Setting up R for VS Code on Windows

Faisal Shaik · Follow
Published in Towards Dev
6 min read · May 11, 2024

Listen Share

Hello everyone, this is my first post on Medium. I'm not very familiar with this place to be honest but I wanted to share a guide on here for comprehensively setting up R on VS Code to help others.

You can ditch R Studio without losing out on much if you follow this guide completely. Enjoy all the benefits of using VS Code for R, including intellisense, copilot, version control, and more. Note that this guide is for Windows but you can easily follow it for Mac/Linux as well.

Overview of Steps

Here is a brief overview of the steps before we begin.

1. Have VS Code installed
2. Install R and language server
3. Install Python then Radian.
4. Install **Markdown All in One** and the **R** extension on VS code
5. Configure R.exe path, keyboard shortcuts, and .Rmd file identification for VS Code.

Figuring out the last step was very annoying but if you follow these notes you will not have to go through the same pain as I did.

Install R

Download R for windows from https://cran.r-project.org/bin/windows/base/ and note down where you install it. For me, I installed it in the default location: `C:\Program Files\R\R-4.4.0`. Note my version is 4.4.0, for you it might be newer.

Add R's bin to System PATH

You will need to add the bin folder of your R installation: `path\to\R\R-4.4.0\bin` to your system's PATH. If you don't know how to, read this.

Begin by pressing Windows start button and searching "*edit the system environment variables*". Click the control panel option that shows up then do the following:

- A popup called *System Properties* should open. Go to the *Advanced* tab if you're not already in it.
- Click *Environment Variables*.
- In *System Variables* click *Path* then the *Edit…* button.
- Click the *New* button then enter: `path\to\R\R-4.4.0\bin` then press enter.
- Click OK, OK, Apply, OK.
- Good job you're done.

Check if R has been added to the system's path by typing `R --version` in cmd outside of the R installation folder. If R is successfully installed and in your system's PATH, you should get a message showing the version of R you have.

At this point you can type R in a terminal to run R. Note that in PowerShell you will need to run type `R.exe` because R already stands for a different existing command.

Install languageserver

In `path\to\R\R-4.4.0\bin\x64`, run `Rgui.exe`. Type the following in the R console:

```
install.packages("languageserversetup")
```

You may be prompted to select a CRAN mirror. Select `0-Cloud`. Now type this in the R console:

```
languageserversetup::languageserver_install()
```

You will get a prompt saying *"This will attempt to use source(https...)..."* etc. Select yes. Another pop up will appear saying *"Do you want to install from sources the package..."*, doesn't matter if you select yes or no.

Now lastly, type in the following into R console:

```
languageserversetup::languageserver_add_to_rprofile()
install.packages("rlang")
install.packages("jsonlite")
```

If you are asked to create a personal library, select yes.

Edit .Rprofile

You may or may not need to do this step depending on whether starting R is giving you any errors. In cmd, run R and see what happens. You will need to edit your .Rprofile you get any errors like the following:

```
Error: '\U' used without hex digits in character string (<input>:4:36)
```

This means the profile is using backslashes instead of forward slashes. You will need to locate where the `.Rprofile` file is stored and change the slashes. It is commonly in the `C:Users\User\Documents` directory, where `C:Users\User` is your *user profile*. To verify you user profile, in cmd you can type `cd %USERPROFILE%`.

If you cannot find a file called `.Rprofile` in Documents, launch `Rgui.exe` again, click *File* at the top right then *Display file(s)*, it may be there.

If you still cannot find it, then do a full file search in file explorer, it should have been added somewhere in your computer after you ran `languageserversetup::languageserver_add_to_rprofile()` earlier.

Once you have found the `.Rprofile` file, change all the backslashes into forward slashes. For example, mine looks like this after I have fixed it:

```
# LanguageServer Setup Start (do not change this chunk)
# to remove this, run languageserversetup::remove_from_rprofile

if (requireNamespace('languageserversetup', quietly = TRUE)) {
  options(langserver_library = 'C:/Users/Admin/Documents/languageserver-library
  languageserversetup::languageserver_startup()
  unloadNamespace('languageserversetup')
}
# LanguageServer Setup End
```

Now launch R and you should no longer get any errors.

Install Radian

You could skip this step but I highly recommend you install Radian to use as a shell for R instead of anything else. You will need Python installed. If you don't have Python installed then look up a tutorial online. Ensure your installation of Python is added to the system's PATH.

