

Programação Estatística

Introdução ao R-Iniciação

Rachid Muleia
rachidmuleia.com



Compiled: 2026-03-08

O que esperar deste módulo ??

- Compreender os princípios fundamentais da linguagem R
- Aplicar com rigor funções básicas de R
- Importar, estruturar e transformar conjuntos de dados em R
- Conceptualizar e implementar estruturas de controlo (condicionais e ciclos)
- Desenvolver funções definidas pelo utilizador
- Produzir gráficos adequados a diferentes tipologias de dados

O que é o R

R é uma linguagem e ambiente de programação criada para manipulação de dados, análises estatísticas e visualização de dados **dados em conhecimento**.

Software bastante poderoso

- **Livre e open source** — qualquer pessoa pode usar e contribuir.
- Feito para **estatística, ciência de dados e investigação**.
- Permite **análises rigorosas, visualizações impressionantes e reprodutibilidade científica**.
- Tem milhares de pacotes no **CRAN**, que expandem o que é possível fazer.
- Usado mundialmente em **epidemiologia, machine learning, bioestatística e ciências sociais**.

Estes slides foram criados usando R

Como obter o R ?

- O R pode ser encontrado na Internet através do link <https://cran.r-project.org>



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

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

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

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

Donations
[Donate](#)

The Comprehensive R Archive Network

Download and Install R

Precompiled binary distributions of the base system and contributed packages, **Windows and Mac** users most likely want one of these versions of R:

- [Download R for Linux \(Debian, Fedora/Redhat, Ubuntu\)](#)
- [Download R for macOS](#)
- [Download R for Windows](#)

R is part of many Linux distributions, you should check with your Linux package management system in addition to the link above.

Source Code for all Platforms

Windows and Mac users most likely want to download the precompiled binaries listed in the upper box, not the source code. The sources have to be compiled before you can use them. If you do not know what this means, you probably do not want to do it!

- The latest release (2025-10-31, [Not] Part in a Rumble) [R-4.5.2.tar.gz](#), read [what's new](#) in the latest version.
- The CRAN directory [src/base-prerelease](#) contains R alpha, beta, and rc releases as daily snapshots in time periods before a planned release.
- Between releases, the same directory [src/base-prerelease](#) contains snapshots of current patched and development versions. Please read about [new features and bug fixes](#) before filing corresponding feature requests or bug reports.
- Alternatively, daily snapshots are [available here](#).
- Source code of older versions of R is [available here](#).
- Contributed extension [packages](#).

Questions About R

- If you have questions about R like how to download and install the software, or what the license terms are, please read our [answers to frequently asked questions](#) before you send an email.

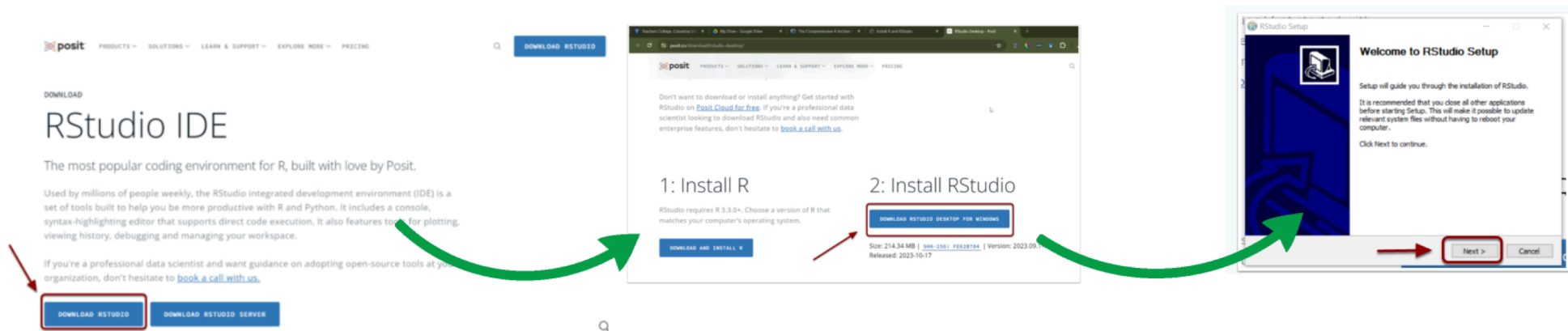
Supporting CRAN

- CRAN operations, most importantly hosting, checking, distributing, and archiving of R add-on packages for various platforms, crucially rely on technical, emotional, and financial support by the R community.

