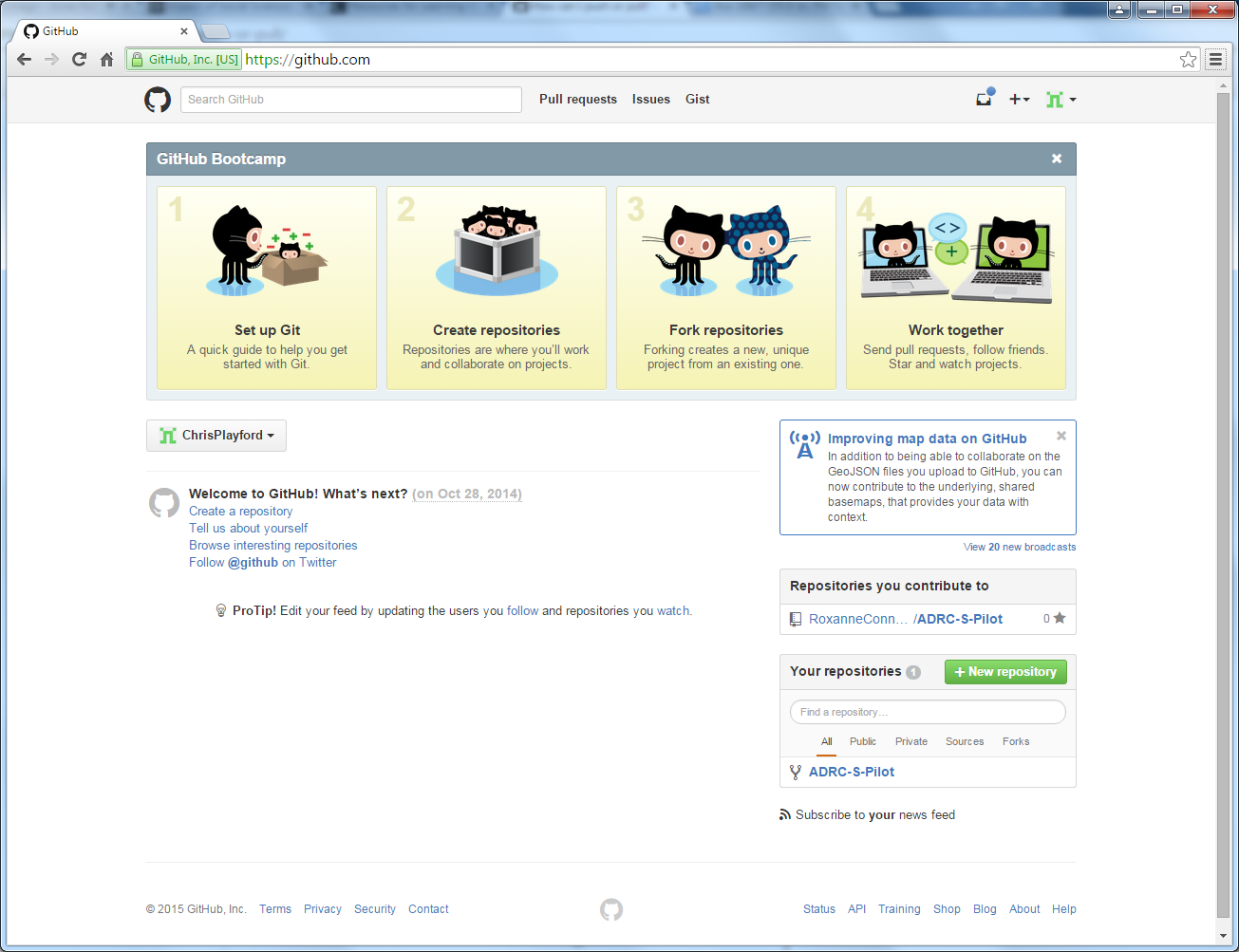
31st July 2015

# ADRC-S: GitHub Training

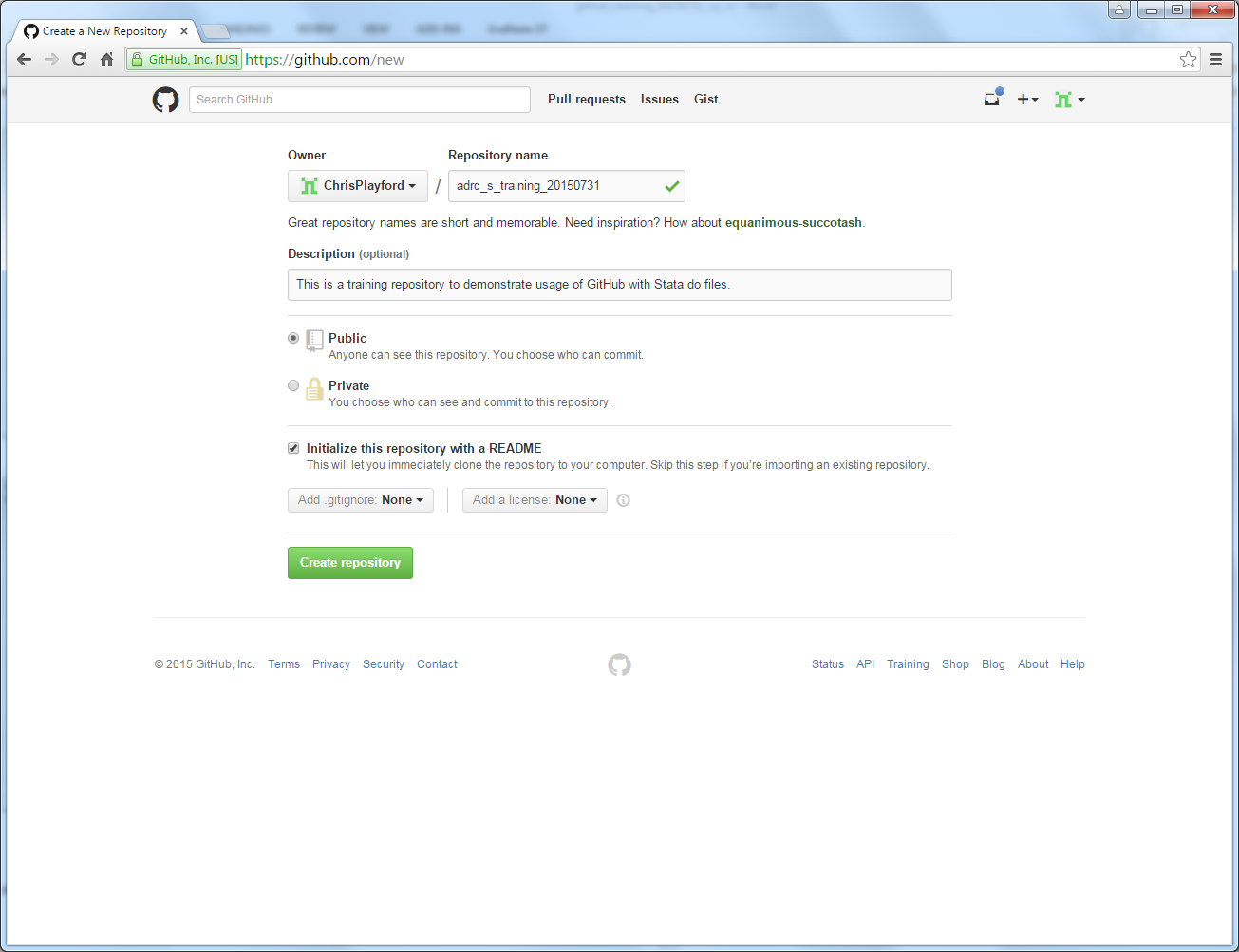
See <https://github.com/ChrisPlayford/adrc_s_training_20150804> to view the pages referred to.

NB: I initially created a repository called “adrc\_s\_training\_20150731”, then deleted it and created a new one called “adrc\_s\_training\_20150804”.

