

Definimos **lista** como um tipo de objeto constituído por uma sequência ordenada de objetos, as componentes, que podem ser de tipo diferente e de tamanho ou dimensão diferente, podendo ter nomes associados. A função generérica para a definição de uma lista é `list` e tem como argumentos obrigatórios os objetos que são as seus componentes. Exemplo:

```
lst<-list(x=1:50,y=matrix(-9:5,nc=3,nr=5),z=c(TRUE,FALSE,TRUE,TRUE))
```

em que a primeira componente é um vetor numérico, a segunda uma matriz e a terceira um vetor lógico. Podemos aceder às componentes da lista usando parênteses retos duplos `[[]]` ou o nome da componente após o nome da lista. Por exemplo `lst[[2]]` devolve, tal como `lst$y`, a matriz guardada em `y` (objeto definido dentro do ambiente `list`). A utilização de parênteses simples, tal como nos vetores, devolve um objeto da classe `list` e não da classe do objeto componente. Por exemplo:

```
> class(lst[2])
[1] "list"
> class(lst[[2]])
[1] "matrix"
> class(lst$y)
[1] "matrix"
```

Definimos **data frame** como uma tabela em que as colunas são objetos não obrigatoriamente da mesma classe (tal como as componentes nas listas) mas todos devem ter o mesmo comprimento e nomes associados. Num data frame os vetores do tipo **character** são fatores (objetos da classe **factor**), com tantos níveis quantos os valores distintos que ocorrem.

A função generérica para a definição de um data frame é `data.frame` e tem como argumentos obrigatórios os vetores que constituem as colunas da tabela. A função `str` aplicada a um objeto da classe **data frame** devolve informação de fácil leitura, como podemos ver no exemplo seguinte.

```
df<-data.frame(idade=c(21, 43, 25, 34, 15), gen=c("F","F","M","M","F"),
at.fis=c(TRUE,TRUE,FALSE,TRUE,TRUE), peso=c(70,66,86,74,54))
str(df)
'data.frame': 5 obs. of 4 variables:
 $ idade : num 21 43 25 34 15
 $ gen : Factor w/ 2 levels "F","M": 1 1 2 2 1
 $ at.fis: logi TRUE TRUE FALSE TRUE TRUE
 $ peso : num 70 66 86 74 54
```

Podem-se obter as colunas/entradas do data frame quer com a notação usada nas listas quer com a usada nas matrizes. Por serem objetos centrais na análise de dados, podemos trabalhar dentro do seu ambiente

através da função `attach`. A operação “inversa” realiza-se com a função `detach`. Observe o efeito destas operações no “caminho de pesquisa” `search()`.

