
Introduction

Dans le milieu des années 1970, John Chambers, chercheur à Bell Lab AT&T (devenu par la suite Lucent Technology) développe le langage `S` pour l'analyse statistique et l'exploitation graphique de données. Depuis, il est devenu un logiciel commercialisé sous le nom de `S-PLUS` par la société Insightful Corporation.

En 1995, le projet `R` est lancé par Robert Gentleman et Ross Ihaka du département de statistique de l'Université d'Auckland (Nouvelle-Zélande), avec une syntaxe proche de celle de `S`. Ce langage est décrit dans un article paru en 1996 par ses auteurs¹. En 1997, `R` est intégré dans le projet GNU et il devient disponible sur plusieurs systèmes d'exploitation (Linux/Unix, Windows, Mac OS). Le développement et la distribution du noyau sont assurés par une équipe, le « R Development Core Team ». Cependant, les utilisateurs peuvent contribuer en proposant de nouvelles fonctions à travers des *packages* qui sont disponibles sur des serveurs.

1.1 Caractéristiques `R`

C'est donc un logiciel libre (open-source), indépendant et gratuit (distribué sous la licence « GNU Public Licence »). C'est un langage interprété et orienté-objet. Les fonctions disponibles se trouvent dans une bibliothèque, organisée en *packages* contenant les fonctions, opérateurs, jeux de données, etc. Toutes les variables, données, fonctions, etc. intervenant dans une session `R` sont stockés sous forme d'objets en mémoire dans un espace de travail (appelé *workspace*). Il est basé sur les langages C, C++, Fortran et Java.

Ce logiciel connaît un développement considérable depuis sa création. Un article lui a été consacré dans le *New York Times*². D'après cet article, de nombreuses

1. R. Gentleman & R. Ihaka. R: a language for data analysis and graphics. *Journal of Computational and Graphical Statistics*, 5:3, 299–314, 1996.

2. Data Analysts Captivated by R's Power. *New York Times*, 6 janvier 2009.

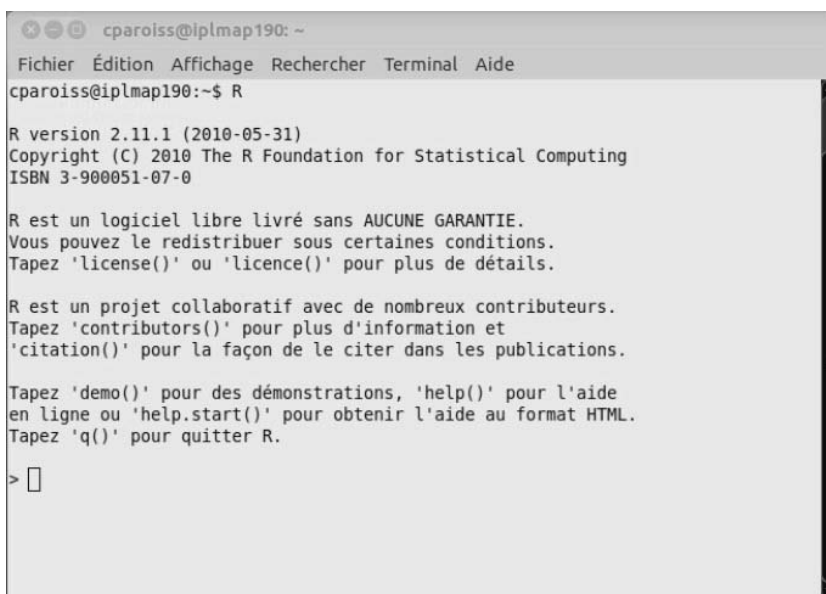
entreprises internationales utilisent ce logiciel : Google, Bank of America, InterContinental Hotels Group, Shell, Pfizer et Merck (pour ces deux derniers, des groupes pharmaceutiques, cela ne concerne pas les essais cliniques qui doivent répondre à des normes qui ne sont pas garanties avec R).

1.2 Installation de R

L'installation sous Linux est simple : les paquets correspondant à R sont déjà présents dans la plupart des distributions de Linux et parfois R est directement installé. Sous Windows, il suffit de télécharger l'archive pour l'installation sur le site internet du projet (voir à la fin de ce chapitre) et de lancer l'installation.

