

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

2025-11-04

Table of contents

1 Checklist of Everything Needed to Install for AI-Assisted-Coding-In-R	1
2 Installing GitHub	2
2.1 Optional: Two-Factor Authentication (2FA)	3
3 Sign Up for GitHub Copilot	8
3.1 Free Version	8
3.2 Copilot Pro (Instructions for Verification)	8
3.3 Section on Getting Copilot Set Up (NEEDS WORK -> WAITING ON ACCESS)	16
4 Installing R	17
4.1 Mac	17
4.2 Windows	17
4.3 Linux	18
5 Install VSCode	18
5.1.a Mac	19
5.1.b Windows	19
5.1.c Linux	19
5.2 VSCode Documentation	21
6 Setting up R in VSCode	21
7 Download Zip File	33

If you are having any issues with installing anything for the workshop, think about [submitting a consulting request](#) where D-Lab consultants can help you get prepared for your workshop.

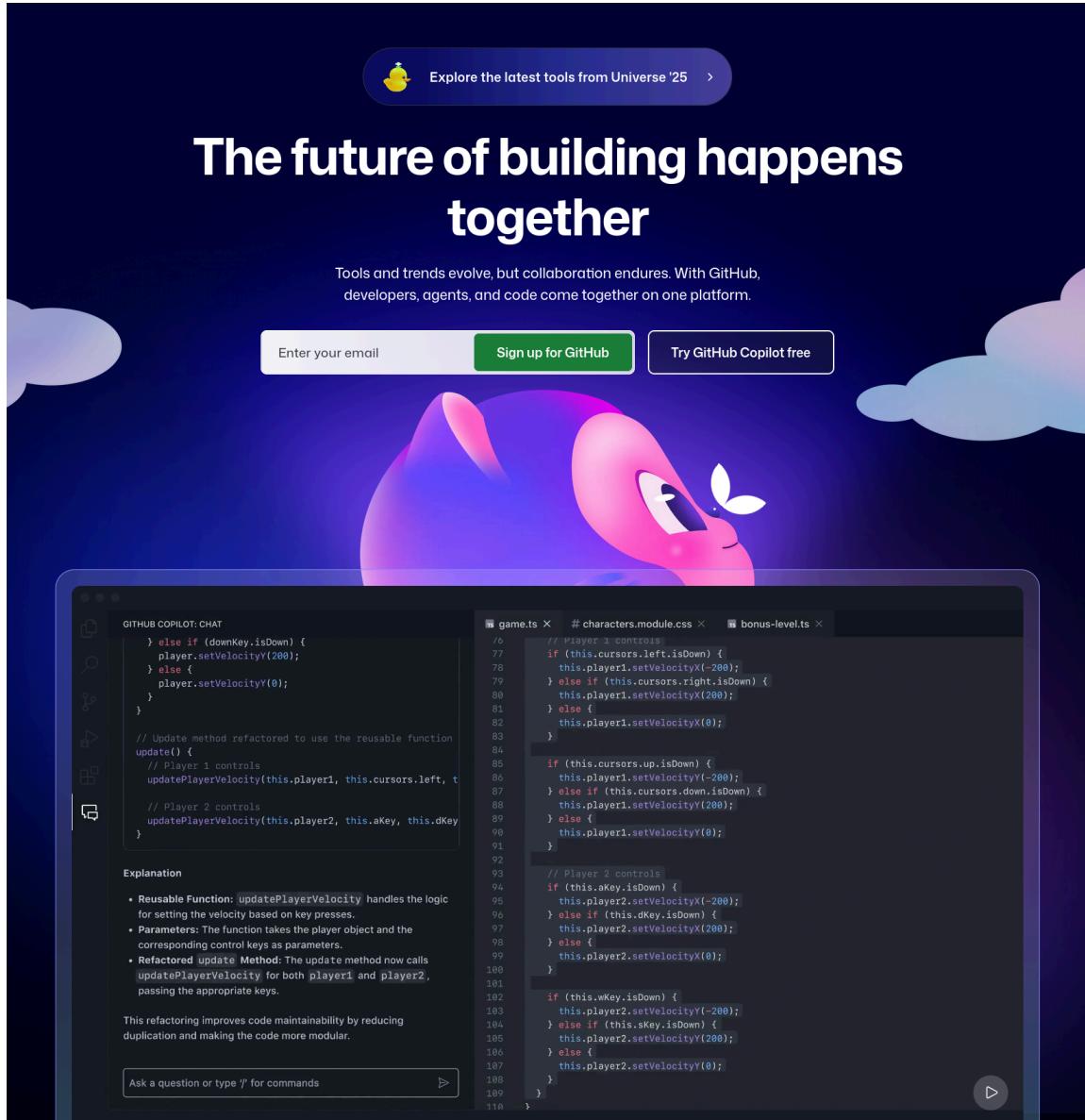
1 Checklist of Everything Needed to Install for AI-Assisted-Coding-In-R

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

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

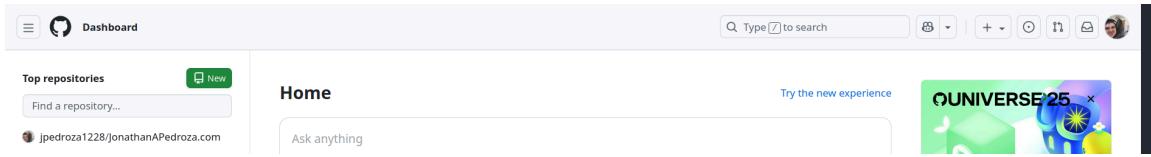
2 Installing GitHub

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 (_)

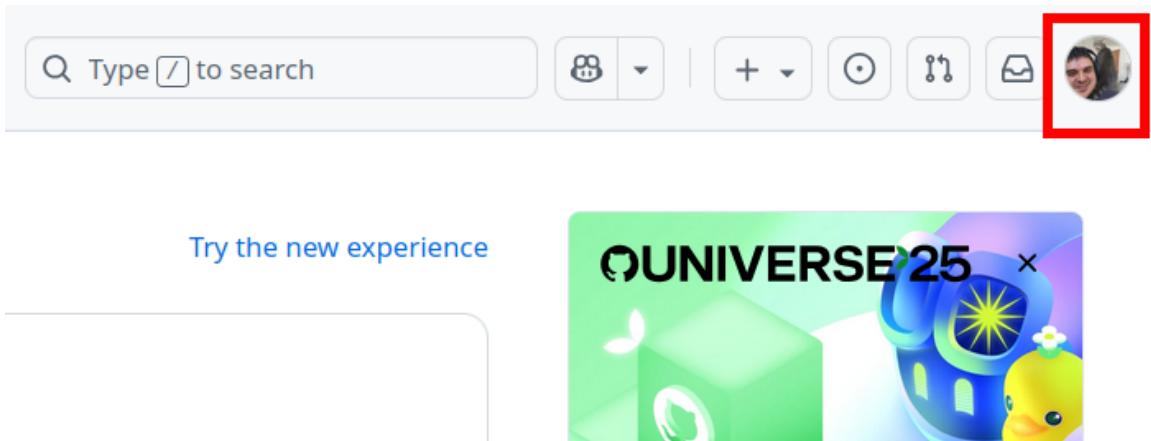


Once you sign in, you will be at your dashboard. You have now downloaded GitHub! Congrats!

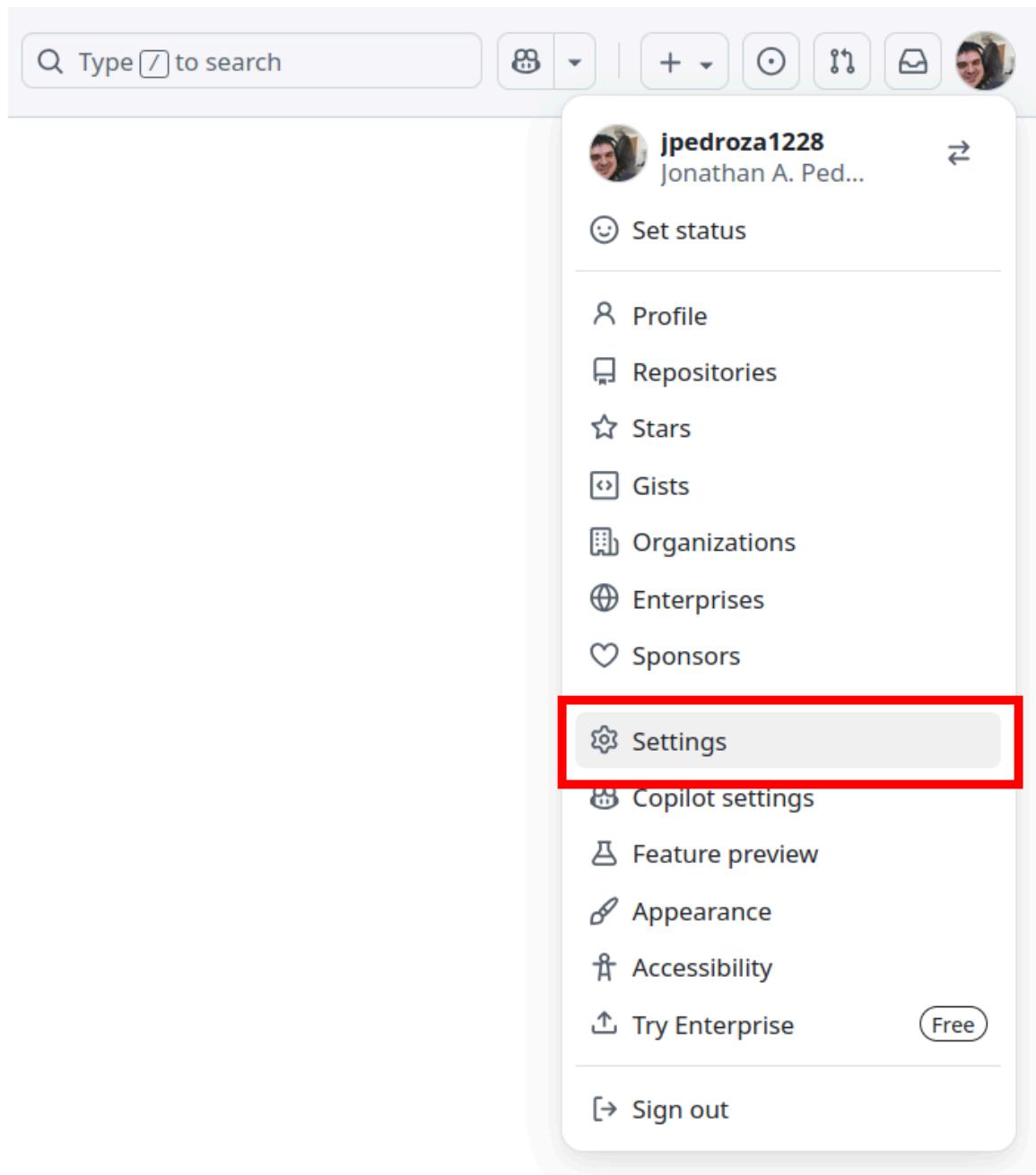
2.1 Optional: Two-Factor Authentication (2FA)

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 go 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**.

