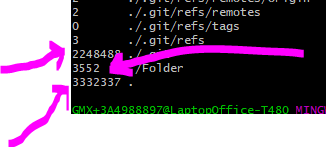
**Steps to free up space (get rid of .git files) in a repository**

1. Note doing this will also involve steps (**How to delete an existing Repository on github)** and (**How to create a new Repository on github and clone it on Computer)** later in the process. Those steps are listed in detail in this document later.
2. Before we begin – you can use “**du -ks**” (if you want a short version) or “**du -k**” in the directory (repository) that you want to reduce the size of on github. For “historical” this is what it shows today:



That means that “historical” is using 3.3 GB of space, and out of which ./.git is using around 2.2 GB and so the actual files may be using around 1.1 GB of data. That means if we clean this repository up, then we should be able to free up considerable space on github. (Of course, you can get the space information from Windows Explorer as well).

1. I will use the repository “historical” as an example here, but it can be done for any other repository (“Charts”, “Latest\_Charts” and “python”, though doing some of them would be more complicated as they have sub-directories under them, but the general flow should be similar). The sub-steps here would be to:
   1. For Charts (Read in entirety before doing anything) –
      1. **Copy** Charts directory to Charts\_org directory (You will need to create Charts\_org and then copy over). This can be done more easily from windows explorer, though you can do it from command line as well. This is only needed in case something goes wrong. In that case we have all the original files still available in Charts\_org
      2. **Dry run** the script by commenting out the line   
         **shutil.move(source\_file\_with\_path, dest\_file\_with\_path)**

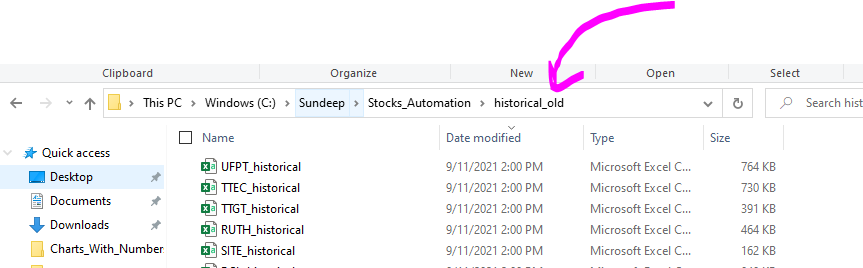
in the script **SC\_MoveCharts\_to\_OlderCharts.py**

* + 1. Verify that the dry run looks okay by running  
       **grep -i "Found Older Chart" SC\_MoveCharts\_to\_OlderCharts\_debug.txt**

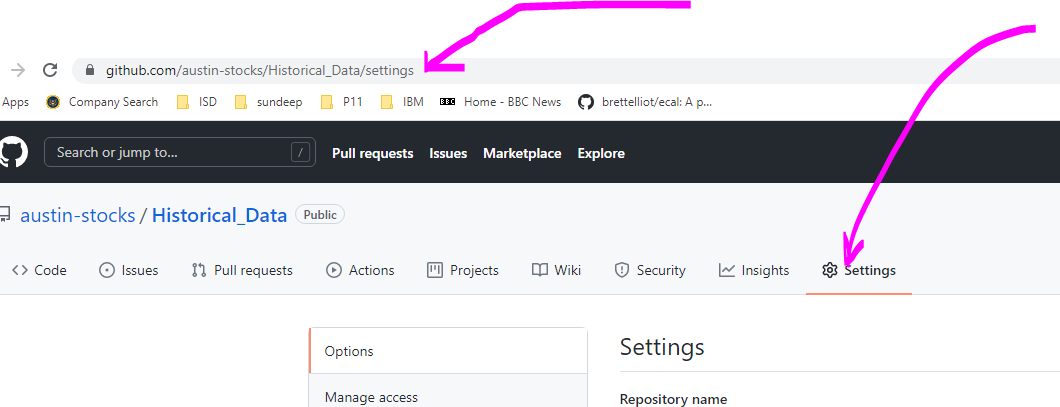
This will give the files that the script would have moved if it had not been a dry run. Make sure that the files that the script is planning to move look okay (older than one year). Make sure that the files are included from all the sub-directories (Linear, Long\_Linear and Log etc). This can also be done by looking at debug file in the Log directory.

