

Table des matières

Introduction	i
1 Présentation du langage R	1
1.1 Bref historique	1
1.2 Description sommaire de R	2
1.3 Interfaces	3
1.4 Stratégies de travail	4
1.5 Éditeurs de texte	5
1.6 Anatomie d'une session de travail	8
1.7 Répertoire de travail	9
1.8 Consulter l'aide en ligne	9
1.9 Où trouver de la documentation	9
1.10 Exemples	10
1.11 Exercices	11
2 Bases du langage R	13
2.1 Commandes R	13
2.2 Conventions pour les noms d'objets	15
2.3 Les objets R	16
2.4 Vecteurs	20
2.5 Matrices et tableaux	21
2.6 Listes	24
2.7 <i>Data frames</i>	26
2.8 Indixage	26
2.9 Exemples	28
2.10 Exercices	39
3 Opérateurs et fonctions	41
3.1 Opérations arithmétiques	41

3.2	Opérateurs	42
3.3	Appels de fonctions	43
3.4	Quelques fonctions utiles	44
3.5	Structures de contrôle	50
3.6	Fonctions additionnelles	51
3.7	Exemples	52
3.8	Exercices	60
4	Exemples résolus	63
4.1	Calcul de valeurs présentes	63
4.2	Fonctions de probabilité	64
4.3	Fonction de répartition de la loi gamma	66
4.4	Algorithme du point fixe	68
4.5	Exercices	69
A	GNU Emacs et ESS : la base	63
A.1	Mise en contexte	63
A.2	Installation	64
A.3	Description sommaire	64
A.4	<i>Emacs-ismes</i> et <i>Unix-ismes</i>	65
A.5	Commandes de base	66
A.6	Anatomie d'une session de travail (bis)	69
A.7	Configuration de l'éditeur	70
A.8	Aide et documentation	70
B	GNU Free Documentation License	71
B.1	APPLICABILITY AND DEFINITIONS	71
B.2	VERBATIM COPYING	73
B.3	COPYING IN QUANTITY	74
B.4	MODIFICATIONS	74
B.5	COMBINING DOCUMENTS	76
B.6	COLLECTIONS OF DOCUMENTS	77
B.7	AGGREGATION WITH INDEPENDENT WORKS	77
B.8	TRANSLATION	78
B.9	TERMINATION	78
B.10	FUTURE REVISIONS OF THIS LICENSE	78
	ADDENDUM: How to use this License for your documents	79
	Bibliographie	81

4 Exemples résolus

Ce chapitre propose de faire le point sur les concepts étudiés jusqu'à maintenant par le biais de quelques exemples résolus. On y met particulièrement en évidence les avantages de l'approche vectorielle du langage R.

La compréhension du contexte de ces exemples requiert quelques connaissances de base en mathématiques financières et en théorie des probabilités.

4.1 Calcul de valeurs présentes

De manière générale, la valeur présente d'une série de paiements P_1, P_2, \dots, P_n à la fin des années $1, 2, \dots, n$ est

$$\sum_{j=1}^n \prod_{k=1}^j (1 + i_k)^{-1} P_j, \quad (4.1)$$

où i_k est le taux d'intérêt effectif annuellement durant l'année k . Lorsque le taux d'intérêt est constant au cours des n années, cette formule se simplifie en

$$\sum_{j=1}^n (1 + i)^{-j} P_j. \quad (4.2)$$

Un prêt est remboursé par une série de cinq paiements, le premier étant dû dans un an. On doit trouver le montant du prêt pour chacune des hypothèses ci-dessous.

- a) Paiement annuel de 1 000, taux d'intérêt de 6 % effectif annuellement.

Avec un paiement annuel et un taux d'intérêt constants, on utilise la formule (4.2) avec $P_j = P = 1\,000$:

```
> 1000 * sum((1 + 0.06)^(-(1:5)))  
[1] 4212.364
```

Remarquer comme l'expression R se lit exactement comme la formule mathématique.

- b) Paiements annuels de 500, 800, 900, 750 et 1 000, taux d'intérêt de 6 % effectif annuellement.

