

# Curs d'introducció a R

## Classe 1: Introducció a R

Eudald Correig i Fraga

IISPV

11 de desembre de 2018

Primera sessió amb R

Objectes

Subseleccions

Llegir dades externes

Exploració de les dades

Manipulació de variables

Gràfics amb 


# Primera sessió amb R

## Consola

- ▶ Aquest símbol “>” és el prompt i és on haurem d'escriure les nostres ordres.
- ▶ Aquest símbol “#” serveix per introduir un comentari.
- ▶ Per separar expressions podem fer servir el “;”

## Consola

- ▶ Si ens equivoquem podem pitgem la tecla “ESC” i tornarà a aparèixer el símbol “>”
- ▶ Podem navegar per les instruccions que ja hem executat fent

servir les tecles 

- ▶ Per sortir del programa podem tancar o executar “>q()”

## Càlculs simples

```
4+5
```

```
## [1] 9
```

```
sqrt(49)
```

```
## [1] 7
```

```
log(2)
```

```
## [1] 0.6931472
```

```
rnorm(3)
```

```
## [1] -0.5141032 -0.2980893 -1.5789849
```

## Càlculs simples

```
w<-4+5 #deso el resultat de 4+5 a l'objecte w  
w
```

```
## [1] 9
```

```
sqrt(w)
```

```
## [1] 3
```

```
x<-rnorm(3)  
print(x)
```

```
## [1] -0.2510199 -0.2471895  0.1899792
```

## Espai de treball

- ▶ És l'espai on es desen de forma temporal o no tots els objectes que anem creant o carregant.
- ▶ Quan tanquem el programa el podem desar.

```
getwd() #on és el meu espai de treball  
ls() #quins objectes hi ha?
```

```
setwd("PATH") # hem de substituir el "path" pel de l'espai  
# treball on volem estar  
save.image("Prova.Rdata")
```





## Com interpretar l'ajuda

Imaginem que volem saber més sobre les funcions de distribució normals en R, escrivim:

```
?sample
```

► Ens surt una informació com aquesta:

sample (base)

R Documentation

### Random Samples and Permutations

#### Description

`sample` takes a sample of the specified size from the elements of `x` using either with or without replacement.

#### Usage

```
sample(x, size, replace = FALSE, prob = NULL)
```

```
sample.int(n, size = n, replace = FALSE, prob = NULL,  
           useHash = (!replace && is.null(prob) && size <= n/2 && n > 1e7))
```

#### Arguments

`x` either a vector of one or more elements from which to choose, or a positive integer. See 'Details.'

`n` a positive number, the number of items to choose from. See 'Details.'

## Com interpretar l'ajuda

- ▶ Hi ha informació sobre diverses funcions, amb tots els paràmetres explicats.
- ▶ Com veiem, les funcions a R sempre van amb el nom i parèntesis després on, si cal, hi poso els arguemnts (no s'ha de confondre amb les subseleccions, que van amb claudàtors).
- ▶ Important: els paràmetres que no tenen signe "=", per exemple `x`, `q`, `p` o `n`, són *obligatoris*
- ▶ Els que sí que tenen un signe "=", per exemple, `mean` o `sd` són opcionals i tenen valor per defecte el valor que trobem després de l'=. Per exemple el *valor per defecte* de la mitja és 0 i el de la desviació estàndard és 1.

## Paquets i les llibreries

- ▶ R consta d'un sistema “base”, que inclou molts mètodes estadístics, i un sistema per paquets que permeten anar molt més enllà.
- ▶ La comunitat R és qui elabora i manté aquests paquets (alguns signats per estadístics de gran renom).

## Paquets i les llibreries

Triem un repositori:

```
install.packages("foreign") #llegir Stata, SPSS...
```

Un cop instal·lat l'haurem de carregar a la biblioteca:

```
library(foreign) #require(foreign)
```

Si volem saber-ne més... :

```
library(help="foreign")
```



# Objectes

# Variables

- ▶ Les variables són els objectes més simple i poden ser:
  1. numèric: números reals (3.1415)
  2. complex: números complexes ( $2 + 5i$ )
  3. character: cadenes alfanumèriques de text ("patata")
  4. logical: variables lògiques (TRUE)



## Objectes

- ▶ Recordem que R és un llenguatge de programació orientat a objectes.
- ▶ Els objectes estan formats per elements.

```
w <- 3+4 #Creem l'objecte w  
print(w)
```

```
## [1] 7
```

```
# per cert: <- és equivalent a = !
```

- ▶ 7 és l'únic element de l'objecte w
- ▶ Els objectes més habituals són: variables, vectors, matrius, llistes i data frames.

