

   scores.df <- data.frame(score=scores, text=sentences)

   return(scores.df)

 }

 pos <- scan('C:\\Users\\Akshay Gvs\\Desktop\\positive-words.txt', what='character', comment.char=';') #folder with positive dictionary

 neg <- scan('C:\\Users\\Akshay Gvs\\Desktop\\negative-words.txt', what='character', comment.char=';') #folder with negative dictionary

 pos.words <- c(pos, 'upgrade')

 neg.words <- c(neg, 'wtf', 'wait', 'waiting', 'epicfail')

 Dataset <- stack

 Dataset$text <- as.factor(Dataset$text)

 scores <- score.sentiment(Dataset$text, pos.words, neg.words, .progress='text')

 write.csv(scores, file=paste(searchterm, '\_scores.csv'), row.names=TRUE) #save evaluation results

 #total score calculation: positive / negative / neutral

 stat <- scores

 stat$created <- stack$created

 stat$created <- as.Date(stat$created)

 stat <- mutate(stat, tweet=ifelse(stat$score > 0, 'positive', ifelse(stat$score < 0, 'negative', 'neutral')))

 by.tweet <- group\_by(stat, tweet, created)

 by.tweet <- summarise(by.tweet, number=n())

 write.csv(by.tweet, file=paste(searchterm, '\_opin.csv'), row.names=TRUE)

 #chart

 ggplot(by.tweet, aes(created, number)) + geom\_line(aes(group=tweet, color=tweet), size=2) +

   geom\_point(aes(group=tweet, color=tweet), size=4) +

   theme(text = element\_text(size=18), axis.text.x = element\_text(angle=90, vjust=1)) +

   #stat\_summary(fun.y = 'sum', fun.ymin='sum', fun.ymax='sum', colour = 'yellow', size=2, geom = 'line') +

   ggtitle(searchterm)

 ggsave(file=paste(searchterm, '\_plot.jpeg'))

}

search("Donald Trump")

Streamingapi.R

install.packages("twitteR")

install.packages("RCurl")

install.packages("base64enc")

install.packages("devtools")

install.packages("tm")

install.packages("googleVis")

install.packages("streamR")

install.packages("RJSONIO")

install.packages("stringr")

install.packages("ROAuth")

install.packages("googleVis")

install.packages("rworldmap")

install.packages("rworldxtra")

library(rworldmap)

library(rworldxtra)

library(streamR)

library(RCurl)

library(RJSONIO)

library(stringr)

library(ROAuth)

library(streamR)

library(googleVis)

requestURL <- "https://api.twitter.com/oauth/request\_token"

accessURL <- "https://api.twitter.com/oauth/access\_token"

authURL <- "https://api.twitter.com/oauth/authorize"

consumerKey <- "rI1mXmZRL0rETGBK12yaIijjt"

consumerSecret <- "cqPQFDqyTnwY1mnW7tXgbMTUrdUP1O9gr8nkAHfH8JYxwgK4BG"

#make sure these show up in the global env

my\_oauth <- OAuthFactory$new(consumerKey = consumerKey, consumerSecret = consumerSecret, requestURL = requestURL, accessURL = accessURL, authURL = authURL)

#environment set up !

download.file(url="http://curl.haxx.se/ca/cacert.pem", destfile="cacert.pem")

#download cacert.pem file

my\_oauth$handshake(cainfo = system.file("CurlSSL", "cacert.pem", package = "RCurl"))

# end of part 1

#part 2 begins

save(my\_oauth, file = "my\_oauth.Rdata")

load("my\_oauth.RData")

for(x in 1:20)

{

 x= x+1

 filterStream(file.name = "tweets\_trump.json", track = c("Donald Trump"),language = "en", timeout = 10, oauth = my\_oauth)

 tweets\_trump.df <- parseTweets("tweets\_trump.json", simplify = FALSE)

 if(x==16) break;

}

tweets.df <- parseTweets("tweets.json", simplify = FALSE)

combined.df <- rbind(tweets.df, tweets\_trump.df)

newdf.df <- combined.df[tweets.df$place\_lat != "NaN",]

keeps <- c("place\_lat", "place\_lon")

latlondata.df <- newdf.df[keeps]

#save all of the tweets in a data frame for further processing

tweets.df[[37]]

L = tweets.df$place\_lat !=0

L

tweets.df[L,]

dim(tweets.df)

subset(tweets.df,COLUMNNAME=="created\_at")

newmap <- getMap(resolution = "low")

plot(newmap)

points(latlondata.df$place\_lon, latlondata.df$place\_lat, col = "#ff6666", cex = 1.3)

#next part : run the code for the sentiment analysis .

