

Walkthrough of Installing Everything Needed for GitHub Copilot

2025-11-03

Table of contents

1	Checklist of Everything Needed to Install for AI-Assisted-Coding-In-Python	1
2	Installing GitHub	2
2.1	Adding Additional Email Addresses	10
2.1.a	Optional Two-Factor Authentication (2FA)	10
2.2	GitHub TLDR	12
3	Sign Up for GitHub Copilot (THIS SECTION NEEDS WORK)	13
3.1	Copilot Pro (Instructions for Verification)	13
3.2	Free Version	13
3.3	Copilot TLDR	13
4	Installing Python	13
4.1.a	Mac	13
4.1.b	Windows	13
4.1.c	Linux	13
5	Install VSCode	14
5.1.a	Mac	14
5.1.b	Windows	14
5.1.c	Linux	14
5.2	VSCode Documentation	16
6	Setting up Python in VSCode	16
7	Download Zip File	29
8	Python Specifics	33

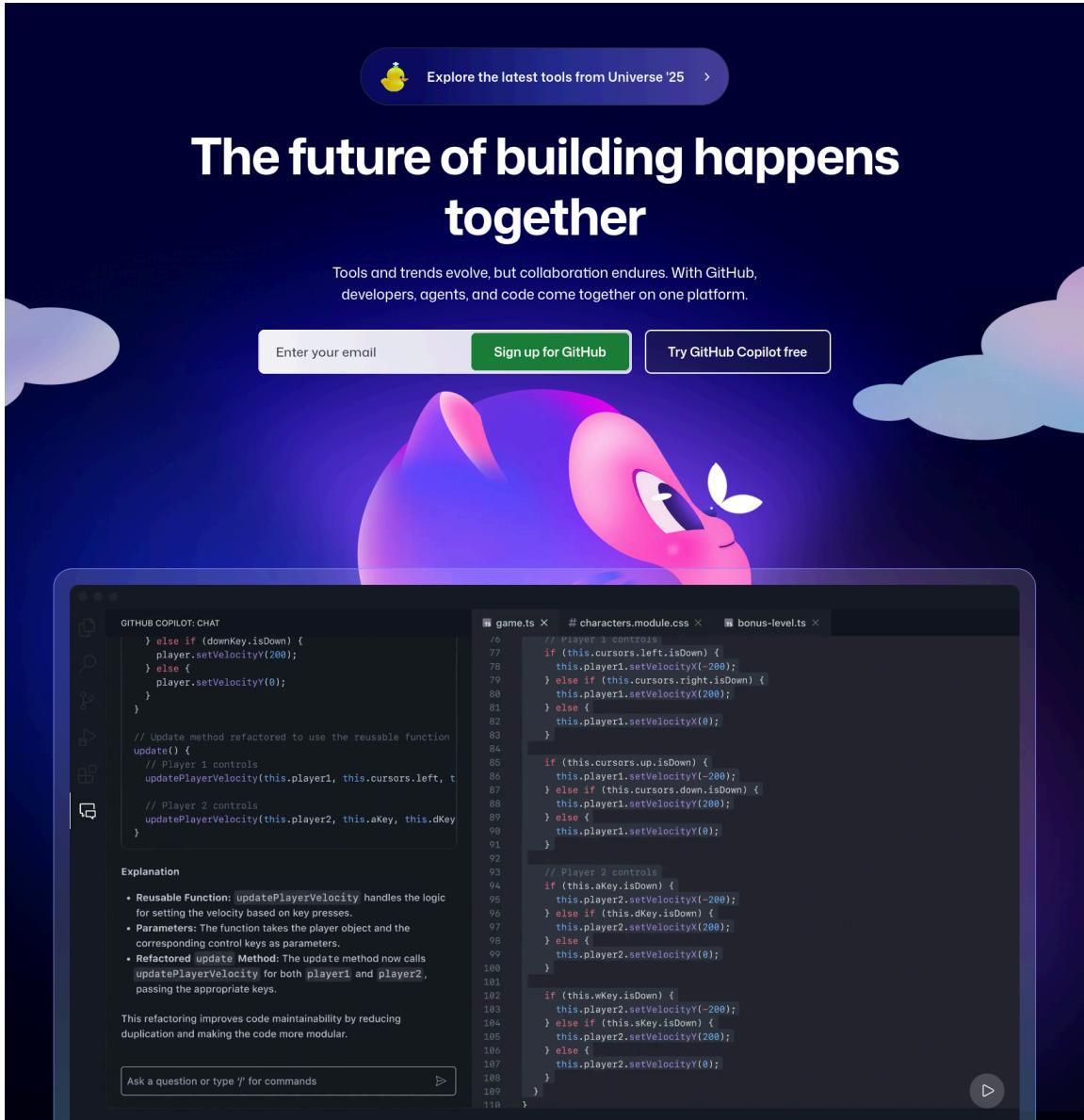
1 Checklist of Everything Needed to Install for AI-Assisted-Coding-In-Python

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

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.

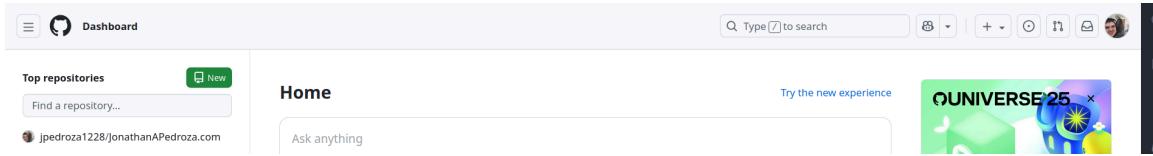


At this page, you will sign up for GitHub with your @berkeley.edu email account. Follow the directions to verify your account. Below are some recommendations for creating a username (inspired by <https://happygitwithr.com/github-acct>).

- 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 (_)

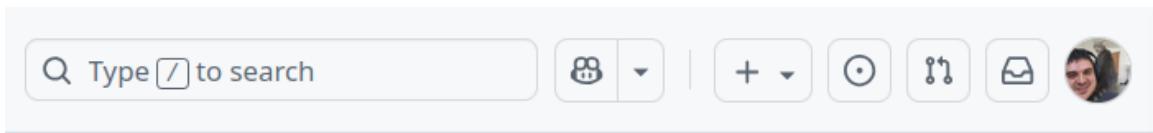
Once you sign in, you will be at your dashboard. You will come back to GitHub shortly to copy the information for the workshop materials.



At the top, you'll see the dropdown menu for your repositories and your profile.



Click on your profile circle and go to your settings.



[Try the new experience](#)



INCLUDE TEXT HERE

 **Jonathan A. Pedroza (JP) (@jpedroza1228)**
Your personal account

[Go to your personal profile](#)

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](#)

Public profile

Name

Jonathan A. Pedroza (JP)

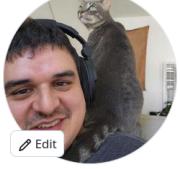
Your name may appear around GitHub where you contribute or are mentioned. You can remove it at any time.

Public email

jpedroza1228@berkeley.edu

[Remove](#)

You can manage verified email addresses in your [email settings](#)

Profile picture


[Edit](#)

Bio

Data Science Education Postdoctoral Scholar.
Prevention Scientist

You can @mention other users and organizations to link to them.

Pronouns

he/him

URL

<https://jonathanapedroza.com/>

Social accounts

 <https://bsky.app/profile/jonathanpedroza.bsky.social>

 <https://www.linkedin.com/in/jonathan-a-pedroza-phd-5721a7120/>

 [Link to social profile 3](#)

 [Link to social profile 4](#)

Company

[Link to company profile](#)

You can @mention your company's GitHub organization to link it.

Location

Berkeley, CA

Display current local time
Other users will see the time difference from their local time.

 You have a connected ORCID ID 0009-0000-5276-0835 for the account @jpedroza1228.

Display your ORCID ID on your GitHub profile

Disconnecting your ORCID ID may affect areas of your profile where your ORCID ID is displayed.

[Disconnect your ORCID ID](#)

INCLUDE TEXT HERE



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

INCLUDE TEXT HERE

The screenshot shows the GitHub Education dashboard for user Jonathan A. Pedroza (JP). At the top, there's a profile picture and the name "Jonathan A. Pedroza (JP) (jpedroza1228)". Below the name, it says "Your personal account". To the right, there's a button "Go to your personal profile". On the left, there's a sidebar with links: "Public profile", "Account", "Appearance", "Accessibility", "Notifications", "Access", and "Billing and licensing". Under "Billing and licensing", there are several sub-links: "Overview", "Usage", "Premium request analytics (New)", "Budgets and alerts", "Licensing", "Payment information", "Payment history", "Additional billing details", and "Education benefits" (which is highlighted with a blue bar). The main content area is titled "GitHub Education" and features a section for "Education Benefits" with a graduation cap icon. It says "Complete a teacher or student application to unlock tools and resources for your educational journey." There's also a green button "Start an application".