Once you have Python installed, open PowerShell or cmd as an administrator and run the following command:

```
pip3 install -U radian
```

Now you can run Radian,

```
radian
```

Note that Radian is useful for not only R, but also Python and Julia. You will set this to be the default terminal for R in VS Code later.

Install R and Markdown All in One Extensions in VS Code

Before installing the two extensions you will need to get the path to your R executable. Open a terminal and run R, then type R.home("bin") like so:

```
> R.home("bin")
[1] "C:/PROGRA~1/R/R-44~1.0/bin/x64"
```

For me, R returns `C:/PROGRA~1/R/R-44~1.0/bin/x64`, which is the path to my R executable. Note this down.

Install the *R* and *Markdown All in One* VS Code extensions. Now go in the extension settings for the R extension and search *Rpath: Windows*. Set this to the R path you noted down.

At this point you're all set to use VS Code for .r files. However, further configuration is needed for .Rmd files.

Set .Rmd File Association to rmarkdown

VS Code might have the file association for .Rmd files set to just plain markdown. If this is the case, you wont be get the benefit of R intellisense, nor will you be able to

run code chunks in your .Rmd file.

Open `settings.json` from the command palette (*CTRL* + *SHIFT* + *P*) and ensure the following is present:

```
"files.associations": {
  "*.rmd": "rmarkdown"
},
```

Make sure that it's **rmarkdown**, not just markdown.

Add Keyboard Shortcut for .Rmd Files

Note that you can run lines of code in .R files by having your cursor on them and pressing CTRL + ENTER. This may not work if you try to run code inside code chunks in .Rmd files.

Open command palette and search *Keyboard Shortcuts (JSON)*. Add the following code to the JSON file:

```
{
  "key": "ctrl+enter",
  "command": "r.runSelection",
  "when": "editorTextFocus && !editorReadonly && (editorLangId == 'r' || editor
}
```

This adds R's run selection command for .Rmd files. Be careful because this will run anything your cursor is on, including text outside of code chunks. I'm not sure if there's a better way to do this. Definitely reach out to me if you know of one.

You may also add other shortcuts for .Rmd files for commands such as *Run Current Chunk, Run Above Chunks, Run All Chunks,* and so on. Add them with your desired key binds in the following format

```
{
  "key": "shortcut+here",
  "command": "r.command",
  "when": "editorTextFocus && !editorReadonly && (editorLangId == 'rmd' || edit
}
```

Here is a list of relevant commands to replace `command` with above.

- runAboveChunks
- runAllChunks
- runCurrentChunk

For example,

```
{
  "key": "ctrl+alt+p",
  "command": "r.runAboveChunks",
  "when": "editorTextFocus && !editorReadonly && (editorLangId == 'rmd' || edit
}
```

Set Radian to be Terminal for R

The last thing we want to do is set Radian to be the default terminal VS Code runs R on when we press CTRL + ENTER. Open the extension settings for the R extension and search *Rterm: Windows*. Set this value to path to your Radian executable.

To find where your Radian executable is just type `where radian` in any terminal and you should get the path.

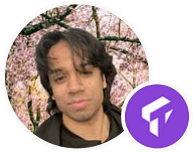
End

Good job, you're done. Enjoy coding R with all the benefits of Visual Studio Code. Feel free to give me any feedback for my R notes and/or installation guide.

You can contact me on LinkedIn for a quick reply. If you want to read some notes on using R for Data Analysis and ML checkout my GitHub repository.

Hope this helped you.

R Coding Programming Guides And Tutorials Visual Studio Code



Follow

Written by Faisal Shaik

1 Follower · Writer for Towards Dev

Sup, I'm Faisal, president of the Physics club at UofT. Triple major in CS, math, and physics. ML researcher at UofT DSI, SWE at DataAnnotation.

