Sentiment Analysis

Oracy Martos

11/18/2018

Sentiment Analysis on Twitter

This project is integrant part of Big Data Analytics with R and Microsoft Azure of Data Scientist Formation. The goal is gather data from social media Twitter then realize sentiment alaysis with the data gathered. To this project can be done, many packages must be installed and loaded.

All this project is descript with all steps. First of all we will use sentiment score then we will use Naive Bayes as classifier algorithm.

```r
# install.packages("twitteR")
# install.packages("httr")
# install.packages("knitr")
# install.packages("rmarkdown")
library(twitteR)
library(httr)
library(knitr)
library(rmarkdown)

# Load library created to clean the data.
source('C:\\Users\\Oracy\\Desktop\\DSA_Projetos\\DSA_Projetos\\Big Data Analytics com R e Microsoft Azu
options(warn=-1)
```

Step 1 - Authentication

Below we can find the authentication proccess. Rememer that you need to have a developer account on twitter (https://developer.twitter.com/en/apps) and create an app. All steps to create the application are specified and detailed on the project specification.

```r
# Twitter authentication.
# Font: https://medium.com/@GalarnykMichael/accessing-data-from-twitter-api-using-r-part1-b387a1c7d3e
consumer <- "ZiB0QzMeBY0JFwQGZNisMrBuj"
consumerSecret <- "b8tfwK6bYTB0iQLKP0e4hLCs5kWFdSqtoQNDGhtk7PdC4laqAV"
accessToken <- "199032609-j4014nhYoo0V8xDm6Ngl71jHNUGtcghkWhfIdr23"
accessSecret <- "pX1AffKYylkjqSUNNiSwaeVXWa0MF11ppA8SZ5PBco5j3"

# Twitter Authentication.
# Font: https://www.rdocumentation.org/packages/twitteR/versions/1.1.9/topics/setup_twitter_oauth
twitteR::setup_twitter_oauth(consumer, consumerSecret, accessToken, accessSecret)
```

```
## [1] "Using direct authentication"
```

Step 2 - Connection and data gathering.

Here we will test the connection and get the tweets. How big is your sample, more accurate is your analysis. But this step may take a long time, depending of your internet connection. We will start with Trump query.

```r
# Check user timeline if everything is going fine.
# Font: https://www.r-bloggers.com/visualising-twitter-user-timeline-activity-in-r/
#twitteR::userTimeline("elonmusk")

# Get tweets.
# Font: https://www.rdocumentation.org/packages/twitteR/versions/1.1.9/topics/searchTwitter

# SearchString
query <- "Trump"
# How many tweets will get
#quantity <- 500
# Which language
#language <- "pt"
# Since Date
#sinceDate <- "2018-11-14"
tweet <- twitteR::searchTwitter(query)#, since = sinceDate)

# Check the first 5 tweets.
head(tweet)
```

```
## [[1]]
## [1] "omahabe2: RT @PrisonPlanet: Trump: \"(Osama bin Laden) lived in Pakistan. We're supporting Paki
##
## [[2]]
## [1] "DonnaWa33775564: RT @FoxNewsSunday: Chris Wallace during his interview at the White House with
##
## [[3]]
## [1] "E1A2p3S4: RT @John_KissMyBot: LOVE IT <U+0001F602> TRUMP TAGS DEMOCRAT ADAM SCHIFF WITH A NICKNA
##
## [[4]]
## [1] "DuffShawna: RT @mkraju: Trump calls for "decorum" at the White House and then a couple days late
##
## [[5]]
## [1] "valleylea: RT @sahluwal: White women have overwhelmingly voted for white supremacists the last
##
## [[6]]
## [1] "danieltwisner: RT @tribelaw: Trump’s trashing of Navy Seal Commander General McRaven, the man wh
```

Step 3 - Text mining

Here we will install TM (Text Mining) package. We will convert the tweets from an object to Corpus type, that store data and metadata, after that we will do some clean up proccess, as remove punctuation, convert data to lower case and remove the stopwords.

```r
# Package for Text Mining.
# Font: https://cran.r-project.org/web/packages/tm/vignettes/tm.pdf
# Font: https://eight2late.wordpress.com/2015/05/27/a-gentle-introduction-to-text-mining-using-r/
#install.packages("tm")

# Font: https://cran.r-project.org/web/packages/SnowballC/SnowballC.pdf
#install.packages("SnowballC")
library(SnowballC)
library(tm)
```

```
## Loading required package: NLP

##
## Attaching package: 'NLP'

## The following object is masked from 'package:httr':
##
##     content

##
## Attaching package: 'tm'

## The following object is masked _by_ '.GlobalEnv':
##
##     removePunctuation
```

```r
library(stringr)
options(warn=-1)

# TM Cleaning, organizing and transformation
tweetlist <- sapply(tweet, function(x) x$getText())
tweetlist <- iconv(tweetlist, to = "utf-8", sub="")
tweetlist <- limpaTweets(tweetlist)
tweetcorpus <- VCorpus(VectorSource(tweetlist))
tweetcorpus <- tm_map(tweetcorpus, removePunctuation)
tweetcorpus <- tm_map(tweetcorpus, tolower)
#tweetcorpus <- tm_map(tweetcorpus, function(x)removeWords(x, c(stopwords("en"), "Trump")))
tweetcorpus <- tm_map(tweetcorpus, function(x)removeWords(x, c(stopwords("en"))))
# Test to see how it is going
strwrap(tweetcorpus[[1]])
```

```
## [1] "trump osama bin laden lived pakistan re supporting pakistan gave"
## [2] "billion dollars year wa"
```

```r
# Should convert to plan text before to create the matrix.
tweetcorpusPlan <- tm_map(tweetcorpus, PlainTextDocument)
#tweetListSecond = as.matrix(TermDocumentMatrix(tweetcorpusPlan), control = list(stopwords = c(stopword
tweetListSecond = as.matrix(TermDocumentMatrix(tweetcorpusPlan), control = list(stopwords = c(stopwords
```

