



UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH

Departament d'Estadística
i Investigació Operativa

ACADEMIC YEAR 2024-25

STATISTICAL
INFERENCE AND
MODELLING –
MASTER OF
DATA SCIENCE

Introduction to R: EDA
Introduction to R software
Lecturer: Lídia Montero
September 2024 – Version 1.1

MASTER OF DATA SCIENCE

1. LAB SESSIONS

- 2 hours every 1 week, in a PC's classroom.
- Practical assignments posted through ATENEA - TASKS. Formative assessment will be given by the lecturer before the next laboratory session when deliverable is indicated.
- Guidelines for laboratory session posted in ATENEA Course webpage
- Datasets posted on ATENEA Course webpage.

FIRST SESSION: *Introduction to R and R Studio statistical software*

R Core Team (2023). *_R: A Language and Environment for Statistical Computing_*. R Foundation for Statistical Computing, Vienna, Austria. <<https://www.R-project.org/>>.

2. UNITS 1-2: EXPLORATORY DATA ANALYSIS (EDA)-UNIVARIATE

⇒ Davis data: `davis.RData (data.frame)` - Use comands in `davis.R` for basics

```
> library(car)
Loading required package: MASS
Loading required package: nnet

> data(Davis)
> ls()
[1] "Davis"

> attributes(Davis)
$names
[1] "sex"      "weight"  "height"  "repwt"   "repht"

$class
[1] "data.frame"

$row.names
[1] "1"  "2"  "3"  "4"  "5"  "6"  "7"  "8"  "9"  "10" "11" "12"
...
[193] "193" "194" "195" "196" "197" "198" "199" "200"

>
```

UNITS 1 AND 2: EXPLORATORY DATA ANALYSIS (EDA)-UNIVARIATE

2.1 Univariate descriptive analysis - Numeric data

- Missing and Outliers might occur
- Numerical values
 - Measures of Central Tendency: *Mean, Median, Mode*
 - Measures of Dispersion: *Variance, Standard Deviation, Quartiles, IQR, Maximum, Minimum.*
- Graphical Representations
 - Histogram, Cumulative Histogram. Absolute or relative.
 - *BoxPlot.*
 - *Dotplot*

UNITS 1 AND 2: EXPLORATORY DATA ANALYSIS (EDA)-UNIVARIATE

2.1.1 Continuous Univariate Descriptive Analysis: Numeric statistics

> summary(dataframe) # R command

- Mean $\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i$
- Median: Value of the variable such that
50% Observations are < Median (Q2) & 50% Observations are > Median (Q2)
- Quartile Q1 of the 25% and quartile Q3 of the 75%: Values of the variable that
25% Observations are < Q1 & 75% Observations are > Q1
75% Observations are < Q3 & 25% Observations are > Q3
- Variance $s_x^2 = \frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2$
- Standard Deviation s_x (square root of variance)

UNITS 1 AND 2: EXPLORATORY DATA ANALYSIS (EDA)-UNIVARIATE

⇒ Davis data: `davis.RData (data.frame)` - Use comandns in `davis.R` for basics

```
> summary(Davis)
```

sex	weight	height	repwt	repht
F:112	Min. : 39.0	Min. : 57.0	Min. : 41.00	Min. :148.0
M: 88	1st Qu.: 55.0	1st Qu.:164.0	1st Qu.: 55.00	1st Qu.:160.5
	Median : 63.0	Median :169.5	Median : 63.00	Median :168.0
	Mean : 65.8	Mean :170.0	Mean : 65.62	Mean :168.5
	3rd Qu.: 74.0	3rd Qu.:177.2	3rd Qu.: 73.50	3rd Qu.:175.0
	Max. :166.0	Max. :197.0	Max. :124.00	Max. :200.0
			NA's :17	NA's :17

```
> var(Davis[,3:4])
```

	weight	height
weight	227.85930	34.37588
height	34.37588	144.19055

- Missing data: Do not miss them! Track them.

NA: Not available - Missing data

NaN: Not available for numerical reasons (divided by 0)

UNITS 1 AND 2: EXPLORATORY DATA ANALYSIS (EDA)-UNIVARIATE

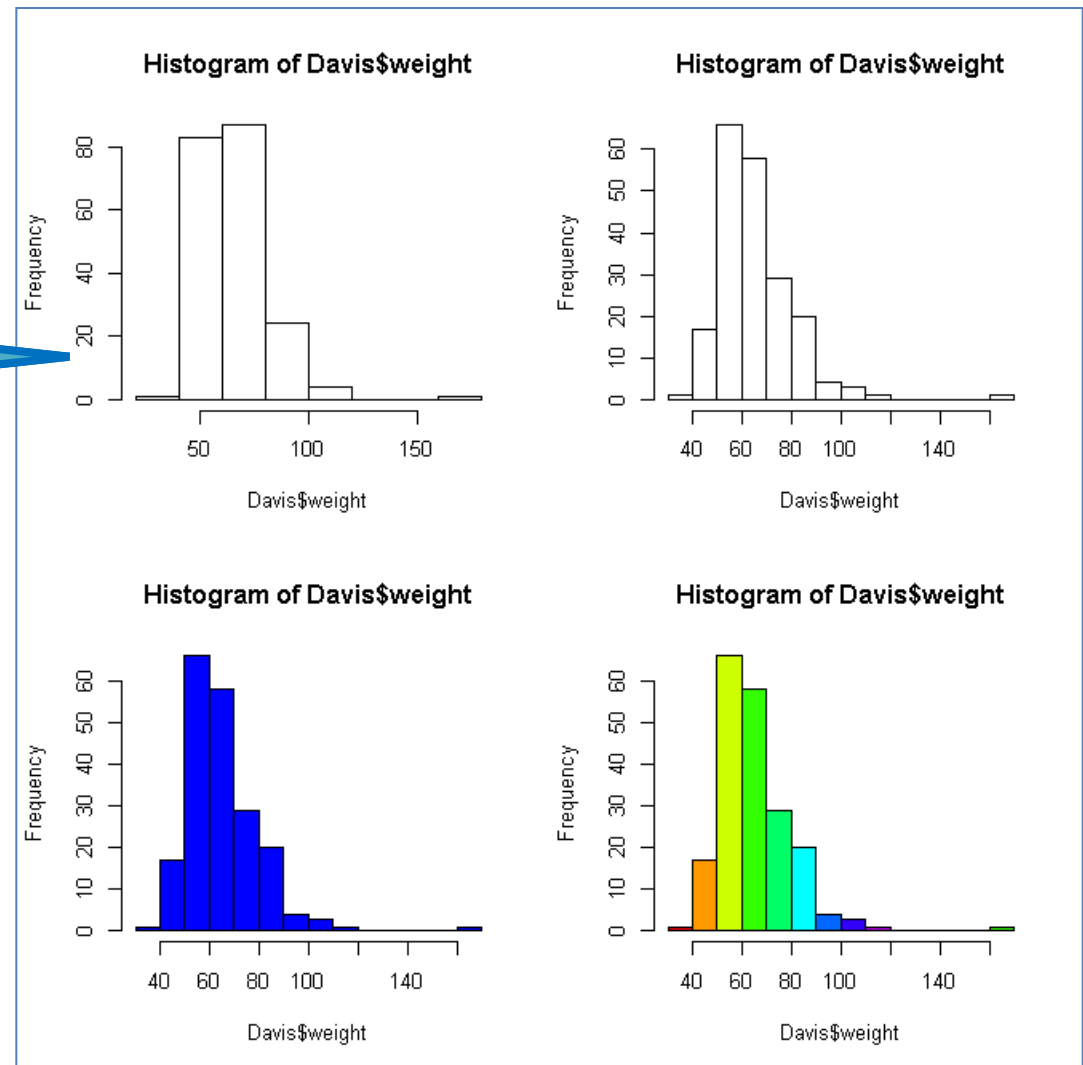
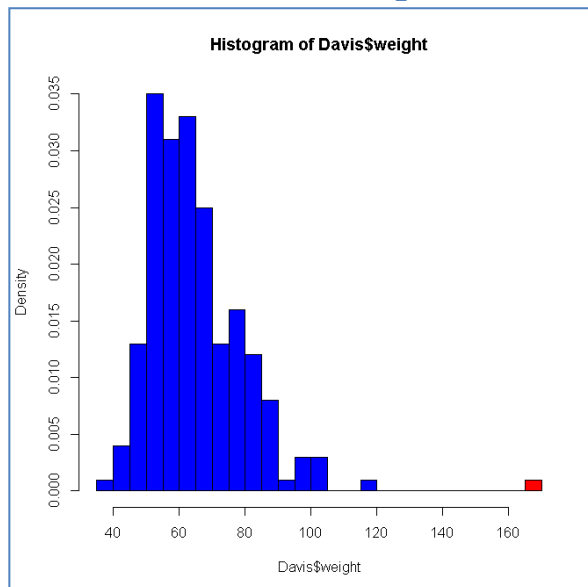
2.1.2 Continuous Univariate Analysis Description: Histogram

Histogram (non-acumulative):

```

par(mfrow=c(2,2))
hist(Davis$weight)
hist(Davis$weight,10)
hist(Davis$weight,10,col="blue")
hist(Davis$weight,10,col=rainbow(10))

hist(Davis$weight,freq=F) # Proportions
hist(Davis$weight,freq=F,
     col=...,breaks=seq(35,170,5))
    
```



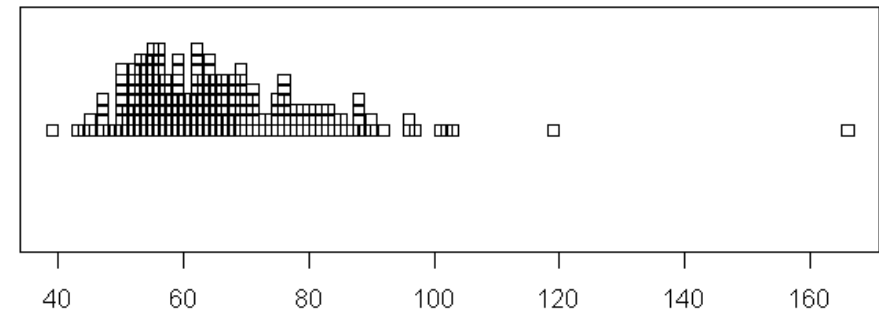
UNITS 1 AND 2: EXPLORATORY DATA ANALYSIS (EDA)-UNIVARIATE

2.1.3 Continuous Univariate Analysis Description: Dotplot

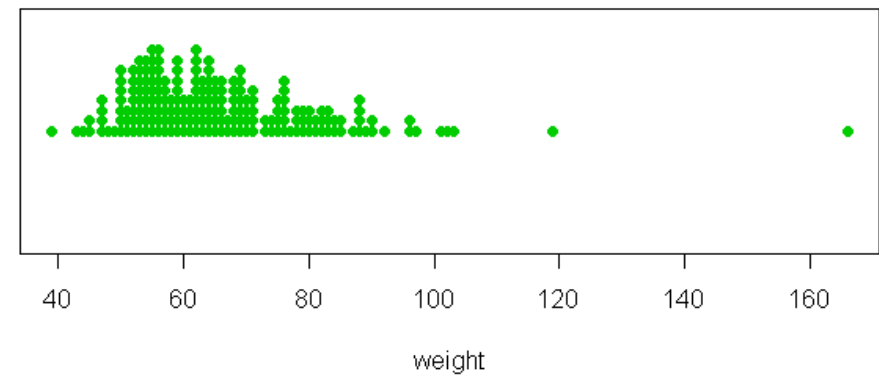
Dotplot:

```
# Dotplot
par(mfrow=c(2,1))
stripchart(Davis$weight,method="stack")

stripchart(Davis$weight,method="stack",
           ,xlab="weight",pch=19,
           col=3,
           main="Dotplot Weight in Davis dataset")
```



