Reproducible Workflow

Git

Michelle Donzallaz

3. What is `git` and how can I use it?

Slides and Material

You can find the slides and materials here: https://github.com/jstbcs/ReproducibleWorkflowWorkshop.

Materials heavily build on previous workshops by Julia Haaf

What I would like to show you about `git`

- How to use a terminal
- Git
  - What is it good for?
  - What is it?
  - What can it do?
- Set-up for your computer
  - GUI/terminal
  - R Studio & git
  - SSH
  - Set name & email address
- Your first repo
  - Github and GitLab
  - In R Studio
  - gitignore
  - README
- Workflow
  - Add, Commit, Push
  - Diff
  - Merge, Branches, Tagging… (all the cool stuff)
  - What happens if something goes wrong? (And it will.)

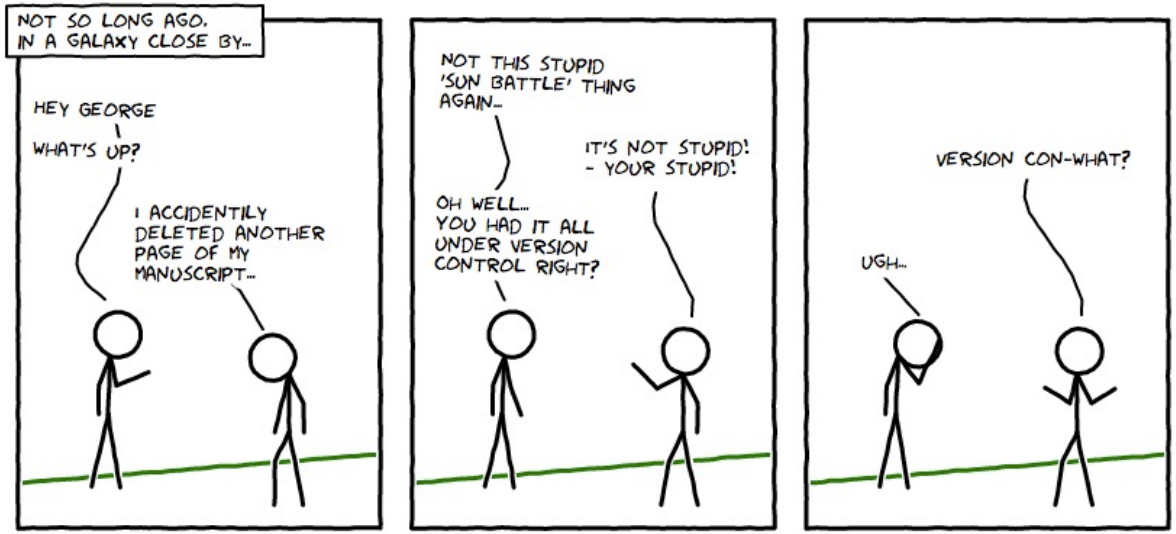
What we have time for

- *How to use a terminal*
- Git
  - **What is it good for?**
  - **What is it?**
  - What can it do?
- Set-up for your computer
  - *GUI/terminal*
  - **R Studio & git**
  - *SSH*
  - Set name & email address
- Your first repo
  - *Github and GitLab*
  - **In R Studio**
  - **gitignore**
  - **README**
- Workflow
  - **Add, Commit, Push**
  - Diff
  - Merge, Branches, Tagging… (all the cool stuff)
  - What happens if something goes wrong? (And it will.)

What is it good for?

Version control:

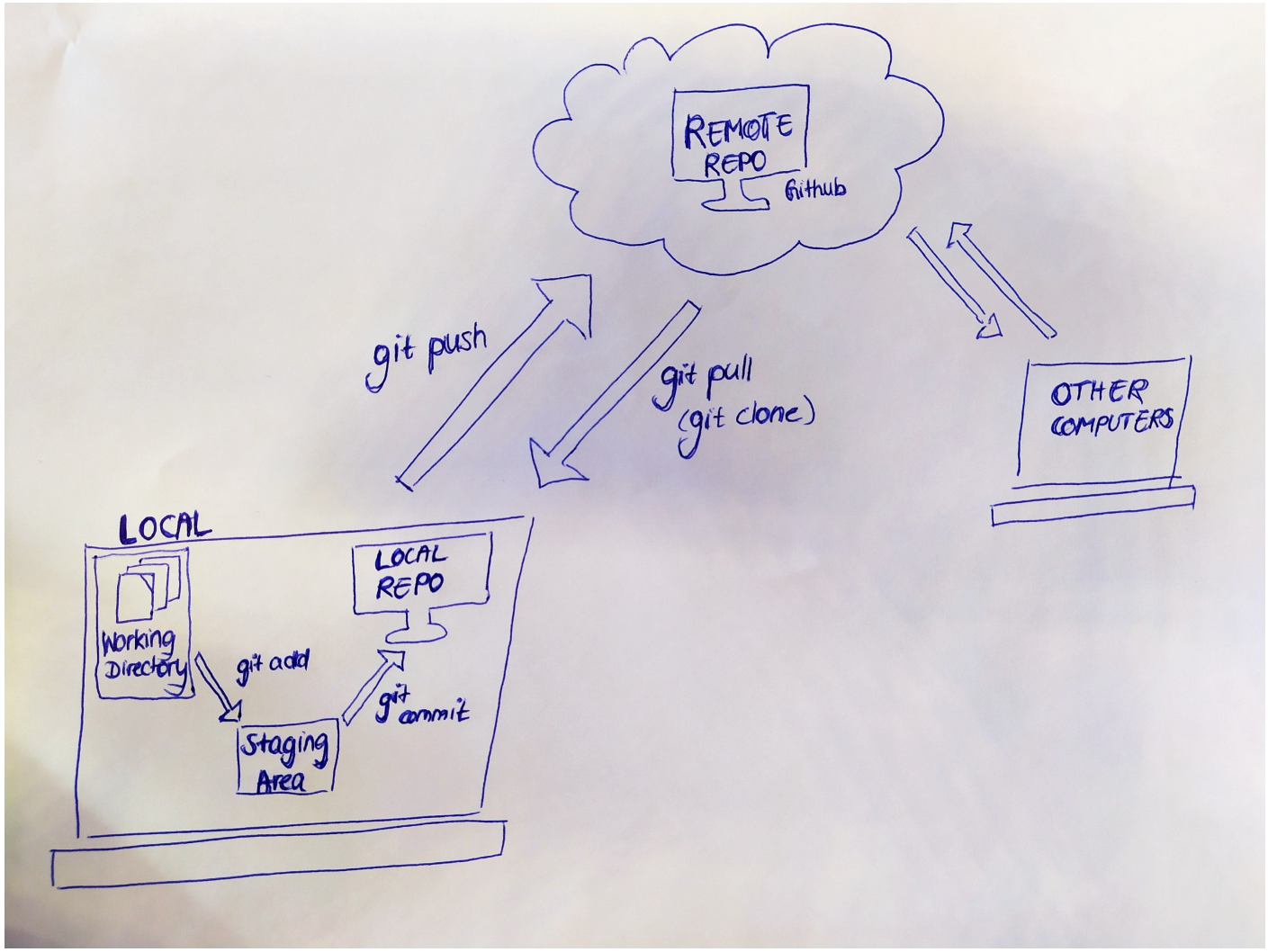
- Version control is a system that records changes to a file or set of files over time so that you can recall specific versions later.
- History of all changes (who, what, when).
- Helps to avoid mistakes (working on the wrong version, deleting, …).
- Merging changes of multiple collaborators in one file.



What is it?

- Git is a essentially a program on your computer, you have to communicate with it.
- GitHub (or GitLab) is a Git repository hosting service.

What is it?


REMOTE REPO
Github
git push
git pull
(git clone)
OTHER COMPUTERS
LOCAL
LOCAL REPO
Working Directory
git add
git commit
Staging Area

What can it do?

