Instructions for configuring RStudio and working with repos

Instructor: Soumya Mukherjee
Content credit: Dr. Matthew Beckman and Olivia Beck

July 10, 2023

How to configure GitHub and RStudio for the very first time

You only need to do these steps once.

1. Make an account on github (You all have already done this!)
2. **ONLY IF YOU ARE NOT SURE THAT Git IS INSTALLED ON YOUR LAPTOP**: On you personal computer/laptop, open RStudio, and follow the instructions in Chapter 6 of Happy Git and GitHub for the useR: https://happygitwithr.com/install-git.html
3. Set up your global git configurations: **In the console** replace with your information (Your First and Last name, and psu email ID):

```
install.packages("usethis")

library(usethis)
use_git_config(user.name = "Jane Doe", user.email = "jane@example.org")
```

4. Generate a personal access token:

Think of a personal access token as a super secret password that you use to push and pull with. Let it be known that the password that you use to login to GitHub's website is NOT an acceptable credential when talking to GitHub as a Git server.

4A. Make sure you are logged in to GitHub.

4B. Copy and paste the following R command into the R console, and press enter.

```
usethis::create_github_token()
```

4C. A new browser window will open on GitHub. Select "No expiration date" and make sure all the boxes are checked. At a minimum, select repo, workflow, and user.

4D. Click Generate Token. COPY DOWN THIS TOKEN SOMEWHERE SAFE, LIKE NOTEPAD. GITHUB WILL NEVER GIVE IT TO YOU AGAIN.

5. Copy and paste the following R command into the R console, and press enter.

```
gitcreds::gitcreds_set()
```

6. Paste the token you generated and press enter. You are done!

How to configure GitHub and RStudio when you want to start a new Repo

You do these steps every time you want to create a local version of repo on your computer.

1. Find the repo on GitHub by either copying an existing repo (1A) or creating a new repo (1B).

1A. If you want to copy and existing repo, go to that repo's github page, click on the green code drop down button and copy that link. It will be in the form of https://github.com/user-name/repo-name.git

1B. If you want to create a repo, go to this list of your repo's on github, click the green New button a the top right, name your repo and give it a brief description, decide if you want it public or private, check the box "Add ReadMe file", add a .gitignore for R (open the drop down box and select R), choose a licence (usually the GNU General Public Licence is a good one), and click Create Repository. You will then need to open that repo on github, and copy the .git link the same way you would in step A.

2. Create an Rproject linked to this repo on your computer

2A. Open Rstudio. Click "Project (None)" on the top right-hand corner. Click New Project.

2B. Click Version Control. Click Git.

2C. In the repo URL paste the repo link that you got in step 1. The project directory name will automatically generate. Select the location on your computer where you want to save it.

2D. Click Create Project.

2E. Voila! You've downloaded/cloned a repo on your personal computer!

How to work with GitHub and RStudio

You do these steps every time.

1. Open the RProject. Either…

Option 1A: Open RStudio. Click on the the project button at the very upper left-hand corner of RStudio, click open project, find the location of the project on your computer, click the .Rproj file and open.

Option 1B: Find the location of the project in your computer directory, click the .Rproj file and open. This should open your project in a new session in RStudio.

If you did step 1 correctly there will be a tab named "Git" in the upper right panel (next to where the Environment tab is located) AND the project button (at the top left of RStudio) will have the project name (i.e. will not say "Project (none)"). If you did this step incorrectly, this will not be there and you will not be able to commit, push, or pull.

2. Pull from GitHub. This will ensure your local version (on your computer) is up to date with the master version (on GitHub).

2A. Open the project on RStudio using the Project > Open on the top left hand side.

2B. Click Pull. If your version is already up to date, it will tell you so. If your version is behind what is on GitHub, then Git will try update your version (merge conflicts might happen)

2C. Note, all changes must be committed before you can pull.

3. Edit/Add/Delete documents as you want. Save them to your computer. (Doing this does not save them to GitHub, just your personal computer).
4. When your are ready to send your changes to GitHub

4A. Click the "Git" tab on the upper right panel (this will not show up if you did step #1 incorrectly). Click Commit. Select the files you wish to commit. Type out a brief message of what the changes are that you made (1-2 sentences). NEVER LEAVE A COMMITT MESSAGE BLANK. Click Commit. You can (and should) have different commit messages for different (groups of) files.

4B. When you have committed all of your changes, click Push. Most times this should run without reqiring anything more from you.

If prompted, Enter your GitHub username. When it prompts you to enter a password, enter the personal access token you generated (not your github password). If you forgot where you have saved the token, genrate a new one again.

4C. If it was successful it will say something along the lines of "Merge to master complete". If it was not successful, it will tell you why (usually it's either merge conflicts, or you are trying to push a file that is too large).

5. (Optional Step I recommend) When you are done working on your current code project,

5A. close the project by clicking on the the project button at the very upper left-hand corner of RStudio, click close project. (This will ensure any code you do from here on out does not affect the .Rproj). The projects button should now say "Project (none)".

5B. When you want to open this project back up, start back at step 1!