

**Frédéric Bertrand • Myriam Maumy-Bertrand**

# **Initiation à La statistique avec R**

**4<sup>e</sup> ÉDITION**

**DUNOD**

## Des mêmes auteurs :

*Mathématiques pour les sciences de l'ingénieur, Tout le cours en fiches – 3<sup>e</sup> édition*, Dunod, 2022  
*PASS Biostatistiques, Mathématiques, Probabilités – 4<sup>e</sup> édition*, Dunod, 2021  
*Mathématiques Licence 1 – Exercices et méthodes*, Dunod, 2020  
*Modélisation statistique par la pratique avec R, Cours et exercices corrigés*, Dunod, 2019  
*Mathématiques Licence 2 – Exercices et méthodes*, Dunod, 2018  
*Mathématiques : concours des catégories A et B*, Dunod, 2012  
*Maxi fiches de Statistique. En 80 fiches*, Dunod, 2011  
*Probabilités, Statistiques et Économétrie : concours des catégories A et B*, Dunod, 2011  
*Mathématiques L1/L2 : Statistique et Probabilités en 30 fiches*, Dunod, 2009  
*Mathématiques L1/L2 : Analyse en 30 fiches*, Dunod, 2009  
*Mathématiques L1/L2 : Algèbre/Géométrie en 30 fiches*, Dunod, 2009

Illustration de couverture : © Diana913 – Shutterstock.com

<p>Le pictogramme qui figure ci-contre mérite une explication. Son objet est d'alerter le lecteur sur la menace que représente pour l'avenir de l'écrit, particulièrement dans le domaine de l'édition technique et universitaire, le développement massif du photocopillage.</p> <p>Le Code de la propriété intellectuelle du 1<sup>er</sup> juillet 1992 interdit en effet expressément la photocopie à usage collectif sans autorisation des ayants droit. Or, cette pratique s'est généralisée dans les établissements</p>		<p>d'enseignement supérieur, provoquant une baisse brutale des achats de livres et de revues, au point que la possibilité même pour les auteurs de créer des œuvres nouvelles et de les faire éditer correctement est aujourd'hui menacée.</p> <p>Nous rappelons donc que toute reproduction, partielle ou totale, de la présente publication est interdite sans autorisation de l'auteur, de son éditeur ou du Centre français d'exploitation du droit de copie (CFC, 20, rue des Grands-Augustins, 75006 Paris).</p>
--	--	--

© Dunod, 2010, 2014, 2018, 2023

11 rue Paul Bert, Malakoff  
www.dunod.com

ISBN 978-2-10-084794-5

Le Code de la propriété intellectuelle n'autorisant, aux termes de l'article L. 122-5, 2° et 3° a), d'une part, que les « copies ou reproductions strictement réservées à l'usage privé du copiste et non destinées à une utilisation collective » et, d'autre part, que les analyses et les courtes citations dans un but d'exemple et d'illustration, « toute représentation ou reproduction intégrale ou partielle faite sans le consentement de l'auteur ou de ses ayants droit ou ayants cause est illicite » (art. L. 122-4).

Cette représentation ou reproduction, par quelque procédé que ce soit, constituerait donc une contrefaçon sanctionnée par les articles L. 335-2 et suivants du Code de la propriété intellectuelle.

# Table des matières

<b>COMMENT UTILISER CE LIVRE ?</b>	VIII
<b>AVANT-PROPOS</b>	IX
<b>CONVENTIONS TYPOGRAPHIQUES</b>	X
<b>CHAPITRE 1 • INITIATION AU LANGAGE R</b> .....	1
1.1 Présentation du langage R .....	1
1.2 Maintenant c'est à vous .....	4
1.3 Vos premières lignes de commande .....	8
1.4 Quelques remarques sur la fenêtre R Console .....	14
1.5 Pour en savoir plus sur le langage R .....	14
1.6 R sans les mains .....	15
<b>EN RÉSUMÉ</b> .....	16
<b>EXERCICES</b> .....	16
<b>SOLUTION DES EXERCICES</b> .....	18
<b>CHAPITRE 2 • OBJETS, FONCTIONS ET DONNÉES AVEC LE LANGAGE R</b> .....	21
2.1 Le fichier de données iris .....	21
2.2 Quelques définitions .....	23
2.3 Quelques manipulations élémentaires .....	24
2.4 Les objets sous R .....	26
2.5 Les fonctions de R .....	48
2.6 Les données sous R .....	51

EN RÉSUMÉ .....	72
EXERCICES .....	74
PROBLÈME .....	76
SOLUTION DES EXERCICES .....	79
SOLUTION DU PROBLÈME .....	89
<b>CHAPITRE 3 • STATISTIQUES DESCRIPTIVES UNIVARIÉES .....</b>	<b>91</b>
3.1 Introduction .....	91
3.2 Vocabulaire de la statistique .....	92
3.3 Les fichiers de données : Mesures et Mesures5 .....	95
3.4 Description d'une série statistique .....	100
3.5 Représentations graphiques d'une série statistique .....	107
3.6 Représentations numériques d'une série statistique .....	141
EN RÉSUMÉ .....	156
EXERCICES .....	157
PROBLÈME .....	161
SOLUTION DES EXERCICES .....	164
SOLUTION DU PROBLÈME .....	178
<b>CHAPITRE 4 • LOIS DE PROBABILITÉS USUELLES .....</b>	<b>189</b>
4.1 Variable aléatoire - Variable aléatoire discrète - Variable aléatoire continue	190
4.2 Variables aléatoires discrètes : cas fini .....	191
4.3 Variables aléatoires discrètes : cas infini .....	195
4.4 Variables aléatoires continues .....	197
4.5 Approximations .....	205
EN RÉSUMÉ .....	206
EXERCICES .....	207
SOLUTION DES EXERCICES .....	211

CHAPITRE 5 • <b>STATISTIQUES DESCRIPTIVES BIVARIÉES</b> .....	223
5.1 Présentation des données .....	223
5.2 Distributions conjointe, marginale et conditionnelle .....	224
5.3 Représentations graphiques .....	231
5.4 Indépendance statistique .....	232
5.5 Liaison entre deux variables .....	232
5.6 Et si vous avez plus que deux variables .....	236
EN RÉSUMÉ .....	248
EXERCICES .....	249
SOLUTION DES EXERCICES .....	251
CHAPITRE 6 • <b>ÉCHANTILLONNAGE. ESTIMATION D'UN PARAMÈTRE. ESTIMATION PAR INTERVALLE DE CONFIANCE</b> .....	256
6.1 Échantillonnage .....	257
6.2 Estimation d'un paramètre .....	258
6.3 Estimation par intervalle de confiance .....	264
EN RÉSUMÉ .....	279
EXERCICES .....	279
PROBLÈME .....	281
SOLUTION DES EXERCICES .....	284
SOLUTION DU PROBLÈME .....	288
CHAPITRE 7 • <b>TESTS PARAMÉTRIQUES ET NON PARAMÉTRIQUES</b> .....	293
7.1 Introduction à la notion de tests .....	293
7.2 Tests de comparaison avec une norme .....	298
7.3 Test de comparaison entre deux populations indépendantes .....	311
7.4 Test de comparaison entre deux populations non indépendantes .....	321
EN RÉSUMÉ .....	322
EXERCICES .....	323

PROBLÈMES .....	325
SOLUTION DES EXERCICES .....	330
SOLUTION DES PROBLÈMES .....	335
<b>CHAPITRE 8 • TESTS DU KHI-DEUX</b> .....	<b>342</b>
8.1 Test du $\chi^2$ d'indépendance .....	342
8.2 Test du $\chi^2$ d'adéquation d'une loi à une loi donnée .....	345
8.3 Test exact de Fisher .....	346
8.4 Test de Mac-Nemar .....	349
EN RÉSUMÉ .....	351
EXERCICES .....	351
SOLUTION DES EXERCICES .....	357
<b>CHAPITRE 9 • RÉGRESSION LINÉAIRE SIMPLE</b> .....	<b>365</b>
9.1 Ajustement linéaire en statistique descriptive .....	365
9.2 Estimation ponctuelle des paramètres de la droite de régression .....	369
9.3 Intervalle de confiance pour les paramètres $\beta_1$ et $\beta_0$ .....	371
9.4 Intervalle de confiance de la valeur moyenne .....	372
9.5 Intervalle de prévision d'une valeur individuelle .....	373
9.6 Comparaison des paramètres de la droite à des valeurs de référence ....	373
EN RÉSUMÉ .....	375
EXERCICES .....	376
SOLUTION DES EXERCICES .....	378
<b>CHAPITRE 10 • ANALYSE DE LA VARIANCE À UN FACTEUR</b> .....	<b>400</b>
10.1 Introduction .....	400
10.2 Modélisation statistique .....	402
10.3 Test de comparaison de plusieurs espérances .....	405
10.4 Vérification des trois conditions fondamentales et conclusion .....	410