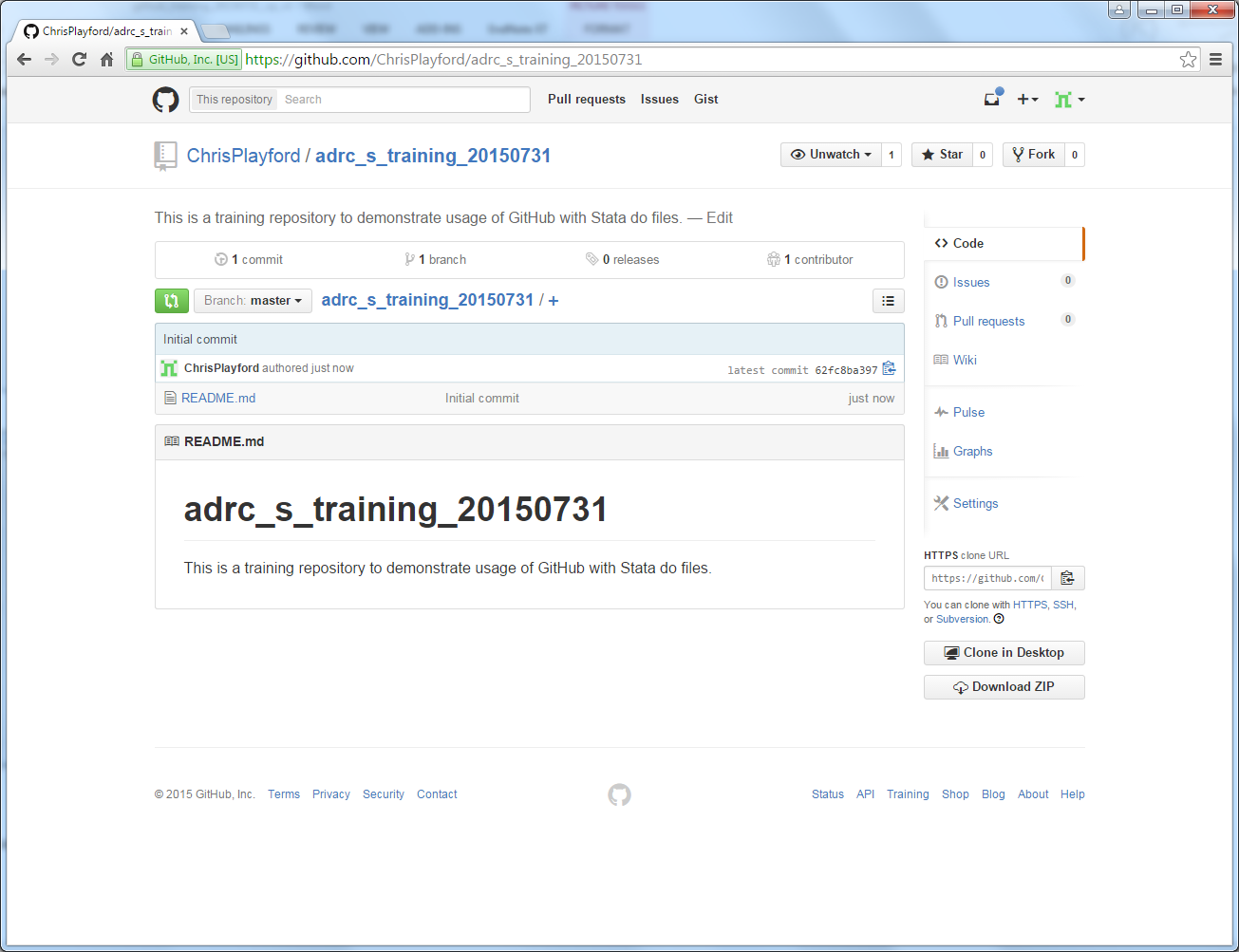
1. **Create account**



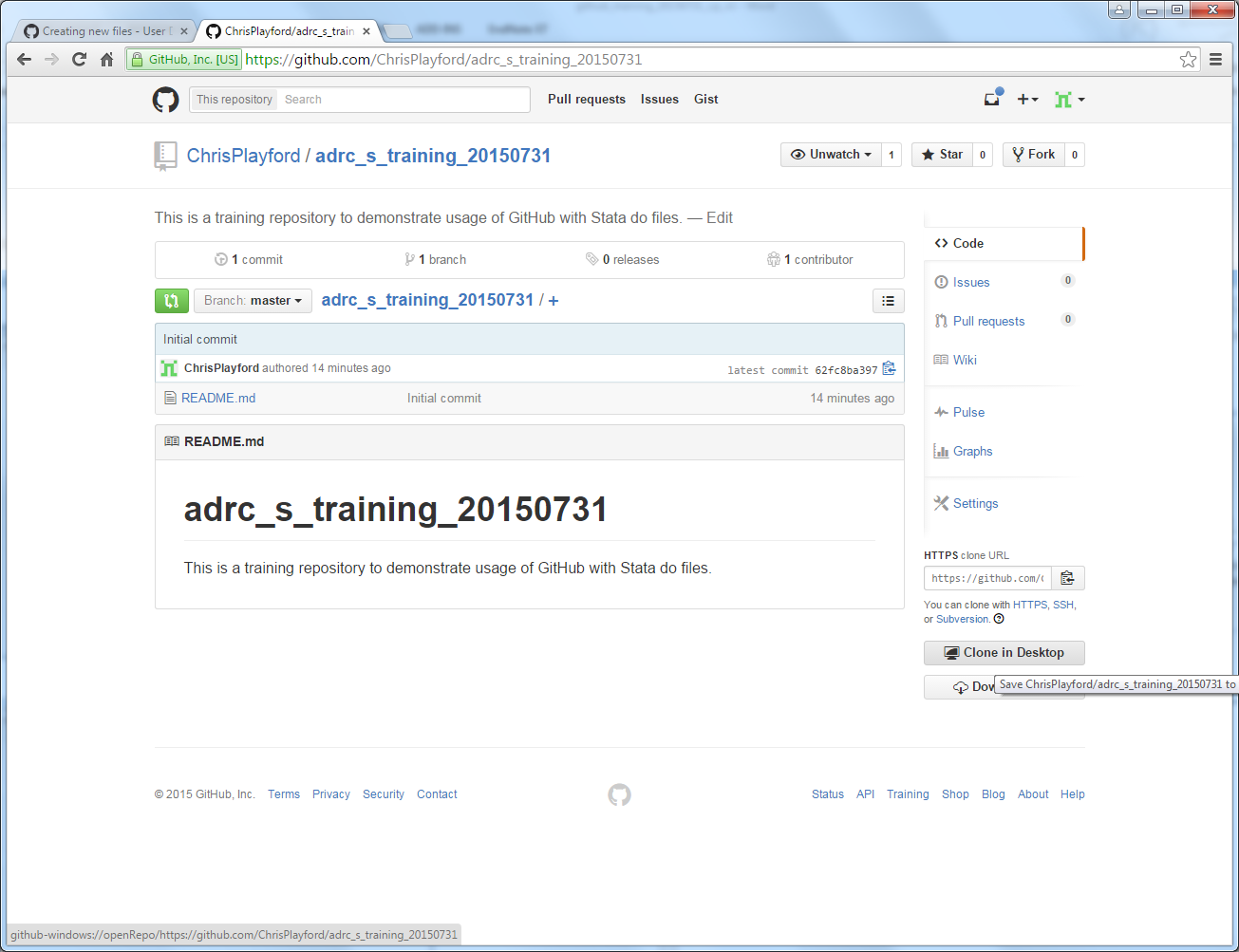
1. **Create repository – “adrc\_s\_training\_20150731”**



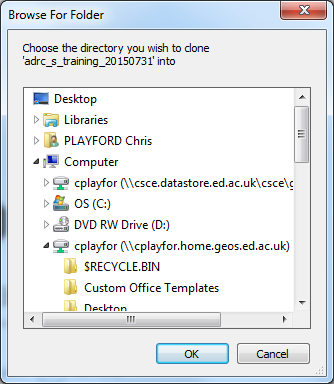
1. **New repository – “adrc\_s\_training\_20150731”**



1. **Clone in desktop – “adrc\_s\_training\_20150731”**



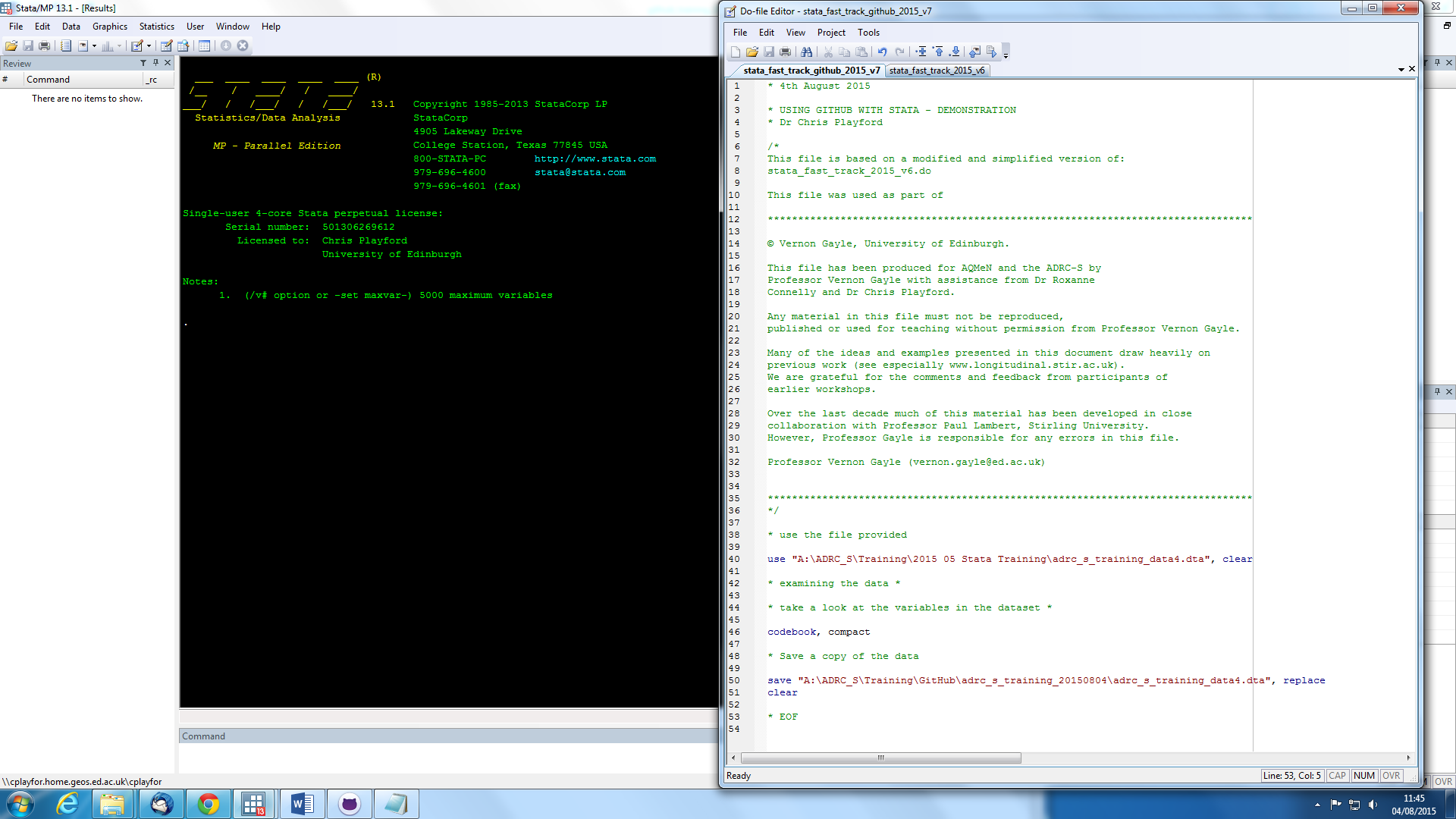
1. **Choose folder you which to clone into – “adrc\_s\_training\_20150731”**



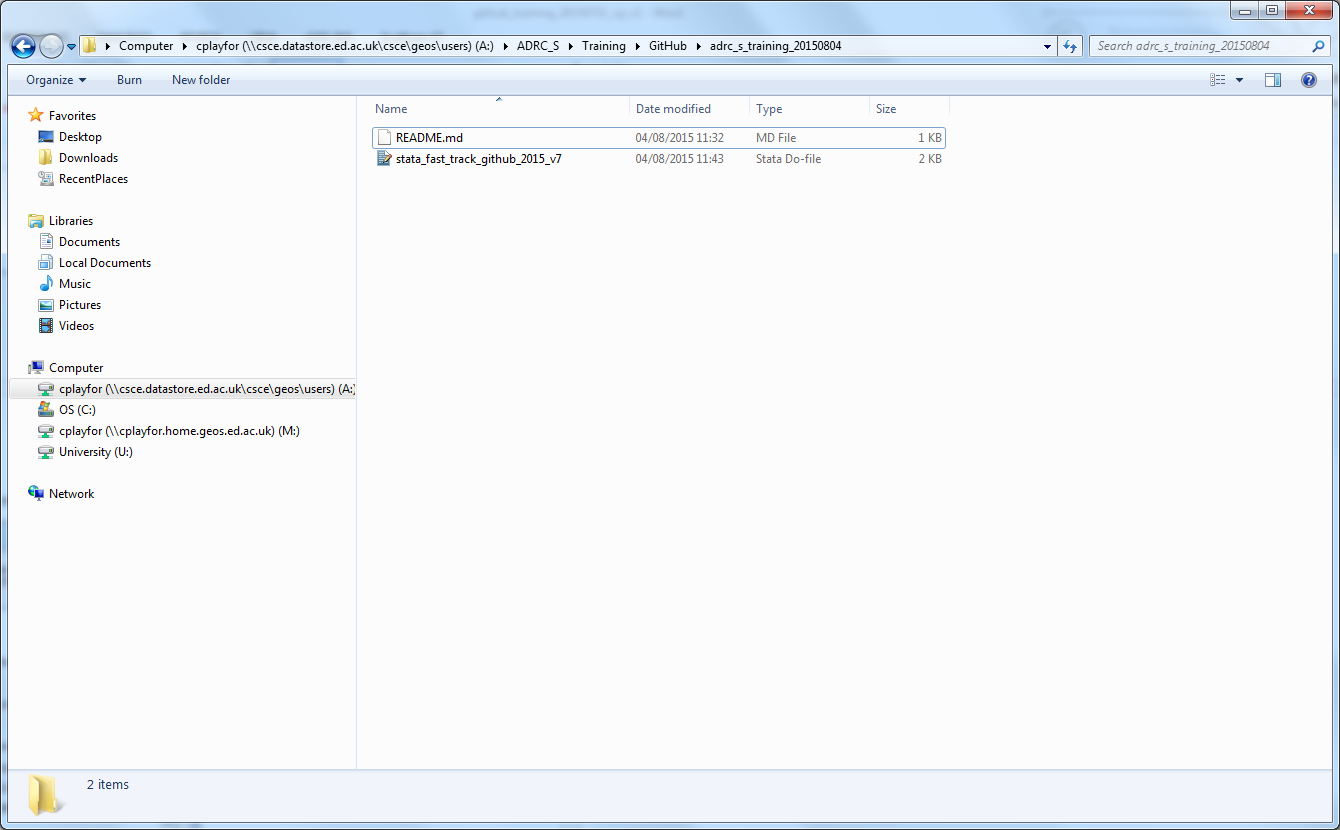
* I have created a new repository called “adrc\_s\_training\_20150804”
* The principles above are consistent

