DSCI 100 - Introduction to Data Science

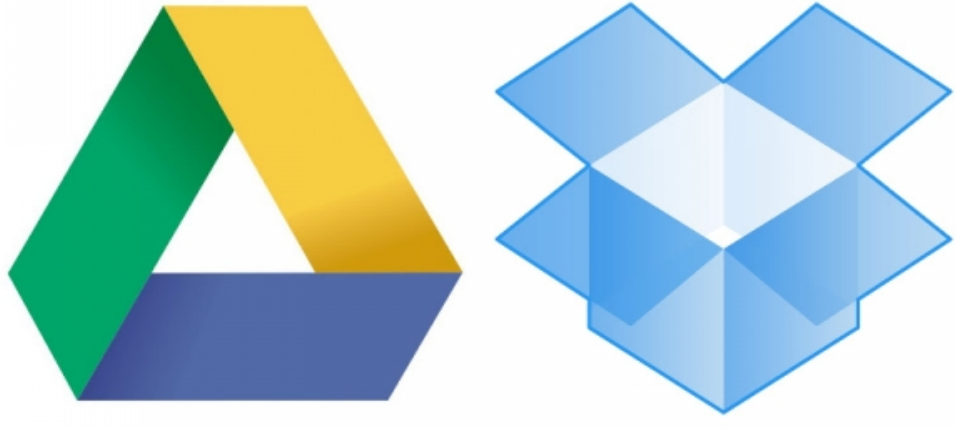
Lecture 5 - Introduction to version control using GitHub

2019-01-31

What is version control?

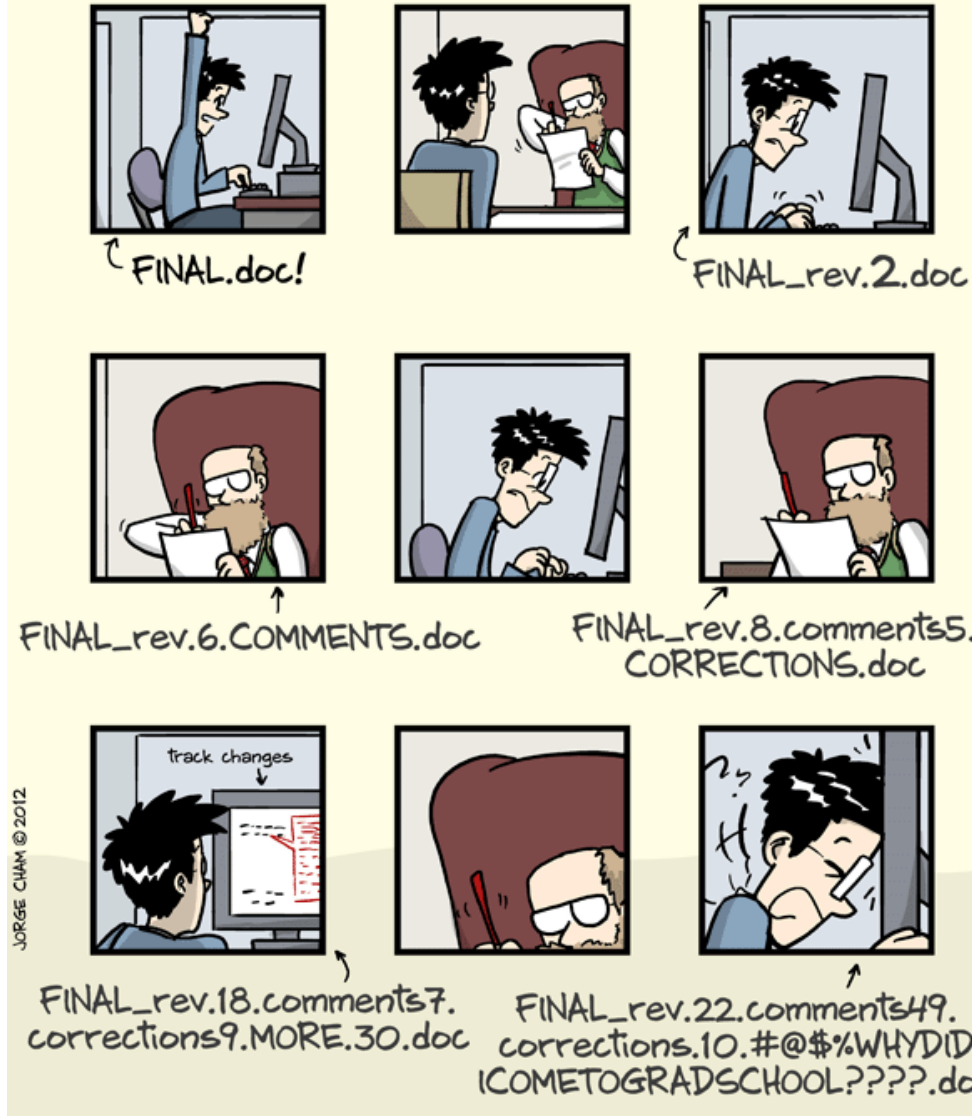
A tool/method to keep track of document versions and work collaboratively with others!

