

## Join GitHub today

Dismiss

GitHub is home to over 36 million developers working together to host and review code, manage projects, and build software together.

Sign up

Branch: master ▼

rstudio\_test / README.md

Find file

Copy path



bioboot Update README.md

c951667 on Feb 13

1 contributor

123 lines (61 sloc) | 6.23 KB

Raw

Blame

History



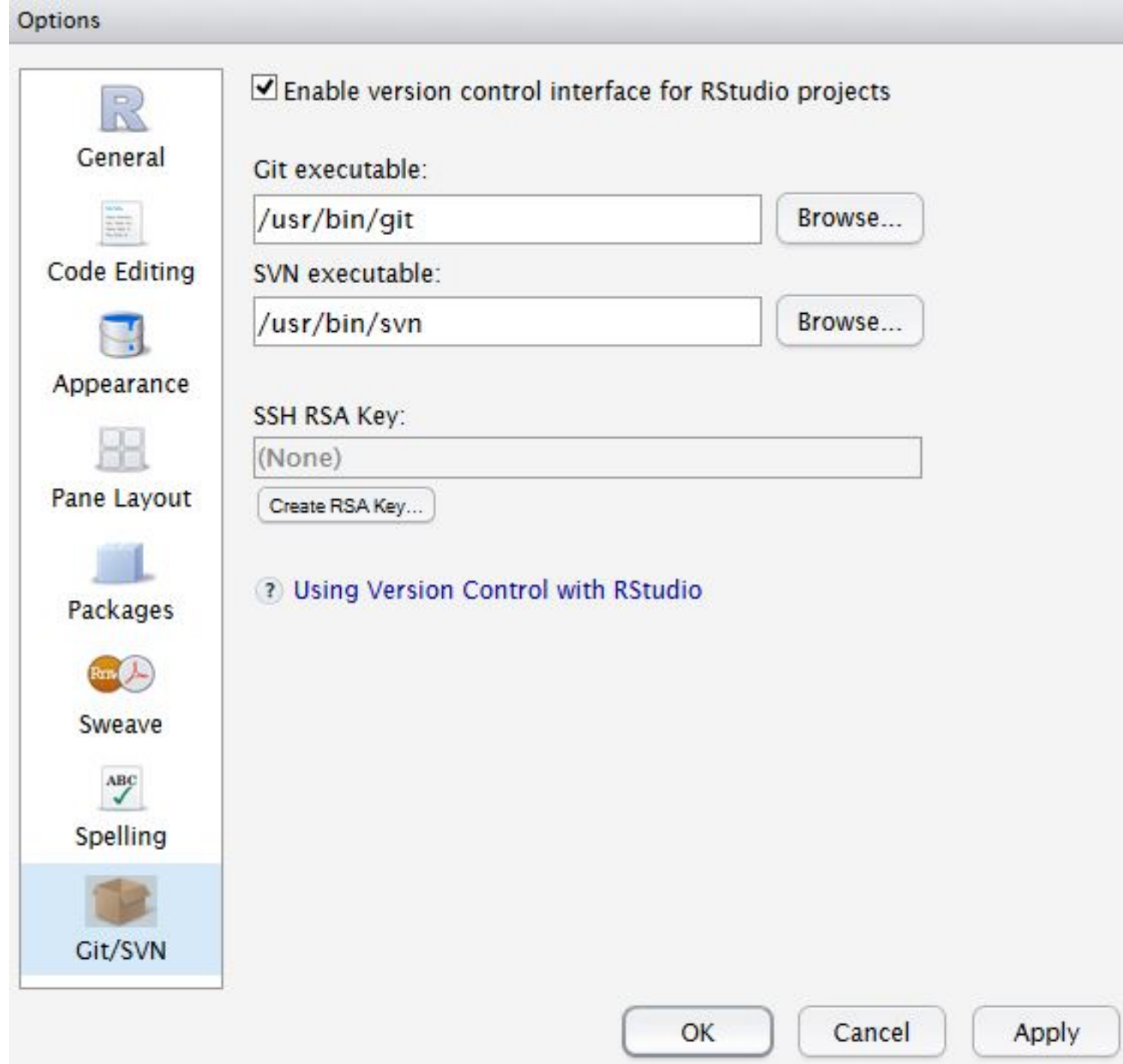
# Using Git and GitHub with RStudio

## 1. Setting up Git with RStudio

Prior to using Git with RStudio you should install git using the appropriate method for your type of computer platform. See: <http://git-scm.com/downloads>

Once you have git installed you'll need to activate it within RStudio by following these steps in RStudio itself:

Click **Tools > Global Options** then select the **Git/SVN** tab and click **Enable version control interface for RStudio projects**.



