

Visualização de dados e testes de hipóteses com R

– *uma breve abordagem prática* –

N. Sousa

ÍNDICE

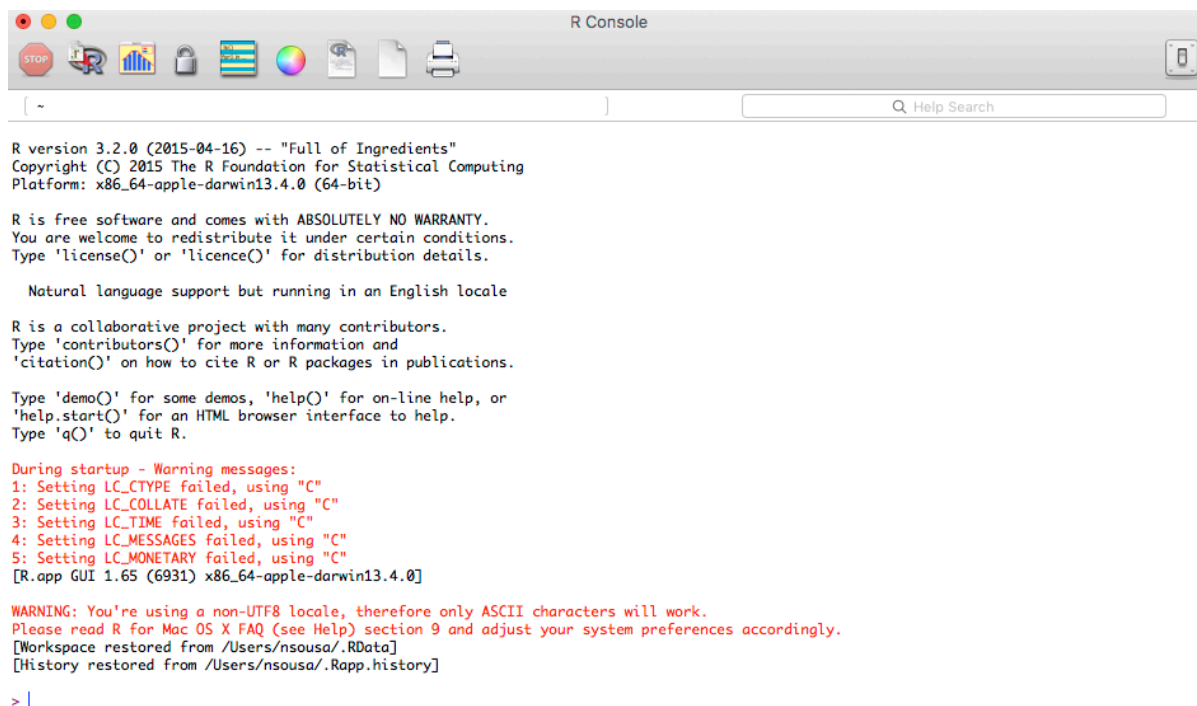
Instalação do R	3
Entrada de dados	4
Tabelas de frequências	6
Variáveis discretas	6
Variáveis contínuas	7
Construção manual de classes	9
Visualização de dados	11
Variáveis discretas	11
Diagrama de barras	11
Diagramas circulares.....	13
Variáveis contínuas	15
Histogramas.....	15
Diagramas circulares.....	17
Média e desvio-padrão amostrais	18
Média amostral	18
Desvio-padrão amostral	20
Testes de hipóteses	22
Filosofia de um teste de hipóteses	22
Teste à média de uma população	23
Validade de um teste à média	26
Teste à proporção	27
Análise de variância	28
Execução de uma ANOVA com R – caso prático	28
Validação de pressupostos	29
Formatação de dados e <i>data frames</i>	30
Tabela ANOVA e sua interpretação	32
Testes de comparações múltiplas.....	32
Exercícios	34

Instalação do R

O R é um poderoso software estatístico gratuito e *open source*. A sua versão base faz praticamente tudo o que é elementar e avançado, e tem suplementos para todo o tipo de temáticas mais específicas.

O R pode ser descarregado de <http://cran.r-project.org/> para os três sistemas operativos mais usados: Windows, Mac OS, Linux (várias distribuições). É só seguir as instruções na página-mãe do CRAN.

A instalação base do CRAN é do tipo linha-de-comandos, i.e. os commands são dados por escrita e não recorrendo a menus. Eis o aspeto da linha-de-comandos em Mac OS:



```
R version 3.2.0 (2015-04-16) -- "Full of Ingredients"
Copyright (C) 2015 The R Foundation for Statistical Computing
Platform: x86_64-apple-darwin13.4.0 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

During startup - Warning messages:
1: Setting LC_CTYPE failed, using "C"
2: Setting LC_COLLATE failed, using "C"
3: Setting LC_TIME failed, using "C"
4: Setting LC_MESSAGES failed, using "C"
5: Setting LC_MONETARY failed, using "C"
[R.app GUI 1.65 (6931) x86_64-apple-darwin13.4.0]

WARNING: You're using a non-UTF8 locale, therefore only ASCII characters will work.
Please read R for Mac OS X FAQ (see Help) section 9 and adjust your system preferences accordingly.
[Workspace restored from /Users/nsousa/.RData]
[History restored from /Users/nsousa/.Rapp.history]

> |
```

No que se segue, vai-se assumir que o estudante leu os outros recursos de estatística entretanto disponibilizados na página da UC e que, por conseguinte, está familiarizado com a terminologia estatística que se segue.

Entrada de dados

Existem essencialmente duas formas de entrar dados no R: manualmente, através da linha-de-comandos, ou carregando um ficheiro externo. Neste documento vamos abordar apenas a entrada manual de dados.

No R os dados são guardados sob a forma de *vetores*, ou seja, listas ordenadas de valores. Para exemplificar a entrada de dados, vamos recorrer a um exemplo.

Suponhamos que queremos estudar a pluviosidade mensal num ponto do país. Os dados recolhidos da estação meteorológica foram:

Mês	Pluviosidade (mm)	Mês	Pluviosidade (mm)	Mês	Pluviosidade (mm)
Janeiro	110	Maio	70	Setembro	42
Fevereiro	100	Junho	18	Outubro	89
Março	60	Julho	17	Novembro	108
Abril	80	Agosto	17	Dezembro	143

Para entrar estes dados, basta escrever na linha-de-comandos

```
> pluv = c(110, 100, 60, 80, 70, 18, 17, 17, 42, 89, 108, 143)
```

Fica assim criada uma variável, `pluv`, que é um vetor com os 12 valores indicados. O comando “`c`” significa: “combinar a sequência de valores que se segue num vetor”. Para vermos o valor que `pluv` tem, basta escrever `pluv` na linha-de-comandos:

```
> pluv
[1] 110 100 60 80 70 18 17 17 42 89 108 143
```

Se tiver havido um erro na entrada de dados, podemos substituir individualmente o elemento errado. P.ex. se o 5º elemento fosse 75 (em vez de 70), essa alteração seria feita com

```
> pluv[5] = 75
> pluv
[1] 110 100 60 80 75 18 17 17 42 89 108 143
```

O comando `pluv[5] = 75` significa, em linguagem corrente, “o 5º elemento de `pluv` passa a ser igual a 75”.

Também se pode sempre voltar a entrar todo o vetor `pluv`, o que é mais fácil usando as setas “up/down” para navegar na lista de comandos entrados anteriormente e, encontrado o comando `pluv = c(...)`, substituir o valor 70 por 75.

No restante deste documento vamos assumir que os dados são entrados de forma manual.

Note-se que o R suporta “copy-paste”, pelo que para entrar `pluv` no seu computador bastará copiar a linha azul acima para a linha-de-comandos e fazer paste.

A pluviosidade é uma variável estatística contínua, ou seja, uma variável que pode tomar qualquer valor numa escala. Vejamos agora um exemplo de variável estatística discreta, i.e. uma variável que só pode tomar uma gama fixa de valores, normalmente inteiros. P.ex. o n.º de irmãos que cada uma de 40 pessoas têm:

```
> irmaos =  
c(0,1,2,0,0,2,4,1,2,3,2,1,1,1,1,0,0,0,1,2,3,4,2,2,1,1,0,1,0,1,0,1,0,  
  2,1,1,2,0,1,1)  
[1] 0 1 2 0 0 2 4 1 2 3 2 1 1 1 1 0 0 0 1 2 3 4 2 2 1 1 0 1 0 1 0 1  
0 2 1 1 2 0 1 1
```

As variáveis `pluv` e `irmaos` vão servir-nos de ferramenta de trabalho para o resto destas breves notas, pelo que se sugere ao estudante que as entre no seu terminal R (com `pluv[5] = 75`).

Tabelas de frequências

A tabela de frequências é a forma mais simples de fazer a contabilidade dos dados recolhidos. Tal como na teoria geral de estatística, no R a criação destas tabelas depende se a variável em estudo é discreta ou contínua. Começemos com o caso mais simples, variáveis discretas.

Variáveis discretas

O comando `table` devolve uma tabela:

```
> table(irmaos)
irmaos
 0  1  2  3  4
11 16  9  2  2
```

Ou seja “11 pessoas têm zero irmãos, 16 pessoas têm 1 irmão, etc.”

Para organizar esta informação de uma forma mais fácil de ler, e também mais comum, podemos usar o comando `transform`:

```
> transform(table(irmaos))
  irmaos Freq
1      0   11
2      1   16
3      2    9
4      3    2
5      4    2
```

A coluna `Freq` pode ser passada a outros argumentos de `transform`. P.ex. se quisermos escrever uma 3ª coluna, com frequências relativas, bastará escrever

```
> transform(table(irmaos), FreqRel = Freq/40)
  irmaos Freq FreqRel
1      0   11  0.275
2      1   16  0.400
3      2    9  0.225
4      3    2  0.050
5      4    2  0.050
```

(Recordemos que foram as 40 pessoas a responder à questão do n.º de irmãos.) É necessário dar um nome, p.ex. `FreqRel`, ao cabeçalho da coluna, caso contrário o comando `transform` não imprimirá essa coluna (experimente!).

