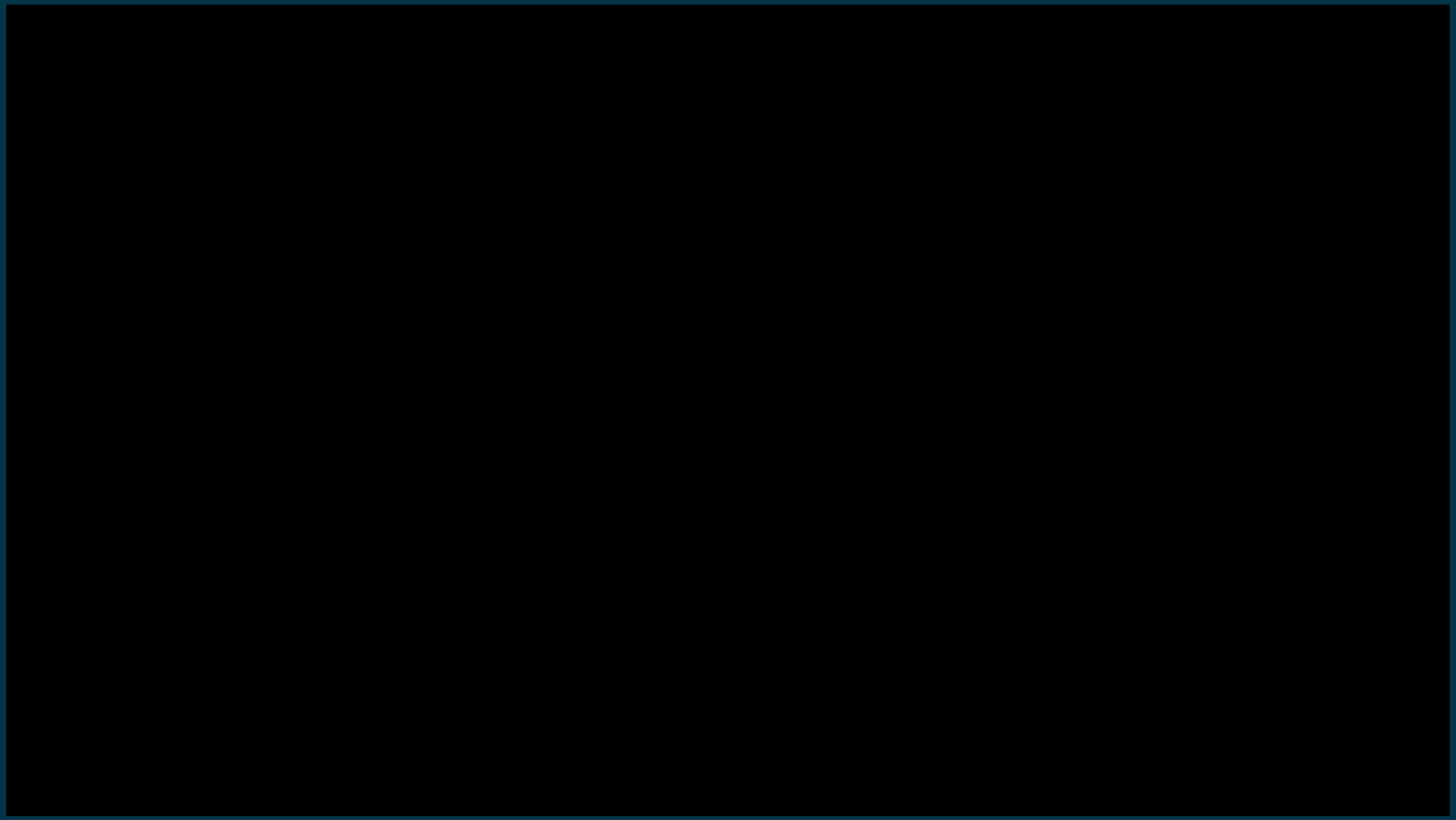
What is GitHub?



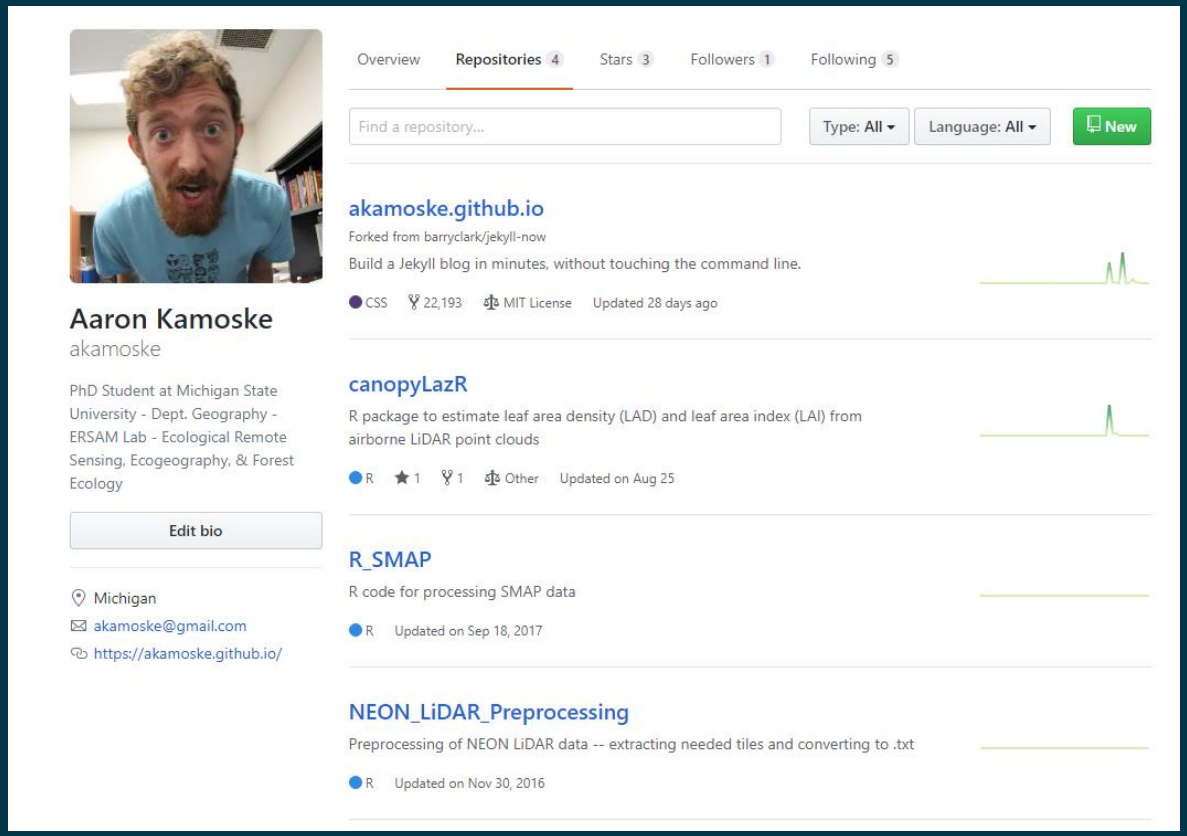
https://youtu.be/w3jLJU7DT5E

Why should we use GitHub?

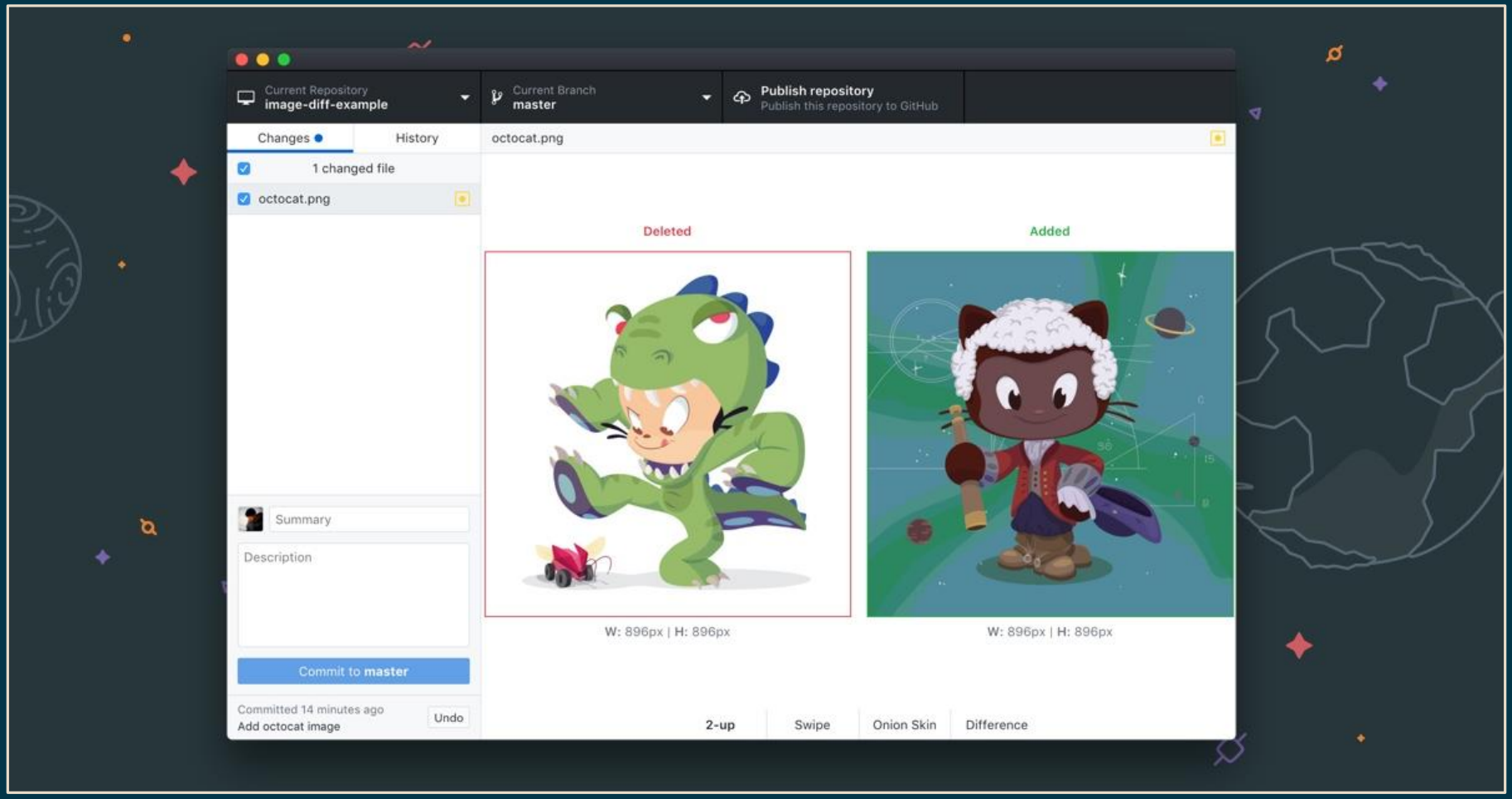
- Stay organized
- Easily collaborate with others
- Roll back to earlier versions
- Have all your code in one secure place
- Track your changes
- Share your code while keeping track of future changes

Where do we store projects on GitHub?

A GitHub repo – or repository – is a storage space where your projects can live. It can be a local folder on your computer or an online storage space such as on GitHub or the cloud.



