

Aula 01 – Introdução ao R e RStudio - Curso R para iniciantes

Adhemar Ranciaro Neto

Bem-vindo ao Curso de R!

Neste primeiro encontro, vamos conhecer o R e o RStudio, aprender a usar o console e os scripts, entender operações matemáticas, variáveis e comandos básicos.

O que é o R?

R é uma linguagem de programação livre voltada para estatística, ciência de dados e gráficos.

O que é o RStudio?

RStudio é um ambiente que facilita escrever, salvar e executar códigos R. É uma “tela amigável” para trabalhar com dados.

Operações Matemáticas no R

```
# Adição
2 + 3

# Subtração
10 - 4

# Multiplicação
6 * 7

# Divisão
20 / 5

# Potência
2^3

# Raiz quadrada
sqrt(25)

# Módulo (resto da divisão)
10 %% 3

# Parte inteira da divisão
10 %/% 3
```

Criando Objetos (Variáveis)

```
# Atribuição com "<-"
a <- 10
b <- 5
soma <- a + b
print(soma)
```

Também funciona com =:

```
x = 100
```

Verificando o tipo e classe dos objetos

```
class(soma)      # tipo de estrutura (numeric, character, etc)
typeof(soma)     # tipo interno do R
```

Lista de Objetos e Remoção de Variáveis

Quando criamos variáveis (ou “objetos”) no R, elas ficam armazenadas na memória. Podemos listar, remover ou limpar tudo, dependendo da necessidade.

Listando os objetos criados

```
ls()             # Lista todos os objetos (variáveis) que estão carregados no ambiente
```

Removendo um objeto específico

```
rm(b)            # Remove o objeto 'b'
```

Removendo vários objetos ao mesmo tempo

```
rm(a, soma)      # Remove os objetos 'a' e 'soma'
```

Limpendo todos os objetos do ambiente

```
rm(list = ls())   # Remove tudo do ambiente
```

Essa última linha é útil para começar um script “do zero”, com o ambiente limpo. Use com cuidado: ela apaga **todas as variáveis** que você criou.

Acessando ajuda

```
help(mean)        # abre a documentação da função mean()
?mean             # atalho

help.search("regression") #Se você não souber o nome exato da função
??regression
```

```
args(mean) #Ver a estrutura da função  
help(package = "ggplot2") #Ajuda com pacotes específicos
```

<https://www.rdocumentation.org> Documentação dos comandos

Operadores relacionais e lógicos

```
# Relacionais  
5 > 3  
4 == 4  
2 != 3  
  
# Lógicos  
TRUE & FALSE  
TRUE | FALSE  
!TRUE
```

Comentários no código

```
# Isto é um comentário. O R ignora essa linha.  
salario <- 3000 # atribuição de valor
```

Mini Desafios

1. Calcule o lucro de uma empresa com receita de 25.000 e custo de 17.500.
2. Crie um objeto com o nome da empresa.
3. Use `print()` para exibir o nome e o lucro.

```
empresa <- "Delta Ltda"  
receita <- 25000  
custo <- 17500  
lucro <- receita - custo  
  
print(empresa)  
print(lucro)
```

Dica: use scripts!

- Salve seus comandos em arquivos `.R` ou `.Rmd` para reutilizá-los.
 - Use a tecla **Ctrl + Enter** para executar linha por linha.
-

Resumo da Aula

- Conhecemos o R e o RStudio
- Aprendemos operações básicas e como criar variáveis

- Vimos comandos úteis como `print()`, `class()`, `ls()`, `help()`
 - Aprendemos a usar comentários e a estrutura de um script
-

Tarefa para casa

1. Crie um script com os dados fictícios de uma pequena empresa (nome, receita, despesa).
2. Calcule o lucro líquido.
3. Use `print()` para mostrar os resultados com uma frase explicativa.

Exemplo:

```
print(paste("A empresa", empresa, "teve um lucro de R$", lucro))
```

Boa prática!

Pacotes no R: o que são e como usar

O R vem com várias funções prontas, mas você pode **aumentar suas capacidades** instalando **pacotes** (ou “bibliotecas”).

O que é um pacote?

Um **pacote** é um conjunto de funções, dados e documentação que adiciona novos recursos ao R.

Exemplos: - `ggplot2` → gráficos - `dplyr` → manipulação de dados - `readxl` → leitura de arquivos Excel

Instalando um pacote

Você só precisa instalar **uma vez** (por computador ou usuário):

```
install.packages("nome_do_pacote")
```

Exemplo:

```
install.packages("ggplot2")
```

Carregando um pacote

Você precisa carregar o pacote **sempre que iniciar um novo script ou sessão**:

```
library(ggplot2)
```

Atualizando pacotes

Para manter os pacotes atualizados:

```
update.packages()
```

Você também pode usar:

```
install.packages("ggplot2") # novamente, para instalar a versão mais recente
```

Dica importante

Se você tentar usar uma função sem carregar o pacote, o R dará erro:

```
Error in ggplot(...) : could not find function "ggplot"
```