

Git/GitHub Step-by-Step Guide
Prepared for the Southern California Coastal Water Research Project
By Heili Lowman



How to: **Start a project and add files/changes via GitHub and RStudio.**

Start a project (GitHub first).

- Open GitHub in your browser and log in to your personal account.
- On your newsfeed page, find the “repositories” pane on the left-hand side and click on the green “New” button.
- Choose a helpful repository name, like “la_river_restore”, and add a helpful description for the project’s intended contents.
- Unless your data/work is proprietary, keep the repository “Public”. In either case, be sure to “Add a README file” to accompany your other project files.
- Click “Create Repository.”
- Check out your new repository page!

Add files/changes via GitHub.

- You may add files directly to GitHub by clicking on “Add file” > “Upload files” and navigating to them on your computer.
- You may edit the README file by clicking on the pencil icon in the top right corner. This will open the R Markdown document, where you may add your text/code, and then click “Commit changes.”

Add files/changes via RStudio.

- On your GitHub repository page, click on the green “Code” button and copy the hyperlink that appears in the HTTPS section of the dropdown to your clipboard.
- Open RStudio.
- Navigate through File > New Project > Version Control > Git and paste the URL into the “Repository URL:” section. Press the “Tab” key, and the name of the project directory should autopopulate (best to keep it identical to the name on GitHub). Double check your project is being saved to the correct directory on your computer (“Browse”), and then click “Create Project.” Your project should automatically open in a new window.
- Check the folder on your computer where this project was saved to be certain all of the files are there.
- You may add files to this project by dragging and dropping them into the project folder in your finder. They should then appear in your “Files” pane in RStudio.
- You may 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 (ignore .gitignore), and click “Commit.” In the pop-up window, add a descriptive commit message, and click “Commit.”
- **To save files & changes remotely:** Once you have committed your changes, click the green up arrow in your “Git” pane to send your files/changes to GitHub. Navigate to your GitHub account on your browser, and check to see that they are there.
- Celebrate your wizardry!



How to: **Collaborate on a project via GitHub.**

- On your GitHub repository page, click on “Settings” > “Manage access” (it may prompt you to type in your password once more).
- Click on “Invite a collaborator” and search for your collaborator by their GitHub username. Then, click on the green button to add them as a collaborator.
- *Note, they must accept your invitation within 7 days or the invitation expires.*
- If you are invited to collaborate on a project, navigate to the invitation email and click on “View invitation.”
- To download (or clone) the project to your personal computer, follow the same instructions as above.
- To add edits to the project from your personal computer, follow the same instructions as above.
- Overall, when working collaboratively, you want to keep the following workflow in mind:
 - o **Pull** changes from GitHub at the start of your day.
 - o **Save + Stage + Commit** frequently (every time you make a significant change to a chunk of code).
 - o **Push** changes to GitHub at the end of your day.
- You are now officially a champion of open data science!



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 as usual.
- To push your changes to GitHub, click the green arrow in your "Git" pane.
- Navigate to your project 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!



For more help and information visit: <https://happygitwithr.com/>