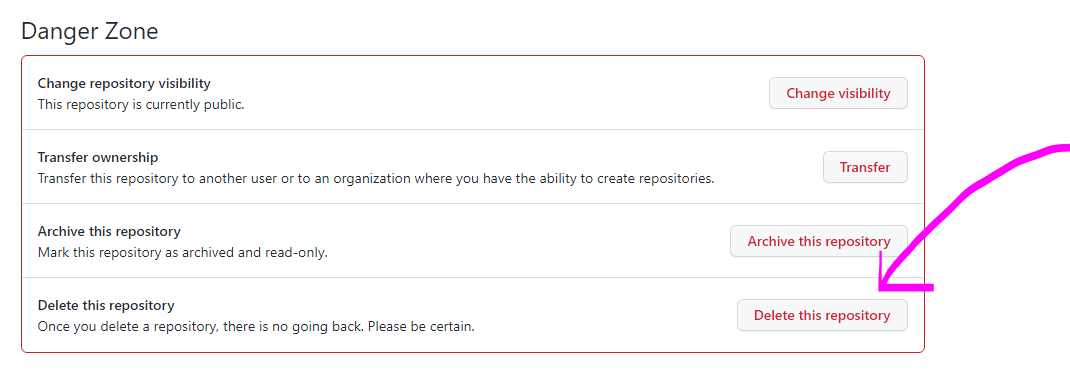
* + 1. Once you are satisfied, run the script **SC\_MoveCharts\_to\_OlderCharts.py**
    2. This will **move** the Chart files that are older than one year to Older\_Charts directory from Charts directory.
    3. At this point you will have
       1. Charts directory - it should ONLY have the charts that are newer than the last year as the older charts have been moved to Older\_Charts directory
       2. Older\_Charts directory that should have all the older charts (from the previous runs) and the **newly moved** charts – from Charts directory - that are older than one year
       3. Charts\_org directory that should have maybe 2 years worth of charts (whatever the original Charts directory had when we started)
    4. Now **move** Charts directory to Charts\_old. This will now become the directory from which the one year old charts will be **moved** to the NEW and EMPTY freshly created git (Charts) repository on github (and then cloned) on local windows machine.
    5. Once everything goes well – then Charts\_org and Charts\_old will be deleted, and we will be left with Older\_Charts and Charts directory
  1. For Non- Charts –
     1. Rename (move) that directory on the local windows computer(local clone). Say move historical to historical\_old or Latest\_Charts to Latest\_Charts\_old.
     2. This \_old directory will be used later to get the files that you want to populate the NEW and EMPLY created repository (with the same name – historical or Latest\_Charts, in our case). So, make sure that you do not do anything with \_old unless everything is done and sanity checked. Otherwise, you would have to go to the backup from the cloud to get the files and that can be messy and time consuming…
  2. Delete that repository on github (**How to delete an existing Repository on github)**
  3. Create (NEW) repository on github with the same name and clone that newly created (empty!!!) repository on local computer (**How to create a new Repository on github and clone it on Computer)**
  4. Copy (you can move, but it can be dangerous) the files that you want the git to be populated with into the freshly created (NEW) repository clone directory, from the Charts\_old or historical\_old or Latest\_Charts\_old directory (or whatever you re-named it to in step a. or b.)
  5. Commit and push into git
  6. **MAKE SURE THE git on github NOW HAS THE FILES THAT YOU JUST MOVED**
  7. If you now feel safe, then you can remove the \_org/\_old directory that you created on local computer, to free up local space

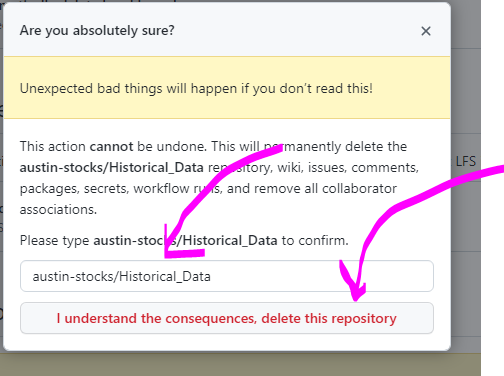
1. So – first step – rename the “historical” directory to “historical\_old”. Again – this can be done from either Windows Explorer or command line on Git bash – your choice. I will do it from Git bash this time around.   
   

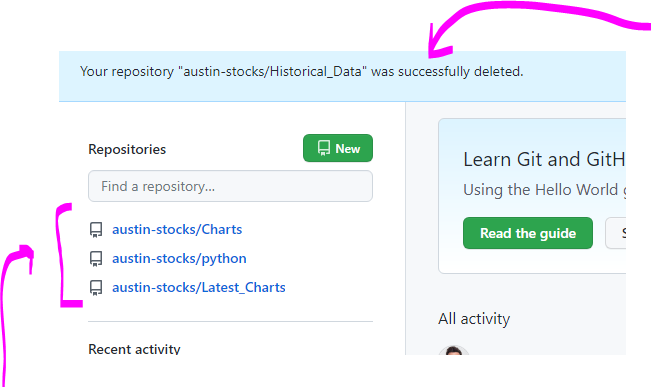
Make sure that historical\_old has all the historical file before moving to the next step – yes, I am being paranoid, but better safe than sorry  


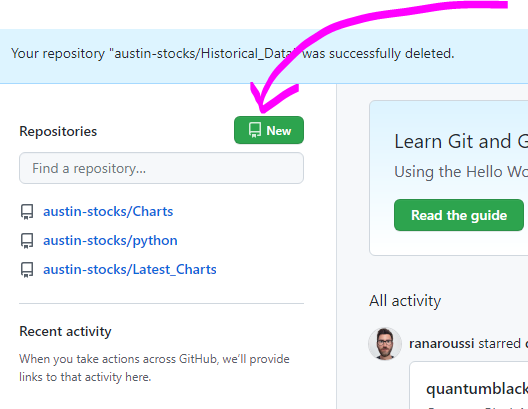
1. Now, let’s delete the repository on github (steps are listed in the section “**How to remove a repository on github**”). You should go a read that section – then the steps that I am listing below will make more sense.

