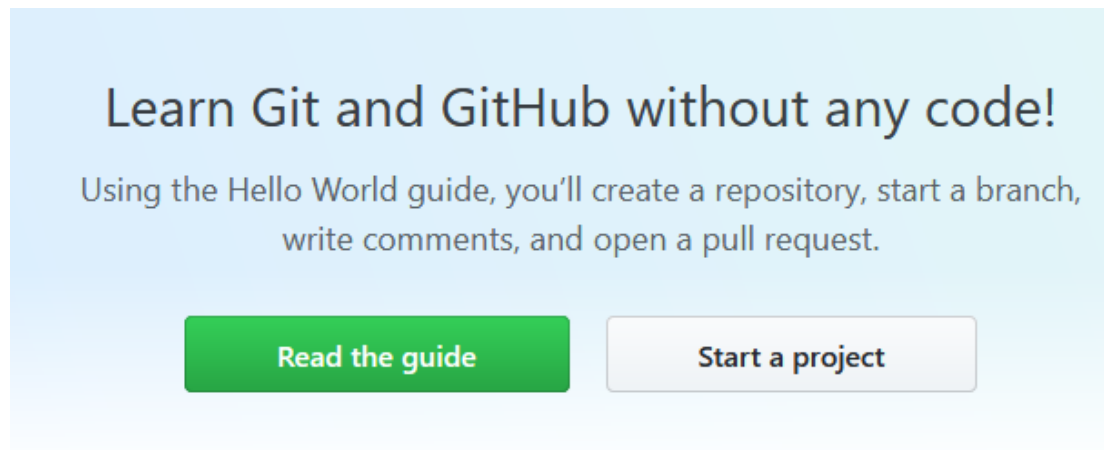


RStudio – Using Git with Github

The most important step before starting this guide is to create a GitHub account, this can be done at <https://github.com/join?source=header-home>. (Note: A student account can be activated which gives free access to private repositories so only invited people can access it as opposed to the public)

Student: <https://education.github.com/pack>

Once you have created your account you will be greeted by this on your homepage:




Start a project which will bring you to this page:

Create a new repository

A repository contains all the files for your project, including the revision history.

Owner

 rtho2192 ▾


Repository name

✓


Great repository names

Your new repository will be created as My-First-Project not verbose-octo-goggles.

Description (optional)

☒  **Public**

Anyone can see this repository. You choose who can commit.


☐  **Private**

You choose who can see and commit to this repository.

☐ **Initialize this repository with a README**

This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

Add .gitignore: **None** ▾

Add a license: **None** ▾ 

Create repository

The repository name is the name of your work and allows others to get a quick overview of what your project is about, so give it a good name. 😊

When you create your repository you will be greeted by this setup page. The most important part is the link next to 'HTTPS' and underneath 'Quick setup – if...' you **WILL** need this for RStudio.

rtho2192 / My-First-Project

Watch 0

Star 0

Fork 0

<> Code

Issues 0

Pull requests 0

Projects 0

Wiki

Insights

Settings

Quick setup — if you've done this kind of thing before

Set up in Desktop

 or

HTTPS

SSH

https://github.com/rtho2192/My-First-Project.git

We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

...or create a new repository on the command line

```
echo "# My-First-Project" >> README.md
git init
git add README.md
git commit -m "first commit"
git remote add origin https://github.com/rtho2192/My-First-Project.git
git push -u origin master
```

...or push an existing repository from the command line

```
git remote add origin https://github.com/rtho2192/My-First-Project.git
git push -u origin master
```

