

Preparado para:



# REFORM/SC2022/126

## DELIVERABLE 4

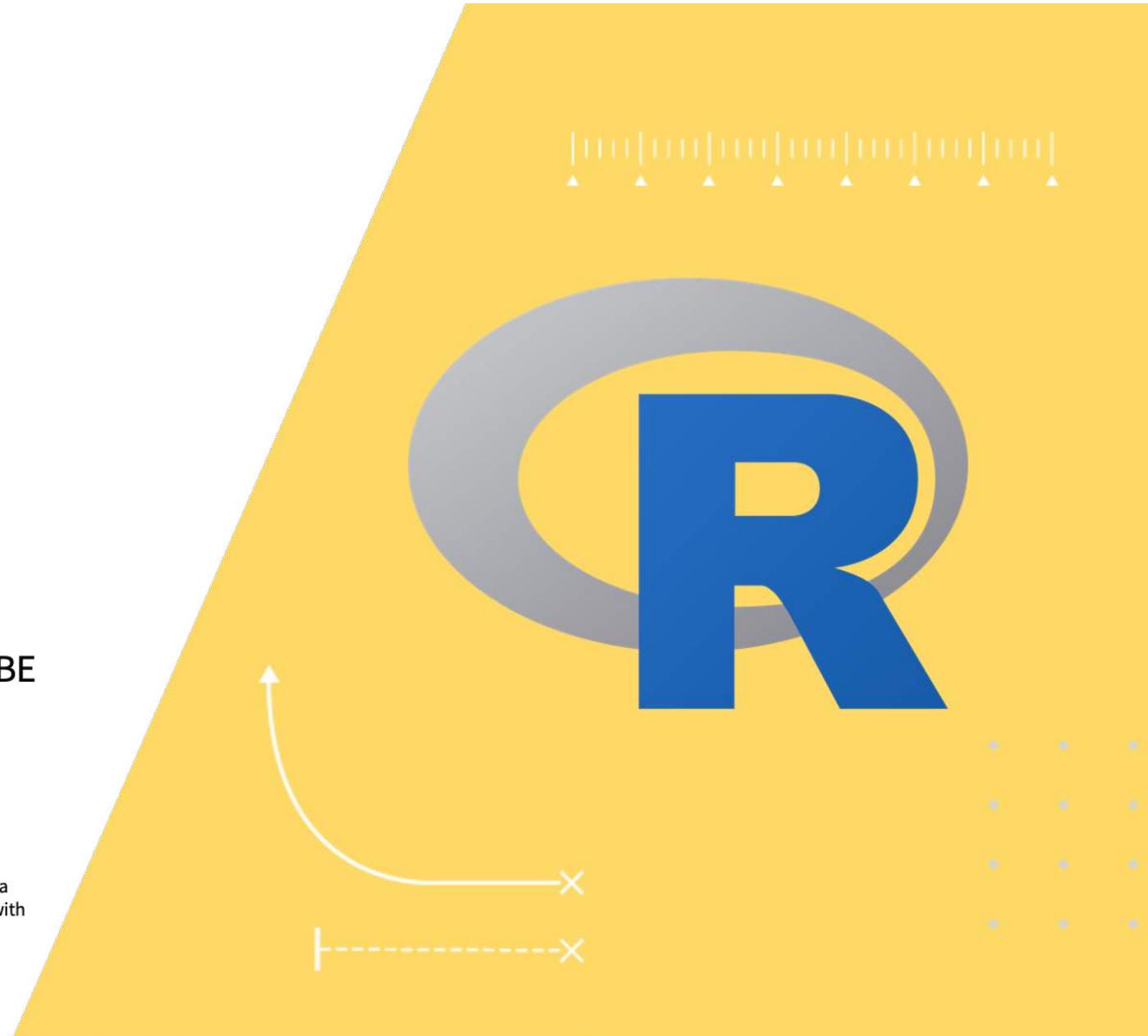
### MÓDULO 1

### INTRODUÇÃO AO R

DESIGNING A NEW VALUATION MODEL  
FOR RURAL PROPERTIES IN PORTUGAL

Formador: Luís Teles Morais | Nova SBE  
*Lisboa, 4 maio 2023*

 This project is carried out with funding by the European Union via the Structural Reform Support Programme and in cooperation with the Directorate General for Structural Reform Support of the European Commission



# Programa

MÓDULOS	DURAÇÃO
<b>Módulo 1 – Introdução ao R:</b> - O que é o R? - Como instalar e configurar o R. - Sintaxe básica e comandos. - Tipos de dados, objetos e classes.	<b>4 Horas</b>
<b>Módulo 2 – Gestão e tratamento de dados em R:</b> - Carregar dados no R. - Perceber as estruturas de dados e <i>subsetting</i> . - Limpeza de dados: <i>missing values</i> , <i>outliers</i> e transformações - Juntar bases de dados	<b>8 Horas</b>
<b>Módulo 3 – Estatística básica em R:</b> - Estatísticas descritivas: medidas de dispersão central e variação. - Distribuições probabilísticas: variáveis discretas e contínuas. - Testes de hipóteses.	<b>8 Horas</b>

MÓDULOS	DURAÇÃO
<b>Módulo 4 – Regressão Linear:</b> - O modelo classico linear. - Estimação de parametros segundo o MMQ. - Testes de hipóteses: significância estatística e ajuste do modelo. - Modelo de regressão múltipla. - Testar as premissas: multicolinearidade, heteroscedasticidade e normalidade dos resíduos. - Critérios de seleção dos modelos.	<b>12 Horas</b>
<b>Módulo 5 – O modelo:</b> - Estrutura do modelo e premissas – Perceber o modelo (4 Hours). - Uso e tratamento dos dados (4 Hours). - Descrição do modelo (4 Hours). - Aplicação do modelo a cada piloto (12 Hours). - Aplicação autónoma do modelo a uma região (8 Hours).	<b>32 Horas</b>

O que é o





Linguagem de  
programação

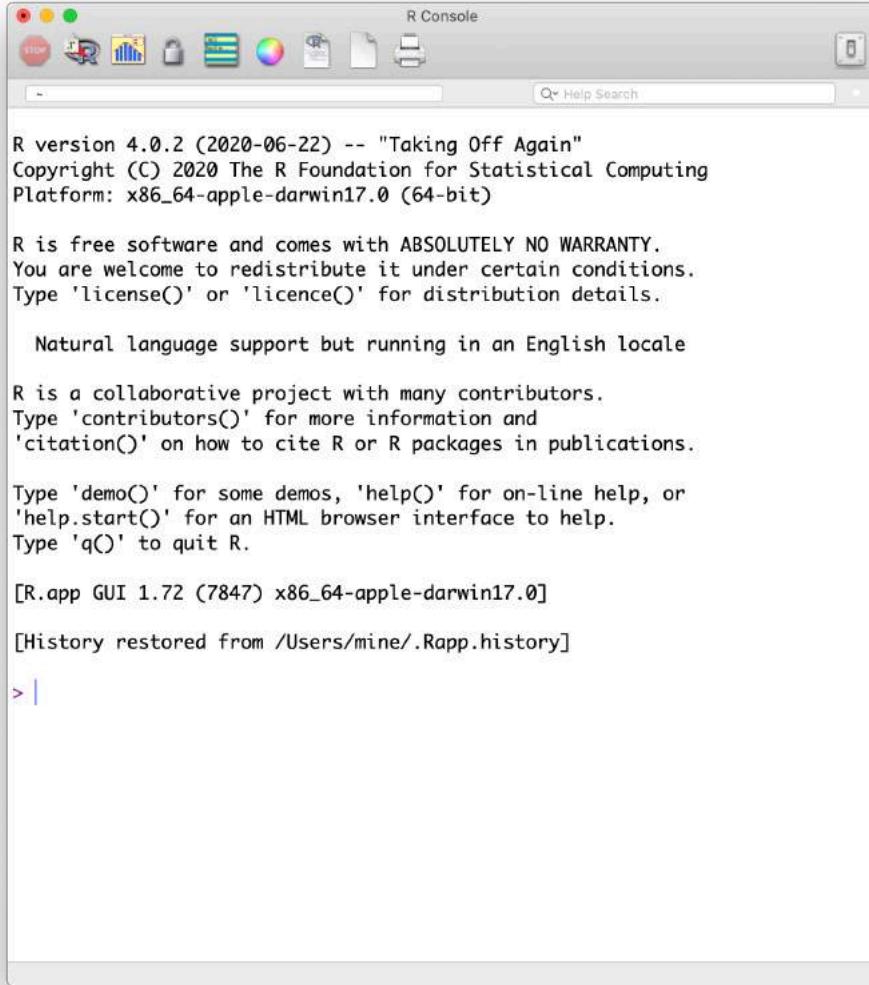


Software de programação e  
edição (IDE, *Integrated Development Environment*)

Vamos escrever código R no RStudio



## Linguagem de programação



R version 4.0.2 (2020-06-22) -- "Taking Off Again"  
Copyright (C) 2020 The R Foundation for Statistical Computing  
Platform: x86\_64-apple-darwin17.0 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.  
You are welcome to redistribute it under certain conditions.  
Type 'license()' or 'licence()' for distribution details.

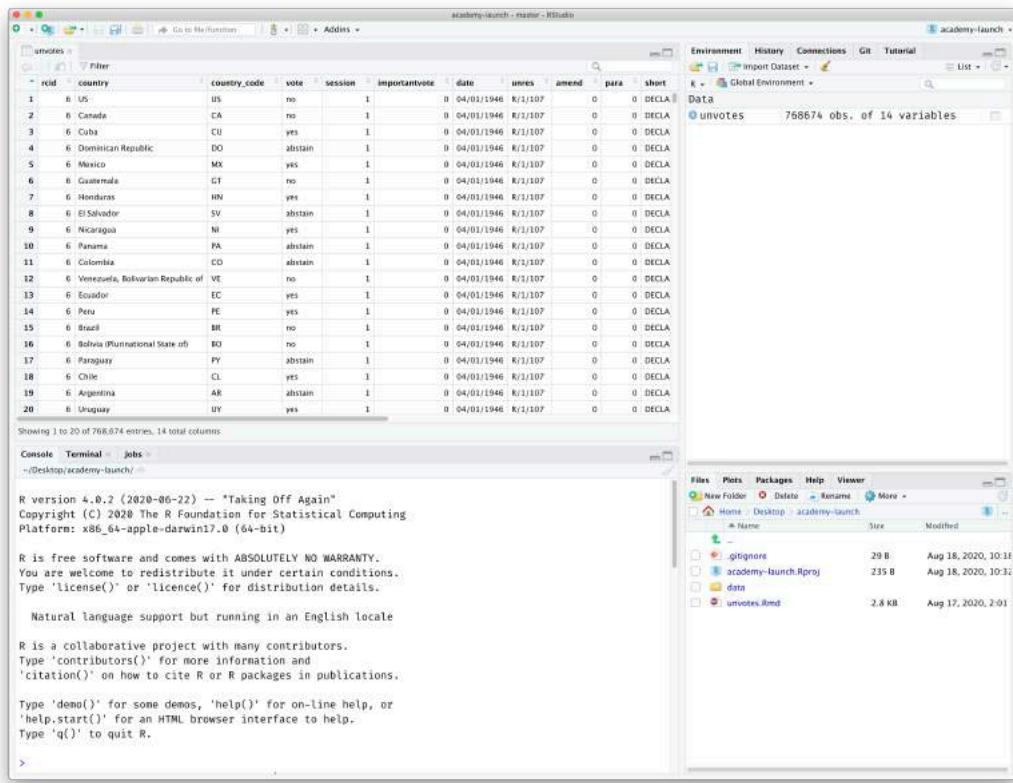
Natural language support but running in an English locale

R is a collaborative project with many contributors.  
Type 'contributors()' for more information and  
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or  
'help.start()' for an HTML browser interface to help.  
Type 'q()' to quit R.

[R.app GUI 1.72 (7847) x86\_64-apple-darwin17.0]  
[History restored from /Users/mine/.Rapp.history]

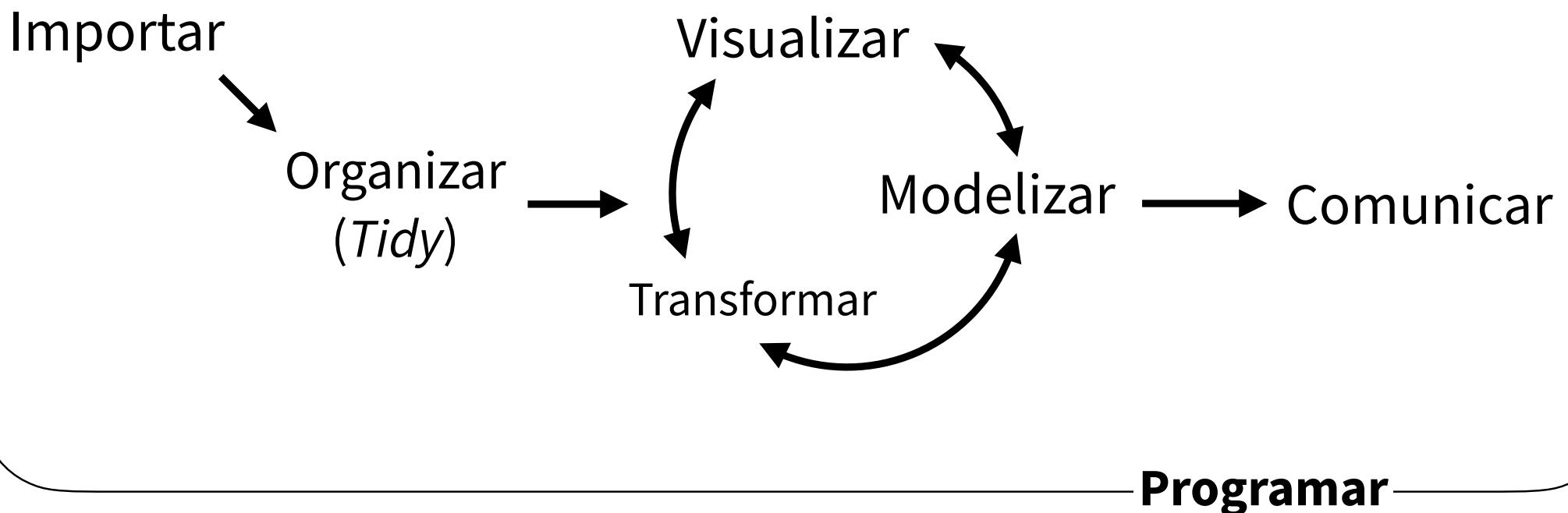
> |



# Software de programação e edição

Para que serve?