Please consider making [financial contributions](#) to the R Foundation for Statistical Computing.

What are R and CRAN?

- Embora o R pode funcionar de forma independente é sempre bom usar IDE, um ambiente de desenvolvimento
- O IDE integra um editor de código e outras várias ferramentas que facilitam o trabalho do programador
- Existem vários IDE, mas a comunidade de usuários do R, na sua maioria, usa o Rstudio
- O Rstudio pode ser descarregado a partir do link <https://posit.co/downloads/>



R vs Rstudio

R: Motor



Rstudio: Cockpit



O Rstudio nunca irá funcionar se não tiver o R instalado

Interface do Rstudio

The image shows the RStudio interface with four main panels highlighted by colored borders and annotated with text in Portuguese:

- Editor (Red border):** The top-left panel where R scripts are written. It contains the text: "Editor: Neste ambiente escrevemos o nosso script. Todo o código escrito aqui pode ser executado".
- Environment (Purple border):** The top-right panel showing the current environment. It contains the text: "Aqui irá aparecer todas as variáveis/objectos definidos no script.".
- Console (Green border):** The bottom-left panel showing the output of R commands. It contains the text: "Console: os resultados das execuções serão impresso aqui: Pode usar para testar as suas ideias".
- Packages (Orange border):** The bottom-right panel showing a list of installed and available packages. It contains the text: "Aqui poderá visualizar os graficos ver os pacotes instalados, etc".

The RStudio window title is "RStudio". The menu bar includes: File, Edit, Code, View, Plots, Session, Build, Debug, Profile, Tools, Help. The toolbar includes icons for saving, running, and other functions. The status bar at the bottom shows "1:1 (Top Level) R Script".

Name	Description	Version
☐ abind	Combine Multidimensional Arrays	1.4-5
☐ alphahull	Generalization of the Convex Hull of a Sample of Points in the Plane	2.2
☐ askpass	Safe Password Entry for R, Git, and SSH	1.1
☐ assertthat	Easy Pre and Post Assertions	0.2.1
☐ automap	Automatic interpolation package	1.0-14
☐ backports	Reimplementations of Functions Introduced Since R-3.0.0	1.1.5
☐ base64enc	Tools for base64 encoding	0.1-3
☐ bayestestR	Understand and Describe Bayesian Models and Posterior Distributions	0.4.0
☐ BH	Boost C++ Header Files	1.69.0-1
☐ BiocManager	Access the Bioconductor Project Package Repository	1.30.8
☐ bit	A Class for Vectors of 1-Bit Booleans	1.1-14

O que pode ser feito com o R

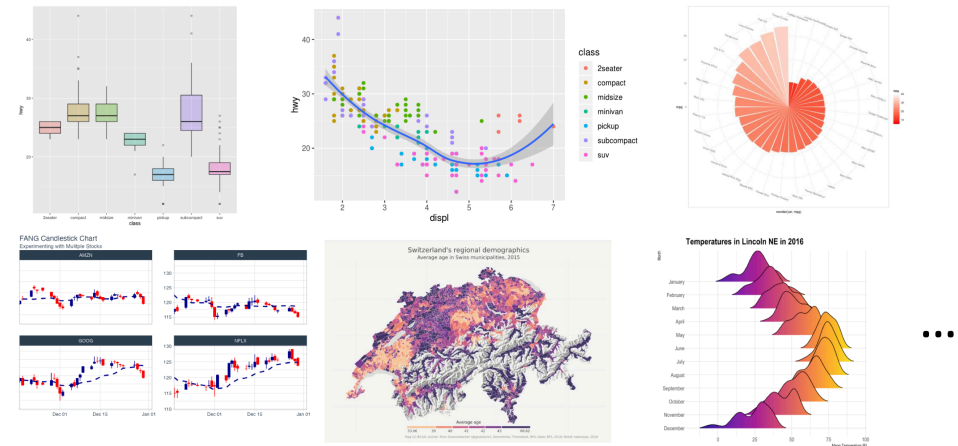
Análises estatísticas e econométricas

- Análise de regressão
- Análise multivariada de dados
- Análise de sobrevivência
- Inferência Bayesiana

Optimização e Análise matemática

- Álgebra linear, operações com matrizes
- Análise numérica
- Equações diferenciais

O R é extremamente poderoso para **visualização de dados**



Variáveis/objectos

- O R é uma linguagem orientada a objectos

O que é uma variável?

