

GitHub Tutorial

Congyi Zeng

1: Create an account

Go to <https://github.com/>

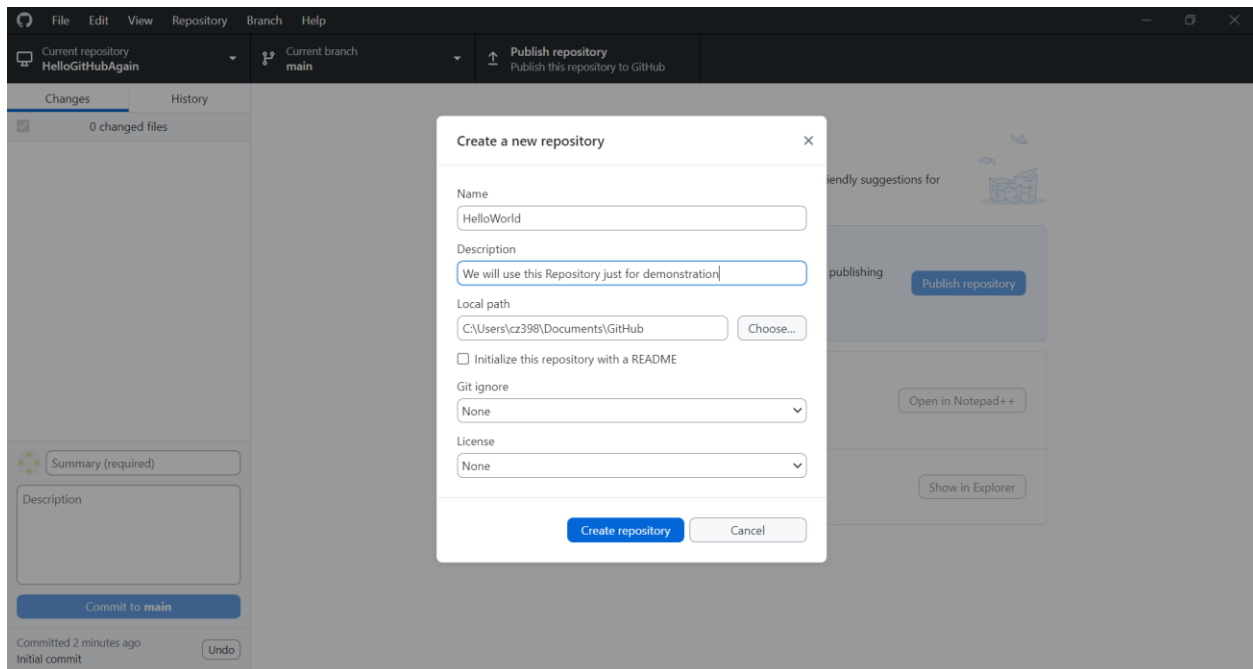
2: Download GitHub Desktop

Go to <https://desktop.github.com/>

