

Traitements statistiques et programmation avec R

Gilles Hunault

Maître de conférences à l'université d'Angers

DUNOD

Conception de la couverture : Hokus Pokus

Illustration de couverture : © mk. / Fotolia

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. Le Code de la propriété intellectuelle du 1^{er} juillet 1992 interdit en effet expressément la photocopie à usage collectif sans autorisation des ayants droit. Or, cette pratique	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. 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).
	

© Dunod, Paris, 2017
11, rue Paul Bert, 92240 Malakoff
www.dunod.com
ISBN 978-2-10-076145-6

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

Remerciements	VII
Avant-propos	IX
Chapitre 1 Principes généraux de programmation	1
1. Qu'est-ce que la programmation R ?	1
2. Comment apprendre à programmer ?	2
3. Principes de programmation	3
4. Spécificités du langage R	6
Les points clés	7
Exercices	8
Chapitre 2 Affectations, structures de données et affichages	9
1. Variables simples et affectations	9
2. Autres exemples d'affectation	11
3. Structures de données et affectations en R	15
4. Autres « objets » en R	19
5. Affichage des variables en R	20
6. Spécificités du langage R	21
7. Si vous savez déjà programmer	25
Les points clés	28
Exercices	29
Chapitre 3 Conditions logiques et tests	31
1. Valeurs logiques et tests en R	31
2. Indexation et filtrage vectoriel	36
3. Autres exemples de tests et de filtrage vectoriel	37
4. Applications aux matrices et data frames	44
5. Spécificités du langage R	45

6. Si vous savez déjà programmer	48
Les points clés	50
Exercices	51
Chapitre 4 Boucles et itérations	53
1. Boucles <i>tant que</i>	53
2. Boucles <i>pour</i>	56
3. Boucles <i>répéter jusqu'à</i>	57
4. Imbrications, itérations et sorties de boucles	59
5. Autres exemples d'itérations et de boucles	61
6. Spécificités du langage R	63
7. Si vous savez déjà programmer	64
Les points clés	65
Exercices	66
Chapitre 5 Sous-programmes : les fonctions	67
1. Définition des fonctions nommées et anonymes	67
2. Définition et utilisation des paramètres	72
3. Tests des paramètres	74
4. L'ellipse notée ...	77
5. Autres exemples de fonctions	79
6. Spécificités du langage R	85
7. Si vous savez déjà programmer	86
Les points clés	89
Exercices	90
Chapitre 6 Éviter de programmer en R	95
1. Que faut-il vraiment programmer en R ?	95
2. À l'aide des fonctions classiques de R	98
3. À l'aide des fonctions *apply	101
4. À l'aide des packages	104
5. Spécificités du langage R	108

6. Si vous savez déjà programmer	109
Les points clés	111
Exercices	112
Chapitre 7 Programmation soutenue	113
1. Structures de données classiques des autres langages	113
2. Problèmes demandant des compétences techniques en algorithmique	114
3. Problèmes de grande taille (<i>big data volume</i>)	115
4. Problèmes de grande ampleur (<i>big data computing</i>)	115
5. Récursivité	116
6. Spécificités du langage R	118
7. Si vous savez déjà programmer	119
Les points clés	121
Exercices	122
Chapitre 8 Programmation et développement	125
1. Programmation, choix et tests	125
2. Programmation, spécifications et documentations	131
3. Mises à jour, maintenances, objets et packages	132
4. Programmation et interfaçage	132
5. Spécificités du langage R	134
6. Si vous savez déjà programmer	136
Les points clés	137
Exercices	138
Chapitre 9 Débogage, profilage et optimisation	141
1. Un comportement bizarre de R	142
2. Débogage	145
3. Écrire <code><-</code> n'est pas une faute de frappe	149
4. Profilage	151
5. Optimisation	157

6. Spécificités du langage R	159
7. Si vous savez déjà programmer	160
Les points clés	162
Exercices	163
Solutions des exercices	165
1. Solutions des exercices du chapitre 1	165
2. Solutions des exercices du chapitre 2	168
3. Solutions des exercices du chapitre 3	179
4. Solutions des exercices du chapitre 4	187
5. Solutions des exercices du chapitre 5	192
6. Solutions des exercices du chapitre 6	208
7. Solutions des exercices du chapitre 7	221
8. Solutions des exercices du chapitre 8	231
9. Solutions des exercices du chapitre 9	247
Conclusion	267
Bibliographie	268
Packages cités et conseillés	269
Sites Web cités et conseillés	270
Index des fonctions et des packages	271

À Paul Calés, Laurence Brunet et Timothée Hamon.
À Séverin et Aude.

