Statistics in R – PART 1

STATISTICAL DATA ANALYSIS

1

LECTURE PLANNING

Lesson	Week	Date	TOPICS	Teacher
1	35	1/Sep	Introduction to the course	MLC
			Descriptive statistics – Part I	
2	36	8/sep	Descriptive statistics – Part II	MLC
3	37	15/Sep	Probability distributions	MLC
4	38	22/Sep	Hypothesis testing (one sample)	VBV
5	39	29/Sep	Hypothesis testing (two samples)	VBV
6	40	6/Oct	ANOVA one-way	VBV
7	41	13/Oct	R class (Introduction to R and descriptive statistics) **Point-giving activity (in class)**	MLC+VBV
-	42	20/Oct	NO CLASS (Autum holidays)	
8	43	27/Oct	R class (hypothesis testing + ANOVA)	MLC
9	44	3/Nov	ANOVA two-way	VBV
-	45	10/Nov	NO CLASS	
10	46	17/Nov	Regression analysis	VBV
11	47	24/Nov	Notions of experimental design and questions **Point-giving activity (in class)**	VBV+MLC
12	48	1/Dec	Multiple regression	MLC

Descriptive statistics

Inferential statistics

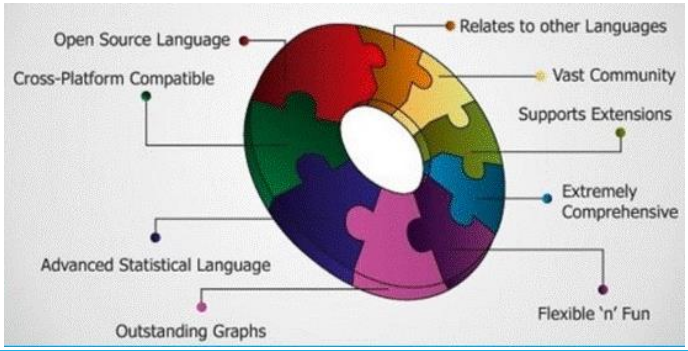
2

Content

1) What is R and what is R Studio?
2) Installing R and R studio
3) Support materials
4) R components and layout
5) Opening the data in R
6) Descriptive statistics in R: summary functions and basic plots
7) Basic operations in R
8) Types of variables in R
9) Inferential statistics in R: Hypothesis testing + ANOVA

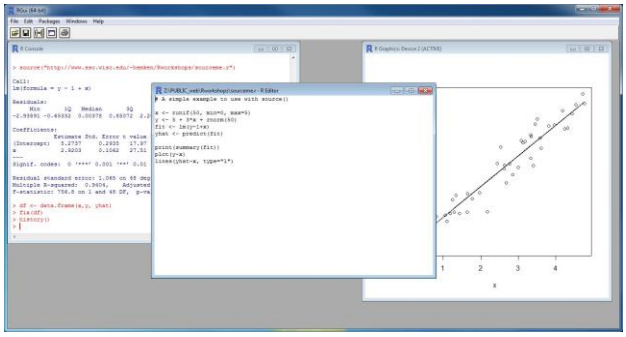
What is R?

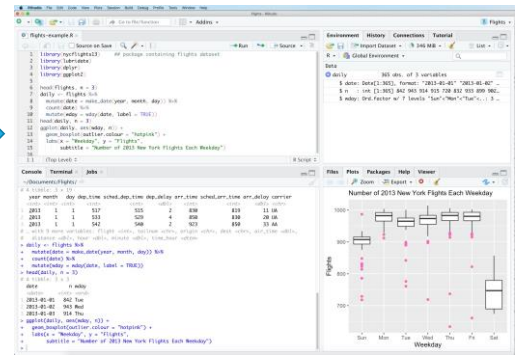
- **R** is an open-source software widely used among statisticians and data miners for conducting statistical and data analysis.
- R is highly extensible through the use of user-submitted **packages** for specific functions or specific areas of study.



What is RStudio?