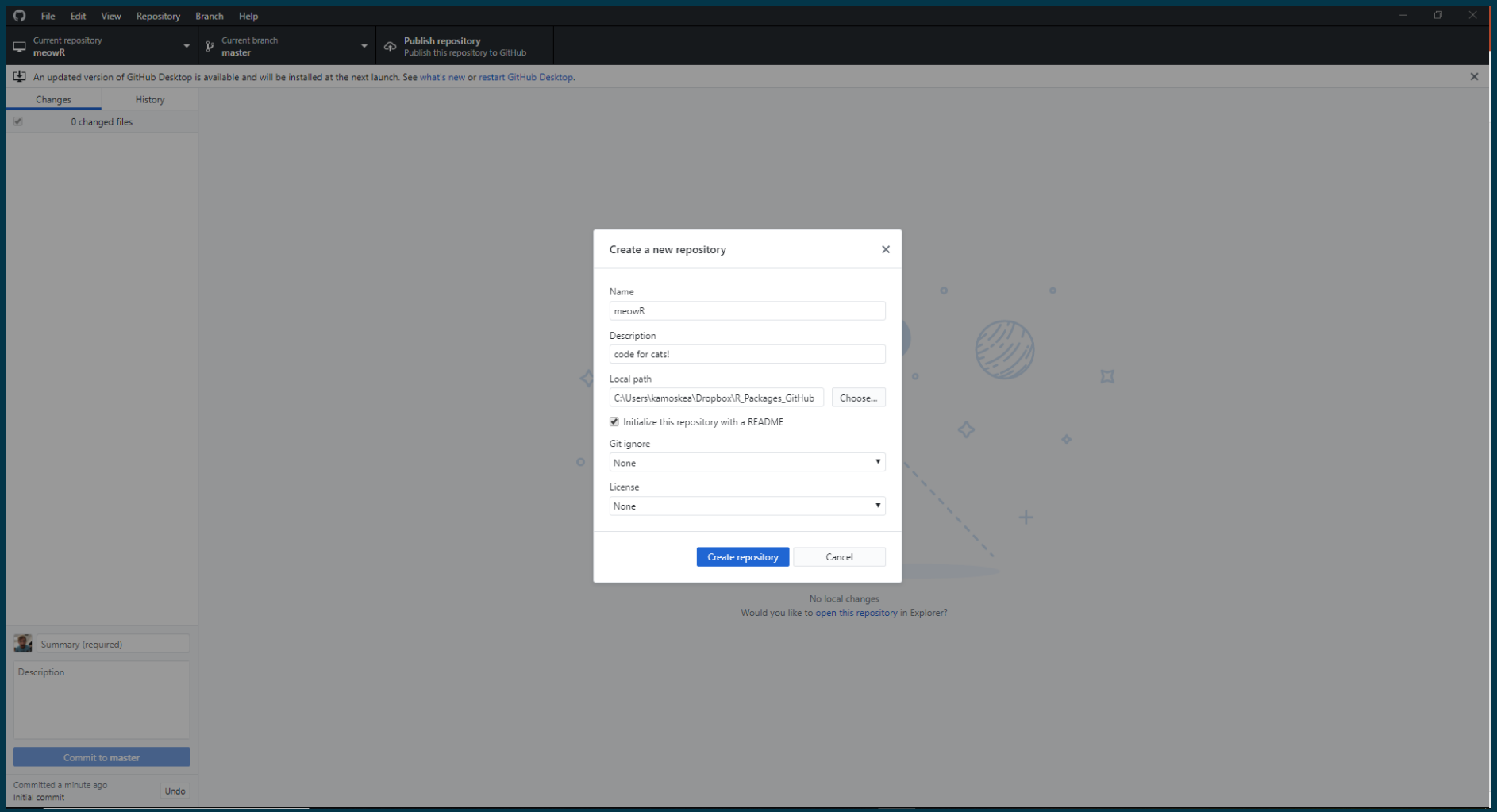
How do we access GitHub?



GitHub Desktop is awesome!

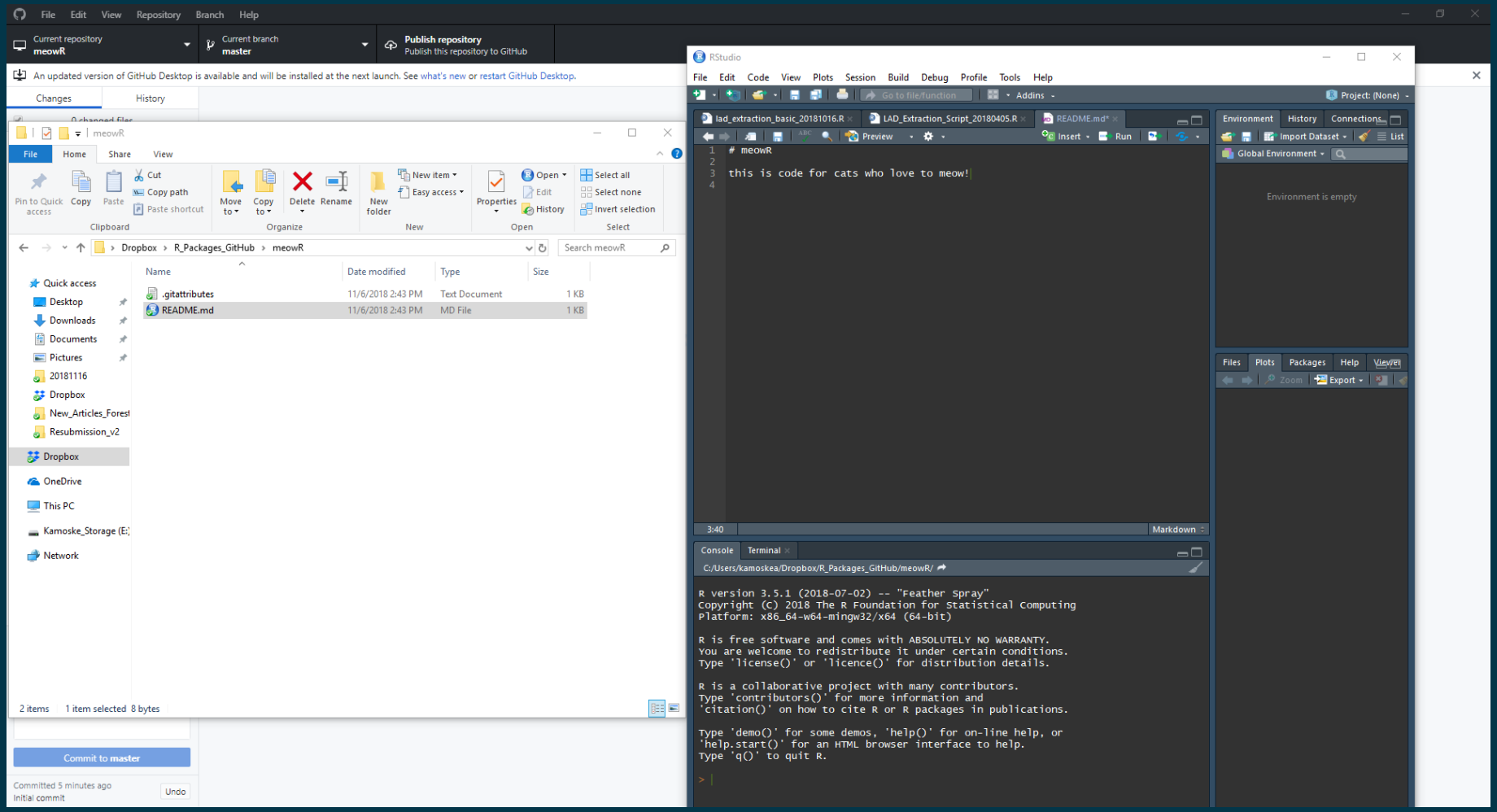
https://desktop.github.com/

GitHub Desktop Quick Demo



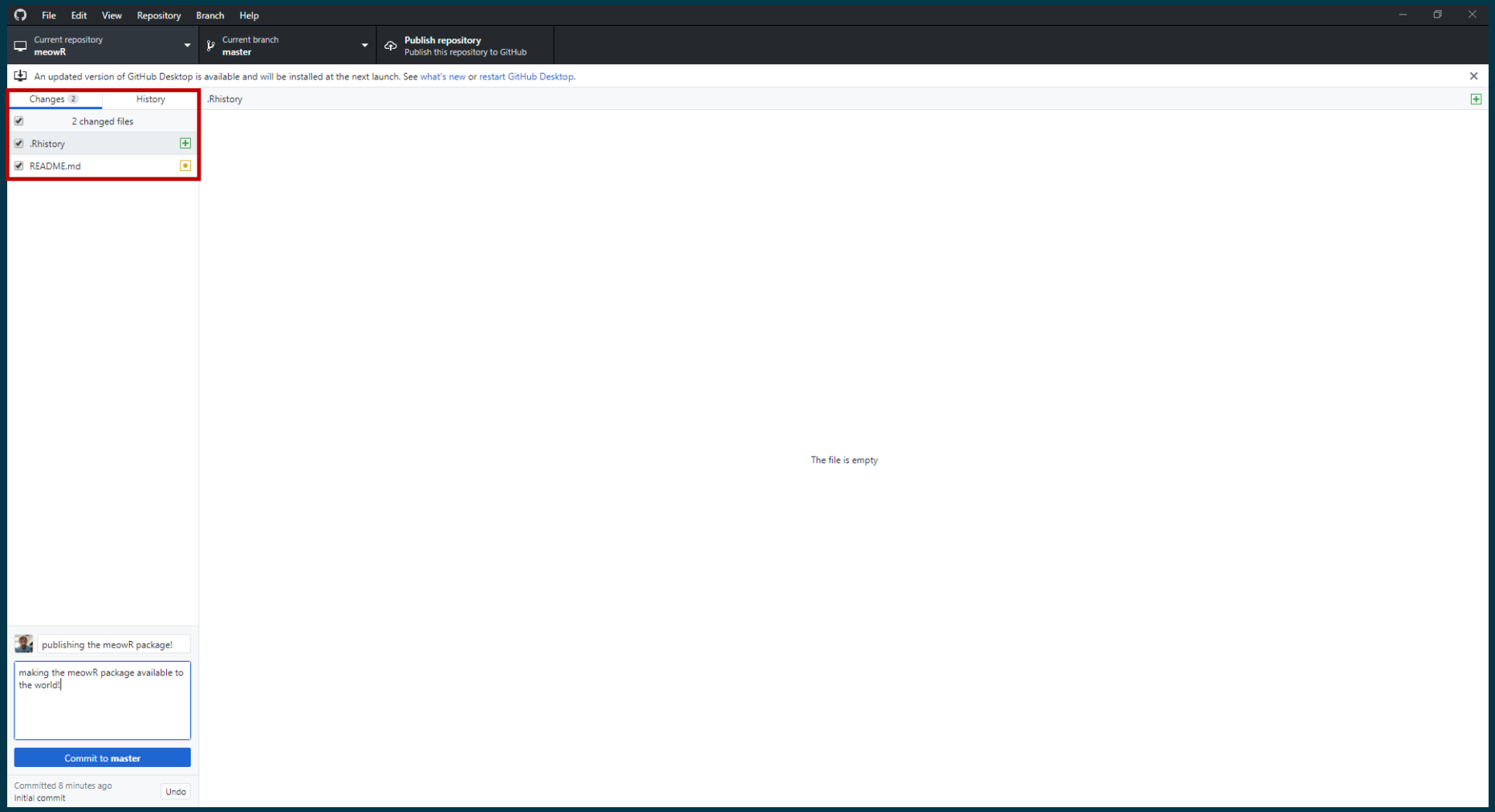
First, let’s create a new repository in our GitHub folder!

GitHub Desktop Quick Demo



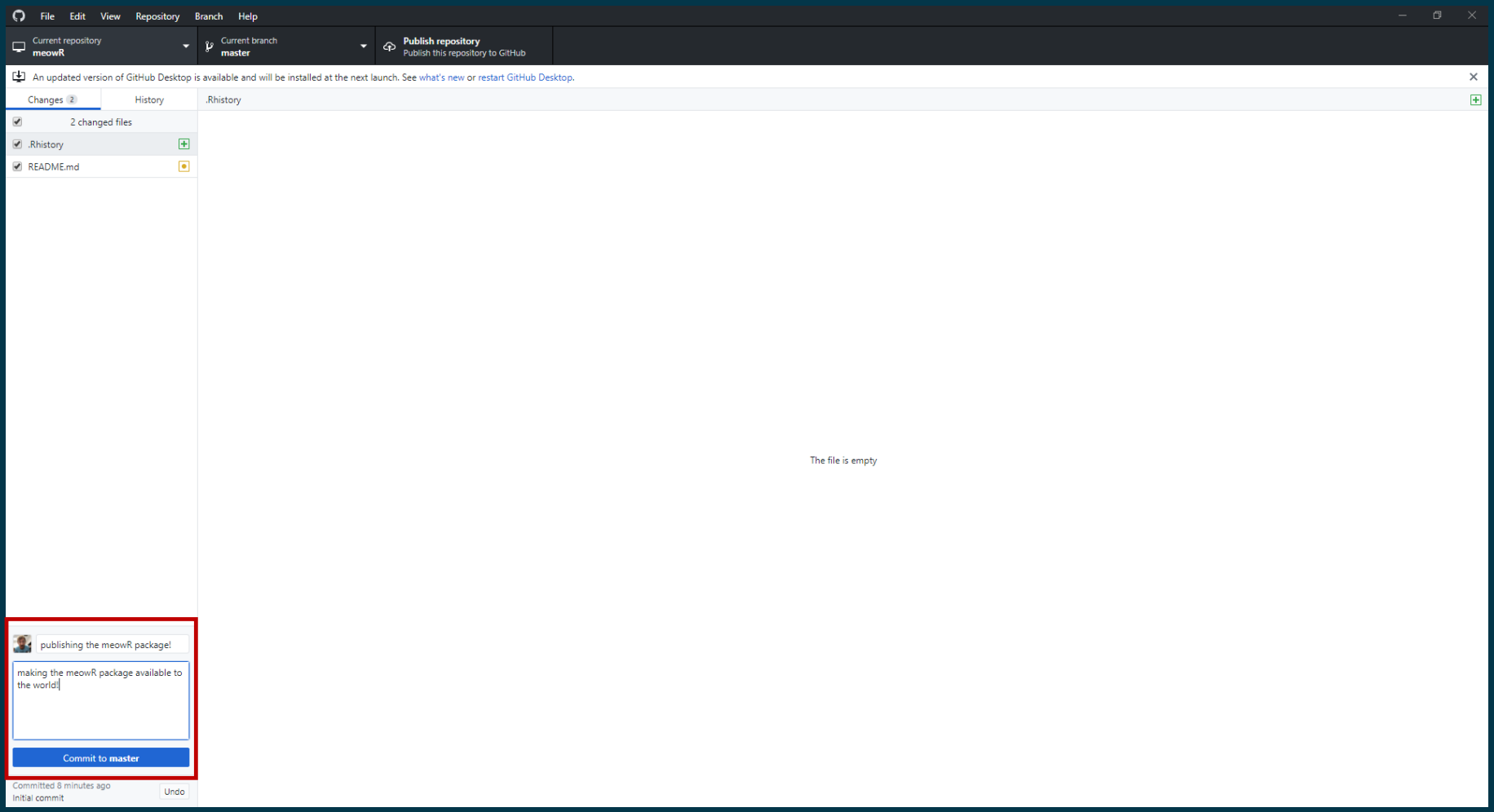
Let's open the README.md file and let the world know what our project does!

