

# **Métodos Computacionais**

**Departamento de Estatística e Matemática Aplicada**

Ronald Targino, Rafael Braz, Juvêncio Nobre e Manoel Santos-Neto

2025-09-14

# Índice

<b>Prefácio</b>	<b>3</b>
<b>1 Introdução</b>	<b>4</b>
<b>2 Uma pequena introdução ao R</b>	<b>5</b>
Introdução . . . . .	5
Instalação . . . . .	5
Windows . . . . .	6
macOS . . . . .	6
Linux . . . . .	6
O que é o R . . . . .	10
Números, Aritmética, Atribuição e Vetores . . . . .	26
Exercícios . . . . .	32
<b>3 Motivação</b>	<b>33</b>
Da teoria à simulação . . . . .	33
Um atalho analítico útil . . . . .	34
O papel da simulação . . . . .	35
Atividade: Problema do Aniversário (22 jogadores) . . . . .	36
Exercícios . . . . .	37
<b>Referências</b>	<b>38</b>

# Prefácio

Este livro resulta de anos de experiência em sala de aula dos professores Ronald Targino, Rafael Braz, Juvêncio Nobre e Manoel Santos-Neto. Destina-se a apoiar os alunos da graduação em Estatística e do Programa de Pós-Graduação em Modelagem e Métodos Quantitativos (PPGMMQ) do Departamento de Estatística e Matemática Aplicada (DEMA) da Universidade Federal do Ceará (UFC).

Ao longo dos capítulos, abordamos a geração de números aleatórios (discretos e contínuos); métodos de suavização; simulação estocástica por inversão, rejeição e composição, bem como métodos de reamostragem; métodos de aproximação e integração; quadratura Gaussiana, integração de Monte Carlo e quadratura adaptativa; métodos de Monte Carlo em sentido amplo; amostradores MCMC, com ênfase em Gibbs e Metropolis–Hastings; otimização numérica via Newton–Raphson, Fisher scoring e quase-Newton, além do algoritmo EM; Bootstrap e Jackknife; diagnóstico de convergência; e aspectos computacionais em problemas práticos, com foco em implementação eficiente, estabilidade numérica e reprodutibilidade dos resultados.

Esperamos que este material sirva não apenas como texto-base para as disciplinas Estatística Computacional (graduação em Estatística) e Métodos Computacionais em Estatística (Mestrado-PPGMMQ), mas também como suporte para aqueles que desejam programar com qualidade na área de Estatística.

# 1 Introdução

A simulação tem um papel preponderante na estatística moderna, e suas vantagens no ensino de Estatística são conhecidas há muito tempo. Em um de seus primeiros números, o periódico *Teaching Statistics* publicou artigos que aludem precisamente a isso. Thomas e Moore (1980) afirmaram que “a introdução do computador na sala de aula escolar trouxe uma nova técnica para o ensino, a técnica da simulação”. Zieffler e Garfield (2007) e Tintle et al. (2015) discutem o papel e a importância da aprendizagem baseada em simulação no currículo de graduação em Estatística. No entanto, outros autores (por exemplo, Hodgson e Burke 2000) discutem alguns problemas que podem surgir ao ensinar uma disciplina por meio de simulação, a saber, o desenvolvimento de certos equívocos na mente dos estudantes (Martins 2018).

Todo estudante da Universidade Federal do Ceará conhece a cena. É hora do almoço no Restaurante Universitário (RU). A fila se alonga pelo pátio, colegas conversam, alguns reclamam da espera, outros aproveitam o tempo para revisar o conteúdo da próxima prova. O tempo parece correr de forma diferente quando estamos na fila. Para alguns, são apenas alguns minutos. Para outros, parece uma eternidade.

Agora, pense um pouco. Quanto tempo, em média, um aluno passa esperando para se servir? Qual a chance de alguém que chega por volta das 12h30 esperar mais de vinte minutos? Por que em alguns dias a fila anda rápido e em outros parece não ter fim?

Essas perguntas podem parecer simples, mas abrem caminho para um universo fascinante. Elas revelam como a **Probabilidade e a Estatística** estão presentes em situações que vivemos todos os dias. A fila do RU não é apenas um detalhe da rotina estudantil. Ela é um retrato de como os fenômenos aleatórios acontecem ao nosso redor. Cada chegada de estudante, cada tempo de atendimento, cada variação de um dia para o outro forma um sistema dinâmico que pode ser estudado e compreendido.

É esse o convite deste livro. Explorar como traduzir a realidade em modelos, como usar simulações para observar padrões e como a Estatística pode ajudar a responder perguntas sobre o nosso cotidiano. Mais do que fórmulas, ela é uma maneira de olhar o mundo e encontrar nele sentido.

Aprender Estatística é aprender a lidar com a incerteza. É descobrir que até na fila do almoço existe conhecimento escondido, esperando para ser revelado.

## 2 Uma pequena introdução ao R

### Introdução

O R é peça-chave em inúmeros trabalhos de pesquisa e análise de dados porque reúne, de forma prática, um conjunto amplo de técnicas estatísticas atuais, das mais básicas às mais sofisticadas, e facilita seu uso no dia a dia. Quem começa no R, porém, muitas vezes também está dando os primeiros passos em programação. Assim, além de aprender as ferramentas do R para seus objetivos, é preciso desenvolver a “cabeça” de programador. Essa fase inicial ajuda a explicar a fama de que o R é “difícil”. Mesmo assim, com prática e uma boa orientação, ele se revela bem mais acessível do que parece.

### Instalação

Este guia mostra como **baixar e instalar o R a partir do CRAN** (Comprehensive R Archive Network), com orientações específicas para **Windows**, **macOS** e **Linux**. Ao final, você testará a instalação e configurará um **espelho (mirror) brasileiro** para baixar pacotes mais rápido.

#### Nota

##### **O que é o CRAN?**

É a rede oficial de servidores que distribui o R e seus pacotes. Você pode usar o endereço inteligente <https://cloud.r-project.org> ou definir um espelho no Brasil (ex.: C3SL/UFPR).

Você pode simplesmente usar:

- **Cloud CRAN (recomendado):** <https://cloud.r-project.org> (redireciona para um espelho próximo).
- **Brasil (ex.: UFPR/C3SL):** <https://cran-r.c3sl.ufpr.br>

Mais adiante, mostraremos como fixar o mirror no R permanentemente.

## Windows

1. Acesse a página **Download R for Windows** → **base** e baixe o instalador.
2. Execute o instalador e avance com as opções padrão (recomendado para iniciantes).
3. (Opcional) Se pretende **compilar pacotes a partir do código-fonte**, instale o **Rtools** compatível com a sua versão do R.

### Dica

**Rtools:** após instalar, reinicie o R/RStudio. Em geral, o Rtools adiciona as ferramentas ao *PATH* automaticamente.

## macOS

1. Acesse **R for macOS** no CRAN e baixe o arquivo `.pkg` da versão atual.
2. Abra o `.pkg` e conclua a instalação.
3. (Opcional) Para compilar pacotes, instale também as **Command Line Tools** do Xcode:

```
xcode-select --install
```

## Linux

### Ubuntu/Debian

Opção rápida (repositório da distribuição):

```
sudo apt update
sudo apt install -y r-base
```

Para obter versões mais novas (repositório do CRAN), siga as instruções do CRAN para adicionar o repositório oficial e então:

```
sudo apt update
sudo apt install -y r-base r-base-dev
```

### Verificar a instalação

No terminal/Prompt:

```
R --version
R
```

Dentro do R:

```
version

platform      x86_64-pc-linux-gnu
arch           x86_64
os             linux-gnu
system         x86_64, linux-gnu
status
major          4
minor          5.1
year           2025
month          06
day            13
svn rev        88306
language       R
version.string  R version 4.5.1 (2025-06-13)
nickname       Great Square Root
```

Se o R abriu no console, a instalação está ok!

