

What is GitHub?

<https://youtu.be/w3jLJU7DT5E>

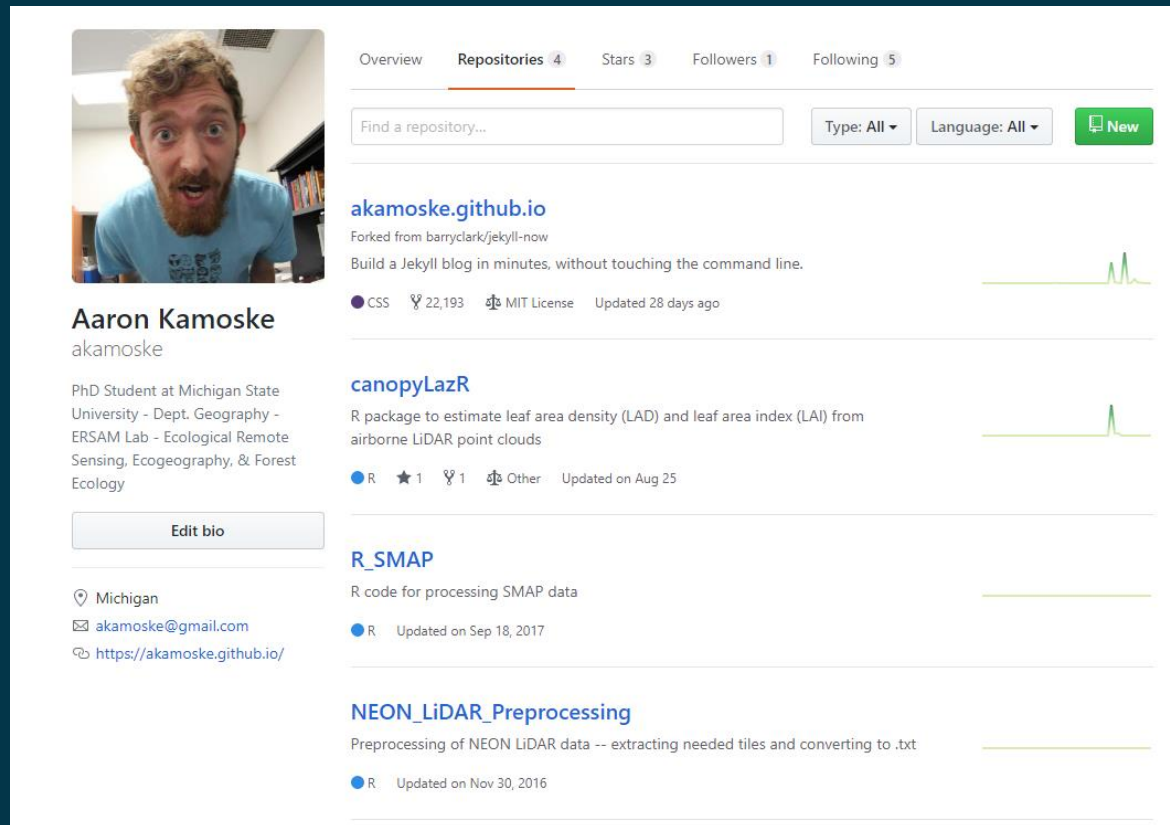
Why should we use GitHub?

- 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.



The screenshot displays the GitHub profile of Aaron Kamoske (akamoske). The profile includes a profile picture of a man with a beard and curly hair, a bio identifying him as a PhD student at Michigan State University, and his location as Michigan. The 'Repositories' tab is active, showing a list of four repositories. Each repository entry includes the repository name, a brief description, the programming language (R), star and fork counts, the license, and the last update date. A green 'New' button is visible in the top right of the repository list.

Aaron Kamoske
akamoske

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

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

Overview **Repositories 4** Stars 3 Followers 1 Following 5

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

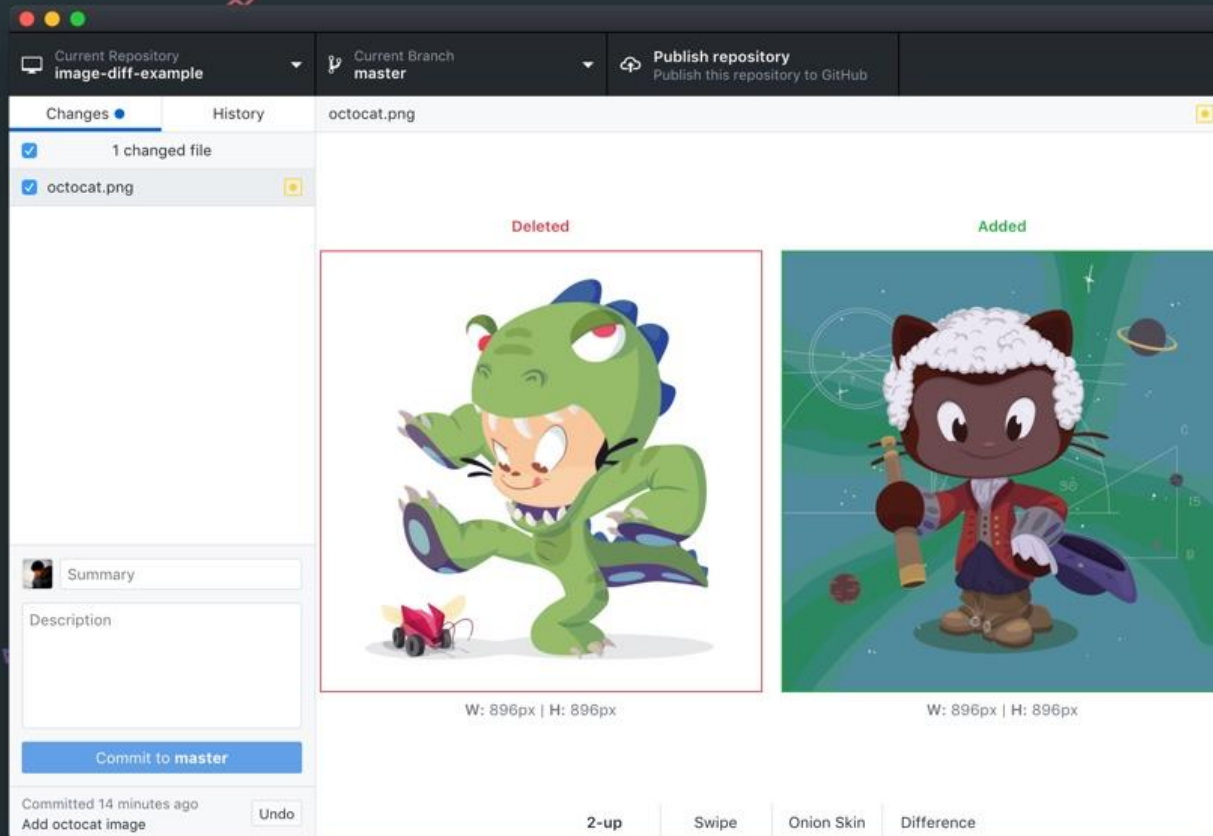
akamoske.github.io
Forked from barryclark/jekyll-now
Build a Jekyll blog in minutes, without touching the command line.
CSS 22,193 MIT License Updated 28 days ago

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

NEON_LiDAR_Preprocessing
Preprocessing of NEON LiDAR data -- extracting needed tiles and converting to .txt
R Updated on Nov 30, 2016

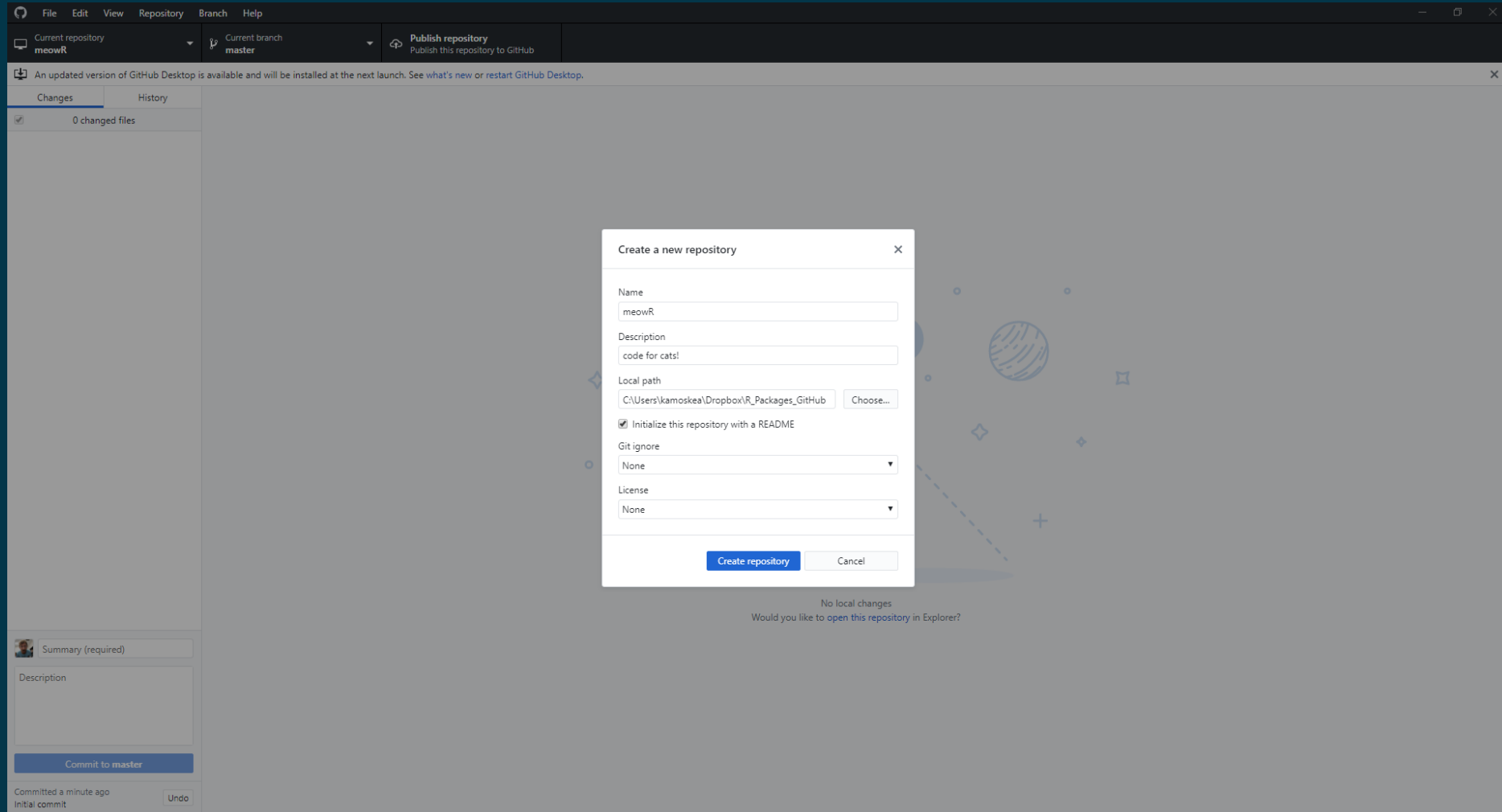
How do we access GitHub?



