Por que usar R?

Caio Lente

Credenciais

- Caio Lente
- Bacharel em Ciência da Computação pelo IME-USP
- Mestando em Ciência da Computação no IME-USP
- Sócio e cientista de dados na Terranova Consultoria
- Sócio e professor na Curso-R Treinamentos

História do R

- 1993: a primeira aparição do R
- 2000: sai a v1.0, a primeira versão pronta para produção
- 2006: ocorre a primeira useR!, a conferência anual de programadores R
- 2011: é lançado o RStudio, a IDE mais popular para R
- Atualmente (12/2020), ela já é a 9ª linguagem mais popular do mundo

R Básico

- R é uma linguagem para programação estatística
  - A classe **data frame** é embutida na linguagem
  - Para facilitar a realização de experimentos, ela usa scripting
- Preocupação central em compatibilidade reversa
- Por trás dos panos, praticamente roda C

```
df <- data.frame(x = runif(5))

df
```

```
#>           x
#> 1 0.2509955
#> 2 0.8634157
#> 3 0.4522537
#> 4 0.6273118
#> 5 0.3001410
```

```
sd(df$x)
```

```
#> [1] 0.2512464
```

R Básico

```
vec <- c(1, 3, 4, 7, 9)
lst <- list(1, "b", TRUE)
```

```
vec[vec < 5]
```

```
#> [1] 1 3 4
```

```
vec * 2
```

```
#> [1]  2  6  8 14 18
```

```
lst[[2]]
```

```
#> [1] "b"
```

```
factor(c("G", "G", "M", "P", "P"))
```

```
#> [1] G G M P P
#> Levels: G M P
```

```
lm(mpg ~ wt, mtcars)
```

```
#>
#> Call:
#> lm(formula = mpg ~ wt, data = mtcars)
#>
#> Coefficients:
#> (Intercept)           wt
#>      37.285       -5.344
```

R Moderno

- O advento do `tidyverse` mudou o R para sempre
  - Programação paralela
  - Programação funcional
  - **Tibbles** vs. data frames
  - NSE: non-standard evaluation
- Novas funcionalidades sem prejudicar código já implementado
- Origem: Tidy Data Manifesto

```
library(tidyverse)

tb <- tibble(x = runif(5))
tb <- mutate(tb, y = x * 2)
tb <- arrange(tb, y)

tb
```

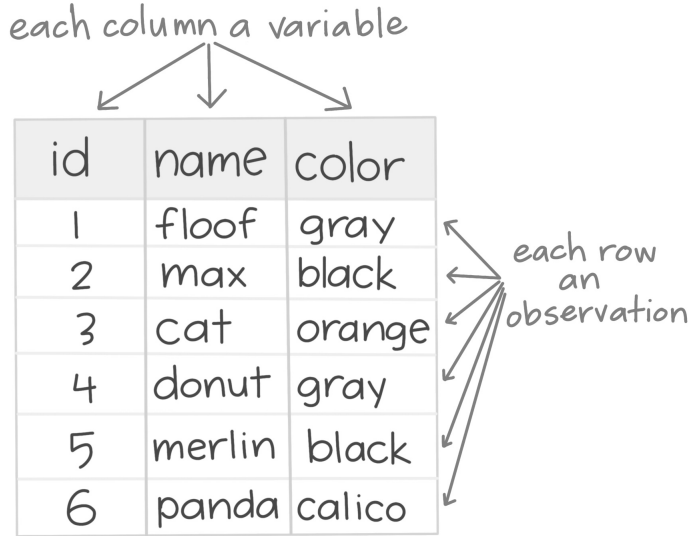
```
#> # A tibble: 5 x 2
#>       x     y
#>   <dbl> <dbl>
#> 1 0.106 0.213
#> 2 0.116 0.232
#> 3 0.205 0.410
#> 4 0.758 1.52
#> 5 0.866 1.73
```

Tidyverse

> "TIDY DATA is a standard way of mapping the meaning of a dataset to its structure."
>
> –HADLEY WICKHAM

In tidy data:

- each variable forms a column
- each observation forms a row
- each cell is a single measurement