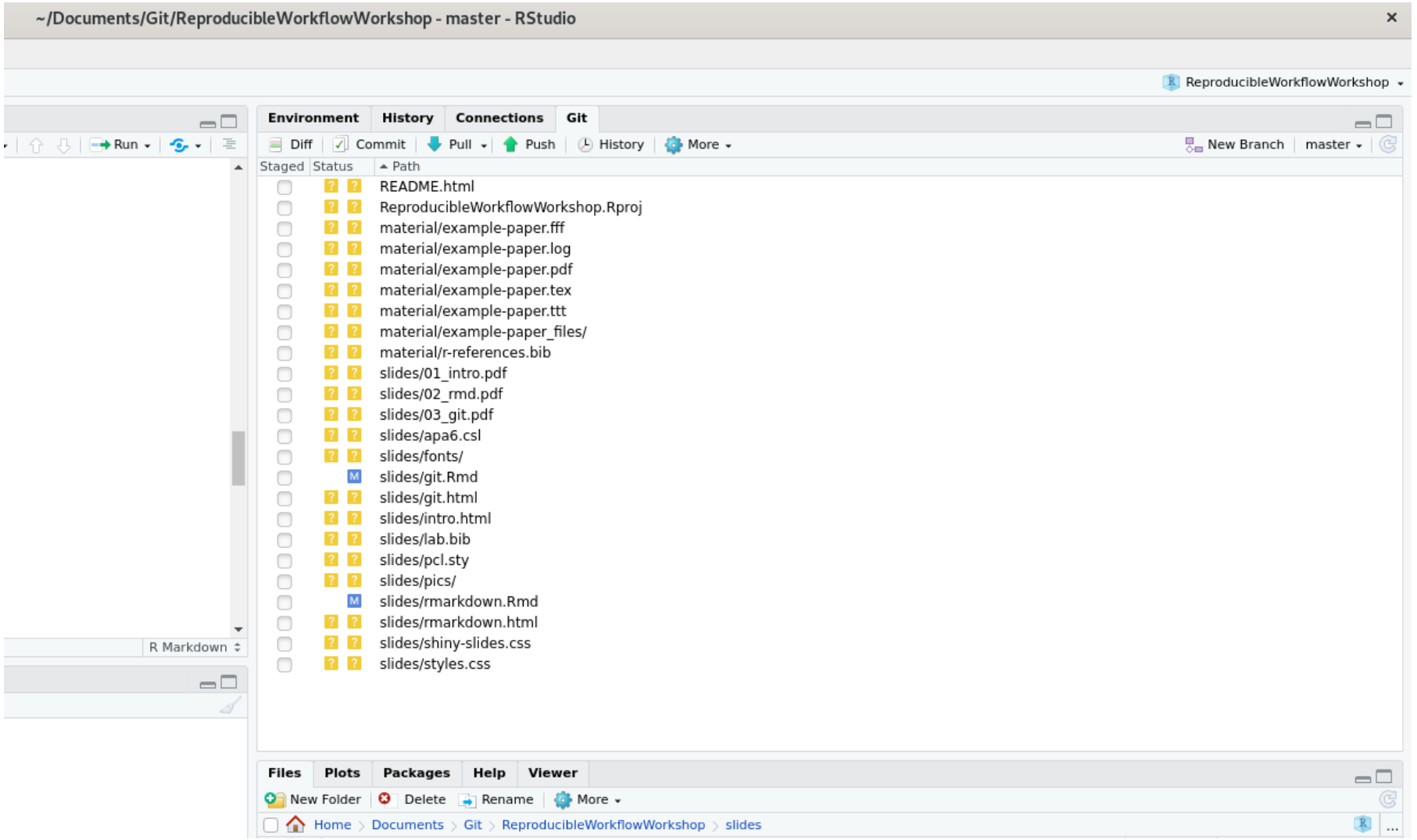
- A lot! Which is why I can only mention part of its functionality here.
- Working on one product in (large) teams.
- Working on things that can break.
- git can only integrate and show changes in text files.
- binary files (images, etc.) can be tracked and uploaded but changes cannot be shown.

Set-up for your computer

Using git

- Git does not have a user interface.
- You can either use the terminal, or install an additional interface.
- Github has its own GUI. Some people like it.
- We will use Rstudio (and the terminal) as user interface.

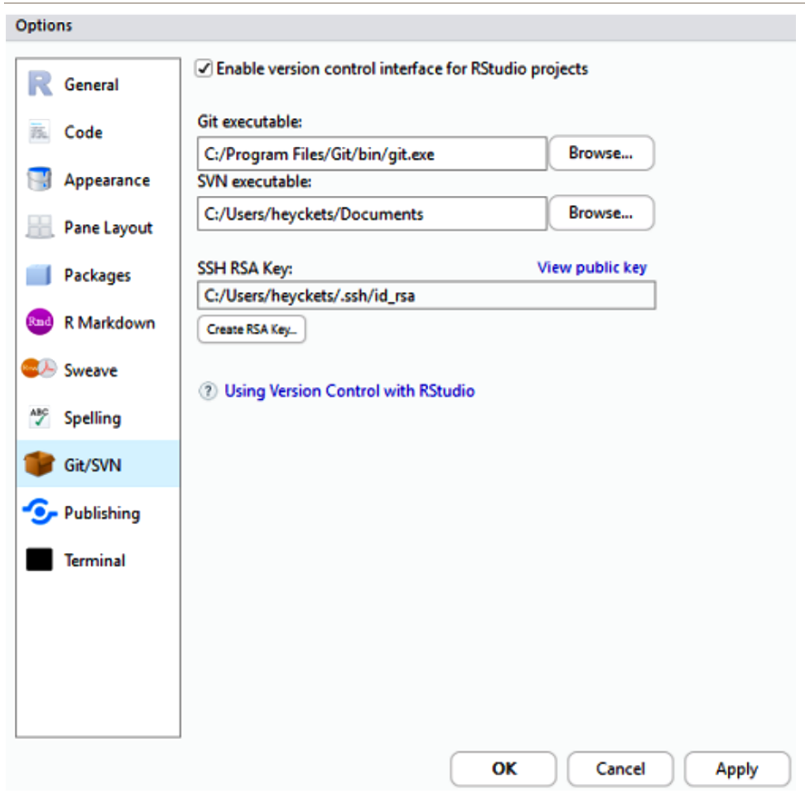
R Studio & git


~/Documents/Git/ReproducibleWorkflowWorkshop - master - RStudio
ReproducibleWorkflowWorkshop
Environment
History
Connections
Git
Diff
Commit
Pull
Push
History
More
New Branch
master
Staged
Status
Path
README.html
ReproducibleWorkflowWorkshop.Rproj
material/example-paper.fff
material/example-paper.log
material/example-paper.pdf
material/example-paper.tex
material/example-paper.ttt
material/example-paper_files/
material/r-references.bib
slides/01_intro.pdf
slides/02_rmd.pdf
slides/03_git.pdf
slides/apa6.csl
slides/fonts/
slides/git.Rmd
slides/git.html
slides/intro.html
slides/lab.bib
slides/pcl.sty
slides/pics/
slides/rmarkdown.Rmd
slides/rmarkdown.html
slides/shiny-slides.css
slides/styles.css
Run
R Markdown
Files
Plots
Packages
Help
Viewer
New Folder
Delete
Rename
More
Home
Documents
Git
ReproducibleWorkflowWorkshop
slides

R Studio & git

Tools ➤ Global Options ➤ Git/SVN.

Make sure the first box is ticked and “git.exe” (Windows) “usr/bin/git” (MacOS) is included in the first box.



Set name & email address

- Open the Terminal in R Studio.
- Set an email address and user name for git.

Set name & email address

- Open the Terminal in R Studio.
- Set an email address and user name for git.

```
git config --global user.email "myemail@email.com"
git config --global user.name "My commit name"
```

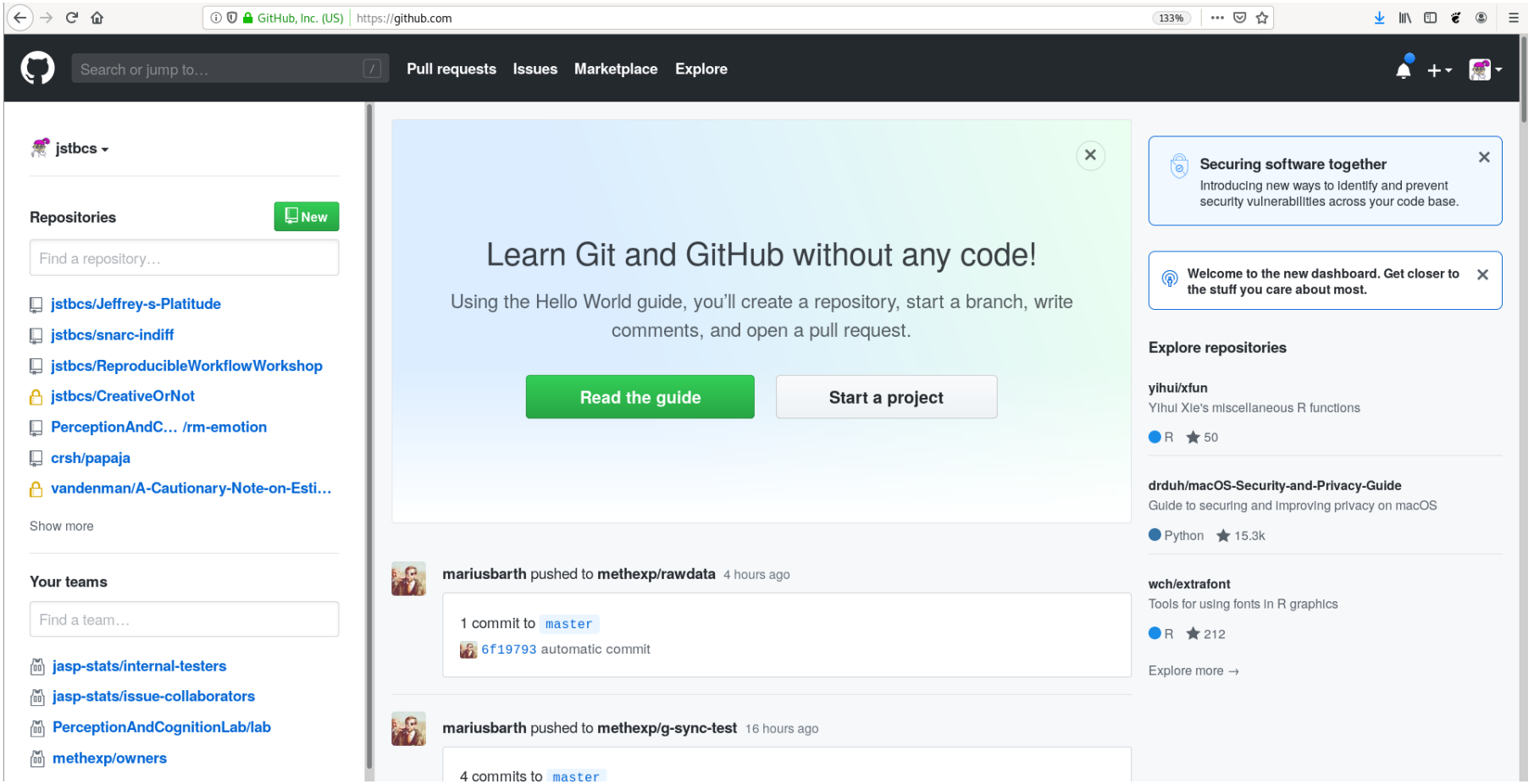
Set name & email address

- In the R console

```
## set your user name and email:
usethis::use_git_config(user.name = "My commit name", user.email = "myemail@email.com")
```

