Git: Distributed Version Control

An Introduction

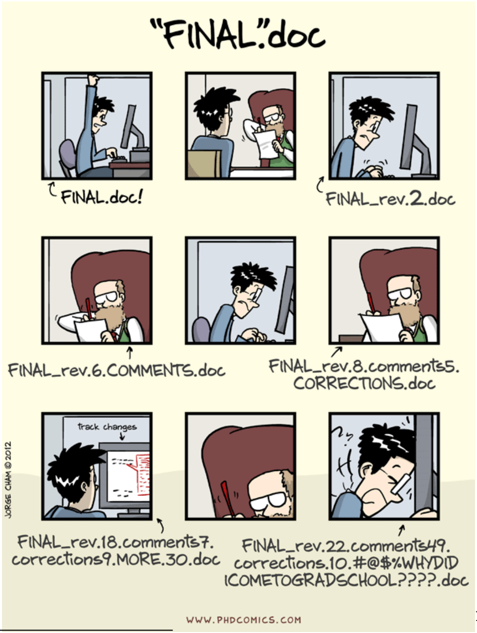
Peter E. DeWitt, Ph.D.[1]
peter.dewitt@cuanschutz.edu

SOM — Department of Pediatrics — Informatics and Data Science

5 December 2019

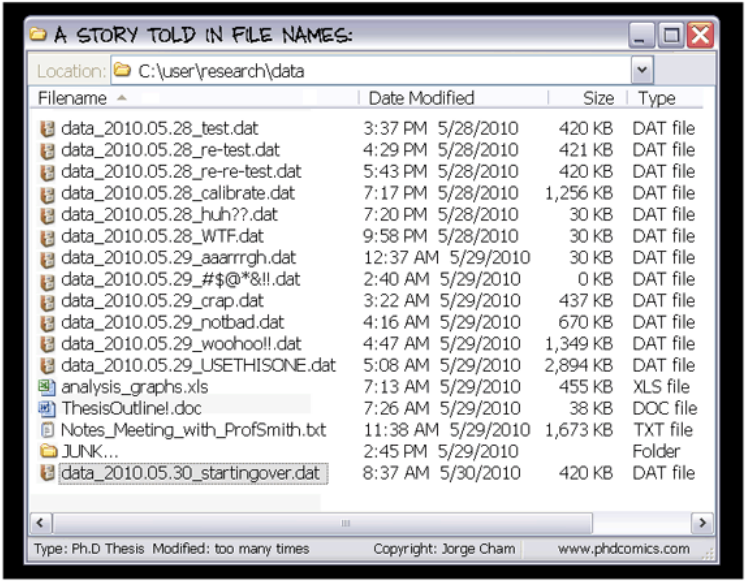
SCHOOL OF MEDICINE
Department of Pediatrics
UNIVERSITY OF COLORADO **ANSCHUTZ MEDICAL CAMPUS**

Topics



[1]

[1] http://phdcomics.com/comics/archive.php?comicid=1531



A STORY TOLD IN FILE NAMES:

Location: C:\user\research\data

Filename	Date Modified	Size	Type
data_2010.05.28_test.dat	3:37 PM 5/28/2010	420 KB	DAT file
data_2010.05.28_re-test.dat	4:29 PM 5/28/2010	421 KB	DAT file
data_2010.05.28_re-re-test.dat	5:43 PM 5/28/2010	420 KB	DAT file
data_2010.05.28_calibrate.dat	7:17 PM 5/28/2010	1,256 KB	DAT file
data_2010.05.28_huh??.dat	7:20 PM 5/28/2010	30 KB	DAT file
data_2010.05.28_WTF.dat	9:58 PM 5/28/2010	30 KB	DAT file
data_2010.05.29_aaarrrgh.dat	12:37 AM 5/29/2010	30 KB	DAT file
data_2010.05.29_#$@*&!!.dat	2:40 AM 5/29/2010	0 KB	DAT file
data_2010.05.29_crap.dat	3:22 AM 5/29/2010	437 KB	DAT file
data_2010.05.29_notbad.dat	4:16 AM 5/29/2010	670 KB	DAT file
data_2010.05.29_woohoo!!.dat	4:47 AM 5/29/2010	1,349 KB	DAT file
data_2010.05.29_USETHISONE.dat	5:08 AM 5/29/2010	2,894 KB	DAT file
analysis_graphs.xls	7:13 AM 5/29/2010	455 KB	XLS file
ThesisOutline!.doc	7:26 AM 5/29/2010	38 KB	DOC file
Notes_Meeting_with_ProfSmith.txt	11:38 AM 5/29/2010	1,673 KB	TXT file
JUNK...	2:45 PM 5/29/2010		Folder
data_2010.05.30_startingover.dat	8:37 AM 5/30/2010	420 KB	DAT file

Type: Ph.D Thesis Modified: too many times Copyright: Jorge Cham www.phdcomics.com

[2]

[2] http://phdcomics.com/comics/archive.php?comicid=1323

Tracking Changes

- The history of a project can be viewed as a series of changes:
  - A unique identifier
  - What changed?
  - When did it change?
  - Who changed it?
  - Why did it change?
- Difficult to manually track multiple files

Git: A Version Control System

A snapshot of the working directory is taken and *commit*ed to the git data base.

