Lab Assignment 1: How to Get Yourself Unstuck

DS 6001: Practice and Application of Data Science

Instructions

Please answer the following questions as completely as possible using text, code, and the results of code as needed. Format your answers in a Jupyter notebook. To receive full credit, make sure you address every part of the problem, and make sure your document is formatted in a clean and professional way.

Problem 0

Import the following libraries:

In [1]:
```
import numpy as np
import pandas as pd
import os
import math
```

Problem 1

Python is open-source, and that's beautiful: it means that Python is maintained by a world-wide community of volunteers, that Python develops at the same rate as advancements in science, and that Python is completely free of charge. But one downside of being open-source is that different people design many alternative ways to perform the same task in Python.

Read the following Stack Overflow post: https://stackoverflow.com/questions/11346283/renaming-columns-in-pandas/46912050. The question is simply how to rename the columns of a dataframe using Pandas. Count how many unique different solutions were proposed, and write this number in your lab report. (Hint: the number of solutions is not the number of answers to the posted question.)

Remember: your goal as a data scientist needs to be to process/clean/wrangle/manage data as quickly as possible while still doing it correctly. A big part of that job is knowing how to seek help to find the right answer quickly. Given the number of proposed solutions on this Stack Overflow page, what's the problem with developing a habit of using Google and Stack Overflow as your first source for seeking help? (2 points)

Problem 2

There are several functions implemented in Python to calculate a logarithm. Both the `numpy` and `math` libraries have a `log()` function. Your task in this problem is to calculate $\log_3(7)$ directly (without using the change-of-base formula). Note that this particular log has a base of 3, which is unusual. For this problem:

- Write code to display the docstrings for each function.
- Read the docstrings and explain, in words in your lab report, whether it is possible to use each function to calculate $\log_3(7)$ or not. Why did you come to this conclusion?

If possible, use one or both functions to calculate $\log_3(7)$ and display the output. (2 points)

Problem 3

Open a console window and place it next to your notebook in Jupyter labs. Load the kernel from the notebook into the console, then call up the docstring for the `pd.DataFrame` function. Take a screenshot and include it in your lab report. (To include a locally saved image named `screenshot.jpg`, for example, create a Markdown cell and paste

```
<img src="screenshot.jpg" width=600>
```

(2 points)

Problem 4

Search through the questions on Stack Overflow tagged as Python questions: https://stackoverflow.com/questions/tagged/python. Find a question in which an answerer exhibits passive toxic behavior as defined in this module's notebook. Provide a link, and describe what specific behavior leads you to identify this answer as toxic. (2 points)

Problem 5

Search through the questions on Stack Overflow tagged as Python questions: https://stackoverflow.com/questions/tagged/python. Find a question in which a questioner self-sabotages by asking the question in a way that the community does not appreciate. Provide a link, and describe what the questioner did specifically to annoy the community of answerers. (2 points)

Problem 6

These days there are so many Marvel superheros, but only six superheros count as original Avengers: Hulk, Captain America, Iron Man, Black Widow, Hawkeye, and Thor. I wrote a function, `is_avenger()`, that takes a string as an input. The function looks to see if this string is the name of one of the original six Avengers. If so, it prints that the string is an original Avenger, and if not, it prints that the string is not an original Avenger. Here's the code for the function:

In [2]:
```
def is_avenger(name):
    if name=="Hulk" or "Captain America" or "Iron Man" or "Black Widow" or "
        print(name  + "'s an original Avenger!")
    else:
        print(name + " is NOT an original Avenger.")
```

To test whether this function is working, I pass the names of some original Avengers to the function:

In [3]:
```
is_avenger("Black Widow")
```

```
Black Widow's an original Avenger!
```

In [4]:
```
is_avenger("Iron Man")
```

```
Iron Man's an original Avenger!
```

In [5]:
```
is_avenger("Hulk")
```

```
Hulk's an original Avenger!
```

Looks good! But next, I pass some other strings to the function

In [6]:
```
is_avenger("Spiderman")
```

```
Spiderman's an original Avenger!
```

In [7]:
```
is_avenger("Beyonce")
```

```
Beyonce's an original Avenger!
```

Beyonce is a hero, but she was too busy going on tour to be in the Avengers movie. Also, Spiderman definitely was NOT an original Avenger. It turns out that this function will display that any string we write here is an original Avenger, which is incorrect. To fix this function, let's turn to Stack Overflow.

Part a

The first step to solving a problem using Stack Overflow is to do a comprehensive search of available resources to try to solve the problem. There is a post on Stack Overflow that very specifically solves our problem. Do a Google search and find this post. In your lab report, write the link to this Stack Overflow page, and the search terms you entered into Google to find this page.

