PharmaSUG 2021 - Paper AP-204

R Reproducibility Best Practices for Pharma

Phil Bowsher, RStudio PBC;
Sean Lopp, RStudio PBC;
Alex Gold, RStudio PBC

Abstract

Reproducibility of data analyses is essential in clinical work, but achieving it can be challenging in any tool or language. Moreover, reproducibility is a spectrum, with greater degrees of reproducibility requiring more investment from users. As these efforts advance, organizational, IT and user investments are critical. This paper will introduce some simple tools that can be used with clinical R workflows to help establish a foundation. The audience for this paper are those that are new to reproducibility in R and want to take the first steps towards reproducibility of their work. This paper will not discuss IT tools like Docker that are also important in this space.

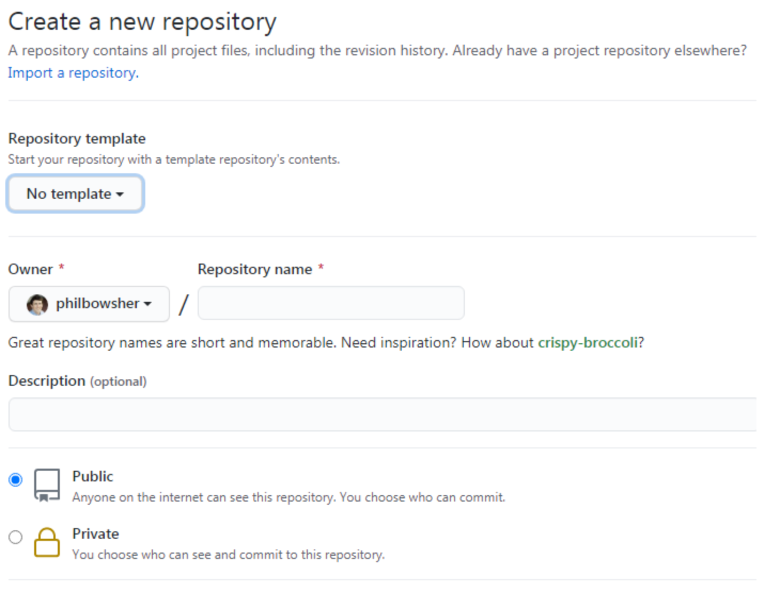
Introduction

Many people start with R in college and often lack an understanding of reproducibility as the workflows do not require it. This paper will encourage you to think about reproducibility early in your development and the productization of R code. This paper will introduce tools that will help connect these spaces - Development and Production. Below we will look at some user steps to follow when starting your R analysis.

Step 1 - Use Version Control

Version control is especially important when collaborating with other statistical programmers. Git is very popular in the software development space and GitHub is the most common interface. Many organizations have accounts at GitHub.com, and many others host their own git repositories using GitHub or other similar tools like GitLab or Bitbucket.

Public repositories and limited private repositories are free at GitHub.com. To start this process, head over to GitHub.com and create a repository:



Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? Import a repository.

Repository template

Start your repository with a template repository's contents.

No template

Owner * philbowsher / Repository name *

Great repository names are short and memorable. Need inspiration? How about crispy-broccoli?

Description (optional)

[✓ Public] Anyone on the internet can see this repository. You choose who can commit.

Private You choose who can see and commit to this repository.

The GitHub repository is going to house your R files created and modified in RStudio. After the repository has been created, users can connect to it in RStudio by creating a RStudio Project. RStudio projects will be covered in Step 2 below.

In addition to the central GitHub repository, you will need git where you plan to do your analysis (RStudio either locally or on a server). Installing git can either be an IT or user effort. Below are resources for installing Git in these environments.

https://support.rstudio.com/hc/en-us/articles/200532077-Version-Control-with-Git-and-SVN

https://happygitwithr.com/rstudio-git-GitHub.html

For most users, the majority of git functionality will be enabled via RStudio. This is great for users that are committing and pushing changes to a repository. For more complex tasks, users may occasionally want to interact directly with git from the command line. The RStudio Shell provides a way to do this. Below are articles that describe this task:

https://happygitwithr.com/shell.html

https://support.rstudio.com/hc/en-us/articles/115010737148-Using-the-RStudio-Terminal

Step 2 - Use RStudio Projects

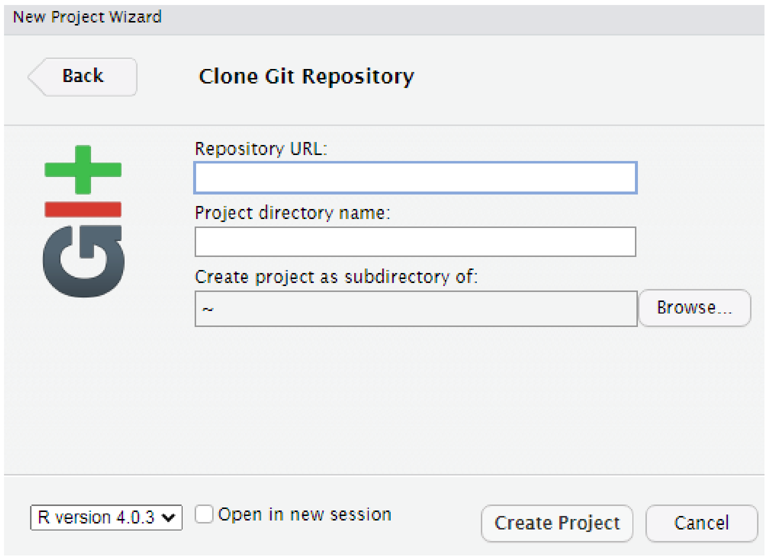
RStudio projects give you a solid workflow that will serve you well in the future. A project in RStudio is a collection of work ***organized*** in a folder. We would recommend that you always start your work as a RStudio Project. The links below describe how to create a Project in RStudio:

https://r4ds.had.co.nz/workflow-projects.html

https://support.rstudio.com/hc/en-us/articles/200526207-Using-Projects

You can connect a new RStudio project to your GitHub repository by going to:

File -> New Project -> Version Control -> Git ->



Repository URL:

Project directory name:

Create project as subdirectory of: ~

R version 4.0.3

Open in new session

Add the GitHub repo link to the Repository URL. After you click Create Project, you will now have RStudio connected to your GitHub repository. Please note that the server and or computer will require Git for this to work. If you need to install Git, please see the links in step 1.

RStudio projects are designed to be self-contained units of work. It is often convenient to reference files within the project relative to the project root, because then you know that the file will be in that location regardless of where the project itself lives. The `here::here()` command in R is a convenient way to get the directory of the project root.

Step 3 - Use renv

renv is a toolkit used to create and manage project-specific libraries of R packages. Using renv means that your local package environment (system and/or user library) will be insulated from work you're doing on any specific project, and that your project will be safe from changes to your local package library. This makes your project library self-contained and portable.

The project will start off with a clean slate and you will install your needed packages. As you do analysis, renv will detect what packages you are using and save the information to a lock file, which other people can use to re-create the package state.

It is not recommended that you install packages from CRAN for clinical work, but rather, install packages from an IT repository that is aligned with supported R versions.

To connect your newly created project above to renv, you will need to run:

renv::init()

This will initialize a new project-local environment with a private R library.

The statistical programmer then creates a renv.lock file using **renv::snapshot()**. The lock file is committed alongside the application code into a version control system like Git.

If a colleague is going to also collaborate on the analysis code, that can be done by having the collaborator cloning the Git repository into a new project with the included renv.lock file.

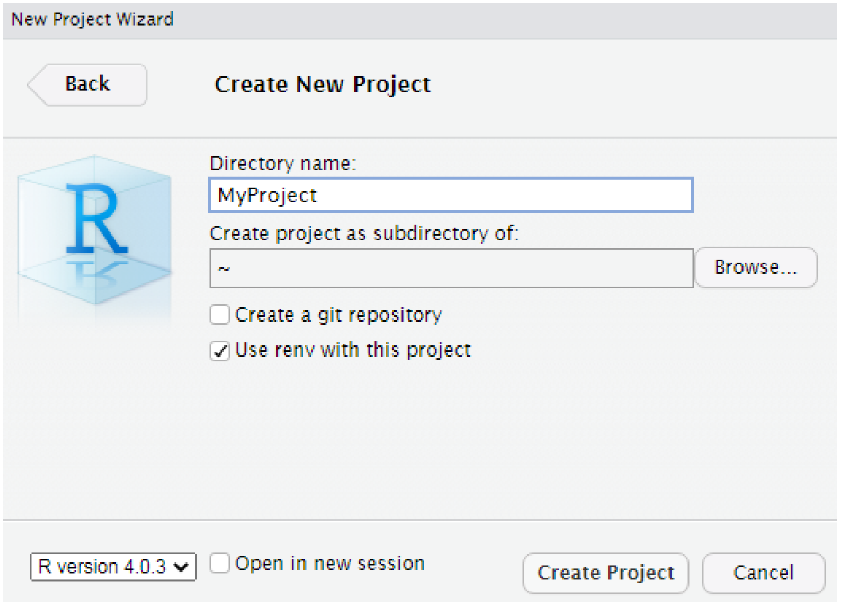
The collaborator can use **renv::restore()** to recreate the necessary package environment for the project. The renv environment can be activated with **renv::activate()**.

You can read more here:

https://rstudio.GitHub.io/renv/articles/collaborating.html

https://environments.rstudio.com/collaborate.html

Alternatively, you could initialize your project above as a renv project like this:



Directory name: MyProject

Create project as subdirectory of: ~

☐ Create a git repository

✓ Use renv with this project

R version 4.0.3

☐ Open in new session

If you go this route, be sure to check the box titled "Create Git repository". Once the new project is created, you can link it to GitHub following these instructions: https://happygitwithr.com/existing-github-last.html

Contact & Summary

The information above highlights simple approaches to reproducibility that can have big improvements for statistical programmers. I would argue that it is a best practice to always use RStudio Projects, versioning, and renv for work that needs to be reproducible. Moreover, these tools have added benefits of supporting further downstream processes like deploying your code to production. These well established tools in the R ecosystem can help modernize clinical processes that are starting to incorporate more software engineering practices. The authors believe that reproducibility will be even more systematic in the RStudio IDE in future years.

Phil Bowsher

phil@rstudio.com

https://GitHub.com/philbowsher