GitHub Desktop is awesome!

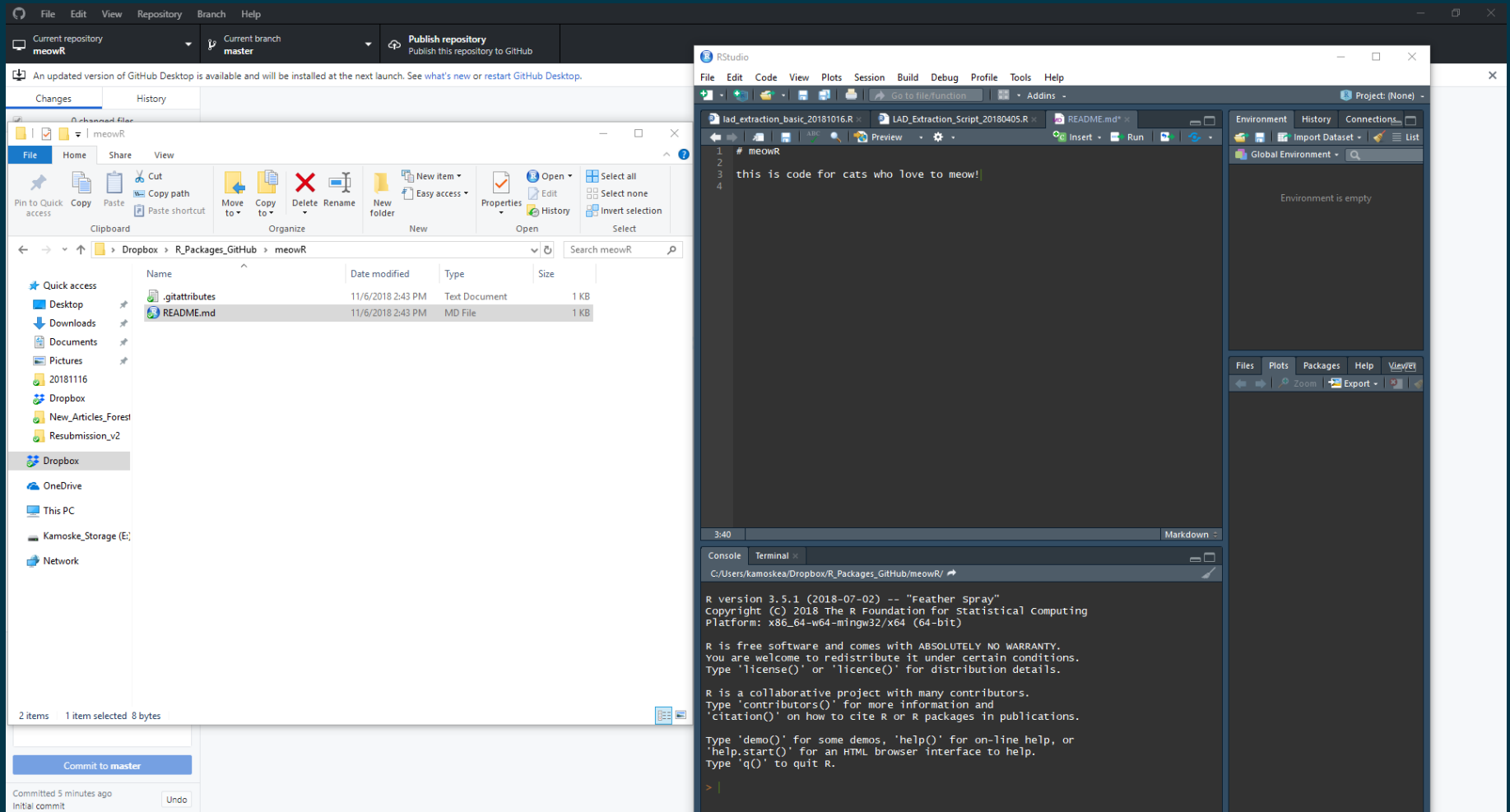
<https://desktop.github.com/>

GitHub Desktop Quick Demo



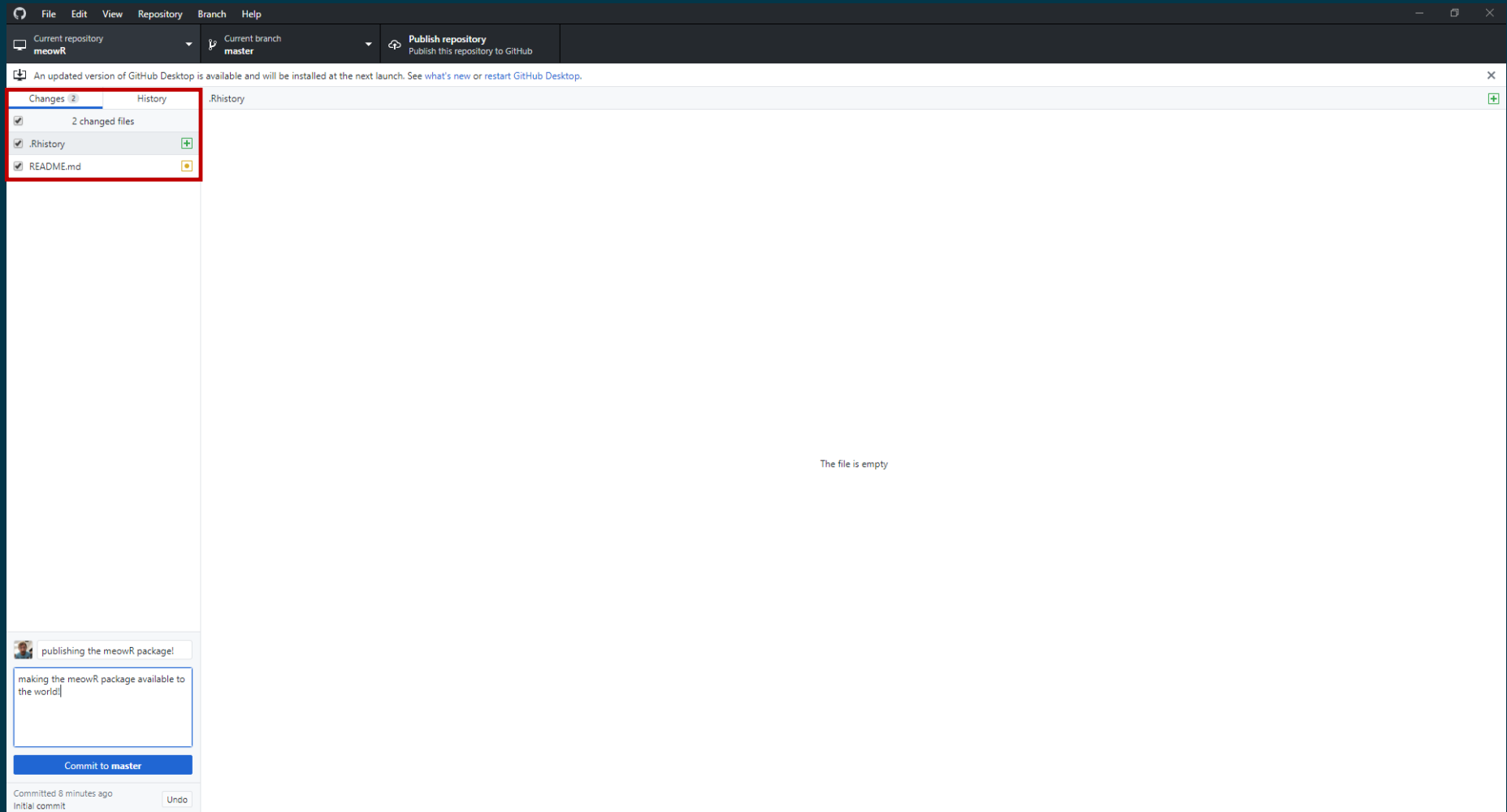
First, let's create a new repository in our GitHub folder!

