ETC5513 Assignment 2

AUTHOR
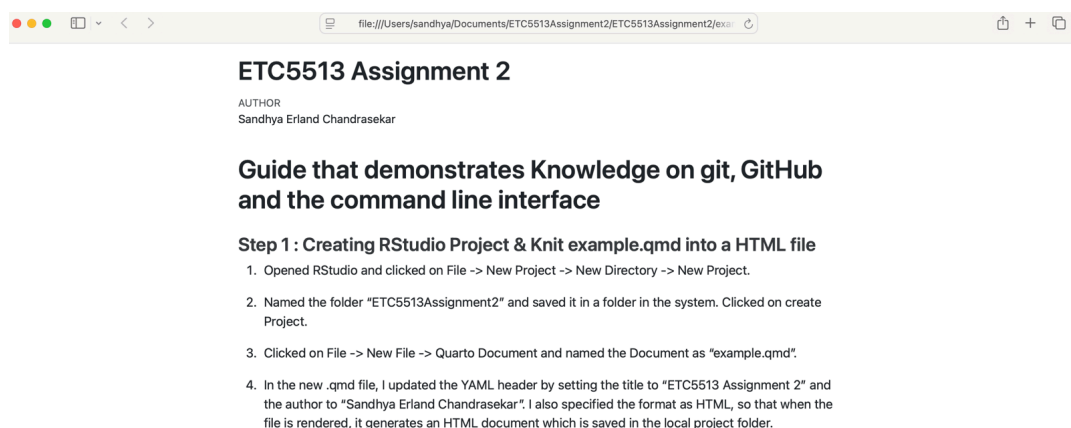
Sandhya Erland Chandrasekar

Guide that demonstrates Knowledge on git, GitHub and the command line interface

Step 1 : Creating RStudio Project & Knit example.qmd into a HTML file

1. Opened RStudio and clicked on File -> New Project -> New Directory -> New Project.

2. Named the folder "ETC5513Assignment2" and saved it in a folder in the system. Clicked on create Project.

3. Clicked on File -> New File -> Quarto Document and named the Document as "example.qmd".

4. In the new .qmd file, I updated the YAML header by setting the title to "ETC5513 Assignment 2" and the author to "Sandhya Erland Chandrasekar". I also specified the format as HTML, so that when the file is rendered, it generates an HTML document which is saved in the local project folder.

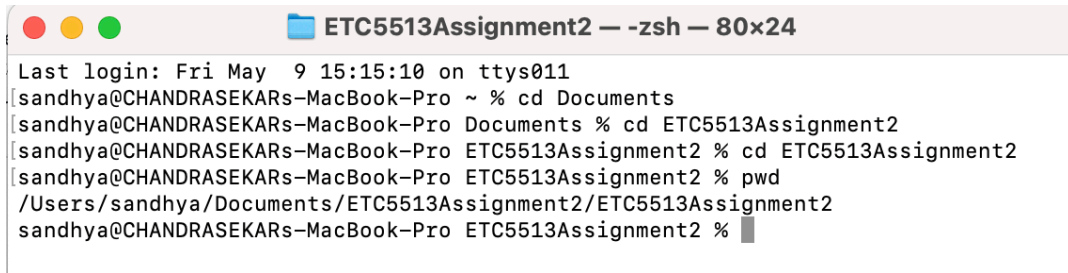5. The result for the knitted file is provided via a screenshot.

```
knitr::include_graphics("Screenshots/Screenshot1.png")
```


Step 2 : Initialising Git and Pushing it to GitHub

1. To create a git repository from existing work first I opened the terminal and navigated it to the root directory of my existing project using the cd command.

```
knitr::include_graphics("Screenshots/Screenshot2.png")
```

```
● ● ●              📁 ETC5513Assignment2 — -zsh — 80×24
Last login: Fri May  9 15:15:10 on ttys011
[sandhya@CHANDRASEKARs-MacBook-Pro ~ % cd Documents
[sandhya@CHANDRASEKARs-MacBook-Pro Documents % cd ETC5513Assignment2
[sandhya@CHANDRASEKARs-MacBook-Pro ETC5513Assignment2 % cd ETC5513Assignment2
[sandhya@CHANDRASEKARs-MacBook-Pro ETC5513Assignment2 % pwd
/Users/sandhya/Documents/ETC5513Assignment2/ETC5513Assignment2
sandhya@CHANDRASEKARs-MacBook-Pro ETC5513Assignment2 % ▌
```

2. I initialized a Git repository in the current directory with git init. This created a hidden .git folder, enabling Git to track changes and manage version control.