- The R interface is not considered very user-friendly
- **RStudio** is an Integrated Development Environment (IDE) for R, which is much more user-friendly and organized





Installing R and RStudio

- Three main steps:

1

For those using Windows:
Figure out whether your computer is running the 32-bit or 64-bit version

For Mac and Linux users
Jump this step and perform STEPS 2 and 3 based on your operational system

2

Install R

Go to:
https://mirrors.dotsrc.org/cran/

3

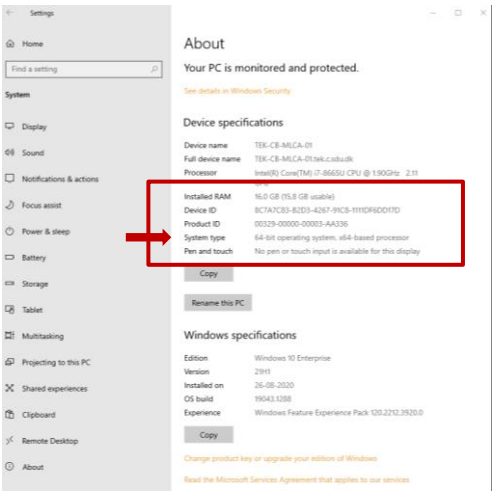
Install Rstudio Desktop

Go to:
https://www.rstudio.com/products/rstudio/download/#download

Installing R and RStudio

In Windows 10:

Go to ***Settings*** → ***System*** → ***About***



1

For those using Windows:
Figure out whether your computer is running the 32-bit or 64-bit version

For Mac and Linux users
Jump this step and perform STEPS 2 and 3 based on your operational system

Installing R and RStudio

- Three main steps:

1

For those using Windows:
Figure out whether your computer is running the 32-bit or 64-bit version

For Mac and Linux users
Jump this step and perform STEPS 2 and 3 based on your operational system

2

Install R

Go to:
https://mirrors.dotsrc.org/cran/

3

Install Rstudio Desktop

Go to:
https://www.rstudio.com/products/rstudio/download/#download

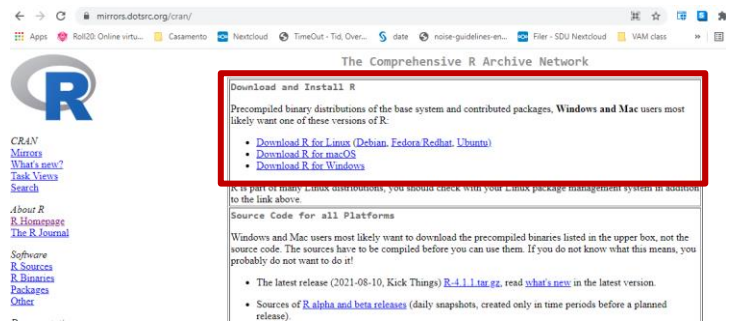
Installing R and RStudio

1- Select the version of R that adapts to your operational system

2

Install R

Go to:
https://mirrors.dotsrc.org/cran/



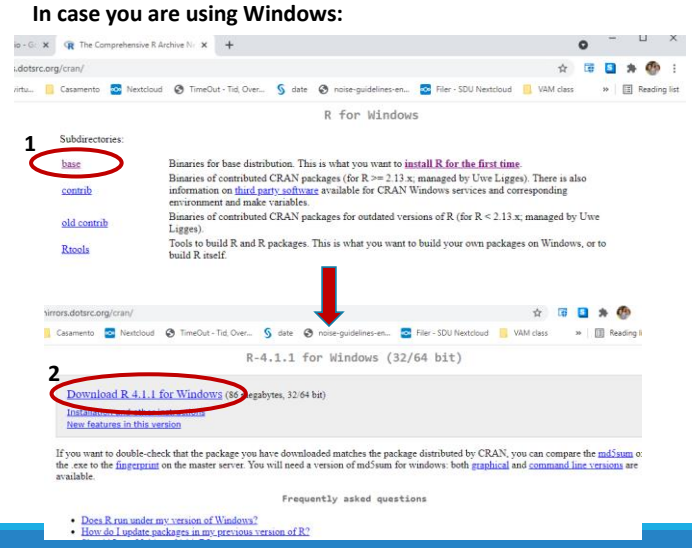
Installing R and RStudio

In case you are using Windows:

2

Install R

Go to:
https://mirrors.dotsrc.org/cran/



Installing R and RStudio

- Three main steps:

1

For those using Windows:
Figure out whether your computer is running the 32-bit or 64-bit version

For Mac and Linux users
Jump this step and perform STEPS 2 and 3 based on your operational system

