

Introdução à Linguagem R

Descubra o poder da linguagem R para análise estatística e ciência de dados



Eduardo Ogasawara

eduardo.ogasawara@cefet-rj.br

<https://eic.cefet-rj.br/~eogasawara>

CONCEITOS FUNDAMENTAIS

O que é Programação?

Programação é o processo de **escrever instruções** que um computador pode executar. Essas instruções devem ser precisas, não ambíguas e executáveis passo a passo.

Precisão

Cada instrução deve ser exata e específica, sem margem para interpretação

Clareza

As instruções não podem ser ambíguas - devem ter apenas um significado

Execução

Um programa descreve como transformar dados de entrada em resultados

O que é um Algoritmo?

Um algoritmo é uma **sequência finita de passos** bem definidos que resolvem um problema específico. Cada passo executa uma ação simples e clara.

Características de um algoritmo

- Recebe dados de entrada
- Processa esses dados de forma sistemática
- Produz uma saída ou resultado



Linguagens de Programação e o R

Uma linguagem de programação define como escrever instruções e como elas são interpretadas pelo computador.

O R é especial porque:

- É uma linguagem **interpretada** - executa instruções diretamente
- É voltada para **análise de dados** e estatística
- É baseada em **operações vetorizadas**

Observação importante: No R, muitas operações atuam sobre **conjuntos de valores**, não apenas valores individuais.
Isso torna o R especialmente poderoso para análise de dados.

O que é a Linguagem R?



Software Livre

Linguagem de programação e ambiente gratuito para computação estatística, mantido pela R Foundation

Origem Acadêmica

Criado por Ross Ihaka e Robert Gentleman na Universidade de Auckland, Nova Zelândia

Amplamente Utilizado

Ferramenta preferida de estatísticos, mineradores de dados e cientistas de dados em todo o mundo

R foi derivado da linguagem S (Bell Laboratories - AT&T). A versão atual é a 4.4. Saiba mais em <https://www.r-project.org/>

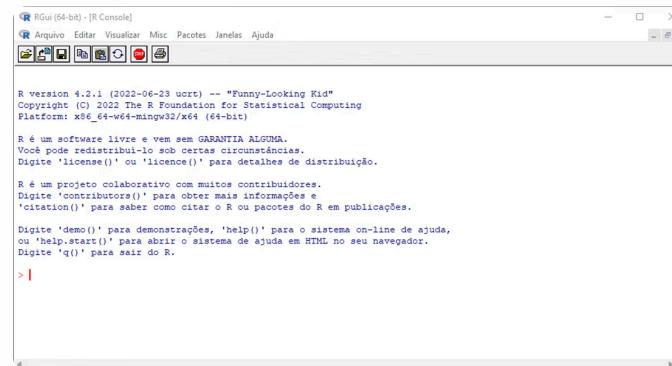
Console R: Sua Primeira Interface

Multiplataforma

O R está disponível para Windows, Mac e Linux, garantindo flexibilidade em qualquer ambiente de desenvolvimento

Interface Simples

O console R oferece uma interface direta para executar comandos e visualizar resultados imediatamente



Console R: Operações Básicas

Aritmética Simples

```
1 + 2  
## [1] 3  
  
1:10  
## [1] 1 2 3 4 5 6 7 8 9 10
```

Criando Variáveis

```
a <- 1  
a  
## [1] 1  
  
a + 2  
## [1] 3
```

Trabalhando com Vetores

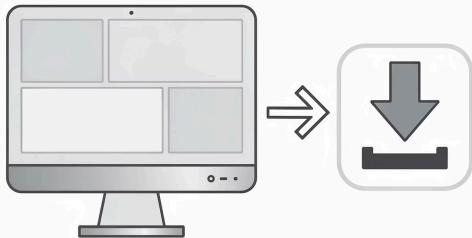
```
dado <- 1:6  
dado  
## [1] 1 2 3 4 5 6  
  
dado[1]  
## [1] 1  
  
dado[6]  
## [1] 6  
  
dado - 1  
## [1] 0 1 2 3 4 5
```

Instalando o R

01

Acesse o Site Oficial

Visite <https://www.r-project.org> para baixar a versão mais recente do R



02

Escolha Seu Sistema

Selecione a versão compatível com seu sistema operacional (Windows, Mac ou Linux)