Se quisermos incluir frequências acumuladas, recorreremos à função `cumsum`:

```
> transform(table(irmaos), FreqRel = Freq/40, FreqAcum =
cumsum(Freq))
  irmaos Freq FreqRel FreqAcum
1      0   11  0.275      11
```

2	1	16	0.400	27
3	2	9	0.225	36
4	3	2	0.050	38
5	4	2	0.050	40

E, se quisermos ainda as frequências relativas acumuladas,

```
> transform(table(irmaos), FreqRel = Freq/40, FreqAcum =
cumsum(Freq), FreqRelAcum = cumsum(Freq)/40)
```

	irmaos	Freq	FreqRel	FreqAcum	FreqRelAcum
1	0	11	0.275	11	0.275
2	1	16	0.400	27	0.675
3	2	9	0.225	36	0.900
4	3	2	0.050	38	0.950
5	4	2	0.050	40	1.000

O R é uma linguagem muito flexível e existem muitas outras formas de obter tabelas, e de as formatar a gosto. A que apresentámos aqui é a mais simples e permite copy-paste para outras ferramentas como p.ex. uma tabela de processador de texto ou folha de cálculo (Word, Excel, Pages, Numbers, etc.). Este copy-paste não vai ser, no entanto, livre de problemas... vai ser preciso posteriormente usar as ferramentas de formatação para ajustar ao contexto.

Variáveis contínuas

Vejamos agora como tratar variáveis contínuas. Se tentarmos simplesmente correr o comando `table` sobre uma série de dados contínuos, vamos obter um resultado algo inútil:

```
> table(pluv)
```

pluv
17 18 42 60 75 80 89 100 108 110 143
2 1 1 1 1 1 1 1 1 1 1

Isto acontece porque variáveis contínuas não repetem valores. E quando repetem, é por falta de precisão na medição (se tivéssemos mais algarismos, os dois 17 acima poderiam ser p.ex. 17,1 e 17,3).

O tratamento correto de variáveis contínuas exige construir-se *classes*, i.e. intervalos de valores, nos quais vamos encaixar as medições observados. A primeira questão é quantas classes construir, e quais. Há várias regras para o fazer. Aqui vamos usar a regra mais simples, que é construir \sqrt{N} classes de igual amplitude, com N o número de valores da amostra (ou dimensão da amostra). Como normalmente uma raiz quadrada não é inteira, arredonda-se para o inteiro a seguir.

Para o caso de `pluv` temos então $\sqrt{12} \approx 3,46 \rightarrow 4$ classes. No entanto, como veremos, o R não vai conseguir sempre fazer a divisão em 4 classes automaticamente.

A tabela pode agora ser construída recorrendo à função `cut`, que divide o intervalo das medições em parcelas iguais:

```
> table(cut(pluv, breaks = pretty(pluv, n = 4)))
```

(0,50]	(50,100]	(100,150]
4	5	3

O parâmetro `breaks` indica o tipo de divisão que se quer, e a função `pretty` tenta encaixar `pluv` nas 4 classes desejadas, com limites dados por números redondos (bonitinhos, “pretty”).

A tradução do comando acima para linguagem corrente seria algo como “*construa uma tabela de `pluv`, dividindo o intervalo tanto quanto possível em 4 classes de limites redondos*”. Como vemos, o R construiu apenas 3 classes. Na próxima secção vamos ver como fazer se quiséssemos mesmo insistir nas 4 classes.

A notação $(a, b]$ significa intervalo aberto à esquerda e fechado à direita. Se quiséssemos aberto à direita e fechado à esquerda, como é comum em outras fontes na literatura, teríamos que acrescentar o parâmetro `right = FALSE` em `cut`: (necessárias letras maiúsculas)

```
> table(cut(pluv, breaks = pretty(pluv, n = 4), right = FALSE))
```

[0,50)	[50,100)	[100,150)
4	4	4

Para uma apresentação mais polida podemos novamente usar o comando `transform`:

```
> transform(table(cut(pluv, breaks = pretty(pluv, n = 4))))
```

	Var1	Freq
1	(0,50]	4
2	(50,100]	5
3	(100,150]	3

Se quisermos acrescentar mais frequências, relativas e acumuladas, teríamos:

```
> transform(table(cut(pluv, breaks = pretty(pluv, n = 4))), FreqRel = Freq/12, FreqAcum = cumsum(Freq), FreqRelAcum = cumsum(Freq)/12)
```

	Var1	Freq	FreqRel	FreqAcum	FreqRelAcum
1	(0,50]	4	0.3333333	4	0.3333333
2	(50,100]	5	0.4166667	9	0.7500000
3	(100,150]	3	0.2500000	12	1.0000000

O leitor atento terá notado que o nome da variável, `pluv`, foi misteriosamente substituído por `Var1`. Porque é que isso acontece é uma longa história.... dá para alterar, mas por agora não vale a pena.

Mais importante é notar que todas as classes construídas são, por omissão, intervalos abertos. Se porventura os dados tiverem valores que coincidam com os limites da 1ª ou última classe, pode acontecer que estes valores não sejam incluídos na tabela, o que levaria a erros de contagem. Imagine p.ex. que o mês de agosto não tinha pluviosidade:

```
> pluv2 = c(110, 100, 60, 80, 70, 18, 17, 0, 42, 89, 108, 143)
```



```
> transform(table(cut(pluv2, breaks = pretty(pluv2, n = 4))))
      Var1 Freq
1   (0,50]    3
2  (50,100]    5
3 (100,150]    3
```

Como vemos, o valor zero respeitante a agosto não foi incluído na tabela. Isto porque o intervalo da classe 1 é aberto à esquerda, i.e. só inclui valores *superiores* a zero.

Para termos a garantia de que todos os valores são tomados em conta, mesmo os que sejam iguais aos limites de classes, basta adicionar ao comando `cut` o parâmetro `include.lowest=TRUE`:

```
> transform(table(cut(pluv2, breaks = pretty(pluv2, n = 4),
include.lowest = TRUE)))
      Var1 Freq
1   [0,50]    4
2  (50,100]    5
3 (100,150]    3
```

E agora já todos os valores são incluídos. Esta cautela é especialmente importante quando os dados são contínuos, mas estão apresentados como inteiros, i.e. sem casas decimais.

Construção manual de classes

A função `pretty` tem a vantagem de gerar automaticamente classes com limites que são números redondos, mas tem, como vimos, a desvantagem de nem sempre permitir o número de classes que queremos. Se quiséssemos p.ex. 6 classes (em vez das 4 recomendadas pela regra da \sqrt{N}), bastaria fazer `n = 6` em `pretty`:

```
> transform(table(cut(pluv, breaks = pretty(pluv, n = 6))))
      Var1 Freq
1   (0,20]    3
2  (20,40]    0
3  (40,60]    2
4  (60,80]    2
5  (80,100]   2
6 (100,120]   2
7 (120,140]   0
8 (140,160]   1
```

O melhor que o R conseguiu automaticamente foi expandir as anteriores 3 classes em 6 (e não 6). Isto teve como consequência que duas classes ficaram desertas de valores observados, o que deve ser evitado.

Em todo o caso, o resultado da geração automática das classes é normalmente satisfatório. Se quisermos mesmo 4 classes, isso é possível, mas teremos de as construir manualmente. Para tal basta, no parâmetro `breaks`, dizer quais são exatamente os limites que se pretende, e isso exige alguma reflexão por parte do utilizador.

Uma possibilidade é p.ex. considerar pluviosidade de 0 a 160 mm e dividir isto em 4 intervalos iguais, ou seja, colocar quebras (breaks) em 0, 40, 80, 120, 160. O comando R para isto é indicar no parâmetro `breaks` as quebras, sob a forma de vetor, usando o comando `c` (que, recordemos, significa “combine into vector”):

```
> transform(table(cut(pluv, breaks = c(0,40,80,120,160))))  
      Var1 Freq  
1   (0,40]    3  
2  (40,80]    4  
3  (80,120]    4  
4 (120,160]    1
```

E agora já funciona como queremos. A desvantagem é que, como dito, a geração das classes não foi automática e o utilizador teve de olhar para os dados para pensar como construir essas classes.

Visualização de dados

Tratada a questão das tabelas, vamos agora ver como visualizar em gráficos os dados. Os tipos de gráfico mais usados são os diagramas de barras, os histogramas e os diagramas circulares. Novamente aqui o R tem comandos diferentes, consoante as variáveis sejam discretas ou contínuas.

Variáveis discretas

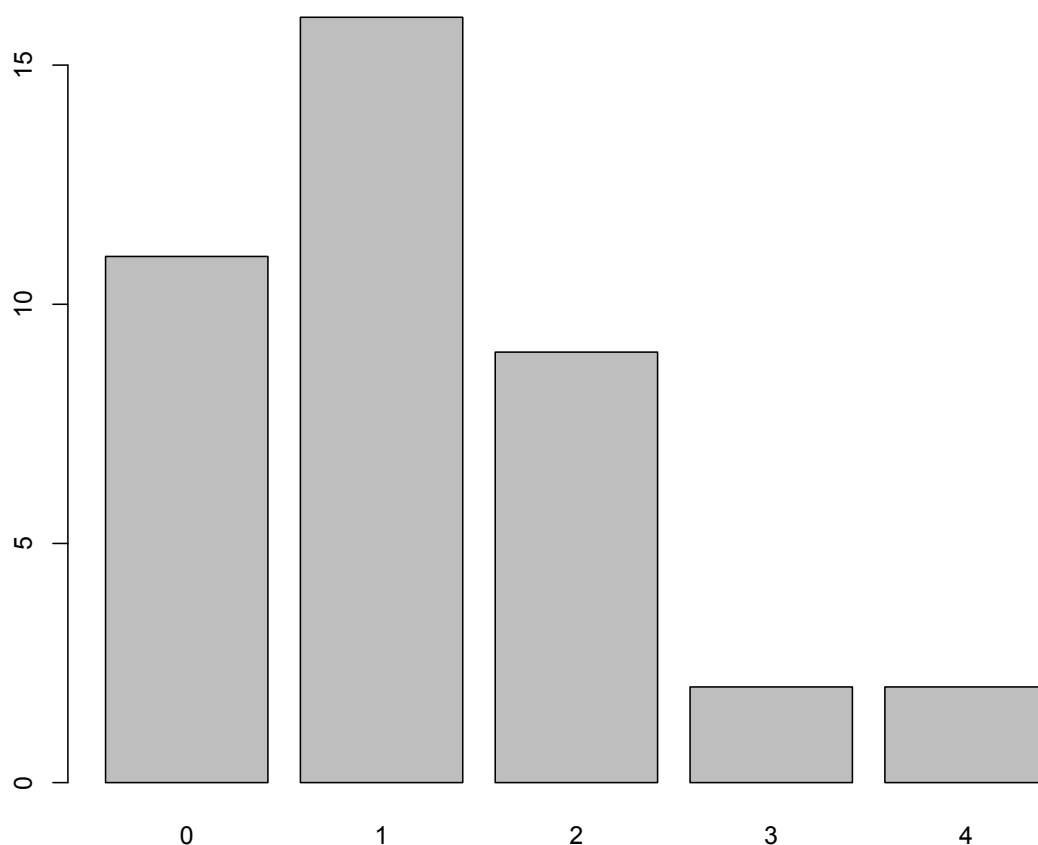
Para este tipo de variável podemos fazer um diagrama de barras ou um diagrama circular.

