Quick tips for efficient interaction with the quant club Github repo

repo = repository = folder containing all files
clone = download a copy of everything

- Forking is similar to cloning, but also creates a copy of the repo on your personal Github page. If you wanted to make changes to files in the repo and have those changes incorporated into the master versions, you would typically fork the repo, make those changes, and then submit a pull request to have those changes merged, which I would have to approve. However, I made you all collaborators, which means you can make changes to anything without my approval, which makes it easier.

commit = locking in changes that you've made to files.

- Every time you commit, you save a version of that file that you can go back to later. It's good to commit often if you are making a lot of changes to code, to leave yourself a safety net/bread crumbs (choose your own metaphor) to get back to previous versions if needed

push = add/update files from your local machine to the master version in the cloud (do this when you are done for the day)
pull = update the files on your local machine with the versions in the cloud (do this to start your day)

Github workflow (words version)

1a. If you are just starting – Clone the repo!
    - This will download everything in the repo
    - Save the folder in a sensible place on your computer
    - Open the R project and enjoy!

1b. If you've already cloned the repo – Open the Rproj and pull updates to your machine
    - It's important to always pull new updates first, otherwise when you go to merge the files on your machine with the master branch you can run into conflicts and issues that are solvable but annoying

2. Change/update files
    - Write some code, debug a Shiny app, copy any other files into the folder, rearrange the file structure, whatever!

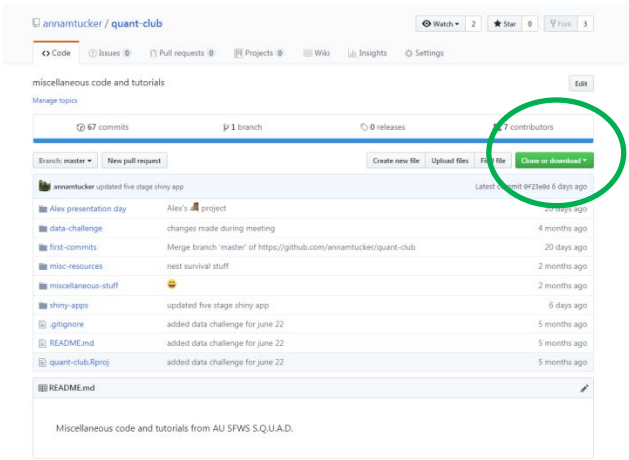
3. Stage those changes to be committed
    - In Rstudio, click the check boxes next to all of the files you changed/added

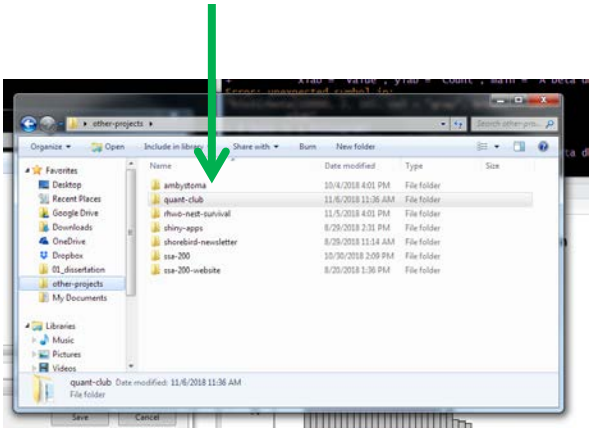
4. Commit those changes
    - You must write a brief commit message that indicates what you did/why you did it
    - Use of emojis is encouraged: https://gist.github.com/rxaviers/7360908

5. Push your commits
    - Update the master branch with what you've changed

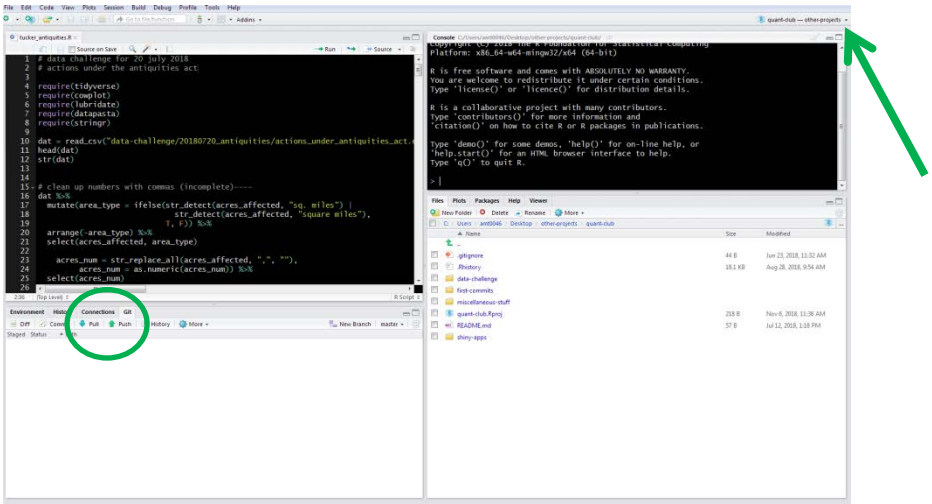
Github workflow (pictures version)

1a. If you are just starting – Clone the repo!

