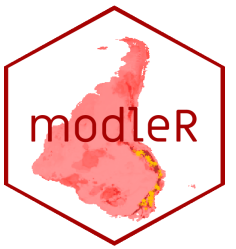
Trabalhos reprodutíveis com R e git: um exemplo usando o pacote coronabr

Sara Mortara

27 maio 2020

sobre

- bióloga, usuária de R desde 2009
- trabalho com reprodutibilidade em Ecologia, modelagem estatística e estudos de biodiversidade
- uma das desenvolvedoras do pacote **modler**, liderado por **Andrea Sánchez-Tapia**
- **liibre**
- **Observatório COVID-19 BR**
- **@RLadiesRio**



sobre hoje

1. por que reprodutibilidade?

2. usando R de forma reprodutível

3. perdendo o medo de git

4. um fluxo reprodutível com o pacote coronabr

1. por que reprodutibilidade?

motivos para reprodutibilidade

- por você no futuro
- por colegas
- evidências de que seus resultados estão corretos
- permitir que outres usem seus métodos e resultados

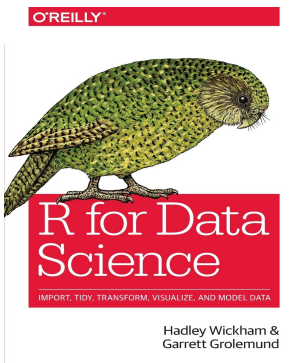

DON'T WORRY, YOU DON'T HAVE TO START YOUR CODE FROM SCRATCH.
YOU CAN RE-USE THE SOFTWARE THAT THE PREVIOUS PERSON ON THE PROJECT WROTE SEVERAL YEARS AGO.
ARE THERE INSTRUCTIONS FOR HOW TO USE IT?
I DOUBT IT.
IS THE CODE COMMENTED?
NOT LIKELY.
WHERE ARE THE FILES?
WHO KNOWS.
THIS IS GOING TO BE PAINFUL, ISN'T IT?
JUST A SCRATCH.
JORGE CHAM © 2014
WWW.PHDCOMICS.COM

passos para reprodutibilidade

- priorizar ferramentas baseadas em *scripts* como `R`
- usar sistemas de **controle de versões** como `git`
- documentar bem todos os passos e decisões
- publicar os protocolos e o código
- fomentar a revisão de metodologia e do código entre pares

por que R?

- script é essencial para reprodutibilidade, mas não a garante
- **código aberto**, **livre** & **sem custo**
- acessível (em comparação a outras linguagens de programação)
- muito comum na Biologia, Ciência de Dados e em diversas áreas



por que git?

- **controle de versão**
- permite acompanhar o histórico do desenvolvimento
- facilita o trabalho colaborativo
- facilita o compartilhamento de todas as etapas do trabalho

2. usando R de forma reprodutível

estrutura de pastas

nomes e **caminhos** são essenciais para o trabalho reprodutível!

```
.
├── codigo/              # Scripts em R
├── dados/               # Dados brutos
├── output/              # Outputs gerados a partir dos códigos
├── figs/                # Figuras geradas a partir dos códigos
├── docs/                # Relatórios reprodutíveis produzidos a partir dos outputs
├── *.Rproj              # Projeto de RStudio
├── .gitignore           # Lista dos arquivos e/ou pastas que não serão controlados
└── README.md            # Documentação -> Leia e escreva sempre que possível ;)
```