Please ensure that the path to the Git executable is correct. To do this click **Tools > Shell...** then at the shell prompt type:

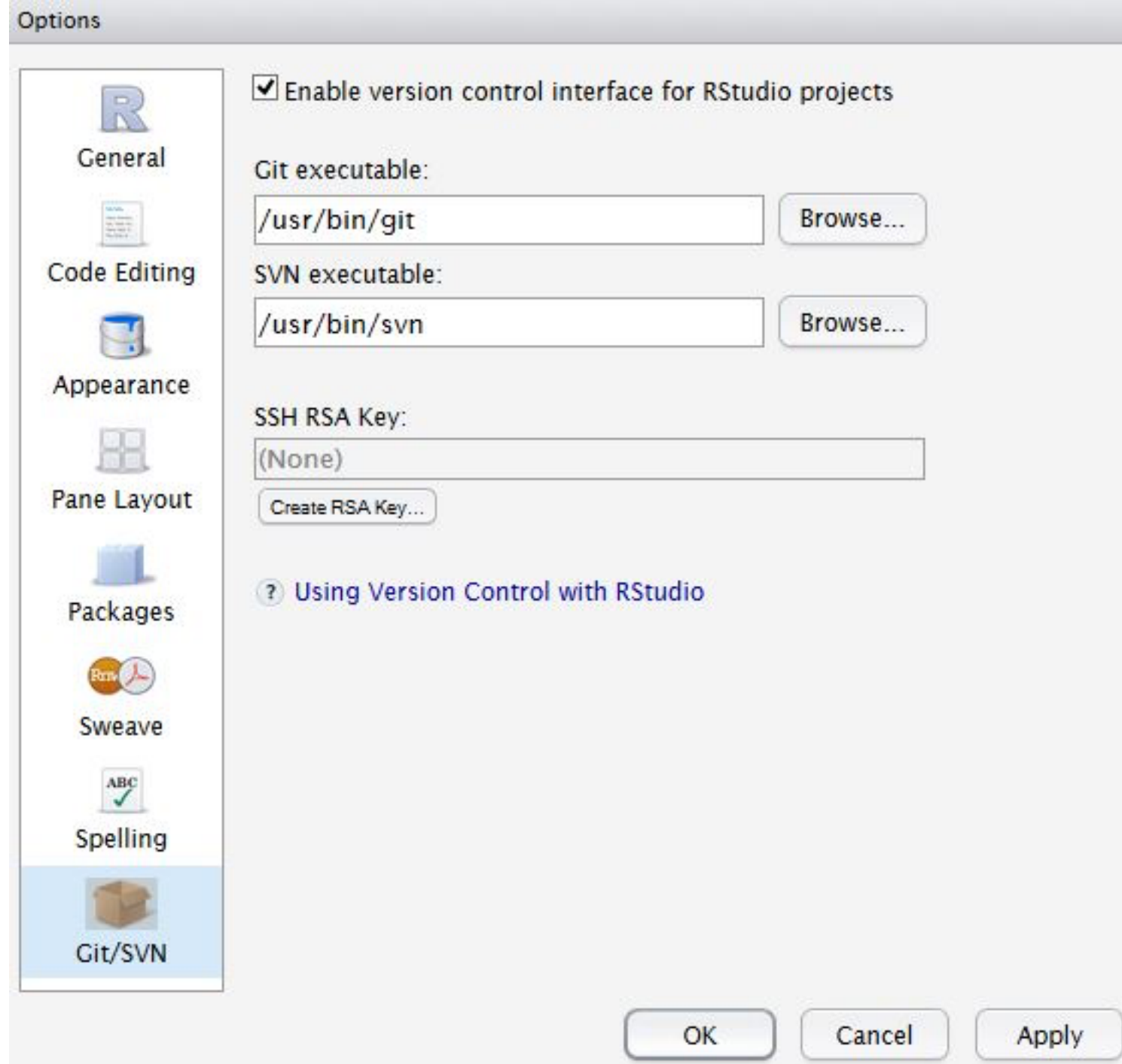
```
which git
```

The the returned path should match that in the RStudio Options window. If not please change the RStudio path.

