GitHub

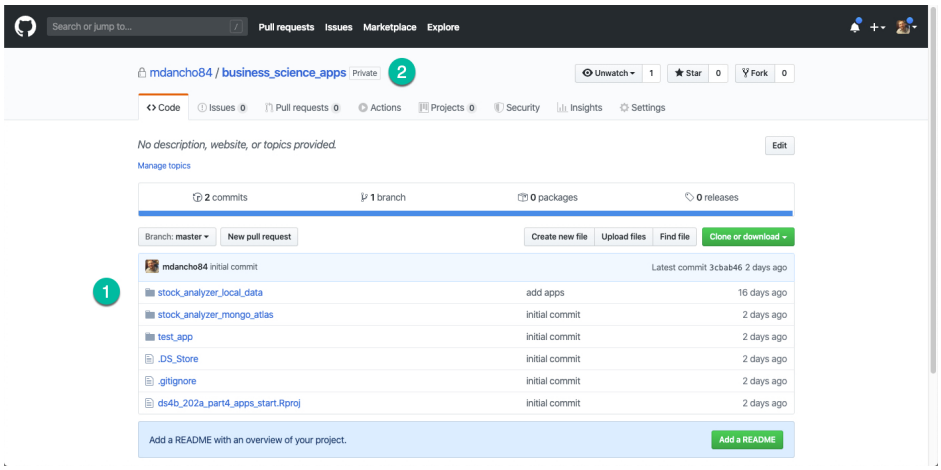
Business Science

11/26/2019

Contents

GitHub

This document covers the using GitHub (https://github.com/) for a version-controlled software workflow connecting both local and remote repositories. At the end of this, you will have connected a local and remote repository containing your Stock Analyzer application.



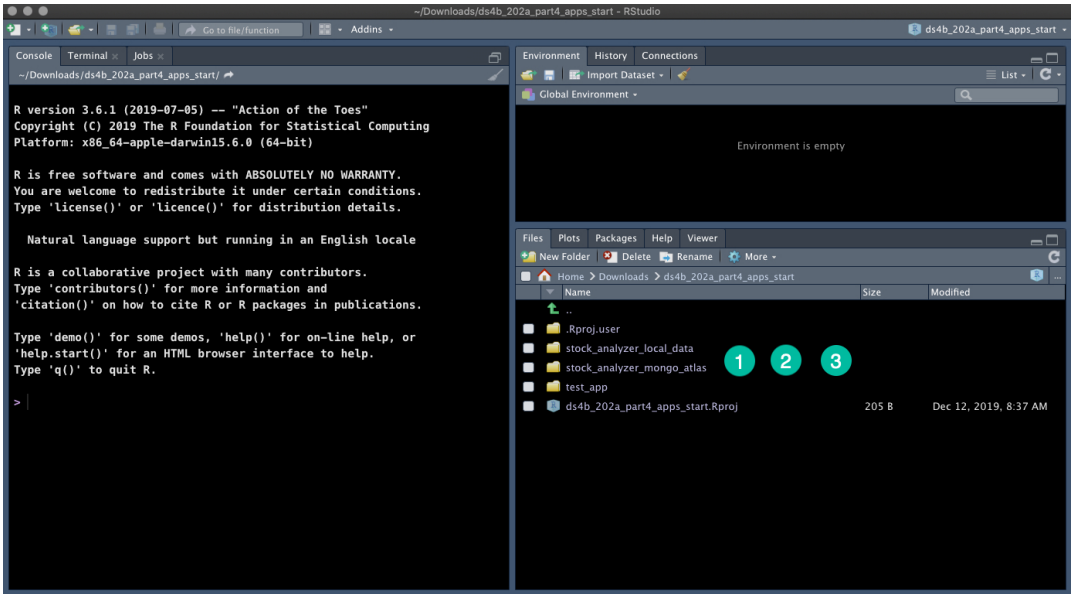
Prerequisites

Download the zip file (`ds4b_202a_part_4_apps_start`). Unzip, and open the R Project, `ds4b_202a_part_4_apps_start.Rp` to get started. You will see 3 application directories:

1. Test App - `ds4b_202a_part_4_apps_start/test_app`
2. Stock Analyzer (Local Data Storage) - `ds4b_202a_part_4_apps_start/stock_analyzer_local_data`
3. Stock Analyzer (MongoDB Atlas) - `ds4b_202a_part_4_apps_start/stock_analyzer_mongo_atlas`

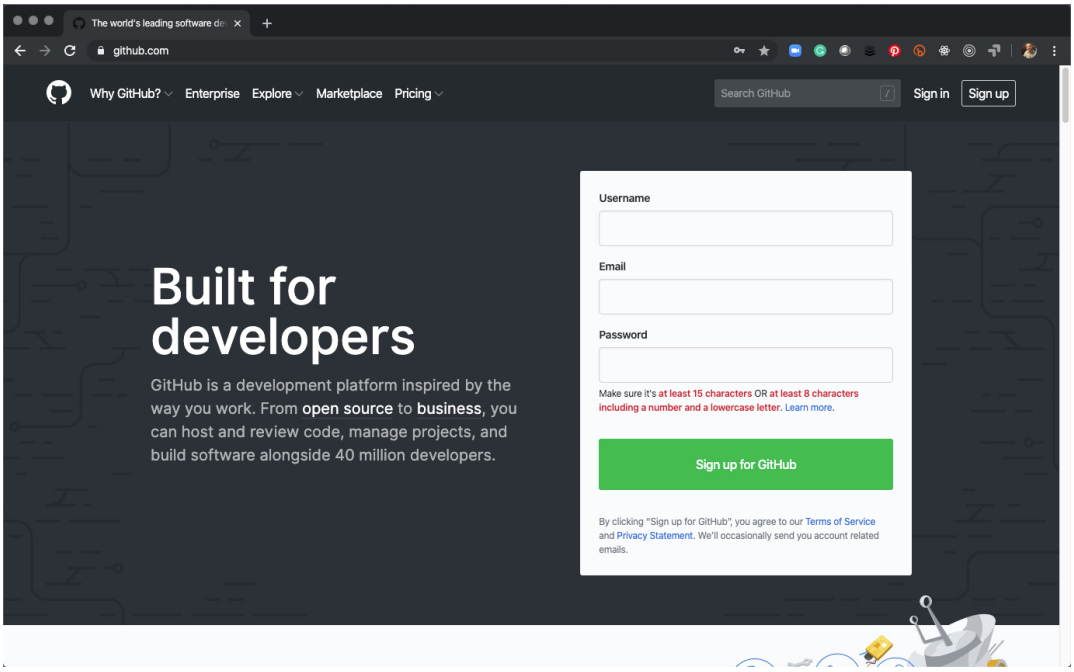
Test each of the applications out.

Important Note - You will need to adjust the `config.yml` file for the Stock Analyzer that connects to MongoDB Atlas



Create a Git Account

If you don't have a GitHub account, create one at GitHub (https://github.com/).



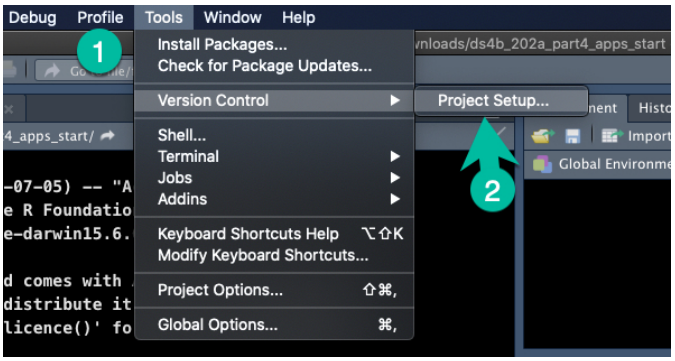
Check Git Installation on Your Local Computer

