Modo cientista de dados: ativar

Usando R para analisar sobre comportamento automatizado no Twitter

Julia Hellen e Malu Mondelli
Instituto de Tecnologia e Sociedade do Rio
28 de junho de 2021

Julia Hellen Ferreira

- Estagiária de TI / Pesquisadora no ITS Rio;
- Graduanda em Estatística e Pesquisadora na UFF;
- Programadora na Iniciativa Rio Mais+;
- Professora de Python e R;
- Sou híbrida! (Uma parte de exatas e outra de humanas);
- Beyoncé é minha religião;

Malu Mondelli

- Cientista de dados no ITS Rio
- Doutoranda em Modelagem Computacional no LNCC
- E bordadeira

Como a aula está dividida hoje

- Nosso fluxo de análise [Malu]
- Rstudio Cloud [Malu]
- API do Twitter e rtweet [Malu]
- Análise de dados: readr e dplyr [Julia]
- Análise de redes: conceitos e igraph [Malu]
- Links úteis

Nosso fluxo de análise hoje

1 Buscamos por determinado termo no Twitter através da API

Isso retorna uma tabela com os registros de tweets

2 Filtramos os @s dos usuários sem repetição

Ficamos com um arquivo com a listagem de @s

PEGABOT

3 Passamos a listagem para análise no Pegabot

Isso retorna uma tabela com o resultado das análises para cada perfil

Com as duas bases de dados seguimos para as análises

Como fica o relatório?

Mas vamos por partes...

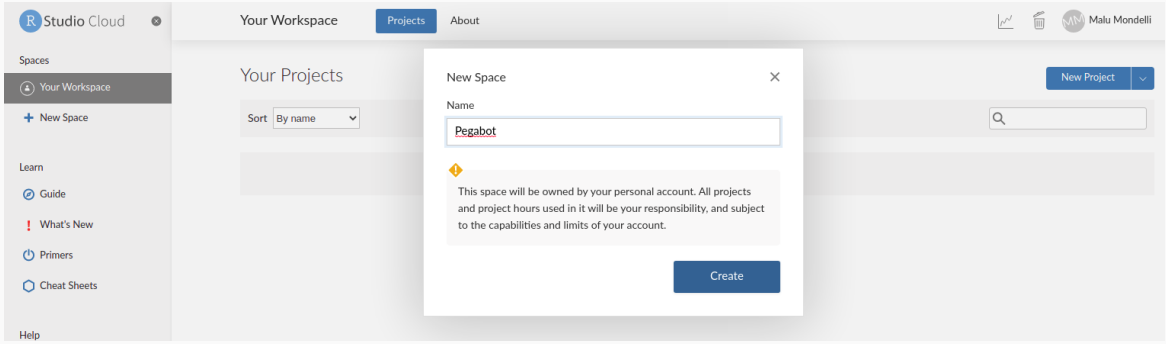
RStudio Cloud

Ambiente para desenvolvimento das análises usando R, link [aqui]