Les paiements annuels sont différents, mais le taux d'intérêt est toujours le même. La formule (4.2) s'applique donc directement :

```
> sum(c(500, 800, 900, 750, 1000) * (1 + 0.06)^(-(1:5)))
[1] 3280.681
```

- c) Paiements annuels de 500, 800, 900, 750 et 1 000, taux d'intérêt de 5 %, 6 %, 5,5 %, 6,5 % et 7 % effectifs annuellement.

Avec différents paiements annuels et des taux d'intérêt différents, il faut employer la formule (4.1). Le produit cumulatif des taux d'intérêt est obtenu avec la fonction `cumprod` :

```
> sum(c(500, 800, 900, 750, 1000) /
+     cumprod(1 + c(0.05, 0.06, 0.055, 0.065, 0.07)))
[1] 3308.521
```

4.2 Fonctions de probabilité

On doit calculer toutes ou la majeure partie des probabilités de deux lois de probabilité, puis vérifier que la somme des probabilités est bien égale à 1.

Cet exemple est quelque peu artificiel dans la mesure où il existe dans R des fonctions internes pour calculer les principales caractéristiques des lois de probabilité les plus usuelles. Nous utiliserons d'ailleurs ces fonctions pour vérifier nos calculs.

- a) Calculer toutes les masses de probabilité de la distribution binomiale pour des valeurs des paramètres n et p quelconques. La fonction de masse de probabilité de la binomiale est

$$f(x) = \binom{n}{x} p^x (1-p)^{n-x}, \quad x = 0, \dots, n.$$

Soit $n = 10$ et $p = 0,8$. Les coefficients binomiaux sont calculés avec la fonction `choose` :

```
> n <- 10
> p <- 0.8
> x <- 0:n
> choose(n, x) * p^x * (1 - p)^rev(x)
```

```
[1] 0.0000001024 0.0000040960 0.0000737280
[4] 0.0007864320 0.0055050240 0.0264241152
[7] 0.0880803840 0.2013265920 0.3019898880
[10] 0.2684354560 0.1073741824
```

On vérifie les réponses obtenues avec la fonction interne dbinom :

```
> dbinom(x, n, prob = 0.8)
[1] 0.0000001024 0.0000040960 0.0000737280
[4] 0.0007864320 0.0055050240 0.0264241152
[7] 0.0880803840 0.2013265920 0.3019898880
[10] 0.2684354560 0.1073741824
```

On vérifie enfin que les probabilités somment à 1 :

```
> sum(choose(n, x) * p^x * (1 - p)^rev(x))
[1] 1
```

- b) Calculer la majeure partie des masses de probabilité de la distribution de Poisson, dont la fonction de masse de probabilité est

$$f(x) = \frac{\lambda^x e^{-\lambda}}{x!}, \quad x = 0, 1, \dots,$$

où $x! = x(x-1)\cdots 2 \cdot 1$.

La loi de Poisson ayant un support infini, on calcule les probabilités en $x = 0, 1, \dots, 10$ seulement avec $\lambda = 5$. On calcule les factorielles avec la fonction `factorial`. On notera au passage que `factorial(x) == gamma(x + 1)`, où la fonction R `gamma` calcule les valeurs de la fonction mathématique du même nom

$$\Gamma(n) = \int_0^\infty x^{n-1} e^{-x} dx = (n-1)\Gamma(n-1),$$

avec $\Gamma(0) = 1$. Pour n entier, on a donc $\Gamma(n) = (n-1)!$.

```
> lambda <- 5
> x <- 0:10
> exp(-lambda) * (lambda^x / factorial(x))
[1] 0.006737947 0.033689735 0.084224337
[4] 0.140373896 0.175467370 0.175467370
[7] 0.146222808 0.104444863 0.065278039
[10] 0.036265577 0.018132789
```

Vérification avec la fonction interne dpois :

```
> dpois(x, lambda)
[1] 0.006737947 0.033689735 0.084224337
[4] 0.140373896 0.175467370 0.175467370
[7] 0.146222808 0.104444863 0.065278039
[10] 0.036265577 0.018132789
```

Pour vérifier que les probabilités somment à 1, il faudra d'abord tronquer le support infini de la Poisson à une «grande» valeur. Ici, 200 est suffisamment éloigné de la moyenne de la distribution, 5. Remarquer que le produit par $e^{-\lambda}$ est placé à l'extérieur de la somme pour ainsi faire un seul produit plutôt que 201.

```
> x <- 0:200
> exp(-lambda) * sum((lambda^x / factorial(x)))
[1] 1
```

