

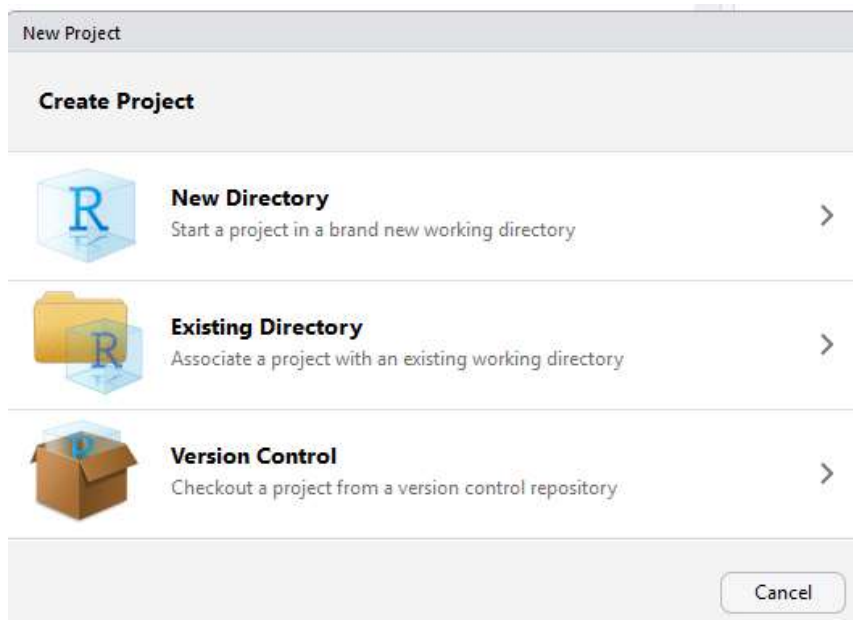
Rstudio and Version Control (Git/Github)

To use version control for your code/project in RStudio, you need the following:

- 1) Rstudio
- 2) Git (<https://git-scm.com/downloads>)
- 3) An account with Github (if you need a remote repository)

1. Create a New Project

File -> New Project (has the following options)



New Directory -> New Project -> Create New Project

Choose the directory name and path(folder) of your new Project -> **Create Project**

For this example, I created a project directory "Tutorial", you can check that your new directory is created in the folder specified.

For the sake of this tutorial, I also created a new markdown in this project (File -> New File -> R Markdown) . Rstudio has a beautiful feature to provide you the template for a markdown file, I used this and saved this file as Test.rmd and created my knitted file Test.html

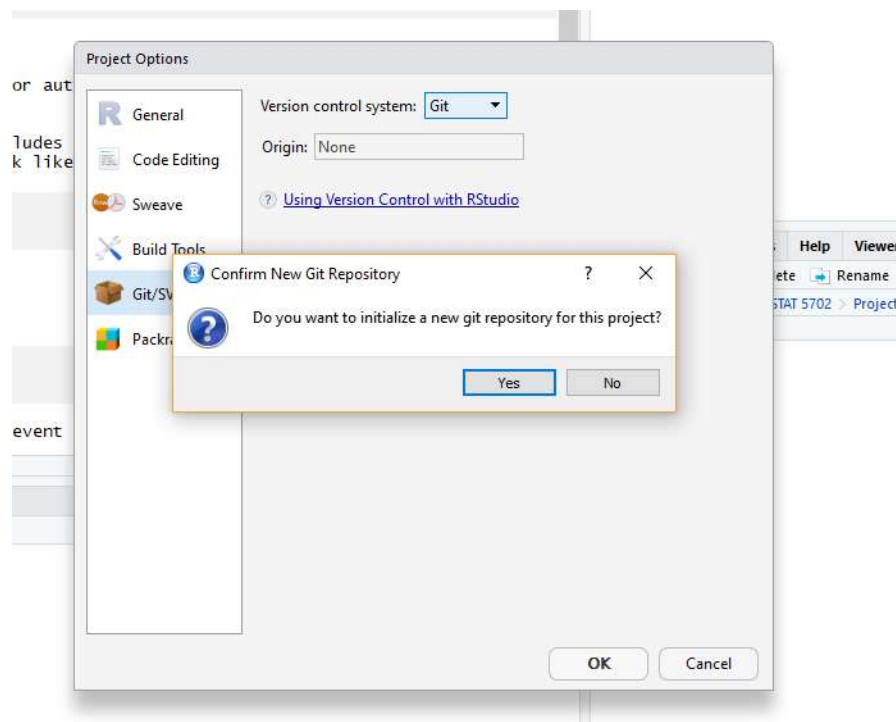
To verify your steps so far, in your project directory "Tutorial", you should be able to see the following files