GitHub Desktop Quick Demo



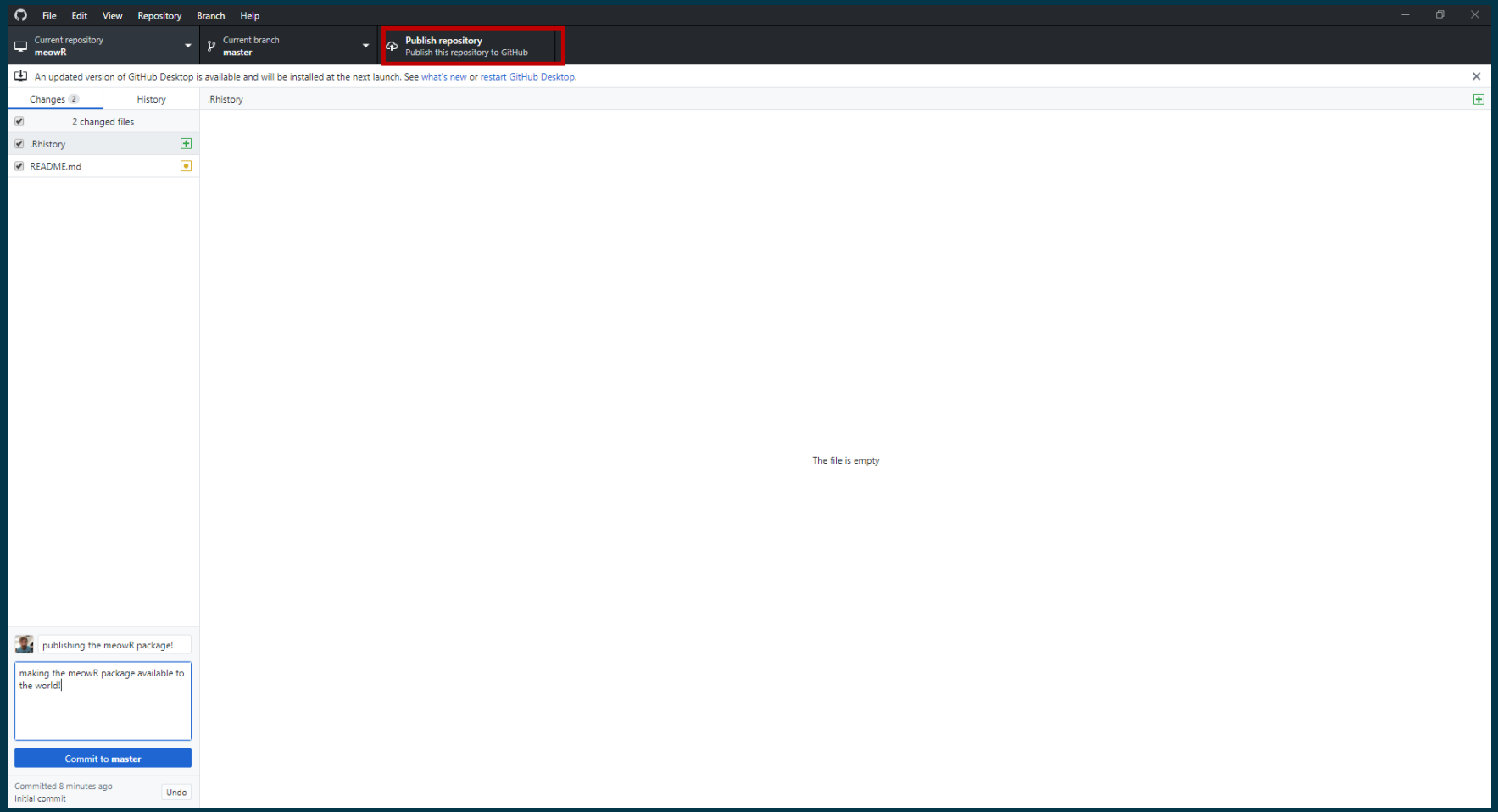
We can see the changes that we made to our project here – notice the README file.

GitHub Desktop Quick Demo



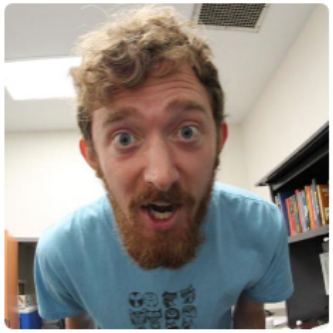
It is a good practice to write a quick commit about what we changed and why – especially for our future selves!

GitHub Desktop Quick Demo



Hit that publish button!

GitHub Desktop Quick Demo



Aaron Kamoske
akamoske

PhD Student at Michigan State University - Dept. Geography - ERSAM Lab - Ecological Remote Sensing, Ecogeography, & Forest Ecology

Edit bio

Michigan
akamoske@gmail.com
https://akamoske.github.io/

Overview | **Repositories 5** | Stars 3 | Followers 1 | Following 5

Find a repository... | Type: All | Language: All | New

meowR
code for cats!
Updated a minute ago

akamoske.github.io
Forked from barryclark/jekyll-now
Build a Jekyll blog in minutes, without touching the command line.
CSS 22,193 MIT License Updated 28 days ago

canopyLazR
R package to estimate leaf area density (LAD) and leaf area index (LAI) from airborne LiDAR point clouds
R 1 1 Other Updated on Aug 25

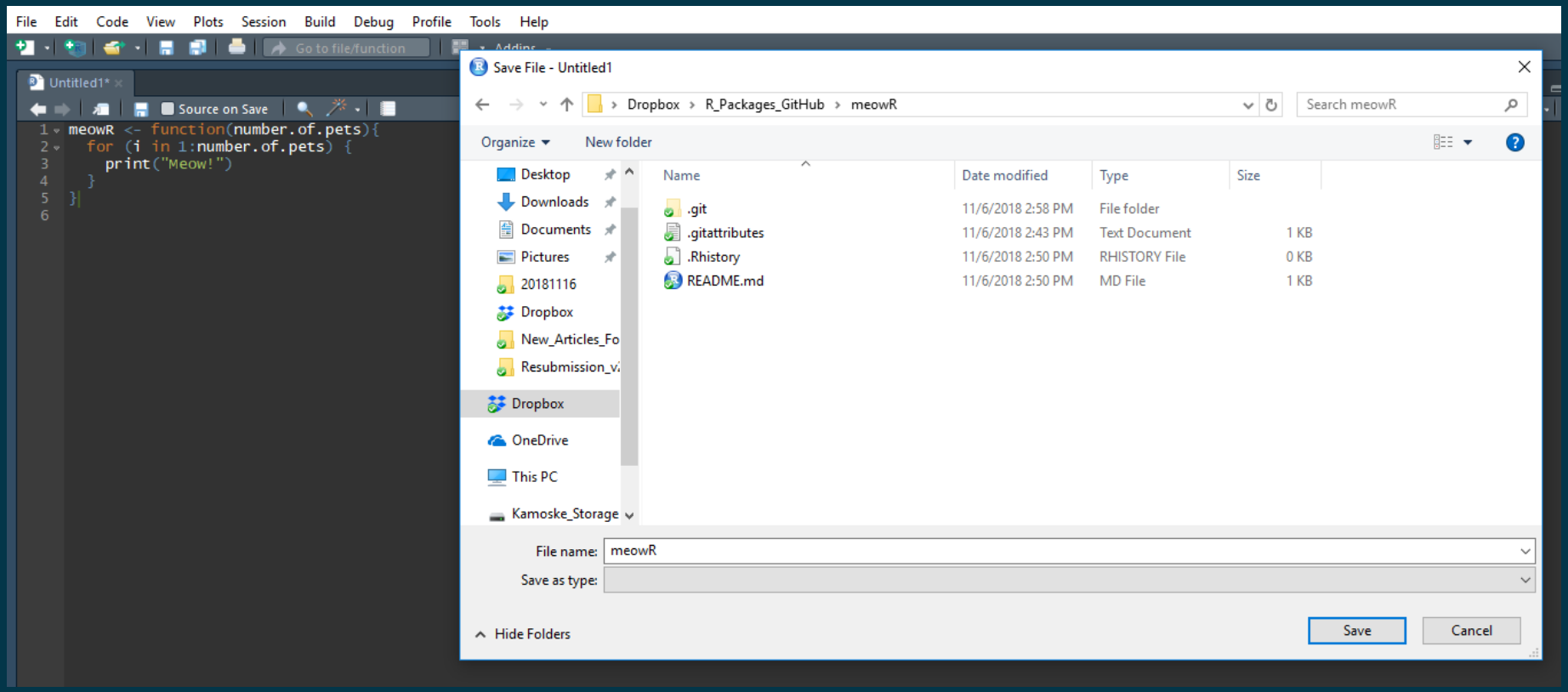
R_SMAP
R code for processing SMAP data
R Updated on Sep 18, 2017

Now our project is available to world!

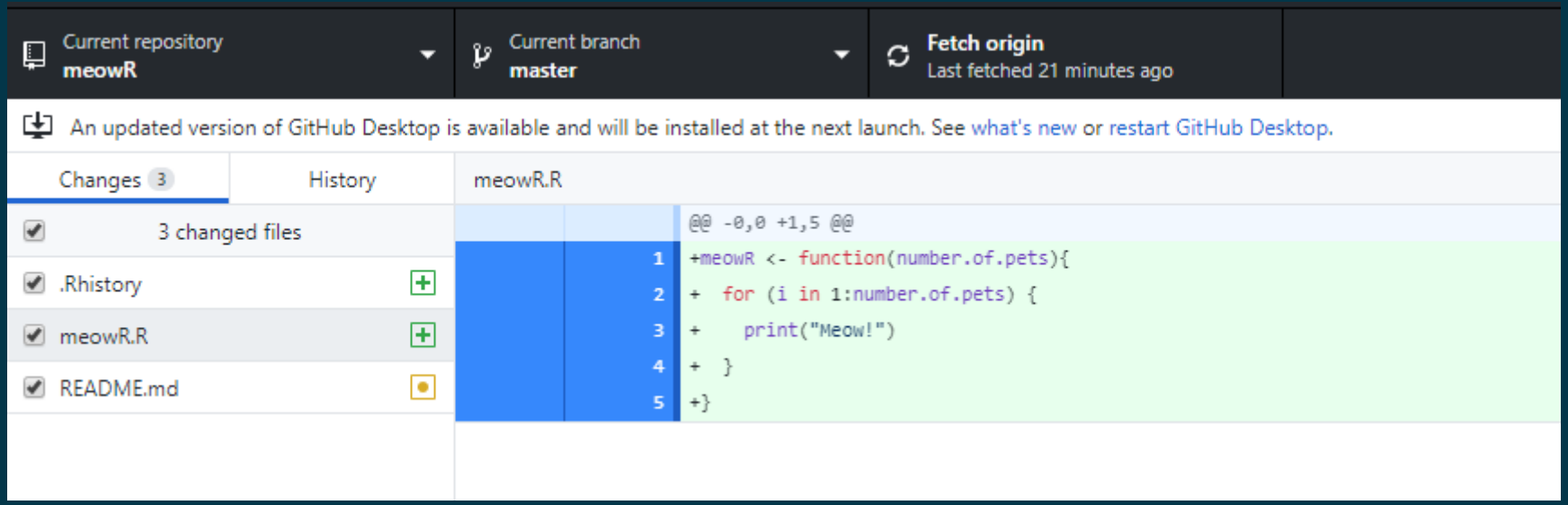
GitHub Desktop Quick Demo

Now that our project is in a GitHub repo it is really easy to populate it with code or anything else you may want to store on GitHub.

Let's write a quick function that all cats will appreciate and save it in our local repo folder!

