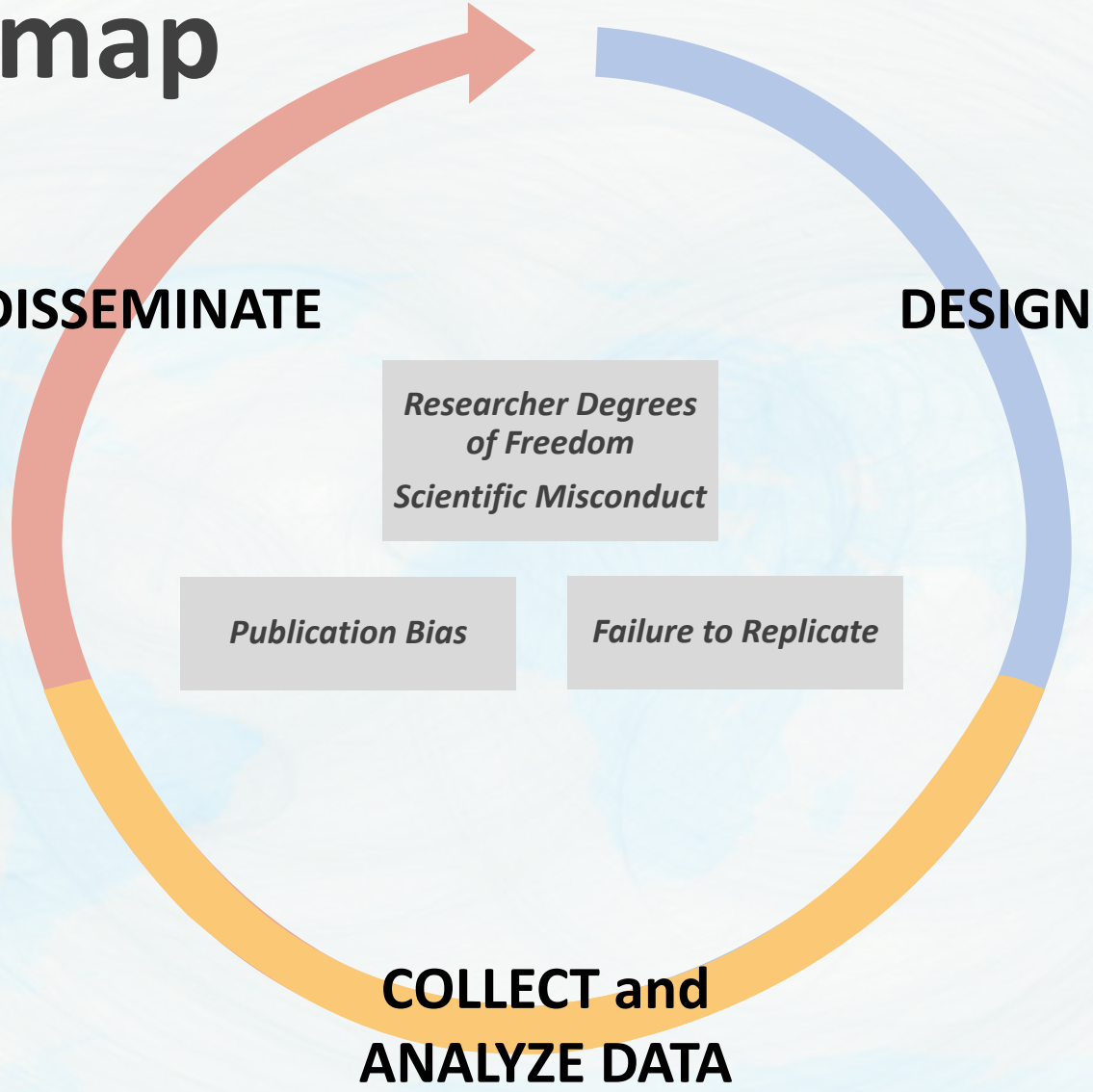
Version Control w/ Command Line
Git and GitHub

Tim Dennis, UCLA/Library Data Science Center: I support researchers engaged in data-intensive scholarship primarily in the social and behavioral sciences