GitHub Desktop Quick Demo



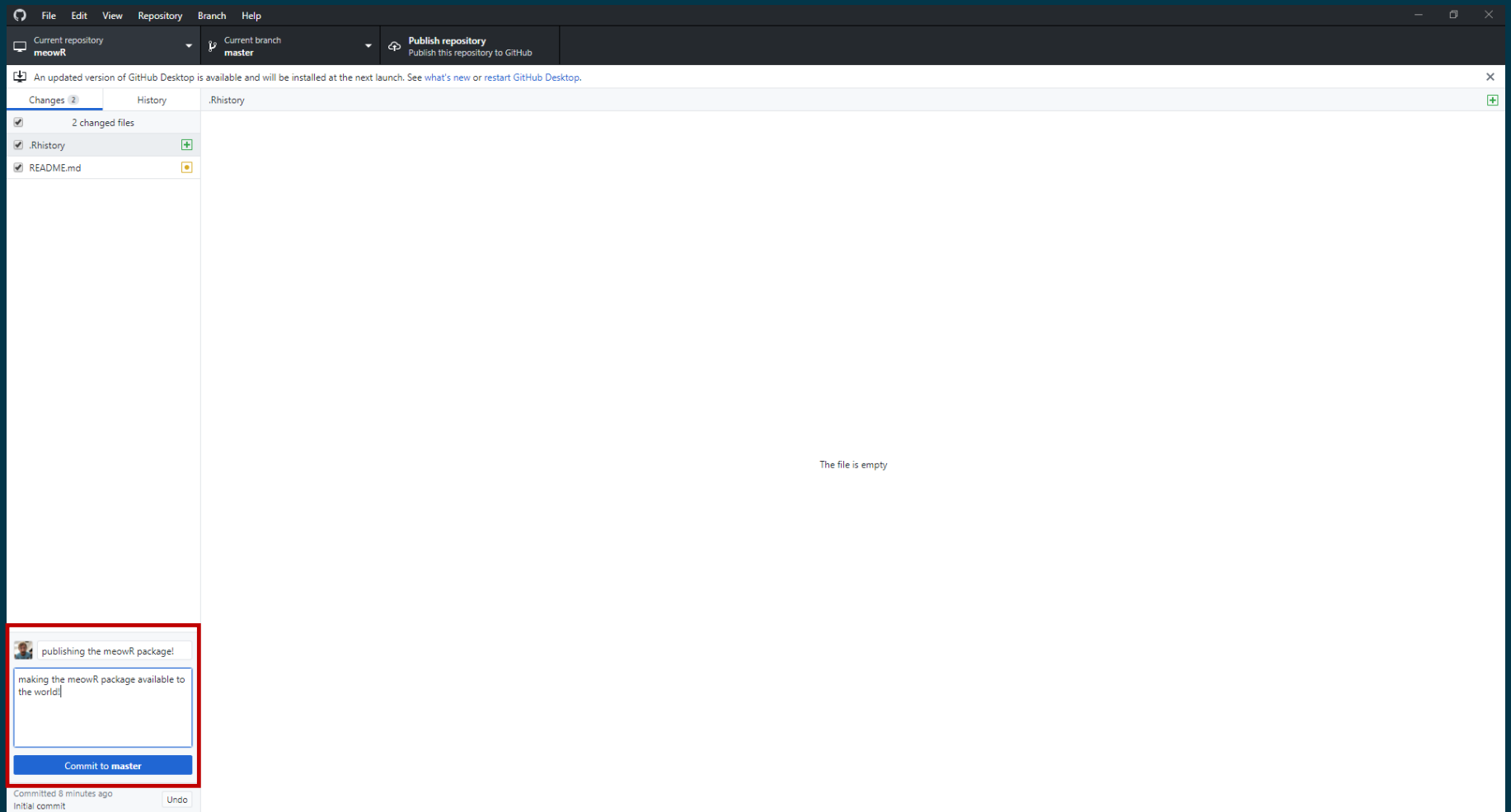
Let's open the README.md file and let the world know what our project does!

GitHub Desktop Quick Demo



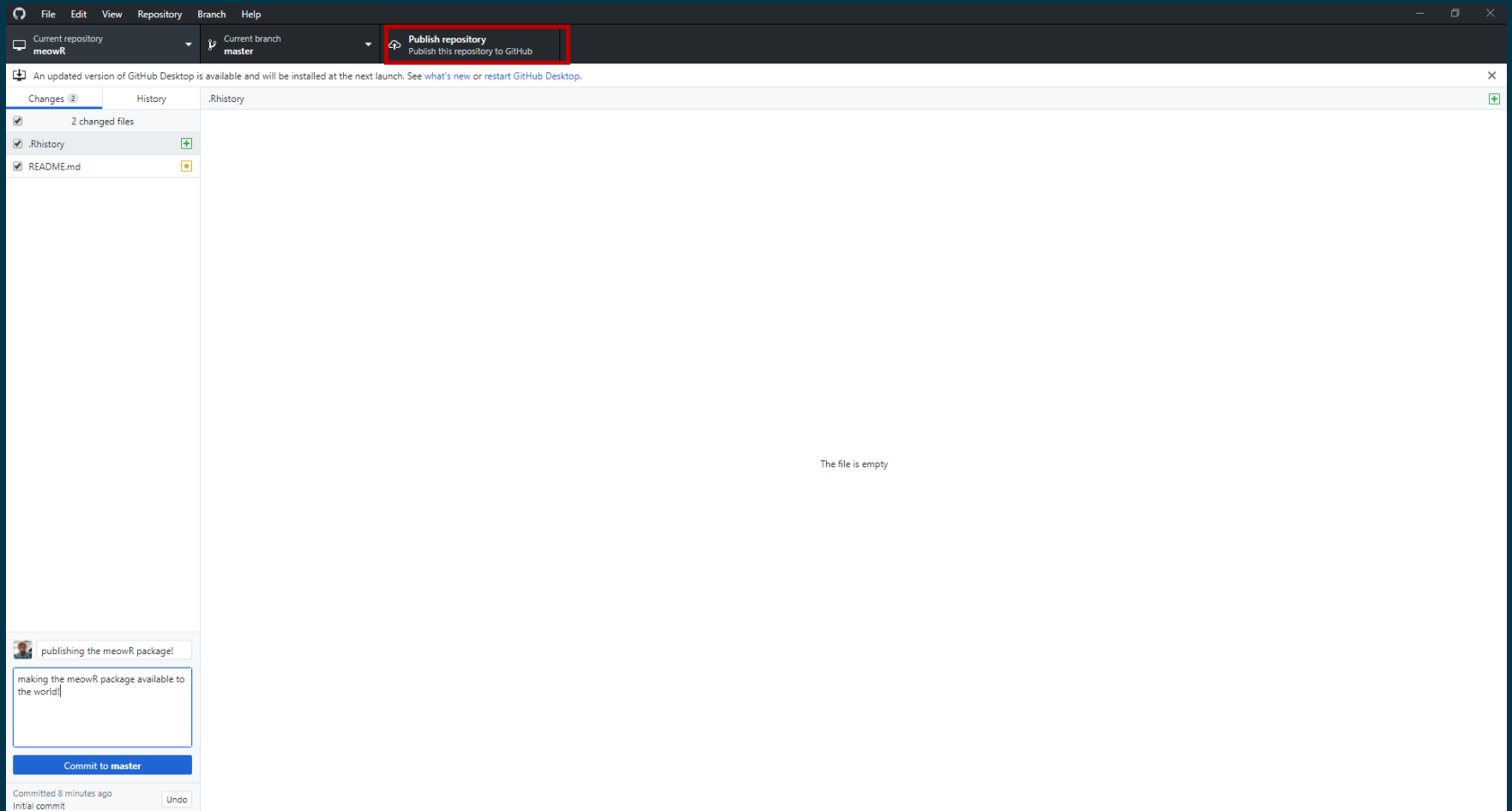
We can see the changes that we made to our project here – notice the README file.

GitHub Desktop Quick Demo



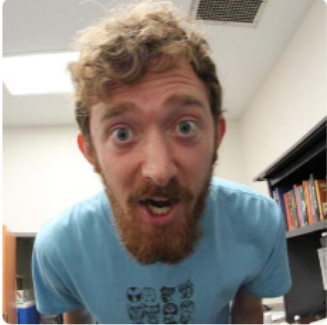
It is a good practice to write a quick commit about what we changed and why – especially for our future selves!

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

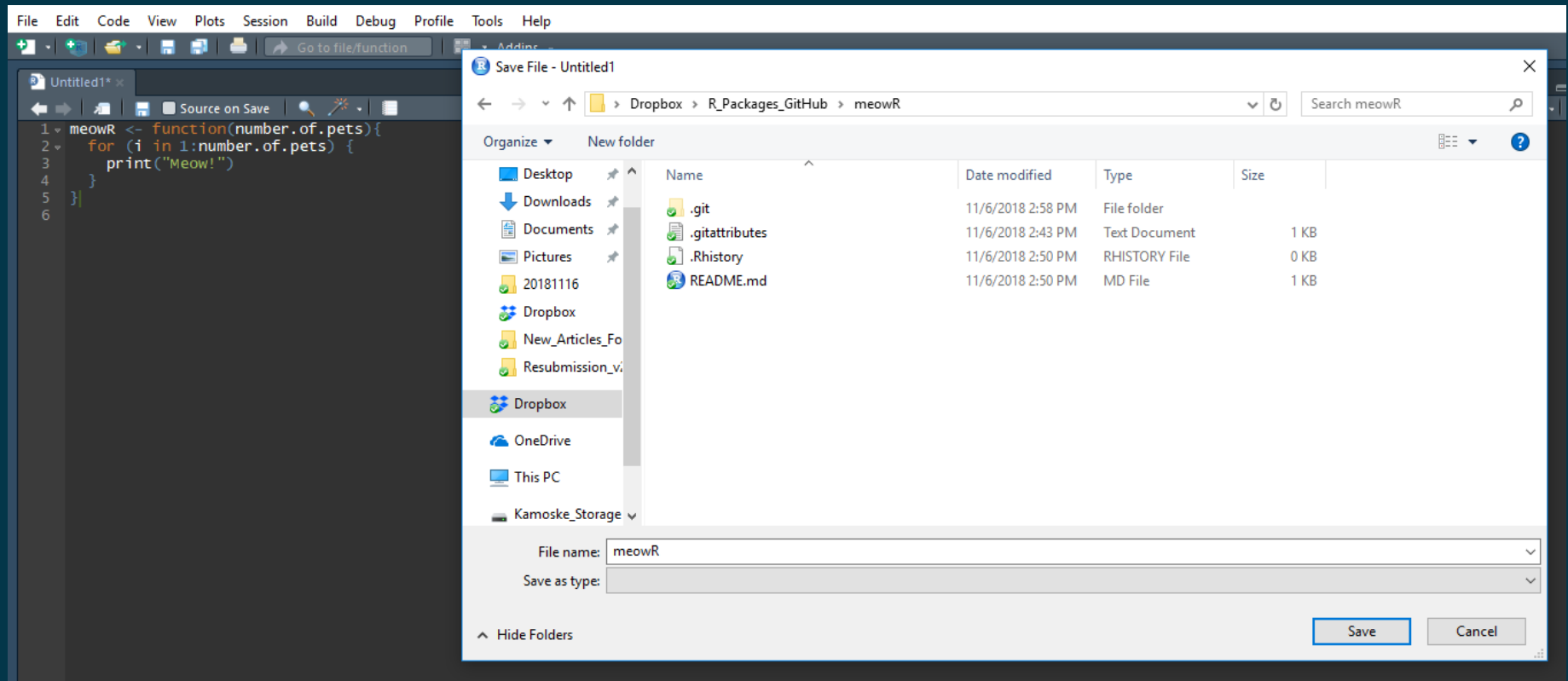
R_SMAP
R code for processing SMAP data
R Updated on Sep 18, 2017

Now our project is available to world!

