



THE UNIVERSITY  
of EDINBURGH



# Version Control with Git

Instructor: Emmanouil (Manos) Farsarakis  
Imperial College London, April 2015

# Reusing this material



This work is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License.

[http://creativecommons.org/licenses/by-nc-sa/4.0/deed.en\\_US](http://creativecommons.org/licenses/by-nc-sa/4.0/deed.en_US)

This means you are free to copy and redistribute the material and adapt and build on the material under the following terms: You must give appropriate credit, provide a link to the license and indicate if changes were made. If you adapt or build on the material you must distribute your work under the same license as the original.

Note that this presentation contains images owned by others. Please seek their permission before reusing these images.

# Lesson Plan

- What is version control?
- Why do we need it?
- Why “Git”?
- How does it work?

# What is Version Control?

# What is Version Control?

Revision control, also known as version control and source control (and an aspect of software configuration management), is the management of changes to documents, computer programs, large web sites, and other collections of information. Changes are usually identified by a number or letter code, termed the "revision number", "revision level", or simply "revision". For example, an initial set of files is "revision 1". When the first change is made, the resulting set is "revision 2", and so on. Each revision is associated with a timestamp and the person making the change. Revisions can be compared, restored, and with some types of files, merged.



WIKIPEDIA

# What is Version Control?

Version control is the management of changes to documents, computer programs, large web sites, and other collections of information.

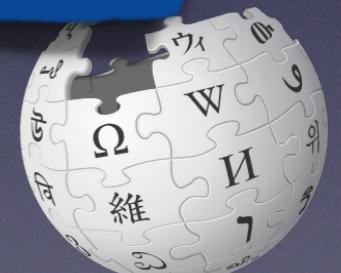


WIKIPEDIA

# What is Version Control?

Version control is the management of changes to documents, computer programs, large web sites, and other collections of information.

Changes are usually identified by a number or letter code, termed the "revision number", "revision level", or simply "revision".



WIKIPEDIA

# What is Version Control?

Version control is the management of changes to documents, computer programs, large web sites, and other collections of information.

Changes are usually identified by a number or letter code, termed the

Example:

An initial set of files is "revision 1".

When the first change is made, the resulting set is "revision 2", and so on.



WIKIPEDIA

# What is Version Control?

Changes are usually identified by a number or letter code, termed the "revision number", "revision level", or simply "revision".

Example:

An initial set of files is "revision 1".

Revisions can be compared, restored, and with some types of files, merged.



WIKIPEDIA

# What is Version Control?

Think “Time-capsule”,  
but... “with benefits”.