Tidyverse

- `dplyr` é como o `pandas` do R, mas muito mais poderoso

- Pipeline: a saída de uma linha vira entrada da próxima

```
mtcars %>%
  filter(gear < 5) %>%
  group_by(cyl) %>%
  summarise(mpg = mean(mpg)) %>%
  mutate(kpl = mpg * 0.425) %>%
  select(cyl, kpl)
```

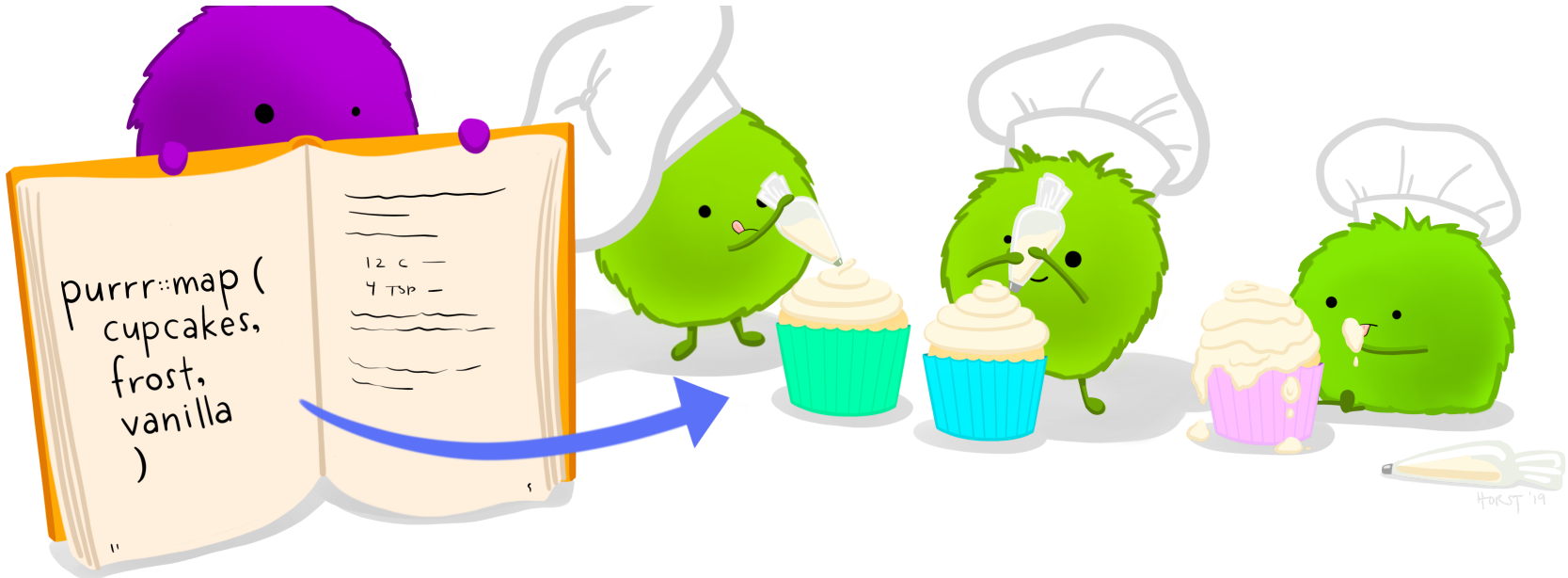
```
#> # A tibble: 3 x 2
#>     cyl   kpl
#>   <dbl> <dbl>
#> 1     4 11.2
#> 2     6  8.39
#> 3     8  6.40
```

R Funcional

- Uma das maiores adições ao R foi o pacote purrr
- Programação funcional e lambdas