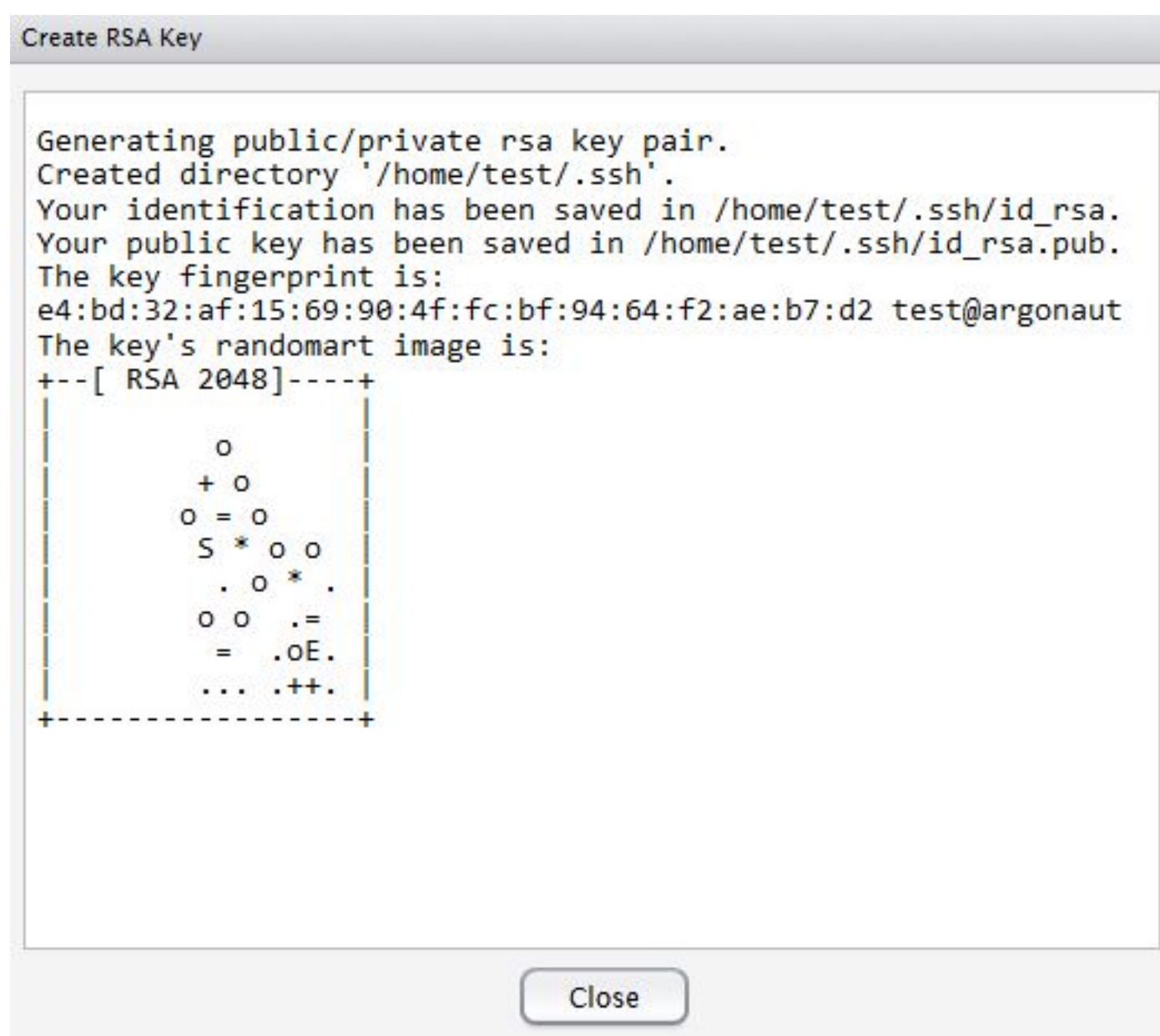
This is particularly important on Windows where it may not default correctly (e.g. C:/Program Files (x86)/Git/bin/git.exe). On Mac it may be either /usr/bin/git or /usr/local/bin/git.

