

---

INSTITUTO SUPERIOR DE AGRONOMIA  
ESTATÍSTICA E DELINEAMENTO – 2022-23  
Resoluções dos Exercícios Introdutórios

1. Os comandos do R necessários para a resolução são os seguintes:

- (a) `> precip <- c(101.0, 60.7, 75.1, 19.9, 26.7, 10.5, 2.5, 39.8, 5.7, 51.7, 50.1, 170.6)`  
O resultado pode ser visualizado escrevendo o nome do objecto criado:

```
> precip
[1] 101.0 60.7 75.1 19.9 26.7 10.5 2.5 39.8 5.7 51.7 50.1 170.6
```

- (b) `meses <- c("Jan", "Fev", "Mar", "Abr", "Mai", "Jun", "Jul", "Ago", "Set", "Out", "Nov", "Dez")`  
Resultado:

```
> meses
[1] "Jan" "Fev" "Mar" "Abr" "Mai" "Jun" "Jul" "Ago" "Set" "Out" "Nov" "Dez"
```

- (c) `names(precip) <- meses`  
Resultado:

```
> precip
  Jan  Fev  Mar  Abr  Mai  Jun  Jul  Ago  Set  Out  Nov  Dez
101.0 60.7 75.1 19.9 26.7 10.5 2.5 39.8 5.7 51.7 50.1 170.6
```

- (d) i. `> sum(precip)`  
[1] 614.3  
ii. `> mean(precip)`  
[1] 51.19167  
iii. `> median(precip)`  
[1] 44.95  
iv. Assinale-se que o R define, por omissão, os quantis de forma ligeiramente diferente da forma como foram definidos na disciplina de Estatística dos primeiros ciclos do ISA:

```
> quantile(precip, 0.75)
75%
64.3
```

Na realidade, o comando `quantile` pode calcular nove diferentes definições de quantis (caso tenha interesse, veja `help(quantile)` para os pormenores de cada tipo). A definição de quantis usada em Estatística corresponde ao segundo tipo:

```
> quantile(precip, 0.75, type=2)
75%
67.9
```

```
v. > var(precip)
[1] 2291.604
```

```
vi. > sd(precip)          ou          > sqrt(var(precip))
[1] 47.87071              [1] 47.87071
```

```
vii. > min(precip)
[1] 2.5
```

```
viii. > max(precip)
[1] 170.6
```

- 
- (e) `> summary(precip)`
- | Min. | 1st Qu. | Median | Mean  | 3rd Qu. | Max.   |
|------|---------|--------|-------|---------|--------|
| 2.50 | 17.55   | 44.95  | 51.19 | 64.30   | 170.60 |
- (f) i. `> precip[10]`
- ```
Out
51.7
ou, em alternativa,
> precip["Out"]
Out
51.7
```
- ii. `> precip[6:9]`
- | Jun  | Jul | Ago  | Set |
|------|-----|------|-----|
| 10.5 | 2.5 | 39.8 | 5.7 |
- (g) i. O comando `precip > 50` cria um vector de valores lógicos “verdade” (TRUE) ou “falso” (FALSE), resultantes da comparação de cada valor do vector `precip` com o valor 50:
- ```
> precip > 50
Jan  Feb  Mar  Abr  Mai  Jun  Jul  Ago  Set  Out  Nov  Dez
TRUE TRUE TRUE FALSE FALSE FALSE FALSE FALSE FALSE TRUE TRUE TRUE
```
- Este vector lógico pode ser usado, por sua vez, para indexar o vector `precip`, fazendo com que apenas os valores de precipitação correspondentes ao valor lógico TRUE (ou seja, apenas as precipitações superiores a 50mm) sejam devolvidos:
- ```
> precip[precip > 50]
Jan  Feb  Mar  Out  Nov  Dez
101.0 60.7 75.1 51.7 50.1 170.6
```
- ii. Para obter apenas as precipitações superiores à média pode adequar-se a condição lógica, da seguinte forma: `precip > mean(precip)`. Assim, as precipitações acima da média podem ser obtidas pelo comando:
- ```
> precip[precip > mean(precip)]
Jan  Feb  Mar  Out  Dez
101.0 60.7 75.1 51.7 170.6
```
- (h) i. Tal como na alínea anterior, o comando `precip == min(precip)` devolve um vector de valores lógicos (TRUE ou FALSE) correspondentes a cada elemento do vector `precip` ter, ou não, valor igual ao valor de precipitação mínima (atenção ao duplo sinal de igualdade, que é a forma obrigatória de perguntar se se verifica uma igualdade de valores). Esse vector lógico tem comprimento igual ao vector original `precip`, o que em vectores com muitos dados pode dificultar a identificação do(s) elemento(s) que verificam a condição lógica. O comando `which` facilita essa identificação, uma vez que selecciona apenas os elementos dum vector que verificam a condição lógica. Nesta alínea é pedido para identificar o mês onde se verificou a precipitação mínima, e isso pode ser feito através do seguinte comando:
- ```
> which(precip == min(precip))
Jul
7
```
- Repare-se que o valor devolvido (7) *não* é a precipitação mínima, mas o índice da posição no vector `precip` onde se encontra o valor mínimo (neste caso, o sétimo mês, Julho). Uma forma alternativa de identificar o mês de menor precipitação seria o de utilizar a mesma condição lógica para indexar o vector `meses`. Esta indexação cruzada é possível

