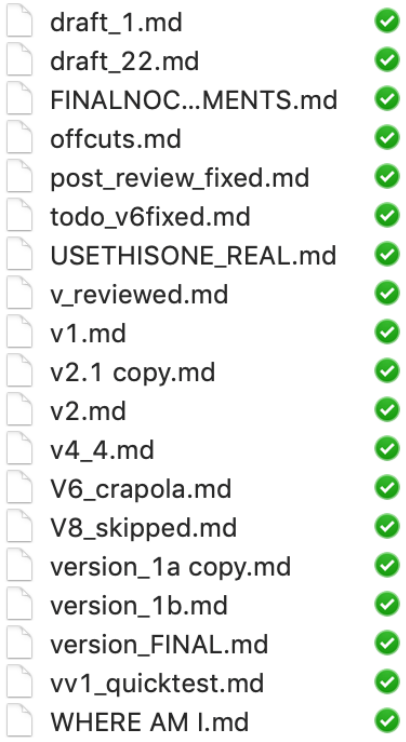
SOLES codeRs

Version Control with Git and GitHub

Thomas White

What is version control?

Everyone uses it in one way or another:



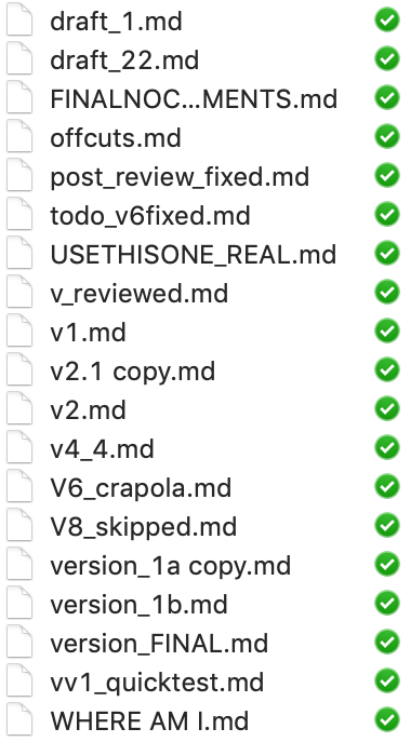
Why use proper version control?

- Explicit, well-documented project history.
- Tight integration between tidily organising, completing, disseminating work. Satisfying. Sparks joy.
- Ease of revision, experimentation, and/or recovery from error.
- Structured collaboration, sharing, & reproducible research.
- Bonus skills: build personal website, host your own blog (lol), database, etc.

When use proper version control?

- Writing papers & managing projects (alone or collaboratively).
- Writing lectures and talks.
- Sharing: teaching material, code, data, papers…
- Writing/maintaining software.

Can we do better?



Yes

Git

- A ‘distributed’ version control system

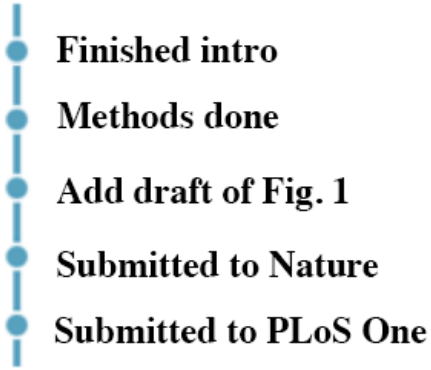
paper_v1.Rmd
paper_v2.Rmd
paper_v3.Rmd
paper_v3.3.Rmd
paper_v3.3.1.Rmd
paper_v7?.Rmd

Yes

Git

- A ‘distributed’ version control system

manuscript.Rmd



Yes

Git

- A ‘distributed’ version control system
- Manages the evolution of a set of files—a **repository** (‘repo’)—in a sensible, flexible way
- Takes ‘snapshots’ of a project when you tell it to, and stores it alongside your informative description of that snapshot
- Kind of in the vein of Word’s ‘track changes’ + Dropbox
- Can use via command line or friendly apps, like GitHub Desktop
- n.b. A ‘repository’ is just a folder on your computer that you’ve told Git to keep an eye on

