Git/GitHub Collaboration Step-by-Step Guide
Prepared for the Blaszczak Lab, University of Nevada Reno
Adapted from **Happy Git With R** by Jenny Bryan
Assembled by Heili Lowman

How to: **Establish or join a collaborative project via GitHub.**

If you are the **owner** of the original project repository:

- On your GitHub repository, click on "Settings" > "Collaborators". GitHub will prompt you to re-enter your password.
- Click on "Add people" and search for your collaborator by their GitHub username. Then, click on the green button to add them as a collaborator on the project.
- *Note, invitations to join a GitHub repository are sent to the e-mail address the user first inputted when they set up their GitHub account. They must accept your invitation within 7 days or the invitation expires and you have to re-send it.*
- Optional: Establish a new branch that you'd like all your collaborators to send their commits to (see how to create a branch in the next section).

If you are invited **to join** a project repository:

- Navigate to the invitation e-mail and click on "View invitation."
- To download the project to your personal computer, since you are not the owner of the repository, it is best practice to first **fork** and then **clone** the repository.
- Navigate to the project on your collaborator's GitHub and click on the green "Code" button. Copy the hyperlink that appears in the HTTPS section of the dropdown to your clipboard.
- Open RStudio.
- Install the package "usethis" in your R console. This is the package we will be using to fork and clone the new project to your computer.
- Run the following code in your R console, using the "repo_spec" field to populate the URL you previously copied from the project on GitHub:

```
usethis::create_from_github(
     repo_spec="https://github.com/OWNER/REPO",
     destdir = "~/path/to/where/you/want/the/repo/",
     fork = TRUE)
```

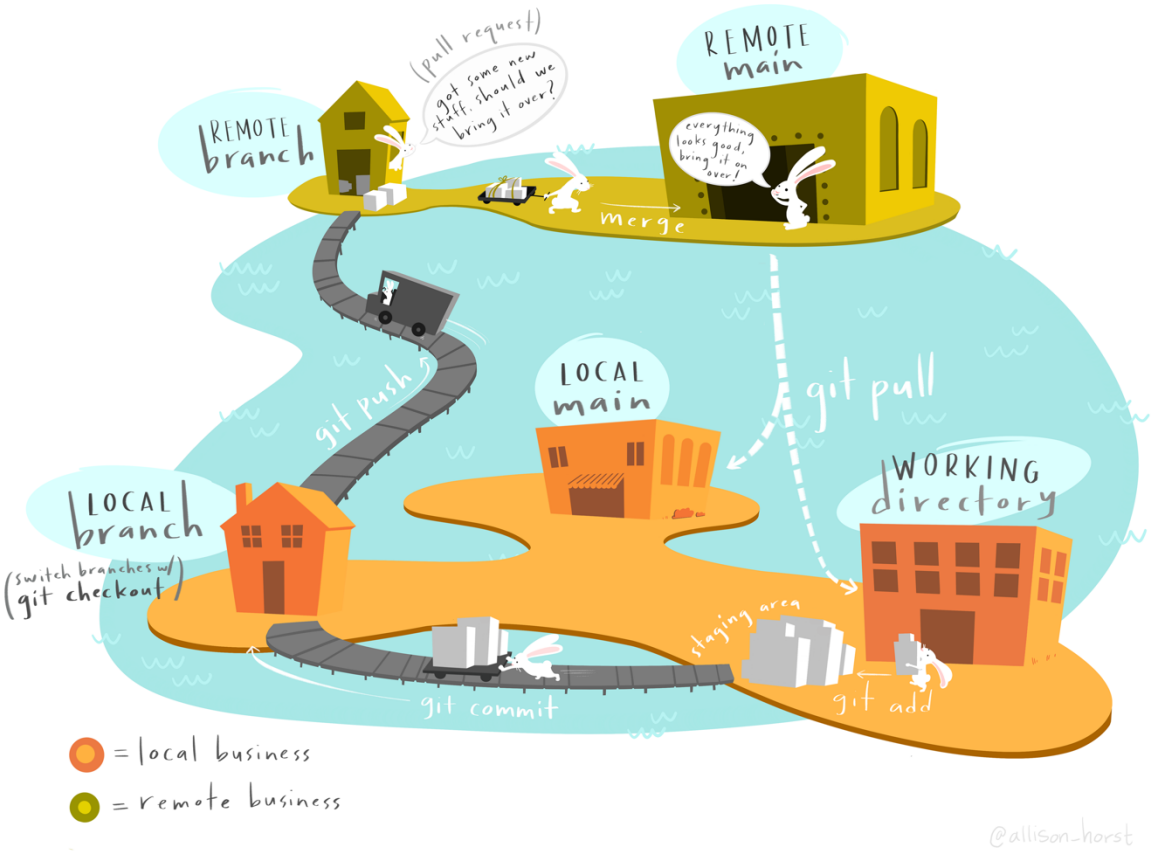
- The project should open in a new RStudio instance. Check and be sure that the files that appear in the original repository that you were invited to on GitHub also now exist on your computer.
- Woohoo! You're in!!