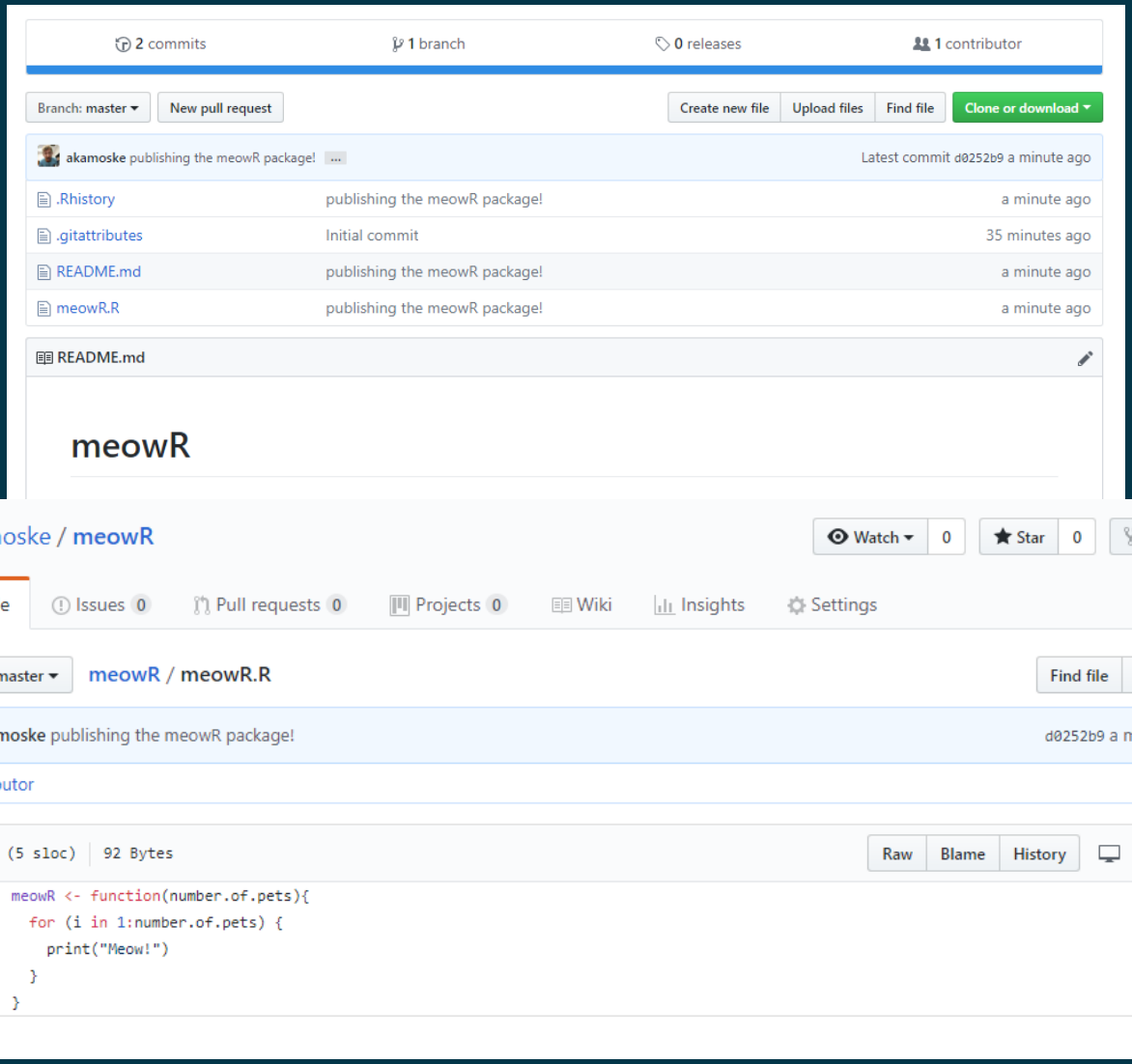


GitHub Desktop Quick Demo



Now we can push our function to our repo!

GitHub Desktop Quick Demo

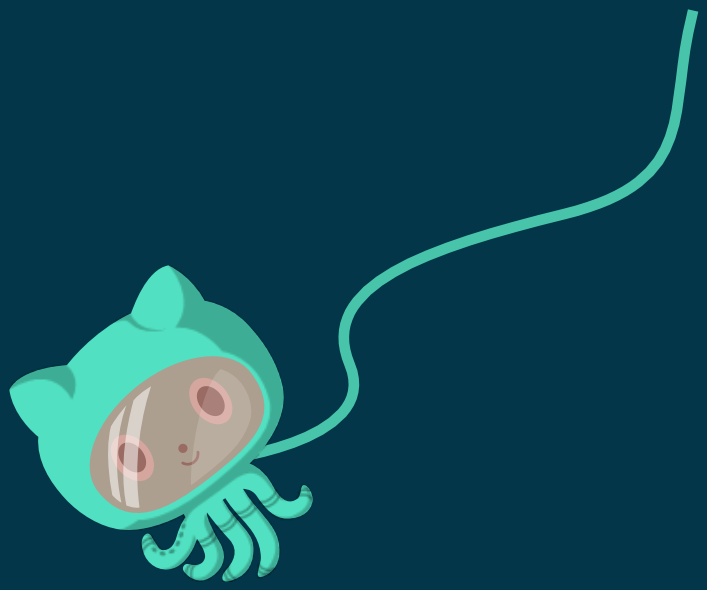


meowR is alive!!!

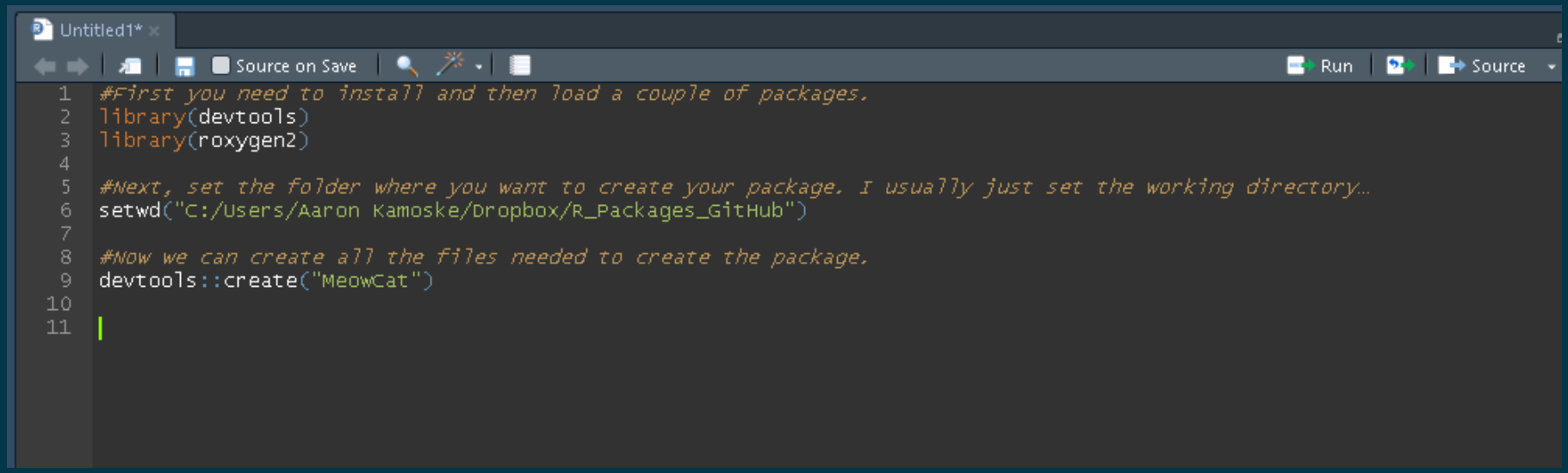
Creating R Packages With GitHub

Its pretty cool that we can host our code on GitHub, but what if we could create a package that we could load into R – just like any other package that we use in R.

R and GitHub make it really easy to do just this!

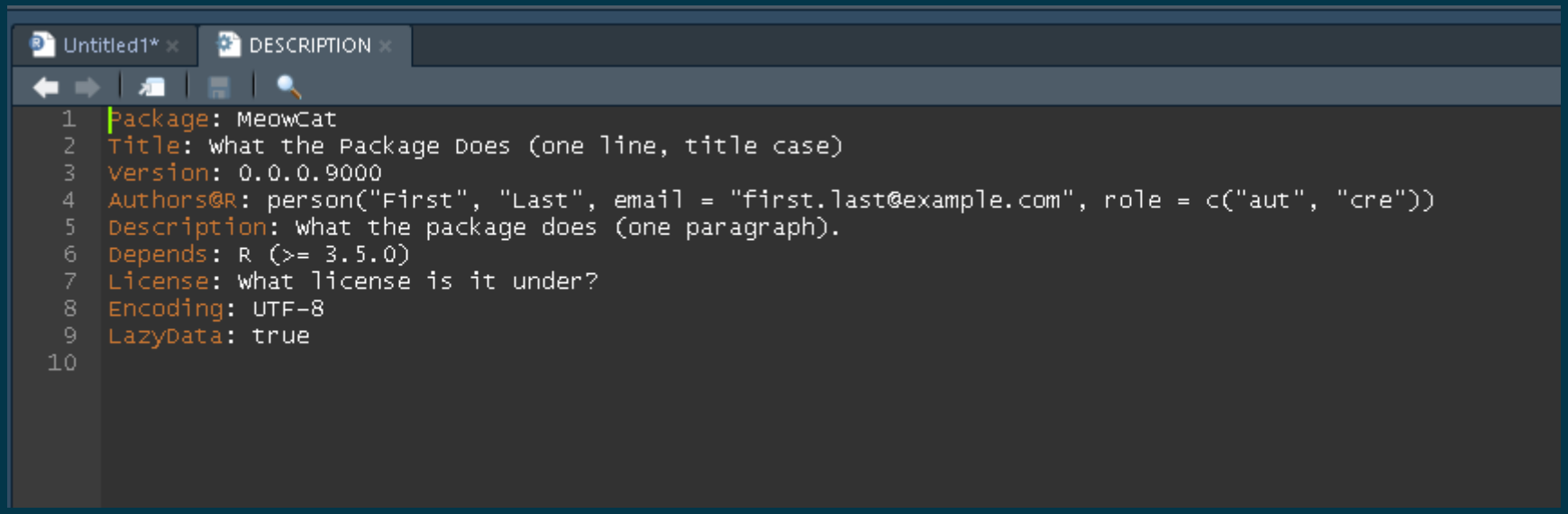


Creating R Packages With GitHub



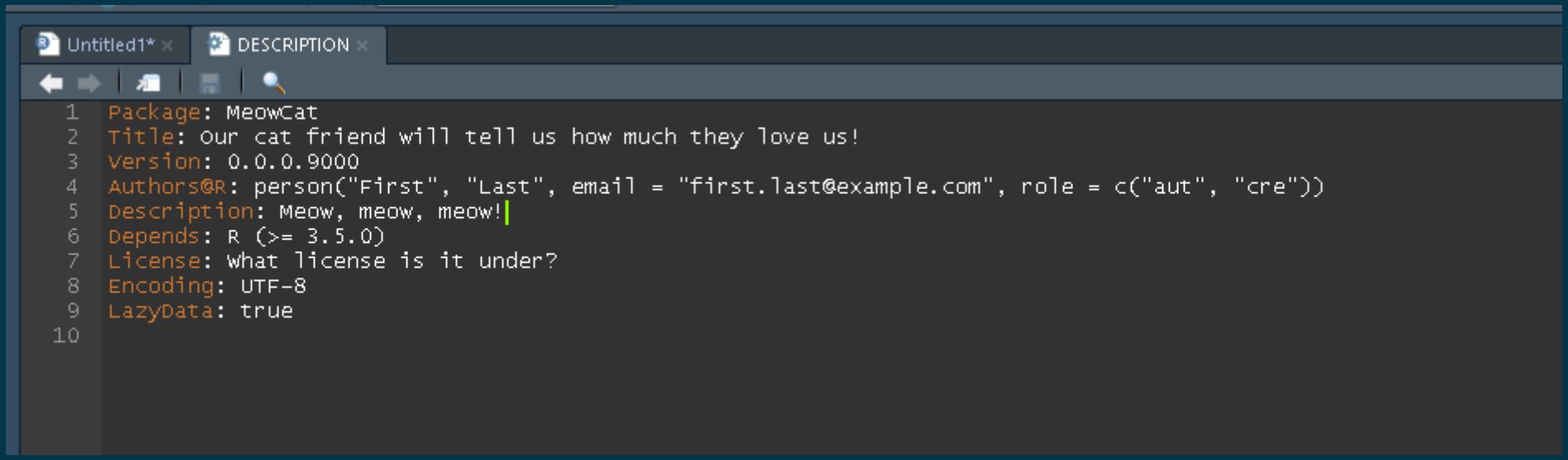
First we need to generate all the files that are necessary to host a package on GitHub. All the files that you need should now be in your folder. Open the DESCRIPTION file in RStudio so that you can edit the appropriate lines.

