

Pre-conference course: Let's git it started

2024 ISPE Annual Meeting: August 25th, Berlin

Interactive Git Workshop: Instructions

Setup:

Please ensure you have followed the [setup steps](#) for participating in this workshop before attempting the tasks described in Parts 1-3.

Helpful resources:

The following resources, from the 'Introduction to Git' session, may be helpful for completing the tasks in Parts 1-3:

- [Git glossary](#)
- ['How to' for common tasks](#)

Part 1: Creating and cloning a repository

You have recently joined a team where your manager has asked you to use Git for at the start of a new project, investigating the effectiveness of proton pump inhibitors (PPIs).

You will first have to **create a repository** for this project and then **clone this repository** to your local computer.

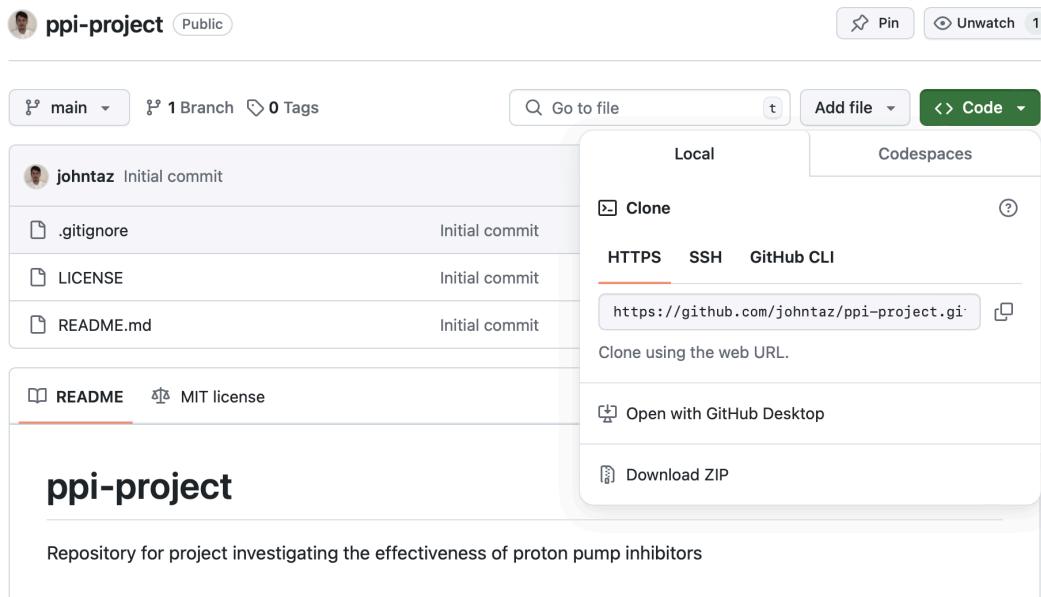
Create a repository

1. Navigate to [GitHub.com](#) and **login** to your account.
2. Find and click the '**New**' repository button (shown to the right)
3. Provide a **repository name** and **description**
 - E.g 'ppi-project' and 'Repository for project investigating the effectiveness of proton pump inhibitors'
4. Tick the box '**Public**' under repository visibility options
 - Note: this can be changed later and some projects, in practice, may start 'Private' and then become 'Public' at the point of pre-printing/publication.
5. Tick the box '**Add a README file**'
6. Under **.gitignore**, select 'R' from the dropdown menu.
 - Note: you might initially select .gitignore based on the language you anticipate coding the most in and update, as necessary, if other languages are also used.
7. Under **license**, select 'MIT' from the dropdown menu.
8. Click '**Create repository**'



Clone this repository

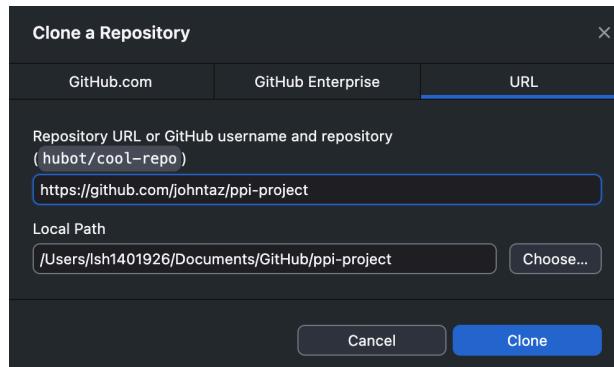
1. On **GitHub.com**, navigate to the **main page of the repository**
 - a. Note: if following directly from the previous step - this will be the current page.
2. Above the list of files, click '**Code**' and a drop-down menu should appear (see below)



3. Follow the steps below for **either** GitHub Desktop or Command Line

GitHub Desktop

1. Click '**Open with GitHub Desktop**'
2. Check the **repository URL** refers to ["https://github.com/YOUR-USERNAME/YOUR-REPOSITORY"](https://github.com/YOUR-USERNAME/YOUR-REPOSITORY) and specify your own **local path** (the folder where the repository will be cloned to). Click '**Clone**'.