- Unique identifier
  - SHA-1 (determined by the files, the author, date, description of change, and the prior history)
- What changed
  - git diff
- Who changed it
  - git blame
- Why did it change
  - git log

- for example:

```
*   e036fc7 - Tue, 26 Nov 2019 15:18:19 -0700  (HEAD -> master)
|\            Merge branch 'startover-again' - Peter DeWitt (BigBlue)
| * 8bc6b04 - Sun, 30 May 2010 09:14:00 -0600  (startover-again)
| |           Starting over, again - Peter DeWitt (BigBlue)
* | 8085a35 - Sun, 30 May 2010 08:37:00 -0600
|/            Starting over - Peter DeWitt (BigBlue)
* e79c166 - Sat, 29 May 2010 11:38:00 -0600
|           Add notes from meeting with Prof Smith - Peter DeWitt (BigBlue)
* 0191da8 - Sat, 29 May 2010 04:47:00 -0600  (tag: v0.1.0)
|           USETHISONE - Peter DeWitt (BigBlue)
* 8e62afb - Sat, 29 May 2010 04:47:00 -0600
|           woohoo!! - Peter DeWitt (BigBlue)
* 26917f2 - Sat, 29 May 2010 04:16:00 -0600
|           notbad - Peter DeWitt (BigBlue)
* 5afefa4 - Sat, 29 May 2010 03:22:00 -0600
|           crap - Peter DeWitt (BigBlue)
* cfe6a5f - Sat, 29 May 2010 02:40:00 -0600
|           #$@*&!! - Peter DeWitt (BigBlue)
* 2c6f677 - Sat, 29 May 2010 00:37:00 -0600
|           aaarrgh - Peter DeWitt (BigBlue)
* 8b77c86 - Fri, 28 May 2010 21:58:00 -0600
|           WTF - Peter DeWitt (BigBlue)
* 57faba6 - Fri, 28 May 2010 19:20:00 -0600
|           huh? - Peter DeWitt (BigBlue)
* 67b675d - Fri, 28 May 2010 19:17:00 -0600
|           callibrate - Peter DeWitt (BigBlue)
* bbe5171 - Fri, 28 May 2010 17:43:00 -0600
|           re-re-test - Peter DeWitt (BigBlue)
* 740af35 - Fri, 28 May 2010 16:29:00 -0600
|           re-test - Peter DeWitt (BigBlue)
* a1dfd5b - Fri, 28 May 2010 15:37:00 -0600
            my test data and analysis - Peter DeWitt (BigBlue)
```

What changed?

```
diff --git a/data.dat b/data.dat
index 88c8fc0..55fa6d2 100644
--- a/data.dat
+++ b/data.dat
@@ -1,2 +1,3 @@
 1,2,3
 5,6,7
+1,1,1,2,2,3,4,5,7,9,16,...
diff --git a/notes_metting_with_prof_smith.txt b/notes_metting_with_
new file mode 100644
index 0000000..e69de29
```

- Diffs are easier to see in several GUI thanks to color coding. More on this later.

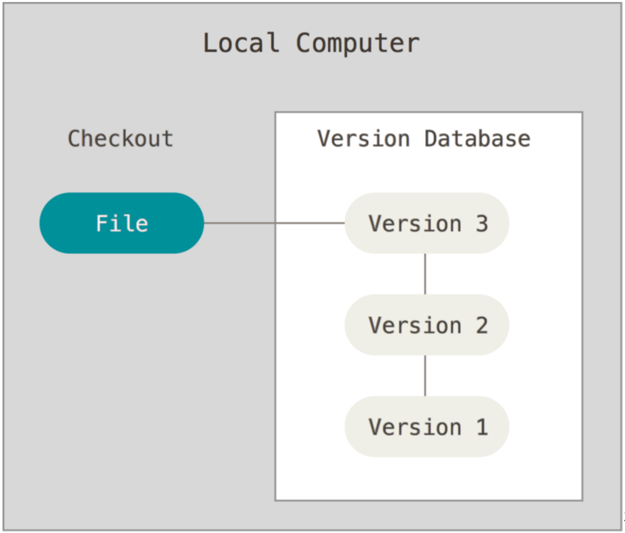
Skipping over errors

```
*   e036fc7 - Tue, 26 Nov 2019 15:18:19 -0700  (HEAD -> master)
|\            Merge branch 'startover-again' - Peter DeWitt (BigBlue
| * 8bc6b04 - Sun, 30 May 2010 09:14:00 -0600  (startover-again)
| |           Starting over, again - Peter DeWitt (BigBlue)
* | 8085a35 - Sun, 30 May 2010 08:37:00 -0600
|/            Starting over - Peter DeWitt (BigBlue)
* e79c166 - Sat, 29 May 2010 11:38:00 -0600
|           Add notes from meeting with Prof Smith - Peter DeWitt (
* 0191da8 - Sat, 29 May 2010 04:47:00 -0600  (tag: v0.1.0)
|           USETHISONE - Peter DeWitt (BigBlue)
```

