Walkthrough of Installing Everything Needed for AI-Assisted-Coding-In-Python Workshop

2025-11-04

Table of contents

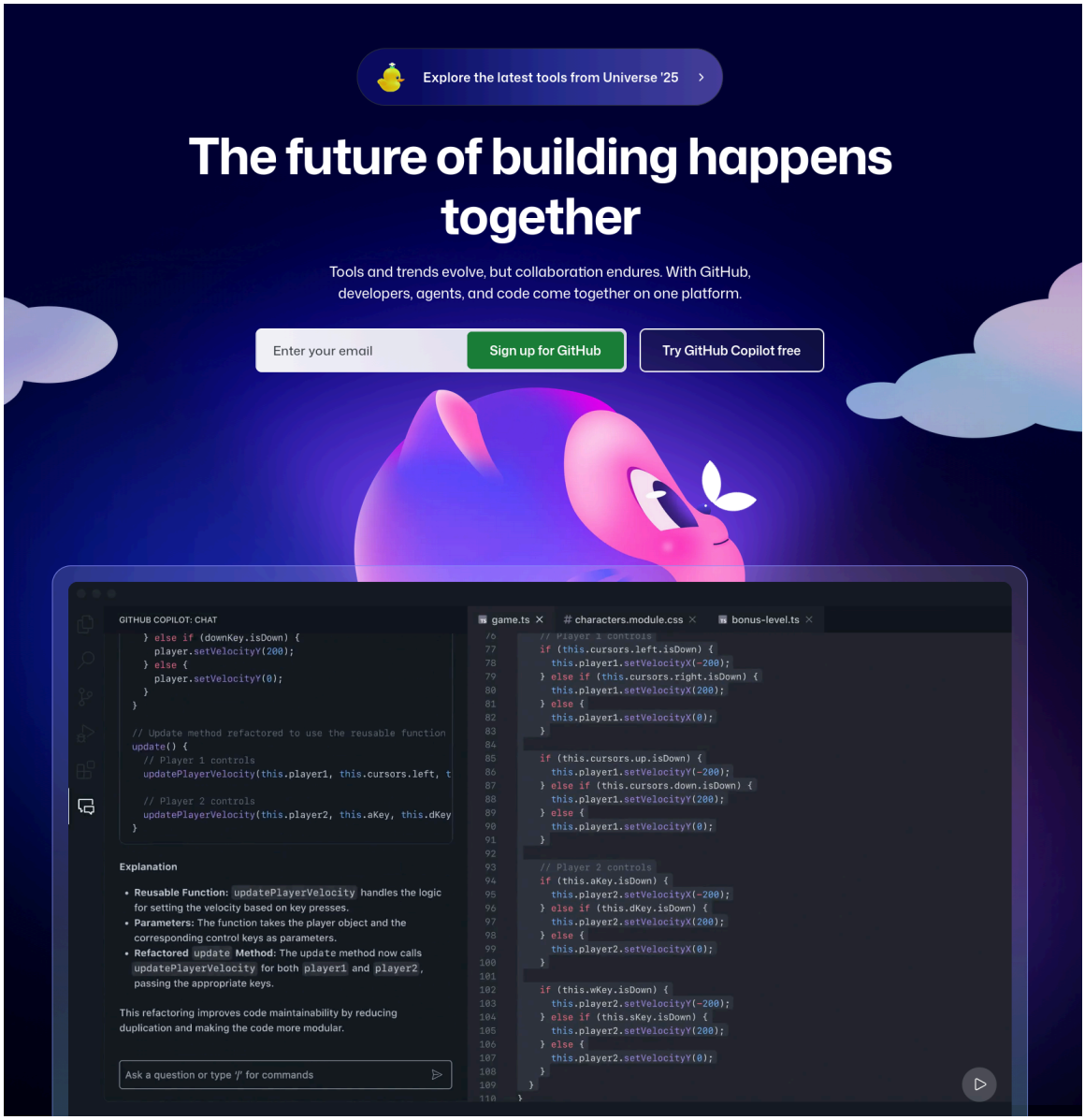
1 Checklist of Everything Needed to Install for AI-Assisted-Coding-In-Python

- [] Install GitHub
- [] Sign up for GitHub Copilot
- [] Install Python
- [] Install Visual Studio (VS) Code
- [] Adjust VSCode to work with Python
- [] Download Zip file

When these are completed, you are ready for your AI-Assisted-Coding-In-Python workshop.

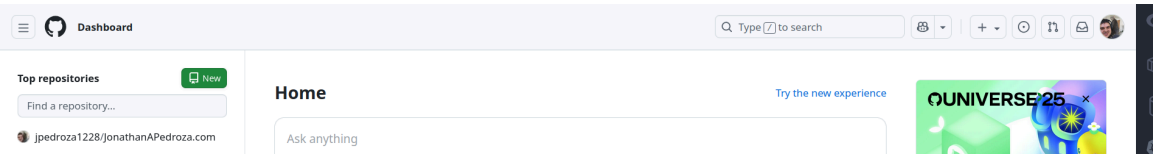
2 Installing GitHub

Go to GitHub and sign up for a GitHub account.



At this page, you will sign up for GitHub with your `@berkeley.edu` email account. Follow the directions to verify your account. Below are some recommendations for creating a username (inspired by https://happygitwithr.com/github-acct).

- Use part of your real name so it is easier for people to know who you are
- Try and keep it short, you may have to type it a lot
- Keep everything lowercase. If you really want to separate words, use a hyphen (-) or an underscore (_)

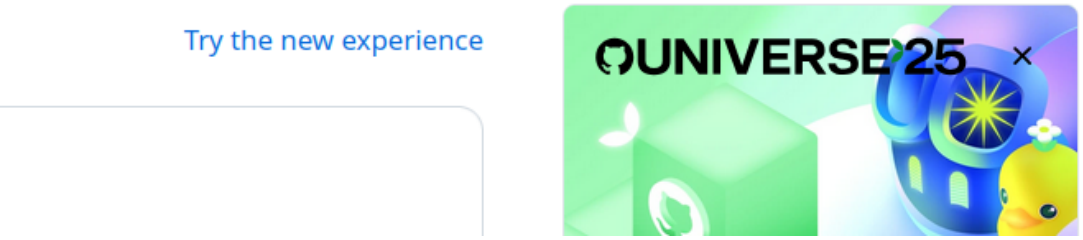


Once you sign in, you will be at your dashboard. You have now downloaded GitHub! Congrats!

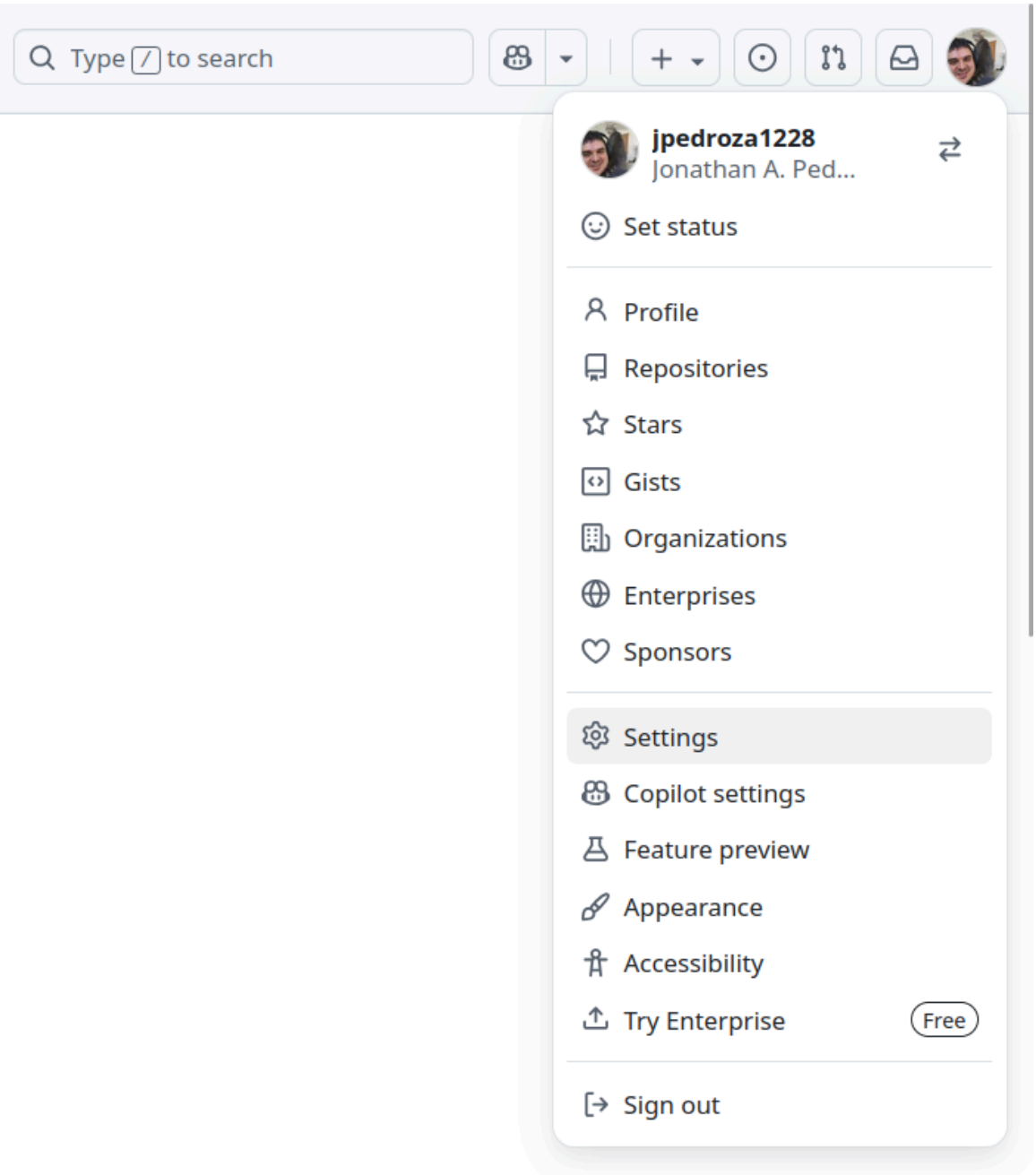
2.1 Optional: Two-Factor Authentication (2FA)

While not part of the tutorial, GitHub can house public and private data. If you are planning on continuing to use GitHub, please think about creating additional safeguards by setting up your Two-Factor Authentication (2FA). Below are some resources and a quick look into 2FA.

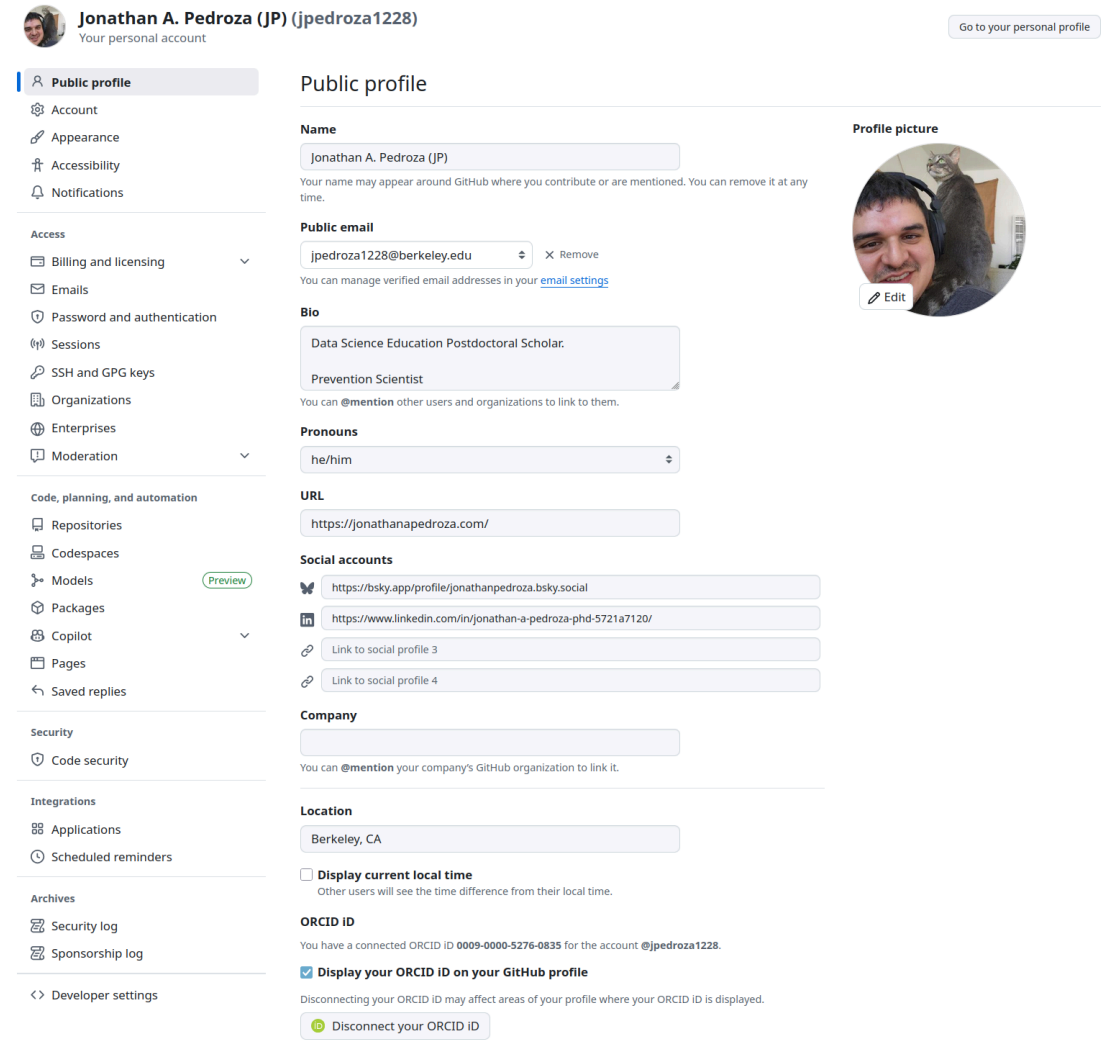
For more information on 2FA, you can find resources here (About 2FA) or here (Securing account with 2FA).



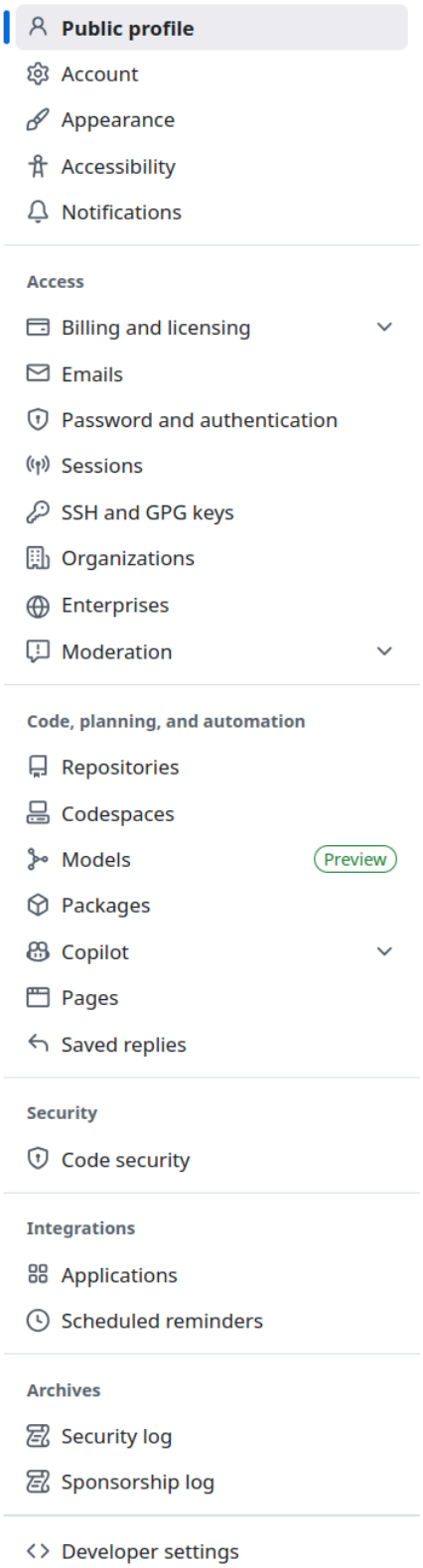
From your dashboard, you will want to to to your profile. If you just created your account, you will not have a profile picture. You'll then click on your profile circle to show a dropdown menu.