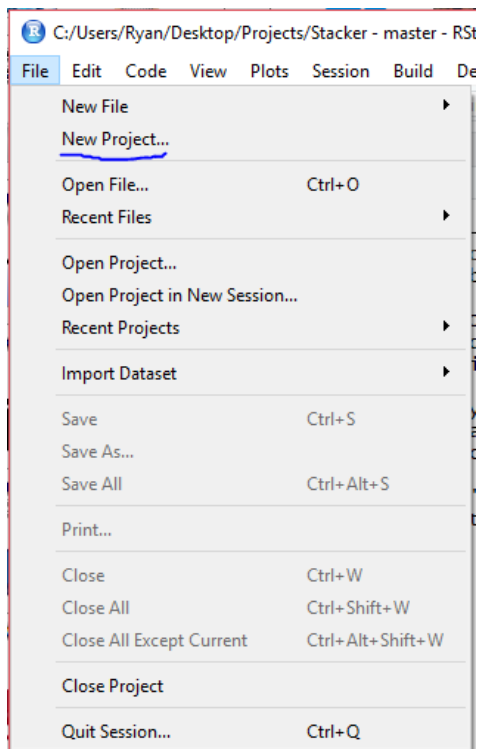
...or import code from another repository

You can initialize this repository with code from a Subversion, Mercurial, or TFS project.

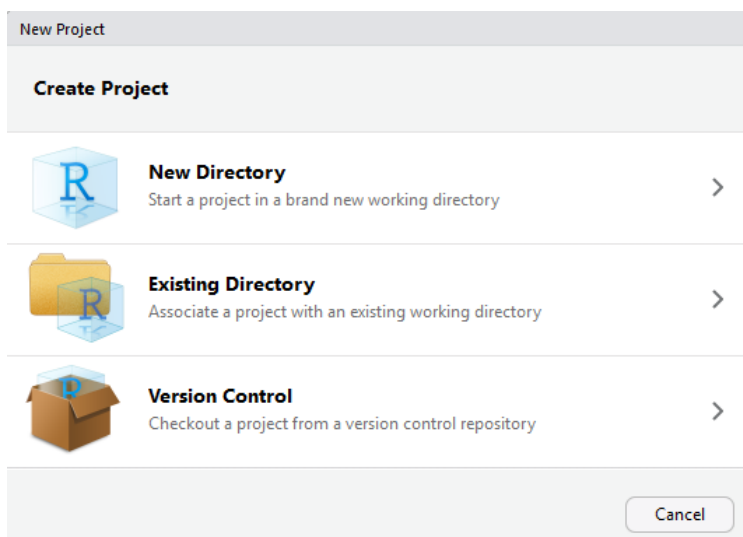
Import code

💡 ProTip! Use the URL for this page when adding GitHub as a remote.

In order to utilise git and GitHub with RStudio we need to make what's known as a project in RStudio. In order to do this select file then 'New Project...':



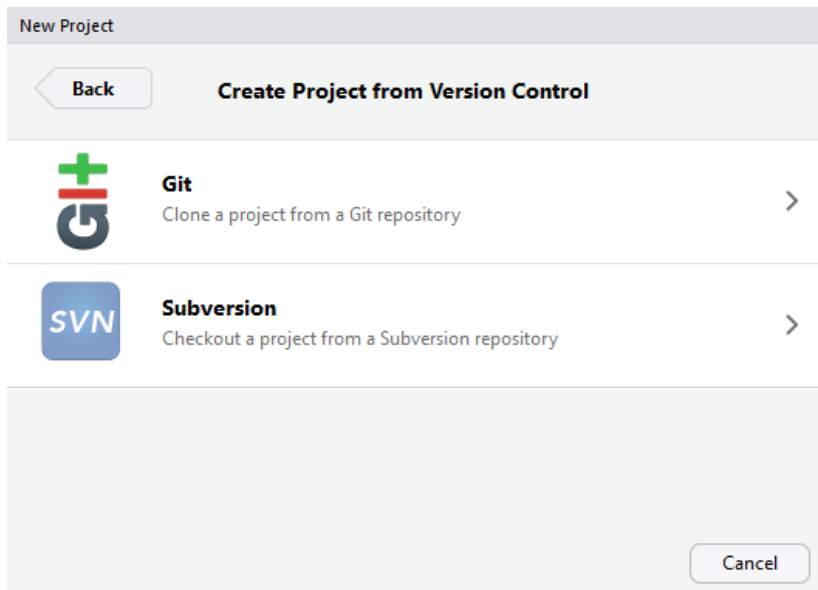
This brings up the following menu:



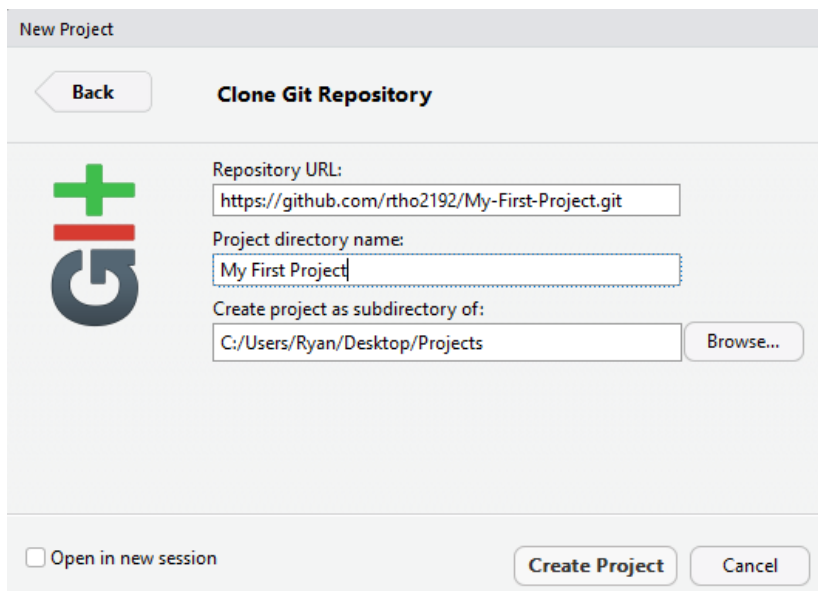
Version Control: This is the option you will select it will allow us to integrate Git with GitHub as opposed to the other options which will create a local Git.

Version Control

If you want to integrate GitHub with Git in RStudio select the 'Version Control' option. This will create the following menu:



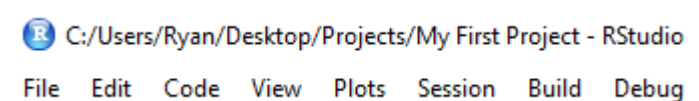
At this point select the 'Git' option.



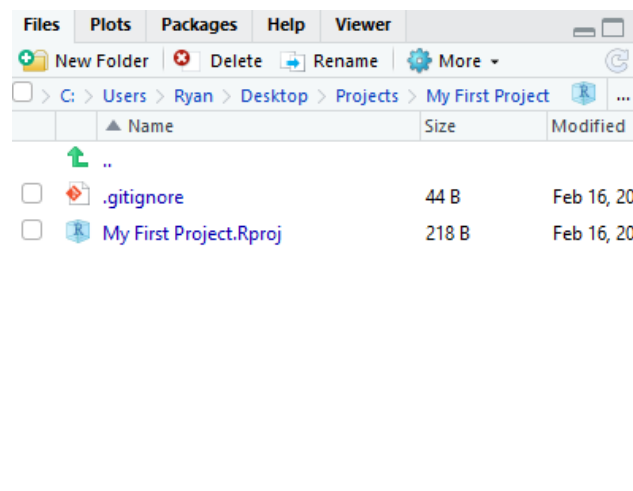
The 'Repository URL:' is the link GitHub provided when you created the repository which I said to keep a note of for this reason. Just paste that link in there.

I highly recommend creating a projects folder that will store all your projects that you will ever 'Push' or 'Pull' to GitHub. Directory name is what you want to call your project, give it a good clear name as your folder structure will become 'Projects/Directory Name' helps keep track of multiple projects.