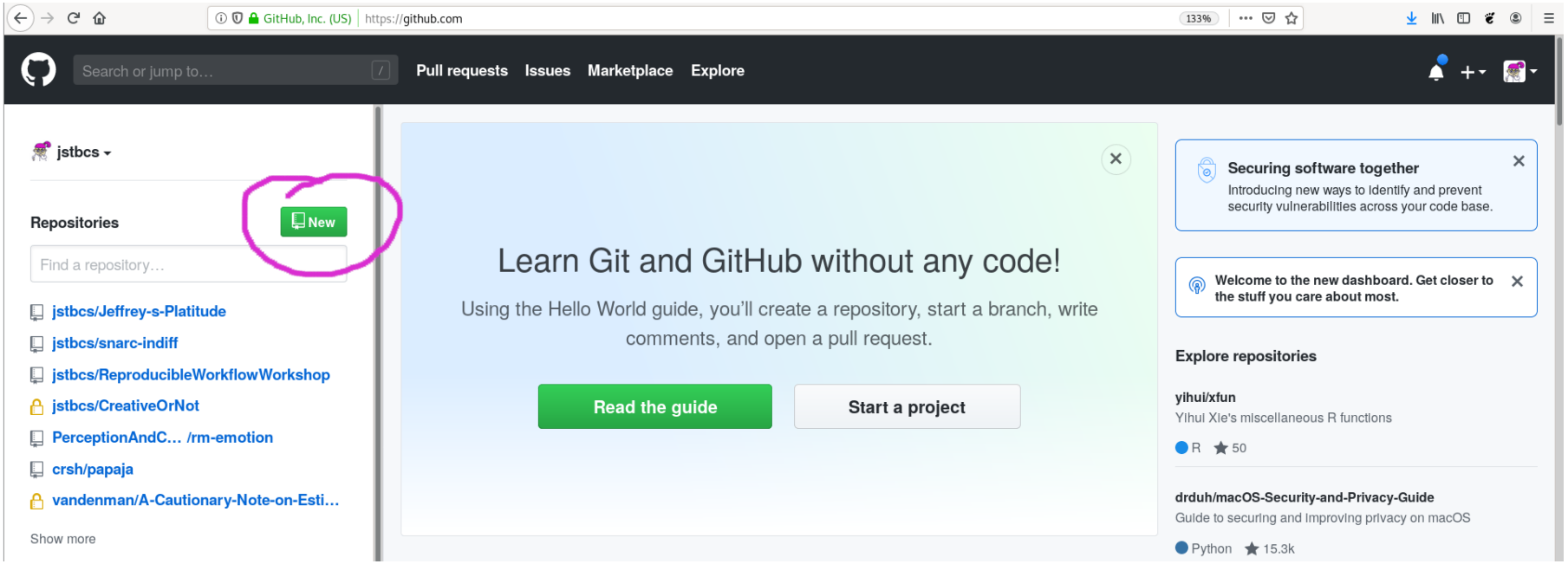
First Repository!

Github


GitHub, Inc. (US) https://github.com
Search or jump to…
Pull requests
Issues
Marketplace
Explore
jstbcs
Repositories
New
Find a repository…
jstbcs/Jeffrey-s-Platitude
jstbcs/snarc-indiff
jstbcs/ReproducibleWorkflowWorkshop
jstbcs/CreativeOrNot
PerceptionAndC… /rm-emotion
crsh/papaja
vandenman/A-Cautionary-Note-on-Esti…
Show more
Your teams
Find a team…
jasp-stats/internal-testers
jasp-stats/issue-collaborators
PerceptionAndCognitionLab/lab
methexp/owners
Learn Git and GitHub without any code!
Using the Hello World guide, you'll create a repository, start a branch, write comments, and open a pull request.
Read the guide
Start a project
mariusbarth pushed to methexp/rawdata 4 hours ago
1 commit to master
6f19793 automatic commit
mariusbarth pushed to methexp/g-sync-test 16 hours ago
4 commits to master
Securing software together
Introducing new ways to identify and prevent security vulnerabilities across your code base.
Welcome to the new dashboard. Get closer to the stuff you care about most.
Explore repositories
yihui/xfun
Yihui Xie's miscellaneous R functions
R 50
drduh/macOS-Security-and-Privacy-Guide
Guide to securing and improving privacy on macOS
Python 15.3k
wch/extrafont
Tools for using fonts in R graphics
R 212
Explore more →