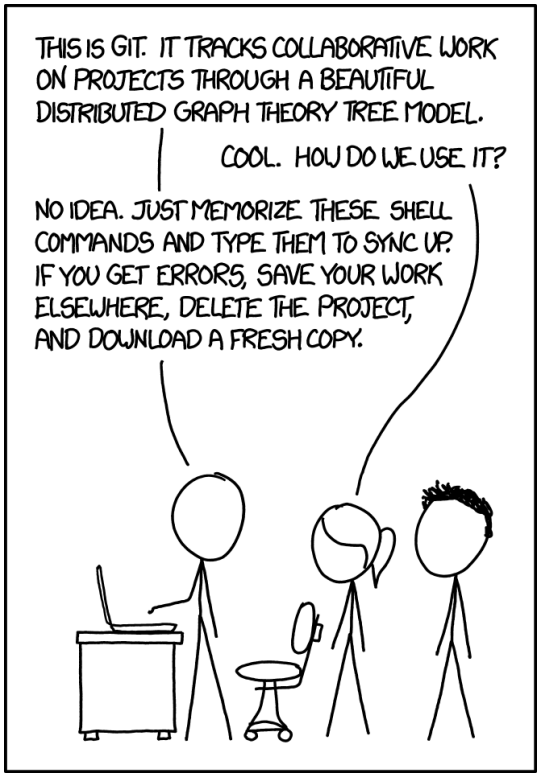
The workflow: outline

Solo projects

1. Verbal step-by-step
2. Visual step-by-step
3. Joint adventure

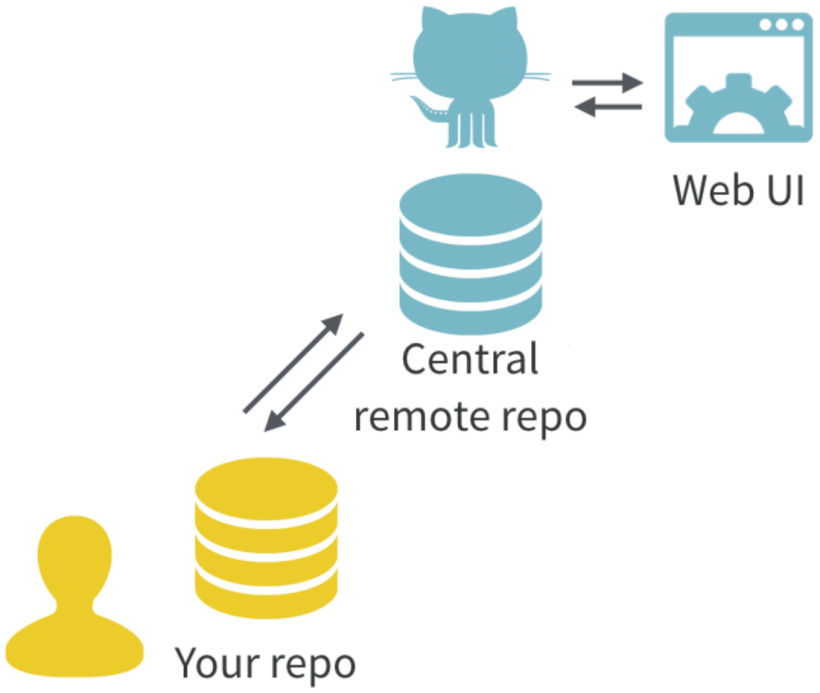
Collaborative projects

1. Visual step-by-step
2. Visual step-by-step
3. Joint adventure



The workflow: solo

The workflow: solo



Keywords: **commit** your changes, **push** to the remote repo.

The workflow: solo

Setting up a new project:

- **Create a new local repository** using GitHub Desktop
- Add whatever files you like to the repository, as you would for any other project.

Day-to-day:

- Make some changes anywhere in your project
- **commit** those changes, with some helpful descriptive comments
- **push** those changes to the remote repository
- less frequently: **revert** changes, use **branches**

The workflow: solo

Setting up a new project:

- **Create a new local repository** using GitHub Desktop
- Add whatever files you like to the respository, as you would for any other project.

