Getting started with Github

Kath Sherratt

2024-02-20

Installing Git

"Git" is a piece of underlying software installed on your computer that can recognise and track changes to files (when you ask it to). This needs to be installed. The installation should be smooth but can look quite technical because it needs access to your entire operating system. Bear with it!

A comprehensive and friendly installation guide for Git is available at: https://happygitwithr.com/install-intro. While this tutorial is a summary of the necessary steps, I strongly recommend keeping this guide handy while setting up and getting started with Git.

The first thing you'll need is a Github account. Set one up for free at https://github.com/. Choose a username you're happy working with!

Install Git

Some systems will already have Git installed. To check, open a terminal in RStudio (next to the Console, or at RStudio » Tools » Terminal » New Terminal) and then enter the following commands, one at a time, after the dollar sign prompt ($) as shown below.

```
which git
git --version
```

- If git is installed, the result will indicate where the software is located and the version you have available
- If Git has not yet been installed, you might see something like `git: command not found`. In this case:
  - If your computer prompts you to install Git, follow the prompts to do so.
  - If nothing happens, go to the Git website and select the version which is compatible with the OS of your computer (e.g. Windows/Mac/Linux/Solaris). Follow the instructions to download and install

Login to Git

You now need to login to Git. You can do this within RStudio using the handy `usethis` package. Change the `use_git_config()` settings to your Github username and email:

```
## install if needed (do this exactly once):
## install.packages("usethis")

library(usethis)
use_git_config(user.name = "Your Username", user.email = "your@email.com")
```

Install a friendly interface

Like R, it's possible to use Git only from the command line. However this can be hard to read and frustrating to use. I suggest using an extra piece of software that provides a user interface to Git - just like RStudio does for R. A simple option is Github Desktop:

- Go to Github Desktop and follow the download instructions

Communicate with Git online

There's one last step before you can get up and running with Github. You'll need to authenticate yourself in order to connect between the Git you now have on your computer, and the online storage of your files on Github.

This is done using a Personal Access Token (similar to a password). Follow these steps, starting within RStudio:

1. Create a PAT. This will take you to the Github webpage to create a token. Give your token a simple name, accept the default settings, and continue. Copy the PAT (the long string) - you won't be able to see it again.

```
create_github_token()
```

2. Execute this code, and when prompted, input the token you have just copied to authenticate yourself with Git.

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

You are now all set up to start working with Github.

Integrating RStudio Projects with Github

Github works really well with RStudio Projects. You can pretty much continue operating exactly as you already do - but now with quick access to back ups and a new suite of tools for tracking and collaborating on your files. The best way to get comfortable with using Git is to just start using it in your everyday work. Here is a quick look at the basic Github workflow:

- Github works by storing your files in a "repository" (or "repo") - essentially a directory that holds all files (data, code, output) associated with one Project
- You take a snapshot of this directory to record the state of all your files - this is called a "commit"
- You then upload the snapshot to Github - this is called a "push". The files as they are in the snapshot are stored remotely (and can be browsed online)
- Repeat the commit-push process as often as you want to store changes in your files
- Collaborate with others by sending them a link to the repository
  - They can download your codebase ("clone")
  - See your ongoing changes ("pull")
  - Modify and suggest changes to files ("pull request") - all without the your files being affected

Get going by converting an existing RProject into a Github repository.

Create a repository from an RProject

Open an existing RProject and load the `usethis` package. Then try the following commands to turn your RProject into a repository hosted on Github:

```
library(usethis)
use_git()
use_github()
```

You should be taken to a Github page showing your files - you've created your first Github repo!

- Browse your files online and explore how Github stores and displays your files in the web browser.
- Note: Github creates all new projects as public by default. If you want to keep your project files private: browse to "Settings" at the top of the Github webpage > scroll to the end, until you see "Change repository visibility" > follow the instructions to make the repository private.

Modify and track files

To start using Github, a snapshot of your files and upload them to Github

For practice, try creating something new in the RProject directory that you've just converted to Github repo. Then open up Github Desktop and follow these steps to

Continue learning

- R for the rest of us: How to Use Git/GitHub with R
- Happy Git with R