

BIOSTATISTICS WITH

SUMMER WORKSHOP
(MITGEST network)

24-27 JULY 2024

Maria Chiara Mimmi, PhD

WORKSHOP SCHEDULE

- 4 days
 - 1. Intro to R and data analysis
 - 2. Statistical inference & hypothesis testing
 - 3. Modeling correlation and regression
 - 4. Intro to Machine Learning
- Each day will include:
 - Frontal class (MORNING)
 - Practical training with R about the topics discussed in the morning. (AFTERNOON)

DAY 1 – LECTURE OUTLINE

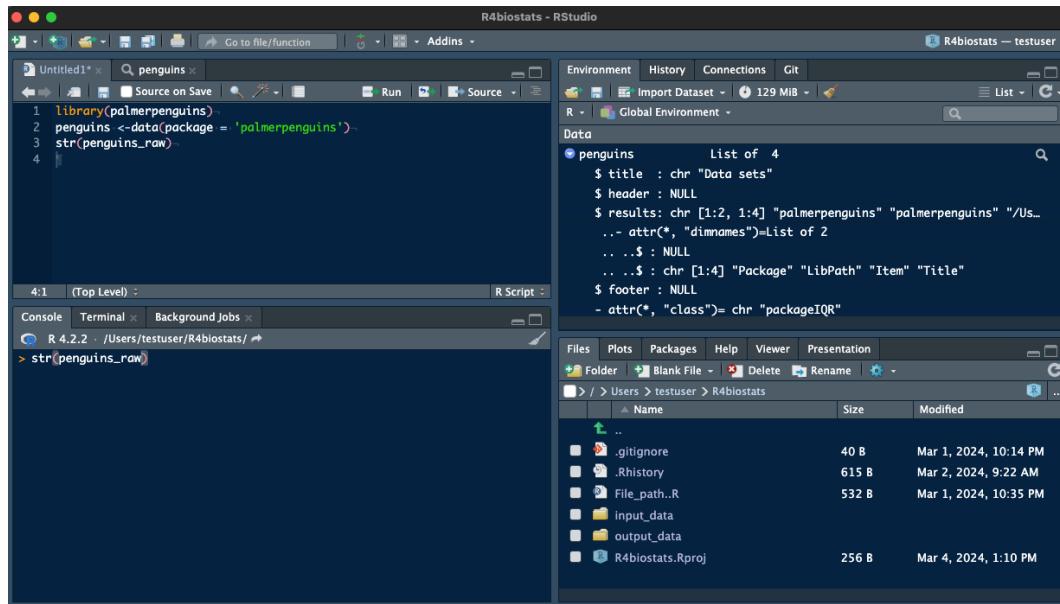
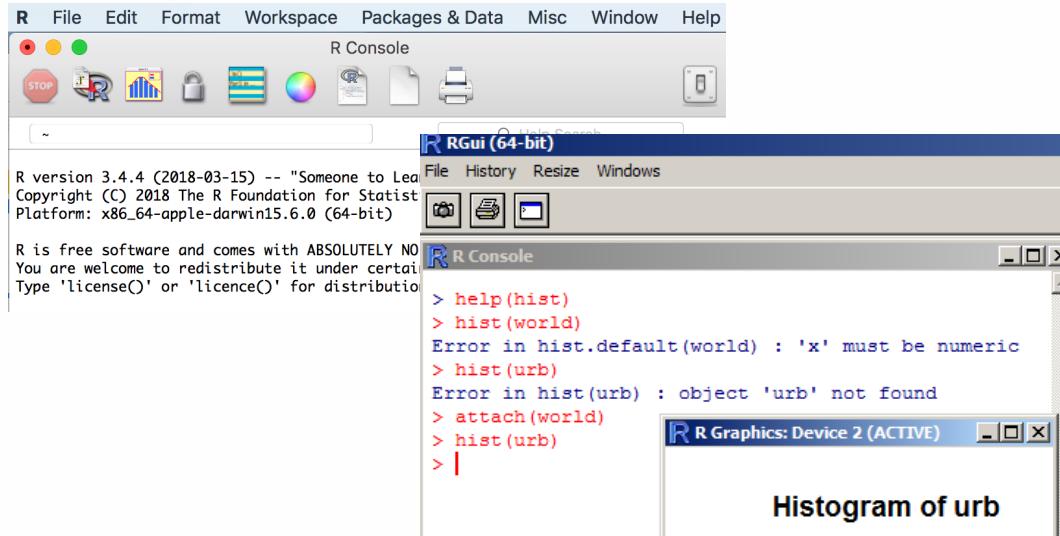
- Introduction to R and R-studio
 - Why R?
 - Principles of reproducible analysis with R + RStudio
- R objects, functions, packages
- Understanding different types of variables
 - Principles of “tidy data”
 - Data cleaning and manipulation
- Descriptive statistics
 - measures of central tendency, measures of variability (or spread), and frequency distribution
- Visual data exploration
 - {ggplot2}
- Foundations of inference

WHY learning and using ?

- R is an analytical tool created over 25 years ago by and for statisticians
- It is **free and open-source**, i.e. built by a large, vibrant community of users (including from life science/epidemiology/healthcare fields), in academia and industry
- Beyond statistical computing, the R ecosystem enables many different tasks: data visualization, academic publishing, web scraping, automated reporting, website building...
- ...*while* it inherently promotes **reproducible research**:
 - It is a “scripted” language (better than “point & click tools”)
 - It is OS agnostic
 - It integrates files’ version control systems (git)
 - It integrates literate programming tools (.Rmd, .qmd) for combining code + comments

Different ways to code with R

1. The native **R GUI** (Graphic User Interface)
2. Your **CLI** (command line interface) – on Mac terminal, On PC shell
3. Practically any **Code Editor** (VS Code, Sublime Text, Jupyter Notebook, etc.)
4. Our favorite: **Rstudio IDE** (Interactive Development Environment)!



How do I install R and RStudio?

1. To install **R** go to <https://cloud.r-project.org/>
 - Download appropriate version (Windows, Linux, macOS)

2. (Then) you can install **Rstudio Desktop** going to <https://posit.co/download/rstudio-desktop/>
 - Download appropriate version (Windows, Linux, macOS)
 - “Desktop” is the free version (there is also a paid pro one)
 - No need to have the latest version, but I recommend **v2022.07.1 or later** (it includes Quarto*)

Check out the RStudio “CHEATSHEET”!

<https://rstudio.github.io/cheatsheets/rstudio-ide.pdf>

We will use RStudio Desktop as our IDE

The screenshot displays the RStudio Desktop interface with four main panes:

- Source**: Shows an R script named "ggplot2.R" with code to create a scatter plot. The code includes `library(ggplot2)`, `mpg_plot <- ggplot(mpg, aes(x = displ, y = hwy)) + geom_point(aes(colour = class))`, and `mpg_plot`.
- Console**: Shows the R command-line interface with the following session:

```
R 4.2.0 · ~/rstudio-user-guide/
> library(ggplot2)
> mpg_plot <- ggplot(mpg, aes(x = displ, y = hwy)) +
+   geom_point(aes(colour = class))
>
> mpg_plot
>
```
- Environments**: Shows the Global Environment pane with one entry: `mpg_plot` of type "List of 9".
- Output**: Shows a scatter plot of highway fuel economy (hwy) versus engine displacement (displ). The plot is color-coded by vehicle class. The legend indicates the following classes and colors:

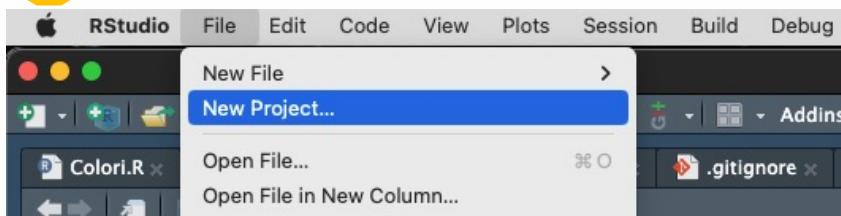
class	Color
2seater	Red
compact	Yellow
midsize	Green
minivan	Cyan
pickup	Blue
subcompact	Purple
suv	Magenta

The 4 primary panes in the Rstudio interface

Source: <https://docs.posit.co/ide/user/ide/guide/ui/ui-panes.html>

How do I create an Rproject in RStudio?

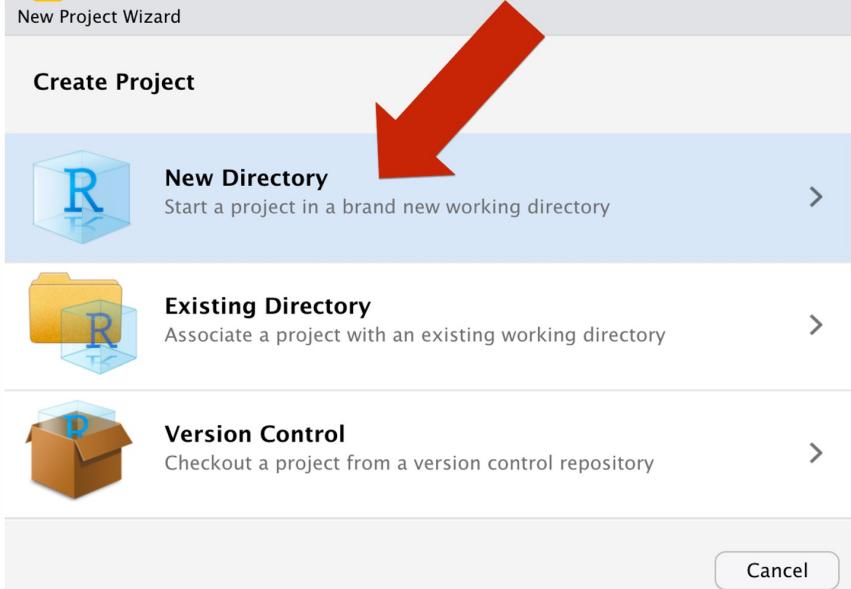
1



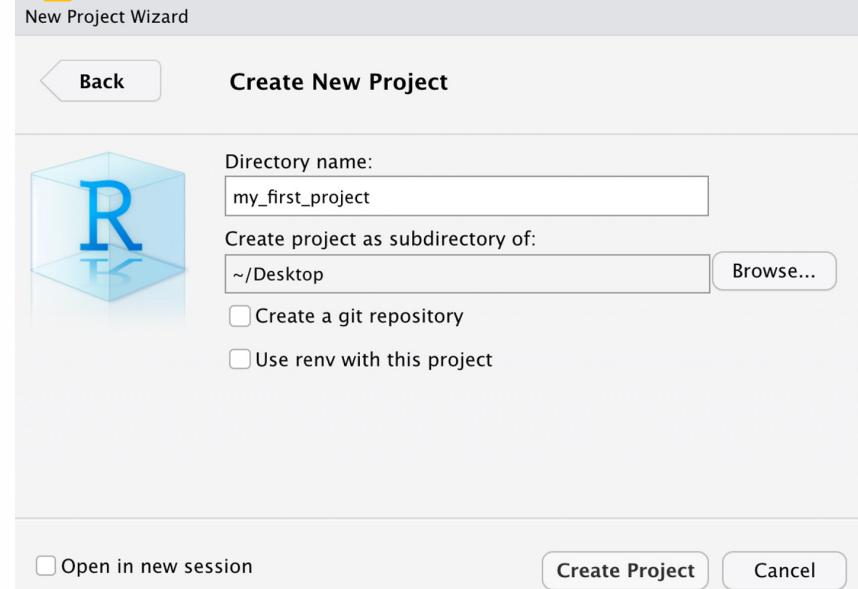
- Easy, just follow RStudio prompts

- Create a new directory for each project
- Select parent folder

2

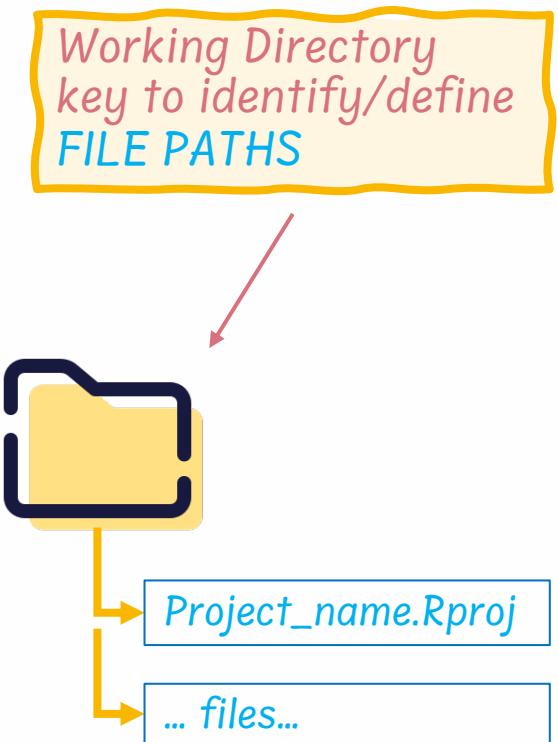
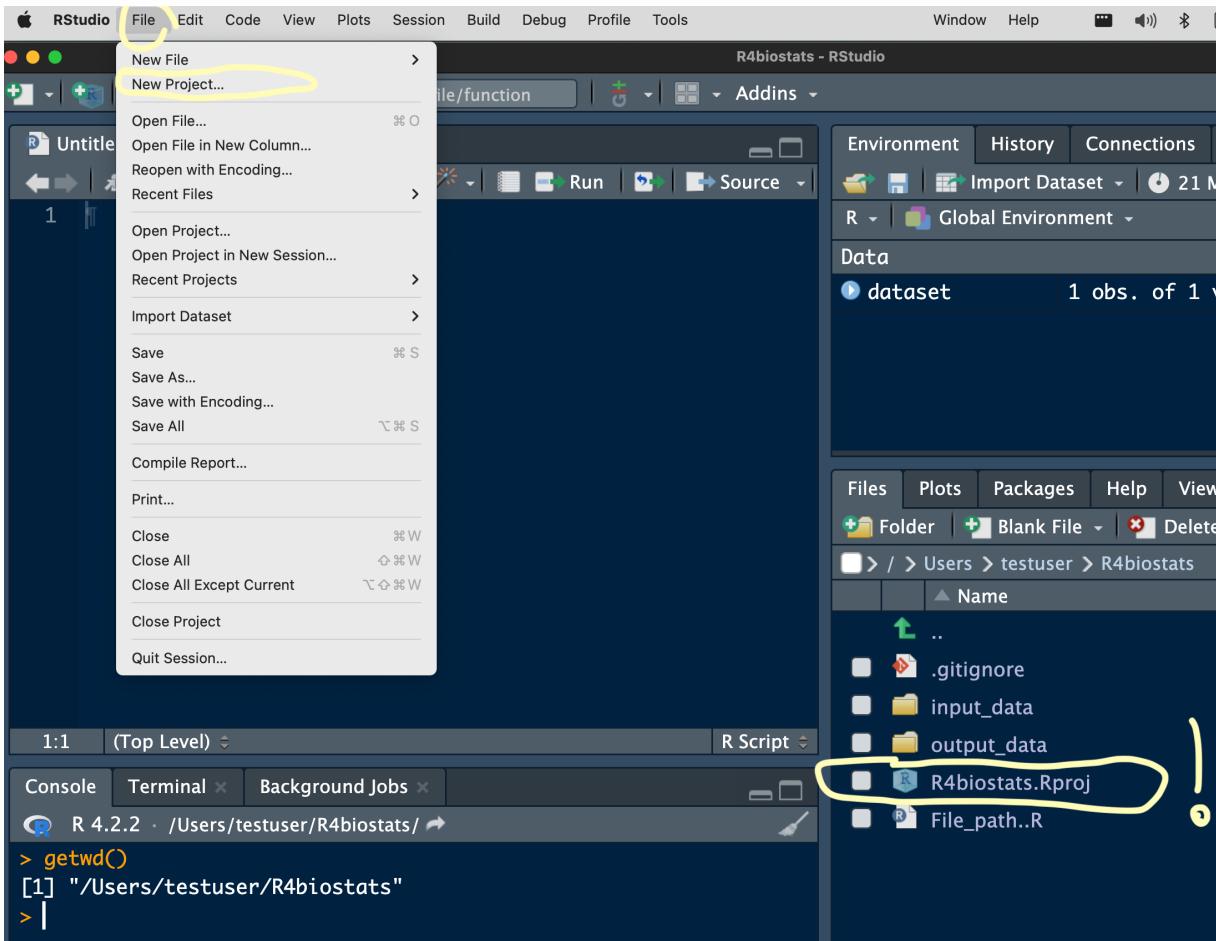


3

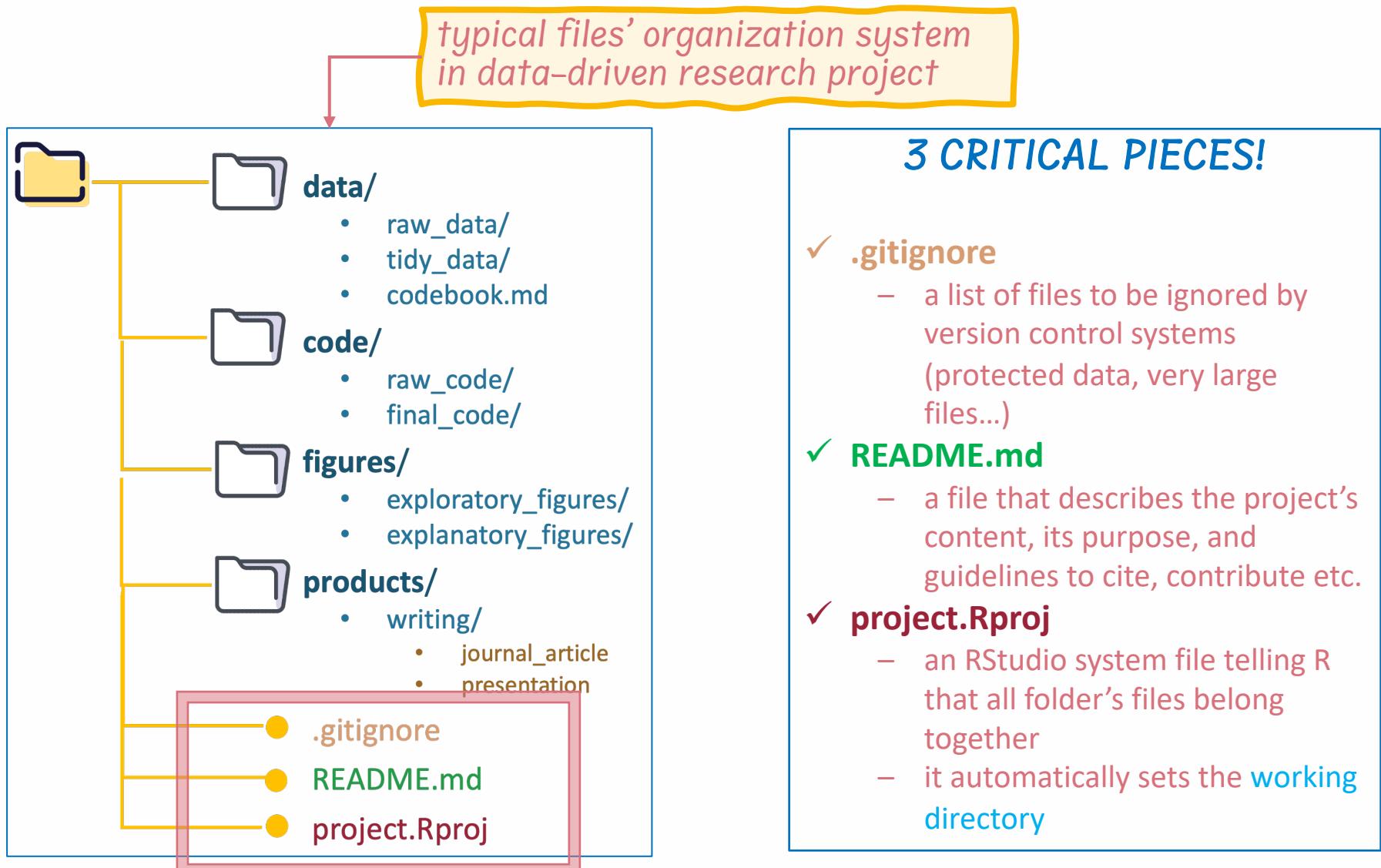


What does the Rproject do?

