

## Exercícios 1

### 1 Expressões

Uma expressão em **R** corresponde a uma sequência de operações, e que pode incluir chamadas de variáveis e de funções, que poderá ser avaliada, tendo um (e apenas um) retorno.

1. Avalie as seguintes expressões:

- `43 * 3`
- `8 ** 4`
- `8 ^ 4`
- `28 - 2 * 5`
- `(28 - 2) * 5`
- `4 + 2 == 42`
- `5 - sum(1, 4)`

#### Solução:

```
> 43 * 3
[1] 129
# o expoente pode ser obtido com os operadores ** e ^ (acento circunflexo)
> 8 ** 4
[1] 4096
> 8 ^ 4
[1] 4096
# a precedência das operações aritméticas é respeitada
> 28 - 2 * 5
[1] 18
> (28 - 2) * 5
[1] 130
# exemplo com o operador lógico de igualdade
# os operadores lógicos têm menor precedência que os operadores aritméticos
> 4 + 2 == 42
[1] FALSE
# uma expressão pode envolver a chamada de uma função
> 5 - sum(1, 4)
[1] 0
```

#### Notas:

- Para inserir um comentário no código, ou comentar uma instrução, basta colocar um cardinal `#` antes do texto pretendido;
- O **R** respeita a precedência dos operadores aritméticos, bem como de outros operadores próprios, pelo que deve-se ter cuidado com a ordem das operações, e usar parêntesis caso seja necessário.

2. Avalie as seguintes expressões matemáticas em **R**:

- $-2^4$
- $12 + 5 \times \pi$
- $3^{7-4} + \sqrt{2}$
- $\sqrt{(5-2)^2 + (7-3)^2}$
- $\frac{20+18+24}{3}$
- $\sqrt[3]{(20 \times 18 \times 24)}$

**Solução:**

```
# neste caso obteria um resultado errado sem os parêntesis
> (-2)^4
[1] 16
> 12 + 5 * pi
[1] 27.70796
> 3 ^ (7 - 4)
[1] 27
# exemplo com a distância euclidiana entre dois pontos no espaço 2-D
> sqrt((5 - 2) ^ 2 + (7 - 3) ^ 2)
[1] 5
# exemplo de média aritmética
> (20 + 18 + 24) / 3
[1] 20.66667
# exemplo de média geométrica
> (20 * 18 * 24) ^ (1/3)
[1] 20.51971
```

Notas:

- O valor da constante  $\pi$  pode ser acedido pela variável pré-definida **pi**.
- A raiz quadrada pode ser obtida pela função **sqrt()**.
- Para se obter o valor de uma raiz que não a quadrada, pode-se usar a relação  $\sqrt[n]{x} = x^{1/n}$ .

3. A Tabela 1 contém variáveis e expressões. Guarde as expressões da coluna **Expressão** nas variáveis correspondentes, indicadas na coluna **Variável**.

Tabela 1: Variáveis e expressões.

Variável	Expressão
a	1
b	-5
c	6
A	$\pi \times 6371^2$

**Solução:**

```
> (a = 1)
[1] 1
> (b <- -5)
[1] -5
> (6 -> c)
[1] 6
> (A = pi * 6371 ^ 2)
[1] 127516118
```

**Notas:**

- Em **R**, existem três operadores de atribuição: =, <- e ->, pode-se usar qualquer um deles, mas convém ser consistente na sua utilização durante a escrita do código;
- Colocando uma expressão de atribuição entre parêntesis é uma forma rápida de guardar o resultado e visualizá-lo de seguida, sem ter de se escrever o nome da variável.

