LR1 – Introduction au Logiciel R

L'objectif de cette séance est de vous familiariser avec les outils utilisés lors des travaux dirigés. La première partie concerne le Logiciel R et les manipulations de base effectuées en lignes de commandes à travers l'interface *R Console* (*RGui*). La seconde partie concerne l'Interface de Développement Intégré *RStudio* permettant de simplifier un certain nombre d'opérations classiques en R (gestion des librairies et des fichiers de données et de script R, configuration de la session R, mise à jour du Logiciel R, etc.). La troisième partie concerne la mise en œuvre d'un court script R existant qui va importer une page web et filtrer les informations HTML de cette page afin d'afficher sur une carte des états des Etats-Unis d'Amérique les espérances de vie des populations de différentes origines ethniques.

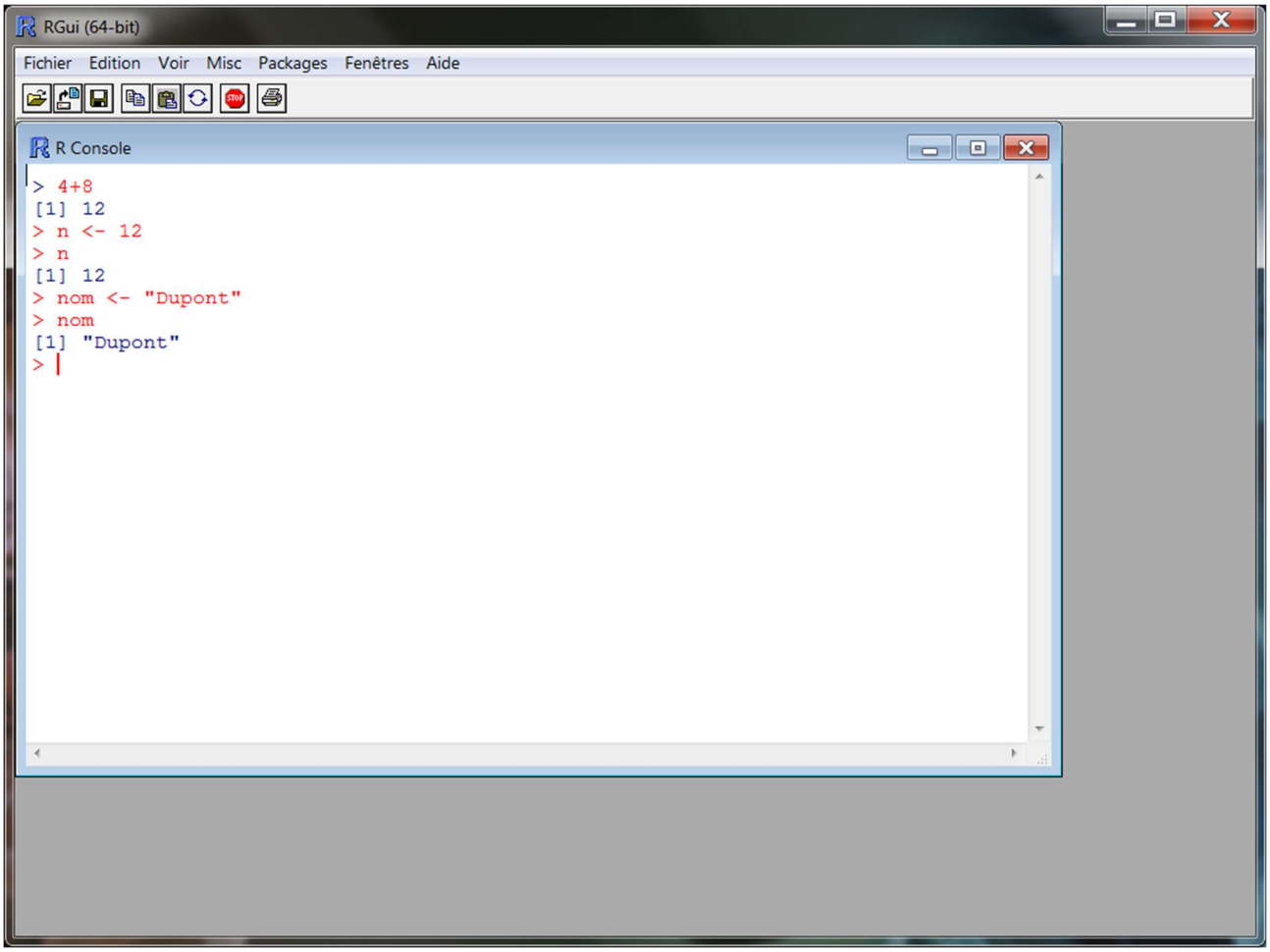
1. Le logiciel R

R est un logiciel de calcul scientifique interactif libre qui possède une large collection d'outils statistiques, d'analyse de données et graphiques.

