

# TD de la 1° séance

---

## Installer R

Pour installer R il faut se rendre sur le site internet : <http://www.r-project.org>

## Installer un package dans la bibliothèque

1) installer le package epiR

`install.packages()` #on choisira d'installer le package epiR par exemple

`library(epiR)`

## Premiers exercices de prise en main du logiciel R

1) Ouvrir un nouveau script et vérifier le répertoire de travail avec la fonction `getwd()`

`getwd()`

2) Différentes façons de changer le répertoire de travail

a) méthode1

A partir du menu fichier => choisir l'option changer le répertoire courant et suivre les indications, notre répertoire bureau/Atelier

b) méthode2

En utilisant les fonctions `setwd( )`, `dirname( )` pour extraire le chemin et `file.choose( )` pour choisir le répertoire de façon interactive.

`setwd(dirname(file.choose()))`

NB: attention cependant il faut avoir déjà un fichier dans ce répertoire quand on utilise la fonction `file.choose()`

3) Différentes fonctions

Pour obtenir de l'aide sur une fonction

a) `help(sum)` #aide sur la fonction `sum()`

b) `?sum` #autre façon d'appeler l'aide

`citation()` #permet d'avoir la référence du logiciel pour une publication

`history ( )` #permet d'avoir les dernières instructions saisies

history(Inf) #permet d'avoir toutes les instructions saisies

Ctrl+I #efface le contenu visible de Rconsole

4) R comme calculateur

5-6

6\*5

6^2

6/5

sum(1,5)#fait la somme de 1 et de 5

#attention la virgule sert à séparer les arguments

sum(1.5,0.5)

5) Les 3 opérateurs possibles pour assigner une valeur à un objet :<-,=,->

x<-4

y=5

6->z

6) Retrouver rapidement les différents objets créés

ls() #liste tous les objets créés

7) Effacer un objet

rm(x) #efface l'objet x, vérifier avec la fonction ls( ) que l'objet x a disparu

NB : on peut recréer l'objet x en recherchant la commande initiale avec les flèches de déplacement.

rm(list=ls()) #efface tous les objets

8) opérateurs de logique et de comparaison

Créer un objet vecteur appelé Age avec la fonction c( )

Age<-c(2,4,5,10,8,7,12,11)

Age[Age>=5 & Age<10] # extraction à l'aide de [ ]

Age[Age<5 | Age>10]

Age[Age!=5]

9) Création de vecteurs

Classe<-c("primaire", "secondaire", "universitaire")

Sexe<-c("Homme", "Femme")

Sexe<-rep(c("Homme", "Femme"),c(15,16)) #fonction rep( ) permet la répétition

length(Sexe) #pour vérifier le nombre de données saisies dans l'objet vecteur Sexe

Tranchage<-rep(c("0-4", "5-9", ">=10"),c(8,9,14))

Tranchage

Poids<-seq(10,25,by=.5) #séquence de nombre de 10 à 25 en avançant de 0.5 en 0.5

Poids

10) Extraire une partie d'un vecteur à l'aide de [ ]

Poids[16] #utilisation de [ ] pour choisir le 16° poids du vecteur poids  
Poids[c(1,16,31)] #choix du 1°, 16° et 31° poids  
Poids[Poids>20] #choix des poids supérieurs à 20

#### 11) Création de liste

On a vu que la fonction ls() présente tous les objets créés sous forme d'une liste. Une liste est donc un objet hétérogène composé d'objets de mode et de longueur différents.

```
Maliste<-list(x,Poids,Classe)
length(Maliste)
mode(Maliste)
Maliste
```

#### 12) création de Matrices

Une matrice est une série de valeurs à 2 dimensions (lignes et colonnes)

```
Chiffre<-c(1:15) #vecteur des chiffres de 1 à 15
Chiffre #pour voir les valeurs du vecteur Chiffre
```

- création d'une matrice à 3 colonnes à partir du vecteur Chiffre avec la fonction matrix()

```
Matrice1<-matrix(Chiffre,ncol=3)
Matrice1 # voir le contenu de Matrice1
```

- Matrices (autres méthodes)

```
Matrice.1<-c(1:15) #création d'un vecteur de 1 à 15
dim(Matrice.1)<-c(5, 3) # dim() transforme le vecteur en matrice
Matrice.1
```

- Création d'une matrice à 3 lignes et remplissage en commençant par les lignes

```
Matrice2<-matrix((1:15)*100,nrow=3, byrow=TRUE)
Matrice2
```