Conceptual Git and Version Control

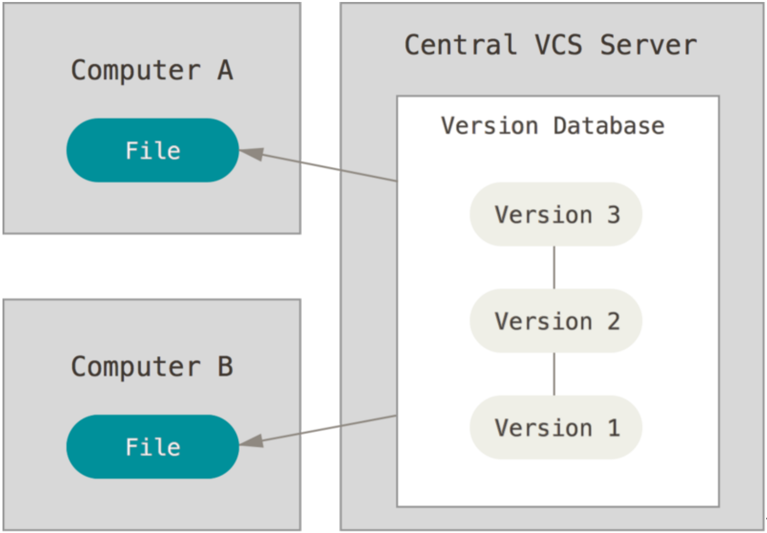
- Local Version Control
- Centralized Version Control System
- Distributed Version Control System
- Short History and Design of Git

Local Version Control System

[3]

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

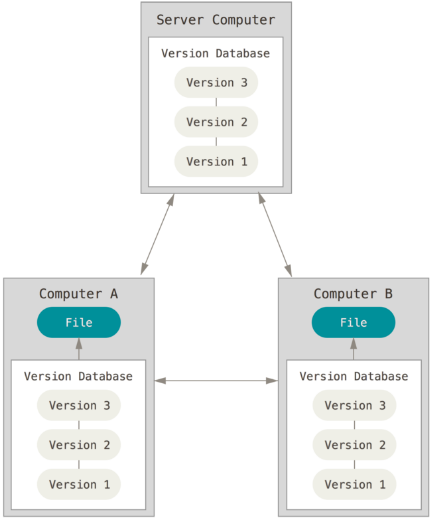
Centralized Version Control System



[4]

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

Distributed Version Control System



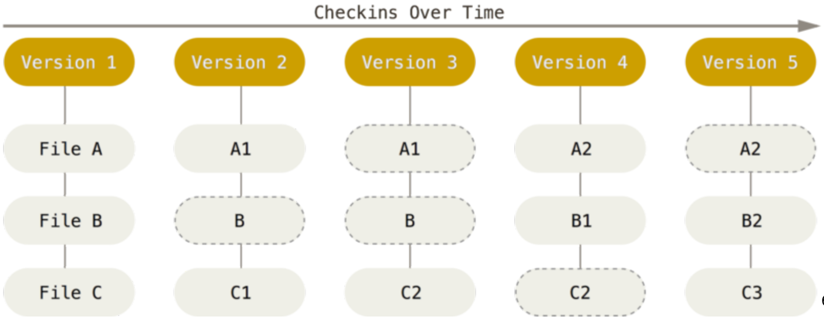
[5]

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

Short History of Git

- Original Author: Linus Torvalds
- Version 0.99 released July 2005
- Current version 2.24
- Linux vs BitKeeper
- Linux dev community sets out to develop their own DVCS with the goals of:
    - speed
    - simple design
    - strong support for non-linear development (thousands of parallel branches)
    - fully distributed
    - able to handle large projects, line the Linux kernel, efficiently

Snapshots; Not Differences



[6]

[6]https://git-scm.com/book/en/v2/Getting-Started-What-is-Git%3F

Nearly Every Operation is Local

- Most operations are local
  - No need to talk to other computers on a network
  - The entire project history is on your local machine
- You can work off vpn
- You can work offline

