

---

INSTITUTO SUPERIOR DE AGRONOMIA  
ESTATÍSTICA E DELINEAMENTO – 2021-22  
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)`

```
[1] 47.87071
```

ou

```
> sqrt(var(precip))
```

```
[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 mínimo (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).

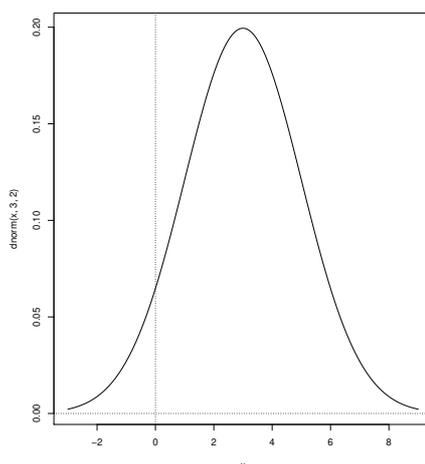
4. Neste exercício considera-se uma distribuição Normal, concretamente,  $\mathcal{N}(\mu=3, \sigma=2)$ .

- (a) Como indicado no enunciado, o comando `dnorm` devolve os valores da função densidade duma distribuição Normal. Os parâmetros da Normal são especificados pelos argumentos `mean` (que indica o valor de  $\mu$ ) e `sd` (que indica o desvio padrão  $\sigma$ ). Assim, o valor da função densidade duma  $\mathcal{N}(\mu=3, \sigma=2)$  no ponto 0 é dada por:

```
> dnorm(0, mean=3, sd=2)
[1] 0.0647588
```

A fim de traçar a curva densidade desta Normal, será preciso aplicar o comando `curve` à função `dnorm`. Os pontos  $x$  onde se calcula o valor da densidade irão percorrer um intervalo cujos extremos são especificados pelos argumentos `from` e `to`. O número de valores de  $x$  que se deseja usar é indicado pelo argumento `n`. Assim, o aspecto da função densidade da distribuição  $\mathcal{N}(\mu=3, \sigma=2)$  no intervalo  $]-3, 9[$ , usando  $n=1001$  valores de  $x$ , é dado da seguinte forma (os comandos `abline` servem para traçar os eixos, a ponteados devido ao argumento `lty=3`):

```
> curve(dnorm(x,3,2), from=-3, to=9, n=1001)
> abline(v=0, lty=3)
> abline(h=0, lty=3)
```



- (b) Começamos por assinalar que os acontecimentos  $X \leq 4$  e  $X > 4$  são complementares, logo  $P[X > 4] = 1 - P[X \leq 4]$ . Também, pelas propriedades das probabilidades tem-se  $P[-1 < X < 4] = P[X < 4] - P[X \leq -1]$ . Recorde-se ainda que, para variáveis aleatórias contínuas, a probabilidade de se tomar um único valor individual é nula. Logo,  $P[X < 4] = P[X \leq 4]$ .

- i. Usemos o comando `pnorm` para obter as três probabilidades pedidas:

```
> pnorm(4, mean=3, sd=2)
[1] 0.6914625
> 1-pnorm(4, mean=3, sd=2)
[1] 0.3085375
> pnorm(4, mean=3, sd=2) - pnorm(-1, mean=3, sd=2)
[1] 0.6687123
```

A fim de se poder usar as tabelas da Normal, torna-se necessário converter os acontecimentos cuja probabilidade é pedida em acontecimentos equivalentes envolvendo uma distribuição Normal Reduzida,  $\mathcal{N}(0,1)$ , a única para a qual existem tabelas. Sabemos que, se  $X \sim \mathcal{N}(\mu, \sigma)$ , então a variável  $Z = \frac{X-\mu}{\sigma}$  tem distribuição  $\mathcal{N}(0,1)$ . Logo,  $P[X \leq 4] = P[Z \leq \frac{4-3}{2}] = P[Z \leq 0.5] = \Phi(0.5)$ . Consultando as tabelas disponíveis na página da disciplina, verifica-se que  $\Phi(0.5) = 0.69146$ . Logo,  $P[X > 4] = 1 - P[X \leq 4] = 1 - 0.69146 = 0.30854$ . Analogamente,  $P[X \leq -1] = P[Z \leq \frac{-1-3}{2}] = P[Z \leq -2] = \Phi(-2)$ . Como indicado na nota das tabelas,  $\Phi(-z) = 1 - \Phi(z)$ , para qualquer número real  $z$ . Logo,  $\Phi(-2) = 1 - \Phi(2) = 1 - 0.97725 = 0.02275$ . Assim,  $P[-1 < X < 4] = P[X < 4] - P[X \leq -1] = \Phi(0.5) - \Phi(-2) = 0.69146 - 0.02275 = 0.66871$ .

- ii. A definição de quantil de ordem 0.975 de  $X \sim \mathcal{N}(\mu = 3, sd = 2)$  é um valor  $q$  tal que  $P[X \leq q] = 0.975$ . Usa-se o comando `qnorm` para obter quantis. Concretamente,