- Organizes all R code + complimentary files associated to a project
- Set the project's “**working directory**” as the folder containing the “***.Rproj**” file



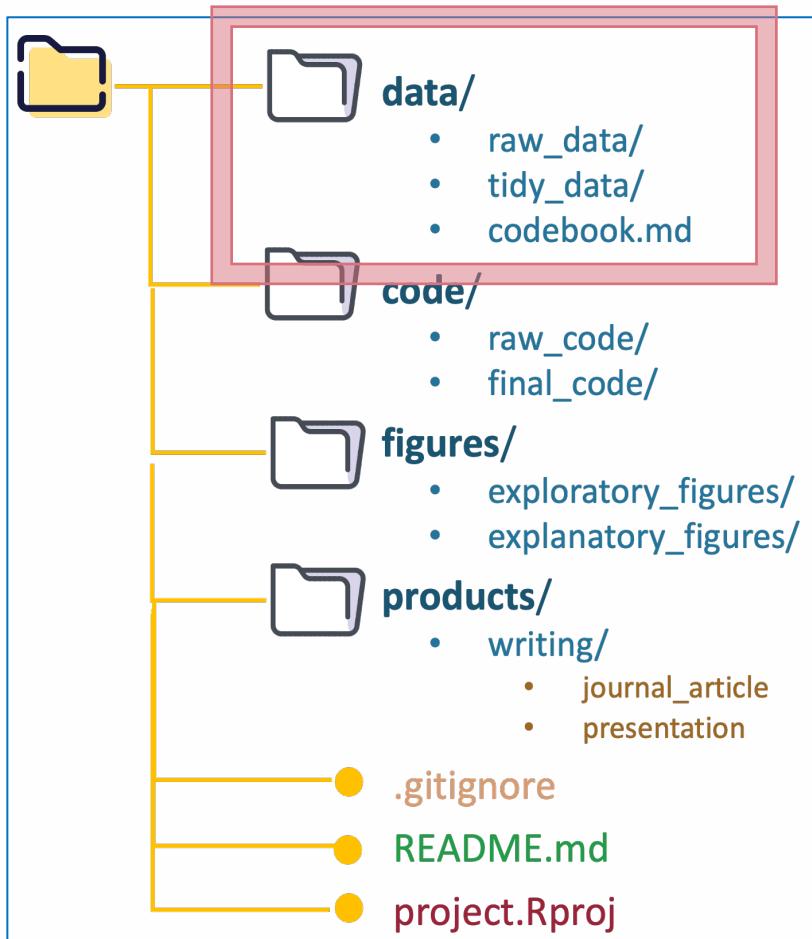
Projects' files organization with Rproject



Organizing data/folder (for reproducible analysis)



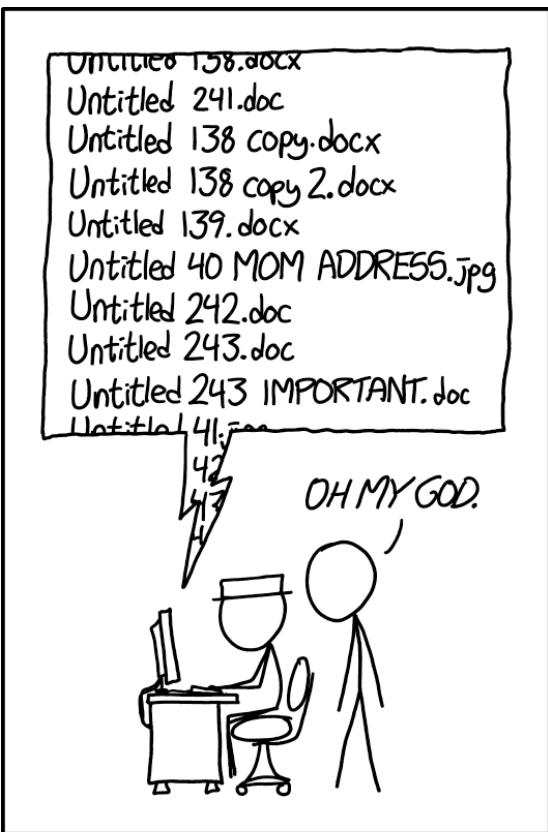
Future-YOU will thank you!



REMEMBER:

- ✓ **raw_data/...**
 - must be saved **BEFORE** being processed in any way
 - ...then treated as **READ ONLY!!**
- ✓ You need a **data-codebook** file
 - with all **metadata** on your variables (name, type, range of values, missing values, etc)
- ✓ **tidy_data/...** or **output_data/...**
 - must be kept separately from raw data
 - ideally in “**tidy**” form

A file naming system is crucial too...



Source: <https://xkcd.com/1459>

*"Pick a filename convention.
Any convention.
Just pick one."*

Jenny Bryan, statistics professor and R developer

Bad naming	Good naming
1 my file.md	01_file_labA.csv
10.my.ten th .file.md	02_file_labB.csv
2 (actual) second?-file.md	YYYY_MM_DD_version.md
Extremely-long LOOOONG toolong-name.csv	YYYY_MM_DD_author_report.md
M@ybe THE-worst! Name 3.md	reportA_v01.txt
	reportA_v02.txt

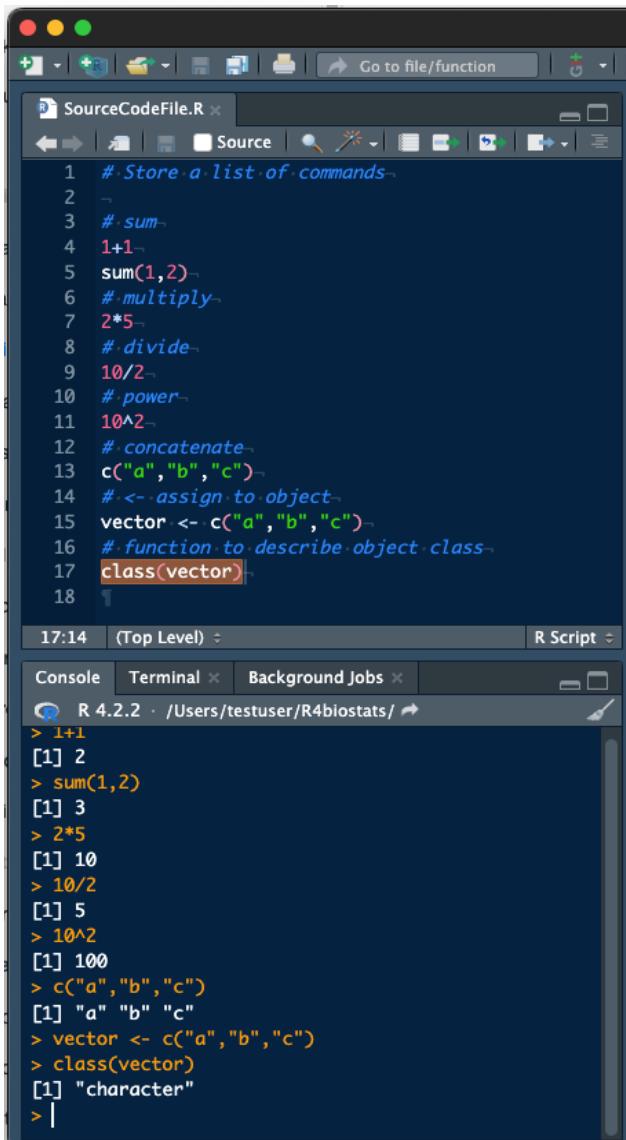


Other techniques for reproducible analysis

1. Analyze data via “scripts” (to store code)—instead of manually or with point-&-click tools leaving no trace of analytical steps
 - eg. with R, Python, Stata, Excel macros
2. Automate repeated tasks
 - “DRY” principle (Don't Repeat Yourself!)
 - Organize procedures into dedicated functions (that clean data, split datasets, create graphs...)
3. Use a version control system (VCS)
 - Git, Github, OSF files
4. Use open source software (where possible)
 - Yay R!
5. Use and create open data (where possible)

How to execute R commands (2 ways)

1



```
SourceCodeFile.R
1 #·Store·a·list·of·commands·
2 ·
3 #·sum·
4 1+1
5 sum(1,2)
6 #·multiply·
7 2*5
8 #·divide·
9 10/2
10 #·power·
11 10^2
12 #·concatenate·
13 c("a","b","c")
14 #·<- assign to object·
15 vector <- c("a","b","c")
16 # function to describe object class
17 class(vector)
18 ·
```

17:14 (Top Level) R Script

```
Console Terminal × Background Jobs ×
R 4.2.2 · /Users/testuser/R4biostats/ ↵
> 1+1
[1] 2
> sum(1,2)
[1] 3
> 2*5
[1] 10
> 10/2
[1] 5
> 10^2
[1] 100
> c("a","b","c")
[1] "a" "b" "c"
> vector <- c("a","b","c")
> class(vector)
[1] "character"
> |
```

entering 3 or more characters
of a command into the
console or a script will trigger
autocomplete!



- 1) In the **Source** pane, you write/save code scripts (***.R** or ***.Rmd, *.qmd**) including:
- R commands
 - *#comments' lines*

2



- 2) In the **Console** pane, you write R commands interactively
- & see the output just below

DAY 1 – LECTURE OUTLINE

- Introduction to R and R-studio
 - Why R?
 - Principles of reproducible analysis with R + RStudio
- R objects, functions, packages
- Understanding different types of variables
 - Principles of “tidy data”
 - Data cleaning and manipulation
- Descriptive statistics
 - measures of central tendency, measures of variability (or spread), and frequency distribution
- Visual data exploration
 - {ggplot2}
- Foundations of inference

R objects

- Everything in R is an **object!** (*hence it's called Object-Oriented-Language*)
- The ↗ («assign») operator gives a name to a value (object) and saves it in your workspace

Object Type	Meaning	Example									
integer	whole numbers	1									
logical	values of true or false	TRUE									
double	floating point non-whole numbers	2.5									
character	character strings	“R is very cool”									
function	code defining a function	func <- plot()									
vector	values (of same type) stored in an object together	intg_vector <- c(1, 2, 5) char_vector <- c("blue", "red", "green")									
matrix	Rectangular data object where every element is of the same type	matrix_data <- cbind(1:3, 4:5)									
data.frame	Rectangular data object that can contain different object types (numeric, character, factor, dates, etc.)	data_frame_data <- dplyr::starwars <table><thead><tr><th>name</th><th>height</th><th>homeworld</th></tr></thead><tbody><tr><td>Luke Skywalker</td><td>172</td><td>Tatooine</td></tr><tr><td>C-3PO</td><td>167</td><td>Tatooine</td></tr></tbody></table>	name	height	homeworld	Luke Skywalker	172	Tatooine	C-3PO	167	Tatooine
name	height	homeworld									
Luke Skywalker	172	Tatooine									
C-3PO	167	Tatooine									
list	Objects which contain elements of different types (numbers, strings, vectors, dataframe or even list) ...	list_data <- list("Red", "Green", c(21,32,11), TRUE, 51.23, 119.1)									

R functions

- A **function** is a set of statements organized together to perform a specific task
- R functions:
 - take 1 or more inputs (*arguments*)
 - contain **statements** inside **{...}** that define what the function does
 - either **produce** an output (*return value/object*) or perform a task (i.e. save a file)
- R functions can be:
 - built-in
 - user defined (and shared by other programmers inside packages)
 - created by you!

DEFINING +
CALLING YOUR OWN function !

```
# Define a function that takes input `temp_F` -----
fun_fahrenheit_2_celsius <- function(temp_F) {
  # operations to be executed inside `{}`
  temp_C <- (temp_F - 32) * 5 / 9
  # returned output
  return(temp_C)
}

# Call the function -----
fun_fahrenheit_2_celsius(75)

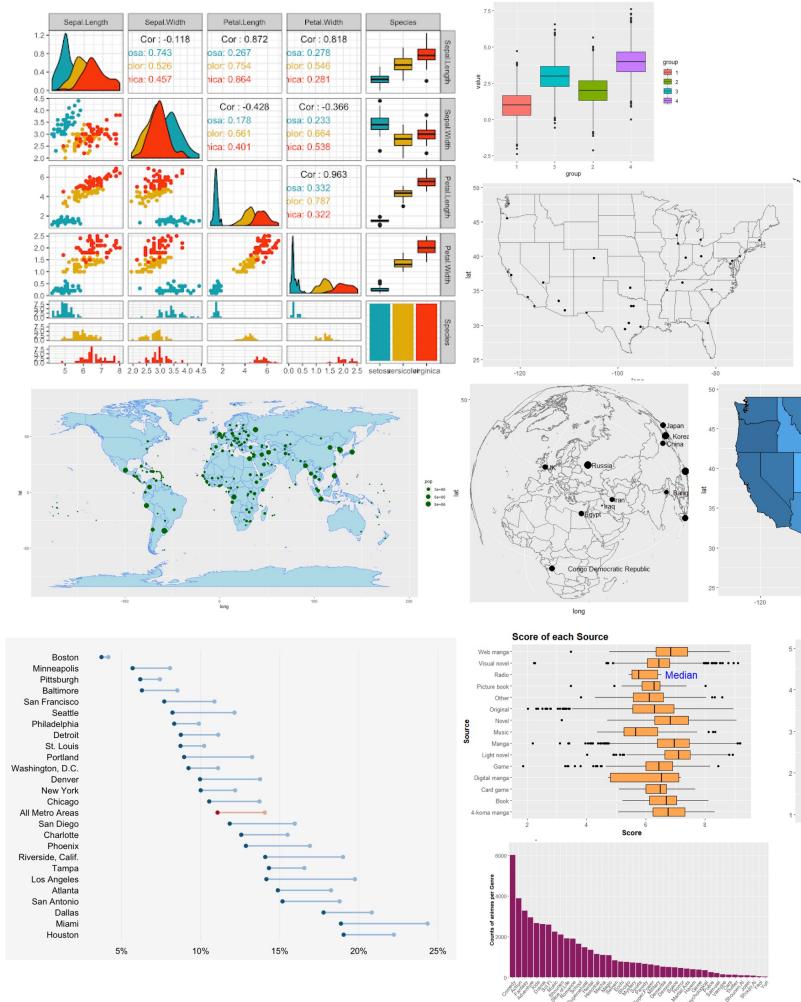
# response
# [1] 23.88889
```

R packages

- R has many useful functions built in...
- ...or you can add custom libraries or “**packages**” = coherent *collections of functions* for specific needs



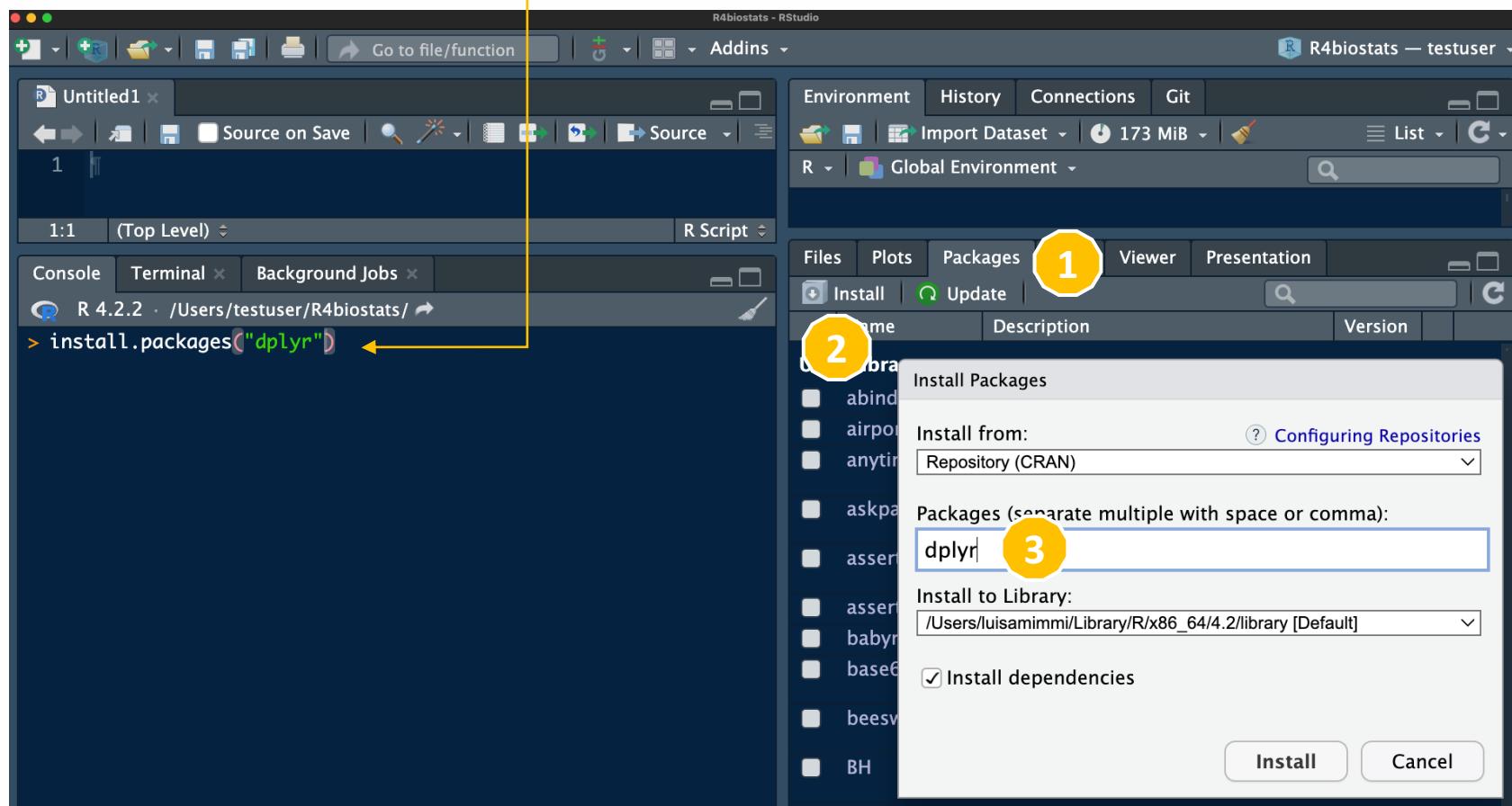
- EXAMPLE: **ggplot2** is an R package which introduces a variety of powerful visualisation and graphing options
 - (we will use it in our Lab session)



R package installation in RStudio: 2 ways

1) Writing the command in the console

2) Using the “Packages” tab



Getting Help with R 1/2

- There are a number of good resources on the web for learning R and seeking answers to questions... ~~copying code~~ 😊
- Here are just a few general ones:
 - POSIT resources (blog, cheatsheets, YT videos from conferences)
<https://posit.co/resources/>
 - *R for Data Science* book <https://r4ds.had.co.nz/>
 - POSIT community discussion board <https://community.rstudio.com/>
 - Stackoverflow with [r] tag <https://stackoverflow.com/>
 - Many courses/ tutorials / conferences by University professors/R programmers, personal websites, etc.
 - (on Coursera) Duke's Data Analysis with R Specialization
 - (Tutorialspoint) R Tutorial <https://www.tutorialspoint.com/r/index.htm>

Getting Help with R 2/2

- Here are some more R resources focused on epidemiology/ biostatistics:
 - R for Epidemiology <https://www.r4epi.com>
 - The Epidemiologist R Handbook <https://epirhandbook.com/en/>
 - Interactive R Tutorials <https://appliedepi.org/tutorial>
 - Introductory Biostatistics with R <https://tuos-bio-data-skills.github.io/intro-stats-book/index.html>
 - Biostatistics - Concepts and approaches for collecting good data and turning it into knowledge
https://jsgosnell.github.io/cuny_biosci_book/content/getting_started/getting_started.html
- Here are package repositories geared towards biostatistics:
 - rOpenSci: tools for open and reproducible research <https://ropensci.org/>
 - Bioconductor <https://www.bioconductor.org/>

DAY 1 – LECTURE OUTLINE

- Introduction to R and R-studio
 - Why R?
 - Principles of reproducible analysis with R + RStudio
- R objects, functions, packages
- Understanding different types of variables
 - Principles of “tidy data”
 - Data cleaning and manipulation
- Descriptive statistics
 - measures of central tendency, measures of variability (or spread), and frequency distribution
- Visual data exploration
 - {ggplot2}
- Foundations of inference

Qualitative and quantitative variables

- **Qualitative variables** express a qualitative attribute
 - E.g. hair color, religion, gender, favorite movie, etc.
- The **values** of a qualitative variable do not imply a numerical ordering
 - E.g. value of the variable “eye color” differ qualitatively; no ordering of eye color is implied
- Qualitative variables are also referred to as **categorical variables**
- **Quantitative variables**, instead, are measured in terms of numbers
 - E.g. height, weight, shoe size, etc.

Discrete and continuous (quantitative) variables

Quantitative

- **Discrete variables** can only take **certain** (countable) **values**
 - E.g. “children in a household” could be 3 or 6, but never 4.53 children
- **Continuous variables** can take **any value** within the range of the scale
 - [they are measured in units that can be subdivided]
 - E.g. “time to respond to a question” could be 1.64 seconds, because it is measured on a scale that is continuous and not made up of discrete steps

Another classification of variables (within experimental settings)

- When conducting research, experimenters often manipulate variables
 - E.g. if comparing the effectiveness of 4 types of antidepressants, the treatment variable is “type of antidepressant”
- When a variable is manipulated by an experimenter, it is called an **independent** (or *explanatory*) variable
 - Our experiment seeks to determine the effect of the independent variable on the effect in terms of “relief from depression”
- In this example, the measured effect/outcome is a **dependent** (or *response*) variable

In general, the *independent variable* is manipulated by the researcher and its effects on the *dependent variable* are measured

Variables' levels of measurement

- **Nominal** variables assign a **label/name** to each of the possible response categories, but there is **no natural order**
 - e.g. gender, blood type, favorite color, religion
- **Ordinal** variables assign a **label/name** to each of the possible response categories, but the **ranking of responses is meaningful**
 - e.g. consumer satisfaction levels, military rank, class ranking
- **Interval** variables assign **ordered values** that have **equal intervals**, but the zero-point is arbitrary (i.e. you can have negative values)
 - e.g. Celsius temperature scale, pH
- **Ratio** variables contain all the informative qualities of the *nominal*, *ordinal*, and *interval* scales, plus have a **true zero point** (i.e. zero has a meaning, like absence of the phenomenon)
 - e.g. age, weight, height, dose amount, pulse

informative value

Qualitative

Quantitative

The measurement scale matters for data analysis



Knowing the types of data and levels of measurement is crucial for choosing the appropriate statistical tools to analyze data

OK to compute....	Level of measurement			
	Nominal	Ordinal	Interval	Ratio
Frequency distribution	Yes	Yes	Yes	Yes
Median and percentiles	No	Yes	Yes	Yes
Add or subtract	No	No	Yes	Yes
Mean, standard deviation, standard error of the mean	No	No	Yes	Yes
Ratios, coefficient of variation	No	No	No	Yes

Data types in R corresponding to types of variables

Type of Variable	Data types in R	Value type	Example	Notes
continuous	Numeric	decimals	<code>num_dec <- c(3.4, 7.1, 2.9)</code>	
numeric	Integer <i>(special case of numeric)</i>	whole numbers	<code>num <- c(3, 7, 2)</code>	use if you are sure that the numbers you store will never contains decimals
nominal	Character	text contained in ""	<code>char <- "some text"</code> <code>also_char <- "2.3"</code> <code>also_char_2 <- c("text", 1, 3.72, 4)</code>	If a number or an element in a vector is inside "" --> R will store as character
ordinal	Factor <i>(special case of character)</i>	text contained in ""	<code>ranking_char <- c("low", "medium", "high")</code> <code>ranking_factor <- factor(ranking_char, levels = c("low", "medium", "high"))</code>	
boolean	Logical	TRUE or FALSE	<code>greater <- 1 > 10</code> <code>greater</code> <code>[1] FALSE</code>	

 Storing a variable as the wrong data type can generate errors in R programming

Data types in R... and the objects storing them

Data type	Type
vector	1D collection of variables of the same type
matrix	2D collection of variables of the same type
data.frame	2D collection of variables of multiple types

Variables	Example
integer	100
numeric	0.05
character	“hello”
logical	TRUE
factor	“Green”

Vector



Matrix



Data frame

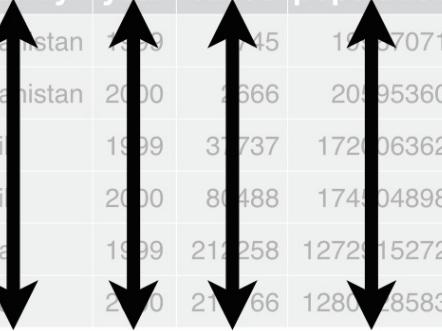


Tidy (rectangular) data makes analysis easier

- In R, we are working with tables, which are stored in special data structures called “**data frames**” (or “**tibbles**”).
 - a typical example is a table in a relational database
- In (**tidy**) rectangular data structures:
 - **columns** will represent different **variables** associated with each data point or instance e.g. Name, ID, location, time, value...
 - **rows** will represent instances or **individual observations** e.g. data points, patients, events, samples, etc. while
 - each **cell** (intersection of row and column) will contain one and only **value** (i.e. the state of a variable measured for one specific individual/unit)

country	year	cases	population
Afghanistan	1999	745	1997071
Afghanistan	2000	2666	20595360
Brazil	1999	37737	172006362
Brazil	2000	80488	174504898
China	1999	212258	1272015272
China	2000	213766	128042583

variables



country	year	cases	population
Afghanistan	1999	745	1997071
Afghanistan	2000	2666	20595360
Brazil	1999	37737	172006362
Brazil	2000	80488	174504898
China	1999	212258	1272015272
China	2000	213766	128042583

observations



country	year	cases	population
Afghanistan	1999	745	1997071
Afghanistan	2000	2666	20595360
Brazil	1999	37737	172006362
Brazil	2000	80488	174504898
China	1999	212258	1272015272
China	2000	213766	128042583

values

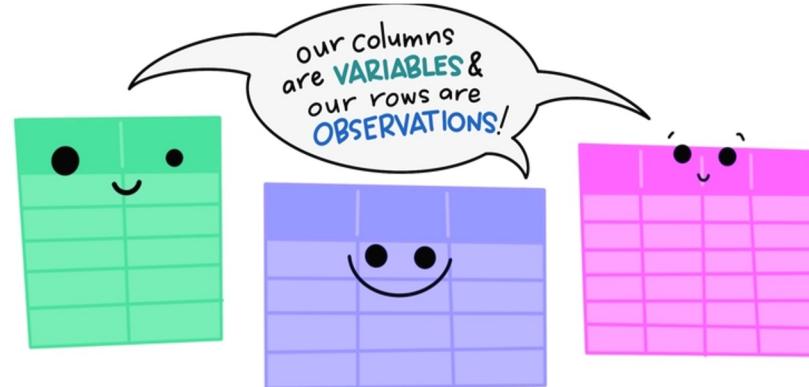


Source: <https://r4ds.hadley.nz/data-tidy.html>

Why do we care about tidy data form?

