Text/Opinion Mining no Twitter: Prisao do Presidente da Dolly

Rafael Bicudo

10 de maio de 2018

Projeto 1 - Analise de Sentimentos em Redes Sociais - Dolly

O objetivo deste trabalho e captutar dados da rede social Twitter e realizar analises de texto e sentimento. Escolheu-se 10/05/2018 por ser a data da prisão preventiva do dono da empresa de refrigerantes 'Dolly' (Laerte Codonho), e em que a palavra "dolly" manteve-se como um dos assuntos mais citados na plataforma (fonte: https://trends24.in/brazil/), portanto sendo um bom referencial para a proposta do projeto. Para sua execucao, diversos pacotes devem ser instalados e carregados.

Todo o projeto sera descrito de acordo com suas etapas. Primeiro, apos a coleta dos tweets e limpeza, far-se-a uma analise preliminar dos termos e suas relacoes estatisticas para, em seguida, prosseguir com a analise de sentimento.

```
library(rtweet)
library(dplyr)
library(knitr)
library(rmarkdown)
```

Etapa 1 - Autenticacao

Abaixo, montam-se as variaveis contendo as chaves necessarias a autenticacao e conexao. Lembre-se que precisa ter uma conta criada no Twitter e se criar uma aplicacao. Para fins de privacidade, os dados referentes a essas chaves foram omitidos.

```
# Criando autenticacao no Twitter
app_rbr <- "xxxxx"
consumer_key_rbr <- "xxxxx"
consumer_secret_rbr <- "xxxxx"
```

Etapa 2 - Conexao

Aqui, testa-se a conexÃ£o e se capturam os tweets. Quanto maior sua amostra, mais precisa sera a analise, assim a quantidade escolhida foi de 1000 tweets e, da mesma forma como descrito no inicio do trabalho, o objeto de procura 'dolly'. (Obs.: Como a Rest API do Twitter possui algumas limitacoes, das quais nao reaver dados com mais de 2 semanas, após se fazer a pesquisa, os dados dos tweets brutos foram salvos no arquivo 'tweetsSearch_dolly.csv' para se manter a reprodutibilidade)

```
# Realizando a coleta dos tweets (10/05/18)
tema <- "Dolly"
qtd_tweets <- 1000
lingua <- "pt"
#tweetsSearch <- search_tweets(tema, n = qtd_tweets, lang = lingua, include_rts = FALSE)
tweetsSearch <- as.tbl(read.csv('tweetsSearch_dolly.csv', stringsAsFactors = F))
# Selecionando o vetor de textos para analise e criando um indicador para cada tweet
tweets_text <- tweetsSearch %>%
```

