

Pacotes e Ajuda no R

Um guia prático sobre instalação de pacotes, uso de funções e recursos de ajuda na linguagem R

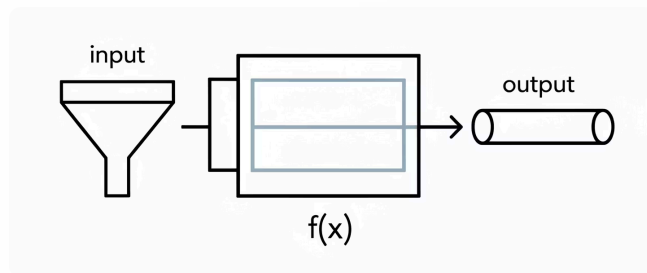


Eduardo Ogasawara

eduardo.ogasawara@cefet-rj.br

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

O que é uma Função?



Uma função é um **bloco de código reutilizável** que encapsula uma tarefa específica.

Estrutura de uma função

- Recebe **entradas** (chamadas de argumentos)
- Executa um **processamento** interno
- Produz uma **saída** (retorno)

01

Organização

Mantém o código estruturado e legível

02

Reutilização

Evita repetição de código

03

Manutenção

Facilita correções e melhorias

Ideia central: Uma função encapsula uma tarefa para ser usada sempre que necessário.

Chamando Funções e Argumentos

Para usar uma função, precisamos informar o **nome da função** e os **argumentos** entre parênteses.

Argumentos Posicionais

```
mean(c(1, 2, 3, 4))
```

Os valores são passados na ordem esperada pela função

Argumentos Nomeados

```
mean(  
  x = c(1, 2, 3, 4),  
  na.rm = TRUE  
)
```

Os nomes dos parâmetros são explicitados



Entrada

Dados passados para a função




Processamento

Função executa seu algoritmo



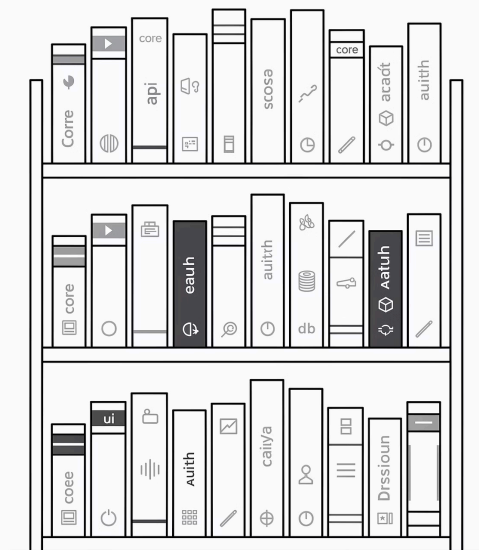
Retorno

Resultado pode ser usado ou armazenado

 **Observação:** O valor retornado pode ser usado diretamente em expressões ou armazenado em uma variável para uso posterior.

INTRODUÇÃO

O Ecossistema de Pacotes do R



Repositório CRAN

Mais de 22.000 pacotes com controle de qualidade rigoroso e documentação completa


GitHub

Versões em desenvolvimento e experimentais disponíveis para testes

Comunidade Ativa

Pesquisadores, professores, programadores e estatísticos contribuindo constantemente

Os pacotes são coleções organizadas de funções que expandem as capacidades do R base. Eles são publicados principalmente no repositório CRAN (Comprehensive R Archive Network), garantindo qualidade e confiabilidade.

 **DAL Packages:** Explore os 7 pacotes desenvolvidos pelo DAL em [daltoolbox](#), [harbinger](#) e [tspredict](#)

Instalação de Pacotes



Instalar o Pacote

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

Use esta função apenas uma vez para baixar o pacote do CRAN



Carregar na Sessão

```
library(ggplot2)
```

Carregue o pacote sempre que iniciar uma nova sessão do R



Usar as Funções

Agora todas as funções do pacote estão disponíveis para uso

📌 **Dica importante:** Deixe o comando `install.packages()` comentado após a primeira instalação para evitar downloads desnecessários em execuções futuras.

