Reproducible Workflows for Small Data Teams

Using RStudio, Rmarkdown and Github

Dr. Karin Neff

karin.neff@bsd7.org github.com/karinneff

The Plan

- Introductions
- What
- Why
- How
- Tricks
- Project

What

- Reproducible
  - Not just for research
  - Repeated tasks
  - Variations on a theme
- Workflow
  - Plan
  - Documentation
  - Organization
  - Reports

Why

- Future you
- Current team
- Future team (may or may not include you!)
- Conscientiousness
- Transparency/accountability
- Efficiency
- Reproduction/revision

How

- PLAN before you code!
    - Data sources, meta-data/documentation requirements, analysis plan, packages, output
- Create a project-specific Repo in GitHub
- Create and connect a unique RStudio project to that Repo
- RMarkdown analysis plan - for you and your team (can be README)
- Use R scripts for analysis - remember to commit and pull/push frequently
- RMarkdown final report - for stakeholders/public/project completion

Set-Up Instructions

Please arrive at the workshop with current versions of the following software installed on your laptops:

Obtain a free GitHub account
Install/update R and RStudio
Install Git: windows, mac, linux
Install the RMarkdown package and all dependencies
For PDF output, install LaTeX via Tinytex

Happy Git with R by Jenny Bryan is a fantastic resource to walk you through the steps of setting up Git and GitHub and establishing connections between the software.

Please take the time to update R and RStudio before you arrive. Really old versions may not play nicely with Git and RMarkdown.

Tricks

- Each work session, pull before pushing from local disk (avoids merge errors if changes have happened online)
- Time travel is only as good as your commit habits
- Ammend commits while working out code chunks
- Branch for experiments and variations on a theme
- Rmarkdown output: github_document
- Disaster recovery for new git users: Burn it down

Session Questions

Embed data output in RMarkdown without linking data:

From THIS stack conversation:

The reason knitting RMD files requires embedded data files is intentional to force/promote reproducibility.

Tech info: In order to perform the render in the background, RStudio actually creates a separate R session to render the document. That background R session cannot see any of the environments in the interactive R session you see in RStudio.

If it is inappropriate to embed data (privacy, proprietary data, etc), instead of using the *Knit HTML* button, type `rmarkdown::render("your_doc.Rmd")`at the R console. This will knit in the current session instead of a background session.

Allow collaborators to push to private projects

From GitHub: Collaborators on a personal repository can push to (write), pull from (read), and fork (copy) the repository

Project

Navigate to: github.com/karinneff/CSP2020_workflow_project

Fork

From your copy of the repo, clone and copy URL

Open a new version control project in RStudio using the URL from YOUR fork (make sure it has your GitHub handle, not mine!!!)

Best practice: Make a branch to work on, rather than the local master

Resources

RMarkdown cheatsheet and Reference Guide

RStudio cheatsheets page

Git cheatsheet

Happy Git with R - start here

Git in Practice - then here for more

Git Magic - and here for command line