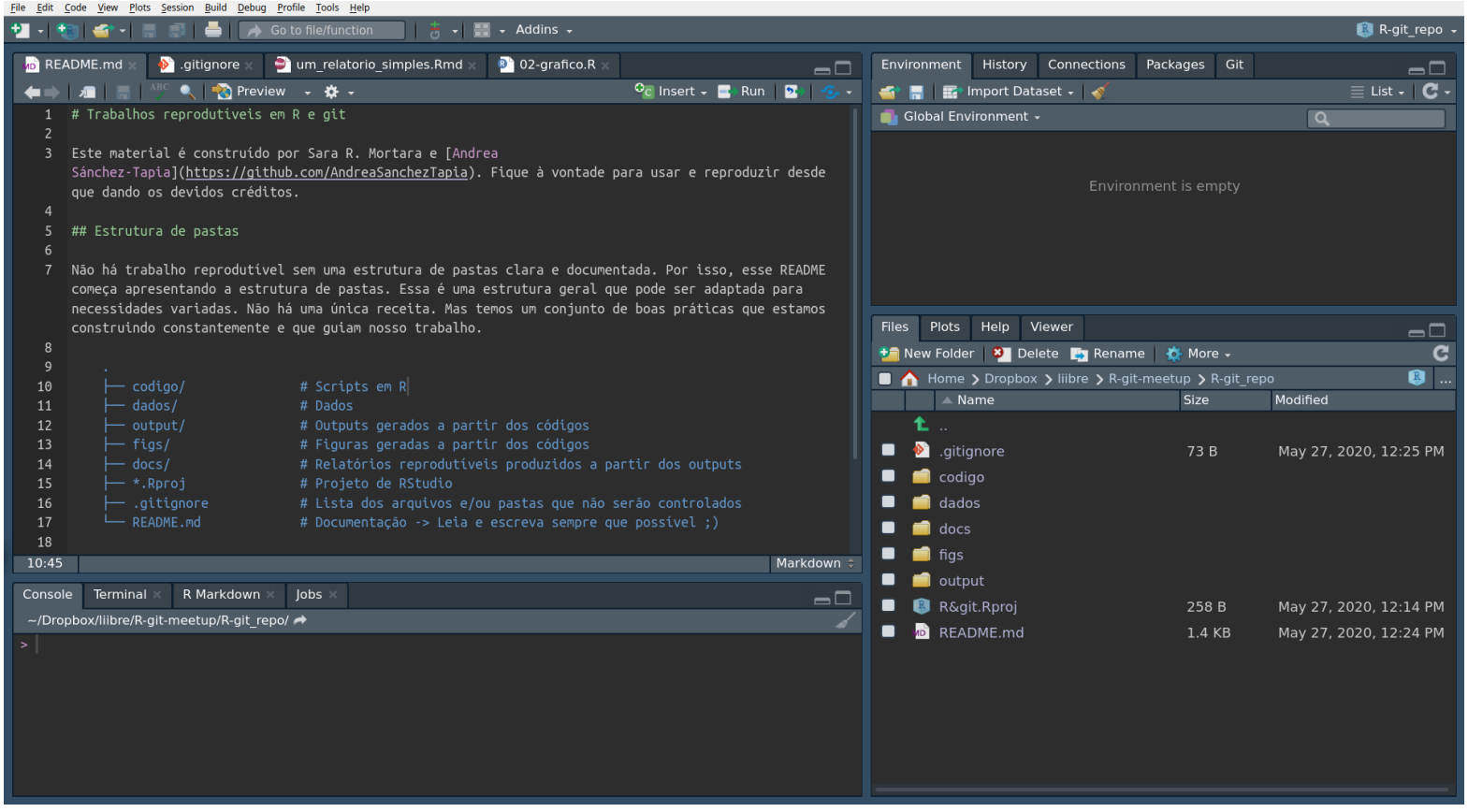
usando projetos de RStudio

esqueça `setwd()` e conheça Jenny Bryan

> *If the first line of your R script is*
>
> ```
> setwd("C:\Users\jenny\path\that\only\I\have")
> ```
>
> *I will come into your office and SET YOUR COMPUTER ON FIRE 🔥.*

> *If the first line of your R script is*
>
> ```
> rm(list = ls())
> ```
>
> *I will come into your office and SET YOUR COMPUTER ON FIRE 🔥.*

.Rproj define o wd



3. perdendo o medo de git

git é o sistema de controle de versão

- registra mudanças ao longo do tempo
- volta atrás se houver algum erro
- entende a diferença entre uma versão e outra do mesmo arquivo
- facilita colaboração: **compartilhar** as análises e **trabalhar em equipe**

hospedagem web para repositórios git

GitHub GitLab Atlassian Bitbucket

- *remotes*
- controla detalhadamente o conteúdo de **arquivos de texto**: `.txt`, `.csv`, `.md`, `.R`, `.tex`, `.Rmd`
- pode incluir outro tipo de arquivos (binários)
- o usuário decide **quais arquivos** incluir
- serve **localmente** e **offline**
- se comunica com **servidores remotos** que servem de *backup* e para distribuir (colaboração)

quatro estados

trabalho > **mudanças** > **salvar uma versão** > **mandar as versões para o remoto**

- *working directory*: arquivo adicionado para ser monitorado -> avisa quando for modificado
- *staging area*: `add` arquivos/mudanças adicionadas
- `commit`: cria uma versão com os arquivos adicionados
- `push`: manda para o remoto os commits que ainda não tiverem sido enviados