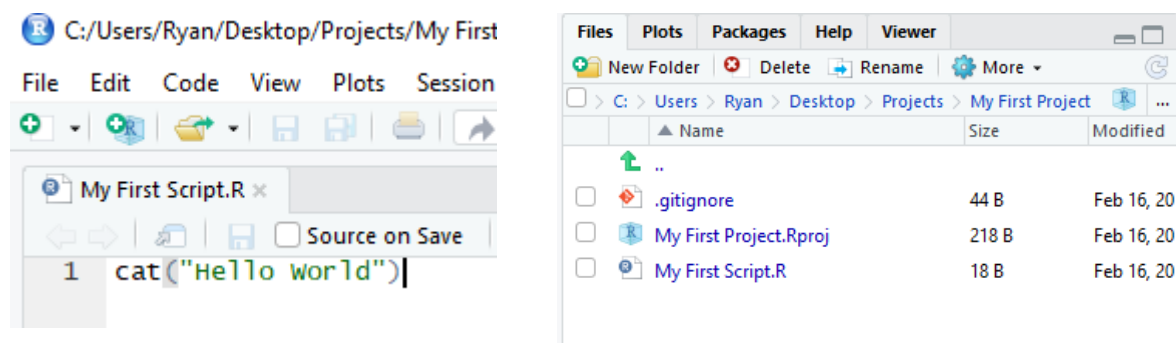
You will now be in project mode in RStudio, to make sure check the top of the RStudio window, it should say your project name:



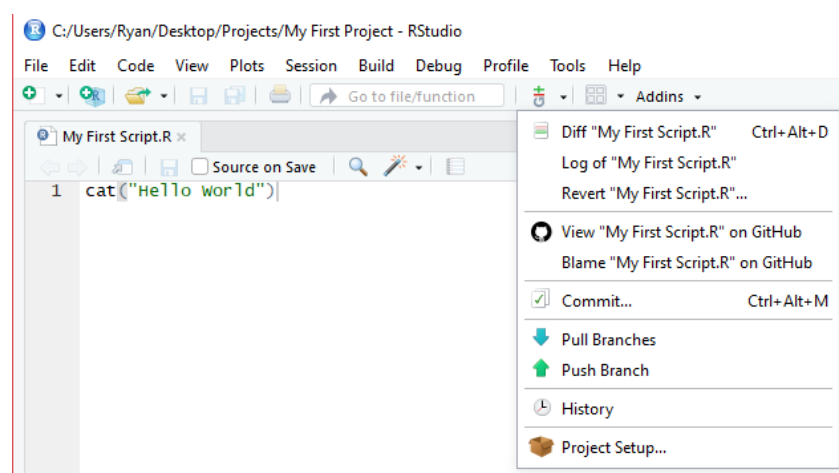
Project mode comes with the benefit that the working directory will automatically become your project folder:



Now let's make a simple script to go inside our project folder.