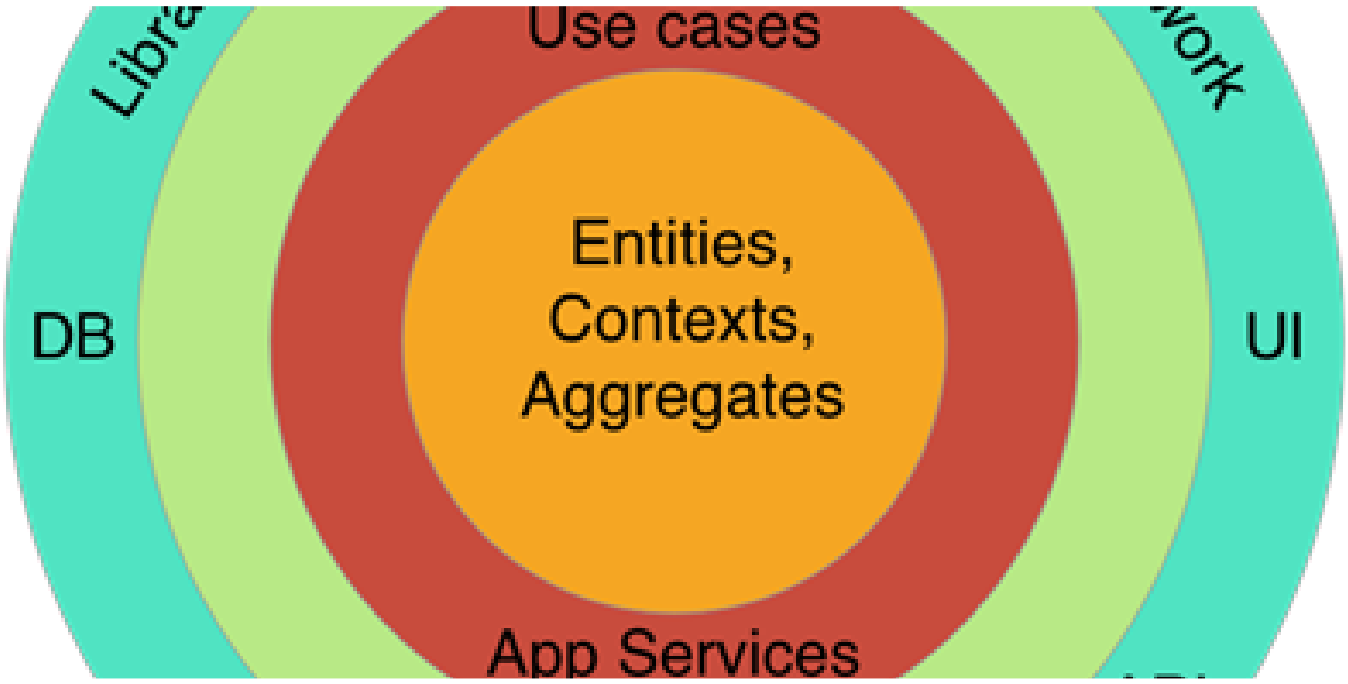
More from Faisal Shaik and Towards Dev

Faisal Shaik in Towards Dev

Managing Virtual Environments with Different Python Interpreters

Hi everyone, I'm back with another code guide. This time I wanted to share a detailed guide demonstrating how to install and manage using...

