

Pourquoi créer des packages R en recherche?

Judith Bourque

2023-06-27

Table of contents

- Introduction aux packages
 - C'est quoi un package R?
 - Pourquoi créer un package?
 - Types de packages
- Développer un package
 - Comment créer un package?
 - Comment partager un package?
- Ressources

Introduction aux packages

C'est quoi un package R?

Unité de base de code reproductible en R

Contient:

- Fonctions R réutilisables
- Documentation qui montre comment les utiliser
- Échantillons de données

Pourquoi créer un package?

Partager son code...

... ou pas! On peut sauver du temps en adhérant aux conventions des packages:

- Parce qu'on a un gabarit de structure de code (`R/`, `tests/`, `data/`)
- En adhérant aux conventions, on peut bénéficier d'outils standardisés

Types de packages

- Wrapper d'API
- Données
- Meta-package: Collection de packages
`library("tidyverse")`

Exemples à l'interne: `cllessnverse`, `cllessnhub`,
`locateip`, etc.

Développer un package

Comment créer un package?

`devtools`: package R pour développer des packages :)

- `create_package()`
- Créer une fonction
- `load_all()`
- `check()`
- `use_testthat()`

Comment partager un package?

Publier sur un site comme GitHub

Publier dans CRAN ou autre

Ressources

Créer un package:

- [devtools](#)
- [R Packages](#), par Hadley Wickham et Jennifer Bryan
- [Building Tidy Tools](#), par Charlotte Wickham et Hadley Wickham

Bonnes pratiques:

- [Building a team of internal R packages](#), par Emily Riederer

