

Working Solo with GitHub

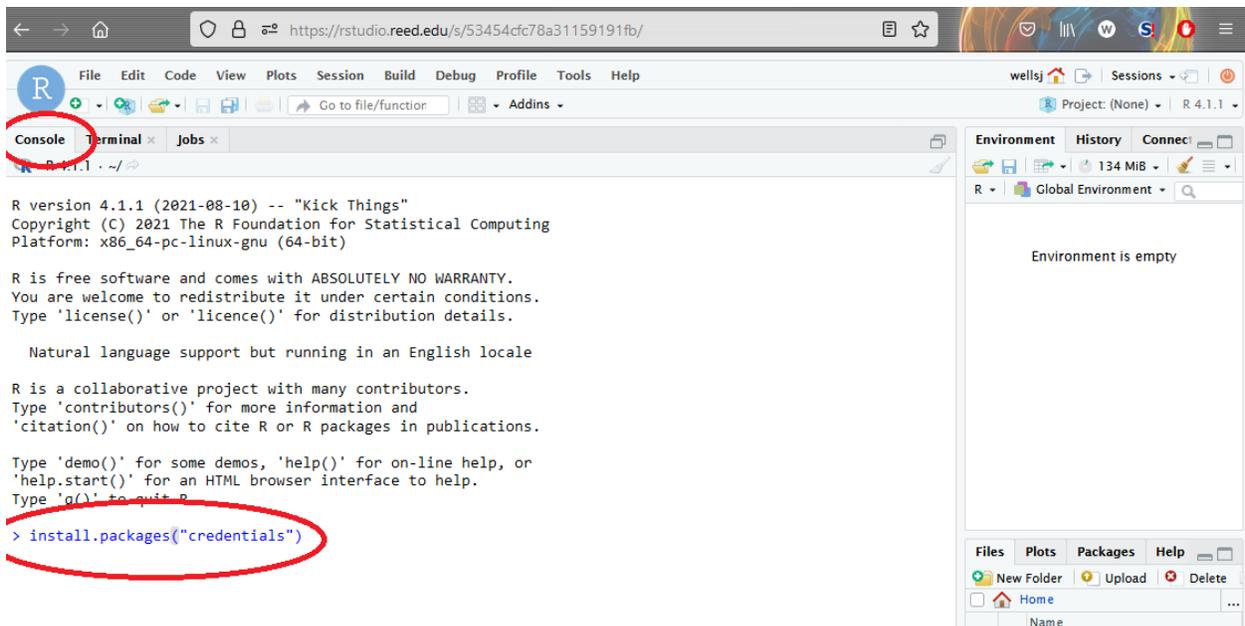
1. Introduce yourself to your group members and exchange GitHub user names. Then, work through the following steps (feel talk with your group members as you go through this).

These instructions assume you will access GitHub primarily via the RStudio Server. If you wish to work locally instead, talk to me after class.

Part 1: Cache Credentials

NOTE: Password authentication is deprecated as of August 13th, 2021. So even if you've used Git, GitHub and RStudio before, you will need to follow these instructions to push commits to GitHub. However, once you've completed these steps once, you shouldn't need to do them again for at least several months.

2. Navigate to your personal GitHub page, and in the upper-right corner, click your photo and click Settings.
3. In the left sidebar, find **Developer Settings**, then **Personal access tokens**, and click **Generate new token**.
4. Give your token a name like my token 243 (or whatever else you want), select an expiration time (recommended 6 months), then select all permissions, and the click **Generate token**.
5. Copy the token to a new text document and save somewhere on your computer. *Once you leave the current page, you will not be able to see your token again. (If you lose the token, you can delete it and create a new one, but it is inconvenient)*
6. Navigate to rstudio.reed.edu and login.
7. In the console pane, install the `credentials` package by typing `install.packages("credentials")`.