2

Install R

Go to:
https://mirrors.dotsrc.org/cran/

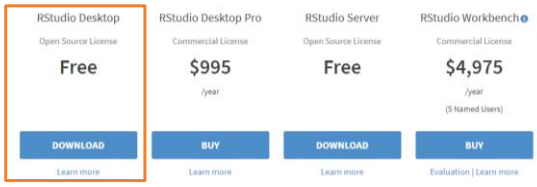
3

Install Rstudio Desktop

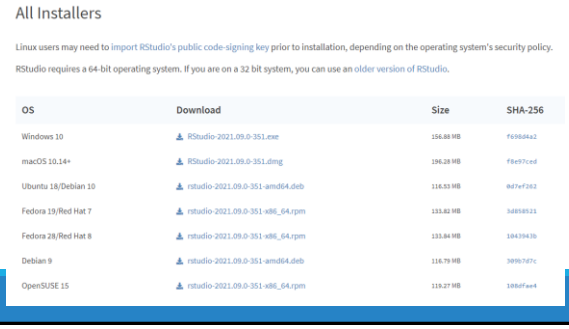
Go to:
https://www.rstudio.com/products/rstudio/download/#download

Installing R and RStudio

1- Select the free version



2 - Select the version appropriate to your machine:



3

Install Rstudio Desktop

Go to:
https://www.rstudio.com/products/rstudio/download/#download

Support material

Books:

1. A Handbook of Statistical Analyses Using R / Brian S. Everitt, Torsten Hothorn , ISBN 1420079336
2. Learning Statistics with R / Danielle Navarro
Can be found at: https://learningstatisticswithr.com/

Youtube channels:

1. MarinStatsLectures-R Programming & Statistics
https://www.youtube.com/channel/UCaNIxVagLhqupvUiDK01Mgg
Observation: I recommend starting with the series: Getting Started with R-Series
2. Statistics Globe
https://www.youtube.com/channel/UCyHEww8_SCdxZvEnkCfi55w

Websites:

1. Cookbook for R: http://www.cookbook-r.com/
2. R-bloggers: https://www.r-bloggers.com/
3. Stack overflow: https://stackoverflow.com/

13

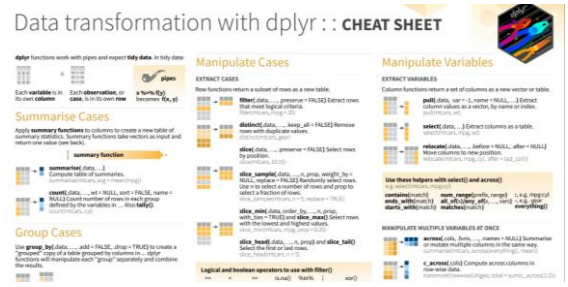
13

Support material

R cheatsheets

Cheatsheets summarize information in a graphic way. They usually provide good content especially regarding specific packages.

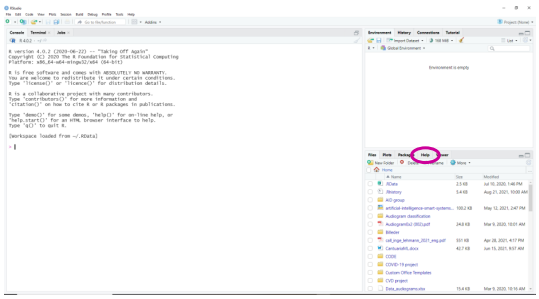
Available at:
https://www.rstudio.com/resources/cheatsheets/



Courses

1. **Learn R** with **Codecademy** (focus on statistical operations and how to use some very relevant packages, i.e. ggplot2 and dplyr).
Available at: https://www.codecademy.com/learn/learn-r
2. **Introduction to R** with **Datacamp** (focus on basic functions and R language structure). Availble at:
https://www.datacamp.com/courses/free-introduction-to-r