Remerciements

L’auteur tient à remercier Benoit Da Mota pour sa relecture attentive de l’ouvrage et ses conseils avisés quant aux rapports entre la programmation R et la programmation en général. Il faudrait aussi remercier toutes les étudiantes et les étudiants, en formation initiale comme en formation continue, pour leurs nombreuses questions, même les plus naïves, parce qu’elles ont permis de cerner les difficultés de comprendre ce qu’est la découverte et l’apprentissage initial de la programmation. Qu’ils et elles soient aussi remerciés pour avoir supporté mes blagues parfois trop nombreuses lorsque je les trouvais trop fatigués par la lecture de mes codes R !

Avant-propos

Pourquoi lire ce livre sur R ?

La plupart des personnes qui programment en R programment mal en R. Il y a deux causes à cela :

- beaucoup d'utilisateurs de R commencent par écrire quelques commandes ou des petits scripts et du coup ils apprennent la programmation R sur le tas sans savoir ce que cela représente vraiment de programmer vectoriellement ;
- lorsqu'on sait déjà programmer dans un autre langage, sauf peut-être avec le langage APL, on ne sait pas programmer vectoriellement.

De ce fait, la plupart des programmes R qui sont écrits sont inefficaces, surchargés de boucles lentes, de code long et pas toujours facile à maintenir. En conséquence, le but de cet ouvrage est double :

1. aider celles et ceux qui ne savent pas programmer du tout à bien débiter la programmation et en particulier la programmation vectorielle ;
2. fournir à celles et ceux qui savent déjà programmer les moyens de comprendre comment s'adapter à la programmation vectorielle en R.

Il y a d'autres particularités dans la programmation R : les packages de base regorgent de fonctions qui dispensent de reprogrammer des actions usuelles, les autres packages contiennent tout ce qu'il faut pour réaliser efficacement des analyses statistiques et des représentations graphiques de tout ordre et de toute nature, la définition des fonctions avec leurs paramètres nommés et l'ellipse fournit des moyens souples pour programmer, la gestion de la mémoire est spécifique aux langages vectoriels interprétés etc.

Ne pas savoir tout cela rend la programmation R parfois déroutante ou « bizarre » au premier abord et il n'est pas étonnant d'être perplexe devant toutes les structures de données et les fonctions associées, devant la difficulté de comprendre comment optimiser du code R... et avec les données dont on peut disposer aujourd'hui, ces fameux *Big Data*, il est important de savoir optimiser du code R.

À l'inverse, savoir tout cela rend la programmation R plus concise, plus efficace, même si parfois le code est – un peu – moins lisible qu'avec la programmation itérative classique.

Ce que contient cet ouvrage

Les cinq chapitres qui suivent sont toujours structurés de la même manière : d'abord le but du chapitre, puis des notions algorithmiques et leur traduction dans le cadre de la

programmation avec de nombreux exemples puis les spécificités de R liées à ces notions et enfin des remarques pour celles et ceux qui savent déjà programmer.

Une programmeuse, un programmeur classique pourra ainsi survoler les notions (mais pas trop vite, parce que R est « spécial ») pour se concentrer sur les spécificités et sur ces remarques, de façon à pouvoir désapprendre ce qui était considéré comme des bonnes pratiques dans un autre langage et qui se révèle souvent inadapté ou inefficace en R.

Pour les débutantes et les débutants, cela permettra ainsi de séparer ce qui tient de la programmation classique de ce qui ne ressort que de la programmation vectorielle, fournissant ainsi la possibilité d'utiliser plus tard d'autres langages de programmation.

L'enchaînement des chapitres est très traditionnel, même si le chapitre 1 ne fait pas toujours partie des manuels classiques de programmation. Dans ce chapitre figurent les principes généraux de programmation avec une insistance particulière sur la rigueur et l'effort soutenu que demande la programmation. Le chapitre 2 présente les affectations, les structures de données et explique comment réaliser de « beaux » affichages. Le cœur du chapitre 3 est constitué par les conditions logiques et les tests, complété au chapitre 4 par les boucles et itérations. Ensuite le chapitre 5 détaille la définition des fonctions et la notion de script. Les bases de la programmation sont alors posées.