- a) Test.rmd
- b) Tutorial (of type R Project)
- c) Test.html

2. Next step is to add version control to this project directory.

Tools -> Project Options -> Git/SVN -> Version Control System -> Choose "Git"

Once you choose Git, you will be prompted (see below) to initialize a new git repository for your current project. Choose **Yes**. You will be prompted to restart Rstudio.

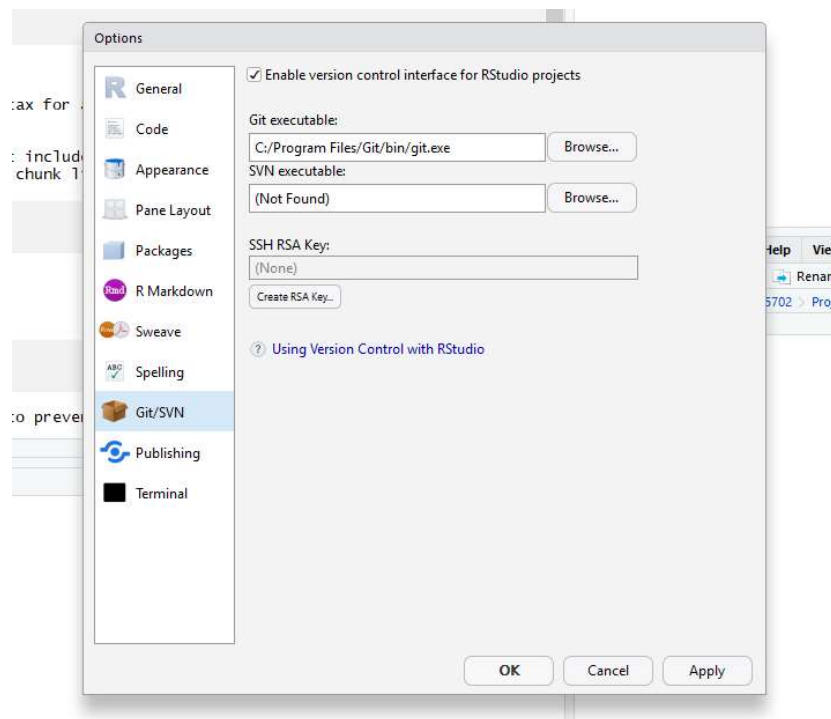


Now, your local project, has version control enabled in your local Git. If you want to enable remote version control for your project on Github (useful for collaboration and also to save an online copy of your work as backup), create a Github account and a new repository

3. Connecting RStudio and Github

In Rstudio, Go to, Tools -> Global Options -> Git/SVN -> Enable version control interface for RStudio projects

Provide the path of your local Git executable, click OK



- 1) Create your Github account, if you don't have one already
- 2) Create New Repository in Github, for the sake of this tutorial, I created a repo called tutorial
- 3) In Rstudio, click Tools -> Shell

Type the following in the shell/terminal to set up your credentials

git config --global user.name "your_github_user_name"

