

# 1 Installer des packages

Fabrice Dessaint

Inra - UMR Agroécologie, Dijon

10 mai 2014

*R est un système d'analyse statistique et graphique, qui est distribué librement et dont le développement et la distribution sont assurés par un groupe de personnes rassemblées dans le « R Development Core Team ». Un « package », ou bibliothèque, est généralement, un ensemble de fonctions R qui sont mises à la disposition des utilisateurs pour effectuer certains traitements qui n'existaient pas dans R.*

*Cette note présente différentes manières d'installer et de mettre à jour les packages de R.*

## Sommaire

1.1 Installer un package . . . . .	2
1.1.1 À partir de menus . . . . .	3
1.1.2 Par la ligne de commande . . . . .	4
1.2 Mise à jour des packages . . . . .	4
1.2.1 À partir de menus . . . . .	5
1.2.2 Par la ligne de commande . . . . .	5
1.3 Suppression des packages . . . . .	5
1.4 Utilisation des packages . . . . .	5
1.4.1 Charger un package . . . . .	5
1.4.2 Décharger un package . . . . .	6
1.5 Les « views » . . . . .	6

Un *package*<sup>1</sup> est un ensemble de fonctions, de fichiers d'aide et de jeux de données, construit autour d'un « thème ». Généralement, c'est une méthode statistique (*cluster*, *nlme* par exemple) mais on peut aussi trouver des *packages* ne contenant que des jeux de données (*datasets*) ou des fonctions permettant d'augmenter les possibilités graphiques de R (*grid*, *lattice* par exemple), etc.

Lors de l'installation de R, un certain nombre de *packages* sont installés (Tableau 1.1). Les *packages* présents sont de deux types :

- *packages* de « base » au nombre de 14 : *base*, *compiler*, *datasets*, *graphics*, *grDevices*, *grid*, *methods*, *parallel*, *splines*, *stats*, *stats4*, *tcltk*, *tools*, *utils*. Ils constituent en quelque sorte, le cœur du logiciel. Ils définissent le langage, permettent de faire les analyses courantes, donnent accès aux fonctions de base (mathématique, etc.) et aux fonctions permettant de faire les graphiques.
- *packages* « recommended » au nombre de 15 : *boot*, *class*, *cluster*, *codetools*, *foreign*, *KernSmooth*, *lattice*, *MASS*, *Matrix*, *mgcv*, *nlme*, *nnet*, *rpart*, *spatial*, *survival*. Ces *packages* complètent les fonctions de base.

Les *packages* installés contiennent de très nombreuses fonctions mais évidemment pas toutes les fonctions dont on pourrait avoir besoin. Cela n'est pas un problème puisque, R étant un langage de programmation, on peut écrire ses propres fonctions et les utiliser de la même façon que les fonctions déjà présentes.

Néanmoins, dans de nombreux cas, on aura tout intérêt à rechercher si les fonctions souhaitées n'ont pas déjà été écrites. C'est là un autre des avantages de R. En effet, il existe une très grande communauté d'utilisateurs qui ayant développé des fonctions, les mettent ensuite à disposition sous forme de *packages*.

---

1. Bibliothèque, paquet ou extension

Package	Type	Description
base*	base	The R Base Package
boot	recommended	Bootstrap R (S-Plus) Functions (Canty)
class	recommended	Functions for Classification
cluster	recommended	Cluster Analysis Extended Rousseeuw et al.
codetools	recommended	Code Analysis Tools for R
compiler	base	The R Compiler Package
datasets*	base	The R Datasets Package
foreign	recommended	Read Data Stored by Minitab, S, SAS, SPSS, Stata, Systat, dBase,
graphics*	base	The R Graphics Package
grDevices*	base	The R Graphics Devices and Support for Colours and Fonts
grid	base	The Grid Graphics Package
KernSmooth	recommended	Functions for kernel smoothing for Wand & Jones (1995)
lattice	recommended	Lattice Graphics
MASS	recommended	Support Functions and Datasets for Venables and Ripley's MASS
Matrix	recommended	Sparse and Dense Matrix Classes and Methods
methods*	base	Formal Methods and Classes
mgcv	recommended	GAMs with GCV/AIC/REML smoothness estimation and GAMMs by PQL
parralel	base	Support for Parralel computation in R
nlme	recommended	Linear and Nonlinear Mixed Effects Models
nnet	recommended	Feed-forward Neural Networks and Multinomial Log-Linear Models
rpart	recommended	Recursive Partitioning
spatial	recommended	Functions for Kriging and Point Pattern Analysis
splines	base	Regression Spline Functions and Classes
stats*	base	The R Stats Package
stats4	base	Statistical Functions using S4 Classes
survival	recommended	Survival analysis, including penalised likelihood.
tcltk	base	Tcl/Tk Interface
tools	base	Tools for Package Development
utils*	base	The R Utils Package

TABLE 1.1 – Liste des *packages* installés par *R*. Les *packages* suivis d'une étoile (\*) sont « chargés » au lancement de *R*.