---

porque os vectores `meses` e `precip` têm o mesmo tamanho, e posições correspondentes. Eis a resposta utilizando esta indexação cruzada:

```
> meses[precip == min(precip)]
[1] "Jul"
```

ii. Tal como no ponto anterior, a resposta pode ser obtida da seguinte forma:

```
> which(precip == max(precip))
Dez
12
```

Nota: Inspeccione o resultado do comando `precip == max(precip)`. Atenção ao duplo sinal de igualdade.

(i) Executar o comando `plot(precip)`

(j) Executar os comandos `plot(precip, type="l")` e `plot(precip, type="h")`. Para dados de precipitação mensal será mais adequado o gráfico tipo histograma, produzido pela opção `type='h'`.

2. Para visualizar os dados, basta escrever `sunspots`.

(a) `> length(sunspots)`

```
[1] 2820
```

(b) Os comandos necessários são:

i. `hist(sunspots)`

ii. `hist(sunspots, breaks=(0:26)*10)`

(c) i. `> quantile(sunspots)`

```
0%      25%     50%     75%    100%
0.000  15.700  42.000  74.925 253.800
```

ii. `> quantile(sunspots, 0.9)`

```
90%
112
```

(d) `> summary(sunspots)`

```
Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
0.00  15.70   42.00   51.27  74.93 253.80
```

(e) O comando pedido é `boxplot(sunspots)`.

3. Eis a resolução do exercício com os dados relativos a coelhos.

(a) Eis a criação do vector dos pesos:

```
> peso <- c(1.5, 1.4, 1.4, 1.2, 1.4, 2.7, 2.9, 2.1, 3.0, 3.3, 2.1, 2.2, 2.4,
+          2.0, 2.5, 1.3, 1.0, 1.1, 1.3, 1.5)
> peso
[1] 1.5 1.4 1.4 1.2 1.4 2.7 2.9 2.1 3.0 3.3 2.1 2.2 2.4 2.0 2.5 1.3 1.0 1.1 1.3
[20] 1.5
```

Para criar `dieta` e `tratamento` será necessário usar o comando `factor`. Uma vez que existem numerosas repetições, utilizar-se-á o comando `rep`, que permite gerar valores repetidos. O primeiro argumento do comando `rep` é sempre o vector de valores que se desejam repetir. O segundo argumento, de nome `times`, pode ser um vector numérico do mesmo tamanho, indicando quantas vezes deve ser repetido o correspondente valor do vector original. É o que sucede na seguinte criação do factor `dieta`, onde se indica que cada uma das letras deve ser repetida 5 vezes:

---

```

> dieta <- factor(rep( c("A","B","C","D") , times=c(5,5,5,5) ))
> dieta
[1] A A A A A B B B B B C C C C C D D D D D
Levels: A B C D

```

Quando o segundo argumento do comando `rep` fôr um único número inteiro, a totalidade do vector é repetido esse número de vezes. É o que sucede na criação do vector `tratamento`, onde o vector de inteiros de 1 a 5 (gerado por `1:5`) é repetido quatro vezes.

```

> tratamento <- factor(rep( 1:5 , times=4 ))
> tratamento
[1] 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5
Levels: 1 2 3 4 5

```

(b) Aqui está a criação da *data frame*, e o resultado obtido:

```

> coelhos <- data.frame(peso, dieta, tratamento)
> coelhos
  peso dieta tratamento
1  1.5    A           1
2  1.4    A           2
3  1.4    A           3
4  1.2    A           4
5  1.4    A           5
6  2.7    B           1
7  2.9    B           2
8  2.1    B           3
9  3.0    B           4
10 3.3    B           5
11 2.1    C           1
12 2.2    C           2
13 2.4    C           3
14 2.0    C           4
15 2.5    C           5
16 1.3    D           1
17 1.0    D           2
18 1.1    D           3
19 1.3    D           4
20 1.5    D           5