# Análise estatística **Ciência de dados**



# **Instalar e configurar**

# Instalação local de ambiente R

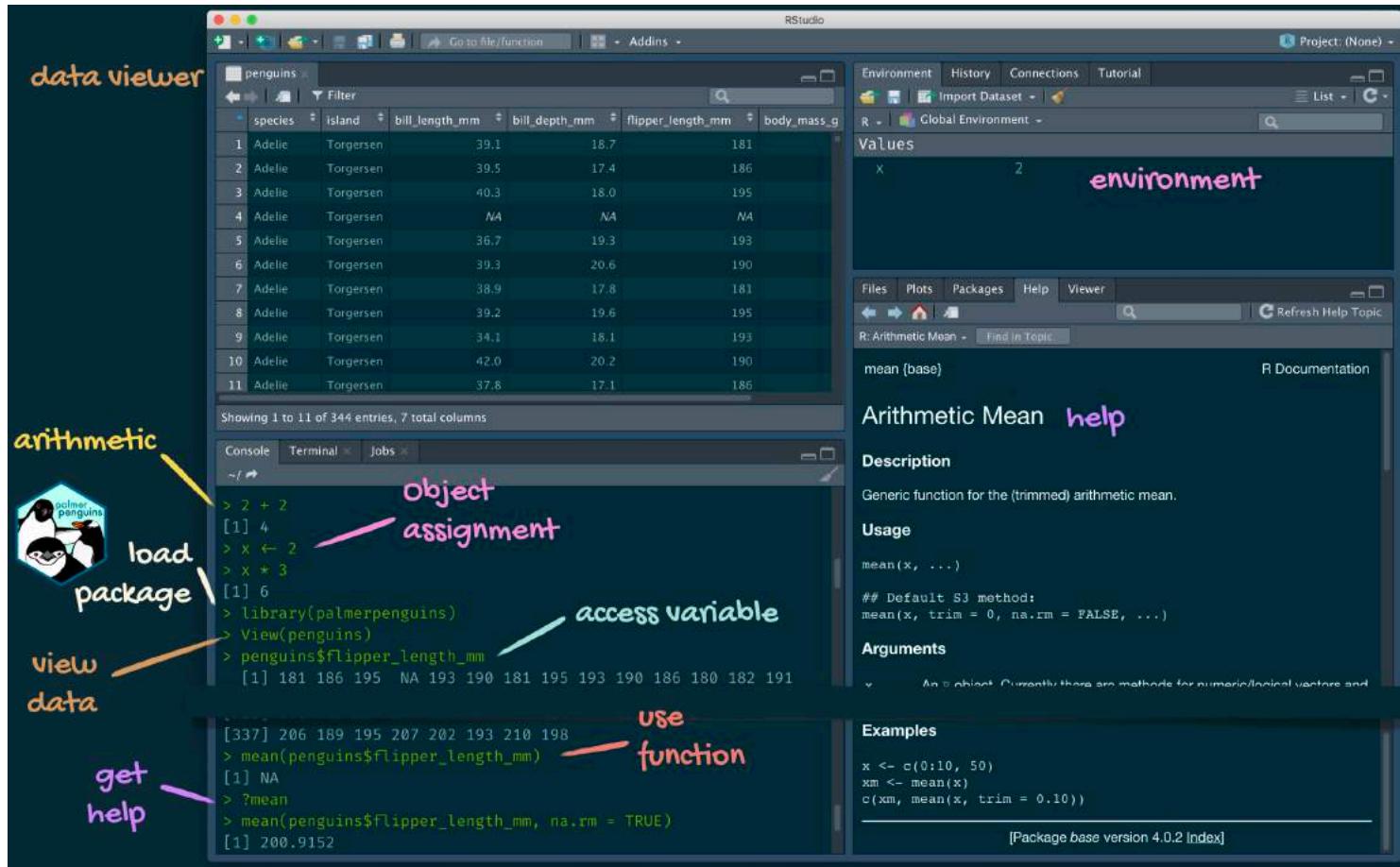
1. Versão recente do R ( $\geq 3.5.0$ )
  - [cran.r-project.org](http://cran.r-project.org)
2. Versão recente do RStudio Desktop ( $\geq 1.1.456$ )
  - [www.rstudio.com/download](http://www.rstudio.com/download)

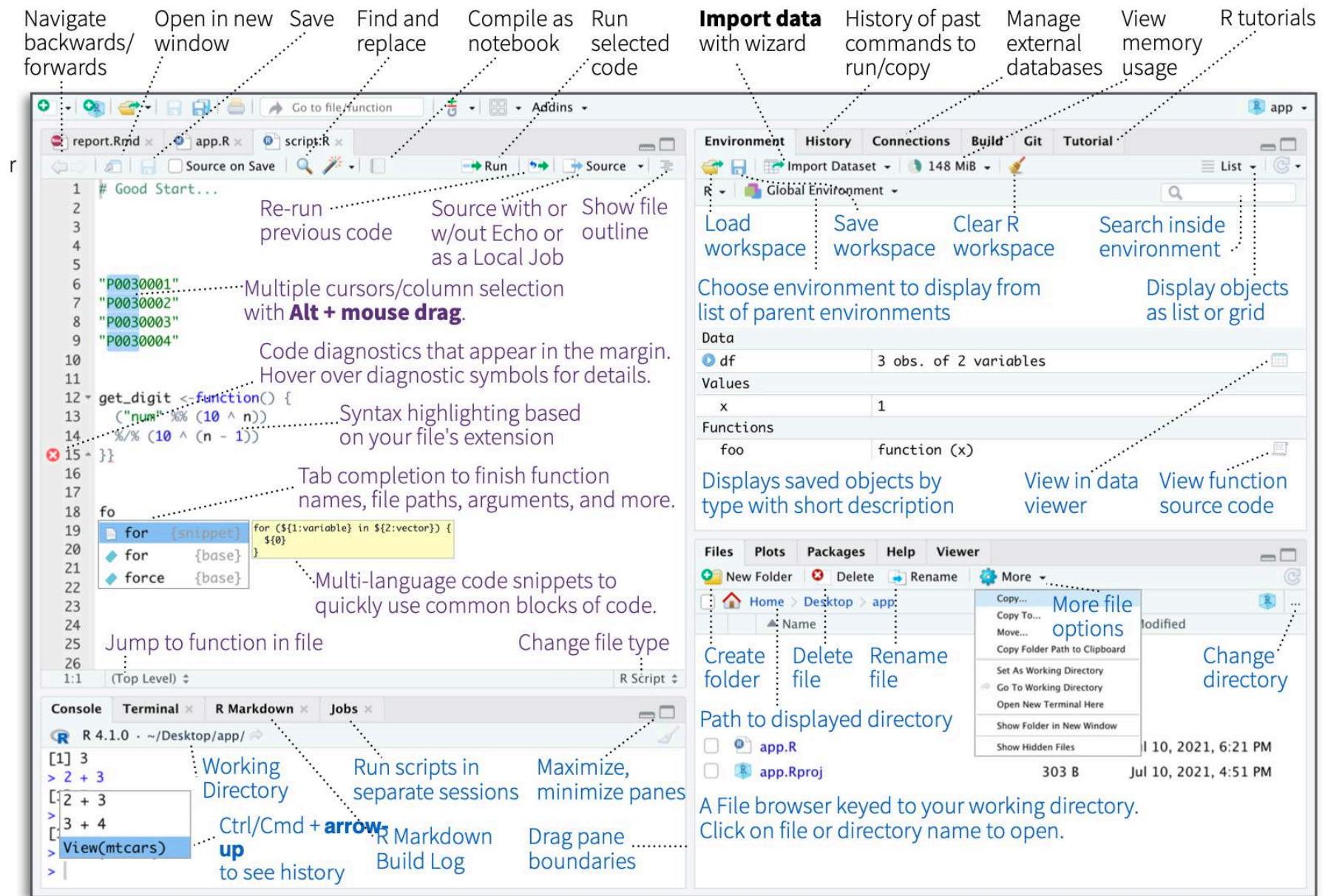
# Vamos a isso

Aceda a este link para começar já

<https://posit.cloud/content/5906356>

# O ambiente R Studio





# Working directory

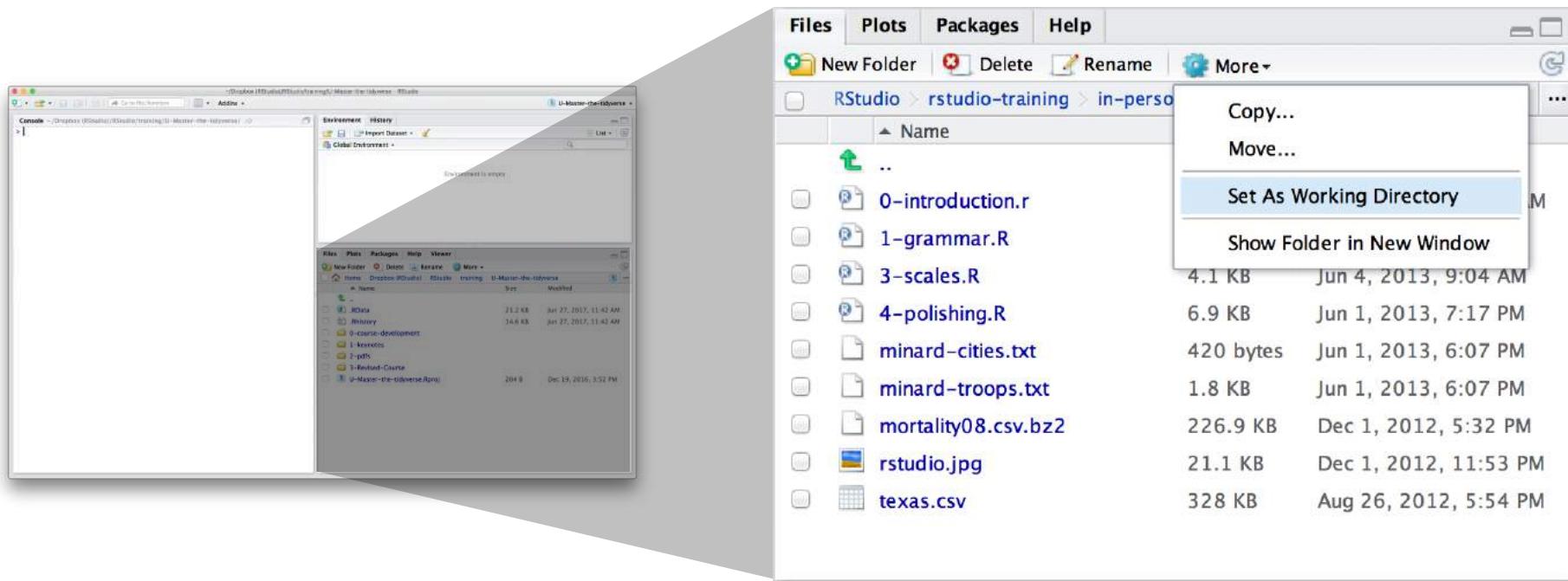
Uma sessão de R está sempre associada a uma pasta específica no computador (ou na *cloud*).

- > "working directory"
- Quaisquer ficheiros serão gravados pelo R nesta pasta
- Quando se pretenda carregar ficheiros, o R tentará encontrá-los nesta pasta



# Alterar Working directory

Navegar para a pasta no painel **Files** e clicar:  
[More > Set As Working Directory](#)



# **Sintaxe básica e comandos**

# Básicos da linguagem R

**Valores** - 1, "Florida", "2010-01-25"

# Básicos da linguagem R

**Valores** - 1, "Florida", "2010-01-25"

**Objectos** - x <- 22/7

Nome sem  
“plicas”

< seguido de -  
(tipo seta p/ direita)

Valor, objeto  
ou resultado  
de função

# Básicos da linguagem R

**Valores** - 1, "Florida", "2010-01-25"

**Objectos** - x <- c(22/7, 0.99, 3)

c() é um objeto composto por vários  
valores (um vetor)

# Básicos da linguagem R

**Valores** - 1, "Florida", "2010-01-25"

**Objectos** - x <- c(22/7, 0.99, 3)

**Funções** - round(x, digits = 3)

Nome sem  
“plicas”

seguido de ()  
para *correr* a  
função

Argumentos da função:  
**valores ou objectos +**  
**opções da função**

# A função mais importante em R



# A função mais importante em R

?geom\_freqpoly

?

nome da função  
(sem parênteses)



geom\_freqpoly {ggplot2}

R Documentation

## Histograms and frequency polygons

### Description

Visualise the distribution of a single continuous variable by dividing the x axis into bins and counting the number of observations in each bin. Histograms (geom\_histogram()) display the counts with bars; frequency polygons (geom\_freqpoly()) display the counts with lines. Frequency polygons are more suitable when you want to compare the distribution across the levels of a categorical variable.

### Usage

```
geom_freqpoly(mapping = NULL, data = NULL, stat = "bin",
  position = "identity", ..., na.rm = FALSE, show.legend = NA,
  inherit.aes = TRUE)

geom_histogram(mapping = NULL, data = NULL, stat = "bin",
  position = "stack", ..., binwidth = NULL, bins = NULL,
```



# Experimente

Porque é que este código não funciona?

```
num_golos <= 835  
num_golos
```

Porque é que este código não funciona?

```
num_golos <= 835  
num_golos
```

# Tipos de dados em R

1. **numeric** (10.5, 55, 787)
2. **character** ou "string" ("ronaldo", "FALSE", "11.5")
3. **logical** ou *boolean* - (TRUE ou FALSE)
4. **integer** (1L, 55L, 100L)
5. **complex** (9 + 3i)

# Só para confirmar

Quais dos seguintes objetos são números?

1

"1"

“um”

um

Quais dos seguintes objetos são números?

1                  "1"                  "one"                  one  
\_\_\_\_\_

números

Quais dos seguintes objetos são números?

1  
número

"1"                "one"                one  

---

*palavras (strings)*

Quais dos seguintes objetos são números?

1  
número

"1"  
palavras (strings)

"one"  
objeto

# Experimente

Quais destes funcionam, depois de executar:

`um <- 1`

`1 * 2`

`“1” * 2`

`“um” * 2`

`um * 2`

Quais destes funcionam, depois de executar:

um <- 1

1 \* 2

“1” \* 2

“um” \* 2

um \* 2

# O poder da consola

```
num_golos <- 835
```

```
1 / 200 * 30  
#> [1] 0.15
```

```
num_golos / 3  
#> [1] 44.66667
```

```
log(2 ** 3)  
#> [1] 1.791759
```

```
num_golos > 1000  
#> [1] FALSE
```

```
num_golos <= 835  
#> [1] TRUE
```

# Lido por humanos

```
num_golos <- 835  
# Número de golos do Ronaldo
```

Lido pelo R

Lido por  
humanos

Comentário

# Lido por humanos

```
num_golos <- 835  
# Número de golos de Ronaldo
```

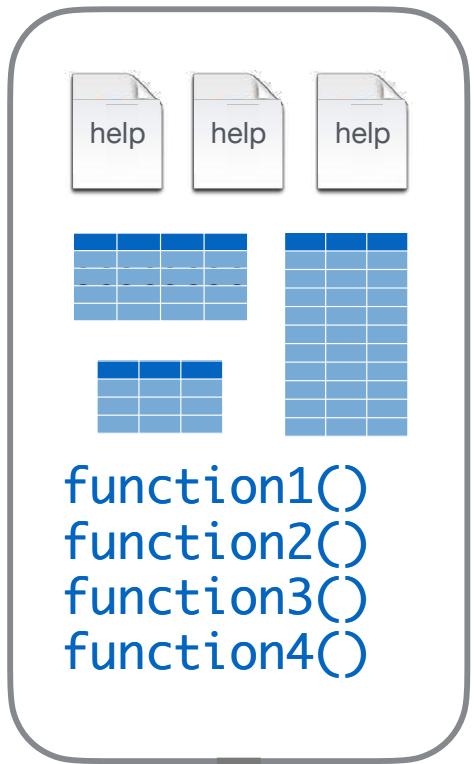
Boas práticas

# Lido por humanos

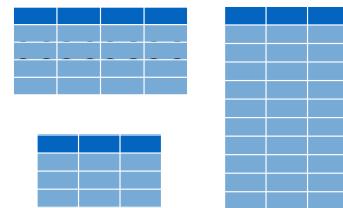
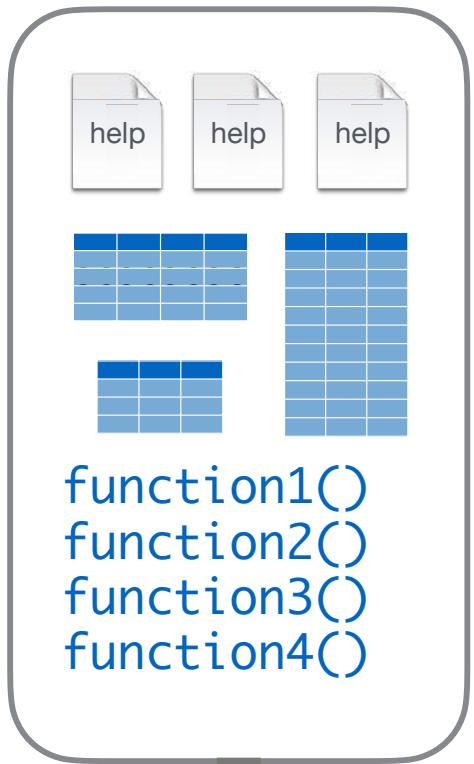
```
num_golos <- 835  
# Número de golos do Borussia
```

