Lucerne R User Group

Using GitHub in RStudio

Dr Andrea De Angelis

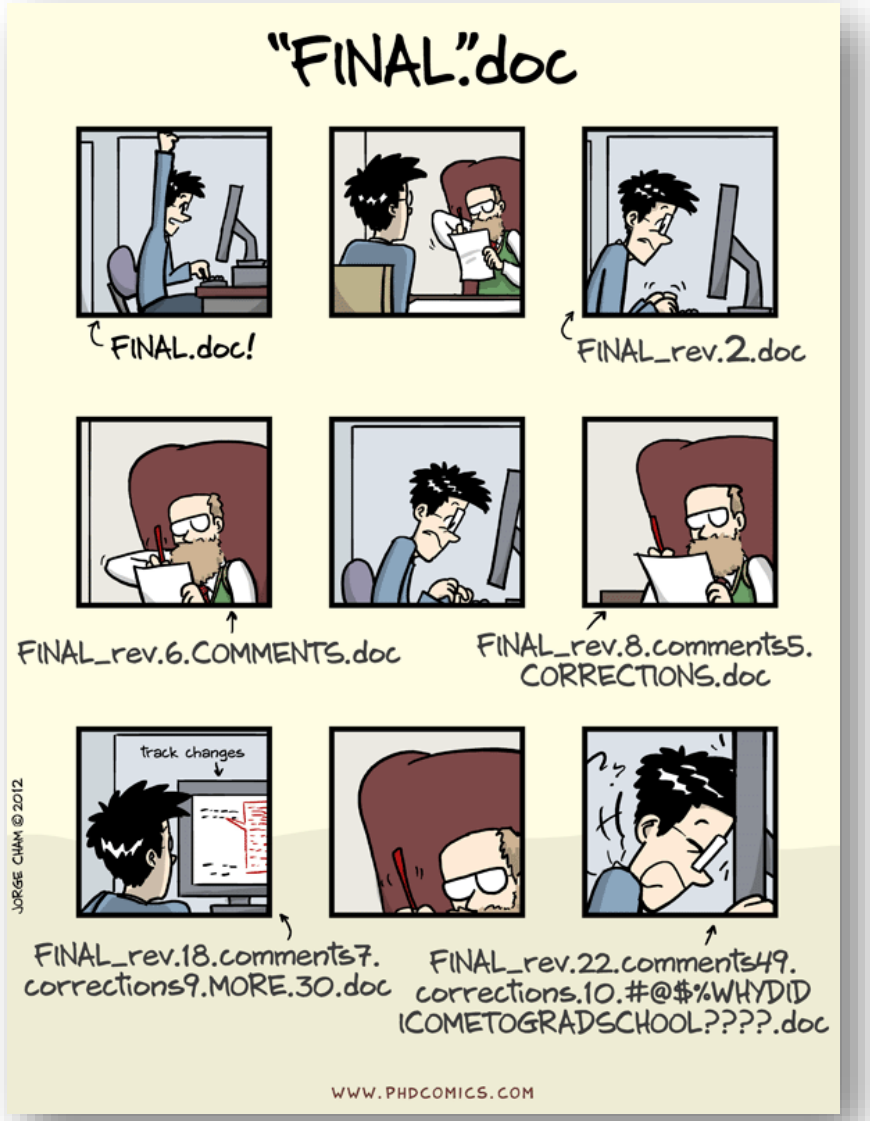
28 October 2020

https://www.meetup.com/Lucerne-R-User-Group/

https://github.com/Lucerne-R-User-Group/2020_06_24-meeting-2

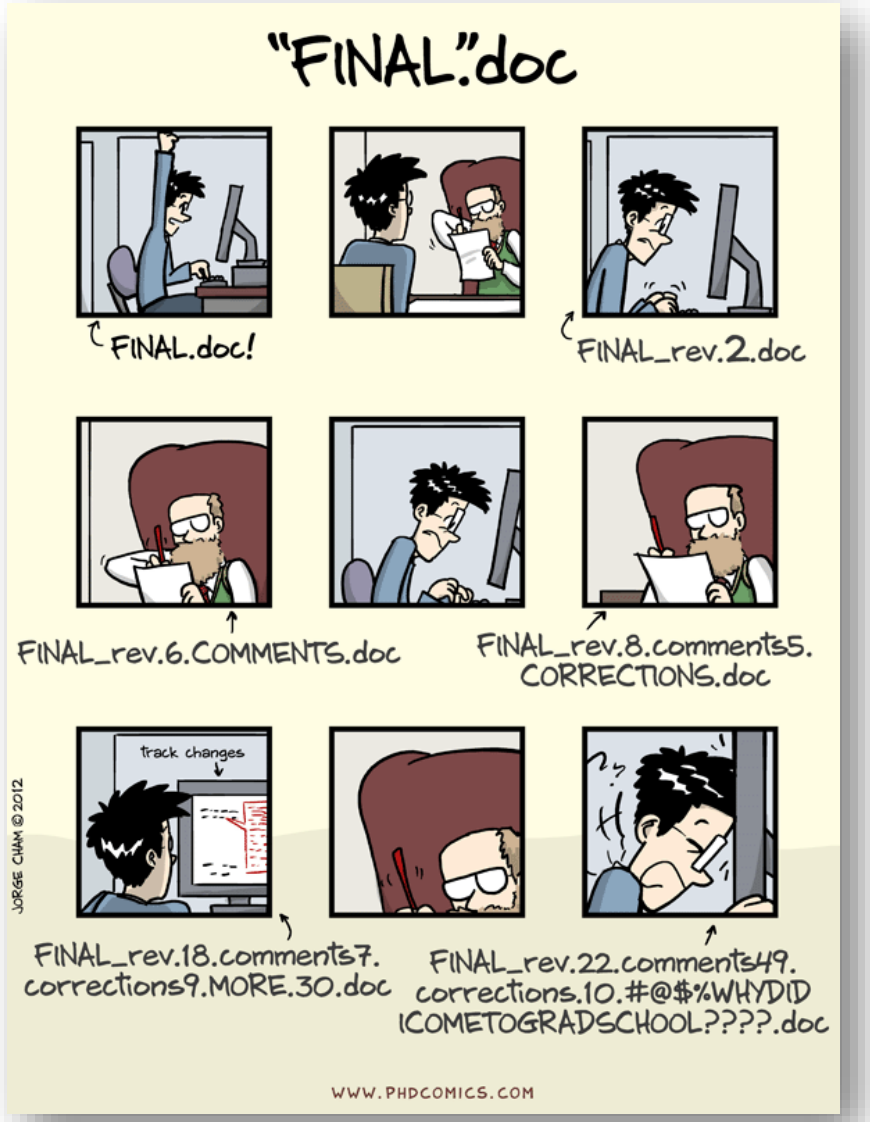
Version Control?
Why?

File system

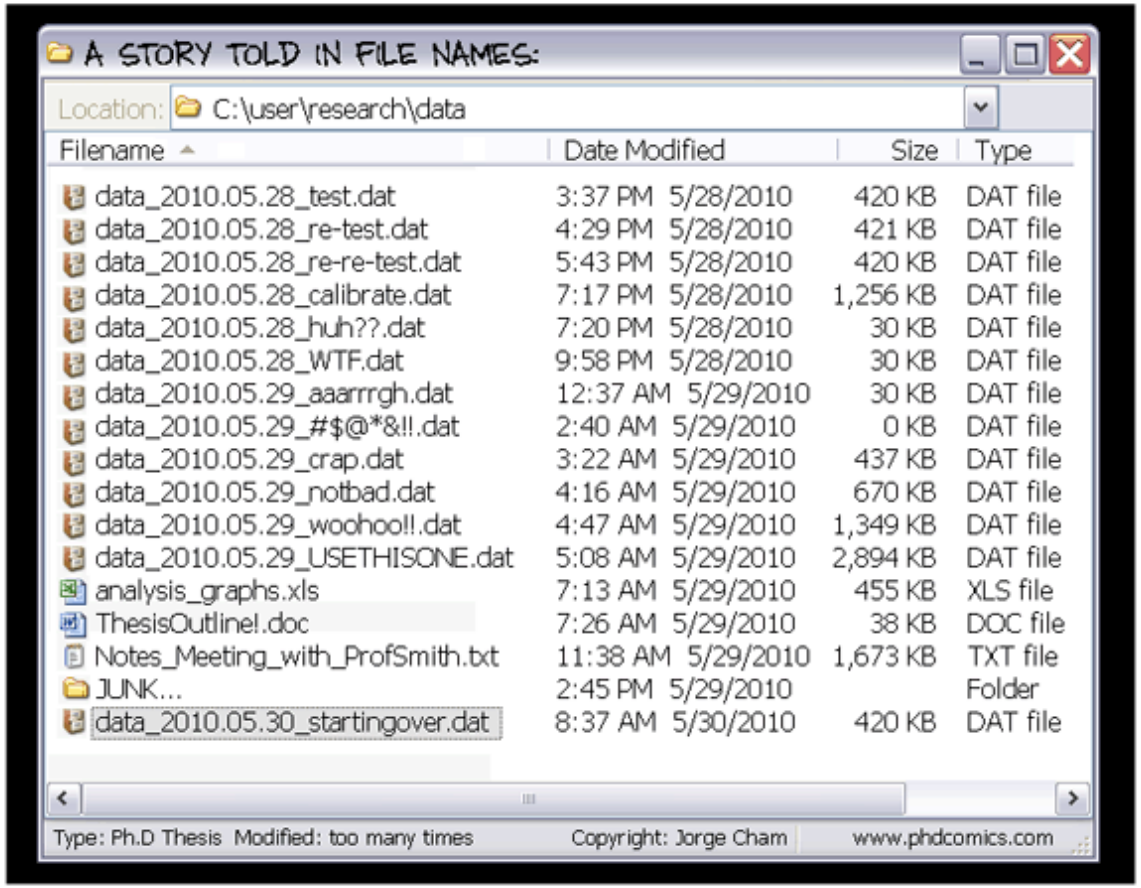


http://phdcomics.com/comics/archive.php?comicid=1531

File system

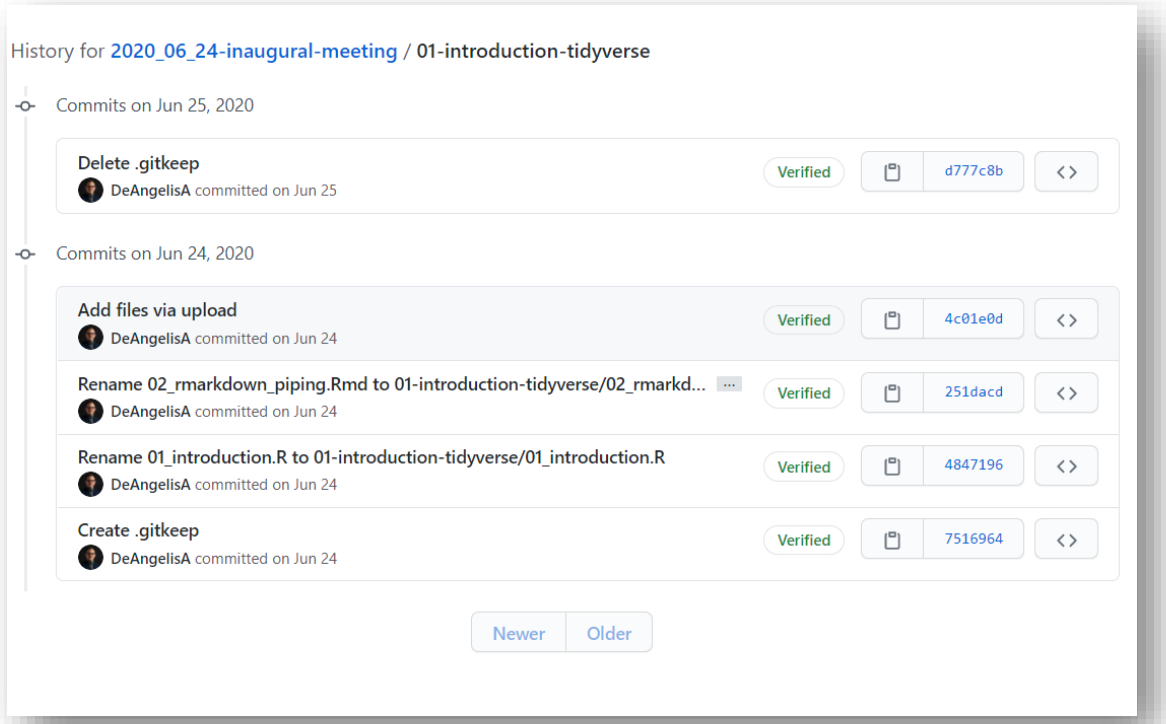


http://phdcomics.com/comics/archive.php?comicid=1531



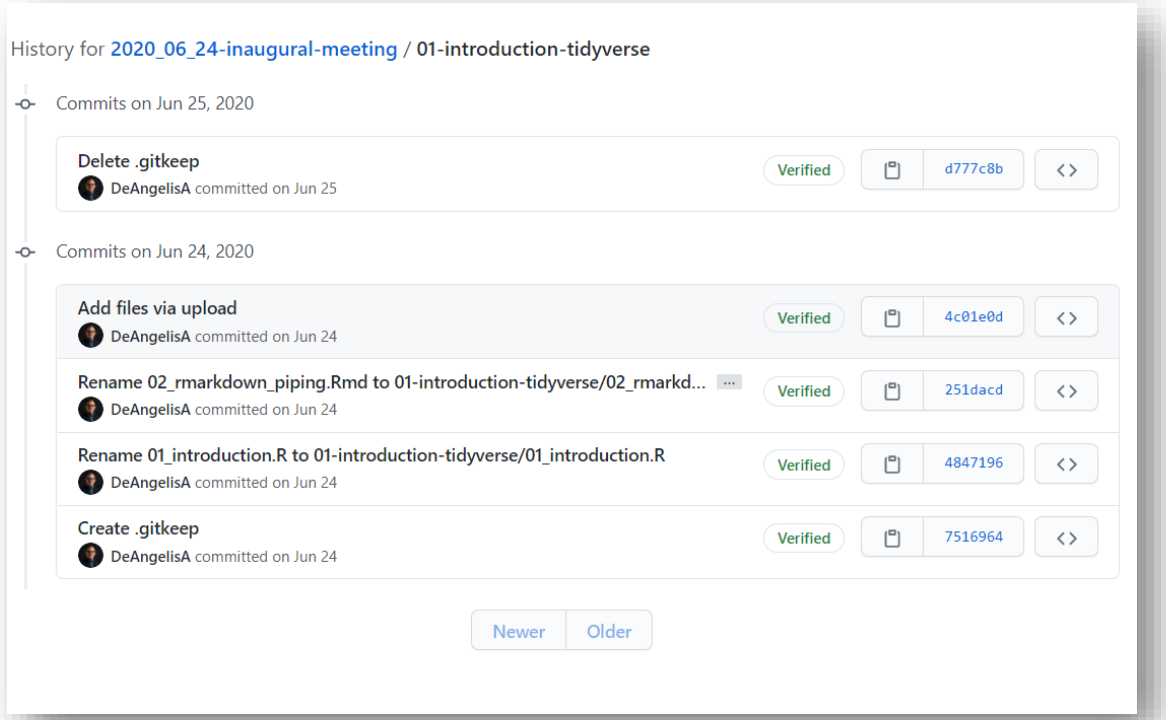
https://phdcomics.com/comics/archive_print.php?comicid=1323

