

Packaging

Techniques avancées en programmation statistique R

Patrick Fournier

Automne 2020

Université du Québec à Montréal

Introduction

Qu'est-ce qu'un package R?

~→ Pour une description complète, la référence est *Writing R Extensions* [5].

Qu'est-ce qu'un package R?

- ~> Pour une description complète, la référence est *Writing R Extensions* [5].
- ~> Peu d'exigences :

Qu'est-ce qu'un package R?

- ~> Pour une description complète, la référence est *Writing R Extensions* [5].
- ~> Peu d'exigences :
 - ~> Un répertoire

Qu'est-ce qu'un package R?

- ~> Pour une description complète, la référence est *Writing R Extensions* [5].
- ~> Peu d'exigences :
 - ~> Un répertoire contenant
 - ~> un fichier nommé "DESCRIPTION"

Qu'est-ce qu'un package R?

- ~> Pour une description complète, la référence est *Writing R Extensions* [5].
- ~> Peu d'exigences :
 - ~> Un répertoire contenant
 - ~> un fichier nommé "DESCRIPTION"
- ~> Nom du répertoire = nom du package (convention).

Qu'est-ce qu'un package R?

- ~> Pour une description complète, la référence est *Writing R Extensions* [5].
- ~> Peu d'exigences :
 - ~> Un répertoire contenant
 - ~> un fichier nommé "DESCRIPTION"
- ~> Nom du répertoire = nom du package (convention).
- ~> DESCRIPTION contient les méta informations (nom, auteurs, description,...)

Qu'est-ce qu'un package R?

- ~> Pour une description complète, la référence est *Writing R Extensions* [5].
- ~> Peu d'exigences :
 - ~> Un répertoire contenant
 - ~> un fichier nommé "DESCRIPTION"
- ~> Nom du répertoire = nom du package (convention).
- ~> DESCRIPTION contient les méta informations (nom, auteurs, description,...)
- ~> Un tel package minimal n'est pas d'une grande utilité...

Qu'est-ce qu'un package R?

Un peu de terminologie [5]...

Qu'est-ce qu'un package R?

Un peu de terminologie [5]...

Package

Répertoire de fichiers ajoutant des fonctionnalités à R

Qu'est-ce qu'un package R?

Un peu de terminologie [5]...

Package

Répertoire de fichiers ajoutant des fonctionnalités à R
source

L'ensemble des fichiers utilisés pour le développement d'un package.

Qu'est-ce qu'un package R?

Un peu de terminologie [5]...

Package

Répertoire de fichiers ajoutant des fonctionnalités à R
source

L'ensemble des fichiers utilisés pour le développement d'un package.

bundle

Tarball contenant le package [4].

Qu'est-ce qu'un package R?

Un peu de terminologie [5]...

Package

Répertoire de fichiers ajoutant des fonctionnalités à R
source

L'ensemble des fichiers utilisés pour le développement d'un package.

bundle

Tarball contenant le package [4].

installed

Ce que l'on obtient en exécutant `R CMD INSTALL` sur un package.

Qu'est-ce qu'un package R?

Un peu de terminologie [5]...

Package

Répertoire de fichiers ajoutant des fonctionnalités à R
source

L'ensemble des fichiers utilisés pour le développement d'un package.

bundle

Tarball contenant le package [4].

installed

Ce que l'on obtient en exécutant `R CMD INSTALL` sur un package.

binary package

zip ou tarball contenant les fichiers d'un package installé (installés package).

Qu'est-ce qu'un package R ?

~→ Il est parfaitement possible de créer un package “à la main”.

Qu'est-ce qu'un package R?

- ~> Il est parfaitement possible de créer un package “à la main”.
- ~> Toutefois, le package **devtools** simplifie le travail.

Qu'est-ce qu'un package R?

- ~> Il est parfaitement possible de créer un package “à la main”.
- ~> Toutefois, le package **devtools** simplifie le travail.
 - ~> En quelque sorte le couteau suisse du développement R.

Qu'est-ce qu'un package R?

- ~> Il est parfaitement possible de créer un package “à la main”.
- ~> Toutefois, le package **devtools** simplifie le travail.
 - ~> En quelque sorte le couteau suisse du développement R.
- ~> Pour référence rapide, voir [3].

Qu'est-ce qu'un package R?

- ~> Il est parfaitement possible de créer un package “à la main”.
- ~> Toutefois, le package **devtools** simplifie le travail.
 - ~> En quelque sorte le couteau suisse du développement R.
- ~> Pour référence rapide, voir [3].
- ~> Encore mieux, plusieurs fonctionnalités de devtools sont intégrées à RStudio!