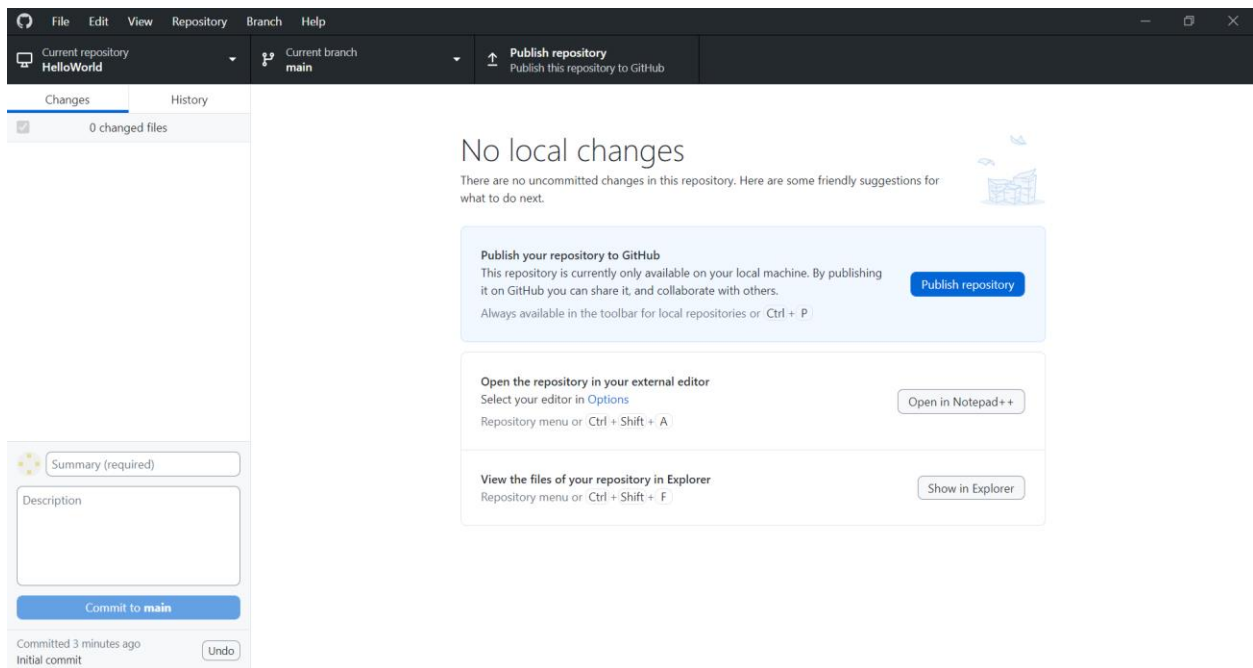
3: Create a Repository

Open GitHub Desktop -> Go to “File” -> Click “New repository”

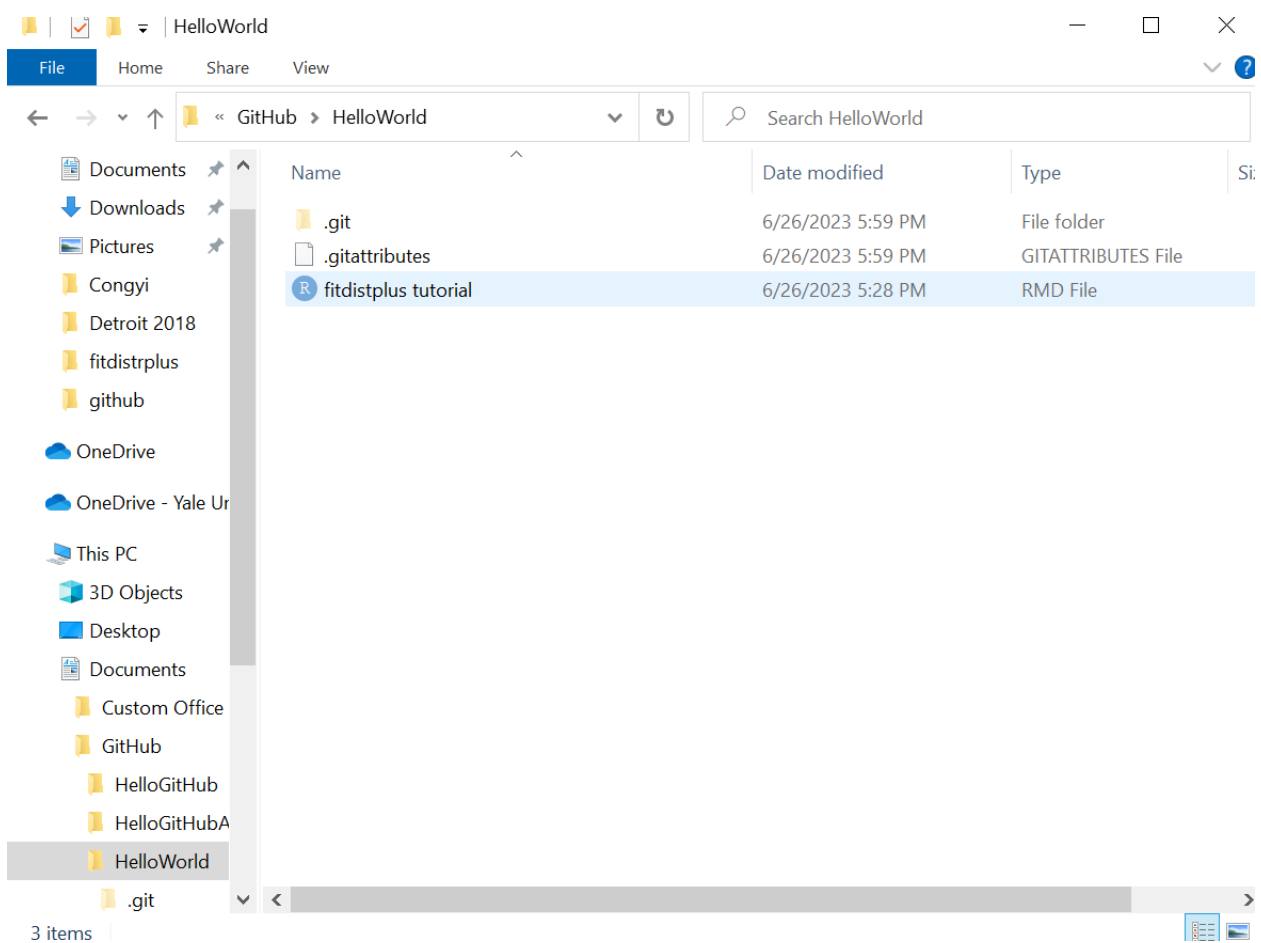
A window like this should pop up:



We created a repository with name “HelloWorld”, and this is what a repository should be looked like initially.

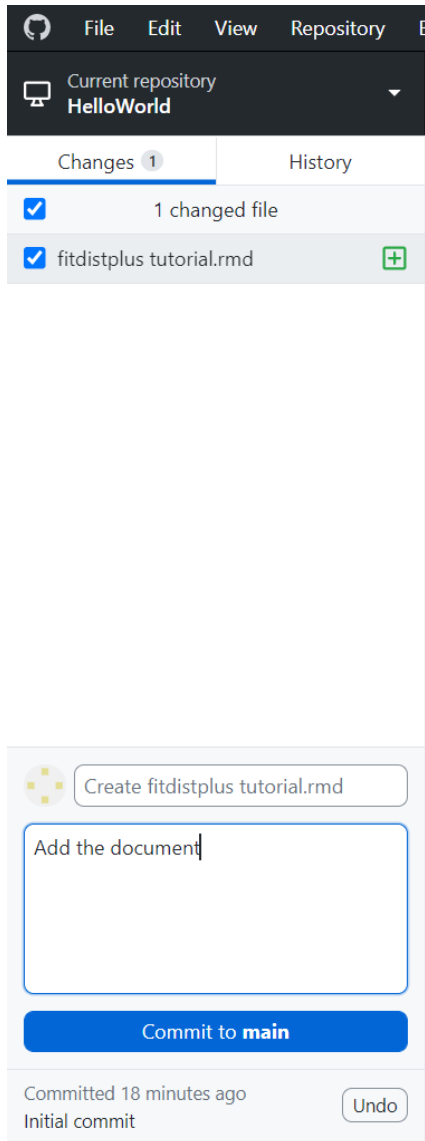


To add a file to the repository, click on “Show in Explorer”, and copy the file into the folder.



4: Making a change

After this step (or making any changes to the code), you will find GitHub Desktop show there is a change being made.



One need to click on “Commit to main” if you want the change being archived. The version before/after the change will both be saved. You could leave comments as a note for the change you made.

Now we want to make some extra changes on the code. For example, I added an illustration to the “fitdistrplus.rmd” file.

RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

Go to file/function Addins

fitdistplus tutorial.rmd

Knit on Save Knit Run

Source Visual Outline

```

214
215 We can extract the model parameters and other information to compare the performance of the
distribution. log-likelihood is a commonly used criterion, which is minimized when **MLE**
method being used when fitting the distribution.
216
217 {r}
218 gofstat(all)
219
220
221 Goodness-of-fit (which is minimized when **MGE** method is being applied) is also a criterion.
**gofstat** function returns 3 goodness-of-fit statistic (smaller is better).
222
223
221:173 MGE

```

R Markdown

After save the changes, one can find the changes being recorded by Github desktop.

File Edit View Repository Branch Help

Current repository HelloWorld

Current branch main

Fetch origin Last fetched 13 minutes ago

Changes History

1 changed file

fitdistplus tutorial.rmd

```

@@ -205,13 +205,18 @@ for(i in 1:length(all)){
205 205 shape[i] <- all[[i]]$estimate[1]]
206 206 scale[i] <- all[[i]]$estimate[2]]
207 207 log.likelihood[i] <- all[[i]]$loglik
208 -
209 208 }
210 209
211 210 data.frame(models,shape,scale,log.likelihood)
212
213 +We can extract the model parameters and other information to compare the performance of the distribution. log-likelihood is a commonly used criterion, which
is minimized when **MLE** method being used when fitting the distribution.
214
215 +{r}
216 +gofstat(all)
217 +
218
219 +Goodness-of-fit (which is minimized when **MGE** method is being applied) is also a criterion. **gofstat** function returns 3 goodness-of-fit statistic (smaller is better).
220
221
222 Usually, different estimating method would not results in large difference in distribution parameters. So using default "MLE" should be fine in most cases.

