

Métodos Computacionais em R

Claudio Neto, Dr.

Universidade Federal do Recôncavo da Bahia
Cruz das Almas, BA, BRA

Novembro, 16, 2021

tipos de dados em R

- R entende dados **numéricos** (*numeric*, e *integer* para números inteiros), de **texto** (*character*), valores lógicos (**TRUE/FALSE**), entre outros.

```
populacao <- c(1500, 2000)
estados <- c("Acre", "Alagoas", "Amapá", "Amazonas", "Tocantins")
class(populacao)
```

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

```
class(estados)
```

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

tipos de dados em R

```
verdadeiro_falso <- c(FALSE, TRUE)
verdadeiro_falso
```

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

```
class(verdadeiro_falso)
```

```
## [1] "logical"
```

```
posicao <- c(1L, 2L)
posicao
```

```
## [1] 1 2
```

```
class(posicao)
```

vetores

- `c()` para criar - "concatenar"

```
str(estados) #estrutura
```

```
## chr [1:5] "Acre" "Alagoas" "Amapá" ...
```

```
length(estados)
```

```
## [1] 5
```

selecionando elementos de vetores (*subsetting*)

- entre **colchetes** `[]`
- com vetores de **posição** ou vetores **lógicos** (`TRUE/FALSE`)

```
estados <- c("Acre", "Alagoas", "Amapá", "Amazonas", "Tocantins")
estados[1]
estados[c(TRUE, FALSE, TRUE, FALSE)]
estados[1:2]
estados[1:4]
#estados[ 1,3,5] #NAO! a vírgula marca dimensões
estados[c(1,3,5)] #tem que ser um vetor
```

subsetting com cláusulas lógicas

```
casos <- c(150, 200, 400, 500, 500, 600)
casos > 150
casos[casos > 150]
casos >= 150 # superior ou igual
casos < 300
casos == 200
casos != 200
```

operadores lógicos

- igualdade: `==`
- diferença: `!=` (! em geral é negação)
- desigualdades: `<`, `>`, `<=`, `>=`
- união (OR/OU): `|`
- interseção (AND): `&`
- pertencimento: `%in%`
- não pertencimento: `!a %in% b`
- `casos == 200`
- `casos != 200`
- `casos < 300, casos >= 150`
- `casos < 200 | casos > 500`
- `casos > 200 & casos < 500`
- `casos %in% c(100, 200, 300)`
- `!casos %in% c(400)`

O operador lógico cria um **vetor lógico**, a seleção vai entre colchetes:

```
casos[casos > 150]
```

```
## [1] 200 400 500 500 600
```


criando vetores numéricos

```
1:10  
seq(1, 10, 2)  
rep(1:4, 2)  
rep(1:4, each = 2)  
unique(casos)
```

estruturas de dados em R

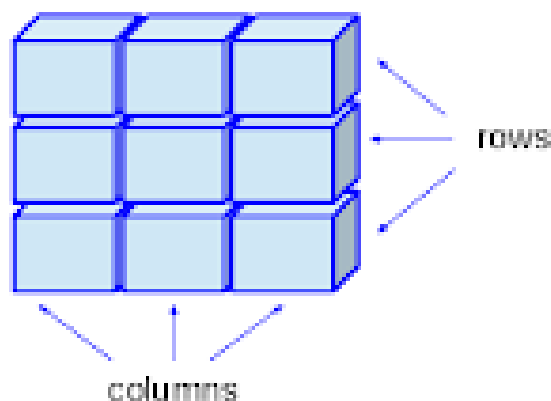
- **vetor**: lineal, uma dimensão só: `length()`
- **fatores**: vetores que representam **variáveis categóricas** e portanto têm níveis (`levels()`)
`factor(estados)`
- **matrizes**: arranjos de duas dimensões de **dados do mesmo tipo** (`dim()`, `nrow()`, `ncol()`).
- **listas**: literalmente listas de qualquer outro objeto (inclusive listas)
- **data frames**: arranjos bi-dimensionais de dados de diferentes tipos (i.e., uma coluna numérica, outra com nomes, outra com um fator etc.)

