MY472 Data for Data Scientists

Seminar - Week 1

Friedrich Geiecke

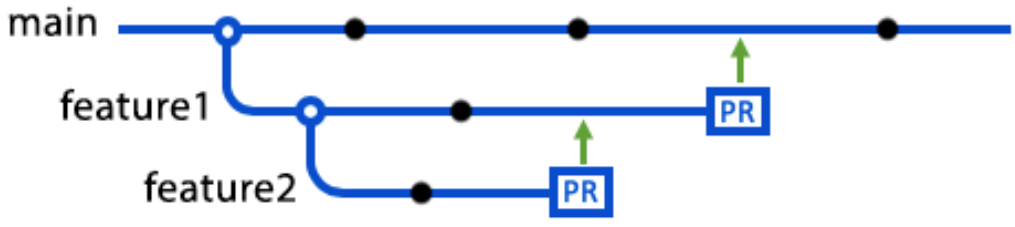
London School of Economics and Political Science

Review of key commands from the lecture

- `git clone ...`: Download online repository to local computer
- `git status`: See status of files in repository
- `git add .`: Stage all changes made (alternatively add distinct file names to be staged)
- `git commit -m "some message describing edits"`: Commit (i.e. record) staged changes
- `git push`: Upload local changes to remote repository
- `git pull`: If files changed online, update local repository first

Some further concepts

- Fork: Own copy of a repository (pushed changes to this copy do not affect the original remote repository - different from `git clone`)
- Branch: A parallel version of the code originating from a duplication at one point
- Merge: Combine branches
- Pull request: GitHub based request to merge a branch or a fork into other code (discussed in exercises)



This seminar: Branches and merges

- In the following, we will briefly discuss branches and merges
- We will also see an example of a simple merge conflict

Branching and merging - a simple example

- For this, first got the the repository from the lecture with `cd /path/to/firstrepository` in Git Bash or Terminal
- Check with `git status` whether all previous changes are committed and if not, stage and commit them
- Check on which branch you are with `git branch`
- It should be "main" (or previously named "master")
- Create a new branch called "development" with `git checkout -b development` (`git checkout` switches between branches, the `-b` flag here creates the new branch at the same time)
- Add a new line below the previous code in the `.R`-file, e.g. `print("this could be a new feature.")`
- Stage and commit these changes

Branching and merging - a simple example

- To see that the changes in the `.R`-file only exist on the new branch, switch back to the main branch with `git checkout main` and have a look at the file. The changes are gone. Switching back to development makes them reappear
- Now we could merge the new development branch into main. For this simply run `git merge development` while being on the main branch
- With RStudio or VS Code, check that now also the main branch version contains the new feature
- This merge was very easy, but there could be merge conflicts (next slide)

Branching and merging - resolving a merge conflict

- Again go to the development branch and edit something in the last print statement. Stage and commit the changes on development
- Move to the main branch, but now also edit something in the last print statement there. Stage and commit the changes on main. The same line was now changed on both branches and hence there will be a conflict
- Running `git merge development` while being on the main branch results in a warning of the merge conflict. Some text was added to the code file to show where the conflict occured
- VS Code has very nice functionalities to resolve the conflict with a few clicks, but we could also just edit the text manually
- Note: Merge conflicts can also happen when pulling a remote repository which contains changes relative to the local version. They are resolved in the same way

Merging via GitHub Pull Requests

- Lastly, let us look at Pull Requests which allow to merge forks or branches directly via GitHub
- Go to the development branch via the command line
- Record your new branch now also in the remote GitHub repo with `git push --set-upstream origin development` (this only has to be done once)
- Then make some changes to the file on the development branch, commit, and push them
- Afterwards go to the repository website on GitHub which now shows the option to merge the changed branch into main via a Pull Request

Git/GitHub - further options and study

- People often use a combination of Git via the command line and the user interface of the GitHub website
- Yet, there is also a graphical user interface from GitHub to replace the command line (GitHub Desktop), or Git can be used directly through RStudio as an R-specific alternative to using the more general command line
- For detailed online manuals and books that discuss Git, see e.g. `https://git-scm.com/book/en/v2`
- To review GitHub, see e.g. `https://docs.github.com/en`
- For GitHub Desktop, see `https://docs.github.com/en/desktop`