

Comandos básicos do R: Guia de bolso

Lucas C. Germano

2022-05-14

Contents

Sobre este livro	5
1 Leitura de arquivos de texto	7
1.1 Definição de diretório de trabalho	7
1.2 Arquivos .csv	7
2 Cross-references	17
2.1 Chapters and sub-chapters	17
2.2 Captioned figures and tables	17
3 Parts	21
4 Footnotes and citations	23
4.1 Footnotes	23
4.2 Citations	23
5 Blocks	25
5.1 Equations	25
5.2 Theorems and proofs	25
5.3 Callout blocks	25
6 Sharing your book	27
6.1 Publishing	27
6.2 404 pages	27
6.3 Metadata for sharing	27

Sobre este livro

Sejam bem-vindos!

O objetivo deste livro é disponibilizar para consulta anotações de códigos R de forma prática e rápida. Não há explicações aprofundadas nem se pretende esgotar as possibilidades do conteúdo apresentado, assim, esta documentação deve ser utilizada somente como um guia rápido, pois não passa de um conjunto de rascunhos apreendidos no dia-a-dia da manipulação de dados e na apresentação de resultados. O conteúdo poderá ser baixado nos formatos **.pdf** ou **epub**, mas a proposta é que o conteúdo seja dinâmico, com atualizações semanais. A estrutura de construção está disponível no GitHub.

Críticas, sugestões ou contribuições de código e conteúdo podem ser enviadas para lucascgermano@gmail.com. Ficarei muito feliz, qualquer que seja o motivo do contato.

Chapter 1

Leitura de arquivos de texto

1.1 Definição de diretório de trabalho

Comando	Definição
<code>setwd()</code>	Define diretório de trabalho.
<code>Ctrl + Shift + h</code>	Abre janela de navegação para definir diretório.
<code>file.choose()</code>	Abre janela de navegação e ao selecionar o arquivo, ele retorna o caminho (diretório). Pode-se usar também dentro do comando, como em <code>read.csv2(file = file.choose())</code> .
No RStudio: Session > Setting Working Directory	Equivalente a <code>Ctrl + Shift + h</code>
Inserir aspas ' ' + Tab entre elas	Navegação que pode servir para explorar caminhos.

1.2 Arquivos .csv

1.2.1 `read.csv2()`

`.csv` = Arquivos separados por vírgula (,)

`.csv2` = Arquivos separados por ponto e vírgula (;)

Os argumentos das funções são os mesmos, por isso o exemplo será dado somente para `.csv2` (mais usado)

```
dados <- read.csv2(file = 'dados/dados.csv')
head(dados, 5)           # Exibir as 5 primeiras linhas dos dados.
```

```
##      X      data code_mn      muni  faixa casos obitos masc fem  ano mes semana
## 1 1 2020-01-01 353070 Mogi Guaçu 30 a 39      1      0      0      1 2020      1      1
## 2 2 2020-01-20 353070 Mogi Guaçu 50 a 59      1      0      1      0 2020      1      3
## 3 3 2020-01-29 352380      Itobi 30 a 39      1      0      1      0 2020      1      5
## 4 4 2020-01-30 353050      Mococa 70 a 79      1      0      0      1 2020      1      5
## 5 5 2020-02-02 353080 Mogi Mirim 40 a 49      1      0      0      1 2020      2      5
##      pop
## 1 150713
## 2 150713
## 3      7830
## 4 68788
## 5 92715
```

Argumentos principais

Os argumentos são os mesmos da função `read.table()`.

Argumento	Definição
<code>file</code>	Nome do arquivo que será lido, contendo o caminho do diretório.
<code>header</code>	Logical. Indica se o arquivo contém os nomes das colunas na primeira linha
<code>sep</code>	Tipo de separador de campo. Default é = “;”.
<code>dec</code>	Tipo de separador de decimal. Default é = “.”.
<code>nrows</code>	Integer. Número máximo de linhas a serem lidas.
<code>skip</code>	Integer. Número de linhas que serão puladas antes de iniciar a leitura dos dados.
<code>fill</code>	Logical. Se TRUE, caso as linhas tenham comprimento desigual, são adicionados campos em branco.
<code>blank.lines.skip</code>	Logical. Se TRUE linhas vazias serão ignoradas.

Argumento	Definição
stringsAsFactors	Logical. Se TRUE os vetores character serão convertidos para factors. Se houver distorção dos caracteres, utilizar FALSE para serem mantidos os caracteres originais, sem conversão.
fileEncoding	Character string. Define o encoding que será usado. Ex. fileEncoding = “UTF-8” ou “Latin-1” ou “ISO-8859-1”.
skipNull	Logical. Se TRUE os nulos (NA) devem ser ignorados.
colClasses	character. Um vetor de classes referentes as colunas. Valores possíveis são NA (default, quando type.convert é usado), “NULL” (quando a coluna é pulada), um vetor atômico de classes (logical, integer, numeric, complex, character, raw), or “factor”, “Date” or “POSIXct”.