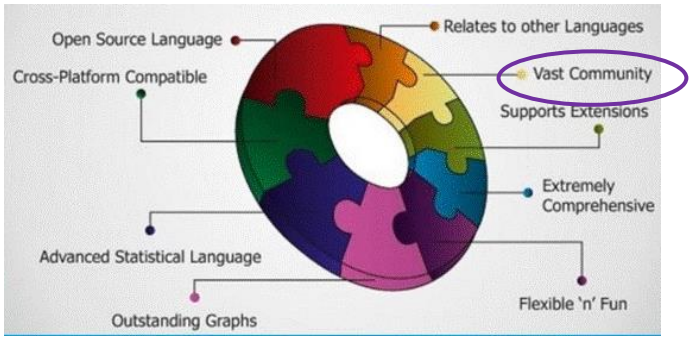
HELP tab in RStudio



14

14

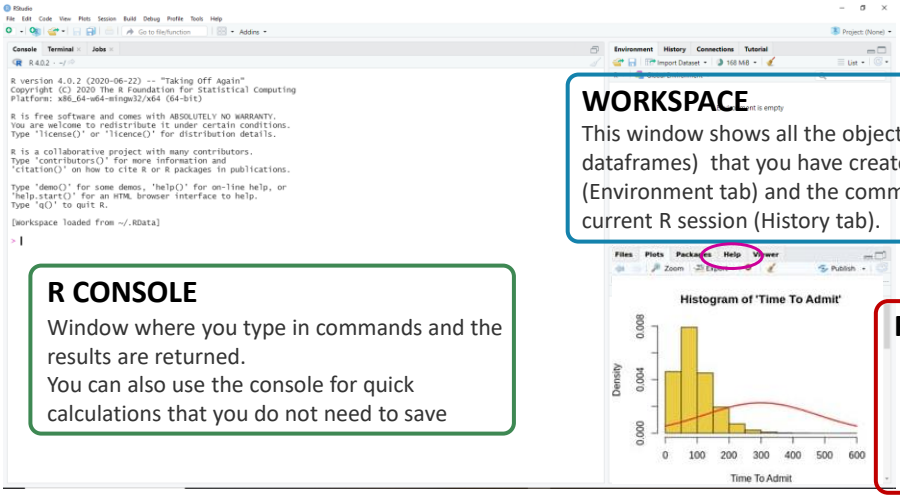
Support material



Content

1) What is R and what is R Studio?
2) Installing R and R studio
3) Support materials
4) R components and layout
5) Opening the data in R
6) Descriptive statistics in R: summary functions and basic plots
7) Basic operations in R
8) Types of variables in R
9) Inferential statistics in R: Hypothesis testing + ANOVA

R components and layout



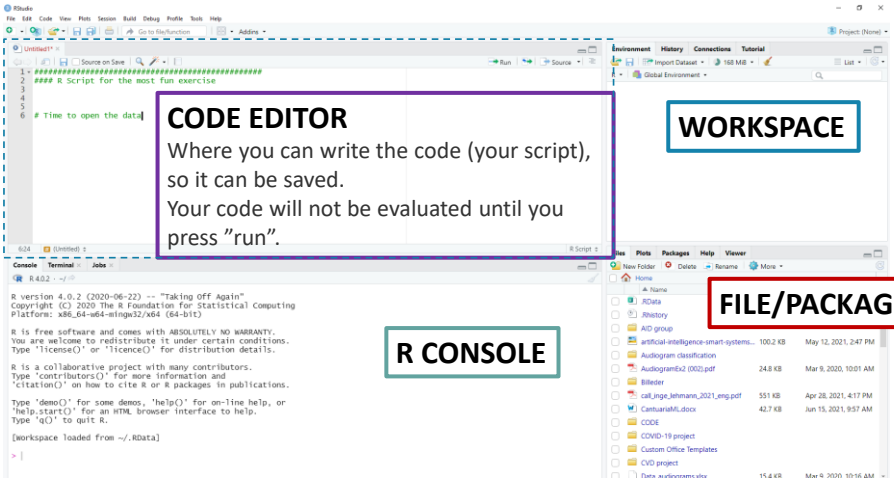
WORKSPACE
This window shows all the objects (like vectors, matrices, and dataframes) that you have created in the current R session (Environment tab) and the commands you have used in the current R session (History tab).

R CONSOLE
Window where you type in commands and the results are returned.
You can also use the console for quick calculations that you do not need to save

FILE/PACKAGES/PLOT
This window is primarily used for displaying plots (graphs), for inspecting packages, and for using the help system.

R components and layout

To open the code editor: **File -> New file -> R script**



CODE EDITOR
Where you can write the code (your script), so it can be saved.
Your code will not be evaluated until you press "run".