Version Control system

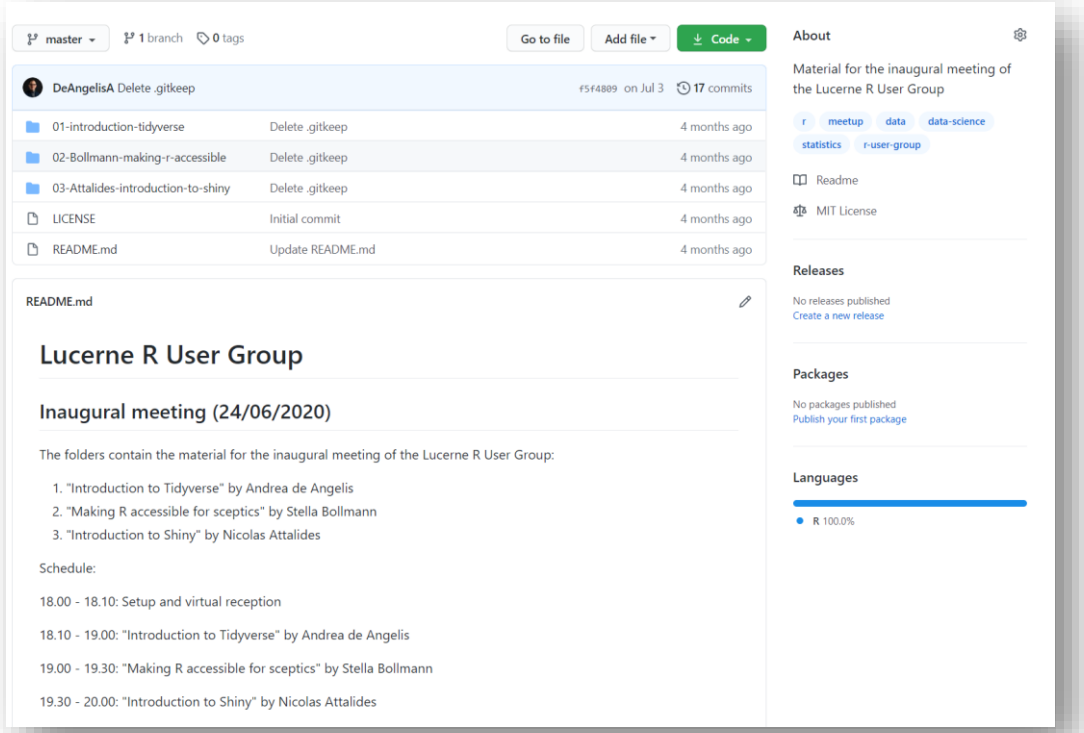


Every change is tracked and revertible

Version Control system



Every change is tracked and revertible



You see the last version

Version Control system

Version-control System, VCS
a sytem to manage and organize the story of a project

Git
the most popular distributed version-control systems (free and open-source)

GitHub
Web-hosting service for version control using Git

Getting started with git and GitHub

Setup

1. Install git

[Supporting link]

%>%

(read: 'then')

2. Activate Git on RStudio

[Supporting link]

%>%

3. Configure Git

[Supporting link]

%>%

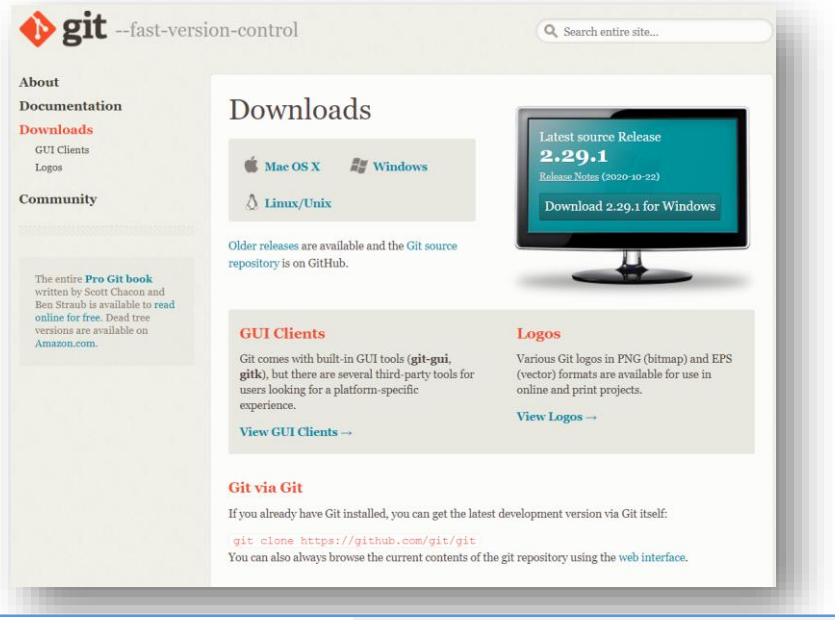
4. Configure GitHub

[Supporting link]

Let's git started!

1. Install / update Git

- Download Git at this link: https://git-scm.com/downloads
- Click on the file and go through the installation process



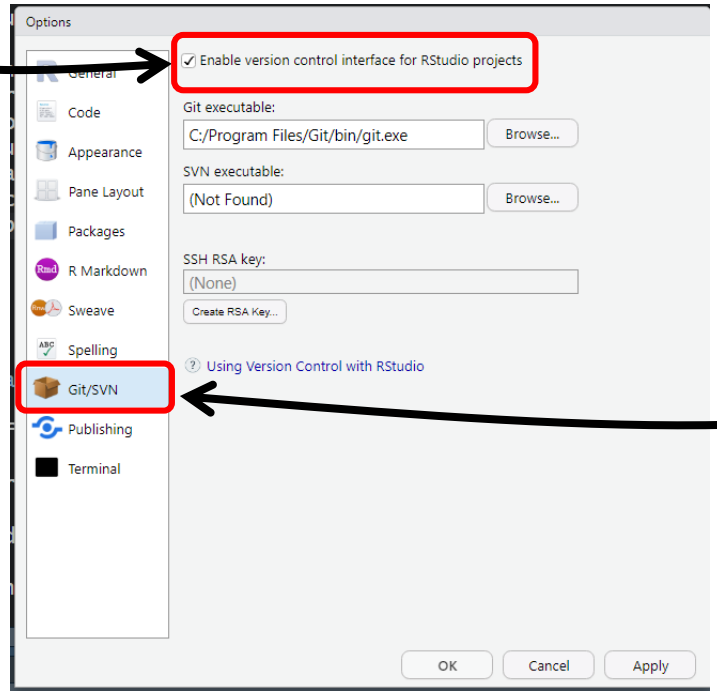
Keep all default settings: Next > Next > Next >

02:00

Let's git started!

1. Install / update Git
2. **Activate Git on RStudio**

- Download Git at this link
- Click on the file and go through the installation process
- In RStudio: **`Tools %>% Global Options %>% Git/SVN`**
- Tick “Enable version control interface for RStudio projects”



00:30

Let's git started!

1. Install / update Git
2. Activate Git on RStudio
3. **Configure Git**

- Download Git at this link
- Click on the file and go through the installation process
- In RStudio: `Tools %>% Global Options %>% Git/SVN`
- Tick "Enable version control interface for RStudio projects"
- Search the **Git Bash** program and open it

Display you Git configurations typing:

```
$ git config --list
```

Reference
https://happygitwithr.com/hello-git.html

00:30

Let's git started!

1. Install / update Git
2. Activate Git on RStudio
3. **Configure Git**

- Download Git at this link
- Click on the file and go through the installation process
- In RStudio: `Tools %>% Global Options %>% Git/SVN`
- Tick "Enable version control interface for RStudio projects"
- Search the **Git Bash** program and open it 
- **Set a Git username** typing:

```
$ git config --global user.name "YOUR FULL NAME"
```

- **Set your commit email** typing:

```
$ git config --global user.email "YOUR EMAIL"
```

Reference
https://happygitwithr.com/hello-git.html

01:00

Let's git started!

1. Install / update Git
2. Activate Git on RStudio
3. Configure Git
4. **Set GitHub**

- Download Git at this link
- Click on the file and go through the installation process
- In RStudio: `Tools %>% Global Options %>% "Git/SVN"`
- Tick “Enable version control interface for RStudio projects”
- Search the **Git Bash** program and open it
- **Set a Git username**
- **Set your commit email**
- **Sign up** on GitHub.com: review ToS, Privacy and click "Create Account"
- Choose the Educational plan if you qualify

02:00

Let's git started!

1. **Install / update Git**
2. **Activate Git on RStudio**
3. **Configure Git**
4. **Set GitHub**
5. **Validate Git email**

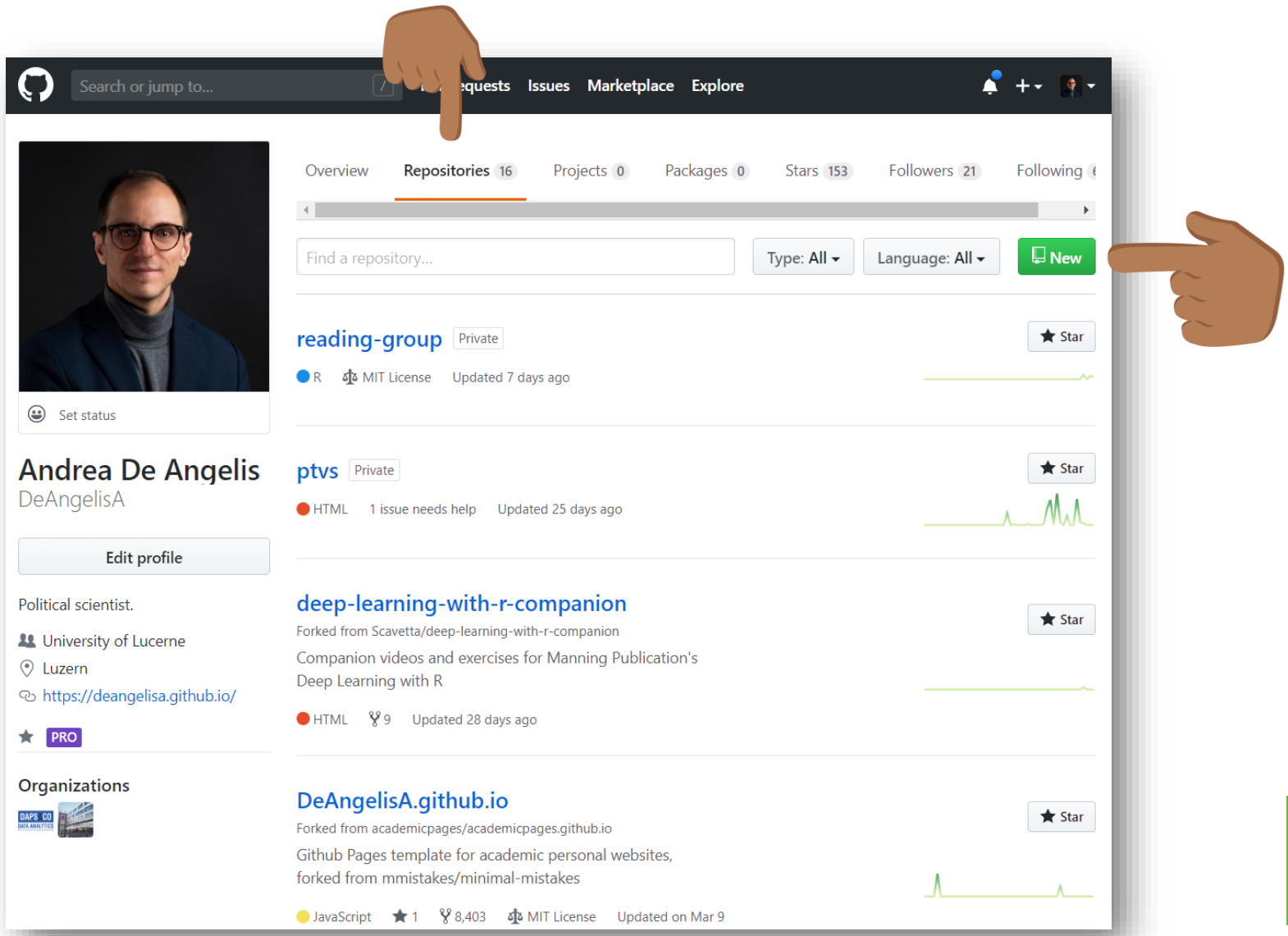
- Download Git at this link
- Click on the file and go through the installation process
- In RStudio: `Tools %>% Global Options %>% "Git/SVN"`
- Tick “Enable version control interface for RStudio projects”
- Search the **Git Bash** program and open it
- **Set a Git username**
- **Set your commit email**
- Register on GitHub.com: review ToS, Privacy and click "Create Account"
- Choose the Educational plan if you qualify
- GitHub **%>%** Click your photo **%>%** `Settings` **%>%** `Email`
- Add and verify your **Git email** **%>%** "Keep my email address private"

02:00

Git basics

Initialize a GitHub "repo"