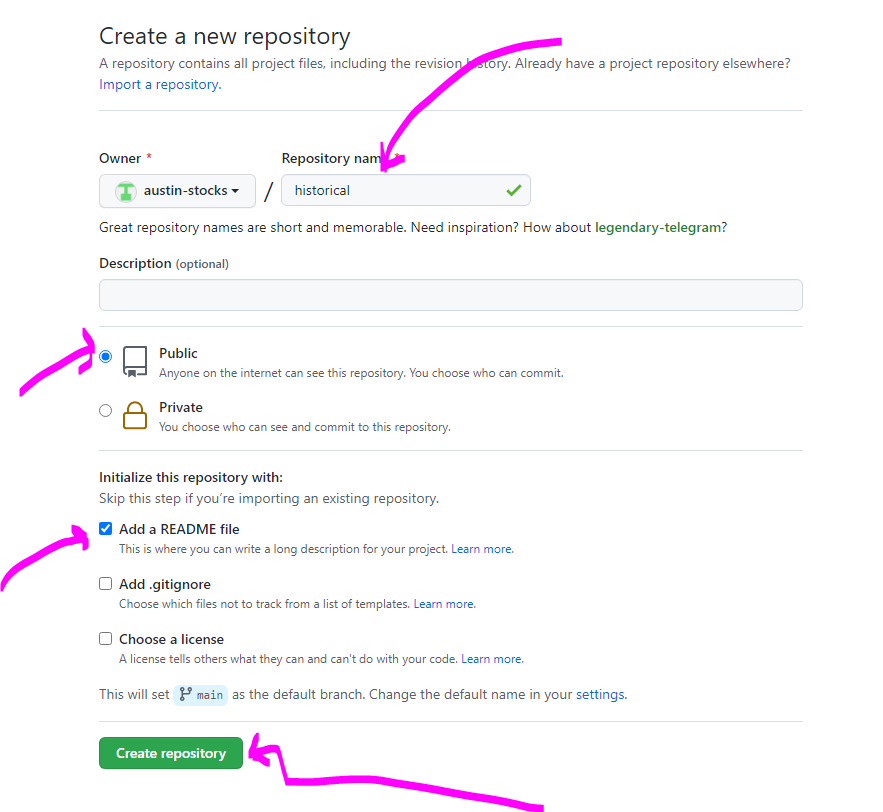
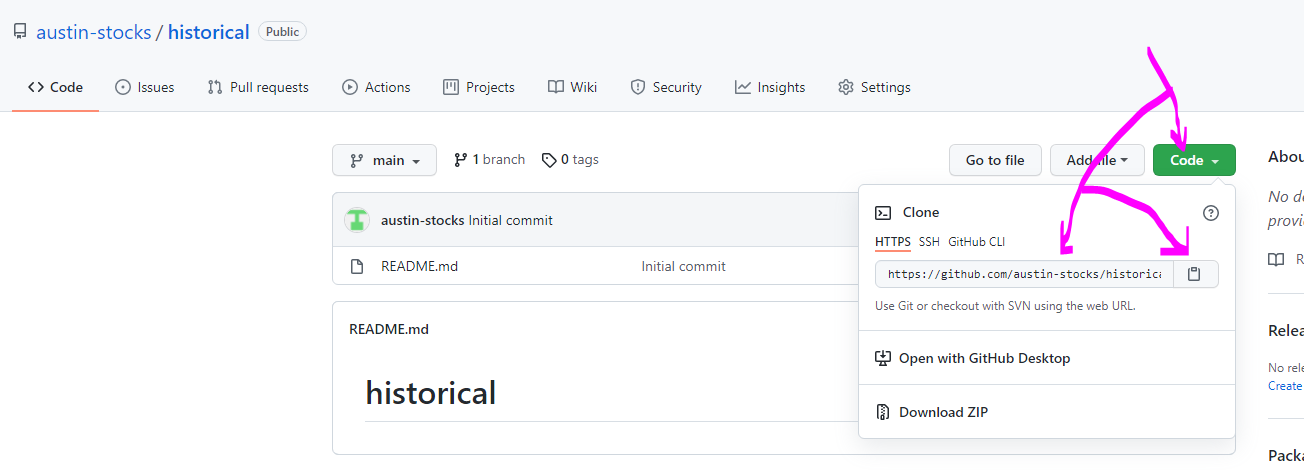


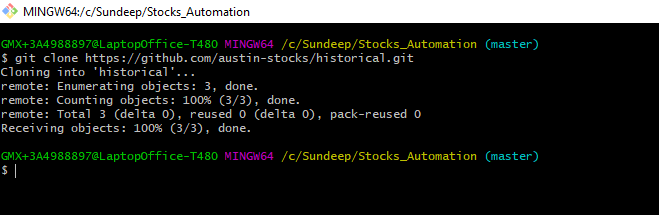


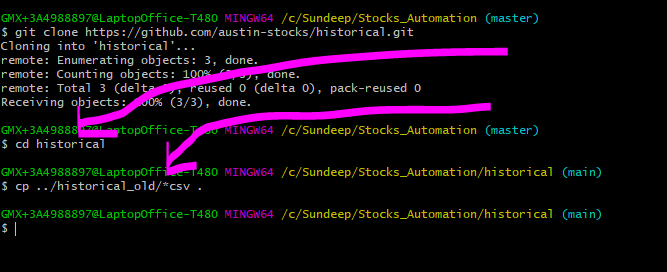


1. Now that repository is gone from austin-stocks  
     
   
2. Now create a new / fresh repository with the same name (the steps are listed in the section **“How to create a new Repository on github and clone it on Computer**“ below). Probably better to read those steps first, so that the subsequent steps here make more sense.

