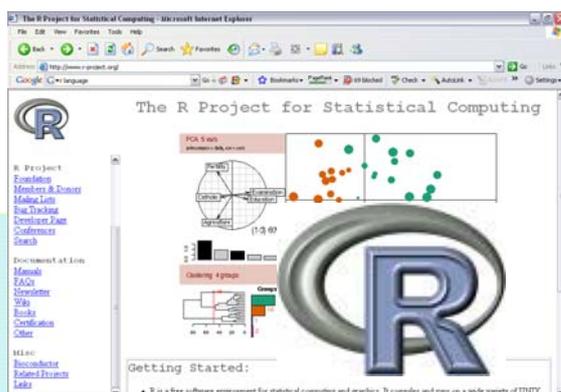




www.alea.pt



Dossiês Didáticos



XIV – Estatística com R

Uma Iniciação para o Ensino Básico e Secundário

Pedro Campos

Rita Sousa

Colaboração de Emília Oliveira

Julho de 2009

Introdução

O projecto ALEA - Acção Local Estatística Aplicada - constitui-se como um contributo para a elaboração de novos suportes de disponibilização de instrumentos de apoio ao ensino da Estatística para os alunos e professores do Ensino Básico e Secundário. Este projecto nasceu de uma ideia conjunta da Escola Secundária de Tomaz Pelayo e do INE, assente nas necessidades e estruturas que os intervenientes possuem. A página Internet do ALEA está no endereço: www.alea.pt.

Índice

1. A utilização de software no Ensino da Estatística
2. O que é o R e para que serve?
3. Primeiros passos
 - 3.1. Instalar o R
 - 3.2. Abrir e Encerrar o R, Ajuda e os Packages
 - 3.3. Menus e comandos principais
 - 3.4. Regras de sintaxe e Objectos
 - 3.5. Introdução de dados com *c()*
 - 3.6. Importação e exportação de dados
 - 3.7. Primeiros passos na Estatística Descritiva
4. O “R Commander”: um ambiente gráfico
5. Análise de Dados
6. Gráficos
7. Exemplos de Aplicação
8. *Para saber mais*: recursos práticos para aprendizagem do R

Números anteriores

- Dossiê I – População e Demografia*
- Dossiê II – Ambiente e Recursos*
- Dossiê III – A Inflação e o Índice de Preços no Consumidor.*
- Dossiê IV – Estatística com Excel.*
- Dossiê V – Censos 2001 - «Tu Também Contas!»*
- Dossiê VI – Notas sobre a História da Estatística*
- Dossiê VII – Probabilidades com Excel.*
- Dossiê VIII – Números do Cinema*
- Dossiê IX – Representações Gráficas*
- Dossiê X – EuropALEA*
- Dossiê XI – O Inquérito Estatístico*
- Dossiê XII – Software Estatístico*
- Dossiê XIII – Estatística Descritiva com Excel – Complementos*

O R é uma linguagem (e ambiente de computação estatística e construção de gráficos) aberta e gratuita cujo número de utilizadores tem vindo a aumentar consideravelmente. O dossiê começa por apresentar o R, referindo os seus aspectos fundamentais e descrevendo, de seguida, os principais comandos. No capítulo 4 apresenta-se o *R-Commander*, uma ferramenta importante que permite tornar a interface gráfica do R mais apelativa. No final há um conjunto de exercícios resolvidos utilizando o código R.

1. A utilização de software no Ensino da Estatística

O software estatístico que foi sendo introduzido nas últimas décadas trouxe novas formas de explorar a Estatística, proporcionando maior rapidez na resolução de problemas e permitindo a comparação expedita de soluções. Além disso, abriu caminho a um conjunto de utilizadores nos meios académico, empresarial e administrativo que desta forma puderam passar a utilizar a Estatística como uma ferramenta eficaz na resposta aos seus problemas.

No ensino em geral a utilização do computador permitiu introduzir diversas melhorias, pois no contexto escolar usual, “os alunos têm grande dificuldade em aprender novos assuntos cujo significado não vislumbram e que não lhes despertam qualquer interesse” (ver João Pedro da Ponte na Introdução de “A Família em Rede”, de Seymour Papert, 1997). O computador e, em particular, o software estatístico permitiram incentivar a participação voluntária do aprendiz no processo educativo, fazendo com que o aluno passe a explorar os dados e a ser cada vez mais o centro desse desafio do ensino/aprendizagem da estatística.

No entanto, apesar de serem reconhecidas as vantagens da utilização do software estatístico, nomeadamente no que respeita ao ensino da estatística, a sua utilização deve ser sempre suportada por um adequado conhecimento das técnicas estatísticas envolvidas ou orientada por quem detenha esses conhecimentos (ALEA, Dossiê Didático X – Software Estatístico, Luís Cunha e Helder Alves).

No Dossier Didático X (Software Estatístico - Uma introdução a alguns aplicativos, numa abordagem inicial dos dados, Helder Alves, Luís Cunha) foram apresentadas algumas aplicações informáticas (Minitab, SAS, SPSS, Statistica) para a análise estatística de dados, numa abordagem preliminar dos dados, ao nível da estatística descritiva. Neste dossiê, concentramos as atenções no R, um importante e poderoso veículo de análise interactiva de dados que, devido à sua crescente utilização nos meios académico e empresarial, não poderia passar despercebido no contexto do ALEA.

2. O que é o R e para que serve?

O R é uma linguagem e ambiente de computação estatística e construção de gráficos; é considerada uma variante da linguagem S (laboratórios Bell, desenvolvida por John Chambers e seus colegas). Surge pela criação da *R Foundation for Statistical Computing*, com o objectivo de criar uma ferramenta gratuita e de utilização livre, para análise de dados e construção de gráficos.



O R é compatível com diversas plataformas: UNIX, Windows e MacOS e permite a ligação a interfaces de diferentes formatos: Excel, Access, SPSS, SAS, SQL Server. Sendo *Open Source*, permite ao utilizador aceder ou alterar funcionalidades existentes, bem como criar novas funcionalidades para responder aos seus problemas específicos de forma mais eficaz. Tal é possível graças à possibilidade de o R se estender a partir de um crescente conjunto de livrarias (packages) que podem ser acedidas pelo utilizador.

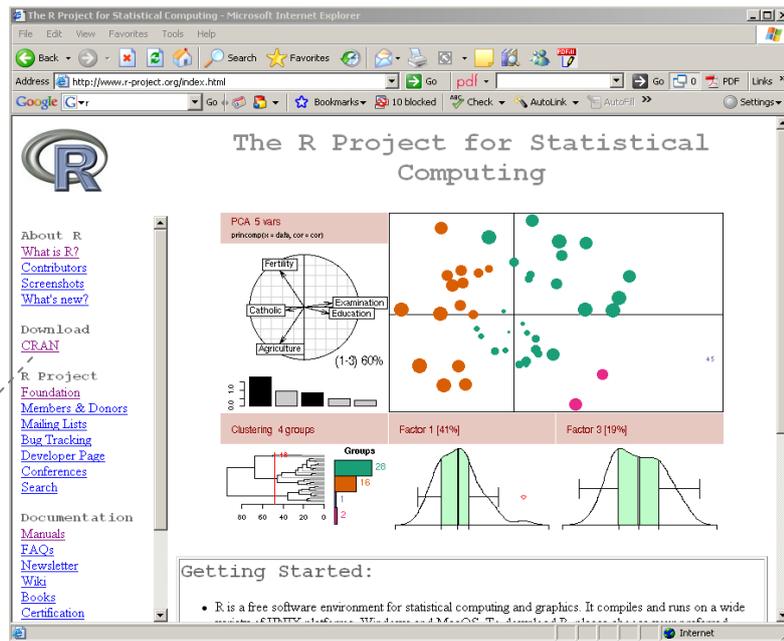
A interacção com o utilizador é baseada numa janela de comandos e exige o recurso a programação, embora existam packages gráficos que permitem a interacção através de menus. Um desses packages é o *R Commander* que será abordado no contexto deste dossiê.

Apesar de existirem muitas facilidades de entreaajuda na comunidade de utilizadores do R, esta linguagem não tem suporte técnico assegurado.

3. Primeiros passos

3.1. Instalar o R

A instalação do R é gratuita e pode ser feita directamente a partir da página principal do *R Project for Statistical Computing* em <http://www.r-project.org/>. A figura seguinte indica o local onde se pode efectuar a importação do R.



Permite fazer o download da aplicação (com as funcionalidades base) e dos vários *packages* adicionais.

Fig. 1 - O download do R é feito a partir da página principal do Projecto R na área CRAN (*Comprehensive R Archive Network*)

Para a importação do R é necessário escolher: um país a partir do qual o ficheiro será transferido, o sistema operativo (MacOS X, Linux, ou Windows), o link *base* e, finalmente, o programa executável. A última versão à data deste dossiê é: *R-2.9.0-win32.exe*.

Após importação deste ficheiro, a instalação é rápida e intuitiva.

3.2. Abrir e Encerrar o R, Ajuda e os Packages

O “prompt”

Ao iniciar o R mostra-se imediatamente a janela de comandos (V. Fig. 2). Esta janela exhibe um cursor vermelho em forma de sinal “maior” (>) designado por *prompt* onde são escritos os comandos. Por exemplo, para se obter o número da versão do R em causa deve-se escrever:

> R.version

>
Prompt ou linha de comando

>R.version
Permite obter o número da versão de R

Para sair do R, pode-se utilizar o menu (File/Exit) ou então escrever:

> q()

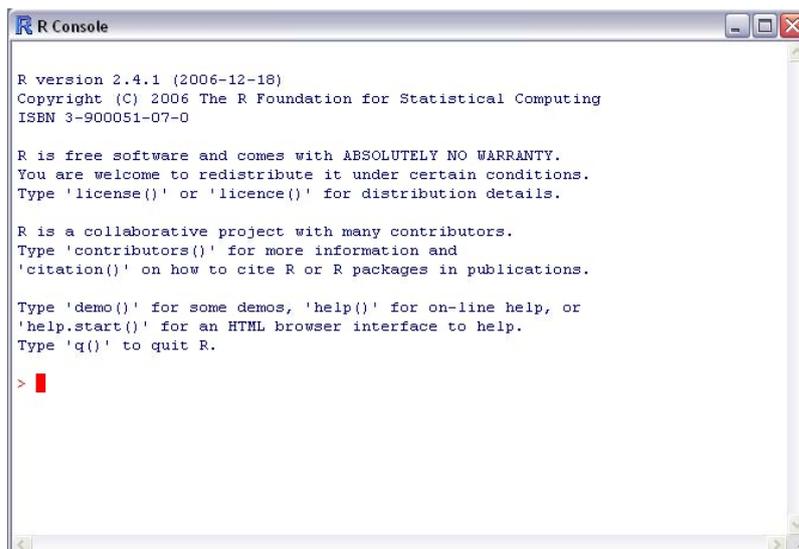
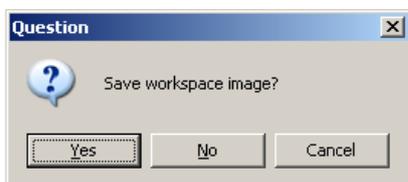


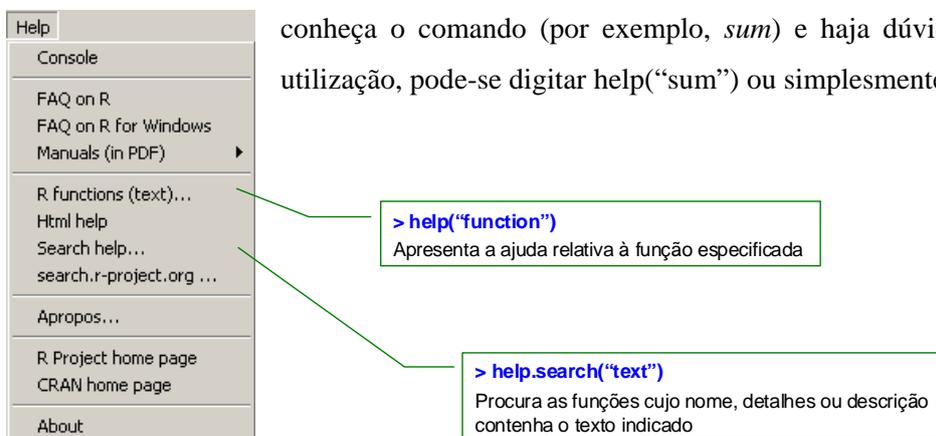
Fig. 2 - Janela de comandos do R de uma das versões anteriores

Entrar e Sair

Uma das perguntas que surge habitualmente ao abandonar o R é se pretende guardar o espaço de trabalho (workspace). De facto, o R pode guardar no seu workspace o nome e o valor dos objectos criados. Veremos nas secções seguintes como criar esses objectos.



