Exercise 1

The exercises are not mandatory and need not be handed in, but we still recommend them strongly. This exercise will be discussed in the tutorial online on October 28th, 2022.

1 Introduction to Git and GitHub

In this exercise we want to introduce basic concepts of git and GitHub. If you still need to install git, you can download the latest release from https://git-scm.com. If you require a quick refresher on the most important git commands, checkout this git cheat sheet. We are not able to discuss all important git concepts in this exercise, but you can find many in-depth resources online. Note that many graphical interfaces exist for git, and many IDEs have integrated tools that can simplify the git workflow.

1.1 Git Fundamentals

1. Globally set your user name and email address. This information will be tied to every commit you make.

2. Create a new directory and initialize a new git repository inside.

3. Create a new text file in the directory. Add the new file to the staging area. Check the current status of the staging area.

4. Commit the file to the repository and submit a sensible commit message. Recheck the status of the working directory and inspect the project history to ensure everything worked as intended.

5. Change the content of the submitted file. Check the current status of the staging area and display the differences to the previous commit. Add the changes to the staging area.

6. Under the assumption that you accidentally added the changes, unstage the staged file again. Afterwards discard the changes you made to the file, and in doing so restore it to what it looked like in the first commit.

7. Change the content of the file again, and add and commit the changes. Undo the local commit you just made, without discarding the uncommitted changes. Finally, add and commit the changes again.

8. Create a second text file in a subdirectory called `logs`. Create a `.gitignore` file to ensure that git ignores all files in the `logs` subdirectory. Check the status of the repository and add and commit the `.gitignore` file.

1.2 GitHub

If you have not done so already, create a new GitHub account at https://github.com/. To connect to GitHub via git and the command line, you will need to setup your preferred authentication method. HTTPS is generally recommended when starting out, as the setup is a bit more straightforward.

Unai Fischer Abaigar
Unai.FischerAbaigar@stat.uni-muenchen.de

1. Create a new empty repository in GitHub. GitHub will prompt you with different options to setup a repository. Since we have already created a local repository, follow the instructions under "...or push an existing repository from the command line" and push the previously created repository to GitHub. Take care to use the URL that corresponds to the authentication method you chose before. Check that the repository has been updated in GitHub.

2. Navigate to the text file in GitHub, change a line in the file and directly commit your edit. Pull your changes from GitHub to update your local repository.

3. Create a change locally and commit it to the repository. Push the local commit to GitHub.

4. We will now create a merge conflict on purpose and try to fix it. First change the first line of your text file in GitHub and commit the edit. Now change the same line in a different way in your local repository. Add and commit your local change. Now attempt to pull from GitHub. Resolve the merge conflict in some way, and push the finalized changes to the remote repository. If you need some guidance check out the GitHub Docs.

2 Getting Started with R Markdown

Have a look at the `intro-to-rmarkdown.Rmd` file in moodle. You will require the `rmarkdown` package, but RStudio should automatically prompt you to install it. Work through the small tasks in the notebook and play around with the different functionalities R Markdown provides.