

TP : Introduction au logiciel R

1. Introduction

Le logiciel R (disponible sur <http://www.r-project.org/>) est un logiciel de Statistique libre ayant un certain nombre d'atouts :

- il permet l'utilisation des méthodes statistiques classiques à l'aide de fonctions prédéfinies
- il permet de créer ses propres programmes dans un langage de programmation assez simple d'utilisation (proche de Matlab)
- il permet d'utiliser des techniques statistiques innovantes et récentes à l'aide de package développés par les chercheurs et mis à disposition sur le site du CRAN (<http://cran.r-project.org/>).

Le logiciel R fonctionne initialement en ligne de commande, mais des interfaces permettent désormais une utilisation plus conviviale. Nous proposons ici de travailler avec l'interface RStudio, téléchargeable sur : <http://www.rstudio.com/>

1.1 L'interface RStudio

L'interface RStudio est composée de quatre fenêtres :

- Fenêtre d'édition (en haut à gauche) : dans cette fenêtre apparaissent les fichiers contenant les scripts R que l'utilisateur est en train de développer. Enregistrer le fichier avec une extension .r permet une coloration syntaxique adaptée au langage R. En entête de cette fenêtre, des icônes permettent de sauvegarder le fichier, d'exécuter un morceau de code sélectionné (icône run) ou l'intégralité du code contenu dans le fichier (icône source)
- Fenêtre de commande (en bas à gauche) : cette fenêtre contient une console dans laquelle les codes R sont saisis pour être exécutés
- Fenêtre espace de travail / historique (en haut à droite) : contient les objets en mémoire, que l'on peut consulter en cliquant sur leur noms, ainsi que l'historique des commandes exécutées
- Fenêtre explorateur / graphique / package / aide (en bas à droite) : l'explorateur permet de se déplacer dans l'arborescence des répertoires, la fenêtre graphique contient les graphiques tracés via R (il est possible de les exporter), la fenêtre package montre les packages installés et actuellement chargés et la fenêtre d'aide contient la documentation sur les fonctions et packages.

1.2 Le répertoire de travail

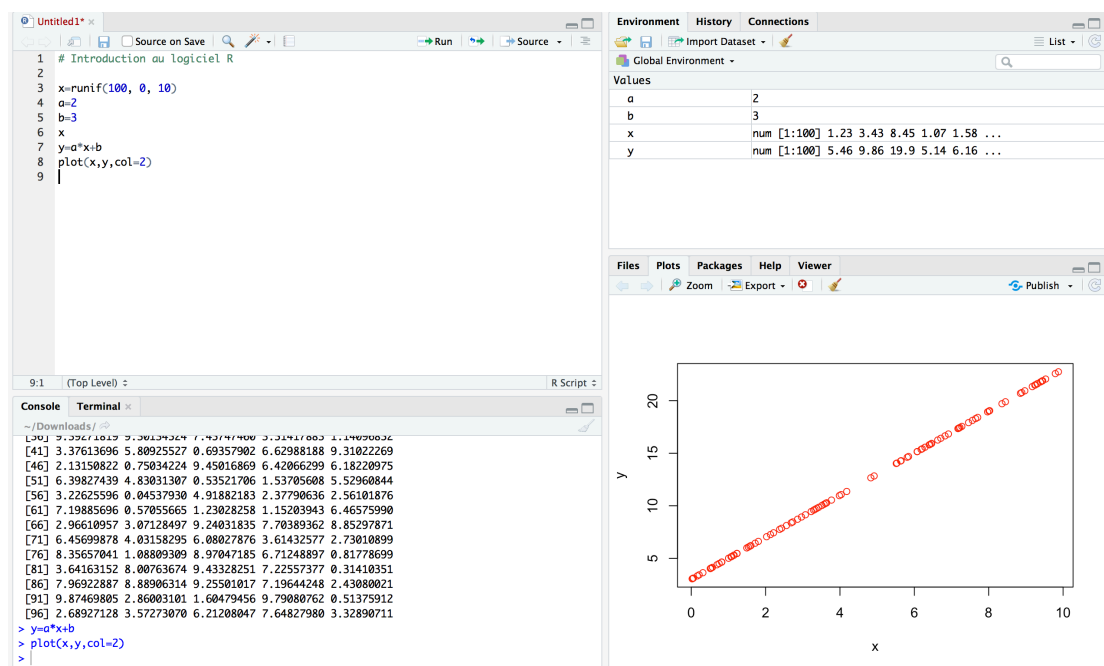
Le répertoire de travail est celui à partir duquel vous avez lancé l'interface RStudio. Il sera pratique de se placer dans un répertoire de travail bien défini, celui par exemple contenant le fichier .r dans lequel vous tapez vos scripts R. Pour ce faire, vous pouvez soit utiliser la commande `setwd` pour vous déplacer dans l'arborescence des répertoires, soit utiliser le menu de l'interface :

- ✓ Session
 - ✓ Set Working Directory
 - ✓ To Source File Location

Par la suite, lorsque vous serez amené à charger des jeux de données, si ceux-ci sont placés dans le répertoire courant dans lequel vous vous êtes placé, vous n'aurez pas à saisir le chemin complet de ce répertoire.