Command Line



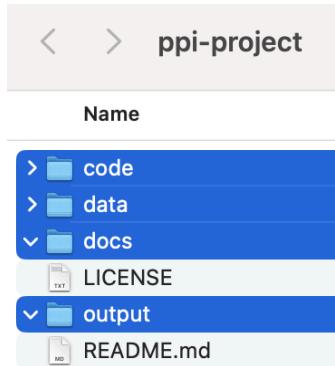
1. Click the '**Copy**' button next to the '**HTTPS address** (see right)
2. Open Command Line/Terminal
4. Change the current working directory to the location where you want the cloned repository
 - a. E.g 'cd "Users/lsh1401926/Documents/GitHub/”'
 - b. Type '**git clone**', and then **paste the URL** you copied in Step 6. Press **Enter** (see below for steps 8 and 9)

```
(base) MACB0180:~ lsh1401926$ cd ./Desktop/  
  
(base) MACB0180:Desktop lsh1401926$ git clone https://github.com/johntaz/ppi-project.git  
Cloning into 'ppi-project'...  
remote: Enumerating objects: 8, done.  
remote: Counting objects: 100% (8/8), done.  
remote: Compressing objects: 100% (8/8), done.  
remote: Total 8 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)  
Receiving objects: 100% (8/8), done.  
Resolving deltas: 100% (1/1), done.
```

Part 2: Initial Commit and Push

Your manager sends you some files that a previous analyst developed for a similar project and asks you to **add these files to your repository**.

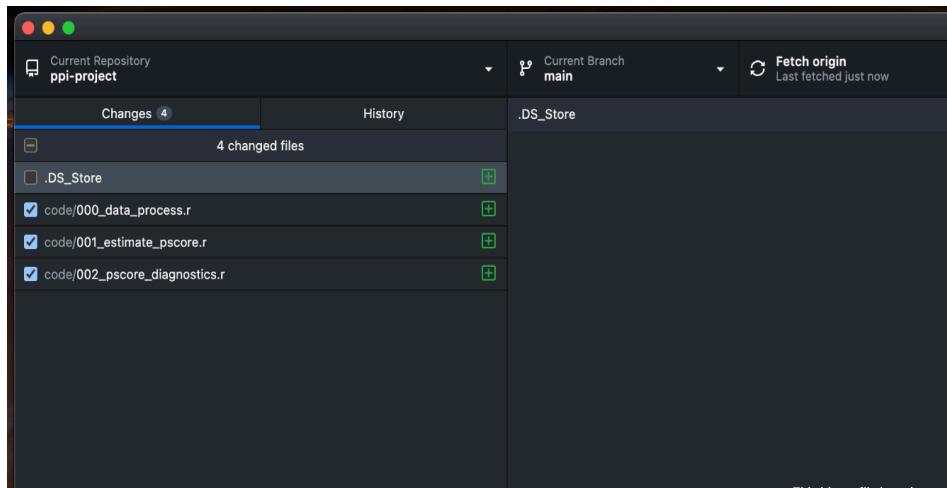
1. **Navigate** to the folder where you downloaded the 'Project files' zip folder (obtained from the Course Webpage) and unzip this folder - you should see 4 folders named "code", "data", "docs", "output".
2. **Copy or Move** these folders to the location where you cloned the repository.



