

Pair-programming with LLMs

Alex van Vorstenbosch

2025-02-01

Google Colab

You can make these exercises using the general [google colab notebook](#). Please take a few minutes to read through the notebook to familiarize yourself with the environment.

1 Exercise: Explaining code

For this exercise, please begin by deciding which coding language you are most proficient in Python or R, so you can work with the other one:

- If you chose R:

Check out the code under `./data/code.py`, and the data under `./data/input-data-script.txt`. Can you figure out what is going on here, what is the code doing? Try asking your LLM what the code is doing.

- If you chose Python:

Check out the code under `./data/code.R`, and the data under `./data/input-data-script.txt`. Can you figure out what is going on here, what is the code doing? Try asking your LLM what the code is doing.

This is an example of a complex task where bigger models tend to work a lot better than smaller models.

Answer

[4 day 4 This code is a solution to a Bingo puzzle on Advent of Code 2021](#)

In the next sections we are going to keep working with this code

1.1 Exercise: Document code

Part of the reason why this code was so hard to read was that it was written without documentation. We're going to fix that. Use your LLM to help you write decent documentation for your file of choice. Adjust the file accordingly, this will be needed for the next section.

1.2 Exercise: Translation between code languages

While it is fun to work in a different language, we prefer to work in the language we know best. Use your LLM to help you translate the code to your language of choice.

Don't blindly run code generated by LLMs, make sure you understand what is going on! For this toy example the risks are very low, but it's best to make this a habit from the start.

Tips:

- Doing this per section of the code makes it easier
- The answer it should generate is giving below, use this as your unit test:

'The winner is board 7 with a score of 53217!'

1.3 Exercise (optional): Generate unit-test

We are going to optimize the code you translated from language A to language B. Here we need to be a bit more careful if nothing breaks by editing the code. Use your LLM to write unit tests for the following:

- Does reading the data give the expected result?
- Does drawing a number do what we expect?
- Does return_winner detect a win?

1.4 Exercise (optional): Optimize the code

Use your LLM to see if there are any parts of your code that can be optimized. I suggest doing this per section to make sure the LLM remains focussed on the relevant parts.

2 Exercise: (Challenge) Generating code

Try to solve part 2 of the puzzle on: <https://adventofcode.com/2021/day/4>

Answer

The answer to the second part of the puzzle for our input should be 2776 with board 34.

Your code should now solve for the puzzle inputs of Advent-of-Code day 4 2021 as a second check.

3 Exercise: (Challenge) Generating code

See if you can also get the solution to the second part for the language that wasn't your first choice!