

Paul Rosen

paul.rosen@utah.edu
@paulrosenphd
<https://cspaul.com>



Visualization for Data Science

DS-4630 / CS-5630 / CS-6630

Managing Projects with Git

Git is a distributed **version-control** system

- Terminology: In git-speak, a “version” is called a “commit.”
- Git keeps track of the history of your commits, so you can go back and look at earlier versions or just give up on the current version and go back to some earlier version.
- Can be used to implement a variety of software configuration management models and workflows

Git is a **distributed** version-control system

- You keep your files in a *repository* on your local machine.
- You synchronize your repository with a remote repository on a server (in our case, GitHub).
 - You protect your code from system crashes by synchronizing with the server.
 - If you move from one machine to another, you can pick up the changes by synchronizing with the server.
 - If you work on a team, other people's uploads can be synchronized using the server.

Git Tools

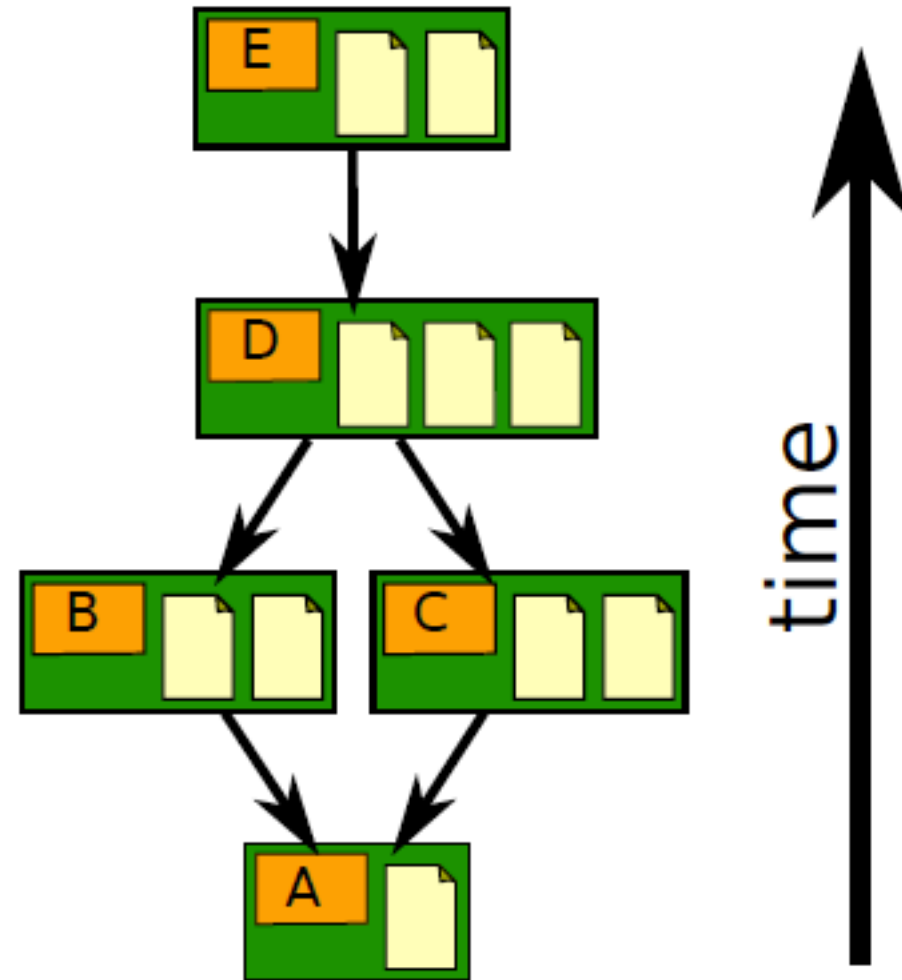
- A collection of many tools
 - Very flexible
- You can do anything the model permits
 - Including shooting yourself in the foot
- Need to understand the underlying model

