

Formation R Perfectionnement

Drees – 16-17 avril 2018



Martin CHEVALIER (Insee)

Introduction : Se perfectionner avec R

Connaître plus ou connaître mieux ?

Comme tout langage statistique ou de programmation, R repose sur un ensemble d'instructions plus ou moins complexes.

Introduction : Se perfectionner avec R

Connaître plus ou connaître mieux ?

Comme tout langage statistique ou de programmation, R repose sur un ensemble d'instructions plus ou moins complexes.

Se perfectionner dans la maîtrise de R peut donc signifier deux choses :

- ▶ étendre son « vocabulaire » d'instructions connues ;
- ▶ mieux comprendre les instructions déjà connues.

Introduction : Se perfectionner avec R

Connaître plus ou connaître mieux ?

Comme tout langage statistique ou de programmation, R repose sur un ensemble d'instructions plus ou moins complexes.

Se perfectionner dans la maîtrise de R peut donc signifier deux choses :

- ▶ étendre son « vocabulaire » d'instructions connues ;
- ▶ mieux comprendre les instructions déjà connues.

En pratique, les deux **vont de pair** : en découvrant de nouvelles fonctions, on est souvent amené à mieux comprendre le fonctionnement de celles que l'on croyait maîtriser.

Introduction : Se perfectionner avec R

Plan de la partie

Chercher (et trouver !) de l'aide

Découvrir de nouvelles fonctionnalités

Utiliser de nouveaux *packages*

Chercher (et trouver!) de l'aide

Savoir utiliser l'aide du logiciel

À tout moment, taper `help(nomFonction)` ou ?
`nomFonction` affiche l'aide de la fonction `nomFonction`.

```
# Aide de la fonction read.csv  
? read.csv
```

Chercher (et trouver !) de l'aide

Savoir utiliser l'aide du logiciel

À tout moment, taper `help(nomFonction)` ou ?
`nomFonction` affiche l'aide de la fonction `nomFonction`.

```
# Aide de la fonction read.csv  
? read.csv
```

Remarque Pour afficher l'aide sur une fonction d'un *package*, il faut que celui-ci soit au préalable chargé (avec `library()` ou `require()`).

Chercher (et trouver !) de l'aide

Savoir utiliser l'aide du logiciel

À tout moment, taper `help(nomFonction)` ou `? nomFonction` affiche l'aide de la fonction `nomFonction`.

```
# Aide de la fonction read.csv  
? read.csv
```

Remarque Pour afficher l'aide sur une fonction d'un *package*, il faut que celui-ci soit au préalable chargé (avec `library()` ou `require()`).

La fonction `help.search()` ou la commande `??` permettent d'effectuer une recherche approximative :

```
# Recherche à partir du mot-clé csv  
?? csv
```


Chercher (et trouver !) de l'aide

Chercher de l'aide en ligne

Bien souvent, le problème que l'on rencontre a **déjà été rencontré par d'autres**.

Pour progresser dans la maîtrise de R, il ne faut donc surtout pas hésiter à s'appuyer sur les forums de discussion, comme par exemple [stackoverflow](#).

Chercher (et trouver !) de l'aide

Chercher de l'aide en ligne

Bien souvent, le problème que l'on rencontre a **déjà été rencontré par d'autres**.

Pour progresser dans la maîtrise de R, il ne faut donc surtout pas hésiter à s'appuyer sur les forums de discussion, comme par exemple [stackoverflow](#).

On gagne ainsi souvent beaucoup de temps en formulant le problème que l'on rencontre dans un **moteur de recherche** pour consulter certaines réponses.

Quand une question semble ne pas avoir été déjà posée, ne pas hésiter à la poser soi-même, en joignant alors un **exemple reproductible** (*minimal working example* ou MWE).

Chercher (et trouver !) de l'aide

Afficher le code d'une fonction

Quand l'utilisation d'une fonction pose problème (message d'erreur inattendu), il est souvent utile d'**afficher son code** pour comprendre d'où vient le problème.

Chercher (et trouver !) de l'aide

Afficher le code d'une fonction

Quand l'utilisation d'une fonction pose problème (message d'erreur inattendu), il est souvent utile d'**afficher son code** pour comprendre d'où vient le problème.

Pour ce faire, il suffit de saisir son nom sans parenthèse.

```
# Code de la fonction read.csv
```

```
read.csv
```

```
## function (file, header = TRUE, sep = ",", quote = "\"", dec =  
##      fill = TRUE, comment.char = "", ...)  
## read.table(file = file, header = header, sep = sep, quote =  
##      dec = dec, fill = fill, comment.char = comment.char, .  
## <bytecode: 0x000000000a6d5f20>  
## <environment: namespace:utils>
```

Chercher (et trouver !) de l'aide

Afficher le code d'une fonction

Quand l'utilisation d'une fonction pose problème (message d'erreur inattendu), il est souvent utile d'**afficher son code** pour comprendre d'où vient le problème.