"New project, GitHub first" workflow

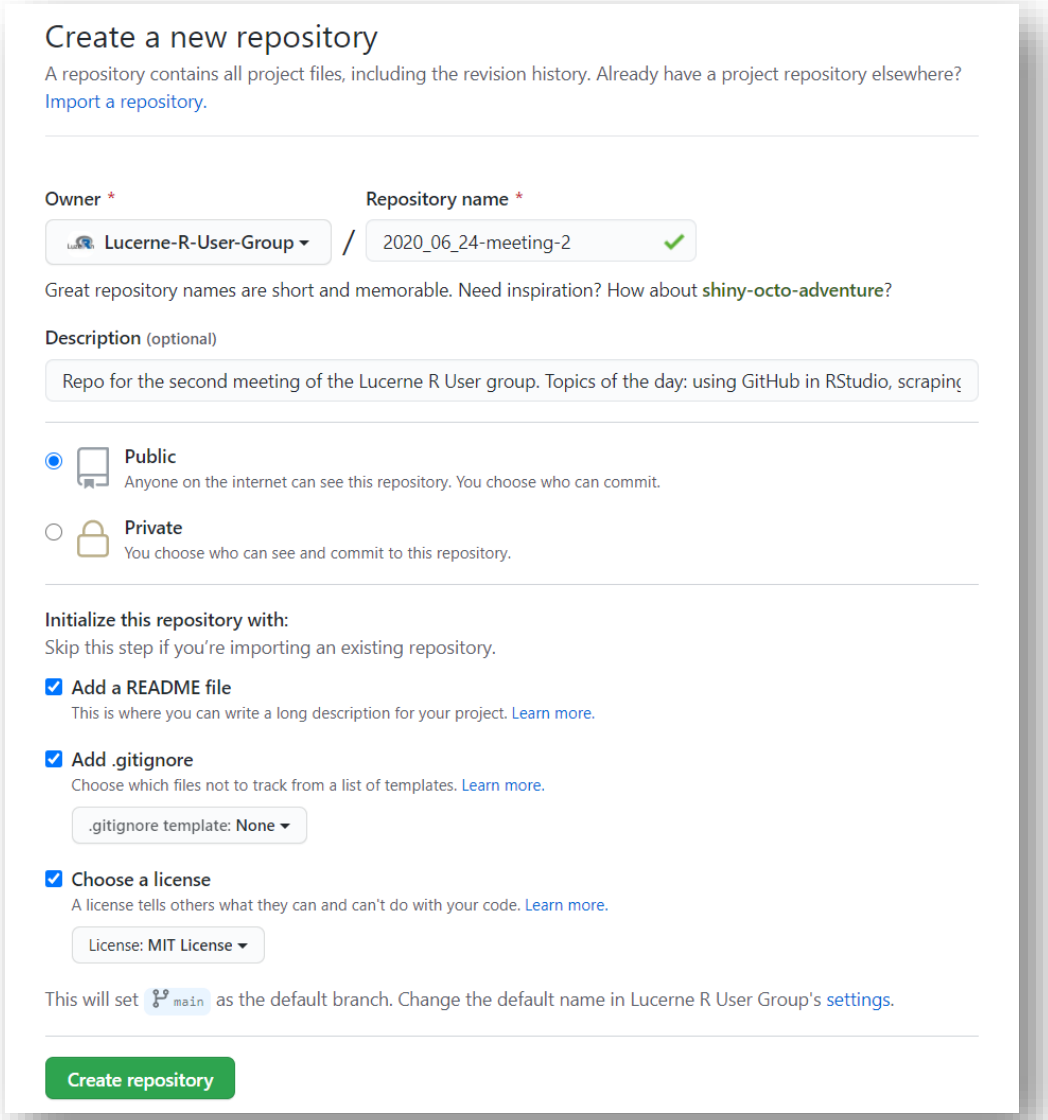


GitHub

[Cache credentials](#)

[HTTPS or SSH?](#)

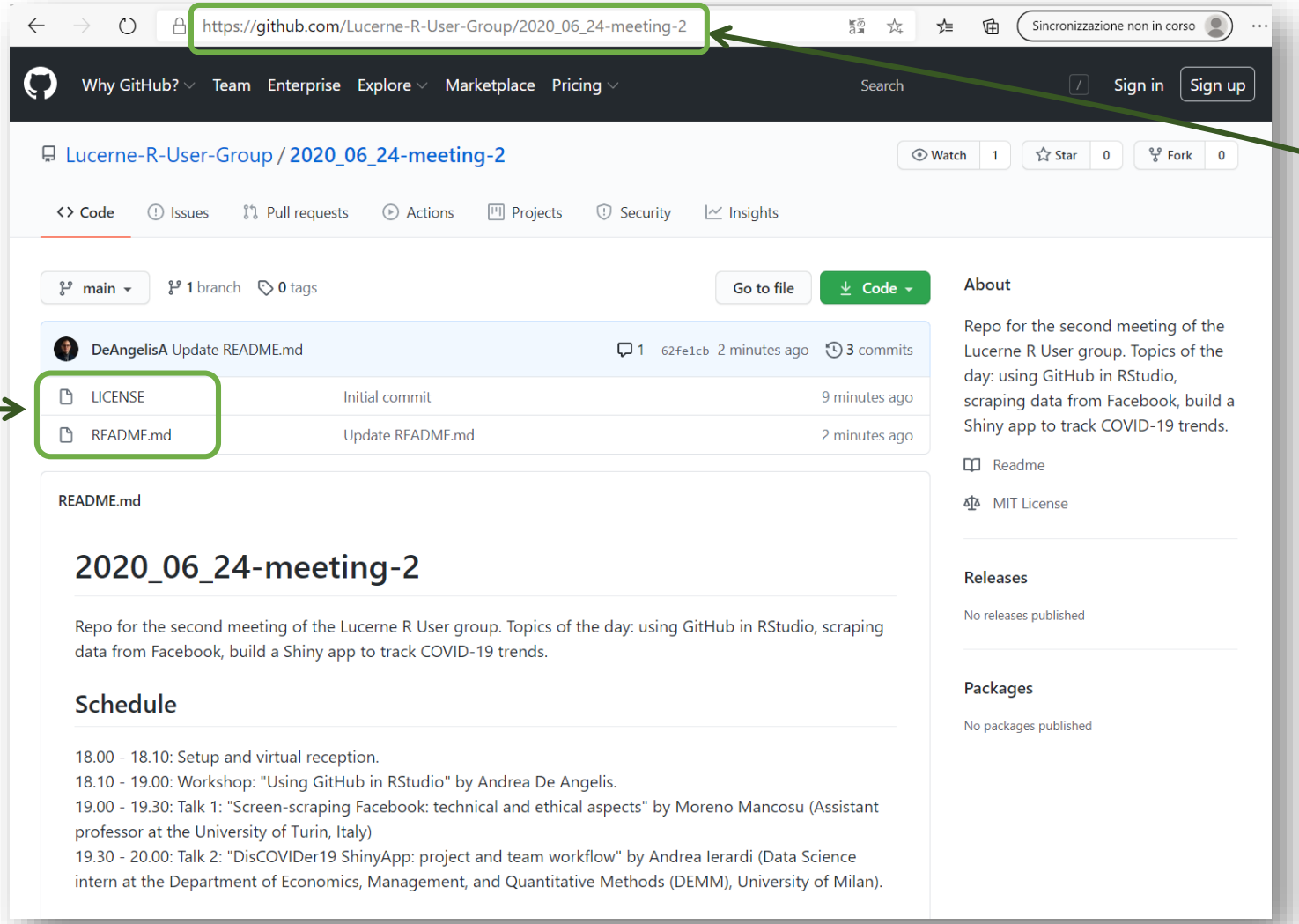
Initialize a GitHub "repo"



GitHub

Initialize a GitHub "repo"

Uploaded folders, files and scripts



Repo URL

GitHub

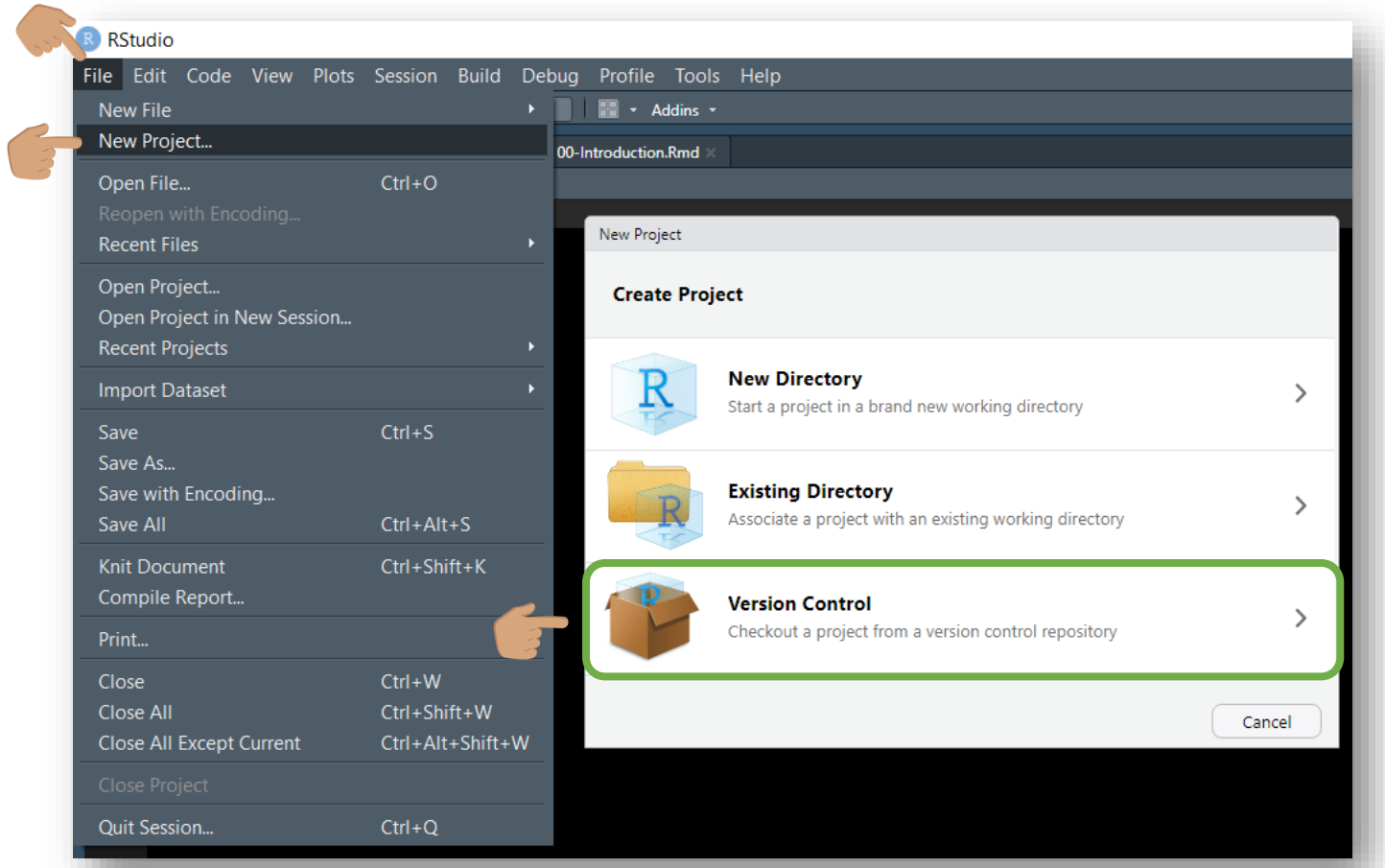
Your turn!

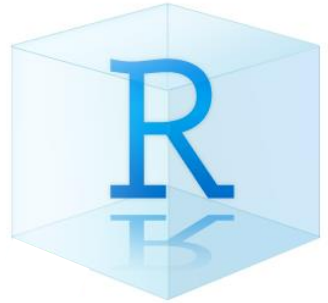
Initialize your first GitHub repo!

Call it: "git-sandbox"

02:00

GitHub & RStudio Projects



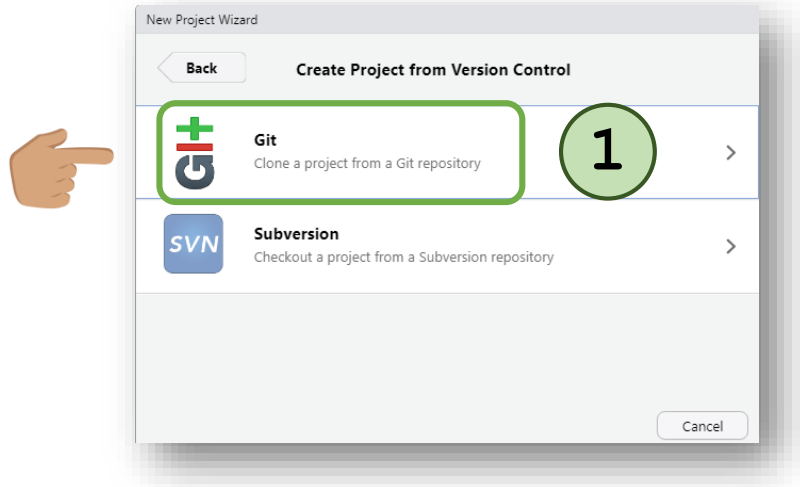


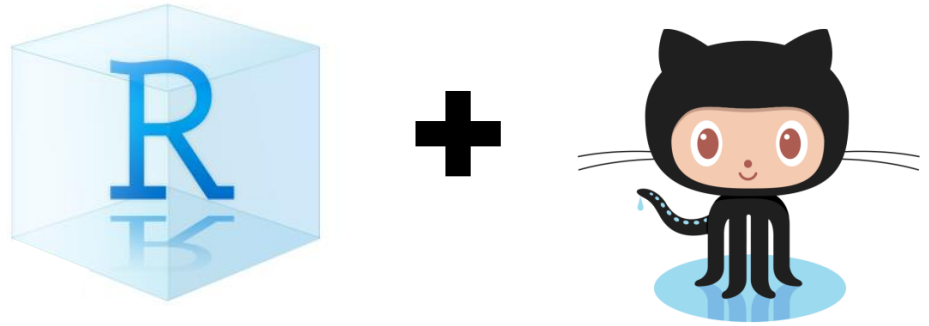
+

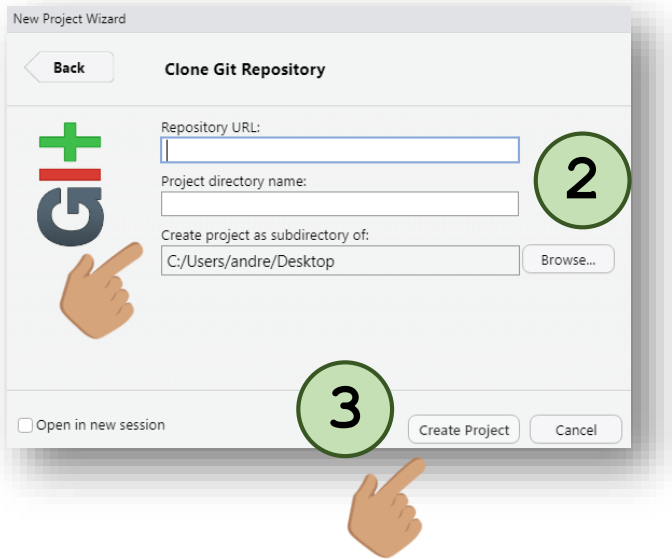
RStudio projects...

...on GitHub!