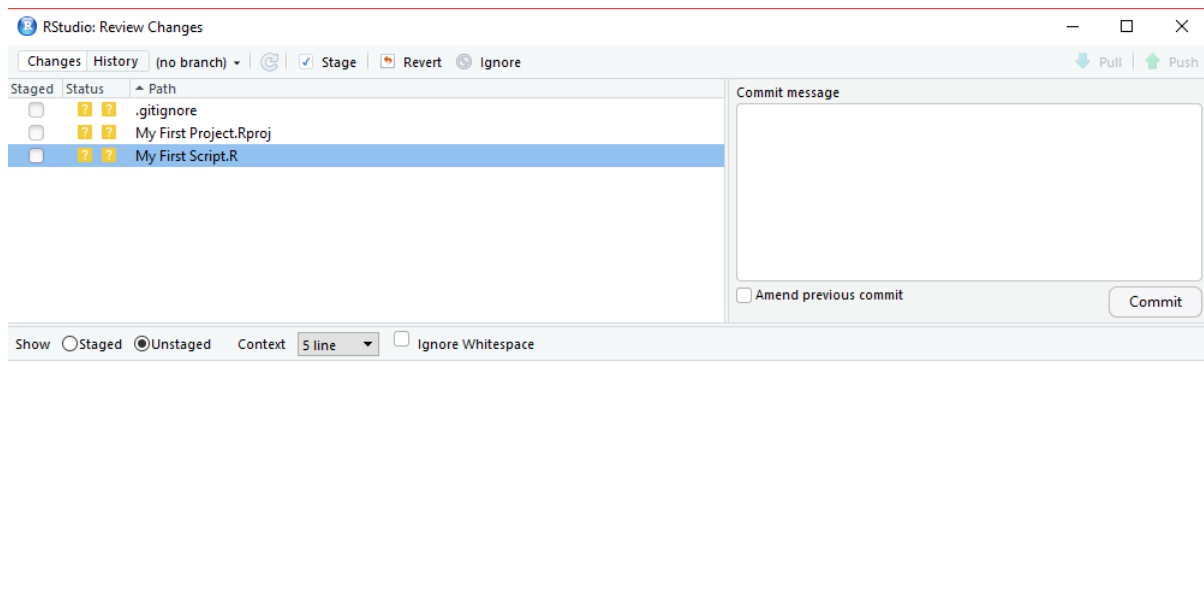


With the script created and saved as you can see it automatically saves into the project folder. Now to push this project into a new repository. In RStudio git is accessible by selecting the following button:



Pressing the Git button will bring up a number of options, 'Commit...' is the main option you will choose as this holds the GUI for staging files, making commits and push/pull to the repository. 'Diff "Script.R"' will bring up the 'Commit...' menu with 'Script.R' highlighted straight away. 'Log of' will bring up a GUI with the changes made to the code since the last commit. Finally, 'Revert' will

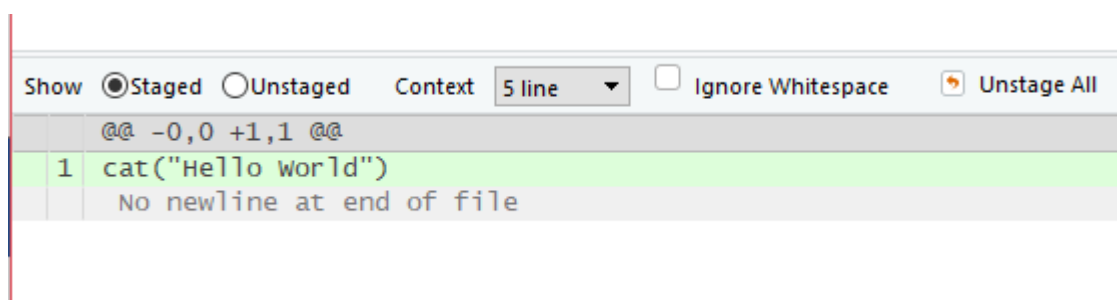
rollback your script to a previous commit if you make a big mistake implementing something new. Pressing 'Commit...' brings up the following menu:



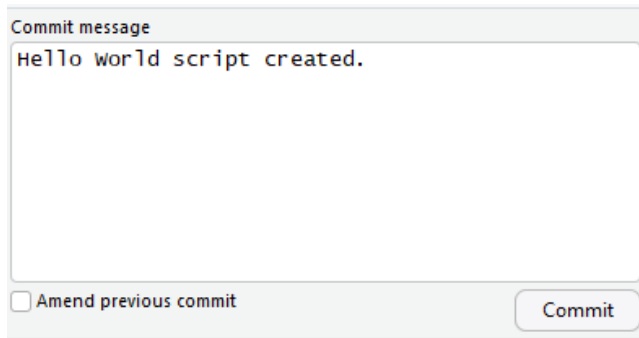
The top left of the menu shows which files will be changed since the last commit, in order to do that the files must be first staged. There are two ways to do this, either select the check box next to the file under the 'Staged' column or press the 'Stage' button. After staging it should look like this:

Staged	Status	Path
<input checked="" type="checkbox"/>	A	My First Script.R
<input checked="" type="checkbox"/>	A	My First Project.Rproj
<input checked="" type="checkbox"/>	A	.gitignore