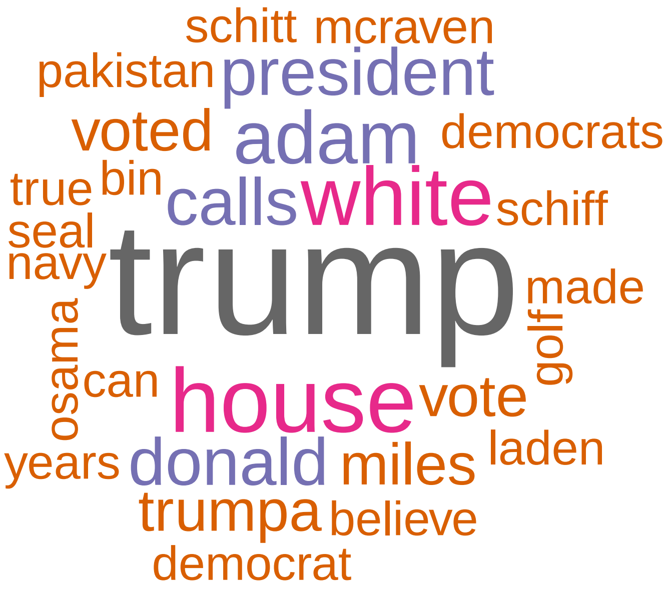
Step 4 - Wordcloud, and dendograma

We will create a wordcloud to check the relationshop between the words that occur with high frequecy. A table was created with the words frequency then we generate a dendogram, that shows how the words relate and associate with the main theme. (Trump)

```r
# Font: http://www.sthda.com/english/wiki/text-mining-and-word-cloud-fundamentals-in-r-5-simple-steps-y
# Install and load wordcloud and RColorBrewer packages
#install.packages("wordcloud") # word-cloud generator
#install.packages("RColorBrewer") # color palettes
library(wordcloud)
```

```
## Loading required package: RColorBrewer
```

```
library(RColorBrewer)

# Generate a wordcloud
pal2 <- brewer.pal(8,"Dark2")

wordcloud(tweetcorpusPlan,
          min.freq = 2,
          scale = c(5,1),
          random.color = F,
          random.order = F,
          colors = pal2)
```



```
# Convert text object to Matrix
tweetMatrix <- TermDocumentMatrix(tweetcorpusPlan)
tweetMatrix
```

```
## <<TermDocumentMatrix (terms: 213, documents: 25)>>
## Non-/sparse entries: 260/5065
## Sparsity           : 95%
## Maximal term length: 14
## Weighting          : term frequency (tf)
```

```r
# Find more frequent word
# Font: https://rdrr.io/rforge/tm/man/findMostFreqTerms.html
findMostFreqTerms(tweetMatrix)
```

```
## $`character(0)`
## pakistan  billion      bin  dollars     gave    laden 
##        2        1        1        1        1        1 
## 
## $`character(0)`
##     check     chris     house interview  listings     local 
##         1         1         1         1         1         1 
## 
## $`character(0)`
##     adam democrat      lil   little     love    namea 
##        3        1        1        1        1        1 
## 
## $`character(0)`
##    calls    house chairman   couple     days decoruma 
##        2        2        1        1        1        1 
## 
## $`character(0)`
##          voted          white           last            moa overwhelmingly 
##              3              2              1              1              1 
##            roy 
##              1 
## 
## $`character(0)`
## arlington       bin commander   general     laden       man 
##         1         1         1         1         1         1 
## 
## $`character(0)`
##       calls distraction       focus        will 
##           1           1           1           1 
## 
## $`character(0)`
##    campaign   candidate centerpiece     history        made      office 
##           1           1           1           1           1           1 
## 
## $`character(0)`
##   better    first     lady  melania michelle    obama 
##        1        1        1        1        1        1 
## 
## $`character(0)`
##   brushes       buy       can    donald excellent      indy 
##         1         1         1         1         1         1 
## 
## $`character(0)`
##  trumpa     adm  column conceit   light mcraven 
##       2       1       1       1       1       1 
## 
## $`character(0)`
##  america      amp  execute flawless  justice  mission 
##        1        1        1        1        1        1
```

```
##
## $`character(0)`
##  actual  almost believe     can  donald dumbest
##       1       1       1       1       1       1
##
## $`character(0)`
##        amera     american conversation       donald      finland
##            1            1            1            1            1
##         knew
##            1
##
## $`character(0)`
##       ana      bush    donald     fight    george goddammit
##         1         1         1         1         1         1
##
## $`character(0)`
##      miles       golf      house      white bedminster     course
##          3          2          2          2          1          1
##
## $`character(0)`
## democratic  democrats       four      house  incumbent       lost
##          1          1          1          1          1          1
##
## $`character(0)`
##      admin    comment       join opposition   proposed submitting
##          1          1          1          1          1          1
##
## $`character(0)`
##     attacking          bina    criticized disrespecting         ended
##             1             1             1             1             1
##       getting
##             1
##
## $`character(0)`
##      adam     calls  democrat      post president    schiff
##         2         1         1         1         1         1
##
## $`character(0)`
##      dems   despite districts    gained   popular     rural
##         1         1         1         1         1         1
##
## $`character(0)`
##       house        inta interesting      mciver    meredith         see
##           1           1           1           1           1           1
##
## $`character(0)`
## named numeric(0)
##
## $`character(0)`
##       true    believe definitely  democrats       orda    percent
##          2          1          1          1          1          1
##
## $`character(0)`
##   canna   elect foreign happens   hates leftist
##       1       1       1       1       1       1
```

```
# Search for Association
# Font: https://rdrr.io/rforge/tm/man/findAssocs.html
findAssocs(tweetMatrix, "fascist", 0.6)
```

```
## $fascist
## numeric(0)
```

```
# Removing sparse terms
# Font: https://stackoverflow.com/questions/28763389/how-does-the-removesparseterms-in-r-work
tweetMatrix2 <- removeSparseTerms(tweetMatrix, .90)
tweetMatrix2
```

```
## <<TermDocumentMatrix (terms: 7, documents: 25)>>
## Non-/sparse entries: 38/137
## Sparsity           : 78%
## Maximal term length: 9
## Weighting          : term frequency (tf)
```

