user_stats

October 30, 2018

```
Parsed with column specification:
cols(
  name = col_character(),
  id = col_double(),
  total_followers = col_integer(),
  total_following = col_integer(),
  total_posts = col_integer(),
  total_likes_sum = col_integer(),
  total_likes_mean = col_double(),
  total_likes_median = col_double(),
  total_retweets_sum = col_integer(),
  total_retweets_mean = col_double(),
  total_retweets_median = col_double(),
  total_replies_sum = col_integer(),
  total_replies_mean = col_double(),
  total_replies_median = col_double(),
  posts_count = col_integer()
)
```

1 User Stats - Data Gathering and Analysis

1.1 Step 1

A number n of tweets is requested to the Twitter User Timeline API. The Twitter API Access methods and data retrieval is provided by the python package birdy. In order to generate a request to the Twitter API, the users IDs and the number n of posts are provided to those methods.

As a result, a JSON file was generated for each user containing the n latest posts.

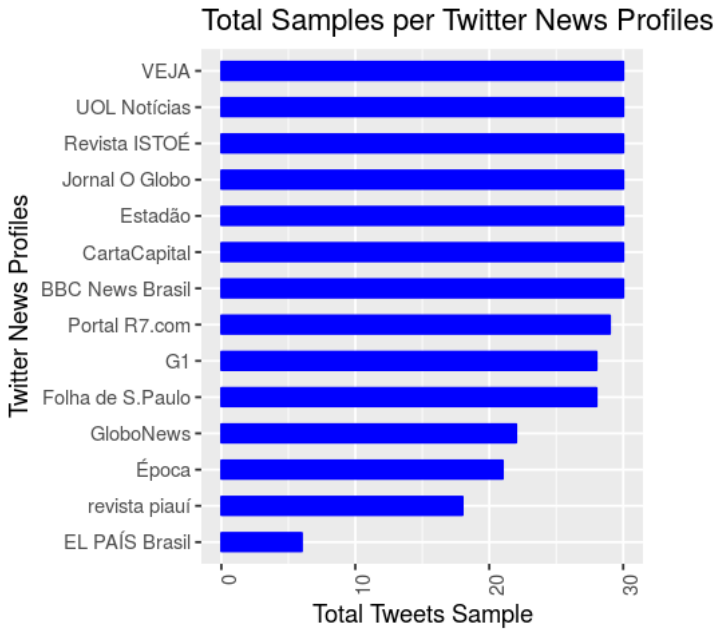
1.2 Step 2

Due to the API limitations, the number of replies is not provided for each tweet. In order to bypass this limitation, the tweet page was scraped using the package Selenium. To handle the case where the content of the tweet page we want to crawl is being added to the page dinamicaly via JavaScript, we used the browser provided by this package to capture the source code after all the content is loaded.

After scraping each tweet page, a css file was created with all the users profile and tweets data and inputed into this notebook to generate the graphs.