Le site http://www.r-project.org/ fournit une description exhaustive du langage R ainsi que les liens pour son téléchargement, pour accéder aux différentes bibliothèques de fonctions et pour les documents d'aide. Des versions compilées de R sont disponibles pour Linux, Windows, Mac OS-X et Android.

R est un langage interactif, donc toujours en attente d'une instruction signalée par l'invite « > » dans la fenêtre de commande *R Console* :

Fenêtre de commande R Console



Toutes les opérations peuvent être exécutées en lignes de commandes à l'invite de R. Toutes les instructions seront validées en tapant sur la touche *Entrée* (*Return*) du clavier.

Noter que R est sensible à la casse : les caractères majuscules et minuscules sont différents caractères.

☛ Téléchargez et installez R depuis le site http://www.r-project.org/.

1.1. L'aide en ligne

R dispose d'une aide en ligne très complète et qui vous sera très utile car la plupart des fonctions de R nécessitent plusieurs arguments et options. Il faut donc ne pas hésiter à faire appel à cette aide en ligne afin d'obtenir la description et des exemples d'utilisation des commandes.

- En tapant `? nom_commande` (e.g. `? sqrt`) ou bien `help("nom_commande")` (e.g. `help(sqrt)`), la description de cette commande, ses arguments, le type d'objet qu'elle renvoie ainsi que quelques exemples d'utilisation sont affichés.
- La commande `example(nom_commande)` permet d'afficher des exemples avancés d'utilisation de l'instruction.
- La commande `> help.start()` permet de lancer un navigateur Internet et d'accéder directement à l'aide stockée localement dans l'installation de R : http://127.0.0.1:11490/doc/html/index.html.

1.2. Premières manipulations avec R

☛ Exécutez les instructions de calcul ci-dessous dans la fenêtre R Console.

```
> 4+8
[1] 12
> (6+5*2)/2
[1] 8
> 2^2                          # calcul de puissance
[1] 4
> sqrt(9)                      # fonction racine carrée
[1] 3
```

Note : à l'aide des flèches de déplacement du clavier vous pouvez rappeler les dernières instructions pour les modifier et les réexécuter.

☛ Un objet peut être créé à l'aide de l'opérateur « assigner » qui s'écrit avec le symbole « <- ». Cet objet est stocké en mémoire vive et peut être modifié en lui assignant une autre valeur :

```
> n <- 100
```

Note : en tapant le nom d'un objet le contenu de ce dernier est affiché (quand cet objet ne requiert pas des arguments).

☛ Exécutez les commandes d'affichage des objets en mémoire ci-dessous.

```
> n
[1] 100
> n <- 100/2+5
> n
[1] 55
```

Note : il est possible d'afficher le nom des objets stockés en mémoire vive et/ou leur contenu par la fonction `ls()`.

☛ Exécutez les commandes ci-dessous.

```
> n <- 12
> m <- 8
> s <- 12+8
> nom <- "Dupont"
> ls()
[1] "n" "m" "s" "nom"
> ls.str()
m : num 8                      // num : variable numérique
n : num 12
nom : chr "Dupont"             // chr : variable de type caractère(s)
s : num 20
```

1.3. Les vecteurs

L'objet de base en R est le vecteur. Même lorsque vous créez une variable contenant une unique valeur (ex : `n <- 5`), un vecteur contenant un seul élément est créé.

En R, tous les objets possèdent un mode et une longueur.

Le mode détermine le type de données stockées dans le vecteur. Un vecteur contient un ensemble d'éléments du même type atomique qui peut être : caractère, logique (`T`/`F` ou `TRUE`/`FALSE`), numérique ou complexe.

La longueur détermine le nombre d'éléments dans le vecteur. Elle peut être récupérée à l'aide de la fonction `length()`.

☛ Créez un vecteur `v` à l'aide de la fonction `c()`, qui combine ses arguments pour constituer un vecteur, par la commande suivante :

```
> v <- c(4, 7, 23.5, 76.2, 80)
> v
[1] 4.0 7.0 23.5 76.2 80.0
> length(v)
[1] 5
> mode(v)
[1] "numeric"
```

Note : tous les éléments d'un vecteur doivent être du même type. Si ce n'est pas le cas, R forcera le typage par coercition : le type (appelé mode en R) du vecteur sera celui qui permet de stocker toutes les valeurs.

