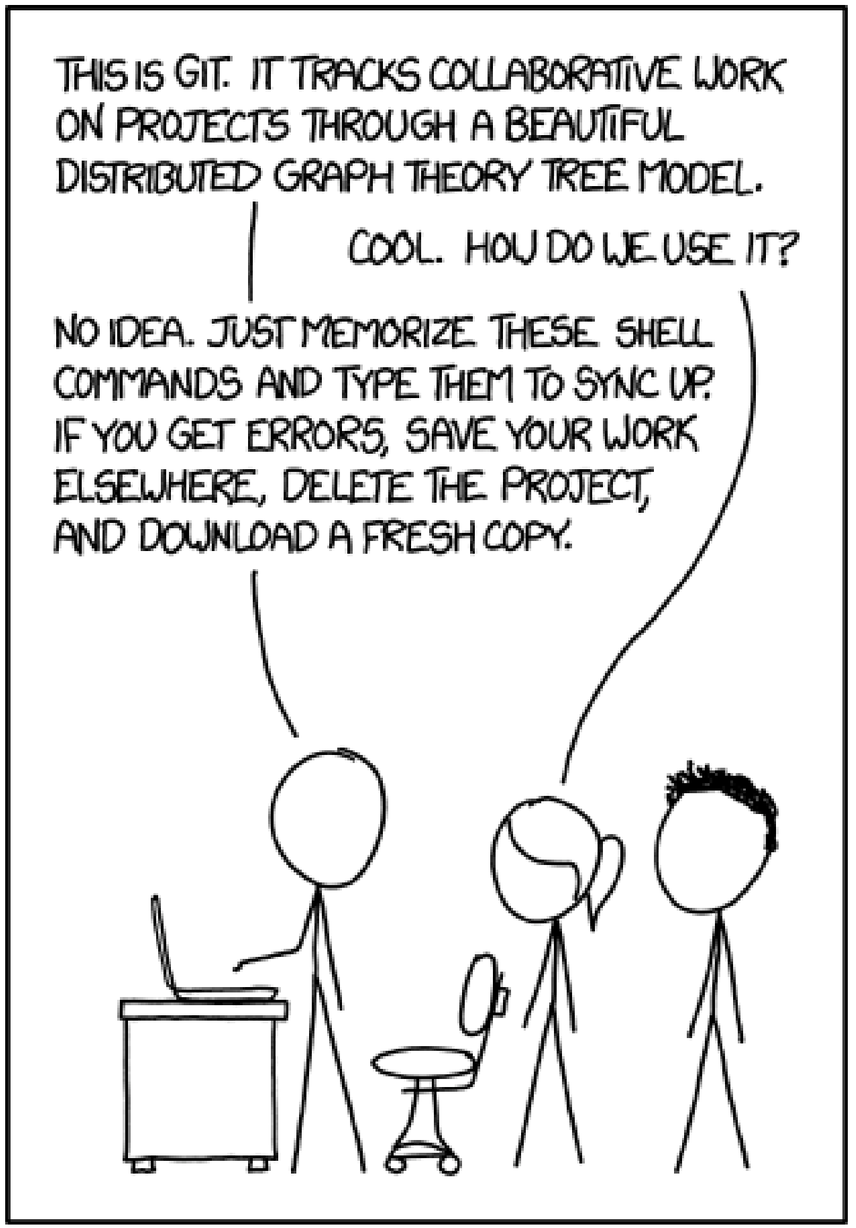
Getting Started with Git and GitHub pt. 2

Coffee, Cookie and Coding (C^3)

Workshops are by the Public Health Data Science and Data Equity team

Shelby Golden, M.S.

April 28th, 2025



xkcd Comic 1597 - Git Downloaded April 26th, 2025.

Yale SCHOOL OF PUBLIC HEALTH
Data Science and Data Equity

Shelby Golden, M.S.

- Worked 7 years as a Molecular Biologist and Biochemist.
- Received a Masters in Applied Computational Mathematics from Johns Hopkins University in 2024.

Today's Learning Objectives

01 Understand why Git and GitHub are valuable tools for version control and managing coding projects. (~ 5 minutes)

02 Get hands-on experience using Git and GitHub for solo projects through a worked through example showing common workflows. (~ 25 minutes)

03 Learn how to use GitHub to support collaboration and teamwork on group projects. (~ 20 minutes)

Our Choice Resources

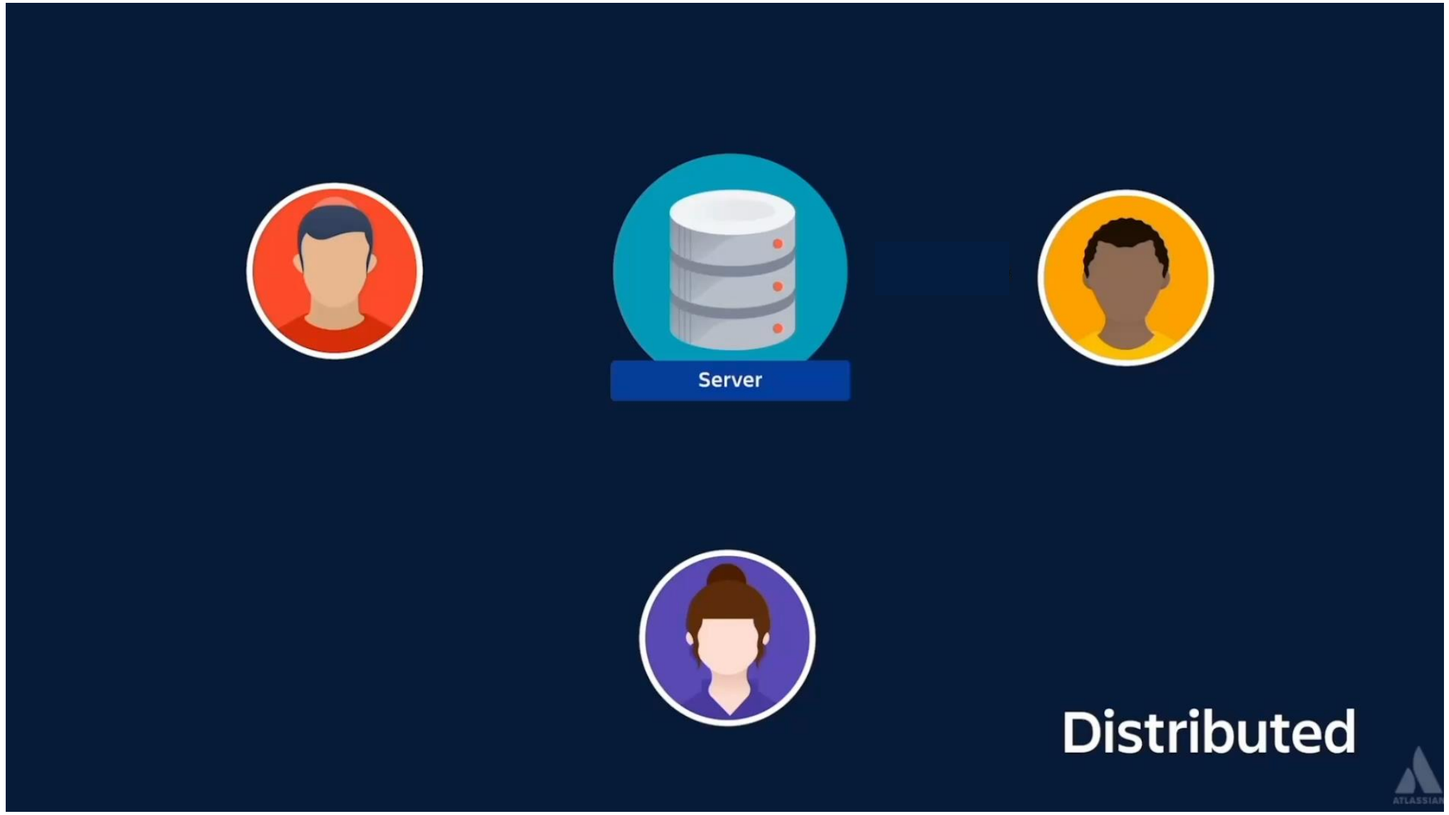
- Yale's Center for Research Computing workshop Version Control by Git by Kaylea Nelson
- Yale's Harvey Cushing/John Hay Whitney Medical Library workshop Git & GitHub: An Introduction To Version Control by Justin DeMayo
- Getting Git Right by Atlassian
- Git and GitHub Tutorial by W3 Schools
- Introduction to GitHub by GitHub
- Happy Git and GitHub for user by Jenny Bryan

Going Beyond Basic Git/GitHub

- Reviewing the developer documentation: git-scm.com/docs and docs.github.com
- What is git commit, push, pull, log, aliases, fetch, config & clone by Amit Prajapati
- Git Guides by various Graphite contributors
- How to Write a Git Commit Message by cbeams
- Git Graphical User Interface (GUI) Clients by various contributors

What is Git and GitHub?

How do they relate to one another?

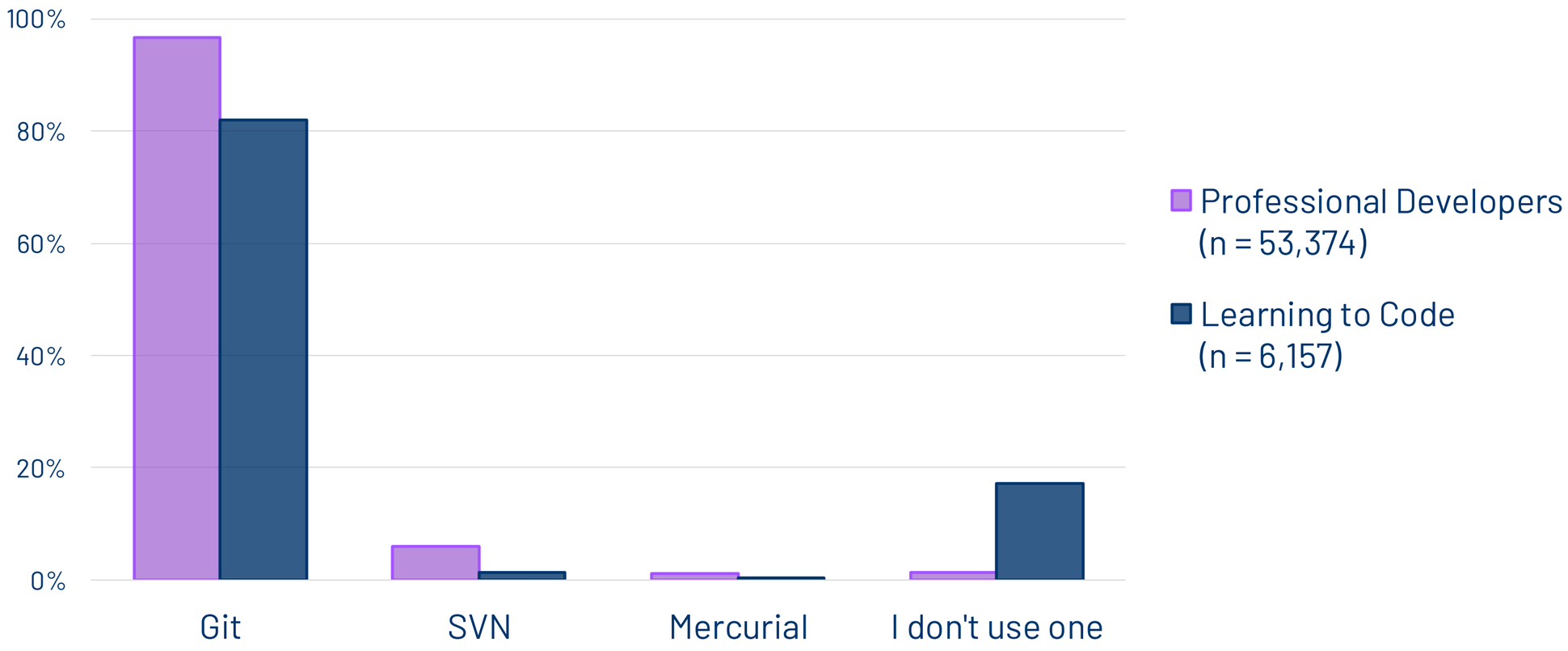


What is Version Control by Atlassian. Updated February 23rd, 2020.
Git logo. Downloaded October 10th, 2024.
GitHub logo. Downloaded October 10th, 2024.

System for project management by distributive version control (DVCS).

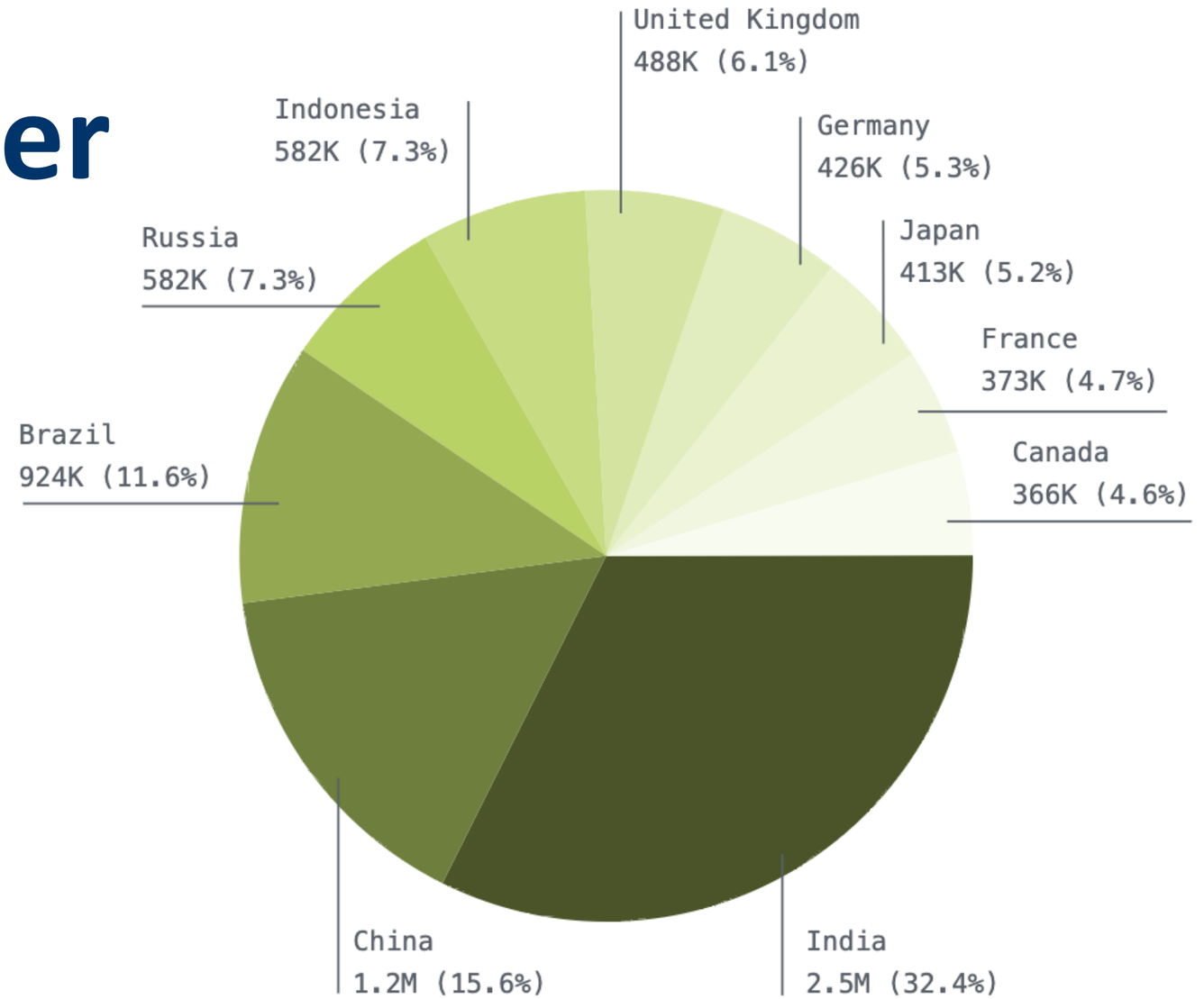
Developer platform for housing and managing projects and acts as the DVCS server.

Git is the most widely used VCS but is underutilized by those learning to code.



2022 Developer Survey by StackOverflow. Published 2022.

Total New Developers per Country



[A global community of developers.](#) by GitHub, Octoverse 2022 report.

Local Git Overview

Please access the code:

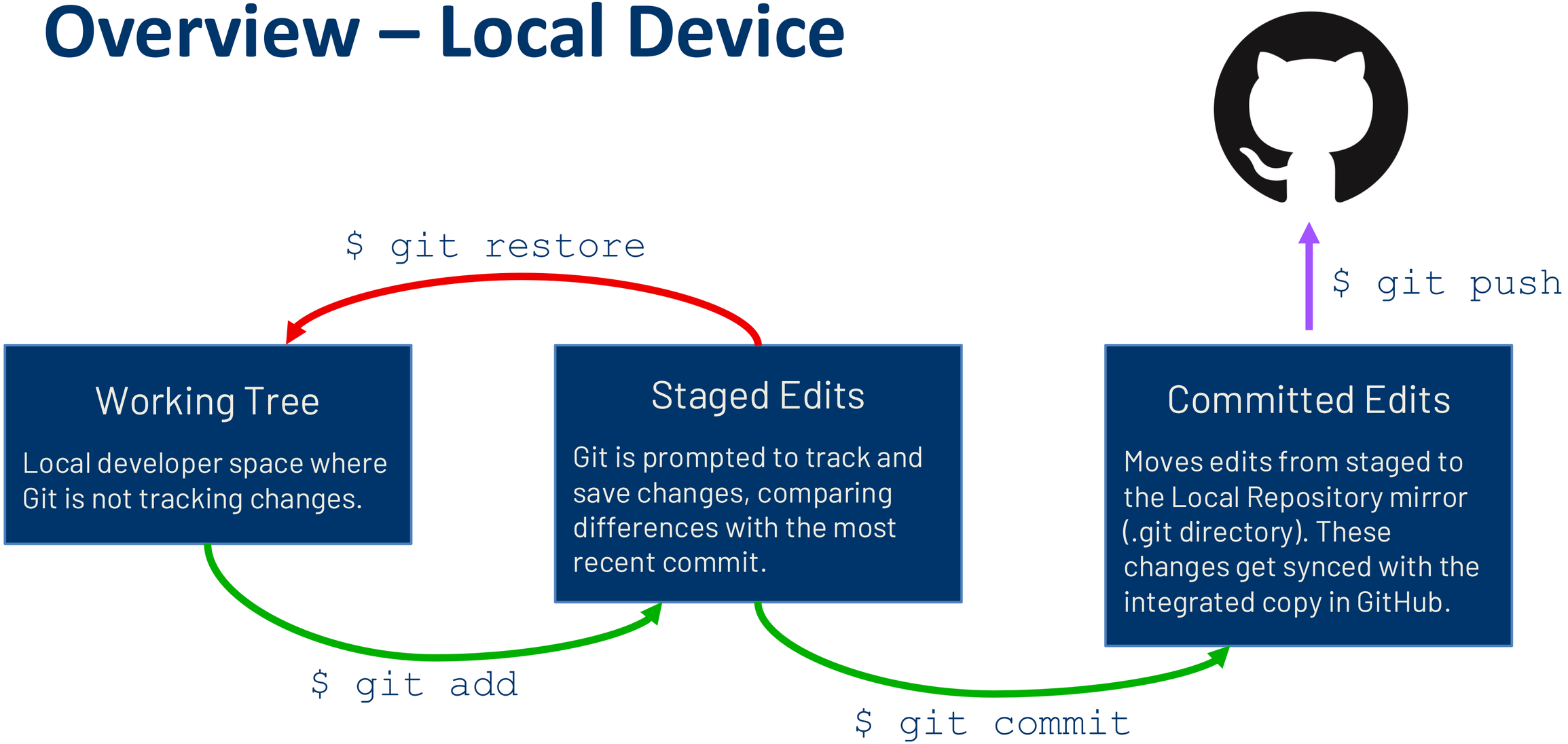
1. Open the workshop webpage:

   https://ysph-dsde.github.io/Book-of-Workshops/Git-and-GitHub/

2. Follow the steps under “Codespaces” and “Making a Clean-Break Copy” at the top.

   - Everyone import the Vaccinations repo.
   - Choose one person in the group to import the Cases and Deaths.

Overview – Local Device



CLONE

- A one-time operation that creates a local working copy of the remote repository.
- Used when first creating a mirror from a remote repository.

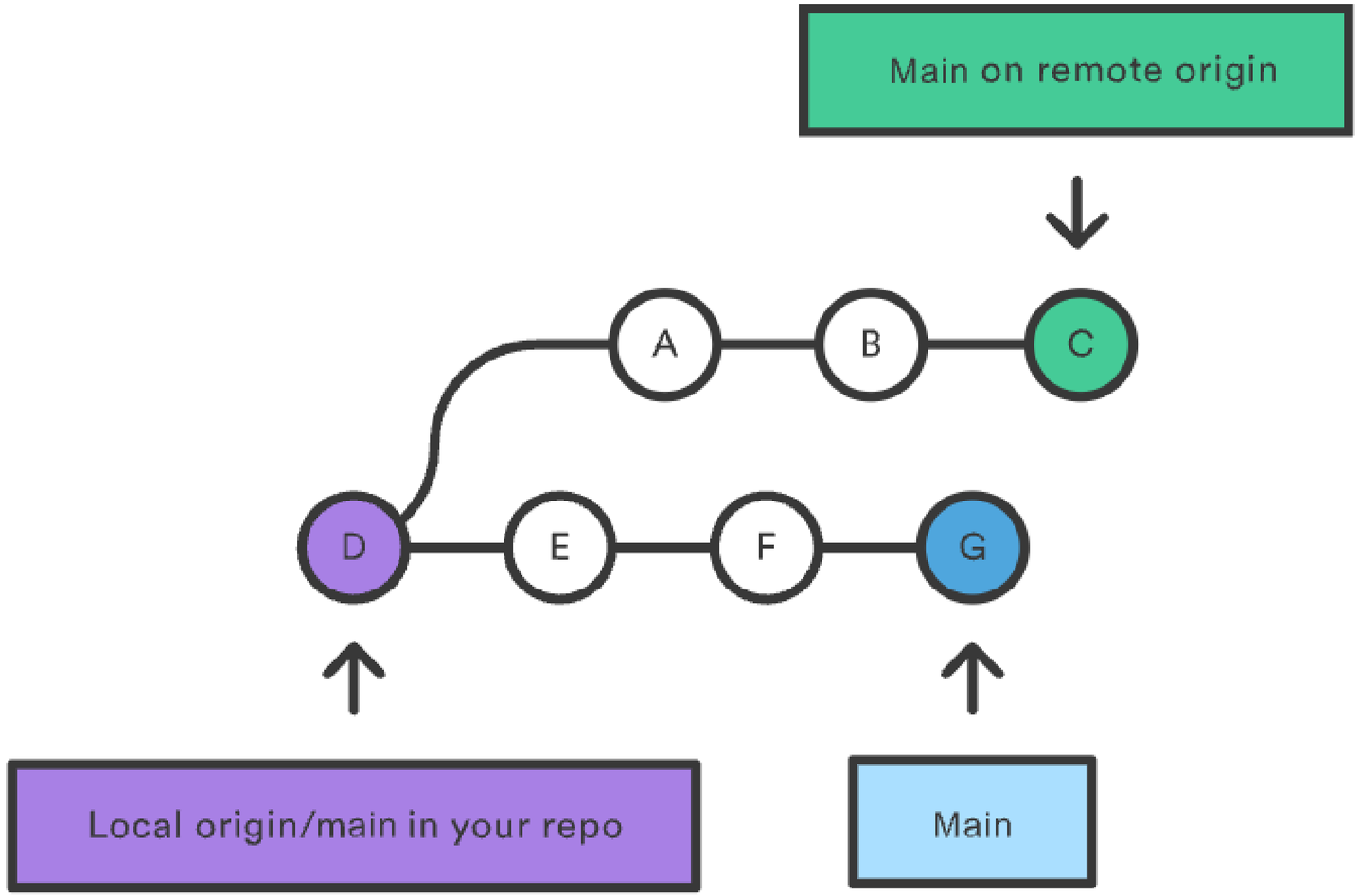
```
$ git clone <repo>
```

PULL

- Combines the action of "`fetch`" and either "`merge`" or "`rebase`".
- Best practice is to pull before you push your committed changes.

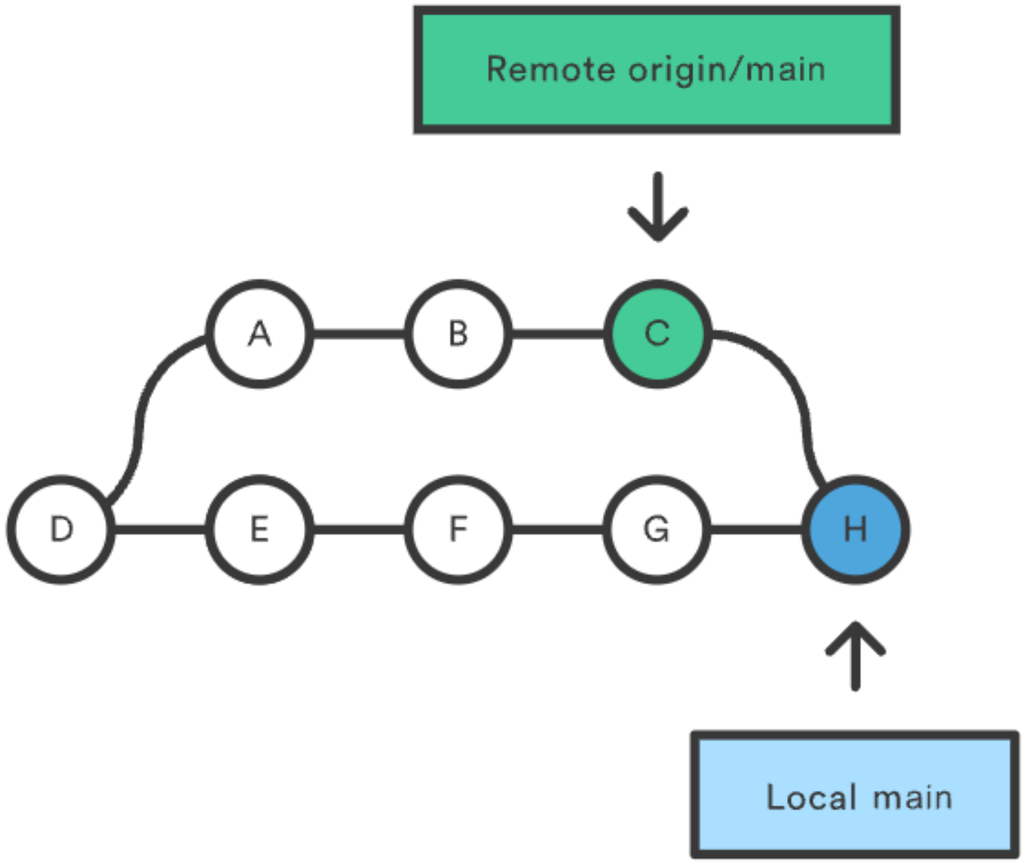
```
$ git pull <repo>
```

Consider the example…

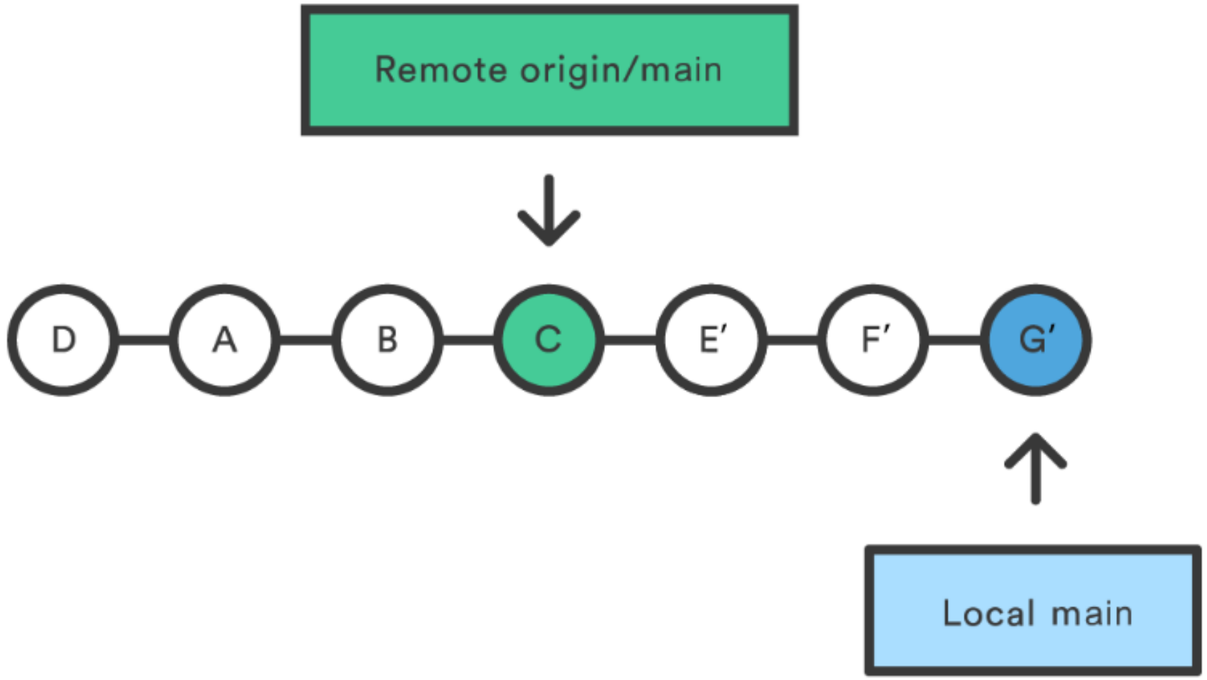


git pull from Atlassian's Git tutorials, Downloaded April 24th, 2025.

git merge



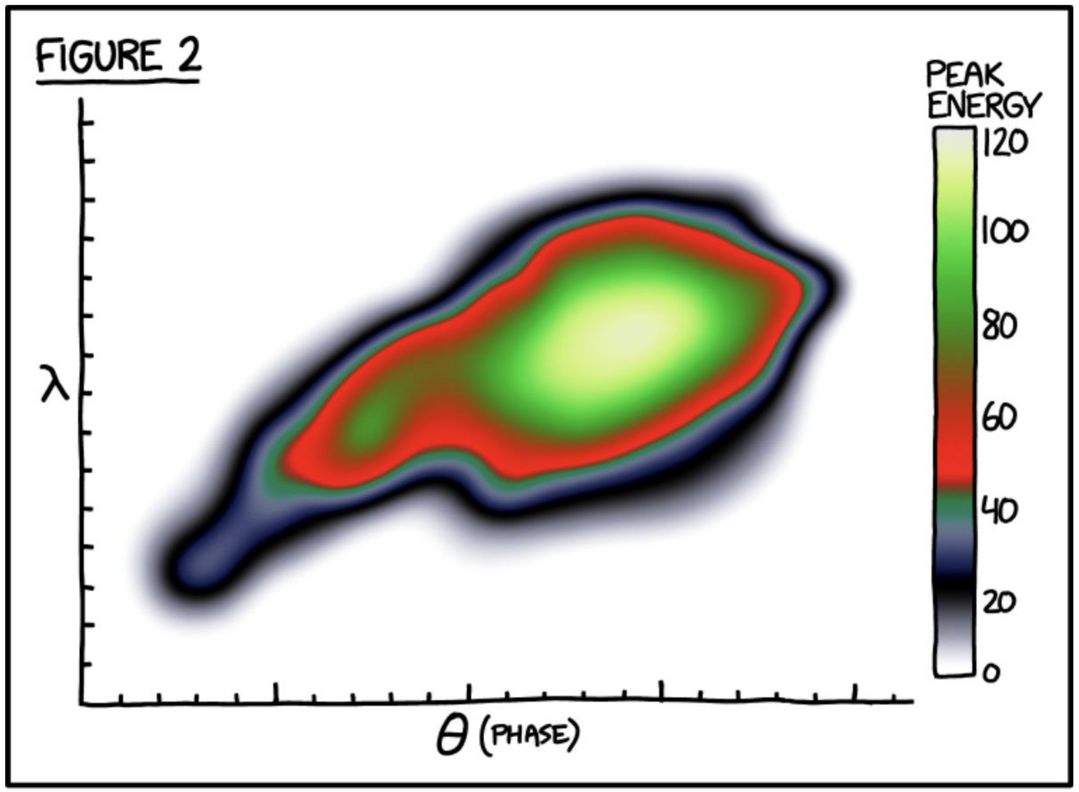
git rebase



[git pull](#) from Atlassian's Git tutorials, Downloaded April 24th, 2025.

Detailed Walk Through

Painbow color scale



EVERY YEAR, DISGRUNTLED SCIENTISTS COMPETE FOR THE PAINBOW AWARD FOR WORST COLOR SCALE.

xkcd Comic 2537 - Git Downloaded from, Flowing Data April 26th, 2025.

Command-Line Application

```
git status
```

Command-Line Output

```
On branch main
Your branch is up to date with 'origin/main'.

Changes not staged for commit:
    (use "git add/rm ..." to update what will be committed)
    (use "git restore ..." to discard changes in working directory)

       modified:    making_cool_plots.R

Untracked files:
    (use "git add ..." to include in what will be committed)

       earth_shattering_color_scheme.png
```

Command-Line Application

```
# OPTION #1: List each file
git add "making_cool_plots.R" "earth_shattering_color_scheme.png"

# OPTION #2: Use the wild card "." to add all files
git add .

# View the results of git add.
git status
```

Command-Line Output

```
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
   (use "git restore --staged ..." to unstage)

      modified:    making_cool_plots.R
      new file:    earth_shattering_color_scheme.png
```

```
git commit -m "Revelatory message elucidating the hidden secrets of git."
```

Command-Line Application

```
git commit
```

Command-Line Output

```
# Please enter the commit message for your changes. Lines starting
# with '#' will be ignored, and an empty message aborts the commit."
#
# On branch main
# Your branch is ahead of 'origin/main' by 1 commit.
#    (use "git push" to publish your local commits)
#
# Changes to be committed:
#       modified:    making_cool_plots.R
#       new file:    earth_shattering_color_scheme.png
```

Command-Line Output

```
Outstanding progress on color schemes for density plot fill scaling.
# Please enter the commit message for your changes. Lines starting
# with '#' will be ignored, and an empty message aborts the commit."
#
# On branch main
# Your branch is ahead of 'origin/main' by 1 commit.
#    (use "git push" to publish your local commits)
#
# Changes to be committed:
#       modified:    making_cool_plots.R
#       new file:    earth_shattering_color_scheme.png
:wq
```

Command-Line Output

```
[main f9b4cf2] Outstanding progress on color schemes for density plot
fill scaling.
   2 files changed, 11 insertions (+), 2 deletions (-)
   create mode 100644 earth_shattering_color_scheme.png
```

Command-Line Application

```
# -----------
# Pull with one command.

# OPTION #1: Integrate the fetched copy of "origin/main" into "main" with merge.
git pull                          # Assuming the default protocol is a merge

# OPTION #2: Rebases "main" with the new parent history reflected in "origin/main".
git pull --rebase                 # Override the default merge to do a rebase
```

Command-Line Application

```
# -----------
# Two-step pull.

# Download branch main from the remote repository, origin.
git fetch origin main

# If needed, return to the local copy of main, not the fetched branch.
git checkout main

# OPTION #1: Integrate the fetched copy of "origin/main" into "main" with merge.
git merge FETCH_HEAD

# OPTION #2: Rebases "main" with the new parent history reflected in "origin/main".
git rebase FETCH_HEAD

# If needed, remove the fetched copy of "origin/main" saved as a branch.
git branch -d FETCH_HEAD
```

Command-Line Application

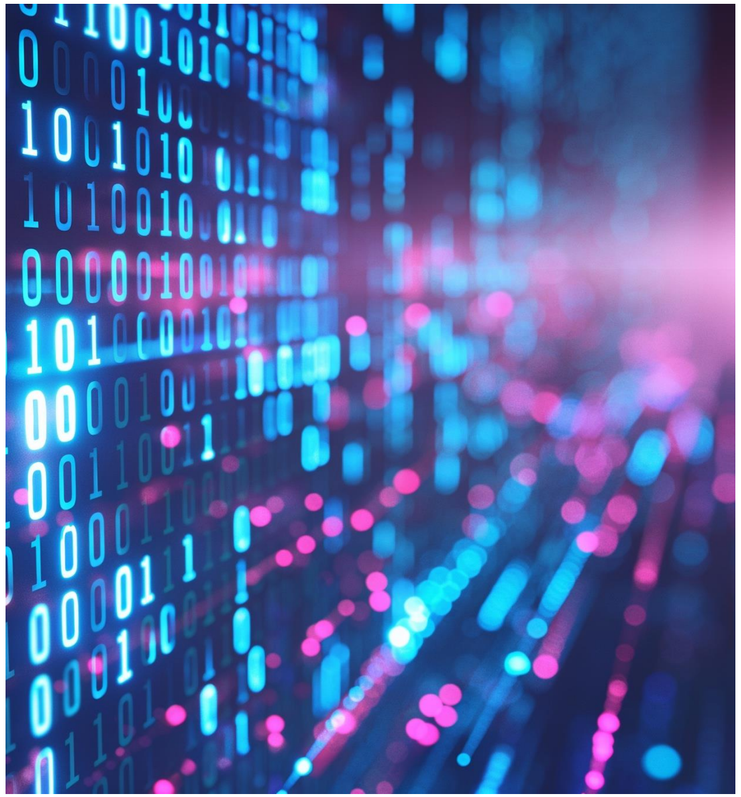
```
git push origin main
```

Command-Line Output

```
Enumerating objects: 14, done.
Counting objects: 100% (14/14), done.
Delta compression using up to 8 threads
Compressing objects: 100% (9/9), done.
Writing objects: 100% (9/9), 90.13 KiB | 22.53 MiB/s, done.
Total 9 (delta 5), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (5/5), completed with 5 local objects.
To github.com:ysph-dsde/PROJECT-REPOSITORY.git
   7d1b339..f9b4cf2 main -> main
```

Starting Individual Projects

Personal Repository



START LOCALLY

Initialize Git in a local project folder.

Push “Hello World” in a README.md file to an empty GitHub repository.

START REMOTELY

Clone the clean-break copy of the JHU-CRC-Vaccinations repository.

Make a bar plot and push to the remote repository.

REAL-WORLD EXAMPLE

Generate a new plot and run through the status through push commands.

Start Locally

1. Open Terminal and navigate to your destined directory.

```
$ cd “/file location”          # Project location
$ mkdir “First Repo”           # Make a project folder
$ cd “First Repo”              # Enter the project
```

2. Initialize Git in the project folder with a branch called “main”.

```
$ git init -b main
```

3. Create a new README.md file.

```
$ touch "README.md"
```

continued ...

Start Locally

4. Edit the newly created README.md file

```
$ vim README.md    # Opens the file for editing
```

5. In file edit mode hit “a” and type in “Hello World!!”. Hit Esc to exit file edit mode and type “:wq” to quit viewing the README.md contents in Terminal.
6. Stage our file for version control and commit changes.

```
$ git status                          # Show branch status
$ git add .                           # Stage all files
$ git status                          # Show status changed
$ git commit –m “First push”          # Commit the changes
```

continued …

Start Locally

7. In GitHub, create a new repository. Change the name, owner (as needed), add a description, and switch sharing permissions to Private.

   NOTE: Do not add a README, .gitignore, or license.

8. Designate the remote location, transfer protocol (SSH/HTTPS), and associate both with the "`origin`" repository alias.

   ```
   $  git remote add origin <SSH/HTTPS link>
   ```

9. Push the project file to the empty remote repository.

   ```
   $  git push -u origin main
   ```

10. Refresh the GitHub page to see the changes reflected.

Start Remotely

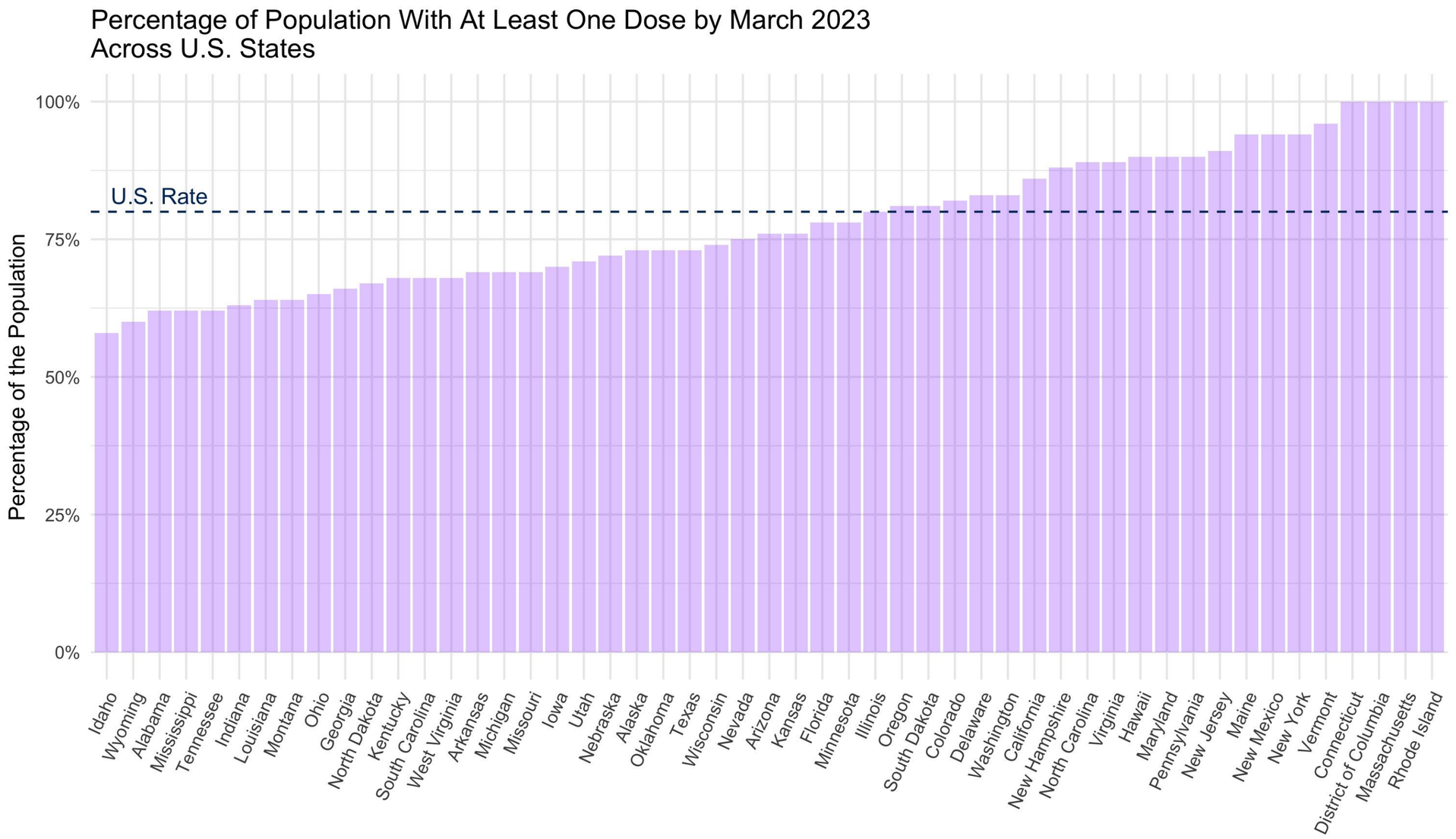
1. Open the JHU-CRC-Vaccinations GitHub remote repository URL.
2. Follow Method #1 or #2 to create a clean-break copy of the remote repository in your personal GitHub account.
3. Navigate to the file location you want to store the repository copy.

```
$ cd "/file location"        # Project location
```

4. Clone the repository that was copied into your personal GitHub.

```
$ git clone <SSH/HTTPS link>
```

continued …


Percentage of Population With At Least One Dose by March 2023
Across U.S. States
Percentage of the Population
100%
75%
50%
25%
0%
U.S. Rate
Idaho
Wyoming
Alabama
Mississippi
Tennessee
Indiana
Louisiana
Montana
Ohio
Georgia
North Dakota
Kentucky
South Carolina
West Virginia
Arkansas
Michigan
Missouri
Iowa
Utah
Nebraska
Alaska
Oklahoma
Texas
Wisconsin
Nevada
Arizona
Kansas
Florida
Minnesota
Illinois
Oregon
South Dakota
Colorado
Delaware
Washington
California
New Hampshire
North Carolina
Virginia
Hawaii
Maryland
Pennsylvania
New Jersey
Maine
New Mexico
New York
Vermont
Connecticut
District of Columbia
Massachusetts
Rhode Island

Start Remotely

5. Enter the cloned project file.

```
$ cd "repo name"                # Enter the project
```

6. Open the R project environment.

```
$ open JHU-CRC-Vaccinations.Rproj
```

7. Initialize the environment.

```
> renv::init()      # Initialize the project
> renv::restore()   # Download packages and their
                    # version saved in the lockfile.
```

continued ...

Start Remotely

8. Generate the vaccinations bar graph and save a JPEG to the project folder.
9. Stage and commit the changes.

```
$ git add .                          # Stage all files
$ git commit –m “New plot”           # Commit the changes
$ git push                           # Merge with remote
```

10. Refresh the GitHub page to see the changes reflected.

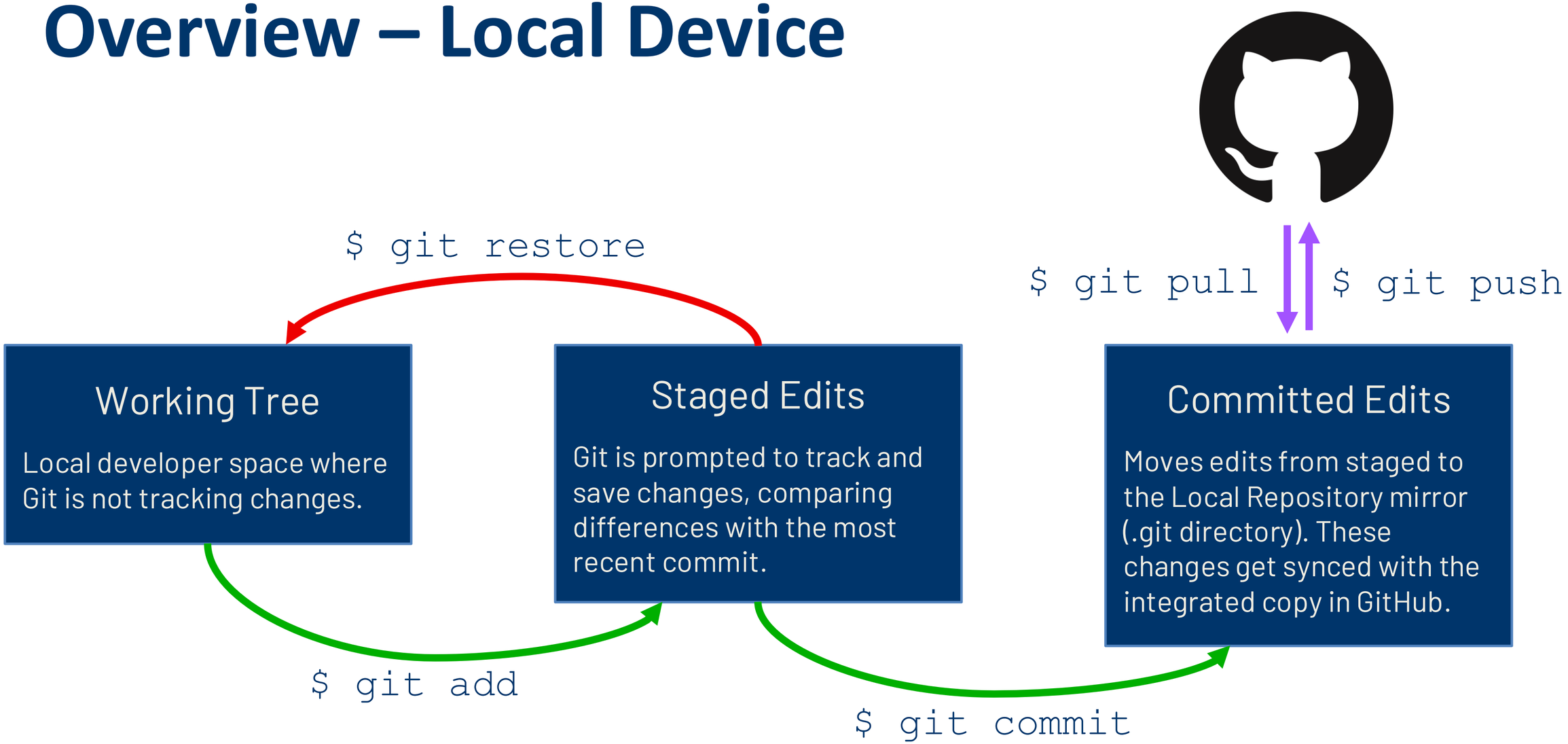
Discussion:

Try generating another plot and go through the same version control process.

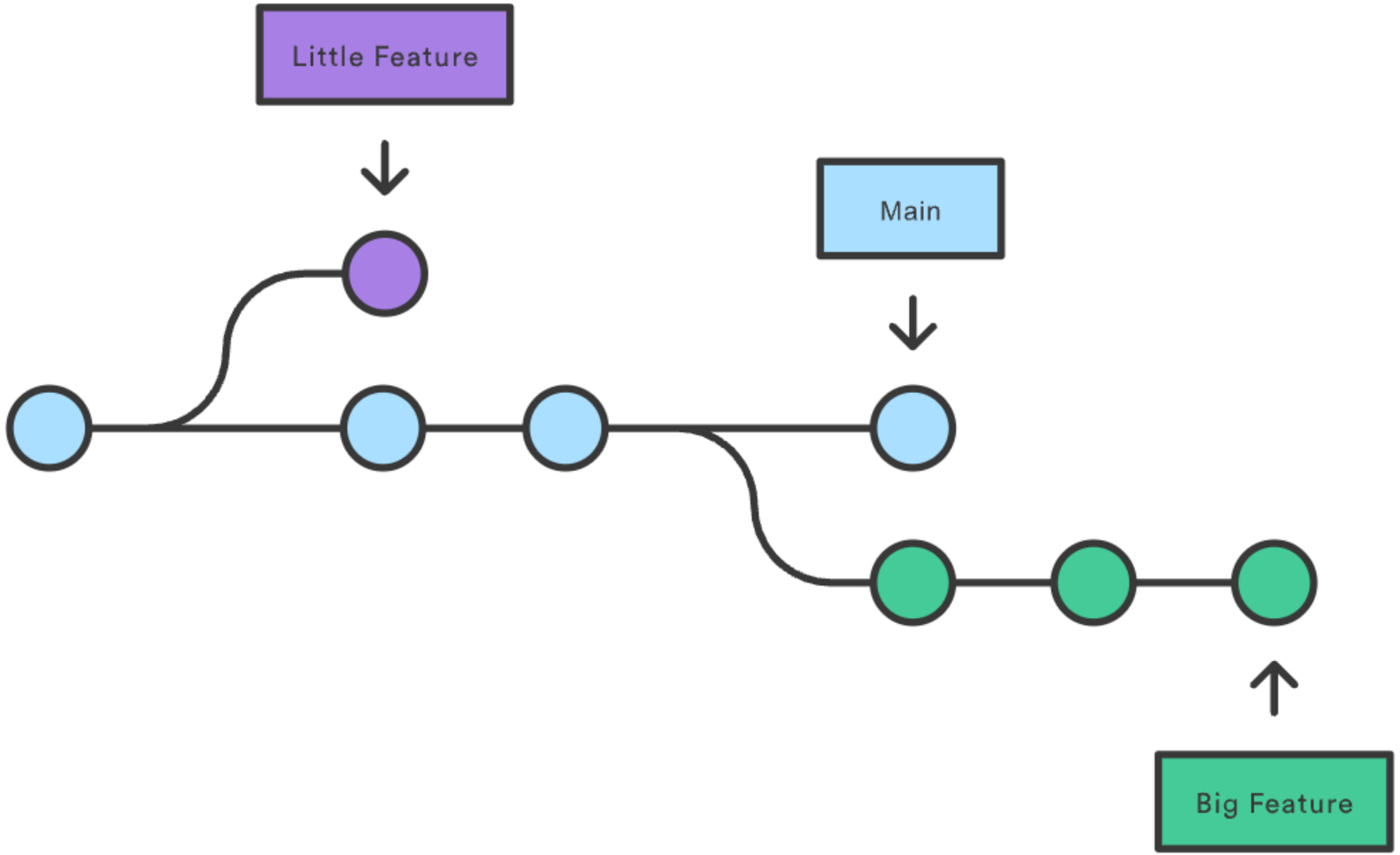
Discuss your work with your group members.

Leveraging Branches for Collaboration

Overview – Local Device



Consider the example…

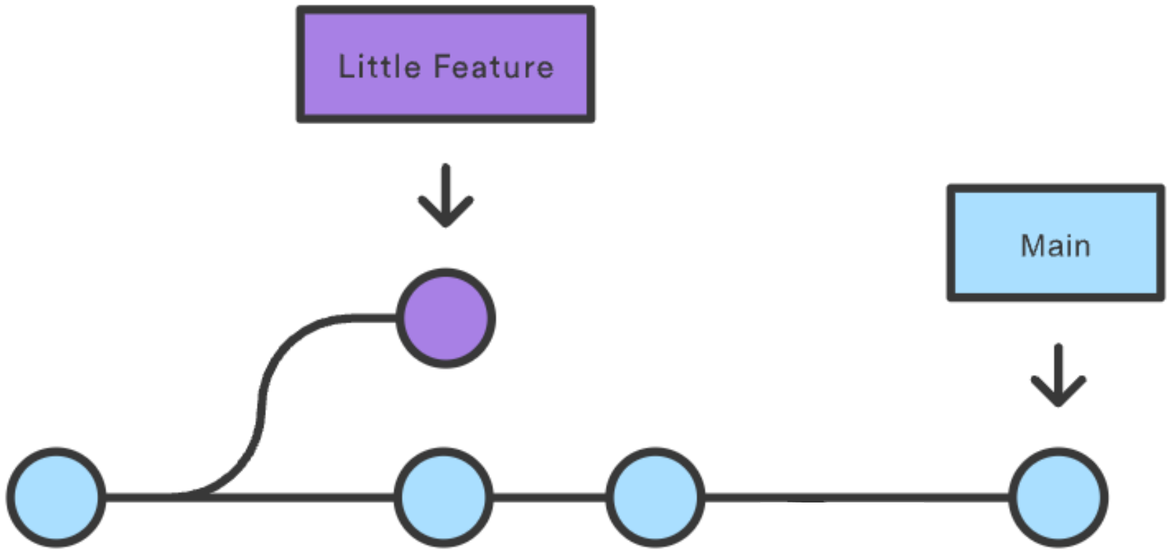


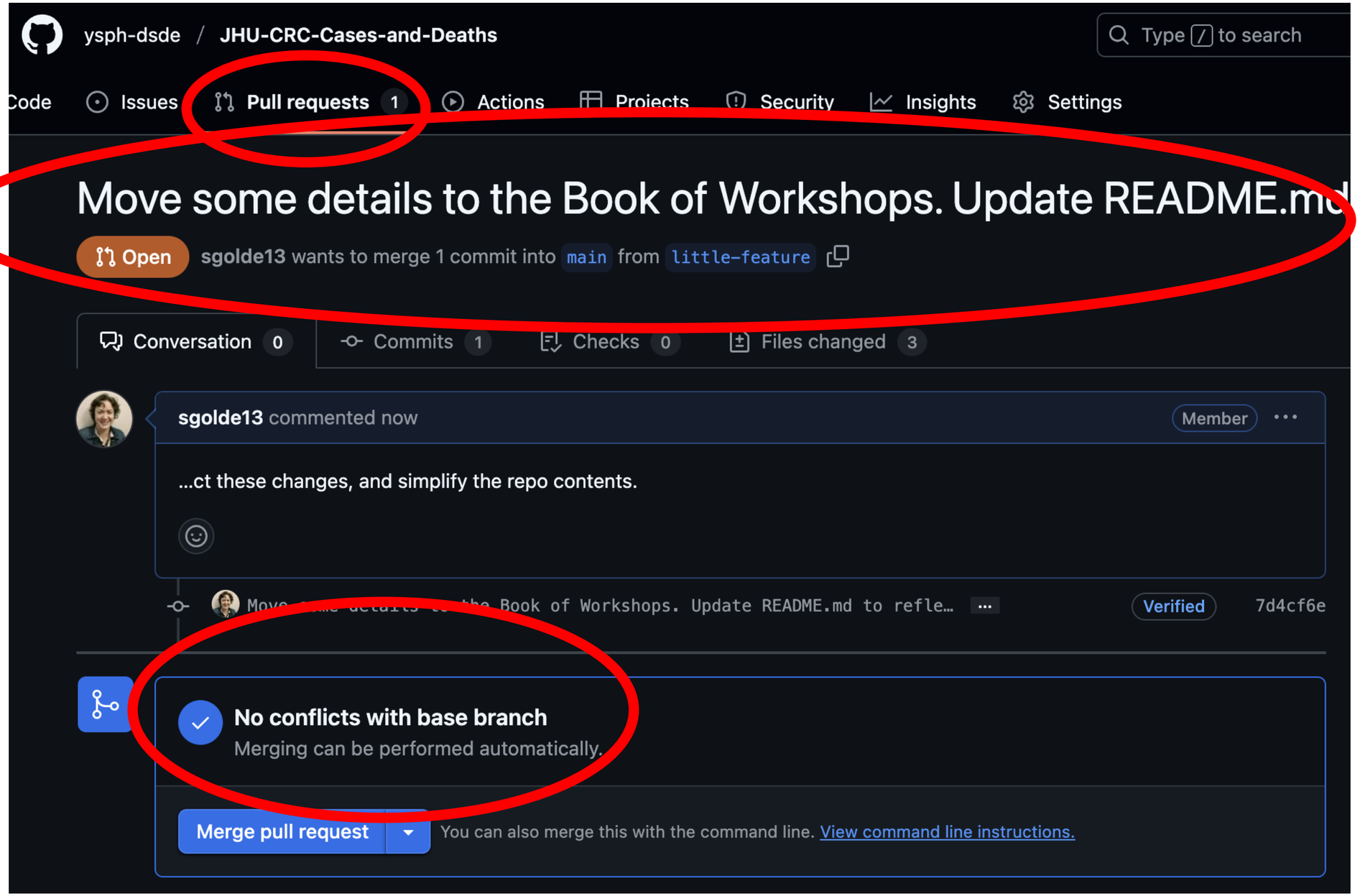
[git branch](#) from Atlassian's Git tutorials, Downloaded April 27th, 2025.

Command-Line Application

```
# Create a new branch called "little-feature".
git branch little-feature

# Move from the main branch into the new one.
git checkout little-feature
```




ysph-dsde / JHU-CRC-Cases-and-Deaths
Type / to search
Code
Issues
Pull requests 1
Actions
Projects
Security
Insights
Settings
Move some details to the Book of Workshops. Update README.md
Open
sgolde13 wants to merge 1 commit into main from little-feature
Conversation 0
Commits 1
Checks 0
Files changed 3
sgolde13 commented now
Member
...ct these changes, and simplify the repo contents.
Book of Workshops. Update README.md to refle…
Verified
7d4cf6e
No conflicts with base branch
Merging can be performed automatically.
Merge pull request
You can also merge this with the command line. View command line instructions.

Collaborating with the Team

Pull Request

1. In the ysph-dsde/JHU-CRC-Cases-and-Deaths repository, navigate to its "Settings" → "Collaborators and teams" page.
2. Under the "Manage Access" section, click "Add people" and enter in the GitHub username of your teammates.
3. Your team should receive an email about the invitation to collaborate.
4. Navigate to the file location you want to store the repository copy.

```
$ cd "/file location"       # Project location
```

5. Clone the repository that was copied into your personal GitHub.

```
$ git clone <SSH/HTTPS link>
```

continued ...

Pull Request

6. Enter the cloned project file.

```
$ cd "repo name"                    # Enter the project
```

7. Check that we are on branch "main" and list other available branches.

```
$ git branch
```

8. Have each teammate create a new branch with their name.

```
$ git checkout -b Shelby
```

9. Confirm you are on the new branch.

```
$ git branch
```

continued ...

Pull Request

10. Open the R project environment.

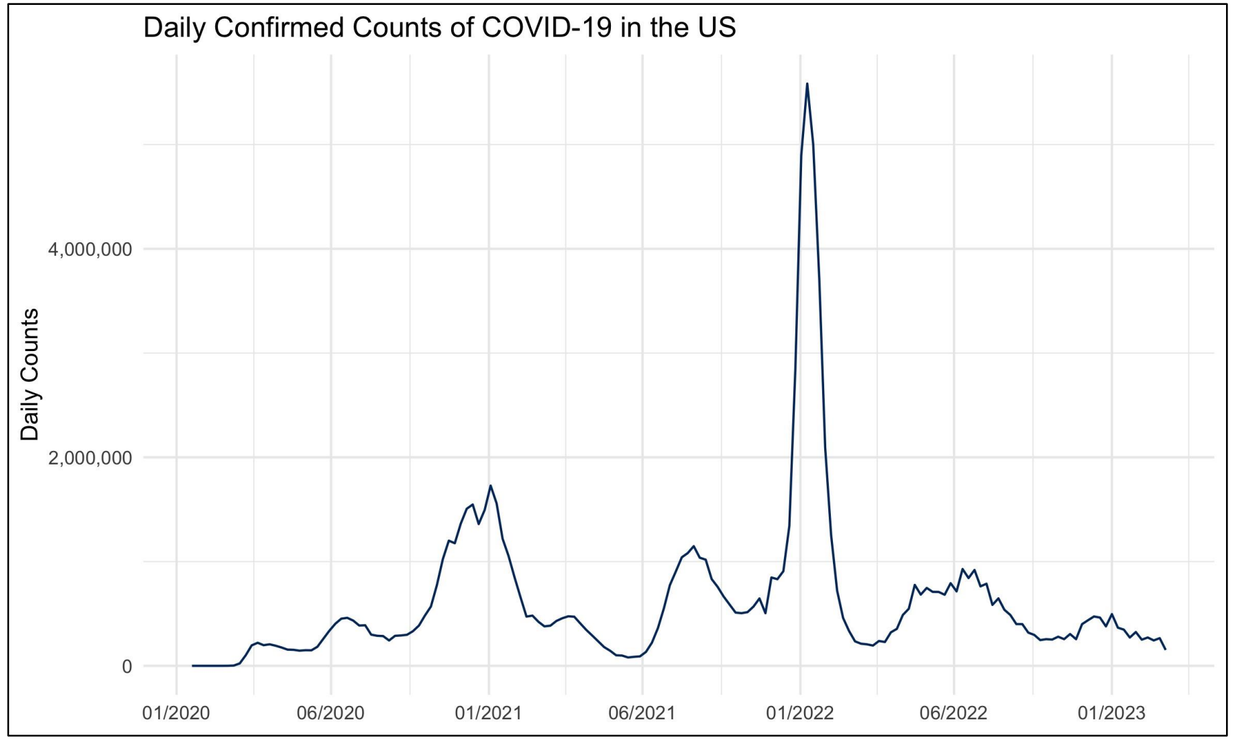
```
$ open JHU-CRC-Cases-and-Deaths.Rproj
```

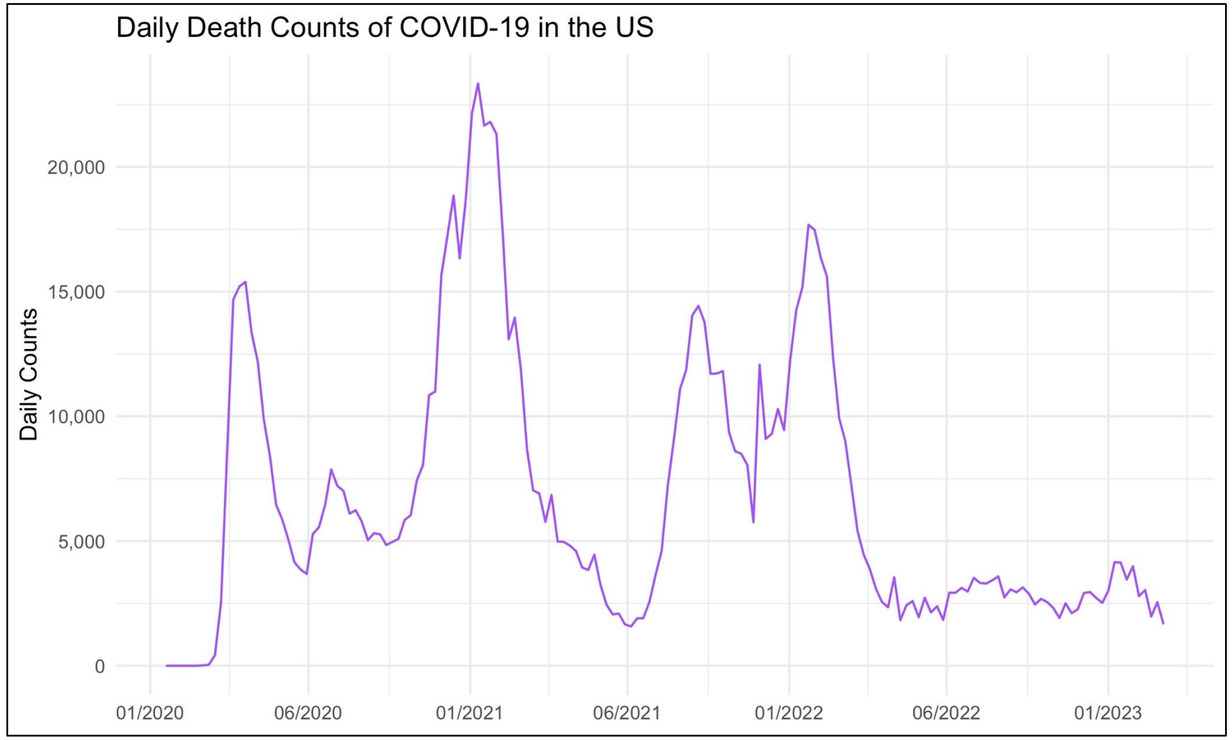
11. Initialize the environment.

```
> renv::init()      # Initialize the project
> renv::restore()   # Download packages and their
                    # version saved in the lockfile.
```

12. Have each person choose a different state or region to visualize.
13. Generate the cases and deaths line graph and save a JPEG to the project folder.

continued ...


Daily Confirmed Counts of COVID-19 in the US
Daily Counts
4,000,000
2,000,000
0
01/2020
06/2020
01/2021
06/2021
01/2022
06/2022
01/2023


Daily Death Counts of COVID-19 in the US
Daily Counts
20,000
15,000
10,000
5,000
0
01/2020
06/2020
01/2021
06/2021
01/2022
06/2022
01/2023

Pull Request

14. Stage and commit the changes.

```
$ git add .                          # Stage all files
$ git commit –m “New plot”           # Commit the changes
```

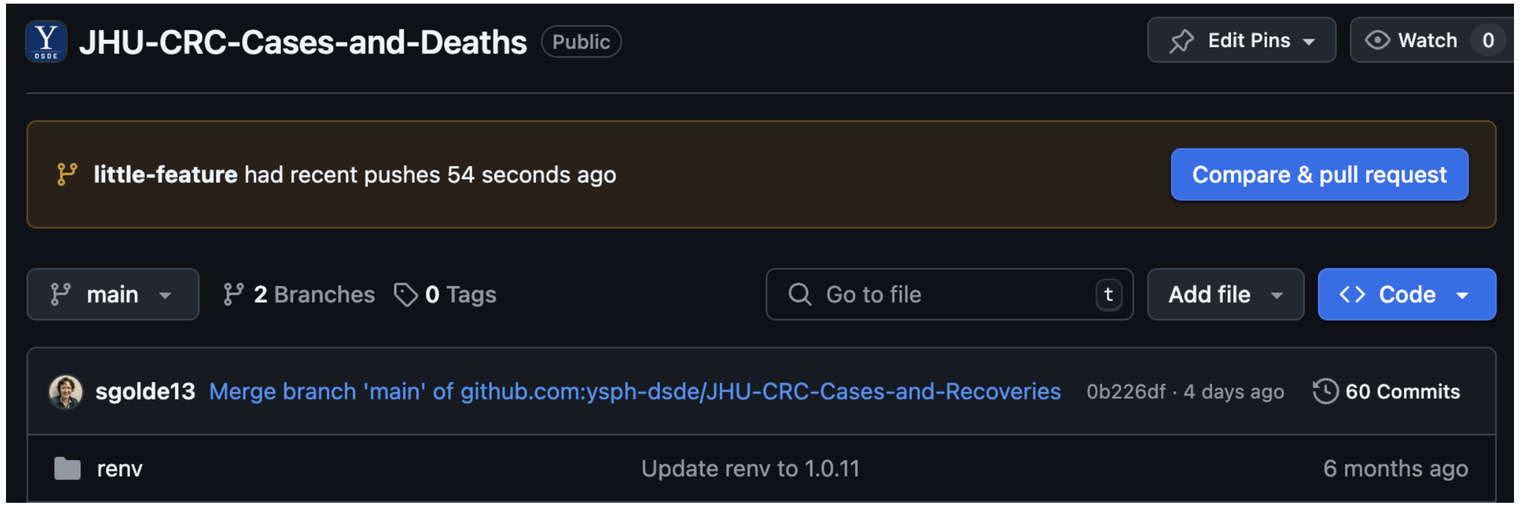
15. Push the changes (set remote as upstream).

```
$ git push –u origin Shelby
```

continued …

Pull Request

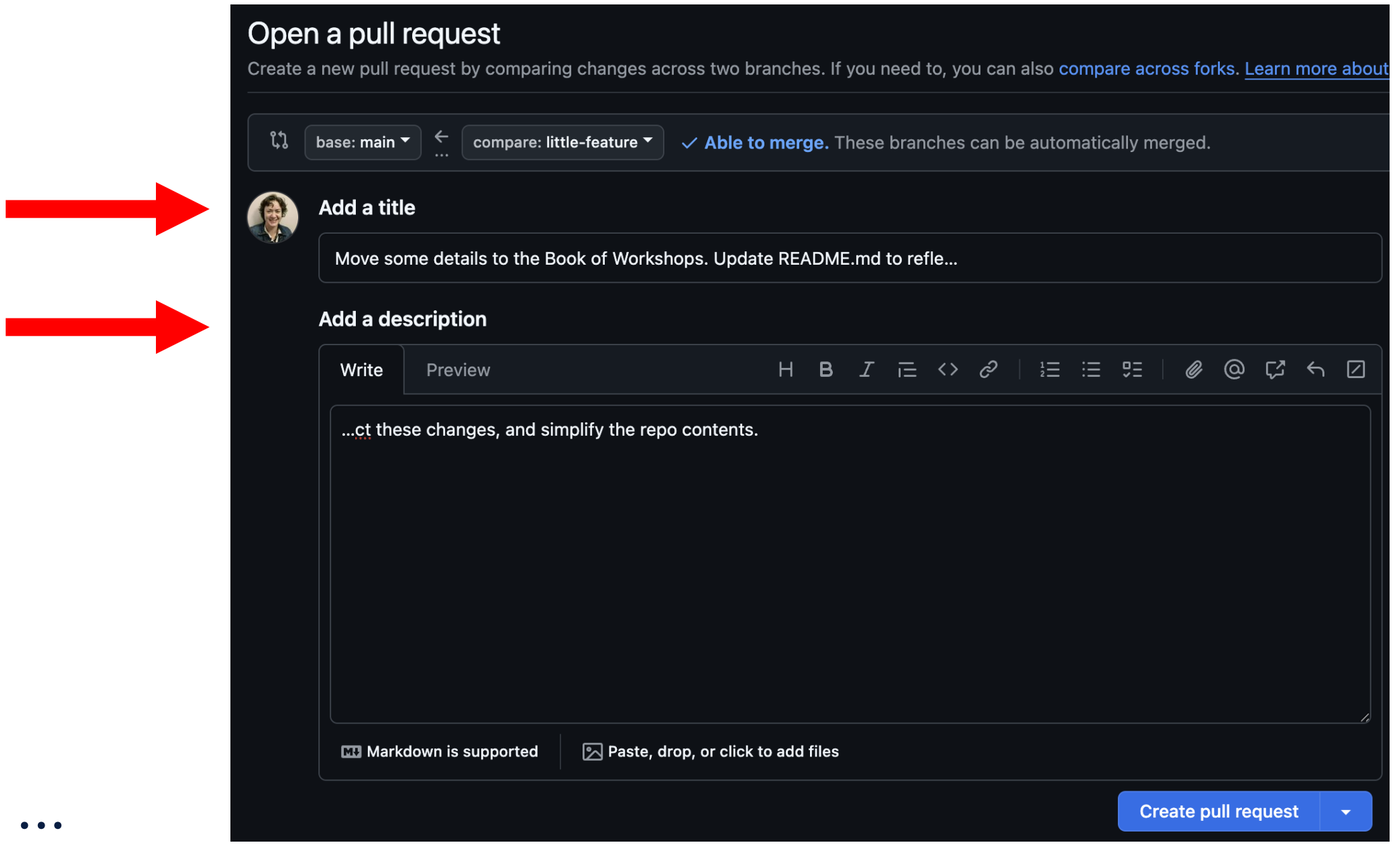
16. Refresh the GitHub page to see the changes reflected. In the top of the GitHub page for the repository, a yellow banner should appear.
    - If there is no banner, navigate to the “Pull Request” tab and submit a request manually.
    - If one does appear, then click “Compare & pull request”.



continued ...

Pull Request

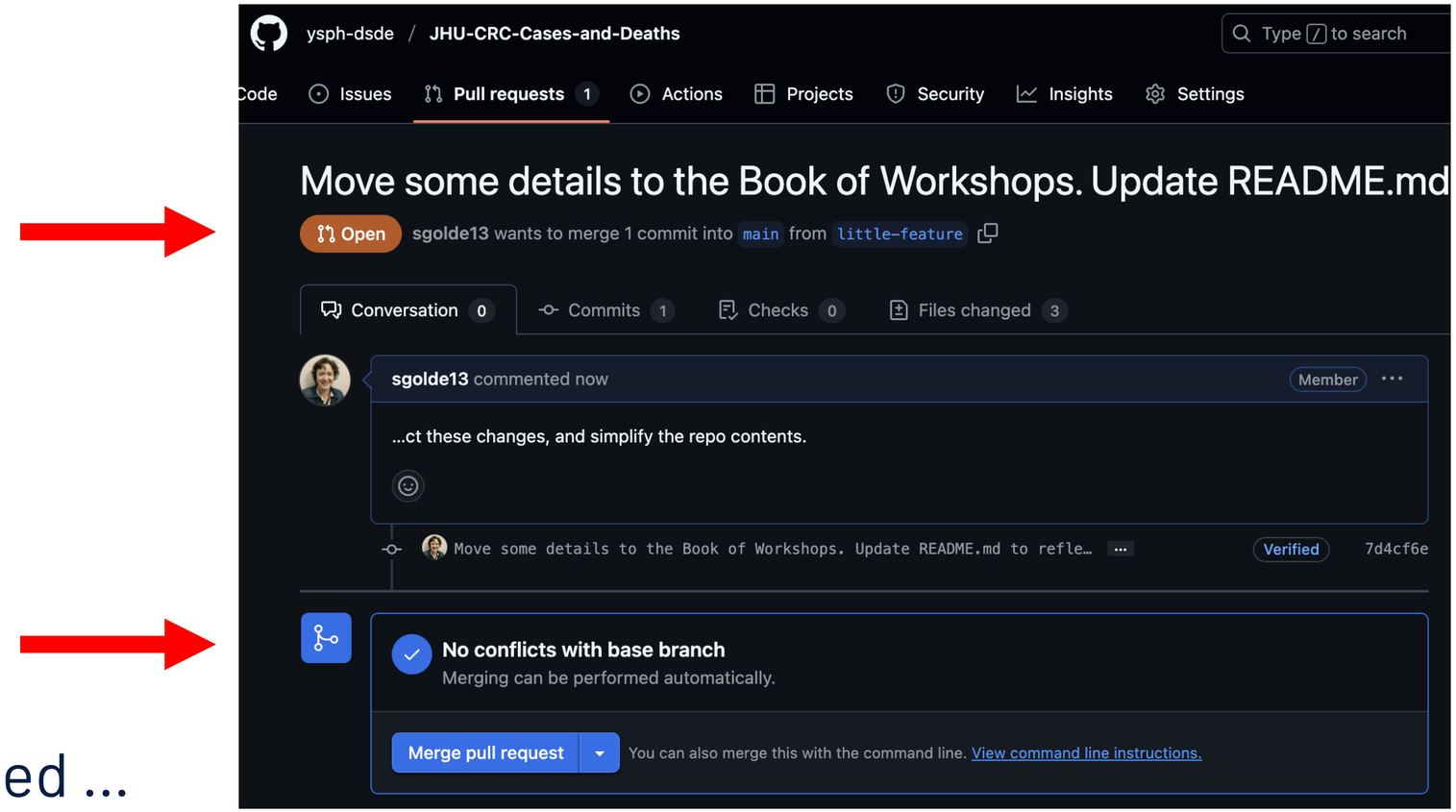
17. OPTIONAL: In the “Pull Request” page for the open request, change the title or add a description for the request.



continued ...

Pull Request

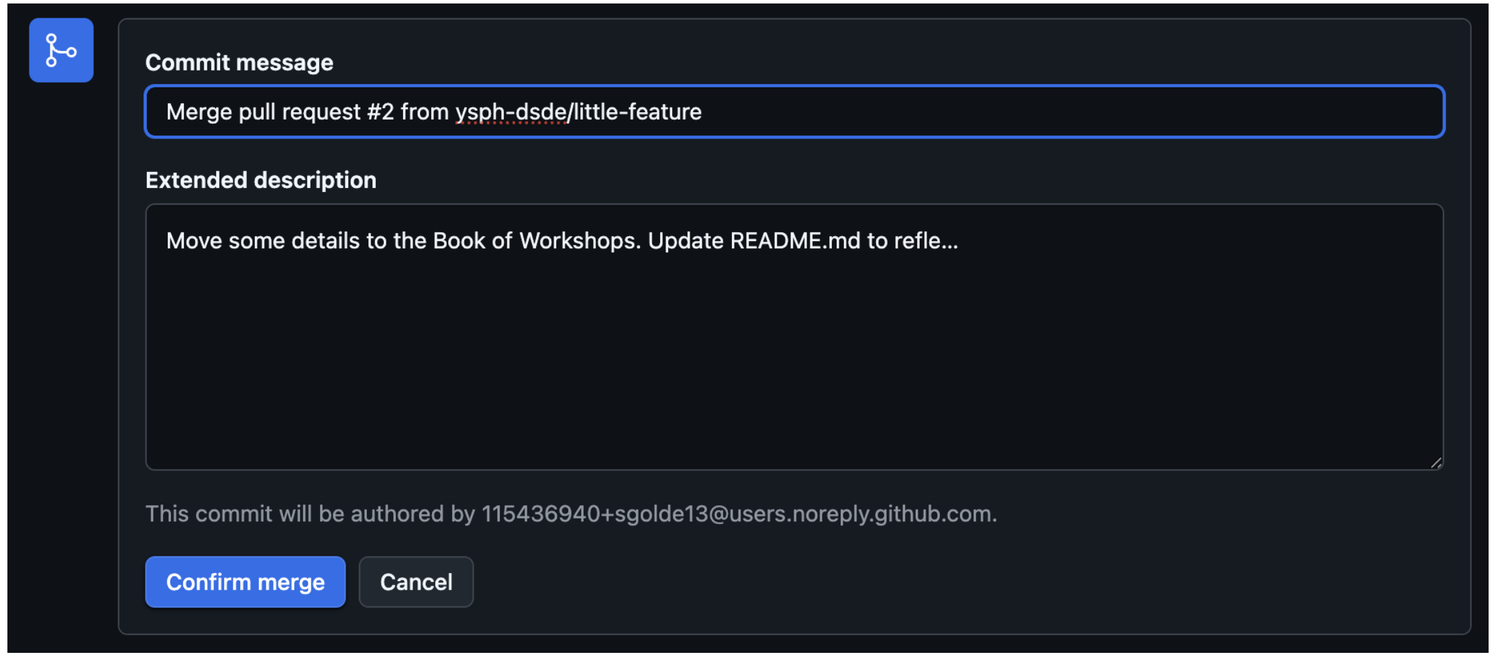
18. Review the conflicts detected. Discuss with your team how to proceed with merging the branches to main.



continued ...

Pull Request

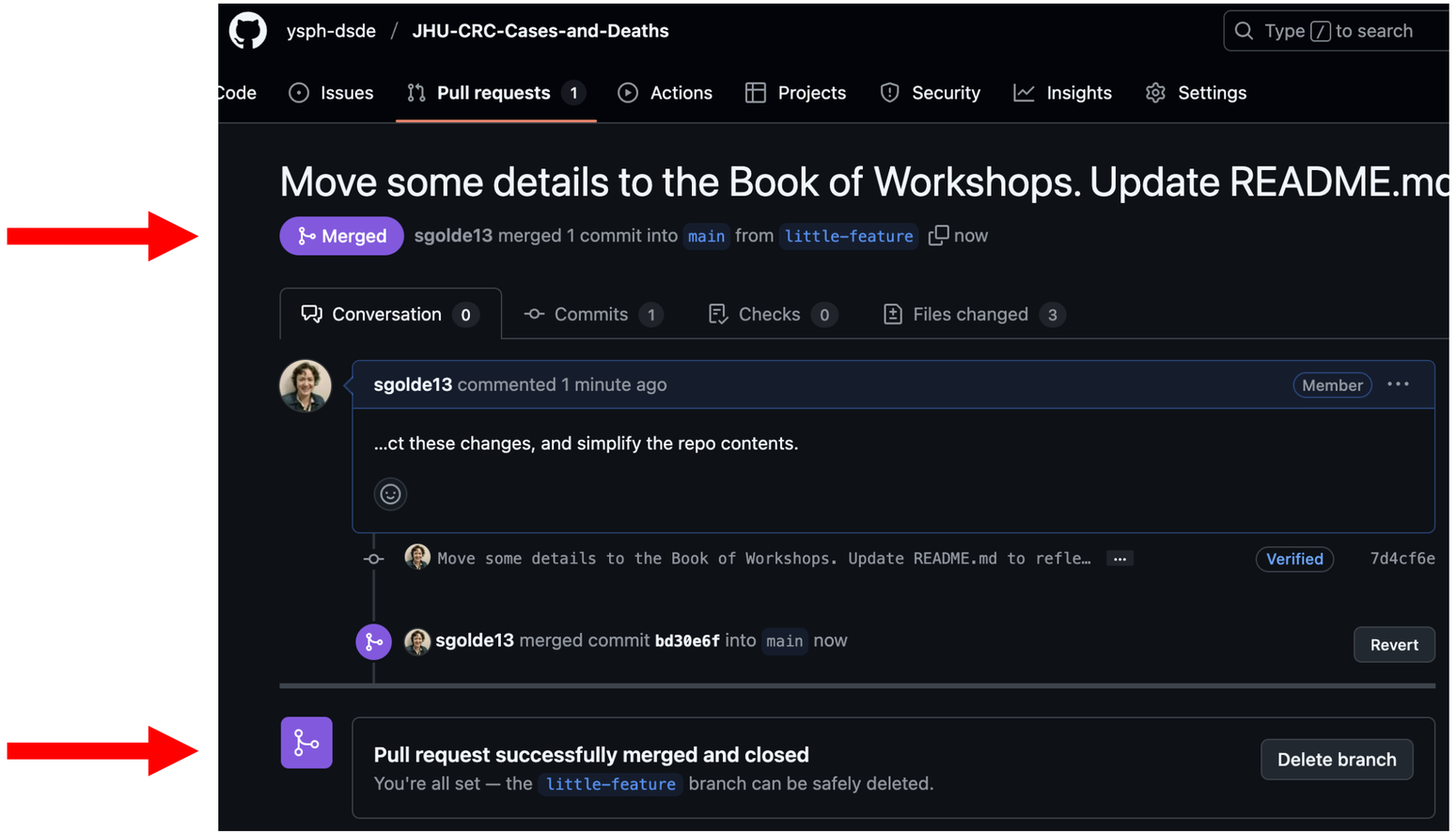
19. After reconciling your conflicts, confirm the branch merge to main.



continued ...

Pull Request

20. A successful merge should give the following window.



Discussion:

Did you generate any conflicts? Discuss why they came up, or why they did not.

How did you reconcile the conflicts?

Feedback Form for Workshop: Getting Started with Git & GitHub Part 2

Thank you!

ysph.yale.edu
sph.yale.edu/dsde

@YaleSPH

Public Health Data Science and Data Equity
Yale School of Public Health
60 College Street, New Haven, CT 06510

Yale SCHOOL OF PUBLIC HEALTH

Appendix

Glossary

Version Control	Manage, organize, and track different versions of files. These systems identify differences between versions and allows reverting to older versions. Example: Google docs.
Distributive Version Control System	The project codebase is copied as a mirror to each contributor's local computer. Local changes get synched via patches sent peer-to-peer through the server.
Command-Line Interface (CLI)	A texted-base application that directly interacts with the computer’s operating system, manages files, and can run programs. It typically lacks a graphical user interface (GUI).

Glossary

Shell A program used by the CLI to mediate communication between the user and computer by interpreting commands and outputs. Examples: Bash, Power Shell, etc.

Mirror An exact copy of a project from a server, including a full-change history.

Server Computer or system that provides resources (i.e. data or programs) to other computers, known as clients, over a network.

Patch Snippets of code or data used to update existing software.

Glossary

Peer-to-peer Participants in a network act as both client and server by trading resources and services with one another.

`git add` Prompt git to track changes that have been made to specific files and compare those differences to previously saved version in the .git directory.

`git commit` Save your changes as a snapshot in your project's history, present in the .git directory.

`git push` Upload your committed work to a shared online location, so others can see it.

Glossary

`git merge`	One method to reconcile different committing histories in divergent branches. Creates a new version integrating the head of the two branches in a three-way commit.
`git rebase`	An alternative to merge. The branch commit histories are realigned so that the leading one defines the commit parent history of the following branch, thus rebasing its commits.
`git clone`	Make a copy of a project from the internet to your own computer.
`git branch`	Create or list different “versions” or “paths” of a project that you can work on separately.

Glossary

`git checkout` Command to switch between different the branches.

Helpful Cheat Sheets:

1. Git Cheat Sheet by Atlassian
2. Vim Cheat Sheet by Richard Torruellas
3. Bash Shortcuts by Mohan Balasundaram
4. Command Line Cheat Sheet by Tobias Günther

References

Slide 1

1. xkcd, “Git.” Accessed: Apr. 27, 2025. [Online]. Available: https://xkcd.com/1597/

Slide 4

1. K. Nelson Ph.D., “Version Control with Git,” Yale Center for Research Computing (YCRC). Accessed: Oct. 09, 2024. [Online]. Available: https://research.computing.yale.edu/training/version-control-git
2. J. DeMayo, “gitdemo,” Harvey Cushing/John Hay Whitney Medical Library. Accessed: Nov. 01, 2024. [Online]. Available: https://github.com/CWML/gitdemo/tree/main
3. Atlassian, “Learn Git,” Atlassian Tutorials. Accessed: Oct. 09, 2024. [Online]. Available: https://www.atlassian.com/git
4. W3Schools, “Git Tutorial,” W3Schools. Accessed: Oct. 09, 2024. [Online]. Available: https://www.w3schools.com/git/default.asp?remote=github
5. GitHub, “introduction-to-github,” GitHub Skills. Accessed: Nov. 01, 2024. [Online]. Available: https://github.com/skills/introduction-to-github

References

Slide 4 continued

6. J. Bryan and J. Hester, “Let’s Git started,” Happy Git and GitHub for the useR. Accessed: Oct. 09, 2024. [Online]. Available: https://happygitwithr.com/

Slide 5

1. cbeams, “How to Write a Git Commit Message,” cbeams. Accessed: Oct. 28, 2024. [Online]. Available: https://cbea.ms/git-commit/
2. Git Developers, “Git - Reference,” Git. Accessed: Apr. 24, 2025. [Online]. Available: https://git-scm.com/docs
3. Git-SCM, “Git - GUI Clients,” Git-SCM. Accessed: Oct. 28, 2024. [Online]. Available: https://git-scm.com/downloads/guis
4. GitHub Developers, “GitHub Docs,” GitHub. Accessed: Apr. 24, 2025. [Online]. Available: https://docs.github.com/en

References

Slide 5 continued

5. A. Prajapati, “What is git commit, push, pull, log, aliases, fetch, config & clone,” Medium. Accessed: Oct. 28, 2024. [Online]. Available: https://medium.com/mindorks/what-is-git-commit-push-pull-log-aliases-fetch-config-clone-56bc52a3601c

6. Various contributors, “Git guides,” Graphite. Accessed: Oct. 28, 2024. [Online]. Available: https://graphite.dev/guides/topic/git

Slide 7

1. Atlassian, “What is version control,” Atlassian Tutorials. Accessed: Oct. 14, 2024. [Online]. Available: https://www.atlassian.com/git/tutorials/what-is-version-control

2. “Distributed version control,” Wikipedia. Accessed: Oct. 14, 2024. [Online]. Available: https://en.wikipedia.org/wiki/Distributed_version_control

3. “GitHub,” Wikipedia. Accessed: Oct. 14, 2024. [Online]. Available: https://en.wikipedia.org/wiki/GitHub

References

Slide 7 continued

4. “Git,” Wikipedia. Accessed: Oct. 14, 2024. [Online]. Available: https://en.wikipedia.org/wiki/Git
5. K. Finley, “The Problem With Putting All the World’s Code in GitHub,” WIRED Magazine. Accessed: Oct. 14, 2024. [Online]. Available: https://www.wired.com/2015/06/problem-putting-worlds-code-github/
6. S. Chaudhary, “How GitHub became the largest source code host?,” Medium. Accessed: Oct. 14, 2024. [Online]. Available: https://srishtichy.medium.com/how-github-became-the-largest-source-code-host-a1c37ea5e5f

Slide 8

1. StackOverflow, “Version control systems results,” StackOverflow Developer Survey. Accessed: Oct. 14, 2024. [Online]. Available: https://survey.stackoverflow.co/2022/#version-control-version-control-system-prof

References

Slide 9

1. GitHub, “Global distribution of developers,” GitHub Octoverse. Accessed: Oct. 14, 2024. [Online]. Available: https://octoverse.github.com/2022/global-tech-talent

Slide 12

1. D. Answered by Epp, “In Git, what is the difference between origin/master vs origin master?,” StackOverflow. Accessed: Oct. 28, 2024. [Online]. Available: https://stackoverflow.com/questions/18137175/in-git-what-is-the-difference-between-origin-master-vs-origin-master
2. Atlassian, “git pull,” Atlassian Tutorial. Accessed: Oct. 28, 2024. [Online]. Available: https://www.atlassian.com/git/tutorials/syncing/git-pull
3. G. Foster, “Git HEAD,” Graphite. Accessed: Oct. 28, 2024. [Online]. Available: https://graphite.dev/guides/git-head

References

Slide 12 continued

4. A. Prajapati, “What is git commit, push, pull, log, aliases, fetch, config & clone,” Medium. Accessed: Oct. 28, 2024. [Online]. Available: https://medium.com/mindorks/what-is-git-commit-push-pull-log-aliases-fetch-config-clone-56bc52a3601c

Slide 13

1. Atlasian, “git clone,” Atlassian Tutorial. Accessed: Oct. 28, 2024. [Online]. Available: https://www.atlassian.com/git/tutorials/setting-up-a-repository/git-clone
2. Atlassian, “git fetch,” Atlassian Tutorial. Accessed: Oct. 28, 2024. [Online]. Available: https://www.atlassian.com/git/tutorials/syncing/git-fetch
3. Atlassian, “git merge,” Atlassian Tutorial. Accessed: Oct. 28, 2024. [Online]. Available: https://www.atlassian.com/git/tutorials/using-branches/git-merge

References

Slide 14-15

1. Atlassian, “git rebase,” Atlassian Tutorial. Accessed: Apr. 26, 2025. [Online]. Available: https://www.atlassian.com/git/tutorials/rewriting-history/git-rebase
2. Atlassian Tutorial, “git fetch.” Accessed: Apr. 26, 2025. [Online]. Available: https://www.atlassian.com/git/tutorials/syncing/git-fetch
3. Atlassian, “git merge,” Atlassian Tutorial. Accessed: Apr. 26, 2025. [Online]. Available: https://www.atlassian.com/git/tutorials/using-branches/git-merge

Slide 17

1. xkcd, “Painbow color scale.” Accessed: Apr. 27, 2025. [Online]. Available: https://flowingdata.com/2021/11/08/painbow-color-scale/

References

Slide 18-24

1. Answered by Go Dan, “File Permissions - How to read the mode field of git-ls-tree’s output,” StackExchange. Accessed: Apr. 24, 2025. [Online]. Available: https://stackoverflow.com/questions/737673/how-to-read-the-mode-field-of-git-ls-trees-output/8347325#8347325
2. K. Hess, “An introduction to the vi editor.” Accessed: Apr. 22, 2025. [Online]. Available: https://www.redhat.com/en/blog/introduction-vi-editor

Slide 27-29

1. GitHub, “Adding locally hosted code to GitHub,” GitHub Docs. Accessed: Oct. 28, 2024. [Online]. Available: https://docs.github.com/en/migrations/importing-source-code/using-the-command-line-to-import-source-code/adding-locally-hosted-code-to-github

References

Slides 37-39

1. Atlassian, “Creating and merging branches in Git,” Git Guides. Accessed: Oct. 28, 2024. [Online]. Available: https://youtu.be/S2TUommS300?si=nntkOXHjPL5SnGMx
2. Atlassian, “git branch,” Atlassian Tutorial. Accessed: Apr. 26, 2025. [Online]. Available: https://www.atlassian.com/git/tutorials/using-branches
3. GitHub, “About branches,” GitHub Docs. Accessed: Oct. 28, 2024. [Online]. Available: https://docs.github.com/en/pull-requests/collaborating-with-pull-requests/proposing-changes-to-your-work-with-pull-requests/about-branches
4. GitHub, “About Pull Requests,” GitHub Docs. Accessed: Oct. 28, 2024. [Online]. Available: https://docs.github.com/en/pull-requests/collaborating-with-pull-requests/proposing-changes-to-your-work-with-pull-requests/about-pull-requests

References

Glossary

1. anukruti16, “What is P2P (Peer-to-Peer Process)?,” GeeksforGeeks. Accessed: Nov. 02, 2024. [Online]. Available: https://www.geeksforgeeks.org/what-is-p2p-peer-to-peer-process/
2. Atlassian, “Git cheat sheet,” Atlassian Tutorial. Accessed: Nov. 02, 2024. [Online]. Available: https://www.atlassian.com/git/tutorials/atlassian-git-cheatsheet
3. M. Balasundaram, “Bash keyboard shortcuts,” GitHub Gist. Accessed: Nov. 02, 2024. [Online]. Available: https://gist.github.com/tuxfight3r/60051ac67c5f0445efee
4. G. Foster, “Git add, commit, and push,” Graphite. Accessed: Nov. 02, 2024. [Online]. Available: https://graphite.dev/guides/git-add-commit-push
5. GeeksforGeeks Contributor, “Difference between Terminal, Console, Shell, and Command Line.” Accessed: Apr. 25, 2025. [Online]. Available: https://www.geeksforgeeks.org/difference-between-terminal-console-shell-and-command-line/

References

Glossary continued

6. T. Gunther, “Command Line Cheat Sheet | Tower Blog.” Accessed: Apr. 25, 2025. [Online]. Available: https://www.git-tower.com/blog/command-line-cheat-sheet

7. Paessler GmbH, “Server - Definition and details,” IT Explained. Accessed: Nov. 02, 2024. [Online]. Available: https://www.paessler.com/it-explained/server

8. R. Torruellas, “Vim Cheat Sheet,” Vim. Accessed: Nov. 02, 2024. [Online]. Available: https://vim.rtorr.com/