

## TP N°2

### Manipulation avec R

#### Objectifs du TP :

- Savoir comment installer des packages.
- Savoir comment manipuler les données.
- Savoir comment travailler avec les graphiques.
- Savoir comment sauvegarder un fichier de données.
- Diverses manipulations.

#### Etapes du TP

```
#vider le contenu de la mémoire
rm(list=ls())
#installer le package xlsx
install.packages("xlsx")
#charger la librairie xlsx
library(xlsx)
#lister les fonctions de la librairie
#library(help=xlsx)
#changement du répertoire courant
setwd("chemin du répertoire r_manip_data")
#charger les données
heart.full <- read.xlsx(file="heart.xlsx",sheetIndex=1,header=T)
#description des variables de l'ensemble de données
print(summary(heart.full))
#lister le contenu de la mémoire
ls()
#accès à la variable 'age'
print(heart.full$age)
#type de 'age'
class(heart.full$age)
# 'longueur' de 'age' -> nombre d'observations
length(heart.full$age)
#accès par indices - plage de valeurs
heart.full$age[1:10]
#accès par indices - qqs valeurs dispersées
heart.full$age[c(2,5,8)]
#stat.descriptives - moyenne
mean(heart.full$age)
#stat.descriptives - quantiles
quantile(heart.full$age,probs=c(0.1,0.5,0.9))
#stat.descriptives sur une partie des valeurs
mean(heart.full$age[1:10])
#type de 'sexe'
class(heart.full$sexe)
```

```

#description des valeurs
levels(heart.full$sexe)
#nombre d'observations de 'sexe'
length(heart.full$sexe)
#accès par indices - plage de valeurs
heart.full$sexe[1:10]
#accès par indices - qqs valeurs dispersées
heart.full$sexe[c(2,5,8)]
#fréquences
table(heart.full$sexe)
#codes internes des valeurs
unclass(heart.full$sexe)
#data.frame ~ matrice -> accès indicé
m <- heart.full
nrow(m)
ncol(m)
m[1,1]
m[1:5,2:4]
m[c(2,5,8),2:4]
m[2:4,c(1,3,6)]
m[1,]
m[5:6,c("age","angine")]
#data.frame ~ matrice -> restrictions
m[m$age<30,]
m[m$age<=34 & m$sexe=="masculin",c("age","sexe","coeur")]
m[m$age<=34 | m$age>=76,c("age","sexe")]
a <- m[m$age<=45 & m$sexe=="masculin",c("angine","coeur")]
nrow(a)
ncol(a)
table(a$coeur)
k <- table(a$angine,a$coeur)
class(k)
print(k)
print(k[1,2])
#*****
#quelques calculs récapitulatifs
#*****
#croisement coeur-angine
e <- table(m$angine,m$coeur)
print(e)
#nombre de angine=oui
print(sum(e[2,]))
#proportion de malades parmi les angine=oui
print(e[2,2]/sum(e[2,]))
#age moyen selon le sexe
tapply(X=m$age,INDEX=m$sexe,mean)
#age moyen selon le sexe et l'occurrence de l'angine
b <-tapply(X=m$age,INDEX=list(m$sexe,m$angine),mean)
print(b)
#écart entre la plus petite et la plus grande moyenne
d <- max(b)-min(b)
print(d)
#écart entre min et max dans chaque sous-groupe
tapply(X=m$age,INDEX=m$sexe,function(x){max(x)-min(x)})
ls()
#*****
#tris des vecteurs et tableaux

```

```

#*****
#afficher les 6 premières valeurs de age
head(m$age)
#age2 est un nouveau vecteur des valeurs triées
age2 <- sort(m$age)
#afficher les 6 premières valeurs de age2
head(age2)
#trier un data frame selon une ou plusieurs variables
head(order(m$age)) #création d'un index selon l'âge
head(m[order(m$age),]) #tri selon un critère
head(m[order(m$age,m$tauxmax),]) #tri selon deux critères
#*****
#graphiques
#*****
#graphiques 1D - distributions
hist(m$tauxmax)
plot(density(m$tauxmax))
#installer le package sm
install.packages("sm")
#graphiques pour comparaisons
library(sm)
sm.density.compare(m$tauxmax,m$coeur)
#comparaisons avec boxplot
boxplot(m$tauxmax)
boxplot(m$tauxmax ~ m$coeur)
#graphiques 2D
plot(m$age,m$tauxmax)
plot(m$age,m$tauxmax,pch=21,bg=c("green","red")[unclass(m$coeur)])
#*****
#création de nouvelles variables
#*****
#lister les variables du data frame
colnames(m)
#graphique nuage de points
plot(m$age,m$tauxmax,pch=21,bg=c("green","red")[unclass(m$coeur)])
#création de la variable tauxnet
tauxnet <- m$taux*log(m$age)
#graphique nuage de points
plot(m$age,tauxnet,pch=21,bg=c("green","red")[unclass(m$coeur)])
#ajouter la nouvelle variable au data frame "m"
m <- cbind(m,tauxnet)
#lister les variables
colnames(m)
#sauvegarde des données dans un fichier XLSX
write.xlsx(m,file="heart-output.xlsx",row.names=F)

```

**Dr. Tahar Mehenni**