INCLUDE TEXT HERE



Education Benefits Application

X

Select your role in education: *

Teacher

Student



You have verified the email address on your GitHub account.
That academic domain is associated with the school **University of California, Berkeley**.

Select this school

What is the name of your school? *



If your school is not listed, then enter the full school name and continue. You will be asked to provide further information about your school on the next page. A minimum of two characters is required to find your school.

What is your school email address? *

jpedroza1228@berkeley.edu



Have a different email address you use with your school? [Add it here.](#)

[Privacy Policy](#)

[Share Location](#)

[Continue](#)

INCLUDE TEXT HERE



Education Benefits Application

X

Select your role in education: *

Teacher

Student



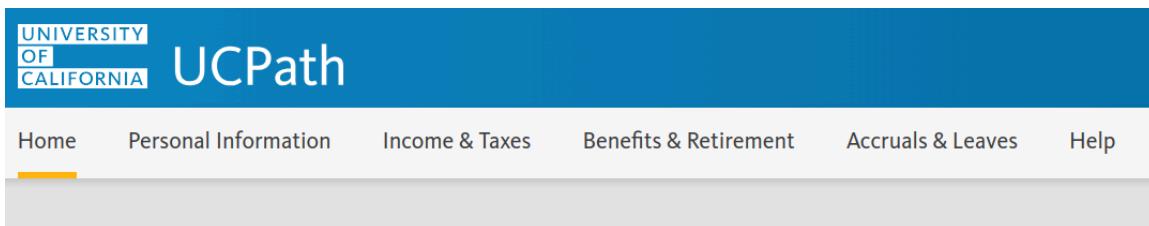
You have verified the email address on your GitHub account.
That academic domain is associated with the school **University of California, Berkeley**.

[Unselect this school](#)

✓ Location shared

[Continue](#)

INCLUDE TEXT HERE



The screenshot shows the UCPath website interface. At the top, there's a blue header bar with the "UNIVERSITY OF CALIFORNIA" logo on the left and "UCPath" in large white letters on the right. Below the header is a navigation menu with links: Home, Personal Information, Income & Taxes, Benefits & Retirement, Accruals & Leaves, and Help. The "Home" link is underlined with a yellow bar.

INCLUDE TEXT HERE

INCLUDE TEXT HERE

For your convenience, the University of California (UC) provides a simple method for employment verification.

If you are applying for a loan, an apartment or job, your employment verifier (e.g. bank, leasing agent, or employer) accesses your employment information through The Work Number website.

Please Note

Employees who opted out from sending their information to The Work Number must contact UCPATH for assistance.

The Work Number

The Work Number is a third-party provider of employment and income verification. All verifiers (banks, employers or leasing agents) must access your information through its website.

How to Provide Proof of Your Employment and Income

Please provide your employment verifier the following information:

- Inform them that UC uses the [The Work Number](#)
- Provide them the University of California Employer Code: 15975
- Provide them your Social Security Number

Employment verification summary for employees only

If you simply need your employment information for your records, you may download a summary below.

Employees who Opted Out

Employees who opted out from sending their information to The Work Number must refer verifiers to UCPATH to complete employment & income verifications. Employment & income verifications for this population must be completed manually by UCPATH.

Verifiers may contact UCPATH via

- Email: ucpath@universityofcalifornia.edu
- Phone: 1-855-982-7284 (Monday through Friday, 8 a.m.– 5 p.m.)
- Fax: 1-855-982-2329

UCPATH data at The Work Number

UCPATH data sent to The Work Number excludes Employees who opted out from sending their information. For all other populations, UCPATH demographic data such as job record information is submitted daily and income information is submitted after each pay date.

[Generate Summary Report](#)

INCLUDE TEXT HERE

GitHub Education

Education Benefits

You have a current pending application. See below for more details.

Pending

Submitted less than a minute ago

Start an application

INCLUDE TEXT HERE

GitHub Education

The screenshot shows a GitHub Education application status page. At the top, there's a section for 'Education Benefits' with a graduation cap icon, a 'Start an application' button, and a progress bar indicating the application is 'Approved'. Below this, a green header bar shows the approval date ('Approved on November 03, 2025') and the application type ('Application Type: Faculty'). The main content area contains several messages: 'Your academic status has been verified. Congratulations!', 'Your academic benefits will become available after 72 hours of your verification.', 'Once the benefits become available, you will be able to upgrade your organizations to GitHub Teams via your [GitHub Education dashboard](#).', 'To redeem your Copilot Pro coupon, please sign up via this [link](#).', 'We hope you enjoy your GitHub Education benefits.', and 'Your benefits will expire on **November 03, 2027**'.

INCLUDE TEXT HERE

The form allows users to manage their email addresses. It includes fields for adding a new email address, setting a primary email address, adding a backup email address, and keeping email addresses private. The 'Primary email address' field is set to 'jpedroza1228@berkeley.edu'. The 'Backup email address' field is set to 'jonpedroza1228@gmail.com'. The 'Keep my email addresses private' toggle is turned off.

Add email address *

Email address Add

Primary email address

Select an email to be used for account-related notifications and can be used for password reset.

jpedroza1228@berkeley.edu

Backup email address

Your backup GitHub email address will be used as an additional destination for security-relevant account notifications and can also be used for password resets.

jonpedroza1228@gmail.com

Keep my email addresses private

We'll remove your public profile email and use `43424923+jpedroza1228@users.noreply.github.com` when performing web-based Git operations (e.g. edits and merges) and sending email on your behalf. If you want command line Git operations to use your private email you must [set your email in Git](#).

Previously authored commits associated with a public email will remain public.

Off

2.1 Adding Additional Email Addresses

2.1.a Optional Two-Factor Authentication (2FA)

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

 **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 
-  Packages
-  Copilot 
-  Pages
-  Saved replies

Security

-  Code security

Integrations

-  Applications
-  Scheduled reminders

Archives

-  Security log
-  Sponsorship log

 Developer settings

Two-factor authentication

...

Two-factor authentication adds an additional layer of security to your account by requiring more than just a password to sign in. [Learn more about two-factor authentication.](#)

Preferred 2FA method

Set your preferred method to use for two-factor authentication when signing into GitHub.

Authenticator app 

Two-factor methods

Authenticator app Configured

Use an authentication app or browser extension to get two-factor authentication codes when prompted.

[Edit](#)

SMS/Text message Less secure

Get one-time codes sent to your phone via SMS to complete authentication requests. We strongly advise against using SMS because it is susceptible to interception, does not provide resistance against phishing attacks, and deliverability can be unreliable. It is recommended to use an Authenticator app instead of SMS.

[Add](#)

Security keys

Security keys are webauthn credentials that can only be used as a second factor of authentication.

[Edit](#)

GitHub Mobile

GitHub Mobile can be used for two-factor authentication by installing the GitHub Mobile app and signing in to your account.

[Add](#)

Recovery options

 Your two-factor authentication recovery codes have not been downloaded or printed in the last one year. Make sure your recovery codes are up-to-date by viewing and downloading or printing them again.

Recovery codes Viewed

Recovery codes can be used to access your account in the event you lose access to your device and cannot receive two-factor authentication codes.

[View](#)

2.2 GitHub TLDR

1. Go to [GitHub](#) to sign up
2. Create account (use @berkeley.edu email)
3. Create a good username
4. Add additional email addresses
5. Set up 2FA

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

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 Copilot Pro (Instructions for Verification)

To get Copilot Pro for teachers and students (for free) you need to [apply here](#)

3.2 Free Version

The free version of GitHub Copilot comes with VSCode. You can install the necessary extensions in the following section.