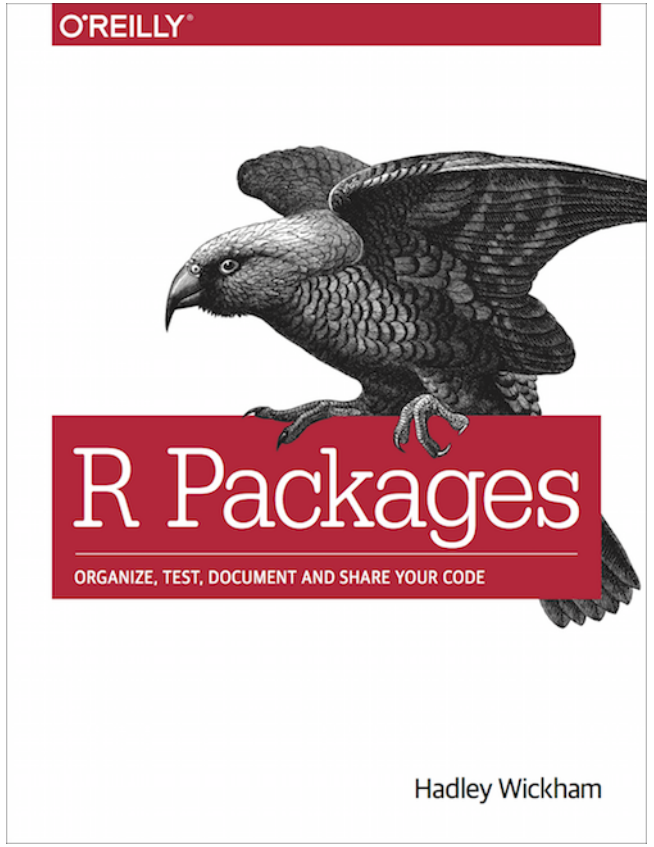
```
map_dbl(c(1,10), ~runif(1, max=.x))
```

```
#> [1] 0.1170238 5.2411597
```



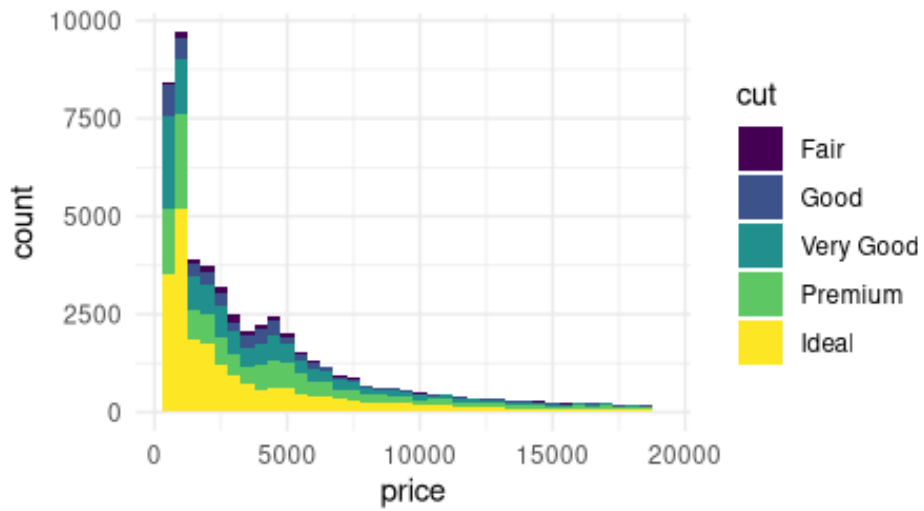
Pacotes e o CRAN

- Os pacotes do R são como as bibliotecas de outras linguages
- O jeito mais fácil de instalar um pacote é do CRAN
    - O CRAN é um órgão que garante a qualidade dos pacotes
    - O processo de aceite é **rigoroso**
    - Há testes para 12 arquiteturas
    - A qualidade dos pacotes do R é excepcionalmente alta



R para Visualização

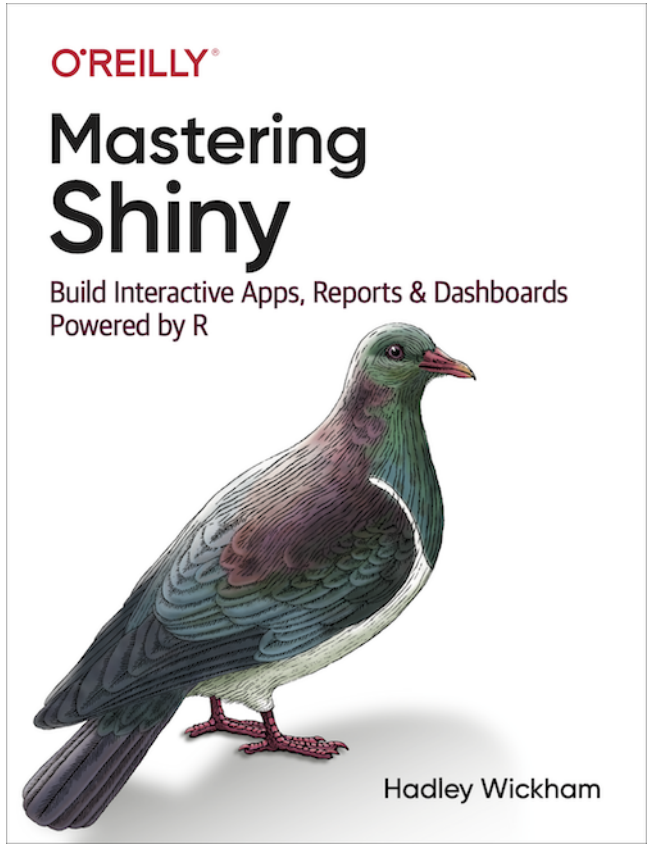
```r
library(ggplot2)

diamonds %>%
  ggplot(aes(price, fill = cut)) +
  geom_histogram(binwidth = 500)
```



