Curso de Especialização em

Análise de Dados Geográficos com

Sessão 2 – Funções pré-definidas

Marta Mesquita / Fernanda Valente
Instituto Superior de Agronomia
DCEB – Secção de Matemática
Dezembro 2015

Sessão 2 - Sumário

- O ambiente RStudio
- Funções pré-definidas no R
- Leitura e Escrita em ficheiros

Pasta de trabalho (*Working directory*)

- Na consola com os comandos

```
> getwd() # para obter o caminho da actual wd
> setwd(dir) # para especificar uma nova wd
```

em que `dir` é o caminho completo da nova pasta de trabalho (utilizando o caracter /, ou dois \\, como separador de pastas)

- No ambiente RStudio

menu *Session* → *Set Working Directory* → *Choose Directory ...*

Funções

- Os objectos do tipo `function` são objectos do R que implementam funções, e que podem ser usados em expressões, em instruções e na implementação de outras funções.
- Uma função é definida por um nome, uma lista de argumentos separados por vírgulas e um bloco de instruções (corpo da função)
- Cada argumento tem um nome e uma posição e pode ter um valor definido por omissão

```
matrix(data = NA, nrow = 1, ncol = 1, byrow = FALSE,dimnames =
       NULL)    # cria uma matriz
```

Algumas funções úteis

- `length(x)` devolve o número de elementos de `x`
- `sum(x)` devolve a soma dos elementos de `x`
- `prod(x)` devolve o produto dos elementos de `x`
- `cumsum(x)` devolve um vector cujos elementos são a soma acumulada dos elementos de `x`
- `cumprod(x)` devolve um vector cujos elementos são o produto acumulado dos elementos de `x`
- `max(x)/min(x)` devolve o máximo/mínimo dos elementos de `x`
- `which(x, arr.ind = FALSE, useNames = TRUE)` devolve o índice dos elementos TRUE do vector lógico `x`
- `print(x); cat(x,...)` escrita de objecto; objectos(s)

Algumas funções estatísticas

- `mean(x)` devolve a média dos elementos de `x`
- `median(x)` devolve a mediana dos elementos de `x`
- `var(x)` devolve a variância dos elementos de x; é igual a `sum((x-mean(x))^2)/(length(x)-1)`
- `sd(x)` devolve o desvio padrão dos elementos de `x`; é igual a `sqrt(var(x))`
- `summary(x)` devolve os extremos, os quartis e a média do vector `x`
- `sample(x, size, replace = FALSE, prob = NULL)` gera um vector de tamanho "size" com elementos de x

Funções

- Os argumentos são passados pela posição e/ou nome.

```
> sample(1:10,5)
> sample(1:10, replace=TRUE)
```

- Não é obrigatório passar argumentos com valor definido por omissão

- Trabalhar com `NA`

```
## mean(x, trim = 0, na.rm = FALSE, …)
> y<-c(12:18,NA)
> mean(y)
> mean(y,na.rm=TRUE)
> mean(y,TRUE)  # Error
```


Funções

- Os argumentos podem ser em número variável (…)

```
max(..., na.rm = FALSE)
> x<-1:10; y<-12:14
> max(x)
> max(x,y)
```

- Qualquer argumento que esteja após (…) tem de ser passado por nome

```
> y<-c(12:18,NA)
> y
> max(x,y)
> max(x,y,na.rm=TRUE)
```

Leitura de ficheiros

- Um quadro de valores, registado num ficheiro de texto (ASCII), pode ser atribuído a uma *data frame*, através da função

```
read.table(file,header=FALSE,sep="", dec =
  ".",row.names, col.names, as.is = FALSE,
  na.strings = "NA")
```

em que,

file é o nome do ficheiro de dados; se este não se encontrar na pasta de trabalho, é necessário indicar o endereço completo (utilizando / ou \\ como separador de pastas);

header é uma variável lógica que indica se o ficheiro tem ou não os nomes das variáveis na primeira linha. Por omissão o valor é FALSE;

sep é o caracter que separa os valores em cada linha; por omissão é "" que corresponde a um ou mais espaços em branco;

Leitura de ficheiros (cont.)

dec é o caracter utilizado como separador decimal. Por omissão é o ponto;

row.names é um vector com os nomes das linhas. Por omissão `1,2,3,...`;

col.names é um vector com os nomes das colunas. Por omissão `V1,V2,V3,...`;

as.is é uma variável lógica que controla a conversão das variáveis alfanuméricas em factores. Se o seu valor é TRUE a leitura mantém o tipo original dos dados;

na.strings é o valor usado para os dados desconhecidos no ficheiro e que será convertido em `NA`.

