**Hands-on Lab: Download & Install R and RStudio**

Estimated time needed: **15** minutes

Multiple programmers are moving towards data science, and in this process, R and RStudio play an essential role. So in this lab, you will understand how to install R and RStudio.

**Objectives**

* Download and Install R
* Download and Install RStudio

**Overview of R and RStudio**

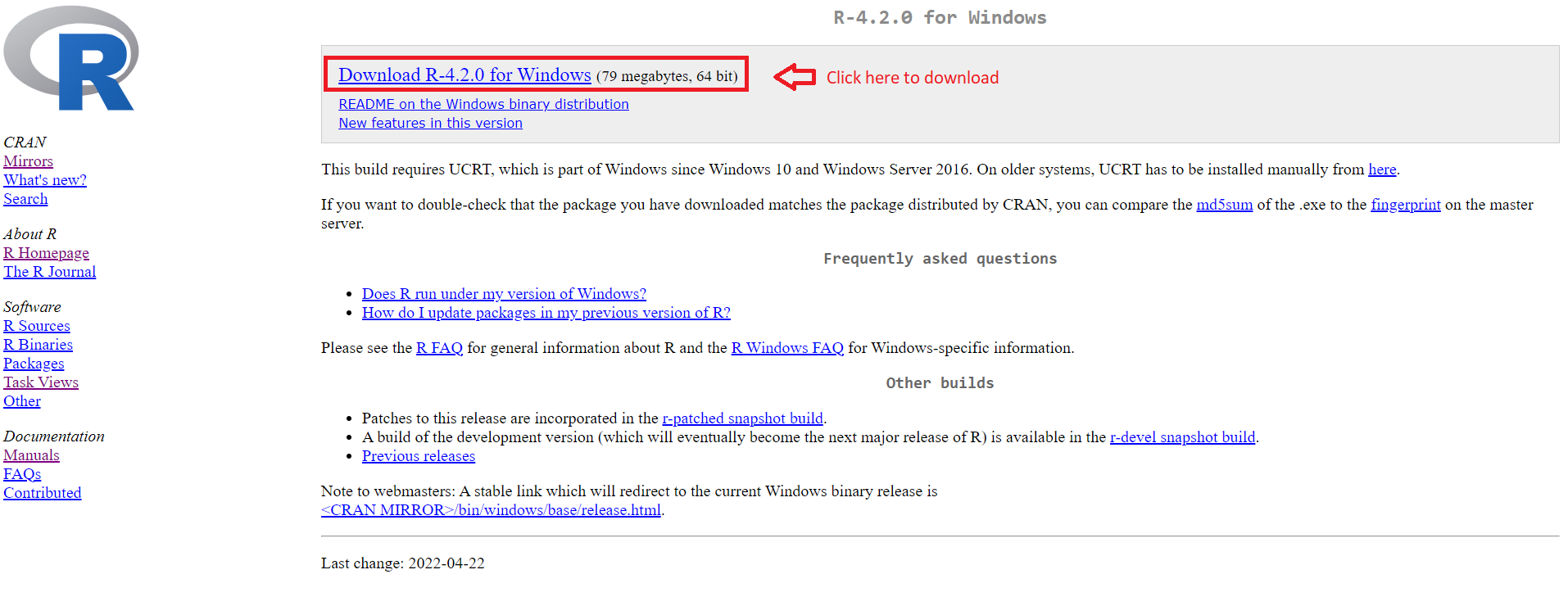
There are several cloud based data science tools that make team collaboration accessible. At times it is useful to work directly on your desktop.

R is a command-line interface; there are several graphical front-ends available. RStudio is an IDE (integrated development environment) for R. It includes the environmental tab, which shows the generated variables. In the history tab, you can see the commands used since starting, and there are other tabs such as files, plots, packages, help, and viewer. It has binaries available for major platforms, including Windows, Linux, and MacOS. This lab includes instructions for downloading and installing R and RStudio on Windows. Mac OS users can download the appropriate .pkg file from <https://cran.r-project.org/bin/macosx/> and follow the instructions.

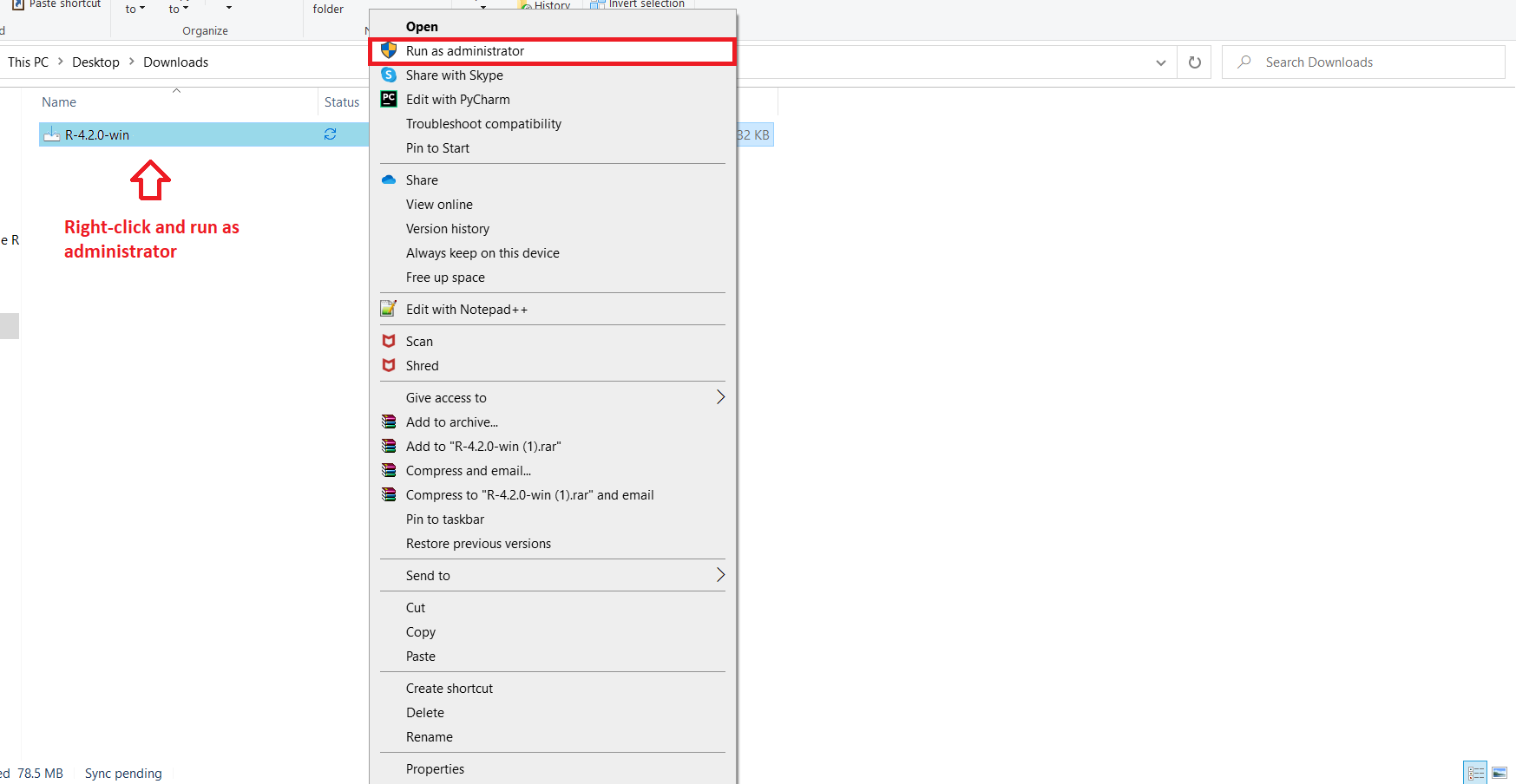
**Exercise 1: Download & Install R on Windows**

**Step 1**: The **latest version** of R can be downloaded by clicking the link.

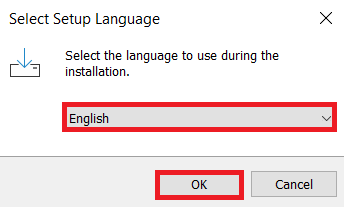
Windows: <https://cran.r-project.org/bin/windows/base/>



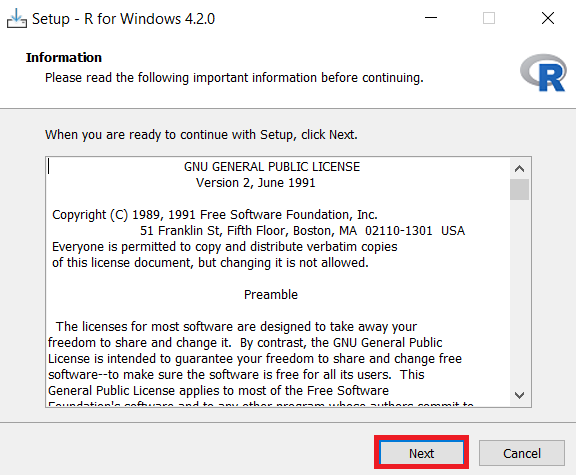
**Step 2**: Once the download completes, **right-click** the downloaded file, and click **Run as administrator**.



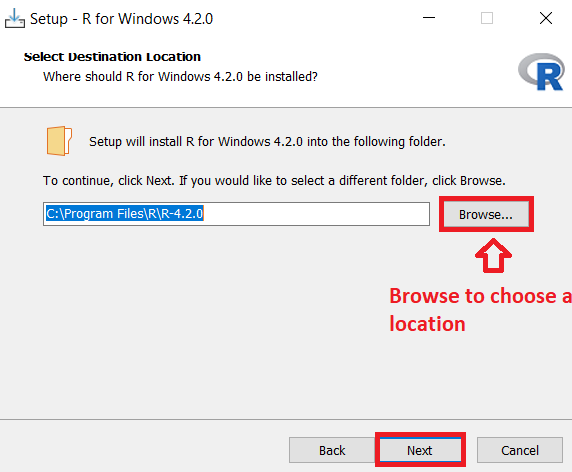
**Step 3**: Select your preferred installation **language**, and click **OK**.



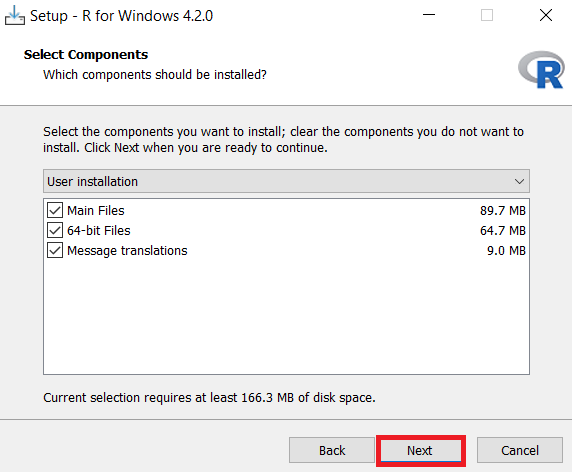
**Step 4**: Read and accept the license and click **Next**.