Diagrama de barras

Começemos por um diagrama de barras. O comando R fazer um tal diagrama é `barplot`, e a forma de o usar é

```
> barplot(table(irmaos))
```

que gera, numa janela à parte, um gráfico como o abaixo:



No comando acima, o `table` é essencial. O comando direto `barplot(irmaos)` seria incorretamente interpretado pelo R e constituiria um gráfico sem pés nem cabeça (experimente!).

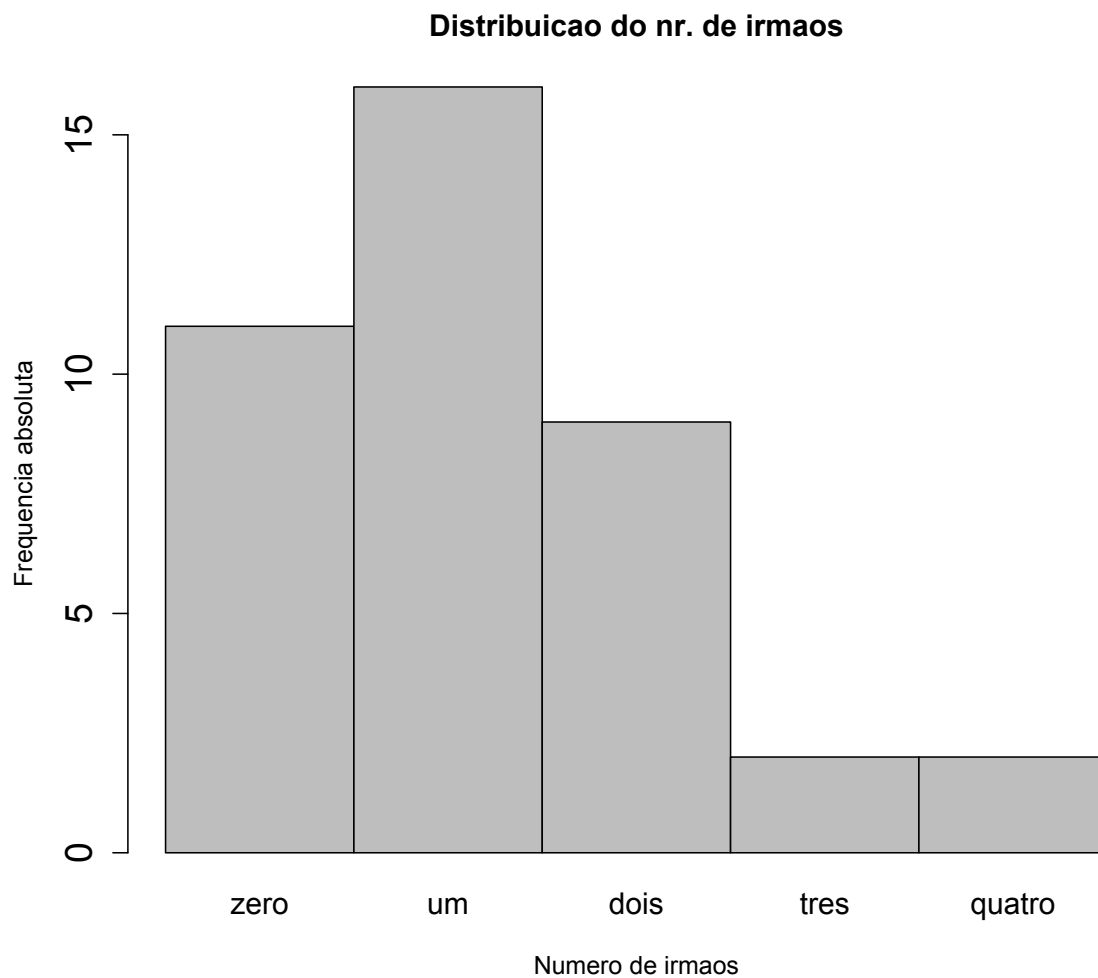
O diagrama acima é algo escasso em informação. Para o embelezar há que recorrer aos parâmetros que o comando `barplot` permite. P.ex.:

<code>space</code>	Controla o espaçamento entre barras.
<code>names.arg</code>	Permite substituir o nome das barras (0,1,2,3,4) por outra coisa qualquer. P.ex. para colocar zero, um, dois, etc., faz-se <code>names.arg=c("zero", "um", "dois", "tres", "quatro")</code> . As aspas significam que o que está entre elas é texto, e são essenciais. Caso contrário o R pensará que são variáveis, cujo valor não estará definido (e dará erro). Evitar também colocar acentos no texto, dado que o tratamento correto destes depende um pouco da versão do R e pode gerar problemas.
<code>main</code>	Dá um título ao diagrama.
<code>xlab, ylab</code>	Dá nomes aos eixos horizontal e vertical.
<code>cex.axis</code>	Controla o tamanho dos números no eixo vertical.
<code>cex.names</code>	Controla o tamanho dos números/letras no eixo horizontal.

E há mais parâmetros (p.ex. cores, sombreados das barras, etc.). Consultar o help file do R. Ver também <http://www.stat.columbia.edu/~tzheng/files/Rcolor.pdf> para uma lista de nomes de cores.

Um exemplo com todos estes parâmetros:

```
> barplot(table(irmaos), space = 0, names.arg =
c("zero","um","dois","tres","quatro"), main = "Distribuicao do nr.
de irmaos", xlab = "Numero de irmaos", ylab = "Frequencia
absoluta", cex.axis = 1.5, cex.names = 1.2)
```

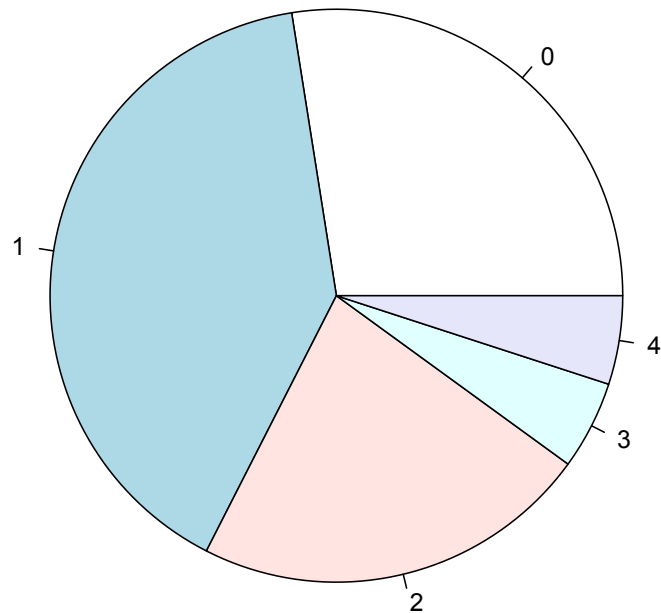


Diagramas circulares

O comando para diagramas circulares (“piechart”) é pie:

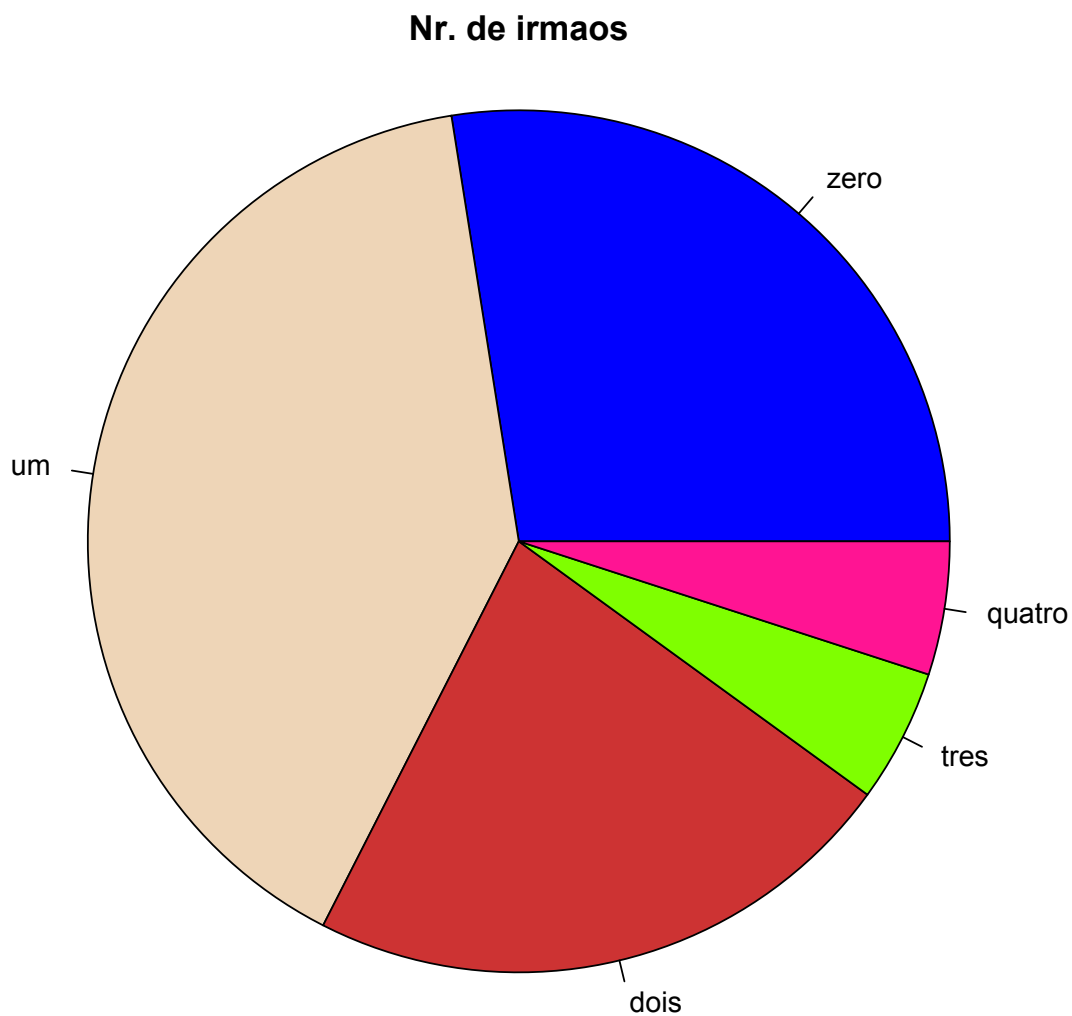
```
> pie(table(irmaos))
```

que devolve:



Também aqui se pode embelezar o diagrama. Um exemplo:

```
> pie(table(irmaos), labels =  
c("zero","um","dois","tres","quatro"), radius = 1.05,  
col=c("blue1","bisque2","brown3","chartreuse1","deeppink1"), main =  
"Nr. de irmaos")
```



Variáveis contínuas

No caso de variáveis contínuas vamos ver como fazer um histograma (o equivalente ao diagrama de barras de uma variável discreta) e o diagrama circular.