Usando Funções do ggplot2

Código

```
library(ggplot2)

# Criando vetores
x <- c(-1, -0.8, -0.6, -0.4,
      -0.2, 0, 0.2, 0.4,
      0.6, 0.8, 1)

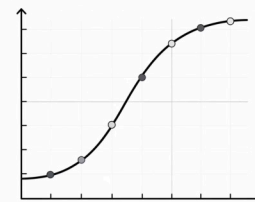
y <- x^3

# Visualizando
qplot(x, y)
```

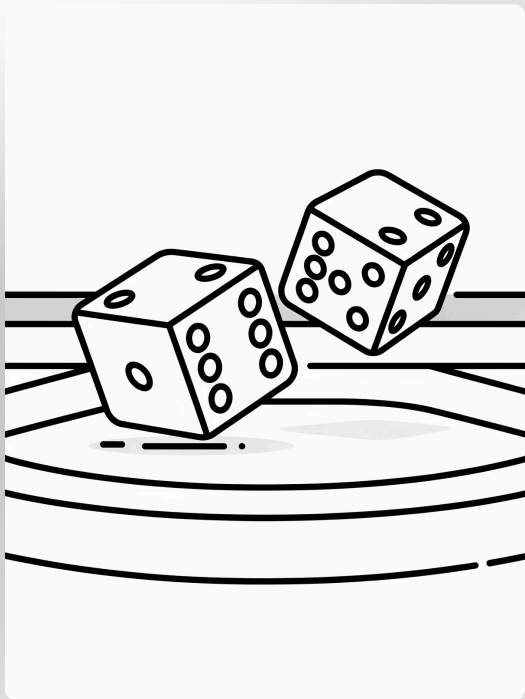
Resultado

A função `qplot()` (quick plot) cria rapidamente gráficos de dispersão, permitindo visualizar a relação entre as variáveis x e y .

Este exemplo demonstra uma função cúbica, onde y é igual a x elevado ao cubo.



Chamadas com parâmetros explícitos não precisam seguir a ordem dos argumentos, tornando o código mais legível e flexível.



Prototipando uma Função

CONCEITOS BÁSICOS

Definir as Operações

```
dados <- 1:6  
dados <- sample(dados, size = 2,  
               replace = TRUE)  
sum(dados)
```

Primeiro, testamos a lógica passo a passo diretamente no console

Encapsular em Função

Após validar a lógica, transforme o código em uma função reutilizável

Avaliar o Código

Execute cada linha separadamente para verificar se o comportamento está correto

Este processo de prototipagem permite testar e refinar a lógica antes de criar a função final. O exemplo simula o lançamento de dois dados e calcula a soma dos valores.

Criando a Função Jogada

Estrutura da Função

```
jogada <- function() {  
  dado <- 1:6  
  dados <- sample(dado,  
                  size = 2,  
                  replace = TRUE)  
  return(sum(dados))  
}
```

Componentes

- **function():** Define uma nova função sem parâmetros
- **Corpo:** Contém a lógica de lançamento dos dados
- **return():** Retorna a soma dos valores obtidos



Limpeza de ambiente: Clique no ícone de vassoura no painel Environment do RStudio para limpar as variáveis temporárias e manter seu workspace organizado.