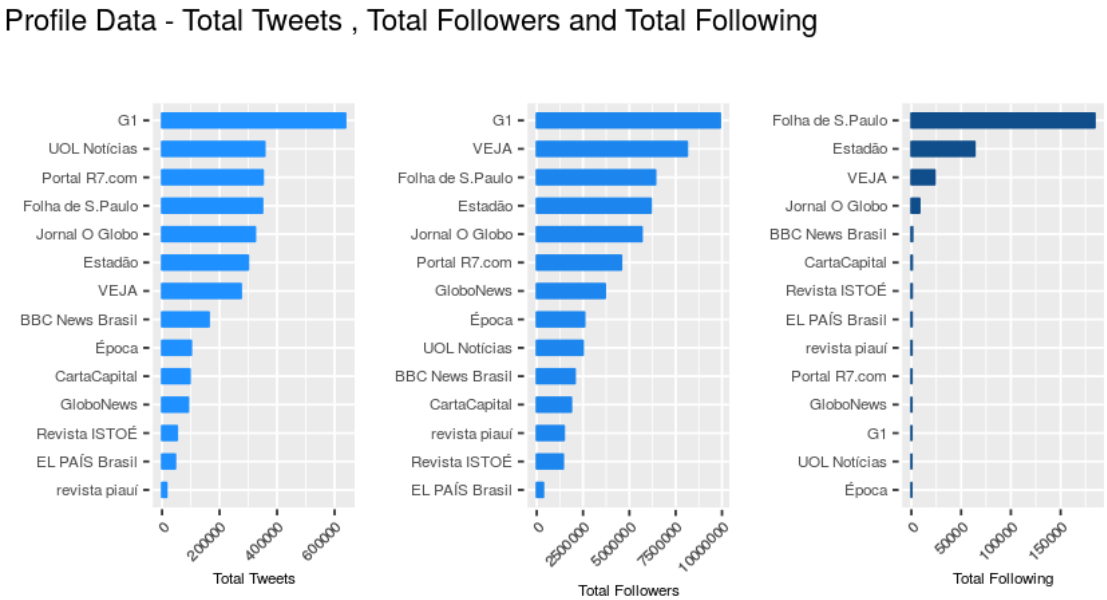
2 Tweets Sample

2.1 Total Samples per user



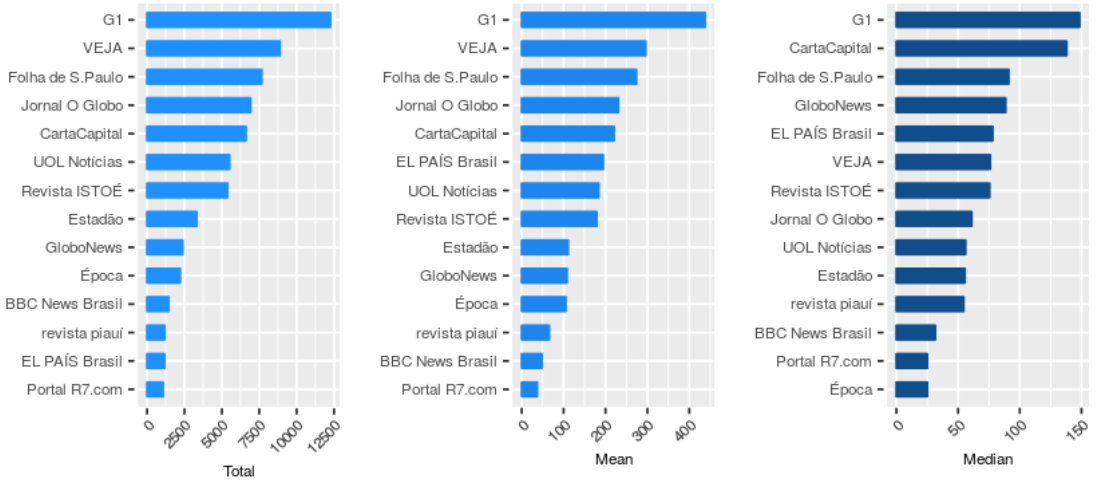
2.2 Profile Data

2.2.1 Total tweets



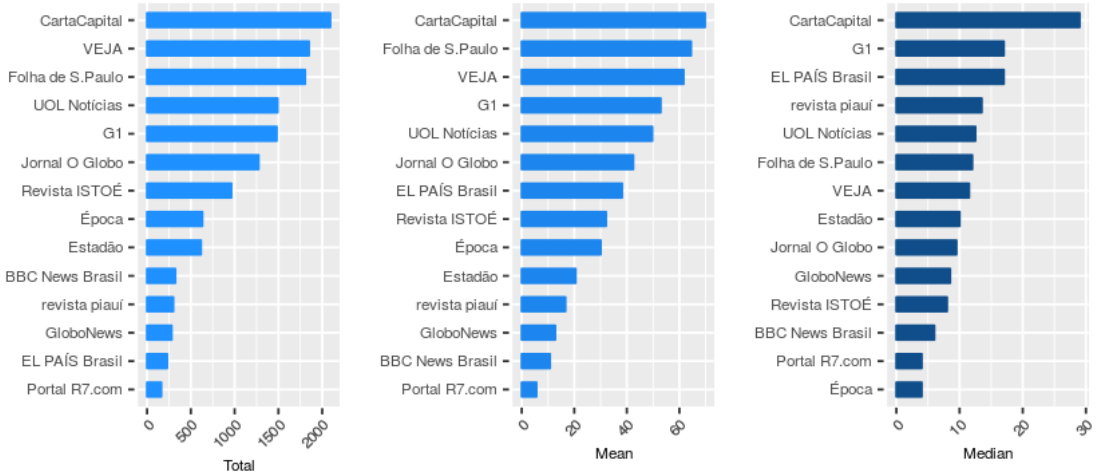
2.3 Tweets Data - Likes

Tweets Likes - Total , Mean and Median



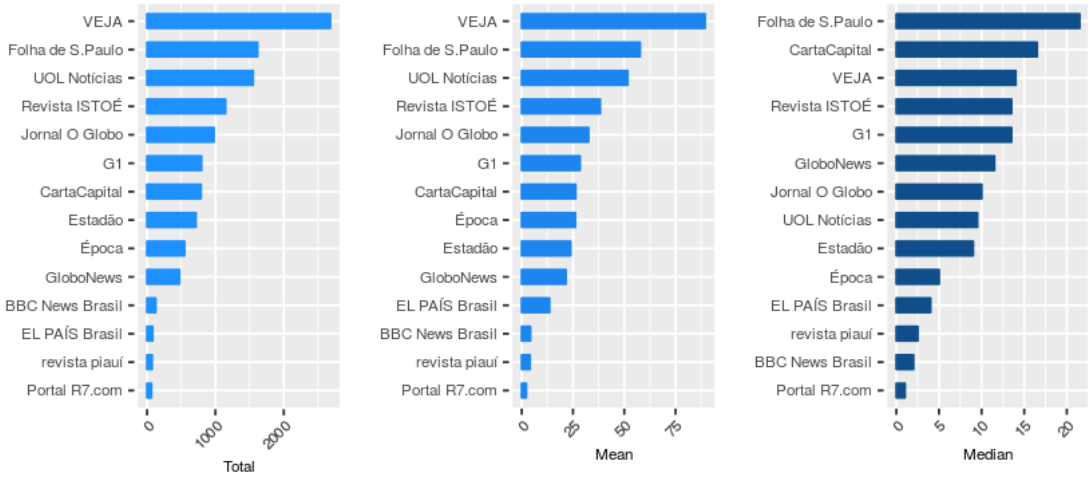
