


# Primeiros passos com R

## Instruções, área de trabalho, documentação e mais

Prof. Dr. **Walmes Zeviani** · UFPR

2021-05-04



# Instruções na linguagem R

O que é código e o que não é?

# Usando o R a primeira vez



A sensação de abrir o R pela primeira vez.

- **CLI** - *Command Line Interface*.
- **REPL** - *Read, Eval, Print and Loop*.

# Instruções na linguagem R

## Modos de uso

- Modo **REPL**
  - Script com instruções (receita).
  - Instruções avaliadas no console (interpretador).
  - Analista superviosiona o processo.
  - Salva/duplica/modifica o script conforme necessidade.
- Modo **Batch**
  - Script executa ser supervisão.
  - Usado em ambientes de produção ou simulação computacional.

## Comentários no código

- Tudo que vem após # é comentário.
- Eles documentam o que o código faz.

```r
# Faz uma soma.
2 + 2
```

```
## [1] 4
```

```r
# Meu índice de massa corporal.
83/1.85^2
```

```
## [1] 24.25128
```

```r
# Quantos segundos tem um dia?
24 * 60 * 60
```

```
## [1] 86400
```

# Instruções e comentários

## Faça comentários relevantes

- Evite comentários óbvios.
- Evite comentários ambiguos.

## Instruções

- Podem ocupar uma única linha.
- Ocupar várias linhas.
- Uma linha ter várias instruções.

## Recomendações

- Evite ultrapassar 72 ou 80 caracteres.
- Mantenha o código devidamente indentado (`ctrol` + `i`).
- Evite muitas instruções em uma linha.

```r
# Uma única linha.
2 + 2 + 7 + 5
```

```
## [1] 16
```

```r
# Em várias linhas.
2 +
    2 +
    7 +
    5
```

```
## [1] 16
```

```r
# Várias em uma linha.
2 + 2; 7 + 5
```

```
## [1] 4
```

```
## [1] 12
```


?
Dúvidas, comentários, sugestões?



# Área de trabalho

Onde estão as coisas que eu crio?

# Área ou espaço de trabalho

## Área de tabalho

- Ao fazer atribuições, são criados objetos na área de trabalho.
- Usa-se `<-` para atribuir algo para um objeto (`alt` + `-`)
- Objetos podem ser reusados para criar outros (essa é a intenção).
- Objetos podem ser sobrescritos.
- Objetos podem ser apagados.

```
peso <- 83      # Meu peso (kg).
altura <- 1.18 # Minha altura (m)
imc <- peso/altura^2
imc
```

```
## [1] 59.60931
```

```
# Lista objetos na área de trabalho.
ls()
```

```
## [1] "altura" "imc"    "peso"
```

```
# Apaga objetos.
rm(peso, altura)
ls()
```

```
## [1] "imc"
```

# Espaços de trabalho

## Onde o R procura por objetos?

- `ls()` mostra o conteúdo do `.GlobalEnv`, a área de trabalho.
- Mas existem objetos em outros ambientes ou espaços.
- Esses são os espaços dos pacotes.
- Quando não encontra no `.GlobalEnv`, ele vai para o próximo espaço de trabalho.
- `search()` retorna a lista de espaços de trabalho.
- Cada pacote tem um seu espaço (*namespace*).

```r
# Mas woman não está no .GlobalEnv.
# De onde veio?
women

# Mostra o .GlobalEnv.
ls()

# Mostra os demais espaços.
search()

# Lista o conteúdo de um espaço.
ls("package:datasets")

# O que acontece se eu fizer?
women <- c("Gertrude Mary Cox",
           "Florence Nightingale David")

women           # Está no .GlobalEnv.
datasets::women # Está no pacote.
```


?
Dúvidas, comentários, sugestões?

