

TP0 : PRISE EN MAIN RAPIDE DE R/ R STUDIO/RMARKDOWN

On utilisera R, R studio pour tous les TPs, il est disponible sous Linux (de préférence pour les salles de TP) ou Windows.

Lors des TPs nous n'utiliserons que les commandes utiles à une initiation à l'analyse des données. D'autres commandes plus complexes et plus conviviales existent. L'approche utilisée ici est un juste milieu entre une initiation rapide à R/Rstudio/Rmarkdown et l'analyse des données.

Les fichiers sont à envoyer à l'adresse **anne.yao@uca.fr** ou le chargé de TP...

Remarque : Faire un copier/coller directement du pdf au R peut induire des caractères informatiques invisibles ; dans ce cas récrire la commande.

Prise en main rapide de R

Quelques références mais à cette date il y a une large majorité de références sur internet probablement mieux que ce qui est donné ici. Ne pas hésitez à échanger avec l'enseignant sur vos références...

— <http://www.math.univ-toulouse.fr/~besse/Wikistat/pdf/st-tutor2-R-init.pdf>

Prise en main rapide de Rstudio

— Thibault Laurent : ingénieur de recherche à l'Université de Toulouse.

<http://www.thibault.laurent.free.fr/pedago.html>

— Pour plus de renseignements sur le logiciel, voir les différents cours existants par exemple.

Installations...

— Installation de R...

<https://cran.r-project.org/>

— Installation de Rstudio...

<https://posit.co/download/rstudio-desktop/>

Présentation par l'enseignant.

— Présenter les différents "panel" de Rstudio, comment les organiser,...

— Sur les scripts R (5min)... savoir que ça existe à titre indicatif....

Nous ferons essentiellement des notebooks Rmarkdown.

— Faire un premier notebook :

* Titre 1, Titre 2, Titre 3

* Les gras, italics

* Les blocs R

* Gestion des blocs R

* Installation de package (exemple ggplot2)

— En version courte aller directement aux vecteurs..

Exercice 0 : Définir un répertoire de travail.

Date:

- (1) Choisissez votre répertoire de travail (hors R)
- (2) Enregistrer les deux fichiers (figis_guestnull.csv et statlab.txt) de données fournies dans ce répertoire.

1. L'ÉDITEUR RSTUDIO/ RMARKDOWN/ ENVIRONNEMENT DE TRAVAIL

Exercice 1 : Ouvrir R studio.

- (1) Présentation du tableau de bord : la console, help, les packages,... .
 - Voir aussi l'utilisation des menus :
 - * “File”(ouvrir de nouveaux fichiers et type de fichier),
 - * “Code”, “View”.
 - Quelques raccourcis : <https://bookdown.org/yihui/rmarkdown-cookbook/rstudio-shortcuts.html>
- (2) **Gestion du répertoire de travail**

```
getwd()-> donne le répertoire de travail
setwd(/chemin/vers/mon/repertoire/de/travail)
```

(3) Les fichiers pour faire du R

(a) **Création de fichiers R** (voir présentation de l'enseignant)-> *hors programme*

- Dans l'éditeur, on peut créer un fichier de type script R d'extension **.R**. Enregistrer le fichier sous le nom **.../chemin/TP1votrenom.R**
- Pour exécuter une ligne de code du fichier en plaçant le curseur sur cette ligne et en cliquant sur le bouton « Run ».
- Exemple dans votre fichier R, exécuter les lignes successives.

```
0 :5
seq(0.5,10, by=0.5)
x=seq(0.5,10, by=0.5)
x
y=0 :9
y
c(x,y) # concatène x et y
```

(b) **Création d'un notebook** (voir présentation de l'enseignant)

- Requiert l'installation de package “Rmarkdown”
- Qu'est qu'un package? -> point suivant

- (4) **Les packages R** (comme python pour datascience) fonctionnent à base de bibliothèques (library en anglais), package.

Appeler un package permet d'accéder aux différentes fonctions R programmées par le(s) développeur(s) du ce packages.

- (a) Cela n'est pas nécessaire pour certaines opérations de base (arithmétique, statistiques élémentaires, multiplication matricielle,...). Les packages sont chargés automatiquement au lancement de R : "stat", "base"

- `library()` # permet de voir les packages installés. ou...
- Cliquez sur l'onglet "Packages" de RStudio pour avoir les packages installés.

- (b) **Certains packages sont installés par défaut.** Il suffit de les "charger" ou "appeler".

La fonction `library(nom du package)` permet de charger le packages.

- Exemple : `library(MASS)` : "Modern Applied Statistics with S" permet d'accéder aux fonctions :
- <https://www.rdocumentation.org/packages/MASS/versions/7.3-53>
- On peut utiliser directement le tableau de bord de Rstudio (voir présentation).