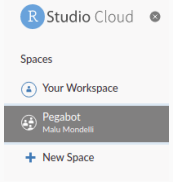
Passo a passo para utilizar

1. Log in ou Sign Up

2. New space (menu à esquerda)

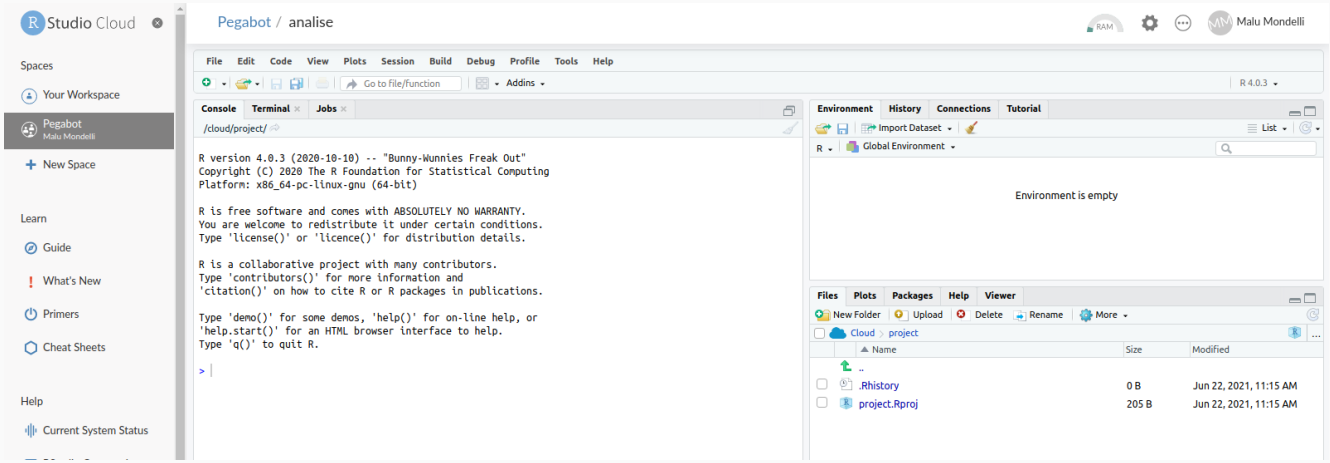


3. Para os próximos passos, certifique-se de que o workspace está selecionado, no menu à esquerda:



RStudio Cloud

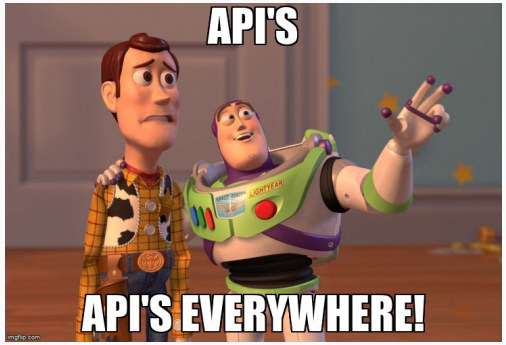
4. New Project (isso pode levar alguns segundos ou minutos)



Para instalar o RStudio localmente, você pode fazer o download da aplicação aqui.
Diferente do RStudio Cloud, você pode organizar seus workspaces por diretório/pasta.
Mas **atenção**: nesse caso é necessário fazer a instalação do R

API

- Traduzimos como **interface de programação de aplicação**
- **Definem um conjunto de regras** que permitem interações entre plataformas e usuários
- Facilitam a vida dos desenvolvedores de software - e a nossa também
- Cada plataforma define **o que é possível fazer/acessar** através das APIs e **o que pode ser disponibilizado** (funcionalidade ou dados)
- Esse acesso, na maioria das vezes, precisa de uma **autenticação**
- Um olho no que precisamos e queremos fazer, o outro na documentação



API do Twitter

Como ter o acesso à API do Twitter

1. Faça login na sua conta

2. Acesse o portal de desenvolvedor e solicite acesso de acordo com a opção que melhor se aplica

   Atenção: Para opção Acadêmica, certifique-se de que o projeto atende todos os requisitos exigidos pelo Twitter. Caso o pedido seja rejeitado, ainda não é possível editar ou tentar novamente.

3. Com o acesso liberado, crie um app preenchendo os campos:

   - Nome
   - Descrição
   - Website
   - Callback URL: `http://127.0.0.1:1410`

4. Acesse a aba Keys and tokens
   Lá estão as infos para acessar a API pelo R (ou por outra linguagem)

Pacote rtweet

Acessando a API do Twitter pelo R

Para acessar pelo R, podemos usar o pacote `rtweet`

```
## Instalação
install.packages('rtweet')
## Carregar o pacote
library(rtweet)

## Para autenticação da API
app_name <- "my_twitter_app"
# Copie e cole as suas chaves (essas são só de exemplo)
consumer_key <- "XYznzPFOFZR2aaaqqwa39FwWKN1Jp41"
consumer_secret <- "CtkGEWmSevZqdascvfJuKl6HHrBxbCybxI1xGLqrD5ynPd9jG0SoHZbD"
access_token <- "138249743873-FhHkahsdkjadveRqEZDVS0Y2iQzVX"
access_secret <- "rSuNSpLQkjsuyieasCfLYLpntaXcHzApZ3evy03QY"
```

e passamos essas variáveis para a função `create_token()`:

```
token <- create_token(app_name, consumer_key, consumer_secret, access_token, access_secret)
```

para saber se deu tudo certo você pode executar alguma função do pacote.