Para qualquer tipo de ajuda (que é muito útil quando se tem uma linguagem como o R) existem muitas opções, sendo a mais intuitiva a que está acessível pelo menu Help da barra de menus. Outra forma muito prática para obter ajuda para qualquer função consiste em digitar `help.search("text")` em que *text* representa o que pretendemos pesquisar. Em alternativa, caso se conheça o comando (por exemplo, *sum*) e haja dúvidas quanto a sua utilização, pode-se digitar `help("sum")` ou simplesmente `?sum`.

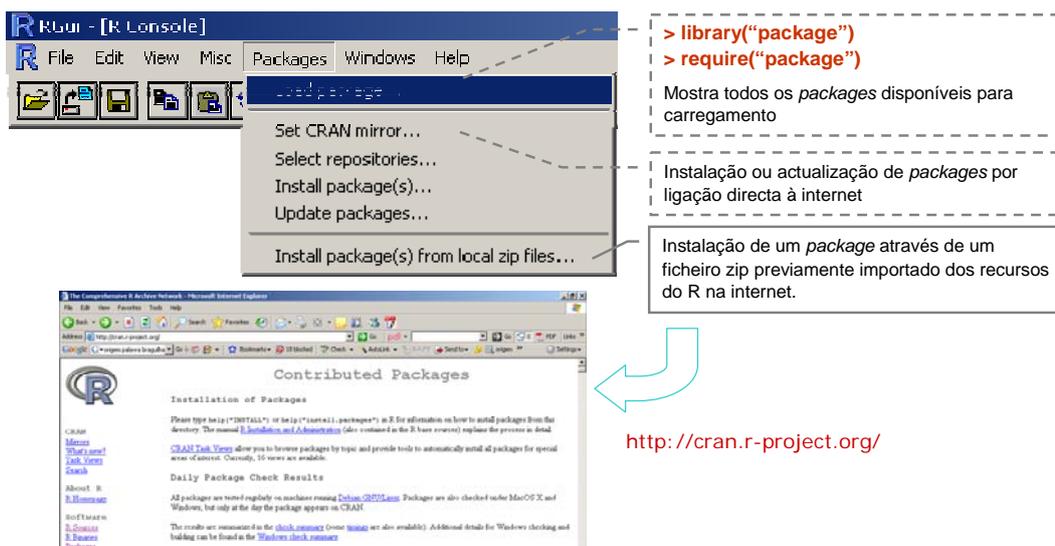


Os Packages

Todos os recursos do R (dados ou funções) estão armazenados em packages. O conteúdo de um determinado package só fica disponível quando este é carregado. O package base (standard) é considerado parte integrante dos recursos do R, sendo carregado automaticamente aquando da instalação do programa. As funções básicas que permitem ao R trabalhar os principais objectos de dados, funções estatísticas e gráficas, já estão disponíveis no package *base*.

Existem funções específicas para extrair informação sobre os packages: por exemplo, para ver os packages que estão instalados no PC deverá escrever o comando `library()`. Para carregar um determinado package deve usar `library("package")`.

A instalação dos packages e o seu carregamento (Install package(s) from zip files...) e (*load package*) devem ser feitos por esta ordem e podem ser executados directamente a partir dos menus do R. Os packages pretendidos podem ser previamente importados em formato *zip* através do site do R (<http://www.cran.r-project.org/>) e carregados posteriormente.



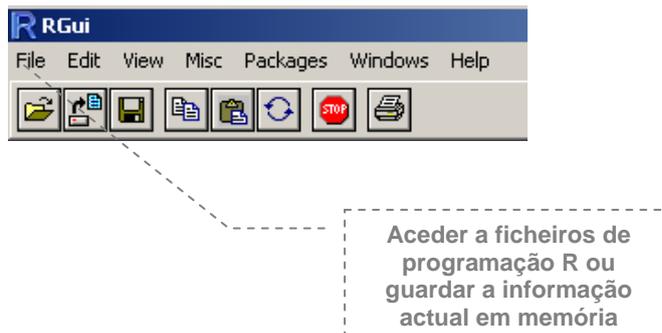
The image shows the RGui interface with the 'Packages' menu open. The menu options are: 'Load package...', 'Set CRAN mirror...', 'Select repositories...', 'Install package(s)...', 'Update packages...', and 'Install package(s) from local zip files...'. Annotations with dashed boxes explain these options:

- 'Load package...' is linked to the R commands `> library("package")` and `> require("package")`, with the note 'Mostra todos os packages disponíveis para carregamento'.
- 'Install package(s)...' is linked to the note 'Instalação ou actualização de packages por ligação directa à internet'.
- 'Install package(s) from local zip files...' is linked to the note 'Instalação de um package através de um ficheiro zip previamente importado dos recursos do R na internet.'

 Below the menu, a browser window shows the 'Contributed Packages' page on the CRAN website. A red arrow points from the URL <http://cran.r-project.org/> to the browser window.

3.3. Menus e comandos principais

O R exibe uma barra de ferramentas e um sistema de menus que permite executar algumas operações. Basicamente o menu File permite Gravar e abrir sequências de comandos (*scripts*), abrir ou gravar espaço de trabalho (*workspace*), sair do R, etc. Permite ainda, carregar livrarias (*packages*), que serão descritas mais adiante neste dossiê.



The image shows the top menu bar of RGui: File, Edit, View, Misc, Packages, Windows, Help. Below the menu bar is a toolbar with icons for file operations (open, save, print), editing (undo, redo), and execution (stop, refresh). A dashed box points to the 'File' menu and the toolbar with the text: 'Aceder a ficheiros de programação R ou guardar a informação actual em memória'.

Uma das opções disponíveis neste menu principal é a ajuda (*help*). O R dispõe de um completo sistema de ajuda, composto pelas seguintes opções:

- Opções de ajuda sobre a aplicação;
- Opções de ajuda com acessos a partir do browser;
- Opções de ajuda na janela de comandos do R.

Algumas dessas opções podem ser efectuadas directamente no *prompt* da seguinte forma:

> **help("function")** ou

>?**function**

Apresenta a ajuda relativa à função especificada;

> **help.start()**

Dá acesso a informação auxiliar a partir do browser;

> **help.search("text")**

Procura as funções cujo nome, detalhes ou descrição contenha o texto indicado;

> **apropos("text")**

Procura as funções cujo nome contenha o texto indicado.

> **help("function")**
 > **help.start()**
 > **help.search("text")**
 > **apropos("text")**

Funções que permitem obter ajuda no R

3.4. Regras de sintaxe e Objectos

Primeiras Regras

Uma das regras importantes do R é o facto de ser *case sensitive*. Por esta razão as letras ‘a’ e ‘A’ podem corresponder a diferentes variáveis. Além disso, o R ignora espaços, ou seja, os resultados ‘8+3’ e ‘8+ 3’ dão origem exactamente ao mesmo resultado. Outras regras importantes:

- Podemos agrupar comandos, para serem executados em simultâneo, se estiverem entre chavetas ‘{ }’ e separados por ‘;’;
- O ‘#’ é utilizado para comentários;
- Quando um comando não está completo, o R coloca o sinal de ‘+’ na linha seguinte, permitindo que este seja terminado.

Objectos

No R todos os diferentes conteúdos tais como números, textos, vectores, matrizes, expressões, chamadas funções, etc. são guardados na memória do computador sob a forma de **objectos**. Todos os objectos têm um nome associado e para armazenamento num objecto usa-se o

operador de atribuição, '<-' ou '='. Para visualizar o conteúdo do objecto basta digitar o nome do mesmo.

```
> texto <- "teste"
> texto
[1] "teste"
Forma possível de criação
de um objecto designado
por texto, contendo "teste"
```

3.5 Introdução de dados com c()

O vector coluna c()

Uma das formas práticas de armazenar valores em R é feita através de objectos denominados *vectores*. O vector é considerado a estrutura de dados mais simples e consiste numa colecção organizada de elementos. A atribuição é feita a partir da função `c()`, cujos argumentos correspondem aos próprios elementos do vector.

A atribuição pode ser feita também por intermédio da função `assign()` que é particularmente útil nas atribuições automáticas, em que desconhecemos os nomes dos objectos.

```
> x <- c(3.5,1.4,5,2.6,7,4.8)
> x
[1] 3.5 1.4 5.0 2.6 7.0 4.8
Atribuição de valores ao vector x
```

Operações com vectores

Uma das vantagens do R é a facilidade na operação com vectores. O vector exemplo, `x` (composto pelos números 1, 2, 3, 4, 5, pode ser transformado num vector `y` (que seja igual a $2x+1$) desta forma simplificada:

```
> x <- c(1,2,3,4,5)
> y <- 2*x + 1
> y
[1] 3 5 7 9 11
```

```
> assign("x",c(3.5,1.4,5,2.6,7,4.8))
> x
[1] 3.5 1.4 5.0 2.6 7.0 4.8
Atribuição de valores ao vector x
(alternativa)
```

De uma forma simples podemos também listar todos os números que sejam superiores a um certo limite, utilizando operadores lógicos. Assim sendo, se pretendermos guardar num outro vector `z` apenas os valores de `y` superiores a 3, devemos escrever:

```
> z <- y[y>3]
> z
[1] 5 7 9 11
```

3.6. Importação e exportação de dados

O R dispõe de um conjunto de funções que permitem a importação ou exportação de dados. Para importar ou exportar ficheiros externos, o R dispõe de conjunto de funções que variam de acordo com o formato do ficheiro.

Para ler ficheiros de dados em formato de tabela existem funções mais específicas (dependendo do tipo de ficheiro) e a função *read.table* que é mais abrangente:

```
> read.table(file,...)
```

```
> read.csv(file,...)
```

```
> read.csv2(file,...)
```

```
> read.delim(file,...)
```

```
> read.delim2(file,...)
```

```
> read.table("C:/.../info.txt",sep="\t",dec=";",header = TRUE)
```

```
Ano Pop Fam  
1 2007 25709 260  
2 2007 28329 286  
3 2007 1327 37  
4 2007 1390 26  
5 2007 34224 205
```

Ler uma tabela a partir de um ficheiro externo em formato txt

Para saber como se deve usar cada um destes comandos, basta escrever, no R, o nome do comando antecedido de ?, por exemplo:
> ?read.csv

Na importação de ficheiros há alguns parâmetros que é importante definir para garantir a correcta leitura dos dados, tais como:

- **sep="\t"**, para indicação do caracter tabulação como separador entre variáveis;
- **dec=","**, para indicação do separador decimal;
- **header = TRUE**, para indicação da existência dos nomes das variáveis na primeira linha.

Ao importar um ficheiro para o R, este deve ficar associado a um objecto. Para tal, o resultado do comando de importação deve ser atribuído ao nome do objecto a que se quer associar. Para importar, através da função *read.csv*, um ficheiro de texto designado por "ex.csv" e o associar a um objecto *Dataset*, dever-se-á fazer:

```
> Dataset<-read.csv("C:/.../ex.csv",sep="\t",dec=";",header = TRUE)
```

3.7. Primeiros passos na Estatística Descritiva

Análise descritiva

O R dispõe de um conjunto de funcionalidades que permitem fazer uma análise descritiva de dados bastante completa. As medidas descritivas utilizadas e a forma de sumarização da informação deve sempre atender ao tipo de dados de que dispomos, ou seja, às características das variáveis que estamos a analisar. É sabido que para as variáveis quantitativas se podem aplicar, entre outras, medidas de localização e de dispersão¹.