push

add

pull

branch

revert

commit



archer | epcc



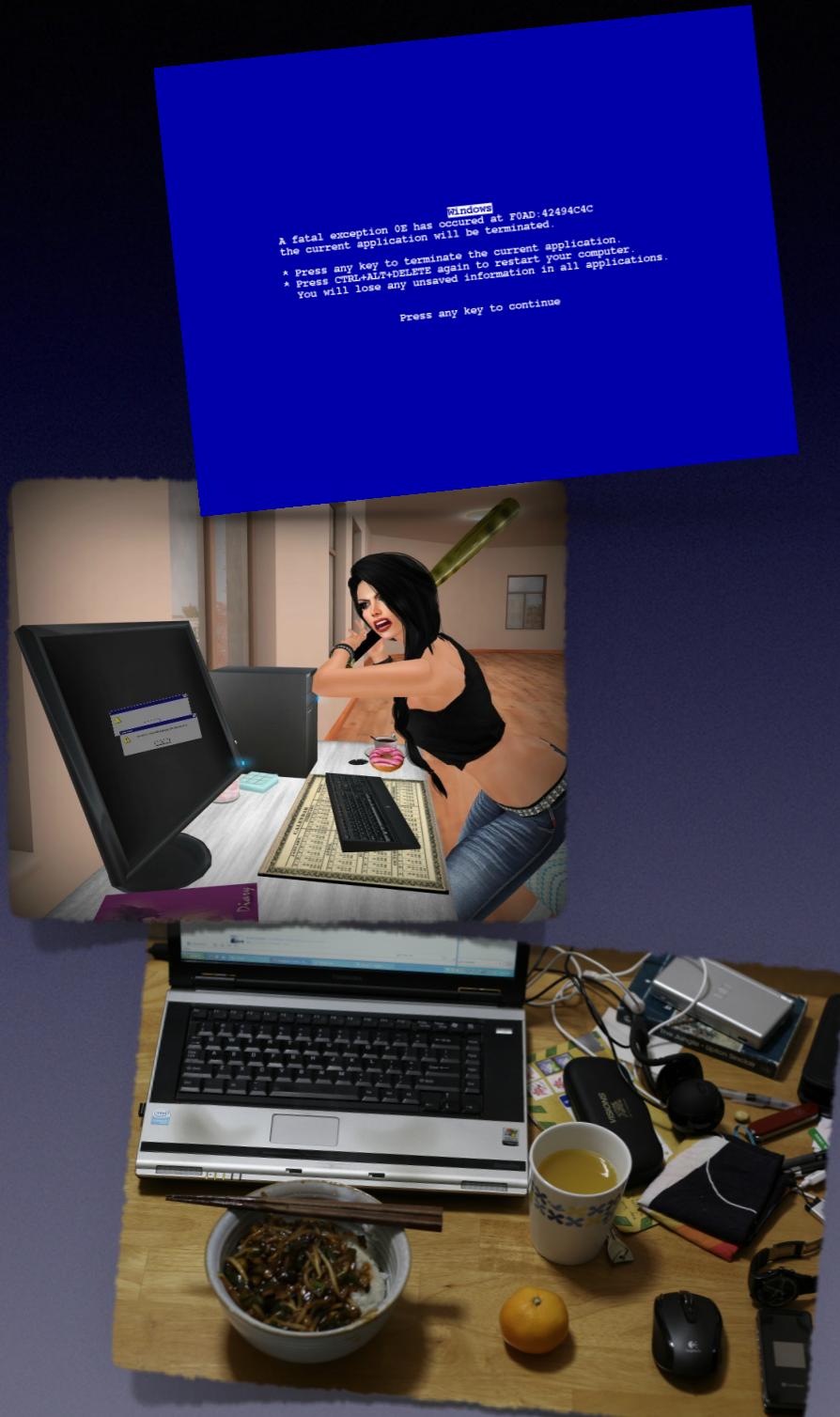
THE UNIVERSITY  
of EDINBURGH

software carpentry

# Why use it?

# Version Control with Git

## Why use it?



A screenshot of a GitHub commit history for a repository named "git-source-term". The history shows 17 commits over 8 days, with the most recent commit being a merge from "archer-training". The code snippets in the commits show various C-like syntax, including `IDRNS assume_aligned` and `omp simd` directives.



# Why Git?

...and not  
subversion?

...or mercurial?

# Why Git?

- Some may prefer other VC systems

**BUT**

- Knowing Git WILL help you use any other VC system.
- Git Hub is becoming increasingly popular