May 22 3

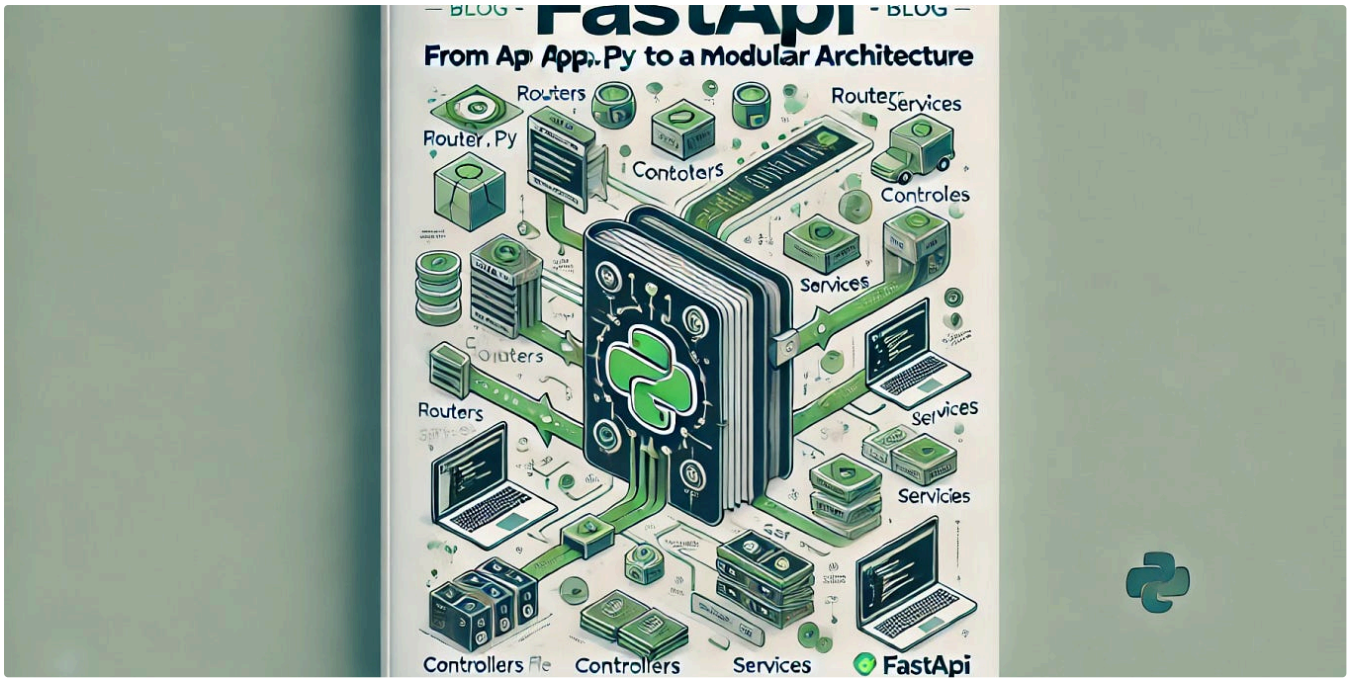


David Lee in Towards Dev

You don't Need a Book to Know DDD(Domain-Driven Design)

It took me a while to figure out the patterns behind DDD, though the most important thing are ubiquitous language and bounded context. In...