Em resumo, podemos recordar que as **Medidas de Localização** são medidas que localizam um determinado ponto da distribuição tais como os quartis, o mínimo e o máximo. Quando o ponto em questão corresponde ao centro da distribuição, estas denominam-se por medidas de tendência central (exemplos: média, moda, mediana). As **Medidas de Dispersão** são as medidas que aferem a variação dos dados em relação ao centro da distribuição (exemplos: variância, desvio padrão, coeficientes de variação e de dispersão). De seguida apresentam-se alguns exemplos simples de utilização das medidas de localização e de dispersão com R.

Medidas de Localização

- **Média aritmética:** `mean()` calcula a média aritmética simples, para variáveis quantitativas (discretas e contínuas).
- **Mediana:** `median()` calcula a mediana ou valor central de uma distribuição após ordenação da amostra (é definida pela sua posição na sucessão das observações ou na distribuição de frequências); é também conhecida por percentil 50 ou segundo quartil.
- **Quantis:** `quantile()` a função calcula os quantis que são estatísticas de ordem que separam a distribuição de acordo com um limite percentual de observações. No caso dos quartis, a distribuição é dividida em quatro partes iguais; estando ordenadas as observações, por ordem crescente, o 1º e o 3º quartis acumulam (até si) 25% e 75% das observações, respectivamente.

```
> a<-c(1,2,3,4,5)
> mean(a)
[1] 3
```

A função `mean()` calcula a média de uma lista de valores

```
> a<-c(1,2,3,4,5)
> median(a)
[1] 3
```

A função `median()` calcula a mediana de uma lista de valores

```
> a<-c(1,2,3,4,5)
> quantile(a)
 0% 25% 50% 75% 100%
 1  2  3  4  5
```

A função `quantile()` calcula os quartis de uma lista de valores

¹ Geralmente definem-se dois tipos de variáveis: qualitativas e quantitativas. Para saber mais sobre os tipos de dados e sobre as medidas a aplicar em cada caso, consultar o ALEA em 'Noções de Estatística: III – Dados, tabelas e gráficos, disponível em: http://www.alea.pt/html/nocoes/html/cap3_1_i.html.

Medidas de Dispersão

- **Variância: `var()`** - calcula a variância para uma variável quantitativa.
- **Desvio padrão: `sd()`** - calcula o desvio padrão de uma variável quantitativa.

```
> a<-c(1,2,3,4,5)
> var(a)
[1] 2.5
A função var() calcula a variância de
uma lista de valores
```

```
> a<-c(1,2,3,4,5)
> sd(a)
[1] 1.581139
A função sd() calcula o desvio padrão
de uma lista de valores de uma
variável quantitativa
```

O R dispõe de algumas funções que permitem fazer uma sumarização de dados, essencialmente para variáveis quantitativas (discretas e contínuas). Uma dessas funções é o `summary()`, que calcula para as variáveis quantitativas as seguintes medidas: Mínimo (Min), 1º quartil (1st Qu), Mediana (Median), Média (Mean), 3º quartil (3rd Qu) e Máximo (Max).

```
> a<-c(1,2,3,4,5)
> summary(a)
  Min. 1st Qu. Median  Mean 3rd
Qu.   Max.
  1     2     3     3     4     5
> A função summary calcula
algumas estatísticas básicas de uma
lista de valores
```

Em resumo, sintetizamos no quadro seguinte os nomes das funções apresentadas, bem como de outras mais específicas, que permitem calcular as respectivas medidas estatísticas no R:

Função	Descrição
<code>table()</code>	Cruzamento de variáveis
<code>mean()</code>	Média aritmética
<code>median()</code>	Mediana
<code>sum()</code>	Soma
<code>summary()</code>	Sumarização de dados
<code>var()</code>	Variância
<code>sd()</code>	Desvio padrão
<code>quantile()</code>	Quartis com descrição
<code>fivenum()</code>	Quartis sem descrição
<code>IQR()</code>	Amplitude inter-quartil
<code>cor()</code>	Coefficiente de correlação

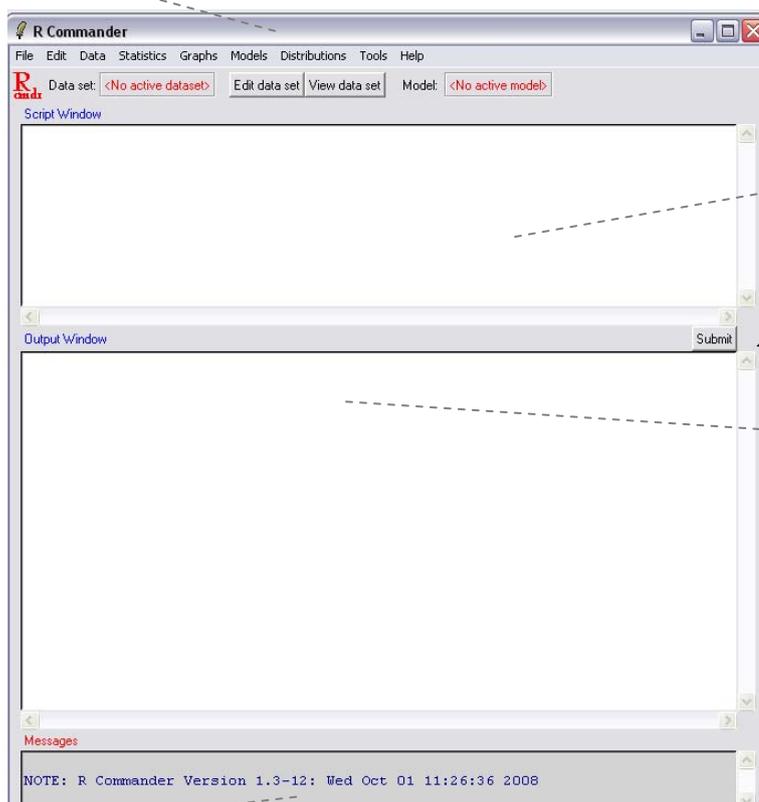
4. “R Commander”: um ambiente gráfico

O que é?

Devido ao seu tipo de interface o R torna-se muitas vezes uma ferramenta de utilização pouco amigável. Por essa razão, têm surgido alguns ambientes gráficos que permitem uma utilização do R de uma forma mais intuitiva. O *R-Commander* é uma dessas interfaces gráficas que abre uma janela inicial contendo vários menus e botões de acesso a diferentes procedimentos. Além disso, este ambiente contém uma janela que gera os comandos R que são utilizados em cada procedimento, permitindo assim repetir ou alterar esses comandos. O aspecto geral da janela do *R-Commander* é apresentado de seguida.

Os menus do *R-Commander* são facilmente configuráveis através de um ficheiro texto ou através dos packages.

Apenas as linhas da janela *script window* (que contém os comandos gerados pelo R) podem ser editadas e submetidas novamente para execução. Para serem submetidas basta carregar em *submit*.



As acções executadas via menus dão origem a comandos do R que são mostrados na janela de output (*output window*), juntamente com a informação de output, como consequência do comando executado.

As mensagens de erro e os avisos são mostrados na *messages window*.

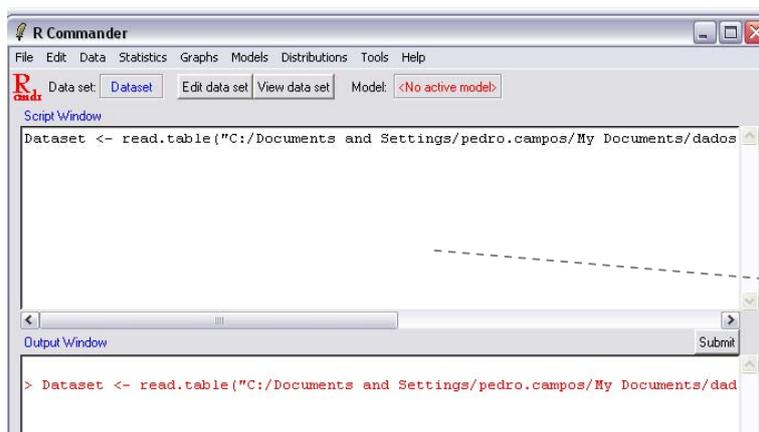
Como se instala?

O *R-Commander* é um package standard (designado por *Rcmdr*) e os processos de instalação e carregamento fazem-se da mesma forma do que nos outros *packages* (seguir o procedimento *install packages* – escolhendo o *package Rcmdr* e, depois, *load package*). Existem, por vezes, alguns aspectos a ter em conta durante a instalação: um dos pontos a ter em conta é que o *R-Commander* utiliza alguns “contributed” packages que devem estar instalados para que o *R-Commander* funcione adequadamente².



Como funciona?

Um dos primeiros passos a dar depois de entrar no *R-Commander* consiste em activar um conjunto de dados. A partir desse momento, todas as acções serão executadas nesse conjunto de dados. Ao abrir-se um novo conjunto de dados, este passará a ser o conjunto de dados activo. O utilizador pode, em qualquer momento, seleccionar o conjunto que pretende, entre todos os conjuntos de dados que já estiveram activos anteriormente.



Para activar um conjunto de dados pode-se importar um ficheiro de texto através do menu: (Data/Import Data/from text file or clipboard

O ficheiro em causa contém dados sobre as peças produzidas numa determinada fábrica de peças para automóveis. Para cada peça produzida dispõe-se de informação sobre:

- **secao**: Secção onde a peça foi produzida (*var. qualitativa: valores de 1 a 6*);
- **cod**: código da peça (*var. qualitativa: valores possíveis: 12, 45, 78, 96*);
- **peso**: peso da peça (*var. quantitativa*);

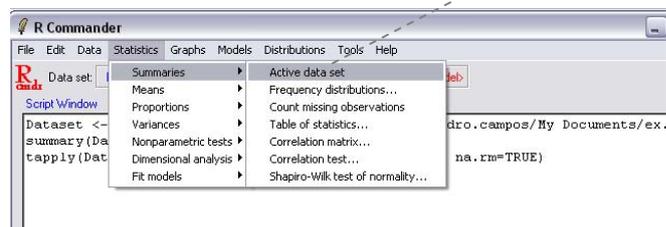
² No caso da versão 1.4-2 do *R-Commander* esses packages são: *abind*, *car*, *effects*, *lmtest*, *multcomp*, *mvtnorm*, *relimp*, *sandwich*, *strucchange*, e *zoo*. Além destes packages, deve-se instalar também o package *rgl* no caso de se pretender construir gráficos 3D.



- **diametro:** diâmetro da peça (*var. quantitativa*);
- **empregado:** empregado que executou/verificou a peça (*var. qualitativa: valores de 1 a 3*);
- **tipo:** Tipo de aplicação da peça: (*var. qualitativa: (c) coluna ou (d) dentro*);
- **qualidade:** resultado da verificação: (*var. qualitativa: (0) rejeitada ou (1) aprovada*).