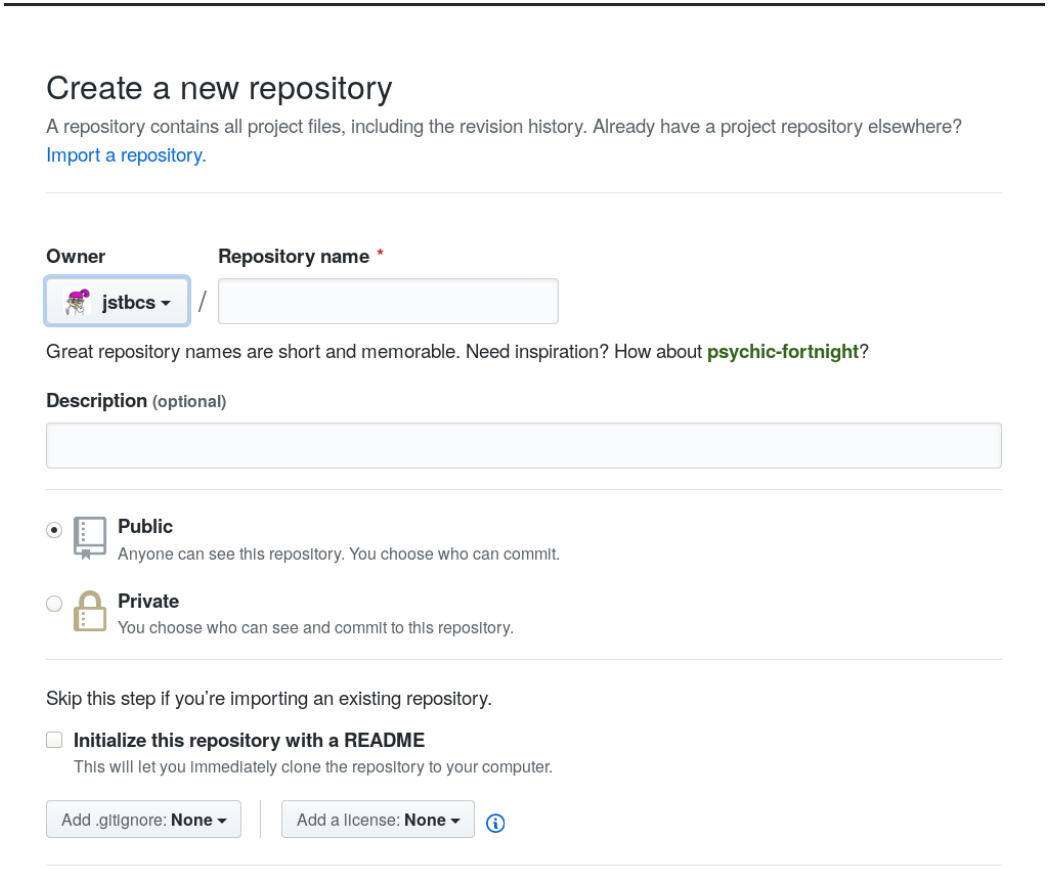
Github

New Repository



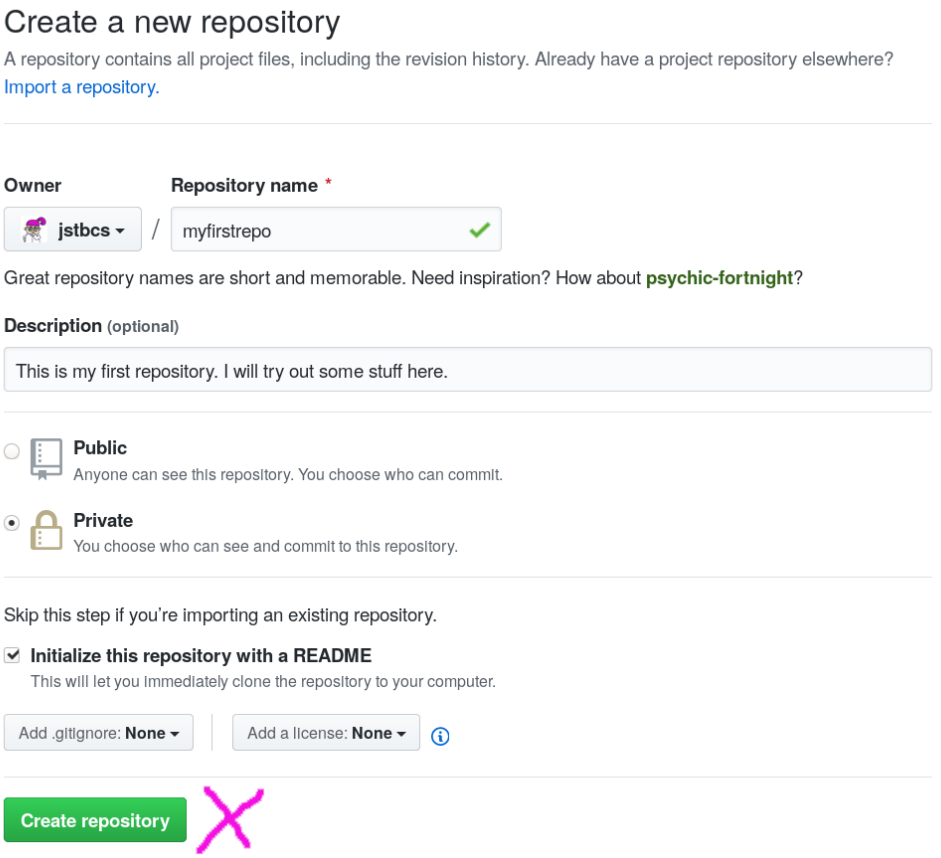
Github

New Repository



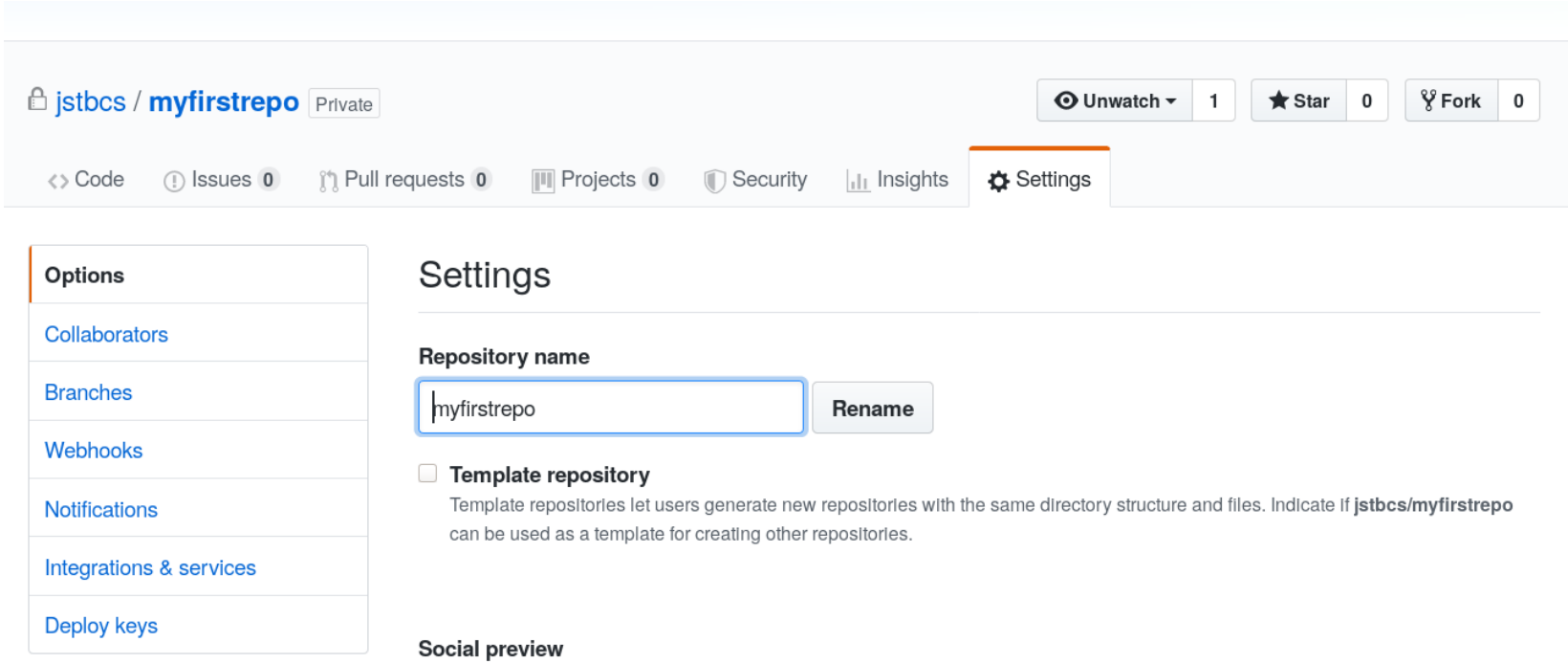
Github

New Repository



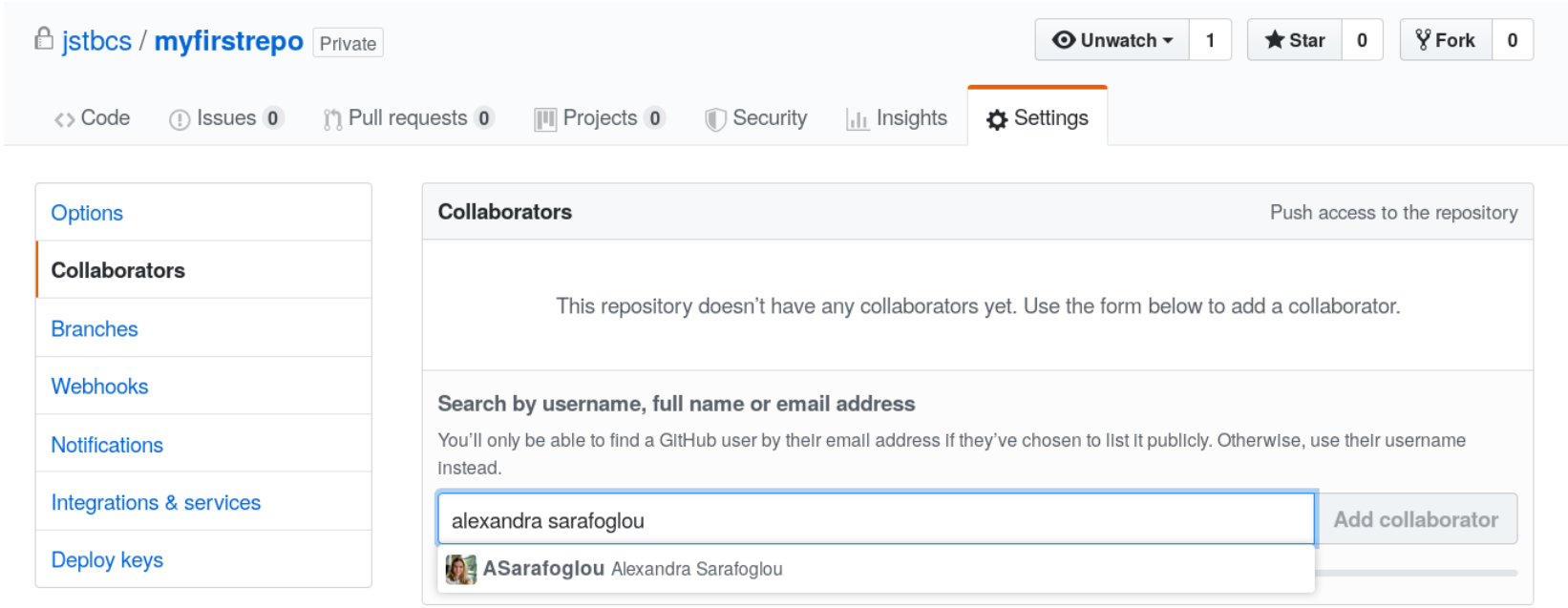
Github

Settings



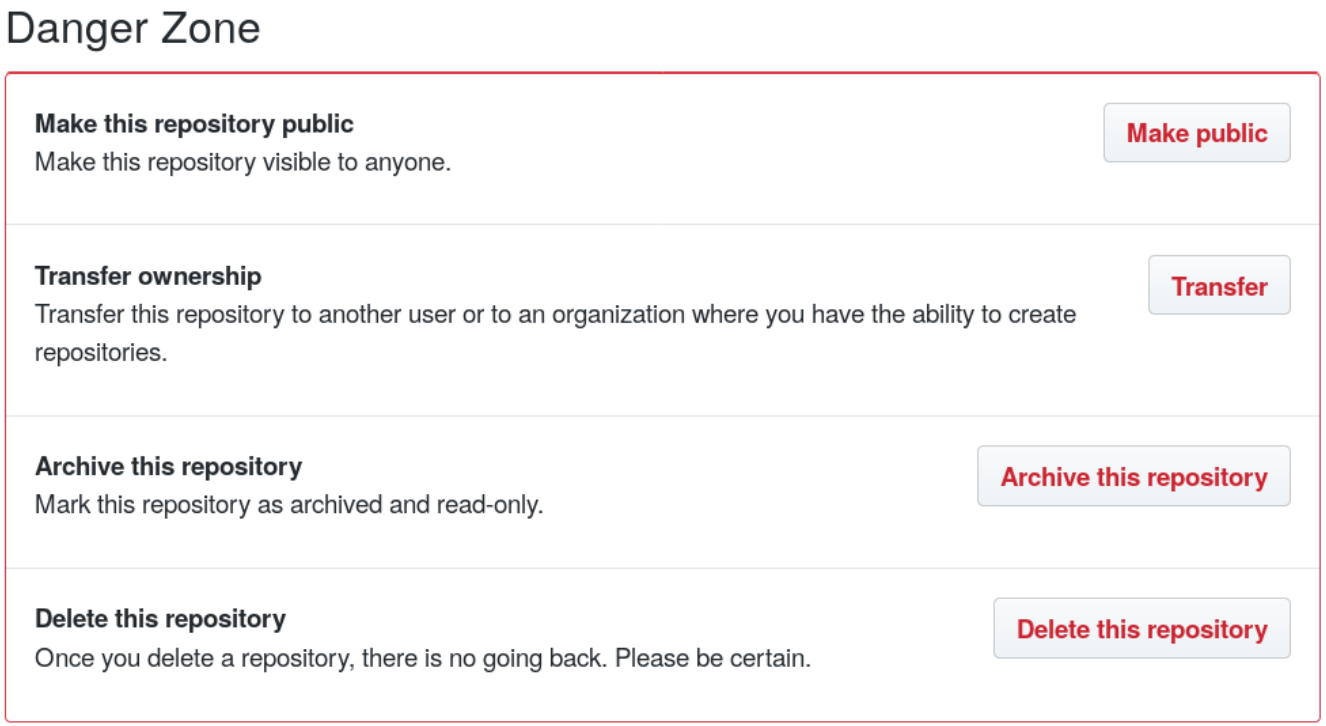
Github

Settings



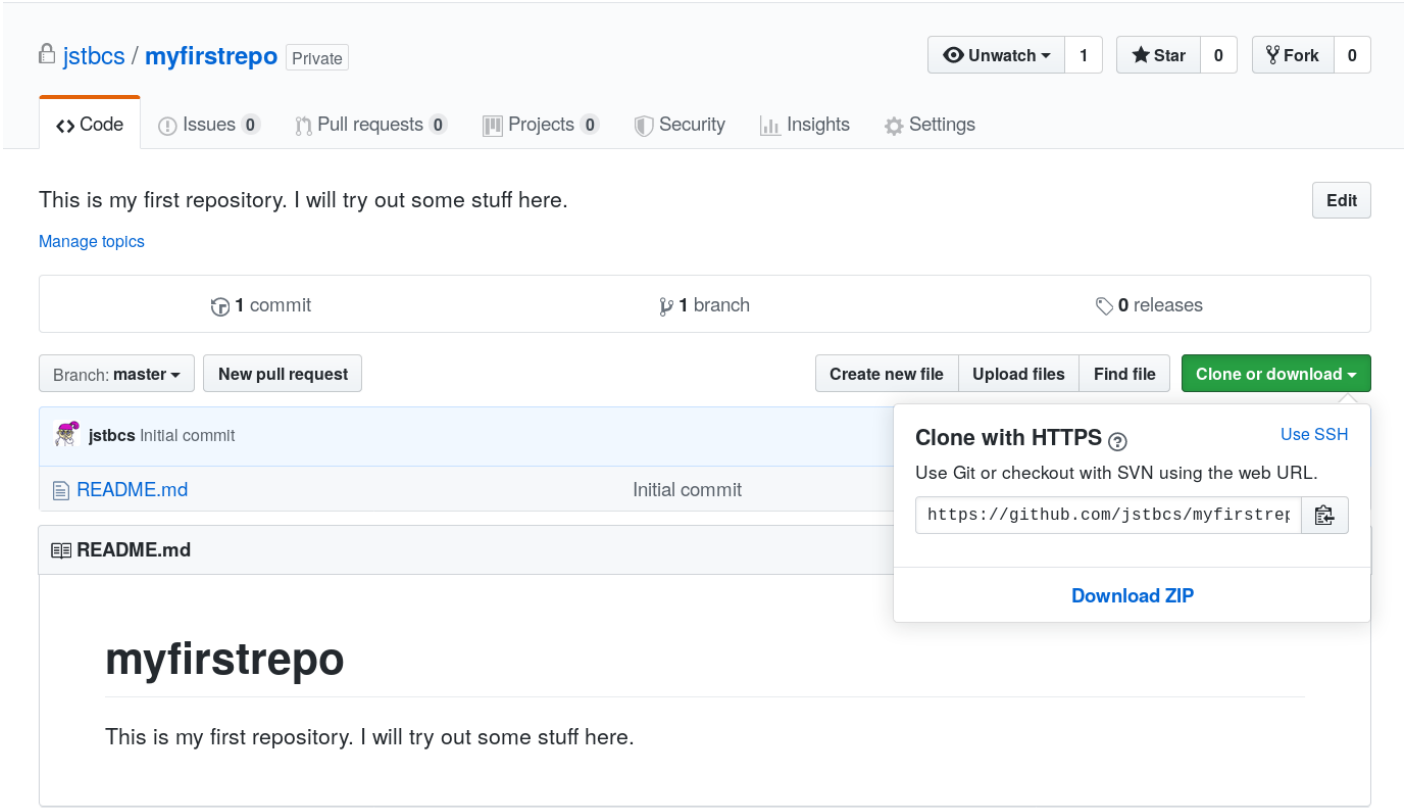
Github

Settings



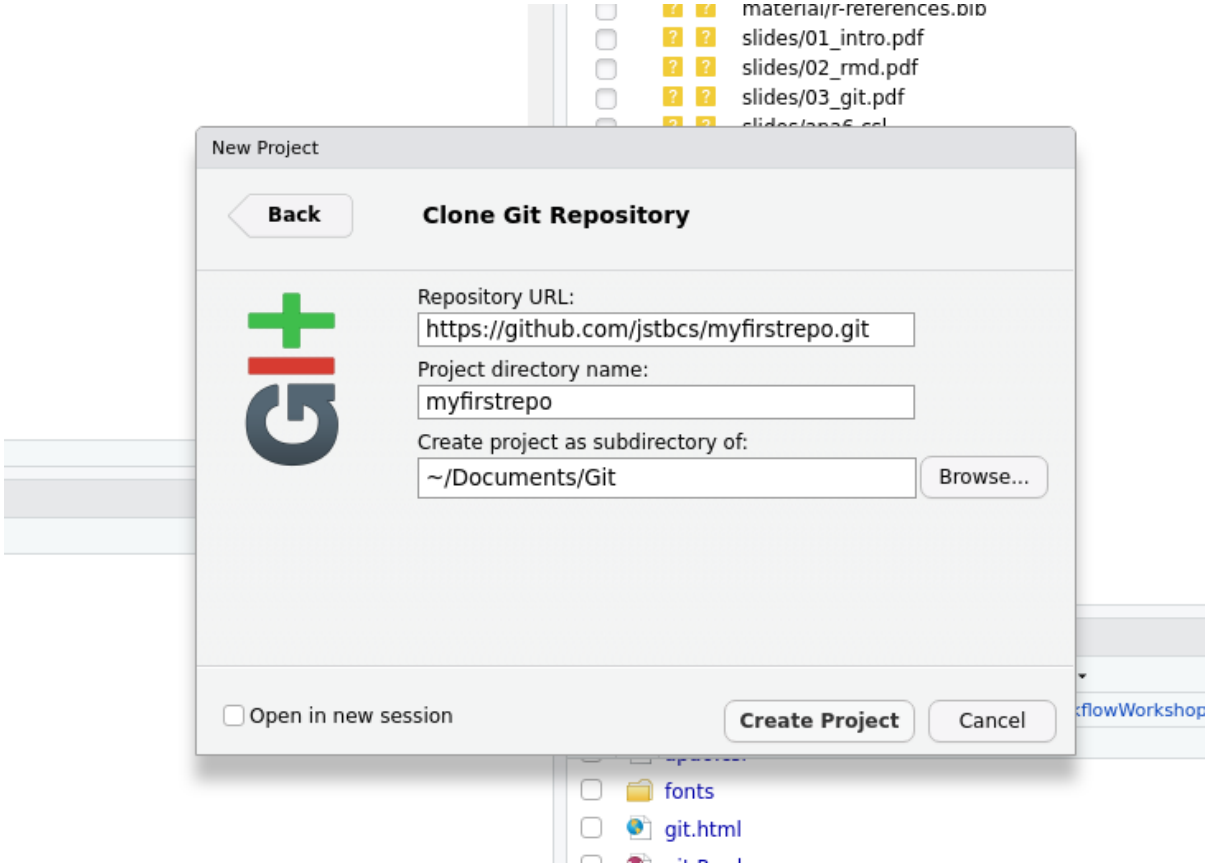
Github

Clone It!



In R Studio

File ➤ New Project ➤ Version Control ➤ Git

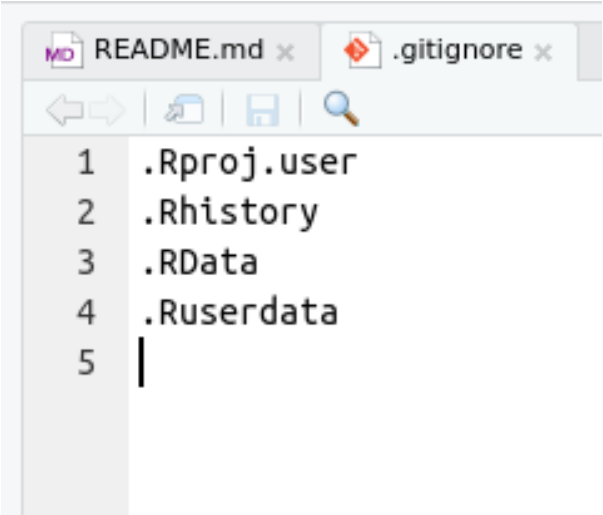


In R Studio

- Initializes a local git repository with an R project (opening a clean R Studio session when opening).
- You can see the README file from github.
- Adds a `.gitignore` file.

gitignore

- Specifies intentionally untracked files to ignore.
- Each line in a `gitignore` file specifies a pattern.
- R Studio pre-specifies some useful patterns.
- For R Markdown: Cache files! .tiff, .eps, .rdb, .rdx
- Pro tip: ?usethis::git_vaccinate()



README

README.md

Git and Rmarkdown: Two Useful Tools for a Fully Reproducible Workflow

This repository contains materials and slides for a workshop organized by the Open Science Community Amsterdam. It is meant as a short (two hour) teaser for different tools that can be used for a fully reproducible workflow in science.