```

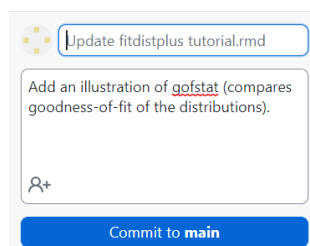
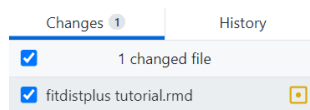
Update fitdistplus tutorial.rmd

Description

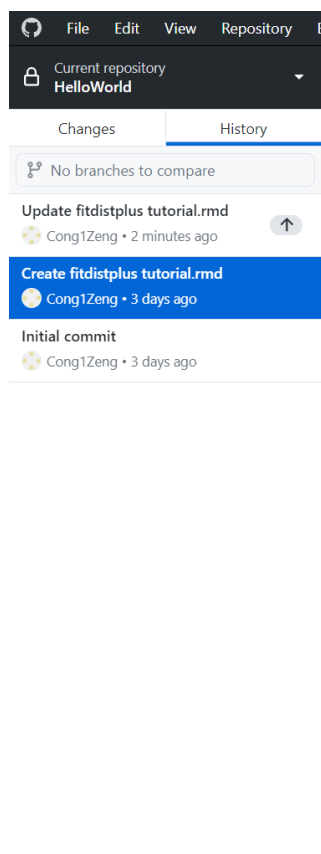
R+

Commit to main

Type in some description of the changes and Commit.

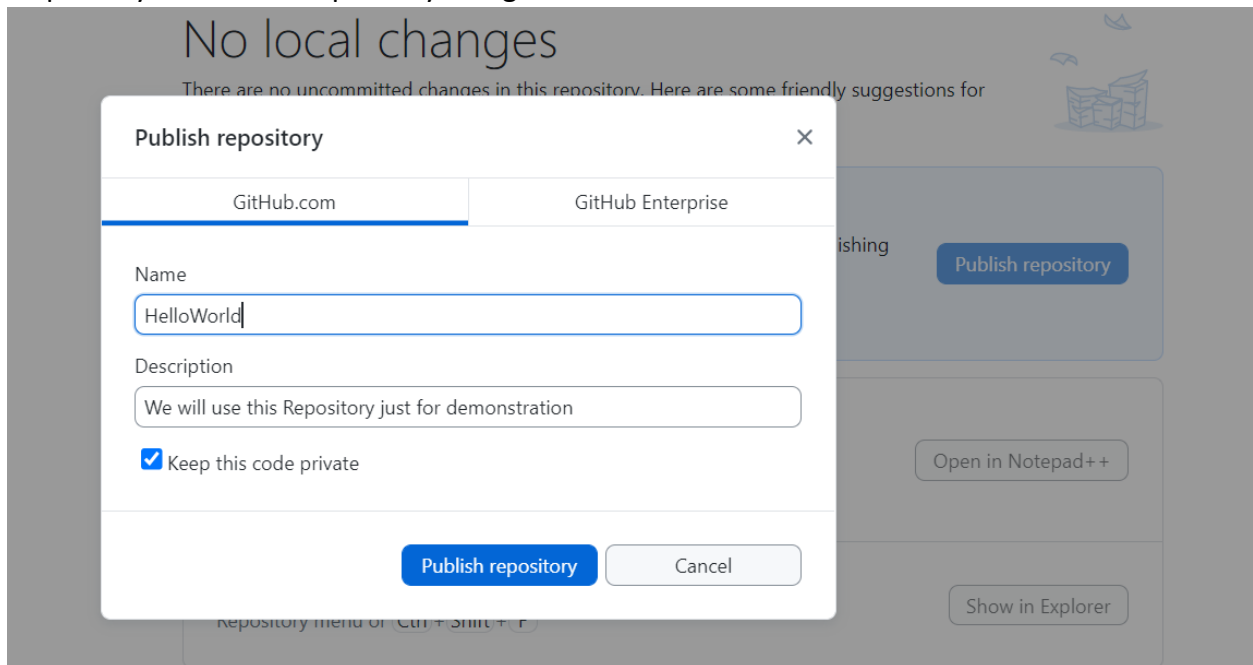


In “history”, one can view on changes and track on the historical versions.