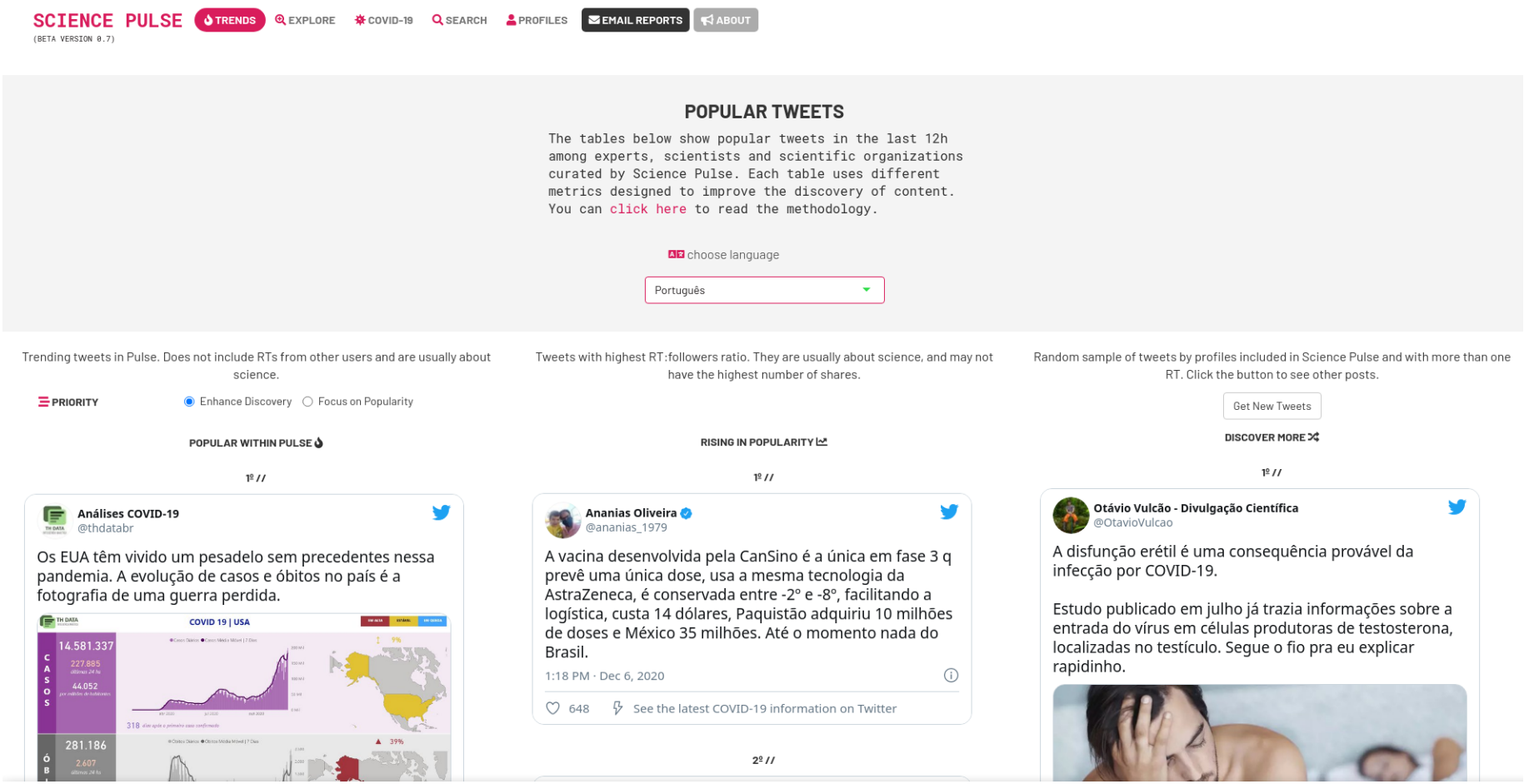
- Na gramática dos gráficos, definimos um gráfico em camadas
  - A **estética** define os eixos
  - A **geometria** define as formas
  - Podemos empilhas estéticas e geometrias
- O `ggplot2` permite criar gráficos a partir de tabelas
- Em poucas linhas, temos um gráfico preciso e elegante