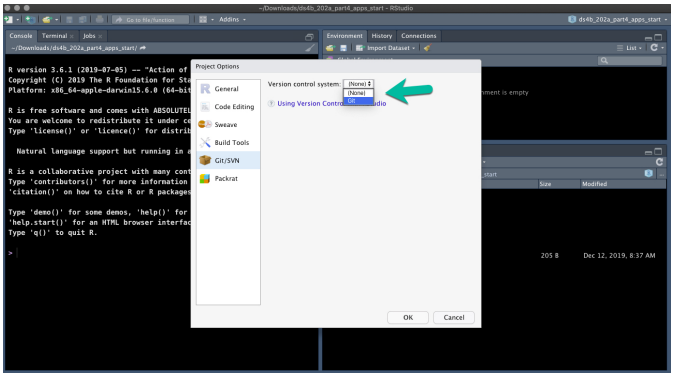
Check to make sure you have `git` installed with `git --version`.

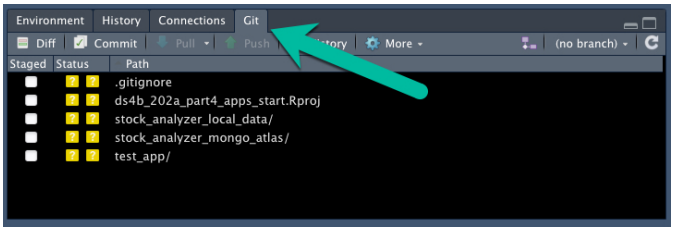
Local Repository Setup

Step 1 - Add Git Version Control to Your Project

Add version control. When RStudio restarts, you will have a "Git" pane.



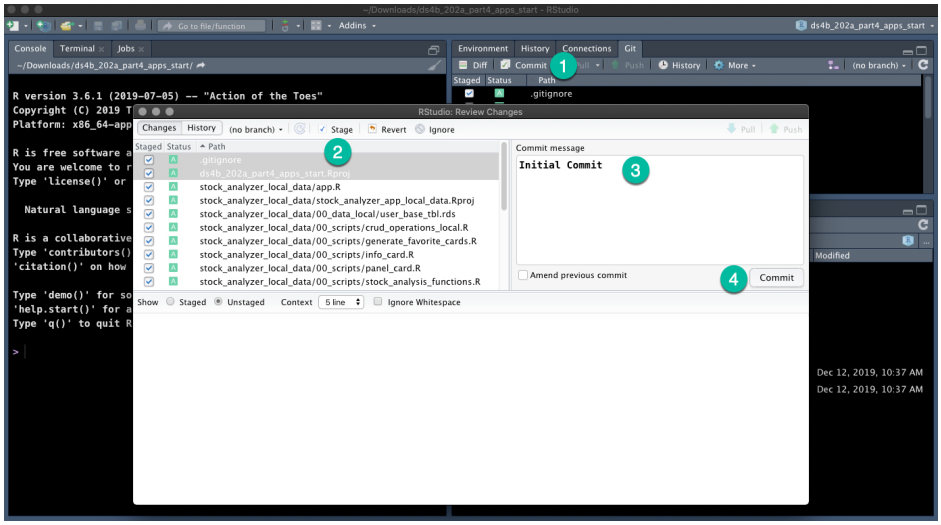




Step 2 - Make your Initial Commit

This moves the files from working to local repository, or in other words takes your first snapshot.

1. Click "Commit" in the Pane
2. "Stage" each of the directories and files
3. Add a "Commit" Message
4. Click "Commit Button"



Remote Repository Setup

Next, we'll create a remote repo on GitHub to house the files.

Important Note - This will be a *private* repo because the Stock Analyzer MongoDB Atlas app contains a config.yml file that includes your Connection Information. We do not want others outside of our organization to see this information.

