Structuring your project

The general recommendations of this section in regards to establishing a consistent structure for your project should apply whether or not you plan to use version control software to manage your project 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).

Developing a project mindset

Like many grad students, I finished my thesis with

- one master folder called `Data` containing a bunch of sub-folders containing the various data sets (mostly Excel and CSV files) and some relational databases (Microsoft 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 (chapter) and/or set of analyses of some set of my data);
- one master folder called `Mathematica` that similarly contained a bunch of sub-folders for various projects;
- one master folder called `Manuscripts` that contained all the papers and chapters I'd attempted or completed;

and a bunch more similar folders all variously named and "type-specific" within my overall `Research` folder. You might currently have something similar for your thesis work (Fig. 1).

Turns out that's a poor way to organize your work for a variety of reasons, including the ease of backing-up new data and code; the ease and efficiency with which you might expand, modify or branch off of your prior work; and even the simple reproducability of past work (by yourself down the road, or by someone else for whom you'll have to pull together all the necessary parts).

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 papers (thesis chapters)) and collaborative projects. By project mindset, I mean that (almost) everything associated with a given project is contained in one folder. I do still use a combination of `Git` and `Dropbox` to organize my projects based on my plans for projects[1] and collaboraton needs, but within each of `Dropbox` and `Git` I have my project folders organized within a master folder.[2]

All that said, defining "a project" can get difficult (esp. within your thesis work), so a fair bit of forethought is often needed. We'll definitely talk about this as a group in class. It's not trivial, but becomes quicker with time and seeing what others do.

> *Ben adds:* 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 the project is fundamentally about by having to name the folder. Asking "what should this project folder (or Git repo) be called?" (and insisting on an *informative name*)[3] is pretty close to asking "what is this project about?" So a project mindset supports clear thinking.
>
> I also find that a project mindset promotes better time management. For instance, all my project folders are contained within three superfolders: Active, Complete and Archive. The

[1] For example, whether I plan to make the entire project publicly available.

[2] Note that you're asking for trouble if you put a `Git` folder within your `Dropbox` or `GoogleDrive` folder, or vice versa.

[3] Naming conventions are something we'll come back to when we talk about good coding practices.

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 succeed in moving project folders 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 both (a) and (b) are true, into the Archive folder they go! OK, back to Mark, and the structure of an individual project folder.

Structuring your Project folder / Repository

For most projects, 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
`results`	all the output of the analyses
`biblio`	bibliographic files
`figs`	final figures (and tables) that go into the manuscript
`manuscript`	manuscript(s) derived from the project
`pdfs`	collection of relevant papers, manuals, etc.

The first three (`data`, `code` and `output`) are definites for any project (repository) that I plan to make public (e.g., on GitHub). For such public repositories, I do not include the other four folders. I use `figs`, `manuscript`, `biblio` and `pdfs` for manuscript-writing "projects" (which I place in an `InPrep` manuscripts-only sub-folder within my master `Git` folder). The contents of `figs` differs from the rough-and-dirty figures I save into the `output` folder. Sometimes `tables` get their own folder. Within `code` I might have an `R` and a `Mathematica` folder, as relevant. Note that `data` contains both the un-

touched original data[4] and the clearly-identified cleaned or otherwise derived data (for ease of subsequent access).

> *Ben says*: My sub-folder structure is similar, 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* is 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 in this course), then you may also have an `.Rproj` file in the repository. It's good practice to have a `README.md` file in each of the "important" folders, describing what in them.

[4] Leaving your original data as untouched as possible is something we'll come back to during Coding Best Practices

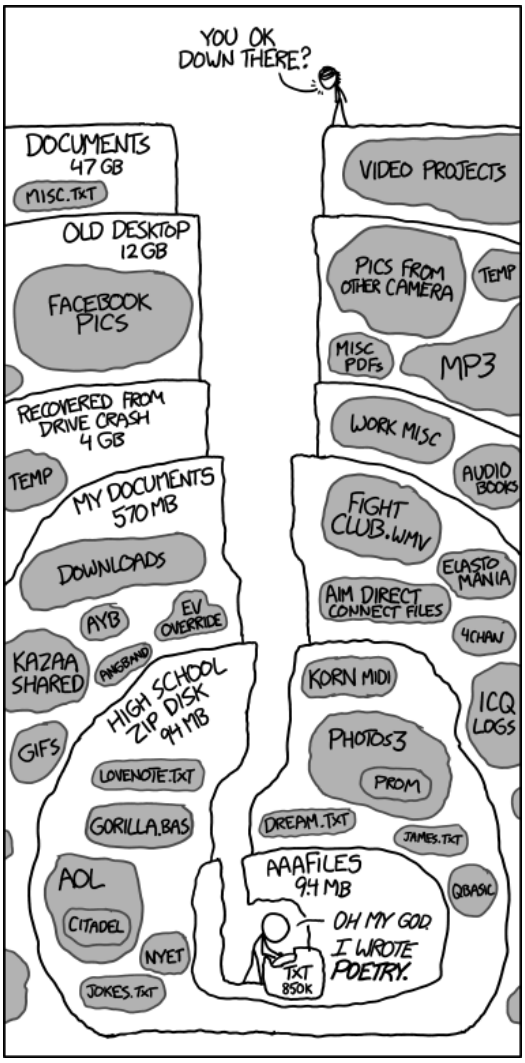

Figure 1: Old files (source: http://xkcd.com/1360/)