1. **Create new file in Stata – saved to relevant folder on local machine**

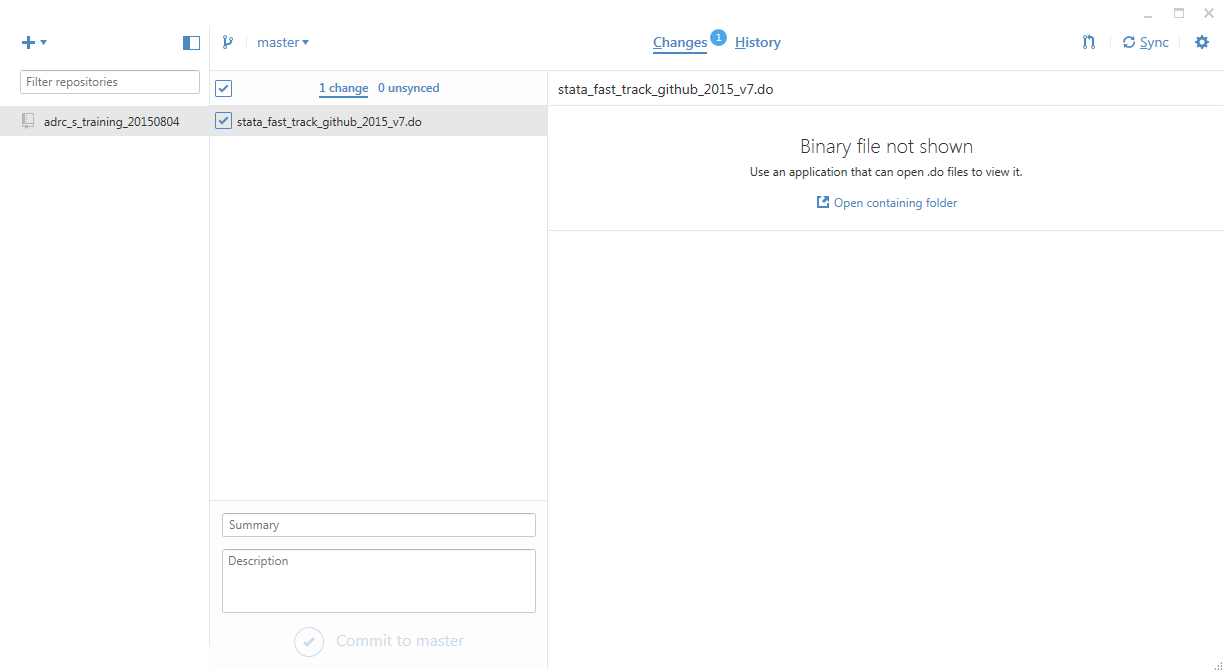
**In this example, this is: A:\ADRC\_S\Training\GitHub\adrc\_s\_training\_20150804**



**The file is visible in the folder:**



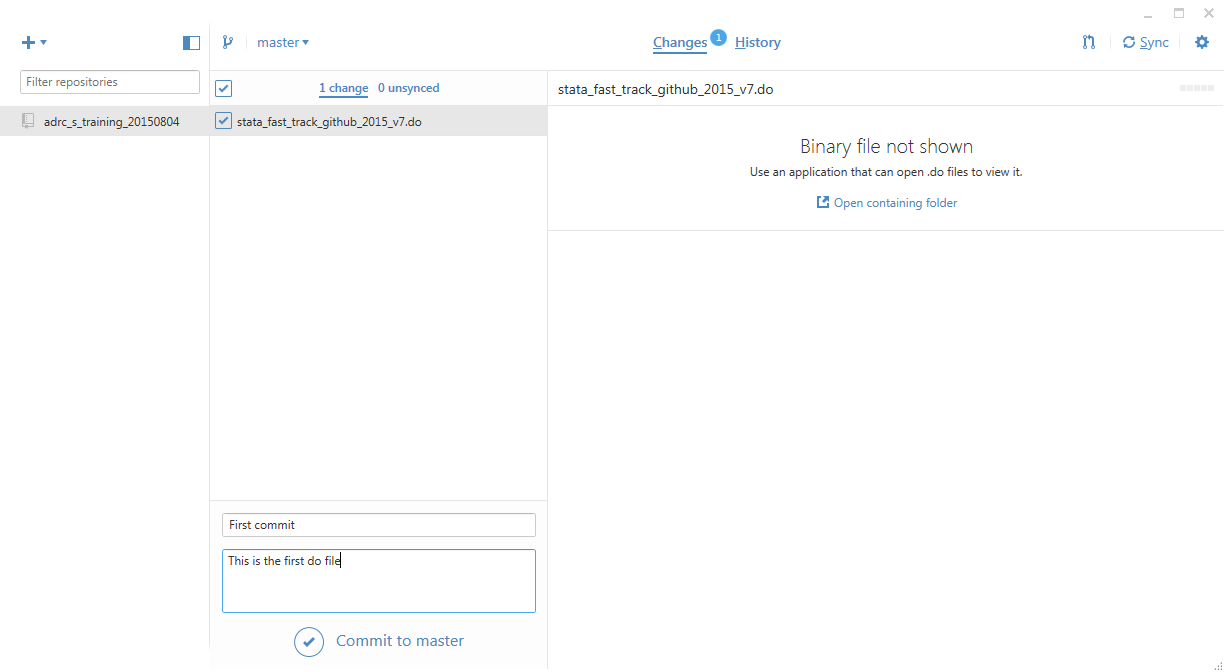
1. **The file is also visible in the GitHub for Windows programme.**



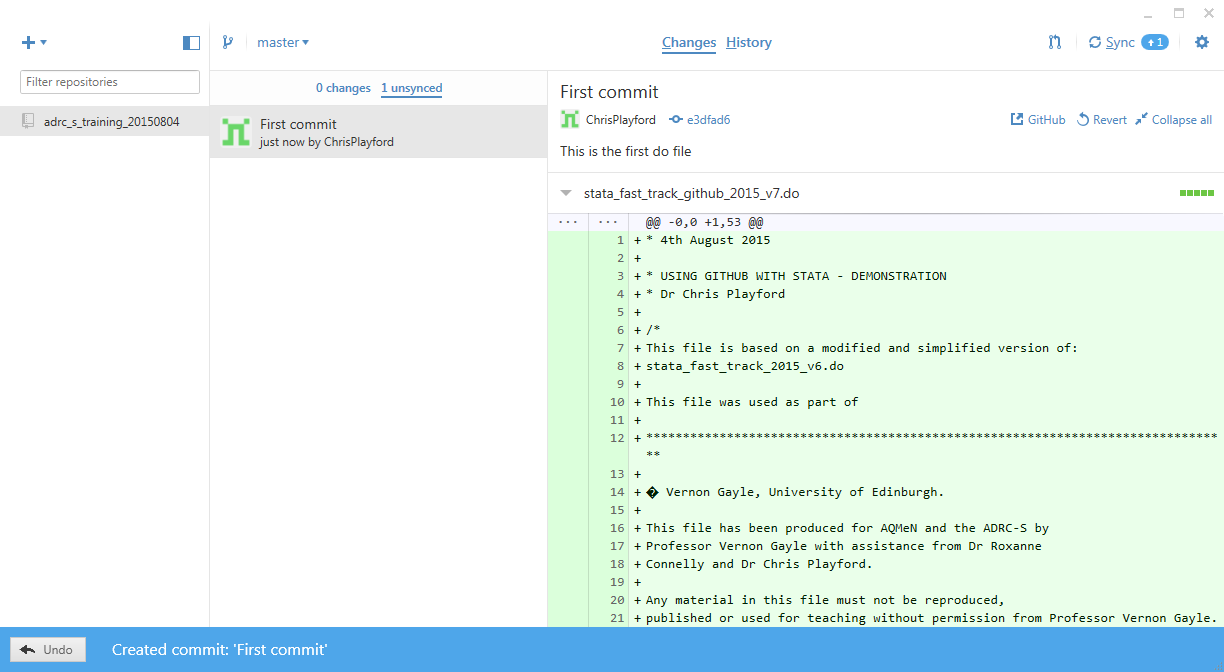
**\* I had to add a line to the .gitattributes file to inform GitHub for windows that .do file are to be treated as text. See below:**

\*.do diff=astextplain

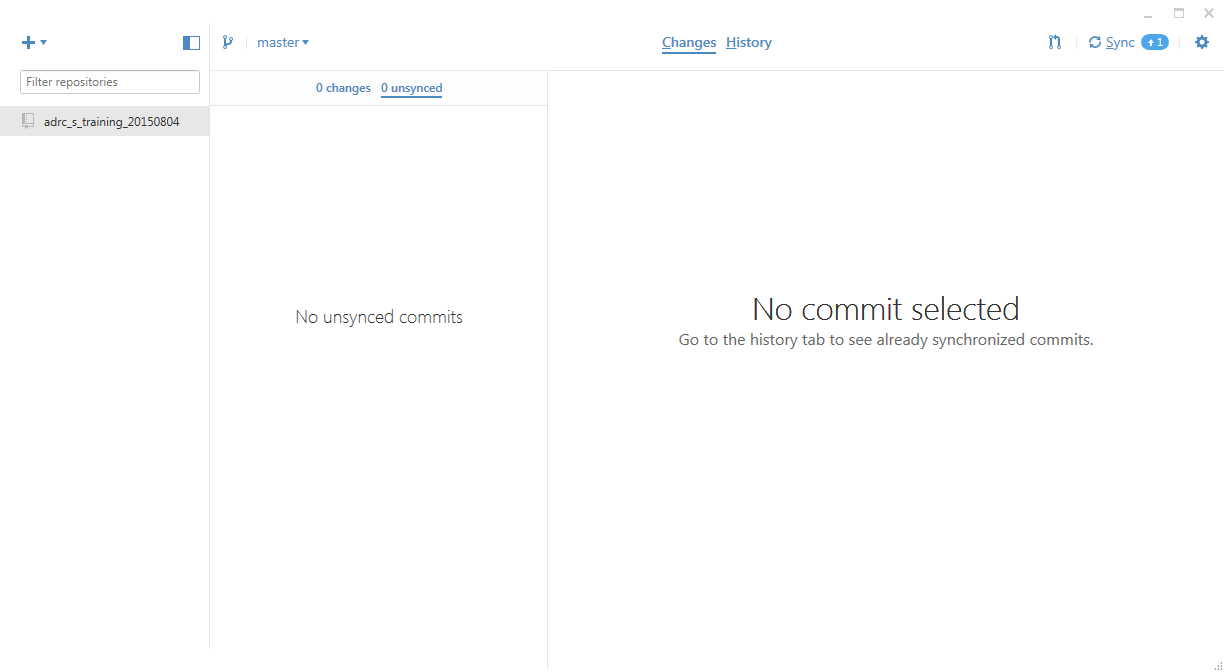
1. **Write Commit message (see bottom of screen)**



