Data Science Academy - Mini-Projeto 1

Equipe DSA

10 de Julho, 2018

% !TEX encoding = UTF-8 Unicode

Mini-Projeto 1 - Análise de Sentimentos em Redes Sociais

Este projeto é parte integrante do curso Big Data Analytics com R e Microsoft Azure da Formação Cientista de Dados. O objetivo é captutar dados da rede social Twitter e realizar análise de sentimentos com os dados capturados. Para que este projeto possa ser executado, diversos pacotes devem ser instalados e carregados.

Todo o projeto será descrito de acordo com suas etapas. Primeiro usaremos o cálculo de score de sentimento e em seguida usaremos um classificador com o algoritmo Naive Bayes.

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

# Carregando a biblioteca com funções de limpeza
source('utils.R')
options(warn=-1)
```

Etapa 1 - Autenticação

Abaixo você encontra o processo de autenticação. Lembre-se que você precisa ter uma conta criada no Twitter e criar uma aplicação. Os passos para criação da aplicação estão detalhados na especificação do projeto.

```
# Criando autenticação no Twitter
key <- "9GiNf3GH6LJJstwTZ3518FvT5"
secret <- "d2DOy4mozbBX9v95KPAq9hMovxrCqzO7rzl0pweaLW3klbYLqV"
token <- "703383646602981377-M3SYWCKiYOZdqnBlfFbLlvZPFCANCp5"
tokensecret <- "FYObmCWM2d6KzJ7YUSUMrqgdlrarThr9DUQnsrKmN6Nw8"

# Autenticação. Responda 1 (Yes) quando perguntado sobre utilizar direct connection.
setup_twitter_oauth(key, secret, token, tokensecret)
```

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

Etapa 2 - Conexão

Aqui vamos testar a conexão e capturar os tweets. Quanto maior sua amostra, mais precisa sua análise. Mas a coleta de dados pode levar tempo, dependendo da sua conexão com a internet. Comece com 100 tweets, pois à medida que você aumenta a quantidade, vai exigir mais recursos do seu computador. Buscaremos tweets com referência a hashtag #BigData.

```r
# Verificando a timeline do usuário
userTimeline("dsacademybr")
```

```
## [[1]]
## [1] "dsacademybr: Um novo mundo bem a nossa frente! No futuro, pais permitirão que filhos sejam subm
##
## [[2]]
## [1] "dsacademybr: @MaurlioJoseDos1 Oi Maurílio. Dá uma olhada na lista de empresas que estão fazendo
##
## [[3]]
## [1] "dsacademybr: Trilha de Aprendizagem - Formação Engenheiro de Dados: https://t.co/MRopryEKaK via
##
## [[4]]
## [1] "dsacademybr: Trilha de Aprendizagem - Formação Java Para Data Science e IA: https://t.co/4aJM93
##
## [[5]]
## [1] "dsacademybr: Trilha de Aprendizagem - Formação Inteligência Artificial: https://t.co/KyLqFOQNRY
##
## [[6]]
## [1] "dsacademybr: Trilha de Aprendizagem - Formação Cientista de Dados: https://t.co/q37MUJAVrv via
##
## [[7]]
## [1] "dsacademybr: A Diferença Entre Cientistas de Dados, Engenheiros de Dados, Estatísticos e Engenh
##
## [[8]]
## [1] "dsacademybr: Passo importante para o Brasil! Em caráter de urgência, Congresso aprova PL de pro
##
## [[9]]
## [1] "dsacademybr: Mais um capítulo do Deep  Learning Book. Em português, online e gratuito! Já receb
##
## [[10]]
## [1] "dsacademybr: Você já está aprendendo a trabalhar com Inteligência Artificial? Não? Sério?\n\nDi
##
## [[11]]
## [1] "dsacademybr: Qual a Diferença Entre o Analista de BI e o Cientista de Dados? https://t.co/8uvuA
##
## [[12]]
## [1] "dsacademybr: Para pensar! https://t.co/6p13YB3NFn"
##
## [[13]]
## [1] "dsacademybr: 29 Certificações em Big Data e Data Science \nhttps://t.co/kQ2LEtYxVz\n#BigData #Da
##
## [[14]]
## [1] "dsacademybr: Finalmente) cientistas estão indo para as empresas, neste caso, startups \nhttps:/
##
## [[15]]
## [1] "dsacademybr: Gostaria de dar as boas vindas a turma de alunos da TV Globo e do SBT, que se junta
##
## [[16]]
## [1] "dsacademybr: A Inteligência Artificial está perto do seu ponto de inflexão\nhttps://t.co/pC6KlNu
##
## [[17]]
## [1] "dsacademybr: Prezados Alunos do Curso Data Lake - Design, Projeto e Integração, O capítulo 6: Da
##
```