4.3 Fonction de répartition de la loi gamma

La loi gamma est fréquemment utilisée pour la modélisation d'événements ne pouvant prendre que des valeurs positives et pour lesquels les petites valeurs sont plus fréquentes que les grandes. Par exemple, on utilise parfois la loi gamma en sciences actuarielles pour la modélisation des montants de sinistres. Nous utilisons la paramétrisation où la fonction de densité de probabilité est

$$f(x) = \frac{\lambda^\alpha}{\Gamma(\alpha)} x^{\alpha-1} e^{-\lambda x}, \quad x > 0, \quad (4.3)$$

où $\Gamma(\cdot)$ est la fonction gamma définie dans l'exemple précédent.

Il n'existe pas de formule explicite de la fonction de répartition de la loi gamma. Néanmoins, la valeur de la fonction de répartition d'une loi gamma de paramètre α entier et $\lambda = 1$ peut être obtenue à partir de la formule

$$F(x; \alpha, 1) = 1 - e^{-x} \sum_{j=0}^{\alpha-1} \frac{x^j}{j!}. \quad (4.4)$$

a) Évaluer $F(4; 5, 1)$.

Cet exercice est simple puisqu'il s'agit de calculer une seule valeur de la fonction de répartition avec un paramètre α fixe. Par une application directe de (4.4), on a :


```
> alpha <- 5
> x <- 4
> 1 - exp(-x) * sum(x^(0:(alpha - 1))/gamma(1:alpha))
[1] 0.3711631
```

Vérification avec la fonction interne pgamma :

```
> pgamma(x, alpha)
[1] 0.3711631
```

On peut aussi éviter de générer essentiellement la même suite de nombres à deux reprises en ayant recours à une variable intermédiaire. Au risque de rendre le code un peu moins lisible (mais plus compact!), l'affectation et le calcul final peuvent même se faire dans une seule expression.

```
> 1 - exp(-x) * sum(x^(-1 + (j <- 1:alpha))/gamma(j))
[1] 0.3711631
```

- b) Évaluer $F(x; 5, 1)$ pour $x = 2, 3, \dots, 10$ en une seule expression.

Cet exercice est beaucoup plus compliqué qu'il n'y paraît au premier abord. Ici, la valeur de α demeure fixe, mais on doit calculer, en une seule expression, la valeur de la fonction de répartition en plusieurs points. Or, cela exige de faire d'un coup le calcul x^j pour plusieurs valeurs de x et plusieurs valeurs de j . C'est un travail pour la fonction `outer` :

```
> x <- 2:10
> 1 - exp(-x) * colSums(t(outer(x, 0:(alpha-1), "^")) /
+                        gamma(1:alpha))
[1] 0.05265302 0.18473676 0.37116306 0.55950671
[5] 0.71494350 0.82700839 0.90036760 0.94503636
[9] 0.97074731
```

Vérification avec la fonction interne pgamma :

```
> pgamma(x, alpha)
[1] 0.05265302 0.18473676 0.37116306 0.55950671
[5] 0.71494350 0.82700839 0.90036760 0.94503636
[9] 0.97074731
```

Il est laissé en exercice de déterminer pourquoi la transposée est nécessaire dans l'expression ci-dessus. Exécuter l'expression étape par étape, de l'intérieur vers l'extérieur, pour mieux comprendre comment on arrive à faire le calcul en (4.4).

4.4 Algorithme du point fixe

Trouver la racine d'une fonction g — c'est-à-dire le point x où $g(x) = 0$ — est un problème classique en mathématiques. Très souvent, il est possible de reformuler le problème de façon à plutôt chercher le point x où $f(x) = x$. La solution d'un tel problème est appelée *point fixe*.

L'algorithme du calcul numérique du point fixe d'une fonction $f(x)$ est très simple :

1. choisir une valeur de départ x_0 ;
2. calculer $x_n = f(x_{n-1})$ pour $n = 1, 2, \dots$;
3. répéter l'étape 2 jusqu'à ce que $|x_n - x_{n-1}| < \epsilon$ ou $|x_n - x_{n-1}|/|x_{n-1}| < \epsilon$.