1.2.2 readr::read_csv2()

Exemplo 1

```
dados <- readr::read_csv2(file = 'dados/dados.csv', # Caminho e arquivo
                          col_select = c(2,4:7),    # Seleção de colunas de forma numérica (é in
                          guess_max = 1000,          # Máximo de linhas utilizadas para adivinha
                          skip_empty_rows = TRUE)    # Pular linhas vazias
head(dados, 5)                                     # Exibir as 5 primeiras linhas dos dados.
```

```
## # A tibble: 5 x 5
##   data      muni      faixa  casos obitos
##   <date>    <chr>    <chr>   <dbl> <dbl>
## 1 2020-01-01 Mogi Guaçu 30 a 39     1     0
## 2 2020-01-20 Mogi Guaçu 50 a 59     1     0
## 3 2020-01-29 Itobi     30 a 39     1     0
## 4 2020-01-30 Mococa    70 a 79     1     0
## 5 2020-02-02 Mogi Mirim 40 a 49     1     0
```

Exemplo 2

```

dados <- readr::read_csv2('dados/dados.csv',           # Caminho e arquivo
                          guess_max = 1000,           # Máximo de linhas uti
                          skip_empty_rows = TRUE,     # Pular linhas vazias
                          skip = 1,                  # Pular primeira linha
                          col_names = c('a','b','c','d','e'), # Definir nomes das co
                          col_select = c('a','b','c','d','e')) # Selecionar colunas
head(dados, 5)                                       # Exibir as 5 primeira

```

```

## # A tibble: 5 x 5
##       a b      c d      e
##   <dbl> <date>   <dbl> <chr>   <chr>
## 1     1 2020-01-01 353070 Mogi Guaçu 30 a 39
## 2     2 2020-01-20 353070 Mogi Guaçu 50 a 59
## 3     3 2020-01-29 352380 Itobi      30 a 39
## 4     4 2020-01-30 353050 Mococa      70 a 79
## 5     5 2020-02-02 353080 Mogi Mirim 40 a 49

```

Argumentos principais

Argumento	Definição
file	Nome do arquivo que será lido, contendo o caminho do diretório (admite http). Arquivos terminados em .gz, .bz2, .xz, ou .zip serão automaticamente descomprimidos.
col_names	TRUE ou FALSE ou um vetor tipo caracter com nomes das colunas. Se TRUE, a primeira linha será usada para nomear as colunas. Se FALSE, nomes das colunas serão gerados automaticamente (X1, X2, X3 etc). Se col_names for um vetor com nomes, os valores serão usados como nomes das colunas, mas a primeira linha será considerada no banco (nomes errados), assim, pode-se usar o argumento renomeando as colunas, mas fazendo a leitura sem considerar a primeira linha, com [-1,] ou skip = 1. Colunas sem nome (NA) receberão nomes fictícios.

Argumento	Definição
<code>col_types</code>	Se for NULL, todos as classes de coluna serão imputadas a partir do máximo de linhas lidas (<code>guess_max</code>) intercaladas por todo o arquivo. Se a imputação falhar, você precisará aumentar o <code>guess_max</code> ou fornecer os tipos corretos você mesmo. As especificações de coluna criadas por <code>list()</code> ou <code>cols()</code> devem conter uma especificação de coluna para cada coluna. Se você quiser ler apenas um subconjunto das colunas, use <code>cols_only()</code> . Para compactar um vetor com as classes, usar as letras <code>c</code> = character, <code>i</code> = integer, <code>n</code> = number, <code>d</code> = double, <code>l</code> = logical, <code>f</code> = factor, <code>D</code> = date, <code>T</code> = date time, <code>t</code> = time, <code>?</code> = guess, <code>_</code> or <code>-</code> = skip; Por padrão, a leitura de um arquivo sem uma especificação de coluna imprimirá uma mensagem mostrando o que o leitor adivinhou. Para remover esta mensagem, defina <code>show_col_types = FALSE</code> ou defina <code>'options(readr.show_col_types = FALSE)</code> .
<code>col_select</code>	Colunas a serem incluídas nos resultados, equivale a <code>dplyr::select()</code> para se referir às colunas pelo nome. Use <code>c()</code> ou <code>list()</code> para usar mais de uma expressão de seleção. Embora esse uso seja menos comum, <code>col_select</code> também aceita um índice de coluna numérica.
<code>locale</code>	A localidade controla os padrões que variam de lugar para lugar. A localidade padrão é centrada nos EUA (como R), mas você pode usar <code>locale()</code> para criar sua própria localidade que controla coisas como o fuso horário padrão, codificação, marca decimal, marca grande e nomes de dia/mês.