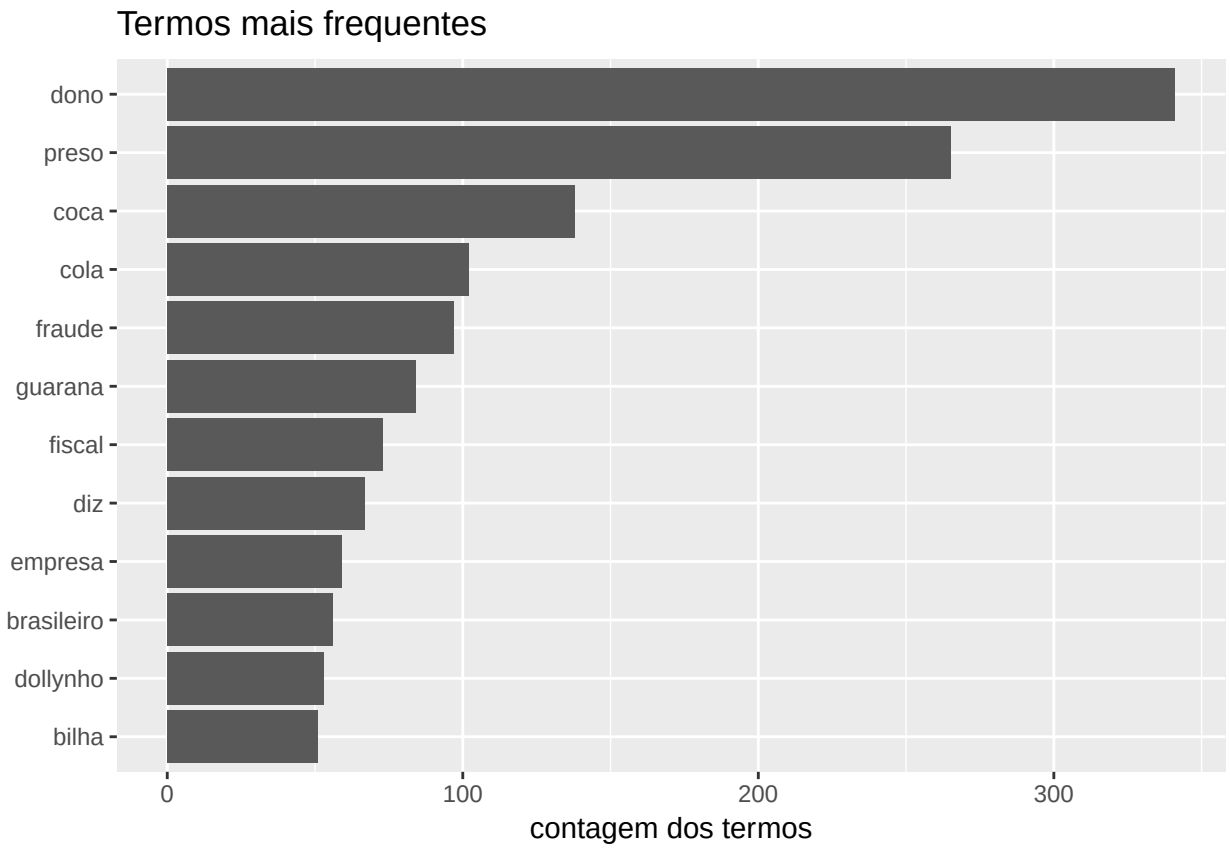
```
  select(text) %>%
  mutate(tweet_id = row_number()) %>%
  select(tweet_id, text)
```

Etapa 3 - Tratamento dos dados coletados

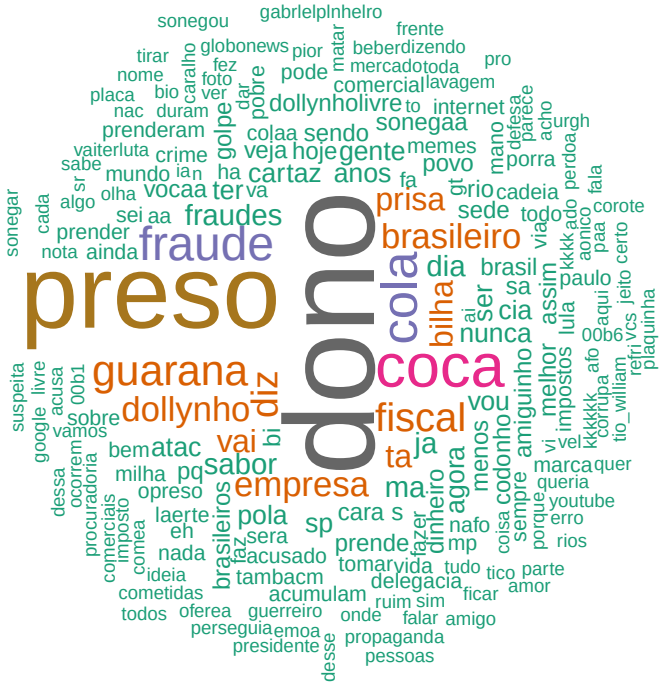
Aqui, instalam-se os pacotes tidytext, stringr e stopwords para text mining. Comecando pela decodificacao de alguns encodings comuns de arquivos retirados diretamente da web, parte-se para a transformacao dos tweets coletados em tokens, processo que transforma nosso dataset de tweetsem um com 1 termo por observacao/linha, e realiza algumas etapas de limpez; para, na sequencia, fazerem-se as transformacoes de limpeza remanescentes específicas (nomes de usuários, alguns resíduos, dentre outros) e, através de analises suscetivas dos termos mais frequentes, a criacao das stopwords. Por fim acontecem duas visualizacoes para ilustrar os termos mais usados: um gráfico de barras para contagem, e uma nuvem de palavras (wordcloud),para ilustrar os termos com maior frequencia de acordo com seu tamanho e cor.

```
## Etapa 2: Tratamento dos dados

# Ativacao dos pacotes necessarios ao Text Mining e manipulacao dos dados
library(tidytext)
library(stringr)
library(stopwords)
library(ggplot2)
library(wordcloud)
library(RColorBrewer)

# Remocao de problemas de encoding comuns ao fazer text mining pela internet

tweets_text$text <- tweets_text$text %>%
  iconv(to = "ASCII//TRANSLIT") %>% # Corrige encoding do texto do post (ao menos uma parte deles)
  iconv(sub="", 'UTF-8', 'ASCII') # Remove emojis

# Criacao dos tokens e limpeza final dos dados

tweets_tidy <- tweets_text %>%
  unnest_tokens(word, text) %>% # Transformando texto em tokens
  filter(!word %in% c("https", "t.co", "amp", 'http', 'htt','bmr9ju8t8y', 'fwplgwxu4o', 'f0', 'es','009
         !word %in% tolower(tweetsSearch$screen_name), # Removendo nome de usuarios
         !grepl("^\\d+$", word)) # Removendo numeros

# Definicao das stopwords
stopwords_assunto <- c('nao', 'sao', 'pra', 'q', 'dolly','r', 'u', 'refrigerante', 'refrigerantes')
stopwords_tweets <- data_frame(word = c(stopwords('portuguese'), stopwords_assunto))

# Termos mais frequentes
tweets_tidy %>%
  anti_join(stopwords_tweets) %>%
  count(word, sort = TRUE) %>%
  filter(n > 50) %>%
  mutate(word = reorder(word, n)) %>%
  ggplot(aes(word, n)) +
  geom_col() +
  xlab(NULL) +
  coord_flip() +
```

```
  labs(title = 'Termos mais frequentes', y = 'contagem dos termos')
```




```
# Gerando uma nuvem palavras
tweets_tidy %>%
  anti_join(stopwords_tweets) %>%
  count(word) %>%
  with(wordcloud(word, n, min.freq = 5,
                 max.words = 200, random.order = FALSE, rot.per = 0.35,
                 colors = brewer.pal(8, "Dark2")))
```



