Séances d'initiation ou de remise à niveau en Statistique sur R

Introduction au logiciel R

Prise en main de R

Manipulation de jeux de données

- Savoir se placer dans un répertoire de travail ?
- Importation d'un jeu de données texte/CSV/Excel

Application sur un cas pratique

Alseny NIARE : Ingénieur statisticien à la Plate forme Universitaire de Données de Caen (PUDC)

A. Introduction au logiciel R

Qu'est-ce que le logiciel R ?

R est un logiciel permettant de faire des analyses statistiques et de produire des graphiques. Mais R est également un langage de programmation complet, c'est cet aspect qui fait que R est différent des autres logiciels statistiques. Les informations sur R sont disponibles sur la homepage du projet :

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

• R fonctionne avec plusieurs fenêtres sous Windows. En particulier nous distinguons la fenêtre **R console**, c'est- à-dire la fenêtre principale où sont réalisées par défaut les entrées de commandes et sorties de résultats en mode texte.

• **Le menu File ou Fichier** contient les outils nécessaires à la gestion de l'espace de travail, tels que la sélection du répertoire par défaut, le chargement de fichiers sources externes, la sauvegarde et le chargement d'historiques de commandes,. . .

• **Le menu Edit ou Edition** contient les commandes habituelles de copier-coller, ainsi que la boîte de dialogue autorisant la personnalisation de l'apparence de l'interface.

• **Le menu Misc** traite de la gestion des objets en mémoire et permet d'arrêter une procédure en cours de traitement.

• **Le menu Packages** automatise la gestion et le suivi des librairies de fonctions, permettant leur installation et leur mise à jour de manière transparente au départ du site CRAN (Comprehensive R Archive Network) ou de toute autre source locale.

• Enfin, **les menus Windows** (ou Fenêtres) et **Help** (ou Aide) assument des fonctions similaires à celles qu'ils occupent dans les autres applications Windows à savoir la définition spatiale des fenêtres et l'accès en ligne et aux manuels de références du logiciel R.

• Ce qui est entré par l'utilisateur figure en rouge et la réponse de R est en bleu.

• Les nombre entre crochets au début de chaque ligne donnent l'indice du premier nombre de la ligne.

Objectifs

Prendre en main, puis utiliser les fonctionnalités statistiques de base du logiciel R.
A l'issu de ce cours, l'utilisateur sera capable d'utiliser le logiciel R de façon automne, pour mener des analyses statistiques de base, ainsi que connaître les outils nécessaires à des analyses plus complexes.

B. Prise en main de R

1. Téléchargement et installation

Windows : la version pour systèmes d'exploitation Windows est téléchargeable à l'adresse http://cran.univ-lyon1.fr/bin/windows/base/ en cliquant sur le lien Download R 3.2.5 for Windows.

2. **Premiers pas avec R**

Objectifs

- Découvrir l'environnement de développement de R
- Acquérir les bases du langage R
- Comprendre l'utilité d'un script et savoir gérer ses sauvegardes

2.1 **R :Un petit calculateur**

Commençons par prendre en main la console de RStudio pour y saisir quelques données et réaliser quelques opérations. Si la console contient déjà des instructions, il est possible de les effacer :

- soit par la combinaison de touches: Ctrl + L
- soit par le menu Edition ou Edit : Effacer la console ou Clear console

Une fois la console vidée, tapons un certain nombre d'instructions comme indiqué ci-dessous.
Pour exécuter une instruction, il suffit de la taper dans la console (par exemple 15*32) et de valider par la touche Entrée. Le résultat apparaît sur la ligne suivante (480).

Remarque :

- La première ligne représente le code entrée dans la console
- La seconde ligne affiche le résultat de l'opération demandée, le chiffre 1 entre crochets indique l'indice du premier élément de la ligne.