On doit trouver, à l'aide de la méthode du point fixe, la valeur de i telle que

$$a_{\overline{10}} = \frac{1 - (1 + i)^{-10}}{i} = 8,21,$$

c'est à dire le taux de rendement d'une série de 10 versements de 1 pour laquelle on a payé un montant de 8,21.

Puisque, d'une part, nous ignorons combien de fois la procédure itérative devra être répétée et que, d'autre part, il faut exécuter la procédure au moins une fois, le choix logique pour la structure de contrôle à utiliser dans cette procédure itérative est `repeat`. De plus, il faut comparer deux valeurs successives du taux d'intérêt, nous devons donc avoir recours à deux variables. On a :

```
> i <- 0.05
> repeat
+ {
+   it <- i
+   i <- (1 - (1 + it)^(-10))/8.21
+   if (abs(i - it)/it < 1E-10)
+     break
+ }
> i
[1] 0.03756777
```

Vérification :

```
> (1 - (1 + i)^(-10))/i
[1] 8.21
```

Nous verrons au chapitre ?? comment créer une fonction à partir de ce code.

4.5 Exercices

Dans chacun des exercices ci-dessous, écrire une expression R pour faire le calcul demandé. Parce qu'elles ne sont pas nécessaires, il est interdit d'utiliser des boucles.

4.1 Calculer la valeur présente d'une série de paiements fournie dans un vecteur P en utilisant les taux d'intérêt annuels d'un vecteur i .

4.2 Étant donné un vecteur d'observations $\mathbf{x} = (x_1, \dots, x_n)$ et un vecteur de poids correspondants $\mathbf{w} = (w_1, \dots, w_n)$, calculer la moyenne pondérée des observations,

$$\sum_{i=1}^n \frac{w_i}{w_\Sigma} x_i,$$

où $w_\Sigma = \sum_{i=1}^n w_i$. Tester l'expression avec les vecteurs de données

$$\mathbf{x} = (7, 13, 3, 8, 12, 12, 20, 11)$$

et

$$\mathbf{w} = (0,15, 0,04, 0,05, 0,06, 0,17, 0,16, 0,11, 0,09).$$

4.3 Soit un vecteur d'observations $\mathbf{x} = (x_1, \dots, x_n)$. Calculer la moyenne harmonique de ce vecteur, définie comme

$$\frac{n}{\frac{1}{x_1} + \dots + \frac{1}{x_n}}.$$

Tester l'expression avec les valeurs de l'exercice 4.2.

4.4 Calculer la fonction de répartition en $x = 5$ d'une loi de Poisson avec paramètre $\lambda = 2$, qui est donnée par

$$\sum_{k=0}^5 \frac{2^k e^{-2}}{k!},$$

où $k! = 1 \cdot 2 \cdots k$.

4.5 a) Calculer l'espérance d'une variable aléatoire X dont le support est $x = 1, 10, 100, \dots, 1\,000\,000$ et les probabilités correspondantes sont $\frac{1}{28}, \frac{2}{28}, \dots, \frac{7}{28}$, dans l'ordre.

b) Calculer la variance de la variable aléatoire X définie en a).

- 4.6** Calculer le taux d'intérêt nominal composé quatre fois par année, $i^{(4)}$, équivalent à un taux de $i = 6\%$ effectif annuellement.
- 4.7** La valeur présente d'une série de n paiements de fin d'année à un taux d'intérêt i effectif annuellement est

$$a_{\overline{n}|} = v + v^2 + \cdots + v^n = \frac{1 - v^n}{i},$$

où $v = (1 + i)^{-1}$. Calculer en une seule expression, toujours sans boucle, un tableau des valeurs présentes de séries de $n = 1, 2, \dots, 10$ paiements à chacun des taux d'intérêt effectifs annuellement $i = 0,05, 0,06, \dots, 0,10$.

- 4.8** Calculer la valeur présente d'une annuité croissante de 1 \$ payable annuellement en début d'année pendant 10 ans si le taux d'actualisation est de 6 %. Cette valeur présente est donnée par

$$I\ddot{a}_{\overline{10}|} = \sum_{k=1}^{10} k v^{k-1},$$

toujours avec $v = (1 + i)^{-1}$.

