

Manual de Laboratori  
Iniciació a la Informàtica  
Grau en Estadística

Natàlia Pallarès  
Jaume Baixeries

29 de novembre de 2017

# Índex

<b>1</b>	<b>Consideracions Bàsiques</b>	<b>2</b>
1.1	Com podem saber en quin directori estem treballant? . . . . .	2
1.2	Com podem canviar el directori de treball? . . . . .	2
1.3	Com podem crear un directori? . . . . .	3
<b>2</b>	<b>Scripts</b>	<b>4</b>
2.1	Com creem un script nou? . . . . .	4
2.2	Com guardem les instruccions? . . . . .	5
2.3	Com obrim un script ja guardat? . . . . .	5
2.4	Com executem un script? . . . . .	5
2.5	Exemple d'Execució d'un Script . . . . .	5
<b>3</b>	<b>Entorn de Treball</b>	<b>7</b>
3.1	Com podem mirar quines variables tenim definides i què emmagatzemen? . . . . .	7
3.2	Com esborrem tota la informació que tenim emmagatzemada? . . . . .	7
3.3	Com podem guardar les comandes que hem fet servir? . . . . .	7
3.4	Com podem guardar la sessió de treball? . . . . .	8
<b>4</b>	<b>Lectura de Dades d'un Fitxer</b>	<b>9</b>
<b>5</b>	<b>Exemple d'una Sessió de Treball</b>	<b>11</b>

# Capítol 1

## Consideracions Bàsiques

Normalment, si treballeu en un entorn **Windows**, haureu d'anar al botó d'inici, i buscar el programa R a **Aplicacions**.

Si treballeu en un entorn **Linux**, llavors només cal que obriu una pantalla de terminal, i que piqueu R.

Cal que tinguem en compte que ens trobarem, per defecte, en un directori de treball. Aquest directori és el lloc on obrirem els fitxers R o els de dades. També cal saber que, en cas que ens interessi, el podrem canviar.

Us aconsellem que verifiqueu en quin directori de treball us trobeu abans de començar a treballar, i que, si cal, canvieu aquest directori a un de la vostra creació, el nom del qual no tingui ni espais ni accents ni símbols que no siguin alfanumèrics.

### 1.1 Com podem saber en quin directori estem treballant?

Escriure `getwd()` a la consola de R.

### 1.2 Com podem canviar el directori de treball?

Escriure `setwd("directori")`.

Per exemple: `setwd("E:/Curs2014_2015/Q1/Info")`

### 1.3 Com podem crear un directori?

Amb la comanda `dir.create()`. Per exemple, si volem crear un directori que es digui `programacio` a la unitat `F:` ho farem així:

```
dir.create("F:/programacio")
```

En cas que vulguéssim crear el directori `sessio1` a dins del directori `programacio`, primer caldria que el directori `programacio` existís, altrament, si no existís i miréssim de fer una cosa així:

```
dir.create("F:/programacio/sessio1")
```

ens donaria un error:

```
In dir.create("F:/programacio/sessio1") :  
cannot create dir 'F:/programacio/sessio1',  
reason 'El fitxer o directori no existeix'
```

Això vol dir que el directori `programacio` no existeix, i que cal crear-lo primer:

```
dir.create("F:/programacio")  
dir.create("F:/programacio/sessio1")
```

# Capítol 2

## Scripts

Un **script** és un fitxer que té l'extensió R (per exemple `exercici1.R`) i que conté un conjunt d'instruccions en R que, un cop s'invoca, s'executen de manera seqüencial. Per exemple, si tenim el fitxer `prova.R` amb el següent contingut:

```
x <- scan()
cat (x, "\n")
cat ("Ja hem acabat \n")
```

quan l'invocuem, l'entorn en R passarà el control a aquest script, i l'executarà. Per tant, l'script ens demanarà primer un valor per a la variable `x` (amb l'instrucció `x <- scan()`), després n'escriurà el valor pel terminal (instrucció `cat (x, "\n")`) i, finalment, escriurà pel terminal el missatge `Ja hem acabat` (instrucció `cat ("Ja hem acabat \n")`)

Us aconsellem que feu servir els scripts per a tots els vostres exercicis. Entre molts avantatges, us permetrà desar-los en un lllapis de memòria o enviar-vos-els per correu-e.