Creating R Packages With GitHub



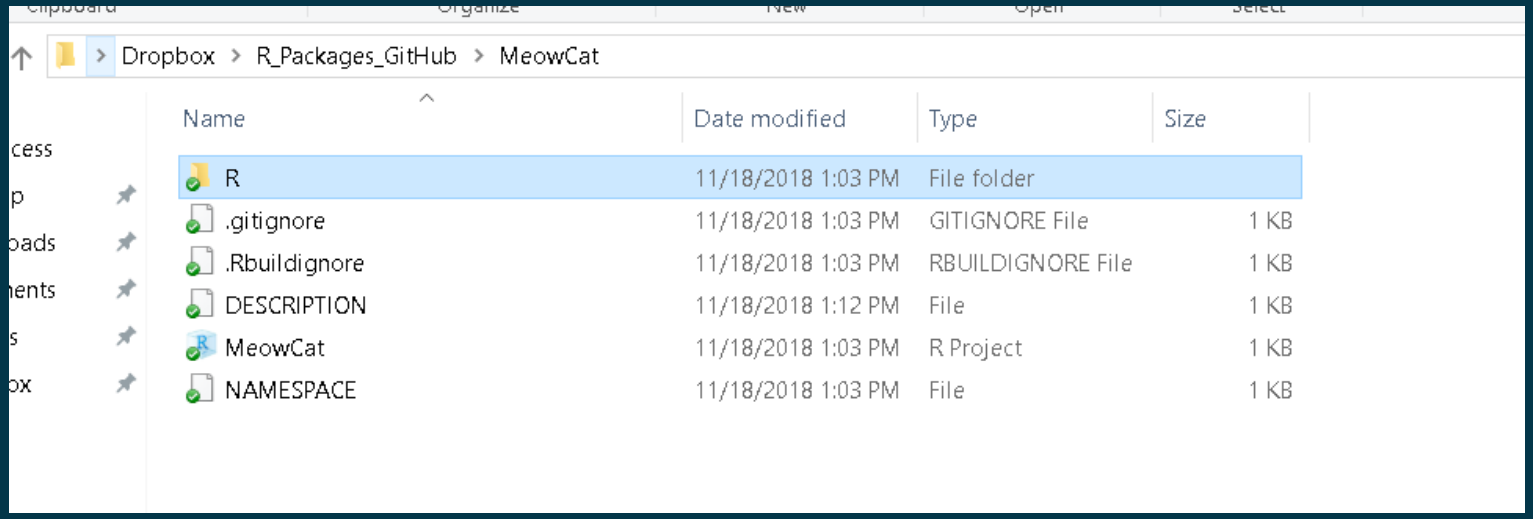
It should look like this – we can update this file with our information. GitHub and R will use this information to create our package.

Creating R Packages With GitHub



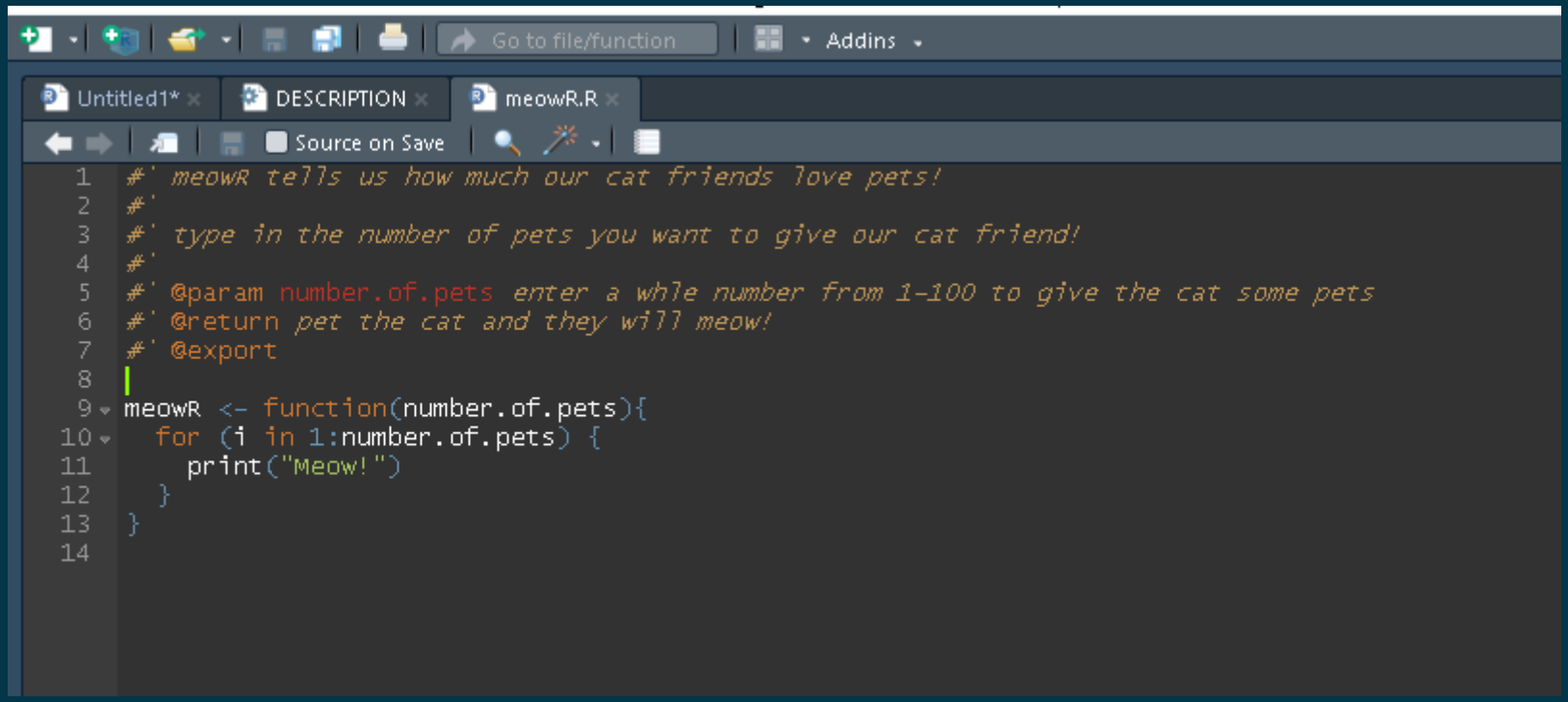
We can update this as needed!

Creating R Packages With GitHub



Now that the DESCRIPTION file is set, we can load our R functions (as individual R scripts) into the R folder that was created by the above line of code. Next, open the R script because we need to add a couple of lines of code to the top so that there will be help documentation when the package is downloaded.

Creating R Packages With GitHub



We can just reuse the function that we created in the previous example – but we need to add a couple lines of code to it.

Creating R Packages With GitHub

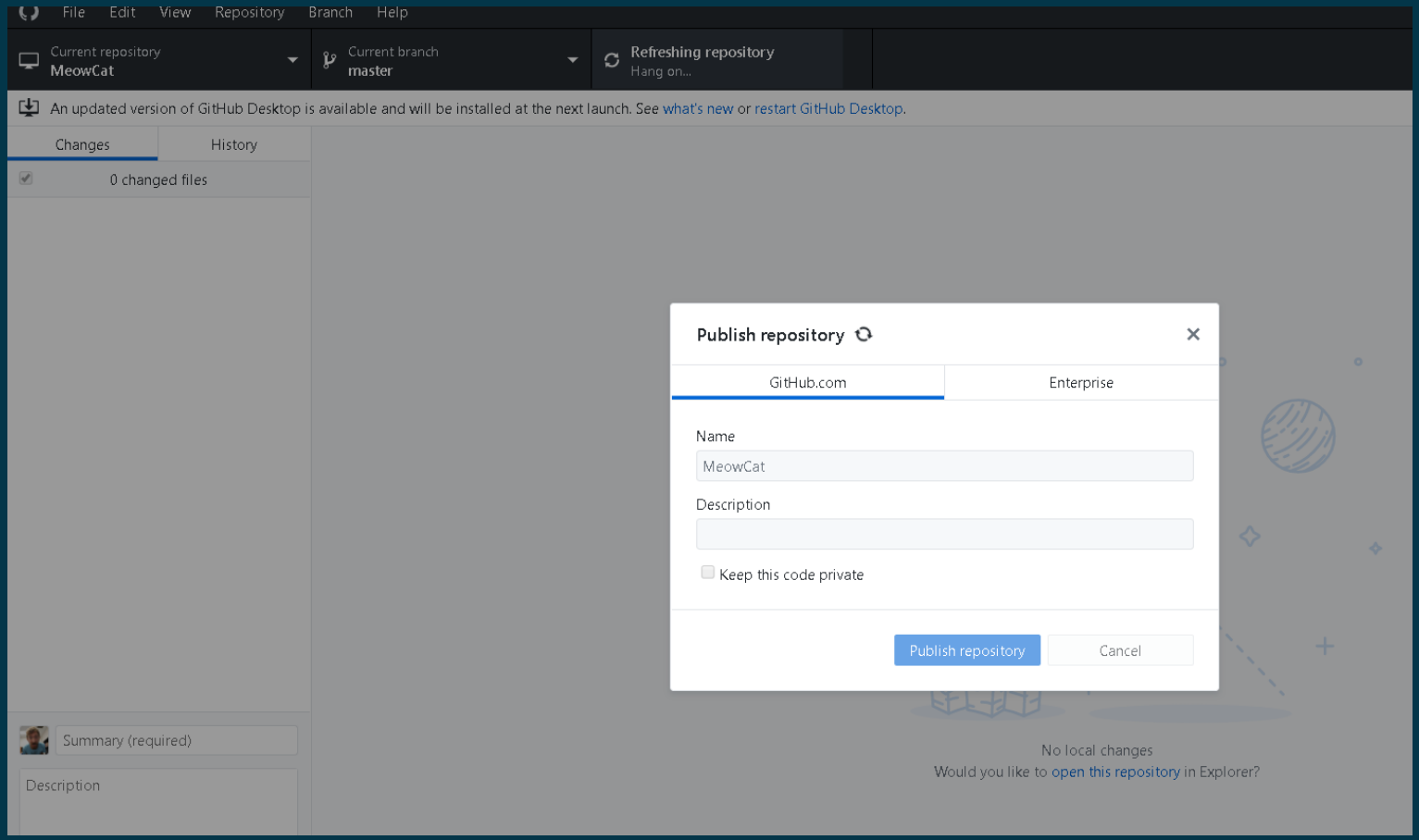
Untitled1* | meowR.R

Source on Save | Run | Source

```
1  #First you need to install and then load a couple of packages.
2  library(devtools)
3  library(roxygen2)
4
5  #Next, set the folder where you want to create your package. I usually just set the working directory…
6  setwd("C:/Users/Aaron Kamoske/Dropbox/R_Packages_GitHub")
7
8  #Now we can create all the files needed to create the package.
9  devtools::create("MeowCat")
10
11 #set the working directory to our new R package folder
12 setwd("./MeowCat")
13
14 #create a markdown document that can be used as our help information for our functions
15 devtools::document()
16
```

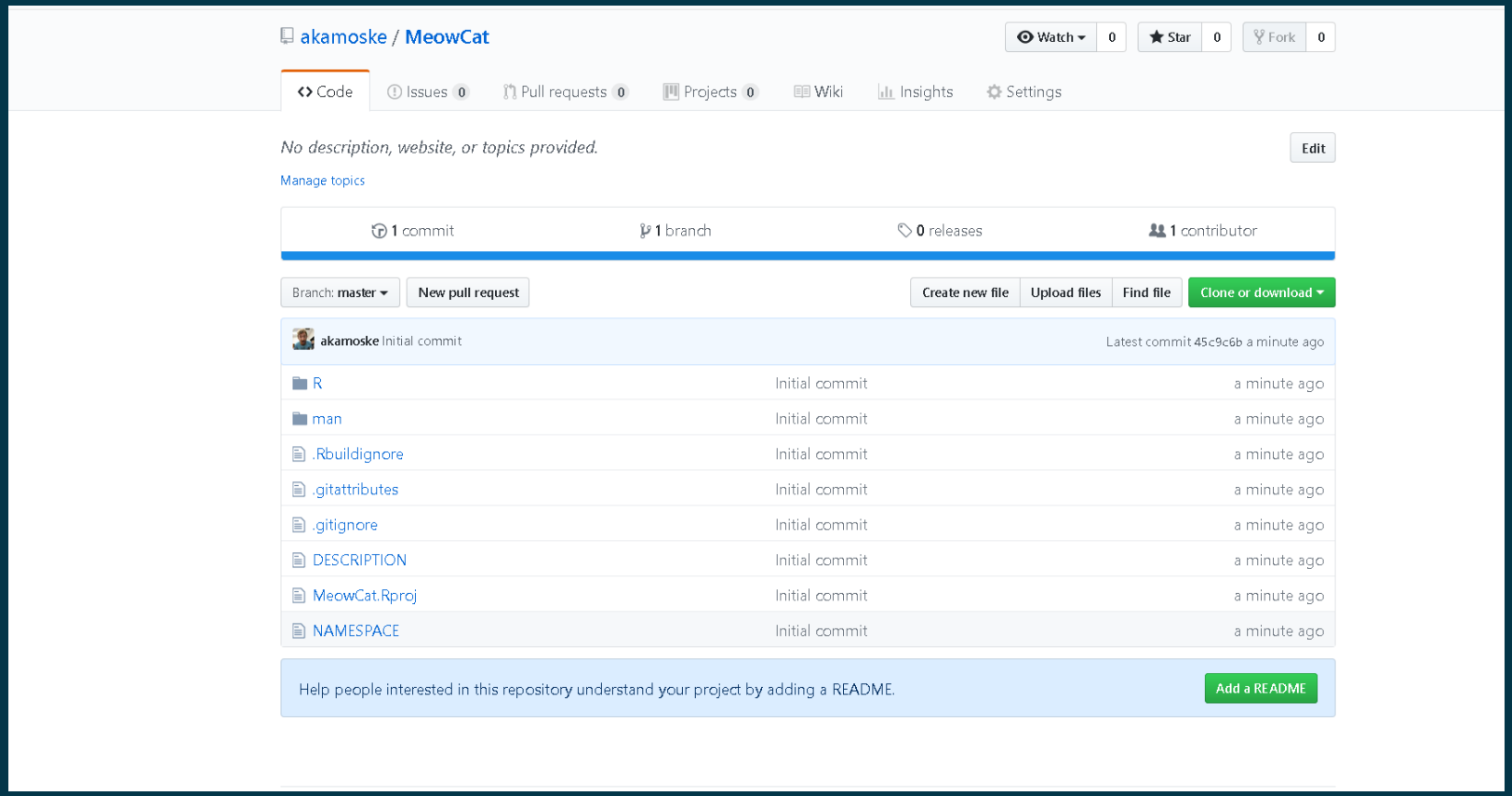
Now that these lines are added to the top of the function there are just a couple more steps to complete the process.