Groups of **Git** commands

- Setup and branch management
 - **init**, **checkout**, **branch**, **clone**
- Modify
 - **add**, **delete**, **rename**, **commit**
- Get information
 - **status**, **diff**, **log**
- Create reference points
 - **tag**, **branch**
- Synchronization with remote
 - **push**, **pull**, **fetch**, **sync**

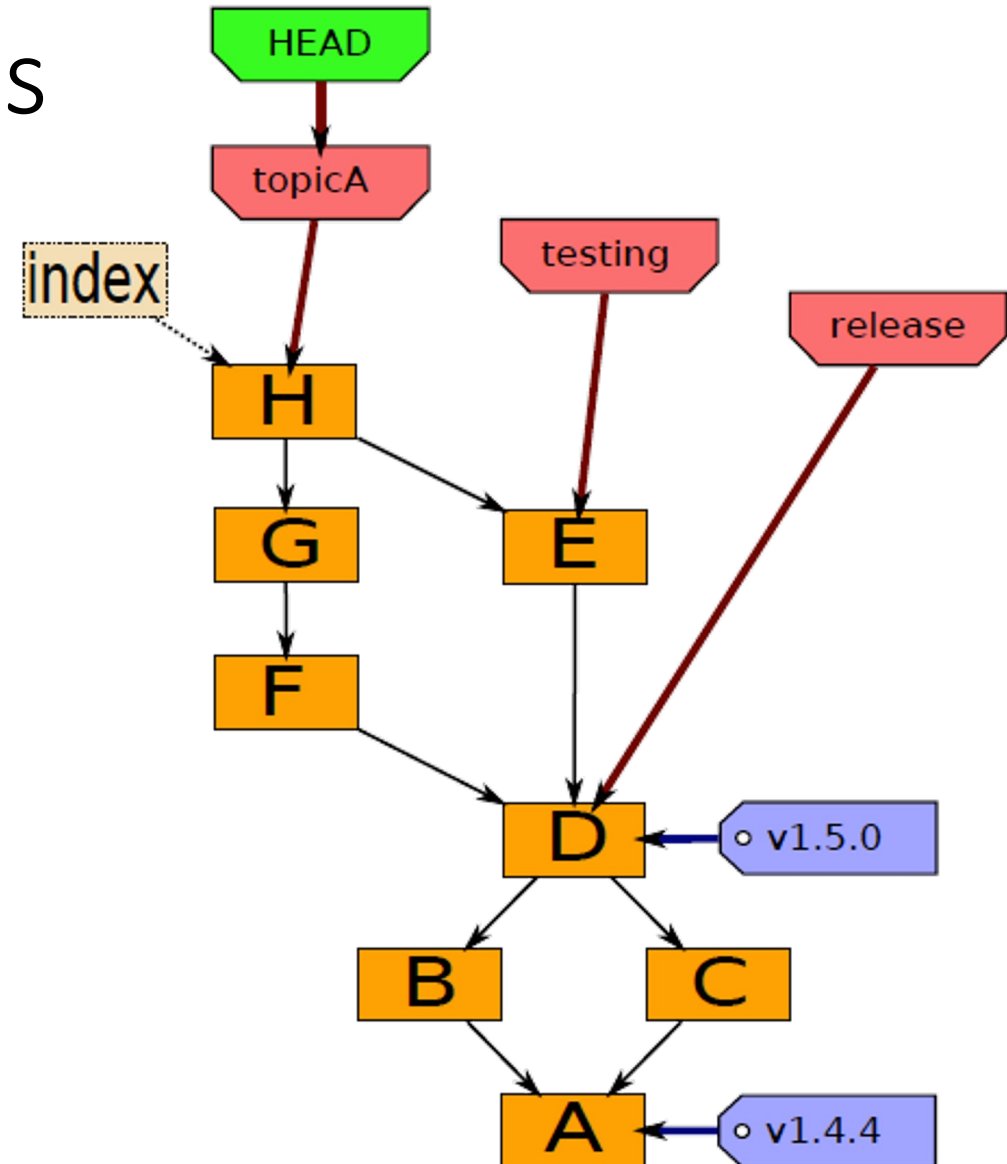
Repository Contains

- files & directories
- commits
- ancestry relationships

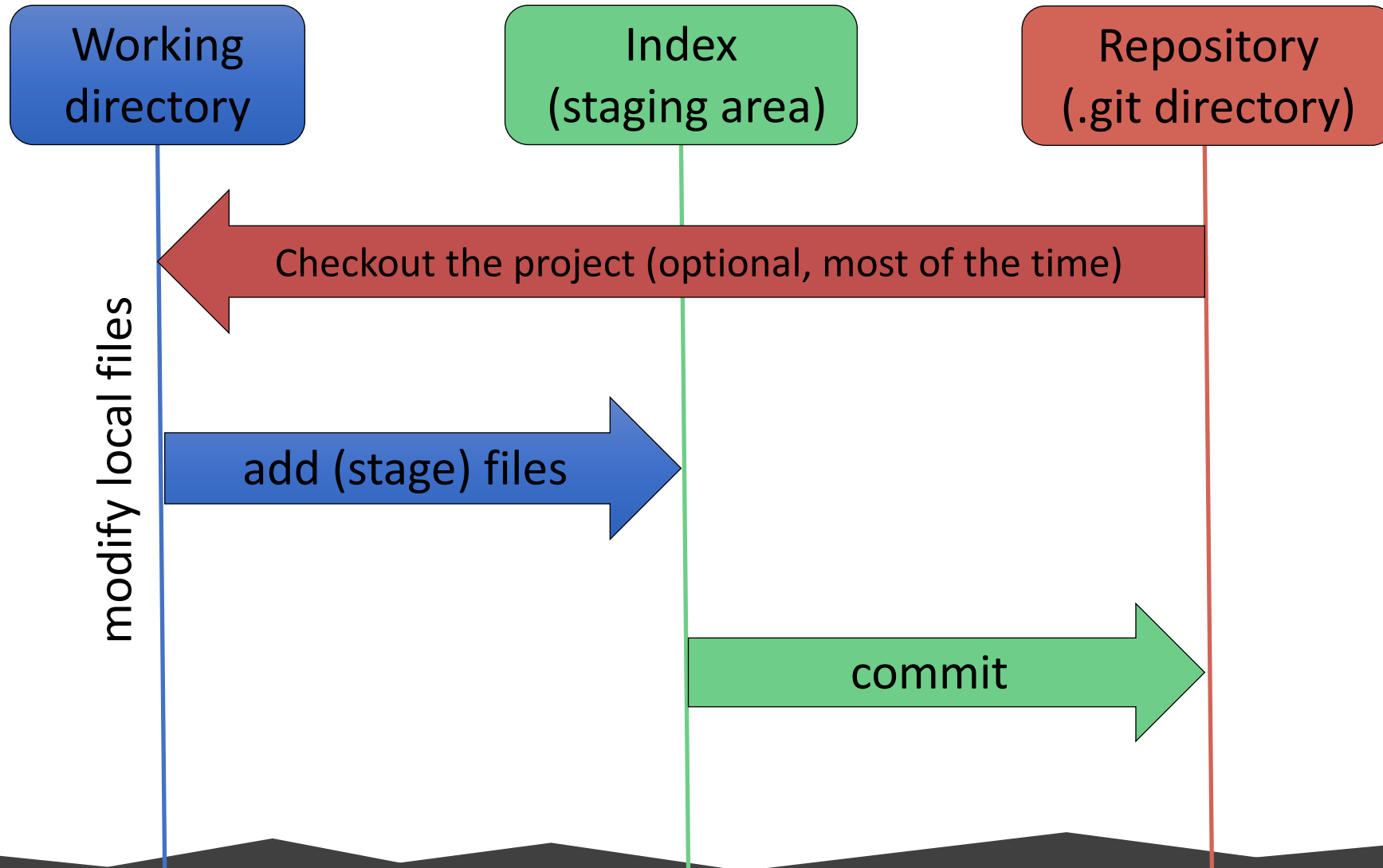


Ancestry graph features

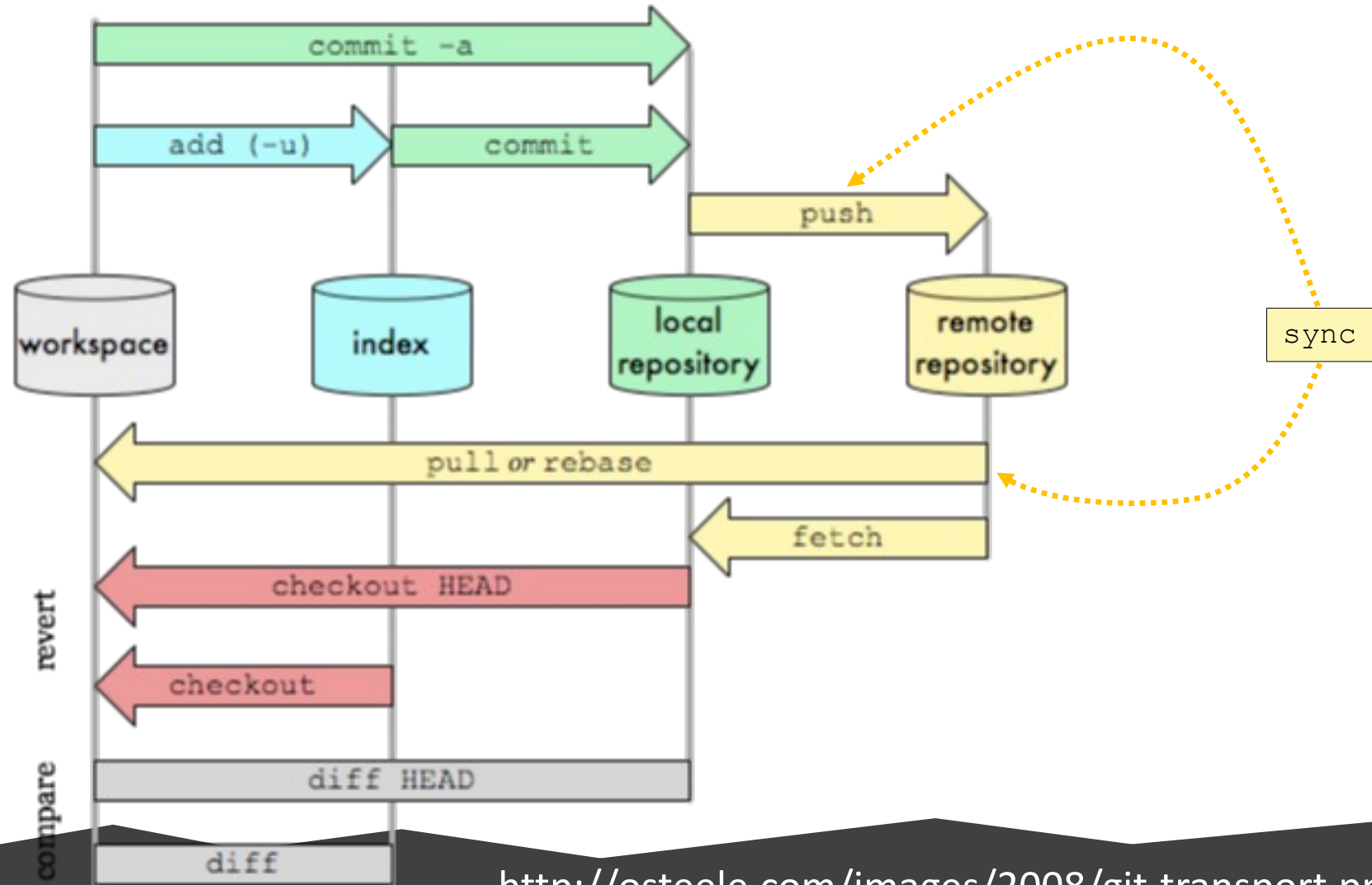
- form a directed acyclic graph (DAG)
- **Commits**
 - Snapshots of file status
- **Tags**
 - identify versions of interest
 - including “releases”
- **Branches**
 - divergent path for source code modification
- **HEAD**
 - is current checkout
 - usually points to a branch
- **Index**
 - “staging area”
 - what is to be committed