☛ Exécutez les commandes ci-dessous pour illustrer ce forçage de type :

```
> v <- c(4, 7, 23.5, 76.2, 80, "rrt")
> v
[1] "4" "7" "23.5" "76.2" "80" "rrt"
```

Tous les éléments ont été convertis en caractères (type le plus permissif) qui est le seul qui permette de stocker tous ces éléments (du fait de la présence d'une chaîne de caractères).

Les caractères et chaînes de caractères sont entourées de symboles `"` ou de symboles `'` indifféremment.

Les vecteurs peuvent contenir la valeur particulière `NA` qui représente une valeur manquante.

☛ Exécutez les commandes utilisant la valeur `NA` ci-dessous :

```
> u <- c(4, 6, NA, 2)
> u
[1] 4 6 NA 2
> k <- c(T, F, NA, TRUE)
> k
[1] TRUE FALSE NA TRUE
```

Les principales fonctions de manipulation de vecteurs sont listées dans la table ci-dessous.

Principales fonctions de manipulation de vecteurs

Fonction	Description
`sum(x)`	Somme des éléments de x
`prod(x)`	Produits des éléments de x
`max(x)`	Plus grand des éléments de x
`min(x)`	Plus petit des éléments de x
`range(x)`	Domaine de valeur des éléments de x
`length(x)`	Nombre d'éléments de x
`mean(x)`	Moyenne des éléments de x
`median(x)`	Médiane des éléments de x

var(x)	Variance des éléments de x (calculée avec n-1)
sd(x)	Écart-type de x (standard déviation)
quantile(x)	Bornes définissant les 4 quartiles de x
cov(x)	Matrice de variances-covariances
cor(x)	Matrice des corrélations
cov(x,y)	Covariance entre x et y
cor(x,y)	Corrélation entre x et y
sort(x)	Ordonne les éléments de x
table(x)	Table d'effectifs de x
table(x,y)	Table de contingence

1.4. Data frames

Un *data frame* permet de stocker une matrice de données hétérogènes (i.e. des données de différents types), par exemple un fichier de données, dans R.

Les colonnes d'un data frame, chacune correspondant à un vecteur, peuvent être de modes (types) différents.

☛ Exécutez les instructions ci-dessous afin de créer un data frame `my_dataset` contenant des données de différents types, avec trois attributs nommés `site`, `season` et `pH`, puis l'afficher :

```
> my_dataset <- data.frame(site=c('A','B','A','A','B'), season = c('Winter',
'Summer', 'Summer', 'Spring', 'Fall'), pH = c(7.4,6.3,8.6,7.2,8.9))
> my_dataset
```

Le contenu d'une colonne peut être affiché (ou référencé dans une commande) par l'instruction *nom_data_frame*`$`*nom_colonne*.

Le contenu d'une cellule par l'instruction de sélection *nom_data_frame*`[`*ligne*`, `*colonne*`]`.

☛ Affichez le contenu de la colonne `pH` puis le contenu de la 2ème cellule de la 3ème ligne.

```
> my_dataset$pH
[1] 7.4 6.3 8.6 7.2 8.9
> my_dataset[3,2]
[1] Summer
Levels: Fall Spring Summer Winter
```

Il est possible de faire des sélections simplement en ajoutant des conditions au sélecteur *nom_data_frame*`[`*selecteur_lignes*`, `*nom_colonnes*`]`. Par exemple :

```
> my_dataset[my_dataset$pH > 7, ]
  site season  pH
1    A Winter 7.4
3    A Summer 8.6
4    A Spring 7.2
5    B   Fall 8.9
> my_dataset[my_dataset$site == "A", "pH"]
[1] 7.4 8.6 7.2
> my_dataset[my_dataset$season == "Summer", c("site", "pH")]
  site  pH
2    B 6.3
3    A 8.6
```

☛ Afficher les valeurs des attributs `site` et `pH` pour les lignes dont `season` est différent de `'Fall'`.

☛ Afficher les valeurs de tous les attributs pour les lignes dont `pH` est inférieur ou égal à `8.1`.

2. Environnement de développement intégré RStudio

RStudio est un IDE open source libre pour R. Il est disponible pour les systèmes d'exploitation Windows, Mac OS-X et Linux.

