

Introduction to GitHub

Python-do-ECARES

Fabrizio Leone

Why GitHub, Why Here

- ▶ GitHub helps to achieve reproducibility.
- ▶ Reproducibility is key for credible empirical research.

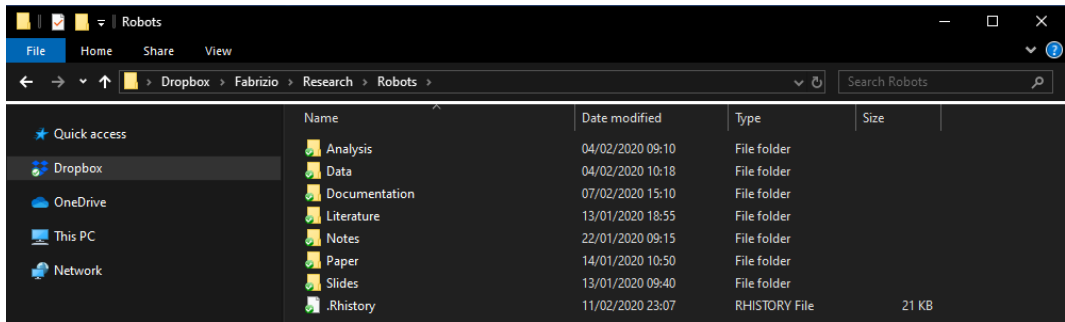
Goals of This Session

- ▶ Understand what is GitHub.
- ▶ Learn how to use GitHub for this course.
- ▶ Get an idea of how you can use GitHub in your own research.

Some Background

- ▶ GitHub has long been the largest open source platforms for versioning. From 2018, it belongs to Microsoft.
- ▶ It has over 37M users.
- ▶ Most of the software and packages that you use are on GitHub. Python is among them.
- ▶ We use GitHub to organize our work throughout the course.

Combine Version Control and Cloud Services



I only push “Analysis” to GitHub. The rest is on the cloud.

First steps

Essential Vocabulary

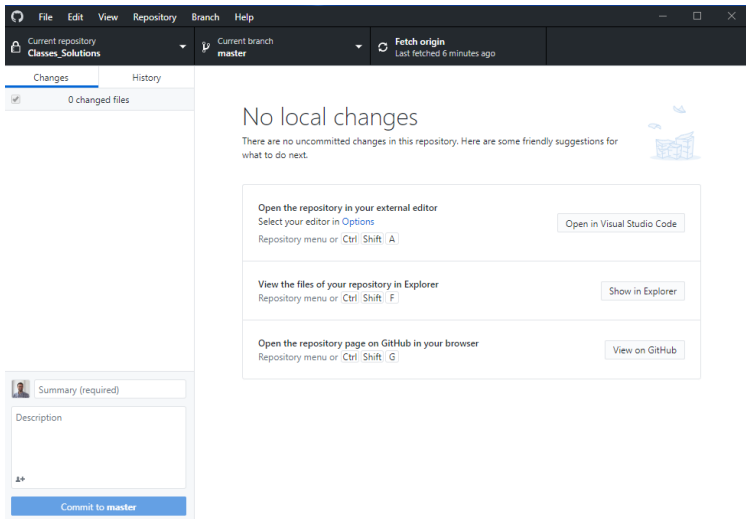
- ▶ **Repository** are the containers of your project. They can include folders, files, images, etc. Each repository is made of one or more branches.
- ▶ The default **branch** of your repository is called *master*. One can copy the master into other branches. In this way, you can work on multiple versions of the same file in parallel.
- ▶ **Commits** are changes you make to files contained in branches.
- ▶ Commits are only displayed locally until you **push** them to a branch online.
- ▶ You can **merge** branches by opening **pull** requests.

You can read more about these concepts [here](#).

First Steps With GitHub Desktop

- ▶ You should already have (1) opened a GitHub account and (2) downloaded GitHub Desktop.
- ▶ If not, please do it now [here](#) and [here](#).

Desktop Interface



Desktop Interface

Please take a few moments to familiarise with GitHub Desktop. It allows you to

- ▶ Manage existing repositories, create versions, see history of changes, revert changes (this class).
- ▶ Create and manage new repositories (homework).
- ▶ Collaborate with other people (try it yourself).

This Course

Use GitHub Desktop For This Course

- ▶ This section shows how to clone a repository, how to create your own version and how to commit and push changes to it.
- ▶ Please, refer to this procedure when working on your homework.
- ▶ Make sure to be familiar with each step. You will do this many times in the future.

Step 1. Clone Repository

Go to the Classes repository on GitHub and click on the "Clone or download" button.

Python-do-ECARES / Classes

Watch 1

Star 2

Fork 1

Code

Issues 0

Pull requests 0

Projects 0

Wiki

Security

Insights

This repository contains codes and materials of each meeting.