Overview das principais funções do rtweet

- `get_friends('pegabots')`: lista as contas que o usuário segue
- `get_followers('pegabots')`: lista as contas que são seguidas pelo usuário
- `search_tweets(q = "rstats")`: procura por tweets contendo um ou mais termos de busca
- `get_timeline('pegabots')`: procura pelos tweets da timeline do usuário
- `lookup_users('pegabot')`: retorna dados dos usuários
- `get_trends()`: retorna os 50 trending topics mais recentes (23424768 é o código do Brasil)
- Documentação detalhada com todas a funções e parâmetros

Análise de dados: readr e dplyr

Banco de Dados

Base Geral

Vamos realizar as análises a seguir utilizando uma base de tweets já coletados, com registros entre os dias 14 de junho de 2021 e 17 de junho de 2021. O termo usado para busca foi #FechadocomBolsonaro2022.

- A base contém 5202 observações (linhas) e 39 variáveis (colunas);
- Variáveis que vamos usar: date, username, tweet, nlikes e nretweets.

PEGABOT

Com ajuda do *PEGABOT* vamos ter uma outra base de dados já com as informações sobre comportamento automatizado.

- A base contém 2628 observações (linhas) e 19 variáveis (colunas);
- Variável que vamos usar: Análise Total;
- Vamos criar a variável: Resultado.

Bora aprender mais sobre o R?

Instalando os pacotes

Hoje existem diversos pacotes que nos auxiliam as nossas analises. Por isso, vou apresentar 3 pacotes essenciais e que irão facilitar seu dia a dia.

- readr -> install.packages("readr");
- dplyr -> install.packages("dplyr");
- ggplot -> install.packages("ggplot2").

Pacote readr

Este pacote tem como objetivo realizar a leitura das bases de dados. A função que iremos usar é read_csv() pois nosso arquivo está no formato .csv

A função escolhida dependerá do tipo de arquivo que você deseja ler. Como escolhermos **csv** teremos:

- read_csv()

Pacote dplyr

O Dplyr foi desenvolvido pelo Hadley Wickham. O pacote veio para facilitar o uso de funcionalidades já existentes no R. Sendo assim, fazer as análises de dados de *data frame* se tornou mais simples e muitas vezes com poucas linhas de comando.

Funções que vamos aprender:

- select();
- filter();
- mutate();
- arrange();
- group_by() e summarise();
- slice_max();
- distinct().

É ele o pipe! Tá passada?

dplyr

*Quem é esse tal de **pipe**?*

Sem pipe

```
julia = select(casa,
 quarto, sala)

julia = filter(julia,
quarto == "2 camas")
```

Com pipe

```
julia = casa %>%
select(quarto, sala) %>%
filter(quarto == "2 camas")
```

Conhecendo o select()

dplyr

select() *- selecionar as colunas da base.*

Código

```
info = base %>%
    select(Cidade, Matr)
```

Nome	Cidade	Idade	Matr
Julia	Niterói	24	111.111
Lucas	Itaboraí	30	115.711
João	Niterói	32	111.891
Bianca	Itaboraí	18	881.111

Cidade	Matr
Niterói	111.111
Itaboraí	115.711
Niterói	111.891
Itaboraí	881.111

Conhecendo o filter()

dplyr

filter() *- filtrar as linhas da base.*

Código

```
inf_pais = base %>%
    filter(Sigla == "BR")
```

País	Idioma	Sigla
Brasil	Português	BR
Japão	Japones	JP
Brasil	Português	BR
Brasil	Português	BR