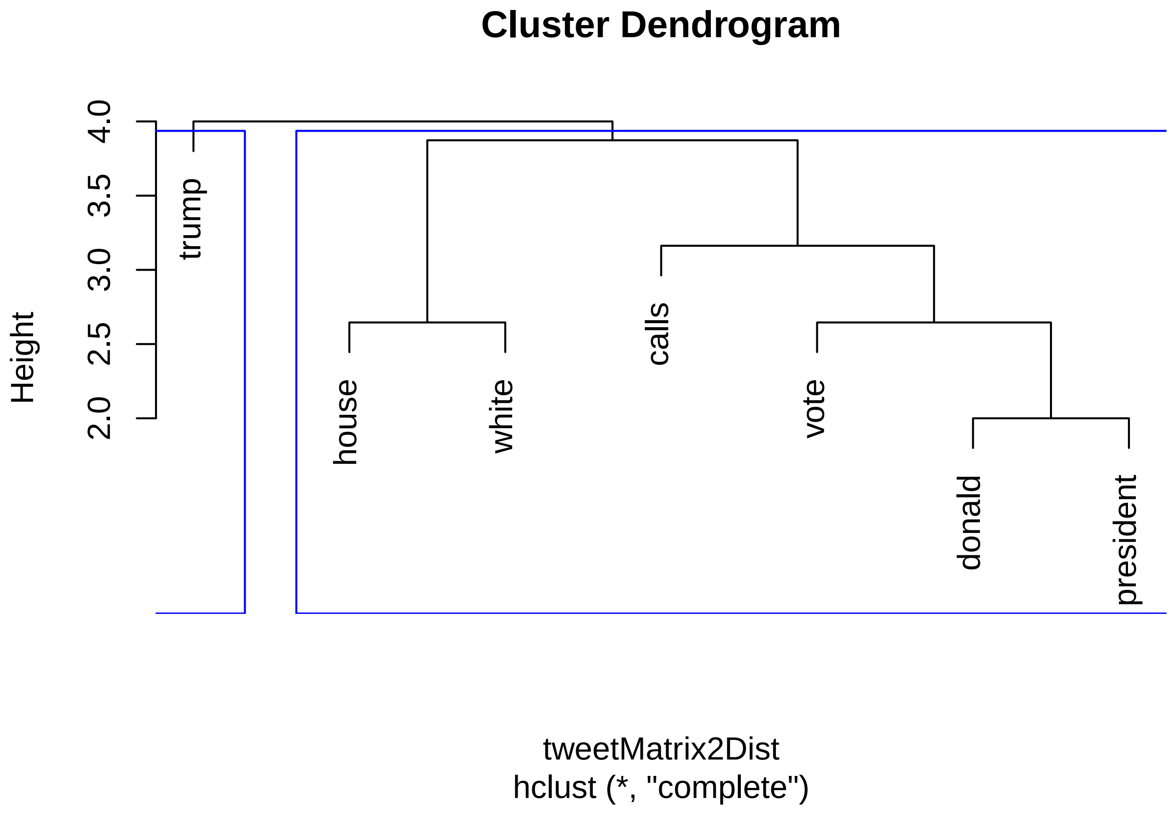
```
# Creating scale
tweetMatrix2Scale <- scale(tweetMatrix2)
tweetMatrix2Scale
```

```
##            Docs
## Terms       character(0) character(0) character(0) character(0)
##   calls       -0.3779645   -1.0690450   -0.3779645    1.2702147
##   donald      -0.3779645   -1.0690450   -0.3779645   -0.9526610
##   house       -0.3779645    0.8017837   -0.3779645    1.2702147
##   president   -0.3779645    0.8017837   -0.3779645   -0.9526610
##   trump        2.2677868    0.8017837    2.2677868    0.1587768
##   vote        -0.3779645   -1.0690450   -0.3779645   -0.9526610
##   white       -0.3779645    0.8017837   -0.3779645    0.1587768
##            Docs
## Terms       character(0) character(0) character(0) character(0)
##   calls       -0.5447048          NaN    2.2677868   -0.3779645
##   donald      -0.5447048          NaN   -0.3779645   -0.3779645
##   house       -0.5447048          NaN   -0.3779645   -0.3779645
##   president   -0.5447048          NaN   -0.3779645   -0.3779645
##   trump        0.7262730          NaN   -0.3779645    2.2677868
##   vote        -0.5447048          NaN   -0.3779645   -0.3779645
##   white        1.9972509          NaN   -0.3779645   -0.3779645
##            Docs
## Terms       character(0) character(0) character(0) character(0)
##   calls         -0.58554     -0.58554          NaN          NaN
##   donald        -0.58554      1.46385          NaN          NaN
##   house         -0.58554     -0.58554          NaN          NaN
##   president     -0.58554     -0.58554          NaN          NaN
##   trump          1.46385      1.46385          NaN          NaN
##   vote           1.46385     -0.58554          NaN          NaN
##   white         -0.58554     -0.58554          NaN          NaN
##            Docs
## Terms       character(0) character(0) character(0) character(0)
##   calls       -0.8017837   -0.8017837     -0.58554   -0.7509393
##   donald       1.0690450    1.0690450      1.46385   -0.7509393
##   house       -0.8017837   -0.8017837     -0.58554    1.3516907
##   president    1.0690450    1.0690450     -0.58554   -0.7509393
##   trump        1.0690450    1.0690450      1.46385    0.3003757
##   vote        -0.8017837   -0.8017837     -0.58554   -0.7509393
##   white       -0.8017837   -0.8017837     -0.58554    1.3516907
##            Docs
## Terms       character(0) character(0) character(0) character(0)
##   calls       -0.3779645   -0.3779645   -0.3779645    1.0690450
##   donald      -0.3779645   -0.3779645   -0.3779645   -0.8017837
##   house        2.2677868   -0.3779645   -0.3779645   -0.8017837
##   president   -0.3779645   -0.3779645   -0.3779645    1.0690450
##   trump       -0.3779645    2.2677868    2.2677868    1.0690450
##   vote        -0.3779645   -0.3779645   -0.3779645   -0.8017837
##   white       -0.3779645   -0.3779645   -0.3779645   -0.8017837
##            Docs
## Terms       character(0) character(0) character(0) character(0)
##   calls       -0.3779645   -0.3779645          NaN   -0.3779645
##   donald      -0.3779645   -0.3779645          NaN   -0.3779645
##   house       -0.3779645    2.2677868          NaN   -0.3779645
##   president   -0.3779645   -0.3779645          NaN   -0.3779645
##   trump       -0.3779645   -0.3779645          NaN   -0.3779645
##   vote         2.2677868   -0.3779645          NaN    2.2677868
##   white       -0.3779645   -0.3779645          NaN   -0.3779645
##            Docs
## Terms       character(0)
##   calls              NaN
##   donald             NaN
##   house              NaN
##   president          NaN
##   trump              NaN
##   vote               NaN
##   white              NaN
## attr(,"scaled:center")
## character(0) character(0) character(0) character(0) character(0)
##    0.1428571    0.5714286    0.1428571    0.8571429    0.4285714
## character(0) character(0) character(0) character(0) character(0)
##    0.0000000    0.1428571    0.1428571    0.2857143    0.2857143
## character(0) character(0) character(0) character(0) character(0)
##    0.0000000    0.0000000    0.4285714    0.4285714    0.2857143
## character(0) character(0) character(0) character(0) character(0)
##    0.7142857    0.1428571    0.1428571    0.1428571    0.4285714
## character(0) character(0) character(0) character(0) character(0)
##    0.1428571    0.1428571    0.0000000    0.1428571    0.0000000
## attr(,"scaled:scale")
## character(0) character(0) character(0) character(0) character(0)
##    0.3779645    0.5345225    0.3779645    0.8997354    0.7867958
## character(0) character(0) character(0) character(0) character(0)
##    0.0000000    0.3779645    0.3779645    0.4879500    0.4879500
## character(0) character(0) character(0) character(0) character(0)
##    0.0000000    0.0000000    0.5345225    0.5345225    0.4879500
## character(0) character(0) character(0) character(0) character(0)
##    0.9511897    0.3779645    0.3779645    0.3779645    0.5345225
## character(0) character(0) character(0) character(0) character(0)
##    0.3779645    0.3779645    0.0000000    0.3779645    0.0000000
```

```
# Distance Matrix
tweetMatrix2Dist <- dist(tweetMatrix2)

# Dendogram
# Font: https://dendrolab.wordpress.com/2010/11/03/construindo-dendrogramas-usando-o-r/
tweetMatrix2Hclust <- hclust(tweetMatrix2Dist)

# Creating dendograma (verify how words clustering each other)
plot(tweetMatrix2Hclust)

# Checking groups
cutree(tweetMatrix2Hclust, k = 2)
```

```
##     calls    donald     house president     trump      vote     white
##         1         1         1         1         2         1         1
```

```
# Visualizing the word groups on dendogram
# Font: https://stat.ethz.ch/R-manual/R-devel/library/stats/html/rect.hclust.html
rect.hclust(tweetMatrix2Hclust, k = 2, border = "blue")
```



