

Good Practice for R Packages

Hannah Frick, PhD

Data Scientist

hfrick@mango-solutions.com

🔰 @hfcfrick





Hello, it's me!

- 🐄 Data scientist @ Mango
- PhD in statistics
- 🗬 R-Ladies
- 📦 psychomix
- 📦 trackeR
- **p**goodpractice



Why write an R package?

One place for

- Code
- Documentation
- Tests
- Data



Why write an R package?

- Easy to distribute
- No need to develop personal conventions
- Standard tools available for standard conventions



Who to write an R package for?

Can be for

- Yourself only
- Your team / internal use
- A client
- The wider public



Resources for writing R packages

Long form

- Hadley Wickham's <u>R Packages</u> book
- Writing R Extensions from CRAN
- <u>rOpenSci onboarding</u>



Resources for writing R packages



Blog posts

- Hilary Parker's <u>Writing an R package</u> from scratch
- Maëlle Salmon's How to develop good R packages

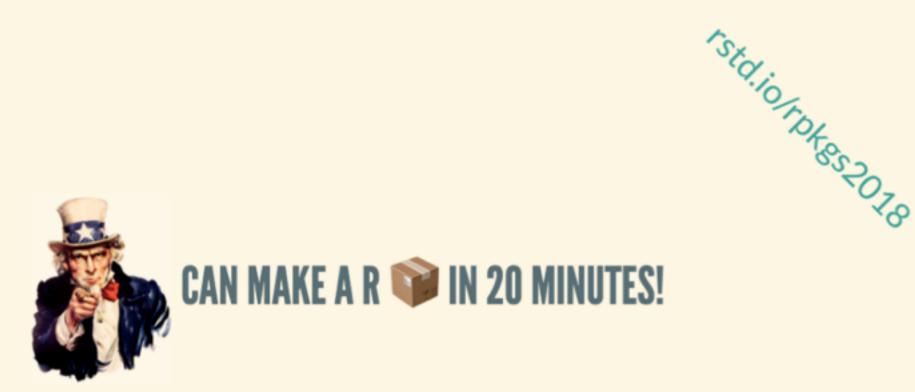


Tools available in R



- usethis::create_package()
 devtools
- devtools::load_all()
- devtools::document()
- devtools::check()





Jim Hester R Studio



Good practice for R packages

Essential

- Make sure CRAN checks pass
- Use tests test (most of) your code
- Use a coding standard for readability
- Keep your functions simple



Good practice for R packages

Bonus points

- Make it user-friendly, e.g., add a vignette
- Keep track of changes via version control
- Use automation
- Make your code public



Standing on the shoulders of...

- CRAN checks: rcmdcheck
- Test coverage: covr
- Source code linting: lintr
- Cyclomatic complexity: cyclocomp

All together in: goodpractice



The **goodpractice** package

- Performs static code analysis
 - goodpractice::gp(path_to_pkg)
- Gives advice on what (not) to do, why and where
- Originally written by Gábor Csárdi
- Used in rOpenSci onboarding process
- Now released to CRAN JM



Use cases for **goodpractice**

Writing a package

- By yourself: learn something new or get a reminder
- In a team: share common standards

Assessing other people's work

Quick review tool



Demo



4 M P2

2

5

40 m

So you got your own ideas?

- Customise gp()
- Choose those checks you agree with
- Add your own checks



The inner workings of gp()

gp () mainly does three steps

- Carry out preparations and add results to the object
- Carry out checks and add results
- Return it all



Writing a custom check

Use make_check() and give it:

- A short description of the check
- The check itself
- The gp advice to be given in case the check fails
- Optional: preps, tags, ...



Writing a custom check

- Add a preparation step with make_prep()
- Refer to this prep step in the definition of the check
- Include all prep steps and check in the call to gp() via extra_preps and extra_checks



Demo



4 M P2

2

5

40 m

Outlook

- Community interest
- Mostly new checks
- RStudio add-in?
- Badge?
- Sticker!





TL;DL

The **goodpractice** package can be used

- To assess and improve R packages
- Just by yourself or in a team



Where to next?





install.packages("goodpractice") Code

https://github.com/MangoTheCat/goodpractice

Se Documentation

https://mangothecat.github.io/goodpractice/



Mango-solutions.com