Fiche TP1 : Introduction au logiciel R

1 Présentation du logiciel R

Le logiciel R est un outil informatique permettant de réaliser des analyses statistiques et de produire des graphiques. Il peut être également utilisé comme "simple" calculatrice et est également un langage de programmation complet.

Des informations (bibliothèques de fonctions, documents d'aide...) sur le logiciel R sont disponibles à l'adresse :

`http://www.r-project.org/`

et il est téléchargeable gratuitement à partir du site

`http://www.cran.r-project.org/`

ou de l'un des sites miroirs disponibles à cet endroit. Il est distribué *gratuitement* pour différents systèmes d'exploitation (Microsoft Windows, MacOS X, mais aussi de type Unix...).

Il existe plusieurs interfaces graphiques (**Rstudio**, RGUI, R Commander...) qui permettent d'accéder à une partie des fonctions de R et qui se téléchargent avec ou en même temps que le logiciel et certaines de ces interfaces s'accompagnent d'éditeurs de texte qui facilitent l'écriture et l'envoi des instructions et des programmes R.

2 Pour commencer

a. Pour démarrer R, cliquez sur l'icône `Rstudio` ou taper `Rstudio` à partir d'un invite de commandes (ou Terminal).

b. Tapez `getwd()` pour connaître quel est le répertoire de travail associé à votre session R.

c. Suivez les instructions de votre enseignant pour changer le répertoire de travail en choisissant le chemin dédié avec la souris.

d. Pour en savoir plus sur une fonction (par exemple la fonction `mean`, on consulte l'aide en tapant la commande `help(mean)` ou `? mean`.

e. La fenêtre en haut à gauche de `Rstudio` est un éditeur de fichier **script** dans lequel vous pouvez **écrire vos commandes, commentaires et les sauvegarder**.

f. **Il est préférable de travailler à partir du script.** On envoie ensuite les commandes à la session R pour qu'elles soient interprétées. Cet envoi se fait à l'aide des touches `Ctrl+Entrée` ou en cliquant sur `Run`.

g. Enfin pensez à **sauvegarder** vos travaux (votre script) régulièrement !

3 comme calculatrice

a. On peut effectuer de simples calculs avec comme avec une calculatrice, en utilisant les opérateurs `+,-,*,/` et des fonctions usuelles telles que `cos,sin,tan,exp,log`, mais aussi `sqrt` pour la racine carrée et `^` ou `**` pour notifier la puissance.

b. Le logiciel permet d'affecter à des variables (dont vous choisissez le nom) des résultats de calculs. L'affectation s'effectue grâce au symbole `<-` ou `=`. L'affichage de la valeur d'une variable s'opère en tapant simplement le nom de celle-ci. Pour bien comprendre comment fonctionnent les affectations, tapez par exemple les commandes suivantes :

```
x <- sqrt(3)
x
x = "a"
x
y <- x
y <- "toto"
x
y
```

4 Objets

Parmi les principaux objets on trouve les vecteurs, les matrices, les `data.frame` (jeux de données mises sous forme de colonnes, autrement dit matrices sans contrainte d'uniformité de type pour les éléments des colonnes).

Ces objets contiennent des valeurs qui peuvent être de différents types :

- ⋆ entier, réel, complexe ;
- ⋆ chaîne de caractères ;
- ⋆ booléen : `TRUE` ou `T` et `FALSE` ou `F`.
- ⋆ `NA` (Not Available).

4.1 Vecteurs

a. Pour créer des vecteurs, on utilise généralement la fonction `c`, mais on peut également avoir recours aux fonctions `seq`, `rep` ou à l'opérateur binaire `:`. La fonction `c` permet aussi de concaténer plusieurs vecteurs. Testez par exemple les commandes suivantes en modifiant ensuite certaines valeurs ; **tapez le nom du vecteur pour voir ses éléments** :

```
vec1 <- c(3,4.6,1.2,-7.8)
vec2 <- -4:7
```

```
vec3 <- seq(2,18,3)
vec4 <- rep(vec2,3)
vec5 <- c(vec1,vec2,vec3)
couleur <- c("vert","rouge","jaune","bleu")
vec6 <- c(T,F,T,T,F)
length(vec3)
```

Appliquez les fonctions `mode`, `length` à ces vecteurs.

b. Les opérations sur les vecteurs se font à l'aide des opérateurs et des fonctions classiques déjà mentionnées. Remarquez que les calculs sont effectués éléments par éléments :

```
vec1 + 10
vec2 * (-2)
vec1 * vec2
```

c. L'extraction ou le remplacement de certains éléments (même sous condition) d'un vecteur est possible (**tapez le nom du vecteur en question avant et après la commande pour constater les modifications**) :

```
vec1[3]
vec1[c(1,3)]
vec1[3] <- 33
vec3 <- vec3[-2]
vec4[vec4==5] <-1
vec2[vec2>=6] <-10
```

d. Pour trier des vecteurs par ordre croissant ou décroissant, on emploie les fonctions `sort`, `rev` et la fonction `rank` peut s'avérer utile. On peut également sommer progressivement les éléments d'un vecteur à l'aide de la fonction `cumsum`.

```
sort(vec1)
rank(v1)
cumsum(vec2)
rev(cumsum(vec2))
```