#Then try running it on the induvidual data frames and see what happens

tweets1.df <- parseTweets("tweets1.json", simplify = TRUE)

tweets.df$created\_at <- as.POSIXct(tweets.df$created\_at, format="%a, %d %b %Y %H:%M:%S %z")

df <- df[order(df$date),]

Wordcloud2.R

install.packages("tm")

install.packages("stringi")

library(stringi)

library(twitteR)

library(tm)

library(wordcloud)

library(RColorBrewer)

require(twitteR)

library(twitteR)

library(tm)

library(wordcloud)

library(RColorBrewer)

library(rworldmap)

library(rworldxtra)

library(streamR)

library(RCurl)

library(RJSONIO)

library(stringr)

library(ROAuth)

library(streamR)

library(googleVis)

#connect to API

download.file(url='http://curl.haxx.se/ca/cacert.pem', destfile='cacert.pem')

reqURL <- 'https://api.twitter.com/oauth/request\_token'

accessURL <- 'https://api.twitter.com/oauth/access\_token'

authURL <- 'https://api.twitter.com/oauth/authorize'

consumer\_key <- 'rI1mXmZRL0rETGBK12yaIijjt' #put the Consumer Key from Twitter Application

consumer\_secret <- 'cqPQFDqyTnwY1mnW7tXgbMTUrdUP1O9gr8nkAHfH8JYxwgK4BG'  #put the Consumer Secret from Twitter Application

access\_token <- '296656578-OkYiWH7aTI2pQLEfw4vuFvthZXmb0I0Usn0dF2y8'

access\_secret <- 'Fhba7QzqgzGqHK9jqgtpo1xZxngbKO1usv7JtFBGCisGt'

setup\_twitter\_oauth(consumer\_key , consumer\_secret , access\_token , access\_secret)

mach\_tweets = searchTwitter("Donald Trump", n=500, lang="en")

mach\_text = sapply(mach\_tweets, function(x) x$getText())

mach\_text = iconv(mach\_text, 'UTF-8', 'ASCII')

# create a corpus

mach\_corpus = Corpus(VectorSource(mach\_text))

# create document term matrix applying some transformations

tdm = TermDocumentMatrix(mach\_corpus,

                        control = list(removePunctuation = TRUE,

                                       stopwords = c("Donald", "Trump","donald","trump", stopwords("english")),

                                       removeNumbers = TRUE, tolower = TRUE))

# define tdm as matrix

m = as.matrix(tdm)

# get word counts in decreasing order

word\_freqs = sort(rowSums(m), decreasing=TRUE)

# create a data frame with words and their frequencies

dm = data.frame(word=names(word\_freqs), freq=word\_freqs)

# plot wordcloud

wordcloud(dm$word, dm$freq, random.order=FALSE, colors=brewer.pal(8, "Dark2"))

# save the image in png format

png("MachineLearningCloud.png", width=12, height=8, units="in", res=300)

wordcloud(dm$word, dm$freq, random.order=FALSE, colors=brewer.pal(8, "Dark2"))

dev.off()

createddat\_vs\_fav.R

install.packages("ggplot2")

require(ggplot2)

theme\_set(theme\_bw())

ggplot(aes(x=combined.df$created\_at, y=combined.df$favourites\_count), data= combined.df) + geom\_point() + ylab("Number of favourites") + xlab("Date and time created at")

d <- density(combined.df$retweet\_count,combined.df$favourites\_count) # returns the density data

plot(d) # plots the results

install.packages("aplpack")

library(aplpack)

attach(combined.df)

bagplot(combined.df$retweet\_count,combined.df$favourites\_count, xlab="rt", ylab="fav",

       main="Bagplot, used to visualize the location, spread, skewness, and outliers of the data set")

Retweet\_favourite\_compare.R

install.packages("ggplot2")

require(ggplot2)

x=combined.df$favourites\_count

y=combined.df$retweet\_count

theme\_set(theme\_bw())

ggplot(aes(x, y), data= combined.df) + geom\_point() + ylab("Number of favourites") + xlab("Number of retweets")

coef(lm(y ~ x))

p+geom\_abline(intercept = 7563.487, slope = 0.08485780)

Friends\_vs\_retweet.R

install.packages("ggplot2")

require(ggplot2)

x=combined.df$friends\_count

y=combined.df$retweet\_count

theme\_set(theme\_bw())

ggplot(aes(x, y), data= combined.df) + geom\_point() + ylab("retweet count") + xlab("friends count")

friends\_vs\_favourite.R

install.packages("ggplot2")

require(ggplot2)

x=combined.df$friends\_count

y=combined.df$favourites\_count

theme\_set(theme\_bw())

ggplot(aes(x, y), data= combined.df) + geom\_point() + ylab("favourite count") + xlab("friends count")

follower\_vs\_retweet.R

install.packages("ggplot2")

require(ggplot2)

x=combined.df$followers\_count

y=combined.df$retweet\_count

theme\_set(theme\_bw())

ggplot(aes(x, y), data= combined.df) + geom\_point() + ylab("retweet count") + xlab("follower count")

follower\_vs\_favourite.R

install.packages("ggplot2")

require(ggplot2)

x=combined.df$followers\_count

y=combined.df$favourites\_count

theme\_set(theme\_bw())

ggplot(aes(x, y), data= combined.df) + geom\_point() + ylab("favourite count") + xlab("follower count")

status\_vs\_retweet.R

install.packages("ggplot2")

require(ggplot2)

x=combined.df$statuses\_count

y=combined.df$retweet\_count

theme\_set(theme\_bw())

ggplot(aes(x, y), data= combined.df) + geom\_point() + ylab("retweet count") + xlab("statuses count")

status\_vs\_favourite.R

install.packages("ggplot2")

require(ggplot2)

x=combined.df$statuses\_count

y=combined.df$favourites\_count

theme\_set(theme\_bw())

ggplot(aes(x, y), data= combined.df) + geom\_point() + ylab("favourite count") + xlab("statuses count")