```
## [[18]]
## [1] "dsacademybr: Artigo interessante mostrando a estratégia de diversos países para a Inteligência A
##
## [[19]]
## [1] "dsacademybr: Anunciamos várias novidades em nossos cursos e os próximos lançamentos para o segu
```

```
# Capturando os tweets
tema <- "BigData"
qtd_tweets <- 100
lingua <- "pt"
tweetdata = searchTwitter(tema, n = qtd_tweets, lang = lingua)

# Visualizando as primeiras linhas do objeto tweetdata
head(tweetdata)
```

```
## [[1]]
## [1] "gabrielskyz: @tarsisazevedo @raelmax Tu me traiu tarsinho, vim pra bigdata pra sentar com você
##
## [[2]]
## [1] "luansql: Big Data nas redes sociais promete estreitar a relação entre artistas e fãs\n\nhttps://
##
## [[3]]
## [1] "Manifattura40: RT @Postmetria: \\o/ Postmetria na coluna de Tulio Milman na GaúchaZH sobre as 5
##
## [[4]]
## [1] "Postmetria: \\o/ Postmetria na coluna de Tulio Milman na GaúchaZH sobre as 5 startups de Porto A
##
## [[5]]
## [1] "rstatstweet: RT @RosanaFerrero: TOP 37 RECURSOS ONLINE SOBRE DATA SCIENCE + R\n\nhttps://t.co/eQ
##
## [[6]]
## [1] "RosanaFerrero: TOP 37 RECURSOS ONLINE SOBRE DATA SCIENCE + R\n\nhttps://t.co/eQ5SB8WbKL\n#RStudi
```

Etapa 3 - Tratamento dos dados coletados através de text mining

Aqui vamos instalar o pacote tm, para text mining. Vamos converter os tweets coletados em um objeto do tipo Corpus, que armazena dados e metadados e na sequência faremos alguns processo de limpeza, como remover pontuação, converter os dados para letras minúsculas e remover as stopwords (palavras comuns do idioma inglês, neste caso).

```
# Instalando o pacote para Text Mining.
# install.packages("tm")
# install.packages("SnowballC")
library(SnowballC)
library(tm)
```

```
## Loading required package: NLP
```

```
##
## Attaching package: 'NLP'
```

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

```r
# Tratamento (limpeza, organização e transformação) dos dados coletados
tweetlist <- sapply(tweetdata, function(x) x$getText())
tweetlist <- iconv(tweetlist, to = "utf-8", sub="")
tweetlist <- limpaTweets(tweetlist)
tweetcorpus <- Corpus(VectorSource(tweetlist))
tweetcorpus <- tm_map(tweetcorpus, removePunctuation)
tweetcorpus <- tm_map(tweetcorpus, content_transformer(tolower))
tweetcorpus <- tm_map(tweetcorpus, function(x)removeWords(x, stopwords()))

# Convertendo o objeto Corpus para texto plano
termo_por_documento = as.matrix(TermDocumentMatrix(tweetcorpus), control = list(stopwords = c(stopwords
```

Etapa 4 - Wordcloud, associação entre as palavras e dendograma

Vamos criar uma nuvem de palavras (wordcloud) para verificar a relação entre as palavras que ocorrem com mais frequência. Criamos uma tabela com a frequência das palavras e então geramos um dendograma, que mostra como as palavras se relaiconam e se associam ao tema principal (em nosso caso, o termo BigData).

```r
# Instalando o pacote wordcloud
# install.packages("RColorBrewer")
# install.packages("wordcloud")
library(RColorBrewer)
library(wordcloud)

# Gerando uma nuvem palavras
pal2 <- brewer.pal(8,"Dark2")

wordcloud(tweetcorpus,
          min.freq = 2,
          scale = c(5,1),
          random.color = F,
          max.word = 60,
          random.order = F,
          colors = pal2)
```


```
# Convertendo o objeto texto para o formato de matriz
tweettdm <- TermDocumentMatrix(tweetcorpus)
tweettdm
```

```
## <<TermDocumentMatrix (terms: 357, documents: 100)>>
## Non-/sparse entries: 815/34885
## Sparsity           : 98%
## Maximal term length: 15
## Weighting          : term frequency (tf)
```

```
# Encontrando as palavras que aparecem com mais frequência
findFreqTerms(tweettdm, lowfreq = 11)
```

```
##  [1] "big"          "data"         "online"       "science"
##  [5] "artificial"   "evento"       "inteligencia" "por"
##  [9] "que"          "dados"        "para"
```