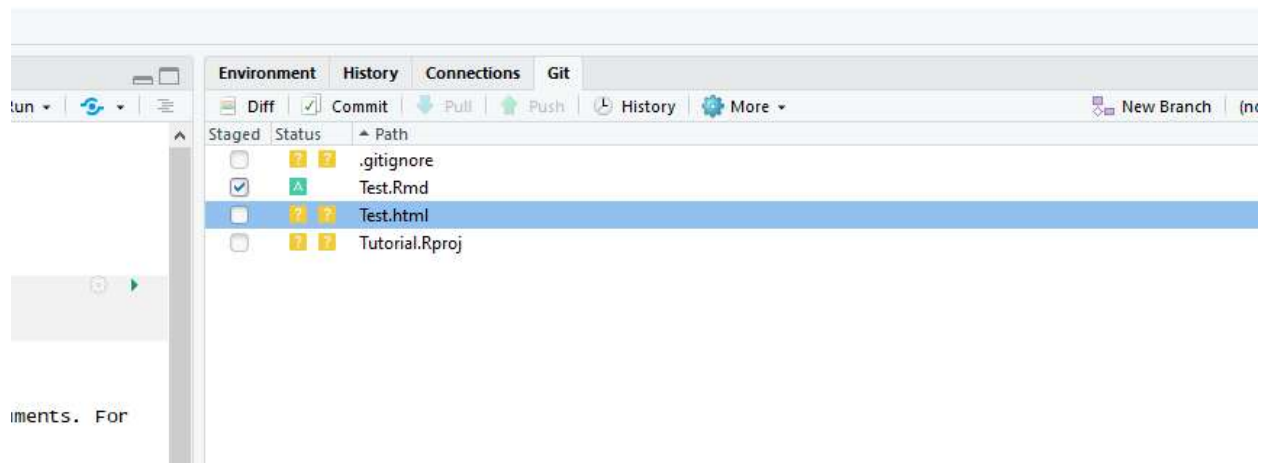
git config --global user.email "your github user email"

Link your Github Repo

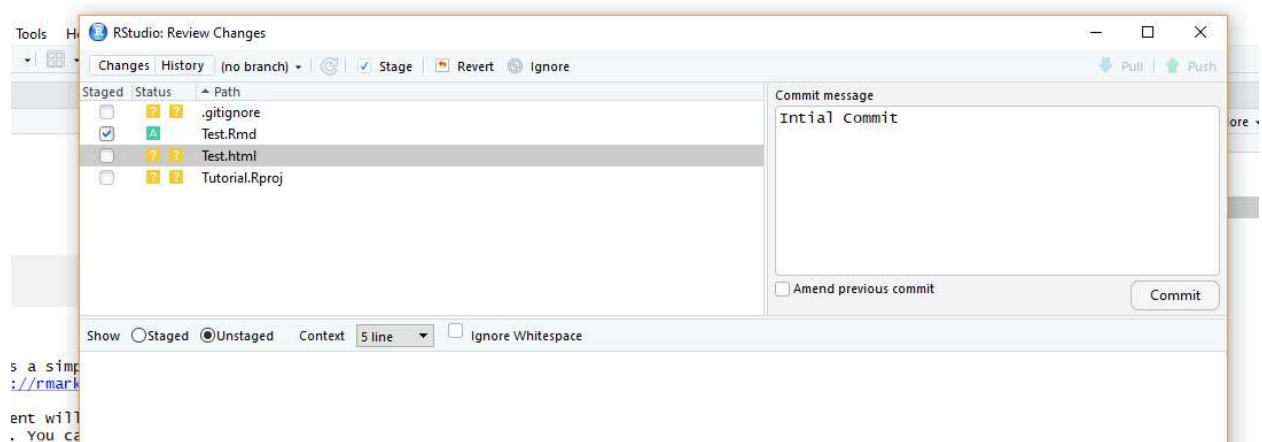
git remote add origin https://github.com/<your user name>/<your repo_name>.git

Commit your File to Local Git

In Rstudio, you will notice a Git Tab and see the files in your project, Choose the file you want to add to Git, in this example, it is Test.Rmd. Once the file is selected, Click Commit