[meeting](#)
[schedule](#)
[material](#)

139 commits

1 branch

0 packages

0 releases

3 contributors

Apache 2.0

Branch: master

New pull request

Create new file

Upload files

Find file

Clone or download

Fab993 Update README.md

Latest commit f09605d yesterday

Session_1	Add files via upload	5 days ago
Session_10	Update Readme.md	19 days ago

Step 1.A. Open Repository

Choose "Open in Desktop".

Python-do-ECARES / Classes

Watch 1 Star 2 Fork 1

<> Code Issues 0 Pull requests 0 Projects 0 Wiki Security Insights

This repository contains codes and materials of each meeting.

meeting schedule material

139 commits 1 branch 0 packages 0 releases 3 contributors Apache-2.0

Branch: master New pull request Create new file Upload files Find file Clone or download

Fab993 Update README.md

Session_1	Add files via upload
Session_10	Update Readme.md
Session_11	MINOR UPDATE
Session_12	MINOR UPDATE

Clone with HTTPS

Use Git or checkout with SVN using the web URL.

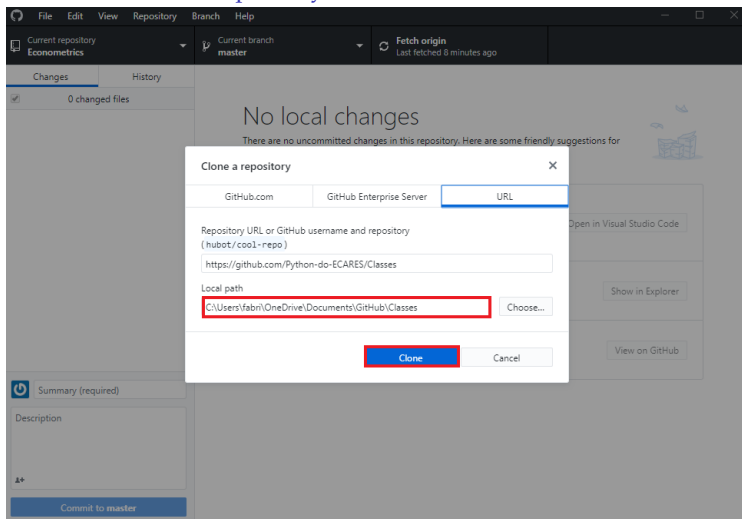
https://github.com/Python-do-ECARES/Clas

Open in Desktop

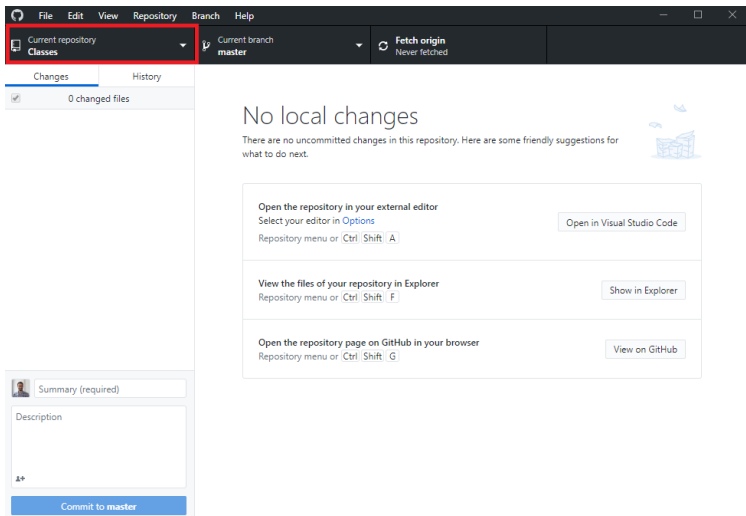
Download ZIP

Step 1.B. Clone Repository

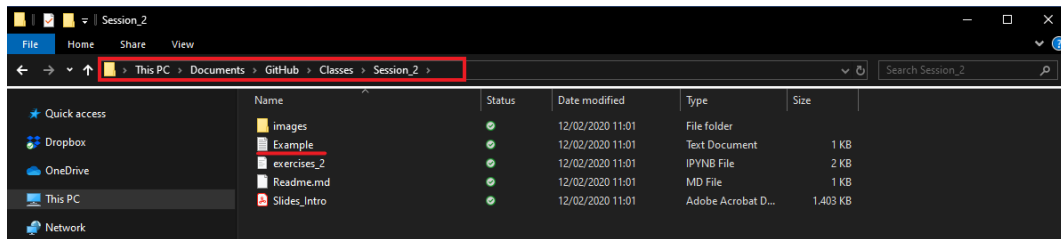
Check the local path where to clone the repository and hit "Clone".