1.3 Les packages

Un grand nombre de fonctions, contenus dans différents packages, sont installés dans la version de base du logiciel R. Il est possible (et nous en aurons besoin dans les différents cours de Statistique) d'installer des packages supplémentaires. Pour cela, lorsque vous disposez d'une connexion internet, il suffit d'utiliser la commande suivante en indiquant le nom du package que l'on veut installer :



```
R> install.packages('ordinalClust')
```

RStudio vous proposera alors de choisir le serveur à utiliser pour télécharger le package et procédera ensuite à l'installation. Il faudra ensuite charger le package à l'aide de la commande `library()` :

```
R> library('ordinalClust')
```

L'installation n'est à réaliser qu'une seule fois, alors que le chargement du package doit être fait au lancement de chaque nouvelle session.

2. Les objets R

Tout dans le langage R est un objet : les variables contenant des données, les fonctions, les opérateurs, même le symbole représentant le nom d'un objet est lui-même un objet. Les objets possèdent au minimum un *mode* et une *longueur* et certains peuvent être dotés d'un ou plusieurs attributs.

Le mode d'un objet est obtenu avec la fonction `mode` :

```
> v <- c(1, 2, 5, 9)
> mode(v)
[1] "numeric"
```

La longueur d'un objet est obtenue avec la fonction *length* :

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

Le mode prescrit ce qu'un objet peut contenir. À ce titre, un objet ne peut avoir qu'un seul mode. À chacun de ces modes correspond une fonction du même nom servant à créer un objet de ce mode.

- ✓ Les objets de mode "numeric", "complex", "logical" et "character" sont des objets simples (atomic en anglais) qui ne peuvent contenir que des données d'un seul type.
- ✓ En revanche, les objets de mode "list" ou "expression" sont des objets récurifs qui peuvent contenir d'autres objets. Par exemple, une liste peut contenir une ou plusieurs autres listes.
- ✓ La fonction `typeof` permet d'obtenir une description plus précise de la représentation interne d'un objet (c'est-à-dire au niveau de la mise en œuvre en C). Le mode et le type d'un objet sont souvent identiques.

3. Premières commandes R

Cf TP1.R

4. Premiers pas sur R

Exercice 1 :

Trouver le nom des fonctions de référence vues précédemment :

- a. $f(x) = x^2$
- b. $g(x) = |x|$
- c. $h(x) = \sqrt{x}$
- d. $i(x) = e^x$
- e. $j(x) = \ln(x)$
- f. $k(x) = \frac{1}{x}$

Exercice 2 :

Effectuer l'opération suivante : $4850 / 26$. Afficher le résultat avec seulement trois décimales. Comment est réalisé l'arrondi?

Exercice 3 :

Dans R, la valeur de la variable π est pré calculé et nommée « pi ».

Afficher cette valeur à l'écran avec une précision de 3 puis de 5 décimales.
Ajouter 12 à la variable pi, puis affecter à cette même variable la valeur 9.
Imprimer successivement les nouvelles valeurs de pi.

Exercice 4 :

Explorer les commandes de création c(), seq(), rep()

Exercice 5 :

Construire les vecteurs suivants :

V1 = (-1,3.2,-2,8)

V2 = (-2,-1,0,1,2,3,4,5,6)

V3 = (0.05,0.1,0.15,0.2)

V4 = (1,1,1,1,1,1,1,1,1,1)

V5 = ('OUI','NON')

Exercice 6 :

Ordonner le vecteur V1.

Exercice 7 :

Quel est le résultat de l'opération $V6 = 2 * V2 - 3$, puis de l'opération $V3 + V2$.

Qu'en concluez-vous sur les opérations arithmétiques utilisant des vecteurs ?

Prendre le logarithme des coordonnées de V3.

Exercice 8 :

Accéder à la deuxième coordonnée de V5.

Créer le vecteur V8 formée des 3 dernières composantes de V6.

Exercice 9 :

Afficher la longueur de V6 ; calculer la somme des composantes de V6 en faisant le calcul puis en utilisant la fonction R correspondante.

Exercice 10 :

Créer le vecteur « vec1 » contenant la suite des entiers de 1 à 12.

Ajouter à la fin de ce vecteur les valeurs, 16, 17, 18.

Exercice 11 :

Créer le vecteur « vec2 » contenant les valeurs 0, 0.5, 1.0, 1.5, 2.0, 2.5, 3.0, 3.5, 4.0, 4.5, 5.0.

Exercice 12 :

Créer le vecteur « vec3 » contenant tous les multiples de 2 compris entre 1 et 50.

Exercice 13 :

Créer un vecteur « vec4 » de taille 10, contenant des valeurs tirées aléatoirement entre 1 et 100.

Exercice 14 :

Créer un vecteur « vec5 » contenant les valeurs

1 2 3 4 5 6 7 8 9 10 1 2 3 4 5 6 7 8 9 10 1 2 3 4 5 6 7 8 9 10

Exercice 15 :

Charger les données fournies dans une variable « data ».

Le tableau donne, pour l'année 1982, le pourcentage des communes qui disposent d'une boulangerie en fonction de la taille de la commune. (Source INSEE-SCEES).

Calculer le pourcentage de communes qui disposent d'une boulangerie sur l'ensemble du territoire.

Exercices BONUS :

Créer la matrice identité « matIdentite » de dimension 10 lignes x 10 colonnes (chiffre 1 sur la diagonale et chiffre 0 ailleurs)

Créer une matrice « matAleatoire » contenant des valeurs tirées aléatoirement entre 0 et 20 de dimension 10 lignes x 10 colonnes.