

Introduction au logiciel R - Statistique descriptive

Sarah Ouadah

Module DYST, M2 BEE, Septembre 2023

Contents

1) Présentation du logiciel R	1
1.2) Généralités	1
1.3) L'éditeur	1
1.4) Importation des données	2
1.4.1) Répertoire de travail	2
1.4.2) Importation	2
1.5) L'aide de R	2
1.6) Quelques commandes	2
1.6.1) Objets et Création de variables	2
1.6.2) Opérateurs	3
1.6.3) Sauvegarde	4
1.6.4) Installation d'un package R	5
1.7) Graphiques	6
2) Statistique descriptive	6
2.1) Importation	6
2.2) Statistique descriptive univariée	7
2.2) Statistique descriptive bivariée	9

1) Présentation du logiciel R

1.2) Généralités

R est un logiciel de calcul statistique. Il est constitué d'un noyau de base et de multiples packages développés et mis à disposition de tous par des utilisateurs. Le logiciel RStudio, lui, fournit une interface graphique très conviviale. L'interface de RStudio est divisée en plusieurs fenêtres :

1. l'éditeur, en haut à gauche, qui permet d'écrire du code et de le sauvegarder.
2. la console, en bas à gauche, qui sert à exécuter le code. C'est aussi dans cette fenêtre qu'apparaîtront les résultats des analyses statistiques exécutées.
3. une fenêtre en haut à droite qui contient l'espace de travail et l'historique des commandes.
4. une fenêtre en bas à droite qui regroupe les onglets Files-Plots-Packages-View.

1.3) L'éditeur

L'éditeur disponible dans RStudio permet d'écrire le code et de le sauvegarder dans un fichier texte. Cela est particulièrement utile pour y apporter des corrections, pour conserver une trace de vos travaux et relancer des programmes en cas d'interruption de RStudio.

- Si l'éditeur n'apparaît pas à l'ouverture de RStudio, on peut le créer à partir du menu File- New File - R Script.

- Pour sauvegarder le contenu de l'éditeur, on suivra le chemin suivant dans le menu : File -Save as .

Attention : il est vivement conseillé de toujours écrire les instructions dans l'éditeur avant de les exécuter dans la console et d'enregistrer le contenu de l'éditeur régulièrement.

1.4) Importation des données

1.4.1) Répertoire de travail

Il est pratique de se placer dans un répertoire de travail où se trouvent les données et où seront sauvegardés les résultats et les fichiers de programme. Le choix du répertoire peut se faire à partir du menu Session - Set working directory - choose directory .

– Spécifiez dans R votre répertoire de travail.

1.4.2) Importation

Il est nécessaire d'importer les données du fichier dans un tableau de données au format R avant de démarrer toute analyse.

– Chargez vos données file="mes_donnees.extension" dans une variable nommée par exemple data, à l'aide de la commande

```
data <- read.table(file, header = FALSE, sep = ",", dec = ".")
```

ou encore

```
data <- read.csv(file, header = TRUE, sep = ",", quote = "\"", dec = ".")
```

Pour cela, placez vous dans l'éditeur et écrivez-y l'instruction. Pour l'exécuter, il faut se placer sur la ligne de la commande puis presser simultanément les touches Ctrl et Entrée du clavier, ou encore cliquer sur Run.

Attention : R est sensible à la casse.

Il est à noter que les arguments des fonctions d'importation de données doivent être adaptés à ces données, la description des différents arguments possibles est disponible dans l'aide de R.

1.5) L'aide de R

R possède une aide assez complète, accessible via l'onglet Help situé dans la fenêtre en bas à droite de l'interface RStudio. On a accès à un moteur de recherche qui mène à la description des fonctions et à la documentation des packages. La description des fonctions aide à comprendre leur utilisation et leur fonctionnement via des exemples souvent proposés à la fin.

1.6) Quelques commandes

1.6.1) Objets et Création de variables

R travaille avec des objets. Les données, les graphiques, les fonctions, les résultats d'analyse sont des objets. Des objets courants sont : vector pour un vecteur composé d'une seule variable, data.frame pour un tableau de données et list pour une liste d'objets, généralement le résultat d'une analyse. En ce qui concerne les listes, chacun de leurs éléments sont accessibles en tapant le nom de la liste suivi du symbole \$ et du nom de l'élément.