Boas práticas

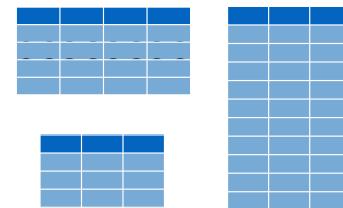
Más prácticas:  
NúmeroDeGolos  
Num.Gls



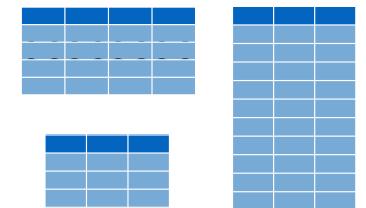
Base R



`function5()`  
`function6()`  
`function7()`  
`function8()`

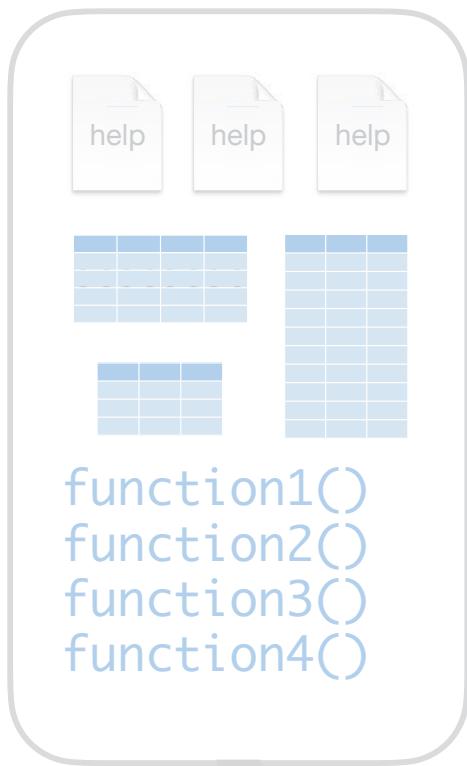


`function9()`  
`functionA()`  
`functionB()`  
`functionC()`

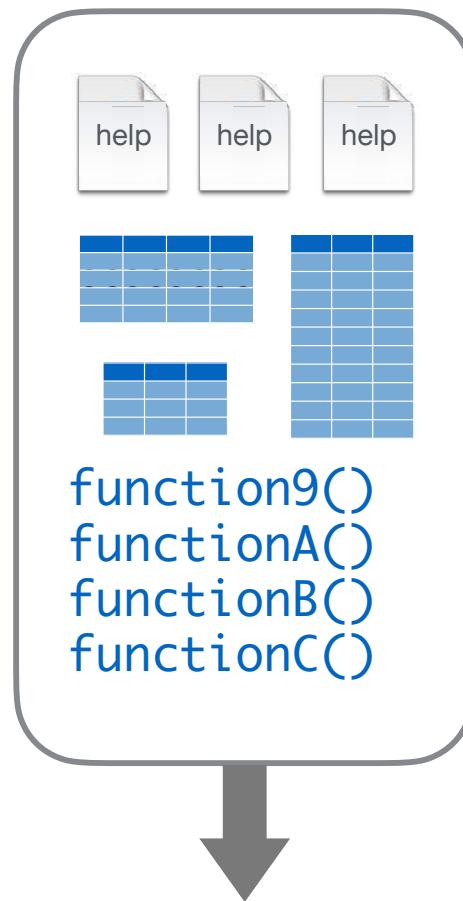
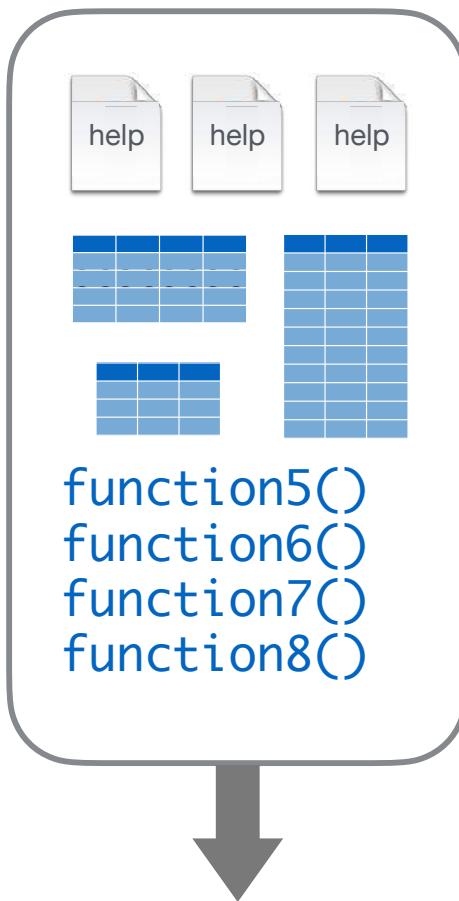


`functionD()`  
`functionE()`  
`functionF()`  
`functionG()`

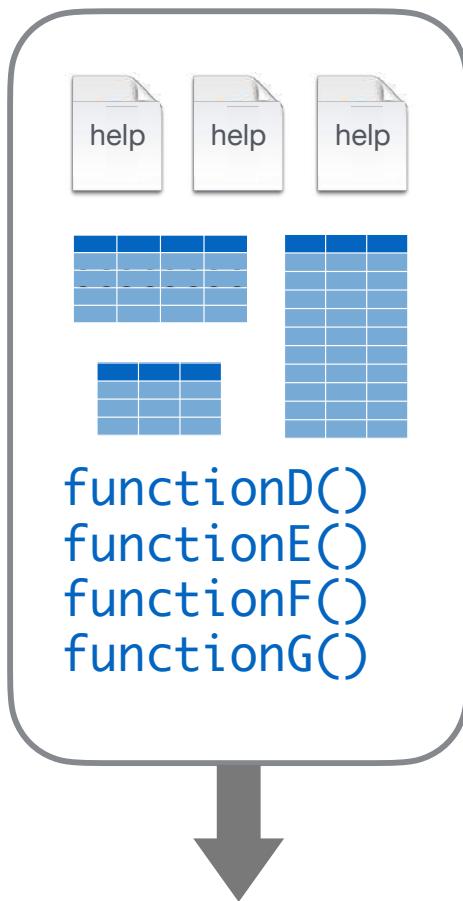
Base R



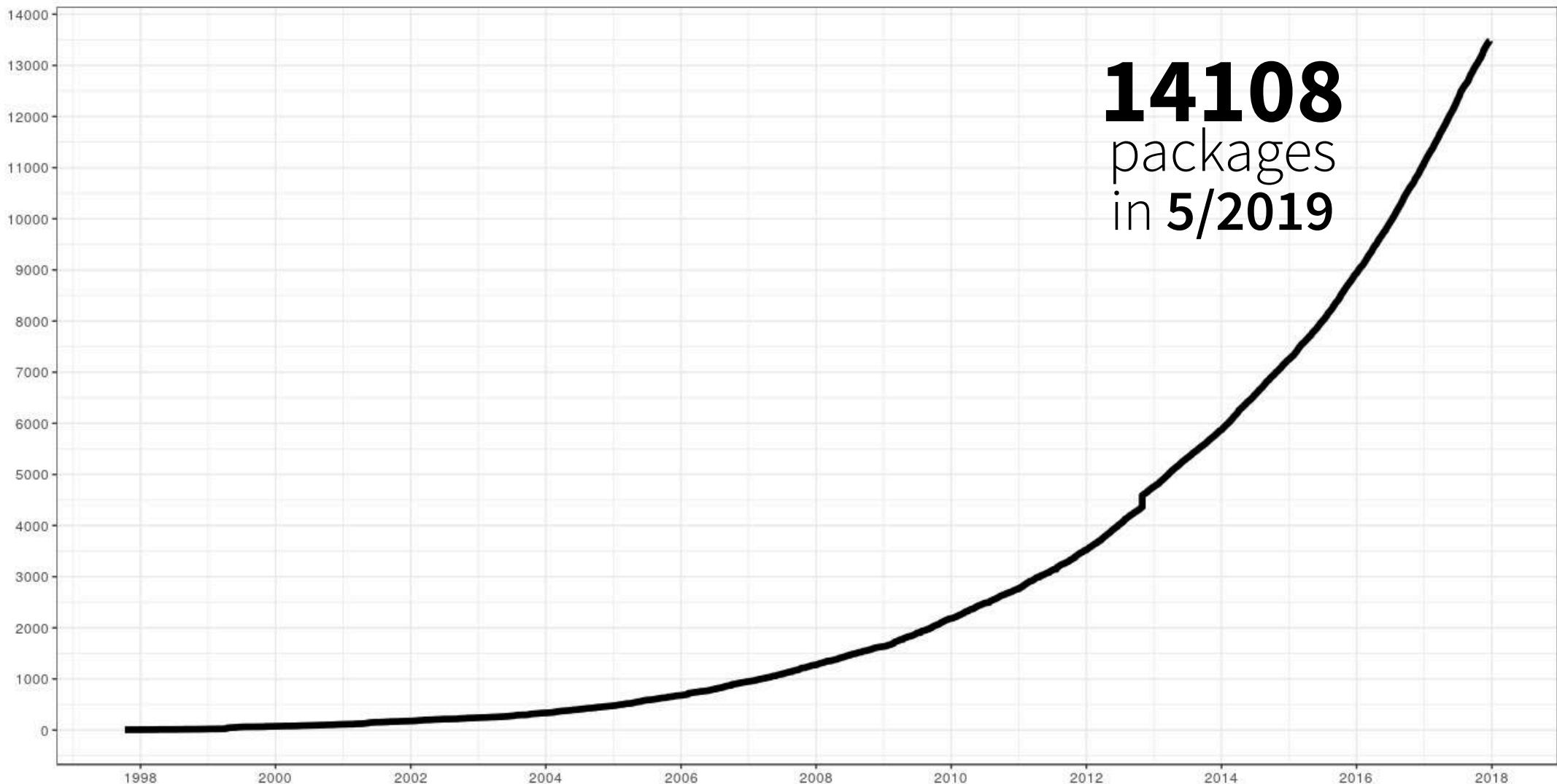
Base R



Pacotes R (**Packages**)



Number of R packages ever published on CRAN



# Como usar pacotes R

**1**

```
install.packages("foo")
```

Descarrega e instala pacotes

**1 x por máquina**

The Comprehensive R Archive 

Secure | <https://cran.r-project.org>



[CRAN](#)  
[Mirrors](#)  
[What's new?](#)  
[Task Views](#)  
[Search](#)

[About R](#)  
[R Homepage](#)  
[The R Journal](#)

[Software](#)  
[R Sources](#)  
[R Binaries](#)  
[Packages](#)  
[Other](#)

[Documentation](#)  
[Manuals](#)  
[FAQs](#)  
[Contributed](#)

[\*\*A3\*\*](#)  
[\*\*abbyyR\*\*](#)  
[\*\*abc\*\*](#)  
[\*\*ABCanalysis\*\*](#)  
[\*\*abc.data\*\*](#)  
[\*\*abcdeFBA\*\*](#)

[\*\*ABCOptim\*\*](#)  
[\*\*ABCp2\*\*](#)  
[\*\*ABC.RAP\*\*](#)  
[\*\*abcrf\*\*](#)  
[\*\*abctools\*\*](#)  
[\*\*abd\*\*](#)  
[\*\*abf2\*\*](#)  
[\*\*ABHgenotypeR\*\*](#)  
[\*\*abind\*\*](#)  
[\*\*abjutils\*\*](#)  
[\*\*abn\*\*](#)  
[\*\*abodOutlier\*\*](#)

**Available CRAN Packages By Name**

[\*\*A\*\*](#) [\*\*B\*\*](#) [\*\*C\*\*](#) [\*\*D\*\*](#) [\*\*E\*\*](#) [\*\*F\*\*](#) [\*\*G\*\*](#) [\*\*H\*\*](#) [\*\*I\*\*](#) [\*\*J\*\*](#) [\*\*K\*\*](#) [\*\*L\*\*](#) [\*\*M\*\*](#) [\*\*N\*\*](#) [\*\*O\*\*](#) [\*\*P\*\*](#) [\*\*Q\*\*](#) [\*\*R\*\*](#) [\*\*S\*\*](#) [\*\*T\*\*](#) [\*\*U\*\*](#) [\*\*V\*\*](#) [\*\*W\*\*](#) [\*\*X\*\*](#) [\*\*Y\*\*](#) [\*\*Z\*\*](#)

Accurate, Adaptable, and Accessible Error Metrics for Predictive Models  
Access to Abbyy Optical Character Recognition (OCR) API  
Tools for Approximate Bayesian Computation (ABC)  
Computed ABC Analysis  
Data Only: Tools for Approximate Bayesian Computation (ABC)  
ABCDE\_FBA: A-Biologist-Can-Do-Everything of Flux Balance Analysis with this package  
Implementation of Artificial Bee Colony (ABC) Optimization  
Approximate Bayesian Computational Model for Estimating P2  
Array Based CpG Region Analysis Pipeline  
Approximate Bayesian Computation via Random Forests  
Tools for ABC Analyses  
The Analysis of Biological Data  
Load Gap-Free Axon ABF2 Files  
Easy Visualization of ABH Genotypes  
Combine Multidimensional Arrays  
Useful Tools for Jurimetical Analysis Used by the Brazilian Jurimetrics Association  
Modelling Multivariate Data with Additive Bayesian Networks  
Angle-Based Outlier Detection

# Mais ajuda

A screenshot of the RStudio interface. The main window shows a data frame titled 'starwars' with columns: name, height, mass, hair\_color, skin\_color, eye\_color, birth\_year, and sex. The data includes rows for Luke Skywalker, C-3PO, R2-D2, Darth Vader, Leia Organa, Owen Lars, Beru Whitesun Lars, R5-D4, and Biggs Darklighter. Below the table, the console shows the command `glimpse(starwars)` and its output: Rows: 87, Columns: 14. The Help menu is open, and a blue arrow points to the 'Cheat Sheets' option in the dropdown menu. A large blue arrow points down from the 'Help' menu towards the bottom right corner of the screen.

R-tidy-intro\_PT - main - RS

name height mass hair\_color skin\_color eye\_color birth\_year sex

1	Luke Skywalker	172	77.0	blond	fair	blue	19.0	male
2	C-3PO	167	75.0	NA	gold	yellow	112.0	none
3	R2-D2	96	32.0	NA	white, blue	red	33.0	none
4	Darth Vader	202	136.0	none	white	yellow	41.9	male
5	Leia Organa	150	49.0	brown	light			
6	Owen Lars	178	120.0	brown, grey	light			
7	Beru Whitesun Lars	165	75.0	brown	light			
8	R5-D4	97	32.0	NA	white, red			
9	Biggs Darklighter	183	84.0	black	light			

Showing 1 to 9 of 87 entries, 14 total columns

Console Background Jobs

```
R 4.2.2 · ~/GitHub/R-tidy-intro_PT/ ↵
> glimpse(starwars)
Rows: 87
Columns: 14
```

Help

Pesquisa

R Help

Search R Help

About RStudio

Check for Updates

Accessibility

RStudio Docs

RStudio Community Forum

Cheat Sheets

Keyboard Shortcuts Help

Markdown Quick Reference

Roxygen Quick Reference

Diagnostics

R: United Nations General Assembly voting

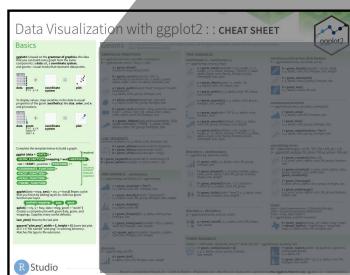
un\_votes {unvotes}

United Nations General



# Sem batota

# Cheat sheets



**Geoms** Use a geom function to represent data points, use the geom's aesthetic properties to represent variables. Each function returns a layer.