03

Execute a Instalação

Siga o assistente de instalação e aceite as configurações padrão recomendadas

04

Teste os Exemplos

Após instalar, abra o console R e execute os exemplos das operações básicas para verificar

RStudio: O Ambiente Integrado



IDE Completo

Interface de desenvolvimento integrada com editor de código, console e visualizador de dados



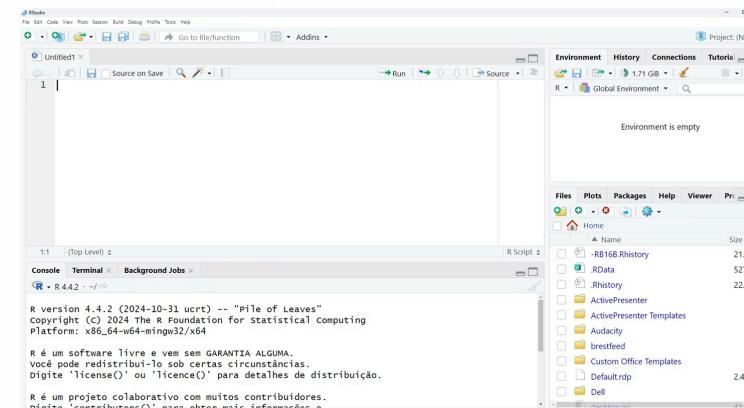
Ferramentas Avançadas

Recursos de depuração e criação de perfil para otimizar seu código



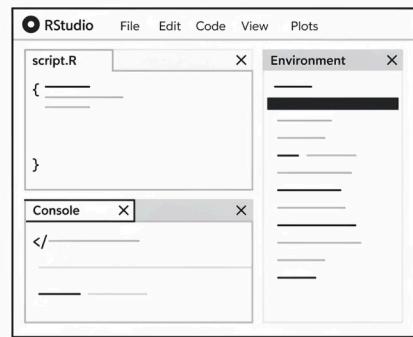
Execução Flexível

Execute código linha por linha, em blocos ou completamente



Disponível gratuitamente em <http://www.rstudio.com>

Instalando o RStudio



- 1
- 2
- 3

Download do RStudio

Acesse <https://www.rstudio.com> e baixe a versão gratuita do RStudio Desktop

Execute os Exemplos

Teste os exemplos anteriores no ambiente RStudio e explore a interface intuitiva

Código de Exemplo

Todo o código está disponível no repositório: <https://github.com/eogasawara/R>

Operações Aritméticas em R



Soma e Subtração

```
3 + 2  
## [1] 5
```



Multiplicação e Divisão

```
3 / 2  
## [1] 1.5
```



Potência

```
3 ^ 2  
## [1] 9
```



Resto da Divisão

```
3 %% 2  
## [1] 1
```

R suporta todas as operações aritméticas básicas necessárias para cálculos estatísticos e científicos

Operações com Vetores

Escalar com Vetor

Multiplique um número único por todos os elementos de um vetor

```
dado <- 1:6  
2 * dado  
## [1] 2 4 6 8 10 12
```

Vetor com Vetor

Realize operações elemento por elemento entre dois vetores

```
dado * dado  
## [1] 1 4 9 16 25 36
```

As operações vetoriais em R são eficientes e intuitivas, aplicando-se automaticamente a cada elemento

Trabalhando com Funções

Funções para Valores Atômicos

```
round(3.1415)
```

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

```
round(3.1415, 2)
```

```
## [1] 3.14
```

```
#?round
```

Use **?** antes do nome da função para acessar a ajuda

Funções sobre Vetores

```
mean(dado)
```

```
## [1] 3.5
```

```
round(mean(dado))
```