1.3 Démarrer et quitter une session R

Avant toute chose, il faut commencer une session R. Sous Linux, il suffit de lancer la fonction R dans un terminal. On obtient le résultat suivant à l'écran représenté sur la figure 1.1. Sous Windows, il suffit de lancer le programme Rgui qui est une inter-

A screenshot of a Linux terminal window. The window title is 'cparoiss@iplmap190: ~'. The terminal shows the command 'R' being executed, which outputs the R version information and usage instructions. The output text is as follows:

```
Fichier  Édition  Affichage  Rechercher  Terminal  Aide
cparoiss@iplmap190:~$ R

R version 2.11.1 (2010-05-31)
Copyright (C) 2010 The R Foundation for Statistical Computing
ISBN 3-900051-07-0

R est un logiciel libre livré sans AUCUNE GARANTIE.
Vous pouvez le redistribuer sous certaines conditions.
Tapez 'license()' ou 'licence()' pour plus de détails.

R est un projet collaboratif avec de nombreux contributeurs.
Tapez 'contributors()' pour plus d'information et
'citation()' pour la façon de le citer dans les publications.

Tapez 'demo()' pour des démonstrations, 'help()' pour l'aide
en ligne ou 'help.start()' pour obtenir l'aide au format HTML.
Tapez 'q()' pour quitter R.

> 
```

FIGURE 1.1 – Capture d'écran de R sous Linux

face graphique assez simple. Une capture d'écran est représentée sur la figure 1.2. Dans les deux cas, on saisit alors les fonctions dans le terminal. On quitte une session en utilisant la fonction `q` sans argument (il s'agit bien d'une fonction, il ne faut

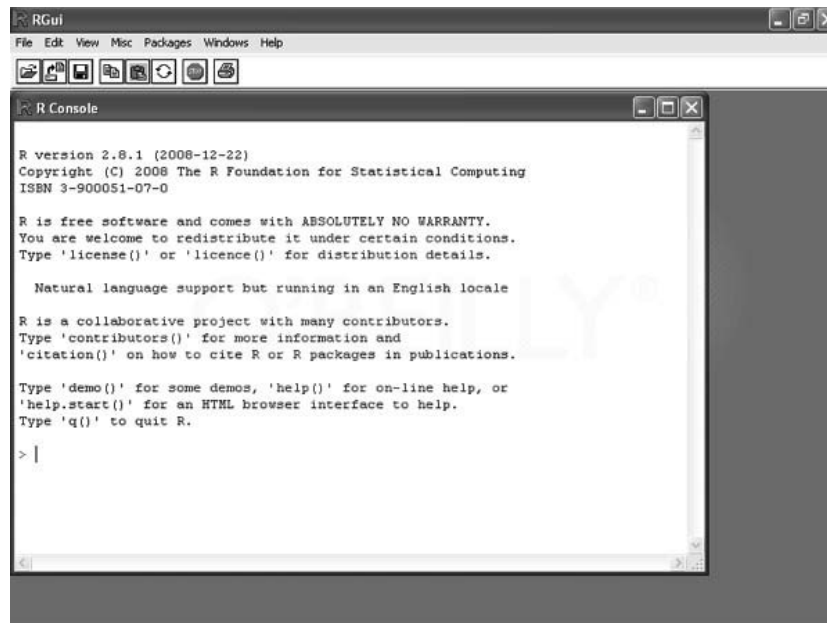


FIGURE 1.2 – Capture d'écran de R sous Windows (logiciel Rgui)

pas oublier les parenthèses bien que l'appel à cette fonction ne nécessite pas obligatoirement d'arguments en entrée - autrement R renvoie les instructions composant cette fonction).

```
q()
```

Sous Windows, on peut également utiliser le menu pour quitter R. Dans les deux cas, il est alors demandé si on souhaite enregistrer l'espace de travail. En fait, lorsqu'on quitte R, il propose d'appliquer les fonctions `save.history()` et `save.image()`. La première fonction permet d'avoir la liste des fonctions saisies (fichier, nommé `.Rhistory` par défaut, au format texte) et la seconde de sauvegarder les variables (fichier, nommé `.RData` par défaut, à un format propre à R). Ces deux fonctions peuvent être utilisées à tout moment. Quand on ouvre une session, R charge automatiquement les données si un fichier `RData` est présent dans le répertoire courant. Si, de plus, un fichier `Rhistory` est présent dans le répertoire courant, l'historique est aussi chargé (mais ce n'est pas le cas s'il n'y a pas de fichier `RData`). À tout moment, on peut charger l'historique en utilisant la fonction `loadhistory`. Enfin, il est possible d'arrêter un script ou une fonction en appuyant sur « ctrl-c »

