library(twitteR)

library(ROAuth)

library(SnowballC)

library(ggplot2)

library(RColorBrewer)

library(httr)

library(RCurl)

library(XML)

library(devtools)

library(sqldf)

#download the next

library(BiocGenerics)

library(graph)

library(Rgraphviz)

library(Rstem)

library(slam)

library(tm) #CRAN

library(modeltools) #CRAN

library(sentiment)

library(wordcloud)

#Download gsl dependency first

#sudo yum install gsl-devel

library(topicmodels)

library(shiny)

devtools::install\_github("hadley/tidyr")

devtools::install\_github("timelyportfolio/sunburstR")

library(sunburstR)

library(stringr)

library(tm)

myCorpus <- Corpus(VectorSource(tweets.df$text))

dataframe <- data.frame(text=sapply(myCorpus, identity), stringsAsFactors=F)

removeURL <- function(x) gsub("http[^[:space:]]\*", "", x)

myCorpus <- tm\_map(myCorpus, content\_transformer(removeURL))

####

removeNumPunct <- function(x) gsub("[[:punct:]]\*", "", x)

myCorpus <- tm\_map(myCorpus, content\_transformer(removeNumPunct))

dataframe1 <- data.frame(text=sapply(myCorpus, identity), stringsAsFactors=F)

usableText= function(x) str\_replace\_all(x,"[^[:graph:]]", " ")

myCorpus <- tm\_map(myCorpus, content\_transformer(usableText))

####

myCorpus <- tm\_map(myCorpus, content\_transformer(tolower))

myStopwords <- c(setdiff(stopwords('english'), c("a", "n")),"use","rt", "see", "a", "used", "in","is", "by", "for")

myCorpus <- tm\_map(myCorpus, removeWords, myStopwords)

myCorpus <- tm\_map(myCorpus, stripWhitespace)

myCorpusCopy <- myCorpus

dataframe1 <- data.frame(text=sapply(myCorpus, identity), stringsAsFactors=F)

#################################################################################################################

myCorpus <- tm\_map(myCorpus, stemDocument) # stem words

wordFreq <- function(corpus, word) {

results <- lapply(corpus,

function(x) { grep(as.character(x), pattern=paste0("\\<",word)) }

)

sum(unlist(results))

}

n.miner <- wordFreq(myCorpusCopy, "jaguar")

n.mining <- wordFreq(myCorpusCopy, "land rover")

cat(n.miner, n.mining)

for (str in tweets.df$created\_at)

{

str <- toString(str)

str <- strptime(str, "%a %b %d %H:%M:%S %z %Y", tz = "GMT")

dt.gmt <- as.POSIXct(str, tz = "GMT")

z <- format(dt.gmt, tz = "EST", usetz = TRUE)

date1=as.Date(z)

tweets.df$created\_at\_date <- date1

}

dataframe2 <- data.frame(text=sapply(myCorpus, identity), stringsAsFactors=F)

tdm <- TermDocumentMatrix(myCorpus, control = list(wordLengths = c(1, Inf)))

tdm

(freq.terms <- findFreqTerms(tdm, lowfreq = 3))

term.freq <- rowSums(as.matrix(tdm))

term.freq <- subset(term.freq, term.freq >= 3)

word\_freq <- data.frame(term = names(term.freq), freq = term.freq)

X <- word\_freq

p2 <- ggplot(X, aes(x = reorder(X$term, -(X$freq)), y = X$freq)) + geom\_bar(stat = "identity")

p2

m <- as.matrix(tdm)

# calculate the frequency of words and sort it by frequency

word.freq <- sort(rowSums(m), decreasing = T)

# colors

#RColorBrewer

pal <- brewer.pal(9, "BuGn")[-(1:4)]

wordcloud(words = names(word.freq), freq = word.freq, min.freq = 10, random.order = F, colors = pal)

findAssocs(tdm, "price", 0.2)

findAssocs(tdm, "service", 0.2)

dtm <- as.DocumentTermMatrix(tdm)

lda <- LDA(dtm, k = 2) # find 10 topics

term <- terms(lda, 4) # first 10 terms of every topic

(term <- apply(term, MARGIN = 2, paste, collapse = ", "))

term.df <- as.data.frame(term)

newtopic\_df <- as.data.frame(c(1:2))

names(newtopic\_df)[1] <- "topic"

newtopic\_df <- cbind(term.df,newtopic\_df)

topics <- topics(lda) # 1st topic identified for every document (tweet)

topics <- data.frame(date=as.Date(tweets.df$created\_at\_date), topic=topics)

ggplot(topics, aes(date, fill = term[topic])) + geom\_density(position = "stack")

tweet\_text <- as.data.frame(tweets.df$text)

polarity\_classified <- classify\_polarity(tweets.df$text, algorithm= "bayes", pstrong = 0.5, pweak= 1.0 , prior = 1.0, verbose = FALSE)

polarity\_classified\_df <- as.data.frame(polarity\_classified)

results\_polarity <- as.data.frame(polarity\_classified\_df$BEST\_FIT)

combined\_result <- cbind(tweet\_text,results\_polarity)

Visual\_DF <- as.data.frame(combined\_result)

head(Visual\_DF)

names(Visual\_DF)[1] <- "Tweet"

names(Visual\_DF)[2] <- "Sentiment"

Visual\_DF$title <- "JLR"

Visual\_DF$count <- rep(1, nrow(Visual\_DF))

combinedtopic\_terms <- sqldf("select \* from newtopic\_df INNER JOIN topics ON topics.topic = newtopic\_df.topic;")

combinedtopic\_terms[4] <- NULL

newVisual\_DF <- cbind(Visual\_DF, combinedtopic\_terms$term)

names(newVisual\_DF)[5] <- "Topics"

newVisual\_DF$path <- paste(newVisual\_DF$title, newVisual\_DF$Topic, newVisual\_DF$Sentiment, sep="-")

yellow <- c("#FFFF00")

blue <- c("#5687d1")

brown <- c("#7b615c")

orange <- c("#de783b")

l\_green <- c("#6ab975")

purple <- c("#a173d1")

l\_yellow <- c("#FFFFCC")

aqua <- c("#00CC99")

d\_aqua <- c("#009999")

occur <- c("#CC6600")

pink <- c("#CC33FF")

red <- c("#FF3300")

green <- c("#339900")

grey <-c("#999999")

#x= list(c9)

x=list(yellow, blue , orange , red , grey , green , aqua , d\_aqua , occur , purple , l\_green , brown , l\_yellow , pink)

typeof(x)

head(x)

sunburst(data.frame(xtabs(count~path,newVisual\_DF)), count = TRUE)

newVisual\_DF$Source <- tweets.df$url

newVisual\_DF$Created\_At\_Date <- tweets.df$created\_at\_date

newVisual\_DF$count <- NULL

View(newVisual\_DF)