Step 5 - Sentiment Analysis

Now we can proceed with the sentiment analysis.

```r
# Load packages
library(syuzhet)
library(stringr)
library(plyr)
```

```
##
## Attaching package: 'plyr'

## The following object is masked from 'package:twitteR':
##
##     id
```

```r
# Getting sentiment score for each tweet
# Font: http://dataaspirant.com/2018/03/22/twitter-sentiment-analysis-using-r/
tweetlistVector <- as.vector(tweetlist)
emotion <- get_nrc_sentiment(tweetlistVector)
emotion2 <- cbind(tweetlist, emotion)
head(emotion2)
```

```
##
## 1                                   trump osama bin laden lived in pakistan we re supporting pakistan
## 2                         chris wallace during his interview at the white house with president trump
## 3 love it a a e a trump tags democrat adam schiff with a nickname little adam a a e schitta a a a a a
## 4        trump calls for a a a decoruma a a at the white house and then a couple days later calls the
## 5                                 white women have overwhelmingly voted for white supremacists the la
## 6                trumpa a a s trashing of navy seal commander general mcraven the man who took down o
##   anger anticipation disgust fear joy sadness surprise trust negative
## 1     0            0       0    0   0       0        1     1        1
## 2     0            2       0    1   1       0        1     2        0
## 3     0            1       0    0   2       0        1     1        0
## 4     0            2       0    0   1       0        1     3        1
## 5     0            1       0    0   1       0        1     1        0
## 6     0            0       0    0   0       0        0     2        1
##   positive
## 1        1
## 2        2
## 3        2
## 4        2
## 5        1
## 6        2
```

```r
# get_sentiment function to extract sentiment score for each of the tweets.
sentimentValue <- get_sentiment(tweetlistVector)

mostPositive <- tweetlistVector[sentimentValue == max(sentimentValue)]

mostPositive
```

```
## [1] "this is seal team six they trained for years to execute a flawless mission amp win justice for a
```

```r
# Segregating positive and negative tweets
# Positive Tweets
positiveTweets <- tweetlistVector[sentimentValue > 0]

head(positiveTweets)
```

```
## [1] "trump osama bin laden lived in pakistan we re supporting pakistan we gave them billion dollars a
## [2] "chris wallace during his interview at the white house with president trump check your local list
## [3] "love it a a e a trump tags democrat adam schiff with a nickname little adam a a e schitta a a a
## [4] "trump calls for a a a decoruma a a at the white house and then a couple days later calls the inc
## [5] "white women have overwhelmingly voted for white supremacists the last few years voted for trump
## [6] "trumpa a a s trashing of navy seal commander general mcraven the man who took down osama bin lac
```

```r
# Negative Tweets
negativeTweets <- tweetlistVector[sentimentValue < 0]

head(negativeTweets)
```

```
## [1] "s distraction so he calls s will focus on ta a a"
## [2] "in light of trumpa a a s swipe at adm mcraven reupping my stopinions column a a a so much for th
## [3] "sta we the american people knew you finland s president had no such conversation with lying dona
## [4] "when the rest of us were sent to fight george bush s war s plural goddammit donald trump stayed
## [5] "miles from white house to mar a lago for golf miles from white house to trump bedminster golf co
## [6] "join us for tomorrow s submitting a comment in opposition to the trump admin s proposed"
```

```r
# Neutral Tweets
neutralTweets <- tweetlistVector[sentimentValue == 0]

head(neutralTweets)
```

```
## [1] "a a"
```

```r
# Alternate way to classify as Positive, Negative or Neutral tweets
categorySentiment <- ifelse(sentimentValue < 0, "Negative", ifelse(sentimentValue > 0, "Positive", "Neu

head(categorySentiment)
```

```
## [1] "Positive" "Positive" "Positive" "Positive" "Positive" "Positive"
```

```r
categorySentiment2 <- cbind(tweetlistVector, categorySentiment)

head(categorySentiment2)
```

```
##      tweetlistVector
## [1,] "trump osama bin laden lived in pakistan we re supporting pakistan we gave them billion dollars
## [2,] "chris wallace during his interview at the white house with president trump check your local lis
## [3,] "love it a a e a trump tags democrat adam schiff with a nickname little adam a a e schitta a a a
## [4,] "trump calls for a a a decoruma a a at the white house and then a couple days later calls the in
## [5,] "white women have overwhelmingly voted for white supremacists the last few years voted for trump
## [6,] "trumpa a a s trashing of navy seal commander general mcraven the man who took down osama bin la
##      categorySentiment
## [1,] "Positive"
## [2,] "Positive"
## [3,] "Positive"
## [4,] "Positive"
## [5,] "Positive"
## [6,] "Positive"
```

```
# Tabule information
table(categorySentiment)
```

```
## categorySentiment
## Negative  Neutral Positive
##        9        1       15
```

```
# Other way to Sentiment Analysis
# Create sentiment.score function
sentiment.score = function(sentences, pos.words, neg.words, .progress = 'none')
{

  # Criando um array de scores com lapply
  scores = laply(sentences,
                 function(sentence, pos.words, neg.words)
                 {
                   sentence = gsub("[[:punct:]]", "", sentence)
                   sentence = gsub("[[:cntrl:]]", "", sentence)
                   sentence =gsub('\\d+', '', sentence)
                   tryTolower = function(x)
                   {
                     y = NA

                     # Tratamento de Erro
                     try_error = tryCatch(tolower(x), error=function(e) e)
                     if (!inherits(try_error, "error"))
                       y = tolower(x)
                     return(y)
                   }

                   sentence = sapply(sentence, tryTolower)
                   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(text = sentences, score = scores)
  return(scores.df)
}
```

```
# Mapping the positive and negative words
pos = readLines("C:\\Users\\Oracy\\Desktop\\DSA_Projetos\\DSA_Projetos\\Big Data Analytics com R e Micr
neg = readLines("C:\\Users\\Oracy\\Desktop\\DSA_Projetos\\DSA_Projetos\\Big Data Analytics com R e Micr

# Testing function on our tweets
tweetSentiment = sentiment.score(tweetlistVector, pos, neg)
class(tweetSentiment)
```

```
## [1] "data.frame"
```

```
# Checking Score
# 0 - Expression doesn't have any word on our lists either positive or negative, or there is positive a
# 1 - Expression has positive words
# -1 - Expression has negative words
tweetSentiment$score
```

```
##  [1]  2  1  3  1  0  0 -1  1  2  2 -2  2  1  0  0  0  0  0 -1  1  2  1  0
## [24]  0 -1
```

