As far as we have gotten, we already know how to do a `git pull` to get what the LII have pushed today to his repository, and how `to add, commit and push`***** your own modifications to your repository.

*****
```
> git add -A
> git commit -m "what are you doing?"
> git push origin master
```

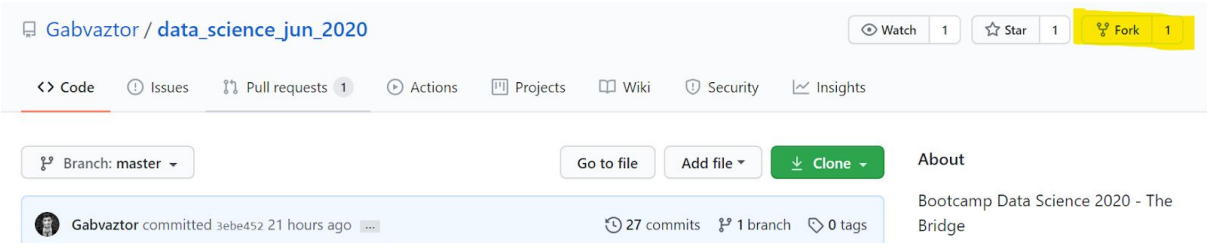
Links

- https://rogerdudler.github.io/git-guide/index.es.html

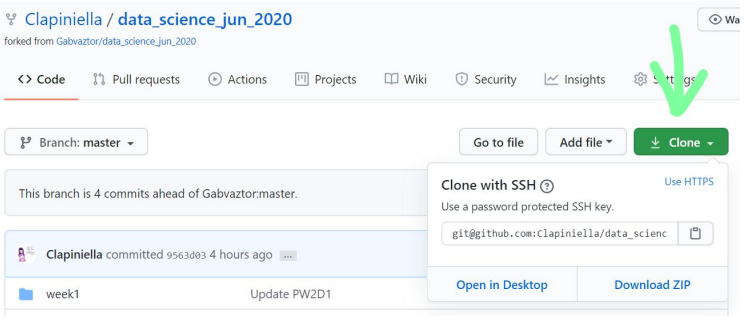
Time has come for you to do both on the **same repository.**

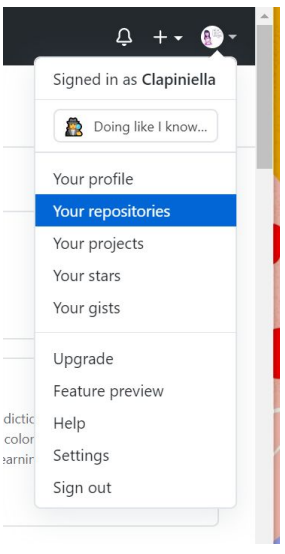
How to make possible for you to do modifications on data-science-jun-2020?

1. go to the repository while being on **github**



2. press **Fork**
3. go to **your repositories** and enter the one with the same name as the one you have just forked
4. copy the url you'll find where the green arrow is
5. open a terminal and type the following → **`git clone url`**





¡Now you have a copy of data-science-jun-2020 on your computer!

How to make a PULL REQUEST?

Must do everyday as you have been doing with `git pull`

`git status`

```
$ git status
On branch master
Your branch is up to date with 'origin/master'.

nothing to commit, working tree clean
```

if not:

`git add/git commit/git push*`

* **`> git add -A`**
`> git commit -m "what are you doing?"`
`> git push origin master`

`git checkout master`

double check you are on branch master

how? **`git status`**

`git fetch upstream`

`git merge upstream/master`

something like this could appear...

How to handle this?

1. **press i** → to insert the description of why are you merging
2. Write the description
3. **Esc** → once you have finished writing
4. Type `:qa!` (or `:wq`)
5. Press **ENTER**

