Introduction to Collaborative Coding with GitHub

This work by Elizabeth Waterfield, Sara Lil Middleton and Sarah von Grebmer zu Wolfsthurn is licensed under a CC-BY 4.0 Creative Commons Attribution 4.0 International License.

These practical exercises will build on your existing knowledge of navigating GitHub for collaborative coding by giving you practice with using Git commands in your system's terminal. It will use the same repository that you cloned from the lesson that was originally forked from LMU OSC's Collaborative-RStudio-GitHub repository.

Setting up the terminal

- Every operating system uses a different default terminal.

Windows – typically used through Git Bash
macOS – used through built-in Terminal app
Linux – runs in the system's default terminal

The interface may differ, but Git works the same across all of them. Use it by typing the Git command and then hit 'Enter'.

- Change the **working directory** to your project folder so that the Git commands apply to these files.

Do this by entering "`cd`" followed by the path to your project.

For example: `$ cd /c/Projects/Collaborative-RStudio-GitHub`

(Tip: if the path to your folder has spaces, wrap the path in quotation marks when entering it in the terminal)

- Keyboard shortcuts work differently in some terminals.
  For these activities, follow this:
  To copy, right-click → Copy
  To paste, right-click → Paste

Practical exercise 1

Make and check changes with: `git status`

☐ In the project folder on your local device, open the "Params" folder.

☐ Right-click the "params_tmpl" R file and select "Copy" then right-click again and select "Paste" to make a copy in the local folder.

☐ Right click the copied R file then click "Rename" to give it a new name.

☐ Open the copied R file on your local device and make new parameters by editing the 'sig2', 'species.name', and 'color' elements.

☐ Save the edited file by clicking "File" then "Save"

☐ Open your **Terminal** and set the working directory to the folder on your device that has the cloned repository.

☐ Enter the command "`git status`" to check what changes are there in the local copy that aren't on the remote copy. These changed files will show up in red.

Practical exercise 2

Stage and commit changes with: `git add` and `git commit -m`

You have made new changes!

☐ In **the terminal**, stage all the changes by entering the command "`git add .`"

☐ Enter the command "`git status`" to see the files are now green meaning they are staged to commit.

☐ Enter the command "`git commit -m`" followed by a commit message in quotation marks.

For example: `$ git commit -m "Added a new parameter"`

These changes are now committed.

Practical exercise 3

Push and pull changes with: `git push` and `git pull`

☐ To push your changes to your forked remote repository, enter the command "`git push`"

On **GitHub**, refresh the page and your changes should appear.

To pull the changes made to the remote repository to your local copy, enter the command "`git pull`"

Extra: git clone

Clone a repo with: `git clone` + "repository URL"

You can use a simple Git command to clone a remote repository to your local device. Here's how:

On GitHub, copy the repository URL
(Hint: Find the URL by clicking the <> Code button)

In the terminal, set the working directory to the destination on your device where you would like this clone to be.

Type the command "`git clone`" and paste the URL for the repository. Here's an example:

```
$ git clone https://github.com/lmu-osc/Collaborative-RStudio-GitHub.git
```

There should now be a copy of the repository in the folder on your local device.

OSC LMU Open Science Center FORRT