Pour ce faire, il suffit de saisir son nom sans parenthèse.

```
# Code de la fonction read.csv
```

```
read.csv
```

```
## function (file, header = TRUE, sep = ",", quote = "\"", dec =  
##      fill = TRUE, comment.char = "", ...)  
## read.table(file = file, header = header, sep = sep, quote =  
##      dec = dec, fill = fill, comment.char = comment.char, .  
## <bytecode: 0x000000000a6d5f20>  
## <environment: namespace:utils>
```

Afficher le code d'une fonction est dans certains cas plus difficile, cf. [stackoverflow](#).

Découvrir de nouvelles fonctionnalités

Se repérer dans les CRAN *Task Views*

Les CRAN *Task Views* recensent les fonctions et *packages* de façon thématique. Elles sont mises à jour régulièrement et portent sur des thèmes variés :

Découvrir de nouvelles fonctionnalités

Se repérer dans les CRAN *Task Views*

Les CRAN *Task Views* recensent les fonctions et *packages* de façon thématique. Elles sont mises à jour régulièrement et portent sur des thèmes variés :

Bayesian, ChemPhys, ClinicalTrials, Cluster, DifferentialEquations, Distributions, Econometrics, Environmetrics, ExperimentalDesign, ExtremeValue, Finance, FunctionalData, Genetics, Graphics, HighPerformanceComputing, MachineLearning, MedicalImaging, MetaAnalysis, ModelDeployment, Multivariate, NaturalLanguageProcessing, NumericalMathematics, OfficialStatistics, Optimization, Pharmacokinetics, Phylogenetics, Psychometrics, ReproducibleResearch, Robust, SocialSciences, Spatial, SpatioTemporal, Survival, TimeSeries, WebTechnologies, gR

Découvrir de nouvelles fonctionnalités

Se repérer dans les CRAN *Task Views*

Les CRAN *Task Views* recensent les fonctions et *packages* de façon thématique. Elles sont mises à jour régulièrement et portent sur des thèmes variés :

Bayesian, ChemPhys, ClinicalTrials, Cluster, DifferentialEquations, Distributions, Econometrics, Environmetrics, ExperimentalDesign, ExtremeValue, Finance, FunctionalData, Genetics, Graphics, HighPerformanceComputing, MachineLearning, MedicalImaging, MetaAnalysis, ModelDeployment, Multivariate, NaturalLanguageProcessing, NumericalMathematics, OfficialStatistics, Optimization, Pharmacokinetics, Phylogenetics, Psychometrics, ReproducibleResearch, Robust, SocialSciences, Spatial, SpatioTemporal, Survival, TimeSeries, WebTechnologies, gR

La liste de toutes les *Task Views* est accessible à la page :
<https://cran.r-project.org/web/views>.

Découvrir de nouvelles fonctionnalités

Consulter des sites, tutoriels, livres

De plus en plus de supports sont consacrés à la présentation et à l'enseignement des fonctionnalités de R, comme par exemple :

Découvrir de nouvelles fonctionnalités

Consulter des sites, tutoriels, livres

De plus en plus de supports sont consacrés à la présentation et à l'enseignement des fonctionnalités de R, comme par exemple :

- ▶ le site R-bloggers : articles en général courts sur des exemples d'applications (de qualité inégale) ;

Découvrir de nouvelles fonctionnalités

Consulter des sites, tutoriels, livres

De plus en plus de supports sont consacrés à la présentation et à l'enseignement des fonctionnalités de R, comme par exemple :

- ▶ le site R-bloggers : articles en général courts sur des exemples d'applications (de qualité inégale) ;
- ▶ le site bookdown.org : dépôt de livres numériques consacrés à R élaborés avec R Markdown (très riches et très complets) ;

Découvrir de nouvelles fonctionnalités

Consulter des sites, tutoriels, livres

De plus en plus de supports sont consacrés à la présentation et à l'enseignement des fonctionnalités de R, comme par exemple :

- ▶ le site R-bloggers : articles en général courts sur des exemples d'applications (de qualité inégale) ;
- ▶ le site bookdown.org : dépôt de livres numériques consacrés à R élaborés avec R Markdown (très riches et très complets) ;
- ▶ le site de RStudio : nombreux aides-mémoires ou articles présentant les fonctionnalités de l'écosystème RStudio ;

Découvrir de nouvelles fonctionnalités

Consulter des sites, tutoriels, livres

De plus en plus de supports sont consacrés à la présentation et à l'enseignement des fonctionnalités de R, comme par exemple :

- ▶ le site R-bloggers : articles en général courts sur des exemples d'applications (de qualité inégale) ;
- ▶ le site bookdown.org : dépôt de livres numériques consacrés à R élaborés avec R Markdown (très riches et très complets) ;
- ▶ le site de RStudio : nombreux aides-mémoires ou articles présentant les fonctionnalités de l'écosystème RStudio ;
- ▶ les ouvrages de Hadley Wickham : ggplot2: elegant graphics for data analysis (à compiler soi-même), Advanced R.

