

Aritmética básica com R

Operações matemáticas e lógicas

Prof. Dr. **Walmes Zeviani** · UFPR

2021-05-04



Operações matemáticas e lógicas

R como uma calculadora científica

Operações matemáticas

As operações básicas

```
2 + 3     # Soma.
2 - 3     # Subtração.
2 * 3     # Multiplicação.
2/3       # Divisão.
2^3       # Potenciação.
2^(1/3)   # Radiciação.
10 %% 3   # Resto.
10 %/% 3  # Parte inteira.
```

Logarítmo

```
exp(2)             # Exponencial neperiano.
log(10)            # Neperiano.
log10(10)          # Base 10.
log2(10)           # Base 2.
log(10, base = 5)  # Base qualquer.
```

Trigonométricas

```
sin(3) # Seno.
cos(3) # Cosseno.
tan(3) # Tangente.

asin(0.5) # Arco seno.
acos(0.5) # Arco cosseno.
atan(0.5) # Arco tangente.
```

Estatísticas

```
y <- c(1.85, 1.78, 1.81, 1.69, 1.71)
mean(y)   # Média.
median(y) # Mediana.
sd(y)     # Desvio-padrão amostral.
var(y)    # Variância amostral.
max(y)    # Máximo.
min(y)    # Mínimo.
length(y) # Número de valores.
```

Operadores lógicos

Comparações de valor

```r
x <- 3
y <- 4

2 == 2  # Igualdade.
2 != 2  # Desigualdade.
x <= y  # Outros operadores:
        # "<", ">", and ">=".
```

```r
(2 < 5) & (7 >= 3) # Operador AND.
(2 < 5) | (7 >= 3) # Operador OR.
!(2 < 5)           # Operador NOT.
```

```r
"a" == "b" # Compara strings.
"a" < "b"  # Ordem alfanumérica.
"1" < "a"
```

Tipos especiais

- `NA`: para valores ausentes.
- `NULL`: para objetos vazios.
- `Inf` e `-Inf`: para infinitos.
- `NaN`: para resultados não razoavelmente definidos.

```r
5 + NA          # O resultado é NA.
is.na(5 + NA)   # Verifica se é NA.

10 + NULL       # Retorna objeto vazio.
is.null(NULL)   # Verifica se é nulo.

5/0             # Infinito.
is.finite(5/0) # Verifica se é finito.

0/0             # Valor indeterminado.
is.nan(0/0)     # Verifica se é not a number.
```

Cartões de referência do R

R Reference Card

R Reference Card · Statistics

R base

R base · PT

RStudio IDE

RStudio IDE · PT


?
Dúvidas, comentários, sugestões?



Manipulando vetores de dados

Como trabalhar com dados?

Vetores

- O vetor é uma estrutura que armazena vários valores.
- O tipo de valor é **homogêneo**, i.e. todos os elementos do mesmo tipo de valor.

```
v1 <- c(1, 5, 11, 33)       # Numérico.
v2 <- c("hello", "world")   # De caracteres.
v3 <- c(TRUE, TRUE, FALSE) # Lógico.

# ATTENTION: Coerção para o tipo mais geral.
v4 <- c(v1, v2, v3, "boo")
```

Importante

Regra da Reciclagem

- Vetores menores são reclicados até ter tamanho do maior para fazer a operação.

```
v1 <- c(2, 4, 6, 8, 10)
v2 <- c(1, 3)
length(v1)
length(v2)

# Operações entre vetores.
v1 * 10 # Vetor-escalar.
v1/2    # Vetor-escalar.
v1 + v2 # Vetor-vetor.
```

Tipos de valores e classes

Tipos de valor

Os tipos básicos são:

- Lógico: TRUE, FALSE.
- Inteiro: 0, 5, 10, 20, etc.
- Numérico (aka double, float): 3.14, 10.22, etc.
- Caractere (aka string): "Curitiba", "Campinas", etc.
- Fator: "Casado", "Solteiro", etc.
- Complexo: números complexos com parte imaginária.
- Datas: yyyy-mm-aa.
- Data tempo: yyyy-mm-aa hh:mm:ss.

