# Install and Activate Packages

install.packages("twitteR")
install.packages("tm")
install.packages("RCurl")
install.packages("wordcloud")
install.packages("RColorBrewer")
install.packages("NLP")
install.packages("httr")
install.packages("sqldf")
install.packages("stringr")
install.packages("sentimentr")
install.packages("sentiment")
install.packages("RSentiment")
install.packages("RStem")
install.packages("ggplot2")
library(twitteR)
library(tm)
library(RCurl)
library(wordcloud)
library(RColorBrewer)
library(NLP)
library(httr)
library(sqldf)
library(stringr)
library(sentimentr)
library(sentiment)
library(RSentiment)
library(Rstem)
library(ggplot2)

# Here we are establishing our keys and tokens from the Twitter API
consumer_key = 'M37CCVpSyQ9bjZpI4fP7vEs3B'
consumer_secret = 'UlCEXDluSrQ9AUy69rZPsyzLjP4Sz0h18o0BVOKej9pg6TFtic'
access_token = '232194592-zQ1fHpsqphC7KPAdyyvutXLJx4liQ2hahLwo8YJm'
access_secret = 'DijvPVlYgZSgoSHc0m81j1MUfQ4GFpCS92nE221sMQMJD'

# Here we are creating the "handshake" with Twitter
setup_twitter_oauth(consumer_key= consumer_key, consumer_secret= consumer_secret, access_token= access_token, access_secret= access_secret)

# Once you have created your handshake, you can start searching for tweets
# Note that if you select a common term like "Atlanta" you will generate a lot of Tweets quickly
# But if you select an esoteric term like "heteroscedasticity", it might take a while to get any
tw <- searchTwitter("NBA Draft", n=1000, lang='en', resultType = "recent")
class(tw)

tw_text <- sapply(tw, function(x) x$getText())
str(tw_text)

# Here we are creating a corpus, which will allow us to format the text data for analysis
tw_corpus <- Corpus(VectorSource(tw_text))
tw_corpus
inspect(tw_corpus[2])

# Here we are cleaning the corpus
tw_clean <- tm_map(tw_corpus, removePunctuation)

tw_clean <- tm_map(tw_clean, content_transformer(tolower))

tw_clean <- tm_map(tw_clean, removeWords, stopwords("english"))

tw_clean <- tm_map(tw_clean, removeNumbers)

tw_clean <- tm_map(tw_clean, stripWhitespace)

tw_clean <- tm_map(tw_clean, removeWords, c("NBA", "nba", "draft"))

tw_clean

# Creating a word cloud
wordcloud(tw_clean, random.order = F, max.words=50, scale = c(2, 0.5), colors = brewer.pal(8, "Dark2"))

# Find the top terms from the tweets

dtm <- TermDocumentMatrix(tw_clean)

m <- as.matrix(dtm)

v <- sort(rowSums(m), decreasing=TRUE)

d <- data.frame(word = names(v), freq=v)

head(d,10)

str(d)

# Classifying the sentiment analysis of the tweets

tw_emo <- sentiment(tw_text)
```r
tw.df <- data.frame(tw_text)

test <- sentiment_by(tw_text)

extract <- extract_sentiment_terms(tw_text)

extract$sentence
extract$neutral
extract$positive
data.table::as.data.table(extract)

attributes(extract_sentiment_terms(extract))$counts
attributes(extract_sentiment_terms(extract))$elements

summary(tw_emo)

# Use the searchTwitter function to only get tweets within 50 miles of Atlanta
?searchTwitter

tweets_geolocated <- searchTwitter("NBA Draft", n=100, lang="en", geocode='33.7490, -84.3880, 20mi')
tweets_geolocated.df <- twListToDF(tweets_geolocated)
write.table(tweets_geolocated.df, "NBAtweets.csv", sep = ",", col.names = NA, row.names = TRUE)
```
# Get Tweets from specific people

c@tw< userTimeline('Cityofatlanta',n=25,includeRts = TRUE)

mentions ('Cityofatlanta', n=15) # get your tweets that were retweeted

favs <- favorites("Cityofatlanta", n =10) # tweets a user has favorited

#Determine where a tweeter is located

sarah <- getUser("Sarah_Hyland")
location(sarah)

#Determine how many followers a person has

ggetUser("sarah")$followersCount

#See who their friends are

user <- getUser("Sarah_Hyland")
friends <- user$getFriends()

###WHAT IS TRENDING WHERE?

#http://www.woeidlookup.com/

tw@n<-getTrends(woeid=2459115)
twny1<-sqldf('select name as NYNAMES from twny limit 20')
twny1

twla<-getTrends(woeid=2442047)
twla1<-sqldf('select name as LANAMES from twla limit 20')
twla1

twatl<-getTrends(woeid=2357024)
twatl1<-sqldf('select name as ATLNAMES from twatl limit 20')
twatl1

Trends<- cbind(twny1, twla1, twatl1)

Trends