Exemple :

```
> 15*32
[1] 480

> log(2)
[1] 0.6931472

> 100/3
[1] 33.33333
```

Consulter l'aide de R : Pour toutes les commandes, vous pouvez consulter une fiche de documentation en tapant, par exemple:

help(read.table) ou ?read.table
help(log) ou ?log

La fonction *help(fonction)* ou *?fonction* permet d'accéder à l'aide.

Remarque :

Notons au passage que le symbole décimal est le point et non la virgule.
Si vous voulez répéter une instruction déjà tapée, vous pouvez utiliser la flèche du haut.

2.2 La gestion des variables

Variables élémentaires

Une variable permet de stocker une valeur et de réaliser du calcul formel**. "<−"** est une instruction d'affectation. Elle est équivalente à l'instruction "=" utilisée dans la seconde instruction.

Exemple :

La première instruction permet d'affecter la valeur 2 à la variable x et la deuxième la valeur 3 à la variable y.

```
> x<-2
> y=3
```

Les variables x et y sont alors utilisées pour réaliser plusieurs calculs formels dont le résultat est soit affiché,

```
> 1/x
[1] 0.5
```

soit stocké dans une nouvelle variable z.

```
> z=x+y-5
```

Le contenu (ou valeur) de la variable peut être affiché ou réutilisé dans une expression.

Exemple :

L'expression z=z+4 permet d'ajouter 4 à la valeur de et de stocker le résultat dans la même variable z.

```
> z
[1] 0

> z=z+4
> z
[1] 4
```

Chaîne de caractères

Un autre type de donnée est également utilisé ici, ce sont les chaînes de caractères. R permet d'affecter une chaîne de caractère à une nouvelle variable.

Exemple :

chaîne= "bonjour"
La variable chaîne s'est vu affecter la chaîne de caractères "Bonjour".

La fonction *paste* permet de concaténer des chaînes de caractères entre elles.

Exemple :

```
> chaîne= paste (chaîne,"tout le monde!")
> chaîne

[1] "Bonjour tout le monde!"
```

Variables vectorielles et matrices

Création d'un vecteur : c ( )

Il est également possible de manipuler des vecteurs (c'est-à-dire des séries de valeurs) et de les stocker dans des variables. C'est une propriété essentielle du langage R car elle va nous permettre de manipuler des variables au sens statistique du terme.

La fonction c( ) permet ici d'initialiser le vecteur. Le vecteur peut contenir des données numériques (variable x) ou qualitatives (variable lettres).

Exemple :

Définissons ici une variable comme un vecteur de 4 valeurs numériques et une variable lettres comme un vecteur de 4 valeurs qualitatives.

```
> x = c(1,4,6,3)
> x
[1] 1 4 6 3

> lettres = c("A","B","C","D")
> lettres
[1] "A" "B" "C" "D"
```

Dans le cas de vecteurs de valeurs numériques, il est possible de leur appliquer des opérations arithmétiques :

Exemple :

L'instruction x-2 permet de soustraire la valeur 2 à chaque élément de x tandis que l'instruction x^2 élève chacun des éléments au carré.

```
> x^2
[1]  1 16 36  9
```

Exemple :

Il est également possible d'effectuer des opérations sur plusieurs vecteurs.

```
> y=c(2,3,1,7)
> z=c(1,2,3,5)
> y+z
[1]  3  5  4 12
```

La taille d'un vecteur peut être obtenue par l'instruction length( ).

Exemple :

```
> length(z)
[1] 4
```

Création d'une matrice : *matrix (... , nrow = , ncol = , byrow =)*

Le langage R donne aussi la possibilité de manipuler des matrices (tableau de données multidimensionnel). L'instruction *matrix( )* permet de définir une matrice à partir d'un vecteur. Les options *nrow* et *ncol* indiquent le nombre de lignes et de colonnes. L'option *byrow* permet quant à elle d'indiquer que les données sont organisées par lignes. Par défaut, *byrow = FALSE*.

