Version control with git and github

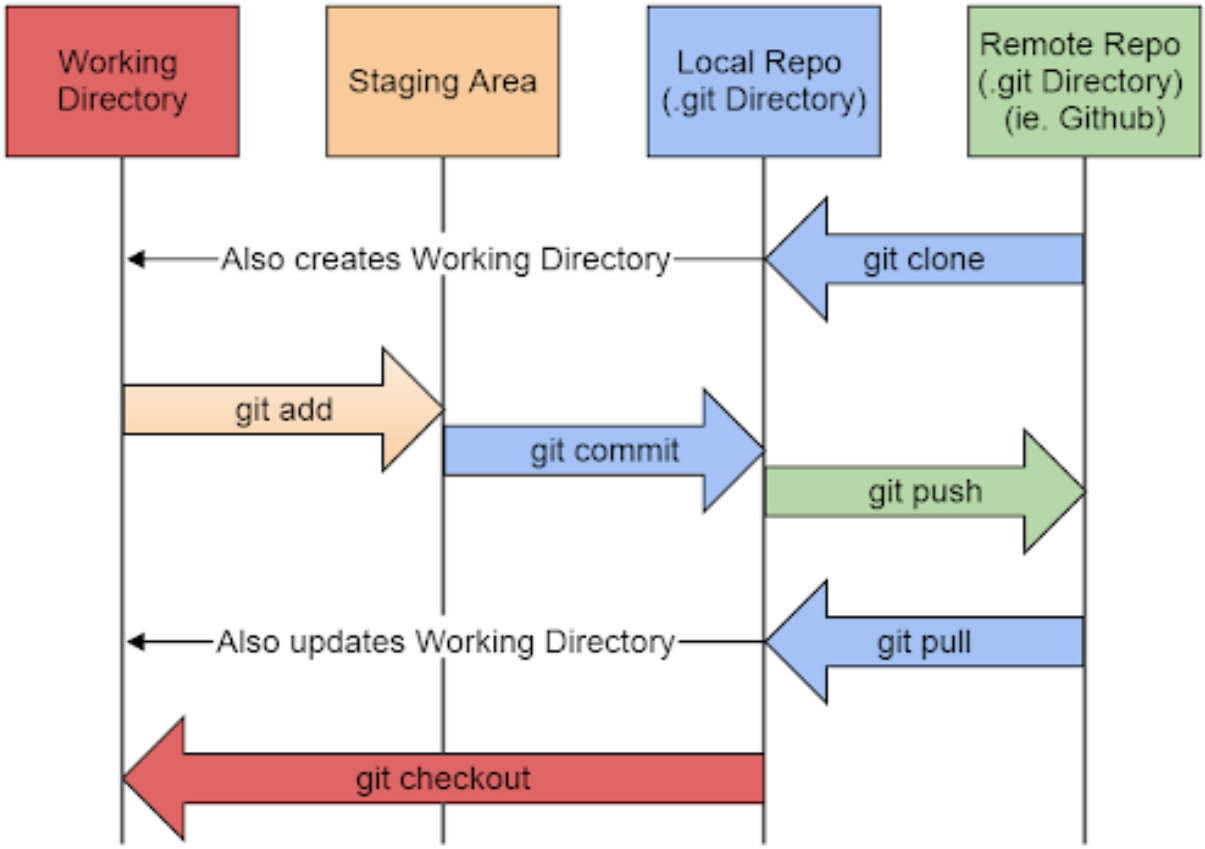
What is git and github?

Git

Git allows you to save the "history" of your code. You can save (**commit**) the momentary version of your code, then change it and save the new version. If you later realize, that the first version was better or that somethng in the second version does not work anymore, you can go back to the first version.

Github

Github is an internet platform where you can upload (**push**) your git versions. Then others can download (**pull**) your version, add something to it and push their new version again.



Usefull links

Some helpfull links, that i also cite a lot here: - https://support.rstudio.com/hc/en-us/articles/200532077-Version-Control-with-Git-and-SVN

Install

Prior to using RStudio's version control features you will need to ensure that you have Git and/or Subversion installed on your system. The following describes how to do this for various platforms. Git

Prior to using Git with RStudio you should install it using the appropriate method for your platform:

- Windows & OS X: http://git-scm.com/downloads
- Debian/Ubuntu: sudo apt-get install git-core
- Fedora/RedHat: sudo yum install git-core

Task 1 install git on the computer you are usually working with.

