Create an RStudio Cloud Demo

Requirements:

- Github Account and some familiarity with GitHub version control terminology.
- RStudio Cloud Account.
- R and RStudio installed on your own computer.

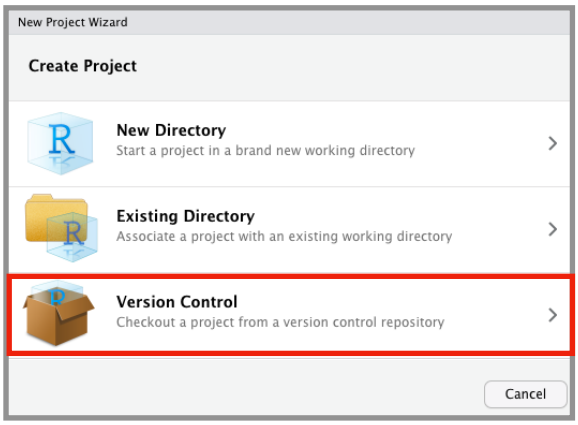
1. **Configure RStudio for use with GitHub**
    1. You only do this once (or as needed).
    2. The directions here are pretty good: https://happygitwithr.com/https-pat.html
    3. Generate a Personal Access Token (PAT) with “repo”, “user”, and “workflow” permissions at https://github.com/settings/token
    4. Use the `gitcreds_set()` function in the `gitcreds` library to store your personal access token with R.

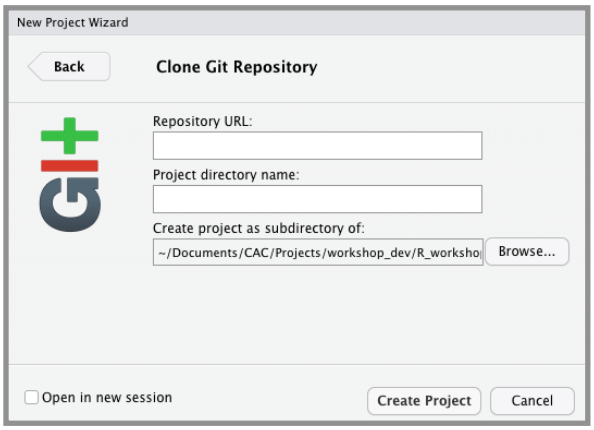
2. Set up a *public* project on GitHub

5. GitHub Docs - https://docs.github.com/en/get-started
6. Quickstart - https://docs.github.com/en/get-started/quickstart/create-a-repo
7. Tips:
    1. Make it a public repository
    2. Choose the R template for the .gitignore file.
    3. You can do this entirely from the GitHub web site.

3. Create an R Studio Project on your computer

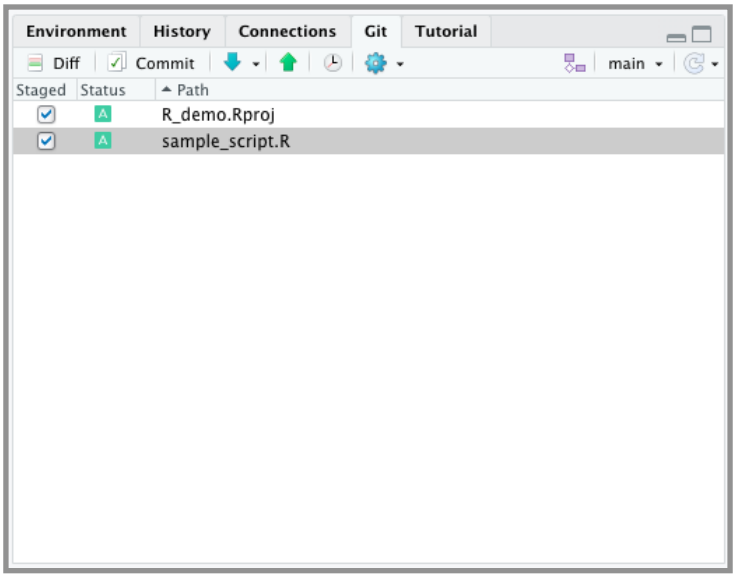
8. Create a new project in your local installation of R Studio (e.g., on your computer)…
    1. In the menu, select “File > New Project…” **OR** click the button that says “Create a project” when you hover the mouse over it. It looks like a blue cube with an R on it.
    2. Choose the Version Control option.
    3. Paste in the name of the repo you just created. It will be in this form: https://github.com/username/repo_name
    4. Choose the local name and location.
    5. Tip: use “open in new session” to run the project in its own copy of R.





4. Make some edits to the project

1. When you install a package, keep track of it in a file with a name like “packages.R”
2. Create a sample script to test the git sync process. Perhaps it will contain `install.packages(“ggExtra”)`
3. In the Git tab, *stage* the new files for upload by clicking the box in the Staged column for each file you want to upload.
4. Click Commit and add a short note about the changes you made in the Commit Notes field. Click commit again.
5. Push (upload) the project to GitHub using the Push button (green up arrow).
6. When the upload is complete, close the extra windows that git leaves behind.
7. Tip: Repeat this stage/commit/push whenever you want to make changes to the file.



5. Create a project on RStudio Cloud from the GitHub repo

1. On the Projects tab of the RStudio web interface, choose “New Project from Git Repository” from the “New Project” menu.
2. Paste in the URL from the repo you created in step 2.
3. Initially, the new project will exactly match the git repo contents.
4. If you make changes to the project on you local computer and commit them to GitHub, you can pull the changes into the RStudio Cloud project using the Git tab.
5. Tip: It is easiest if you refrain from editing files on RStudio Cloud. If files are listed on the Git tab, right-click and choose revert (this discards any changes).
6. Click the green down arrow to download changes from GitHub.

6. Prepare to Share (install packages and test code)

1. GitHub syncs files but does not sync installed packages!
2. As you install packages locally, make sure to keep track by adding packages to the "packages.R" script file.
3. Once you have completed the project locally, sync it with GitHub and pull the changes into RStudio Cloud.
4. Run the package installer script you created on RStudio Cloud. These packages will remain installed and will be available to people you share the project with. If there is any trouble, the packages.R file shows exactly what to install.
5. Test your code on RStudio. Make any changes locally, commit and push them to GitHub, then pull them down to the RStudio Cloud project to test. You might need to revert changed files on RStudio Cloud in order to pull the new changes.

7. Share the Project

1. Access the settings by clicking the gear icon in the upper right of the R Studio workspace.
2. Choose the Access tab
3. Make the project viewable by everyone.
4. Copy the URL from the browser URL bar and share it with others. Each person will view a separate, independent copy.
5. Users may save a copy to their own RStudio account if they wish to save changes.