# Why Bitbucket?

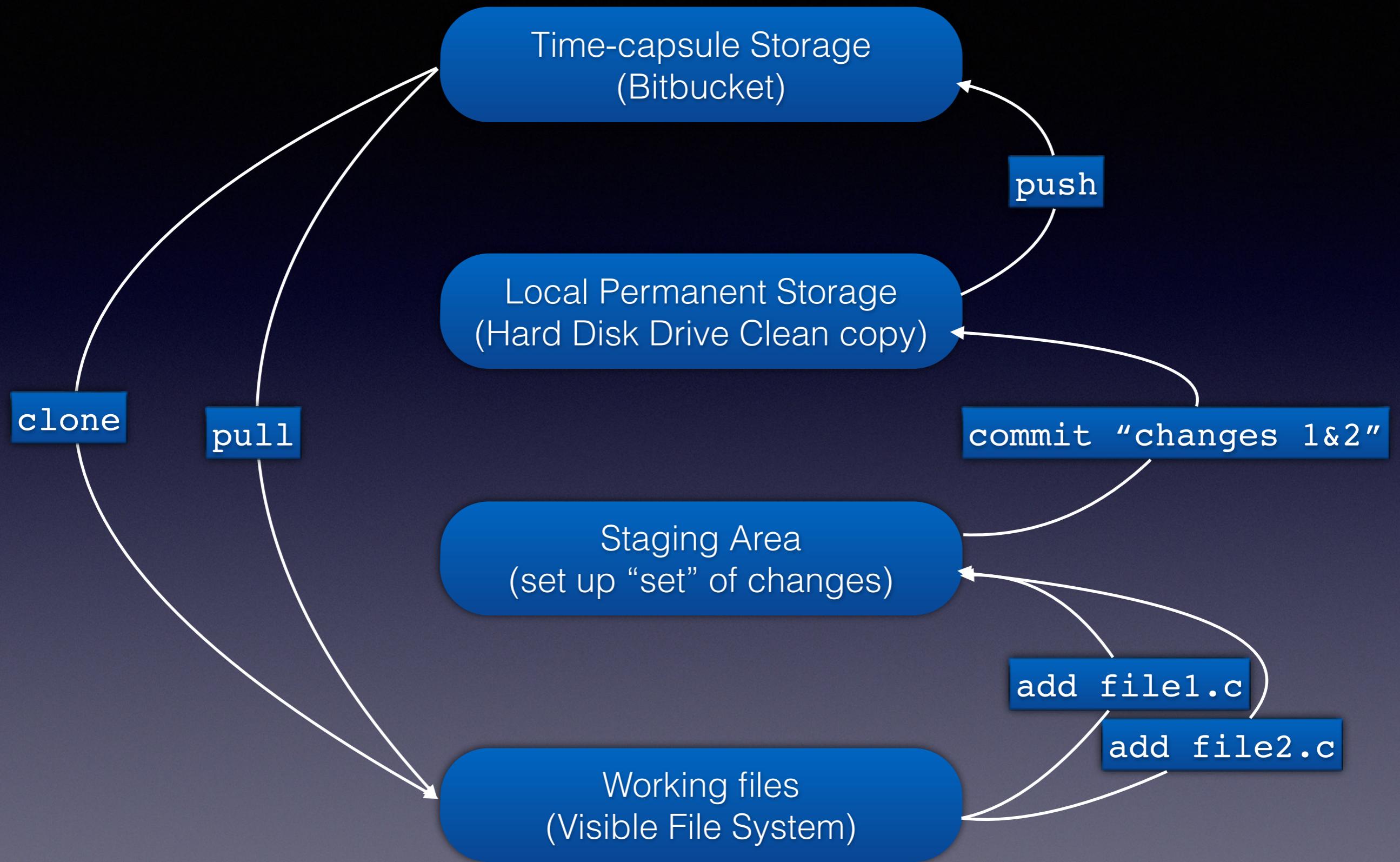
- It offers PRIVATE repositories by default

You can use  
others (as do we):

- [Bitbucket](#)
- [GitHub](#)
- [GitLab](#)

A quick overview  
before we get started

# Version Control with Git



# Version Control

# Quiz Time!

# Git Basics

Which is the best way to place to save your work?

- G** A Git repository
- R** An SVN repository
- O** A hard drive
- W** Dropbox

# Git Basics

Which is the best way to place to save your work?

- G** A Git repository
- R** An SVN repository
- O** A hard drive
- W** Dropbox

# Live coding time!

- Setting up a repository
- Day-To-day use
- Sharing
- Conflicts

# Version Control

# Quiz Time!

# Committing Changes to Git

**Which command(s) below would save changes to myfile.txt to my local Git repository?**

- G** git commit -m "my recent changes"
- R** git init myfile.txt && git commit -m "my recent changes"
- O** git add myfile.txt && git commit -m "my recent changes"
- W** git commit -m myfile.txt "my recent changes"



# Committing Changes to Git

**Which command(s) below would save changes to myfile.txt to my local Git repository?**

**G** git commit -m "my recent changes"

**R** git init myfile.txt && git commit -m "my recent changes"

**O** git add myfile.txt && git commit -m "my recent changes"

**W** git commit -m myfile.txt "my recent changes"



# Recovering Older Version of a File

Jennifer has made changes to the Python script that she has been working on for weeks, and the modifications she made this morning “broke” the script and it no longer runs. She has spent ~1hr trying to fix it, with no luck...

Luckily, she has been keeping track of her revisions using Git! Which commands below will let her revoke the last committed version of her Python script called `data_cruncher.py`?