Consulta do tipo de valor

```
# Funções que começam com is.
apropos("^is\\.")
is.integer(1)
is.numeric(1)

is.integer(1L)
is.numeric(1L)

is.character("Curitiba")

y <- factor(c("Solteiro", "Casado"))
is.factor(y)
is.character(y)

is.logical(c(TRUE, FALSE))

# Conversões de tipo.
apropos("^as\\.")
as.numeric("123")
as.logical(c(0, 1))
```

Classe e atributos

- Algo importante no aprendizado de R é dominar a lógica de **classes e métodos**.
- As funções genéricas (métodos) atuam conforme a classe do objeto.
- Os atributos são metadados os objetos usados para várias situações.

```r
# Um vetor numérico.
x <- c(1, 2, 3)
class(x)
methods(class = "numeric")

# Numérico mas com valores nomeados.
notas <- c("João" = 7.8,
           "Bianca" = 10,
           "Eduarda" = 8.5)
class(notas)
attributes(notas)
names(notas)

# Uma tabela.
class(women)
attributes(women)
```

Sequências, repetições e números aleatórios

- Útil para aprender R, saber gerar tipos de objetos e preencher com valores.
- Vamos ver como gerar:
  - Sequências regulares: 1, 2, 3, 4, 5, e assim por diante.
  - Repetições: 1, 1, 1, 2, 2, 2, 3, 3, 3.
  - Amostras aleatórias e números aleatórios.

```r
# Sequências regulares.
1:7
seq(from = 1, to = 10, by = 2)
seq(from = 1, to = 20, length.out = 7)
seq(from = 1, by = 2, length.out = 7)

# Repetições.
rep(0, 5)
rep(1:3, times = 2)
rep(1:3, each = 2)

# Amostras aleatórias.
sample(1:20, size = 10,
       replace = FALSE)
sample(c("a", "b", "c"), size = 10,
       replace = TRUE)

# Números aleatórios.
runif(n = 10, min = 0, max = 1)
rnorm(n = 10, mean = 1.80, sd = 0.1)
```

Seleção de elementos no vetor

Seleção por posição

- A seleção por posição/indexação considera a posição dos elementos.

```
# Numérico mas com valores nomeados.
notas <- c("João" = 7.8,
           "Bianca" = 10,
           "Eduarda" = 8.5,
           "Felipe" = 7.0,
           "Márcia" = 6.5)

notas[1]        # A posição 1.
notas[5]        # A posição 5.
notas[1:2]      # Um intervalo.
notas[c(1, 3)] # Um conjunto.
notas[-1]       # Remove.
```

Seleção condicional

- A seleção condicional considera uma máscara lógica para selecionar.

```
mask <- notas > 7.0
mask
notas[mask]

notas[notas > 9.0]
```

Seleção por nomes

- Usa o nome associado aos elementos.

```
# Seleciona valores pelo nome associado.
notas["João"]
notas[c("Márcia", "Eduarda")]
```

Seleção e modificação

Modificação do conteúdo

- Uma vez que valores podem ser selecionados, o conteúdo pode ser modificado.

```
# Atribui nota para um aluno.
notas["João"] <- 0
notas

# Atribui nota "desconhecida" para aluno.
notas["Felipe"] <- NA
notas

# Remove elemento do vetor.
notas <- notas[-4]
notas
```

Adicionando mais valores

- O vetor pode ser incrementado ou concatenado com outros.

```
append(notas, value = c("Carlos" = 9.0))
append(notas, value = c("Simone" = 7.2),
       after = 0)

novas_notas <- c(notas,
                 c("Pedro" = 8.0,
                   "Luana" = 8.3))
novas_notas
```


?
Dúvidas, comentários, sugestões?