Dotplot Weight in Davis dataset

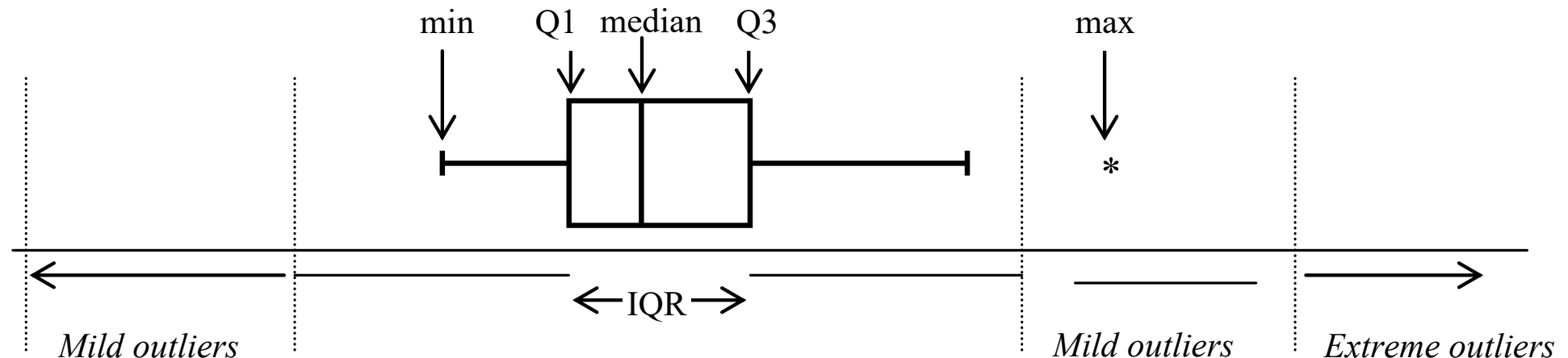


UNITS 1 AND 2: EXPLORATORY DATA ANALYSIS (EDA)-UNIVARIATE

2.1.4 Continuous Univariate Analysis Description: Boxplot

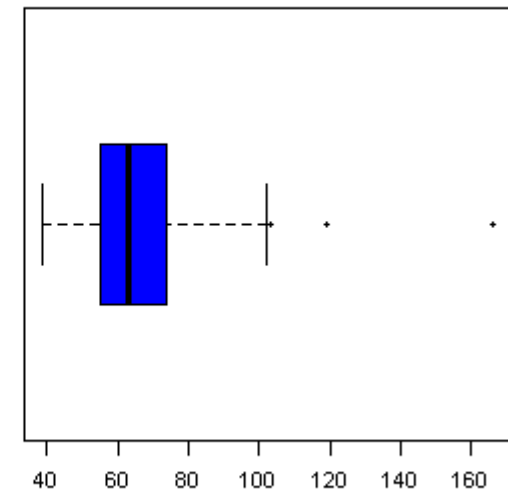
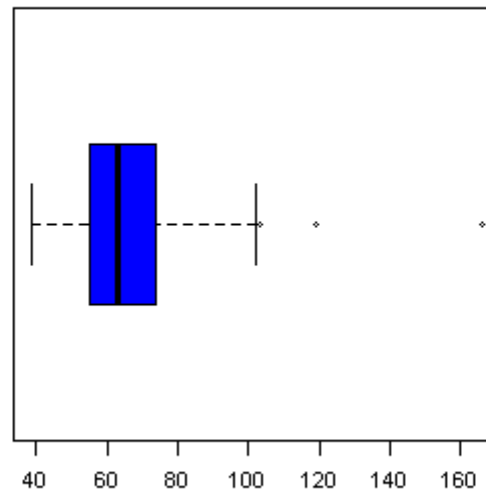
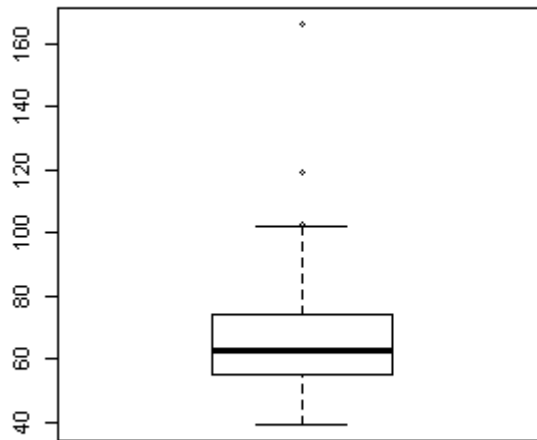
Boxplot: Basic implementation in `boxplot()` method, recommended `Boxplot()` method in `car` library

"Five issues Summary" (Min, Q1, Me, Q3, Max) for Univariate EDA, useful to detect the presence of outliers.



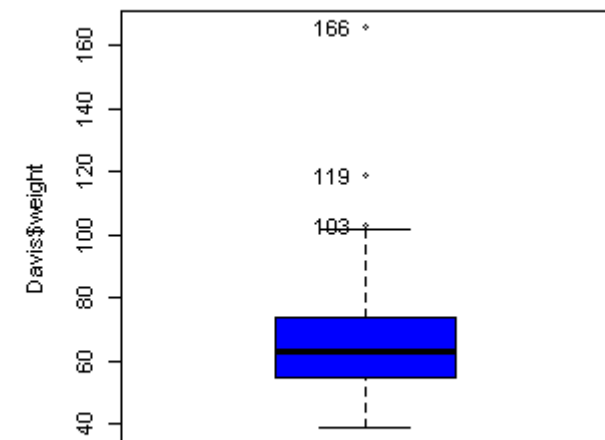
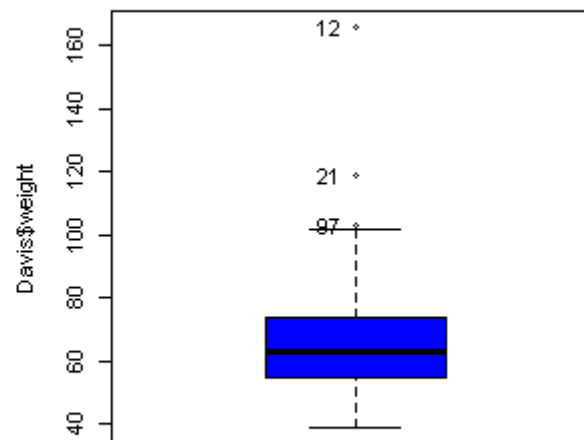
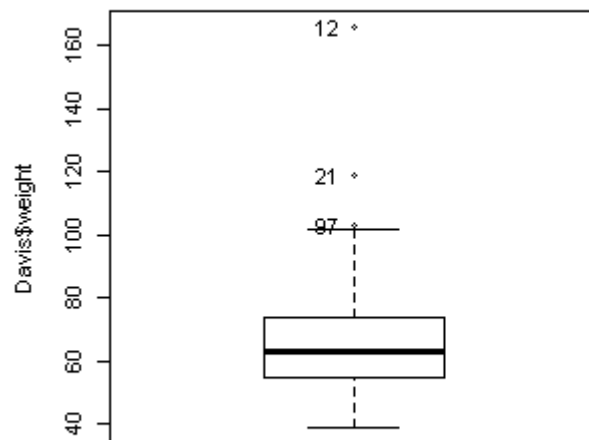
```

# Boxplot
par(mfrow=c(2,3))
boxplot(Davis$weight)
boxplot(Davis$weight,col="blue",horizontal = TRUE)
boxplot(Davis$weight,col="blue",horizontal = TRUE, pch=19,labels=Davis$weight)
Boxplot(Davis$weight)
Boxplot(Davis$weight,col="blue",main= "Weight in Davis dataset - row name Id")
Boxplot(Davis$weight,col="blue",main=" Boxplot Weight - Weight Label for
Outliers",labels=Davis$weight)
    
```



Weight in Davis dataset - row name Id

Boxplot Weight - Weight Label for Outliers



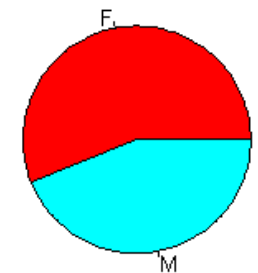
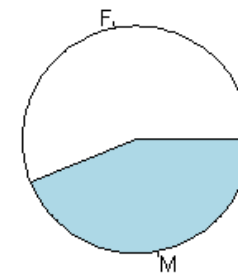
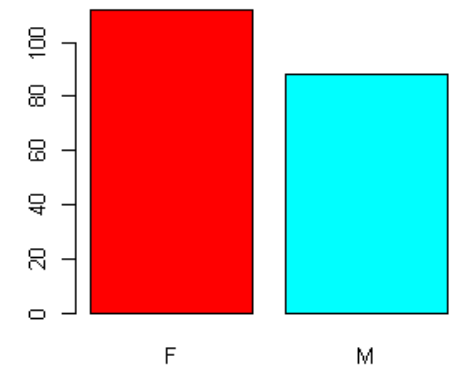
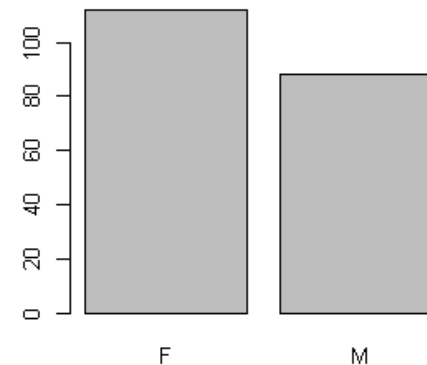
UNITS 1 AND 2: EXPLORATORY DATA ANALYSIS (EDA)-UNIVARIATE

2.2 Univariate descriptive analysis - Categorical data

Description of categorical variables: only 'missings' might occur. Graphical representations:

- **barplot** (a) absolute or relative (proportions)
b) density or accumulated.
- Suitable for graphical description of discrete-qualitative data (factor) with a few levels or categories.

- **Pie Chart.**



```

table(Davis$sex)
margin.table(table(Davis$sex))
prop.table(table(Davis$sex))

par(mfrow=c(2,2))
barplot(table(Davis$sex))
barplot(table(Davis$sex), col=rainbow(2))
pie(table(Davis$sex))
pie(table(Davis$sex), col=rainbow(2))
    
```

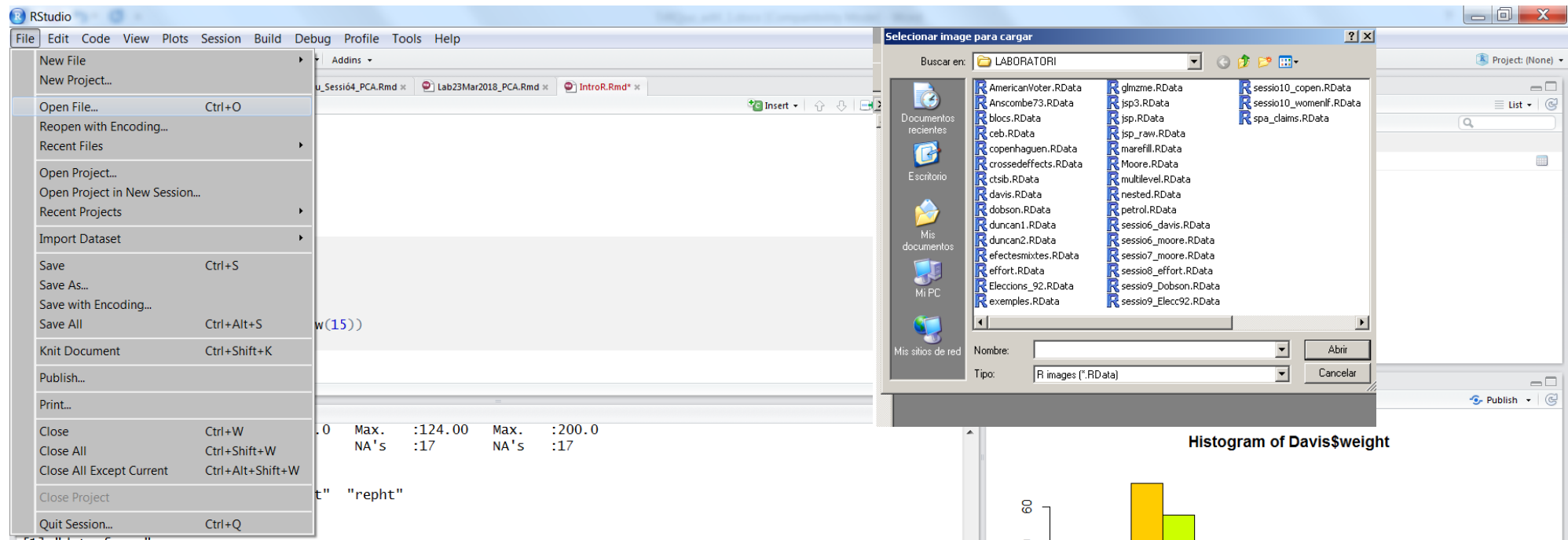
3. INTRODUCTION TO R

- RStudio for Windows: Basic Input/Output (R is *case sensitive*)
- Select working directory in R console window (Change dir / Cambiar dir ...)