```
# Buscando associações
findAssocs(tweettdm, 'datascience', 0.60)
```

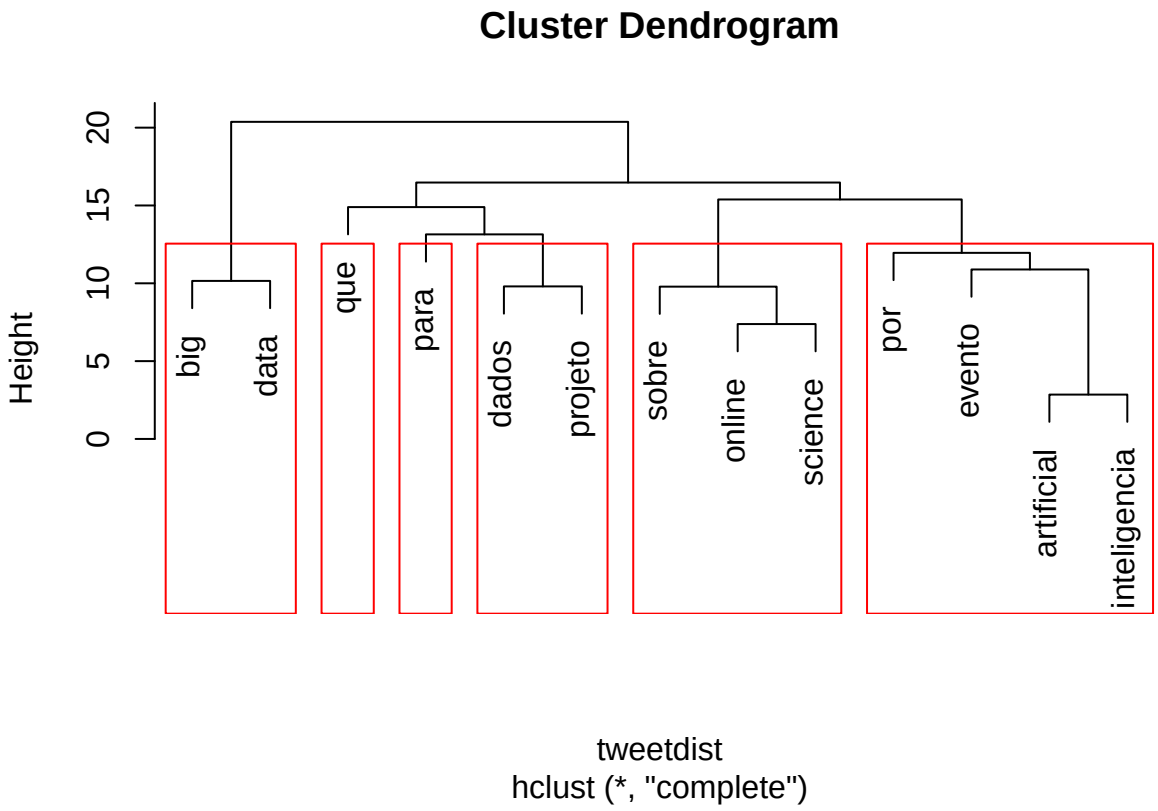
```
## $datascience
## numeric(0)
```

```
# Removendo termos esparsos (não utilizados frequentemente)
tweet2tdm <-removeSparseTerms(tweettdm, sparse = 0.9)

# Criando escala nos dados
tweet2tdmscale <- scale(tweet2tdm)

# Distance Matrix
tweetdist <- dist(tweet2tdmscale, method = "euclidean")
```

```
# Preprando o dendograma
tweetfit <- hclust(tweetdist)

# Criando o dendograma (verificando como as palvras se agrupam)
plot(tweetfit)

# Verificando os grupos
cutree(tweetfit, k = 6)
```

```
##          big         data        sobre       online      science
##            1            1            2            2            2
##   artificial       evento inteligencia          por          que
##            3            3            3            3            4
##        dados      projeto         para
##            5            5            6
```

```
# Visualizando os grupos de palavras no dendograma
rect.hclust(tweetfit, k = 6, border = "red")
```



Etapa 5 - Análise de Sentimento

Agora podemos proceder com a análise de sentimento. Construímos uma função (chamada sentimento.score) e uma lista de palavras positivas e negativas (essas listas acampanham este projeto). Nossa função verifica cada item do conjunto de dados e compara com as listas de palavras fornecidas e a partir daí calcula o score de sentimento, sendo positivo, negativo ou neutro.

