**L3 EBO**

**TP1 Exploration et analyse des données environnementales**

**Manipulation et présentation des données avec R**

**1) Quelques rappels**

- Dans R, tout ce qui suit le caractère # est un commentaire et n’est pas pris en compte par R.

- La commande "help" permet d'afficher l'aide pour une fonction particulière :

> help(sort) : aide sur la fonction "sort"

- Pour charger le fichier, il faut dire à R où il se trouve :

→ Pour savoir dans quel répertoire on se trouve, utiliser getwd() pour "get working directory"

→ Pour changer de répertoire : setwd("chemin du fichier") pour "set working directory"

Vous pouvez aussi spécifier un dossier de travail dans lequel vous travaillerez une bonne fois pour toute : il faut utiliser la commande « changer de répertoire de travail » dans le menu de R.

**2) Types de données**

R peut manipuler :

* des nombres entiers
* des décimaux
* des chaînes de caractères
* des booléens (ex : étudiant=TRUE)

La valeur NA (Non Available) est utilisée lorsqu’une donnée est manquante.

**3) Les Tableaux**

a) Les Vecteurs :

- Un vecteur est un tableau à une dimension. Toutes les cases du vecteur doivent contenir des données du même type (des entiers, des chaînes de caractères,...).

- La fonction c() permet de créer un vecteur : *créez le vecteur suivant*:

> ages <- c(28, 25, 23, 24, 26, 23, 21, 22, 24, 29, 24, 26, 31, 28, 27, 24, 23, 25, 27, 25, 24, 21, 24, 23, 25, 31, 28, 27, 24, 23)

puis *tapez* > ages.

*Regardez ce qu’il se passe.*

*Que représente le nombre qui s’affiche au début de chaque ligne entre crochets ? Est-ce un élément du vecteur ?*

- Pour créer un vecteur contenant une suite de nombres entiers de 2 à 6, on peut utiliser la fonction "seq". Par exemple : > seq(2,6).

*Tapez cette ligne et regardez ce que R vous donne en sortie*.

*Tapez*: > seq(1,10, by=0.5). *Que fait R ?*

- La fonction c() permet aussi de concaténer (= mettre bout à bout) des vecteurs.

*Créez 2 vecteurs comme vu ci-dessus.* Vous leur donnerez un nom à chacun, par exemple : >vecteur1<-c(2,5).

*Ensuite, utilisez la fonction c() pour créer un troisième vecteur qui sera les 2 premiers vecteurs mis bout à bout.*

- Il est possible d’accéder à un élément du vecteur avec des crochets. Par exemple pour accéder au second élément du vecteur1 :

> vecteur1[2].

*Essayez et vérifiez*.

- Il est aussi possible d’accéder à l’ensemble des nombres répondant à une condition, par exemple l’ensemble des nombres supérieurs à la valeur x :

> vecteur1[vecteur1> x].

*Essayez et vérifiez*.

- Enfin, la fonction length() permet de connaître la longueur d’un vecteur (aussi le nombre d’éléments d’un tableau) :

> length(vecteur1)

aussi :

> length(vecteur1[vecteur1 > x])

*Essayez et vérifiez*.

b) Les Matrices :

- Une matrice est un tableau à deux dimensions. Comme pour les vecteurs, toutes les cases d’une matrice doivent contenir des données du même type. Une matrice a un nombre de ligne (« nr », pour Number of Row) et de colonne (« nc », pour Number of Column).

*Créez la matrice suivante :*

> matrice <- matrix(c(1.5, 2.1, 3.2, 1.6, 1.4, 1.5),nr=3, nc=2)

*Tapez* > matrice et regardez ce que R fait.

- Les éléments de la matrice peuvent être accédés en donnant entre crochets le numéro de la ligne **puis** celui de la colonne :

> matrice[1, 1]

Essayez et vérifiez.

- Il est aussi possible de récupérer une ligne ou une colonne entière, en omettant le numéro correspondant :

> matrice[1,]

*Essayez et vérifiez.*

- matrice[2:3,1:2] donne une « sous-matrice » correspondant aux éléments de la 2ème et 3ème ligne, 1ère et 2ème colonne.

*Essayez et vérifiez.*

c) Les Listes :

Une liste est un tableau à une dimension, qui peut contenir des données **de différents types** (contrairement au vecteur).