Au départ, dans R, le symbole ← était utilisé pour l'affectation d'un objet; les versions récentes admettent également le symbole = pour l'affectation.

```
# un vecteur d'attributs quantitatifs
x1 <- c(1,7,8,3,1,8,1,1,2)
x1 = c(1,7,8,3,1,8,1,1,2)
```

```
# un vecteur d'attributs qualitatifs
x2 = c("F","M","F", "F","F", "M", "M","M")

# une matrice
x3=matrix(1:6, nrow=3, ncol=3)
```

Pour visualiser les objets créés, il suffit de taper leurs noms, ou d'utiliser la fonction print

```
x1

## [1] 1 7 8 3 1 8 1 1 2
print(x1)

## [1] 1 7 8 3 1 8 1 1 2
x2

## [1] "F" "M" "F" "F" "F" "M" "M" "M"
x3

##      [,1] [,2] [,3]
## [1,]    1    4    1
## [2,]    2    5    2
## [3,]    3    6    3
```

La plupart des objets ont un attribut de classe. Les plus courants sont numeric, logical, integer, factor, character, data.frame, etc.

```
class(x1)

## [1] "numeric"
class(x2)

## [1] "character"
class(x3)

## [1] "matrix" "array"
```

1.6.2) Opérateurs

Opérateurs mathématiques +, *, /, -, ^

```
x1/(10*100)^2

## [1] 1e-06 7e-06 8e-06 3e-06 1e-06 8e-06 1e-06 1e-06 2e-06
```

Opérateurs logiques & (et), | (ou), ! (négation), == (identité).

```
x1

## [1] 1 7 8 3 1 8 1 1 2
(x1 < 5) & (x1 ==1)

## [1] TRUE FALSE FALSE FALSE TRUE FALSE TRUE TRUE FALSE
x2

## [1] "F" "M" "F" "F" "F" "M" "M" "M"
```

```
x2=="F"
```

```
## [1] TRUE FALSE TRUE TRUE TRUE FALSE FALSE FALSE
```

Fonctions mathématiques sqrt(), log(), exp(), ...

```
sqrt(x1)
```

```
## [1] 1.000000 2.645751 2.828427 1.732051 1.000000 2.828427 1.000000 1.000000
## [9] 1.414214
```

Fonctions statistiques min(), max(), sum(), mean(), var(), median(), range(), order(), ...

```
mean(x1)
```

```
## [1] 3.555556
```

```
var(x1)
```

```
## [1] 10.02778
```

```
min(x1)
```

```
## [1] 1
```

```
range(x1)
```

```
## [1] 1 8
```

```
summary(x1)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      1.000   1.000   2.000   3.556   7.000   8.000
```

```
sort(x1)
```

```
## [1] 1 1 1 1 2 3 7 8 8
```

Opérations vectorielles extraction vec[i], mat[i,j], mat[,j] où (i : ligne, j : colonne)

```
x1[9]
```

```
## [1] 2
```

```
x3[1,2]
```

```
## [1] 4
```

```
x3[,2]
```

```
## [1] 4 5 6
```

```
# suppression vec[-i]
```

```
x1[-5]
```

```
## [1] 1 7 8 3 8 1 1 2
```

1.6.3) Sauvegarde

R propose à la fin de chaque session de sauvegarder tout l'espace de travail. Celui-ci est en général énorme et contient beaucoup d'objets intermédiaires inutiles. Pour sauvegarder uniquement quelques objets, utiliser la fonction save(). Pour charger les données sauvegardées utiliser la fonction load().

```
x = runif(20)
y = list(a = 1, b = TRUE, c = "oops")
save(x, y, file = "xy.RData")
rm(x)
rm(y)
load("xy.RData")
x
```

```
## [1] 0.08191166 0.54939928 0.82941886 0.81147063 0.27055682 0.45108147
## [7] 0.45440481 0.70925850 0.35902754 0.47257269 0.89841517 0.32156519
## [13] 0.61429300 0.89540486 0.52504674 0.38836625 0.33720726 0.60009756
## [19] 0.25641292 0.65235574
```

```
y
```

```
## $a
## [1] 1
##
## $b
## [1] TRUE
##
## $c
## [1] "oops"
```

Pour exporter un tableau :

```
x3
```

```
##      [,1] [,2] [,3]
## [1,]    1    4    1
## [2,]    2    5    2
## [3,]    3    6    3
```

```
write.table(x3,file="matrix.txt",quote=FALSE,row.names=FALSE,col.names=c("A1","A2", "A3"))
```

Pour sauvegarder un graphique : on peut utiliser Export de l’onglet Plots situé dans la fenêtre en bas à droite de l’interface RStudio.

1.6.4) Installation d’un package R

- Avec une connexion internet :

```
install.packages("nom_package")
```

ou

aller dans l’onglet Tools - Install Packages - puis choisir Repository (Cran) pour la case “Install from” et inscrire le nom du package pour la case “Packages”.

- A partir d’un fichier stocké sur votre ordinateur :

```
install.packages("~/chemin/nom_package", repos = NULL, type = "source")
```

où chemin indique où est le dossier où se trouve le package, par exemple “/home/R/packages/”

ou

aller dans l’onglet Tools - Install Packages - , puis choisir Package Archive File (.tar.gz) pour la case “Install from”, puis cliquer sur Browse pour sélectionner le fichier package.