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

[Go to your personal profile](#)

Public profile

-  [Account](#)
-  [Appearance](#)
-  [Accessibility](#)
-  [Notifications](#)

Access

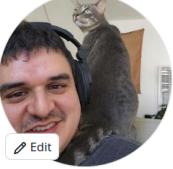
-  [Billing and licensing](#)
-  [Emails](#)
-  [Password and authentication](#)
-  [Sessions](#)
-  [SSH and GPG keys](#)
-  [Organizations](#)
-  [Enterprises](#)
-  [Moderation](#)

Public profile

Name

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

Profile picture



 [Edit](#)

Public email

 Remove

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

Bio

Pronouns

URL

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

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

Location

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

ORCID ID

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](#)

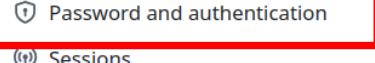
At your settings, there will be a lot of information.

5

 **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

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.

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.

Two-factor methods

Authenticator app Configured

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

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.

Security keys

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

GitHub Mobile

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

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.

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 R
 - Install Visual Studio (VS) Code
 - Adjust VSCode to work with R
 - Download Zip file

3 Sign Up for GitHub Copilot

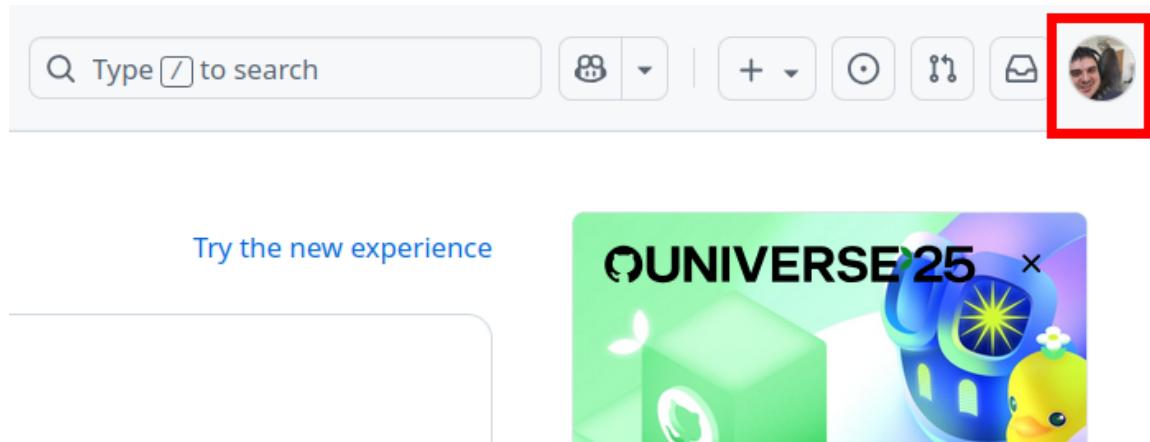
Signing up for GitHub Copilot will depend on whether you plan to sign up for the free version, CoPilot Pro using verified information, or pay for a Pro plan ([see pricing information here](#)).

3.1 Free Version

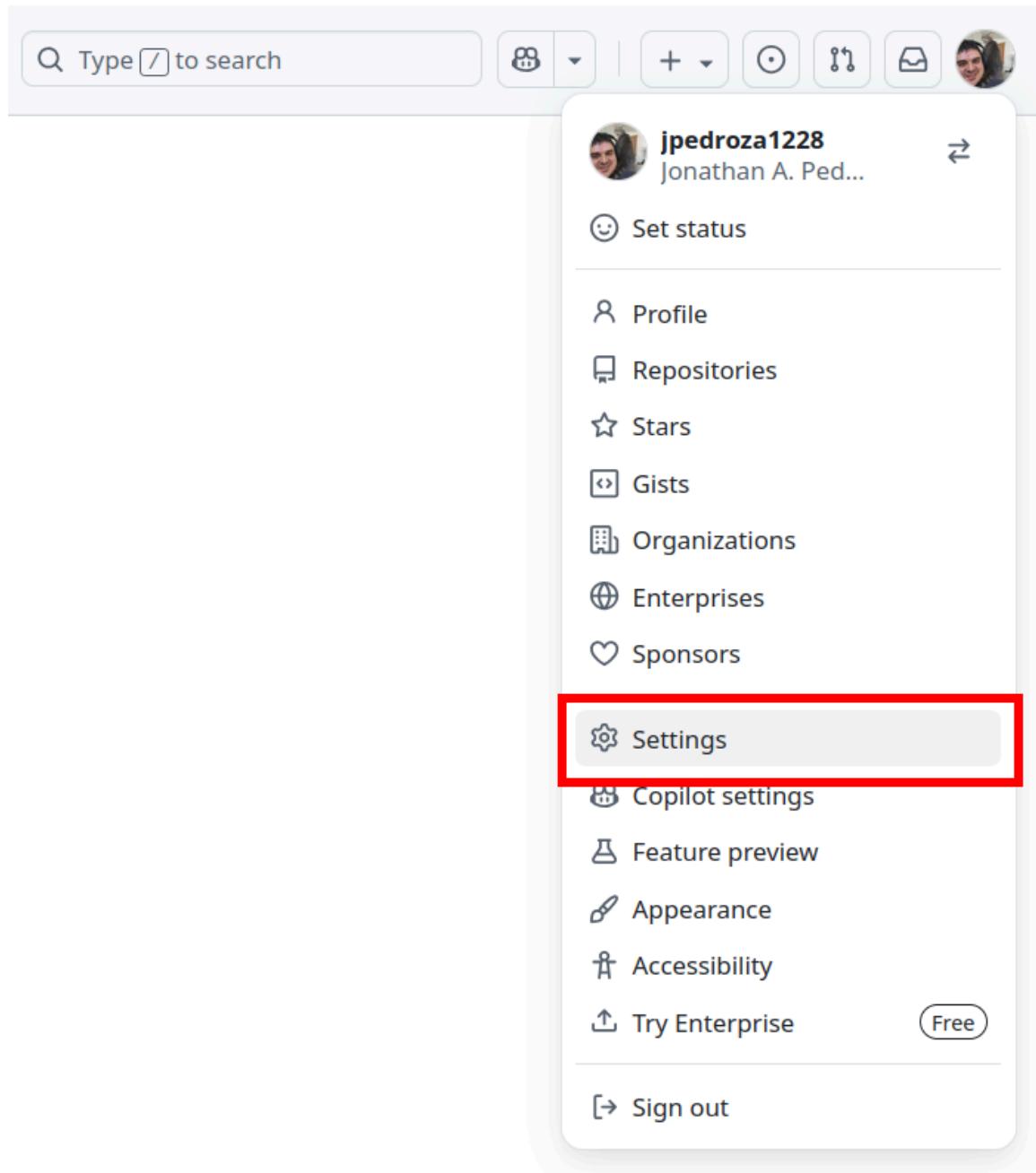
The free version of GitHub Copilot comes with VSCode. You can install the necessary extensions in the following section. You can also try out a [30-day trial here](#).

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 go 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.



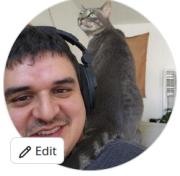
At this dropdown menu, you will go to your **Settings**.

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

[Go to your personal profile](#)

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.

Profile picture


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 email
jpedroza1228@berkeley.edu
You can manage verified email addresses in your [email settings](#)

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
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.

ORCID ID
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.

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

Access

Billing and licensing ^

Overview

Usage

Premium request analytics

New

Budgets and alerts

Licensing

Payment information

Payment history

Additional billing details

Education benefits

Emails

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

The screenshot shows the GitHub profile page for Jonathan A. Pedroza (JP) (jpedroza1228). The top navigation bar includes links for Public profile, Account, Appearance, Accessibility, and Notifications. Below the navigation is a sidebar with sections for Access, Billing and licensing (which is expanded), and Education benefits (which is selected and highlighted with a blue border). The main content area is titled "GitHub Education" and features a "Education Benefits" section with a graduation cap icon, a description: "Complete a teacher or student application to unlock tools and resources for your educational journey.", and a green "start an application" button.

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



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](#)

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.

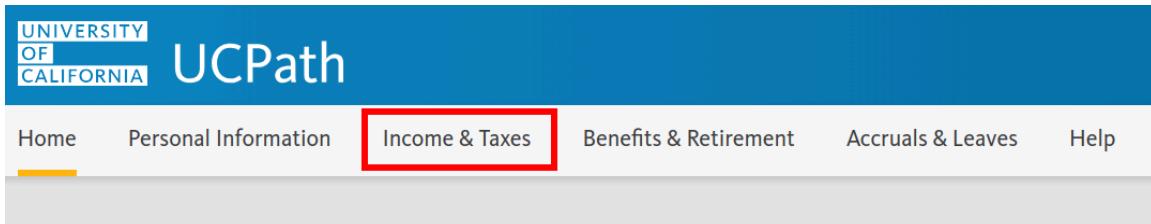
On the application, you can include the full name of the university (University of California, Berkeley) and when given an option to Select a school, click Select this school. You will also include your school email address in the dropdown menu. Make sure you are including your @berkeley.edu email address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

The screenshot shows a web-based application titled "Education Benefits Application". At the top, there is a graduation cap icon and the title. A close button (X) is in the top right corner. Below the title, the text "Select your role in education: *" is displayed. Two radio button options are shown: "Teacher" (selected, indicated by a blue outline) and "Student" (unselected, indicated by a grey outline). A green callout box contains the message: "You have verified the email address on your GitHub account. That academic domain is associated with the school **University of California, Berkeley**". Below this message is a button labeled "Unselect this school". At the bottom left, there is a button labeled "✓ Location shared". At the bottom right, there is a green button labeled "Continue".