Step 1.C. Visualize Repository on GitHub Desktop

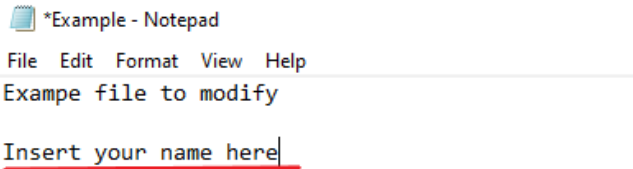


Step 2. Go to the Local Path on Your Computer

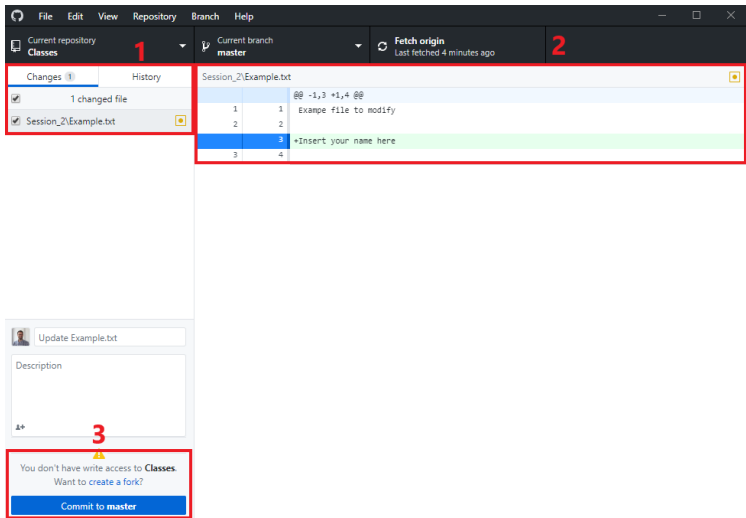


Step 2.A Modify a File

Open Example.txt, type in your name and **save**

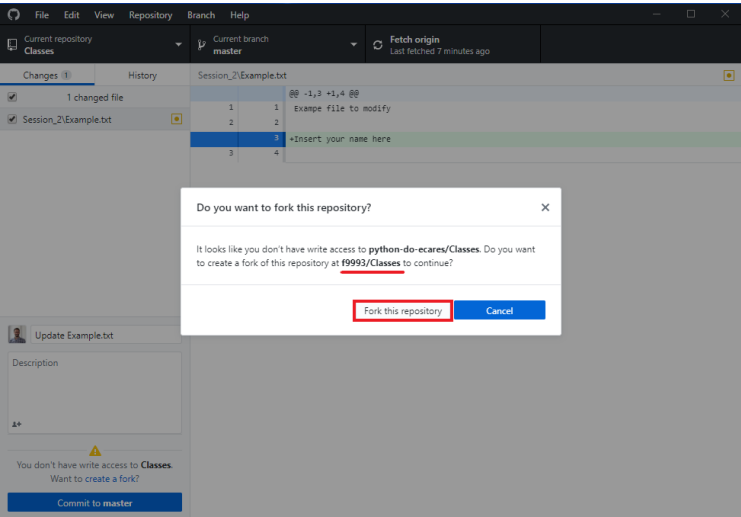


Step 2.B Changes Will Appear on GitHub Desktop As Follows



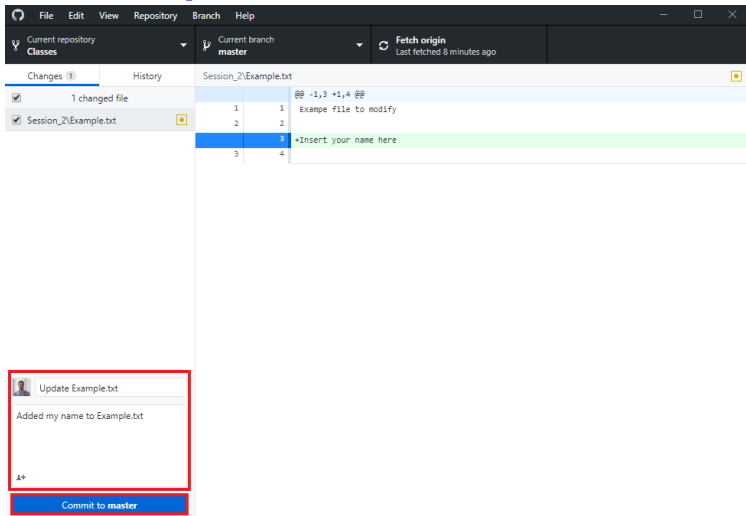
Step 2.C Fork The Master on Your Account

You cannot directly push to the Master. Only administrators can.