### Teste rápido de pacotes

No R:

```
#install.packages("tidyverse") # teste de instalação/espelho
library(tidyverse)
```

```
-- Attaching core tidyverse packages ----- tidyverse 2.0.0 --
v dplyr      1.1.4      v readr      2.1.5
v forcats    1.0.0      v stringr    1.5.1
v ggplot2    4.0.0      v tibble     3.3.0
v lubridate  1.9.3      v tidyr      1.3.1
v purrr      1.0.2
```

```
-- Conflicts ----- tidyverse_conflicts() --
```

```
x dplyr::filter() masks stats::filter()
```

```
x dplyr::lag()     masks stats::lag()
```

```
i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become
```

```
tibble(x = 1:5, y = x^2)
```

```
# A tibble: 5 x 2
```

	x	y
	<int>	<dbl>
1	1	1
2	2	4
3	3	9
4	4	16
5	5	25

Se o pacote instalou e carregou sem erros, está tudo certo.

### Fixar um mirror brasileiro do CRAN

Defina o espelho apenas nesta sessão:

```
options(repos = c(CRAN = "https://cran-r.c3sl.ufpr.br"))
install.packages("ggplot2")
```

Para tornar **permanente**, adicione a linha abaixo ao seu arquivo ~/.Rprofile:

```
options(repos = c(CRAN = "https://cloud.r-project.org"))
# ou, se preferir, o espelho da UFPR:
# options(repos = c(CRAN = "https://cran-r.c3sl.ufpr.br"))
```

### Como editar o ~/.Rprofile

- Linux/macOS:

```
echo 'options(repos = c(CRAN = "https://cloud.r-project.org"))' >> ~/.Rprofile
```

- Windows: o ~ normalmente aponta para C:\\Users\\SEU\_USUARIO\\Documents.

Você pode criar/editar C:\\Users\\SEU\_USUARIO\\Documents\\.Rprofile com um editor de texto.



## (Opcional) IDE recomendada: RStudio

Após instalar o R, instale o RStudio Desktop (Posit) para um ambiente de desenvolvimento mais amigável:

- Criação/edição de scripts
- Gerenciamento de projetos
- Visualização de plots e help integrados

## Dicas e solução de problemas

- **Permissões de administrador:** em ambientes corporativos, pode ser necessário pedir para TI instalar o R.
- **Firewall/Proxy:** se a instalação de pacotes falhar, verifique configurações de proxy e tente trocar o mirror.
- **Compatibilidade de versões:** ao compilar pacotes, garanta que as ferramentas (Rtools no Windows; CLT/Xcode no macOS) correspondam à sua versão do R.
- **Atualização do R:** ao atualizar o R, alguns pacotes precisarão ser reinstalados; use `install.packages()` novamente.
- **Testes mínimos:**

```
sessionInfo()
```

```
R version 4.5.1 (2025-06-13)
Platform: x86_64-pc-linux-gnu
Running under: Ubuntu 24.04.3 LTS
```

```
Matrix products: default
```

```
BLAS: /usr/lib/x86_64-linux-gnu/openblas-pthread/libblas.so.3
```

```
LAPACK: /usr/lib/x86_64-linux-gnu/openblas-pthread/libopenblas-p0.3.26.so; LAPACK version 3.11.0
```

```
locale:
```

```
[1] LC_CTYPE=pt_BR.UTF-8      LC_NUMERIC=C
[3] LC_TIME=pt_BR.UTF-8      LC_COLLATE=pt_BR.UTF-8
[5] LC_MONETARY=pt_BR.UTF-8  LC_MESSAGES=pt_BR.UTF-8
[7] LC_PAPER=pt_BR.UTF-8     LC_NAME=C
[9] LC_ADDRESS=C             LC_TELEPHONE=C
[11] LC_MEASUREMENT=pt_BR.UTF-8 LC_IDENTIFICATION=C
```

```
time zone: America/Recife
tzcode source: system (glibc)
```

```
attached base packages:
```

```
[1] stats      graphics  grDevices  utils      datasets  methods   base
```

```
other attached packages:
```

```
[1] lubridate_1.9.3 forcats_1.0.0  stringr_1.5.1  dplyr_1.1.4
[5] purrr_1.0.2     readr_2.1.5    tidyr_1.3.1    tibble_3.3.0
[9] ggplot2_4.0.0   tidyverse_2.0.0
```

```
loaded via a namespace (and not attached):
```

```
[1] gtable_0.3.6      jsonlite_1.8.8    compiler_4.5.1    tidyselect_1.2.1
[5] dichromat_2.0-0.1 scales_1.4.0      yaml_2.3.8        fastmap_1.1.1
[9] R6_2.6.1           generics_0.1.3    knitr_1.50        pillar_1.11.0
[13] RColorBrewer_1.1-3 tzdb_0.4.0        rlang_1.1.6       stringi_1.8.3
[17] xfun_0.52          S7_0.2.0          timechange_0.3.0  cli_3.6.5
[21] withr_3.0.2        magrittr_2.0.3    digest_0.6.34     grid_4.5.1
[25] rstudioapi_0.15.0 hms_1.1.3         lifecycle_1.0.4   vctrs_0.6.5
[29] evaluate_1.0.4     glue_1.8.0        farver_2.1.2      rmarkdown_2.29
[33] tools_4.5.1        pkgconfig_2.0.3   htmltools_0.5.8.1
```

```
capabilities() # checa recursos gráficos, etc.
```

jpeg	png	tiff	tcltk	X11	aqua
TRUE	TRUE	TRUE	TRUE	FALSE	FALSE
http/ftp	sockets	libxml	fifo	cledit	iconv
TRUE	TRUE	FALSE	TRUE	FALSE	TRUE
NLS	Rprof	profmem	cairo	ICU	long.double
TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
libcurl					
TRUE					

## O que é o R

R é uma implementação moderna da linguagem S, voltada à computação estatística e à visualização de dados. Ele reúne, no mesmo lugar, um ambiente interativo para análise e criação de gráficos e uma linguagem de programação completa.

### Nota

**Em uma frase:** R é um ambiente estatístico + uma linguagem, criada para trabalhar bem com dados, gráficos e métodos modernos.

## Principais características

- **Interativo e interpretado**, com suporte a **JIT/bytecode** via pacote **compiler**.
- **Orientado a objetos** (S3, S4 e R6) e com forte base **funcional**.
- **Modelo “tudo é objeto”**: números, vetores, data frames, funções, ambientes e modelos ajustados.
- **Vetorização nativa** e operações matriciais eficientes.
- **Ecossistema de pacotes** amplo (CRAN, Bioconductor) para estatística e ciência de dados.
- **Extensível** com C/C++/Fortran e integração com Python, SQL e serviços externos.
- **Multiplataforma** (Windows, macOS, Linux) e foco em **reprodutibilidade** (scripts, Quarto/R Markdown).

## Exemplo - “tudo é objeto”

```
x <- 1:5          # vetor (objeto)
media <- mean(x)  # função aplicada ao objeto

f <- function(z) z^2
classe_f <- class(f) # "function" - funções também são objetos

attr(x, "nota") <- "exemplo de atributo"
lista <- list(x = x, media = media, classe_f = classe_f)

str(lista)        # inspeciona a estrutura
```

```
List of 3
 $ x      : int [1:5] 1 2 3 4 5
  ..- attr(*, "nota")= chr "exemplo de atributo"
 $ media   : num 3
 $ classe_f: chr "function"
```

```
lista
```

```
$x  
[1] 1 2 3 4 5  
attr("nota")  
[1] "exemplo de atributo"
```

```
$media  
[1] 3
```

```
$classe_f  
[1] "function"
```

#### Dica

Dica: use `str(obj)` para entender a estrutura de qualquer objeto no R.  
Outras funções úteis: `class()`, `attributes()`, `typeof()`, `methods(class = ...)`.