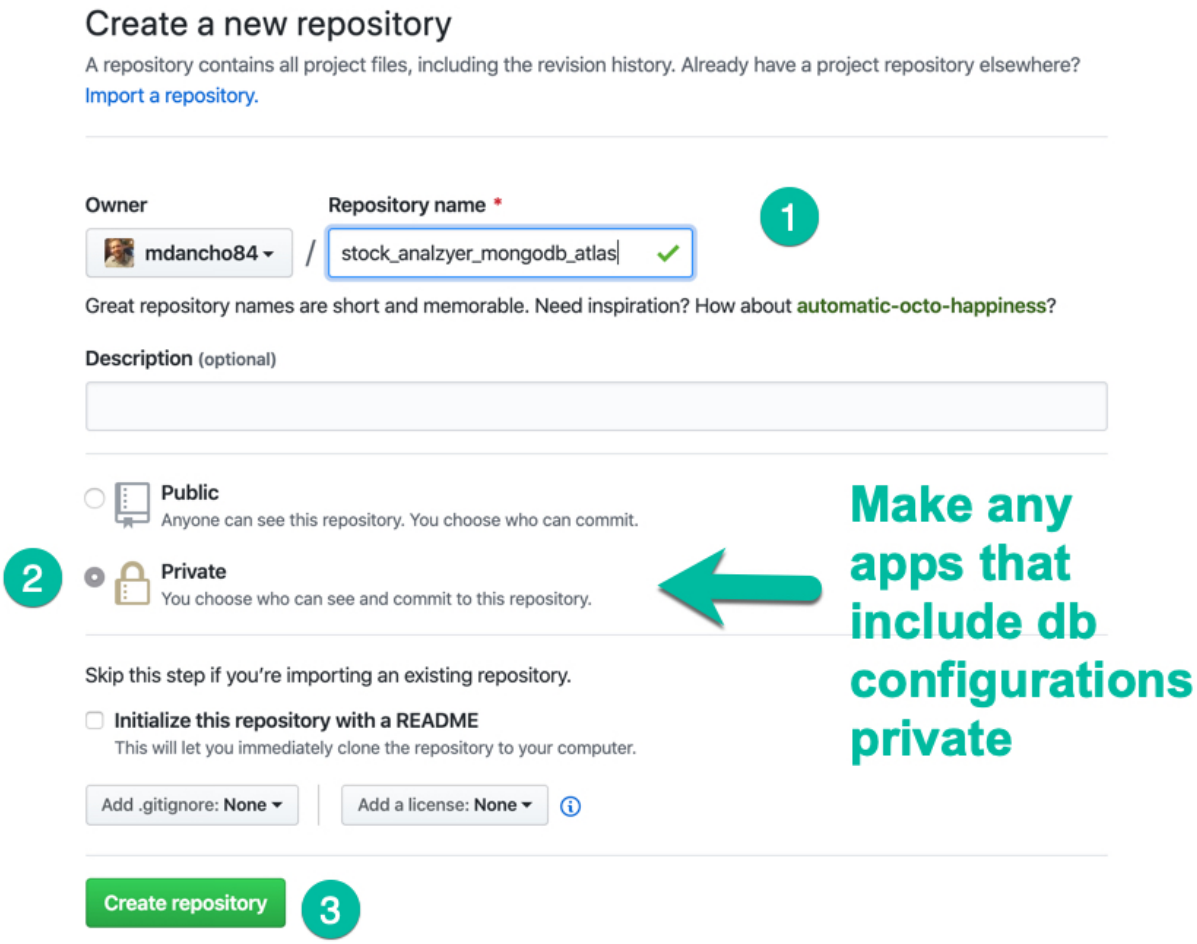
Step 1 - Sign In & Go To Repositories

Step 2 - Make a Private Repo

We can make Private Repos via GitHub.