vários arquivos por *commit*, vários *commits* por *push*

fluxo básico de trabalho

sem ramificações

comandos básicos de `git`

`git clone URL` clona um repositório já existente

`git status` checa em que pé está

`git pull origin master` atualiza o repo localmente

`git add filename` adiciona um arquivo novo ou mudanças a arquivos monitorados

`git commit -m "uma mensagem informativa"` **cria uma versão**

`git push origin master` atualiza o repositório remoto

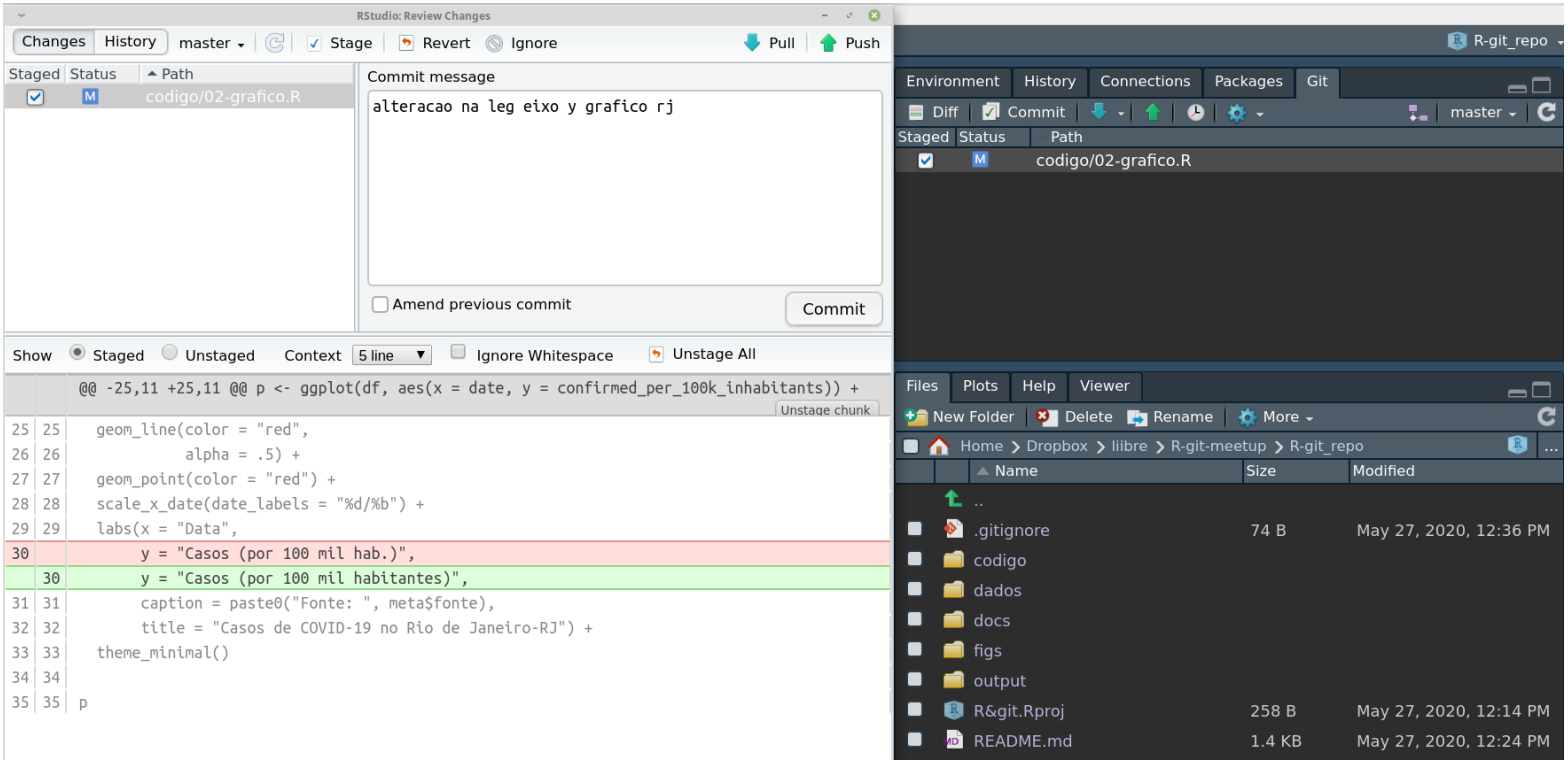
.Rproj + git = <3

```r
# 3. Fazendo um gráfico simples ####
p <- ggplot(df, aes(x = date, y = confirmed_per_100k_inhabitants)) +
  geom_line(color = "red",
            alpha = .5) +
  geom_point(color = "red") +
  scale_x_date(date_labels = "%d/%b") +
  labs(x = "Data",
       y = "Casos (por 100 mil hab.)",
       caption = paste0("Fonte: ", meta$fonte),
       title = "Casos de COVID-19 no Rio de Janeiro-RJ") +
  theme_minimal()
```

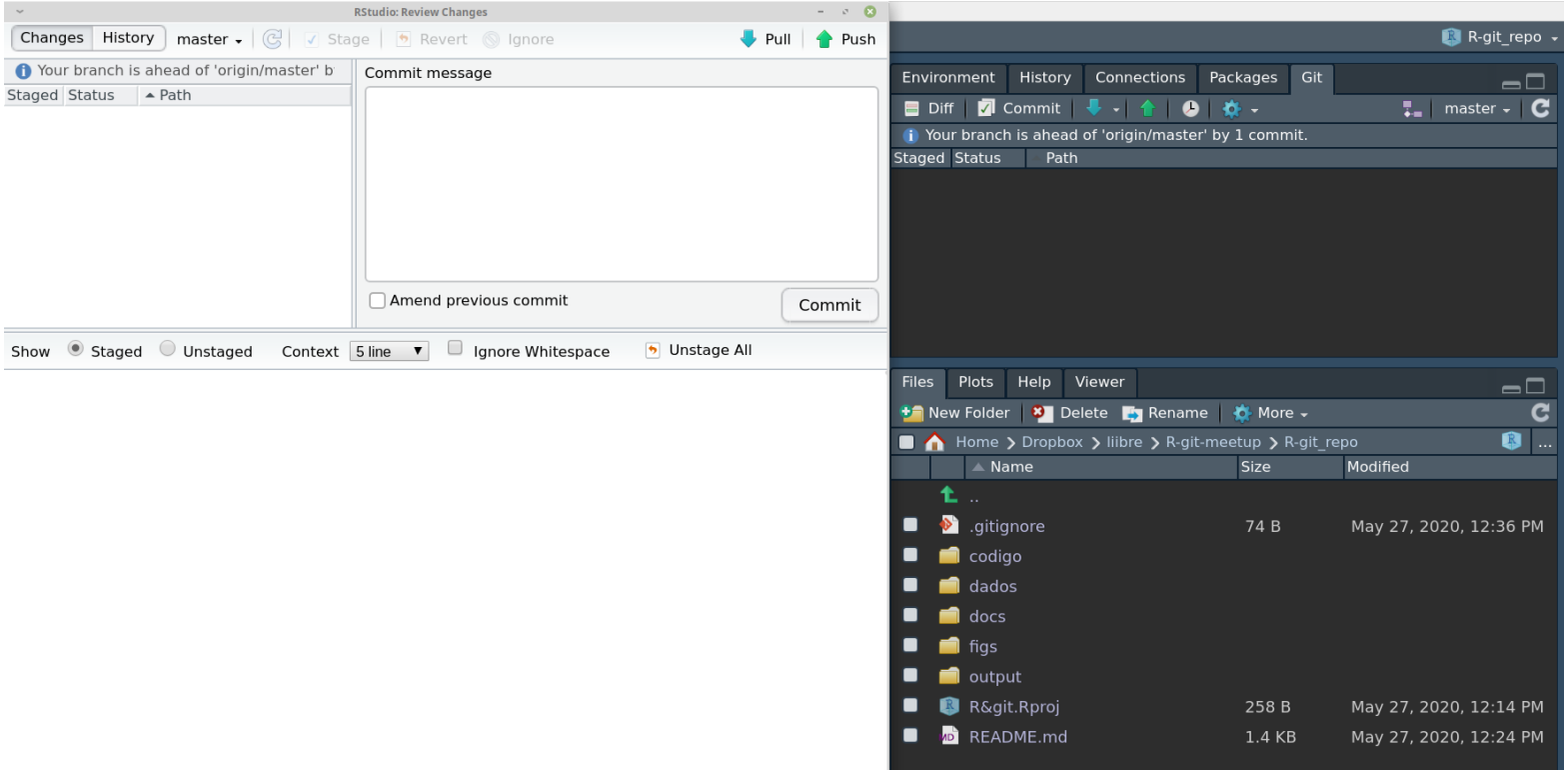
.Rproj + git = <3

```
23 # 3. Fazendo um gráfico simples ####
24 p <- ggplot(df, aes(x = date, y = confirmed_per_100k_inhabitants)) +
25   geom_line(color = "red",
26             alpha = .5) +
27   geom_point(color = "red") +
28   scale_x_date(date_labels = "%d/%b") +
29   labs(x = "Data",
30        y = "Casos (por 100 mil habitantes)",
31        caption = paste0("Fonte: ", meta$fonte),
32        title = "Casos de COVID-19 no Rio de Janeiro-RJ") +
33   theme_minimal()
```