3. To avoid tracking unnecessary files like .Rproj.user/, I created a .gitignore file and added .Rproj.user/ to it. I verified it with git status, ensuring it no longer appeared as untracked.

4. I staged all files using git add . and committed them with git commit -m "Initial push".

5. On GitHub, I created a new repository (without README, .gitignore, or license), then copied its SSH URL.

6. I used git branch to view the current branch and git log –oneline for a concise commit history.

7. I linked the local repo to GitHub using: git remote add origin git@github.com:serl0001/ETC5513Assignment2.git

8. Finally, I pushed the project with: git push -u origin master This uploaded all local files to the GitHub repository.
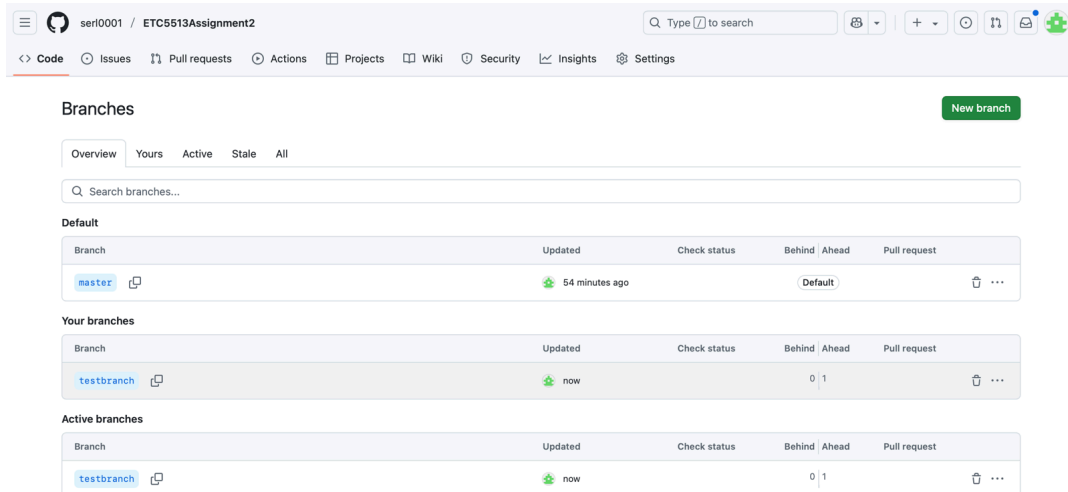
Step 3 : Initialising Git and Pushing it to GitHub

1. I Opened my terminal and ran the command : "git switch -c testbranch" . This creates a new branch called testbranch and switches to it.

2. In the example.qmd in RStudio I add a few changes and save the document.

3. Then I add and commit the modified files using the commands: "git add example.qmd git commit -m 'Add new section to example.qmd on testbranch'"

4. Then I Push the testbranch to the remote GitHub repository using the command : "git push -u origin testbranch" . The testbranch and its changes are now on GitHub.

This can be confirmed by visiting my GitHub repo and checking under the "branches" dropdown.

```
knitr::include_graphics("Screenshots/Screenshot3.png")
```


Step 4 : Creating a Data Folder, Adding Assignment 1 Data, and Amending Previous Commit

1. Created a Data folder in my system and added the Data file from Assignment 1 to that folder. The csv file is called "child-mortality.csv"

2. Staged the new folder and contents using the command: git add Data

3. **Amended with a new commit message : git commit –amend -m 'Add data folder from Assignment 1 to the previous commit'**
Include the staged data folder in the last commit and replaces the old message with the new one