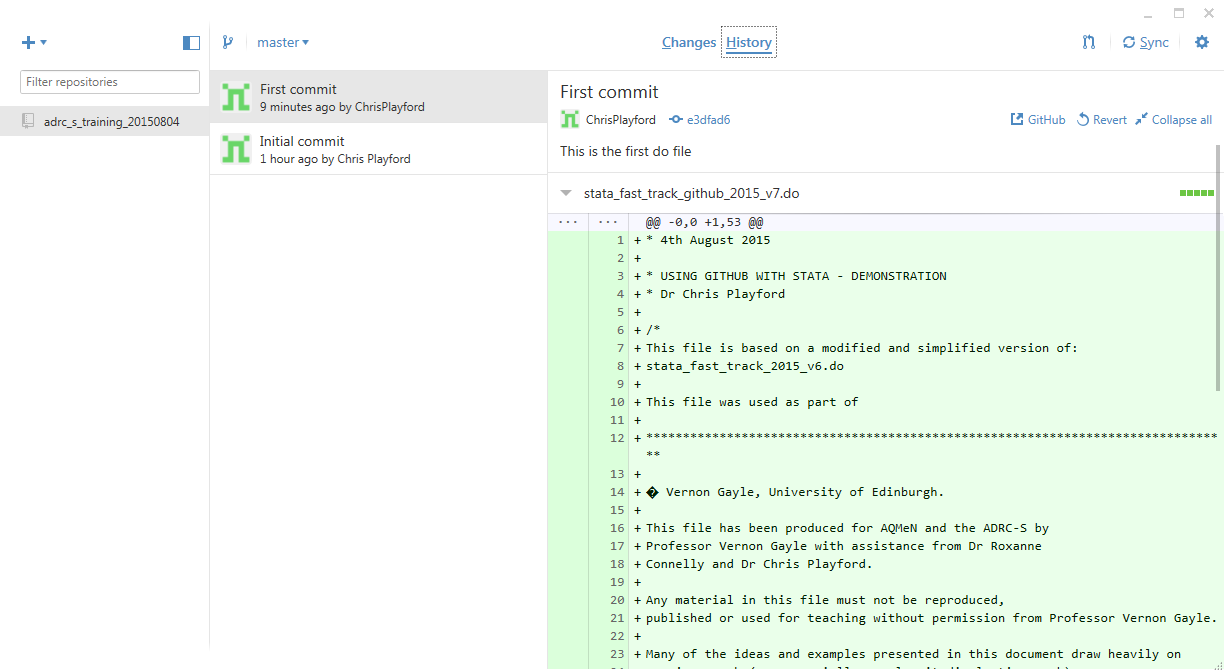
1. **“Commit to master” (main branch)**



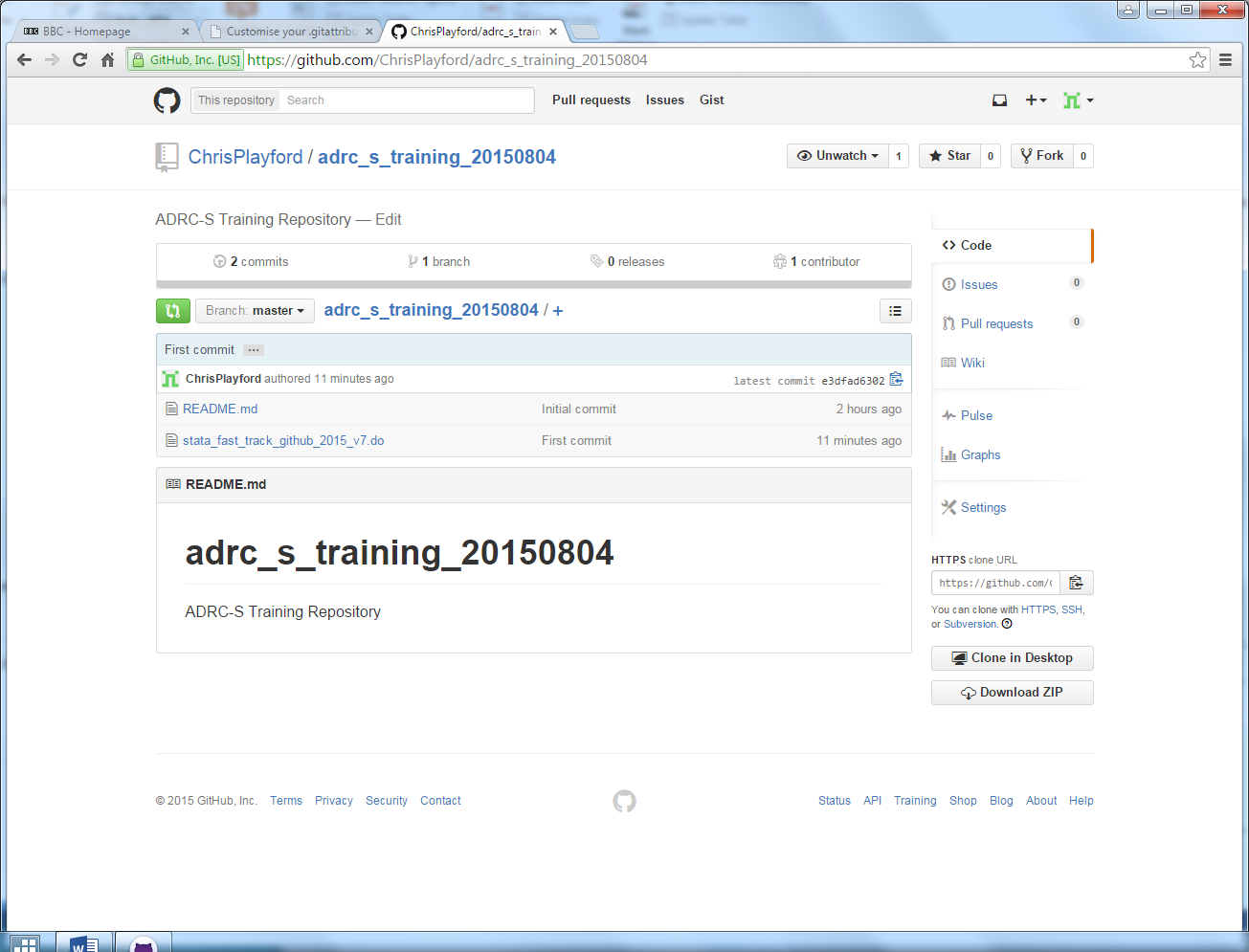
1. **Click on Sync (see top right). The +1 means that the local folder is ahead of the online repository by one commit.**
2. **The commits folder will now be empty as the changes have been pushed to the online repository (I.e. server).**



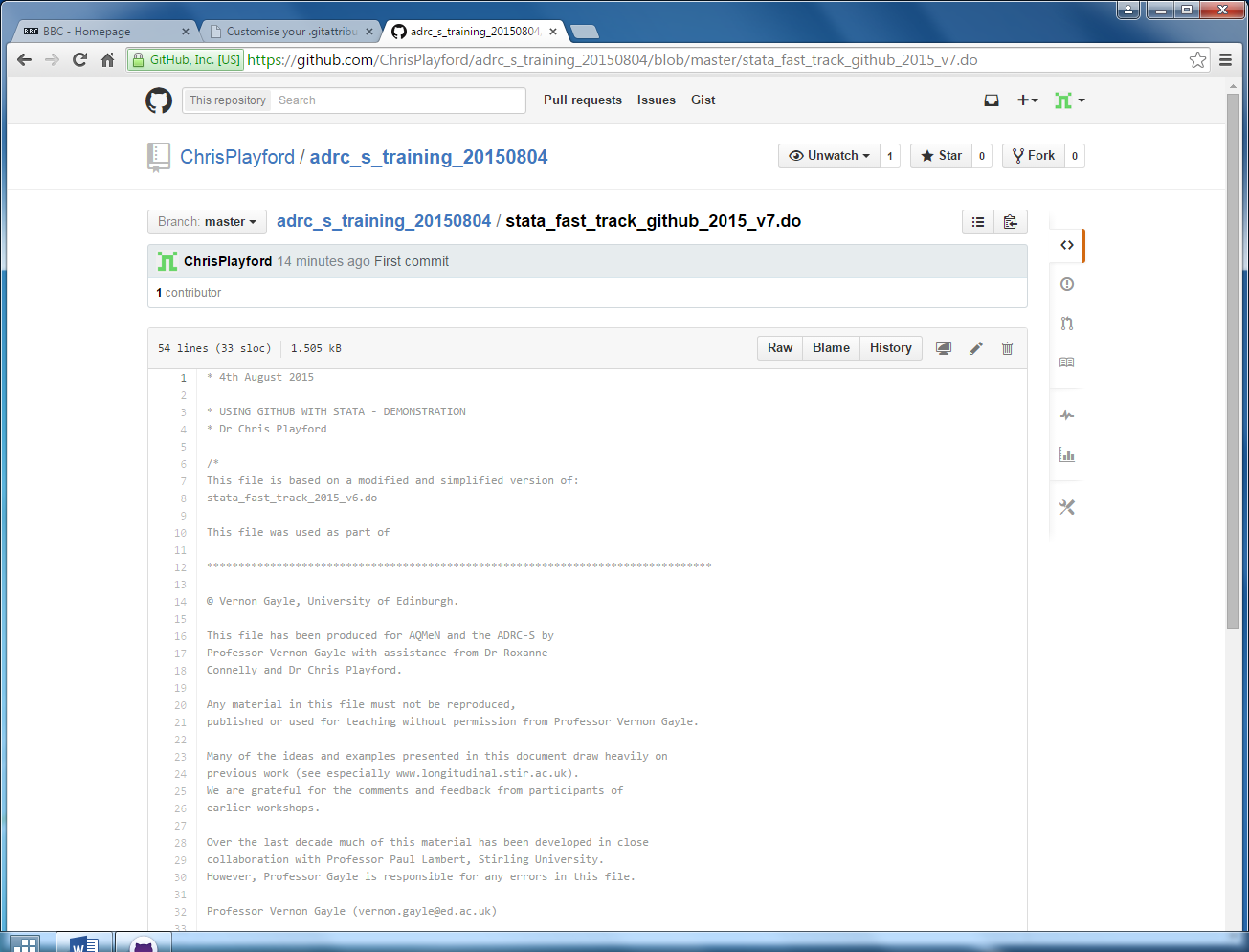
1. **Click on History to view commits**



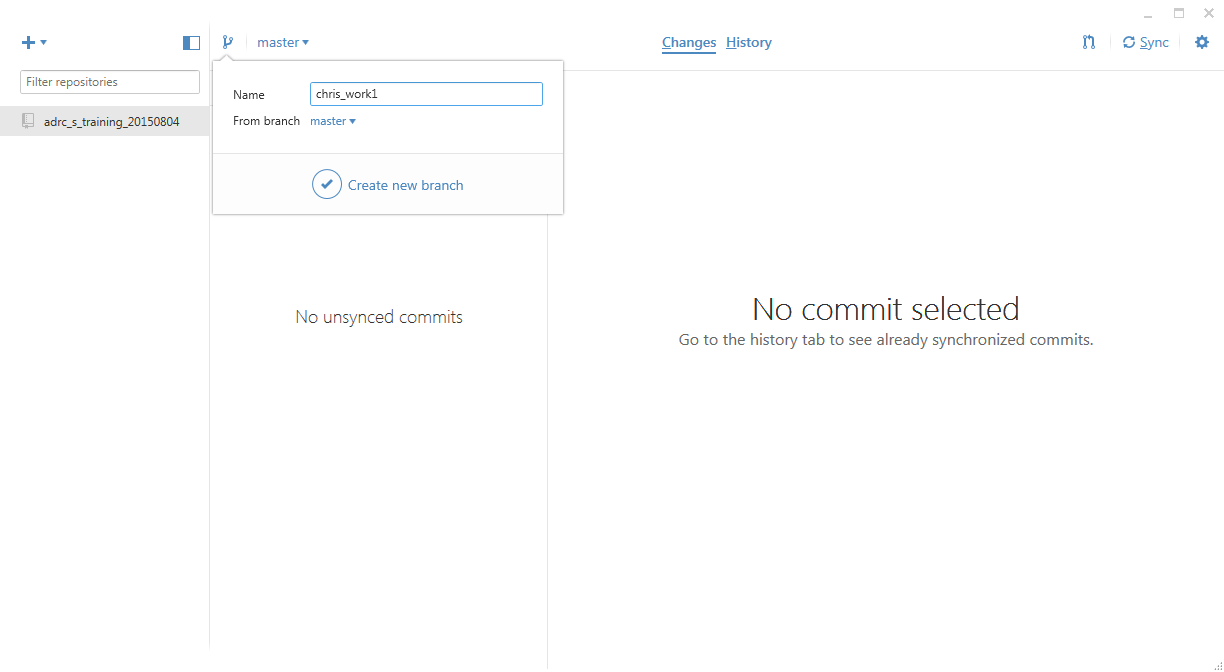
1. **New changes visible on GitHub server (2 commits are visible). Clicking on these will show the commits.**