Argumento	Definição
na	Vetor de caracteres de strings para interpretar como valores ausentes. Defina esta opção como <code>character()</code> para indicar que não há valores ausentes.
trim_ws	Os espaços em branco à esquerda e à direita (espaços e tabulações ASCII) devem ser cortados de cada campo antes de analisá-lo?
skip	Número de linhas para pular antes de ler os dados.
n_max	Número máximo de linhas a ler.
guess_max	Número máximo de linhas a serem usadas para adivinhar os tipos de coluna.
show_col_types	Se FALSE, não mostre os tipos de coluna adivinhados. Se TRUE sempre mostra os tipos de coluna, mesmo que sejam fornecidos. Se NULL (o padrão) mostrar apenas os tipos de coluna se eles não forem fornecidos explicitamente pelo argumento <code>col_types</code> .
skip_empty_rows	As linhas em branco devem ser ignoradas completamente? ou seja, se esta opção for TRUE, as linhas em branco não serão representadas. Se for FALSE, eles serão representados por valores NA em todas as colunas.

1.2.3 `data.table::fread()`

Vantagem de realizar a leitura de arquivos de grande extensão de forma rápida. Além disso, tem boa identificação de separador, encoding e tipos de classe.

Exemplo 1

Argumentos principais

Argumento	Definição
file	Nome do arquivo no diretório de trabalho, caminho para o arquivo ou um URL começando http://, file://, etc. Arquivos compactados ‘gz’ e ‘bz2’ são suportados se o pacote R.utils estiver instalado.
sep	O separador entre colunas. O padrão é o caractere no conjunto [.,;:].
nrows	Número máximo de linhas a serem lidas.
header	Logical. Primeira linha é o nome das colunas.
na.strings	Para ler NA, como NA, defina na.strings=“NA”. Para ler „ como string em branco “”, defina na.strings=NULL.
stringsAsFactors	Converter todas as colunas de caracteres em fatores?
skip	skip>0 ignora as primeiras linhas. skip=“string” procura por “string” no arquivo (por exemplo, uma substring da linha de nomes de coluna) e começa nessa linha (inspirada em read.xls no pacote gdata).
select	Um vetor de nomes de colunas ou números para manter e eliminar as demais. Pode especificar também tipos da mesma forma que colClasses; ou seja, um vetor de pares colname=type, ou uma lista de pares type=col(s). Em todas as formas de seleção, a ordem em que as colunas são especificadas determina a ordem das colunas no resultado.
drop	Vetor de nomes de colunas ou números a serem descartados, mantenha o resto.

Argumento	Definição
<code>colClasses</code>	Pode receber um vetor nomeado especificando tipos para um subconjunto das colunas por nome. O padrão NULL significa que os tipos são inferidos dos dados no arquivo. Suporta uma lista nomeada de vetores de nomes de colunas ou números onde os nomes das listas são os nomes das classes (procurar exemplos na internet).
<code>dec</code>	Separador de decimal como em <code>read.csv2</code> .
<code>col.names</code>	Inserir um vetor de nomes para as colunas se quiser substituir os originais.

`check.names` default is FALSE. If TRUE then the names of the variables in the `data.table` are checked to ensure that they are syntactically valid variable names. If necessary they are adjusted (by `make.names`) so that they are, and also to ensure that there are no duplicates.

`encoding`

default is “unknown”. Other possible options are “UTF-8” and “Latin-1”. Note: it is not used to re-encode the input, rather enables handling of encoded strings in their native encoding.

`quote`

By default (“”), if a field starts with a double quote, `fread` handles embedded quotes robustly as explained under Details. If it fails, then another attempt is made to read the field as is, i.e., as if quotes are disabled. By setting `quote=“ “`, the field is always read as if quotes are disabled. It is not expected to ever need to pass anything other than “” to `quote`; i.e., to turn it off.

`strip.white` default is TRUE. Strips leading and trailing whitespaces of unquoted fields. If FALSE, only header trailing spaces are removed.

`fill`

logical (default is FALSE). If TRUE then in case the rows have unequal length, blank fields are implicitly filled.

`blank.lines.skip`

logical, default is FALSE. If TRUE blank lines in the input are ignored.

`key` Character vector of one or more column names which is passed to `setkey`. It may be a single comma separated string such as `key=“x,y,z”`, or a vector of names such as `key=c(“x”, “y”, “z”)`. Only valid when argument

`data.table=TRUE`. Where applicable, this should refer to column names given in `col.names`.

