[Support] Formation : Premiers pas avec le logiciel R

Plateforme universitaire de données de Bretagne

Paul Pinard & Axelle Senou

Les Bases :

- **Définition de son répertoire de travail :**
  - Session > Set Working directory > Choose Directory
  - Enregistrer le fichier R dans un même dossier que la table à analyser
- **Type de script :**
  - Fichier R script Classique : **File > New File > R Script**
  - Autre Possibilités : **Markdown, Quarto**
    - *Prise en main légèrement plus complexe la première fois, mais beaucoup plus pratique sur le long terme*
    - *Créer des zones de codes :* **Ctrl + Alt + i (maintenir Ctrl + Alt et ensuite appuyer en répétition sur i)**

- **Opérateur pour créer des objets :** « **<-** »

- **Importation de la Base de données :**
  - En cliquant sur des boutons :
    - **File > Import Dataset >**
      - **From Text (base)** (Dans le cas d'un fichier CSV)
      - **From Excel** (Dans le cas d'un fichier Excel)
    - **COPIER / COLLER** le code dans la nouvelle fenêtre qui vient de s'ouvrir
  - En utilisant directement du Code :
    - Fichier CSV :
      - `read.csv2("nom_fichier.csv")`
      - Possible de rajouter des arguments comme *(sep = " ; ")* ou *(header = TRUE)* si la table ne donne pas le résultat voulu
    - Fichier Excel :
      - `library(readxl)`
      - `read_excel("nom_fichier.xlsx")`
      - Besoin d'importer une extension « readxl » pour pouvoir lire les fichiers Excel

- **Visualisation de la table :**
  - Afficher la base :
    - Possible pour les deux fichiers : **View(bdd)**
    - Si Fichier Markdown :
      - Double Cliquer sur « bdd »
      - Ctrl + Entrée

- Afficher un aperçu des variables :
  - Même procédé pour les deux fichiers : **str(bdd)**
- Afficher le nom des variables :
  - Même procédé pour les deux fichiers : **names(bdd)**
- Afficher le nombre de lignes et de colonnes :
  - Même procédé pour les deux fichiers : **dim(bdd)**
  - *Bonus pour les fichiers Mardown :*
    - *Depuis la visionneuse, le nombre de lignes et le nombre de colonnes sont des informations déjà visibles sans utiliser de code supplémentaire !*

Les types de variables :

- Numeric (num), Double (dbl) ou Integer (Int) : Variable numérique (Age, Nb d'enfants, etc…)
- Factor (fct) : Variables Catégoriels (Sexe, Niveaux Etudes, '' Possède une voiture ? '', etc…)
- Character (char) : Chaine de Charactères (Réponses développées d'un questionnaire)
- Logical (lgl) : Variable contenant uniquement des TRUE/FALSE/NAs

Dans la pratique, les types les plus intéressants seront les numériques (peu importe le type), les facteurs et les chaines de charactères dans une moindre mesure.

NB : Attention aux Majuscules / minuscules !!!

Prise en Main / Remodelage de la base de données :

- Signe « $ » pour appeler une variable : **bdd$Var_factor**
- Récupérer les informations de :
  - La ligne n°1 : **bdd[1,]**
  - De la ligne 1 à la ligne 10 : **bdd[c(1 :10),]**
  - De la colonne 1 à 10 : **bdd[,c(1:10)]**
- Supprimer une variable :
  - **bdd$variable <- NULL**
- Renomer une variable :
  - **names(bdd)[names(bdd) == "ancien_nom_var"] <- "nouveau_nom"**
- Passer d'une variable :
  - Character / Numeric en facteur :
    - **bdd$variable <- as.factor(bdd$variable)**
  - Facteur en Character :
    - **bdd$variable <- as. character(bdd$variable)**
  - Facteur en Numérique :
    - **bdd$variable <- as.numeric(as.character(bdd$variable))**

- Remplacer une valeur :
  - Dans une colonne Character :
    - **bdd[bdd$Var_factor == "hOMmE",]$Var_factor <- "Homme"**
  - Dite comme « NA » (« Not Available ») :
    - **bdd[bdd$Var_factor %in% c("na","NA","Na","N/A"),]$Var_factor <- NA**
- Créer une nouvelle variable :
  - Grâce à un vecteur créer à la main
    - **bdd$Age_participant <- sample(size = 100, x = c(10 :70), replace = TRUE)**
  - En fonction de condition sur d'autres colonnes :
    - **bdd$Etat_Mineur <- NA**
    - **bdd[bdd$Age_participant <= 18 ,]$Etat_Mineur <- "Oui"**
    - **bdd[bdd$Age_participant > 18 ,]$Etat_Mineur <- "Non"**

Fonctions D'analyses :

- Récupérer un résumer de la table :
  - **summary(bdd)**
  - Possible **Summary(bdd$Age_participant)**
  - *Utile uniquement pour des variables numériques*
- Récupérer la moyenne d'une variable numérique :
  - **mean(bdd$Age_participant, na.rm = TRUE)**
  - "*na.rm = TRUE* " *permet que même si des valeurs manquantes sont parmi les âges dans notre cas, alors la moyenne n'en prendra pas en compte*
- Récupérer la médiane d'une variable numérique :
  - **median(bdd$Age_participant, na.rm = TRUE)**
- Récupérer le minimum d'une variable numérique :
  - **min(bdd$Age_participant, na.rm = TRUE)**
- Récupérer le maximum d'une variable numérique :
  - **max(bdd$Age_participant, na.rm = TRUE)**
- Récupérer l'écart-type d'une variable numérique :
  - **sd(bdd$Age_participant, na.rm = TRUE)**
- Connaitre la répartition des différentes catégories (Tri à plat):
  - **table(bdd$Var_Char)**
- Connaitre la répartition en fonction d'une seconde variable (Tri Croisé) :
  - **table(bddVar_Char, bddVar_hab)**

Les Graphiques :

Remarque : Les options ajoutées à l'histogramme sont transposables aux autres graphiques, et sont non exhaustives.

- Histograme :
  - **hist(bdd$Age_participant, main = "Titre du graphique", xlab = "Nom Axe X", ylab = "Nom Axe Y", col = "blue")**
- Boite à moustache :
  - **boxplot(bdd$Age_participant)**
- Diagramme Circulaire :
  - **pie(bdd$Age_participant)**
- Diagramme en bâton :
  - **barplot(bdd$Age_participant)**

BONUS :

- Visualisation d'une table de tri croisée avec pourcentage :
  - `library(sjPlot)`
  - **sjt.xtab(var.row = bdd$Etat_Mineur, var.col = bdd$Age_participant, show.row.prc = TRUE, show.col.prc = TRUE)**

Ressources supplémentaires :

- ✓ 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/