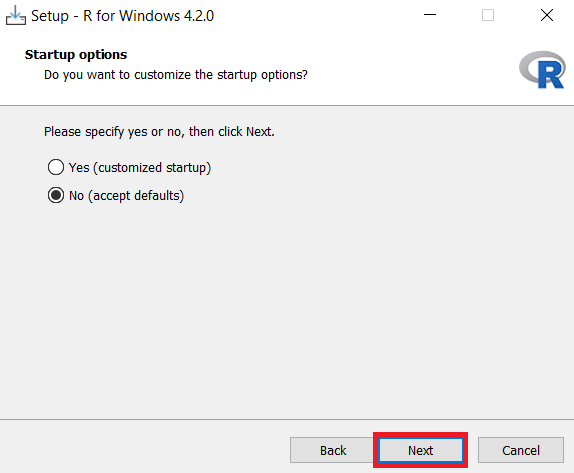
**Step 5**: Select the **Folder** where you would like to install R, or use the **Default** location, and click **Next**.



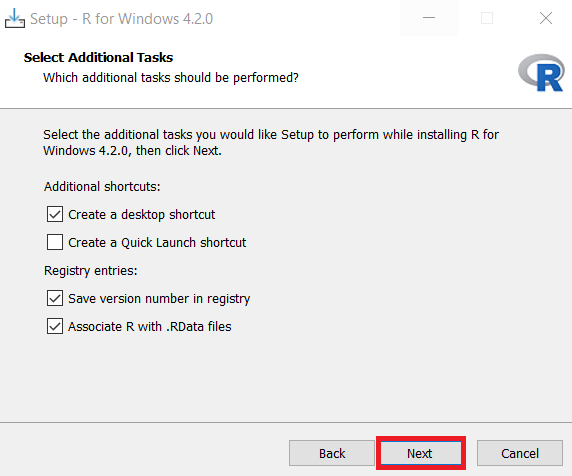
**Step 6**: Select the **Components** you want to install and click **Next**.



**Step 7**: In the **Startup options**, select the **Default** option and click **Next**.



**Step 8**: In the **Select Additional Tasks** window, retain **Default** and click **Next**.



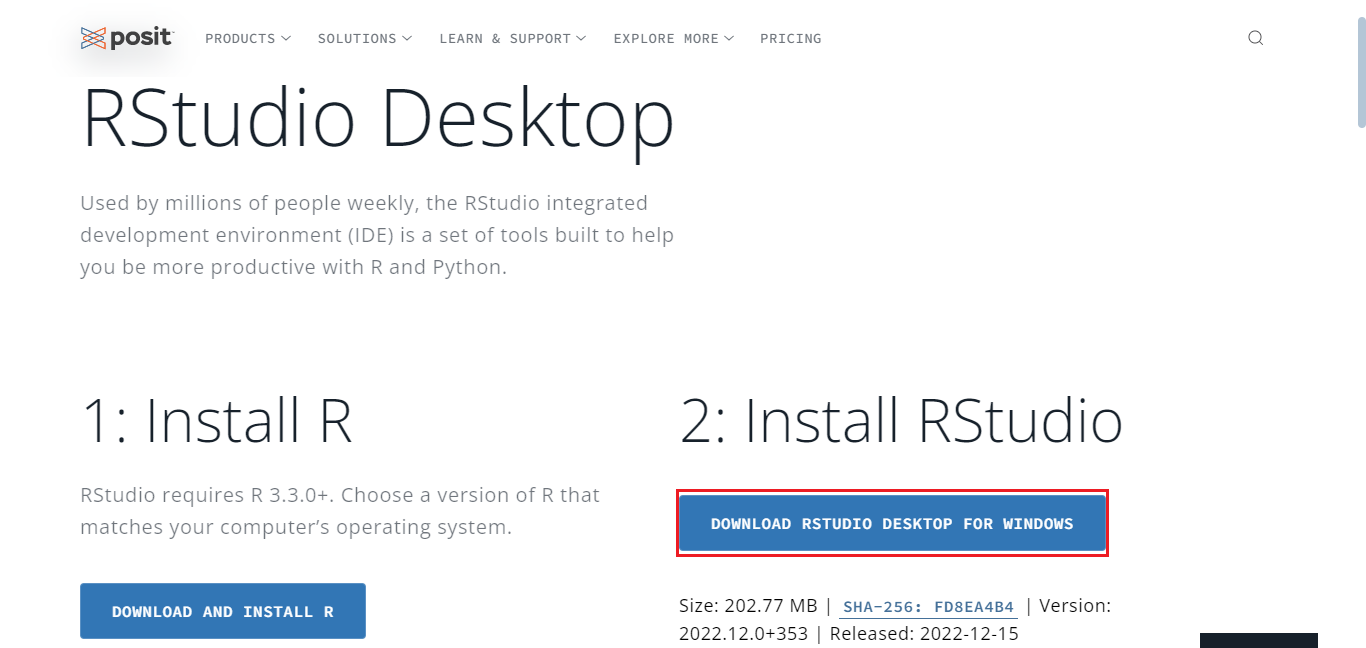
**Step 9**: Once installation is successful, click **Finish** to close the setup.

**Exercise 2: Download & Install RStudio**

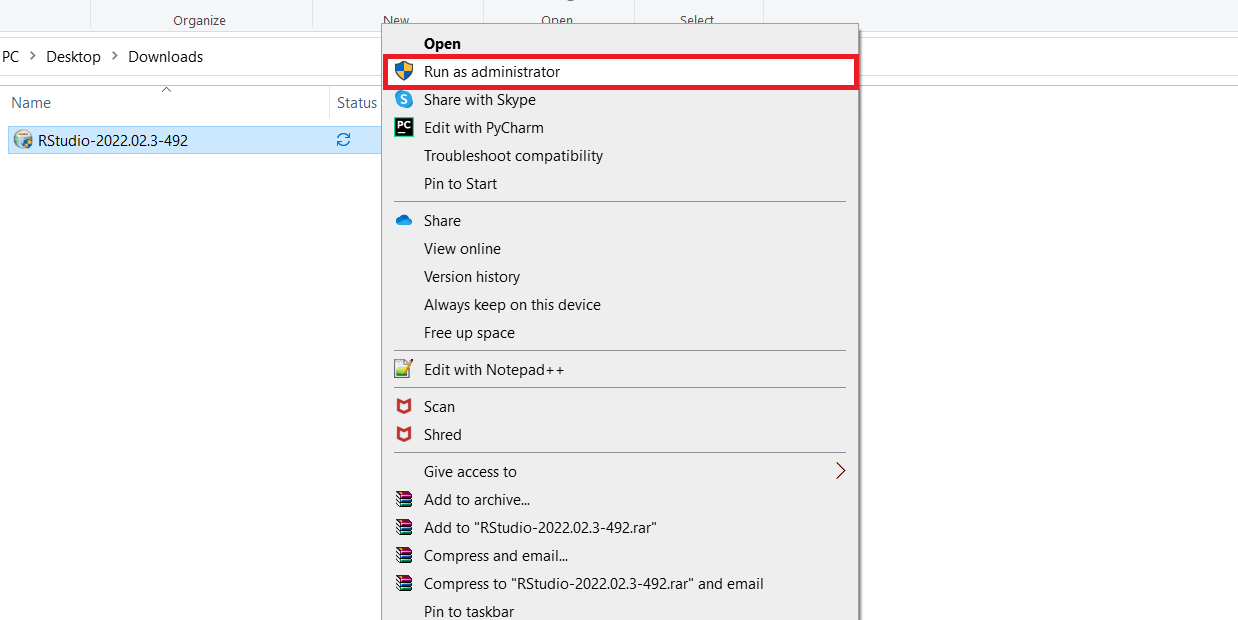
**Step 1**: Use the link below to download **RStudio Desktop** on your local machine.

Link for Download RStudio for windows and mac: <https://posit.co/download/rstudio-desktop/>

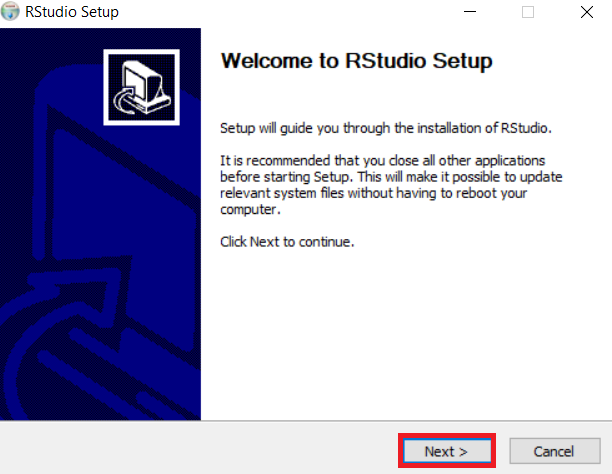
**Step 2**: Click **Download RStudio desktop For Windows**, and your download will start.



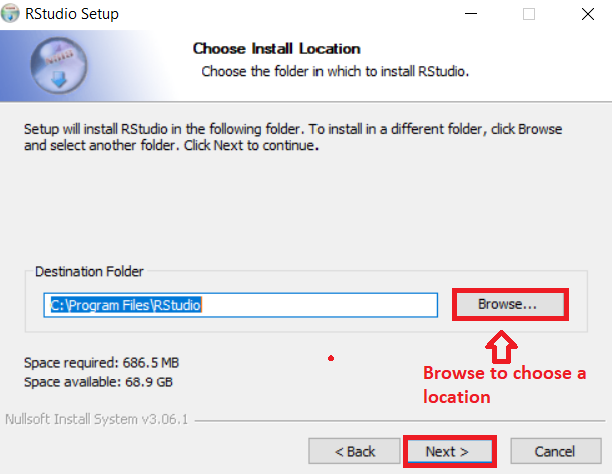
**Step 3**: Once the download completes, **right-click** the setup file, and click **Run as administrator**.



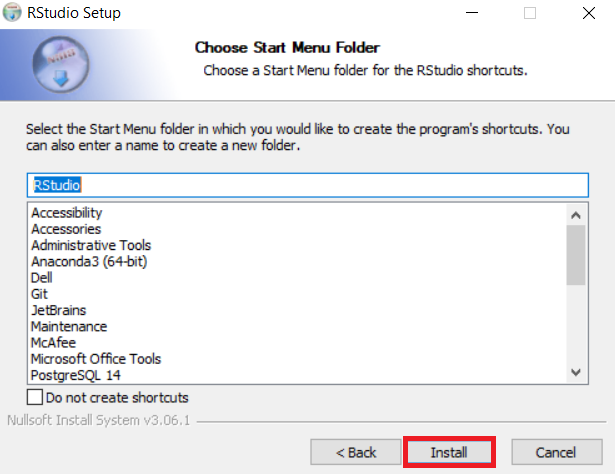
**Step 4**: In the RStudio setup window, click **Next**.