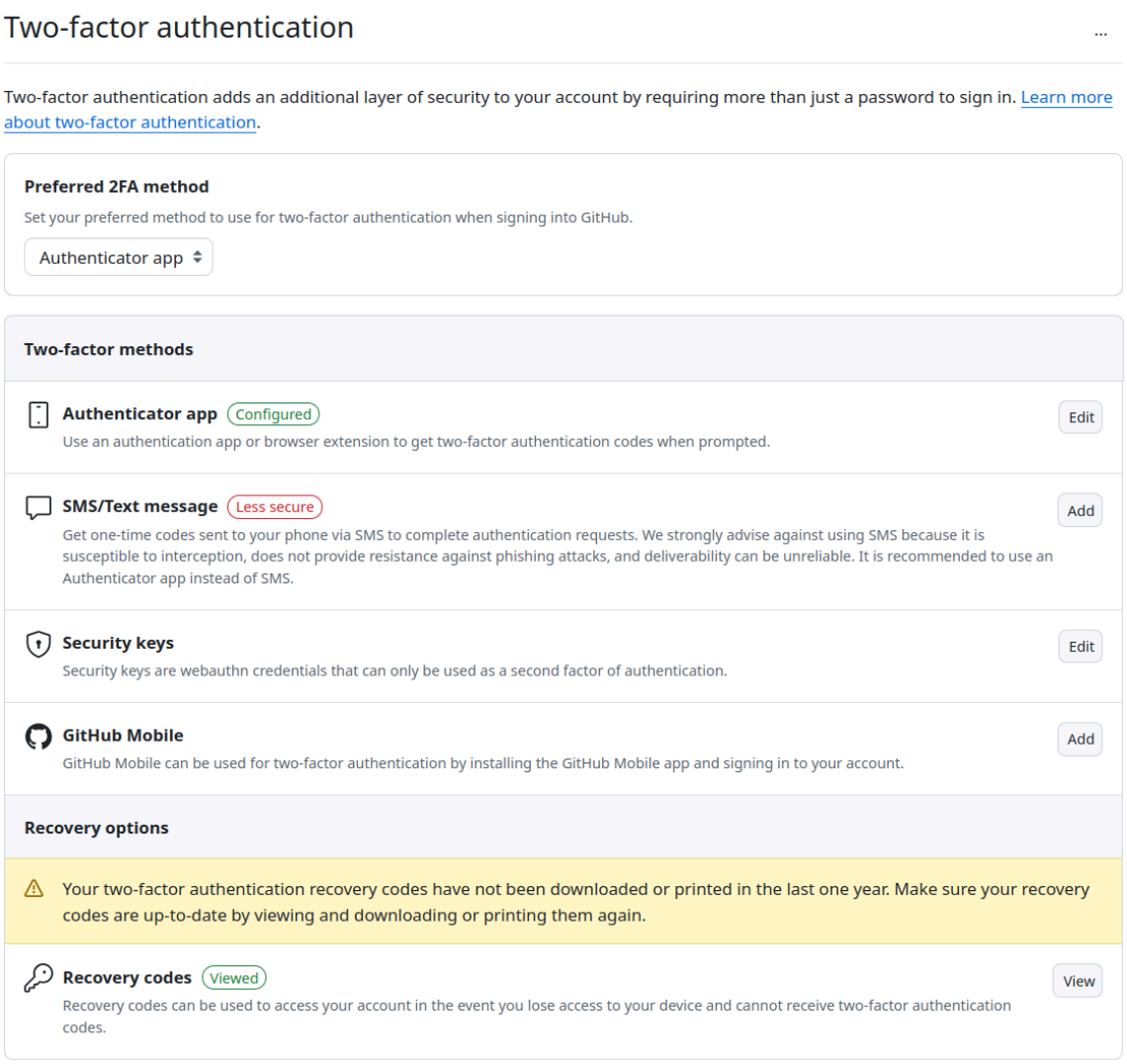
At this dropdown menu, you will go to your `Settings`.



At your settings, there will be a lot of information.


Public profile
Account
Appearance
Accessibility
Notifications
Access
Billing and licensing
Emails
Password and authentication
Sessions
SSH and GPG keys
Organizations
Enterprises
Moderation
Code, planning, and automation
Repositories
Codespaces
Models
Preview
Packages
Copilot
Pages
Saved replies
Security
Code security
Integrations
Applications
Scheduled reminders
Archives
Security log
Sponsorship log
Developer settings

On the left sidebar, you will want to go to `Password and authentication`. Here you can customize how you would like to sign into GitHub.



While there are options for 2FA, I would recommend using an authenticator app. So every time you sign in (among other actions on GitHub), you will sign in with your username and password then verify it with a code from your authenticator app.

- ☒ Install GitHub
- [] Sign up for GitHub Copilot
- [] Install Python
- [] Install Visual Studio (VS) Code
- [] Adjust VSCode to work with Python
- [] Download Zip file

3 Sign Up for GitHub Copilot (THIS SECTION NEEDS WORK)

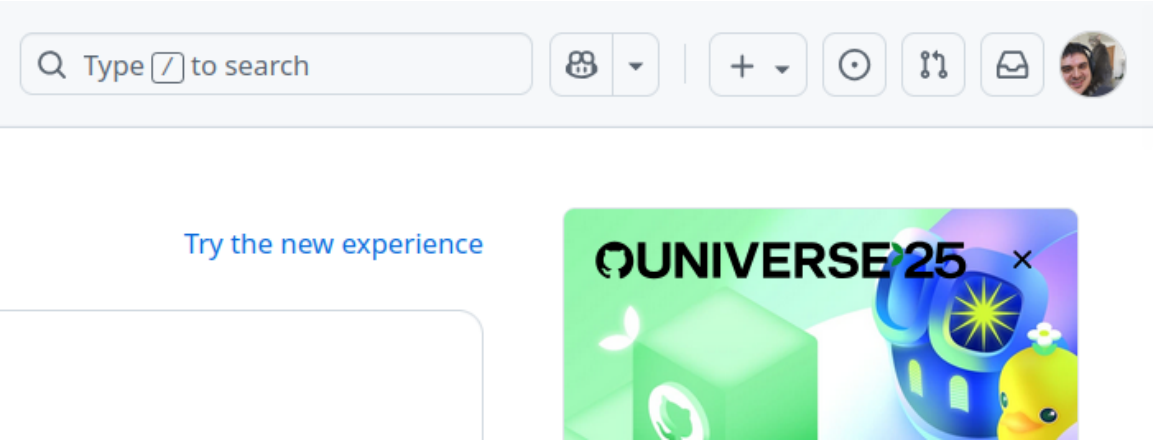
Signing up for GitHub Copilot will depend on whether you plan to sign up for the free version, CoPilot *Pro* using verified information, or pay for a Pro plan (see pricing information here).

3.1 Free Version

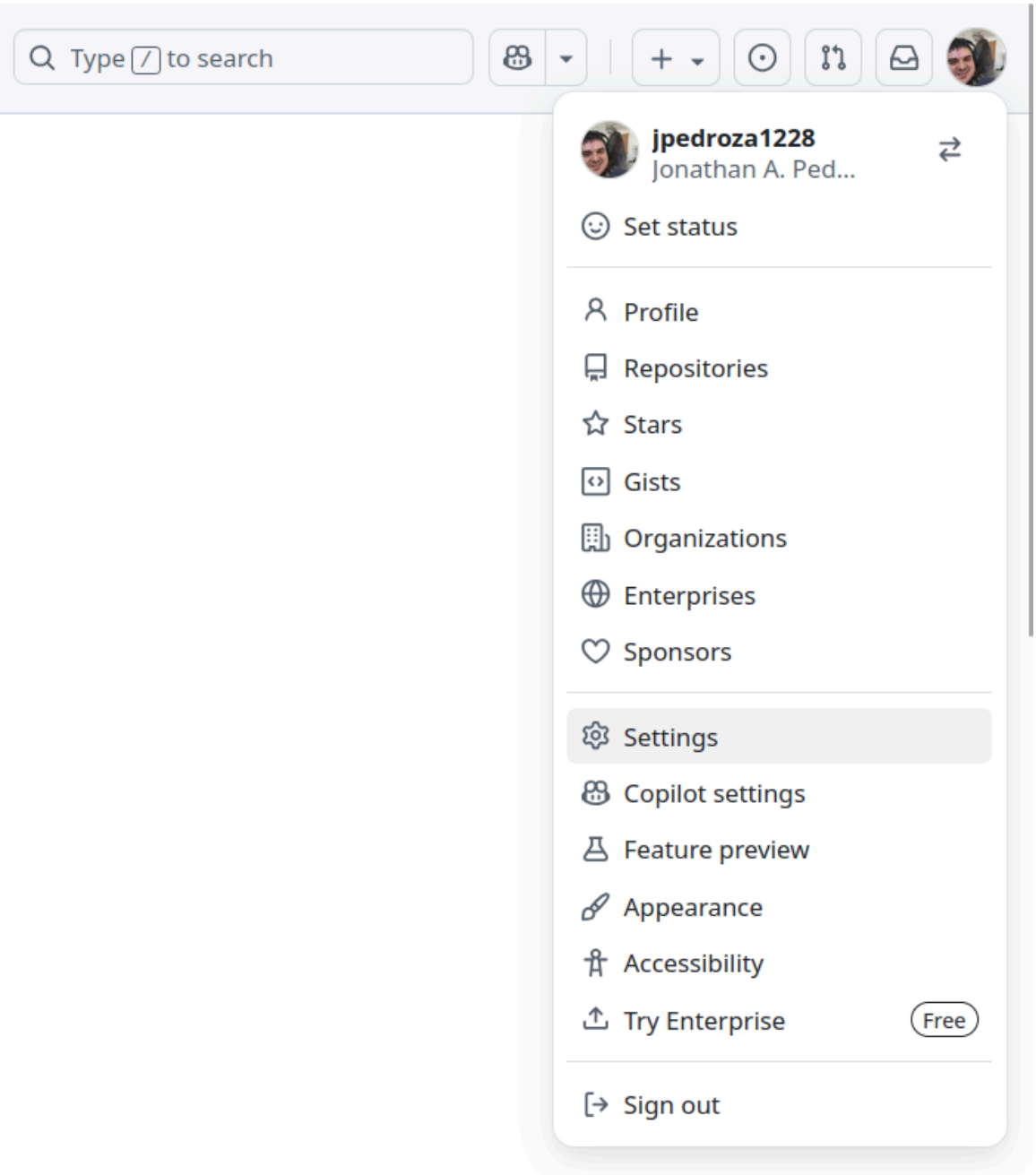
The free version of GitHub Copilot comes with VSCode. You can install the necessary extensions in the following section. You can also try out a 30-day trial here.

3.2 Copilot Pro (Instructions for Verification)

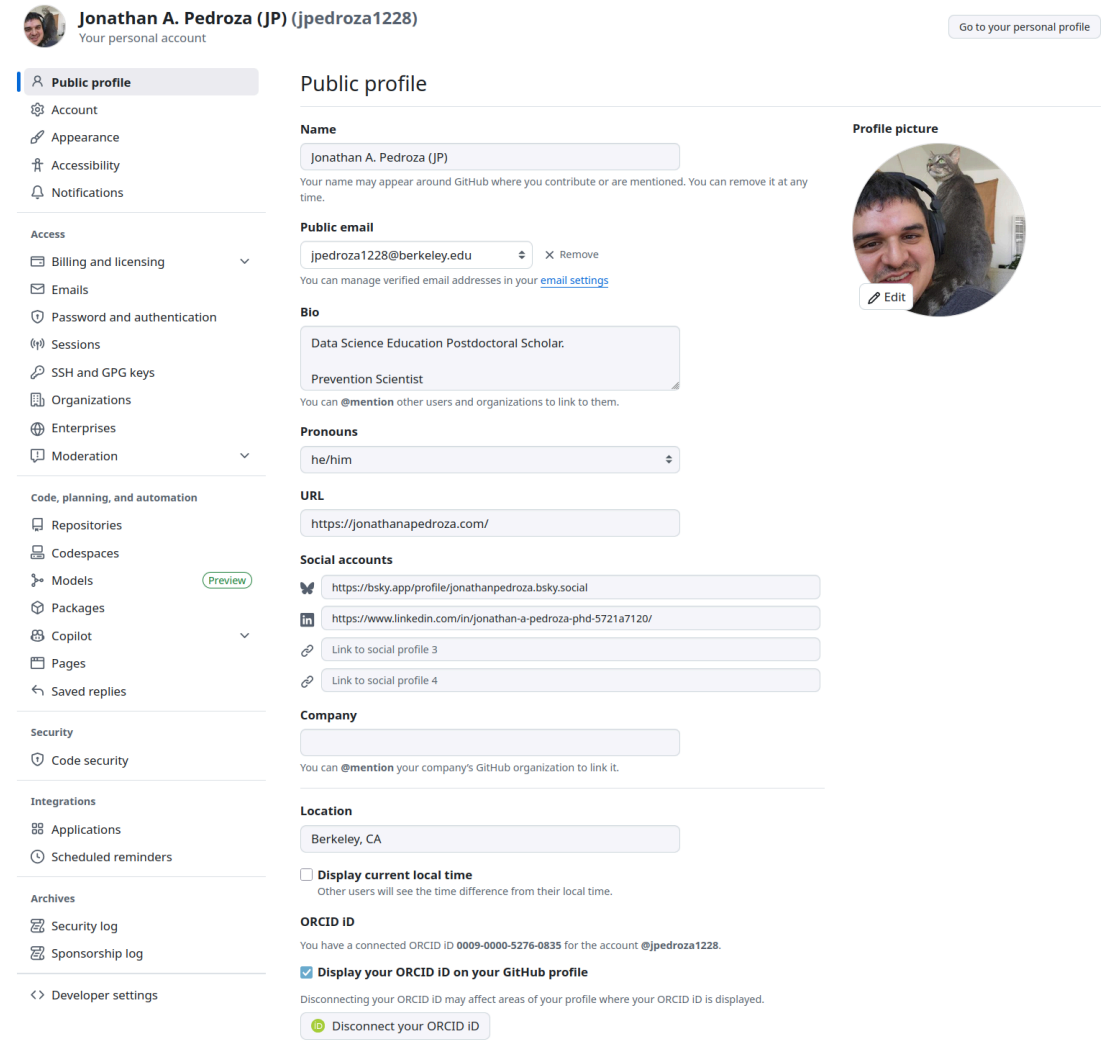
To get Copilot *Pro* for teachers and students (for free), you will need to follow the following steps. Below are some steps from the optional 2FA section above.



From your dashboard, you will want to to to your profile. If you just created your account, you will not have a profile picture. You'll then click on your profile circle to show a dropdown menu.



At this dropdown menu, you will go to your `Settings`.



At your settings, there will be a lot of information. This time, you will click on the dropdown menu for `Billing and licensing`.