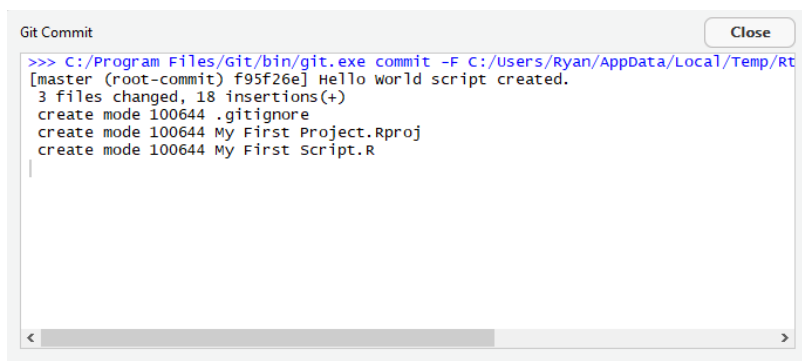
Also the bottom section of the GUI will essentially show the 'Log of' option from before so that changes that will be made to the file can be seen.



Now that all the files are staged it is time to make a commit, this is done with the top right box in the GUI, and in order to make a commit, a commit message **MUST** be present. In this message you need to give an overview of what has changed since the last commit, i.e. a new function has been added to the script.



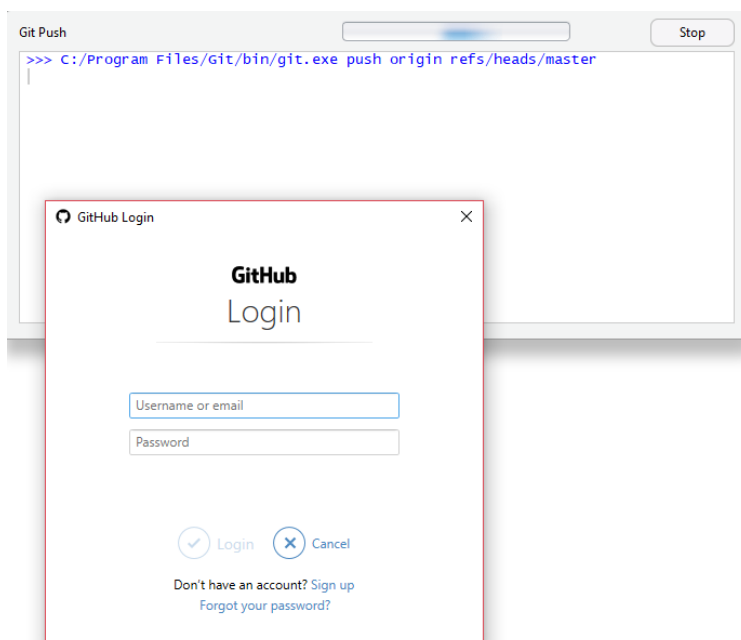
Once you have your commit message you can press the 'Commit' button to say you're happy with the files you staged. This will produce a console window which shows the progress of the commit.



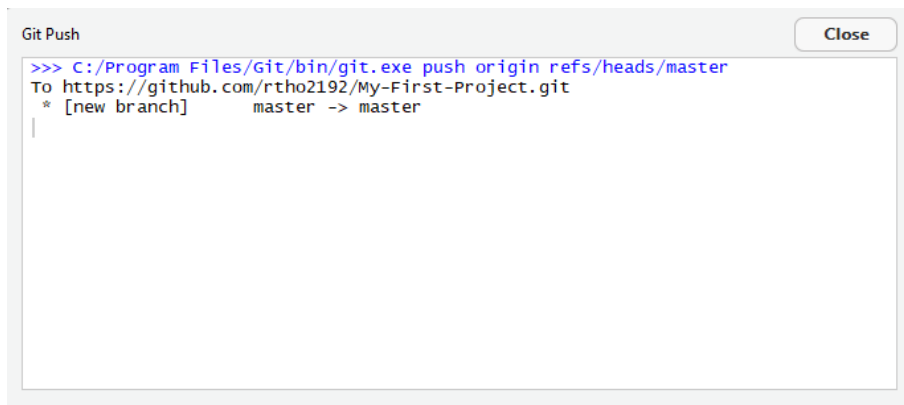
When it is finished you can press 'Close' and you are now ready to push to your repository. In order to do so press the 'Push' button above the commit box.