- Um “envelope” ou “balde” onde a informação pode ser armazenada e referenciada
- Uma abstracção que nos permite lidar com a estrutura subjacente e as relações
- Cada “balde” tem um nome por fora e contém informação no interior
 - **Nome da variável:** o nome escrito no exterior do balde
 - **Valor da variável:** a informação guardada dentro do balde
- O objecto é criado no momento da atribuição da variável: o processo de colocar informação dentro do balde
- A atribuição de uma variável pode ser feito usando `<-` ou `=`

```
1 x <- 3
```

Tipos de Variáveis

Note

Em qualquer linguagem de programação, existe o conceito essencial de **tipos de dados**. É o tipo do objecto que determina que operações podem ser realizadas com ele

Tipos mais comuns

- *integer (inteiro)*: ..., -1, 0, 1, 2, ...
- *double (números reais)*:
valores com casas decimais (por exemplo, 3.14, 2.0, -7.5)
- *character (texto/caracter)*:
"a", "R é fantástico", "the quick brown fox...", "1"
- *Boolean (lógico)*:
TRUE ou FALSE

```
1 y <- 2
2
3 z <- "a"
4
5 y+z
```

```
Error in `y + z`:
! non-numeric argument to binary operator
```

Criação de objectos

Nomes permitidos

- `meu.objecto`, `meu_objecto`, `meu0bjecto`, `a`, `b`, `x1`, `x2`, `data1`, `1data`

Nomes não permitidos

- `.` seguido de um número no início: `.4you`, mas `.you` é aceitável
- O nome de uma variável não pode iniciar com um número ou underscore (`_`)
- Não se pode usar palavras reservadas: `if`, `else`, `repeat`, `for`, `while`, `function`, `TRUE`, `FALSE`, etc.
- Mais descrições sobre palavras reservadas podem ser encontradas executando o código: `help("reserved")`

O R é *case sensitive*, isto é, distingue maiúsculas de minúsculas

R como calculadora

O R pode funcionar como uma calculadora científica

Cálculos aritméticos

```
1 1+2 # adição
2 4-1 # subtração
3 2*4 # multiplicação
4 8/4 # divisão
```

Mais funções

`log10()`, `log2()`, `tan()`, `asin()`, `acos()`, `atan()`,
`sinh()`, `cosh()`, `tanh()`,
`asinh()`, `acosh()`, `atanh()`,
`abs()`, `round()`, `floor()`, `ceiling()`,
`trunc()`, `signif()`, *entre outras...*

- Visitar a documentação do R. **A documentação é a chave para o R**

Algumas funções aritméticas

```
1 exp(1)
2 exp(log(5))
3 sin(pi/2)
4 cos(pi/2)
5 max(4,2,5,1)
6 min(4,2,5,1)
7 sum(4,2,5,1)
8 prod(4,2,5,1)
9 sqrt(16)
10 factorial(4)
11 choose(5,2)
```

Inciar uma sessão

- Primeiro inicializar o RStudio
- Abra um script novo (**File -> New File -> R script**) e grave com o nome que quiser (**File -> Save**)
- Especificar o directório de trabalho usando o RStudio no campo Session (**Session -> Set Working Director**) ou `setwd('D:/Documents/Folder')`. Use `getwd()` para verificar
- Os outputs serão gravados no seu directório de trabalho (caso não especifique outro directório)
- Pode usar a função `list.files()` para ver os ficheiros dentro to seu directório de trabalho

```
1 list.files()
```

[1] "01_md_rstudio.png"	"15.jpg.avif"	"arrow.jpg"
[4] "AULA1_cache"	"AULA1_files"	"AULA1_PE.html"
[7] "AULA1_PE.Rmd"	"AULA1.html"	"AULA1.qmd"
[10] "AULA1.rmarkdown"	"callouts.css"	"cfe-fonts.css"
[13] "cfe.css"	"Consola_Rstudio.png"	"custom.css"
[16] "custom.scss"	"grafico_exemplo.png"	"index.html"
[19] "InstallationR.png"	"libs"	"passo3.png"
[22] "Posit_help.png"	"R_logo.svg.png"	"R_versus_Rstudio.png"
[25] "R_vs_R_packages.png"	"R_vs_RStudio.png"	"rbin_fig.png"
[28] "right_join.PNG"	"Rstudio_consola.PNG"	"Rstudio_logo_copy.png"
[31] "Rstudio_logo.png"	"Satckoverflow_help.png"	"scared_image.jpg"
[34] "title-slide.html"		

