\*WEEKLY REVIEW\*

\*PREPARE FOR THE QUIZ\*

**Course goal 1**

1. Use the Unix command line to navigate the computer's filesystem;
2. Define and distinguish between absolute file paths and relative file paths;
3. Acknowledge the importance of version control;
4. Use git's basic functions: clone, add, commit, pull, and push
5. Use Git and GitHub to access, edit and submit MDS homework.
6. Create, edit and run Jupyter notebooks (a type of reproducible literate code document).)



Reading command docs:

man

--help

pwd : prints the absolute path of the current working directory

cd : changes the directories to the directory listed after the command. By default it changes to home directory

ls : list files and directories of a directory. By default it returns for current working directory.

Absolute path vs. Relative path:

* relative: tries to find that location from where we are, rather than from the root of the file system.
* absolute: the entire path from the root directory

When to use them: use the one that’s easier for life!

The importance of (using Github) version control:

- See the editing history and keep versions of documents

- Github can act as a back-up for files housed there

- Gihub can be used to host websites/blogs

- Github has a fantastic search functionality

Difference between Git and Github:

* Git is a revision control system, a **tool** to manage the source code history
* Github is a web-based hosting **service** for Git repositories. Allows sharing, access and store remote copies of the repositories.

**Course goal 2**

1. Create a new repository on GitHub
2. Viewing your Git history and checking out an older version of the file
3. Deal with merge conflicts at the command line
4. Use nbdime to deal with merge conflicts involving Jupyter notebooks

Create a repository:

* Locally: In the folder, git init -> create a new repository on Github -> tell the computer where the remote is
* Remotely: start in github and clone to local

Viewing history:

* on Github through the repo’s code commit view
* on local machine using git log

check the commits: git show SHORTSHA

Comparing different commits:



Restore an older version:

* git restore -s SHORTSHA FILENAME
* then git add/commit/push

Merged Conflicts:

- Happens when the same line(s) are modified.

To fix a merge conflict:

* pull the changes from github
* open the conflict text
* <<<<< HEAD: the change in local
* ===========: the remote change
* fix the change(delete/keep) and add/commit/push

Jupyter notebook:

- .ipynb files are in text files

- encoded in JSON

Problems with version control of Jupyter notebook:

- `git diff` looks horrendous because of the JSON

- manually fixing conflicts is arduous because of JSON

In addition to providing a better diff view for jupyter notebooks, with nbdiff-web [<commit> [<commit>]] [<path>],

**Course goal 3**

1. [Tell Git to ignore irrelevant files using a .gitignore file](https://render.github.ubc.ca/view/ipynb?commit=81b29b8d75542a5831a1f0b8ccfc1fb5cb1917a9&enc_url=68747470733a2f2f7261772e6769746875622e7562632e63612f4d44532d323031392d32302f445343495f3532315f706c6174666f726d732d647363695f73747564656e74732f383162323962386437353534326135383331613166306238636366633166623563623139313761392f6c656374757265732f30335f6c6563747572652d6d6f72652d6769742d6d61726b75702d7765622f30335f6c6563747572652d6d6f72652d6769742d6d61726b75702d7765622e6970796e623f746f6b656e3d414141484e2d597a6773646d753848365153314e59475633456f774851394b436b73356468755f597741253344253344&nwo=MDS-2019-20%2FDSCI_521_platforms-dsci_students&path=lectures%2F03_lecture-more-git-markup-web%2F03_lecture-more-git-markup-web.ipynb&repository_id=37571&repository_type=Repository#1.-Tell-Git-to-ignore-irrelevant-files-using-a-.gitignore-file)