Exemple :

```
> M= matrix(c (1,2,3,4,5,6), nrow = 2, ncol = 3, byrow = TRUE)
> M
     [,1] [,2] [,3]
[1,]    1    2    3
[2,]    4    5    6
```

La taille d'une matrice peut être récupérée à l'aide de la fonction *dim( )*. Le résultat donne un vecteur à deux valeurs: le nombre de lignes suivi du nombre de colonnes.

Exemple :

```
> dim(M)
[1] 2 3
```

Remarque :

L'utilisation de matrices restera toutefois marginale. La manipulation de tableaux de données se fera par le biais de *dataframe* (ou jeux de données).

C. Manipulation de jeux de données

Objectifs

- Savoir se placer dans un répertoire de travail
- Savoir importer un jeux de données avec R
- Savoir manipuler ce jeu de données : gérer les variables et les individus
- Être capable d'extraire un sous-ensemble de données

1. Savoir se placer dans un répertoire de travail

Quand les données sont plus volumineuses, il n'est pas très conseillé d'utiliser R comme outil de saisie. Dans ce cas, vous pouvez utiliser un éditeur de texte ou un tableur quelconque pour saisir vos données (Excel par exemple) et le transférer ensuite sous R. Il est nécessaire d'indiquer au logiciel R l'endroit où sont stockés les fichiers de données. Ceci peut être fait soit à chaque chargement de fichier soit pour la durée de chaque utilisation du logiciel. Pour connaître le répertoire de travail actuellement utilisé par R, qui est par défaut le répertoire où le logiciel est installé, il suffit de taper l'instruction suivante :

```
> getwd( )
```

Pour changer le répertoire de travail par défaut, pour la durée de la session R, pour par exemple le répertoire "Z :/Bureau/R", il suffit de taper :

```
> setwd(normalizePath ("Z:/Bureau/R"))
```

normalizePath( ) permet de standardiser automatiquement les chemins d'accès selon le système d'exploitation que l'on utilise (Windows, Unix).

Nous pouvons également changer de répertoire de travail en se plaçant dans la console R : menu Fichier (changer le répertoire courant) puis choisir le chemin d'accès à votre fichier.

2. Importation d'un jeu de données texte/CSV/Excel

2.1 Importation d'un jeu de données texte : read.table( )

Cas pratiques : Sur le bureau, vous trouverez un dossier R contenant un fichier aux format texte (.txt) Melons.txt, un fichier au format csv (.csv) Melons.csv et un fichier au format excel (.xls) Melons.xls. Conservons le même répertoire de travail "Z:/Bureau/R".

La commande suivante permet d'importer les données de mon fichier texte Melons.txt

```
> Melons= read.table("Melons.txt",header = TRUE, sep=";", dec=",")
> Melons
ou
> Melons= read.table("Z:/ Bureau/ R / Melons.txt",header = TRUE, sep=";", dec=",")
> Melons
```

Le nom du fichier le chemin du répertoire de travail sont entre guillemets (" ").

header = TRUE permet de conserver le nom exact des variables de ma table, la première ligne est considérée comme nom de la variable.
Sep= ";" indique que le caractère qui sépare les colonnes est ici ";".
dec= "," indique que le séparateur décimal est le caractère ","
quote= " \" (non utilisée dans notre exemple) permet de contrôler les caractères qui entourent éventuellement une chaîne de caractère.

2.2 Importation d'un jeu de données Excel au format csv : read.csv2( )

La commande suivante permet d'importer les données de mon fichier csv Melons.csv

```
> Melons=read.csv2("Z:/ Bureau/ R/ Melons.csv")
> Melons
```

3. Manipulation d'un jeu de données

3.1 Éditer, accéder à un individu, à une variable

Voyons à présent comment gérer les données une fois qu'elles ont été importées.