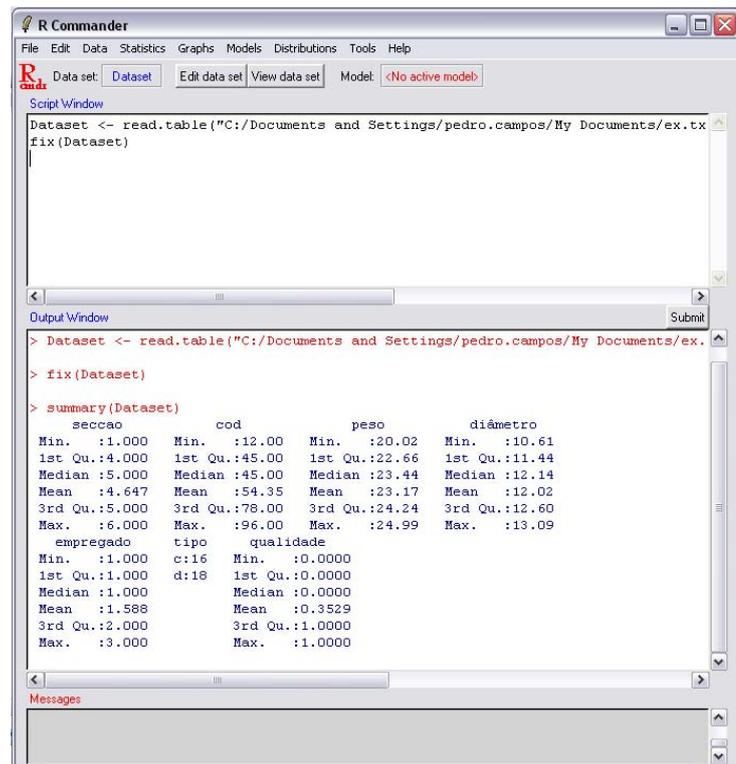
De seguida, no menu ‘Statistics/Summary/Active Data Set’ pode solicitar as estatísticas básicas (mínimo, máximo, mediana, quartis) que correspondem à execução do comando `summary`.

No menu Statistics seleccione a opção Summary/Active Data Set que permite calcular as estatísticas básicas (mínimo, máximo, mediana, quartis), que correspondem à execução do comando `summary()`.



Os resultados encontram-se na figura ao lado (*output window*). Para cada variável foram calculadas as estatísticas: mínimo, máximo, 1º, 2º e 3º quartis, a média e a mediana. Estes resultados poderiam ter sido obtidos directamente através do comando:

`>summary(dataset)`

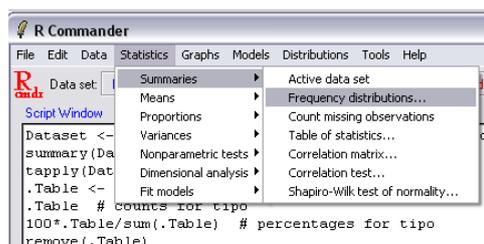


Como neste conjunto de dados existem variáveis de vários tipos, podemos utilizar algumas funcionalidades disponíveis do *R-Commander*, tais como distribuições de frequências, cálculos de estatísticas variadas, representação gráfica, etc. Desenvolveremos esta análise nos próximos capítulos do dossiê.

5. Análise de Dados

Frequências absolutas e relativas

Prosseguindo com o exemplo anterior, em que dispomos de variáveis de vários tipos (qualitativas e quantitativas), interessa analisar agora as potencialidades do *R-Commander*. Após a primeira sumarização, onde se calcularam as medidas de localização, podemos agora, por exemplo, calcular as frequências absolutas das variáveis qualitativas. Para tal, deve-se escolher no menu *Statistics* a opção ‘*Summatize/Frequency Distributions*’.



O resultado é mostrado na janela *output window* como sendo a aplicação da função *table()* da seguinte forma:

```
> .Table <- table(Dataset$tipo)
```

```
> .Table # counts for tipo
```

```
c d
```

```
16 18
```

É de notar que a expressão *Dataset\$tipo* é a forma como correctamente nos referimos à variável tipo do conjunto de dados denominado *Dataset* e que é equivalente a utilizar a expressão *Dataset[, “tipo”]*.

No *R-Commander* mostram-se ainda as frequências relativas associadas a estas frequências absolutas.

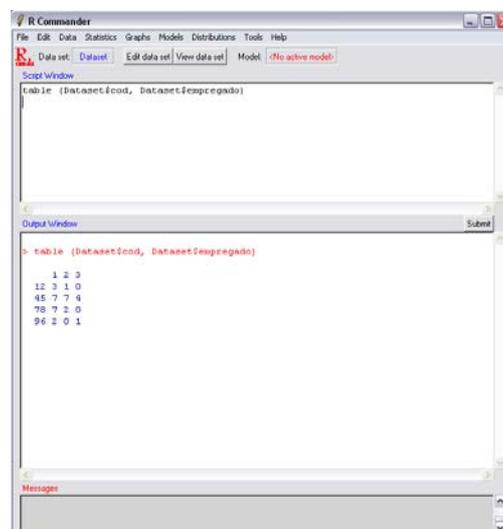
```
> 100*.Table/sum(.Table) # percentages for tipo
```

```
c d
```

```
47.05882 52.94118
```

Tabelas de contingência

Podemos também combinar variáveis e calcular tabelas de contingência que resultam das frequências cruzadas entre variáveis qualitativas. Embora não exista um comando directamente acessível, através dos menus do *R-Commander*, pode-se escrever o comando na janela *Script Window* e carregar no botão *Submit* para executar o comando. Assim sendo, para podermos, por exemplo,



identificar quantas (e quais) as peças que foram feitas por cada empregado, devemos escrever:

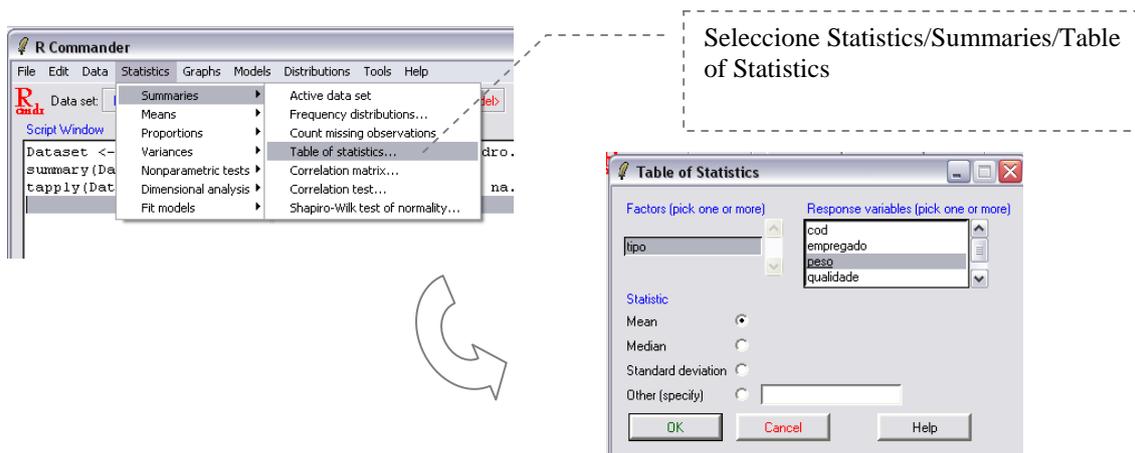
```
>table (Dataset$cod, Dataset$empregado)
```

O resultado é o seguinte:

```
  1 2 3
12 3 1 0
45 7 7 4
78 7 2 0
96 2 0 1
```

Medidas de localização e de dispersão:

De seguida podemos também calcular as medidas de localização e de dispersão para uma variável quantitativa, por grupos definidos segundo as modalidades de uma variável qualitativa. Por exemplo, podemos calcular estatísticas sobre o peso das peças produzidas, tendo em conta o tipo de peça. Para tal devemos escolher a opção ‘*Statistics/Summaries/Table of Statistics*’ e, de seguida, escolher como Factor a variável *tipo*. Neste caso, o *tipo* é aqui considerada uma variável independente.



O resultado é a execução do comando *tapply* que aplica um procedimento à variável quantitativa para grupos distintos (identificados pela variável qualitativa).

```
> tapply(Dataset$peso, list(tipo=Dataset$tipo), mean, na.rm=TRUE)
tipo
  c    d
26.02323 29.12170
```

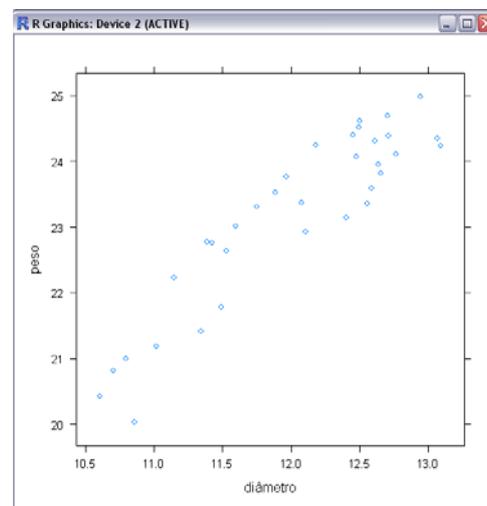
Correlação

Quando numa base de dados se dispõe de mais do que uma variável, pode fazer sentido calcular o nível ou grau de associação existente entre essas variáveis. Em geral, estes coeficientes medem a força e a direcção (no mesmo sentido ou em sentidos opostos) da relação entre as variáveis. Existem vários tipos de coeficientes de correlação conforme o tipo de variáveis em estudo: qualitativas nominais, qualitativas ordinais, quantitativas, etc. O coeficiente de correlação linear de *Pearson* é um dos mais conhecidos e aplica-se quando as variáveis são quantitativas³.

Para se perceber que tipo de relação existe entre um par de variáveis, é habitual começar-se por desenhar um diagrama de pontos. Este tipo de representação é muito útil, pois permite realçar algumas propriedades entre os dados, nomeadamente no que diz respeito ao tipo de associação entre as variáveis.

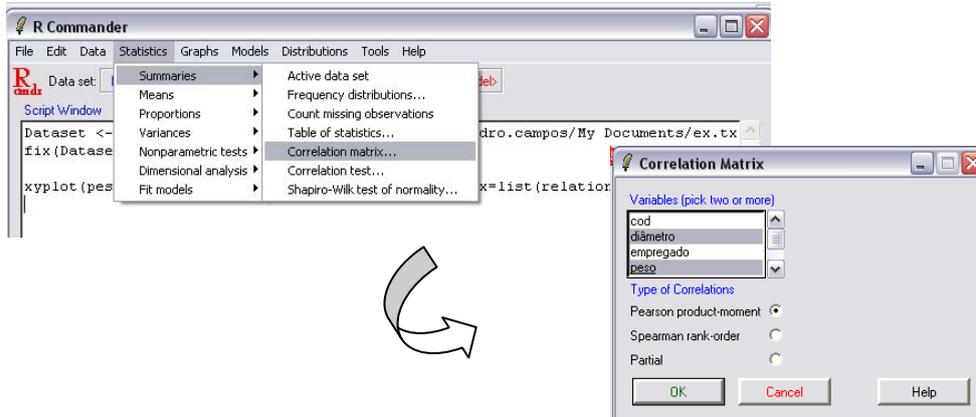
No caso do conjunto de dados em estudo, vamos verificar a relação existente entre as variáveis *peso* e *diâmetro* das peças. Para tal escolhemos no *R-Commander* a opção ‘*Graphs/XY Conditioning plot*’⁴.

Este gráfico sugere a existência de uma relação directa entre as variáveis *diâmetro* e *peso*, ou seja, a valores grandes de diâmetro correspondem, de um modo geral, valores grandes de peso e vice-versa. Esta informação pode ser confirmada pelo cálculo do coeficiente de correlação linear de *Pearson* (ou *r* de *Pearson*). Este procedimento pode ser desencadeado através do menu (ver figura seguinte) e corresponde à execução do comando `cor(x,y)`, em que *x* e *y* representam as variáveis em estudo para as quais se pretende calcular o coeficiente de correlação.



³ Embora este coeficiente se aplique especialmente no caso em que as variáveis seguem distribuição Normal, esta restrição é muitas vezes ignorada. Para saber mais sobre o coeficiente de correlação, consulte o curso de Noções de Estatística no ALEA, Capítulo VI – Distribuições Bidimensionais, em http://www.alea.pt/html/nocoos/html/cap6_3_1.html e/ou *ActivALEA* n.º 4 “Associação entre variáveis quantitativas: O coeficiente de Correlação.”

⁴ No capítulo 6 deste dossiê pretende-se aprofundar um pouco mais a questão da representação gráfica em R.



Na janela *Output Window* podemos observar o resultado:

```
> cor(Dataset[,c("diâmetro", "peso")], use="complete.obs")
           diâmetro      peso
diâmetro 1.0000000 0.9166048
peso      0.9166048 1.0000000
```

De facto, podemos notar que a correlação existente entre o diâmetro das peças (x) e o peso das peças (y) é de, aproximadamente, 0.92.