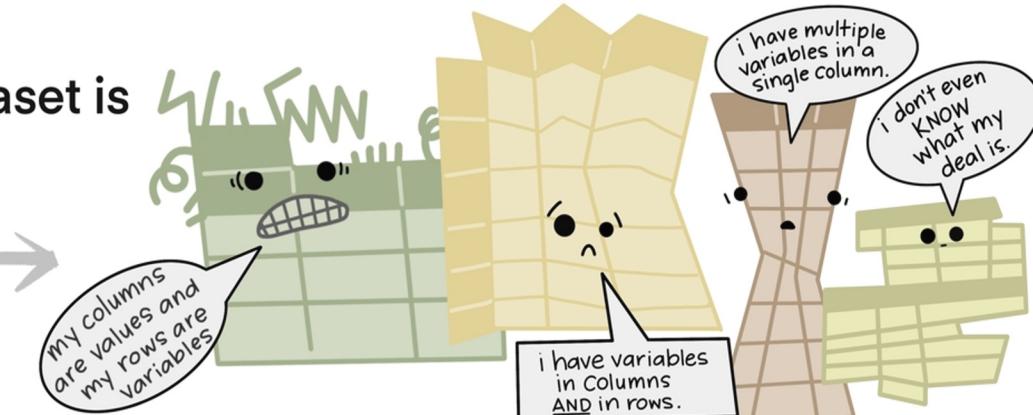
- uniformity & standardization
- readability
- ease to clean/analyze/model/plot data

"tidy datasets are all alike..."



"...but every messy dataset is
messy in its own way."

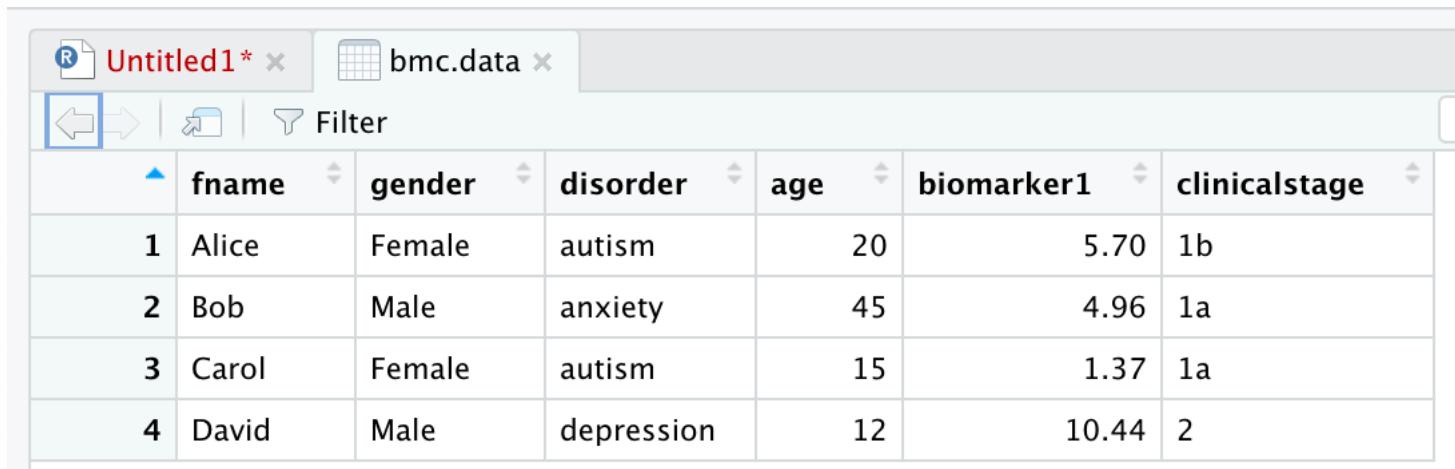
—HADLEY WICKHAM



Source: art by [Allison Horst](#)

An example of “tidy”, rectangular data frame

```
bmc_data <- data.frame (name = c("Alice", "Bob", "Carol", "David"),
                         gender = as.factor(c("Female", "Male", "Female", "Male")),
                         disorder = c("autism", "anxiety", "autism", "depression"),
                         age = c(20, 45, 15, 12),
                         biomarker1 = c(5.70, 4.96, 1.37, 10.44),
                         clinicalstage = c("1b", "1a", "1a", "2"),
                         stringsAsFactors = FALSE)
```



The screenshot shows the RStudio interface with two tabs open: "Untitled1*" and "bmc.data". The "bmc.data" tab is active, displaying a data frame with the following structure:

	fname	gender	disorder	age	biomarker1	clinicalstage
1	Alice	Female	autism	20	5.70	1b
2	Bob	Male	anxiety	45	4.96	1a
3	Carol	Female	autism	15	1.37	1a
4	David	Male	depression	12	10.44	2

Reshaping the data: from wide to long form

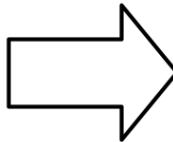
Suppose we have 3 patients with ids A, B, and C, and we take 2 blood pressure measurements on each patient

- We want our new dataset to have three variables: patient **id**, **measurement**, and **value**



You typically need data in "long" form for modeling and plotting

id	bp1	bp2
A	100	120
B	140	115
C	120	125



id	measurement	value
A	bp1	100
A	bp2	120
B	bp1	140
B	bp2	115
C	bp1	120
C	bp2	125

Source: <https://r4ds.hadley.nz/data-tidy.html>

Reshaping the data: from long to wide form

You may also need to convert a dataset from long to wide format

- Data stored in 1 column (variable **year**) is now splitted in **separate variables**, so that the table has only 1 row per country



You typically need data in "wide" form for better readability

country	year	cases
Angola	1999	800
Angola	2000	750
Angola	2001	925
Angola	2002	1020
India	1999	20100
India	2000	25650
India	2001	26800
India	2002	27255
Mongolia	1999	450
Mongolia	2000	512
Mongolia	2001	510
Mongolia	2002	586

country	1999	2000	2001	2002
Angola	800	750	925	1020
India	20100	25650	26800	27255
Mongolia	450	512	510	586

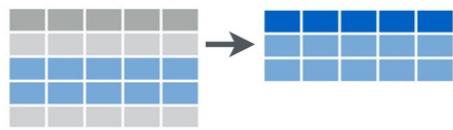


Source: <https://epirhandbook.com/>

Subsetting the data

- Subsetting is a fundamental action that serves many purposes:
 - viewing and getting a quick sense of the variables
 - analyzing a sub-group of individuals
 - splitting samples
 - etc.
- We will see in the practice sessions that there are **many ways** to subset a dataframe in R.

Subset Observations (Rows)



Subset Variables (Columns)



Dealing with missing data

- **Missing data** = the value of the variables of interest are not measured or recorded for all subjects in the sample
- In clinical research, missing data can be handled with different approaches:
 - complete-case analysis -> subsetting complete cases only
 - mean-value imputation -> missing values are replaced with the mean/median/mode value of that variable
 - model-based imputation -> missing values imputed based on a linear regression model or K-Nearest Neighbors predictive model, interpolation, etc....
 - multiple imputation (MI) -> multiple plausible values are imputed or filled in for each subject who has missing data for that variable and then multiple datasets are analyzed
 - domain-specific imputation -> justified imputation based on external data (i.e. blood type when you have family history)



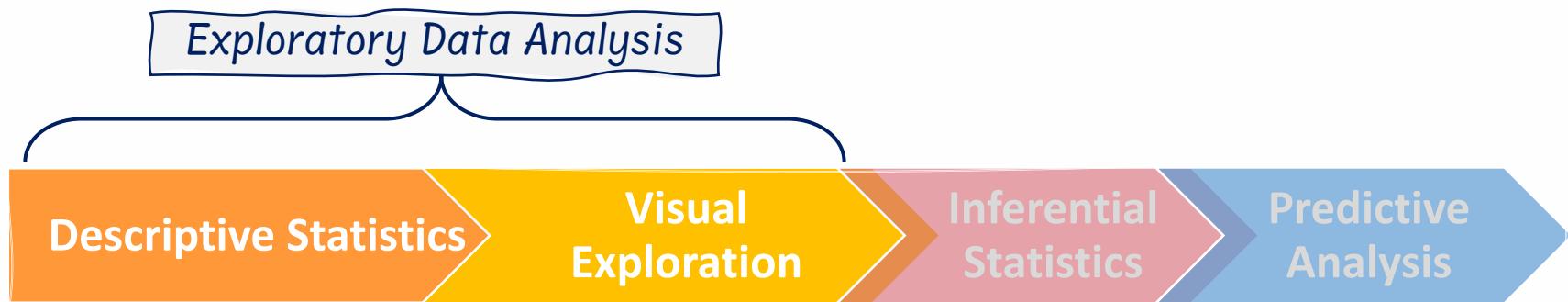
1. It is SUPER important to understand the reason some data are missing (random or not) and to what extent they are related to the dependent variable
2. Dealing with missing data can lead to biased estimates of statistics (e.g., of regression coefficients) and/or confidence intervals that are artificially narrow.



DAY 1 – LECTURE OUTLINE

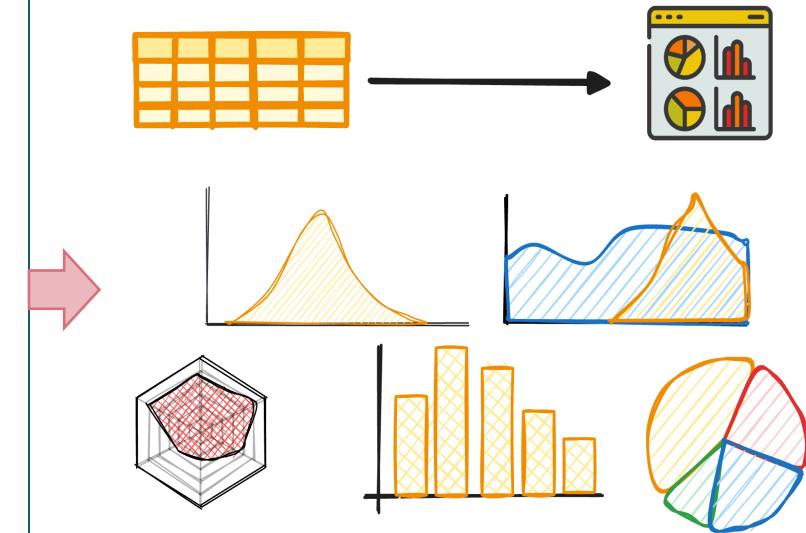
- Introduction to R and R-studio
 - Why R?
 - Principles of reproducible analysis with R + RStudio
- R objects, functions, packages
- Understanding different types of variables
 - Principles of “tidy data”
 - Data cleaning and manipulation
- Descriptive statistics
 - measures of central tendency, measures of variability (or spread), and frequency distribution
- Visual data exploration
 - {ggplot2}
- Foundations of inference

Describe dataset to: familiarize with it(!), check quality, generate hypotheses before inferential modeling



Quantitative & visual approach to learn:

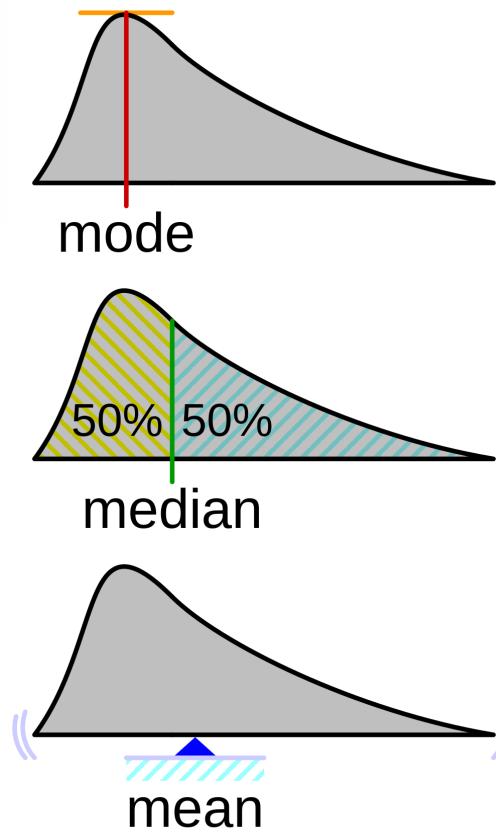
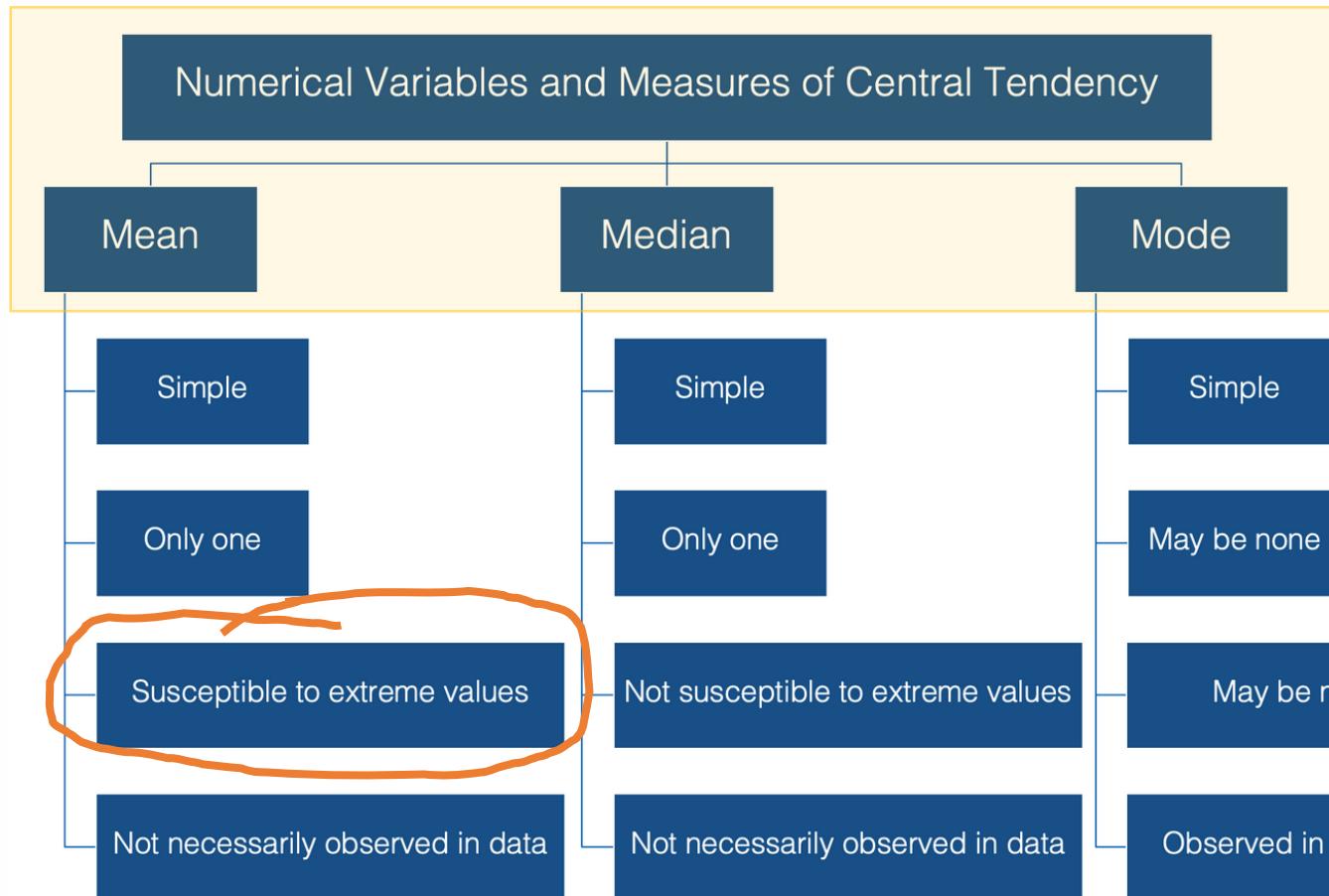
- **VARIABLES' MEASURES OF CENTER & DISPERSION**
 - Mean / Median / Mode
 - Min & max
 - Standard deviation
 - Variance
 - Range / Interquartile Range (IQR)
- **VARIABLES' DISTRIBUTION AND PATTERNS**
 - Frequency distributions
 - Geographical distribution
 - Correlation and joint variability
 - Missing observations, anomalies
 - Outliers



Source: <https://www.kdnuggets.com/7-steps-to-mastering-exploratory-data-analysis>

Measures of central tendency: mean, median, mode

- Mean (sum of all observations divided by n of observations)
- Median (value at the midpoint of a frequency distribution of observed values)
- Mode (the value that is most often observed in the data)



Source: <https://www.r4epi.com/measures-of-central-tendency>

Mean, variance and Standard deviation equations

Mean

$$\text{Sample } \mu = \frac{\sum_{i=1}^n x_i}{n} \text{ & Population } \bar{x} = \frac{\sum_{i=1}^n x_i}{n}$$

Variance

$$\text{Sample } s^2 = \frac{\sum (x_i - \bar{x})^2}{n-1} \text{ & Population } \sigma^2 = \frac{\sum_{i=1}^n (x_i - \mu)^2}{n}$$

Standard deviation

$$\text{Sample } s = \sqrt{\frac{\sum (x_i - \bar{x})^2}{n-1}} \text{ & Population } \sigma = \sqrt{\frac{\sum_{i=1}^n (x_i - \mu)^2}{n}}$$

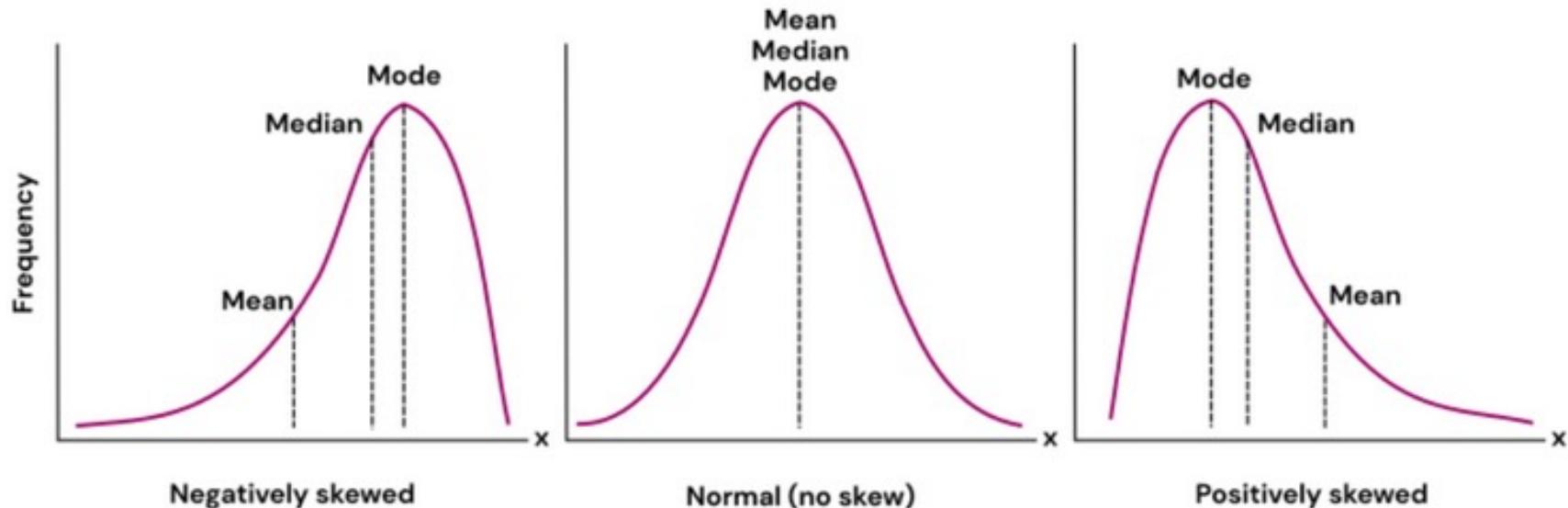
Median equations

For uneven n : $\frac{x_{(n+1)}}{2}$

For even n : $\frac{x_{(n/2)} + x_{(n/2+1)}}{2}$

The median is not influenced by:

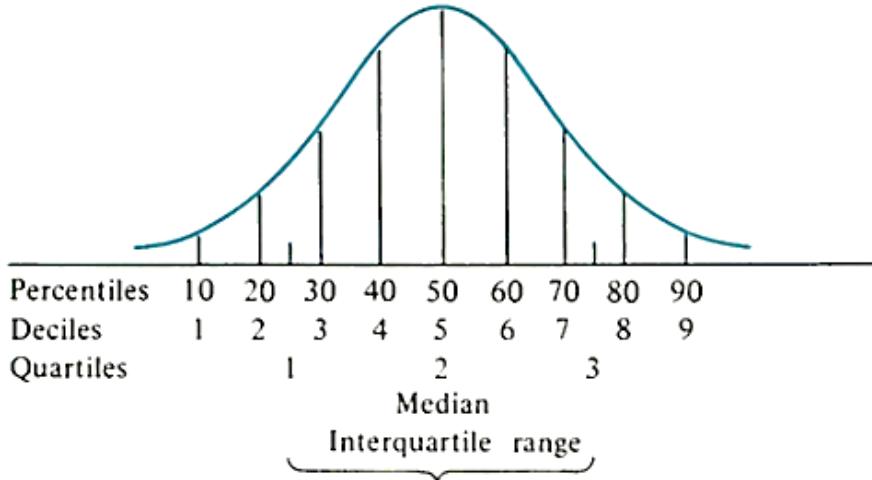
- variability in the data
- extreme values/outliers



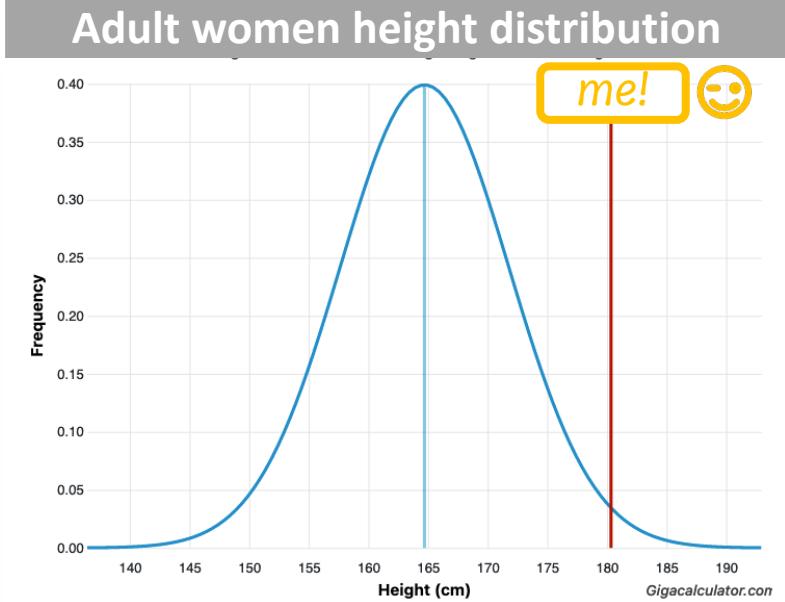
Partition values (quantiles)

- Partition values partition the same collection of observations in several ways
- Similar to the median, depending on their position, quantiles contain portion of observations in the frequency distribution of a variable:
 - quartiles: distribution divided into quarters
 - quintiles: distribution divided into fifths
 - deciles: distribution divided into tenths
 - percentiles : distribution divided into hundredths

For example, I am in the top 1st percentile of world's adult women for height!

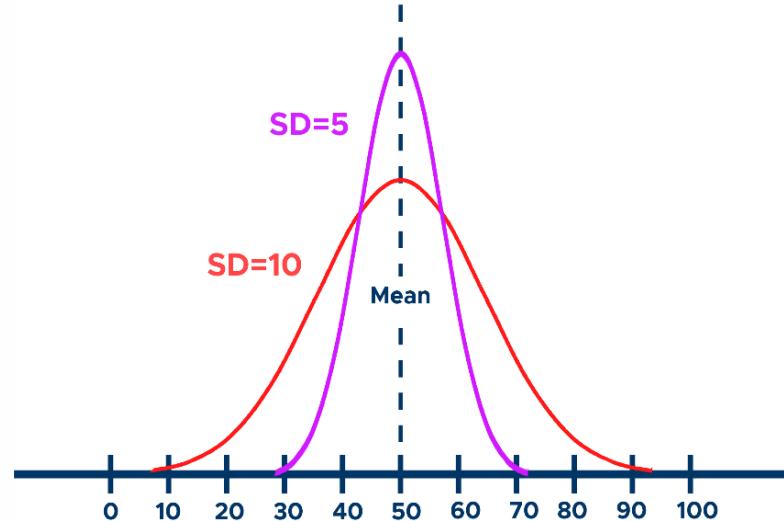
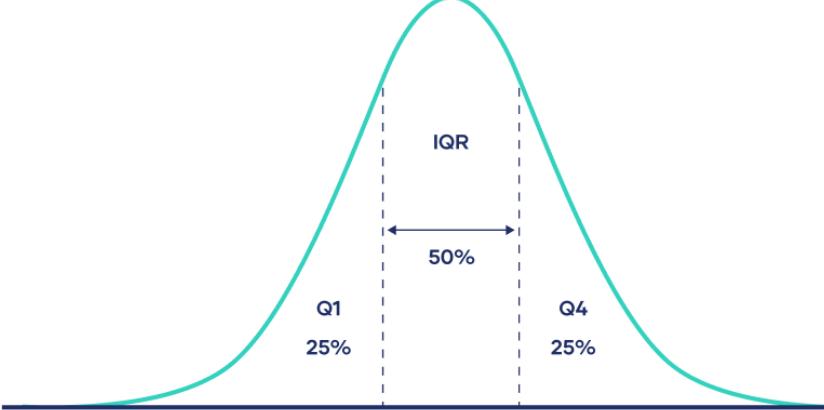


Source: <https://itfeature.com/statistics/quantiles-or-fractiles/>



Measures of variability or dispersion

- Capable of expressing in one measure the elements of heterogeneity of (quantitative) data:
 - Range: difference between the maximum and minimum values
 - Interquartile Range (IQR): range of the middle half of a distribution
 - Variance: average of squared distances from the mean
 - Standard deviation: average distance from the mean
 - Coefficient of variation



Examples of {base} R built-in functions

(central tendency and dispersion)

Examples of {base} R functions to summarize your data

Built-in {base} R functions

- `summary(bmc_data)` # can take dataframe as argument
- `summary(bmc_data$age)` # results change according to var type
- `table (bmc_data$age, useNA = "ifany")`
- `table (bmc_data$gender, useNA = "ifany")`
- `mean (bmc_data$age, na.rm = TRUE)` # sample mean $\bar{x} = \frac{\sum(x_i))}{n}$
- `median (bmc_data$age, na.rm = TRUE)` # observation corresponding to index $\frac{n+1}{2}$ (when values are in order low to high)

Examples of user-defined function to summarize your data

Our own custom function to calculate mode

```
# Create custom function to calculate statistical mode
|
f_calc_mode <- function(x) {
  # `unique` returns a vector of unique values
  uni_x <- unique(x)
  # `match` returns the index positions of 1st vector against 2nd vector
  match_x <- match(x, uni_x)
  # `tabulate` count the occurrences of integer values in a vector.
  tab_x <- tabulate(match_x)
  # returns element of uni_x that corresponds to max occurrences
  uni_x[tab_x == max(tab_x)]
}
```

Use new function

```
f_calc_mode(bmc_data$age)
f_calc_mode(bmc_data$biomarker1)
```

DISPERSION: range, min, max, IQR

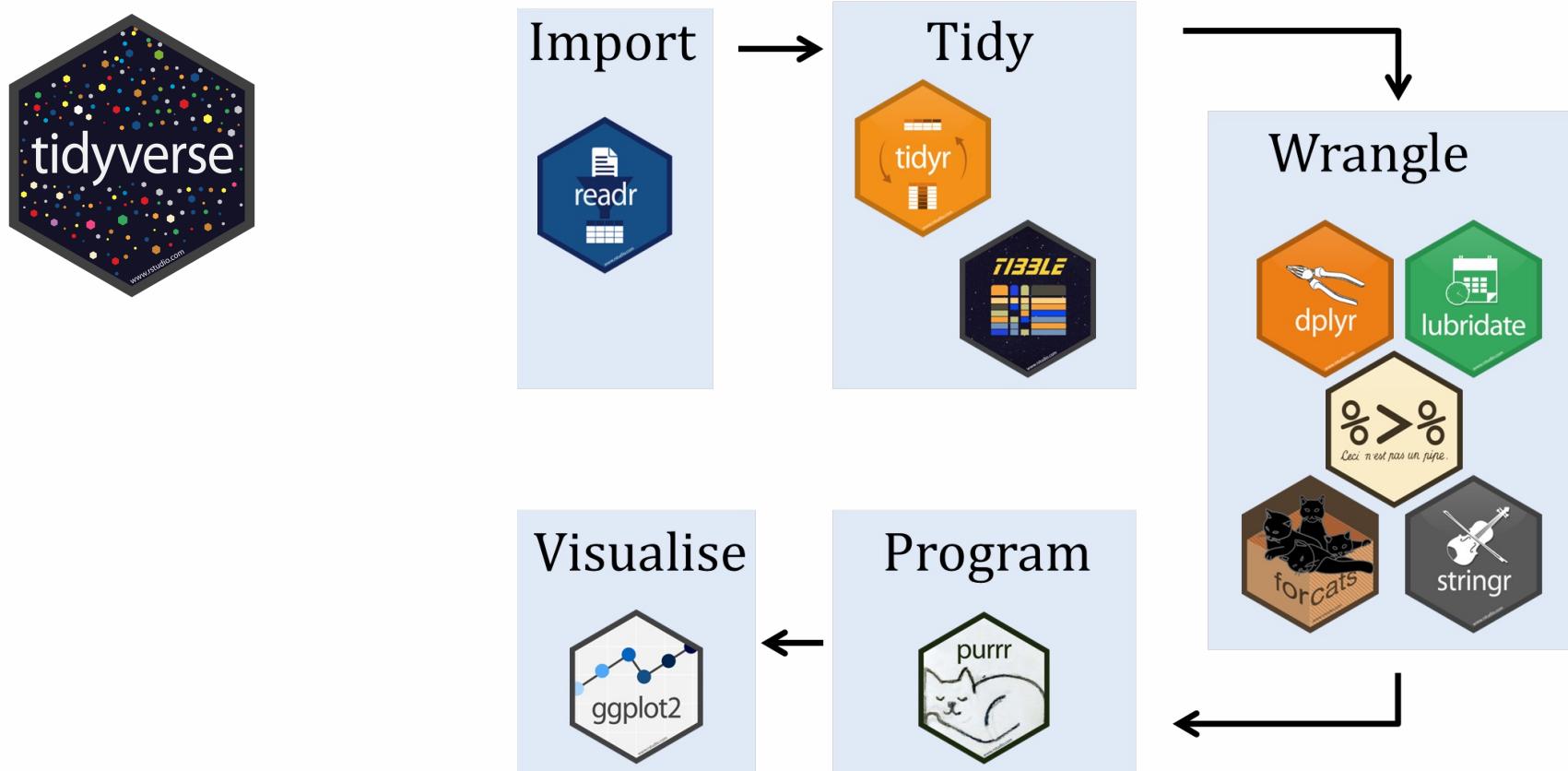
- `range` = difference between the largest and smalles value in a distribution
- `IQR (interquartile range)` = difference between the first quartile (the 25th percentile) and the third quartile (the 75th percentile) of a dataset

Built-in {base} R functions

- `min(bmc_data$age)`
- `max(bmc_data$age)`
- `range(bmc_data$age)`
- `IQR(bmc_data$age, na.rm=TRUE) # Q3 – Q1 (where Q1 and Q3 are the 25th and 75th percentiles)`
- `var(bmc_data$age) # variance (for sample of size n)` $s^2 = \frac{\sum(x_i - \bar{x})}{n-1}$
- `sd(bmc_data$age) # standard deviation (for sample of size n)` $s = \sqrt{\frac{\sum(x_i - \bar{x})}{n-1}}$

The {tidyverse} paradigm

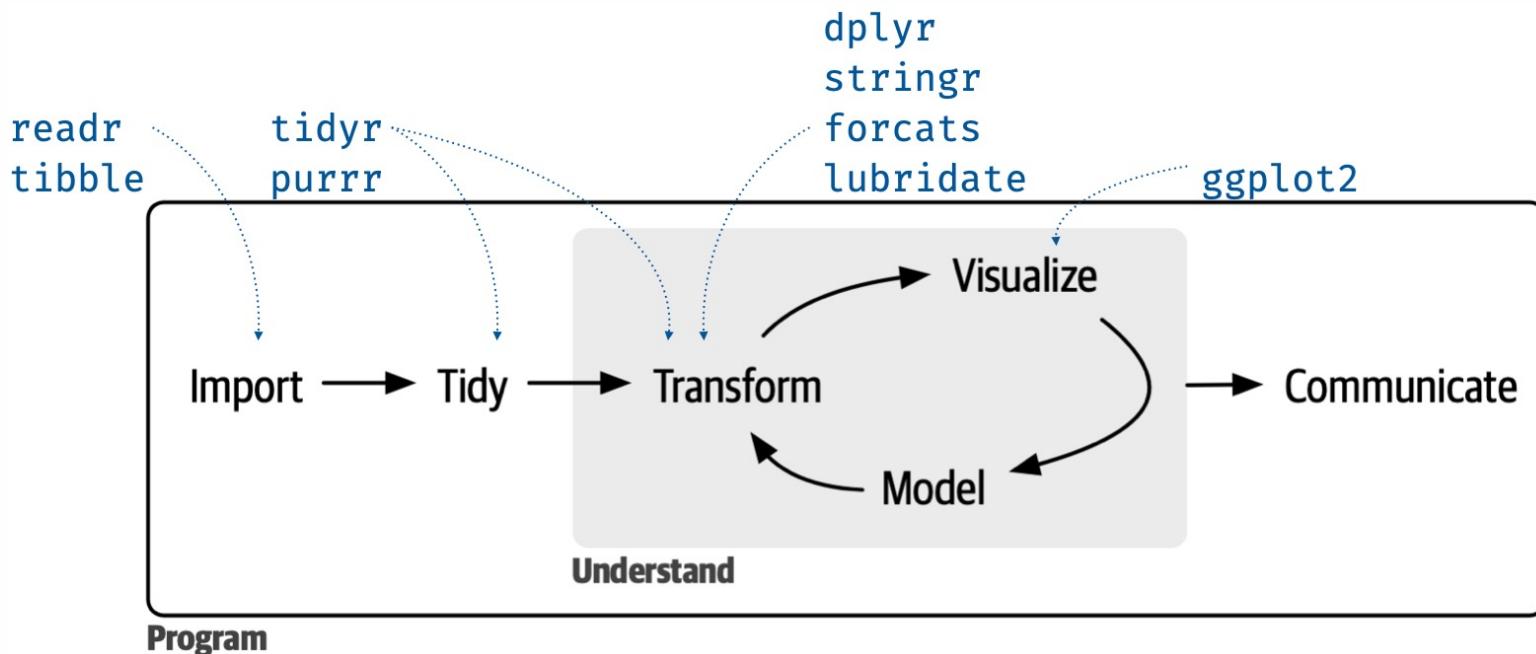
A coherent collection of R packages to carry out the entire analytical process



Source: <https://ajsmit.github.io>

The `{tidyverse}` “frame of mind” for data transformation & plotting

- Besides `{base}` R functions, many R programmers (especially beginners) enjoy using `{tidyverse}`, a collection of packages with:
 - consistent philosophy, grammar, and data structures
 - the pipe `%>%` linking functions!!
- An alternative package/framework for data analysis (*we won't look at*) is `{data.table}`



Source: <https://www.tidyverse.org/blog/2023/08/teach-tidyverse-23/>

Some `{tidyverse}` R functions to transform data

A great package for data wrangling is `{dplyr}`

These **dplyr functions** (reminiscent of SQL commands) reflect the fact what that they 'do' to data.

For example:

- `select` obtains a subset of variables,
- `mutate` constructs new variables,
- `filter` obtains a subset of rows,
- `arrange` reorders rows,
- `group_by` groups based on a categorical variable... then
- `summarise` calculates information about groups.

All `{tidyverse}` functions can be chained by the pipe `%>%` operator!

- This operator (from the `{magrittr}` package) is similar to the one used in other programming languages (e.g. `|` in the Shell or Terminal)
- `%>%` allows to easily and intuitive chain sequences of functions:
 - the output of one function becomes the input used by the next function
 - instead of `f(g(x))`, we write our code as `x %>% g() %>% f()`

From

```
leave_house(get_dressed(get_out_of_bed(wake_up(me, time =  
"8:00"), side = "correct"), pants = TRUE, shirt = TRUE), car  
= TRUE, bike = FALSE)
```

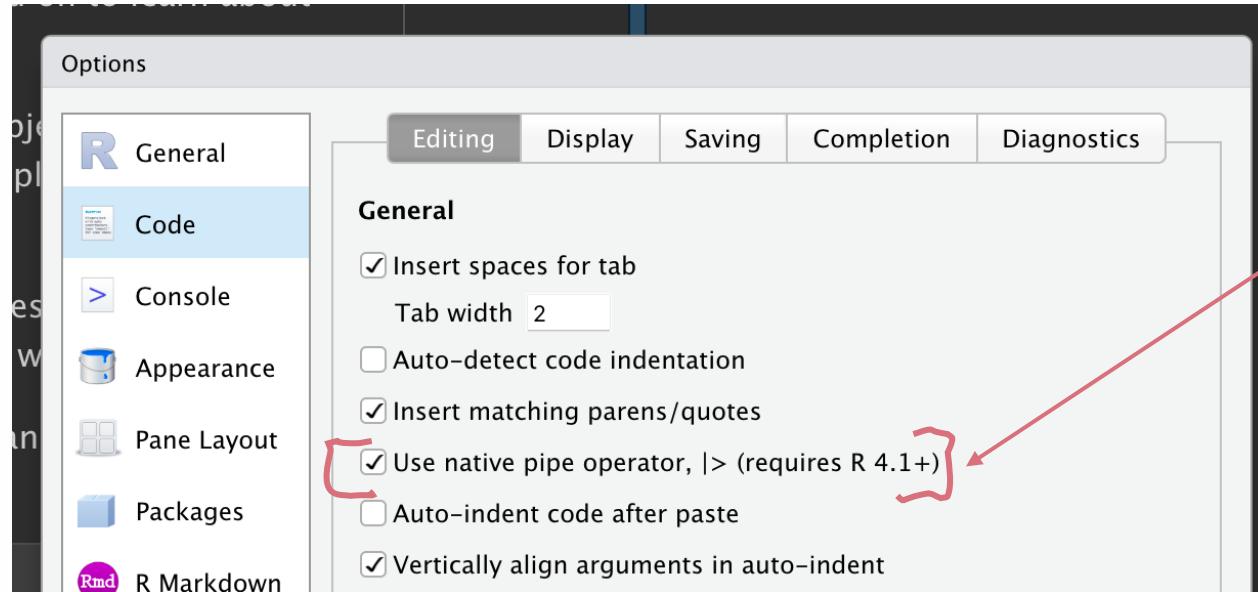
To

```
me %>%  
  wake_up(time = "8:00") %>%  
  get_out_of_bed(side = "correct") %>%  
  get_dressed(pants = TRUE, shirt = TRUE) %>%  
  leave_house(car = TRUE, bike = FALSE)
```

Source: https://talks.andrewheiss.com/2021-seacen/01_welcome-tidyverse/slides/01_tidyverse-dplyr.html#65

R has a new native pipe |> which can replace the {magrittr} package's one (%>%)

- R 4.1.0 introduced a native pipe operator, |>, which behaves very similarly to %>%



In RStudio options, you can choose to use the native |> here

Examples of {dplyr} functions for summarizing data

Functions: filter, group_by, summarise

```
# input the dataframe name
bmc_data %>%
  # filter some rows
  dplyr::filter(age >= 15) %>%
  # split by some column
  dplyr::group_by(gender) %>%
  # summarise in each group
  dplyr::summarise(mean(biomarker1)) # returns a dataframe!
```

Results

<code><fct></code>	<code><dbl></code>
1 Female	3.54
2 Male	4.96

3 alternative R packages/paradigms for data manipulation

OPERATION	PARADIGM		
	base	tidyverse	data.table
Supported data class	<code>data.frame</code>	<code>tibble</code>	<code>data.table</code>
Reading data	<code>read.csv</code>	<code>read_csv</code>	<code>fread</code>
Subset by column	<code>[, ...]</code>	<code>select()</code>	<code>[, ... ,]</code>
Subset by rows	<code>[... ,]</code>	<code>filter()</code>	<code>[... , ,]</code>
Create new column	<code>df\$y = ...</code>	<code>mutate(tb, y = ...)</code>	<code>[, y := ... ,]</code>
Delete a column	<code>df\$y = NULL</code>	<code>select(tb, -y)</code>	<code>[, y := NULL,]</code>
Summarize	<code>apply(df[, y], 2, ...)</code>	<code>summarise()</code>	<code>[, ...(y),]</code>
Grouping	<code>aggregate()</code>	<code>group_by()</code>	<code>[, , by = ...]</code>
Pivot to long	<code>reshape()</code>	<code>pivot_longer()</code>	<code>melt()</code>
Pivot to wide	<code>reshape()</code>	<code>pivot_wider()</code>	<code>dcast()</code>
Joining tables	<code>merge()</code>	<code>left_join()</code>	<code>dt1[dt2, on = ...]</code>