1.4 Historique des fonctions et aide

Lors d'une session, les fonctions sont stockées en mémoire. On peut rechercher les fonctions déjà exécutées (lors de la session active, à moins que la dernière session de travail n'ait été sauvée) à l'aide des flèches haut et bas du clavier. De plus, on peut accéder à tout l'historique d'une session en utilisant la fonction `history()`. Bien que ces fonctionnalités soient pratiques, il est recommandé de noter les fonctions dans un fichier texte et de les exécuter pas à pas. Sous Windows, `Rgui` propose un éditeur de texte. Sous Linux, on choisira un éditeur de texte, comme `emacs` ou `vi`, certains d'entre eux coloriant le code pour repérer les éléments d'un script R.

Il est également très important et très utile de savoir consulter l'aide de R et de savoir rechercher dans l'aide. Commençons par le premier point : rechercher de l'aide sur une fonction dont on connaît l'existence. On y accède en invoquant la fonction `help` ou bien `?` suivi du nom de la fonction dont on souhaite consulter l'aide. Voici, par exemple, comment obtenir de l'aide sur la fonction `mean` (qui calcule la moyenne empirique d'un vecteur). L'exemple ci-après illustre la première manière de procéder.

```
help(mean)
```

Dans le code ci-après, on utilise la seconde manière (attention à ne pas insérer un espace après le point d'interrogation).

```
?mean
```

Sous Windows, une fenêtre s'ouvre et, sous Linux, l'aide s'affiche dans le terminal de la session R (il faut taper `q` pour sortir de l'aide). En général, chaque aide contient au moins un exemple. Il est possible d'exécuter ces exemples en utilisant la fonction `example`.

```
example(mean)
```

Passons au second point. Pour effectuer une recherche dans l'aide, il suffit d'utiliser la fonction `help.search()` avec en argument le mot-clef entre guillemets, ou bien en utilisant le symbole `??` devant le mot-clef. L'exemple ci-après illustre la première manière de procéder.

```
help.search(hessian)
```

Dans le code ci-après, on utilise la seconde manière (attention à ne pas insérer un espace après les deux points d'interrogation).

```
??hessian
```

On peut également parcourir l'aide, sous Windows ou sous Linux, en utilisant la fonction `help.start()` qui ouvre le navigateur internet (pas besoin de connexion à internet).

1.5 Chargement et installation de packages

De nombreuses fonctionnalités sont proposées dans la version de base de R. Mais celle-ci est enrichie par des contributions personnelles d'utilisateur qui sont rendues disponibles sur internet : les packages. La plupart des packages sont déposés sur un serveur, le CRAN (Comprehensive R Archive Network). De plus, comme le nombre de packages croît sans cesse, il existe des pages sur le CRAN recensant les différents packages pour certains domaines, les Task Views.

Un package R est une archive compressée contenant des fonctions, des données permettant d'illustrer les fonctions, ainsi que des aides sur ces fonctions et données. Cette archive est au format `.zip` sous Windows et au format `.tar.gz` sous Linux. L'installation d'un package est très facile tant sous Windows que sous Linux.

La fonction `installed.packages()` permet d'obtenir la liste des packages déjà installés, ainsi que leur version. Si le package qu'on souhaite utiliser est déjà installé, il faut le charger à l'aide de la fonction `library()`.

```
library(MASS)
```

On peut également utiliser la fonction `require` qui est généralement utilisée à l'intérieur de fonctions. La différence entre ces deux fonctions réside dans le message renvoyé si le package n'est pas installé : la fonction `library` renvoie un message d'erreur alors que la fonction `require` renvoie un avertissement.