The screenshot shows the RStudio environment with the following components:

- Source Editor:** Contains R code for loading the 'car' package, reading the 'Davis' dataset, and creating a histogram of 'Davis\$weight' with 15 bins and a rainbow color palette.
- Console:** Displays the output of the R code, including the attributes of the 'Davis' data frame and the execution of the histogram command.
- Environment:** Shows the 'Davis' data frame with 200 observations and 5 variables.
- Files:** Shows the project files, including 'DRT-PassDA_v4.Rmd' and 'ODEM...'. A 'Choose directory' dialog box is open, showing the path 'E:\LTIDIA\LIDIA\MLG2000\MLG2_07_1\LABORATORI'.
- Plots:** Displays a histogram titled 'Histogram of Davis\$weight' with the x-axis labeled 'Davis\$weight' and the y-axis labeled 'Frequency'.

INTRODUCTION TO R: LOAD WORKSPACE (RETRIEVE PREVIOUS USED DATA)



- Open (load) and Save Workspace - *File Menu (Archivo)*

File/Archivo → Cargar área de trabajo (load workspace)

File/Archivo → Guardar área de trabajo (save workspace)

Example: Open/Load Davis.RData from a Workspace.

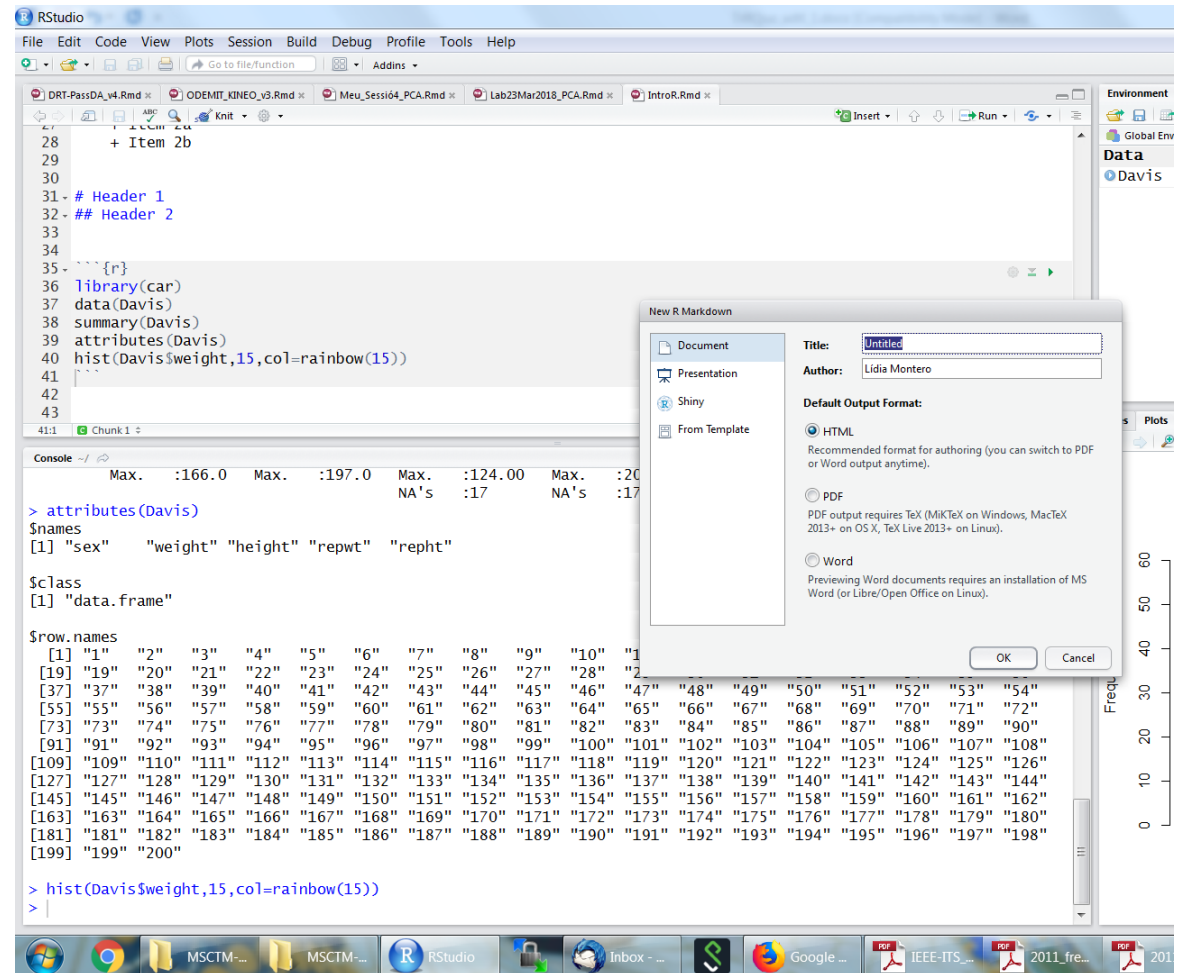
- To exit: *File/Archivo → Salir* or `quit()` command in R Console

INTRODUCTION TO R: SCRIPTS

From **Archivo (File)** menu: you can **open, close, save, create a new, save as** scripts.

Scripts are text files containing R command. Always use them to track lab session commands

Markdown documents are dynamic documents combining ordinary text and R commands. They can be interpreted to produce an output: html, pdf or word.

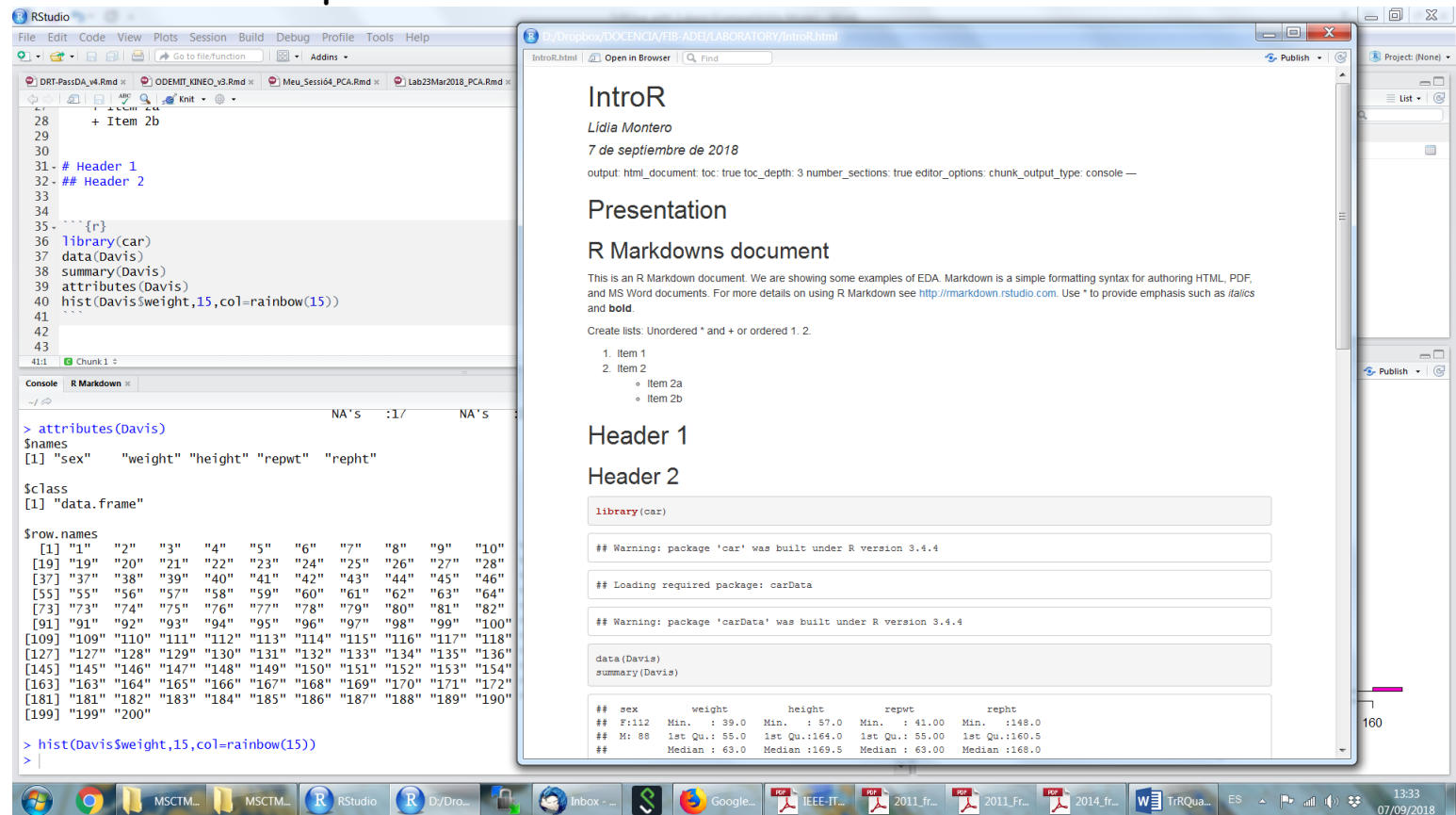


INTRODUCTION TO R: FILE MENU

Knit to produce R Markdown output:

Critical Elements in R:

- Expressions and Objects (escalars, vectors, matrices, lists, etc)
- The basic object is a list: `list()`.
- Data matrix - rows are individuals and columns are variables: `data.frame`.



INTRODUCTION TO R: CONSOLE, DEVICES AND SCRIPTS

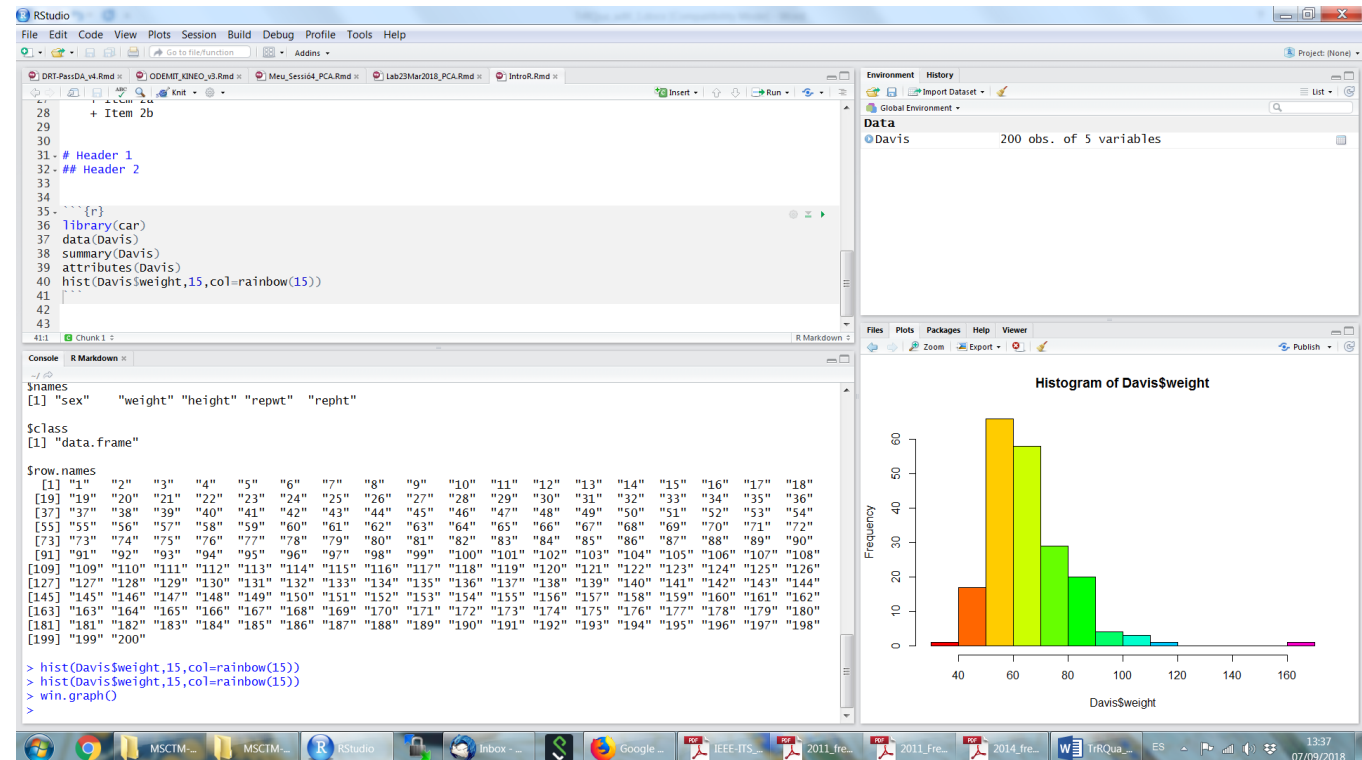
Available ones:

R Console (to write command and obtain results)

As many script windows as you want.