Creating R Packages With GitHub



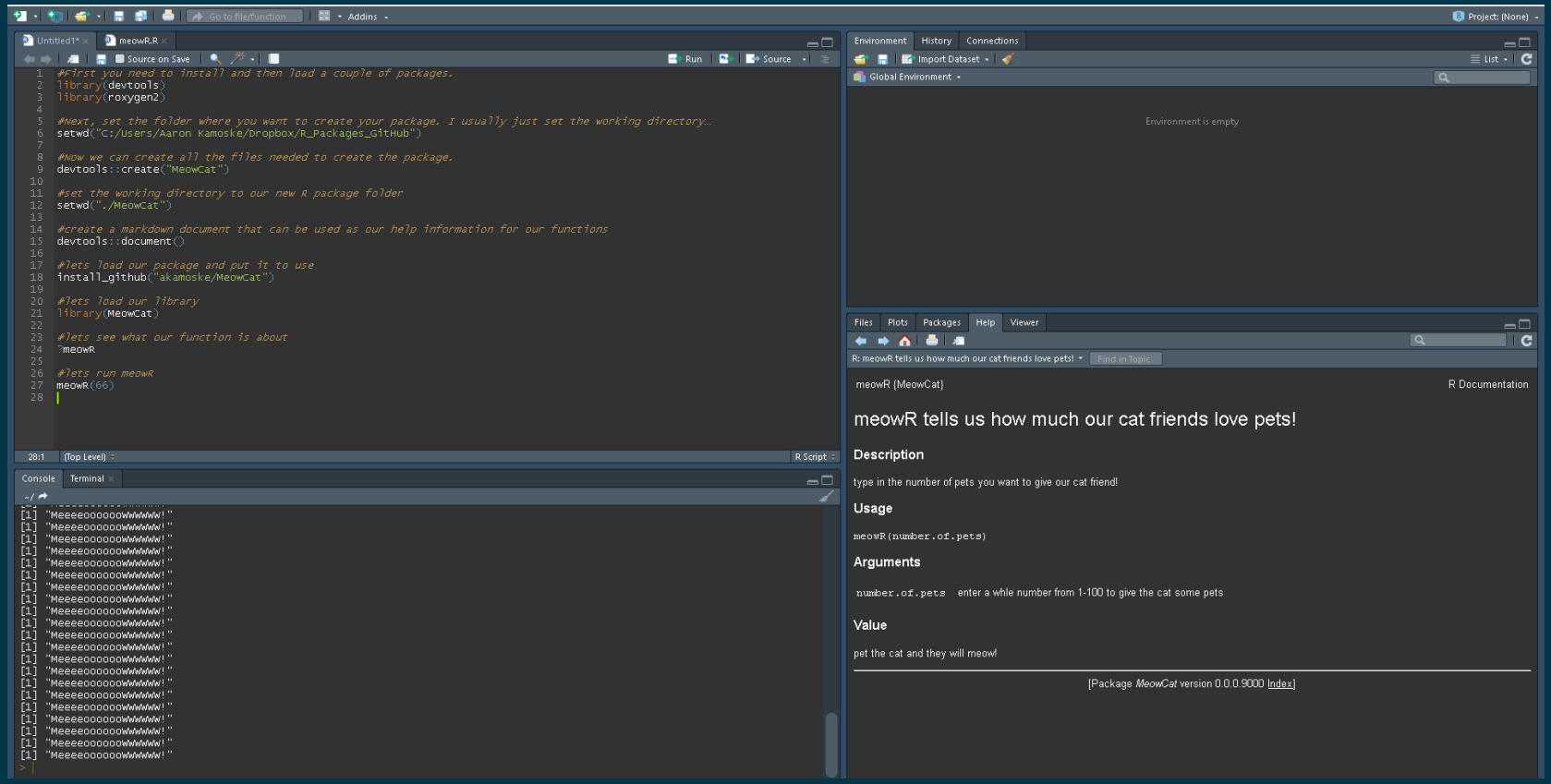
Now lets push our new package to GitHub using the same method we discussed previously.

Creating R Packages With GitHub



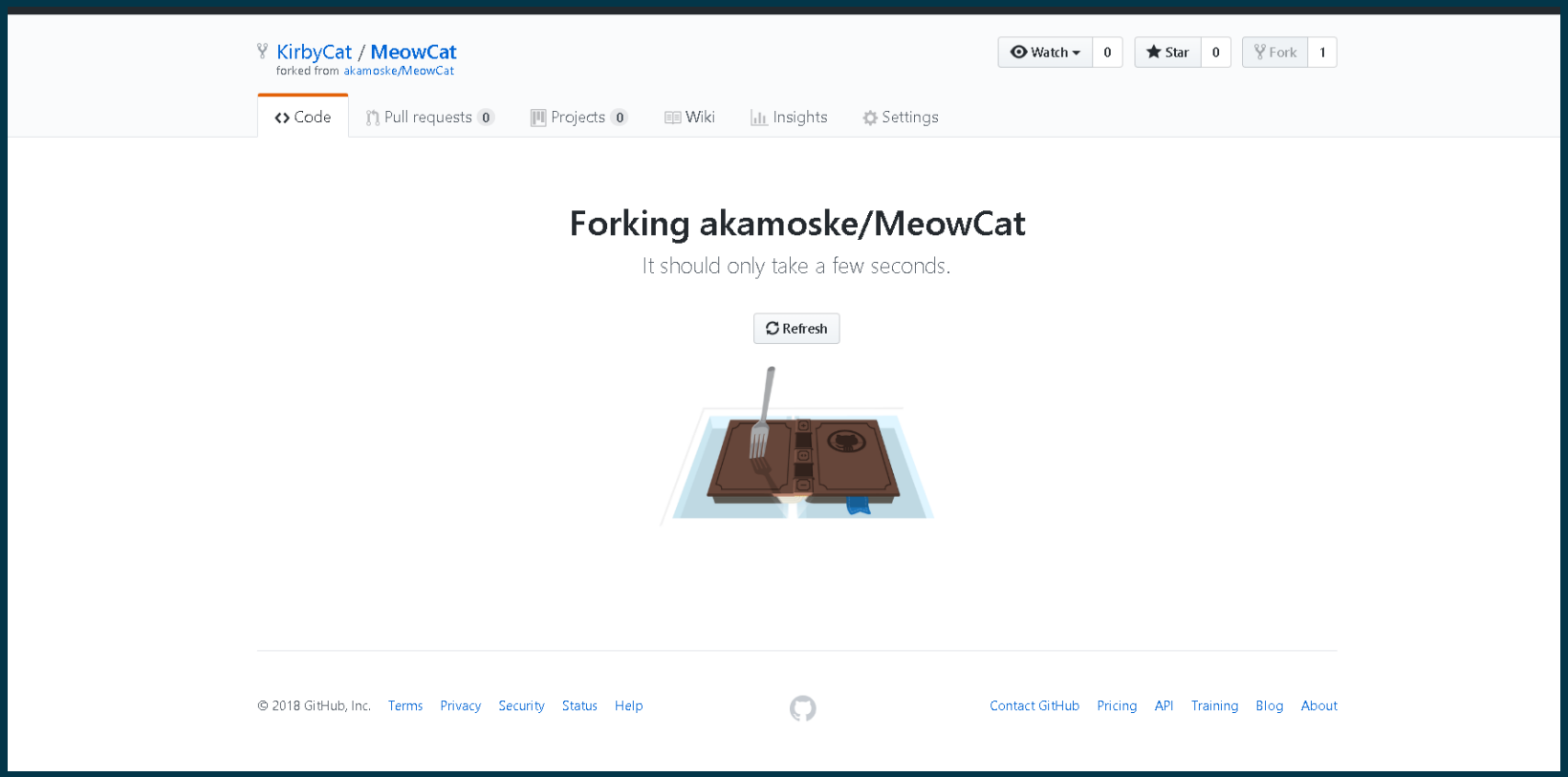
Its alive!

Creating R Packages With GitHub



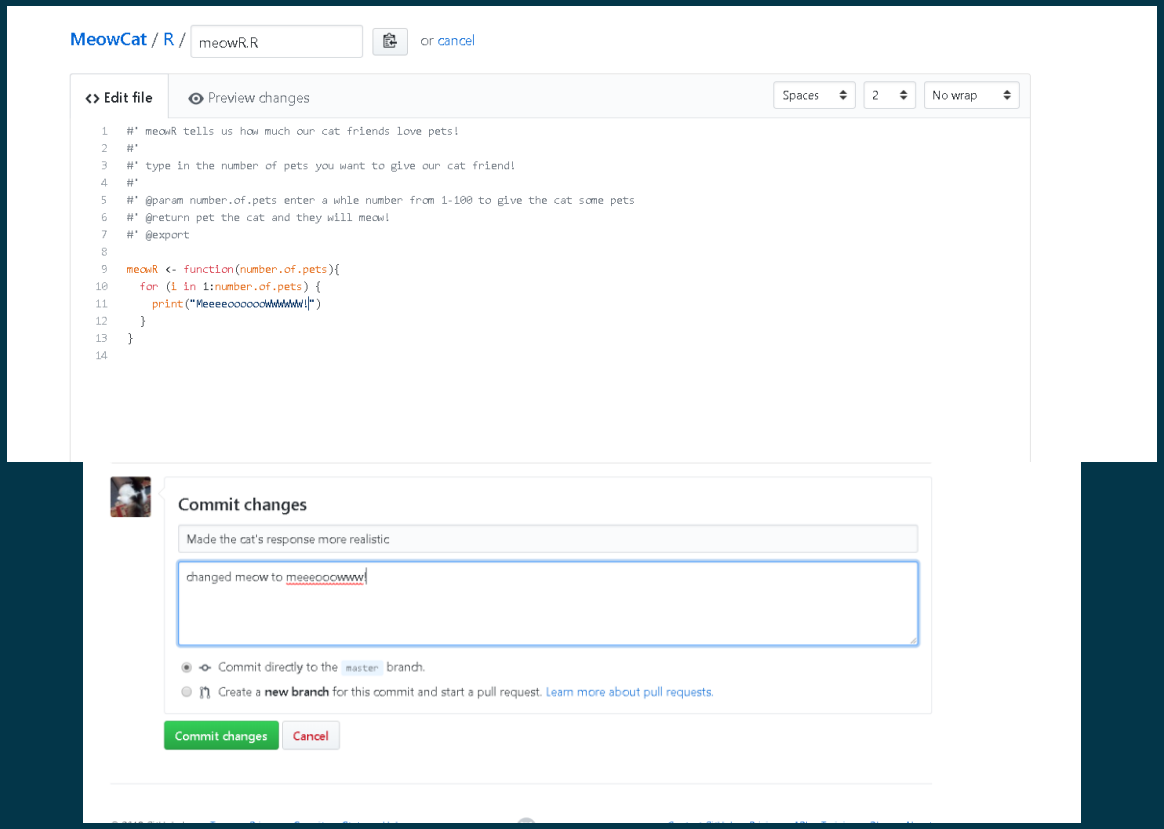
Lets load our package and put it to use!

