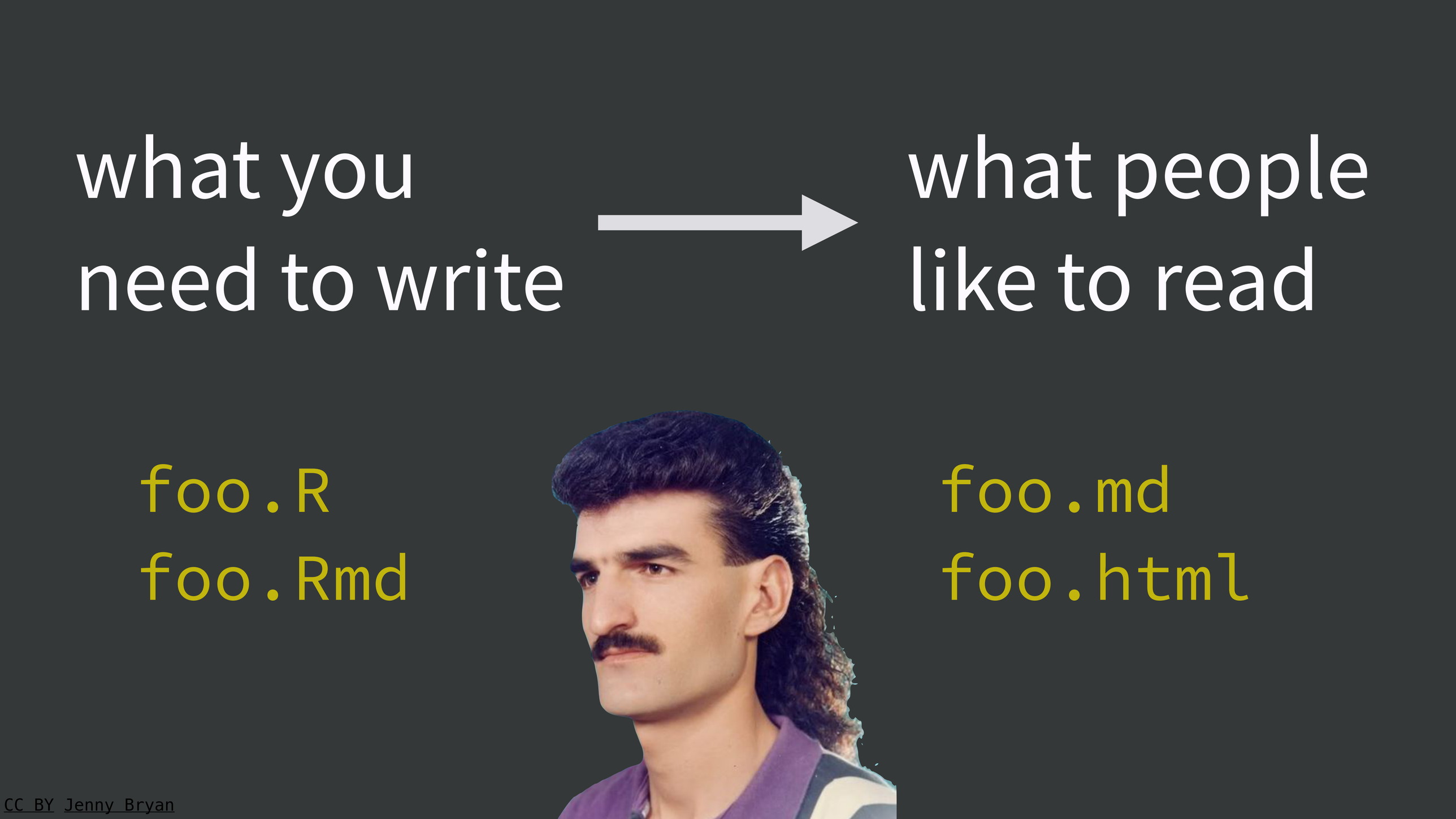
More git and Github

1. Accept the organization invitation for STAT360 in your email

what you
need to write



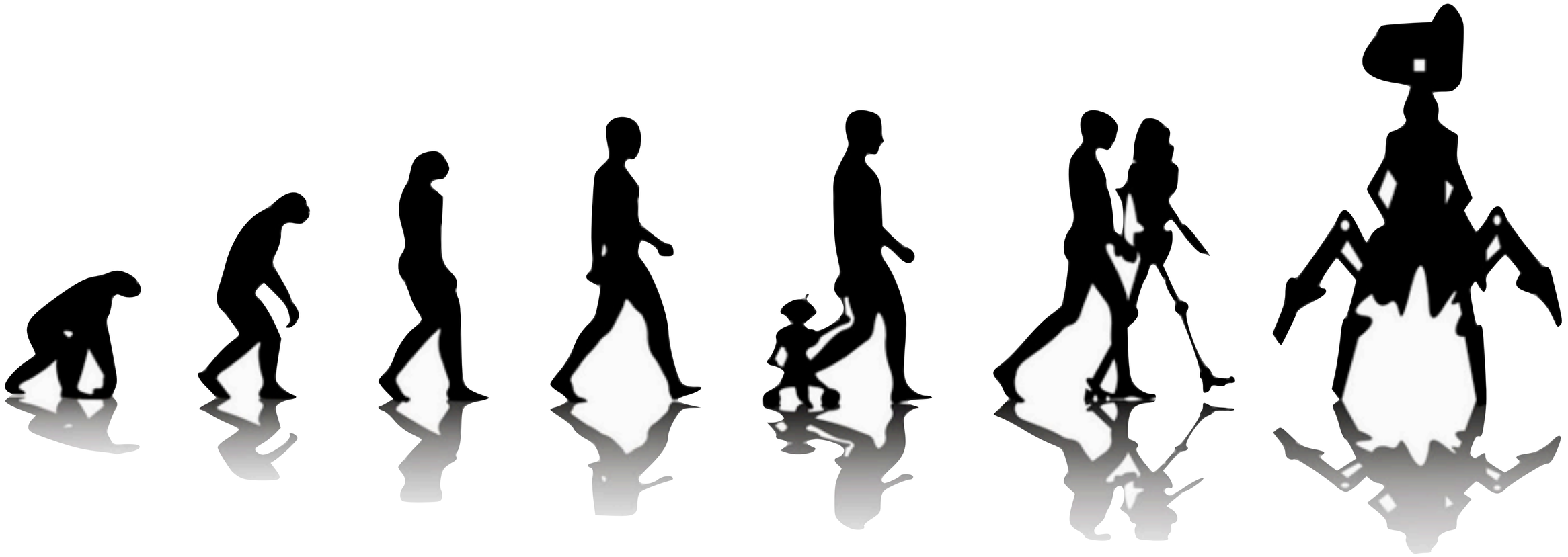
what people
like to read

`foo.R`
`foo.Rmd`

`foo.md`
`foo.html`

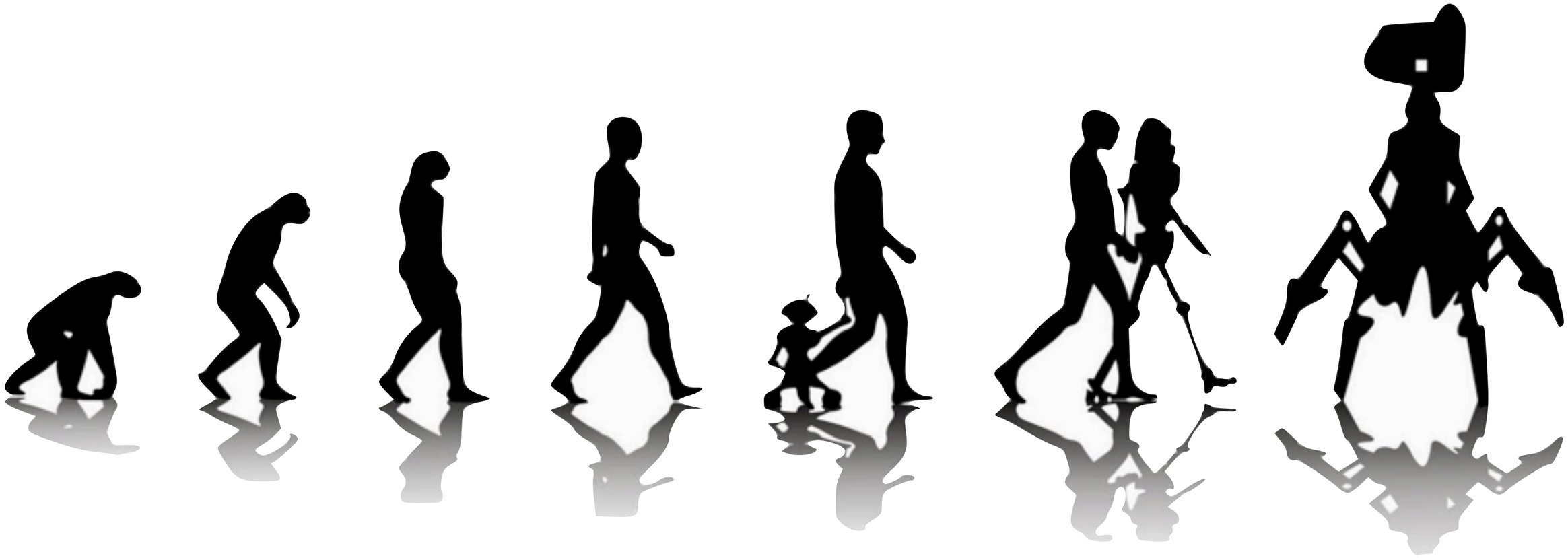
CC BY Jenny Bryan

CC BY Jenny Bryan



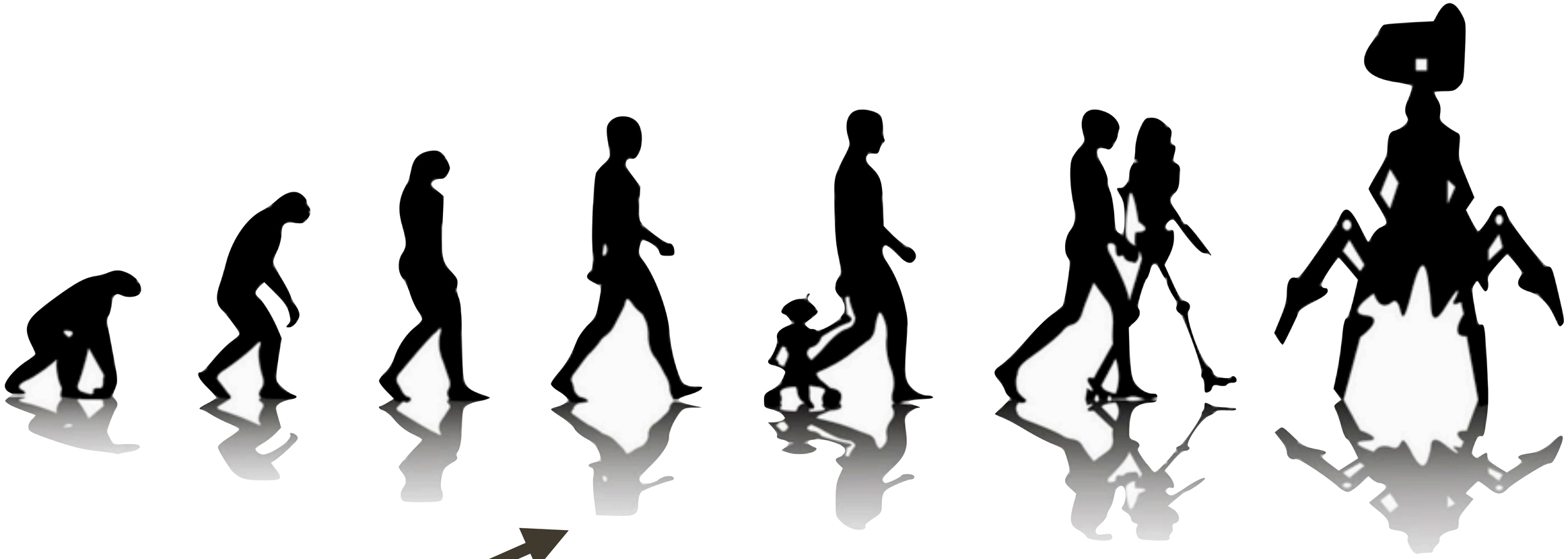
use version control

CC BY Jenny Bryan



we teach Git + GitHub

CC BY Jenny Bryan

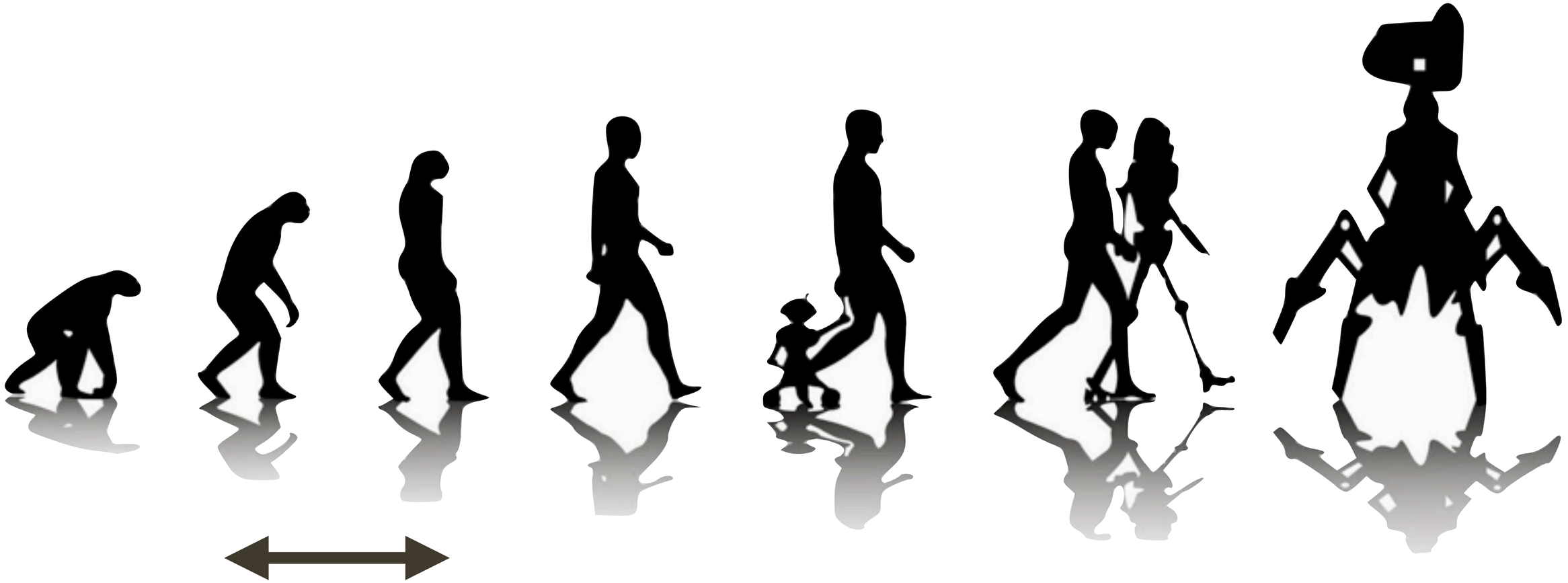


"commit"

a file or project state that is **meaningful to you**

for inspection, comparison, restoration

CC BY Jenny Bryan



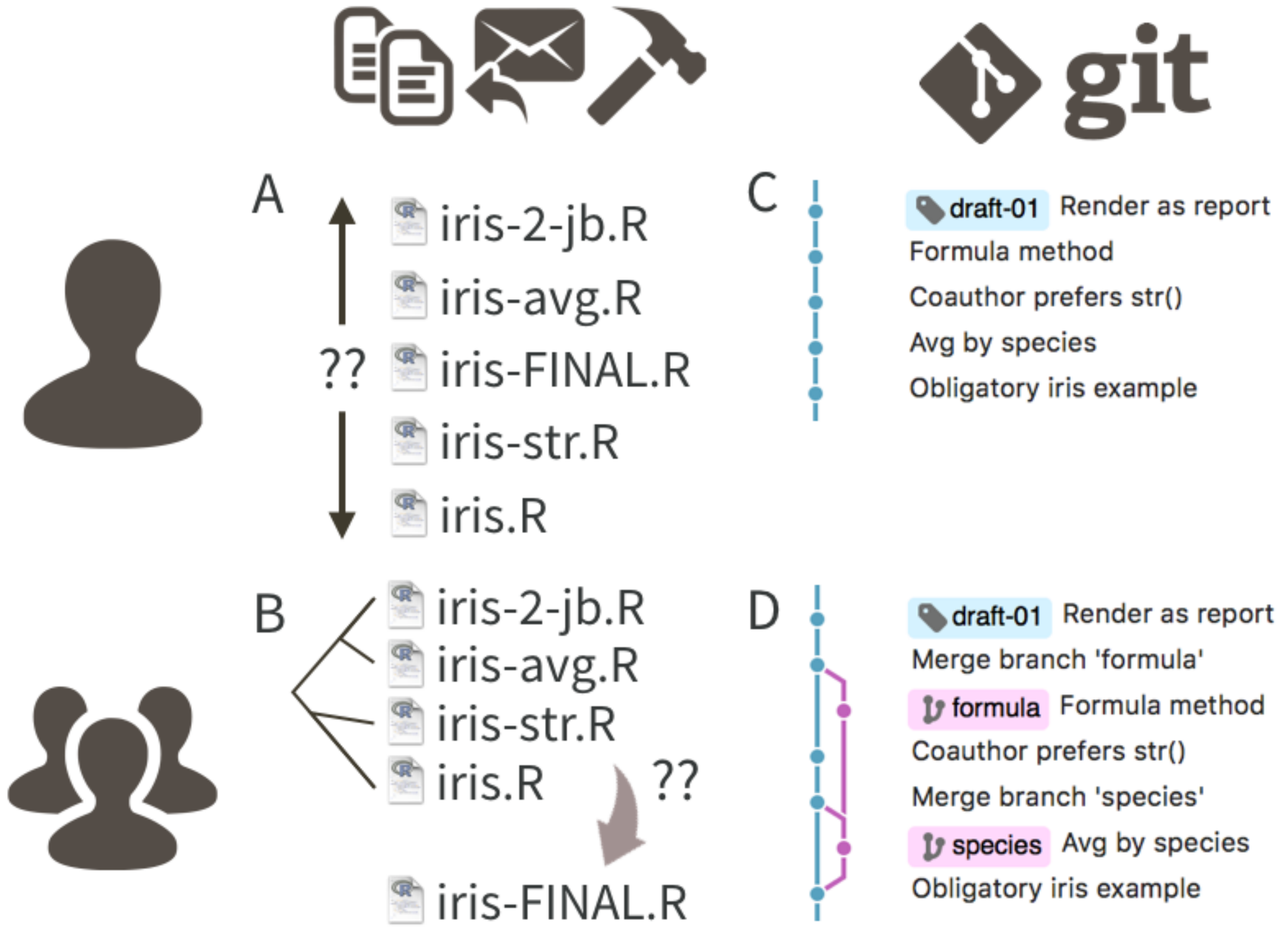
Δ

"diff"

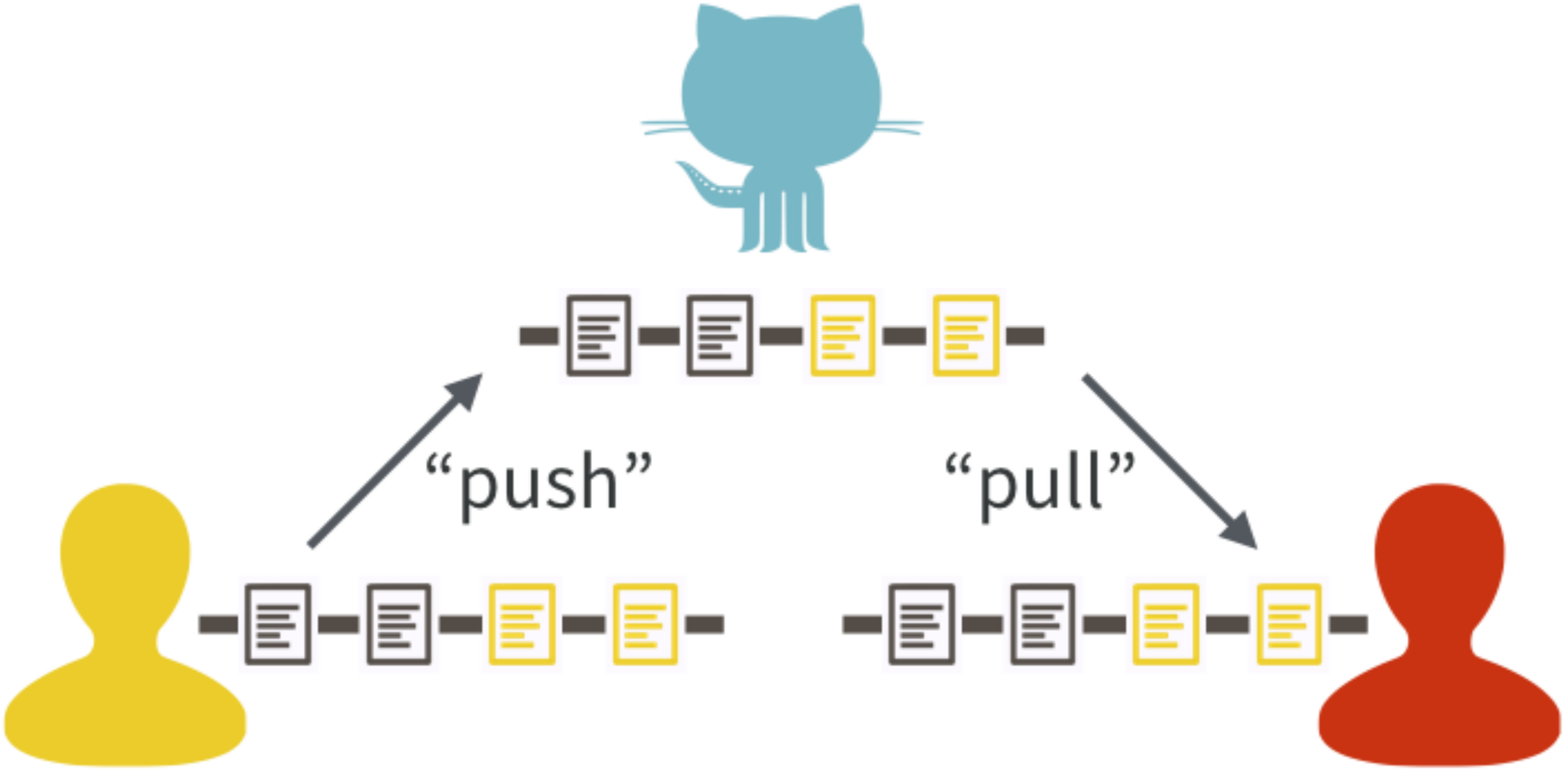
What changed here?

Why?

CC BY Jenny Bryan

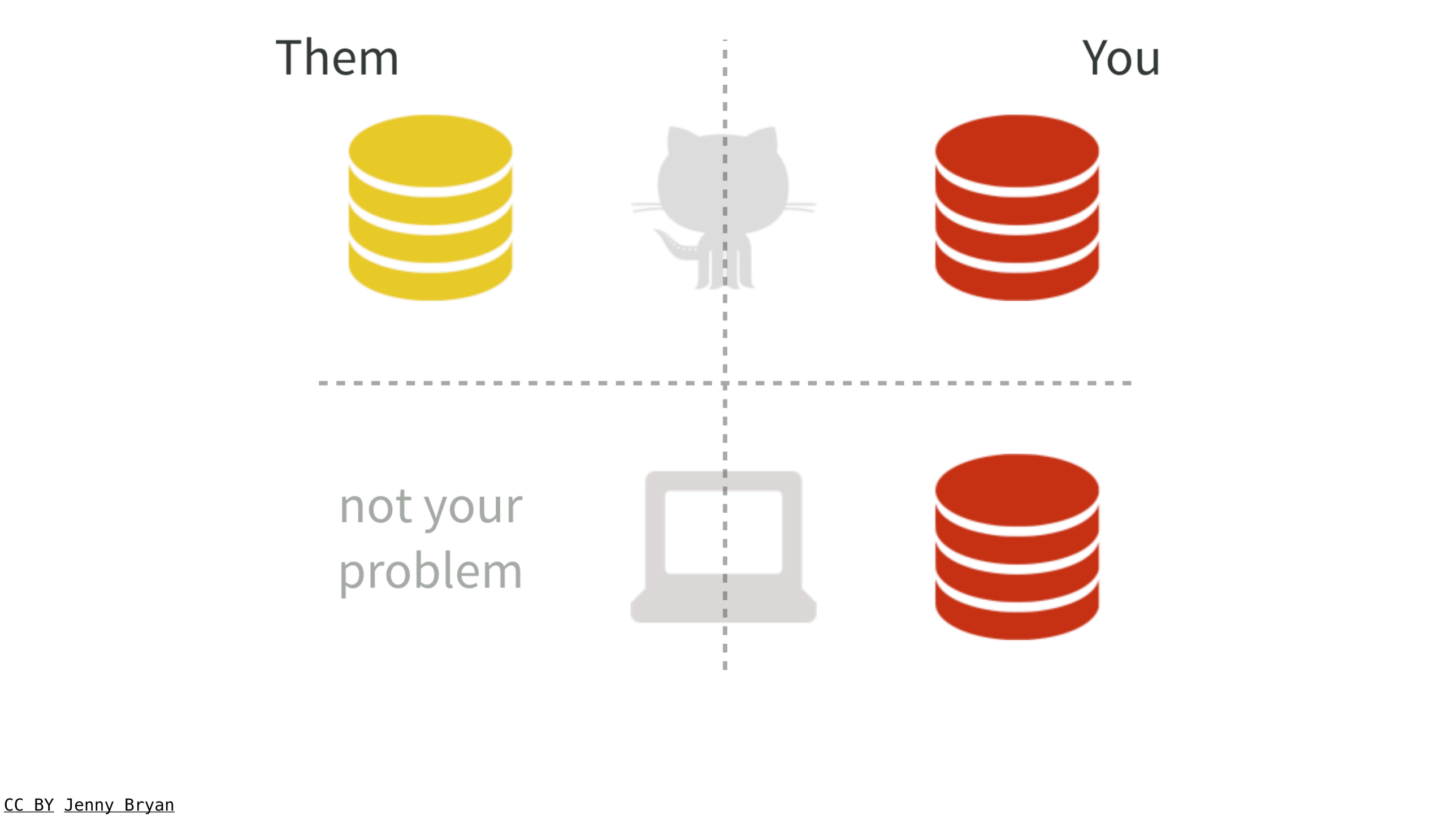


CC BY Jenny Bryan



collaboration

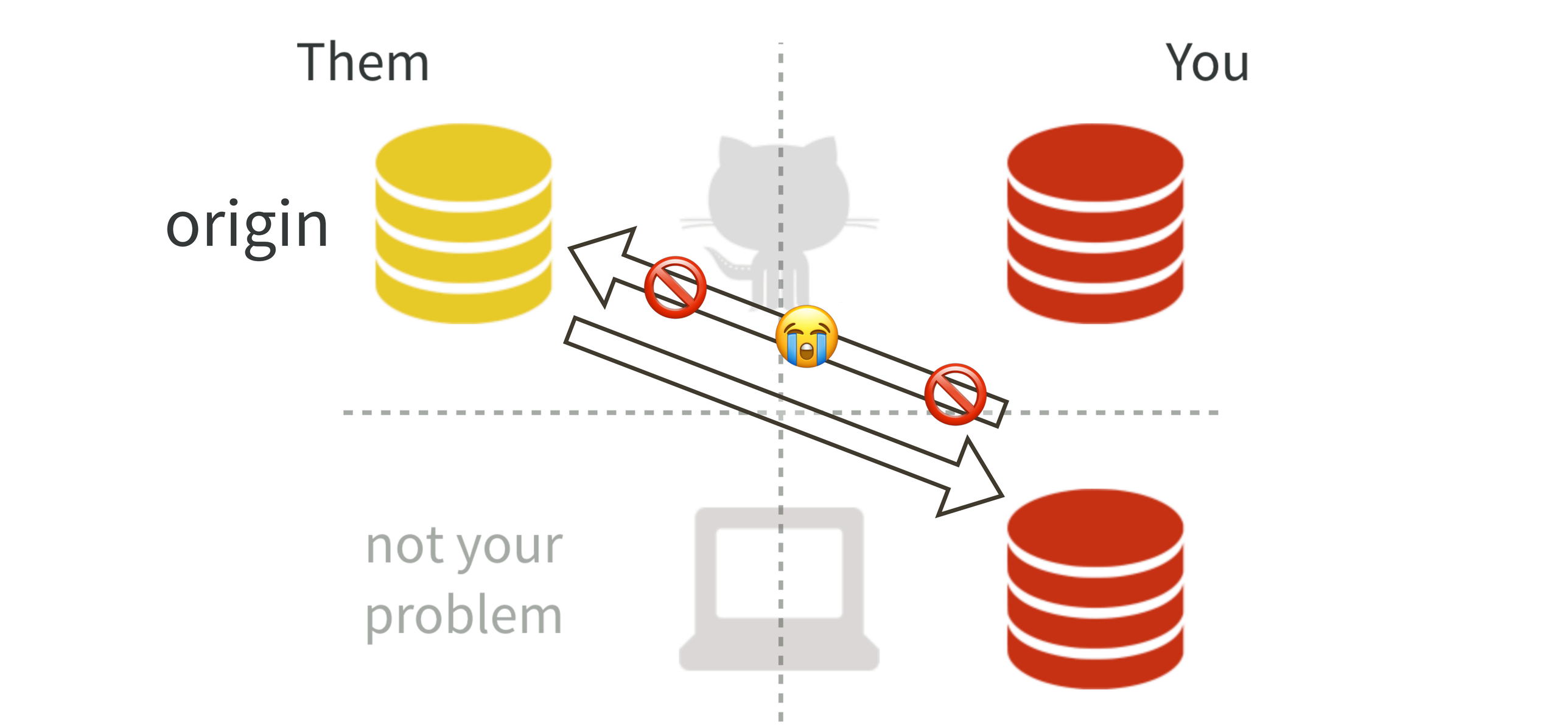
CC BY Jenny Bryan



CC BY Jenny Bryan

"clone"

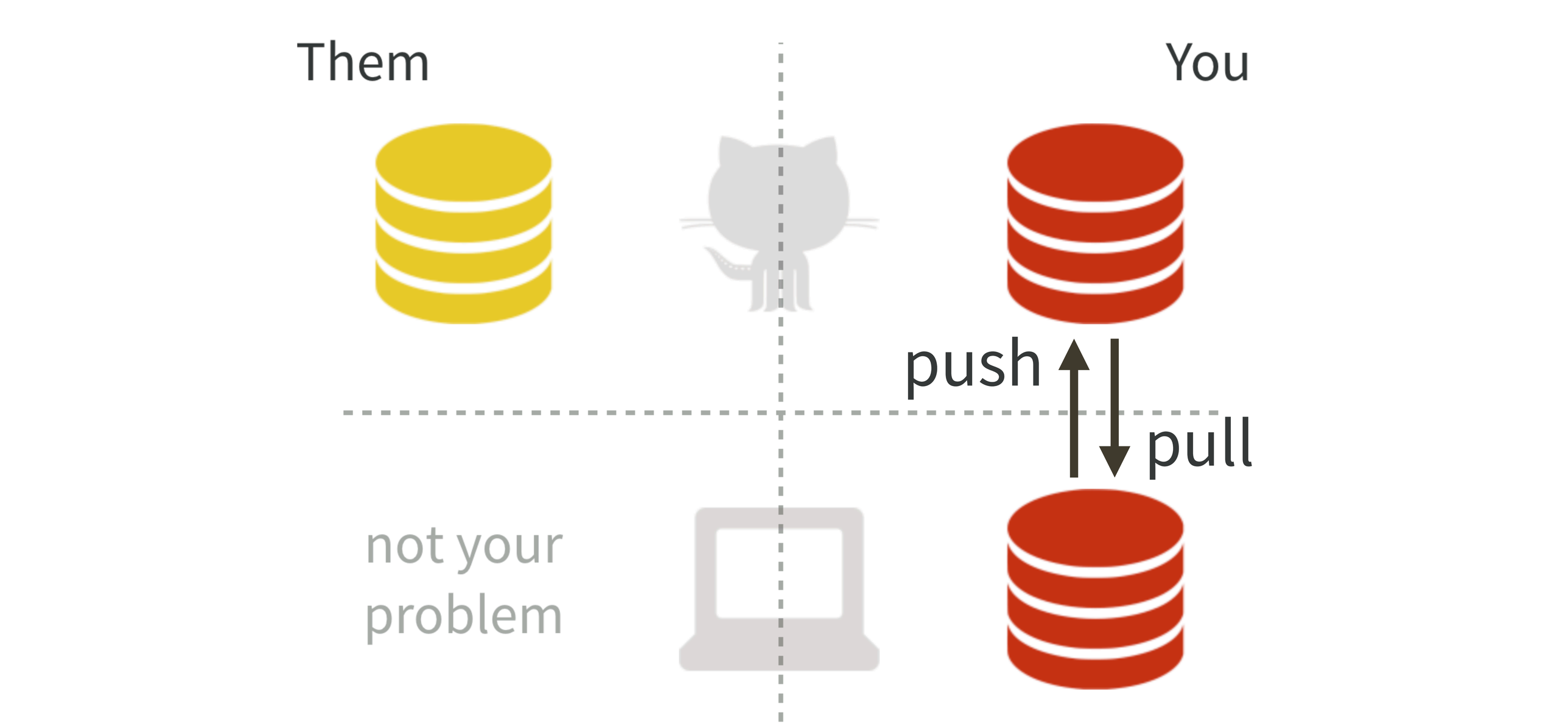
CC BY Jenny Bryan



"clone"

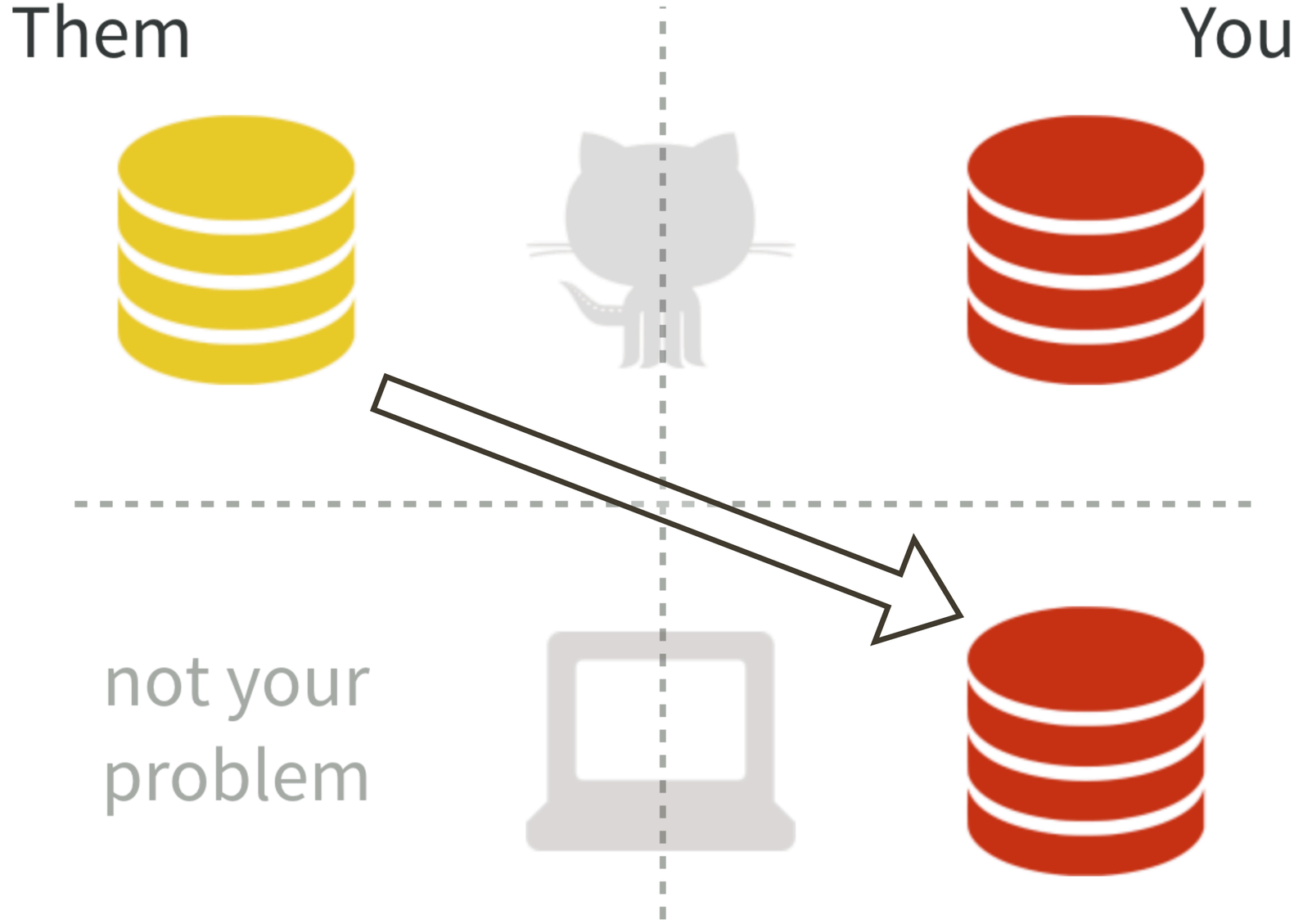
*not as useful as you might think because you can never send a PR

CC BY Jenny Bryan



daily work, your stuff

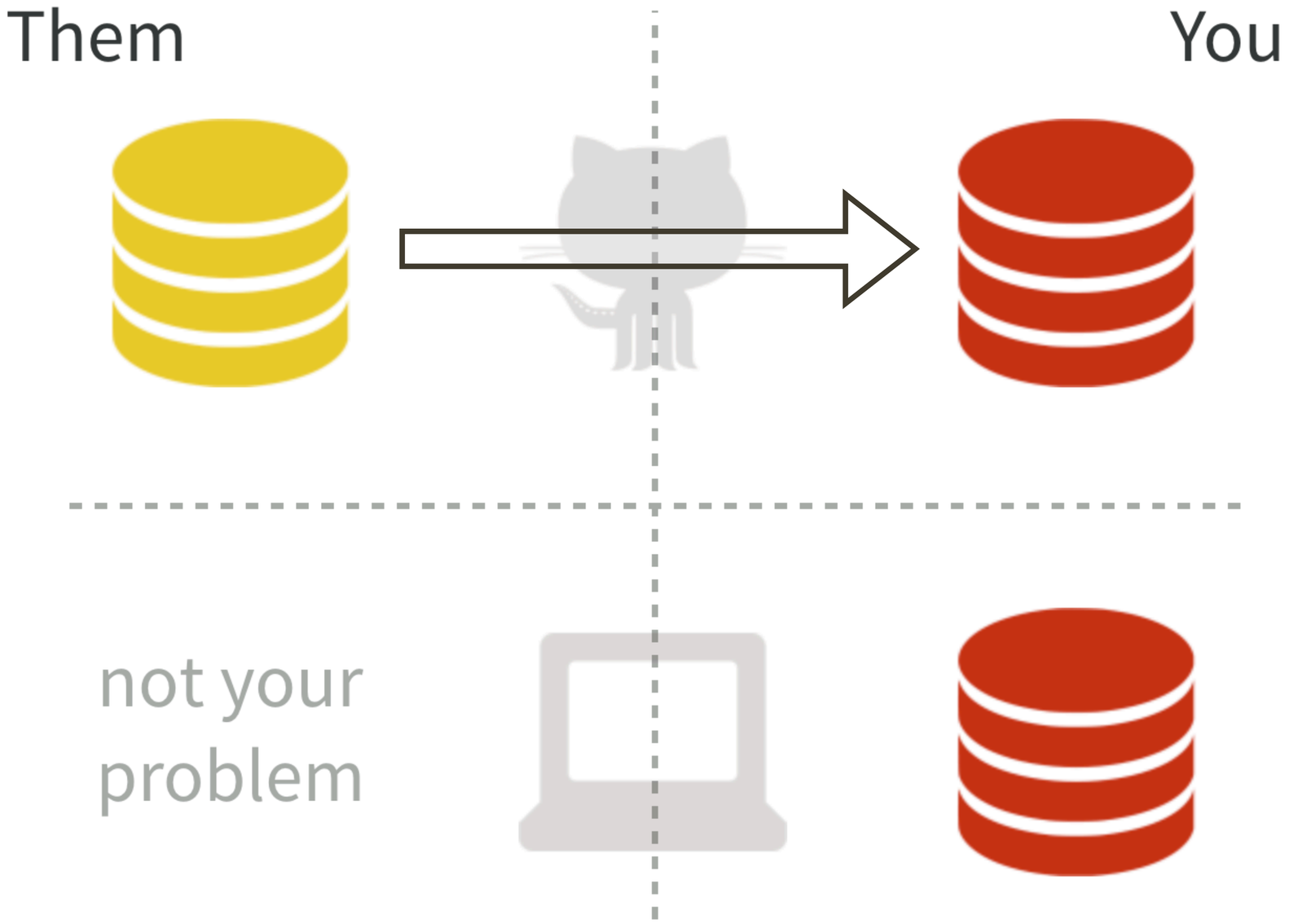
CC BY Jenny Bryan



"clone"

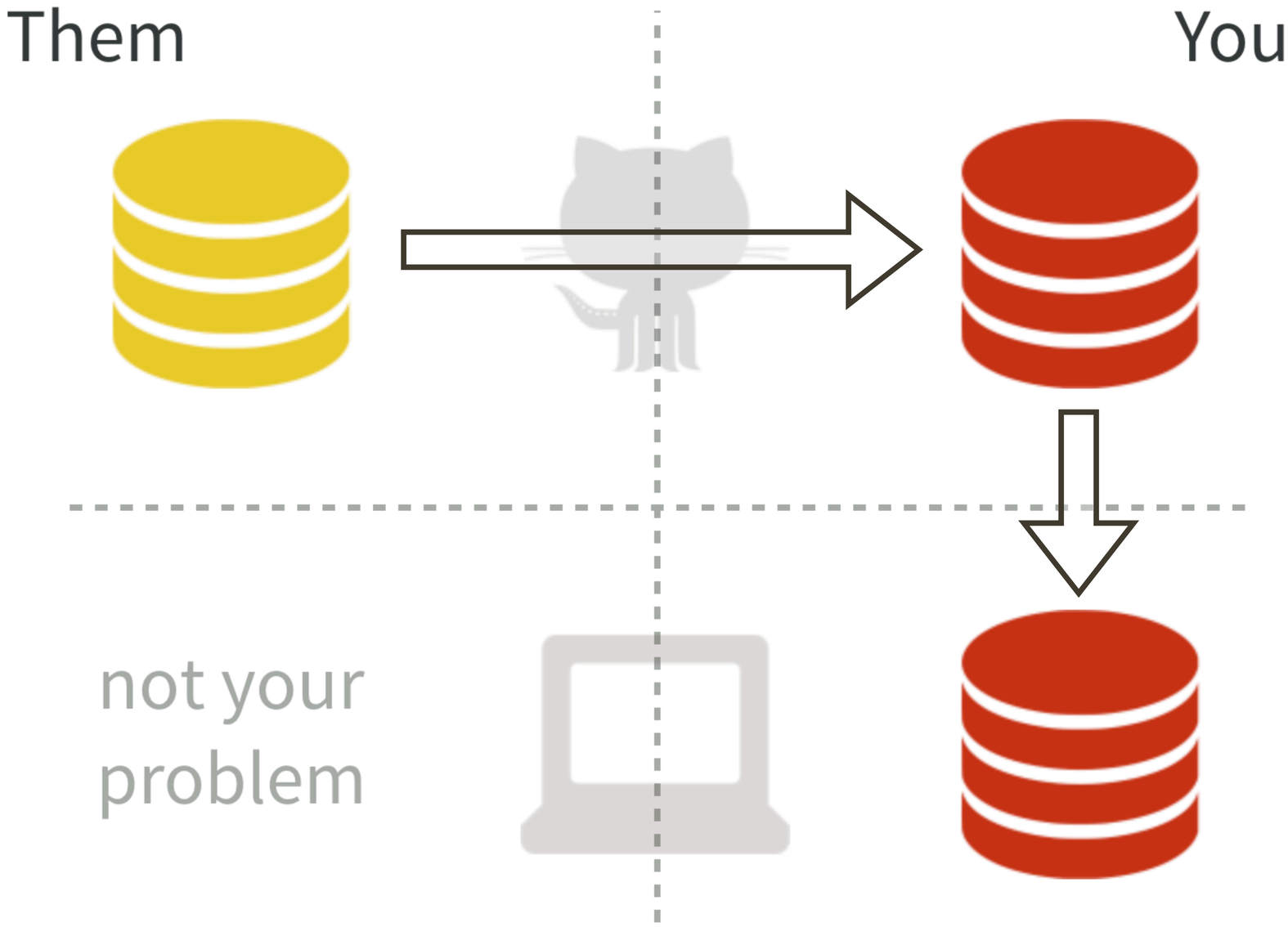
*not as useful as you might think

CC BY Jenny Bryan



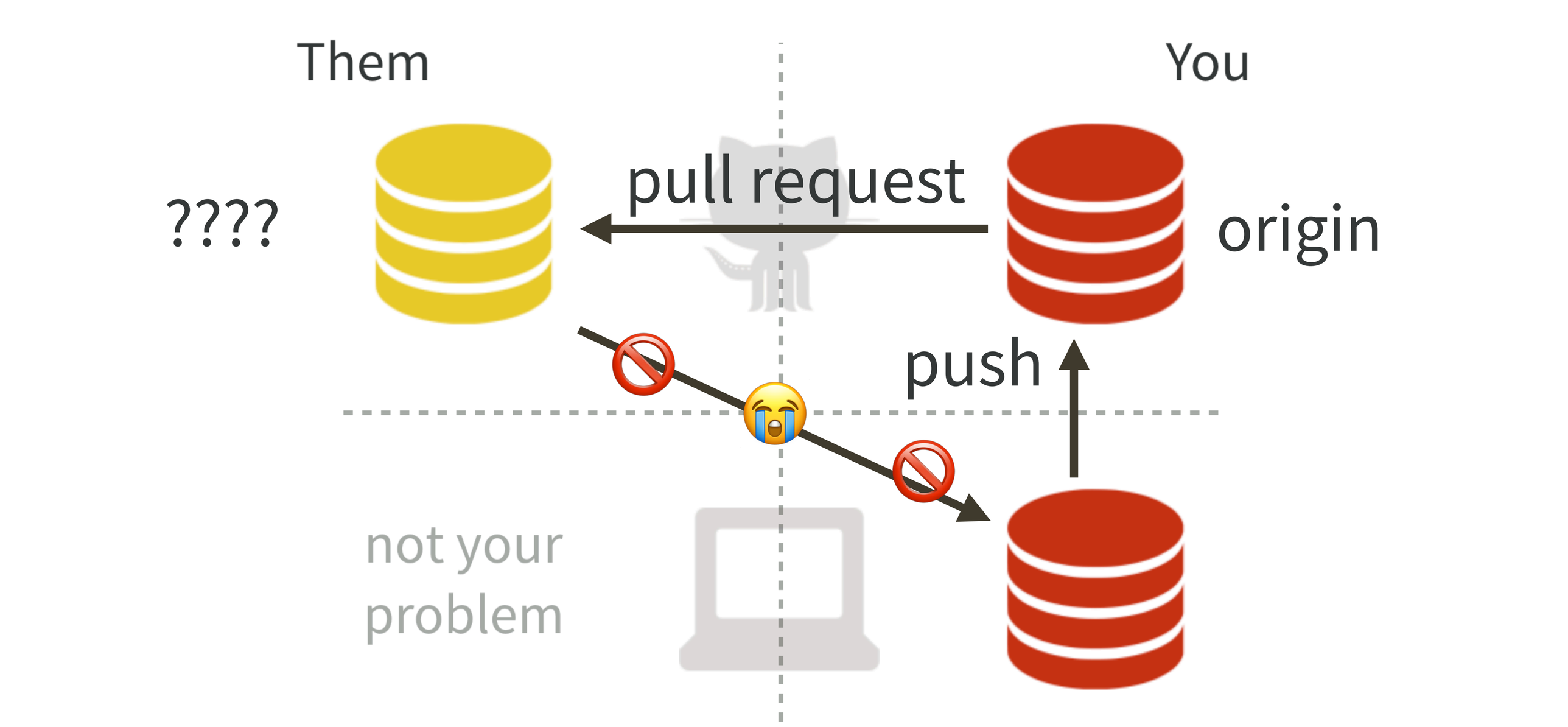
"fork"

CC BY Jenny Bryan



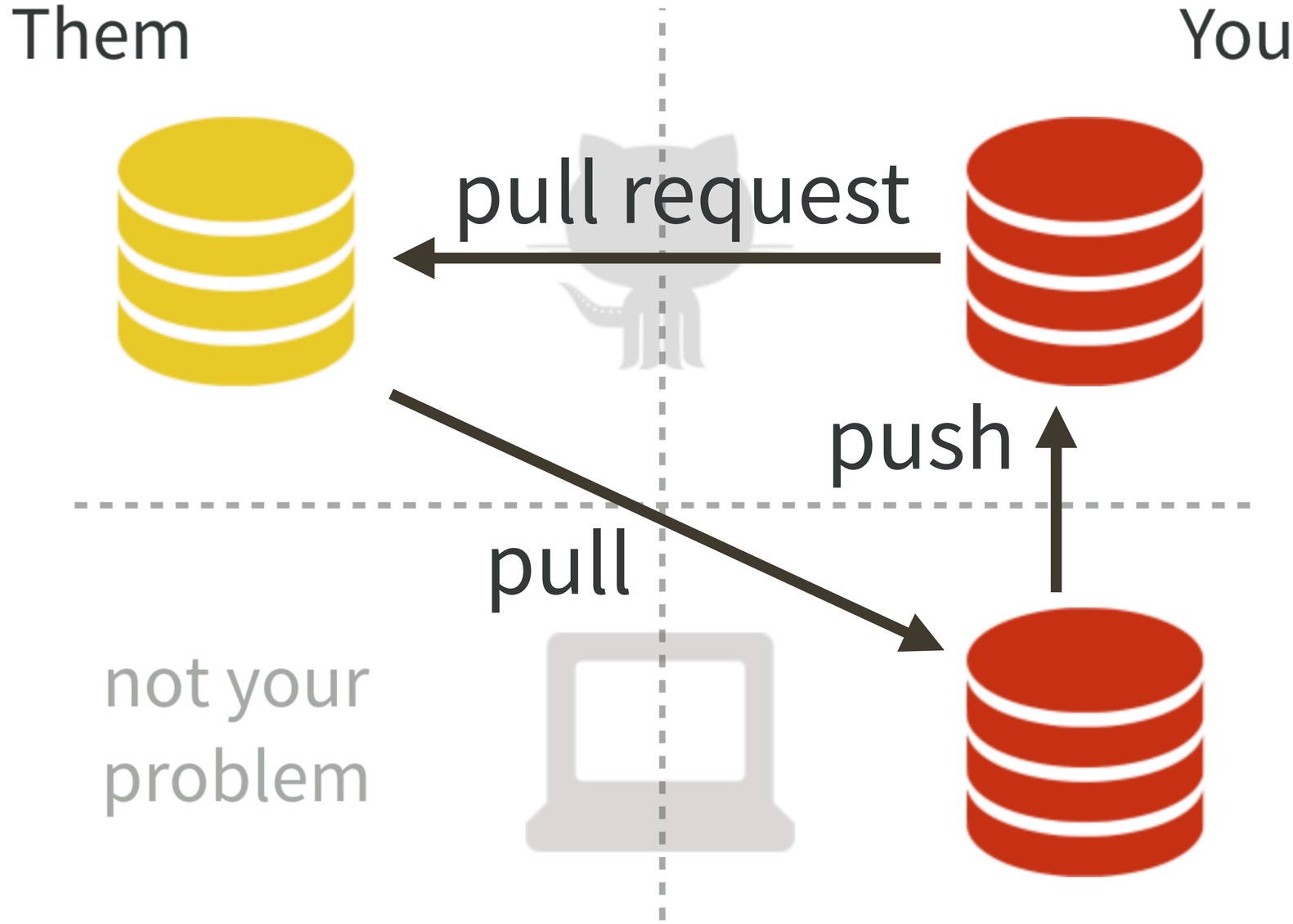
"fork and clone"

CC BY Jenny Bryan



get changes from the main repo

CC BY Jenny Bryan

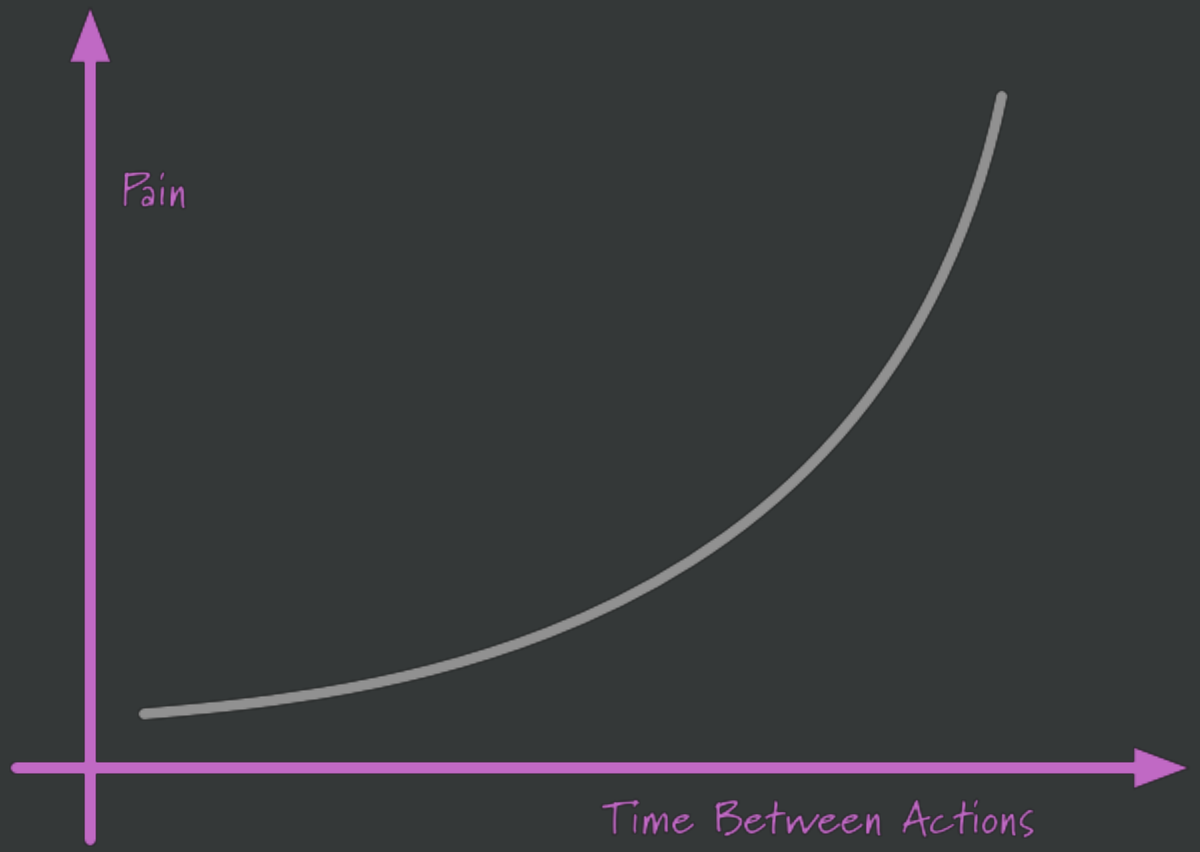


contribute to other people's stuff

CC BY Jenny Bryan

CC BY Jenny Bryan

"If it hurts, do it more often."



https://martinfowler.com/bliki/FrequencyReducesDifficulty.html

CC BY Jenny Bryan

"If it hurts, do it more often."

Apply this to git commit, pull, merge, push.
(and restarting R, re-running your scripts)

Why?
Take your pain in smaller pieces.
Tight feedback loop can reduce absolute pain.
Practice changes what you find painful.

CC BY Jenny Bryan

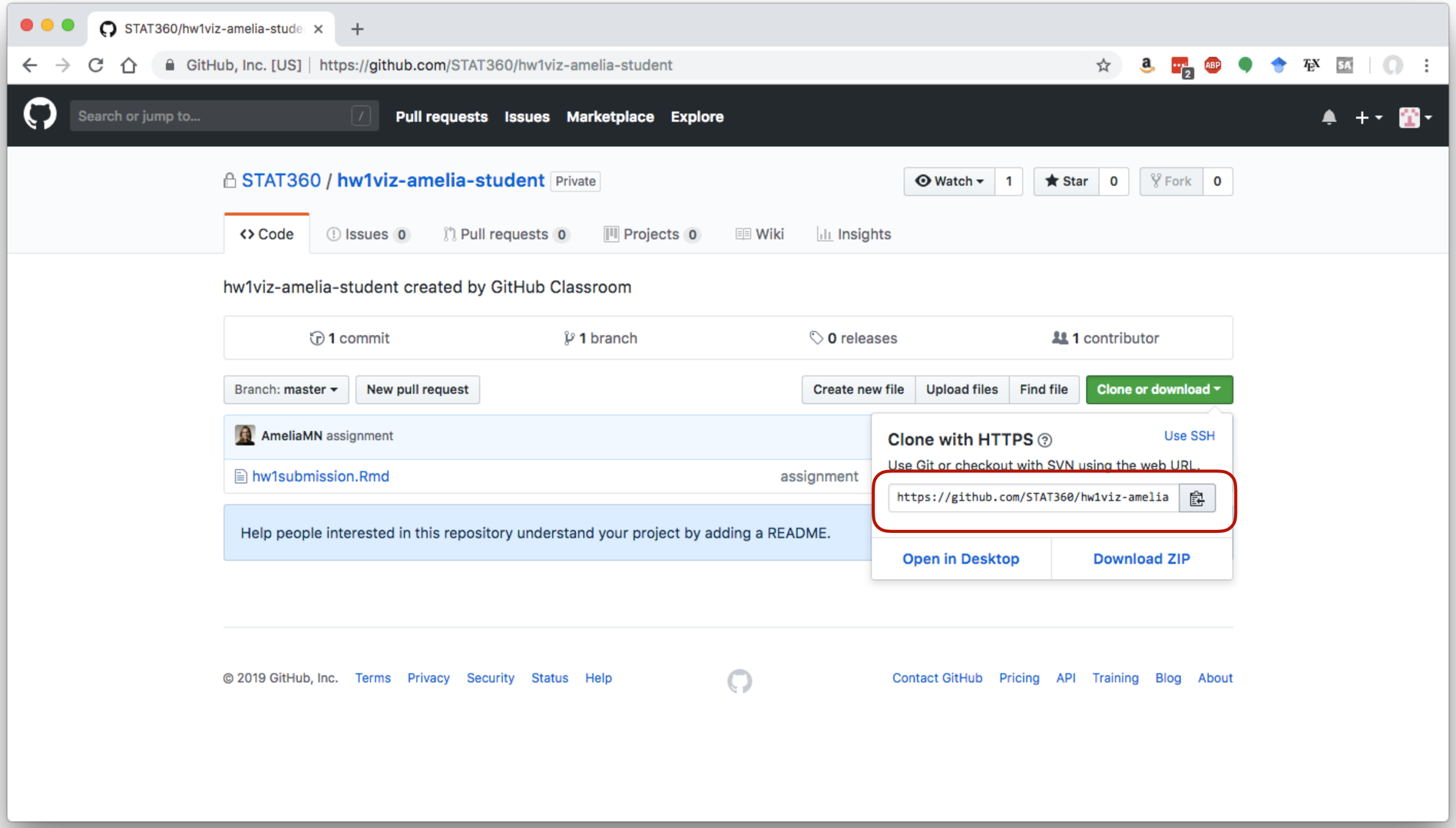
1. Accept the organization invitation for STAT360 in your email

2. Go to github.com/STAT360 and create a team (??)

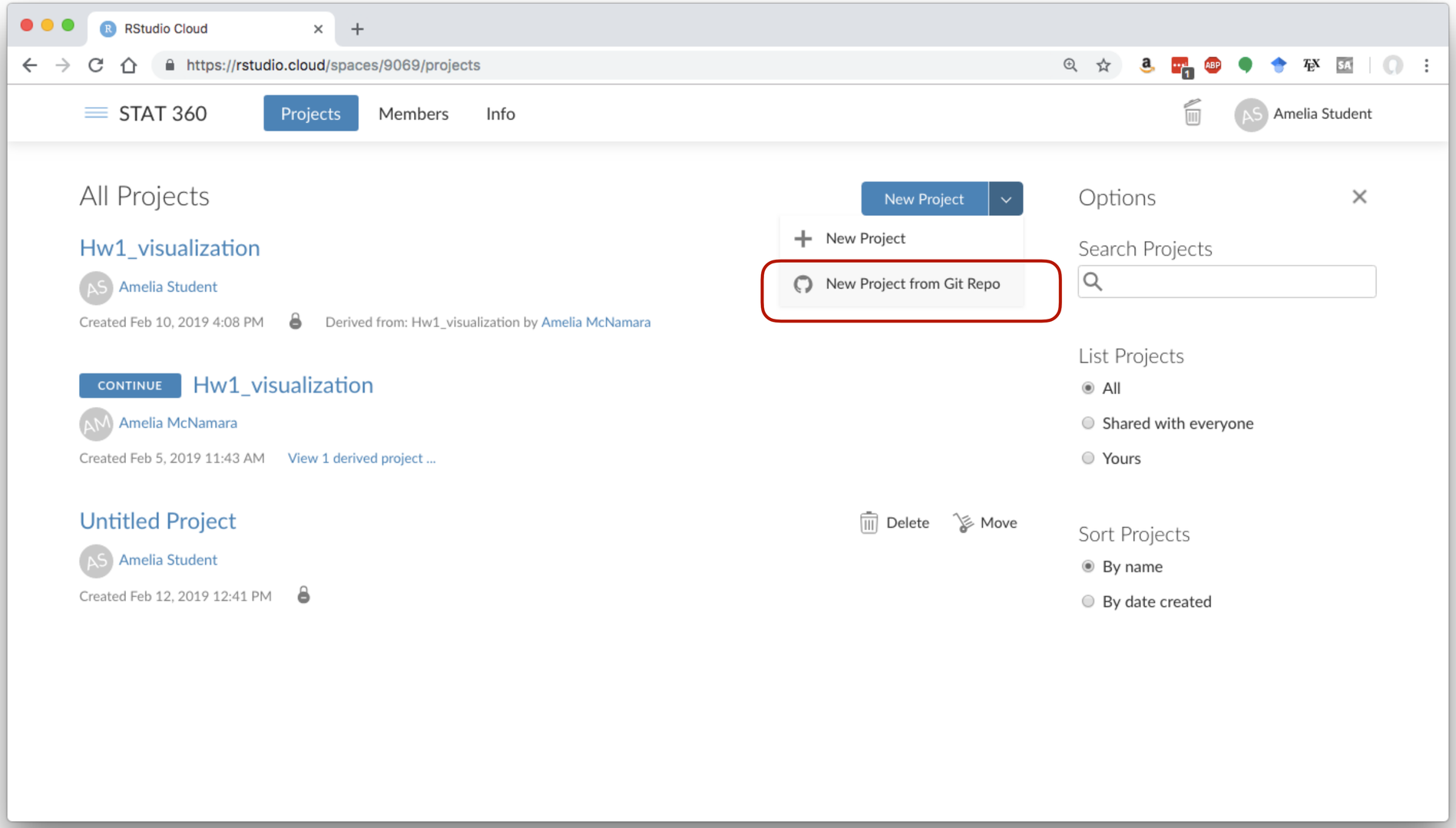
3. One (??) person from your group goes to https://classroom.github.com/g/5uxcrcBk to initialize the repo (fork)

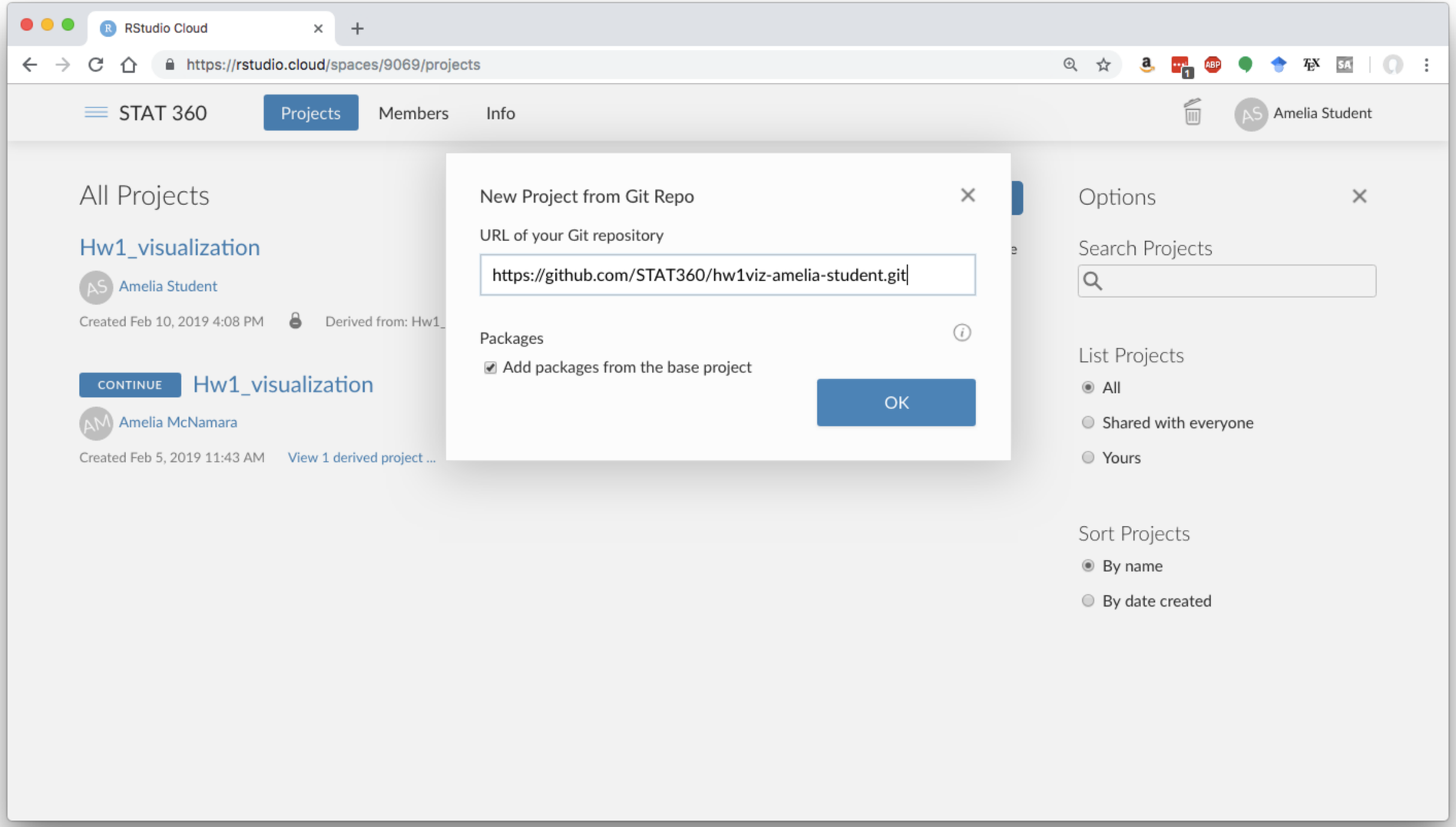
4. From rstudio.cloud, create new project from version control, copy the git url like last time

Copy this URL

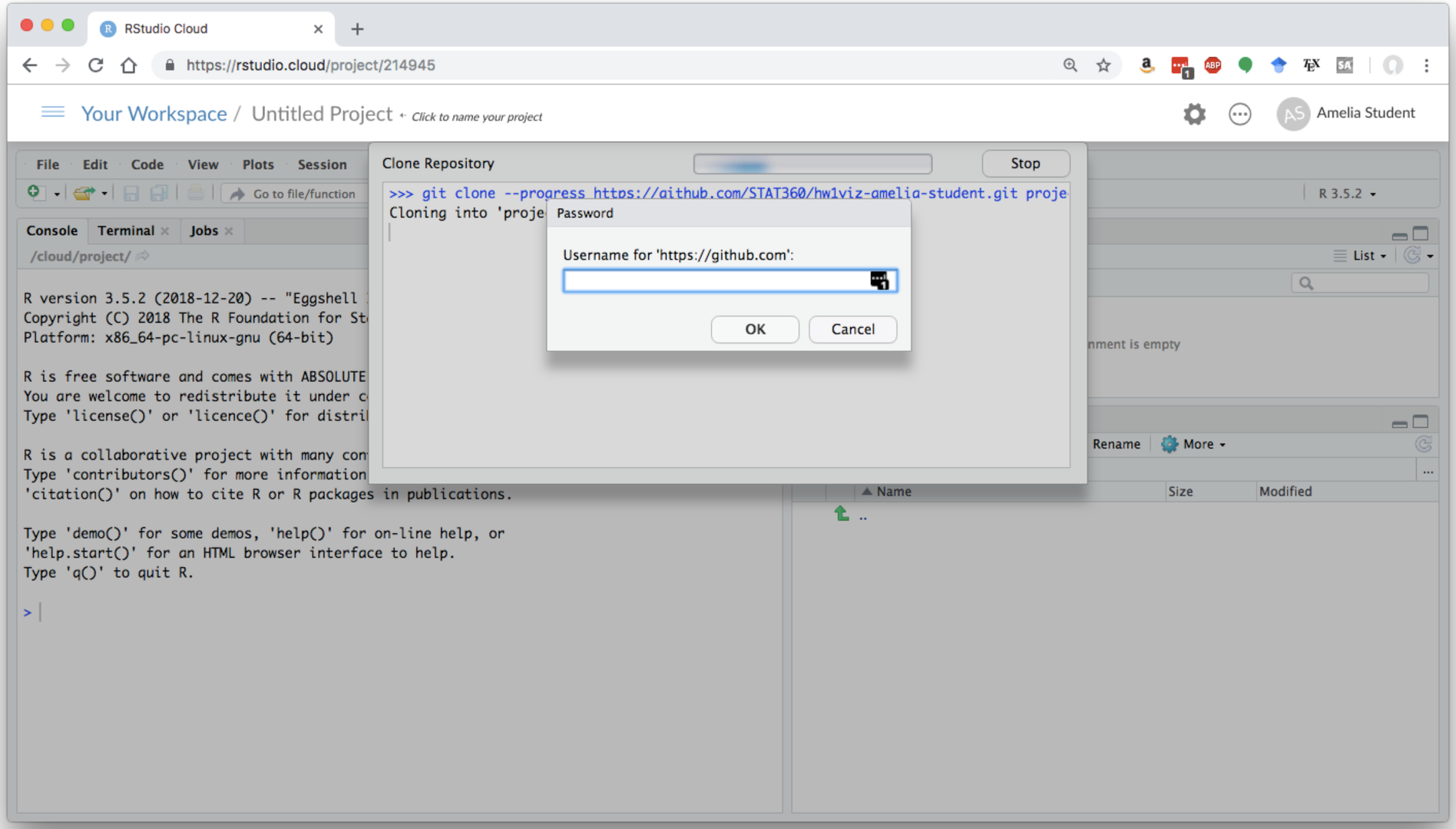


3. Make a new RStudio Cloud Project "from Git Repo"




RStudio Cloud
https://rstudio.cloud/spaces/9069/projects
STAT 360
Projects
Members
Info
Amelia Student
All Projects
Hw1_visualization
Amelia Student
Created Feb 10, 2019 4:08 PM
Derived from: Hw1_
CONTINUE
Hw1_visualization
Amelia McNamara
Created Feb 5, 2019 11:43 AM
View 1 derived project ...
New Project from Git Repo
URL of your Git repository
https://github.com/STAT360/hw1viz-amelia-student.git
Packages
Add packages from the base project
OK
Options
Search Projects
List Projects
All
Shared with everyone
Yours
Sort Projects
By name
By date created

Probably— enter Github credentials



Introduce yourself to git

```
git config --global user.name 'Jane Doe'
git config --global user.email 'jane@example.com'
git config --global credential.helper 'cache --timeout 3600'
git config --global --list
```

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

1. Accept the organization invitation for STAT360 in your email

2. Go to github.com/STAT360 and create a team (??)

3. One (??) person from your group goes to https://classroom.github.com/g/5uxcrcBk to initialize the repo (fork)

4. From rstudio.cloud, create new project from version control, (copy the git url like last time)

5. (??) Make a new branch

6. Have one person work from their machine and make some changes, as the rest of you watch and contribute (maybe, on the screen)

7. One person commits + pushes to Github

8. Everyone else, pull from Github

Recovering from Git(Hub) failure

Scenario: You try to push and cannot

What's the problem?
There are changes on GitHub that you don't have.

Pull. If the gods smile upon you, merge works. Now push.

CC BY Jenny Bryan

Let's create this situation.

Make sure local Git pane is clear.
Make sure local and remote are synced (push, pull).

Edit & commit to file A locally.
Edit & commit to file B remotely.

Try to push. You will fail.

CC BY Jenny Bryan

```
jenny@2015-mbp bunny-scarf $ git push
To github.com:jennybc/bunny-scarf.git
 ! [rejected]        master -> master (fetch first)
error: failed to push some refs to 'git@github.com:jennybc/bunny-scarf.git'
hint: Updates were rejected because the remote contains work that you do
hint: not have locally. This is usually caused by another repository pushing
hint: to the same ref. You may want to first integrate the remote changes
hint: (e.g., 'git pull ...') before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.
```

Remedy? Do what it says!

pull, then push ... pull, then push ... pull, then push

CC BY Jenny Bryan

Look at your Git history.

You will see a merge commit, where the local and remote changes were reconciled.

This is best case scenario and is likely with good Git habits (lots of small frequent commits and merges, no binary files in repo).

CC BY Jenny Bryan

Recovering from Git(Hub) failure

Scenario: You pull and get a merge conflict.

What's the problem?

GitHub can't figure out how to reconcile diffs.

Resolve the conflicts.

Or abort ... and come back later.

CC BY Jenny Bryan

Let's create this situation.

Make sure local Git pane is clear.
Make sure local and remote are synced (push, pull).

Edit & commit to file A locally.
Make conflicting edit & commit to file A remotely.

Try to push. You will fail. Try to pull. You will fail. All is fail.

CC BY Jenny Bryan

```
From github.com:jennybc/bunny-scarf
   958548f..3357952  master     -> origin/master
Auto-merging README.md
CONFLICT (content): Merge conflict in README.md
Automatic merge failed; fix conflicts and then commit the result.


<<<<<<< HEAD
Wingardium Leviosaaaaaaaaa

=======
Wing-GAR-dium Levi-O-sa
>>>>>>> 33579525d88af071268b0a0c64c54f357712589a
```

CC BY Jenny Bryan

```
From github.com:jennybc/bunny-scarf
   958548f..3357952  master     -> origin/master
Auto-merging README.md
CONFLICT (content): Merge conflict in README.md
Automatic merge failed; fix conflicts and then commit the result.


<<<<<<< HEAD
Wingardium Leviosaaaaaaaaa

=======
Wing-GAR-dium Levi-O-sa
>>>>>>> 33579525d88af071268b0a0c64c54f357712589a
```

CC BY Jenny Bryan