4.2 Matrices

a. On crée généralement une matrice à l'aide de la fonction `matrix` et à partir des éléments d'un vecteur comme dans les exemples ci-dessous :

```
vec4
mat1 <- matrix(vec4,ncol=3,byrow=TRUE)
mat2 <- matrix(vec4,ncol=4,byrow=FALSE)
```

b. Les fonctions `nrow` et `ncol` permettent respectivement la récupération du nombre de lignes et de colonnes d'une matrice ; la fonction `dim` affiche, quant à elle, ces deux informations.

c. Le stockage en mémoire des matrices se fait en mode "colonne majeure" (comme en Fortran) alors que Python (et le C) utilisent le mode "ligne majeure" :

```
A = matrix(1:9,ncol=3)
A / c(1,2,3) # À ne pas faire si on souhaite diviser
               la colonne 1 par 1, la colonne 2 par 2, etc
t(t(A)/c(1,2,3)) # À faire si on souhaite diviser
                   la colonne 1 par 1, la colonne 2 par 2, etc
```

4.3 Jeux de données ou `data.frame`

4.3.1 Depuis un fichier texte

Dans un fichier texte, saisir les deux colonnes suivantes, et les sauvegarder dans un fichier "table.txt" :

```
Note1 Note2
12    18.2
14     7.1
18.3  15.8
 9.1  10.1
 5.7   7.4
```

a. Importer le fichier que vous venez de créer en utilisant l'une des commandes
   `tab <- read.table("table.txt",header=TRUE)`

b. Tester les commandes suivantes :

```
tab
head(tab)
tab$Note1
tab[4,2]
tab[,2]
```

4.3.2 Directement dans R

Il est possible de créer un jeu de données par lecture d'un fichier de données (au format `.csv`) ou directement sous R en utilisant la fonction `data.frame` comme décrit dans l'exemple suivant :

```
taille<-c(1.40,1.67,1.90,2.14)
poids<-c(54,62,75,91)
danger<-c("extrême","forte","forte","faible")
tab1<-data.frame(taille,poids,danger,row.names=noms)
tab1
```

4.3.3 Depuis un tableur

On peut importer un jeu de données à partir d'un fichier `Excel` ou `OpenOffice`. Il faut au préalable sauvegarder le fichier au format `.csv` dans le répertoire de travail en précisant le type d'indicateur

décimal utilisé (généralement le caractère , en Français) et le séparateur de champs (par exemple ;). On transcrit ce fichier en `data.frame` à l'aide de la commande `read.csv`.
Par exemple, importer les données du fichier `oiseaux.csv`. Ces données ont été recueillies en février 1968 par Herman Bumpus sur des oiseaux échoués à Rhode Island. Il mesura sur les volatiles des longueurs caractéristiques (en `mm`) :

- ⋆ LOT : longueur totale de l'oiseau
- ⋆ AIL : envergure des ailes
- ⋆ TET : longueur de la tête et du bec
- ⋆ HUM : longueur de l'humérus
- ⋆ BRE : longueur du bréchet

Pour ce faire, taper les commandes suivantes :

```
oiseaux<-read.csv("oiseaux.csv",header=TRUE,dec=",",sep=";")
oiseaux
head(oiseaux)
```

L'option `header` permet de signifier à R si les colonnes contiennent un intitulé sur leur première ligne. Les options `dec` et `sep` précisent le type d'indicateur décimal et le séparateur de champs respectivement (notez bien l'utilisation des guillemets pour écrire ces caractères).

a. On peut extraire des éléments du jeu de données ou même des colonnes entières :

```
oiseaux$TET
oiseaux[4,3]
oiseaux[,2]
```

b. Enfin, la fonction `attach` permet de travailler sur les colonnes d'un jeu de données sans écrire son nom.

```
attach(oiseaux)
HUM
LOT
```

5 Indicateurs numériques

a. Calculer les différents indicateurs numériques des variables du jeu de données `oiseaux` :

- ⋆ moyenne (fonction `mean`),
- ⋆ variance corrigée (fonction `var`),
- ⋆ écart-type corrigé (fonction `sd`),
- ⋆ médiane (fonction `median`).

b. Retrouver par le calcul la valeur de la médiane de la variable `HUM`.

c. Déterminer par le calcul le quantile d'ordre 0.8 de la variable `HUM`. Vérifier le résultat à l'aide de la fonction `quantile`.

d. Déterminer par le calcul le quantile d'ordre 0.845 de la variable `BRE`. Vérifier le résultat à l'aide de la fonction `quantile`.