.Rproj + git = <3


RStudio: Review Changes
Changes
History
master
Stage
Revert
Ignore
Pull
Push
Staged
Status
Path
codigo/02-grafico.R
Commit message
alteracao na leg eixo y grafico rj
Amend previous commit
Commit
Show
Staged
Unstaged
Context
5 line
Ignore Whitespace
Unstage All
@@ -25,11 +25,11 @@ p <- ggplot(df, aes(x = date, y = confirmed_per_100k_inhabitants)) +
Unstage chunk
geom_line(color = "red",
alpha = .5) +
geom_point(color = "red") +
scale_x_date(date_labels = "%d/%b") +
labs(x = "Data",
y = "Casos (por 100 mil hab.)",
y = "Casos (por 100 mil habitantes)",
caption = paste0("Fonte: ", meta$fonte),
title = "Casos de COVID-19 no Rio de Janeiro-RJ") +
theme_minimal()
p
R-git_repo
Environment
History
Connections
Packages
Git
Diff
Commit
master
Staged
Status
Path
codigo/02-grafico.R
Files
Plots
Help
Viewer
New Folder
Delete
Rename
More
Home
Dropbox
liibre
R-git-meetup
R-git_repo
Name
Size
Modified
..
.gitignore
74 B
May 27, 2020, 12:36 PM
codigo
dados
docs
figs
output
R&git.Rproj
258 B
May 27, 2020, 12:14 PM
README.md
1.4 KB
May 27, 2020, 12:24 PM

.Rproj + git = <3


RStudio: Review Changes
Changes
History
master
Stage
Revert
Ignore
Pull
Push
Your branch is ahead of 'origin/master' b
Staged
Status
Path
Commit message
Amend previous commit
Commit
Show
Staged
Unstaged
Context
5 line
Ignore Whitespace
Unstage All
R-git_repo
Environment
History
Connections
Packages
Git
Diff
Commit
master
Your branch is ahead of 'origin/master' by 1 commit.
Staged
Status
Path
Files
Plots
Help
Viewer
New Folder
Delete
Rename
More
Home
Dropbox
liibre
R-git-meetup
R-git_repo
Name
Size
Modified
..
.gitignore
74 B
May 27, 2020, 12:36 PM
codigo
dados
docs
figs
output
R&git.Rproj
258 B
May 27, 2020, 12:14 PM
README.md
1.4 KB
May 27, 2020, 12:24 PM

R + git = <3

```
git pull origin master

git add codigo/02-grafico.R

git commit -m "alteracao na leg eixo y grafico rj"

git push origin master
```

o que você precisa

1. não ter medo
2. ter uma conta de github | gitlab | bitbucket
3. instalar git no seu computador
4. configurar uma chave ssh no serviço remoto e no seu computador local

   (aula e tutorial)
5. aprender errando
6. seja sincere, não tente mudar a história, permita-se errar e recomeçar

4. um fluxo reprodutível com o pacote coronabr

https://github.com/saramortara/R-git-tutorial

o pacote coronabr

coronabr 0.1.0 | funções | como usar | mais exemplos e mapas | sobre

Download de dados de COVID-19 no Brasil

coronabr é um pacote de R para fazer *download* e visualizar os dados dos casos diários de coronavírus (COVID-19) disponibilizados por diferentes fontes:

- Ministério da Saúde;
- Brasil I/O;
- Johns Hopkins University

Nosso objetivo

O nosso objetivo é facilitar o acesso aos dados de diferentes fontes, usando ferramentas de acesso aberto e que permitam reprodutibilidade.