Data

Command `win.graph()` to create a new graphic device.



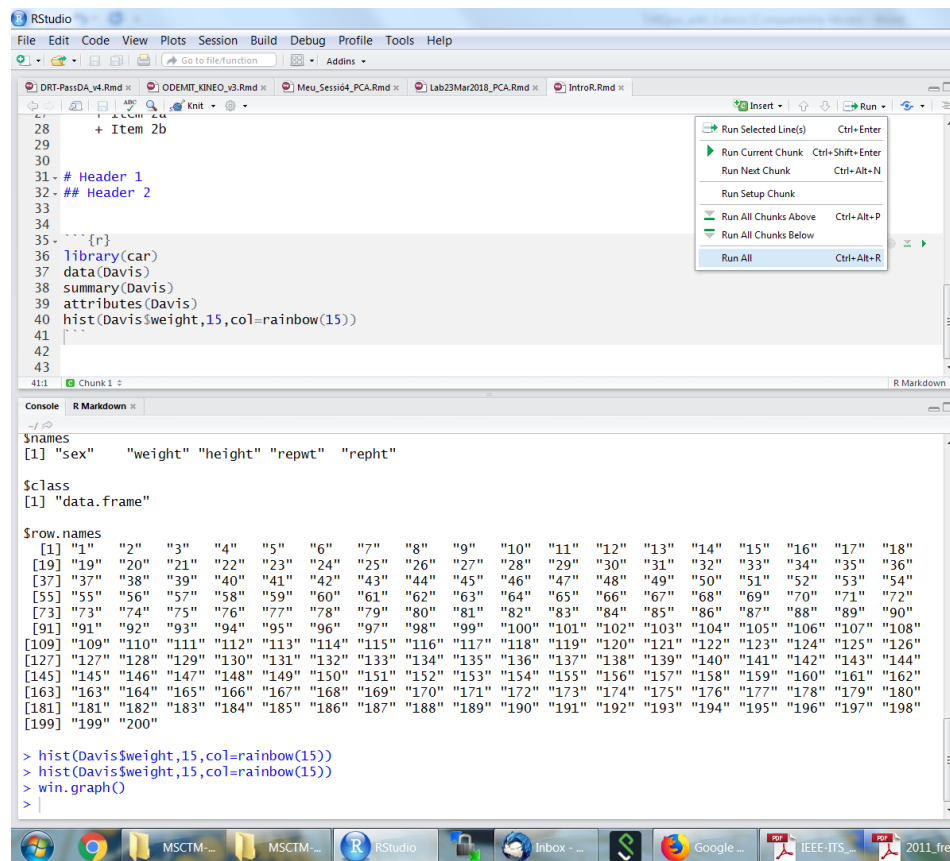
Graphic Devices: *R graphics* has a matrix structure that allows to obtain several figures: for ex. 2 rows and 2 columns

`par(mfrow=c(2,2))`

INTRODUCTION TO R: COMMAND STRUCTURE

R Command structure:

- > *Command parameters* <CR>
- > *Command parameters ; Command parameters* <CR>



The screenshot shows the RStudio environment. The script editor contains R code for loading the 'car' package, loading the 'Davis' dataset, and creating a histogram of 'Davis\$weight' with 15 bins and a rainbow color palette. The console window shows the output of the commands, including the variable names, class, and a large matrix of row names. The Run menu is open, showing options like 'Run Selected Line(s)', 'Run Current Chunk', 'Run Next Chunk', 'Run Setup Chunk', 'Run All Chunks Above', 'Run All Chunks Below', and 'Run All'.

```

28 + Item 2b
29
30
31 # Header 1
32 ## Header 2
33
34
35 {r}
36 library(car)
37 data(Davis)
38 summary(Davis)
39 attributes(Davis)
40 hist(Davis$weight,15,col=rainbow(15))
41
42
43
44 Chunk 1
  
```

```

$names
[1] "sex" "weight" "height" "repwt" "repht"

$class
[1] "data.frame"

$row.names
[1] "1" "2" "3" "4" "5" "6" "7" "8" "9" "10" "11" "12" "13" "14" "15" "16" "17" "18"
[19] "19" "20" "21" "22" "23" "24" "25" "26" "27" "28" "29" "30" "31" "32" "33" "34" "35" "36"
[37] "37" "38" "39" "40" "41" "42" "43" "44" "45" "46" "47" "48" "49" "50" "51" "52" "53" "54"
[55] "55" "56" "57" "58" "59" "60" "61" "62" "63" "64" "65" "66" "67" "68" "69" "70" "71" "72"
[73] "73" "74" "75" "76" "77" "78" "79" "80" "81" "82" "83" "84" "85" "86" "87" "88" "89" "90"
[91] "91" "92" "93" "94" "95" "96" "97" "98" "99" "100" "101" "102" "103" "104" "105" "106" "107" "108"
[109] "109" "110" "111" "112" "113" "114" "115" "116" "117" "118" "119" "120" "121" "122" "123" "124" "125" "126"
[127] "127" "128" "129" "130" "131" "132" "133" "134" "135" "136" "137" "138" "139" "140" "141" "142" "143" "144"
[145] "145" "146" "147" "148" "149" "150" "151" "152" "153" "154" "155" "156" "157" "158" "159" "160" "161" "162"
[163] "163" "164" "165" "166" "167" "168" "169" "170" "171" "172" "173" "174" "175" "176" "177" "178" "179" "180"
[181] "181" "182" "183" "184" "185" "186" "187" "188" "189" "190" "191" "192" "193" "194" "195" "196" "197" "198"
[199] "199" "200"

> hist(Davis$weight,15,col=rainbow(15))
> hist(Davis$weight,15,col=rainbow(15))
> win.graph()
>
  
```

To be written in R console or any script or inside a chunk in R Markdown.

To execute a command line included in a script: press <ctrl- Enter>.

To execute several command lines: select and <ctrl- Enter>.

To execute one or several chunks use R Studio menu.

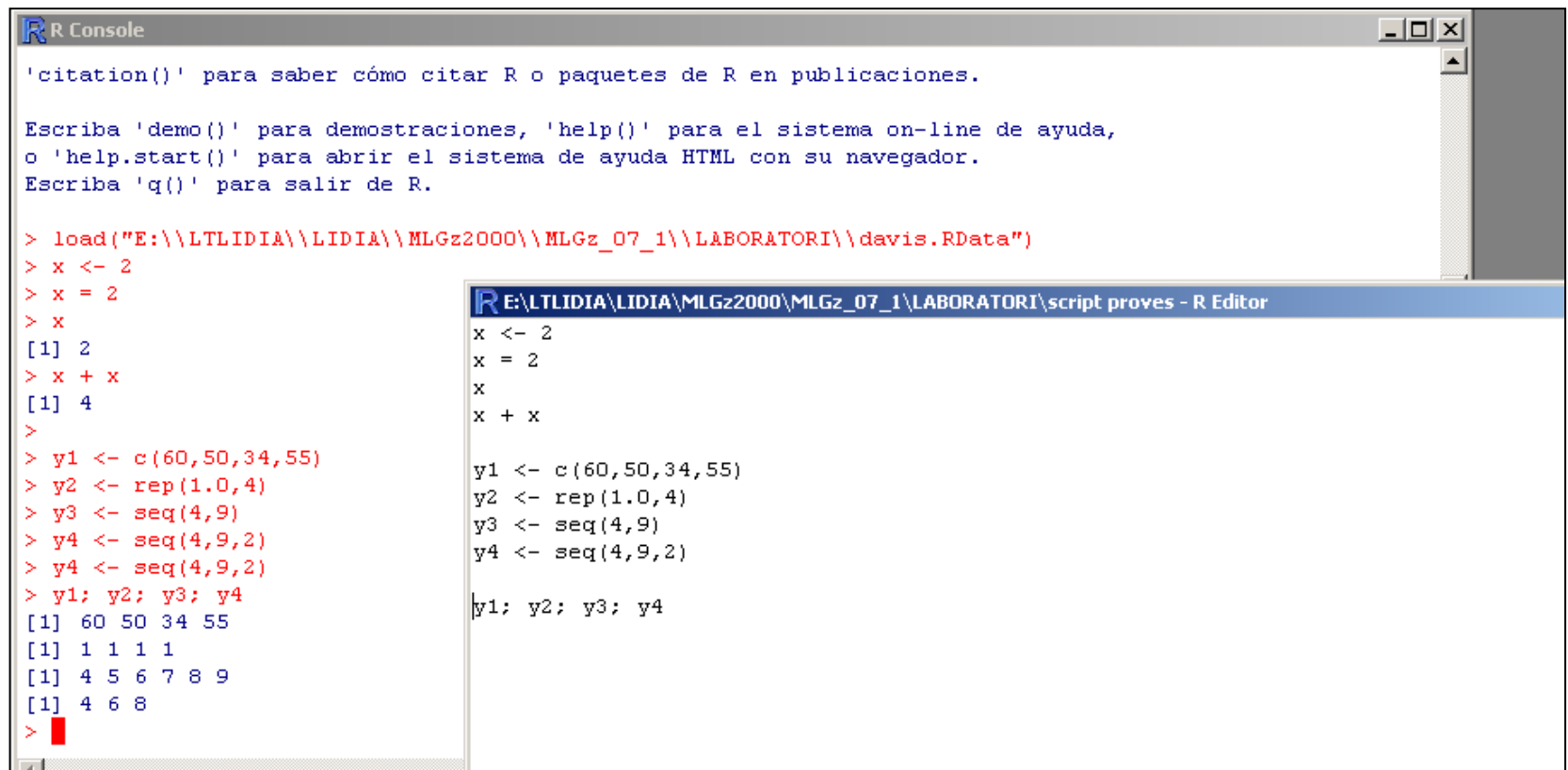
INTRODUCTION TO R: SEQUENCES ...

Example: create a vector with 4 integer elements

Concatenation: `c(.)`

Sequence:
`seq(.)`

Replication:
`rep(.)`



```
R Console
'citation()' para saber cómo citar R o paquetes de R en publicaciones.

Escriba 'demo()' para demostraciones, 'help()' para el sistema on-line de ayuda,
o 'help.start()' para abrir el sistema de ayuda HTML con su navegador.
Escriba 'q()' para salir de R.

> load("E:\\LTLIDIA\\LIDIA\\MLGz2000\\MLGz_07_1\\LABORATORI\\davis.RData")
> x <- 2
> x = 2
> x
[1] 2
> x + x
[1] 4
>
> y1 <- c(60,50,34,55)
> y2 <- rep(1.0,4)
> y3 <- seq(4,9)
> y4 <- seq(4,9,2)
> y4 <- seq(4,9,2)
> y1; y2; y3; y4
[1] 60 50 34 55
[1] 1 1 1 1
[1] 4 5 6 7 8 9
[1] 4 6 8
>

R Editor
E:\\LTLIDIA\\LIDIA\\MLGz2000\\MLGz_07_1\\LABORATORI\\script proves - R Editor
x <- 2
x = 2
x
x + x

y1 <- c(60,50,34,55)
y2 <- rep(1.0,4)
y3 <- seq(4,9)
y4 <- seq(4,9,2)

y1; y2; y3; y4
```

INTRODUCTION TO R - BASIC OBJECTS:

