

---

## Séance de TP N°1 : Introduction au langage R

---

### 1. Présentation :

- R est un langage de programmation et un logiciel libre dédié aux statistiques et à la science des données soutenu par la R Foundation for Statistical Computing.
- Le langage R est largement utilisé par les statisticiens et les data miner pour le développement de logiciels statistiques et l'analyse des données.
- Le projet R naît en 1993 comme un projet de recherche de Ross Ihaka et Robert Gentleman à l'université d'Auckland (Nouvelle-Zélande)
- La version R 1.0.0, première version officielle du langage R, est publiée le 29 février 2000.
- En 2015, plusieurs acteurs économiques importants comme IBM, Microsoft ou encore la société RStudio créent le R Consortium pour soutenir la communauté R et financer des projets autour de ce langage.

### 2. Pour commencer avec R :

- **Démarrer R** : Vous lancez le logiciel R en cliquant sur l'icône R. Le symbole > signifie que R est prêt à travailler. Il ne faut pas taper ce symbole au clavier car il est déjà présent en début de ligne sur la R Console. C'est à la suite de ce symbole > que vous pourrez taper les commandes R. Une fois la commande tapée, vous devez toujours la valider par la touche Entrée.
- **Quitter R** : Pour quitter R, vous utilisez la commande

`>q()`

La question Save workspace image? [y/n/c] est posée : R propose de sauvegarder le travail effectué. Trois réponses possibles : y (pour yes), n (pour no) ou c (pour cancel, annuler). En tapant c, la procédure de fin de session sous R est annulée. Si vous tapez y, cela permet que les commandes tapées pendant la session soient conservées en mémoire et soient donc « rappelables » (mais vous ne pouvez pas les imprimer).

- **Sauvegarder sous R** : Si vous quittez R en choisissant la sauvegarde de l'espace de travail, deux fichiers sont créés :
  - (i) le fichier **.Rdata** contient des informations sur les variables utilisées,
  - (ii) le fichier **Rhistory** contient l'ensemble des commandes utilisées.
- **Travailler avec R** : Par exemple, tapez la commande suivante et validez :

`>2 + 5`

Le résultat s'affiche sous la forme :

`[1] 7`

Le chiffre 1 entre crochets indique l'indice du premier élément de la ligne 1, le second chiffre est le résultat de l'opération demandée.

- **Consulter l’aide de R** : Pour toutes les commandes, vous pouvez consulter une fiche de documentation en tapant, par exemple pour la commande **read.table** :

```
> ?read.table
```

Faire défiler le texte avec la touche **Entrée** ou **Flèche vers la bas**. Une fois arrivé à **END**, taper **q**. Grâce à cette aide, il suffit de retenir le nom de la commande, mais pas toute la syntaxe.

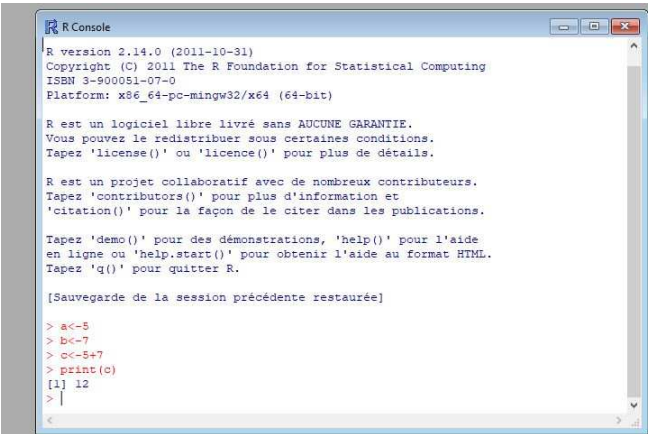
Vous pouvez rappeler les commandes déjà exécutées (pendant cette séance) en utilisant la touche **Flèche vers le haut**.

3. **Affectation et calcul** : R fonctionne un peu comme une calculatrice. Si vous tapez  $5+7$ , le logiciel vous retournera la valeur 5. Néanmoins, on utilisera R davantage comme un langage de programmation en suivant les principes de l’affectation informatique.

**Exemple d’Affectation avec R** :

```
a<-5
b<-7
c<-a+b
print(c)
```

L’affichage d’une variable avec R se fera alors en utilisant une fonction : « `print()` ».



```
R Console
R version 2.14.0 (2011-10-31)
Copyright (C) 2011 The R Foundation for Statistical Computing
ISBN 3-900051-07-0
Platform: x86_64-pc-mingw32/x64 (64-bit)

R est un logiciel libre livré sans AUCUNE GARANTIE.
Vous pouvez le redistribuer sous certaines conditions.
Tapez 'license()' ou 'licence()' pour plus de détails.

R est un projet collaboratif avec de nombreux contributeurs.
Tapez 'contributors()' pour plus d'information et
'citation()' pour la façon de le citer dans les publications.

Tapez 'demo()' pour des démonstrations, 'help()' pour l'aide
en ligne ou 'help.start()' pour obtenir l'aide au format HTML.
Tapez 'q()' pour quitter R.

[Sauvegarde de la session précédente restaurée]

> a<-5
> b<-7
> c<-a+b
> print(c)
[1] 12
> |
```

4. **Les Types de données** : Il existe de nombreux types de variables dans R.

