Version Control and Github

Socorro Dominguez

10/22/2020

What is Version Control?

Version Control is a class of systems responsible for monitoring changes to computer programs, documents or other collections of information. When you work using Version Control, it means that if you accidentally destroy a file or lose it, you will be able to recover it.

Why use Version Control?

Examples

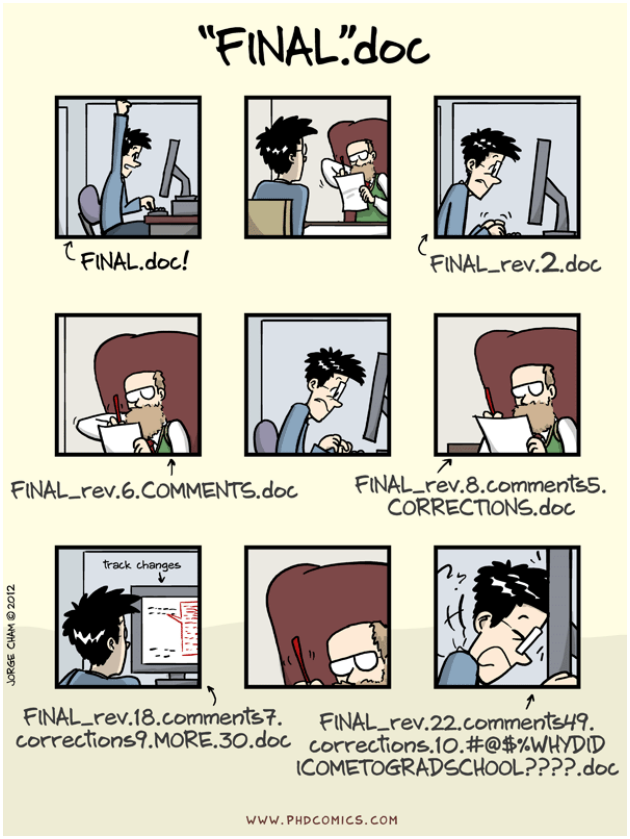


Figure 1: *"Piled Higher and Deeper" by Jorge Cham, http://www.phdcomics.com*

An introduction to GitHub

At the heart of GitHub is Git, a Version Control System and the language we need to speak to communicate with GitHub. Although it's mostly used for code, it can be used to manage any other type of file, such as Word documents. Think of it as a filing system for every draft of a document.

GitHub is a Git repository hosting service with many of its own features. While Git is a command line tool, GitHub provides a Web-based graphical interface.

Beyond managing and keeping versions of documents under control, there are other reasons for using the Git/Github version control system:

- Github (website) can act as a back-up for files housed there
- Github can be used to host websites/blogs
- Github has a fantastic search functionality

