

Packrat - A Dependency Management System for R

Kevin Ushey^{1,*}

1. RStudio, Inc

*Contact author: kevin@rstudio.com

Keywords: reproducible research, dependency management, project management

Dependency management in *R* is difficult. Different *R* projects can have different dependencies, and can often depend on different versions of the same *R* packages. The suite of *R* packages served by CRAN and BioConductor is constantly evolving and growing, and while *R core* and package authors make large efforts to maintain backwards compatibility, it is not guaranteed as *R* and its packages evolve.

There has been a lot of discussion as to how the *R* project, alongside the CRAN repository, could be augmented to support better versioning in projects (Ooms 2013). **packrat** uses a form of **local versioned package management**, to ensure a project and its versioned dependencies are coupled together – similar to JavaScript’s *node.js* and the packages on its associated repository *NPM*.

As a result, **packrat** helps the user by isolating dependencies within a project, ensuring that they do not conflict with package requirements in other projects. In addition, package sources are recorded, so that packages can be easily upgraded and rolled back, using the archives available on CRAN and BioConductor, or to local package sources packaged alongside the project. Furthermore, users collaborating on a project can use **packrat** to ensure that their *R* environments are compatible, hence avoiding compatibility problems in collaborative projects.

packrat helps solve the following problems:

- **Local Dependency Management:** Because each **packrat** project uses its own private library, dependencies are effectively isolated from other projects, and so versioning conflicts can be controlled and avoided. A user can *bootstrap* a project to infer and set up the local library the project requires, and with later modifications to this local library, the user can *snapshot* and save the current library state, or *restore* and roll back to the last *snapshot* state.
- **Portability:** **packrat** makes it easy to *bundle* a project for sharing. A *bundled* project can easily be *unbundled* to restore the same *R* environment that was originally used in the project, even across different platforms.
- **Reproducibility:** **packrat** records the exact package versions a project depends on, and ensures those exact versions are the ones that get installed wherever the **packrat** project is used.

Replication is the ultimate standard by which scientific claims are judged. (Peng 2011)

In this talk, we will outline a number of common usage scenarios with **packrat**, and demonstrate how it can be used to control and manage dependencies within your project.

References

- Ooms, Jeroen. 2013. “Possible Directions for Improving Dependency Versioning in R.” *CoRR* abs/1303.2140.
- Peng, Roger D. 2011. “Reproducible Research in Computational Science.” *Science* 334 (6060): 1226–27. doi:[10.1126/science.1213847](https://doi.org/10.1126/science.1213847). <http://www.sciencemag.org/content/334/6060/1226.abstract>.