- G** git checkout HEAD
- R** git checkout HEAD data\_cruncher.py
- O** git checkout HEAD~1 data\_cruncher.py
- W** git checkout <unique\_ID of **last** revision> data\_cruncher.py

# Recovering Older Version of a File

Jennifer has made changes to the Python script that she has been working on for weeks, and the modifications she made this morning “broke” the script and it no longer runs. She has spent ~1hr trying to fix it, with no luck...

Luckily, she has been keeping track of her revisions using Git! Which commands below will let her revoke the last committed version of her Python script called `data_cruncher.py`?

G git checkout HEAD

R git checkout HEAD data\_cruncher.py

O git checkout HEAD~1 data\_cruncher.py

W git checkout <unique\_ID of **last** revision> data\_cruncher.py

# Open Science. Basics!

(or... “I’m not a lawyer”.)



# The old way:

A scientist:

- Collects data on local machine
- Writes or modifies a program to analyse that data
- From results, writes and submits paper (may include data - probably not code)
- Time passes.....
- The journal sends her reviews - anonymous handful of people in her field
- She makes changes accordingly - resubmits
- More time.....
- The paper is eventually published

# The old way:

In the end...

- The paper may include a link to an online copy of her data
- The paper is behind a paywall
  - Only users with personal or institutional access will be able to read it
- Where is her code? How can I build on this?

# A new way:

A scientist:

- Stores data in open access repo like figshare or Dryad - gets DOI (digital object identifier).
- Creates an online repo on GitHub (or the like) for her work.
- As she does analysis, she pushes changes and some results to her repo. She also uses the repo for her paper. The repo becomes a hub for collaboration.
- When she's happy with the paper, she submits it to arXiv (or other pre-print server) and invites feedback from peers.
- Feedback may lead to several revisions before paper is submitted to journal.
- Published paper includes links to preprint, repo and figshare.

# A new way:

This way:

- Other scientists can use her work as a starting point for their own research.
- Discovery is accelerated.
- Open work is more widely cited and re-used.

But...

- What does “Open” mean exactly?

# Licenses

## Software:

- GPL
- MIT
- BSD

Unrestricted sharing and modification of code

# Licenses

## Software:

- **GPL** → Modified code must reuse GPL
  - Re-contribution to community
  - Using many codes gets complicated
  - Necessary to force this?
- MIT
- BSD

# Licenses

## Software:

- GPL
- MIT
- BSD

U.S. Law -> May be important to you.

# Licenses

## Software:

### Most important of all:

- Use a license: State it in a file in your project's home directory named LICENSE or LICENSE.TXT
- **Don't write your own license!** (unless maybe you ARE a lawyer...)

# Licenses

## Data and publications:

Six “Creative Commons” (CC) licenses available.

They use combinations of these four basic restrictions:

- BY: Attribution
- ND: No Derivatives
- SA: Share Alike
- NC: NonCommercial

# Example: Software Carpentry

**Software (code): MIT**

**Data (lessons): CC-BY**

...with the purpose or encouraging the  
widest possible re-use.

# Example: This lesson



This work is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License.

[http://creativecommons.org/licenses/by-nc-sa/4.0/deed.en\\_US](http://creativecommons.org/licenses/by-nc-sa/4.0/deed.en_US)

This means you are free to copy and redistribute the material and adapt and build on the material under the following terms: You must give appropriate credit, provide a link to the license and indicate if changes were made. If you adapt or build on the material you must distribute your work under the same license as the original.

Note that this presentation contains images owned by others. Please seek their permission before reusing these images.

# Citation note

CITATION or CITATION.txt

To reference Software Carpentry in publications, please cite both of the following:

Greg Wilson: "Software Carpentry: Lessons Learned". arXiv:1307.5448, July 2013.

```
@online{wilson-software-carpentry-2013,  
  author      = {Greg Wilson},  
  title       = {Software Carpentry: Lessons Learned},  
  version     = {1},  
  date        = {2013-07-20},  
  eprinttype   = {arxiv},  
  eprint      = {1307.5448}  
}
```

Open Science  
Quiz Time!



# Code Licenses

You want to make your code as freely available as possible. Which option is best for you?

- G** No license = everyone can use it!
- R** Use the GPL license
- O** Put a license file in my repo that says “Use freely”
- W** Use the MIT or BSD license

# Code Licenses

You want to make your code as freely available as possible. Which option is best for you?