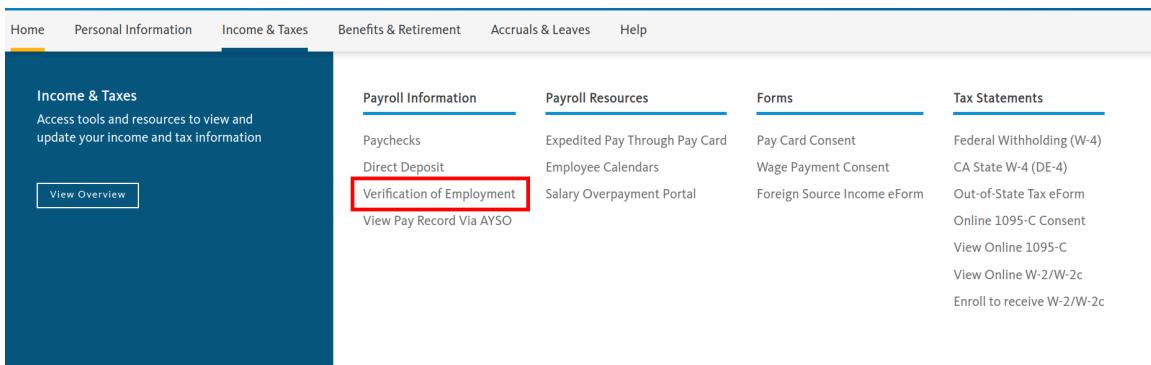
You will also be asked to share your location, which will then allow you to click on Continue and finish this section of the application. You will then be asked for additional information in the form of proof of you being associated with UC Berkeley. The following information will be for teacher roles.

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.

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

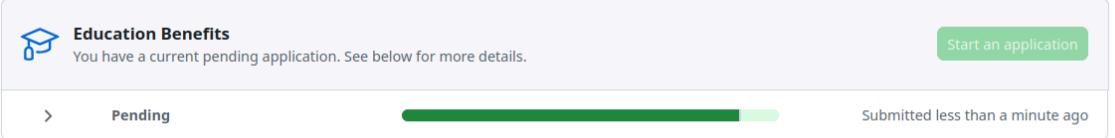
UCPath data at The Work Number
UCPath data sent to The Work Number excludes Employees who opted out from sending their information. For all other populations, UCPath demographic data such as job record information is submitted daily and income information is submitted after each pay date.

Generate Summary Report

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.

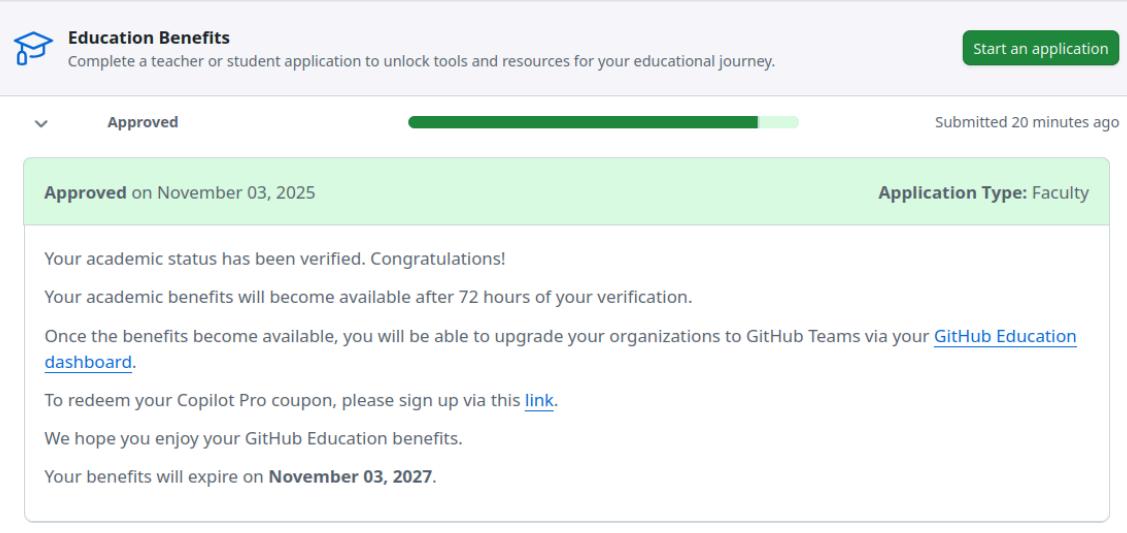
GitHub Education



A screenshot of a GitHub Education application status page. At the top, there is a blue graduation cap icon followed by the text "Education Benefits". Below this, a message says "You have a current pending application. See below for more details." To the right is a green button labeled "Start an application". In the center, the word "Pending" is displayed next to a progress bar that is mostly filled with a dark green color. To the right of the progress bar, the text "Submitted less than a minute ago" is shown. At the bottom left, there is a small downward arrow icon.

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

GitHub Education



A screenshot of a GitHub Education application status page showing an approved status. At the top, there is a blue graduation cap icon followed by the text "Education Benefits". Below this, a message says "Complete a teacher or student application to unlock tools and resources for your educational journey." To the right is a green button labeled "Start an application". In the center, the word "Approved" is displayed next to a progress bar that is almost entirely filled with a dark green color. To the right of the progress bar, the text "Submitted 20 minutes ago" is shown. At the bottom left, there is a small downward arrow icon. The main content area has a green header bar with the text "Approved on November 03, 2025" and "Application Type: Faculty". Below this, there is a message: "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. Your benefits will expire on **November 03, 2027**".

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 R
 - Install Visual Studio (VS) Code

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

4 Installing R

To install R, you will go to [The Comprehensive R Archive Network \(CRAN\) website](#). There you can download R for your operating system.

4.1 Mac

Follow the directions for installing R.

R for macOS

This directory contains binaries for the base distribution and of R and packages to run on macOS. R and package binaries for R versions older than 4.0.0 are only available from the [CRAN archive](#) so users of such versions should adjust the CRAN mirror setting (<https://cran-archive.r-project.org>) accordingly.

Note: Although we take precautions when assembling binaries, please use the normal precautions with downloaded executables.

R 4.5.2 "[Not] Part in a Rumble" released on 2025/10/31

Please check the integrity of the downloaded package by checking the signature:

`pkutil --check-signature R-4.5.2-arm64.pkg`

in the *Terminal* application. If Apple tools are not available you can check the SHA1 checksum of the downloaded image:

`openssl sha1 R-4.5.2-arm64.pkg`

Latest release:

For Apple silicon (M1,2,...) Macs:

[R-4.5.2-arm64.pkg](#)

SHA1-hash: 1cc0b3d78bc3b3857c6bf3128a9d414b130d938e
(ca. 97MB, notarized and signed)

For older Intel Macs:

[R-4.5.2-x86_64.pkg](#)

SHA1-hash: 0184504a11da63b26cc31f91a812e5456d523e0e
(ca. 100MB, notarized and signed)

R 4.5.2 binary for macOS 11 (**Big Sur**) and higher, signed and notarized packages.

Contains R 4.5.2 framework, R.app GUI 1.82, Tcl/Tk 8.6.12 X11 libraries and Texinfo 6.8. The latter two components are optional and can be omitted when choosing "custom install", they are only needed if you want to use the `tcltk` R package or build package documentation from sources.

macOS Ventura users: there is a known bug in Ventura preventing installations from some locations without a prompt. If the installation fails, move the downloaded file away from the *Downloads* folder (e.g., to your home or Desktop).

Note: the use of X11 (including `tcltk`) requires [XQuartz](#) (version 2.8.5 or later). Always reinstall XQuartz when upgrading your macOS to a new major version.

This release uses Xcode 16.2/arm64/14.2/x86_64 and GNU Fortran 14.2. If you wish to compile R packages which contain Fortran code, you may need to download the corresponding GNU Fortran compiler from <https://mac.R-project.org/tools>. Any external

Depending on the type of mac you have, you will choose one of the latest releases below. When going through the installation process, you can keep the defaults for everything. When installed, you should see a message stating **The installation was successful**.

4.2 Windows

Follow the directions for installing R.

Subdirectories:

base	Binaries for base distribution. This is what you want to install R for the first time .
contrib	Binaries of contributed CRAN packages (for R >= 4.0.x).
old contrib	Binaries of contributed CRAN packages for outdated versions of R (for R < 4.0.x).
Rtools	Tools to build R and R packages. This is what you want to build your own packages on Windows, or to build R itself.

Please do not submit binaries to CRAN. Package developers might want to contact Uwe Ligges directly in case of questions / suggestions related to Windows binaries.

You may also want to read the [R FAQ](#) and [R for Windows FAQ](#).

Note: CRAN does some checks on these binaries for viruses, but cannot give guarantees. Use the normal precautions with downloaded executables.

When installing for windows, you will use click on the `base` option and then click `Download R-## for Windows` to choose the latest R option for Windows. You can keep the defaults for everything. When installed, you should see a message stating `The installation was successful`.

