# Visual exploration of data

# An introduction to PowerBI

## A report generator

- Power BI is a reporting and data analytics tool that simplifies data collection, transformation, and visualization tasks.
- It is designed to be extensible and has become the de facto standard in the industry.

## What PowerBI is not

- While it performs well in terms of data visualization, it is not a full-fledged statistical software.
- The use of a superior data transformation and analysis tool is a highly effective strategy.

i

# R

**The workhorse of statistics**

- A free and open source piece of software.
- R boasts implementation of nearly all developments in statistics.
- The number of free programs written in R is keeping increasing.

**A companion to PowerBI**

- PowerBI can collect data from a R script.
- PowerBI can transform data using a R script.
- PowerBI can display the output of a R script.

# R IN A NUTSHELL

## R as a desk calculator

- Open R by double-clicking on the R icon.
- Simply type ”2+3” and observe the result.

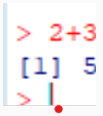


**Figure 1:** A simple sum

- A value can be stored in a variable with the operator $<-$.

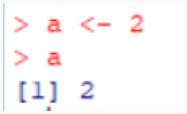


**Figure 2:** Using a variable.

# VECTORS

## Vectors

- Vectors are created using the ”c” (concatenate) or range operator ”min:max”.
- Try ”c(0.1, 2, 3)” and ”1:100”.
- If you want a non-unit step, use the ”seq(min, max, step)” command.

```
> seq(0,1,0.1)
 [1] 0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0
```

**Figure 3:** User-defined step

- All the elements of a vector must be of the same type: numeric (double, integer, complex), logical, string.

# VECTORS

## Numeric and boolean types

- By default, a numeric value is encoded as a double precision number on 64 bits.
- A complex number is declared with syntax $x + yi$:

```
> 2+3i
[1] 2+3i
> (2+3i)*(1+2i)
[1] -4+7i
```

**Figure 4:** Complex numbers

- Booleans are TRUE or FALSE.

```
> c(TRUE, TRUE,FALSE, FALSE)
[1]  TRUE  TRUE FALSE FALSE
```

**Figure 5:** Booleans

# VECTORS

## Character strings

- A string is enclosed with double or simple quotes.

```
> c("hello",', world')
[1] "hello"   ", world"
```

**Figure 6:** Character strings

- Any R object can be converted to an informative string with the function `str()`.

```
> str(a)
 logi [1:4] TRUE TRUE FALSE FALSE
```

**Figure 7:** Str function.

# VECTORS

## Operators

- Operators and functions are applied elementwise.

```
> a <- 1:10
> b <- 1:10
> a+b
 [1]  2  4  6  8 10 12 14 16 18 20
> a*b
 [1]   1   4   9  16  25  36  49  64  81 100
> sin(a)
 [1]  0.8414710  0.9092974  0.1411200 -0.7568025 -0.9589243 -0.2794155
 [7]  0.6569866  0.9893582  0.4121185 -0.5440211
```

**Figure 8:** Operating on vectors.

- This is true also for comparison operators.

```
> c(1,3,5,8) <= c(0,4,5,8)
[1] FALSE  TRUE  TRUE  TRUE
```

**Figure 9:** Comparison operators.

# VECTORS

## Accessing elements

- An element in a vector can be referred to by its index.

```
> a<-1:10
> a[1]
[1] 1
> a[2:4]
[1] 2 3 4
> a[c(2,6,8)]
[1] 2 6 8
```

**Figure 10:** Accessing elements.

- A selection by booleans is also possible.

```
> a[c(TRUE, FALSE, TRUE)]
[1]  1  3  4  6  7  9 10
> a[a >= 5]
[1]  5  6  7  8  9 10
```

**Figure 11:** Boolean selection.

# Data frames

**A convenient object.**

- Data frames are arrays holding observed values.
- Observed characteristics are in columns, observations in rows.
- The columns are named and can be referred to by their names.

```
> dataset <- data.frame(id=1:100,val=0:0.01:99)
> str(dataset["id"])
'data.frame':   100 obs. of  1 variable:
 $ id: int  1 2 3 4 5 6 7 8 9 10 ...
> str(dataset$val)
 int [1:100] 0 1 2 3 4 5 6 7 8 9 ...
```

**Figure 12:** A data frame.

- Data frames are used by PowerBI to communicate with R.

# PASSENGERS DATASET

## Data collection

- The dataset, collected from the site `https://data.europa.eu/data/datasets?locale=en` is stored in the file `avia_paoc_monthly.csv` that can be retrieved on ecampus in the folder `Datasets`.
- Launch PowerBI and select `Get data from other sources`.
- Select `Text/CSV` and open the file.
- A table view of the dataset appears.