Licence: cc by-sa-nc 4.0

Author: Julia Haaf

Abstract

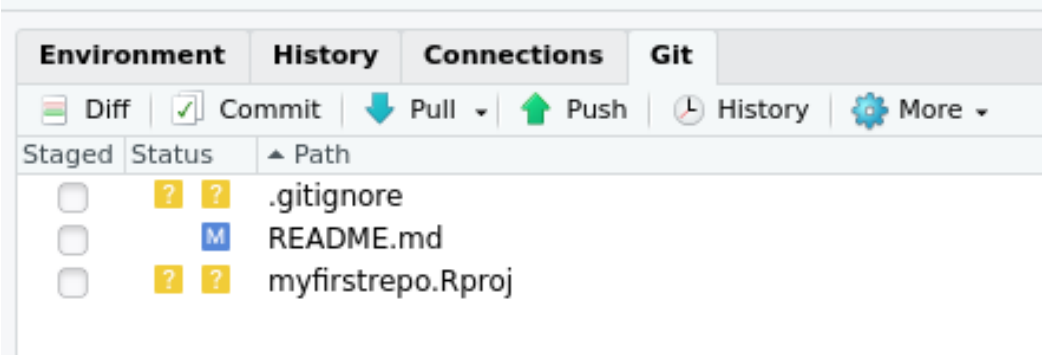
In the debate around the reproducibility crisis, most attention is paid to instances when researchers either trick themselves or others. Recently, however, some of the discussion has also shifted to scientific and reporting mistakes sneaking into the literature without anyone noticing. Whether these are mistakes in reporting statistics (Nuijten et al., 2016) or mistakes in the data collection process (Rouder et al., 2019), they prevent computational and empirical reproducibility of results in the literature. Rouder and colleagues (2019) proposed standardization and computer automation to minimize mistakes in the literature. In this workshop I will give an introduction to two tools we used for automation in the Perception and Cognition Lab at the University of Missouri. Git is a version control system that can be used to publish data or to collaboratively work

README

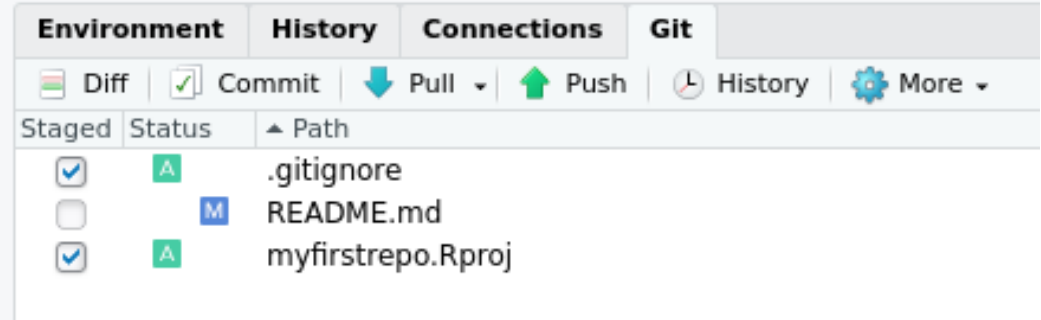
- Tell other people (and yourself in a year) why your project is useful, what they can do with your project, and how they can use it.
- On github default README files are Markdown files!

Git Workflow

Do some work


Environment
History
Connections
Git
Diff
Commit
Pull
Push
History
More
Staged
Status
Path
.gitignore
README.md
myfirstrepo.Rproj

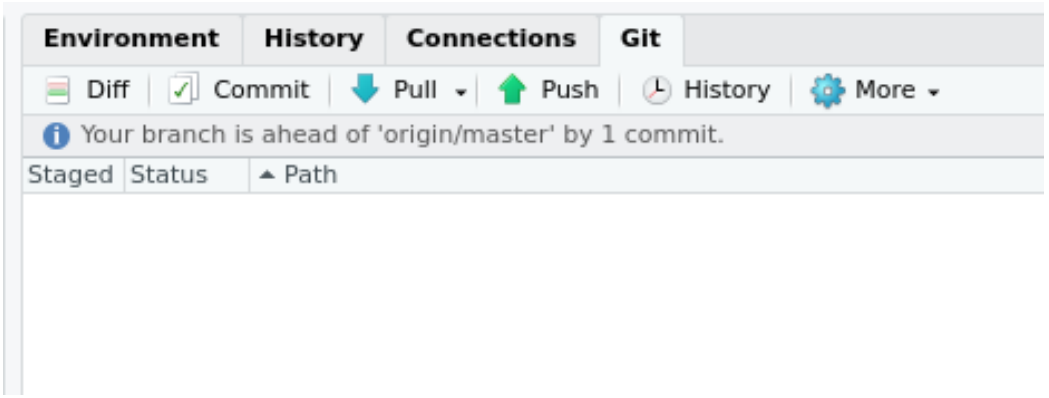
Git Add




```
git add .gitignore myfirstrepo.Rproj
```

- `git` can do auto-complete for file names!
- Note that many user interfaces combine `git add` and `git commit` (next step).

Git Commit

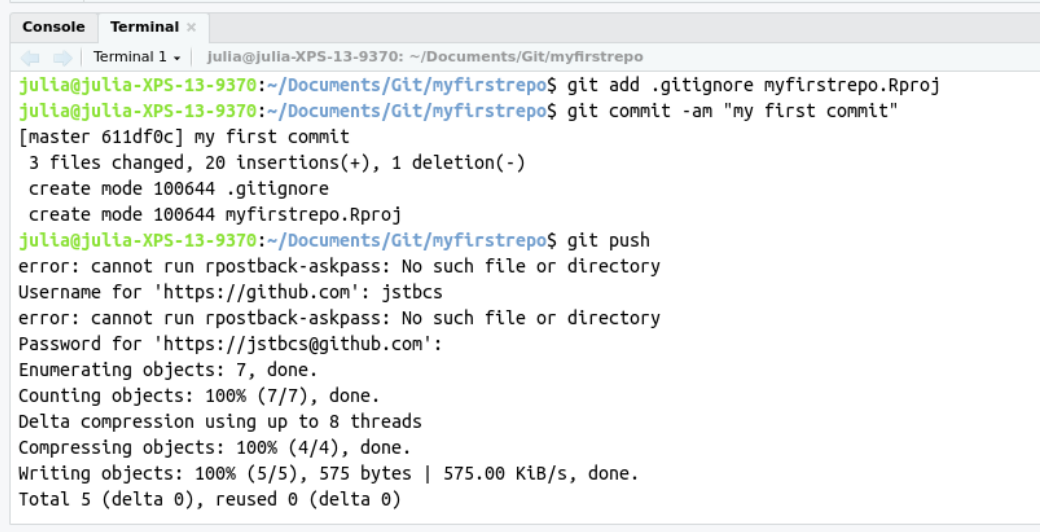



```
git add .gitignore myfirstrepo.Rproj
git commit -am "My first commit"
```

Commits always have a commit message.

Commit message

Git Push




```
git add .gitignore myfirstrepo.Rproj
git commit –am "My first commit"
git push
```

Congrats! You have done it! Now local and remote repositories are up to date!

Git Pull

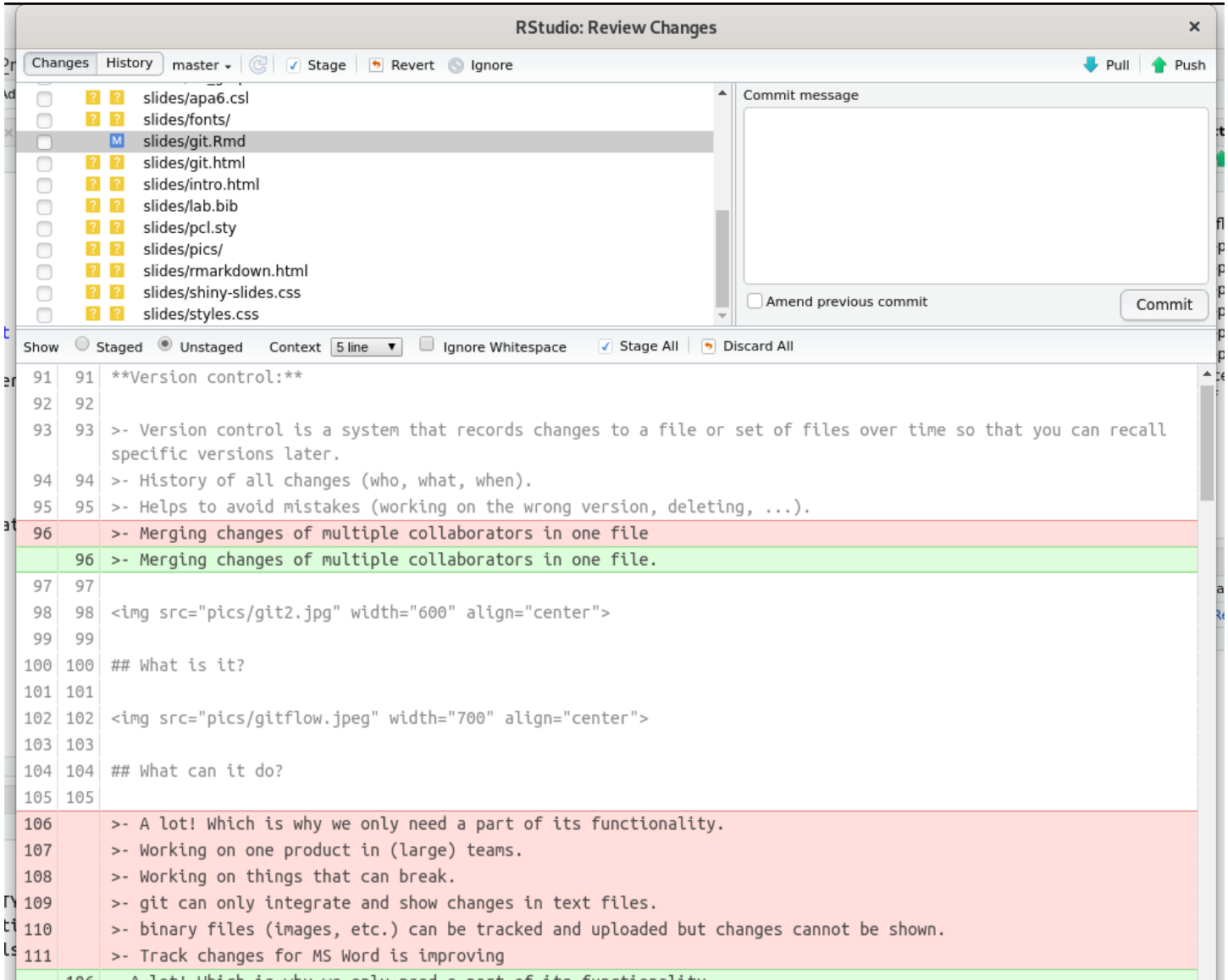
Before you start working on the project the next time (either in terminal or within R Studio):

```
git pull
```

