

Introdução à linguagem R

Jean S. S. Resende, Jéssica M. Magno, João C. D. Muzzi, Mauro A. A. Castro

2023-06-03

Contents

1	Prefácio	5
2	Introdução	7
2.1	Contextualizando a linguagem de programação R	7
2.2	Instalação do R	7
2.3	Instalação do RStudio	8
2.4	Interface do RStudio	8
2.5	Pacotes	8
2.6	Começando de fato a programar em R	9
2.7	Acessando o manual da função	10
2.8	Comentando códigos no R	10
3	Fundamentos básicos da programação	13
3.1	Condições	13
3.2	<i>Loops</i>	14
4	Parts	15
5	Footnotes and citations	17
5.1	Footnotes	17
5.2	Citations	17

6	Blocks	19
6.1	Equations	19
6.2	Theorems and proofs	19
6.3	Callout blocks	19
7	Sharing your book	21
7.1	Publishing	21
7.2	404 pages	21
7.3	Metadata for sharing	21

Chapter 1

Prefácio

A linguagem R foi criada por professores do departamento de estatística da universidade de Auckland no ano 2000. A intenção destes professores era disponibilizar uma linguagem *open-source* para computação estatística. Com o avanço e popularização da linguagem, profissionais de diversas áreas passaram a utilizá-la em suas análises de dados.

Esta apostila contém uma introdução à linguagem de programação R. Aborda desde conteúdos teóricos quanto práticos, com exemplos didáticos a fim de facilitar o aprendizado. Esta apostila foi produzida principalmente pelos autores: Jean Silva de Souza Resende, Jéssica Maria Magno, João Carlos Degram Muzzi sob orientação de Mauro Antônio Alves Castro. Em versões anteriores, tivemos a colaboração dos autores: Sheyla Trefflich, Danrley R. Fernandes e Giuseppe Pasqualato Neto.

Chapter 2

Introdução

2.1 Contextualizando a linguagem de programação R

A linguagem R é uma linguagem de programação com o foco em computação estatística e manipulação de gráficos. Criada no início dos anos 90 por Geroge Ross Ihaka e Robert Clifford Gentleman, o R é usado mais utilizado por estatísticos, bioinformatas, analistas de dados e desenvolvedor de *software* estatístico. No entanto ele tem se destacado na comunidade científica. Em maio de 2023, o R ocupava a 16^a posição no índice TIOBE, uma medida de popularidade da linguagem de programação, sendo que em agosto de 2020 o R atingiu seu pico em ficando 8^o lugar.

O R é um ambiente de *software* livre de código aberto, disponível sob a *GNU General Public License*. Seus executáveis pré-compilados são fornecidos para vários sistemas operacionais. Ele tem uma interface de linha de comando, mas também possui interfaces gráficas de usuário (GUI) de terceiros como o Rstudio - que será a IDE (*Integrated Development Envirenment*) que iremos utilizar na apostila.

2.2 Instalação do R

1. Acesse o repositório do R (clique [aqui](#)).
2. Acesse o link referente ao seu sistema operacional: Linux, macOS ou Windows.
 - 2.1. Linux: escolha a distribuição linux (debian, fedora, redhat, suse ou ubuntu) e então prossiga com os comandos no terminal.
 - 2.2. macOS: escolha o instalador conforme o modelo da sua máquina e execute-o.
 - 2.3. Windows: acesse o link Base e então baixe o instalador e execute-o.

2.3 Instalação do RStudio

1. Clique aqui para acessar o repositório do RStudio.
2. Baixe o instalador conforme o sistema operacional da sua máquina (Linux/macOS/Windows).

2.4 Interface do RStudio

Por padrão o RStudio abre quatro janelas (pode ocorrer de uma estar oculta, mas observe o botão de minimizar/maximizar no canto superior direito de cada janela).

- **Editor de código** (*janela do canto superior esquerdo*): Nela você digita os comando a serem executados no RStudio. Para executá-los aperte as teclas ‘CTRL’ e ‘ENTER’ simultaneamente na linha ou bloco de código selecionado.
- **Console** (*janela do canto inferior esquerdo*): É visto as saídas dos comandos que são rodados. Também é possível digitar e rodar códigos diretamente nesta janela.
- **Histórico** (*janela do canto superior direito*): Nesta janela ficam salvos os objetos, históricos de comandos e conexões com outros aplicativos.
- **Visualização** (*janela do canto inferior direito*): Aqui você pode visualizar os gráficos no RStudio, navegar entre os arquivos do seu computador, visualizar os pacotes instalados e ver a ajuda de comandos e descrições de tabelas de dados e por fim navegar entre os arquivos html.