Aqui vai como mudar o *prompt* do R para R>, na sessão atual usando:

```
options(prompt = "R> ", continue = "+")
```

Para adicionar comentários no R, como você observou nos exemplos acima, basta usar `#`, como no exemplo abaixo:

```
#Este é um comentário!  
#Que o semestre seja leve para todos nós!
```

Uma sessão ativa do R sempre está “apontando” para um diretório de trabalho. Se você não informar um caminho completo ao salvar ou importar arquivos, o R usará esse diretório por padrão. Para descobrir onde ele está:

```
getwd()
```

```
[1] "/home/manoel/Documentos/Book_Computacional_UFC"
```

Se você deseja modificar o diretório de trabalho, isso pode ser feito de maneira simples:

```
# Windows
setwd("C:/Users/SeuUsuario/Documents/meu-projeto")

# macOS / Linux
setwd("/Users/seuusuario/meu-projeto")
```

O R não vem com todos os pacotes adicionais que a comunidade desenvolve. Para usá-los, você precisa baixar e instalar a partir de um repositório, em geral, o CRAN.

```
# define um espelho (mirror) do CRAN para esta sessão
options(repos = c(CRAN = "https://cloud.r-project.org"))

# instala um ou mais pacotes
install.packages("tidyverse")
install.packages(c("ggplot2", "dplyr", "readr"))

# carrega para usar
library(tidyverse)
```

A instalação pode ser feita também facilmente usando o RStudio.

Pacotes da comunidade são atualizados com frequência para corrigir erros e acrescentar recursos. Periodicamente, vale checar se há novas versões para o que você já tem instalado. A partir do console do R, você pode verificar e atualizar com um único comando.

```
# defina um mirror (opcional, mas recomendável)
options(repos = c(CRAN = "https://cloud.r-project.org"))

# procura versões mais novas e atualiza automaticamente
update.packages(ask = FALSE, checkBuilt = TRUE)
```

- `ask = FALSE` atualiza sem perguntar pacote a pacote.
- `checkBuilt = TRUE` recompila pacotes se sua versão do R mudou.

**Ver o que está desatualizado antes:**

```
old.packages() # retorna uma tabela com pacotes que têm versão mais nova
```

	Package
bookdown	"bookdown"
credentials	"credentials"

crosstalk	"crosstalk"
curl	"curl"
DT	"DT"
emmeans	"emmeans"
evaluate	"evaluate"
gamlss	"gamlss"
gamlss.data	"gamlss.data"
GGally	"GGally"
ggfx	"ggfx"
ggimage	"ggimage"
ggmap	"ggmap"
ggspatial	"ggspatial"
gld	"gld"
kSamples	"kSamples"
magick	"magick"
osmdata	"osmdata"
pagedown	"pagedown"
patchwork	"patchwork"
PKI	"PKI"
qqplotr	"qqplotr"
ragg	"ragg"
robustbase	"robustbase"
roxygen2	"roxygen2"
rprojroot	"rprojroot"
rsconnect	"rsconnect"
rsvg	"rsvg"
sftime	"sftime"
snowflakeauth	"snowflakeauth"
TH.data	"TH.data"
usethis	"usethis"
visNetwork	"visNetwork"
xaringan	"xaringan"
xaringanthemer	"xaringanthemer"
xfun	"xfun"
yulab.utils	"yulab.utils"
askpass	"askpass"
backports	"backports"
bit	"bit"
bit64	"bit64"
broom	"broom"
bslib	"bslib"
cachem	"cachem"
callr	"callr"

cli	"cli"
colorspace	"colorspace"
commonmark	"commonmark"
cpp11	"cpp11"
crayon	"crayon"
curl	"curl"
data.table	"data.table"
DBI	"DBI"
dbplyr	"dbplyr"
digest	"digest"
dtplyr	"dtplyr"
evaluate	"evaluate"
fansi	"fansi"
farver	"farver"
fastmap	"fastmap"
fontawesome	"fontawesome"
fs	"fs"
gargle	"gargle"
generics	"generics"
ggplot2	"ggplot2"
glue	"glue"
googledrive	"googledrive"
googlesheets4	"googlesheets4"
gtable	"gtable"
haven	"haven"
highr	"highr"
htmltools	"htmltools"
httpuv	"httpuv"
jsonlite	"jsonlite"
knitr	"knitr"
later	"later"
lubridate	"lubridate"
magrittr	"magrittr"
mime	"mime"
munsell	"munsell"
openssl	"openssl"
pillar	"pillar"
pkgKitten	"pkgKitten"
processx	"processx"
promises	"promises"
ps	"ps"
purrr	"purrr"
R6	"R6"

ragg	"ragg"	
Rcpp	"Rcpp"	
readxl	"readxl"	
reprex	"reprex"	
rlang	"rlang"	
rmarkdown	"rmarkdown"	
rstudioapi	"rstudioapi"	
rvest	"rvest"	
sass	"sass"	
scales	"scales"	
shiny	"shiny"	
stringi	"stringi"	
stringr	"stringr"	
sys	"sys"	
systemfonts	"systemfonts"	
textshaping	"textshaping"	
tibble	"tibble"	
tidyselect	"tidyselect"	
tinytex	"tinytex"	
tzdb	"tzdb"	
utf8	"utf8"	
uuid	"uuid"	
withr	"withr"	
xfun	"xfun"	
xml2	"xml2"	
yaml	"yaml"	
boot	"boot"	
lattice	"lattice"	
mgcv	"mgcv"	
spatial	"spatial"	
	LibPath	Installed
bookdown	"/home/manoel/R/x86_64-pc-linux-gnu-library/4.5"	"0.43"
credentials	"/home/manoel/R/x86_64-pc-linux-gnu-library/4.5"	"2.0.2"
crosstalk	"/home/manoel/R/x86_64-pc-linux-gnu-library/4.5"	"1.2.1"
curl	"/home/manoel/R/x86_64-pc-linux-gnu-library/4.5"	"6.4.0"
DT	"/home/manoel/R/x86_64-pc-linux-gnu-library/4.5"	"0.33"
emmeans	"/home/manoel/R/x86_64-pc-linux-gnu-library/4.5"	"1.11.2"
evaluate	"/home/manoel/R/x86_64-pc-linux-gnu-library/4.5"	"1.0.4"
gamlss	"/home/manoel/R/x86_64-pc-linux-gnu-library/4.5"	"5.4-22"
gamlss.data	"/home/manoel/R/x86_64-pc-linux-gnu-library/4.5"	"6.0-6"
GGally	"/home/manoel/R/x86_64-pc-linux-gnu-library/4.5"	"2.3.0"
ggfx	"/home/manoel/R/x86_64-pc-linux-gnu-library/4.5"	"1.0.2"
ggimage	"/home/manoel/R/x86_64-pc-linux-gnu-library/4.5"	"0.3.3"



