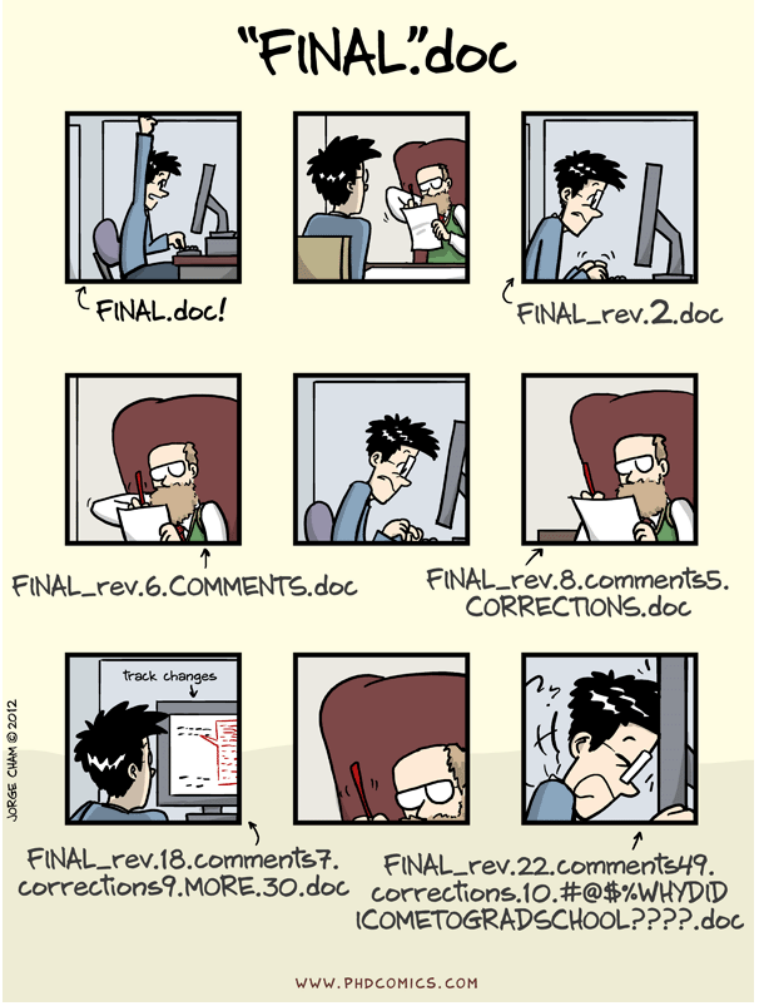
Introduction to Scientific Computing
Meeting 10
Version Control with git



Jeremiah Lant, Hydrologist
USGS Kentucky Water Science Center
jlant@usgs.gov

Last Meeting Objectives

1. **Understand what version control is.**
2. **Learn the basic git commands; git init, git add, git commit**
3. **Create your first Git repository on your computer**

Last Meeting Review

- Version Control – used **to keep track (a history)** of what you have done and **to collaborate** with others.
- From Software Carpentry:
  - **Version Control** - A tool for managing changes to a set of files. Each set of changes creates a new revision of the files; the version control system allows users to recover old revisions reliably, and helps manage conflicting changes made by different users.

- Git
  - Book – Pro Git by Scott Chacon
    - http://git-scm.com/book
  - Some Introductory Videos
    - http://git-scm.com/videos

Last Meeting Review

- **git init, git add, git commit**

$ git init *some-project-name* # initializes a project to be version controlled
$ cd *some-project-name*
OR
$ mkdir some-project-name
$ cd *some-project-name*
$ git init

$ git add . # add everything to “staging area”
$ git commit –m “*some-comment*” # commit (version) everything in “staging area”

Last Meeting Review

- Create a Git repository called “myhobbies” and write 2 of your hobbies in a file called “hobbies.txt”. Commit the changes, then add another hobby to “hobbies.txt”, and display the differences between the updated and original file.