GitHub & RStudio Projects



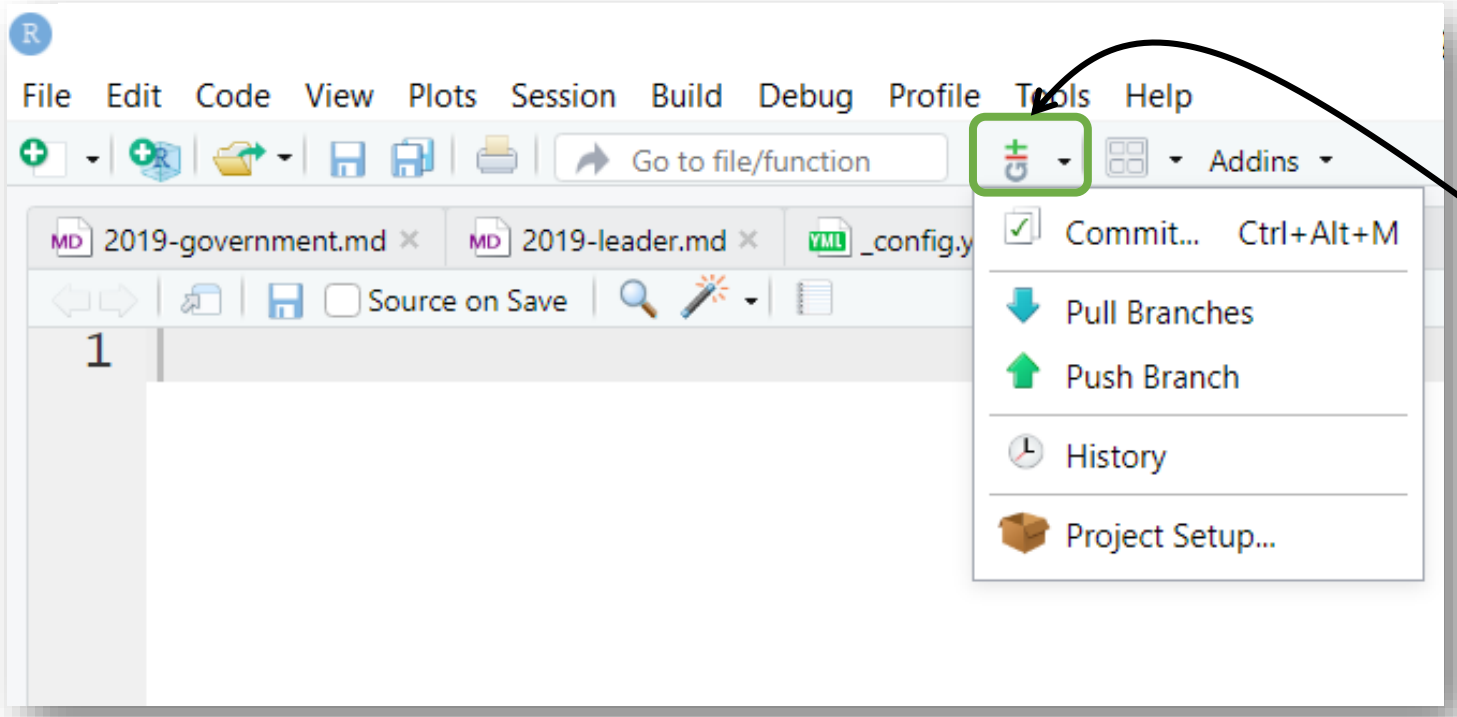




Three steps:

1. Choose Git
2. Paste repo URL, give a name, select local folder
3. Create Project

GitHub & RStudio Projects



+

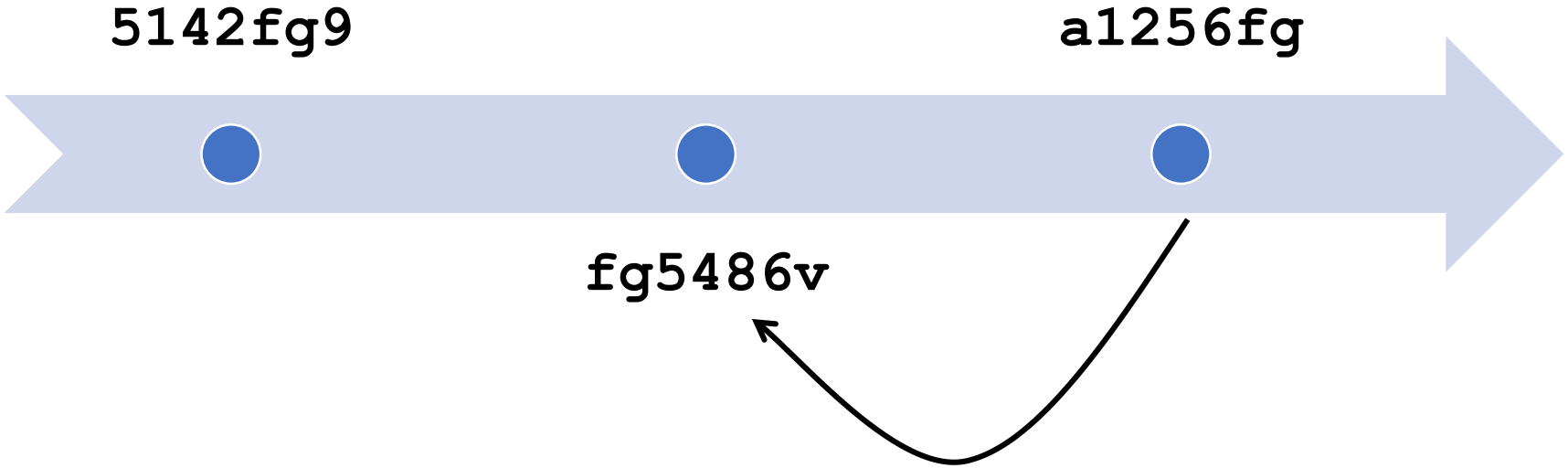
RStudio's Git toolbar is active

Your turn!

Git that toolbar!

02:00

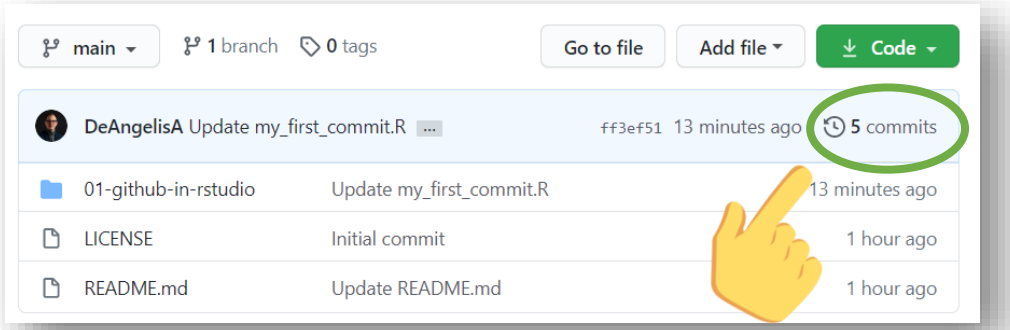
Commits

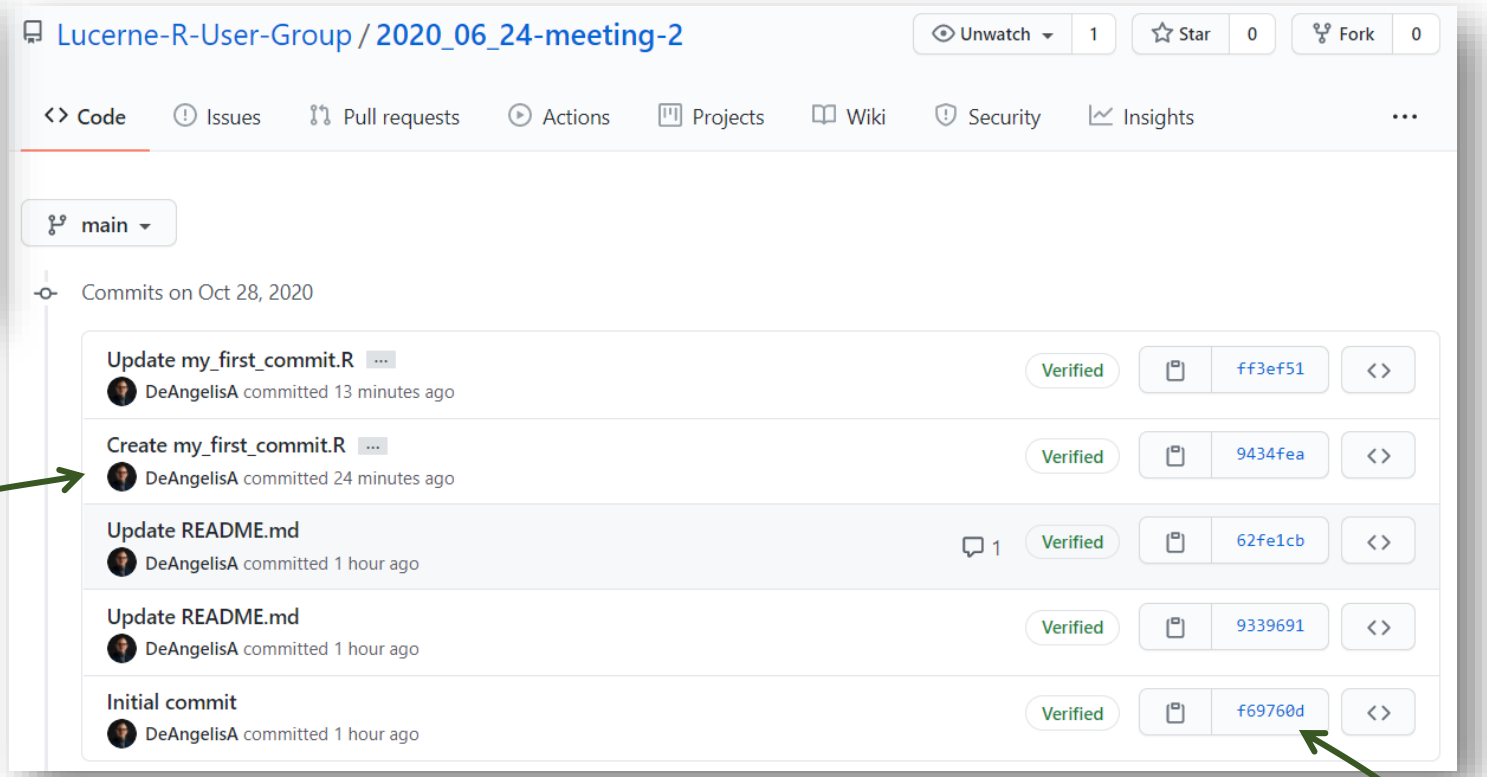


Commit: a snapshot of the whole project

{you can undo mistakes}

Commits in a repo





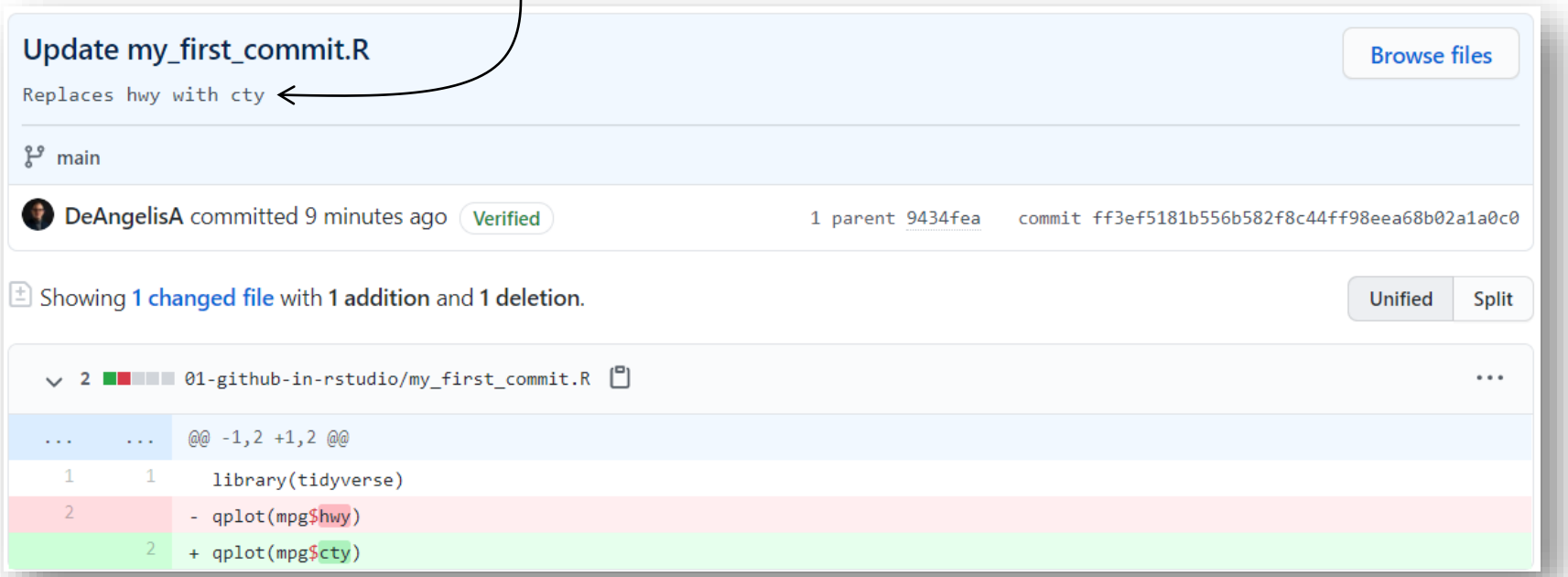
Five commits in this repo

Commit id (SHA)

One commit

A commit message is required
"Replace A with B" "Fix issue2 scraper bug"

Deletions **Additions**



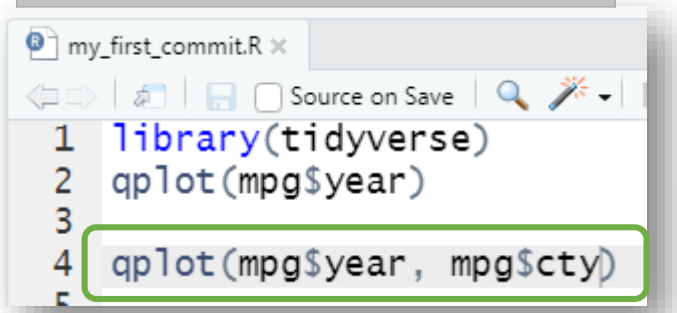
Commit id (SHA):
`ff3ef5181b556b582f8c44ff98eea68b02a1a0c0`

Link to commit:
https://github.com/Lucerne-R-User-Group/2020_06_24-meeting-2/commit/ff3ef5181b556b582f8c44ff98eea68b02a1a0c0#diff-301cb80351ee6dce13646cf3a5dd2a36f2e925ad6788aa66403b930bf1a96f29