3.3 Copilot TLDR

1. Sign up for Copilot Pro using student/teacher verification, paying a monthly fee, or using the free version.

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

4 Installing Python

To install Python, you will go to [The Comprehensive 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
- Install Visual Studio (VS) Code
- Install Python
- Sign up for GitHub Copilot
- Copy GitHub repository/Download Zip file
- Adjust VSCode to work with Python

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

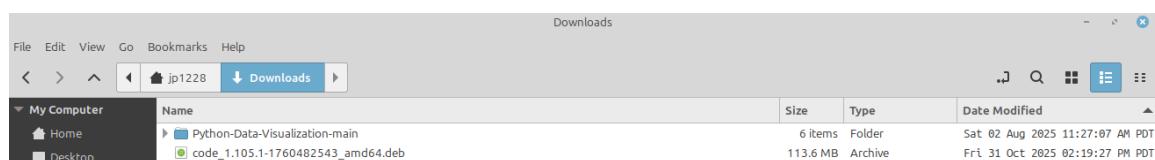
Note: Everything below shows installation using Linux Mint.

```
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
```

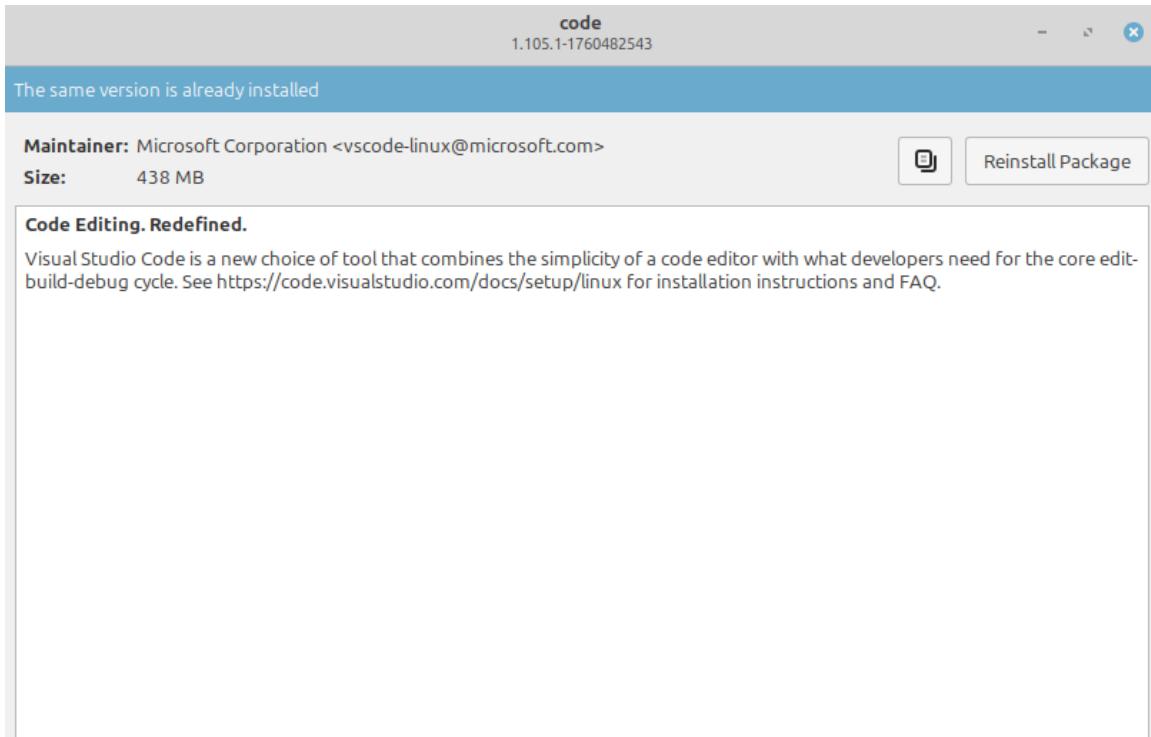
Link: <https://code.visualstudio.com/docs/setup/linux>