**Step 5**: Select the folder where you would like to install RStudio, or retain the **Default** installation location and click **Next**.



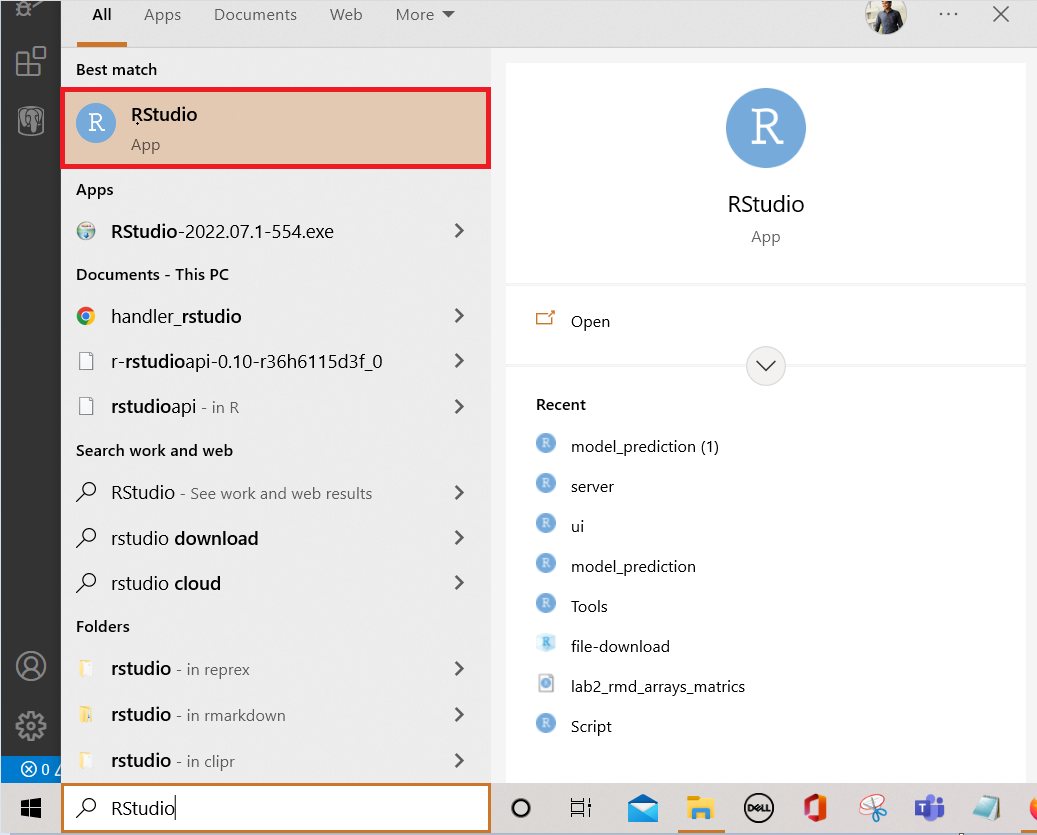
**Step 6**: In the Start menu window, click **Install** to install RStudio.



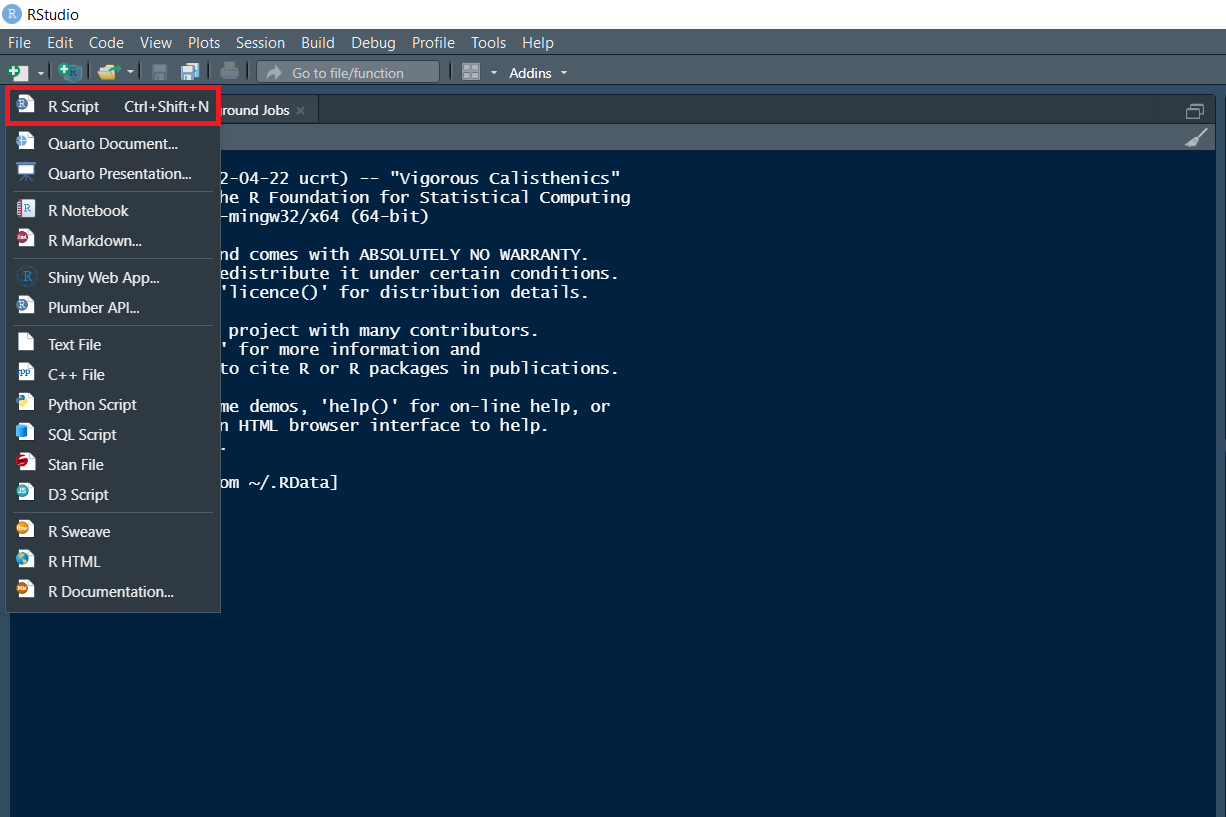
**Step 7**: Once installation completes, click **Finish** to close the window.

**Exercise 3: Execute R code in RStudio**

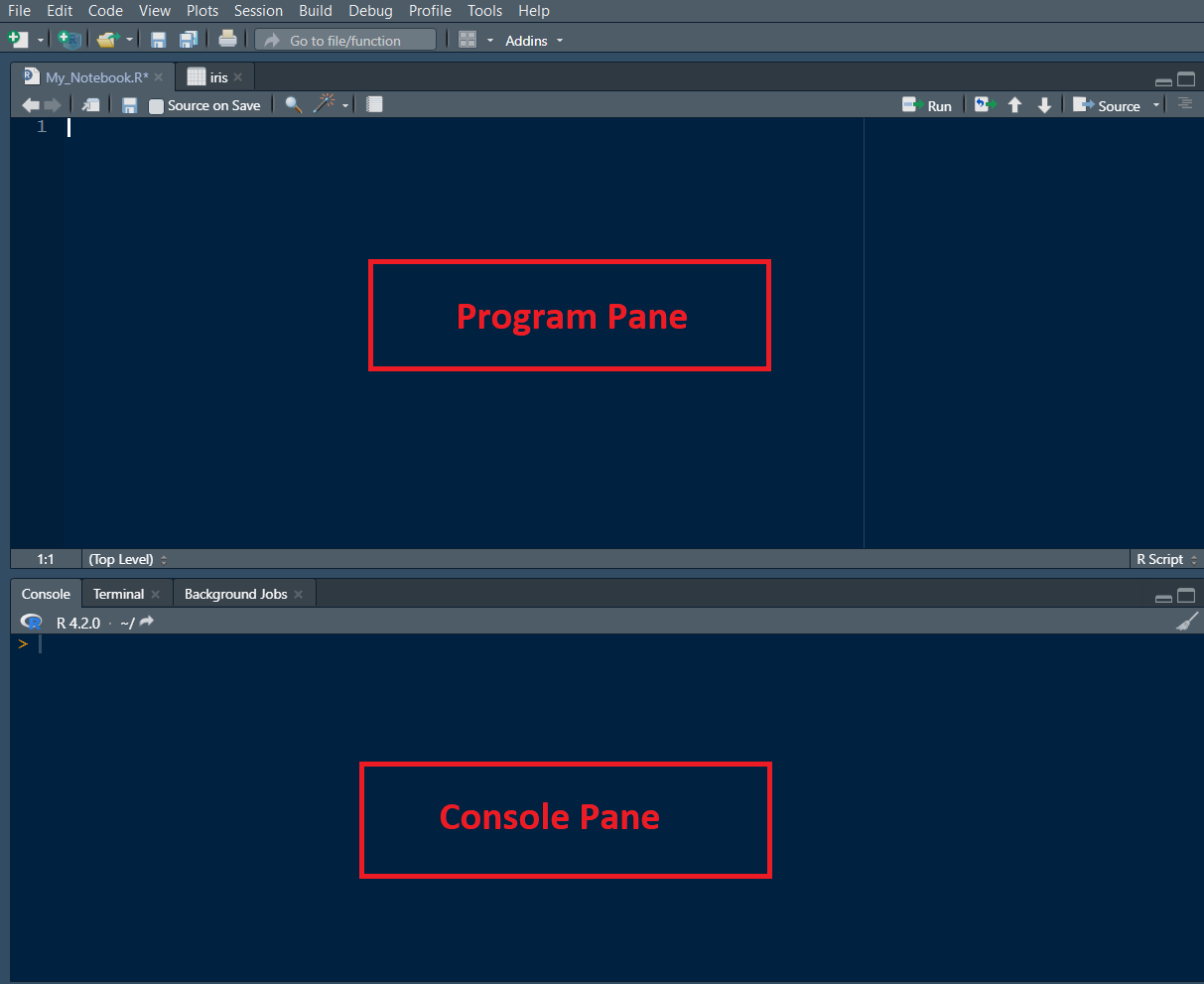
1. Open **RStudio** from the Windows start menu.



