

Structuring your project

September 8, 2020

Contents

Project structure	2
Your Project folder / Respository	3

Project structure

The general recommendations of this section in regards to establishing a consistent structure for your project(s) should apply whether or not you plan to use version control software to manage your project(s) or not. For example, the recommendations apply equally if you plan to use **Dropbox** or the equivalent (which you should *most definitely* be using if you're not going to use version control software).

Like many grad students, I finished my thesis with

- one master folder called **Data** containing a bunch of sub-folders within it containing all the various data sets (mostly Excel and CSV files) and databases (mostly MS Access) that I'd collected or collated over the years;
- one master folder called **Rcodes** containing a bunch of sub-folders within it (each with a different project or analysis of my data);
- one master folder called **Mathematica** that similarly contained a bunch of sub-folders;
- one master folder called **Manuscripts** that contained all the papers and chapters I'd attempted, completed or published;

and a bunch more similar folders all variously named within my overall **Research** folder. You might currently have something similar for just your Thesis.

Turns out that's a poor way to organize your work for a variety of reasons, including reproducibility (by yourself down the road or someone else if you managed to pull all the necessary parts together for that person); the ease and efficiency with which you might expand, modify or branch off of prior work; and the ease of performing data and code backups.

I now organize my work using a *project mindset*. I don't do this for each and every project idea or analysis I try out, but I do use it for "definitely doing this" (i.e. planned out paper (thesis chapter)) and collaborative projects. By *project mindset*, I mean that everything associated with a given project is contained in one folder. I still use a combination of **Dropbox** and **git** to organize my projects (and you should *never* put a **git** folder within your

Dropbox folder, or vice versa), but within each of either Dropbox or git I have all my project folders organized within a master **Projects** folder.

That said, defining “a project” can get difficult (esp. within your thesis work), so a fair bit of forethought can be needed. It’s not trivial.

Ben says: I also use *project mindset* to organize my folders. Something I like about project mindset is that it encourages what you might call deliverables-based thinking. By identifying and naming the “definitely doing this” projects, I am encouraged to consider my priorities, both within and among projects.

For each project I am forced to think clearly about what it is about by needing to name the folder. Asking “what should this project folder (or Git repo) be called?” (and insisting on an *informative name*, eg not “Mark-Bencoursething”) is pretty close to asking “what is this project about?” So project organization supports clear scientific thinking.

I find project mindset also promotes better time management. For instance, all my project folders are contained within three superfolders: Active, Complete and Archive. The folders within Complete are named with dates and brief titles of publication. The Active folder contains stuff I am working on right now, that has not “shipped” yet. Archive is for stuff that is on the back burner.

In this setup, the project folders within the Active folder, each with a name that reminds me of the objective for that project, becomes a kind of high-level to-do list. The goal is to be able to one day drag those folders from Active to Complete. Crucially, if the Active folder gets too full, I know I will not be able to do that because my attention has become too divided. So then I ask myself which are the most important few projects to me, and drag the rest to the Archive folder. It’s not that I can’t do them later. It is just recognizing that a) they are not done yet, and b) they are not the first, second or even third priority. If a) and b) are true, into the Archive folder they go!

OK, back to Mark, and the structure of an individual project folder.

Your Project folder / Respository

Within each project folder I usually have the following sub-folders:

data	The original (and cleaned) data required for the project
code	All the scripts needed to perform the analyses
output	The results of the analyses
figures	The final figures (and tables) that go into the manuscript
manuscript	Manuscript(s) derived from the project
pdfs	(<i>as appropriate</i>) Collection of relevant papers, manuals, etc.
biblio	(<i>as appropriate</i>) Bibliographic files
talks	Presentations you've given on the project

The last two (**pdfs** and **biblio**) sometimes get put in sub-folders within the **manuscript** folder. The **manuscript** folder often contains sub-folders, one for each journal I've submitted to. The **figures** folder sometimes gets put within each of the **manuscript** sub-folders, depending on how much I decide to change what the final figures are for different journals. The contents of **figures** differs from the rough-and-dirty figures I save into the **output** folder. Sometimes **tables** get their own folder (if there are a lot of them).

Ben says: My sub-folder structure is similar to what is listed below, with variations depending on the project and preferences. For example, my reference manager of choice keeps all my pdfs in one place, so instead of having a pdfs folder in each project folder, I have 'folders' for each project within my reference manager. Either way, the same goal is achieved: a logical hierarchical structure that makes it easy to find and keep the various pieces of a project. Back to you Mark.

Most of the time when using **git** you'll have one *repository* associated with each one of your *projects*. A *repository* as thus synonymous with a *project folder*. When using **git** you'll also have a few other files within the repository: a **README.md** file and a **.gitignore** file. If you're using **R-Studio** in combination with **git** (as we will below), then you'll also have an **.Rproj** file in the repository.