**GRAPHICAL PRIMITIVES**

```
a <- ggplot(economics, aes(date, unemploy))
b <- ggplot(seals, aes(x = long, y = lat))

a + geom_blank()
#(Useful for expanding limits)
```

**b + geom\_curve(aes(yend = lat + 1, xend = long + 1), curvature = 2)) -> x, yend, y, vend, alpha, angle, color, curvature, linetype, size**

**a + geom\_path(linewidth = "butt", linejoin = "round", linmitre = 1)** x, y, alpha, color, group, linetype, size

**a + geom\_polygon(aes(group = group))** x, y, alpha, color, fill, group, linetype, size

**b + geom\_rect(aes(xmin = long, ymin = lat, xmax = long + 1, ymax = lat + 1))** -> xmax, xmin, ymax, ymin, alpha, color, fill, linetype, size

**a + geom\_ribbon(aes(ymin = unemploy - 900, ymax = unemploy + 900))** -> x, ymax, ymin, alpha, color, fill, group, linetype, size

**LINE SEGMENTS** common aesthetics: x, y, alpha, color, linetype, size

```
b + geom_abline(aes(intercept = 0, slope = 1))
b + geom_hline(aes(intercept = lat))
b + geom_vline(aes(intercept = long))

b + geom_segment(aes(yend = lat + 1, xend = long + 1))
b + geom_spoke(aes(angle = 1:115, radius = 1))
```

**ONE VARIABLE continuous**

```
c <- ggplot(mpg, aes(hwy)); c2 <- ggplot(mpg)

c + geom_area(stat = "bin")
x, y, alpha, color, fill, group, linetype, size
```

**c + geom\_density(kernel = "gaussian")** x, y, alpha, color, fill, group, linetype, size, weight

**c + geom\_dotplot()** x, y, alpha, color, fill

**c + geom\_freqpoly()** x, y, alpha, color, group, linetype, size

**c + geom\_histogram(binwidth = 5)** x, y, alpha, color, fill, linetype, size, weight

**c2 + geom\_qq(aes(sample = hwy))** x, y, alpha, color, fill, linetype, size, weight

**discrete**

```
d <- ggplot(mpg, aes(f1))
d + geom_bar()
```

**continuous x , continuous y**

```
e + geom_label(aes(label = ctv), nudge_x = 1, nudge_y = 1, check_overlap = TRUE)
x, y, label, alpha, angle, color, family, fontface, hjust, lineheight, size, vjust
```

**e + geom\_jitter(height = 2, width = 2)** x, y, alpha, color, fill, shape, size

**e + geom\_point()** x, y, alpha, color, fill, shape, size, stroke

**e + geom\_quantile()** x, y, alpha, color, group, linetype, size, weight

**e + geom\_rug(sides = "bl")** x, y, alpha, color, linetype, size

**e + geom\_smooth(method = lm)** x, y, alpha, color, fill, group, linetype, size, weight

**e + geom\_text(aes(label = ctv), nudge\_x = 1, nudge\_y = 1, check\_overlap = TRUE)** x, y, label, alpha, angle, color, family, fontface, hjust, lineheight, size, vjust

**discrete x , continuous y**

```
f <- ggplot(df, aes(grp, fit, ymin = fit - se, ymax = fit + se))

f + geom_col()
x, y, alpha, color, fill, group, linetype, size
```

**f + geom\_boxplot()** x, y, lower, middle, upper, ymax, ymin, alpha, color, fill, group, linetype, shape, size, weight

**f + geom\_dotplot(binaxis = "y", stackdir = "center")** x, y, alpha, color, fill, group

**f + geom\_violin(scale = "area")** x, y, alpha, color, fill, group, linetype, size, weight

**discrete x , discrete y**

```
g <- ggplot(diamonds, aes(cut, color))

g + geom_count()
x, y, alpha, color, fill, shape, size, stroke
```

**continuous bivariate distribution**

```
h <- ggplot(diamonds, aes(carat, price))

h + geom_bin2d(binwidth = c(0.25, 500))
x, y, alpha, color, fill, linetype, size, weight
```

**continuous function**

```
i <- ggplot(economics, aes(date, unemploy))

i + geom_area()
x, y, alpha, color, fill, linetype, size
```

**i + geom\_line()** x, y, alpha, color, group, linetype, size

**i + geom\_step(direction = "hv")** x, y, alpha, color, group, linetype, size

**visualizing error**

```
j <- data.frame(grp = c("A", "B"), fit = 4.5, se = 1:2)
j <- ggplot(j, aes(grp, fit, ymin = fit - se, ymax = fit + se))

j + geom_crossbar(fatten = 2)
x, y, ymax, ymin, alpha, color, fill, group, linetype, size
```

**j + geom\_errorbar()** x, ymax, ymin, alpha, color, fill, group, linetype, size, width (also geom\_errorbarh())

**j + geom\_linerange()** x, ymin, ymax, alpha, color, group, linetype, size

**j + geom\_pointrange()** x, y, ymin, ymax, alpha, color, fill, group, linetype, shape, size

**maps**

```
data <- data.frame(murder = USArrests$Murder,
state = tolower(rownames(USArrests)))
map <- map_data("state")
k <- ggplot(data, aes(fill = murder))

k + geom_map(aes(map_id = state), map = map)
+ expand_limits(x = map$long, y = map$lat),
map_id, alpha, color, fill, linetype, size
```

**THREE VARIABLES**

```
seals$z <- with(seals, sqrt(delta_long^2 + delta_lat^2)) |<- ggplot(seals, aes(long, lat))

l + geom_contour(aes(z = z))
x, y, z, alpha, colour, group, linetype, size, weight
```

**l + geom\_raster(aes(fill = z), hjust = 0.5, vjust = 0.5, interpolate = FALSE)** x, y, alpha, fill

**l + geom\_tile(aes(fill = z))** x, y, alpha, color, fill, linetype, size, width



# tidyverse

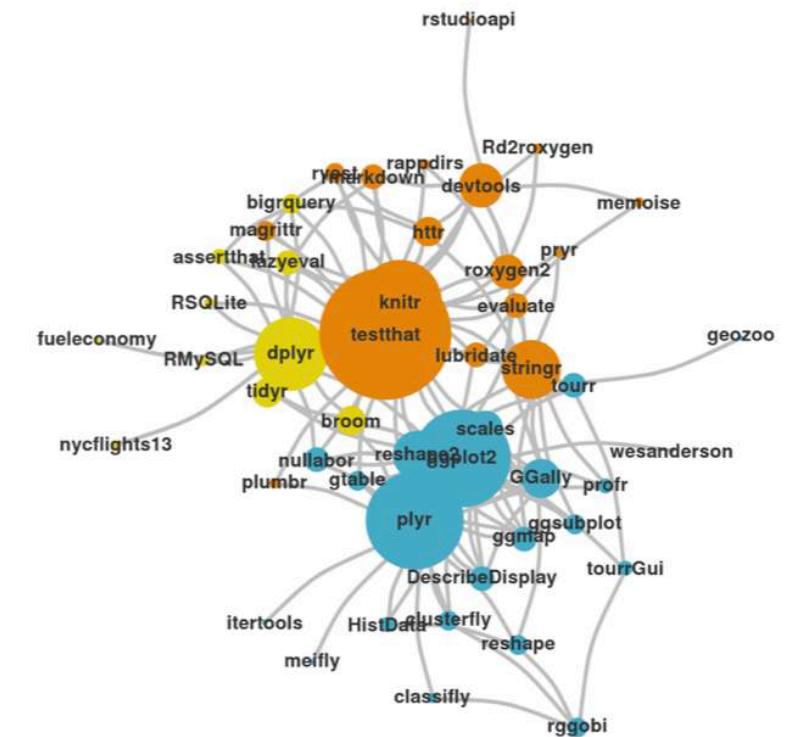


# Bem-vindos ao Tidyverse

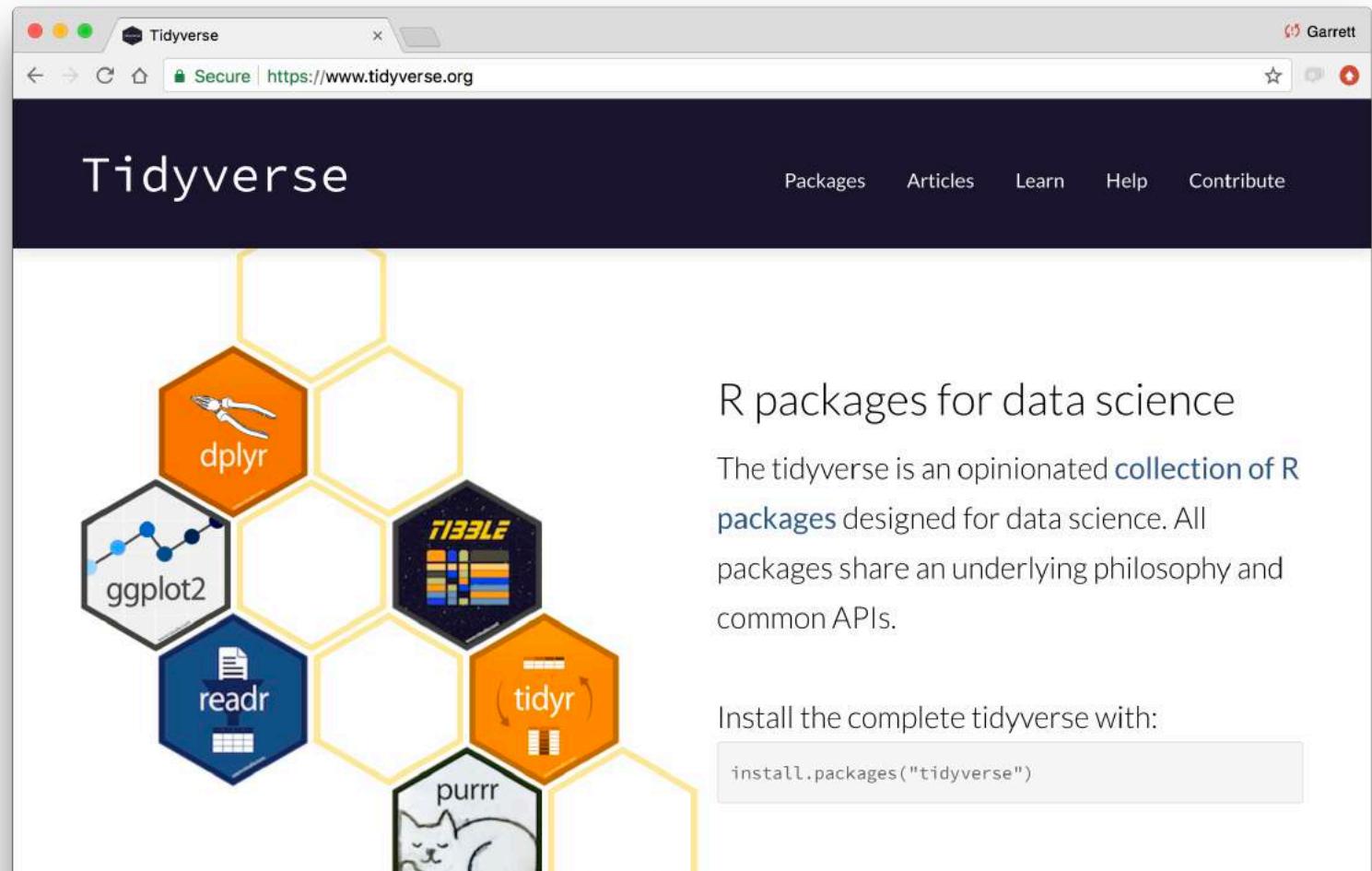


# Bem-vindos ao Tidyverse

- Uma coleção de pacotes R modernos, com uma filosofia comum e que funcionam em conjunto
- **Tudo o que precisa para trabalhar com dados em R**



# tidyverse.org



The screenshot shows the tidyverse.org homepage in a web browser. The page has a dark header with the word "Tidyverse" and navigation links for "Packages", "Articles", "Learn", "Help", and "Contribute". Below the header is a large graphic featuring six hexagonal icons arranged in a hexagonal pattern, representing the core packages of the tidyverse: dplyr (orange, top), ggplot2 (grey, bottom-left), readr (blue, middle-left), purrr (light orange, bottom-right), tibble (dark blue, middle), and tidyr (orange, top-right). To the right of the graphic, the text reads "R packages for data science" followed by a detailed description of what the tidyverse is. At the bottom, there is a section titled "Install the complete tidyverse with:" containing the R command "install.packages("tidyverse")".

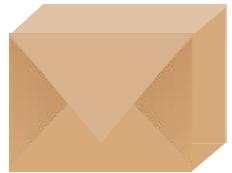
R packages for data science

The tidyverse is an opinionated collection of R packages designed for data science. All packages share an underlying philosophy and common APIs.

Install the complete tidyverse with:

```
install.packages("tidyverse")
```

# tidyverse



Um pacote que serve de atalho para instalar e carregar todos os componentes do **tidyverse**

```
install.packages("tidyverse")
```

```
install.packages("tidyverse")
```

equivale a

```
install.packages("ggplot2")
install.packages("dplyr")
install.packages("tidyr")
install.packages("readr")
install.packages("purrr")
install.packages("tibble")
install.packages("hms")
install.packages("stringr")
install.packages("lubridate")
install.packages("forcats")
install.packages("DBI")
install.packages("haven")
install.packages("httr")
install.packages("jsonlite")
install.packages("readxl")
install.packages("rvest")
install.packages("xml2")
install.packages("modelr")
install.packages("broom")
```

# Como usar pacotes R

**1**

```
install.packages("foo")
```

Descarrega e instala pacotes

**1 x por máquina**

**2**

```
library("foo")
```

Carrega pacote

**1 x por sessão de R**

```
install.packages("tidyverse")
```

equivale a

```
install.packages("ggplot2")
install.packages("dplyr")
install.packages("tidyr")
install.packages("readr")
install.packages("purrr")
install.packages("tibble")
install.packages("stringr")
install.packages("forcats")
install.packages("lubridate")
install.packages("hms")
install.packages("DBI")
install.packages("haven")
install.packages("httr")
install.packages("jsonlite")
install.packages("readxl")
install.packages("rvest")
install.packages("xml2")
install.packages("modelr")
install.packages("broom")
```

```
library("tidyverse")
```

equivale a

```
library("ggplot2")
library("dplyr")
library("tidyr")
library("readr")
library("purrr")
library("tibble")
library("stringr")
library("forcats")
```

**O que entendemos por  
dados?**

# Dados



nome masculino

1. cada um dos elementos conhecidos de um problema
2. base para a formação de um juízo ou cálculo
3. informação
4. **INFORMÁTICA** informação que pode ser aceite, armazenada, tratada ou fornecida pelo computador

Dados → Dataset  
*(Conjunto de dados)*

- Cada coluna é uma **variável**
- Cada linha é uma **observação**

# Dados → Dataset

(Conjunto de dados)

Ex.: `starwars`

```
## # A tibble: 87 × 14
##   name    height  mass hair_color skin_color eye_color birth_year
##   <chr>     <int> <dbl> <chr>       <chr>      <chr>        <dbl>
## 1 Luke S...     172    77 blond      fair       blue          19
## 2 C-3PO        167    75 <NA>       gold       yellow        112
## 3 R2-D2         96    32 <NA>      white, bl... red          33
## 4 Darth ...     202   136 none      white       yellow        41.9
## 5 Leia O...     150    49 brown     light       brown          19
## 6 Owen L...     178   120 brown, gr... light       blue          52
## # ... with 81 more rows, and 7 more variables: sex <chr>,
## #   gender <chr>, homeworld <chr>, species <chr>, films <list>,
## #   vehicles <list>, starships <list>
```