Important Objects: lists, vectors, matrices and arrays

```

RGui
Archivo  Editar  Paquetes  Ventanas  Ayuda

R Console

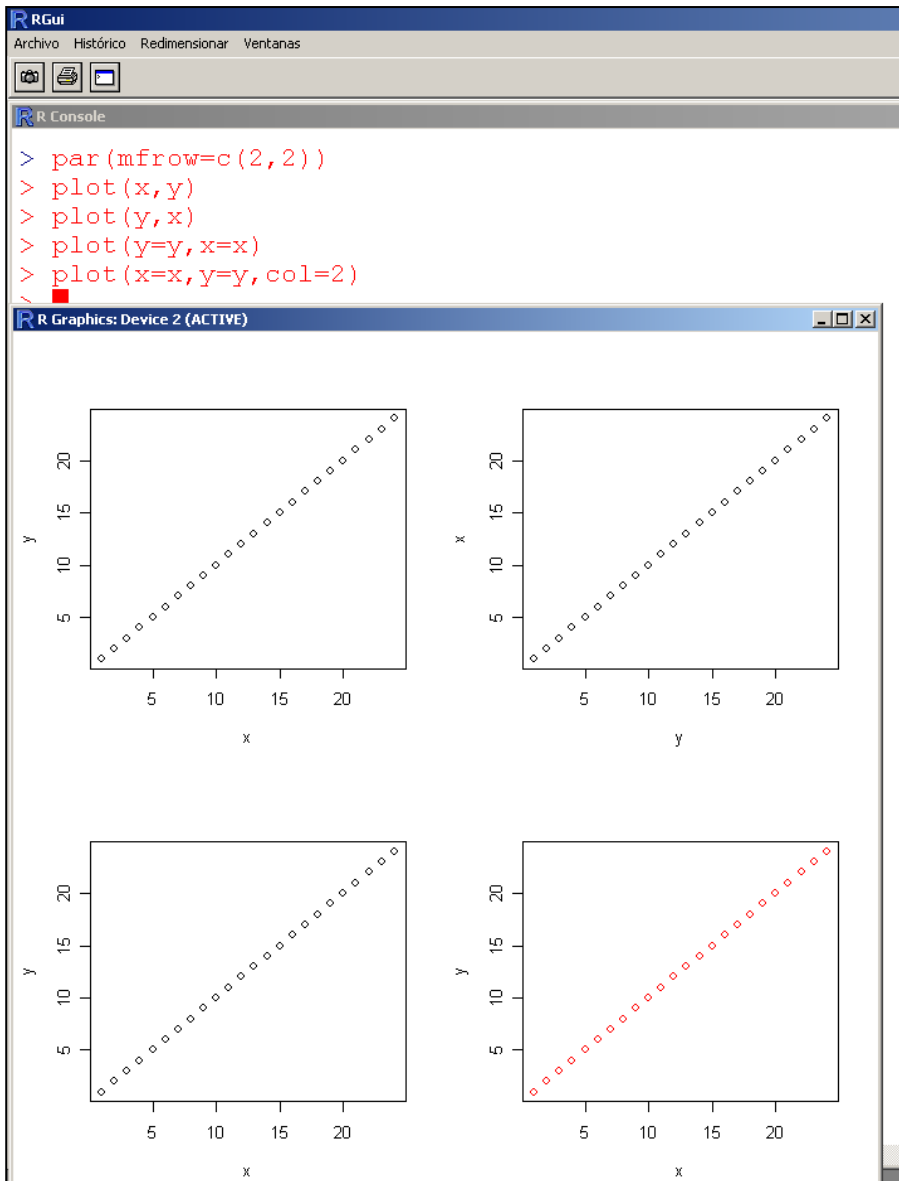
> x<-1:24
> dim(x)<-c(6,4)
> x<-matrix(1:24, nrow=6)
> rownames(x) <- letters[1:6]
> colnames(x)<-c("A","B","C","D")
> colnames(x)<-list("A","B","C","D")
> y<-x
> dim(y)<-c(4,3,2)
> x
  A  B  C  D
a 1  7 13 19
b 2  8 14 20
c 3  9 15 21
d 4 10 16 22
e 5 11 17 23
f 6 12 18 24
> y
, , 1
[1,] 1  5  9
[2,] 2  6 10
[3,] 3  7 11
[4,] 4  8 12
, , 2
[1,] 13 17 21
[2,] 14 18 22
[3,] 15 19 23
[4,] 16 20 24
>
    
```

- Matrices are arrays of 2 dimensions.
- Matrices and arrays of dimension greater than a 2 are allowed.
- Related commands: rownames(), colnames(), dim() to check dimensions.
- To create matrices:

```

> x<-matrix(1:24, nrow=6)
> rownames(x) <- letters
[1:6]
> colnames(x)<-
c("A","B","C","D")
> colnames(x)<-
list("A","B","C","D")
    
```

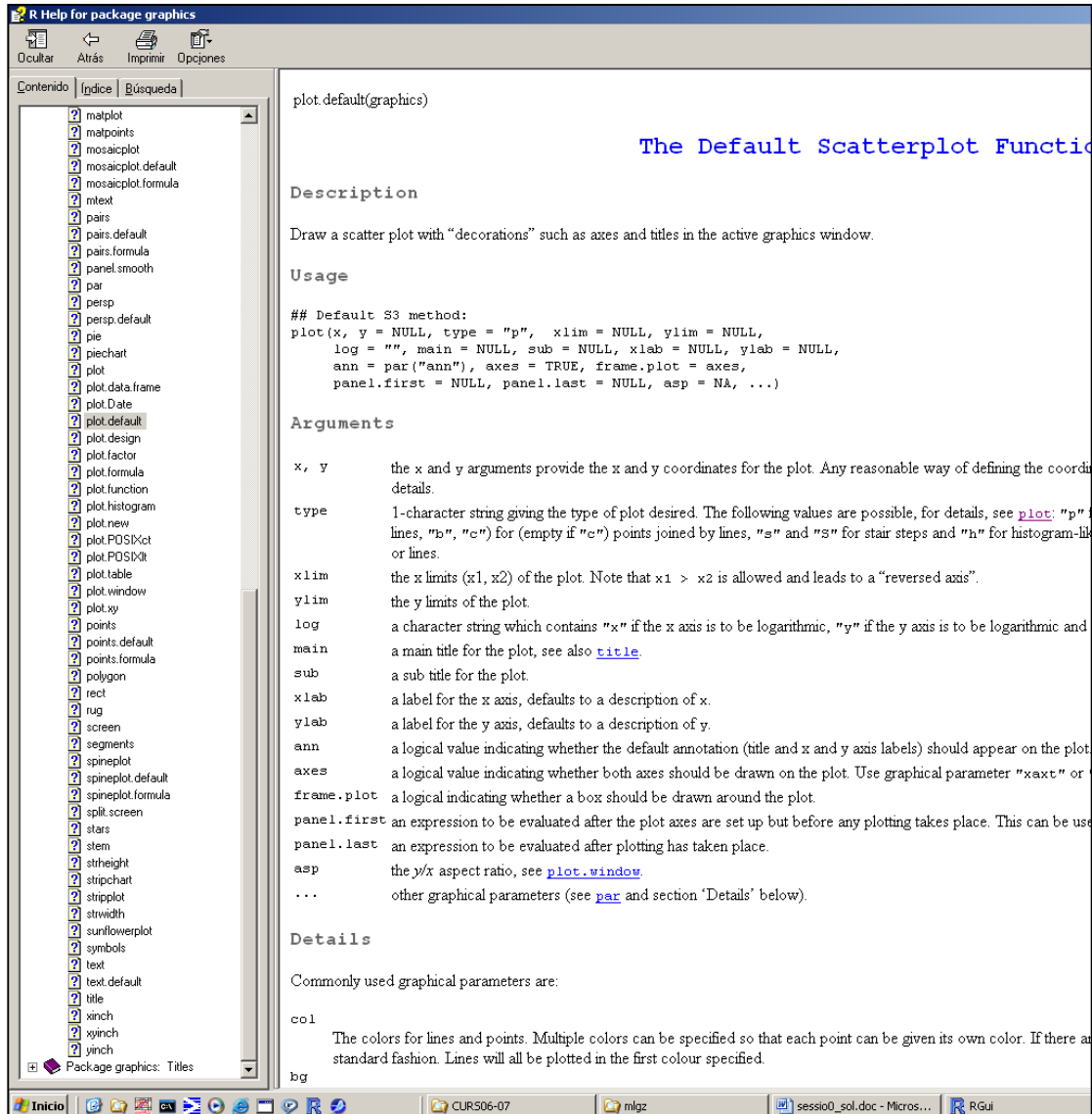
INTRODUCTION TO R - FUNCTIONS AND ARGUMENTS



Functions and arguments:

- An R function might be a mathematic or statistical function, as `log(x)`, but there are additional functions as `plot(height, weight)`.
- Functions have actual parameters (**actual arguments**) and formal parameters (*formal arguments*).
- Most arguments have default values and can be omitted.
- R functions arguments can be positionally matched (*positional matching*) or by name (*keyword matching*). You can mix *positional matching* with *matching by name*.

INTRODUCTION TO R - FUNCTIONS AND ARGUMENTS



The screenshot shows the R Help window for the 'graphics' package, specifically the documentation for the `plot.default` function. The left pane shows a list of functions, with `plot.default` selected. The right pane displays the function's description, usage, arguments, and details.

plot.default(graphics)

The Default Scatterplot Function

Description
Draw a scatter plot with "decorations" such as axes and titles in the active graphics window.

Usage
Default S3 method:
`plot(x, y = NULL, type = "p", xlim = NULL, ylim = NULL, log = "", main = NULL, sub = NULL, xlab = NULL, ylab = NULL, ann = par("ann"), axes = TRUE, frame.plot = axes, panel.first = NULL, panel.last = NULL, asp = NA, ...)`

Arguments

- `x, y`: the x and y arguments provide the x and y coordinates for the plot. Any reasonable way of defining the coordinates.
- `type`: 1-character string giving the type of plot desired. The following values are possible, for details, see [plot](#): "p" for points, "l" for lines, "b" for (empty if "c") points joined by lines, "s" and "S" for stair steps and "h" for histogram-like or lines.
- `xlim`: the x limits (x1, x2) of the plot. Note that `x1 > x2` is allowed and leads to a "reversed axis".
- `ylim`: the y limits of the plot.
- `log`: a character string which contains "x" if the x axis is to be logarithmic, "y" if the y axis is to be logarithmic and "" otherwise.
- `main`: a main title for the plot, see also [title](#).
- `sub`: a sub title for the plot.
- `xlab`: a label for the x axis, defaults to a description of x.
- `ylab`: a label for the y axis, defaults to a description of y.
- `ann`: a logical value indicating whether the default annotation (title and x and y axis labels) should appear on the plot.
- `axes`: a logical value indicating whether both axes should be drawn on the plot. Use graphical parameter "xaxt" or "yaxt" to control this.
- `frame.plot`: a logical indicating whether a box should be drawn around the plot.
- `panel.first`: an expression to be evaluated after the plot axes are set up but before any plotting takes place. This can be used to draw a grid or other decorations.
- `panel.last`: an expression to be evaluated after plotting has taken place.
- `asp`: the y/x aspect ratio, see [plot.window](#).
- `...`: other graphical parameters (see [par](#) and section 'Details' below).

Details
Commonly used graphical parameters are:

- `col`: The colors for lines and points. Multiple colors can be specified so that each point can be given its own color. If there are more points than colors, the colors will be recycled in standard fashion. Lines will all be plotted in the first colour specified.
- `bg`: The background color of the plot area.