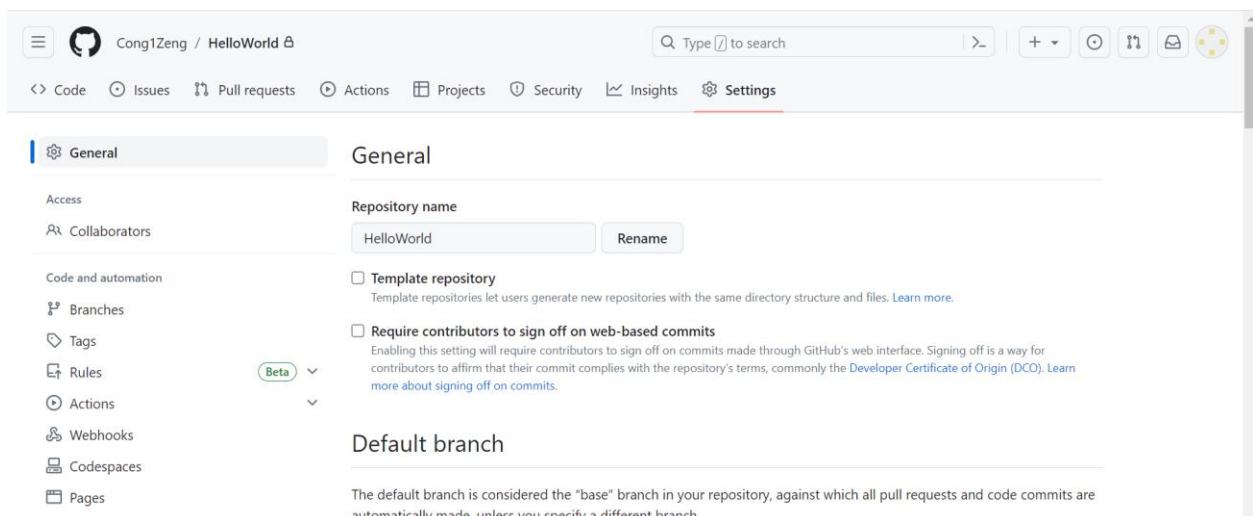


5: Publish /Push /Fetch/ Pull a Repository

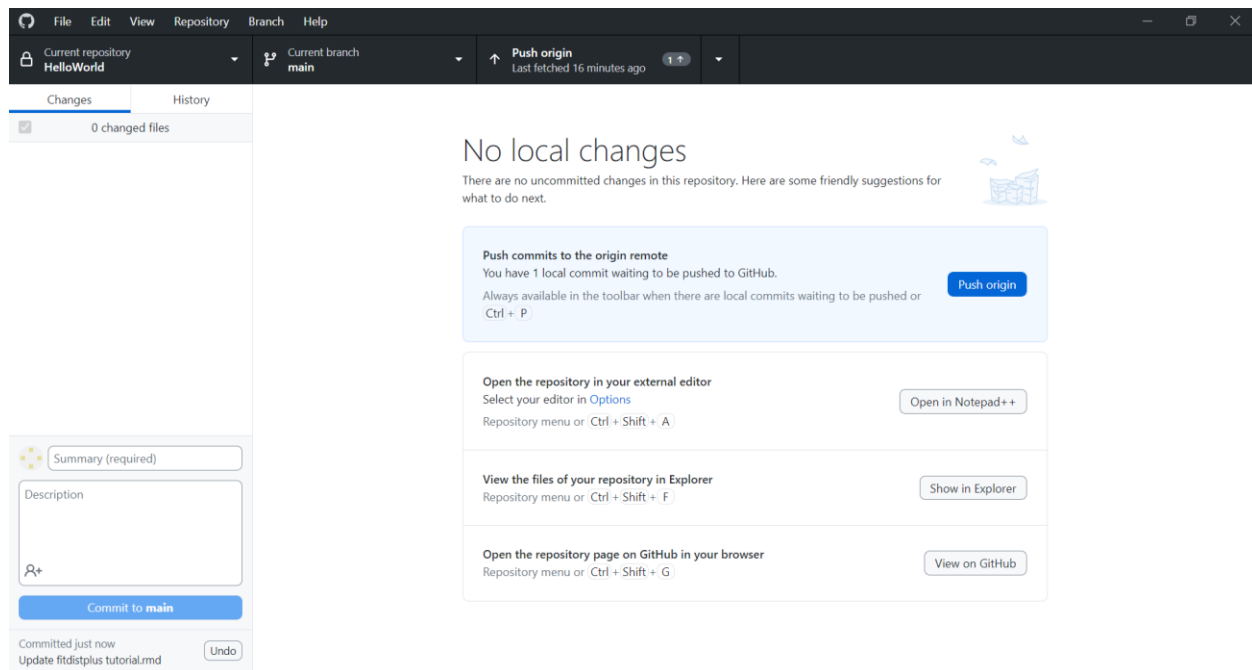
Normally we want one Repository to be accessible by multiple people. Just click on “Publish repository” to make a repository being accessible.