```

(c) O enunciado pede o seguinte comando:

```

> summary(coelhos)
  peso      dieta tratamento
Min.   :1.000  A:5      1:4
1st Qu.:1.375  B:5      2:4
Median :1.750  C:5      3:4
Mean   :1.915  D:5      4:4
3rd Qu.:2.425          5:4
Max.   :3.300

```

Como é visível, na coluna numérica `peso` o comando `summary` produz as habituais indicadores (mínimo, primeiro quartil, mediana, média, terceiro quartil e máximo). Já para factores, o comando `summary` lista as diferentes categorias (níveis do factor) e, à direita do símbolo `‘:’`, o número de vezes que cada nível é repetido.

- 
- (d) Uma vez que as colunas da *data frame* `coelhos` têm nomes atribuídos, a forma mais fácil de seleccionar uma coluna é escrever o nome da *data frame* e, separado por um cifrão (`$`), o nome da coluna que se pretende. Assim, para extrair apenas a coluna `peso` e calcular a respectiva média, escreve-se:

```
> coelhos$peso
[1] 1.5 1.4 1.4 1.2 1.4 2.7 2.9 2.1 3.0 3.3 2.1 2.2 2.4 2.0 2.5 1.3 1.0 1.1 1.3
[20] 1.5
> mean(coelhos$peso)
[1] 1.915
```

Alternativamente, pode seleccionar-se qualquer subconjunto de linhas e/ou colunas da *data frame* indicando, entre parênteses rectos após o nome do objecto, os números de linha e/ou coluna que se deseja (separados por uma vírgula). Se quisermos seleccionar as três primeiras linhas das colunas 1 e 3, podemos escrever:

```
> coelhos[c(1,2,3),c(1,3)]
  peso tratamento
1  1.5           1
2  1.4           2
3  1.4           3
```

Para obter a *totalidade* da primeira coluna, basta deixar em branco o índice de linhas (antes da vírgula) e escrever apenas o número da coluna após a vírgula:

```
> coelhos[,1]
[1] 1.5 1.4 1.4 1.2 1.4 2.7 2.9 2.1 3.0 3.3 2.1 2.2 2.4 2.0 2.5 1.3 1.0 1.1 1.3
[20] 1.5
```

- (e) Na sequência do que se viu na alínea anterior, e sabendo que a dieta C corresponde às linhas 11 a 15, podemos corresponder ao pedido do enunciado escrevendo:

```
> coelhos[11:15,]
  peso dieta tratamento
11 2.1    C           1
12 2.2    C           2
13 2.4    C           3
14 2.0    C           4
15 2.5    C           5
```

No entanto, o R oferece a poderosa possibilidade de escolher elementos dum vector ou *data frame* através da especificação de alguma condição. No nosso exemplo, a condição de selecção será que a coluna `dieta` tenha o valor C. É possível (à semelhança do que se viu no primeiro Exercício) criar um vector de valores lógicos (verdadeiro/falso) correspondendo à condição:

```
> coelhos$dieta=="C"
[1] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE  TRUE
[13]  TRUE  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE
```

Este vector de valores lógicos pode ser usado para indicar quais as linhas da *data frame* que desejamos escolher (apenas sendo escolhidas as que correspondem ao valor lógico TRUE):

```
> coelhos[coelhos$dieta=="C" , ]
  peso dieta tratamento
11 2.1    C           1
12 2.2    C           2
13 2.4    C           3
14 2.0    C           4
15 2.5    C           5
```

- 
- (f) O comando `apply` permite aplicar uma mesma função a todas as linhas ou a toda as colunas duma *data frame*. O nome da *data frame* é o primeiro argumento do comando `apply`. O segundo argumento especifica a dimensão à qual se pretende aplicar a função: 1 indica as linhas e 2 indica as colunas. O terceiro argumento é o nome da função que se deseja aplicar. Assim, para obter o valor máximo de cada coluna pode invocar-se o seguinte comando:

```
> apply(coelhos, 2, max)
      peso      dieta tratamento
"3.3"      "D"          "5"
```

Registe-se como a função `max` tanto age sobre valores numéricos, como sobre níveis de um factor (usando-se neste caso a ordem alfabética dos nomes dos níveis do factor).