Au chapitre 6, le lecteur apprendra comment éviter de programmer des actions usuelles, comment se passer des boucles « POUR » dans pratiquement tous les cas. Le but du chapitre 7 qui est intitulé « Programmation soutenue » est de faire passer le lecteur du statut de programmeur débutant à celui de programmeur averti, *via* des exemples plus approfondis alors qu'au chapitre 8 on viendra détailler les nombreuses différences entre programmation et développement. Enfin, le chapitre 9 fournira quelques repères sur le débogage et l'optimisation d'applications écrites en R. Les solutions détaillées des exercices sont en fin d'ouvrage, juste avant la conclusion et les annexes (liste de livres, packages et sites conseillés).

Les exercices sont une partie essentielle de ce livre. C'est pourquoi ils représentent avec leurs corrigés pratiquement la moitié de l'ouvrage. Il est important de les résoudre, ou tout au moins d'essayer de les résoudre pour vérifier si les concepts ont été compris. Il y a souvent en programmation plusieurs façons de faire. Nous avons essayé dans les solutions de fournir des exemples de code lisible, robuste et portable.

Les solutions des exercices sont souvent très détaillées de façon à bien montrer comment on trouve une première solution, comment on l'améliore pour obtenir une solution complète et générale car il est rare qu'en programmation on pense à tout dès le premier jet. Il nous semble important de faire passer ce message qu'un programme ne s'écrit pas d'une seule traite et en une seule fois.

À défaut de résoudre explicitement tous les exercices, il est conseillé de lire au moins les solutions, pour profiter de la progression pédagogique (les exercices sont la plupart du temps de difficulté croissante pour chaque chapitre) et pour voir, souvent après des essais corrects mais incomplets ou trop « itératifs », des « vraies » solutions R.

Ce qu'on ne trouvera pas dans cet ouvrage

Pour que cet ouvrage reste d'une taille raisonnable et joue son rôle d'initiation et de sensibilisation à la programmation vectorielle, nous avons décidé de ne présenter ni la création de classes d'objets, ni la réalisation de packages sauf sous forme de mini-exemples dans les exercices du chapitre 8. Car si les concepts sous-jacents sont relativement simples, leur exécution en pratique requiert une technicité qui demande une forte expérience de la programmation. De plus R utilise trois systèmes de classe d'objets (objets R5, S3 et S4), ce qui complique leur présentation.

On ne trouvera pas non plus beaucoup de calculs statistiques compliqués ou de réalisations graphiques sophistiquées car la plupart des autres livres sur R traitent ces sujets. Nous utiliserons de temps en temps des calculs statistiques simples comme la moyenne, la médiane, nous ferons de temps en temps références à des tracés graphiques comme les *boxplots* (ou *boîtes à moustaches* en français), mais aucune compétence particulière en statistique ou en mathématique n'est requise pour lire cet ouvrage.

L'idée centrale pour comprendre ce que contient ce livre, pour reprendre une analogie classique, consiste à dire que nous voulons apprendre au lecteur comment tenir sur un vélo, ou sur des skis, comment prendre de la vitesse, tenir la distance, mais pas comment faire le tour de France ou participer à une épreuve des jeux olympiques. Après tout, il ne s'agit que d'une introduction à la programmation R.

Prérequis

Pour profiter pleinement de cet ouvrage, il y a quelques prérequis. Tout d'abord il faut disposer de temps libre et être reposé(e) car la programmation est une activité intellectuelle qui demande de la concentration.

Ensuite il faut de la patience et accepter de travailler de temps en temps avec un papier et un crayon pour réfléchir aux solutions possibles avant d'utiliser l'ordinateur. Programmer demande du recul et de l'expérience, il ne faut pas être dans le feu de l'action et du Web si on veut programmer sereinement.

Il faut aussi avoir envie plutôt qu'avoir besoin de programmer. Lire cet ouvrage en dilettante ne vous apprendra pas grand chose ou ne vous sera pas très utile par ce que vous n'aurez rien intériorisé. Si vous êtes forcé(e) de programmer, vous programmerez mal par ce que vous ne mettrez pas assez de *cœur à l'ouvrage*.

Enfin, bien sûr, comme cet ouvrage n'est pas une initiation au logiciel R, il faut connaître *un peu* R et avoir déjà utilisé R. Par exemple, n'avoir jamais utilisé au moins une fois les fonctions **read.table()**, **cat()**, **paste()** ou **names()** signifie sans doute que vous n'avez pas assez manipulé R en ligne de commande et vous aurez du mal à suivre les exemples utilisés, non pas à cause de la programmation mais à cause des fonctions de R utilisées.