For example:

- `plot(height, weight)` is a *positional matching call*.
- `plot(height, weight, col=2)` adds a *keyword matching argument*.
- `plot(y=weight, x=height, col=2)` allows arguments in any order (all arguments by *keyword matching*).
- Check parameters in:

`help(plot)`
`args(plot.default)`

INTRODUCTION TO R - FACTORS

```

RGui
Archivo  Editar  Visualizar  Misc  Paquetes  Ventanas  Ayuda

R Console
> opinio <- sample(seq(1:5), 20, replace = TRUE)
> opinio
[1] 3 3 2 2 2 5 3 5 1 5 2 5 3 2 4 4 5 2 4 5
> summary(opinio)
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  1.00   2.00   3.00   3.35   5.00   5.00
>
> opinio1<- factor(opinio, labels=c("molt desacord","desacord","no sap","d'acord","molt d'acord"))
> summary(opinio1)
molt desacord      desacord      no sap      d'acord      molt d'acord
           1              6              4              3              6
>
> opinio2<-factor(opinio)
> opinio2
[1] 3 3 2 2 2 5 3 5 1 5 2 5 3 2 4 4 5 2 4 5
Levels: 1 2 3 4 5
>
> opinio3<- factor(opinio,levels=1:5 )
> opinio3
[1] 3 3 2 2 2 5 3 5 1 5 2 5 3 2 4 4 5 2 4 5
Levels: 1 2 3 4 5
> summary(opinio3)
 1  2  3  4  5
1  6  4  3  6
> levels(opinio3)=c("molt desacord","desacord","no sap","d'acord","molt d'acord")
> opinio3
[1] no sap      no sap      desacord    desacord    desacord    molt d'acord no sap
[8] molt d'acord molt desacord molt d'acord desacord    molt d'acord no sap      desacord
[15] d'acord      d'acord      molt d'acord desacord    d'acord      molt d'acord
Levels: molt desacord desacord no sap d'acord molt d'acord
> summary(opinio3)
molt desacord      desacord      no sap      d'acord      molt d'acord
           1              6              4              3              6
>
> help(factor)
> ordered(opinio2)
[1] 3 3 2 2 2 5 3 5 1 5 2 5 3 2 4 4 5 2 4 5
Levels: 1 < 2 < 3 < 4 < 5
> summary(opinio2)
 1  2  3  4  5
1  6  4  3  6
  
```

Factors:

- Vectors to represent qualitative variables.
- Ordered or not.
- Consider values as *levels* or *labels*.
- To convert labels into numeric values:
`as.numeric(factor)`

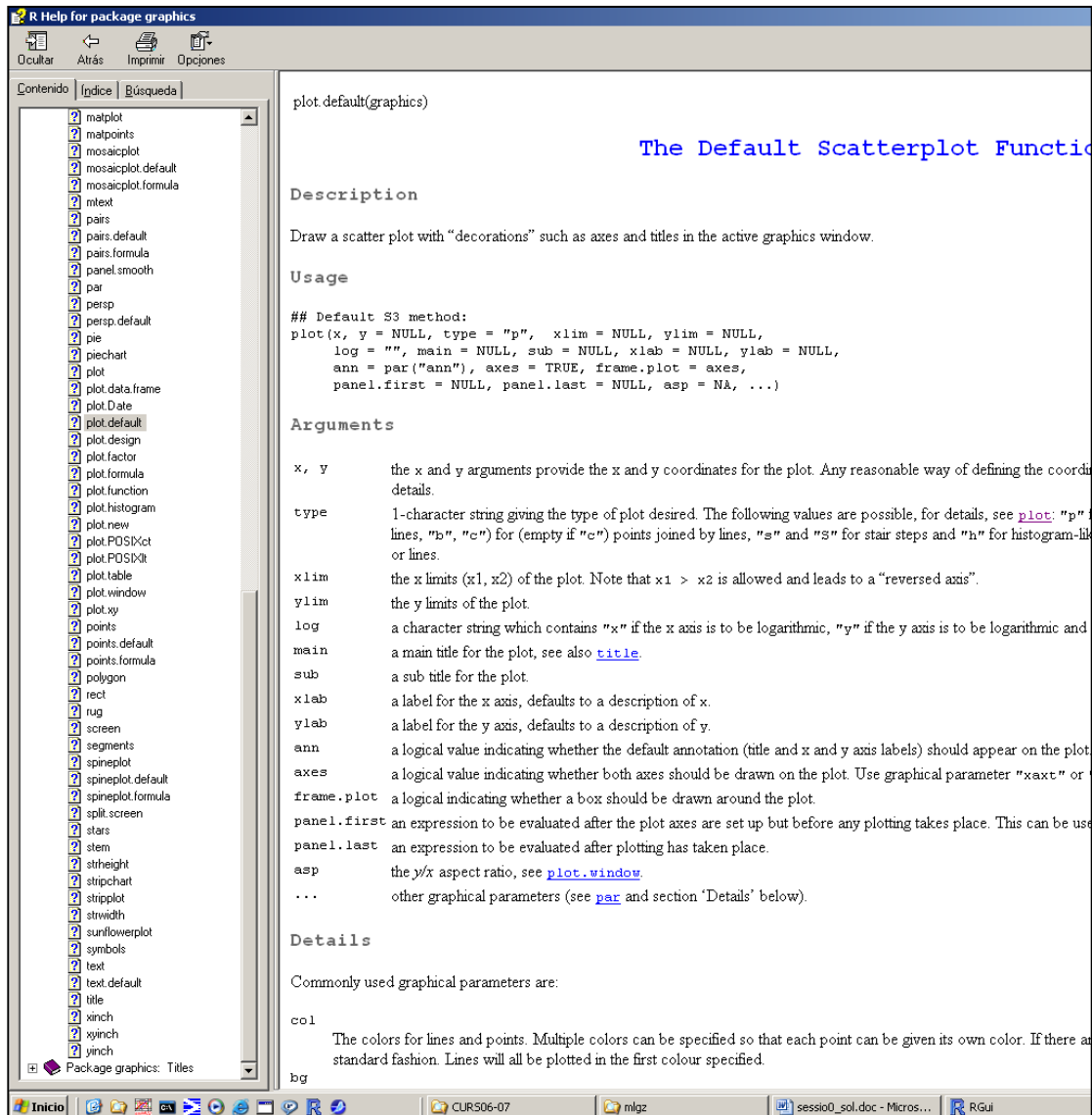
```

factor(x = character(),
       levels = sort(unique.default(x),
                       na.last = TRUE),
       labels = levels, exclude = NA,
       ordered = is.ordered(x))
  
```

```

ordered(x, ...)
is.factor(x)
is.ordered(x)
as.factor(x)
as.ordered(x)
  
```

INTRODUCTION TO R - NEW VARIABLES



The screenshot shows the R Help window for the 'graphics' package, specifically the 'plot.default' function. The left pane lists various plot functions, with 'plot.default' selected. The right pane displays the function's description, usage, arguments, and details.

plot.default(graphics)

The Default Scatterplot Function

Description
Draw a scatter plot with "decorations" such as axes and titles in the active graphics window.

Usage
Default S3 method:
plot(x, y = NULL, type = "p", xlim = NULL, ylim = NULL, log = "", main = NULL, sub = NULL, xlab = NULL, ylab = NULL, ann = par("ann"), axes = TRUE, frame.plot = axes, panel.first = NULL, panel.last = NULL, asp = NA, ...)

Arguments

x, y	the x and y arguments provide the x and y coordinates for the plot. Any reasonable way of defining the coordinates.
type	1-character string giving the type of plot desired. The following values are possible, for details, see plot : "p" for points, "l" for lines, "b" for (empty if "c") points joined by lines, "s" and "S" for stair steps and "h" for histogram-like or lines.
xlim	the x limits (x1, x2) of the plot. Note that x1 > x2 is allowed and leads to a "reversed axis".
ylim	the y limits of the plot.
log	a character string which contains "x" if the x axis is to be logarithmic, "y" if the y axis is to be logarithmic and "" for neither.
main	a main title for the plot, see also title .
sub	a sub title for the plot.
xlab	a label for the x axis, defaults to a description of x.
ylab	a label for the y axis, defaults to a description of y.
ann	a logical value indicating whether the default annotation (title and x and y axis labels) should appear on the plot.
axes	a logical value indicating whether both axes should be drawn on the plot. Use graphical parameter "xaxt" or "yaxt" for details.
frame.plot	a logical indicating whether a box should be drawn around the plot.
panel.first	an expression to be evaluated after the plot axes are set up but before any plotting takes place. This can be used to draw a grid.
panel.last	an expression to be evaluated after plotting has taken place.
asp	the y/x aspect ratio, see plot.window .
...	other graphical parameters (see par and section 'Details' below).

Details
Commonly used graphical parameters are:

col	The colors for lines and points. Multiple colors can be specified so that each point can be given its own color. If there are more points than colors, the colors will be recycled in standard fashion. Lines will all be plotted in the first colour specified.
bg	

Manipulation of data matrices (data.frame) :

- Create a new variable from existent variables in the current workspace using mathematic functions:
 - For example, $y \leftarrow \log(x) + z + 4.5$ (x and z existent vectors).
 - In a data.frame: `attach(Davis)`
 - `weight2 <- weight^2` new variable not included in Davis data.frame.
 - `Davis$weight2 <- weight^2` new variable included in Davis data.frame, but a `detach(Davis)` and new `attach(Davis)`.
- Remove an object: `rm(object-name)`.

INTRODUCTION TO R - NEW VARIABLES

- To remove a variable included in a data.frame: `Davis$weight2<-NULL`.
- Remove all objects in the current workspace: `rm(list=ls())`.
- Remove all objects in the current workspace beginning with 'la': `rm(list=ls(pattern="la"))`.
- R can deal with multiple datasets at the same time.
 - You just need to specify the name of the dataset and a "\$" symbol before each variable name.
 - If you don't want to write again and again the name of the dataset as a prefix for each variable, you can use `attach()`

```

RGui - [R Console]
Archivo  Editar  Visualizar  Misc  Paquetes  Ventanas  Ayuda

> ls()
[1] "davis"      "last.warning" "opinio"      "opinio1"     "opinio2"     "opinio3"
> # M'interessa la classe de davis (és un data.frame o matriu de dades)
> # Vec les columnes que conté (característiques de les observacions)
> attributes(davis)
$names
[1] "id"      "sex"      "weight"    "height"    "r_weight"  "r_height"

$row.names
[1] "1"  "2"  "3"  "4"  "5"  "6"  "7"  "8"  "9"  "10" "11" "12" "13" "14" "15" "16" "17" "18"
[19] "19" "20" "21" "22" "23" "24" "25" "26" "27" "28" "29" "30" "31" "32" "33" "34" "35" "36"
[37] "37" "38" "39" "40" "41" "42" "43" "44" "45" "46" "47" "48" "49" "50" "51" "52" "53" "54"
[55] "55" "56" "57" "58" "59" "60" "61" "62" "63" "64" "65" "66" "67" "68" "69" "70" "71" "72"
[73] "73" "74" "75" "76" "77" "78" "79" "80" "81" "82" "83" "84" "85" "86" "87" "88" "89" "90"
[91] "91" "92" "93" "94" "95" "96" "97" "98" "99" "100" "101" "102" "103" "104" "105" "106" "107" "108"
[109] "109" "110" "111" "112" "113" "114" "115" "116" "117" "118" "119" "120" "121" "122" "123" "124" "125" "126"
[127] "127" "128" "129" "130" "131" "132" "133" "134" "135" "136" "137" "138" "139" "140" "141" "142" "143" "144"
[145] "145" "146" "147" "148" "149" "150" "151" "152" "153" "154" "155" "156" "157" "158" "159" "160" "161" "162"
[163] "163" "164" "165" "166" "167" "168" "169" "170" "171" "172" "173" "174" "175" "176" "177" "178" "179" "180"
[181] "181" "182" "183" "184" "185" "186" "187" "188" "189" "190" "191" "192" "193" "194" "195" "196" "197" "198"
[199] "199" "200"

$class
[1] "data.frame"

> # O només les columnes
> names(davis)
[1] "id"      "sex"      "weight"    "height"    "r_weight"  "r_height"
> # Quin és el nb d'observacions: dimensió files
> dim( davis )
[1] 200      6
> dim( davis )[ 1 ]
[1] 200
> # No es pot crear el quadrat del pes com a variable no és visible
> pes2 <- weight^2
Error: object "weight" no encontrado
> # La variable és visible referenciada dins data.frame davis
> pes2 <- davis$weight^2
> ls()
[1] "davis"      "last.warning" "opinio"      "opinio1"     "opinio2"     "opinio3"     "pes2"
>

```


INTRODUCTION TO R - SCOPE OF VISIBILITY: ATTACH COMMAND

```

RGui - [R Console]
R Archivo Editar Visualizar Misc Paquetes Ventanas Ayuda

> # Podem fer visibles totes les variables d'un data.frame
> attach(davis)
> # Ara es pot crear una nova variable fora del data.frame
> pes2 <- weight^2
> ls()
[1] "davis"          "last.warning" "opinio"        "opinio1"       "opinio2"       "opinio3"       "pes2"
> detach(davis)
> # Si es vol crear dins del data.frame davis
> davis<-transform( davis, pes2=weight^2 )
> summary(davis)
      id      sex      weight      height      r_weight      r_height      pes2
Min.   : 1.00   F:112   Min.   : 39.0   Min.   : 57.0   Min.   : 41.00   Min.   :148.0   Min.   : 1521
1st Qu.: 50.75   M: 88    1st Qu.: 55.0   1st Qu.:164.0   1st Qu.: 55.00   1st Qu.:160.5   1st Qu.: 3025
Median :100.50                Median : 63.0   Median :169.5   Median : 63.00   Median :168.0   Median : 3969
Mean   :100.50                Mean   : 65.8   Mean   :170.0   Mean   : 65.62   Mean   :168.5   Mean   : 4556
3rd Qu.:150.25                3rd Qu.: 74.0   3rd Qu.:177.2   3rd Qu.: 73.50   3rd Qu.:175.0   3rd Qu.: 5476
Max.   :200.00                Max.   :166.0   Max.   :197.0   Max.   :124.00   Max.   :200.0   Max.   :27556
                        NA's   : 17.00   NA's   : 17.0
> # O bé:
> davis$pes2<- davis$weight
> # Esborrar objecte de l'espai de treball: rm()
> rm( pes2 )
> # Esborrar una columna d'un data.frame
> davis$pes2<- NULL
> # Noms de les característiques (variables) d'un data.frame
> names( davis )
[1] "id"      "sex"     "weight"  "height"  "r_weight" "r_height"
>