O código é aberto. Entre em como usar para um exemplo de como utilizar o pacote. Compartilhe.

Fazemos ciência aberta, democrática e reprodutível. Este é um trabalho em desenvolvimento. Para entender como contribuir, clique aqui.

responsabilidade com dados

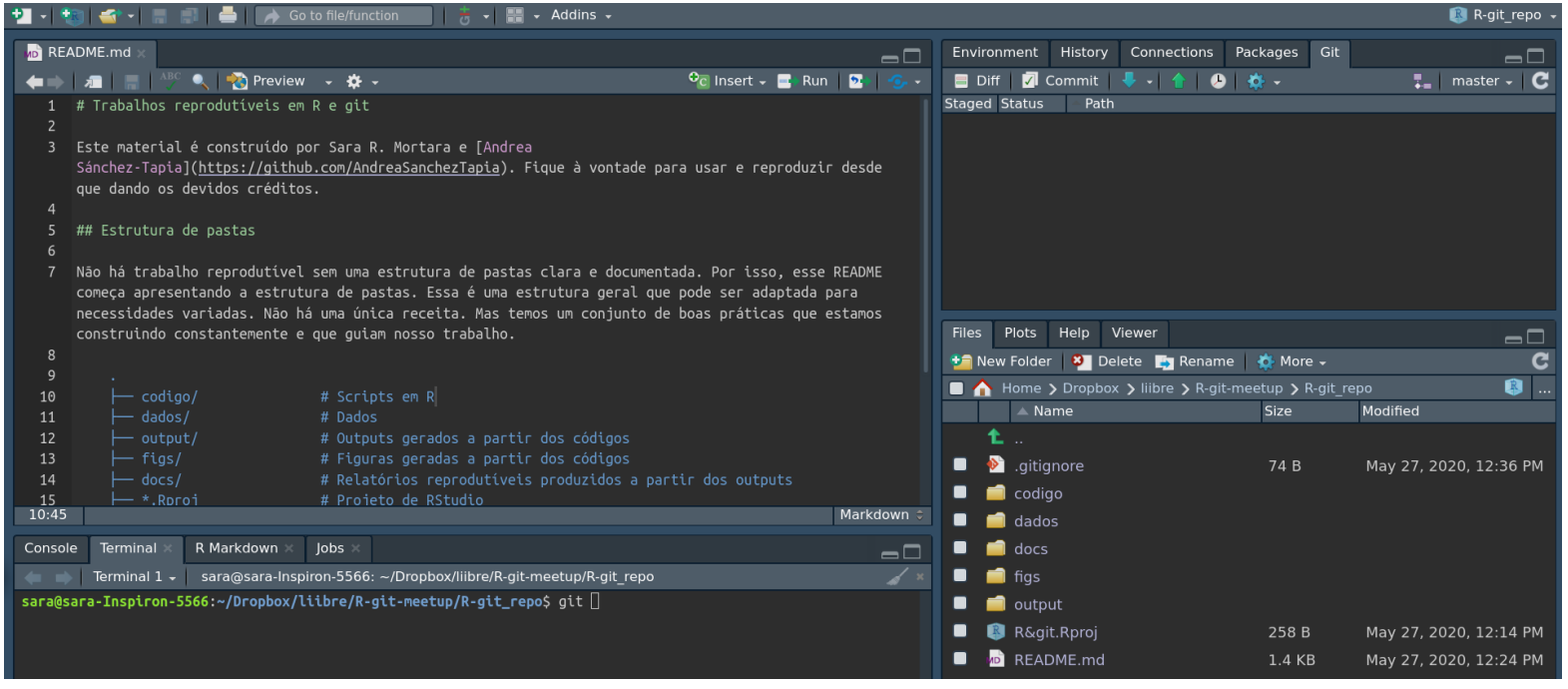
- dados deveriam ser abertos e acessíveis

  Trânsparência COVID-19 OPEN KNOWLEDGE BRASIL

- nem toda análise que **pode** ser feita, **deve** ser feita
- cada dado diz respeito a uma pessoa
- para COVID-19 e SRAG: **subnotificação** & **atraso**
- inconsistência com dados reportados em diferentes escalas: município, estado, país (no Rio de Janeiro)

um repositório remoto com pastas :)

o repositório local no R Studio :)



a estrutura de pastas

```
.
├── codigo/
│   ├── 01-download_dados.R
│   ├── 02-grafico.R
│   └── 03-nowcasting.R
├── dados/
│   ├── nowcasting_acumulado_covid_2020_05_18.csv
│   ├── README.md
├── output/
│   ├── 01-rio_de_janeiro.csv
│   └── metadado_corona_br.csv
├── figs/
│   └── 02-rj_dados_brutos.png
├── docs/
│   └── um_relatorio_simples.Rmd
...
```

rodando os códigos

01-download_dados.R

```
##############################################################################
# Script para download de dados de covid-19 no município do Rio de Janeiro
# por Sara Mortara, para rladiesrio
##############################################################################

# Para instalar o pacote use:
#remotes::install_github("liibre/coronabr")

# 1. bibliotecas ####
library(coronabr)

# 2. download ####
## dados rio de janeiro usando o geocode IBGE
rj <- get_corona_br(filename = "01-rio_de_janeiro",
                    ibge_cod = "3304557")
```