Option 1: Use the Link

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

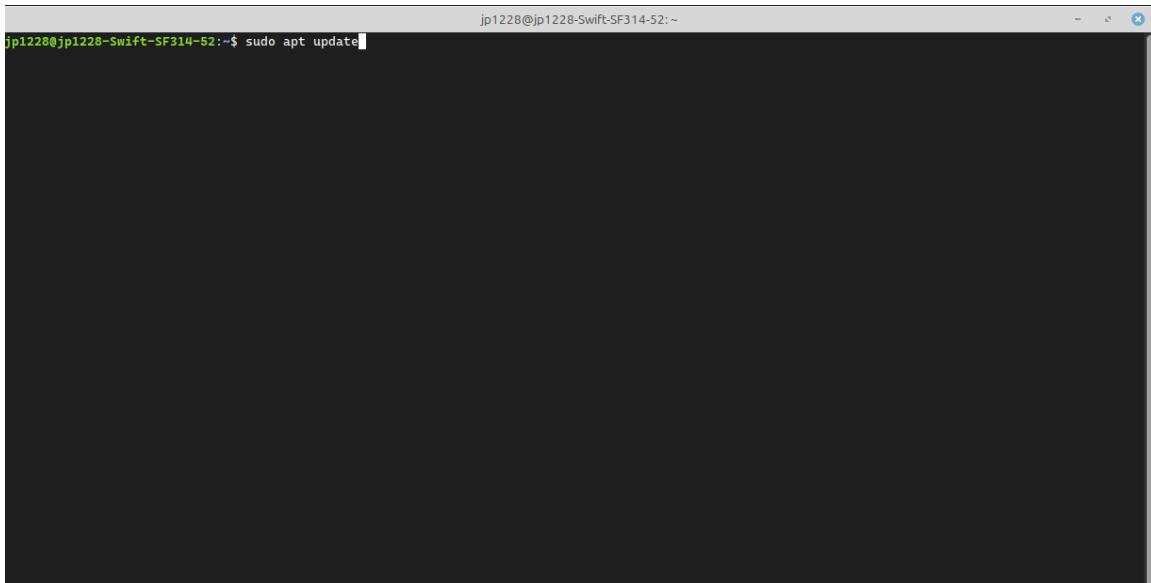


2. Click Install Package to start the install.



Option 2: Download Using Terminal

1. Update your programs.



- 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
```

```
jp1228@jp1228-Swift-SF314-52: ~
Hit:4 https://cloud.r-project.org/bin/linux/ubuntu noble-cran0/ InRelease
Hit:5 https://download.docker.com/linux/ubuntu noble InRelease
Hit:6 https://packages.microsoft.com/repos/code_stable InRelease
Get:7 https://s3.amazonaws.com/repo.deb.cyberduck.io stable InRelease [3,245 B]
Hit:8 http://archive.ubuntu.com/ubuntu noble InRelease
Get:9 http://archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:10 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Hit:11 https://packages.cloud.google.com/apt cloud-sdk InRelease
Hit:2 https://repository.spotify.com stable InRelease
Get:13 http://archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Err:7 https://s3.amazonaws.com/repo.deb.cyberduck.io stable InRelease
  The following signatures couldn't be verified because the public key is not available: NO_PUBKEY FE7097963FEFBET2
Get:14 http://archive.ubuntu.com/ubuntu noble-updates/main i386 Packages [545 kB]
Get:15 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [1,573 kB]
Get:16 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 Components [175 kB]
Get:17 http://archive.ubuntu.com/ubuntu noble-updates/restricted amd64 Components [212 B]
Get:18 http://archive.ubuntu.com/ubuntu noble-updates/universe amd64 Packages [1,498 kB]
Get:19 http://archive.ubuntu.com/ubuntu noble-updates/universe i386 Packages [988 kB]
Get:20 http://archive.ubuntu.com/ubuntu noble-updates/universe amd64 Components [378 kB]
Get:21 http://archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 Components [948 B]
Get:22 http://archive.ubuntu.com/ubuntu noble-backports/main amd64 Components [7,144 B]
Get:23 http://archive.ubuntu.com/ubuntu noble-backports/restricted amd64 Components [216 B]
Get:24 http://archive.ubuntu.com/ubuntu noble-backports/universe amd64 Components [11.0 kB]
Get:25 http://archive.ubuntu.com/ubuntu noble-backports/multiverse amd64 Components [212 B]
Get:26 http://security.ubuntu.com/ubuntu noble-security/main amd64 Components [21.5 kB]
Get:27 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 Components [212 B]
Get:28 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Components [52.3 kB]
Get:29 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Components [208 B]
Reading package lists... Done
W: GPG error: https://s3.amazonaws.com/repo.deb.cyberduck.io stable InRelease: The following signatures couldn't be verified because the public key is not available: NO_PUBKEY FE7097963FEFBET2
E: The repository 'https://s3.amazonaws.com/repo.deb.cyberduck.io stable InRelease' is not signed.
N: Updating from such a repository can't be done securely, and is therefore disabled by default.
N: See apt-secure(8) manpage for repository creation and user configuration details.
in1228@in1228-Swift-SF314-52:~$ sudo apt install ./code_1.105.1-1760482543_amd64.deb
```

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.

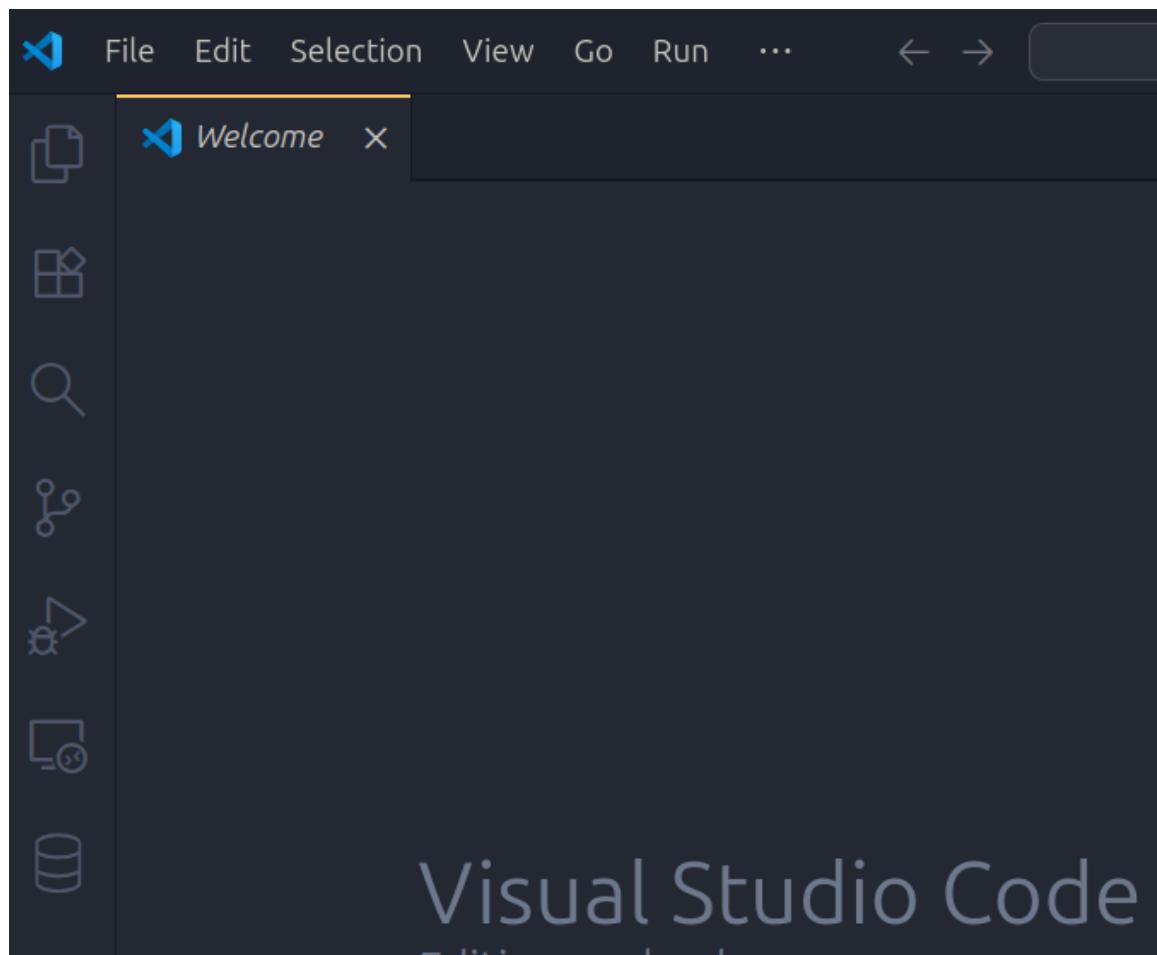
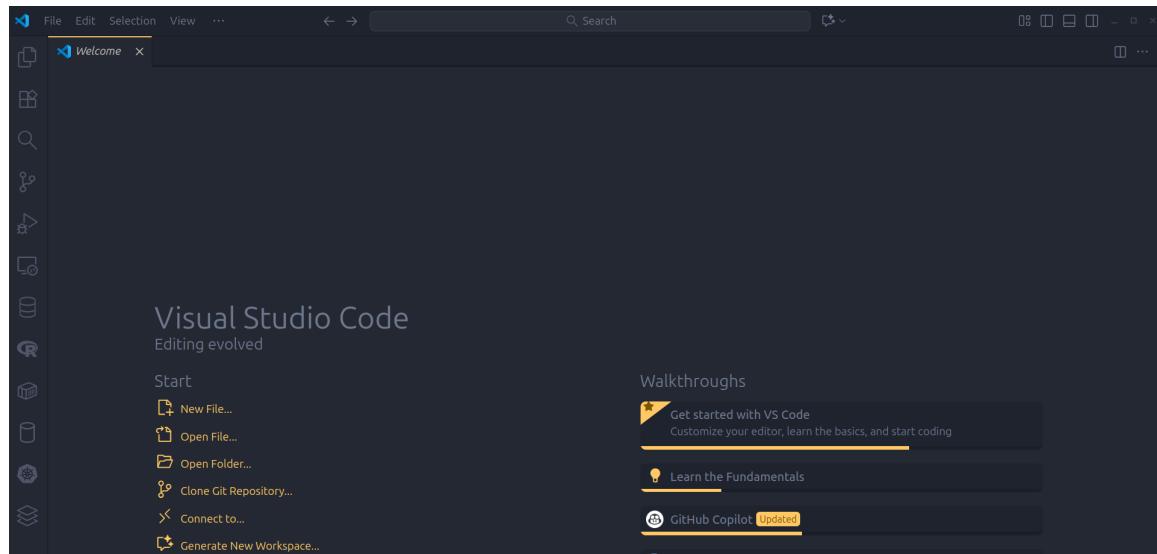
-
- Install GitHub
 - Install Visual Studio (VS) Code
 - Install Python
 - Sign up for GitHub Copilot
 - Copy GitHub repository/Download Zip file
 - Adjust VSCode to work with Python

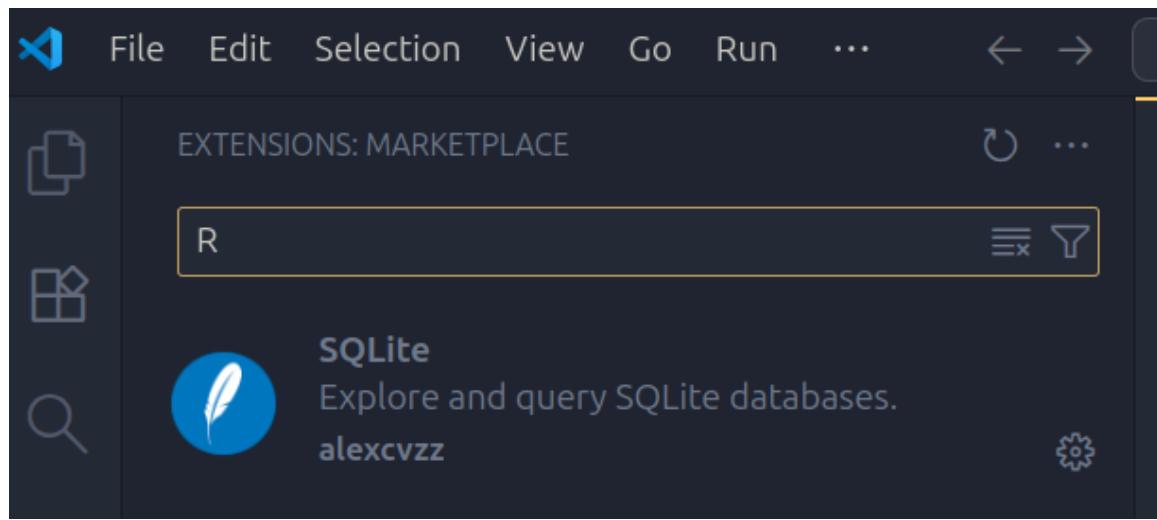
6 Setting up Python in VSCode

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.

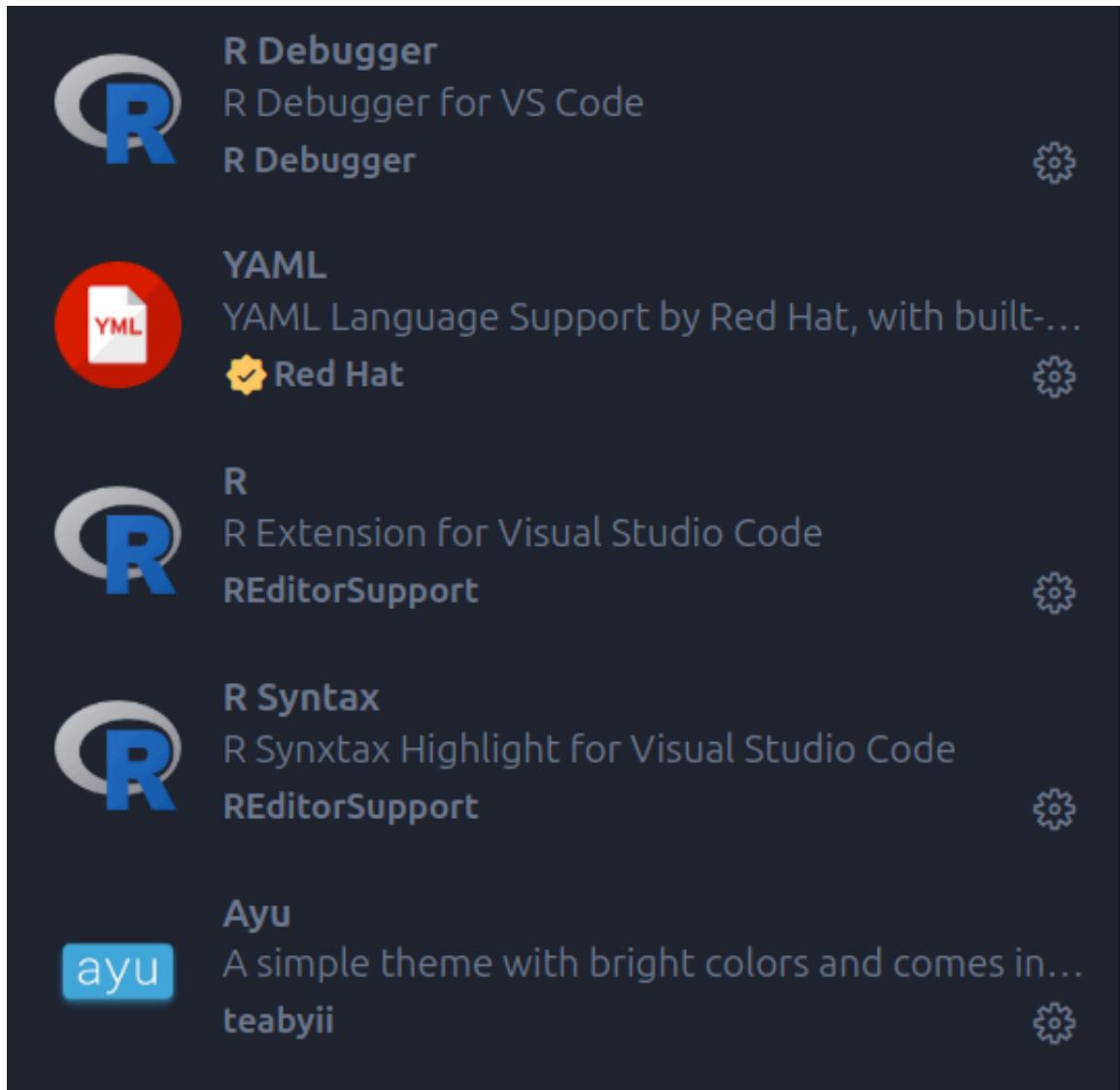
Depending on the version of Python you installed, there may be some issues in incorporating some of the packages. I will include alternatives to try and make sure everything works.

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.





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.

R
REditorSupport | ⚡ 2,867,998 | ★★★★★(43)
R Extension for Visual Studio Code
[Disable](#) [Uninstall](#) Auto Update

[Live Share](#) [Install](#)

This VS Code extension provides support for the R programming language, including features such as R language service based on code analysis, interacting with R terminals, viewing data, plots, workspace variables, help pages, managing packages, and working with R Markdown documents.

The R and R Markdown syntaxes are located in a sibling package `vscode-r-syntax`.

Go to the [wiki](#) to view the documentation of the extension.

Getting started

1. Install R (>= 3.4.0) on your system. For Windows users, Writing R Path to the registry is recommended in the installation.
2. Install `languageServer` in R.

