RStudio GitHub Setup

Sanjeev Kumar

10/13/2020

RStudio + GitHub

So let's say that you have to work on a group project for your data analytics class. You have a dataset for the project - and you are going to write a project report as a RMarkdown file and build data analysis models using R. The problem - you need to do this collaboratively as a team of five, globally distributed, across time zones, students.

Clearly, each student working independently and then merging code/files together is sub-optimal. You are bound to repeat things, do stuff that will not match other students, have different approaches that may not merge well and so on. Sharing files over email, google drive etc just introduces needless complexity. What you need to do is what software development teams all over the world so - collaborate over a cloud based versioning and remote collaboration plaforms like GitHub.

> This is your absolutely bare-bones, just enough to get you started, beginners guide to how to collaborate on R based group projects using RStudio and GitHub.

Let's Begin with the Building Blocks

Let's get the basic out of the way:

- You have installed R. If not, you can get it from here: https://cran.rstudio.com/.
- You have installed RStudio. If not, you can get it from here: https://rstudio.com/products/rstudio/download/. Go for the free, Desktop version.
- You have installed Git in your machine. If not, get the download for your operating system here: https://git-scm.com/downloads. You can install git with all the default options. We will use git only through the RStudio GUI so most of the options are not relevant for us.
- You have created an account on GitHub.com. If not, go create one here: https://github.com/join

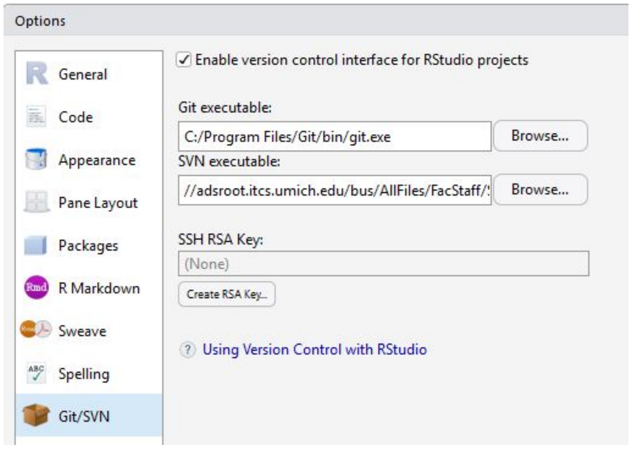
Alright - now you are ready.

Configure RStudio for Git/GitHub

We need to make sure that our RStudio session is set up to work with Git/GitHub. Go to Tools > Global Options > Git/SVN. Ensure that:

- Option `Enable version control interface for RStudio Projects` is **checked**.
- Git Executable is showing the path to the your installed Git.
- SVN and RSA options are not relevant for us - you can ignore those.

In my RStudio session, the dialog box looks like the following:



Options

☑ Enable version control interface for RStudio projects

Git executable:
C:/Program Files/Git/bin/git.exe

SVN executable:
//adsroot.itcs.umich.edu/bus/AllFiles/FacStaff/

SSH RSA Key:
(None)

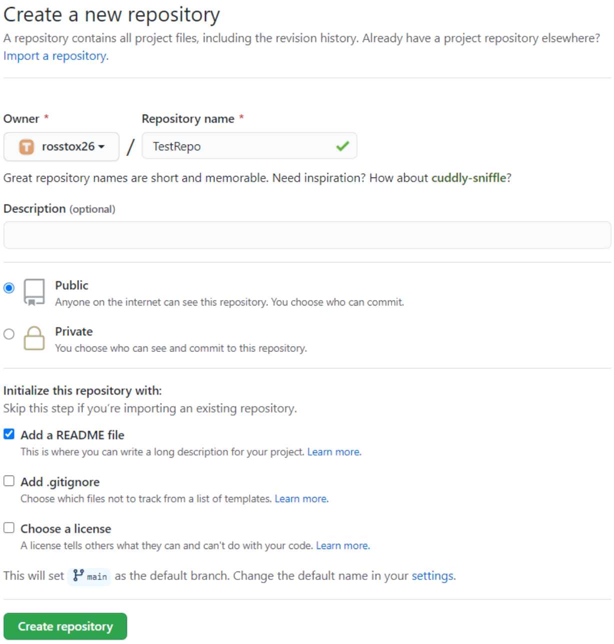
Figure 1: Global Options Setup for RStudio

Easiest Path to a GitHub + RStudio Integration

It is important to note that RStudio's Git/GitHub functionality is tied to the idea of a "RStudio Project". Each project is a self contained unit that lives in a specific folder. The project folder has a git repository that can be synced with a cloud based remote repository. The easiet path to creating a RStudio Project that is synced with GitHub is follow the step by step process explained below.

Create an Empty Repository in GitHub

- Sign into GitHub and click the button for creating a new repository



Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? Import a repository.

Owner * / Repository name *
rosstox26 / TestRepo ✓

Great repository names are short and memorable. Need inspiration? How about cuddly-sniffle?

Description (optional)

● Public
Anyone on the internet can see this repository. You choose who can commit.

○ Private
You choose who can see and commit to this repository.

Initialize this repository with:
Skip this step if you're importing an existing repository.

☑ Add a README file
This is where you can write a long description for your project. Learn more.

☐ Add .gitignore
Choose which files not to track from a list of templates. Learn more.

☐ Choose a license
A license tells others what they can and can't do with your code. Learn more.

This will set `main` as the default branch. Change the default name in your settings.

Figure 2: Create Empty Repository in GitHub

- Provide a repository name. I am using the name `TestRepo` in the example image above

- Choose either `Public` or `Private` as your repository's access level. If you are unsure, choose `Public`
- Choose the option for `Add a Readme File` so that you have some material in your repo and its not *completely* empty