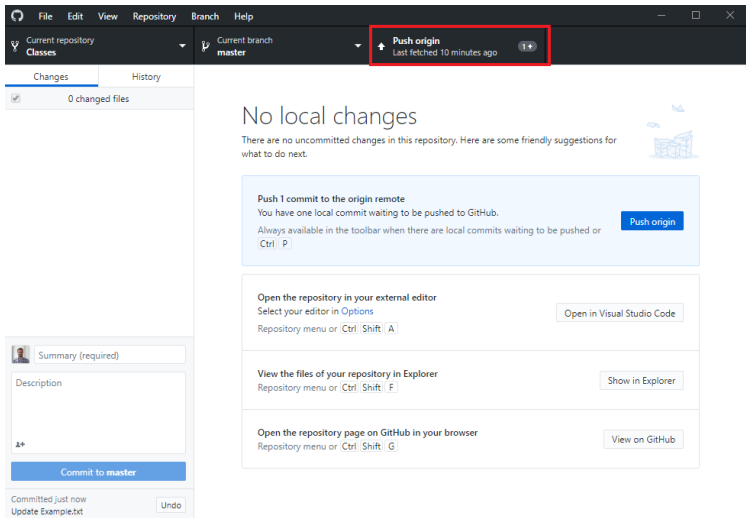


Step 2.D Commit to Your Master

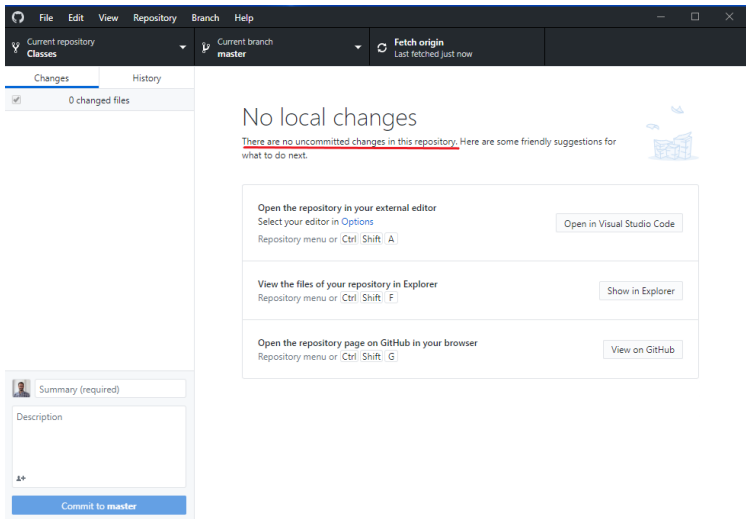
Give an informative name and description



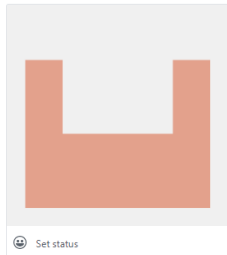
Step 2.D Push Changes Online



Step 2.E Done!



What Happens Online?



😊 Set status

f9993

Edit profile

🕒 Joined 28 days ago

Overview

Repositories 1

Projects 0

Packages 0

Stars 0

Followers 0

Following 0

Popular repositories

Customize your pins

Classes

Forked from Python-do-ECARES/Classes

This repository contains codes and materials of each meeting.

● Jupyter Notebook

3 contributions in the last year

Contribution settings ▼



[Learn how we count contributions.](#)

See Your Repository

f9993 / **Classes**

Watch 0 Star 0 Fork 2

forked from [Python-do-ECARES/Classes](#)

Code Pull requests 0 Actions Projects 0 Wiki Security Insights Settings

Branch: master Classes / Session_2 /

Create new file Upload files Find file History

This branch is 1 commit ahead of Python-do-ECARES:master.

Pull request Compare

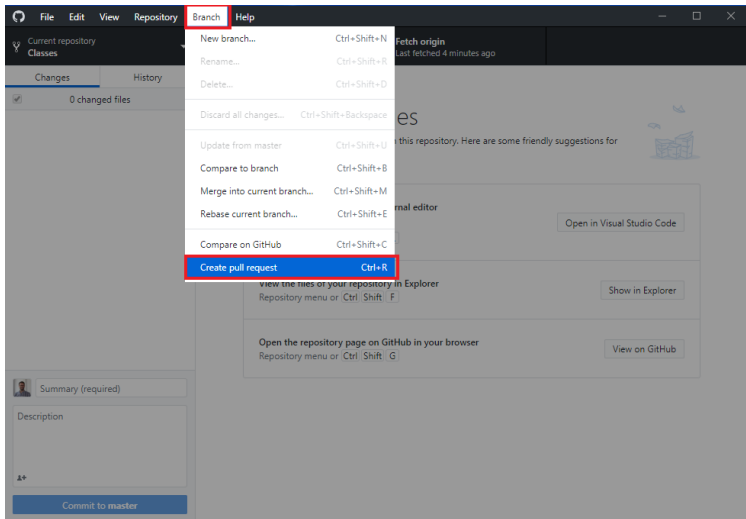
Fab993 Update Example.txt Latest commit f3d5b82 2 minutes ago

..		
images	update slides	28 days ago
Example.txt	Update Example.txt	2 minutes ago
Readme.md	Update Readme.md	28 days ago
Slides_Intro.pdf	MINOR UPDATE	19 days ago
exercises_2.ipynb	BUG FIXES	19 days ago

Next Steps

- ▶ You can now edit, commit and push changes to your copy of the course repository.
- ▶ When you are done with the homework, **pull** changes to the original master.