Il existe plus de 5513 *packages* (mai 2014) disponibles sur les sites CRAN (« Comprehensive *R* Archive Network » : voir [packages](#)). C'est à cet endroit que vous pouvez avoir la liste des *packages* disponibles et, surtout, une brève description de leurs fonctionnalités.

## 1.1 Installer un package

Il existe plusieurs manières d'installer un nouveau *package*. L'installation peut se faire :

- à partir de l'interface graphique de *R* : menu *Packages* de la fenêtre RConsole ;
- en utilisant la fonction `install.package()`.

Quelque soit la méthode utilisée, il faut ouvrir une session *R*. Pour cela, il suffit de double cliquer sur le raccourci présent sur le bureau ou de lancer le programme depuis le menu démarrer de Windows.

On ouvre la fenêtre de commande RConsole. L'invite de *R*, par défaut le signe supérieur (>), indique que le programme est prêt à accepter une commande. Le principe de *R* est le suivant : vous saisissez une ligne de commande et quand vous la validez par **Entrée**, la commande est exécutée.

On peut afficher les *packages* déjà installés avec la fonction `installed.packages()` ou à partir de l’item *Charger le package ...* du menu *Package*. Les *packages* déjà installés, sont présentés par ordre alphabétique sans tenir compte de la casse (minuscule/majuscule).

### 1.1.1 À partir de menus

On suppose pour la suite, que l’ordinateur sur lequel se fait l’installation dispose d’une connexion internet. Dans ce cas, c’est le programme qui se charge d’aller chercher le *package* sélectionné puis de l’installer. Si ce n’est pas le cas, on peut toujours installer un *package* à partir d’un fichier. Ce fichier peut avoir été téléchargé depuis une autre machine puis avoir été stocké sur un support comme un CD, une clé USB, etc.

**Remarque** : Sous Windows, la façon la plus simple d’installer de nouveaux *packages* passe par l’utilisation du menu *Packages* de la fenêtre RConsole.

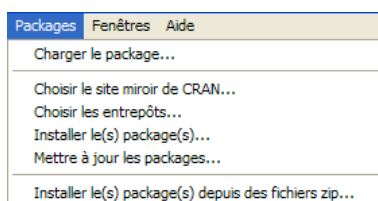


FIGURE 1.1 – Menu *Package*

Dans ce menu, vous sélectionnez l’item *Installer le(s) package(s) ...*

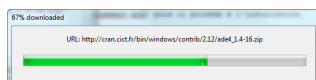
*R* vous demande alors de choisir le site à partir duquel il va effectuer le téléchargement (Figure 1.2a).

Dans la mesure du possible, et pour éviter de saturer le site principal, il vous ait demandé de choisir un site miroir géographiquement proche. Tous les sites miroirs possèdent la même information. Il n’y a donc aucun intérêt à choisir un site en Chine alors qu’il existe cinq sites miroirs en France : deux à Paris, un à Montpellier et deux à Lyon.

Une fois, le site choisi, vous allez pouvoir sélectionner le(s) *package(s)* à installer.

Vous pouvez sélectionner plusieurs *packages* en utilisant les touches SHIFT (sélection contigüe) ou CTRL (sélection espacée) (Figure 1.2b).

Une fois le choix fait et validé par **OK**, *R* commence le téléchargement,



puis il procède à l’installation.

Si l’un des *packages* utilise des fonctions qui ne sont pas disponibles — elles appartiennent à un *package* que vous n’avez pas installé — il va aussi le télécharger et l’installer.

Cette méthode est à privilégier car elle permet de gérer les dépendances : c’est-à-dire, le fait d’installer non seulement le *package* sélectionné mais aussi tous les *packages* qu’il utilise. En effet, l’auteur d’un *package* peut trouver plus efficace d’utiliser des fonctions déjà présentes dans un autre *package* plutôt que de les recoder.