Create a Github account and link it to your git

Task 2 If you haven't already, create a github account for yourself at https://github.com/

Setting your commit email address in Git

You can use the git config command to change the email address you associate with your Git commits. The new email address you set will be visible in any future commits you push to GitHub from the command line. Any commits you made prior to changing your commit email address are still associated with your previous email address. Setting your email address for every repository on your computer

- Open Terminal.
- Set an email address in Git. You can use your GitHub-provided no-reply email address or any email address.

```
(bash) git config --global user.email "email@example.com"
```

- Confirm that you have set the email address correctly in Git:

```
(bash) git config --global user.email
```

- Add the email address to your account on GitHub, so that your commits are attributed to you and appear in your contributions graph. (For more information, see https://docs.github.com/en/github/setting-up-and-managing-your-github-user-account/managing-email-preferences/adding-an-email-address-to-your-github-account)

Task 3 Set up your email address in git (on your computer) and on the github platform

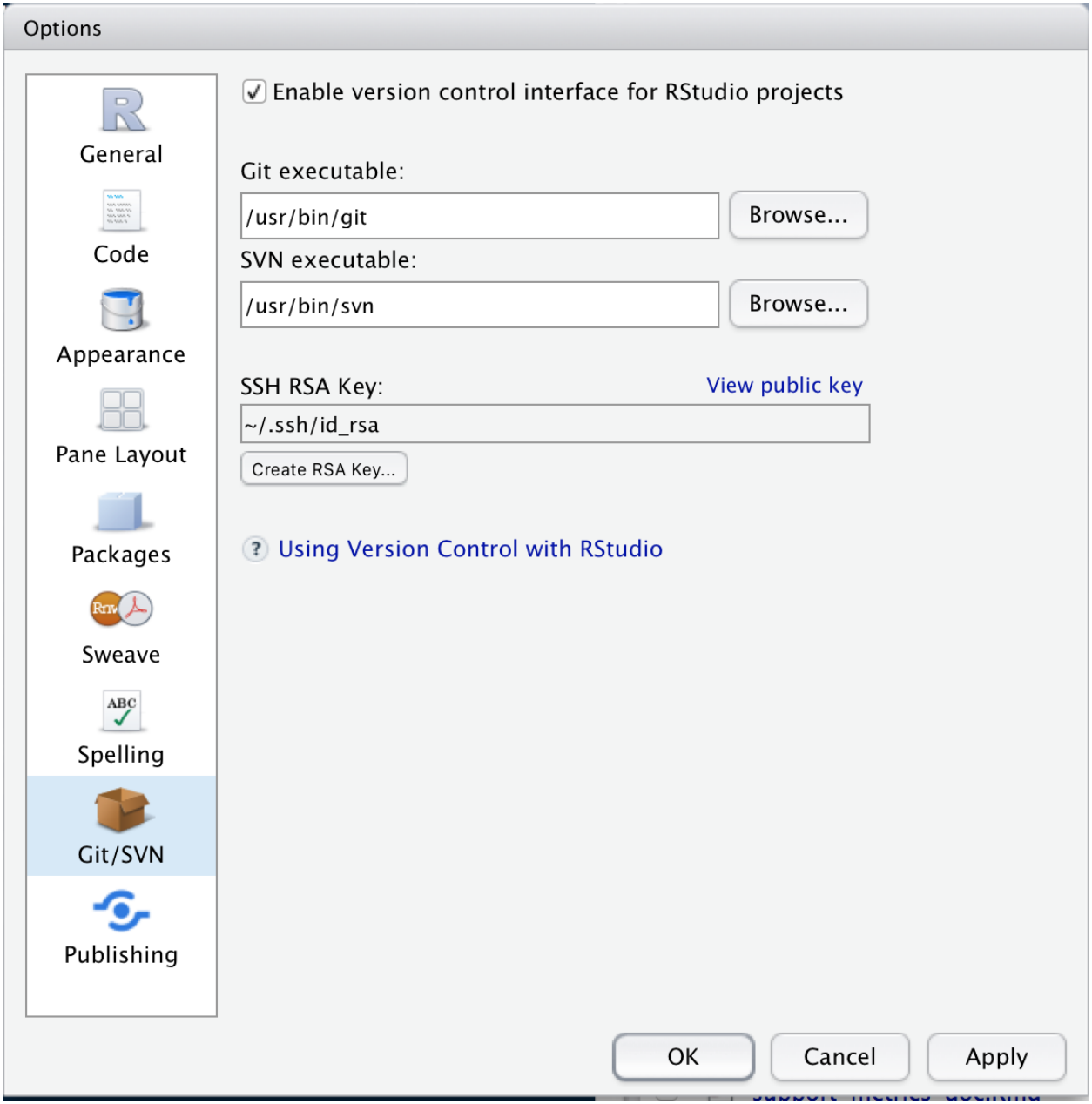
Add version control to your R project

First we now want to add version control to the code of your project (e.g. your thesis)

Enable Git in RStudio

Once you've installed your preferred Version Control system, you'll need to activate it on your system by following these steps:

- Go to Global Options (from the Tools menu)
- Click Git/SVN
- Click Enable version control interface for RStudio projects
- If necessary, enter the path for your Git or SVN executable where provided. You can also create or add your RSA key for SSH if necessary.



Task 4 Enable git in your RStudio

Using RStudio projects

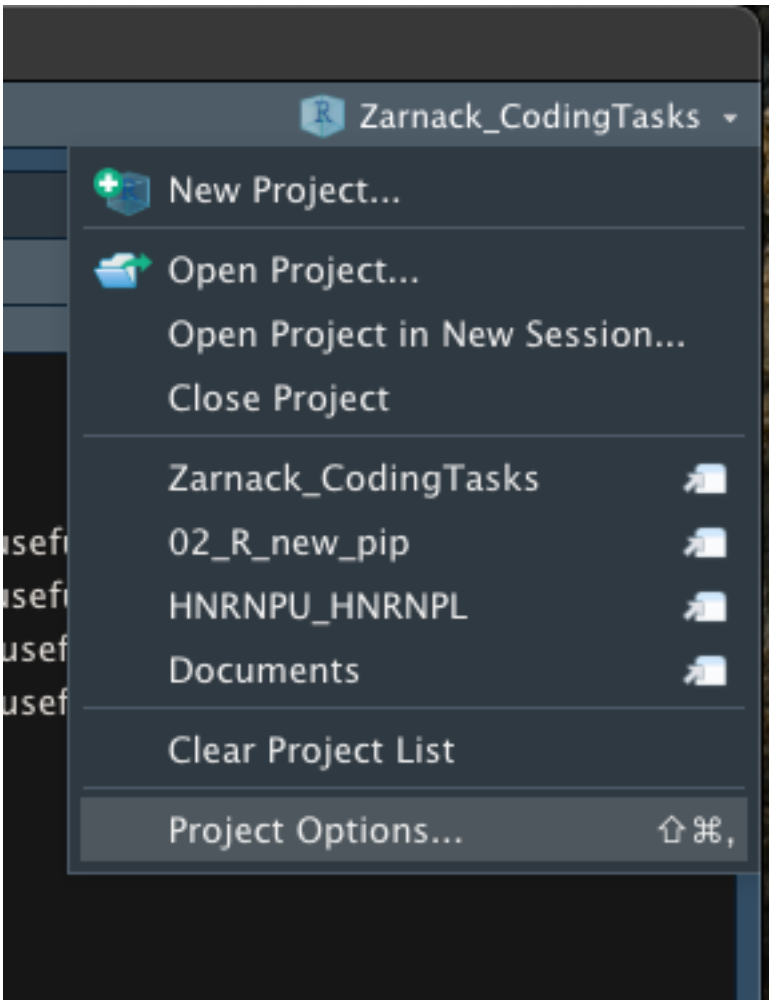
RStudio's version control features are tied to the use of Projects (which are a way of dividing work into multiple contexts, each with their own working directory).The steps required to use version control with a project vary depending on whether the project is new or existing as well as whether it is already under version control.