Step 3. Open a Pull Request



Step 3.A You Will Be Redirected to GitHub Online

Python-do-ECARES / Classes Watch 2 Star 3 Fork 2

Code Issues 0 Pull requests 0 Projects 0 Wiki Security Insights

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: Python-do-ECARES/Classes

base: master

head repository: f9993/Classes

compare: master

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

Create pull request

 Discover how the changes in this comparison with others.

1 commit

1 file changed

0 commit comments

1 contributor

Commits on Feb 12, 2020

Fab993

Update Example.txt

f3d5b82

Showing 1 changed file with 1 addition and 0 deletions. Unified Split

1 Session_2/Example.txt

... @@ -1,3 +1,4 @@

1 1 Exampe file to modify

2 2

3 + Insert your name here

3 4

Step 3.B Document Your Pull Request

Python-do-ECARES / Classes

Watch 2Star 3Fork 2

CodeIssues 0Pull requests 0Projects 0WikiSecurityInsights

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: Python-do-ECARES/Classes

base: master

head repository: f9993/Classes

compare: master

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

Commit of Name Surname

WritePreview

AA B i “ < > ↺ ⋮ ⋮ ⋮ @ 📎 ↻

Hi.

I am adding my proposed change to file Example.txt to your master file.

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

☒ Allow edits from maintainers. [Learn more](#)

Create pull request

Helpful resources

[GitHub Community Guidelines](#)

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

Step 3.C Wait For Review

[Python-do-ECARES](#) / [Classes](#)

Watch 2

Star 3

Fork 2

<> Code ① Issues ① Pull requests ① Projects ① Wiki Security Insights

Commit of Name Surname #3

Edit

Open f9993 wants to merge 1 commit into [Python-do-ECARES:master](#) from [f9993:master](#)

Conversation ①Commits ①Checks ①Files changed ①+1 -0

f9993 commented now

+ 👤 ...

Hi,

I am adding my proposed change to file Example.txt to your master file.

Update Example.txt f3d5b82

Add more commits by pushing to the **master** branch on [f9993/Classes](#).

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.

Reviewers

Suggestions

Fab993 [Request](#)

At least 1 approving review is required to merge this pull request.

Assignees

No one assigned

Labels

None yet

Projects

None yet

Milestone

Bonus. What Happens to the Original Repository?

Only Admins can merge pull requests

Python-do-ECARES / Classes

Watch 2 Star 3 Fork 2

<> Code Issues 0 Pull requests 1 Projects 0 Wiki Security Insights Settings

Commit of Name Surname #3

Edit

Open f9993 wants to merge 1 commit into Python-do-ECARES:master from f9993:master

Conversation 0 Commits 1 Checks 0 Files changed 1 +1 -0

f9993 commented 2 minutes ago

First-time contributor

Hi,

I am adding my proposed change to file Example.txt to your master file.

Update Example.txt f3d5b82

Reviewers

Suggestions

Fab993 Request

At least 1 approving review is required to merge this pull request.

Assignees

No one—assign yourself

Labels

None yet

Projects

None yet

Milestone

No milestone

Linked issues

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.

As an administrator, you may still merge this pull request.

Merge pull request

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

Bonus. What Happens to the Original Repository?

Check changes that people made on the original file

Python-do-ECARES / Classes

Watch 2

Star 3

Fork 2

<> Code

Issues 0

Pull requests 1

Projects 0

Wiki

Security

Insights

Settings

Commit of Name Surname #3

Edit

Open

f9993 wants to merge 1 commit into Python-do-ECARES:master from f9993:master

Conversation 0

Commits 1

Checks 0

Files changed 1

+1 -0

Changes from all commits

File filter...

Jump to...

0 / 1 files viewed

Review changes

1 Session_2/Example.txt

Viewed

@@ -1,3 +1,4 @@

1 1 Exame file to modify

2 2

3 + Insert your name here

3 4

ProTip! Use `n` and `p` to navigate between commits in a pull request.

Bonus. What Happens to the Original Repository?

Rejecting a pull request

Commit of Name Surname #3

Open

 f9993 wants to merge 1 commit into python-do-EGAPES/master from f9993/master

Conversation 0

Commits 1

Checks 0

Files changed 1

+1 -0

f9993 commented 5 minutes ago

First-time contributor

+ @

...

Hi,

I am adding my proposed change to file Example.txt to your master file.

Update Example.txt

#3d5b2

Add more commits by pushing to the **master** branch on **f9993/Classes**.

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.

As an administrator, you may still merge this pull request.

Merge pull request

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

Write

Preview

AA B i “ < > ☰ ☷ ☰ ☷ @ 📎 ↻

Hi,

I am sorry, but I cannot accept your request because x, y and z.

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

Close and comment

Comment

Reviewers

Suggestions

Fab993 [Request](#)

At least 1 approving review is required to merge this pull request.

Assignees

No one—assign yourself

Labels

None yet

Projects

None yet

Milestone

No milestone

Linked issues

Successfully merging this pull request may close these issues.