```r
# Criando uma função para avaliar o sentimento
# install.packages("stringr")
# install.packages("plyr")
library(stringr)
library(plyr)
```

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

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

```r
sentimento.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)
}

# Mapeando as palavras positivas e negativas
pos = readLines("palavras_positivas.txt")
neg = readLines("palavras_negativas.txt")

# Criando massa de dados para teste
teste = c("Big Data is the future", "awesome experience",
```

```
          "analytics could not be bad", "learn to use big data")

# Testando a função em nossa massa de dados dummy
testesentimento = sentimento.score(teste, pos, neg)
class(testesentimento)
```

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

```
# Verificando o score
# 0 - expressão não possui palavra em nossas listas de palavras positivas e negativas ou
# encontrou uma palavra negativa e uma positiva na mesma sentença
# 1 - expressão possui palavra com conotação positiva
# -1 - expressão possui palavra com conotação negativa
testesentimento$score
```

```
## [1]  0  1 -1  0
```

Etapa 6 - Gerando Score da Análise de Sentimento

Com o score calculado, vamos separar por país, neste caso Canadá e EUA, como forma de comparar o sentimento em regiões diferentes. Geramos então um boxplot e um histograma usando o pacote lattice.

```
# Tweets por país
catweets = searchTwitter("ca", n = 500, lang = "en")
usatweets = searchTwitter("usa", n = 500, lang = "en")

# Obtendo texto
catxt = sapply(catweets, function(x) x$getText())
usatxt = sapply(usatweets, function(x) x$getText())

# Vetor de tweets dos países
paisTweet = c(length(catxt), length(usatxt))

# Juntando os textos
paises = c(catxt, usatxt)

# Aplicando função para calcular o score de sentimento
scores = sentimento.score(paises, pos, neg, .progress = 'text')
```

```
##
  |
  |                                                                      |   0%
  |
  |                                                                      |   1%
  |
  |=                                                                     |   1%
  |
  |=                                                                     |   2%
  |
  |==                                                                    |   2%
  |
  |==                                                                    |   3%
  |
  |==                                                                    |   4%
  |
```

```
|===                                                    |   4%
|
|===                                                    |   5%
|
|====                                                   |   5%
|
|====                                                   |   6%
|
|====                                                   |   7%
|
|=====                                                  |   7%
|
|=====                                                  |   8%
|
|======                                                 |   8%
|
|======                                                 |   9%
|
|======                                                 |  10%
|
|=======                                                |  10%
|
|=======                                                |  11%
|
|=======                                                |  12%
|
|========                                               |  12%
|
|========                                               |  13%
|
|=========                                              |  13%
|
|=========                                              |  14%
|
|=========                                              |  15%
|
|==========                                             |  15%
|
|==========                                             |  16%
|
|===========                                            |  16%
|
|===========                                            |  17%
|
|===========                                            |  18%
|
|============                                           |  18%
|
|============                                           |  19%
|
|=============                                          |  19%
|
|=============                                          |  20%
|
```

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

```
|======================                                |  37%
|
|======================                                |  38%
|
|=======================                               |  38%
|
|=======================                               |  39%
|
|========================                              |  39%
|
|========================                              |  40%
|
|=========================                             |  41%
|
|=========================                             |  41%
|
|==========================                            |  42%
|
|==========================                            |  42%
|
|===========================                           |  43%
|
|===========================                           |  44%
|
|============================                          |  44%
|
|============================                          |  45%
|
|=============================                         |  45%
|
|=============================                         |  46%
|
|=============================                         |  47%
|
|==============================                        |  47%
|
|==============================                        |  48%
|
|===============================                       |  48%
|
|===============================                       |  49%
|
|===============================                       |  50%
|
|================================                      |  50%
|
|================================                      |  51%
|
|================================                      |  52%
|
|=================================                     |  52%
|
|=================================                     |  53%
|
```

```
|==================================                    |  53%
|
|==================================                    |  54%
|
|==================================                    |  55%
|
|===================================                   |  55%
|
|===================================                   |  56%
|
|====================================                  |  56%
|
|====================================                  |  57%
|
|====================================                  |  58%
|
|=====================================                 |  58%
|
|=====================================                 |  59%
|
|=====================================                 |  59%
|
|======================================                |  60%
|
|======================================                |  61%
|
|======================================                |  61%
|
|=======================================               |  62%
|
|=======================================               |  62%
|
|=======================================               |  63%
|
|========================================              |  64%
|
|========================================              |  64%
|
|========================================              |  65%
|
|=========================================             |  65%
|
|=========================================             |  66%
|
|=========================================             |  67%
|
|==========================================            |  67%
|
|==========================================            |  68%
|
|===========================================           |  68%
|
|===========================================           |  69%
|
```