- 4.9** Calculer la valeur présente de la séquence de paiements 1, 2, 2, 3, 3, 3, 4, 4, 4, 4 si les paiements sont effectués en fin d'année et que le taux d'actualisation est de 7 %.
- 4.10** Calculer la valeur présente de la séquence de paiements définie à l'exercice 4.9 en supposant que le taux d'intérêt d'actualisation alterne successivement entre 5 % et 8 % chaque année, c'est-à-dire que le taux d'intérêt est de 5 %, 8 %, 5 %, 8 %, etc.

Bibliographie

- Abelson, H., G. J. Sussman et J. Sussman. 1996, *Structure and Interpretation of Computer Programs*, 2^e éd., MIT Press, ISBN 0-26201153-0.
- Becker, R. A. 1994, «A brief history of S», cahier de recherche, AT&T Bell Laboratories. URL <http://cm.bell-labs.com/cm/ms/departments/sia/doc/94.11.ps>.
- Becker, R. A. et J. M. Chambers. 1984, *S: An Interactive Environment for Data Analysis and Graphics*, Wadsworth, ISBN 0-53403313-X.
- Becker, R. A., J. M. Chambers et A. R. Wilks. 1988, *The New S Language: A Programming Environment for Data Analysis and Graphics*, Wadsworth & Brooks/Cole, ISBN 0-53409192-X.
- Braun, W. J. et D. J. Murdoch. 2007, *A First Course in Statistical Programming with R*, Cambridge University Press, ISBN 978-0-52169424-7.
- Cameron, D., J. Elliott, M. Loy, E. S. Raymond et B. Rosenblatt. 2004, *Leaning GNU Emacs*, 3^e éd., O'Reilly, Sebastopol, CA, ISBN 0-59600648-9.
- Chambers, J. M. 1998, *Programming with Data: A Guide to the S Language*, Springer, ISBN 0-38798503-4.
- Chambers, J. M. 2000, «Stages in the evolution of S», URL <http://cm.bell-labs.com/cm/ms/departments/sia/S/history.html>.
- Chambers, J. M. 2008, *Software for Data Analysis: Programming with R*, Springer, ISBN 978-0-38775935-7.
- Chambers, J. M. et T. J. Hastie. 1992, *Statistical Models in S*, Wadsworth & Brooks/Cole, ISBN 0-53416765-9.
- Hornik, K. 2011, «The R FAQ», URL <http://cran.r-project.org/doc/FAQ/R-FAQ.html>, ISBN 3-90005108-9.

- Iacus, S. M., S. Urbanek et R. J. Goedman. 2011, «R for Mac OS X FAQ», URL <http://cran.r-project.org/bin/macosx/RMacOSX-FAQ.html>.
- IEEE. 2003, *754-1985 IEEE Standard for Binary Floating-Point Arithmetic*, IEEE, Piscataway, NJ.
- Ihaka, R. et R. Gentleman. 1996, «R: A language for data analysis and graphics», *Journal of Computational and Graphical Statistics*, vol. 5, n° 3, p. 299–314.
- Ligges, U. 2003, «R-winedt», dans *Proceedings of the 3rd International Workshop on Distributed Statistical Computing (DSC 2003)*, édité par K. Hornik, F. Leisch et A. Zeileis, TU Wien, Vienna, Austria, ISSN 1609-395X. URL <http://www.ci.tuwien.ac.at/Conferences/DSC-2003/Proceedings/>.
- Redd, A. 2010, «Introducing NppToR: R interaction for Notepad++», *R Journal*, vol. 2, n° 1, p. 62–63. URL http://journal.r-project.org/archive/2010-1/RJournal_2010-1.pdf.
- Ripley, B. D. et D. J. Murdoch. 2011, «R for Windows FAQ», URL <http://cran.r-project.org/bin/windows/base/rw-FAQ.html>.
- Venables, W. N. et B. D. Ripley. 2000, *S Programming*, Springer, New York, ISBN 0-38798966-8.
- Venables, W. N. et B. D. Ripley. 2002, *Modern Applied Statistics with S*, 4^e éd., Springer, New York, ISBN 0-38795457-0.
- Venables, W. N., D. M. Smith et R Development Core Team. 2011, *An Introduction to R*, R Foundation for Statistical Computing. URL <http://cran.r-project.org/doc/manuals/R-intro.html>.

