

Setting up GitHub and RStudio

Evan Ray, adapted from Nicholas Horton

Overview of Git and GitHub

Git is a system that lets you:

- Store versions of your files at key points that you want to be able to get back to
- Revert to previous versions of your files if need be
- Collaborate with other people making changes to the same files you are working on
- Share your work with the world

Your code (or other work) is kept in a *repository*. There are typically at least two copies of this repository:

1. One on a server on the internet; in this class we will keep a copy of our code on GitHub
2. One on the computer where you are editing the files; in this class I will suggest that you keep these files on RStudio Server.

If other people are also working on these files, they may have their own copies of the repository as well. In particular, I will be making my own copy of all assignment repositories with work you submit in this class.

Purpose of this Activity

Today's activity will allow us to move forward in terms of the use of GitHub in the class. We will do three things. The first of these steps will have to be done only once, today. The others will be done every time you start a new assignment or lab, or make changes to files in an assignment or lab.

1. Create a GitHub user account and set up RStudio Server to work with that account (today only!)
2. Clone a repository from GitHub (every time you start a new lab or homework assignment)
3. Make changes to files in that repository (in this case, we will create a new file), add and commit those files to a local repository, and push our changes to GitHub. (every time you make changes you want to save)

Note that the procedures we outline today will work if you are the only person working on a project. The workflow for collaborating on projects and sharing code via Git is more involved; we will get there later in the semester.

Step 1: Create a GitHub User Account and set it up on RStudio Server (today only! one-time only task)

1. A) Create a GitHub User Account

- 0) If you already have a GitHub account, please just **write your GitHub user name on the board** and go to step 1. B). Otherwise, read on...
- 1) Go to <https://github.com/>
- 2) Click "Sign Up" in the top right corner and complete the registration process.
- 3) Write your GitHub user name on the board.

1. B) Configure github on the rstudio.mtholyoke.edu server

- 1) Go to <https://rstudio.mtholyoke.edu> and sign in with your Mount Holyoke user name (it's your email **without** the @mtholyoke.edu part) and password

2) Select “Shell” from the “Tools” menu

3) Run the commands (replacing XX with your GitHub username and your email address)

```
git config --global user.name 'XX'
```

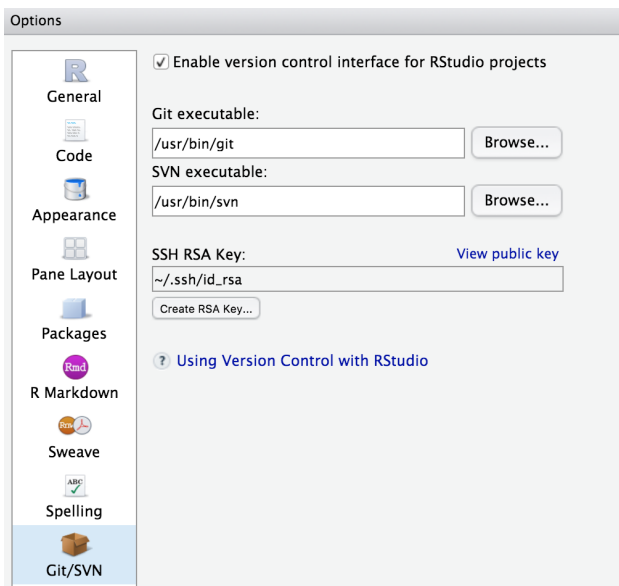
```
git config --global user.email 'XX'
```

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

4) Close the shell

1. C) Generate ssh (secure shell) keys

1) Generate ssh keys on the RStudio server, select Tools; Global Options; Git/SVN; SSH



2) Click on the “Create RSA key” link

3) I would recommend *not* using a passphrase (otherwise you will be regularly prompted for your password)

4) Click “Create”

5) Close the “Create RSA key” box.

1. D) Add public key to GitHub (One time only)

1) In RStudio, click on the “View public key” link (under “Tools”, “Global Options”, “Git/SVN”). You may still be here from the end of Step 2.

2) Copy the public key.

3) Open up github.com (with you logged in)

4) Click on the icon on the top right and select “Settings”

5) Click on “SSH and GPG keys”

6) Click on “New SSH key”

7) Title this ‘MHC R server’