How to commit

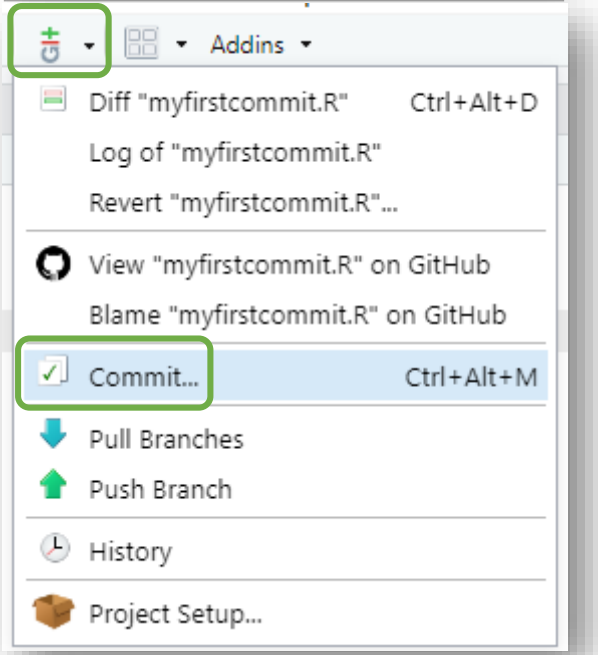
Make your changes and save

1

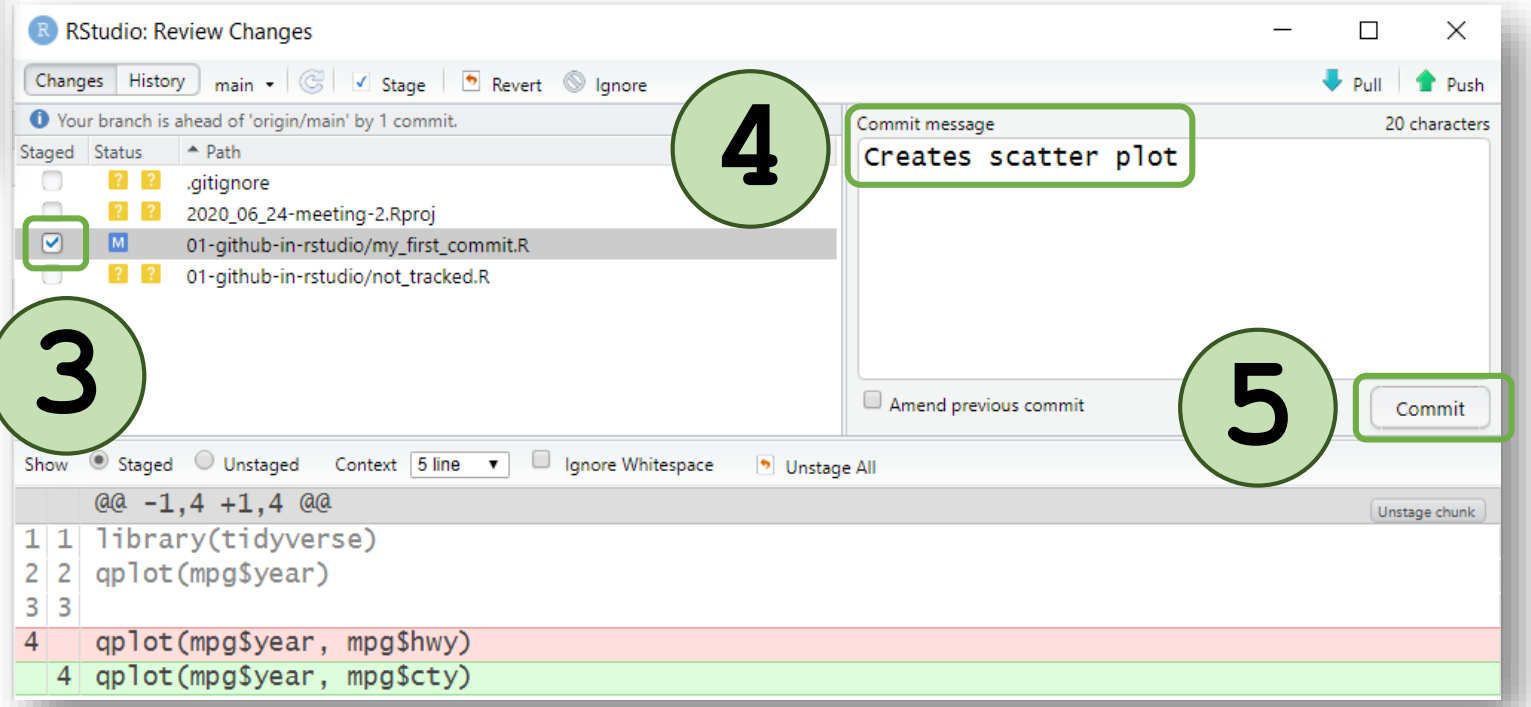


Click **`commit`** in the git bar

2



Stage changes, add a message and commit



3

4

5

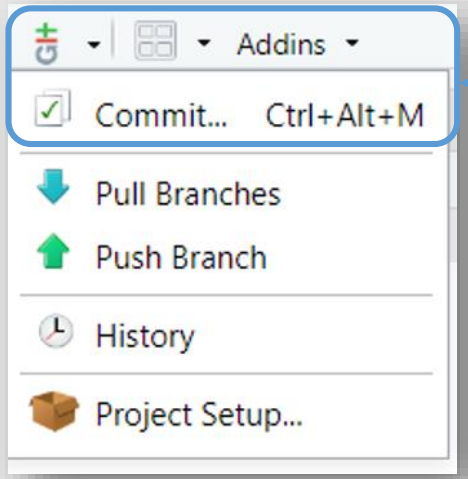
Your turn!

Git your first commit!

1. Create a new R script
2. Type in something
3. Save the R script
4. Git **commit**!

my_first_commit.R

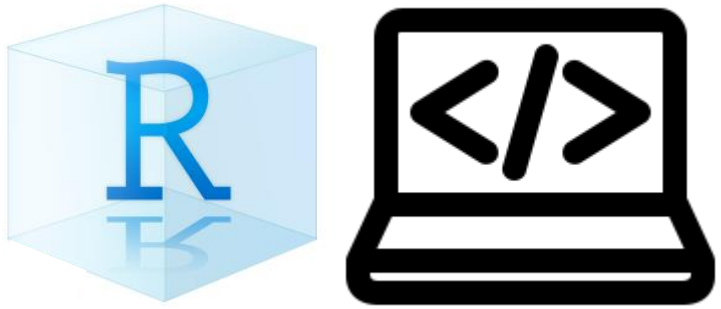
```
# E.g.:
library(tidyverse)
qplot(mpg$hwy)
```



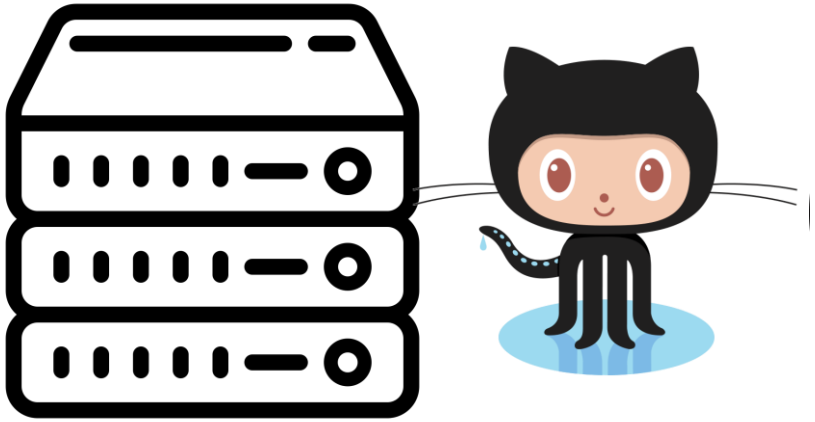
02:00

Git workflow

Basic workflow

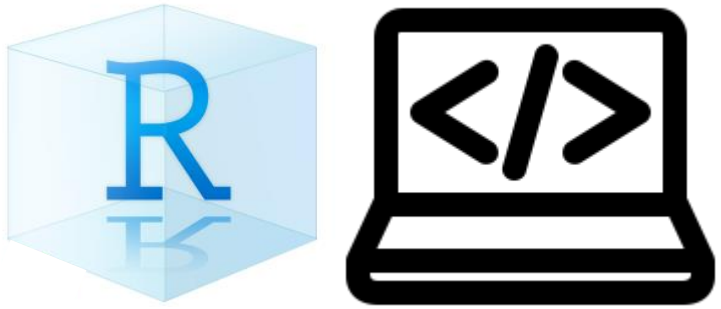


You working locally



remote repo hosted on github

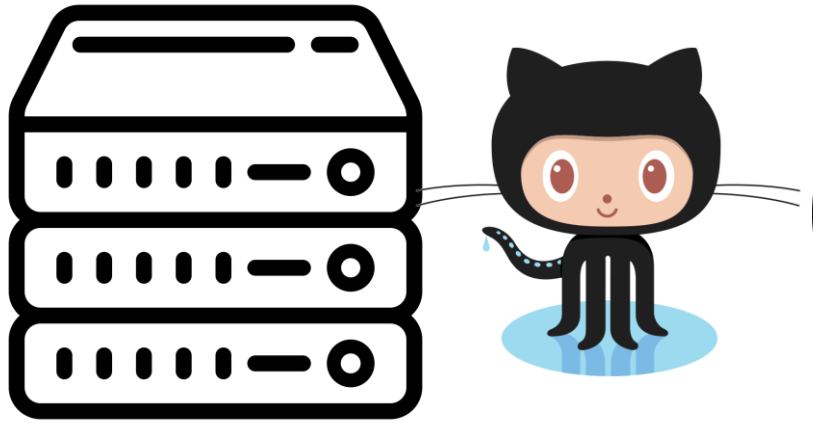
Basic workflow



You working locally

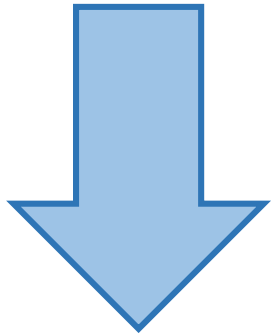
How to get changes from collaborators

?



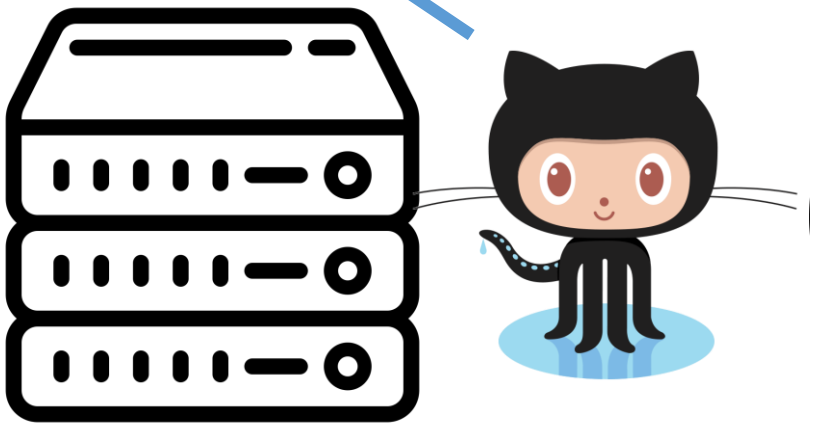
remote repo hosted on github

Basic workflow

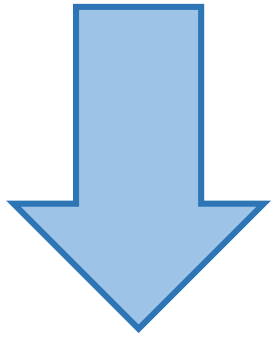


pull

Local project gets updates from the remote repo

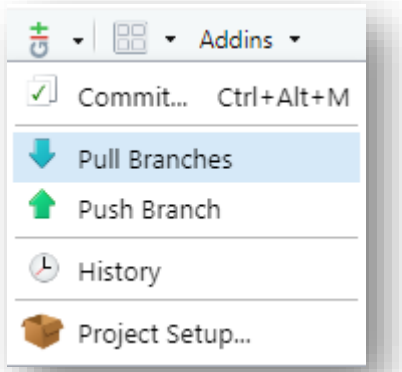


Basic workflow



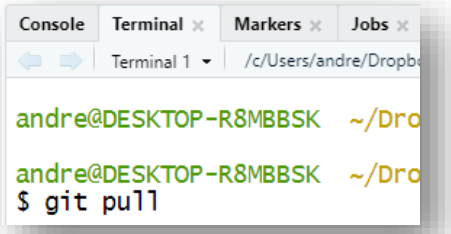
pull

In RStudio



In terminal

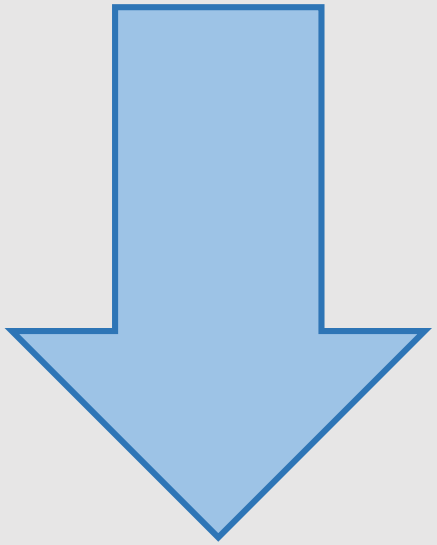
```
> git pull
```



Your turn!

