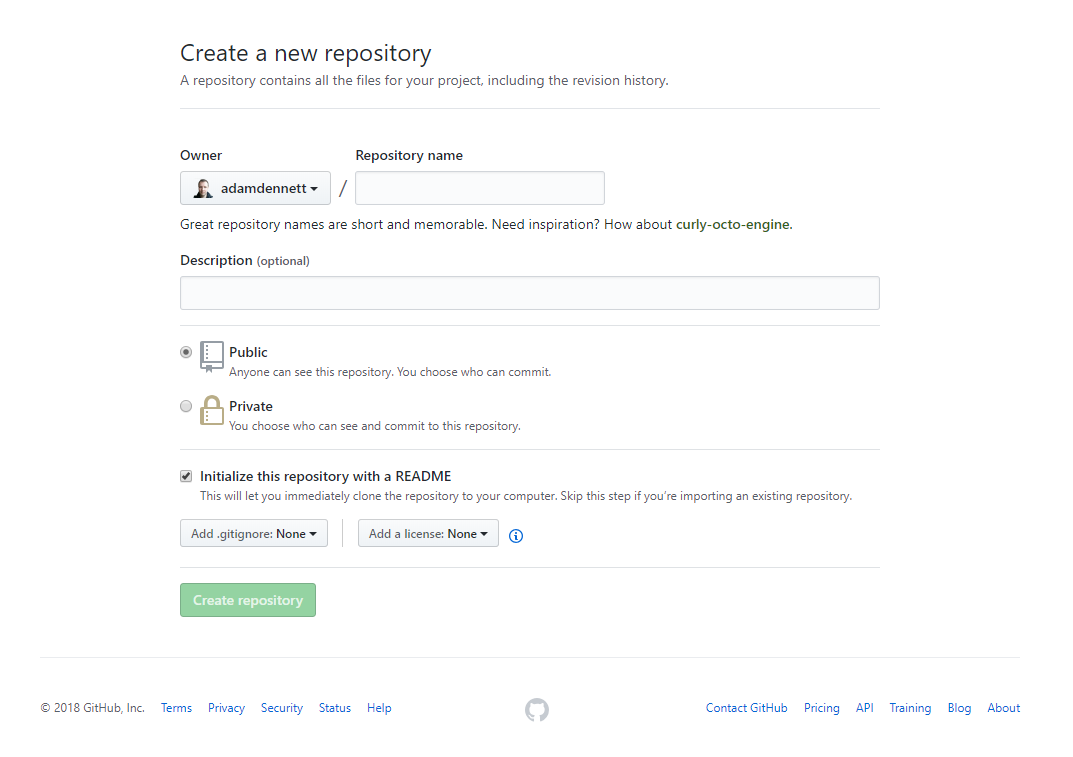
Producing Reproducible Research Using RMarkdown and Github

In this practical session, you will learn how to produce work that is open, reproducible and portable using RStudio, RMarkdown, Git and Github.

* [RStudio](https://www.rstudio.com/) is a graphical user interface (that you should already be familiar with) - it contains a number of features which make it excellent for authoring reproducible and open geographic data science work.
* RMarkdown is a version of the [Markdown](https://en.wikipedia.org/wiki/Markdown) markup language which enables plain text to be formatted to contain links to data, code to run, text to explain what you a producing and metadata to tell your software what kinds of outputs to generate from your markdown code. For more information on RMarkdown, visit [here](https://rmarkdown.rstudio.com/lesson-1.html).
* [Git](https://git-scm.com/) is a software version control system which allows you to keep track of the code you produce and the changes that you or others make to it.
* [GitHub](https://github.com/) is an online repository that allows anyone to view the code you have produced (in whatever language you choose to program in) and use/scrutinise/contribute to/comment on it.

### Setting up GitHub to store your code

1. If you are working on your own computer, you will first need to install git - <https://git-scm.com/> - if you are working on the UCL Remote Desktop, you won't need to do this as it is already installed for you.
2. Go to <http://github.com> and install github (if working on your own computer). Create an account and create a new repository (call it anything you like - 'gis\_code' or something similar), making sure it is public and you check the box that says 'initialise new repository with a README' - click 'create repository' at the bottom