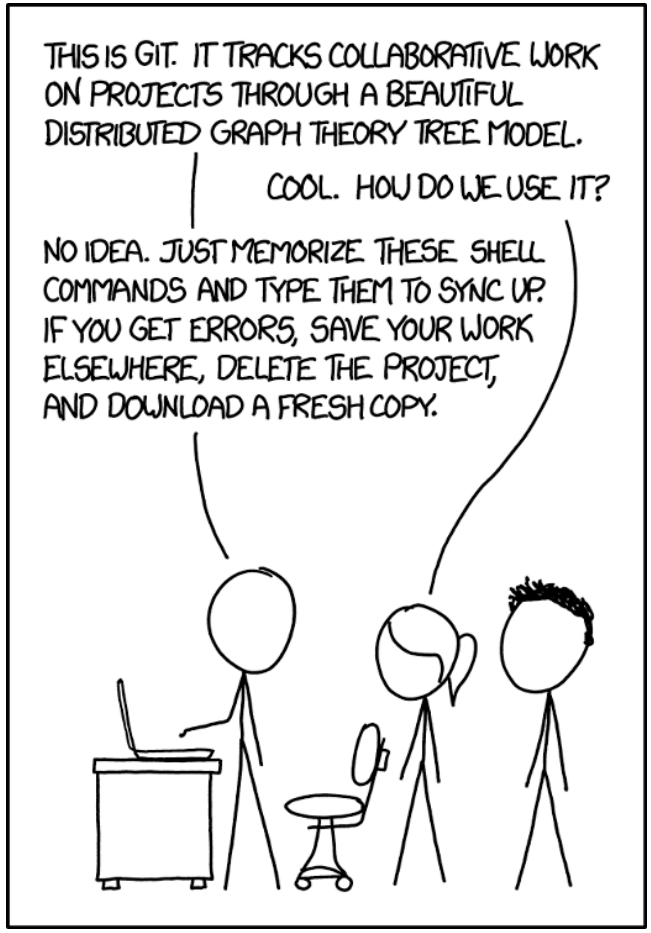
Pull, work some more, repeat.

What changed since the last commit?

`git diff`



What happens if something goes wrong? (And it will.)

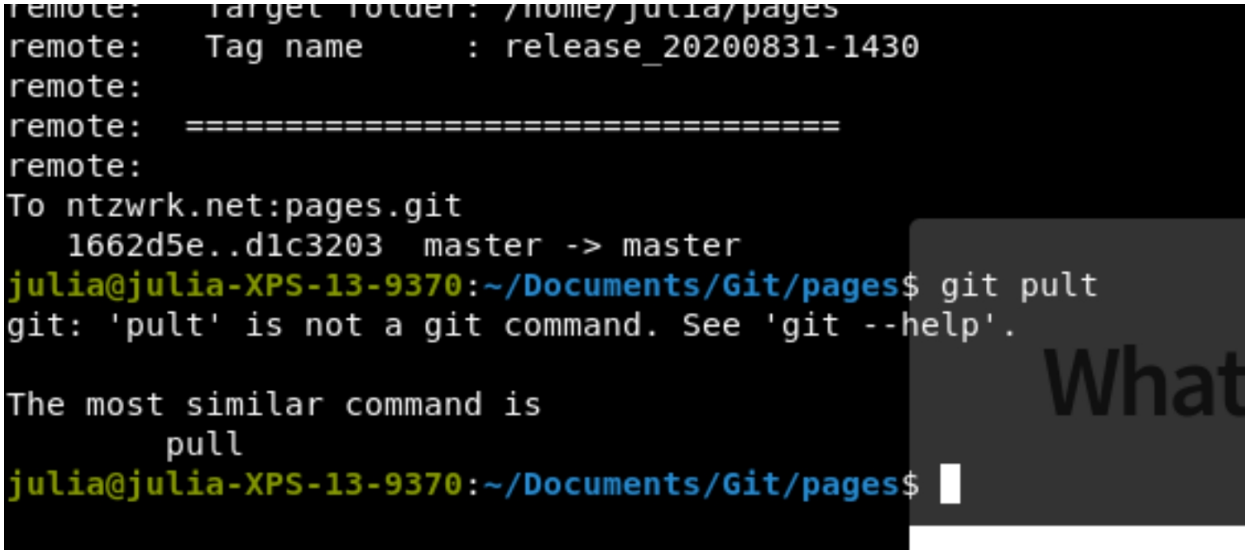


What happens if something goes wrong? (And it will.)

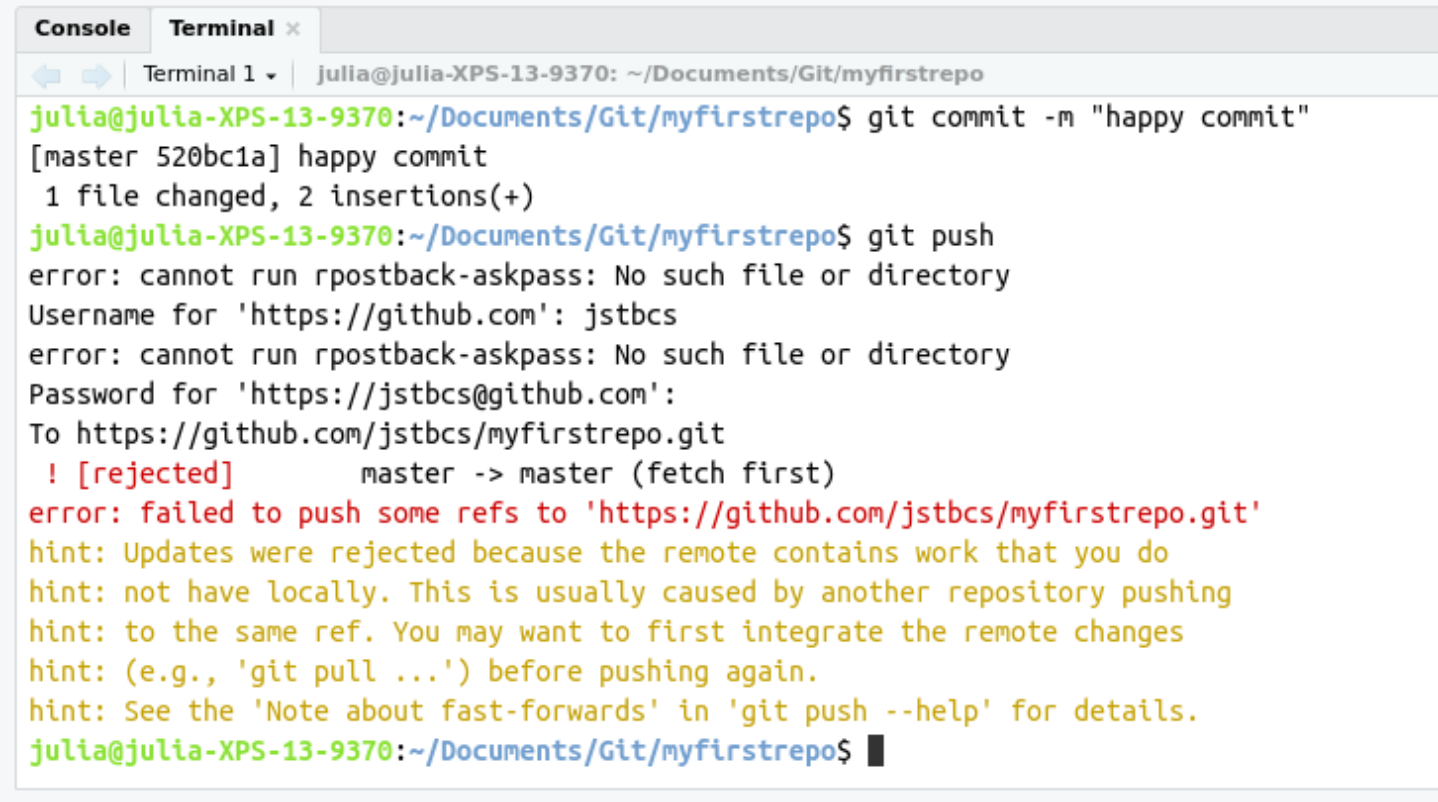
First things first: You cannot break things.

What happens if something goes wrong? (And it will.)