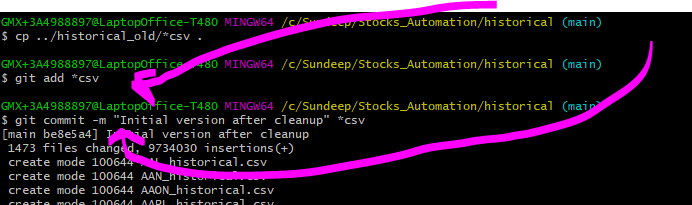


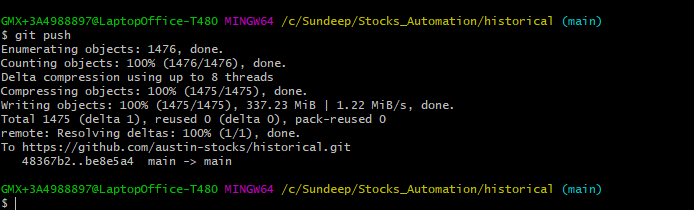
1. Creating a new repository  
     
     
   
2. Repository created – now cloning that on the local computer   
     
   

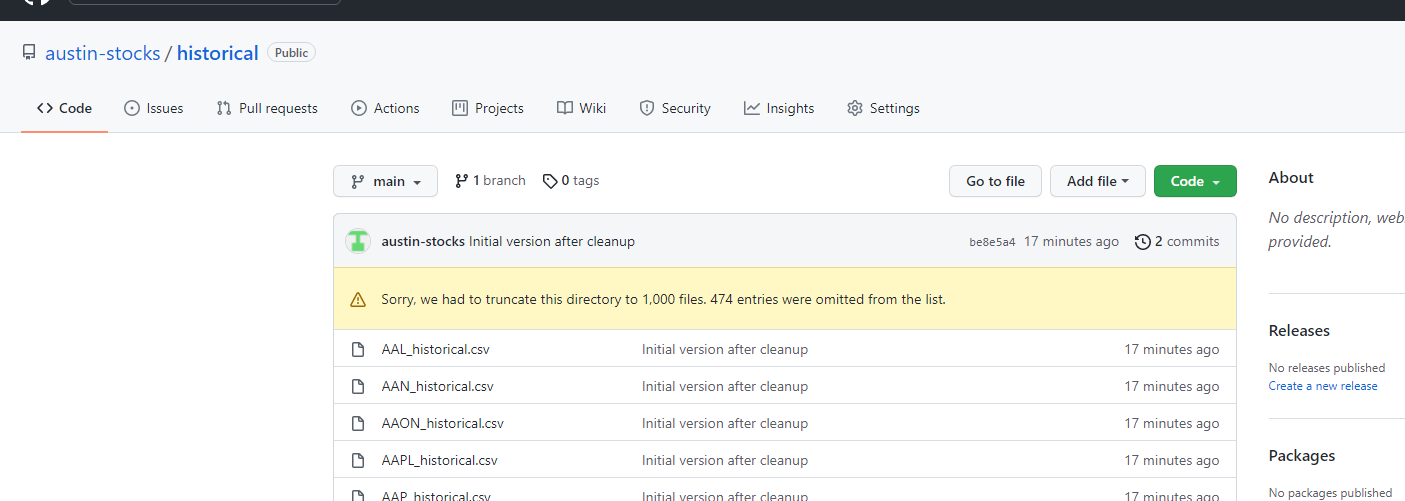


1. Now that the repository is cloned – copy/move the files from \_old to the new directory and git adding and git committing and git pushing them and viloa we are done 😊  
     
   

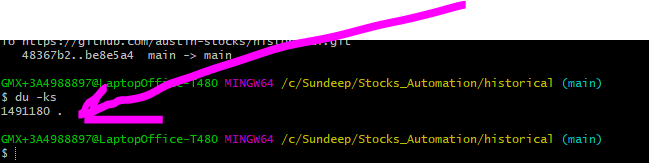
Note that I used cp here – and that can be done with windows explorer too…I could have used mv instead, but just being paranoid here to make sure that everything is still available on local computer, in case something messes up.

