# Quanteda and Twitter 2

Rodrigo Esteves de Lima-Lopes
State University of Campinas
rll307@unicamp.br

## Contents



## 1 Introduction

Quanteda is a package for managing and analyse text quantitatively. It is quite easy to use and will bring us a number of interesting functions.

### 1.1 You will need:

1. The package `Quanteda`, which can be installed using RStudio
2. The package `rtweet`, we installed last tutorial.
3. Package `DT` for viewing the KWIC inside R.
4. `Ggplot` for viewing some graphics

### 1.2 Data

We are going to use the same data we have used in the previous tutorials

## 2 Making some comparissions

First we need to save our texts as vectors

```
boulos.v <- boulos$text
covas.v <- covas$text
```

Kwic in the terminal

```
kwic(boulos.v,"prefeitura")
```

Kwic in the Viewer (it views but does not save)

```
kwic(covas.v,"prefeitura")%>%View()
```

Now, saving as a variable

```
prefeitura.b <- kwic(boulos.v,"prefeitura")
```

This command can also be applied to a corpus:

```
kwic(boulos.corpus,"prefeitura")%>%View()
```

Now lets keep on our analysis. We can use the same DFMs we created before, but I will create a couple just to apply some new commands

First step now is to make all lower caps:

```
boulos.lower.v <- char_tolower(boulos.v)
covas.lower.v <- char_tolower(covas.v)
```

Now, we create a lower character vector

```
boulos.word.v <- tokens(boulos.lower.v,
                        remove_punct = TRUE) %>% as.character()
covas.word.v <- tokens(covas.lower.v,
                       remove_punct = TRUE) %>% as.character()
```

So now, we have our second DFM:

```
boulos.dfm.2 <- dfm(boulos.lower.v,
                    remove_punct = TRUE,
                    remove = stopwords("portuguese"))


covas.dfm.2 <- dfm(covas.lower.v,
                    remove_punct = TRUE,
                    remove = stopwords("portuguese"))
```

Now we are creating our word list

```
boulos.wl<-textstat_frequency(boulos.dfm.2)
View(boulos.wl)

covas.wl<-textstat_frequency(covas.dfm.2)
View(covas.wl)
```

Lets us plot a single candidate at a time:

```
library(ggplot2)
theme_set(theme_minimal())
textstat_frequency(covas.dfm.2, n = 50) %>%
  ggplot(aes(x = rank, y = frequency)) +
  geom_point() +
  labs(x = "Frequency rank", y = "Term frequency")
```

```
theme_set(theme_minimal())
textstat_frequency(boulos.dfm.2, n = 50) %>%
  ggplot(aes(x = rank, y = frequency)) +
  geom_point() +
  labs(x = "Frequency rank", y = "Term frequency")
```

The results are:

Quanteda makes key wordlists. Keywords are a comparison between two groups of texts. One is the reference and tells me the statistical baseline for comparison, the other is my research text (or texts). Here, for the