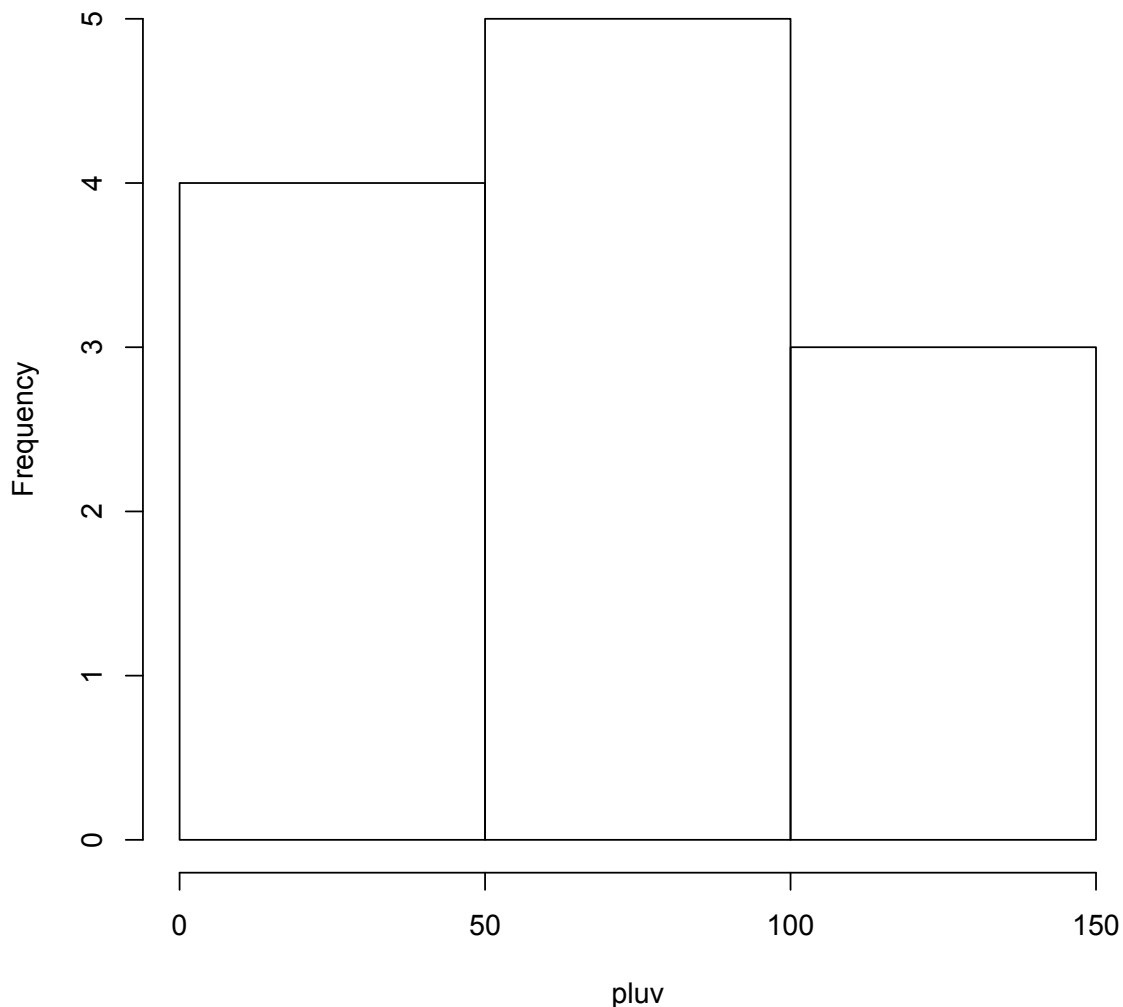
Histogramas

O comando R para um histograma é `hist`. O histograma deve ter as mesmas classes que se usou quando se fez a tabela de frequências. Ou seja, deve-se incluir um parâmetro `breaks` igual ao que se usou na tabela. A vantagem é `hist` assume algumas coisas por omissão. Bem, o melhor é ver um exemplo:

```
> hist(pluv, breaks = 4)
```

gera-nos

Histogram of pluv



Note-se que:

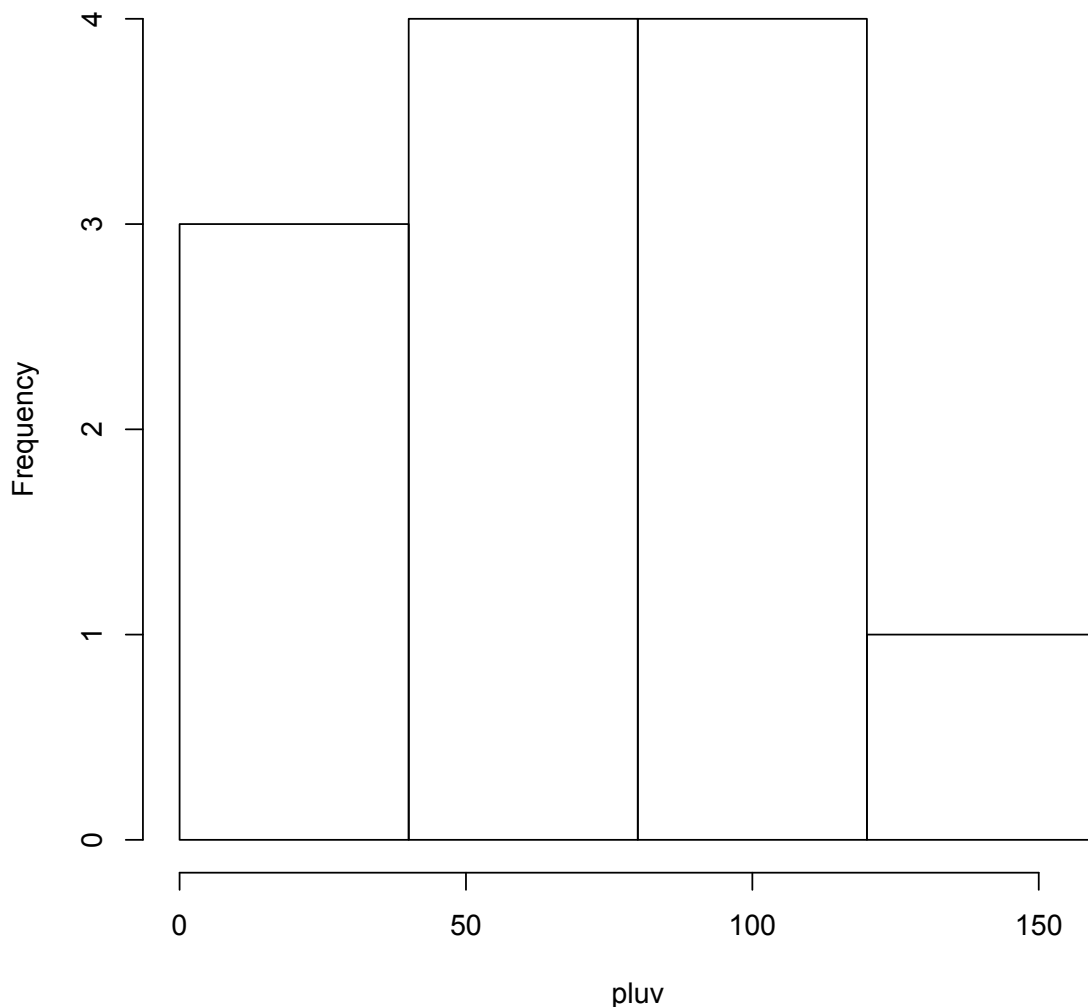
1. Não foi preciso dar indicação de divisão (`cut`). Bastou indicar a variável a ser representada e `hist` assume que esta tem necessariamente de ser dividida.
2. Também não foi preciso escrever `breaks = pretty(pluv, n = 4)`, tendo bastado `breaks = 4`. O comando `hist` simplesmente invoca, por omissão, `pretty` com parâmetro 4.