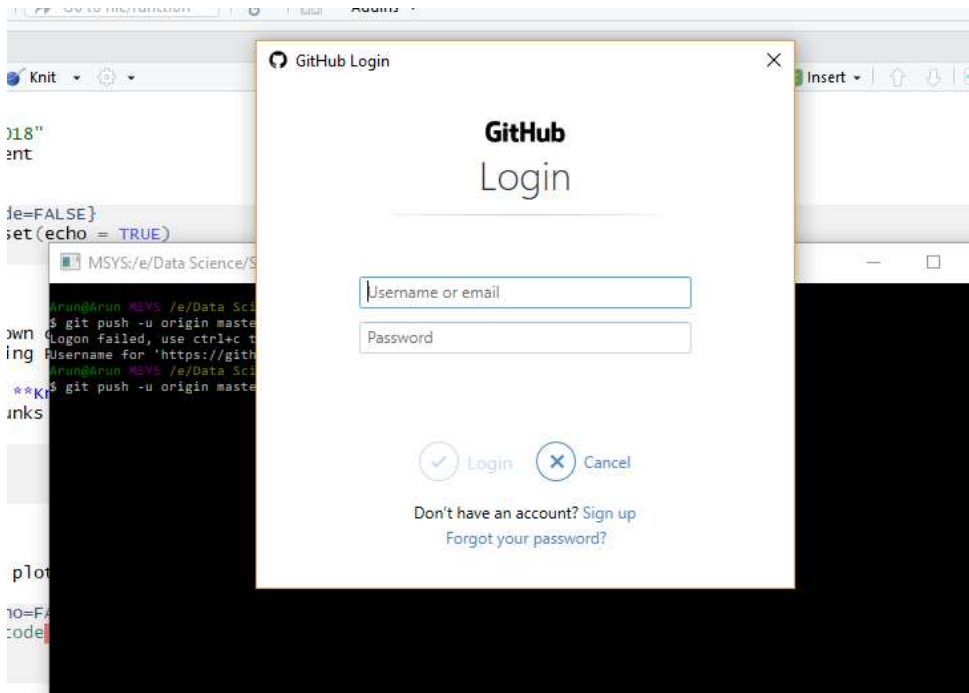
Add Commit Comments to Your file



Now to push this to your repo, on your shell, type the following (To open shell -> Tools -> shell)

git push -u origin master

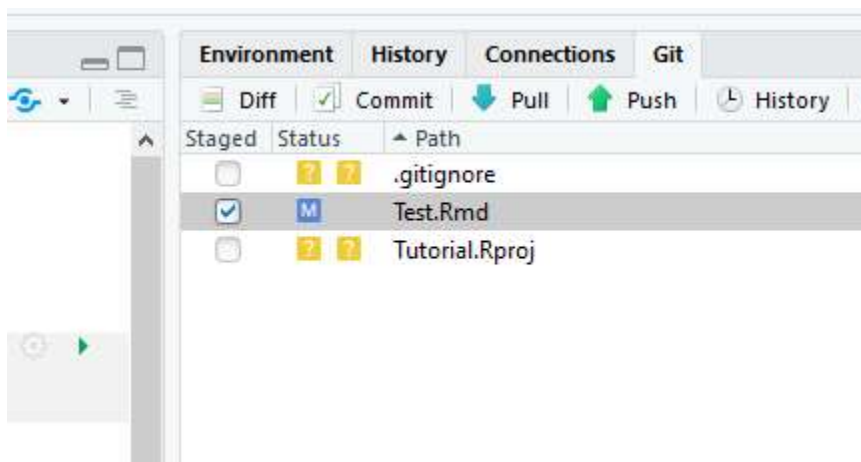
You will be prompted for your Github user name / password (one time entry)



Once the command is successful, you can check in your Github repository if the file is visible

The convenience of doing Push and Pull in Rstudio is, you don't have to type on the shell/terminal after the initial set up.

For example, I added comments in the Test.Rmd file, in the Git Tab, the Test.Rmd is shown as "M" (Modified)



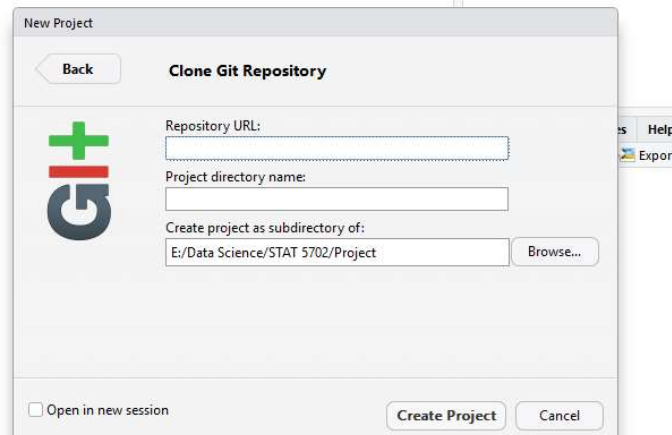
Repeat the Commit Step

Once Commit Step is completed, you can use the Push Green Button on the Gui to push the changes to Github directly. Viola!! No more commands on the terminal!!