4. Pushed the amended commit to GitHub Because I rewrote the last commit, I need to force push : git push –force

```
knitr::include_graphics("Screenshots/Screenshot4.png")
```

🖈 Pin | 👁 Unwatch 1 ▾ | ⑂ Fork 0 ▾

⑂ **testbranch** had recent pushes 3 minutes ago | **Compare & pull request**

⑂ testbranch ▾ | ⑂ 2 Branches | ◌ 0 Tags | 🔍 Go to file | t | Add file ▾ | <> Code ▾

This branch is 1 commit ahead of `master` . | ⇄ Contribute ▾

Sandhya and Sandhya Add data folder from Assignment 1 to the previous commit | e0e1fbb · 3 minutes ago | 🕐 3 Commits

📁 Data	Add data folder from Assignment 1 to the previous commit	3 minutes ago
📁 Screenshots	Initial Push	1 hour ago
📁 example_files	Initial Push	1 hour ago
🗎 .DS_Store	Initial Push	1 hour ago

About

No description, website provided.

-∿- Activity
☆ 0 stars
👁 1 watching
⑂ 0 forks

Releases

No releases published
Create a new release

Packages

Step 5: Switch to master, create a conflicting change, and push it

1. Switched back to the master branch using the command : "git switch master". Which switches to the master branch and the changes made int the test branch are not visible

2. Modified example.qmd in a conflicting way . Now, in main, I edited the same part of example.qmd (e.g., the same section heading) to something different and saved the qmd file.

3. Now I add and commit and push the changes to the remote repository via the terminal : git add . git commit -m "Modify example.qmd in a conflicting way on main branch" git push origin master

4. I've now created a conflict scenario by editing the same part of example.qmd on both main and testbranch.

```
knitr::include_graphics("Screenshots/Screenshot5.png")
```

```
sandhya@CHANDRASEKARs-MacBook-Pro ETC5513Assignment2 % git branch
  master
* testbranch
sandhya@CHANDRASEKARs-MacBook-Pro ETC5513Assignment2 % git switch master
Switched to branch 'master'
Your branch is up to date with 'origin/master'.
sandhya@CHANDRASEKARs-MacBook-Pro ETC5513Assignment2 % git add .
sandhya@CHANDRASEKARs-MacBook-Pro ETC5513Assignment2 % git commit -m 'modified example.qmd in a conflicting way'
[master 9ee554f] modified example.qmd in a conflicting way
 Committer: Sandhya <sandhya@CHANDRASEKARs-MacBook-Pro.local>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly:

    git config --global user.name "Your Name"
    git config --global user.email you@example.com

After doing this, you may fix the identity used for this commit with:

    git commit --amend --reset-author

 1 file changed, 4 insertions(+)
sandhya@CHANDRASEKARs-MacBook-Pro ETC5513Assignment2 % git push origin master
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 385 bytes | 385.00 KiB/s, done.
Total 3 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To github.com:serl0001/ETC5513Assignment2.git
   1c3212b..9ee554f  master -> master
```

Step 6: Merge testbranch into main and Resolve the Conflict

1. Merged testbranch into main using the command : git merge testbranch. The following screens shot displays the message that is thrown.

```
knitr::include_graphics("Screenshots/Screenshot6.png")
```

```
sandhya@CHANDRASEKARs-MacBook-Pro ETC5513Assignment2 % git branch
* master
  testbranch
sandhya@CHANDRASEKARs-MacBook-Pro ETC5513Assignment2 % git merge testbranch

Auto-merging example.qmd
CONFLICT (content): Merge conflict in example.qmd
Automatic merge failed; fix conflicts and then commit the result.
sandhya@CHANDRASEKARs-MacBook-Pro ETC5513Assignment2 %
```

Git has found a merge conflict in example.qmd.