RT2 Roadmap



- Transparent Reporting & Disclosure
- Data Repositories
- Dynamic Documents
- Results-Blind Review / Registered Reports
- Preprints
- Open Access

- Pre-Registration
- Pre-Analysis Plans
- Power Planning

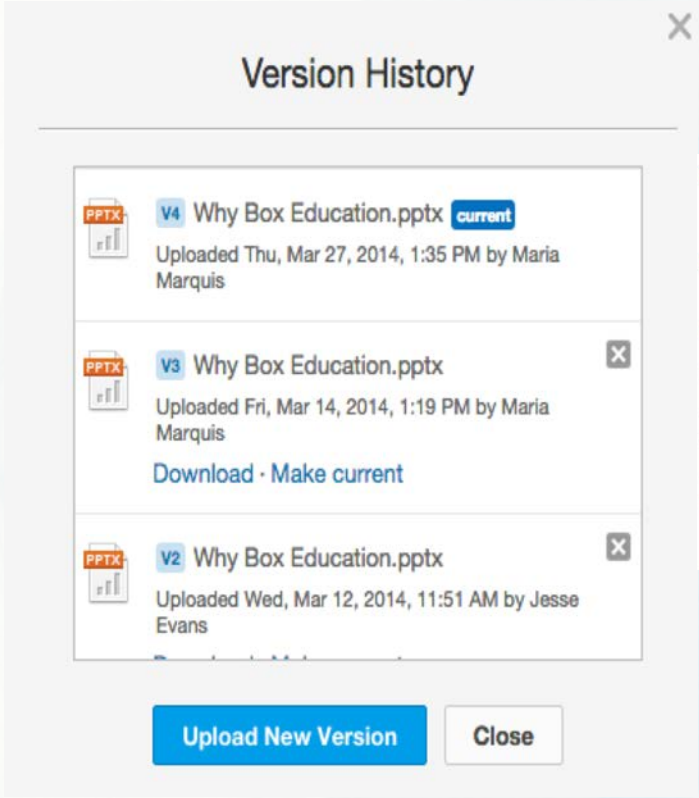
- Data Management, De-Identification
- Version Control
- Dynamic Documents

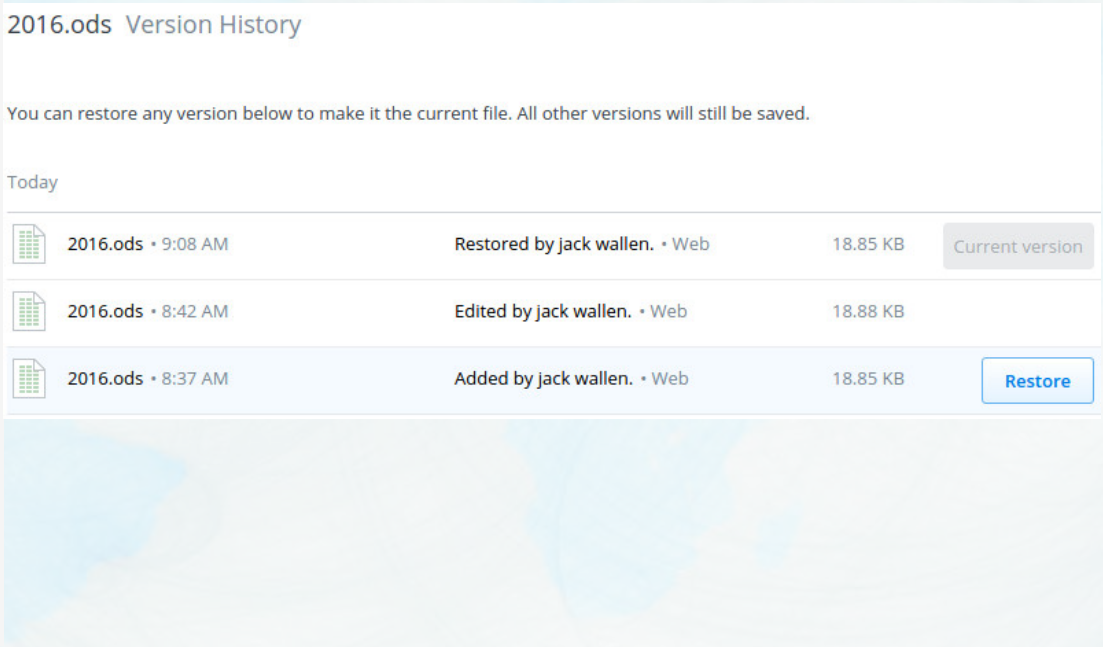
What are we covering?