> liste <- list("AH", 55)

*Essayez et vérifiez.*

d) Les Tableaux de données (data.frame en anglais) :

- Un tableau de données est un tableau où chaque colonne correspond à un attribut différent (âge, taille, poids par exemple) et chaque ligne à un individu différent. Il peut y avoir des éléments manquant dans le tableau.

- Pour charger un fichier dans R :

> nom\_tableau <- read.table("fichier.csv", sep=",", header=TRUE)

sep=”,” indique que dans le fichiers, les différentes colonnes sont séparées par des virgules, et header=TRUE indique que la première ligne du fichier contient les noms des colonnes. Ces valeurs correspondent aux fichiers CSV (Comma-Separated Value file : fichier dont les valeurs sont séparées par des virgules). Dans le cas de fichiers txt, il faut mettre sep= "\t" pour tabulation.

*Essayez d’ouvrir le fichier trees.csv puis le fichier trees.txt* (Rappel : pour charger le fichier, vous pouvez soit donner tout le chemin d’accès soit spécifier dès le départ votre répertoire de travail).

La colonne « Girth » donne la circonférence des arbres, la colonne « Height » donne la hauteur des arbres, la colonne « Volume » donne le volume des arbres et la colonne « Species » donne les espèces. *Vous utiliserez ce fichier pour essayer les commandes suivantes et voir l’utilité de chacune*:

>names(nom\_tableau)

>ls()

>dim(nom\_tableau)

>summary(nom\_tableau)

Vous pouvez également ajouter la commande : "row.names=1" pour spécifier si la colonne 1 n’est pas une variable mais l’identifiant des lignes.

*Ouvrez le fichier trees\_modif.txt sans utiliser la fonction row.names=1. Quelle est la différence avec le fichier trees.csv ?*

*Réessayez d’ouvrir le fichier trees\_modif.txt en ajoutant row.names=1 dans la commande read.table. Observez la différence.*

- Comme pour les matrices, il est possible d’accéder aux cases, lignes et colonnes d’un tableau. Les noms des colonnes peuvent être utilisés à la place de leurs index.

*Essayez les commandes suivantes et voir l’utilité de chacune:*

> nom\_tableau[1, 2]

> nom\_tableau [1, "Height"]

> nom\_tableau [1,]

> nom\_tableau [,"Girth "]

- Avec un tableau, pour identifier une variable, il faut la nommer : nom\_tableau$nom\_colonne

Par exemple :

> nom\_tableau$col1 : col1 est la première colonne du tableau. *Essayez* :

> nom-tableau$Girth

- Il est aussi possible d’indexer avec une condition : par exemple, pour obtenir un tableau avec seulement les valeurs de Girth supérieures à 12 (ne pas oublier la virgule, qui sert à indiquer que l’on veut récupérer des lignes !).

*Essayez et regardez l’utilité des commandes suivantes :*

> nom\_tableau [nom\_tableau $Girth > 12,]

>nom\_tableau [nom\_tableau $Girth == 12,]

>subset(nom\_tableau, Girth==12.9)

>sub<- nom\_tableau [nom\_tableau $Girth == 12,]

- On peut également définir « par défaut » le tableau sur lequel on va travailler et ne plus avoir besoin du $ pour nommer les variables. Pour cela, la fonction :

> attach(nom\_tableau) permet de définir le tableau “par défaut” sur lequel porteront les analyses.

*Utilisez cette fonction avec l’un des tableaux ouverts précédemment.*

Ensuite, il est possible d’accéder aux colonnes du tableau directement par leur nom.

*Essayez d’afficher la colonne Girth.*

La fonction inverse est : detach(nom\_tableau).