Source: <https://jtr13.github.io/cc21fall2/comparison-among-base-r-tidyverse-and-datable.html#summary-of-key-functions>

DAY 1 – LECTURE OUTLINE

- Introduction to R and R-studio
 - Why R?
 - Principles of reproducible analysis with R + RStudio
- R objects, functions, packages
- Understanding different types of variables
 - Principles of “tidy data”
 - Data cleaning and manipulation
- Descriptive statistics
 - measures of central tendency, measures of variability (or spread), and frequency distribution
- Visual data exploration
 - {ggplot2}
- Foundations of inference

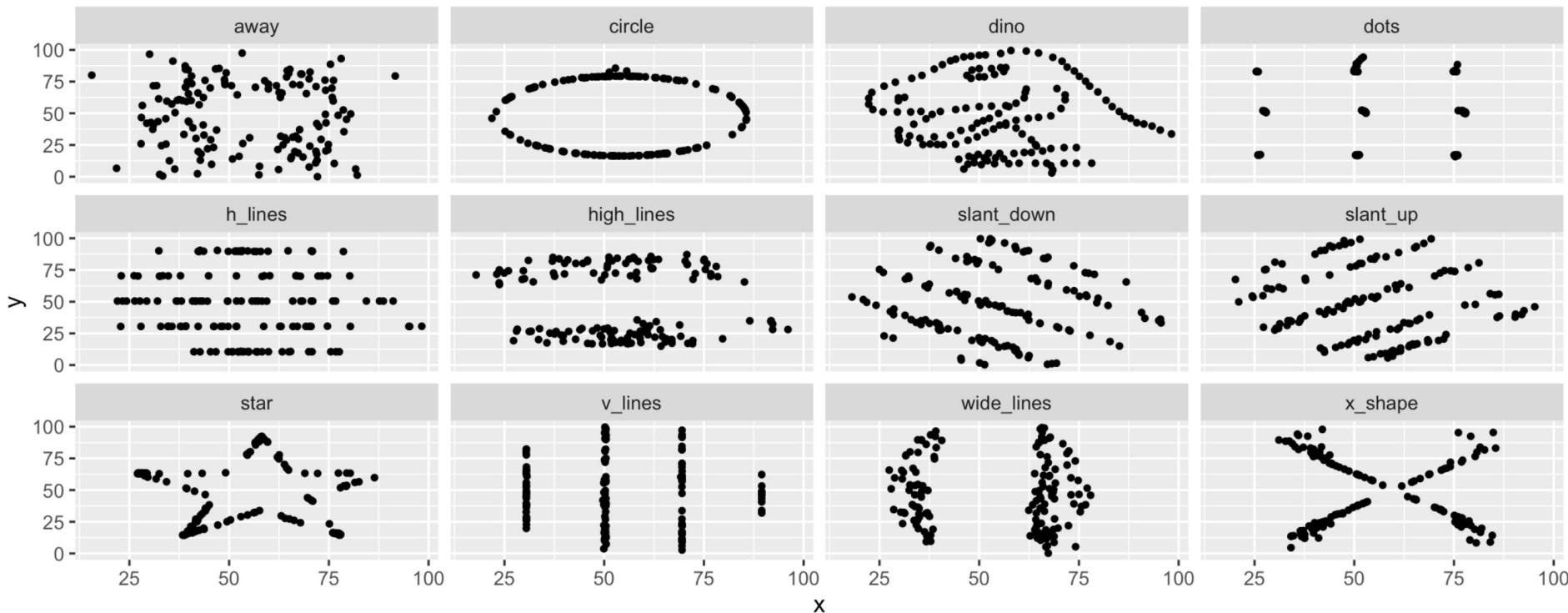
Why pay attention to data visualization?

- To **explore** data
- To **diagnose** data patterns & preliminary insights
- To **display** results and reinforce messages shared with readers of a publication

Believe it or not...

Each of the datasets depicted below (sets of x,y coordinates) has the same **mean**, **standard deviation**, **variance**, and **correlation**!

- x mean = 54.26 y mean = 47.83
- x sd = 16.76 y sd = 26.93
- correlation (x,y) = -0.06



Source: <https://talks.andrewheiss.com/2021-seacen/02-ggplot2.html>

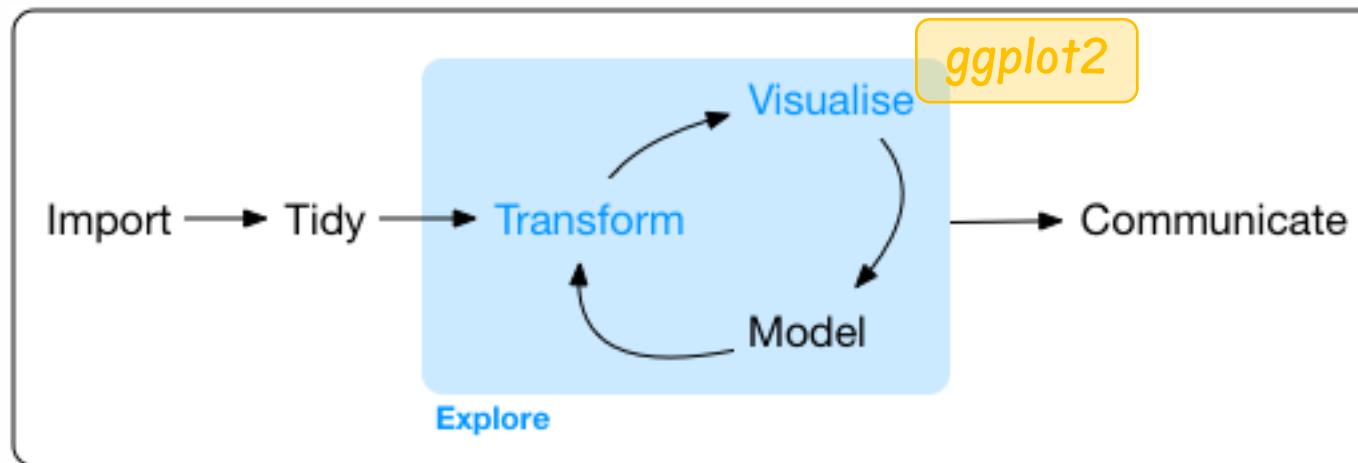
The `{tidyverse}` package `ggplot2` is very helpful for the visualization of data

`ggplot2`'s conception of plot builds on some fundamental parts:

- **data** is a data frame
- **aesthetics** is used to indicate x and y variables. It can also be used to control the color, the size or the shape of points, the height of bars, etc.
- **geometry** defines the type of graphics (*histogram, box plot, line plot, density plot, dot plot, ...*)

with the addition of:

- **statistics** transform data on the way to visualization
- **scales** manipulate the labels, breaks, transformations and palettes
- **facets** to split the plot by a category
- **coordinate systems** to flip the graph
- **labels** for clarity
- **theme** for esthetic improvements

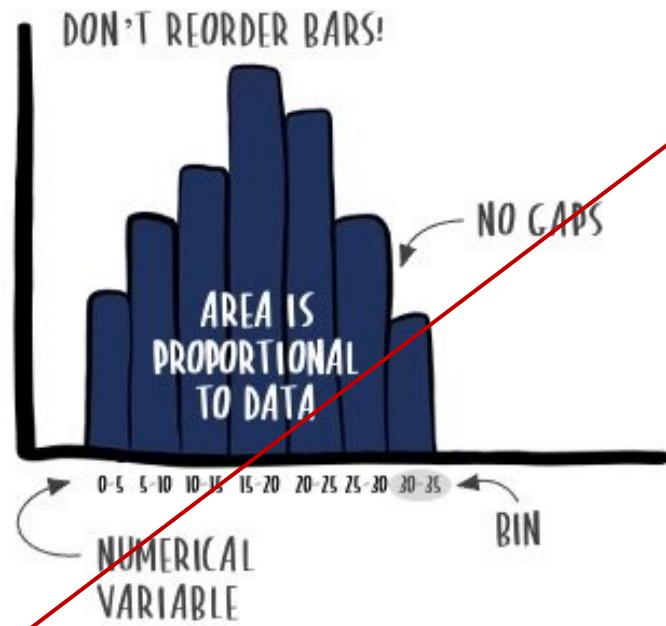


Source: <https://www.tidyverse.org/blog/2023/08/teach-tidyverse-23/>

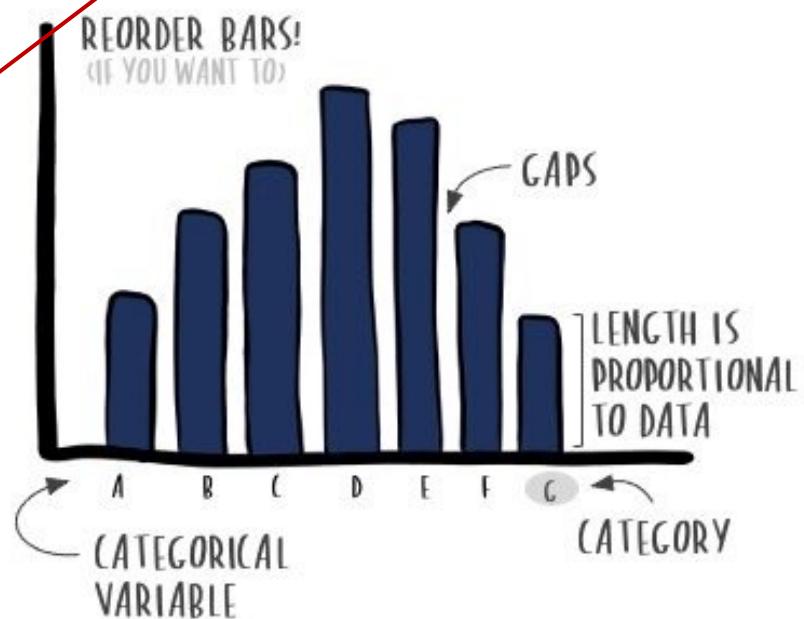
Histogram and barchart

non so cosa dire

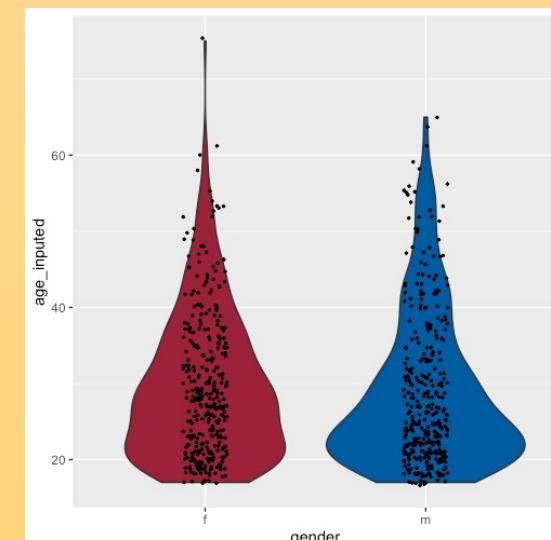
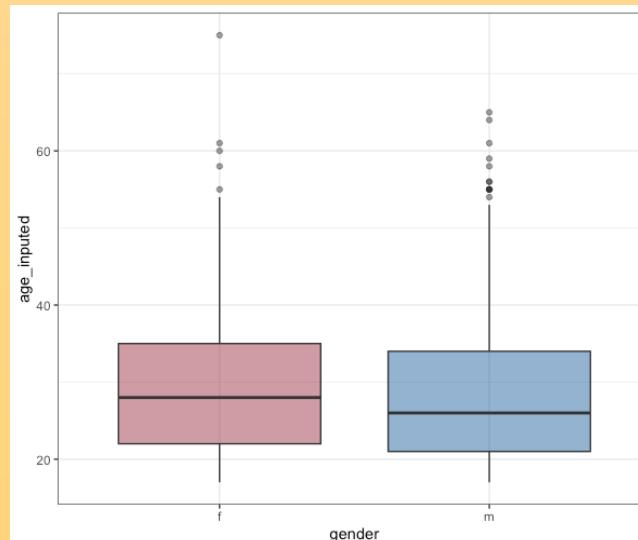
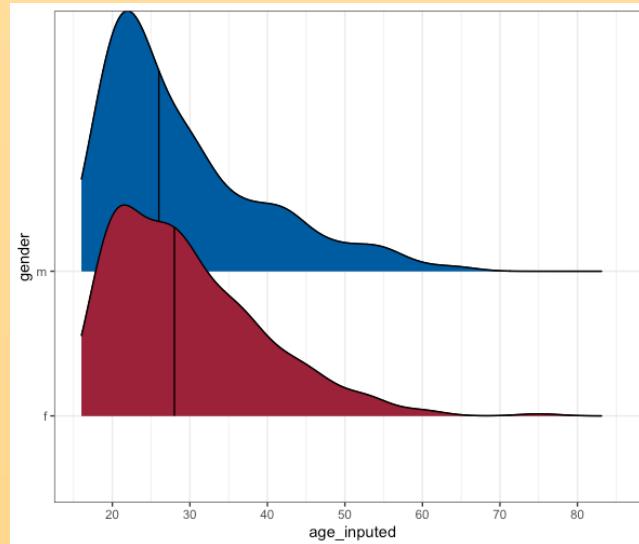
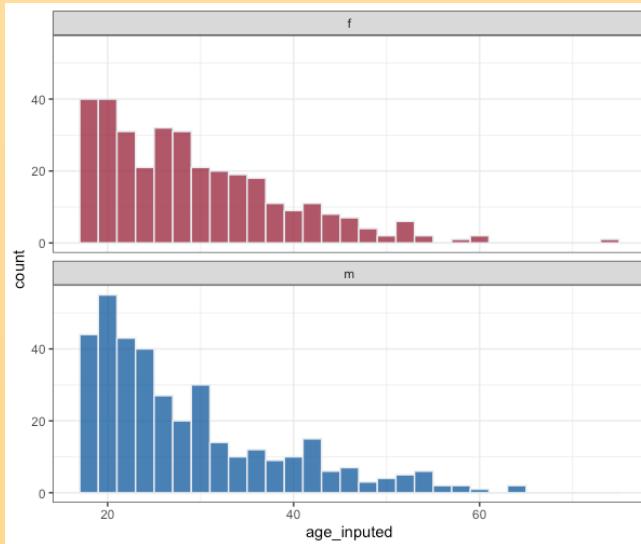
This is a **histogram**...



This is a **bar chart**...



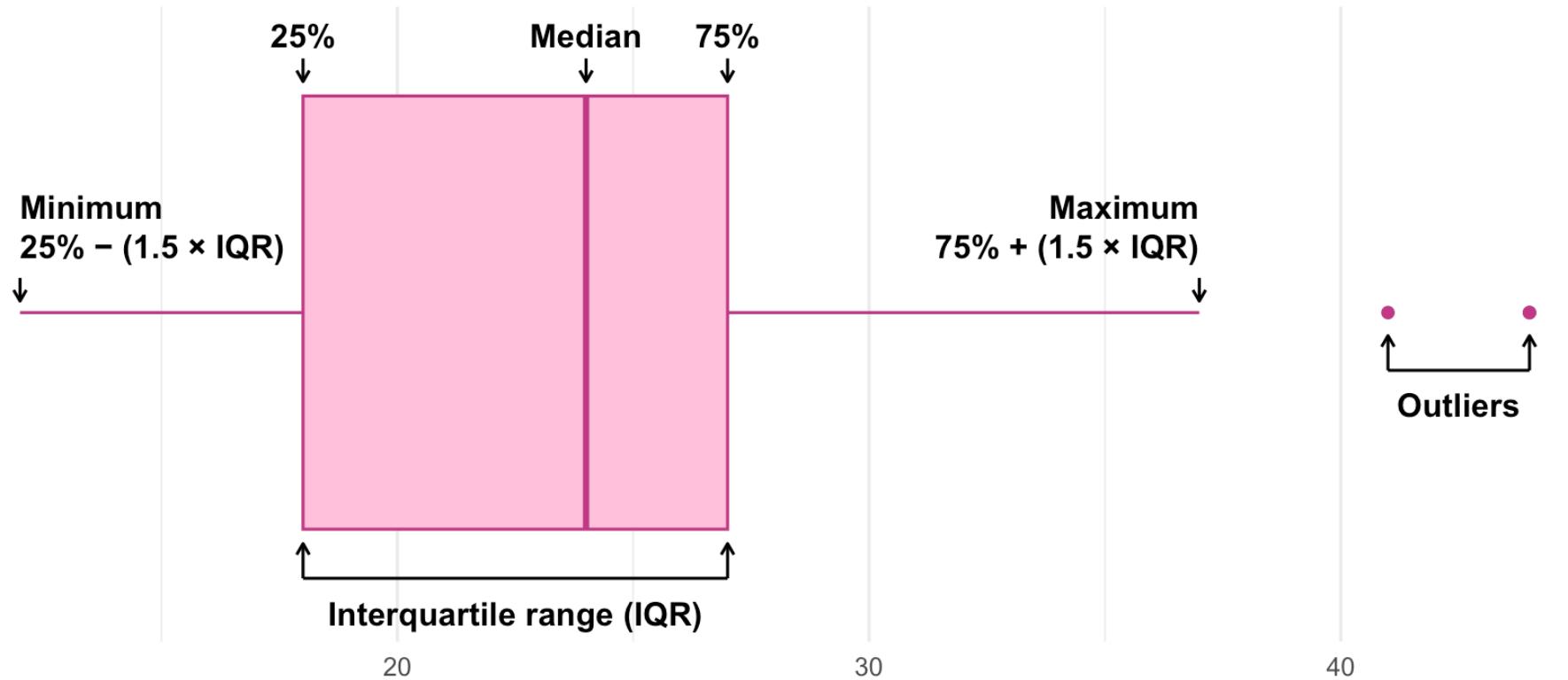
Different ways to plot a frequency distribution of a continuous variable: histogram / density plot / boxplot / violin plot



Source: autism dataset (see lab)

Anatomy of a boxplot

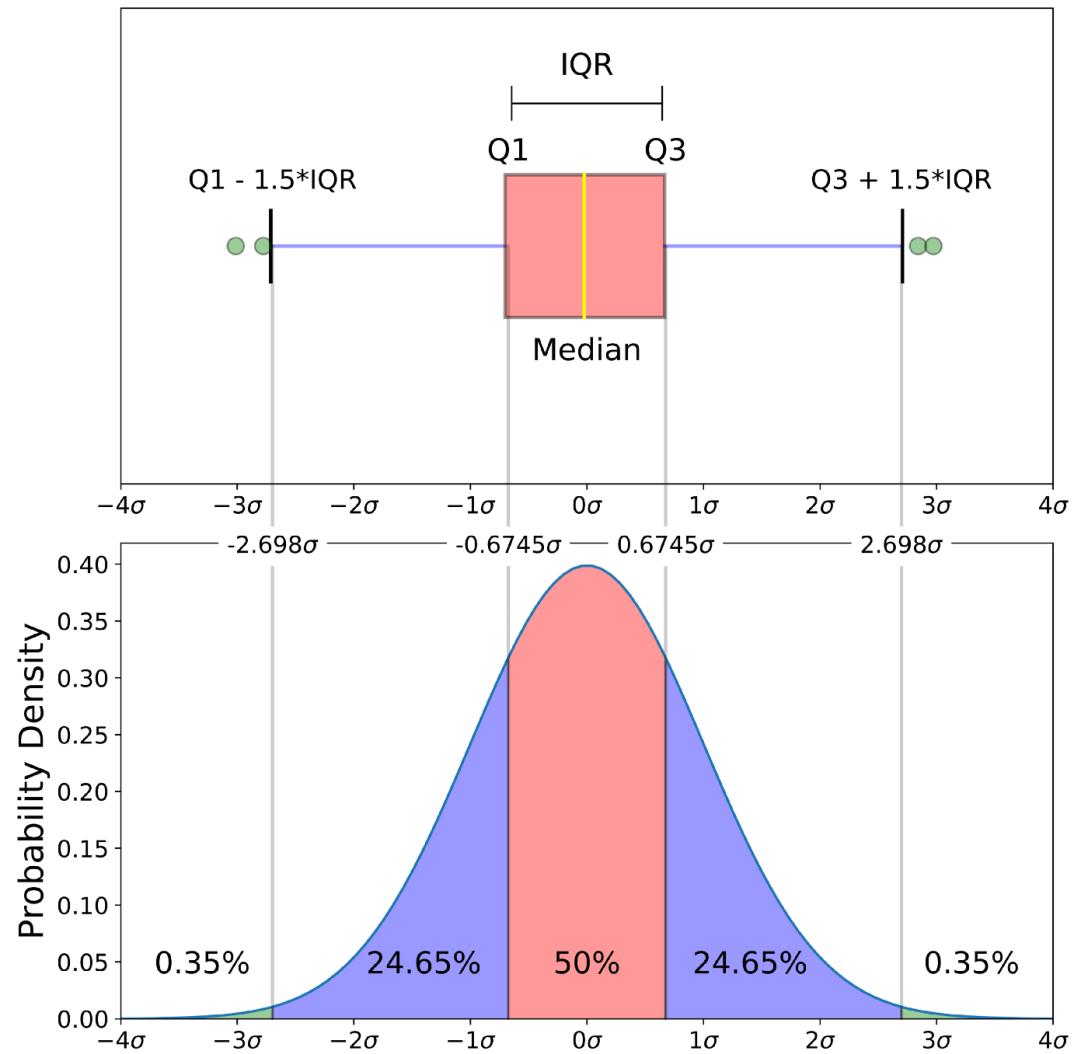
Box plots show the distribution of a variable by highlighting specific details, like the 25th, 50th (median) and 75th percentile, as well as the assumed minimum, assumed maximum, and outliers.