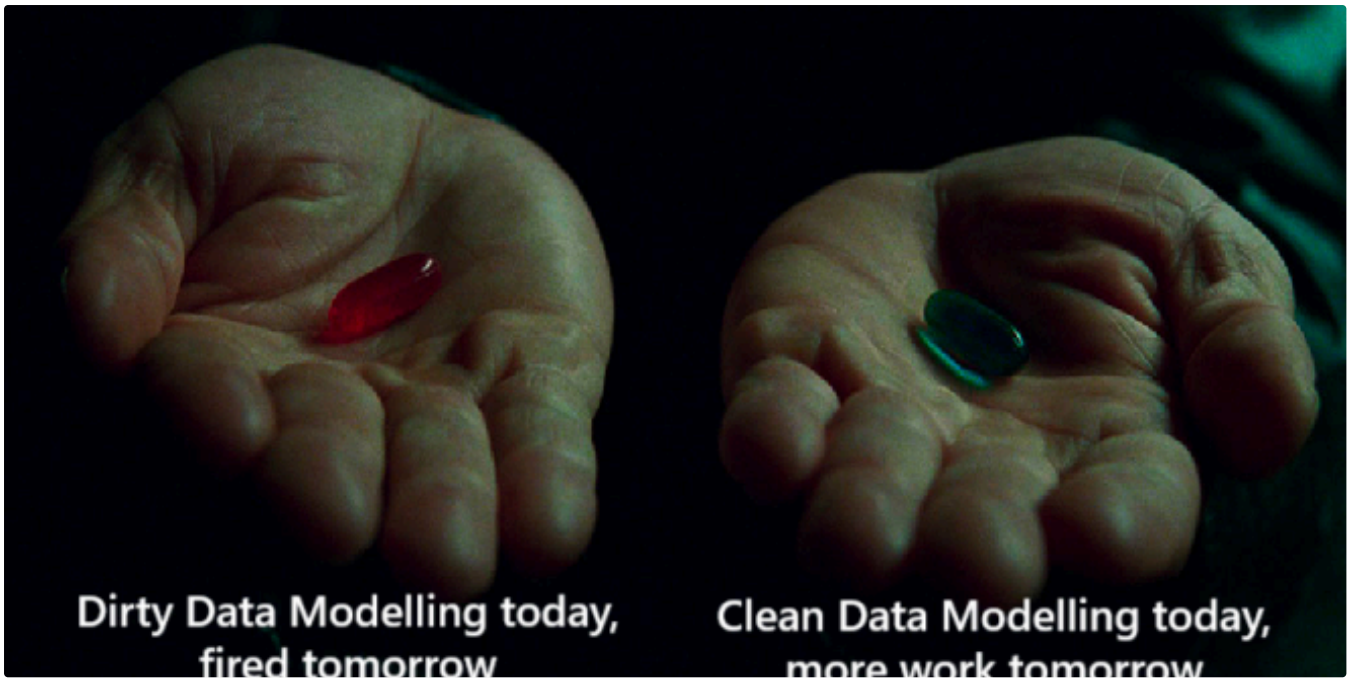
Aug 17 327 5



AST-LW in Towards Dev

FastAPI: From App.py to a Modular Architecture

Jun 24 765 9



Avin Kohale in Towards Dev

Kimball vs. One Big Table vs. Data Vault in Data Modeling

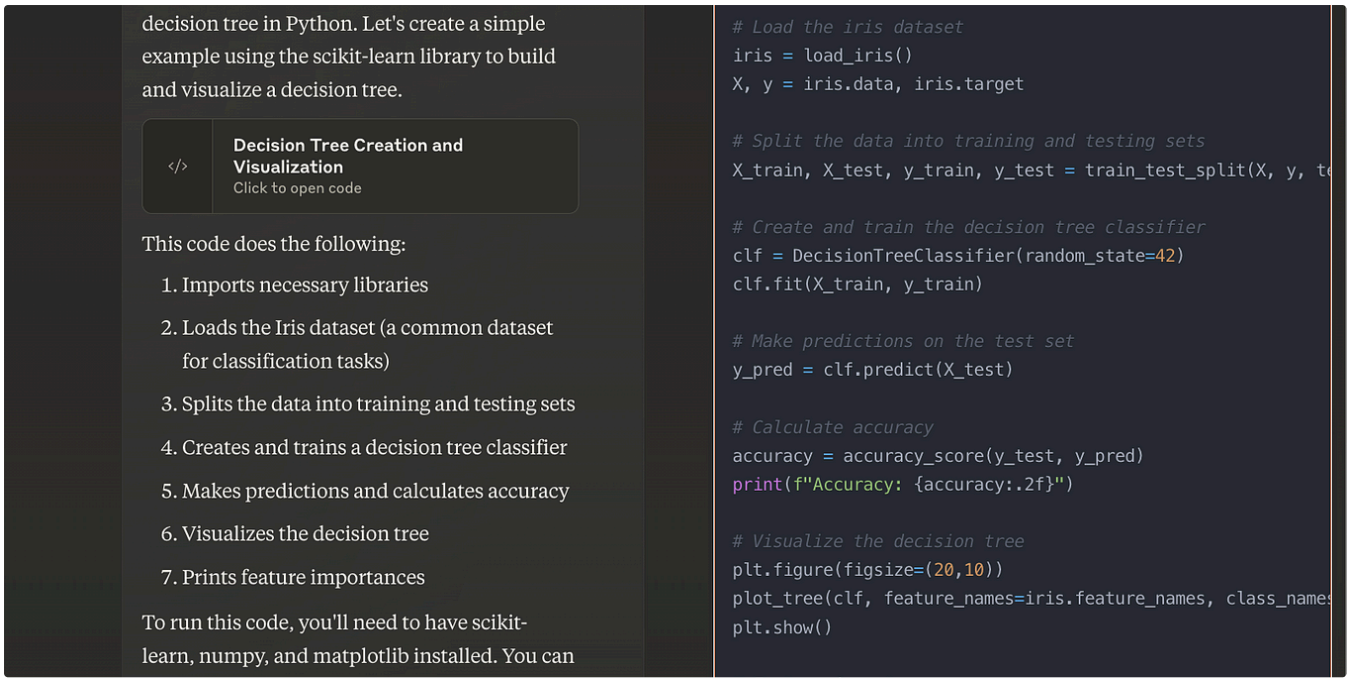
Choose the best data modelling technique after reading this blog!