Step 6 - Generating Sentiment Analysis Score

With the score calculate, we will split by country, this case CA and USA, as way to compare the sentiment between two different region. Generate boxplot and a histogram using lattice package.

```
# Tweets by country
caTweets = twitteR::searchTwitter("ca", n = 300, lang = "en")
usaTweets = twitteR::searchTwitter("usa", n = 300, lang = "en")

# Getting text
# Font: https://producaoanimalcomr.wordpress.com/2015/12/10/entendendo-o-uso-das-funcoes-apply-lapply-s
caTxt = sapply(caTweets, function(x) x$getText())
usaTxt = sapply(usaTweets, function(x) x$getText())

# Tweet vector by country
countryTweet = c(length(caTxt), length(usaTxt))

# Append both text
countries = c(caTxt, usaTxt)

# Applying function to calculate sentiment score.
scores = sentiment.score(countries, pos, neg, .progress = 'text')
```

```
##
  |
  |                                                                 |   0%
  |
  |                                                                 |   1%
  |
  |=                                                                |   1%
  |
  |=                                                                |   2%
```

```
|
|==                                                                |   2%
|
|==                                                                |   3%
|
|==                                                                |   4%
|
|===                                                               |   4%
|
|===                                                               |   5%
|
|====                                                              |   6%
|
|====                                                              |   7%
|
|=====                                                             |   7%
|
|=====                                                             |   8%
|
|======                                                            |   8%
|
|======                                                            |   9%
|
|======                                                            |  10%
|
|=======                                                           |  10%
|
|=======                                                           |  11%
|
|=======                                                           |  12%
|
|========                                                          |  12%
|
|========                                                          |  13%
|
|=========                                                         |  13%
|
|=========                                                         |  14%
|
|==========                                                        |  15%
|
|==========                                                        |  16%
|
|===========                                                       |  16%
|
|===========                                                       |  17%
|
|===========                                                       |  18%
|
|============                                                      |  18%
|
|============                                                      |  19%
|
|=============                                                     |  19%
```

```
|
|=============                                                    |  20%
|
|=============                                                    |  21%
|
|==============                                                   |  21%
|
|==============                                                   |  22%
|
|===============                                                  |  22%
|
|===============                                                  |  23%
|
|===============                                                  |  24%
|
|================                                                 |  24%
|
|================                                                 |  25%
|
|=================                                                |  26%
|
|=================                                                |  27%
|
|==================                                               |  27%
|
|==================                                               |  28%
|
|===================                                              |  28%
|
|===================                                              |  29%
|
|===================                                              |  30%
|
|====================                                             |  30%
|
|====================                                             |  31%
|
|====================                                             |  32%
|
|=====================                                            |  32%
|
|=====================                                            |  33%
|
|======================                                           |  33%
|
|======================                                           |  34%
|
|=======================                                          |  35%
|
|=======================                                          |  36%
|
|========================                                         |  36%
|
|========================                                         |  37%
```

```
|
|========================                                         |  38%
|
|=========================                                        |  38%
|
|=========================                                        |  39%
|
|==========================                                       |  39%
|
|==========================                                       |  40%
|
|==========================                                       |  41%
|
|===========================                                      |  41%
|
|===========================                                      |  42%
|
|============================                                     |  42%
|
|============================                                     |  43%
|
|============================                                     |  44%
|
|=============================                                    |  44%
|
|=============================                                    |  45%
|
|==============================                                   |  46%
|
|==============================                                   |  47%
|
|===============================                                  |  47%
|
|===============================                                  |  48%
|
|================================                                 |  48%
|
|================================                                 |  49%
|
|================================                                 |  50%
|
|=================================                                |  50%
|
|=================================                                |  51%
|
|=================================                                |  52%
|
|==================================                               |  52%
|
|==================================                               |  53%
|
|===================================                              |  53%
|
|===================================                              |  54%
```

```
|
|====================================                             |  55%
|
|====================================                             |  56%
|
|=====================================                            |  56%
|
|=====================================                            |  57%
|
|=====================================                            |  58%
|
|======================================                           |  58%
|
|======================================                           |  59%
|
|=======================================                          |  59%
|
|=======================================                          |  60%
|
|=======================================                          |  61%
|
|========================================                         |  61%
|
|========================================                         |  62%
|
|=========================================                        |  62%
|
|=========================================                        |  63%
|
|=========================================                        |  64%
|
|==========================================                       |  64%
|
|==========================================                       |  65%
|
|===========================================                      |  66%
|
|===========================================                      |  67%
|
|============================================                     |  67%
|
|============================================                     |  68%
|
|=============================================                    |  68%
|
|=============================================                    |  69%
|
|=============================================                    |  70%
|
|==============================================                   |  70%
|
|==============================================                   |  71%
|
|==============================================                   |  72%
```