Local Operations



Git transport commands



Git Software

- Windows
 - Git command line tools – <https://git-scm.com/download/win>
 - Git GUI – <https://tortoisegit.org/> (also requires download of command line tools)
- MAC
 - Install xcode and the command-line tools
 - <https://developer.apple.com/xcode/>
 - <http://railsapps.github.io/xcode-command-line-tools.html>
- Linux
 - git should already be installed. If not, use the appropriate package manager (e.g. apt or yum) to install it.

Getting Started

- Create a GitHub account, if you don't already have one (<https://github.com/>)
 - GitHub Education account is optional (https://education.github.com/discount_requests/new)
- Checkout the assignments for link to setup your repositories
- Once the repository is created (this can take a few minutes) determine the remote path and pick a local directory for code.

Finding Remote Path

USFDataVisualization / s19-PaulRosenPhD Private

Unwatch 1 Star 0 Fork 0

Code Issues 0 Pull requests 0 Projects 0 Wiki Insights Settings

s19-PaulRosenPhD created by GitHub Classroom Edit

Manage topics

1 commit 1 branch 0 releases 1 contributor GPL-3.0

Branch: master New pull request Create new file Upload files Find file Clone or download

PaulRosenPhD Initial commit Latest commit 2962491 3 days ago

LICENSE	Initial commit	3 days ago
README.md	Initial commit	3 days ago

README.md

Data-Visualization-Skeleton

USF CIS 4930/6930-002: Data Visualization (Skeleton Code)

USFDataVisualization / s19-PaulRosenPhD Private

Unwatch 1 Star 0 Fork 0

Code Issues 0 Pull requests 0 Projects 0 Wiki Insights Settings

s19-PaulRosenPhD created by GitHub Classroom Edit

Manage topics

1 commit 1 branch 0 releases 1 contributor GPL-3.0

Branch: master New pull request Create new file Upload files Find file Clone or download

PaulRosenPhD Initial commit Latest commit 2962491 3 days ago

LICENSE	Initial commit	3 days ago
README.md	Initial commit	3 days ago

README.md

Data-Visualization-Skeleton

USF CIS 4930/6930-002: Data Visualization (Skeleton Code)

Clone with HTTPS Use SSH

Use Git or checkout with SVN using the web URL.