When you do this the console window will appear and the first time you push, the GitHub login page will appear, enter your credentials and login.

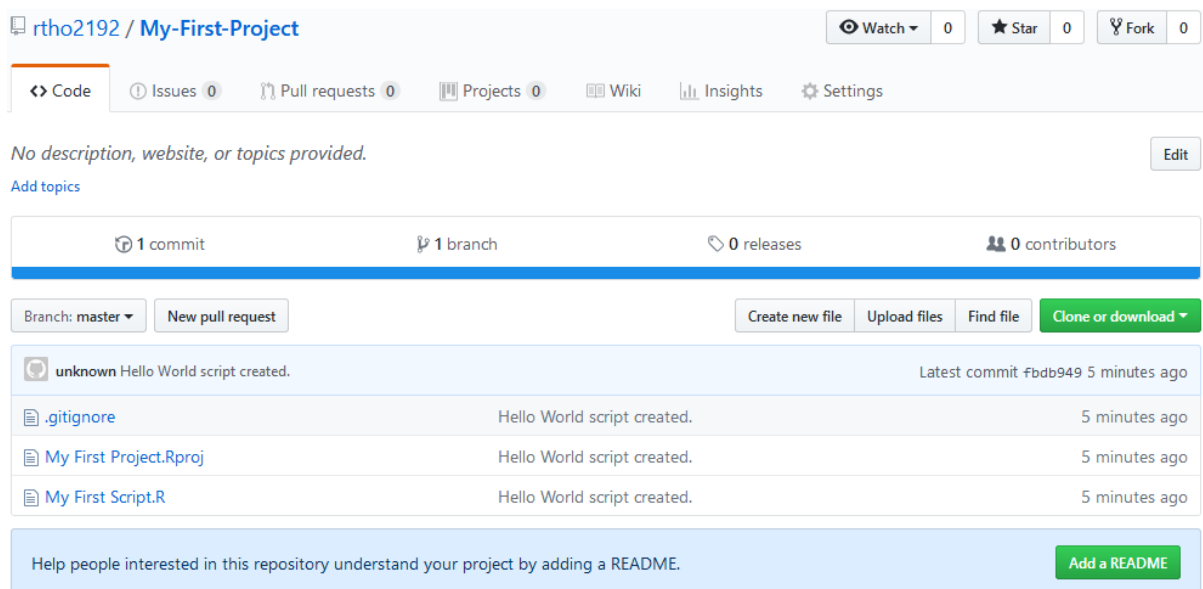


The console will now update to say it has pushed to the master branch.

A terminal window titled "Git Push" with a "Close" button in the top right corner. The terminal shows the command `>>> C:/Program Files/Git/bin/git.exe push origin refs/heads/master` and its output: `To https://github.com/rtho2192/My-First-Project.git` and `* [new branch] master -> master`.

```
>>> C:/Program Files/Git/bin/git.exe push origin refs/heads/master
To https://github.com/rtho2192/My-First-Project.git
* [new branch] master -> master
```

The repository has been updated! In order to check let's take a look on GitHub.



SUCCESS! The repository has been updated.

Pulling from a repository

If you want to take your project to another computer it is as simple as following the previous steps to setup a 'Version Control' project then going straight to the 'Commit...' GUI and selecting the 'Pull' option. This will take the current files in the master branch of the GitHub repository.

Further note

This is a basic guide to Git and GitHub with RStudio if you wish to know more such as branches from master and rolling back files, please let me know and I will create another guide. 😊