## 1B. Optional, enable passwd-less push and pull to GitHub

Click **Tools > Global Options** then select the **Git/SVN** tab and make sure the **Enable version control interface for RStudio projects** is clicked.



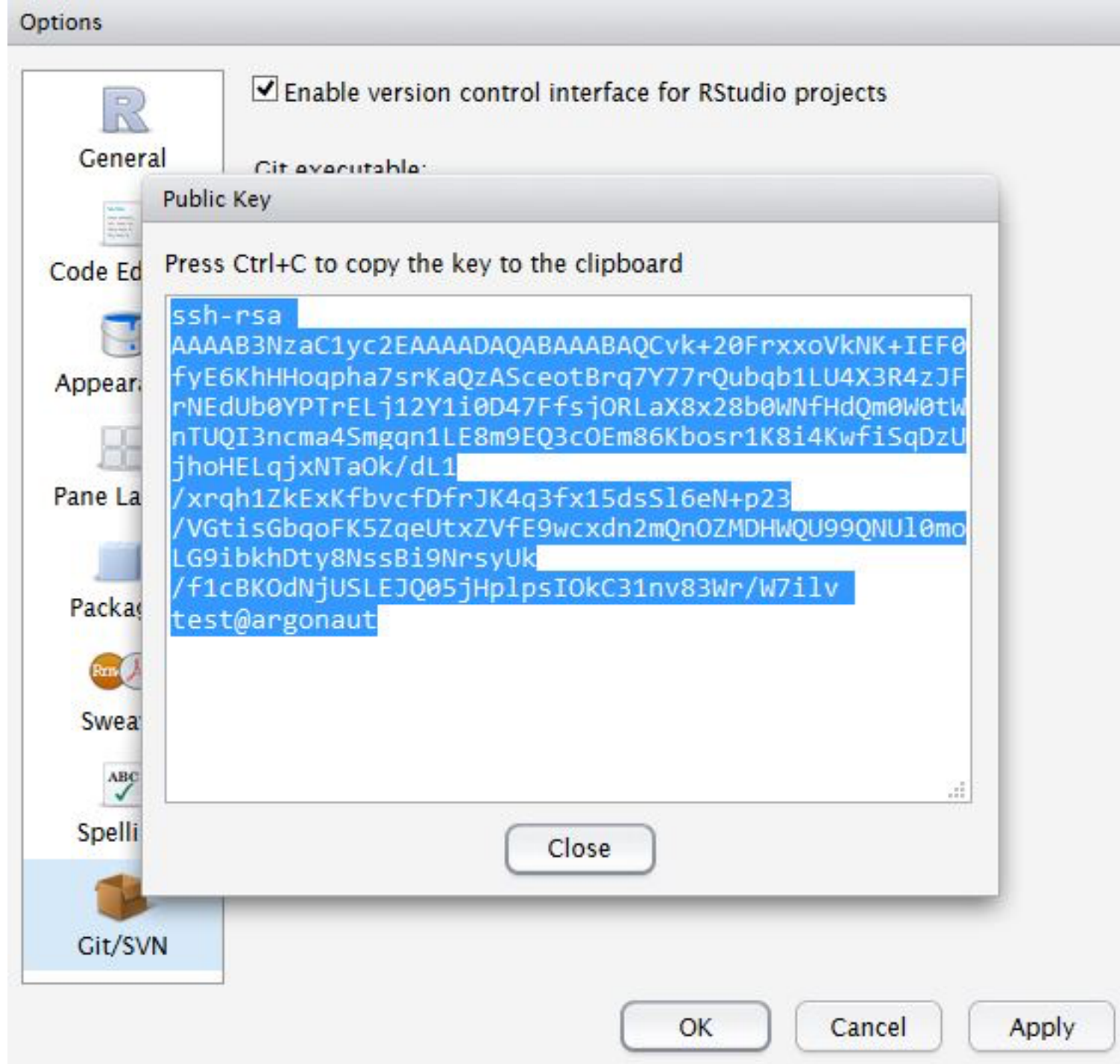
Now hit, **Create RSA Key**.



Close this window.