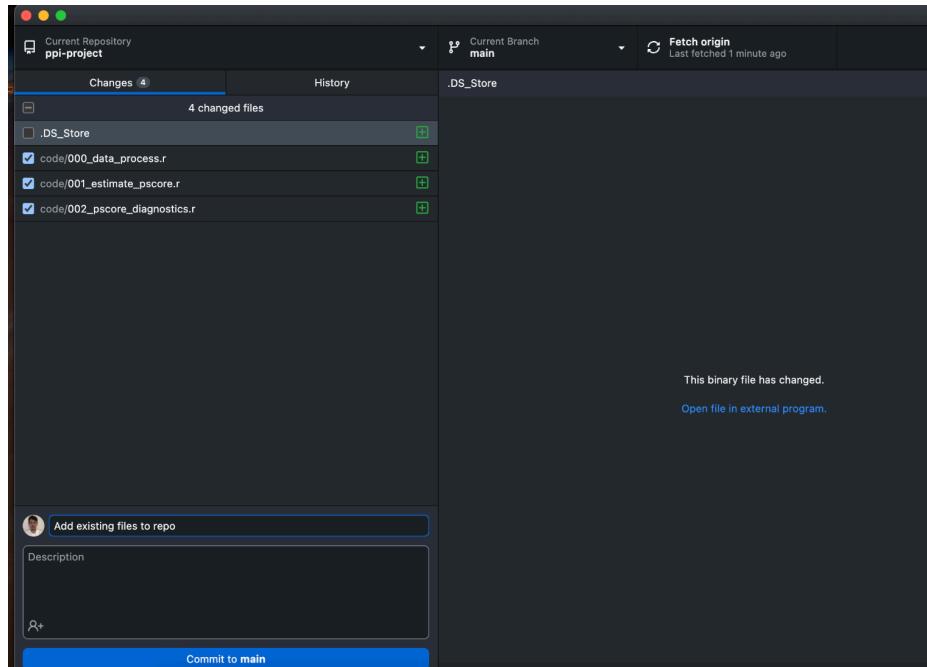
5. Follow the steps below for **either** GitHub Desktop or Command Line

GitHub Desktop

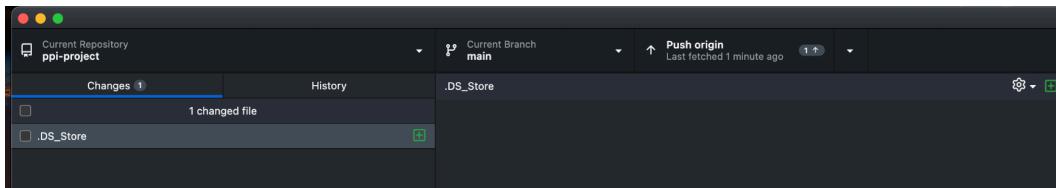
1. Return to GitHub Desktop. You should now see the code files in the ‘Change Files’ window. Make sure these are ticked, the ‘ticking’ indicates the files are **staged**.



2. Add a ‘Commit’ message e.g. “Add existing files to repo”. Click ‘Commit to main’.



- Click **Push Origin** in the top right-hand corner within the repository bar. Go check your repository on GitHub.com and see the files!



Command Line

- Ensure your working directory is the local Git repository (usually on the command line after cloning you will need to do this).
- Type 'git status' into the console. You can see there are untracked files in the 'code' folder, but not what these are.

```
(base) MACB0180:Desktop lsh1401926$ cd ./ppi-project
(base) MACB0180:ppi-project lsh1401926$ git status
On branch main
Your branch is up to date with 'origin/main'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    .DS_Store
    code/
nothing added to commit but untracked files present (use "git add" to track)
(base) MACB0180:ppi-project lsh1401926$
```

- Type 'git add code/*' into the console.
 - Note: We would not usually use a wildcard (*) as this may risk committing a file that you do not want to commit. As we have already inspected the contents of the folder, for the purposes of this workshop, this is a convenient shortcut.
- Type 'git status' again into the console. This should show all the 3 scripts in green.
 - Note: The fact these scripts are displayed in green indicates they are **staged**.

```
nothing added to commit but untracked files present (use "git add" to track)
(base) MACB0180:ppi-project lsh1401926$ git add code/*
(base) MACB0180:ppi-project lsh1401926$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   code/000_data_process.r
    new file:   code/001_estimate_pscore.r
    new file:   code/002_pscore_diagnostics.r

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    .DS_Store
    code/.DS_Store

(base) MACB0180:ppi-project lsh1401926$
```

5. Type 'git commit -m "Add existing files to repo"' into the console.

```
(base) MACB0180:ppi-project lsh1401926$ git commit -m 'Add existing files to repo'  
[main 00d60b8] Add existing files to repo  
 3 files changed, 51 insertions(+)  
   create mode 100644 code/000_data_process.r  
   create mode 100644 code/001_estimate_pscore.r  
   create mode 100644 code/002_pscore_diagnostics.r  
(base) MACB0180:ppi-project lsh1401926$ █
```