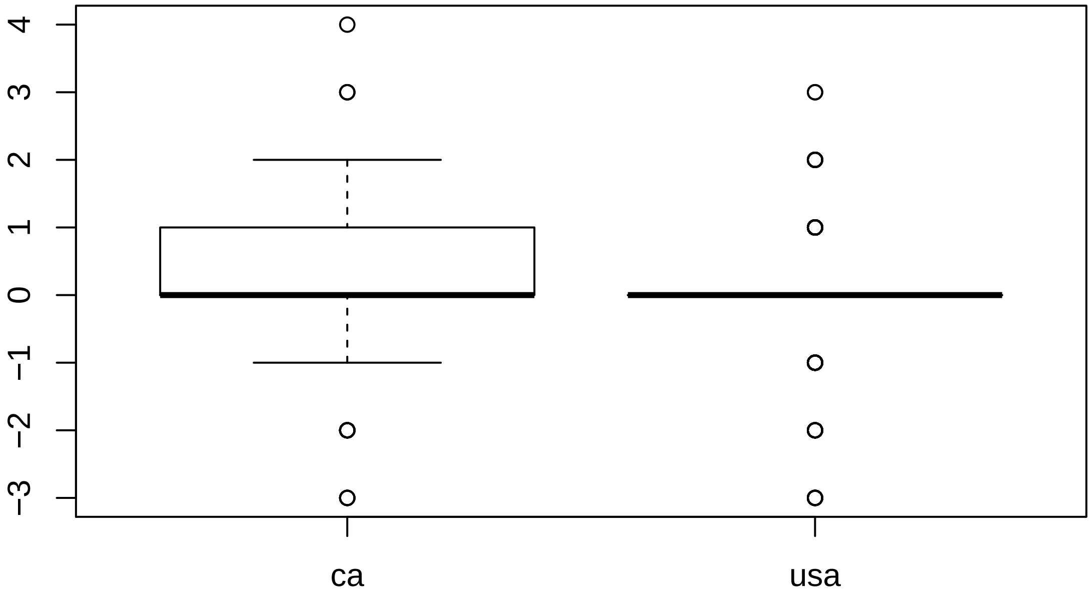
```
|
|===============================================                  |  72%
|
|===============================================                  |  73%
|
|================================================                 |  73%
|
|================================================                 |  74%
|
|=================================================                |  75%
|
|=================================================                |  76%
|
|==================================================               |  76%
|
|==================================================               |  77%
|
|==================================================               |  78%
|
|===================================================              |  78%
|
|===================================================              |  79%
|
|====================================================             |  79%
|
|====================================================             |  80%
|
|====================================================             |  81%
|
|=====================================================            |  81%
|
|=====================================================            |  82%
|
|======================================================           |  82%
|
|======================================================           |  83%
|
|======================================================           |  84%
|
|=======================================================          |  84%
|
|=======================================================          |  85%
|
|========================================================         |  86%
|
|========================================================         |  87%
|
|=========================================================        |  87%
|
|=========================================================        |  88%
|
|==========================================================       |  88%
|
|==========================================================       |  89%
```

```
  |
  |==========================================================       |  90%
  |
  |===========================================================      |  90%
  |
  |===========================================================      |  91%
  |
  |===========================================================      |  92%
  |
  |============================================================     |  92%
  |
  |============================================================     |  93%
  |
  |=============================================================    |  93%
  |
  |=============================================================    |  94%
  |
  |==============================================================   |  95%
  |
  |==============================================================   |  96%
  |
  |===============================================================  |  96%
  |
  |===============================================================  |  97%
  |
  |===============================================================  |  98%
  |
  |================================================================ |  98%
  |
  |================================================================ |  99%
  |
  |=================================================================|  99%
  |
  |=================================================================| 100%
```