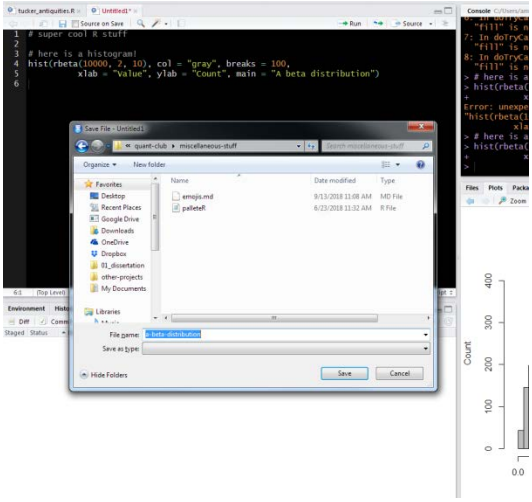




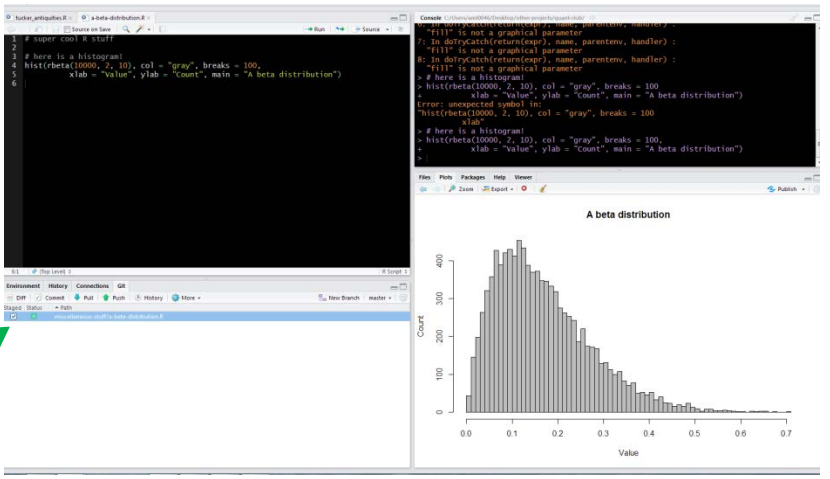
1b. If you've already cloned the repo – open the Rproj and pull updates to your machine



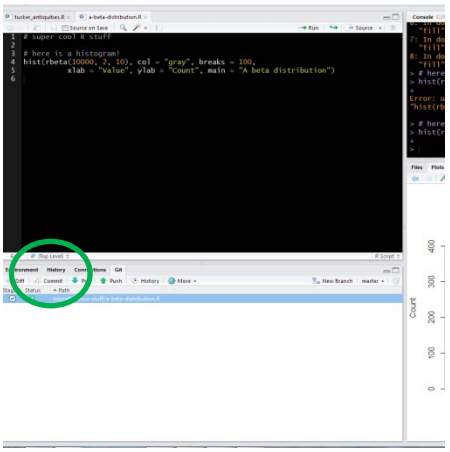
2. Change/update files

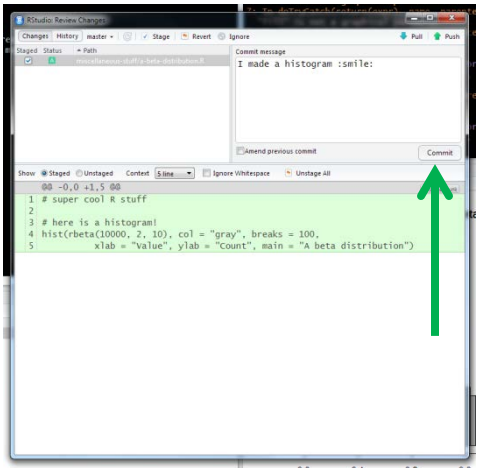


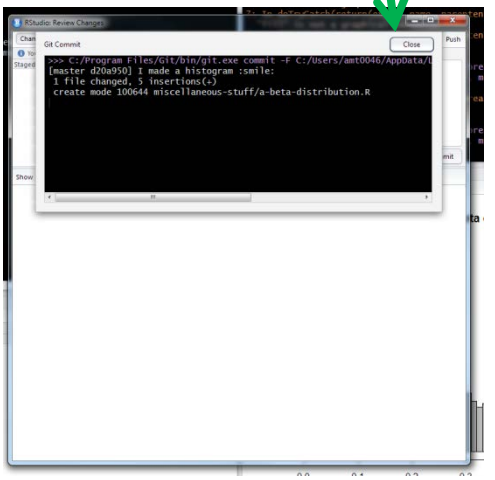
3. **Stage those changes to be committed**