#### 13) Création de tables de données (Data frames : les colonnes sont des variables)

```
Dataframe1<-data.frame(Matrice2)
```

la fonction data.frame( ) transforme la matrice Matrice2 en Dataframe1

```
Dataframe1 #observer la différence entre Matrice2 et Dataframe1
```

#### 14) donner un nom à une variable, en utilisant la fonction names( )

```
names(Dataframe1)[1]<- "TOUS" # ici c'est le nom de la première colonne
Dataframe1
```

#### 15) manipulations de champs "dates"

D'un point de vue pratique, l'import des données vers R convertit les dates au format texte. Il est utile de les transformer donc au format date pour leur manipulation.

Construisez les deux vecteurs suivants

```
Datedeb<-c('12/01/90','02/01/90','22/04/95','05/06/91') #date de début
```

```
Datefin<-c('25/02/90','15/06/91','20/05/96','14/04/92') #date de fin
```

Vérifier leur type

```
mode(Datedeb) ; Datedeb
```

```
mode(Datefin); Datefin
```

Pour les convertir utiliser la fonction `as.Date()`

```
Datedeb<-as.Date(Datedeb,'%d/%m/%y')
```

```
Datefin<-as.Date(Datefin,'%d/%m/%y')
```

```
mode(Datedeb) ; Datedeb
```

```
mode(Datefin); Datefin
```

Comparer ces résultats avec ceux obtenus précédemment

Calculer une durée - fonction `difftime()`

```
Duree.1<-difftime(Datefin,Datedeb,units="days")#résultat en jours
```

```
Duree.1
```

Autre méthode

```
Duree.2<-Datefin-Datedeb
```

```
Duree.2
```

Autres fonctions utiles

```
Sys.Date() #renvoie la date du système
```

```
Sys.time() #renvoie le temps du système
```

```
Sys.Date() + 40 #renvoie la date dans 40 jours
```

```
Sys.time() + 3600 #renvoie le temps dans une heure (3600 secondes)
```

Méthode d'extraction de date - fonction `substr()`

```
Annee<-substr(Datefin,1,4) #extraction de l'année
```

```
Mois<-substr(Datefin,6,7) #extraction du mois
```

```
Jour<-substr(Datefin,9,10) #extraction du jour
```

```
Annee;Mois;Jour
```

Autres méthodes

```
Annee.1<-strftime(Datefin,"%Y")
```

```
Mois.1<-strftime(Datefin,"%m")
```

```
Jour.1<-strftime(Datefin,"%d")
```

```
Annee.1;Mois.1;Jour.1
```

```
Annee.2<-as.character(Datefin,"%Y")
```

```
Mois.2<-as.character(Datefin,"%m")
```

```
Jour.2<-as.character(Datefin,"%d")
```

Annee.2;Mois.2;Jour.2

Pour le numéro de semaine

```
Numsem<-strftime(Datefin,"%U")
```

```
Numsem.1<-as.character(Datefin,"%U")
```

```
Numsem; Numsem.1
```

16) création d'une table de données à partir de plusieurs vecteurs

```
Taille<-c(130,132,145,136,140,162,148,151) #création du vecteur Taille
```

```
Poids2<-c(45,46,48,34,44,43,38,49) #création du vecteur Poids2
```

```
Sex2<-c(1,2,1,2,1,2,2,1) #création du vecteur Sex2
```

```
Age2<-c(12,11,13,12,12,13,12,13) #création du vecteur Age2
```

```
Dataframe2<-data.frame(Taille,Poids2,Sex2,Age2) #création de Dataframe2 à partir des 4 vecteurs
```

```
Taille, Poids2, Sex2, Age2
```

```
head(Dataframe2) # pour voir le début du contenu de Dataframe2
```

```
View(Dataframe2) # pour voir le contenu de Dataframe2 dans une autre fenêtre
```

17) autre exemple

```
Sexe<-rbinom(31,size=1,prob=.48) #distribution binomiale pour 48% de mâle
```

```
Sexe1<-ifelse(Sexe==0,'Femme','Homme')
```

```
Mydata<-data.frame(Sexe,Sexe1,Poids,Tranchage)
```

```
Mydata[1:5,]
```

```
summary(Mydata)
```

```
Mydata$Sexe<-factor(Mydata$Sexe) #transforme la variable sexe en variable dichotomique
```

```
summary(Mydata) #vérifier le changement par rapport au précédent résultat.
```