Se quiséssemos exatamente 4 classes teríamos, tal como no caso da tabela de frequências, que indicar explicitamente os limites:

```
> hist(pluv, breaks = c(0,40,80,120,160))
```

com o qual obteríamos

Histogram of pluv

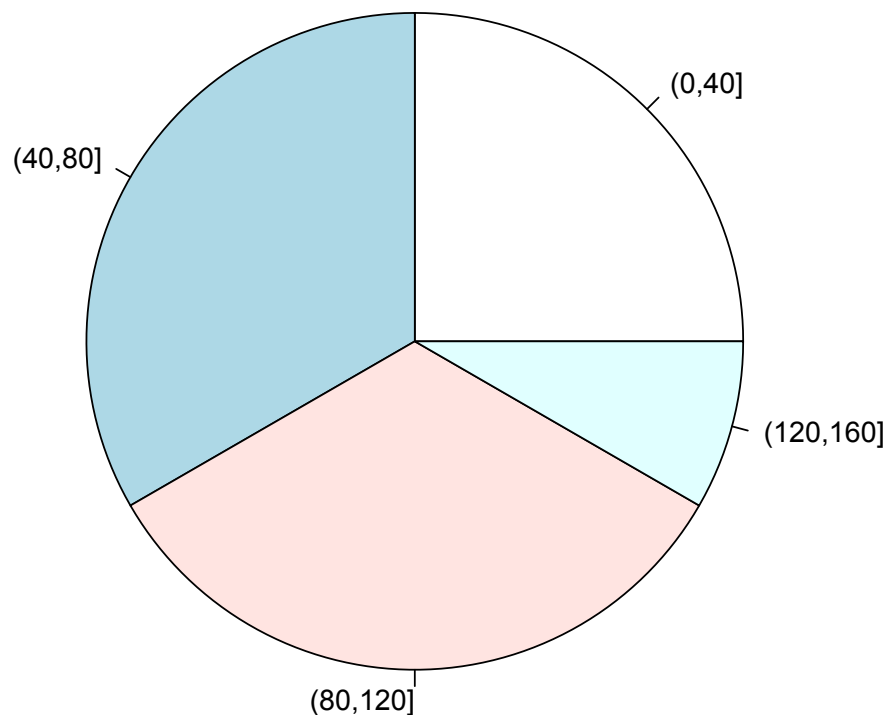


O comando `hist` partilha com `barplot` alguns parâmetros, como p.ex. `xlab`, `ylab`, `main`, etc. Tal como nas tabelas de frequência, para usar intervalos fechados à esquerda e abertos à direita o parâmetro é `right = FALSE`.

Diagramas circulares

O comando é o mesmo da variável discreta, `pie`. No entanto, ao contrário de `hist`, `pie` não herda valores por omissão e é necessário dar-lhe todas essas indicações. Na prática o diagrama circular correspondente ao histograma acima é obtido com o seguinte comando:

```
> pie(table(cut(pluv, breaks = c(0,40,80,120,160))))
```



Novamente, todo o tipo de etiquetas e embelezamento é possível.

Média e desvio-padrão amostrais

A média e o desvio-padrão são duas grandezas que resumem, num número apenas, informação sobre os dados recolhidos. A média é uma medida de localização, no sentido em que nos diz onde está localizada uma determinada característica dos dados (já vamos ver qual), e o desvio-padrão é uma medida de dispersão, uma vez que nos indica o quão dispersos os dados estão.

Média amostral

Todos nós temos uma noção intuitiva do que é uma média. Matematicamente, a média é o valor ao qual os dados mais se aproximam e é representada quase universalmente pelo símbolo \bar{x} . A média pode ser calculada pela fórmula geral:

$$\bar{x} = \frac{1}{N} \sum_{i=1}^N x_i$$

Nesta expressão, que é válida para qualquer tipo de variável (discreta ou contínua), N é a dimensão da amostra e o símbolo “ Σ ”, que se designa por “somatório”, não é senão a soma de todos os valores da amostra (x_i). A letra i é um índice: uma mera etiqueta que distingue os valores observados da variável estatística em estudo.

Se dissermos que o x_1 é o primeiro dos valores observados, x_2 o segundo, x_3 o terceiro e assim por diante, então em linguagem corrente a expressão “ $\sum_{i=1}^N x_i$ ” significa: “pegue no 1º valor observado, some-lhe o segundo, o terceiro, e assim por diante até ao último”. No caso de `pluv` temos $x_1 = 110$, $x_2 = 100$, $x_3 = 60$, etc. e a expressão para a média torna-se

$$\begin{aligned} \bar{x} = \frac{1}{N} \sum_{i=1}^N x_i &\Leftrightarrow \bar{x} = \frac{1}{12} (110 + 100 + 60 + 80 + 75 + 18 + 17 + 17 + 42 + 89 + 108 + 143) \\ &= \frac{859}{12} = 71,58333 \dots \approx 71,6 \end{aligned}$$

No R a média é dada pelo comando `mean`:

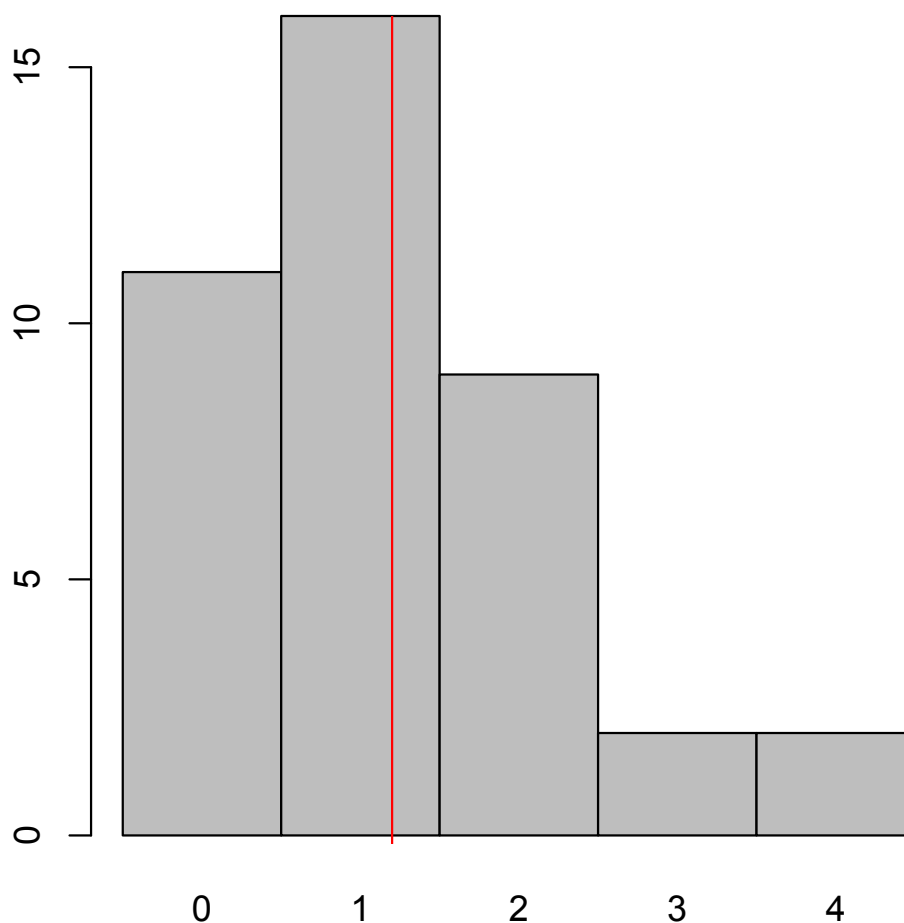
```
> mean(pluv)
[1] 71.58333
```

Para o caso dos irmãos temos:

```
> mean(irmaos)
[1] 1.2
```

Por vezes na literatura vemos outras expressões para o cálculo da média (média pesada, média para dados em classes, etc.). Essas expressões referem-se a situações em que os dados estão *agregados* em categorias ou classes. Quando os dados estão *desagregados*, ou seja, quando são considerados um-a-um (que é como os entramos no R), a fórmula acima é suficiente.

Uma interpretação curiosa da média é que esta é como que um fiel da balança. Ou seja, se dividirmos um histograma ou um diagrama de barras em partes esquerda e direita, a média é o ponto de equilíbrio desse diagrama:



Este é o diagrama de barras de `irmaos`. A média é a linha a vermelho. Se imaginarmos este diagrama como sendo sólido e o equilibrarmos na base exatamente no ponto da média, ele manter-se-á em equilíbrio sem tombar para nenhum dos lados!

Desvio-padrão amostral

O desvio-padrão de uma amostra mede, como dito acima, a dispersão dos dados de uma amostra. Tem como símbolo “*s*” e é dado pela expressão:

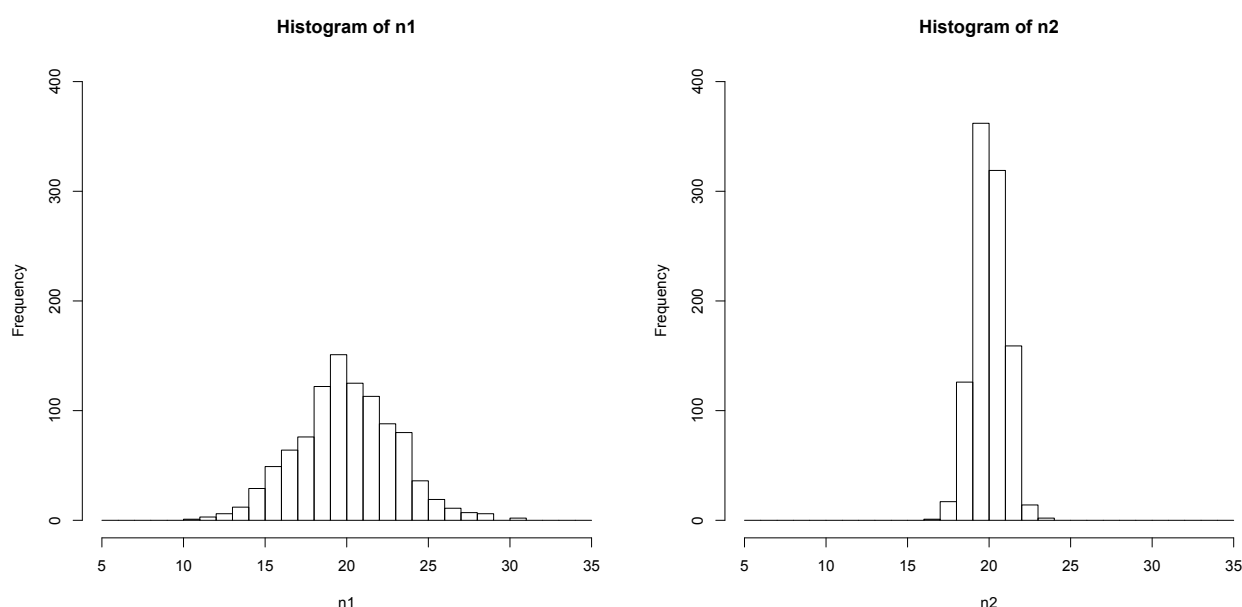
$$s = \sqrt{\frac{1}{N-1} \sum_{i=1}^N (x_i - \bar{x})^2}$$

A fórmula é um pouco mais complicada de entender do que a da média, mas no R basta-nos usar o comando `sd` para o obter o seu valor sem esforço:

```
> sd(irmaos)
[1] 1.066987
> sd(pluv)
[1] 41.48923
```

Como o desvio-padrão de `pluv` é maior do que o de `irmaos`, pode-se pensar que há mais dispersão em `pluv` que em `irmaos`. Não é bem assim: o valor em si não tem significado. Só o ganha quando o comparado com o resto dos dados.

