Setting up rrtools R Data Analysis Project Framework

Ronald G. Thomas

2025-07-01

Table of contents

1 Executive Summary

Setting up a working repo can be time consuming and error-prone. This document presents a solution to this problem by providing a step-by-step guide to creating a reproducible R project using the rrtools package. The guide includes instructions for generating a project structure, creating a README file, And generating a Dockerfile for containerization. The goal is to provide a

2 Motivation

Imagine you've written code that you want to share with a colleague. At first glance, this may seem like a straightforward task—simply send the R files via email. However, ensuring that your colleague can run the code without errors and obtain the same results is often much more challenging than anticipated.

When sharing R code, several potential problems can arise:

- Different versions of R installed on each machine
- Mismatched R package versions
- Missing system dependencies (like pandoc or LaTeX)
- Missing supplemental files referenced by the program (bibliography files, LaTeX preambles, datasets, images)
- Different R startup configurations (.Rprofile or .Renviron)

A real-world scenario often unfolds like this:

1. You email your R Markdown file to your colleague, Joe
2. Joe attempts to run it with `R -e "source('peng1.Rmd')"`
3. R isn't installed on Joe's system
4. After installing R, Joe gets an error: "could not find function 'render' "
5. Joe installs the rmarkdown package
6. Now pandoc is missing
7. After installing pandoc, a required package is missing
8. After installing the package, external files are missing (bibliography, images)
9. And so on…

3 Conclusion

Achieving full reproducibility in R requires addressing both package dependencies and system-level consistency, while ensuring code quality through testing. By combining renv for R package management, Docker for environment containerization, and automated testing for code validation, data scientists and researchers can create truly portable, reproducible, and reliable workflows.

The comprehensive approach presented in this white paper ensures that the common frustration of "it works on my machine" becomes a thing of the past. Instead, R Markdown projects become easy to share and fully reproducible. A collaborator or reviewer can launch the Docker container and get identical results, without worrying about package versions or system setup.

The case study demonstrates how two developers can effectively collaborate on an analysis while maintaining reproducibility and code quality throughout the project lifecycle. By integrating testing into the workflow, the team can be confident that their analysis is not only reproducible but also correct.

This strategy represents a best practice for long-term reproducibility in R, meeting the high standards required for professional data science and research documentation. By adopting this comprehensive approach, the R community can make significant strides toward the goal of fully reproducible and reliable research and analysis.

4 References

1. Thomas, R.G. "Docker and renv strategy."
2. "Palmer Penguins Analysis."
3. The Rocker Project. https://www.rocker-project.org/
4. renv documentation. https://rstudio.github.io/renv/

5. testthat documentation. https://testthat.r-lib.org/
6. Horst, A.M., Hill, A.P., & Gorman, K.B. (2022). palmerpenguins: Palmer Archipelago (Antarctica) penguin data. R Journal.

---

5 Prerequisites

In development

6 Step-by-Step Implementation

In development

7 Key Takeaways

In development

8 Further Reading

In development

Session Summary

In this session, we worked on refactoring a shell script to convert a generic R research project to follow the rrtools framework for reproducible research. Here's what we accomplished:

Main Achievements

1. Analyzed the existing adjust_to_rtools.sh script that was designed for an MCI analysis project
2. Refactored the script to work with generic research repositories like x19
3. Enhanced the script to align more closely with Ben Marwick's rrtools framework
4. Added proper directory structure, documentation templates, and reproducibility features

Key Script Developed

The main deliverable was a comprehensive shell script that: - Creates the standard rrtools directory structure - Organizes R functions, analysis scripts, and Rmd files - Sets up proper testing infrastructure - Creates documentation files (README, CONTRIBUTING, CONDUCT) - Configures Docker and Binder support - Establishes dependency management with renv - Creates symbolic links for easier navigation

Notable Features

- Backs up the original project structure before making changes
- Detects package name from DESCRIPTION file
- Intelligently sorts existing files into appropriate directories
- Creates templates for R Markdown manuscripts using bookdown
- Sets up Docker containers for reproducibility
- Configures Binder for browser-based analysis
- Adds comprehensive documentation following rrtools standards

This refactored script provides a solid foundation for converting existing R research projects to follow best practices for reproducible research using the rrtools framework.