2. I open and fix the conflict in example.qmd . Git marks the conflict like this:

```
knitr::include_graphics("Screenshots/Screenshot7.png")
```

```
<<<<<<< HEAD
STEP 4 AND STEP 5 ARE MISSING

THEY ARE PREST IN TEST BRANCH
=======
5. I created a new repository on GitHub without initializing it with a README, .gitignore, or license. Then, I copied the
repository's SSH URL to link it with my local project.

6. I used "git branch" to view all the branches in my repository and identify the currently active one, which is marked
```

3. I then fixed the conflicts bu removing the marked lines and conflict marker.

4. And then I add , Commit and Push it to the master branch.

Step 7: Create and Push Annotated Tag v1.0 on main

1. I verified I was on the master branch using the command: git branch The current branch is marked with an asterisk (*), confirming I was on master.

2. I created an annotated tag with: git tag -a v1.0 -m "First stable version after merging testbranch" This tag includes a custom message and stores metadata like my name, email, and the date, marking a stable point in the commit history.

3. I then pushed the tag to GitHub using: git push origin v1.0 This makes the v1.0 tag available in the remote repository for reference or
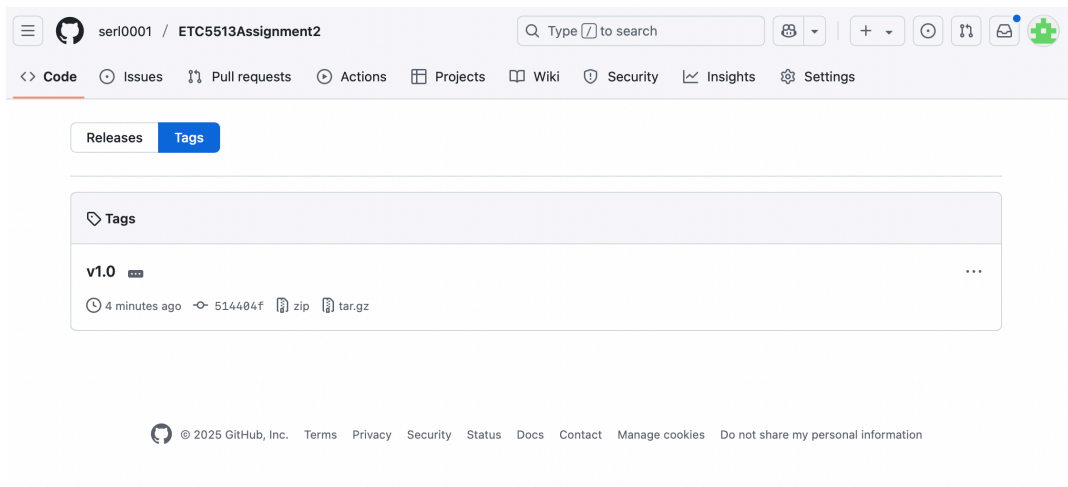
releases.

```
knitr::include_graphics("Screenshots/Screenshot8.png")
```

```
[sandhya@CHANDRASEKARs-MacBook-Pro ETC5513Assignment2 % git branch
* master
  testbranch
[sandhya@CHANDRASEKARs-MacBook-Pro ETC5513Assignment2 % git tag -a v1.0 -m "First stable version after merging testbranch"

[sandhya@CHANDRASEKARs-MacBook-Pro ETC5513Assignment2 % git push origin v1.0

Enumerating objects: 1, done.
Counting objects: 100% (1/1), done.
Writing objects: 100% (1/1), 206 bytes | 206.00 KiB/s, done.
Total 1 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:serl0001/ETC5513Assignment2.git
 * [new tag]         v1.0 -> v1.0
[sandhya@CHANDRASEKARs-MacBook-Pro ETC5513Assignment2 %
```

```
knitr::include_graphics("Screenshots/Screenshot9.png")
```