Pull a change

1. First **commit a change** to you script from GitHub.com

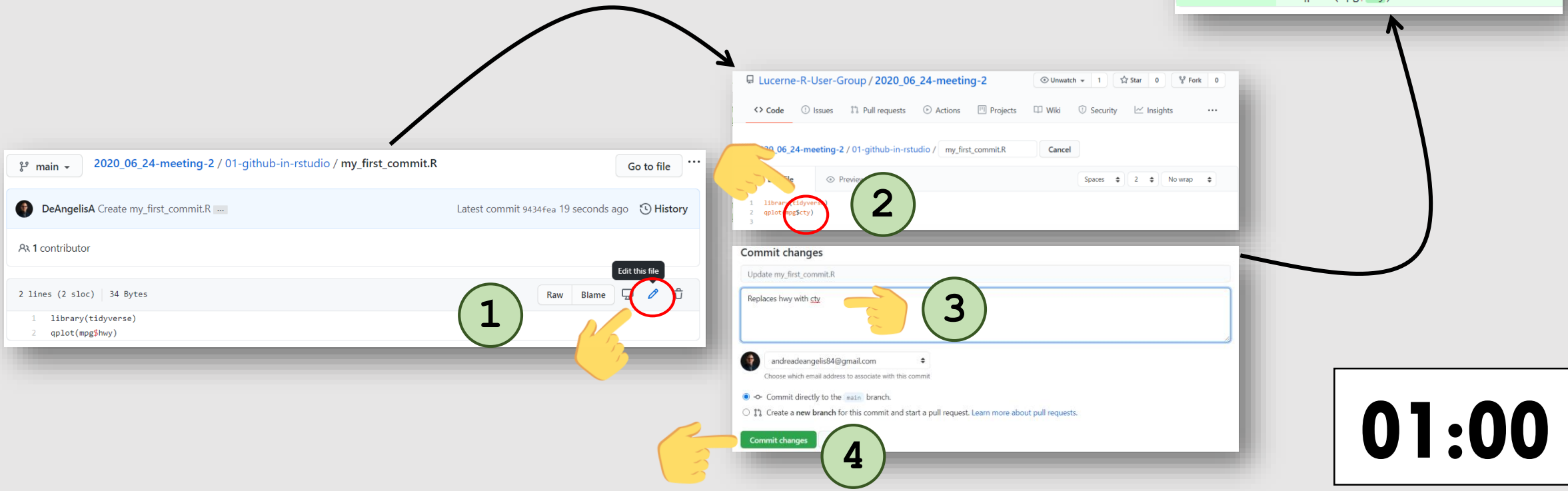


01:00

Your turn!

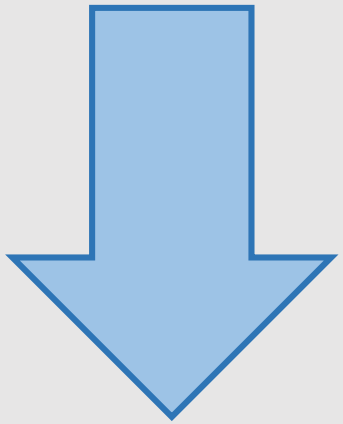
Pull a change

1. First **commit a change** to you script from GitHub.com



01:00

Your turn!

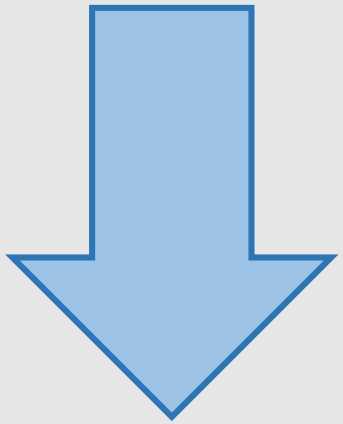


Pull a change

1. First commit a change to you script from GitHub.com
2. **Update the GitHub page** of the repo to note that the GitHub version is now *ahead* of your local project by one commit

00:30

Your turn!

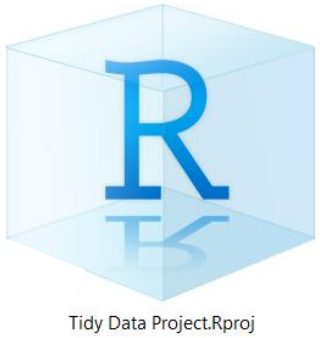


Pull a change

1. First commit a change to you script from GitHub.com
2. Update the GitHub page of the repo to note that the GitHub version is now *ahead* of your local project by one commit
3. **Pull the commit** and… check the script!

00:30

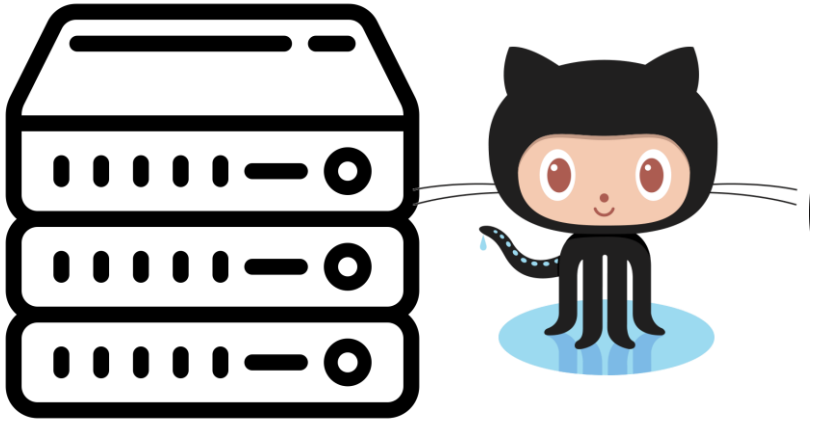
Basic workflow



You working locally

How to <u>send</u> changes to collaborators (or future you)

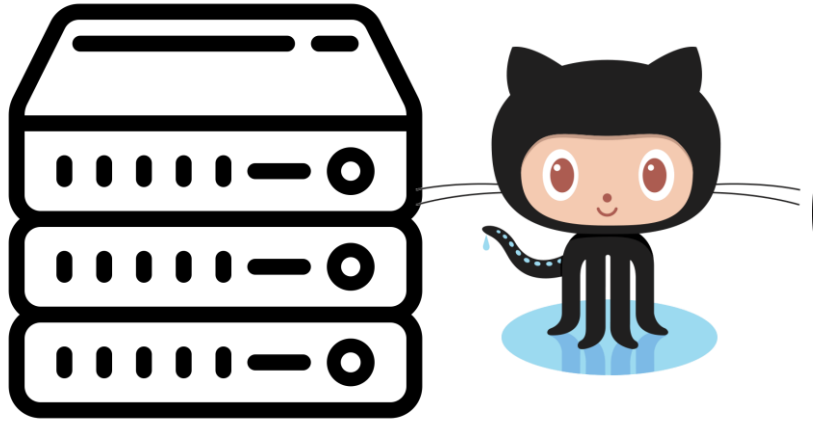
?



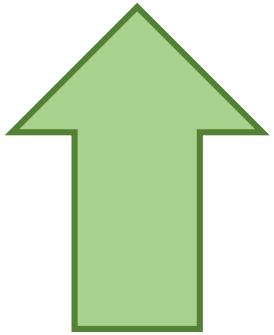
remote repo hosted on github

Basic workflow

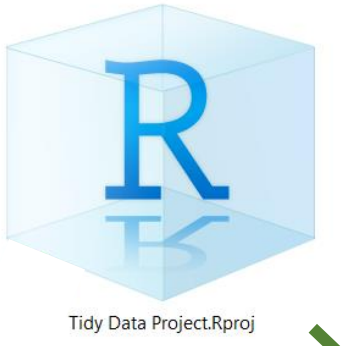
remote repo gets updated with local commits



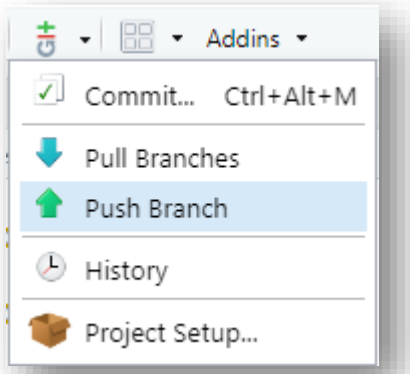
push



Basic workflow

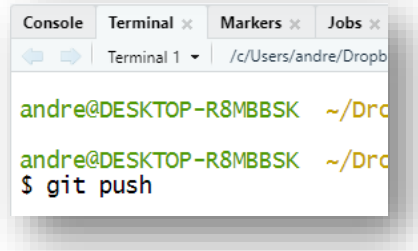


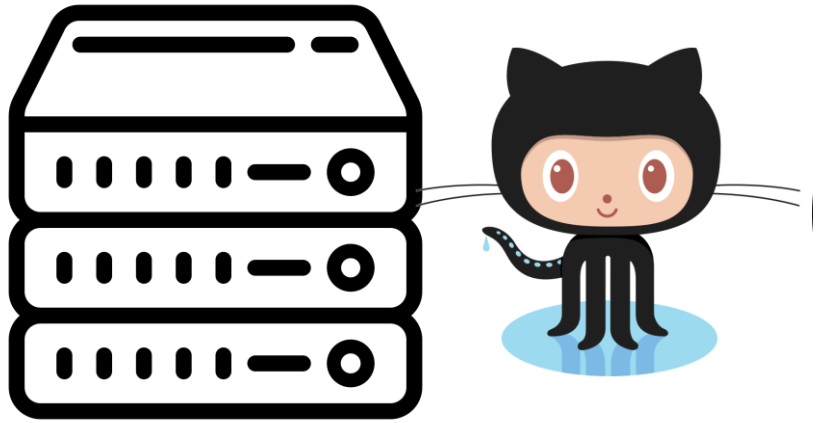
In RStudio



In terminal

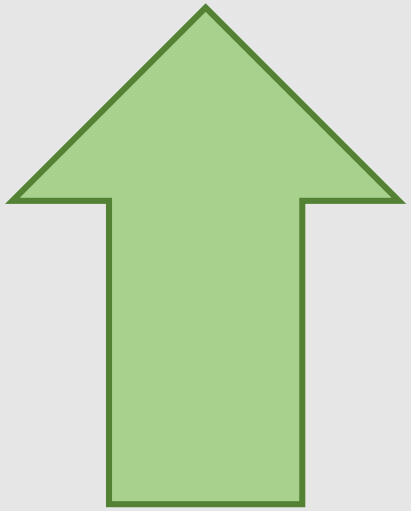
```
> git push
```





push

Your turn!



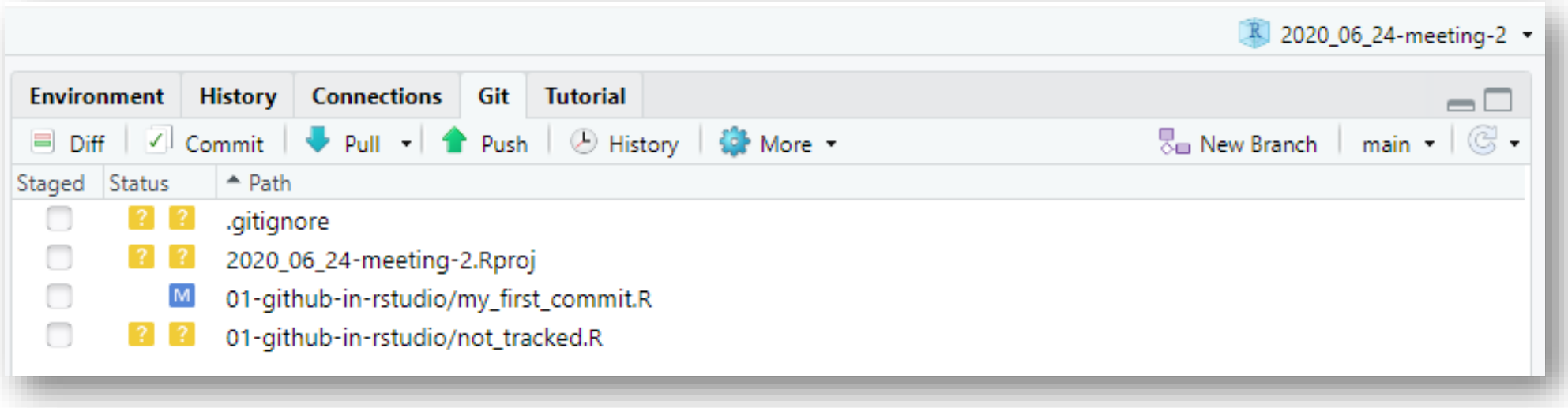
Push a change

1. **Commit** a change in your script.
2. **Push the commit** and… check the repo on github.com!

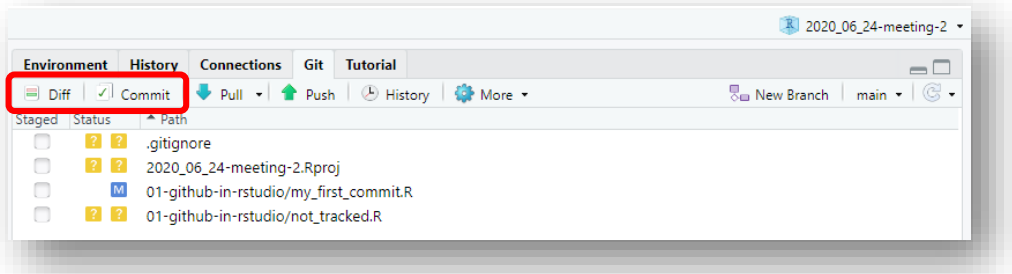
00:30

RStudio git tab

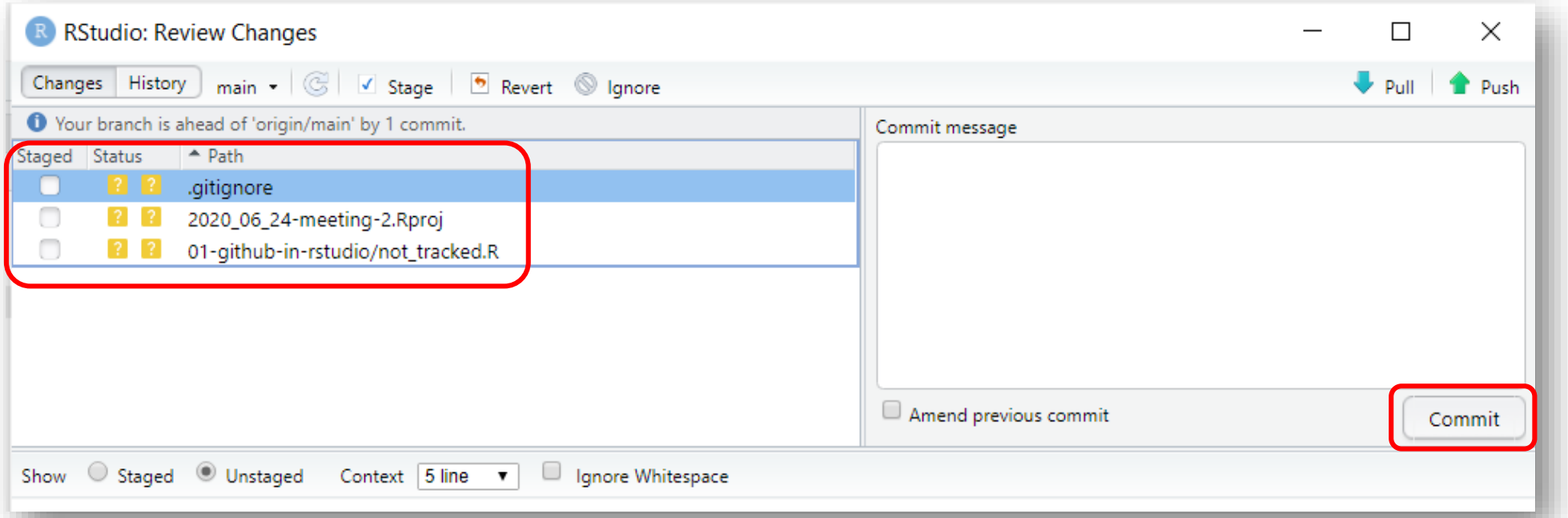
Interface to control RStudio / Git interaction



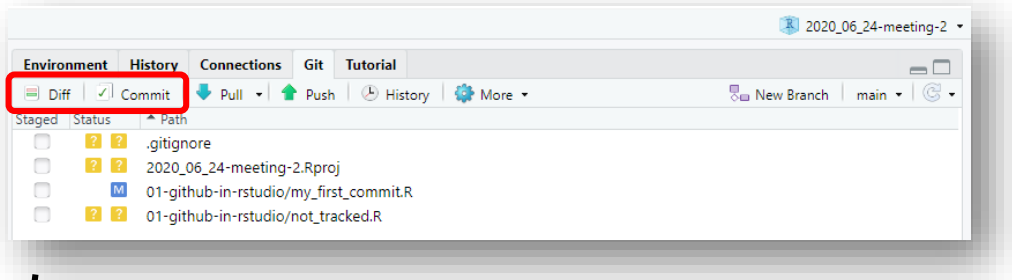
Commit tab



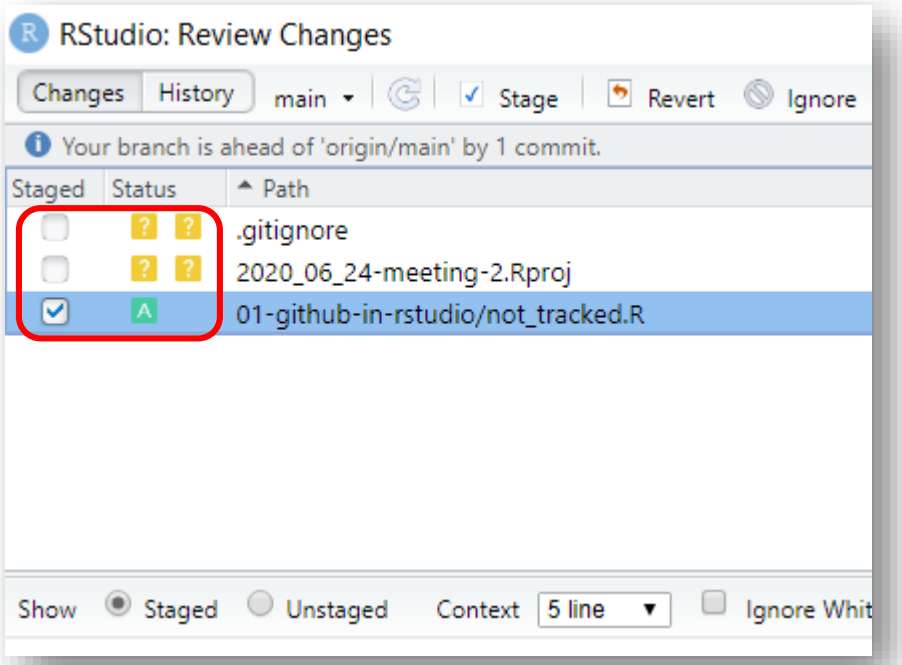
Review status of files in the project
Stage & commit



Commit tab



Review status of files in the project
Stage & commit



File statuses

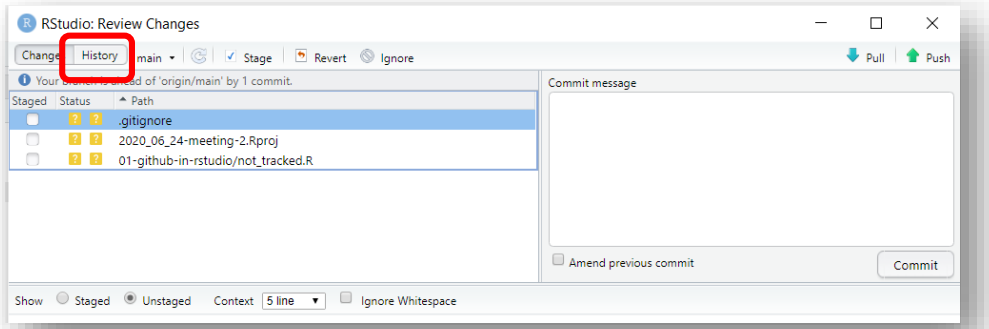
Untracked: changes won't be included in the next commit →

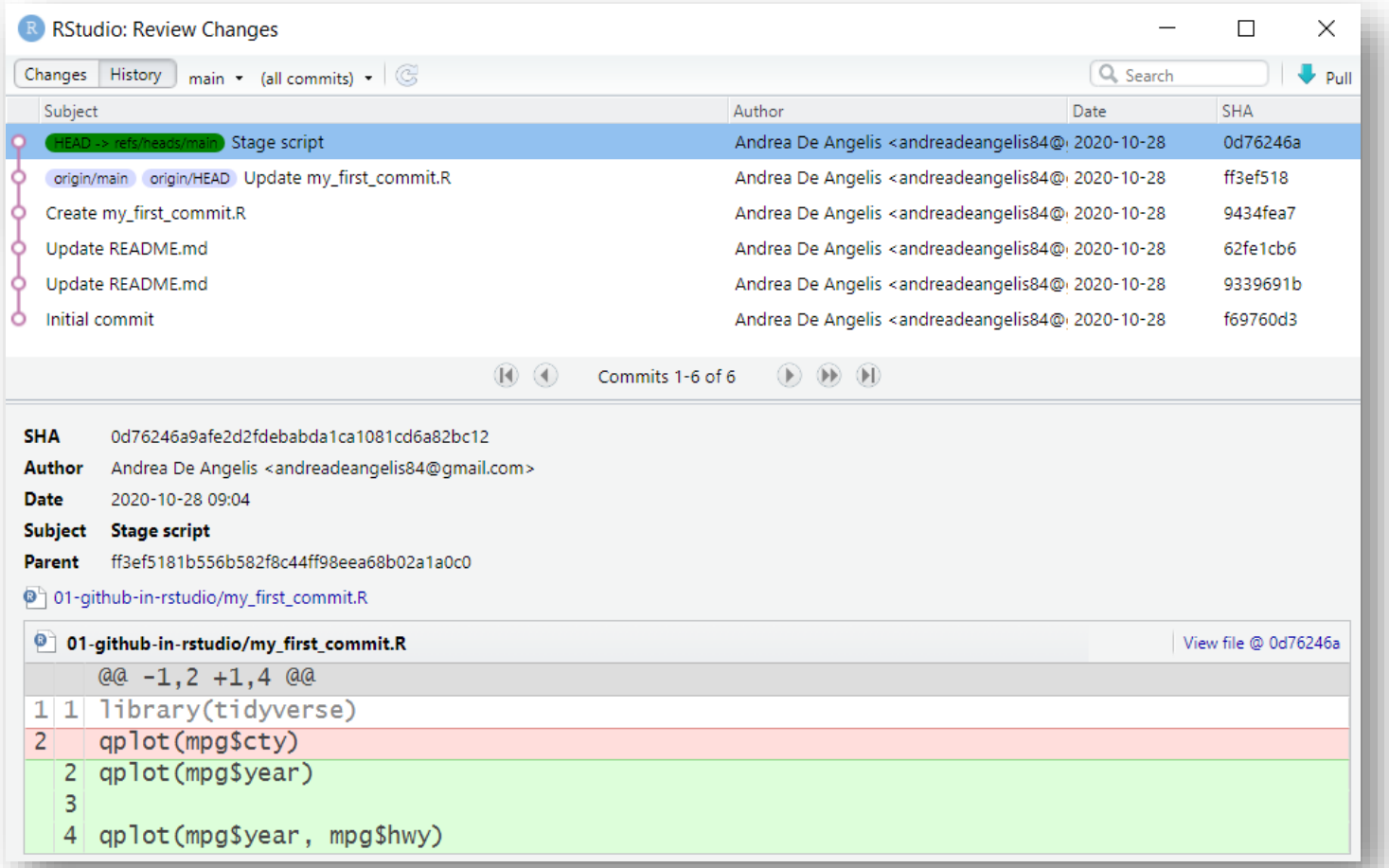
Staged: changes ready for next commit →

Modified: file is tracked and was locally changed →

Command **`git add`** adds changes to the **staging** area.
In RStudio we can git add to the staging area by clicking the "Staged" checkbox.

History tab


RStudio: Review Changes
Changes
History
main
Stage
Revert
Ignore
Pull
Push
Staged
Status
Path
.gitignore
2020_06_24-meeting-2.Rproj
01-github-in-rstudio/not_tracked.R
Commit message
Amend previous commit
Commit
Show
Staged
Unstaged
Context
5 line
Ignore Whitespace

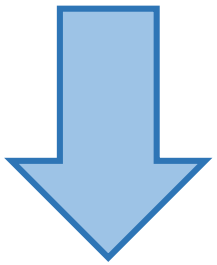

RStudio: Review Changes
Changes
History
main
(all commits)
Search
Pull
Subject
Author
Date
SHA
HEAD -> refs/heads/main Stage script
Andrea De Angelis <andreadeangelis84@ 2020-10-28 0d76246a
origin/main origin/HEAD Update my_first_commit.R
Andrea De Angelis <andreadeangelis84@ 2020-10-28 ff3ef518
Create my_first_commit.R
Andrea De Angelis <andreadeangelis84@ 2020-10-28 9434fea7
Update README.md
Andrea De Angelis <andreadeangelis84@ 2020-10-28 62fe1cb6
Update README.md
Andrea De Angelis <andreadeangelis84@ 2020-10-28 9339691b
Initial commit
Andrea De Angelis <andreadeangelis84@ 2020-10-28 f69760d3
Commits 1-6 of 6
SHA 0d76246a9afe2d2fdebabda1ca1081cd6a82bc12
Author Andrea De Angelis <andreadeangelis84@gmail.com>
Date 2020-10-28 09:04
Subject Stage script
Parent ff3ef5181b556b582f8c44ff98eea68b02a1a0c0
01-github-in-rstudio/my_first_commit.R
01-github-in-rstudio/my_first_commit.R
View file @ 0d76246a
@@ -1,2 +1,4 @@
1 1 library(tidyverse)
2 qplot(mpg$cty)
2 qplot(mpg$year)
3
4 qplot(mpg$year, mpg$hwy)

Basic team workflow

1

2

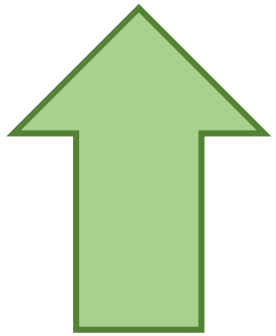
3



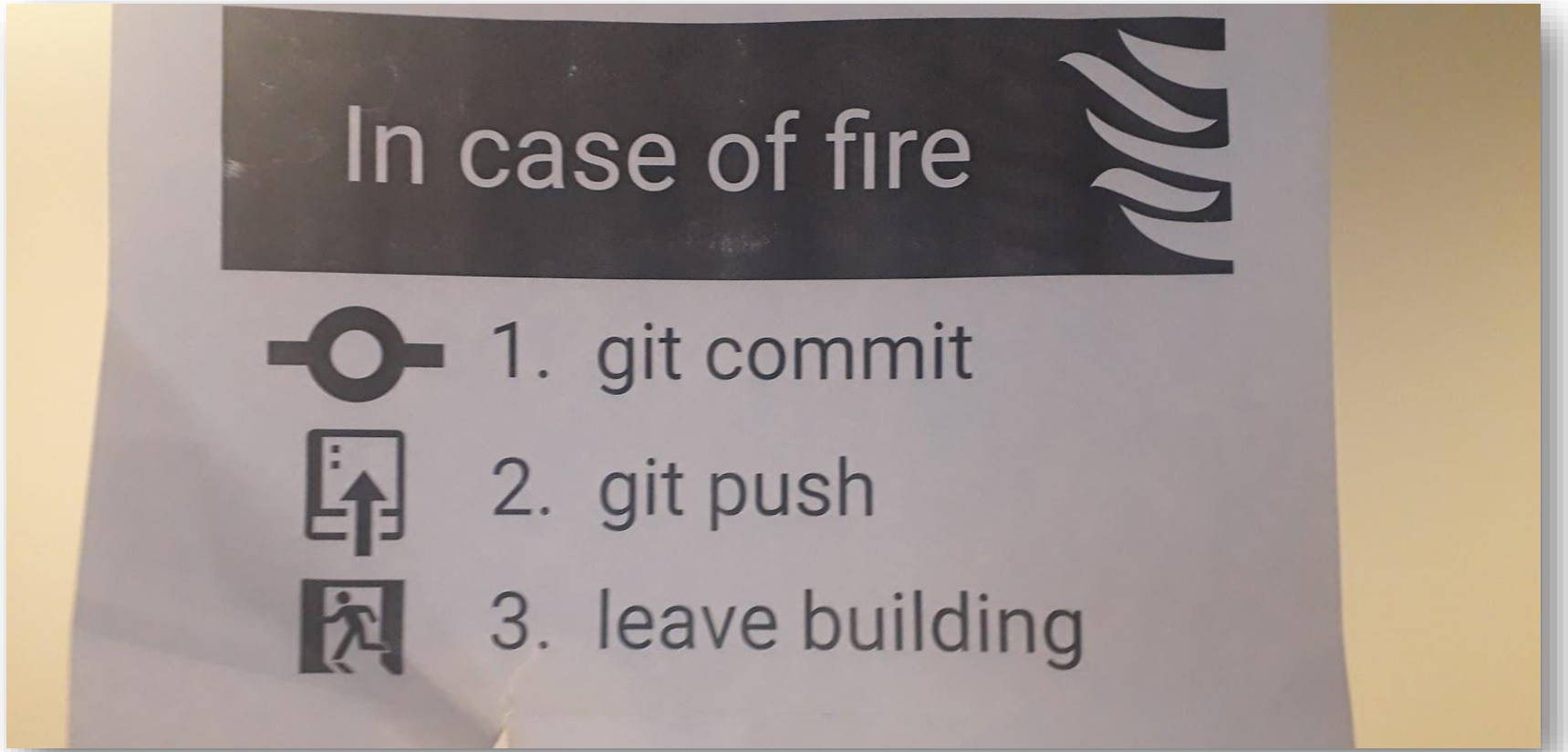
Pull before starting and once in a while

```
git add
git commit
```

Repeat



Push every few commits


In case of fire
1. git commit
2. git push
3. leave building

Test!