Une fois l’installation terminée, *R* vous indique dans la fenêtre de commande, ce qu’il a fait et si tout c’est bien passé.

```
trying URL 'http://cran.univ-lyon1.fr/bin/windows/contrib/3.1/ade4_1.6-2.zip'
Content type 'application/zip' length 2806196 bytes (2.7 Mb)
opened URL
downloaded 2.7 Mb
```

## 1 Installer des packages

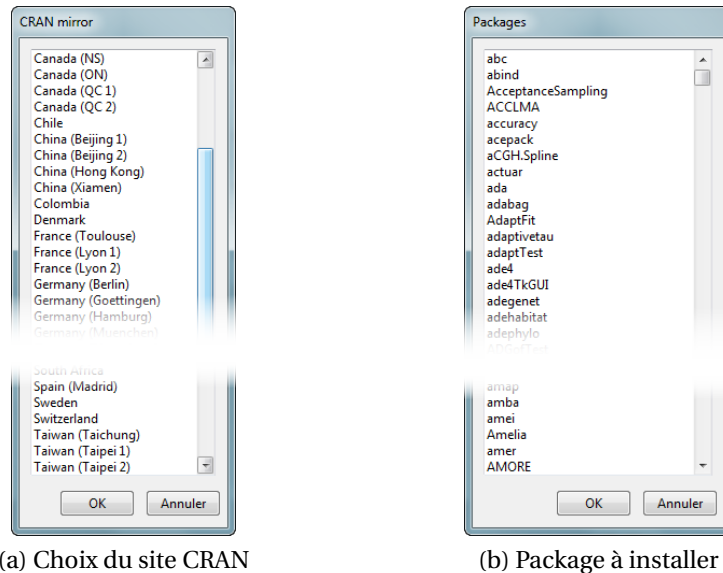


FIGURE 1.2 – L’installation de nouveaux *packages* nécessitent de choisir (a) le site CRAN, puis dans la fenêtre suivante, (b) le(s) *package*(s) à installer

Il peut arriver dans certain cas, que le *package* que vous souhaitez installer ne soit pas sur les sites CRAN mais sur un site personnel. L’auteur du *package* ne souhaite pas ou n’a pas le temps (l’envie) de le faire. Il vous le propose alors sous forme d’un fichier à télécharger (sous forme d’une archive au format *.zip*). Vous pouvez installer ce *package* en utilisant l’option *Installer le(s) package(s) depuis des fichiers zip ...* du menu *Packages*. *R* ouvre alors l’explorateur Windows afin que vous lui indiquiez où se trouve le *package* à installer. La suite des opérations est la même que dans le cas précédent.

### 1.1.2 Par la ligne de commande

Vous pouvez aussi procéder à l’installation de nouveaux *packages* directement depuis la fenêtre de commande (**R Console**) en utilisant la fonction `install.packages()`. On va installer le package *ade4*

```
install.packages("ade4",dependencies="Depends")
```

L’argument `dependencies="Depends"` indique à *R* qu’il doit aussi installer tous les autres *packages* utiles au bon fonctionnement du *package* choisi. *R* vous demande d’abord de choisir le site CRAN sur lequel, il va effectuer le téléchargement

```
--- SVP sélectionner un miroir CRAN pour cette session ---
```

en affichant la liste des sites de téléchargements (Figure 1.2a). Puis il installe le *package* sélectionné et les *packages* complémentaires.

On peut aussi installer des *packages* disponibles sous forme d’une archive au format *.zip*

```
install.packages(repos=NULL, pkgs=file.choose())
```

L’installation d’un *package* n’est à faire qu’une seule fois. Par contre, il sera nécessaire de vérifier régulièrement l’existence de mise à jour.

## 1.2 Mise à jour des packages

Certains *packages* sont en constante évolution. Il y a régulièrement de nouvelles versions qui sont disponibles. Pour profiter des ajouts et corrections, il faut donc régulièrement les mettre à jour.

### 1.2.1 À partir de menus

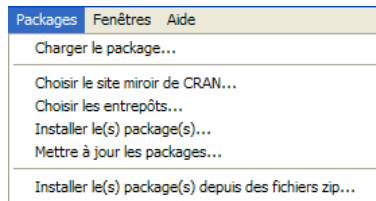


FIGURE 1.3 – Mise à jour des *packages* via le menu *Packages*

Comme pour l’installation, on peut effectuer cette opération à partir du menu *Packages* de la fenêtre de commande.

Cette fois, on va utiliser l’option *Mettre à jour les packages...* *R* commence par rechercher dans la liste des *packages* que vous avez installés, ceux pour lesquels il existe une mise à jour.

Il les affiche alors dans une fenêtre. Par défaut, tous les *packages* sont sélectionnés. Vous pouvez cependant, décider de ne mettre à jour que certains *packages*. Il suffit pour cela de les dé-sélectionner.

Une fois le choix fait, il suffit de valider par **OK**, pour lancer la mise à jour.