Step 8: Delete testbranch Locally and Remotely

1. For deleting the testbranch locally, I should not be in the testbranch (since the branch is already pointing to the master branch I leave it as it is). Using the command : git branch -d testbranch - I can delete the test branch locally

```
knitr::include_graphics("Screenshots/Screenshot10.png")
```

```
[sandhya@CHANDRASEKARs-MacBook-Pro ETC5513Assignment2 % git branch
* master
  testbranch
[sandhya@CHANDRASEKARs-MacBook-Pro ETC5513Assignment2 % git branch -d testbranch

Deleted branch testbranch (was 7960142).
[sandhya@CHANDRASEKARs-MacBook-Pro ETC5513Assignment2 % git branch
* master
[sandhya@CHANDRASEKARs-MacBook-Pro ETC5513Assignment2 %
```
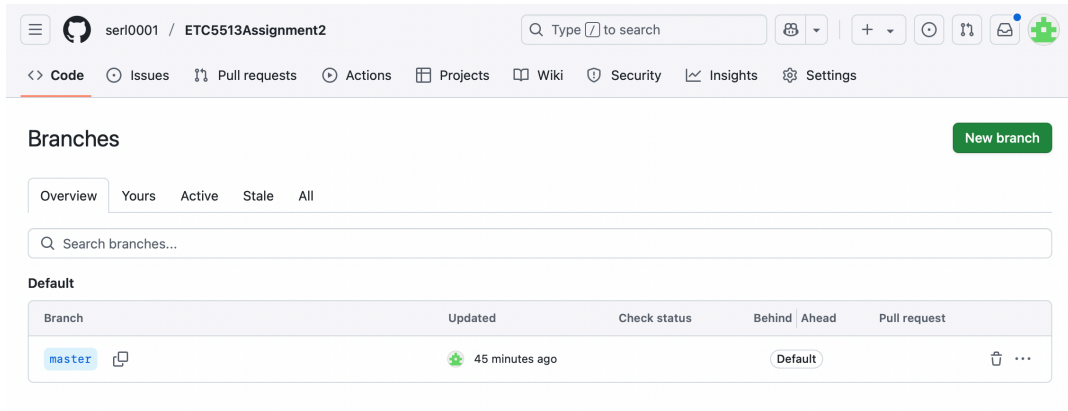
2. For deleting the testbranch on GitHub (remote) : git push origin –delete testbranch : removes testbranch from my GitHub repository.

```
knitr::include_graphics("Screenshots/Screenshot11.png")
```

```
sandhya@CHANDRASEKARs-MacBook-Pro ETC5513Assignment2 % git push origin --delete testbranch

To github.com:serl0001/ETC5513Assignment2.git
 - [deleted]         testbranch
sandhya@CHANDRASEKARs-MacBook-Pro ETC5513Assignment2 %
```

```
knitr::include_graphics("Screenshots/Screenshot12.png")
```


Step 9: The commit log in condensed form in the terminal

1. To show the commit log in condenced form in the terminall used the command : git log –oneline –graph –decorate –all

Which shows:

–oneline: Each commit appears as a single line

–graph: Visualizes branch and merge structure

–decorate: Shows branch names and tags

–all: Includes all branches, not just the current one

```
knitr::include_graphics("Screenshots/Screenshot13.png")
```
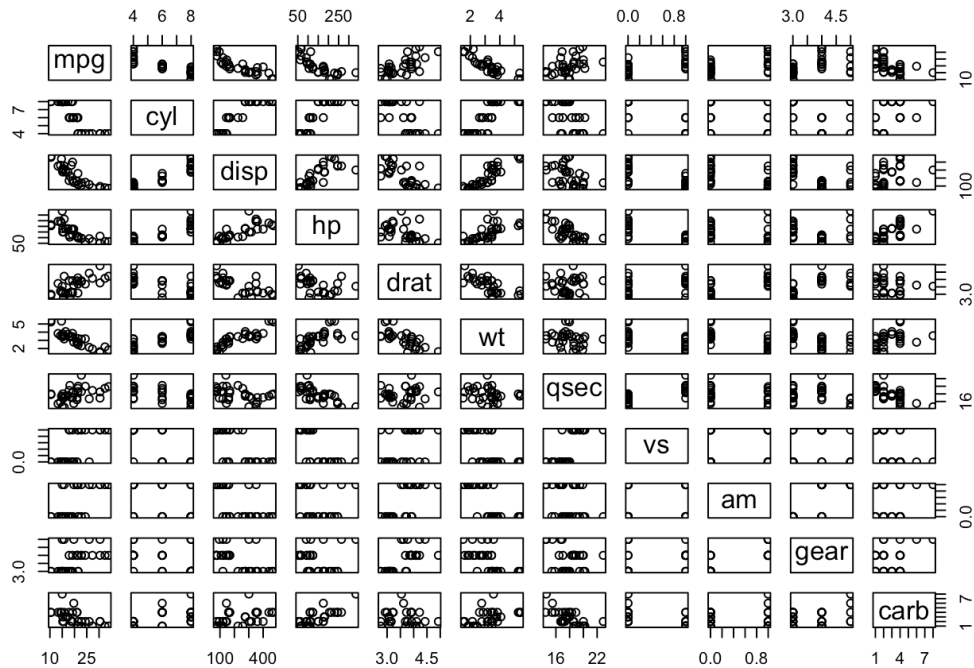
```
sandhya@CHANDRASEKARs-MacBook-Pro ETC5513Assignment2 % git log --oneline --graph --decorate --all
* 5709150 (HEAD -> master, origin/master) Adding all the final changes
*   514404f (tag: v1.0) Resolved merge conflict in example.qmd between main and testbranch
|\
| * 7960142 saving changes in test branch
| * e0e1fbb Add data folder from Assignment 1 to the previous commit
* | 9ee554f modified example.qmd in a conflicting way
|/
* 1c3212b Initial Commit
* 30ee035 Initial Push
sandhya@CHANDRASEKARs-MacBook-Pro ETC5513Assignment2 %
```

Step 10: Add a Plot in example.qmd, Commit It, Then Undo the Commit (Keep Changes)

1.Made a Change in example.qmd and added this new section :

```r
plot(mtcars)
```


2. Add and committed the changes throught the terminal.

```r
knitr::include_graphics("Screenshots/Screenshot14.png")
```

```
[sandhya@CHANDRASEKARs-MacBook-Pro ETC5513Assignment2 % git add .
[sandhya@CHANDRASEKARs-MacBook-Pro ETC5513Assignment2 % git commit -m 'Add simple plot section to example.qmd'
[master 3e8c5a0] Add simple plot section to example.qmd
 Committer: Sandhya <sandhya@CHANDRASEKARs-MacBook-Pro.local>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly:

    git config --global user.name "Your Name"
    git config --global user.email you@example.com

After doing this, you may fix the identity used for this commit with:

    git commit --amend --reset-author

 3 files changed, 43 insertions(+)
 create mode 100644 Screenshots/Screenshot13.png
[sandhya@CHANDRASEKARs-MacBook-Pro ETC5513Assignment2 % git log --oneline
3e8c5a0 (HEAD -> master) Add simple plot section to example.qmd
5709150 (origin/master) Adding all the final changes
514404f (tag: v1.0) Resolved merge conflict in example.qmd between main and testbranch
9ee554f modified example.qmd in a conflicting way
7960142 saving changes in test branch
e0e1fbb Add data folder from Assignment 1 to the previous commit
1c3212b Initial Commit
30ee035 Initial Push
```

3. To Undo the Commit (But Keep Your Changes) I Used: –soft reset to undo the commit, but keep all staged changes: git reset –soft HEAD~1 This command:

Undo the last commit

Leaves your changes staged (still tracked and not lost)

```r
knitr::include_graphics("Screenshots/Screenshot15.png")
```

```
[sandhya@CHANDRASEKARs-MacBook-Pro ETC5513Assignment2 % git reset --soft HEAD~1

[sandhya@CHANDRASEKARs-MacBook-Pro ETC5513Assignment2 % git log --oneline
5709150 (HEAD -> master, origin/master) Adding all the final changes
514404f (tag: v1.0) Resolved merge conflict in example.qmd between main and testbr
9ee554f modified example.qmd in a conflicting way
7960142 saving changes in test branch
e0e1fbb Add data folder from Assignment 1 to the previous commit
1c3212b Initial Commit
30ee035 Initial Push
[sandhya@CHANDRASEKARs-MacBook-Pro ETC5513Assignment2 %
```