## Tipus d'objectes

► Bàsicament podem identificar 4 tipus d'objectes:

1. Vectors
2. Matrius
3. Dataframes
4. Llistes

En aquest curs treballarem vectors i dataframes.

## Tipus d'objectes: vectors

- ▶ És una col·lecció ordenada d'elements del **mateix tipus** (numèric, factor...)
- ▶ Es creen amb la funció `c()`, segurament la funció més important d'R!

```
x<-c(2,4,6) #vector de 3 elements numèrics parells  
x
```

```
## [1] 2 4 6
```

```
y<-c("A","B","C") #vector de 3 elements cadena  
y
```

```
## [1] "A" "B" "C"
```

```
z<-c(TRUE,FALSE,TRUE) #vector de 3 elements lògics  
z
```

```
## [1] TRUE FALSE TRUE
```

## Algunes operacions: vectors

```
x<-1:10 #genera un vector d'1 a 10  
x
```

```
## [1] 1 2 3 4 5 6 7 8 9 10
```

```
x[4] #la posició 4 és
```

```
## [1] 4
```

```
x[x>7] #elements més grans de 7
```

```
## [1] 8 9 10
```

```
x+2 #sumar un escalar
```

```
## [1] 3 4 5 6 7 8 9 10 11 12
```

## Tipus d'objectes: dataframes

- És una col·lecció ordenada d'elements de **qualsevol tipus** amb dues dimensions.

```
id<-1:10  
  
sexe<-rep(c("M", "D"),5)  
  
normal<-rnorm(10)  
  
curacio<- rep(c(FALSE,TRUE),5)  
  
dades<-data.frame(id,sexe,normal,curacio)  
  
head(dades) #Mostrar primeres files/registres
```

```
##   id sexe    normal curacio  
## 1  1    M  1.5137630  FALSE  
## 2  2    D  1.2916383   TRUE  
## 3  3    M -0.6700642  FALSE  
## 4  4    D -1.9334036   TRUE  
## 5  5    M  0.9110426  FALSE  
## 6  6    D -0.9755996   TRUE
```

## Algunes operacions: dataframes

```
str(dades) #mostra estructura de les dades
```

```
## 'data.frame':    10 obs. of  4 variables:
## $ id      : int  1 2 3 4 5 6 7 8 9 10
## $ sexe    : Factor w/ 2 levels "D","M": 2 1 2 1 2 1 2 1 2 1
## $ normal  : num  1.514 1.292 -0.67 -1.933 0.911 ...
## $ curacio: logi  FALSE TRUE FALSE TRUE FALSE TRUE ...
```

```
dades$id #mostra valors variable id
```

```
## [1] 1 2 3 4 5 6 7 8 9 10
```

```
dades.s<- dades[order(dades$id),]#ordena per id
#podem afegir una nova variable
edat<-c(67,78,89,86,56,90,68,82,92,67)
dades.new<-data.frame(dades.s,edat)
```



## Subseleccions



## Subseleccions I

```
# Seleccionem una cel·la en concret, p. e.,  
# fila 2 columna 1:  
dades.new[2,1]
```

```
## [1] 2
```

```
# Seleccionem una fila, per exemple la tercera:  
dades.new[3,] # Noteu la coma i l'espai buit!
```

```
##   id sexe      normal curacio edat  
## 3  3    M -0.6700642   FALSE   89
```

## Subseleccions II

```
# Seleccionem una columna, per exemple la cinquena:  
dades.new[,5]
```

```
## [1] 67 78 89 86 56 90 68 82 92 67
```

```
#També ho podem fer amb el nom fent servir el símbol  
# del dòlar "$"  
dades.new$edat
```

```
## [1] 67 78 89 86 56 90 68 82 92 67
```

## Subseleccions III

- ▶ Podem seleccionar més d'una fila o columna
- ▶ El resultat és una altra dataframe

```
# Per exemple, seleccionem les tres primeres files  
# i totes les columnes:  
dades.new[1:3,]
```

```
##   id sexe    normal curacio edat  
## 1  1   M  1.5137630  FALSE   67  
## 2  2   D  1.2916383   TRUE   78  
## 3  3   M -0.6700642  FALSE   89
```