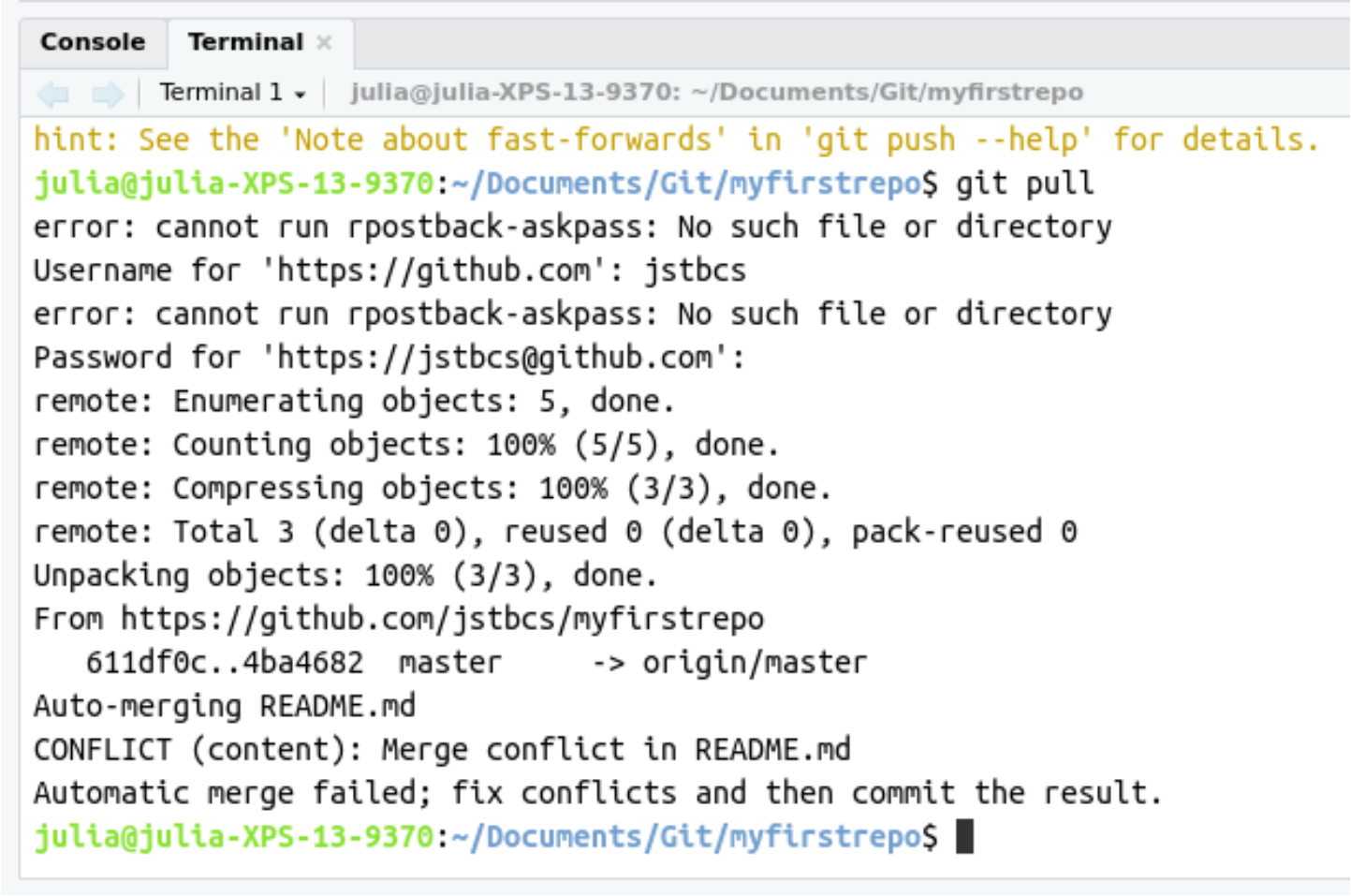
Read the output!



Push Conflict



Merge Conflict



Merge Conflict

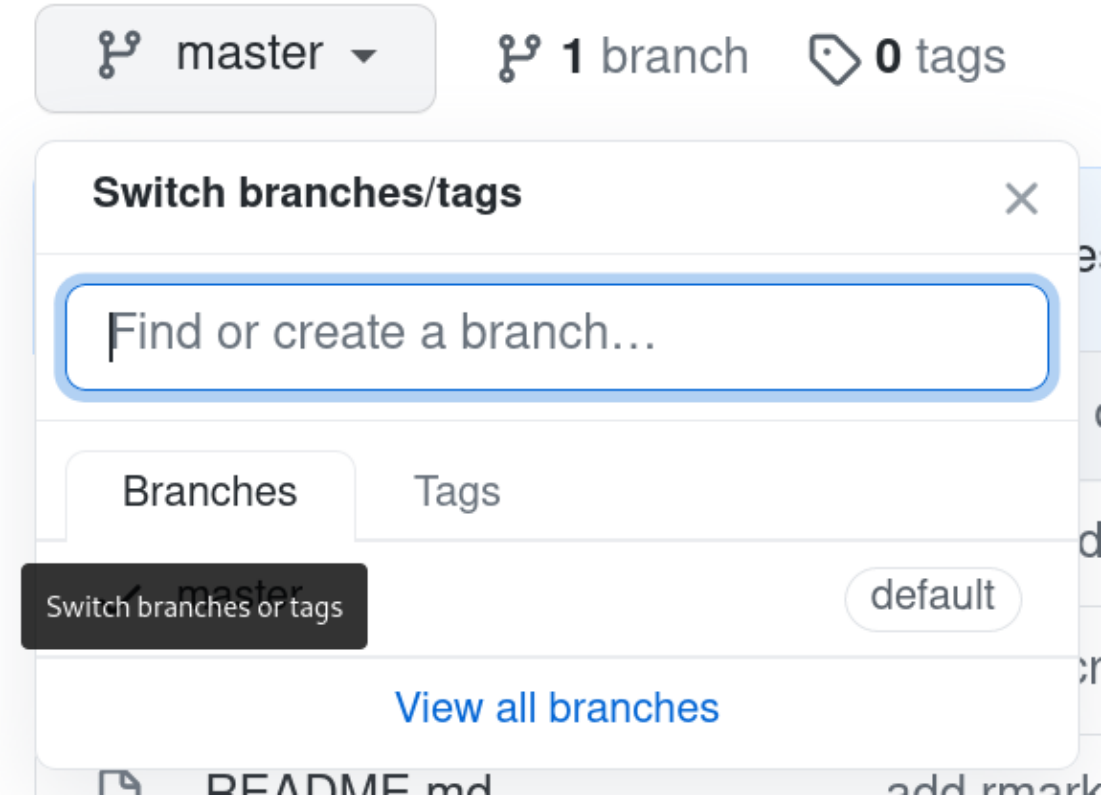
README.md | .gitignore | Preview

```
1  # myfirstrepo
2
3
4  This is my first repo. You cannot do anything with it.
5
6  <<<<<<< HEAD
7  What happens if I do this?
8  =======
9  Here I want a merge conflict.
10 >>>>>>> 4ba46823bce01a104a7aab219043d92f357f2898
11
```

- Resolve the conflict (Choose which changes to keep).
- Commit, and Push.

How to avoid merge conflicts

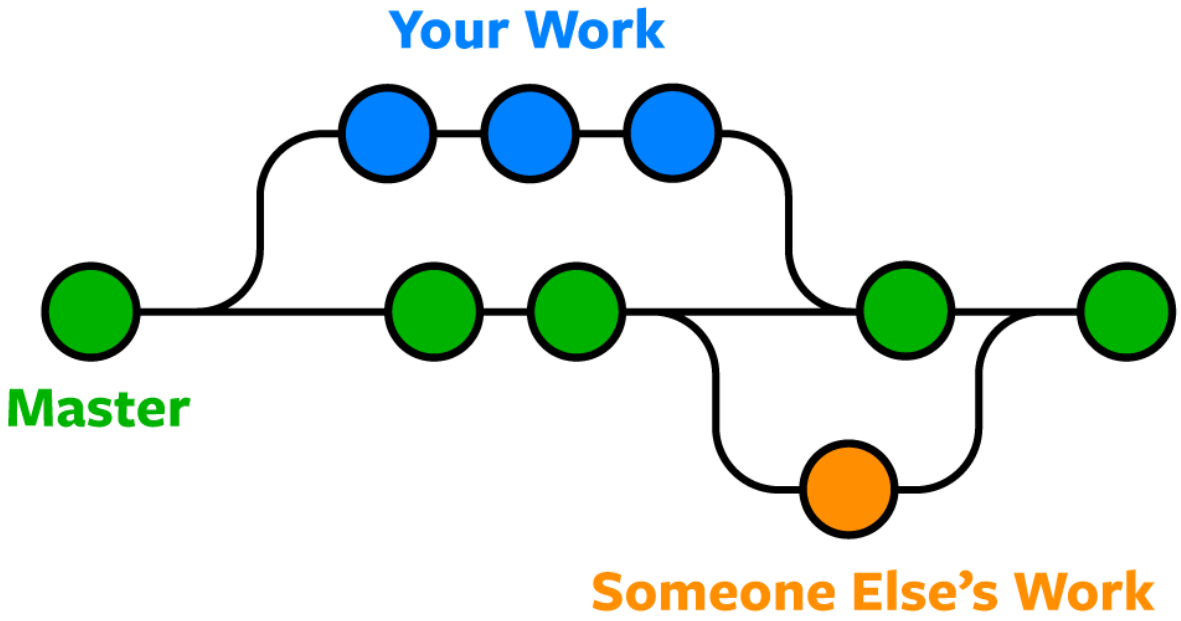
Or: Do not mess with the master



Do not mess with the master

- The master branch is *deployable*: It contains the final product code.
- This is not where people work.
- Instead you use *branches* for development and then *merge* the code into the master branch once you are done.

Do not mess with the master



Do not mess with the master

I don’t really have time to get into this

- Here is the best tutorial I could find on branches: https://thenewstack.io/dont-mess-with-the-master-working-with-branches-in-git-and-github/.
- For working in a professional environment: A tutorial on pull requests: https://www.atlassian.com/git/tutorials/making-a-pull-request

Summary

- Add, commit, push, pull.
- Use it!
- git documentation and error tracking are great!

Your turn!

- Complete the in-class assignment.

References

Additional info (in the material folder):

Vuorre, M., & Curley, J. P. (2018). Curating Research Assets: A Tutorial on the Git Version Control System. *Advances in Methods and Practices in Psychological Science, 1*(2), 219–236.