To add a collaborator to this repository, open the repository in browser (Repository -> View on GitHub), and click (setting -> collaborator).



“Push” is to update the changes being made locally to the repository. After changes have been committed, one can click on “Push Origin” to complete the step.



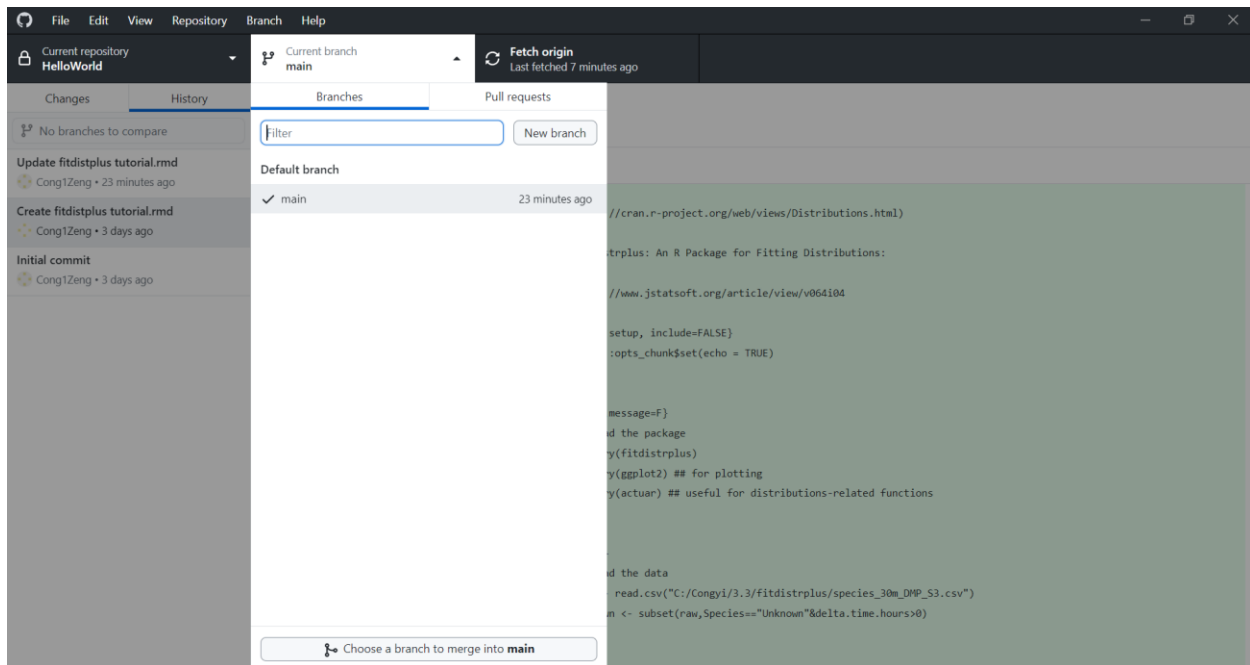
“Fetch”, is to update the local version by download new versions from repository. After this step, one can access to changes made by other collaborators.

“Pull” is an alternative of “Fetch”. Both are to update remote changes to local. The main difference between the two operations is that “Pull” make changes to the local documents, while “Fetch” do not, it just download the new data and code.

