---

title: Git FUN!damentals
subtitle: Introduction
minutes:

Let's avoid this humorous, yet sad situation from PhD Comics (http://www.phdcomics.com/comics/archive.php?comicid=1531).

How might you solve this problem?

1. Maybe you are using sequentially numbered directories
2. Maybe you are using track-changes
3. Maybe you are using Google Docs

What is the problem with each of these systems?

> What if you overwrite something in the wrong directory?
> What if you need to keep track of many files that depend on each other?
> What if you want people to propose changes, but not make them?

Enter version control

Version control lets you offload the work of keeping track of everything related to a project, including *documents*, *visualizations*, and *data*

You `initialize` it in a `directory`, and then tell it when you want your changes to be permanent

You can go back to a previous iteration at any time, and can see exactly what has changed between versions

For this workshop, we'll be using `git`

Automated version control systems are nothing new. Tools like RCS, CVS, or Subversion have been around since the early 1980s and are used by many large companies. However, many of these are now becoming considered as legacy systems due to various limitations in their capabilities. In particular, the more modern systems, such as Git and Mercurial are distributed, meaning that they do not need a centralized server to host the repository. These modern systems also include powerful merging tools that make it possible for multiple authors to work within the same files concurrently.

Git is a *distributed* version control system

```
--distributed-is-the-new-local
```

That means it can track, diff, and merge differences between the versions of a file on your system, and on *any other system on this entire planet of Earth*

This makes it an enormously powerful tool for collaborative work

> All of D-Lab's teaching materials and code are developed collaboratively with `git`

For this workshop, we'll also be using GitHub

GitHub (https://github.com) is a website for keeping tabs on all the modified versions of your file everywhere it appears. While other social coding collaboratory sites exist (BitBucket (https://bitbucket.org/), GitLab (https://gitlab.com), etc.), GitHub is currently the most popular, and has the added advantage for students of free private repositories (https://education.github.com).

GitHub makes it very easy to download, modify, and share code

> Because of difficulties with CRAN, many R writers publish packages on GitHub instead

Acknowledgments

This learning module borrows and adapts materials from the following organizations and individuals. Thank you!

Software Carpentry (https://github.com/swcarpentry/git-novice)