`git push` → to sync your local repository with github

```
clara@DESKTOP-BKRKAJS MINGW64 ~/Desktop/The_Bridge/TA
$ cd data_science_jun_2020/

clara@DESKTOP-BKRKAJS MINGW64 ~/Desktop/The_Bridge/TA/data_science_jun_2020 (spl
it_precurso)
$ git checkout master
Switched to branch 'master'
Your branch is ahead of 'origin/master' by 2 commits.
  (use "git push" to publish your local commits)

clara@DESKTOP-BKRKAJS MINGW64 ~/Desktop/The_Bridge/TA/data_science_jun_2020 (mas
ter)
$ git status
On branch master
Your branch is ahead of 'origin/master' by 2 commits.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean

clara@DESKTOP-BKRKAJS MINGW64 ~/Desktop/The_Bridge/TA/data_science_jun_2020 (mas
ter)
$ git fetch upstream
remote: Enumerating objects: 14, done.
remote: Counting objects: 100% (14/14), done.
remote: Compressing objects: 100% (7/7), done.
remote: Total 10 (delta 3), reused 10 (delta 3), pack-reused 0
Unpacking objects: 100% (10/10), 3.40 MiB | 3.29 MiB/s, done.
From https://github.com/Gabvaztor/data_science_jun_2020
   fe215d8..f051042  master     -> upstream/master
```

```
clara@DESKTOP-BKRKAJS MINGW64 ~/Desktop/The_Bridge/TA/data_science_jun_2020 (mas
ter)
$ git merge upstream/master
Merge made by the 'recursive' strategy.
 week3/day3/class.md                                |    4 +
 week3/day3/exercises/Exercises_1-8_v7.ipynb        | 3845 ++++++++++++++++++++
 ...ive_Statistics_using_Pandas_and_Seaborn (2).pdf |  Bin 0 -> 4115707 bytes
 week3/day3/theory/Estadistica.ipynb                | 1848 ++++++++++
 week3/day3/theory/Ficheros.ipynb                   |  118 +
 week3/day3/theory/content.ipynb                    |   82 +
 6 files changed, 5897 insertions(+)
 create mode 100644 week3/day3/exercises/Exercises_1-8_v7.ipynb
 create mode 100644 week3/day3/theory/A_Quick_Guide_on_Descriptive_Statistics_using_Pandas_and_Seaborn (2).pdf
 create mode 100644 week3/day3/theory/Estadistica.ipynb
 create mode 100644 week3/day3/theory/Ficheros.ipynb
 create mode 100644 week3/day3/theory/content.ipynb
```

Create a branch, code within it

A branch is essentially a unique set of code changes with a unique name. Each repository can have one or more branches. The main branch — the one where all changes eventually get merged back into, and is called master. This is the official working version of your project, and the one you see when you visit the project repository at github.com/yourname/projectname. Read more

To create a branch

`git branch branch_name`

To checkout to that branch

`git checkout branch_name`

To create a branch and checkout out to that branch **BEST OPTION**

`→ git checkout -b branch_name`

Every time you'd like to make any modification to **data_science_jun_2020** and before doing any change, make sure you have created a new branch and that you actually are on that branch you have created. To check that, use **`git status`** command.

It's important to keep in mind that any changes you may make while you are on **nueva_rama**, will stay in that branch, so you won't see those changes in any other branch, as the master one.

"Lo que pasa en la rama, se queda en la rama"

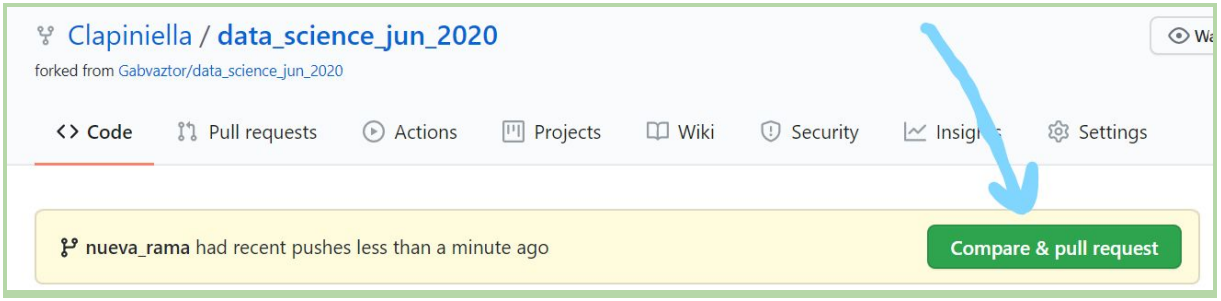
```
clara@TA-Data-Science MINGW64 ~/Desktop/The_Bridge/TA/data_science_jun_2020 (nue
va_rama)
$ git status
On branch nueva_rama
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        week4/day4/theory/cambios.md
        week4/day4/theory/otros_cambios.ipynb

nothing added to commit but untracked files present (use "git add" to track)
```

Make sure to do **`git add/git commit/git push*`**, this time and while being on a branch you'll do **`git push origin EL_NOMBRE_DE_LA_RAMA`**