Utiliser de nouveaux *packages*

Accéder à la documentation d'un *package*

Une des principales forces de R est d'être un langage hautement modulaire comptant **plusieurs milliers de *packages*** (12 184 au 01/06/2018).

Utiliser de nouveaux *packages*

Accéder à la documentation d'un *package*

Une des principales forces de R est d'être un langage hautement modulaire comptant **plusieurs milliers de *packages*** (12 184 au 01/06/2018).

Toutes les informations sur un *package* sont accessibles sur sa page du *Comprehensive R Archive Network* (CRAN).

Exemple <https://CRAN.R-project.org/package=haven>

Utiliser de nouveaux *packages*

Accéder à la documentation d'un *package*

Une des principales forces de R est d'être un langage hautement modulaire comptant **plusieurs milliers de *packages*** (12 184 au 01/06/2018).

Toutes les informations sur un *package* sont accessibles sur sa page du *Comprehensive R Archive Network* (CRAN).

Exemple <https://CRAN.R-project.org/package=haven>

On trouve en particulier sur cette page :

- ▶ les **dépendances** du *package* (*Depends* et *Imports*) ;
- ▶ un lien vers sa **page de développement** (*URL*) ;
- ▶ une **version .pdf de son aide** (*Reference manual*)
- ▶ éventuellement un ou plusieurs **documents de démonstration** (*Vignettes*).

Utiliser de nouveaux *packages*

Installer un *package* automatiquement

La fonction `install.packages("nomPackage")` permet d'installer automatiquement le *package* `nomPackage`.

Les données nécessaires sont téléchargées depuis un des dépôts du CRAN (*repositories* ou en abrégé *repos*).

C'est la **méthode à privilégier** : les dépendances nécessaires au bon fonctionnement du *package* sont détectées et automatiquement installées.

Utiliser de nouveaux *packages*

Installer un *package* manuellement

La page d'information d'un *package* comporte également des liens vers les fichiers qui le composent.

Quand l'installation directe depuis un dépôt du CRAN est indisponible, il suffit de **télécharger ces fichiers** et d'**installer manuellement le *package***.

Pour une installation sous Windows, il faut privilégier les **fichiers compilés** (*Windows binaries*).

Utiliser de nouveaux *packages*

Installer un *package* manuellement

La page d'information d'un *package* comporte également des liens vers les fichiers qui le composent.

Quand l'installation directe depuis un dépôt du CRAN est indisponible, il suffit de **télécharger ces fichiers** et d'**installer manuellement le *package***.

Pour une installation sous Windows, il faut privilégier les **fichiers compilés** (*Windows binaries*).

```
# Note : Le fichier haven_1.1.0.zip est situé  
# dans le répertoire de travail  
install.packages(  
  "haven_1.1.0.zip", repos = NULL, type = "binaries"  
)
```

Utiliser de nouveaux *packages*

Installer des *packages* depuis github

En règle générale, le développement de *packages* s'appuie sur des plate-formes de **développement collaboratif** comme Github.

Utiliser de nouveaux *packages*

Installer des *packages* depuis github

En règle générale, le développement de *packages* s'appuie sur des plate-formes de **développement collaboratif** comme Github.

La **page de développement** d'un *package* comporte plusieurs informations précieuses :

- ▶ la dernière version du *package* et de sa documentation ;
- ▶ des informations sur son développement ;
- ▶ une zone pour rapporter d'éventuels *bugs* (*bug reports*).

Exemple <https://github.com/tidyverse/haven>

Utiliser de nouveaux *packages*

Installer des *packages* depuis github

En règle générale, le développement de *packages* s'appuie sur des plate-formes de **développement collaboratif** comme Github.

La **page de développement** d'un *package* comporte plusieurs informations précieuses :

- ▶ la dernière version du *package* et de sa documentation ;
- ▶ des informations sur son développement ;
- ▶ une zone pour rapporter d'éventuels *bugs* (*bug reports*).

Exemple <https://github.com/tidyverse/haven>

La fonction `install_github()` du *package* `devtools` permet d'installer un *package* directement depuis GitHub.

```
library(devtools)
install_github("tidyverse/haven")
```

Utiliser de nouveaux *packages*

Utiliser les données d'exemples d'un *package*

La plupart des **packages** contiennent des **données d'exemples** utilisées notamment dans son aide ou ses vignettes.

Une fois le *package* installé, il suffit d'utiliser la fonction `data(package = "nomPackage")` pour afficher les données qu'il contient.

```
library(ggplot2)  
data(package = "ggplot2")
```

Utiliser de nouveaux *packages*

Utiliser les données d'exemples d'un *package*

La plupart des **packages** contiennent des **données d'exemples** utilisées notamment dans son aide ou ses vignettes.

Une fois le *package* installé, il suffit d'utiliser la fonction `data(package = "nomPackage")` pour afficher les données qu'il contient.

```
library(ggplot2)  
data(package = "ggplot2")
```

Pour « rapatrier » dans l'environnement global les données d'un *package*, c'est de nouveau la fonction `data()` qu'il faut utiliser.

```
data(mpg)
```