## Subseleccions IV

```
# Seleccionem les tres últimes columnes i totes les  
# files:  
dades.new[,3:5]
```

##	normal	curacio	edat
## 1	1.51376299	FALSE	67
## 2	1.29163830	TRUE	78
## 3	-0.67006423	FALSE	89
## 4	-1.93340362	TRUE	86
## 5	0.91104257	FALSE	56
## 6	-0.97559956	TRUE	90
## 7	-0.06710561	FALSE	68
## 8	-0.31411474	TRUE	82
## 9	0.01677476	FALSE	92
## 10	0.82617624	TRUE	67

## Subseleccions V

```
# Seleccionem totes les files menys la 4 i  
# totes les columnes menys la primera:  
dades.new[-4,-1]
```

##	sexe	normal	curacio	edat
## 1	M	1.51376299	FALSE	67
## 2	D	1.29163830	TRUE	78
## 3	M	-0.67006423	FALSE	89
## 5	M	0.91104257	FALSE	56
## 6	D	-0.97559956	TRUE	90
## 7	M	-0.06710561	FALSE	68
## 8	D	-0.31411474	TRUE	82
## 9	M	0.01677476	FALSE	92
## 10	D	0.82617624	TRUE	67

## Subseleccions VI

*# Podem crear una nova dataframe amb la subselecció:*

```
dades.sub = dades.new[c(4,5,7:9),-c(1,3)]  
dades.sub
```

##	sexe	curacio	edat
## 4	D	TRUE	86
## 5	M	FALSE	56
## 7	M	FALSE	68
## 8	D	TRUE	82
## 9	M	FALSE	92

## Subseleccions VII

*# Podem posar una condició dins de la subselecció*

```
dones = dades.new[dades.new$sexe == "D",]  
dones
```

##	id	sexe	normal	curacio	edat
## 2	2	D	1.2916383	TRUE	78
## 4	4	D	-1.9334036	TRUE	86
## 6	6	D	-0.9755996	TRUE	90
## 8	8	D	-0.3141147	TRUE	82
## 10	10	D	0.8261762	TRUE	67

## Subseleccions VIII

*# Un altre exemple:*

```
grans = dades.new[dades.new$edat > 85,]  
grans
```

##	id	sexe	normal	curacio	edat
## 3	3	M	-0.67006423	FALSE	89
## 4	4	D	-1.93340362	TRUE	86
## 6	6	D	-0.97559956	TRUE	90
## 9	9	M	0.01677476	FALSE	92





## Llegir dades externes

# Introducció

- ▶ R sap llegir dades de gairebé qualsevol format.
- ▶ El sistema és habitualment senzill i només cal tenir en compte quina llibreria cal en cada cas.
- ▶ En general, l'objecte generat és un dataframe

Format	Llibreria	Sintaxis
Text (.csv)	Base/Readr	<code>read.csv(ruta, header=TRUE, row.names='id')</code>
Text (.txt/ .csv)	Base/Readr/data.table	<code>read.table(ruta, header=TRUE, sep=',', row.names='id')</code>
MSExcel (.xls / .xlsx)	xlsx	<code>read.xlsx( ruta, sheetName = 'Fulla1')</code>
SPSS (.sav)	foreign	<code>read.spss(ruta)</code>
Stata (.dta)	foreign	<code>read.dta( ruta)</code>
SAS (.xpt)	foreign	<code>Read.xport(ruta)</code>

## Importar fitxers de text (.txt, .csv)

```
read.table("ruta",  
  header=TRUE,    #primera línia noms variables  
  sep="," ,       #separador entre variables  
  stringsAsFactors = FALSE,  #no factors per defecte  
  na.strings=c("NA","**"), #codis per a missings  
  dec="." #simbol decimal  
)
```

## Importar fitxers MS Excel

```
library(openxlsx)

read.xlsx("ruta",
  sheetIndex,  #nº de fulla (evitar posar el nom)
  sheetName="Full1", #nom de la fulla
  as.data.frame=TRUE, #com un data frame
  header=TRUE,  #noms variables
  rowIndex=20,  #número de files per llegir
)
```

## Importar fitxers SPSS

```
library(foreign)

read.spss("ruta",
  use.value.labels=TRUE,  # factors
  to.data.frame=TRUE,    #data frame
)
```