Aug 22 222 8

See all from Faisal Shaik

See all from Towards Dev

Recommended from Medium

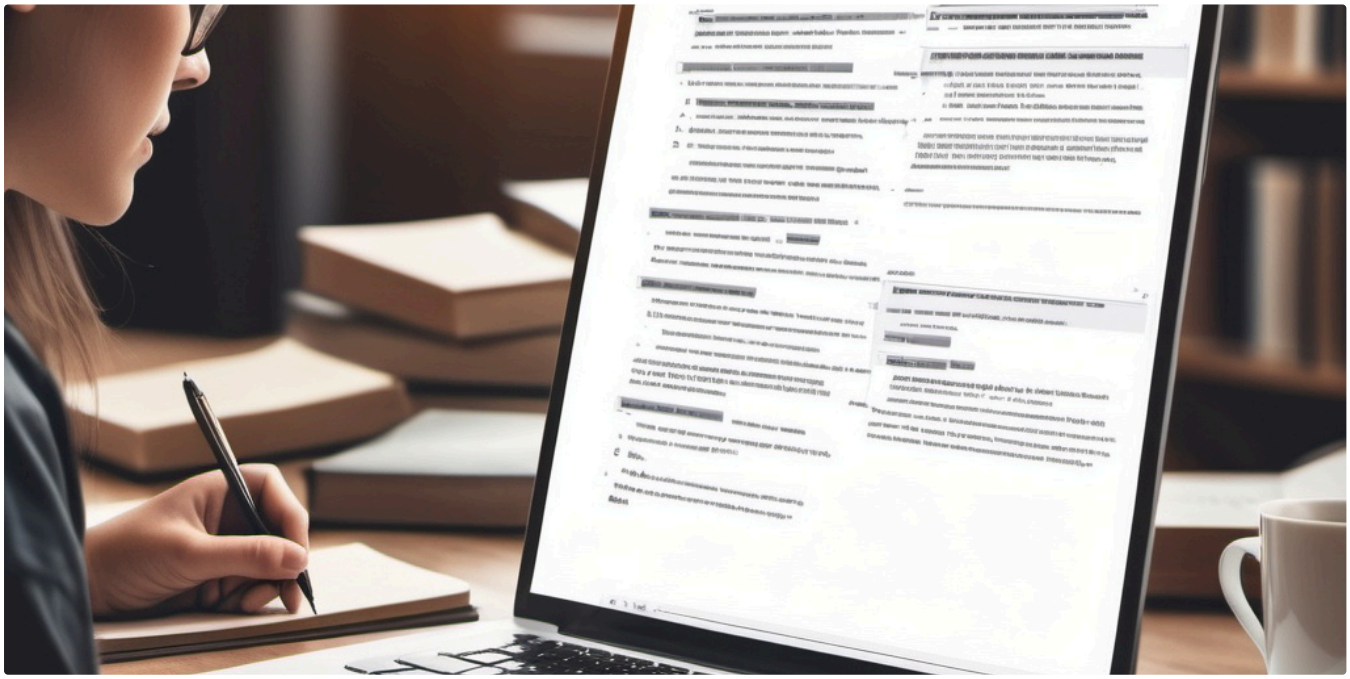


Pierre DeBois

IT Pro Today—How Claude Artifacts Enhances AI for R Programming

Claude is becoming a valuable AI assistant for programming languages, including R. Learn how Claude's latest feature, Artifacts, enhances...

Sep 29 3



Liv (Leslie) McFarlin in Dev Genius

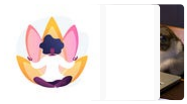
Ollama + tidychatmodels: Categorizing Text Data in R

As someone who handles a lot of qualitative data on a weekly basis, I try to spend some of my spare time exploring new ways of gathering or...