1. Now git add and git commit and git push  
     
   





1. At this point, I would run a python script to make sure that things are working fine (in this case to make chart that reads historical, or creates a historical…I verified it and so now – I can delete historical\_old directory but now before I find out how much space did I cleanup on github

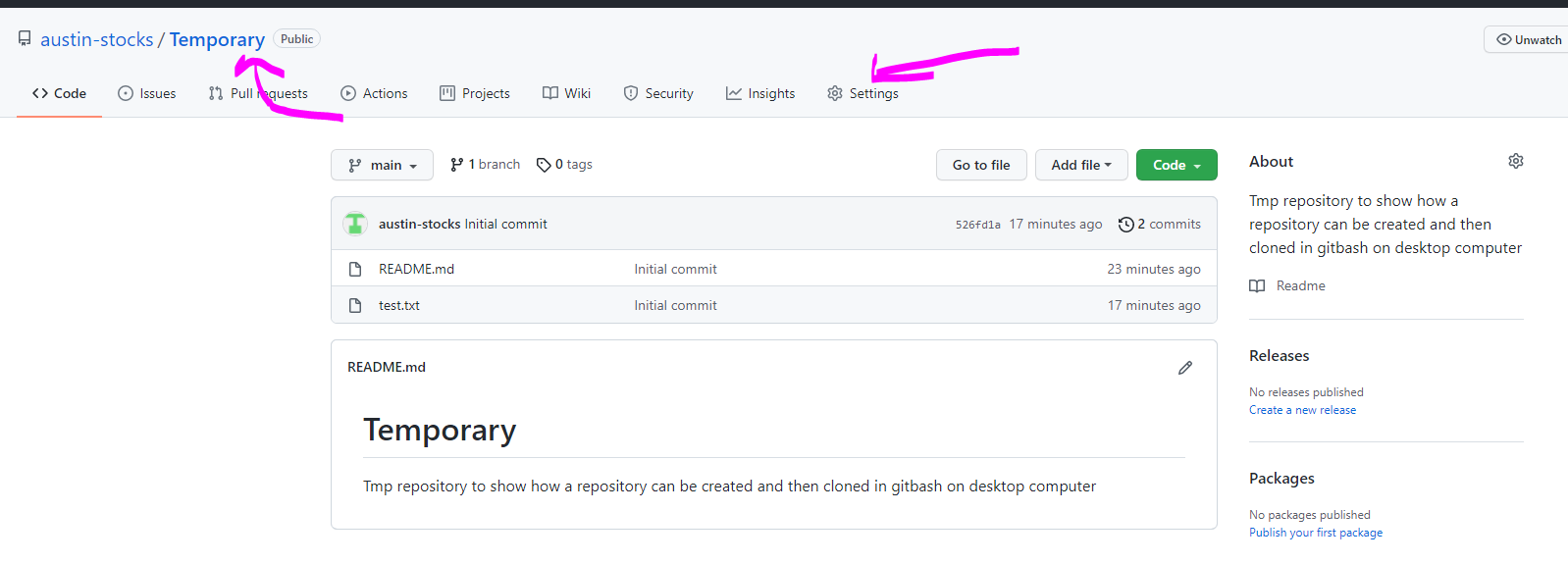


So – it seems we cleaned up around 1.5 GB of space on github….not bad.

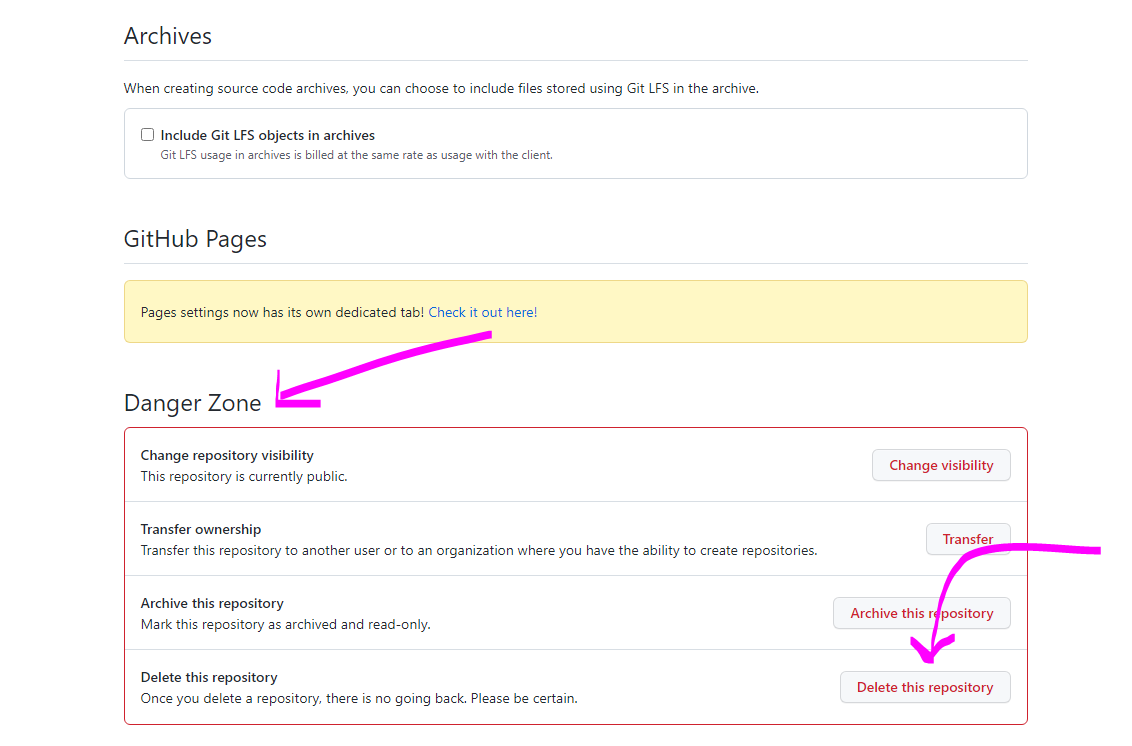
1. This can be used as an example for Latest\_Charts
   1. For Latest\_Charts here is what I did after I moved Latest\_Charts to Latest\_Charts\_old and deleted, recreated, cloned the Latest\_Charts directory on local computer and then copied the directories Linear, Long\_Linear and Log and ./git\_commit.sh from Latest\_Charts\_Old to Latest\_Charts
      1. git add Linear/Charts\_With\_Numbers/\*jpg -v (verbose get you more messages)
      2. git add Linear/Charts\_Without\_Numbers/\*jpg -v
      3. and so on….
      4. git add git\_commit.sh
      5. git commit -m “More Updates” Linear/Charts\_With\_Numbers/\*jpg -v
      6. git commit -m “More Updates” Linear/Charts\_Without\_Numbers/\*jpg -v
      7. and so on…
      8. git commit -m “More Updates” git\_commit.sh -v
      9. Finally – git push -v
      10. AND THAT IS IT