# PASSENGERS DATASET

## Data transformation

- PowerBI offers the opportunity to transform the data.

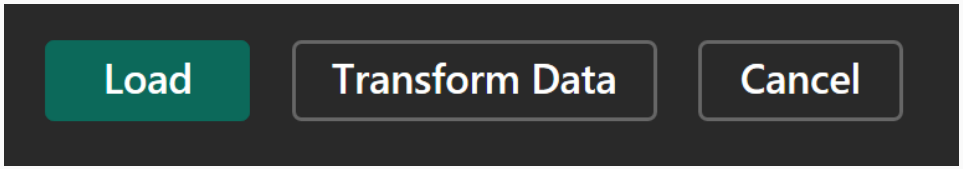


**Figure 13:** Data transform.

- We are only interested in the total number of passengers carried by country, so all the rows below row 32 must be removed.
- Select `Keep rows`, then `Keep top rows` and finally enter 32.

# Passengers dataset

## Final transformations

- In the first column, a description of the data includes at the end a geographic information.
- Select the first column, right click on it, the select `replace values`.
- Copy all the text in the first cell, except the final two-characters country code.
- Keep the replacement text empty and click `replace`.

# Passengers dataset

**Transposing datasets**

- Datasets must have their values in columns.
- Select the tab `Transform`, then `Transpose`.
- Return to the `Home` tab and select `Use first row as headers`.
- The label of the first column is not very informative. Right click on it and select `Rename`.

# PASSENGERS DATASET

## Cleaning data

- A dataset may have blank rows. Check yours and remove them if needed (`Remove rows` tab).
- Some values may be missing. In this dataset, they are coded as ”:”.
- The column ”MK” has such values. Right click on it, then use `Replace values` to change ”:” by ”0”.
- finally, change the type of the column to `Whole number`.

## Applying transforms

- Select the `Close and apply` to apply the transforms.
- It is time to save your work (tab `File`) !

# PASSENGERS DATASET

## Visual exploration of the dataset

- Return to the `Home` tab, and start populating the report.
- Text boxes or visual can be added.
- Select a text box and enter the text you wish to start your report.
- Select `New visual`. A new box appears (by default a bar chart).
- Select `Date` in the `Data` right pane and drag the `Month` field to the visual. This defines the ”x” axis.
- You can add data to your visual by selecting columns in the `Data` pane or by dragging them.
- Try changing the visual and see what happens.

# PASSENGERS DATASET

## Data aggregation

- In the ”passengers” dataset, there is a one-to-one mapping between a month and the number of passengers.
- What happens if we decide to plot against the quarter ?
- Uncheck `Month` in `Date` on the right pane and select `Quarter`.
- The visual now displays the total number of passengers for each quarter.
- In the `Y axis` part of the `visualization` pane, try to change `Sum` to `Average`.
- The aggregation of values is an important feature in PowerBI.

# Passengers dataset

## Measures

- A measure is a value that can be computed from the dataset.
- It may be viewed as a practical implementation of a random variable, a concept introduced later in the course.
- Since measures are computed on-the-fly, they always reflect the most up-to-date dataset.
- Using them is a must in PowerBI.
- Create a `New quick measure` on the ”passengers” dataset that sums the amount of carried passengers by quarter for a given country (select the one you prefer).
- Use a visual to plot the result.
- Custom measures can be used, provided you know the ”DAX” language.

# PowerBI and R

## R as a data source

- Open the `Get data` tab, then select `R`.
- In the script box, type:

```
1 dataset <- data.frame(x=1:100,y=runif(n=100,min=0,max=10))
```

- Observe the result in PowerBI. What happens if the name `dataset` for the ouput variable is changed ?
- Create two data frames in the R script: what happens in PowerBI?

# POWERBI AND R

## R as a data processor

- Open the `Get data` tab, then select `Run R script`.
- In the script box, type:

```
1 output <- dataset
2 output$z <- as.numeric(dataset$y)+1
```

- A new column named `z` was added and its values are those of column `y` plus 1.

# PowerBI and R

## Visuals

- On the `visualizations` pane, select the `R` icon.
- Drag the `y` field in the visual.
- In the script box, type:

```
1 hist(dataset$y)
```

- An histogram is displayed in the visual.
- This may be useful when a given visual, like the histogram, is not implemented in PowerBI or requires a plugin.