4.3 Linux

There are some really good instructions on how to [install R here](#) for Linux distributions. One issue I have faced when installing R from these instructions is that I can install R, but common R packages do not install and lead to a `non-zero status` error. To prevent that, you will run the following code in your terminal. This will install necessary components that work under the hood for R. If you are having any issues with installing R, think about [submitting a consulting request](#) where D-Lab consultants can help you get started with R on your Linux machine.

```
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 R
- Install Visual Studio (VS) Code
- Adjust VSCode to work with R
- 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 for Mac install](#)

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

5.1.b Windows

[Link for Windows install](#)

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

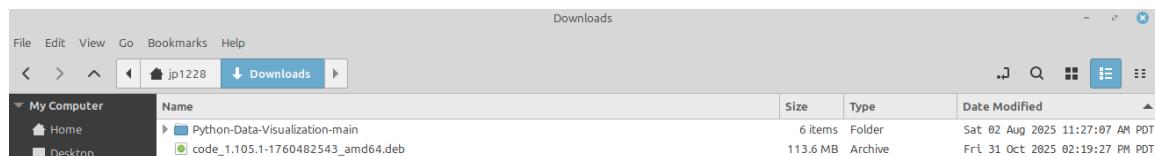
Note: Everything below shows the installation using Linux Mint.

System:

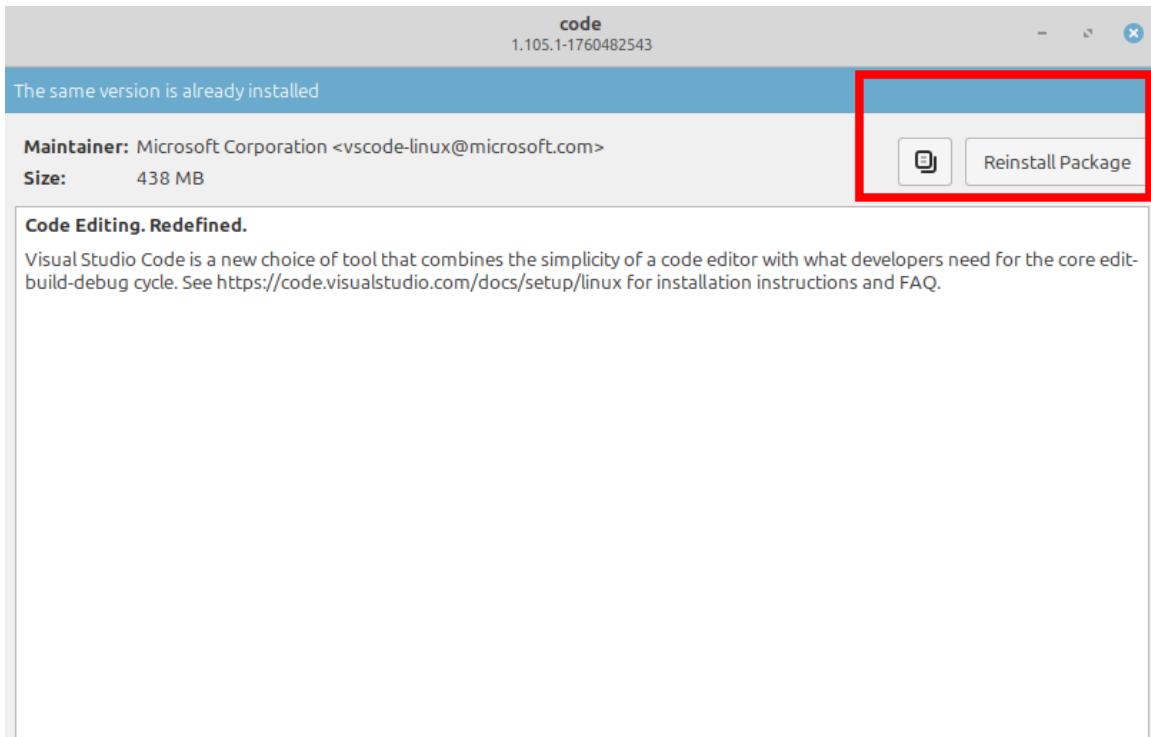
Host: jp1228-Swift-SF314-52 Kernel: 6.8.0-87-generic arch: x86_64 bits: 64
Desktop: Cinnamon v: 6.4.8 Distro: Linux Mint 22.1 Xia

[Link for Linux](#)

Option 1: Use the Link



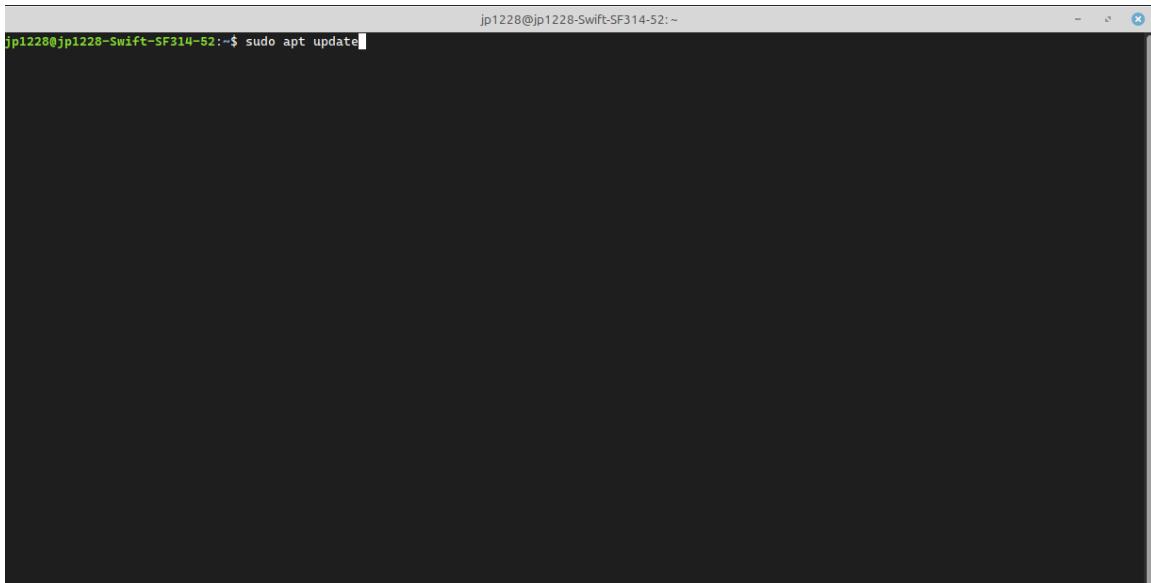
1. Click on the link above and open the download file.



2. Click Install Package to start the install.

You should be set to use VSCode.

Option 2: Download Using Terminal



1. Update your programs.

```

jp1228@jp1228-Swift-SF314-52: ~
Hit:4 https://cloud.r-project.org/bin/linux/ubuntu noble-cran#0/ 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://packagers.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.
jp1228@jp1228-swift-sF314-52:~$ sudo apt install ./code_1.105.1-17604825b3_amd64.deb

```

- 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

```

You should now be set up to use VSCode. If you are having difficulties, it may be easier to use option #1 above.

5.2 VSCode Documentation

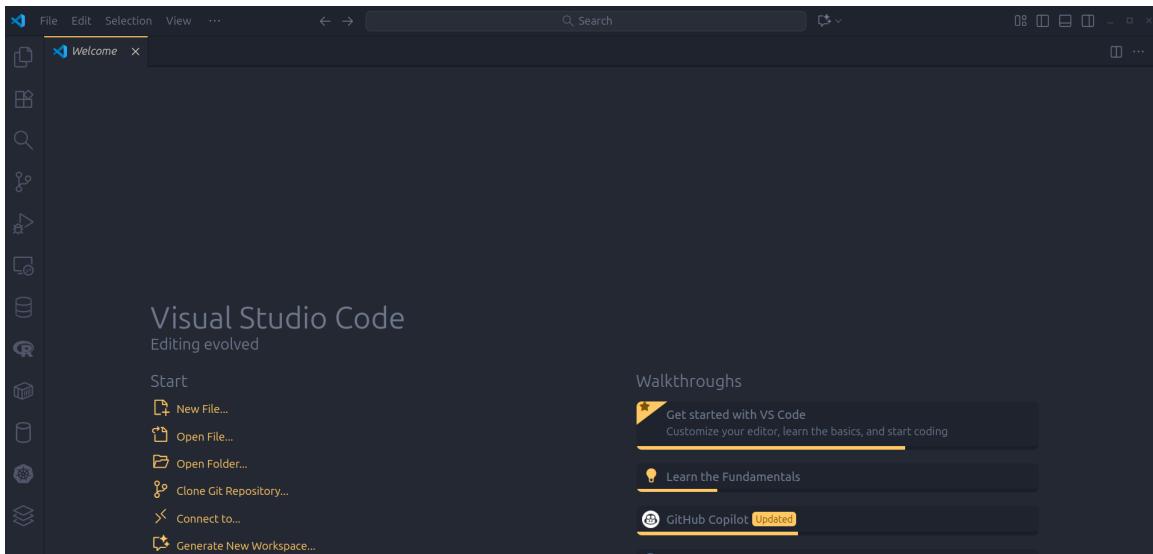
When you installed VSCode, it should have brought you to the documentation page. If not, you can find all the [documentation here](#). [This tutorial](#) also provides an in-depth tutorial on getting started with VSCode.

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

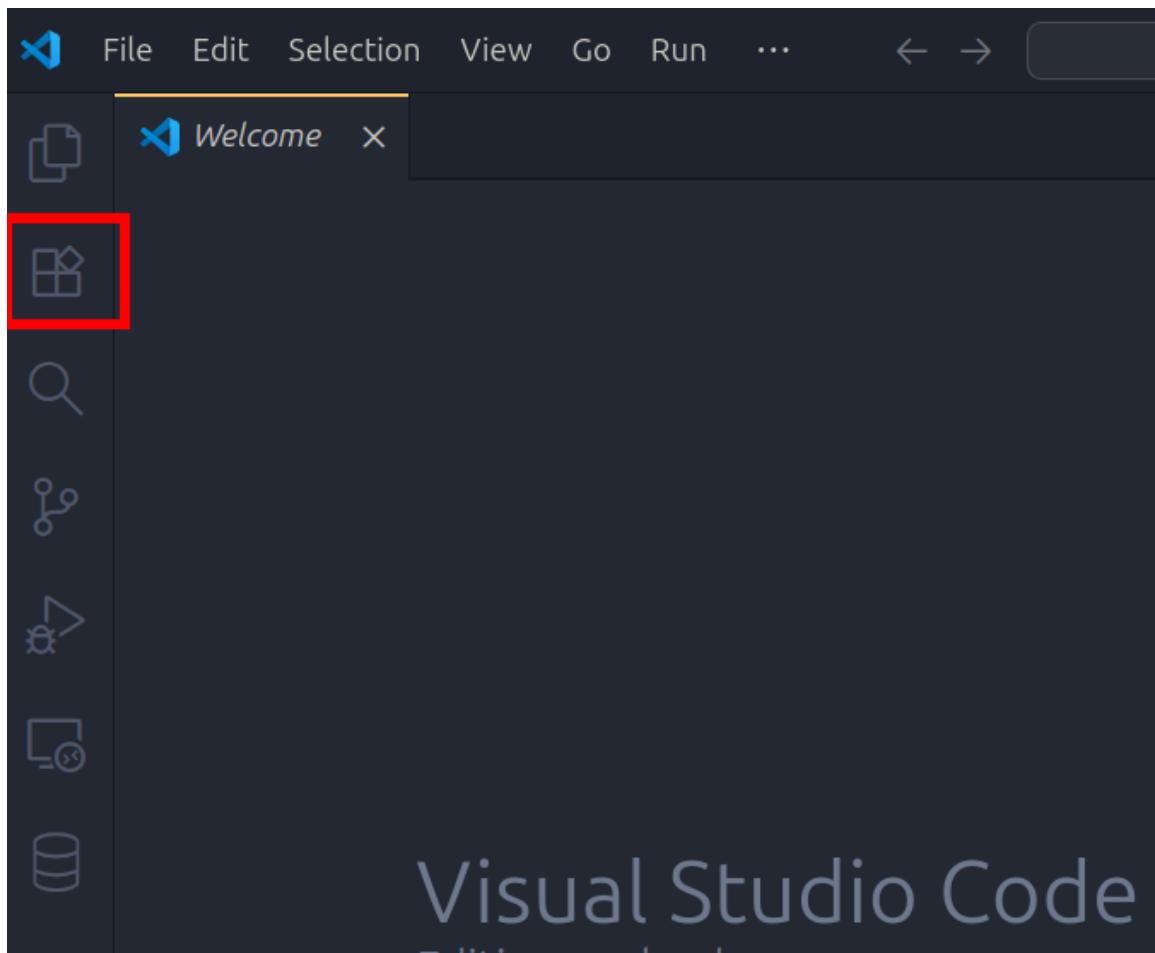
6 Setting up R in VSCode

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.