01-download_dados.R

```
# 3. inspeção dos dados ####
## checando os dados
head(rj)
```

```
##         date state           city place_type confirmed deaths is_last
## 1 2020-05-25    RJ Rio de Janeiro       city     22466   2831    True
## 2 2020-05-24    RJ Rio de Janeiro       city     21775   2755   False
## 3 2020-05-23    RJ Rio de Janeiro       city     21043   2702   False
## 4 2020-05-22    RJ Rio de Janeiro       city     20161   2520   False
## 5 2020-05-21    RJ Rio de Janeiro       city     18743   2376   False
## 6 2020-05-20    RJ Rio de Janeiro       city     17066   2249   False
##   estimated_population_2019 city_ibge_code confirmed_per_100k_inhabitants
## 1                   6718903        3304557                        334.3701
## 2                   6718903        3304557                        324.0856
## 3                   6718903        3304557                        313.1910
## 4                   6718903        3304557                        300.0639
## 5                   6718903        3304557                        278.9592
## 6                   6718903        3304557                        253.9998
##   death_rate
## 1     0.1260
## 2     0.1265
## 3     0.1284
## 4     0.1250
## 5     0.1268
## 6     0.1318
```

01-download_dados.R

```
## intervalo de tempo
range(rj$date)
```

```
## [1] "2020-03-06" "2020-05-25"
```

```
## casos acumulados (inclui recuperados ~ 19 mil)
max(rj$confirmed)
```

```
## [1] 22466
```

```
## casos proporcional à populaçcao
max(rj$confirmed_per_100k_inhabitants)
```

```
## [1] 334.3701
```

```
## obitos
max(rj$deaths)
```

```
## [1] 2831
```

discrepância entre dado estadual e municipal!

Mudança em método da prefeitura faz Rio registrar menos 1.177 óbitos por Covid-19

Prefeito Marcelo Crivella anunciou alteração protocolo de contagem da doença na última sexta-feira (22).

Por G1 Rio

26/05/2020 22h47 · Atualizado há 10 horas

02-grafico.R

```
#########################################################
# Fazendo um gráfico simples para o município do RJ
# por Sara Mortara, para rladiesrio
#########################################################

# 1. bibliotecas ###
library(ggplot2)

# 2. lendo e padronizando os dados ####
## data frame com os dados
df <- read.csv("output/01-rio_de_janeiro.csv")

## metadados
meta <- read.csv("output/metadado_corona_br.csv")
```

02-grafico.R

```
## metadados
meta
```

```
##               intervalo                                      fonte  acesso_em
## 1 2020-02-25;2020-05-26 https://brasil.io/dataset/covid19/caso 2020-05-27
```

```
## convertendo coluna com data p/ classe Date
class(df$date)
```

```
## [1] "character"
```

```
df$date <- as.Date(df$date)
class(df$date)
```

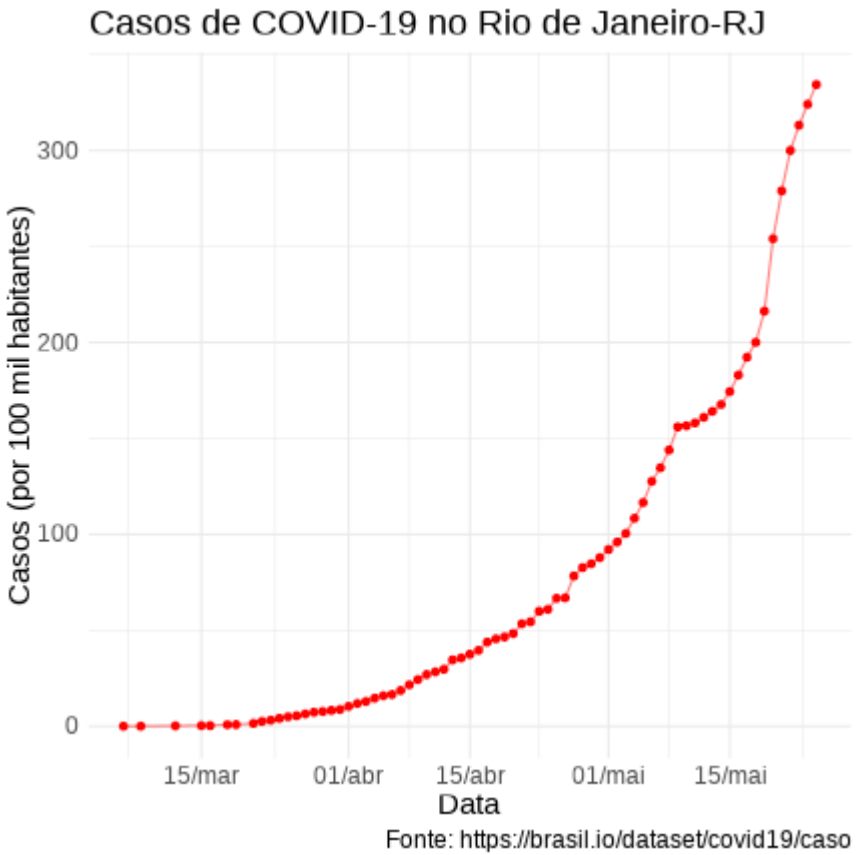
```
## [1] "Date"
```

criando um gráfico com ggplot2

```
# 3. Fazendo um gráfico simples ####
p <- ggplot(df, aes(x = date, y = confirmed_per_100k_inhabitants)) +
  geom_line(color = "red",
            alpha = .5) +
  geom_point(color = "red") +
  scale_x_date(date_labels = "%d/%b") +
  labs(x = "Data",
       y = "Casos (por 100 mil habitantes)",
       caption = paste0("Fonte: ", meta$fonte),
       title = "Casos de COVID-19 no Rio de Janeiro-RJ") +
  theme_minimal()
```

nosso gráfico

```
p
```



exportando o gráfico

```
png("figs/02-rj_dados_brutos.png", res = 300,
    width = 1400, height = 1200)
p
dev.off()
```

```
.
...
├── figs/
|      └── 02-rj_dados_brutos.png
...
```