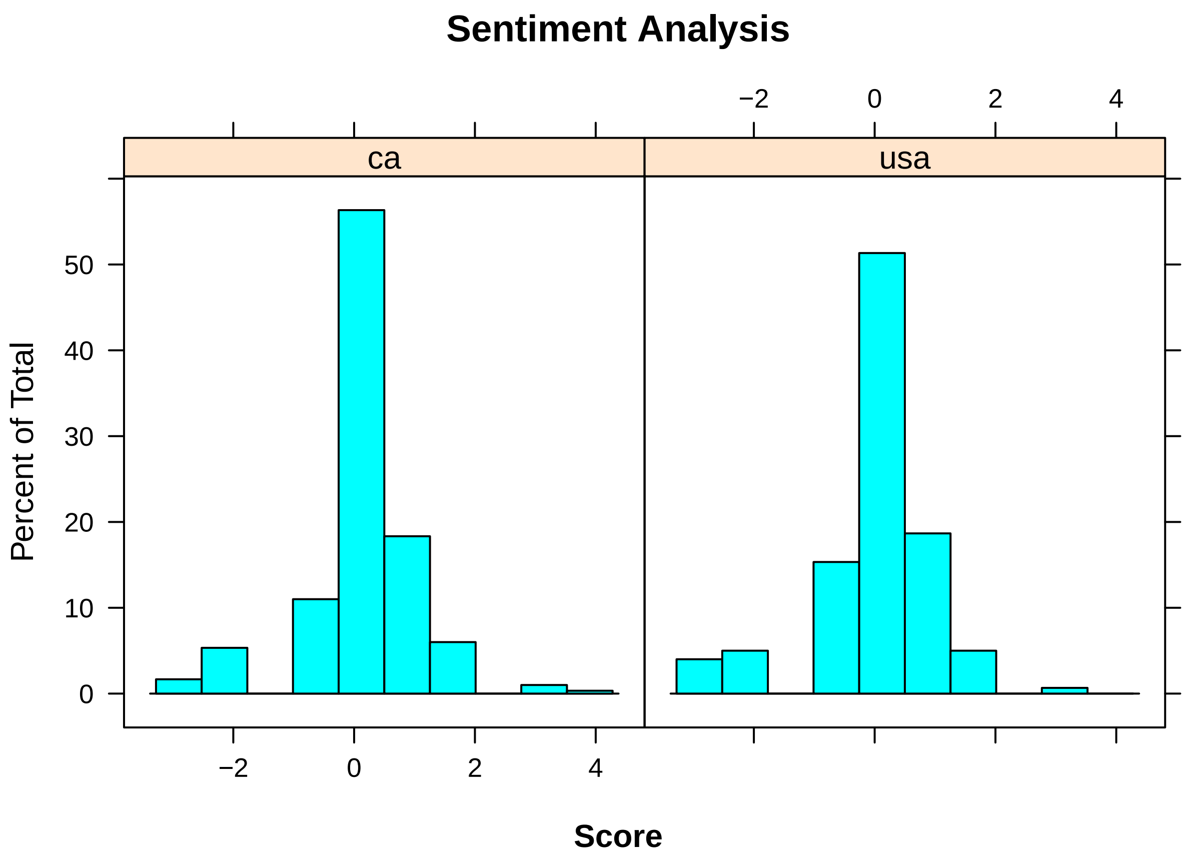
```r
# Calculating score by country
scores$countries = factor(rep(c("ca", "usa"), countryTweet))
scores$muito.pos = as.numeric(scores$score >= 1)
scores$muito.neg = as.numeric(scores$score <= -1)

# Calculating the total
numpos = sum(scores$muito.pos)
numneg = sum(scores$muito.neg)

# Score global
global_score = round( 100 * numpos / (numpos + numneg) )
head(scores)
```

```
##
## 1                                  RT @derrrmonasterio: -far far away-\nThere’s always happiness in
## 2     JOB; Stockton CA USA - Per Diem Home Health Registered Nurse - ... and hospital ICU intensive
## 3                12W night sensor Solar Light LED Flood Lamp indoor and Outdoor Garden Spotlights ht
## 4 RT @MikeLevinCA: Some facts on CA wildfires for @realDonaldTrump:\n\n-Your administration cut fundi
## 5    RT @chalkomilk: Gun violence is falling in Canada, while gangs with smuggled guns are killing e
## 6    RT @SportsCentre: The #Stampeders advance to their third straight #GreyCup with a 22-14 win ove
##   score countries muito.pos muito.neg
## 1     1        ca         1         0
## 2     0        ca         0         0
## 3     1        ca         1         0
## 4     0        ca         0         0
## 5    -2        ca         0         1
## 6     1        ca         1         0
```

```
boxplot(score ~ countries, data = scores)

# Generating a histogram with lattice package
# install.packages("lattice")
library("lattice")
```



```
histogram(data = scores, ~score|countries, main = "Sentiment Analysis", xlab = "", sub = "Score")
```



Extra

```
# install.packages("Rstem_0.4-1.tar.gz", repos = NULL, type = "source")
# install.packages("sentiment_0.2.tar.gz", repos = NULL, type = "source")
# install.packages("ggplot2")
library(Rstem)
```

```
##
## Attaching package: 'Rstem'

## The following objects are masked from 'package:SnowballC':
##
##     getStemLanguages, wordStem
```

```
library(sentiment)
library(ggplot2)
```

```
##
## Attaching package: 'ggplot2'

## The following object is masked from 'package:NLP':
##
##     annotate
```

Get Tweets

The tweets are collected by function searchTwitter() from twitteR package.

```
# Gathering tweets
tweetEn = searchTwitter("Trump", n = 1500, lang = "en")

# Get text
tweetEn = sapply(tweetEn, function(x) x$getText())
```

Cleaning, Organazing and Data Transformation

```
# Remove http links
tweetEn = gsub("(https?://*.[^\\s]+)", "", tweetEn)
# Remove retweets
tweetEn = gsub("(RT|via)((?:\\b\\W*@\\w+)+)", " ", tweetEn)
# Remove "#Hashtag"
tweetEn = gsub("(#\\w*.[^\\s]+)", "", tweetEn)
# Remove username "@people"
tweetEn = gsub("(@\\w[^\\s]+)", "", tweetEn)
# Remove punctuation
tweetEn = gsub("(\\W)", " ", tweetEn)
# Remove numbers
tweetEn = gsub("(\\d)", "", tweetEn)
# Remove unnecessary blank space
tweetEn = gsub("\\s+", " ", str_trim(tweetEn))

# Removing NAs Value
tweetEn = tweetEn[!is.na(tweetEn)]
names(tweetEn) = NULL
```