5d ago 3 1

Lists

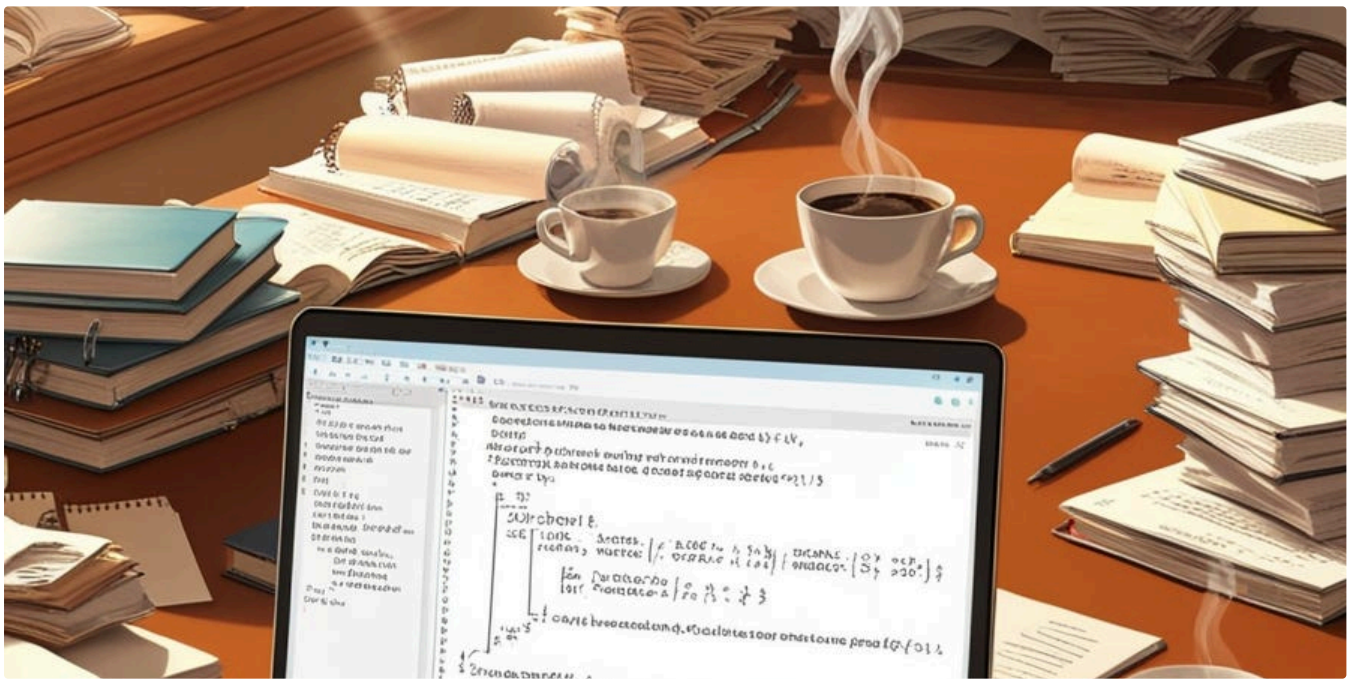
General Coding Knowledge
20 stories · 1622 saves