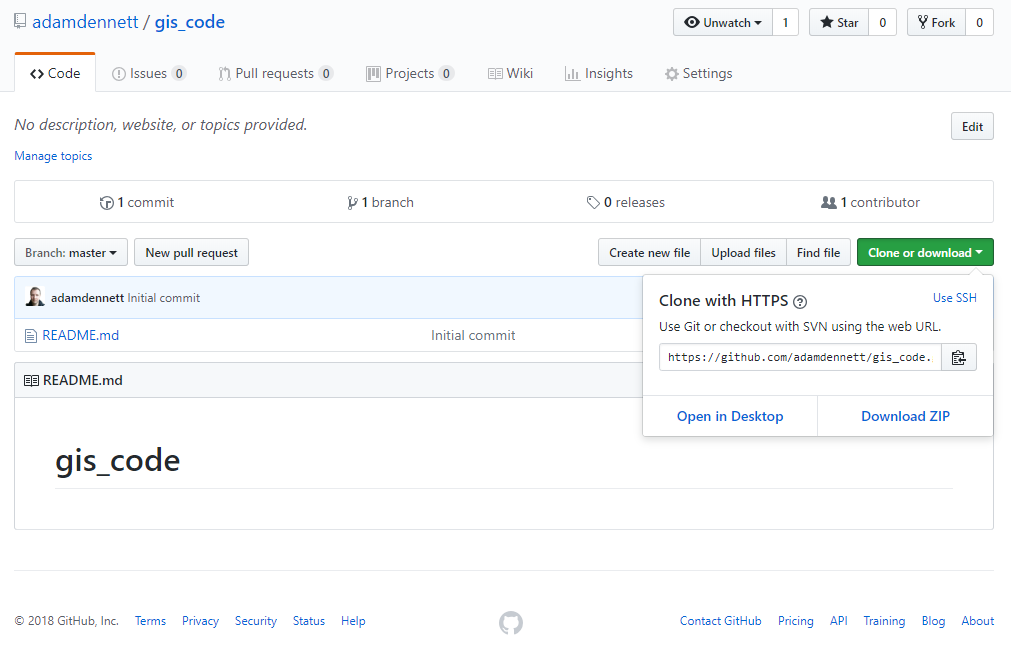


repo

1. Your new repository ('repo') will be created and this is where you will be able to store your code online. You will notice that a README.md markdown file has also been created. This can be edited to tell people what they are likely to find in this repository.
2. Now you have created your repo online, you need to 'clone' it so that there is an identical copy of it in a local folder on your computer.

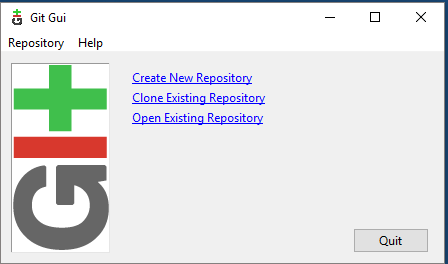
There are a couple of ways of doing this, but the easy one is to use the GUI that comes packaged with your git installation.

1. The first thing you need to do is copy the Clone URL for your repo from the github website - click the green button in your repo for 'Clone or Download' and copy the link:



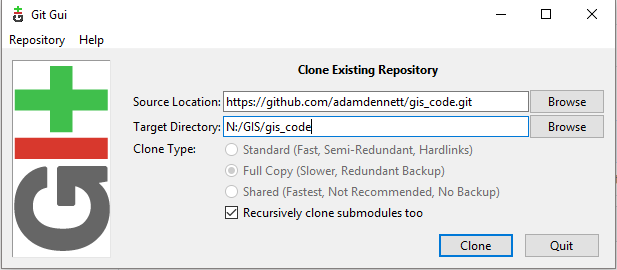
repo

1. Now in the windows start menu, go to git > GUI



gui1

1. Select 'Clone Existing Repository' and paste the link from your GitHub account into the top box and the local directory that you want to create to store your repo in the bottom box (*note, you will need to add a name for a new folder, once you have selected an existing directory*).



gui2

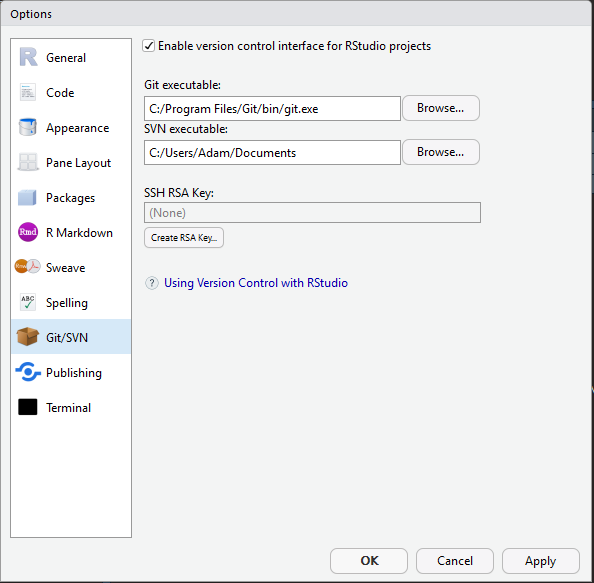
1. After a few moments, you should now be able to view a copy of your GitHub repo on your local machine. This is where you will be able to store all of your code and some other files for your reproducible research.

### Using RStudio with git

Now, as I've mentioned before, RStudio is totally bad-ass. Not only does it make R fun to use, but the lovely people who created it also built in support for things like git!

Before you can use this functionality, you need to enable it:

1. Open RStudio. In RStudio Tools > Global Options, under 'Git/SVN' check the box to allow version control and locate the folder on your computer where the git.exe file is located. and Allow Version Control for new Projects and navigate to where the git.exe file is on your computer. Click OK.

