Github for fun and profit

Nicholas Horton (nhorton@amherst.edu)

February 1, 2018

Github

Figure 1:

The American Statistical Association's 2014 Guidelines for Undergraduate Programs in Statistics reinforced the need for students to be able to undertake increasingly complex data analysis using sophisticated approaches. According to the guidelines, students should undertake these analyses in a "well-documented and reproducible way". In addition, statistics is increasingly a team sport, and students need to "demonstrate ability to collaborate in teams and to organize and manage projects".

The recent Park City group paper on data science programs (http://www.annualreviews.org/doi/abs/10.1146/annurev-statistics-060116-053930) reinforced the importance of the capacity to undertake reproducible workflows.

Version control systems serve a critical role in helping individual analysts track their code and data in repositories and facilitate sharing of files with research teams. These systems allow users to maintain multiple versions of files with multiple users. Github is a wildly popular web-based interface to the flexible and powerful distributed Git revision control and file management system, with more than 24 million users and more than 67 million repositories.

Today's activity will allow us to move forward in terms of the use of Github in the class.

Step 1: Configure github on the r.amherst.edu server (This is a one-time only task)

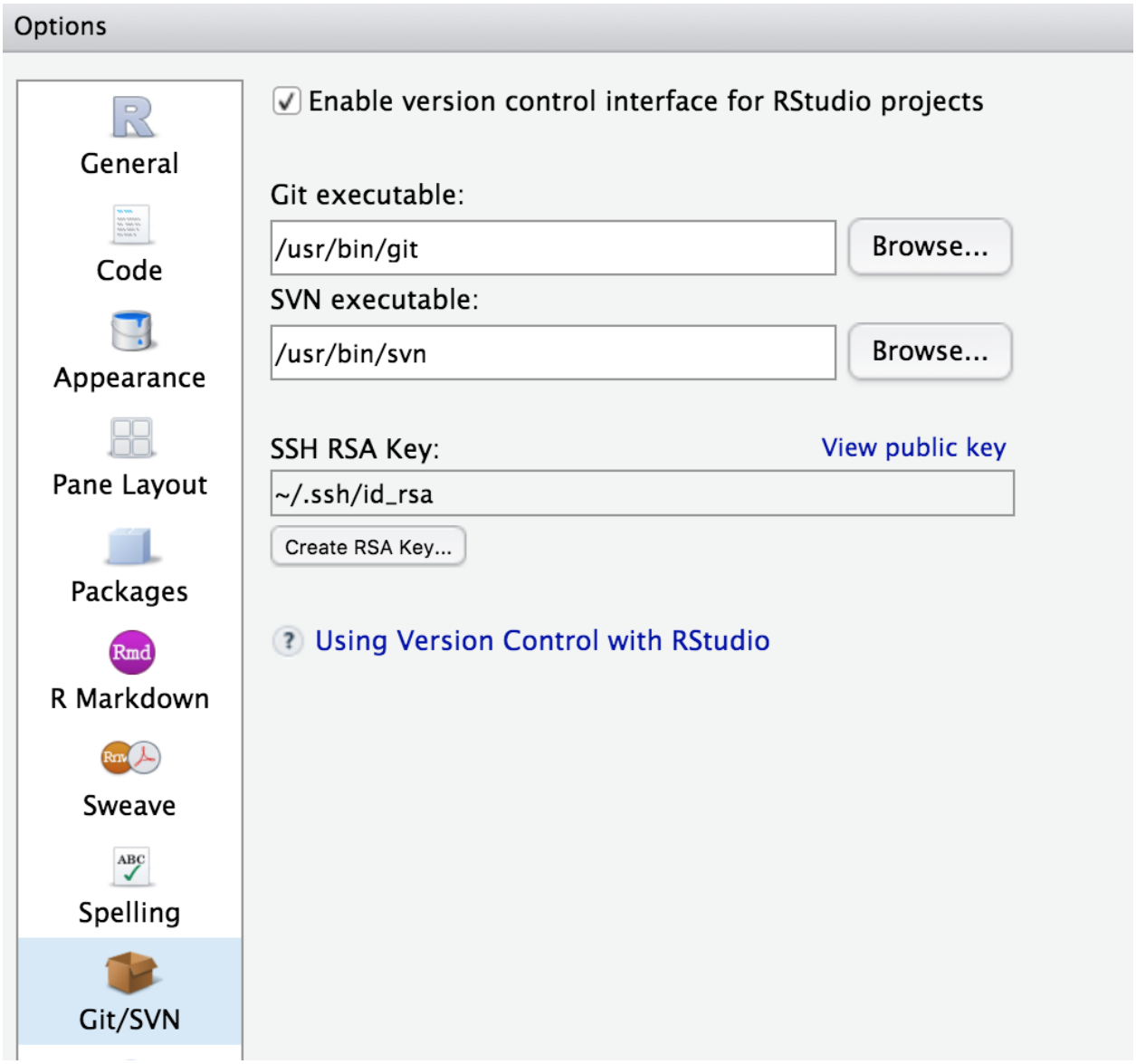
1) Select "Shell" from the "Tools" menu
2) Run the commands (replacing XX with your username and email address)

```
git config --global user.name 'XX'
git config --global user.email 'XX'
git config --global --list
```