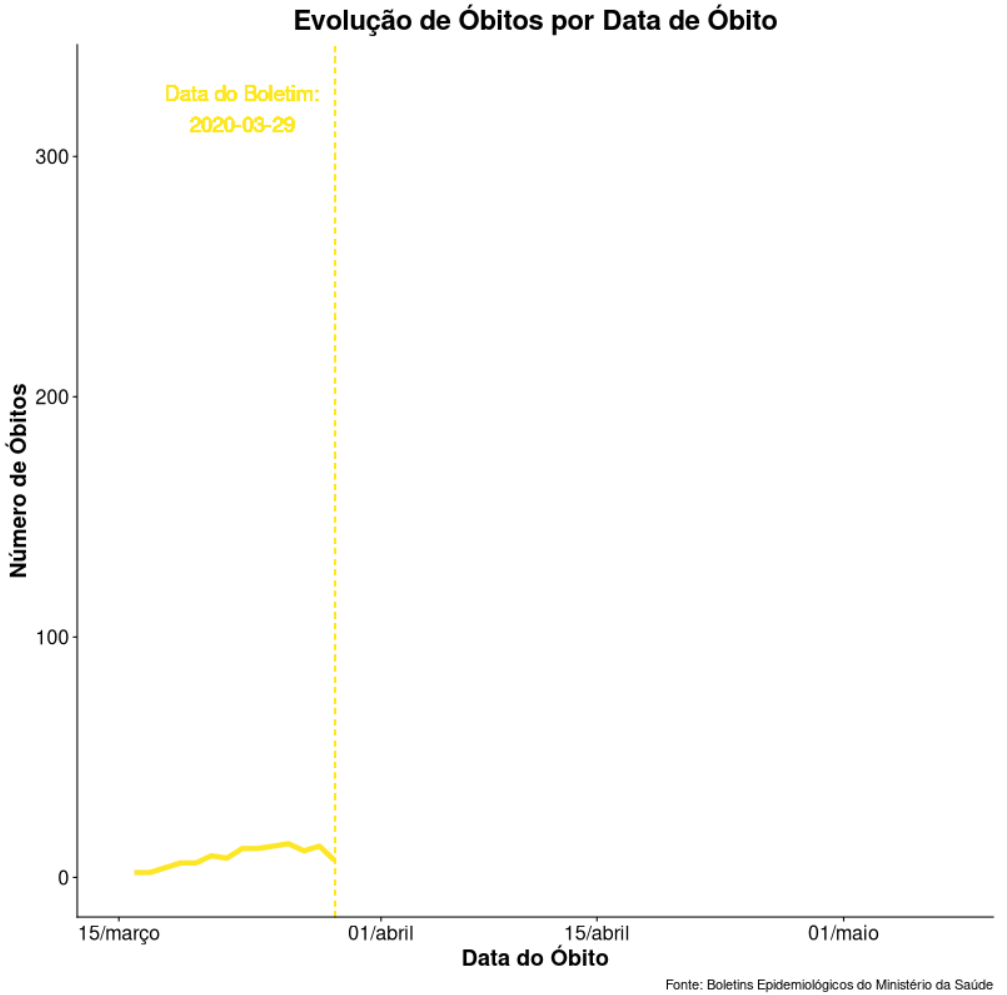
um alerta sobre os dados brutos de COVID-19