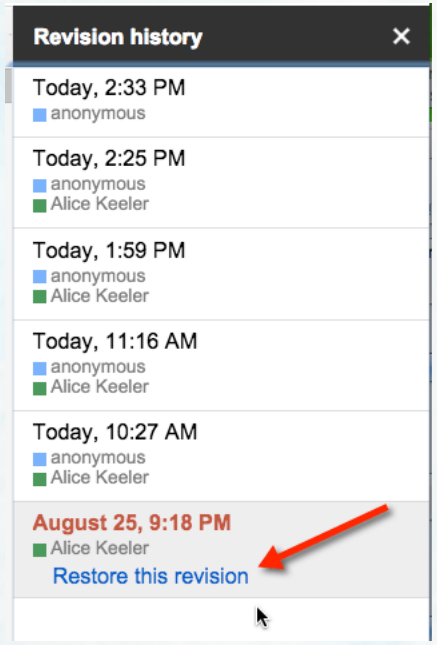
- What is version control?
- Why version control?
- Creating a Git project locally via the command line
- How to make revisions with Git
- Collaborating with GitHub

What is version control?

You might recognize:







Version Control

- Version control is a system that **records changes** to a file or set of files over time so that you can recall specific versions later.
- It allows you to **compare, restore, and merge changes** to your files.
- Examples: RCS, CVS, Subversion, GIT, Mercurial
- Git and Mercurial are distributed (do not need centralized server)

Source: https://git-scm.com/book/en/v2/Getting-Started-About-Version-Control

GIT

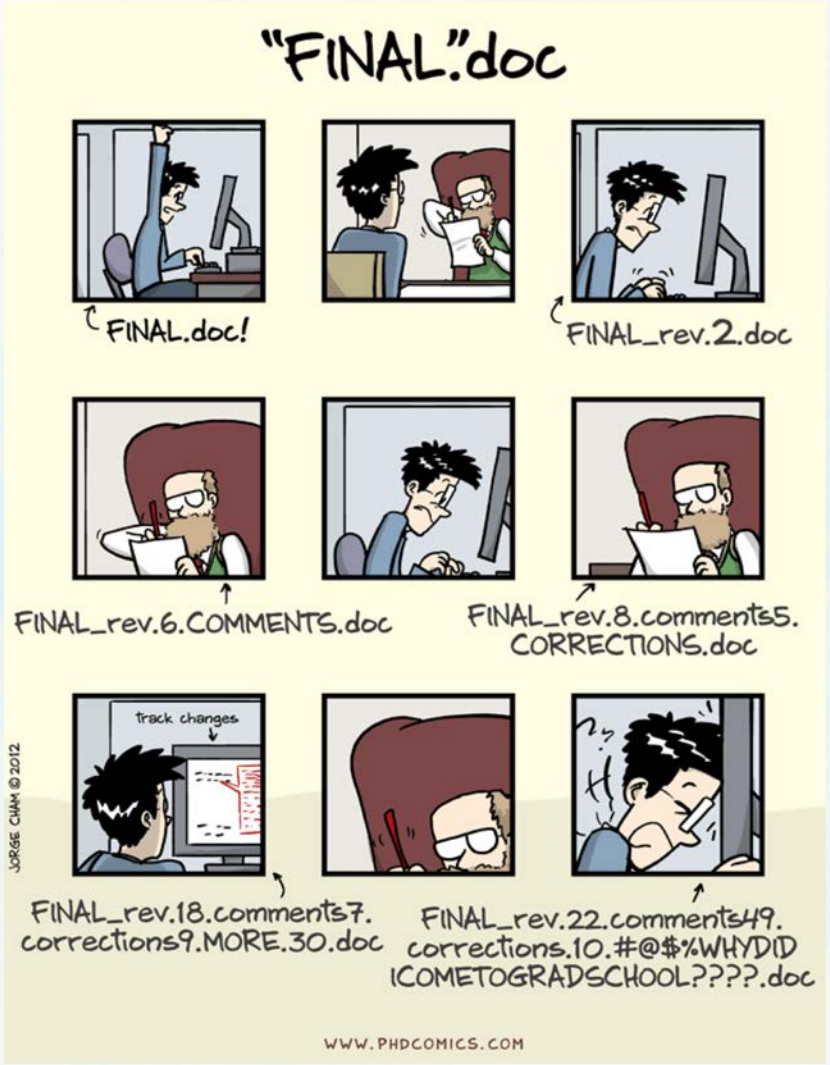
- Git is a free and open source version control system **for tracking changes in files** and coordinating work on those files among multiple people (supports distributed, nonlinear work)