Como buscar ajuda

- O R possui uma extensa lista de recursos para ajuda
- Dentro do R podemos usar a função `help()`

```
1 help("max")
```

- Alternativamente podemos usar `?`

```
1 ?max
```

- Podemos também procurar dentro da documentação tudo que tem a ver com um determinada palavra usando `help.search()` ou `??`

```
1 help.search("max")
2
3 #ou
4 ??max
```

Como buscar ajuda

É normal enfrentar obstáculos ao aprender R. Não desanime, faz parte do processo. Quando precisar, recorra à internet: há muitos recursos, exemplos e comunidades prontos para ajudar.

- Antes de entrar em pânico com um erro no R, procure-o no **Google**. É muito provável que outras pessoas já tenham enfrentado o mesmo problema e encontrado uma solução.
- Pode usar o **Stack Overflow**, que é um fórum onde muitos usuários colocam as suas dúvidas.

2. Stack Overflow » R



This section of Stack Overflow is for asking questions about R programming language.

stackoverflow.com/.. [+ Follow](#)

[f](#) 517.4K · [t](#) 230.6K · [d](#) 30 posts / day · [c](#) Sep 2008

Get Email Updates ⓘ

Continue

OR

[G](#) Continue with Google

3. Reddit » The Statistical Computing with R



[l](#) San Francisco, California, US

This subreddit is for discussing R stories, questions, and news from the Reddit community.

reddit.com/r/rstats [+ Follow](#)

[f](#) 1.7M · [t](#) 1.1M · [d](#) 5 posts / day · [c](#) Oct 2009

Dicas para aprendizagem do R

- Use o R sempre e de forma regular. O R é como uma língua, se não usas esqueces
- O R não é um teste de memória, tens o script para sempre lembra-se do que fez
- Existem várias forma de resolver um determinado problema. Procure ver como os outros fazem

Note

Sempre irá ter momentos de frustração. Aceite estes momentos, fazem parte do processo. Dedique mais tempo, em momento certo irá notar o progresso.

Estruturas de dados

Visão geral

Dimensão	Homogéneo	Heterogéneo
1-d	Vector	Lista
2-d	Matriz	Data frame
n-d	Array	

- As estruturas homogéneas levam o mesmo tipo de dados
- Estruturas heterogéneas podem levar qualquer tipo de dados

Vectores

- Estrutura unidimensional
- Os vectores levam um único tipo de dados

2	3	4	4
---	---	---	---

TRUE	TRUE	FALSE	TRUE
------	------	-------	------

“Maria”	“Alice”	“Marta”	“Pedro”
---------	---------	---------	---------

Como criar vectores no R?

- `c()`: função para combinar valores individuais
- `seq()`: função para criar uma sequência de valores
- `rep()`: função para criar réplica de valores

Vectores

Exemplo de criação de vectores usando a função `c()`

```
1 > c(1,2,3,4,5,6,7)
2 ## [1] 1 2 3 4 5 6 7
3 > c(2:5,11:6)
4 ## [1] 2 3 4 5 11 10 9 8 7 6
```

```
1 > x<-c(2,7,8,12,3,25)
2 > x
3 ## [1] 2 7 8 12 3 25
```

```
1 > y<-c(red="Bob",blue="Dave",green="Jenny")
2 > y
```

```
red    blue    green
"Bob"  "Dave"  "Jenny"
```

Vectores

Exemplo de criação de vectores usando a função `seq()`

```
1 > seq(from=1,to=8)
2 ## [1] 1 2 3 4 5 6 7 8
3 > seq(from=4,to=10,by=2)
4 ## [1] 4 6 8 10
5 > seq(from=1,to=10,length=4)
6 ## [1] 1 4 7 10
```

Vectores

Exemplo de criação de vectores usando a função `seq()`

```
1 > seq(from=1,to=8)
2 ## [1] 1 2 3 4 5 6 7 8
3 > seq(from=4,to=10,by=2)
4 ## [1] 4 6 8 10
5 > seq(from=1,to=10,length=4)
6 ## [1] 1 4 7 10
```

```
1 > seq(1,8)
2 ## [1] 1 2 3 4 5 6 7 8
3 > seq(4,10,2)
4 ## [1] 4 6 8 10
5 > seq(1,10,,4)
6 ## [1] 1 4 7 10
```