Source: <https://datavizf23.classes.andrewheiss.com/lesson/06-lesson.html#boxes-violins-and-dots>

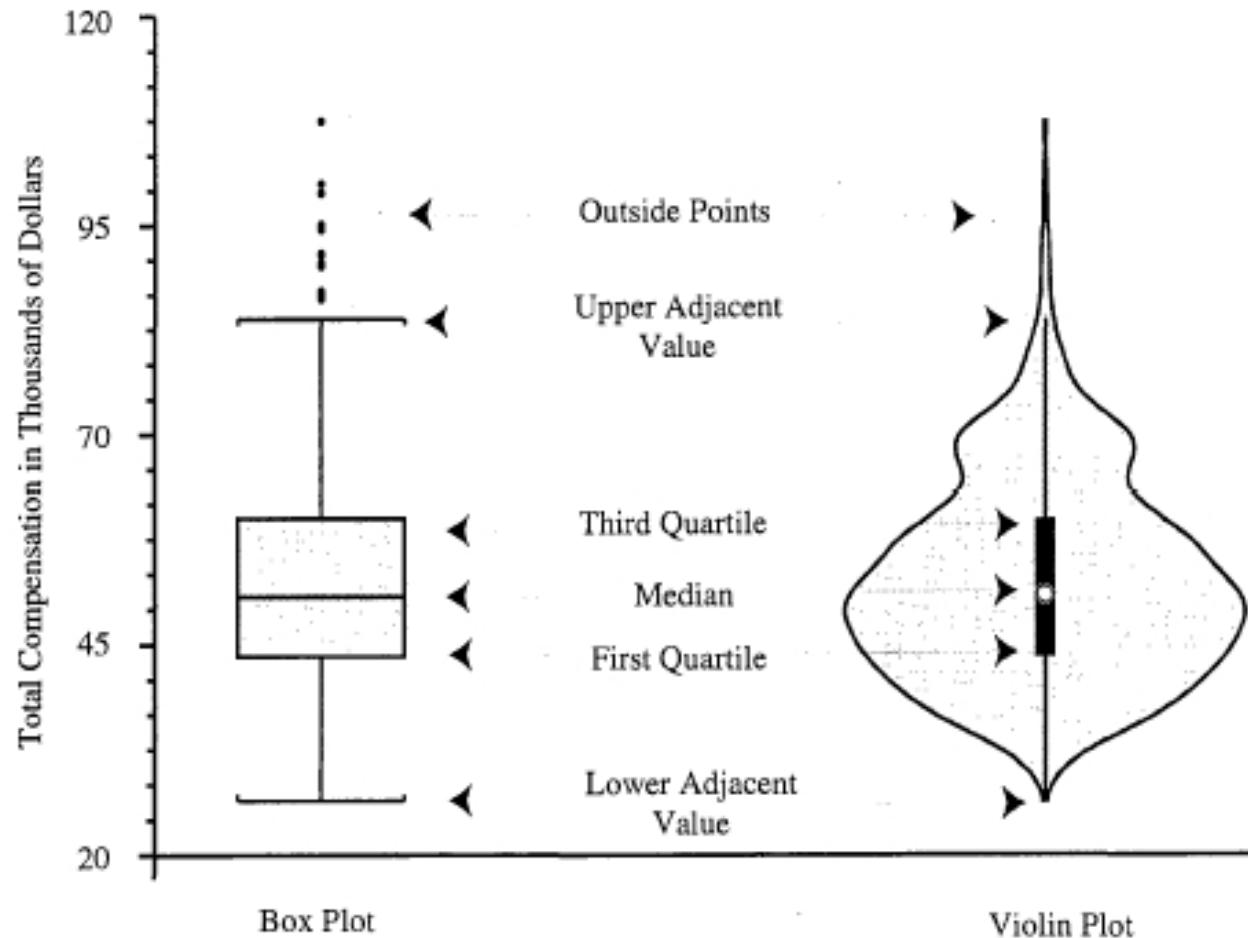
Boxplot (partitions) and distributions

Comparison of a boxplot of a nearly normal distribution and a probability density function (pdf) for a normal distribution

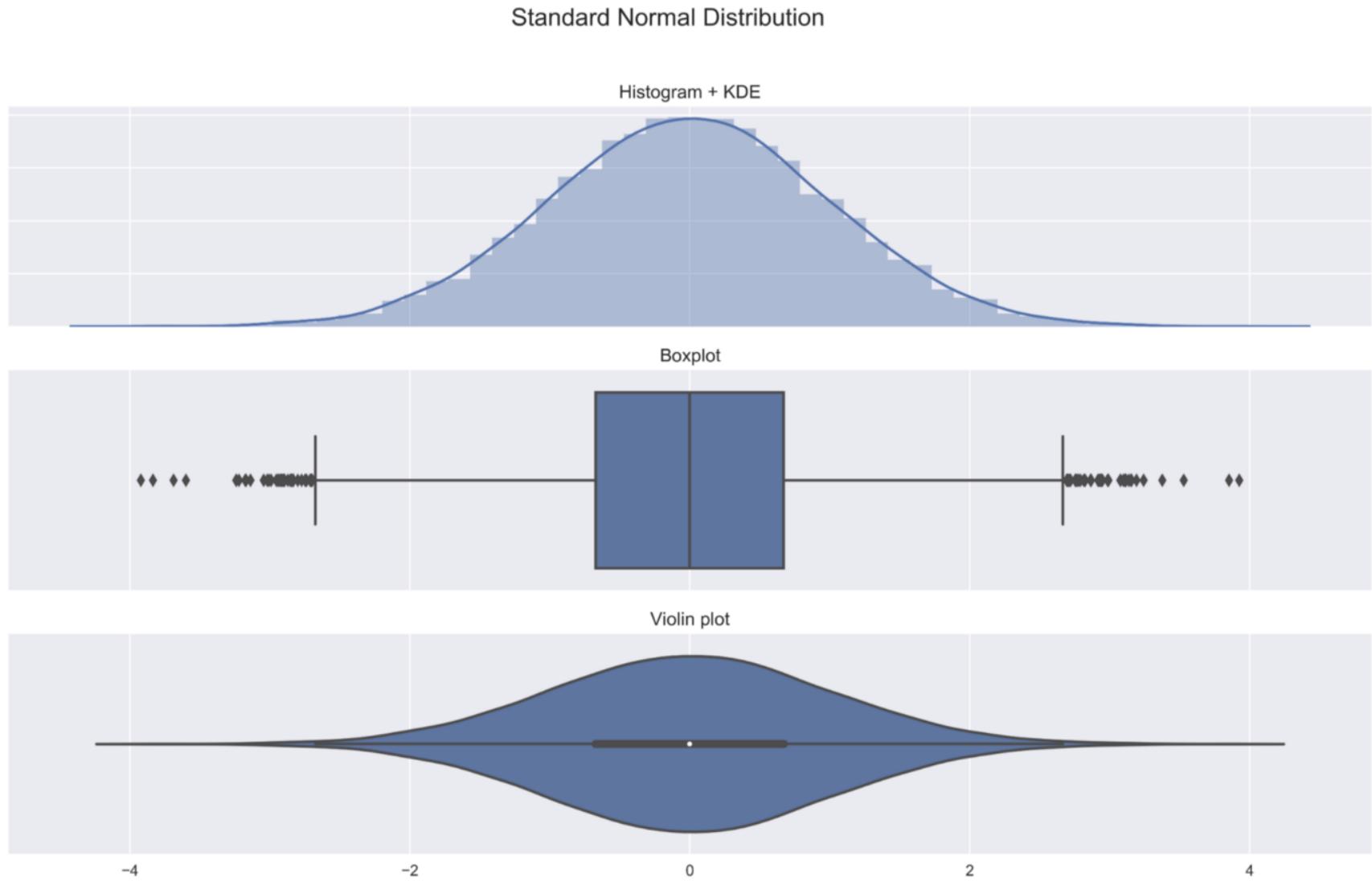


Boxplot and violin plots comparison

Common components of box plot and violin plot



Comparisons for a normal distribution 1/3

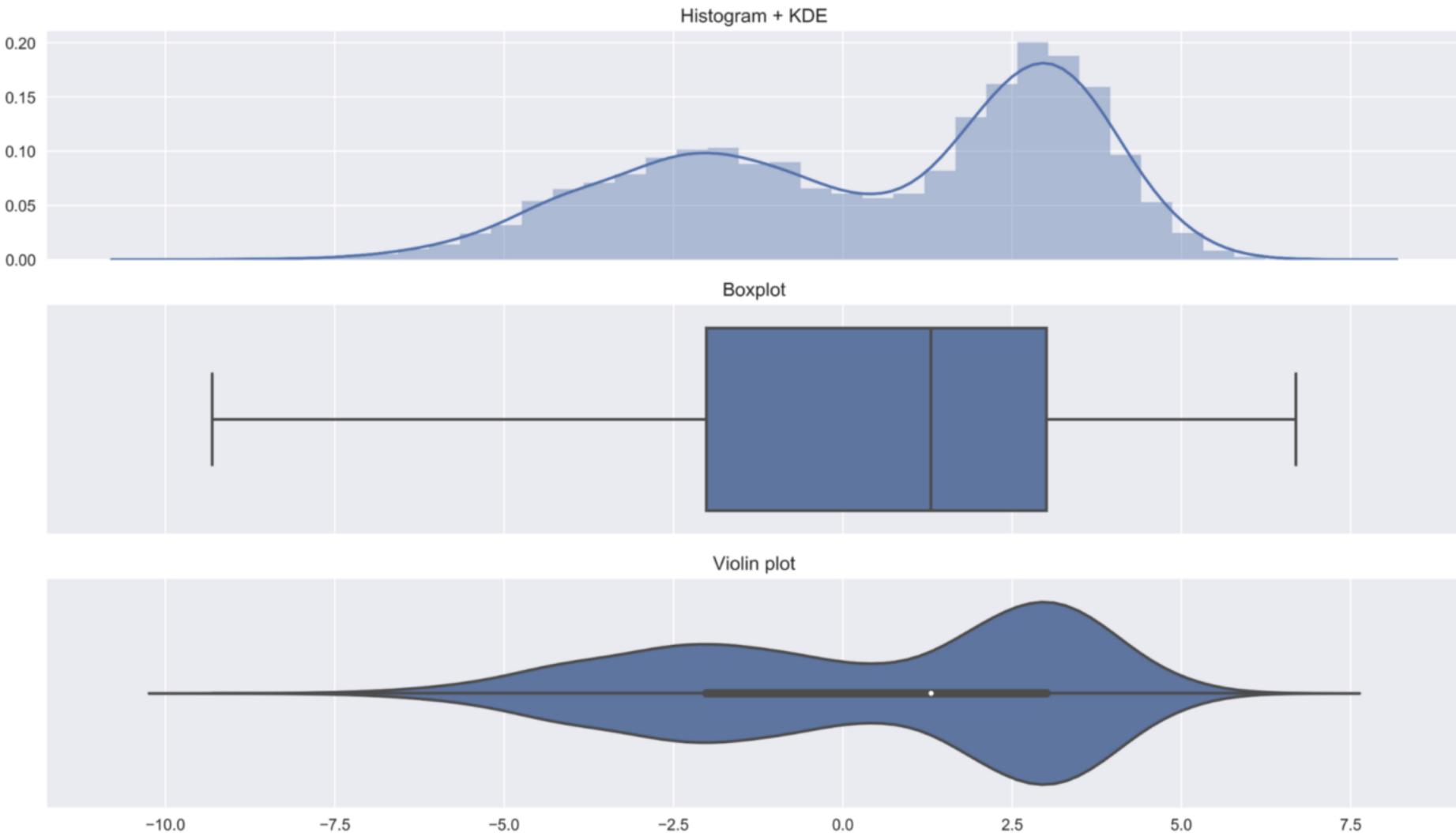


Comparisons for a log-normal distribution 2/3



Comparisons for a bimodal distribution 2/3

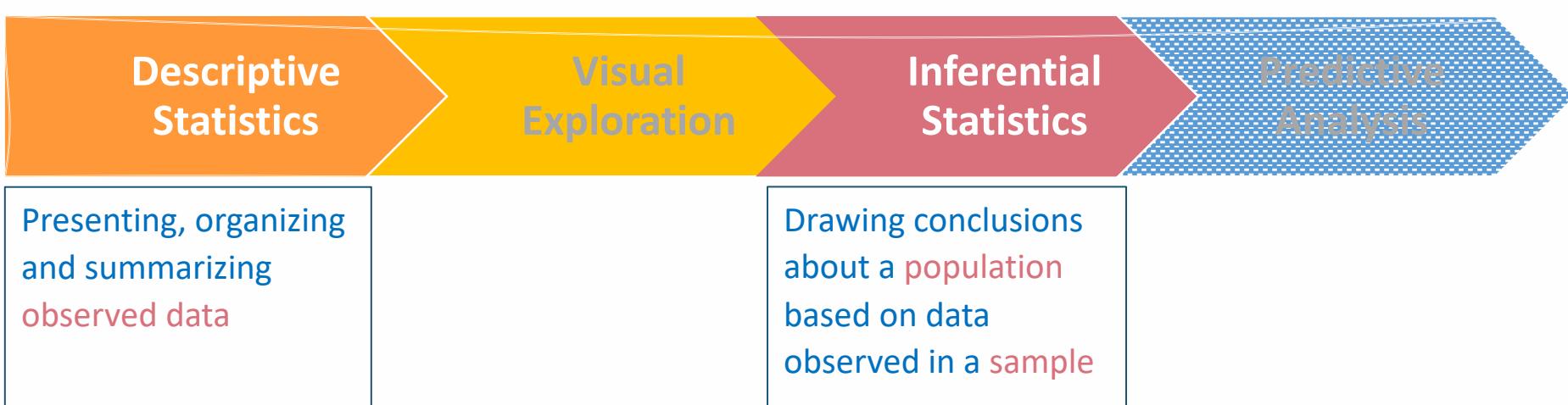
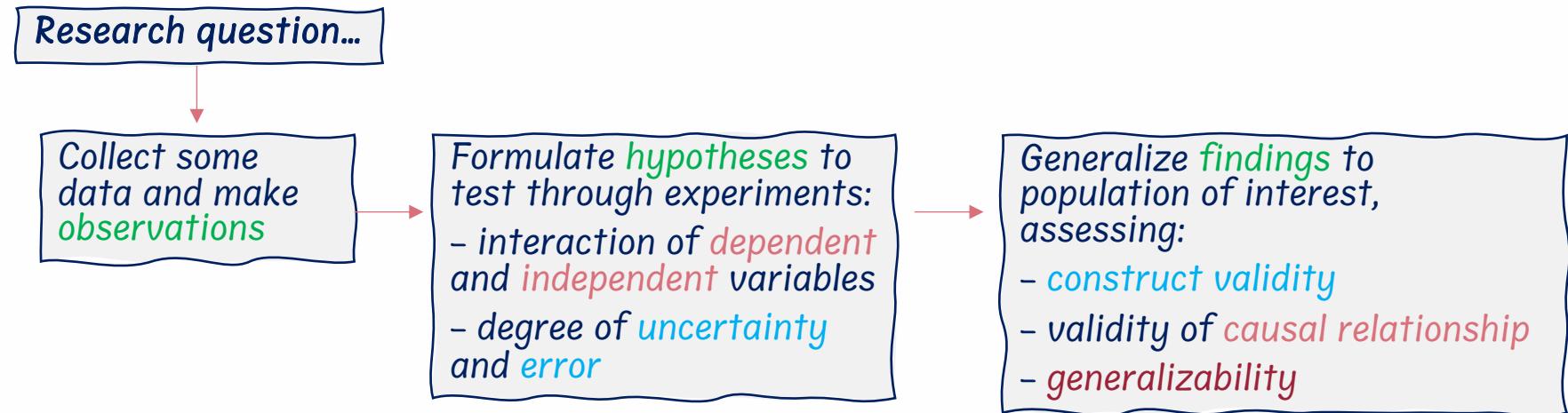
Mixture of Gaussians - bimodal



DAY 1 – LECTURE OUTLINE

- Introduction to R and R-studio
 - Why R?
 - Principles of reproducible analysis with R + RStudio
- R objects, functions, packages
- Understanding different types of variables
 - Principles of “tidy data”
 - Data cleaning and manipulation
- Descriptive statistics
 - measures of central tendency, measures of variability (or spread), and frequency distribution
- Visual data exploration
 - {ggplot2}
- Foundations of inference

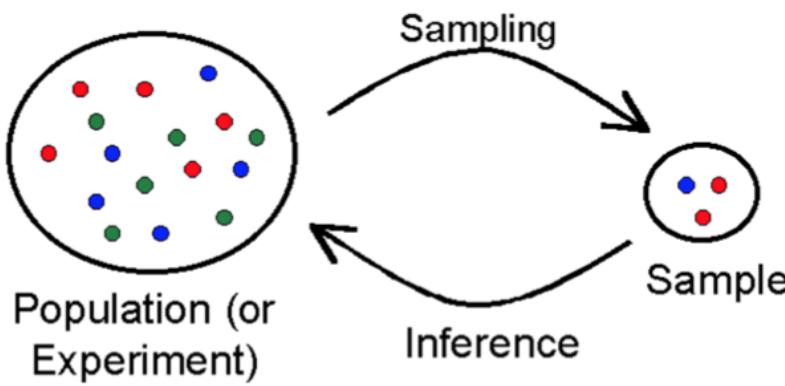
Statistics in (experimental) scientific research



Inferential Statistics: population and samples

Gathering all data is not always possible due to barriers like time, accessibility, or cost. Therefore, we often gather information from a smaller subset of the population: a **SAMPLE**.

- **POPULATION** = the universe of all possible observations we are interested in.
- **SAMPLE** = a subset of the population from which information is actually collected
- **INFERRENTIAL STATISTICS** = a collection of methods for using sample data to make conclusions about a population



Kinds of samples

- A **complete sample** is a set of objects from a parent population that includes ALL such objects that satisfy a set of well-defined selection criteria.

For example, a complete sample of Australian men taller than 2m would consist of a list of every Australian male taller than 2m. But it wouldn't include German males, or tall Australian females, or people shorter than 2m.

- An **unbiased (representative) sample** is a set of objects chosen from a complete sample using a selection process that does not depend on the properties of the objects.

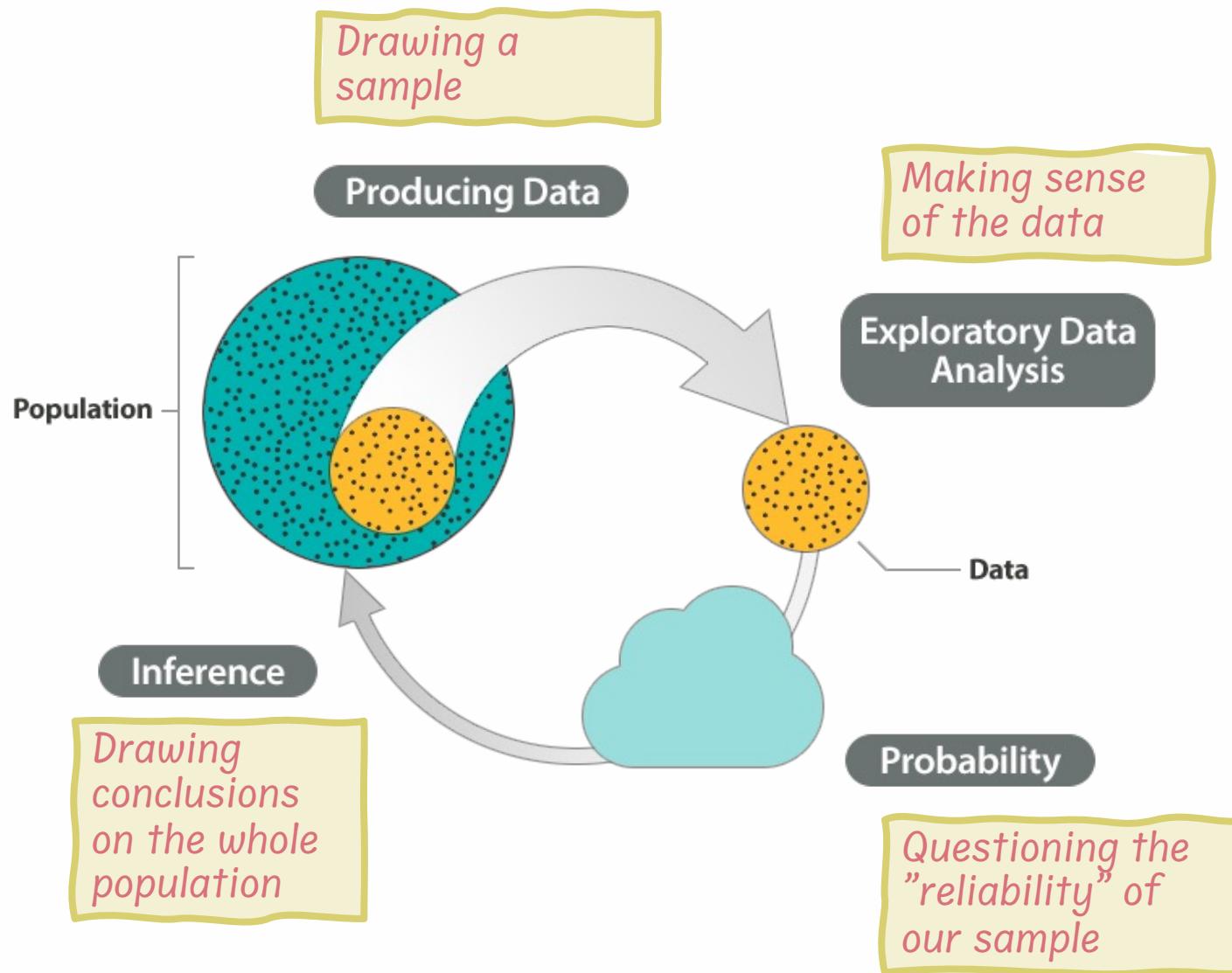
For example, an unbiased sample of Australian men taller than 2m might consist of a randomly sampled subset of 1% of Australian males taller than 2m. But one chosen from the electoral register might not be unbiased since, for example, males aged under 18 will not be on the electoral register.

Random sample

- The best way to avoid a biased or unrepresentative sample is to select a **random sample**.
- A **random sample** is a sample where each individual member of the population has a known, non-zero chance of being selected as part of the sample.

The types of random samples are **simple random samples**, **systematic samples**, **stratified random samples**, and **cluster random samples**.

