

Lista Prática 1

Exercício 1 (swirl). O pacote *swirl* permite aprender a usar o R de forma interativa diretamente no console do R. Para instalar e inicializar o *swirl*, use os comandos¹:

```
1 install.packages("swirl")  
2 library(swirl)  
3 swirl()
```

Aparecerá uma mensagem de inicialização e, então, digite seu nome diretamente no console. Se esta for a primeira vez utilizando o *swirl* aparecerá algumas instruções e, depois, será mostrado o seguinte repositório de cursos:

```
1 1: R Programming: The basics of programming in R  
2 2: Regression Models: The basics of regression modeling in R  
3 3: Statistical Inference: The basics of statistical inference in R  
4 4: Exploratory Data Analysis: The basics of exploring data in R  
5 5: Don't install anything for me. I'll do it myself.
```

Neste exercício, utilizaremos o curso “R Programming”, portanto, digite 1 para instalá-lo e, logo, digite 1 novamente para acessá-lo. Então, para iniciar uma lição, é necessário escolhê-la seguinte lista:

```
1 1: Basic Building Blocks      2: Workspace and Files  
2 3: Sequences of Numbers      4: Vectors  
3 5: Missing Values           6: Subsetting Vectors  
4 7: Matrices and Data Frames  8: Logic  
5 9: Functions                 10: lapply and sapply  
6 11: vapply and tapply        12: Looking at Data  
7 13: Simulation               14: Dates and Times  
8 15: Base Graphics
```

Faça as seguintes lições no *swirl*:

- a) 1: Basic Building Blocks
- b) 3: Sequences of Numbers
- c) 4: Vectors
- d) 5: Missing Values
- e) 6: Subsetting Vectors
- f) 7: Matrices and Data Frames
- g) 8: Logic
- h) 9: Functions

*Note que, para atribuição de valores a objetos, o *swirl*() exige que isso seja feito via “<-”, ou seja, não aceita respostas com atribuição por “=”).

¹Caso tenha dificuldades, assistir vídeo: <https://youtu.be/o10JfAjzd08>

Exercício 2. Considere os dois vetores $x = c(1, 3, 5:8)$ e $y = c(3, 2:-1, 10)$.

- a) O que é produzido pela expressão `rbind(x, y)`? Qual é a sua classe?
- b) Crie um vetor z com os elementos dos vetores x e y , ou seja, $z = c(1, 3, 5:8, 3, 2:-1, 10)$. Usando a função `matrix()` e o vetor z como input dela, crie o mesmo objeto produzido no item (a).

Resposta:

- a) Produz uma matriz com 2 linhas e 6 colunas:

```
1      [,1] [,2] [,3] [,4] [,5] [,6]
2 x         1   3   5   6   7   8
3 y         3   2   1   0  -1  10
```

- b)

```
1 z = c(x, y)
2 matrix(z, nrow=2, ncol=6, byrow=TRUE)
```

```
1      [,1] [,2] [,3] [,4] [,5] [,6]
2 [1,]     1   3   5   6   7   8
3 [2,]     3   2   1   0  -1  10
```

□

Exercício 3. Considere a lista $x = list(2, "a", "b", TRUE)$.

- a) O que `x[[2]]` retorna e qual é a sua classe?
- b) E `x[[1]]`?
- c) E `x[2]`?

Resposta:

- a) Retorna o vetor texto (*character*) “a”
- b) Retorna o vetor numérico (*numeric*) 2
- c) Retorna a lista composta pelo vetor texto (*character*) “a”.

Note que o operador `[]` retorna um objeto da mesma classe do objeto originário (aqui uma lista), enquanto `[[]]` retorna um objeto na classe do vetor que compõe a lista. □

Exercício 4. Considere o vetor $x = \{3, 5, 1, 10, 12, 6\}$. Escreva o comando em R que extraia do vetor os seguintes números:

- a) Maiores ou iguais do que 7
- b) Entre 5 e 10
- c) Diferentes de 10

d) *Pertencentes ao conjunto/vetor {7, 8, 9, 10, 11, 12}. Dica: Use o operador `%in%` para criar a expressão lógica.*

e) *Não-pertencentes ao conjunto/vetor {7, 8, 9, 10, 11, 12}.*

Resposta:

```
1 x = c(3, 5, 1, 10, 12, 6)
2
3 #(a)
4 x[x >= 7]
5
6 #(b)
7 x[x >= 5 & x <= 10]
8
9 #(c)
10 x[x != 10]
11
12 #(d)
13 x[x %in% 7:12] # ou x[x %in% c(7,8,9,10,11,12)]
14
15 #(e)
16 x[!(x %in% 7:12)]
```