# Acesso à documentação do R

Como aprender R sem sair do R

# Documentação interna

## A documentação do R

- Consiste de documentação de objetos e funções.
- Tutoriais chamados de vinhetas (*vignettes*).
- Existem funções específicas para a consulta destes.
- Pode-se procurar na web também.

```r
# Duas formas iguais de chamar
# a documentação.
?woman
help(woman)

# Procura por ocorrêncians de  women.
help.search("woman")

# Objetos que batem com o termo.
apropos("tukey")

# Exibe as vinhetas de um pacote.
browseVignettes(package = "survival")

# Procura pelo termo no
# r-project.org.
RSiteSearch("spider plot")
```

# Campos da documentação

**Cabeçalho**
Indica o pacote.

**Título**
Título da função.

**Description**
Descrição do que o objeto é/faz.

**Usage**
Como usar ou fazer a chamada.

**Arguments**
Quais os argumentos formais da função.

**Value**
O que a função retorna.

**Details**
Detalhes adicionais de implementação.

**Note**
Notas adicionais sobre uso e afins.

**See Also**
Referências para documentação relacionada.

**References**
Referências bibliográficas.

**Authors**
Autores da função.

**Examples**
Exemplos de uso.

# Guia de sobreviência no R

1. ls()
2. apropos()
3. help()
4. help.search()
5. help.start()
6. RSiteSearch()
7. example()
8. demo()
9. browseVignettes()
10. vignette()
11. str()
12. args()
13. class()
14. methods()
15. find()




Kit de sobreviência no R. Aprenda a usá-lo e tudo será mais fácil.



# Arquivos da linguagem R

Customizando a experiência com a linguagem

# Arquivos da linguagem R

Existem arquivos relacionados à linguagem R que servem para melhorar a experiência do usuário.

- `.Rhistory`:
  Arquivo que contém as instruções enviadas pro console.
- `.RData`:
  Arquivo binário que salva sua área de trabalho.
  - Serve para restaurar a área de trabalho.
  - Útil quando o processamento é demorado.
- `.Rproj`:
  Arquivo que define configurações do projeto (para RStudio apenas).
- Eles são salvos no diretório de trabalho.
- Pode ser configurado em RStudio IDE > Tools > Global Options.

- `.Rprofile`:
  Arquivo de configurações lido no início das seções.
  - Carregar pacotes muito usados e configurar opções.
  - Mensagem de boas-vindas e diagnóstico do sistema.
  - Pode ser definido por projeto ou por usuário.

# Diretório de trabalho

- **Diretório de trabalho** é local no sistema operacional para onde o R está apontando.
- Isto é, de onde ele lê e escreve arquivos por padrão.
- Você pode definir por comandos (recomendável) ou usando RStudio IDE > Session > Setting Working Directory.

```
# Mostra o atual diretório de trabalho.
getwd()

# Troca por outro endereço.
setwd("~/Downloads")

# Lista o conteúdo do diretório de trabalho.
dir()
```



# Pacotes

## Instalação e gerenciamento

# Instalação de pacotes

## Pacotes

- Pacotes são coleções de funções e conjuntos de dados organizados e documentados.
- O pacote contém código R e eventualmente códigos de outras linguagens.
- Pacotes podem depender de *libs* do sistema operacional.

## Formas de instalação

- Pacotes podem ser instalados de repositórios: CRAN, Bioconductor, MRAN, etc.
- De aquivos de instalação `*.tar.gz`.
- De repositórios Git: GitHub, GitLab, etc.
- Para mais, visite r-packages-guide.

```r
# Para instalar um pacote do repositório.
install.package("tidyverse")

# Para carregar o pacote e usá-lo.
library(tidyverse)

# Para ver o conteúdo dele.
ls("package:tidyverse")

# Documentação do pacote.
help(package = "tidyverse")

# Para ver onde foi instalado.
system.file(package = "tidyverse")

# Os caminhos para endereços de instalação.
.libPaths()

# Para remover o pacote da sessão.
detach("package:tidyverse", unload = TRUE)

# Funções relacionadas a pacotes.
apropos("package")
```


?
Dúvidas, comentários, sugestões?