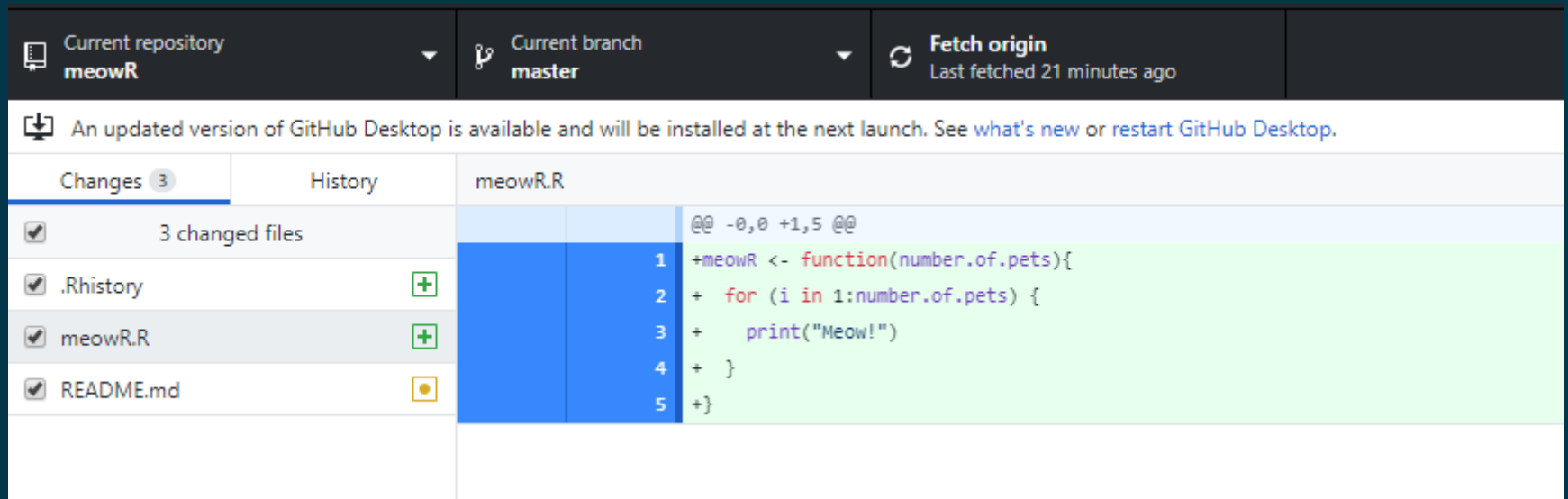
GitHub Desktop Quick Demo

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!

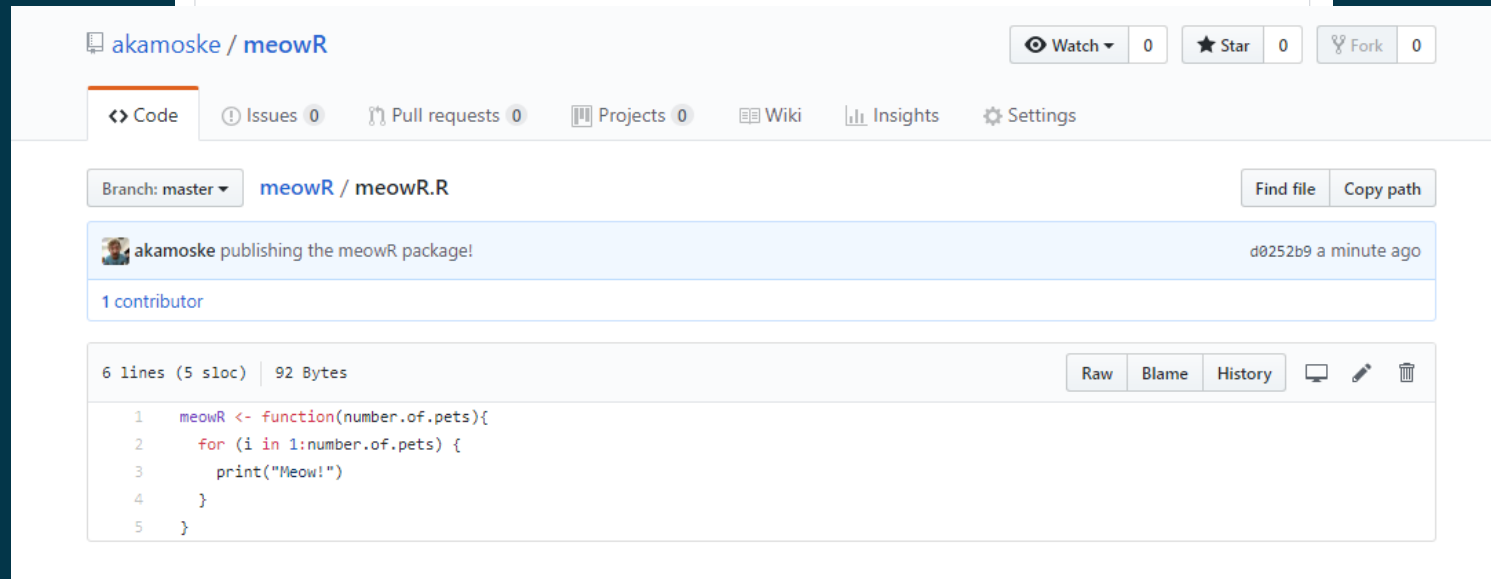
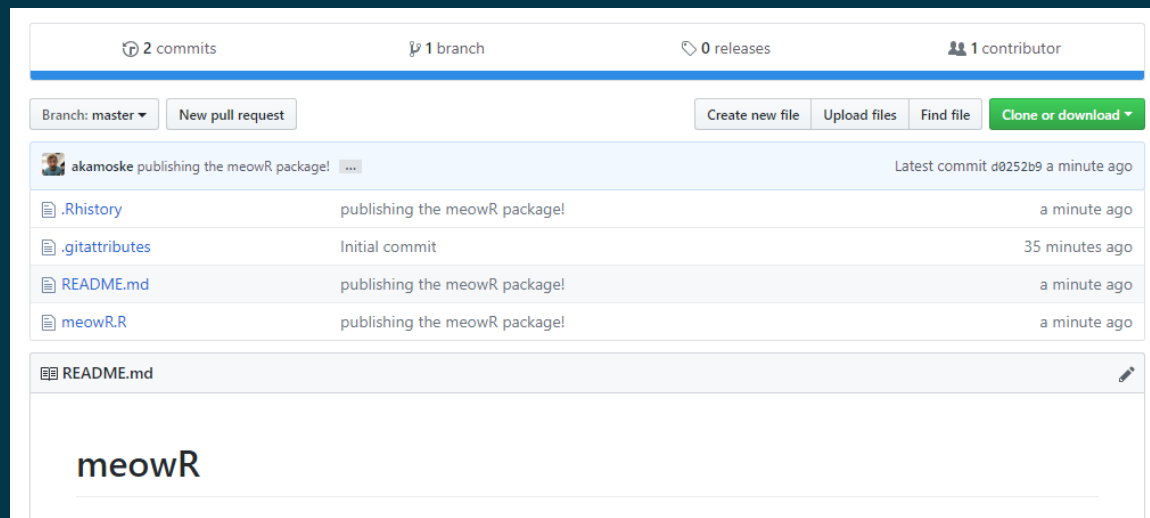


GitHub Desktop Quick Demo



Now we can push our function to our repo!

GitHub Desktop Quick Demo



meowR is alive!!!