WORKSPACE

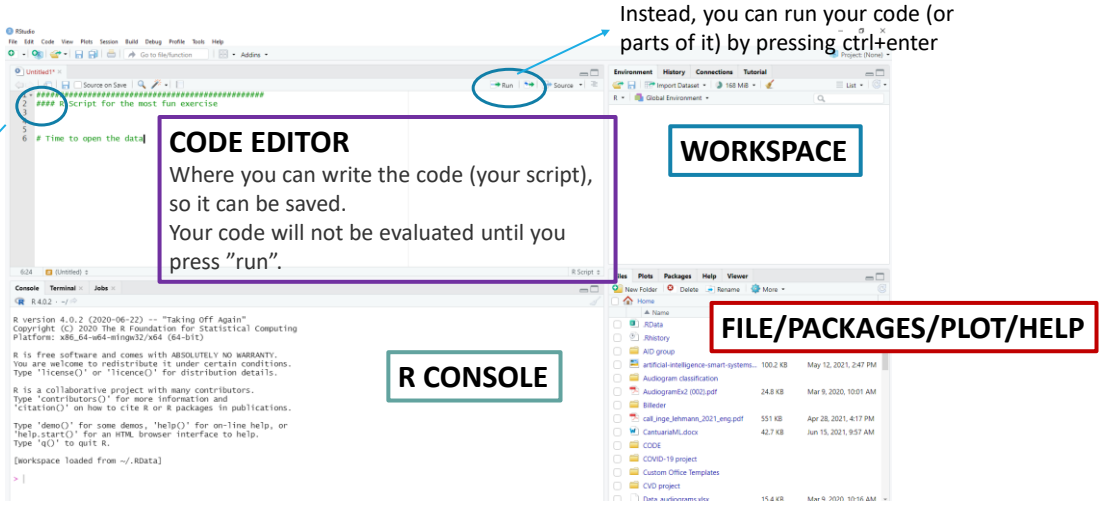
R CONSOLE

FILE/PACKAGES/PLOT/HELP

R script

R scripts are just text files with the ".R" extension, which contain the code you developed. Remember to name and save your script by going to **File -> Save as**

Press here to run the code.
Instead, you can run your code (or parts of it) by pressing ctrl+enter



is used when you want to add comments on your code!

Content

1) What is R and what is R Studio?
2) Installing R and R studio
3) Support materials
4) R components and layout
5) Opening the data in R
6) Descriptive statistics in R: summary functions and basic plots
7) Basic operations in R
8) Types of variables in R
9) Inferential statistics in R: Hypothesis testing + ANOVA

Work directory

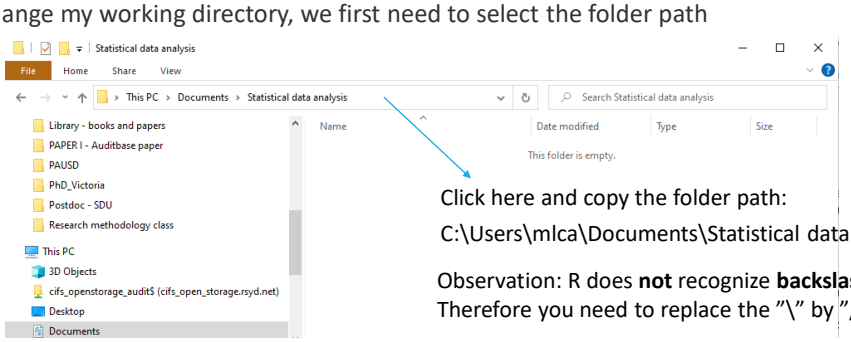
- The **working directory** is just a file path on your computer that sets the default location of any files you read into R, or save out of R.
- If you ask R to import a dataset or save a dataframe as a text or csv file, it will assume that the file is inside of your working directory.
- You can only have one working directory active at any given time. This is called the **current** working directory

```
## R Script for learning R
## Statistical data analysis


## To see where the current working directory is:
getwd()
#"C:/Users/mlca/Documents/Teaching"
```


Work directory

- To change my working directory, we first need to select the folder path



Click here and copy the folder path:

C:\Users\mlca\Documents\Statistical data analysis

Observation: R does **not** recognize **backslash **.
Therefore you need to replace the "\" by "/"

```
## ESTABLISHING WORKING DIRECTORY:
setwd("C:/Users/mlca/Documents/Statistical data analysis")
```