Leitura de ficheiros - exemplo

- O ficheiro de texto (com valores separados por vírgulas e com cabeçalho) `EstacMeteo.csv` contem dados referentes às estações meteorológicas automáticas do Instituto Português do Mar e da Atmosfera (IPMA).
  - Importe este ficheiro, que se encontra na página do curso https://fenix.isa.ulisboa.pt/qubEdu/cursos/ce.adg_r/lateral/material-pedagogico/sessao-2 para a sua área de trabalho.
  - Crie a *data frame* `estac.meteo` com o conteúdo deste ficheiro.

```
> estac.meteo <- read.table("EstacMeteo.csv",
  header=TRUE, as.is=TRUE,sep=",")

> estac.meteo <- read.table("EstacMeteo.csv",
  header=TRUE, as.is=TRUE,sep=",", row.names=1)
```


Leitura de ficheiros - exercícios

1) Num programa de controlo da poluição de um rio são efectuadas medições antes de lançar uma campanha antipoluição e um ano após. As medições são combinaçõoes de vários índices, quanto maior for o valor resultante maior é a poluição.

   Obtiveram-se os seguintes resultados:

Controle	A	B	C	D	E
Antes	68	88	101	82	96
Após	67	87	90	76	98

a) Introduza estas observações no ficheiro poluicao.csv (ou .txt).

b) Importe a informação contida no ficheiro poluicao.csv (.txt) para o R.

2) Importe para o R o ficheiro salary.dat disponível em `http://data.princeton.edu/wws509/datasets/salary.dat`

Leitura de dados

- A função `scan` é mais flexivel que a função `read.table`, permitindo ler da consola e criando um vector ou uma lista

```
scan(file = "", what = double(), sep = "",
dec = ".", skip = 0, nlines = 0, na.strings =
"NA",text)
```

em que,

file é o nome do ficheiro de dados; por omissão, lê directamente da consola;

what especifica o(s) tipo(s) dos dados (numéricos por omissão);

text conjunto de caracteres a ler, caso o argumento file não seja especificado;

Leitura de dados - exemplo

- Leitura de dados directamente da consola

```
> ex <- scan()
1: 3 5 6 9
5: 2 5 6
8:
Read 7 items

> ex<-scan(text="Ana Maria Rui Lia Tó",what="")
> ex1<-scan(what="") # scan(what="",sep="\n")
1: Ana Maria
3: Rui
4: Lia
5: Tó
6:
Read 5 items
```


Escrita em ficheiros

- Para escrever num ficheiro o conteúdo de uma *dataframe* ou de uma matriz usa-se a função

```
write.table(x,file="",quote=TRUE,sep=" ",
na="NA",dec=".",row.names=TRUE,col.names=TRUE)
```

em que,

x é o nome do objecto a guardar;

file é o nome do ficheiro, incluindo o caminho de pastas;

quote é uma variável lógica que indica se as variáveis alfanuméricas são ou não escritas entre aspas;

sep é o caracter que separa os valores em cada linha; por omissão é um espaço em branco;

Escrita em ficheiros (cont.)

na é a representação no ficheiro dos NA's de `x`;

dec é o caracter utilizado como separador decimal. Por omissão é o ponto;

row.names, **col.names** ou são variáveis lógicas que indicam se o ficheiro vai ou não conter os nomes das linhas/colunas de `x`, ou são vectores alfanuméricos com os nomes a atribuir às linhas/colunas no ficheiro.

Escrita em ficheiros - exemplos

- **Exemplo**: Guardar no ficheiro "trab.txt" a `dataframe` trees disponível em `data()`

```
> write.table(trees,"trab.txt")
> write.table(trees,"trab.txt",row.names=F,
 + quote=F,sep="\t")
```

- **Exemplo**: Guardar no ficheiro "trab.csv" a `dataframe` trees disponível em `data()`

```
> write.table(trees,"trab.csv",row.names=F,
 + quote=F,sep=",")
```