Efficient data analysis workflow in *R*

Peter Baker[1]

1. Statistical Consultant & Senior Lecturer, School of Population Health, University of Queensland, Herston, QLD. Australia
*Contact author: p.baker1@uq.edu.au

Keywords: workflow, unix tools, reproducible research, project templates, data handling

After the fifth large data set came through my door in as many months, I thought it would be more efficient to automate workflow rather than start afresh each time. For instance, I knew my collaborators and clients were likely to *tweak* their data a number of times because even though they had provided data with hundreds or even thousands of variables, they'd undoubtedly left out a few important variables or discovered that some were coded wrongly or recorded inconsistently. And of course, I also thought that my collaborators were likely to contact me a couple of days before a grant application or final report was due with a revised data set and new questions in the hope that I could just *press the button* to instantly extract some final answers.

About the same time, I noticed a few interesting posts on R–bloggers and stackoverflow. I particularly liked the software engineering term **DRY** (don't repeat yourself) with suggestions about automating processing with *R* functions or packages. Another post referred to Long (2009) which provides a useful guide to managing workflow for data analysis in large projects. Long's ideas revolve around using *stata* in a `Windows` environment in order to efficiently facilitate replication of work by following a cycle of Planning, Organising, Computing and Documenting. *stata* has some useful features like using variable "labels" in plots and tables (unlike standard *R*), datasignatures and Long provides good strategies for using codebooks for data handling and checking. However, the approach concentrates on manual methods rather than programming tools like *make*, automatic initial data processing, regular expressions for text processing or version control.

Many tools are available for efficiently managing projects and carrying out routine programming tasks. One such tool is *GNU make*. It is standard on `linux` and `MacOSX` and available via `Rtools` or `cygwin` for `Windows`. Originally developed for programming languages like *C* it is well suited to statistical analysis. Since the late 80's I've used *make* to project manage data analysis using *GENSTAT*, *SAS*, *R* and other statistical packages. It is very efficient in only re–running analyses or producing reports when dependencies such as program files (*R*, *Rnw*, . . .) or data files change. *R* is used for all data analysis steps described below.

Using the *R* **ProjectTemplate** as a starting point, the following will be outlined:

- the set up of project directories, Makefiles, R program files to read and check data, initial documentation and log files for use with *emacs* `Org-mode` or other editor;
- using codebook(s) to label variables, set up factors with suitable labels;
- producing initial data summaries and plots for data checking and exploration;
- setting up an initial *git* repository for version control; and
- producing initial summaries using literate programming via `Sweave`.

An *R* package which automates the steps above is under development.

References

Kuhn, M. (2011). ReproducibleResearch cran task view. `http://www.r-project.org/`.

Long, J. S. (2009, February). *The Workflow of Data Analysis Using Stata* (1 ed.). Stata Press.

White, J. M. (2010). *ProjectTemplate: Automates the creation of new statistical analysis projects.* R package version 0.1-3.