GitHub as a Collaborative Tool



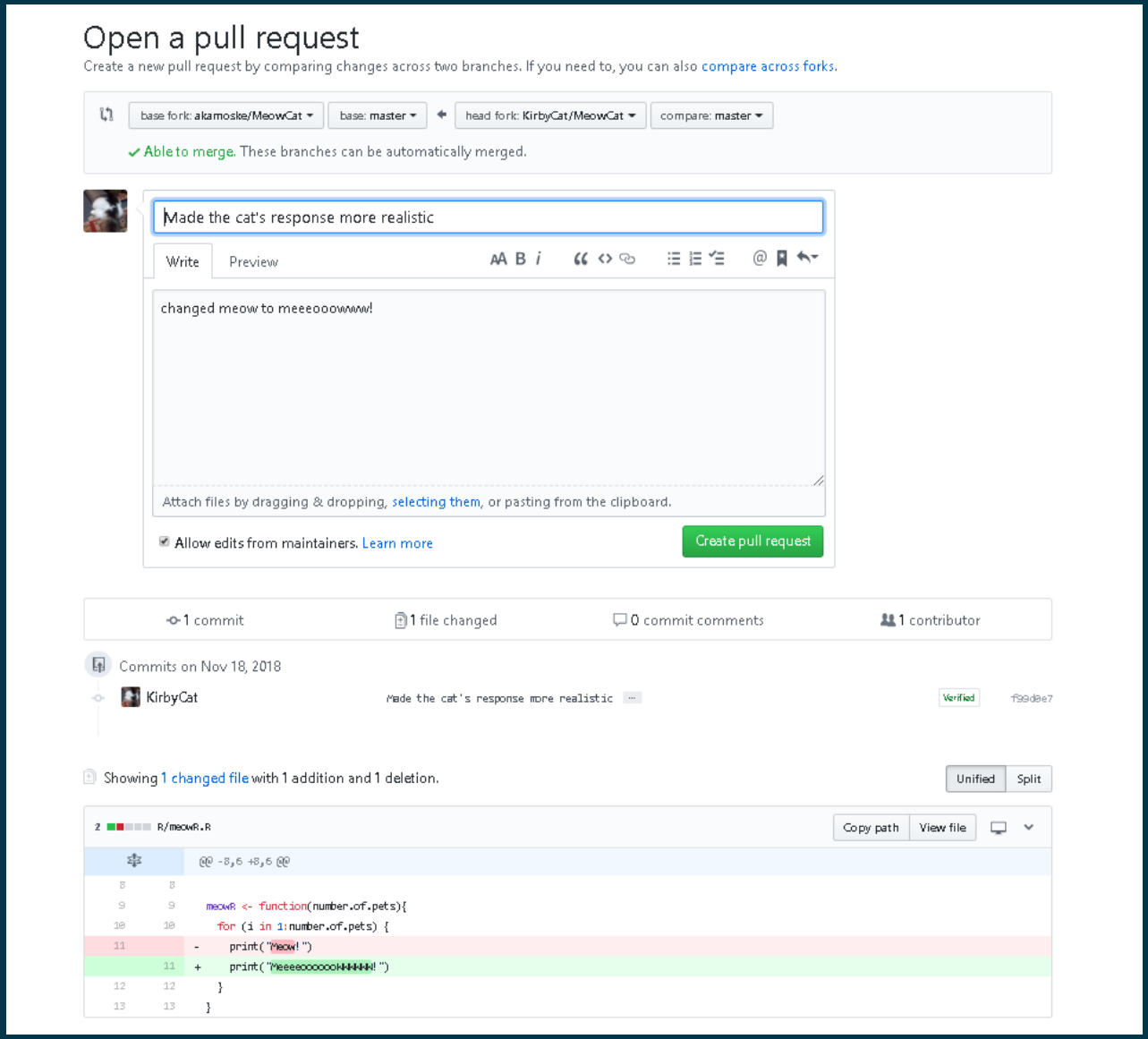
One of the best parts about GitHub is how you can use it for collaborative research. This is done via forking and pull requests! It looks like someone is forking our new package!

GitHub as a Collaborative Tool



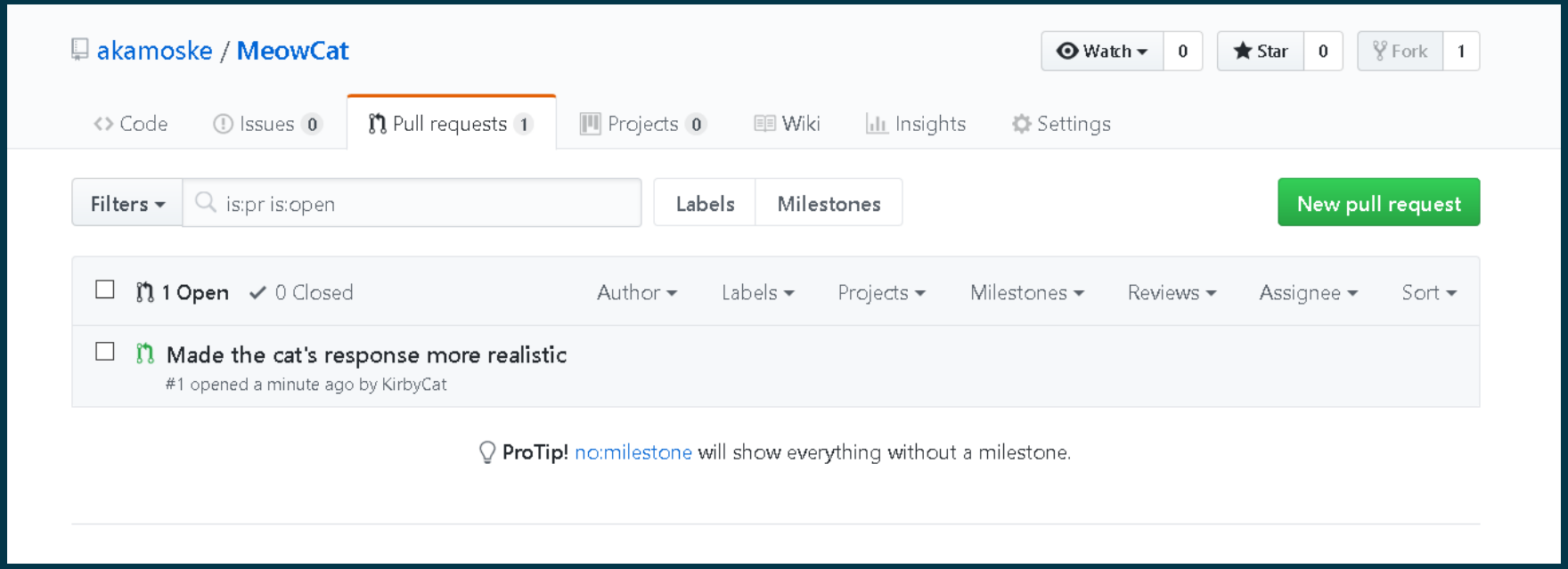
Once you fork someone's code you can make changes as needed and then alert the original coder to these changes. Looks like KirbyCat is making some changes.

GitHub as a Collaborative Tool



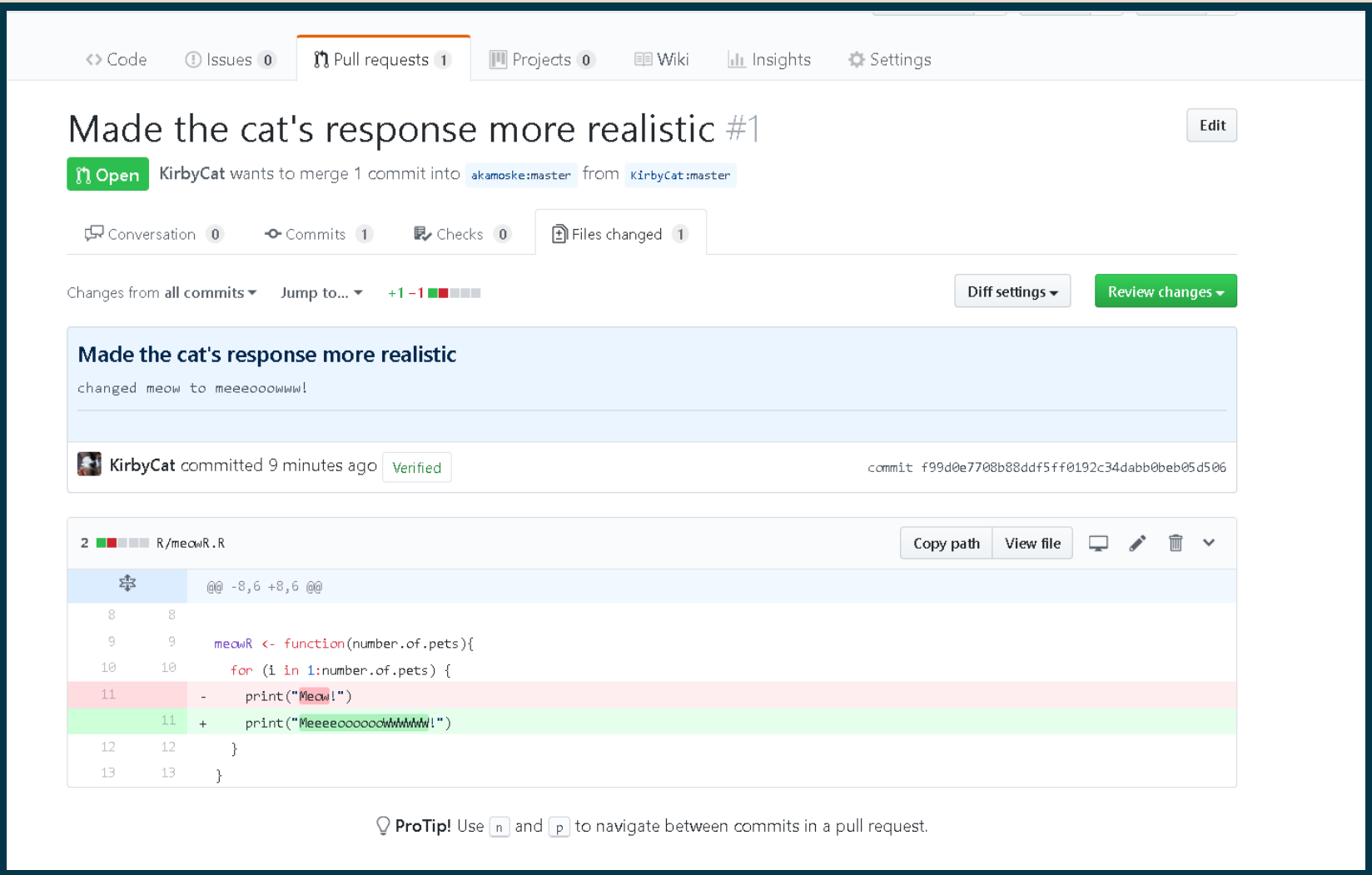
Now KirbyCat would make a pull request to let us know about their changes.

GitHub as a Collaborative Tool



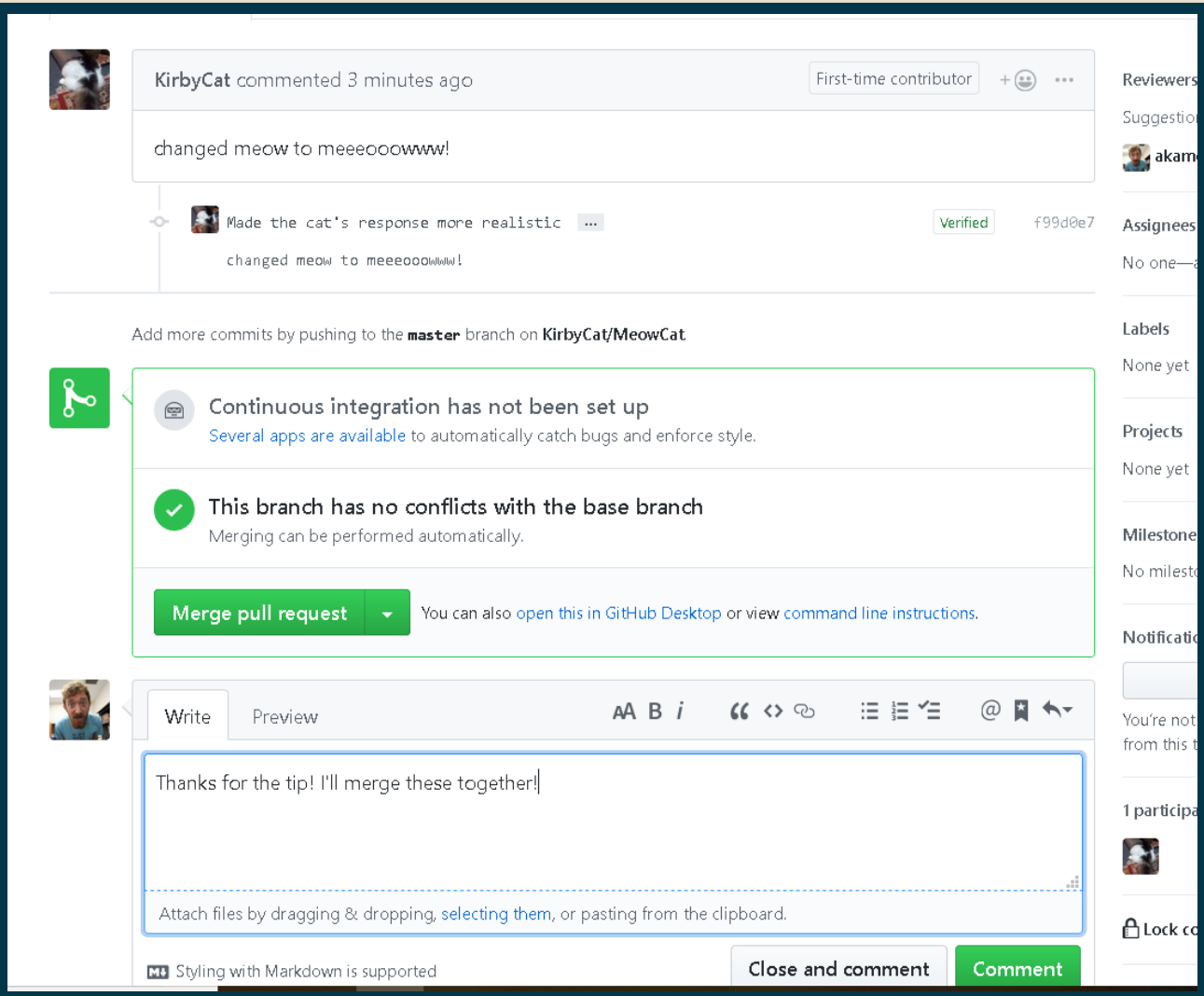
Now we can see the pull request that KirbyCat made and see if we want to implement it.