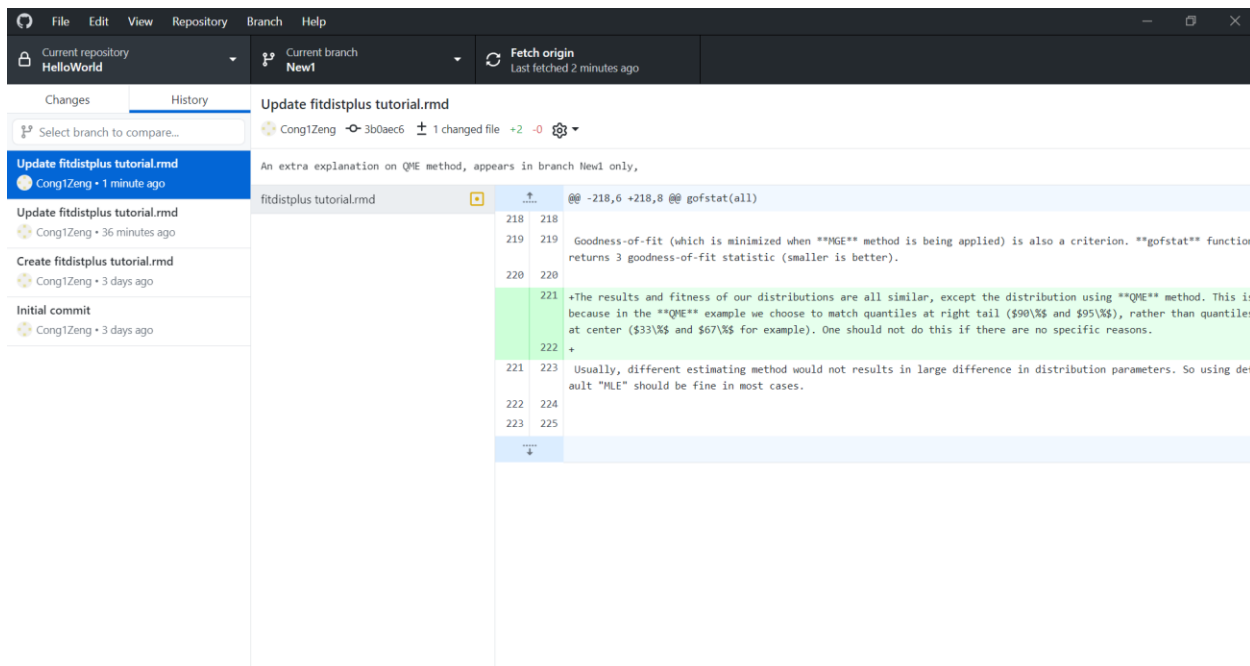
6: Branch

“Branch” enables people make changes to the project independently.

To create a new branch, click on “New branch”.



Now I created a branch named “New1” and made some extra changes. One can find it in the history of “New1” but not the history of “main”.



One can compare between the branches to get a clear view of what changes being made to each branches. “Create a merge commit” will merge the selected change in a branch to the current branch you are working on.

File Edit View Repository Branch Help

Current repository
HelloWorld

Current branch
main

Fetch origin
Last fetched 4 minutes ago

Changes
History

New1

Behind (1) Ahead (0)

Update fitdistplus tutorial.rmd
Cong1Zeng • 4 minutes ago

Update fitdistplus tutorial.rmd

Cong1Zeng 3b0aec6 1 changed file +2 -0

An extra explanation on QME method, appears in branch New1 only,

fitdistplus tutorial.rmd

```

@@ -218,6 +218,8 @@ gofstat(all)
218 218
219 219 Goodness-of-fit (which is minimized when **QME** method is being applied) is also a criterion. **gofstat** function
220 220 returns 3 goodness-of-fit statistic (smaller is better).
221 221 *The results and fitness of our distributions are all similar, except the distribution using **QME** method. This is
222 222 because in the **QME** example we choose to match quantiles at right tail ($90\%$ and $95\%$), rather than quantiles
223 223 at center ($33\%$ and $67\%$ for example). One should not do this if there are no specific reasons.
224 224
225 225 Usually, different estimating method would not results in large difference in distribution parameters. So using def
226 226 ault "MLE" should be fine in most cases.

```

This will merge 1 commit from New1 into main

Create a merge commit

If select all differences between 2 branches and merge, then the branch is “behind” will be updated to he “ahead” version.