**Les variables de type texte** :

```
A<-"Texte"
```

Ces variables peuvent être ordonnées dans une liste (un vecteur) ou dans plusieurs listes pour former une matrice (un tableau de valeurs).

**Les vecteurs et les matrices** :

```
B <- c(18, 182, 1.5, 15, 200, 5)
C <- matrix(c(18, 182, 1.5, 15, 200, 5), nrow = 2)
D <- matrix(c(18, 182, 1.5, 15, 200, 5), ncol = 2)
```

Pour accéder à une valeur ou à un ensemble de valeurs, il faut utiliser les index des vecteurs ou des matrices.

**Accès aux valeurs des vecteurs et des matrices :**

```
E <- B[2]+B[3]
F <- C[1,2] +C[2,3]
col <- C[,1]
ligne <- C[1,]
```

**Accès avancé aux valeurs des vecteurs et des matrices :**

```
e <-B[C(2,4)]
f <- C[(c<15)]
g <- B[2 :5]
```

Les data frames permettent de manipuler des tableaux bien structurés. Ce type de données est particulièrement bien adapté aux importations de fichiers textes.

**Les Data Frames :**

```
articles <- c( "un", "le", "la", "les")
sujets <- c( "mot", "terme", "chose", "images")
dfmots <- data.frame(articles , sujets)
dfmots2 <- data.frame(col1 = articles , col2 = sujets)
```

**Appel des valeurs des Data Frames :**

```
print(dfmots$sujets)
print(dfmots[,1])
```

**5. L'import de données et premières fonctions :**

**Importation de fichiers textes :**

```
MyTexte <- read.table(file="c :/TheData.csv", header=TRUE, sep=",")
MyData <- read.csv(file="c :/TheData.csv", header=TRUE, sep=",")
fichier <- file.choose()
```

**Fonctions de base :**

```
res <- summary(b)
plot(d[,1],d[,2])
hist(b)
reg <- lm(d[,1] ~d[,2])
res3 <- summary(reg)
t.test(d[,1], d[,2])
```

**6. Les bibliothèques :** Ce qui constitue la puissance de R, ce sont ses nombreuses bibliothèques qu'il faut télécharger.

**Les librairies cartographiques :**

```
library(rgdal)
nuts3 <- readOGR(dsn = "data", layer = "nuts3", verbose = TRUE)
library(sp)
class(nuts3)
```

```

nuts3@proj4string
head(nuts3@data)
plot(nuts3[1, ], col = "#5C99AD", border = " #2A5F70", lwd = 4)
library(rgeos)
europeBuffer <- gBuffer(spgeom = europe, width = 50000)

```

7. **Les boucles et la programmation :** Enfin, comme tout langage de programmation, R permet de répéter les mêmes instructions plusieurs fois en changeant seulement quelques paramètres. Ce sont les boucles. Ces boucles peuvent alors permettre d'effectuer des tests. Ce sont par exemple les Si.

**Les boucles :**

```

for (i in 1 :10) {
  print(i)
}
for (i in 1 :10) {
  if (i > 5 & i < 8) {
    print(i)
  }
}

```

8. **Utilisation de la fonction kmeans :** Les techniques de clustering font parties des techniques essentielles de l'analyse statistique, par conséquent R propose par défaut des fonctions de clustering, notamment la fonction `kmeans`.

**Avoir des informations sur la fonction kmeans :**

```
help(kmeans)
```

En entrée, il faut au minimum préciser :

- la matrice (ou la data frame) qui contient les variables sur lesquelles effectuer le clustering ;
- le nombre de clusters. (*Un cluster est une grappe de serveurs sur un réseau, appelé ferme ou grille de calcul ; un cluster Beowulf utilise des ordinateurs hétérogènes. Un cluster est un terme anglais qui désigne un bloc d'un système de fichiers.*)

En sortie, la fonction renvoie notamment les clusters associés à chaque objet, les centres des clusters.

9. **D'autres logiciels permettant de faire du clustering :**

**Scilab :** Logiciel libre, permettant de faire de nombreux calculs mathématiques. Il s'appuie sur un langage de programmation.

**MATLAB :** Logiciel propriétaire comparable à Scilab.

**Tanagra :** Logiciel gratuit de Data Mining spécialisé dans les méthodes de fouilles de données issues du domaine de la statistique exploratoire et de l'apprentissage automatique. Il s'appuie sur le principe de la programmation visuelle.

**Stata :** Logiciel libre comparable à R, spécialisé dans l'économétrie.

**SAS :** Logiciel propriétaire comparable à R.

**SPSS :** Logiciel propriétaire utilisé pour l'analyse statistique et s'appuyant sur une interface graphique comparable à Excel.

**XLSTAT :** Logiciel propriétaire qui propose des fonctions avancées par rapport à Excel. L'intégration entre les deux logiciels est parfaite.