ressalvas em relação aos dados brutos

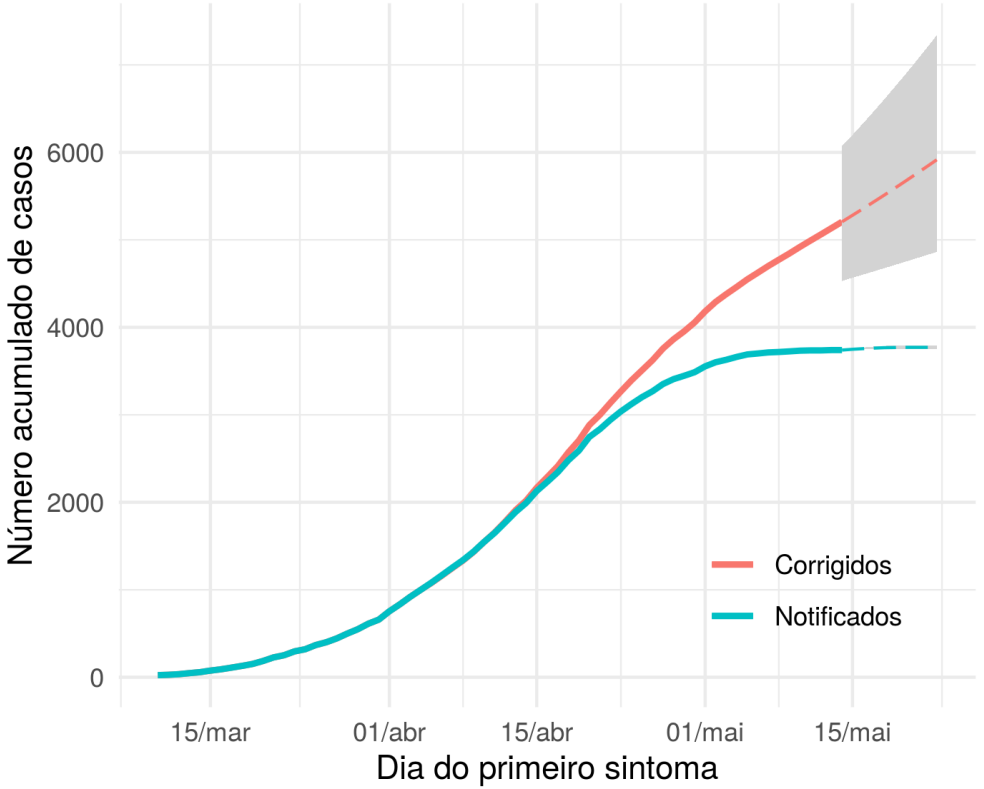
1. **subnotificação**

2. **atraso** na entrada dos dados no sistema - tanto para casos como óbitos

como vemos o atraso



como corrigir o atraso com nowcasting



Fonte: SIVEP-Gripe, dados processados por OBSERVATÓRIO COVID-19 BR

um relatório simples com Rmarkdown

- Rmarkdown :)
- embebe diretivas de formato em um documento de texto
- inspirado em $\LaTeX$ e Markdown
- um tutorial aqui

marcações de Rmarkdown

`*italico*`	*italico*
`**negrito**`	**negrito**
`` `codigo` ``	`codigo`

um exemplo com nosso trabalho

````
---
title: "Fazendo um relatório simples"
author: "Sara Mortara"
date: "27 de maio 2020"
output: html_document
---

```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = FALSE)
```

```{r dados}
df <- read.csv("output/01-rio_de_janeiro.csv")
meta <- read.csv("output/metadado_corona_br.csv")
```

## Casos acumulados de COVID-19 no Rio de Janeiro-RJ

De acordo com o portal [Brasil.io](`r meta$fonte`) que extrai dados diretamente da Secretaria de Saúde do Estado do Rio
de Janeiro, entre `r format(as.Date(min(df$date)), "%d/%m/%Y")` e `r format(as.Date(max(df$date)), "%d/%m/%Y")` o Rio
de Janeiro possui `r max(df$confirmed)` casos de COVID-19. Esse número inclui casos recuperados e não inclui
subnotificação ou atrasos nas notificações.

![](../figs/02-rj_dados_brutos.png)
````

um exemplo com nosso trabalho

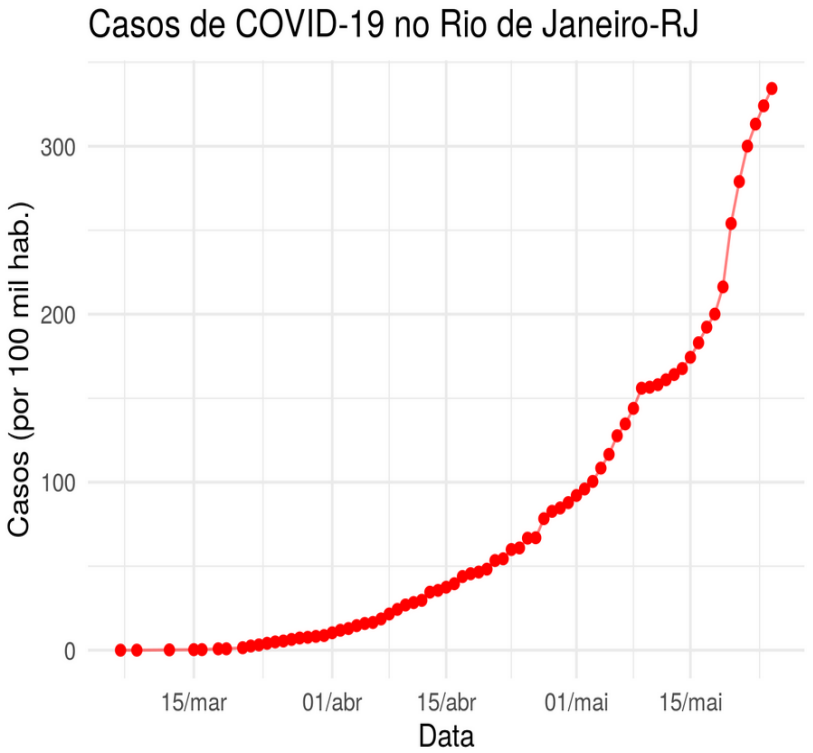
Fazendo um relatório simples

Sara Mortara

27 de maio 2020

Casos acumulados de COVID-19 no Rio de Janeiro-RJ

De acordo com o portal Brasil.io que extrai dados diretamente da Secretaria de Saúde do Estado do Rio de Janeiro, entre 06/03/2020 e 25/05/2020 o Rio de Janeiro possui 22466 casos de COVID-19. Esse número inclui casos recuperados e não inclui subnotificação ou atrasos nas notificações.



kit básico de ferramentas: R, git, Rmarkdown

para saber mais

- material curso liibre
- Jenny Bryan - workflows
- Jenny Bryan - happy git with R
- Daniele Navarro - robust tools
- Page Piccinini - R & git setup

obrigada!

@MortaraSara