Having your data in R

Two options

- **Creating vectors + Creating a dataframe**
- **Importing a dataset**
  - Using clickable actions in the environment tab
  - Using the appropriate lines of code

Having your data in R

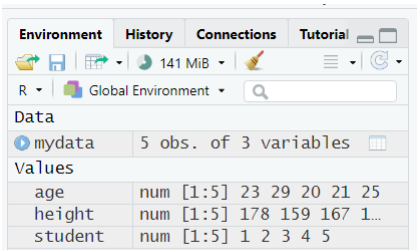
Creating vectors + Creating a dataframe

- Let's say we want to create a data frame in R where we have the data on the age and height of five students. Based on the data I collected, I know that the students are 23, 29, 20, 21, 25 years old, and their height was 178, 159, 167, 186, 184 cm.
- Let's create a data frame in R with the data I just collected?

```
## OPTION 1 - Creating vectors and dataframes

# Creating vectors:
student <- c(1, 2, 3, 4, 5)
age <- c(23, 29, 20, 21, 25)
height <- c(178, 159, 167, 186, 184)

#Creating a dataframe
mydata <- data.frame(student, age, height)
```



The Environment tab will show the new objects we just created. Here we can also see e.g. the number of observations and rows in data objects.

Having your data in R

Creating vectors
+
Creating a dataframe

- Let's say we want to create a data frame in R where we have the data on the age and height of five students. Based on the data I collected, I know that the students are 23, 29, 20, 21, 25 years old, and their height was 178, 159, 167, 186, 184 cm.
- Let's create a data frame in R with the data I just collected?

```
## OPTION 1 - Creating vectors and dataframes

# Creating vectors:
student <- c(1, 2, 3, 4, 5)
age <- c(23, 29, 20, 21, 25)
height <- c(178, 159, 167, 186, 184)

#Creating a dataframe
mydata <- data.frame(student, age, height)
```

student	age	height
1	23	178
2	29	159
3	20	167
4	21	186
5	25	184

Having your data in R

Two options

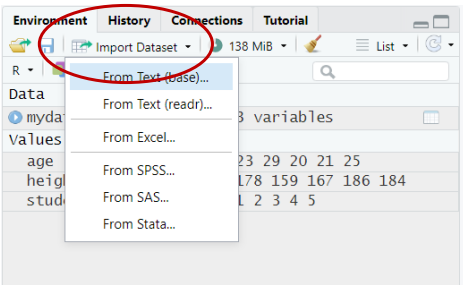
- **Creating vectors + Creating a dataframe**
- **Importing a dataset**
  - Using clickable actions in the environment tab
  - Using the appropriate lines of code

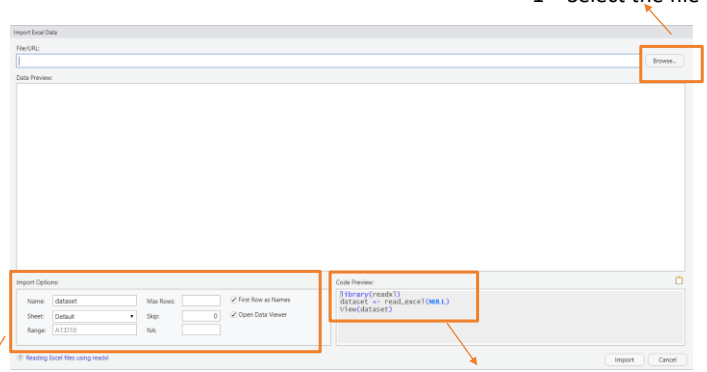
Having your data in R

Using clickable actions in the environment tab

- These clickable actions are not so used for those that are more fluent in R. However, it can be a good alternative in some specific cases, for example, when opening data from an Excel (".xlsx") file.

1 – Select the file





2 – Here you can also rename the file and do other changes on it

3 – You can also copy the code and use it for the next times

27

Having your data in R

Two options

- **Creating vectors + Creating a dataframe**
- **Importing a dataset**
  - Using clickable actions in the environment tab
  - Using the appropriate lines of code

28

28

Having your data in R

Using the appropriate lines of code

- We can also use lines of code to import our dataset to R.
- These are usually much more flexible and reproducible.

