Introdução à Linguagem de Programação em R para tratamento de dados de poluição do ar

Mario Gavidia-Calderón, Rafaela Squizzato, Thiago Nogueira

January 23, 2024

Universidade de São Paulo

Introdução

Por que o curso?

(Inserir Thiago comments)

Por que R?

- R é uma **Linguagem de programação** para a análise de dados.
  - Um sistema para **estatistica**.
  - Um sistema de computação gráfica e **estatistica**.
  - Um ambiente para a análise de dadose e **estatistica**.

Por que R?

- É *open source* (é **libre**).
- Funciona em qualquer **sistema operacional**.
- Podemos trabalhar **com muitos dados** e **tipos**.
- Grande comunidade de usuários: **Muita ajuda on-line**.
- **Reproducibilidade** das ciências.

Por que R?

- Muitos **pacotes** para muitas áreas das ciências.
  - `openair` → poluição do ar.
  - `sf` e `raster` → GIS.
  - `Rmarkdown` → Documentos e apresentações.
  - `shiny` → applicações.
  - etc, etc, etc

RStudio

RStudio

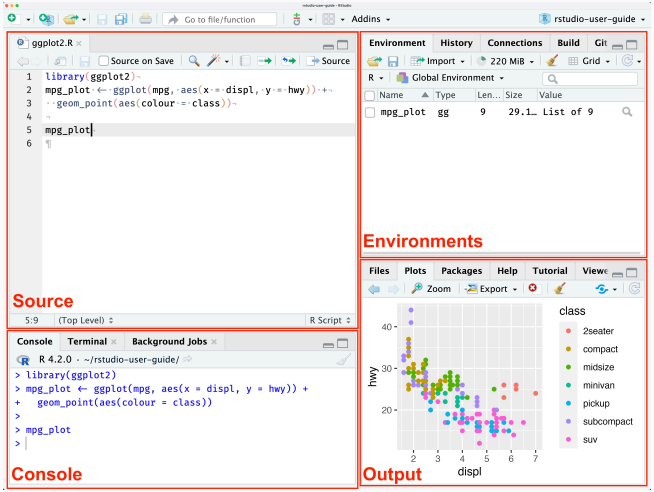


Figure 1: Distribuição das janelas do RStuido. Fonte: RStudio User Guide

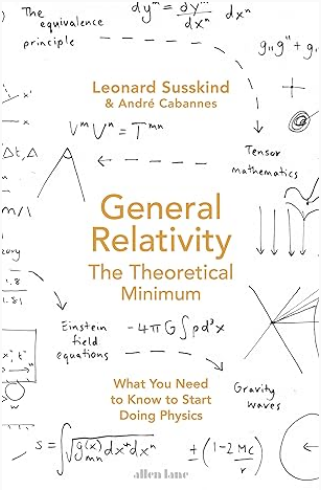
RStudio

- É importante aprender os *keyboard shortcuts*.
  - `Ctrl + 1` : Janela scripts.
  - `Ctrl + 2` : Janela consola.
  - `Alt + -` : `<-`
  - `TAB`: Autocompleta nome de funções e direções de arquivos.

Sintaxe Básica

The theoretical minimum

- What you need to know to start doing R



R como calculadora

- R pode ser usado como calculadora. Segue a ordem das operações

```
(5 + 10 * 2 / 4) ^ 2 - 5
```

```
## [1] 95
```

Declarar variavéis

- No R usamos '<-' em vez de = para definir variavéis.

```
R <- 8.314
R
```

```
## [1] 8.314
```

Comentar

- Para comentar **#**

```
R <- 8.314 # Constante universal dos gases (J K / mol)
R
```

```
## [1] 8.314
```

- Para usar funções: **nome_da_função()**

```
class(R)
```

```
## [1] "numeric"
```

R: Objetos

- **character**

```
uma_palavra <- "palavra"
class(uma_palavra)
```

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

- **numeric**

```
this_year <- 2022
g <- 9.81 # m/s2
class(this_year)
```

```
## [1] "numeric"
```

R: Objetos

- booleans

```
verdade <- TRUE
verdade
```

```
## [1] TRUE
```

```
falso <- 5 > 10
falso
```

```
## [1] FALSE
```

```
muito_falso <- "cinco" == "5"
muito_falso
```

R: Objetos - Vetores

- É definido usando a função **c()**

```
pontos_cardeais <- c("N", "E", "S", "W")
pontos_cardeais
```

```
## [1] "N" "E" "S" "W"
```

```
pontos_cardeais_graus <- c(0, 90, 180, 270)
class(pontos_cardeais_graus)
```

```
## [1] "numeric"
```

R: Objetos - Vetores

- Uma sequência é definida **seq(inicio, final, intervalo)**

```
de_1ate5 <- seq(1, 5)
de_1ate5
```

```
## [1] 1 2 3 4 5
```

```
pares_ate10 <- seq(0, 10, 2)
pares_ate10
```

```
## [1]  0  2  4  6  8 10
```

```
sec_float <- seq(0, 1, 0.2)
sec_float
```

