Wrap up and final discussion

Sophie Schmidt, Clemens Schmid and Petr Pajdla

Recap

Recap

- Adopting package development techniques is good for:
  - future-self
  - the community (reproducibility etc.)
- Don't be afraid to use the *wizards*, they are here to help!
  - `devtools`
  - `usethis`
  - `roxygen2`

The basic R package directory tree

```
.
|--- DESCRIPTION
|--- NAMESPACE
|--- mypackage.Rproj
|--- R
|    |--- myfunction.R
|    |--- myotherfunction.R
|--- man
|    |--- myfunction.Rd
|    |--- myotherfunction.Rd
```

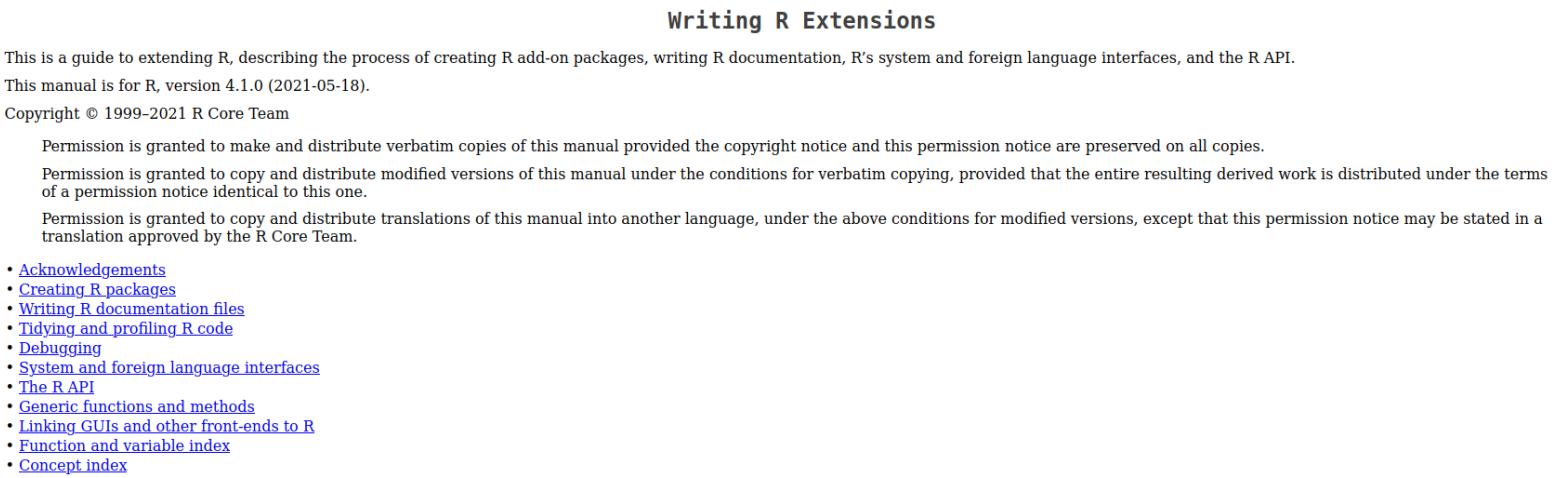
Package structure after fluffing it up

```
.
|--- DESCRIPTION
|--- NAMESPACE
|--- mypackage.Rproj
|--- R
|    |--- myfunction.R
|    |--- myotherfunction.R
|--- man
|    |--- myfunction.Rd
|    |--- myotherfunction.Rd
|--- README.md
|--- LICENSE
|--- LICENSE.md
|--- INST
|    |--- CITATION
```

Going further

Where to learn more I

- Official CRAN **Writing R Extensions** manual
  - https://cran.r-project.org/doc/manuals/R-exts.html
  - very long and throughout



Writing R Extensions

This is a guide to extending R, describing the process of creating R add-on packages, writing R documentation, R's system and foreign language interfaces, and the R API.