Click, **View public key**, and copy the displayed public key.





This text should now be pasted into GitHub. On GitHub open your account settings and click the SSH keys tab. Click Add SSH key. Now paste in the public key you have copied from RStudio.

Personal settings

Profile

Account settings

Emails

Notification center

Billing

SSH keys

Security

Applications

Personal access tokens

Repositories

Organizations

Need help? Check out our guide to [generating SSH keys](#) or troubleshoot [common SSH Problems](#)

SSH keys

Add SSH key

This is a list of SSH keys associated with your account. Remove any keys that you do not recognize.

GitHub for Windows - ADMIN-CKU3T4WST

73:d4:b6:72:f2:f3:08:1a:7a:ad:b8:45:f1:c7:a4:6c

Added on 26 Jan 2013 — ⓘ No recent activity

Delete

GitHub for Windows - VaioZ\_EMH

8b:da:41:ae:84:95:41:10:a0:15:e0:c8:09:71:aa:90

Added on 26 Feb 2013 — ⓘ No recent activity

Delete

GitHub for Windows - VaioZ\_EMH

a5:2b:50:3a:9f:d7:9a:e8:7d:47:f9:47:13:5d:45:6a

Added on 18 Jul 2013 — ⓘ No recent activity

Delete

Argonaut

c7:28:82:38:0e:5b:93:15:55:0a:6d:20:81:26:71:a2

Added on 30 Dec 2014 — Last used within the last 3 days

Delete

Vaio 2015

d8:9b:e0:66:ea:dd:60:6d:99:bb:1a:00:d5:89:34:47

Added on 30 Apr 2015 — Last used within the last 5 days

Delete

## 2. Configuring Git

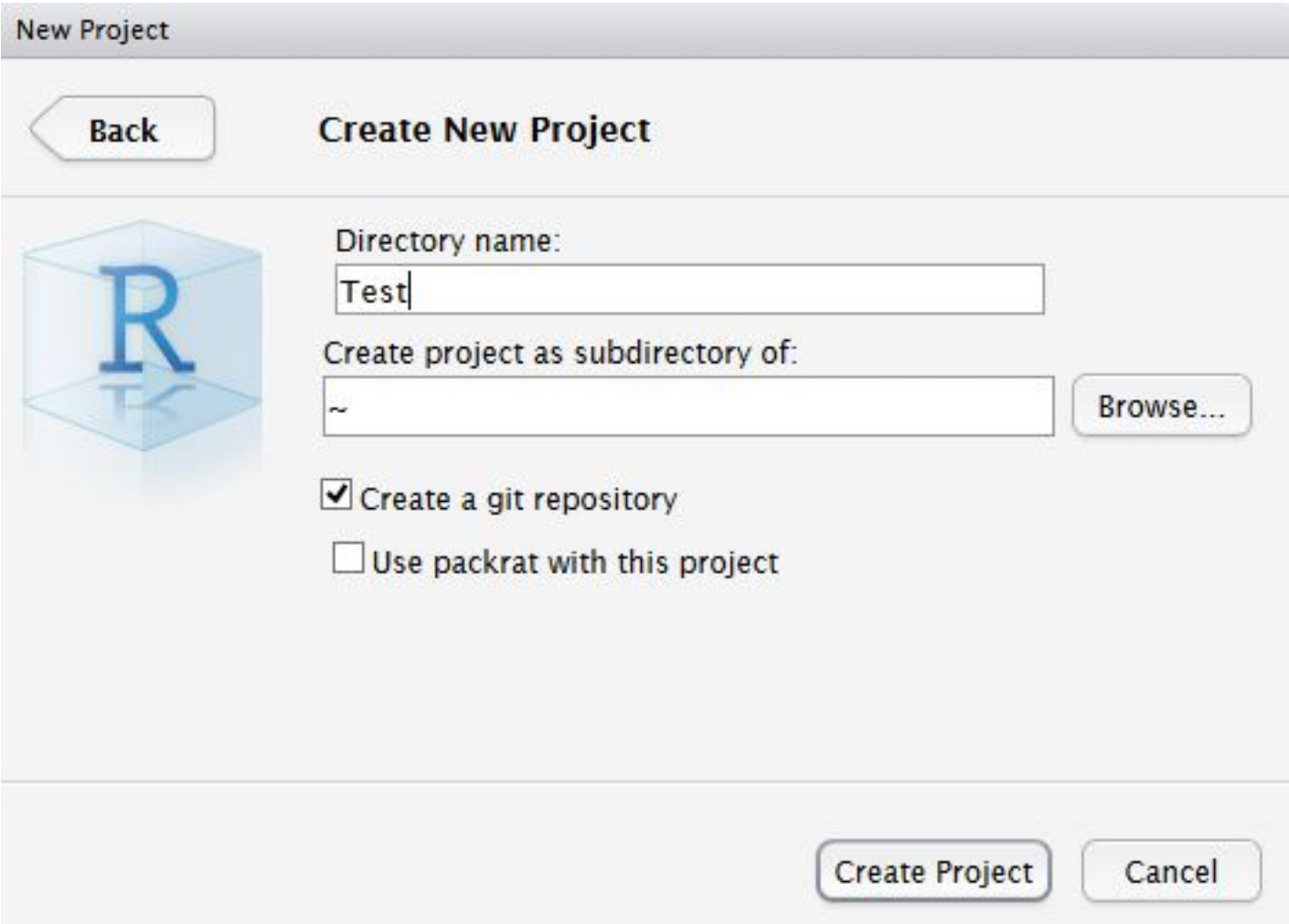
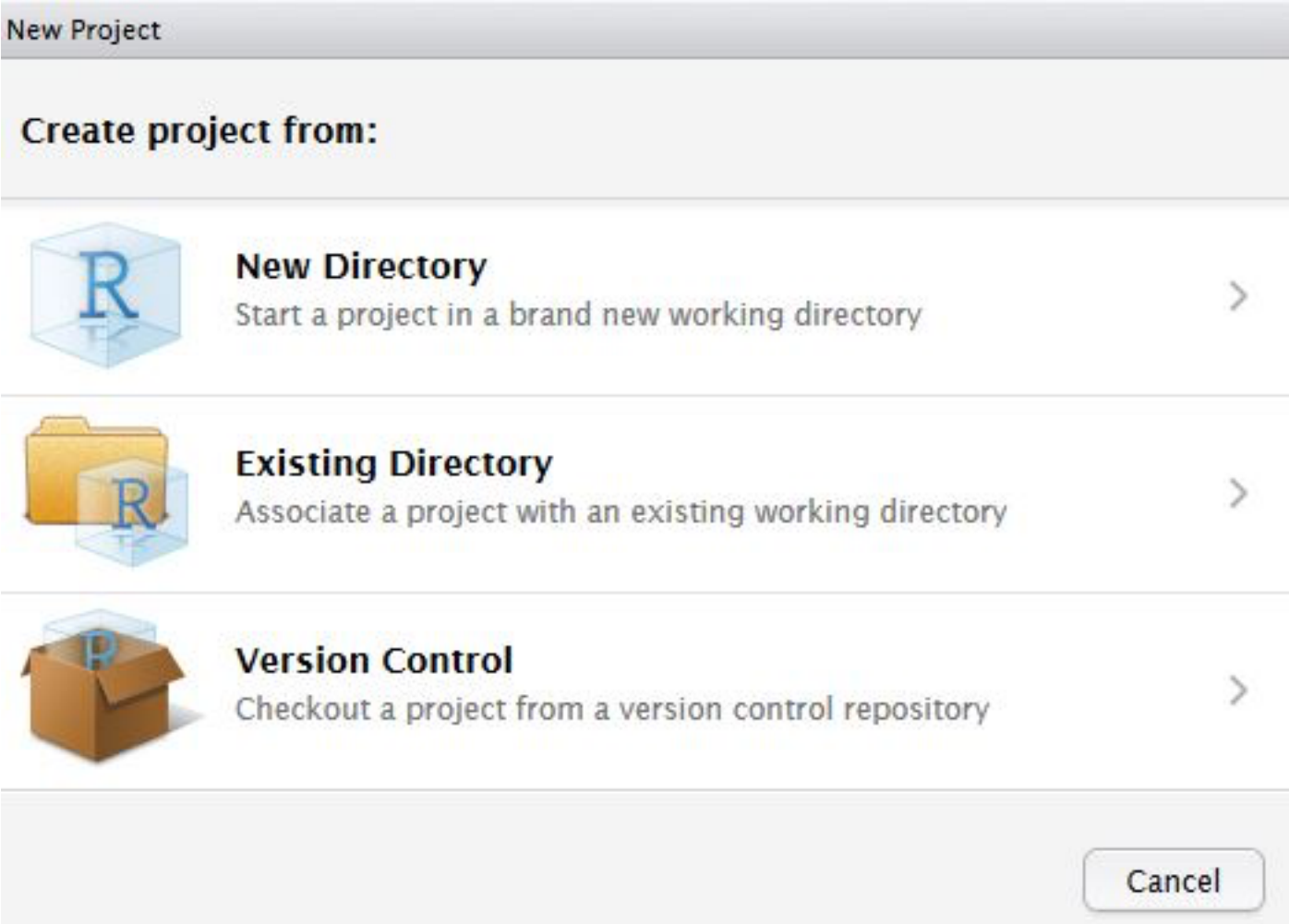
Tell Git who you are. Remember Git is a piece of software running on your own computer. This is distinct to GitHub (that we will talk about later), which is the repository website. In RStudio, click **Tools > Shell ... > Enter**. Then paste in the following with substituting your name and email in place of mine below:

```
git config --global user.email "bjgrant@umich.edu"
git config --global user.name "Barry Grant"
```