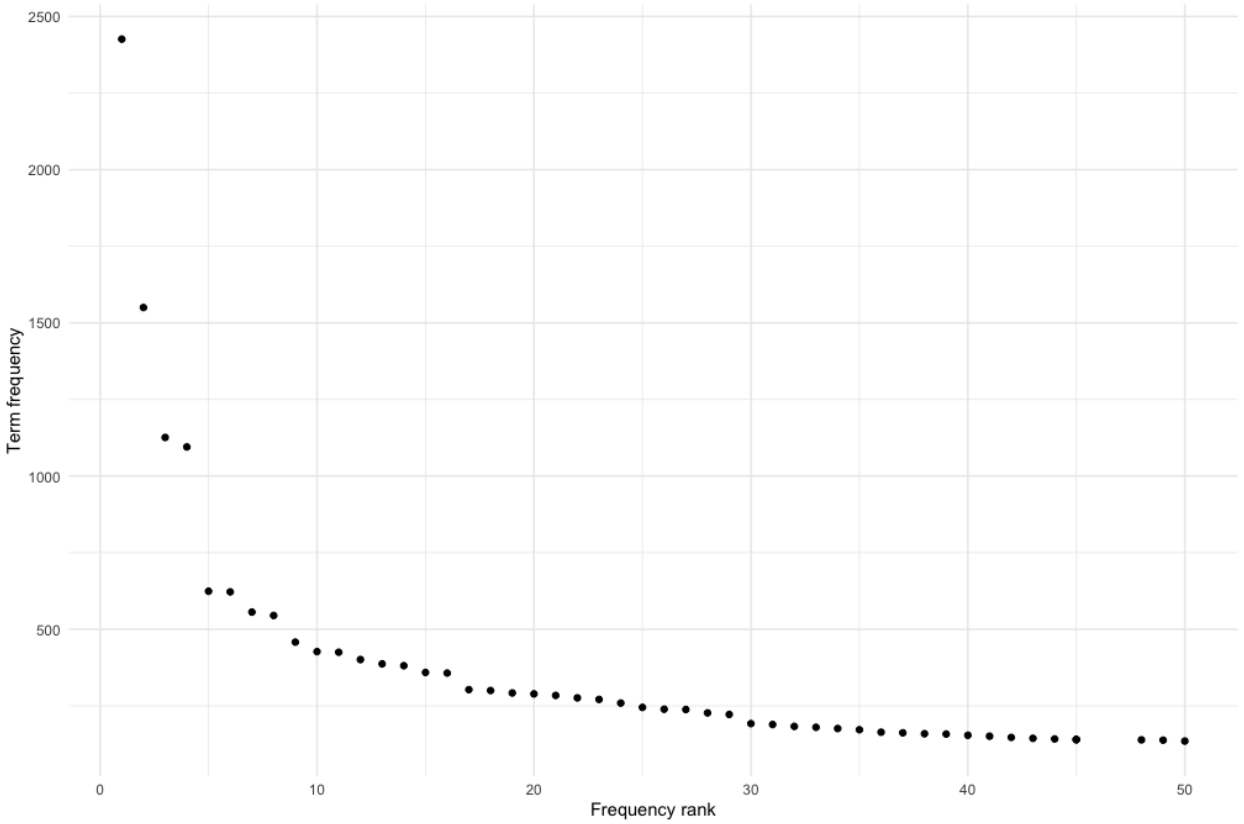


Figure 1: Covas' wordlist

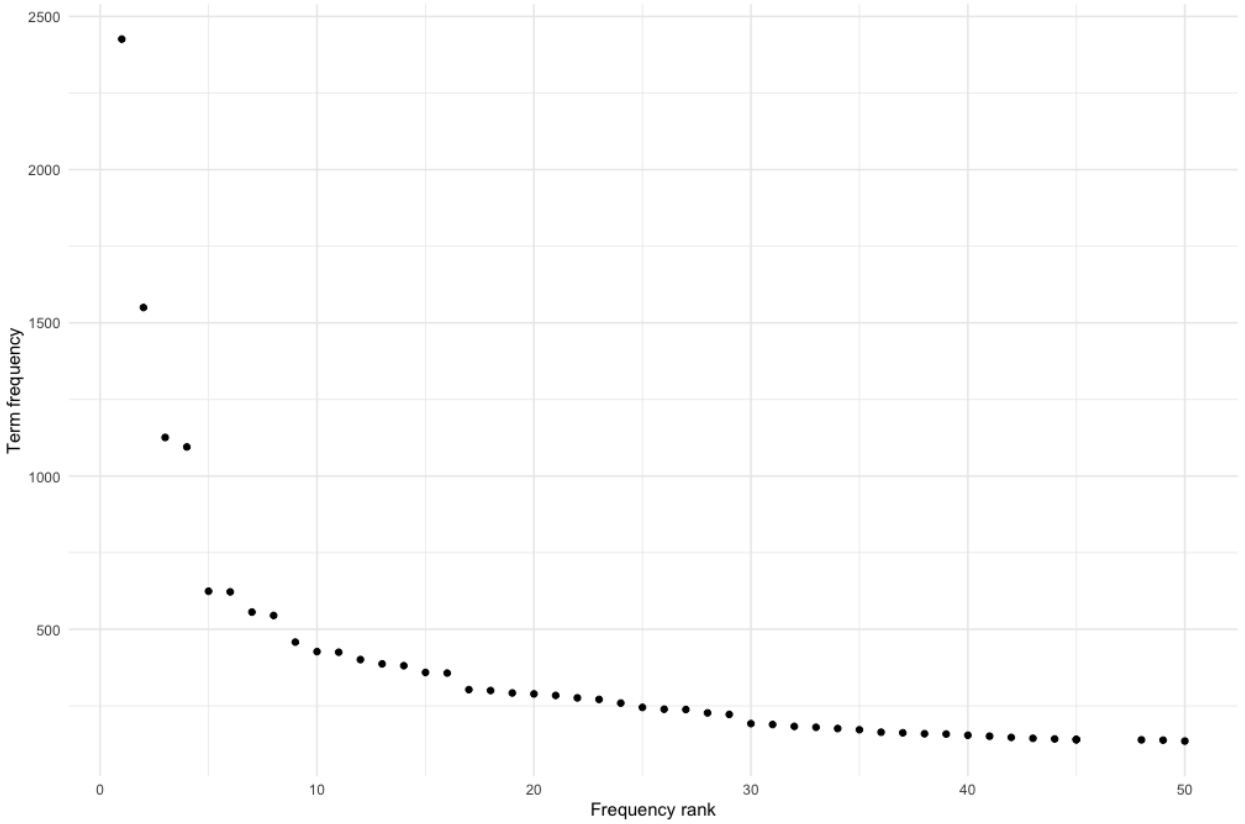


Figure 2: Boulos' wordlist

sake of the exercise, we will compare Boulos and Covas.

```
all.dfm3 <- dfm(all.corpora, groups = "screen_name", remove = stopwords("portuguese"),
                remove_punct = TRUE)

keyness <- textstat_keyness(all.dfm3, target = "brunocovas")
textplot_keyness(keyness)
```

