

# Éléments de traitement de statistiques avec le logiciel R

## Séance de TP N° 1 : Initiation à R

### Activité 1 : Commentaires

Le caractère '#' sert à ajouter des commentaires dans le programme : R néglige tout ce qui est après le caractère '#' et passe automatiquement à la ligne suivante.

Ecrire des commentaires de votre choix.

### Activité 2 : Opérations de base

#### 1. Tester les lignes de code suivante

```
> 10*9*8*7*6*5*4*3*2*1
> 70 / 1.84^2
> monPoids = 70
> maTaille = 1.84
> monPoids / maTaille^2
> log(2)/log(10)
```

#### 2. L'affectation dans l'autre sens

```
> x =10      # l'affectation par = ou bien par <-
> x <- 15
> 8 -> x      # l'affectation l'autre sens
> 7 = x      # !!! par l'affectation par = à un seul sens !!!
```

### Activité 3 : Variables

#### 1. Déclaration d'une variable:

```
> vec = 10    # en R nous n'avons pas besoin de déclarer une variable
> vec = 11    # la valeur 10 est automatiquement écrasée
> vec = "statistique" # le type est dynamique
```

#### 2. autres opérations sur les variables

```
> rm(vec)      # suppression d'une variable
> ls()         # lister toutes les variables
> rm(list = ls()) # suppression de toutes les variables
```

### Activité 4 : Manipulation de Vecteurs

#### 3. Création de vecteurs

##### a. Fonction de concaténation : c

Taper le code suivant :

```
> Vec1 = c(10,3,4)
> Vec2 = c("aaa","bbb") # Création d'un vecteur de caractères
> Vec3 = c("H","F")
> Vec3 = c(10, 3,4,5,6,10,100,100, 3,4,5,6,10,100,100,10,20, 30,40)
> vec4 = 1:10
```

##### b. Création d'un vecteur de booléen

```
> c( T, F)      # vecteur booléen TRUE FALSE
> c( T, F) + 0   # Conversion implicite du booléen en numérique
> c( T, F) + 1   # TRUE = 2 , FALSE =1
> c( T, F) + 2   # TRUE = 3 , FALSE =2
> c( T&T, F&T, F&F ) # tableau de vérité en utilisant l'opérateur ET
> c( T | T, F | T, F | F ) # même chose pour l'opérateur OU
```

### c. Autres techniques de création de vecteurs

```
> vec = 1:10
> vec = (1:10)*10
> vec = seq (10,100)
> vec <- seq (10,100, by = 10)
> vec <- seq(1,9,2)
> vec <- rep(c(0,6),3)
> vec <- rep(1:3,4)
> vec <- rep(1:3,each=4)
> vec <- rep(1:3,1:3)
> vec <- rep(1:3,3:1)
> vec <- c(10,3:6,10,rep(100,2), seq(10,40,by=10))
```

### d. Vecteur avec composantes manquantes

```
> vec1 = c(10,3,4,NA,6,10,100,NA,10,20, NA,40)
```

### e. Longueur d'un vecteur

```
> vec = c(10,3,4,0,6,10,0,100,3,4,5,6,0,100,100,10,20,0,40)
> length(vec)
> length(vec[vec==0]) # pour compter le nombre de 0 dans vec
```

### f. Ecrire dans un papier le résultat de l'expression ci-dessous sans faire la compilation

```
> vec1 = c(10,3,4, seq(1,9,2),rep(1:3,1:3),NA,6,10,100,NA,
rep(1:3,3:1),10,20, NA,40,seq(10,40,by=10))
```

## Activité 5 : Manipulation de Matrices

### g. Création d'une matrice

```
> vec <- 1:12
> matrix(vec,3)
> matrix(vec,2)
> matrix(vec, ,2)
> matrix(vec, ,2, byrow=T)
> mat<-matrix(1:15,ncol=5,byrow=T)
> mat<-matrix(1:15,5,byrow=T)
> mat<-matrix(1:15,ncol=5,byrow=F)
> mat<-matrix(1:15,nrow=5,byrow=T)
> mat<-matrix(c(1:3),ncol=2,nrow=3)
> mat<-matrix(c(1:3),ncol=6,nrow=5)
> diag(mat) # renvoie la diagonale d'une matrice
```

## Activité 6 : Manipulation de vecteurs / matrices

### h. Extraire d'un vecteur/ matrices

```
> vec <- c(10,3:6,10,rep(100,2), seq(10,40,by=10))
> a <- vec[2]
> B <- vec[2:4]
> x=c(1,4,5)
> C <- vec[x]
```

```
> mat<-matrix(c(1:3),ncol=6,nrow=5)mat[1,2]
> a <- mat[1,2]
> B <- mat[1,]
> B <- mat[,2]
> mat[,]
> mat[1:3,]
> mat[1:3,2]
```

```
> mat[1:3,2:4] # sous matrice d'une matrice
```

i. Modifier un vecteur/ matrices

```
> vec[2] <- 123  
> vec[2:3] <- c(111,222)
```

j. Supprimer une partie d'un vecteur

```
> vec[-2]  
> vec[-2:-4]
```

k. Opérations arithmétiques

Les opérations + , \* , - , / entre 2 vecteurs ou matrices de même dimension sont des opérations terme à terme

```
> vec2 <- vec + 1  
> vec[1:4]+1  
> (vec+1)[1:4]  
> x <- 1:4  
> y <- x * vec[1:4]  
> vec[2:3] <- -y  
> D <- vec[vec>0]
```

## Activité 6 : Data frame

l. Définition

C'est une matrice dont toutes les colonnes ne sont pas nécessairement du même type : scalaire, booléen, caractère. Par exemple

```
> c("F","M")  
> c("F",2)  
> data1= data.frame(x1=1,x2=1:10,a=letters [1:10])  
> data2= data.frame(x1=1,x2=1:10,sexe=c("F","M"))  
>  
> names( data1 ) # Pour visualiser le nom des colonnes  
> names(data1)<- c("c1","c2","c3")# Pour modifier le nom des colonnes  
>  
> head(data1) #Pour visualiser les premières lignes  
>  
> row.names(data1) #Pour visualiser le nom des lignes  
> row.names(data1)<-c("l1","l2","l3","l4","l5","l6","l7","l8","l9","l10")  
> #Pour définir le nom des lignes  
> dim(data1) #renvoie a dimension de data.frame  
>
```

### 1.1. Opération sur les data frame

```
> A = data.frame(x=1:3,y=2:4)  
> B = data.frame(xx=1,yy=1:3)  
> C= data.frame(x=1:3,y=rep("a",3))  
> A+B # somme élément par élément  
> A+C # rencontre un problème au niveau de la deuxième colonne  
> data.frame(A,B) # concaténer des dataframes ayant le même nombre de lignes  
>
```