

Exercícios 2

1 Factores

Em diversos contextos, pode ser necessário lidar com um tipo de dados que não seja numérico, mas que o seu valor possa ser avaliado qualitativamente.

Por exemplo, num inquérito feito aos alunos, considere a questão: “O que acha dos docentes de CEGI?”, tendo as seguintes hipóteses de escolha “a) Terríveis”, “b) Meh.”, “c) über-cool!”. Este tipo de dados é então denominado categórico ou nominal. Neste caso em particular, têm ainda uma relação de ordem:

“Terríveis” < “Meh.” < “über-cool!”.

Em **R**, existe um tipo de dados indicado para estes casos, mais conveniente e vantajoso do que apenas ter um vector de caracteres: **factor**.

1. Fez-se um levantamento de algumas características fisionómicas dos alunos de uma turma. Crie um vector de factores, `cor.olhos`, a partir do seguinte vector:

```
c("azul", "castanho", "castanho", "castanho", "verde", "azul", "castanho")
```

2. Verifique os níveis (categorias) criados.
3. Um outro conjunto de alunos foi registado no seguinte vector:

```
c("castanho", "castanho", "azul")
```

Defina os factores para este vector, mantendo os mesmos níveis.

4. Considerando ainda os mesmos níveis, transforme em factores os dados de um terceiro grupo de alunos:

```
c("azul", "castanho", "cinzento", "castanho")
```

O que aconteceu ao elemento na 3ª posição?

5. No vector `gen` registou-se o género dos alunos:

```
gen = c("m", "f", "m", "m", "f", "f", "m")
```

Crie uma tabela de contingência¹, mostrando a cor dos olhos por sexo.

6. Crie duas tabelas de frequências marginais, uma mostrando o número total de homens e mulheres, outra mostrando as contagens pela cor dos olhos.
7. Selecciona os primeiros três elementos do vector `cor.olhos`. O que aconteceu aos níveis? Como manter apenas os níveis dos elementos seleccionados?
8. Por fim, guardou-se a informação contendo a altura dos alunos no vector `alturas`. Esta informação é qualitativa, mas tem relação de ordem.

```
alturas = c("baixo", "baixo", "médio", "alto", "alto", "médio", "médio")
```

Transforme em factores, mantendo a relação de ordem. Compare a altura entre dois alunos.

¹https://pt.wikipedia.org/wiki/Tabela_de_conting%C3%Aancia

2 Matrizes

1. Uma matriz é uma extensão de um vector para duas dimensões. Na verdade, em **R**, um vector não tem dimensão. Considere o seguinte vector, que contém a pauta de uma disciplina:

```
notas = as.integer(rnorm(20, 12, 4))
```

Verifique as suas dimensões, usando a função `dim()`. Transforme-o numa matriz `pauta`, com 4 linhas e 5 colunas. Verifique agora as dimensões da matriz criada.

2. Junte nomes às linhas e às colunas da matriz `pauta`, em dois passos distintos.
3. Aceda aos elementos da matriz de diferentes formas:
 - duas posições
 - dois nomes
 - uma posição e um nome
 - uma linha inteira pela posição
 - uma coluna inteira pelo nome
 - primeiras 3 linhas, todas as colunas excepto a última
4. Junte uma nova coluna à matriz. Verifique agora os nomes das colunas da matriz. Como adicionar um nome à coluna recentemente adicionada?
5. Junte uma nova linha à matriz, atribuindo-lhe logo um nome.
6. Calcule a média dos valores em cada linha e em cada coluna.
7. Experimente avaliar o resultado de algumas operações algébricas sobre a seguinte matriz:

```
ma = matrix(as.integer(runif(16, 1, 20)), 4, 4)
```

Exemplos:

- `ma + 1`
- `ma * 2`
- `ma * ma`
- `ma %% ma`
- determinante, com a função `det()`
- transposta, com a função `t()`
- inverter, com a função `solve()`

3 Arrays

Em **R**, as matrizes são uma extensão dos vectores para duas dimensões. Os arrays aumentam essa extensão para qualquer número de dimensões. Seguem as mesmas regras e possibilidades de indexação que os vectores e matrizes.

1. Crie um array com 50 elementos, distribuídos por 2 linhas, 3 colunas, e 5 camadas.
2. Considere a matriz `pauta` da secção anterior. Junte várias cópias dessa matriz num array, como se, por exemplo, estivesse a agregar as pautas de diferentes anos, num só objecto.
3. Experimente aceder a elementos do array do mesmo modo que tentou para a matriz.

4 Listas

As listas, ao contrário dos vectores, matrizes e arrays, permitem guardar elementos de diferentes tipos (e.g., `"character"`, `"numeric"` e `"logical"`) num único objecto. Assim, podem ser vistas como uma colecção ordenada de objectos.

Os objectos que constituem uma matriz são designados por componentes, são numerados, e podem ter um nome associado.

Considere, como exemplo, a lista `estudante`, que corresponde a uma ficha com os seguintes dados de um aluno:

nro 1999001

nome John Doe

notas 14, 3, 12, 10, 19

inscrito Sim

1. Crie a lista `estudante` com as informações do aluno.
2. Visualize todos os componentes da lista. Compare com o resultado da função `str()`, quando chamada com a mesma lista.
3. Aceda ao primeiro componente da lista, usando a mesma sintaxe que nos vectores. Verifique o tipo de objecto desse resultado, usando a função `typeof()`.
4. Corrija o nome do primeiro componente para `"número"`.
5. Recebeu mais um dado do aluno: tem 33 anos de idade. Actualize a sua ficha com essa informação.
6. Acrescente ainda a seguinte informação à ficha do aluno, em apenas um passo:

sexo masculino

país "Maria" e "Zé"

7. O aluno recebeu a nota de mais um exame: 10 val. Adicione essa nota às restantes notas.
8. Quantos componentes já tem a ficha do aluno?

5 Data Frames

A estrutura de dados `data.frame` é das mais usadas em **R**, em boa parte, devido à sua versatilidade no tipo de objectos suportado e na indexação: é um tipo de dados derivado das listas, suportando elementos de diferentes tipos num só objecto; e adiciona as funcionalidades de indexação dos vectores.

É mais usado para guardar tabelas de dados, sendo parecido com uma matriz com nomes nas linhas e nas colunas, mas suportando elementos de diferentes tipos. Este tipo de tabela pode ser visto como uma tabela de uma base de dados, ou de uma folha de cálculo, em que cada linha corresponde a um registo diferente.

1. Transforme a matriz `pauta`, criada na secção 2, num `data.frame`, e guarde-o na variável `pautadf`.
2. Repita o exercício 2.3, agora aplicado à `pautadf`.
3. Determine o número de alunos (linhas) e o número de componentes de avaliação (colunas). No exercício 2.1 referiu-se a função `dim()`. Procure por uma forma alternativa de encontrar o número de linhas e de colunas no ecrã de ajuda da função `dim()`.
4. Os alunos foram aleatoriamente distribuídos em dois grupos, conforme mostra o bloco de código em baixo. Explique-o da melhor maneira que conseguir.

```
Grupos = sample(paste("Grupo",1:2), nrow(pautadf), replace=T)  
pautadf = data.frame(pautadf, Grupos)
```

5. Depois da execução das instruções dadas na alínea anterior, uma nova coluna surge no data frame `pautadf`. Verifique o tipo de objecto com que os elementos dessa coluna ficaram. O que aconteceu?
6. Utilize o editor interactivo, disponibilizado pela função `edit()`, para alterar alguns valores (à sua escolha) do data frame.