2. [Get a copy of someone else's repo on GitHub by forking it](https://render.github.ubc.ca/view/ipynb?commit=81b29b8d75542a5831a1f0b8ccfc1fb5cb1917a9&enc_url=68747470733a2f2f7261772e6769746875622e7562632e63612f4d44532d323031392d32302f445343495f3532315f706c6174666f726d732d647363695f73747564656e74732f383162323962386437353534326135383331613166306238636366633166623563623139313761392f6c656374757265732f30335f6c6563747572652d6d6f72652d6769742d6d61726b75702d7765622f30335f6c6563747572652d6d6f72652d6769742d6d61726b75702d7765622e6970796e623f746f6b656e3d414141484e2d597a6773646d753848365153314e59475633456f774851394b436b73356468755f597741253344253344&nwo=MDS-2019-20%2FDSCI_521_platforms-dsci_students&path=lectures%2F03_lecture-more-git-markup-web%2F03_lecture-more-git-markup-web.ipynb&repository_id=37571&repository_type=Repository#2.-Get-a-copy-of-someone-else's-repo-on-GitHub-by-forking-it)

3. [Catch up to a GitHub repo you forked once you fall behind](https://render.github.ubc.ca/view/ipynb?commit=81b29b8d75542a5831a1f0b8ccfc1fb5cb1917a9&enc_url=68747470733a2f2f7261772e6769746875622e7562632e63612f4d44532d323031392d32302f445343495f3532315f706c6174666f726d732d647363695f73747564656e74732f383162323962386437353534326135383331613166306238636366633166623563623139313761392f6c656374757265732f30335f6c6563747572652d6d6f72652d6769742d6d61726b75702d7765622f30335f6c6563747572652d6d6f72652d6769742d6d61726b75702d7765622e6970796e623f746f6b656e3d414141484e2d597a6773646d753848365153314e59475633456f774851394b436b73356468755f597741253344253344&nwo=MDS-2019-20%2FDSCI_521_platforms-dsci_students&path=lectures%2F03_lecture-more-git-markup-web%2F03_lecture-more-git-markup-web.ipynb&repository_id=37571&repository_type=Repository#3.-Catch-up-to-a-GitHub-repo-you-forked-once-you-fall-behind)

4. [Use GitHub pages to create and host a website](https://render.github.ubc.ca/view/ipynb?commit=81b29b8d75542a5831a1f0b8ccfc1fb5cb1917a9&enc_url=68747470733a2f2f7261772e6769746875622e7562632e63612f4d44532d323031392d32302f445343495f3532315f706c6174666f726d732d647363695f73747564656e74732f383162323962386437353534326135383331613166306238636366633166623563623139313761392f6c656374757265732f30335f6c6563747572652d6d6f72652d6769742d6d61726b75702d7765622f30335f6c6563747572652d6d6f72652d6769742d6d61726b75702d7765622e6970796e623f746f6b656e3d414141484e2d597a6773646d753848365153314e59475633456f774851394b436b73356468755f597741253344253344&nwo=MDS-2019-20%2FDSCI_521_platforms-dsci_students&path=lectures%2F03_lecture-more-git-markup-web%2F03_lecture-more-git-markup-web.ipynb&repository_id=37571&repository_type=Repository#4.-Use-GitHub-pages-to-create-and-host-a-website)

5. [Set up keys for SSH for use with GitHub](https://render.github.ubc.ca/view/ipynb?commit=81b29b8d75542a5831a1f0b8ccfc1fb5cb1917a9&enc_url=68747470733a2f2f7261772e6769746875622e7562632e63612f4d44532d323031392d32302f445343495f3532315f706c6174666f726d732d647363695f73747564656e74732f383162323962386437353534326135383331613166306238636366633166623563623139313761392f6c656374757265732f30335f6c6563747572652d6d6f72652d6769742d6d61726b75702d7765622f30335f6c6563747572652d6d6f72652d6769742d6d61726b75702d7765622e6970796e623f746f6b656e3d414141484e2d597a6773646d753848365153314e59475633456f774851394b436b73356468755f597741253344253344&nwo=MDS-2019-20%2FDSCI_521_platforms-dsci_students&path=lectures%2F03_lecture-more-git-markup-web%2F03_lecture-more-git-markup-web.ipynb&repository_id=37571&repository_type=Repository#5.-Set-up-keys-for-SSH-for-use-with-GitHub)