O *R-Commander* dispõe também de outras opções de análise de dados: análise factorial, testes paramétricos e não paramétricos, etc. Estas técnicas não são abordadas no contexto deste dossiê.

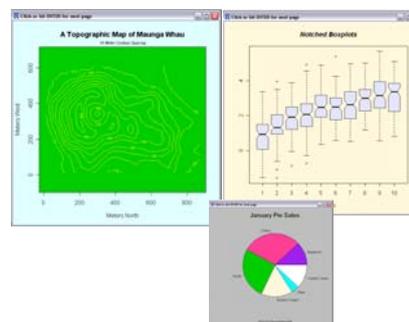
Gestão das variáveis

No *R-Commander* existe a possibilidade de se fazer a gestão do conjunto de dados: acrescentar novas variáveis, novas observações, agregar valores em classes, etc. Esta opção encontra-se disponível através de ‘*Data/Manage variables in active data set*’.



6. Gráficos

As facilidades gráficas são uma componente importante e muito versátil no ambiente R, sendo possível utilizar essas facilidades numa larga variedade de gráficos estatísticos predefinidos, bem como construir gráficos novos que podem ser formatados e apresentados com grande qualidade.



Os gráficos constituem uma forma de sumariar a informação, sendo que a sua representação gráfica deve ser feita de forma a dar relevo às propriedades importantes dos dados. A construção dos gráficos deve ter em conta o tipo de variáveis que se pretende representar. Na tabela seguinte apresenta-se um resumo do tipo de gráficos, mais comuns, que deve ser feito para cada tipo de variável:

Tipo de variável	Representação gráfica
Qualitativa (ordinal, nominal)	Gráficos de barras, diagramas circulares.
Quantitativa discreta	Gráficos de barras, diagramas circulares, diagramas de dispersão, diagramas de caixas e bigodes, etc.
Quantitativa contínua	Histogramas, diagramas de dispersão, diagramas de caixa e bigodes, etc.

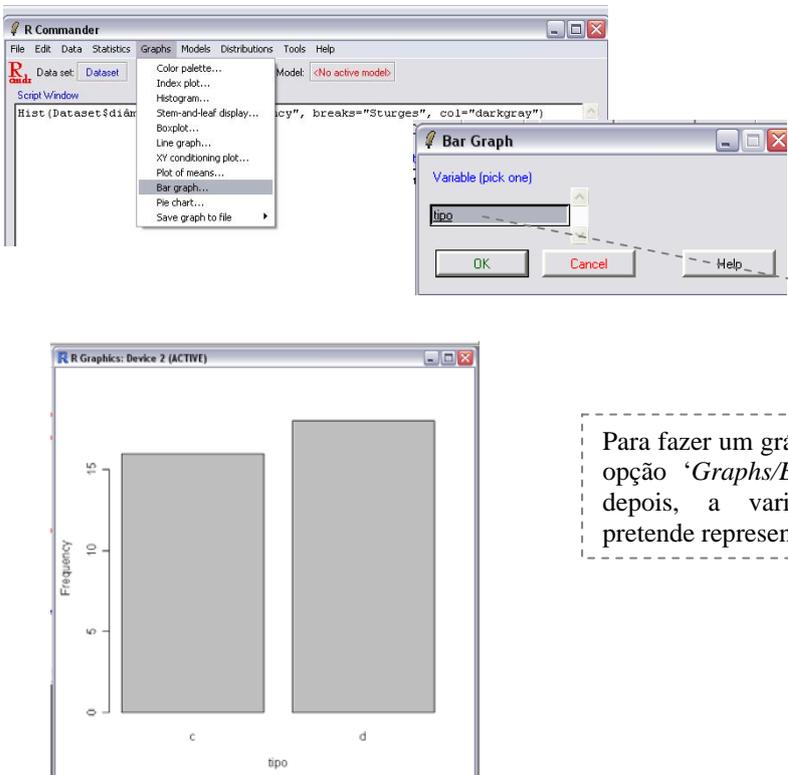
Neste capítulo pretende-se fazer uma visita geral a alguns tipos de gráficos mais conhecidos (gráficos de barras, diagramas circulares, histogramas e gráficos de pontos) e à forma com se podem construir através do *R-Commander*. A apresentação específica de cada gráfico e a sua formatação não são objectivo principal desta abordagem, pelo que deverá consultar as ajudas do R para comandos adicionais.

Apresenta-se, de seguida, a forma como pode fazer alguns destes gráficos tomando por base o mesmo conjunto de dados dos exemplos anteriores.

Gráfico de barras e diagramas circulares

O gráfico de barras é uma forma de representação adequada a variáveis qualitativas e quantitativas discretas. No gráfico de barras cada valor associado a uma modalidade da variável é representado através de uma barra cuja altura é proporcional à sua frequência.

De seguida apresentam-se os passos necessários para fazer um gráfico de barras no *R-Commander* para a variável *tipo* (variável qualitativa relacionada com o tipo de aplicação da peça: (c) coluna ou (d) dentro).



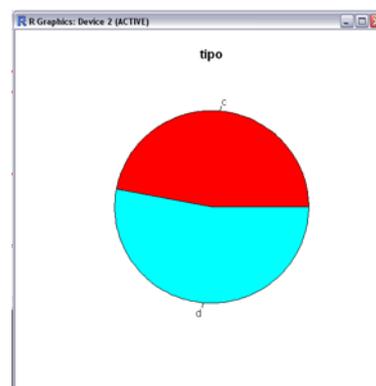
The image shows the R Commander interface. The 'Graphs' menu is open, and 'Bar graph...' is selected. The 'Bar Graph' dialog box is open, showing the variable 'tipo' selected. Below, the 'R Graphics: Device 2 (ACTIVE)' window displays a bar chart with two bars for categories 'c' and 'd'. The y-axis is labeled 'Frequency' and ranges from 0 to 15. The bar for 'c' has a frequency of approximately 15, and the bar for 'd' has a frequency of approximately 16.

Para fazer um gráfico de barras recorra à opção 'Graphs/Bar Graph' e escolha, depois, a variável qualitativa que pretende representar

O comando gerado pelo *R-Commander* que permite fazer este gráfico directamente no R é o seguinte:

```
>barplot(table(Dataset$tipo), xlab="tipo", ylab="Frequency")
```

Para construir um diagrama circular, igualmente adequado a este tipo de dados, o procedimento é idêntico, excepto na opção de gráficos, onde se deve escolher *pie chart* em vez de *bar graph*. O

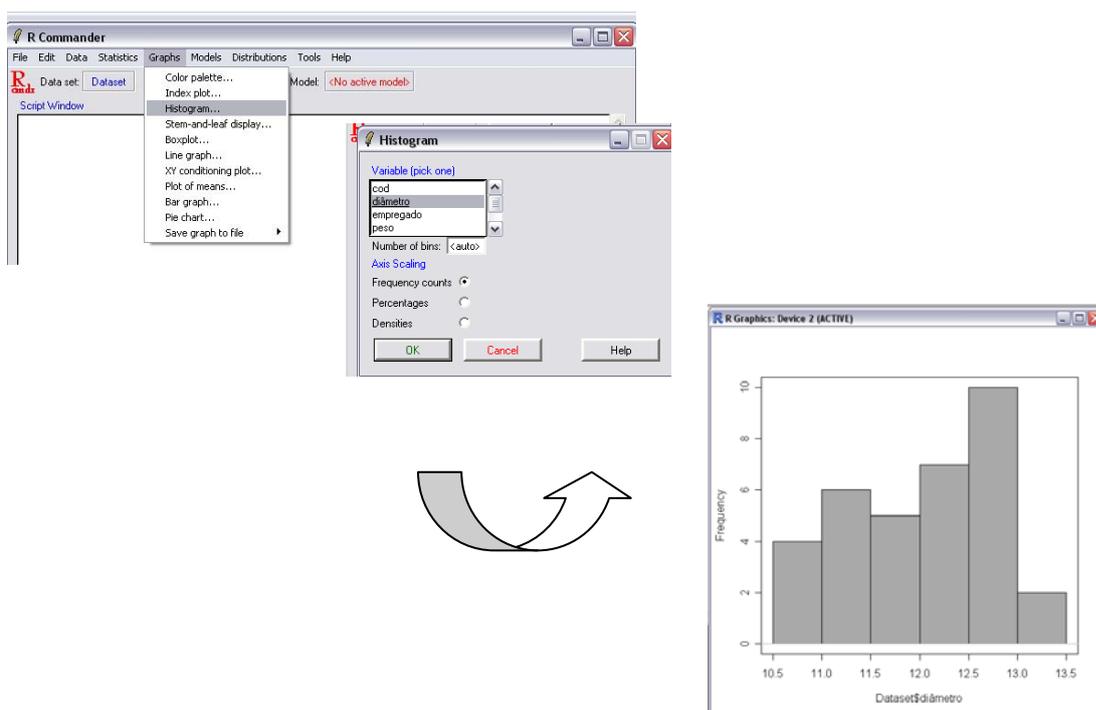


comando gerado no R é o seguinte:

```
>pie(table(Dataset$tipo),labels=levels(Dataset$tipo),main="tipo",col=rainbow(length(levels(Dataset$tipo))))
```

Histograma

O histograma é uma das formas mais importantes de representar dados quantitativos. Para se fazer um histograma é necessário começar por agrupar as observações em classes e depois representar, para cada classe, uma barra cuja altura seja proporcional ao número de observações. Uma vez que as classes ou intervalos em que os dados são agrupados são contíguas, as barras são apresentadas sem separação. Para fazer um histograma no *R-Commander* considerando a variável *diâmetro* proceda como se indica na figura:



O comando gerado pelo *R-Commander* que permite fazer este gráfico directamente no R é o seguinte:

```
>hist(Dataset$diâmetro, scale="frequency", breaks="Sturges", col="darkgray")
```

Diagrama de pontos

Também conhecido por diagrama de dispersão, o gráfico de pontos é muito adequado nos casos em que pretendemos representar duas variáveis quantitativas (discretas ou contínuas), particularmente quando pretendemos analisar a sua correlação.

The image shows the R Commander interface. The 'Graphs' menu is open, and 'XY conditioning plot...' is selected. The 'XY Conditioning Plot' dialog box is open, showing 'diâmetro' and 'peso' as response variables and 'tipo' as a grouping variable. The 'X-Axis Scales in Different Panels' and 'Y-Axis Scales in Different Panels' are both set to 'Identical'. The 'OK' button is highlighted. Below the dialog box, a curved arrow points to the resulting scatter plot in the 'R Graphics: Device 2 (ACTIVE)' window. The plot shows a positive correlation between 'diâmetro' (x-axis, ranging from 10.5 to 13.0) and 'peso' (y-axis, ranging from 20 to 25).

O comando gerado pelo *R-Commander* que permite fazer este gráfico directamente no R é:

```
> xyplot(peso~diâmetro,auto.key=TRUE,scales=list(x=list(relation='same'),
y=list(relation='same')), data=Dataset)
```

7. Exemplos de Aplicação

Este capítulo contém alguns exercícios de aplicação imediata e problemas resolvidos através do R tais como: “Número de irmãos dos alunos da turma H do 9º ano”, “Alturas dos Alunos”, “Construir um Triângulo”, “Uma Corrida Com Dados” e “Resultados de um teste” (este último associado ao programa PISA).

Pensamos que estes exercícios e problemas ajudam a aprofundar os conhecimentos de R apresentados neste dossiê, sendo que, para a sua resolução, se utilizaram conceitos que são usualmente trabalhados no ensino básico e secundário.