**How to delete an existing Repository on github**

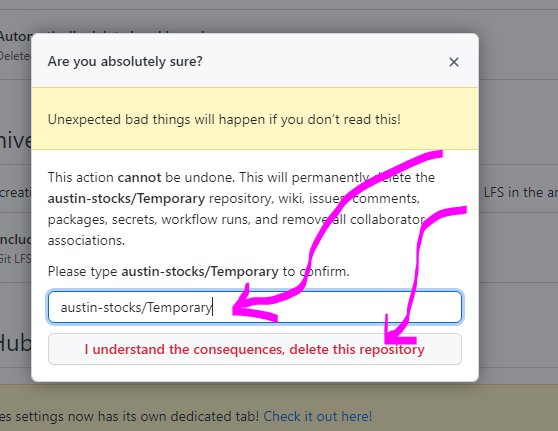
1. We are now going to delete the repository that we just created. Note : You can delete the directory on your computer and that will get rid of the the local version of the clone and that Iis all that you need to do on the local computer.
2. On the repository page, click on “**Settings”**

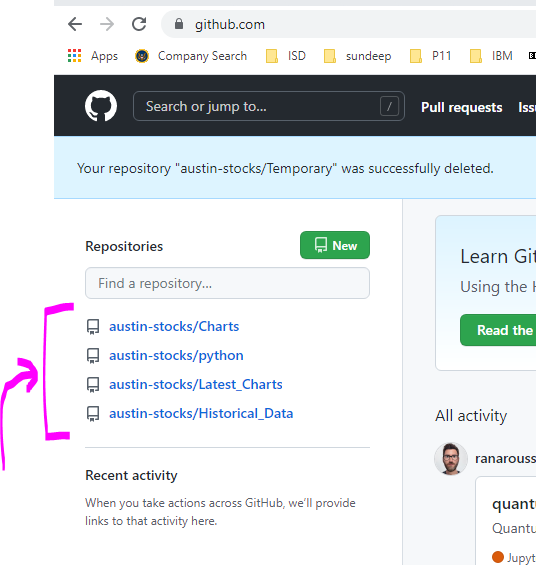


1. This will take you the setting page 😊, go all the way down in the “Danger Zone”. There we have the option to “**Delete this repository**”