Vectores

Criação de vectores usando a função `rep()`

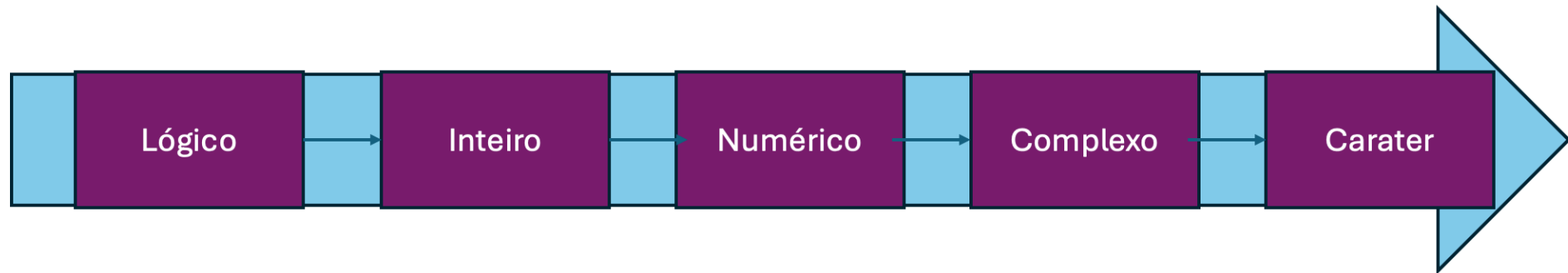
```
1 > rep(1,4)
2 ## [1] 1 1 1 1
3 > rep(4:5,3)
4 ## [1] 4 5 4 5 4 5
5 > rep(1:4,each=2)
6 ## [1] 1 1 2 2 3 3 4 4
7 > rep(1:4,times=2,each=2)
8 ## [1] 1 1 2 2 3 3 4 4 1 1 2 2 3 3 4 4
```

Vectores

Conversão do tipo de dados

- Vectores levam apenas um único tipo de dados
- Ao combinar diferentes tipos de dados, o R irá fazer a coerção para o tipo de dados mais flexível

[Regra para coerção](#)



Vectores

Coerção de dados- exemplo

```
1 > x<-c(5,'b') # será convertido para character
2 > x           # " " indica tipo character
3 ## [1] "5" "b"
4 > y<-c(FALSE, 3) # será convertido para número
5 > y
6 ## [1] 0 3
```

Vectores

Coerção de dados

A coerção de dados também pode ser feita usando as funções `as.class_name`.

- Exemplo

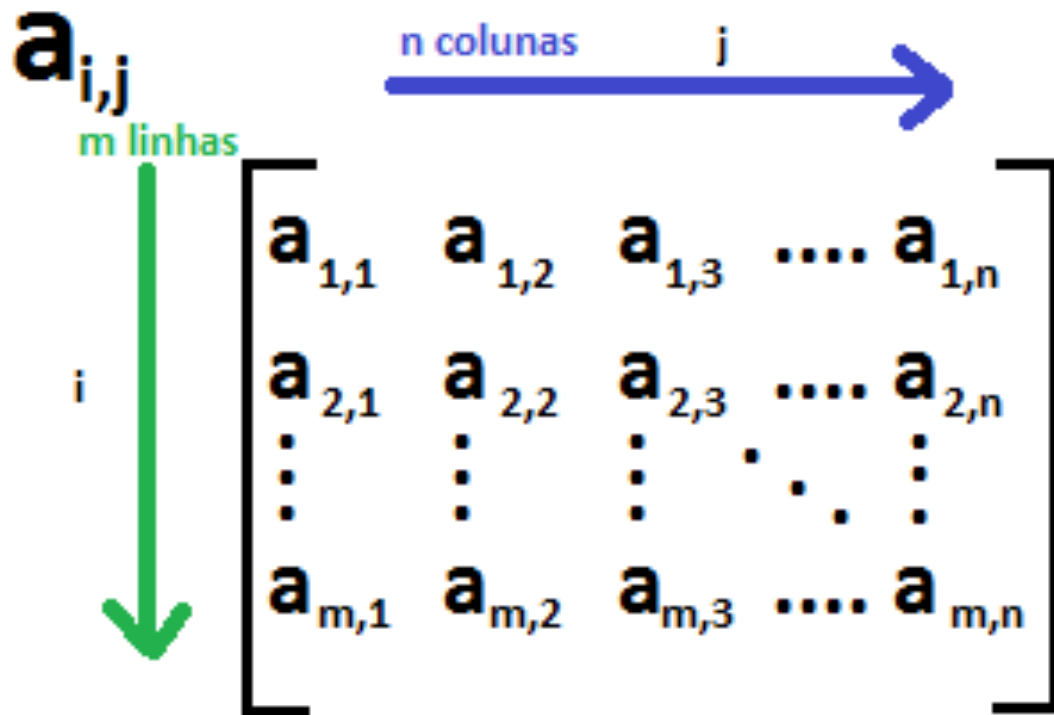
- `as.numeric(x)`
- `as.logical(x)`
- `as.character(x)`
- `as.integer(x)`
- `as.factor(x)`
- `as.complex(x)`