On peut obtenir de l'aide sur un package en utilisant la fonction `help`.

```
help(package="MASS")
```

Si un package n'est pas installé, il faut utiliser la fonction `install.packages()`. Il y a deux possibilités : soit le package (comme archive) est présent physiquement sur le disque dur, soit on le récupère par internet sur le CRAN (dans ce cas, il faut commencer par choisir un site miroir du CRAN où R ira télécharger le package à installer).

```
install.packages("pracma")
```

Sous Windows, on peut utiliser le menu de Rgui. Si le package est présent sur le disque dur, il faut aller dans le menu `Packages` puis choisir `Installer le(s) package(s) depuis des fichiers zip`. Si le package est sur le serveur CRAN, alors il faut d'abord choisir un site miroir et sélectionner le package. Une fois installé, il faut le charger dans R : dans le même menu que précédemment, il faut choisir `Charger le package`.

En utilisant la fonction `old.packages()`, il est possible de vérifier si les packages installés sont des versions anciennes ou pas. On peut aussi les mettre à jour en appliquant la fonction `update.packages()`. Par défaut, cette fonction demande, pour chaque package dans une vieille version, si l'utilisateur souhaite mettre à jour celui-ci. On peut accepter systématiquement les mises-à-jour avec l'option `ask`.

1.6 Environnements graphiques pour R

Il existe des environnements graphiques offrant une utilisation plus conviviale de R. On se rapproche alors de logiciels presse-boutons qui facilitent l'utilisation de R par des non-spécialistes. Dans ce livre, nous n'avons pas mis l'accent sur ces environnements pour privilégier l'apprentissage du langage. Néanmoins, c'est un point qui mérite d'être abordé. Ces environnements sont soit des packages, soit des logiciels. Nous allons voir un exemple pour chacun des deux cas : le package `r-commander` et le logiciel `RStudio`.

1.6.1 Le package `r-commander`

Le package `r-commander` propose une interface graphique avancée pour une utilisation simplifiée de R. Ce package est disponible sous Windows et sous Linux. Une fois installé, on obtient la fenêtre graphique représentée sur la figure 1.3. On y retrouve plus ou moins les mêmes menus que dans `Rgui`, ainsi que d'autres dédiés à l'analyse statistique des données : un menu `Données` pour l'importation et l'exportation de données, un menu `Statistiques` pour l'analyse descriptive des données, un menu `Graphes` dédié à la représentation graphique des données, un menu `Modèles` pour l'application de modèles (comme, par exemple, la régression linéaire) et enfin un menu `Distributions` sur l'ajustement de lois.

1.6.2 Le logiciel `RStudio`

`RStudio` est un logiciel présenté sur la figure 1.4. L'aspect visuel n'est pas sans rappeler le logiciel `Matlab`. En effet, la fenêtre est divisée en trois parties. Sur la sous-fenêtre de gauche, on retrouve la console où l'on saisit les instructions. Sur la sous-fenêtre en haut à droite, l'onglet `Workspace` concerne l'espace de travail (on retrouve, en particulier, les objets créés) et l'onglet `History` contient la liste des instructions exécutées. Enfin, sur la sous-fenêtre en bas à droite, on trouve quatre onglets. Le premier onglet permet d'avoir la liste des fichiers du répertoire courant, le second onglet contient les éventuels graphiques réalisés, le troisième onglet est dédié à la gestion des packages et enfin le dernier concerne l'aide sur R.

1.7 Exécution en mode batch

On peut lancer un script en mode batch, c'est-à-dire exécuter un script directement depuis une ligne de commande. Cela peut être utile, par exemple, quand on souhaite exécuter un programme R sur un serveur distant, comme un cluster (qui peut avoir de meilleures capacités de calcul). Pour cela, il faut ouvrir, sur le serveur distant, un terminal (pour un serveur sous Linux) ou une invite de fonctions (pour un serveur sous Windows), puis de lancer la fonction.

```
R CMD BATCH scriptfile outputfile
```