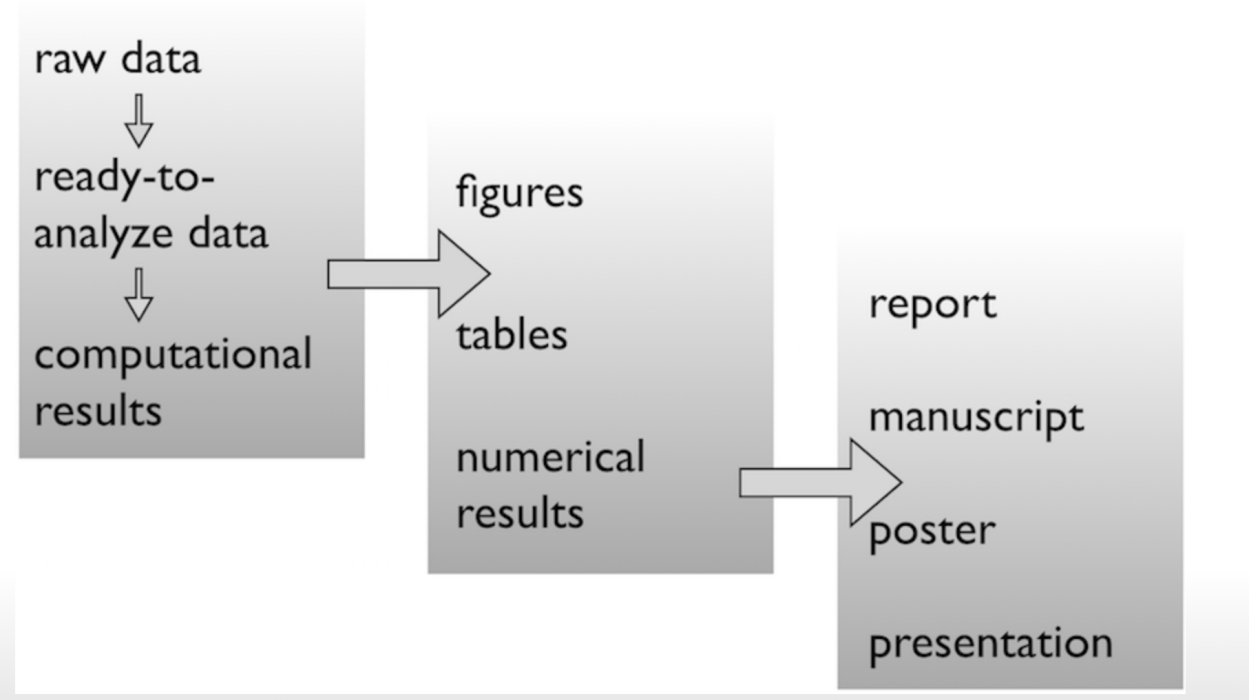
What is a repository?

We have said that GitHub is a repository hosting service. But what is even a repository? A repository or a "repo" is a directory or storage space where your projects can live. It can be local to a folder on your computer, or it can be a storage space on GitHub. You can keep code files, text files, image files inside a repository.

What are important pieces in a repository (Introducing the Template keywords etcetera)

When making a repository, there are some things that you should keep in mind:

- Make your repository a well documented one. This means comments throughout, readable code and a brief summary at the top of each file that answers who wrote it, when it was written and what it does.
- Use a README.md file at the root of your repository where you document the basics of the project you are working on. Use this file to also explain what the file structure looks like and what each file contains.
- If you are going to be collaborating with others, write a Code of Conduct. A code of conduct defines standards for how to engage in a community. It signals an inclusive environment that respects all contributions. It also outlines procedures for addressing problems between members of your project's community.
- Public repositories on GitHub are often used to share open source software. For your repository to truly be open source, license it so that others can enjoy it. MIT License is one of the most common used. It is a simple permissive license with conditions only requiring preservation of copyright and license notices.
- "A place for everything, everything in its place". File organization is crucial against chaos. Make a file's name and location VERY INFORMATIVE about what it is, why it exists, how it relates to other things. The more things are self-explanatory, the better.



- Names **matter**. When naming your files, remember that you want to make sure that:

  - They are *machine readable*
  - They are *human readable*
  - They play well with default ordering - order files either chronologically or logically.

    Examples of bad names:

    - myabstract.docx
    - Joe's Filenames Use Spaces and Punctuation.xlsx
    - figure 1.png
    - JW7d^(2sl@deletethisandyourcareerisoverWx2*.txt

    Examples of good names:

    - 2014-06-08_abstract-for-sla.docx
    - joes-filenames-are-getting-better.xlsx
    - fig01_scatterplot-talk-length-vs-interest.png
    - fig02_histogram-talk-attendance.png
    - 1986-01-28_raw-data-from-challenger-o-rings.txt

  - Commit to git every time you work on the project. When you commit, make your commit messages meaningful!
  - When creating a repository, you have to think about your future users. You should be able to walk away from the project and come back to it a year later and resume work fairly quickly. Other people should be able to figure out what you did and how to continue your work.

Sounds complicated? Take a look at this template to help you out: Throughput Template