- Git stores the metadata and files in a directory/data structure called a **repository**.

Git vs. GitHub

Git	GitHub
a tool that allows creating a local repository	allows hosting the central repository on a remote server
a version control **system**	a repository hosting **service**

Why should we use version control?

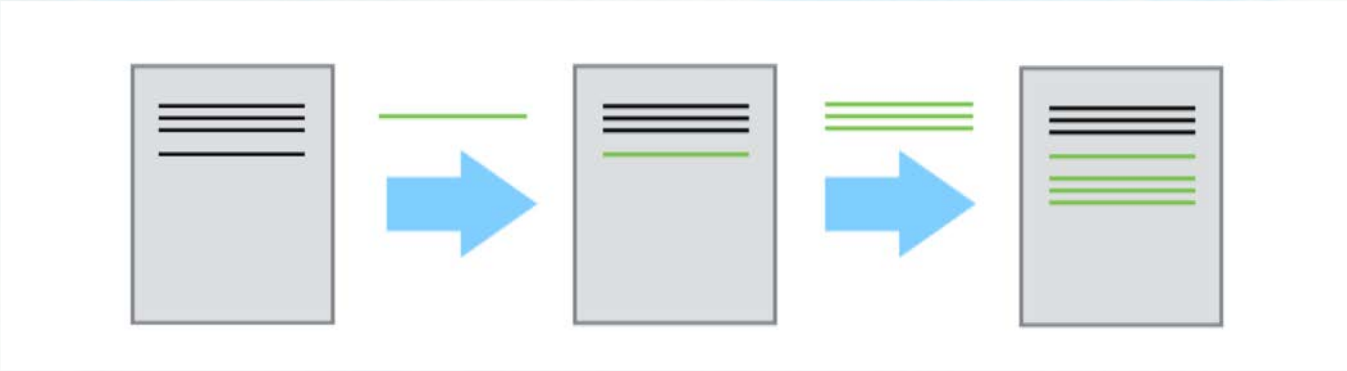


Source: http://www.phdcomics.com/comics/archive/phd101212s.gif

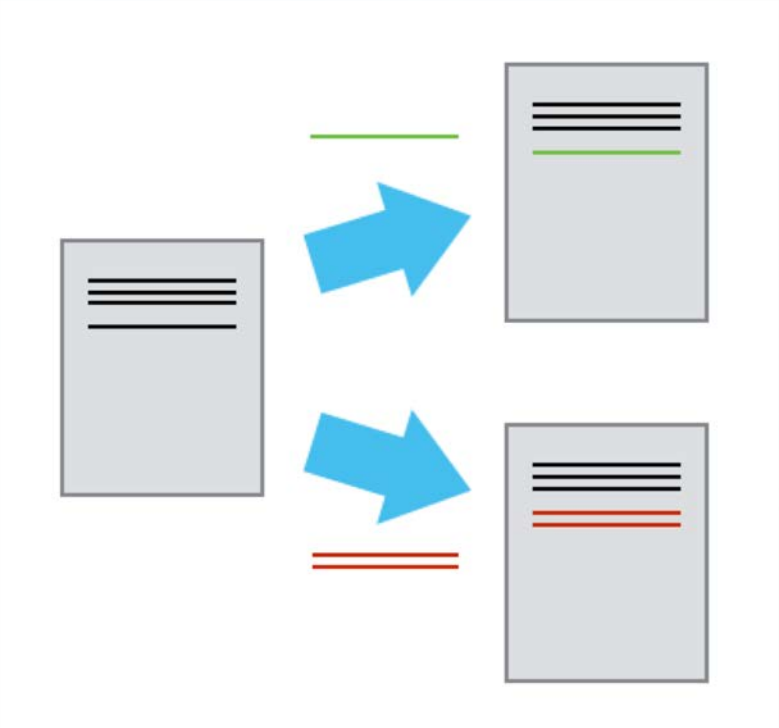
PROTIP: NEVER LOOK IN SOMEONE ELSE'S DOCUMENTS FOLDER.