País	Idioma	Sigla
Brasil	Português	BR
Brasil	Português	BR
Brasil	Português	BR

Conhecendo o mutate()

dplyr

mutate() *- criar uma nova coluna na base.*

Código

```
Meses = base %>%
    mutate( Abr = c(5, 8,
    23, 4))
```

Jan	Fev	Mar
5	6	7
1	3	0
11	25	4
7	2	3

Jan	Fev	Mar	Abr
5	6	7	5
1	3	0	8
11	25	4	23
7	2	3	4

Conhecendo o arrange()

dplyr

arrange() *- Ordenar a base.*

Código

```
time = base %>%
    arrange(Pos)
```

Pos	Time	Pontos
3	Vasco	5
4	Botafogo	1
2	Flumin.	10
1	Flamengo	25

Pos	Time	Gols
1	Flamengo	25
2	Flumin.	10
3	Vasco	5
4	Botafogo	1

Conhecendo o group_by() e summarise()

dplyr

group_by() e summarise() *- Ordenar a base.*

Nome	Idade	Prof
Julia	24	Estatística
Hellen	27	Estatística
Jorge	19	Herdeiro
Lucas	29	Estatística

Código

```
profis = base %>%
    group_by(Prof) %>%
    summarise(N = n())
```

Prof	N
Estatística	3
Herdeiro	1

Conhecendo o slice_max()

dplyr

slice_head() *- Selecionar as primeiras linhas.*

Código

```
toptimes = base %>%
  slice_head(n = 2)
```

Pos	Time	Gols
1	Flamengo	25
2	Flumin.	10
3	Vasco	5
4	Botafogo	1

Pos	Time	Gols
1	Flamengo	25
2	Flumin.	10

Conhecendo o distinct()

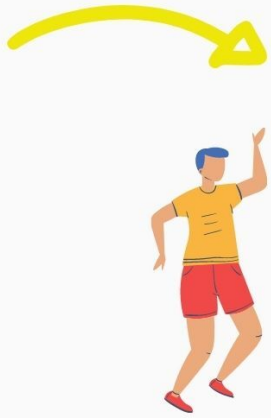
dplyr

distinct() *- Selecioanr linhas únicas.*

Código

```
Aniv = base %>%
       distinct(Dia, Ano)
```

Dia	Mês	Ano
28	Março	1997
04	Fev	1998
28	Jan	1997
03	Março	2000



Dia	Ano
28	1997
04	1997
03	1997

Bora programar!

Análise de redes: conceitos e igraph

Antes, alguns conceitos importantes

- Análise de redes sociais | Ciência de redes | Teoria dos grafos
- Uma rede é um conjunto de **entidades** conectadas entre si por meio de **relações**

- O que são essas entidades? Quais são os tipos de relações?
  Depende...
- Podemos atribuir propriedades à entidades e relações:
  Peso, cor, tamanho, tipo
- Podemos estabelecer direcionamento entre as relações:
  Redes direcionadas ou não direcionadas
- Técnicas de análise de redes servem como **ferramenta** para o estudo das características dessa estrutura.

Antes, alguns conceitos importantes

Objetivos e possibilidades

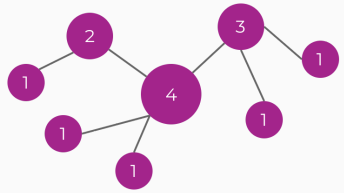
- Identificar atores importantes ou mais engajados;
- Identificar atores centrais na circulação de determinado conteúdo/tema;
- Entender sobre o volume de interações entre os atores;
- Entender como se dá a conexão entre diferentes tipos de entidades;
- Identificar grupos ou comunidades com interesses em comum;
- Identificar fenômenos e práticas de interferência na rede, nociva ou não.

Métricas úteis para hoje: centralidade

O quão importante é um vértice/nó na rede em relação aos demais?