```
install.packages("languageServer")
```

3. Install the R extension for VS Code from the VS Code Extension Marketplace or the Open VSX Registry.
4. Create an R file and start coding.

The following software or extensions are recommended to enhance the experience of using R in VS Code:

- `radian`: A modern R console that corrects many limitations of the official R terminal and supports many features such as syntax highlighting and auto-completion.
- `VSCode-R-Debugger`: A VS Code extension to support R debugging capabilities.
- `httpgd`: An R package to provide a graphics device that asynchronously serves SVG graphics via HTTP and WebSockets.

Go to the installation wiki pages ([Windows](#) | [macOS](#) | [Linux](#)) for more detailed instructions.

Features

- Snippets for R and R Markdown.
- R Language Service: Code completion, function signature, symbol highlight, document outline, formatting, definition, diagnostics, references, and more.

Installation

Identifier	reeditorsupport.r
Version	2.9.6
Last Updated	1 month ago
Size	5.4MB

Marketplace

Published	8 years ago
Last Released	5 months ago

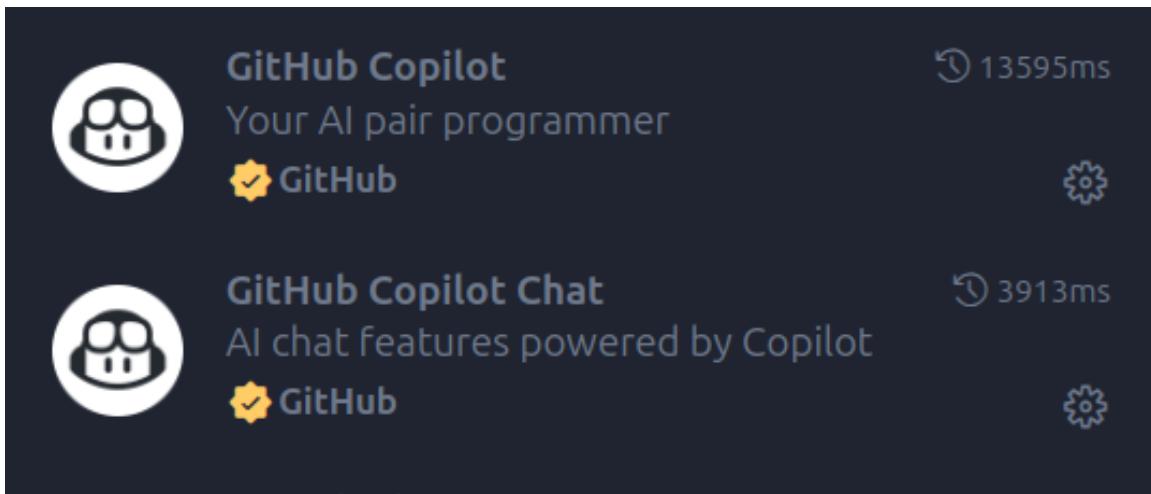
Categories

Programming Languages	Snippets	Other
-----------------------	----------	-------

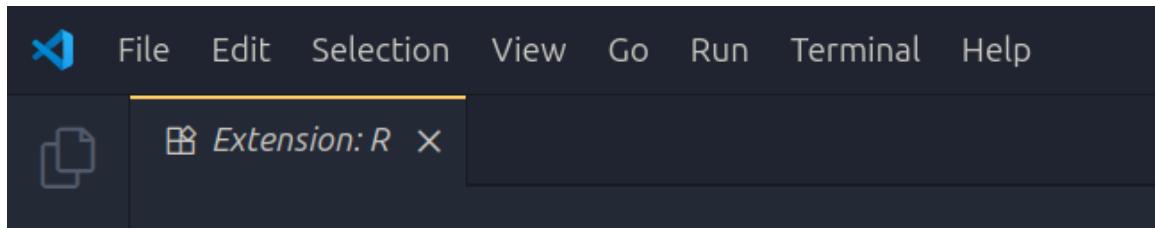
Resources

- [Repository](#)
- [Issues](#)
- [License](#)
- [REditorSupport](#)
- [Marketplace](#)

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.

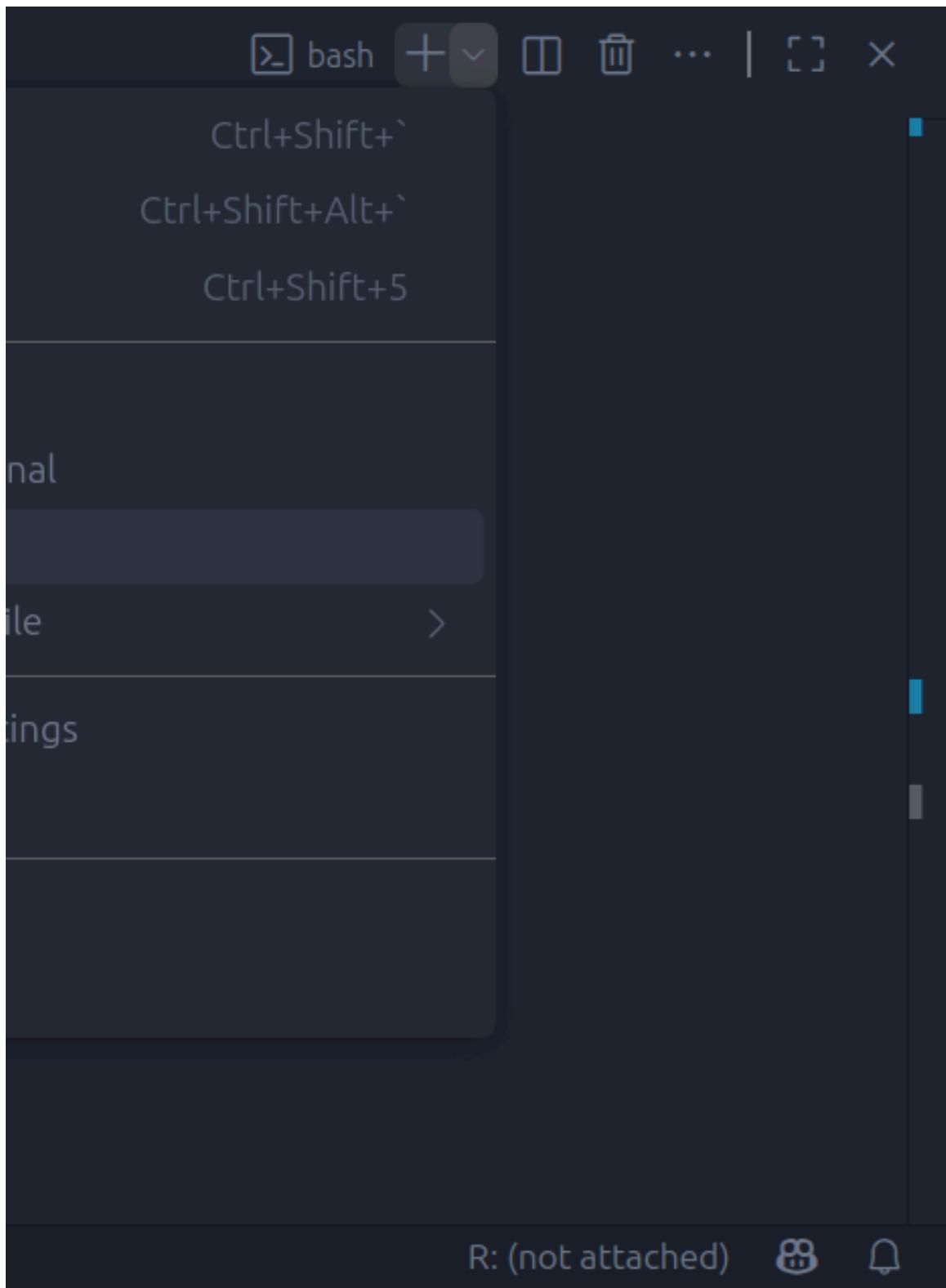
A screenshot of a terminal window. The title bar says "Terminal". The window contains the following text:

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS

● jp1228@jp1228-Swift-SF314-52:~$ R --version
R version 4.5.2 (2025-10-31) -- "[Not] Part in a Rumble"
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 the terms of the
GNU General Public License versions 2 or 3.
For more information about these matters see
https://www.gnu.org/licenses/.

● jp1228@jp1228-Swift-SF314-52:~$ which R
/usr/bin/R
○ jp1228@jp1228-Swift-SF314-52:~$ █
```