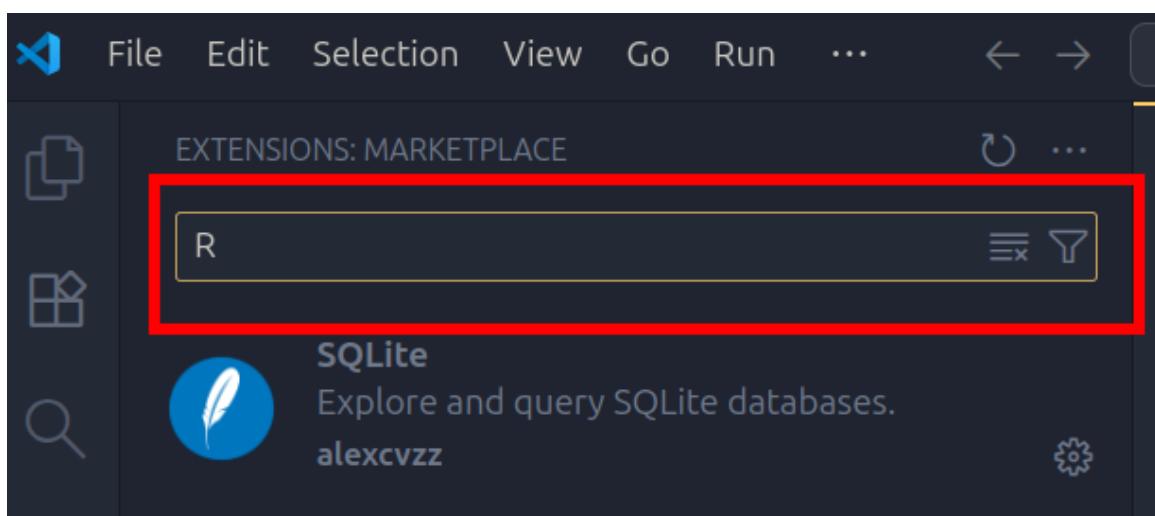
Depending on the version of R you installed, there may be some issues in incorporating some of the packages. I will include alternatives to try and make sure everything works. If you have not downloaded R, then you will want to do that first. Once you have R, then you can move forward with installing extensions for R.



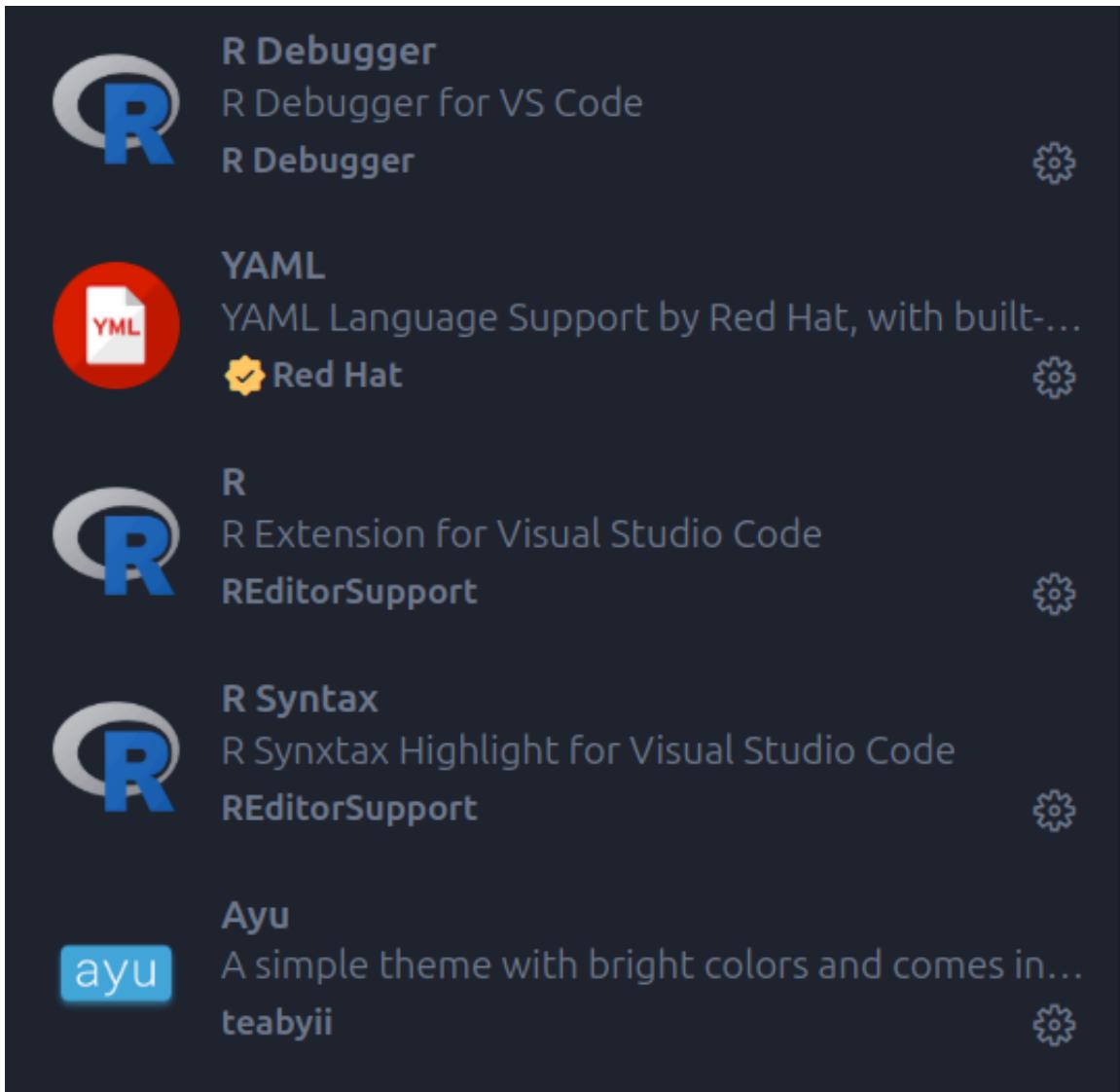
Once you open VScode, you will be on the Welcome page.



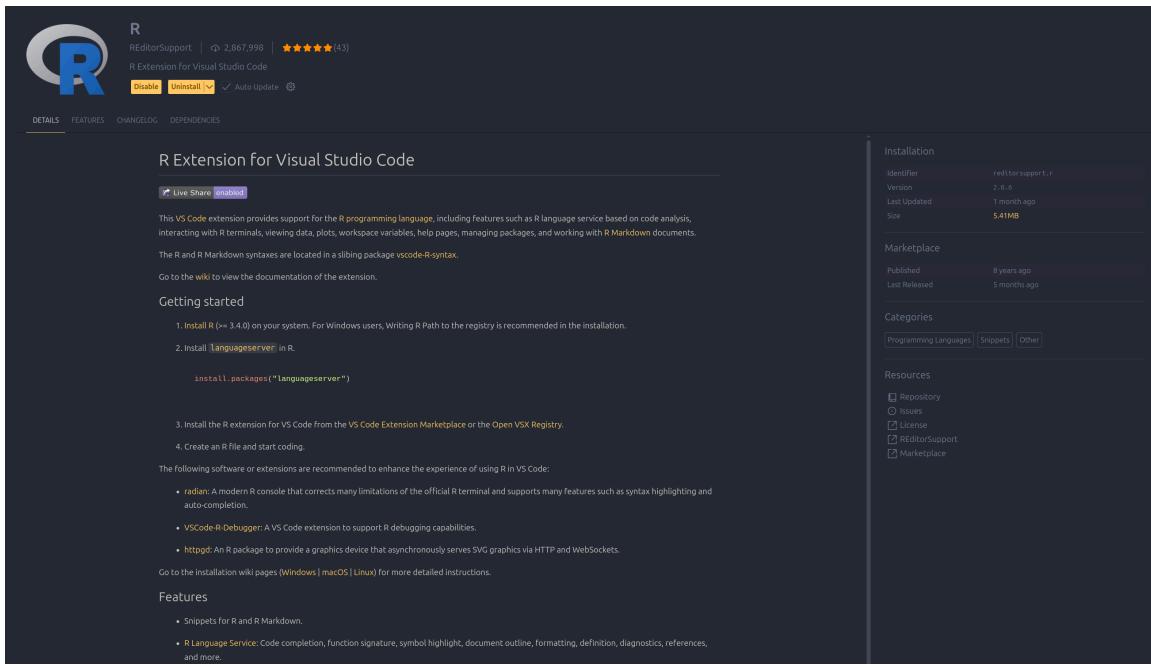
From there, the first place you will want to go is to the left sidebar. There you will see several tabs. You will click on the Extensions tab shown above.



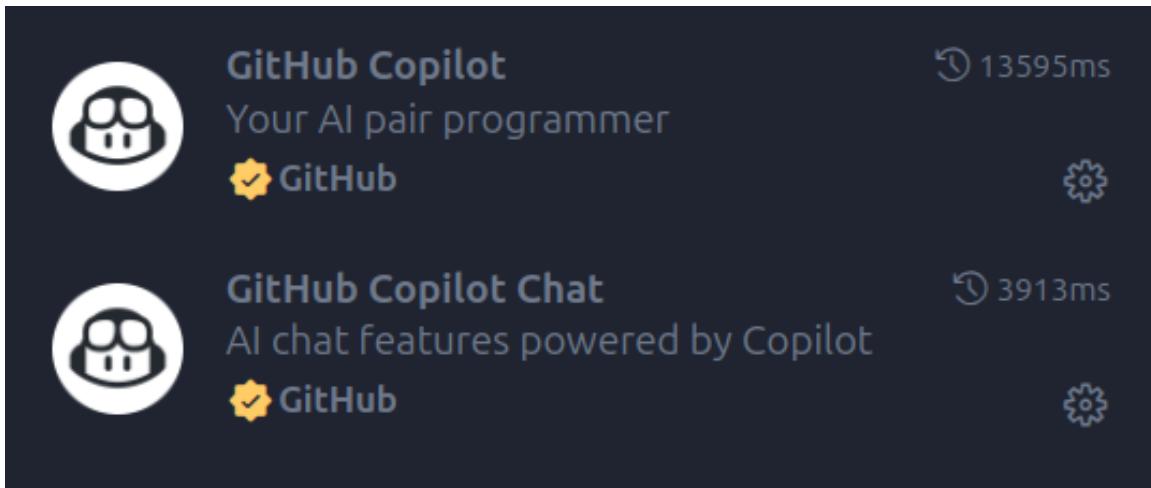
The easiest way of finding all the extensions you will need is to use the search bar at the top. *You can also use this to find a theme for your VScode. [See several examples of themes here](#) that you can then search for by name.