1. Since this is a pretty dangerous step (un-recoverable), so github will confirm once more



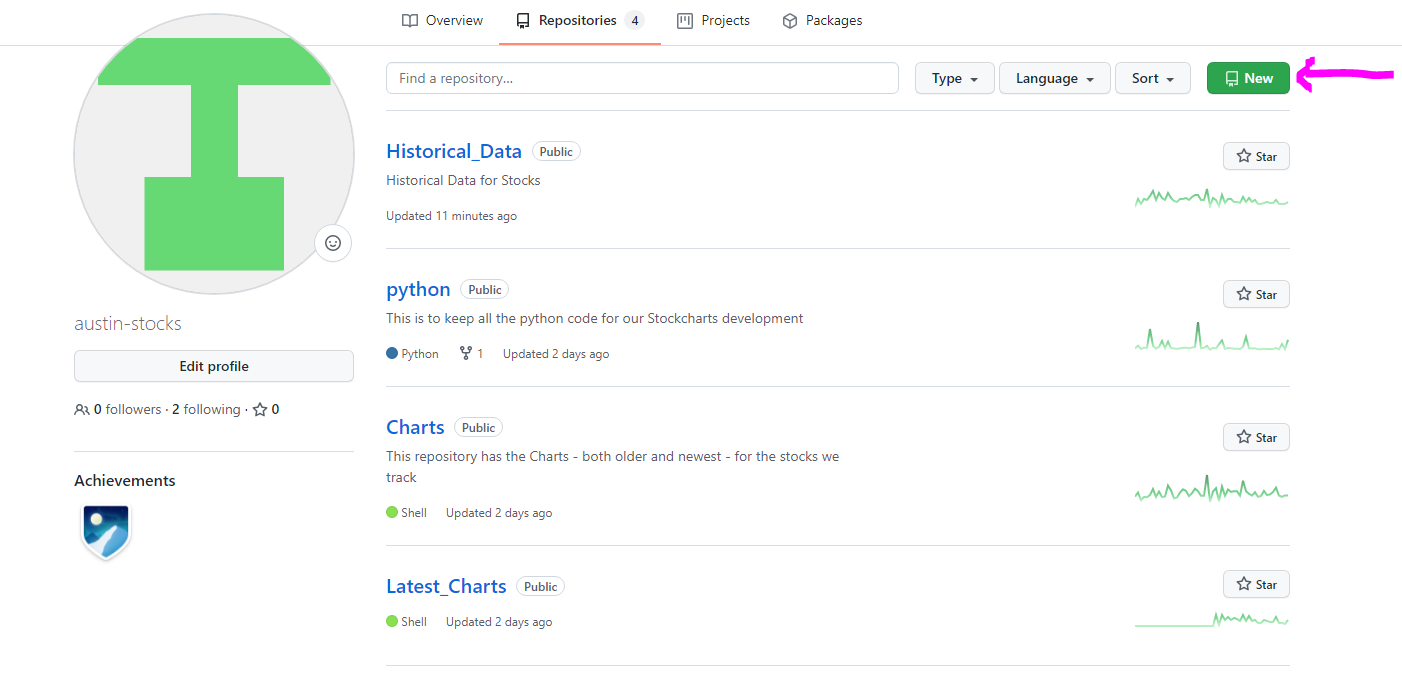
1. So, if you are really sure, then click “**I understand the consequences, delete this repository**” button and the repository will get deleted. Now we can go the the main page of austin-stocks and see that “Temporary” repository does not exist  
     
   
2. You can now delete “Temporary” directory on your local computer and that should be the end of any remnants of the that repository anywhere. Git bash not able to clone the repository shows that it is gone  
     
   

**How to create a new Repository on github and clone it on Computer**

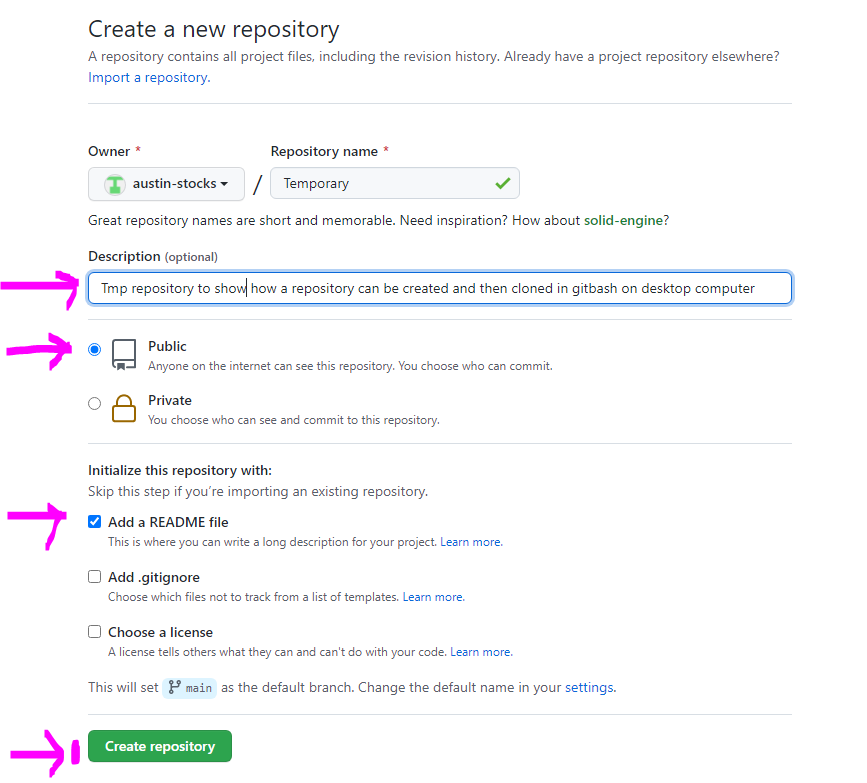
1. To create a new repository on github, go the page where it lists the repositories for austin-stocks