Démonstration en classe...

Structure d'un package R

.gitignore

Fichiers ignorés par git.

.gitignore

Fichiers ignorés par git.

.Rbuildignore

Fichiers ignorés lors de la création du bundle.

.gitignore

Fichiers ignorés par git.

.Rbuildignore

Fichiers ignorés lors de la création du bundle.

<nom_du_projet>.Rproj

Fichier de configuration de RStudio.

DESCRIPTION

Métadonnées du package.

DESCRIPTION

Métadonnées du package.

NAMESPACE

Importation/exportation de variables.

DESCRIPTION

Métadonnées du package.

NAMESPACE

Importation/exportation de variables.

man/

Contient la documentation.

DESCRIPTION

Métadonnées du package.

NAMESPACE

Importation/exportation de variables.

man/

Contient la documentation.

R/

Contient le code R.

Champs obligatoires

Package Nom du package.

Champs obligatoires

Package Nom du package.

Version Version du package.

Champs obligatoires

Package Nom du package.

Version Version du package.

License License à laquelle le package est soumis.

Champs obligatoires

Package Nom du package.

Version Version du package.

License License à laquelle le package est soumis.

Description Description longue du package.

Champs obligatoires

Package Nom du package.

Version Version du package.

License License à laquelle le package est soumis.

Description Description longue du package.

Title Description courte du package.

Champs obligatoires

Package Nom du package.

Version Version du package.

License License à laquelle le package est soumis.

Description Description longue du package.

Title Description courte du package.

Author Auteurs du packages (nom, e-mail et roles [2]).

Champs obligatoires

Package Nom du package.

Version Version du package.

License License à laquelle le package est soumis.

Description Description longue du package.

Title Description courte du package.

Author Auteurs du packages (nom, e-mail et roles [2]).

Maintainer Responsable (unique) du package (nom et e-mail).

Champs obligatoires

Package Nom du package.

Version Version du package.

License License à laquelle le package est soumis.

Description Description longue du package.

Title Description courte du package.

Author Auteurs du packages (nom, e-mail et roles [2]).

Maintainer Responsable (unique) du package (nom et e-mail).

Il est possible et *recommandé* de fournir un seul champ Authors@R en remplacement de Author et Maintainer.

Champs facultatifs usuels

Copyright Détenteur du copyright si différent de l'auteur.

Champs facultatifs usuels

Copyright Détenteur du copyright si différent de l'auteur.

Date Date de sortie du package (yyy-mm-dd).

Champs facultatifs usuels

Copyright Détenteur du copyright si différent de l'auteur.

Date Date de sortie du package (yyy-mm-dd).

Depends Dépendances attachés du package. (packages et/ou version de R)

Champs facultatifs usuels

Copyright Détenteur du copyright si différent de l'auteur.

Date Date de sortie du package (yyy-mm-dd).

Depends Dépendances attachés du package. (packages et/ou version de R)

Suggest Packages utiles dont ne dépend pas le package courant.

Champs facultatifs usuels

Copyright Détenteur du copyright si différent de l'auteur.

Date Date de sortie du package (yyy-mm-dd).

Depends Dépendances attachés du package. (packages et/ou version de R)

Suggest Packages utiles dont ne dépend pas le package courant.

Imports Dépendances non attachés du package.

Champs facultatifs usuels

Copyright Détenteur du copyright si différent de l'auteur.

Date Date de sortie du package (yyy-mm-dd).

Depends Dépendances attachés du package. (packages et/ou version de R)

Suggest Packages utiles dont ne dépend pas le package courant.

Imports Dépendances non attachés du package.

Collate Ordre de chargement des fichiers R. (alphabétique)

URL Liste d'URL liés au package.

DESCRIPTION

URL Liste d'URL liés au package.

BugReports URL pour la soumission des rapports de bugs.

DESCRIPTION

URL Liste d'URL liés au package.

BugReports URL pour la soumission des rapports de bugs.

LazyData Les données doivent-elle être chargés
paresseusement?

DESCRIPTION

URL Liste d'URL liés au package.

BugReports URL pour la soumission des rapports de bugs.

LazyData Les données doivent-elle être chargés
paresseusement?

ByteCompile Le package doit-il être byte compilé? (**true**)

DESCRIPTION

URL Liste d'URL liés au package.

BugReports URL pour la soumission des rapports de bugs.