10.5 Comparaisons multiples .....	417
10.6 Puissance des tests de l'analyse de la variance à un facteur .....	420
EN RÉSUMÉ .....	423
EXERCICES .....	424
SOLUTION DES EXERCICES .....	426
<b>INDEX</b>	<b>441</b>

# Comment utiliser ce livre ?

## Les rubriques



Les points clé à retenir sont regroupés en fin de chapitre



Des remarques pour aller plus loin



Les pièges et difficultés sont signalés



Des conseils méthodologiques sont donnés tout au long de l'apprentissage



Pour aller plus loin et mettre toutes les chances de votre côté, des ressources complémentaires sont disponibles sur le site [www.dunod.com](http://www.dunod.com).

Connectez-vous à la page de l'ouvrage (grâce aux menus déroulants, ou en saisissant le titre, l'auteur ou l'ISBN dans le champ de recherche de la page d'accueil).  
Sur la page de l'ouvrage, sous la couverture, cliquez sur le lien « LES + EN LIGNE ».



Pour vous, Anaëlle et David, nos deux modèles !

## Avant-propos

Chers lecteurs,

Vous tenez entre vos mains le fruit de nos expériences.

Au cours de nos activités d'enseignement, de recherche ou de consultance, nous avons acquis une connaissance et une maîtrise avancées du langage R. Suite à des demandes de plus en plus fréquentes et nombreuses, nous avons partagé celles-ci depuis plusieurs années avec des publics variés : étudiants en Licence ou en Master de biologie, de mathématique ou de statistique, élèves dans chacune des trois années d'écoles d'ingénieurs, utilisateurs de la statistique lors de stages donnés dans le cadre d'écoles doctorales et de la formation continue de chercheurs, enseignants-chercheurs ou de statisticiens.

Ce livre est articulé autour des notions de base des programmes de Licence et Master. Il possède l'avantage d'offrir simultanément une formation progressive à la statistique et au langage R en réduisant au strict nécessaire les notions théoriques abordées. Ce n'est pas un livre-audio mais presque !

Certains d'entre vous se font peut-être une montagne de la statistique mais il suffit d'être bien chaussé, chaudement vêtu et bien guidé pour atteindre le sommet et pouvoir profiter de la vue.

Les chapitres 1 et 2 vous permettent de vous chauffer : installation et prise en main du langage R. Puis vous vous équipez dans les chapitres 3, 4 et 5 en étudiant les statistiques descriptives pour une ou deux variables et les lois de probabilité usuelles. L'escalade commence à partir du chapitre 6 où vous êtes initiés à la théorie de l'échantillonnage et à une notion trop mal connue : les intervalles de confiance. Le chapitre 7 introduit la théorie des tests d'hypothèse qui est complétée au chapitre 8 par les tests du Khi-deux. Le sommet est atteint avec l'exposé de la régression linéaire simple au chapitre 9 et celui du modèle d'analyse de la variance à un facteur au chapitre 10.

Les données utilisées pour la plupart des exercices ont été recueillies par les auteurs avec l'aide du jardinier. Ce sont donc des données réelles sur lesquelles vous découvrirez la statistique et vous exercerez.

Toutes vos remarques, vos commentaires, vos critiques, et mêmes vos encouragements seront accueillis avec plaisir aux adresses électroniques suivantes :  
frederic.bertrand@utt.fr et myriam.maumy@utt.fr

Bonne lecture (ascension) à tous !

Frédéric Bertrand  
Myriam Maumy-Bertrand

# Conventions typographiques

Cet ouvrage suit un certain nombre de conventions typographiques qui ont pour objectif de faciliter la lecture.

Les noms de fichiers sont imprimés avec une police de type machine à écrire.

**Exemple :** R, SAS, Linux, ade4, etc.

Les noms de fichiers sont également imprimés avec la même police que celle ci-dessus.

**Exemple :** test.R, donnees.txt, etc.

Il en va de même pour les fonctions de R qui apparaissent dans le corps du texte.

**Exemple :** summary, mean, <-, etc.

Nous présentons des commandes saisies sous R et leurs résultats de la manière suivante. La commande saisie est indiquée, dans un encadré, avec une police à chasse fixe précédée de l'invite de commande >.

**Exemple :**

```
> summary(donnees)
```

Lorsque la commande R doit être répartie sur plusieurs lignes, les lignes suivantes sont précédées du symbole + :

**Exemple :**

```
> filled.contour(volcano, color.palette = terrain.colors, asp = 1,  
+ main = "jeu de données volcano : courbes de niveaux")
```

Le résultat de la commande tel qu'il est affiché par R est indiqué avec une police à chasse fixe dans un cadre avec une trame en gris clair.

**Exemple :**

```
Min. 1st Qu. Median Mean 3rd Qu. Max.
```

Version utilisée : nous avons utilisé dans ce livre le langage R version 4.2.1 (2022-06-23).

## Chapitre 1

---

# Initiation au langage R

« Comme dans toute initiation, c'est dans le fait même de sourire qu'est le triomphe. »

De Paul Auster, *La chambre dérobée*, 1994

### Objectifs de ce chapitre

- présenter R
- installer R
- prise en main des premières lignes de commande incontournables



Les informations sur R sont disponibles sur le site internet<sup>1</sup> dédié au projet :  
<https://www.r-project.org/>

## 1.1 PRÉSENTATION DU LANGAGE R

### 1.1.1 Qu'est-ce-que le langage R ?

- Le langage R est un langage de programmation et un environnement mathématique utilisés pour le traitement de données. Il permet de faire des analyses statistiques aussi bien simples que complexes comme des modèles linéaires ou non-linéaires, des tests d'hypothèse, de la modélisation de séries chronologiques, de la classification, etc. Il dispose également de nombreuses fonctions graphiques très utiles et de qualité professionnelle.

---

1. En juillet 2018, il s'agit de l'adresse officielle qui peut depuis avoir été éventuellement modifiée.

- R un langage de programmation complet. Cette caractéristique rend par conséquent le langage R différent des autres logiciels de statistique.
- R est disponible pour Microsoft Windows, Macintosh et de nombreux systèmes de type Unix.
- Actuellement, quatre à cinq nouvelles versions de R apparaissent par an.
- R est distribué gratuitement sous les termes de la « GNU », *General Public Licence Version 2*, Juin 1991.
- R est écrit en C, C++, FORTRAN et Java. De plus, R est plus orienté programmation objet que la plupart des autres logiciels ou langage de programmation statistique.
- R est un clone gratuit du langage S-Plus, actuellement commercialisé par Tibco Software Inc., créé autour du langage S qui a été développé par John Chambers des laboratoires Bell.
- R a été créé en 1993 par Ross Ihaka et Robert Gentleman à l'Université d'Auckland, Nouvelle Zélande, et est maintenant développé par la R Development Core Team. L'origine du nom du langage provient, d'une part, des initiales des prénoms des deux auteurs (Ross Ihaka et Robert Gentleman) et, d'autre part, d'un jeu de mots sur le nom du langage S auquel il est apparenté.

### 1.1.2 Comment se procurer le langage R ?

- L'adresse <https://www.r-project.org/> est le premier résultat pour la recherche de la lettre R avec le moteur de recherche Google™ et la meilleure source d'informations sur le langage R. Vous y trouverez les différentes distributions du langage, de nombreuses bibliothèques de fonctions et des documents d'aide.
- Le langage R est gratuit et se télécharge directement depuis l'internet. Il évolue très rapidement et à peu près tous les six mois une nouvelle version du langage est proposée au public. Elle est accessible via la page officielle consacrée au projet <https://www.r-project.org/>.
- Pour faire face au très grand nombre de téléchargements du langage, un système de miroirs, le Comprehensive R Archive Network <https://cran.r-project.org/>, a été mis en place. Les mêmes fichiers sont ainsi disponibles simultanément sur différents serveurs situés à plusieurs endroits dans le monde<sup>2</sup>. Cette organisation présente au moins deux avantages majeurs pour vous : pouvoir choisir un miroir proche de chez vous où que vous soyez dans le monde ou un miroir de secours lorsque le miroir que vous avez l'habitude d'utiliser est indisponible.