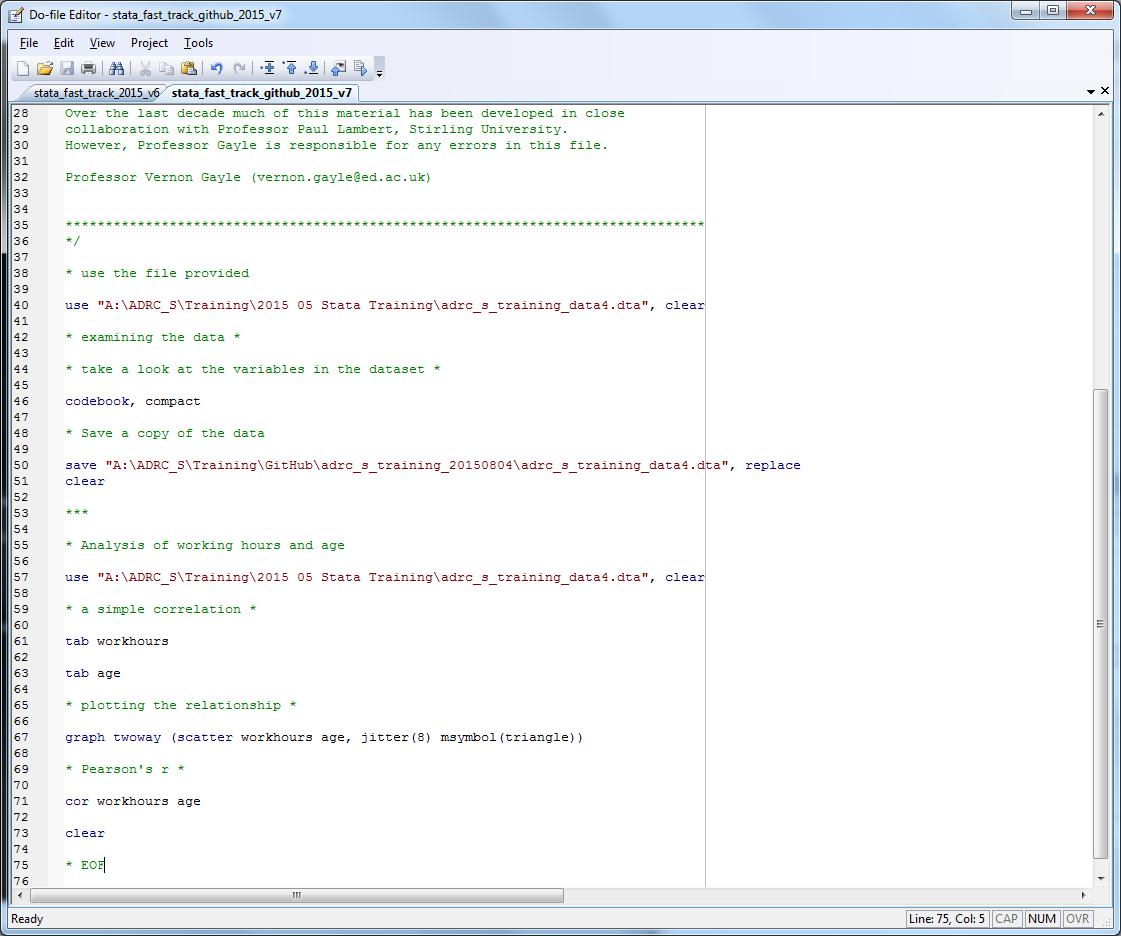
1. **Clicking on the “stata\_fast\_track\_github\_2015\_v7.do” will show preview of file.**



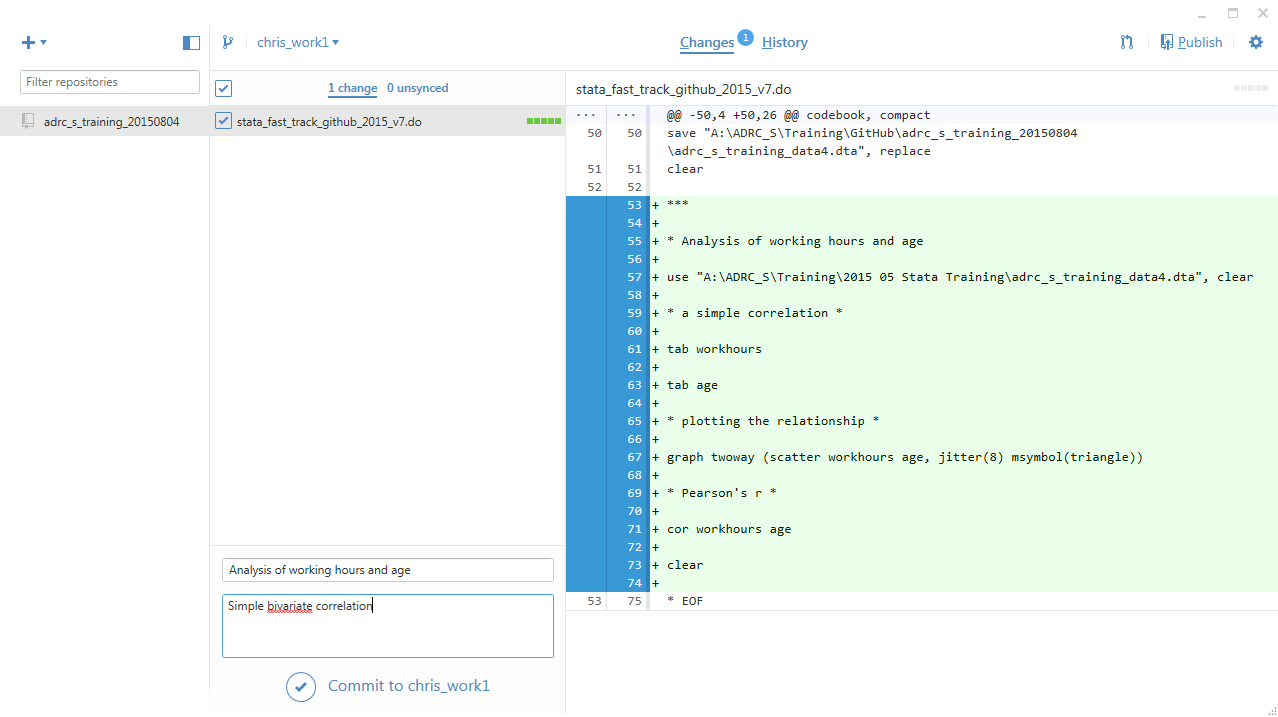
1. **Create new branch for “adrc\_s\_training\_20150804” called “chris\_work1”**



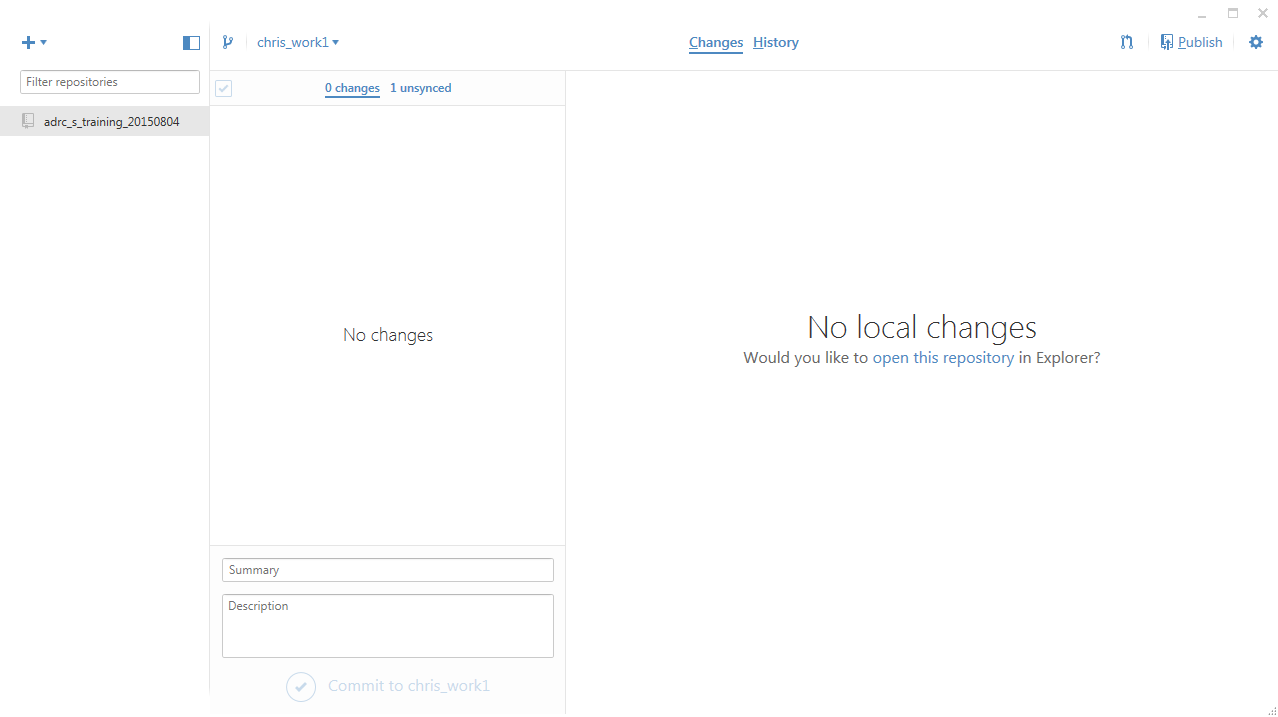
1. **This is the branch working from. Then make some changes to the file:**



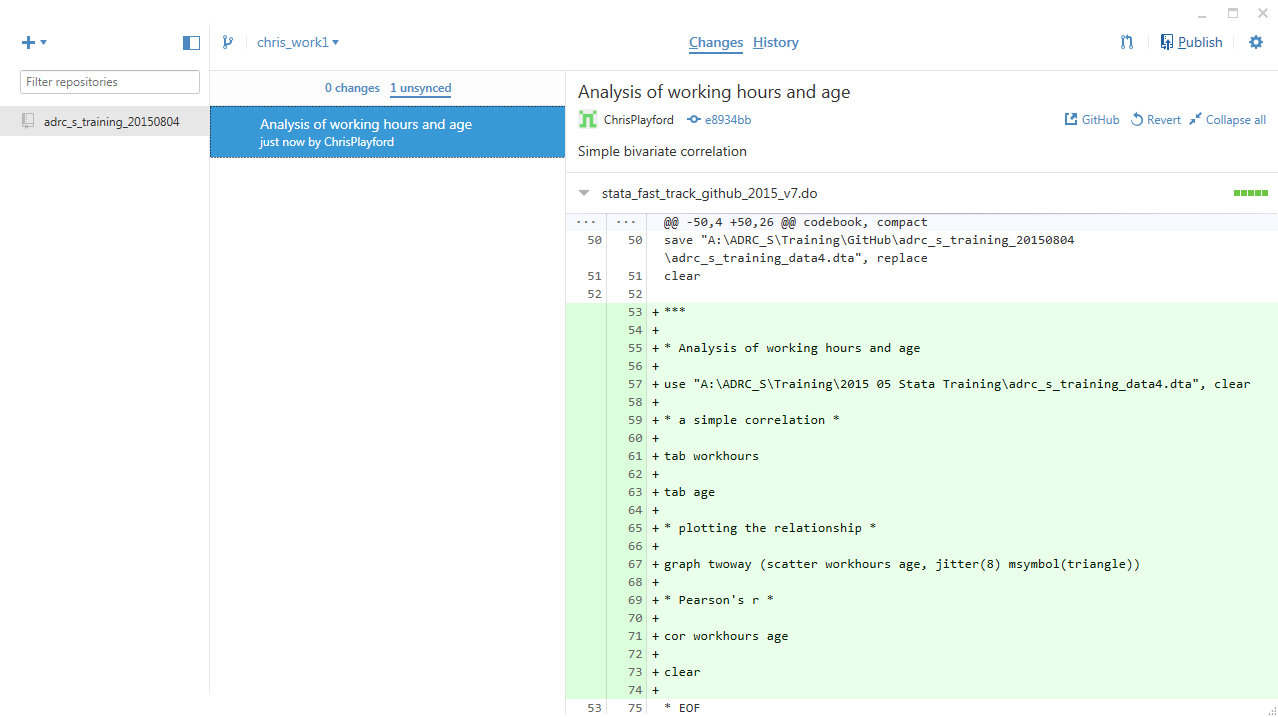
1. **View changes in GitHub for Windows programme. I have written the commit message and am ready to commit this change to the branch “chris\_work1”**