2.5 Pacotes

Por ser *Open Source*, o R permite que qualquer usuário disponibilize funções e bancos de dados a comunidade. As funções/bancos de dados são disponibilizados através de pacotes. A instalação de um pacote depende do repositório que ele está armazenado: máquina local, CRAN, GitHub, Bioconductor, entre outros. O repositório CRAN contém muitos pacotes e não é direcionado à uma área específica (como é o caso do Bioconductor que se destina a pacotes voltados para área de biotecnologia). A instalação de um pacote do repositório CRAN é feita pelo menu “Tools > Install Packages” ou simplesmente utilizando o seguinte comando:

```
installed.packages("nomeDoPacote")
```

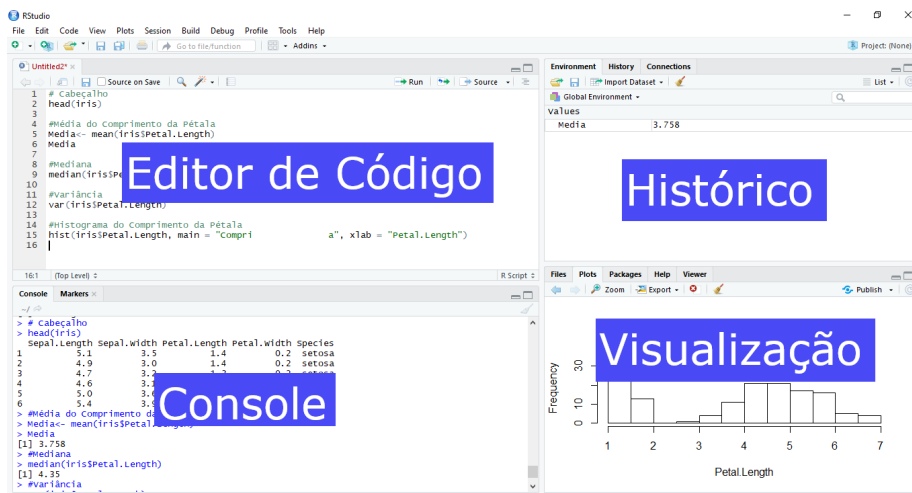



Figure 2.1: Janelas do RStudio. Ref: <https://www.est.ufmg.br/~cristianocs/Pacotes2021/Intro.html#6>

Para que você possa utilizar as funções do pacote que instalou, você deve usar um dos dois comandos a seguir, para de fato carregar as funções do pacote para o ambiente R:

```
library(nomeDoPacote)
require(nomeDoPacote)
```

A função `library()` é utilizada normalmente no corpo do script, enquanto que a função `require()` é utilizada dentro de outras funções.

2.6 Começando de fato a programar em R

No console (janela do canto inferior esquerdo) digite o comando a seguir e tecle ENTER:

```
print("Hello World")
```

Agora digite o mesmo comando no editor de código (janela do canto superior esquerdo) e com o cursor na mesma linha do comando, tecle CTRL e ENTER simultaneamente:

```
print("Hello World")
```

A diferença é que quando executamos os comandos no editor de código, o comando continua no editor para ser executado, ou seja, você está construindo um script. Mas você executa comando diretamente no console, eles não ficam gravados em um editor.

O dado de saída da função, foi um `print` do que estava dentro da função. Mas como você saber o que usar dentro de uma determinada função, como `print()`? Você precisa acessar o manual desta função.

2.7 Acessando o manual da função

Esta é uma etapa muito importante que antecede a sua caminhada no aprendizado do R. Você pode visualizar o manual da função executando um comando onde um ponto de interrogação (?) precisa anteceder a função:

```
?print
```

Mas se você deseja encontrar funções que realizam uma determinada ação, basta inserir dois pontos de interrogação antecedendo a ação desejada:

```
??priting
```

O comando acima realizará uma busca por tópicos que contenham a palavra *plotting*. Outra opção alternativa ao `?` é o uso da função `help()` e `help.search()` para `??`

```
help("print")  
help.search("priting")
```

Algumas funções possuem exemplos de sua execução. Se você quer saber como utilizar uma determinada função através de exemplos, execute a função `example()`.

```
example("print")
```

2.8 Comentando códigos no R

A maioria das linguagens de programação e até linguagem de marcação, possuem uma forma de inserção de textos que não serão executados pela linguagem. Esse procedimento é denominado de comentário. Você pode comentar os seus códigos. Isso é algo essencial para todos os programadores, indiferente da linguagem. Pois, códigos comentados facilitam a interpretação do mesmo por

outros programadores e até mesmo pelo autor, devido a um período de tempo que se passou desde a criação daquele código.

Para comentar linhas no R você precisa inserir o # antes do que seria o comentário:

```
# isto é um comentário
```

Exemplo aplicado:

```
print("Hello World") # imprimindo na tela Hello World
```

Perceba que o conteúdo após o # não é interpretado no R, ou seja, este conteúdo é um comentário.

Chapter 3

Fundamentos básicos da programação

Existem dois passos que são trilhados por toda linguagem de programação, e alguns programadores dizem que se uma linguagem de programação não permite a execução destes dois passos, ela não é bem considerada uma linguagem de programação. Um exemplo é a linguagem HTML, essa linguagem é dita como **linguagem de marcação** sua finalidade é trabalhar com estruturação de textos. Não iremos utilizá-la para cálculos ou procedimentos que demandam de uma rotina computacional com base em cálculos e nos dois passos. Mas quais são estes dois passos? R: condições e *loops*.