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 `httpgd` package, you can [follow the installation instructions here](#).

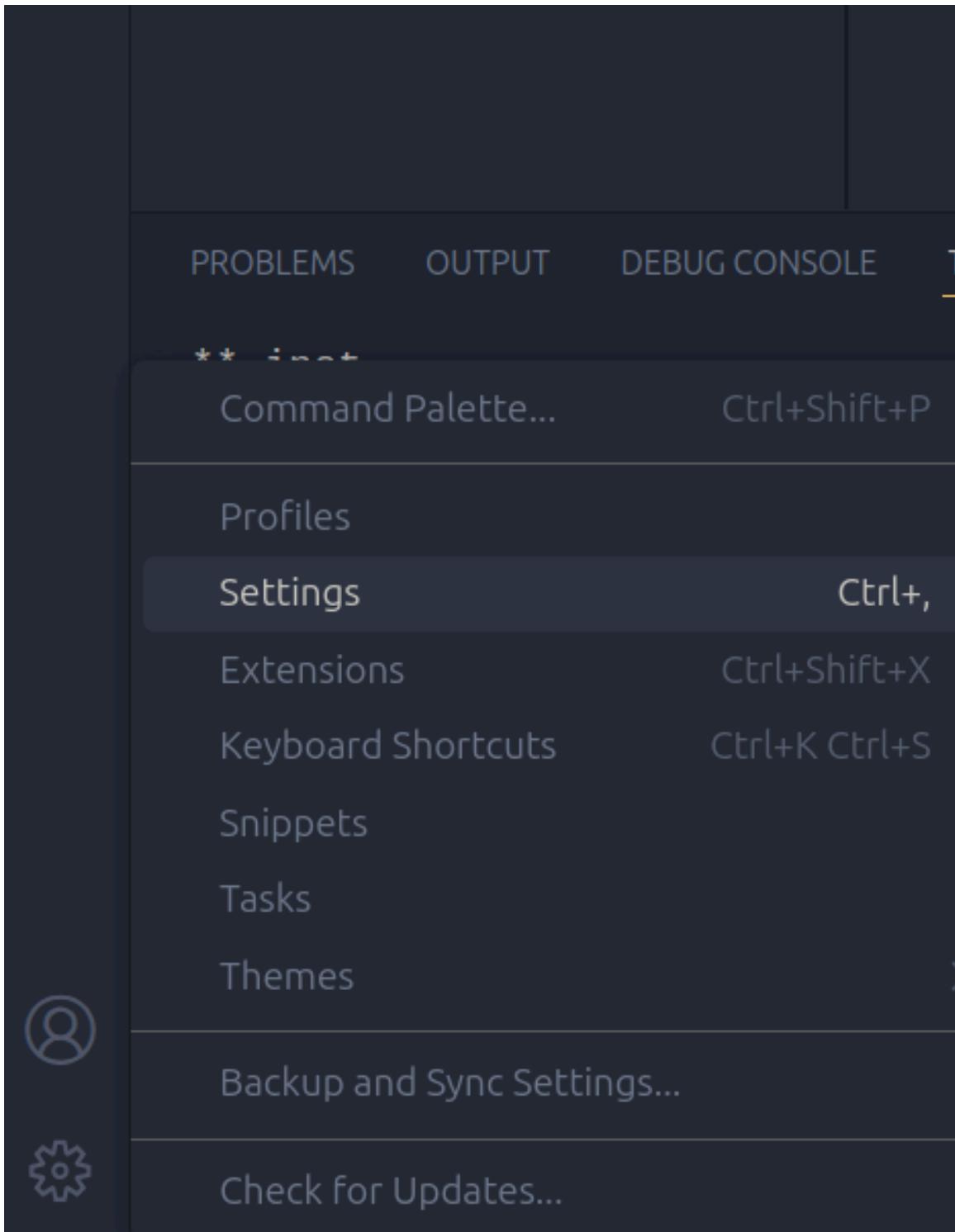
Lastly, I will show some extra customizable settings to make VSCode similar to using RStudio. Below are some settings that you can change in your `settings.json` file. You can also change the settings by going to the gear (see directions below) to make changes to your VSCode.

1. Making Changes by Settings Tab

The screenshot shows a code editor interface with a dark theme. On the left is a sidebar with various icons for file operations like copy, paste, search, and refresh. The main area displays two files side-by-side:

```
home > jp1228 > .config > Code > r.json
1   {
2     "editor.minimap": true,
3     "editor.tabSize": 4,
4     "editor.fontSize": 14,
5     "editor.lineHeight": 1.5,
6     "window.zoomLevel": 100,
7     "python.REPL.sendSelection": true,
8     "python.terminal.integrated.shellArgs": [
9       "python.terminal.integrated.shellPath"
10      ],
11      "python.terminal.integrated.sendSelectedText": true,
12      "workbench.colorscheme": "vscode-dark",
13      "editor.selectionHighlight": true,
14      "editor.occurances": true,
15      "explorer.confirmDelete": false,
16      "editor.quickSuggestions": true,
17      "editor.parameterHints": true,
18      "editor.acceptSuggestionOnEnter": true,
19      "editor.inlineSuggest": true,
20      "database-client": "PostgreSQL",
21      "explorer.fileNames": true,
22      "*.ts": "${
23        ".js": "${
24          "*.jsx": "${
25            "*.tsx": "${
26              "tsconfig.json": true,
27              "package.json": true,
28              "*sqlite": true,
29              "*db": true,
30              "*sqlite3": true,
31              "*db3": true,
32              "*sdb": true,
33              "*s3db": true
34            }",
35          "r.plot.default": true,
36          "r.plot.useHttp": true,
37          "editor.stickySelection": true
38        }"
39      },
40      // {
41      //   "editor.tabSize": 4,
42      //   "editor.fontSize": 14,
43      //   "editor.lineHeight": 1.5,
44      //   "explorer.confirmDelete": false,
45      //   "editor.quickSuggestions": true,
46      //   "editor.parameterHints": true,
47      //   "explorer.fileNames": true,
48      //   "*.ts": true,
49      //   "*.js": true,
50      //   "*.jsx": true,
51      //   "*.tsx": true,
52      //   "tsconfig.json": true,
53      //   "package.json": true,
54      //   "*sqlite": true,
55      //   "*db": true,
56      //   "*sqlite3": true,
57      //   "*db3": true,
58      //   "*sdb": true,
59      //   "*s3db": true
60      // },
61      // "sqliteView": true,
62      // "workbench.editorAssociations": [
63      //   {
64      //     "language": "duckdb"
65      //   }
66    ]
67  }
68
69  PROBLEMS OUTPUT DEBUG
70
71
72  ** inst
73  ** byte-compile and prepare vignettes
74  ** help
75  *** installing help indexes
76  ** building package index
77  ** installing vignettes
78  ** testing if installed package has examples
79  ** testing if installed package has tests
80  ** testing if installed package has vignettes
81  * DONE (remotes)
```

Below the code editor, a terminal window shows the output of an R command. It includes messages about installing help indexes, building package indices, and skipping installs of certain packages.