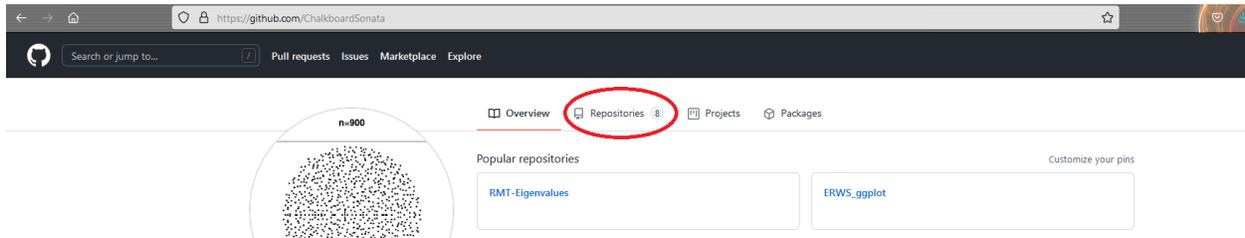


8. In the same pane, load the package and call the `set_github_pat()` function, and then enter your personal access token:

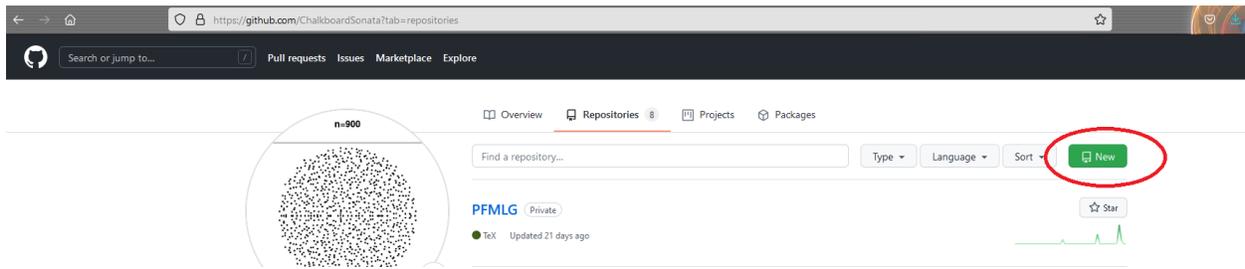
```
library(credentials)
set_github_pat()
```

Part 2: Create a Repo

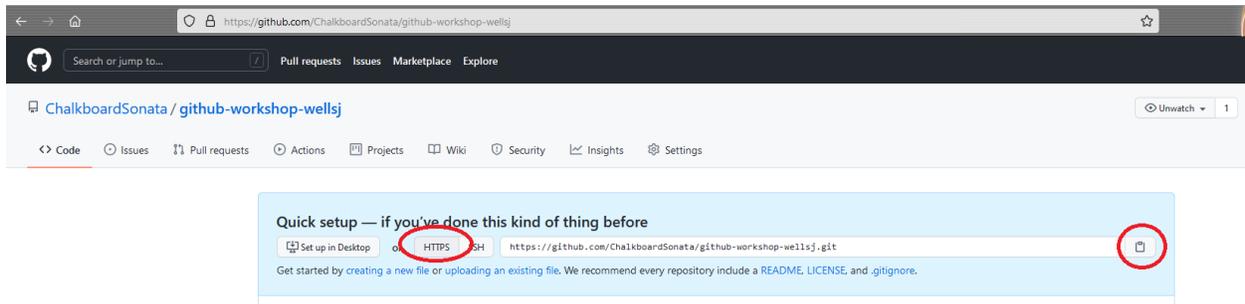
9. Return to your personal github page and click the Repository tab along the top menu.