Jonathan A. Pedroza (JP)

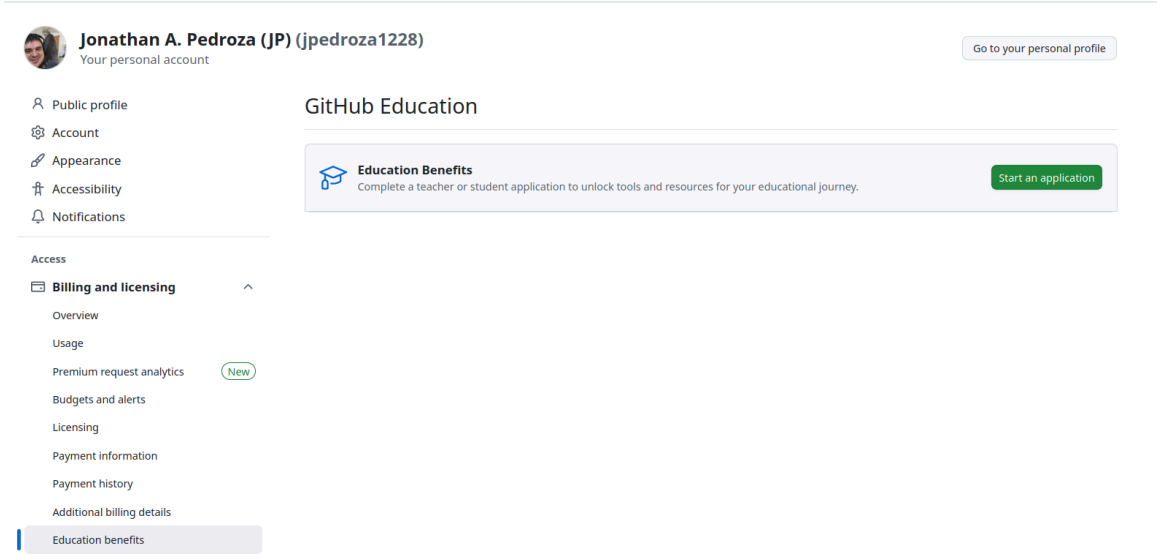
Your personal account

- **Public profile**
- Account
- Appearance
- Accessibility
- Notifications

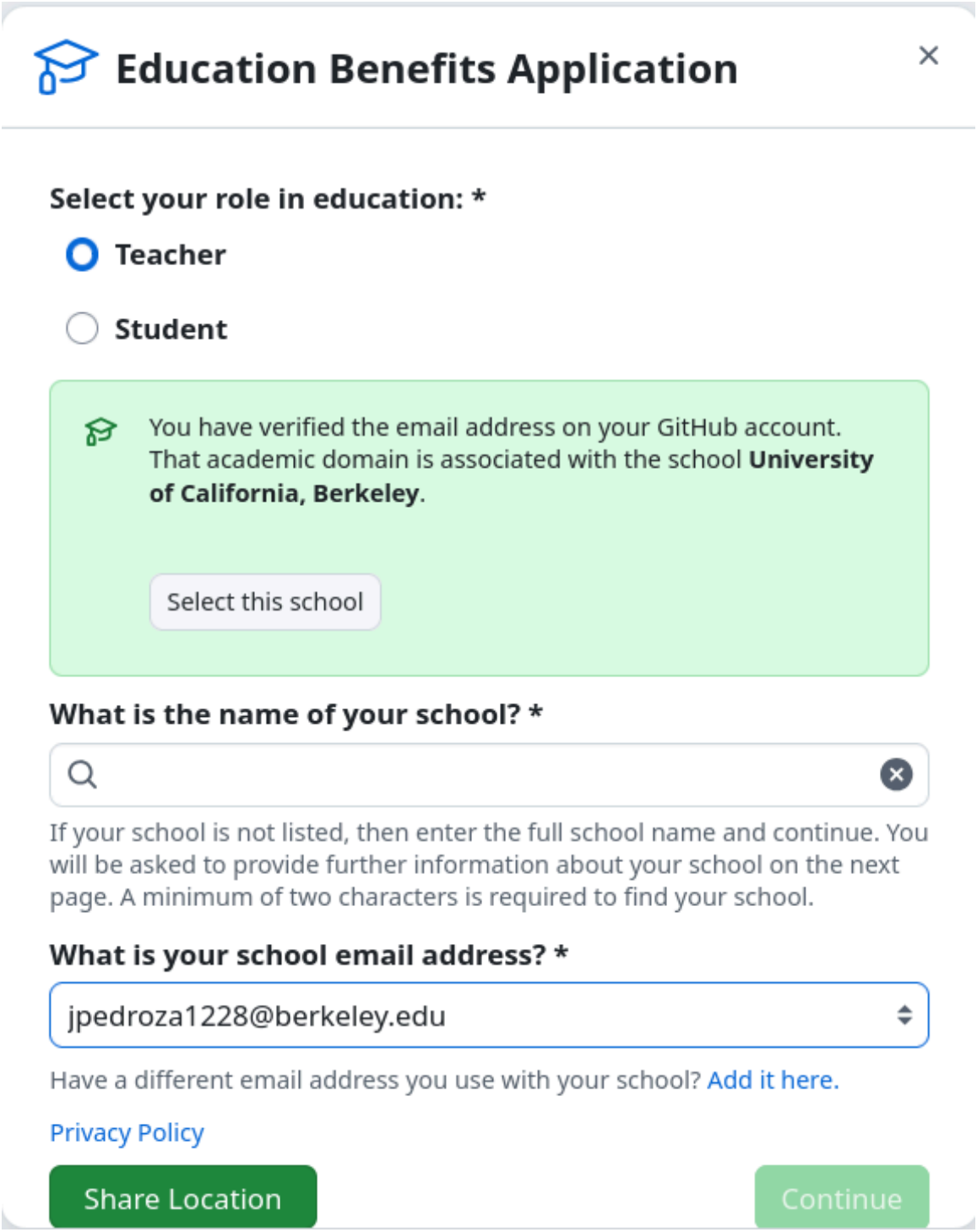
Access

- Billing and licensing
  - Overview
  - Usage
  - Premium request analytics — New
  - Budgets and alerts
  - Licensing
  - Payment information
  - Payment history
  - Additional billing details
  - Education benefits
- Emails
- Password and authentication
- Sessions

From the dropdown menu, you will click on `Education benefits`.

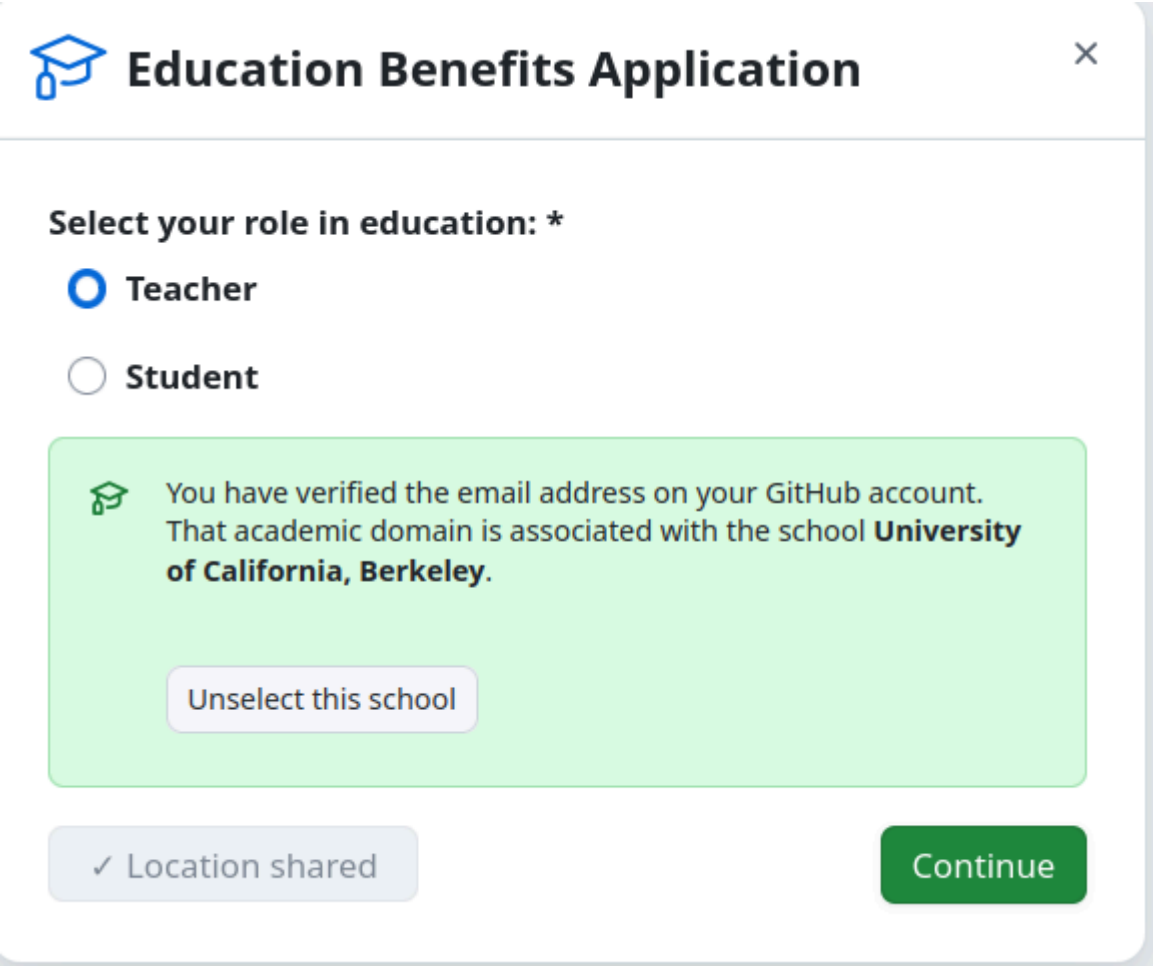


Clicking on `Education benefits` will take you to `GitHub Education`. There you will start your application to get additional benefits, including GitHub Copilot Pro.



Once you start your application, you will have the option of choosing your role At UC Berkeley. Below are general instructions for teachers and students; however, the instructions below will start to shift toward specific instructions for teachers.

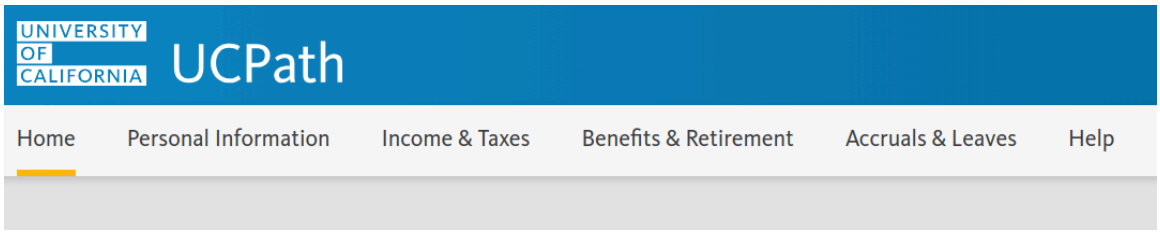
On the application, you can include the full name of the university (University of California, Berkeley) and when given an option to Select a school, click `Select this school`. You will also include your school email address in the dropdown menu. Make sure you are including your `@berkeley.edu` email address since verification relies on that email address.



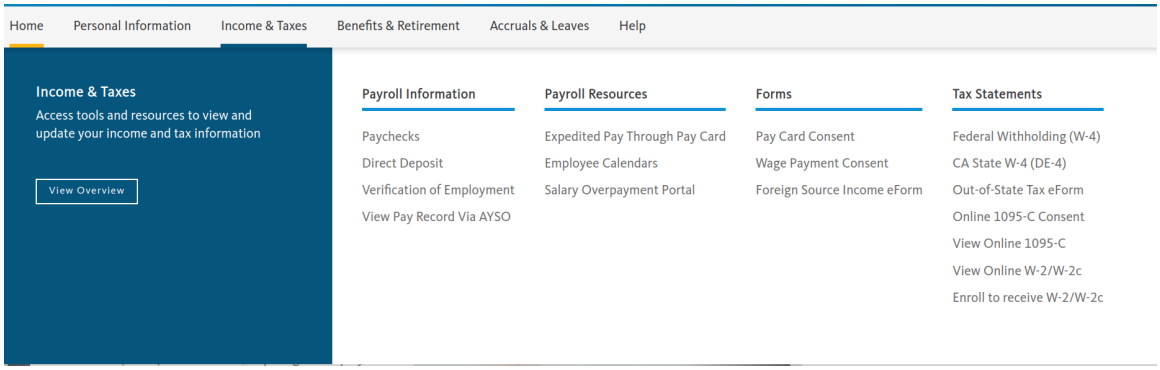
You will also be asked to share your location, which will then allow you to click on `Continue` and finish this section of the application. You will then be asked for additional information in the form of proof of you being associated with UC Berkeley. The following information will be for teacher roles.

You will need to provide proof of your affiliation. **This proof must have your name, a current date, and the name of your institution on it**. If your application is rejected, it is most likely because your proof of affiliation was missing one of these three things (but, you can always apply again with new proof!). For example, some UC Berkeley student IDs do not have a date on them, so they will not be accepted. You may also have to follow some additional steps to verify your GitHub account, make sure to check your email for instructions. **Note** We have also heard of a bug that results in .png files not being accepted while .jpeg files are.

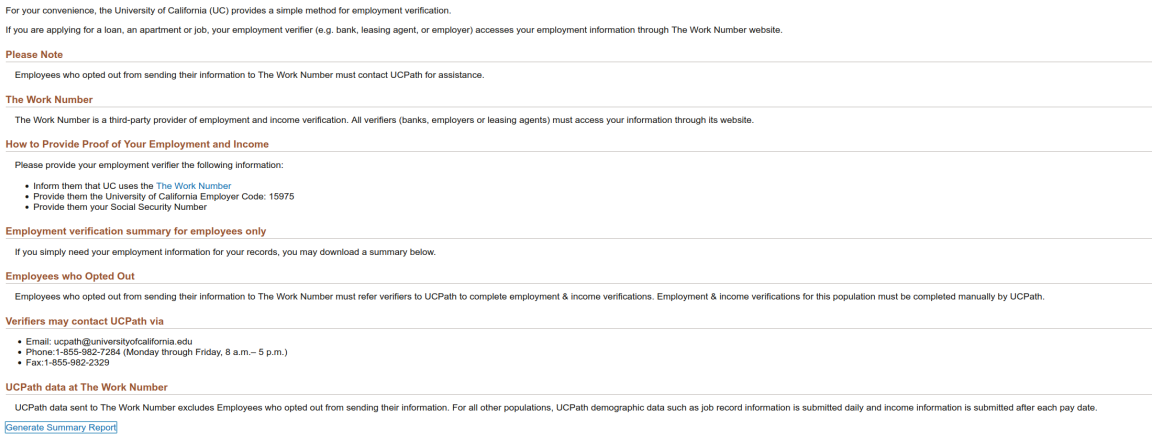
If you are a UC Berkeley student, the most straightforward way to get proof is to download a certificate of enrollment verification by going to CalCentral –> My Academics –> Enrollment Verification (under Academic Records) –> View or Print Enrollment via Self Service –> Obtain an enrollment certificate. This will give you a PDF enrollment certificate which you can screenshot and submit for proof of affiliation (you need to use a screenshot because the application does not accept PDFs).



For teacher roles, you will need to go to UCPath.

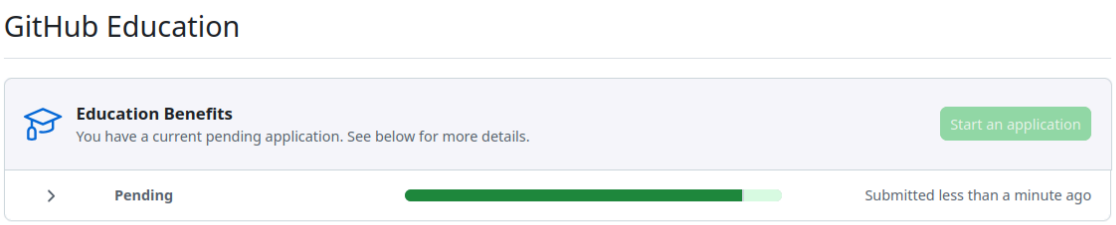


From the main page, you will want to click on `Income & Taxes` and go to `Verification of Employment` under `Payroll Information`.