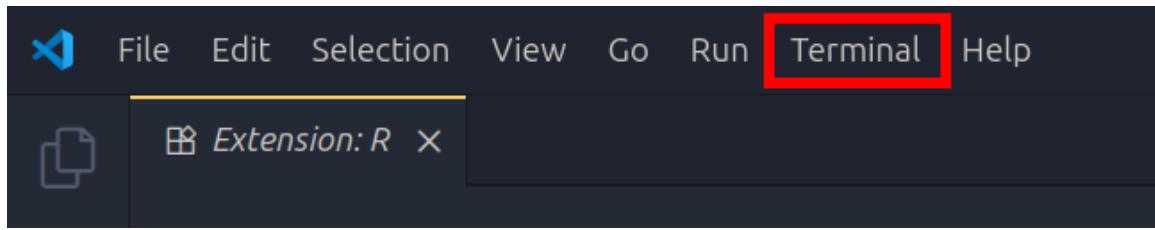
First, you will want to install the R extension. In addition to installing the R extension, you may also want to install the **R Debugger** and the **R Syntax** extensions. These add some additional tools when using R.



Once you click on the extension on the sidebar, a tab will open up on the main window section of VSCode, there you can click Install and read the documentation behind each extension. For the R extension, there are summarized instructions below.



Right now would be a great time to search and install GitHub Copilot. There are two extensions that should be installed, GitHub Copilot and the GitHub Copilot Chat extension. GitHub Copilot Chat should be installed once you install GitHub Copilot.



The easiest way to install the necessary packages is to go to the top left corner on VSCode and click on Terminal. There you will then click on New Terminal. A new terminal will open at the bottom of your VSCode. It should be listed as a bash terminal.

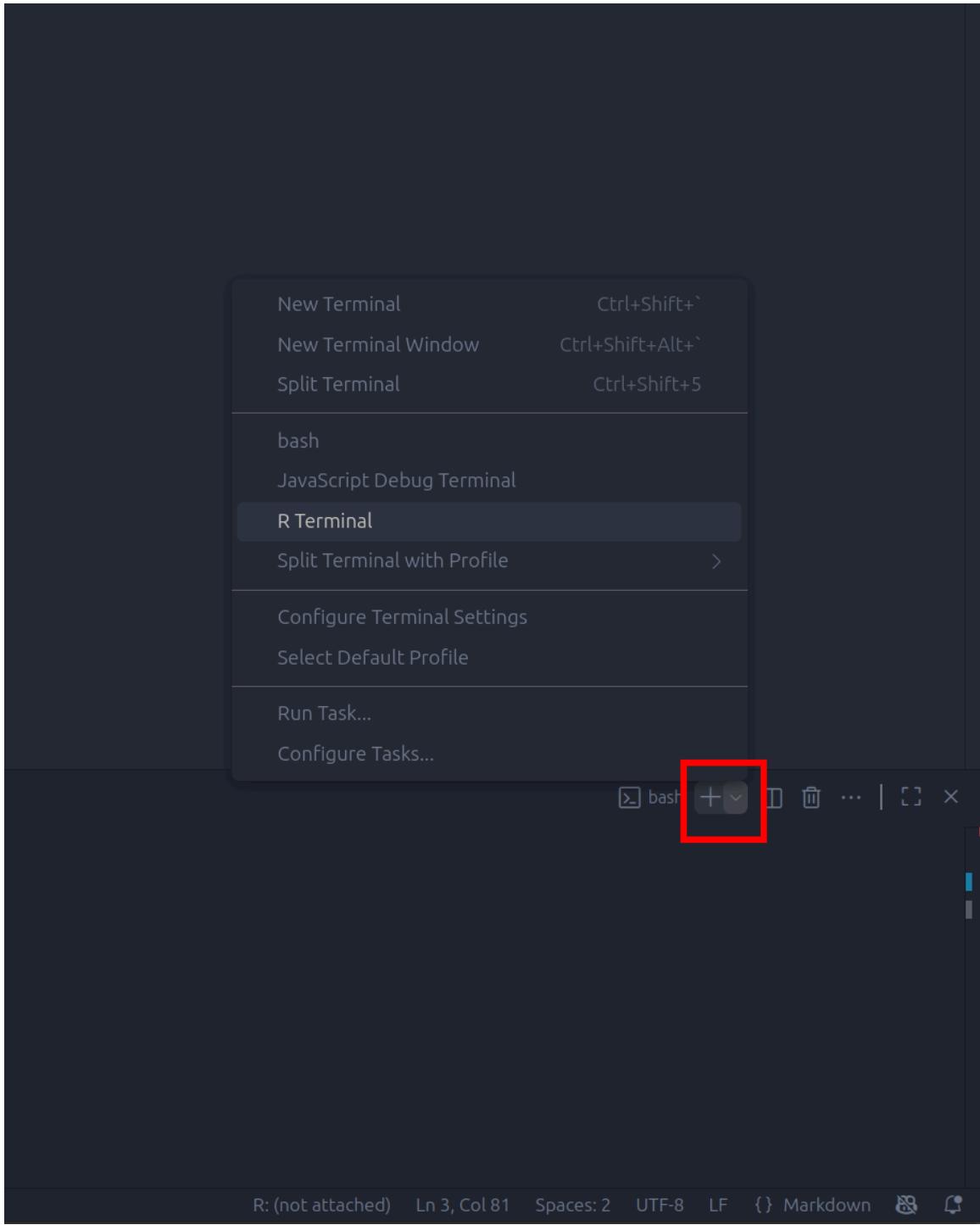
```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

● 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:~$ █
```

If you want to double check the status of your R installation, you can type `R --version` into your bash terminal to see if R is installed correctly. You can also see where R is located by typing `which R`. This will show you the location of R that VSCode is using.



The easiest way to install the `languageserver` and `httpgd` packages is by going to the right side of your terminal and clicking on the dropdown menu and choosing a R Terminal. As mentioned in the documentation for the R extension, you can install the `radian` package and make the appropriate changes in the settings.

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")
```

As this is now a R terminal, you can install the languageserver package by using `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
>
```

When installing the `httpgd` package, I have found some issues when trying to install it from newer versions of R. If you experience these issues, you should install the development version of the package.

```

> 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

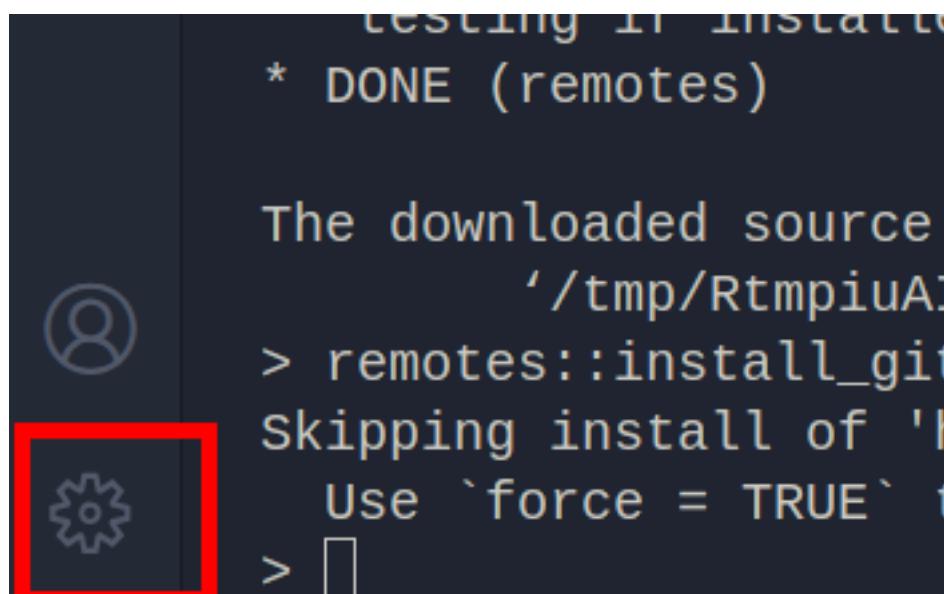
* installing *source* package 'remotes' ...
** this is package 'remotes' version '2.5.0'
** package 'remotes' successfully unpacked and MD5 sums checked
** using staged installation
** R
** inst
** byte-compile and prepare package for lazy loading
** help
*** installing help indices
** building package 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/RtmpiuIbt/downloaded_packages'
> remotes::install_github("nx10/httpgd")
```