Atraves da observacao dos grafico, percebe-se a quantidade alta de termos relacionados as noticias em pauta do dia (dono, preso, fraude, fiscal, coca, cola), confirmando a suposicao inicial de que alavancagem do assunto (dolly) em relacao aos termos mais usados deveu-se a prisao do dono da marca e a grande repercussao publica do evento.

Etapa 4 - Associacoes estatísticas entre a frequencia das palavras por termo e dendograma

Cria-se, agora, com o auxilio do pacote tm, gera-se uma TDM, ou Matriz de Termos por Documento, com o intuito de se verificar relacoes estatísticas em relação a quatidade de termos em cada tweet, terminando com a visualização de um processo de clusterizacao hierarquica (dendograma).

```
# Carregado pacotes necessario
library(tm)

# Criacao do Corpus e TDM para analise estatistica
tweets_tdm <- tweets_tidy %>%
  anti_join(stopwords_tweets) %>%
  group_by(tweet_id, word) %>%
  summarise(count = n()) %>%
  cast_tdm(document = tweet_id, term = word, value = count)

# Encontrando as palavras que aparecem com maior frequencia por termo
findFreqTerms(tweets_tdm, lowfreq = 90)
```

```
## [1] "coca"   "cola"   "fraude" "preso"  "dono"
```