Naive Bayes Classifier

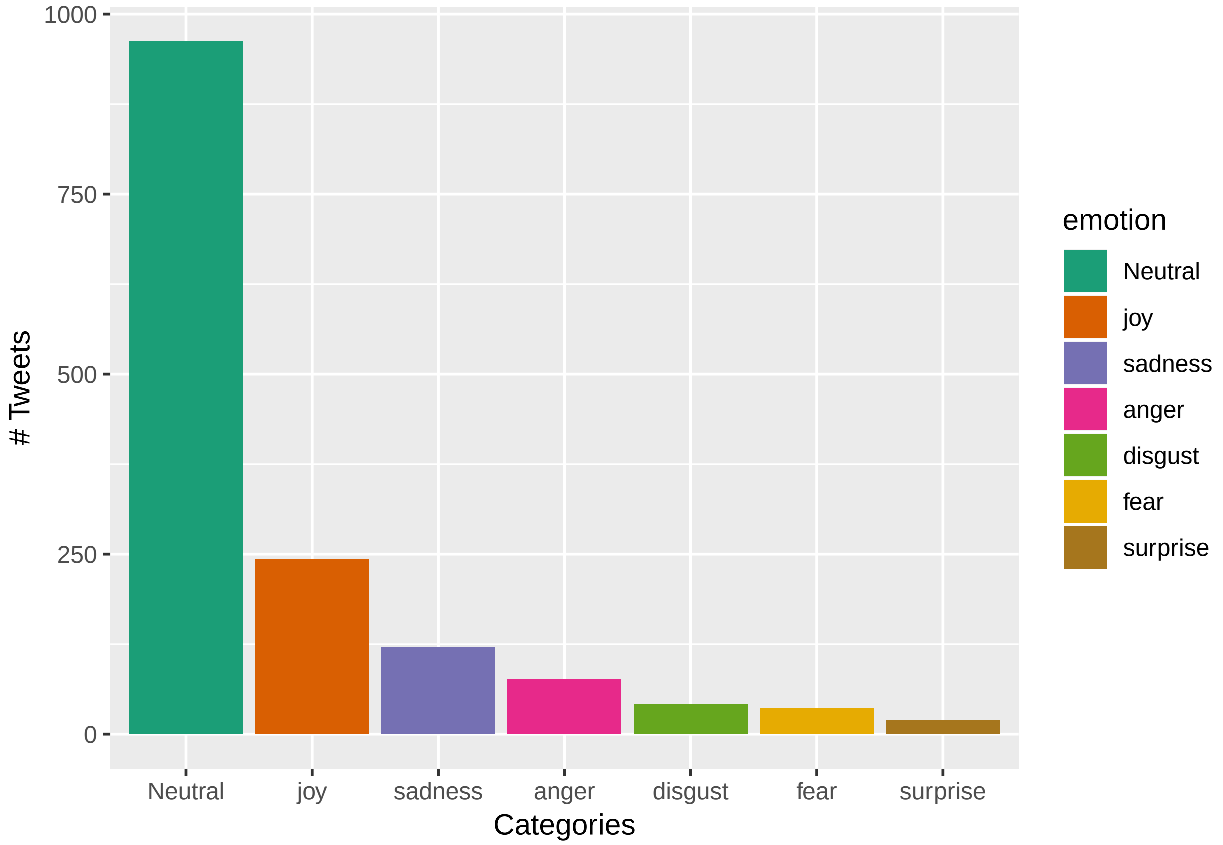
I used the functions classify_emotion() and classify_polarity() from sentiment package, that they are based on Naive Bayes to sentiment analysis. This case the own algorithm do the word classification and we do not need to create words lists, positives neither negatives.

```
# Classifying emotion
class_emo = classify_emotion(tweetEn, algorithm = "bayes", prior = 1.0)
emotion = class_emo[,7]

# Replacing NAs to "Neutral"
emotion[is.na(emotion)] = "Neutral"

# Classifying polarity
class_pol = classify_polarity(tweetEn, algorithm = "bayes")
polarity = class_pol[,4]

# Generating a dataframe with the results
sent_df = data.frame(text = tweetEn, emotion = emotion,
                       polarity = polarity, stringsAsFactors = FALSE)

# Ordering dataframe
sent_df = within(sent_df,
                 emotion <- factor(emotion, levels = names(sort(table(emotion),
                                                                 decreasing=TRUE))))
```

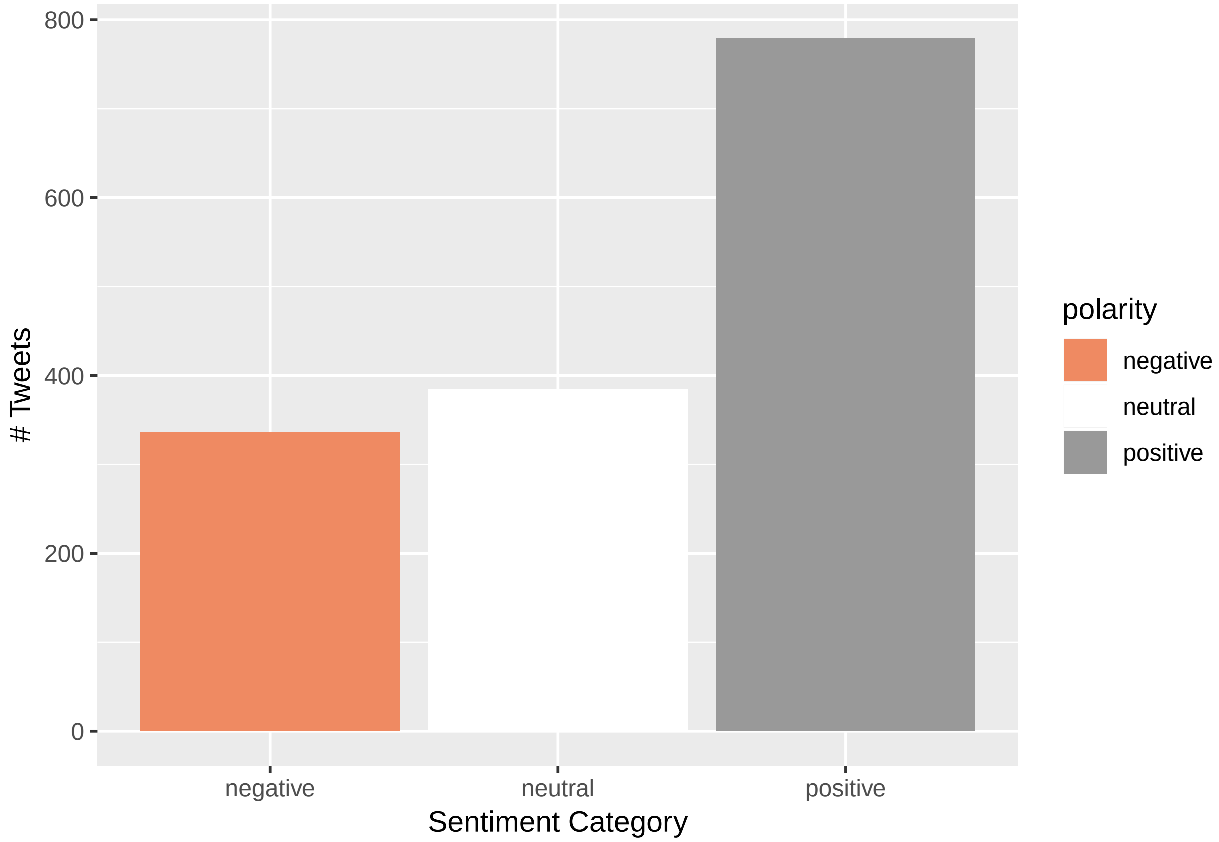
Visualization

Finally, using ggplot2 to visualize the results.

```
# Emotions found
ggplot(sent_df, aes(x = emotion)) +
  geom_bar(aes(y = ..count.., fill = emotion)) +
  scale_fill_brewer(palette = "Dark2") +
  labs(x = "Categories", y = "# Tweets")
```



```
# Polarity
ggplot(sent_df, aes(x = polarity)) +
  geom_bar(aes(y = ..count.., fill = polarity)) +
  scale_fill_brewer(palette = "RdGy") +
  labs(x = "Sentiment Category", y = "# Tweets")
```


800
600
400
200
0
# Tweets
negative
neutral
positive
Sentiment Category
polarity
negative
neutral
positive

End