### 2.1 Com creem un script nou?

Fent Fitxer -> Nou script s'obre l'editor de R. Aquí podem escriure les instruccions.

## 2.2 Com guardem les instruccions?

Un cop tenim les instruccions en un script, fent `Ctrl + S` (o `Fitxer -> Desar`) les guardem escrivint el nom que volem pel fitxer i recordant posar sempre `.R` al final per especificar el tipus d'arxiu.

## 2.3 Com obrim un script ja guardat?

Fent `Fitxer -> Obrir script`. Si estem treballant en el directori on hi ha l'script, ens sortirà directament. Si no, haurem d'anar a buscar-lo al directori on estigui guardat.

## 2.4 Com executem un script?

Si les instruccions que volem executar estan guardades en un fitxer `main.R`, fent `source("main.R")` R executarà totes les instruccions que hi hagi en aquest fitxer.

## 2.5 Exemple d'Execució d'un Script

Ara expliquem com podem executar un script que crida un altre script. Pensem que tenim el fitxer `funcio.R` que té el següent contingut:

```
1 la_meva_funcio <- function (x) {  
2 # Aquesta és una funció de prova  
3   y <- x + 500  
4   return (y)  
5 }
```

Com podem veure, aquest fitxer només té una funció i prou. Si nosaltres l'executem:

```
source("funcio.R")
```

l'R executarà el fitxer, però com que aquest fitxer només té una declaració de la funció `la_meva_funcio`, no farà res. Bé, sí que farà alguna cosa: carregarà la funció `la_meva_funcio` a memòria i la tindrem disponible per

si la volem fer servir. Però normalment, no és això el que voldrem fer, sinó que voldrem que un altre script cridi aquesta funció.

Ara suposem que tenim el fitxer `crida.R` amb el següent codi:

```
1 source("funcio.R")
2 cat ("Entra el valor... ")
3 valor <- scan(n=1)
4 resultat <- la_meva_funcio (valor)
5 cat ("El resultat de la crida és:",resultat,"\n")
```

Ara anem a veure quin és el resultat de la crida a aquest script:

```
> source ("crida.R")
Entra el valor... : 22
Read 1 item
El resultat de la crida és: 522
```

Què ha fet l'R? Simplement ha executat el fitxer `crida.R`. Això vol dir que ha executat, línia per línia, el contingut d'aquest script. La primera línia era una `source("funcio.R")`, això ha fet que es carregués i executés el fitxer `funcio.R`. En aquest cas, això ha implicat només que es declarés la funció `la_meva_funcio` i que, per tant, estigués disponible per a ser cridada. Després s'ha executat `cat` i `scan`, i després la línia `resultat <- la_meva_funcio (valor)`. Com que prèviament hem fet la crida `source("funcio.R")`, podem fer aquesta crida a la funció, altrament, hi hauria un error.

# Capítol 3

## Entorn de Treball

Cada cop que obrim R se'ns crea una sessió de treball. Això vol dir que R va guardant en memòria les variables, funcions... que anem creant per poder accedir-hi durant la sessió.

### 3.1 Com podem mirar quines variables tenim definides i què emmagatzemen?

Fent `ls()` a la consola, obtenim els noms de les variables. Escrivint el nom de la variable, veiem què conté.

### 3.2 Com esborrem tota la informació que tenim emmagatzemada?

Escrivint `rm(list=ls())` a la consola s'esborra tota la informació de variables, funcions... que teníem emmagatzemada.

### 3.3 Com podem guardar les comandes que hem fet servir?

Fent `savehistory(file = "comandes.Rhistory")` se'ns crearà (en el directori on estem treballant) un fitxer `comandes.Rhistory` on hi ha emmagatzemades totes les instruccions que hem escrit durant la sessió. Després, en qual-



sevol moment, podem carregar-les fent `loadhistory(file = "comandes.Rhistory")` i podem veure-les fent `history()`.