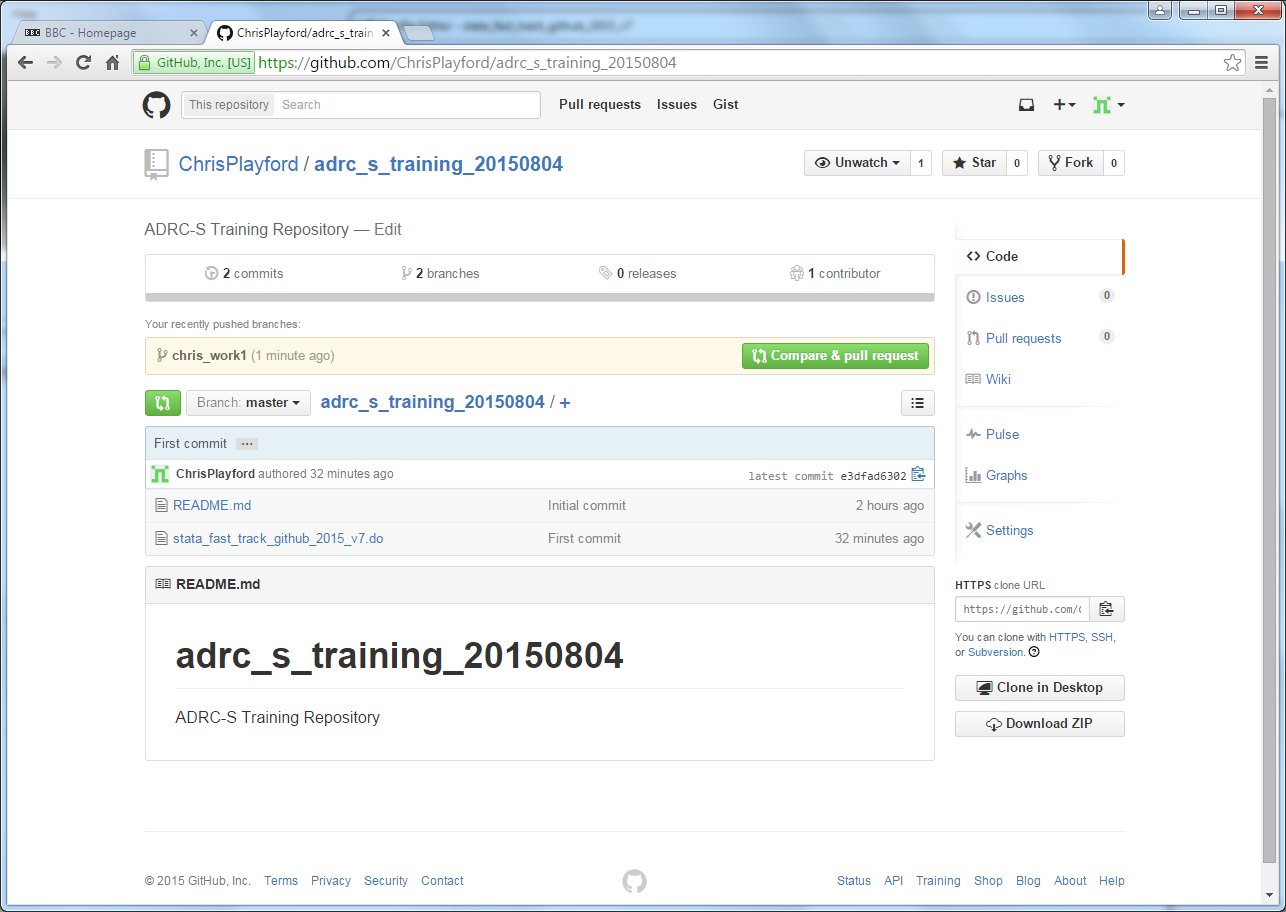
1. **The changes will now not display anything but there is 1 unsynced changed. Click on this.**



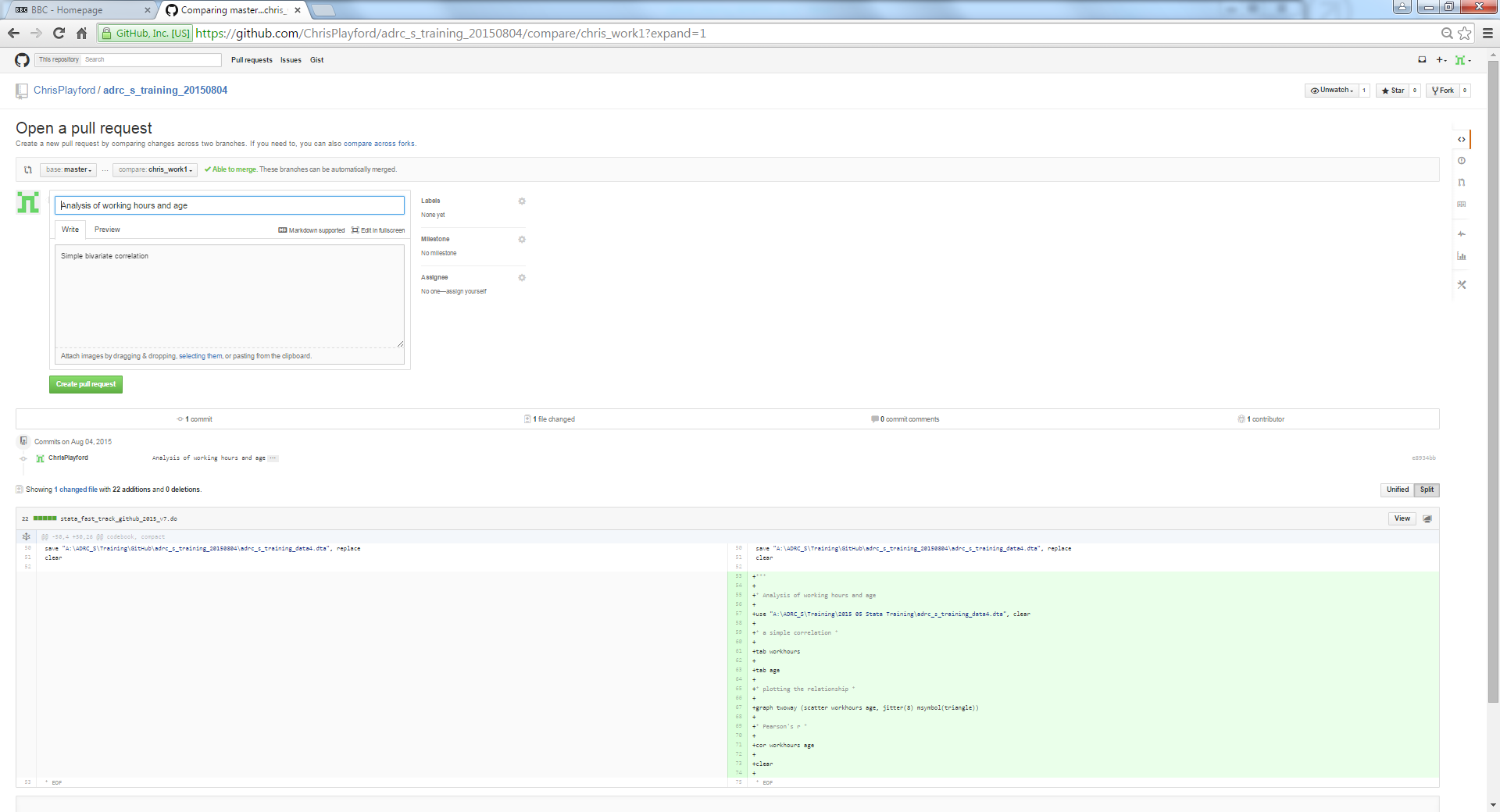
1. **Unsynced change. Because this is a new branch, the option is to publish this branch (see top right) rather than to sync it (it does not yet exist on the online repository).**



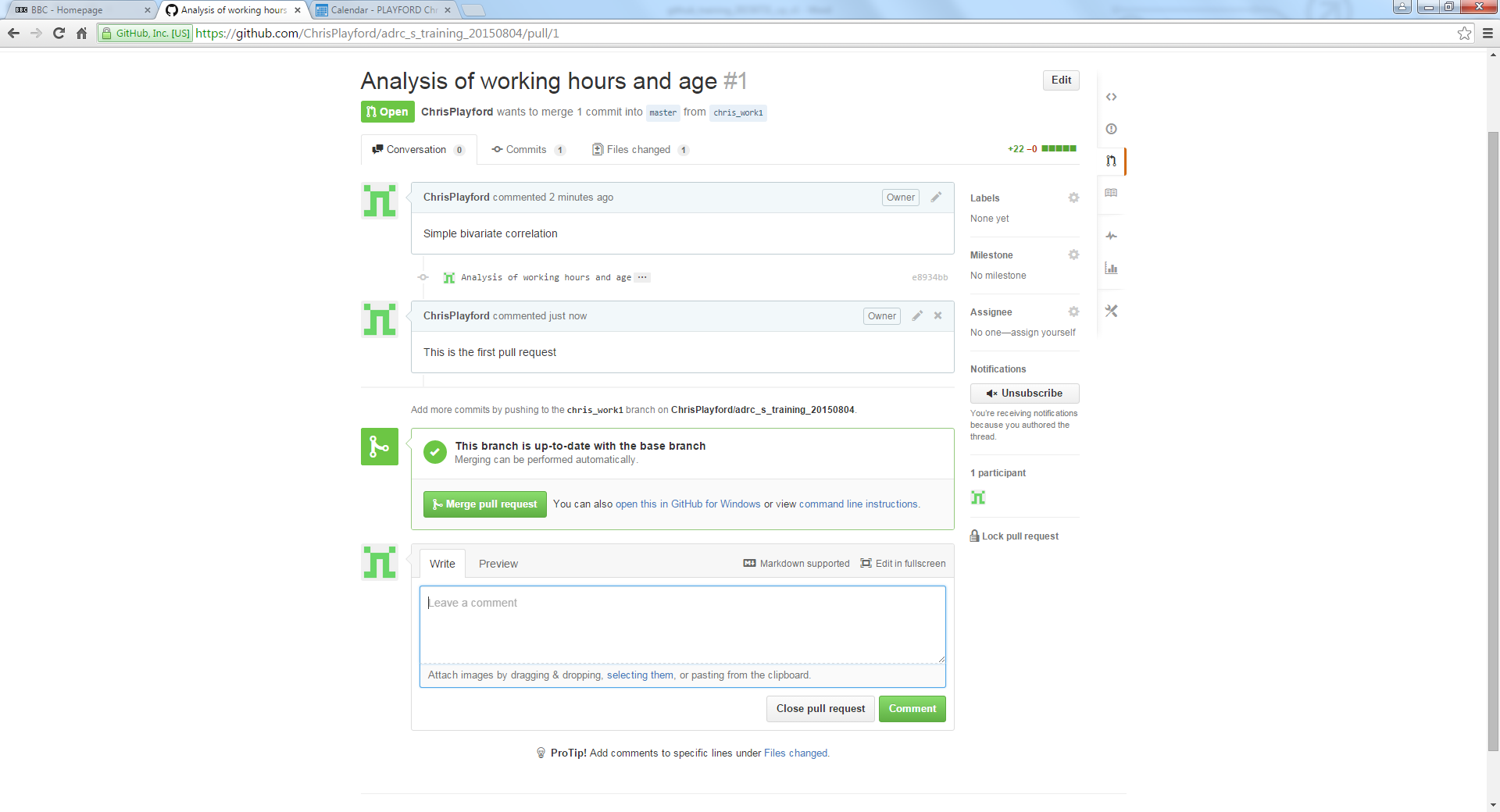
1. **The changes are visible in the GitHub repository. There are now two branches. The new branch is visible in the middle of the screen. There is also the option to “Compare and Pull Request”.**



