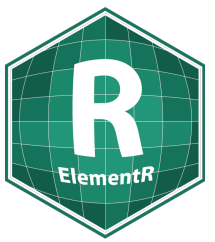
Création de packages avec R

ElementR



Timothée Giraud

Centre pour l'analyse spatiale et la géovisualisation

28 mai 2026

Les idées reçues

- créer un package est difficile
- la création de packages est réservée aux informaticiens
- il faut être un expert de R pour créer un package
- un package est à destination du CRAN
- un package sert à partager son code
- je n’ai pas beaucoup de fonctions, pas besoin de faire un package
- cela prend du temps

Quand créer un package

- Si vous utilisez plus de deux fois une même portion de code dans une analyse, créez une **fonction**.
- Si vous réutilisez une même fonction dans deux analyses différentes, faites un **package**.

> Créer un package permet de gagner du temps et de s'astreindre à documenter ses fonctions.

Qu’est ce qu’un package

La structure des packages R est assez simple : quelques dossiers et fichiers.

Dans le dossier d’un package il faut obligatoirement :

- Un dossier **R/** contenant les fichiers .R avec les fonctions
- Un fichier **DESCRIPTION** décrivant le package (description, auteurs, licence, dépendances, version)
- Un fichier **NAMESPACE** décrivant les fonctions importées et exportées du package
- Un dossier **man/** contenant la documentation des fonctions

> En pratique, le fichier **NAMESPACE** et le dossier **/man** sont construits automatiquement à partir des informations fournies dans les fichiers contenant les fonctions.

Que faire avec mon package

- le mettre uniquemement sur ma machine
- le partager avec une équipe, un projet
- le déposer sur une forge logicielle (github / gitlab / codeberg)
- le rendre disponible sur le r-universe
- le soumettre sur le CRAN

Pourquoi mettre son package sur une forge

- gérer le développement de manière ouverte et transparente
- coopérer avec d'autres personnes et faciliter les contributions
- gérer les remontées de bug et les demandes de fonctionnalités via les *issues*
- CI/CD : bénéficier de l'intégration continue (tests, builds) et du déploiement continu (site web)

Pourquoi mettre son package sur le CRAN

- Robustesse
- Cohérence
- Prestige

Faire connaître son travail

Via des publications :

- JOSS (+ ROpenSci)
- Journal of Statistical Software
- R Journal
- Rzine
- blogs
- Mastodon

ou des conférences :

- FOSS4G
- Rencontres R
- useR
- SAGEO
- ECTQG
- …

Maintenir un package et prendre soin de ses utilisateurs

- utiliser une méthode de versionnement explicite
- maintenir une documentation à jour
- communiquer sur les évolutions du package (*changelog*, blog…)
- veiller à la *backward compatibility*
- veille sur les packages dont dépend le mien (*upstream dependencies*)
- veille sur les packages qui dépendent du mien (*downstream dependencies*)

Ressources

- Writing R Extensions, la documentation officielle
- R Packages de Hadley Wickham et Jennifer Bryan
- rOpenSci Packages: Development, Maintenance, and Peer Review
- Créer un package R par la pratique de Romain Lesur dans *Travail collaboratif avec R* de Lino Galiana
- Créer son premier package R de Juliette Engelaere-Lefebvre et Maël Theuliere
- Fabriquer un package R en moins de 6 minutes de Diane Beldame