```
> qnorm(0.975, mean=3, sd=2)
[1] 6.919928
```

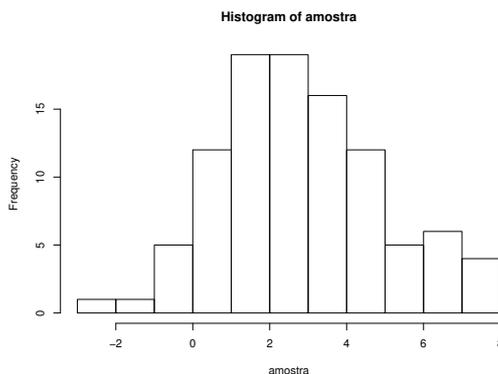
Para usar as tabelas, será necessário de novo procurar uma condição equivalente numa Normal Reduzida. Assim, procura-se o valor  $q^* = \frac{q-3}{2}$  tal que  $P[Z \leq q^*] = 0.975$ , sendo  $Z$  uma Normal Reduzida. Para identificar o valor  $q^*$  procuramos 0.975 no interior da tabela dum  $\mathcal{N}(0,1)$ . Esse valor encontra-se na linha correspondente a 1.9 e na coluna (que indica a casa decimal seguinte) 0.06. Logo,  $q^* = 1.96$ , ou seja,  $q = 3 + 2q^* = 6.92$ .

- iii. Usemos o comando `rnorm`:

```
> amostra <- rnorm(100, mean=3, sd=2)
```

Uma vez que o comando `rnorm` gera uma amostra (pseudo-)aleatória, o que se segue terá resultados diferentes de cada vez que se correr o comando anterior. Ou seja, os resultados seguintes não serão iguais aos que cada um de vocês obterá ao repetir estes comandos. O histograma é dado por:

```
> hist(amostra)
```



A média e variância amostrais são dadas por:

```
> mean(amostra)
[1] 2.858052
```

---

```
> sd(amostra)
[1] 2.135481
```

Naturalmente que estes valores não correspondem exactamente aos valores populacionais, nem o histograma é exactamente o duma Normal  $\mathcal{N}(\mu=3, sd=2)$ . De facto, estamos apenas perante uma *amostra*, escolhida de forma aleatória duma população  $\mathcal{N}(\mu=3, sd=2)$ . Esta é a situação que encontramos quando, na impossibilidade de analisar uma população inteira, apenas dispomos duma amostra dessa população.

- iv. Sabemos (ver formulário aqui) que o intervalo de confiança a  $(1 - \alpha) \times 100\%$  de confiança para  $\mu$  é dado por  $[\bar{x} - t_{\frac{\alpha}{2}(n-1)} \frac{s}{\sqrt{n}}, \bar{x} + t_{\frac{\alpha}{2}(n-1)} \frac{s}{\sqrt{n}}]$ . Tendo presente os valores  $\bar{x}=2.858052$ ,  $s=2.135481$  e  $n=100$ , falta apenas calcular o quantil de ordem 0.975 duma distribuição *t-Student* com parâmetro  $n-1=99$ ,  $t_{0.025(99)}$ , que é dada por:

```
> qt(0.975,99)
[1] 1.984217
```

Alternativamente, recorrendo às tabelas *t-Student* e usando o valor aproximado, correspondente aos 100 graus de liberdade, temos  $t_{0.025(99)} \approx 1.98397$ .

Substituindo (o valor exacto) na expressão, tem-se o seguinte intervalo a 95% de confiança para  $\mu$ :  $[2.434326, 3.281778]$ . Neste caso, o intervalo contém o valor exacto  $\mu=3$ . Recorde-se que o intervalo de confiança obtido será diferente para cada amostra de tamanho  $n=100$  seleccionada. O resultado teórico subjacente ao intervalo de confiança afirma que, se fosse extraído um grande número de amostras, o valor 3 estaria contido em 95% dos intervalos de confiança resultantes (mas não nos restantes 5%).

- v. Vejamos os cinco passos deste teste de hipóteses:

**Hipóteses:**  $H_0 : \mu = 3$  vs.  $H_1 : \mu \neq 3$ .

**Estatística do teste:**  $T = \frac{\bar{X} - \mu_{|H_0}}{\frac{s}{\sqrt{n}}} \sim t_{n-1}$ , se  $H_0$  verdade.

**Nível de significância:**  $\alpha = 0.05$ .

**Região Crítica (Bilateral):** Rejeitar  $H_0$  se  $|T_{calc}| > t_{\frac{\alpha}{2}(n-1)} = t_{0.025(99)} = 1.984217$ .

**Conclusões:** O valor calculado da estatística do teste é:  $T_{calc} = \frac{2.858052 - 3}{\frac{2.135481}{\sqrt{100}}} = -0.6647121$ .

Logo, não se rejeita a hipótese nula, ou seja, admite-se que  $\mu=3$ . Uma vez que sabemos ser esse o verdadeiro valor de  $\mu$ , este resultado é natural (e era previsível a partir do intervalo de confiança construído na alínea anterior). De novo, o valor calculado da estatística de teste,  $T_{calc}$  depende da amostra concreta que fôr extraída. Pode até acontecer que  $T_{calc}$  recaia na região de rejeição de  $H_0$ , mesmo sendo verdade que  $\mu=3$ . Aliás, pela construção do teste, será de esperar que isso aconteça em cerca de 5% das amostras, caso fosse extraído um grande número de amostras (recorde-se que o nível de significância  $\alpha$  é a probabilidade de cometer o Erro de Tipo I, ou seja a probabilidade de rejeitar  $H_0$  quando  $H_0$  é verdade).

- vi. No caso da Hipótese Nula ser agora  $\mu \geq 3$ , haverá que alterar também a Hipótese Alternativa e, por conseguinte, a natureza da região de rejeição. No entanto, o valor calculado da estatística mantém-se igual. Em concreto:

**Hipóteses:**  $H_0 : \mu \geq 3$  vs.  $H_1 : \mu < 3$ .

**Estatística do teste:**  $T = \frac{\bar{X} - \mu_{|H_0}}{\frac{s}{\sqrt{n}}} \sim t_{n-1}$ , se  $H_0$  verdade.

**Nível de significância:**  $\alpha = 0.05$ .

**Região Crítica (Unilateral esquerda):** Rejeitar  $H_0$  se  $T_{calc} < -t_{\alpha(n-1)} = -t_{0.05(99)} = -1.660391$ .

---

**Conclusões:** O valor calculado da estatística do teste é:  $T_{calc} = \frac{2.858052-3}{\frac{2.135481}{\sqrt{100}}} = -0.6647121$ .

Logo, não se rejeita a hipótese nula, ou seja, admite-se que  $\mu \geq 3$ .

- vii. A informação relativa ao intervalo de confiança e aos testes de hipóteses pode ser obtida no R através do comando `t.test`. Será necessário especificar o valor de  $\mu$  que se pretende testar. Por omissão, o comando assume que pretendemos efectuar um teste bilateral ao nível de significância  $\alpha=0.05$ . Eis o comando para o nosso caso:

```
> t.test(amostra,mu=3)
```

```
One Sample t-test
```

```
data: amostra
t = -0.66471, df = 99, p-value = 0.5078
alternative hypothesis: true mean is not equal to 3
95 percent confidence interval:
 2.434326 3.281777
sample estimates:
mean of x
 2.858052
```

Repare-se como o intervalo a 95% de confiança para  $\mu$  é dado automaticamente (tal como a média amostral  $\bar{x}$ ). Repare-se ainda como é dado o valor de prova (*p-value*) correspondente à probabilidade de se ter um valor tão, ou mais, extremo do que  $T_{calc}$ , no caso de ser verdade  $H_0 : \mu = 3$ . Concretamente, neste teste bilateral, o valor  $p=0.5078$  indicado é a probabilidade  $2 \times P[T > |T_{calc}|]$ , caso  $T \sim t_{n-1}$ :

