

Introduction to R - Young Researchers Fellowship Program

Lecture 3 - More on data manipulation/cleaning

Daniel Sánchez Pazmiño

Laboratorio de Investigación para el Desarrollo del Ecuador

October 2024

Command line approach

Git Bash/Command Line

- In order to maintain a version control of your code, you can use Git.
- You need not use RStudio to use Git, you can use the Git Bash or the Command Line.
- This would mean having created an R Project, and using Git to initialize a repository in the project folder.
- `git remote add origin` is the command to connect the local repository with the remote repository.
 - You can find the URL of the remote repository in GitHub, after you have created it.

Bash commands to initialize a repository and connect it to GitHub

```
cd path/to/your/project  
  
git init  
  
git add .  
  
git commit -m "Initial commit"  
  
git remote add origin  
  
git push -u origin master
```

Sending commits

- Every time you make a change in your code, you can send a commit to the remote repository.
- You can check the status of the repository with `git status`.
- You can add the changes to the staging area with `git add ..`
- You can commit the changes with `git commit -m "Message"`.
- You can send the commit to the remote repository with `git push`.

Caveats

- In order to do this, you must have successfully installed Git in your computer and connected it to your GitHub account.
- This can be done following our previous instructions in the Git and GitHub module.
- Otherwise, you may follow an R-specific resource: Happy Git and GitHub for the useR.
 - This resource is a book that explains how to use Git and GitHub with R.
 - It is written by Jenny Bryan, a professor at the University of British Columbia.

RStudio approach

RStudio Git pane

- RStudio has a Git pane that allows you to interact with Git.
- This means you need not use the command line to interact with Git, only a visual interface.
- You can initialize a repository, add files to the staging area, commit changes, and push commits to the remote repository.
- You can also pull changes from the remote repository.

RStudio Git pane

- You can find the Git pane in the top right corner of RStudio, once you've initialized a Git repository in your project.

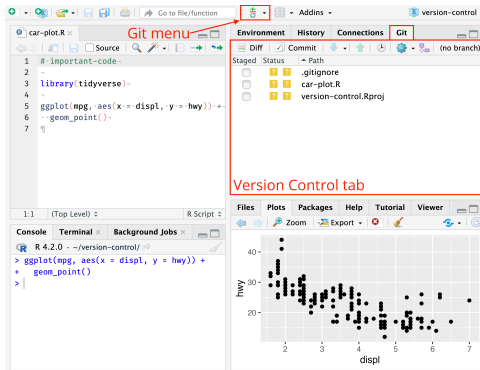


Figure 1: Git pane

Initializing repositories from RStudio

- Four major ways to initialize a repository in RStudio:
 - 1 `git init` in the command line (this you already know)
 - 2 Create a new project in RStudio and check the “Create a Git repository” box.
 - 3 Get the link of a repository in GitHub and open a new project in RStudio from that repository (File -> New Project -> Version Control -> Git) (not technically initializing a repo)
 - 4 With the `usethis` package: `usethis::use_git()` and `usethis::use_github()`.

Creating a new project in RStudio with a Git repository

- You can create a new project in RStudio and check the “Create a Git repository” box.

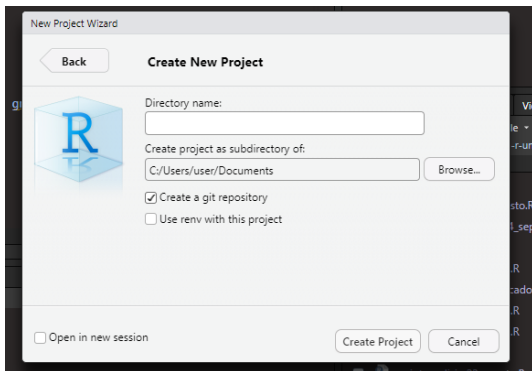


Figure 2: Git from scratch in RStudio

Connecting to GitHub

- You can connect your local repository to a remote repository in GitHub.
- You will need to have done at least one commit.
- If you haven't made a GitHub repository yet, you can do so from RStudio.

Connecting to GitHub

```
usethis::use_github()
```

- Might prompt you to feed your PAT (Personal Access Token) to RStudio.
 - If you use another method to connect, such as SSH, you may need to do additional steps. See Happy Git.
- Otherwise, you may decide to create a private or public repository.
 - If you have several accounts (e.g. an organization account), you can choose the account to create the repository in in the argumnets

usethis::use_git()

- If you already have stuff in your project or simply don't want to use the menus, you can use `usethis::use_git()`.
- This will initialize a Git repository in your project.
- You can also use `usethis::use_github()` to connect to GitHub after you've initialized the repository.
 - Additionally, use Git Bash to connect to an existing new repository in GitHub.