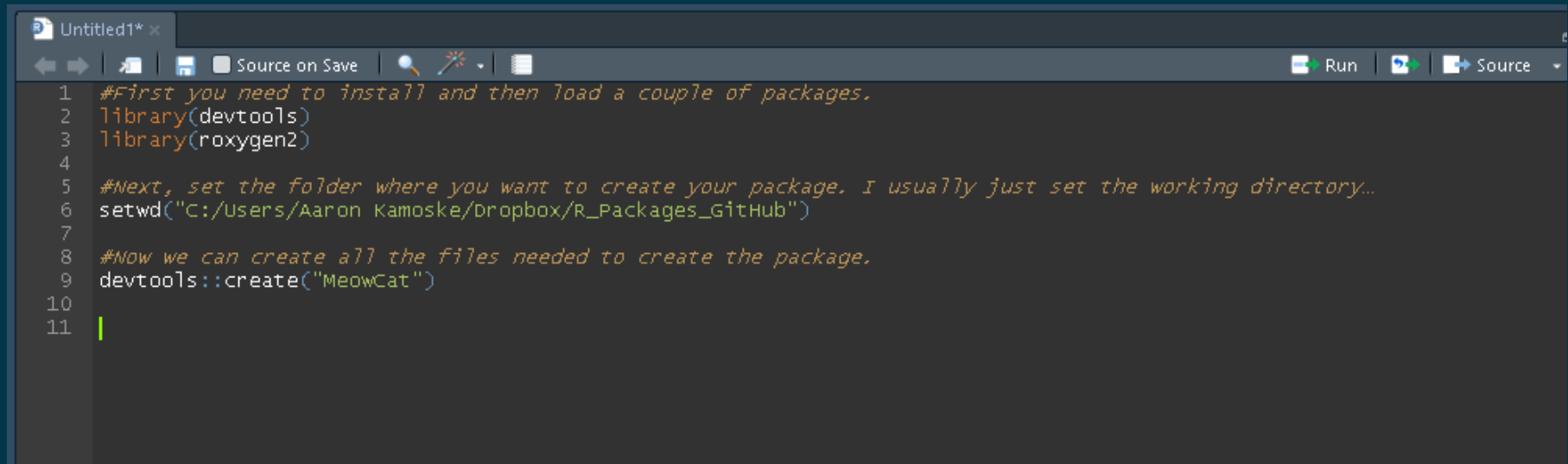
Creating R Packages With GitHub

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!



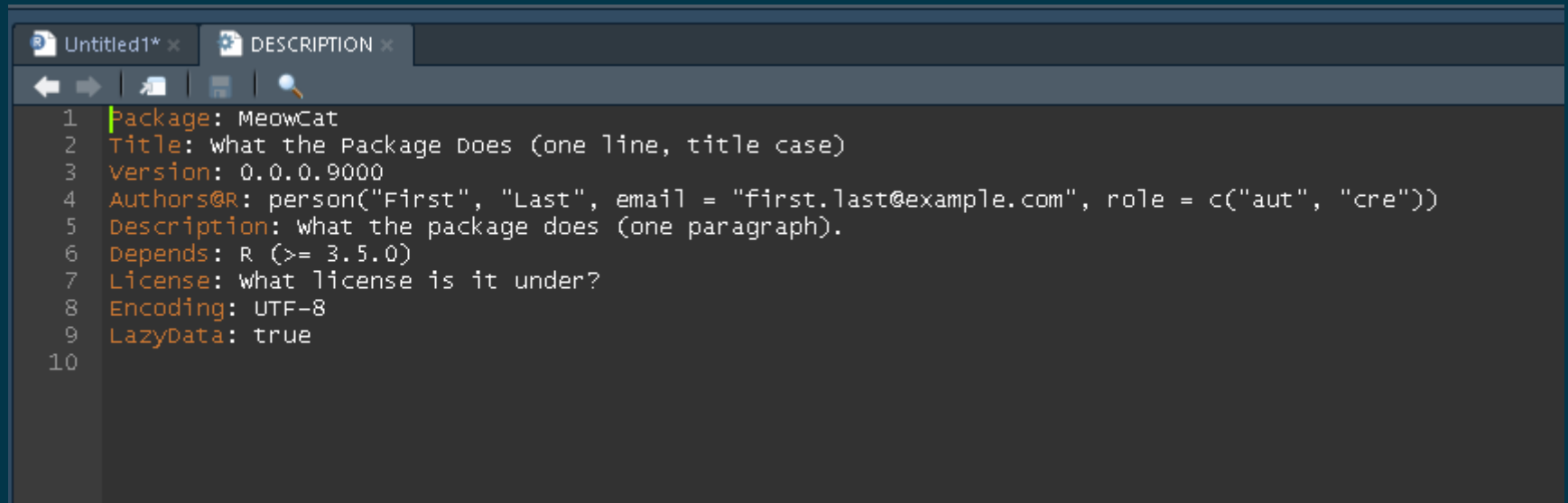
Creating R Packages With GitHub

A screenshot of the RStudio script editor. The window title is 'Untitled1*'. The toolbar shows 'Source on Save', 'Run', and 'Source' buttons. The script contains the following R code:

```
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
```

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.

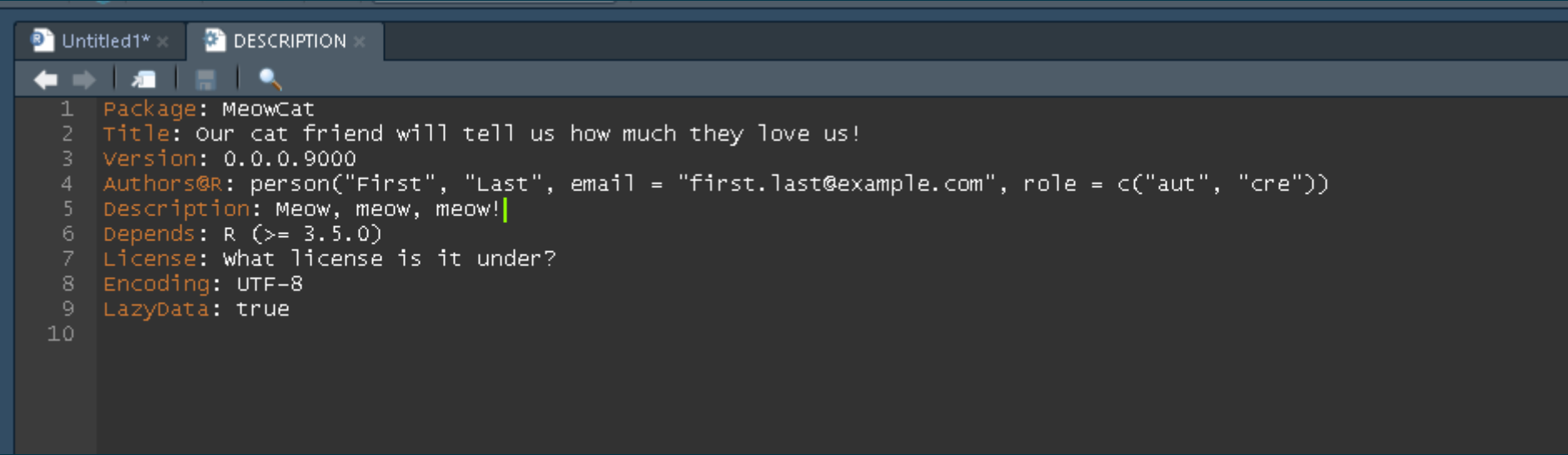
Creating R Packages With GitHub

A screenshot of a code editor window showing a file named 'DESCRIPTION'. The editor has a dark theme and a tab bar at the top with 'Untitled1*' and 'DESCRIPTION'. Below the tab bar is a toolbar with icons for back, forward, save, and search. The main area contains a text file with the following content:

```
1 Package: MeowCat
2 Title: What the Package Does (one line, title case)
3 Version: 0.0.0.9000
4 Authors@R: person("First", "Last", email = "first.last@example.com", role = c("aut", "cre"))
5 Description: what the package does (one paragraph).
6 Depends: R (>= 3.5.0)
7 License: what license is it under?
8 Encoding: UTF-8
9 LazyData: true
10
```

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

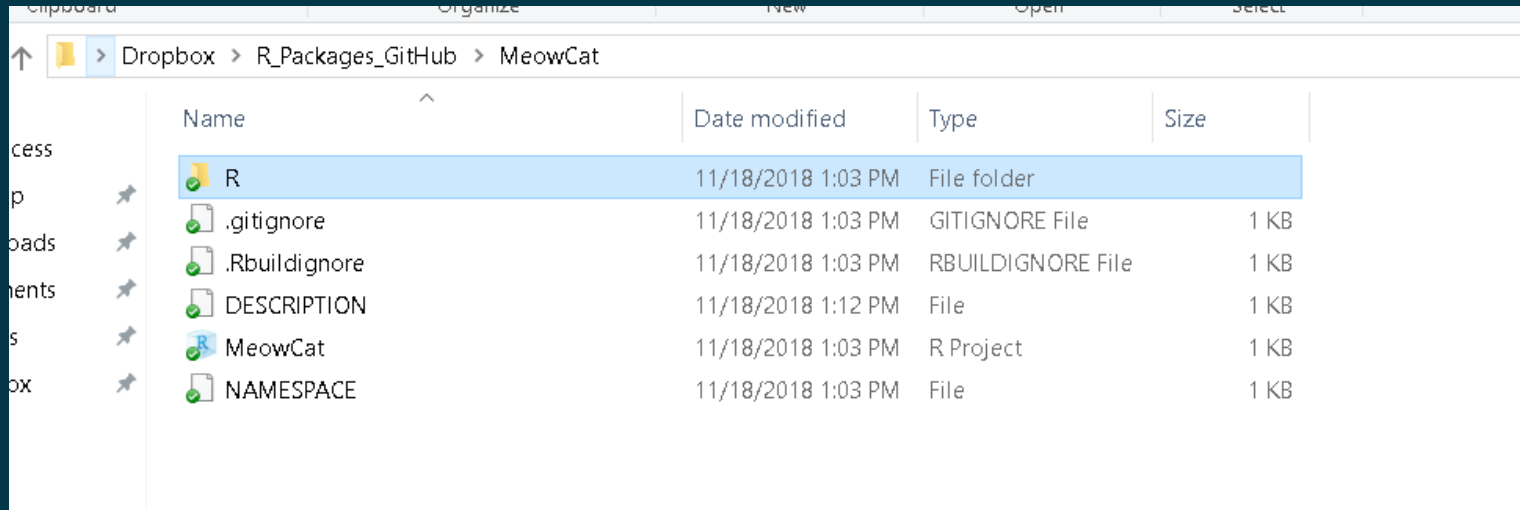
Creating R Packages With GitHub

A screenshot of a text editor window showing the contents of a DESCRIPTION file for an R package named 'MeowCat'. The editor has two tabs: 'Untitled1*' and 'DESCRIPTION'. The 'DESCRIPTION' tab is active, displaying the following text:

```
1 Package: MeowCat
2 Title: Our cat friend will tell us how much they love us!
3 Version: 0.0.0.9000
4 Authors@R: person("First", "Last", email = "first.last@example.com", role = c("aut", "cre"))
5 Description: Meow, meow, meow!
6 Depends: R (>= 3.5.0)
7 License: What license is it under?
8 Encoding: UTF-8
9 LazyData: true
10
```