Para perceber isto melhor, vamos recorrer a uma interpretação gráfica. Os dois histogramas abaixo foram gerados a partir de 1000 observações de variáveis estatísticas com média 20 e desvios diferentes. No caso da esquerda, a variável tem desvio-padrão 3, no da direita tem desvio-padrão 1.



O gráfico da esquerda representa a variável com *maior* desvio-padrão. O seu pico é bem mais esbatido e disperso do que o do gráfico da direita, que evidencia um pico muito mais pronunciado e bem definido, fruto de um *menor* desvio-padrão, logo *menor* dispersão. É esse o significado de desvio-padrão: quanto maior for, por comparação com o resto dos dados, mais dispersos vão estar esses dados. Quanto menor for, menos dispersos estarão, i.e. estarão mais concentrados em torno da média.

Uma última interpretação que se pode fazer do desvio-padrão é a seguinte:

“Numa amostra, aproximadamente 2/3 dos dados vão estar no intervalo entre $\bar{x} - s$ e $\bar{x} + s$.”

Faz-se notar, no entanto, que esta interpretação tem um regime de validade algo limitado.

Testes de hipóteses

Um teste de hipótese é uma técnica estatística cujo intuito é verificar se uma dada amostra de dados é, ou não, compatível com uma hipótese feita sobre a população que lhe deu origem.

Exemplos:

Q1. “Será plausível que, em média, os cidadãos portugueses tenham 2 irmãos?”

Q2. “Atendendo a que a média mensal de pluviosidade em Portugal é de 70,6 mm, terá o ano de *pluv* sido excepcionalmente chuvoso?”

Como veremos, o conceito de teste de hipóteses vai-nos permitir obter respostas a estas questões. Esta resposta poderá não ser, no entanto, peremptória. Tudo dependerá da evidência estatística que a técnica usada nos devolver.

Usar dados de uma amostra para tentar dizer algo sobre a população que lhes deu origem é um importante ramo da estatística, chamado *inferência estatística*, e o teste de hipóteses uma das suas principais ferramentas. Vejamos então do que se trata.

Filosofia de um teste de hipóteses

Um teste de hipóteses coloca lado-a-lado duas hipóteses sobre a população que deu origem à amostra de dados que temos à disposição. Uma hipótese inicial, ou *hipótese nula*, e uma *hipótese alternativa*. Estas são normalmente designadas por H_0 e H_1 respetivamente e referem-se a uma característica de uma população. Desta população é retirada uma amostra, cuja informação será tratada. Do resultado desse tratamento vamos encontrar evidência para se rejeitar, ou não, a hipótese nula. Caso haja rejeição da hipótese nula, o investigador deve daí em diante considerar na sua pesquisa a hipótese alternativa.

Mas vejamos o que isto significa na prática.

Consideremos o primeiro dos exemplos acima. Neste exemplo a população são todos os cidadãos portugueses, da qual temos uma amostra à disposição: *irmãos*. Em Estatística é comum designar-se a média de uma população pela letra μ (ao contrário da média da amostra, que, como vimos, designamos por \bar{x}). A hipótese nula colocada pela questão Q1 pode-se escrever, em linguagem matemática, por

$$H_0: \mu = 2$$

A hipótese alternativa terá de ser algo diferente disto. Há três possibilidades:

$$H_1: \mu \neq 2, \quad H_1: \mu < 2, \quad H_1: \mu > 2$$

Ou seja, podemos assumir, em contraste com a hipótese nula “média de irmãos = 2”, que essa média possa, em alternativa, ser maior que 2, menor que 2 ou simplesmente diferente de 2.

Levanta-se então uma questão: qual destas hipóteses devemos considerar para hipótese alternativa? Aqui devemos ter cuidado. Se o leitor espreitar umas páginas atrás, verá que a média da amostra é

$\bar{x} = 1,2$ irmãos. Isto poderá levá-lo a pensar que a hipótese alternativa a considerar deva ser “média de irmãos < 2 ”. Este raciocínio é, no entanto, enviesado porque só sabemos a média da amostra *a posteriori*, i.e. depois de a obter. Por outras palavras, terei eu, antes de perguntar às 40 pessoas de irmãos, alguma razão para desconfiar que a verdadeira média de irmãos seja menor que 2? Em princípio não. Quando não temos nenhuma indicação *a priori* de tendência, a hipótese alternativa deve ser enunciada como “diferente”. É só quando temos alguma suspeita de tendência que devemos considerar a alternativa como “menor” ou “maior”.

No caso da Q1 devemos, pois, considerar $H_1: \mu \neq 2$. Não quer dizer que não se possa considerar $H_1: \mu < 2$ ou $H_1: \mu > 2$; apenas que se o fizermos, devemos ter uma razão justificativa.

Resumindo: o teste de hipóteses levantado por Q1 pode ser formalmente descrito por

$$H_0: \mu = 2 \quad vs \quad H_1: \mu \neq 2$$

Existem formas alternativas de se enunciar matematicamente um teste de hipóteses. No entanto, em todas estas formas, a hipótese nula contém sempre uma igualdade e a hipótese alternativa sempre uma desigualdade. Testes com a alternativa escrita como “diferente” dizem-se *bilaterais*. Testes com alternativa “menor” designam-se por *unilaterais esquerdos*, e com alternativa “maior” *unilaterais direitos*.

Vejamos agora o teste que se pode enunciar a partir de Q2. Este pode ser formalmente descrito como p.ex.

$$H_0: \mu = 70,6 \quad vs \quad H_1: \mu \neq 70,6$$

Como exemplo de um teste unilateral, podemos pegar na Q2 e reescrevê-la:

Q2b. “A média mensal de pluviosidade em Portugal é de 70,6 mm. Poderão as mudanças climáticas terem diminuído a pluviosidade média?”

Neste caso, porque queremos testar uma possível tendência, o teste teria representação formal

$$H_0: \mu = 70,6 \quad vs \quad H_1: \mu < 70,6$$

E `pluv` seria a amostra a usar para executar o teste. (Na verdade, uma amostra com média amostral no sentido contrário à alternativa não irá nunca levar a rejeição da hipótese nula, mas isso é outra história.)

Vejamos agora como é que podemos usar o R para executar estes testes.

Teste à média de uma população

Os testes de hipóteses enunciados na secção anterior são, todos eles, testes à média de uma população. A teoria matemática que permite executar um teste é algo complicada e está fora do âmbito deste texto. Assim, vamos limitar-nos a indicar como o fazer no R.

Tendo uma amostra no R, o comando `t.test` permite realizar-se um teste à média, baseado nessa amostra. Para Q1 vem

```
> t.test(irmaos, mu = 2)

One Sample t-test

data:  irmaos
t = -4.742, df = 39, p-value = 2.817e-05
alternative hypothesis: true mean is not equal to 2
95 percent confidence interval:
 0.858761 1.541239
sample estimates:
mean of x
      1.2
```

Em primeiro lugar vejamos os parâmetros do comando: `irmaos` indica a amostra a usar e `mu = 2` que esse é o valor da média da população a testar.

Vejamos agora o output. O fulcral no output é o *valor de prova*, `p-value`. Para compreender este número, consideremos o seguinte raciocínio: mesmo que a média da população seja realmente $\mu = 2$, nem todas as amostras dela tiradas vão ter média amostral $\bar{x} = 2$. Como a variável em estudo é aleatória, umas amostras irão ter mais que 2, outras menos que 2. Ou seja, é normal haver algum desvio. O que não é normal é que esse desvio seja grande demais! Ora o `p-value` é o número que nos diz a probabilidade de, sendo H_0 verdadeira, o desvio entre a média amostral e a média da população ser igual ou superior ao observado.

Para o caso acima, o `p-value` é de $2,817 \times 10^{-5}$, i.e. 0,002817%. Ou seja, se $\mu = 2$ a probabilidade de uma amostra tirada dessa população apresentar uma média tão pequena quanto a observada ($\bar{x} = 1,2$) ou menos essa é *extremamente baixa*. Assim, é de suspeitar que a verdadeira média da população não seja $\mu = 2$, mas sim algo diferente (neste caso menor que 2), ou seja, é de *rejeitar* a hipótese nula e considerar a hipótese alternativa.

Para se perceber bem este ponto vamos gerar aleatoriamente algumas amostras com 40 elementos de uma população com $\mu = 2$ e gerar o `p-value` do teste à média associado:

Am.	média amostral	p-value
1	2.136109	43%
2	2.051206	80%
3	1.856278	31%
4	1.753305	14%
5	1.885642	47%

Como vemos, nas 5 amostras acima, a média amostral oscilou um pouco, mas nunca se afastou muito de 2 e, conseqüentemente, os `p-values` nunca ficaram abaixo dos 10%. Ou seja, em nenhum dos casos houve evidência estatística para rejeitar $\mu = 2$. Agora, quando aparece uma amostra com média amostral 1,2 e `p-value` 0,0028% é pouco provável que ela seja oriunda de uma população $\mu = 2$. Pode acontecer, mas é pouco provável e deve-se rejeitar essa hipótese.

Tipicamente a fronteira entre rejeição e não-rejeição situa-se entre os 1 e 10% de p-value. Ou seja, para p-values abaixo de 1%, é comum rejeitar-se sempre a hipótese nula. Para valores acima de 10% não se costuma rejeitar. Entre 1 e 10% estamos na chamada “zona cinzenta”, em que o julgamento de rejeição ou não fica ao critério do decisor. É por esta razão que se disse que o resultado de um teste de hipóteses nem sempre é peremptório.

Vejamos agora o teste Q2. O output R é

```
> t.test(pluv, mu = 70.6)

One Sample t-test

data:  pluv
t = 0.082102, df = 11, p-value = 0.936
alternative hypothesis: true mean is not equal to 70.6
95 percent confidence interval:
 45.22234 97.94433
sample estimates:
mean of x
 71.58333
```

O p-value de 93,6% indica claramente não-rejeição de H_0 . Ou seja, não há razão para rejeitar a hipótese de que o ano de `pluv` tenha sido um ano de pluviosidade normal.