## Exemple

```
dades = read.csv("input/dades.csv")
```





## Exploració de les dades

## Estructura

Fem una ullada a les dades

```
str(dades) #mostra estructura d'un objecte
```

```
## 'data.frame':    10 obs. of  5 variables:
## $ id      : int   1 2 3 4 5 6 7 8 9 10
## $ sexe    : Factor w/ 2 levels "D","M": 2 1 2 1 2 1 2 1
## $ normal  : num   0.501 -0.309 -0.657 -0.779 -1.681 ...
## $ curacio: logi   FALSE TRUE FALSE TRUE FALSE TRUE ...
## $ edat    : int   67 78 89 86 56 90 68 82 92 67
```

## Primeres files

```
head(dades) #mostra les primeres files
```

##	id	sexe	normal	curacio	edat
## 1	1	M	0.5010061	FALSE	67
## 2	2	D	-0.3094905	TRUE	78
## 3	3	M	-0.6568773	FALSE	89
## 4	4	D	-0.7792013	TRUE	86
## 5	5	M	-1.6813112	FALSE	56
## 6	6	D	0.6612151	TRUE	90

## Altres funcions

```
tail(dades) #mostra les darreres files
```

```
View(dades) #mostra tot el conjunt de dades
```



## Manipulació de variables

## Creació noves variables

Entrem les alçades i els pesos:

```
dades$alcada = rnorm(nrow(dades), mean = 170, sd = 10)
dades$pes = rnorm(nrow(dades), mean = 75, sd = 10)
```

Índex de massa corporal  $imc = \frac{pes}{alcada^2}$

```
alcada.m<-dades$alcada/100
dades$imc<-dades$pes/ alcada.m^2
```

## Visualitzem

```
head(dades)
```

##	id	sexe	normal	curacio	edat	alcada	pes	imc
## 1	1	M	0.5010061	FALSE	67	163.7355	90.11781	33.61438
## 2	2	D	-0.3094905	TRUE	78	171.8364	78.89843	26.72009
## 3	3	M	-0.6568773	FALSE	89	161.6437	68.78759	26.32646
## 4	4	D	-0.7792013	TRUE	86	185.9528	52.85300	15.28496
## 5	5	M	-1.6813112	FALSE	56	173.2951	86.24931	28.71992
## 6	6	D	0.6612151	TRUE	90	161.7953	74.55066	28.47866



## Recodificació de variables

re-codificar imc:  $<18; 18- <25; 25- <30; >30$

```
dades$imccat <- rep(0, nrow(dades))
dades$imccat[dades$imc<18]<-0
dades$imccat[dades$imc>=18 & dades$imc<25]<-1
dades$imccat[dades$imc>=25 & dades$imc<30]<-2
dades$imccat[dades$imc>=30]<-3

dades$imccat<-factor(dades$imccat,levels=c(0,1,2,3),
                     labels=c("baix pes","normal",
                              "sobrepes","obesitat"))
```

*# Alternativa*

```
dades$imccat <- cut(dades$imc, breaks=c(-Inf,18,25,30,Inf),
                   labels = c("baix pes","normal",
                              "sobrepes","obesitat"))
```

## Modificació de nivells d'un factor

```
library(plyr) #llibreria tractament de dades  
  
dades$imccat <-revalue(dades$imccat, c("normal"="estàndar"))
```

## Funcions de transformació de variables

- ▶ `as.numeric()`
- ▶ `as.character()`
- ▶ `as.factor()`

## Transformació de variables:exemples

### ► as.numeric

```
a<-c("1", "3", "5")  
as.numeric(a)
```

```
## [1] 1 3 5
```

Compte quan passem de caràcter a numèric!!!

```
b<-c("1", "est", "5")  
as.numeric(b)
```

```
## Warning: NAs introduced by coercion
```

```
## [1] 1 NA 5
```

## Transformació de variables:exemples

Compte quan passem de factor a numèric!!!

```
b<-factor(c("1", "10", "5"))  
as.numeric(b)
```

```
## [1] 1 2 3
```

Passem sempre per as.character:

```
as.numeric(as.character(b))
```

```
## [1] 1 10 5
```

## Ordenar dataframes

- ▶ Ordenar per una variable

```
dades<-dades[order(dades$imc),]
```

- ▶ Ordenar per més d'una variable

```
dades<-dades[order(dades$sexe,dades$imc),]
```

- ▶ Ordre descendent

```
dades<-dades[order(dades$sexe,-dades$imc),]
```

## Keep/ drop

- ▶ Triem les variables que volem

```
variables<-c("id","sexe","imc")  
dades<-dades[variables]
```

- ▶ Descartem variables (símbol d'exclamació "!" equival a "NO")

```
dades<-dades[!variables]
```





Gràfics amb 

# Introducció

- ▶ En R hi ha moltes llibreries per fer gràfics
- ▶ R base té capacitat per fer gràfics simples
- ▶ Per gràfics més avançats la llibreria més utilitzada és “ggplot2”
- ▶ N'hi ha d'altres, com ara “plotly”.
- ▶ En aquest curs utilitzarem el paquet base i ggplot2 a l'última classe.