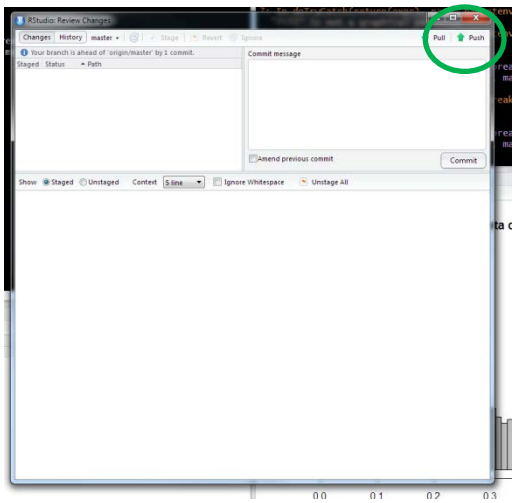
4. **Commit those changes**



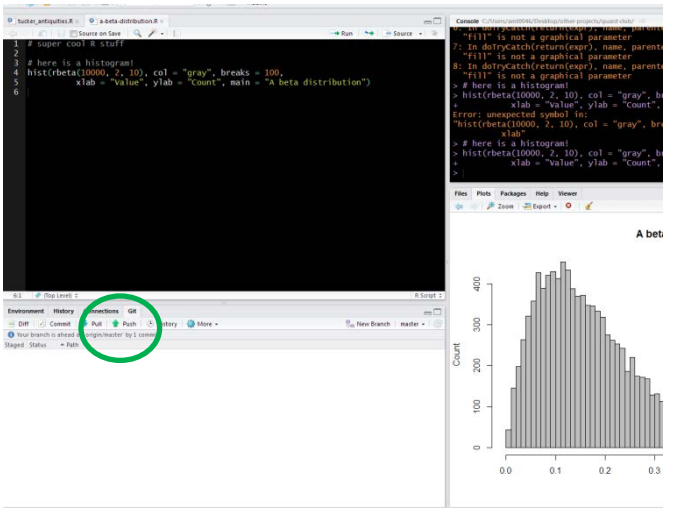




5. **Push your commits**



OR



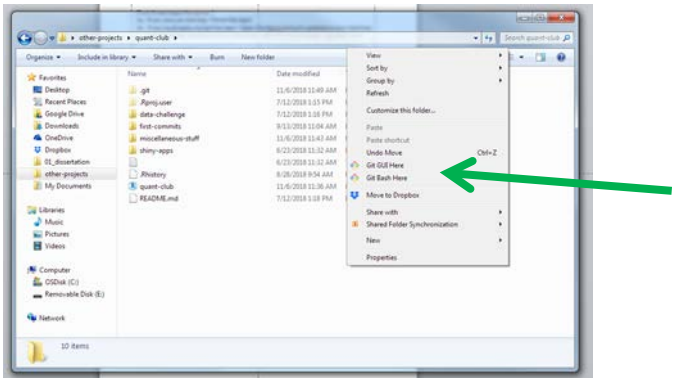
Advanced/non-Rstudio version:

(First three steps the same…)

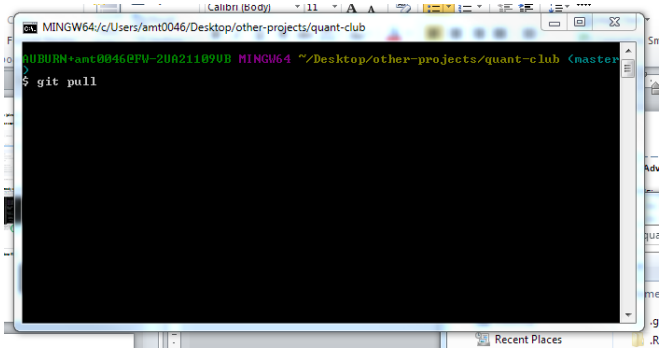
1a. If you are just starting – Clone the repo!

1b. If you've already cloned the repo –pull updates to your machine

- Open the folder location on your computer, right click in and select "Git bash here"



- Now you will interact with the terminal to pull, add, commit, and push



2. Change/update files
3. Stage those changes to be committed

```
AUBURN+amt0046@FW-
$ git add .
```

Add all new/changed files

OR

```
AUBURN+amt0046@FW-2UA21109VB MINGW
$ git add miscellaneous-stuff
```

Add a whole folder

OR

```
AUBURN+amt0046@FW-2UA21109VB MINGW64 ~/Desktop/other-proj
$ git add miscellaneous-stuff/a-beta-distribution.R
```

Add a specific file

4. Commit those changes

```
AUBURN+amt0046@FW-2UA21109VB MINGW64 ~/Desktop/othe
$ git commit -m "I made a histogram :smile: "
```

5. Push your commits

```
AUBURN+amt0046@FW-2
$ git push
```