---

2. La liste des serveurs est disponible à l'adresse <https://cran.r-project.org/mirrors.html> et leur état à l'adresse [https://cran.r-project.org/mirmon\\_report.html](https://cran.r-project.org/mirmon_report.html).



En août 2022, il existait quatre miroirs en France dont les adresses sont les suivantes :

1. Laboratoire de Biométrie et Biologie Évolutive, CNRS et Université Lyon I  
<https://pbil.univ-lyon1.fr/CRAN/>
2. Institut de Biologie et Chimie des Protéines, CNRS et Université Lyon I  
<https://mirror.ibcp.fr/pub/CRAN/>
3. Institut de Biologie du Développement de Marseille, CNRS et Université Aix-Marseille  
<https://cran.biotoools.fr/>
4. Institut de Radioprotection et de Sûreté Nucléaire, Paris  
<https://cran.irsn.fr/>

Ces miroirs proposent un accès sécurisé avec le protocole https.

### 1.1.3 Remarques sur le langage R avant l'installation

- R fonctionne avec une grande variété de systèmes d'exploitation et en particulier avec Microsoft Windows, Mac OS X et de nombreux systèmes de type Unix. R est soit disponible sous la forme de fichiers prêts à être installés soit sous la forme de fichiers source à compiler soi-même.
- Il existe une version française du langage R même si le site officiel est rédigé en langue anglaise.
- La très grande majorité des fonctions du langage ne diffère pas d'un système d'exploitation à l'autre bien que les interfaces graphiques ne sont pas similaires.

Il existe plusieurs interfaces graphiques, en anglais « GUI » pour *Graphical User Interface*, qui permettent d'accéder à une partie des fonctions du langage R.

- RGUI, l'interface graphique installée par défaut sous Windows.
- JGR, une interface graphique programmée en Java pour R. Elle fonctionne pour tous les systèmes d'exploitation sur lesquels le langage Java est disponible et donc aussi bien pour Microsoft Windows que pour Mac OS X et de nombreux systèmes de type Unix.
- Rattle, une interface graphique pour le data mining utilisant R.
- R Commander, une interface graphique pour faire des statistiques usuelles avec R.
- Statistical Lab.
- RExcel, pour exécuter les fonctionnalités de R et de R Commander à partir de Microsoft Excel.
- rggobi, une interface pour le logiciel GGobi spécialisé dans la visualisation de données multidimensionnelles.
- RKWard, une interface graphique basée sur les bibliothèques de KDE, une interface graphique disponible sur de nombreux systèmes de type Unix.

Il existe également des programmes qui facilitent l'écriture des instructions et des programmes en langage R.

Ces programmes se regroupent en deux grandes catégories :

- Les éditeurs de texte comme le bloc-note de Microsoft Windows, Microsoft Wordpad ou Microsoft Word.
- Les environnements de programmation, appelés en anglais *Integrated development environments* (IDEs), comme Tinn-R, Emacs (*Emacs Speaks Statistics*), Jedit, Kate, WinEdt (R Package RWinEdt) ou Vim.

Pour les utilisateurs qui en auraient besoin, par exemple pour rédiger un mémoire ou un rapport de stage, il est également possible d'intégrer directement des résultats obtenus avec R dans des documents  $\text{L}^{\text{T}}\text{E}^{\text{X}}$  à l'aide du langage Sweave ou dans des documents au format OpenDocument (ODF) à l'aide du langage odfWeave.

Il existe plusieurs sites internet, principalement en langue anglaise, consacrés au langage R.

- La section sur R à l'Open Directory Project.
- RSeek et R site search qui sont des moteurs de recherche spécialisés dans la recherche de documents consacrés à R.
- Plusieurs listes de diffusion permettent aux utilisateurs novices ou expérimentés de R de poser directement leurs questions aux autres utilisateurs afin qu'ils leur fassent partager leur expérience du langage.
- Le R Journal est un journal à comité de lecture comportant des articles consacrés aux problèmes de calcul statistique et au développement du langage R. Il peut aussi bien intéresser de simples utilisateurs du langage que des programmeurs.
- R Books comprend une liste importante de livres consacrés à R.
- Le R Graphical Manual explique par l'exemple comment construire des graphiques à partir de n'importe quelle bibliothèque du langage. Il comporte également un index exhaustif des fonctions de toutes les bibliothèques d'extensions du langage R.
- Le R Wiki est un site coopératif de documentation sur le langage R.

## 1.2 MAINTENANT C'EST À VOUS

### 1.2.1 Installer le langage R

Ces premières instructions sont communes aux trois systèmes d'exploitation.

1. Rendez-vous sur le site <https://www.r-project.org/>.
2. Puis, à gauche sur la page d'accueil, vous trouverez un menu Download. Dans ce menu, cliquez sur CRAN.
3. Choisissez un site miroir proche de chez vous.

4. Un encadré blanc intitulé `Download and Install R` doit apparaître sur votre écran.



### 1.2 Installation de la version 4.2.1 (2022-06-23) de R sous Windows

1. Cliquez sur Download R for Windows puis sur `base`.
2. Un encadré grisé doit apparaître dans lequel, à la première ligne, est inscrit `Download R-4.2.1 for Windows (79 megabytes, 64 bits)`, la version 4.2.1 étant celle disponible au mois d'août 2022. Cliquez dessus.
3. Procédez au téléchargement.
4. Exécutez le fichier que vous venez de télécharger en choisissant une installation par défaut.

**Remarque :** le téléchargement de R n'est pas très long. En effet, sa taille est de 79,0 Mo.



### 1.3 Installation de la version 4.2.1 (2022-06-23) de R sous Mac OS X

1. Cliquez sur Download R for macOS.
2. Une liste de fichiers à télécharger, intitulée `Files` : apparaît. Le premier élément de la liste est `R-4.2.1.pkg`, la version 4.2.1 étant celle disponible au mois d'août 2022. Cliquez dessus. Attention, il vous faudra peut-être, en fonction de votre navigateur internet, appuyer sur la touche « control » en cliquant pour pouvoir télécharger le fichier.
3. Pour installer R, double-cliquez sur l'icône du package d'installation `R-4.2.1.pkg`.



### 1.4 Installation de la version 4.2.1 (2022-06-23) de R sous Linux

1. Cliquez sur Download R for Linux (Debian, Fedora/Redhat, Ubuntu) puis sur le nom de la distribution Linux installée sur votre ordinateur.
2. Suivez les instructions détaillées sur le site. Celles-ci varient trop d'une distribution à l'autre pour être reproduites ici mais si vous utilisez Linux vous ne devriez pas avoir de mal à installer ou à faire installer le langage par votre administrateur réseau.

**Remarque :** dans la grande majorité des situations, l'installation de R est très simple et ne nécessite que peu de connaissances techniques.

## 1.2.2 Le fonctionnement du langage R sous les trois principaux systèmes d'exploitation

L'interface graphique du langage est très similaire d'un système d'exploitation à l'autre. Voici, en détails, la procédure à suivre dans le cas de l'environnement Windows.



### ENCART 1.5 Environnement Windows

- R fonctionne avec plusieurs fenêtres sous Windows. La fenêtre R Console est la fenêtre principale où sont réalisées par défaut les entrées de commandes et les sorties de résultats en mode texte. À celle-ci peuvent s'ajouter des fenêtres facultatives, telles que les fenêtres graphiques, les fenêtres d'informations (historique des commandes, aide, visualisation de fichier, etc), toutes appelées par des commandes spécifiques via la fenêtre R Console.
- 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 des commandes exécutées, etc.
- Le menu Edit ou Edition contient les habituelles commandes de copier-coller, ainsi que la boîte de dialogue autorisant la personnalisation de l'apparence de l'interface.
- Le menu View ou Voir permet d'afficher ou de masquer la barre d'outils et la barre de statut.
- Le menu Misc traite de la gestion des objets en mémoire et permet d'arrêter un calcul ou des calculs en cours de traitement.
- Le menu Packages automatise la gestion et le suivi des bibliothèques de fonctions, permettant leur installation et leur mise à jour de manière transparente depuis l'un des miroirs du CRAN (Comprehensive R Archive Network) <https://cran.r-project.org/>.
- Enfin, le menu Windows ou Fenêtres et le menu 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 à l'aide en ligne et aux manuels de références du langage R.