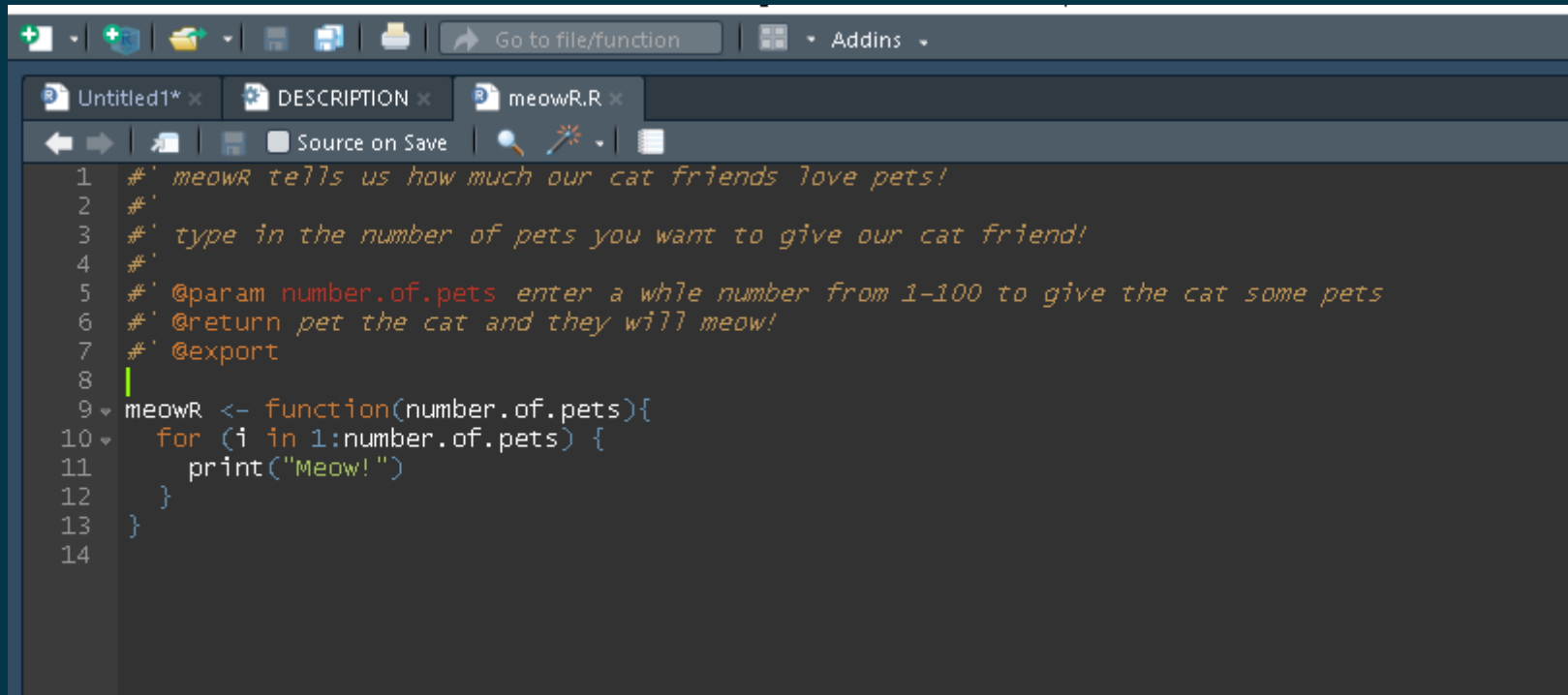
We can update this as needed!

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.

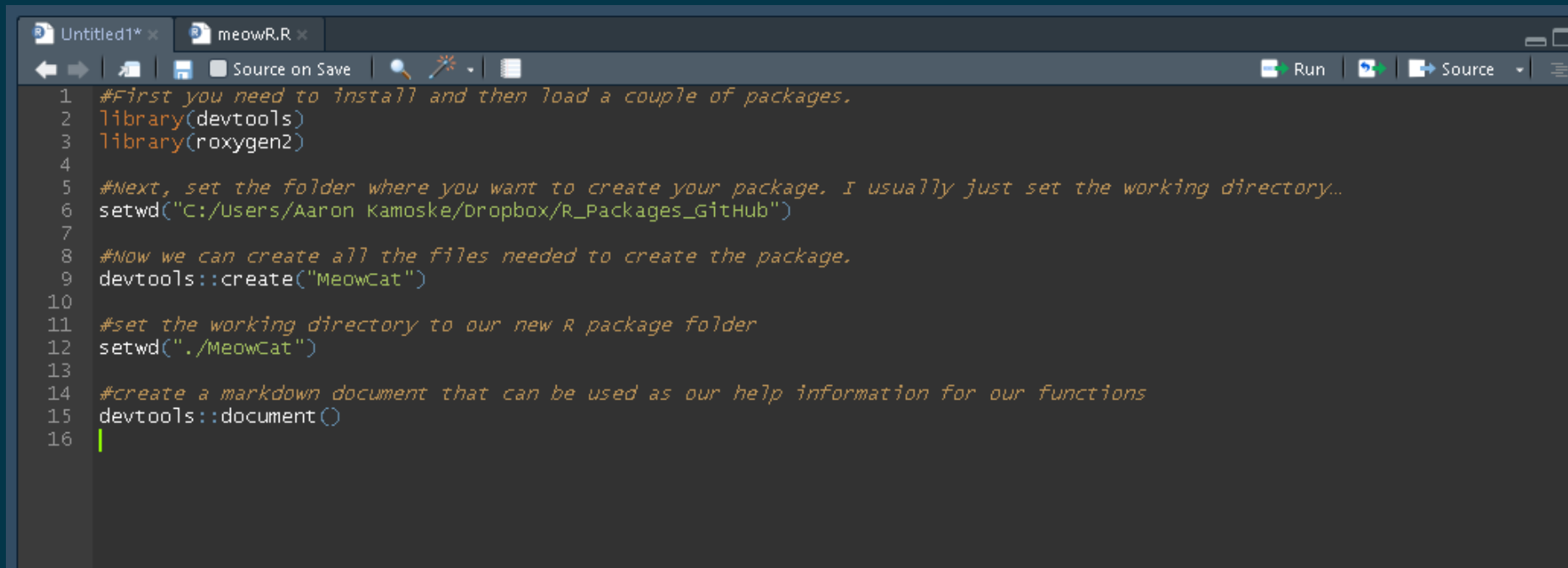
Creating R Packages With GitHub



```
1 #' meowR tells us how much our cat friends love pets!
2 #'
3 #' type in the number of pets you want to give our cat friend!
4 #'
5 #' @param number.of.pets enter a whole number from 1-100 to give the cat some pets
6 #' @return pet the cat and they will meow!
7 #' @export
8
9 meowR <- function(number.of.pets){
10   for (i in 1:number.of.pets) {
11     print("Meow!")
12   }
13 }
14
```

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

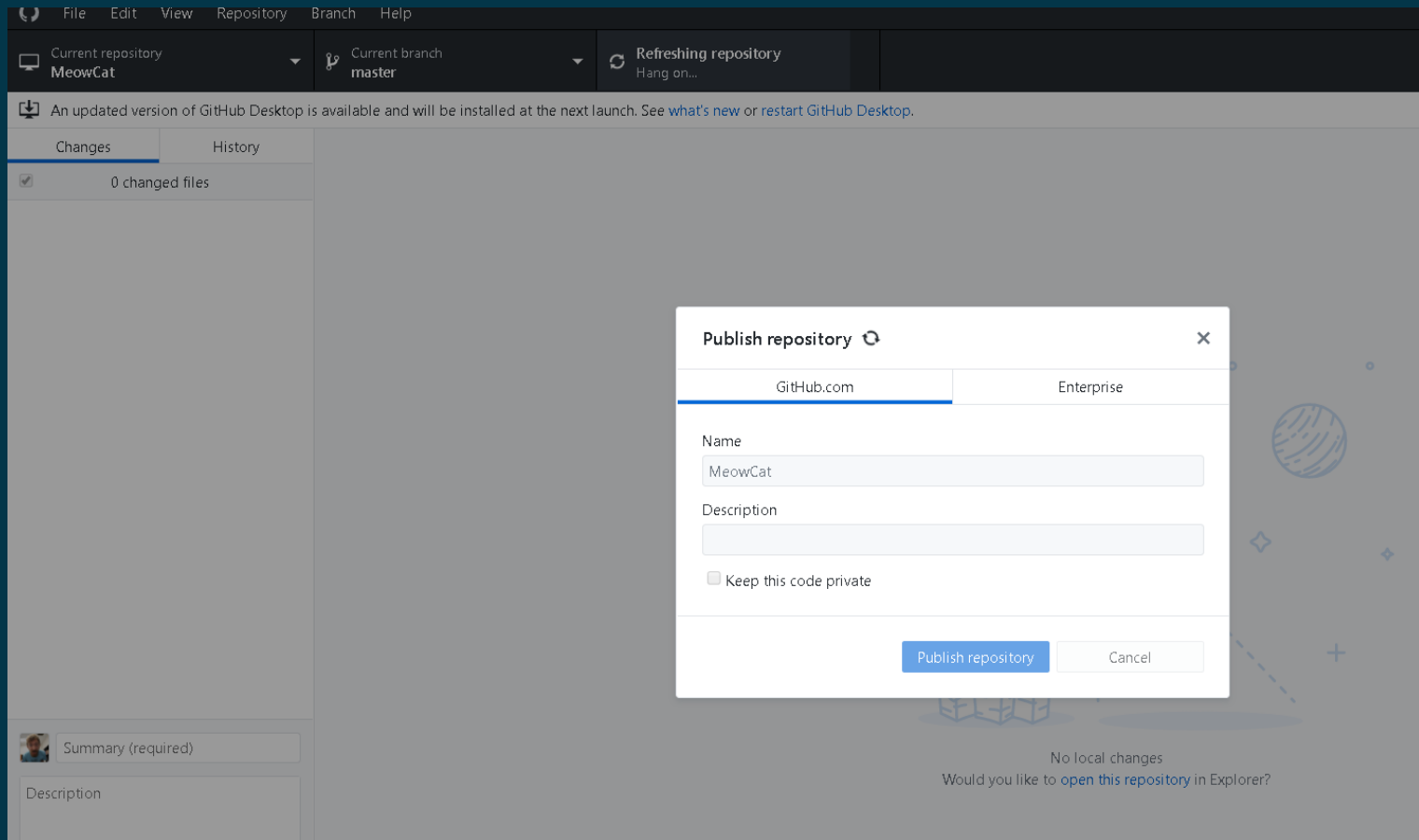
A screenshot of the RStudio IDE interface. The top pane shows two tabs: 'Untitled1*' and 'meowR.R'. The 'meowR.R' tab is active, displaying R code. The code includes comments in orange and function calls in black. The code is as follows:

```
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
```