3.1 Condições

Se alguma coisa for verdadeira (TRUE) o R vai agir de uma maneira, caso seja mentira (FALSE) ele vai agir de outra maneira. Você pode estabelecer algumas condições para que seja feita uma função. Veremos alguns operadores de comparação e operadores lógicos utilizados na linguagem R para trabalhar com condições.

3.1.1 Operadores de comparação

- igualdade: ==
- diferente: !=
- menor: <
- maior: >
- menor ou igual: <=
- maior ou igual: >=

3.1.2 Operadores lógicos

- E: &&
- OU: ||

3.1.2.1 Condição: if (função if())

Determinado código será executado somente se a condição for verdadeira, abaixo é apresentado a estrutura do if.

```
# -- estrutura  
  
# if(condicao){  
#   comandos a serem executados  
# }
```

Vamos agora fazer uma aplicação: se o número dois for maior que o número um, então imprima na tela a frase: dois é maior que um. Caso contrário não faça nada.

```
# -- aplicacao  
## -- verdadeiro  
if(2>1){  
  print("dois é maior que um")  
}
```

```
## [1] "dois é maior que um"
```

```
## -- falso  
if(2<1){  
  print("dois é menor que um")  
}
```

3.2 Loops

Chapter 4

Parts

You can add parts to organize one or more book chapters together. Parts can be inserted at the top of an .Rmd file, before the first-level chapter heading in that same file.

Add a numbered part: `# (PART) Act one {-}` (followed by `# A chapter`)

Add an unnumbered part: `# (PART*) Act one {-}` (followed by `# A chapter`)

Add an appendix as a special kind of un-numbered part: `# (APPENDIX) Other stuff {-}` (followed by `# A chapter`). Chapters in an appendix are prepended with letters instead of numbers.

Chapter 5

Footnotes and citations

5.1 Footnotes

Footnotes are put inside the square brackets after a caret `^[]`. Like this one ¹.

5.2 Citations

Reference items in your bibliography file(s) using `@key`.

For example, we are using the **bookdown** package [Xie, 2023] (check out the last code chunk in `index.Rmd` to see how this citation key was added) in this sample book, which was built on top of R Markdown and **knitr** [Xie, 2015] (this citation was added manually in an external file `book.bib`). Note that the `.bib` files need to be listed in the `index.Rmd` with the YAML `bibliography` key.

The RStudio Visual Markdown Editor can also make it easier to insert citations: <https://rstudio.github.io/visual-markdown-editing/#/citations>

¹This is a footnote.

Chapter 6

Blocks

6.1 Equations

Here is an equation.

$$f(k) = \binom{n}{k} p^k (1-p)^{n-k} \quad (6.1)$$

You may refer to using `\@ref{eq:binom}`, like see Equation (6.1).

6.2 Theorems and proofs

Labeled theorems can be referenced in text using `\@ref{thm:tri}`, for example, check out this smart theorem 6.1.

Theorem 6.1. *For a right triangle, if c denotes the length of the hypotenuse and a and b denote the lengths of the **other** two sides, we have*

$$a^2 + b^2 = c^2$$

Read more here <https://bookdown.org/yihui/bookdown/markdown-extensions-by-bookdown.html>.

6.3 Callout blocks

The R Markdown Cookbook provides more help on how to use custom blocks to design your own callouts: <https://bookdown.org/yihui/rmarkdown-cookbook/custom-blocks.html>

Chapter 7

Sharing your book

7.1 Publishing

HTML books can be published online, see: <https://bookdown.org/yihui/bookdown/publishing.html>

7.2 404 pages

By default, users will be directed to a 404 page if they try to access a webpage that cannot be found. If you'd like to customize your 404 page instead of using the default, you may add either a `_404.Rmd` or `_404.md` file to your project root and use code and/or Markdown syntax.

7.3 Metadata for sharing

Bookdown HTML books will provide HTML metadata for social sharing on platforms like Twitter, Facebook, and LinkedIn, using information you provide in the `index.Rmd` YAML. To setup, set the `url` for your book and the path to your `cover-image` file. Your book's `title` and `description` are also used.

This `gitbook` uses the same social sharing data across all chapters in your book—all links shared will look the same.

Specify your book's source repository on GitHub using the `edit` key under the configuration options in the `_output.yml` file, which allows users to suggest an edit by linking to a chapter's source file.

Read more about the features of this output format here:

<https://pkgs.rstudio.com/bookdown/reference/gitbook.html>

Or use:

```
?bookdown::gitbook
```

Bibliography

Yihui Xie. *Dynamic Documents with R and knitr*. Chapman and Hall/CRC, Boca Raton, Florida, 2nd edition, 2015. URL <http://yihui.org/knitr/>. ISBN 978-1498716963.

Yihui Xie. *bookdown: Authoring Books and Technical Documents with R Markdown*, 2023. URL <https://CRAN.R-project.org/package=bookdown>. R package version 0.34.