$ mkdir myhobbies

$ cd myhobbies

$ git init

$ ls -a

$ start notepad++ hobbies.txt

$ git status # view current status of repo

$ git add hobbies.txt

$ git status

$ git commit –m “a list of my hobbies”

$ git status

$ git log # view all commits

$ start notepad++ hobbies.txt

$ git diff

Last Meeting Questions

- Can Git version control word documents (*.docx)?
  - **Yes**

$ mkdir testword
$ cd testword
$ git init
$ ls -a
$ start winword hello.docx
$ git status # view current status of repo
$ git add hello.docx
$ git status
$ git commit –m “testing git with word documents”
$ git status
$ git log # view all commits
$ start winword hello.docx
$ git diff

Finish up from last meeting

- Configure your name, email, and editor for git

$ git config –list
$ git config –global user.name Jeremiah Lant
$ git config –global user.email jlant@usgs.gov
$ git config –global core.editor notepad

- Add a few more hobbies and commit changes
- Explore the history of hobbies.txt using **HEAD~** and **unique 40 character identifier**

$ git diff HEAD~1 hobbies.txt
$ git diff HEAD~2 hobbies.txt
$ git log
$ git diff *your-unique-40-character-identifier* hobbies.txt

- Recover old versions using **checkout**

$ git status
$ git log
$ git checkout *some-unique-40-character-identifier* hobbies.txt
$ cat hobbies.txt
$ git status
$ git add hobbies.txt
$ git commit –m “back to older version”

Today's Objective

1. **Explore the history of version controlled file.**
2. **Learn how to recover old version of a file.**
3. **Learn what a remote repository is.**
4. **Sign up with a hosting site:**
    1. **GitHub (https://github.com/) or**
    2. **Bitbucket (https://bitbucket.org/)**
5. **Set up a remote repository on hosting site of your choice and push a local repository up to the remote repository.**

What is a remote repository?

- **Remote repositories** are versions of your project that are hosted on the Internet or network somewhere. You can have several of them, each of which generally is either read-only or read/write for you.
  - http://git-scm.com/book/en/Git-Basics-Working-with-Remotes

Sign up with hosting service

- **Public repositories -** Anyone can view the repository, but you choose who can commit.
- **Private repositories -** You choose who can view and commit to the repository.

- **GitHub - https://github.com/**
  - Pricing plan based on **number of private repositories.**
  - Public repositories are free.
  - Private repositories cost a certain amount based on the number of private repositories you have.
  - 5 private repositories cost $7 per month
  - https://github.com/pricing

- **Bitbucket - https://bitbucket.org/**
  - Pricing plan based on **number of users.**
  - Public and private repositories are free up to a certain number of users
  - 10 users plan cost $10 per month
  - https://bitbucket.org/plans

Create a remote repository and push local repository to remote

- Recall from meeting-9 a directory called “myhobbies” or “my-hobbies”
- Create a remote repository called “myhobbies” or “my-hobbies”
  - Just make sure that directory names match between local and remote

$ cd meetings/meeting-9/my-hobbies

$ git remote add origin https://jlant@bitbucket.org/jlant/my-hobbies.git

$ git remote -v # check that it worked

$ git push origin master

- The name “origin” is a local nickname for your remote repository
- Go to your remote repository and check that it worked.

Try out commands

- Create a local repo in a directory called “bash-scripts” at same level as your “data” directory we have used before.
- Move “peak2.sh” from data/nwis-files to “bash-scripts” and rename it to “peaks-nwisdv.sh”. Make necessary changes to make script work.
  - *Hint: need to edit path to data files in script.
- Add “peaks-nwisdv.sh” to staging area and commit it.
- Create a remote repository called “bash-scripts” and push local repository to it.
- Improve script with header comments
  - add and commit
- Improve script for any filename containing “dv”.
  - add and commit
- Push changes to remote repository.

- If time permits, make a bash script to find peaks in nwis unit value files. Version control the script from the very start. Push this script to your remote repository called “bash-scripts”

Create a local repository and push it remote

- Create a local repository for peak2.sh and push it to remote repository

```
for filename in *dv.txt
do
        echo $filename
        grep -v -e “#” -e “agency_cd” -e “5s” $filename | cut -f 4 | sort - n | uniq | tail -1
done
```

OR

```
for filename in *dv.txt
do
        echo $filename
        grep ^USGS $filename | cut -f 4 | sort - n | uniq | tail -1
done
```

Summary

- Remote repositories are versions of your project that are hosted on the Internet or network somewhere.
- A local Git repository can be connected to one or more remote repositories.
- **git remote add origin** https://jlant@bitbucket.org/jlant/my-hobbies.git
- **git push** copies changes from a local repository and “pushes” them to a remote repository.

Next meeting

- More with version control with Git
  - Collaborating
  - Conflicts