Número de irmãos dos alunos da turma H do 9º ano⁵

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

Construa:

- a) a tabela de frequências.
- b) o diagrama de barras

Resolução com R:

a) Para construir a tabela de frequências:

> `cbind(fa=table(dados), fr=prop.table(table(dados)))`

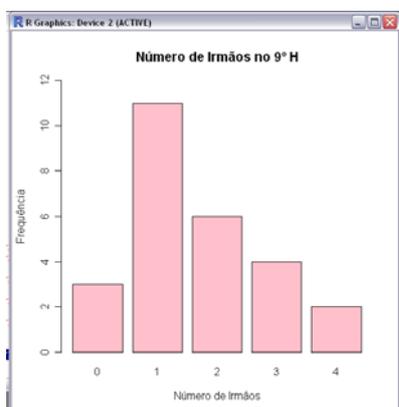
```

R Console
> cbind(fa=table(dados), fr=prop.table(table(dados)))
  fa      fr
0  3 0.11538462
1 11 0.42307692
2  6 0.23076923
3  4 0.15384615
4  2 0.07692308
> |
    
```

⁵ Exercício 1 da página 3 (tabelas de frequência e histograma) do ALEA: http://www.alea.pt/html/nocoos/html/cap7_2_1.html

b) Para construir o diagrama de barras:

```
> barplot(table(dados), main="Número de Irmãos no 9º H", xlab="Número de Irmãos",
ylab="Frequência", col=rep("pink",5), ylim=c(0,12))
```



Alturas dos Alunos⁶

Para este exercício, foram registadas as alturas, em centímetros, dos alunos de uma turma do 10º ano:

150	169	174	155	165	170	172
152	158	163	158	166	158	166
170	171	162	171	161	154	168
161	164	166	164	162	156	167

Construa uma tabela de frequências, agrupando os dados em classes e represente graficamente os dados, utilizando o tipo de gráfico que achar mais conveniente. Faça ainda um diagrama de caule-e-folhas.

⁶ Exercício 2 da página 4 (tabelas de frequência e histograma) do ALEA:
http://www.alea.pt/html/nocoos/html/cap7_2_2.html



Resolução com R:

- O primeiro passo consiste em transmitir os dados ao R. Para tal, podemos criar um ficheiro com estes dados (exercício1.csv) ou lê-los através de um vector.

```
> dados<-read.csv("Exercicio1.csv")
```

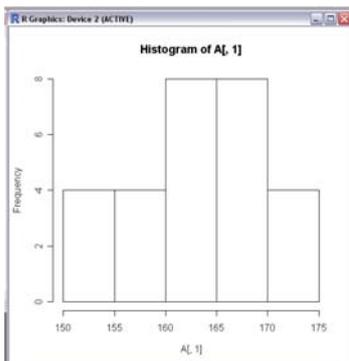
ou

```
>dados<-
```

```
c(150,169,174,155,165,170,172,152,158,163,158,166,158,166,170,171,162,171,161,154,168,161,164,166,164,162,156,167)
```

- De seguida aplicamos o comando *hist*.

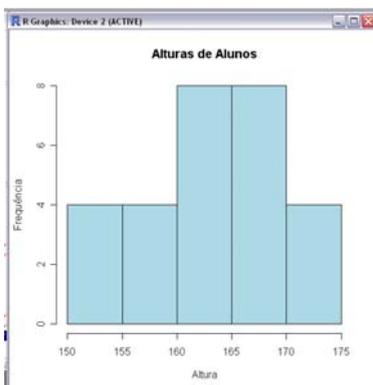
```
> hist(dados[,1])
```



Para formatar melhor o gráfico, podemos recorrer aos parâmetros do comando *hist*:

```
> hist(A[,1],breaks="Sturges", col="light blue",  
xlab="Altura", ylab="Frequência", main="Alturas de Alunos")
```

E o resultado é...



A partir do comando do histograma, poderemos construir uma tabela de frequências.

Para tal, basta guardar o resultado do comando *hist*.

```
> s<- hist(dados[,1])
```

```
> s
```

```
$breaks
```

```
[1] 150 155 160 165 170 175
```

```
$counts
```

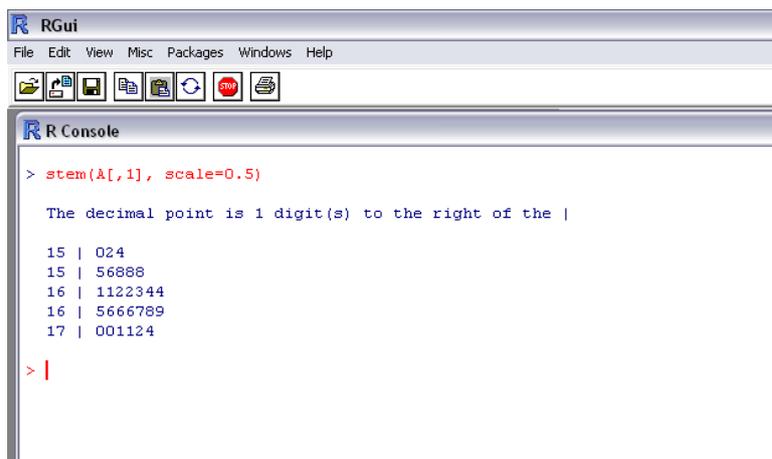
```
[1] 4 4 8 8 4
```

```
$intensities
```

```
[1] 0.02857142 0.02857143 0.05714286 0.05714286 0.02857143
```

```
(...)
```

Para fazer um diagrama de caule-e-folhas⁷ deveremos aplicar o comando *stem*:



```
RGui
File Edit View Misc Packages Windows Help
[Icons]

R Console
> stem(A[,1], scale=0.5)

The decimal point is 1 digit(s) to the right of the |

15 | 024
15 | 56888
16 | 1122344
16 | 5666789
17 | 001124

> |
```

⁷ Para saber mais sobre este tipo de gráfico consulte o AELA em:
http://www.alea.pt/html/nocoes/html/cap3_2_20.html

Construir um triângulo...

Um segmento de comprimento unitário é dividido em 3 partes, aleatoriamente. Qual a probabilidade de as partes resultantes poderem formar um triângulo?

A resolução deste problema prende-se com uma regra que estabelece que a soma dos comprimentos de dois lados de um triângulo é superior ao comprimento do outro lado.

Nota – Quando se fala em números aleatórios, estamos intuitivamente a pensar em números com uma distribuição uniforme, no intervalo $[0,1]$.

Resolução do problema por simulação no R:

Vamos fazer um determinado número de simulações e calcular a frequência relativa das situações que dão origem a triângulos.

Para tal, vamos gerar dois números aleatórios entre 0 e 1 e estes números irão representar os pontos P e Q em que um segmento $[MN]$ de comprimento 1 fica dividido:



Vamos considerar para P o menor dos valores obtidos, que será o comprimento de MP. Calcula-se o comprimento dos segmentos PQ e QN e depois testa-se se dois quaisquer dos comprimentos obtidos são superiores ao terceiro comprimento. Terminado o número de simulações, calcula-se o número das situações que dão origem a triângulos e divide-se pelo número de simulações.

Script 1 “Problema do triângulo”

```

cont=0
NumSim=1000
segmentos=array(0,dim=c(NumSim,3))
for (i in 1:NumSim) {
  M=0
  N=1
  A=runif(1,0,1)
  B=runif(1,0,1)
  MP=min(A,B)
  PQ=abs(A-B)
  QN=1-max(A,B)
  if (MP+PQ > QN & MP+QN>PQ & PQ+QN>MP) cont=cont+1
  segmentos[i,1]=MP
  segmentos[i,2]=PQ
  segmentos[i,3]=QN
}
cat("frequência relativa",cont/NumSim)
    
```

Por exemplo, pedindo 1000 simulações, obteve-se:

Frequência relativa de triângulos 0.256

Acrescentando ao script anterior, o cálculo do comprimento médio de cada segmento nos casos em que é possível construir um triângulo:

Script 2 “Problema do triângulo “

```

cont=0
NumSim=1000
segmentos=array(0,dim=c(NumSim,3))
for (i in 1:NumSim) {
  M=0
  N=1
  A=runif(1,0,1)
  B=runif(1,0,1)
  MP=min(A,B)
  PQ=abs(A-B)
  QN=1-max(A,B)
  if (MP+PQ > QN & MP+QN>PQ & PQ+QN>MP) {
    cont=cont+1
    segmentos[cont,1]=MP
    segmentos[cont,2]=PQ
    segmentos[cont,3]=QN
    par(mfrow=c(2,2))
    cor1=c("blue")
    cor2=c("pink")
    cor3=c("yellow")
  }
}
segmentos=segmentos[1:cont,]
hist(segmentos[,1],col=cor1,xlab="comprimento",ylab="frequência",main="Segmento MP")
hist(segmentos[,2],col=cor2,xlab="comprimento",ylab="frequência",main="Segmento PQ")
hist(segmentos[,3],col=cor3,xlab="comprimento",ylab="frequência",main="Segmento QN")

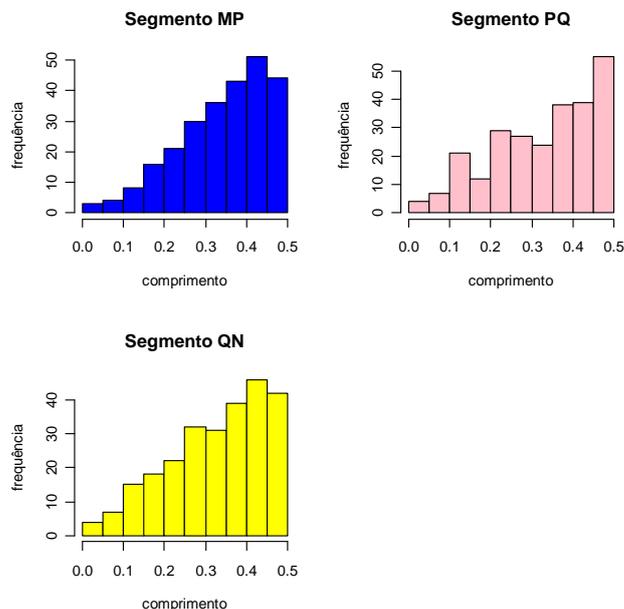
cat("frequência relativa de triângulos",cont/NumSim)
cat("comprimento médio do segmento MP",mean(segmentos[,1]))
cat("comprimento médio do segmento PQ",mean(segmentos[,2]))
cat("comprimento médio do segmento QN",mean(segmentos[,3]))
    
```

Fizemos nova simulação e obtivemos:

Comprimento médio do segmento MP: 0.3432921

Comprimento médio do segmento PQ: 0.3286406

Comprimento médio do segmento QN: 0.3280673



“Curiosamente” o comprimento médio dos segmentos aproxima-se de 1/3.

Efectuando maior número de simulações, a frequência relativa dos casos em que é possível construir um triângulo aproxima-se de 0,25 e o comprimento médio dos segmentos desses triângulos é um valor próximo de 0,33.

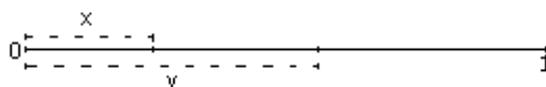
Voltando à simulação no R...

O script elaborado inicialmente pareceu-nos o processo mais indicado para ser explicado aos alunos, mas explorando um pouco mais as potencialidades do R, fizemos um novo script tendo por base o seguinte raciocínio: considere-se duas variáveis aleatórias X e Y (com distribuição uniforme no intervalo [0,1]) e independentes:

- X tem distribuição uniforme no intervalo [0,1]
- Y tem distribuição uniforme no intervalo [0,1]

Quando se seleccionam 2 números, um com distribuição X e outro com distribuição Y, podemos ter uma de duas situações: $X < Y$ ou $X > Y$.

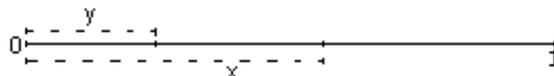
$X < Y$



Lados do (suposto) triângulo:

$$X \quad (Y-X) \quad (1-Y)$$

$X > Y$



Lados do (suposto) triângulo:

$$Y \quad (X-Y) \quad (1-X)$$

Para que possam, efectivamente, ser os lados de um triângulo, cada lado tem de ser inferior à soma dos outros dois⁸.