LazyData Les données doivent-elle être chargés
paresseusement?

ByteCompile Le package doit-il être byte compilé? (**true**)

BuildVignette Les vignettes doivent-elles être construites?
(true)

DESCRIPTION

URL Liste d'URL liés au package.

BugReports URL pour la soumission des rapports de bugs.

LazyData Les données doivent-elle être chargés
paresseusement?

ByteCompile Le package doit-il être byte compilé? (**true**)

BuildVignette Les vignettes doivent-elles être construites?
(true)

NeedsCompilation Le package nécessite-t-il une phase de
compilation? ("yes"/"no")

DESCRIPTION

URL Liste d'URL liés au package.

BugReports URL pour la soumission des rapports de bugs.

LazyData Les données doivent-elle être chargés
paresseusement?

ByteCompile Le package doit-il être byte compilé? (**true**)

BuildVignette Les vignettes doivent-elles être construites?
(true)

NeedsCompilation Le package nécessite-t-il une phase de
compilation? ("yes"/"no")

Des champs arbitraires peuvent aussi être utilisés.

Champs facultatifs obscurs

Enhances Packages qui seront améliorés par le présent package.

LinkingTo Packages dont les headers sont utilisés par le présent package.

Additional_repositories Dépôts (autres que CRAN) contenant les packages cités dans **DESCRIPTION**.

SystemRequirements Dépendances externes du package.

KeepSource Le formatage du fichier source doit-il être conservé? (false)

StagedInstall Le package doit-il être construit dans un répertoire temporaire? (**true**)

Type Type du package. (Package/Translation)

DESCRIPTION

Biarch Le package doit-il être construit pour en version 32 et 64 bits? (Windows seulement)

VignetteBuilder Packages utilisés pour fournir le moteur servant à la construction des vignettes.

Encoding Encodage du fichier **DESCRIPTION** si différent de ascii.

OS_type OS pour lequel est destiné le package.
(unix/windows)

Classification/ACM(-2012) Sujet du package selon le Computing Classification System.

Language Langage(s) de la documentation si différent de l'anglais.

RdMacros Liste de packages desquels des macro Rd sont importés.

Quelques recommandations

~→ Le champ **Depends** ne devrait être utilisé que pour spécifier une version minimale de R (par exemple **Depends: R(>= 3.1)**).

Quelques recommandations

- ~> Le champ **Depends** ne devrait être utilisé que pour spécifier une version minimale de R (par exemple **Depends: R(>= 3.1)**).
- ~> Les champs **Imports**, **Depends**, **Suggests**, **Enhances**, et **Linkingto** *ne devraient jamais être modifiés à la main!* Utilisez plutôt la fonction `use_package` de `devtools`.

Quelques recommandations

- ~> Le champ **Depends** ne devrait être utilisé que pour spécifier une version minimale de R (par exemple **Depends: R(>= 3.1)**).
- ~> Les champs **Imports**, **Depends**, **Suggests**, **Enhances**, et **Linkingto** *ne devraient jamais être modifiés à la main!* Utilisez plutôt la fonction `use_package` de `devtools`.
- ~> Le champ **Authors@R** doit contenir un vecteur d'objets de classe **person**.



Démonstration en classe...

(contre)Exemple : shadowing

```
1 > address <- function() "PK-5323"
2 > x <- 42
3 > address()
4 [1] "PK-5323"
5 > pryr::address(x)
6 [1] "0x55ba5c1263c0"
7 > library(pryr)
8
9 Attaching package: 'pryr'
10
11 The following object is masked _by_ '.GlobalEnv':
12
13     address
14 > address(x)
15 Error in address(x) : unused argument (x)
```

↪ Le répertoire `man/` contient la documentation.

- ~> Le répertoire **man/** contient la documentation.
- ~> Il n'est pas nécessaire de la modifier par nous-même.

- ~> Le répertoire **man/** contient la documentation.
- ~> Il n'est pas nécessaire de la modifier par nous-même.
- ~> Le package **roxygen2** peut se charger de générer la documentation à notre place!

- ~> Le répertoire **man/** contient la documentation.
- ~> Il n'est pas nécessaire de la modifier par nous-même.
- ~> Le package **roxygen2** peut se charger de générer la documentation à notre place!
- ~> Pour ce faire, **roxygen2** repère les commentaires commençant par **'** et les transforme en documentation.