1. Considere os dados `iris` (objeto da biblioteca de dados `package datasets` do R).
 - (a) Usando a função `help` (ou alternativamente o símbolo `?`), obtenha informações sobre estes dados. Visualize a estrutura do dataframe usando a função `str`.
 - (b) Em que espécie(s), observa-se o maior comprimento de pétala?
 - (c) Qual o comprimento médio da pétala na espécie `virginica`?
 - (d) Usando a função `tapply`, obtenha o comprimento médio da pétala em função da espécie.
 - (e) Usando a função `hist`, obtenha um histograma para os dados correspondentes ao comprimento da pétala na espécie `virginica`. Usando a opção `plot=F`, verifique que intervalos e frequências foram usados na construção do histograma.
 - (f) Considerando apenas as colunas `Petal.Length`, `Petal.Width` e `Species`, construa num novo dataframe em que a ordenação corresponde à ordem crescente dos dados em `Petal.Length`.
2. Usando a função `read.table` com opção `header=T`, importe a tabela de dados constante no ficheiro de texto `trafego.txt` guardando-a no objeto `traf` da classe `data.frame`. Os dados são relativos a 184 observações de duas variáveis, `limite`, `acidentes`, cujos valores dizem respeito à indicação, ou não, de limite de velocidade em autoestradas e respetivo número de acidentes durante determinado intervalo de tempo.
 - (a) Qual o número máximo de acidentes observado? Verifique se na(s) autoestrada(s) correspondente(s) havia, ou não, limite de velocidade e obtenha a(s) linha(s) correspondente(s) da dataframe.
 - (b) Obtenha o número médio de acidentes em função da existência de limite de velocidade.
 - (c) Obtenha dois subconjuntos distintos de dados, `acs`, relativo ao número de acidentes nas autoestradas sem indicação do limite de velocidade e `acc`, relativo ao número de acidentes nas autoestradas com indicação do limite de velocidade. Construa histogramas para a representação da distribuição empírica dos dois conjuntos de dados.
3. Considere o data frame `airquality` que contém dados sobre a qualidade do ar em Nova York entre maio e setembro de 1973.
 - (a) Qual a temperatura máxima durante o mês de julho?
 - (b) Obtenha a temperatura máxima em cada mês usando `tapply`.
 - (c) Use a função `split` para obter as temperaturas por mês. Obtenha novamente a temperatura máxima por mês usando a função `sapply`.
 - (d) Verifique quantos valores dos campos `Ozone` e `Solar.R` não estão disponíveis (missing values – `Not Available`; função de teste: `is.na`).
 - (e) Quais os índices das linhas que têm missing values na variável `Ozone`?
 - (f) Defina um novo data frame a partir de `airquality` em que não existam observações NA.
 - (g) Usando o data frame definido na alínea anterior e uma função apropriada, obtenha, para cada um dos meses de maio a setembro, o máximo, mínimo, média de radiação solar.

4. Use a função `read.csv` para importar a tabela de dados constante no ficheiro `C02Quebec.csv` e guarde a num objeto `C02Quebec`. Estes dados referem-se ao consumo de CO_2 (em $\mu mol/m^2/seg$) por plantas no Quebec quando expostas a diferentes concentrações de CO_2 (em mL/L) ou quando sujeitas, ou não, a arrefecimento.
- (a) Quantas plantas foram sujeitas a tratamento (`Treatment`) de arrefecimento (`chilled`)?
 - (b) Obtenha o consumo médio de CO_2 em função do nível de concentração para as plantas que não foram sujeitas a tratamento de arrefecimento. Obtenha uma representação gráfica por um linha quebrada (use `plot` com a opção `type="l"`). Obtenha a mesma informação para as plantas sujeitas a tratamento e adicione a representação gráfica ao gráfico já obtido. Adicione legendas.
5. Escreva uma função `dado` que tenha um só argumento, `n`, e que implemente o seguinte algoritmo:
- (1) Simule `n` lançamentos de um dado equilibrado. Guarde os resultados obtidos num objeto `x`.
 - (2) Calcule a média dos números em `x` e guarde-a no objeto `medx`.
 - (3) Obtenha uma tabela de frequências indicando o número de vezes que saiu cada face e guarda-a num objeto `tabx`.
 - (4) Calcule $\sum_{i=1}^6 (e_i - o_i)^2 / e_i$ onde e_i e o_i são as frequências esperadas e observadas das faces e guarde o resultado num objeto `estatx`
 - (5) Devolve uma lista com o conteúdo de `x`, `medx`, `tabx`, `estatx` com nomes `lanc`, `media`, `tabela`, `estat`.
6. Execute a função `dado` do exercício anterior para $n = 100$ e guarde o resultado num objeto `result`. Verifique o valor de `result$estat`. Se este não for inferior a 2, use o comando `while` para obter um conjunto de 100 lançamentos que verifique a condição `estat < 2`.
7. Usando a função `replicate` obtenha 400 médias correspondentes a 400 experiências de 10 lançamentos de um dado equilibrado. Represente os resultados obtidos através de um histograma.
8. Considere a experiência que consiste em contar o número de vezes que sai `coroa` antes que saia pela primeira vez `cara` no lançamento de uma moeda equilibrada. Simule esta experiência usando a função `while`. Repita a experiência 4000 vezes e elabore uma representação gráfica adequada dos resultados obtidos.