R:Objetos - vetores - Seleção de elementos

- Para selecionar elementos do vetor: **nombre_vetor[posição]**:

```
# Primeiro elemento
pontos_cardeais_graus[1]
```

```
## [1] 0
```

```
# Último elemento
pontos_cardeais_graus[4]
```

```
## [1] 270
```

R:Objetos - vetores - Seleção de elementos

- Podemos selecionar varios elementos usando **outro vetor**

```
# Segundo y tercero
pontos_cardeais[c(2, 3)]
```

```
## [1] "E" "S"
```

- Podemos eliminar elementos usando **nome_vetor[-posição]**

```
GEE <- c("H2O", "CO2", "O2", "CH4")
GEE
```

```
## [1] "H2O" "CO2" "O2"  "CH4"
```

```
# Oxigênio não é GEE
```

R:Objetos - vetores - Substituição

- Podemos Substituir um elemento do vetor assim:

```
# Reemplazamos Oxígeno por Ozone
GEE[3] <- "O3"
GEE
```

```
## [1] "H2O" "CO2" "O3"  "CH4"
```

R: Objetos - `data frames`

- Um `data frame` é uma **tabela**
- Uma matriz **indexada**: tem nomes das **colunas** e **linhas**.
- Cada **coluna** é uma **variable**.
- Cada **linha** é uma **observação**.
- Um conjunto de vetores.

R: Objetos - `data frame`

- Criamos um **`data frame`** usando a função **`data.frame()`**

```
gases <- c("N2", "O2", "Ar", "CO2")
massa_molar <- c(28, 32, 40, 12 + 2 * 16)
percentagem <- c(78.08, 20.95, 0.9, 0.04)

ar <- data.frame(gas = gases,
                 W = massa_molar,
                 per = percentagem)
ar
```

```
##   gas  W   per
## 1  N2 28 78.08
## 2  O2 32 20.95
```

R: Objetos - `data frame`

- Criamos um **`data frame`** usando a função **`data.frame()`**

```
# Ou diretamente
ar <- data.frame(gas = c("N2", "O2", "Ar", "CO2"),
                 W = c(28, 32, 40, 12 + 2 * 16),
                 per = c(78.08, 20.95, 0.9, 0.04))
ar
```

```
##   gas  W   per
## 1  N2 28 78.08
## 2  O2 32 20.95
## 3  Ar 40  0.90
## 4 CO2 44  0.04
```

R: data frame - $ (dolar sign)

- Selecionamos uma **coluna** de un **data frame** como um **vetor**
- Sintaxis: `df$nome_coluna`
- E.g. Nome dos componentes do ar

```
ar$gas
```

```
## [1] "N2"  "O2"  "Ar"  "CO2"
```

```
class(ar$gas)
```

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

R: data frame - [ ] (colchetes?)

- Selecionamos uma **coluna** de un **`data frame`** como um **`data frame`**
- Sintaxis: `df[interiro]` ou `df[nome_coluna]`
- E.g. Nome dos componentes do ar

```
ar[1] # ou ar[”gas”]
```

```
##   gas
## 1  N2
## 2  O2
## 3  Ar
## 4 CO2
```

```
class(ar[1])
```

R: data frame - $ (signo de dolar)

- Algumas funcções precisam `vetores` como input
- e.g. média massa molar

```
mean(ar["W"])
```

```
## Warning in mean.default(ar["W"]): argument is not numeric or logical
## NA
```

```
## [1] NA
```

```
mean(ar$W)
```

```
## [1] 36
```

R: data frame - Criando novas colunas

- Usamos $: **df$nova_coluna <-**
- Nome completo dos gases:

```
ar$name <- c("Nitrogênio",
             "Oxigênio",
             "Argônio",
             "Diôxido de Carbono")
ar
```

```
##   gas  W   per               name
## 1  N2 28 78.08         Nitrogênio
## 2  O2 32 20.95           Oxigênio
## 3  Ar 40  0.90            Argônio
## 4 CO2 44  0.04 Diôxido de Carbono
```

R: data frame - Algumas funções

- Número de linhas: **`nrow()`**
- Número de colunas: **`ncol()`**

```
nrow(ar)
```

```
## [1] 4
```

```
ncol(ar)
```

```
## [1] 4
```

R: data frame - Algumas funções

- Tipo de objeto de cada coluna: **str()**

```
str(ar)
```

```
## 'data.frame':    4 obs. of  4 variables:
##  $ gas : chr  "N2" "O2" "Ar" "CO2"
##  $ W   : num  28 32 40 44
##  $ per : num  78.08 20.95 0.9 0.04
##  $ name: chr  "Nitrogênio" "Oxigênio" "Argônio" "Diôxido de Carbono"
```

- nome das colunas

```
names(ar)
```

```
## [1] "gas"  "W"    "per"  "name"
```