1. Click the **plus symbol** on the top left and select **R Script**.

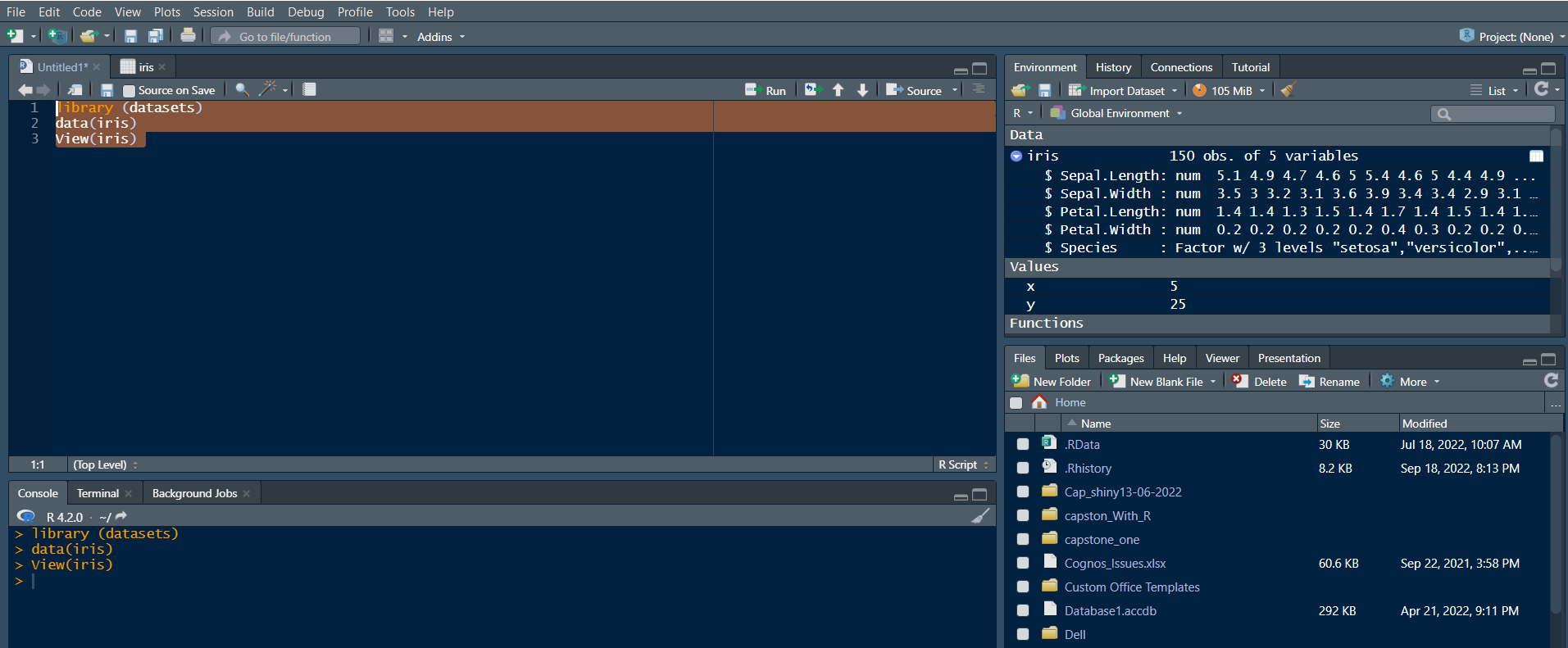


1. An **untitled** R Script panel opens. It would look as follows.

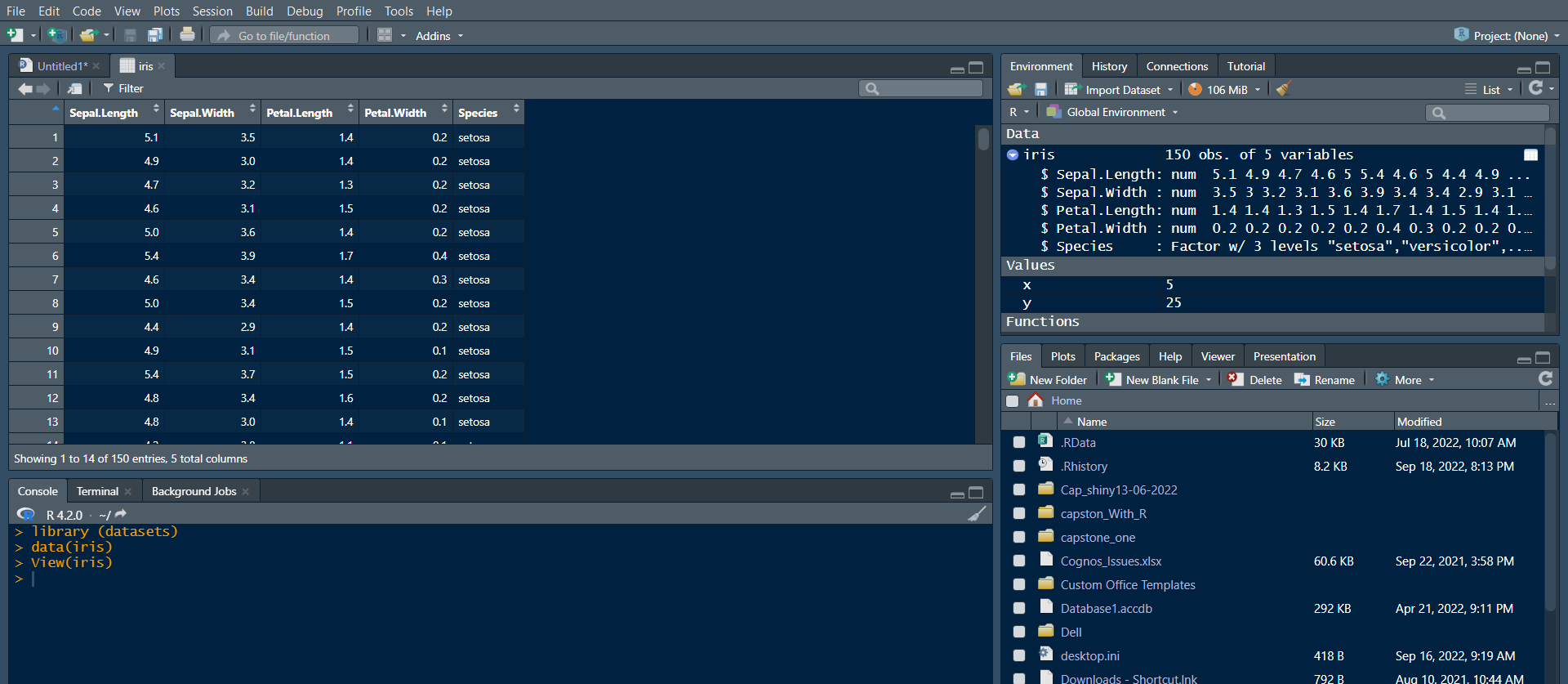


1. Now, load the **iris dataset**. Enter the following **lines** into the **Editor window** which appears. Next, select all of them. Then click the **Run icon** just above the Editor window.
2. 1
3. 2
4. 3
5. library (datasets)
6. data(iris)
7. View(iris)

Copied!

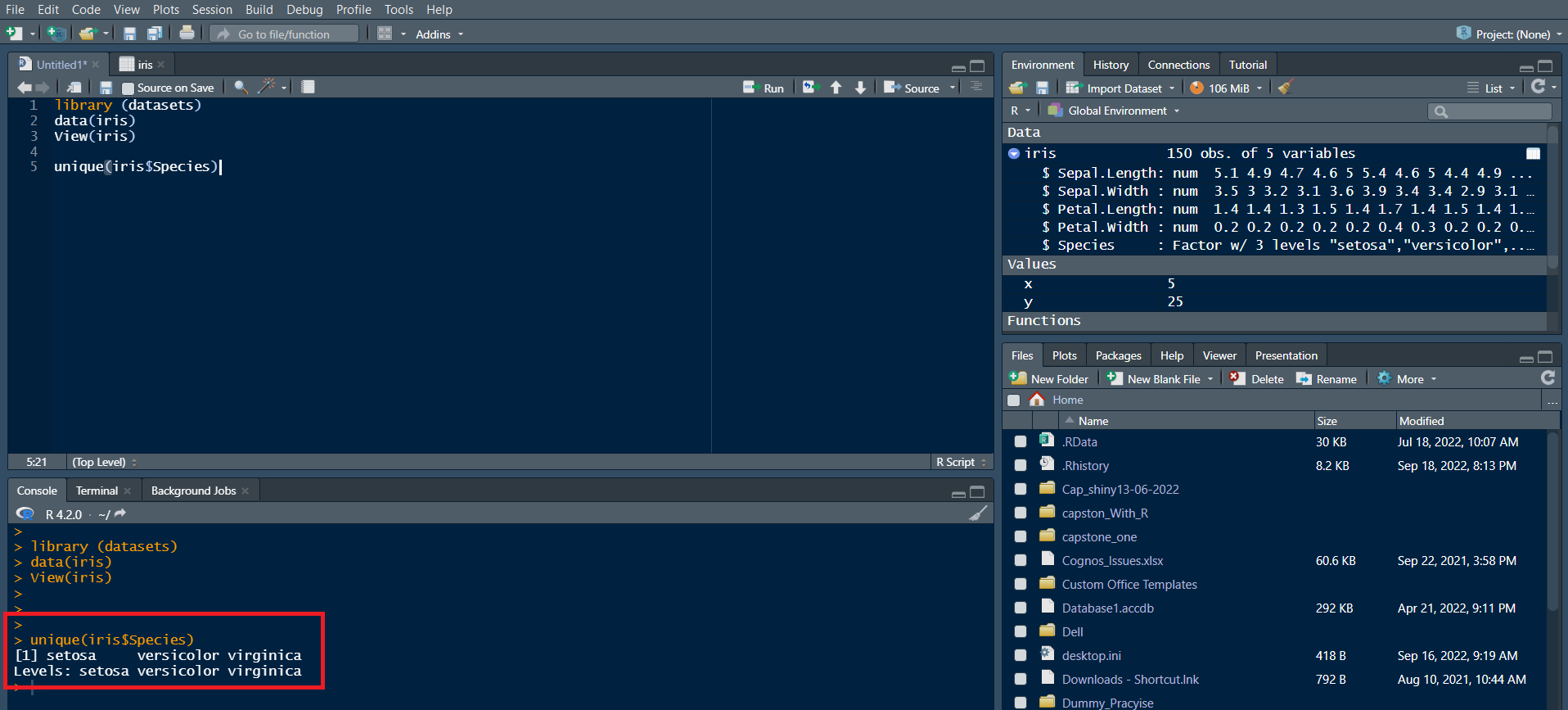
Output

1. You are taken directly to the data view tab to inspect your dataset. You can see five columns in this data set, the first four are floating point, and the last one is the label of the data type string, which contains the category value of your data set. You can see that there are total of 150 entries.



