

# Éléments de traitement de statistiques avec le logiciel R

## Séance de TP N° 2 : Liste & Data frame

### Activité 1 : complément pour les vecteurs

- Quelques vecteurs remarquables : R fournit quelques vecteurs particuliers qui sont directement accessibles :

- LETTERS : les 26 lettres de l'alphabet en majuscules

```
> LETTERS
[1] "A" "B" "C" "D" "E" "F" "G" "H" "I" "J" "K" "L" "M" "N" "O" "P"
[17] "Q" "R" "S" "T" "U" "V" "W" "X" "Y" "Z"
```

- letters : les 26 lettres de l'alphabet en minuscules

```
> letters
[1] "a" "b" "c" "d" "e" "f" "g" "h" "i" "j" "k" "l" "m" "n" "o" "p"
[17] "q" "r" "s" "t" "u" "v" "w" "x" "y" "z"
```

- month.name : les noms des 12 mois de l'année en anglais

```
> month.name
[1] "January" "February" "March" "April" "May"
[6] "June" "July" "August" "September" "October"
[11] "November" "December"
```

- month.abb : la version abrégée des 12 mois en anglais

```
> month.abb
[1] "Jan" "Feb" "Mar" "Apr" "May" "Jun" "Jul" "Aug" "Sep" "Oct"
[11] "Nov" "Dec"
```

- Indexation par position :

- Pour accéder à un élément d'un vecteur, il faut indiquer entre crochets sa position dans le vecteur :

```
> letters[2]
[1] "b"
> month.name[4]
[1] "April"
> LETTERS[3:5]
[1] "C" "D" "E"
```

- L'indice négatif sert pour supprimer l'élément dont la position est indiquée par la valeur absolue de l'indice

```
> month.name[c(-2, -4, -6, -8, -12)]
[1] "January" "March" "May" "July" "September"
[6] "October" "November"
```

- Indexation par nom

- Vous pouvez nommer les éléments d'un vecteur en utilisant la fonction `names()`

```
> x=1:3
> x
[1] 1 2 3
> names(x)=c("a", "b", "c")
> x
a b c
1 2 3
```

- Vous pouvez nommer les éléments d'un vecteur lors de sa création

```

> sexe = c(Michel = "h", Anne = "f", Dominique = NA, Jean = "h",
Claude = NA, Marie = "f")
>
> sexe["Alex"]
<NA>
NA
> sexe <- c(Michel = "h", Anne = "f", Dominique = NA, Jean = "h",
Claude = NA, Marie = "f")
> sexe["Michel"]
Michel
" h "
> sexe["Alex"]
<NA>
NA
> sexe["Alex"] = "f"
> sexe["Alex"]
Alex
" f "
> sexe
      Michel      Anne Dominique      Jean      Claude      Marie      Alex
      " h "      " f "         NA      " h "         NA      " f "      " f "

```

### Activité 1 : fonctions `mode()` et `length()`

- Le mode: il en existe quatre principaux: numérique, caractère, complexe et logique ; pour connaître le mode d'un objet, on utilise la fonction ***mode()***
- La longueur : nombre d'éléments de l'objet ; pour connaître la longueur d'un objet, on utilise la fonction ***length()***

- Fonctions : ***is.numeric()***, ***is.logical()***, ***is.character()***, ***is.factor()***

```

> a=4
> is.numeric(a)
[1] TRUE
> is.logical(a)
[1] FALSE
> is.character(a)
[1] FALSE
> is.factor(a)
[1] FALSE
>

```

- Fonctions : ***as.numeric()***, ***as.logical()***, ***as.character()***, ***as.factor()***

```

> a=4
> as.numeric(a)
[1] 4
> as.character(a)
[1] "4"
> as.factor(a)
[1] 4
Levels: 4
> as.logical(a)
[1] TRUE
>

```

### Activité 2 : List

Par nature, les vecteurs ne peuvent contenir que des valeurs de même type (numérique, textuel ou logique). Or, on peut avoir besoin de représenter des objets plus complexes composés d'éléments disparates. C'est ce que permettent les listes.