Day-to-day:

- Make some changes anywhere in your project
- **commit** those changes, with some helpful comments
- **push** those changes to the remote repository
- less frequently: **revert** changes, use **branches**

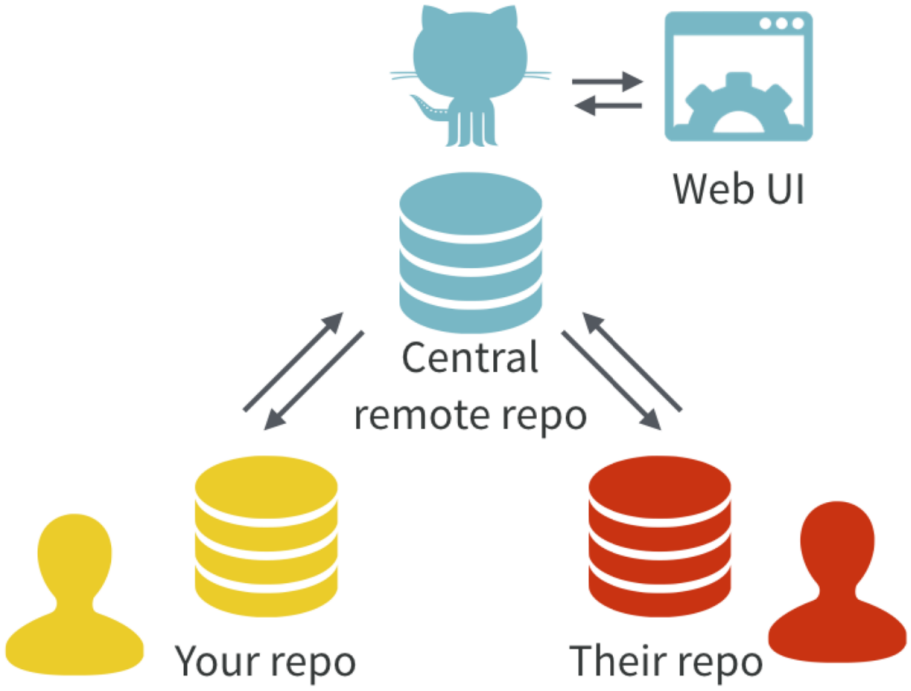
Let's do it!

Lets do it

1. Create a GitHub account
2. Download Github Desktop here
3. Create a new ‘public’ repository via Github Desktop
4. Open the repository (folder) on your computer, and add some files to it
5. Commit those changes, including a helpful ‘summary’ message and ‘description’
6. Push those changes to the remote repository. The first time you do so, you’ll need to ‘publish’ the repository.

The workflow: collaborative

The workflow: collaborative



Keywords: **pull** any new changes, **commit** your changes, **push** to the remote repo.

The workflow: collaborative

Two forms:

- You & collaborators have full access rights (new skills: **cloning** & **fetching/pulling**)
- You and/or collaborators do not have full rights (new skills: **forking**, **cloning**, **fetching/pulling**, **pull requests**)

The workflow: collaborative

Two forms:

- **You & collaborators have full access rights (new skills: cloning & pulling)**
- You and/or collaborators do not have full rights (new skills: **forking**, **cloning**, **pulling**, **pull requests**)

The workflow: collaborative

Setting up a new project:

If you're leading the charge:

- **Create a new local repository** using GitHub Desktop
- Add whatever files you like to the respository, as you would for any other project.
- **Add your buddies** to the remote repo on GitHub, so they can do what they like to it too

The workflow: collaborative

Setting up a new project:

If someone else is leading the charge:

- **Clone** their *remote* repository from GitHub, via GitHub Desktop
- That's it. Use it like you would your own repo.

The workflow: collaborative

Day-to-day:

- **Fetch/pull** any new changes from the remote repository
- Make some changes to files in your local repository
- **Commit** those changes, with helpful comments
- **Push** those changes to the remote repository
- less frequently: **resolve conflicts**
- Actually now I'm here it's exactly the same as solo, but for the occassional conflict

The workflow: collaborative

Day-to-day:

- **Fetch/pull** any new changes from the remote repository
- Make some changes in your local repository
- **Commit** those changes, with helpful comments
- **Push** those changes to the remote repository
- less frequently: **resolve conflicts**
- Actually now I'm here it's exactly the same as solo, but for the occassional conflict

Let's do it!

Thanks!