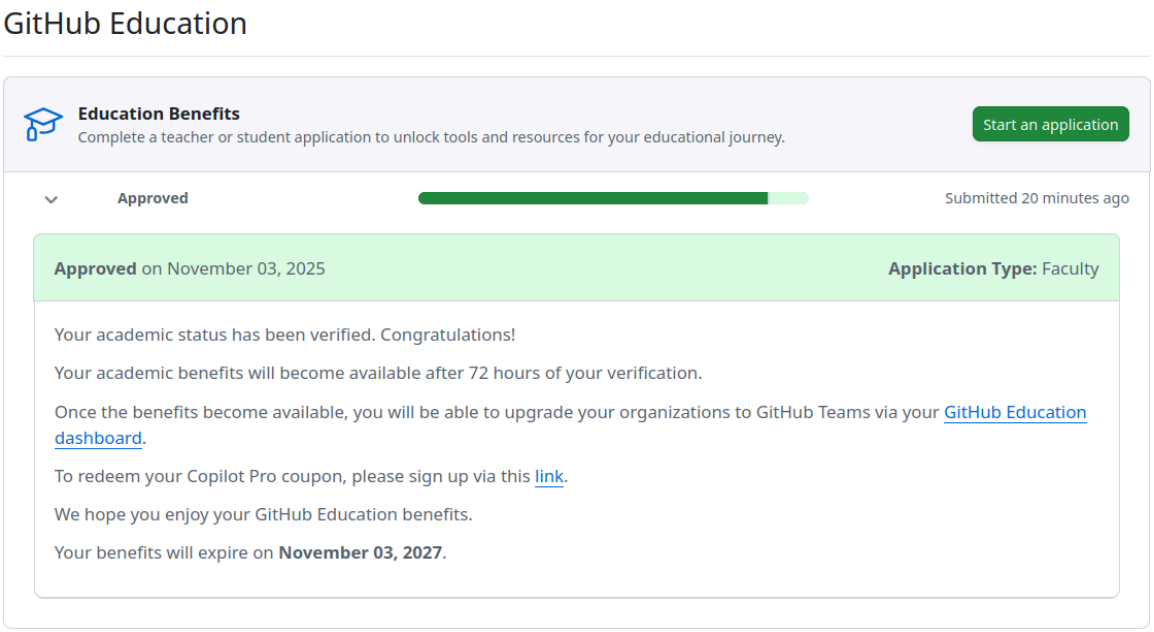


To get verification of your employment, you will then go to the bottom of the page and click on `Generate Summary Report`. You will be showed a pdf with your title, the current date of when

you generated report (today's date), and other information. Since you cannot submit PDFs, make sure you get a screenshot of the report and save it as a **.jpg** file.



Once you have submitted your proof, your application will show that it is Pending.



After some time, it will state that you are approved. This is not a full approval, and full approval will take some time to gain access to Copilot Pro.

3.3 Section on Getting Copilot Set Up (NEEDS WORK –> WAITING ON ACCESS)

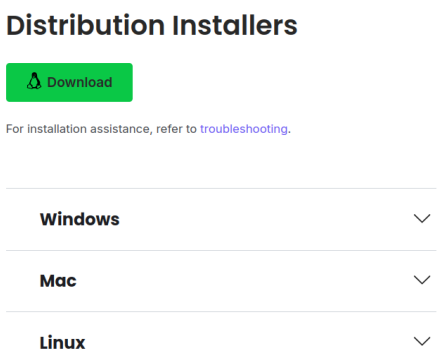
NEED TEXT HERE

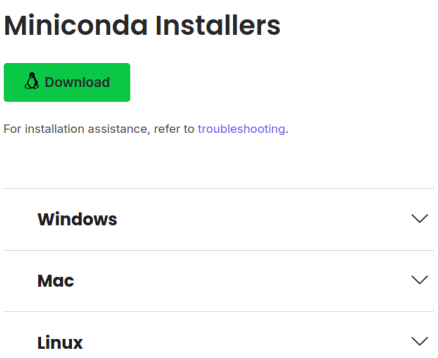
- ☒ Install GitHub
- ☒ Sign up for GitHub Copilot
- [] Install Python
- [] Install Visual Studio (VS) Code

- [] Adjust VSCode to work with Python
- [] Download Zip file

4 Installing Python

1. Go to anaconda to download Python
2. Sign up (can go through Google, entering the information, or GitHub)
3. Verify your Email Address
4. Choosing installer





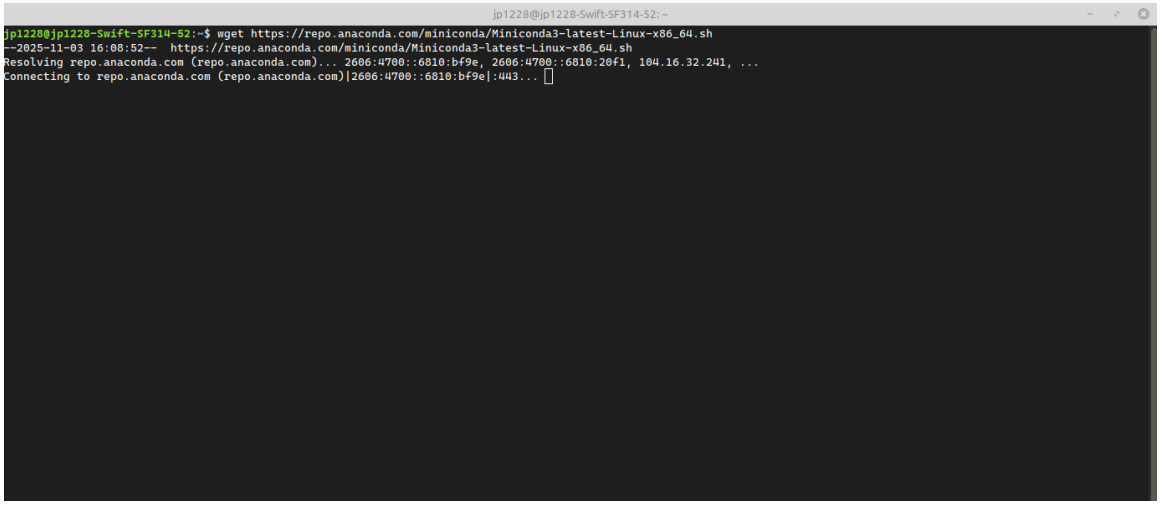
Free Download Sign In Get Demo >

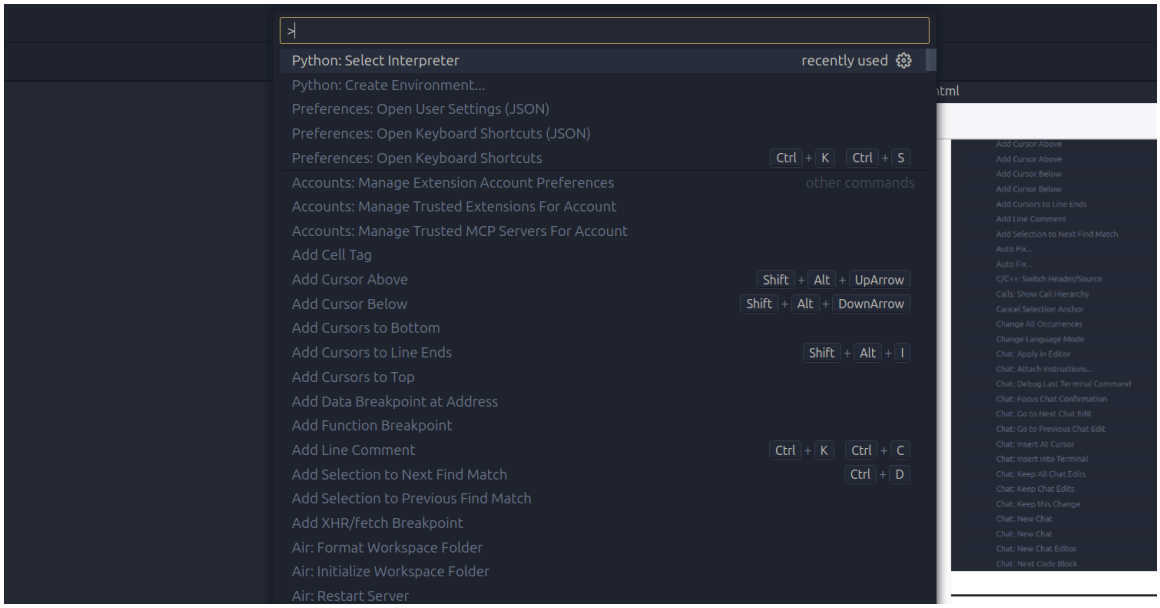
Download Now

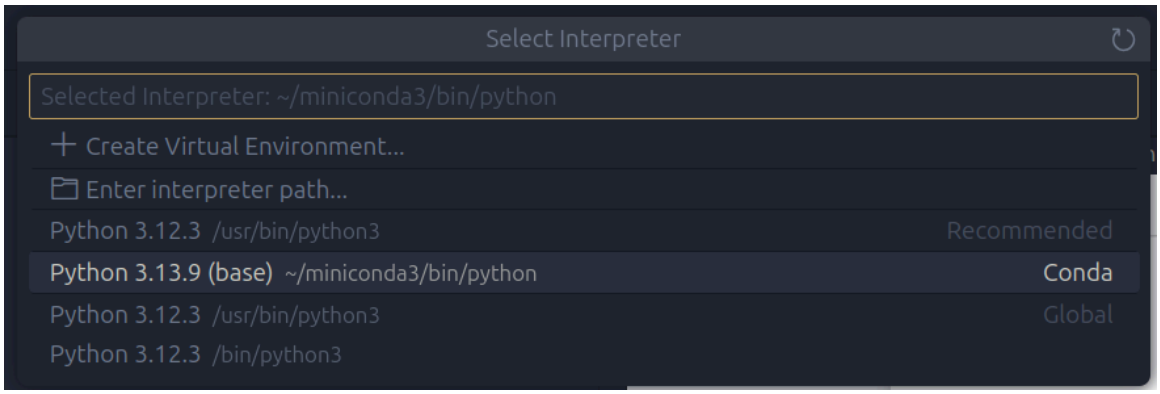
Get access in 30 seconds. Completely free.*

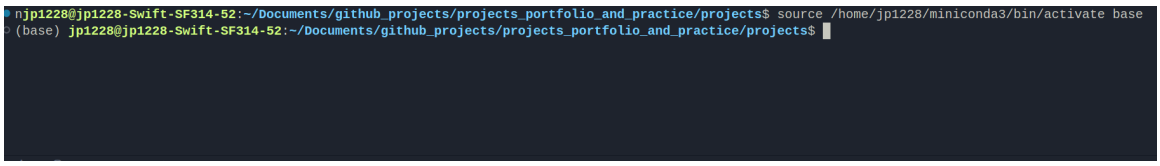
Get Started > Returning Users >

*Subject to our Terms of Service. Use of Anaconda's offerings at an organization of more than 200 employees/contractors requires a paid business license unless your organization is eligible for discounted or free use. See Pricing.









Linux Installation

Go here (https://www.anaconda.com/docs/getting-started/miniconda/install)

Find the linux code to install miniconda

```
wget https://repo.anaconda.com/miniconda/Miniconda3-latest-Linux-x86_64.sh
```

You can then install the file by running the following code and answering the prompts in the terminal. You can answer yes and press enter to keep the default values.

```
bash ~/Miniconda3-latest-Linux-x86_64.sh
```

Closing out of your terminal, you will be ready to use Python in VSCode.

Choosing an intepreter

Go to the command palette (Windows/Linxu: Ctrl + Shift + P; Mac: Cmd + Shift + P) and search for `Python: Select Interpreter`.

Now when you open a bash terminal, you will see that it starts with:

```
source <miniconda-path-location>/bin/activate base
```

That means you are now using conda as your Python interpreter. This should translate to using the Jupyter notebook for the workshop materials.

More information can be found on environments here, as well as general start up assistance with Python in VSCode.

To install Python, you will go to The Comprhensive R Archive Network (CRAN) website. There you can download Python for your operating system.

4.1.a Mac

Follow the directions for installing Python. You can keep the defaults for everything. When installed, you should see a message stating `The installation was successful.`

4.1.b Windows

Follow the directions for installing Python. You can keep the defaults for everything. When installed, you should see a message stating `The installation was successful.`

4.1.c Linux

There are some really good instructions on how to install R here.

```
sudo apt update
sudo apt install -y build-essential libcurl4-openssl-dev libssl-dev libxml2-dev

# optionally
#  sudo  apt  install  -y  libfontconfig1-dev  libharfbuzz-dev  libfribidi-dev
libfreetype6-dev libpng-dev libtiff5-dev libjpeg-dev
```

- [] Install GitHub
- [] Sign up for GitHub Copilot
- [] Install Python
- [] Install Visual Studio (VS) Code
- [] Adjust VSCode to work with Python
- [] Download Zip file

5 Install VSCode

Let's move forward with installing VSCode. You can install VSCode here for your operating system. Below are detailed instructions on how to install VSCode.

5.1.a Mac

Link: https://code.visualstudio.com/docs/setup/mac

Follow the directions to install VSCode. Below, I will include some helpful tips for VSCode extensions that may help when using Python.

5.1.b Windows

Link: https://code.visualstudio.com/docs/setup/windows

Follow the directions to install VSCode. Below, I will include some helpful tips for VSCode extensions that may help when using Python. **For Windows users, it is recommended to check "Save version number in registry" during installation so that the R extension can find your R installation automatically. If you have not done this you may need to add the location of your R to your PATH manually (see FAQ 3.1 I am using windows and my VS Code can't find R!).**

5.1.c Linux

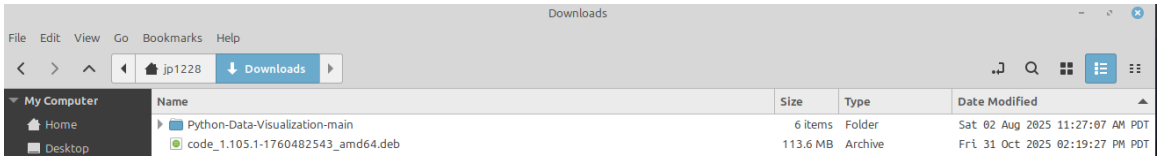
> **Note**: Everything below shows installation using Linux Mint.

```
System:
  Host: jp1228-Swift-SF314-52 Kernel: 6.8.0-87-generic arch: x86_64 bits: 64
  Desktop: Cinnamon v: 6.4.8 Distro: Linux Mint 22.1 Xia
```