You've probably already used version control



You've probably already used version control

"FINAL".doc



WWW.PHDCOMICS.COM

Reasons to learn another tool for version control (GitHub)

- designed for versioning and sharing code
- can be used to host/build websites/blogs

1. Sign-up for a GitHub account and share your Github username with us:

Please visit https://github.com/ (https://github.com/) and sign-up for a free account (if you don't already have one). And then share your GitHub username with us so that we can get you set up for the group project for this course: https://canvas.ubc.ca/courses/19078/modules/items/1030189 (https://canvas.ubc.ca/courses/19078/modules/items/1030189)

2. Create a GitHub repository

- Let's work through a demo together where we create and edit a repository for a fictionary Data Science project we are going to create.
- **Work in groups of 2** (to help each other out and for the collaboration exercise coming later in the lecture...).

Steps to follow:

One one person's laptop:

1. Go to https://github.com (https://github.com) and make sure you are logged in.
2. Click green “New repository” button. Or, if you are on your own profile page, click on “Repositories”, then click the green “New” button.
3. Choose/set:
4. Repository name: exampleDataProject (or whatever you wish)
5. Public
6. YES Initialize this repository with a README
7. Click big green button “Create repository.”
8. On GitHub, click the settings button on the right and select Collaborators (top left). Enter your partner's GitHub username (they will get an email invitation to access and edit the repository).
9. That's it! You now have a new repository on GitHub that you two can work together on!

3. Editing files directly on GitHub

There are two ways to make changes to your files:

1. Edit files directly on Github (good for text files)
2. Make changes on files that live on a computer (e.g., the server you are working on) and then "push" the changes back to Github (good for code files)

We will try out method 1 today, Tuesday you will learn method 2.

Let's edit a file called `README.md` that contains some information about a fictionary Data Science project we are going to create.

Steps to follow:

in your groups, do this one person at a time...

1. Click on the `README.md` file link
2. Click on the pen tool (right-hand side of document)
3. Add your name as the author to the document (e.g., "author: Tiffany Timbers")
4. Click on the big green button "Commit changes" to save your work

4. Getting a GitHub repository onto your computer

You need to do two things:

1. Introduce yourself to Git on the computer (already installed on the server)
2. Clone (think download) the repository onto the computer (here server)

What is Git? Git is the software on a computer that talks to GitHub.

4.1 Introduce yourself to Git

only need to do this once on your computer/server

1. Open a terminal from the JupyterHub Home/Control Panel (New > Terminal)
2. Type the following to tell Git about yourself (change your name and email to your own):

```
git config --global user.name 'Tiffany Timbers'
git config --global user.email 'tiffany.timbers@stat.ubc.ca'
```

1. Check that you didn't make a typo (if you did then just repeat the commands above):

```
git config --global --list
```

4.2 Clone (think download) the repository onto the computer

1. Visit your repository on GitHub.com
2. Click on the green "Clone or download" button (make sure the pop-up says "Clone with HTTPS")
3. copy the URL to the clipboard
4. Go back to the terminal and type `git clone` and paste the URL and press enter:

```
git clone https://github.com/github_username/repository_name.git
```

5. Add a Jupyter notebook to GitHub

Two big steps:

1. Create (or copy) a Jupyter notebook into the GitHub repository you cloned to your computer (using JupyterHub's Home/Control Panel)
2. Use the terminal to tell Git to send the changes to GitHub

5.1 Add a Jupyter notebook to the GitHub repository on the computer

1. Go to JupyterHub's Home/Control Panel
2. Navigate inside the GitHub repository you cloned (downloaded) by click on it
3. Create a new Notebook (New > R) there. Give it a name and add a code or Markdown cell to it

5. 2 Use the terminal to tell Git to send the changes to GitHub

But first, some useful unix shell commands for navigating and manipulating the filesystem:

Command	Purpose	Example use
`pwd`	Prints current working directory	`pwd`
`ls`	List contents	`ls Documents`
`cd`	Change directory	`cd Desktop`

In class activities

1. Discuss/brainstorm what are the 3 things you do/think about (as a human) when you navigate your computer's filesystem using Finder, Explorer, Nautilus, etc)
2. Instructor - Demo using unix to navigate filesystem
3. Students - use command line to: a. figure out where you are when you open your command line b. navigate to your Documents folder c. navigate from your Documents folder to your Desktop

5. 2 Use the terminal to tell Git to send the changes to GitHub

1. Go back to the JupyterHub Terminal
2. Type `ls` to see all the files and directories there
3. Type `cd REPOSITORY_NAME` to navigate into that repository (type `pwd` to see you are where you expect to be)
4. Type `git add NOTEBOOK.ipynb` (change `NOTEBOOK.ipynb` to the name of the file you created
5. Type `git commit -m "added a notebook"
6. Type `git push` to send the notebook to GitHub
7. Visit your repository on GitHub.com to see that the notebook made it there!

Attribution

1. Happy Git and GitHub for the useR by Jenny Bryan and the STAT 545 TAs (http://happygitwithr.com/)
2. Software Carpentry (https://software-carpentry.org/), specifically the Unix Shell and Git lessons