- R: `data.frame` ou `tibble`

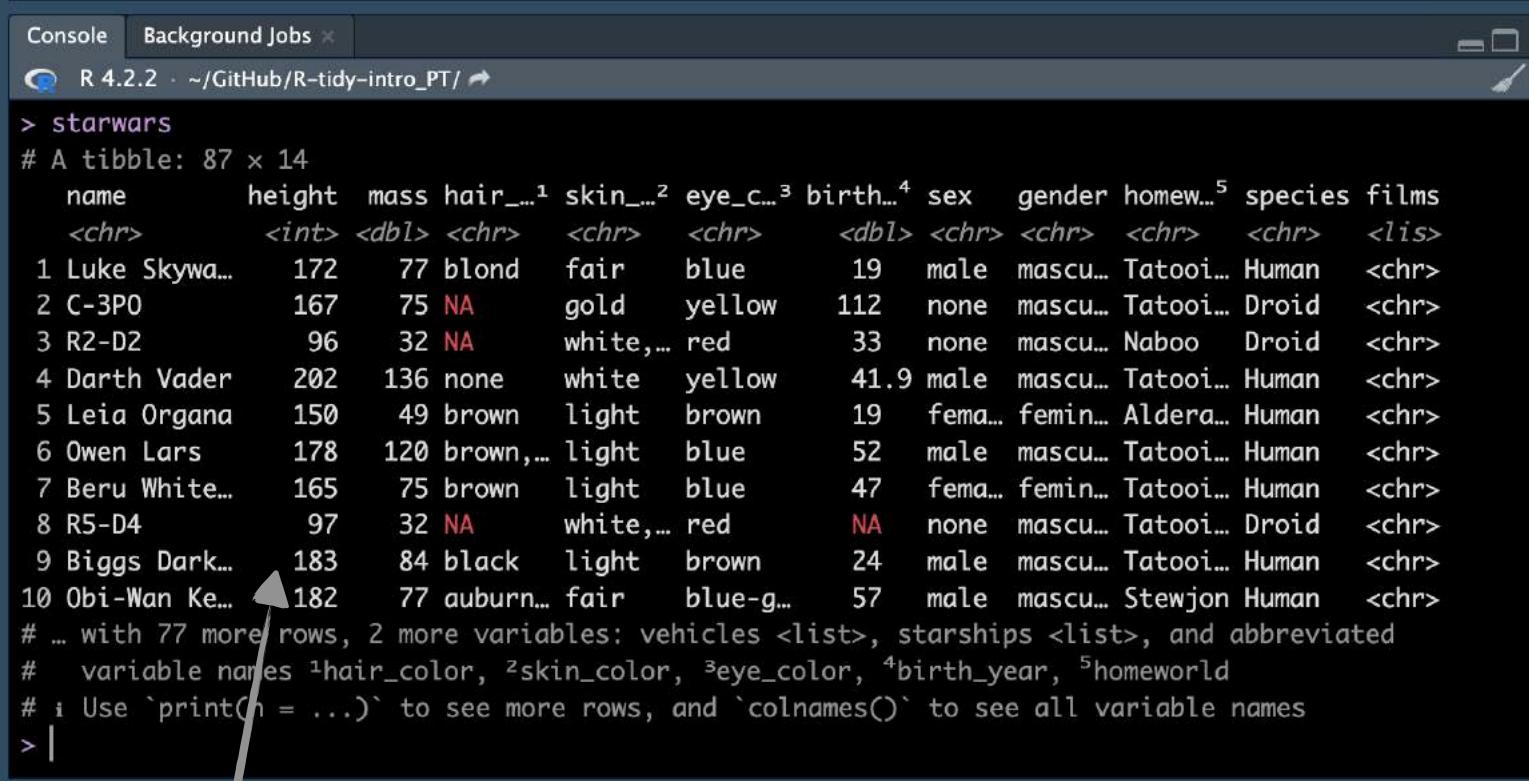
- Cada coluna é uma **variável**
- Cada linha é uma **observação**



## ● Uma observação

```
height = 172 cm      name = "Luke Skywalker"  
weight = 77 kg        hair_color = "blond"  
eye_color = "blue"    birth_year = 19 BBY  
skin_color = "fair"   films = c("The Empire Strikes Back",  
species = "Human"     "Revenge of the Sith",  
sex = "male"          "Return of the Jedi",  
gender = "masculine"  "A New Hope",  
homeworld = "Tatooine" "The Force Awakens")  
                                     vehicles = c("Snowspeeder",  
                                     "Imperial Speeder Bike")  
                                     starships = c("X-wing",  
                                     "Imperial shuttle")
```

- Dar uma olhadela: `starwars`



```
Console Background Jobs x
R 4.2.2 · ~/GitHub/R-tidy-intro_PT/ ↗
> starwars
# A tibble: 87 × 14
  name      height  mass hair_color¹ skin_color² eye_color³ birth_year⁴ sex   gender homeworld⁵ species films
  <chr>     <int> <dbl> <chr>    <chr>    <chr>    <dbl> <chr> <chr> <chr> <chr> <chr>
1 Luke Skywalker 172     77 blond    fair     blue      19 male   masculin Tatooine Human  <chr>
2 C-3PO          167     75 NA       gold     yellow    112 none   masculin Tatooine Droid  <chr>
3 R2-D2          96      32 NA       white,... red      33 none   masculin Naboo   Droid  <chr>
4 Darth Vader    202     136 none    white     yellow    41.9 male   masculin Tatooine Human  <chr>
5 Leia Organa    150     49 brown   light     brown     19 female feminin Alderaan Human  <chr>
6 Owen Lars      178     120 brown,... light     blue      52 male   masculin Tatooine Human  <chr>
7 Beru White...   165     75 brown   light     blue      47 female feminin Tatooine Human  <chr>
8 R5-D4          97      32 NA       white,... red      NA none   masculin Tatooine Droid  <chr>
9 Biggs Dark...   183     84 black   light     brown     24 male   masculin Tatooine Human  <chr>
10 Obi-Wan Kenobi 182     77 auburn... fair     blue-green 57 male   masculin Stewjon Human  <chr>
# ... with 77 more rows, 2 more variables: vehicles <list>, starships <list>, and abbreviated
#   variable names ¹hair_color, ²skin_color, ³eye_color, ⁴birth_year, ⁵homeworld
#   Use `print(n = ...)` to see more rows, and `colnames()` to see all variable names
> |
```

- Uma variável `starwars$height`

- Dar uma olhadela: `view(starwars)`

The screenshot shows the RStudio interface with the starwars dataset loaded. The top panel displays a grid viewer with 19 rows of data and 14 columns. The columns are labeled: name, height, mass, hair\_color, skin\_color, eye\_color, birth\_year, sex, and gender. The bottom panel shows the R console output, which includes the first few rows of the data, information about the structure of the dataset, and a note about abbreviations.

	name	height	mass	hair_color	skin_color	eye_color	birth_year	sex	gender
1	Luke Skywalker	172	77.0	blond	fair	blue	19.0	male	masculine
2	C-3PO	167	75.0	NA	gold	yellow	112.0	none	masculine
3	R2-D2	96	32.0	NA	white, blue	red	33.0	none	masculine
4	Darth Vader	202	136.0	none	white	yellow	41.9	male	masculine
5	Leia Organa	150	49.0	brown	light	brown	19.0	female	feminine
6	Owen Lars	178	120.0	brown, grey	light	blue	52.0	male	masculine
7	Beru Whitesun Lars	165	75.0	brown	light	blue	47.0	female	feminine
8	R5-D4	97	32.0	NA	white, red	red	NA	none	masculine
9	Biggs Darklighter	183	84.0	black	light	brown	24.0	male	masculine
10	Obi-Wan Kenobi	182	77.0	auburn, white	fair	blue-gray	57.0	male	masculine
11	Anakin Skywalker	188	84.0	blond	fair	blue	41.9	male	masculine
12	Wilhuff Tarkin	180	NA	auburn, grey	fair	blue	64.0	male	masculine
13	Chewbacca	228	112.0	brown	unknown	blue	200.0	male	masculine
14	Han Solo	180	80.0	brown	fair	brown	29.0	male	masculine
15	Greedo	173	74.0	NA	green	black	44.0	male	masculine
16	Jabba Desilijic Tiure	175	1358.0	NA	green-tan, brown	orange	600.0	hermaphroditic	masculine
17	Wedge Antilles	170	77.0	brown	fair	hazel	21.0	male	masculine
18	Jek Tono Porkins	180	110.0	brown	fair	blue	NA	male	masculine

Showing 1 to 19 of 87 entries, 14 total columns

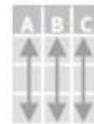
```

Console Background Jobs
R 4.2.2 - ~/GitHub/R-tidy-intro_PT/
9 Biggs Dark... 183 84 black light brown 24 male masculin... Tatooi... Human <chr>
10 Obi-Wan Ke... 182 77 auburn... fair blue-g... 57 male masculin... Stewjon Human <chr>
# ... with 77 more rows, 2 more variables: vehicles <list>, starships <list>, and abbreviated
#   variable names `hair_color`, `skin_color`, `eye_color`, `birth_year`, `homeworld`
# i Use `print(n = ...)` to see more rows, and `colnames()` to see all variable names
> view(starwars)
> |

```

Dados → Dataset  
*(Conjunto de dados)*

- Cada coluna é uma **variável**
- Cada linha é uma **observação**



Each **variable** is in  
its own **column**

Each **observation**, or  
**case**, is in its own **row**

Sempre?

Dados → Dataset  
*(Conjunto de dados)*

- Cada coluna é uma **variável**
- Cada linha é uma **observação**



Each **variable** is in its own **column** & Each **observation**, or **case**, is in its own **row**

Sempre?

Nem sempre. Só com dados bem arrumados (**tidy**)

# Exemplos de dados não-tidy

---

TABELAS DE RETENÇÃO NA FONTE PARA O CONTINENTE - 2023 Semestre 1

TABELA I - TRABALHO DEPENDENTE

NÃO CASADO

Remuneração Mensal Euros	Número de dependentes					
	0	1	2	3	4	5 ou mais
Até 762.00	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
Até 790.00	2.0%	0.4%	0.0%	0.0%	0.0%	0.0%
Até 812.00	4.0%	0.7%	0.0%	0.0%	0.0%	0.0%
Até 863.00	7.0%	4.4%	0.9%	0.0%	0.0%	0.0%
Até 964.00	9.3%	6.6%	3.4%	0.0%	0.0%	0.0%
Até 1 051.00	11.2%	7.8%	5.6%	1.3%	0.0%	0.0%
Até 1 113.00	12.0%	8.7%	6.4%	3.2%	0.0%	0.0%
Até 1 194.00	13.0%	10.6%	8.2%	5.0%	2.6%	0.1%
Até 1 280.00	14.0%	11.7%	9.2%	6.0%	3.5%	1.1%
Até 1 380.00	15.1%	12.7%	10.4%	6.9%	4.5%	2.1%
Até 1 466.00	16.2%	13.8%	11.4%	8.0%	6.5%	4.0%
Até 1 609.00	17.2%	14.8%	12.3%	10.0%	7.5%	5.0%

# Exemplos de dados não-tidy

LISTA DAS TAXAS IMI POR MUNICÍPIO DO DISTRITO BEJA PARA O ANO 2021					
Código Municipio	Município	Prédios Urbanos Avaliados nos Termos do CIMI	Prédios Rústicos	Taxas por freguesia	Dedução fixa por agregado
0201	ALJUSTREL	0,3000 %	0,80 %		
0202	ALMODOVAR	0,3000 %	0,80 %		+Info
0203	ALVITO	0,3000 %	0,80 %		+Info
0204	BARRANCOS	-	-	+Info	+Info
0205	BEJA	0,3200 %	0,80 %		+Info
0206	CASTRO VERDE	0,3000 %	0,80 %		
0207	CUBA	0,3000 %	0,80 %		
0208	FERREIRA DO ALENTEJO	0,3750 %	0,80 %		+Info
0209	MERTOLA	0,3750 %	0,80 %		+Info

# Exemplos de dados não-tidy

	A	AA	AB	AC	AD	AE	AF	AG	AH
1	Estimated HIV Prevalence% - (Ages 15-49)	2004	2005	2006	2007	2008	2009	2010	2011
2	Abkhazia						0.06	0.06	0.06
3	Afghanistan								
4	Akrotiri and Dhekelia								
5	Albania								
6	Algeria	0.1	0.1	0.1	0.1	0.1			
7	American Samoa								
8	Andorra								
9	Angola	1.9	1.9	1.9	1.9	2.1	2.1	2.1	
10	Anguilla								
11	Antigua and Barbuda								
12	Argentina	0.4	0.4	0.4	0.4	0.5	0.4	0.4	0.4
13	Armenia	0.1	0.1	0.1	0.1	0.1	0.2	0.2	0.2
14	Aruba								
15	Australia	0.1	0.1	0.1	0.1	0.1	0.2	0.2	0.2
16	Austria	0.2	0.2	0.2	0.3	0.3	0.3	0.4	0.4
17	Azerbaijan	0.06	0.06	0.06	0.1	0.1	0.1	0.1	0.1
18	Bahamas	3	3	3	3.1	3.1	2.9	2.8	2.8

# **Visualização de dados no R**

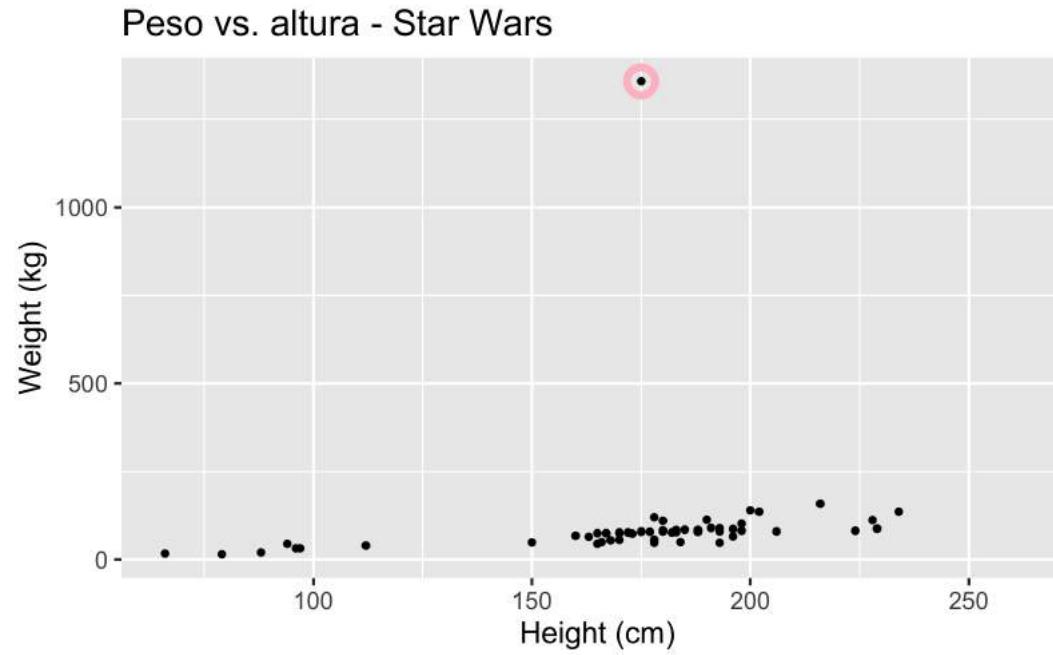
# Análise exploratória de dados

- Análise a conjuntos de dados de forma a summarizar e entender as suas principais características
- Muitas vezes, é mais fácil fazê-lo visualmente com gráficos (agora)
- Mas também pode ser necessitar de manipulação e transformação dos dados, e cálculo de estatísticas básicas (próximas sessões)



