The R Statistical Computing Environment
Basics and Beyond
Writing and Building R Packages

John Fox

McMaster University

ICPSR/Berkeley 2016

Building R Packages

- To check a package, assuming that the R bin directory is on the Windows path, and that the package subdirectory is in the current directory:
  `R CMD check` *package-name*
- To check a package intended for CRAN:
  `R CMD check -as-cran -run-donttest` *package-name*
- To build the package, producing a `tar.gz` file:
  `R CMD build -force` *package-name*
- To build a Windows binary package, producing a `.zip` file:
  `R CMD INSTALL -binary` *package-name-version*`.tar.gz`
- To install source package:
  `R CMD INSTALL` *package-name-version*`.tar.gz`

Building R Packages

- The procedure is similar under Mac OS X and Linux/Unix.
- There is convenient access to these tools from inside RStudio, though to build vignettes (see below), you'll want the **devtools** package.
  - To install a package with vignettes:
    `devtools::install(build_vignettes=TRUE)`
  - To build a package with vignettes:
    `devtools::build(vignettes=TRUE)`

Building R Packages
Package Structure

- Package structure is described in detail in the R manual *Writing R Extension* (R Core Team, 2015).
- An R source package consists of a directory containing:
  - A `DESCRIPTION` file with meta information such as the package name, version, and author; how vignettes are to be processed; and other packages on which the package depends.
  - A `NAMESPACE` file, enumerating, e.g., the objects that are "imported" into and "exported" from the package.
  - An R subdirectory containing `.R` files with R code for creating objects such as functions.
  - A `man` ("manual") subdirectory that includes `.Rd` documentation files (using LaTeX-like markup). All public objects in the package should be documented.
  - A `data` subdirectory containing data objects, such as text or `.RData` files that can be read as R data frames. Thus, a file named `Duncan.txt` or `Duncan.RData` would normally produce a data frame named `Duncan`.

Building R Packages
Package Structure

- Source package directory structure (continued):
    - Possibly an `inst` ("install") subdirectory containing arbitrary files and subdirectories to be installed in the package.
    - An optional `vignettes` subdirectory, normally containing Sweave (`.Rnw`) or RMarkdown (`.Rmd`) files to be compiled into extended documentation, typically in the form of PDF or HTML files.
    - Possibly other subdirectories (e.g., for compiled C code).

Building R Packages

- The function `package.skeleton` creates the "skeleton" of a source package for objects that are currently in the R workspace.
- Alternatively Hadley Wickham's **devtools**, **roxygen2**, and **testthat** packages provide integrated tools for building, documenting, and testing packages (see Wickham, 2015).
- Example: The **matrixDemos** package.

References

R Core Team (2015). *Writing R Extensions*. version 3.2.3, https://cran.r-project.org/doc/manuals/r-release/R-exts.pdf.

Wickham, H. (2015). *R Packages*. O'Reilly, Sebastopol CA. http://r-pkgs.had.co.nz/.