

Instructions

Working with data is core to computing. Whether you building a website to allow users to send messages to each other (like WhatsApp), trying to find the most efficient bus route between two locations, or building the next new Large Language Model (LLM), you will have to input, process, and analyze large amounts of data.

In this exercise, you'll put together a program that analyzes a collection of data to reveal a hidden insight!

The data is contained in the file "students_study_habits.csv". Are there any insights that you can tell by just reading the file? Maybe a program could help speed up that process and help us gain those insights!

Let's break this down into some smaller steps as we go, that way the problem won't seem so overwhelming and will help you focus on one step at a time. We are trying to build an intuition for what's going on.

Instructions

1. Read the Data

The first step is to get the data from a file into your program so that you can begin to work with it. Data is commonly stored in "CSV" or "comma-separated value" format. To read a file into the program. This code opens up a file in "r" (for reading) mode. It then reads the whole file and splits it into a list of lines. After we are done reading the file, we close it.

2. Aggregate the Data

The code here will take our lines of data and aggregate the parts that we care about apart so that we can make some generalizations.

We will store the data in a "dictionary". Much like a dictionary of words, which maps words to their definitions, a dictionary in Python lets us map words to values. In this case, we are going to map a study habit to the total effectiveness and the number of times it appears.

We will then go through every line of data that we previously read. For each line, we will add the study habit to the dictionary (if it isn't already there), and then we will update the total effectiveness and the count for that study habit.

3. Process the Data

We now have a mapping of a study habit to the total effectiveness across all the students, and the total number of times that the study habit appears. To determine which study habits worked best, we will find the average effectiveness.

So, we will go through each habit, and divide the total effectiveness by the number of students who used the habit. Finally, we will sort it so that the highest average effectiveness comes first and the lowest average effectiveness comes last.

4. Output the Results

For the last step, we are just going to output the top 3 study habits. This time, we will use the `for` statement to print the first three of the sorted habits along with each of their effectiveness.

Reference

1. [Try Kibo Admission Challenge.](#)