Quanto a Q2b temos:

```
> t.test(pluv, mu = 70.6, alternative = "less")

One Sample t-test

data:  pluv
t = 0.082102, df = 11, p-value = 0.532
alternative hypothesis: true mean is less than 70.6
95 percent confidence interval:
 -Inf 93.09248
sample estimates:
mean of x
 71.58333
```

O parâmetro `alternative = "less"` indica que o teste deve ser unilateral esquerdo. Para um teste lateral direito deve-se usar `alternative = "greater"`. A sintaxe mesmo de ser esta. Sintaxes aparentemente mais naturais dão erro...

```
> t.test(pluv, mu < 70.6)
Error in t.test.default(pluv, mu < 70.6) : object 'mu' not found
```

Se não se usar o parâmetro `alternative`, o R faz um teste bilateral.

Validade de um teste à média

Como vimos, é relativamente simples fazer um teste à média com o R. No entanto, estatisticamente existem alguns pressupostos teóricos que se devem cumprir para que se possa confiar no resultado desse teste (p-value).

Para amostras pequenas, $N \leq 30$, a população em estudo deve ser proveniente de uma distribuição *gaussiana*, ou *normal*. A distribuição normal é uma distribuição contínua, usualmente associada a coisas como características físicas (peso, altura, etc.) ou erros de medição.

Para amostras maiores, $N > 30$, a população pode ser qualquer.

Nos nossos dois casos temos que `pluv` é uma amostra pequena ($N = 12$) e `irmaos` uma amostra grande ($N = 40$). Ou seja, para `irmaos` podemos confiar no resultado dos testes que fizemos. E para `pluv`? Como podemos verificar que `pluv` é (pelo menos aproximadamente) uma distribuição normal? Verificar o pressuposto de normalidade pode ser feito através de um teste de hipóteses preliminar, o teste de Shapiro-Wilk. O comando R para executar este teste é `shapiro.test`:

```
> shapiro.test(pluv)

Shapiro-Wilk normality test

data:  pluv
W = 0.93505, p-value = 0.4368
```

Neste teste a hipótese nula é “a distribuição de `pluv` é normal” e a hipótese alternativa é “a distribuição não é normal”. O valor de prova 44% diz-nos que não há evidência estatística para rejeitar a hipótese nula, pelo que podemos considerar que `pluv` provém efetivamente de uma distribuição normal. Assim sendo, os resultados obtidos anteriormente para `pluv` estão validados.

Se porventura o p-value do teste de Shapiro-Wilk fosse baixo (p.ex. $< 1\%$), então seria necessário recolher mais dados de `pluv` para se chegar a uma amostra com pelo menos 30 valores.

Como curiosidade, abaixo temos o teste de Shapiro-Wilk para `irmaos`:

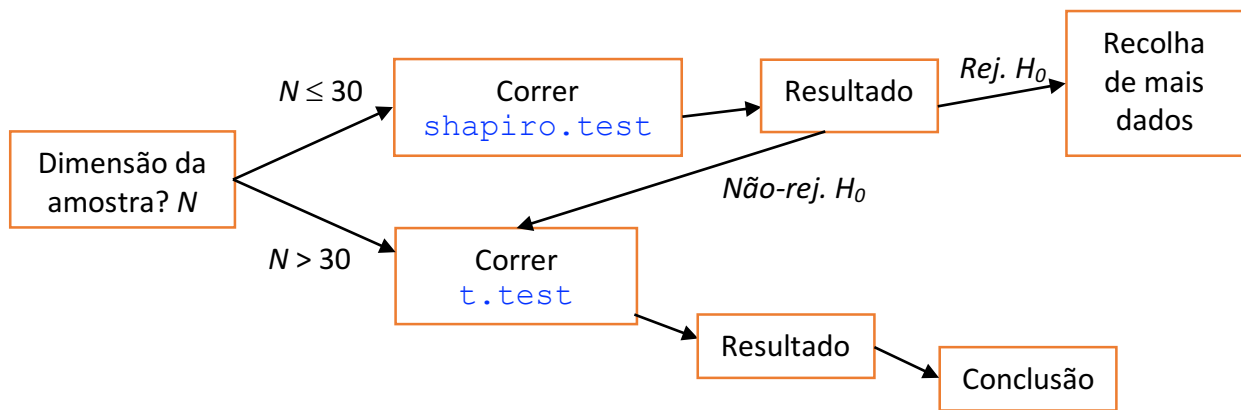
O valor de prova de 0,012% era esperado, dado que a distribuição de `irmaos` não é contínua, mas sim discreta.

```
> shapiro.test(irmaos)

Shapiro-Wilk normality test

data:  irmaos
W = 0.8557, p-value = 0.0001241
```

Podemos resumir a filosofia de validação de um teste à média na seguinte chave dicotómica:



Teste à proporção

Uma outra situação comum é o teste a uma proporção. Vejamos uma situação típica: lanço uma moeda ao ar 100 vezes, tendo saído 60 vezes cara (logo 40 vezes coroa). Estará a moeda viciada? O teste à proporção é a ferramenta estatística que ajuda a ajuizar situações como esta.

Este tipo de teste reporta-se a situações em que uma experiência, que só tem dois resultados possíveis (sim ou não), é repetida n vezes, tendo ocorrido x vezes um “sim”. Se a proporção esperada for p , o comando R `binom.test` permite descortinar a plausibilidade de a proporção real, ser de facto, p . Vejamos este teste em ação para o exemplo da moeda:

```
> binom.test(60, 100, p = 0.5)
```

```
Exact binomial test
```

```
data: 60 and 100
number of successes = 60, number of trials = 100, p-value = 0.05689
alternative hypothesis: true probability of success is not equal to 0.5
95 percent confidence interval:
 0.4972092 0.6967052
sample estimates:
probability of success
                0.6
```

No comando, o 60 é o número de sucessos, 100 o número de lançamentos e $p = 0.5$ a proporção que esperaríamos se a moeda não estivesse viciada. O R devolve-nos $p\text{-value} = 0.05743$, que é a plausibilidade do desvio em relação ao esperado (60 sucessos, em vez dos 50 esperados) ser fruto do acaso. Os resultados apontam para uma plausibilidade de cerca de 6%, o que cai na dita “zona cinzenta”. O melhor aqui seria continuar a lançar a moeda, para ver se os 6% ou baixam para os 1%, caso em que a moeda provavelmente estaria viciada, ou se, pelo contrário, sobe para 10%, caso em que não teríamos razão para desconfiar de vício.

Por omissão, o teste à proporção é bilateral. Nos casos em que há uma desconfiança *a priori* que a moeda poderá estar viciada num dos sentidos, pode-se, tal como no caso do teste à média, acrescentar o parâmetro `alternative = "less"` ($p < 0,5$) ou `alternative = "greater"` ($p > 0,5$), que torna o teste unilateral esquerdo ou direito.

O algoritmo de teste à proporção programado no R é exato, portanto a questão da validade do teste não se coloca. Ele é válido para qualquer valor de n , x ou p .

Análise de variância

A análise de variância (*analysis of variance* – ANOVA) é um teste de hipóteses adequado a comparar médias de mais de duas amostras. Esta técnica foi desenvolvida originalmente para fins agrícolas, como uma forma de determinar se diferentes tratamentos aplicados aos terrenos cultivados resultavam em melhoria de produtividade.

Embora a ANOVA seja normalmente usada para comparar dados provenientes de mais de duas amostras, esta técnica pode ainda assim ser usada com apenas duas amostras. No entanto, o leitor deve saber que nesse caso existem testes mais abrangentes, no sentido em que cobrem mais situações do que aquelas em que a ANOVA pode ser aplicada.

Por último, há que referir que existem muitos tipos de ANOVA. Aqui vamos estudar o tipo mais simples, a ANOVA de 1 fator de efeitos fixos (*one-way fixed effects analysis of variance*).

Execução de uma ANOVA com R – caso prático

Vejamos um exemplo de ANOVA, neste contexto original. Suponhamos que três terrenos são submetidos a três tratamentos diferentes, p.ex. adubo A, B ou C. São recolhidas 5 colheitas de cada terreno, tendo-se obtido as seguintes produtividades por hectare:

Tratamento	Produtividade (ton/ha)
A	14, 13, 20, 15, 13
B	13, 14, 13, 18, 15
C	19, 16, 17, 20, 19

A primeira coisa a fazer é entrar estes dados no R:

```
> A = c(14, 13, 20, 15, 13)
> B = c(13, 14, 13, 18, 15)
> C = c(19, 16, 17, 20, 19)
```

A média destes três conjuntos, ou grupos, de dados é

```
> mean(A)
[1] 15
> mean(B)
[1] 14.6
> mean(C)
[1] 18.2
```

e a questão a responder é

Q: “Será que a diferença entre estas médias amostrais é estatisticamente relevante, ou serão apenas flutuações estatísticas?”

Em linguagem matemática poderíamos escrever

Q: “Será que os três grupos têm a mesma média de população, ou haverá pelo menos um que tenha a média de população diferente dos outros?”

Que se traduz, em linguagem formal,

$$H_0: \forall_{ij} \mu_i = \mu_j \text{ vs. } H_1: \exists_{ij} \mu_i \neq \mu_j$$

A rejeição, ou não, de H_0 será baseada nas amostras que tirámos dos três grupos de dados e na técnica ANOVA.

Validação de pressupostos

Antes de prosseguir com a ANOVA há que lembrar que, à semelhança dos testes à média, há uma série de pressupostos a cumprir para que o resultado final seja estatisticamente aceitável. Caso estes pressupostos não sejam todos validados, haverá que procurar na literatura por outros testes de hipóteses alternativos, como p.ex. o teste de Kruskal-Wallis ou de Friedman.

O primeiro pressuposto é a *independência dos grupos*. Ou seja, que não há interferência de um grupo no outro. Tal poderia acontecer p.ex. se os terrenos fossem contíguos e a fertilização se desse por polinização: nesse caso haveria mistura genética entre os grupos, complicando a análise. Outro exemplo é o mesmo conjunto de pessoas ser administrado três medicamentos diferentes. O facto de serem as mesmas pessoas em três situações diferentes coloca em causa a independência e há que recorrer a testes alternativos (neste caso o teste de Friedman). Se fossem três medicamentos administrados a três conjuntos diferentes de pessoas, aí sim, já haveria independência.

O pressuposto de independência não pode ser verificado com nenhum teste estatístico preliminar. Cabe ao investigador zelar para que as condições da experiência aleatória garantem, pelo menos aproximadamente, a independência dos grupos. No caso em estudo vamos assumir que não há problemas de infestação de um terreno pelo outro e que, por conseguinte, há independência.