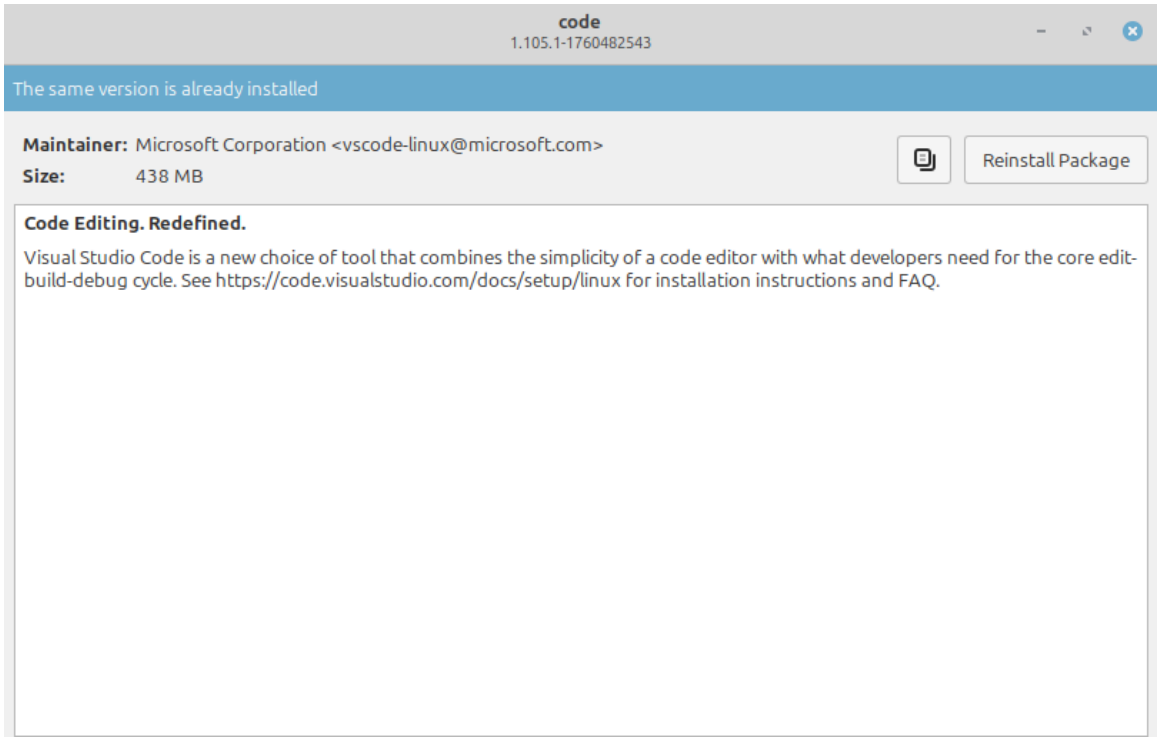
Link: https://code.visualstudio.com/docs/setup/linux

Option 1: Use the Link

1. Click on the link above and open the download file.

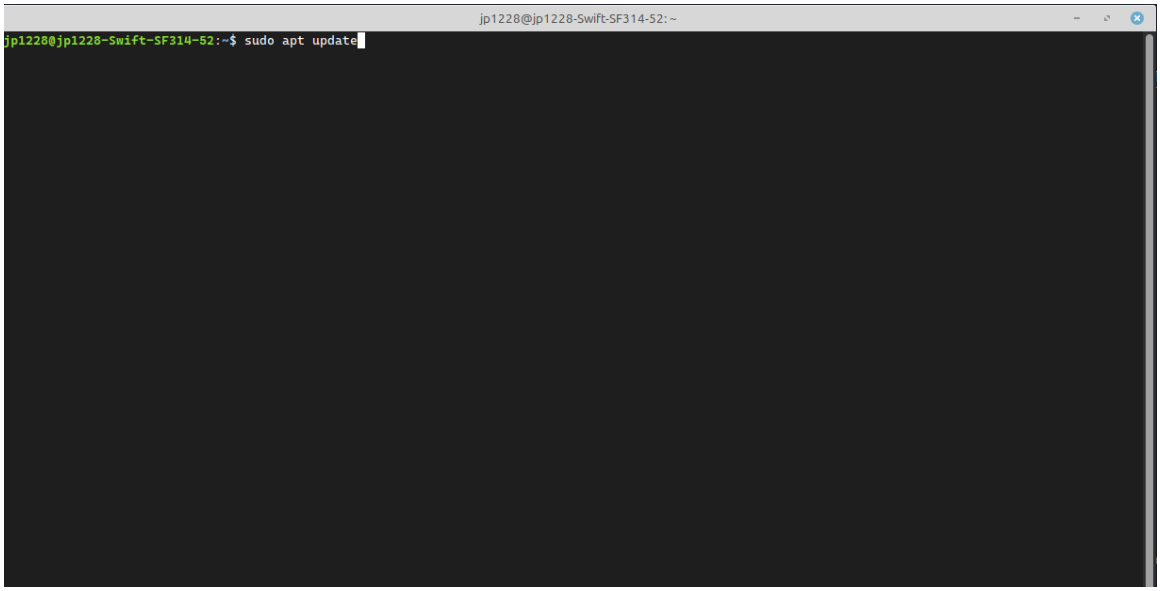


2. Click Install Package to start the install.



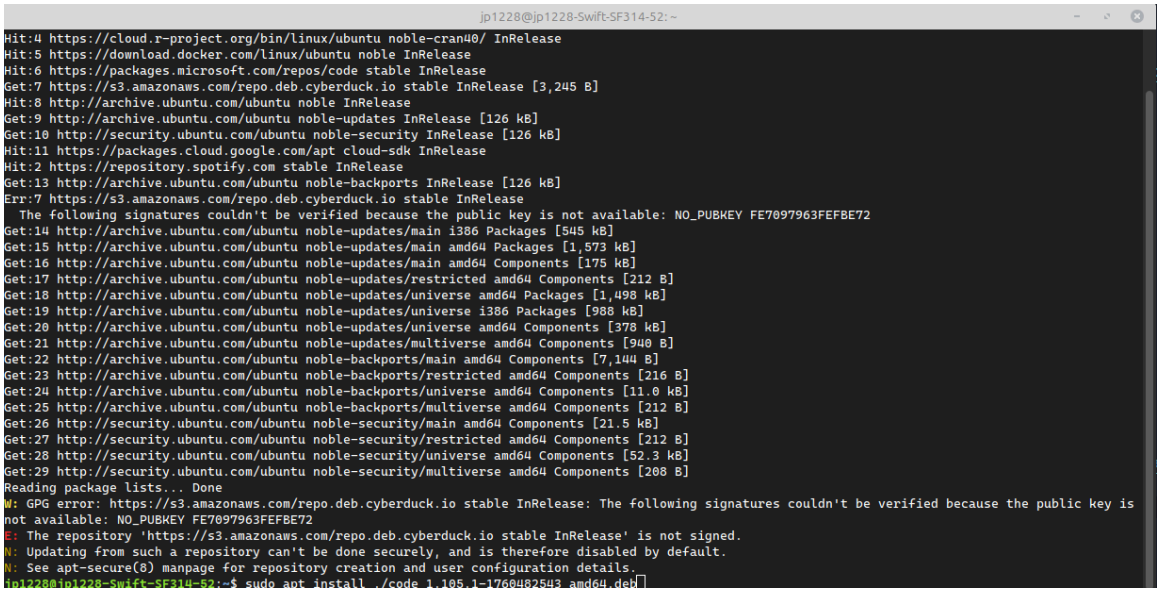
Option 2: Download Using Terminal

1. Update your programs.



2. Then you will install the file that you downloaded from this page as shown below. Your file will look different, depending on the version and differences in your linux distribution, but it should start downloading after running the code below.

```
sudo apt install ./<file_name>.deb
# include the name of your file and change <file_name> to the name of your file
```



5.2 VSCode Documentation

When you installed VSCode, it should have brought you to the documentation page. If not, you can find all the documentation here. This tutorial also provides an in-depth tutorial on getting started with VSCode.

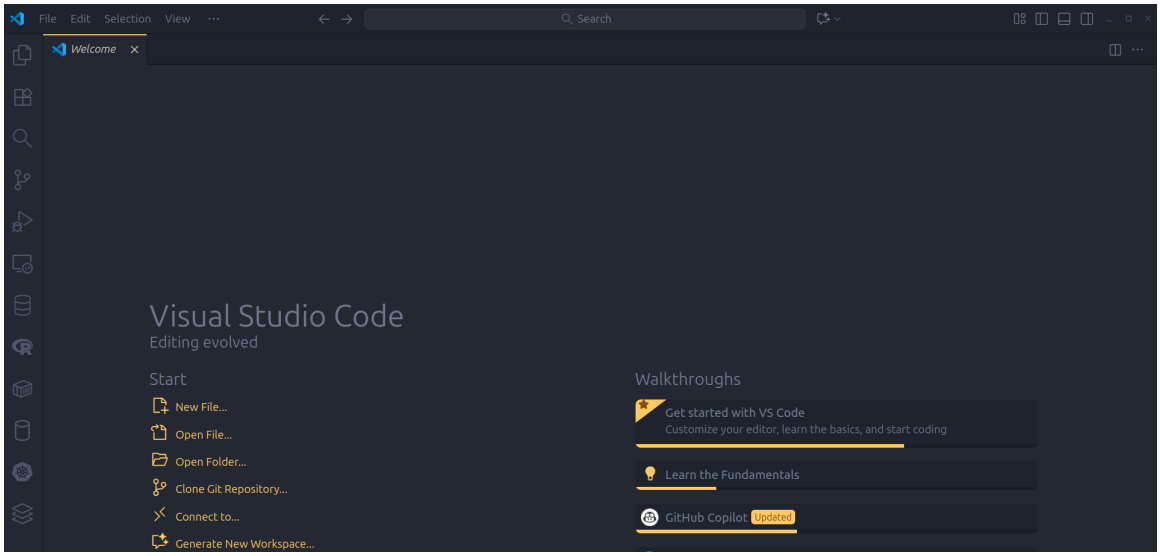
- [] Install GitHub
- [] Sign up for GitHub Copilot
- [] Install Python
- [] Install Visual Studio (VS) Code
- [] Adjust VSCode to work with Python
- [] Download Zip file

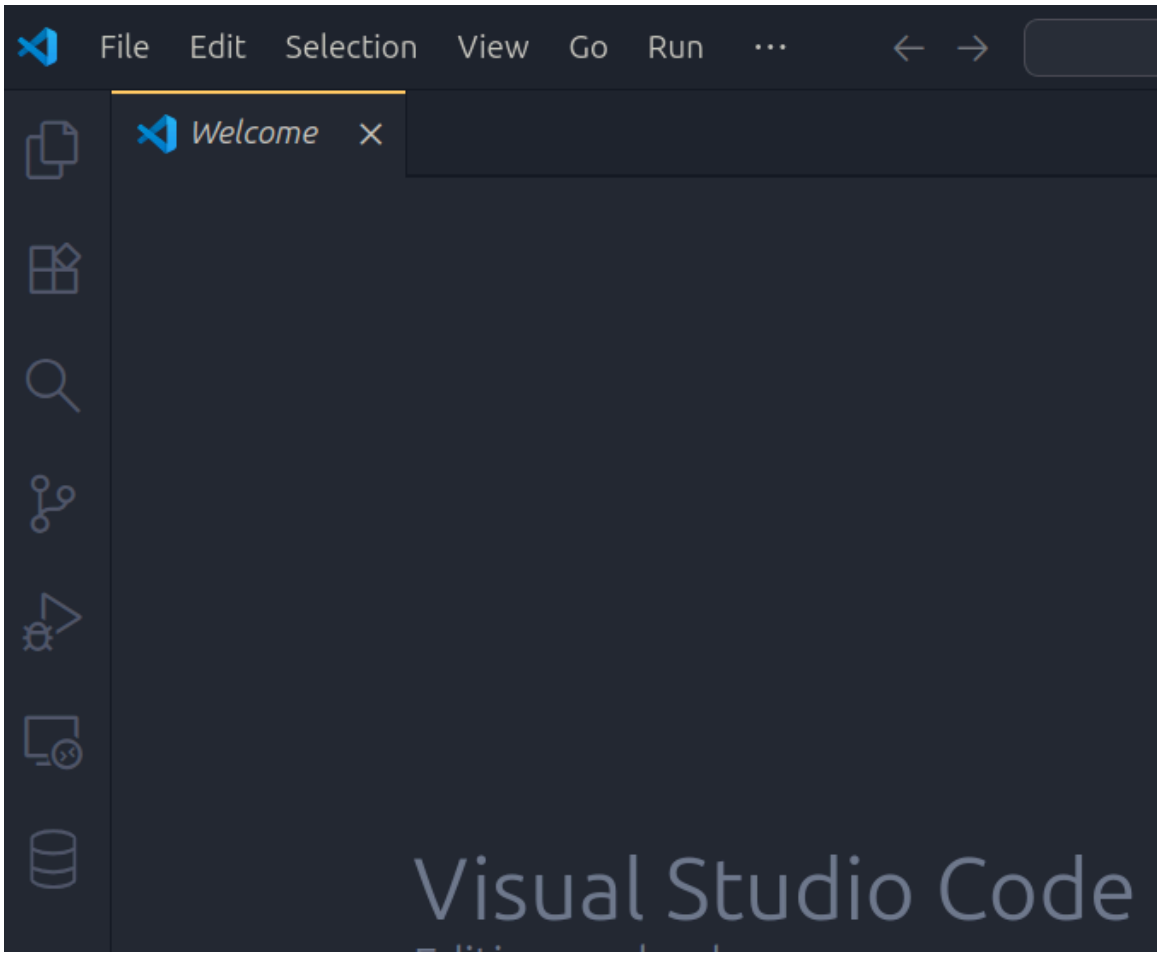
6 Setting up Python in VSCode

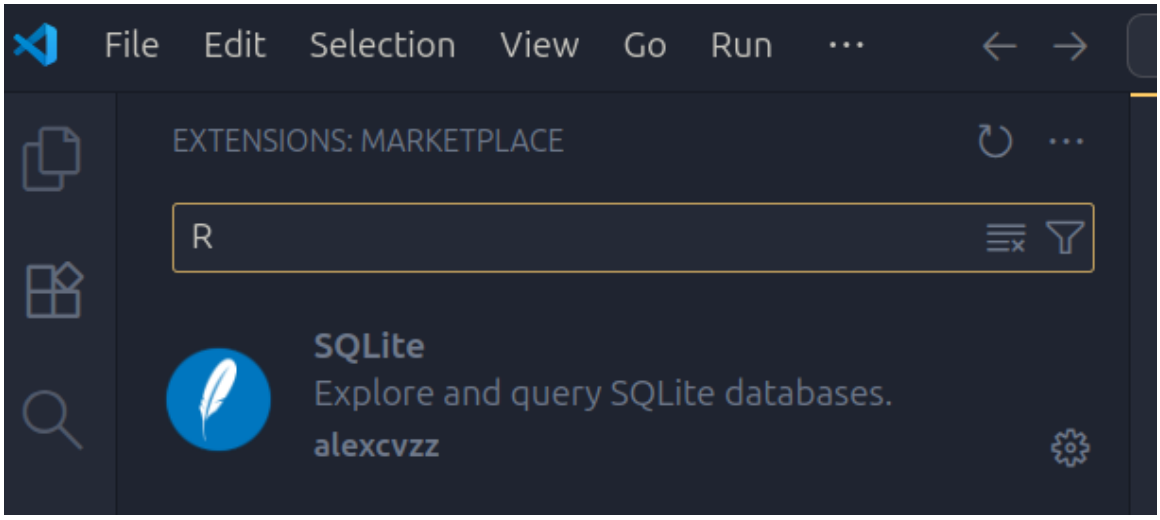
Note Your VSCode will look slightly different, as the screenshots are from a custom VSCode theme. Additionally, the ordering and number of tabs on the sidebar may be different from the VSCode default.

Depending on the version of Python you installed, there may be some issues in incorporating some of the packages. I will include alternatives to try and make sure everything works.