## 1.2.3 Installer des packages du langage R

**Qu'est ce qu'un package ?** Un package est une compilation d'outils. Certains sont déjà présents dans l'installation de base de R. En effet, lors de l'installation de R, un dossier `library` s'est créé par défaut. Il comprend les packages de base de R. Mais d'autres packages qui vous seront utiles pour réaliser vos analyses statistiques seront à télécharger puis à installer.





### ENCART 1.6 Pour les trois environnements

1. Reprenez la procédure de téléchargement de R vue à la section « Installer le langage R ».
2. Cette fois-ci au lieu de cliquer sur Windows, MacOS X ou Linux, cliquez sur packages dans la liste intitulée Source Code for all Platforms.
3. Une page apparaît sur laquelle est indiqué le nombre de packages actuellement disponibles sur le CRAN (18 489 fin août 2022). Pour obtenir la liste des packages rangés dans l'ordre alphabétique, cliquez sur Table of available packages, sorted by name.
4. La liste des packages apparaît alors.
5. Puis cliquez sur le package dont vous avez besoin.
6. Une brève description du package apparaît suivie d'une liste proposant plusieurs versions du package. La première (.tar.gz) est celle contenant le code source du package et ne sert a priori qu'aux utilisateurs de Linux. La deuxième (.tgz) est destinée aux utilisateurs de Mac OS X et la troisième (.zip) aux utilisateurs de Windows.
7. Sous Mac OS X et Windows, il faut alors démarrer l'interface graphique de R.
8. Sous Mac OS X, allez dans le menu Packages et utilisez le Package Manager. Indiquez alors à R le fichier .tgz que vous venez de télécharger.
9. Sous Windows, allez dans le menu Packages et choisissez Installer depuis un fichier .zip. Indiquez alors à R le fichier .zip que vous venez de télécharger.

**Remarque :** vous verrez bientôt comment installer très facilement les packages disponibles sur le site du CRAN directement depuis R.



### ENCART 1.7 Environnement Windows

Il existe une procédure alternative pour les utilisateurs d'un environnement Windows.

1. Reprenez la procédure de téléchargement de R vue à la section « Installer le langage R ».
2. Cette fois-ci au lieu de cliquer sur base, cliquez sur contrib.
3. Cliquez ensuite sur le dossier de la version R que vous avez installée.
4. Puis cliquez sur le package dont vous avez besoin.
5. Un fichier .zip est enregistré sur votre disque dur.
6. Démarrez l'interface graphique de R, allez dans le menu Packages et choisissez Installer depuis un fichier .zip. Indiquez alors à R le fichier .zip que vous venez de télécharger.



Lorsque vous aurez besoin de `packages` qui ne sont pas installés par défaut pour réaliser les analyses statistiques qui vous seront demandées dans les exercices, cela vous sera signalé. Il faudra alors installer ces `packages` supplémentaires sur votre ordinateur.



Beaucoup de bibliothèques contiennent un ou plusieurs documents détaillant leurs fonctionnalités et montrant leur application pas à pas à un exemple. Pour obtenir la liste des vignettes présentes sur votre ordinateur, il suffit d'exécuter la commande `vignette()`.

## 1.3 VOS PREMIÈRES LIGNES DE COMMANDE

### 1.3.1 Démarrer R

Pour démarrer R, vous pouvez par exemple lancer le langage R en double-cliquant sur l'icône R qui se trouve par exemple sur votre bureau. La fenêtre R console s'ouvre. Elle vous affiche tout ce texte :

```
R version 4.2.1 (2022-06-23) - "Funny-Looking Kid"
Copyright (C) 2022 The R Foundation for Statistical Computing
Platform : x86_64-apple-darwin17.0 (64-bit)

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.

[R.app GUI 1.79 (8095) x86_64-apple-darwin17.0]
```

Puis sous ce texte, le symbole rouge `>`, appelé `prompt`, apparaît également. Il signifie que R est prêt à travailler.



Il ne faut jamais taper ce symbole au clavier car il est déjà présent en début de ligne sur la fenêtre R Console.

C'est à la suite de `>` que vous taperez les lignes de commande de R. Une fois la commande tapée, vous devez toujours la valider en appuyant sur la touche « Entrée » du clavier.



Si votre commande est incomplète, le symbole `>` est remplacé par `+`. Ce symbole `+` signifie que R attend la suite de la commande. Si vous ne savez pas compléter la ligne de commande ou qu'elle présente une erreur vous pouvez appuyer sur la touche « Echap » pour annuler la commande et créer un nouveau symbole `>`.

### 1.3.2 Quitter R

Pour quitter R, vous utilisez la commande suivante :

```
> q()
```

```
Save workspace image? [y/n/c]
```

R vous propose de sauvegarder le travail effectué. Trois réponses vous sont proposées : y (pour *yes*), n (pour *no*) ou c (pour *cancel*, annuler).

Si vous tapez y, cela permet que les commandes exécutées pendant la session et les objets enregistrés en mémoire soient conservés et soient donc « rappelables » et « réutilisables ».

Si vous tapez n, vous quittez R qui oubliera tout le travail que vous avez réalisé. Attention, vous risquez de tout perdre !

Si vous tapez c, la procédure de fin de session sous R est annulée.

Sous l'environnement Windows ou Mac OS X, c'est une boîte de dialogue en français qui apparaîtra à l'écran et qui comportera les mêmes questions.

### 1.3.3 Sauvegarder sous R

Si vous quittez R en choisissant la sauvegarde de l'espace de travail, deux fichiers sont créés :

1. le fichier `.Rdata` ou `.Rapp.history` contient des informations sur les variables utilisées,
2. le fichier `.Rhistory` contient l'ensemble des commandes utilisées.

### 1.3.4 Consulter l'aide de R

Il y a quatre sources principales d'aide :

1. les fichiers d'aide,
2. les manuels,
3. les archives R-help,
4. et enfin R-help lui-même.

Pour une fonction, dont le nom est `fonction`, vous pouvez consulter une fiche de documentation en tapant `?fonction` ou `help("fonction")`. Grâce à cette aide, il suffit que vous reteniez le nom de la fonction, mais pas forcément toute la syntaxe.

**Exemple** : vous cherchez à obtenir des informations sur la fonction `read.table`, vous taperez alors la commande suivante :

```
> ?read.table
```

ou encore :

```
> help(read.table)
```

Pour une bibliothèque d'extension, dont le nom est `package`, vous pouvez consulter une fiche de documentation en tapant :

```
> help(package="package")
```

Grâce à cette aide, il suffit que vous reteniez le nom de la bibliothèque où se trouve la fonction que vous souhaitez utiliser, mais pas forcément le nom exact de cette fonction.

Les pages d'aide sont généralement très détaillées. Elles contiennent souvent, entre autres :

- une section *See Also* qui donne les pages d'aide sur des sujets apparentés.
- une section *Description* qui précise ce que fait la fonction.
- une section *Examples* avec des lignes de commande illustrant ce que fait la fonction documentée. Ces exemples peuvent être exécutés directement en utilisant la fonction `example`, essayez par exemple :

```
> example(plot)
```

Pour afficher successivement les différents graphiques ainsi créés, vous devez cliquer plusieurs fois de suite sur la fenêtre où sont situés les graphiques.

### a) Affichage de l'aide dans la console

Lorsque l'aide s'affiche dans la console, vous pouvez faire défiler le texte ligne par ligne avec la touche « Entrée » ou « Flèche vers la bas » ou page par page en appuyant sur la barre « Espace ». Une fois arrivé à *END*, tapez `q`.

Il s'agit du mode d'affichage par défaut de l'aide dans un terminal R et donc dans un environnement Linux.