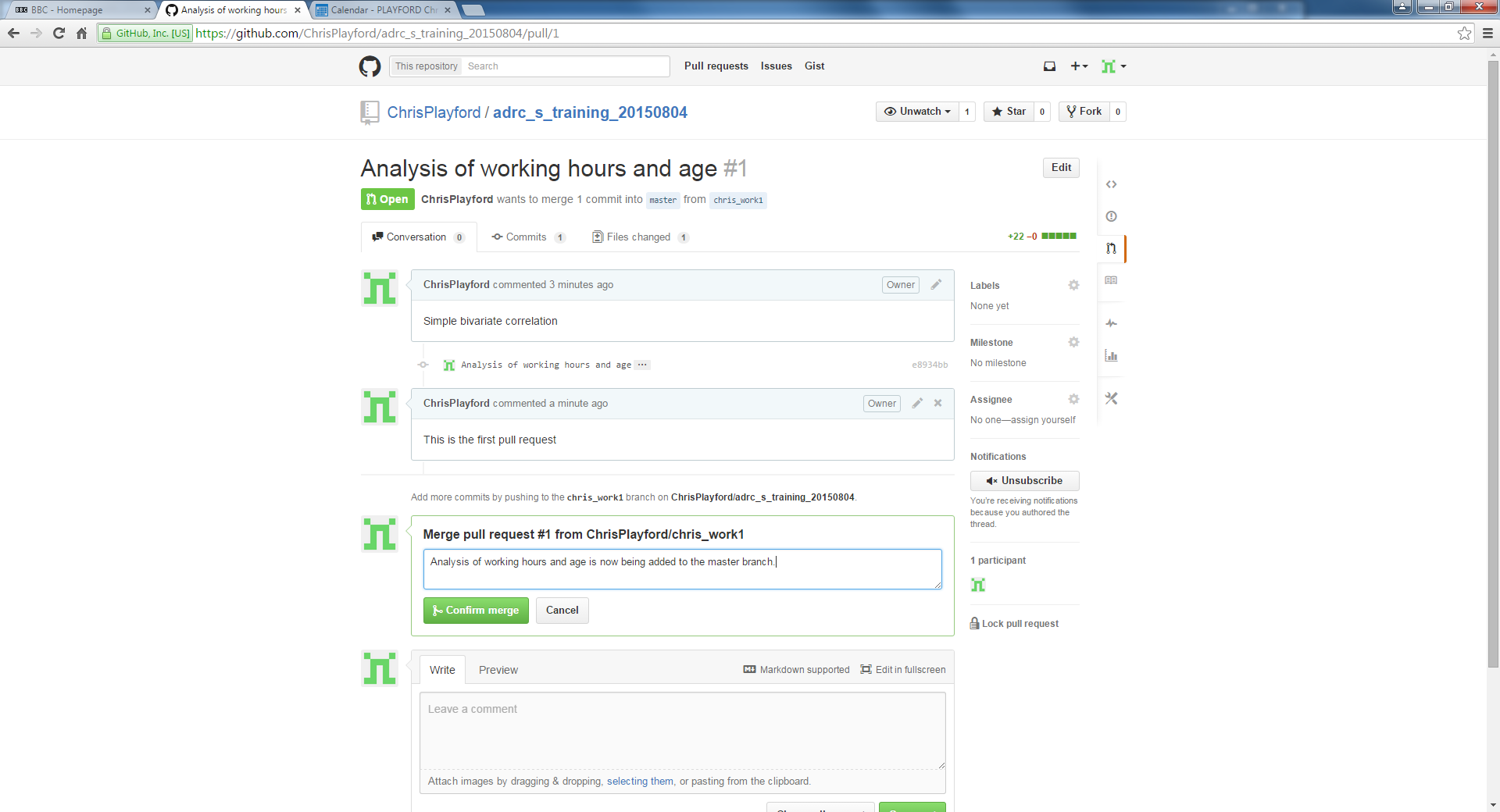
1. **The changes made are visible**



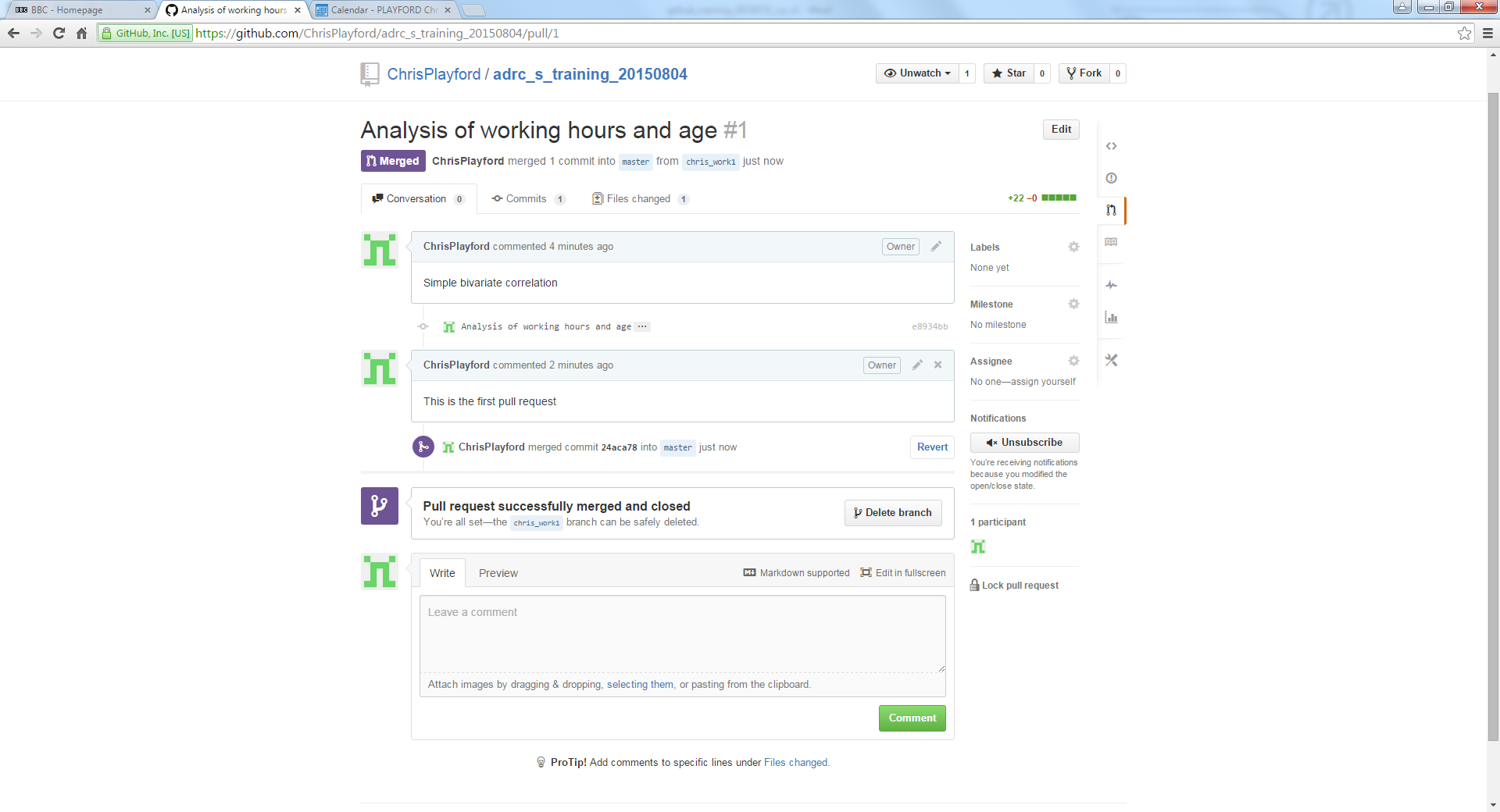
1. **Click on “Create Pull Request” and add a comment “this is the first pull request”.**



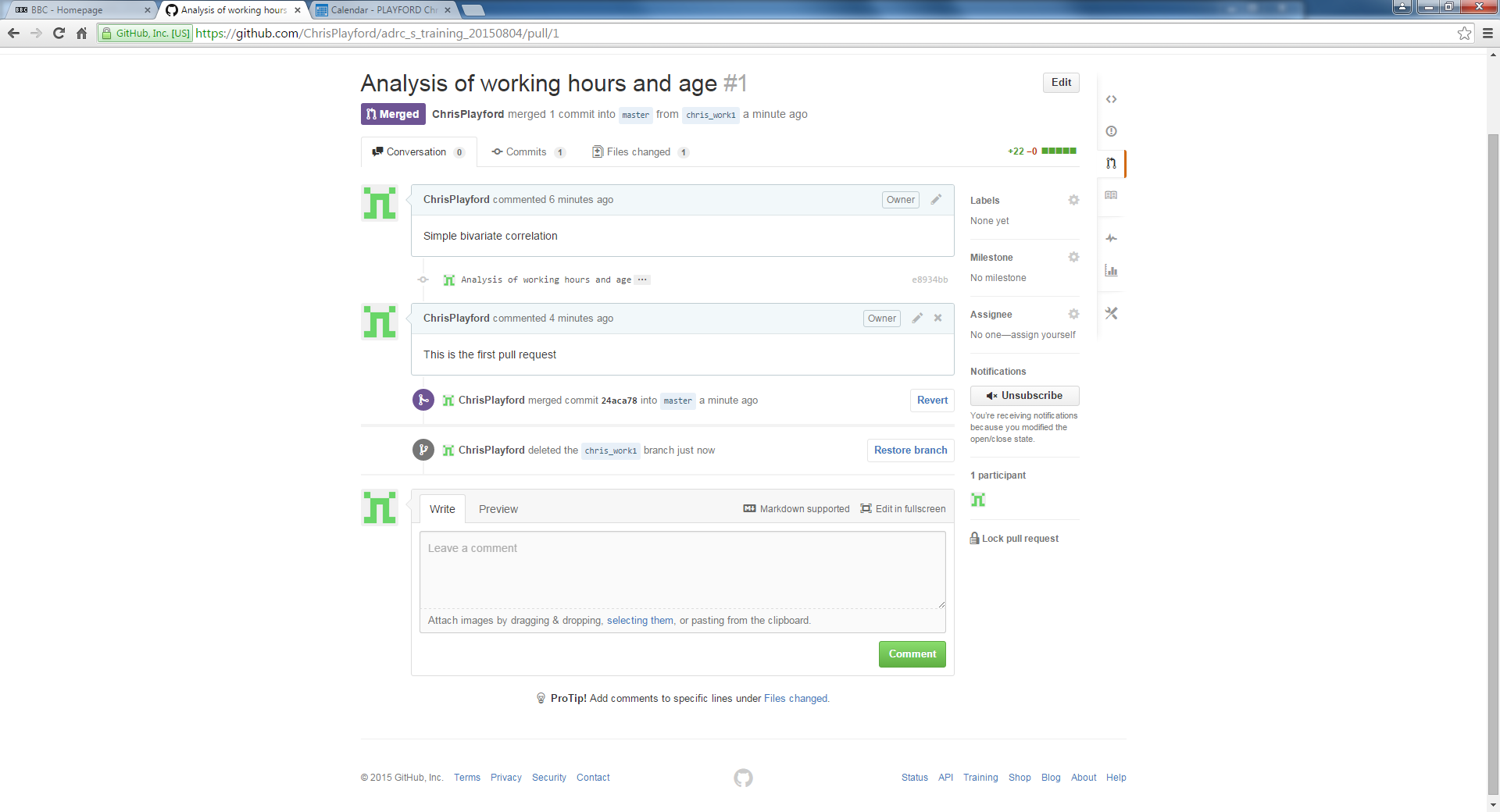
1. **Click on “Merge Pull Request” and add comment “Analysis of working hours and age is now being added to the master branch”.**



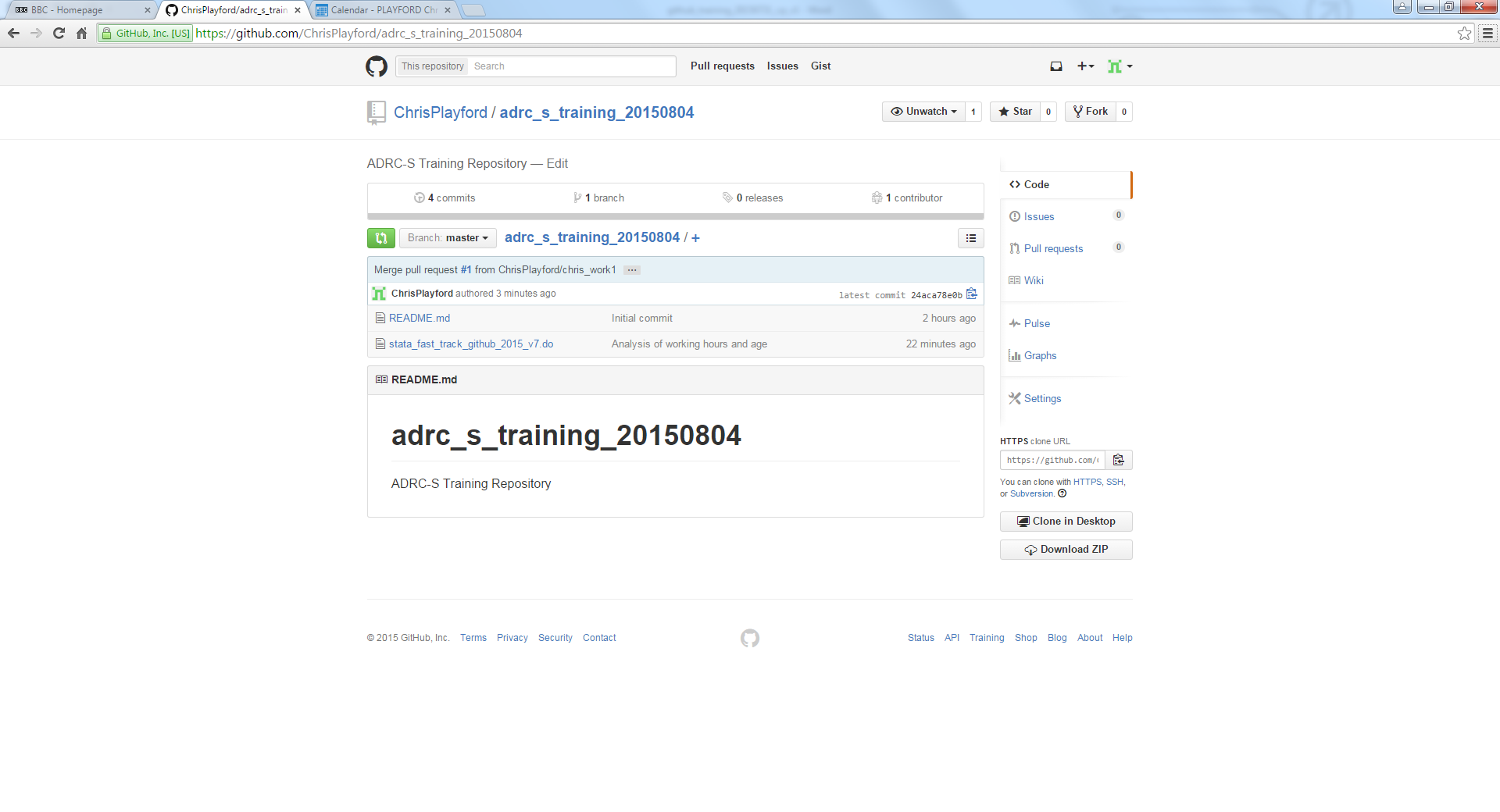
1. **Click on “Confirm Merge”. The pull request has been successfully merged.**