## Exercices d'application sans aide

- A1. Créer un objet de type vecteur, appelé  $\text{Var.x} = (0, 2, 5, 6)$ .
- A2. A l'aide de la fonction `seq` construire l'objet de type vecteur appelée  $\text{Var.v}=(0,4,8,\dots,100)$ .
- A3. Extraire la première, puis la troisième composante du vecteur  $\text{Var.v}$ ,
- A4. Extraire ensemble la première et la troisième composante et les mettre dans un objet de type vecteur appelée `Extract.v`
- A5. Remplacer la première composante de l'objet  $\text{Var.v}$  par la valeur -1.
- A6. Remplacer la sixième et la dixième composante de la variable  $\text{Var.v}$  par -2.
- A7. Créer un objet de type logique  $\text{Var.x2}$  qui prend la valeur `TRUE` si  $\text{Var.x}$  est strictement plus grande que 2 et la valeur `FALSE` dans le cas contraire.
- A8. Puis à l'aide de cet objet extraire les composantes de  $\text{Var.x}$  ayant des valeurs plus grande que 2.

- B1. Créer un vecteur  $X=(\text{"bleu"},\text{"blanc"},\text{"rouge"},\text{"rouge"})$ .
- B2. A l'aide de la fonction `rep` dupliquer ce vecteur deux fois.
- B3. Construire un vecteur  $Y$  en dupliquant chacune des composantes de  $X$  3 fois.
- B4. Construire un vecteur logique qui vaut `VRAI` si et seulement si la couleur est "rouge".
- B5. Extraire du vecteur  $Y$  les valeurs qui ne sont pas "rouge".

### Les matrices

Soit la matrice  $5 \times 4$  à coefficients réels.

Matrice =

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 2 & 3 & 4 \\ 0 & 0 & 0 & 3 \\ 1 & 1 & 1 & 1 \end{pmatrix}$$

- C1. Créer la matrice  $\text{Mat.1}$  avec les coefficients rangés par ligne.
- C2. Créer un vecteur  $\text{Var.1}$  valant  $(\text{"Y1"},\text{"Y2"},\text{"Y3"},\text{"Y4"})$ .
- C3. Créer un vecteur de noms de lignes  
 $\text{Ident.1}=(\text{"EPI001"},\text{"EPI002"},\text{"EPI003"},\text{"EPI004"},\text{"EPI005"})$ .
- C4. Nommer les colonnes de la matrice  $\text{Mat.1}$  à l'aide de l'objet  $\text{Var.1}$  et les lignes à l'aide de l'objet  $\text{Ident.1}$
- C5. Extraire la ligne correspondant à l'individu `EPI004`,
- C6. Extraire la colonne nommée `Y3`,
- C7. Extraire la sous matrice  $\text{Smat.1}$  où l'on a sélectionné la deuxième et troisième ligne ainsi que la première et la troisième colonne.
- C8. Ajouter la colonne  $(0, 0, 0, 0, 1)$  à l'aide de la fonction `cbind` à la matrice  $\text{Mat.1}$ ,
- C9. Ajouter la ligne nulle après la dernière ligne de la matrice  $\text{Mat.1}$

#### Data frame

- D1. Créer l'objet Ident contenant les noms de 7 personnes suivantes : ("Marie","Jacques","Anna","Jean","Joseph","Paul","Isabelle").
- D2. Créer l'objet Age contenant 7 valeurs entre 20 et 50 ans avec un pas de 5 ans.
- D3. Créer l'objet Poids contenant 7 valeurs entre 50 et 86 avec un pas de 6.
- D4. Créer l'objet Taille contenant 7 valeurs entre 1,6 à 1,66 avec un pas de 0,01.
- D5. Créer l'objet Heavy prenant la valeur TRUE si poids >78kg et FALSE dans les autres cas
- D6. Créer l'objet Imc(Poids/Taille en m au carré )
- D8. Créer un data frame Mydata.1 contenant les différentes variables créer au dessus.
- D9. Si IMC >25, on parle de surpoids , extraire le nom des personnes en surpoids

#### E1.Créer une base de données avec les informations suivantes:

Une variable "Sexe" comprenant 4 garçons, 3 filles, 2 garçons, 6 filles, 5garçons classés dans cet ordre.

Une variable "Tabac" comprenant 30% de fumeurs

Variable "Toux" comprenant 25% de personnes avec une toux

Afficher le résultat dans une autre fenêtre.

