**Git and Github 101**

Git is a version control software. This means that it allows you to make changes to files (be they text files, scripts, spreadsheets, etc) while git keeps a log of the changes. While it is not necessary to use git with every project you do, it is worth getting familiar with git so that you can use it for bigger projects and group projects.

Github is an online host for files tracked via git. It is very useful for group collaboration on a project and for storing commonly shared files (such as the set of documentation this document is part of).

Git can be downloaded for Windows [here](https://gitforwindows.org/) and for Mac [here](https://git-scm.com/download/mac).

In order to use Github, you will need to make on account on [github.com](https://github.com/).

You can read more about getting started with git and Github [here](https://medium.com/@itswisdomagain/git-101-introduction-to-git-for-newbies-bb14f6f9fc1).

**Initializing and updating a project with git and Github works as follows**:

Let’s assume you are developing an analysis pipeline which involves updating a main script as you test it as well as all of the documentation and accessory files and functions that make the program run.

1. cd into your main project folder where all of your files are stored.
2. Use the command git init to initialize git in your folder. This creates a hidden file called .git which will track all of your file changes.
3. You will then git add and git commit any changes you make to your files, which will work like this:
   1. You make changes to your code which is store in a file called, for example, my\_code.py
   2. You use the command git add my\_code.py which will tell git that you’ve made changes to the file in question and you want to add those changes into a staging area. This is a temporary space where git is recognizing changes you made to your files.
   3. You then use the command git commit -m “Updates my\_code.py” which tells git to commit these changes to a change log. The “-m” indicates that there is a message the follow. A brief message of what was changed should go in quotes after the “-m”.
4. If you would like to track all of your changes and store your files in Github, you will need to do the following:
   1. Go to Github and click the green button at the top of the page labeled “New”
   2. Enter your repository name as prompted. This can be anything you want.
   3. Select if you’d like this to be a private or public repository. If you will be working with other people on this project, it should be public.
   4. Click “create repository”
   5. On the next page you will have two options:
      1. If you had no local repository (a project folder on your local computer), the first option “**…or create a new repository on the command line**” gives you the commands you need to set up a new local repository
      2. If you already have a local repository set up (which you do if you followed steps 1-3), you can use the commands from the second options: “**…or push an existing repository from the command line**”. This will tell Github to look at files from your local project folder and your local .git file and copy those files and change log to Github.
   6. When you make future changes to your files locally, after you git add and git commit those changed, you can then use git push to push the changes to Github.