1. Name it something related to the "Stock Analyzer Project".
2. Select "Private" - Never including `config.yml` files in publically facing software.

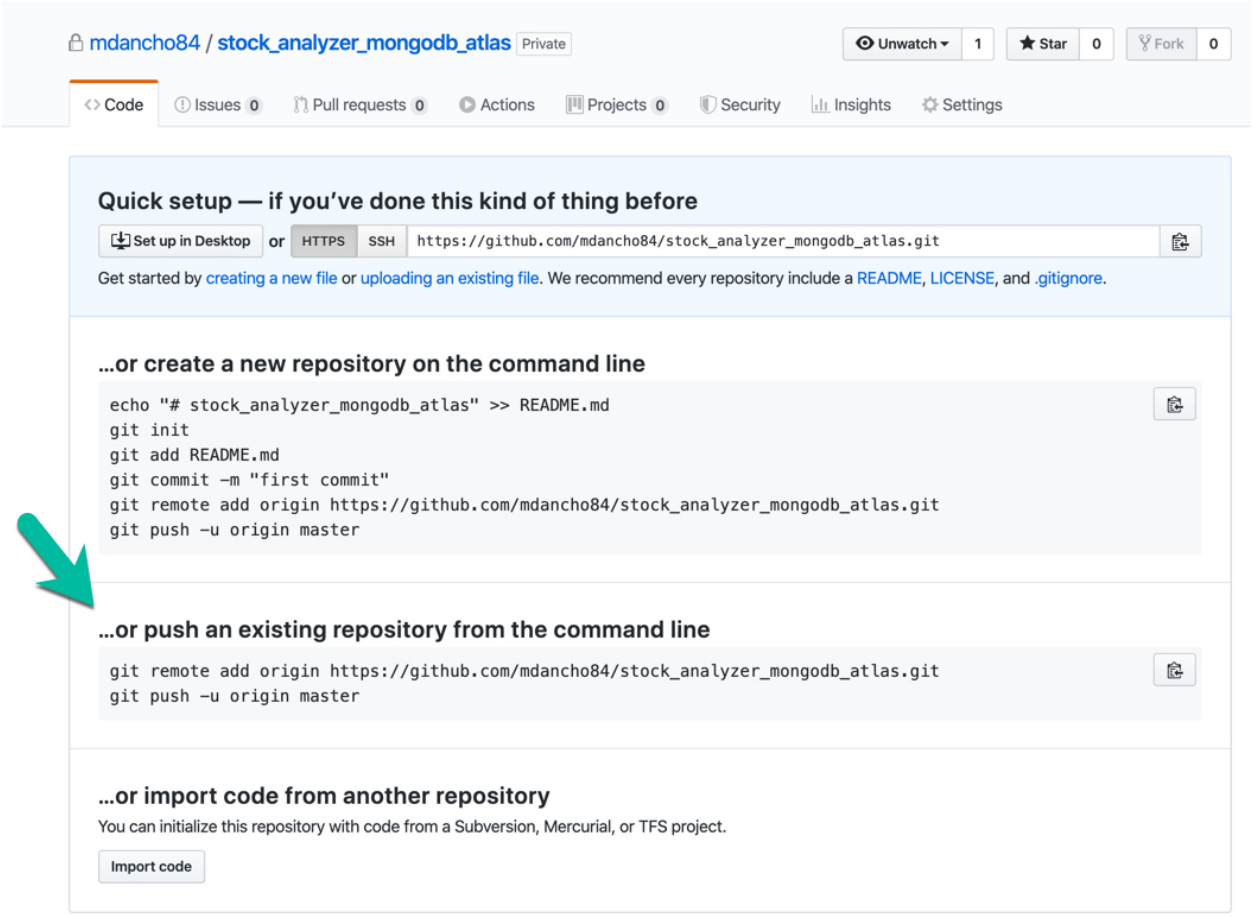
You will then be given setup instructions for linking the remote and local repo.