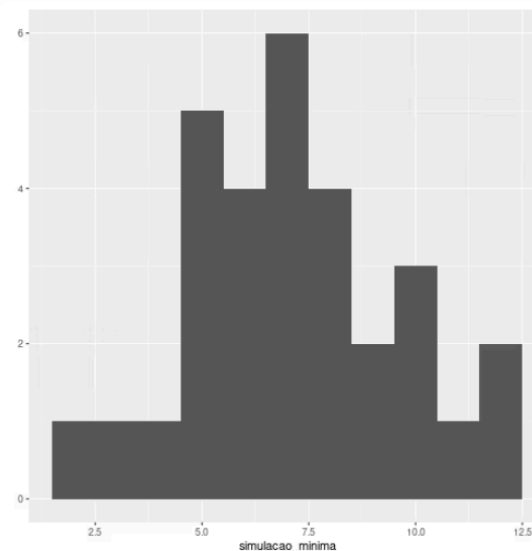
Simulação Mínima com 30 Amostras

```
simulacao_minima <- replicate(30,  
                               jogada())
```

```
qplot(simulacao_minima,  
       binwidth = 1)
```

A função `replicate()` executa a função `jogada()` 30 vezes, criando um vetor com os resultados.

O parâmetro `binwidth = 1` define a largura das barras do histograma.



O número mínimo de 30 amostras é considerado adequado para calcular uma média estatisticamente válida, mas pode não ser suficiente para visualizar claramente a distribuição de probabilidades.

Espaço de Possibilidades

Soma	Combinações (D1, D2)
2	(1,1)
3	(1,2), (2,1)
4	(1,3), (2,2), (3,1)
5	(1,4), (2,3), (3,2), (4,1)
6	(1,5), (2,4), (3,3), (4,2), (5,1)
7	(1,6), (2,5), (3,4), (4,3), (5,2), (6,1)
8	(2,6), (3,5), (4,4), (5,3), (6,2)
9	(3,6), (4,5), (5,4), (6,3)
10	(4,6), (5,5), (6,4)
11	(5,6), (6,5)
12	(6,6)

Esta tabela mostra todas as 36 combinações possíveis ao lançar dois dados. Note que o valor 7 possui 6 combinações possíveis (1/6 de probabilidade), enquanto os valores 2 e 12 possuem apenas uma combinação cada (1/36 de probabilidade).

Simulação Completa com 10.000 Amostras



Escala Massiva

10.000 repetições para garantir precisão estatística



Distribuição Clara

A forma triangular emerge claramente com mais dados

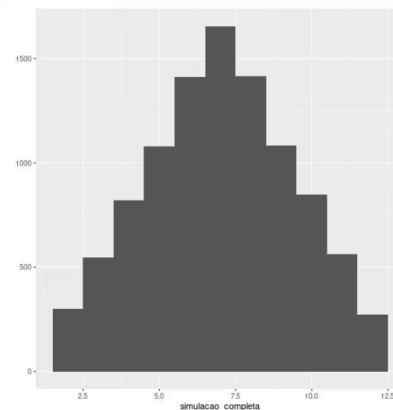


Convergência

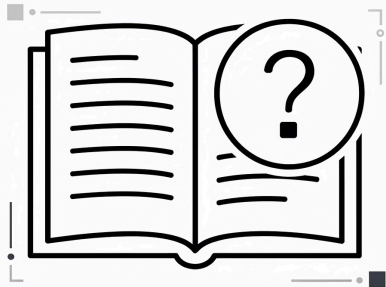
Os resultados se aproximam das probabilidades teóricas

```
simulacao_completa <-  
  replicate(10000, jogada())
```

```
qplot(simulacao_completa,  
      binwidth = 1)
```



Com uma simulação em larga escala, a distribuição de probabilidades se torna evidente, mostrando claramente que o valor 7 é o mais provável, seguido por 6 e 8, e assim por diante.



Sistema de Ajuda do R

RECURSOS INTEGRADOS

01

Calcular a Média

```
mean(simulacao_completa)
```

Resultado: 7.0064 (muito próximo do valor esperado de 7)

02

Acessar a Ajuda

```
?sqrt
```

O operador `?` abre a documentação da função especificada

03

Explorar Exemplos

A documentação inclui descrição, parâmetros, valores de retorno e exemplos práticos



Outras formas de ajuda: Use `help(funcao)`, `??termo` para busca ampla, ou `example(funcao)` para executar os exemplos diretamente.