<https://github.com/austin-stocks?tab=repositories>

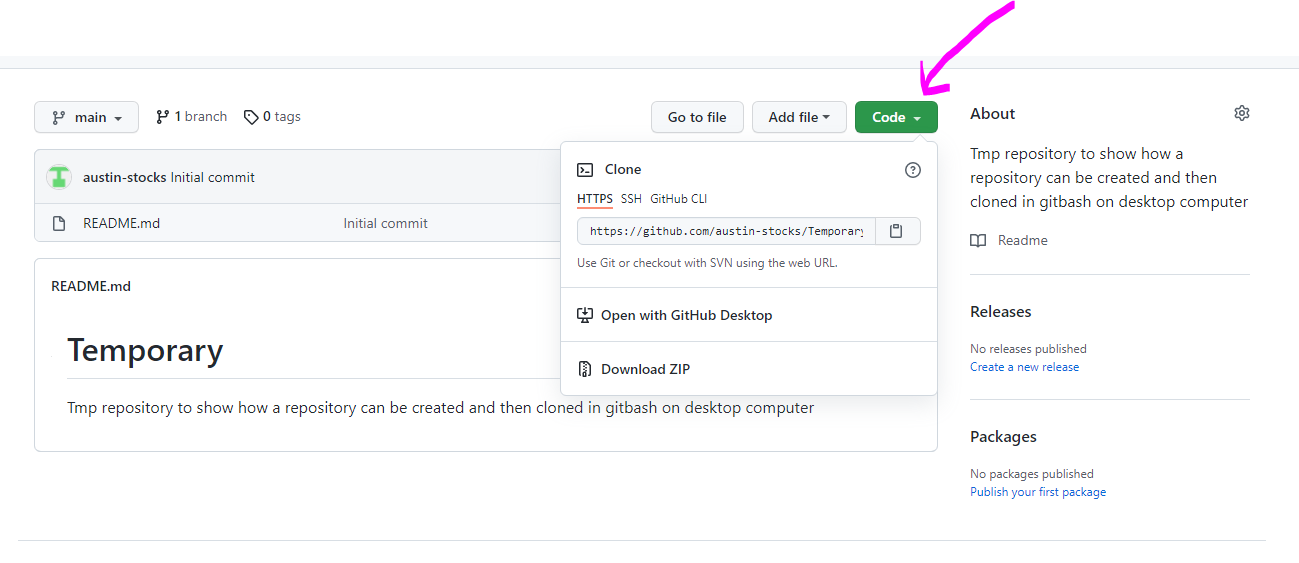
Then click on “**New**“



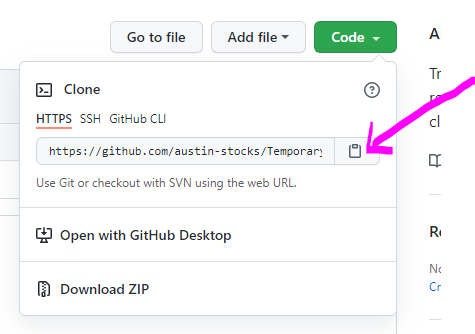
1. This takes to the window, where a new repository can be created. Fill in the name of the repository (This will be the name of the directory that will be the default name of the cloned directory on computer, so choose that with care) and Click on “Add a README file”, though it is probably not needed. If you want to fill something in the “Description”, then you can do so. Then click on “**Create Repository**”.

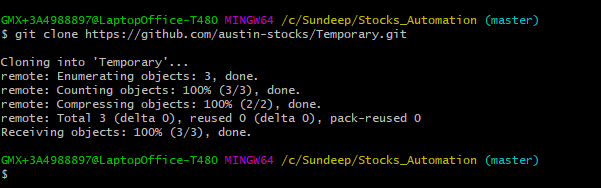


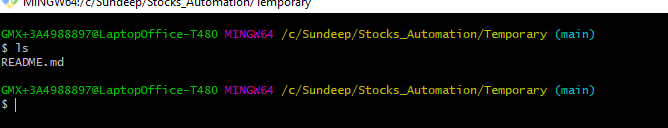
1. Once you click the repository button, the repository is created and we come to this window. Here click on the green “**Code**” button, it will give a HTTPS link to clone the repository. I think that the link can also be found by click the “gear wheel” on the right side of the “About” the repository. I did not try that, but that should work too.



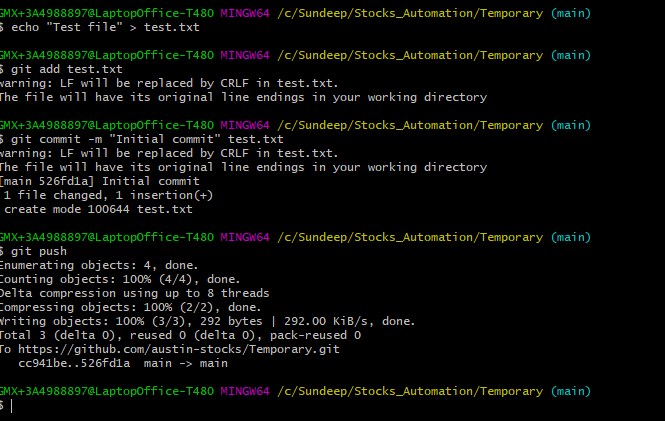
1. Now we can use that HTTPS link to clone the repository using Git bash on windows (need to make sure that we are logged in as austin-stocks (how??? – maybe later try to figure it out)) in Git bash on that computer. The link can be copied by clicking







1. Now you can add files to the repository and they will appear on github



1. The file test.txt now appears on github…we are in business **voila 😊**