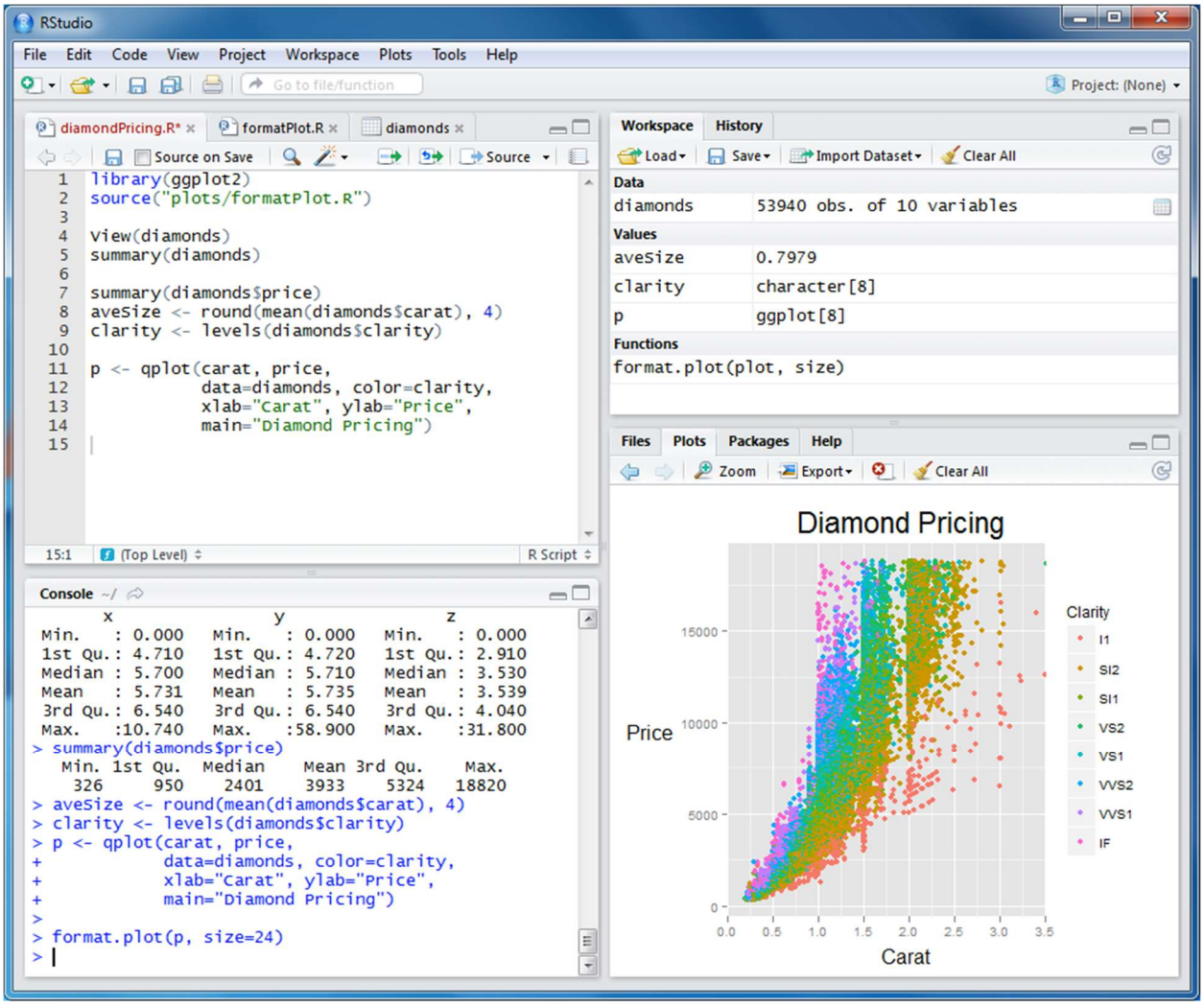
L'interface de RStudio est découpée en 4 zones, comme il est visible dans la capture de la fenêtre de l'interface RStudio ci-après :

- La console (bas-gauche) qui permet la saisie des lignes de commandes R.
- L'afficheur de code (haut-gauche) qui permet l'affichage de codes (scripts, etc.) et résultats textuels (objets, tables, etc.).
- Le gestionnaire d'environnement (haut-droite) qui fournit des informations sur l'environnement (objets en mémoire, historiques de commandes, etc.).
- Le gestionnaire de fichiers, de l'aide et des résultats graphiques (bas-droite) qui permet notamment l'affichage de diagramme, la gestion des librairies et fichiers, et l'affichage des rubriques d'aide.