### 3.4 Com podem guardar la sessió de treball?

Quan hem acabat la sessió de treball podem sortir de R fent `q()` (o `quit()`). Per defecte, R no guarda les sessions de treball. Si volem guardar la sessió de treball per tenir emmagatzemades les variables i funcions definides durant aquesta sessió en un altre moment, podem fer `q("yes")` (o `q("yes")`). Això ens crearà (en el directori on estem treballant) un fitxer `.Rhistory` (com l'explicat en l'apartat anterior) i un fitxer `.RData` que és el que conté la informació emmagatzemada i que podem voler utilitzar en un altre moment. Si només volem guardar la informació de les variables o funcions que hem anat creant (sense les comandes) podem executar la comanda `save.image("sessio1.RData")` i se'ns creara un fitxer `sessio1.RData` en el directori on estem treballant que conté la informació guardada durant la sessió i que podrem obrir en qualsevol moment.

# Capítol 4

## Lectura de Dades d'un Fitxer

Assumim que tenim un fitxer `entrada.txt` (que podem crear amb un editor de textos que ho desi com a fitxer de text) amb aquest contingut:

```
3
1
2
3
-1
```

El que farem és que `scan` llegeixi d'aquest fitxer d'entrada en comptes d'anar llegint del teclat. Primer de tot, cal *connectar* el fitxer d'entrada:

```
f <- file("entrada.txt","r")
```

Fixeu-vos que hem creat una variable `f` amb la funció `file` amb dos paràmetres: `entrada.txt` és el nom del fitxer on tenim les dades que hauríem entrat pel teclat, i `"r"` indica que volem llegir el fitxer (read). Ara, quan fem `scan` haurem d'afegir-hi que volem llegir del fitxer:

```
x <- scan(f,n=1,quiet=TRUE)
```

La diferència amb la crida anterior és que hi hem afegit un primer paràmetre a `scan`, que és la variable que fa de lligam amb el fitxer d'entrada.

Com a última instrucció del programa, caldrà utilitzar la comanda `close`, que, simplement *tanca* el fitxer que ja no utilitzarem més:

```
close(f)
```

Ara, el programa anterior el refem per incorporar-hi la lectura del fitxer en comptes del teclat:

```

1 f <- file("entrada.txt","r")
2
3 suma <- 0 # suma de tots els elements de la seqüè
      ncia.
4
5 x <- scan(f,n=1,quiet=TRUE) # x és el primer element
      de la seqüència
6
7 while (x != -1)
8 {
9     suma <- suma + x
10    x <- scan(f,n=1,quiet=TRUE)
11 }
12
13 cat ("La nova suma és ",suma,"\n")
14
15 close (f)

```

L'execució d'aquest programa no demana les dades de teclat, ja que les llegeix directament del fitxer:

```

1 > source("exemple.R")
2 La nova suma de tots els elements és 9

```

Si el que volem és llegir tot un vector, llavors el que podem fer és:

```

1 f <- file ("elmeuvector.txt","r")
2 N <- scan(f,n=1)
3 v <- scan(f,n=N)

```

on el fitxer *elmeuvector.txt* té aquest contingut:

```

6
1 2 3 4 5 6

```

Fixeu-vos que primer hi ha la mida del vector, que llegim amb la instrucció `N <- scan(f,n=1)`, i després llegim tot el contingut del vector amb la instrucció `v <- scan(f,n=N)`.

# Capítol 5

## Exemple d'una Sessió de Treball

Donem un exemple d'una sessió de treball en l'entorn R. Un cop dins de l'entorn de treball de l'R, verifiquem en quin directori ens trobem:

```
> getwd()  
[1] "/home/usuari/tmp"
```

Decidim que aquest no és el directori on volem treballar, sinó que volem treballar en un llapis de memòria que tenim a la unitat `F:`. Per tant, no només canviarem el directori de treball a aquesta unitat, sinó que, primer, hi crearem un directori per desar-hi els nostres fitxers:

```
> dir.create ("F:/programacio")  
> dir.create ("F:/programacio/sessio1")
```

Primer creem el directori `programacio` a `F:`. Si aquest directori ja existís, no caldria crear-lo, òbviament. Seguidament, creem el directori `sessio1` a dins del directori `programacio` a la unitat `F:`. Fixeu-vos que el nom del directori no té ni espais en blanc ni accents ni altres símbols que no siguin lletres i números.

Tot seguit, ja podem posar aquest directori com el directori de treball:

```
setwd("F:/programacio/sessio1")
```

I ara podem crear, editar i executar els nostres scripts. Un cop haguem acabat, si sortim de l'R tindrem que tots els scripts i també la sessió, quedarà desada a dins del directori `F:/programacio/sessio1`.