- (c) La plupart des **packages sont à installer**. C'est par exemple le cas des packages comme

- ***Rmarkdown, gglplot2 ; FactoMineR, factoextra***, dont nous aurons besoins par la suite.

Exercice :: Installer ces 3 packages.

* Conseil rajouter l'option "dependency =T"

- Lancer / charger les packages : `library(le nom du package)`
- Cliquer sur l'onglet "Environnement" et sectionner un des 3 packages.

- (5) **Notebook Rmarkdown**, (voir présentation de l'enseignant)

Très pratique pour faire à la fois du code et un compte rendu, un manuscrit, document, pdf, word, html,...

- <https://rmarkdown.rstudio.com/gallery.html>
- Voir aussi les cheatsheets (menu "help" de Rstudio).

- (a) Installer le package "Rmarkdown".

- (b) Créer un notebook.

- Enregistrer le fichier sous le nom **.../chemin/TP1votrenom.Rmd**
- Exécuter le "preview"
- Afficher la sortie "html" (cela crée un fichier **.html**)
- Exécuter les commandes précédentes dans le notebook.

1.1. Les vecteurs sous R. Vecteurs simples

```
0 :5 # signal le vecteur de entier successif de 1 à 5
#[1] 0 1 2 3 4 5
seq(a,b,by=l)# sequence de valeurs de a à b avec un pas de 0.5
seq(0.5,10, by=0.5)
seq(1,50,by=2)
seq(a,b,length=l)# sequence de valeurs de a à b de longueur l.
seq(1,50,length=2)
```

Concaténation de vecteurs : soient x et y deux vecteurs. Le vecteur $c(x,y)$ concatène x et y .

```
x=seq(0.5,10, by=0.5)
#[1] 0.5 1.0 1.5 2.0 2.5 3.0 3.5 4.0 4.5 5.0 5.5 6.0 6.5 7.0 7.5 8.0 8.5 9.0 9.5
10
y=0 :9
c(x,y) # concatène x et y
[1] 0.5 1.0 1.5 2.0 2.5 3.0 3.5 4.0 4.5 5.0 5.5 6.0 6.5 7.0 7.5
[16] 8.0 8.5 9.0 9.5 10.0 0.0 1.0 2.0 3.0 4.0 5.0 6.0 7.0 8.0 9.0
```

Extraction d'élément d'un vecteur

```
x[1] # la première valeur de x
x[5]# la 5ème valeur de x
x[1 :4]# les 4 premières valeurs de x
x[c(1 :2,4 :8)]# affiche les deux premières valeurs et les 4ème à 8ème de
x.
x[(1 :4)]# les 4 premières valeurs de x
x[-2] # enlève sa deuxième valeur à x.
x[-c(1 :4)]# enlève les 4 premières valeurs de x
```

1.2. Les matrices. Soit M une matrice :

```
set.seed(10)
M=matrix(round(rnorm(50),1),10,5)
```

Affichage de base pour les matrices :

```
dim(M)
nrow(M)
ncol(M)
names(M)
dimnames(M)
```

La position des nombres par rapport à la virgule indique s'il s'agit de lignes ou de colonnes

```
M[4,] # affiche la 4ème ligne de M.
M[,5] # affiche la 5ème colonne de M.
M[1,4] #affiche l'élément situé à la 1 ligne et 4eme colonne
M[1 :4,]# les 4emes premières lignes de M
M[,1 :4]# les 4emes premières colonnes de M
M[-4,]# Enlève la 4eme ligne de M.
M[,-5]# Enlève la 5eme colonne de M.
Logiquement qu'indique les affichages ci dessous ?
M[c(1 :2,4 :8),]# .....
M[c(1 :2,5),]###
M[-c(1 :2,5),]###
```

Création de matrice à partir de vecteurs et de matrices.

```
x=seq(0.5,10, by=0.5)
y=seq(2,15, length=10)
M2=cbind( x,y)
M3=cbind(M2,x)
z=seq(0,1, length=10)
M4=cbind(x,y,z)
##
M5=cbind(M2,M3)
M6=rbind(x,y)
M7=rbind(M5,z)
```

Remarque. Il est possible de créer des vecteurs et matrices contenant des lettres, mots,....

Exécuter les commandes suivantes.

```
c(1,"a","ddd")
c(1,"a","mon premier fichier R")
paste("a",1 :2,sep="")
paste("b","a",1 :4,sep="")
paste("b","a",1 :3,sep=".")
paste("b","a",1 :4,sep="")
```

1.3. Les objets “dataframes” (tableau de données). Voir illustration par tableau de données “**dat-don**” dans notre exemple. Importation

Les types de matrices sous R.