Aplicando Exemplos da Documentação

Código da Ajuda

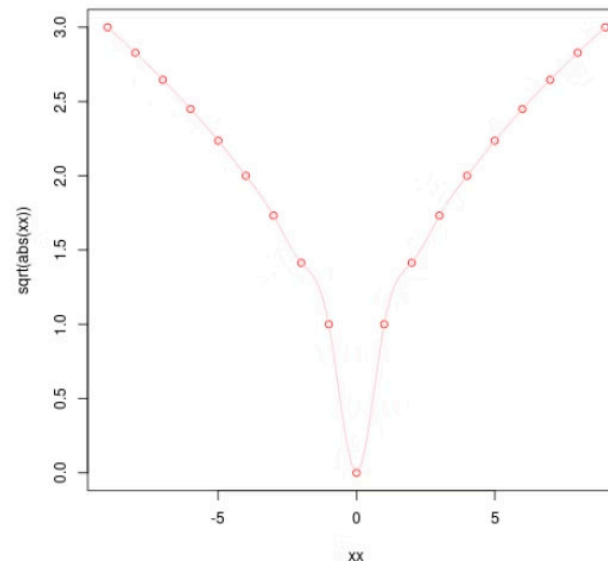
```
require(stats)
require(graphics)

xx <- -9:9

plot(xx, sqrt(abs(xx)),
     col = "red")

lines(spline(xx,
             sqrt(abs(xx)),
             n=101),
      col = "pink")
```

Visualização



A documentação do R inclui exemplos executáveis que demonstram o uso prático das funções. Este exemplo mostra como plotar a raiz quadrada de valores absolutos com uma linha suavizada usando spline.

Explore sempre os exemplos da documentação - eles são uma excelente forma de aprender novas técnicas e funcionalidades!

Amostragem com Distribuição de Probabilidades

```
simulacao = replicate(10000,  
  sample(2:12,  
    replace=TRUE,  
    prob=c(1,2,3,4,5,6,5,4,3,2,1)/36))  
  
qplot(simulacao, binwidth=1)
```

1

Definir Probabilidades

O vetor `c(1,2,3,4,5,6,5,4,3,2,1)/36` representa as probabilidades exatas de cada soma

2

Aplicar na Amostragem

O parâmetro `prob` pesa cada valor de acordo com sua probabilidade real

3

Resultados Precisos

Esta abordagem gera resultados que refletem perfeitamente a distribuição teórica

Ao usar a distribuição de probabilidade exata, eliminamos a variação aleatória e obtemos resultados que convergem mais rapidamente para os valores esperados.

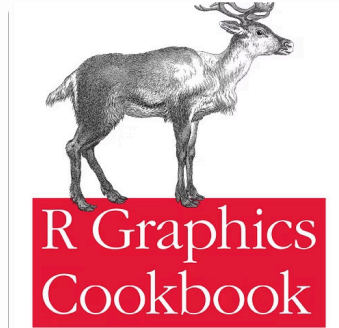
Referências



Hands-on Programming

Aprenda R criando suas próprias funções e simulações

<https://rstudio-education.github.io/hopr/index.html>



R Graphics Cookbook

Domine visualizações de dados em R

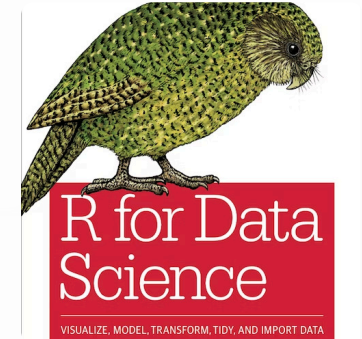
<https://r-graphics.org>



R Packages

Desenvolva seus próprios pacotes R

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



R for Data Science

Guia completo para ciência de dados

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