Then apply the solution on this Stack Overflow page to fix the `is_avenger()` function, and test the function to confirm that it works as we expect. (2 points)

Part b

Suppose that no Stack Overflow posts yet existed to help us solve this problem. It would be time to consider writing a post ourselves. In your lab report, write a good title for this post. Do NOT copy the title to the posts you found for part a. (Hint: for details on how to write a good title see the slides or https://stackoverflow.com/help/how-to-ask) (2 points)

Part c

One characteristic of a Stack Overflow post that is likely to get good responses is a minimal working example. A minimal working example is code with the following properties:

1. It can be executed on anyone's local machine without needing a data file or a hard-to-get package or module
2. It always produces the problematic output
3. It using as few lines of code as possible, and is written in the simplest way to write that code

Write a minimal working example for this problem. (2 points)

Problem 7

This problem will test your ability to use a chatbot based on a large-language model, such as ChatGPT, to do data wrangling. Please sign-up for a free account to use ChatGPT if you have not already done so, and log on to the chat interface website for ChatGPT.

Part a

The following data comes from a Kaggle project on Jobs and Salaries in Data Science, compiled by Lucas Galanti (though I don't see an attribution to the original data source, so please take these numbers with a grain of salt). Load the data by running this cell:

In [8]:
```
jobs = pd.read_csv('jobs_in_data.csv')
jobs
```

Out[8]:

	work_year	job_title	job_category	salary_currency	salary	salary_in_usd	em
0	2023	Data DevOps Engineer	Data Engineering	EUR	88000	95012	
1	2023	Data Architect	Data Architecture and Modeling	USD	186000	186000	
2	2023	Data Architect	Data Architecture and Modeling	USD	81800	81800	
3	2023	Data Scientist	Data Science and Research	USD	212000	212000	
4	2023	Data Scientist	Data Science and Research	USD	93300	93300	
...	...	...	...	...	...	...	
9350	2021	Data Specialist	Data Management and Strategy	USD	165000	165000	
9351	2020	Data Scientist	Data Science and Research	USD	412000	412000	
9352	2021	Principal Data Scientist	Data Science and Research	USD	151000	151000	
9353	2020	Data Scientist	Data Science and Research	USD	105000	105000	
9354	2020	Business Data Analyst	Data Analysis	USD	100000	100000	

9355 rows × 12 columns

Our goal is to manipulate the `jobs` dataframe to create a table with four rows: one for each of the job titles Data Analyst, Data Engineer, Data Scientist, and Machine Learning Engineer; and two columns: one for the year 2022 and one for 2023. Inside each cell should be the average salary (`salary_in_usd`) for that job title and year, rounded to the nearest dollar. The resulting table should look something like this:

	2022	2023
Data Analyst	108658	110988
Data Engineer	139803	149945
Data Scientist	138529	163714
Machine Learning Engineer	151775	191026

Your task is to use ChatGPT -- and ONLY chatGPT -- to generate Python code that uses `pandas` that can generate a dataframe that looks like the above table. For this problem, use markdown cells in your notebook to display both your prompts and ChatGPT's responses. You will almost certainly need to issue several follow-up prompts to get to an answer, and you should list all of your prompts and the responses in your answer.

A few points to keep in mind:

- You will receive better responses by following the guidelines listed here: https://jkropko.github.io/surfing-the-data-pipeline/ch1.html#method-5-using-a-large-language-model-a-chatbot-to-generate-and-debug-code
- You are more likely to get an answer that works by chunking the task into discrete steps. Some of the steps that are needed here are:
  - Keep only the rows from 2022 and 2023
  - Keep only the rows with job titles Data Analyst, Data Engineer, Data Scientist, and Machine Learning Engineer
  - Collapse the data by taking the average `salary_in_usd` within each remaining year and job title combination
  - Reshape the data by moving the years to the columns
  - Round the average salaries to the nearest dollar

Whether you state these specific steps in your prompts or not, and the order in which you state them if you do, is up to you.

One last note: remember you are trying to generate code to generate the average salary table, not the table itself. If the code that ChatGPT generates yields using a small sample of the data calculates incorrect averages, that's OK as long as the code works properly for the full dataframe. (3 points)

Part b

Having worked through using ChatGPT for data wrangling, take a moment to reflect on when it makes sense and when it doesn't make sense to use ChatGPT for working with data prior to an analysis. Write a short paragraph here summarizing your thoughts. (1 point)