Step 3 - Connect Local to Remote & Push

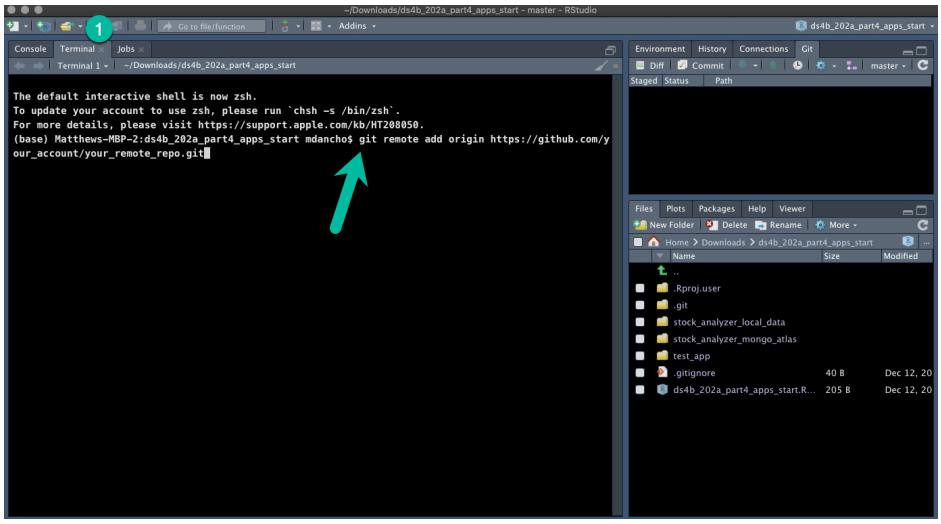
Follow the instructions to push an existing repository from the Command Line.

1. Copy the code to add the remote repository to our local repository

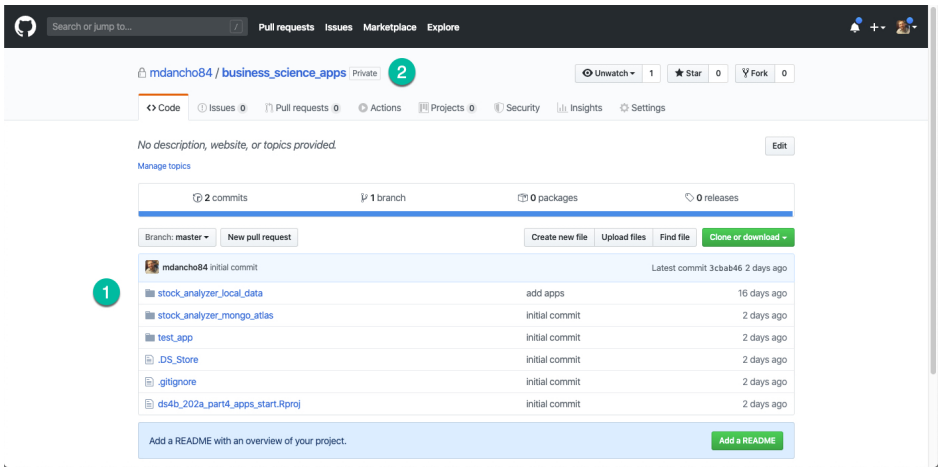


2. We can use RStudio's terminal to create the link between Local and Remote.

   - `git remote add origin https://github.com/your_account/your_remote_repo.git` - Links your remote and local repos
   - `git push -u origin master` - Pushes all of your commited files in your Local Repo to the Remote Repo



3. Check the Remote (GitHub) repository by refreshing the page. You should now see your files in a Private Repo. *Note - You can also change the name of your Repo if you don't like the previous name.*



Wrapup

Congrats. You now have successfully:

1. Created a Local Repository
2. Created a Remote Repository and linked the local to the remote