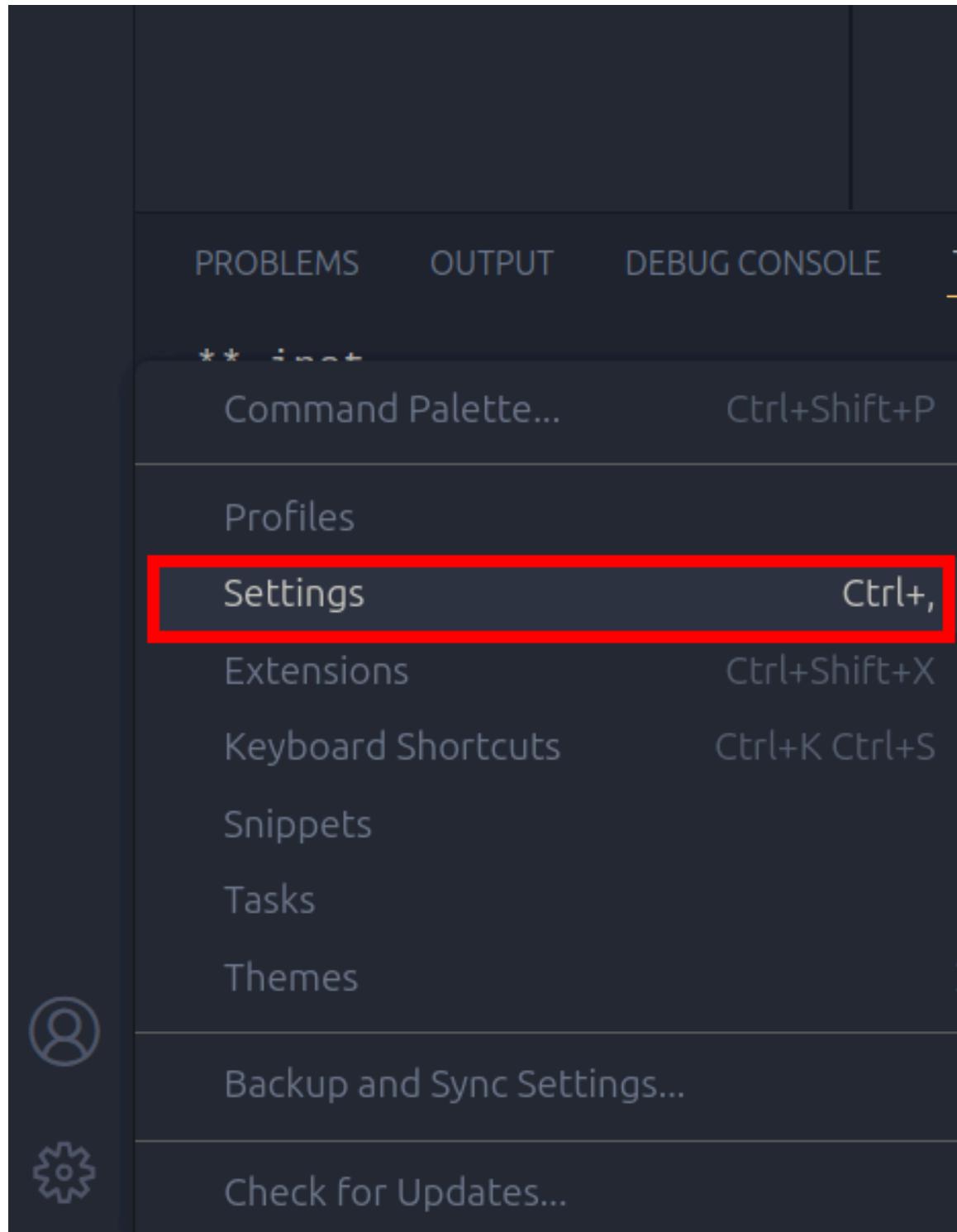
You can get the `httpgd` package by first installing the `remotes` package (using the typical method of `install.packages("remotes")`). This will then allow you to install the development version from GitHub. If you want to follow along with the instructions from the developer(s) of the `httpgd` package,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

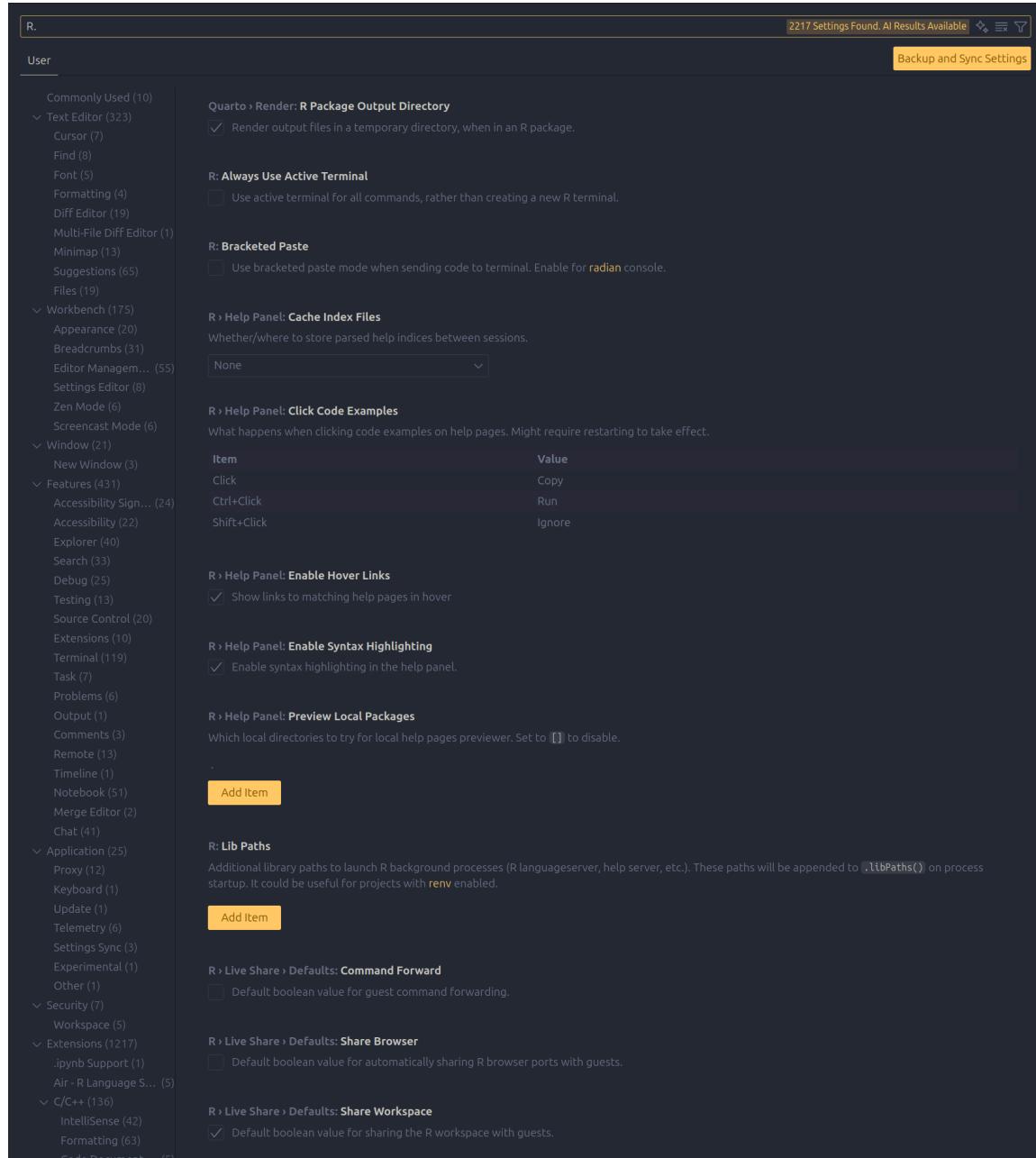
1. Making Changes by Settings Tab



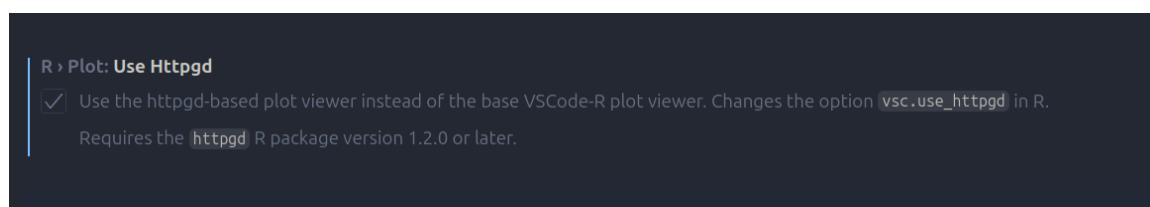
At the bottom of the sidebar you will see a gear. Clicking on the gear will open options for customizing your VSCode.



Clicking on settings will show you everything that you can alter in VSCode.



To make changes to R in VSCode, you will type out R. in the search bar.

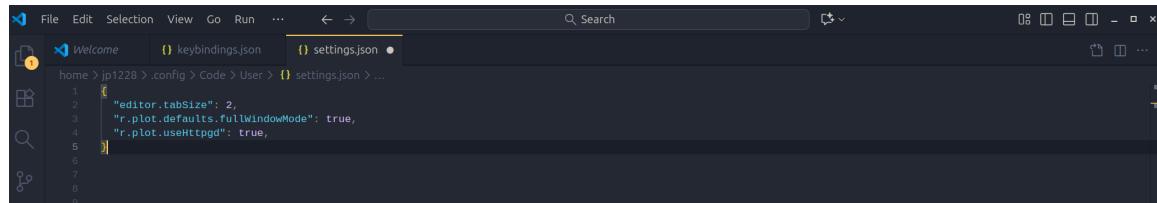


For your purposes, you can scroll through the R options until you find the `Use Httpgd` option and click on the check box.

2. Making Changes Using the `settings.json` File

The second option is to use 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 R 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 R version 4.1.0 or 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.

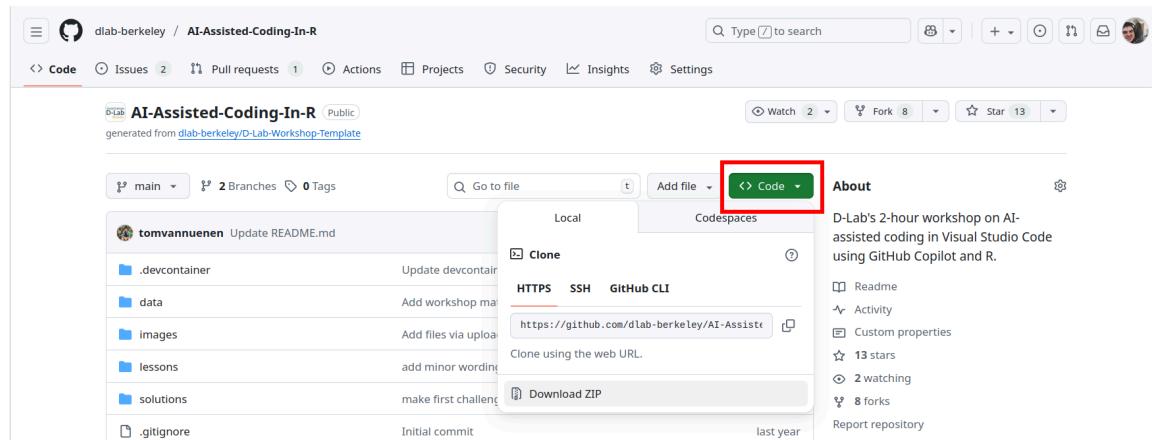
Type to search in keybindings			
Command	Keybinding	When	Source
Accept Inline Completion	Ctrl + /	accessibleViewIsShown && accessibleViewCurrentProviderId == 'inl...	System
Accept Inline Suggestion	Tab	inlineEditIsVisible && tabShouldAcceptInlineEdit && !editorHover...	System
Accept Inline Suggestion	Tab	inInlineEditsPreviewEditor	System
Accept Next Word Of Inline Suggestion	Ctrl + RightArrow	cursorBeforeGhostText && inlineSuggestionVisible && !accessibili...	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 Above	Shift + Alt + UpArrow	editorTextFocus	System
Add Cursor Below	Ctrl + Shift + DownArrow	editorTextFocus	System
Add Cursor Below	Shift + Alt + 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
 - Sign up for GitHub Copilot
 - Install R
 - Install Visual Studio (VS) Code
 - Adjust VSCode to work with R
 - [] Download Zip file