1. Now let’s find how many **different species** are present in the data set. Type the following command in the **Editor window** and **click Run**.
2. 1
3. unique(iris$Species)

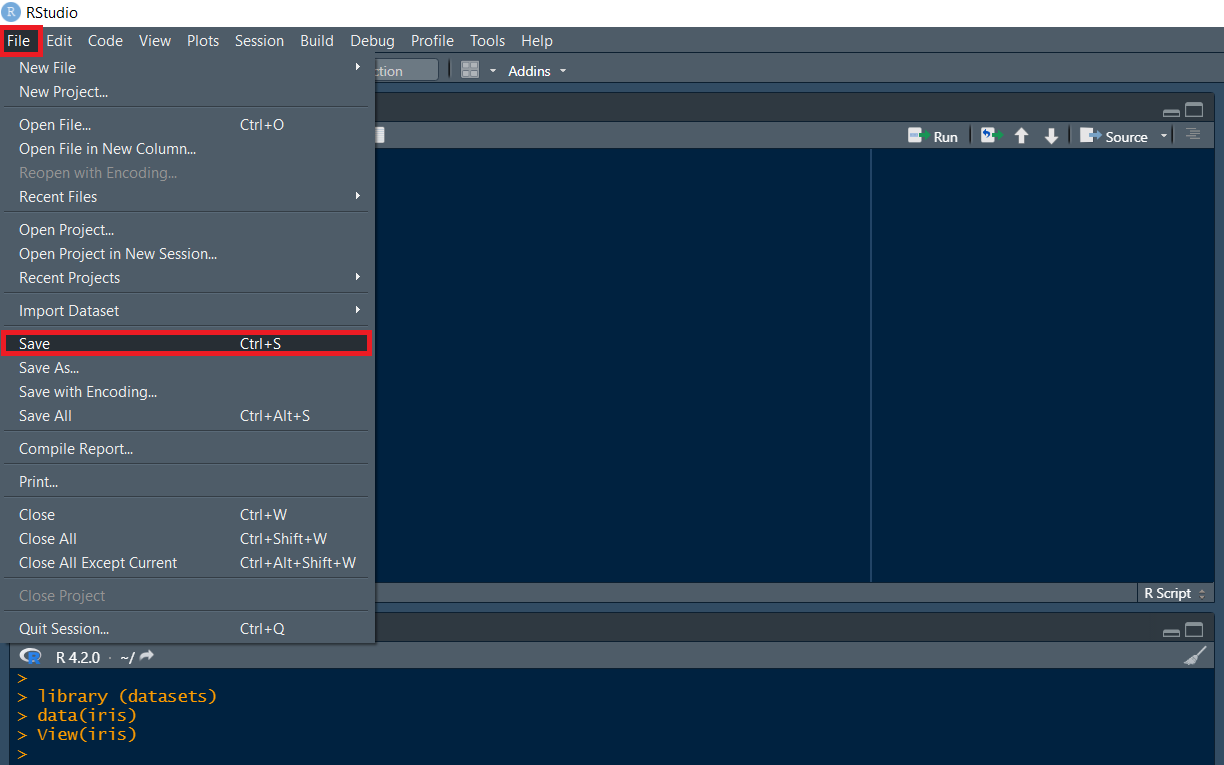
Copied!

Output

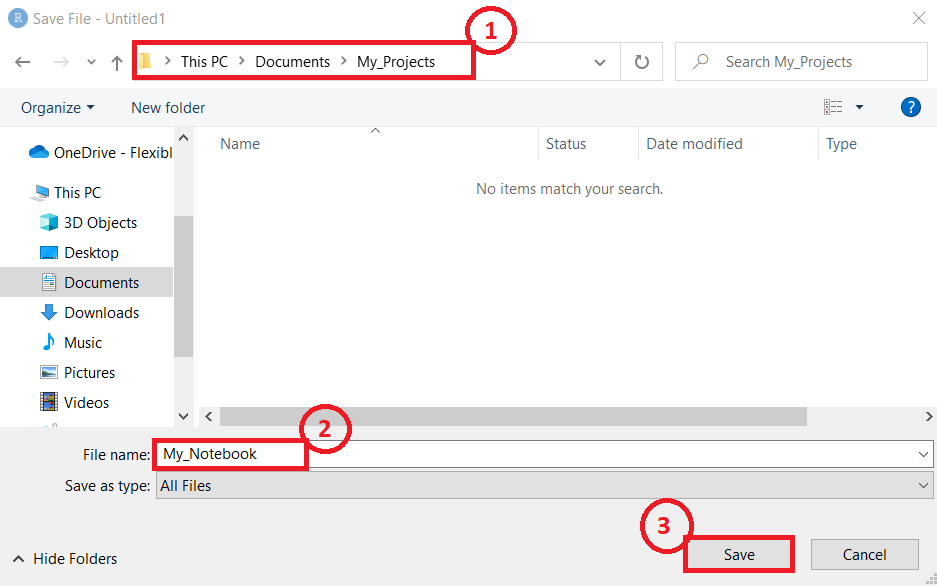
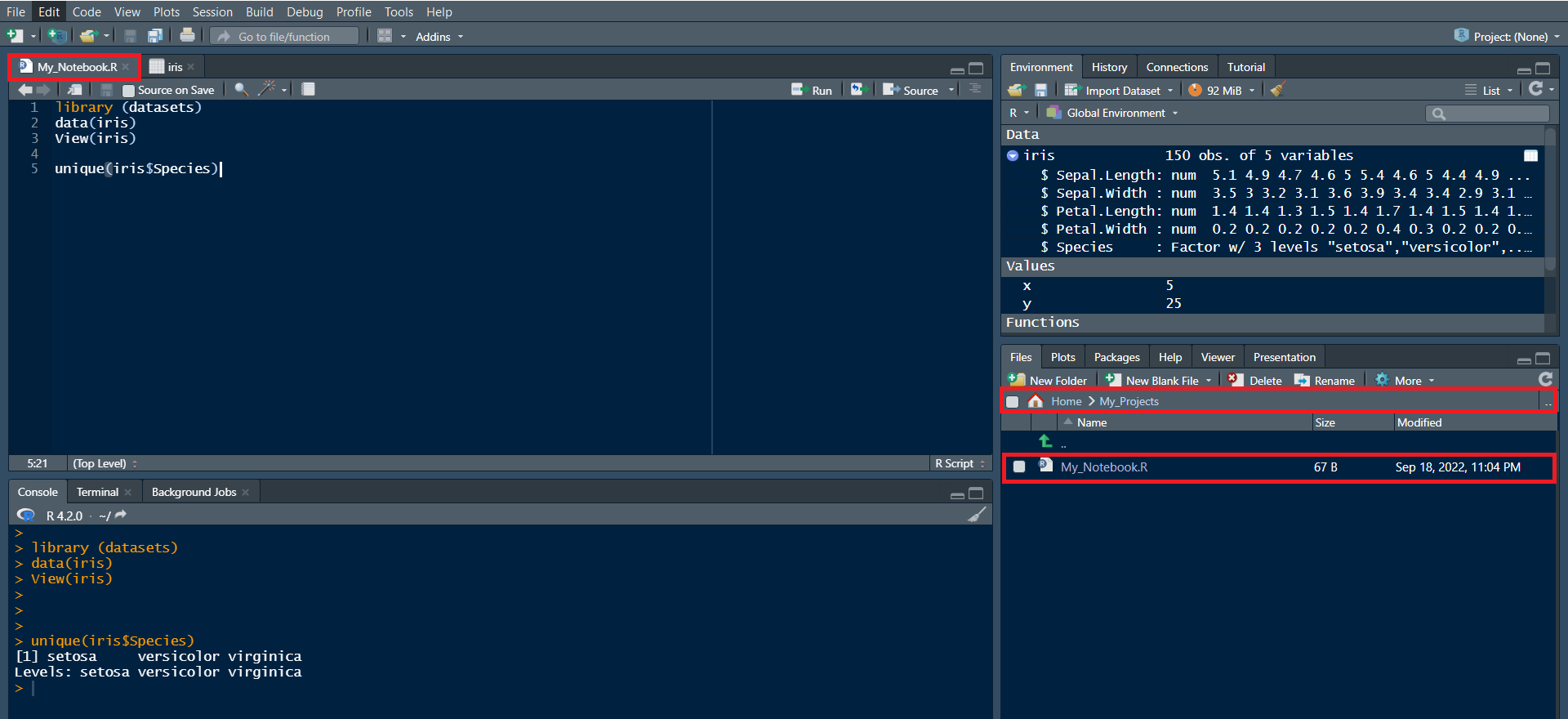
***Note****: In the Console window at the bottom, you will see the result of the executed command and know that only three different species are present in the data set.*

1. **Save & provide a name to your Notebook**.

* To save the notebook, click **Save** or **Save as** in the **File** menu.



* Select the working folder to rename your notebook to ***My\_Notebook***.

**Congratulations! In this document, you have learned how to download and install R and RStudio on your local machine. You also created a R notebook and saved it.**

## Objectives

After completing this lab, you will be able to:

1. Describe GitHub
2. Create a GitHub account
3. Add a project and repo
4. Edit and create a file
5. Upload a file and Commit

## GitHub Overview

First, let us introduce you to GitHub. GitHub is a collection of folders and files. It is a Git repository hosting service, but it adds many of its own features. Git is a command-line tool. It hosts and maintains a server via command line. GitHub provides this Git server and a Web-based graphical interface for you. It also provides access control and collaboration features, such as wikis and basic task management tools for every project. In addition, GitHub provides cloud storage for source code, supports all popular programming languages, and streamlines the iteration process. GitHub includes a free plan for individual developers and hosting open source projects.

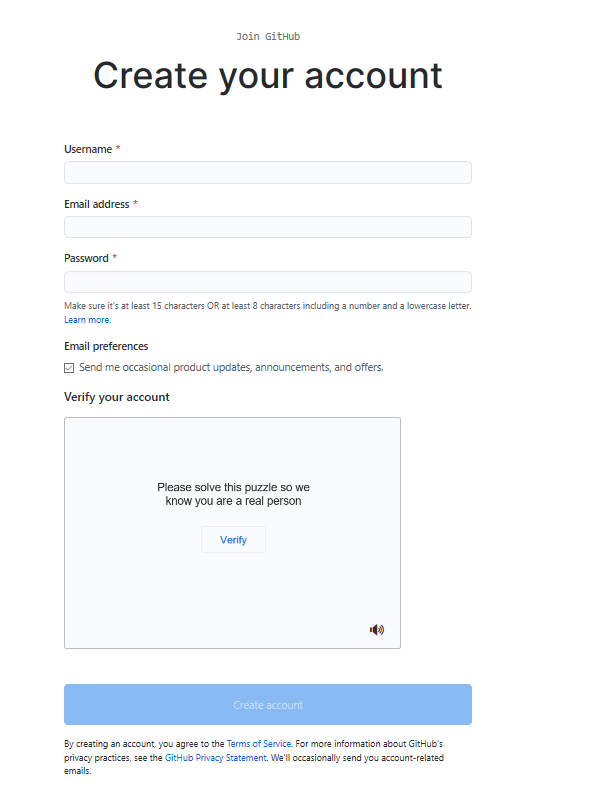
## Exercise 1: Creating a GitHub Account

Please use the following steps to create an account on GitHub:

Step 1: Create an account: <https://github.com/join>

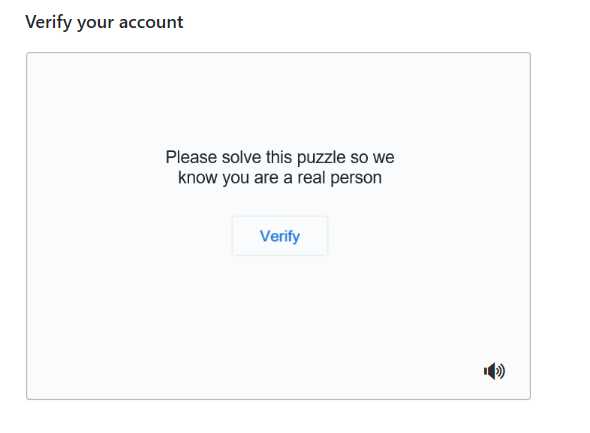
**NOTE:** If you already have a GitHub account, you can skip this step and simply log in to your account.