### b) Affichage de l'aide en-dehors de la console

Lorsque l'aide s'affiche dans la console, elle n'est pas facile à consulter. Il existe des versions au format `.html` de tous les fichiers d'aide dans les trois environnements Linux, MacOS X et Windows. Pour s'en servir à la place des versions texte qui s'affichent dans la console, il faut utiliser l'option `help_type="html"` de la fonction `help`.

**Exemple :** vous cherchez à obtenir des informations au format `.html` sur la fonction `read.table`, vous taperez alors la commande suivante :

```
> help("read.table", help_type="html")
```

L'option `help_type="text"` de la fonction `help` permet d'afficher l'aide au format texte `.txt`.

**Exemple** : vous cherchez à obtenir des informations au format `.txt` sur la fonction `read.table`, vous taperez alors la commande suivante :

```
> help("read.table",help_type="text")
```

Le format `.html` est celui utilisé par défaut lorsque vous utilisez l'interface graphique de R dans un environnement Mac OS X ou Windows.

La fonction `help.start` permet d'accéder à la page d'accueil de l'aide au format `.html`.

```
> help.start()
```

### c) Changer le mode d'affichage par défaut de l'aide

Il est possible de modifier une des options de R qui gère le mode d'affichage par défaut des fichiers d'aide de R. Pour afficher, pour toute la durée d'une session R, les fichiers au format :

- `.html`, tapez l'instruction `options(help_type="html")` dans la console.
- `.txt`, tapez l'instruction `options(help_type="text")` dans la console.



À la fin de l'aide, il y a presque toujours une ou plusieurs lignes de commande d'exemple. Il est judicieux de les exécuter afin d'avoir une idée de ce que les fonctions ont besoin comme informations et des résultats qu'elles vous renvoient. Pour cela, il suffit de copier ces lignes, de les coller sur la fenêtre R `console` et d'observer ce qui se passe. Vous pouvez, bien sûr, les modifier suivant vos besoins. De plus, ces lignes de commande sont souvent un exemple d'analyse de données et pointent parfois sur d'autres fonctions utiles à essayer. La notion de « fonctions » sous R sera développée dans le chapitre suivant.

## 1.3.5 Travailler avec R

Cette section va être courte car elle sera limitée, dans le cadre de ce sous-paragraphe, à quelques fonctionnalités de R.

### a) R comme calculatrice

**Exemple** : tapez la ligne de commande suivante et validez par la touche « Entrée » du clavier :

```
> 2 + 8
```

Le résultat s'affiche sous la forme :

```
[1] 10
```

Le chiffre 1 entre crochets indique l'indice du premier élément de la ligne. Le second chiffre, 10, est le résultat de la ligne de commande.

**Exemple :** voici un autre exemple. Tapez la ligne de commande suivante :

```
> 120:155
```

```
[1] 120 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135
[17] 136 137 138 139 140 141 142 143 144 145 146 147 148 149 150 151
[33] 152 153 154 155
```

Dans le résultat ci-dessus, l'indice de l'élément 120 est 1, celui de 136 est 17 et celui de 152 est 33.



Le chiffre qui se trouve entre les crochets indique l'indice du premier élément de la ligne sur laquelle il se trouve.

**Exemple :** une fonction dont vous pourriez avoir besoin est la fonction `sqrt`, qui n'est rien d'autre que la racine carrée d'un nombre. Tapez la ligne de commande suivante :

```
> sqrt(4)
```

```
[1] 2
```

### b) R comme logiciel de statistique

Vous obtiendrez des précisions sur cette fonctionnalité tout au long des chapitres suivants.

### c) R comme éditeur graphique

Vous obtiendrez des précisions sur cette fonctionnalité tout au long des chapitres suivants.

## 1.3.6 Écrire des scripts avec un éditeur de texte et les compiler sous R

Il est souvent plus pratique de composer le code R dans une fenêtre spécifique de l'interface graphique : la fenêtre de script.

Les entrées `Nouveau script` ou `Ouvrir un script` permettent de créer un nouveau script de commandes R ou d'accéder à un ancien script sauvegardé lors d'une session précédente d'utilisation du logiciel.

Pour exécuter des instructions à partir de la fenêtre de script il suffit de procéder par copier-coller ou de se servir du raccourci clavier « ctrl+R ».

Pour sauvegarder un script, il suffit, lorsque la fenêtre de script est active, de sélectionner l'entrée « Sauver » du menu « Fichier ».

Des scripts s'écrivent avec un éditeur de texte.

**Exemple :** Winedt, Microsoft Wordpad, Open Office, vi, emacs ou Xemacs, etc, sont des éditeurs de texte.



Souvent il est proposé aux débutants d'utiliser le traitement de texte de Word car la plupart des mises en forme du texte n'affecte pas l'exécution des lignes de commande lors du « copier-coller ». Attention! Il faut désactiver la mise en forme automatique du texte.

**Remarque :** certains éditeurs, comme RStudio, emacs, Xemacs ou Tinn-R peuvent vous être conseillés car ils vous rendront la vie plus facile en fournissant la tabulation intelligente, la coloration syntaxique et le complément des commandes.

Indépendamment de l'éditeur, vous sauvegardez les scripts dans un de vos répertoires. Ensuite deux solutions s'offrent à vous :

1. soit vous utilisez le copier-coller sur la fenêtre R Console,
2. soit vous les lisez avec l'une des trois commandes suivantes :

```
> source(file="C://chemin//vers//nomdefichier//fichier.R",echo=TRUE)
```

ou :

```
> source(file="../../../repertoire/fichier.R",echo=TRUE)
```

ou encore :

```
> source("fichier.R",echo=TRUE)
# Si "fichier.R" est dans le répertoire de travail
```



1. Notez l'utilisation des slashes pour séparer les répertoires, même sous l'environnement Windows.
2. Notez la présence du symbole #. Sous R, ce symbole est le symbole des commentaires. Tout ce qui suit un # est alors ignoré. Sachez qu'un élément clef d'une bonne écriture de script est la présence abondante de commentaires.



Écrire des scripts lisibles et commentés est une habitude à prendre dès maintenant car cela vous rendra l'utilisation de R bien plus facile.



### ENCART 1.8 Les quatre avantages d'écrire des scripts

1. Écrire des scripts est un gain de temps car vous pouvez facilement réutiliser, en les adaptant si nécessaire, des lignes de commande déjà tapées.
2. Si vous avez beaucoup de lignes de commande à écrire, c'est beaucoup plus simple de les manipuler, de les modifier dans un éditeur de texte.
3. Écrire des scripts est un outil de collaboration puissant. C'est souvent pratique de pouvoir envoyer à un de vos camarades ou de vos collègues, par fichier attaché dans un mail par exemple, votre code et le fichier de données brut associé et de savoir qu'il lui suffit d'exécuter la fonction `source` sur votre code pour effectuer votre analyse sur sa machine.
4. Enfin il n'existe pas de message d'alerte dans R sauf quand vous quittez R. Vous pouvez alors perdre des données sans vous en rendre compte. Le seul moyen de trouver l'erreur est de recommencer l'écriture de la ligne de commande.



Il est à noter qu'il existe également des langages de programmation libres et gratuits comme `Tinn-R` qui sont destinés à vous faciliter la rédaction de scripts.

## 1.4 QUELQUES REMARQUES SUR LA FENÊTRE R CONSOLE

1. Ce qui est entré par l'utilisateur figure en rouge et la réponse de R est en bleu.
2. R utilise le système anglo-saxon pour les nombres décimaux c'est-à-dire les décimales sont séparées par un point et non par une virgule comme en France.
3. R distingue les majuscules des minuscules.
4. Vous devez faire attention à l'utilisation du point virgule. En effet, sous R, ce dernier sert à séparer deux instructions.
5. Vous pouvez rappeler les commandes déjà exécutées en utilisant la touche « Flèche vers le haut ».
6. Vous pouvez parcourir la ligne de commande que vous êtes en train d'écrire en appuyant sur les touches « Flèche vers la gauche » et « Flèche vers la droite ».

## 1.5 POUR EN SAVOIR PLUS SUR LE LANGAGE R

- Un certain nombre de livres écrits principalement en anglais, mais aussi en français, paraissent chaque année. Vous trouverez, sur le site <http://www.r-project.org/manuals.html> dans l'onglet `Documentation`, une liste de manuels d'introduction ou très spécialisés est à votre disposition.