- **Création d'une liste**

Une liste se crée tout simplement avec la fonction ***list()*** :

```

> l1 = list(1:5, "abc")

```

```

> l1
[[1]]
[1] 1 2 3 4 5

[[2]]
[1] "abc"
> l2 = list(minuscles = letters, majuscles = LETTERS, mois =
month.name)
> l2

$minuscles
 [1] "a" "b" "c" "d" "e" "f" "g" "h" "i" "j" "k" "l" "m" "n"
[15] "o" "p" "q" "r" "s" "t" "u" "v" "w" "x" "y" "z"

$majuscles
 [1] "A" "B" "C" "D" "E" "F" "G" "H" "I" "J" "K" "L" "M" "N"
[15] "O" "P" "Q" "R" "S" "T" "U" "V" "W" "X" "Y" "Z"

$mois
 [1] "January" "February" "March" "April"
 [5] "May" "June" "July" "August"
 [9] "September" "October" "November" "December"

> length(l2)
[1] 3

> str(l)
List of 3
 $ minuscles: chr [1:26] "a" "b" "c" "d" ...
 $ majuscles: chr [1:26] "A" "B" "C" "D" ...
 $ mois      : chr [1:12] "January" "February" "March" "April" ...
>

```

- **Opération sur les listes : liste de liste et Fonction append()**

```

> listAlphabet = list(minuscles = letters, majuscles = LETTERS)
> listAlphabet
$minuscles
 [1] "a" "b" "c" "d" "e" "f" "g" "h" "i" "j" "k" "l" "m" "n" "o"
[16] "p" "q" "r" "s" "t" "u" "v" "w" "x" "y" "z"

$majuscles
 [1] "A" "B" "C" "D" "E" "F" "G" "H" "I" "J" "K" "L" "M" "N" "O"
[16] "P" "Q" "R" "S" "T" "U" "V" "W" "X" "Y" "Z"

> listMois = list(mois = month.name, moisAbb = month.abb)
> listMois
$mois
 [1] "January" "February" "March" "April" "May"
 [6] "June" "July" "August" "September" "October"
[11] "November" "December"

$moisAbb
 [1] "Jan" "Feb" "Mar" "Apr" "May" "Jun" "Jul" "Aug" "Sep" "Oct"
[11] "Nov" "Dec"

> list(listAlphabet, listMois)
[[1]]
[[1]]$minuscles
 [1] "a" "b" "c" "d" "e" "f" "g" "h" "i" "j" "k" "l" "m" "n" "o"
[16] "p" "q" "r" "s" "t" "u" "v" "w" "x" "y" "z"

[[1]]$majuscles
 [1] "A" "B" "C" "D" "E" "F" "G" "H" "I" "J" "K" "L" "M" "N" "O"
[16] "P" "Q" "R" "S" "T" "U" "V" "W" "X" "Y" "Z"

[[2]]

```

```

[[2]]$mois
[1] "January" "February" "March" "April" "May"
[6] "June" "July" "August" "September" "October"
[11] "November" "December"

[[2]]$moisAbb
[1] "Jan" "Feb" "Mar" "Apr" "May" "Jun" "Jul" "Aug" "Sep" "Oct"
[11] "Nov" "Dec"

> append(listAlphabet,listMois)
$minuscules
[1] "a" "b" "c" "d" "e" "f" "g" "h" "i" "j" "k" "l" "m" "n" "o"
[16] "p" "q" "r" "s" "t" "u" "v" "w" "x" "y" "z"

$majuscules
[1] "A" "B" "C" "D" "E" "F" "G" "H" "I" "J" "K" "L" "M" "N" "O"
[16] "P" "Q" "R" "S" "T" "U" "V" "W" "X" "Y" "Z"

$mois
[1] "January" "February" "March" "April" "May"
[6] "June" "July" "August" "September" "October"
[11] "November" "December"

$moisAbb
[1] "Jan" "Feb" "Mar" "Apr" "May" "Jun" "Jul" "Aug" "Sep" "Oct"
[11] "Nov" "Dec"

```

## Activité 2 : Data frame

Une data frame est une table ou une structure de type tableau à deux dimensions dans laquelle chaque colonne contient les valeurs d'une variable et chaque ligne contient un ensemble de valeurs de chaque colonne.