6. Type 'git push origin main' into the console. Go check your repository on GitHub.com and see the files!

```
(base) MACB0180:ppi-project lsh1401926$ git push origin main  
Enumerating objects: 8, done.  
Counting objects: 100% (8/8), done.  
Delta compression using up to 8 threads  
Compressing objects: 100% (6/6), done.  
Writing objects: 100% (6/6), 1.17 KiB | 300.00 KiB/s, done.  
Total 6 (delta 1), reused 0 (delta 0), pack-reused 0  
remote: Resolving deltas: 100% (1/1), completed with 1 local object.  
To https://github.com/johnnaz/ppi-project.git  
 7a7507d..efdcfe1 main -> main  
(base) MACB0180:ppi-project lsh1401926$ █
```

Part 3: Editing/Updating files

At the next meeting, your manager mentions that this project will be able to use a similar propensity score model to the previous project.

However, in this project, it is no longer reasonable to adjust for history of Diabetes ('prior_diabetes') but instead, they ask you to **edit and update a file in your repository** to include a variable representing history of Chronic Pulmonary Obstructive Disease (COPD).

GitHub Desktop

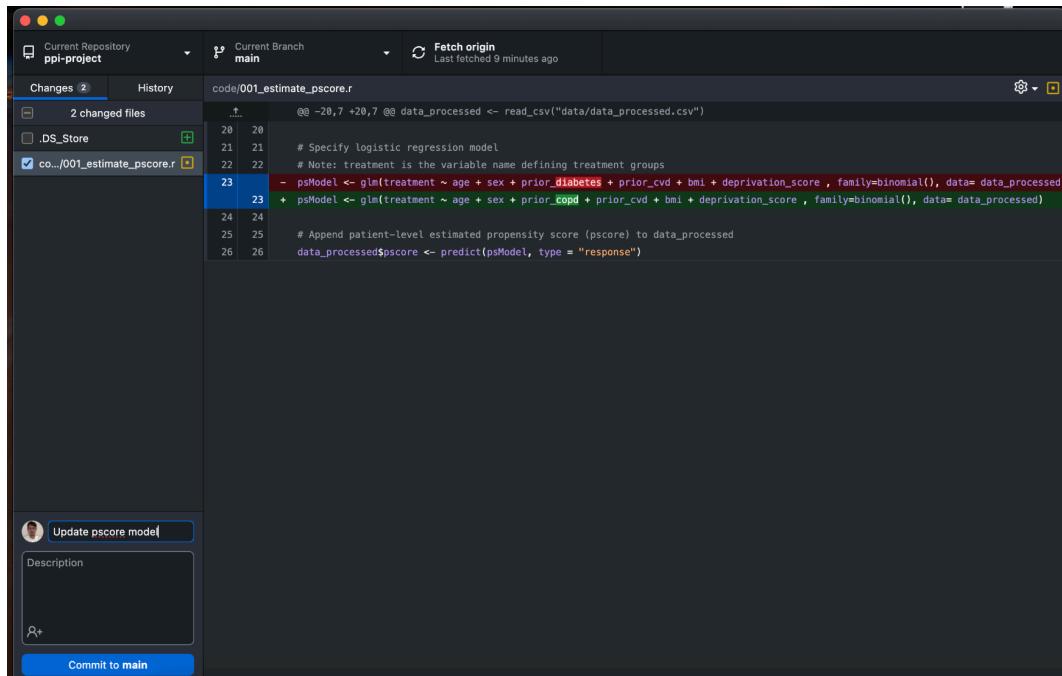
1. Return to GitHub Desktop and click '**Fetch origin**' (nothing should happen if the files are up-to-date, but this is a good habit!).
2. Navigate to your local repository. Open the 'code' subfolder and then open the '**'001_estimate_pscore.R'** file either in R Studio or any text editor available (Notepad,TextEdit, Visual Studio etc).
3. Find the propensity score model code (Line 23) and remove the variable '**prior_diabetes**'.

```
Users > lsh1401926 > Desktop > ppi-project > code > 001_estimate_pscore.r
1 ######
2 #
3 # 001: Estimate propensity score
4 #
5 # Author(s): John Tazare
6 # Purpose: This script is to import processed data and estimate propensity scores
7 # Outputs of this script:
8 # - Processed dataset: ./data/data_processed.rds
9 #
10 #####
11
12 # Load libraries -----
13 library(dplyr)
14 library(tidyverse)
15
16 # Load libraries -----
17 data_processed <- read_csv("data/data_processed.csv")
18
19 # Estimate propensity scores -----
20
21 # Specify logistic regression model
22 # Note: treatment is the variable name defining treatment groups
23 psModel <- glm(treatment ~ age + sex + prior_diabetes + prior_cvd + bmi + deprivation_score , family=binomial(), data= data_processed)
24
25 # Append patient-level estimated propensity score (pscore) to data_processed
26 data_processed$pscore <- predict(psModel, type = "response")
```