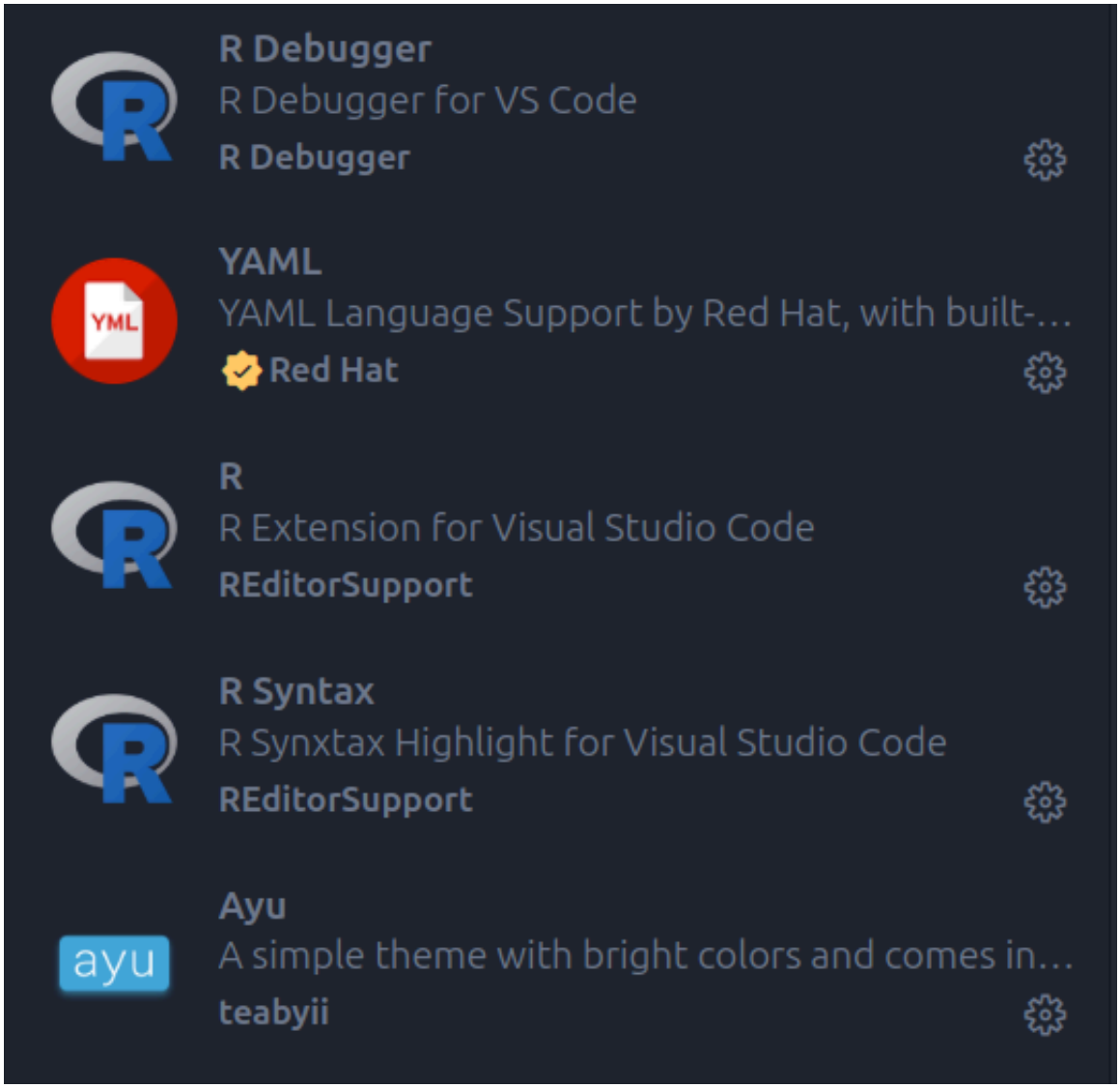
If you have not downloaded Python, then you will want to do that first. Once you have Python, then you can move forward with installing extensions for Python.


File Edit Selection View
Search
Welcome
Visual Studio Code
Editing evolved
Start
New File...
Open File...
Open Folder...
Clone Git Repository...
Connect to...
Generate New Workspace...
Walkthroughs
Get started with VS Code
Customize your editor, learn the basics, and start coding
Learn the Fundamentals
GitHub Copilot
Updated

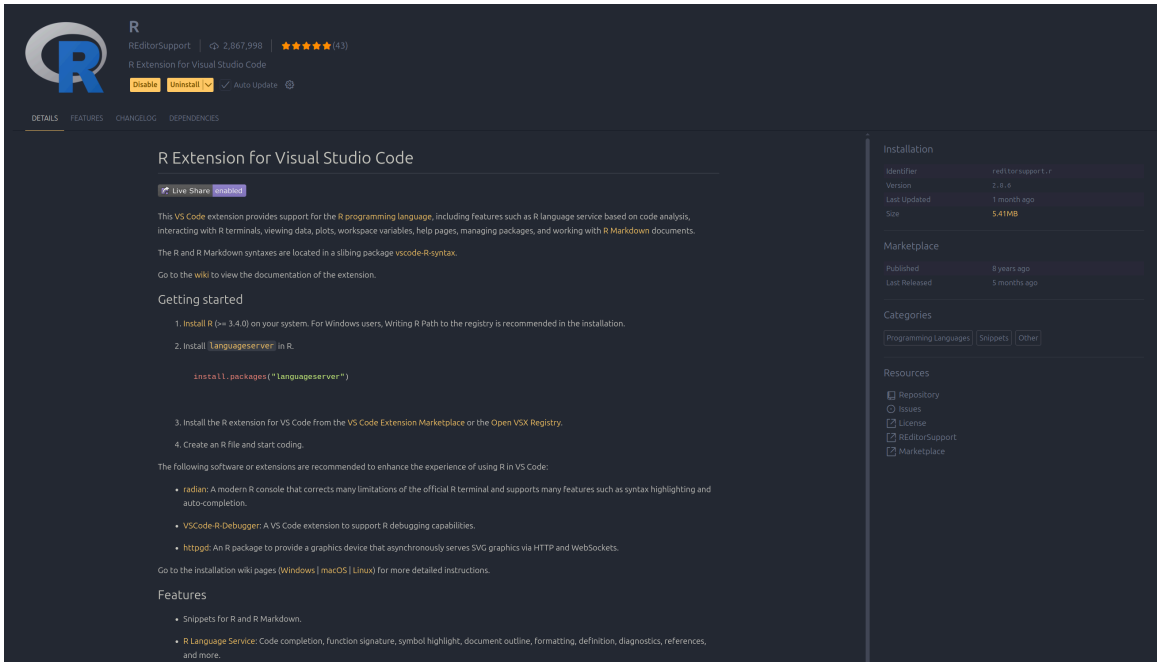

File Edit Selection View Go Run
Welcome
Visual Studio Code



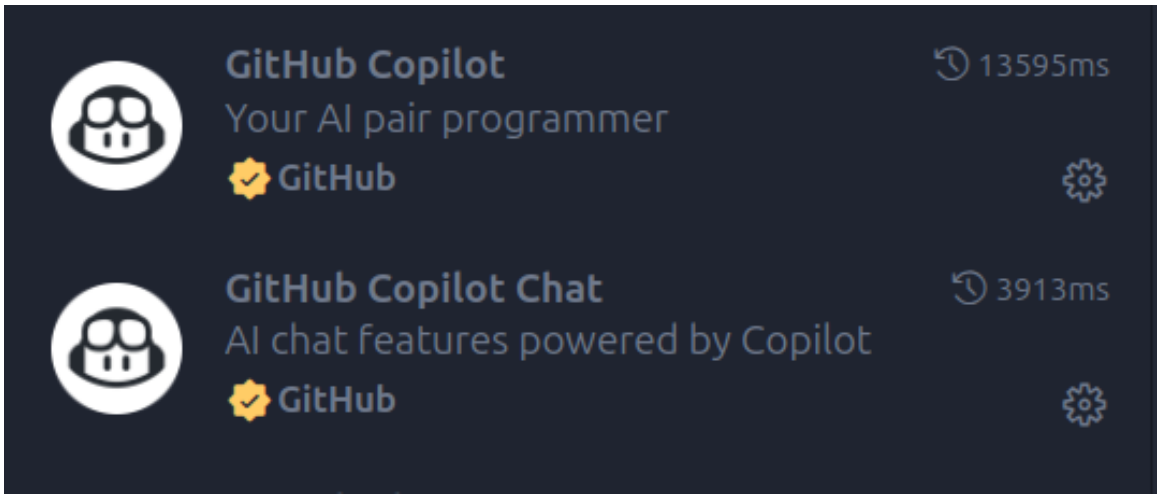
The easiest way of finding all the extensions you will need is to use the search bar. *You can also use this to find a theme for your VScode. See several examples of themes here that you can then search for by name.



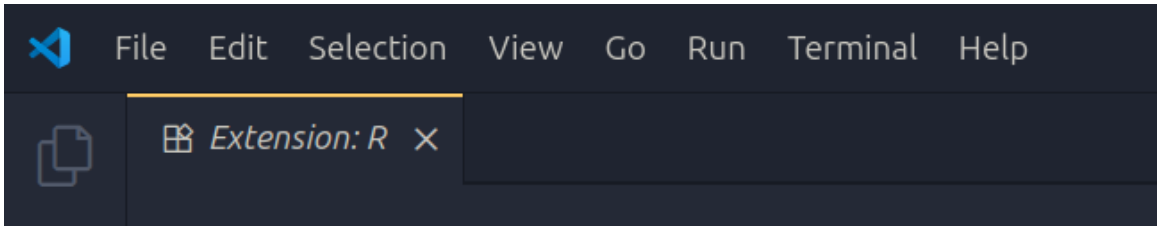
In addition to installing the Python extension, I also have installed the **R Debugger** and the **R Syntax** extensions. These add some additional tools when using Python.



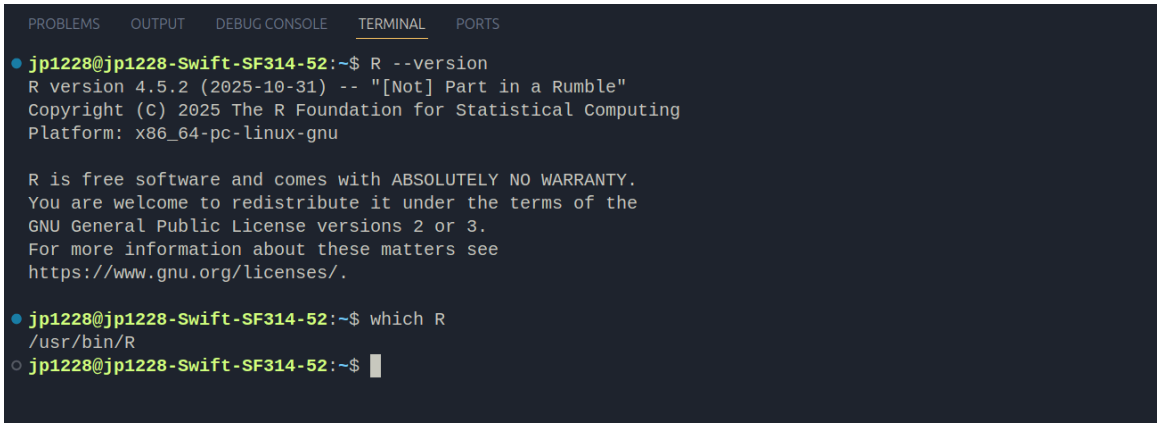
You can also make sure that your GitHub Copilot is installed. There are two extensions that should be installed, `GitHub Copilot` and the `GitHub Copilot Chat` extension.

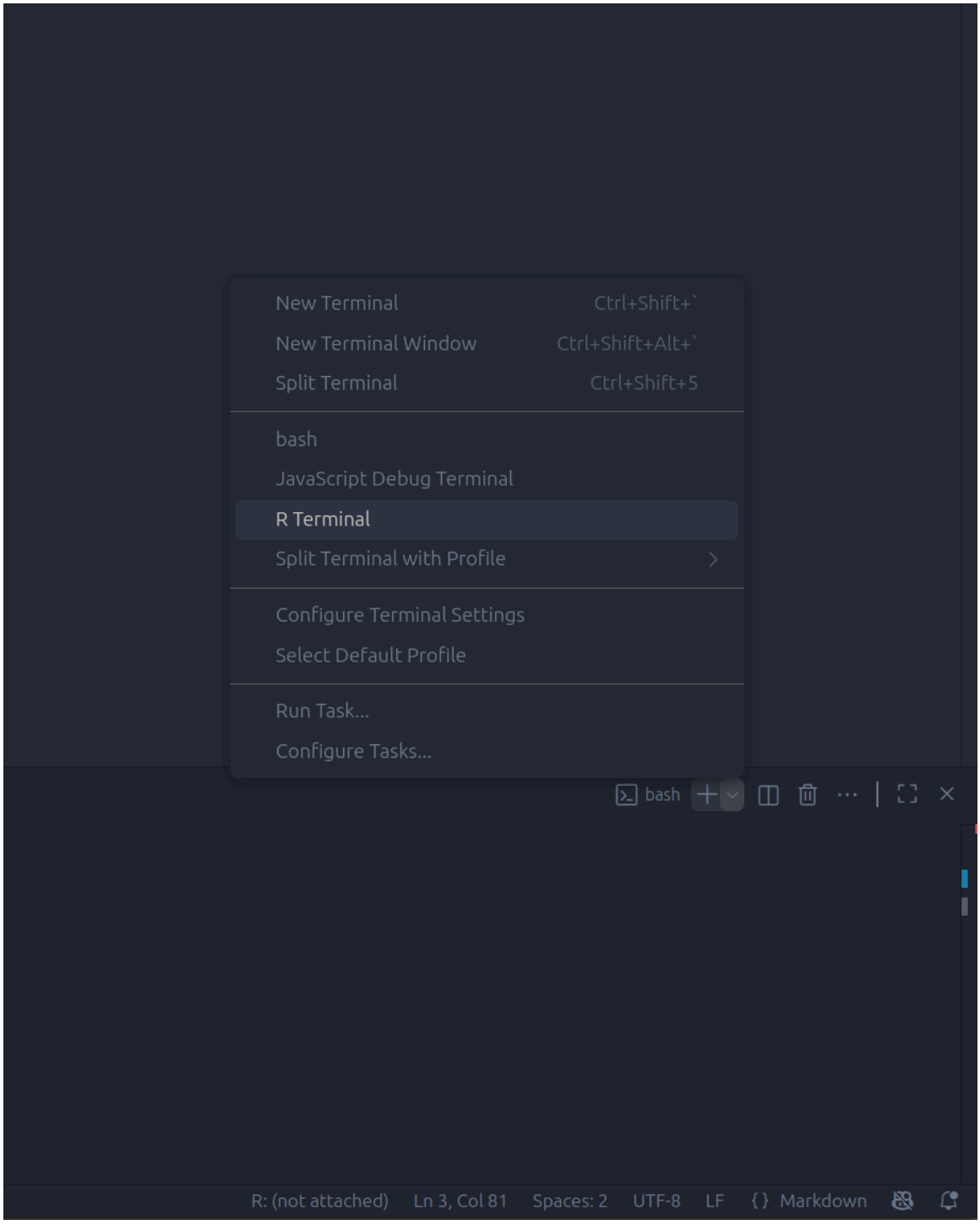


From these instructions, you will want to install the `languageServer` and the `httpgd` packages. You can also install the `radian` package if you would like; however, the trade-offs are not much better for the amount of effort to get it working in VSCode.



To install these packages, you can go to the top and click on `Terminal` followed up by `New Terminal`. That should open a terminal in bash, like in the screenshot below.