O segundo pressuposto é que *os grupos seguem distribuições normais*. Para verificar este pressuposto basta correr o nosso já conhecido teste de Shapiro-Wilk. Tal como no caso dos testes à média, dispensa-se esta verificação para $N > 30$.

No caso em mãos temos:

```
> shapiro.test(A)
```

```
Shapiro-Wilk normality test
```

```
data: A  
W = 0.77559, p-value = 0.0505
```

```
> shapiro.test(B)
```

Shapiro-Wilk normality test

```
data: B  
W = 0.84215, p-value = 0.171
```

```
> shapiro.test(C)
```

Shapiro-Wilk normality test

```
data: C  
W = 0.91367, p-value = 0.4899
```

O grupo 1 está no *borderline* da normalidade (p-value pequeno, à roda dos 5%), mas ainda assim pode ser considerado como seguindo uma distribuição normal.

O terceiro e último pressuposto é que *os grupos têm a mesma variância*. Variância é apenas o quadrado do desvio-padrão. O R dispõe de vários testes para verificar esta homogeneidade da variância. Um dos mais usados é o teste de Bartlett, cujas hipóteses são, em linguagem coloquial, H_0 : os grupos têm a mesma variância vs. H_1 : há pelo menos um grupo com variância diferente dos outros.

O comando R para correr o teste de Bartlett é

```
> bartlett.test(list(A,B,C))
```

Bartlett test of homogeneity of variances

```
data: list(A, B, C)  
Bartlett's K-squared = 1.2051, df = 2, p-value = 0.5474
```

O valor de prova de 55% indica clara não-rejeição de H_0 , pelo que se valida o pressuposto da homogeneidade de variância.

Note-se que é necessário escrever explicitamente `list(A,B,C)`. Se escrevermos sem `list`, p.ex. `bartlett.test(A,B,C)`, o comando interpreta mal o significado de `B`:

```
> bartlett.test(A,B,C)  
Error in bartlett.test.default(A, B, C) :  
  there must be at least 2 observations in each group
```

Validados os três pressupostos, podemos finalmente preparar e correr o teste principal, a ANOVA de 1 fator.

Formatação de dados e *data frames*

Para o R correr o teste ANOVA há que agregar os dados da produtividade num vetor apenas. A função `c` (combine), já nossa conhecida, pode ser usada para isso:

```
> prod<-c(A,B,C)  
> prod
```

```
[1] 14 13 20 15 13 13 14 13 18 15 19 16 17 20 19
```

Temos no total 15 valores. Para as associar aos grupos respetivos há agora que criar um novo vetor com indicação de qual é esse grupo. Há duas maneiras de fazer isto: uma “manual”, i.e. escrever explicitamente o vetor:

```
> grupos <-  
c("A", "A", "A", "A", "A", "B", "B", "B", "B", "B", "C", "C", "C", "C", "C")  
> grupos  
[1] "A" "A" "A" "A" "A" "B" "B" "B" "B" "B" "C" "C" "C" "C" "C"
```

e uma outra, mais compacta, que faz uso do comando `rep` (replicar):

```
> grupos<-c(rep("A", 5), rep("B", 5), rep("C", 5))  
> grupos  
[1] "A" "A" "A" "A" "A" "B" "B" "B" "B" "B" "C" "C" "C" "C" "C"
```

O comando `rep("A", 5)` significa: “*replique o carater A cinco vezes*”. O resto é autoexplicativo. Há apenas que ter o cuidado de não nos enganarmos na afetação nem na contagem de valores: se tivermos p.ex. 7 valores no grupo A devemos usar `rep("A", 7)`. Também convém não esquecer da aspas, caso contrário o R pensará que A é uma variável e dará erro mais tarde.

Uma forma de verificar se a afetação ficou bem feita é juntar os dois vetores num único quadro de dados (*data frame*). Para isso usa-se o comando `data.frame` e:

```
> quadro<-data.frame(prod, grupos)  
> quadro  
  prod grupos  
1   14      A  
2   13      A  
3   20      A  
4   15      A  
5   13      A  
6   13      B  
7   14      B  
8   13      B  
9   18      B  
10  15      B  
11  19      C  
12  16      C  
13  17      C  
14  20      C  
15  19      C
```

Esta visualização permite verificar rapidamente se os dados foram bem entrados. O comando `data.frame` tenta juntar os dois vetores `prod` e `grupos` num objeto do tipo *data frame*. O formato *data frame* é semelhante ao já conhecido `table`, mas não exatamente igual. Os *data frames* são importantes em R porque é o formato mais simples para importação de dados de ficheiros externos, como p.ex. folhas de cálculo .XLS ou .XLSX.

Tabela ANOVA e sua interpretação

Estamos finalmente prontos para correr o comando que executa a ANOVA. Este é, juntamente com o resultado,

```
> anova(lm(prod ~ grupos))
Analysis of Variance Table

Response: prod
      Df Sum Sq Mean Sq F value    Pr(>F)    
grupos   2  38.933  19.4667   3.7677 0.05372 .
Residuals 12  62.000   5.1667
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

O comando significa “execute uma ANOVA sobre o modelo linear (*lm*) em que cada valor da produtividade (*prod*) está associada ao grupo respectivo (*grupos*)”. O comando *lm* cria um modelo linear geral, que serve tanto para a ANOVA como para regressões e outras técnicas. Notar que se se trocar *prod* com *grupos*, i.e. tentar *anova(lm(grupos ~ prod))*, o R não vai compreender o que se pretende e vai devolver erro. É possível invocar o comando *anova* a partir do *data frame* *quadro*, mas esse procedimento tem algumas subtilezas que não vamos focar agora.

Vejamos os resultados agora. O p-value do teste ANOVA é o valor na última coluna $Pr(>F) = 0.05372$, cerca de 5%. É um valor no limiar da rejeição de H_0 , ou seja, há uma suspeita considerável de que pelo menos um tratamento tenha gerado diferente produtividade.

Embora a evidência estatística não seja totalmente concludente (sê-lo-ia se o p-value fosse abaixo de 1%), há uma questão que se torna pertinente: qual será, ou quais serão, os tratamentos responsáveis pelas diferenças? Também aqui a estatística nos permite dar uma resposta a esta questão, mediante o que se designa por “teste post-hoc”.

Testes de comparações múltiplas

Os testes de comparações múltiplas, ou testes *post-hoc*, tentam identificar qual/is o(s) grupo(s) responsáveis pela rejeição de H_0 na ANOVA. Há umas boas duas dezenas de testes *post-hoc*, a maioria dos quais o R faz. Vejamos como funcionam na prática. O comando R para o mais usado deles, o “Tukey HSD” (*honest significant difference*) é semelhante ao da ANOVA:

```
> TukeyHSD(aov(lm(prod ~ grupos)))
Tukey multiple comparisons of means
 95% family-wise confidence level

Fit: aov(formula = (lm(prod ~ grupos)))

$grupos
      diff      lwr      upr      p adj
B-A -0.4 -4.2352956 3.435296 0.9583671
C-A  3.2 -0.6352956 7.035296 0.1068512
C-B  3.6 -0.2352956 7.435296 0.0665354
```


A função `aov` é necessária porque o comando `TukeyHSD` atua sobre objetos dessa classe. Trata-se apenas de uma technicalidade que não vale a pena explorar agora. Vejamos, isso sim, os resultados.

O R devolve-nos uma série de comparações (daí o nome “testes de comparações múltiplas”) entre grupos. Na 1ª linha, `B-A`, os grupos A e B são comparados para o teste “*será a média da população de A estatisticamente igual à de B (H_0) ou diferente (H_1)?*” O p-value de 96% aponta claramente para não-rejeição de H_0 . Na 2ª linha, `C-A`, são comparados A e C, para um p-value de 11%. Embora as amostras apontem para diferenças entre as médias de população mais significativas, ainda não suficientemente fortes para indicar uma diferença. Finalmente, na 3ª linha, `C-B`, a comparação entre B e C devolve um p-value de 6,6%.

A conclusão deste estudo não é completamente perentória. No entanto, uma coisa parece clara: se houver um tratamento diferente dos outros, será o tratamento C. Neste caso o que há a fazer é recolher mais dados e repetir a análise.

Exercícios

Para terminar esta breve introdução à visualização de dados e testes de hipóteses com R, deixa-se aqui alguns exercícios para praticar.

1. A sequência abaixo indica o número de telefonemas recebidos por 30 pessoas durante um dia.

2 2 3 4 3 2 1 3 2 4 2 3 3 2 2 4 2 2 1 4 2 0 3 4 5 4 2 0 3 1

Represente estes dados em tabela de frequências, diagrama de barras e circular, e verifique se é plausível que a média diária de telefonemas recebidos seja menor que 3.

2. Os valores abaixo referem-se à altura de 15 adultos do sexo masculino, em cm.

186 184 177 177 179 189 177 185 179 186 179 175 182 187 175

(a) Organize estes dados em 3-4 classes (manualmente ou deixando ao cuidado de `pretty`) e elabore uma tabela de frequências e um histograma dos mesmos.

(b) Atendendo a que a média da altura dos homens portugueses é de 173 cm, qual será a plausibilidade da amostra acima originar na população portuguesa?

3. Um amigo seu assegura que se um dado for lançado de uma certa maneira, duas das suas faces sairão menos vezes que as outras. Ao fim de 50 lançamentos, essas duas faces saem 15 vezes. Terá o seu amigo razão no que diz? Note que neste caso a probabilidade de sair 2 faces em 6 possíveis será, à partida, 2 em 6, ou seja $p = 1/3$.
4. O exemplo de ANOVA que vimos na secção de análise de variância não era totalmente conclusivo, Assim, foram recolhidas mais 5 colheitas de cada um dos terrenos sujeitos aos tratamentos A, B e C. Cada grupo tem agora 10 observações de produtividade, que são

Tratamento	Produtividade (ton/ha)
A	14, 13, 20, 15, 13, 15, 19, 18, 11, 13
B	13, 14, 13, 18, 15, 15, 16, 15, 16, 15
C	19, 16, 17, 20, 19, 16, 18, 21, 19, 16

Terão os dados suplementares ajudado a determinar se há algum tratamento diferente dos outros?

Responda a esta questão repetindo a ANOVA do exemplo: valide pressupostos, corra o teste ANOVA e, caso a ANOVA detete diferenças significativas entre tratamentos, corra o teste Tukey HSD para tentar identificar o(s) tratamento(s) diferentes.