Neste novo script indicamos apenas o número de simulações desejadas e obtemos graficamente a evolução da frequência relativa, dos casos em que é possível construir um triângulo, observando-se em simultâneo a frequência relativa para os quartis do número n de simulações indicadas.

Script 3 “Problema do triângulo “

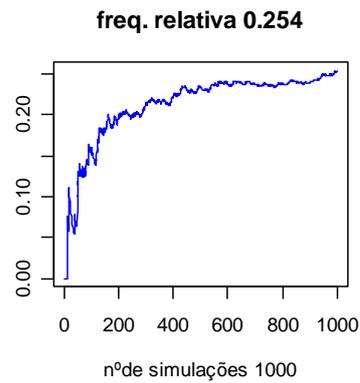
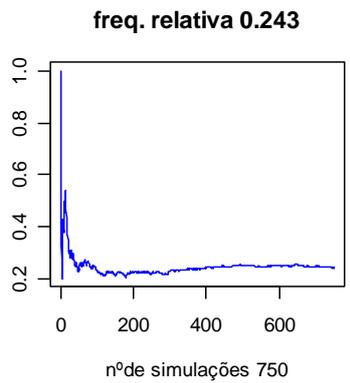
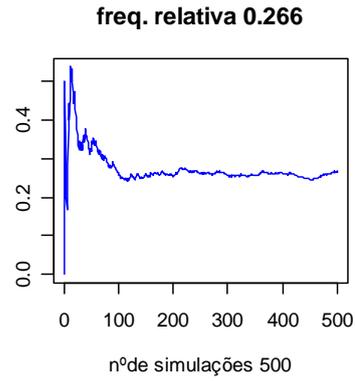
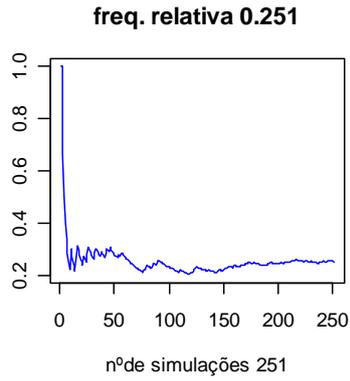
```
grafico=function(n) {
  calculo=function(n) {

    x=runif(n,0,1)
    y=runif(n,0,1)
    cond=((x>1/2 & (x-y)<1/2 & y<1/2) | (x<1/2 & (y-x)<1/2 & y>1/2))
    v=round(sum(cond)/n,3)
    color=c("blue")
    plot(1:n,col=color,cumsum(cond)/(1:n), type="l",main=paste("freq.
    relativa",v ), xlab=paste("n°de simulações",
    round(quantile(n),0)[i]),ylab="")

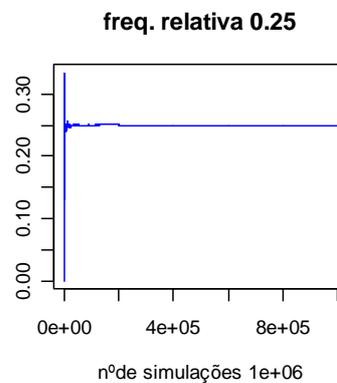
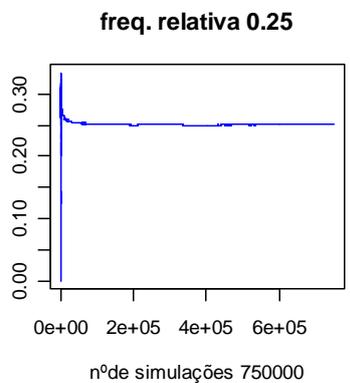
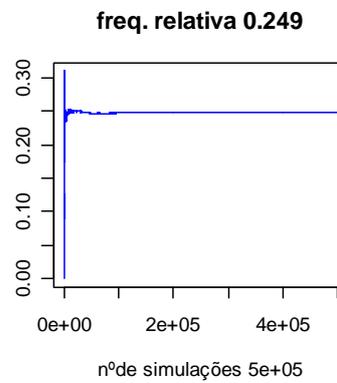
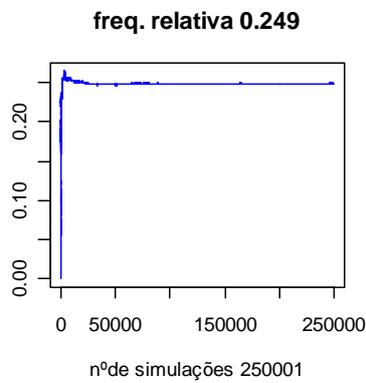
  }
  par(mfrow=c(2,2))
  for (i in 2:5) {calculo(round(quantile(1:n),0)[i])}
}
```

Por exemplo, digitando `grafico(1000)` (designamos a nossa função no R por *grafico*), obtivemos os resultados para 1000 simulações, ilustradas na figura seguinte:

⁸ Ver explicação mais detalhada em <http://www.alea.pt/html/probabil/html/probabilidades.html>



Para 1 000 000 simulações:



Aumentando o número de simulações, a frequência relativa tende a estabilizar à volta do valor 0,25, o que vem comprovar a definição frequencista do conceito de probabilidade: a probabilidade de um determinado acontecimento é o valor obtido para a frequência relativa com que se observou esse acontecimento, num grande número de realizações da experiência aleatória.

Uma Corrida com Dados

O Bruno arranjou um dado especial com a forma de um dodecaedro. Tem 12 faces, numeradas de 1 a 12.

A Tânia tem dois dados normais. São cubos, cada um deles com as faces numeradas de 1 a 6.

Resolveram fazer um jogo. Cada jogada consiste no lançamento dos três dados.

Vão somando os pontos que cada um obtém: o Bruno com o seu dado de 12 faces e a Tânia com os seus dois dados de 6 faces. Ganha quem primeiro chegar aos 100 pontos.

Se por acaso os dois chegarem aos 100 pontos na mesma jogada, ganha quem tiver o total maior. Se esse total for igual para os dois, há empate.



Algum dos jogadores está em vantagem? Ou é o jogo equilibrado?

(Desafios do Público)

Antes da realização das experiências cada elemento do grupo conjecturou sobre quem teria maior probabilidade de vencer, se o Bruno lançando o dodecaedro, se a Tânia lançando dois dados cúbicos. Surgiram opiniões diversas:

- A Tânia obtém, no mínimo, por jogada, dois pontos enquanto que o Bruno pode obter um;
- No dodecaedro a probabilidade de sair **doze** é $\frac{1}{12}$ que é maior que $\frac{1}{36}$, correspondente à probabilidade do mesmo resultado no caso dos dados cúbicos;
- A probabilidade de obter **seis** é maior no lançamento dos dois dados cúbicos, $\frac{5}{36}$, contra $\frac{1}{12}$ no dodecaedro; essa vantagem acentua-se mais no caso da

obtenção do valor **sete** ao qual corresponde as probabilidades $\frac{1}{6}$, nos dados cúbicos, e $\frac{1}{12}$ no outro dado.

Script 1 "Corrida de Dados" em R

```
#Simular um jogo da corrida de dados
L=1
AcumCubico=0
AcumDode=0
while (AcumCubico<100 & AcumDode<100) {
  AcumCubico=AcumCubico+round(runif(1,1,6))+round(runif(1,1,6))
  AcumDode=AcumDode+(round(runif(1, 1, 12)))
  L=L+1
}
if (AcumCubico>AcumDode) print ("Foi o par de dados cubicos") else if
(AcumDode>AcumCubico) print ("Foi o dodecaedro") else if
(AcumCubico==AcumDode) print ("Empate")
print (paste("Total de jogadas", L))
print (paste("Total de pontos dos dados cúbicos", AcumCubico))
print (paste("Total de pontos do dodecaedro", AcumDode))
```

Começamos por elaborar um script para a simulação de um jogo:

Na simulação que realizámos o resultado foi o seguinte: venceu “o par de dados cúbicos”, realizaram-se “16 jogadas”, sendo o total dos pontos dos dados cúbicos “107” e o total de pontos do dodecaedro “105”.

Elaborámos um outro script para simular vários jogos:

Script 2 "Corrida de Dados" em R

```
#Simular vários jogos da corrida de dados
dados=function(n) {
  CUBICO=0
  DODE=0
  EMPATE=0
  for (i in 1:n) {
    L=1
    AcumCubico=0
    AcumDode=0
    while (AcumCubico<100 & AcumDode<100) {
      AcumCubico=AcumCubico+round(runif(1,1,6))+round(runif(1,1,6))
      AcumDode=AcumDode+(round(runif(1, 1, 12)))
      L=L+1
    }
    if (AcumCubico>AcumDode) CUBICO=CUBICO+1 else if (AcumDode>AcumCubico)
    DODE=DODE+1 else if (AcumCubico==AcumDode) EMPATE=EMPATE+1
  }
  print (paste("Freq.relativa do n.ºde vezes em que os dados cubicos
  ganharam", CUBICO/n))
  print (paste("Freq.relativa do n.ºde vezes em que o dodecaedro ganhou",
  DODE/n))
  print (paste("Freq.relativa do n.ºde empates", EMPATE/n))
}
```

Executado o script para simular 100 jogos, digitamos na consola do R "dados (100)" e obtivemos:

"Freq. relativa do n.º de vezes em que os dados cúbicos ganharam 0.67"

"Freq. relativa do n.º de vezes em que o dodecaedro ganhou 0.32"

"Freq. relativa do n.º de empates 0.01"

Se o número de experiências for suficientemente grande, a percentagem de cada resultado estará próxima do valor real da probabilidade (Lei dos Grandes Números).

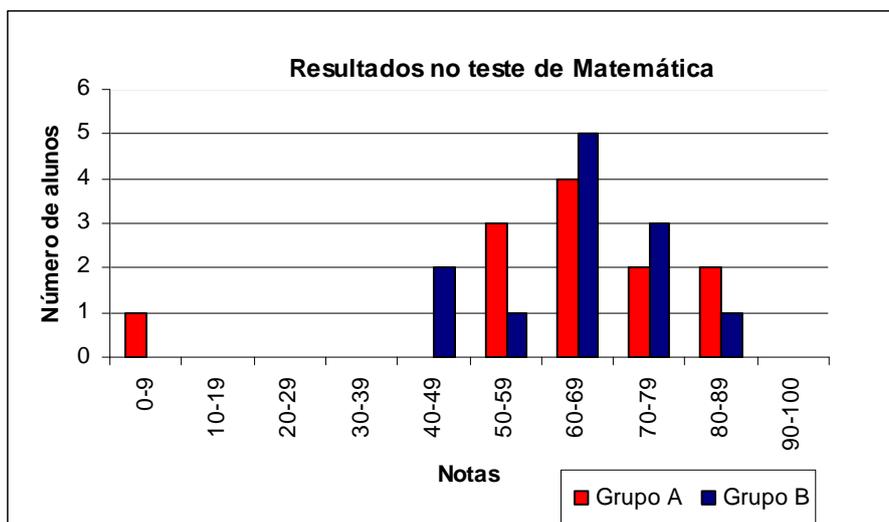
Simulámos no R, um milhão de jogos e ao fim de alguns minutos... obtivemos:

- "Freq. relativa do n.º de vezes em que os dados cúbicos ganharam 0.676556";
- "Freq. relativa do n.º de vezes em que o dodecaedro ganhou 0.304982";
- "Freq. relativa do n.º de empates 0.018462".

Assim, a probabilidade de a Tânia ganhar será aproximadamente 67,7% e a do Bruno 30,5%. A probabilidade de empate é de 1,8%. Claro que estes não são valores exactos... mas estarão próximos dos valores reais.

Resultados de um Teste

O gráfico seguinte mostra os resultados de um teste de Matemática obtidos por dois grupos de alunos, designados por "Grupo A" e "Grupo B". A **nota média** no grupo **A** é de **62,0** e no grupo **B** de **64,5**. Os alunos passam neste teste se tiverem uma nota igual ou superior a 50.



Com base nesta informação, o professor concluiu que o grupo B teve melhores resultados neste teste que o grupo A.