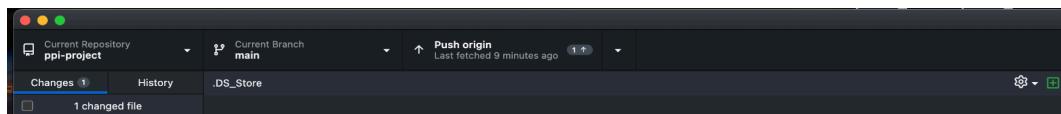
4. Replace this with '**prior_copd**' and **Save** the updated file (do not change the file name).

```
Users > lsh1401926 > Desktop > ppi-project > code > 001_estimate_pscore.r
1 ######
2 #
3 # 001: Estimate propensity score
4 #
5 # Author(s): John Tazare
6 # Purpose: This script is to import processed data and estimate propensity scores
7 # Outputs of this script:
8 # - Processed dataset: ./data/data_processed.rds
9 #
10 #####
11
12 # Load libraries -----
13 library(dplyr)
14 library(tidyverse)
15
16 # Load libraries -----
17 data_processed <- read_csv("data/data_processed.csv")
18
19 # Estimate propensity scores -----
20
21 # Specify logistic regression model
22 # Note: treatment is the variable name defining treatment groups
23 psModel <- glm(treatment ~ age + sex + prior_copd + prior_cvd + bmi + deprivation_score , family=binomial(), data= data_processed)
24
25 # Append patient-level estimated propensity score (pscore) to data_processed
26 data_processed$pscore <- predict(psModel, type = "response")
```

5. You should now see the edited file in the '**Change Files**' window. Check this file is ticked (**staged**). The figure on the next page shows steps 5-7.
6. Check the correct variable has been added in the **diff** window.
7. Add a '**Commit message**', e.g. "Update pscore model". Click '**Commit to main**'.



8. Click **Push Origin** in the top right-hand corner within the repository bar.



9. Visit the repository at GitHub.com and navigate to the updated file.

The screenshot shows a GitHub repository page for 'ppi-project'. The file '001_estimate_pscore.r' is displayed. The commit history shows a recent commit by 'johntaz' with the message 'Update pscore model'. The file content is as follows:

```

1 #####
2 #
3 # 001: Estimate propensity score
4 #
5 # Author(s): John Tazare
6 # Purpose: This script is to import processed data and estimate propensity scores
7 # Outputs of this script:
8 # - Processed dataset: ./data/data_processed.rds
9 #
10 #####
11 #
12 # Load libraries -----
13 library(dplyr)
14 library(tidyverse)
15 #
16 # Load libraries -----
17 data_processed <- read_csv("data/data_processed.csv")
18 #
19 # Estimate propensity scores -----
20 #
21 # Specify logistic regression model
22 # Note: treatment is the variable name defining treatment groups
23 psModel <- glm(treatment ~ age + sex + prior_copd + prior_cvd + bmi + deprivation_score , family=binomial(), data= data_processed)
24 #
25 # Append patient-level estimated propensity score (pscore) to data_processed
26 data_processed$pscore <- predict(psModel, type = "response")

```

The word 'prior_copd' in the 23rd line of the code is highlighted in red, indicating it was added during the commit.

