gitsetup

Github

To link Rstudio and our version control software (Git), you need to first create a GitHub account at:

https://github.com/

You did this already!

Remember you user name and password - you will need thme

and then configure your computer to understand/interact with git, by downloading git software

https://git-scm.com/download/

Done!

Git and Rstudio

Link Git with R studio

To do this: follow these instructions - get as far as creating a new project - for now leave the project empty; but make sure you create with a git repository

- start R studio and create a new project in a directory where you will work on 'stuff' for today

Now follow these instructions to link Git with R studio

https://www.r-bloggers.com/rstudio-and-github/

When you get to the section on opening a shell, Rstudio may download additional software

If you are using a newer version of MacOS, you may also need to install X-quartz so that R can open graphics windows on the fly

https://www.xquartz.org/

You will need to re-start R

Git - how to use version control with your project

R studio links nicely with git; you can use git to track changes both locally and in the remote repository

Lets start by assuming you've set up the project as described above

Tracking things locally with git - example

- download and open the following files and save to your project directory
- **sun.simpleexample.R**
- **sun.txt**
- run the R file to make sure everything works
- go to "Git" tab - see your file - they are ready to be 'staged'
- Check on the file you want to stage

- Commit
  - This will create your master branch
- Make a change to ** sun.simpleexample.R **
- Use diff to see what changed
  - you must save the file first
- Commit changes
- Use history to see how things have changed
- Make some more changes to ** sun.simpleexample.R **

Saving to your global github (for sharing)

- Make sure you have committed all recent changes
- get the http **address** from your github repo
- Open a shell in Git tab in Rstudio

git remote add origin **address**

git push -u origin master

- Notice how now under the Git tab you have colored arrows, you can now use these to push and pull from github to your local repo
- now make a local change and use these to update your github repo
- look at your repo online to see what you've done
- notice how now in git commit it will also track your branch in comparison with remote repo
- if you want other people to be able to make changes to your repo, add them as collaborators on the github website for that repo, under settings (see option on left side)

what if you muck things up

before you commit

- *git revert* this is in the git dropdown menu tools; or you can run on the command line

after you commit

The following steps allow you to get a previous version of your file back and replace the existing one (be careful you will use your work), in the shell/command line

- *git log* this shows you the commit #s of your versions
- *git log –reverse* (same as above but often easier if you have lots of commits)
- write down the version number of the file you want to 'bring back'
- *git reset #commit* changes will be 'staged' but not committed
- *git reset #commit # –hard* same as above but automatically committed;use if you really mucked things up

Using existing (other people) repositories

Find the repository on Github and get its link: Clone with Https:

The link for my repository is https://github.com/naomitague/HebrewU.git

Start R studio again and create a new project

- chose the option ‘create from Git repository’
- use link you found on git site
- make sure you put this in a new directory so files don’t conflict with other projects
- try this with a partners repo