### 1. Création de data frame : fonction data.frame()

- data1= data.frame(x1=1,x2=1:10,a=letters [1:10])
- data2= data.frame(x1=1,x2=1:10,sexe=c("F","M"))

### 2. Opération sur les data frame

#### a. Nommer les lignes et les colonnes

- names(data1)<- c("c1","c2","c3") # Pour modifier le nom des colonnes
- names( data1 ) # Pour afficher le nom des colonnes
- 
- row.names(data1)<-c("l1","l2","l3","l4","l5","l6","l7","l8","l9","l10")  
#Pour définir le nom des lignes
- row.names(data1) #Pour afficher le nom des lignes
- 
- head(data1, nb) #Pour afficher les premières lignes
- 
- dim(data1) #renvoie la dimension de la data frame
- data1[,1] # la première colonne de data1
- data1[, "c1"] # la première colonne nommée c1 de data1
- data1\$c1 # la première colonne nommée c1 de data1
- 
- 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 data frames ayant le même nombre de lignes

### 3. Importation de fichiers

#### b. Changement de répertoire de travail

R utilise le répertoire de travail (le répertoire courant) pour afficher ou modifier le répertoire courant en utilise les fonctions *getwd* et *setwd*.

- `getwd()` # permet de connaître ce répertoire
- `setwd()` # permet de modifier le répertoire de travail  
# exemple : `setwd(« c:/data »)`
- 
- `WD <- getwd()`
- `if (!is.null(WD)) setwd(WD)`
- 

#### c. Importation de données

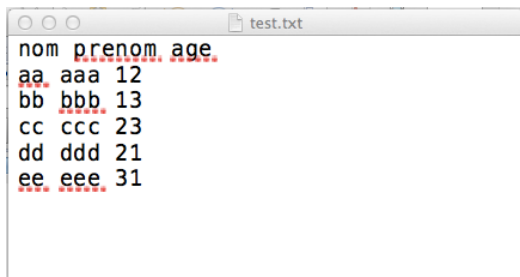
R peut lire des données stockées dans un fichier texte en conservant sa structure. Pour cela on utilise la fonction *read.table()*,

Dans le bureau créer une fichier texte (boc note) nommée « test.txt ». Dans le fichier écrire une phrase de votre choix, en suite appliquer la commande suivante

- `D= read.table("test.txt")`
- `D`

NB. Si la commande *read.table* ne fonctionne pas s'assurer vous que le fichier que vous voulez lire se trouve dans le répertoire courant ; pour cela on utilise les fonctions *setwd* et *getwd*.

Modifier le contenu du fichier test.txt comme suite :



charger à nouveau en appliquant la commande

- `D= read.table("test.txt")`
- `D`

NB : vous allez trouver que R n'a pas reconnu la première ligne comme ligne en-tête, pour cela il faut ajouter l'option : `header = TRUE`

- `D= read.table("test.txt", header = T)`
- `D`

- Découvrir d'autre option de la fonction *read.table* (*sep*, *dec*, *na.strings*, *comment.char*...) en consultant le help de cette fonction
- Vous pouvez utiliser la fonction *file.choose()* qui renvoie à la fonction *read.table* le nom de fichier à lire ; vous allez écrire directement la commande : `read.table(file.choose())`
- De la même façon utiliser la fonction *read.csv()* et la fonction *read.csv2* pour importer des fichier csv et des fichiers Excel.