ggmap	"/home/manoel/R/x86_64-pc-linux-gnu-library/4.5"	"4.0.1"
ggspatial	"/home/manoel/R/x86_64-pc-linux-gnu-library/4.5"	"1.1.9"
gld	"/home/manoel/R/x86_64-pc-linux-gnu-library/4.5"	"2.6.7"
kSamples	"/home/manoel/R/x86_64-pc-linux-gnu-library/4.5"	"1.2-10"
magick	"/home/manoel/R/x86_64-pc-linux-gnu-library/4.5"	"2.8.7"
osmdata	"/home/manoel/R/x86_64-pc-linux-gnu-library/4.5"	"0.2.5"
pagedown	"/home/manoel/R/x86_64-pc-linux-gnu-library/4.5"	"0.22"
patchwork	"/home/manoel/R/x86_64-pc-linux-gnu-library/4.5"	"1.3.1"
PKI	"/home/manoel/R/x86_64-pc-linux-gnu-library/4.5"	"0.1-14"
qqplotr	"/home/manoel/R/x86_64-pc-linux-gnu-library/4.5"	"0.0.6"
ragg	"/home/manoel/R/x86_64-pc-linux-gnu-library/4.5"	"1.4.0"
robustbase	"/home/manoel/R/x86_64-pc-linux-gnu-library/4.5"	"0.99-4-1"
roxygen2	"/home/manoel/R/x86_64-pc-linux-gnu-library/4.5"	"7.3.2"
rprojroot	"/home/manoel/R/x86_64-pc-linux-gnu-library/4.5"	"2.1.0"
rsconnect	"/home/manoel/R/x86_64-pc-linux-gnu-library/4.5"	"1.5.0"
rsvg	"/home/manoel/R/x86_64-pc-linux-gnu-library/4.5"	"2.6.2"
sftime	"/home/manoel/R/x86_64-pc-linux-gnu-library/4.5"	"0.3.0"
snowflakeauth	"/home/manoel/R/x86_64-pc-linux-gnu-library/4.5"	"0.1.2"
TH.data	"/home/manoel/R/x86_64-pc-linux-gnu-library/4.5"	"1.1-3"
usethis	"/home/manoel/R/x86_64-pc-linux-gnu-library/4.5"	"3.1.0"
visNetwork	"/home/manoel/R/x86_64-pc-linux-gnu-library/4.5"	"2.1.2"
xaringan	"/home/manoel/R/x86_64-pc-linux-gnu-library/4.5"	"0.30"
xaringanthemmer	"/home/manoel/R/x86_64-pc-linux-gnu-library/4.5"	"0.4.3"
xfun	"/home/manoel/R/x86_64-pc-linux-gnu-library/4.5"	"0.52"
yulab.utils	"/home/manoel/R/x86_64-pc-linux-gnu-library/4.5"	"0.2.0"
askpass	"/usr/lib/R/site-library"	"1.2.0"
backports	"/usr/lib/R/site-library"	"1.4.1"
bit	"/usr/lib/R/site-library"	"4.0.5"
bit64	"/usr/lib/R/site-library"	"4.0.5"
broom	"/usr/lib/R/site-library"	"1.0.5"
bslib	"/usr/lib/R/site-library"	"0.6.1"
cachem	"/usr/lib/R/site-library"	"1.0.8"
callr	"/usr/lib/R/site-library"	"3.7.5"
cli	"/usr/lib/R/site-library"	"3.6.2"
colorspace	"/usr/lib/R/site-library"	"2.1-0"
commonmark	"/usr/lib/R/site-library"	"1.9.1"
cpp11	"/usr/lib/R/site-library"	"0.4.7"
crayon	"/usr/lib/R/site-library"	"1.5.2"
curl	"/usr/lib/R/site-library"	"5.2.0"
data.table	"/usr/lib/R/site-library"	"1.14.10"
DBI	"/usr/lib/R/site-library"	"1.2.2"
dbplyr	"/usr/lib/R/site-library"	"2.4.0"
digest	"/usr/lib/R/site-library"	"0.6.34"

dtplyr	"/usr/lib/R/site-library"	"1.3.1"
evaluate	"/usr/lib/R/site-library"	"0.23"
fansi	"/usr/lib/R/site-library"	"1.0.5"
farver	"/usr/lib/R/site-library"	"2.1.1"
fastmap	"/usr/lib/R/site-library"	"1.1.1"
fontawesome	"/usr/lib/R/site-library"	"0.5.2"
fs	"/usr/lib/R/site-library"	"1.6.3"
gargle	"/usr/lib/R/site-library"	"1.5.2"
generics	"/usr/lib/R/site-library"	"0.1.3"
ggplot2	"/usr/lib/R/site-library"	"3.4.4"
glue	"/usr/lib/R/site-library"	"1.7.0"
googledrive	"/usr/lib/R/site-library"	"2.1.1"
googlesheets4	"/usr/lib/R/site-library"	"1.1.1"
gtable	"/usr/lib/R/site-library"	"0.3.4"
haven	"/usr/lib/R/site-library"	"2.5.4"
highr	"/usr/lib/R/site-library"	"0.10"
htmltools	"/usr/lib/R/site-library"	"0.5.7"
httpuv	"/usr/lib/R/site-library"	"1.6.14"
jsonlite	"/usr/lib/R/site-library"	"1.8.8"
knitr	"/usr/lib/R/site-library"	"1.45"
later	"/usr/lib/R/site-library"	"1.3.2"
lubridate	"/usr/lib/R/site-library"	"1.9.3"
magrittr	"/usr/lib/R/site-library"	"2.0.3"
mime	"/usr/lib/R/site-library"	"0.12"
munsell	"/usr/lib/R/site-library"	"0.5.0"
openssl	"/usr/lib/R/site-library"	"2.1.1"
pillar	"/usr/lib/R/site-library"	"1.9.0"
pkgKitten	"/usr/lib/R/site-library"	"0.2.3"
processx	"/usr/lib/R/site-library"	"3.8.3"
promises	"/usr/lib/R/site-library"	"1.2.1"
ps	"/usr/lib/R/site-library"	"1.7.6"
purrr	"/usr/lib/R/site-library"	"1.0.2"
R6	"/usr/lib/R/site-library"	"2.5.1"
ragg	"/usr/lib/R/site-library"	"1.2.7"
Rcpp	"/usr/lib/R/site-library"	"1.0.12"
readxl	"/usr/lib/R/site-library"	"1.4.3"
reprex	"/usr/lib/R/site-library"	"2.1.0"
rlang	"/usr/lib/R/site-library"	"1.1.3"
rmarkdown	"/usr/lib/R/site-library"	"2.25"
rstudioapi	"/usr/lib/R/site-library"	"0.15.0"
rvest	"/usr/lib/R/site-library"	"1.0.3"
sass	"/usr/lib/R/site-library"	"0.4.8"
scales	"/usr/lib/R/site-library"	"1.3.0"

shiny	"/usr/lib/R/site-library"	"1.8.0"
stringi	"/usr/lib/R/site-library"	"1.8.3"
stringr	"/usr/lib/R/site-library"	"1.5.1"
sys	"/usr/lib/R/site-library"	"3.4.2"
systemfonts	"/usr/lib/R/site-library"	"1.0.5"
textshaping	"/usr/lib/R/site-library"	"0.3.7"
tibble	"/usr/lib/R/site-library"	"3.2.1"
tidyselect	"/usr/lib/R/site-library"	"1.2.0"
tinytex	"/usr/lib/R/site-library"	"0.49"
tzdb	"/usr/lib/R/site-library"	"0.4.0"
utf8	"/usr/lib/R/site-library"	"1.2.4"
uuid	"/usr/lib/R/site-library"	"1.2-0"
withr	"/usr/lib/R/site-library"	"2.5.0"
xfun	"/usr/lib/R/site-library"	"0.41"
xml2	"/usr/lib/R/site-library"	"1.3.6"
yaml	"/usr/lib/R/site-library"	"2.3.8"
boot	"/usr/lib/R/library"	"1.3-31"
lattice	"/usr/lib/R/library"	"0.22-5"
mgcv	"/usr/lib/R/library"	"1.9-1"
spatial	"/usr/lib/R/library"	"7.3-17"
Built ReposVer Repository		
bookdown	"4.5.0" "0.44"	"https://cloud.r-project.org/src/contrib"
credentials	"4.5.1" "2.0.3"	"https://cloud.r-project.org/src/contrib"
crosstalk	"4.5.0" "1.2.2"	"https://cloud.r-project.org/src/contrib"
curl	"4.5.1" "7.0.0"	"https://cloud.r-project.org/src/contrib"
DT	"4.5.0" "0.34.0"	"https://cloud.r-project.org/src/contrib"
emmeans	"4.5.1" "1.11.2-8"	"https://cloud.r-project.org/src/contrib"
evaluate	"4.5.1" "1.0.5"	"https://cloud.r-project.org/src/contrib"
gamlss	"4.5.0" "5.5-0"	"https://cloud.r-project.org/src/contrib"
gamlss.data	"4.5.0" "6.0-7"	"https://cloud.r-project.org/src/contrib"
GGally	"4.5.1" "2.4.0"	"https://cloud.r-project.org/src/contrib"
ggfx	"4.5.1" "1.0.3"	"https://cloud.r-project.org/src/contrib"
ggimage	"4.5.0" "0.3.4"	"https://cloud.r-project.org/src/contrib"
ggmap	"4.5.1" "4.0.2"	"https://cloud.r-project.org/src/contrib"
ggspatial	"4.5.0" "1.1.10"	"https://cloud.r-project.org/src/contrib"
gld	"4.5.0" "2.6.8"	"https://cloud.r-project.org/src/contrib"
kSamples	"4.5.1" "1.2-12"	"https://cloud.r-project.org/src/contrib"
magick	"4.5.1" "2.9.0"	"https://cloud.r-project.org/src/contrib"
osmdata	"4.5.1" "0.3.0"	"https://cloud.r-project.org/src/contrib"
pagedown	"4.5.0" "0.23"	"https://cloud.r-project.org/src/contrib"
patchwork	"4.5.0" "1.3.2"	"https://cloud.r-project.org/src/contrib"
PKI	"4.5.0" "0.1-15"	"https://cloud.r-project.org/src/contrib"
qqplotr	"4.5.1" "0.0.7"	"https://cloud.r-project.org/src/contrib"

