

Laboratório de Biologia Computacional e Molecular

Centro de Biotecnologia da UFRGS  
Universidade Federal do Rio Grande do Sul



# **R para Ciências da Vida (BCM13065) Aula 3**

**PPGBCM - UFRGS**

**Diego Bonatto  
2024/2**

# Boas práticas em programação - R

- O princípio das boas práticas em programação R é permitir a reprodutibilidade e a compreensão dos códigos gerados para o usuário

# Boas práticas em programação - R

- **Exemplo de um código ruim:**

```
x←1
```

```
y←1
```

```
z←x+y
```

```
print(z)
```

- **Exemplo de um código bom:**

```
#Calcula a soma de duas  
variáveis:
```

```
x←1
```

```
y←1
```

```
soma←x+y
```

```
print(soma)
```

# Organização e Estrutura do Código

- Estruturar o código em seções usando comentários claros.
- Dividir funções complexas em funções menores.
- Utilizar scripts de configuração e scripts principais.

```
# ---- Carregar pacotes ----  
library(dplyr)  
library(ggplot2)  
  
# ---- Importar dados ----  
dados <- read.csv("meus_dados.csv")  
  
# ---- Pré-processamento ----  
dados_limpos <- dados %>% filter(!is.na(var1))  
%>% mutate(nova_var = var1 / var2)
```

# Boas Práticas de Nomeação

- Nomear variáveis e funções de forma descritiva.
- Adotar uma convenção de nomenclatura consistente.
- Usar nomes claros para variáveis temporárias.

```
media_temperatura <-  
mean(dados$temperatura)
```

```
calc_desvio_padrao <-  
function(valores) {  
  sd(valores)  
}
```

```
desvio_padrao_temperatura  
<-  
calc_desvio_padrao(dados$  
temperatura)
```

# Documentação e Comentários

- Comente o código adequadamente.
- Utilize documentação para funções com 'roxygen2'.
- Explique cada etapa do pipeline de análise.

```
# Função para calcular a média e
desvio padrão
# Args:
#   valores: vetor numérico com os
dados
# Returns:
#   Lista com média e desvio padrão
dos valores
calcular_estatisticas <-
function(valores) {
  media <- mean(valores, na.rm =
TRUE)
  desvio <- sd(valores, na.rm = TRUE)
  list(media = media, desvio_padrao =
desvio)
}
```

# Manipulação de Dados Eficiente

- Utilize vetorização e funções de manipulação de dados otimizadas.
- Evite loops desnecessários substituindo-os por operações vetorizadas.

```
# Usando dplyr para  
operações eficientes em  
vez de loops
```

```
resultado <- dados %>%  
  group_by(categoria) %>%  
  summarize(media =  
    mean(valor, na.rm =  
      TRUE) )
```



# Consistência e Boas Práticas de Formatação

- Mantenha um estilo de indentação consistente.
- Use linhas em branco para separar blocos de código.

```
# Exemplo de consistência de indentação e
separação de blocos
calcular_valores <- function(valores) {
  media <- mean(valores, na.rm = TRUE)
  desvio <- sd(valores, na.rm = TRUE)

  return(list(media = media, desvio =
desvio))
}
```

# Controle de Versão e Gerenciamento de Pacotes

- Utilize um controle de versão, como Git.
- Centralize instalação e carregamento de pacotes no início do script.

```
# Pacotes carregados no início do  
script  
library(dplyr)  
library(ggplot2)
```

```
# Sugestão: registrar pacotes em  
um arquivo DESCRIPTION ou no renv  
para rastrear versões
```

# Depuração e Testes

- Adicione mensagens de erro e avisos personalizados.
- Utilize ``testthat`` para testes automatizados de funções.

```
# Função com mensagem de erro
calcular_raiz <- function(numero) {
  if (numero < 0) {
    stop("Número não pode ser
negativo.")
  }
  sqrt(numero)
}
```

```
# Testando a função com `testthat`
library(testthat)
test_that("calcular_raiz funciona
corretamente", {
  expect_equal(calcular_raiz(4), 2)
  expect_error(calcular_raiz(-1))
})
```