The inferential statistics workflow



Source: https://stats.libretexts.org/Courses/Lumen_Learning/

Population versus sample: terms

Measurement	Sample statistic	Population parameter
Mean	\bar{x}	μ (mu)
Standard deviation	s	σ (sigma)
Variance	s^2	σ^2 (sigma squared)
Proportion	p	π (pi)
Correlation	r	ρ (rho)
Regression coefficient	b	β (beta)

Source: <https://www.statology.org/statistic-vs-parameter/>

Probability & statistical models

- Data are **observed values** («realizations») of **random variables**.
- The **probability distribution** of these random variables can be used to reason about *properties* of the unobserved universe (**inference**).
 - These properties are called «**parameters**»
- **Statistical models** are probability distributions for observable data constructed to enable inferences to be drawn or decisions made from data
- The relationship between Absolute Frequency and Probability is further reinforced by the **Law of Large Numbers**

Absolute frequency (af) = *N of favorable events (in a given set of data)*

Relative frequency (rf) = $\frac{N_{\text{favourable events}}}{N_{\text{observations}}}$

Probability of an event = $\frac{N_{\text{favourable events}}}{N_{\text{possible events}}}$ [always between 0 and 1]

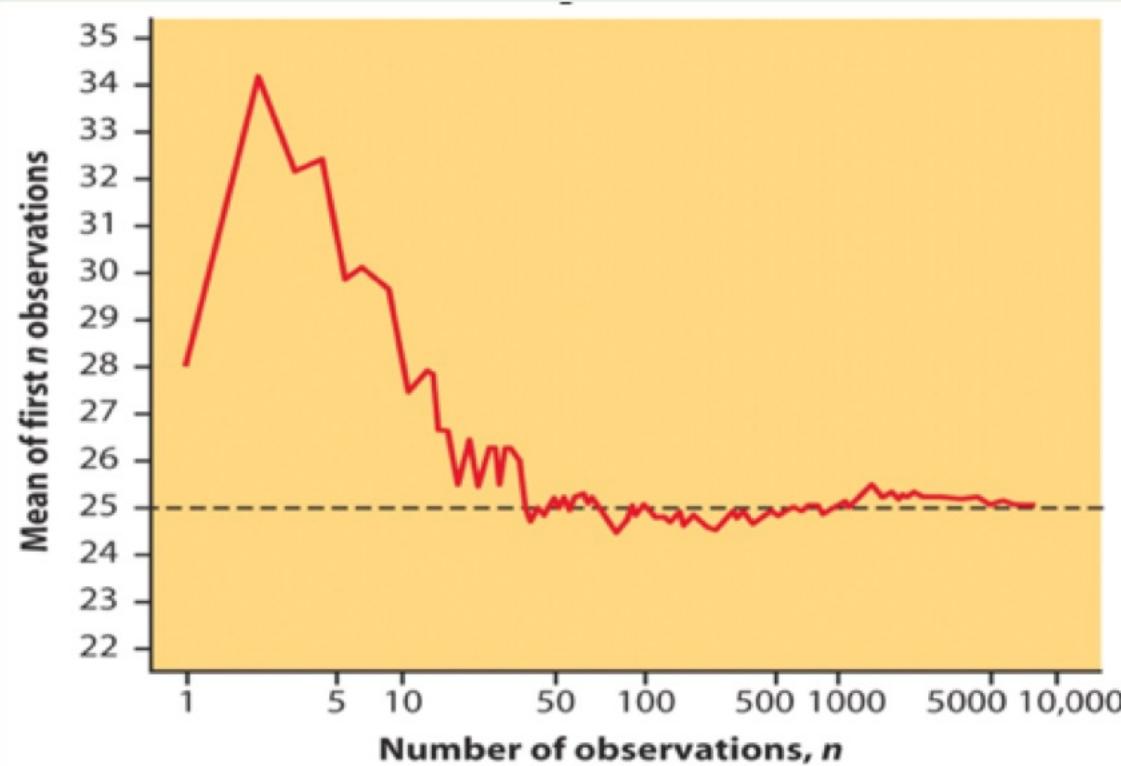
Relationship (af) and probability = $p = \lim_{\text{big } N} rf = \lim_{\text{big } N} \frac{\text{absolute frequency (af)}}{N}$

(By conducting a larger number of trials or observations)
we can derive probabilities from absolute frequency!

The Law of Large Numbers allows us to use probability to predict absolute frequency and viceversa

Draw observations at random from any population with finite mean μ . As the number of observations drawn increases, the sample mean of the observed values \bar{x} gets closer and closer to the mean μ of the population.

Example: How sample means approach the population mean ($\mu = 25$).



2 types of random variables

1. discrete random variables (values are countable)
2. continuous random variables (values are uncountable)

Probability distributions for a discrete random variable

- To visualize the probabilities derived from absolute frequency data, we can construct a probability distribution.
- A **probability distribution** displays the possible outcomes of an event along with their corresponding probabilities.
- The probability distribution of a casual variable (x_i) is a function
 - $p(x)=P(x=x_i)$
 - Where $p(x_i)$ is the probability to have x_i .

In medical research, probability and absolute frequency are utilized to study the effectiveness of treatments.

E.g. clinical trials conducted to estimate the **probability** of a certain treatment (new drug) to effectively cure a disease, gather the **absolute frequency**, i.e. the actual number of patients who experienced the desired outcome.

Frequency distribution of discrete variables

While a probability distribution for a **discrete random variable** tells us the likelihood that the random variable takes on certain values, a **frequency distribution** gives us a descriptive summary of the data

- Frequency table
- Frequency distribution

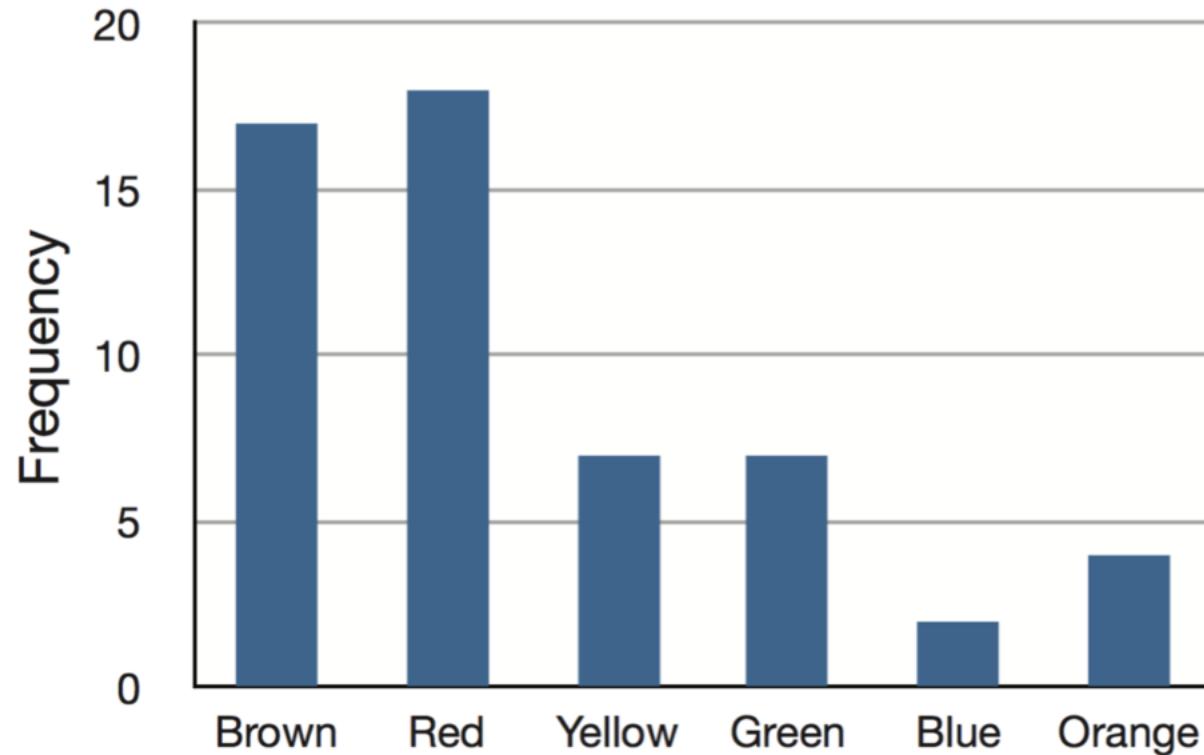


Color	Frequency
Brown	17
Red	18
Yellow	7
Green	7
Blue	2
Orange	4

Distribution of Discrete variables (2)

Absolute frequency (a direct count of occurrences) provides a descriptive summary of the data. It helps us understand the frequency distribution and identify patterns or trends.

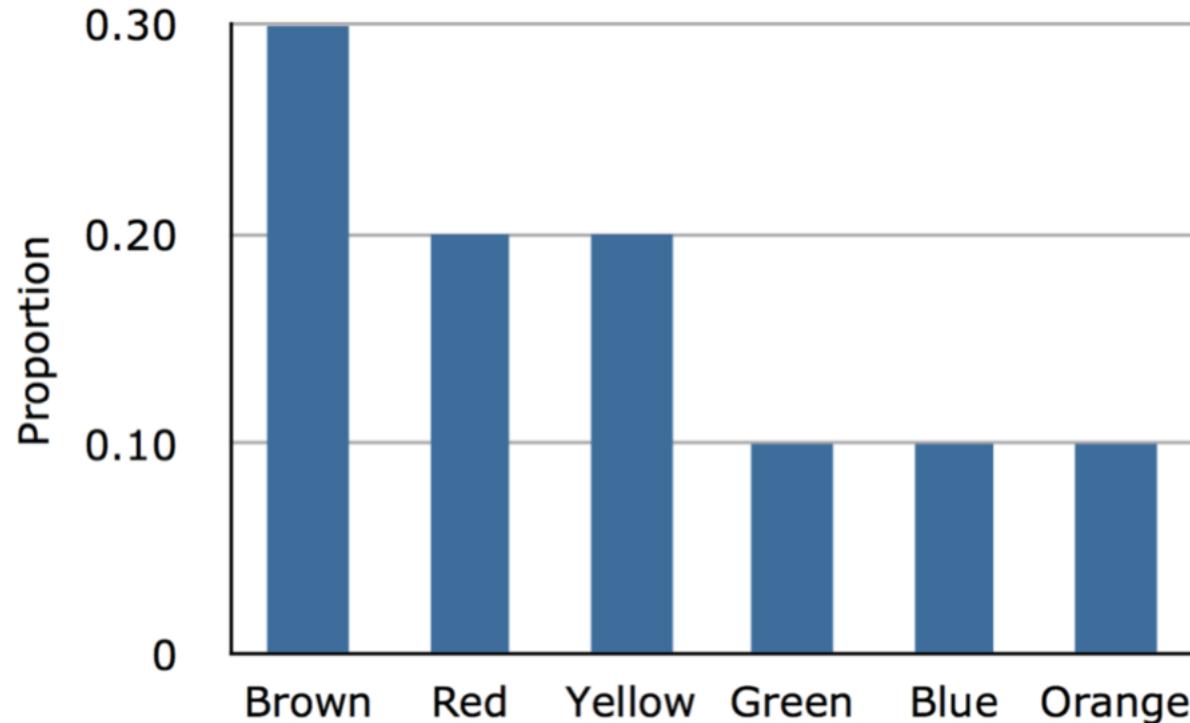
Frequency distribution is shown graphically



Distribution of Discrete variables (3)

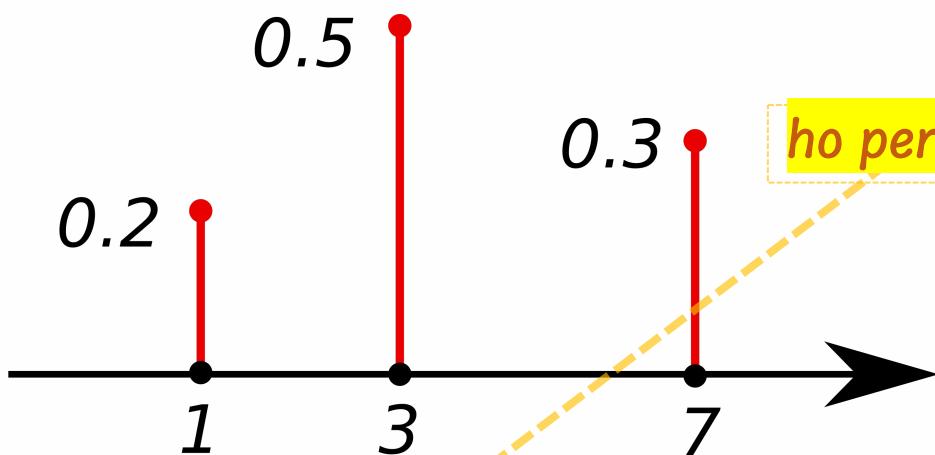
Probability (the likelihood of an event occurring among all possible outcomes) allows us to make predictions about the future events based on the available information.

Probability distribution



Discrete probability distribution

DISCRETE RANDOM VARIABLE \leftrightarrow PMF Probability Mass Function



$$p_X(x) = P(X = x)$$

ho perso il filo logico

$$\sum_x p_X(x) = 1$$

and

$$p_X(x) \geq 0.$$

PMF is a function that gives the probability that a discrete random variable is exactly equal to some value

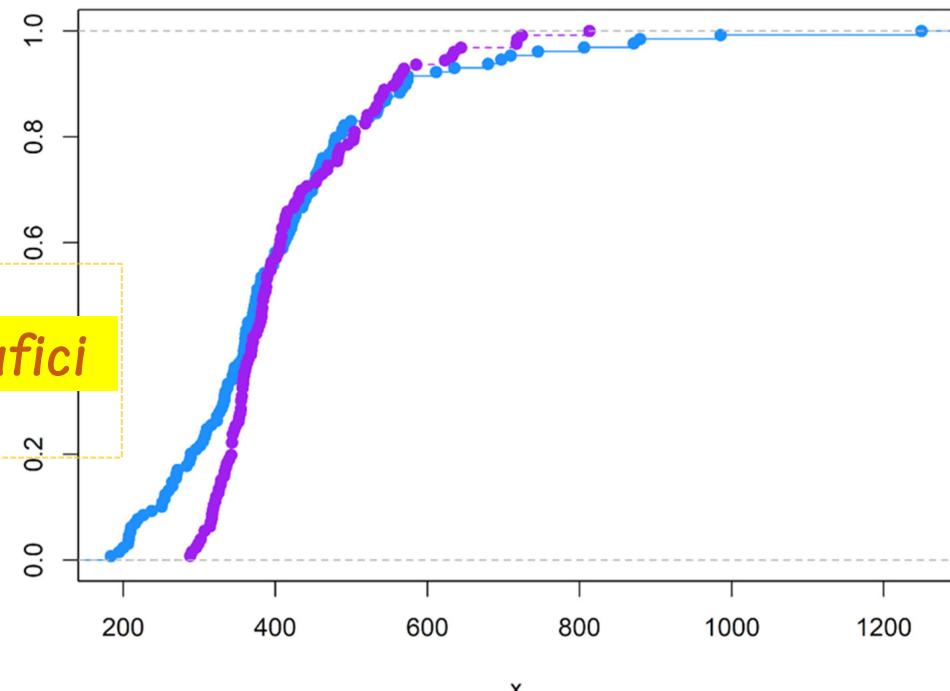
Probability distribution for a Continuous random Variables (1)...

- A **continuous random variable** is a variable which can take on an infinite number of possible values. Hence, reporting the frequency of each entry, is not an effective summary since most entries are unique.
 - Statistics textbooks define a function that reports the proportion of the data below for all possible values of α . This function is called the **empirical cumulative distribution function (eCDF)**, it can be plotted, and it provides a full description of the distribution of our data.

Example: "Response times"

568	720
577	728
581	729
640	
641	
645	
657	
673	865
696	875
703	1007

esempio.... forse si
potrebbe rifare grafici
con autism?



...Distribution of Continuous variables (2)

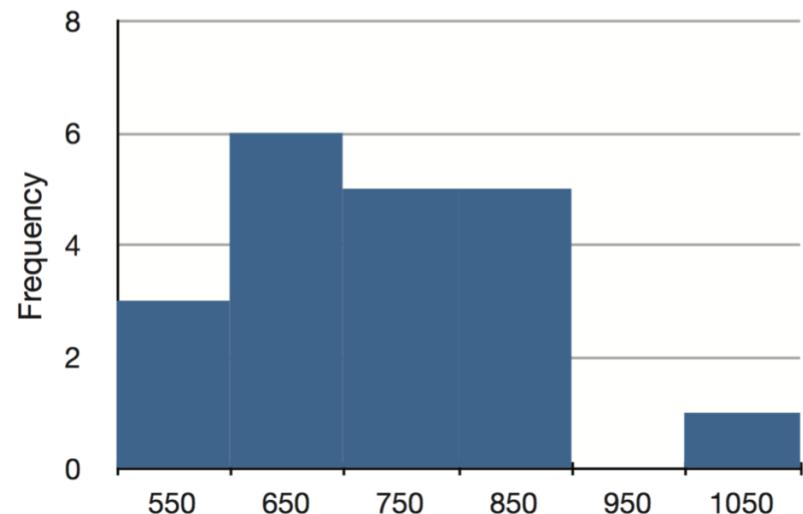
- However, the eCDF does not easily convey characteristics of interest of the distribution such as: measures of center, symmetry, quantiles...
- Hence **histograms** are used to tell us the probability that the variable takes on a **range of values**.
 - here we sacrifice a bit of information, for an easier interpretation
 - to make a histogram, we divide the span of data into non-overlapping **bins** of the same size. Then, for each bin, we count the number of values that fall in that interval

Grouped frequency distribution;

Range	Frequency
500-600	3
600-700	6
700-800	5
800-900	
900-1000	
1000-1100	

*esempio.... forse si
potrebbe rifare grafici
con autism?*

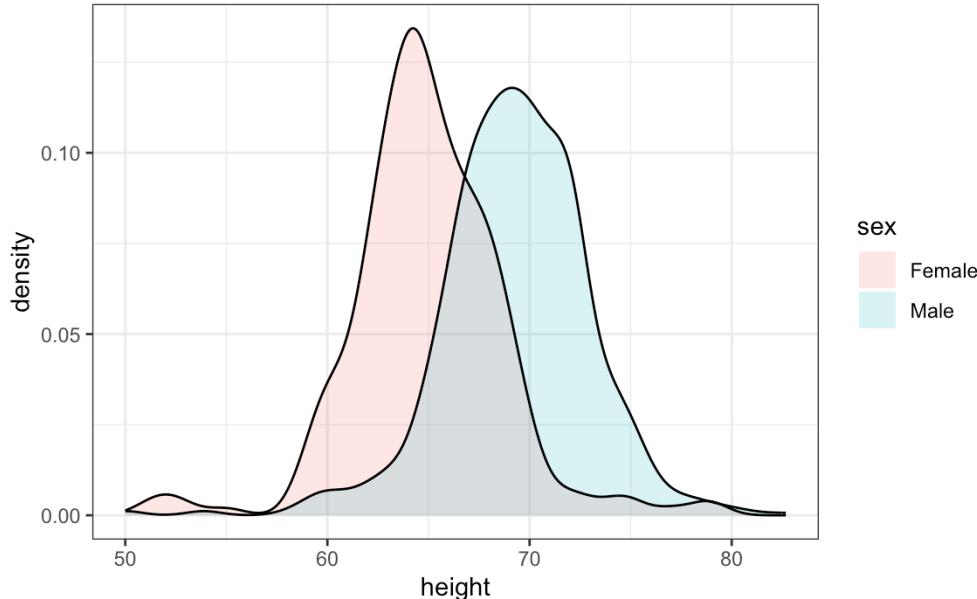
A histogram of the grouped frequency distribution:



- The labels on the X-axis are the middle values of the range they represent.

Probability density

- *Smooth density* plots are similar to histograms
 - the scale of the y-axis changed from counts to *density*
 - smooth densities make it easier to compare two distributions



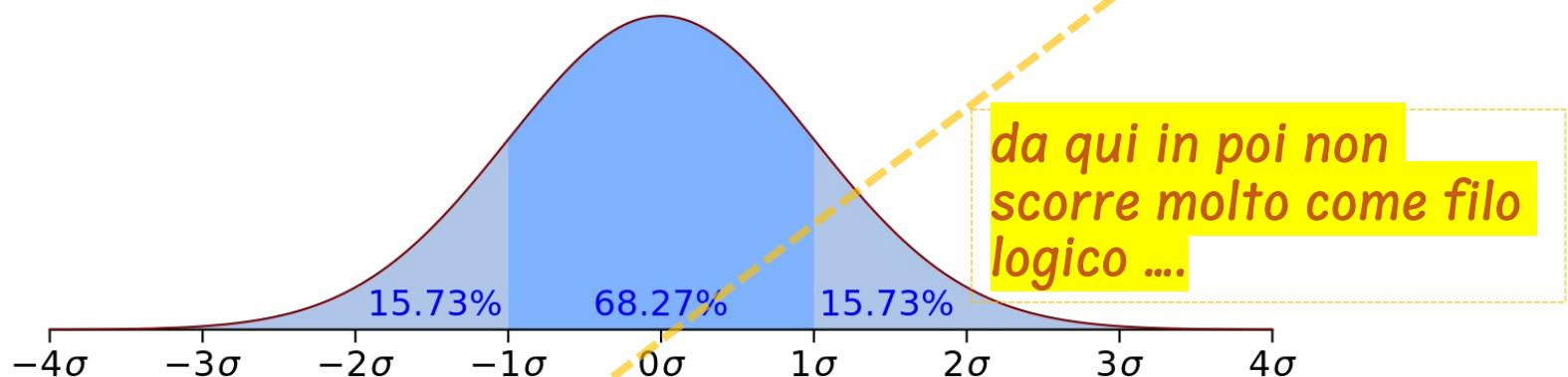
Probability Density Function:

- For a **discrete random variable**, a probability distribution contains the probability of each possible outcome.
- For a **continuous random variable**, the probability of any **one** outcome is zero (if you specify it to enough decimal places).
- A probability density function is a formula that can be used to compute probabilities of a range of outcomes for a continuous random variable.
- The sum of all densities is always 1.0 and the value of the function is always greater or equal to zero.

esempio.... forse si
potrebbe rifare grafici
con autism?

