
INSTITUTO SUPERIOR DE AGRONOMIA
ESTATÍSTICA E DELINEAMENTO – 2014/15
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. `> var(precip)`

```
[1] 2291.604
```

v. `> sd(precip)`

```
[1] 47.87071
```

ou

`> sqrt(var(precip))`

```
[1] 47.87071
```

vi. `> min(precip)`

```
[1] 2.5
```

vii. `> max(precip)`

```
[1] 170.6
```

(e) 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
```

```
(f) i. > precip[precip > 50]
      Jan  Feb  Mar  Out  Nov  Dez
      101.0 60.7 75.1 51.7 50.1 170.6
ii. > precip[precip > mean(precip)]
      Jan  Feb  Mar  Out  Dez
      101.0 60.7 75.1 51.7 170.6
```

Nota: Inspeccione o resultado dos comandos `precip > 50` e `precip > mean(precip)`. Estes comandos criam vectores de valores lógicos “verdade” (TRUE) ou “falso” (FALSE), resultantes da comparação de cada valor do vector `precip` com o valor 50 ou com o valor médio. Esses vectores de valores lógicos podem então ser usados para seleccionar alguns elementos do vector `precip` - apenas aqueles a que corresponda o valor lógico “verdade”.

```
(g) i. > which(precip == min(precip))
      Jul
      7
ii. > which(precip == max(precip))
      Dez
      12
```

Nota: Inspeccione o resultado dos comandos `precip == min(precip)` e `precip == max(precip)`. Atenção ao duplo sinal de igualdade.

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

(i) 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. (a) O comando pedido é

```
> estacas <- matrix(nrow=4, ncol=3, c(26,32,24,39,18,22,24,19,16,6,12,2))
```

O resultado é:

```
> estacas
      [,1] [,2] [,3]
[1,]   26   18   16
[2,]   32   22    6
[3,]   24   24   12
[4,]   39   19    2
```

- (b) `> rownames(estacas) <- c("sI/sB", "cI/sB", "sI/cB", "cI/cB")`
`> colnames(estacas) <- c("Morte", "Calo", "Enraizamento")`

Resultado:

```
> estacas
      Morte Calo Enraizamento
sI/sB   26   18             16
cI/sB   32   22             6
sI/cB   24   24             12
cI/cB   39   19             2
```

- (c) Pedese para extrair a terceira (última) coluna da matriz, como se fosse um vector. O comando adequado envolve a *indexação* da matriz, que utiliza parenteses rectos, indicando-se antes duma vírgula o número das linhas a extrair, e depois da vírgula o número de colunas a extrair ou os respectivos nomes (caso tenham sido atribuídos). Quando se pretende a totalidade das linhas ou colunas, pode-se deixar o respectivo espaço em branco, como se indica de seguida:

```
> estacas[,3]          ou          > estacas["Enraizamento"]
sI/sB cI/sB sI/cB cI/cB          sI/sB cI/sB sI/cB cI/cB
   16    6   12    2          16    6   12    2
```

Repare-se que, por omissão, uma coluna duma matriz é transformada num vector (e não numa matriz com uma única coluna). Repare-se ainda que este vector apenas é mostrado no ecrã, não tendo sido guardado em nenhum novo objecto. caso se desejasse criar um novo objecto contendo esse vector seria necessário utilizar o sinal de atribuição ("`<-`"). Para obter o número total de estacas enraizadas, basta somar os elementos do vector:

```
> sum(estacas["Enraizamento"])
[1] 36
```

- (d) Embora se pudesse repetir para cada linha e coluna aquilo que se fez na alinea anterior, o comando `apply` permite aplicar uma função (indicada com terceiro argumento do comando) a uma matriz (primeiro argumento), na dimensão 1 (linhas) ou 2 (colunas):

```
> apply(estacas,1,sum)
sI/sB cI/sB sI/cB cI/cB
   60    60    60    60
> apply(estacas,2,sum)
      Morte      Calo Enraizamento
      121      83      36
```

- (e) Aplicando o comando `sum` a uma matriz, obtém-se a soma de todos os seus elementos. Assim, basta dividir os vectores de somas de linha/columna obtidos na alinea anterior por essa soma:

```

> apply(estacas,1,sum)/sum(estacas)
sI/sB cI/sB sI/cB cI/cB
0.25 0.25 0.25 0.25
> apply(estacas,2,sum)/sum(estacas)
      Morte      Calo Enraizamento
0.5041667 0.3458333 0.1500000

```

(f) Trata-se da submatriz constituída pelas linhas 1 e 3 da matriz. Assim:

```

> estacas[c(1,3),]
      Morte Calo Enraizamento
sI/sB    26  18             16
sI/cB    24  24             12

```

4. (a) `> dbinom(5,8,0.5)`

```
[1] 0.21875
```

(b) Os valores possíveis são os inteiros de 0 a 8 (inclusive).

```
> dbinom(0:8,8,0.5)
```

```
[1] 0.00390625 0.03125000 0.10937500 0.21875000 0.27343750 0.21875000 0.10937500
[8] 0.03125000 0.00390625
```

(c) `> sum(dbinom(0:8,8,0.5)*(0:8))`

```
[1] 4
```

Confirma-se a expressão geral do valor esperado duma $B(m,p)$: $E[X] = 8 \times 0.5 = 4$.

(d) `> pbinom(5,8,0.5)`

```
[1] 0.8554688
```

(e) `> 1-pbinom(3,8,0.5)`

```
[1] 0.6367187
```

5. Poisson.

(a) `> dpois(3,2.2)`

```
[1] 0.1966387
```

Confirmação (tome nota das funções utilizadas para obter o factorial e a exponencial):

```
> exp(-2.2)*2.2^3/factorial(3)
```

```
[1] 0.1966387
```

(b) `> 1-ppois(4,2.2)`

```
[1] 0.07249631
```

(c) `> ppois(4,2.2)-ppois(1,2.2)`

```
[1] 0.5729336
```

ou, alternativamente,

```
> sum(dpois(2:4,2.2))
```

```
[1] 0.5729336
```