As an example, let's import the csv data "flavor.csv" (available in ItsLearning) which is already located in my working directory:

data_flavor <- read.csv("flavor.csv", sep=";")

Important notes:

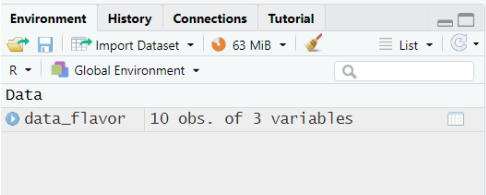
1) Sometimes, you do not need to state what is the separator. It will depend on the file and how your Excel and operational system is configurated.
2) If the file is not in your working directory, you also need to specify the file path. For example:
   data_flavor <- read.csv("C:/Users/mlca/Documents/Statistical data analysis/flavor.csv", sep=";")

29

Having your data in R

Using the appropriate lines of code

After reading the csv file, it should be seen in your environment tab



To view a dataframe, we can either click on its name, or use the function *View()*

Other important functions are *head()* and *tail()*

head() shows the first few rows and *tail()* shows the last few rows

```
> head(data_flavor)
  Flavor1 Flavor2 Flavor3
1      12      14       7
2       8      17      19
3       6      19      15
4      16      12      14
5      12      20      10
6      14      15      16
```

```
> tail(data_flavor)
   Flavor1 Flavor2 Flavor3
5       12      20      10
6       14      15      16
7       10      18      18
8       18      11      11
9        4      13      14
10      11      16      11
```

30

Saving your workspace

- Saving your workspace at the end of the session can be a good idea to not lose the R objects (e.g. vectors, dataframes, model results) you have created in a specific session.
- A way to do this is to use the save.image() command:

```
##To save the workspace(environment) when we are done for the day.
#This will be saved in the WorkingDirectory
save.image("StatClass.Rdata")

##To open the workspace again, the next time you are working with the data
#remember to check if you are using the right WD:
load("StatClass.Rdata")
```

- The load() command is used to open the workspace again, and see the objects you have previously created.

31

Content

1) What is R and what is R Studio?
2) Installing R and R studio
3) Support materials
4) R components and layout
5) Opening the data in R
6) Descriptive statistics in R: summary functions and basic plots
7) Basic operations in R
8) Types of variables in R
9) Inferential statistics in R: Hypothesis testing + ANOVA

32

32

Descriptive statistics

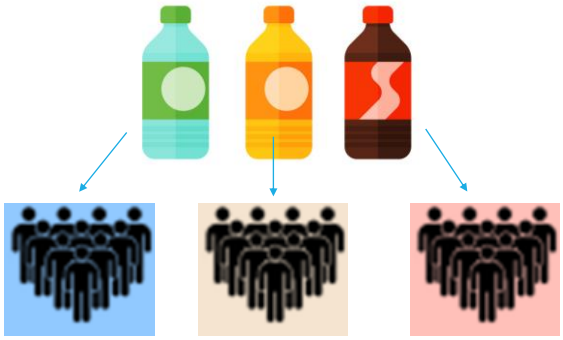
Summary statistics functions for continuous data:

Function	Example	Result
sum(x), product(x)	sum(1:10)	55
min(x), max(x)	min(1:10)	1
mean(x), median(x)	mean(1:10)	5.5
sd(x), var(x), range(x)	sd(1:10)	3.03
summary(x)	summary(1:10)	Min = 1.00. 1st Qu. = 3.25, Median = 5.50, Mean = 5.50, 3rd Qu. = 7.75, Max = 10.0

33

Example: Comparing beverages' flavor

A marketing research firm tests the effectiveness of three new flavorings for a leading beverage using a sample of 30 people, divided randomly into three groups of 10 people each. Group 1 tastes flavor 1, group 2 tastes flavor 2 and group 3 tastes flavor 3. Each person is then given a questionnaire that evaluates how enjoyable the beverage was. The scores are as in the data "flavor.csv".