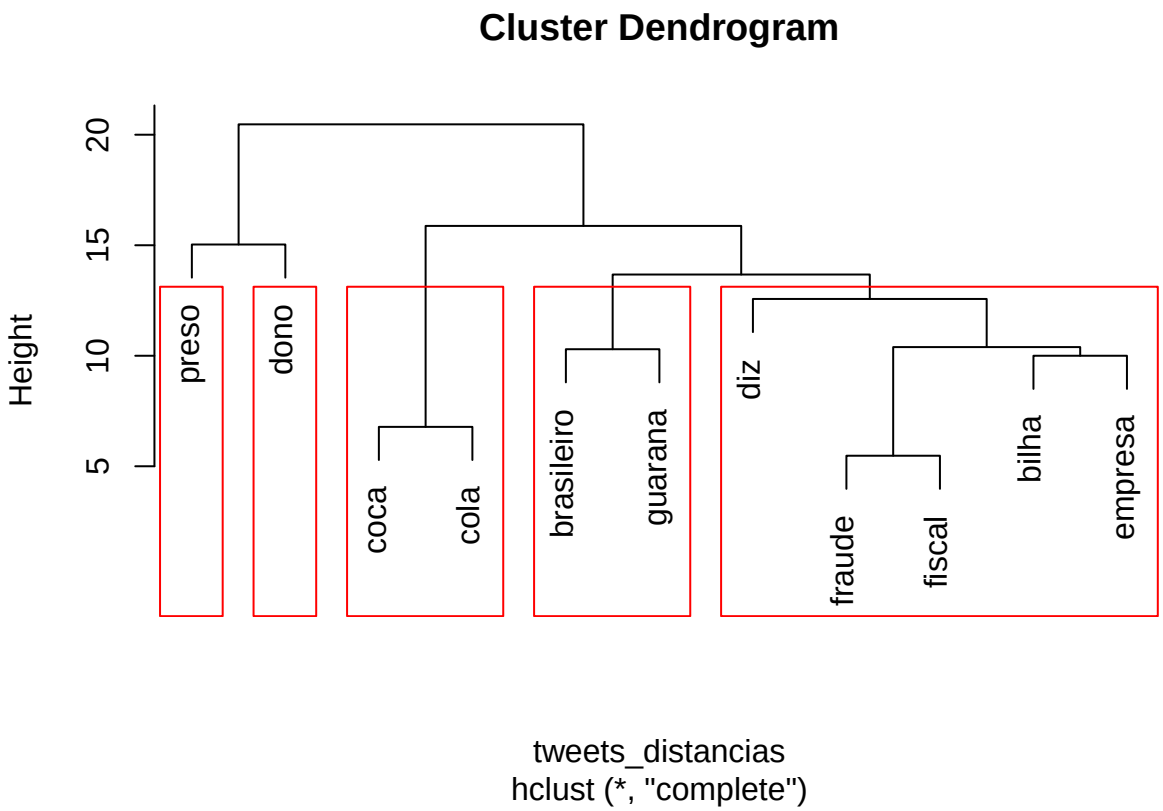
```r
# Buscando associacoes de palavras intra-termos
findAssocs(tweets_tdm, c('dono'), c(0.30))
```

```
## $dono
##  preso fiscal fraude
##   0.51   0.35   0.31
```

```r
# Removendo termos esparsos (nao utilizados frequentemente)
tweets_maisUsados <-removeSparseTerms(tweets_tdm, sparse = 0.95)

# Matriz de distancias
tweets_df = as.data.frame(as.matrix(tweets_maisUsados))
tweets_distancias <- dist(tweets_df)

# Realizando a clusterizacao hierarquica dos dados
tweets_hclust <- hclust(d = tweets_distancias)

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

# Verificando os grupos
cutree(tweets_hclust, k = 5)
```

```
##       coca       cola      bilha     fraude      preso       dono
##          1          1          2          2          3          4
##    empresa     fiscal brasileiro    guarana       diz
##          2          2          5          5          2
```

```r
# Visualizando os grupos de palavras no dendograma
rect.hclust(tweets_hclust, k = 5, border = "red")
```



