

Exame 1ª Época – Versão A – 27/06/2016

Número: _____ Nome: _____

Leia, por favor, com atenção:

1. Este exame deverá ser realizado no enunciado, sem acesso a um computador.
 2. Poderá consultar o formulário dado em anexo ao exame.
 3. É proibido o uso de qualquer outro material de apoio (livros, apontamentos, telemóvel), assim como a troca de qualquer informação com os colegas.
 4. A entrega do exame e a saída da sala só são possíveis no final do exame.
 5. Deverá escrever o seu nome, número e curso no cabeçalho desta folha.
 6. As respostas às questões deverão ser dadas, exclusivamente, na folha do enunciado, no espaço reservado para tal. Estas respostas deverão ser apenas código em **R**.
 7. Não é necessário escrever o resultado do código, mas apenas o código em si.
 8. O não cumprimento de alguma das regras conduzirá à anulação do exame.
 9. A duração do exame é de 75 minutos.
-
1. Uma das vantagens em utilizar uma linguagem de programação interpretada como o **R** é a interactividade com o utilizador, que, por sua vez, é útil na resolução de problemas de matemática.
 - (a) A regressão é uma das técnicas utilizadas em modelação estatística e econometria. Alguns desses modelos são definidos por equações quadráticas que, tipicamente, têm a forma ilustrada na Equação 1: [2 val.]

$$ax^2 + bx + c = 0 \quad (1)$$

onde x representa um número desconhecido, e a , b , e c são os coeficientes da equação e representam números conhecidos, tal que $a \neq 0$. Se $a = 0$, então a equação é linear, e não quadrática.

Uma forma de resolver uma equação quadrática é através da fórmula resolvente (Equação 2).

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \quad (2)$$

Escreva uma função que permita resolver uma equação quadrática, usando a fórmula resolvente. A função deve devolver **NULL** caso os coeficientes não sejam de uma equação quadrática.

Solução:

```
FormulaResolvente = function (A, B, C) {  
  if (A == 0) {  
    return (NULL)  
  } else {  
    zeros = c()  
    num = sqrt(B ^ 2 - 4 * A * C)  
    den = 2 * A  
    zeros[1] = (-B + num) / den  
    zeros[2] = (-B - num) / den  
    return (zeros)  
  }  
}
```

- (b) No exercício anterior, procurava-se o valor de uma variável desconhecida. O **R** também pode ser usado para encontrar os valores de múltiplas variáveis desconhecidas, desde que se trate de um *problema bem-posto*. A função `solve()` serve para resolver um sistema de equações linear. Escreva o código necessário que permita resolver o seguinte sistema de equações. [2 val.]

$$\begin{cases} 3x + 0.14y = \pi \\ 2.71x - 0.82y - 0.82z = 0 \\ y + z = 3 \end{cases}$$

Solução:

```
> coefs = matrix(c(3, .14, 0, 2.71, -0.82, -0.82, 0, 1, 1),  
+ nrow=3, ncol=3, byrow=TRUE)  
> ys = c(pi, 0, 3)  
> solve(coefs, ys)  
[1] 0.90774908 2.98818158 0.01181842
```

2. A linguagem de programação **R** é compatível com o paradigma de programação funcional. Neste paradigma, é dado um maior ênfase à **utilização de funções**, em contraste à programação imperativa, que privilegia mudanças no estado do programa. Geralmente, a programação funcional tem ainda a vantagem de ser possível escrever a mesma funcionalidade em menos código.

- (a) Considere que a variável `islands` é um **vector** de números inteiros, contendo as áreas das maiores massas de terra. Cada posição do vector tem também um nome associado. [2 val.]