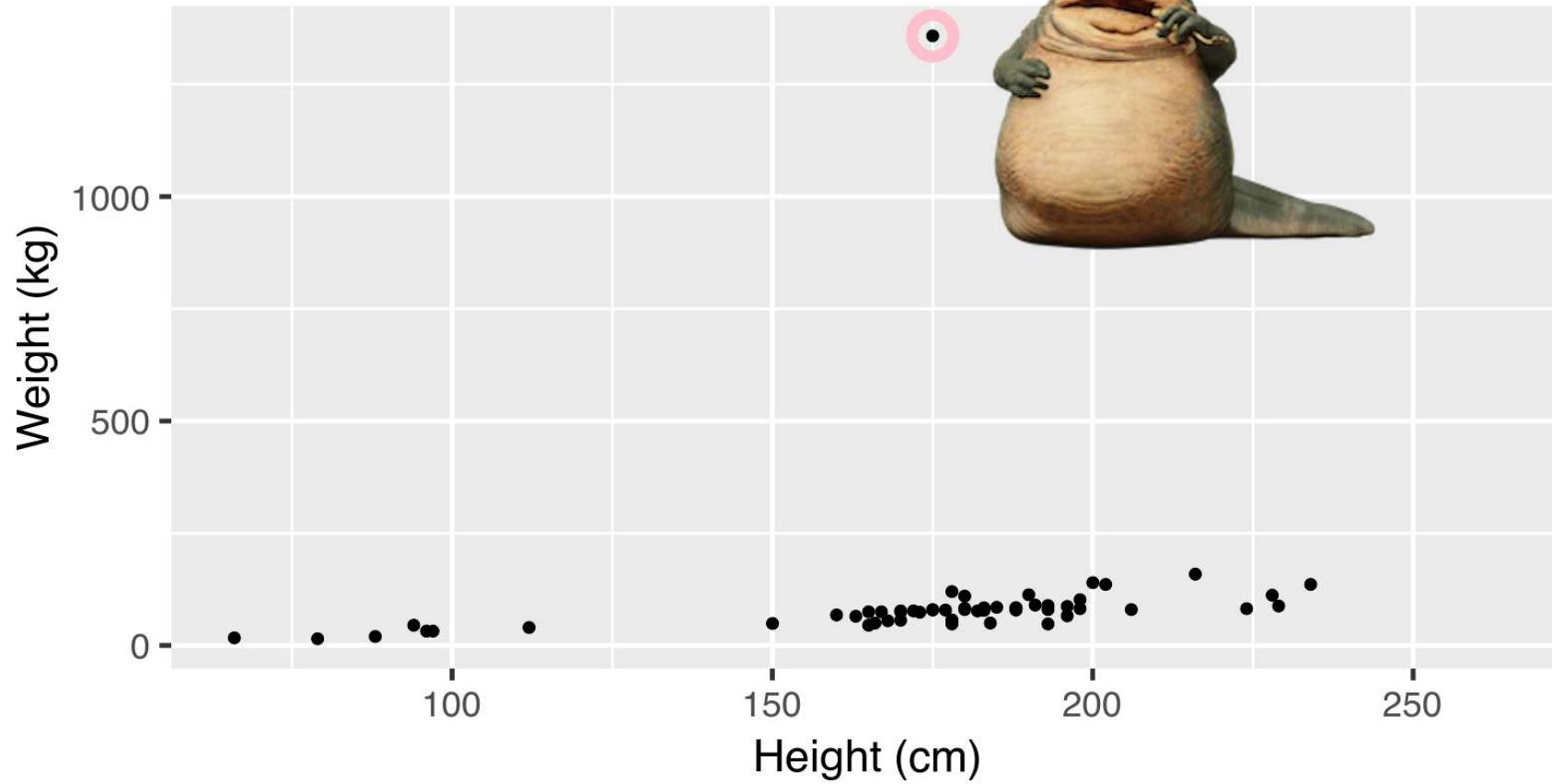
# Peso vs. altura

Qual é a relação peso vs. altura entre as personagens do Star Wars? Haverá outras variáveis que possam ajudar a compreender esta relação?

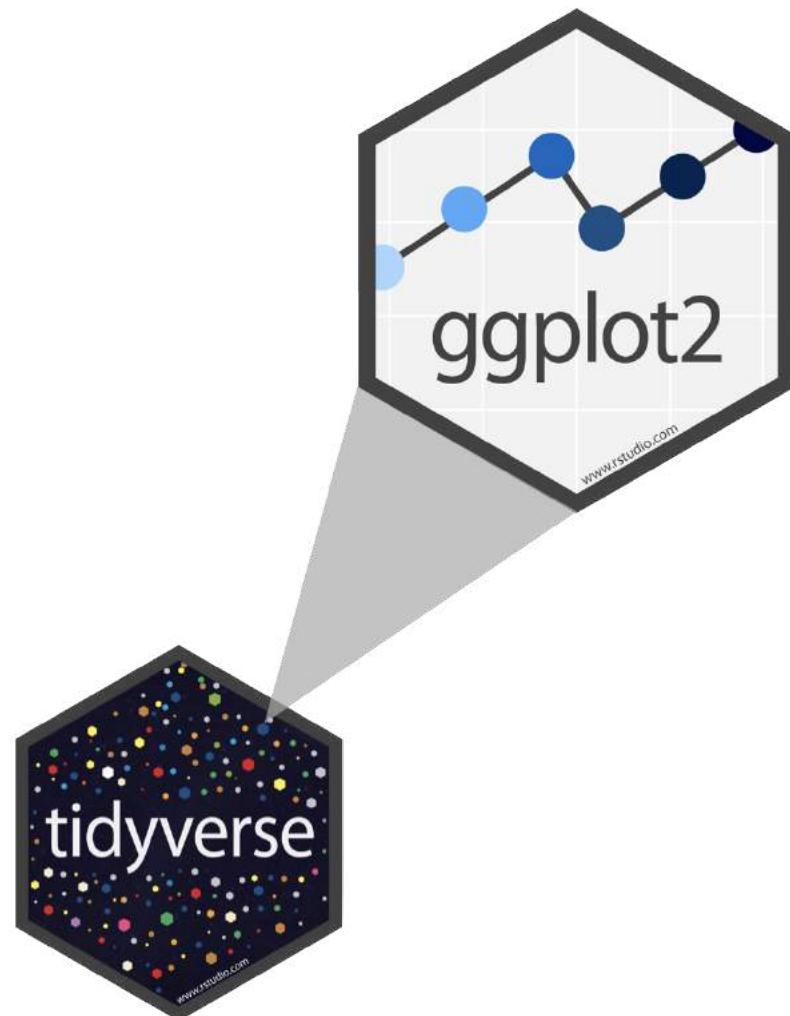


# Jabba!

## Peso vs. altura - Star Wars



# Visualização de dados no tidyverse = **ggplot**

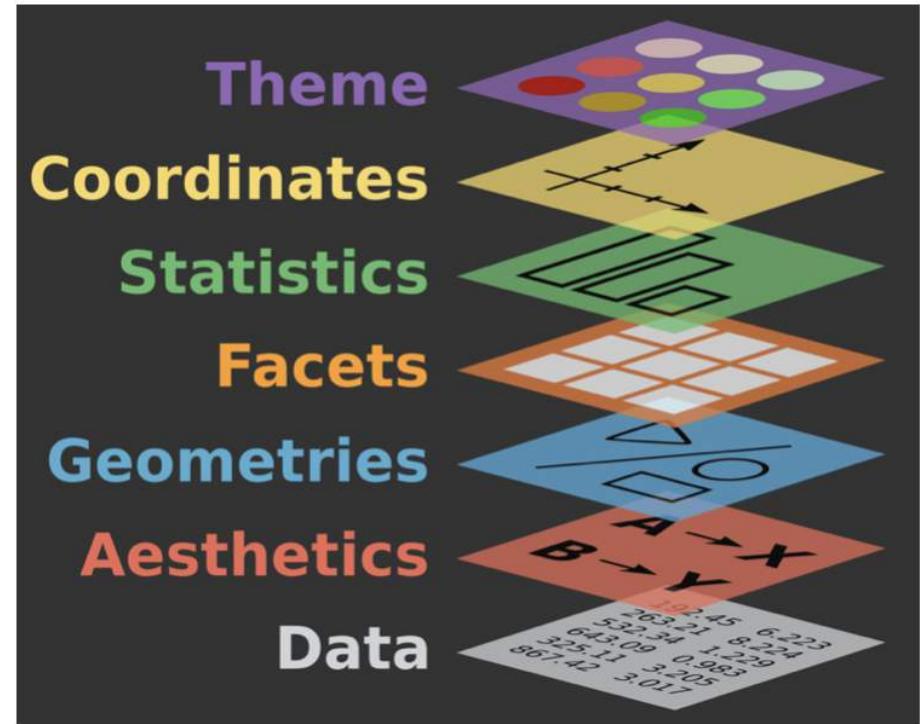
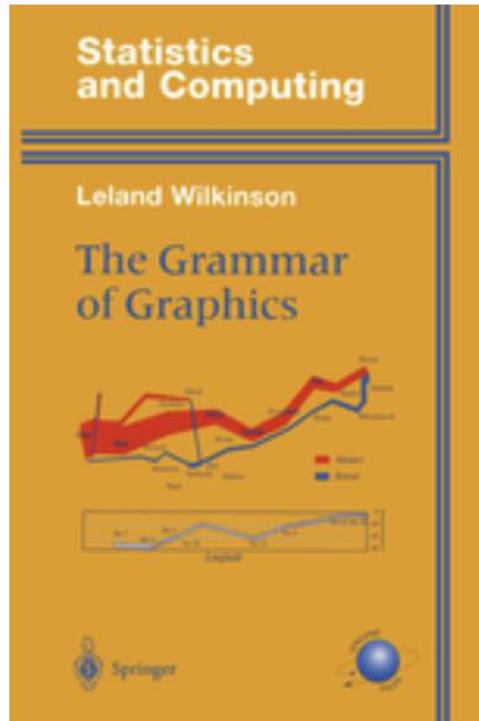


- gg = Gramática dos Gráficos ( *Grammar of Graphics* )
- Inspirado pelo livro **Grammar of Graphics** de Leland Wilkinson
- Estrutura básica do código

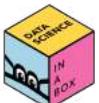
```
ggplot(data = [dataset],  
       mapping = aes(x = [x-variable],  
                      y = [y-variable])) +  
  geom_xxx() +  
  other options
```

# Gramática dos Gráficos

Descrição geral dos vários componentes que **qualquer** gráfico tem

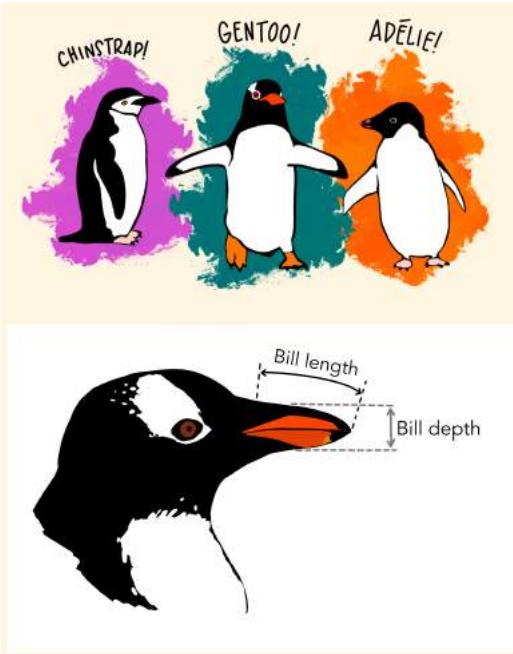


Source: BloggoType



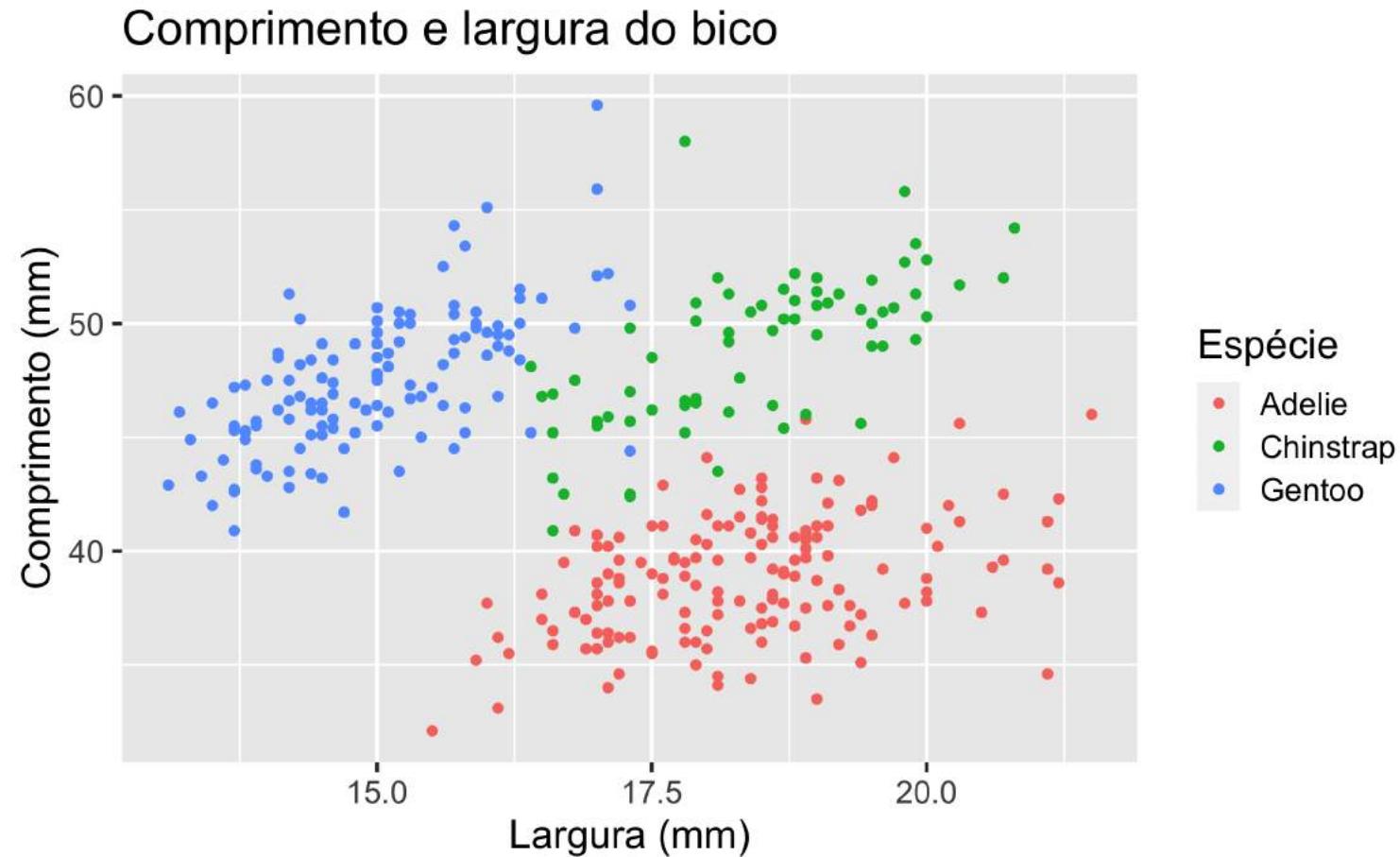
# Dados: Pinguins de Palmer

Medições feitas a diferentes espécies de pinguins numa ilha do Archipelago Palmer:  
dimensões (barbatanas, bico, peso) e sexo.



```
library(palmerpenguins)  
glimpse(penguins)
```

```
## # Rows: 344  
## # Columns: 8  
## # $ species <fct> Adelie, Adelie, Adelie, Adelie,  
## # $ island <fct> Torgersen, Torgersen, Torgersen, Torg  
## # $ bill_length_mm <dbl> 39.1, 39.5, 40.3, NA, 36.7, 39.3, 38.  
## # $ bill_depth_mm <dbl> 18.7, 17.4, 18.0, NA, 19.3, 20.6, 17.  
## # $ flipper_length_mm <int> 181, 186, 195, NA, 193, 190, 181, 195  
## # $ body_mass_g <int> 3750, 3800, 3250, NA, 3450, 3650, 3620  
## # $ sex <fct> male, female, female, NA, female, male  
## # $ year <int> 2007, 2007, 2007, 2007, 2007, 2007, 2007
```