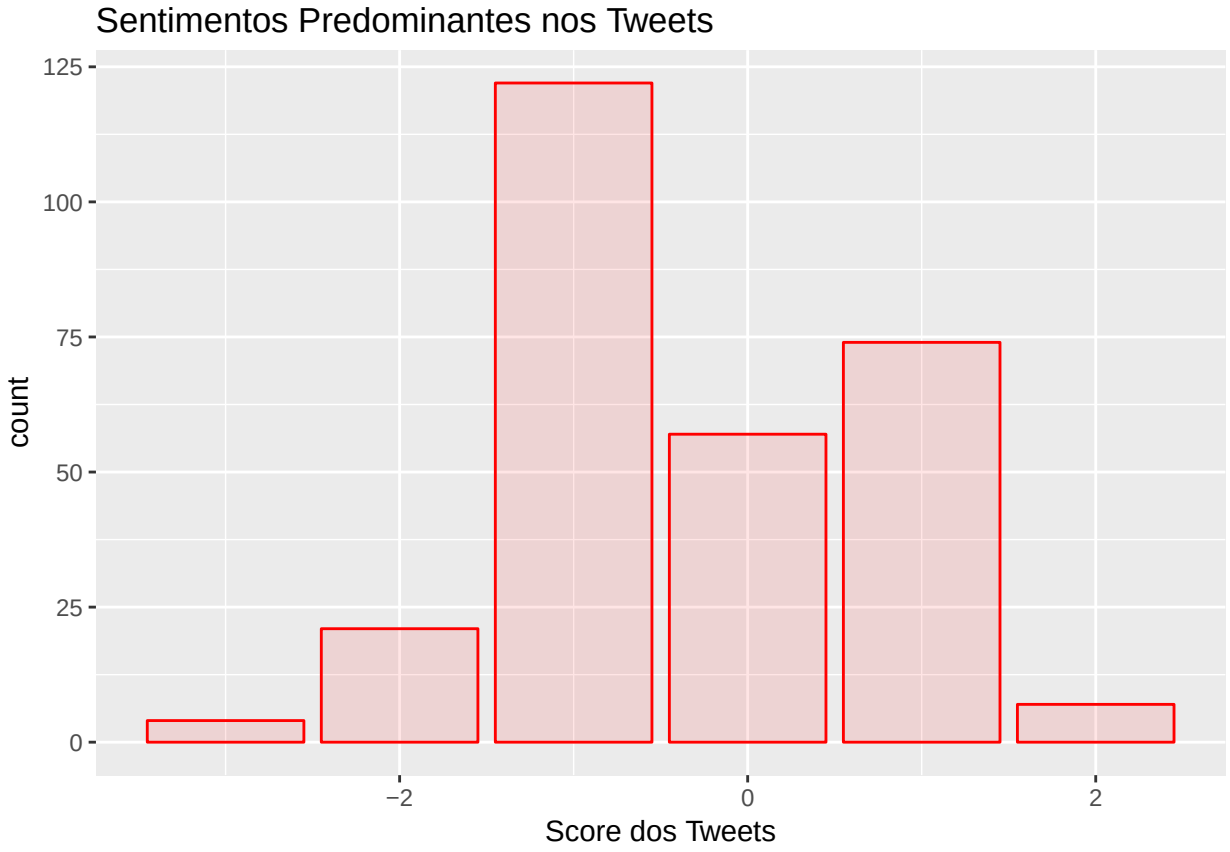
Um fato interessante e a altissima frequencia dos termos “coca” e “cola”, um forte indicador da alta absorcao e aceitacao popular da foto do momento da prisao (em que o acusado segura um cartaz com os dizeres “Preso pela Coca Cola”). Outro fato de importancia e a visualizacao da clusterizacao hierarquica das frequencias: tirando os tweets com maior enfoque no produto e na foto supracitada, os 2 maiores supostos grupos focam-se no fato da prisão em si, e no motivo da acusação (fraude fiscal).

Etapa 5 - Analise de Sentimento

Por ultimo, pode-se proceder a analise de sentimento. A partir do pacote lexiconPT, desenvolvido por um brasileiro e reunindo os principais lexicos nacionais, gera-se uma lista de palavras positivas e negativas a partir da escolha de um dos integrantes do pacote. Apos se atribuir a cada termo uma pontuacao positiva, negativa ou neutra(1, -1, 0, respectivamente), realiza-se uma uma comparacao da distribuicao do sentimento predominate por tweet, e uma comparacao da contagem dos termos mais usados por sentimento.