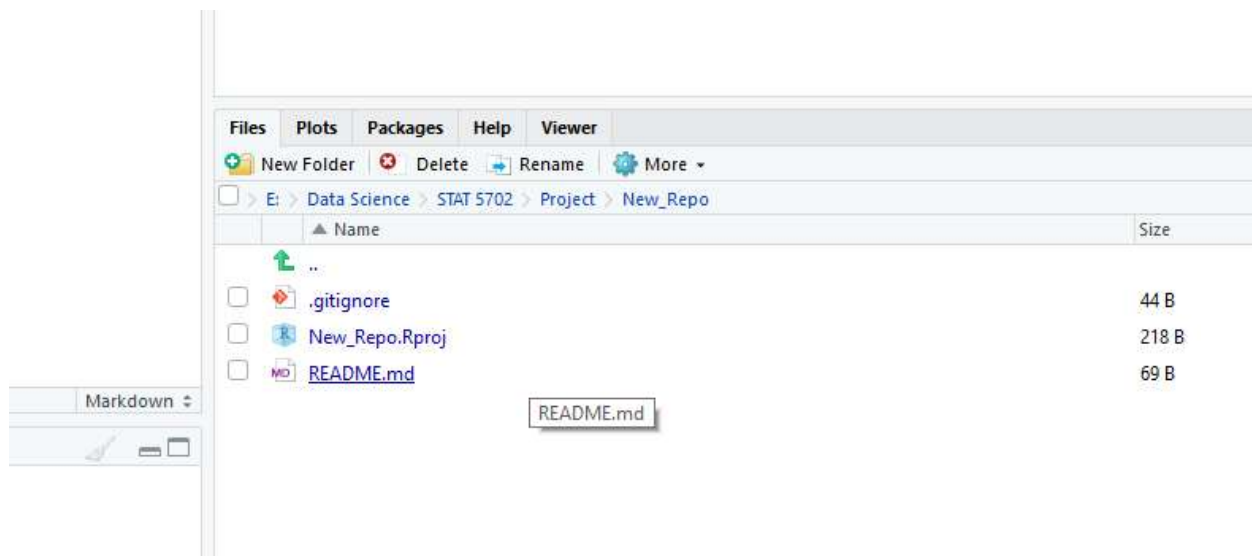
Variations:

You can also clone a remote repository into a new project in Rstudio

File -> New Project -> Create Project -> Version Control -> Create Project from Version Control (Git)



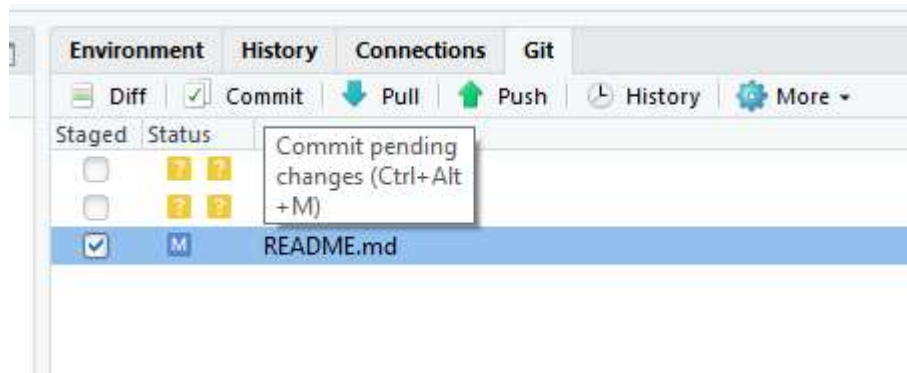
In this example, I had a Github Repository with a README.md file. After cloning this repo to my local Rstudio Project, I could see this README.md file under files tab in my Rstudio



