install.packages("twitteR")

install.packages("ROAuth")

install.packages("tm")

install.packages("ggplot2")

install.packages("wordcloud")

install.packages("sentimentr")

install.packages("plyr")

install.packages("RTextTools")

install.packages("devtools")

require(devtools)

install.packages("e1071")

install\_github("sentiment140","okugami79")

install\_url("https://cran.r-project.org/src/contrib/Archive/Rstem/Rstem\_0.4-1.tar.gz")

install\_url("http://cran.r-project.org/src/contrib/Archive/sentiment/sentiment\_0.2.tar.gz")

library(e1071)

library(twitteR)

library(ROAuth)

library(tm)

library(ggplot2)

library(wordcloud)

library(sentimentr)

library(plyr)

library(RTextTools)

library(sentiment)

library(Rstem)

setup\_twitter\_oauth("PfQnk4Fui9vTQHntbBp59zN4S","Oc0RLUuW8a6F302raGx2yhGYl8sBhay9mBJCibyL92MJnzG5HM","613187277-mzFHgsW46UjS4tvIjr2iPttPcEZvMqmpwa4wN8BY","AfACYy6oLvBknuDwlKbqT84UwBSFVgYgdSgrGoz2gvAJd")

# harvest some tweets

some\_tweets = searchTwitter("mudik 2018", n=500, lang="en")

# get the text

some\_txt = sapply(some\_tweets, function(x) x$getText())

# remove retweet entities

some\_txt = gsub("(RT|via)((?:\\b\\W\*@\\w+)+)", "", some\_txt)

# remove at people

some\_txt = gsub("@\\w+", "", some\_txt)

# remove punctuation

some\_txt = gsub("[[:punct:]]", "", some\_txt)

# remove numbers

some\_txt = gsub("[[:digit:]]", "", some\_txt)

# remove html links

some\_txt = gsub("http\\w+", "", some\_txt)

# remove unnecessary spaces

some\_txt = gsub("[ \t]{2,}", "", some\_txt)

some\_txt = gsub("^\\s+|\\s+$", "", some\_txt)

# define "tolower error handling" function

try.error = function(x)

{

# create missing value

y = NA

# tryCatch error

try\_error = tryCatch(tolower(x), error=function(e) e)

# if not an error

if (!inherits(try\_error, "error"))

y = tolower(x)

# result

return(y)

}

# lower case using try.error with sapply

some\_txt = sapply(some\_txt, try.error)

# remove NAs in some\_txt

some\_txt = some\_txt[!is.na(some\_txt)]

names(some\_txt) = NULL

# classify emotion

#library(sentiment)

class\_emo = classify\_emotion(some\_txt, algorithm="bayes", prior=1.0)

# get emotion best fit

emotion = class\_emo[,7]

# substitute NA’s by "unknown"

emotion[is.na(emotion)] = "unknown"

# classify polarity

class\_pol = classify\_polarity(some\_txt, algorithm="bayes")

# get polarity best fit

polarity = class\_pol[,4]

# data frame with results

sent\_df = data.frame(text=some\_txt, emotion=emotion,

polarity=polarity, stringsAsFactors=FALSE)

# sort data frame

sent\_df = within(sent\_df,

emotion <- factor(emotion, levels=names(sort(table(emotion), decreasing=TRUE))))

# plot distribution of emotions

ggplot(sent\_df, aes(x=emotion)) +

geom\_bar(aes(y=..count.., fill=emotion)) +

scale\_fill\_brewer(palette="Dark2") +

labs(x="emotion categories", y="number of tweets")

######

ggplot(sent\_df, aes(x=polarity)) +

geom\_bar(aes(y=..count.., fill=polarity)) +

scale\_fill\_brewer(palette="RdGy") +

labs(x="polarity categories", y="number of tweets")

######

emos = levels(factor(sent\_df$emotion))

nemo = length(emos)

emo.docs = rep("", nemo)

for (i in 1:nemo)

{

tmp = some\_txt[emotion == emos[i]]

emo.docs[i] = paste(tmp, collapse=" ")

}

####

# remove stopwords

emo.docs = removeWords(emo.docs, stopwords("english"))

# create corpus

corpus = Corpus(VectorSource(emo.docs))

tdm = TermDocumentMatrix(corpus)

tdm = as.matrix(tdm)

colnames(tdm) = emos

# comparison word cloud

comparison.cloud(tdm, colors = brewer.pal(nemo, "Dark2"),

scale = c(3,.5), random.order = FALSE, title.size = 1.5)