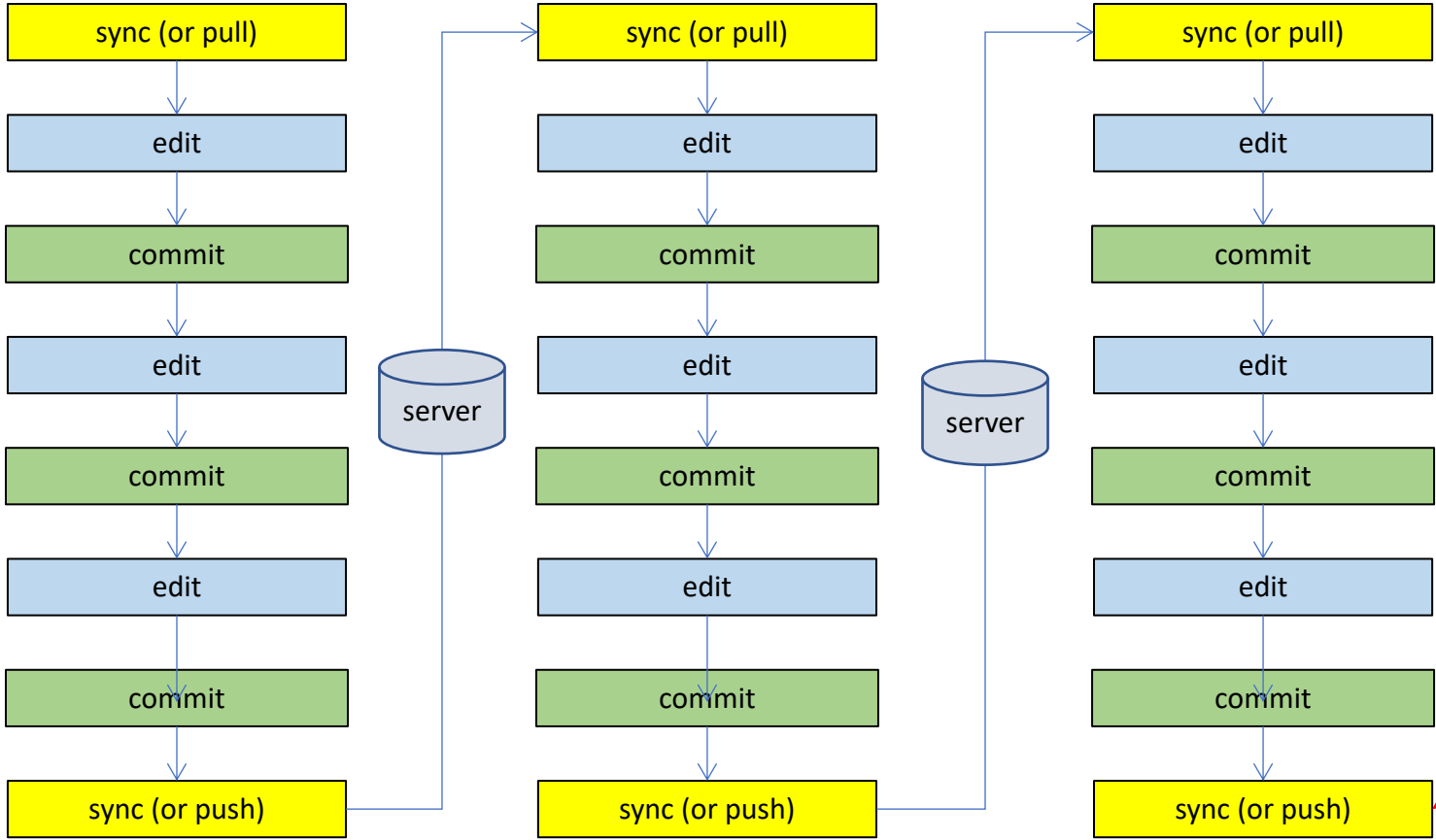
<https://github.com/USFDataVisualizati>

Open in Desktop Download ZIP

Sample session commands

```
> git clone <remote_path> <local_directory>  
> cd <local_directory>  
> git pull  
> touch newfile.txt  
> git add newfile.txt  
> git commit -m "added a new file"  
> git push
```

Suggested workflow



This is what we grade from!

stop working / start working

stop working / change computers

References

- <http://book.git-scm.com/index.html>
- <http://excess.org/article/2008/07/ogre-git-tutorial/>
- <http://www-cs-students.stanford.edu/~blynn/gitmagic/>
- <http://progit.org/book/>
- <http://www.geekherocomic.com/2009/01/26/who-needs-git/>
- Many YouTube videos
 - ex. <https://www.youtube.com/watch?v=HVsySz-h9r4>

RECOMMENDED WATCHING

Git & GitHub Crash Course

https://www.youtube.com/watch?v=SWYqp7iY_Tc



RECOMMENDED READING

The Grammar of Graphics: Chapters 8-9 (pp. 155-254)