□

Exercício 5. *Usando a base de dados `airquality`², responda os seguintes itens:*

a) *Valor de ‘Ozone’ na linha 47*

b) *Quantidade de missing values (NA) na coluna ‘Ozone’*

c) *Média da ‘Ozone’*

d) *Média de ‘Temp’ quando ‘Month’ = 6*

e) *Média de ‘Solar.R’ quando ‘Ozone’ > 31 e ‘Temp’ > 90*

f) *Maior valor de ‘Ozone’ quando ‘Month’ = 5*

Resposta:

a) `airquality$Ozone[47]` ou `airquality[47, 1]` \leadsto Resposta: 21

b) Para criar um vetor de TRUE/FALSE condicionada ao elemento ser um missing value, usamos `is.na()`. Lembre-se também que TRUE e FALSE correspondem, respectivamente aos números 1 e 0. Portanto, ao somarmos um vetor de TRUE/FALSE, obtemos a quantidade total de TRUE's.
`sum(is.na(airquality$Ozone))` \leadsto Resposta: 37

c) Do item anterior, sabemos que Ozone possui missing values e, portanto, ao utilizarmos a função `mean()` sem remover os missing values, o R reporta um NA.
`mean(airquality$Ozone, na.rm = TRUE)` \leadsto Resposta: 42.12931

²Base de dados nativa do R, basta escrever o nome dela para acessá-la.

d) `mean(airquality$Temp[airquality$Month == 6], na.rm = TRUE)` \leadsto Resposta: 79.1

e) `mean(airquality$Solar.R[airquality$Ozone > 31 & airquality$Temp > 90], na.rm = TRUE)` \leadsto Resposta: 212.8

f) `max(airquality$Ozone[airquality$Month == 5], na.rm = TRUE)` \leadsto Resposta: 115

□

Exercício 6. Considere $x = 1$. Crie um loop que, a cada repetição, divide o valor de x pela metade. Faça isso até $x < 1e-100$ (1×10^{-100}). Quantas repetições são necessárias para que essa condição seja satisfeito?

Resposta: 333 loops

```
1 x = 1
2 contador = 0
3
4 while (x > 1e-100) {
5     x = x/2
6     contador = contador + 1
7 }
8
9 contador
```

□

Exercício 7. Considere que João precise definir um x considerando uma função de desutilidade dada por $f(x) = x^4 - 6x^2 + 4x + 4$.

a) Suponha que João só possa escolher $0 \leq x \leq 3$. Como há infinitos pontos neste intervalo (e o computador não consegue lidar com infinitos valores), discretizaremos os possíveis valores de x criando um vetor $x = \text{seq}(\text{from} = 0, \text{to} = 3, \text{length} = 61)$. Qual é o valor de x que minimiza a desutilidade de João, $f(x)$?

b) Agora, use $x = \text{seq}(\text{from} = -3, \text{to} = 3, \text{length} = 121)$. O valor de x que minimiza $f(x)$ permanece o mesmo?

c) Use $x = \text{seq}(\text{from} = -100, \text{to} = 100, \text{length} = 121)$. O que acontece?

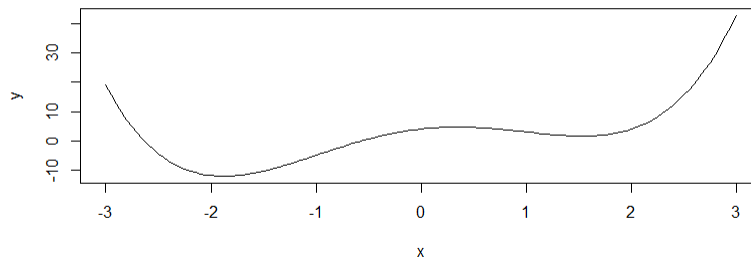
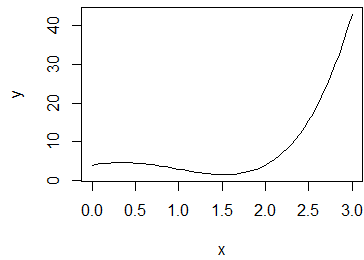
*Dica: Compare os casos usando gráficos: `plot(x, y, type="l")`.

Resposta:

a) Resposta: 1,55

```
1 x = seq(from = 0, to = 3, length = 61)
2 y = x^4 - 6*x^2 + 4*x + 4
3 x[which.min(y)]
```

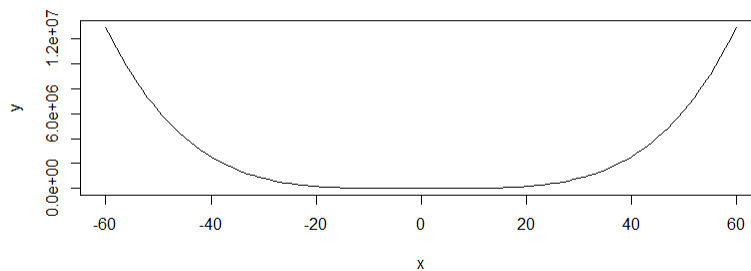
b) Resposta: -1,9 \leadsto valor que minimiza $f(x)$ mudou, pois o intervalo $0 \leq x \leq 3$ selecionava um mínimo local.



```
1 x = seq(from = -3, to = 3, length = 121)
2 y = x^4 - 6*x^2 + 4*x + 4
3 x[which.min(y)]
```

- c) Resposta: $-2 \leadsto$ aumentar o intervalo de potenciais valores de x , sem aumentar proporcionalmente a quantidade de pontos, diminuiu a precisão de seleção do x que minimiza $f(x)$. O mesmo ocorreria se diminuísse a quantidade de pontos dentro do intervalo.

```
1 x = seq(from = -60, to = 60, length = 121)
2 y = x^4 - 6*x^2 + 4*x + 4
3 x[which.min(y)]
```



□