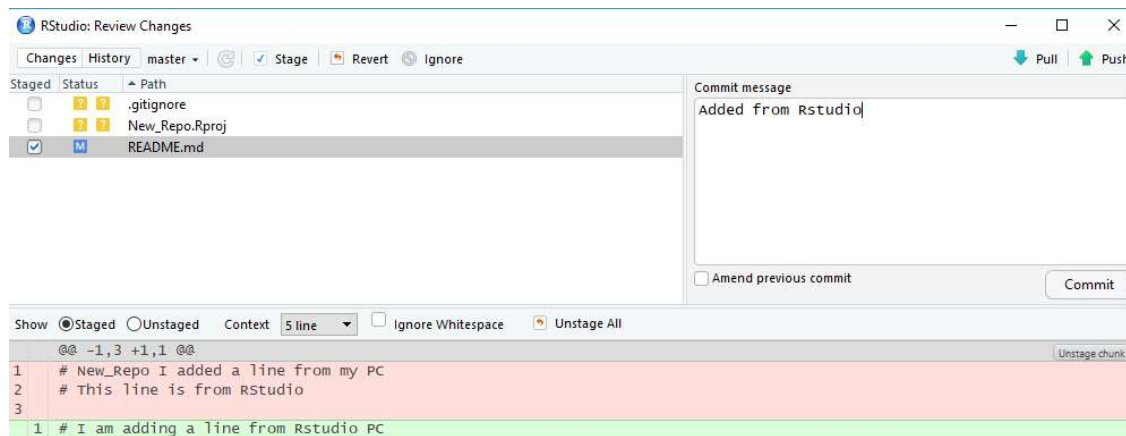
I edited the README.md file to add a line and after saving the file, I observe that under Git tab, the README.md shows “M”(Modified)



Choose README.md under “staged” and click Commit.

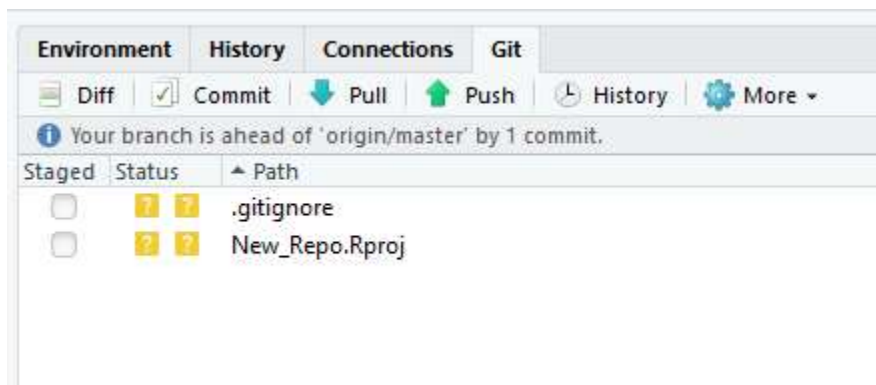


After clicking Commit, Rstudio will show the differences made to README.md, add your comments and confirm Commit .



```
Git Commit
Close
>>> C:/Program Files/Git/bin/git.exe commit -F C:/Users/Anita/AppData/Local/Temp/R
[master dc301dd] Added from Rstudio
1 file changed, 1 insertion(+), 3 deletions(-)
```

Use Push button to push the changes to Github repository, once completed, check your github repository folder and check the file to see if your changes/edit are reflected



Rstudio provides an easy interface to push/pull files/project from Github without having to remember the Git syntax.

Notes:

- In case of any issues, you can always open shell/terminal through Rstudio. Tools -> Shell . One advantage is git automatically opens in the working directory.
- I also use “git status” on the shell, when I am confused about what I have clicked on the Rstudio. Git status gives current status of the working directory, it provides information on what files have been staged, files not tracked etc
- I ran into some trouble, using multiple github accounts with Rstudio. I created a new Github with my .edu email account, but Rstudio kept using my older github account even after I modified git config user.name and user.email. Apparently, my PC(windows) stores user credentials for my old github account, I had to remove it manually (control panel -> user accounts -> credential manager -> Manage Windows Credentials -> Generic Credentials -> (Remove the ones for github/git) I am not sure if this problem props up with Mac/linux
- Instead of typing the github account credentials on shell, you can also map your Rstudio SSH RSA key into your Github account to enable authentication. (In Rstudio. Tools -> Global options -> Git/SVN -> Create RSA SSH Key). View and Copy-Paste this key into your Github Account Settings -> Add SSH keys (refer link 2 in References)
- If you are working in a group for your project, it is always better to pull(update your local project with latest changes) before you push your updates. This is a common Git practice to avoid over-writing updates in your Git Repository

References:

<https://support.rstudio.com/hc/en-us/articles/200532077?version=1.1.419&mode=desktop>

<https://www.r-bloggers.com/rstudio-and-github/>