New Terminal Ctrl+Shift+`
New Terminal Window Ctrl+Shift+Alt+`
Split Terminal Ctrl+Shift+5
bash
JavaScript Debug Terminal
R Terminal
Split Terminal with Profile
Configure Terminal Settings
Select Default Profile
Run Task...
Configure Tasks...
bash
R: (not attached)
Ln 3, Col 81
Spaces: 2
UTF-8
LF
{} Markdown

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

Copyright (C) 2025 The R Foundation for Statistical Computing
Platform: x86_64-pc-linux-gnu

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> install.packages("languageserver")
```

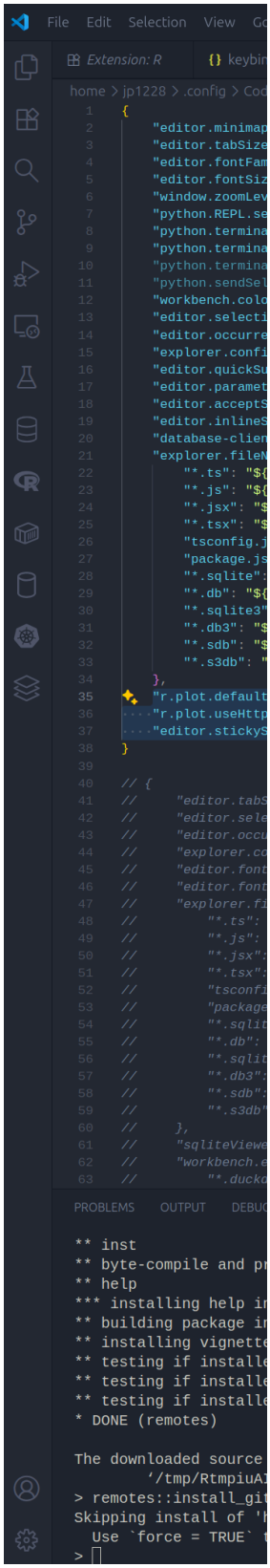
```
> install.packages("httpgd")
Installing package into ‘/home/jp1228/R/x86_64-pc-linux-gnu-library/4.5’
(as ‘lib’ is unspecified)
Warning message:
package ‘httpgd’ is not available for this version of R

A version of this package for your version of R might be available elsewhere,
see the ideas at
https://cran.r-project.org/doc/manuals/r-patched/R-admin.html#Installing-packages
>
```

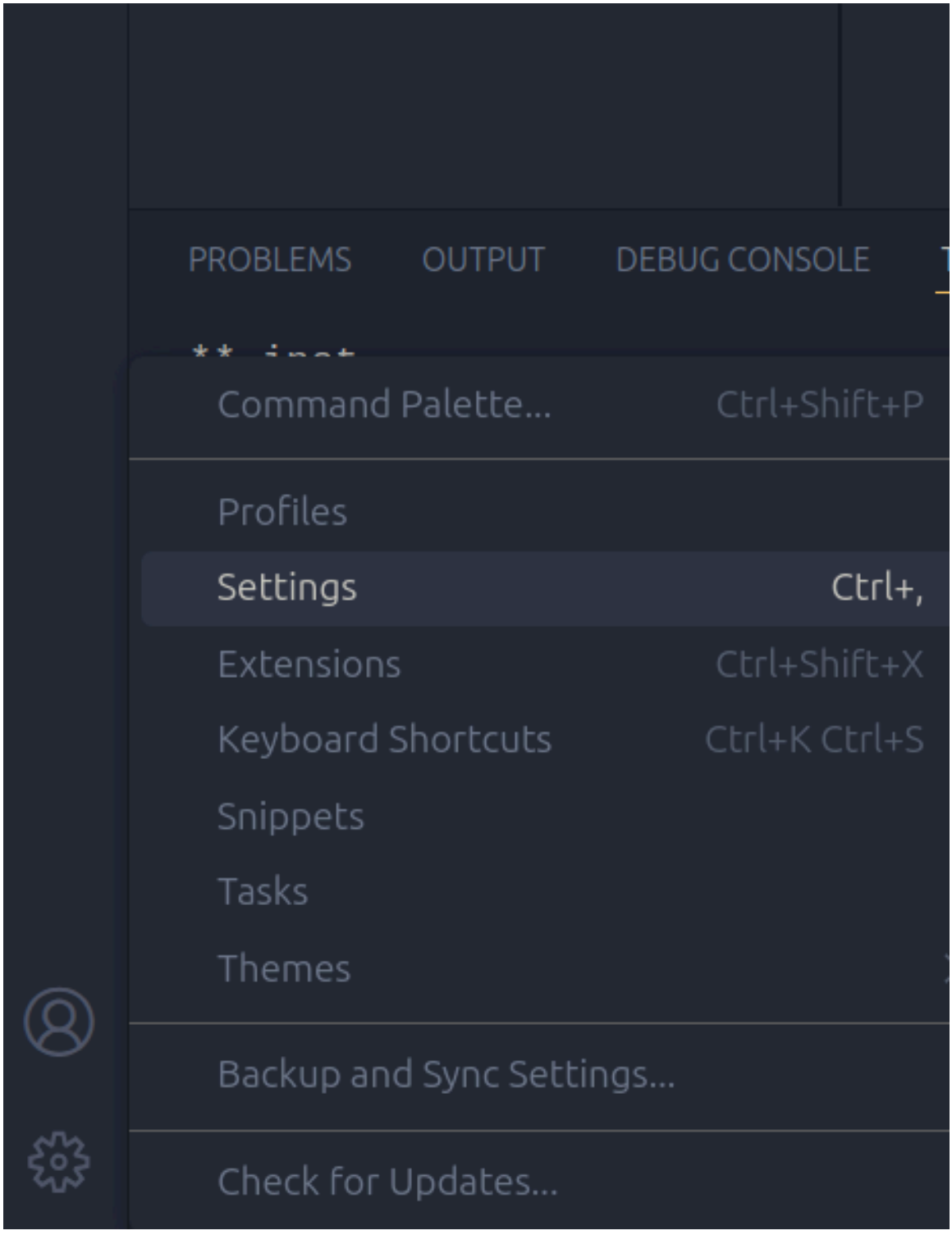
```
> install.packages("remotes")
Installing package into ‘/home/jp1228/R/x86_64-pc-linux-gnu-library/4.5’
(as ‘lib’ is unspecified)
trying URL 'https://p3m.dev/cran/__linux__/manylinux_2_28/latest/src/contrib/remotes_2.5.0.tar.gz'
Content type 'binary/octet-stream' length 165146 bytes (161 KB)
==================================================
downloaded 161 KB

* installing *source* package ‘remotes’ ...
** this is package ‘remotes’ version ‘2.5.0’
** package ‘remotes’ successfully unpacked and MD5 sums checked
** using staged installation
** R
** inst
** byte-compile and prepare package for lazy loading
** help
*** installing help indices
** building package indices
** installing vignettes
** testing if installed package can be loaded from temporary location
** testing if installed package can be loaded from final location
** testing if installed package keeps a record of temporary installation path
* DONE (remotes)

The downloaded source packages are in
        ‘/tmp/RtmpiuAIbT/downloaded_packages’
> remotes::install_github("nx10/httpgd")
```

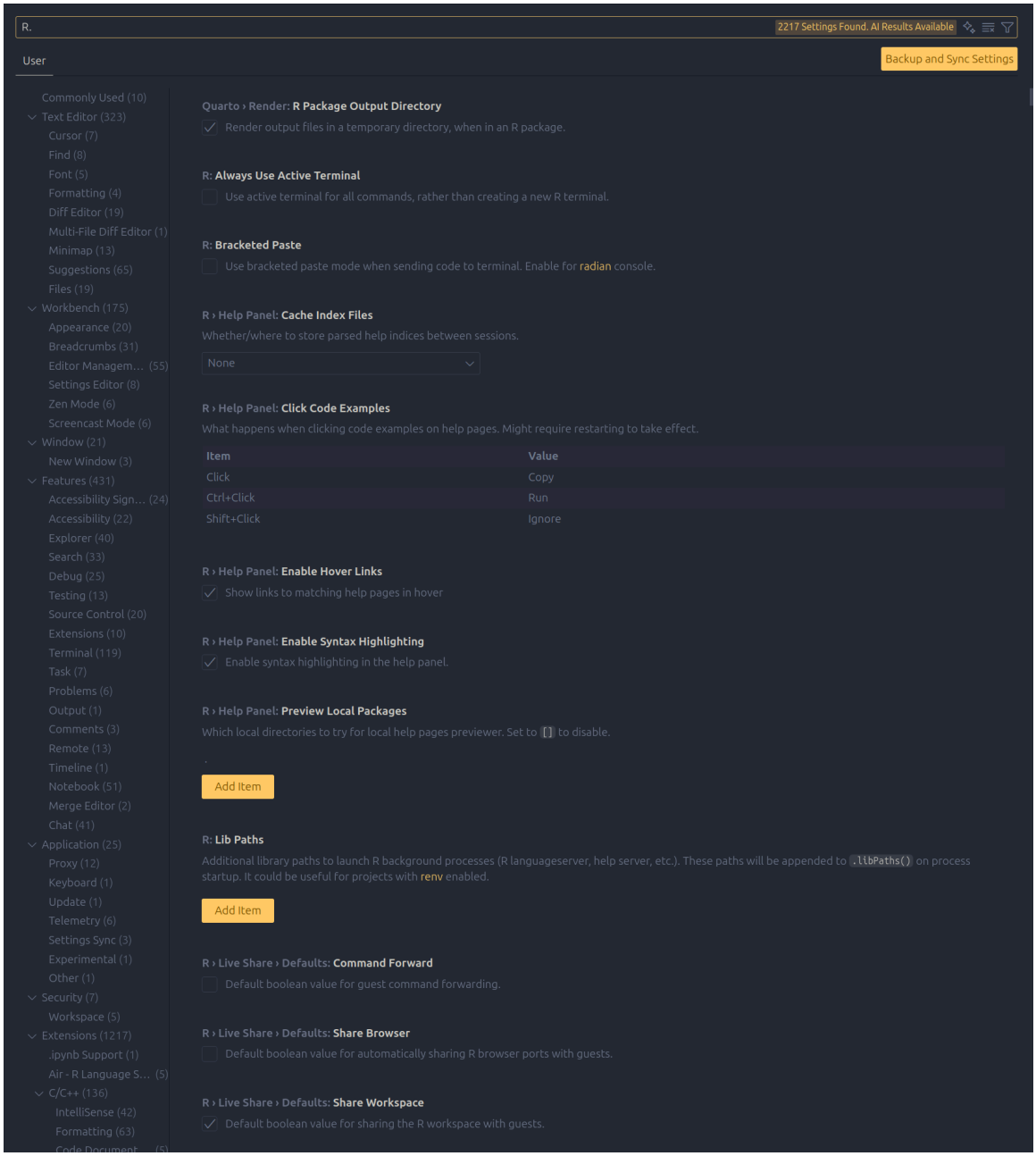
You can get the `httpgd` package by first installing the `remotes` package. This will then allow you to install the development version from GitHub. If you want to follow along with the instructions from the developer(s) of the h`httpgd` package, you can follow the installation instructions here.

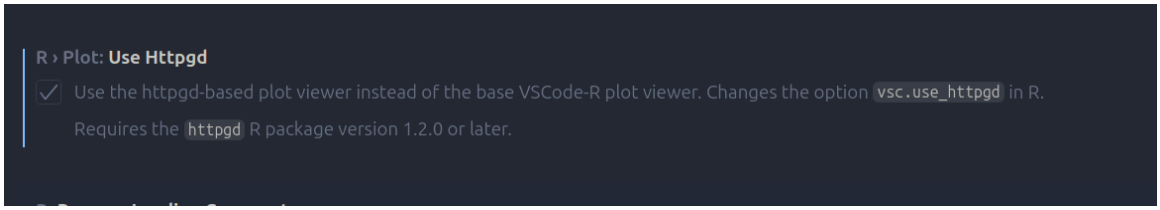
Lastly, I will show some extra customizable settings to make VSCode similar to using RStudio. Below are some settings that you can change in your `settings.json` file. You can also change the settings by going to the gear (see directions below) to make changes to your VSCode.

1. Making Changes by Settings Tab


File Edit Selection View
Extension: R
home > jp1228 > .config
PROBLEMS OUTPUT
** inst
** byte-compile and
** help
*** installing help
** building package
** installing vignette
** testing if installe
** testing if installe
** testing if installe
* DONE (remotes)
The downloaded source
> remotes::install_gi
Skipping install of
Use `force = TRUE`


PROBLEMS
OUTPUT
DEBUG CONSOLE
Command Palette... Ctrl+Shift+P
Profiles
Settings Ctrl+,
Extensions Ctrl+Shift+X
Keyboard Shortcuts Ctrl+K Ctrl+S
Snippets
Tasks
Themes
Backup and Sync Settings...
Check for Updates...

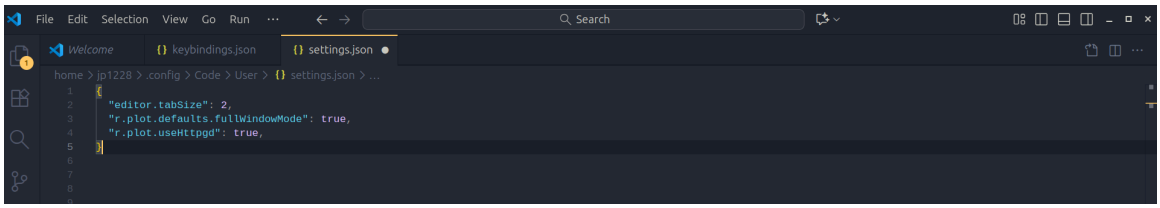




2. Making Changes Using the settings.json File

To get to the `settings_json` file, you will use the following keybinding shortcut (Windows/Linux: Ctrl + Shift + P, Mac: Cmd + Shift + P). This will open the command palette where you can then search for `Preferences: Open User Settings (JSON)`. Here you can then copy and paste the code below to use the `httpgd` package and tab 2 spaces rather than the default 4. If you change your theme, this will also show up here as well as preferences made for other extensions. Once you have made these changes, you can save your settings and close out of the `settings.json` file.

```
{
  "editor.tabSize": 2,
  "r.plot.defaults.fullWindowMode": true,
  "r.plot.useHttpgd": true,
}
```



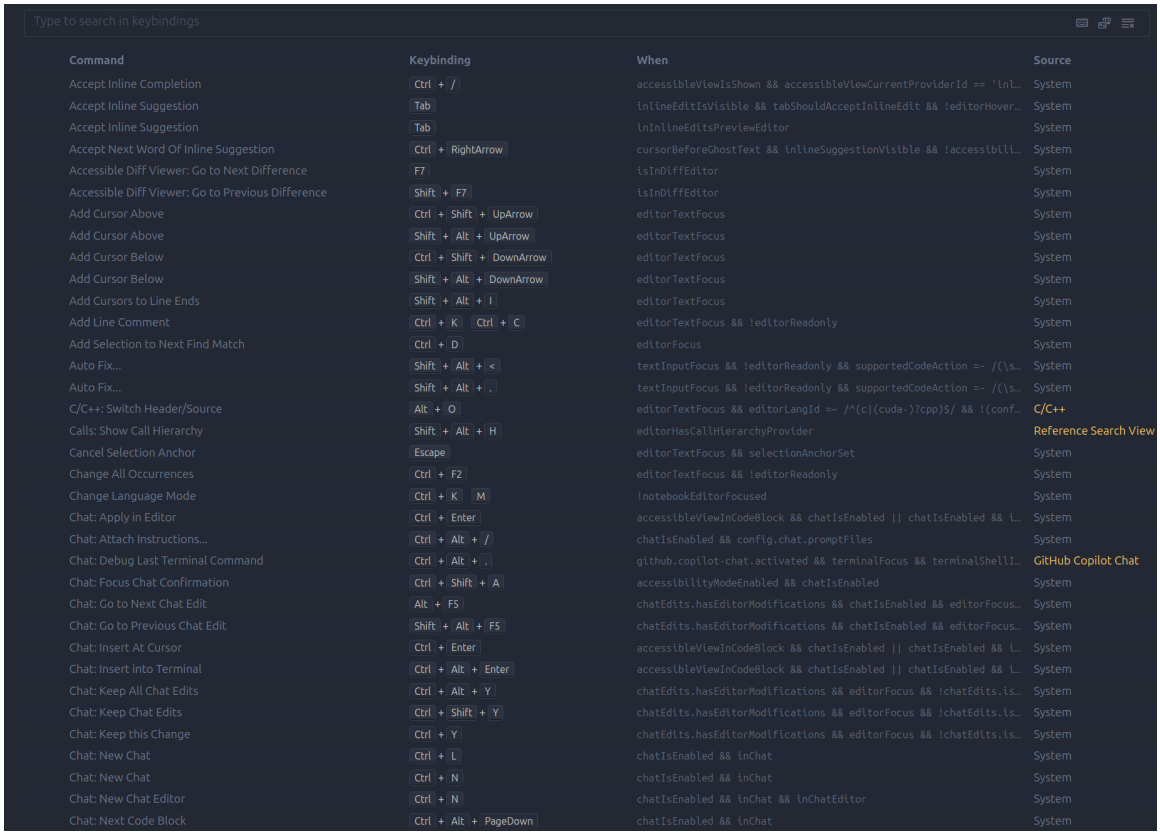
Another preference is to include shortcuts for some common RStudio shortcuts. The main two are being able to comment in/out code in your Python scripts. Similar to RStudio, you can add the shortcuts below using the `keybindings.json` file.

You will use the keybinding shortcut (Windows/Linux: Ctrl + Shift + P, Mac: Cmd + Shift + P) to get to the command palette again. Here you can search for the `Preferences: Open Keyboard Shortcuts (JSON)` and copy and paste the code below. **Note** Be aware that if you do not have $ version 4.1.0 of greater, you will need to change the `{ "text": "|>" }` to `{ "text": "%>%" }`. You can then save the file and close out and you should be able to use these shortcuts.