Grau

considerada a forma mais simples, aponta quais são os usuários mais importantes de acordo com a quantidade de conexões que eles têm com os demais



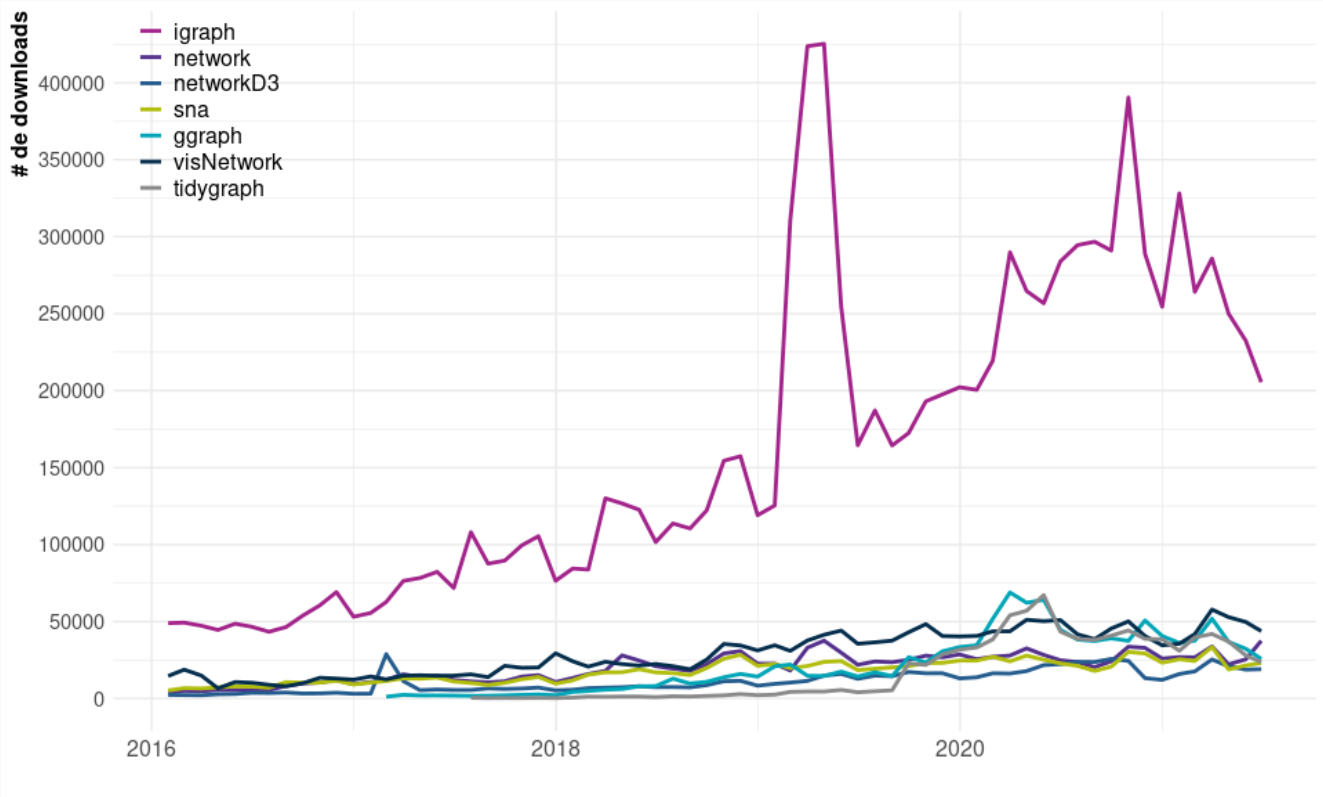
- **Entrada:** quantidade de conexões que apontam para o usuário;
- **Saída:** quantidade de conexões que saem do usuário e apontam para outros.

Intermediação

indica a importância do usuário pela quantidade de vezes que ele atua como 'ponte' na ligação entre outros usuários numa rede.

Análise de redes com igraph

Por quê igraph?



Bora lá no RStudio

Links úteis

Links úteis

- R
  - R basics
  - rwteet
  - Documentação igraph
  - Tutorial igraph
- Gephi
  - Tutorial por Jennifer Golbeck
  - Material da Escola de Dados
- Análise de redes em geral
  - Lista extensa de referências e tutoriais

Isso é tudo (: