



Introduction to Version Control with git and github

Bryan Scott

LSSTC Data Science Fellowship Program Pre-Orientation

August 31, 2023



Some git Resources

Pro Git book available at git-scm.com/docs

and

Beginning Git and GitHub: A Comprehensive Guide to Version Control, Project Management, and Teamwork for the New Developer by Mariot Tsitoara

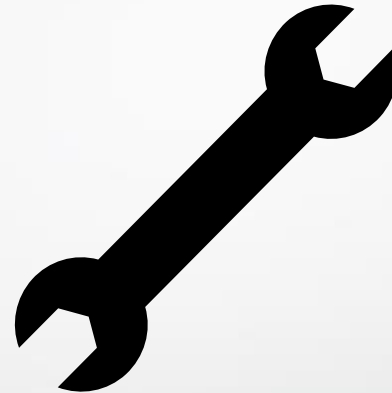
What makes learning git difficult?



What makes learning git difficult?



Let's be real – development workflows...



Brilliant idea!

Prototyping

Iterative Fixes

Oh no!

Problems in typical development workflows...

- 🔗 MG_IM_models_revised_2.py
- 🔗 MG_IM_models_revised_3.py
- 🔗 MG_IM_models_revised_4.py
- 🔗 MG_IM_models_revised_5.py
- 🔗 MG_IM_models_revised_6.py
- 🔗 MG_IM_models_revised_7_test.py
- 🔗 MG_IM_models_revised_7.py
- 🔗 MG IM models revised.py

Paper_draft_final.tex

Paper_draft_final_revised.tex

Paper_draft_final_revised_final.tex

Paper_draft_final_revised_final_submission.tex

Paper_draft_final_revised_final_submission_with_comments.tex

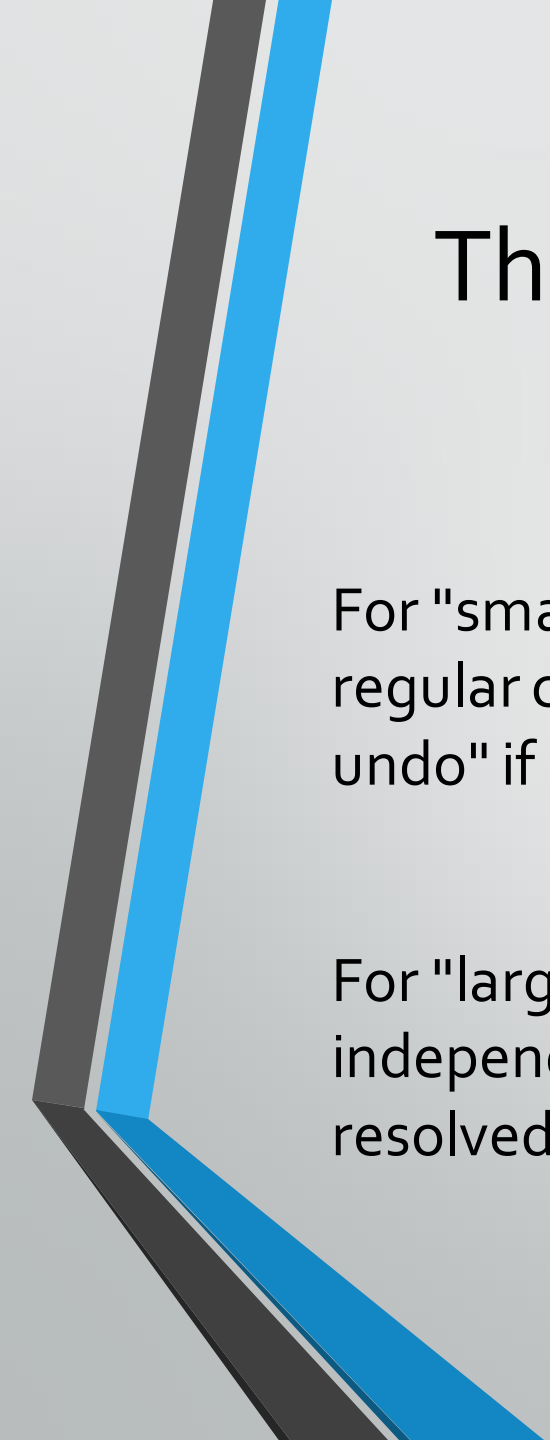
Paper_draft_final_revised_final_submission_with_comments_and_responses_draft.tex
(and so on....)

What is version control?

Version control is a *system* for creating a *reproducible* record of changes to a *project*.

The key idea is that there is only **one** project. *No matter* how big or complicated that project is.

Git is a *distributed* version control system – there is no authoritative master repository and each copy is as valid as any other.



The usefulness of software repositories for different kinds of developers...

For "small projects", tracking changes with git (or another VCS) and making regular commits makes your work more reproducible and gives you a "global undo" if you introduce a catastrophic error into your code.

For "large projects", a distributed VCS can allow multiple developers to work independently and synchronously. Changes are merged later (and conflicts resolved appropriately).



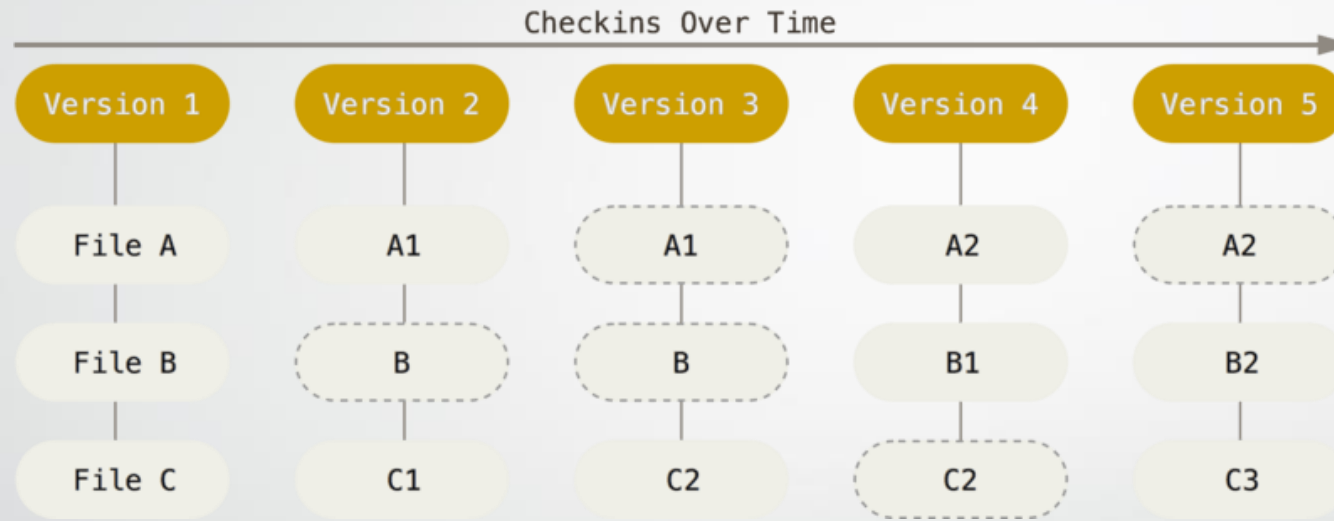
Introduction to git

Distributed Version Control

The Distributed Version Control Model (Git)

- A software repository is a place for storing software and metadata about it.
- In a distributed version control system, each developer has a copy of the repository – while it may be convenient to choose a "central" copy as a way of aiding collaboration – there is no authoritative version of the repository.
- Git is an example of a distributed version control system.

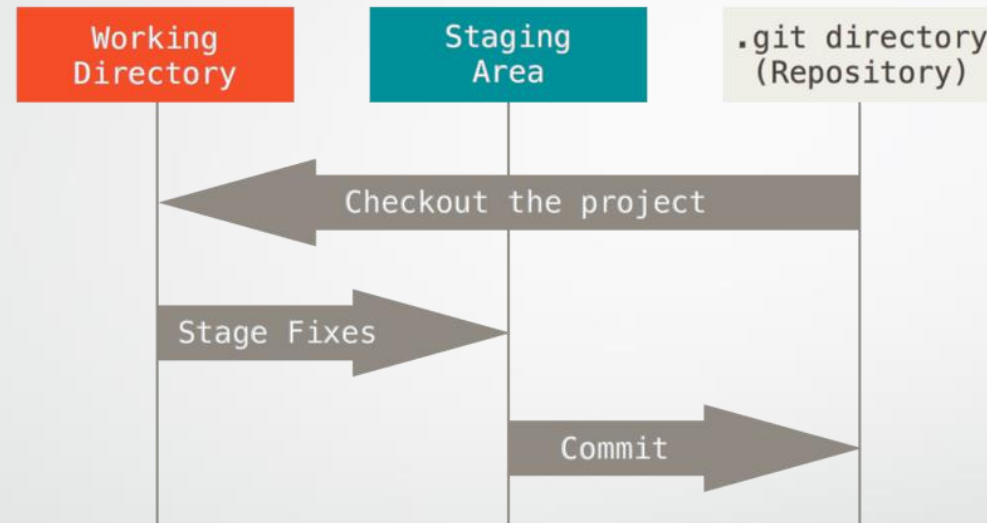
How does Git work?



Git stores snapshots of the project over time. If a new file is added or an old file changed, the new version will be stored in the next snapshot.

But if a file is unchanged, git simply links to the version stored in the previous snapshot.

Git States

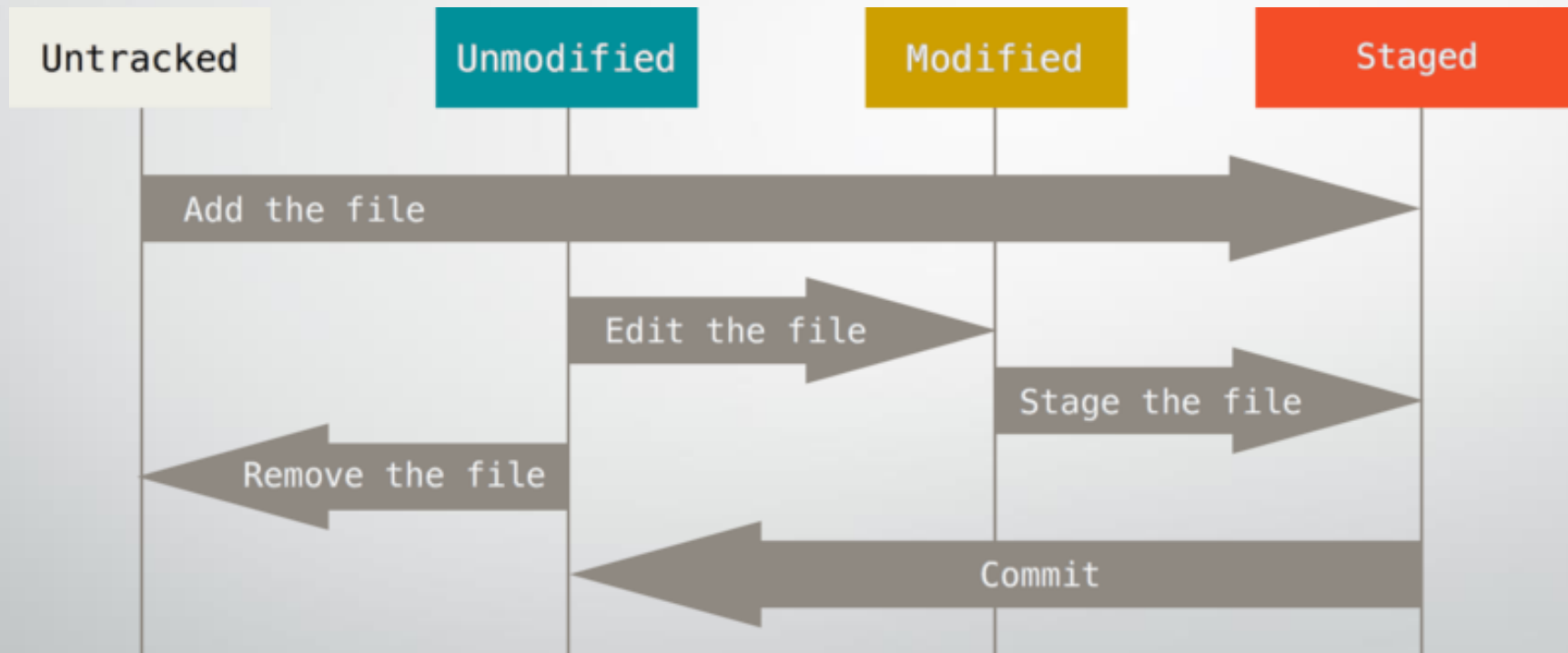


Files can be in one of three states. *Modified, staged, and committed.* Files move between the three stages as you work. After editing a file it is *modified*. You then *stage* the file which tells git to include it in the next snapshot. You then *commit* the changes (save a snapshot of the project).

Git Usage – Changes to the Repository

- Files in your project can be in one of two states – *tracked* or *untracked*
- Git tracks all files that were in the last snapshot (*commit*) or that have been created and staged with the *git add* command.
- To add changed files to the repository, you need to stage them with *add* and then snapshot with *commit*

Git File Cycle





Introduction to Github

+ branches, merges, and more

Git as a distributed version control system

- Remember, git is a distributed version control system. Up until now, we have considered only a local copy of the software repository on our machine.
- But there's nothing special about our copy – we can have n copies of our project that are shared across many developers. We'll need to figure out how to manage n copies (not easy!), but the power of being able to share and collaborate is obvious!
- It would help if we had a convenient place to store a copy that we'll all work from – this is where **github** or **gitlab** come in.


Create a remote repository on github

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Owner *


Repository name *

 bscot ▾


/

Great repository names are short and memorable. Need inspiration? How about [musical-doodle?](#)

Description (optional)

☒  **Public**

Anyone on the internet can see this repository. You choose who can commit.

☐  **Private**

You choose who can see and commit to this repository.

Initialize this repository with:
Skip this step if you're importing an existing repository.

☐ **Add a README file**
This is where you can write a long description for your project. [Learn more.](#)

Add .gitignore
Choose which files not to track from a list of templates. [Learn more.](#)

.gitignore template: None ▾

Choose a license
A license tells others what they can and can't do with your code. [Learn more.](#)

License: None ▾

(i) You are creating a public repository in your personal account.

Two ways to create a Repository

- We can create a local repository by navigating to a directory and typing

```
git init
```

- Or we can clone a repository from remote by typing

```
git clone [link to remote]
```

Linking the local and remote repositories

- Once we have a remote repository, we can link the local to it with `git remote add [name] [link]`. By convention, origin or upstream are used as the name of the remote.

```
Git remote add origin [link to your fork of the DSFP repo]
```

- You will sometimes see this command – this sends your code to the remote repository

```
git push origin main
```

- Where origin is the [remote name] and [main] is the branch we want to push.

Caution: the command `git push origin master` was the default until recently. Main has replaced it as the preferred/default name for your production branch.

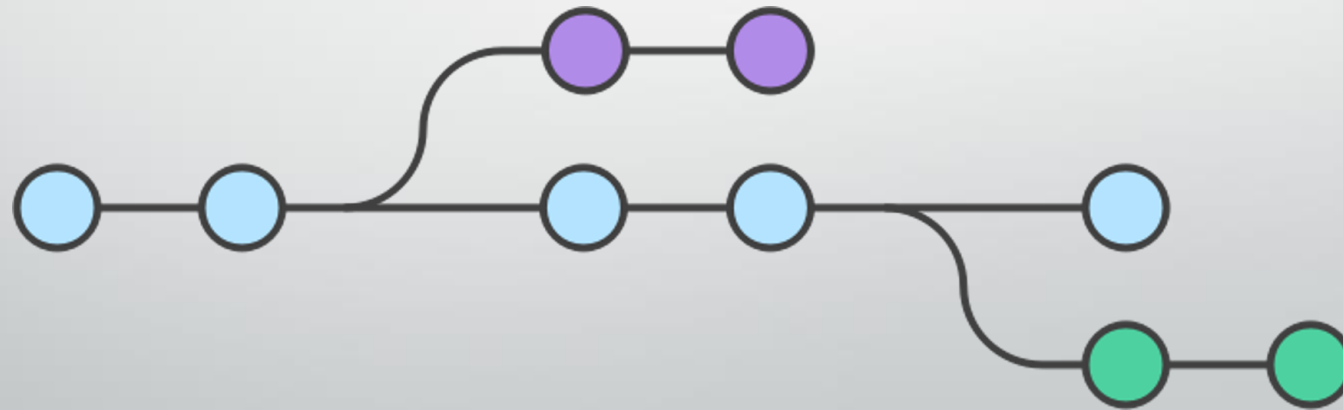
Git branches

- Let's say we want to add a new feature to our code. We don't want to impact our previous version with bugs that the new feature introduces.
- We could copy the project and work on a whole new copy, perhaps, `project_revised.py` would be a new file in the copy. Or we could branch the project – recognizing we're still working on the same project, just adding something to it.

To branch our project

- The branch command is

```
git branch [branch_name]
```



Giving branch the `-d` flag will delete the referenced branch

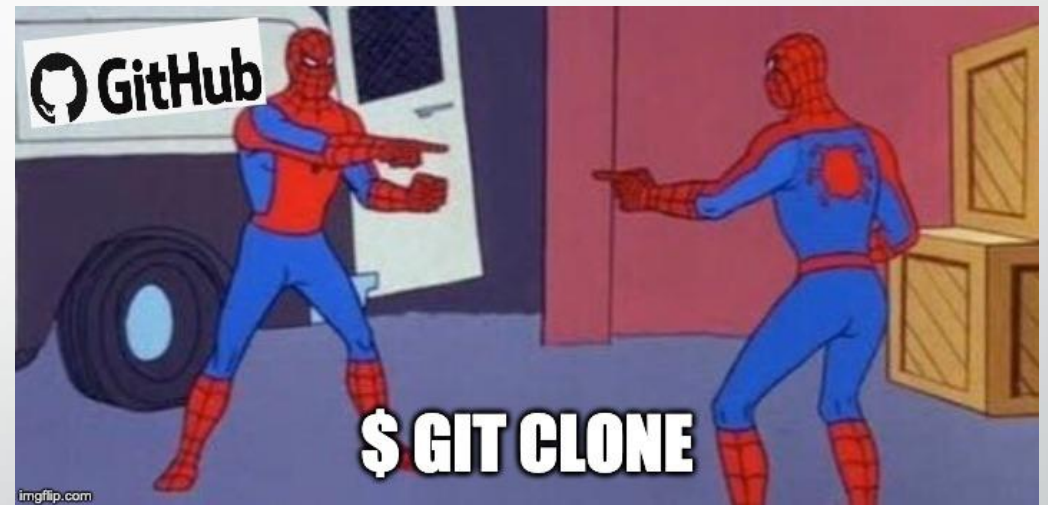
Pushing branches to remote – Pull Requests

- The output of git pushing a branch to the remote repository will either perform a merge automatically or you'll need to create a pull request.
- If there are conflicts, it is sometimes possible to resolve the conflicts through the github GUI. Typically you will need to resolve it locally and push the new version.

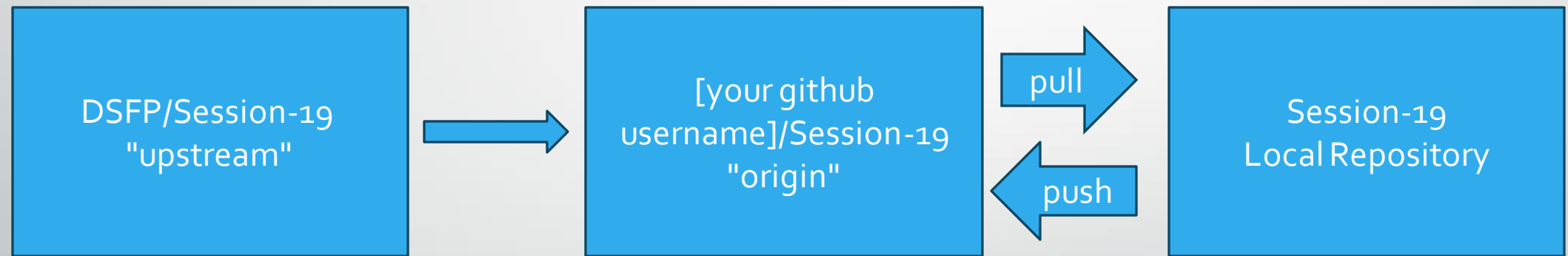
The terminology here is a little weird and confused me for a long time. Push and pull are just opposite actions based on my perspective – am I copying into my branch or copying out of my branch. Remember: git is distributed, so relationships between repos are symmetric.

Putting this into practice

We will now go through forking and cloning the DSFP Repository to create a local version on your machine.



What we want to do



Fork the Session-19 Repo

The screenshot shows the GitHub interface for the 'Session-19' repository. At the top, the repository name 'Session-19' is displayed next to the 'Public' label. Below this, there are buttons for 'Edit Pins', 'Watch' (with 3 watchers), 'Fork' (with 0 forks), and 'Star' (with 0 stars). The 'Fork' button is highlighted with a red rectangular box. Below the repository name, there are buttons for 'main' (selected), '1 branch', and '0 tags'. To the right of these buttons are 'Go to file', 'Add file', and 'Code' buttons. The main content area shows a commit by 'bscot' with the message 'ENH: Create directory structure for the week and add readme files.' and a list of files: 'day0', 'day1', 'day2', 'day3', 'day4', 'day5', and 'README.md'. The 'README.md' file is expanded, showing the title 'Session-19' and the description 'This is a repository for Session-19 (Machine Learning) of the LSSTC Data Science Fellowship Program.' On the right side, there is an 'About' section with a description of the repository, a 'Readme' button, and a list of repository statistics: 0 stars, 3 watching, and 0 forks. Below this are sections for 'Releases' and 'Packages', both indicating no published items.

LSSTC-DSFP / Session-19

Type / to search

ode Issues Pull requests Actions Projects Security Insights Settings

Session-19 Public

Edit Pins Watch 3 Fork 0 Star 0

main 1 branch 0 tags

Go to file Add file Code

bscot ENH: Create directory structure for the week and add readme files. 18e3d87 1 hour ago 2 commits

day0	ENH: Create directory structure for the week and add readme files.	1 hour ago
day1	ENH: Create directory structure for the week and add readme files.	1 hour ago
day2	ENH: Create directory structure for the week and add readme files.	1 hour ago
day3	ENH: Create directory structure for the week and add readme files.	1 hour ago
day4	ENH: Create directory structure for the week and add readme files.	1 hour ago
day5	ENH: Create directory structure for the week and add readme files.	1 hour ago
README.md	Initial commit	last month

README.md

Session-19

This is a repository for Session-19 (Machine Learning) of the LSSTC Data Science Fellowship Program.

About

This is a repository for Session-19 (Machine Learning) of the LSSTC Data Science Fellowship Program.

Readme

Activity

0 stars

3 watching

0 forks

Report repository

Releases

No releases published

[Create a new release](#)

Packages

No packages published

[Publish your first package](#)

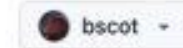
Fork the Session-19 Repo

Create a new fork

A fork is a copy of a repository. Forking a repository allows you to freely experiment with changes without affecting the original project.

Required fields are marked with an asterisk (*).

Owner *



Repository name *

/ Session-19

✔ Session-19 is available.

By default, forks are named the same as their upstream repository. You can customize the name to distinguish it further.

Description (optional)

This is my fork for Session-19 (Machine Learning) of the LSSTC Data Science Fellowship Program.

☒ Copy the `main` branch only

Contribute back to LSSTC-DSFP/Session-19 by adding your own branch. [Learn more](#).

① You are creating a fork in your personal account.

Create fork

Now clone the repository locally

The screenshot shows the GitHub interface for a repository named 'Session-19', which is a fork of 'LSSTC-DSFP/Session-19'. The repository is public and has 1 branch (main) and 0 tags. A 'Code' dropdown menu is open, showing options to clone the repository. The 'Clone' option is selected, and the 'HTTPS' tab is active, displaying the URL `https://github.com/bscot/Session-19.git`. Below the URL, there is a note: 'Use Git or checkout with SVN using the web URL.' Other options in the dropdown include 'SSH', 'GitHub CLI', 'Open with GitHub Desktop', and 'Download ZIP'. The repository's file list is visible in the background, showing a directory structure with folders 'day0' through 'day5' and a 'README.md' file. The 'README.md' file is selected, and its content is displayed at the bottom of the screen.

Session-19 Public
forked from LSSTC-DSFP/Session-19

main 1 branch 0 tags

This branch is up to date with LSSTC-DSFP/Session-19:main.

bscot ENH: Create directory structure for the week and add readme

- day0 ENH: Create directory structure for the week and add readme files.
- day1 ENH: Create directory structure for the week and add readme files.
- day2 ENH: Create directory structure for the week and add readme files.
- day3 ENH: Create directory structure for the week and add readme files.
- day4 ENH: Create directory structure for the week and add readme files.
- day5 ENH: Create directory structure for the week and add readme files. 1 hour ago
- README.md Initial commit last month

README.md

Session-19

This is a repository for Session-19 (Machine Learning) of the LSSTC Data Science Fellowship Program.

Now clone the repository locally

git clone [link you copied]

```
(base) bryan@Bryans-MacBook-Pro-8 Documents % git clone https://github.com/bscot/Session-19.git
Cloning into 'Session-19'...
remote: Enumerating objects: 7, done.
remote: Counting objects: 100% (7/7), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 7 (delta 0), reused 4 (delta 0), pack-reused 0
Receiving objects: 100% (7/7), done.
(base) bryan@Bryans-MacBook-Pro-8 Documents %
```

Finally, let's link this to the 'official' Session-19 Repo

```
git remote add upstream https://github.com/LSSTC-DSFP/Session-19.git
```

```
(base) bryan@Bryans-MacBook-Pro-8 Session-19 % git remote  
origin  
upstream
```

```
git remote pull upstream main
```

```
[(base) bryan@Bryans-MacBook-Pro-8 Session-19 % git pull upstream main  
From https://github.com/LSSTC-DSFP/Session-19  
* branch          main          -> FETCH_HEAD  
Already up to date.
```

Using git in the DSFP

Before each day, pull from the upstream repository to get the day's materials (and any other new material from the previous day)

- You can either pull from upstream directly ("git pull upstream main") [you'll then need to push to origin ("git push origin main") to bring everything up to date.
- Or use the github GUI to bring your origin repository up to date, then pull origin. [preferred]

At the end of each day (or more frequently), push your changes/saved work to origin

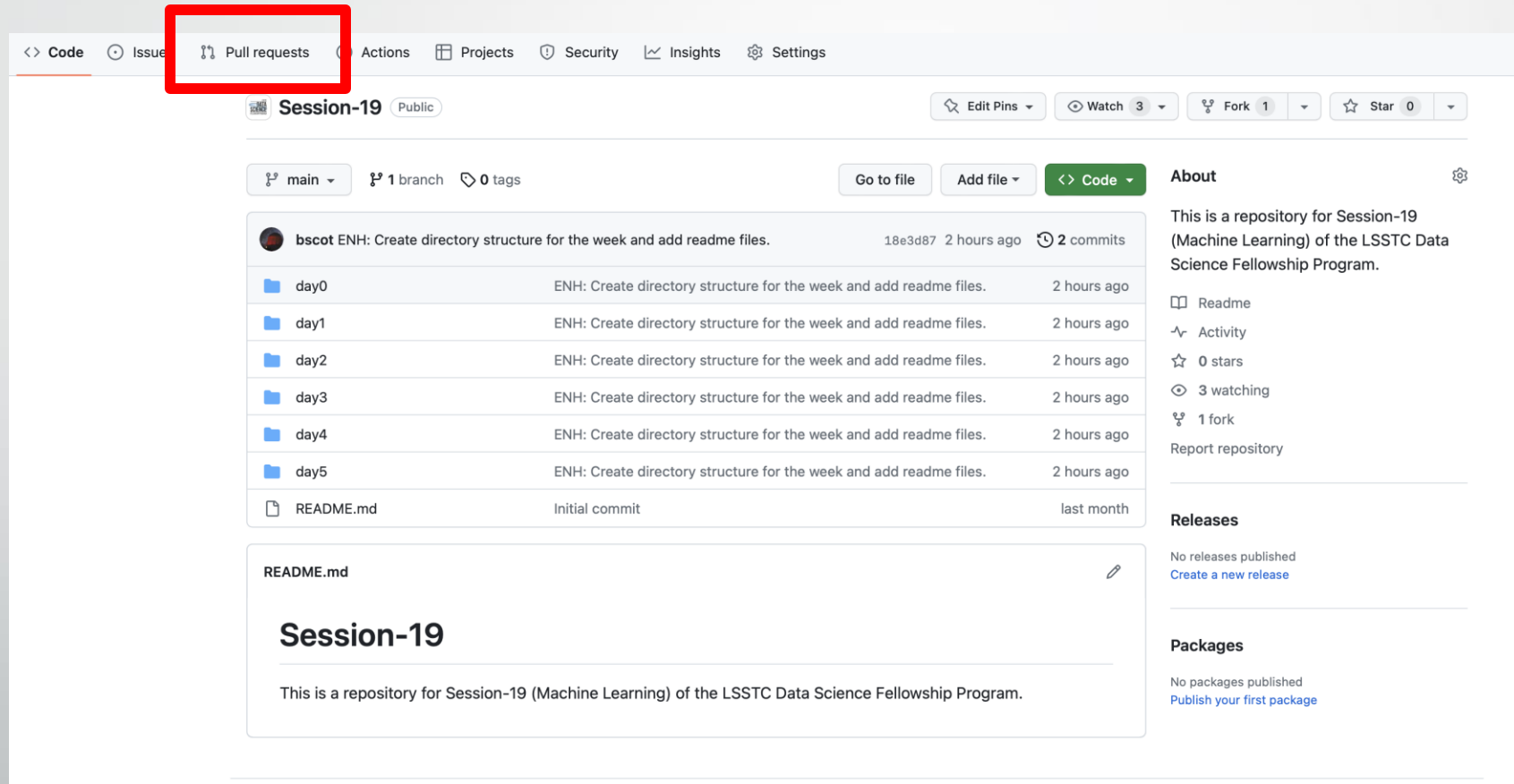
- "git push origin main"
- You may have merge conflicts – if two commits differ on their version of a line of a code – we'll discuss how to handle merge conflicts in detail at a future session.

Bonus: Pull Requests

For this next part, we'll put you pairs in breakout rooms.

- One of you should begin by forking the other's Session-19 repository.
- Then make an edit to the README.md file to add your name as an author of the repository
- Once you've done that, open a pull request in the original repository to merge in the changed README.md
- The other partner should then review the pull request to confirm they agree with the changes and approve or deny the pull.

Pull Requests




The screenshot shows the GitHub interface for a repository named "Session-19". The "Pull requests" tab is highlighted with a red box in the top navigation bar. The repository is public and has 3 watchers, 1 fork, and 0 stars. The main content area shows a commit history table with columns for the commit message, commit hash, and time ago. The commit messages are "ENH: Create directory structure for the week and add readme files." for each day from day0 to day5, and "Initial commit" for the README.md file. The README.md file is shown below the table, with the title "Session-19" and the description "This is a repository for Session-19 (Machine Learning) of the LSSTC Data Science Fellowship Program." The right sidebar contains the "About" section, which describes the repository as being for the LSSTC Data Science Fellowship Program, and the "Releases" and "Packages" sections, both of which indicate that no releases or packages have been published.

<> Code Issue Pull requests Actions Projects Security Insights Settings

Session-19 Public

Edit Pins Watch 3 Fork 1 Star 0

main 1 branch 0 tags Go to file Add file <> Code

 bscot	ENH: Create directory structure for the week and add readme files.	18e3d87 2 hours ago	2 commits
day0	ENH: Create directory structure for the week and add readme files.	2 hours ago	
day1	ENH: Create directory structure for the week and add readme files.	2 hours ago	
day2	ENH: Create directory structure for the week and add readme files.	2 hours ago	
day3	ENH: Create directory structure for the week and add readme files.	2 hours ago	
day4	ENH: Create directory structure for the week and add readme files.	2 hours ago	
day5	ENH: Create directory structure for the week and add readme files.	2 hours ago	
README.md	Initial commit	last month	

README.md

Session-19

This is a repository for Session-19 (Machine Learning) of the LSSTC Data Science Fellowship Program.

About

This is a repository for Session-19 (Machine Learning) of the LSSTC Data Science Fellowship Program.

Readme Activity 0 stars 3 watching 1 fork Report repository

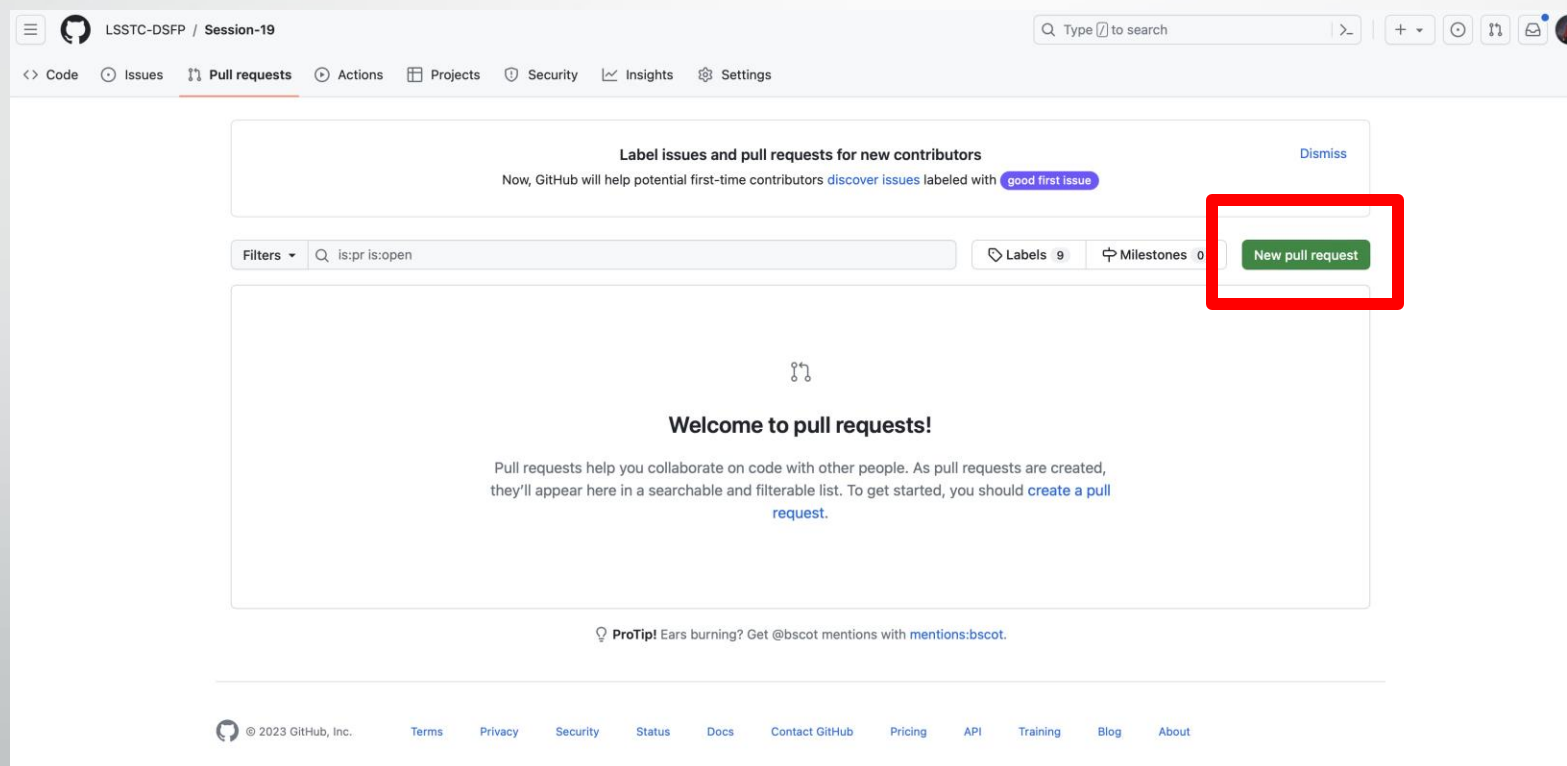
Releases

No releases published
[Create a new release](#)

Packages

No packages published
[Publish your first package](#)

Pull Requests



Create The Pull Request

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](#).

base repository: LSSTC-DSFP/Session-19

base: main

head repository: bscot/Session-19

compare: main

✓ **Able to merge.** These branches can be automatically merged.

Discuss and review the changes in this comparison with others. [Learn about pull requests](#)

Create pull request

1 commit1 file changed1 contributor

Commits on Aug 30, 2023

Update README.md

bscot committed 1 minute ago

Verified

3a

Showing 1 changed file with 2 additions and 0 deletions.

2 README.md

@@ -1,2 +1,4 @@

1 1 # Session-19

2 2 This is a repository for Session-19 (Machine Learning) of the LSSTC Data Science Fellowship Program.

3 +

4 + Author: Bryan Scott

Open a pull request

Create a new pull request by comparing changes across two branches. If you need to, you can also [compare across forks](#).

base repository: LSSTC-DSFP/Session-19

base: main

head repository: bscot/Session-19

compare: main

✓ **Able to merge.** These branches can be automatically merged.

Update README.md

WritePreview

Leave a comment

Attach files by dragging & dropping, selecting or pasting them.

☒ Allow edits by maintainers

Create pull request

Remember, contributions to this repository should follow our [GitHub Community Guidelines](#).

1 commit1 file changed1 contributor

Reviewers

No reviews—at least 1 approving review is required.

Assignees

No one—assign yourself

Labels

None yet

Projects

None yet

Milestone

No milestone

Development



Use [Closing keywords](#) in the description to automatically close issues





Helpful resources



[GitHub Community Guidelines](#)

Approving the Pull Request


Update README.md #1



 **Open** bscot wants to merge 1 commit into `LSSTC-DSFP:main` from `bscot:main` 

 Conversation **0**  Commits **1**  Checks **0**  Files changed **1**


 **bscot** commented now Member 


Added myself as an author to this repository.




  Update README.md Verified `3a951e3`

Add more commits by pushing to the `main` branch on `bscot/Session-19`.



 **Review required**


At least 1 approving review is required by reviewers with write access. [Learn more](#).

 **Merging is blocked**


Merging can be performed automatically with 1 approving review.

☐ **Merge without waiting for requirements to be met (bypass branch protections)**

Merge pull request




You can also [open this in GitHub Desktop](#) or [view command line instructions](#).





Write


Preview



Leave a comment



Attach files by dragging & dropping, selecting or pasting them. 

 **Close pull request**

Comment

Git "muscle memory"

`git add [filename]` # adds a file so that git tracks it

`git status` # shows what files have been added & modified + more

`git commit -m [brief description]` # commits the file to your repo

`git push origin main` # sends your repo to the remote repository





Summary: Git and Version Control

Version control gives you a 'global undo' button to revert changes in your code. It also helps make your code more open and reproducible, which makes your (and everyone else's) science better!

We've walked through cloning and forking the DSFP Session 19 repository, as well as how to push and pull to/from your fork/branch of it. For each session, we'll ask you to repeat this.

Summary: Git and Version Control

Version control gives you a 'global undo' button to revert changes in your code. It also helps make your code more open and reproducible, which makes your (and everyone else's) science better!

We've walked through cloning and forking the DSFP Session 19 repository, as well as how to push and pull to/from your fork/branch of it. For each session, we'll ask you to repeat this.

