

Version Control: Git and GitHub

Jamie Saxon

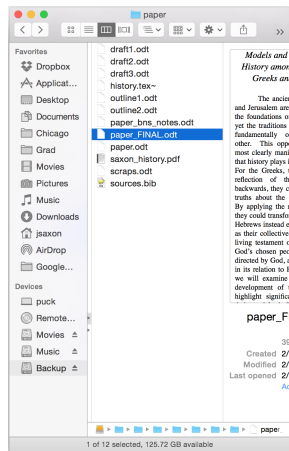
Introduction to Programming for Public Policy

September 27, 2016

What is version control? Why use it?

- ▶ Perhaps a familiar story, for paper drafts. \implies
- ▶ What if several people need to be able to edit simultaneously.
- ▶ What if there are many different files that depend on each other being at a specific version, all of which may be changed?

**Version Control Systems
maintain a history and
facilitate collaborative editing.**



What is git? GitHub?

- ▶ Git is the modern VCS, designed by Linus Torvalds (creator of Linux).
- ▶ Git maintains a history of meaningful 'commits.'
 - ▶ It is tremendously flexible ('branches').
- ▶ Git is distributed: everyone has a copy of the entire history.
- ▶ However, it is often useful to maintain a master copy on a server where anyone can access it or 'push' their changes: GitHub.
- ▶ GitHub is a nice **server** for hosting repositories.



Field Trip!

github.com/harris-ipp/lectures

Create new file

Upload files

Find file

Clone or download ▾

Clone with SSH ?

Use HTTPS

Use an SSH key and passphrase from account.

`git@github.com:harris-ipp/lectures.git`



Open in Desktop

Download ZIP

You'll use these regularly:

- ▶ **git clone**: download repository
- ▶ **git init**: create a repository in this directory
- ▶ **git add**: add a file to 'staging' area
- ▶ **git status**: view status of all files
- ▶ **git commit**: commit staged files to history
- ▶ **git push**: upload all changes to a remote server
- ▶ **git log**: show the history

Start with a single user and a single thread of edits:

1. Download your homework skeleton:
 - ▶ `git clone git@github.com:harris-ipp/01-welcome.git`
2. Make your edits with Atom or vim.
3. Add files to the 'staging' area, and commit them; check the status and log to see that it worked:
 - ▶ `git add q1.py`
 - ▶ `git status` # is everything there?
 - ▶ `git commit -m "started question 1"`
 - ▶ `git log` # now all part of the commit history?
4. Upload it to the server:
 - ▶ `git push`

Repeat steps 2-4 as you work.

Start with a single user and a single thread of edits:

1. Download your homework skeleton:
 - ▶ `git clone git@github.com:harris-ipp/01-welcome.git`
2. Make your edits with Atom or vim.
3. Add files to the 'staging' area, and commit them; check the status and log to see that it worked:
 - ▶ `git add q1.py`
 - ▶ `git status` # is everything there?
 - ▶ `git commit -m "started question 1"`
 - ▶ `git log` # now all part of the commit history?
4. Upload it to the server:
 - ▶ `git push`

Repeat steps 2-4 as you work.

This is what you'll use regularly.

Start with a single user and a single thread of edits:

1. Download your homework skeleton:
 - ▶ `git clone git@github.com:harris-ipp/01-welcome.git`
2. Make your edits with Atom or vim.
3. Add files to the 'staging' area, and commit them; check the status and log to see that it worked:
 - ▶ `git add q1.py`
 - ▶ `git status` # is everything there?
 - ▶ `git commit -m "started question 1"`
 - ▶ `git log` # now all part of the commit history?
4. Upload it to the server:
 - ▶ `git push`

Repeat steps 2-4 as you work. Do this now with a test repo.

This is what you'll use regularly.

A Standard Sequence: Atom

The 'staging', commit, and push commit them; check the status and log to see that it worked:

- ▶ Stage change.
- ▶ Commit with a comment.
- ▶ Click to push.

Easy to upload as you work!