# Documentação de Sessões e Versionamento

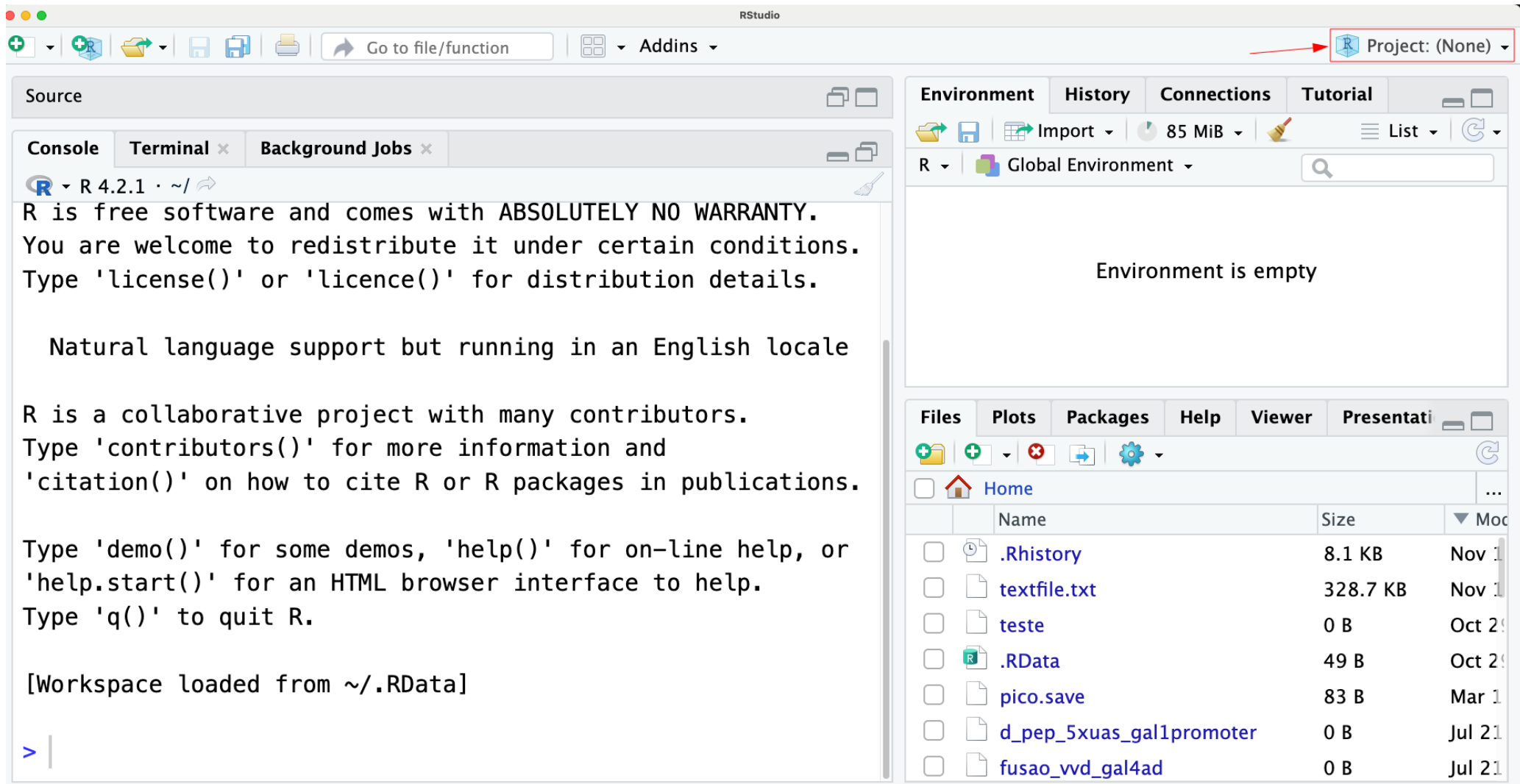
- Documente a sessão do R com `'sessionInfo()'`.
- Mantenha versões dos scripts e dados processados para rastreamento.

```
# Ao final do script, registrar as informações da sessão  
sessionInfo()
```

# Uso de Notebooks para Relatórios

- Considere utilizar RMarkdown ou Quarto para relatar análises.
- Notebooks ajudam a manter código e resultados juntos.

```
---  
title: "Análise de Dados"  
output: html_document  
---  
  
## Carregar os dados  
```${r}```  
dados <- read.csv("meus_dados.csv")  
summary(dados)
```



RCV - RStudio

Go to file/function Addins

R Script 1

Quarto Document... 2

Quarto Presentation...

R Notebook 3

R Markdown... 4

Shiny Web App... 5

Plumber API...

Text File 6

C++ File

Python Script

SQL Script

Stan File

D3 Script

R Sweave

R HTML

R Documentation...

Background Jobs x

RCV/

2-06-23) -- "Funny-Looking Kid"

The R Foundation for Statistical Computing

le-darwin17.0 (64-bit)

and comes with ABSOLUTELY NO WARRANTY.

redistribute it under certain conditions.

'licence()' for distribution details.

upport but running in an English locale

project with many contributors.

' for more information and

to cite R or R packages in publications.

ome demos, 'help()' for on-line help, or

'help.start()' for an HTML browser interface to help.

Type 'q()' to quit R.

> |

Environment History Connections Tutorial

Import Dataset 25 MiB

R Global Environment

Environment is empty

Files Plots Packages Help Viewer Presentation

Folder Blank File Delete Rename

Home > Downloads > RCV

	Name	Size	Modified
	..		
	RCV.Rproj	205 B	Nov 12, 2024, 11:18 AM