Git Has Integrity

- Everything is check-summed (remember the sha?)
- You cannot lose information in transit nor get a file corruption without git being able to detect it
- git stores everything in its database not by file name but by the has value of its contents
- git has been designed for multiple parallel development, i.e., multiple programmers contributing to one project non-linearly in time

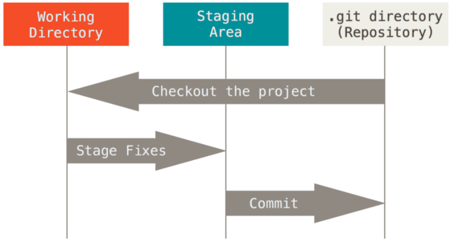
Git Generally Only Adds Dat

- Nearly all actions in git add data to the git database
- It is difficult to do anything that is not undoable
- You can lose/corrupt un-committed changes
- It is very difficult to lose anything after a commit, especially with frequent pushes to other repositories

The Three Stages

- Git has three main states that files reside in
  - Committed: data is safely stored in the local database
  - Modified: changed the file(s) but have not committed to the data base yet
  - Staged: marked modified files(s) in current version to go into the next commit snapshot
- This leads to three main sections of a git project
  - the .git directory
  - the working directory
  - the staging area

The Three Stages



Basic Git workflow

- Modify: make changes in the working directory
- Stage: adding snapshots to the staging area
- Commit: take the files are they are in the staging area and stores the snapshot in the .git directory

Git is Free!

- Download from: `https://git-scm.com/download`
- Other options:
  - Linux: install via your favorite package manager
    - apt-get install git
    - yum install git
  - Mac: Xcode command line tools
  - Install from source: `https://github.com/git/git`

Terminal or GUI

- git was built for the terminal
- Graphical User Interface (GUI) for git exist
  - Gitkraken
  - RStudio

Set Up

- You only need to do this once, per machine:

```
git config –global user.name “Firstname Lastname”
git config –global user.email “first.last@institution.xxx”
```

- Use the terminal (git bash shell even on Windows) or some GUIs will support this

Basic Use

- Since this is an R in Data Science Class Let's walk through the use of git in RStudio.
- Please remember that you can interact with git in many different ways:
  - terminal (my prefered method)
  - Gitkraken
  - RStudio
  - ...
- A lit of GUI clients is available at `https://git-scm.com/downloads/guis/`

Working with Remotes

- So far have only looked at working with git locally
- git is *distributed* version control
- remotes are a copies of the repository on
  - another directory on your computer
  - a network drive
  - another computer
  - a repository hosting server
    - gitlab.com
    - github.com
    - bitbucket.org
    - your own institutional server

Working with remotes
Pros and Cons

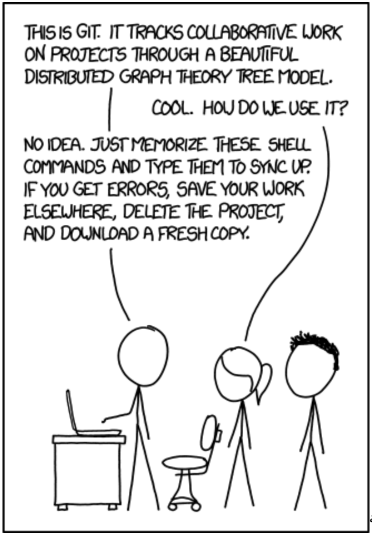
- Pros
  - Collaboration!
  - Every copy of the repo is a whole copy of the history - every copy is a back up for every other copy
  - Additional tools can be used, depending on the remote/server
  - Controls for who can read/write to the project
- Cons
  - Be very careful about publicly hosted repository hosts. *DO NOT COMMIT SENSITIVE DATA TO A REPO*

Setting Up a Remote

- Let’s create a github repository for the example project
- Introduce more git verbs
  - push
  - fetch
  - merge
  - pull

Resources

- git documentation `https://git-scm.com/doc`
- GitKraken `https://www.gitkraken.com/`
- Peter DeWitt: peter.dewitt@cuanschutz.edu

[a]
THIS IS GIT. IT TRACKS COLLABORATIVE WORK ON PROJECTS THROUGH A BEAUTIFUL DISTRIBUTED GRAPH THEORY TREE MODEL.
COOL. HOW DO WE USE IT?
NO IDEA. JUST MEMORIZE THESE SHELL COMMANDS AND TYPE THEM TO SYNC UP. IF YOU GET ERRORS, SAVE YOUR WORK ELSEWHERE, DELETE THE PROJECT, AND DOWNLOAD A FRESH COPY.

[a] https://xkcd.com/1597/