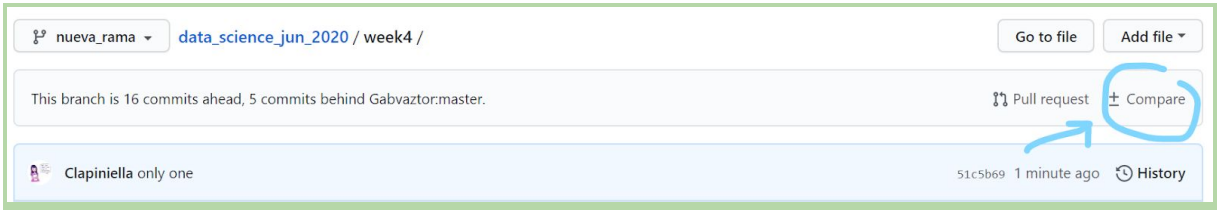
```
*       > git add -A
        > git commit -m "what are you doing?"
        > git push origin nueva_rama
```

Open a PULL REQUEST on Github

Once you've done the **`git push origin EL_NOMBRE_DE_LA_RAMA`**, go to Github, and to your repository… which is the forked repository data-science-jun-2020.
From there you should be able to make a **pull request**, press **Compare & pull request**



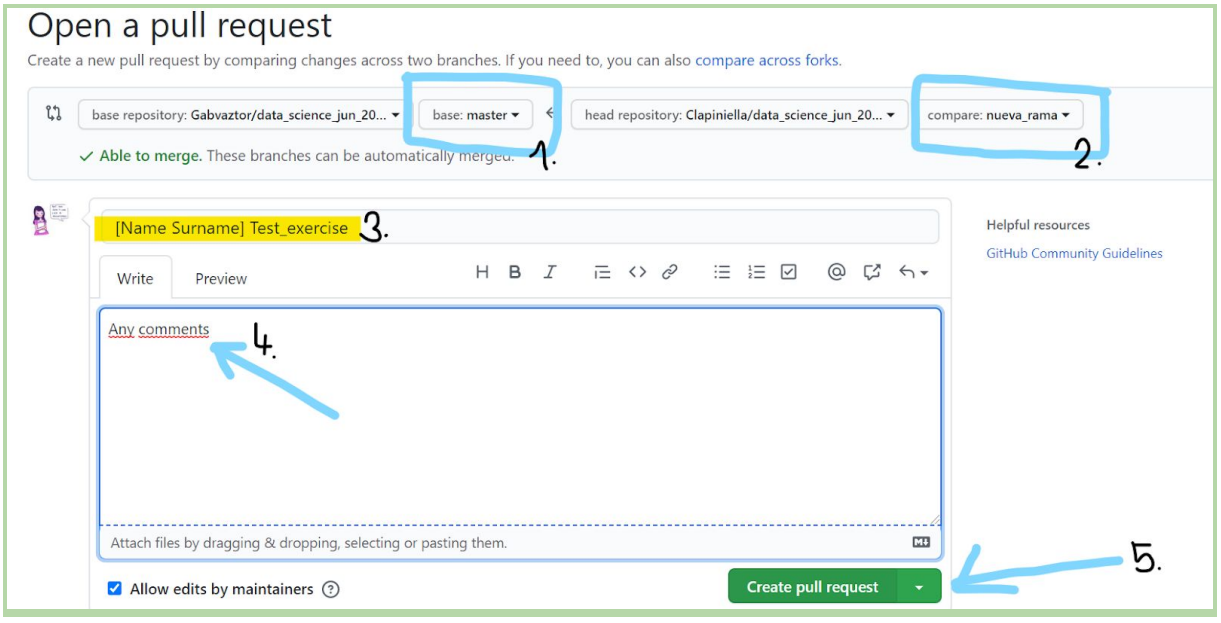
If you cannot see what it is showed before, you can do it manually clicking on **Compare**



You'll access to the following page, where you should follow the next steps:

1. Check the base repository your Lead Instructor repository, on the branch master

2. Check the head repository is your repository, on the branch which contains your exercise or changes which you've just committed
3. Write your name and the exercise name as: **[Name Surname] Test_exercise**
4. Write any comments
5. Press **Create pull request**



Now you have created a pull request.

NOTE: you will submit every file you have committed to the master branch and the ones committed to the branch from where you've created the pull request.