- Pour un public francophone, un point de départ peut être le polycopié d'Emmanuel Paradis, téléchargeable en ligne, intitulé « R pour les débutants », 77 pages (au mois de janvier 2018), qui a la particularité d'exister également en version anglaise « R for Beginners ». Les deux documents sont disponibles à cette adresse <http://cran.r-project.org/> dans la rubrique « Documentation », sous-rubrique « Contributed ».
- Plusieurs milliers de pages d'enseignement de statistiques sous R, rédigées en langue française, sont disponibles à cette adresse <http://pbil.univ-lyon1.fr/R/>.
- Il existe aussi des groupes ou des foires aux questions autour de R auxquels vous pouvez vous abonner.
- Il a existé les R News mais maintenant ces nouvelles sont remplacées par le R Journal. Les articles de ces revues ont pour objectif de mettre en avant certaines bibliothèques de fonctions particulièrement intéressantes.
- Bonne lecture à tous !

## 1.6 R SANS LES MAINS

Il existe plusieurs interfaces graphiques qui ont vocation à faciliter l'utilisation du langage R. En voici deux parmi celles-ci qui méritent une attention particulière.

- RStudio, <https://www.rstudio.com>, est avant tout un outil puissant pour écrire facilement des scripts, des fonctions voire des bibliothèques R. C'est un programme à installer séparément de R et disponible pour les trois environnements Windows, Mac OS X et Linux. Il présente dans un même environnement le script, la console R, la liste ainsi que le contenu des objets présents dans la mémoire de R ainsi que les pages d'aide consultées ou les graphiques produits. Il permet également de transformer automatiquement un script R en un fichier au format html où sont intercalées les commandes R avec les résultats de celles-ci.
- Rcmdr (RCommander) est une bibliothèque pour le langage R, disponible pour les trois environnements Windows, Mac OS X et Linux. C'est avant tout une interface graphique pour un grand nombre de fonctions usuellement utilisées en statistique. Son utilisation rend la pratique de R proche de celles d'autres logiciels de statistique « à menus et boîtes de dialogue » comme par exemple SPSS, Minitab ou Statistica. Elle permet en outre d'importer facilement des fichiers au format .csv, SPSS, SAS ou Minitab. Cette bibliothèque permet, elle aussi comme RStudio, de transformer automatiquement un script R en un fichier au format html où sont intercalées les commandes R avec les résultats de celles-ci.

Pour vous montrer la facilité avec laquelle il est possible de produire ces rapports html ainsi que le rendu final obtenu, vous trouverez en ligne, pour chacun des chapitres du livre, des rapports, faits avec RStudio et Rcmdr.

## EN RÉSUMÉ

Commande	Description
<code>q()</code>	Quitte le langage.
<code>?fonction</code> ou <code>help("fonction")</code>	Demande la fiche de documentation de la fonction.
<code>help(package="nomdupackage")</code>	Demande la fiche de documentation du package intitulé <code>nomdupackage</code> .
<code>install.packages("nomdupackage")</code>	Installe le package intitulé <code>nomdupackage</code> .
<code>n:m</code>	Crée une suite de nombres d'entiers de <code>n</code> à <code>m</code> .
<code>seq(n,m,i)</code>	Crée une suite de nombres de <code>n</code> à <code>m</code> en incrémentant par <code>i</code> .
<code>rep(s,n)</code>	Crée une suite contenant <code>n</code> fois <code>s</code> .
<code>source("fichier")</code>	Exécute le script contenu dans <code>fichier</code> .

*Remarque* : certaines commandes ne sont pas encore apparues dans ce premier chapitre mais vont être introduites dans l'exercice 1.2 de ce chapitre.

## EXERCICES

---

### Exercice 1.1 Installation de R

- 1) Procédez au téléchargement de R et de la bibliothèque BioStatR.
- 2) Consultez l'aide de la bibliothèque BioStatR.

---

### Exercice 1.2 Suites

Pour créer une suite d'entiers sous R, plusieurs fonctions sont disponibles.

La **première méthode** est d'utiliser l'opérateur `:`. Une suite d'entiers consécutifs de `a` à `b` s'obtient par `a:b`.

La **seconde méthode** est d'utiliser la fonction `seq` qui se caractérise par **trois arguments** : le premier, noté `from` par R, est celui qui indique d'où part la suite, le deuxième, noté `to` par R indique la fin de la suite et le troisième et dernier, noté `by` par R, mentionne le pas de la

progression également appelé raison. La suite d'entiers allant de a à b en avançant par pas de k est créée par la ligne de commande suivante :

```
> seq(from=a,to=b,by=k)
```

**Remarque :** ne tapez pas cette ligne de commande en l'état. Il faut que vous remplaciez les valeurs a, b et k par celles que vous avez choisies.

**Remarque :** seq peut créer une suite d'entiers mais aussi une suite de réels ce que ne fait pas l'opérateur ":". En fait, ":" est un cas particulier de la fonction seq avec un pas toujours égal à 1.

- 1) Créez la suite d'entiers consécutifs de 10 à 25 avec les deux méthodes.
- 2) Créez la suite d'entiers en progression arithmétique de premier terme 20, de dernier terme 40 et de raison 5.

Il existe une fonction, la fonction rep, qui peut servir à construire des suites particulières comme les suites constantes ou les suites périodiques. rep se caractérise par **deux arguments** : le premier, noté x par R, est celui qui comporte les chiffres qui composent le motif qui est répété dans la suite et le second, noté times par R, est le nombre de fois que vous souhaitez répéter x.

- 3) Créez la suite dont tous les termes sont identiques et égaux à 28 et de longueur 10.

## DU MAL À DÉMARRER

### ⇒ Exercice 1.1

- 1) Relisez le paragraphe 1.2.1 du chapitre 1.
- 2) Relisez le paragraphe 1.2.3 du chapitre 1.

### ⇒ Exercice 1.2

- 1) Lisez l'aide de la fonction ":", en tapant "?":", et celle de la fonction seq.
- 2) Utilisez la fonction seq.
- 3) Utilisez la fonction rep.

## SOLUTION DES EXERCICES

---

### Exercice 1.1

- 1) Les indications pour télécharger et installer R sont détaillées dans le début de ce chapitre.
- 2) Pour télécharger la bibliothèque depuis l'un des sites miroirs du CRAN, il suffit d'exécuter la ligne suivante :

```
> install.packages("BioStatR")
```

L'option `type` force R à utiliser l'un des trois types de format de fichier pour les bibliothèques d'extension : `type="source"` pour les fichiers source, `type="mac.binary"` pour les binaires Mac OS X et `type="win.binary"` pour les binaires Windows. Il faut bien sûr penser à choisir un format compatible avec son environnement.

Enfin les fichiers de la bibliothèque aux formats source `.tar.gz`, principalement pour Linux, binaire `.tgz`, pour Mac OS X, ou binaire `.zip`, pour Windows, sont également hébergés sur le CRAN.

Pour consulter l'aide de la bibliothèque BioStatR, utilisez la fonction `help(package="BioStatR")`.

Si la bibliothèque est indisponible sur le CRAN au moment où vous en avez besoin, il est possible de la télécharger depuis le dépôt github sur internet des auteurs en utilisant la fonction `install_github` de la library `devtools`.

```
> library(devtools)
> install_github("fbertran/BioStatR")
```



Ce qu'il faut retenir de cet exercice :

La fonction `install.packages` permet d'installer des bibliothèques de fonctions.

---

### Exercice 1.2

- 1) Vous souhaitez créer la suite d'entiers de 10 à 25.  
Avec la **première méthode**, vous tapez alors la ligne de commande suivante :

```
> 10:25
```

```
[1] 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
```

La **seconde méthode** fait appel à la fonction `seq` :

```
> seq(from=10,to=25,by=1)
```

```
[1] 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
```

**Remarque** : vous pouvez aussi écrire simplement :

```
> seq(10,25,1)
```

```
[1] 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
```

**2)** Vous pouvez encore utiliser la fonction `seq` :

```
> seq(from=20,to=40,by=5)
```

```
[1] 20 25 30 35 40
```

**Remarque** : vous auriez pu également abrégé la dernière ligne de commande en tapant celle-ci :

```
> seq(20,40,5)
```

Elle vous donne le même résultat que précédemment :

```
[1] 20 25 30 35 40
```

**3)**

```
> rep(x=28,times=10)
```

```
[1] 28 28 28 28 28 28 28 28 28 28
```

**Remarque** : vous auriez pu également abrégé la dernière ligne de commande en tapant celle-ci :

```
> rep(28, 10)
```

Elle vous donne le même résultat que précédemment :

```
[1] 28 28 28 28 28 28 28 28 28 28
```



Ce qu'il faut retenir de cet exercice :

Les fonctions `seq` et `rep` permettent de créer rapidement des suites de nombres régulières.