2.3.1 Tweets Data - Retweets

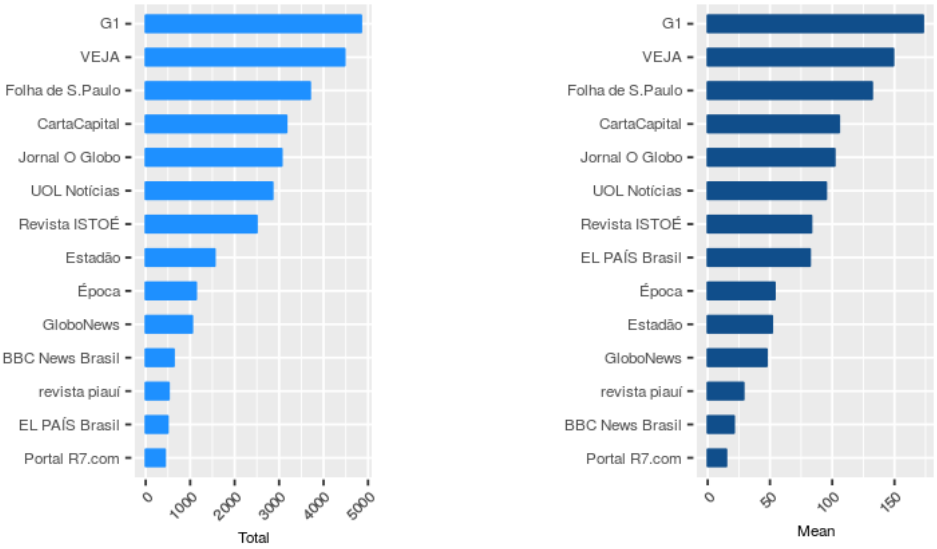
Tweets retweets - Total , Mean and Median



2.3.2 Tweets Data - Replies

Tweets Replies - Total , Mean and Median



2.3.3 Tweets Data - Engagement (replies + retweets + likes) /3

Tweets Engagement* - Total , Mean