If you code is not in a RStudio project yet

You can setup an RStudio project on File > new project (also see herehttps://support.rstudio.com/hc/en-us/articles/200526207) and choose git in the version control options. The path of the RStudio project should be the path to your folder with all your R code.

If your code already is inside an RStudio project

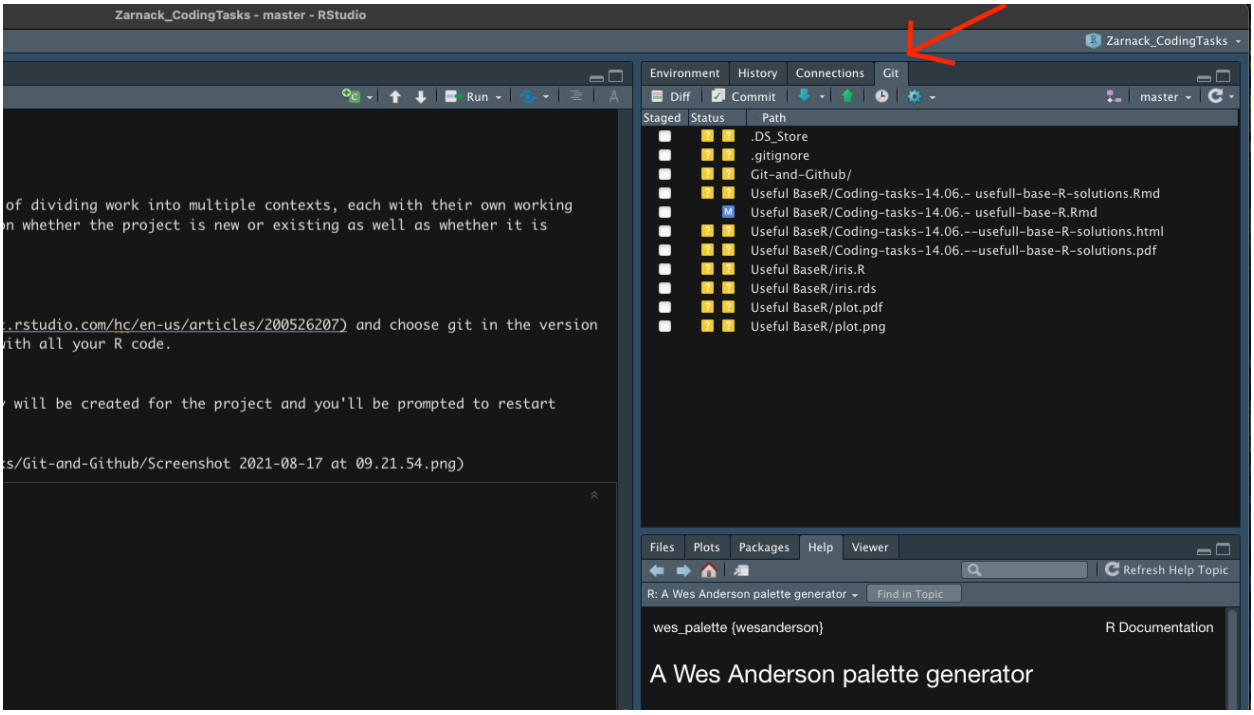
Go to project options and change the version control option to git. A Git repository will be created for the project and you'll be prompted to restart RStudio to enable version control features for the project.



Task 5 Set up an RStudio project with git version control that includes all code of your project.