```
|=========================================                |  70%
|
|=========================================                |  70%
|
|=========================================                |  71%
|
|=========================================                |  72%
|
|=========================================                |  72%
|
|=========================================                |  73%
|
|=========================================                |  73%
|
|=========================================                |  74%
|
|=========================================                |  75%
|
|=========================================                |  75%
|
|=========================================                |  76%
|
|=========================================                |  76%
|
|=========================================                |  77%
|
|=========================================                |  78%
|
|=========================================                |  78%
|
|=========================================                |  79%
|
|=========================================                |  79%
|
|=========================================                |  80%
|
|=========================================                |  81%
|
|=========================================                |  81%
|
|=========================================                |  82%
|
|=========================================                |  82%
|
|=========================================                |  83%
|
|=========================================                |  84%
|
|=========================================                |  84%
|
|=========================================                |  85%
|
|=========================================                |  85%
|
```

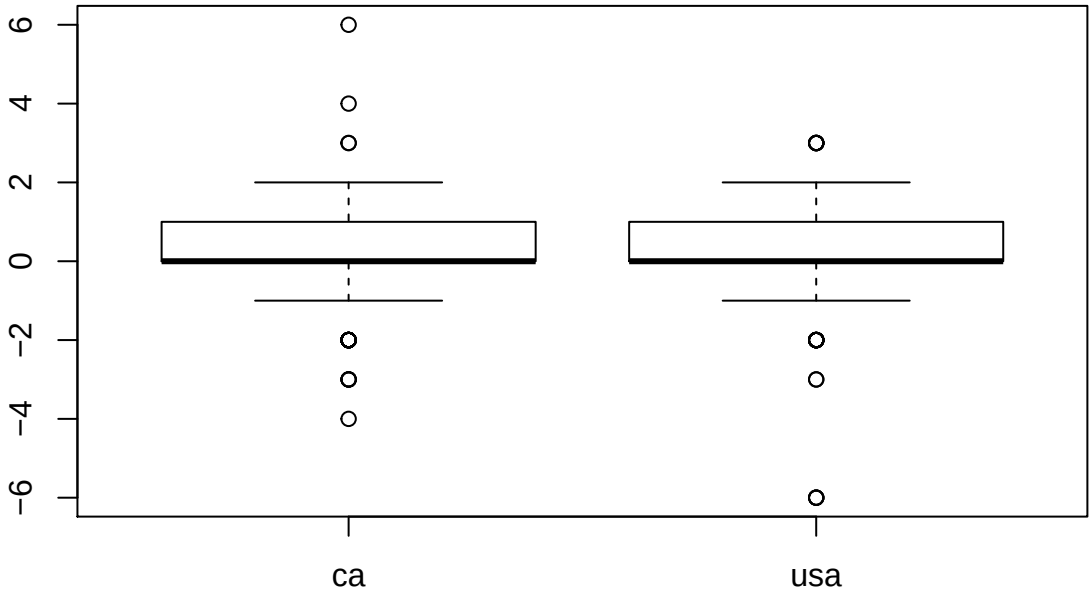
```
  |=========================================================          |  86%
  |
  |=========================================================          |  87%
  |
  |==========================================================         |  87%
  |
  |==========================================================         |  88%
  |
  |===========================================================        |  88%
  |
  |===========================================================        |  89%
  |
  |============================================================       |  90%
  |
  |============================================================       |  90%
  |
  |============================================================       |  91%
  |
  |============================================================       |  92%
  |
  |=============================================================      |  92%
  |
  |=============================================================      |  93%
  |
  |==============================================================     |  93%
  |
  |==============================================================     |  94%
  |
  |==============================================================     |  95%
  |
  |===============================================================    |  95%
  |
  |===============================================================    |  96%
  |
  |================================================================   |  96%
  |
  |================================================================   |  97%
  |
  |================================================================   |  98%
  |
  |================================================================== |  98%
  |
  |================================================================== |  99%
  |
  |===================================================================|  99%
  |
  |===================================================================| 100%
```

```r
# Calculando o score por país
scores$paises = factor(rep(c("ca", "usa"), paisTweet))
scores$muito.pos = as.numeric(scores$score >= 1)
scores$muito.neg = as.numeric(scores$score <= -1)