```

- `attach()` command can be dangerous. Use `detach()` as soon as possible.
- **Suggested command:** Evaluate an R expression in an environment constructed from data, possibly modifying the original data:

```
with(Davis, {boxplot(height) ; summary(height) })
```

INTRODUCTION TO R - NEW VARIABLES

```

RGui - [R Console]
Archivo  Editar  Visualizar  Misc  Paquetes  Ventanas  Ayuda

> names( davis )
[1] "id"      "sex"      "weight"   "height"   "r_weight" "r_height"
> # Vull un data.frame reduït sense pes i alçada reportat
> davis1 <- davis[,1:4]
> attributes( davis1 )
$names
[1] "id"      "sex"      "weight"   "height"

$class
[1] "data.frame"

$row.names
 [1] "1"  "2"  "3"  "4"  "5"  "6"  "7"  "8"  "9"  "10" "11" "12" "13" "14" "15" "16" "17" "18"
[19] "19" "20" "21" "22" "23" "24" "25" "26" "27" "28" "29" "30" "31" "32" "33" "34" "35" "36"
[37] "37" "38" "39" "40" "41" "42" "43" "44" "45" "46" "47" "48" "49" "50" "51" "52" "53" "54"
[55] "55" "56" "57" "58" "59" "60" "61" "62" "63" "64" "65" "66" "67" "68" "69" "70" "71" "72"
[73] "73" "74" "75" "76" "77" "78" "79" "80" "81" "82" "83" "84" "85" "86" "87" "88" "89" "90"
[91] "91" "92" "93" "94" "95" "96" "97" "98" "99" "100" "101" "102" "103" "104" "105" "106" "107" "108"
[109] "109" "110" "111" "112" "113" "114" "115" "116" "117" "118" "119" "120" "121" "122" "123" "124" "125" "126"
[127] "127" "128" "129" "130" "131" "132" "133" "134" "135" "136" "137" "138" "139" "140" "141" "142" "143" "144"
[145] "145" "146" "147" "148" "149" "150" "151" "152" "153" "154" "155" "156" "157" "158" "159" "160" "161" "162"
[163] "163" "164" "165" "166" "167" "168" "169" "170" "171" "172" "173" "174" "175" "176" "177" "178" "179" "180"
[181] "181" "182" "183" "184" "185" "186" "187" "188" "189" "190" "191" "192" "193" "194" "195" "196" "197" "198"
[199] "199" "200"

> names( davis1 )
[1] "id"      "sex"      "weight"   "height"
> # Vull un data.frame pels homes sex == M i un altre per dones amb totes les característiques
>
> homes <- (davis$sex=='M')
> davisM <- davis[ homes, ]
> #davisM
> davisF <- davis[ davis$sex=='F', ]
>
> dim( davisM )
[1] 88  6
> dim( davisF )
[1] 112  6
> # Per quedar-me amb les observacions 20 a 110 més 119 i les 4 columnes:
> davis2<- davis[ c(20:110,119), 1:4]
> dim( davis2 )
[1] 92  4
  
```

Access to columns in a *data.frame* as if there were matrices

INTRODUCTION TO R - INDEXING VARIABLES

- Indexing vectors?: `weight2[29]` position 29 in `weight2` vector.
- Indexing matrices?: `Davis[2,4]` observation 2 and variable in 4th column.
- Row number 2 in a `data.frame`: `Davis[2,]`.
- Column number 4 in a `data.frame` : `Davis[, 4]` (*height* is a vector with 200 observations).
- A set of columns: `Davis[, c(1,3:4)]`.
- A set of rows (observations):
 - `Davis[1:100,]` observations 1, 2, 3 ... 100
 - `Davis[seq(1,100,2),]` ... observations 1, 3, 5, 7 ...
 - `Davis[sample(100:200,50,rep=T),]` 50 random rows contained in row numbers 1 to 100.
 - `Davis[rep(c(1,2),10) ,]` observations (repeated)

1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2

```

> davis3<- Davis[ sample(100:200,10,rep=T), ]
> table(Davis3$id)
104 105 141 173 174 175 177 180 194
  1    1    1    1    1    1    2    1    1
    
```

INTRODUCTION TO R - RECODIFICATION OF VARIABLES

```

R Console
> summary( davis$weight )
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  39.0   55.0   63.0   65.8   74.0   166.0
> davis$tipus <- factor(cut(davis$weight, 4)) # Discretització en 4 intervals
> table(davis$tipus)

(38.9,70.7] (70.7,102] (102,134] (134,166]
      142         55          2          1
> summary( davis$tipus )
(38.9,70.7] (70.7,102] (102,134] (134,166]
      142         55          2          1
> tapply( davis$weight, davis$tipus, median)
(38.9,70.7] (70.7,102] (102,134] (134,166]
       58        80       111       166
> # Discretització per 4 quartils
> davis$tipus <- factor(cut(davis$weight, quantile(davis$weight,c(0,1/4,2/4,3/4,1))))
> table(davis$tipus)

(39,55] (55,63] (63,74] (74,166]
     52      50       48       49
> tapply( davis$weight, davis$tipus, median)
(39,55] (55,63] (63,74] (74,166]
     52      59       68       82
> # Discretització en 4 intervals triats per l'usuari
> davis$tipus <- factor(cut(davis$weight, breaks=c(-1,55,65,75,200)))
> table(davis$tipus)

(-1,55] (55,65] (65,75] (75,200]
     53      64       38       45
> tapply( davis$weight, davis$tipus, median)
(-1,55] (55,65] (65,75] (75,200]
     52      61       69       83
> levels(davis$tipus)<-paste("TYPE",levels(davis$tipus), sep=":")
> summary(davis$tipus)
TYPE: (-1,55] TYPE: (55,65] TYPE: (65,75] TYPE: (75,200]
     53      64       38       45
> levels(davis$tipus) <- c("prim","normal","sobrepes","obes")
> summary(davis$tipus)
   prim  normal sobrepes   obes
     53     64      38     45
> █
    
```

Recodification: Create a new variable from an existent numerical one.

- Discretization of a numeric variable:
 - Equal length intervals.
 - Intervals selected by the users.
 - Intervals defined by quantiles.

INTRODUCTION TO R - DEFINING FACTORS

Recoding: Creating a new variable by working with ranges.

- Grouping categories: create a new variable using `ifelse()` sentence.

```
> as.numeric( davis$tipus )  
[1] 4 2 1 3 2 4 4 3 3 2 3 4 1 2 1 2 4 2 4 2 4 2 2 3 1 1 2 2 1 4 3 3 4 1 3 2 2 3 4 2 1 4 2 4 4 1 3 1 1  
[50] 3 4 2 3 4 2 2 3 3 3 2 3 3 3 4 4 2 2 2 1 4 2 3 1 2 1 1 1 2 3 3 2 1 1 2 2 2 2 1 2 2 3 2 4 3 4 2 4 1  
[99] 1 1 1 2 1 1 2 1 1 2 2 2 4 4 1 3 4 2 4 4 3 2 4 3 3 1 3 1 1 1 3 1 2 4 2 1 4 1 2 2 4 4 3 1 1 1 2 1 1  
[148] 2 3 4 2 1 1 1 1 1 3 2 2 2 1 1 1 2 2 2 2 3 4 1 4 1 2 4 2 3 2 3 4 3 4 1 4 2 3 2 2 2 4 1 4 4 2 1 2 3  
[197] 4 4 4 4  
  
> grup <- rep( 0, dim( davis )[1] )  
> grup <- factor(ifelse( as.numeric(davis$tipus)>2,1,0))  
> levels(grup) <- c("correcte","controlar")  
> summary(grup)  
correcte controlar  
      117      83  
  
>
```

INTRODUCTION TO R - EDA - BIVARIATE: NUMERIC VS FACTOR

TWO VARIABLES ARE INVOLVED:

RESPONSE VARIABLE IS NUMERIC, as `Davis$height`

EXPLANATORY VARIABLE IS A FACTOR, as `Davis$sex` (max 5-6 levels)

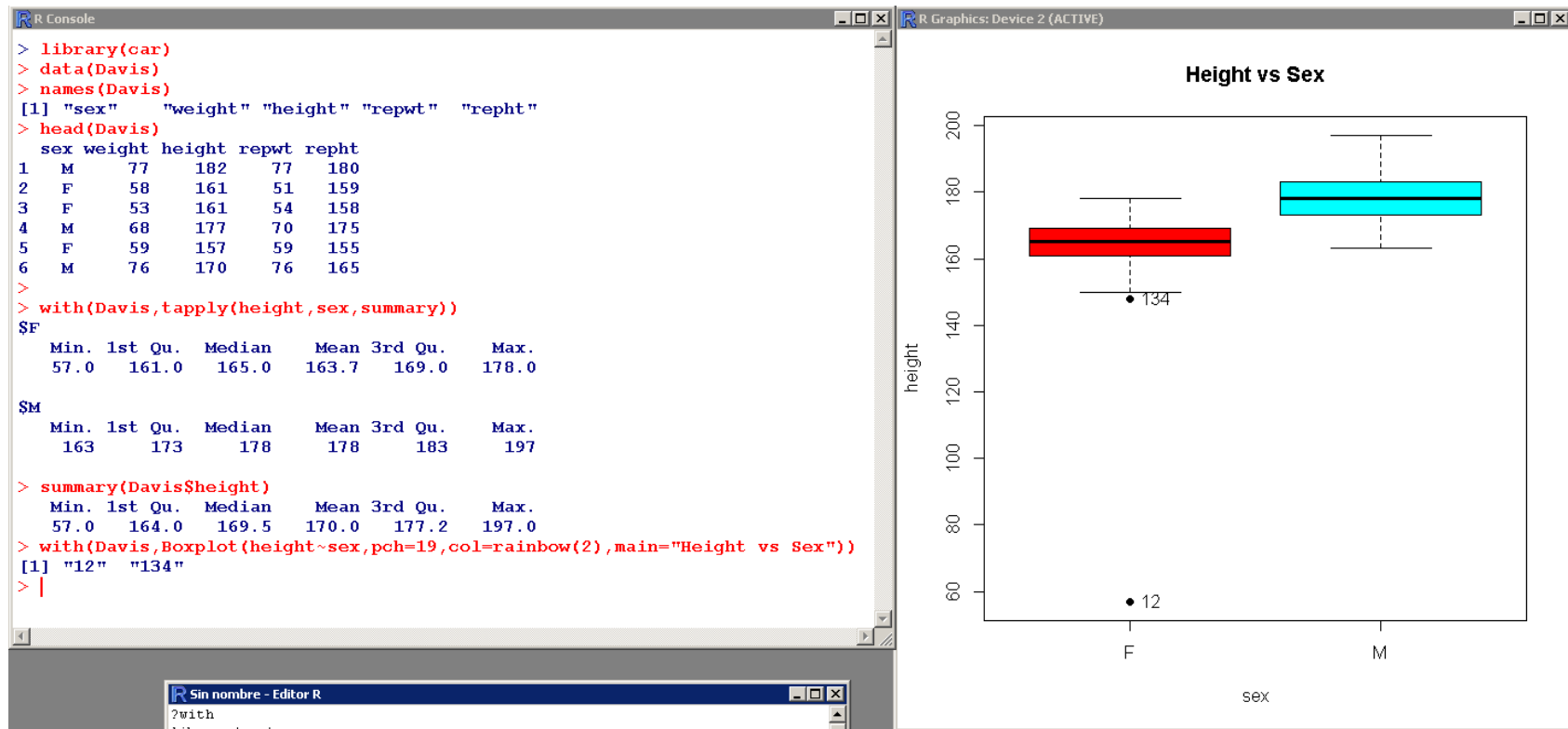
Goal: Do groups defined by levels of the factor determine a difference profile in the numeric response.

- Do height and sex show an independent behavior/profile? Statistical question: Is the profile of height the same for both levels of factor sex?
- If height and sex don't show any relationship- Statistical statement: The profile of height is the same for both levels in sex factor ?