estruturas de dados em R

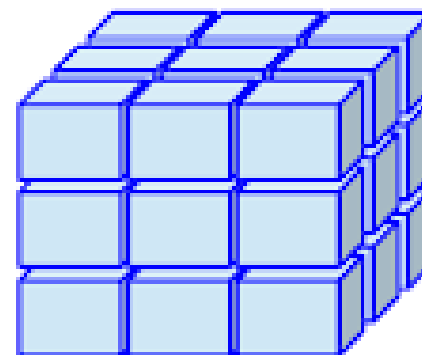
Vector



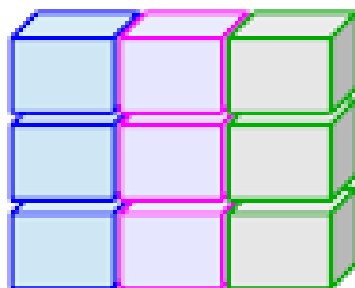
Matrix



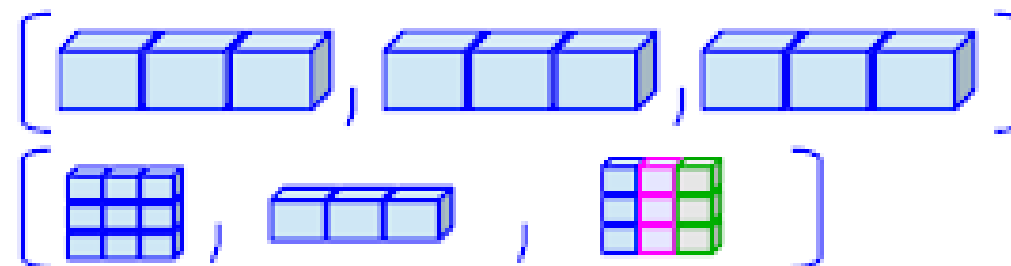
Array



Data Frame
(Table)



Lists



instalando pacotes

```
# Para instalar pacotes desde CRAN  
install.packages("remotes")
```

```
# Para instalar coronabr desde GitHub  
remotes::install_github("cccneto/Ibamam")
```

```
# Para carregar pacotes  
library(name_pckg)
```

```
# Para buscar ajuda  
?name_pckg
```

funções, argumentos e entendendo a ajuda

```
help(funcao)  
?funcao  
??palavra_chave  
args(funcao)
```

```
help("get_corona_br")  
?get_corona_br  
??corona  
args(get_corona_br)
```

- ou selecionar o nome da função e clicar **F1**
- No **help** estão os argumentos na ordem correta, e os valores padrão (por *default*) estão indicados. Se a gente não especificasse a função usaria esses valores.

download dos dados para Amapá

```
#library(coronabr)
```

```
caminho <- "dados/brutos"  
if (!dir.exists(caminho)) {  
  dir.create(caminho)  
}  
get_corona_br(dir = caminho,  
              filename = "01-amapa",  
              uf = "AP")
```

```
amapa <- read.csv("dados/brutos/01-amapa.csv",  
                 stringsAsFactors = FALSE,  
                 encoding = "UTF-8")  
class(amapa)
```

inspecionar objetos data.frame

```
names(amapa)
dim(amapa)
nrow(amapa)
ncol(amapa)
head(amapa) # 6 linhas por padrão
tail(amapa)
rownames(amapa)
length(amapa) # numero de colunas
summary(amapa) # quantis e a média
```

selecionar colunas e filtrar linhas

- entre colchetes também mas **as dimensões separadas por uma vírgula**

`amapa[linhas, colunas]`

```
amapa[1:3, 1:3] # 3 primeiras linhas, e três primeiras colunas
```

```
##      city city_ibge_code      date
## 1 Macapá      1600303 2020-03-20
## 2   <NA>         16 2020-03-20
## 3 Macapá      1600303 2020-03-21
```

```
amapa[4, 13] #quarto elemento da coluna 13
```

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


fontes de ajuda

- Material de estudo em português [link](#)
- [LearnR4Free](#) em inglês e em espanhol
- Disciplina *Projetos de análise de dados usando R* [link](#)
- R Reference Card [PDF](#)
- Base R cheatsheet [PDF](#)
- R Taskviews: listas de pacotes por tema curados por especialistas [link](#)
- Livro [R for Data Science](#)
- Pacote **dados**: os conjuntos de dados de R4DS traduzidos para o português: [GitHub](#)
- RStudio cheatsheet [PDF](#)
- RStudio Data Import Cheatsheet [PDF](#)
- R Bloggers <https://www.r-bloggers.com/>

Onde pedir ajuda?

- Latin-R <http://bit.ly/latinr-slack>
- RLadies+ <https://rladies.org/>
- Slack da comunidade RLadies+ <https://rladies-community-slack.herokuapp.com/>
- YouTube de RLadies+ global <https://www.youtube.com/c/RLadiesGlobal>
- Comunidade de RStudio <https://community.rstudio.com/>
- Configuração das opções globais de RStudio [link](#)
- [Stack OverFlow](#)
- E os coleguinhas de turma!

Proximo titulo da seção