Index

Les numéros de page en caractères gras indiquent les pages où les concepts sont introduits, définis ou expliqués.

!, **43**
!=, **43**
*, **43**
+, **43**
-, **43**
->, **14**, **43**
-Inf, **19**
/, **43**
:, **43**, **60**
;, **14**
<, **43**
<-, **14**, **42**, **43**
<=, **43**
=, **14**
==, **43**
>, **43**
>=, **43**
[, **27**
[<-, **27**
[[]], **25**, **25**
[], **21**, **23**, **25**, **27**
\$, **26**, **27**, **43**
\$<-, **27**
%*%, **43**
%/%, **43**
%%, **43**
%in%, **46**, **56**
%0%, **50**, **58**
&, **43**
&&, **43**
^, **42**, **43**
{ }, **15**

affectation, **13**
apply, **51**, **58**, **61**
array, **22**, **34**
array (classe), **22**
arrondi, **46**
as.data.frame, **26**
attach, **26**, **37**
attr, **19**, **31**
attribut, **19**
attributes, **19**, **31**, **32**

boucle, **51**, **69**
break, **51**, **60**
by, **10**, **55**
byrow, **22**, **44**

c, **20**
cbind, **24**, **26**, **34**, **39**, **57**
ceiling, **46**, **56**
character, **21**, **33**
character (mode), **17**, **21**
choose, **64**

- class, 32–34, 36
- class (attribut), **20**
- colMeans, **49**, 61
- colSums, **48**, 58, 61, 67
- compilé (langage), 2
- complex, 33
- complex (mode), **17**
- cos, 28
- cummax, **48**, 57
- cummin, **48**, 57
- cumprod, **48**, 57, 64
- cumsum, **48**, 57
- data, 31, 44, 55
- data frame, **26**
- data.frame, **26**
- data.frame (classe), 26
- dbinom, 65
- density, 10
- det, **49**
- detach, **26**, 37
- diag, 39, **49**, 57
- diff, **47**, 57
- différences, 47
- dim, 32–34, 36, 38, 57
- dim (attribut), **20**, 21, 22
- dimension, 20, 39
- dimnames, 32, 44
- dimnames (attribut), **20**
- distribution
 - binomiale, 64
 - gamma, 66
 - Poisson, 65, 69
- dossier de travail, voir répertoire de travail
- dpois, 65
- écart type, 47
- else, **50**, 58–60
- Emacs, 7
- C-_, 67
- C-g, 67
- C-r, 67
- C-s, 67
- C-SPC, 67
- C-w, 67
- C-x 0, 68
- C-x 1, 68
- C-x 2, 68
- C-x b, 68
- C-x C-f, 67
- C-x C-s, 67, 70
- C-x C-w, 67
- C-x k, 67
- C-x o, 68
- C-x o , 69
- C-x u, 67
- C-y, 67
- configuration, 70
- M-%, 67
- M-w, 67
- M-x, 67
- M-y, 67
- nouveau fichier, 67
- rechercher et remplacer, 67
- sélection, 67
- sauvegarder, 67
- sauvegarder sous, 67
- ESS, 7
 - C-c C-e, 68
 - C-c C-e , 69
 - C-c C-f, 68
 - C-c C-l, 68
 - C-c C-n, 68
 - C-c C-n , 69
 - C-c C-o, 68
 - C-c C-q, 68, 70
 - C-c C-r, 68
 - C-c C-v, 68
 - h, 68

