[Support] Training Course : First steps with R software

Data Center of University of Brittany

Paul Pinard

Basic :

- **Starting up a new project :**
  - Session > Set Working directory > Choose Directory
  - Tips : Save both your R file and your dataset in the same place !
- **Script Type :**
  - Classic R file : **File > New File > R Script**
  - Others possibilities : **Markdown, Quarto**
    - *A bit more complex for first timer but will have many more tools to offer and ways to make life easier for the user.*
    - *Shortcut to create a code zone :* **Ctrl + Alt + i (keep Ctrl + Alt and after smash your i button)**

- **Operator to create objects :** « **<-** »

- **How to import a dataset ?**
  - By using the « button » way :
    - **File > Import Dataset >**
      - **From Text (base)** (if you have an CSV file)
      - **From Excel** (If you have an Excel file)
    - **COPY / PASTE the small part of code in the small window which just opened**
  - By using directly some code :
    - CSV File :
      - `read.csv2("your_file_name.csv")`
      - Possible to add more parameters in the function *(for ex : sep = " ; ")* or *(header = TRUE)* if there is an error message or the dataset doesn't look right.
    - Excel File :
      - `library(readxl)`
      - `read_excel("your_file_name.xlsx")`
      - Need to import a « Extension » / « Library » because there is no way to read an excel file with basic R
- **How to see your dataset :**
  - View your file :
    - Possible with both types of R file (Markdown & R file) : **View(your_dataset)**

        - If Markdown file, another possibility :
            - Highlight by double clicking you gave the object of your dataset
            - Ctrl + Enter
    - Overview of your dataset types variables :
        - Same thing for both file : **str(your_dataset_name)**
    - View every names of your dataset :
        - Same thing : **names(your_dataset_name)**
    - View the number of individuals and variables of the dataset :
        - Same thing : **dim(your_dataset_name)**
    - *Small tips for the mardown file* **:**
        - **From the viewer, the number of individuals, variables and types of variables can be seen by just looking at the table !**

Variables Types :

- Numeric (num), Double (dbl) or Integer (Int) : numerical variables (Age, number of kids, etc…)
- Factor (fct) : categorial variable (Gender, Levels of education, ''Own a car ?'', etc…)
- Character (char) : Strings (Oppened answer from a survey)
- Logical (lgl) : Variable with only TRUE/FALSE/NAs

In your dataset, the most useful variables would be the numerical and the categoricals ones. We can use also character variables but less often.

Nota Bene : Carefull about Upper / Lower cases !!!

Simple code to start modifying your dataset :

- « $ » sign to call a variable : **your_dataset $your_variable_name**
- View every information about :
    - The first row : **your_dataset [1 ,]**
    - From the first row to the 10th : **your_dataset [c(1 :10) ,]**
    - From the first column to the 10th : **your_dataset [, c(1:10)]**
- Remove a variable :
    - **your_dataset$variable <- NULL**
- Rename a variable :
    - **names(dataset)[names(dataset) == "old_name"] <- "new_name"**
- Go from a variable :
    - Character / Numeric to a factor :
        - **your_dataset$variable <- as.factor(your_dataset$variable)**
    - Factor to Character :
        - **your_dataset$variable <- as. character(your_dataset$variable)**
    - Factor to Numeric :
        - **your_dataset$variable <- as.numeric(as.character(your_dataset$variable))**

- Change a modality in a variable :
  - A mistake inside your dataset :
    - **your_dataset[your_dataset$var == "hOMmE" ,]$var <- "Homme"**
  - Say as « NA » (« Not Available ») :
    - **your_dataset[your_dataset $Var %in% c("NA","Na","N/A"),]$Var <- NA**
- Create a new variable :
  - By using random values from a function
    - **your_dataset$Age <- sample(size = 100, x = c(10 :70), replace = TRUE)**
  - By using others columns :
    - **your_dataset$Minor <- NA**
    - **your_dataset [your_dataset $Age <= 18 ,]$Minor <- "Yes"**
    - **your_dataset [your_dataset $Age > 18 ,]$Minor <- "No"**

Simple functions for Analysis :

- Get the summary of your dataset :
  - **summary(your_dataset)**
  - Possible **Summary(your_dataset$Age)**
  - *Mostly useful for numerical variables*
- Get the mean of a numerical variable :
  - **mean(your_dataset$Age, na.rm = TRUE)**
  - **"***na.rm = TRUE* **"** *allow that even if there is missing values inside th column, then the average will not take it into account.*
- Get the median of the numerical variables :
  - **median(your_dataset$Age, na.rm = TRUE)**
- Get the smallest value of a numerical variable :
  - **min(your_dataset$Age, na.rm = TRUE)**
- Get the highest value of a numerical variable :
  - **max(your_dataset$Age, na.rm = TRUE)**
- Get the standard deviation of a numerical variable :
  - **sd(your_dataset$Age, na.rm = TRUE)**
- Know the distribution of different categories (Flat sorting) :
  - **table(your_dataset$Var_Character)**
- Know the distribution according to a second variable (Cross Sort) :
  - **table(your_dataset$Var_Char, your_dataset$Var_hab)**

Graphics :

Note : Each parameter added in the Histogram graph can also be transferred to the others, graphics and you do not need every parameter to make the graph work.

- Histogram :
  - **hist(your_dataset$Age, main = "Title of the graph", xlab = "Name Axe X", ylab = "Name Axe Y", col = "blue")**
- Boxplots :
  - **boxplot(your_dataset$Age)**
- Pie Chart :
  - **pie(your_dataset$Age)**
- Bar Chart :
  - **barplot(your_dataset$Age)**

BONUS :

- Cross table view with percentage :
  - `library(sjPlot)`
  - **sjt.xtab(var.row = bdd$Etat_Mineur, var.col = bdd$Age_participant, show.row.prc = TRUE, show.col.prc = TRUE)**

Additional ressources :

- ✓ HUSSON, François, CORNILLON, Pierre-André, GUYADER, Arnaud, et al. R pour la statistique et la science des données. Presses universitaires de Rennes, 2018.
- ✓ https://www.book.utilitr.org/