R. 2217 Settings Found. AI Results Available

User

Quarto > Render: R Package Output Directory

Render output files in a temporary directory, when in an R package.

R: Always Use Active Terminal

Use active terminal for all commands, rather than creating a new R terminal.

R: Bracketed Paste

Use bracketed paste mode when sending code to terminal. Enable for `radian` console.

R > Help Panel: Cache Index Files

Whether/where to store parsed help indices between sessions.

None

R > Help Panel: Click Code Examples

What happens when clicking code examples on help pages. Might require restarting to take effect.

Item	Value
Click	Copy
Ctrl+Click	Run
Shift+Click	Ignore

R > Help Panel: Enable Hover Links

Show links to matching help pages in hover

R > Help Panel: Enable Syntax Highlighting

Enable syntax highlighting in the help panel.

R > Help Panel: Preview Local Packages

Which local directories to try for local help pages previewer. Set to `[]` to disable.

Add Item

R: Lib Paths

Additional library paths to launch R background processes (R languageserver, help server, etc.). These paths will be appended to `.libPaths()` on process startup. It could be useful for projects with `renv` enabled.

Add Item

R > Live Share > Defaults: Command Forward

Default boolean value for guest command forwarding.

R > Live Share > Defaults: Share Browser

Default boolean value for automatically sharing R browser ports with guests.

R > Live Share > Defaults: Share Workspace

Default boolean value for sharing the R workspace with guests.

R > Plot: Use Httpgd

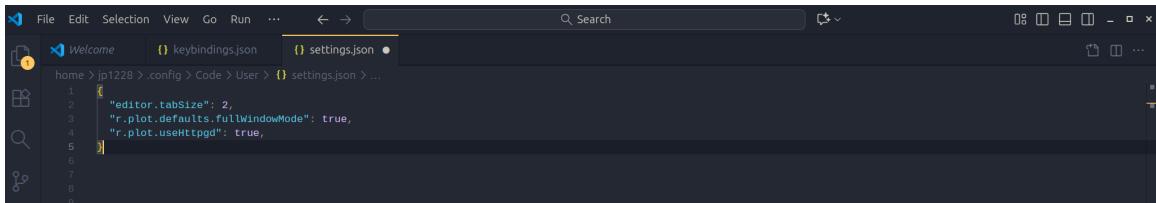
Use the httpgd-based plot viewer instead of the base VSCode-R plot viewer. Changes the option `vsc.use_httpgd` in R.

Requires the `httpgd` R package version 1.2.0 or later.

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.

Type to search in keybindings			
Command	Keybinding	When	Source
Accept Inline Completion	Ctrl + /	accessibleViewIsShown && accessibleViewCurrentProviderId == 'inl...	System
Accept Inline Suggestion	Tab	inlineEditIsVisible && tabShouldAcceptInlineEdit && !editorHover...	System
Accept Inline Suggestion	Tab	inInlineEditsPreviewEditor	System
Accept Next Word Of Inline Suggestion	Ctrl + RightArrow	cursorBeforeGhostText && inlineSuggestionVisible && !accessibl...	System
Accessible Diff Viewer: Go to Next Difference	F7	isInDiffEditor	System
Accessible Diff Viewer: Go to Previous Difference	Shift + F7	isInDiffEditor	System
Add Cursor Above	Ctrl + Shift + UpArrow	editorTextFocus	System
Add Cursor Below	Shift + Alt + UpArrow	editorTextFocus	System
Add Cursor Below	Ctrl + Shift + DownArrow	editorTextFocus	System
Add Cursors to Line Ends	Shift + Alt + I	editorTextFocus	System
Add Line Comment	Ctrl + K Ctrl + C	editorTextFocus && !editor_READONLY	System
Add Selection to Next Find Match	Ctrl + D	editorFocus	System
Auto Fix...	Shift + Alt + <	textInputFocus && !editor_READONLY && supportedCodeAction =~ /(\\s...)	System
Auto Fix...	Shift + Alt + .	textInputFocus && !editor_READONLY && supportedCodeAction =~ /(\\s...)	System
C/C++: Switch Header/Source	Alt + O	editorTextFocus && editorLangId =~ /^(c cuda-)?cpp\$/ && !conf...	C/C++
Calls: Show Call Hierarchy	Shift + Alt + H	editorHasCallHierarchyProvider	Reference Search View
Cancel Selection Anchor	Escape	editorTextFocus && selectionAnchorSet	System
Change All Occurrences	Ctrl + F2	editorTextFocus && !editor_READONLY	System
Change Language Mode	Ctrl + K M	!notebookEditorFocused	System
Chat: Apply in Editor	Ctrl + Enter	accessibleViewInCodeBlock && chatisEnabled chatisEnabled && i...	System
Chat: Attach Instructions...	Ctrl + Alt + /	chatisEnabled && config.chat.promptFiles	System
Chat: Debug Last Terminal Command	Ctrl + Alt + .	github.copilot-chat.activated && terminalFocus && terminalShellI...	GitHub Copilot Chat
Chat: Focus Chat Confirmation	Ctrl + Shift + A	accessibilityModeEnabled && chatisEnabled	System
Chat: Go to Next Chat Edit	Alt + F5	chatEdits.hasEditorModifications && chatisEnabled && editorFocus...	System
Chat: Go to Previous Chat Edit	Shift + Alt + F5	chatEdits.hasEditorModifications && chatisEnabled && editorFocus...	System
Chat: Insert At Cursor	Ctrl + Enter	accessibleViewInCodeBlock && chatisEnabled chatisEnabled && i...	System
Chat: Insert into Terminal	Ctrl + Alt + Enter	accessibleViewInCodeBlock && chatisEnabled chatisEnabled && i...	System
Chat: Keep All Chat Edits	Ctrl + Alt + Y	chatEdits.hasEditorModifications && editorFocus && !chatEdits.is...	System
Chat: Keep Chat Edits	Ctrl + Shift + Y	chatEdits.hasEditorModifications && editorFocus && !chatEdits.is...	System
Chat: Keep this Change	Ctrl + Y	chatEdits.hasEditorModifications && editorFocus && !chatEdits.is...	System
Chat: New Chat	Ctrl + L	chatisEnabled && inChat	System
Chat: New Chat	Ctrl + N	chatisEnabled && inChat	System
Chat: New Chat Editor	Ctrl + N	chatisEnabled && inChat && inChatEditor	System
Chat: Next Code Block	Ctrl + Alt + PageDown	chatisEnabled && inChat	System

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

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](#).

Screenshot of a GitHub repository page for "AI-Assisted-Coding-In-R" (Public) generated from [dlab-berkeley/D-Lab-Workshop-Template](#).

The repository has 2 branches and 0 tags. The main branch contains files: .devcontainer, data, images, lessons, solutions, and .gitignore. The README.md file was updated by tomvannuenen.

The "Clone" section shows options for HTTPS, SSH, and GitHub CLI, with the URL <https://github.com/dlab-berkeley/AI-Assisted-Coding-In-R>. There is also a "Download ZIP" button.

