

Introduction to Git and GitHub

Instructor: Haoran LEI

Hunan University

Motivation

- When you are working on a long-term project (say writing a paper) on your computer, you are likely to have **many different versions**, with each new version adding some (perhaps minor) changes to the previous version.

Motivation

- When you are working on a long-term project (say writing a paper) on your computer, you are likely to have **many different versions**, with each new version adding some (perhaps minor) changes to the previous version.

Example

- `paper-draft.pdf`, `paper-draft2.pdf`, `paper-draft3.pdf`, ...
- `paper-final.pdf`, `paper-real-final.pdf`, `paper-final2.pdf`, ...

How an usual researcher manages a solo project

```
delegation-paper
|—delegation-202107
|—delegation-202110
|—delegation-202203
|—delegation-202205
|— ... ..
```

- Each `delegation-yyyymm` is a folder containing the tex source, figures and pdfs.

How an usual researcher manages a solo project

```
delegation-paper  
|—delegation-202107  
|—delegation-202110  
|—delegation-202203  
|—delegation-202205  
|— ... ..
```

The Good:

- It works well as long as I have a mental map of the folder/directory contents.
- It is **easy** to manage different versions of a paper in this way. One just needs to have a **consistent file naming** habit.

How an usual researcher manages a solo project

```
delegation-paper
|—delegation-202107
|—delegation-202110
|—delegation-202203
|—delegation-202205
|— ... ..
```

The Bad:

- It will be a nightmare to **collaborate**. Your coauthors will hate you.
- This only works for **small projects**.

You need a **version control system** (VCS) if (i) you collaborate with others or (ii) your project will be big (say, thousands lines of code).

What's Git?

- Git is a distributed version control system.
- Git has the "Track Changes" feature in Microsoft Word and the "Sync" feature of Dropbox.
- In fact, Git also has many other features that make it a very popular tool for collaborating on big projects

How to use Git

- Git has a learning curve if you never use a **command line tool** before.
- Using Git is very similar to using R. No mouse clicks. Every job is done via commands.
- If you have exposure to CLI (e.g., you know what `cd`, `ls`, `mkdir` mean), using Git will be easy for you. Just read the Pro Git book, and then learn by doing. <https://git-scm.com/book/en/v2>
- If using R in this course is the first time that you write multiple lines of commands/codes, you should **start with GitHub** and use Git indirectly.

GitHub

Git and GitHub are two distinct things.

- Though most people get to know and use Git through GitHub.

GitHub is an online hosting platform that provides an array of services build on top of the Git system. (Similar platforms include GitLab and Bitbucket)

Just like we do not need RStudio to run R code, we do not need GitHub to use Git.

- But Github makes using Git much easier.

How to use GitHub

- GitHub provides a web interface for using Git.
- See how I do it.