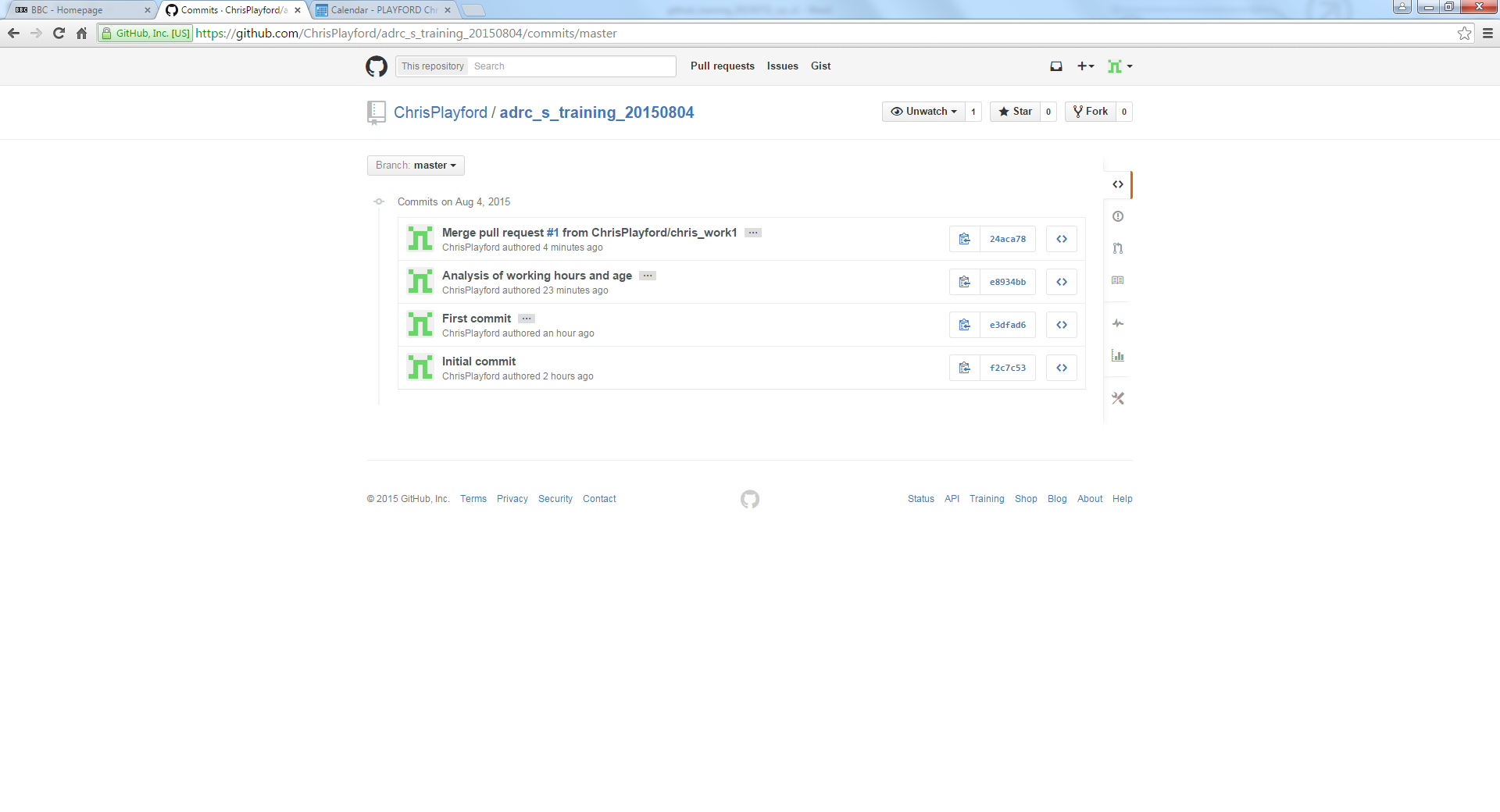
1. **The old branch “chris\_work1” can now be deleted as it has served its purpose.**



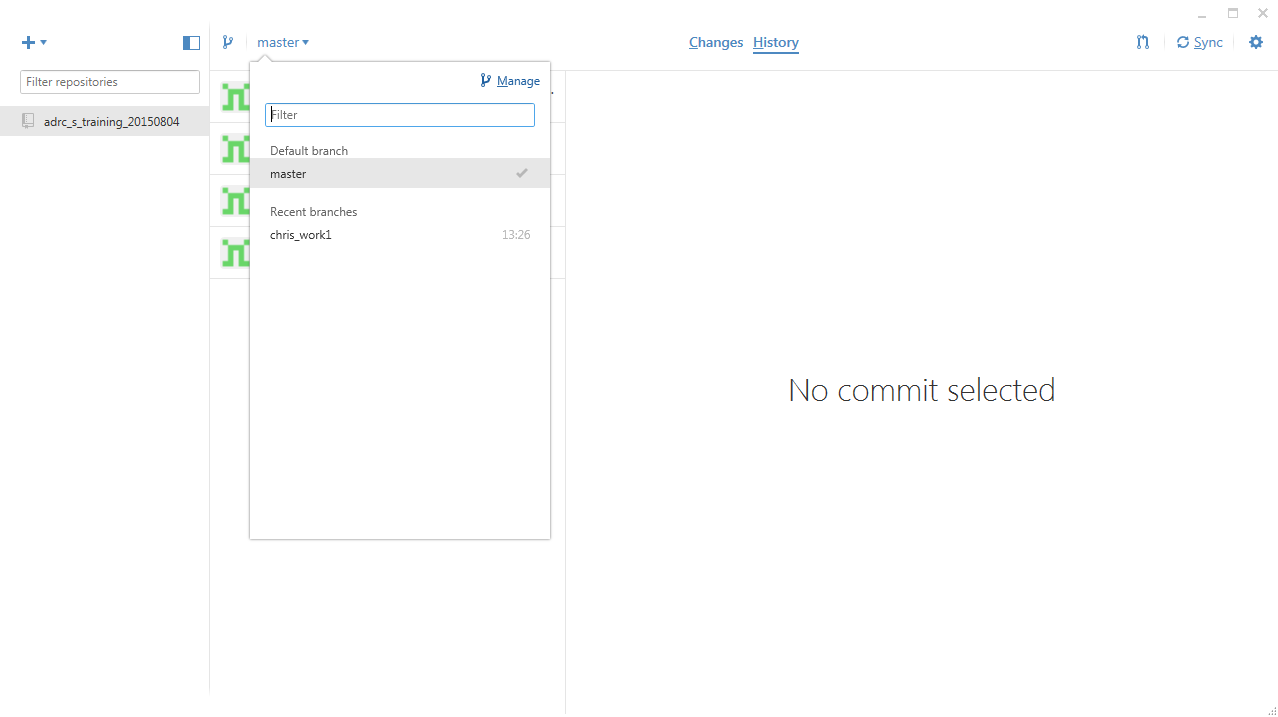
1. **There is now one branch (master)**



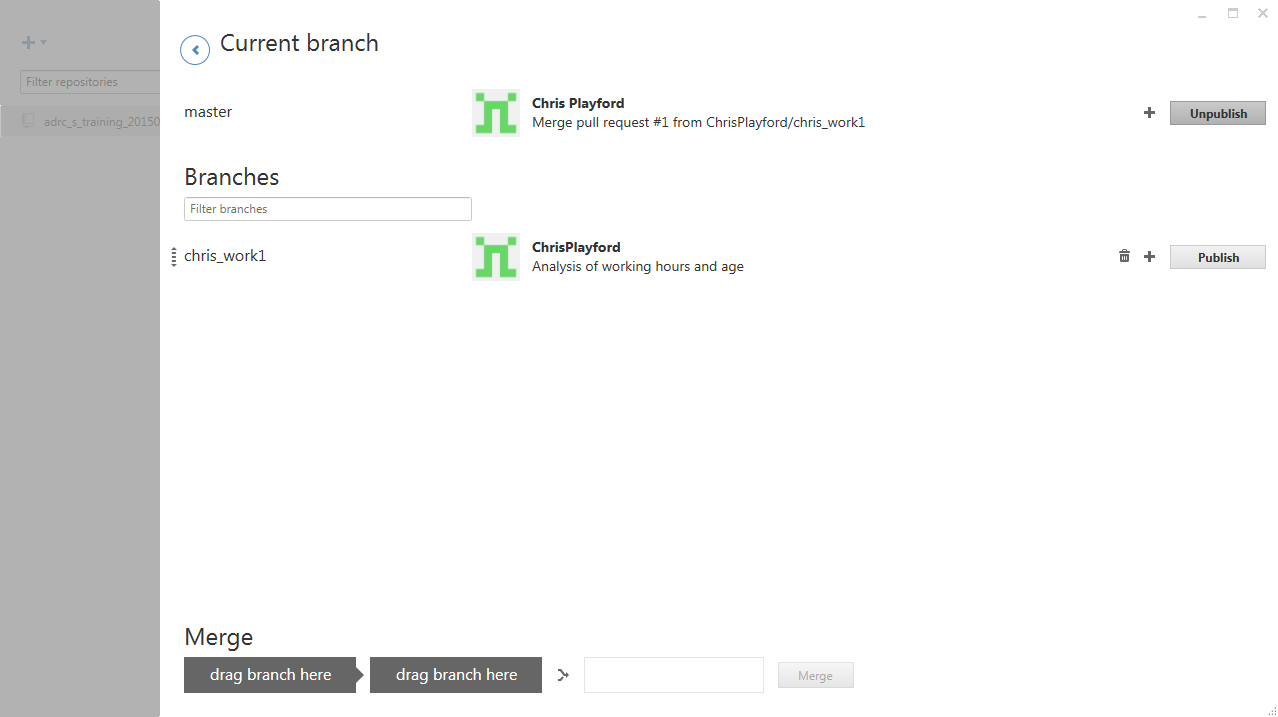
1. **View the history to see all the changes that have been made.**



1. **The old branch will still exist on the GitHub for Windows programme with a record of the changes made. It can be deleted if desired but it is not necessary (this is achieved through clicking on “Manage”).**



**The “Manage” screen is shown below.**



For further details on pull requests, see:

<https://help.github.com/articles/using-pull-requests/>

<https://help.github.com/articles/merging-a-pull-request/>

**Useful websites**

For further guidelines please see the following useful websites:

<http://blogs.lse.ac.uk/impactofsocialsciences/2013/06/04/github-for-academics/>

<http://chronicle.com/blogs/profhacker/tag/github101>

Many of the websites will refer to the use of Git through the command line. This is something I have not covered but may be preferable to other users (or indeed to me in the future). I have chosen to demonstrate the principles using the GUI as I think this is clearer for the uninitiated.

For example, this YouTube clip demonstrates these principles using the command line rather than Git for Windows: <https://www.youtube.com/watch?v=E8TXME3bzNs>