Using Git and Github for Version Control, Code Review and Code Sharing

A non-intimidating workshop

Anna Schultze
Electronic Health Records Group, LSHTM
7 Feb 2023

With thanks to…

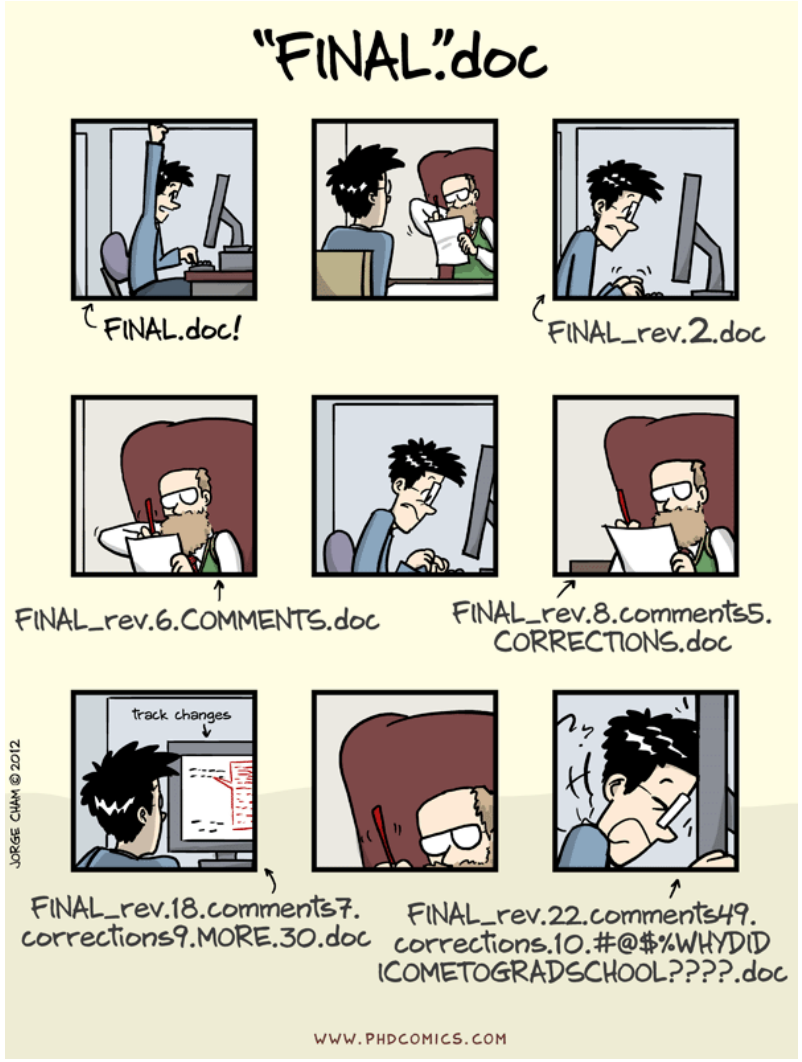
- Caroline Morton (QMUL) and the Bennet Institute software engineers, Oxford
- R user group trainings for slide inspiration

Plan for today

1. **Why git?**
   - Introduction and motivating examples
2. **What is git?**
   - git basics (repositories, local and remote, push and pull)
3. **Collaborative coding**
   - intermediate git (branches, forking, pull requests)
4. **How to use git?**
   - Steps involved in git workflow
5. **Practical**
   - Trying everything out

Why?

Keep track of versions and changes



add handling of between dose time for sccs

Browse files

main (#4)

annaschultze committed on Nov 21, 2022

1 parent cc75001 commit 51cb455024b4af5f01c1293672c6b75c47780b80

Showing **3 changed files** with **51 additions** and **186 deletions**.

Split | Unified

Filter changed files

- 00_sensitivity_functions.R
- 01_sensitivity_clean_dat...
- 02_sensitivity_analysis.R

227 00_sensitivity_functions.R

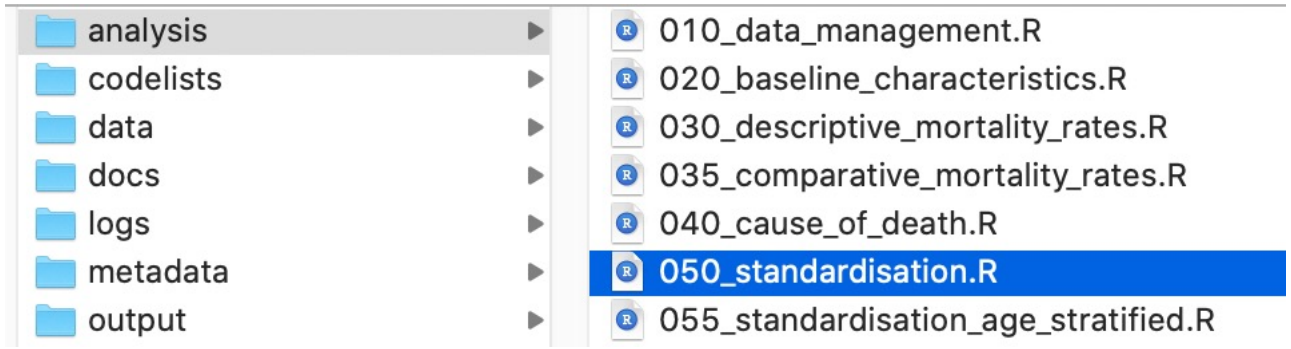
```
@@ -32,12 +32,12 @@
dir.create(file.path(paste0(diroutput,
"sccs_sensitivity")),        showWarning
32 32  #' @description
33 33  #' Takes clean data for self-controlled designs and
       does design/outcome specific data management.
34 34  #'
35     - #' @param data = Input dataframe, should be loaded.
36     - #' @param outcome = Outcome variable name as a string,
       referencing a _date variable.
   35  + #' @param data = Input dataframe, should be loaded
       into R before invoking, as function does not do this.
   36  + #' @param outcome = Outcome variable name as a string,
       referencing a date variable.
```

A sidenote on folder structure…

- A good folder structure is as important as learning git
  - Choosing a consistent folder and naming structure **will** make your life easier

Box 3. Project layout

```
.
|-- CITATION
|-- README
|-- LICENSE
|-- requirements.txt
|-- data
|    |-- birds_count_table.csv
|-- doc
|    |-- notebook.md
|    |-- manuscript.md
|    |-- changelog.txt
|-- results
|    |-- summarized_results.csv
|-- src
|    |-- sightings_analysis.py
|    |-- runall.py
```



Wilson, Bryan et al. Good enough practices in scientific computing, PLOS Biology, 2017

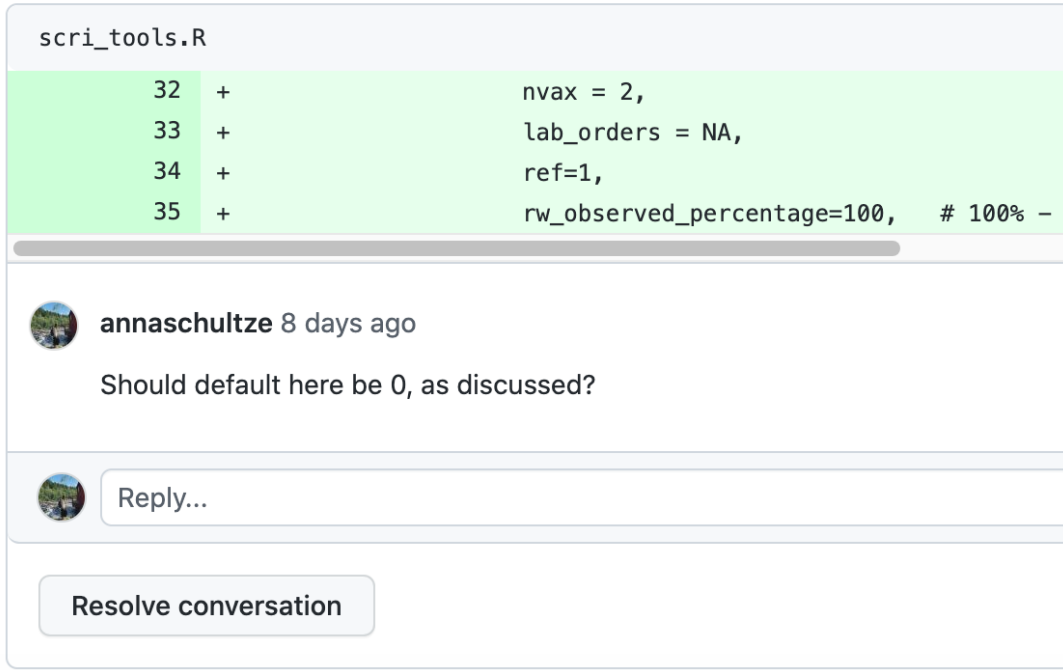
Collaboration

From this…

```
## Sophie: here event_in_rw is set to F
## I believe Svetlana said to ignore this one (even delete it)
## Anna: In favour of deleting whatever we can to simplify
formula_text <-  "~ lab"

res <- scri_strata( output_name   = output_name,
                    formula_text = formula_text,       time_seq
                    event_time = paste0(iae,"_days"), event = ia
                    rws          = rws_def,
                    start_obs    = "study_entry_days", end_obs =
                    data         = scri_input[cond_iae,],
                    rw_observed_percentage = 100,
```

To this



Enhancing Transparency through Code Sharing

Hydroxychloroquine or chloroquine with or without a macrolide for treatment of COVID-19: a multinational registry analysis

Mandeep R Mehra, Sapan S Desai, Frank Ruschitzka, Amit N Patel

Summary

Background Hydroxychloroquine or chloroquine, often in combination with a second-generation m[illegible]ide, are b[illegible] widely used for treatment of COVID-19, despite no conclusive evidence of their benefit. Although [illegible]enerally [illegible]fe whe[illegible] used for approved indications such as autoimmune disease or malaria, the safety and bene[illegible]f th[illegible]treatment regimens are poorly evaluated in COVID-19.

Methods We did a multinational registry analysis of the use of hydroxychloroquine [illegible]oquine wit[illegible]thout a macrolide for treatment of COVID-19. The registry comprised data from 671 hosp[illegible]s in s[illegible]ntinents. We included patients hospitalised between Dec 20, 2019, and April 14, 2020, with a positive laboratory [illegible]g for SARS-CoV-2. Patients who received one of the treatments of interest within 48 h of diagn[illegible]e included in [illegible]f four treatment groups (chloroquine alone, chloroquine with a macrolide, hydroxychlor[illegible]ine alone, or hydroxychloroquine with a macrolide), and patients who received none of these treatments formed [illegible] control gr[illegible]. Patients for whom one of the treatments of interest was initiated more than 48 h after diagnosis o[illegible]ile they w[illegible]on mechanical ventilation, as well as patients who received remdesivir, were excluded. The main outc[illegible]of in[illegible]st were in-hospital mortality and the occurrence of de-novo ventricular arrhythmias ([illegible]tained o[illegible]ed ventricular tachycardia or ventricular fibrillation).

Findings 96 032 patients (mean age 53·8 years, 46·3% women) [illegible]OVID-19 were hospitalised during the study period and met the inclusion criteria. Of the[illegible] patie[illegible]were in the treatment groups (1868 received chloroquine, 3783 received chloroquine with [illegible]macro[illegible]e, 3016 [illegible]eived hydroxychloroquine, and 6221 received hydroxychloroquine with a macrolide) and [illegible] pati[illegible]e in the control group. 10 698 (11·1%) patients died in hospital. After controlling for multiple [illegible]fou[illegible]ctors ([illegible] sex, race or ethnicity, body-mass index, underlying cardiovascular disease and its risk fac[illegible] diabetes[illegible]derlying lung disease, smoking, immunosuppressed condition, and baseline disease severity), wh[illegible]mpared wi[illegible]ortality in the control group (9·3%), hydroxychloroquine (18·0%; hazard ratio 1·335, 95%[illegible]·22[illegible]457), hydro[illegible]ychloroquine with a macrolide (23·8%; 1·447, 1·368–1·531), chloroquine (16·4%; 1·365, [illegible]318–1·531), [illegible]chloroquine with a macrolide (22·2%; 1·368, 1·273–1·469) were each independently associated [illegible]n an increased r[illegible]f in-hospital mortality. Compared with the control group (0·3%), hydroxychloroquine (6[illegible]; 2·36[illegible]·935–2·900), hydroxychloroquine with a macrolide (8·1%; 5·106, 4·106–5·983), chloroquine (4·3%; [illegible]1, 2[illegible]0–4·596), and chloroquine with a macrolide (6·5%; 4·011, 3·344–4·812) were independently associate[illegible]an inc[illegible]d risk of de-novo ventricular arrhythmia during hospitalisation.

Interpretati[illegible] We v[illegible]unabl[illegible]nfirm a benefit of hydroxychloroquine or chloroquine, when used alone or with a macro[illegible]on in[illegible]spital outcomes for COVID-19. Each of these drug regimens was associated with decreased in-hospita[illegible]i[illegible]reased frequency of ventricular arrhythmias when used for treatment of COVID-19.

Funding William[illegible]vey Distinguished Chair in Advanced Cardiovascular Medicine at Brigham and Women's Hospital.

[illegible]ed Online
[illegible]22, 2020
https://doi.org/10.1016/S0140-6736(20)31180-6

This online publication has been corrected. The corrected version first appeared at thelancet.com on May 29, 2020

See Online/Comment
https://doi.org/10.1016/S0140-6736(20)31174-0

Brigham and Women's Hospital Heart and Vascular Center and Harvard Medical School, Boston, MA, USA (Prof M R Mehra MD); Surgisphere Corporation, Chicago, IL, USA (S S Desai MD); University Heart Center, University Hospital Zurich, Zurich, Switzerland (Prof F Ruschitzka MD); Department of Biomedical Engineering, University of Utah, Salt Lake City, UT, USA (A N Patel MD); and HCA Research Institute, Nashville, TN, USA (A N Patel)

Correspondence to:
Prof Mandeep R Mehra, Brigham and Women's Hospital Heart and Vascular Center and Harvard Medical School, Boston, MA 02115, USA
mmehra@bwh.harvard.edu

RETRACTED

Introduction

The absence of an effective treatment against severe [illegible] drugs have been shown in laboratory conditions to have antiviral properties as well as immunomodulatory

A review commissioned by the Secretary of State for Health and Social Care

Better, Broader, Safer: Using Health Data for Research and Analysis

April 2022

Open 3. Make open code a boilerplate feature of all public contracts

Open code sharing should be a required feature of all standard contracts between the NHS and any external provider of code for health data management and analysis; with a similar arrangement for academic funders; and for any university or other body sub-contracting such work.

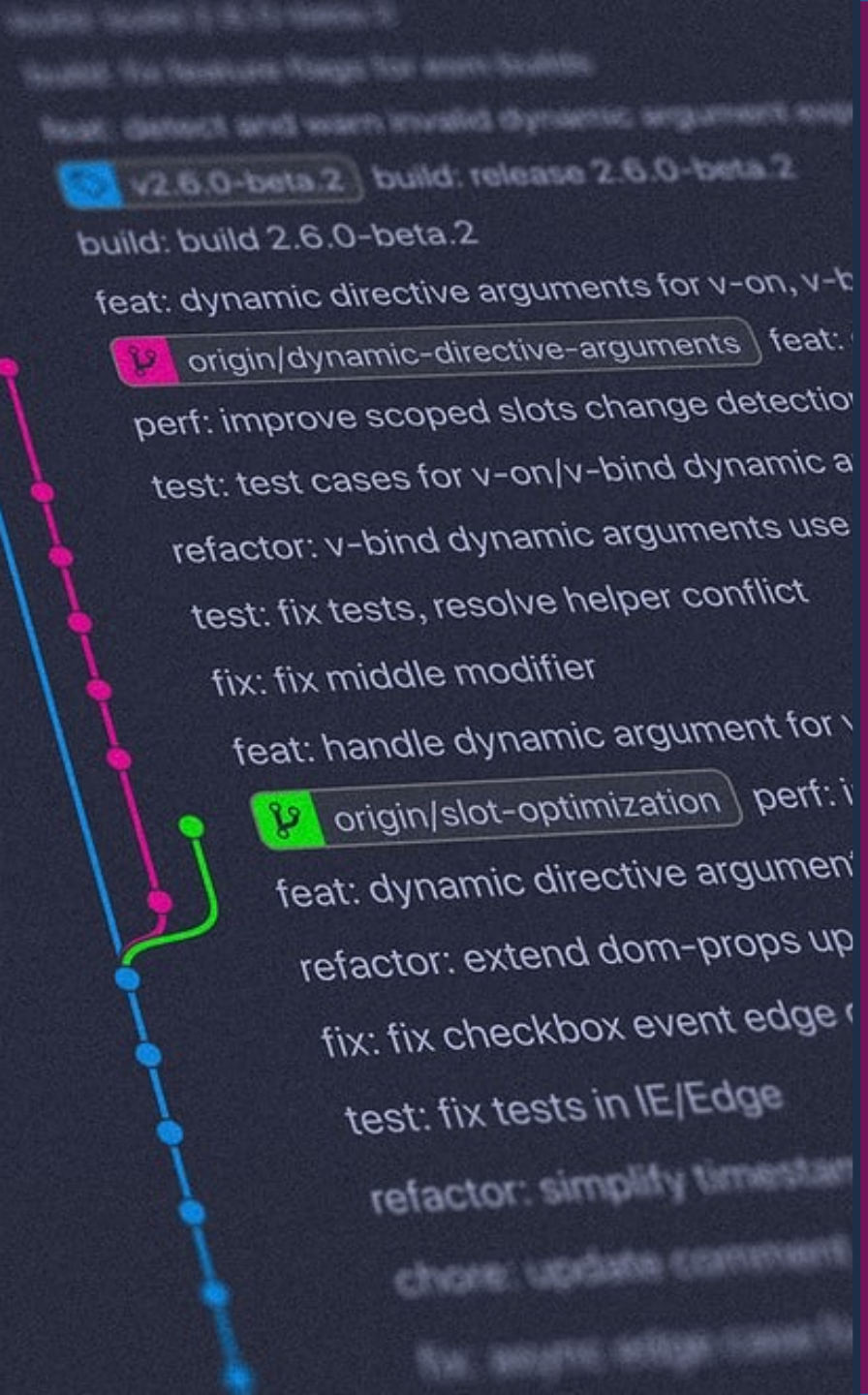
Github is not your ONLY option

For version control/collaboration:

- Gitlab
- Bitbucket

For code sharing:

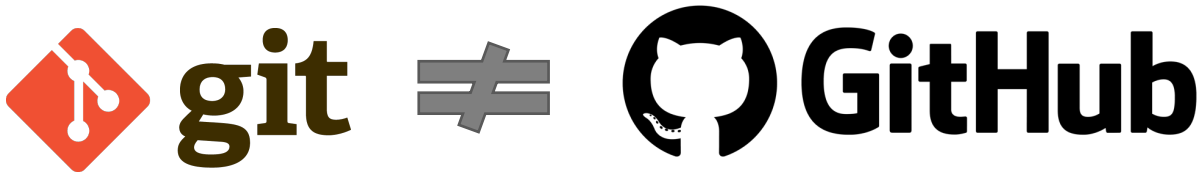
- OpenScienceFramework (OSF)
- User Groups (i.e, PharmaSUG)
    - may be particularly relevant if you want to share “code tricks” but don’t own the code you write


v2.6.0-beta.2
build: release 2.6.0-beta.2
build: build 2.6.0-beta.2
feat: dynamic directive arguments for v-on, v-b
origin/dynamic-directive-arguments
perf: improve scoped slots change detectio
test: test cases for v-on/v-bind dynamic a
refactor: v-bind dynamic arguments use
test: fix tests, resolve helper conflict
fix: fix middle modifier
feat: handle dynamic argument for
origin/slot-optimization
feat: dynamic directive argumen
refactor: extend dom-props up
fix: fix checkbox event edge
test: fix tests in IE/Edge

What?

Git and github…

- Git is a version control tool, and Github is an online platform that hosts git projects
  - Git does the version control
  - Github makes it easier to use git, and also facilitates code sharing and code reviews

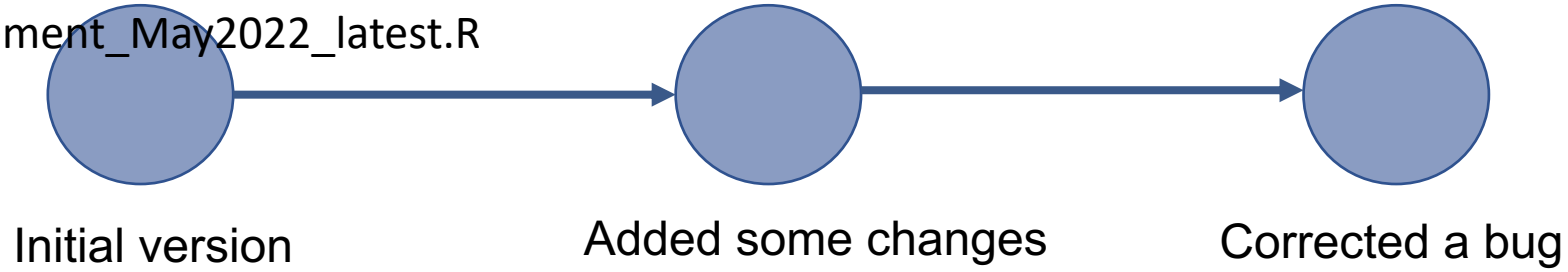


Google docs, but for your code

What does git do?

- Git is a free and open-source version-control software
  - lives on your local computer
  - “watches” files in specific folders (**repositories**) you tell it to monitor
- Tracks changes to files (any plain text file – NOT word documents)
- Allows you to:
  - save changes to a file through a **commit**, typically with some informative message
  - **revert** to previous versions, or see what changed (and why!)

01_data_management.R
01_data_management_May2022.R
01_data_management_May2022_ASreview.R
01_data_management_May2022_latest.R



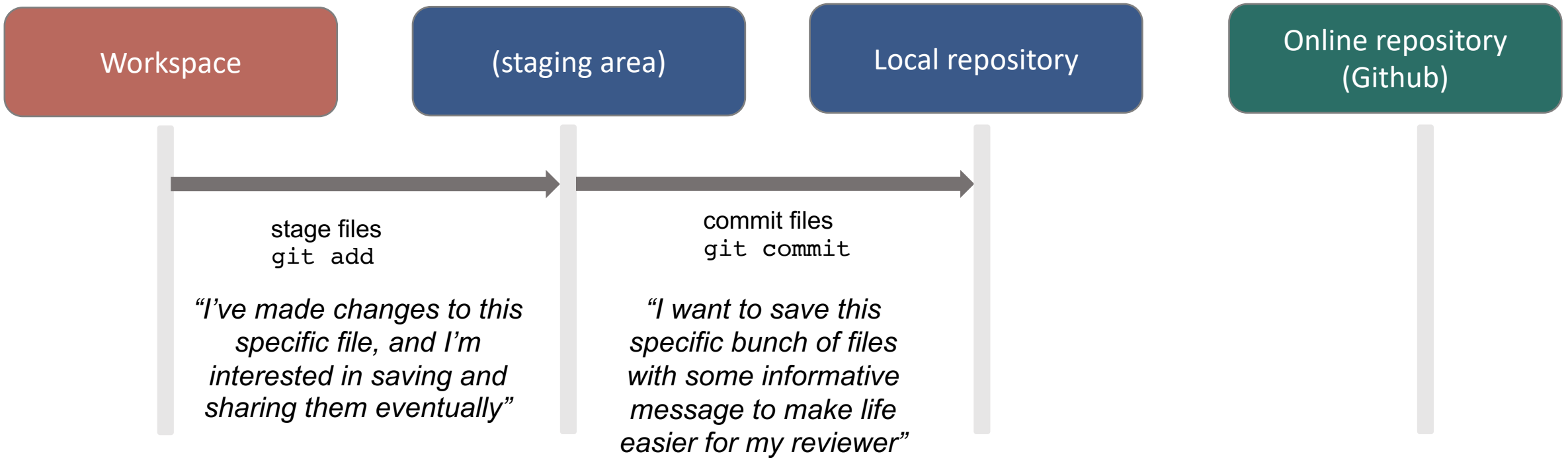
Traditional version control

Filenames	Messages
01_data_management.R	?
01_data_management_May2022.R	?
01_data_management_May2022_ASreview.R	?
01_data_management_May2022_latest.R	?

Git commits

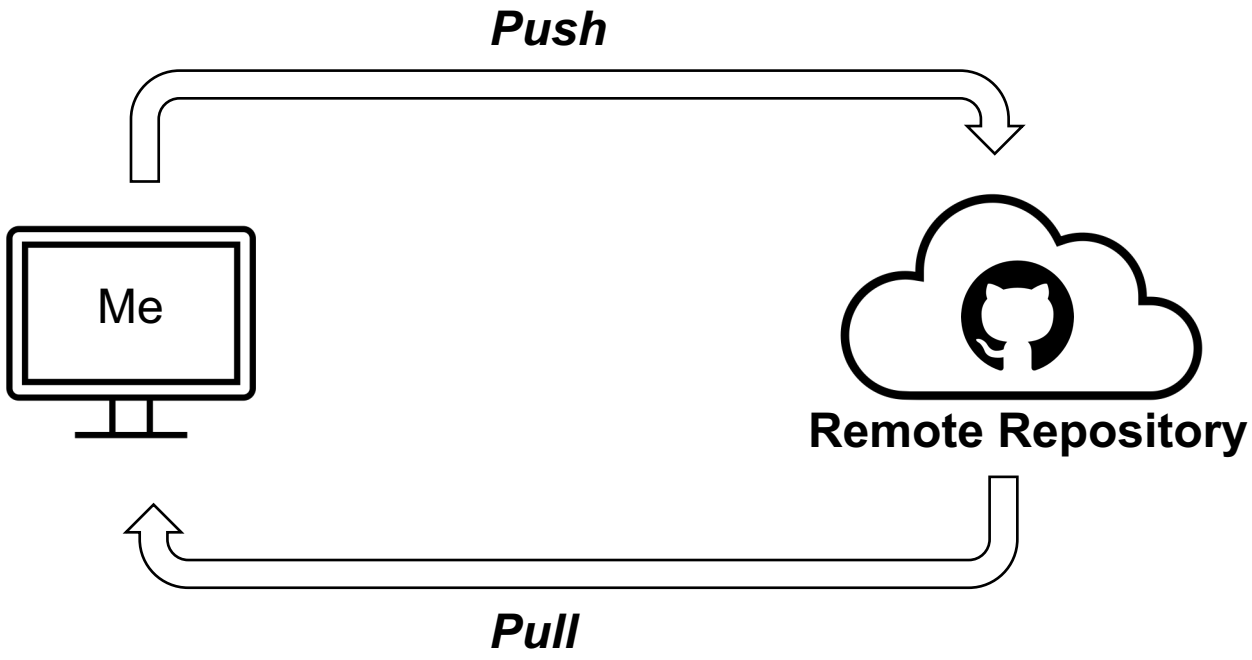
Filenames	Messages
01_data_management.R	initial coding for review
	added creatinine values
	corrections after AS code review
	added in updated ethnicity algorithm

Saving with git - a two-step process

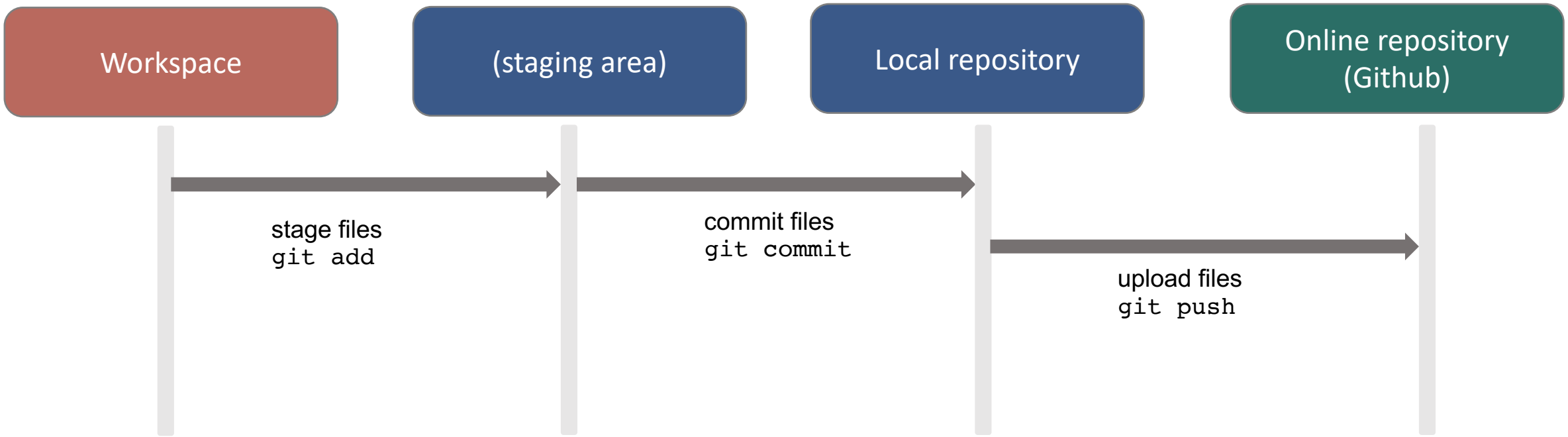


Github

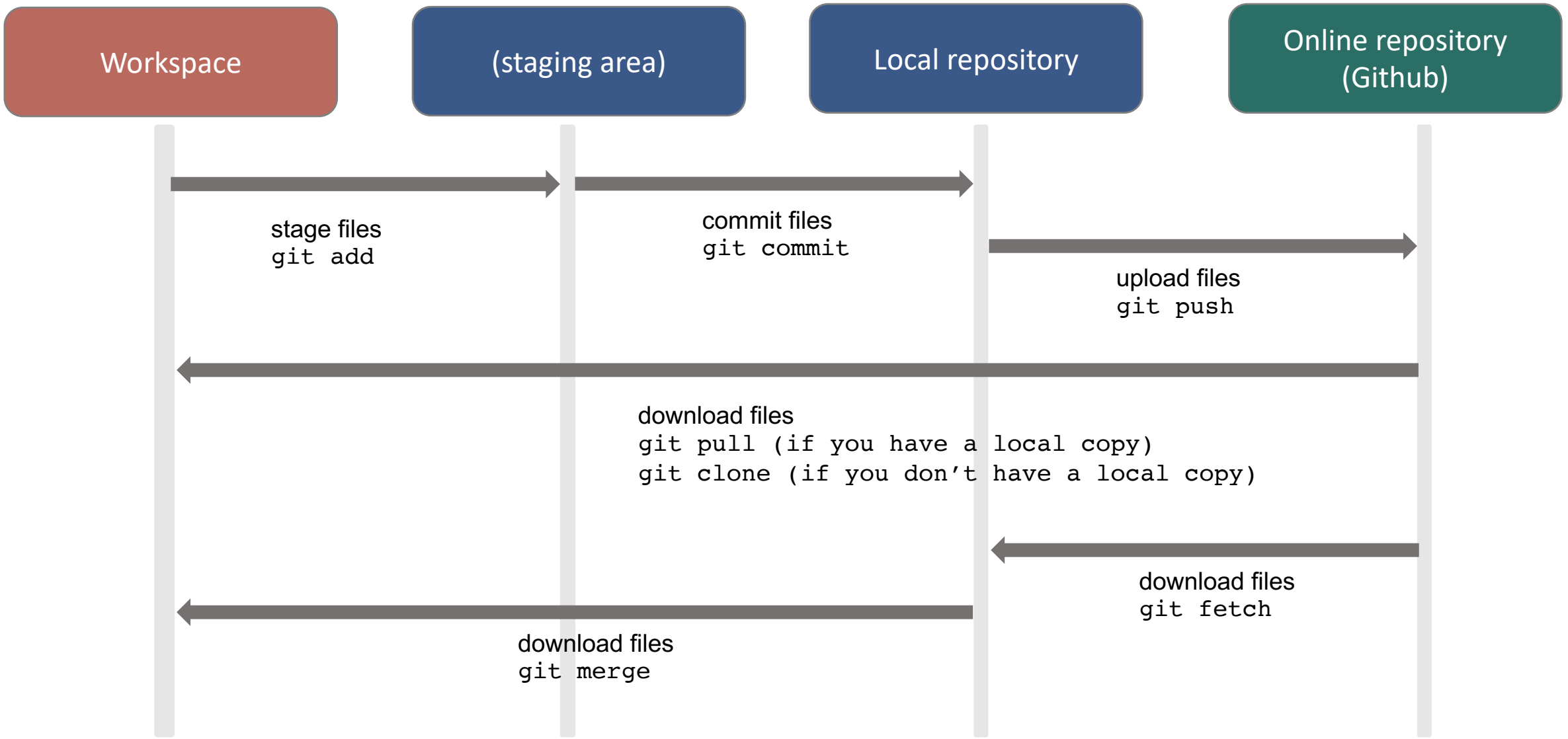
- To facilitate collaboration/sharing, need to upload what you've done locally to a **remote repository**
  - A remote is a version of your project hosted online, which others have access to
  - Think of your remote as where your code now lives – like dropbox or onedrive but for code
  - Backs up all your work, and allows you to access it from anywhere



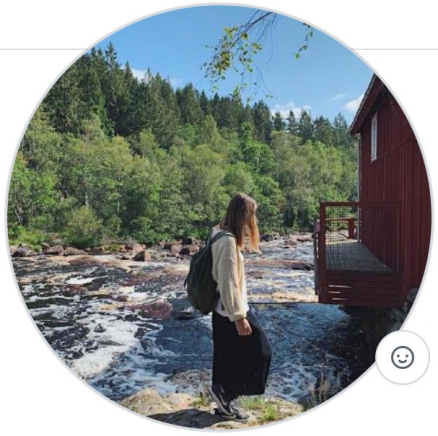
From local to remote



…and back again



People and organizations



Anna Schultze
annaschultze

Edit profile

1 follower · 1 following

LSHTM

ELECTRONIC HEALTH RECORDS RESEARCH GROUP

EHR Research Group @ LSHTM

3 followers · London, UK · https://ehr.lshtm.ac.uk · @ehr_lshtm · ehr@lshtm.ac.uk

Overview | Repositories 11 | Projects | Packages | Teams | People 24

Repositories

Find a repository... | Type | Language

github-training Private

Materials for an Introductory Github Training Worksh

0 0 0 0 Updated 2 days ago

People

View all

Invite someone

Git dictionary - Part 1

Git language	Everyday language
Repository (”repo”)	Project folder
Remote repo (“origin”)	Online copy of project folder
Commit	Save changes
Push	Upload changes to remote
Pull	Download changes from remote
Clone	Download remote for the first time

Collaborative Coding

Collaborative Coding

- Imagine there's two of you working on the same project, and there's a number of tasks that needs to get done:
  - edit data management files
  - make table 1 and a nice plot
- You could decide to just divvy it up and save your programs to some kind of online folder
  - Problems?

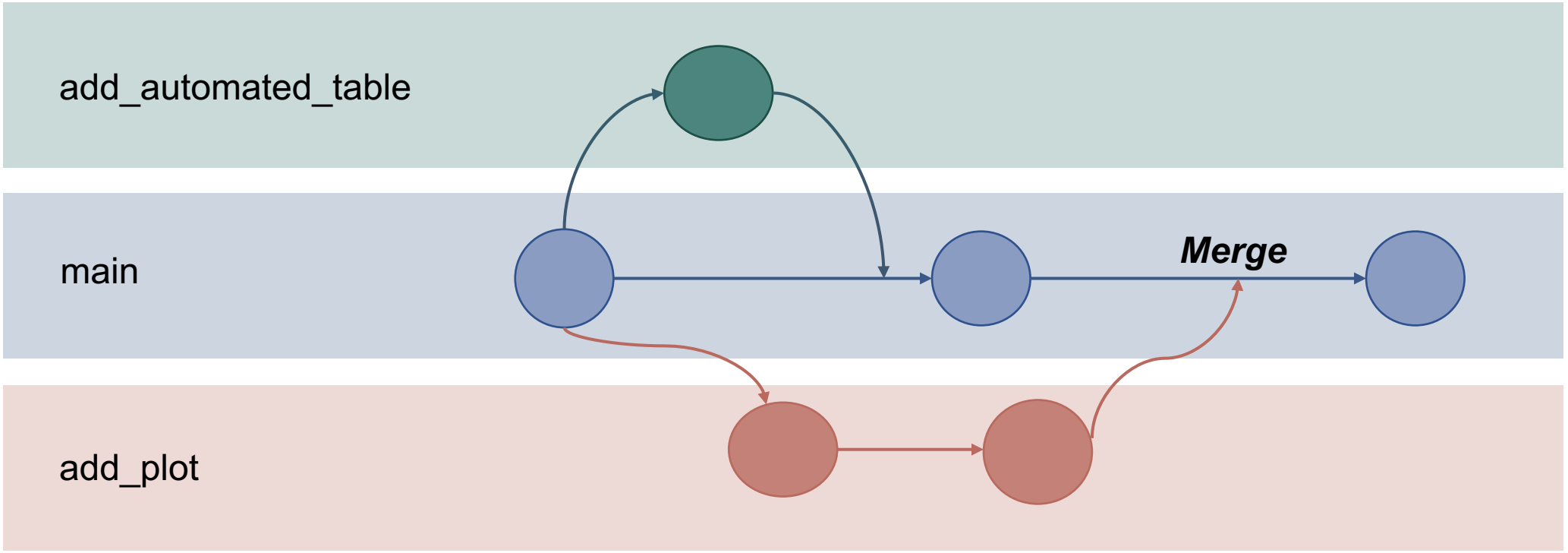
1) Might accidentally overwrite each others work, say both do some changes to the data management code at the same time
2) Difficult to highlight changes or updates, say you the data management file is 500 lines and you change 2. How do you easily flag what's changed?
3) Difficult to review, if you add comments the line numbers get pushed down, but in a separate file you have to manually match the comment to the code line.

Collaborating with Git

Git automatically creates a “main” version of your project

You can use ***branches*** to enable multiple people to work on a single project at the same time

Branches and Merges



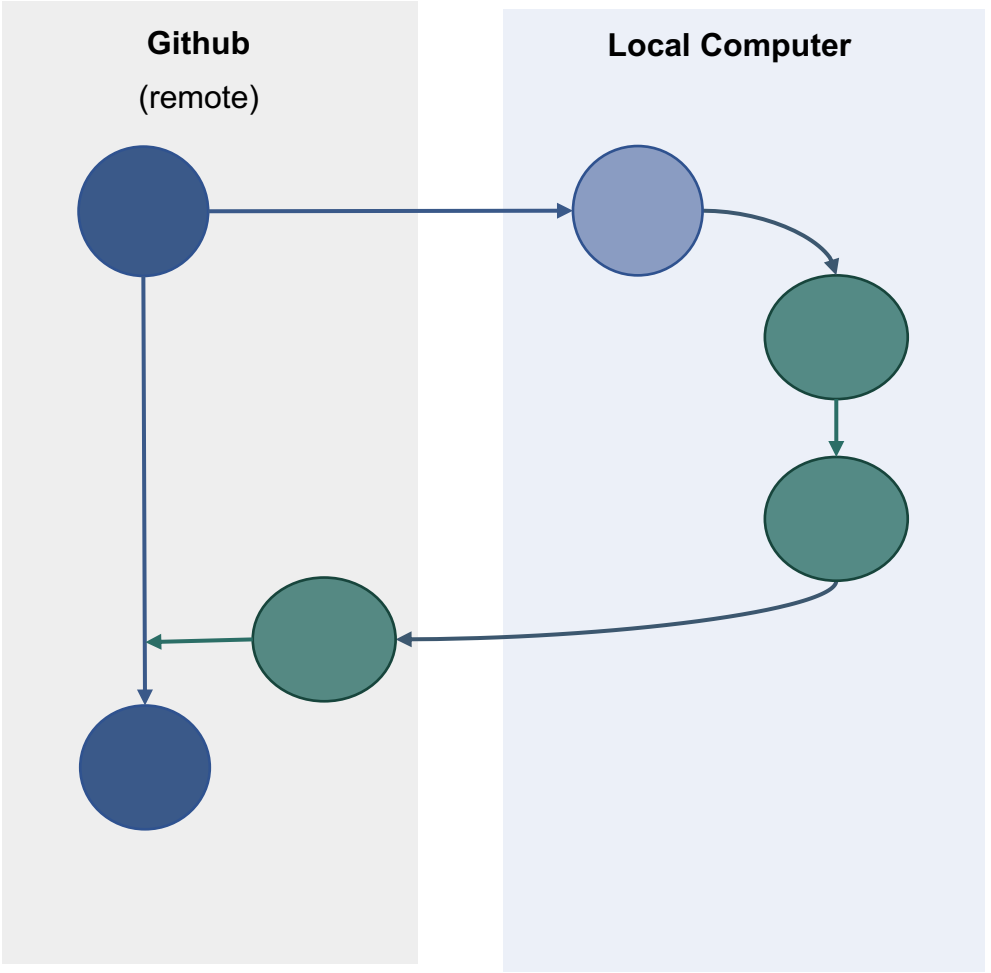
- **Branches** are parallel versions of your repository
- Allows you to try out new changes to your code before changing the main codebase
- Useful for collaboration as several people can work on different branches at once

Image source: https://medium.com/@natetadesse4991/git-branches-and-merging-overview-17810959c28a

The “Feature Branch” Workflow

- All new code is added through branches
- Each branch contains a new “feature”
    - So instead of having the “anna” branch, have “make_table1” and “run_models”

1. **Clone** the repository if it’s new, or **pull** down changes to ensure you’re working on the most recent version
2. Create a new branch
3. Develop your code on this new branch
4. **Push** changes to the remote
5. Request that your changes are merged into the main through a **pull request**
6. Once approved, changes are **merged** into main



Pull Requests and Code Review

- A pull request (PR) is a request for new code to be merged into the main branch
  - “open a PR” → suggest changes to the main code
  - a natural time for **code review**

EDITOR'S CHOICE

Code Review as a Simple Trick to Enhance Reproducibility, Accelerate Learning, and Improve the Quality of Your Team’s Research FREE

Anusha M Vable ✉, Scott F Diehl, M Maria Glymour

American Journal of Epidemiology, Volume 190, Issue 10, October 2021, Pages 2172–2177,

Walking through a pull request

fix typo to include emergency only codes #33

Edit | Code

Merged **annaschultze** merged 2 commits into `main` from `fix_type` on Nov 11, 2021

Conversation 1 | Commits 2 | Checks 1 | Files changed 2

annaschultze commented on Nov 10, 2021 — Member

No description provided.

`fix typo to include emergency only codes` ✓ `3d2d3d6`

annaschultze requested a review from **johntaz** last year

`fix typo in yaml to ensure denominators file is created` ✓ `82b40d8`

johntaz approved these changes on Nov 10, 2021 — View changes

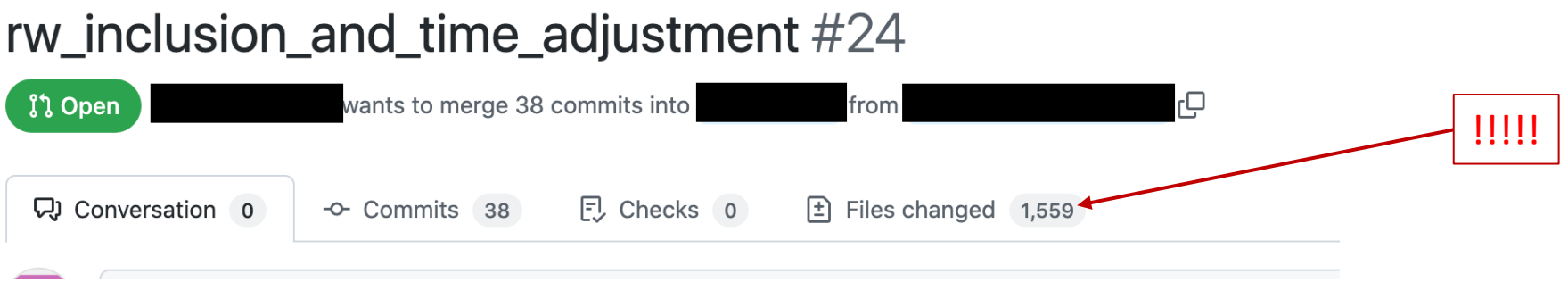
annaschultze merged commit **`59a8b06`** into `main` on Nov 11, 2021 — View details | Revert

2 checks passed

```
291 +        14 "GP and hospital and death and emergency" 15 "Eme
292 292
```

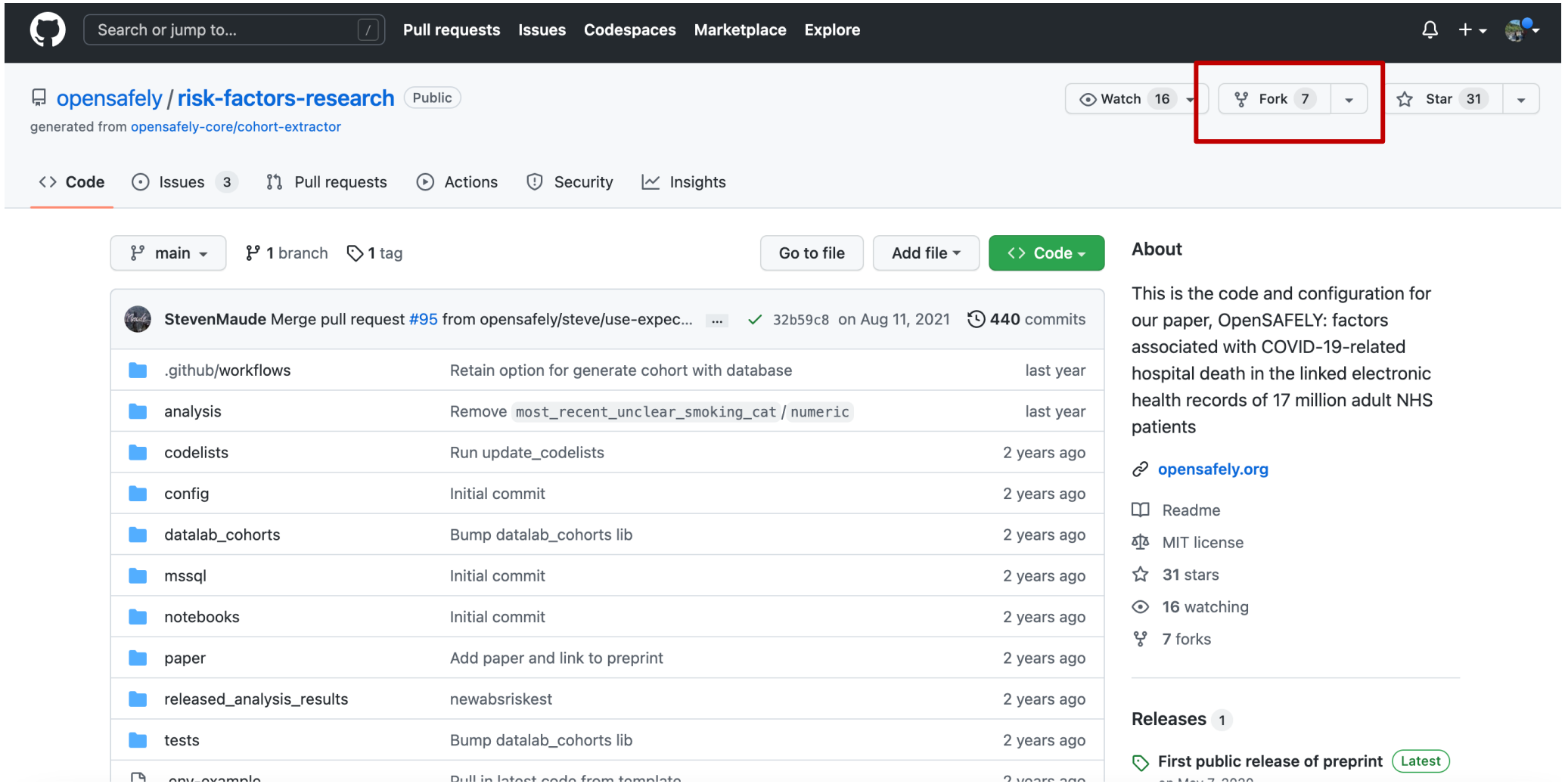
Best Practices

- Ideally, make your pull request small and quite well-defined
  - 10+ commits and/or files changed would be a very large PR



- It’s easiest if you create one branch for each “thing” that you’re trying to do, and request a review once that ”thing” is done
  - create_codelists → review → merge
  - create_variables → review → merge

What the Fork?



Forks for contributing to projects

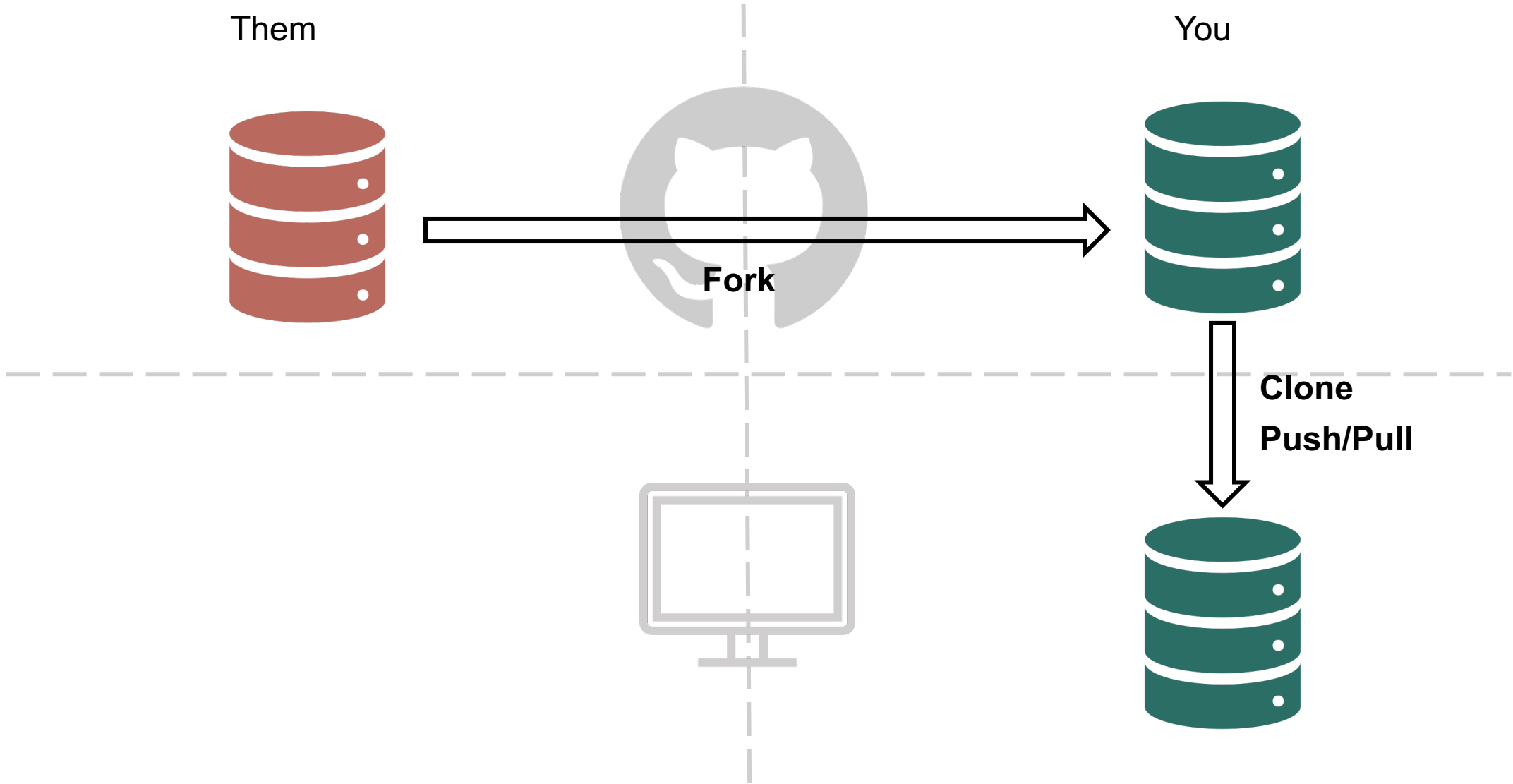


Image source: https://medium.com/swlh/forks-and-pull-requests-how-to-contribute-to-github-repos-8843fac34ce8

Git dictionary - Part 2

Git language	Everyday language
Main (‘master’)	Original or core project (‘master version’)
Branch	Copy of a specific version of a repo, usually the main. Typically used to create a local copy of your online project to allow feature development
Merge	Combine two branches (usually a new feature branch and main)
Pull request	Suggest changes to the code on a certain branch (usually the main)
Fork	Copy of an entire repo. Typically used to get a copy of an organisations’ repository onto your own Github profile, so that you can make changes without bothering the other people.

How?

Workflow for using git on your own

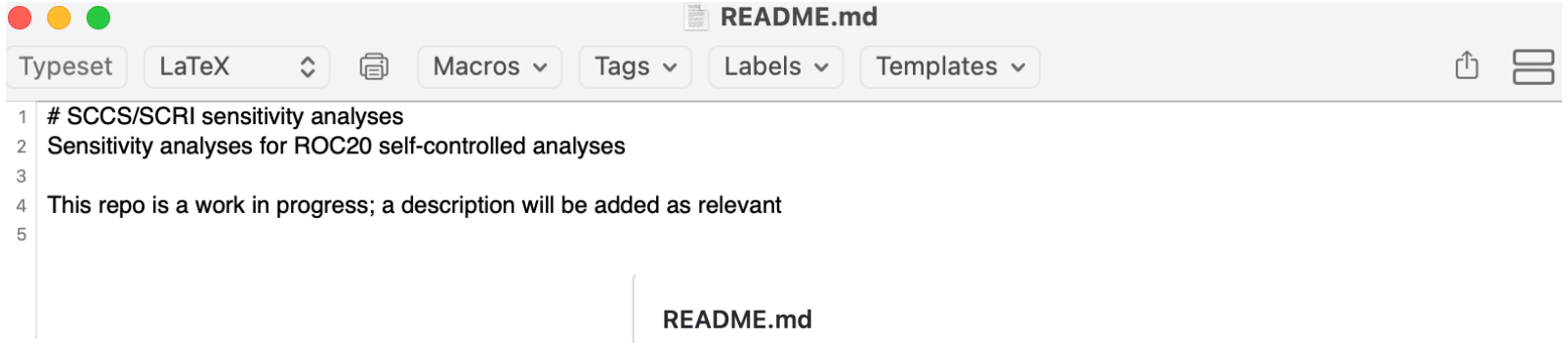
1. Set up a new repository
2. Clone the repository
3. Do some changes and commit them
4. Push the changes to the remote

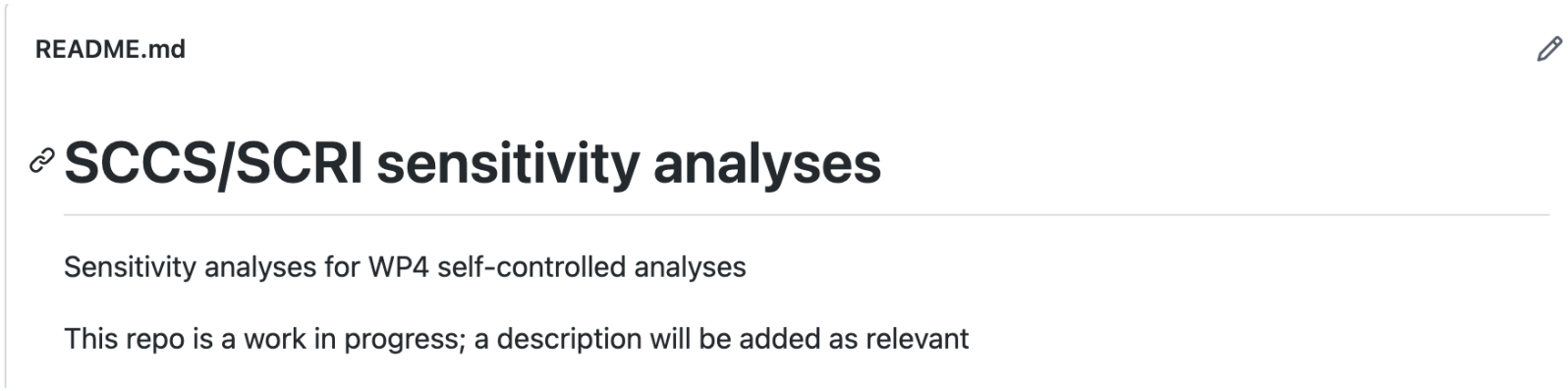
Workflow for using git with someone

1. Set up a new repository (or fork an existing one)
2. Clone the repository
3. Checkout a new branch
4. Do some changes and commit them to the new branch
5. Push the changes to the remote
6. Open a PR
7. Ask for review of your code
8. Merge the PR into your main branch
9. **On your computer, switch back to your main branch, and download the changes before continuing to work (IMPORTANT)**

READMEs

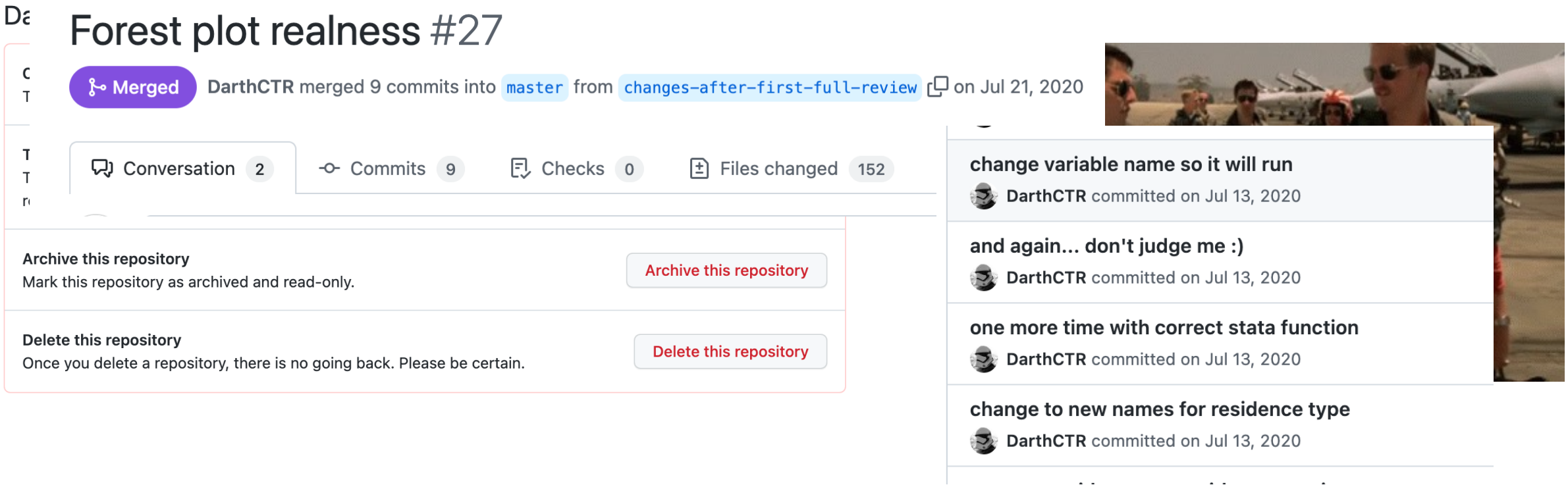
- When setting up a new repository you'll be asked if you want to start with a README
  - The answer is always yes
  - A README is just a description, and forms the landing page for your repo
  - Github will format ('render') these automatically so they looks nice





Public or Private?

- You might feel most comfortable working in a private repository
- You can change visibility at a later stage
  - BUT – remember this will make all your project history is visible
  - This includes all comments and commit messages



BEWARE of uploading data

- CPRD have strict data sharing rules
  - You cannot upload CPRD data to Github, and it's not recommended to upload log-files
  - Technically, nothing with <5 counts should leave secure servers
    - this includes in comments ("Dropping one GBS case with superhigh BMI, data entry error?")
  - PHE have specific extra restrictions around sharing filepaths
    - but unlikely to actually be a major security risk
- Once something is on Github, it's not easy to get it down (because all history is saved)

To prevent accidental data leaks

- Use a **.gitignore file**
  - A text file which tells git to ignore certain types of files (such as csvs, or all files in a data/ folder)
- Store data separate from code
- Store output separate from code
- Store filepaths in a sepa
- Have your repository set

.gitignore.txt

```
# .gitignore file for git projects containing Stata files
# Commercial statistical software: http://www.stata.com

# data folder

data/*

# Stata dataset and output files
*.dta
*.gph
*.log
*.smcl
*.stpr
*.stsem

# Graphic export files from Stata
# Stata command graph export: http://www.stata.com/manuals14/g-2graphexport.pdf
#
# You may add graphic export files to your .gitignore. However you should be
# aware that this will exclude all image files from this main directory
# and subdirectories.
# *.ps
# *.eps
# *.wmf
# *.emf
# *.pdf
# *.png
# *.tif
```

The EHR “Organization” (and our helpful templates)

EHR Research Group @ LSHTM

London, UK https://ehr.lshtm.ac.uk @ehr_lshtm ehr@lshtm.ac.uk

Overview Repositories 9 Projects Packages

Repositories

template-r Private template

0 0 0 0 Updated on 21 Feb

template-stata Private template

0 0 0 0 Updated on 9 Feb

README.md

EHR group template for Stata

Purpose

A template to begin a project with Stata. Click here to use this template, then replace this text with a desription of your project.

Untracked files

By default, `.csv`, Stata output, and all files in the `output/` and `paths/` folders (except README) are untracked, i.e. they will not be uploaded to GitHub. Edit `.gitignore` to ignore or allow (with `!`) specific files or file types.

File tree

```
template-stata/
├── data/
│   ├── README.md
│   ├── raw_data_1.csv
│   └── raw_data_2.csv
├── docs/
│   ├── README.md
│   ├── document1.docx
│   ├── document1.html
│   └── document1.Rmd
├── paths/
│   ├── README.md
│   └── paths.do (untracked)
```

If something goes wrong…

- It’s likely it won’t – it’s quite hard to actually get something onto Github
- Don’t panic
- Same procedures as for any data leak
  - Make repo private if public
  - Flag to your manager/supervisor
  - Post on git slack channel
  - Just deleting the file is typically not enough

Trying it all out!

Before we start

1. Do you have github desktop installed?
2. Do you have a github username?
3. Are you a member of the EHR organization?

Practical Tasks…

Two tasks:

1. Start a new project and add a file (‘working on your own’)
2. Find the mistake in one of Annas files, correct it, and submit the solution (‘collaborative coding’)
    - you can choose to work in Stata or R - will need either installed

Download the instructions from Github: https://github.com/ehr-lshtm/github-training