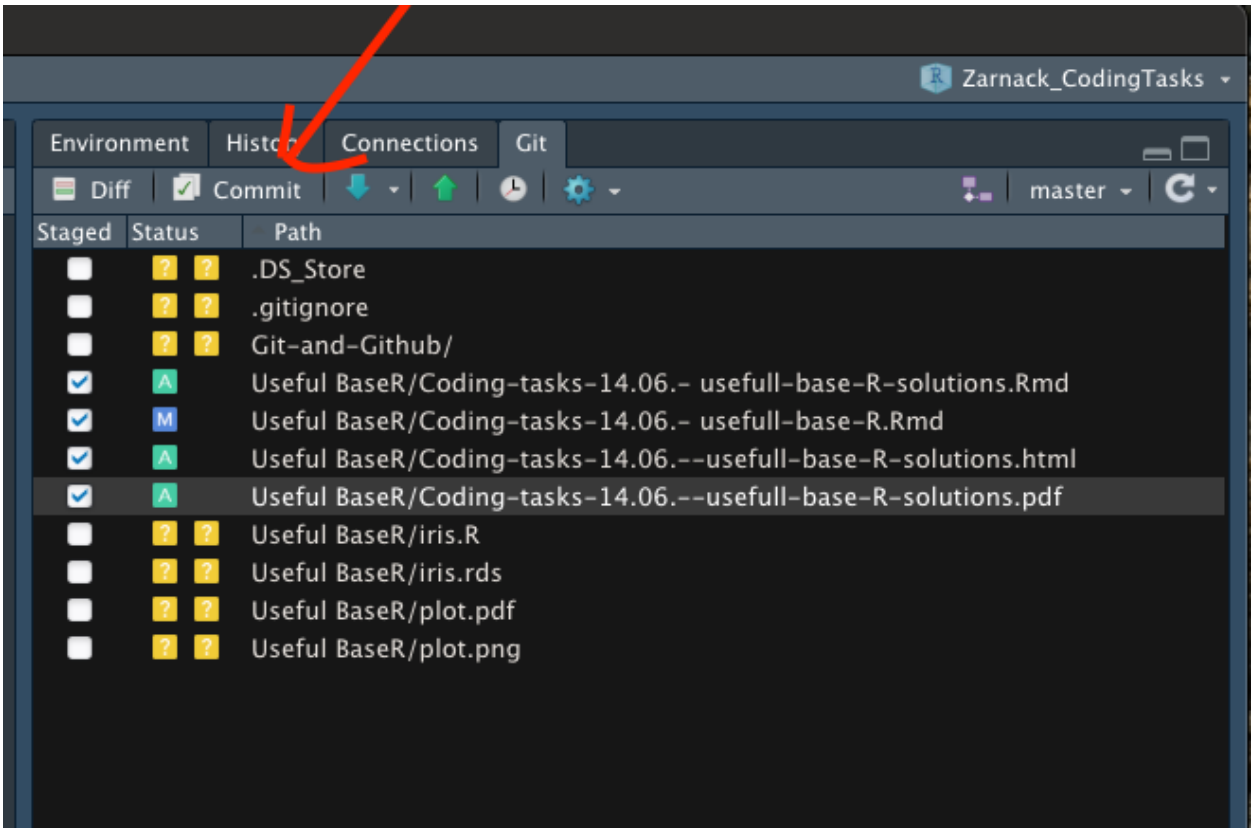
Backup your code

- You should now have a git window in your RStudio.



- It shows all files of your project not in git with a yellow sign. - And files of your project that are in git, but you changed them since the version was last canged in blue (modified) - Files with up to date version control are not shown :)

- To commit your code version to git now you can tick the code and click commit



- You will see all changes in your code since the last version marked in green and red. You have to add a

commit message. E.g. Version 15.08.21 changed plot color and then click commit. Done! You have saved your code version.

- It is helpful to commit your new code regularly (every evening / every Friday) to stay up to date.

Task 6 Commit all your code

Revisit older versions

- Click on the clock symbol to see the history of your project

/Users/melinaklostermann/Documents/non-project-R/Coding_tasks/Zarnack_CodingTasks/Git-and-Github/Screenshot 2021-08-17 at 09.52.05.png

Github - sharing code with others

Pull a project from github

You can pull the coding tasks project by creating a new r project: `File > new project > from version control > git`

- set the url https://github.com/ZarnackGroup/Zarnack_CodingTasks
- set a local path where to save the project

Push your changes back to the repository

First commit your changes like before. Then click the upward arrow to push the commited changes up to the github repository.

Setting your credentials

If you are asked for a access token you can set it like this: https://docs.github.com/en/github/authenticating-to-github/keeping-your-account-and-data-secure/creating-a-personal-access-token

If you want RStudio to save your credentials you can use

```
library(gitcreds)
gitcreds_get()
gitcreds_set()
```

Task 7 Pull the coding tasks, go into this markdown and write your name here. Commit your changed version of this script and push it back to the github repro.