ragg	"4.5.1"	"1.5.0"	"https://cloud.r-project.org/src/contrib"
robustbase	"4.5.0"	"0.99-6"	"https://cloud.r-project.org/src/contrib"
roxygen2	"4.5.1"	"7.3.3"	"https://cloud.r-project.org/src/contrib"
rprojroot	"4.5.1"	"2.1.1"	"https://cloud.r-project.org/src/contrib"
rsconnect	"4.5.1"	"1.5.1"	"https://cloud.r-project.org/src/contrib"
rsvg	"4.5.1"	"2.7.0"	"https://cloud.r-project.org/src/contrib"
sftime	"4.5.1"	"0.3.1"	"https://cloud.r-project.org/src/contrib"
snowflakeauth	"4.5.1"	"0.2.0"	"https://cloud.r-project.org/src/contrib"
TH.data	"4.5.0"	"1.1-4"	"https://cloud.r-project.org/src/contrib"
usethis	"4.5.1"	"3.2.1"	"https://cloud.r-project.org/src/contrib"
visNetwork	"4.5.0"	"2.1.4"	"https://cloud.r-project.org/src/contrib"
xaringan	"4.5.0"	"0.31"	"https://cloud.r-project.org/src/contrib"
xaringanthemer	"4.5.0"	"0.4.4"	"https://cloud.r-project.org/src/contrib"
xfun	"4.5.1"	"0.53"	"https://cloud.r-project.org/src/contrib"
yulab.utils	"4.5.0"	"0.2.1"	"https://cloud.r-project.org/src/contrib"
askpass	"4.3.1"	"1.2.1"	"https://cloud.r-project.org/src/contrib"
backports	"4.1.2"	"1.5.0"	"https://cloud.r-project.org/src/contrib"
bit	"4.2.2"	"4.6.0"	"https://cloud.r-project.org/src/contrib"
bit64	"4.0.2"	"4.6.0-1"	"https://cloud.r-project.org/src/contrib"
broom	"4.3.0"	"1.0.10"	"https://cloud.r-project.org/src/contrib"
bslib	"4.3.2"	"0.9.0"	"https://cloud.r-project.org/src/contrib"
cachem	"4.3.0"	"1.1.0"	"https://cloud.r-project.org/src/contrib"
callr	"4.3.3"	"3.7.6"	"https://cloud.r-project.org/src/contrib"
cli	"4.3.2"	"3.6.5"	"https://cloud.r-project.org/src/contrib"
colorspace	"4.2.2"	"2.1-1"	"https://cloud.r-project.org/src/contrib"
commonmark	"4.3.2"	"2.0.0"	"https://cloud.r-project.org/src/contrib"
cpp11	"4.3.2"	"0.5.2"	"https://cloud.r-project.org/src/contrib"
crayon	"4.2.2"	"1.5.3"	"https://cloud.r-project.org/src/contrib"
curl	"4.3.3"	"7.0.0"	"https://cloud.r-project.org/src/contrib"
data.table	"4.3.2"	"1.17.8"	"https://cloud.r-project.org/src/contrib"
DBI	"4.3.2"	"1.2.3"	"https://cloud.r-project.org/src/contrib"
dbplyr	"4.3.2"	"2.5.1"	"https://cloud.r-project.org/src/contrib"
digest	"4.3.2"	"0.6.37"	"https://cloud.r-project.org/src/contrib"
dtplyr	"4.3.0"	"1.3.2"	"https://cloud.r-project.org/src/contrib"
evaluate	"4.3.2"	"1.0.5"	"https://cloud.r-project.org/src/contrib"
fansi	"4.3.2"	"1.0.6"	"https://cloud.r-project.org/src/contrib"
farver	"4.2.1"	"2.1.2"	"https://cloud.r-project.org/src/contrib"
fastmap	"4.2.2"	"1.2.0"	"https://cloud.r-project.org/src/contrib"
fontawesome	"4.3.1"	"0.5.3"	"https://cloud.r-project.org/src/contrib"
fs	"4.3.3"	"1.6.6"	"https://cloud.r-project.org/src/contrib"
gargle	"4.3.1"	"1.6.0"	"https://cloud.r-project.org/src/contrib"
generics	"4.2.1"	"0.1.4"	"https://cloud.r-project.org/src/contrib"
ggplot2	"4.3.1"	"4.0.0"	"https://cloud.r-project.org/src/contrib"

