Exercícios 4

Manipulação de ficheiros e de dados

Em termos computacionais, a informação é habitualmente transmitida em blocos a que denominamos ficheiros. O **R** disponibiliza uma vasta colecção de funções para manipular ficheiros de diferentes formatos. Neste exercício, pretende-se fazer uso de algumas funções de manipulação de ficheiros, e dos objectos resultantes do seu carregamento numa sessão de **R**.

Para treinar esses conceitos, considere o papel de um analista de dados desportivos. A sua tarefa é recolher, preparar e analisar um conjunto de dados da *Primeira Liga* de futebol portuguesa (Figura 1).

Figura 1: Preferências clubísticas à parte, eis o que os dados revelam.

Recorde algumas das funções que lhe poderão ser úteis na resolução deste tipo de problemas (use o sistema de ajuda para verificar como as usar):

Operações matemáticas comuns mean, sum, colMeans, rowMeans

Outros estatísticos min, max, sd, var

Propriedades de objectos tabulares nrow, ncol, names, colnames, rownames

Sumarizar e contar summary, str, table, margin.table, which, which.max, which.min

Aplicar funções sobre objectos apply, sapply, tapply

Ordenação sort, order, rank

Agregação e manipulação by, aggregate, transform

Manipular strings substr, strsplit, paste, nchar

1. Recolha os dados existentes na seguinte tabela das últimas 25 temporadas:

   `http://www.european-football-statistics.co.uk/atc/atcprt.htm`

   Esta tabela contém alguns dados sobre as 43 equipas que participaram nessas temporadas. Os dados estão agrupados nas seguintes variáveis:

   No. Posição

   Club Nome da equipa

   G Jogos

   W Vitórias

   D Empates

L Derrotas

P Pontos

F Golos marcados

A Golos sofridos

S Temporadas

1 Vencedor

2 Segundo lugar

3 Terceiro lugar

Cup Taças de Portugal

Existem várias formas possíveis de recolher dados de uma página na *World Wide Web*. Uma delas, usando o **R**, e para o caso de os dados estarem em forma de uma tabela, é através da função `readHTMLTable(url)`, que é disponibilizada no pacote (*package*) XML.

Nota: O **R** já vem com alguns pacotes instalados. Pode instalar um novo pacote usando a função `install.packages('nome-do-pacote')`. Para importar o conteúdo de um pacote para a sessão actual, usa-se a função `library(nome-do-pacote)`. Repare que no primeiro caso o nome do pacote é passado como `character`, enquanto que no segundo caso não.

2. Verifique a estrutura da operação anterior. Irá reparar que a função `readHTMLTable()` procura ler todas as tabelas existentes numa página, mesmo as que não sejam óbvias para o leitor, mas que sejam tabelas por definição (código HTML). Guarde a tabela de interesse numa variável chamada `liga`.

3. Uma vez que os dados estejam importados, é necessário confirmar se estão no formato certo e que não há irregularidades.

   Parece que a coluna 8 não é mais do que um artefacto na tabela. Remova-a da tabela.

4. Verifique o tipo de dados em cada uma das colunas. Corrija as colunas que não estiverem no formato mais adequado (por exemplo, uma coluna numérica estar no formato de texto).

5. Frequentemente, os dados incluem erros ou inconsistências na sua formatação. Esta tabela é mais um exemplo disso. Essas incongruências têm que ser tratadas antes de se proceder à análise de dados. Repare nas colunas **F**, **A** e **P**. Qual o problema que identifica?

   Depois de ter identificado o problema, reconhecerá a utilidade do seguinte exemplo para o resolver.

   ```
   > x = c("aaa", "bb.b", "cc,c")
   > x = gsub("\\.", "", x)
   ```

   Nota: É possível resolver este exercício e o anterior numa só vez.

6. Concluída a preparação dos dados, há que passar à análise. Quais são as equipas com mais derrotas do que a média?

7. Acrescente uma nova coluna com a média de golos por jogo de cada equipa.

8. Como acrescentou uma coluna ao conjunto de dados, já vale a pena guardar os dados num ficheiro. Guarde-os num novo ficheiro com a seguinte especificação:

   Nome PrimeiraLiga_25anos.dat

   Separador tabulação

   Separador decimal vírgula

   Valores em falta (`NA`) hífen

   Nome das colunas sim

Nome das linhas não

9. Nem sempre um maior número de pontos corresponde a um maior número de vitórias. Identifique as equipas em que a sua posição na tabela (está ordenada por pontos) não corresponde à posição se a tabela estivesse ordenada por vitórias.

10. Qual é a equipa que tem mais golos marcados do que sofridos, e que simultaneamente tem menos pontos?

    Sugestão: Acrescente primeiro uma coluna que diz se a equipa tem ou não mais golos marcados do que sofridos.

11. Haverá uma relação entre o número de golos e o número de temporadas, isto é, se, em média, as equipas com mais experiência na *Primeira Liga* tendem a marcar mais golos?

    Sugestão: Determine primeiro qual a média de golos marcados por cada valor de experiência (número de épocas jogadas).

12. Normalmente os clubes são mais conhecidos por uma só palavra. Por exemplo, o clube *SC de Braga* é conhecida por *Braga*, e o clube *A Naval 1º de Maio* é mais conhecido apenas por *Naval*.

    Pretende-se alterar o nome dos clubes para a denominação mais comum. De modo a não alterar um a um manualmente, considere o seguinte algoritmo (não é perfeito, mas funciona bem na maior parte dos casos):

    1. Separar o nome por palavras;
    2. Escolher a palavra com maior número de caracteres;
    3. Usar essa palavra como nome para a equipa.

    Por exemplo, o clube com o nome `"CF Os Belenenses"` pode ser separado num vector com 3 palavras:

    ```
    [1] "CF"         "Os"         "Belenenses"
    ```

    Estas palavras têm os seguintes números de caracteres:

    ```
    [1]  2  2 10
    ```

    Neste vector, o valor mais alto está na posição:

    ```
    [1] 3
    ```

    Então, voltando ao vector com as palavras separadas, escolhe-se a palavra nessa posição:

    ```
    [1] "Belenenses"
    ```

    Nota: Neste exercício, há uma função que pode ser particularmente útil, caso utilize listas como um resultado intermediário: `unlist()`. Esta função simplifica uma lista para a forma de um vector.