None yet

Notifications

Customize

Subscribe

You're not receiving notifications from this thread.

2 participants

Done

Python-do-ECARES / Classes

Watch 2

Star 3

Fork 2

<> Code

ⓘ Issues 0

🔗 Pull requests 0

📁 Projects 0

📖 Wiki

🔒 Security

📊 Insights

⚙ Settings

Label issues and pull requests for new contributors

Dismiss

Now, GitHub will help potential first-time contributors [discover issues](#) labeled with **good first issue**

Filters

is:pr is:open

🏷 Labels 9

📅 Milestones 0

New pull request

📦 0 Open ✓ 3 Closed

Author ▾

Label ▾


Projects ▾

Milestones ▾

Reviews ▾

Assignee ▾

Sort ▾



There aren't any open pull requests.

You could search [all of GitHub](#) or try an [advanced search](#).

What Did It Happen?

- ▶ You have **forked** a repo, **committed** to your own version of it and **pulled** your changes to the owner.
- ▶ This is the “GitHub way” of copying the assignment questions, work on it at home and submit it in class.
- ▶ You can scroll pull requests under any repository and see the list of changes that people is proposing. In this course, you can scroll pull requests to see what other people have done.
- ▶ We will only merge into the master a few things (e.g. slides, solution). All other pull request will remain open until the end of the course.
- ▶ Software developers seem to work alike.

Update Your Fork

Make sure that your fork is up to date with the origin

- ▶ If you only fork once, you don't know if the upstream branch has been updated.
- ▶ In this course, we will commit to the original master to upload homework, slides and solutions.
- ▶ You need to make sure to have the latest copy of the original branch.

Step 4. How to Update a Fork

Go to the master of your forked repository and hit “compare”

Fab993 / grf

forked from grf-labs/grf

Watch 0

Star 0

Fork 105

<> Code

Pull requests 0

Actions

Projects 0

Wiki

Security

Insights

Settings

Generalized Random Forests

<https://grf-labs.github.io/grf/>

Edit

Manage topics

1,560 commits

4 branches

0 packages

13 releases

23 contributors

GPL-3.0

Branch: master

New pull request

Create new file

Upload files

Find file

Clone or download

This branch is 4 commits behind grf-labs:master.

Pull request

Compare

erikcs Remove O(node_size^2) work in smallQ splitting rule (grf-labs#592)

Latest commit 7df9202 on Dec 23, 2019

Step 4.A. Click on “Switching The Base”

If you have not modified any file, you will see the following

grf-labs / grf

Watch 40 Unstar 371 Fork 105

Code Issues 44 Pull requests 5 Actions Projects 2 Wiki Security Insights

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: grf-labs/grf ▾ base: master ▾ ◀ head repository: Fab993/grf ▾ compare: master ▾

There isn't anything to compare.

grf-labs:master is up to date with all commits from Fab993:master. Try [switching the base](#) for your comparison.

Showing 0 changed files with 0 additions and 0 deletions.

Unified	Split
<p> $\frac{1}{2} \left(\frac{1}{2} + \frac{1}{2} \right) = \frac{1}{2}$ </p>	<p> $\frac{1}{2} \left(\frac{1}{2} + \frac{1}{2} \right) = \frac{1}{2}$ </p>

No commit comments for this range

Step 4.B. Open a Pull Request to Yourself

Fab993 / grf

forked from grf-labs/grf

Watch 0Star 0Fork 105

CodePull requests 0ActionsProjects 0WikiSecurityInsightsSettings

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: Fab993/grfbase: masterhead repository: grf-labs/grfcompare: master

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

Update my fork

WritePreview

AA B i “ < > ⌂ ⌵ @ 📎 ↶

Leave a comment

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

Create pull request

Reviewers

No reviews

Assignees

No one—assign yourself

Labels

None yet

Projects

None yet

Milestone

No milestone

Step 4.C. Merge the Pull Request

You can do this because you own your fork

Fab993 / grf

forked from grf-labs/grf

Watch 0 Star 0 Fork 105

<> Code

Pull requests 1

Actions

Projects 0

Wiki

Security

Insights

Settings

Update my fork #1

Open Fab993 wants to merge 4 commits into Fab993:master from grf-labs:master

Conversation 0

Commits 4

Checks 0

Files changed 26

+4,674 -86

Fab993 commented now

No description provided.

Owner + @ ...

swagger and others added 4 commits 15 days ago

Use more stable calibration for APE in case of low variation M.hat. (grf-

Verified

75ae0bf

Add structure for local linear splitting (grf-labs#603)

Verified

d56a3b7

Sample weighted splitting (grf-labs#590)

Verified

d5cf22e

Adjust sample weighting is identical to replicating samples test (grf-

Verified

82c74f6

Continuous integration has not been set up

GitHub Actions and several other apps can be used to automatically catch bugs and enforce style.

This branch has no conflicts with the base branch

Merging can be performed automatically.

Merge pull request You can also open this in GitHub Desktop or view command line instructions.