glue	"4.3.2"	"1.8.0"	"https://cloud.r-project.org/src/contrib"
googledrive	"4.3.1"	"2.1.2"	"https://cloud.r-project.org/src/contrib"
googlesheets4	"4.3.1"	"1.1.2"	"https://cloud.r-project.org/src/contrib"
gtable	"4.3.1"	"0.3.6"	"https://cloud.r-project.org/src/contrib"
haven	"4.3.2"	"2.5.5"	"https://cloud.r-project.org/src/contrib"
highr	"4.2.2"	"0.11"	"https://cloud.r-project.org/src/contrib"
htmltools	"4.3.2"	"0.5.8.1"	"https://cloud.r-project.org/src/contrib"
httpuv	"4.3.3"	"1.6.16"	"https://cloud.r-project.org/src/contrib"
jsonlite	"4.3.2"	"2.0.0"	"https://cloud.r-project.org/src/contrib"
knitr	"4.3.2"	"1.50"	"https://cloud.r-project.org/src/contrib"
later	"4.3.2"	"1.4.4"	"https://cloud.r-project.org/src/contrib"
lubridate	"4.3.1"	"1.9.4"	"https://cloud.r-project.org/src/contrib"
magrittr	"4.1.2"	"2.0.4"	"https://cloud.r-project.org/src/contrib"
mime	"4.1.1"	"0.13"	"https://cloud.r-project.org/src/contrib"
munSELL	"4.0.1"	"0.5.1"	"https://cloud.r-project.org/src/contrib"
openssl	"4.3.3"	"2.3.3"	"https://cloud.r-project.org/src/contrib"
pillar	"4.3.0"	"1.11.0"	"https://cloud.r-project.org/src/contrib"
pkgKitten	"4.2.2"	"0.2.4"	"https://cloud.r-project.org/src/contrib"
processx	"4.3.2"	"3.8.6"	"https://cloud.r-project.org/src/contrib"
promises	"4.3.1"	"1.3.3"	"https://cloud.r-project.org/src/contrib"
ps	"4.3.2"	"1.9.1"	"https://cloud.r-project.org/src/contrib"
purrr	"4.3.1"	"1.1.0"	"https://cloud.r-project.org/src/contrib"
R6	"4.1.1"	"2.6.1"	"https://cloud.r-project.org/src/contrib"
ragg	"4.3.3"	"1.5.0"	"https://cloud.r-project.org/src/contrib"
Rcpp	"4.3.2"	"1.1.0"	"https://cloud.r-project.org/src/contrib"
readxl	"4.3.1"	"1.4.5"	"https://cloud.r-project.org/src/contrib"
reprex	"4.3.2"	"2.1.1"	"https://cloud.r-project.org/src/contrib"
rlang	"4.3.2"	"1.1.6"	"https://cloud.r-project.org/src/contrib"
rmarkdown	"4.3.2"	"2.29"	"https://cloud.r-project.org/src/contrib"
rstudioapi	"4.3.1"	"0.17.1"	"https://cloud.r-project.org/src/contrib"
rvest	"4.2.1"	"1.0.5"	"https://cloud.r-project.org/src/contrib"
sass	"4.3.2"	"0.4.10"	"https://cloud.r-project.org/src/contrib"
scales	"4.3.2"	"1.4.0"	"https://cloud.r-project.org/src/contrib"
shiny	"4.3.2"	"1.11.1"	"https://cloud.r-project.org/src/contrib"
stringi	"4.3.2"	"1.8.7"	"https://cloud.r-project.org/src/contrib"
stringr	"4.3.2"	"1.5.2"	"https://cloud.r-project.org/src/contrib"
sys	"4.3.1"	"3.4.3"	"https://cloud.r-project.org/src/contrib"
systemfonts	"4.3.1"	"1.2.3"	"https://cloud.r-project.org/src/contrib"
textshaping	"4.3.1"	"1.0.3"	"https://cloud.r-project.org/src/contrib"
tibble	"4.3.1"	"3.3.0"	"https://cloud.r-project.org/src/contrib"
tidyselect	"4.2.2"	"1.2.1"	"https://cloud.r-project.org/src/contrib"
tinytex	"4.3.2"	"0.57"	"https://cloud.r-project.org/src/contrib"
tzdb	"4.3.1"	"0.5.0"	"https://cloud.r-project.org/src/contrib"

utf8	"4.3.1"	"1.2.6"	"https://cloud.r-project.org/src/contrib"
uuid	"4.3.2"	"1.2-1"	"https://cloud.r-project.org/src/contrib"
withr	"4.1.2"	"3.0.2"	"https://cloud.r-project.org/src/contrib"
xfun	"4.3.2"	"0.53"	"https://cloud.r-project.org/src/contrib"
xml2	"4.3.2"	"1.4.0"	"https://cloud.r-project.org/src/contrib"
yaml	"4.3.2"	"2.3.10"	"https://cloud.r-project.org/src/contrib"
boot	"4.4.2"	"1.3-32"	"https://cloud.r-project.org/src/contrib"
lattice	"4.3.3"	"0.22-7"	"https://cloud.r-project.org/src/contrib"
mgcv	"4.3.2"	"1.9-3"	"https://cloud.r-project.org/src/contrib"
spatial	"4.4.0"	"7.3-18"	"https://cloud.r-project.org/src/contrib"

### Atualizar apenas alguns pacotes:

```
update.packages(oldPkgs = c("gamlss", "ggmap"), ask = FALSE)
```

O R traz um conjunto amplo de arquivos de ajuda que permitem:

- Buscar funcionalidades por nome ou por tema.
- Entender como usar uma função e quais argumentos ela recebe.
- Esclarecer o papel de cada argumento e o que a função retorna.
- Ver exemplos de uso prontos para rodar.
- Saber como citar o R, pacotes e conjuntos de dados em publicações.

Agora iremos ver algumas funções úteis:

- Ajuda com função específica:

```
?lm # atalho

help("lm") #abre a documentação da função

help(package = "stats") # índice de um pacote
```

- Buscar por tema (se você não sabe o nome exato):

```
??"linear model"      # busca por tópicos
help.search("normal")  # alternativa
apropos("plot")        # lista objetos cujo nome contém "plot"
```

- Ver argumentos e formas de chamada

```
args(lm) # nomes dos argumentos (se aplicável)
formals(lm) # valores-padrão dos argumentos
```

- Exemplos, retorno e estrutura

```
example(lm) # roda os exemplos da ajuda
m <- lm(mpg ~ wt, data = mtcars)
str(m) # inspeciona o objeto retornado
methods("predict") # métodos disponíveis para um genérico
```

- Vignettes (tutoriais)

```
vignette() # lista geral
vignette(package = "ggplot2") # vignettes de um pacote
browseVignettes(package = "dplyr") # abre no navegador
```

- Como citar R/pacotes/dados

```
citation() # como citar o R
```

To cite R in publications use:

R Core Team (2025). *\_R: A Language and Environment for Statistical Computing\_*. R Foundation for Statistical Computing, Vienna, Austria.  
<<https://www.R-project.org/>>.

Uma entrada BibTeX para usuários(as) de LaTeX é

```
@Manual{,
  title = {R: A Language and Environment for Statistical Computing},
  author = {{R Core Team}},
  organization = {R Foundation for Statistical Computing},
  address = {Vienna, Austria},
  year = {2025},
  url = {https://www.R-project.org/},
}
```

We have invested a lot of time and effort in creating R, please cite it when using it for data analysis. See also 'citation("pkgname")' for citing R packages.

```
citation("ggplot2") # como citar um pacote
```

To cite ggplot2 in publications, please use

H. Wickham. ggplot2: Elegant Graphics for Data Analysis.  
Springer-Verlag New York, 2016.

Uma entrada BibTeX para usuários(as) de LaTeX é

```
@Book{,  
  author = {Hadley Wickham},  
  title = {ggplot2: Elegant Graphics for Data Analysis},  
  publisher = {Springer-Verlag New York},  
  year = {2016},  
  isbn = {978-3-319-24277-4},  
  url = {https://ggplot2.tidyverse.org},  
}
```

```
toBibtex(citation("ggplot2")) # em BibTeX
```

```
@Book{,  
  author = {Hadley Wickham},  
  title = {ggplot2: Elegant Graphics for Data Analysis},  
  publisher = {Springer-Verlag New York},  
  year = {2016},  
  isbn = {978-3-319-24277-4},  
  url = {https://ggplot2.tidyverse.org},  
}
```

- Conjuntos de dados

```
data() # abre uma lista com os datasets disponíveis
```

```
# ?mtcars # ajuda de um dataset específico
```

Console vs. editor: Você pode digitar comandos diretamente no console ou escrever um script no editor e então executá-lo. Em materiais didáticos, distinguimos os dois formatos para evitar confusão.

- 1) Código mostrado com prompt (R>) = digitado no console



- Em livros/notas, comandos de console costumam aparecer com o prompt na frente.  
Exemplo (divisão de 14 por 6):

```
R> options(prompt = "R> ", continue = "+ ")
R> 14/6
```

- Ao copiar para o R, remova o R>, o console já mostra o seu próprio prompt.

## 2) Código pensado para script (sem prompt)

- Quando o texto disser “escreva no editor e depois execute”, o código aparece sem R>.  
Exemplo (laço simples):

```
for (myitem in 5:7) {
  cat("--INÍCIO DO BLOCO--\n")
  cat("o item atual é", myitem, "\n")
  cat("--FIM DO BLOCO--\n\n")
}
```

```
--INÍCIO DO BLOCO--
o item atual é 5
--FIM DO BLOCO--
```

```
--INÍCIO DO BLOCO--
o item atual é 6
--FIM DO BLOCO--
```

```
--INÍCIO DO BLOCO--
o item atual é 7
--FIM DO BLOCO--
```

**Dica de estilo:** chaves na mesma linha do for, quebra de linha para o corpo e indentação consistente.

## 3) Linhas longas: uma linha só ou “quebradas”

- Chamadas extensas podem ficar em uma linha ou ser quebradas em pontos naturais (geralmente após vírgulas ou antes de argumentos nomeados).
- Ao quebrar, alinhe com o parêntese de abertura ou indente um nível.