O tipo de dados pode ser verificado usando a função `typeof(x)` ou `class(x)`.

Matrizes

- Uma colecção de valores com mesmo tipo de dados
- Estrutura bidimensional disposta em linhas e colunas.

Matriz **m** por **n**



Matrizes

Uma matriz pode ser criada usando as seguintes funções:

- `matrix()` cria uma matrix especificando as linhas e as colunas
- `dim()` cria uma matrix definindo a dimensão do vector
- `cbind()` ou `rbind` cria uma matriz combinando colunas ou linhas, respectivamente

```
1 > matrix(data=1:6,nrow=2,ncol=3,byrow=FALSE)
2 > matrix(data=1:6, # dados para popular a matriz
3 +       nrow=2,   # numero de linhas
4 +       ncol=3,   # numero de colunas
5 +       byrow=FALSE) # preencher a matriz por linha
```

Exemplo de criação de matriz

```
1 > matrix(data=1:6,nrow=2,ncol=3,byrow=FALSE)
2 ##      [,1] [,2] [,3]
3 ## [1,]    1    3    5
4 ## [2,]    2    4    6
```

Matrizes

Criação de matriz usando a função `dim()`

```
1 > x<-1:6
2 > dim(x)<-c(2,3)
3 > x
4 ##      [,1] [,2] [,3]
5 ## [1,]    1    3    5
6 ## [2,]    2    4    6
7 > y<-1:6
8 > dim(y)<-c(3,2)
9 > y
10 ##      [,1] [,2]
11 ## [1,]    1    4
12 ## [2,]    2    5
13 ## [3,]    3    6
```

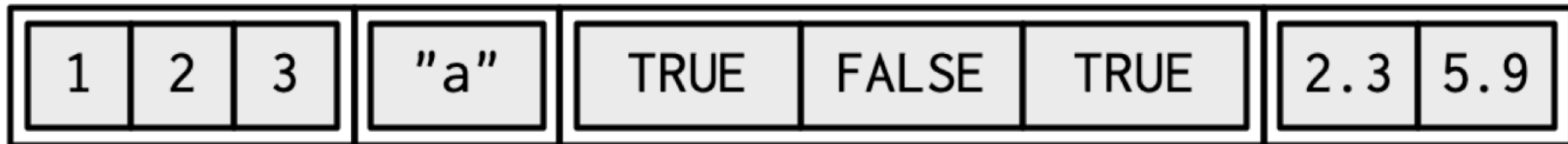
Matrizes

Criação de matriz usando as funções `cbind()` e `rbind()`

```
1 > x<-1:3
2 > y<-10:12
3 >
4 > cbind(x,y)
5 ##      x  y
6 ## [1,] 1 10
7 ## [2,] 2 11
8 ## [3,] 3 12
9 > rbind(x,y)
10 ##      [,1] [,2] [,3]
11 ## x      1   2   3
12 ## y     10  11  12
```

Listas

- Lista é uma coleção de estruturas de dados
- Uma lista pode armazenar qualquer tipo de dados, incluindo lista
- Uma lista pode ser criada usando a função `list()`
- Maior parte das funções em R produzem outputs armazenados numa lista



Lista

Exemplo de criação de lista

```
1 > minha_Lista<-list(1:3,c("a","b"),c(TRUE,FALSE,TRUE))
2 >
3 > str(minha_Lista)
4 ## List of 3
5 ## $ : int [1:3] 1 2 3
6 ## $ : chr [1:2] "a" "b"
7 ## $ : logi [1:3] TRUE FALSE TRUE
```

