```
# Sentiment Analysis of Social Media Using R
# Library to connect with twitter & pulling Tweets & converting into db

library(twitteR)
library(syuzhet)
library(tm)
library(SnowballC)
library(ROAuth)
library(ggplot2)
require(devtools)
library(sentiment)
library(wordcloud)

#Creating access token to connect with twitter rest api

consumer_key = "abcdefghijklmnopqrstuvwxyz"
consumer_secret = " abcdefghijklmnopqrstuvwxyz "
access_token = "10987654321abcdefghijklmnopqrstuvwxyz "
access_secret = "10987654321abcdefghijklmnopqrstuvwxyz "
setup_twitter_oauth(consumer_key, consumer_secret, access_token, access_secret)

tk = twitteR::searchTwitter('#MotoE5Plus', n = 10000, retryOnRateLimit = 1e3)
d = twitteR::twListToDF(tk)
n.tweet<-length(d)
View(d)
head(d$text)

#Pre-processing, removing hashtags, hyperlinks etc
buffer<-Corpus(VectorSource(d$text)) # Building a Corpus, specifying it to be a
charactervector

#Converting to Lower Case
buffer<-tm_map(buffer, content_transformer(tolower))

#Removing URLs
removeURL<- function(x) gsub("http[^[:space:]]*", "", x)
buffer<- tm_map(buffer, content_transformer(removeURL))

# remove anything other than English letters or space
removeNumPunct<- function(x) gsub("[^[:alpha:][:space:]]*", "", x)
buffer<- tm_map(buffer, content_transformer(removeNumPunct))

# removestopwords
myStopwords<- c(setdiff(stopwords('english'), c("r", "big")),
"use", "see", "used", "via", "amp")
buffer<- tm_map(buffer, removeWords, myStopwords)
```

```
# remove extra whitespace
buffer<- tm_map(buffer, stripWhitespace)

# keep a copy for stem completion later
bufferCopy<- buffer

# Stem & Stemming
buffer<- tm_map(buffer, stemDocument) # stem words
writeLines(strwrap(buffer[[190]]$content, 60))

stC2<- function(x, dictionary) {
x <- unlist(strsplit(as.character(x), " "))
x <- x[x != ""]
x <- stemCompletion(x, dictionary=dictionary)
x <- paste(x, sep="", collapse=" ")
PlainTextDocument(stripWhitespace(x))
}
buffer<- lapply(buffer, stC2, dictionary=bufferCopy)
buffer<- Corpus(VectorSource(buffer))
writeLines(strwrap(buffer[[190]]$content, 60))

#Countingthe Frequency of words repeating
wordFreq<- function(corpus, word) {
results<- lapply(corpus,
function(x) { grep(as.character(x), pattern=paste0("\\<",word)) }
)
sum(unlist(results))
}
n.miner<- wordFreq(bufferCopy, "miner")
n.mining<- wordFreq(bufferCopy, "mining")
cat(n.miner, n.mining)
replaceWord<- function(corpus, oldword, newword) {
tm_map(corpus, content_transformer(gsub),
pattern=oldword, replacement=newword)
}
buffer<- replaceWord(buffer, "miner", "mining")
buffer<- replaceWord(buffer, "universidad", "university")
buffer<- replaceWord(buffer, "scienc", "science")

##Build Term Document Matrix for building a matrix that describes the repeating words
tdm<- TermDocumentMatrix(buffer,
control = list(wordLengths = c(1, Inf)))
tdm
idx<- which(dimnames(tdm)$Terms %in% c("r", "data", "mining"))
as.matrix(tdm[idx, 21:30])
```

```
# Inspecting frequent words
(freq.terms<- findFreqTerms(tdm, lowfreq = 20))
term.freq<- rowSums(as.matrix(tdm))
term.freq<- subset(term.freq, term.freq>= 1000)
df<- data.frame(term = names(term.freq), freq = term.freq)

#Plotting the data
ggplot(df, aes(x=term, y=freq)) + geom_bar(stat="identity") +
xlab("Terms") + ylab("Count") + coord_flip() +
theme(axis.text=element_text(size=7))

#Sentiment Analysis
sentiments<- sentiment(d$text)
table(sentiments$polarity)

# Timeline Analysis
library(topicmodels)
dtm <- as.DocumentTermMatrix(tdm)
lda <- LDA(dtm, k = 8)
term <- terms(lda, 7)
(term <- apply(term, MARGIN = 2, paste, collapse = ", "))
topics <- topics(lda)
topics <- data.frame(date=as.IDate(d$created), topic=topics)
ggplot(topics, aes(date, fill = term[topic])) +
 geom_density(position = "stack")


#WordCloud
m <- as.matrix(tdm)
word.freq<- sort(rowSums(m), decreasing = T)
pal<- brewer.pal(9, "BuGn")[-(1:4)]
wordcloud(words = names(word.freq), freq = word.freq, min.freq = 3,
random.order = F, colors = pal)
```