## Chapitre 2

---

# Objets, fonctions et données avec le langage R

« Les mathématiques n'étudient pas des objets, mais les relations entre ces objets. »

H. Poincaré (1854-1912)

### Objectifs de ce chapitre

- définir ce que sont les objets sous R et les manipuler
- présenter les fonctions usuelles de R
- manipuler des données avec R, comme :
  - saisir des données sous R
  - importer des fichiers de données sous R
  - écrire des fichiers de données à partir de R

## 2.1 LE FICHIER DE DONNÉES IRIS

Dans ce chapitre, vous allez avoir besoin d'un jeu de données afin d'illustrer la présentation de certaines des commandes du logiciel. À l'heure actuelle, vous ne savez pas encore saisir ni importer des données sous R, mais ceci ne devrait pas tarder puisque ce point sera abordé à la fin de ce chapitre. C'est pourquoi vous allez utiliser un jeu de données déjà présent sous R. Comme vous l'avez déjà appris au chapitre 1, R est un ensemble de bibliothèques de fonctions appelées packages. Chaque package contient des jeux de données spécifiques qui sont utilisés comme exemples.

**Exemple** : pour connaître les jeux de données contenus dans le package `base`, tapez la commande suivante :

```
> data(package = "datasets")
```



Notez la présence des `"` autour de `datasets`, qui indique à R que `datasets` est une chaîne de caractères et qu'elle n'est à considérer que comme telle puisque c'est le nom du package à explorer.

Le résultat apparaît dans une fenêtre R data sets. En voici un extrait :

```
Data sets in package 'datasets':

AirPassengers.....Monthly Airline Passenger Numbers 1949-1960
BJsales.....Sales Data with Leading Indicator
BJsales.lead (BJsales)..Sales Data with Leading Indicator
BOD.....Biochemical Oxygen Demand
.....
iris.....Edgar Anderson's Iris Data
```



Notez la présence du fichier `iris`. Les données de ce fichier sont célèbres. Elles ont été collectées par Edgar Anderson<sup>1</sup>. Le fichier donne les mesures en centimètres des quatre variables suivantes :

1. longueur du sépale (`Sepal.Length`),
2. largeur du sépale (`Sepal.Width`),
3. longueur du pétale (`Petal.Length`) et
4. largeur du pétale (`Petal.Width`)

pour trois espèces d'iris :

1. `iris setosa`,
2. `iris versicolor` et,
3. `iris virginica`.

Sir R.A. Fisher<sup>2</sup> a utilisé ces données pour trouver des variables<sup>3</sup> permettant de séparer au mieux les trois espèces d'iris. Vous pouvez obtenir ces renseignements en tapant soit la commande suivante (rencontrée au chapitre 1)

```
> ?iris
```

soit la commande (également rencontrée au chapitre 1) :

---

1. E. Anderson, (1935), The irises of the Gaspé Peninsula, *Bulletin of the American Iris Society*, 59, 2-5.  
 2. R.A. Fisher, (1936), The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7, Part II, 179-188.  
 3. Il s'agit en fait de combinaisons linéaires des variables du jeu de données pour ceux qui connaissent cette notion.



```
> help(iris)
```

Le jeu de données `iris` est immédiatement accessible. Pour le visualiser en entier, il suffit de taper la commande suivante :

```
> iris
```

Voici les dix dernières lignes de ce que vous voyez à l'écran.

```
.....
141      6.7      3.1      5.6      2.4 virginica
142      6.9      3.1      5.1      2.3 virginica
143      5.8      2.7      5.1      1.9 virginica
144      6.8      3.2      5.9      2.3 virginica
145      6.7      3.3      5.7      2.5 virginica
146      6.7      3.0      5.2      2.3 virginica
147      6.3      2.5      5.0      1.9 virginica
148      6.5      3.0      5.2      2.0 virginica
149      6.2      3.4      5.4      2.3 virginica
150      5.9      3.0      5.1      1.8 virginica
```



Tapez `iris` et non `Iris` ou `IRIS` car R n'est pas indifférent aux lettres majuscules et aux lettres minuscules, comme vous l'avez déjà appris dans le chapitre 1.

## 2.2 QUELQUES DÉFINITIONS

**Définition 2.1** *Un objet est un espace dans lequel vous pouvez stocker tout ce qui vous intéresse.*

**Définition 2.2** *Un vecteur est un objet d'un même mode pour toutes les valeurs qui le constituent. Vous rencontrerez donc soit des vecteurs numériques, soit des vecteurs de caractères, soit des vecteurs logiques, soit des vecteurs vides. Un vecteur est constitué de **coordonnées** ou encore de **composantes**. Parfois, le mot **éléments** est utilisé à la place de coordonnées.*

**Remarque :** en mathématiques, un vecteur est souvent considéré comme une matrice particulière : une matrice à une ligne ou à une colonne. Dans le langage R, il n'en va pas de même et il faudra donc être très attentif au fait de ne pas confondre les vecteurs avec des matrices à une ligne ou à une colonne.

**Définition 2.3** *Une matrice est un objet d'un même mode pour toutes les valeurs qui la constituent. Comme les vecteurs, vous rencontrerez donc soit des matrices numériques, soit des matrices de caractères, soit des matrices logiques, soit des matrices vides. Contrairement au*

vecteur, la matrice peut avoir plus qu'une colonne. Chaque élément (et c'est élément ici qu'il faut utiliser) d'une matrice est repéré par son numéro de ligne et son numéro de colonne. Toutes les lignes d'une matrice ont la même longueur. Il en va de même pour ses colonnes. En revanche, la longueur des lignes peut être différente de celle des colonnes.

**Définition 2.4** Une **liste** est un objet permettant de stocker des objets qui peuvent être hétérogènes, c'est-à-dire qui n'ont pas tous le même mode ou la même longueur.

**Définition 2.5** Un **tableau de données**, ou `data.frame` en anglais, est une liste particulière dont les composantes sont de même longueur et dont les modes peuvent être différents. Un `data.frame` est un tableau à double entrée : les lignes sont les individus sur lesquels les mesures sont faites et les colonnes sont les variables.

**Remarque :** les jeux de données que vous utilisez seront souvent considérés comme des `data.frames`.

**Définition 2.6** Un **facteur** est un vecteur particulier qui permet de manipuler naturellement des variables qualitatives<sup>4</sup>.

## 2.3 QUELQUES MANIPULATIONS ÉLÉMENTAIRES

### 2.3.1 Affecter

Comme vous l'avez vu dans le chapitre précédent, R sait calculer. En fait R est une grosse calculatrice, mais vous n'avez pas installé R pour cette fonctionnalité-là. En effet, vous aimeriez parfois réutiliser le résultat d'une opération arithmétique sans avoir à le ressaisir ou à le copier/coller. Pour cela, vous affecterez des valeurs à des objets et utiliserez l'opérateur `<-`, appelé **opérateur d'affectation** ou d'assignation. Cet opérateur prend une valeur quelconque à droite et la place dans l'objet indiqué à gauche.

**Exemple :**

```
> n<-28
> N<-20
```

Que signifient ces deux lignes de commande ?

La première ligne de commande peut donc se lire ainsi « mettre la valeur 28 dans l'objet nommé `n` ». La deuxième ligne de commande peut donc se lire ainsi « mettre la valeur 20 dans l'objet nommé `N` ».

---

4. Le chapitre suivant précisera la différence entre les variables qualitatives et les variables quantitatives.

**Remarque :** R ne vous affiche rien dans la console. Pour que R vous renvoie le résultat de cette affectation, il faut lui demander de l'imprimer. Vous verrez comment y parvenir au paragraphe suivant.

Le signe = convient également pour faire des affectations.