- ~> Le répertoire **man/** contient la documentation.
- ~> Il n'est pas nécessaire de la modifier par nous-même.
- ~> Le package **roxygen2** peut se charger de générer la documentation à notre place!
- ~> Pour ce faire, **roxygen2** repère les commentaires commençant par **'** et les transforme en documentation.
- ~> Pour mettre à jour la documentation, il faut appeler **document** du package **devtools**.

- ~> Le répertoire **man/** contient la documentation.
- ~> Il n'est pas nécessaire de la modifier par nous-même.
- ~> Le package **roxygen2** peut se charger de générer la documentation à notre place!
- ~> Pour ce faire, **roxygen2** repère les commentaires commençant par **'** et les transforme en documentation.
- ~> Pour mettre à jour la documentation, il faut appeler **document** du package **devtools**.
- ~> Pour plus de détails, voir la vignette *Generating Rd files* [1].

```
1  #' Titre
2  #'
3  #' Le second paragraphe est toujours la description. Le premier
4  #' paragraphe est le titre et ne devrait pas être plus long
5  #' qu'une phrase. Ce paragraphe donne une description générale
6  #' de l'objet à documenter.
7  #'
8  #' L'ensemble des paragraphes suivants contient les détails
9  #' de l'objet à documenter.
10 #'
```


~> Il est d'usage de documenter le package lui même.

- ~> Il est d'usage de documenter le package lui même.
- ~> Une manière de faire est de documenter la chaîne de caractère `"_PACKAGE"`.



Démonstration en classe...

Roxygen reconnaît des mots-clés (tags) qui sont préfixés de `@`.
Pour les fonctions :

`@param name description`

Décris un des arguments d'une fonction.

Roxygen reconnaît des mots-clés (tags) qui sont préfixés de `@`.
Pour les fonctions :

`@param name description`

Décris un des arguments d'une fonction.

`@examples`

Code R valide.

Roxygen reconnaît des mots-clés (tags) qui sont préfixés de `@`.
Pour les fonctions :

`@param name description`

Décris un des arguments d'une fonction.

`@examples`

Code R valide.

`@return description`

Description de la valeur de retour d'une fonction.

Roxygen reconnaît des mots-clés (tags) qui sont préfixés de `@`.
Pour les fonctions :

@param name description

Décris un des arguments d'une fonction.

@examples

Code R valide.

@return description

Description de la valeur de retour d'une fonction.

Les génériques et les méthodes S3 se documentent comme des fonction (puisque ce sont des fonctions).



Démonstration en classe...

↪ La fonction `add2` n'existe pas!

↪ La fonction **add2** n'existe pas!

↪ En fait, elle existe bel et bien, mais elle se cache.

- ~> La fonction **add2** n'existe pas!
- ~> En fait, elle existe bel et bien, mais elle se cache.
- ~> Pour la rendre facilement utilisable, il faut l'exporter explicitement.

- ~> La fonction **add2** n'existe pas!
- ~> En fait, elle existe bel et bien, mais elle se cache.
- ~> Pour la rendre facilement utilisable, il faut l'exporter explicitement.
- ~> Pour cela, nous devons ajouter une entrée dans le fichier NAMESPACE.

NAMESPACE

- ~> La fonction **add2** n'existe pas!
- ~> En fait, elle existe bel et bien, mais elle se cache.
- ~> Pour la rendre facilement utilisable, il faut l'exporter explicitement.
- ~> Pour cela, nous devons ajouter une entrée dans le fichier NAMESPACE.
- ~> Cela devient vite compliqué. Une meilleure (quoique un peu étrange) solution est de laisser **devtools::document** faire le travail.

NAMESPACE

- ~> La fonction **add2** n'existe pas!
- ~> En fait, elle existe bel et bien, mais elle se cache.
- ~> Pour la rendre facilement utilisable, il faut l'exporter explicitement.
- ~> Pour cela, nous devons ajouter une entrée dans le fichier **NAMESPACE**.
- ~> Cela devient vite compliqué. Une meilleure (quoique un peu étrange) solution est de laisser **devtools::document** faire le travail.
- ~> Pour ce faire, on ajoute simplement le tag **@export** à la fonction.



Démonstration en classe...

↪ R possède un mécanisme de test pour les packages.

- ~> R possède un mécanisme de test pour les packages.
- ~> Il est optionnel mais permet souvent d'attraper des bugs.

- ~> R possède un mécanisme de test pour les packages.
- ~> Il est optionnel mais permet souvent d'attraper des bugs.
- ~> Les exemples sont exécutés lors d'un test.