# Calculando o total
```

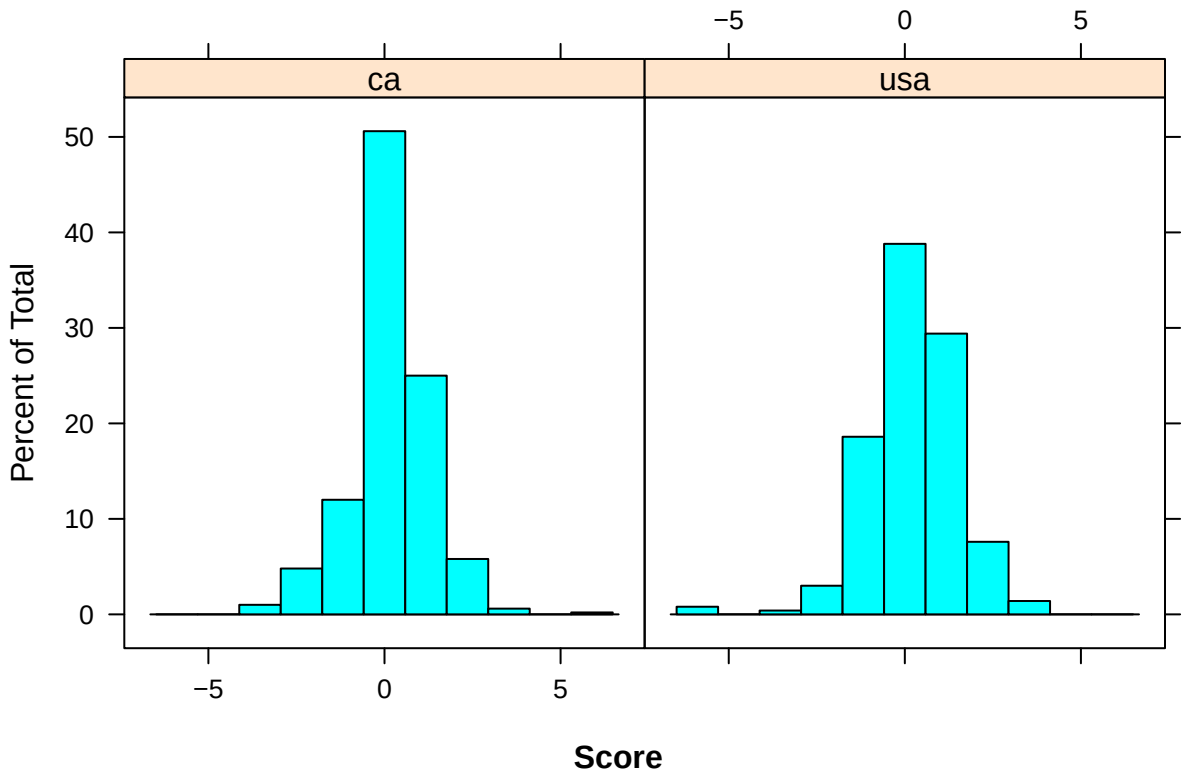
```
numpos = sum(scores$muito.pos)
numneg = sum(scores$muito.neg)

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

```
##
## 1         RT @JackPosobiec: The President of Croatia gave Trump a soccer jersey today \n\nHours late
## 2       RT @JustinTrudeau: Canada will always be there to answer the call for @NATO &amp;amp; help build
## 3                                                                                          RT @talkin
## 4 RT @PMMackenzieKing: \U0001f648 Remember when Stephen Harper received a failing grade on military s
## 5          RT @adnilhoom: CA Gov Jerry Obama Brown  "colludes" with all lawbreaking Dems like Eric H
## 6                                                                  RT @Marylaw55555: @andibeth012 @Bl
##   score paises muito.pos muito.neg
## 1     1     ca         1         0
## 2     1     ca         1         0
## 3     0     ca         0         0
## 4     0     ca         0         0
## 5     0     ca         0         0
## 6     0     ca         0         0
```

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

# Gerando um histograma com o lattice
# install.packages("lattice")
library("lattice")
```




```
histogram(data = scores, ~score|paises, main = "Análise de Sentimentos", xlab = "", sub = "Score")
```



Extra

Usando Classificador Naive Bayes para analise de sentimento

Aqui faremos a análise de sentimento de forma semelhante ao visto anteriormente, mas usando o pacote sentiment. Este pacote foi descontinuado do CRAN, pois não será mais atualizado, mas ainda pode ser obtido através do link de archives do CRAN. Os pacotes estão disponíveis junto com os arquivos do projeto e o procedimento de instalação está descrito abaixo.

```
# install.packages("/opt/DSA/Projetos/Projeto01/Rstem_0.4-1.tar.gz", repos = NULL, type = "source")
# install.packages("/opt/DSA/Projetos/Projeto01/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
```

Coletando Tweets

A coleta dos tweets é feita utilizando a função searchTwitter() do pacote twitteR.

```
# Coletando os tweets
tweetpt = searchTwitter("bigdata", n = 1500, lang = "pt")

# Obtendo o texto
tweetpt = sapply(tweetpt, function(x) x$getText())
```

Limpando, Organizando e Transformando os Dados

Aqui expressões regulares, atraves da função gsub() para remover caracteres que podem atrapalhar o processo de análise.

