

```

---
title: "Pourquoi créer des packages R en recherche?"
author: "Judith Bourque"
date: "2023-06-27"
format:
  revealjs:
    toc: TRUE
editor: visual
---

# 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: `clessnverse`, `clessnhub`, `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``](<https://github.com/r-lib/devtools>)
- [R Packages](<https://r-pkgs.org/>), par Hadley Wickham et Jennifer Bryan
- [Building Tidy Tools](<https://github.com/rstudio-conf-2020/build-tidy-tools>), par Charlotte Wickham et Hadley Wickham

Bonnes pratiques:

- [Building a team of internal R packages](<https://emilyriederer.netlify.app/post/team-of-packages/>), par Emily Riederer