Step 2: Provide the necessary details to create an account as shown below:

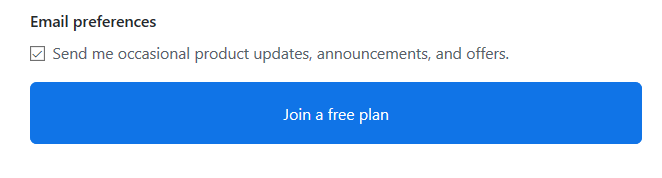


Click Create account.

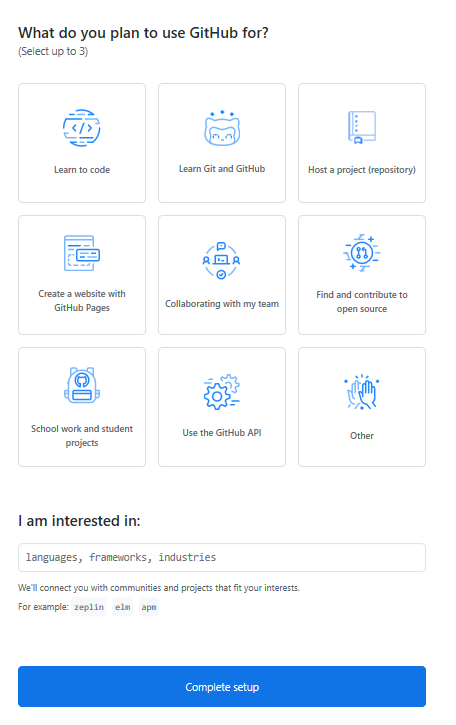
Step 3: Click Verify to verify the account and click Done.



Step 4: After verification, click Join a Free Plan.

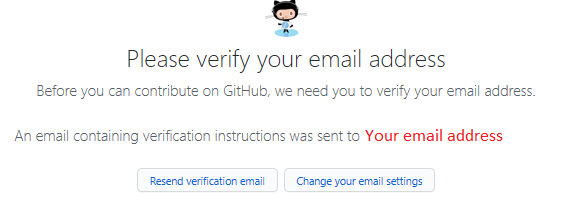


Step 5: Select the details as shown below and click Complete setup.

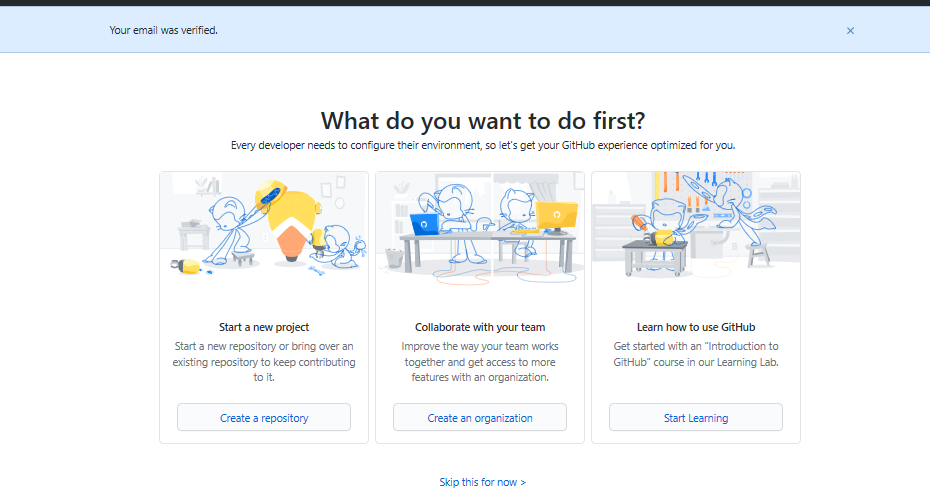


Step 6: Go to your email, find the verification email from GitHub, and click the Verify your email button or link in that email to verify.

**NOTE:** If you do not receive the verification email, click Resend verification email.

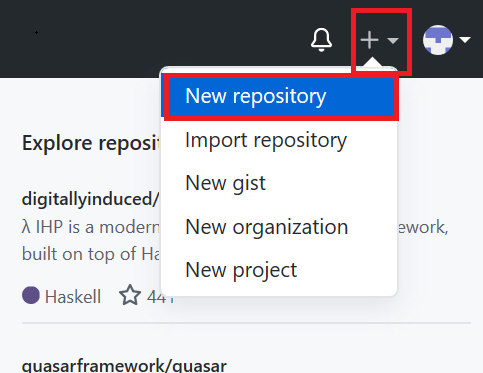


Email is verified.

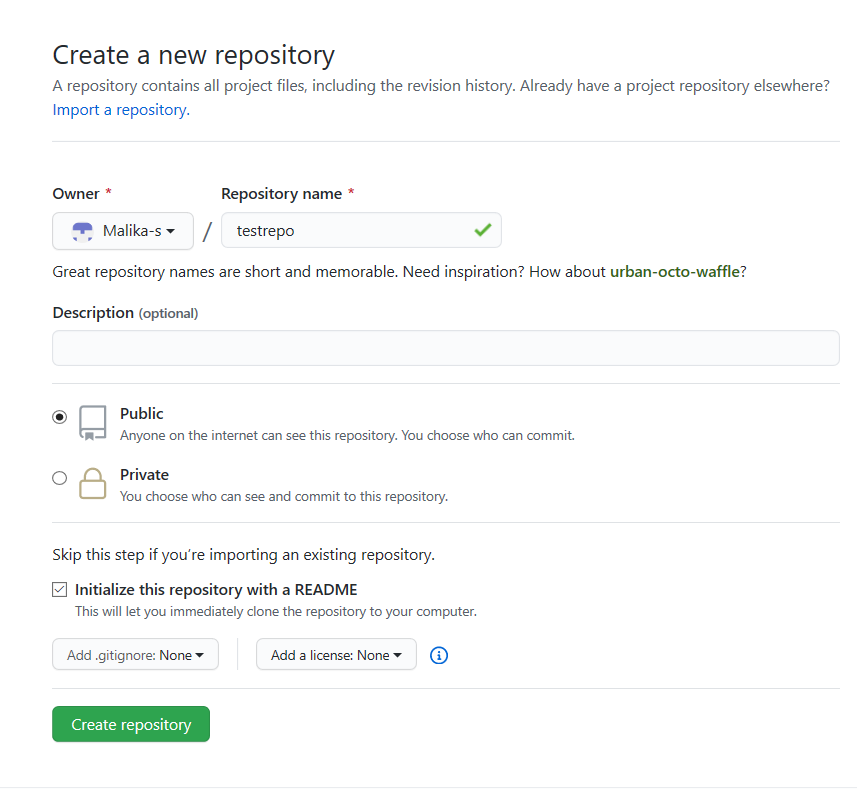


## Exercise 2: Adding a project and repo

Step 1: Click the + symbol and click New repository.



Step 2: Provide a name for the repository and initialize it with the empty README.md file.



Click Create repository.

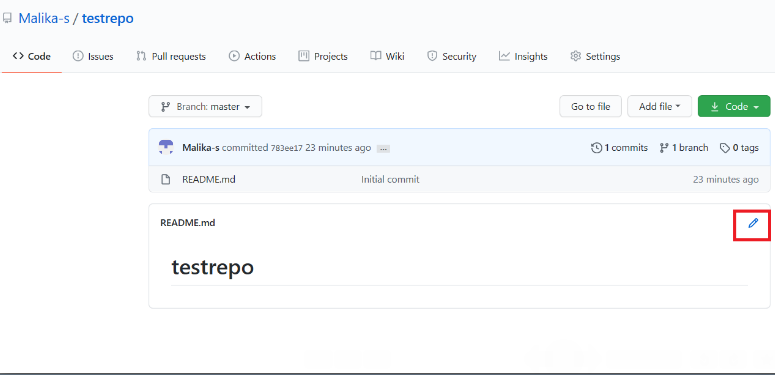
Now, you will be redirected to the repository you have created.

Let’s start editing the repository.

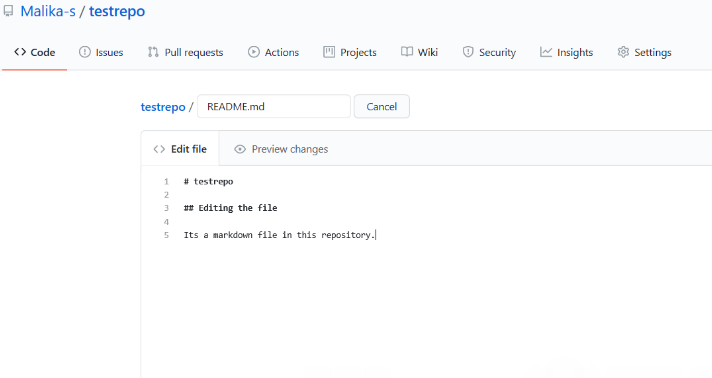
## Exercise 3: Create and edit a file

### Exercise 3a: Edit a file

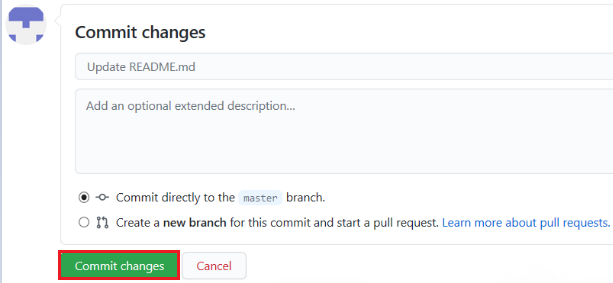
Step 1: Once the repository is created, the root folder of your repository is listed by default, and has just one file, ReadMe.md. Click the pencil icon to edit the file.



Step 2: Add some text to the file.



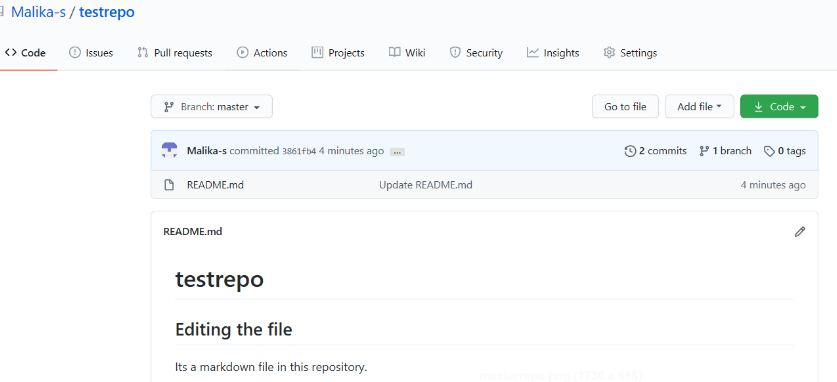
Step 3: Scroll down the page after adding the text and click Commit Changes.