```
# Removendo caracteres especiais
tweetpt = gsub("(RT|via)((?:\\b\\W*@\\w+)+)", "", tweetpt)
# Removendo @
tweetpt = gsub("@\\w+", "", tweetpt)
# Removendo pontuação
tweetpt = gsub("[[:punct:]]", "", tweetpt)
# Removendo digitos
tweetpt = gsub("[[:digit:]]", "", tweetpt)
# Removendo links html
tweetpt = gsub("http\\w+", "", tweetpt)
# Removendo espacos desnecessários
tweetpt = gsub("[ \t]{2,}", "", tweetpt)
tweetpt = gsub("^\\s+|\\s+$", "", tweetpt)

# Criando função para tolower
try.error = function(x)
{
  # Criando missing value
  y = NA
  try_error = tryCatch(tolower(x), error=function(e) e)
  if (!inherits(try_error, "error"))
    y = tolower(x)
  return(y)
}

# Lower case
tweetpt = sapply(tweetpt, try.error)

# Removendo os NAs
tweetpt = tweetpt[!is.na(tweetpt)]
names(tweetpt) = NULL
```

Classificador Naive Bayes

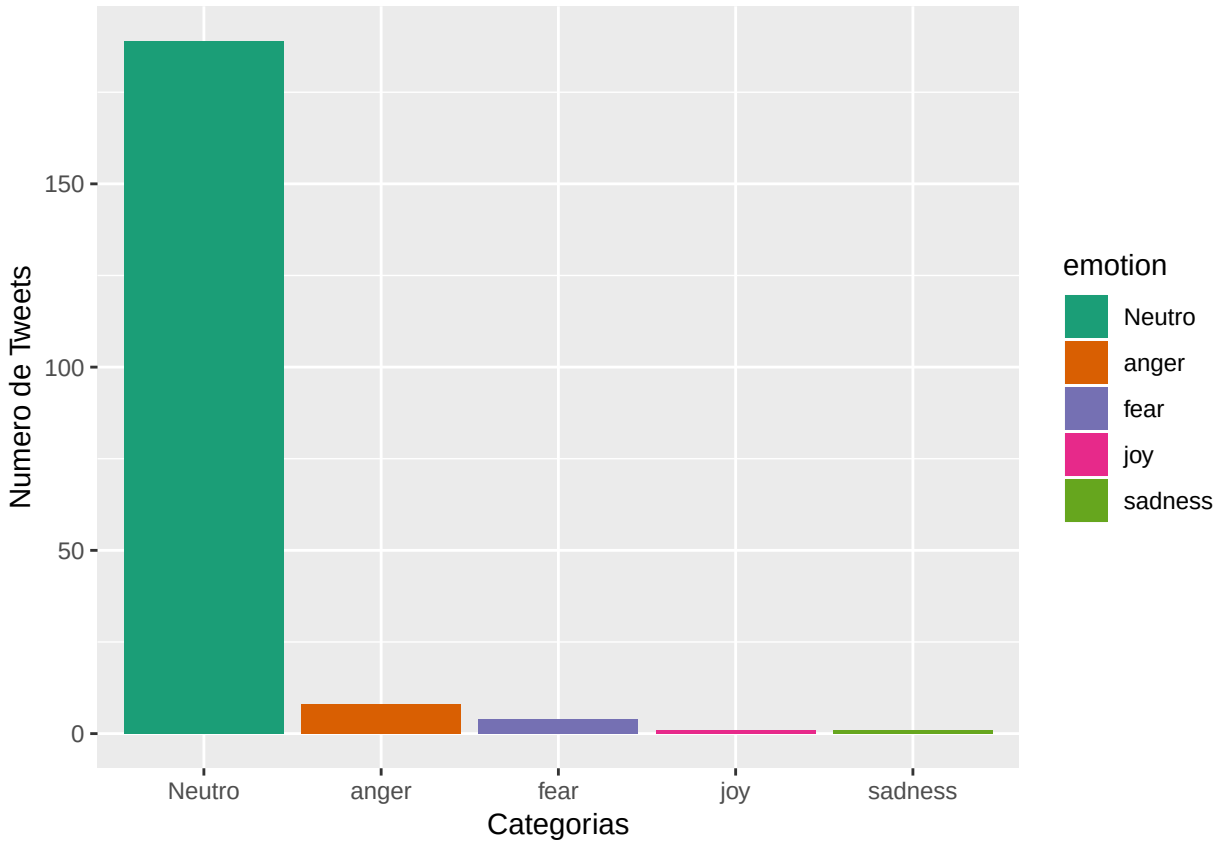
Utilizamos as funções classify_emotion() e classify_polarity() do pacote sentiment, que utilizam o algotimo Naive Bayes para a análise de sentimento. Neste caso, o próprio algoritmo faz a classificação das palavras e não precisamos criar listas de palavras positivas e negativas.