Donc si vous n'avez pas beaucoup de pratique de R ou si vous l'avez utilisé seulement *via* une interface comme Rcmdr (« R commandeur »), Rattle ou rkward, il vaut mieux commencer par vous entraîner à écrire des calculs simples en R et réaliser de petits scripts en ligne de commande avant de vous « attaquer » à cet ouvrage.

Il existe de nombreux cours et pages d'initiation à R sur Internet, sans compter les ouvrages en français qui traitent de R. Vous trouverez quelques titres d'ouvrages que nous recommandons dans la bibliographie située dans l'annexe en fin de ce livre et qui peuvent vous servir de compléments – et non pas de préambule – à cet ouvrage.

Nous avons décidé de rédiger ce livre car, malheureusement, très peu d'ouvrages généraux présentent R comme un « vrai » langage de programmation et encore moins exploitent explicitement le fait que R est vectoriel. Ils se contentent souvent d'une présentation rapide de la syntaxe de R, pas de sa « philosophie ». Vous verrez souvent cette partie de phrase dans cet ouvrage : « *Comme R est vectoriel...* » car c'est un point clé de la programmation en R. Et lorsque vos collègues vous demanderont pourquoi vos programmes à vous sont si courts à écrire et si rapides à exécuter, vous pourrez leur répondre : « *C'est parce que R est vectoriel !* ».

Vous trouverez également en annexe les différents packages que nous considérons comme importants pour R sous l'angle de la programmation et une liste de quelques sites Web utiles pour bien progresser en R et se tenir au courant des évolutions de R.

Nous avons bien conscience qu'on n'apprend pas l'anglais, l'allemand ou le chinois en vingt-quatre heures. On n'apprend pas non plus à programmer en quelques dizaines d'heures juste en lisant des scripts ou des solutions d'exercices ou en « bricolant » quelques petits programmes. C'est à force de pratique que l'on apprend, parfois en commettant des erreurs, ou en lisant les erreurs des autres et rien ne remplace donc vos essais personnels de programmation. De plus il faut souvent plusieurs essais, plusieurs versions progressives avant que le script ne soit complet, testé, aussi bien dans les bons cas que dans la gestion des erreurs les plus courantes, avec au passage quelques idées sur le « profil » du code, sa rapidité, sans oublier sa documentation.

Pour conclure cette introduction, je vous souhaite autant de plaisir que possible à lire ce livre et à programmer. Un programme est un petit peu une œuvre d'art, alors soyez artistes, soyez heureuses et soyez heureux.

À la découverte de votre livre

1 Ouverture de chapitre

Elle donne :

- une introduction aux sujets et aux problématiques abordés dans le chapitre
- un rappel des objectifs pédagogiques
- le plan du chapitre

Chapitre 1 Principes généraux de programmation

Introduction

Avant de créer dans le détail du code R, nous allons explorer ce qu'est le langage R et comment il est utilisé pour la programmation. Nous verrons également comment le langage R est utilisé pour la programmation et comment il est utilisé pour la programmation.

Objectifs

- Comprendre les principes généraux de la programmation en R.
- Comprendre les objectifs pédagogiques du chapitre.
- Comprendre les objectifs pédagogiques du chapitre.

Plan

- Introduction aux sujets et aux problématiques abordés dans le chapitre.
- Un rappel des objectifs pédagogiques
- Le plan du chapitre

1. Qu'est-ce que la programmation R ?

Préparer un programme en langage R, c'est écrire un programme qui va exécuter des commandes. Ces commandes sont des instructions qui vont être exécutées par le langage R. Le langage R est un langage de programmation qui permet de créer des programmes qui vont exécuter des commandes. Ces commandes sont des instructions qui vont être exécutées par le langage R. Le langage R est un langage de programmation qui permet de créer des programmes qui vont exécuter des commandes. Ces commandes sont des instructions qui vont être exécutées par le langage R.

2 Le cours

Un cours concis et structuré.

3 En fin de chapitre

- L'essentiel : les points clés pour réviser les connaissances essentielles
- Des exercices pour tester ses connaissances

4 En fin d'ouvrage

- Les solutions des exercices
- Une bibliographie
- Les packages cités et conseillés
- Les sites web cités et conseillés
- Un index des fonctions et des packages

5 Ressources numériques

Retrouvez sur www.dunod.com l'ensemble des lignes de code des exercices.