Command Line

1. Return to the console and type '**git pull**' (nothing should happen if the files are up-to-date, but this is a good habit!).
 - a. Note: if you have closed the console you will first need to ensure your working directory is the local Git repository.
2. Navigate to your local repository. Open the 'code' subfolder and then open the '**001_estimate_pscore.R**' file either in R Studio or any text editor available (Notepad,TextEdit, Visual Studio etc).
3. Find the propensity score model code (Line 23) and remove the variable '**prior_diabetes**'.

```
Users > lsh1401926 > Desktop > ppi-project > code > 001_estimate_pscore.r
1 #####
2 #
3 # 001: Estimate propensity score
4 #
5 # Author(s): John Tazare
6 # Purpose: This script is to import processed data and estimate propensity scores
7 # Outputs of this script:
8 # - Processed dataset: ./data/data_processed.rds
9 #
10 #####
11
12 # Load libraries -----
13 library(dplyr)
14 library(tidyverse)
15
16 # Load libraries -----
17 data_processed <- read_csv("data/data_processed.csv")
18
19 # Estimate propensity scores -----
20
21 # Specify logistic regression model
22 # Note: treatment is the variable name defining treatment groups
23 psModel <- glm(treatment ~ age + sex + prior_diabetes + prior_cvd + bmi + deprivation_score , family=binomial(), data= data_processed)
24
25 # Append patient-level estimated propensity score (pscore) to data_processed
26 data_processed$pscore <- predict(psModel, type = "response")
```

4. Replace this with '**prior_copd**' and **Save** the updated file (do not change the file name).

```
Users > lsh1401926 > Desktop > ppi-project > code > 001_estimate_pscore.r
1 #####
2 #
3 # 001: Estimate propensity score
4 #
5 # Author(s): John Tazare
6 # Purpose: This script is to import processed data and estimate propensity scores
7 # Outputs of this script:
8 # - Processed dataset: ./data/data_processed.rds
9 #
10 #####
11
12 # Load libraries -----
13 library(dplyr)
14 library(tidyverse)
15
16 # Load libraries -----
17 data_processed <- read_csv("data/data_processed.csv")
18
19 # Estimate propensity scores -----
20
21 # Specify logistic regression model
22 # Note: treatment is the variable name defining treatment groups
23 psModel <- glm(treatment ~ age + sex + prior_copd + prior_cvd + bmi + deprivation_score , family=binomial(), data= data_processed)
24
25 # Append patient-level estimated propensity score (pscore) to data_processed
26 data_processed$pscore <- predict(psModel, type = "response")
```

5. Type 'git status' into the console. You can see there are untracked changes to the '001_estimate_pscore.R' file. The next figure shows steps 5-8.
6. Type 'git add code/001_estimate_pscore.R" into the console.
7. Type 'git status' again into the console. This should show the script (001_estimate_pscore) in green.
8. Type 'git commit -m "update pscore model"' into the console.

```
(base) MACB0180:ppi-project lsh1401926$ git add code/001_estimate_pscore.r
(base) MACB0180:ppi-project lsh1401926$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    modified:   code/001_estimate_pscore.r

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    .DS_Store

(base) MACB0180:ppi-project lsh1401926$ git commit -m 'update pscore model'
[main b0fef0] update pscore model
 1 file changed, 1 insertion(+), 1 deletion(-)
(base) MACB0180:ppi-project lsh1401926$
```

9. Type 'git push origin main' into the console.

```
(base) MACB0180:ppi-project lsh1401926$ git push origin main
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 8 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 357 bytes | 357.00 KiB/s, done.
Total 4 (delta 3), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (3/3), completed with 3 local objects.
To https://github.com/johnnaz/ppi-project.git
  efdfcfe1..5dfef6e main -> main
(base) MACB0180:ppi-project lsh1401926$
```

10. Visit the repository at GitHub.com and navigate to the updated file.

```
ppi-project / code / 001_estimate_pscore.r

johnnaz Update pscore model
4cd4d525 · now History

Code Blame 26 lines (21 loc) · 1.07 KB

1 #####
2 #
3 # 001: Estimate propensity score
4 #
5 # Author(s): John Nazare
6 # Purpose: This script is to import processed data and estimate propensity scores
7 # Outputs of this script:
8 # - Processed dataset: ./data/data_processed.rds
9 #
10 #####
11
12 # Load libraries -----
13 library(dplyr)
14 library(tidyverse)
15
16 # Load libraries -----
17 data_processed <- read_csv("data/data_processed.csv")
18
19 # Estimate propensity scores -----
20
21 # Specify logistic regression model
22 # Note: treatment is the variable name defining treatment groups
23 psModel <- glm(treatment ~ age + sex + prior_cvd ~ prior_cvd * prior_cv + deprivation_score , family=binomial(), data= data_processed)
24
25 # Append patient-level estimated propensity score (pscore) to data_processed
26 data_processed$pscore <- predict(psModel, type = "response")
```