```
# Classificando emocao
class_emo = classify_emotion(tweetpt, algorithm = "bayes", prior = 1.0)
emotion = class_emo[,7]

# Substituindo NA's por "Neutro"
emotion[is.na(emotion)] = "Neutro"

# Classificando polaridade
class_pol = classify_polarity(tweetpt, algorithm = "bayes")
polarity = class_pol[,4]

# Gerando um dataframe com o resultado
sent_df = data.frame(text = tweetpt, emotion = emotion,
                     polarity = polarity, stringsAsFactors = FALSE)

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

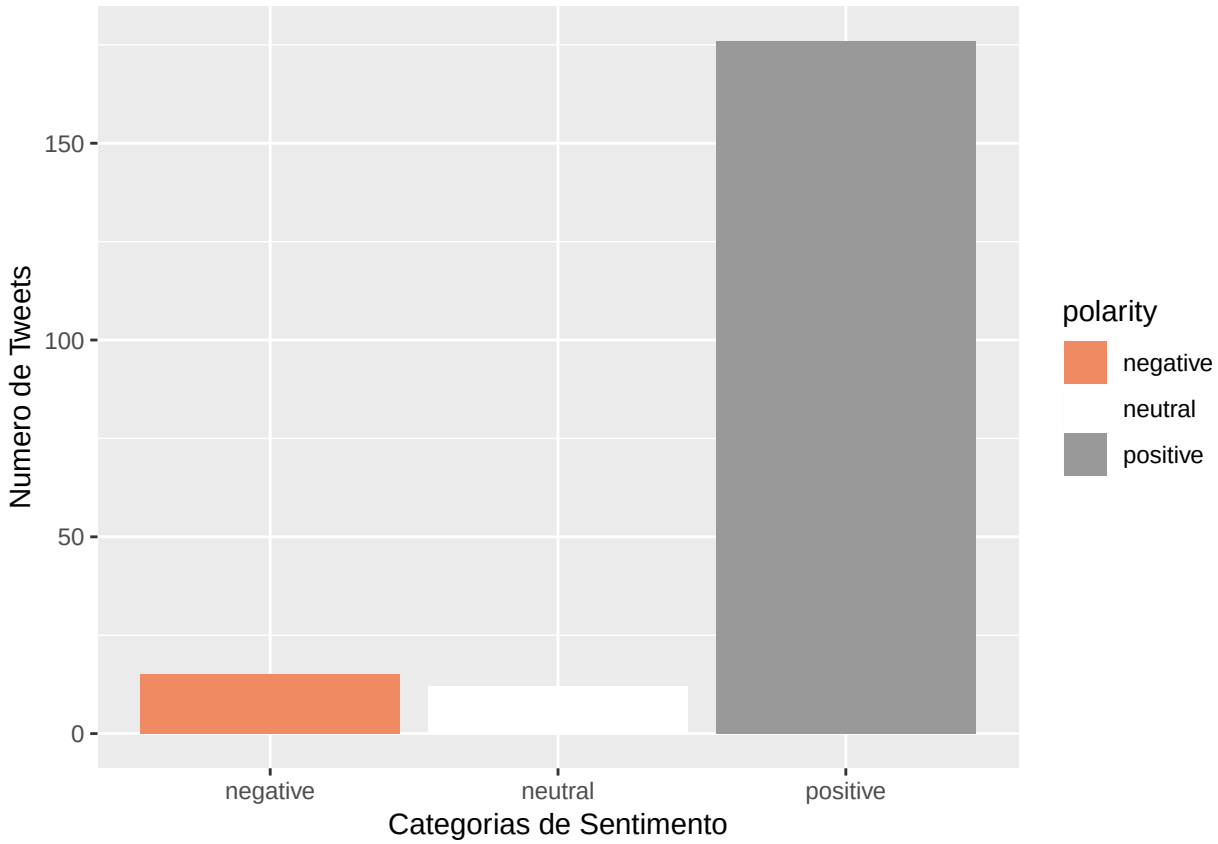
Visualização

Finalmente, usamos o ggplot2 para visualizar os resultados.

```
# Emoções encontradas
ggplot(sent_df, aes(x = emotion)) +
  geom_bar(aes(y = ..count.., fill = emotion)) +
  scale_fill_brewer(palette = "Dark2") +
  labs(x = "Categorias", y = "Numero de Tweets")
```




```
# Polaridade
ggplot(sent_df, aes(x = polarity)) +
  geom_bar(aes(y = ..count.., fill = polarity)) +
  scale_fill_brewer(palette = "RdGy") +
  labs(x = "Categorias de Sentimento", y = "Numero de Tweets")
```



Fim

www.datascienceacademy.com.br