**Exemple :**

```
> m=1973  
> m
```

```
[1] 1973
```

### 2.3.2 Afficher

Quand vous affectez un nom à un objet, l'affichage de celui-ci n'est pas automatique. Il faut que vous le demandiez en tapant uniquement le nom donné à l'objet.

**Exemple :**

```
> n
```

```
[1] 28
```

Vous pouvez aussi utiliser ces objets dans des calculs.

**Exemple :**

```
> N+n
```

```
[1] 48
```

Vous apprendrez dans la suite comment affecter et visualiser à l'aide de la même ligne de commande.

Vous pouvez utiliser autant d'objets que vous souhaitez. Les objets peuvent contenir non seulement des nombres comme vous venez de le voir mais aussi des chaînes de caractères, qui sont alors indiquées par des guillemets droits, et d'autres choses encore.

### 2.3.3 Supprimer

Par défaut R conserve en mémoire tous les objets créés lors de la session. Il n'y a que dans le cas où vous quittez R, sans sauvegarder la session, que les objets que vous avez créés sont supprimés.



Il est donc conseillé de supprimer régulièrement les objets que vous avez créés lors de votre session. Attention, vous devez être sûr de ne plus en avoir besoin car ils seront définitivement perdus.



Pour savoir quels sont les objets qui ont été créés pendant votre session, utilisez, à votre convenance, la commande `ls()` ou la commande `objects()`.

Si vous souhaitez supprimer :

- un objet, par exemple l'objet `m`, utilisez alors la commande `remove`, abrégée en `rm` :

```
> rm(m)
```

- plusieurs objets, par exemple les objets `n` et `N`, utilisez alors la ligne de commande suivante :

```
> rm(n, N)
```

- tous les objets en mémoire, utilisez alors la ligne de commande suivante :

```
> rm(list = ls())
```

## 2.4 LES OBJETS SOUS R

### 2.4.1 Généralités

R est un langage orienté objet, comme vous l'avez appris au chapitre 1, ce qui signifie que R fonctionne avec des objets.

Les différents objets que vous rencontrerez sous R se caractérisent par :

- **un nom** : dans R, le nom doit commencer par une lettre et ne peut comporter que des lettres, des chiffres, des espaces soulignés et des points.  
En aucun cas, vous ne devez utiliser des espaces ou des caractères spéciaux comme les ponctuations autres que le point. Il est fortement déconseillé d'utiliser des lettres à accent ou la cédille,
- **une classe** : un vecteur, une matrice (tableau à deux dimensions), un `array` (tableau à  $n$  dimensions), un `data.frame`, une liste, un facteur, une table,
- **un mode** : numérique (nombre réel), caractère (chaîne de caractères à mettre entre guillemets), logique (booléen), `NULL` (objet vide), liste (liste d'objets), fonction (une fonction),
- **une taille** : la taille d'un objet est le nombre d'éléments qu'il contient.

**Exemple** : vous voulez déterminer les caractéristiques classe, mode et taille de l'objet `iris`.

Pour connaître la **classe de l'objet** `iris` de R, tapez la commande :

```
> class(iris)
```

```
[1] "data.frame"
```

Pour connaître le **mode de l'objet** `iris` de R, tapez la commande :

```
> mode(iris)
```

```
[1] "list"
```

Vous cherchez à connaître les **noms des colonnes de l'objet** `iris` de R. Pour cela, vous tapez la commande :

```
> names(iris)
```

```
[1] "Sepal.Length" "Sepal.Width" "Petal.Length" "Petal.Width" "Species"
```

Pour connaître la **taille d'un objet**, il existe dans R, deux fonctions : `length` et `dim`. `length` renvoie le nombre d'éléments d'un vecteur, d'une matrice, d'un `array`, d'une liste ou d'un facteur, le nombre de colonnes d'un `data.frame` et le nombre de cases d'une table.

**Exemple** : vous cherchez à connaître la **taille de l'objet** `iris` de R, sachant que `iris` est un `data.frame`. Donc si vous tapez la commande :

```
> length(iris)
```

R vous affiche alors :

```
[1] 5
```

ce qui est logique puisque le `data.frame iris` est constitué de cinq colonnes. Maintenant si vous tapez la commande :

```
> dim(iris)
```

R vous affiche :

```
[1] 150 5
```

Ce résultat est plus complet que le précédent. En effet, il vous indique que le `data.frame iris` est un tableau constitué de 150 lignes et de 5 colonnes.



1. La fonction `length` comptabilise les valeurs manquantes dans la longueur des vecteurs.
2. Si vous cherchez à savoir si l'objet étudié possède ou non des valeurs manquantes, vous utiliserez alors la commande `is.na`.
3. La fonction pour connaître la **taille d'un objet** est soit `dim` soit `length`.

Il est possible de convertir le mode d'un objet à un autre mode en utilisant les commandes suivantes :

```
as.character
as.list
as.logical
as.numeric
```

Si vous souhaitez tester si un objet est d'un certain mode, il faut utiliser les commandes suivantes :

```
is.character
is.list
is.logical
is.numeric
is.null
```

*Remarque* : il existe des commandes similaires pour la classe d'un objet.

## 2.4.2 Les vecteurs

### a) Créer des vecteurs

Pour créer un vecteur, différentes commandes sous R sont disponibles. Vous en avez déjà vu trois dans l'exercice 1.2 du chapitre précédent :

1. l'opérateur `" : "`,
2. la fonction `seq`,
3. et la fonction `rep`.

Cette liste peut être complétée en vous introduisant :

4. la fonction `c` (c comme « collection »).

### • Les vecteurs numériques

**Exemple** : tapez les lignes de commande suivantes :

```
> serie1<-c(1.2,36,5.33,-26.5)
> serie1
```

```
[1]  1.20  36.00  5.33 -26.50
```

**Remarque :** que remarquez-vous sur cette ligne de sortie ? R a modifié l'écriture des nombres saisis au clavier en affichant ces nombres décimaux avec la même précision de deux chiffres après la virgule.

```
> mode(serie1)
```

```
[1] "numeric"
```

Donc, vous en déduisez que les éléments du vecteur `serie1` sont des nombres réels.

```
> class(serie1)
```

```
[1] "numeric"
```

Ici le résultat `numeric` signifie que l'objet `serie1` est un vecteur contenant des nombres réels.

Il faut donc faire attention aux deux sens différents du résultat `numeric` lorsqu'il est obtenu avec la commande `class` ou avec la commande `mode`.

Si vous tapez la ligne de commande suivante :

```
> c(1.2,36,5.33,-26.5)
```

Que fait le logiciel R ? R ne retourne pas `>` mais affiche :

```
[1]  1.20  36.00  5.33 -26.50
```

Dans ce cas, le résultat de la commande est affiché directement et n'est pas stocké en mémoire car vous n'avez pas affecté ce résultat à un objet.

Pour affecter un résultat à un objet et visualiser immédiatement ce résultat, il suffit de mettre la ligne de commande R entre parenthèses :

```
> (serie1<-c(1.2,36,5.33,-26.5))
```

```
[1]  1.20  36.00  5.33 -26.50
```





Quand vous donnez un nom à un objet, l'affichage de celui-ci n'est pas automatique. Vous le demandez en tapant le nom donné à l'objet.

### • Les vecteurs de chaînes de caractères

**Exemple :** tapez les lignes de commande suivantes :

```
> serie2<-c("bleu","vert","marron")
> serie2
```

```
[1] "bleu" "vert" "marron"
```

```
> mode(serie2)
```

```
[1] "character"
```

Donc, vous en déduisez que les éléments du vecteur `serie2` sont des caractères.

#### **Remarque :**

1. Si vous mélangez, dans un même vecteur, des caractères et des valeurs numériques, les valeurs numériques sont automatiquement converties en chaînes de caractères.
2. Quand les éléments du vecteur sont des chaînes de caractères, il est obligatoire de les déclarer entre guillemets, sinon R ne reconnaît pas les coordonnées du vecteur.

#### **Exemple :**

```
> serie2<-c(bleu,vert,marron)
```

```
Erreur : objet 'bleu' introuvable
```

ou si vous utilisez la version anglaise de R :

```
Error: Object 'bleu' not found
```

### • Les vecteurs logiques

**Exemple :** tapez les lignes de commande suivantes :

```
> serie3<-c(T,T,F,F,T)
> serie3
```