

Objetivos

- Definir objetos (usar sinal de atribuição `<-`) e verificar a classe e o tipo dos objetos definidos.
- Definir sequências (vetores) numéricas, lógicas ou de caracteres (strings) usando o operador `:` e as funções `c` (concatenate), `seq` (sequence), `rep` (repete).
- Formar subsequências de vetores usando `[]`.
- Usar a ajuda do R (usando `help`, `?`) para verificar os argumentos (opções) das funções (comandos) utilizadas na resolução dos exercícios.
- Operar com vetores.

Antes de iniciar a resolução dos exercícios:

1. Crie uma pasta no seu ambiente de trabalho com o nome `aulas_cR`
2. Abra o programa R.
3. Mude a diretoria para a pasta `aulas_cR` (use a interface gráfica e a opção File).
4. Verifique que a diretoria foi alterada. (para tal use o comando `getwd()`).
5. Abra um documento de texto dentro do programa R (fazendo `new script`) e atribua-lhe o nome `exercicios1`. Este ficheiro poderá ser usado para guardar os comandos utilizados na resolução dos exercícios e para preparar um conjunto de instruções antes de os passar para a consola.

Exercícios

1. Considere os objetos do R definidos da seguinte forma

```
x <- c(2,-1,0,1.5)      y <- 1:4      nomes <- c("Ana", "Bruno", "Carla")
```

Qual a classe e o tipo desses objectos? Verifique também a classe e o tipo dos seguintes objetos:

```
x>y      table(x>y)      nomes=="Ana"
```

Aplique ainda as funções `length` e `sum` a cada um dos objetos considerados.

2. Considere os seguintes vectores numéricos:

$$\mathbf{x} = (2, -1, 0, 1.5) \quad \mathbf{y} = (1, 2, 3, 4) \quad \mathbf{z} = (2, 3)$$

Introduza \mathbf{x} , \mathbf{y} e \mathbf{z} no R, efectue os comandos seguintes e interprete os resultados:

- | | | |
|---|--|---|
| (a) $\mathbf{x} + \mathbf{y}$ | (b) $\mathbf{x} + 1$ | (c) $\mathbf{x} + \mathbf{z}$ |
| (d) $\mathbf{c}(\mathbf{x}, 4)$ | (e) $\mathbf{c}(\mathbf{x}, 4) + \mathbf{z}$ | (f) $\text{length}(\mathbf{c}(\mathbf{y}, \mathbf{c}(0, 2)))$ |
| (g) $\mathbf{x} * \mathbf{y}$ | (h) $\mathbf{x} * 2$ | (i) $\mathbf{x} * \mathbf{z}$ |
| (j) \mathbf{x}^2 | (k) $\text{abs}(\mathbf{x})$ | (l) $\text{sqrt}(\mathbf{x}^2)$ |
| (m) $\text{sum}(\mathbf{x})$ | (n) $\text{prod}(\mathbf{y})$ | (o) $\text{cumsum}(\mathbf{y})$ |
| (p) $\mathbf{x}[3]$ | (q) $\mathbf{x}[1 : 3]$ | (r) $\mathbf{x}[-3]$ |
| (s) $\mathbf{x}[\mathbf{x} > 1]$ | (t) $\text{sum}(\mathbf{x} > 1)$ | (u) $\mathbf{y}[\mathbf{y}! = 2]$ |
| (v) $\mathbf{x}[\mathbf{x} >= 0 \ \& \ \mathbf{x} < 2]$ | (w) $\mathbf{x}[\mathbf{x} < 0 \mid \mathbf{x} > 1]$ | (x) $\mathbf{y}[\mathbf{x} > 1]$ |

3. Defina os seguintes vectores:

- $(1, 2, 3, \dots, 30)$
- $(1, 3, 5, \dots, 27, 29)$
- $(29, 27, \dots, 3, 1)$
- $(1, 2, 3, \dots, 29, 30, 29, 28, \dots, 2, 1)$
- $(\text{FALSE}, \text{TRUE}, \text{FALSE}, \text{TRUE}, \text{TRUE})$.
- $(4, 6, 3)$ e guarde-o num objeto com o nome **vet**. Nas seguintes alíneas use a função **rep** para definir as sequências indicadas:
 - $(4, 6, 3, 4, 6, 3, \dots, 4, 6, 3)$ com 10 ocorrências de 4.
 - $(4, 6, 3, 4, 6, 3, \dots, 4, 6, 3, 4)$ com 11 ocorrências de 4, 10 ocorrências de 6 e 10 ocorrências de 3.
 - $(4, 4, \dots, 4, 6, 6, \dots, 6, 3, 3, \dots, 3)$ com 10 ocorrências de 4, 20 ocorrências de 6 e 30 ocorrências de 3.