EDA for a numeric variable according to groups defined by factor. Particular analysis:
ANOVA - Analysis of Variance

INTRODUCTION TO R - EDA - BIVARIATE: NUMERIC VS FACTOR

For example: Height (Y) vs Sex (A) - Formula expression in R: $Y \sim A$



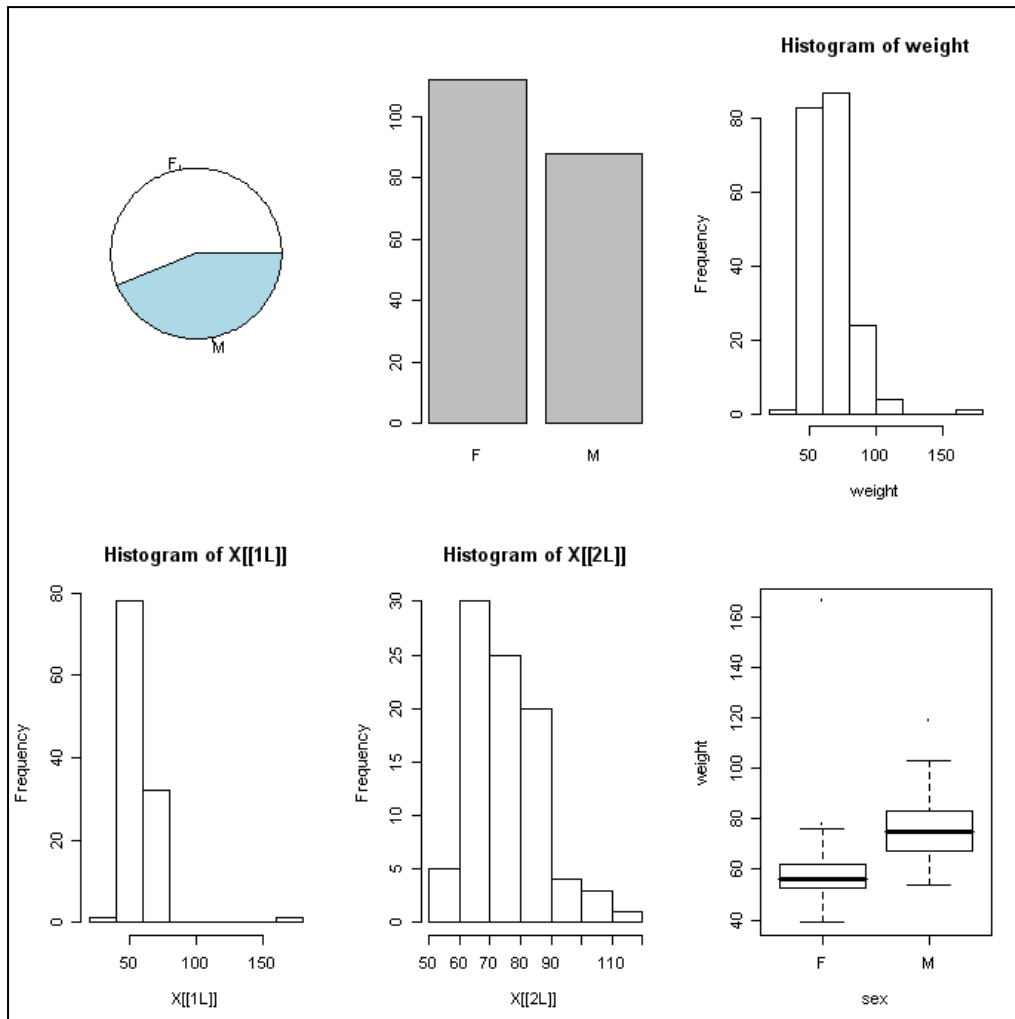
```

library(car)
data(Davis)
names(Davis)
head(Davis)
  
```

```

with(Davis, tapply(height, sex, summary))
summary(Davis$height)
with(Davis, boxplot(height~sex, pch=19,
  col=rainbow(2), main="Height vs Sex"))
  
```

4. INTRODUCTION TO R - EDA - BIVARIATE: NUMERIC VS FACTOR



```
par(mfrow=c(2,3))  
attach(Davis)  
pie( table( sex ))  
barplot( table(sex) )  
hist( weight )
```

```
tapply( weight, sex, hist )# Not nice  
plot( weight ~ sex ) # Boxplot is  
default plot
```


5. EDA – BIVARIATE: 2 NUMERICS Y VS X

5.1 Numeric statistics to assess linear relationship between Y and X

Covariance, $COV(y,x)=COV(x,y)$, defined as $E(YX) - E(X)E(Y)$

- Disadvantage: Depends on units, so not direct interpretation

Pearson's coefficient of correlation, suitable for assessment in normal data

$$\rho(X,Y) = \frac{Cov(X,Y)}{\sigma_X \sigma_Y} \quad \text{and} \quad \sigma_X = \sqrt{Var(X)} \quad \sigma_Y = \sqrt{Var(Y)}$$

- Advantage: Adimensional, no affected by units
 - $\rho(X, Y)$ **range is** $[-1,1]$.
 - $\rho(X, Y) > 0$ means positive relationship X and Y.
 - $\rho(X, Y) < 0$ means negative relationship X and Y,.
 - $\rho(X, Y) = 0$ indicates uncorrelated variables, not equivalent to independence.
 - If $Y = aX + b$ then $|\rho(X, Y)| = 1$.
- **Spearman's coefficient of correlation**, is a nonparametric measure of statistical dependence.

EDA - BIVARIATE: 2 NUMERICS Y VS X

In R, use `var(Davis[,2:3])` or try with Census Data `data("CPS1985")` in library `AER`.

```

> library(AER)
> data("CPS1985")
> df<-CPS1985
> ls()
[1] "CPS1985" "df"
> dim(df) # dimensions: rows and columns
[1] 534 11
> summary(df)

```

wage	education	experience	age	ethnicity	region	gender	occupation
Min. : 1.000	Min. : 2.00	Min. : 0.00	Min. :18.00	cauc :440	south:156	male :289	worker :156
1st Qu.: 5.250	1st Qu.:12.00	1st Qu.: 8.00	1st Qu.:28.00	hispanic: 27	other:378	female:245	technical :105
Median : 7.780	Median :12.00	Median :15.00	Median :35.00	other : 67			services : 83
Mean : 9.024	Mean :13.02	Mean :17.82	Mean :36.83				office : 97
3rd Qu.:11.250	3rd Qu.:15.00	3rd Qu.:26.00	3rd Qu.:44.00				sales : 38
Max. :44.500	Max. :18.00	Max. :55.00	Max. :64.00				

sector	union	married
manufacturing: 99	no :438	no :184
construction : 24	yes: 96	yes:350
other :411		

```

> attach(df)
> # Bivariate analysis: 2 numeric variables
> plot(education,wage,col=as.numeric(ethnicity)+1,
      main="Wage(Y) vs Education (X) | Race",pch=19)
> legend("topleft",legend=levels(ethnicity),col=2:4,
      pch=19)
> cor(wage,education,method="spearman")
[1] 0.3813425
> cor(wage,education,method="pearson") # The one defined in R
[1] 0.3819221

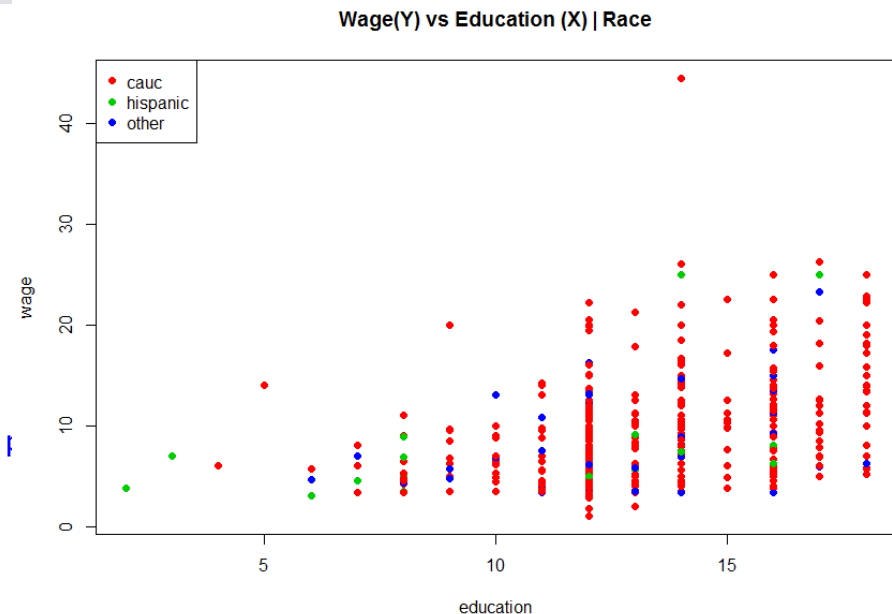
```

Nicer option: `scatterplot`, try in lab session

```

> library(car)
> scatterplot(wage~education|ethnicity,main="Wage(Y) vs Education (X) | Race",smooth=FALSE)

```



6. EDA - BIVARIATE: 2 FACTORS, A AND B

6.1 Numeric statistics to assess linear relationship A and B

Non-existent. Analysis of Contingency Tables and classical inference test to assess Independence of both factors using Chi-Squared Test: `chisq.test()` in R, arguments a contingency table.

```
> ta<-table(ethnicity,sector)
> ta
```

	sector		
ethnicity	manufacturing	construction	other
cauc	81	21	338
hispanic	4	0	23
other	14	3	50

```
> round(prop.table(ta,2),2)
```

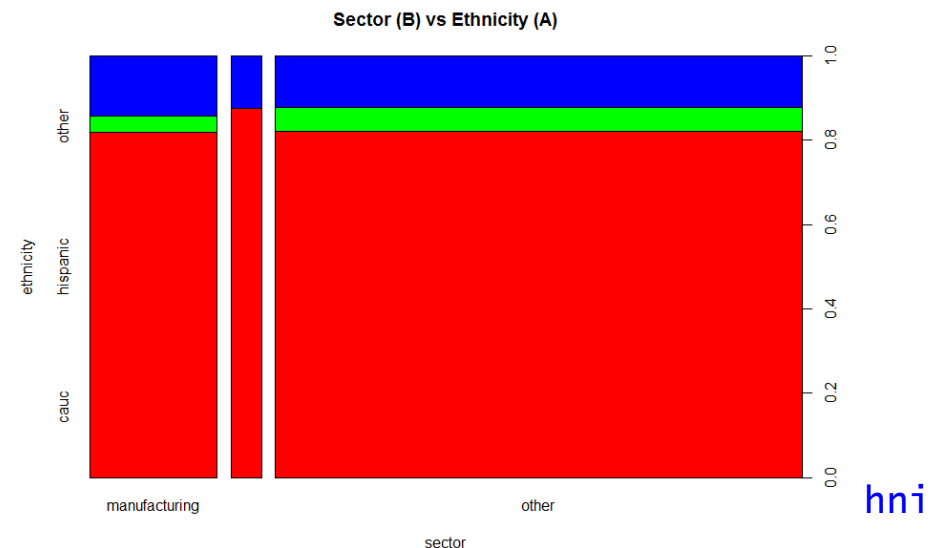
	sector		
ethnicity	manufacturing	construction	other
cauc	0.82	0.88	0.82
hispanic	0.04	0.00	0.06
other	0.14	0.12	0.12

```
> plot(ethnicity~sector,main="Sector (B) vs Et
city (A)",col=rainbow(3))
> chisq.test(ta)
```

```

Pearson's Chi-squared test data: ta
X-squared = 1.9819, df = 4, p-value = 0.7391
Warning message:In chisq.test(ta) : Chi-squared approximation may be incorrect

```



EDA - BIVARIATE: 2 FACTORS, A AND B

Graphic display (default in R): mosaic plot

More than 2 dimensions: use `xtabs()` command in R

```
> xtabs(~gender+ethnicity+sector)
, , sector = manufacturing
```

	ethnicity		
gender	cauc	hispanic	other
male	48	2	10
female	33	2	4

```
, , sector = construction
```

	ethnicity		
gender	cauc	hispanic	other
male	19	0	3
female	2	0	0

```
, , sector = other
```

	ethnicity		
gender	cauc	hispanic	other
male	169	12	26
female	169	11	24

```
> ta<-xtabs(~gender+ethnicity+sector)
> chisq.test(ta)
```

Chi-squared test for given probabilities

```
data: ta
X-squared = 1573.753, df = 17, p-value < 2.2e-16
```