The bottom pane is empty. The RStudio toolbar is visible at the top, showing icons for navigation and execution.

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

The screenshot shows the GitHub interface for a repository named 'MeowCat' by user 'akamoske'. The repository has 0 Watchers, 0 Stars, and 0 Forks. The 'Code' tab is selected, showing a message: 'No description, website, or topics provided.' Below this, it indicates '1 commit', '1 branch', '0 releases', and '1 contributor'. A table lists the files in the repository, all of which were committed 'a minute ago' in the 'Initial commit'. The files include 'R', 'man', '.Rbuildignore', '.gitattributes', '.gitignore', 'DESCRIPTION', 'MeowCat.Rproj', and 'NAMESPACE'. At the bottom, there is a prompt to 'Add a README'.

akamoske / MeowCat

Watch 0 Star 0 Fork 0

Code Issues 0 Pull requests 0 Projects 0 Wiki Insights Settings

No description, website, or topics provided. Edit

Manage topics

1 commit 1 branch 0 releases 1 contributor

Branch: master New pull request Create new file Upload files Find file Clone or download

akamoske Initial commit	Latest commit 45c9c6b a minute ago
R	Initial commit a minute ago
man	Initial commit a minute ago
.Rbuildignore	Initial commit a minute ago
.gitattributes	Initial commit a minute ago
.gitignore	Initial commit a minute ago
DESCRIPTION	Initial commit a minute ago
MeowCat.Rproj	Initial commit a minute ago
NAMESPACE	Initial commit a minute ago

Help people interested in this repository understand your project by adding a README. Add a README

Its alive!

Creating R Packages With GitHub

The screenshot displays the RStudio IDE interface. The main editor window shows an R script with the following code:

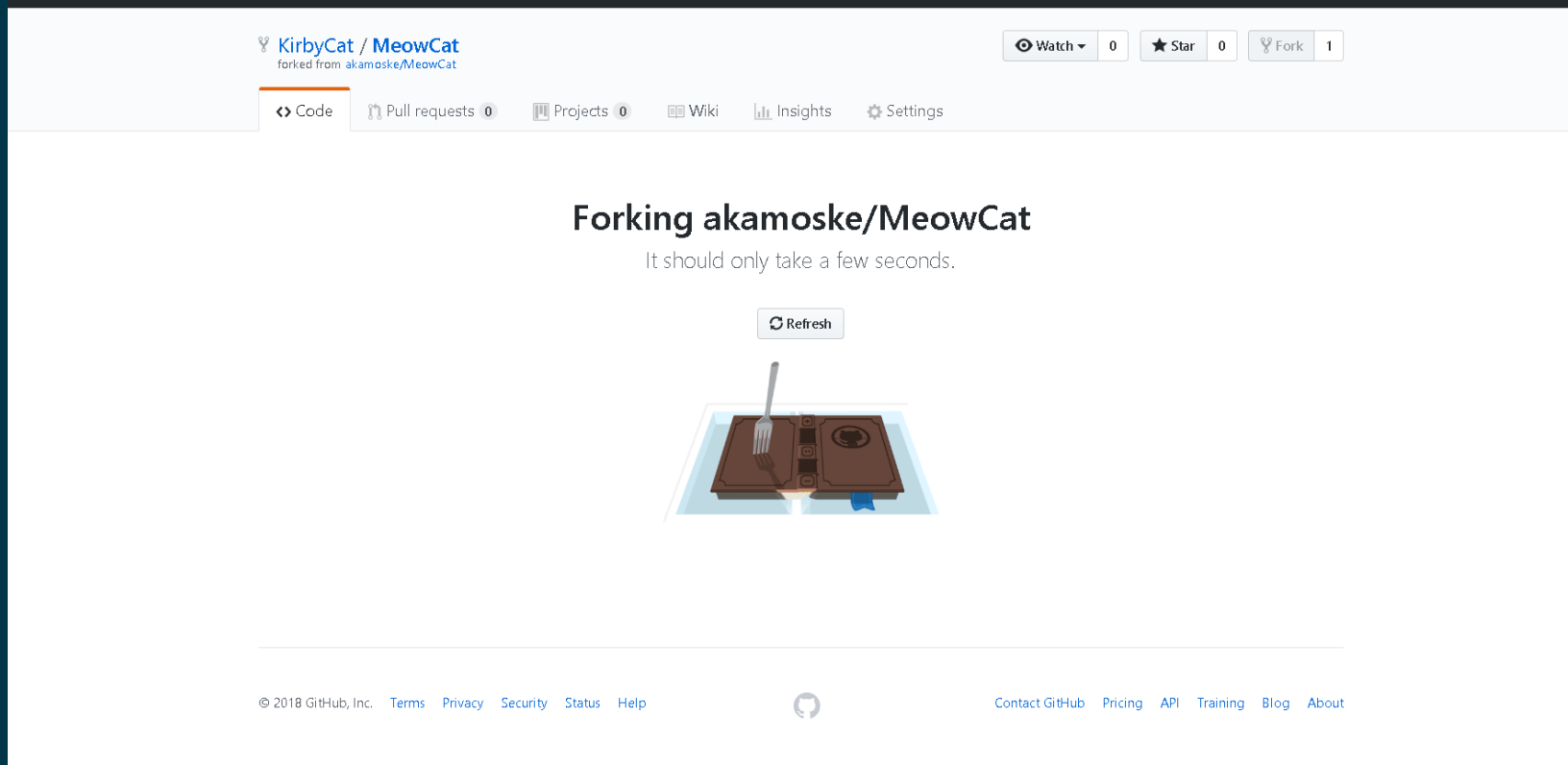
```
1 #First you need to install and then load a couple of packages.
2 library(devtools)
3 library(roxygen2)
4
5 #Next, set the folder where you want to create your package. I usually just set the working directory.
6 setwd("C:/Users/Aaron Kamoske/Dropbox/R_Packages_GitHub")
7
8 #Now we can create all 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
17 #lets load our package and put it to use
18 install_github("akamoske/MeowCat")
19
20 #lets load our library
21 library(MeowCat)
22
23 #lets see what our function is about
24 ?meowR
25
26 #lets run meowR
27 meowR(66)
28
```

The console window at the bottom shows the output of the `meowR(66)` function, which is a series of 66 lines, each starting with `[1]` followed by a string of 66 'M' characters and a space, representing the cat meowing.

The right-hand pane shows the R Documentation for the `meowR` package. The title is `meowR (MeowCat)`. The description is `meowR tells us how much our cat friends love pets!`. The usage is `meowR(number.of.pets)`. The arguments section states: `number.of.pets` enter a whole number from 1-100 to give the cat some pets. The value section states: `pet the cat and they will meow!`. At the bottom, it says `[Package: MeowCat version 0.0.0.9000 Index]`.

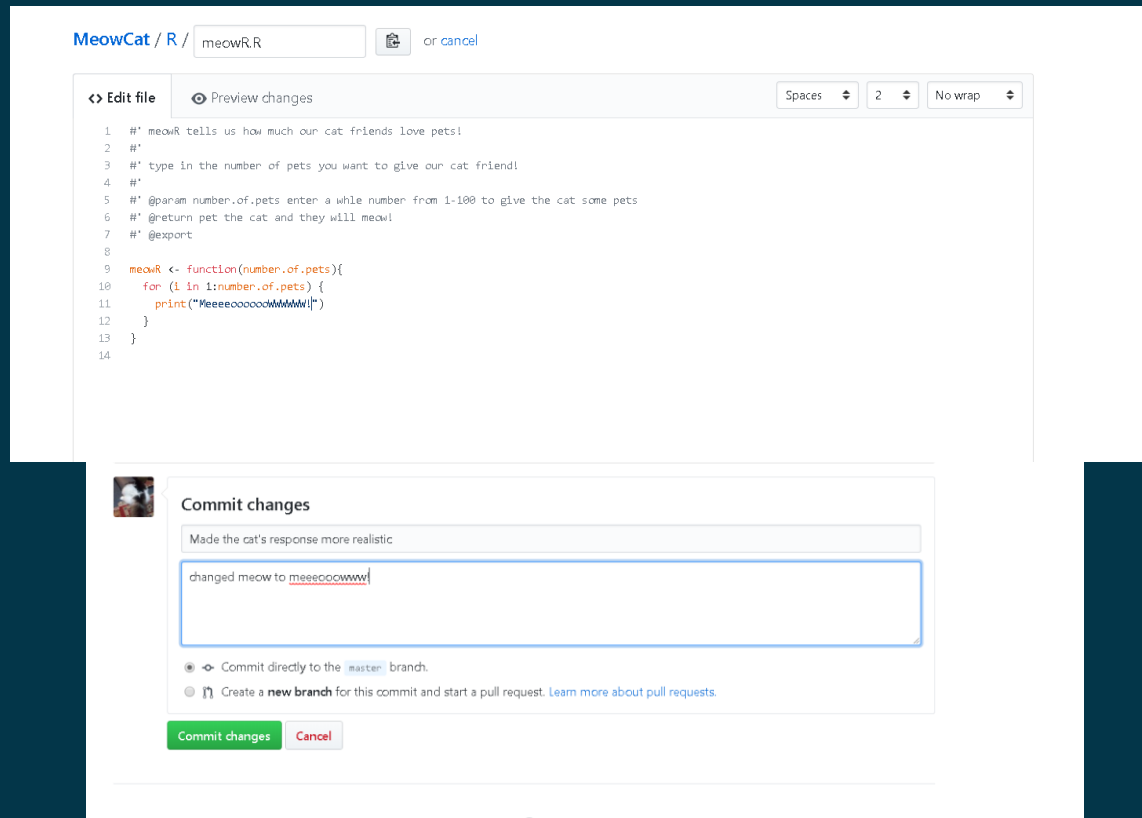
Lets load our package and put it to use!