```
[
  {
    "key": "ctrl+shift+c",
    "command": "editor.action.commentLine",
    "when": "editorTextFocus && !editorReadonly"
  },
  {
    "key": "ctrl+shift+m",
    "command": "type",
    "args": { "text": "|>" },
    "when": "editorTextFocus && editorLangId == 'r'"
  }
]
```

To make changes to any other shortcuts, you can use the command palette to search `Keyboard Shortcuts` to change any other shortcuts. **Note** Be aware that you could possibly overwrite other

important VSCode functions so be cautious when making changes. You can type in the shortcut you want to create in the search bar at the top to see what current functions use that shortcut.

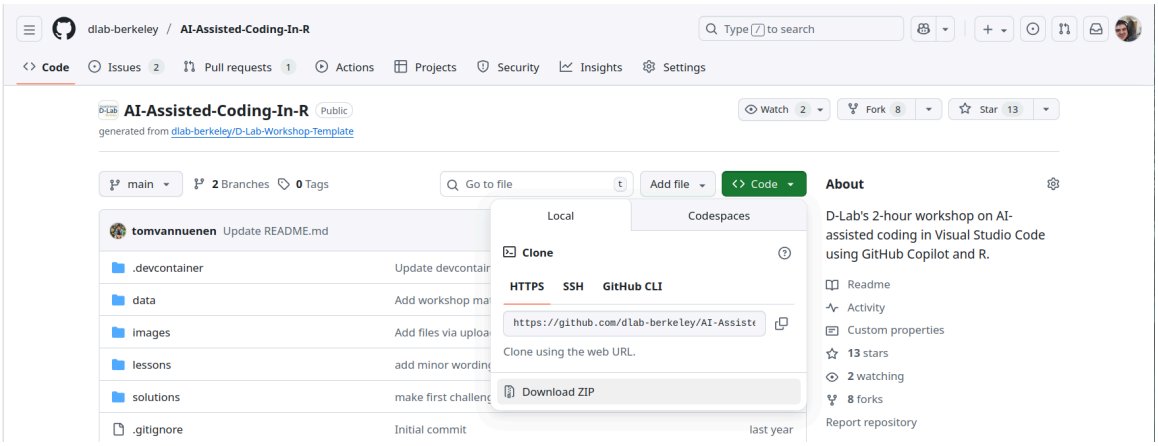


- [] Install GitHub
- [] Sign up for GitHub Copilot
- [] Install Python
- [] Install Visual Studio (VS) Code
- [] Adjust VSCode to work with Python
- [] Download Zip file

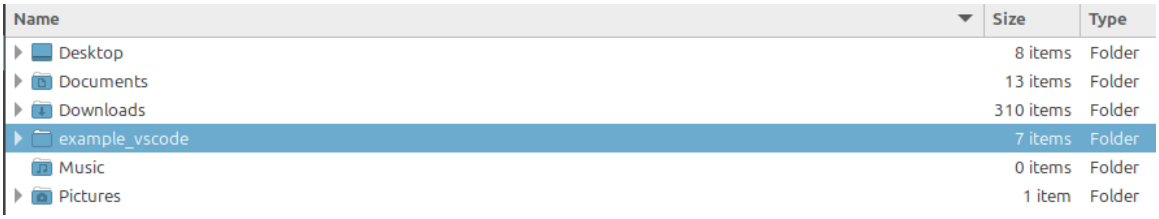
7 Download Zip File

To get all the materials for the AI-Assisted-Coding-In-Python workshop, you will want to download the Zip file here.

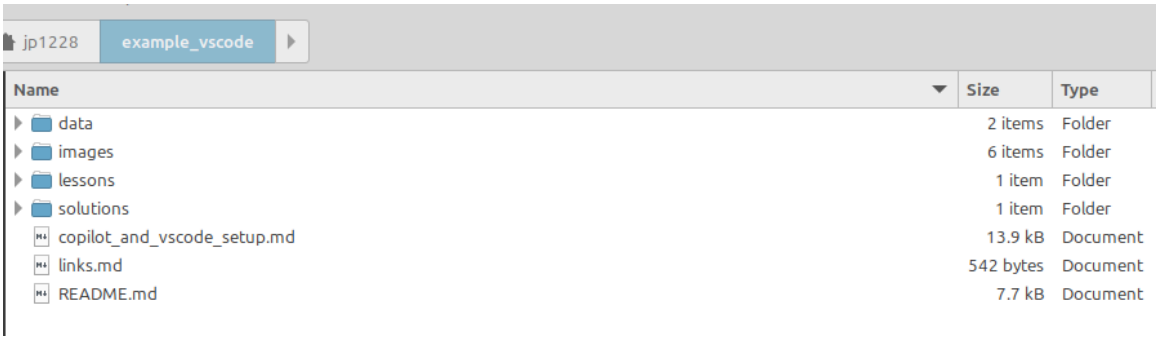
Note You may want to wait until the day of your workshop to make sure you download the most recent version of the contents.



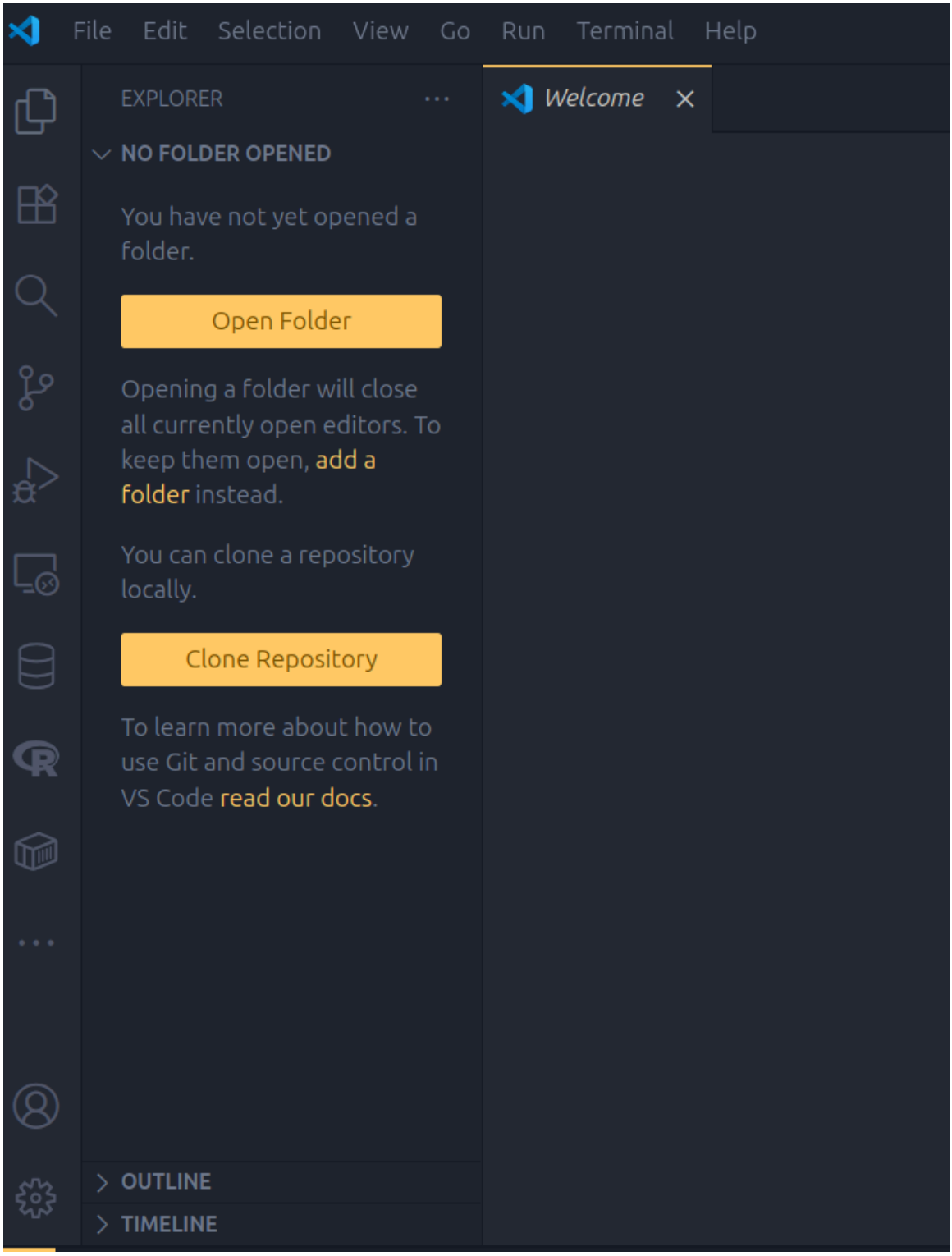
Once you are at the D-Lab GitHub repository for the AI-Assisted-Coding-In-Python workshop, you will go to the green button that says <Code> with a dropdown menu. Clicking on the button will give you options for how to put the repository's contents on your local computer. For now, you can Click on the Download ZIP to download a ZIP file. The ZIP file should then be in your Downloads folder. There you can extract the contents of the ZIP file.



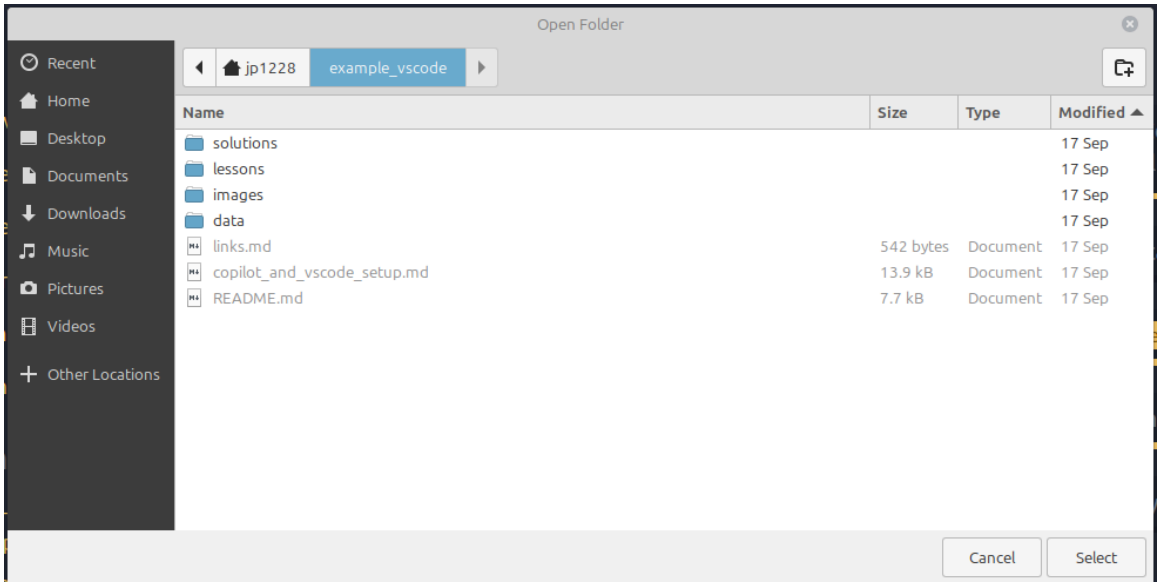
Once you have extracted the contents of your ZIP file. I put all of the contents in a new folder. You can put this folder wherever it makes sense to you. I have my folder (which I named example_vscode) under my username folder. You can create the folder on your Desktop for easy access or within your Documents folder if you would like.



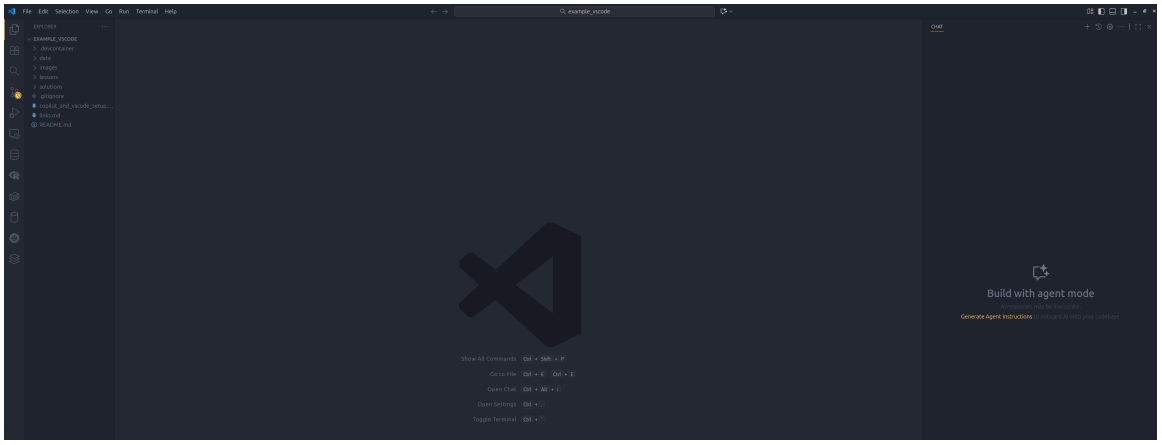
Within the folder, I copied and pasted everything from the ZIP file into this folder. The files should look similar to the screenshot above.



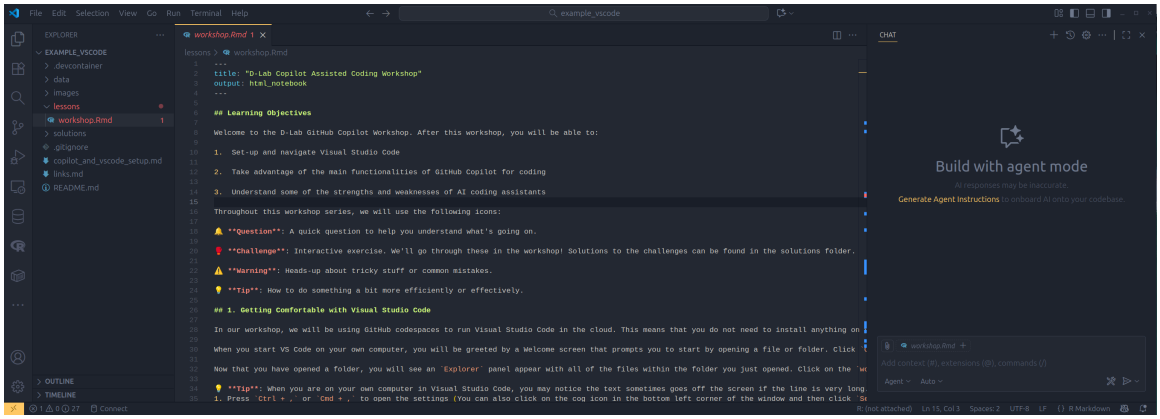
Now you can open VSCode and click on the folder tab on the left sidebar. There you can click on the option `Open Folder` to open the folder with the workshop contents.



In my case, I will look for my `example_vscode` folder and click select at the bottom to start VSCode from this folder.



Once you select your folder, your VSCode will populate with your workshop files on the left and a tab on the right for your prompts with GitHub Copilot.



Finally, you can click on the `lessons` folder on the left and click on your notebook.

Congrats! You are now ready for your AI-Assisted-Coding-In-Python workshop.

- ☒ Install GitHub
- ☒ Sign up for GitHub Copilot
- ☒ Install Python
- ☒ Install Visual Studio (VS) Code
- ☒ Adjust VSCode to work with Python
- ☒ Download Zip file