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

Funções podem ser **encadeadas** para criar operações complexas

Argumentos de Funções

Parâmetros Explícitos

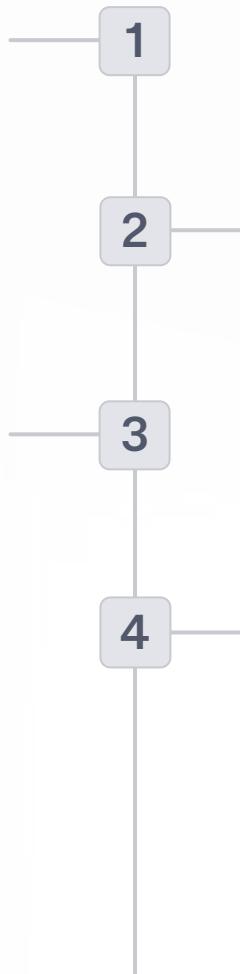
```
sample(x = dado, size = 2)  
## [1] 6 2
```

Especifique claramente cada parâmetro pelo nome

Argumentos Opcionais

```
sample(x=dado, size=2, replace=TRUE)  
## [1] 1 5
```

Adicione parâmetros opcionais conforme necessário



Ordem dos Parâmetros

```
sample(dado, 2)  
## [1] 3 6
```

Use a ordem padrão dos argumentos sem nomeá-los

Descobrindo Argumentos

```
#?sample  
args(sample)  
## function (x, size, replace = FALSE, prob = NULL)
```

Use **args()** para ver todos os argumentos disponíveis

Flexibilidade na Ordem dos Parâmetros

Parâmetros Nomeados

```
set.seed(1)  
sample(x=dado, size=2,  
replace=TRUE)  
## [1] 1 4
```

Quando você nomeia os parâmetros explicitamente, pode colocá-los em qualquer ordem

Ordem Alternativa

```
sample(replace=TRUE,  
x=dado, size=2)  
## [1] 1 2
```

A ordem não importa quando os argumentos são nomeados

Ordem Implícita

```
sample(dado, 2, TRUE)  
## [1] 5 3
```

Sem nomes, os argumentos devem seguir a ordem padrão da função



BOAS PRÁTICAS

Scripts, Reprodutibilidade e Boas Práticas

A reproduzibilidade é fundamental na programação científica e profissional. Seguir boas práticas desde o início garante que seu código seja confiável, compartilhável e fácil de manter ao longo do tempo.

01

Save em Scripts

Todo código deve ser salvo em arquivos .R para documentação e reutilização

02

Execução Completa

O script deve rodar do início ao fim sem erros ou dependências externas

03

Ambiente Limpo

Limpar o ambiente regularmente evita dependências ocultas e bugs sutis

04

Execução do Topo

Sempre execute o script completo desde o início, nunca dependa do console

"Código reproduzível é código confiável. Evite depender do histórico do console — cada execução deve ser independente e completa."



ARMADILHAS COMUNS

Erros Comuns de Iniciantes



Confusão entre Nome e Valor

Misturar o nome da variável com o valor que ela contém

Valores Mutáveis

Esquecer que o valor de uma variável pode mudar durante a execução

Ordem de Execução

Executar linhas fora de ordem ou não considerar a sequência

Intenção vs. Execução

Supor que o computador "entende a intenção" por trás do código

Mensagem-chave: O computador executa exatamente o que foi escrito, não o que foi imaginado.

Referências



Hands-On
Programming
with R

WRITE YOUR OWN FUNCTIONS AND SIMULATIONS

Hands-on Programming

Aprenda R criando suas
próprias funções e simulações

[https://rstudio-
education.github.io/hopr/index.html](https://rstudio-education.github.io/hopr/index.html)



R Graphics
Cookbook

R Graphics Cookbook

Domine visualizações de dados
em R

<https://r-graphics.org>



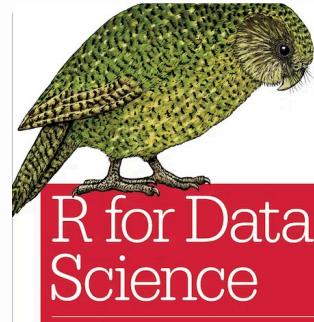
R Packages

ORGANIZE, TEST, DOCUMENT AND SHARE YOUR CODE

R Packages

Desenvolva seus próprios
pacotes R

<https://r-pkgs.org/index.html>



R for Data
Science

VISUALIZE, MODEL, TRANSFORM, TIDY, AND IMPORT DATA

R for Data Science

Guia completo para ciência de
dados

<https://r4ds.had.co.nz>