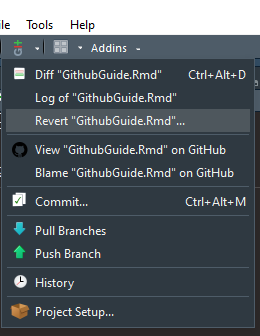


version

1. Now in RStudio, you should create a new project in an existing directory - File > New Project > Existing directory - choose your new git repository as your new project folder. You may not need to restart RStudio.

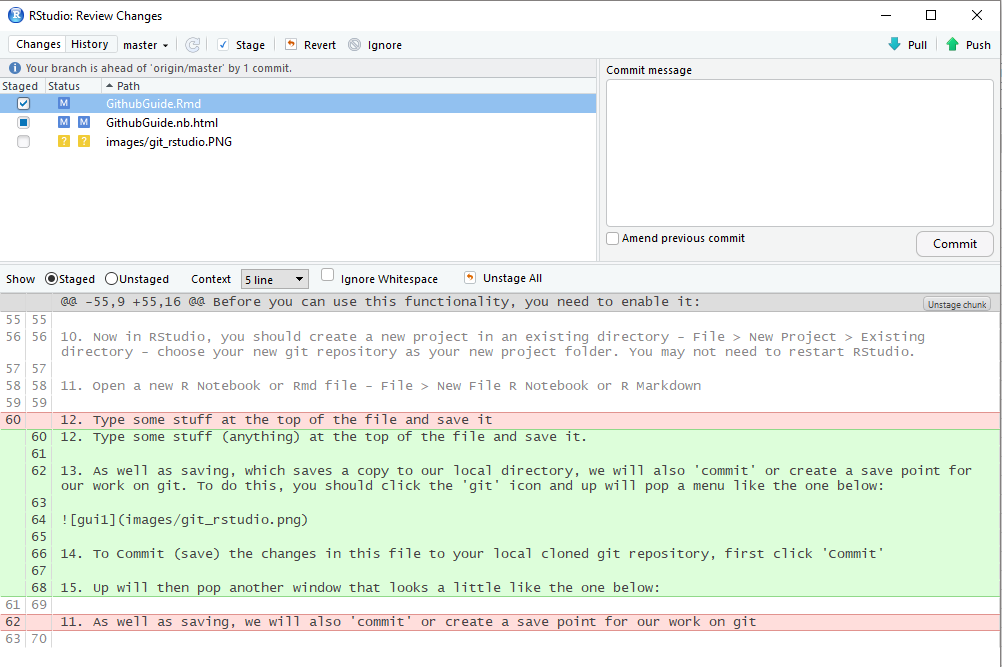
#### Saving your work to your local cloned repo

1. Open a new R Notebook or Rmd file - File > New File R Notebook or R Markdown
2. Type some stuff (anything) at the top of the file and save it.
3. As well as saving, which saves a copy to our local directory, we will also 'commit' or create a save point for our work on git. To do this, you should click the 'git' icon and up will pop a menu like the one below:



git

1. To Commit (save) the changes in this file to your local cloned git repository, first click 'Commit'
2. Up will then pop another window that looks a little like the one below:

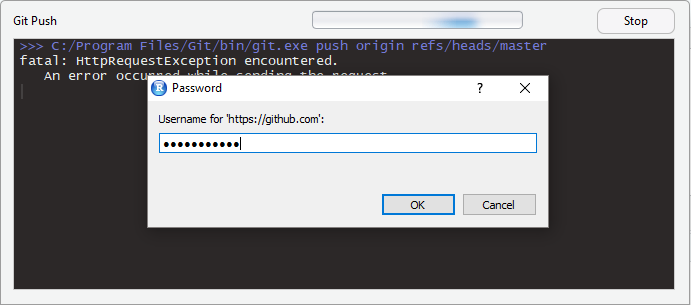


review

1. In this window, you will be able to review the differences between any previous saves you have made to your document and the current changes. If you are happy with the changes, you can then select the file and click 'commit' to save them to your current local repo. If you are not happy, then you can always revert back to a previous version that you know did work!
2. If I were you, I would save R Scripts or RMarkdown Documents using the usual save button quite regularly, and then every time you, then you can commit your changes then.

#### Pushing your changes to GitHub

1. Once you are happy with your progress, you should also then 'Push' your changes to your online GitHub repo. This is important, both for backing up your work, but also for keeping a reproducable record of your research.
2. To do this, first make sure you have committed any changes to your local cloned repo and then click the 'Push' button to whizz your code up to your master GitHub repo - you will probably be prompted to enter your github username and password to enable this...



push

#### Possible git problems

1. There are many many possible git problems. As you continute to commit and push work, especially when using different clones, you may well run into things like merge conflicts and a whole load of other problems. I've just been trying to fix a merge conflict for the last hour and eventually got there. It's not easy. It's not pretty, but as ever, there is usually a guide somewhere on the internet to help you.

For merge conflicts, try here: <https://help.github.com/articles/resolving-a-merge-conflict-using-the-command-line/>

For general github help, try here: <https://help.github.com/>

## Reproducible Research Using RMarkdown

OK, so now you have set everything up so that you can become a reproducable research ninja! All that remains is to do some reproducable research.