10. Click the green New Repository button along the right side.

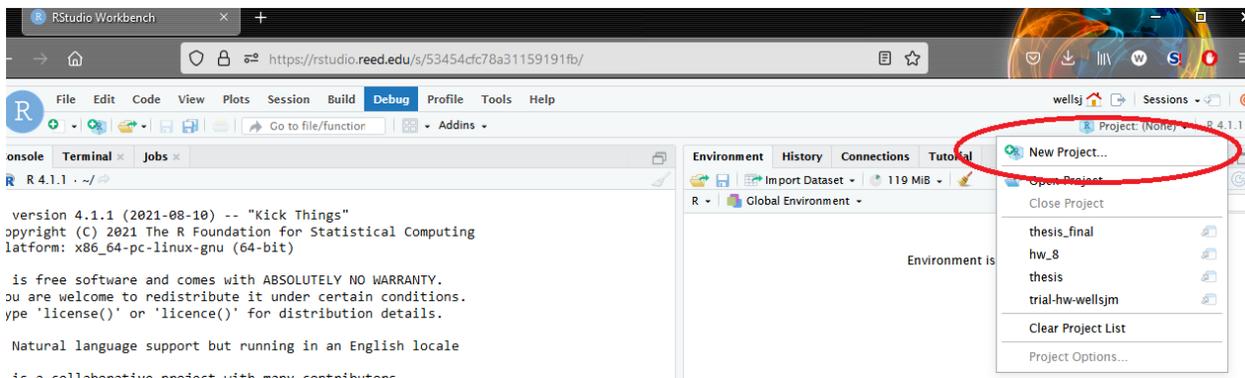


11. Fill in the name of your repository using `github-workshop-wellsj` (but with your own name), leave the other fields blank, then click the green Create Repository button.
12. Be sure the grey HTTP button is selected, then click the clipboard at the right to copy this URL to your clipboard.

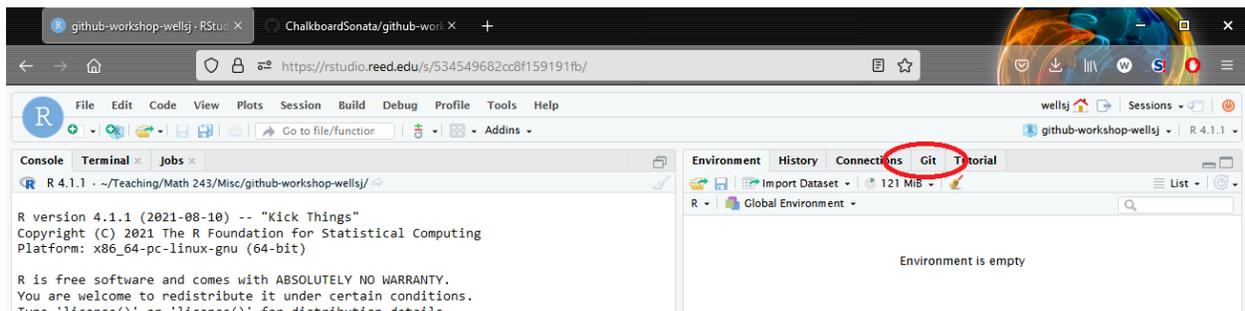


Part 3: Cloning Repos

13. Navigate to `rstudio.reed.edu` and login.
14. At the upper-right corner, click the Project tab and select New Project.



15. Click Version Control > Git > then paste your repository URL into the first field and click Create Project.
16. In the far upper right corner, click File > New File > Markdown File and immediately click the Save icon at the top of the file and name it README.md.
17. Click over to the Git tab in the upper-right quadrant of the screen and check that your new file appears there.



Part 4: Connect RStudio, Git and GitHub

18. Click the check box next to README.md and click the Commit button. On the next screen, enter **initial commit** into the text field, click Commit, then click the Push button. When prompted, provide your username and password for your GitHub account (not your Reed account).
19. In your browser, click over to your GitHub repo and check that it now contains your blank README.md.
20. Back in RStudio, add the following lines to the top of your README.md (except plug in your name) and click the Save button.

```
# Welcome to my repository
```

```
Authors: Nate Wells
```

21. Over in the git pane, click the check box next to README.md and click the commit button. On the next screen, enter **added title and name** into the text field, click Commit, then click the Push button. When prompted, provide your username and personal access token (this will be the last time you need to provide the token).
22. Return to your README.md file, add the line **DONE** at the end, save and commit the change, then push to GitHub. Verify that you do not need to enter your username and password.

Part 5: Typical Workflow (for now)

- Save and knit your document frequently (i.e. any time you complete a chunk of code).

- Get in the habit of making frequent commits on your project (at least once per work session, or even more often!). The more often you make commits, the more easily you can track changes (and revert to an earlier version if necessary).
- All commits are stored locally until you **push** them to github. Once you are done with a work session, always **push** all commits.