Avant d’utiliser un package on va le chercher dans notre library avec la commande :

```
library(nom_package)
```

1.7) Graphiques

R permet de dessiner dans un graphique en y rajoutant des segments, des points, du texte, ou toutes sortes d'autres symboles. Toutefois, pour produire un graphique complet avec les fonctions basiques de R, il faut un peu "bricoler". Le package *ggplot2* devient vite indispensable lorsque l'on souhaite réaliser des graphiques plus jolis et plus complexes. Voir ici : <https://ggplot2.tidyverse.org> ainsi que le tutoriel [Tutoriel_graphiques_en_R.html](#)

2) Statistique descriptive

2.1) Importation

1. Chargez les données dans la variable `tab` à l'aide de la commande

```
tab <- read.table("bats.txt",header=T)
```

Vérifions le bon déroulement de l'importation du jeu de données.

1. Exécutez successivement les commandes `tab`, `head(tab)` et `str(tab)`. Qu'obtenez-vous dans chacun des trois cas?

2. En examinant la structure du fichier `bats.txt` et de l'objet `tab` dans R, quel est à votre avis le rôle de l'option `header=T` dans l'instruction d'import des données `read.table`?

Vous pourrez également trouver cette information dans l'aide de R en exécutant la commande

```
help(read.table)
```

3. Combien y a-t-il d'observations (lignes) et combien y a-t-il de variables (colonnes) dans `tab`?

4. Quels sont le nom et la nature des variables dans `tab` (qualitative ou quantitative)? Dans le cas de variables qualitatives, précisez le nombre de modalités.

5. Vous paraît-il normal que la variable `Diet` soit quantitative?

6. Exécutez la commande

```
tab$Diet = as.factor(tab$Diet)
```

Quelle est la différence avec les données d'origine? Pour répondre à cette question, vous pourrez vous servir de l'instruction `str()`.

7. Que se passe-t-il si vous exécutez l'instruction `Diet` ? l'instruction `tab$Diet`?

2.2) Statistique descriptive univariée

1. Rappelez la nature de la variable `BOW`. A votre avis, quels outils statistiques sont pertinents pour décrire cette variable?

2. Pour obtenir un résumé de l'information contenue dans la variable `BOW`, exécutez les instructions:

(a) `summary(tab$BOW)`

(b) `sd(tab$BOW)`

(c) `min(tab$BOW)`

(d) `max(tab$BOW)`

(e) `mean(tab$BOW)`

à quel type d'information ces différentes fonctions permettent-elles d'accéder?

3. Commentez les résultats obtenus.

4. On donne ensuite une représentation graphique de la variable `BOW` à l'aide de l'instruction:

```
boxplot(tab$BOW)
```

(a) Quel est le nom d'un tel graphique?

- (b) Que représentent les différentes composantes de ce graphique?
- (c) Commentez l'allure du graphique obtenu.
5. Une autre représentation graphique pertinente de la variable `BOW` est donnée par l'instruction
- ```
hist(tab$BOW)
```
- (a) Quel est le nom d'un tel graphique et que représente-t-il?
- (b) Commentez son allure.
6. Rappelez la nature de la variable `Diet`. Quel(s) outil(s) de statistique descriptive vous paraît(paraissent) adapté(s) pour décrire la variable `Diet`?
7. Exécutez l'instruction
- ```
summary(tab$Diet)
```
- Quel résultat cela donne-t-il et pourquoi est-il différent du résultat obtenu avec la même fonction sur la variable `BOW`?
- Un résultat équivalent peut être obtenu avec l'instruction
- ```
table(tab$Diet)
```
8. Quelques descriptions graphiques de la variable `Diet` peuvent être obtenues avec les instructions:
- (a) `barplot(table(tab$Diet))`
- (b) `pie(table(tab$Diet))`

Commentez les graphiques obtenus.

## 2.2) Statistique descriptive bivariée

1. On souhaite décrire la relation entre la masse corporelle et la masse du cerveau (les variables BOW et BRW).

- (a) Rappelez la nature de ces deux variables.

- (b) Quel type de graphique vous paraît le plus adapté à cette étude?

- (c) Exécutez l'instruction

```
plot(tab$BOW~tab$BRW)
```

et commentez.

2. On cherche maintenant à étudier le lien entre la masse corporelle et l'alimentation des chauve-souris.

- (a) Rappelez la nature de ces deux variables.

- (b) Que renvoie l'instruction suivante?

```
by(tabBOW,tabDiet,mean)
```

Complétez cette étude descriptive bivariée en utilisant d'autres caractéristiques de position et de dispersion.

- (c) Une représentation graphique adéquate est donnée par l'instruction:

```
boxplot(tab$BOW~tab$Diet)
```

Expliquez ce que représente ce graphique et commentez-le.