**4) Quelques fonctions utiles :**

>summary(nom\_tableau) : affiche un résumé du tableau

>ncol(nom\_tableau) : donne le nombre de colonne d’un tableau

>nrow(nom\_tableau) : donne le nombre de ligne d’un tableau

>q() : quitte R

>mean(nom\_variable) : calcule la moyenne

>median(nom\_variable) : calcule la médiane

>var(nom\_variable) : calcule la variance

>sd(nom\_variable) calcule la déviation standard (=écart-type)

Parfois, il faut demander à R de ne pas tenir compte des valeurs NA pour ce calcul :

> mean(nom\_variable, na.rm = TRUE)

**5) Représentions graphiques (avec le fichier trees.csv):**

a) Utilisation de la fonction « plot » :

*Tapez :*

> plot(Species) : *que représente ce graphe ?*

*Tapez :*

> plot(Volume~ Height) : *qu’indique le symbole « ~ »?*

*Tapez :*

> plot(Girth~ Species) : *que représente ce graphe ?*

Mise en forme des graphiques :

Il existe de nombreuses options à mettre dans la fonction « plot » pour mettre en forme les graphiques. En voici quelques-unes :

**main** : titre du graphique

**xlab** : titre de l’axe X

**ylab** : titre de l’axe Y

*Faites un graphique* ***complet*** *du volume des arbres en fonction de leur hauteur.*

b) Utilisation de la fonction « hist » :

*Tapez :*

> hist(Girth)

*Décrivez ce graphe. Que représente l’axe des ordonnées nommé « Frequency » ?*

*Tapez :*

> hist(Girth, freq=F)

*Décrivez ce graphe. Que représente l’axe des ordonnées nommé « Density » ?*

c) Utilisation de la fonction « boxplot » :

*Tapez :*

> boxplot(Girth)

*Que représente ce graphe ?*

*Tapez :*

> boxplot(Girth~Species)

*Que représente ce graphe ? Ajoutez-lui un titre et le nom des axes. Ne l’avez-vous pas déjà croisé?*

Pour cela, vous allez afficher les 2 graphes en question. Pour ce faire, c’est à dire mettre plusieurs graphiques dans la même fenêtre, il faut préciser comment on veut les organiser avec la fonction suivante:

>par(mfrow =c(nombre\_lignes, nombre colonnes)).

Ici, vous voulez afficher 2 graphes :

*Tapez :*

>par(mfrow =c(1, 2))

*Puis tapez les lignes de commande pour afficher vos 2 graphes. Quelles sont les différences si il y en a ?*

Lorsqu’on a plusieurs variables INDÉPENDANTES, il est possible de placer plusieurs boîtes à moustaches les unes à côté des autres, en séparant les variables par des virgules dans la fonction boxplot.

*Affichez sur un même boxplot les variables Height, Volume et Girth. Pourquoi la variable Species n’est pas demandée ?*

*Que manque-t-il sur l’axe des abscisses ?*

*Pour remédier à cela, procédez comme suit :*

*Créez un vecteur comportant les noms, dans le bon ordre, de vos labels. Ce vecteur doit avoir un nom d’objet. Puis, refaites votre boxplot en ajoutant dans la commande, names=nom\_vecteur\_labels.*

Exercice :

Nous souhaitons étudier l’efficacité de trois herbicides sur trois plantes : blé, chiendent et liseron. Pour cela, des cultures de ces plantes ont été mises en présence de l’un des trois herbicides, ou d’aucun d’entre eux. Le nombre de plants vivants dans la culture a été compté avant l’expérience, et 10 jours après. Chaque combinaison plante - herbicide a fait l’objet de 20 expérimentations, plus un témoin sans herbicide (soit 240 expérimentations en tout).

Le tableau de données est disponible dans le fichier herbicide.csv.

1. Charger le fichier herbicide.csv et afficher les données.

2. Combien de variables composent ce jeu de données ? Quelle est la nature de ces variables ? Quelles sont les dimensions de ce jeu de données ?

3. Créer 4 objets nommés « sans », « herb1 », « herb2 » et « herb3 » et contenant respectivement uniquement les données sans herbicide, avec herbicide 1, avec herbicide 2 et avec herbicide 3.

4. Tracer la boîte à moustaches de la variable « survivants ». Ce graphe aura des plots représentant chacun des 4 objets créés précédemment. Il devra avoir un titre général, des titres pour les axes et des labels sous chaque plot.

Faire figurer la moyenne sur chacune des boîtes. Pour cela, utiliser la ligne de commande suivante :

>points(1, mean(nom\_variable), pch=16)

« 1 » est la valeur de l’abscisse où R met le point, « mean » calcule la moyenne et « pch » définit le symbole.

Regardez ce qui se passe à la suite de cette ligne de commande et répéter pour chaque boîte.