Assim como vimos na criação de vectores, os elementos da lista podem ter nome

```
1 > minha_Lista<-list(vector1=1:3,vector2=c("a","b"),vector3=c(TRUE,FALSE,TRUE))
2 >
3 > str(minha_Lista)
4 ## $vector1
5 ## [1] 1 2 3
6 ##
7 ## $vector2
8 ## [1] "a" "b"
9 ##
10 ## $vector3
11 ## [1] TRUE FALSE TRUE
```


Lista

Uma outra forma de criar uma lista é usando a função `vector()`

```
1 > lista_vazia<-vector(mode='list', length=5)
2 >
3 > lista_vazia
4 ## [[1]]
5 ## NULL
6 ##
7 ## [[2]]
8 ## NULL
9 ##
10 ## [[3]]
11 ## NULL
12 ##
13 ## [[4]]
14 ## NULL
15 ##
16 ## [[5]]
17 ## NULL
```

Data frame

Colecção de vectores com igual tamanho

- Estrutura bidimensional disposta em linhas e colunas
 - **MAS:** DENTRO de uma coluna, cada célula deve ter o mesmo tipo de dados!
- `data frames` são usadas para representar todo conjunto de dados
- As colunas contêm vectores com diferentes tipos de dados

Data frame-layout

The diagram illustrates the structure of a data frame. At the top, the word "Columns" is written in blue, with four blue arrows pointing down to the column headers: "Name", "Team", "Number", "Position", and "Age". On the left side, the word "Rows" is written in orange, with four orange arrows pointing right to the row indices: 0, 1, 2, and 3. The data frame itself is a table with 7 rows (indices 0 to 6) and 5 columns. The rows are highlighted with alternating light green and white backgrounds. Several cells are highlighted with pink boxes: "Jonas Jerebko" in row 2, "Boston Celtics" in row 3, "8.0" in row 2, "NaN" in row 3, "PG" in row 4, "NaN" in row 5, and "27.0" in row 6. A pink line labeled "Data" in pink text connects these highlighted cells, indicating the data content of the frame.

	<i>Name</i>	<i>Team</i>	<i>Number</i>	<i>Position</i>	<i>Age</i>
0	Avery Bradley	Boston Celtics	0.0	PG	25.0
1	John Holland	Boston Celtics	30.0	SG	27.0
2	Jonas Jerebko	Boston Celtics	8.0	PF	29.0
3	Jordan Mickey	Boston Celtics	NaN	PF	21.0
4	Terry Rozier	Boston Celtics	12.0	PG	22.0
5	Jared Sullinger	Boston Celtics	7.0	C	NaN
6	Evan Turner	Boston Celtics	11.0	SG	27.0

Uma data frame pode ser criada usando a função `data.frame()`.

Data frame

Exemplo de criação de data frame

```
1 > df <- data.frame(ID = 1:3, Sexo = c("F", "F", "M"),  
2 + Peso = c(71, 60, 65))  
3 >  
4 ≥ df <- data.frame(  
5 + ID = 1:3,                # elementos da primeira coluna  
6 + Sexo = c("F", "F", "M"), # elementos da segunda coluna  
7 + Peso = c(71, 60, 65))    # elementos da terceira coluna
```

Data frame

Exemplo de criação de data frame

```
1 > df <- data.frame(ID = 1:3, Sexo = c("F", "F", "M"),  
2 + Peso = c(71, 60, 65))  
3 >  
4 > df  
5   ID Sexo Peso  
6 1 1    F   71  
7 2 2    F   60  
8 3 3    M   65
```

Data frame

Nota: Data frame, automaticamente, converte strings em factores.

```
1 > df_2<-data.frame(x=1:3,y=c("a","b","c"))
2 >
3 ≥
4 ≥ str(df_2) #  exhibe a estrutura interna dos dados
5 'data.frame':  3 obs. of  2 variables:
6 $ x: int  1 2 3
7 $ y: chr  "a" "b" "c"
```

Data frame



- Criar uma `data frame` manualmente leva muito tempo
- Além disso, a digitação convida erros
- Deve evitar digitar grandes conjuntos de dados no R manualmente

Data frame



- Criar uma `data frame` manualmente leva muito tempo
- Além disso, a digitação convida erros
- Deve evitar digitar grandes conjuntos de dados no R manualmente

As `data frames`, geralmente, são importadas usando as funções `read.csv()` e `read.table()`