NOVA IMS – Universidade Nova de Lisboa
Computação em Estatística e Gestão de Informação – 2º Semestre
2014/15

Exame Época Especial – 25/09/2015

Curso: _____ Número: _____ Nome: _____

Leia, por favor, com atenção:

1. Este enunciado corresponde à primeira parte do exame (**R**). O enunciado da segunda parte (SAS) será entregue após o aluno entregar a resolução da primeira parte.
2. Cada parte está cotada para 10 valores e tem a nota mínima de 4 valores.
3. Esta parte deverá ser realizado com acesso a um computador com o programa **R** instalado. Só poderá aceder ao programa **R** e ao sistema de ajuda do mesmo.
4. Os dados necessários para resolver os problemas já se encontram disponíveis na sua instalação do **R**.
5. É proibido o uso de qualquer material de apoio (livros, apontamentos, telemóvel), assim como a troca de qualquer informação com os colegas.
6. Deverá escrever o seu nome, número e curso no cabeçalho desta folha.
7. As respostas às questões deverão ser dadas, exclusivamente, na folha do enunciado, no espaço reservado para tal. Estas respostas deverão ser código em **R**.
8. O não cumprimento de alguma das regras conduzirá à anulação do exame.
9. A duração do exame, considerando ambas as partes, é de **2 horas**.

1. O vector `rivers` contém o comprimento, em milhas, dos 141 principais rios norte americanos.

   (a) Construa um novo vector que contenha o comprimento dos 50 maiores rios, mas considerando apenas os rios numa posição ímpar, aquando da sua ordenação. Ou seja, o rio mais longo, o 3º rio mais longo, o 5º rio mais longo, o 7º mais longo, e assim sucessivamente, até ao máximo de 50 rios. [1 val.]

Solução:

```
> sort(rivers, dec=T)[seq(from=1, by=2, length=50)]
[1] 3710 2348 1885 1459 1306 1243 1171 1054 1000  906  900  870  840  780  735
[16]  730  720  696  671  630  620  610  600  600  560  540  529  525  505  500
[31]  470  460  450  444  431  425  420  411  407  390  383  380  375  360  360
[46]  350  350  340  336  330
```

(b) Escreva a função `perc`, que receba dois argumentos: um vector numérico, `v` e um número, `x`. A função devolve o percentil de `v` correspondente ao valor de `x`. [1 val.]

Por exemplo, considerando o vector `rivers`:

```
> perc(rivers, 680)
[1] 0.751773
> perc(rivers, 425)
[1] 0.5035461
> perc(rivers, 10)
[1] 0
> perc(rivers, 10000)
[1] 1
```

Estes valores correspondem aos obtidos com a função `quantile`.

```
> quantile(rivers)
0%    25%    50%    75%    100%
135   310    425    680    3710
```

Solução:

```
perc = function (v, x) {
        return (length(subset(v, v <= x)) / length(v))
}
```

(c) Suponha que, para uma dado estudo, não interessa considerar os rios com comprimento inferior a 500 km (aproximadamente 311 mi). Utilize a função `replace` para criar o vector `rios`, substituindo o valor do comprimento dos referidos rios, por `NA`. [1 val.]

Solução:

```
rios = replace(rivers, rivers < 311, NA)
```

(d) O coeficiente de variação (c_v) é uma medida de dispersão relativa, útil quando duas distribuições têm valores médios (μ) diferentes, caso em que o desvio padrão (σ) não é comparável (Equação 1). [0.5 val.]

$$c_v = \frac{\sigma}{\mu} \tag{1}$$

Calcule o coeficiente de variação do vector `rios`.

Se não resolveu a alínea anterior, considere `rios = ifelse(rivers > 375, rivers, NA)`.

Solução:

```
> sd(rios, na.rm=T) / mean(rios, na.rm=T)
[1] 0.746284
```

2. A tabela `UCBAdmissions` contém informação sobre os candidatos à Universidade da Califórnia em Berkeley (UCB), em 1973, nos seis maiores departamentos. Os dados estão dispostos num objecto tridimensional, cujas variáveis são:

   1 – Admit indica se o candidato foi aceite (*Admitted*) ou não (*Rejected*);

   2 – Gender género (*Male / Female*);

   3 – Dept departamento que recebeu a candidatura (*A, B, C, D, E, F*).