Your Turn

Why should you use a VC system?

A. To simplify the workflow

B. To manage many files

C. To collaborate

D. To track & undo every change

00:15

Your Turn

What should you do as you start a new working session on GitHub? Assume you work in a team.

A	I push
B	I pull
C	I start working and do commits
D	I check the status and then pull

00:15

Your Turn

Do you always need to pull before start working?

TRUE

FALSE

00:15

Your Turn

Which is a good-enough GitHub workflow?

A `pull, commit, push`

B `pull, commit, commit, commit, push`

C `pull, push, commit`

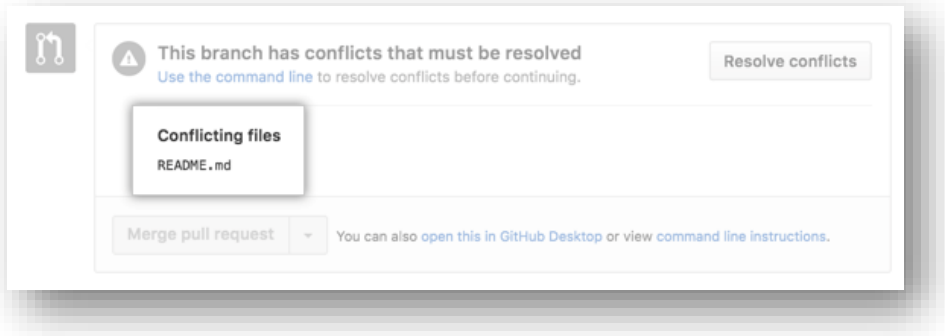
D `push, pull, commit`

00:15

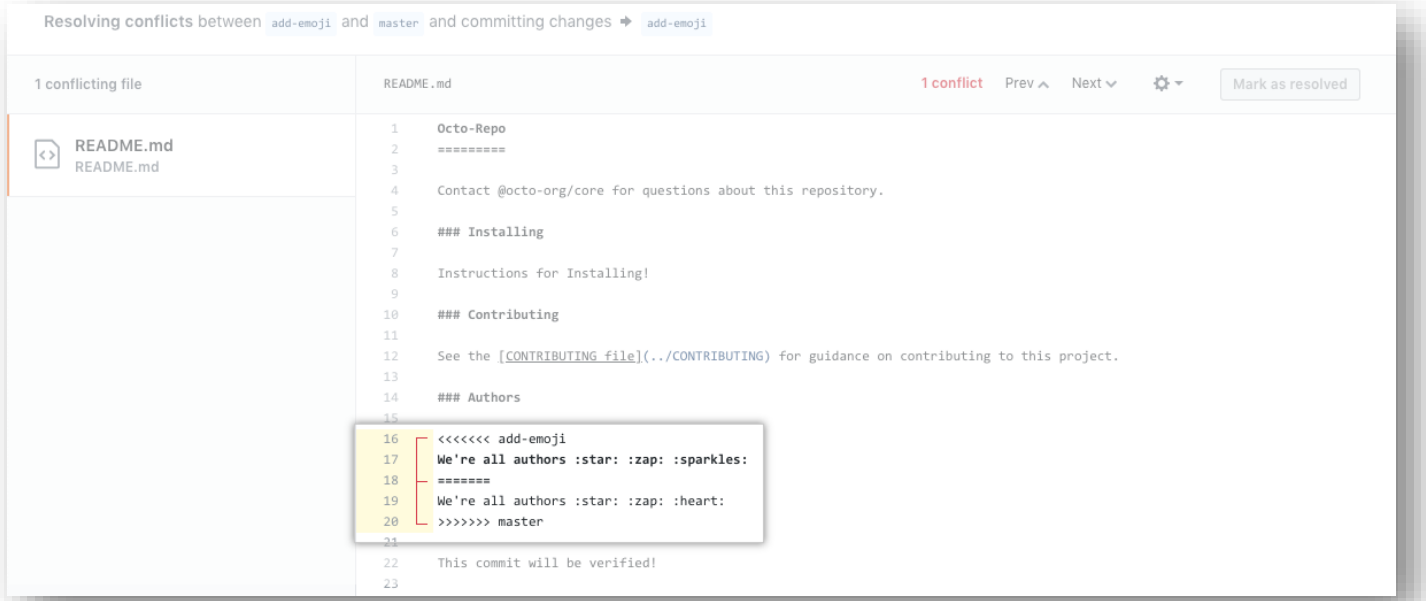
Questions?

"But what if two developers change the same line in the same script in two different ways?"

Merge conflicts



Ref: 22.4 Dealing with Conflicts.
https://happygitwithr.com/git-branches.html

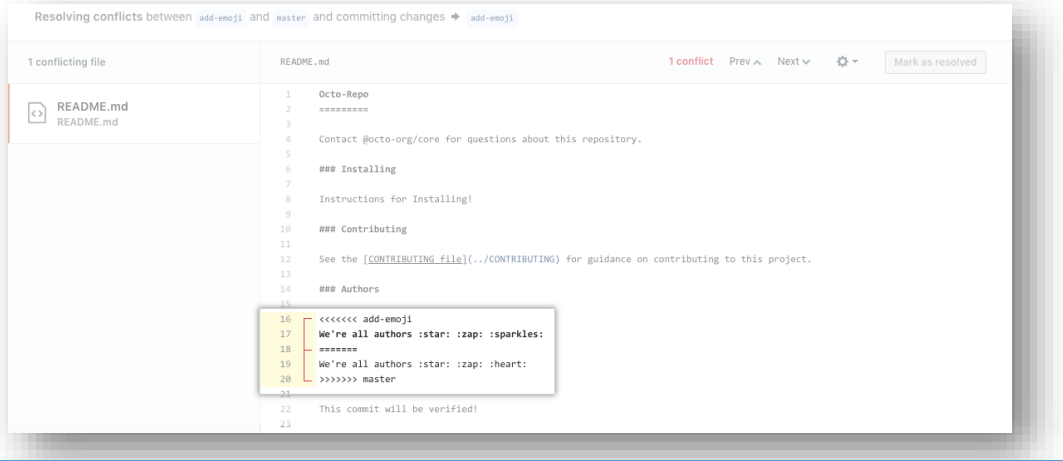


https://help.github.com/en/github/collaborating-with-issues-and-pull-requests/resolving-a-merge-conflict-on-github

Solve conflicts

Just manually delete the bad version and the >>>> and <<<< lines. E.g., leave just:

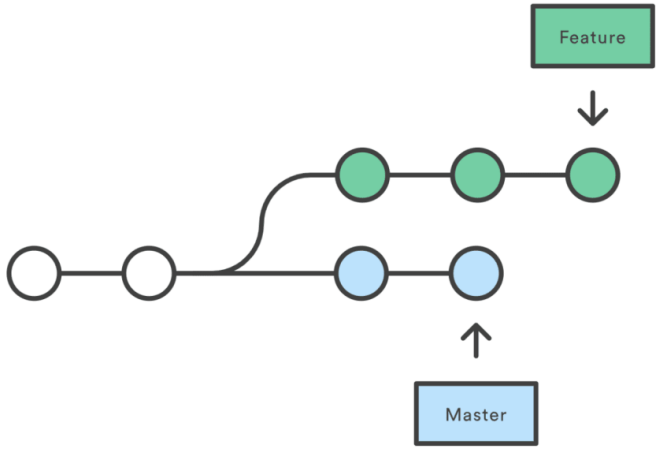
`We're all authors :star: :zap: :sparkles:`



https://help.github.com/en/github/collaborating-with-issues-and-pull-requests/resolving-a-merge-conflict-on-github

Next steps

Use branches to safely experiment new features

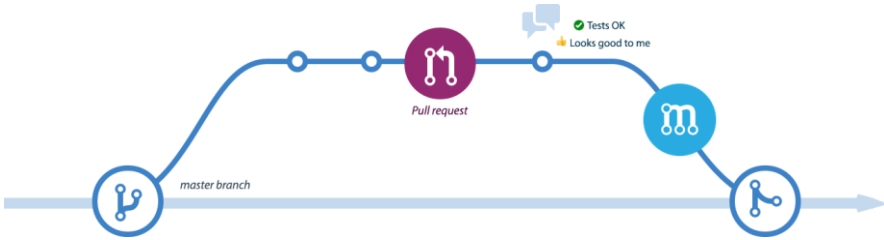


Use issues for effective collaboration

Link: master issues in10-min

- Labels
- Assignees
- Reviewers
- @mentions
- [ ] [X] TODO list
- #issue-references

Use pull requests to disacuss changes



Useful links

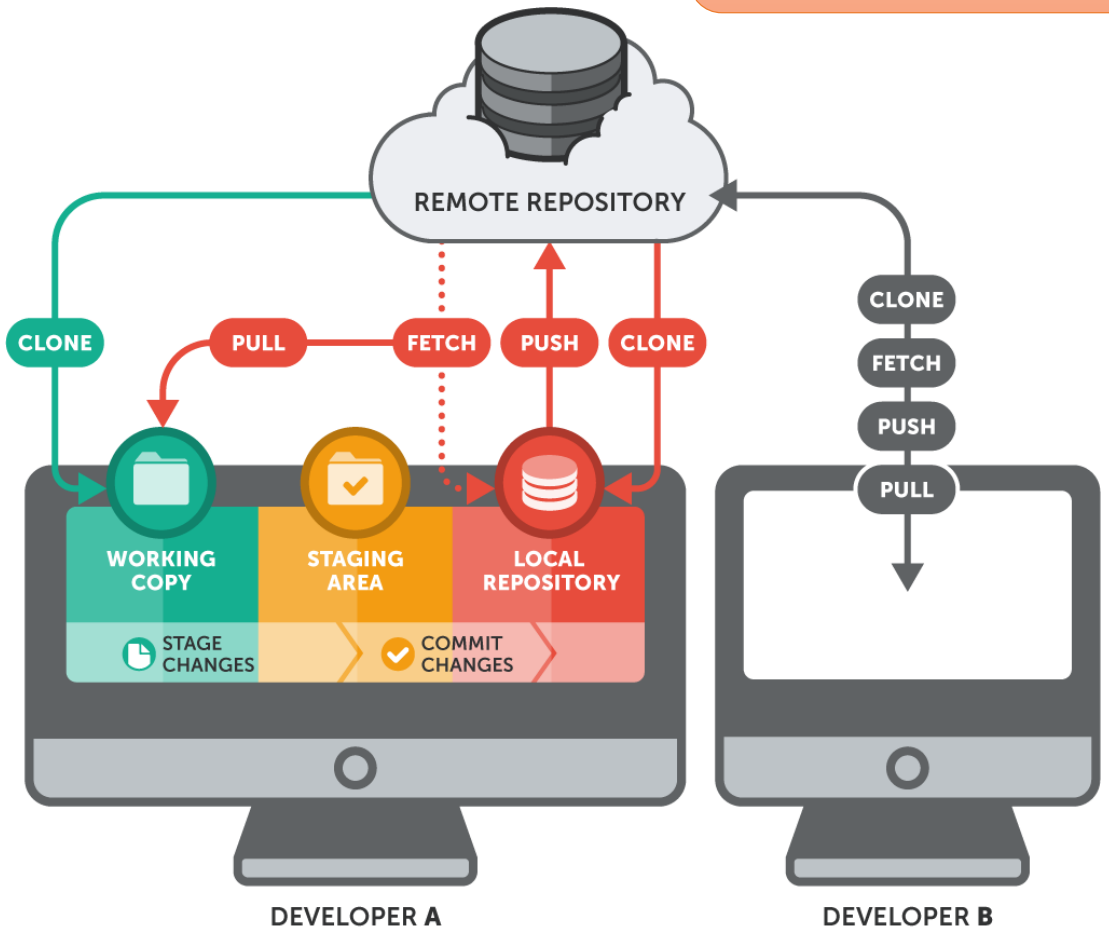
Workflow

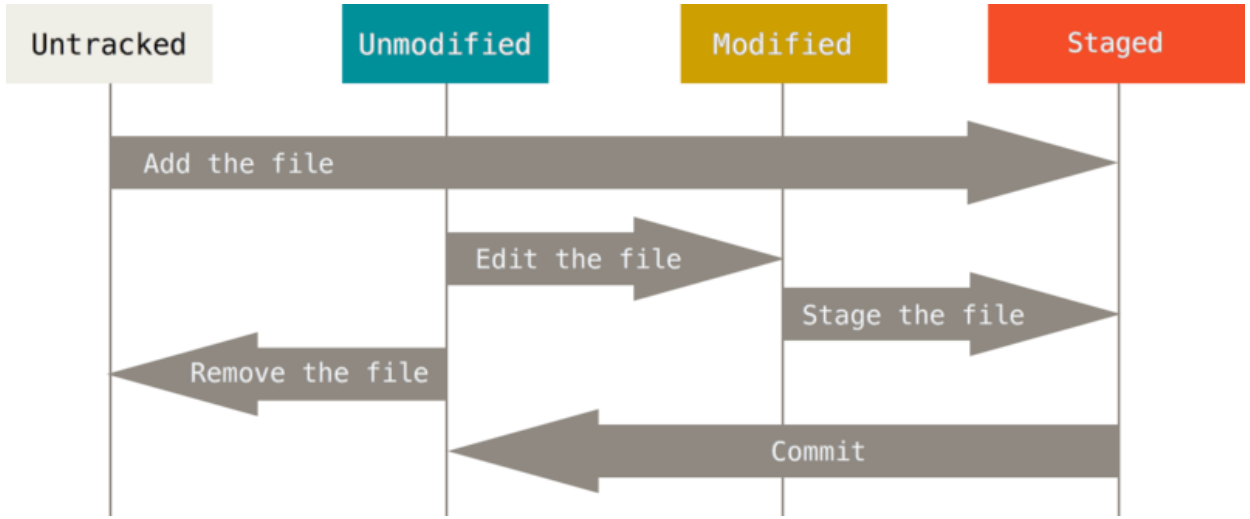
Branching

Pull requests

Manage merge conflicts

Useful links





Main git commands explained

Main references

Git and GitHub chapter R Packages book by H. Wickham	Short and to the point
Happy Git and GitHub for the useR by Jenny Bryan	A whole book about git for RStudio users
GitHub Learning Lab	A GitHub course… on GitHub!
Git Handbook	Git started in one hour

LuzeRn
Lucerne R User Group

Our sponsors

R consortium

R Studio®

UNIVERSITÄT LUZERN

Lucerne Master in Computational Social Science (LUMACSS)

Campus Lucerne

KSF Graduate School of Lucerne

Dr Andrea De Angelis

28 October 2020

https://www.meetup.com/Lucerne-R-User-Group/

https://github.com/Lucerne-R-User-Group/2020_06_24-meeting-2