

gitflow

MdM

March 2, 2019



At the start of each project - GITHUB

- Go to <https://github.com> and make sure you are logged in.
- Click green *New repository* button. Or, if you are on your own profile page, click on *Repositories*, then click the green *New* button.
 - Repository name: myrepo (or whatever you wish)
 - Public
 - YES Initialize this repository with a README
- Click the big green button *Create repository*
- Copy the HTTPS clone URL to your clipboard via the green *Clone or Download* button. Or copy SSH URL if you chose to set up SSH keys.

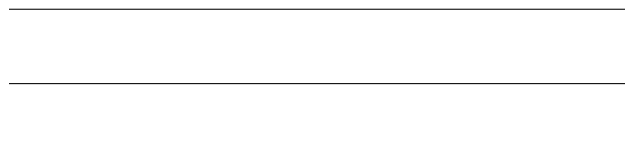


At the start of each project - RSTUDIO

In RStudio, start a new Project:

- File > New Project > Version Control > Git.
- In the *repository URL* paste the URL of your new GitHub repository. It will be something like this <https://github.com/jennybc/myrepo.git>.
 - Suggest you *Open in new session*.
- Click *Create Project* to create a new directory, which will be all of these things:
 - a directory or “folder” on your computer
 - a Git repository, linked to a remote GitHub repository
 - an RStudio Project

In the absence of other constraints, I suggest that all of your R projects have exactly this set-up.



COMMIT

Do this every time you finish a valuable chunk of work, probably many times a day.

- Modify the file
 - IF NEW FILE — From RStudio, Create a file in the project folder
 - IF EXISTING FILE — modify the file, e.g., by adding the line “This is a line from RStudio”. Save your changes.
 - Commit these changes to your local repo. How?
 - Click the *Git* tab in upper right pane
 - Check *Staged* box for any files whose existence or modifications you want to commit.
 - To see more detail on what’s changed in file since the last commit, click on *Diff* for a Git pop-up
 - If you’re not already in the Git pop-up, click *Commit*
 - Type a message in *Commit message*, such as *Commit from RStudio*.
 - Click *Commit*
-

PUSH to GITHUB

Do this a few times a day, but less often than you commit. You have new work in your local Git repo but the changes are not online yet.

- This will seem counterintuitive, but first let’s stop and pull from GitHub.
 - Why? If you make changes to the repo in browser or from another machine, you will be happy if you pull those changes before attempting to push.
 - Click the blue *Pull* button in the *Git* tab in RStudio. I doubt anything will happen, i.e. you’ll get the message “Already up-to-date.”
 - Click the green *Push* button to send your local changes to GitHub.
-

CONFIRM local change to the GitHub remote

- Go back to the browser. Refresh.
 - You should see the new text in the changed file.
 - If you click on *ommits*," you should see one with the commit message.
-

REPEAT

If making a CHANGE ON GITHUB

- Click on the file to change in the file listing on GitHub.
 - In the upper right corner, click on the **pencil** for *Edit this file*.
 - Add a line to this file, such as “Line added from GitHub.”
 - Edit the commit message in *Commit changes* or accept the default.
 - Click the big green button *Commit changes*
-

PULL FROM GITHUB

- Back in RStudio locally. Inspect your file. It should NOT have the added line. It should be as you left it. Verify that.
 - Click the blue *Pull* button.
 - Look at file again. You should now see the new line there.
-
-
-

DELETE A REPO

- On GitHub, navigate to the main page of the repository.
- Under your repository name, click Settings.
- Under Danger Zone, click Delete this repository.