

# 'Git'-ing better: Collaborating on your research with version control and GitHub

Luke & Daiva

2014-12-05

# Welcome back!

## Purpose:

To review Git in more detail, give you more practice exercises that mimic what you may do in your day-to-day research activities, and show you how GitHub can be used for collaboration.

## Caveat: We aren't here to teach statistics

Need help with stats? Use these resources!

- U of T Statistical Consulting Services ([click here](#))
- <http://www.stackoverflow.com>
- <http://stats.stackexchange.com>

## Notes and help during this workshop

Go to this website:

<https://etherpad.mozilla.org/dnsWorkshops>

## Version control refresher

# Version control refresher

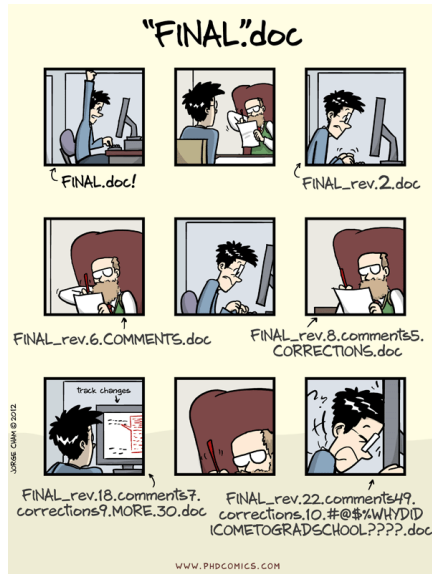


Figure 1:

## What is version control<sup>1</sup> (VC)

- Keeps history of all changes done to files in a folder
- Like a big track changes
- Like your experiment logbook/journal (basic science)
- Can revert to previous change
- Don't have to worry about losing what you wrote!

---

<sup>1</sup>See the Git website ([click here](#)) for more detail.

# What is version control<sup>1</sup> (VC)

- Keeps history of all changes done to files in a folder
- Like a big track changes
- Like your experiment logbook/journal (basic science)
- Can revert to previous change
- Don't have to worry about losing what you wrote!

## Importance of VC!

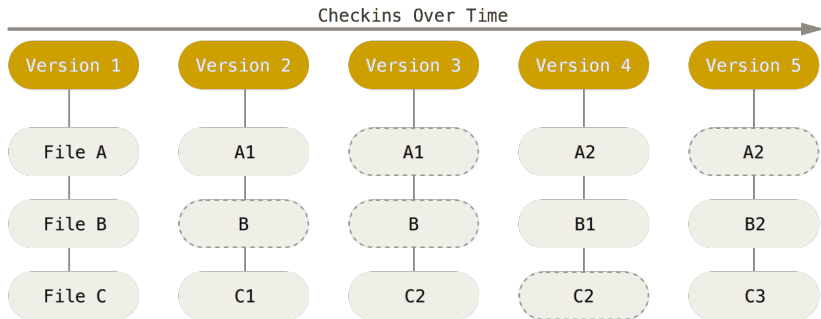
- Future of journals and retractions
- Requests for data and code
- Transparency, scientific rigor
- Protect against accusations of fraud

---

<sup>1</sup>See the Git website ([click here](#)) for more detail.



# Visualization of VC<sup>2</sup>



---

<sup>2</sup>Taken from the Git site ([click here](#))

# Using Git

Before we start... The command line is **not** something to be afraid of!! Open up the terminal (Mac/Linux) or Git Bash (Windows).

# Using Git

Before we start... The command line is **not** something to be afraid of!! Open up the terminal (Mac/Linux) or Git Bash (Windows).

Open up your terminal (Mac or Linux) or Git-Bash (Windows).

```
git config --global user.name "Your Name"
git config --global user.email "you@some.domain"
git config --global color.ui "auto"
git config --global core.editor "your_editor"
git config --list
```

## Download our GitHub repository

GitHub<sup>3</sup> is a place to store your git repo for several reasons:

- 1 As a backup
- 2 To use across computers
- 3 To share with others

---

<sup>3</sup>Or <http://BitBucket.org>

## Download our GitHub repository

GitHub<sup>3</sup> is a place to store your git repo for several reasons:

- 1 As a backup
- 2 To use across computers
- 3 To share with others

In your terminal/Git-Bash, run:

```
cd ~  
git clone \  
    https://github.com/codeasmanuscript/gitWorkshop.git  
cd gitWorkshop/part2
```

---

<sup>3</sup>Or <http://BitBucket.org>

## Download our GitHub repository

GitHub<sup>3</sup> is a place to store your git repo for several reasons:

- 1 As a backup
- 2 To use across computers
- 3 To share with others

In your terminal/Git-Bash, run:

```
cd ~  
git clone \  
    https://github.com/codeasmanuscript/gitWorkshop.git  
cd gitWorkshop/part2
```

Check out the `cheatsheet.html` file.

---

<sup>3</sup>Or <http://BitBucket.org>

## Create a GitHub account

## Live coding – Create git repo from scratch



## **Main Exercise – Collaborate on a project**

## Main Exercise – Collaborate on a project

- 1 Create a git repository in a new folder to practice in
- 2 Create a SAS (or R) file to run analyses on the dataset `sashelp.fish` (SAS) or `airquality` (R)
- 3 Find the means and run an ANOVA on the dataset (or simpler stuff)
- 4 Commit your changes to git
- 5 Make a fake report on your findings
- 6 Commit
- 7 Create a GitHub repo, push your changes to it
- 8 Fork a friend's repo on GitHub (or someone fork yours)
- 9 Change a friend's files
- 10 Push the changes and make a pull request
- 11 Check out your own pull requests!