### 3. Using Git in an RStudio project

Lets begin creating content by opening a new **RStudio Project**. Projects are a good way to organize your work. It is similar to creating a new folder/directory and using it to store all the files related to a particular project.

To do this click **File > New Project > New Directory > Empty Project**



Lets give your new project the name **Test** and set the location to your **Desktop** folder. You can either type this in or **Browse** to this location.

Also please be sure to click the optional **Create a git repository** option!!!

This will start a new clean RStudio session. Note if you have done this properly then two files will be placed in your new directory/folder named Test (you can see these in your Files pane of RStudio interface). We will talk more about these later. If you don't see these files then please let me know.

## 4. Create some new content that we will track with Git and add to GitHub.

---

In RStudio, click **File > New File > R Markdown document > From Template > GitHub Document > OK**.

This will create a new document in your project folder/directory and display it in RStudio. Save this with the name **rstudio\_report**.

Note don't use spaces in your files names please - unix systems don't like this.

After saving your new script (i.e. the file named **rstudio\_report.Rnw**), it should appear in the Git tab on the Environment / history panel.

Also in RStudio a black-red-green GIT icon on the top toolbar should appear. Click this and then **Git > Commit...**

This should bring up a new window where we can both **Stage** our files for version control tracking and provide a descriptive **commit message**.

To do this first click on the **Staged tick boxes** next to each filename then type the **commit message** "First commit" into the text box and click **Commit**.

Congratulations - you have just made your first Git commit!

Stop here for a demo from Barry along with further discussion. If you finish early feel free to explore the result of clicking the **Knit** button on your script. Make changes to your script (i.e. your 'rstudio\_report.Rnw' file) and staging/committing with Git.

## 5. Create a GitHub account


---

If you haven't already, create a GitHub account at <https://github.com>. Make sure you pick the free plan.



Once you have done this we can begin creating and tracking your RStudio projects on GitHub.

In GitHub, create a New repository by clicking the plus + icon in the top navigation bar here and then **New repository**. I named mine "test".




 Search GitHub

[Pull requests](#) [Issues](#) [Gist](#)

 [+](#) 

Owner

 ewenharrison ▾


Repository name


test ✓

Great repository names are short and memorable. Need inspiration? How about **flaming-nemesis**.

Description (optional)


Test

☒  **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

## 6. Add your GitHub repository to your RStudio project

On the resulting GitHub web page copy the text under “...or push an existing repository from the command line”. We will need to add this to our RStudio project.