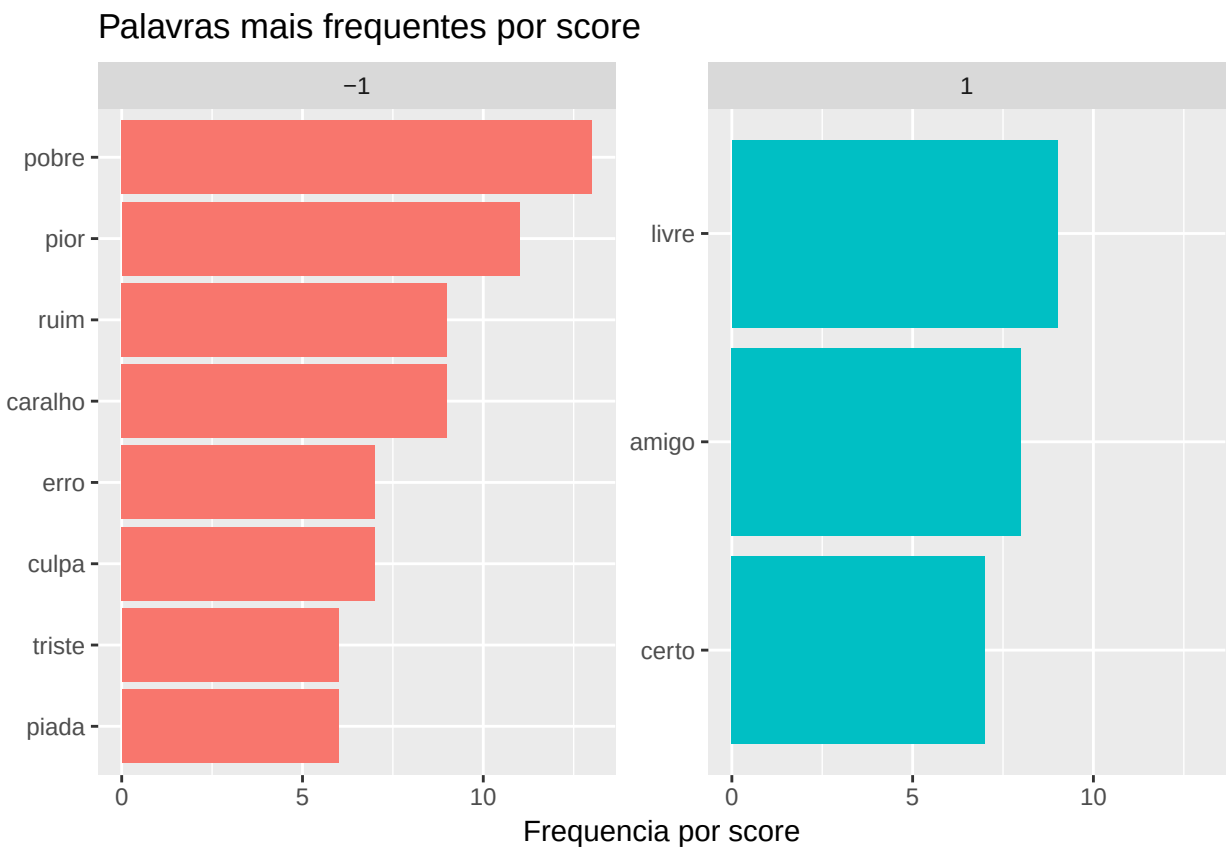
```
## Analise de Sentimento (Opinion Mining)

# Carregando pacotes necessarios
# devtools::install_github("sillasgonzaga/lexiconPT")
library(lexiconPT)
library(tidyr)

# Carregando lexico escolhido
data("sentiLex_lem_PT02")
sentiLex <- as.tbl(sentiLex_lem_PT02)
sentimentos <- sentiLex %>%
  select(word = term, polarity)
```

```r
# Calculando o sentimento de cada palavra
tweets_sentimentos <- tweets_tidy %>%
  filter(!word %in% c('preso', 'fraude', 'acusado', 'imposto', 'acusado')) %>% # excluem-se essas palav
  inner_join(sentimentos)

# Plotando a distribuicao do sentimento agregado dos tweets
tweets_sentimentos %>%
  group_by(tweet_id) %>%
  summarise(sentimento_tweet = sum(polarity)) %>%
  ggplot(aes(sentimento_tweet)) +
  geom_bar(fill = 'red', colour = 'red', alpha = .1) +
  labs(title = 'Sentimentos Predominantes nos Tweets', x = 'Score dos Tweets')
```




```r
# Analisando as palavras mais frequentes por sentimento
tweets_sentimentos %>%
  count(word, polarity, sort = TRUE) %>%
  filter(n > 5, !polarity == 0) %>%
  mutate(word = reorder(word, n)) %>%
  ggplot(aes(word, n, fill = as.factor(polarity))) +
  geom_col() +
  xlab(NULL) +
  coord_flip() +
  facet_wrap(~ polarity, scales = "free_y") +
  labs(title = 'Palavras mais frequentes por score',
       y = "Frequencia por score",
       x = NULL) +
```

```
  coord_flip() +
  theme(legend.position = "none")
```



Por fim, estas ultimas analises graficas permitem concluir por uma tendencia maior a aceitacao negativa do publico com a prisao, provavelmente por possiveis temores em relacao ao futuro da marca assim como se pode ver nas palavras mais frequentes por score: um uso muito alto de termos como “pobre”, “triste”, “ruim”, dentre outros, indicando uma predominancia do sentimento de infelicidade e descontentamento com a acao.

Fim

Obrigado!