R: data.frame - Algunas funciones

- Primeiras observações: **head()**
- ùltimas observações: **tail()**

```
head(ar)
```

```
##   gas  W   per               name
## 1  N2 28 78.08         Nitrogênio
## 2  O2 32 20.95           Oxigênio
## 3  Ar 40  0.90            Argônio
## 4 CO2 44  0.04 Diôxido de Carbono
```

```
tail(ar)
```

```
##   gas  W   per               name
```

R: data.frame - Algunas funciones

- Primeiras observações: **head()**
- ùltimas observações: **tail()**

```
head(ar, 2)
```

```
##   gas  W   per       name
## 1  N2 28 78.08 Nitrogênio
## 2  O2 32 20.95   Oxigênio
```

```
tail(ar, 2)
```

```
##   gas  W  per               name
## 3  Ar 40 0.90            Argônio
## 4 CO2 44 0.04 Diôxido de Carbono
```

R: data.frame - Substituição

```r
# Em espanhol
ar$nombres <- c("Nitrógeno",
                "Oxígeno",
                "Argón",
                "Dióxido de carbono")
ar
```

```
##   gas  W   per               name            nombres
## 1  N2 28 78.08         Nitrogênio          Nitrógeno
## 2  O2 32 20.95           Oxigênio            Oxígeno
## 3  Ar 40  0.90            Argônio              Argón
## 4 CO2 44  0.04 Diôxido de Carbono Dióxido de carbono
```

R: Operaciones *Element-wise*

```r
tempC <- c(27, 32, 28, 26)
tempK <- tempC + 273.15
tempK
```

```
## [1] 300.15 305.15 301.15 299.15
```

```r
tempk_chr <- as.character(tempK)
str(tempk_chr)
```

```
##  chr [1:4] "300.15" "305.15" "301.15" "299.15"
```

Bulleted Lists

- Element A
- Element B
  - B.1
  - B.2
- Element C

Elements

Typography

```
The theme provides sensible defaults to
\emph{emphasize} text, \alert{accent} parts
or show \textbf{bold} results.

In Markdown, you can also use _emphasize_ and **bold**.
```

becomes

The theme provides sensible defaults to *emphasize* text, accent parts or show **bold** results.

In Markdown, you can also use *emphasize* and **bold**.

Math

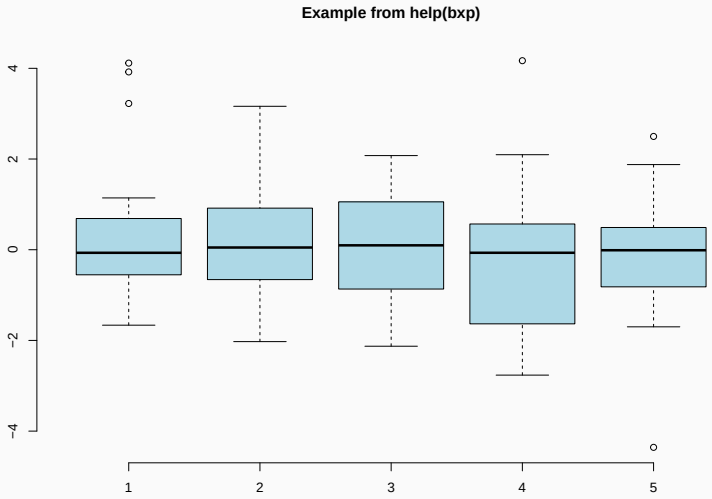
$$e = \lim_{n \to \infty} \left(1 + \frac{1}{n}\right)^n$$

R Figure Example

The following code generates the plot on the next slide (taken from `help(bxp)` and modified slightly):

```
library(stats)
set.seed(753)
bx.p <- boxplot(split(rt(100, 4),
                      gl(5, 20)), plot=FALSE)
bxp(bx.p, notch = FALSE, boxfill = "lightblue",
    frame = FALSE, outl = TRUE,
    main = "Example from help(bxp)")
```

R Figure Example



R Table Example

A simple `knitr::kable` example:

```
knitr::kable(mtcars[1:5, 1:8],
             caption="(Parts of) the mtcars dataset")
```

Table 1: (Parts of) the mtcars dataset

	mpg	cyl	disp	hp	drat	wt	qsec	vs
Mazda RX4	21.0	6	160	110	3.90	2.620	16.46	0
Mazda RX4 Wag	21.0	6	160	110	3.90	2.875	17.02	0
Datsun 710	22.8	4	108	93	3.85	2.320	18.61	1
Hornet 4 Drive	21.4	6	258	110	3.08	3.215	19.44	1
Hornet Sportabout	18.7	8	360	175	3.15	3.440	17.02	0

Resources

For more information:

- See the Metropolis repository for more on Metropolis
- See the RMarkdown repository for more on RMarkdown
- See the binb repository for more on binb
- See the binb vignettes for more examples.