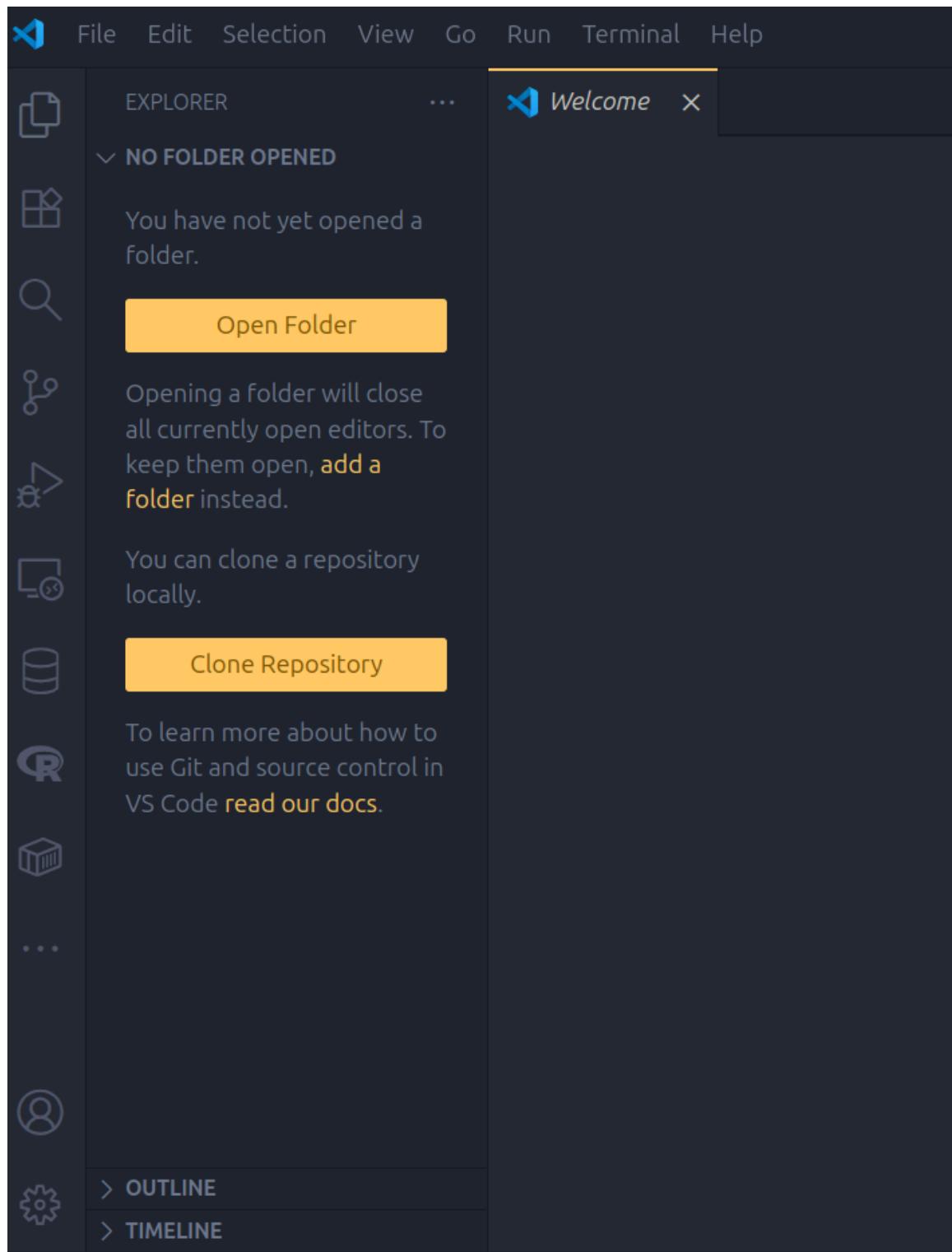
The "About" section describes D-Lab's 2-hour workshop on AI-assisted coding in Visual Studio Code using GitHub Copilot and R. It includes links to Readme, Activity, Custom properties, and Report repository, with 13 stars, 2 watching, and 8 forks.

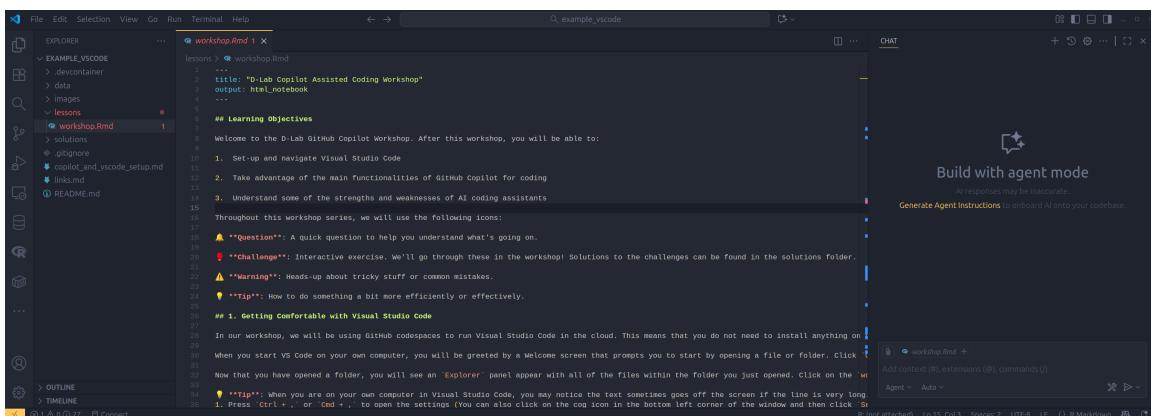
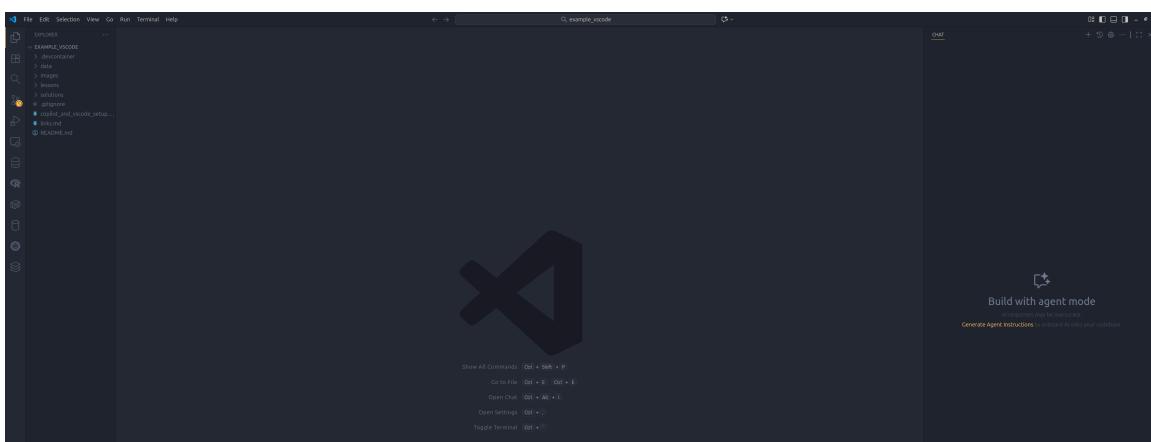
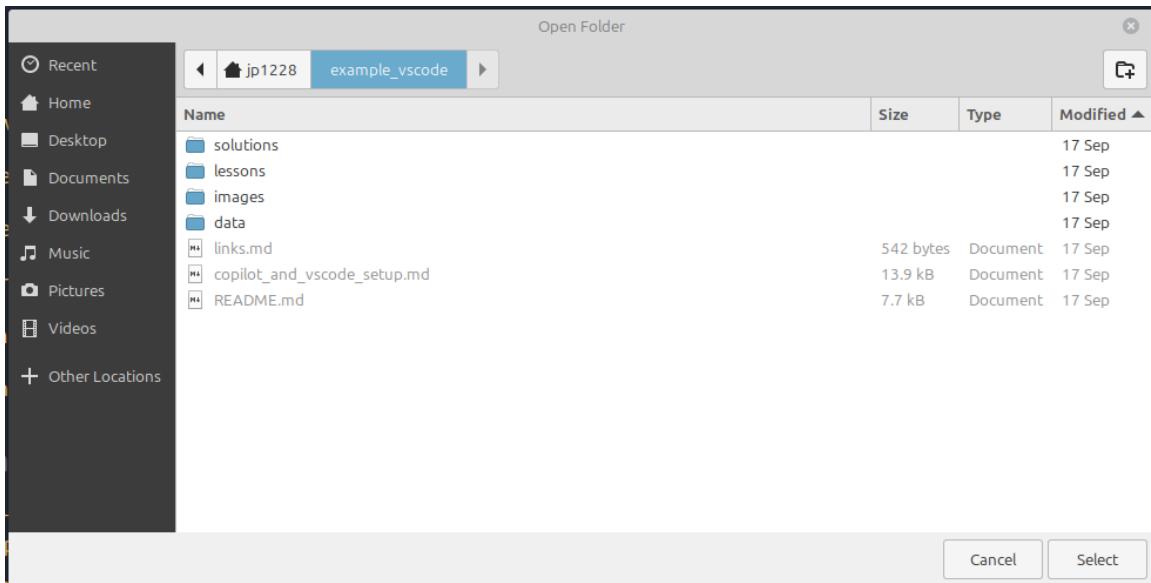
A detailed file listing for the "example_vscode" folder is shown:

Name	Size	Type
Desktop	8 items	Folder
Documents	13 items	Folder
Downloads	310 items	Folder
example_vscode	7 items	Folder
Music	0 items	Folder
Pictures	1 item	Folder

Another file listing for the "example_vscode" folder is shown:

Name	Size	Type
data	2 items	Folder
images	6 items	Folder
lessons	1 item	Folder
solutions	1 item	Folder
copilot_and_vscode_setup.md	13.9 kB	Document
links.md	542 bytes	Document
README.md	7.7 kB	Document





- Install GitHub
- Install Visual Studio (VS) Code

- Install Python
- Sign up for GitHub Copilot
- Copy GitHub repository/Download Zip file
- Adjust VSCode to work with Python

8 Python Specifics

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
 - miniconda has conda and Python, Anaconda has Python, conda, pycharm (different IDE; similar to VSCode), and other programs

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 interpreter

Go to the command palette (Windows/Linux: 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](#).