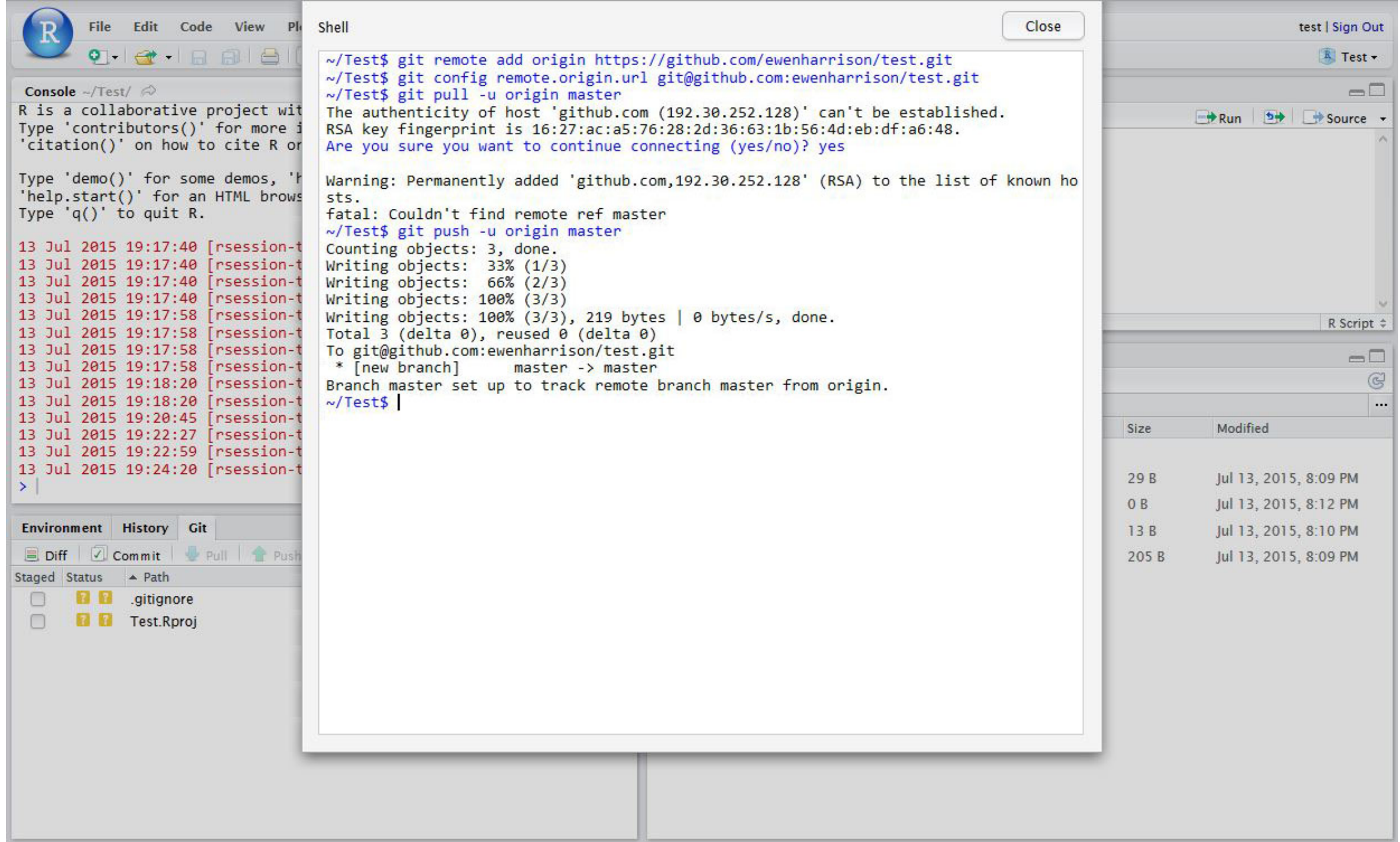
Back in RStudio, again click **Tools > Shell ...** . **Enter**:

And then paste your copied code in to the shell. Mine looked something like this:

```
git remote add origin https://github.com/bioboot/test.git
git push -u origin master
```

You can close the shell after this and return to RStudio.

After saving your new script, it should appear in the Git tab on the Environment / history panel.



You have now pushed your commit to GitHub, and should be able to see your files in your GitHub account by refreshing the web-page (i.e. reloading). The **Pull Push** buttons in RStudio under the **Git** icon will now also work. Remember, after each Commit, you have to Push to GitHub, this doesn't happen automatically.

Congratulations - you have just posted your first commit to GitHub!

If you still have time for more hands-on work consider adding a README file and editing this on-line via GitHub. You can then use the **Git > Pull** option in RStudio to have this synced to your local computer.