### 1.2.2 Par la ligne de commande

Comme pour l’installation de nouveaux *packages*, il existe une fonction, `update.packages()`, qui permet de faire la mise à jour depuis la ligne de commande.

```
update.packages()
```

```
> update.packages()
MASS :
  Version 7.3-31 installed in C:/Program Files/R/R-3.1.0/library
  Version 7.3-33 available at http://cran.univ-lyon1.fr
Update (y/N/c)? y
```

Pour chaque *package* devant être mis à jour, *R* vous demande ce que vous voulez faire : *y* procéder à la mise à jour, *N* passer au *package* suivant sans faire la mise à jour et *c* annuler les mises à jour restantes.

## 1.3 Suppression des packages

On peut aussi vouloir supprimer (désinstaller) un *package*. On le fait avec la fonction `remove.packages()`

```
remove.packages("ade4")
```

```
Elimination de(s) package(s) de 'C:/Program Files/R/R-3.1.0/library'
```

## 1.4 Utilisation des packages

### 1.4.1 Charger un package

Pour utiliser les fonctions contenues dans ces *packages*, il faut préalablement les « charger<sup>2</sup> ». Cela se fait avec les fonctions `library()` ou `require()`.

2. C’est le cas général, sauf pour quelques *packages* de base qui sont chargés au lancement de *R* (Tableau 1.1)

```
library(MASS)
```

On peut afficher la liste des *packages* chargés avec la fonction `sessionInfo()`.

```
sessionInfo()
```

```
R version 3.1.0 (2014-04-10)
Platform: i386-w64-mingw32/i386 (32-bit)

locale:
[1] LC_COLLATE=French_France.1252 LC_CTYPE=French_France.1252
[3] LC_MONETARY=French_France.1252 LC_NUMERIC=C
[5] LC_TIME=French_France.1252

attached base packages:
[1] stats    graphics  grDevices  utils      datasets  methods   base

loaded via a namespace (and not attached):
[1] tools_3.1.0
```

### 1.4.2 Télécharger un package

On va pouvoir « télécharger » un *package* avec la fonction `detach()`.

```
detach(package:MASS)
```

## 1.5 Les « views »

Les « views » sont des compilations, regroupant par grands thèmes, les *packages* jugés les plus pertinents (Tableau 1.2). Il existe actuellement 33 « views » (mai 2014).

Pour installer de façon automatique ces compilations, il faut installer le package *ctv*, puis le charger :

```
install.packages("ctv",dependencies="Depends")
library("ctv")
```

Ensuite on peut installer les compilations avec la fonction `install.view()`

```
install.views("Econometrics")
```

ou avec la fonction `update.view()`

```
update.views("Econometrics")
```

La validation de ces commandes va lancer le téléchargement et l'installation de différents *packages* sauf s'ils ont déjà été installés (pour `update.view()`).

View	Description
Bayesian	Bayesian Inference
ChemPhys	Chemometrics and Computational Physics
ClinicalTrials	Clinical Trial Design, Monitoring, and Analysis
Cluster	Cluster Analysis & Finite Mixture Models
DifferentialEquations	Differential Equations
Distributions	Probability Distributions
Econometrics	Computational Econometrics
Environmetrics	Analysis of Ecological and Environmental Data
ExperimentalDesign	Design of Experiments (DoE) & Analysis of Experimental Data
Finance	Empirical Finance
Genetics	Statistical Genetics
Graphics	Graphic Displays & Dynamic Graphics & Graphic Devices & Visualization
HighPerformanceComputing	High-Performance and Parallel Computing with R
MachineLearning	Machine Learning & Statistical Learning
MedicalImaging	Medical Image Analysis
MetaAnalysis	Meta-Analysis
Multivariate	Multivariate Statistics
NaturalLanguageProcessing	Natural Language Processing
NumericalMathematics	Numerical Mathematics
OfficialStatistics	Official Statistics & Survey Methodology
Optimization	Optimization and Mathematical Programming
Pharmacokinetics	Analysis of Pharmacokinetic Data
Phylogenetics	Phylogenetics, Especially Comparative Methods
Psychometrics	Psychometric Models and Methods
ReproducibleResearch	Reproducible Research
Robust	Robust Statistical Methods
SocialSciences	Statistics for the Social Sciences
Spatial	Analysis of Spatial Data
SpatioTemporal	Handling and Analyzing Spatio-Temporal Data
Survival	Survival Analysis
TimeSeries	Time Series Analysis
WebTechnologies	Web Technologies and Services
gR	gRaphical Models in R

TABLE 1.2 – Liste des « views » actuellement disponibles sur les sites du CRAN