R distingue deux types de matrices :

- (1) Les “vraies” **matrices** considérées comme telles sous R comme en tant des objets mathématiques).
- (2) Les “**data frame**” ce sont les tableaux de données exprimant le fait que l’on va faire du traitement de données avec (voir ci-dessous).
 - Toute opération sur le groupe 1. est applicables au groupe 2. L’inverse est n’est pas systématique.
 - Les objets “dataframe” comporte par défaut des noms de lignes et de colonnes.

1.4. Les objets list. Ce sont des objets concatènent peuvent contenir tout type d’objet de R.

```
maliste=list(vect=x,mat=M[,1 :5],z=2+3,chac=paste("a",1 :2,sep=""))
maliste.
```

2. IMPORTATION DES DONNÉES SOUS R**2.1. Importation de données sous R..**

Plusieurs stratégies pour importer les données en fonction de comment sont fournis les données :

- (1) **read.table()**, **read.csv()**, **load()**,.... A utiliser si le tableau de données est bien structuré en colonne. Ce qui n’est pas notre cas. C’est alors **déjà un data.frame**.
 - Sinon, comme dans notre cas, il faut **reconstituer le data.frame**
- On utilise alors l’approche ci-dessous i.e.
- * On utilise ici la fonction **scan()**. *Valable uniquement si le tableau comporte uniquement des valeurs numériques.*

- (1) Importer les données du fichier : **figis_guestnull.csv** .

```
Dat=read.csv("figis_guestnull.csv")
### Regarder l'aide de read.csv
#Il y a des options pour gérer des fichiers avec d'autres spécificités.
is.data.frame(Dat)
```

Remarque : le tableau importer **est directement** un “**dataframe**”. Dans le 2eme tableau, nous verrons qu’il faudra (re)constituer le dataframe.

- (2) Affichage d’informations sur les dimensions du tableau de données : **Dat**

```
#affichage de la dimension de datdon
dim(Dat )
ncol(Dat )
nrow(Dat )
#affichage des noms de lignes et colonnes
dimnames(Dat)
rownames(Dat)
colnames(Dat)
```

- (3) Importer les données du fichier : **statlab.txt** .

```
Appliquer les commandes précédentes au tableau de données
####Importation des données###
datdon=scan("statlab.txt")#into a vector
datdon=matrix(datdon,ncol=19,byrow=T)
is.data.frame(datdon)
```

Le dataframe est à constituer...

- (4) Affichage d’informations sur la taille du tableau de données : **datdon**

```
#affichage de la dimension de datdon
dim(datdon)
ncol(datdon)
nrow(datdon)
#affichage des noms de lignes et colonnes
dimnames(datdon)
rownames(datdon)
colnames(datdon)
```

- (5) Affichage des lignes et colonnes du tableau.

- (a) Afficher la **4ème ligne** et la **5ème colonne** du tableau de donnée **datdon**.
- (b) Afficher les lignes 2 à 4 et 6 à 10
- (c) Afficher les informations concernant les 10 premiers individus.
- (d) Afficher les informations sur la **4ème colonne** concernant les 10 premiers individus.

- (6) Indiquer à R qu’il s’agit d’un tableau de données, un “data frame” et gestion des noms des variables

- (a) Conversion en dataframe

```
#### Conversion en dataframe####
datdon=data.frame(datdon)
dimnames(datdon)
rownames(datdon)
colnames(datdon)
```

(b) Gestion des noms des variables.

```
#### Gestion des lecture du nombre de variables##
varname=c("sexenf","gsenf","tenf_n","penf_n", "tenf_10","penf_10",
"gsmere","agem_n","pmere_n","consm_n","tmere",
"pmere_10", "consm_10","agep_n", "consp_n","tpere","ppere_10",
"revf_n","revf_10")
dimnames(datdon)
rownames(datdon)
colnames(datdon)

#### On va remplacer les noms initiaux des variables par ceux indiqués dans varname
colnames(datdon)=varname
dimnames(datdon)
rownames(datdon)
colnames(datdon)
```

(7) Affichage des colonnes du tableau après conversion en data.frame.

```
datdon$sexenf # utilise le nom de la variable
datdon[,3]# si on connaît la position de la variable
#ou
datdon[, "sexenf"]
```

2.2. Premier résumés statistiques.

```
#### Résumé des variables et du tableau##
str(datdon)# exprime le type de chaque variable. numérique ou non
summary(datdon)
```

2.3. Refaire les calculs (prévus) du TD en utilisant sous R..

Remarque. Ce Tp est une prise en main rapide en vue d'avoir les éléments de bases de la science des données, d'autres fonctions seront vues au fur et à mesure du besoins.