Reproducible reports and version control: Homework

HEV9009 Scientific Communication, Collaboration and Design in Quantitative Research Using R

The goal of this homework is to get familiar with git and GitHub for reproducible, version-controlled, collaborative science. Follow the list below to complete your part of the homework. Notice also that for the second part, you will contribute to someone else's repository through a pull request. This requires that at least some of you share the repository address before Thursday next week. The basic idea is to create a reproducible report. This report can be about anything; focus your effort on developing your workflow rather than the report's content. The report should, however, contain enough so that your fellow students can suggest changes.

1. Create a GitHub repository in your user name. This can be done from a local repository, or by using the user interface online GitHub.com.
2. Add components of a reproducible report to your repository. These should include, (i) a README file, (ii) a Quarto file used for the report, (iii) a CSL file, and (iv) a bibliography file (`.bib`). You can add your own data or use available example data. A nice way to organize the repository is to put data in a subfolder called `data`, CSL and bibliography files in a folder called `resources`, and any R-code needed for analysis in a folder called `R`. Stage your added files, commit them to the version history, and push them to the online repository.
3. Update the `.gitignore` file to exclude any files you do not want in the version history, such as the resulting report (e.g., by adding `*.html` to the `.gitignore` file).
4. Incrementally add components to your report, making small commits as you go. The report should include some prose on the background for your analyses, a figure, and a table. Each commit should have a descriptive commit message.
5. When adding more complex components to your report (e.g., a figure), make sure to test that it renders before committing.
6. Update the README file with a description of the repository.

When doing these first cycles of development, *send an e-mail to your fellow students with your repository name* `/. When you recieve this information from another student you may start part two of the assignment.

1. Fork a student repository to your own user profile in Github. Clone the content to a local repository for you to work in and inspect the content of the repository.
2. Find a suitable area for improvements. Make changes, add these, commit, and then push these changes to your online repository.
3. Using the GitHub interface, create a pull request based on your fork. Describe the pull request using the interface.

When you receive a pull request, feel free to incorporate it into your repository. If you have done extra work on your repository, the pull request may not be up to date with your current version. In such case, explore what parts you are willing to incorporate. A merge can be selective. Or you might reject the pull request and ask the contributor to make a smaller, more specific pull request. Use the discussion capabilities on GitHub for communicating around pull requests.