Reading Group: R package

LEUNG Man Fung, Heman

Winter 2020

Preliminaries

- Before developing a R package, I recommend additional knowledge of
  - functions, e.g., scoping, … (dot-dot-dot), infix form; see Advanced R Ch6
  - environments; see Advanced R Ch7
  - signals, .e.g., `stop()`, `warning()`; see Advanced R Ch8
  - at least one OO system, e.g., S3; see Advanced R Ch13
- Reference book: R packages by Hadley Wickham and Jenny Bryan
  - read for details as I can only cover working knowledge
  - vs Writing R Extensions
    - the later is official but more difficult to read
    - `package.skeleton()` is also less developer friendly than R studio project
- I will assume we have some R functions ready for bundling into a package

Workflows

1. Name our package (nicely)
   - avoid name collision, e.g., `available::available("rpkg", browse=F)`
2. Create directory via File > New Project… > New Directory > R package
3. Edit the DESCRIPTION file
4. Put the ready R scripts in R/
   - document these scripts with roxygen2, e.g., title, description, @param, @examples, @return
   - put examples in separate files if they are long and use @example
   - (optional) document the package as a whole
5. Prepare unit tests, e.g., `usethis::use_testthat()`
6. Check the package, e.g., `devtools::check()`
7. Write vignettes when our package is relatively stable

Q&A

- How to build a pdf with all available functions' help manual?
  - Use `devtools::build_manual()`.