6. [Use Markdown, HTML tags and LaTeX to format text in literate code documents.](https://render.github.ubc.ca/view/ipynb?commit=81b29b8d75542a5831a1f0b8ccfc1fb5cb1917a9&enc_url=68747470733a2f2f7261772e6769746875622e7562632e63612f4d44532d323031392d32302f445343495f3532315f706c6174666f726d732d647363695f73747564656e74732f383162323962386437353534326135383331613166306238636366633166623563623139313761392f6c656374757265732f30335f6c6563747572652d6d6f72652d6769742d6d61726b75702d7765622f30335f6c6563747572652d6d6f72652d6769742d6d61726b75702d7765622e6970796e623f746f6b656e3d414141484e2d597a6773646d753848365153314e59475633456f774851394b436b73356468755f597741253344253344&nwo=MDS-2019-20%2FDSCI_521_platforms-dsci_students&path=lectures%2F03_lecture-more-git-markup-web%2F03_lecture-more-git-markup-web.ipynb&repository_id=37571&repository_type=Repository#6.-Use-Markdown,-HTML-tags-and-LaTeX-to-format-text-in-literate-code-documents.)

To ignore irrelevant files: (don’t have them to clutter our view)

- create .gitignore

- list files and folders in the file (one per line)

- save, add, commit

-> then files are ignored.

.gitignore tips and tricks:

* append \*\*/ to the beginning of any file/folder names listed in the .gitignore file to have them ignored in subdirectories within the repo as well
* create a [global .gitignore file](https://help.github.com/articles/ignoring-files/#create-a-global-gitignore) so that you do not have to create the same .gitignore for all your homework repos

Copy someone else’s repo

- Fork

- Clone

Catch up to a Github repo forked



Use Github pages to create and host a website

fork repo - change README - visit URL

Set up keys for SSH for use with Github

* Secure Shell - a common method for remote login to another computer which is secure
* server - a machine you are SSHing into. The server sits and waits to be contacted
* client - usually the machine. The client initiates contact with the server.



SSH key-based authentication

public key + private key

- Keys are generated using ssh-keygen, to make private key and a public key

- make copies of public key and distribute them to other machines

- the other machine uses the public key to encrypt a challenge message to you

- You need to show that you can decrypt the message to demonstrate that you have the associated private key

- if the private key does fall into the wrong hands, the person must still know the password/passphrase to use the private key

Why SSH keys over passwords?

- The private key is much longer than a password.

- A standard now is 4096-bit keys, it’s harder to break the password

Authentication vs. encryption

* The SSH system is purely for authentication: the client needs to prove the server that the client is authorized to access the server. The public key is put in *~/.ssh/authorized\_keys* on the server. The server now grants access to anyone possessing a private key matching one of these public keys.
* Separate from encryption where the data flowing between the client and server, prevents eavesdroppers from listening to client-server communication

R Studio introduction

- An integrated development environment (IDE) for R.

- It’s helpful: code completion, linting, debugging tools

- flexible: has literate programming and notebook-like functionality; has editor and console panes for software development of scripts and packages

It can run Rscripts interactively in RStudio



textual output - printed to the console

graphical output - displayed in the Plots pane

getwd() - get current working directory

Setting the working directory:

* setwd() - set current woring directory
* in the files pane, click the file “set as working directory”
* session menu - set w.d

Sample quiz answers:

1. 2
2. 3
3. direct the current working directory to the pathname input. By default, it goes to the root directory
4. cd ~/LinusTorvalds/Documents/numbers

sh nicenumber.sh

1. It is a system that stores and helps managing the source code change history. It is good because people can retrieve the change history; and it is convenient for sharing and modifying codes with other people
2. c
3. It is copying other’s repository to my own Github to become a repository of mine.
4. it is uploading the changes of the locally edited files to the remote repository
5. it is downloading any online changes of the files in the repository to local device to update the local repository