Excerto do vector islands

```
> islands
```

Africa	Antarctica	Asia	Australia	Axel Heiberg
11506	5500	16988	2968	16
Baffin	Banks	Borneo	Britain	Celebes
184	23	280	84	73

Analise o seguinte bloco de código.

```
1 i = 1
2 s = 0
3 while (i < length(islands)) {
4     if ( islands[i] > sd(islands)) {
5         s = islands[i] + s
6     }
7     i = i + 1
8 }
9 # para a sua solução, não precisa de considerar a seguinte linha
10 names(s) = NULL

> s
[1] 53924
```

Reescreva-o sem usar uma estrutura iterativa.

Solução:

```
sum(islands[islands > sd(islands)])
```

(b) Analise o seguinte bloco de código.

[3 val.]

```
1 s = c()
2 for (num in rnorm(10)) {
3
4     if(num > 0) {
5         s = c(s, "+")
6     } else {
7         if(num == 0) {
8             s = c(s, "0")
9         } else {
10            s = c(s, "-")
11        }
12    }
13 }

> s
[1] "-" "+" "+" "-" "-" "-" "+" "+" "+" "+"
```

Reescreva-o sem usar uma estrutura iterativa.

Solução:

```
sapply(rnorm(10), function(x) if(x > 0) {"+"} else if (x == 0) {'0'} else {'-'})
```

3. Em **R** é fácil trabalhar com ficheiros de diferentes formatos e manipular a informação contida neles.

(a) Considere o seguinte conteúdo de um ficheiro de nome *pauta.dat*:

[1.5 val.]

```
Nome;AM2;CEGI;DP1;Est1;ISC;Mrkt
Regina;12,1;14,4;15,2;11,7;10,0;15,5
Dudu;8,2;6,6;12,3;9,9;13,1;16,7
Gioberte Nelson;13,3;13,3;13,3;13,3;13,3;13,3
Enzo;18,0;19,0;20,0;17,0;19,0;15,0
Valter Disney;10,0;11,1;12,2;13,3;14,4;15,5
```

Escreva o código necessário para guardar o conteúdo deste ficheiro num `data.frame`, com 5 linhas e 7 colunas, guardado na variável `pauta`.

Solução:

```
pauta = read.table(file="pauta.dat", header=TRUE, sep=";", dec=",")
```

- (b) Considere a variável `pauta`, contendo a tabela da alínea anterior. Essa tabela contém as notas de 5 alunos em 6 disciplinas. Modifique-a, acrescentando as seguintes colunas:

dif Indica se a diferença entre as notas a CEGI e a AM2 é ou não superior a 2 valores.

med O valor da média ponderada das notas de CEGI e de ISC, dando um peso de 70% a CEGI e 30% a ISC.

Não utilize estruturas iterativas.

Solução:

```
pauta = transform(pauta, dif = abs(CEGI - AM2) > 2,  
                  med = CEGI * 0.7 + ISC * 0.3)
```

Nota: A não utilização da função `abs()` não foi penalizada.

4. Em **R** existe também um vasto leque de funções para criar gráficos, sendo relativamente fácil produzir imagens de elevada qualidade. Nas próximas questões, analise os gráficos e escreva o código necessário para os reproduzir.

- (a) Escreva o código necessário para reproduzir o gráfico na Figura 1, em que `x = rnorm(100)`. [1.5 val.]

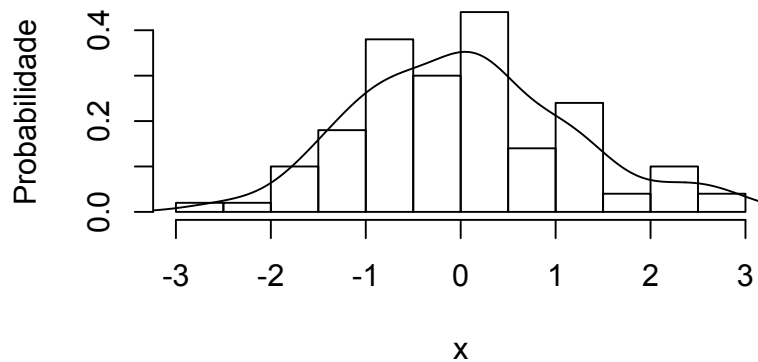


Figura 1

Solução:

```
hist(x, main = "", ylab = "Probabilidade", probability = TRUE)  
lines(density(x))
```