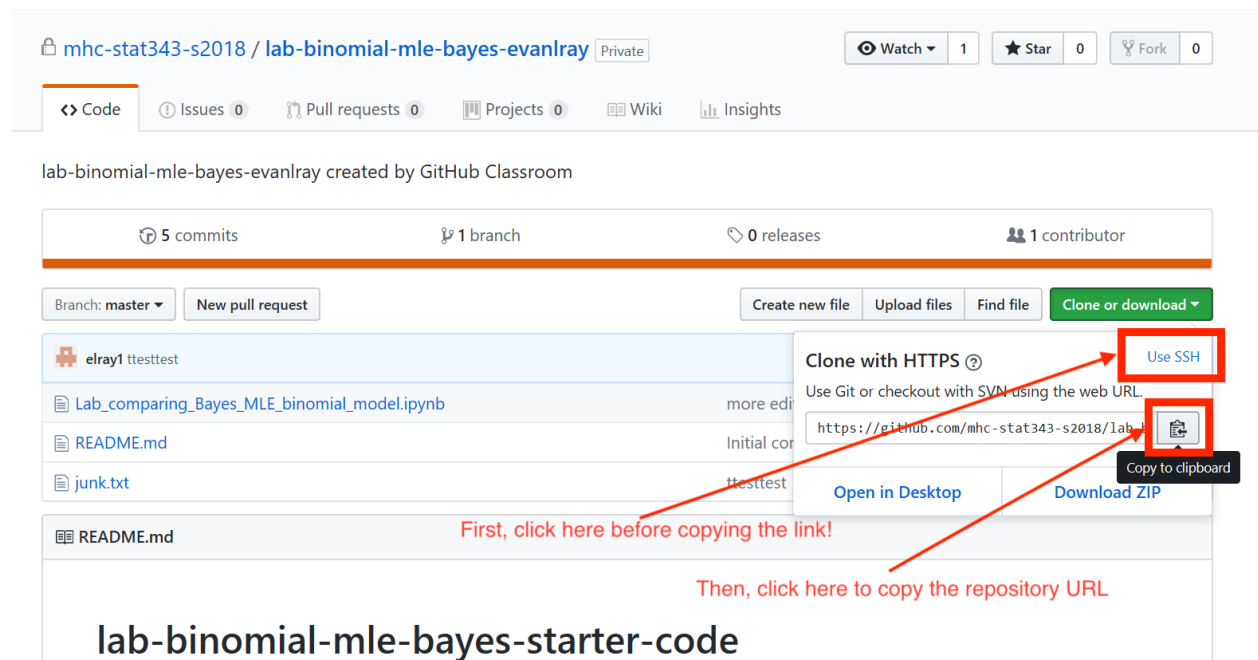
8) Paste in the public key

- 9) Github will prompt you for your password (you don't want people adding in keys without your knowledge!)
- 10) Close the window.

Step 2: Clone a repository from GitHub (every time you start a new lab or homework assignment)

Our goal is to **clone** a repository that already exists on GitHub; after we're done, the files that are there will be available for you to edit in RStudio Server.

- 1) Open the repo I've created on GitHub (<https://github.com/mhc-stat340-f2018/github-username>)
- 2) Click on the Clone or download (green) button
- 3) Select "Use SSH" (rather than https)
- 4) Click on the copy to clipboard button to the right of the URL



- 5) Back in RStudio, click on **File, New Project; Version Control; Git**; then paste the repository URL
- 6) Press the Tab key to fill in the project directory name
- 7) Change the "Create project as subdirectory of" value from "~" (home folder) to "~/stat340" by clicking "Browse" then creating a "stat340" folder and clicking "Choose". (I recommend this as a place to put all of your github repos for this class)
- 8) Click on **Create Project**
- 9) The first time you connect to github from the RStudio server, it may prompt you with a forbidding "authenticity can't be established" message. You should type "yes" into the box to trust this host and click "Okay".

Step 3: Add a file to the GitHub repo and push it to GitHub

Our goal now is to make “local” changes to the copy of the repository in RStudio, and then push those changes to GitHub.

- 1) Create a new text file called EMAIL.txt (where EMAIL should be replaced by your mount holyoke email) that includes the following text:

GitHub user name: USERNAME (replace with your github user name) Email: EMAIL (replace with your MHC email address)

- 2) Save that file
- 3) click on the “git” tab on the top right
- 4) check the box to the left of the file “EMAIL.txt”
- 5) Click on “Pull” (down arrow) to ensure that you have the most recent version of the repo.
- 6) Click “Commit”
- 7) Add a commit message in the box on the right that opens up: “Adding NAME’s file” (where NAME should be replaced with your name)
- 8) Click the “Commit” button. (You may need to do a “Pull” again if other changes have taken place.)
- 9) Click the “Push” button.
- 10) Close the Commit window.

Congrats! You’ve added a file. If you refresh the webpage for the repository on GitHub (<https://github.com/mhc-stat340-f2018/github-username>), you should see your new file listed.

Closing Advice:

- 1) Pull often!
- 2) Don’t add pdfs (or other files that are generated by your Rmd) to your repository until you are ready to submit an assignment.
- 3) Get stuck or run into a conflict? Try running `git status` at the shell or (if you are willing to forget about your changes) `git stash`. We will look at dealing with conflicts in more detail later in the semester.
- 4) Really stuck? Please let me know and I’d be glad to assist.
- 5) Really really stuck? Take comfort in the fact that you’re not alone; it’s a known fact that Git is painful to use. See <https://ohshitgit.com/>