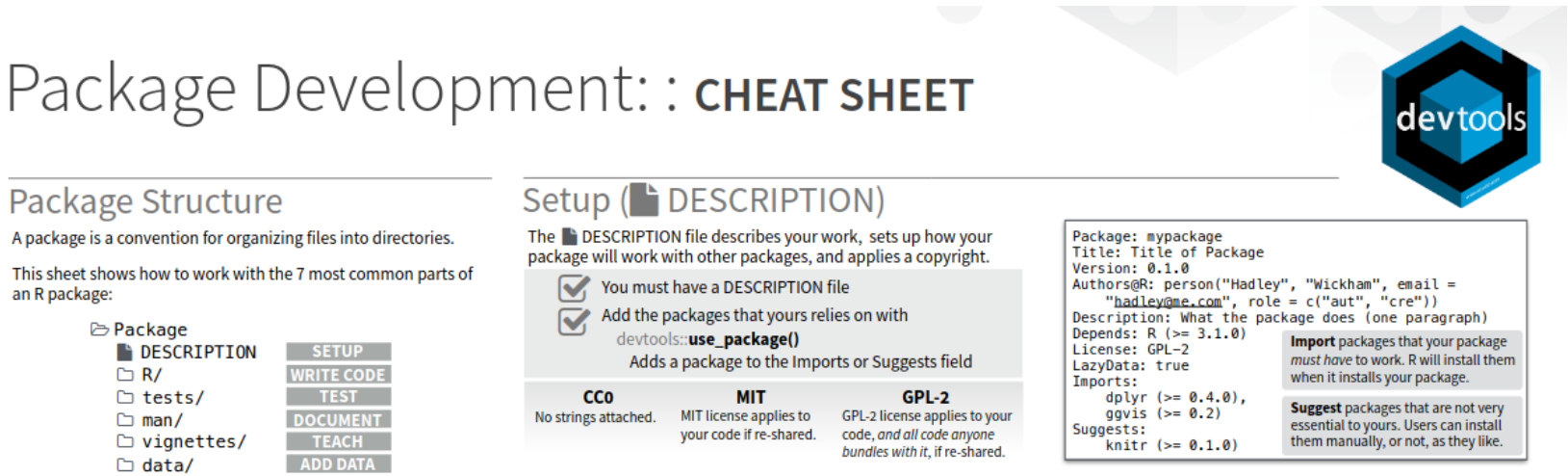
This manual is for R, version 4.1.0 (2021-05-18).

Where to learn more II

- **R Packages** book by Hadley Wickham and Jenny Bryan
  - online version at **https://r-pkgs.org/**

Where to get help?

- Package development **cheat sheet**
  - https://www.rstudio.com/resources/cheatsheets/



Package Development: : CHEAT SHEET

devtools

Package Structure

A package is a convention for organizing files into directories.

This sheet shows how to work with the 7 most common parts of an R package:

Package
- DESCRIPTION — SETUP
- R/ — WRITE CODE
- tests/ — TEST
- man/ — DOCUMENT
- vignettes/ — TEACH
- data/ — ADD DATA

Setup (DESCRIPTION)

The DESCRIPTION file describes your work, sets up how your package will work with other packages, and applies a copyright.

- You must have a DESCRIPTION file
- Add the packages that yours relies on with devtools::**use_package()**
  Adds a package to the Imports or Suggests field

CC0	MIT	GPL-2
No strings attached.	MIT license applies to your code if re-shared.	GPL-2 license applies to your code, *and all code anyone bundles with it*, if re-shared.

```
Package: mypackage
Title: Title of Package
Version: 0.1.0
Authors@R: person("Hadley", "Wickham", email =
    "hadley@me.com", role = c("aut", "cre"))
Description: What the package does (one paragraph)
Depends: R (>= 3.1.0)
License: GPL-2
LazyData: true
Imports:
    dplyr (>= 0.4.0),
    ggvis (>= 0.2)
Suggests:
    knitr (>= 0.1.0)
```

Import packages that your package *must have* to work. R will install them when it installs your package.

Suggest packages that are not very essential to yours. Users can install them manually, or not, as they like.

- and of course **help files** and package **vignettes**. . .

Questions and general discussion

Don't forget to check out session

- **S17 Tools for the Revolution:**

  Developing packages for scientific programming in archaeology

  - organized by Joe Roe, Martin Hinz and Clemens Schmid
  - Wednesday, **June 16th 10:50 - 14:20** EET

For materials from the workshop, see

https://github.com/sslarch/caa2021_Rpackage_workshop

Please give us some feedback!

https://forms.gle/MzTsMntBxtCCmge1A

Thank you for attending the workshop!

Sophie, Clemens and Petr