Using `usethis`

- `usethis` is a package that helps you automate tasks in R.
- All calls that you do with `usethis` are interactive, meaning that you will be prompted to do something.
 - You should be careful with this, as you may not want to do something you didn't intend to.
- Never use `usethis` calls on the script, call them on the R Console.
 - We don't want to create a new repo every time we run the code, correct?

Version Control R Project

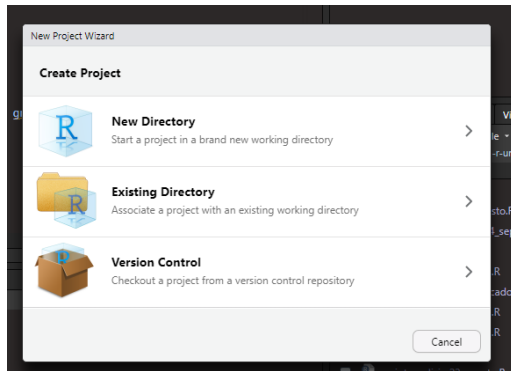


Figure 3: Version Control R Project

Version Control R Project

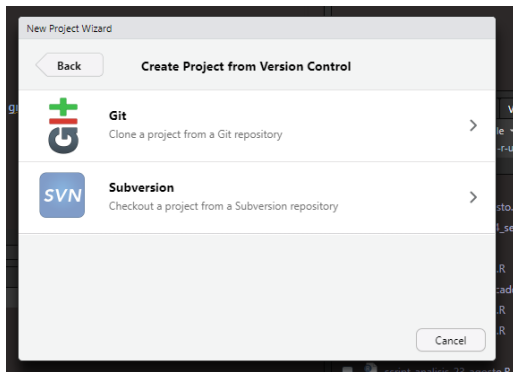


Figure 4: Version Control R Project

Cloning a repository from GitHub into the Version Control R Project

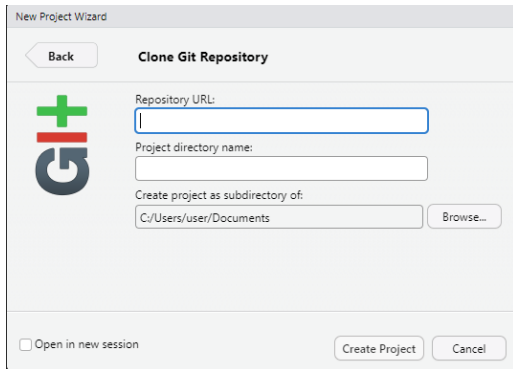


Figure 5: Cloning a repository from GitHub

GUI approach

GUIs for Git

- To avoid all the *command line bullshittery (sic)*¹, you can use a GUI for Git.
- There are several GUIs for Git (many of them are free), such as:
 - GitHub Desktop (free)
 - Sourcetree (free)
 - GitKraken (free with limitations)
- These GUIs allow you to interact with Git without using the command line.
 - You can initialize repositories, add files to the staging area, commit changes, push commits, and pull changes
 - Also allows a nice way to diff files and resolve conflicts.
- RStudio would technically be a GUI for Git, but it is more of an IDE that has Git capabilities.

¹Guo (n.d.) in Bryan (2024)

GitHub Desktop

- GitHub Desktop is a free GUI for Git, made by GitHub.
- Download it from desktop.github.com.
 - Will easily open repositories in GitHub Desktop and clone them to your computer.
- You can initialize repositories, add files to the staging area, commit changes, push commits, and pull changes.
- Diffing files (seeing the differences between two versions of a file) and resolving conflicts is also possible.

GitKraken

- A very functional GUI for Git, with a free version that has some limitations.
- GitHub Student Pack users can get a free Pro version.
- Download it from gitkraken.com.

Command line vs. GUI

- Ultimately, most of the things you can do with Git can be done with the command line.
- There is no prize for using the command line, so use whatever you feel more comfortable with.
- HOWEVER, when things get ugly... the command line will be your only friend.
- The GUIs are more user-friendly, but they may not be as powerful as the command line.

Command line stuff which is important

- Solving complex conflicts.
- Solving a Git LFS (Large File Storage) problem.
- Solving a problem with the SSH key.

VS Code

Visual Studio Code

- Visual Studio Code is a free code editor made by Microsoft.
- It has a lot of extensions that can make your life easier.
- One of them is the Git extension, which allows you to interact with Git from the editor.
- You can initialize repositories, add files to the staging area, commit changes, push commits, and pull changes.
- Git Kraken can also be integrated with VS Code, so you can use both at the same time.

Visual Studio Code with R

- You can also use Visual Studio Code with R.
- It is not trivial how to set it up and have code run, but it is possible.
- This is helpful since the Git extension in VS Code is very powerful
 - Better than the RStudio Git pane.
- VS Code will let you to use it with several languages in the same folder.
 - RStudio allows this too, but VS Code has more seamliness with other languages.