Scores obtained with each of the groups

Flavor1	Flavor2	Flavor3
12	13	7
8	17	19
6	19	15
16	11	14
12	20	10
14	15	16
10	18	18
18	9	11
4	12	14
11	16	11

34

Summary() in R for a dataframe

- The summary() function in R is a generic function used to produce result summaries of dataframes, specific variables, and model fitting functions.
- When used with dataframes, it will show us the results for minimum and maximum values, 1st and 3rd quartiles, median and mean for all variables of the dataset

```
> summary(data_flavor)
    Flavor1         Flavor2         Flavor3
 Min.   : 4.0   Min.   :11.00   Min.   : 7.00
 1st Qu.: 8.5   1st Qu.:13.25   1st Qu.:11.00
 Median :11.5   Median :15.50   Median :14.00
 Mean   :11.1   Mean   :15.50   Mean   :13.50
 3rd Qu.:13.5   3rd Qu.:17.75   3rd Qu.:15.75
 Max.   :18.0   Max.   :20.00   Max.   :19.00
```

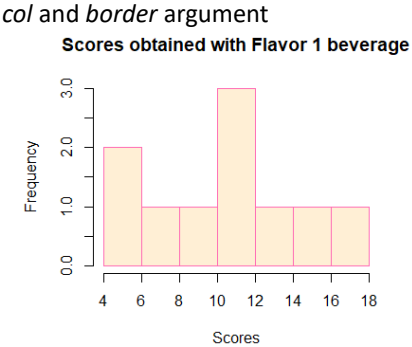
35

Histograms in R

- Histograms are the most common way to plot a vector of numeric data and show how the data is distributed.
- To create a histogram in R, we can use the function ***hist()***. The main argument in *hist()* is x, the vector of numeric data.
- If we want to specify how many histogram bins we want, we can use the *breaks* argument. We can also specify the x limits with the *xlim* argument.
- Color of the border and bars can also be changed with the *col* and *border* argument

```
hist(x = data_flavor$Flavor1,
   main = "Scores obtained with Flavor 1 beverage",
   xlab = "Scores", col="papayawhip", border = "hotpink")
```

The dollar sign ($) in R indicates that we are taking the variable "Flavor1" from the data_flavor dataset

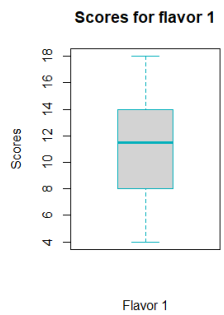


36

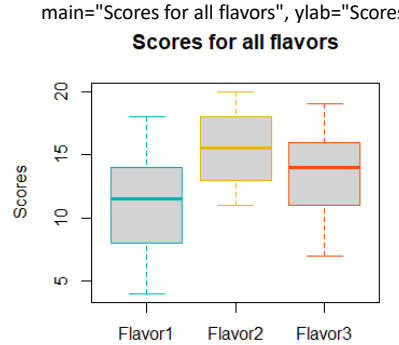
Boxplots in R

- Boxplots can be created for individual variables or for many variables.
- The format is: **boxplot(*x*)**, where *x* is either a specific variable or an entire dataset, in case we want to generate boxplots for all the variables in our data.

```
boxplot(data_flavor$Flavor1, border = "#00AFBB",
        main="Scores for all flavors",
        xlab="Flavors", ylab="Scores")
```




```
boxplot(data_flavor,
        border = c("#00AFBB", "#E7B800", "#FC4E07"),
        main="Scores for all flavors", ylab="Scores")
```

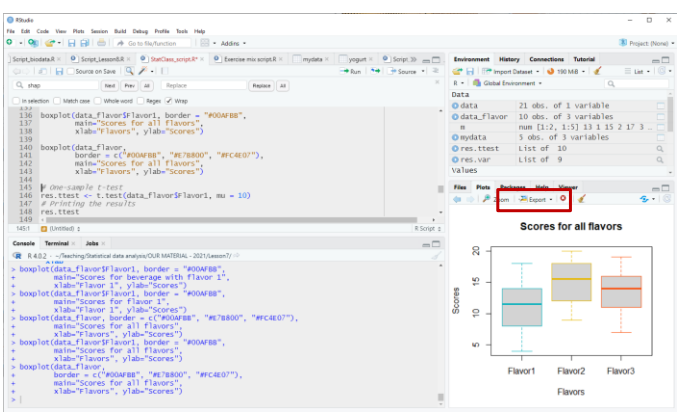


NOTE
We can also use **boxplot(*x*, data=)**, where *x* is a formula and **data=** denotes the data frame providing the data. An example of a **formula** is y~group where a separate boxplot for numeric variable y is generated for each value of a group.

37

Saving a figure you generated in R

After you generated a figure in R, one of the ways to export it is to simply use the **Export** buttom in the Plot viewer.



Here you can change the image format (e.g. jpeg, png, tiff), where you want to save it, and the image size.

38

Questions?

39