Uma linha:

```
ordfac.vec <- factor(x = c("Small","Large","Large","Regular","Small"),
                    levels = c("Small","Regular","Large"),
                    ordered = TRUE)
```

Quebrada (equivalente, apenas formatação):

```
ordfac.vec2 <-
  factor(
    x      = c("Small","Large","Large","Regular","Small"),
    levels = c("Small","Regular","Large"),
    ordered = TRUE
  )
identical(ordfac.vec, ordfac.vec2)
```

```
[1] TRUE
```

## Números, Aritmética, Atribuição e Vetores

### 1) Números (*numerics*)

Em R, o tipo numérico padrão é **double** (ponto flutuante). Você também pode ter **inteiros** (sufixo L) e **complexos**.

```
typeof(1)      # "double"
```

```
[1] "double"
```

```
typeof(1L)     # "integer"
```

```
[1] "integer"
```

```
typeof(1+2i)   # "complex"
```

```
[1] "complex"
```

- Valores especiais:

```
1/0 # Inf
```

```
[1] Inf
```

```
-1/0 # -Inf
```

```
[1] -Inf
```

```
0/0 # NaN
```

```
[1] NaN
```

```
is.finite(c(Inf, 3.14)) # checa finitude
```

```
[1] FALSE TRUE
```

#### Dica

Precisão numérica: comparações com `==` podem falhar por arredondamento binário. Prefira `all.equal(x, y)`.

```
# 1) Exemplo clássico
```

```
x <- 0.1 + 0.2
```

```
y <- 0.3
```

```
x == y # pode dar FALSE
```

```
[1] FALSE
```

```
identical(x, y) # compara "bit a bit": quase sempre FALSE aqui
```

```
[1] FALSE
```

```
# Ver a diferença real (use mais dígitos)
```

```
old <- options(digits = 17)
```

```
x; y; x - y # diferença minúscula de arredondamento binário
```

```
[1] 0.30000000000000004
```

```
[1] 0.29999999999999999
```

```
[1] 5.5511151231257827e-17
```

```
options(old)
```

```
# Comparação com tolerância
```

```
all.equal(x, y) # TRUE (dentro da tolerância padrão)
```

```
[1] TRUE
```

```
isTRUE(all.equal(x, y)) # retorna um lógico (TRUE/FALSE)
```

```
[1] TRUE
```

```
# 2) Outros casos que sofrem com ponto flutuante
```

```
sin(pi) # ~ 1.224646799e-16 (não é 0 exato)
```

```
[1] 1.224647e-16
```

```
sin(pi) == 0 # FALSE
```

```
[1] FALSE
```

```
all.equal(sin(pi), 0) # TRUE
```

```
[1] TRUE
```

```
# 3) Ajustando a tolerância manualmente (quando precisar ser mais/menos rígido)
```

```
all.equal(x, y, tolerance = 1e-12) # ainda TRUE
```

```
[1] TRUE
```

```
all.equal(x, y, tolerance = 1e-20) # agora acusa diferença
```

```
[1] "Mean relative difference: 1.850372e-16"
```

```
# 4) Estratégia base sem all.equal: comparar o |erro| com um limite  
abs((sqrt(2))^2 - 2) < 1e-12 # TRUE (aceita "igualdade" numérica com folga)
```

```
[1] TRUE
```

## 2) Aritmética

Operadores básicos: + - \* / ^ e, ainda, divisão inteira %/% e módulo %%.

```
14/6
```

```
[1] 2.333333
```

```
14 %/% 6 # quociente inteiro
```

```
[1] 2
```

```
14 %% 6 # resto
```

```
[1] 2
```

```
2^3 # potência
```

```
[1] 8
```

- Precedência: ^ → \* / %/% %% → + -. Use parênteses para deixar claro!

## 3) Atribuição

O padrão recomendado é <-. = também atribui, mas use-o preferencialmente para argumentos de função. -> é atribuição para a direita. <<- afeta o ambiente pai (use com parcimônia).

```
x <- 10  
y = 5 # ok, mas prefira <- fora de chamadas de função  
z <- x + y  
  
z -> resultado # atribuição para a direita (menos comum)  
resultado
```

```
[1] 15
```

```
# Exemplo de <- (evite, pode dificultar depuração)
contador <- local({
  n <- 0
  function() { n <- n + 1; n }
})
contador(); contador(); contador()
```

```
[1] 1
```

```
[1] 2
```

```
[1] 3
```

#### Aviso

**Boas práticas:** Evite <- sempre que possível; prefira retornar valores e trabalhar com escopos explícitos.

#### 4) Vetores (atômicos)

Vetores são coleções unidimensionais do mesmo tipo: lógico, inteiro, double, caractere, complexo ou raw.

```
v <- c(2, 4, 6, 8)
length(v); typeof(v)
```

```
[1] 4
```

```
[1] "double"
```

```
seq(1, 5, by = 2)      # sequência
```

```
[1] 1 3 5
```

```
1:5                    # atalho
```

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

```
rep(3, times = 4)      # repetição
```

```
[1] 3 3 3 3
```

- *Coerção automática:* ao misturar tipos, R promove para um tipo comum.

```
c(1, TRUE, "a") # tudo vira character
```

```
[1] "1"      "TRUE" "a"
```

- Indexação e filtragem:

```
x <- c(10, 20, 30, 40, 50)
x[3]      # posição
```

```
[1] 30
```

```
x[-1]      # tudo, exceto o 1º
```

```
[1] 20 30 40 50
```

```
x[x > 25]   # filtragem lógica
```

```
[1] 30 40 50
```

```
which(x > 25) # posições TRUE
```

```
[1] 3 4 5
```

```
names(x) <- letters[1:5]
x["c"]
```

```
c
30
```

#### 4.1. Vetorização e reciclagem

A maioria das operações é vetorizada (aplica-se elemento a elemento). Quando os comprimentos diferem, R tenta reciclar o menor vetor.

#### 4.2. Nomes × Objetos e copy-on-modify

```
x <- 1:3
y <- x
y[1] <- 99
x # permanece 1 2 3
```

```
[1] 1 2 3
```

```
y # 99 2 3
```

```
[1] 99 2 3
```

- Exceção: ambientes e estruturas por referência (p.ex., R6) não seguem copy-on-modify.

## Exercícios

1. Crie um vetor com os números de 5 a 15 e calcule a média.
2. Use `seq()` para gerar  $0, 0.5, 1, \dots, 5$ .
3. Mostre com código a diferença entre `%/%` e `%%` usando 14 e 6.
4. Atribua `x <- 1:4` e some com o vetor `c(10, 20)`. O que acontece?
5. Verifique o tipo de `1`, `1L` e `1+0i`. Explique a diferença entre tipo (`typeof`) e classe (`class`).
6. Crie e armazene uma sequência de valores de **5 a -11**, progredindo em **passos de 0.3**.
7. Sobrescreva o objeto de (a) com a mesma sequência invertida.
8. Repita o vetor `c(-1, 3, -5, 7, -9)` duas vezes, com cada elemento repetido 10 vezes. Apresente os dados ordenados do maior para o menor.
9. Crie e armazene um vetor contendo, em qualquer ordem:
  - a) inteiros de 6 a 12 (inclusive);
  - b) 5.3 repetido 3 vezes;
  - c) o número -3;
  - d) 9 valores iniciando em 102 e terminando em `length(vc)` (do item (8)).
10. Confirme que o comprimento do vetor criado em d) é 20.