- l, 69
- M-h, 68
- M-n, 68
- M-p, 68
- n, 68
- p, 68
- q, 69
- r, 69
- x, 69
- étiquette, 20, 39
- exists, 37
- exp, 28, 65, 66
- expression, 13
- expression, 30
- expression (mode), 17
- extraction, voir aussi `indicateur`
 - derniers éléments, 45
 - éléments différents, 45
 - premiers éléments, 45
- F, voir FALSE
- factorial, 61, 65
- FALSE, 16
- floor, 46, 56
- fonction
 - appel, 43
- for, 51, 53, 54, 58, 59
- function (mode), 17
- gamma, 28, 61, 65
- head, 45, 56
- if, 50, 53, 54, 58–60
- ifelse, 50
- indicateur
 - liste, 25, 39
 - matrice, 23, 26, 40
 - vecteur, 26, 39
- Inf, 19
- install.packages, 52
- interprété (langage), 2
- is.finite, 19
- is.infinite, 19
- is.na, 19, 31, 38, 59
- is.nan, 19
- is.null, 18
- lapply, 51
- length, 10, 17, 30–36, 38, 55–57
- lfactorial, 61
- lgamma, 61
- library, 52, 60
- list, 25, 30, 32, 35, 36
- list (mode), 17, 24
- liste, 24
- logical, 21, 33
- logical (mode), 17, 19, 21
- longueur, 18, 39
- ls, 11, 29
- mapply, 51
- match, 46, 56
- matrice, 61
 - diagonale, 49
 - identité, 49
 - inverse, 49
 - moyennes par colonne, 49
 - moyennes par ligne, 49
 - somme par colonne, 48
 - sommes par ligne, 48
 - transposée, 49
- matrix, 11, 22, 28, 33, 34, 36, 53, 55
- matrix (classe), 21
- max, 10, 11, 47, 57
- maximum
 - cumulatif, 48
 - d'un vecteur, 47
 - parallèle, 48
 - position dans un vecteur, 46
- mean, 19, 31, 47, 57

- median, **47**, 57
- médiane, 47
- min, **10**, **11**, **47**, 57
- minimum
 - cumulatif, 48
 - d'un vecteur, 47
 - parallèle, 48
 - position dans un vecteur, 46
- mode, **17**, 39
- mode, **16**, 30, 31, 35, 36
- moyenne
 - arithmétique, 47
 - harmonique, 69
 - pondérée, 69
 - tronquée, 47
- NA, **19**
- na.rm, **19**, 31
- names, 32, 36–38
- names (attribut), **20**
- NaN, **19**
- nchar, **18**, 30
- ncol, **11**, 33, 34, 44, **48**, 55, 58
- next, **51**
- noms d'objets
 - conventions, 15
 - réservés, 16
- Notepad++, 8
- nrow, **11**, 33, 44, **48**, 55, 57
- NULL, **18**, 20
- NULL (mode), **18**
- numeric, **21**, 31, 33, 38, 58, 59
- numeric (mode), **17**, 21
- order, **45**, 56
- outer, **49**, 51, 58, 67
- package, 51
- pgamma, 67
- plot, 10, 32
- pmax, **48**, 57, 58
- pmin, **48**, 57
- point fixe, 68
- print, 53, 54, 58–60
- prod, **47**, 50, 57, 58
- produit, 47
 - cumulatif, 48
 - extérieur, 49
- q, 8
- quantile, 47
- quantile, **47**, 57
- Répertoire de travail, 9
- répertoire de travail, 9
- rang, 45
- range, **47**, 57
- rank, **45**, 56
- rbind, **24**, 26, 34, 39
- renverser un vecteur, 45
- rep, **10**, **44**, 55, 58, 60
- repeat, **51**, 59, 68
- répétition de valeurs, 44
- replace, 37, 57
- rev, **45**, 56, 57, 64
- rm, 11
- rnorm, 10
- round, **11**, **46**, 56
- row.names, 36
- rowMeans, **49**, 61
- rowSums, **48**, 58, 61
- runif, 10, 11
- S, 1, 2
- S+, 1
- S-PLUS, 1
- sample, 32, 37, 57, 61
- sapply, 51
- save.image, 4, 8, 70
- Scheme, 2
- sd, **47**, 57
- search, **51**, 60

seq, 10, 30, 36, 44, 55, 60

sin, 28

solve, 11, 49, 57

somme, 47

 cumulative, 48

sort, 45, 55

suite de nombres, 44

sum, 19, 47, 57, 58

summary, 47, 57

switch, 51

T, voir TRUE

t, 11, 49, 57

tableau, 61

tail, 45, 56

tri, 45

TRUE, 16

trunc, 46, 57

typeof, 17

unique, 45, 56

unlist, 26, 36

valeur présente, 63, 69, 70

var, 47, 57

variance, 47

vecteur, 20, 41

vector, 33, 35

vide, voir NULL

which, 45, 56

which.max, 46, 56

which.min, 46, 56

while, 51, 59

WinEdt, 8

ISBN
978-2-98