```
> 2*(1-pt(0.66471,99))
[1] 0.5077814
```

Para se obter um teste unilateral, haverá que especificar o argumento `alternative`, no nosso caso com o valor `less`, uma vez que a hipótese alternativa é  $H_1 : \mu < 3$ :

```
> t.test(amostra,mu=3, alternative="less")
```

```
One Sample t-test
```

```
data: amostra
t = -0.66471, df = 99, p-value = 0.2539
alternative hypothesis: true mean is less than 3
95 percent confidence interval:
 -Inf 3.212625
sample estimates:
mean of x
 2.858052
```

Neste contexto, o comando produz um intervalo de confiança de tipo unilateral (matéria não dada nas disciplinas do ISA) e também um diferente valor de prova. Num teste com região crítica unilateral esquerda, o *p-value* define-se como  $p = P[T < T_{calc}]$ . No nosso exemplo:

```
> pt(-0.66471,99)
[1] 0.2538907
```

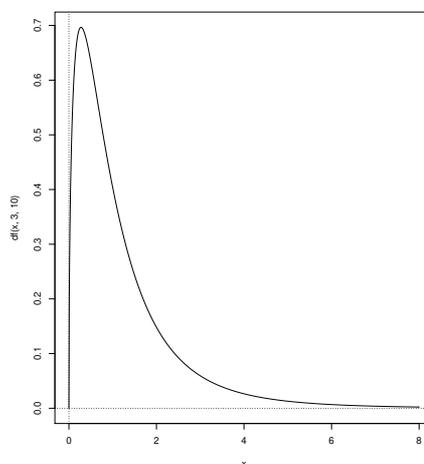
5. Neste exercício considera-se uma distribuição  $F(3, 10)$ .

- (a) Proceda-se como no Exercício 4:

```

> curve(df(x,3,10), from=0, to=8, n=1001)
> abline(v=0, lty=3)
> abline(h=0, lty=3)

```

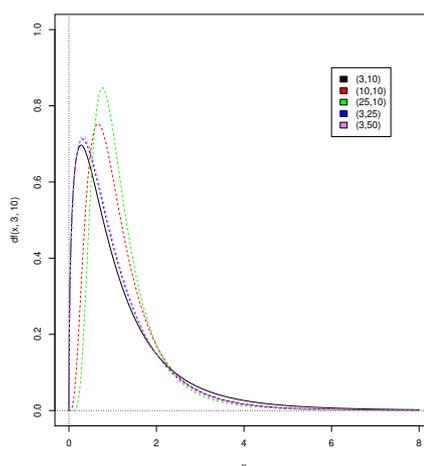


(b) Além da densidade  $F(3, 10)$  considerada na alínea anterior, vejamos algumas outras densidades de distribuições  $F$ , com diferentes valores dos parâmetros.

```

> curve(df(x,3,10), from=0, to=8, n=1001, ylim=c(0,1))
> abline(v=0, lty=3)
> abline(h=0, lty=3)
> curve(df(x,10,10), add=TRUE, n=1001, lty=2, col="red")
> curve(df(x,25,10), add=TRUE, n=1001, lty=2, col="green")
> curve(df(x,3,25), add=TRUE, n=1001, lty=2, col="blue")
> curve(df(x,3,50), add=TRUE, n=1001, lty=2, col="violet")
> legend(6,0.9, fill=c("black","red","green","blue","violet"),
+       c("(3,10)","(10,10)","(25,10)","(3,25)","(3,50)"))

```



(c) De novo para uma distribuição  $F(3, 10)$ .

- i. Usamos o comando `pf`, de forma semelhante ao que foi feito no Exercício 4 para a distribuição Normal. Repare-se que, como para qualquer outra distribuição contínua,

---

$P[X \leq 4] = P[X < 4]$ , sendo em ambos os casos o valor dado pelo primeiro dos comandos em baixo. Como para qualquer variável aleatória,  $P[X > 4] = 1 - P[X \leq 4]$ , sendo o valor dado pelo segundo comando seguinte.

```
> pf(4,3,10)
[1] 0.9586523
> 1-pf(4,3,10)
[1] 0.04134768
```

ii. A mediana é o quantil 0.5. Logo, obtém-se através do comando `qf`:

```
> qf(0.5,3,10)
[1] 0.8450806
```