(a) Encontre o número total de candidaturas rejeitadas. Esse valor corresponde a que percentagem do total de candidaturas? [1 val.]

Solução:

```
# para simplificar e facilitar na escrita do código, pode-se criar uma cópia
# da variável UCBAdmissions com um nome mais curto
> A = UCBAdmissions
> (r = sum(A['Rejected', , ]))
[1] 2771
> r / sum(A) * 100
[1] 61.22404
```

(b) Nesse ano, a UCB foi acusada de favorecer candidatos do sexo masculino. A taxa de aceitação corresponde ao rácio entre o número de candidatos admitidos e o número de candidatos. Calcule a taxa de aceitação de candidatos separadamente por cada um dos géneros, no total dos seis departamentos. [1 val.]

Solução:

```
> (ta.male = sum(A['Admitted', 'Male', ]) / sum(A[, 'Male', ]))
[1] 0.4451877
> (ta.female = sum(A['Admitted', 'Female', ]) / sum(A[, 'Female', ]))
[1] 0.3035422
```

Nos seis maiores departamentos, houve uma aceitação de 44,5% dos candidatos masculinos e de 30,4% dos candidatos femininos.

(c) Numa análise estatística a um conjunto de dados, pode-se encontrar uma determinada tendência que pode desaparecer ou mudar, quando os mesmos dados são analisados por grupos. Este efeito é denominado por Paradoxo de Simpson. Determine a taxa de aceitação, como na alínea anterior, mas agora separando também por cada um dos departamentos. Na sua resolução, evite repetir código. [2 val.]

Solução:

```
ta = function(gen) {
        apply(A, 3, function(t, g) t['Admitted', g] / sum(t[ , g]), gen)
}
> ta('Male')
A         B         C         D         E         F
0.62060606 0.63035714 0.36923077 0.33093525 0.27748691 0.05898123
> ta('Female')
A         B         C         D         E         F
0.82407407 0.68000000 0.34064081 0.34933333 0.23918575 0.07038123
```

Este é um exemplo clássico do Paradoxo de Simpson. Quando se analisa por grupos, verifica-se que, afinal, não houve favorecimento aos candidatos do género masculino. Aconteceu que as candidatas à UCB tentaram, maioritariamente, entrar em departamentos mais competitivos, enquanto que os candidatos optaram, principalmente, por departamentos mais fáceis de entrar.

3. Considere o seguinte conteúdo de um ficheiro de nome *notas.dat*: [0.5 val.]

```
Nome-AM2-CEGI-DP1-Est1-ISC-Mrkt
Geraldo-12,1-14,4-15,2-11,7-10,0-15,5
Justino-8,2-6,6-12,3-9,9-13,1-16,7
Felisberta-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
Toninha-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.

Solução:

```
notas = read.table(notas.dat, header=T, sep="-", dec=",")
```

4. O objecto `ChickWeight` tem 578 linhas e 4 colunas com observações do efeito de diferentes dietas no crescimento de pintainhos. As variáveis disponibilizadas são:

weight vector numérico com o peso de cada pintainho, em grama.

Time número de dias passados, desde o nascimento, até à altura da medição do peso.

Chick `factor` ordenado que indica em que pintainho foi efectuada a medição.

Diet `factor` que indica o tipo de dieta (de 1 a 4) aplicado a cada pintainho.

(a) Calcule o peso médio dos pintainhos, para cada instante de tempo registado. Não utilize uma estrutura iterativa. [1 val.]

Solução:

```
cw = ChickWeight
pm = by(cw$weight, cw$Time, mean)
```

(b) Construa um gráfico que mostre o peso médio dos pintainhos ao longo do tempo. Coloque no eixo horizontal o tempo e no eixo vertical o peso. Respeite o intervalo de tempo do ensaio, que vai de 0 a 21 dias. [1 val.]

Se não resolveu a alínea anterior, considere o vector `pm`, com o peso médio em cada momento de medição:

```
pm = c(41, 49, 60, 74, 91, 108, 129, 144, 168, 190, 210, 219)
```

Solução:

```
plot(x = unique(cw$Time), y=pm, xlab='Tempo', ylab='Peso')
```

Pergunta	1	2	3	4	Total
Cotação	3.5	4	.5	2	10
Cotação obtida					