## 3 Motivação

A Estatística, além de lidar com modelos matemáticos rigorosos, também é permeada por situações em que a intuição humana falha de maneira sistemática. Um exemplo clássico é o **problema do aniversário**, que há décadas desperta curiosidade entre estudantes e pesquisadores.

### Enunciado

*Em uma sala com  $r$  pessoas, qual a probabilidade de que pelo menos duas delas compartilhem o mesmo aniversário?*

Um resultado surpreendente é: com apenas **23 pessoas** em uma sala, a probabilidade de que haja pelo menos uma coincidência de aniversários já é **superior a 50%**. Esse resultado é tão interessante que pode ser uma porta de entrada natural para discutir a diferença entre **probabilidade teórica** e **evidência empírica obtida por simulação**.

### Da teoria à simulação

Do ponto de vista teórico, a probabilidade de que todos os aniversários sejam distintos entre  $r$  pessoas é

$$\Pr(\text{todos distintos}) = \prod_{i=1}^{r-1} \frac{365-i}{365} = \left(1 - \frac{1}{365}\right) \left(1 - \frac{2}{365}\right) \cdots \left(1 - \frac{r-1}{365}\right).$$

Logo, a probabilidade de pelo menos uma coincidência é

$$p_r = 1 - \Pr(\text{todos distintos}).$$

Esse produto é conceitualmente claro, mas fica pouco manejável mentalmente para  $r$  moderados. É aqui que a **simulação computacional** pode entrar como aliada didática e científica.

## Um atalho analítico útil

O produto acima admite uma **aproximação exponencial simples e acurada**, obtida tomando logaritmo e usando a expansão para argumentos pequenos:

$$\ln(1-x) = -x + o(x), \quad (x \rightarrow 0).$$

Aplicando ao produto,

$$\begin{aligned} \ln(1-p_r) &= \sum_{i=1}^{r-1} \ln\left(1 - \frac{i}{365}\right) \\ &\approx - \sum_{i=1}^{r-1} \frac{i}{365} = - \frac{1+2+\dots+(r-1)}{365} = - \frac{r(r-1)}{2 \cdot 365}. \end{aligned}$$

Exponentiando e isolando  $p_r$ , obtemos a aproximação

$$p_r \approx 1 - \exp\left\{-\frac{r(r-1)}{730}\right\}.$$

Essa fórmula tem três virtudes didáticas:

- 1) **Clareza**: exhibe explicitamente o papel do número de pares  $\binom{r}{2}$ .
- 2) **Rapidez**: permite cálculos aproximados para valores de  $r$  de interesse.
- 3) **Boas aproximações** já para  $r$  na casa das dezenas.

### Exemplo Rápido

- Para **23** pessoas:

$$p_{23}^{(\text{aprox})} = 1 - \exp\left\{-\frac{23 \cdot 22}{730}\right\} = 1 - \exp\{-0.69315\} \approx 0.500,$$

alinhando-se ao resultado clássico de que **23** pessoas já superam 50% de chance de coincidência.

## O papel da simulação

A simulação estatística permite reproduzir o experimento de forma empírica: sorteamos aleatoriamente dias de aniversário para os indivíduos e verificamos se há repetições. Repetindo o processo milhares de vezes, obtemos uma estimativa para a probabilidade de coincidência.

Por exemplo, em **R**:

```
r <- 23
birthdays <- sample(1:365, r, replace = TRUE)
any(duplicated(birthdays))
```

```
[1] FALSE
```

Ao repetir esse procedimento muitas vezes (por exemplo, 10.000 simulações), podemos estimar a proporção de conjuntos com coincidência. Pela Lei dos Grandes Números, essa estimativa converge para o valor teórico de aproximadamente 0,507 quando  $r = 23$ .

```
set.seed(123) #reprodutibilidade

r <- 23
B <- 10000

acertos <- 0
i <- 0

repeat {
  i <- i + 1
  bdays <- sample(1:365, r, replace = TRUE)
  acertos <- acertos + as.integer(any(duplicated(bdays)))
  if (i >= B) break
}

p_hat <- acertos / B
p_hat
```

```
[1] 0.5073
```

## Atividade: Problema do Aniversário (22 jogadores)

Nesta motivação consideramos um exemplo discutido em Martins (2018) que é o conhecido e amplamente divulgado problema do aniversário (ver, por exemplo, Falk 2014). Martins (2018) segue o exemplo de Matthews e Stones (1998), considerando duas equipes de futebol e, portanto, coincidências de aniversário entre 22 jogadores. Martins (2018) afirma que um resultado positivo importante dessa atividade é a discussão que surgirá naturalmente entre os estudantes, com o professor atuando como mediador. Além disso, os estudantes adoram jogos e a descoberta prática, e a simulação facilita o engajamento nessas atividades, ao mesmo tempo que ilustra resultados que podem ser não intuitivos, bem como teoria geral, como a **Lei dos Grandes Números**.

Agora iremos considerar o seguinte problema:

### Problema

Em uma partida de futebol, qual é a probabilidade de que pelo menos dois dos 22 jogadores façam aniversário no mesmo dia?

Em um país chamado de país do futebol, o contexto é proposital: o futebol é popular e as probabilidades resultantes são contraintuitivas. Antes de qualquer cálculo, considere as hipóteses: (i) todos os 365 dias do ano são igualmente prováveis para qualquer aniversário; (ii) as datas de aniversário dos jogadores são independentes entre si.

### Objetivos

- Estimar, via simulação, a probabilidade de coincidência de aniversários.
- Relacionar frequência relativa, Lei dos Grandes Números e variação amostral.
- Comparar o resultado exato e aproximado.

### Hipóteses

- 365 dias equiprováveis, datas independentes, ignorar bissexto/gêmeos.

### Materiais

- R (ou Posit Cloud), roteiro com comandos `sample()`, `table()`, `mean()`.

## Exercícios

- 1) Determinar o menor número de pessoas que deve estar em uma sala para que se possa apostar, com mais de 50% de chance de ganhar, que entre elas existam pelo menos duas com o mesmo aniversário.
- 2) Determinar o menor número de outras pessoas que deve estar em uma sala com você para que se possa apostar, com mais de 50% de chance de ganhar, que pelo menos uma delas tenha o mesmo aniversário que o seu.

## Referências

- Falk, Ruma. 2014. “A Closer Look at the Notorious Birthday Coincidences”. *Teaching Statistics* 36 (2): 41–46. <https://doi.org/10.1111/test.12014>.
- Hodgson, Ted, e Maurice Burke. 2000. “On Simulation and the Teaching of Statistics”. *Teaching Statistics* 22 (3): 91–96. <https://doi.org/10.1111/1467-9639.00033>.
- Martins, Rui Manuel Da Costa. 2018. “Learning the Principles of Simulation Using the Birthday Problem”. *Teaching Statistics* 40 (3): 108–11. <https://doi.org/10.1111/test.12164>.
- Matthews, Robert, e Fiona Stones. 1998. “Coincidences: the truth is out there”. *Teaching Statistics* 20 (1): 17–19. <https://doi.org/https://doi.org/10.1111/j.1467-9639.1998.tb00752.x>.
- Thomas, F. H., e J. L. Moore. 1980. “CUSUM: Computer Simulation for Statistics Teaching”. *Teaching Statistics* 2 (1): 23–28. <https://doi.org/10.1111/j.1467-9639.1980.tb00374.x>.
- Tintle, Nathan, Beth Chance, George Cobb, Soma Roy, Todd Swanson, e Jill VanderStoep. 2015. “Combating Anti-Statistical Thinking Using Simulation-Based Methods Throughout the Undergraduate Curriculum”. *The American Statistician* 69 (4): 362–70. <https://doi.org/10.1080/00031305.2015.1081619>.
- Zieffler, Andrew, e Joan B. Garfield. 2007. “Studying the Role of Simulation in Developing Students’ Statistical Reasoning”. Em *Proceedings of the 56th Session of the International Statistical Institute (ISI)*. International Statistical Institute.