4. Use as variáveis definidas na questão anterior para avaliar as seguintes expressões:

- `A == a`
- `A = a`
- $\frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$

**Solução:**

```
# O R é sensível às maiúsculas
> A == a
[1] FALSE
# deste modo perde-se o valor que estava guardado em A
> A = a
# e assim, agora, as duas variáveis já guardam o mesmo valor
> A == a
[1] TRUE
# exemplo com a fórmula resolvente
# é necessário calcular em dois passos, escrever -b +- sqrt(...) daria apenas o segundo zero
> (-b + sqrt(b ^ 2 - 4 * a * c)) / (2 * a)
[1] 3
> (-b - sqrt(b ^ 2 - 4 * a * c)) / (2 * a)
[1] 2
```

Guardar o resultado de uma expressão, que pode ser complicada, numa variável auxiliar, pode ser uma forma de evitar repetição de código e de o tornar mais legível.

## 2 Vectores

Vectores constituem uma forma simples, mas poderosa, de armazenar informação em **R**, mas apenas de um só tipo. Uma forma fácil de criar vectores é utilizando a função `c()`.

1. Crie um vector `notas` com as seguintes notas de um aluno: 13, 15, 12, 16, 10.

**Solução:**

```
> (notas = c(13, 15, 12, 16, 10))  
[1] 13 15 12 16 10
```

2. Calcule a média do aluno.

**Solução:**

```
> mean(notas)  
[1] 13.2
```

3. O aluno recebeu uma bonificação na suas notas. Acrescente um valor a cada nota existente no vector `notas`.

**Solução:**

```
> (notas.bonus = notas + 1)  
[1] 14 16 13 17 11
```

4. Imagine, agora, que o valor de bónus pode ser de 1 ou 0,5 valores. Considere que a forma como é decidida a escolha de uma destas hipóteses (1 ou 0,5) não é relevante.

**Solução:**

```
> (notas.bonus2 = notas + c(1, 0.5))  
[1] 14.0 15.5 13.0 16.5 11.0  
Warning message:  
In notas + c(1, 0.5) :  
longer object length is not a multiple of shorter object length
```

Repare que os elementos do vector `c(1, 0.5)` foram repetidos tantas vezes quantas as necessárias, até serem percorridas todos os elementos do vector `notas`. A mensagem de aviso dada pelo **R** informa que os dois vectores, `notas` e `c(1, 0.5)`, não têm comprimentos múltiplos (o primeiro tem 5 elementos, o outro tem apenas 2). Por esse motivo, o segundo elemento do vector menor foi repetido menos uma vez.

Estas duas soluções mostram, assim, exemplos da *regra da reciclagem*.

5. Considere os seguintes vectores:

```
a = c("L", 3, 3, T)  
b = c(F, 4, 1, 7)
```

Explique o que aconteceu aos valores guardados em ambos os vectores.

### Solução:

```
> a = c("L", 3, 3, T)
> b = c(F, 4, 1, 7)
> a
[1] "L"      "3"      "3"      "TRUE"
> b
[1] 0 4 1 7
```

Vectores constituem uma forma simples, mas poderosa, de armazenar informação em **R**, mas apenas de um só tipo de cada vez. Os tipos de objecto existentes num vector é automaticamente convertido de modo a existir um único tipo.

No caso do vector **a**, foram definidos objectos dos tipos **character**, **numeric** e **logical**, no entanto, o vector resultante contém apenas objectos do tipo **character** (repare nas aspas). O valor **T** (sem aspas) foi convertido para **"TRUE"**, pois **T** é apenas um *atalho* para o valor lógico **TRUE** (o mesmo acontece com os valores **F** e **FALSE**).

No segundo caso, o vector **b** foi definido com objectos de dois tipos: **logical** e **numeric**. O resultado é um vector de apenas valores numéricos, sendo que o valor lógico **FALSE** foi convertido para 0 (zero) (trata-se de uma convenção; o valor lógico **TRUE** teria sido convertido para 1).

A ordem de precedência com que estas conversões automáticas (coerção) se dão pode ser verificada no sistema de ajuda. O seguinte excerto foi retirado da ajuda da função **c()**, através do comando **?c**:

The output type is determined from the highest type of the components in the hierarchy  
NULL < raw < logical < integer < double < complex < character < list < expression.