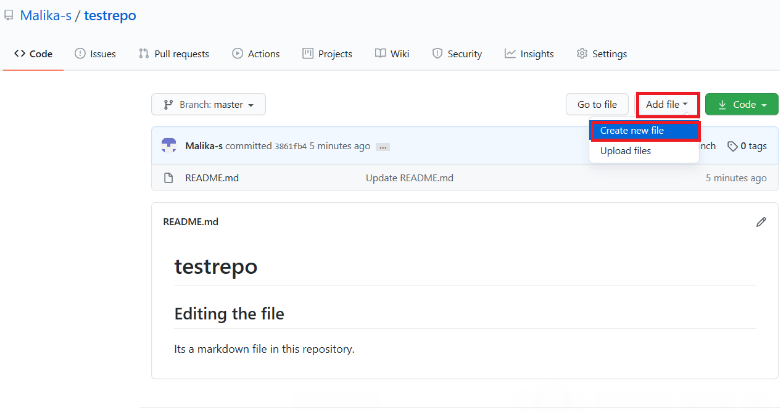
Now, check that your file is edited with the new text.

### Exercise 3b: Create a new file

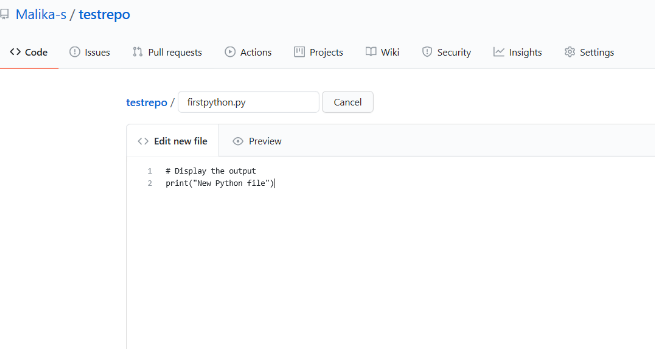
Step 1: Click the repository name to return to the master branch, like in this testrepo.



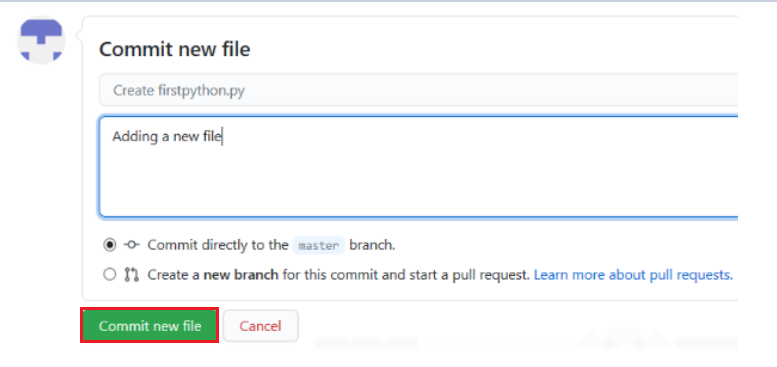
Step 2: Click Add file and select Create New file to create a file in the repository.



Step 3: Provide the file name and the extension of the file. For example, firstpython.py and add the lines.



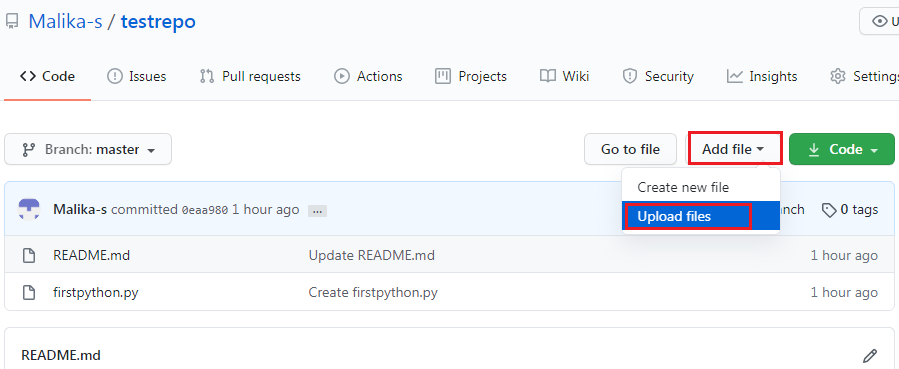
Step 4: Scroll down the page after adding the text. Add description of the file (optional) and click Commit new file.



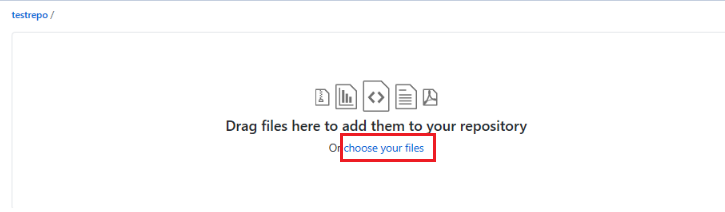
Step 5: Your file is now added to your repository, and the repository listing shows when the file was added and changed.

## Exercise 4: Upload a file & Commit

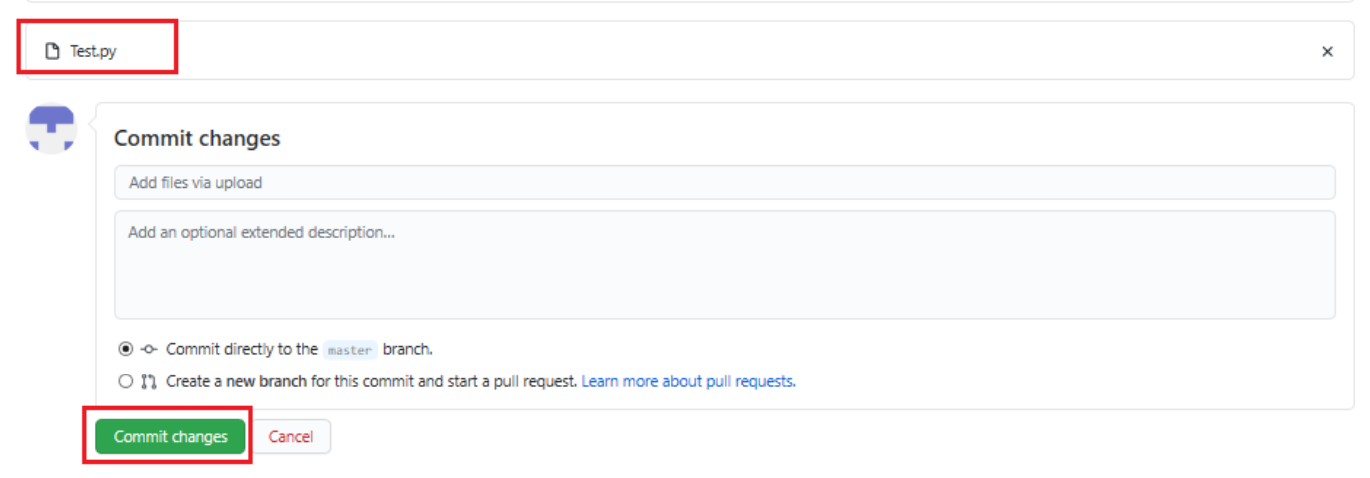
Step 1: Click Add file and select Upload files to upload a file (any .txt, .ipynb, .png file) in the repository from the local computer.



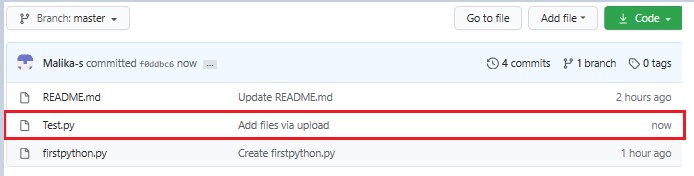
Step 2: Click choose your files and select any files from your computer.



Step 3: Once the file finishes uploading, click Commit changes.



Step 4: Now, your file is uploaded in the repository.



## Summary

In this document, you have learned how to create a new repository, add a new file, edit a file, upload a file in a repository, and commit the changes.

## Objectives

After completing this lab you will be able to:

1. Create a branch
2. Commit changes to a child branch
3. Open a pull request
4. Merge a pull request into the main branch

## Prerequisites

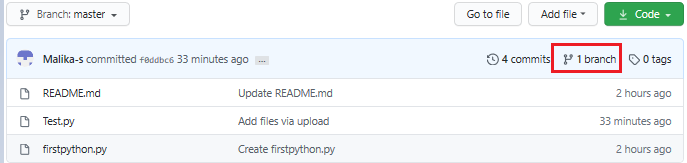
This hands-on lab requires you to have created a GitHub account and added a project to it, as covered in the [Getting started with GitHub](https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDeveloperSkillsNetwork-CD0101EN-SkillsNetwork/labs/GitHubLabs/GitHub1_Getting_Started.md.html) lab.

*NOTE: In the past the default branch in your GitHub repo used the name master. Effective Oct 1. 2020, all new GitHub repositories use the more inclusive term main as the name of the default branch instead of master.*

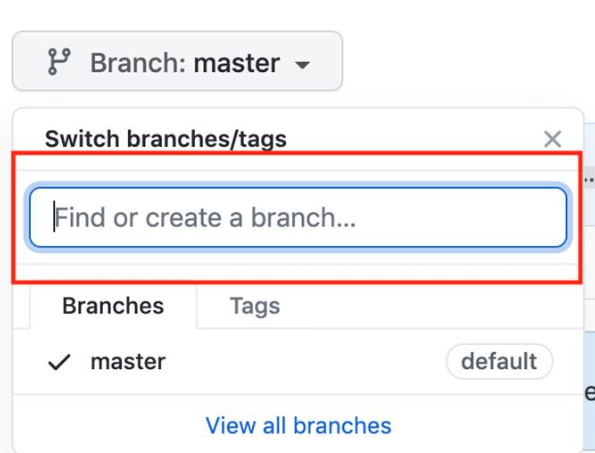
## 1. Create a branch

You can create or delete branches using your repository’s GitHub web page. To add a branch to your repository, complete the following steps:

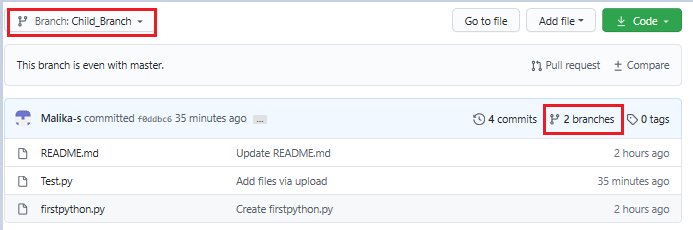
1. Go to you repository’s main page. Note that when you created your repository, the **main** branch was created for you:



1. At the top of the file list, locate the **Branch** drop-down menu. (By default, the menu displays **Branch: main**.) Click the drop-down menu, type the name of the branch you want to create, and press Enter on your keyboard.