How to: **Create a branch to add files/changes via GitHub and RStudio.**

- Make sure your and the original repositories are **synced** up. Navigate to your forked repository on GitHub and click the "Sync fork" button.
- Navigate to the project in RStudio.
- In the upper right-hand corner, click on the button with a collection of small, purple shapes. In the new window that opens, create a descriptive branch name ("heili_figures"), set the remote to **origin**, and leave the "Sync branch with remote" check-box checked. Click "Create."
- You are now operating in your new forked repository, in a new branch ☺
- You made add/edit files in this project via RStudio as you normally would (File > New File > R Script).
- **To save files locally:** Edit your scripts, and save your changes as you normally would (File > Save).
- **To save changes locally:** Once you click "Save", any files that have been updated will appear in your "Git" pane in RStudio. Check the box next to every file you would like to store on GitHub, and click "Commit." In the pop-up window, add a descriptive commit message, and click "Commit." You've now tracked your changes on your computer using the Git software.
- **To add changes to remote repositories on GitHub:** Once you have committed your changes, click the green up arrow in your "Git" pane to send your files/changes to your GitHub repository for the project. Navigate to your GitHub account on your browser, and you should see the pull request banner at the top of the screen. Click on "Compare & pull request". On the new screen, you should see the two branches you're proposing to merge at the top listed as "base" and "head" repositories and their corresponding branches.
    - **To edit <u>your</u> forked repository:** Set the base repository to be your repository on the main branch. Click on "Create pull request" followed by "Merge pull request" and finally "Confirm merge" to merge your new branch's changes to your main branch.
    - **To edit <u>your collaborator's</u> original repository:** Set the base repository to be the original repository on the main branch. Click on "Create pull request" followed by "Merge pull request" and finally "Confirm merge" to merge your new branch's changes to your main branch.
- Overall, when working collaboratively, you want to follow this workflow:
    - From the main branch, **Sync** + **Pull** changes from GitHub at the start of your day.
    - On a new branch, **Save** + **Stage** + **Commit** frequently (every time you make a significant change to a chunk of code). Don't forget descriptive commit comments!
    - **Push** + **Merge** changes to GitHub at the end of your day.
- Celebrate your git wizardry!

How to: **Fix a merge error message via RStudio.**

- When you and a collaborator try to edit similar parts of code, you may encounter what is known as a **merge conflict**. The system may not be able to choose between versions, since it cannot prioritize one collaborator's edits over another.
- In the R script, the issue will be identified by a leading set of <<<<<<<, the two options will be delineated by =======, and the issue will be closed by >>>>>>>. Your edits will appear above the =======, and the other person's edits will appear below.
- To make your final decision, you may delete both changes, select/keep one change, or keep both. Once you've made the necessary edits, delete the conflict markers (symbols described above).
- To add these changes, first save the script.
- To commit these changes to the project, be sure the checkbox next to the script in the "Git" pane has a check symbol in it (it should *not* simply be a filled-in square). Click the commit button, add a descriptive message, and commit/push/merge your changes as usual.
- Navigate to your repository on GitHub, and check to see that your edits are there.
- Take a deep breath, and pat yourself on the back! You just fixed a merge conflict!



Artwork by Allison Horst (allisonhorst.com/git-github)