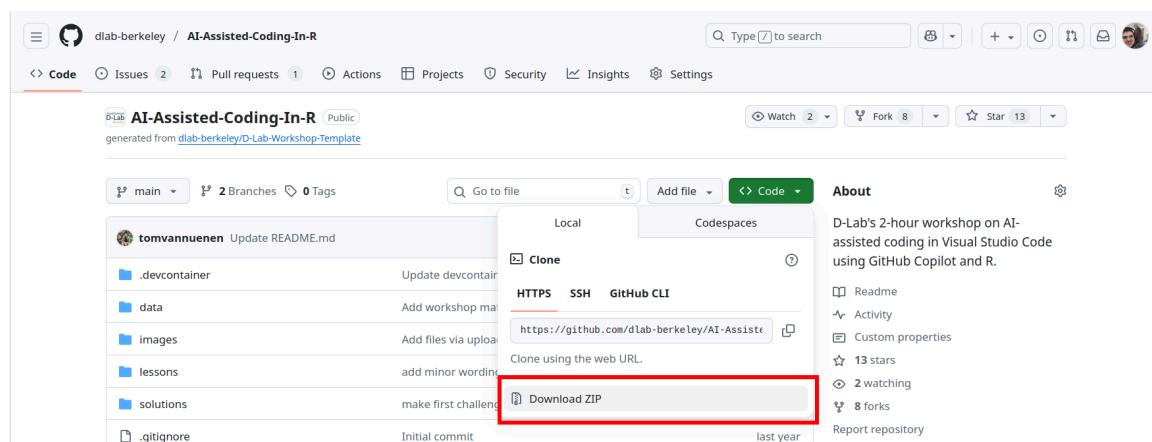
7 Download Zip File

To get all the materials for the AI-Assisted-Coding-In-R 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-R 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.

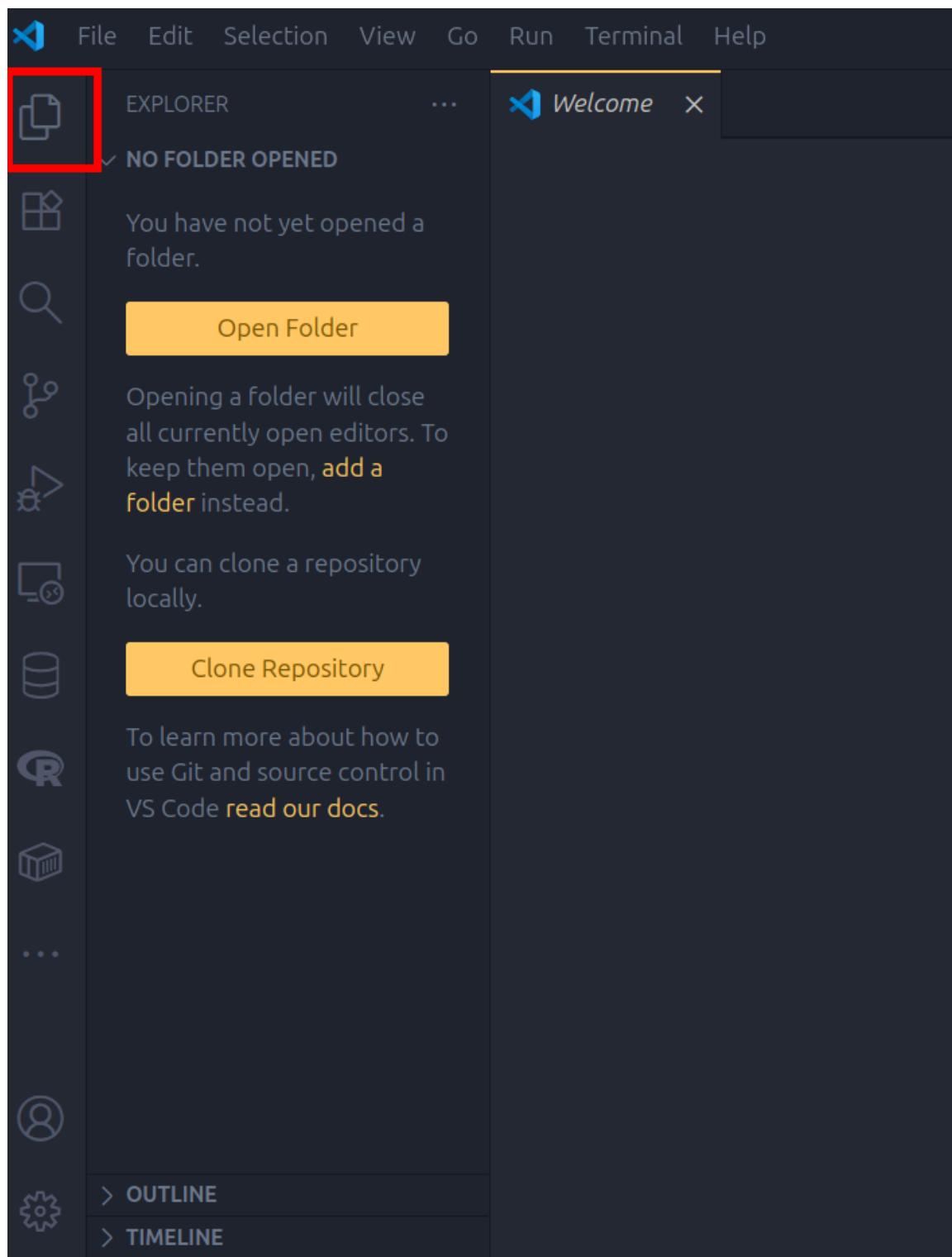
Name	Size	Type
Desktop	8 items	Folder
Documents	13 items	Folder
Downloads	310 items	Folder
example_vscode	7 items	Folder
Music	0 items	Folder
Pictures	1 item	Folder

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 exam-

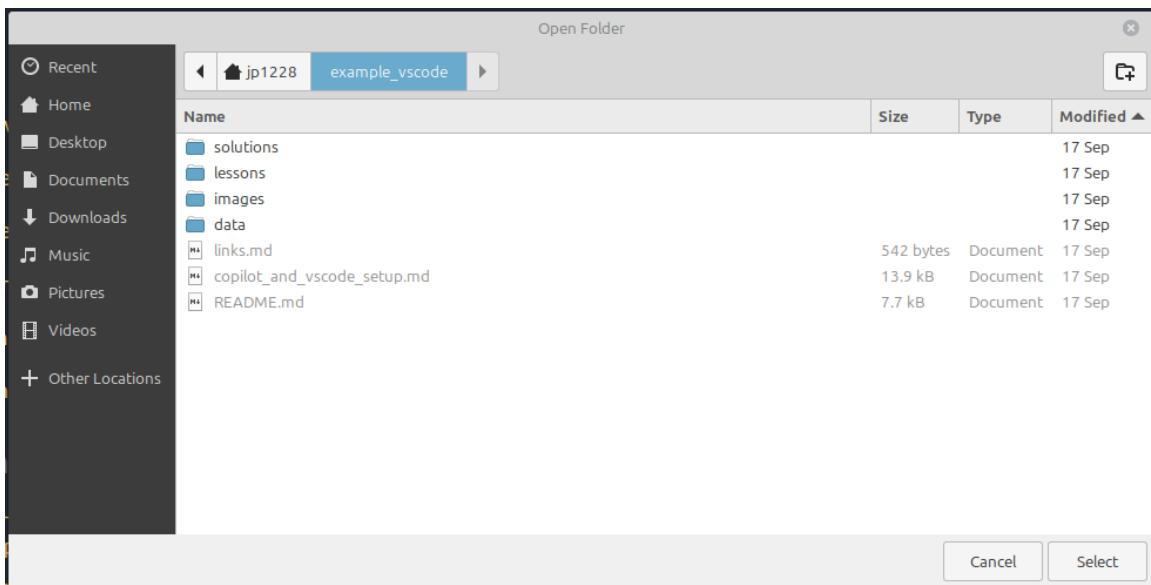
ple_vscode) under my username folder. You can create the folder on your Desktop for easy access or within your Documents folder if you would like.

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

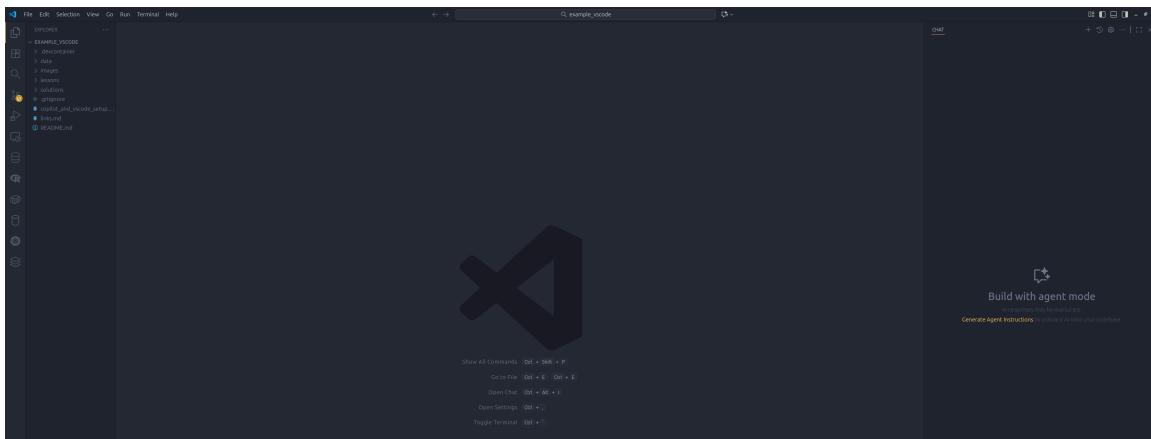
Within the folder, I copied and pasted everything from the ZIP file into this folder. The files should look similar to the screenshot above.



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.

The screenshot shows the Visual Studio Code interface with an R notebook open. The notebook file is named 'workshop.Rmd'. The code in the notebook includes sections like '# Learning Objectives' and '# 1. Getting Comfortable with Visual Studio Code'. A sidebar on the right is titled 'Build with agent mode' with the sub-instruction 'Generate Agent Instructions to onboard AI onto your codebase'. The status bar at the bottom indicates the file is 'R (not attached)'.

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-R workshop.

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