Package webs: Reproducible results from raw data

Kirill Müller

Institute for Transport Planning and Systems (IVT), ETH Zurich; kirill.mueller@ivt.baug.ethz.ch

1 Introduction

For reproducible research, it is crucial to be able to generate all results from original raw data. By automating the process, it is possible to easily verify reproducibility at any stage during the analysis. Automation also allows easy recreation of the entire analysis based on modified inputs or model assumptions. However, rerunning the entire analysis starting from raw data soon becomes too time-consuming for interactive use. Caching intermediate results alleviates this problem but requires a robust mechanism for cache invalidation.

R packages are a suitable container for statistical analyses: They can store data, code, and documentation. Recent efforts have considerably simplified the packaging process. This poster presents an approach to conduct a statistical analysis by creating a **package web** – interdependent packages where each serves a dedicated purpose. Package dependencies define the data flow for the entire analysis. The rpkgweb companion package tracks which downstream packages need to be rebuilt if a package changes, and builds independent packages in parallel. Reproducibility can be monitored continuously with minimal effort, yet the modular structure permits interactive work.

2 Setup

Each step is a package

- R packages contain all the code, data and text used for the analysis
- Packages are rather small, each serves a dedicated purpose:
  - configuration
  - holding raw data
  - input validation, munging data
  - modeling
  - analysis, reporting

  ...

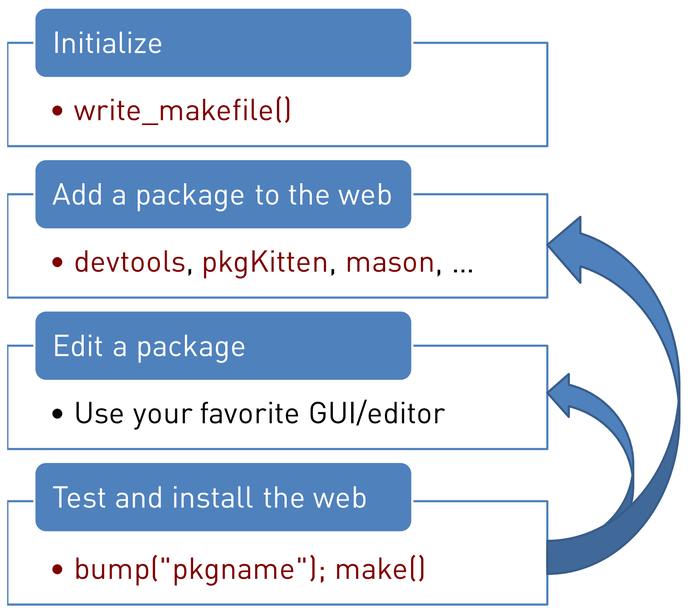
Two principal operations

1. Test
   - Verify code correctness
   - Create data
   - Build vignettes
2. Install
   - Make available downstream

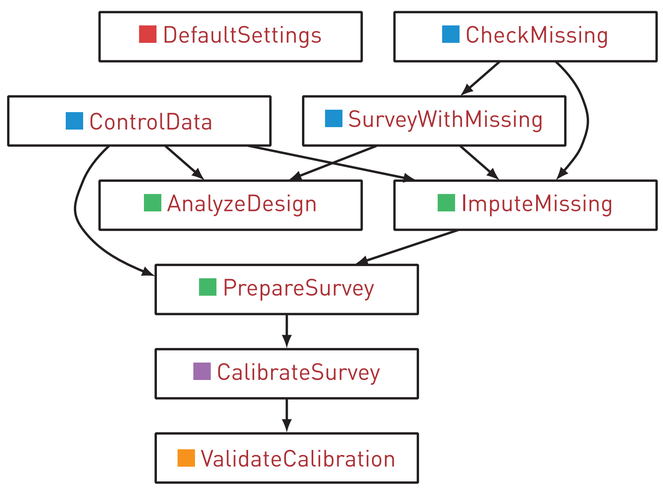
Updating a package

- Work on the package
- Test
- Install
- Update downstream dependencies

3 Workflow



4 Example: Calibrating a survey



5 Design goals and implementation

Usability Packages are "first class citizens" in the R ecosystem. A package web is simply a directory with several R packages. Helper functions perform "housekeeping" tasks unrelated to the analysis.

Robustness and encapsulation Each package accesses only the data (and exported code) of its dependencies.

Explicit data flow through package dependencies.

Reproducibility and automation The entire analysis is run or updated with a single command. When modifying a package, only dependent packages are rebuilt. This allows simple integration with build automation systems such as Jenkins.

Parallel processing Mutually independent packages can be built in parallel. A Makefile is created from the package dependencies, the make utility schedules and executes the tasks.

Interactive processing At the package level, all required data for a step are ready to use. Loading the data from scratch is fast and seamless, no need to save and reload sessions.

Scalability The modular structure allows implementing complex processes. A data package can also access and process large external bodies of data.

Caching Semi-automatic, invalidated explicitly.

Parametrization via multiple package libraries.

Also useful for package development!

install_github(c("MakefileR", "rpkgweb"), "krlmlr")

6 Related work: Comparison to ProjectTemplate

	Path	Purpose	In a package web
■	config/	Configuration	Create a *default* package from which all packages depend
■	data/	Raw data	Create a *raw data* package with data in data-raw/
■	cache/	Munged data	Create a *cache* package with data in data-raw/
■■■■■	lib/	Helper functions	Add to R/ in the package where it fits best
■	munge/	Munging scripts	Add to R/ in the corresponding *cache* package
■■	src/	Analysis scripts	Create an *analysis* package with code in R/
■	diagnostics/	Input validation	Add a test in the *raw data* package to tests/testthat/
■■■■	tests/	Tests	Add test to tests/testthat/ in the corresponding package
■■■■■	doc/	Documentation	Add roxygen2 inline documentation to code and data
■	reports/	Output	Create an *analysis* package with a vignette in vignettes/
■■	graphs/	Plots	Part of a vignette or cached data
	logs/	Log files	—
■	profiling/	Benchmarking	Store profiling data in a *benchmark* package

7 References

Bravington, M. V. (2013). *mvbutils: Workspace organization, code and documentation editing, package prep and editing, etc.*

Csardi, G. (2015). *mason: Friendly Craftsman Who Builds Slick R Packages*.

Eddelbuettel, D. (2015). *pkgKitten: Create Simple Packages Which Do not Upset R Package Checks*.

Flight, R. M. (2014). Analyses as packages. Blog: *Deciphering life: One bit at a time*.

Müller, K. (2015). *MakefileR: Create Makefiles using R*.

Rudolph, K. (2015). *modules: Modules for R*.

Ushey, K., J. McPherson, J. Cheng, and J. Allaire (2015). *packrat: A Dependency Management System for Projects and their R Package Dependencies*.

White, J. M. (2014). *ProjectTemplate: Automates the creation of new statistical analysis projects*.

Wickham, H. (2011). testthat: Get started with testing. *The R Journal 3*, 5–10.

Wickham, H. and W. Chang (2015). *devtools: Tools to Make Developing R Packages Easier*.

Wickham, H., P. Danenberg, and M. Eugster (2015). *roxygen2: In-Source Documentation for R*.

Institut für Verkehrsplanung und Transportsysteme
Institute for Transport Planning and Systems

+41 44 633 31 05
info@ivt.baug.ethz.ch
www.ivt.ethz.ch