- G** No license = everyone can use it!
- R** Use the GPL license
- O** Put a license file in my repo that says “Use freely”
- W** Use the MIT or BSD license

# Hosting

You are working on a project and your data has sensitive personal information about real patients' health history. You should...

- G** Host your data locally on University servers
- R** Use Github or a similar alternative
- O** Buy yourself a domain and pay for ISP hosting
- W** Save the data on my laptop only



# Hosting

You are working on a project and your data has sensitive personal information about real patients' health history. You should...

- G** Host your data locally on University servers
- R** Use Github or a similar alternative
- O** Buy yourself a domain and pay for ISP hosting
- W** Save the data on my laptop only

# Data Licenses

You want people to be able to re-use your work both for free and commercially, to have to cite the original, but NOT be able to make changes to the original.

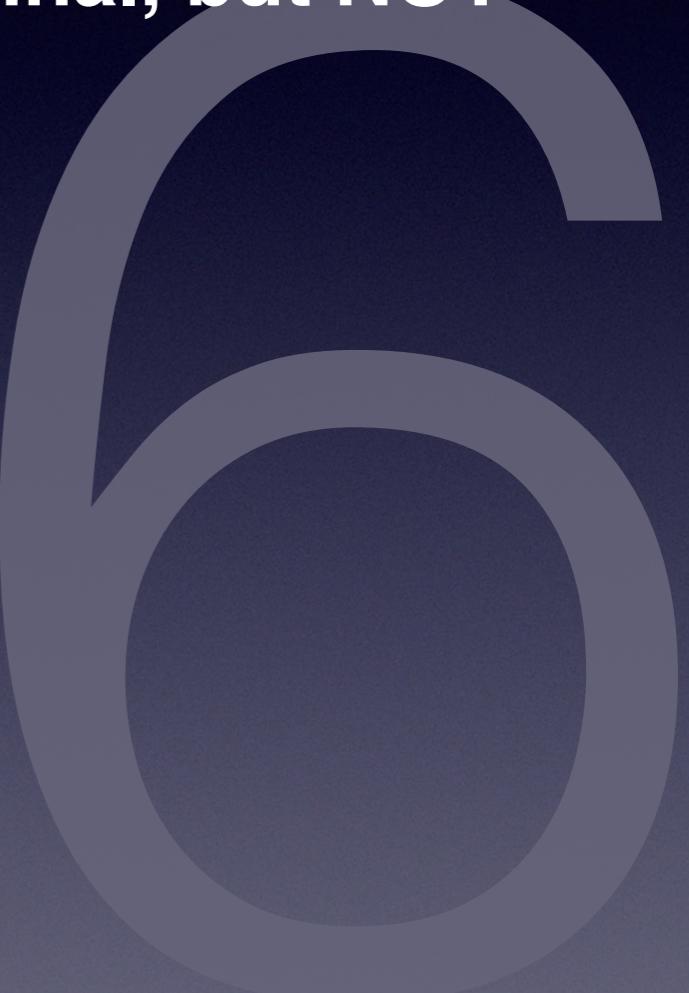
The appropriate license for your work is:

G CC

R CC-BY-ND

O CC-BY

W CC-BY-NC-ND

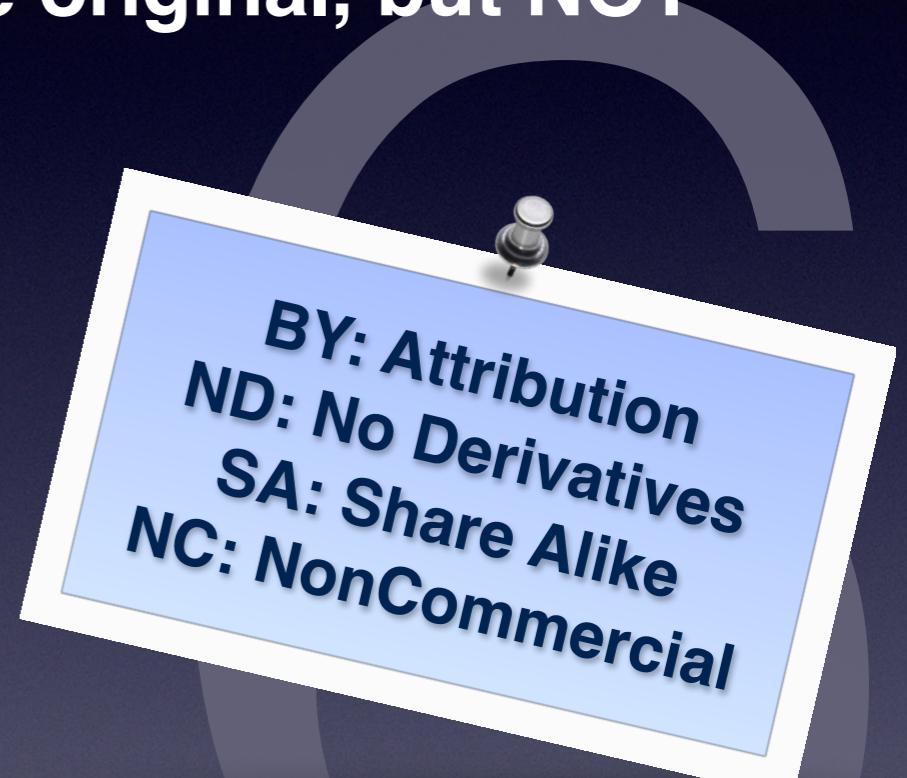


# Data Licenses

You want people to be able to re-use your work both for free and commercially, to have to cite the original, but NOT be able to make changes to the original.

The appropriate license for your work is:

- G CC
- R CC-BY-ND
- O CC-BY
- W CC-BY-NC-ND



# Data Licenses

You want people to be able to re-use your work both for free and commercially, to have to cite the original, but NOT be able to make changes to the original.

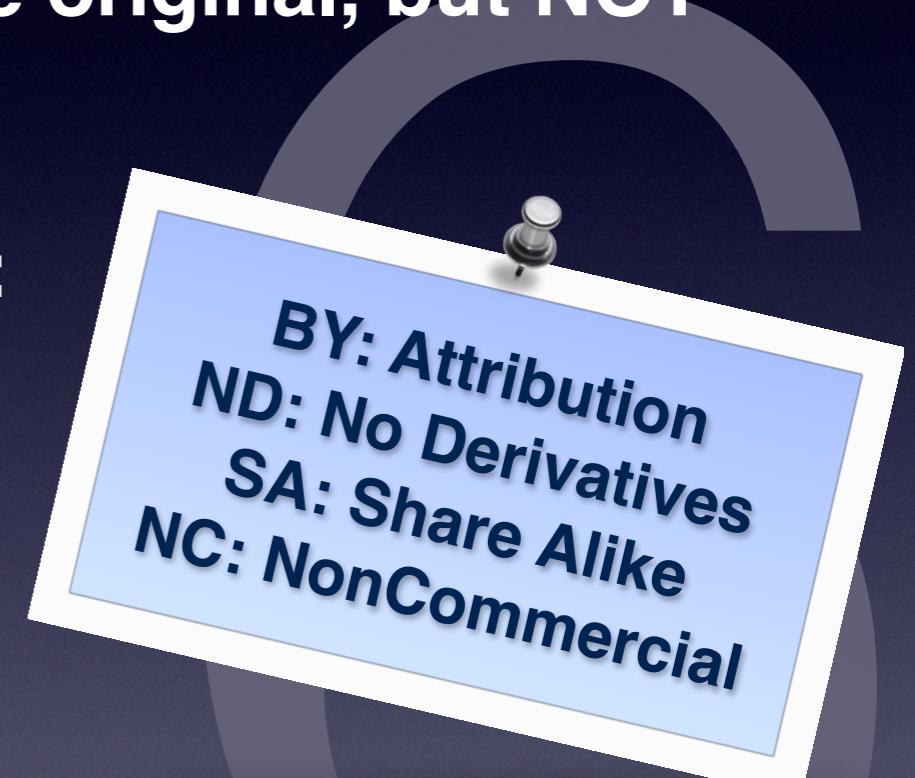
The appropriate license for your work is:

G CC

R CC-BY-ND

O CC-BY

W CC-BY-NC-ND



# Thank you!



# Thank you!

and remember...

Good point  
I learned...  
(and/or)  
I enjoyed...

Bad Point  
...was confusing  
(and/or)  
...bothered me.