Comment visualiser et éditer les données : ***View( )*** *et* ***edit( )***

- View( ) permet de de visualiser le jeu de données
- edit( ) sert à ouvrir les données dans un éditeur de texte et à modifier les données
- fix( ) ouvre le même éditeur, permet de modifier les données et remplace l'objet (Melons ici) par le jeu de données modifié

Exemple

```
> edit(Melons)
> fix(Melons)
```

Remarque:

Lorsque vous utilisez les fonction fix( ) ou edit( ), il est impossible de reprendre la main sur la console R tant que l'éditeur n'est pas refermé. Une fois refermé, le contenu du jeu de données est affiché sur la console. Si vous voulez créer un nouveau jeu de données modifié et conservé l'ancien, vous pouvez utilisez la fonction edit( ) comme ceci:

```
> Melons_modif <- edit(Melons)
```

3.2 Extraire un sous ensemble de données

Accéder à un sous ensemble de variables et/ou d'individus: [,]

Pour accéder à un sous ensemble, on utilise les [,] : avant la virgule pour les individus et après la virgule pour les variables

Exemple:

- Une partie des individus

```
> Melons[c(1:10, 12, 15) , ]
```

désigne le sous-ensemble des données constitué des individus 1 à 10, 12 et 15 mais de toutes les variables (rien n'est mentionné après dernière la virgule).

- Une partie des variables

```
> Melons[ , c(1,3,8:10) ]
```

désigne le sous-ensemble constitué de tous les individus (rien n'est mentionné avant la première virgule) mais uniquement pour les variables 1, 3 et de 8 à 10.

- Une partie des variables et des individus

```
> Melons[c(1:10, 12, 15), c(1,3,8:10)]
```

Il est souvent utile de pouvoir extraire une variable particulière pour y appliquer une opération.

Exemple:

```
> Melons$Rdt
```

L'instruction `Melons$Rdt` permet d'extraire la colonne Rdt (Rendement). Nous remarquons que cette série de données contient beaucoup de données manquantes (NA). Il est toutefois possible de supprimer les valeurs manquantes. Cela se fait par l'option `na.omit( )`.

Exemple:

```
rendement= na.omit(Melons$Rdt)
```

La fonction `Subset( )`

La fonction `subset( )` permet d'extraire un sous ensemble en posant une condition sur une variable.

Exemple:

On extrait ici de la base `Melons`, les lignes pour lesquelles la variable `Variete` est égale à Theo.

```
subset( Melons, Variete == "Theo")
```

Remarque

Vous noterez l'utilisation du double signe ==. L'expression `Variete == "Theo"` est en réalité un

vecteur de booléens, dont les valeurs sont `TRUE` ou `FALSE`, selon que la variété est ou non la variété Theo.

Quelques fonctions utiles

Fonction	Description
`help(fonction)ou ?fonction`	permet d'accéder à l'aide.
`c( )`	permet de créer un vecteur
`length( )`	permet d'obtenir la taille d'un vecteur
`matrix(c(……),nrow= ,ncol= ,byrow= )`	permet de définir une matrice à partir d'un vecteur
`dim( )`	permet d'obtenir la taille d'une matrice
`getwd( )`	permet de connaître le répertoire de travail actuellement utilisé par R
`setwd( )`	permet de changer le répertoire de travail
`read.table( )`	permet d'importer un fichier texte
`read.csv2( )`	Permet d'importer un fichier csv
`na.omit( )`	supprime les valeurs manquantes
`View( )`	permet de de visualiser le jeu de données
`edit( )`	sert à ouvrir les données dans un éditeur de texte et à modifier les données
`fix( )`	permet de modifier les données et remplace l'objet (Melons ici) par le jeu de données modifié
`[ , ]`	permet d'accéder à un sous-ensemble: avant la virgule pour les individus et après la virgule pour les variables
`subset( )`	Permet d'extraire un sous-ensemble en posant une condition
`str( )`	Permet d'afficher le type des différentes variables
`mean( )`	Permet de calculer la moyenne
`summary( )`	Affiche les différentes statistiques descriptives des variables

Sources:campus.isped.u-bordeaux2.fr, Pierre-André CORNILLON et autres. (2010). *Statistiques avec R 2ème édition augmentée*. Presse Universitaire de Rennes.