The result is something like

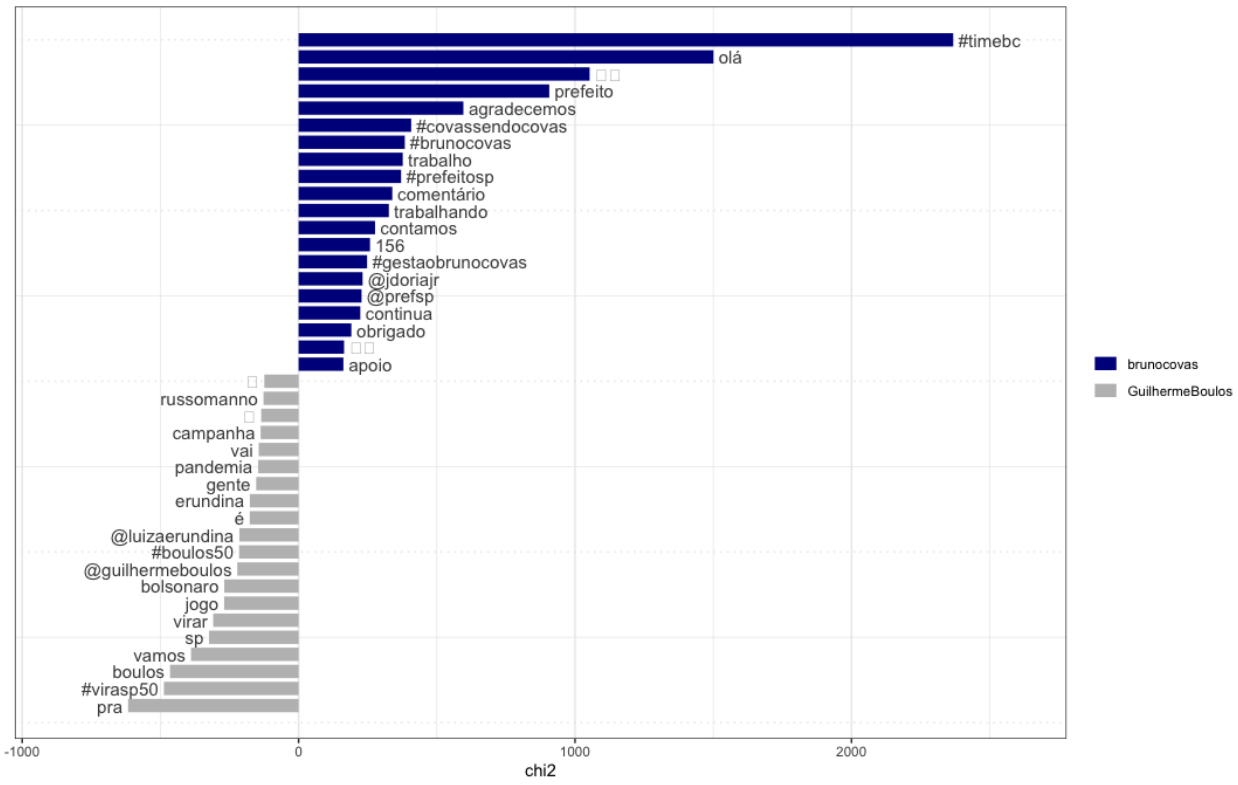


Figure 3: Keywords comparisson

## 3 Some collocates

The command `textstat_collocations` calculates n-grams based on a corpus previously created. Unfortunately, for the time being the only calculation possible is `lambda`.

```
tri.grams<-textstat_collocations(
  boulos.corpus,
  method = "lambda",
  size = 3,
  min_count = 5,
  smoothing = 0.5,
  tolower = TRUE)
```

The result is something similar to:

```r
head(tri.grams)
```

```
##       collocation count count_nested length    lambda        z
## 1     não para de     8            0      3 11.117795 5.407809
## 2    para mais ou    14            0      3  7.132262 4.938017
## 3  apoio a boulos    12            0      3  5.054780 4.731241
## 4 maior cidade do    28            0      3  5.089345 4.534803
## 5        esse é o    30            0      3  3.613462 4.501427
## 6        essa é a    16            0      3  3.445065 4.434219
```