4. Tendo em conta os padrões em cada uma das sequências seguintes, obtenha os seguintes vectores:

- $(1, 3, 3^2, 3^3, \dots, 3^{31})$
- $(0.1^3 0.2^1, 0.1^6 0.2^4, \dots, 0.1^{36} 0.2^{34})$
- $(2, \frac{2^2}{2}, \frac{2^3}{3}, \dots, \frac{2^{25}}{25})$

5. Calcule os resultados das seguintes expressões numéricas.

- $\sum_{i=10}^{100} (i^3 + 4i^2)$
- $\sum_{i=1}^{25} \left(\frac{2^i}{i} + \frac{3^i}{i^2} \right)$

6. Usando a função **cumprod**, calcule o valor da expressão

$$\frac{2}{3} + \frac{2}{3} \frac{4}{5} + \frac{2}{3} \frac{4}{5} \frac{6}{7} + \dots + \frac{2}{3} \frac{4}{5} \frac{6}{7} \dots \frac{38}{39}.$$

7. Usando a função `scan`, introduza no R os vetores disponíveis nos ficheiros `vetor_x.txt` e `vetor_y.txt` atribuindo-lhes os nomes `x` e `y`. Ambos os vetores têm comprimento 250.
- Qual o valor máximo de `x`?
 - Qual a média dos valores de `x`?
 - Que valores de `y` se encontram nas posições 2, 49 e 215?
 - Obtenha o vetor que se obtém de `x` quando se retiram todas as entradas com índice entre 67 e 213.
 - Quantas observações de `y` são superiores a 600?
 - Obtenha o vetor formado pelas observações de `y` superiores a 600.
 - Que posições correspondem às observações consideradas na alínea anterior? (Verifique como funciona a função `which`).
 - Que valores de `x` correspondem aos valores de `y` superiores a 600? (*correspondem* significa com os mesmos índices de posição).
 - Quantas observações de `y` se encontram entre no intervalo $[200, 600]$?
 - Quantos valores de `x` são divisíveis por 2? (o resto de uma divisão inteira é obtido com o operador `%%`).
 - Usando as funções `sort` e `order`,
 - Obtenha os valores de `x` na sua ordem crescente.
 - Obtenha os valores de `x` na ordem crescente dos valores de `y`.

8. Considere um dado (com 6 faces) viciado de modo a que a probabilidade de sair cada face seja dada pela seguinte tabela

face	1	2	3	4	5	6
probabilidade	0.1	0.1	0.1	0.1	0.2	0.4

- Guarde as probabilidades dadas num vetor `proba`. De acordo com essas probabilidades, quantas vezes sairá cada face em 100 lançamentos deste dado? Guarde os resultados num vetor `frequesp`.
- Usando a função `sample` simule 100 lançamentos deste dado e guarde os resultados obtidos.
- Calcule a média dos seus lançamentos.
- Quantas vezes saiu cada face? Guarde os resultados num vetor `frequobs`.
- Obtenha novamente os resultados da alínea anterior usando a função `table`.
- Usando a função `plot` represente graficamente a informação condensada na tabela anterior. Acrescente um título ao gráfico e nomes para os eixos usando os parâmetros `main`, `xlab`, `ylab`. Usando a função `points` e cores (`col`), adicione ao gráfico a informação contida no vetor `frequesp`.
- Calcule $\sum_{i=1}^6 (e_i - o_i)^2 / e_i$ onde e_i e o_i são as frequências esperadas e observadas das faces.

Informação: R é um ambiente de desenvolvimento integrado para diversas análises de dados, incluindo cálculo de estatísticas e execução de vários tipos de gráficos. As funções para a execução de gráficos podem ser *high-level* e criam um novo gráfico (exemplo: `plot`) ou *low-level* e servem para acrescentar informação a um gráfico já existente, tal como pontos, linhas, rótulos (`points`, `abline`, `title`, `lines`, `legend`...) ou preparar a janela gráfica com parâmetros específicos (`par`). Existe também uma função interativa (`locator`) que permite acrescentar informação a um gráfico existente usando o rato.