Using GitHub with RStudio

Becca Selden

2025-09-10

Introduction

The SeldenLab uses Github with RStudio to help us share code and collaborate on analysis. RStudio makes it pretty easy to work with Git as long as always remember this order of operations:

1. Always “pull” before you start working–This brings the most recent code from the cloud onto your personal machine and makes sure that you are modifying the most up to date version. Git can work with new changes to the same file but a “conflict” could arise if you are working on a section of code that was already changed and there isn’t a way to merge the two.
2. You save your code regularly, “commit” fairly regularly when you make a change, and “push” to the remote repository on the cloud immediately after you commit so you are pushing just a single “commit”.

“HappyGit with R”

Folks have developed a fairly user friendly introduction to using R with git: https://happygitwithr.com/ (https://happygitwithr.com/)

Installation

Work through the Installation section of Happy Git with R to:

1. Register a GitHub account (https://happygitwithr.com/github-acct).
2. Install or upgrade R and RStudio (https://happygitwithr.com/install-r-rstudio).
3. Install Git (https://happygitwithr.com/install-git).
4. Introduce yourself to Git (https://happygitwithr.com/hello-git). Set yourself up with git using the `usethis` package directly in RStudio

You don’t need to setup a Git client (which is the last step in the Installation section)

Connect Git, GitHub, and RStudio

Work through all sections in this part of the tutorial (https://happygitwithr.com/connect-intro).

For the first section on a Personal Access Token (PAT) I would recommend the method with the `usethis` package:

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

```
## • Call `gitcreds::gitcreds_set()` to register this token in the local Git credential
store
##   It is also a great idea to store this token in any password-management software tha
t you use
## • Open URL 'https://github.com/settings/tokens/new?scopes=repo,user,gist,workflow&des
cription=DESCRIBE THE TOKEN\'S USE CASE'
```

Follow the instructions (https://happygitwithr.com/https-pat) on copying, pasting this PAT

Early GitHub Wins

1. Create a new repository, get the repository on your computer and then push some changes to GitHub in your personal account (not the SeldenLab). New Project, GitHub first (https://happygitwithr.com/new-github-first).
    - Work through 15.1 to make a repo on GitHub (15.1 has lots of options but I find the first one using the GitHub website the most straight forward).
    - Work through 15.2.2 to connect to the repository on GitHub using RStudio.
        - File –> New Project –> Version Control –> Git and then use the repository url
    - Work through 15.3 to make local changes, save commit
2. Connect to an existing repository on Github Existing project, GiHub first (https://happygitwithr.com/existing-github-first)
    - Follow 16.2.2 to connect to the existing repo using the same process as above where you make a new project in RStudio from version control using the url for the repository
    - Use the url address for one of the SeldenLab repos "oyster_shell" to download it to your computer "https://github.com/selden-lab/oyster_shell (https://github.com/selden-lab/oyster_shell)".
3. Do Section 18 on Test drive R Markdown (https://happygitwithr.com/rmd-test-drive) by creating a new RMarkdown file in the repository you created in your personal acccount (not the one in the SeldenLab). Commit, and push this file to GitHub