- (b) Escreva o código necessário para reproduzir o gráfico na Figura 2. Considere as seguintes linhas de código. [1.5 val.]

```
> y = runif(10, 0, 20)
> mean(y)
[1] 10.49255
```

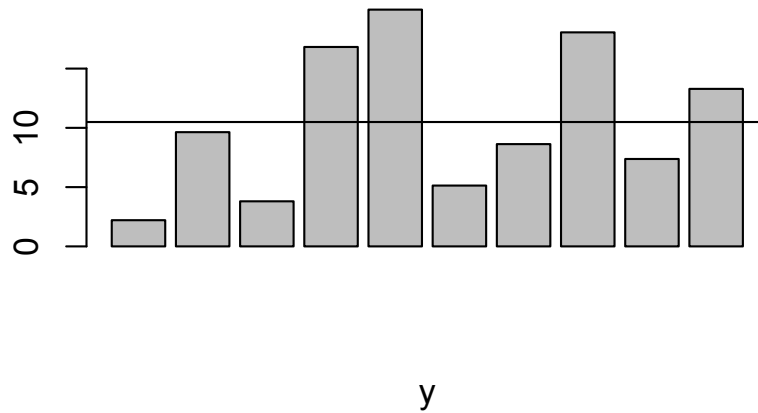


Figura 2

Solução:

```
barplot(y, xlab='y')
abline(h=mean(y))
```

5. O `data.frame` `rock` contém valores de quatro propriedades de 48 amostras de rocha. As seguintes instruções executadas em **R** fornecem uma noção sobre a estrutura deste objecto. [4 val.]

```
# primeiras 5 linhas do data.frame rock
```

```
rock
  area    peri    shape  perm
1  4990 2791.900 0.0903296  6.3
2  7002 3892.600 0.1486220  6.3
3  7558 3930.660 0.1833120  6.3
4  7352 3869.320 0.1170630  6.3
5  7943 3948.540 0.1224170 17.1
```

```
> str(rock)
'data.frame':    48 obs. of  4 variables:
 $ area : int  4990 7002 7558 7352 7943 7979 9333 8209 8393 6425 ...
 $ peri : num  2792 3893 3931 3869 3949 ...
 $ shape: num  0.0903 0.1486 0.1833 0.1171 0.1224 ...
 $ perm : num  6.3 6.3 6.3 6.3 17.1 17.1 17.1 17.1 119 119 ...
```

Explique, por palavras suas, o que se pretende encontrar com o seguinte bloco de código, e como isso é conseguido.

```
1 qs = apply(rock, 2, quantile, .25)
2 rock[apply(rock, 1, function(x) sum(x > qs) == 2), ]
```

Solução:

Pretende-se encontrar quais são as amostras em que o valor de duas das suas variáveis são maiores que o respectivo primeiro quartil.

Na primeira linha, o `data.frame` `rock` é percorrido pelas suas colunas, calculando, para cada uma delas, o valor do primeiro quartil.

Na segunda linha, o mesmo `data.frame` é percorrido pelas linhas, avaliando-se uma função anónima em cada linha. Esta função compara (verifica se é maior) o valor de cada variável numa linha com o respectivo quartil, calculado na linha anterior.

O resultado dessa comparação é um vector de valores lógicos. O número de valores que são `TRUE` é calculado pela função `sum`, sendo feita a conversão de lógico para inteiro implicitamente. De seguida, verifica-se se esta contagem é igual a dois, retornando o valor lógico correspondente.

Por último, esse vector de valores lógicos é usado para seleccionar apenas as amostras (linhas) em que a condição pretendida é verificada.

Pergunta	1	2	3	4	5	Total
Cotação	4	5	4	3	4	20
Cotação obtida						

(a preencher pelo docente)