Reproducible Research Resource Compilation

Margaret Janiczek

2022-10-27

Background

This is a compilation of resources (and opinions on project organization) I find useful in producing reproducible research, particularly for researchers using RStudio and GitHub for collaborative statistical analysis.

I am very open to additions or suggestions, if you would like to do so please clone this repository (https://github.com/mljaniczek/repro-resource), make your changes, and submit a pull request! (That could also be a good way to practice navigating GitHub if you are new to it!)

This guide assumes familiarity with R, Rstudio, and RMarkdown.

Project Organization and Project Environment

- Organize each analysis project separately. Create folders for code, references, data. Have a README for the project that you keep updated with important project meetings, decisions, data changes etc.
- Be intentional with file names.
- Highly recommending utilizing .rproj in all project folders. More on that here

Managing data from collaborators

- Avoid manually going in and changing excel sheets yourself. You should keep received data as is, and only make changes (like data cleaning, filtering) in your code, in such a way that someone could take over your project from square 1 and understand how you came up with your finalized analysis dataset!
- In cases where your collaborator updates data over time, clearly note the date and location for updated data.
- Have a folder for "raw" data (direct from collaborators, don't even change the file name, don't change the file at all). If necessary have folders with dates inside the raw folder, if your collaborator tends to send you updated data over time.
- Then also have a folder for "processed" data. This should be where you save your processed .Rdata file that is ready for analysis. You can consider having date folders in here as well.

Analysis files

- Consider having ONE script or .Rmd file for data processing, which runs all cleaning and filtering, labeling and variable transformations, and saves a final processed dataset that you can load into your other analysis scripts. Great if you can save the data with `Sys.Date()` in the name, so as time goes on if you have to update your processing or dataset, then you can easily refer back to old data versions.

- Have one .Rmd script for each analysis in the project. As the analysis changes, add to the script and then use version control software (Git - more on this below) to avoid having confusing duplicate-ish files.

Keeping track of packages

- We've all been there... you come back to a project after a few months (or years) and the code is broken because of different package versions. How to avoid this? Consider using the `renv` package!
  - Essentially it maintains a record AND environment of what packages and versions your project depends on, so when you come back to a project it should still run smoothly.
  - You can see Keving Ushey's presentation about `renv` from the 2020 RStudio Conference here
- If you don't do this route, at the very least you should keep a record of what packages and versions your project depends on (maybe updating the list each time you share a major report/update with collaborator)

Version control with Git/Github

- This is essential for backing up projects, sharing them with collaborators, and keeping a record for yourself of major changes in your research (without having to have files with names like "Analysis1_final.Rmd", "Analysis2_final_final.Rmd", "Analysis3_return_of_the_collaborator_final.Rmd")
- Instead, you can/should have ONE data processing file, and one .Rmd for each distinct analysis, as well as an Rscript that you can load which contains custom helper functions if you have them. If you keep on top of pushing the updates to Git/Github and clearly label the changes, this should both keep a clean set of files and also keep a record of the changes in your analysis over time (as well as an ability to go "back in time" and restore an old version of the analysis if need-be).
- I HIGHLY recommend Jenny Bryan's "Happy Git and GitHub for the useR" resource. It helped me get up and running and I still refer to it now and then.

Reproducible Reports with RMarkdown

- It is tempting to run some quick code in a script, screenshot a figure or results from the console, paste it in a word document and send it to a collaborator. But what happens months from now when you need to write up your analysis?? "How did I make this figure, what data did I use, why can't I do it again??"
- This is why it is a great idea to make reproducible analysis reports from the start of your project, no matter how small the analysis at first!
- Here's the full RMarkdown Guide which I refer to often: https://bookdown.org/yihui/rmarkdown/
- More on reproducibility with RMarkdown here: https://bioconnector.github.io/workshops/r-rmarkdown.html

Consider making your analysis a package

- I heard this concept a few years back and it brings everything I've said here together! Essentially if you structure your reproducible analysis project in just a *slightly* different way than listed above, you can fairly easily create your very own R package for the analysis, which would make it all the easier for collaborators/journal editors/future readers of your research be able to reproduce your findings.
- You can read more about the concept here or here

- Note: I know there was an Rstudio conference presentation on this concept a few years back but I can't remember who did it and also can't find the slides! If anyone finds it please let me know and I will add it here.

Resources

- This course is a great detailed overview on Reproducible Research in R: https://r-cubed.rostools.org/index.html
- Another very detailed reporducibility in R guide: https://monashdatafluency.github.io/r-rep-res/
- Nice short overview on RProject organization: https://www.rforecology.com/post/organizing-your-r-studio-projects/