Calculer le sex ratio Male/Female

## Exercice de préparation à la séance 2

On se propose de créer une population fictive de 100 personnes comprenant des hommes et des femmes (avec un sex ratio H/F=1,08), d'âge moyen 25 ans (écart type =2), de poids moyen 75kg pour les hommes (ET=5) et de 59kg chez les femmes (ET=4) et de taille moyenne 165cm pour les femmes (ET=3) et 174 m pour les hommes(ET=2,5), et dont 10% de la population a été infecté par le virus de la grippe A(H1N1)pdm09.

Il faut donc créer un objet Sexe mais combien faut-il d'hommes et de femmes dans un groupe de 100 personnes pour avoir un sex ratio de 1,08 ?

Créer un objet Sexe

```
set.seed(13) #fixe la graine du générateur
```

```
Sexe<-rbinom(100,1,.52)
```

Vérifier combien on a d'hommes et de femmes

```
table(as.factor(Sexe))
```

regarder les données dans leur ensemble

```
View(Sexe)
```

Que remarque-t-on en terme de codage ?

On se propose de recoder la variable Sexe, avec le code 1 attribué aux Hommes('M') et le code 0 attribué aux Femmes('F')

Pour se faire il faut utiliser la fonction recode() qui nécessite d'installer le paquet de la bibliothèque Rcmdr (install.package(Rcmdr)) puis appeler le paquet avec la fonction library()

```
Library(Rcmdr)
```

```
Sexe<-recode(Sexe,"1='M';0='F'")
```

Créer la variable Age selon les normes données avec un chiffre après la virgule

```
set.seed(13)
```

```
Age<-round(rnorm(100, mean=25, sd=2),digits=1)
```

Créer la variable Poids selon les normes données avec un chiffre après la virgule, attention il faudra différencier ici les hommes et les femmes

```
Poids<-ifelse (Sexe=="F",round(rnorm(100, mean=59,sd=4),digits=1),  
round(rnorm(100, mean=75, sd=5),digits=1))
```

Créer la variable Taille selon les normes données sans chiffre après la virgule, attention il faudra différencier ici les hommes et les femmes

```
Taille<-ifelse (Sexe=="F",round(rnorm(50, mean=165,sd=3),digits=0),  
round(rnorm(100, mean=174, sd=2.5),digits=0))
```

Créer la variable Maladie qui prendra la valeur 1 si « malade » et 0 si « non malade »

```
set.seed(14) #fixe la graine du générateur  
Maladie<-rbinom(100,1,.35)
```

Créer une table Mydata1 qui contiendra l'ensemble des vecteurs créés

```
Mydata<-data.frame(Sexe,Age,Poids, Taille, Maladie)
```

Regarder le résultat des données  
Mydata

Qu'observez-vous?

Vous préférerez donc:

```
View(Mydata)  
edit(Mydata)  
head(Mydata)
```

Taper  
table(Sexe, Maladie)  
table(Mydata\$Sexe,Mydata\$Maladie)

Changer les noms des variables de Mydata afin de les écrire en majuscule et de pouvoir les différencier des objets créés dans R

```
names(Mydata)[1]<-"SEXE"  
names(Mydata)[2]<-"AGE"  
names(Mydata)[3]<-"POIDS"  
names(Mydata)[4]<-"TAILLE"  
names(Mydata)[5]<-"MALADIE"
```

```
head(Mydata)
```

Que remarquez-vous ?

Taper

```
table(Sexe, Maladie)  
table(SEXE, MALADIE)
```

Que se passe-t-il ?

Taper maintenant

```
table(Mydata$SEXE,MALADIE)  
table(Mydata$SEXE,Mydata$MALADIE)
```

pour avoir la somme des lignes

```
apply(table(Mydata$SEXE, Mydata$MALADIE), MARGIN=1,sum)
```

pour simplifier l'écriture on aurait pu écrire:

```
Mytable<-table(Mydata$SEXE,Mydata$MALADIE)  
apply(Mytable, MARGIN=1,sum)
```

avoir la somme des colonnes

```
apply(Mytable,MARGIN=2,sum)
```

Taper

```
summary(Mydata)
```

```
summary(Mydata$POIDS)
```

```
tapply(Mydata$POIDS,Mydata$SEXE,summary)
```

*Enregistrer vos données de Mydata dans un format.CSV pour pouvoir garder les variables créées au cours de ce TD car nous l'utiliserons par la suite. Vous nommerez le fichier de sortie DataTD1*

```
write.csv2(Mydata,"DataTD1.csv")
```