Os alunos do grupo A não estão de acordo com o professor. Tentam convencer o professor de que o grupo B não teve necessariamente melhores resultados.

Utilizando a informação dada, apresente pelo menos um argumento matemático que possa ser utilizado pelos alunos do grupo A.

(adaptado do Programa para a Avaliação Internacional de Alunos 2003, PISA – *Programme for International Student Assessment*)

Argumentos que podem ser utilizados:

- Há mais alunos que passaram no teste no Grupo A do que no Grupo B (há mais “positivas” no Grupo A do que no Grupo B);
- O Grupo A tem mais alunos com nota igual ou superior a 80 que o grupo B;
- Se ignorarmos o aluno mais fraco do Grupo A, os alunos do Grupo A têm melhores resultados que os do grupo B.

Respeitando a informação dada no problema, consideremos que os resultados obtidos pelos dois grupos foram os seguintes:

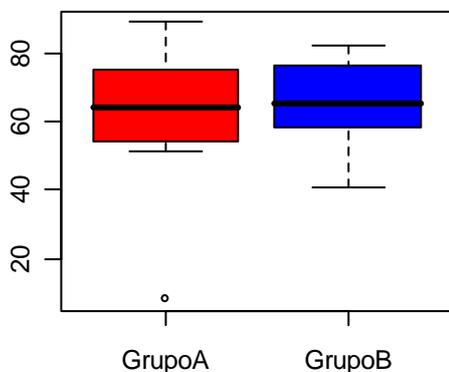
Grupo A: 8, 51, 52, 56, 61, 63, 65, 67, 74, 76, 82, 89

Grupo B: 41, 43, 55, 61, 62, 63, 67, 68, 74, 79, 79, 82

Utilizando o programa R⁹, calculemos as principais estatísticas descritivas destes dois grupos, bem como os respectivos boxplots (caixas de bigodes):

GrupoA

Min. : 8.0
 1st Qu.:55.0
 Median :64.0
 Mean :62.0
 3rd Qu.:74.5
 Max. :89.0

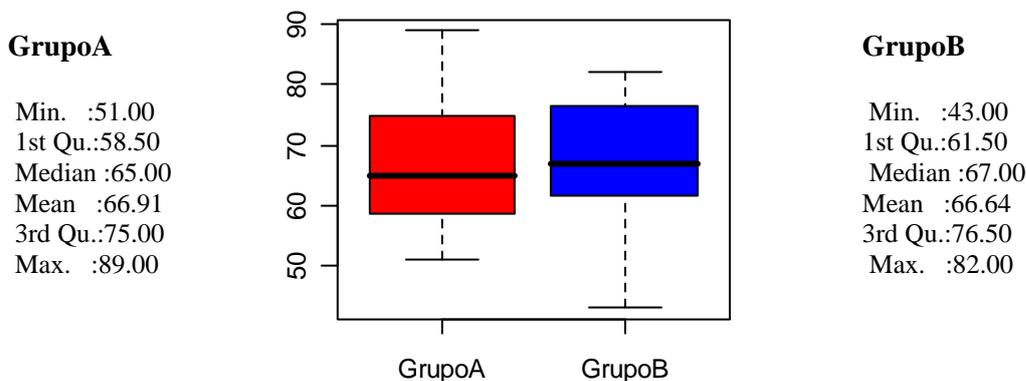


GrupoB

Min. :41.00
 1st Qu.:59.50
 Median :65.00
 Mean :64.50
 3rd Qu.:75.25
 Max. :82.00

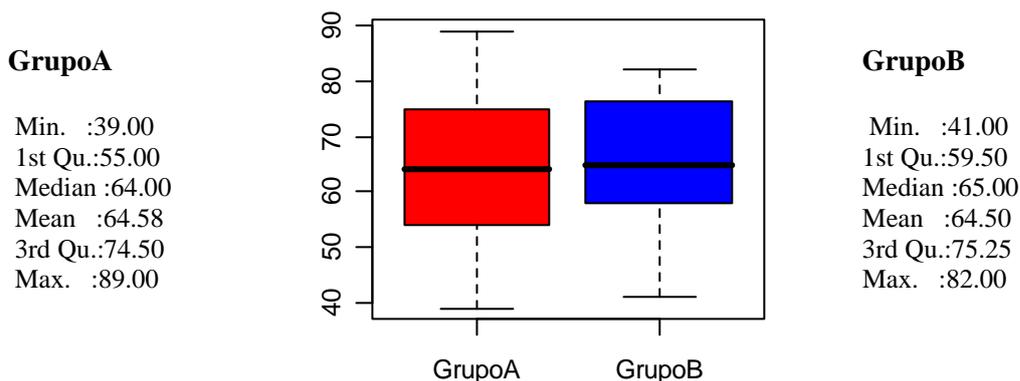
⁹ Ver script elaborado no final deste problema.

Note-se que a nota mais baixa do Grupo A, que se afasta significativamente das restantes (*outlier*), está assinalada com um (ponto). Este valor interfere bastante na média dos resultados do Grupo A. Efectivamente, se retirarmos a nota mais baixa a cada um dos grupos, respectivamente 8 e 41, obtemos:



Com esta alteração obtemos uma melhor leitura do gráfico, dada a redução na dispersão dos dados. Confirma-se assim uma subida das estatísticas descritivas, em particular no Grupo A, em que a média das notas do Grupo A supera a média das notas do Grupo B.

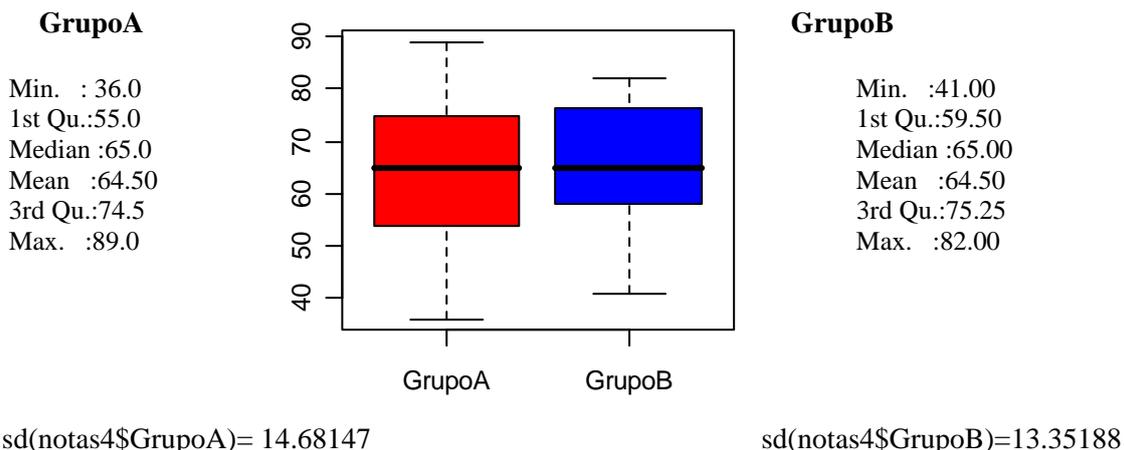
Retomando as doze notas iniciais de cada grupo, alteremos agora apenas o menor valor do Grupo A, a nota 8 para 39 (nota mínima, de qualquer modo inferior à nota mínima do Grupo B).



A alteração do valor extremo teve como consequência uma subida significativa da média, mantendo-se, o valor da mediana. Esta situação ilustra bem a maior resistência da mediana a valores extremos relativamente à média.

Apesar da importância destas duas medidas de tendência central, poderemos ter um conjunto de dados diferentes com igual média e mediana, sendo necessário recorrer a outras medidas estatísticas para analisar melhor os dados.

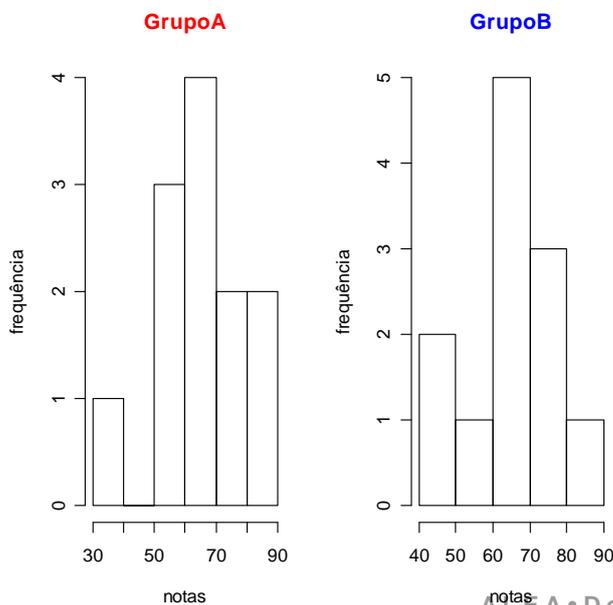
Ainda na situação apresentada, se alterarmos no Grupo A, por exemplo, duas notas: 8 para 36 e 63 para 65, obtemos:



A média e a mediana são iguais, sendo por isso necessário recorrer as outras medidas, por exemplo, de dispersão para analisarmos melhor os dados e concluir, eventualmente qual dos grupos tem melhores resultados.

No Grupo A a amplitude interquartil é superior, bem como o desvio padrão o que significa que neste grupo existe uma maior variabilidade das notas em relação à média.

Os histogramas destes conjuntos de dados apresentam-se a seguir:



Script “Resultados de um teste”

```
notas=data.frame(GrupoA=c(8,51,52,56,61,63,65,67,74,76,82,89),GrupoB=c(41,43,55,61,62,63,67,68,74,79,79,82))
summary(notas)
par(mfrow=c(2,2))
color=c("red","blue")
boxplot(notas,col=color)

notas2=data.frame(GrupoA=c(51,52,56,61,63,65,67,74,76,82,89),GrupoB=c(43,55,61,62,63,67,68,74,79,79,82))
summary(notas2)
boxplot(notas2,col=color)

notas3=data.frame(GrupoA=c(39,51,52,56,61,63,65,67,74,76,82,89),GrupoB=c(41,43,55,61,62,63,67,68,74,79,79,82))
summary(notas3)
boxplot(notas3,col=color)

notas4=data.frame(GrupoA=c(36,51,52,56,61,65,65,67,74,76,82,89),GrupoB=c(41,43,55,61,62,63,67,68,74,79,79,82))
summary(notas4)
boxplot(notas4,col=color)
sd(notas4$GrupoA)
sd(notas4$GrupoB)
# histogramas do problema Resultados de um teste
par(mfrow=c(1,2))
color=c("red")
hist(notas4$GrupoA,main="GrupoA",xlab="notas",ylab="frequência",col.main=col
or)
color=c("blue")
hist(notas4$GrupoB,main="GrupoB",xlab="notas",ylab="frequência",col.main=col
or)
```

8. Para saber mais: recursos práticos para aprendizagem do R

- ALEA, Dossiê X – “Software Estatístico - Uma introdução a alguns aplicativos, numa abordagem inicial dos dados”, Helder Alves, Luís Cunha.
- Ponte, João Pedro da, Introdução, in Seymour Papert, “A Família em rede”, Relógio d'Água, 1997.
- ALEA, Dossiê X – “Software Estatístico - Uma introdução a alguns aplicativos, numa abordagem inicial dos dados”, Helder Alves, Luís Cunha.
- L. Torgo (2009), A Linguagem R – Programação para a Análise de Dados, Escola Editora.
- Paul Murrell (2006), R Graphics, Chapman & Hall/CRC, London.
- Peter Dalgard (2002), Introductory Statistics with R, Springer, New York.

- **The R Project for Statistical Computing:**
<http://www.r-project.org/index.html>
- **R Site Search:**
<http://finzi.psych.upenn.edu/search.html>
- **R mailing lists archive:**
<http://tolstoy.newcastle.edu.au/R/>
- **The R Commander – A Basic-Statistics GUI for R:**
<http://socserv.mcmaster.ca/jfox/Misc/Rcmdr/>
- **Tinn-R:**
<http://www.sciviews.org/Tinn-R/>