Reviewers

No reviews

Assignees

No one—assign yourself

Labels

None yet

Projects

None yet

Milestone

No milestone

Linked issues

Successfully merging this pull request may close these issues.

None yet

Notifications

Customize

Unsubscribe

Taking Stocks

- ▶ You are now able to create and maintain **your own branch** for each session of this course.
- ▶ **You are not allowed to push to the master. If you try, you will get an error.** This is a useful feature if you manage a project and do not want collaborators to modify the master directly.
- ▶ Please remember to update your forked version of the master regularly.

Standards

Commit and Push

- ▶ **Committing** and **pushing** are two important actions that you have to familiarize with.
- ▶ *Committing changes* to a branch, means that you are creating a new version of your file.
- ▶ *Pushing changes* means, instead, that you are publishing them online on GitHub. Think of the pushing action as a way of creating different stable releases of your code.
- ▶ Only push changes online if you have made a stable change.

Commit and Local Saving

- ▶ Notice that **committing to GitHub** and **saving your file locally** are very different things.
- ▶ If you save locally, you only change the file on your device. If you commit, you add a “node” to the chain of your versions.
- ▶ With this respect, we recommend to commit changes regularly, but also to use **standards**.

Commit Standards

- ▶ We encourage you to adopt the following standards to commit tidily.
- ▶ “Summary” should be either **Minor Change**, **Major Change** or **Bug Fixes**.
The first should indicate small changes in syntax or general improvements.
The second to major modifications (e.g. add new section or function), while
the third is to notify that you have fixed some bug.
- ▶ **Description** should briefly explain what the summary refers to.

Example

- ▶ Suppose you **create a new function for data cleaning in your code**. When pushing this change to GitHub, you want to give **Major Change** as summary and "added function for data cleaning" as description.
- ▶ A tidy commit activities will create a full history of changes in GitHub that you can scroll through to check different versions of your code.
- ▶ Finally, it will also help other people to understand your work.

Browse History

Browse Through History Of Changes

With GitHub Desktop, you can also browse through the entire history of your commits. This is very helpful to

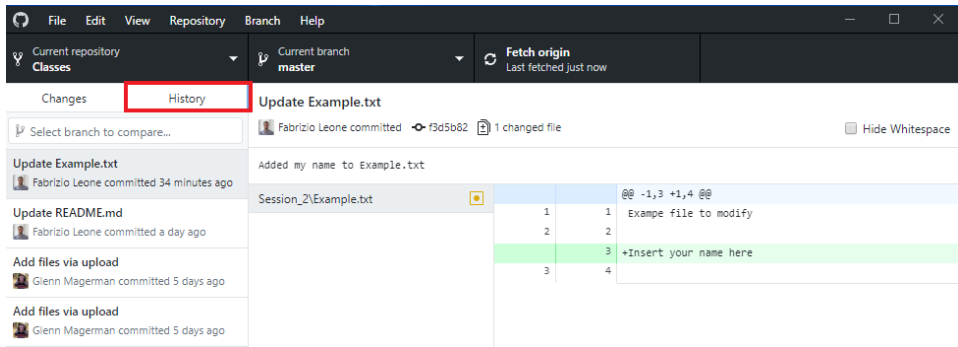
- ▶ **Revert back changes.** Imagine you update a code but, at some point, you realise that one of the previous releases was better.
- ▶ **Compare different versions.** From time to time, you may want to look back at previous versions of your work (slide, paper, code,...).

Commit Folk Theorem

A tidy committing activity will help you to easily browse through meaningful versions of your work. If you commit too often, you have to search a lot (versions only differ marginally from one another). If you commit too infrequently, you may lose information.

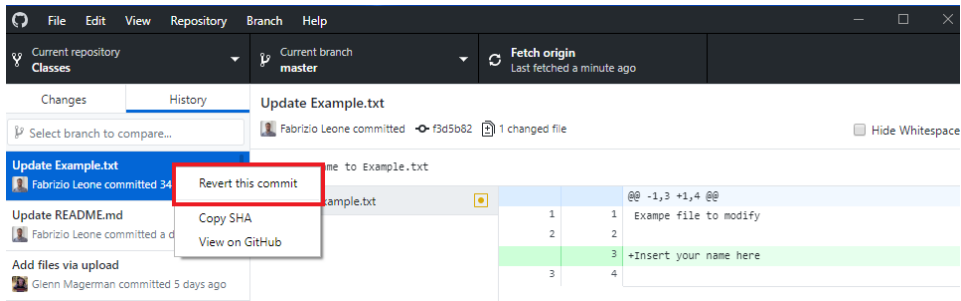
Revert Back Changes

Step 2. Open GitHub Desktop at the current repository. Then hit "History".



Revert Back Changes

Step 2. Find the node you want to restore. Click on "Revert this commit" to go back to that version.



More Details

- ▶ You can switch back and forth your versions as many times as you want.
Your local directories will change accordingly.

- ▶ If instead of "Revert this commit" you hit "View on GitHub", you can see online all the differences between two versions of your file.