How does this work?

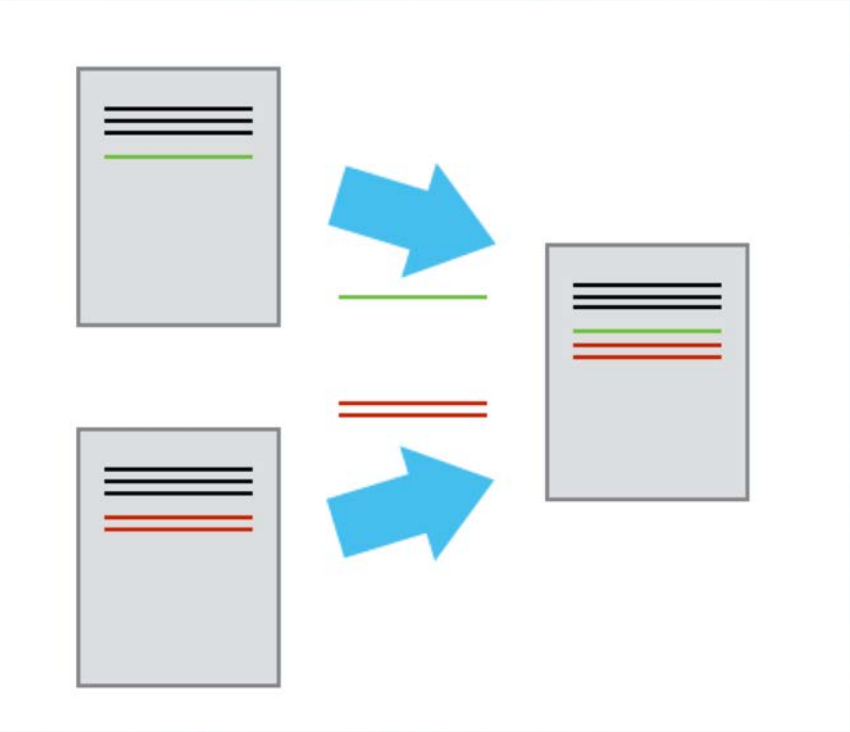
Changes Saved Sequentially



Divergent Versions



Merging



Brief Glossary

- **Version Control** - A tool for managing changes to a set of files. Each set of changes creates a new commit of the files; the version control system allows users to recover old commits reliably, and helps manage conflicting changes made by different users.
- **Commit** - To record the current state of a set of files (a change set) in a version control repository
- **Repository** - A storage area where a version control system stores the full history of commits of a project and information about who changed what, when.
- **Merge** - To reconcile two sets of changes to a repository

Today: Hands-on Carpentries Style Lesson

After this workshop you'll be able to:

- Configure and set up Git
- Create a local repository
- Add files and create revisions (commits)
- Understand how commits and exploring diffs work in Git
- Explain what remote repositories are and why they are useful
- Push to or pull from a remote repository
- Make a pull request and describe a basic collaborative workflow

Lessons, Materials, and Prerequisites

Lesson: https://swcarpentry.github.io/git-novice/
Etherpad: https://pad.carpentries.org/rt2-la
(collaborative document)
GitHub: https://github.com/join (get an account)
Git: https://git-scm.com/book/en/v2/Getting-Started-Installing-Git