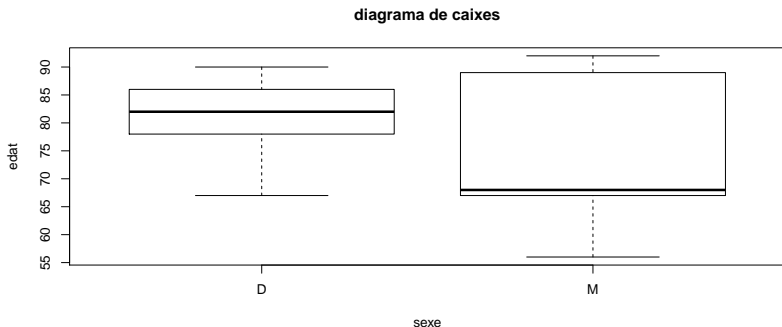
## Funció “plot”

```
plot(x = var.x,  
     y = var.y,  
     type = "tipus",  
     col = "color",  
     pch = "tipus de punt",  
     cex = "mida del punt",  
     lwd = "amplada de la línia",  
     main = "títol",  
     sub = "subtítol",  
     xlab = "nom de l'eix x",  
     ylab = "nom de l'eix y",  
     ...)
```

## Funció “plot”: diagrames de caixes

- Si li passem una variable categòrica sap que ha de fer boxplots:

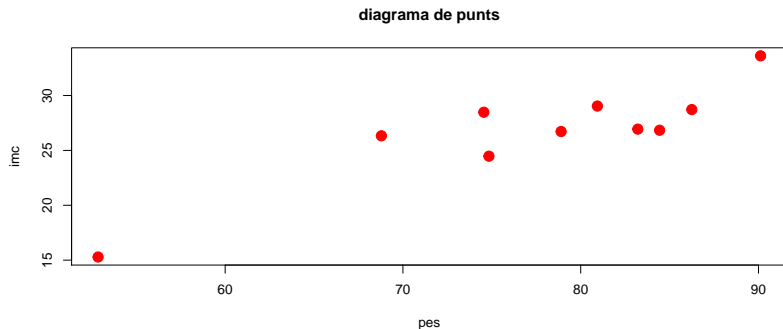
```
plot(dades$sexe, dades$edat, main = "diagrama de caixes",  
     xlab = "sexe", ylab = "edat")
```



## Funció “plot”: diagrames de punts

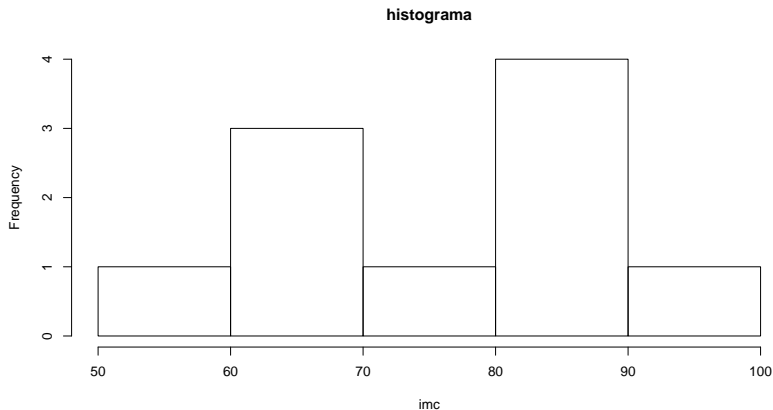
- Si les dues variables són contínues fa un gràfic de punts:

```
plot(dades$pes, dades$imc, pch = 20, col = "red", cex = 2.5,  
     main = "diagrama de punts", xlab = "pes", ylab = "imc")
```



# Histograma

```
hist( dades$edat, main = "histograma", xlab = "imc")
```



Fi

Final de la classe 1.