Introduction	First steps	This Course	Standards	Browse History	More	Conclusion	Exercises
oooooooo	ooooo	oooooooooooooooooooooooooooooooooooo	ooooo	oooooo	●oooooooo	oo	oo

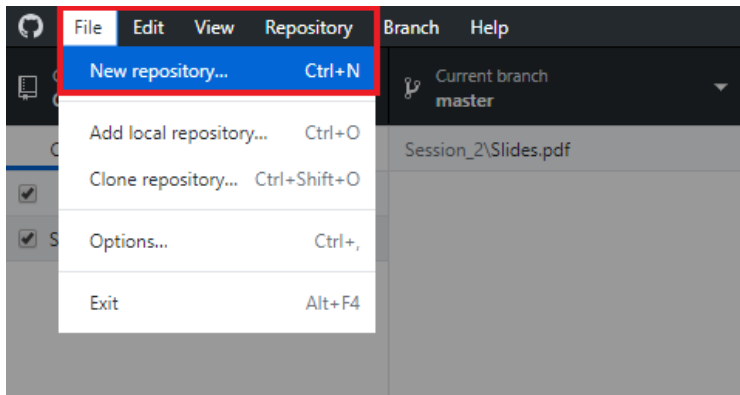
More

Create Your Repository

- ▶ You are starting a new project, or you want to upload to GitHub an existing one.
- ▶ It only takes very few steps to do it.
- ▶ Commit and push work exactly as before.

Step 1. Open GitHub Desktop

File \Rightarrow New repository



Step 2. Setup Repo

Current branch **master** Fetch origin Last fetched a minute ago

Session_2\Slides.pdf

Create a new repository X

Name

My_first_repo

Description

This repo contains x,y and z

Local path

C:\Users\fabri\OneDrive\Desktop\My_project Choose...

☒ Initialize this repository with a README

Git ignore

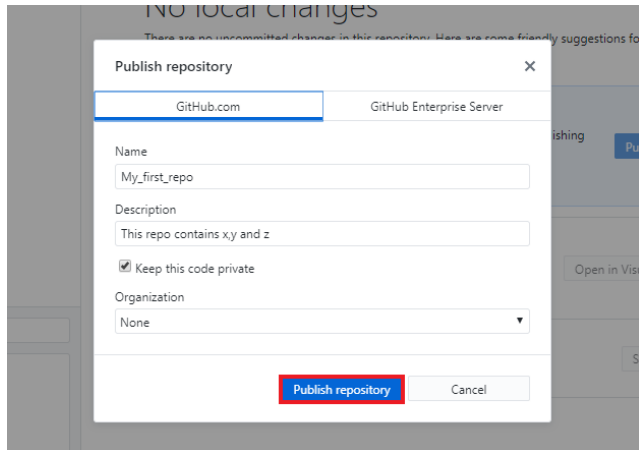
None ▼

License

None ▼

Create repository Cancel

Step 3. Publish Repo



Step 4. See Repo Online

Find a repository...

Type: All ▼

Language: All ▼

 New

My_first_repo Private

★ Star

This repo contains x,y and z

Updated 14 seconds ago

Alternative Procedure

You can also do the following

- ▶ Go to GitHub Online.
- ▶ Create a new repository.
- ▶ Clone it locally.
- ▶ Commit and push work as before.

Further Functionalities

Other useful GitHub functionalities you may want to check

- ▶ Create more **branches** under the same repository. This is particularly useful to work in parallel on different versions of your project (e.g. no more `code_version1`, `code_version2`, etc. in one folder).
- ▶ Communicate with other people (your coauthors, other developers, etc.) by creating **issues**.
- ▶ **Automatic workflow**. No need to manage your work manually.
- ▶ Create a **website**.
- ▶ Create a **Project Board**.
- ▶ GitHub integrations for **R Studio**, **VS code** and **Atom** editors.
- ▶ **Gitignore** files.

Conclusion

Conclusion

- ▶ GitHub and GitHub Desktop are powerful tools that help you to organize and make your research reproducible.
- ▶ **GitHub features a learning curve.** First commits will be quite messy: you will likely commit too often, you will create too many branches and you will be tempted to modify things from the web interface rather than from GitHub Desktop (or the terminal).
- ▶ Before adopting your own standards, **see what more experienced people do.** There are plenty of web articles that suggest best practices and standards for GitHub. Among them, find your favourite workflow.
- ▶ Use GitHub to organize your research from the very initial step until the end. Sooner or later you will like it.

Exercises

Exercises

Try at home

- ▶ Go to GitHub online and create a new repository, call it "My_first_repo".
- ▶ Clone the repository locally on your computer. Create and save there a new file.
- ▶ Create a new branch and call it "branch-try". Commit and push your changes to this branch.
- ▶ Merge the branch into the master.
- ▶ **Extra.** Under the master, open an issue an assign it to yourself. Familiarize with mentions and comments. Then close the issue.