R para Dashboards

- `shiny` é o "killer app" do R
- Esse pacote permite criar dashboards interativos em R
  - Saber um pouco de HTML/CSS ajuda, mas é desnecessário
  - Código R existente pode ser facilmente portado
  - O shinyapps.io hospeda de graça
- Alguns sites nem parecem dashboards normais



R para Dashboards



R vs. Python

```r
# NSE não precisa de aspas
mutate(df, Area = L*H)

# Mutate para N variáveis
mutate(df, A = L*H, V = A*10)


# Chaining com pipes
df %>%
  gather() %>%
  rename("k"="key","v"="value") %>%
  filter(v > 1)



# Operação comum
bind_cols(df1, df2)
```

```python
# Repetição do nome da base
df['Area'] = df.L*df.H

# Lambda prolixo
df.assign(A=lambda df: df.L*df.H,
          V=lambda df: df.A*10)

# Chaining com ponto
df = (pd.melt(df)
        .rename(columns={
                'variable' : 'k',
                'value' : 'v'})
        .query('v > 1')
     )

# Reutilização de operador
pd.concat([df1,df2], axis=1)
```

R vs. Python

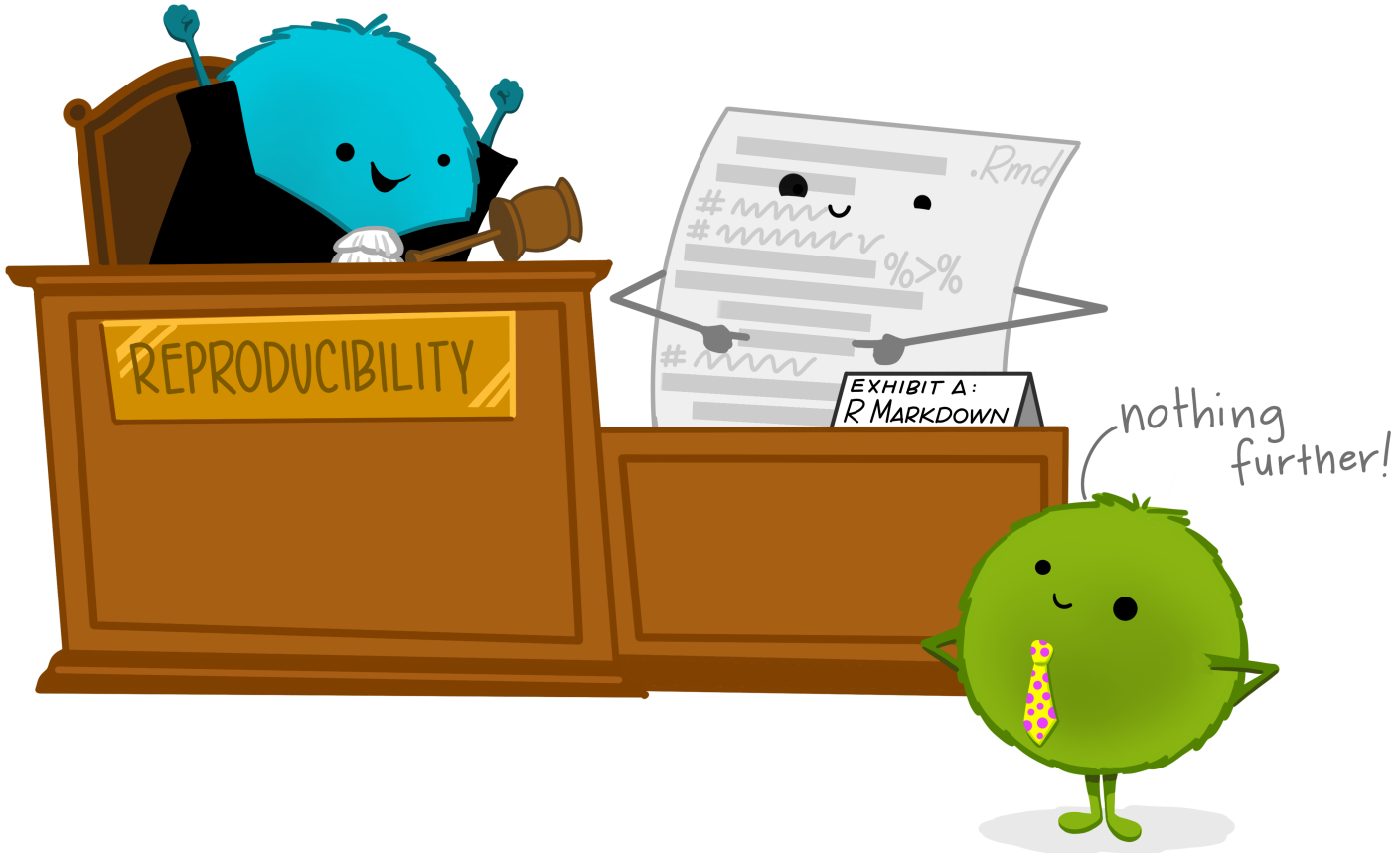
- Não existe discussão sobre a "linguagem melhor", apenas sobre qual linguagem é melhor para quê
- Python claramente é uma linguagem mais completa e amplamente utilizada, mas o R também é uma linguagem moderna e extremamente capaz
- O fato de o R ter sido feito para trabalhar com dados facilita certas operações
  - Modelagem estatística é trivial e embutida no R
  - Leitura e manipulação de dados são parte do núcleo da linguagem
  - Pacotes estáveis e retrocompatíveis acabam com a necessidade de `pyenv`, `virtualenv` ou `anaconda`
  - A comunidade do R tem crescido muito e se ajudado a continuar assim