RStudio est disponible pour les systèmes d'exploitation Windows, Linux et Mac OS-X en deux versions :

- Installer : réalise l'installation parmi les logiciels du système d'exploitation.
- Zip/Tarball : ne nécessite pas d'installation, seulement de :
  - Décompresser le fichier archive `.zip` ou bien `.tar.gz` sur votre machine (cette opération créé la structure de répertoires contenant tous les fichiers nécessaires à RStudio).
  - Lancer l'exécution de RStudio en exécutant le fichier `rstudio.exe` dans le répertoire `/bin/` créé.

Fenêtre d'Interface de RStudio



☛ Téléchargez et installez RStudio Desktop depuis la page : https://www.rstudio.com/products/rstudio/download/.

Note : l'onglet *History* du gestionnaire d'environnement affiche la liste des commandes exécutées qu'il

est possible de copier dans la console sur la ligne de commande, pour les reprendre et les réexécuter.

2.1. Répertoire de travail

Le répertoire courant, ou *working directory*, est le répertoire dans lequel R va lire et écrire les fichiers par défaut.

Afin de définir le répertoire courant , vous pouvez utiliser :

- L'interface RStudio (menu *Session* option *Set working directory*).
- La fonction `setwd("chemin_sur_le_disque")`.

☛ Définissez le répertoire courant comme celui où vous placerez l'ensemble des fichiers utilisés durant la séance.

C'est dans ce répertoire que seront enregistrés les fichiers *.Rhistory* et *.Rdata* contenant respectivement la liste des commandes exécutées et l'environnement (data frames, vecteurs, etc.) de la session.

Pour reprendre la session dans l'état dans lequel vous l'avez quittée, vous pouvez charger ces fichiers au début de la session suivante.

2.2. Librairies R

Il est possible d'étendre les possibilités offertes par R par le biais des librairies disponibles sur le CRAN (The Comprehensive R Archive Network) à l'adresse : https://cran.r-project.org/.

Afin d'installer (télécharger et installer) et charger (activer dans la session R courante) des librairies, vous pouvez utiliser :

- L'interface RStudio (menu *Tools - Install Packages*).
- Les fonctions en lignes de commandes :
  - `install.packages("nom_librairie")` pour télécharger et installer la librairie.
  - `library(nom_librairie)` pour activer cette librairie dans la session actuelle (i.e. pour pouvoir l'utiliser).

Il est possible d'obtenir la liste des librairies installées sur la machine par la commande :

```
> library()
```

et dans un format détaillé (chemin d'installation, numéro de version, etc.) par la commande :

```
> installed.packages()
```

La fonction `RSiteSearch()` permet d'effectuer des recherches dans les listes de diffusion, archives, manuels de R et pages d'aide. Par exemple :

```
> RSiteSearch('association rules')
```

Cette commande nécessite une connexion Internet.

☛ À titre d'exemple, installez la librairie de création de graphiques `ggplot2` et activez-la dans R.

☛ Affichez la liste des librairies installées, dans laquelle `ggplot2` doit désormais apparaître.

3. Exemple d'Application

Cette application consiste à récupérer des données sur une page Internet, structurer les données fournies par la page au format HTML, récupérer les données affichées dans le tableau central et traiter celles-ci afin d'afficher des graphiques comparatifs générés avec R.

Nous allons utiliser les données de la page *Wikipedia* suivante concernant l'espérance de vie aux USA : https://en.wikipedia.org/wiki/List_of_U.S._states_by_life_expectancy.

☛ Téléchargez le fichier exemple de script R sur la page du cours et ouvrez-le dans RStudio (menu *Fichier*).

Ce script (suite d'instructions R) s'affiche alors dans l'onglet qui s'ouvre en haut à gauche dans la fenêtre de RStudio.

☛ Exécutez une à une les commandes en positionnant le curseur sur la première ligne et en tapant successivement le raccourci clavier `CTRL-Entrée` (ou `CTRL-R`) qui exécute la ligne courante et positionne le curseur sur la ligne suivante.

Attention : lorsque vous exécutez une ligne `install.packages()` du script, attendez que cette installation soit terminée avant d'exécuter la suivante (afin d'être sûr d'éviter tout problème, notamment de résolution de dépendances entre les librairies).