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

Open a pull request

Create a new pull request by comparing changes across two branches. If you need to, you can also [compare across forks](#).


base fork: akamoske/MeowCat

base: master

head fork: KirbyCat/MeowCat

compare: master

✓ Able to merge. These branches can be automatically merged.



Made the cat's response more realistic

Write

Preview

AA B i “ <> @

changed meow to meeeoooooww!

Attach files by dragging & dropping, [selecting them](#), or pasting from the clipboard.

☒ Allow edits from maintainers. [Learn more](#)

Create pull request

1 commit

1 file changed

0 commit comments

1 contributor

Commits on Nov 18, 2018

KirbyCat

Made the cat's response more realistic

Verified

f99db27

Showing 1 changed file with 1 addition and 1 deletion.

Unified Split

2 R/meow.R

Copy path View file

@@ -8,6 +8,6 @@

8 8

9 9 meowR <- function(number.of.pets){

10 10 for (i in 1:number.of.pets) {

11 - print("meow")

11 + print("meeeoooooww!")

12 12 }

13 13 }

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

GitHub as a Collaborative Tool

The screenshot shows the GitHub web interface for the repository 'akamoske / MeowCat'. At the top, there are buttons for 'Watch' (0), 'Star' (0), and 'Fork' (1). Below this is a navigation bar with links to 'Code', 'Issues' (0), 'Pull requests' (1), 'Projects' (0), 'Wiki', 'Insights', and 'Settings'. The 'Pull requests' tab is selected and highlighted with an orange border. Below the navigation bar, there is a search filter 'is:pr is:open' and buttons for 'Labels' and 'Milestones'. A green button labeled 'New pull request' is on the right. The main content area shows a list of pull requests. The first pull request is titled 'Made the cat's response more realistic' by KirbyCat, opened a minute ago. It has a checkbox on the left and a status of '1 Open'. Below the list, there is a 'ProTip!' message: 'no:milestone will show everything without a milestone.'

akamoske / MeowCat

Watch 0 Star 0 Fork 1

Code Issues 0 Pull requests 1 Projects 0 Wiki Insights Settings

Filters is:pr is:open Labels Milestones New pull request

1 Open 0 Closed Author Labels Projects Milestones Reviews Assignee Sort

Made the cat's response more realistic
#1 opened a minute ago by KirbyCat

ProTip! no:milestone will show everything without a milestone.

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

GitHub as a Collaborative Tool

The screenshot displays a GitHub pull request page. At the top, navigation tabs include Code, Issues (0), Pull requests (1), Projects (0), Wiki, Insights, and Settings. The main heading is 'Made the cat's response more realistic #1' with an 'Edit' button. Below this, a green 'Open' button is followed by the text 'KirbyCat wants to merge 1 commit into akamoske:master from KirbyCat:master'. A summary bar shows 'Conversation 0', 'Commits 1', 'Checks 0', and 'Files changed 1'. The 'Changes from all commits' section shows a diff of '+1 -1' with a 'Diff settings' dropdown and a green 'Review changes' button. The commit message is 'Made the cat's response more realistic' with a description 'changed meow to meeeooooowww!'. The commit is by KirbyCat, committed 9 minutes ago, and is verified. The commit hash is f99d0e7708b88ddf5ff0192c34dabb0beb05d506. The file 'R/meowR.R' is shown with a diff of @@ -8,6 +8,6 @@. The code shows a function 'meowR' with a loop. Line 11 is highlighted in red, showing a change from 'print("Meow!")' to 'print("Meeeeeooooowww!")'. A 'ProTip!' message at the bottom suggests using 'n' and 'p' to navigate between commits.

<> Code 0 Issues 0 Pull requests 1 Projects 0 Wiki Insights Settings

Made the cat's response more realistic #1

Edit

Open KirbyCat wants to merge 1 commit into akamoske:master from KirbyCat:master

Conversation 0 Commits 1 Checks 0 Files changed 1

Changes from all commits Jump to... +1 -1 Diff settings Review changes

Made the cat's response more realistic
changed meow to meeeooooowww!

KirbyCat committed 9 minutes ago Verified commit f99d0e7708b88ddf5ff0192c34dabb0beb05d506

2 R/meowR.R Copy path View file

```
@@ -8,6 +8,6 @@
8      8
9      9      meowR <- function(number.of.pets){
10     10      for (i in 1:number.of.pets) {
11     - print("Meow!")
11     + print("Meeeeeooooowww!")
12     12      }
13     13      }
```

ProTip! Use `n` and `p` to navigate between commits in a pull request.

We can see these changes.

GitHub as a Collaborative Tool

The screenshot displays a GitHub pull request interface. At the top, a comment by KirbyCat is shown, stating 'changed meow to meeeoooooww!'. Below this, a commit history entry is visible, indicating a change to 'Made the cat's response more realistic' with a commit hash of f99d0e7. A green box highlights the merge status, stating 'Continuous integration has not been set up' and 'This branch has no conflicts with the base branch'. A green button labeled 'Merge pull request' is present. At the bottom, a comment form is open, showing the text 'Thanks for the tip! I'll merge these together!'. The form includes a rich text editor with various formatting options and a 'Comment' button.

KirbyCat commented 3 minutes ago

changed meow to meeeoooooww!

Made the cat's response more realistic

changed meow to meeeoooooww!

Add more commits by pushing to the **master** branch on KirbyCat/MeowCat

Continuous integration has not been set up
Several apps are available to automatically catch bugs and enforce style.

✓ This branch has no conflicts with the base branch
Merging can be performed automatically.

Merge pull request

Thanks for the tip! I'll merge these together!

Attach files by dragging & dropping, selecting them, or pasting from the clipboard.

Styling with Markdown is supported



Close and comment


Comment


Lets leave a comment and then accept the changes!

GitHub as a Collaborative Tool

Add more commits by pushing to the **master** branch on KirbyCat/MeowCat

**Continuous integration has not been set up**
[Several apps are available](#) to automatically catch bugs and enforce style.

**This branch has no conflicts with the base branch**
Merging can be performed automatically.

Merge pull request  You can also [open this in GitHub Desktop](#) or view [command line instructions](#).

Lets leave a comment and then accept the changes!

GitHub as a Collaborative Tool

The screenshot shows the GitHub interface for the repository 'akamoske / MeowCat'. At the top, there are buttons for 'Watch' (0), 'Star' (0), and 'Fork' (1). Below this is a navigation bar with links for 'Code', 'Issues' (0), 'Pull requests' (0), 'Projects' (0), 'Wiki', 'Insights', and 'Settings'. The main content area shows the file 'MeowCat / R / meowR.R' on the 'master' branch. A recent commit by KirbyCat is displayed, with the message 'Made the cat's response more realistic' and a commit hash 'f99d0e7' from 12 minutes ago. Below the commit message, it says '2 contributors' with their avatars. The code for 'meowR.R' is shown, consisting of 14 lines (12 sloc) and 373 Bytes. The code is a function that prints a string of 'M' characters followed by 'www!' based on the number of pets entered. The code is displayed in a light gray box with line numbers on the left and action buttons (Raw, Blame, History, and icons for file operations) on the right.

akamoske / MeowCat

Watch 0 Star 0 Fork 1

Code Issues 0 Pull requests 0 Projects 0 Wiki Insights Settings

Branch: master MeowCat / R / meowR.R Find file Copy path

KirbyCat Made the cat's response more realistic f99d0e7 12 minutes ago

2 contributors

14 lines (12 sloc) 373 Bytes Raw Blame History

```
1  #' meowR tells us how much our cat friends love pets!
2  #'
3  #' type in the number of pets you want to give our cat friend!
4  #'
5  #' @param number.of.pets enter a whole number from 1-100 to give the cat some pets
6  #' @return pet the cat and they will meow!
7  #' @export
8
9  meowR <- function(number.of.pets){
10    for (i in 1:number.of.pets) {
11      print("Meeeeoooooooowww!")
12    }
13  }
```

We can see the changes now!

GitHub as a Collaborative Tool

[illegible]

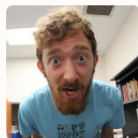
Lets reload our package in R and see how it works now!

GitHub as a Collaborative Tool

The ability to carry out 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!



Aaron G. Kamoske

PhD student interested in remote sensing, ecogeography, and landscape ecology

[About](#) [Research](#) [Publications](#) [CV](#) [Blog](#)

Helpful links!

Pull requests: [https://www.youtube.com/watch?v= NrSWLQsDL4](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>