GitHub as a Collaborative Tool



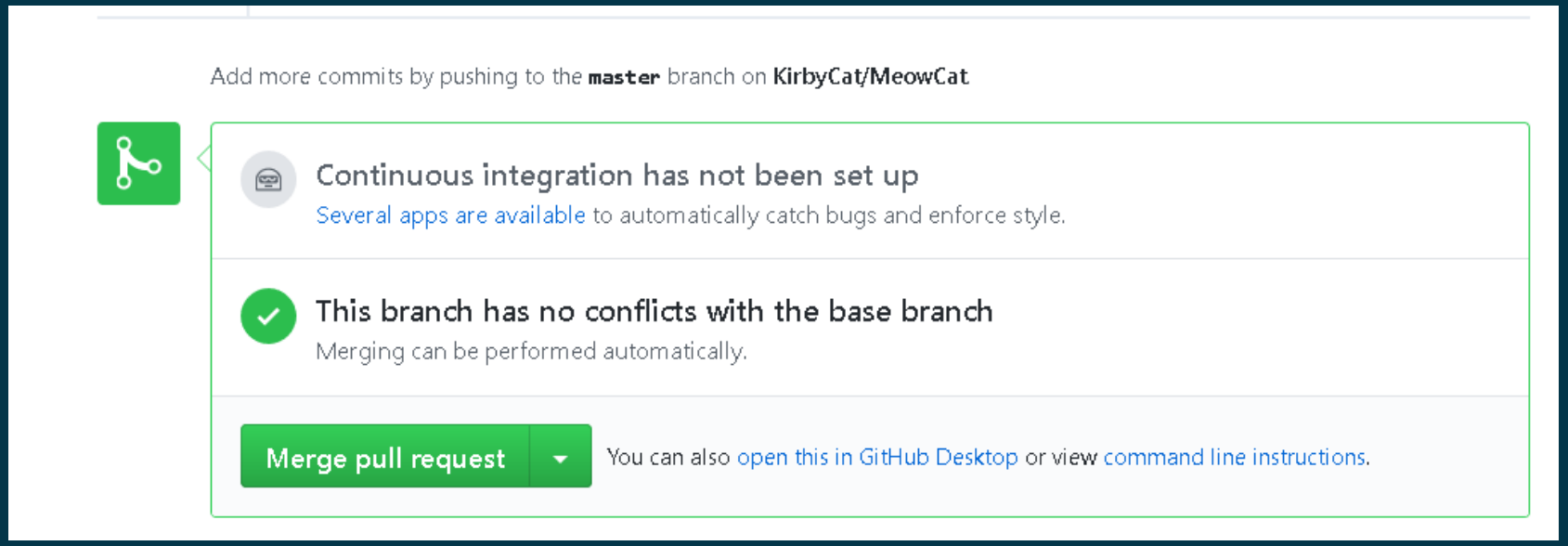
We can see these changes.

GitHub as a Collaborative Tool



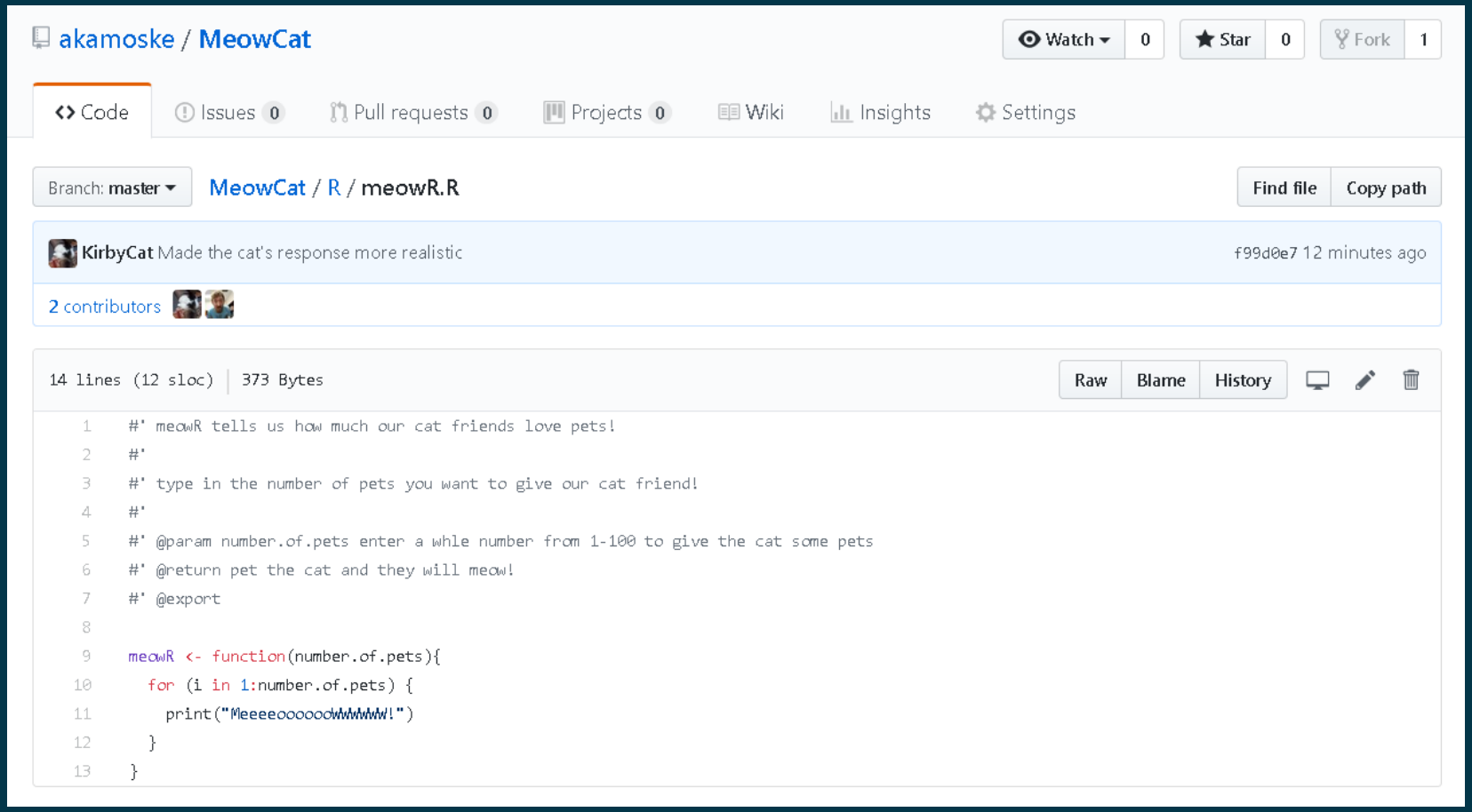
Lets leave a comment and then accept the changes!

GitHub as a Collaborative Tool



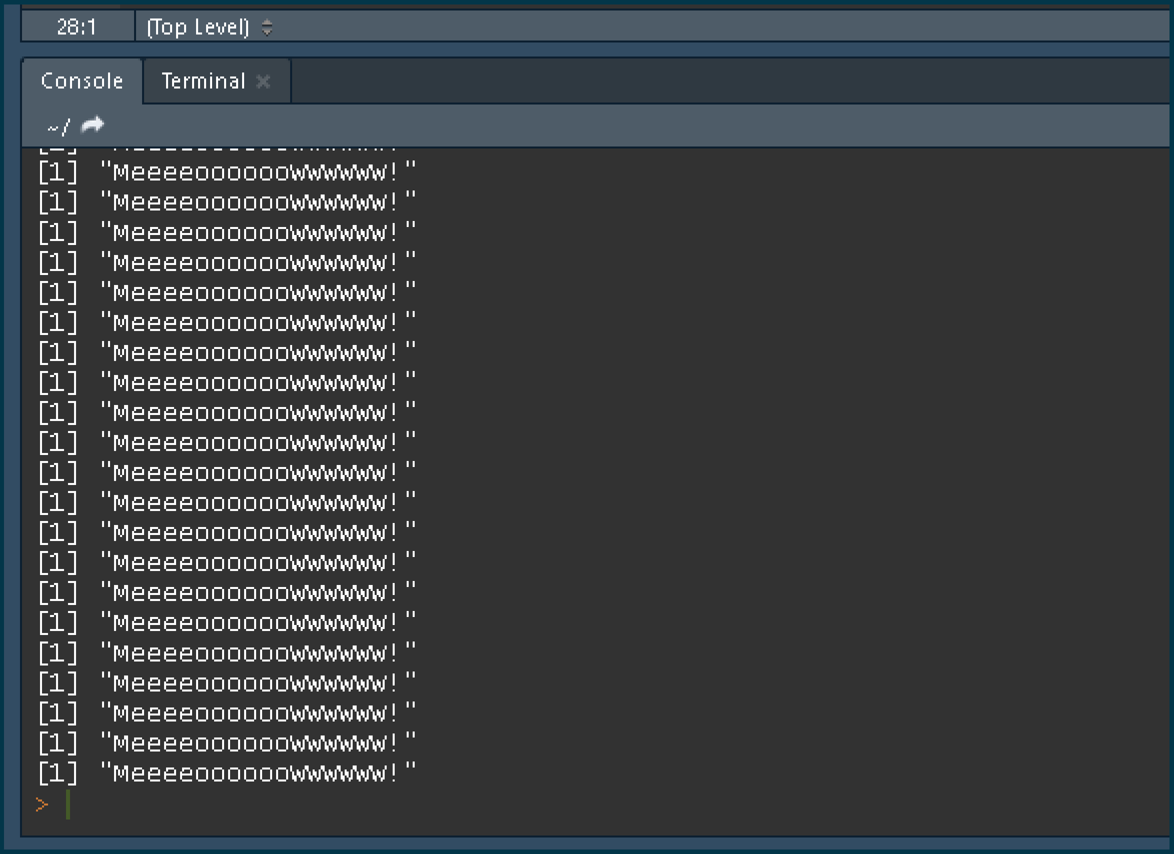
Lets leave a comment and then accept the changes!

GitHub as a Collaborative Tool



We can see the changes now!

GitHub as a Collaborative Tool



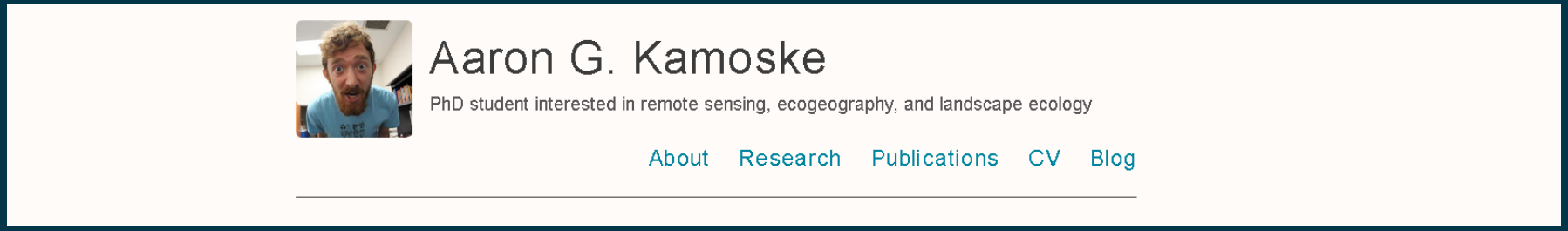
Lets reload our package in R and see how it works now!

GitHub as a Collaborative Tool

The ability to carry our a project from idea, to code, to package, via a collaborative framework, is what makes R and GitHub so unique and useful!

GitHub Pages

GitHub pages is a great way to make a website for free that is hosted on GitHub. These aren't fancy but get the job done and are free! Best of us, there are many themes that use R markdown to create pages!



Helpful links!

Pull requests: https://www.youtube.com/watch?v=_NrSWLQsDL4

Creating R packages: https://hilaryparker.com/2014/04/29/writing-an-r-package-from-scratch/

GitHub Desktop: https://www.youtube.com/watch?v=77W2JSL7-r8

Simple GitHub Page Theme: https://github.com/barryclark/jekyll-now

Official GitHub Help: https://help.github.com/

GitHub Pages: https://pages.github.com/

Jekyll Themes for GitHub Pages: https://jekyllrb.com/

GitHub Pages Tutorial: https://guides.github.com/features/pages/

GitHub and Jekyll Tutorial: http://jmcglone.com/guides/github-pages/

Popular GitHub Pages Supported Jekyll Themes: https://github.com/jekyll/jekyll/wiki/Themes