Jupyter vs. RMarkdown

- Jupyter é incrível, mas péssimo do ponto de vista científico
  - "Hidden state and out-of-order execution"
  - "Encourages bad habits"
  - "Discourages modularity and testing"
  - "Hinders reproducible science"
  - "Makes it easy to teach poorly"

- RMarkdown junta código e texto de forma **reprodutível**
  - Para exportar, o arquivo deve ser rodado por completo
  - É impossível esconder algo no RMarkdown
  - Múltiplos formatos: PDF, HTML, Word, etc.
  - Estes slides foram feitos inteiramente em RMarkdown!

Jupyter vs. RMarkdown



R com Python

- E se o melhor não for R ou Python, mas sim R **e** Python?
  - O pacote `reticulate` integra perfeitamente os dois ambientes
  - Ele permite invocar funções do Python de dentro do R
  - Você consegue até compartilhar objetos de um para o outro
- O port do `tensorflow` para o R e feito com `reticulate`!

```
df_r <- tibble(x = runif(5))
```

```
df_py = r.df_r
df_py['y'] = df_py.x*2
```

```
py$df_py
```

```
#>           x         y
#> 1 0.7111129 1.4222259
#> 2 0.7525411 1.5050823
#> 3 0.9441328 1.8882656
#> 4 0.8226734 1.6453468
#> 5 0.3105045 0.6210089
```

R com C++

- O R também tem uma integração íntima com o C++
  - O pacote `Rcpp` consegue trazer código C++ para o R...
  - ...E código R para o C++
- Útil para aumentar a performance de funções do R
- Alguns ports são feitos completamente pelo `Rcpp`
- Suporte para `armadillo`, `Intel TBB`, etc.

Comunidades

- O R não é nada sem as suas comunidades e livros!
  - R-Ladies
  - Carpentries
  - #rstats
  - R Brasil
  - Curso-R
  - Ciência de Dados em R
  - Zen do R

Fim