Your repository now has two branches: **Main** and **Child\_Branch**. You can click the drop-down menu to see your branches.

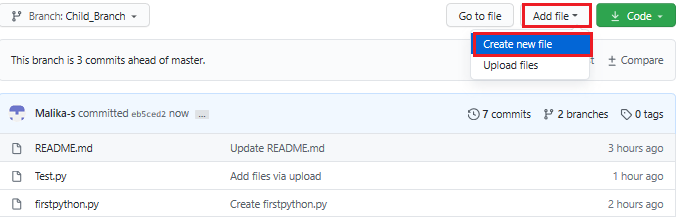


Any files that were in the **main** branch have now been copied to **Child\_Branch**. Note that when you add or edit a file in **Child\_Branch**, that change will not automatically be made in the **main** branch.

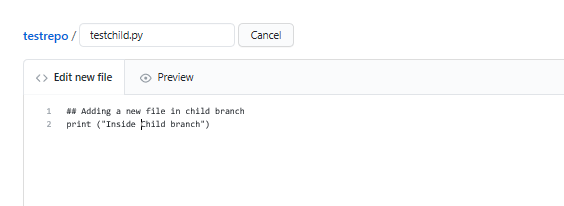
## 2. Add a file to a branch

To add a file to your new branch, ensure that **Child\_Branch** (or whatever name you gave your branch) is displayed in the **Branch** drop-down menu and complete the following steps:

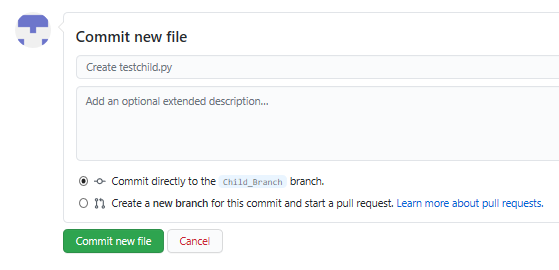
1. Click **Add file > Create new file** to create a file in the repository.



1. Type a name and extension for the file – for example, testchild.py – and add the following lines to the body of the new file:



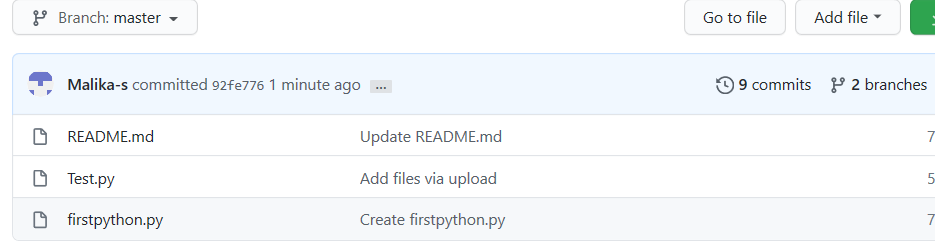
1. Scroll to the bottom of the page, add a description of the file you are about to add (note that the description is optional), and click **Commit**.



The file is added to your child branch.

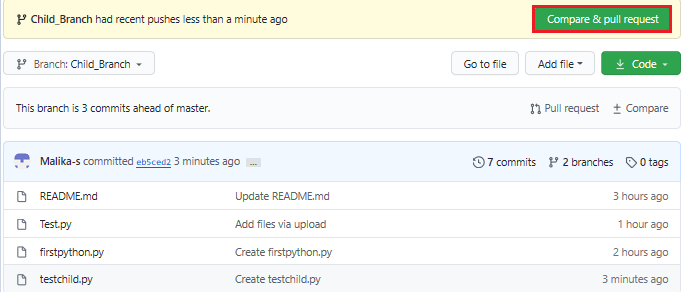
## 3. Open a pull request

The file that you added to your child branch is not automatically added to the **main** branch. (You can check this by using the **Branch** drop-down menu to go to the **main** branch; note that there is no testchild.py file in the file list):

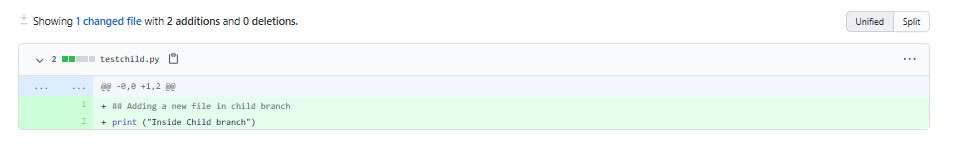


You can also compare the two branches and open a pull request, which will enable you to copy the changes that you’ve made in the child branch – in this case, adding a new file – to the **main** branch.

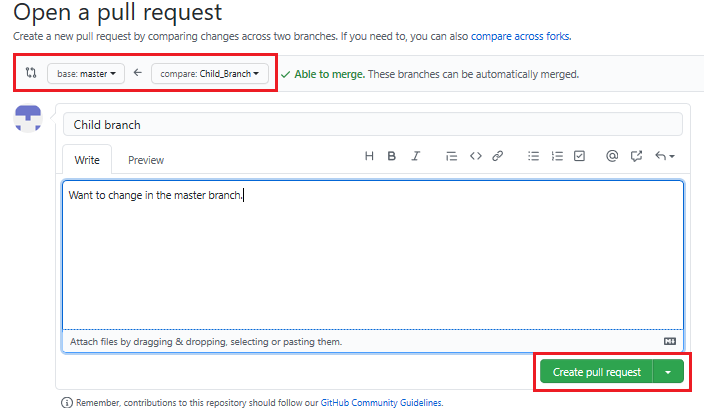
1. In **Child\_Branch**, click the **Compare & pull request** button.



1. Scroll to the bottom of the page and note that there is **1 changed file** listed.



1. Scroll up and note that GitHub is comparing the **main** and **Child\_Branch** branches and that there are no conflicts between the two. Optionally, you can add a comment to the pull request. Click **Create pull request**.

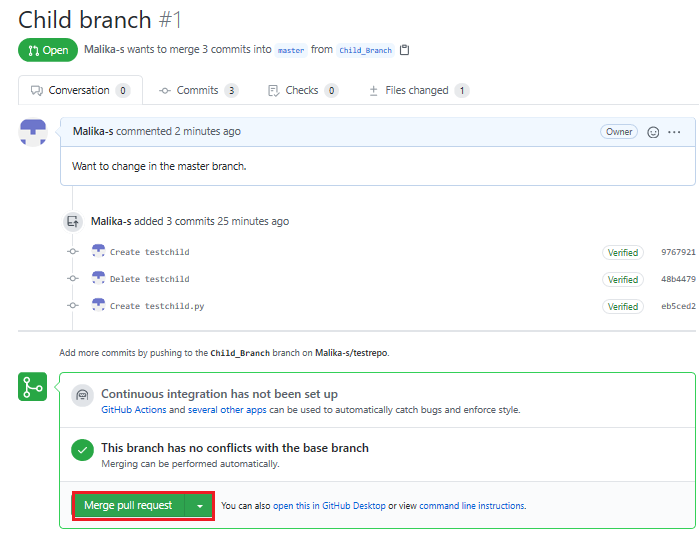


The pull request is created and can now be merged by a repository administrator.

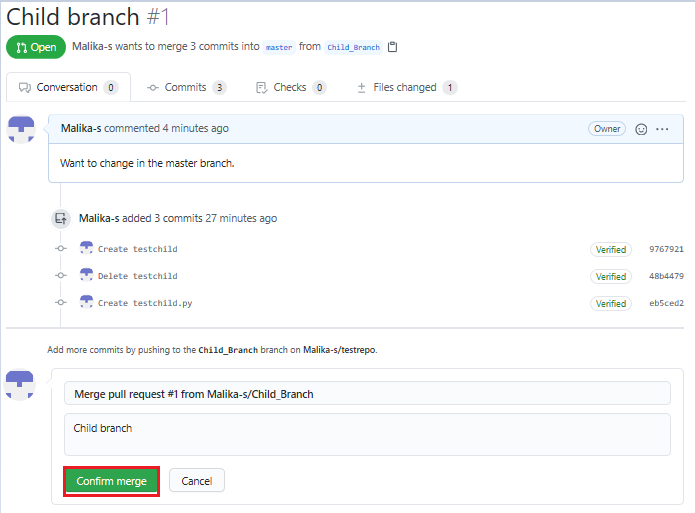
## 4. Merge a pull request

To merge a pull request into a project, complete the following steps:

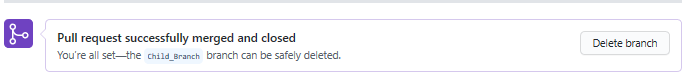
1. Click the **Pull requests** tab. A list of pending pull requests is displayed.
2. Click the pull request that you want to merge into the main project. If you are satisfied with the changes, click **Merge pull request** to accept the pull request and merge the updates. (You can add a comment if you choose.)



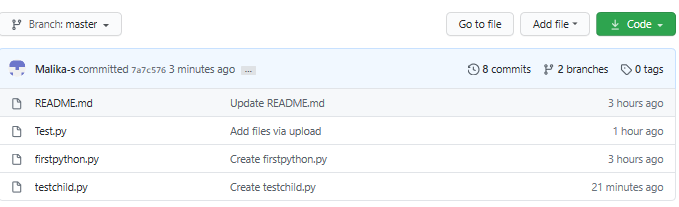
1. When you click **Merge pull request**, a **Confirm merge** button is displayed. Click that button to complete the merge.



The pull request has now been merged successfully. Note that you can delete the child branch because your changes have been incorporated into the **main** branch.



Check the list of files in the **main** branch to confirm that it now includes the file that you added in the pull request.



## Summary

Congratulations! You’ve now learned how to create a branch, edit and commit changes in that branch, open a pull request, and merge the pull request into your main project. We encourage you to continue to experiment with branches and pull requests to become more familiar with the concepts and processes.

# Module 5 Summary

Congratulations! You have completed this module. At this point in the course, you know:

* The capabilities of R and its uses in Data Science.
* The RStudio interface for running R codes.
* Popular R packages for Data Science.
* Popular data visualization packages in R.
* Plotting with the inbuilt R plot function.
* Plotting with ggplot.
* Adding titles and changing the axis names using the ggtitle and lab’s function.
* A Distributed Version Control System (DVCS) keeps track of changes to code, regardless of where it is stored.
* Version control allows multiple users to work on the same codebase or repository, mirroring the codebase on their own computers if needed, while the distributed version control software helps manage synchronization amongst the various codebase mirrors.
* Repositories are storage structures that:
  + Store the code
  + Track issues and changes
  + Enable you to collaborate with others
* Git is one of the most popular distributed version control systems.
* GitHub, GitLab and Bitbucket are examples of hosted version control systems.
* Branches are used to isolate changes to code. When the changes are complete, they can be merged back into the main branch.
* Repositories can be cloned to make it possible to work locally, then sync changes back to the original.