Continuous Probability Distributions

CONTINOUS RANDOM VARIABLE \leftrightarrow PDF Probability Density Function

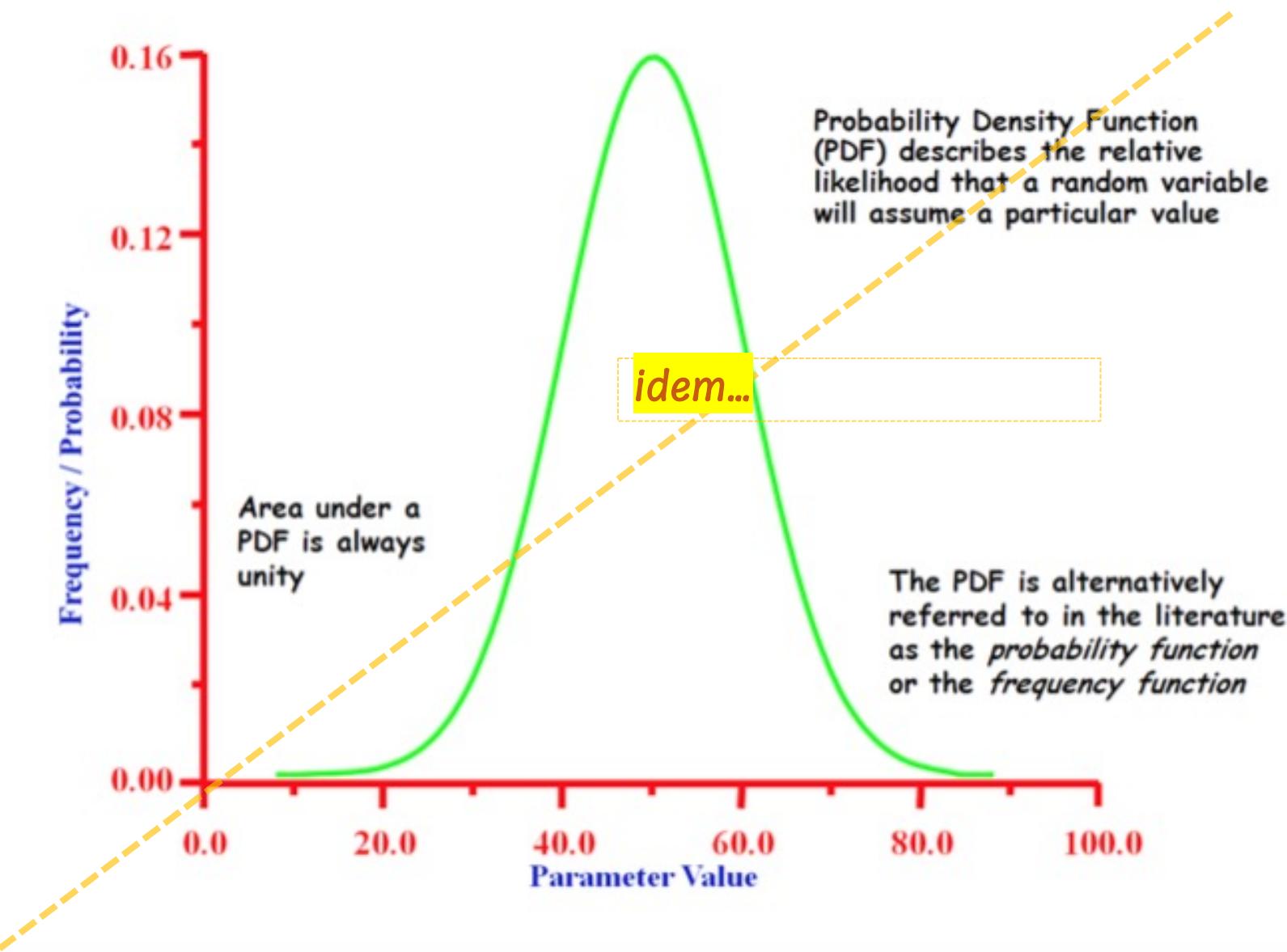


$$\Pr[a \leq X \leq b] = \int_a^b f_X(x) dx.$$

Pr = Probability
X = Variable
 f_X = PDF

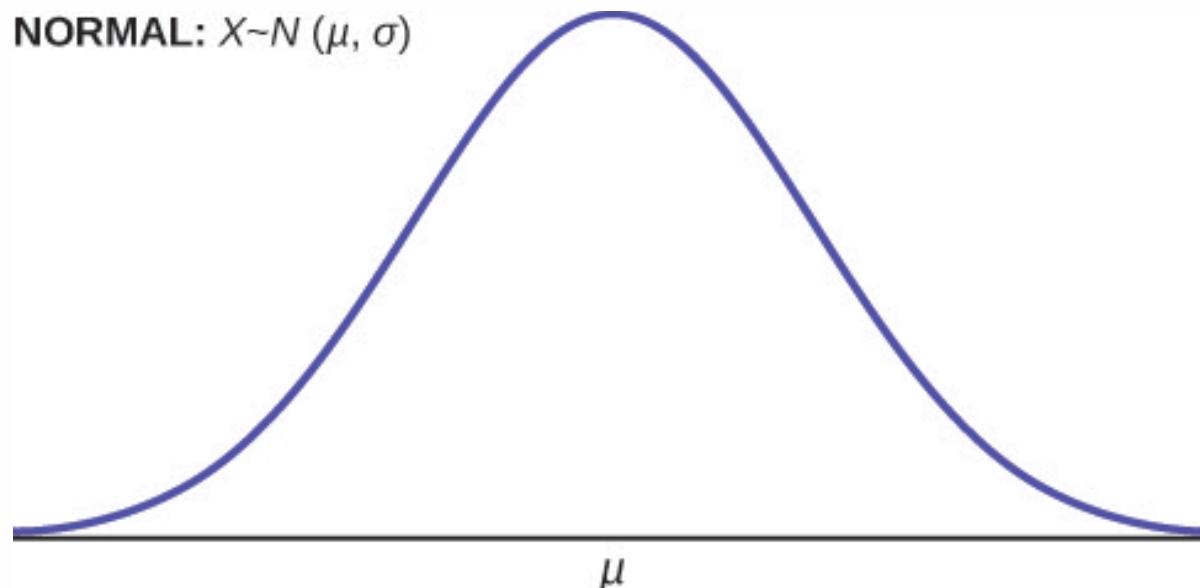
PDF gives the probability of a random variable falling *within a particular range of values* (as opposed to taking on any one value).
This probability is given by the integral of this variable's PDF over that range

PDF



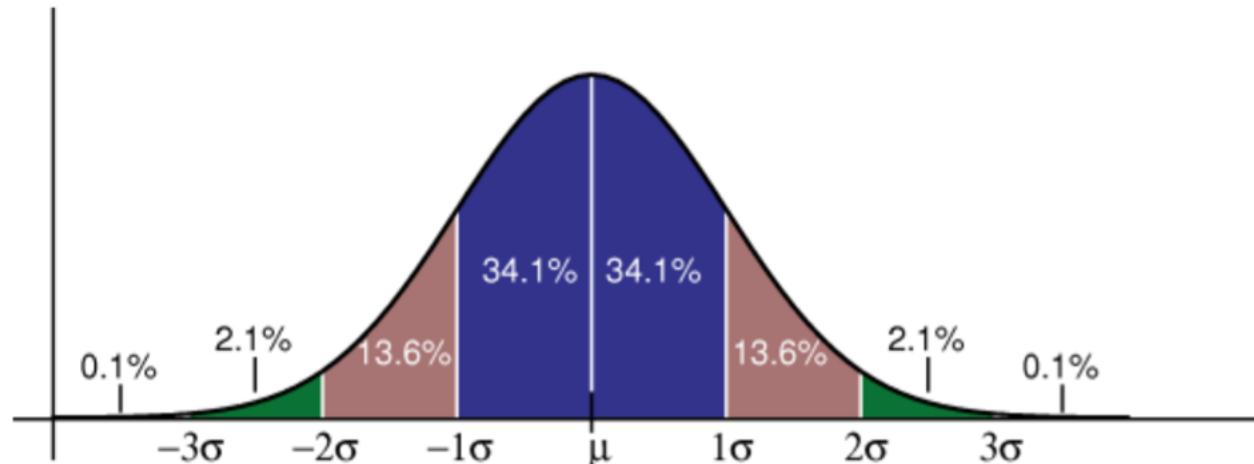
The normal distribution (special case of continuous distr.)

- The **normal distribution** («bell curve» or «Gaussian curve») is extremely important in statistics because it provides an excellent summary of an empirical distribution providing 2 parameters: the **mean (μ)** and **standard deviation (σ)**.
 - it is the most important and widely used distribution
 - many datasets can be approximated with it (e.g. blood pressure, height, weight, and experimental measurement errors)



Features of the normal distribution

- ① Normal distributions are symmetric around their mean.
- ② The mean, median, and mode of a normal distribution are equal.
- ③ The area under the normal curve is equal to 1.0.
- ④ Normal distributions are denser in the center and less dense in the tails.
- ⑤ Normal distributions are defined by two parameters, the mean (μ) and the standard deviation (σ).
- ⑥ 68% of the area of a normal distribution is within one standard deviation of the mean.
- ⑦ Approximately 95% of the area of a normal distribution is within two standard deviations of the mean



Central Limit Theorem

The normal distribution is important because of the following

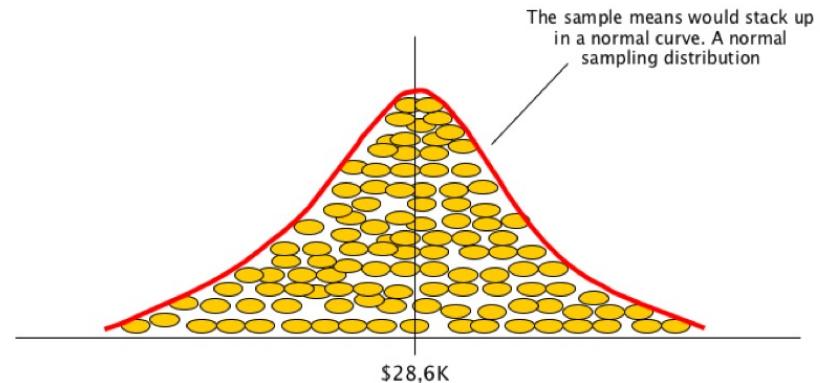
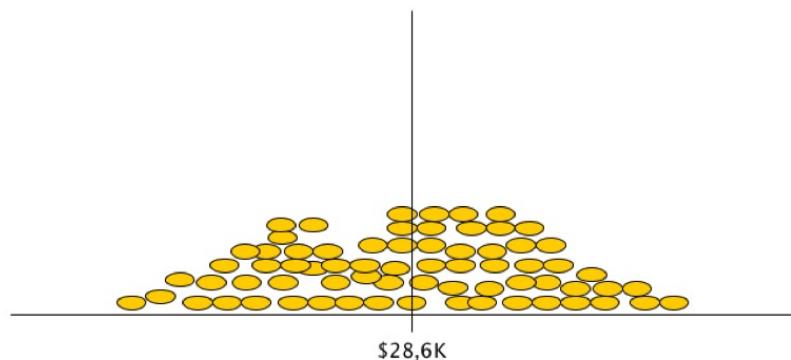
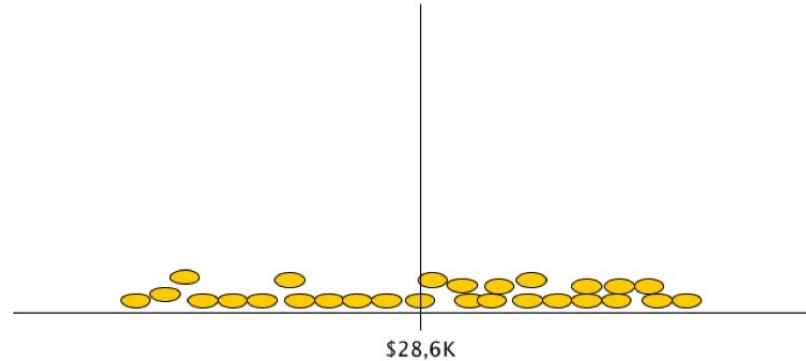
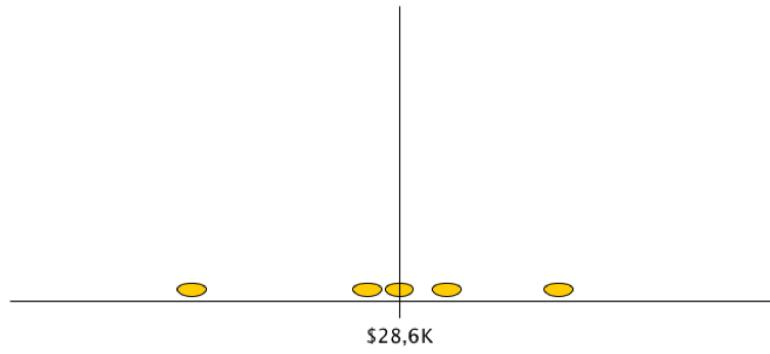
Theorem (Central Limit Theorem)

Given a population with a finite mean μ and a finite non-zero variance σ^2 , the sampling distribution of the mean approaches a normal distribution with a mean of μ and a variance of σ^2/N as N , the sample size, increases.

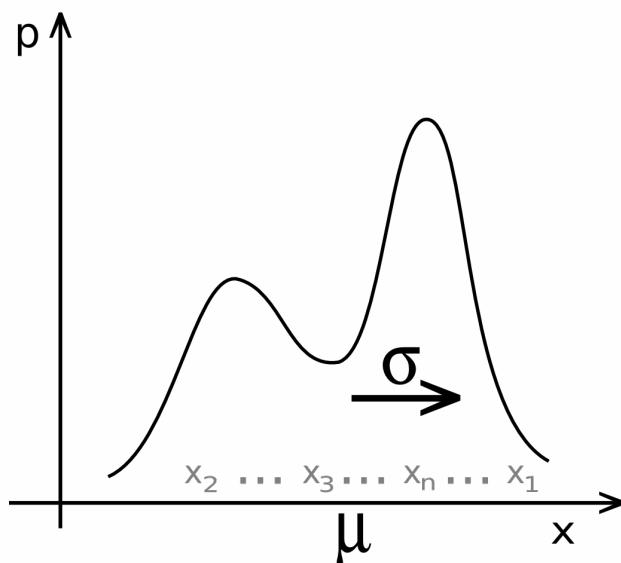
- Most statistical procedures for testing differences between means assume extremely important central limit theorem, the topic of a later section of this normal distributions. Because the distribution of means is very close to normal,
- Quetelet was the first to apply the normal distribution to human characteristics. He noted that characteristics such as height, weight, and strength were normally distributed.

CLT example

- Let's try create a sampling distribution of means...
- Take a sample of size 1,500 from the US. Record the mean income.
- US census says the per capita income in past 12 months (in 2014) has been \$28,555.

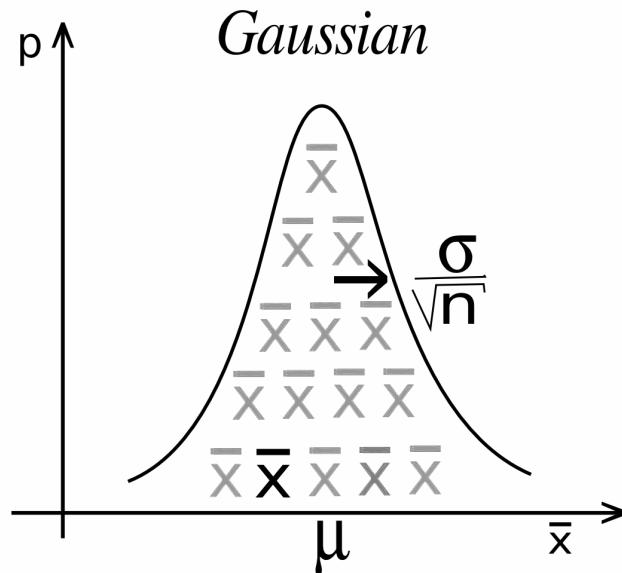


An important insight...



population distribution

samples of size n
 \bar{x}
 \bar{x}



sampling distribution of the mean

Even if the objective population has an unknown or non-normal distribution the mean of all possible casual samples with n sufficiently large ($n>30$) will have a normal distribution with mean μ and standard deviation σ/\sqrt{n} .

The big practical relevance of the central limit theorem is that it makes parametric test on sample means or its linear combination applicable even if the assumption of population normality is violated, provided that the sample dimension is $n>30$.

Confidence intervals and estimate precision

- A **confidence interval (CI)** is a range of values that is likely to contain a population parameter with a certain level of confidence.

Confidence Interval = point estimate +/- margin of error

$$CI = \bar{x} \pm \varepsilon_i$$

Where ε_i = (critical value)(sd of the statistic)

$$CI = \bar{x} \pm z * \frac{\sigma}{\sqrt{n}}$$

- The sample mean \bar{x} is a **point estimate** of μ
 - Each sample mean \bar{x}_i differs from the next one and from μ only by chance
 - The smaller ε_i , the more precise will be the sample mean \bar{x}_i
 - z is a critical value that depend upon a test statistic
-
- The **estimate precision** is evaluated with the CI, which is an interval with a lower bound and an upper bound, which likely contains a population parameter with a certain level of confidence

Confidence Intervals (CI) for a mean

CI with 95% confidence level

$$CI(95\%) = \bar{x} \pm z^* \frac{\sigma}{\sqrt{n}} = \bar{x} \pm 1.96 \frac{\sigma}{\sqrt{n}}$$

Interpretation:

- in 95% of all possible samples of size n , μ will fall in our confidence interval
- in 5% of all possible samples of size n , we would miss μ

The CI has a 95% probability of containing μ .

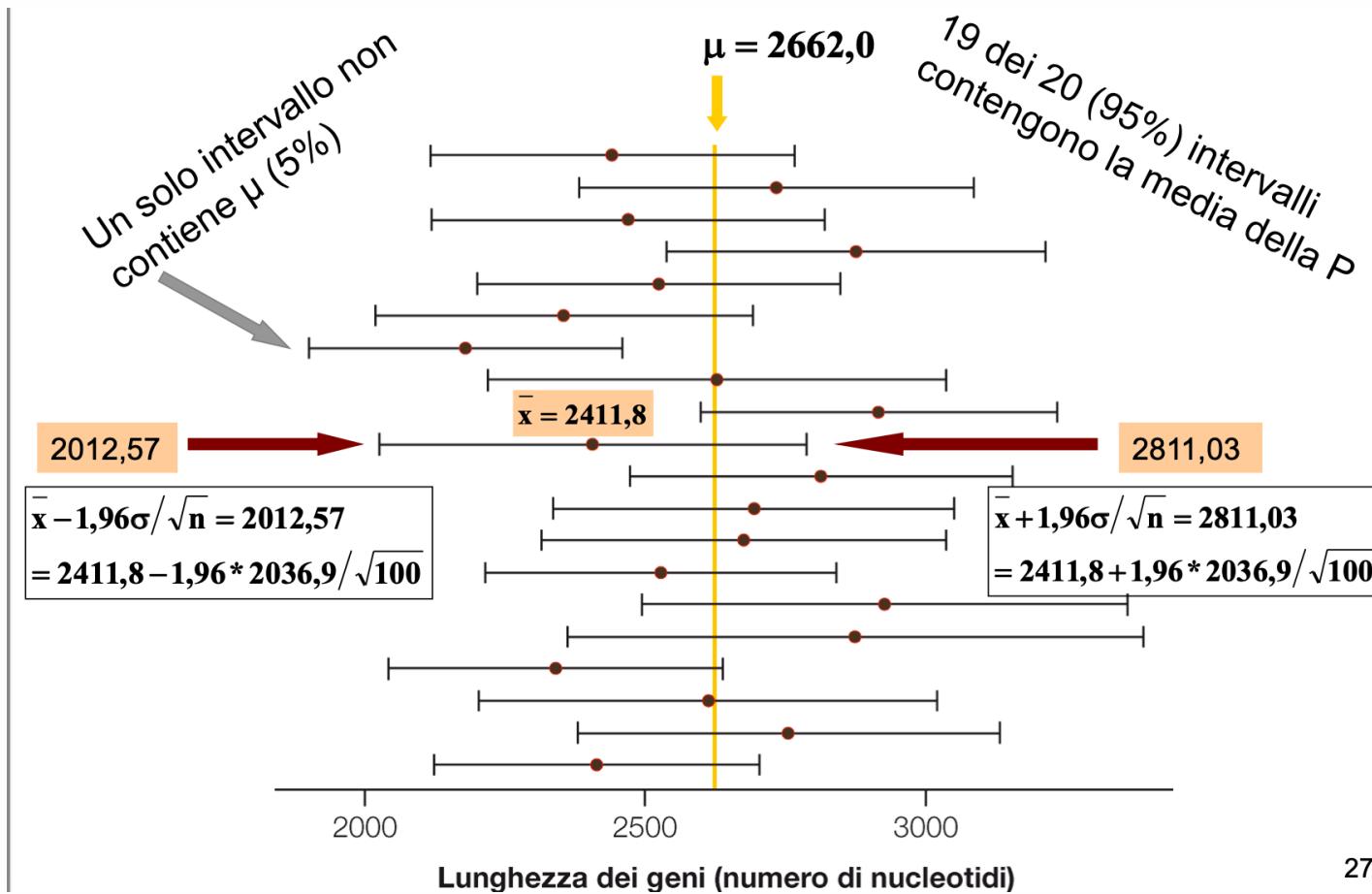
There is a **trade-off** between the level of confidence (z^*) and precision in which the parameter is estimated:

- higher level of confidence wider confidence interval
- lower level of confidence narrower confidence interval

Also, increasing n will reduce the margin of error ϵ for a fixed z^* .

CI intuition

- IC (95%) of the mean calculated from 20 random samples of $n = 100$ genes from the human genome



Degrees of freedom

The **number of degrees of freedom (df)** refers to the number of separate, relevant pieces of information that are available in the calculation of a statistic.

Where “n” is the number of items,

$$df = n - 1$$

Example

- The sample variance has $N - 1$ degrees of freedom;
- The estimate of the variance in a sample is:

$$s^2 = \frac{\sum(X - M)^2}{df} = \frac{\sum(X - M)^2}{N - 1}$$