3) Close the shell

Step 2: Generate ssh (secure shell) keys (This is a one-time only task)

1) Generate ssh keys on the RStudio server, select `Tools`; `Global Options`; `Git/SVN`; `SSH`



2) Click on the “Create RSA key” link"

3) I would recommend *not* using a passphrase (otherwise you will be regularly prompted for your password)

4) Close the “Create RSA key” box.

Step 3: Add public key to github (One time only)

1) In RStudio, click on the “View public key” link (under “Tools”, “Global Options”, “Git/SVN”)

2) Copy the public key.

3) Open up github.com (with you logged in)

4) click on the icon on the top right and select “Settings”

5) Click on “SSH and GPG keys”

6) Click on “New SSH key”

7) Title this ‘Amherst R server’

8) Paste in the public key

9) Github will prompt you for your password (you don’t want people adding in keys without your knowledge!

10) Close the window.

Step 4: Clone the class repository in the RStudio (r.amherst.edu) server (one time per repository)

1) Open the class repo (https://github.com/Amherst-Statistics/STAT231-S18)

2) Click on the Clone or download (green) button

3) Select “Use SSH” (rather than https)

4) Click on the copy to clipboard button to the right of the URL

5) Back in RStudio, click on `File`, `New Project`; `Version Control`; `Git`; then paste the repository URL

6) Press the Tab key to fill in the project directory name

7) Change the “Create project as subdirectory of” value from “~” (home folder) to “~/git” by clicking “Browse” then creating a “git” folder. (I recommend this as a place to put all of your github repos)

8) Click on `Create Project`

9) The first time you connect to github from the RStudio server, it will prompt you with a forbidding “authenticity can’t be established” message. You should type “yes” into the box to trust this host and submit.

Step 5: Add a file to the class github repo

1) Create a new file called LASTNAME.txt (where LASTNAME is your last name) that includes the following text:

This is FIRSTNAME (replace with your name). I'm adding this file to github

2) save that file
3) click on the "git" tab on the top right
4) check the box to the left of the file "LASTNAME.txt"
5) Click on "Pull" (down arrow) to ensure that you have the most recent version of the repo.
6) click "Commit"
7) Add a commit message in the box on the right that opens up: "Adding LASTNAME's files"
8) Click the "Commit" button. (You may need to do a "pull" again if other changes have taken place.
9) Click the "Push" button.
10) Close the Commit window.

Congrats! You've added a file.

CLOSING ADVICE:

1) pull often!
2) don't add pdfs (or other files that are generated by your Rmd) to github until you are ready to submit an assignment. And consider giving it a unique name when you do. For example, if your file is `scorecard1.Rmd` which generates `scorecard1.pdf`, instead of adding `scorecard1.pdf` to the repo, rename the file as `scorecard1-2018-02-03.pdf` by renaming `scorecard1.pdf`.
3) get stuck or run into a conflict? Try running `git status` at the shell or (if you are willing to forget about your changes) `git stash`.
4) Really stuck? Please let me know and I'd be glad to assist.