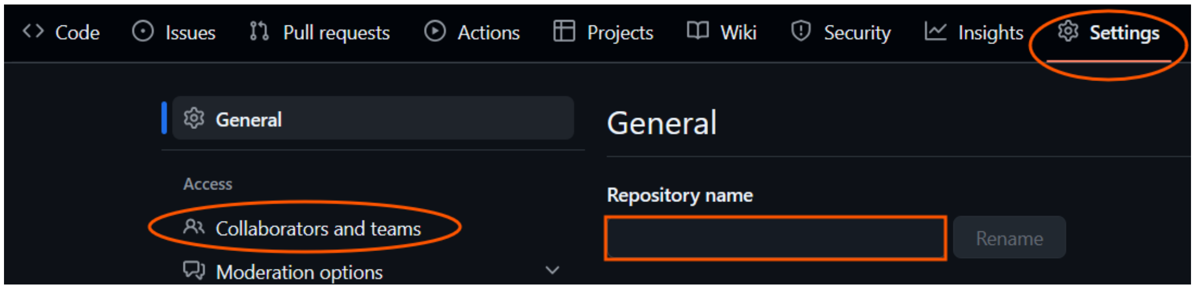
Collaborative Git practice

Follow these steps in your groups of three. Choose one person to be the "Owner" of the repo. That person will create the `GitHub.com` repository on their own personal `GitHub.com` account, and add the other two as collaborators before anyone starts coding.

Create and examine the repo

Choose a repo "Owner". The Owner should clone the repo and add the rest of the team as collaborators

1. Fork the repo `https://github.com/DDI-Talent/collaborative-git-tutorial-rd-2025.git` and then clone it using the familiar workflow - at the end of the process, you should have an R project called collaborative-git-tutorial-rd-2025.
2. Add your members as collaborators on GitHub.com:
    - Settings > Manage access > Invite collaborators > enter GitHub usernames > Invite.



3. Collaborators: you will get an email notification to accept the invitation.
4. Collaborators: Accept the invitation and clone the repo
    - i.e. start a new RStudio project from Version Control using the repo URL.

As a group, investigate the repo. Start by taking a look at the git graph/history. It's very simple, and there is nothing in particular you should be looking for in this case.

Examine the three R scripts in the `scripts/` folder. These are also very simple scripts - each has a small bug: a misnamed variable in a subset or plot call.

The process going forward will be that each person will **work** on a **single specific script**, and **review** the changes made in **one other** script.

Work on only your designated script:

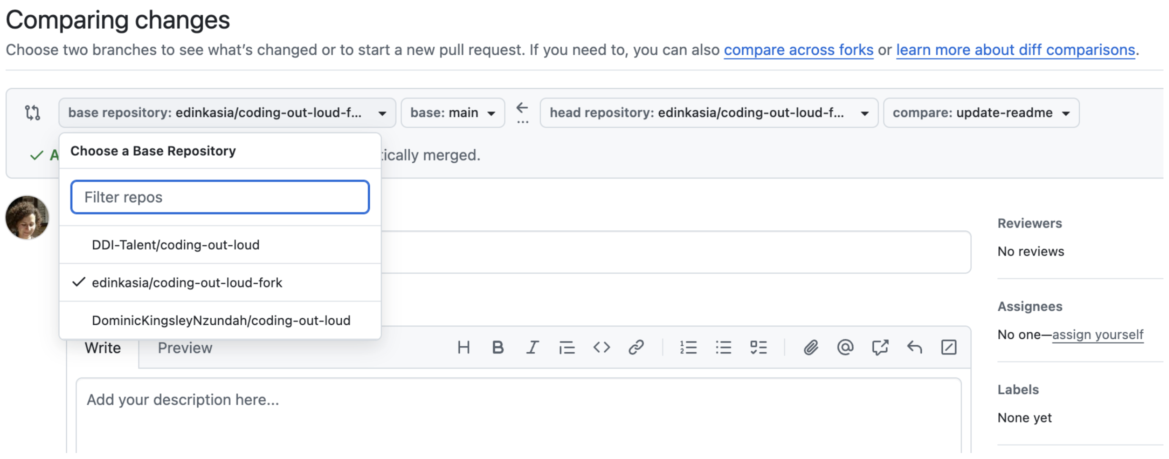
- Team member 1: `scripts/patient_care.R`
- Team member 2: `scripts/patient_experience.R`
- Team member 3: `scripts/who2.R`

There is also a script called `conflict.R`. Ignore it for now.

Making your fix

The aim is for you to fix your bug locally, commit, push, and create pull request (PR).

1. Create a new branch named for your fix
   - perhaps include your name and the dataset name to make it clear who pushed, and who is responsible for the PR review.
2. Confirm you are on your new brach (i.e. not `main`) and open only your assigned script.
   - Locate the bug and correct it so the R code runs as expected.
   - **Do not edit other scripts.**
3. Stage and commit your change, and push
4. On `GitHub.com` create a **Pull Request** from your branch into main. Make sure that you are working with the forked repo, and not the original one (you can pick the correct repo in the dropdown menu). Set the PR title to something short and informative, and give a brief description of the fix. You can be more detailed here.


Comparing changes
Choose two branches to see what's changed or to start a new pull request. If you need to, you can also compare across forks or learn more about diff comparisons.
base repository: edinkasia/coding-out-loud-f...
base: main
head repository: edinkasia/coding-out-loud-f...
compare: update-readme
Choose a Base Repository
Filter repos
DDI-Talent/coding-out-loud
✓ edinkasia/coding-out-loud-fork
DominicKingsleyNzundah/coding-out-loud
ically merged.
Write
Preview
Add your description here...
Reviewers
No reviews
Assignees
No one—assign yourself
Labels
None yet

5. Assign the PR to the appropriate reviewer.

Reviewing

Be sure you each review a different PR!

- Student 1 reviews Student 2's PR.
- Student 2 reviews Student 3's PR.
- Student 3 reviews Student 1's PR.

1. To carry out the review, you can look at the "Files changed" tab in the PR on `GitHub.com`.
2. However, **you will not be able to run the code** directly on `GitHub.com`. So, to test the fix, you should pull the branch locally to your machine.

to pull the branch locally in RStudio:

- In RStudio, open to the Git pane.
- Click on the Branch button, then select the branch you want to review from the list.
- If the branch is not listed, you should be able to update the branch list by pressing the `Pull` button first.

If you are happy that the fix is correct, you can go ahead and merge the PR on `Github.com`. Otherwise, you would leave comments for the author to address.

Review checklist for reviewers:

- Confine changes to the assigned file only.
- Confirm the commit message is descriptive.
- Optionally run the script locally to ensure the plot renders (R and dependencies installed).

After merges: sync your local main

1. In RStudio, switch to main and pull the merged changes:

Additional exercises

- Repeat the above
  - Create a new branch, make a small improvement to your assigned script (formatting, comments, small improvement), push, PR, and follow the same flow.
- Try undoing something
  - Use the `revert` button to undo changes to a file
  - Add a new commit, the use `git reset HEAD~1` **in the terminal** to drop it.
- Do not (ever) attempt to drop a commit that has already been pushed!

Creating and resolving conflicts

You will now create a merge conflict in a controlled environment. To do this, two of you will independently make changes in the script `conflict.R`, on two separate branches.

- Team member 1: Pull the `main` branch of your repo, to make sure that your local version is up to date. Then, create a new branch and add your favourite fruit to the `fruits` vector. Commit and push the change.
- Team member 2: Pull the `main` branch of your repo, to make sure that your local version is up to date. Then, create a new branch and add your favourite fruit to the `fruits` vector. Commit and push the change.
- Team member 1: Create a pull request to merge your branch with `main`. This should not create a conflict.
- Team member 2: Create a pull request to merge your branch with `main`. This will create a conflict because `main` has changed since you last pulled from it. It's a simple conflict and it should be possible to resolve it within the editor on `GitHub.com`. Try it out.

Notes and tips

- Work only on your assigned file to avoid merge conflicts.
- Use descriptive branch names and commit messages.
- If a merge conflict occurs, communicate with the group and resolve it locally, then push the resolved branch.
- If you need to add or update dependencies, mention them in the PR description.