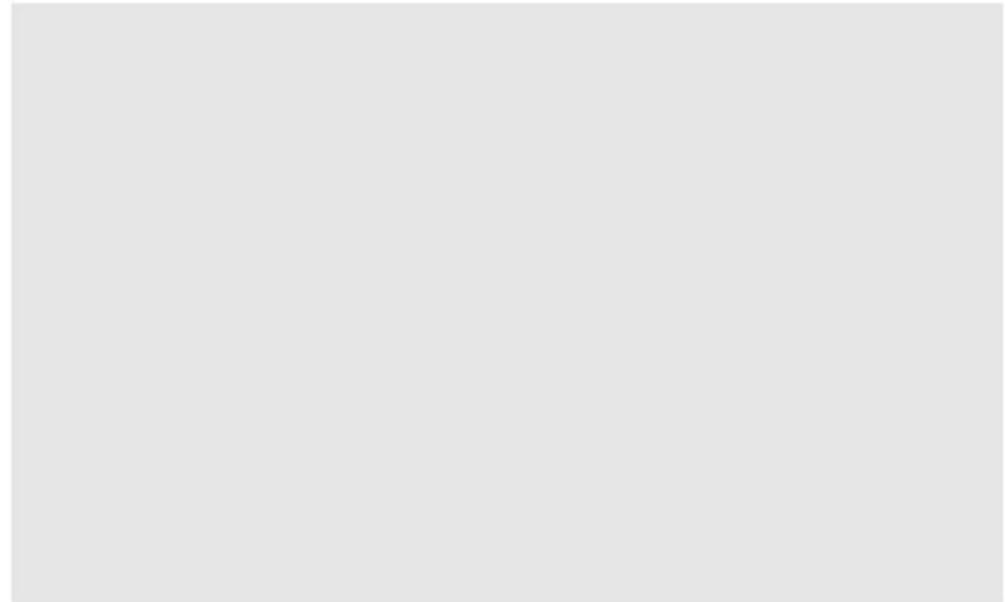


# Construção de um gráfico no ggplot2



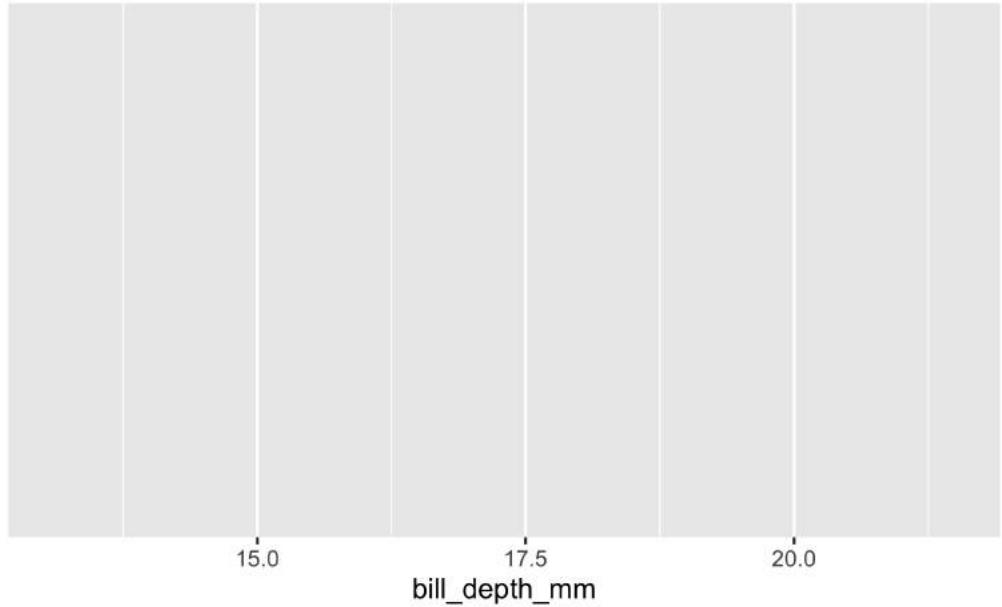
## Começamos com o data frame penguins

```
ggplot(data = penguins)
```



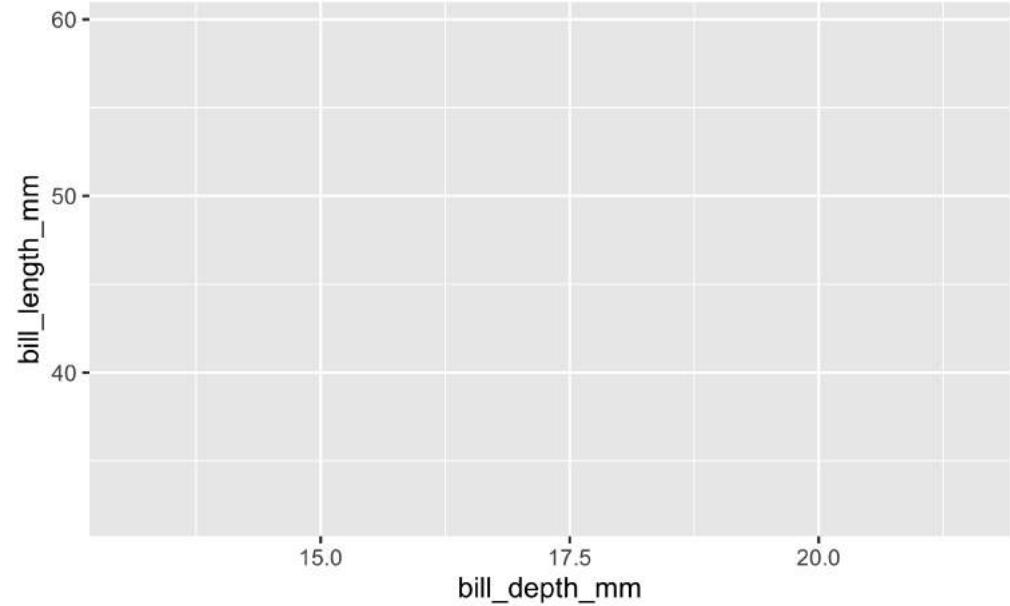
Começamos com o data frame `penguins`, **mapeamos a largura do bico ao eixo dos x**

```
ggplot(data = penguins,  
       mapping = aes(x = bill_depth_mm))
```



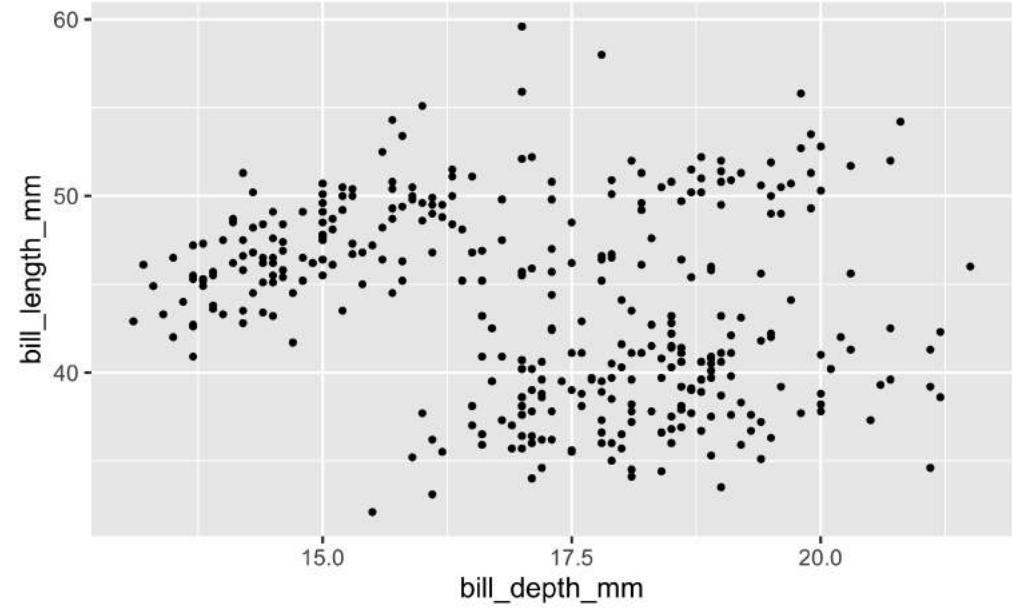
Começamos com o data frame `penguins`, mapeamos a largura do bico ao eixo dos x **e o comprimento ao eixo dos y**.

```
ggplot(data = penguins,  
       mapping = aes(x = bill_depth_mm,  
                      y = bill_length_mm))
```



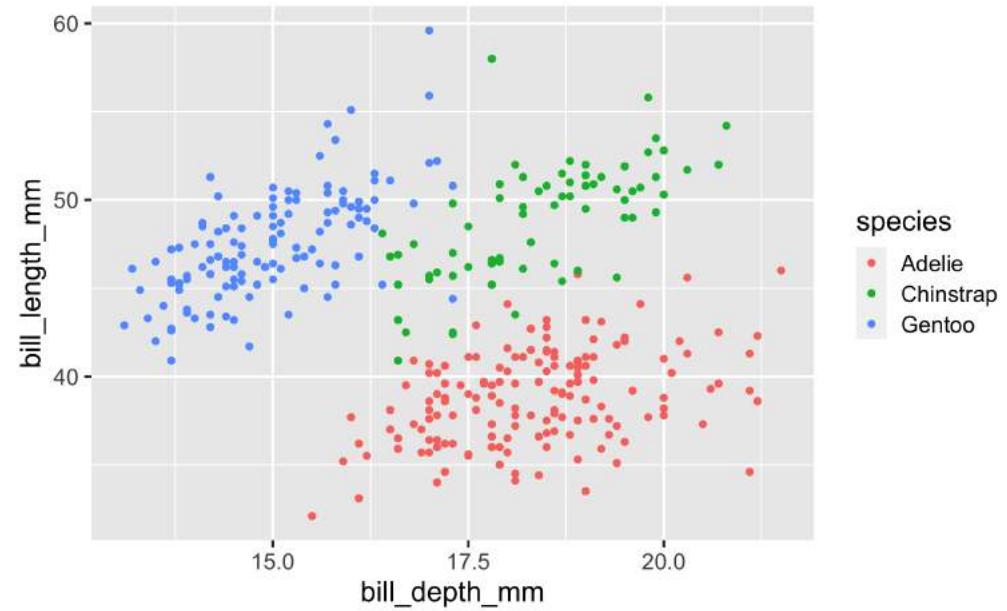
Começamos com o data frame `penguins`, mapeamos a largura do bico ao eixo dos x e o comprimento ao eixo dos y. **Representamos cada observação por um ponto**

```
ggplot(data = penguins,  
       mapping = aes(x = bill_depth_mm,  
                      y = bill_length_mm))  
  geom_point()
```



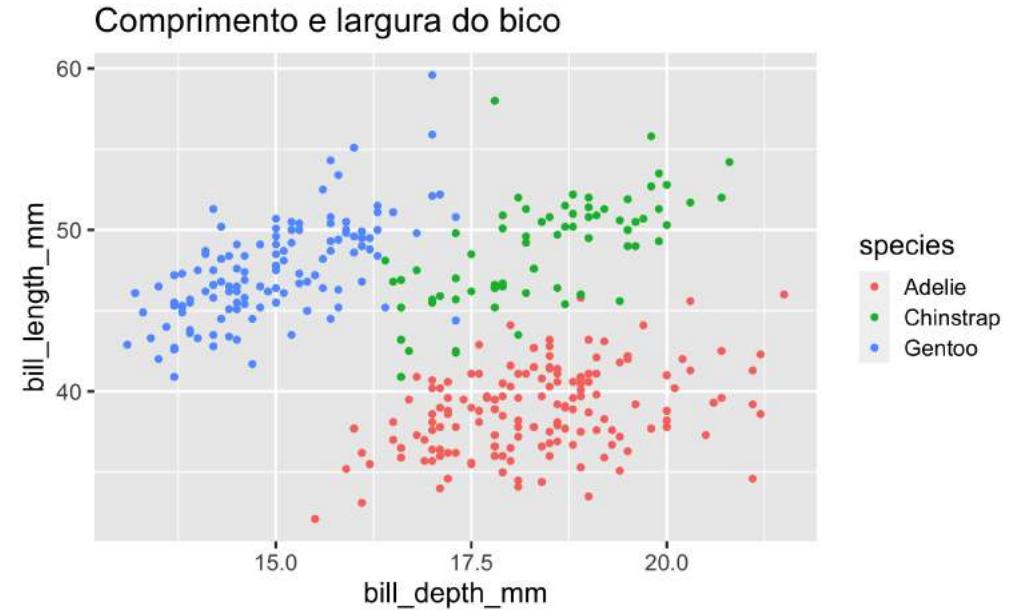
Começamos com o data frame `penguins`, mapeamos a largura do bico ao eixo dos x e o comprimento ao eixo dos y. Representamos cada observação por um ponto **e mapeamos a espécie à cor dos pontos.**

```
ggplot(data = penguins,  
       mapping = aes(x = bill_depth_mm,  
                      y = bill_length_mm,  
                      colour = species)) +  
  geom_point()
```



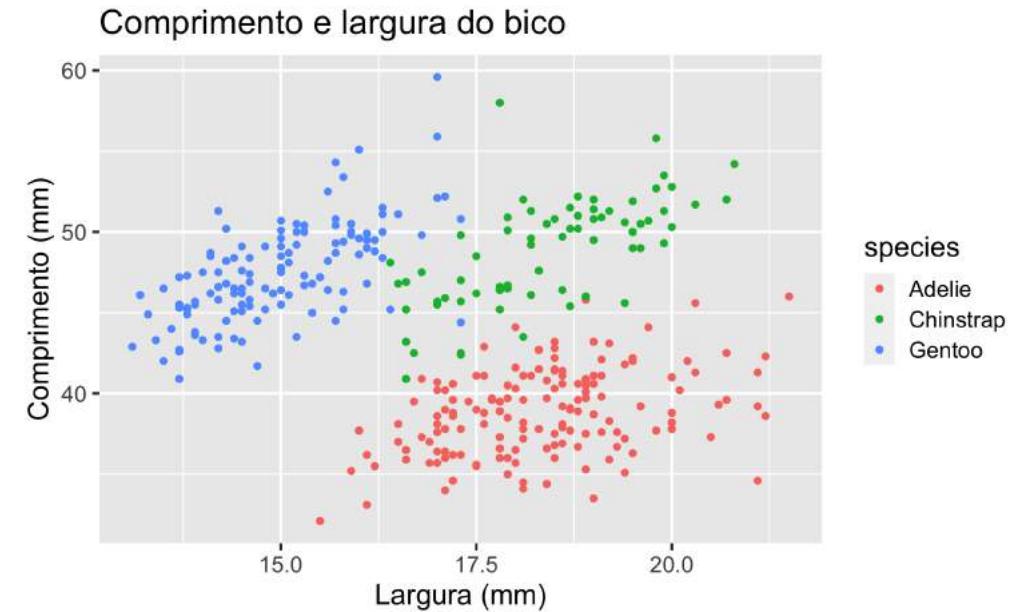
Começamos com o data frame `penguins`, mapeamos a largura do bico ao eixo dos x e o comprimento ao eixo dos y. Representamos cada observação por um ponto e mapeamos a espécie à cor dos pontos. **Damos um título ao gráfico,**

```
ggplot(data = penguins,  
       mapping = aes(x = bill_depth_mm,  
                      y = bill_length_mm,  
                      colour = species)) +  
  geom_point() +  
  labs(title = "Comprimento e largura do bico")
```