- ↪ R possède un mécanisme de test pour les packages.
- ↪ Il est optionnel mais permet souvent d'attraper des bugs.
- ↪ Les exemples sont exécutés lors d'un test.
 - ↪ Pour éviter cela, on peut utiliser `\dontrun {}`.

- ~> R possède un mécanisme de test pour les packages.
- ~> Il est optionnel mais permet souvent d'attraper des bugs.
- ~> Les exemples sont exécutés lors d'un test.
 - ~> Pour éviter cela, on peut utiliser `\dontrun {}`.
- ~> Si vous souhaitez publier votre package sur CRAN, il sera tester de manière *encore plus sévère*.

- ~> R possède un mécanisme de test pour les packages.
- ~> Il est optionnel mais permet souvent d'attraper des bugs.
- ~> Les exemples sont exécutés lors d'un test.
 - ~> Pour éviter cela, on peut utiliser `\dontrun {}`.
- ~> Si vous souhaitez publier votre package sur CRAN, il sera tester de manière *encore plus sévère*.
- ~> Pour reproduire ce test localement, on peut passer l'argument `--as-cran`.

~→ Il est possible de distribuer un package sur CRAN, mais il faut y consacrer beaucoup d'efforts!

- ~> Il est possible de distribuer un package sur CRAN, mais il faut y consacrer beaucoup d'efforts!
- ~> Une manière plus modeste est d'utiliser GitHub.

- ~> Il est possible de distribuer un package sur CRAN, mais il faut y consacrer beaucoup d'efforts!
- ~> Une manière plus modeste est d'utiliser GitHub.
- ~> Un package source disponible sur github peut être installé à l'aide de la fonction `install_github` de devtools.

- ~> Il est possible de distribuer un package sur CRAN, mais il faut y consacrer beaucoup d'efforts!
- ~> Une manière plus modeste est d'utiliser GitHub.
- ~> Un package source disponible sur github peut être installé à l'aide de la fonction `install_github` de devtools.
- ~> Il est aussi possible de générer un bundle que l'on peut donner à ses collaborateurs.



Démonstration en classe...

~> Inclure des jeux de donnée (répertoire **data**).

- ~> Inclure des jeux de donnée (répertoire **data**).
- ~> Inclure du code nécessitant une compilation (répertoire **src**).

Autres possibilités

- ~> Inclure des jeux de donnée (répertoire **data**).
- ~> Inclure du code nécessitant une compilation (répertoire **src**).
- ~> Inclure des tests pour votre package (répertoire **tests**).

Autres possibilités

- ~> Inclure des jeux de donnée (répertoire **data**).
- ~> Inclure du code nécessitant une compilation (répertoire **src**).
- ~> Inclure des tests pour votre package (répertoire **tests**).
 - ~> *Vital* pour le développement de package sérieux.

Autres possibilités

- ~> Inclure des jeux de donnée (répertoire **data**).
- ~> Inclure du code nécessitant une compilation (répertoire **src**).
- ~> Inclure des tests pour votre package (répertoire **tests**).
 - ~> *Vital* pour le développement de package sérieux.
 - ~> Package testthat.

Autres possibilités

- ~> Inclure des jeux de donnée (répertoire **data**).
- ~> Inclure du code nécessitant une compilation (répertoire **src**).
- ~> Inclure des tests pour votre package (répertoire **tests**).
 - ~> *Vital* pour le développement de package sérieux.
 - ~> Package testthat.
 - ~> Pour un exemple d'utilisation, voir les katas du premiers cours.

Références

- [1] *Generating Rd files*. URL :
<https://cran.r-project.org/web/packages/roxygen2/vignettes/rd.html>.
- [2] Kurt HORNIK, Duncan MURDOCH et Achim ZEILEIS. *Who Did What? The Roles of R Package Authors and How to Refer to Them*. en. Paper. Nov. 2011. URL :
<http://statmath.wu.ac.at/>.

- [3] RSTUDIO. *Package Development Cheat Sheet*. English. Jan. 2015. URL :
<https://resources.rstudio.com/rstudio-cheatsheets/package-development-cheat-sheet>.
- [4] Hadley WICKHAM. *R Packages : Organize, Test, Document, and Share Your Code*. en. Google-Books-ID : DqSxBwAAQBAJ. "O'Reilly Media, Inc.", mars 2015. ISBN : 9781491910566.
- [5] *Writing R Extensions*. URL : <https://cran.r-project.org/doc/manuals/R-exts.html>.