`index`

Character vector or list of character vectors of one or more column names which is passed to `setindexv`. As with `key`, comma-separated notation like `index="x,y,z"` is accepted for convenience. Only valid when argument `data.table=TRUE`. Where applicable, this should refer to column names given in `col.names`.

`showProgress`

`TRUE` displays progress on the console if the ETA is greater than 3 seconds. It is produced in `fread`'s C code where the very nice (but R level) `txtProgressBar` and `tkProgressBar` are not easily available.

`data.table`

`TRUE` returns a `data.table`. `FALSE` returns a `data.frame`. The default for this argument can be changed with `options(datatable.fread.datatable=FALSE)`.

`nThread` The number of threads to use. Experiment to see what works best for your data on your hardware.

`logical01`

If `TRUE` a column containing only 0s and 1s will be read as logical, otherwise as integer.

`keepLeadingZeros`

If `TRUE` a column containing numeric data with leading zeros will be read as character, otherwise leading zeros will be removed and converted to numeric.

`yaml`

If `TRUE`, `fread` will attempt to parse (using `yaml.load`) the top of the input as YAML, and further to glean parameters relevant to improving the performance of `fread` on the data itself. The entire YAML section is returned as parsed into a list in the `yaml_metadata` attribute. See Details.

`autostart`

Deprecated and ignored with warning. Please use `skip` instead.

`tmpdir`

Directory to use as the `tmpdir` argument for any `tempfile` calls, e.g. when the input is a URL or a shell command. The default is `tempdir()` which can be controlled by setting `TMPDIR` before starting the R session; see `base::tempdir`.

`tz`

Relevant to datetime values which have no Z or UTC-offset at the end, i.e. unmarked datetime, as written by `utils::write.csv`. The default `tz="UTC"` reads unmarked datetime as UTC POSIXct efficiently. `tz=""` reads unmarked datetime as type character (slowly) so that `as.POSIXct` can interpret (slowly) the character datetimes in local timezone; e.g. by using "POSIXct" in `colClasses=`. Note that `fwrite()` by default writes datetime in UTC including the final Z and

therefore `fwrite`'s output will be read by `fread` consistently and quickly without needing to use `tz=` or `colClasses=`. If the `TZ` environment variable is set to "UTC" (or " " on non-Windows where unset vs " " is significant) then the R session's timezone is already UTC and `tz=""` will result in unmarked datetimes being read as UTC POSIXct. For more information, please see the news items from v1.13.0 and v1.14.0.

Chapter 2

Cross-references

Cross-references make it easier for your readers to find and link to elements in your book.

2.1 Chapters and sub-chapters

There are two steps to cross-reference any heading:

1. Label the heading: `# Hello world {#nice-label}`.
 - Leave the label off if you like the automated heading generated based on your heading title: for example, `# Hello world = # Hello world {#hello-world}`.
 - To label an un-numbered heading, use: `# Hello world {-#nice-label}` or `{# Hello world .unnumbered}`.
2. Next, reference the labeled heading anywhere in the text using `\@ref(nice-label)`; for example, please see Chapter 2.
 - If you prefer text as the link instead of a numbered reference use: any text you want can go here.

2.2 Captioned figures and tables

Figures and tables *with captions* can also be cross-referenced from elsewhere in your book using `\@ref(fig:chunk-label)` and `\@ref(tab:chunk-label)`, respectively.

See Figure 2.1.

```
par(mar = c(4, 4, .1, .1))  
plot(pressure, type = 'b', pch = 19)
```



Figure 2.1: Here is a nice figure!

Don't miss Table 2.1.

```
knitr::kable(  
  head(pressure, 10), caption = 'Here is a nice table!',  
  booktabs = TRUE  
)
```

Table 2.1: Here is a nice table!

temperature	pressure
0	0.0002
20	0.0012
40	0.0060
60	0.0300
80	0.0900
100	0.2700
120	0.7500
140	1.8500
160	4.2000
180	8.8000

Chapter 3

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 4

Footnotes and citations

4.1 Footnotes

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

4.2 Citations

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

For example, we are using the **bookdown** package [Xie, 2022] (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 5

Blocks

5.1 Equations

Here is an equation.

$$f(k) = \binom{n}{k} p^k (1-p)^{n-k} \quad (5.1)$$

You may refer to using `\@ref{eq:binom}`, like see Equation (5.1).

5.2 Theorems and proofs

Labeled theorems can be referenced in text using `\@ref{thm:tri}`, for example, check out this smart theorem 5.1.

Theorem 5.1. *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>.

5.3 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 6

Sharing your book

6.1 Publishing

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

6.2 404 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.

6.3 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*, 2022. URL <https://CRAN.R-project.org/package=bookdown>. R package version 0.26.