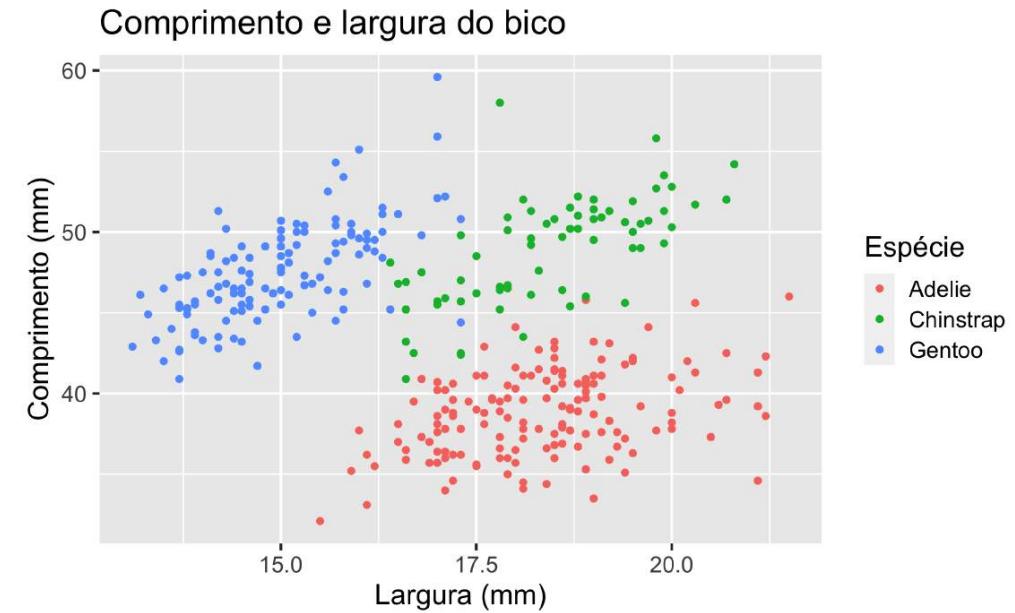
Começamos com o data frame `penguins`, mapeamos a largura do bico ao eixo dos x e o comprimento ao eixo dos y. Representamos cada observação por um ponto e mapeamos a espécie à cor dos pontos. Damos um título ao gráfico, **alteramos as legendas dos eixos**

```
ggplot(data = penguins,  
       mapping = aes(x = bill_depth_mm,  
                      y = bill_length_mm,  
                      colour = species)) +  
  geom_point() +  
  labs(title = "Comprimento e largura do bico",  
       x = "Largura (mm)", y = "Comprimento (mm)")
```



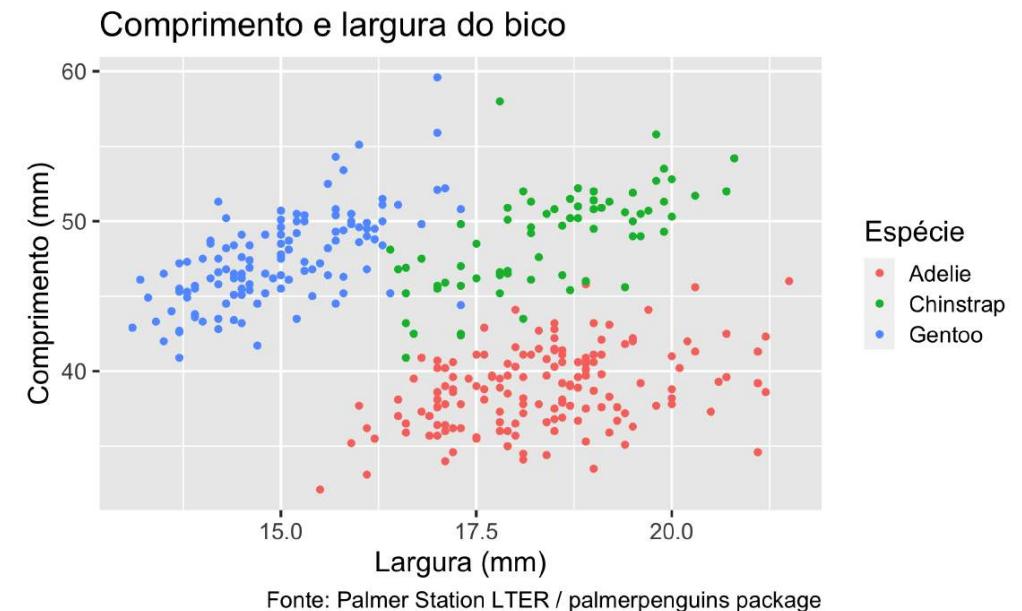
Começamos com o data frame `penguins`, mapeamos a largura do bico ao eixo dos x e o comprimento ao eixo dos y. Representamos cada observação por um ponto e mapeamos a espécie à cor dos pontos. Damos um título ao gráfico, alteramos as legendas dos eixos **e o título da legenda das cores**.

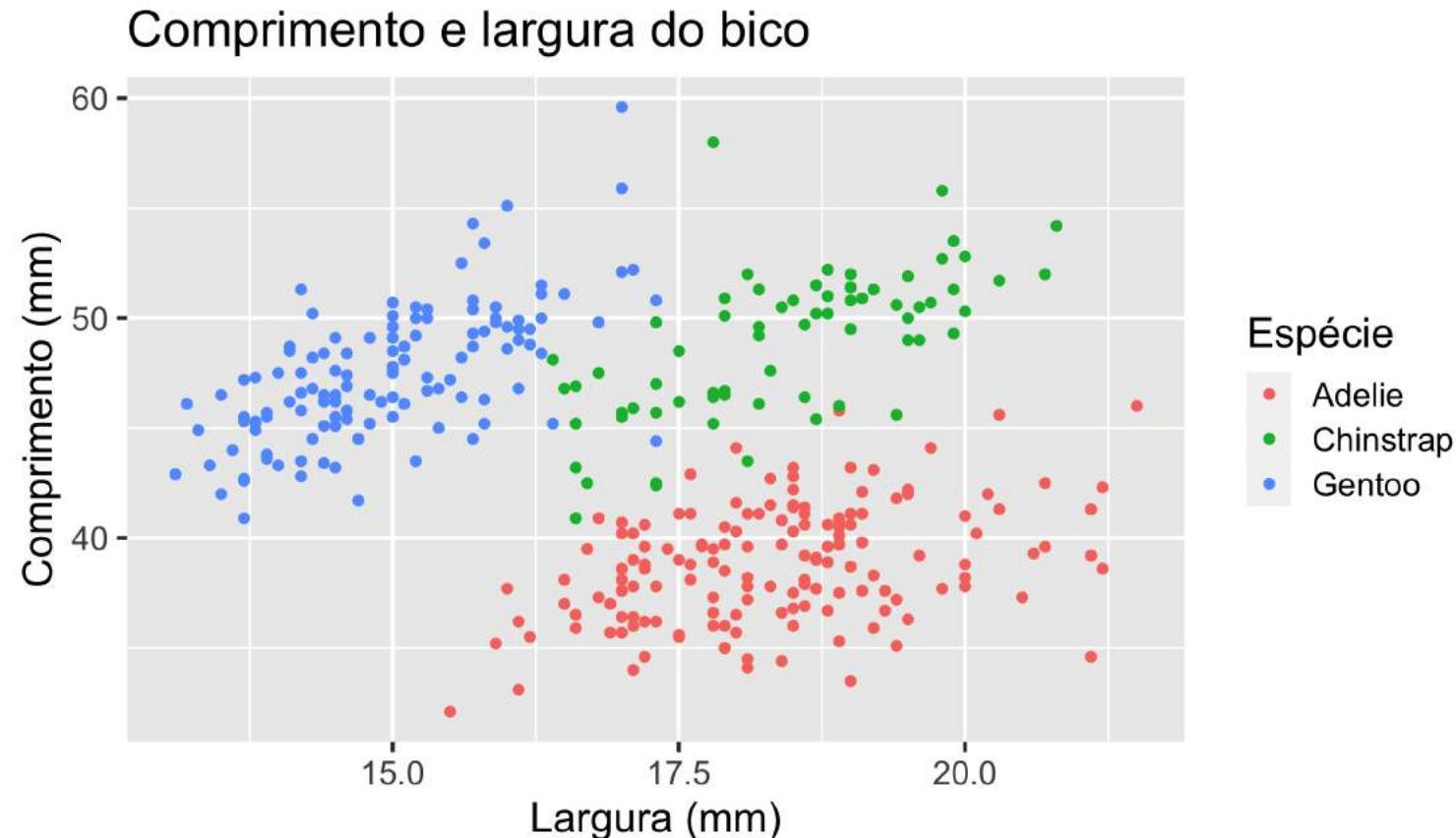
```
ggplot(data = penguins,  
       mapping = aes(x = bill_depth_mm,  
                      y = bill_length_mm,  
                      colour = species)) +  
  geom_point() +  
  labs(title = "Comprimento e largura do bico",  
       x = "Largura (mm)", y = "Comprimento (mm)",  
       colour = "Espécie")
```



Começamos com o data frame `penguins` mapeamos a largura do bico ao eixo dos x e o comprimento ao eixo dos y. Representamos cada observação por um ponto e mapeamos a espécie à cor dos pontos. Damos um título ao gráfico, alteramos as legendas dos eixos e o título da legenda das cores. **Acrescentamos uma nota com a fonte dos dados.**

```
ggplot(data = penguins,  
       mapping = aes(x = bill_depth_mm,  
                      y = bill_length_mm,  
                      colour = species)) +  
  geom_point() +  
  labs(title = "Comprimento e largura do bico",  
       x = "Largura (mm)", y = "Comprimento (mm)",  
       colour = "Espécie",  
       caption = "Fonte: Palmer Station LTER")
```





Fonte: Palmer Station LTER / palmerpenguins package



# Dica: podemos omitir os nomes dos primeiros argumentos em ggplot

```
ggplot(data = penguins,  
       mapping = aes(x = bill_depth_mm,  
                      y = bill_length_mm,  
                      colour = species)) +  
  geom_point()
```

```
ggplot(penguins,  
       aes(x = bill_depth_mm,  
                      y = bill_length_mm,  
                      colour = species)) +  
  geom_point()
```



# **Fontes**

# r4ds.had.co.nz

The screenshot shows a web browser window with the URL [r4ds.had.co.nz](http://r4ds.had.co.nz). The page title is "R for Data Science". The left sidebar contains a table of contents with chapters numbered 1 through 16. The main content area starts with a "Welcome" section, followed by the book's title "R for Data Science" and authors "Garrett Grolemund" and "Hadley Wickham". Below this is a detailed description of the book's purpose and content. A small image of the book cover is displayed at the bottom right.

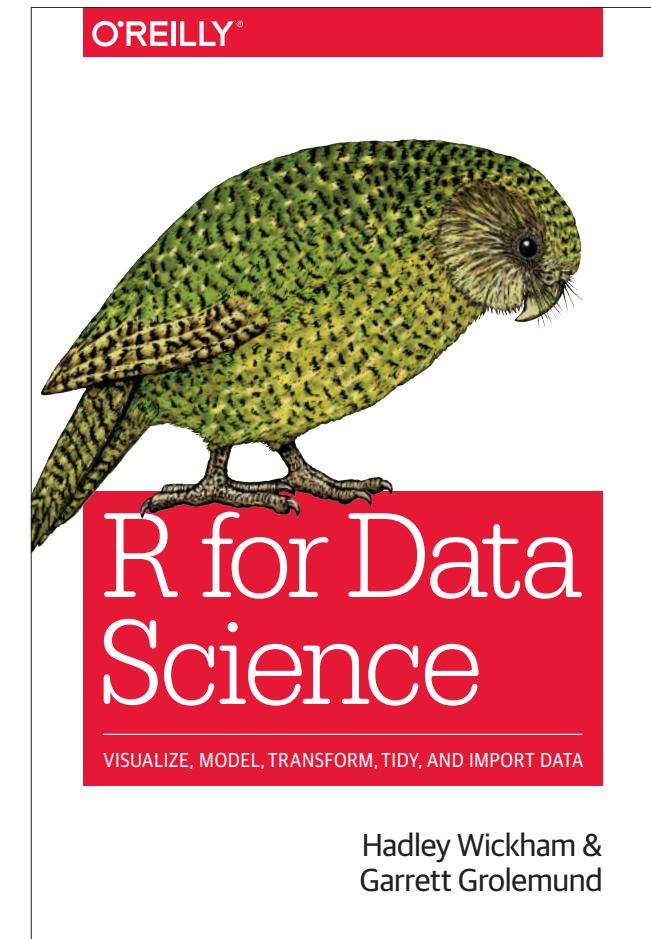
OREILLY

R for Data Science

VISUALIZE, MODEL, TRANSFORM, TIDY, AND IMPORT DATA

Hadley Wickham & Garrett Grolemund

ggplot2



<https://github.com/rstudio-education/remaster-the-tidyverse>

garrettgman Update README.md 7b4e581 on Nov 4, 2019 23 commits

- Data-Wrangling-With-Th... Minor fixes to Data Wrangling with the Tidyverse b... 4 years ago
- Welcome-To-The-Tidyve... Edits after Kansas workshop. Speeds up intro, mut... 4 years ago
- .gitignore Adds edits to Welcome the Tidyverse from BYU SIAS 4 years ago
- README.md Update README.md 4 years ago
- remaster-the-tidyverse.... First draft of abstract 4 years ago

☰ README.md

## Remaster the Tidyverse

This repository contains editable class materials built by Garrett Grolemund for two separate one day workshops:

- **Welcome to the Tidyverse**

A gentle introduction to R and its Tidyverse that focuses on learning to do Exploratory Data Analysis with the `ggplot2`, `dplyr`, `broom`, `modelr`, and `rmarkdown` packages. The course focuses on doing data science, not writing code; but by the



<https://datasciencebox.org/>



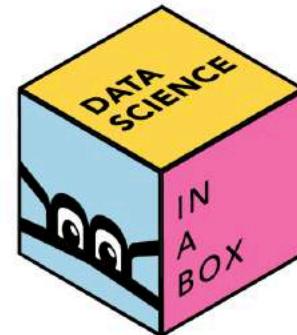
The screenshot shows the top navigation bar of the website. It includes a logo icon (a small yellow cube), followed by the menu items: Overview, Hello #dsbox!, Content, Infrastructure, and Design. To the right of the menu are four small icons: a person, a cloud, a gear, and a search magnifying glass.

## Welcome

How can we effectively and efficiently teach data science to students with little to no background in computing and statistical thinking? How can we equip them with the skills and tools for reasoning with various types of data and leave them wanting to learn more? This introductory data science course is our (working) answer to this question.

The source code for everything you see here can be found [on GitHub](#).

The core content of the course focuses on data acquisition and wrangling, exploratory data analysis, data visualization, inference, modelling, and effective communication of results. Time permitting, the course also introduces additional concepts and tools like interactive visualization and reporting, text analysis, and Bayesian inference. A heavy emphasis is placed on a consistent syntax (with tools from the [tidyverse](#)), reproducibility



On this page

[License](#)  
[Acknowledgements](#)

[Edit this page](#)  
[Report an issue](#)



**Obrigado e até amanhã!**

[luis.morais@novasbe.pt](mailto:luis.morais@novasbe.pt)