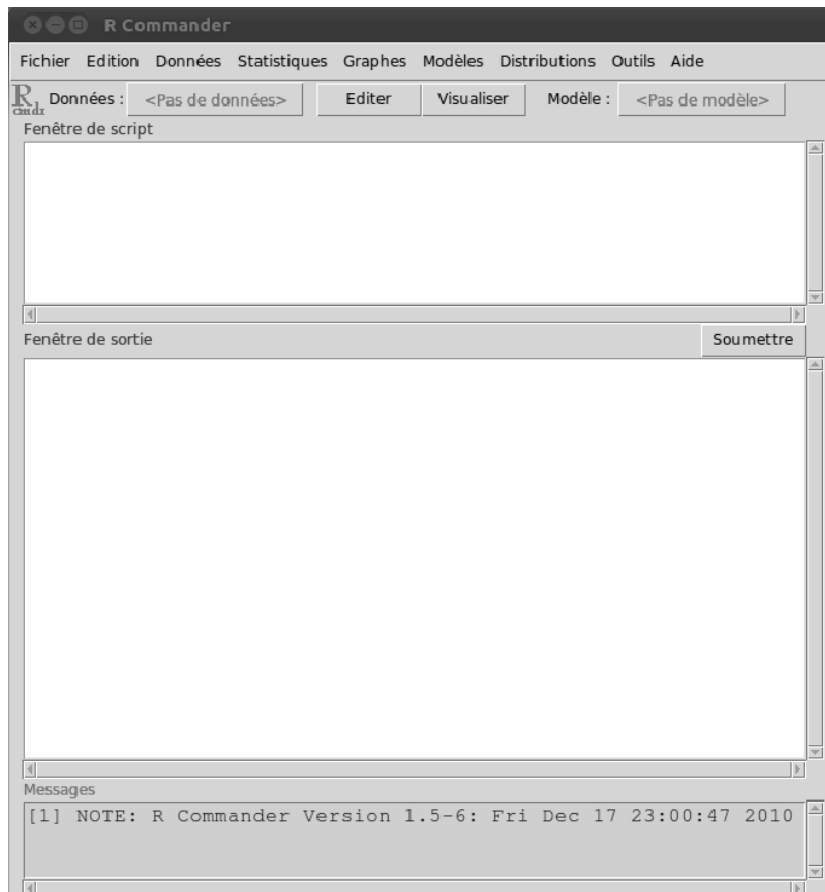


FIGURE 1.3 – Capture d'écran de Rcommander

Cette fonction exécute le fichier `scriptfile` et renvoie (éventuellement) le résultat dans le fichier `outputfile`. Il est possible de passer des arguments au script exécuté (dans l'exemple ci-après, il y en a deux).

```
R CMD BATCH scriptfile arg1 arg2
```

Il faut alors récupérer ces arguments dans le script `scripfile` en utilisant la fonction `commandArgs` et à insérer de la manière suivante.

```
argv <- commandArgs(TRUE)
```

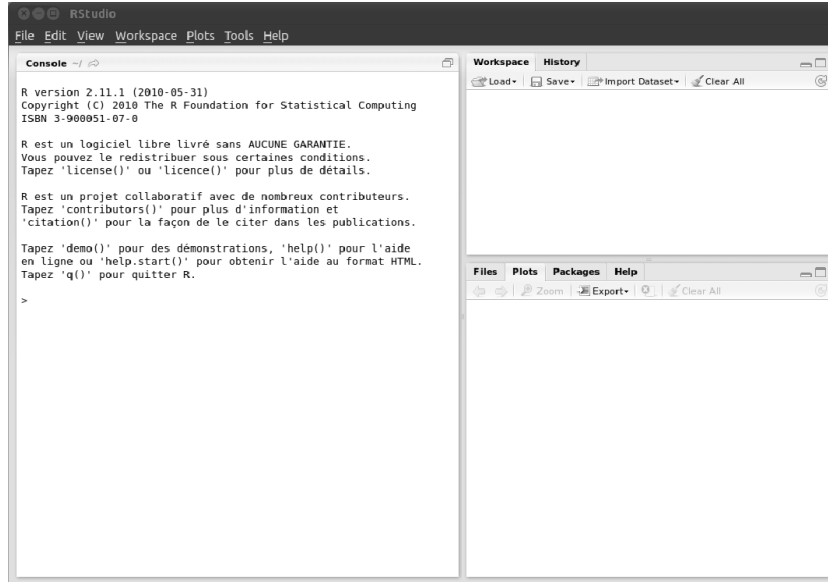


FIGURE 1.4 – Capture d'écran de RStudio

1.8 Sites internet et référence officiels de R

Pour conclure ce premier chapitre, on donne ci-après une liste de quelques sites internet utiles à l'installation de R ou de packages.

Projet R	http://www.r-project.org
CRAN	http://cran.r-project.org
Task Views	http://cran.r-project.org/web/views/

Enfin, on peut être amené à citer R dans un rapport ou dans un article. On peut obtenir la manière de le citer en utilisant la fonction `citation()`.