



Using renv & RStudio Package Manager for Clinical Reproducibility

Mike Stackhouse & Phil Bowsher March 8, 2023



>Intro

- 2010-2019
- Phuse 2019
- Basic R setup has no built-in functionality to store the package versions & dependencies
- CRAN, out of the box, uses the latest of everything
- Default to overwrite the existing versions of packages
- packrat
- Submissions List
- mran
- Posit Validate Your Environment



>In this talk

- Challenge - R Package Management so analyses will be reproducible in the future
- Solution - Open Source Strategy with managed Repos & reproducible environments
- Impact, Best Practices & Next Steps



>Challenge

- CRAN, updated on a daily basis
- Session info, not actionable
- Focus is on easy & natural, replicating normal engagement with R for different use-cases
- Support Collaboration - What do I need to install?
- Balance “conservative” & “excited” package environments

“The problem comes when imposing my workflow onto collaborators. Some collaborators, understandably, want the latest & greatest of everything, & certainly do not want to overwrite any previously installed packages with out-of-date versions. &, if I want to reproduce other people’s work, then I end up having to install versions they used, which ends up polluting my package library with “unsupported” (by me) versions.”



➤Posit Package Manager

- Host all of Cran or curated CRAN sources
- Support Open Source: R, python, Bioconductor etc.
- Versioned package repositories (Snapshots), copy of CRAN from a specific date
- Enables time-travelling across different package versions
- Process to set the appropriate repositories depending on the active version of R
- Track package downloads
- Allows for multiple repos for sharing one Validated source with snapshots or Multiple Validated repos for each version of R



>Support for Admins & Users -

- Implement Cran, Shared Baseline & Validated Strategies in one place & Upgrades
- List system dependencies for each R package
- Provide installation commands
- System dependencies vary between Linux distributions & releases
- Package versions available as fast installing pre-compiled binaries
- Users use `install.packages()` as per usual
- Automatically building internal packages from Git
- Validated Repo & [Here](#)

“The first part of the solution was to begin using (semi-)stable repositories”



> Repositories in Pharma/Healthcare

- Johnson
- Roche
- Merck & here

“Allows Admins/users to Lock down their R package versions using PPM snapshots”



>renv

- Reproducibility of analysis is one of the most important aspects of regulatory deliverables
- R looks for packages in Sys, User & Proj locations
- renv will Put the library inside the project
- “we use renv to lock R package versions & save them in the renv.lock file”
- renv package is compatible with CRAN, github, Bioconductor, RStudio package manager etc.)
- Integrates with pipenv & reticulate for multilingual projects
- Discover dependencies

“This package strikes a great balance between user-friendliness & robustness...renv does not require the users to change their workflow of installing/removing & using packages, & requires little technical skill to setup.”



renv for users

- User project gets set up with isolated package libraries
- Record of repositories used & capture package versions
- Record of “my” repositories & exact package versions used (down to the commit hash)
- JSON formatted file called `renv.lock` that can be versioned with git
- R warning is issued on startup if mismatch
- Prevents polluting the system library
- Packages are installed by default to a user cache
- Orgs can set up a global cache so each user does not create a complete local cache

>renv Video

“One solution is to use software that automatically creates project-specific local libraries. These libraries essentially package software versions along with the relevant code so that the original analysis can be reproduced even after software is updated (44, 45). Such workflows also facilitate returning to one’s own code in the future.”

RStudio Solutions Cast: Safely



Docker != Reproducibility

- Attributes of containers that make them useful for a submission
- Lock down your working environment using a Docker image
- Development within Docker containers
- Accompany projects with a Docker image
- Projects, Git, renv

“asking my collaborators to use docker feels like asking them to travel to my project which is currently hosted on the moon.”

>Posit Package Manager & renv

- Balance responsibility between admin & user
- Help Docker builds be reproducible & here & here
- Dockerfiles with “install.packages” gets latest package as of time the image is rebuilt
- Docker combined with renv helps control the package versions precisely

“While renv & RSPM certainly can work independently of another, they work very well as complements. One reason is that you need simply set the base Posit Package Manager repo URL (e.g. `options(repos = c(RSPM = "https://packagemanager.rstudio.com/all/latest"))`) & renv will automatically transform the URL so that it fetches the binaries appropriate for your platform (OS, distro, etc.), or whoever else you have shared your lockfile with.”