Stories to Help You Grow as a Software Developer
19 stories · 1396 saves

Coding & Development
11 stories · 834 saves

ChatGPT
21 stories · 823 saves

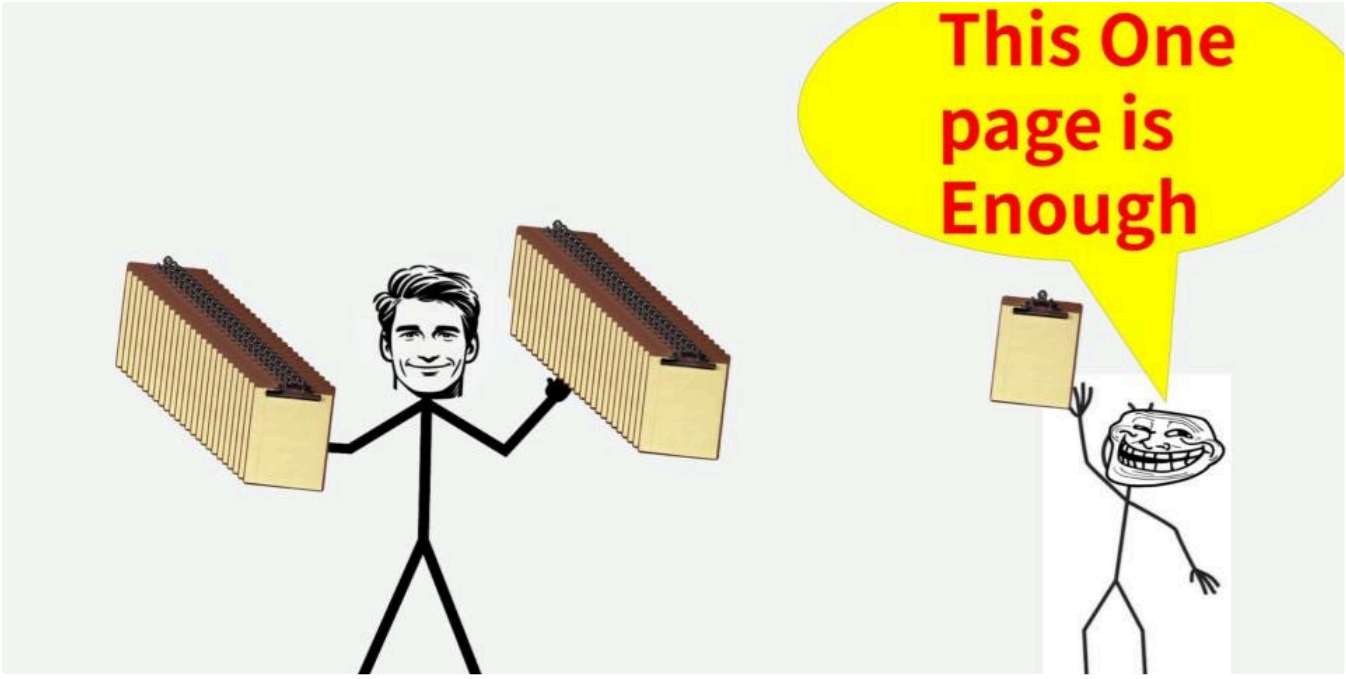


L Data PR

Statistical Functions in R

Hello Folks,

Sep 26 31

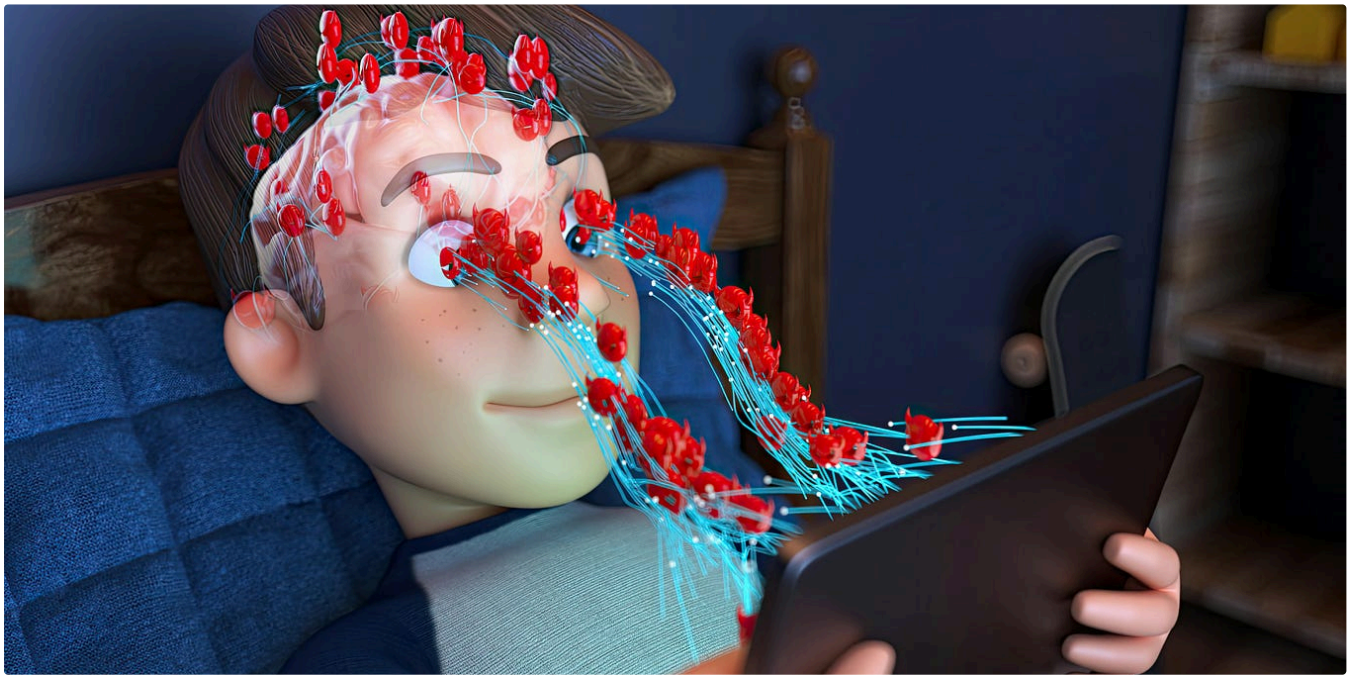


Ajay Parmar in Top Python Libraries

11-Python Packages That Turn Hundreds of Lines of Code into One

🚀 11 Python Libraries That Turn 100 Lines of Code Into One: A Fascinating Journey

Sep 28 589 5



Anshul Kummar in Bouncin' and Behavin' Blogs

Why Obsidian Limits Your Brain and What to Use Instead

Why is it holding your brain back, and how can you break free?

Aug 29 1.3K 52

Harsh Chourasia in The Pythoneers

Python vs. R: Choosing the Right Programming Language for Data Science

A Guide to Selecting the Best Tool for the Job

6d ago 204