INT3075 Programming and Problem Solving for Mathematics

Project Description

I. Objective

This project aims to provide you with an opportunity to analyse the historical prices of a stock. You are required to calculate the 5-month exponential moving average (EMA) stock prices over a given period of time and determine the best and worst months for the stock in terms of its 5-month EMA prices. To complete this project, you need to apply the knowledge and skills in relation to files, lists and functions.

II. Specification

Data File

You will be given a CSV data file of the historical prices of a stock. This file could be opened by WordPad and data fields in each row are delimited by comma. The data fields include:

- **Date**: the day of trading
- **Open**: the stock price at the beginning of the trading day
- **High**: the highest price the stock achieved on the trading day
- **Low**: the lowest price the stock achieved on the trading day
- **Close**: the stock price at the end of the trading day
- **Adj Close:** the adjusted closing price of the trading day (reflecting the stock's value after accounting for any corporate actions like dividends, stock splits and new stock offerings)
- **Volume**: the total number of shares were traded on the trading day

Functions

Given the file of stock prices, you are asked to develop a Python program to process the data by designing appropriate functions. At minimum you need to implement and call the following three functions:

- **`get_data_list(csv_file_name)`**

  This function has one parameter, namely `csv_file_name`. When the function is called, you need to pass along a CSV file name which is used inside the function to open and read the CSV file. After reading each row, it will be split into a list. The list will then be appended into a main list (a list of lists), namely `data_list`. The `data_list` will be returned at the end of the function.

- **`get_monthly_averages(data_list)`**

  This function has one parameter, namely `data_list`. You need to pass the `data_list` generated by the `get_data_list()` function as the argument to this function and then calculate the monthly average prices of the stock. The average monthly prices are calculated in the following way. Suppose the volume and adjusted closing price of a trading day are V_1 and C_1, respectively. The total sale of that day equals $V_1 \times C_1$. Now, suppose the volume and adjusted closing price of another trading day are V_2 and C_2, respectively. The average of these two trading days is the sum of the total sales divided by the total volume:

  $$\text{Average price} = (V_1 \times C_1 + V_2 \times C_2) \ / \ (V_1 + V_2)$$

  To average a whole month, you need to add up the total sales ($V_1 \times C_1 + V_2 \times C_2 + \ldots + V_n \times C_n$) for each day and divide it by the sum of all volumes ($V_1 + V_2 + \ldots + V_n$) where n is the number of trading days in the month.

  A tuple with 2 items, including the date (year and month only) and the average for that month, will be generated for each month. The tuple for each month will be appended to a main list, namely `monthly_averages_list`. The `monthly_averages_list` will be returned at the end of the function.

- **`get_moving_averages(monthly_averages_list)`**

  This function has one parameter, namely `monthly_averages_list`. You need to pass the `monthly_averages_list` generated by `get_monthly_averages()` as the argument to this function and then calculate the 5-month exponential moving average (EMA) stock prices. In general, the EMA for a particular month can be calculated by the following formula:

  ```
  EMA = (Monthly average price – previous month's EMA) x smoothing constant
        + previous month's EMA
  ```

  where

  ```
  smoothing constant = 2 / (number of time periods in months + 1)
  ```

For example, the following table shows the stock prices between Oct 2020 and Apr 2021:

Month	***Monthly Average Price***
Oct 2020	14
Nov 2020	13
Dec 2020	14
Jan 2021	12
Feb 2021	13
Mar 2021	12
Apr 2021	11

The *initial* 5-month EMA for **Feb 2021** can be calculated by the simple average formula, as shown below:

```
5-month EMA for Feb 2021 = (14 + 13 + 14 + 12 + 13) / 5 = 13.2
```

The 5-month EMA for **Mar 2021** can be calculated by the EMA formula, as shown below:

```
5-month EMA for Mar 2021 = (Monthly average price – previous month's EMA) x
                           smoothing constant + previous month's EMA

                         = (12 – 13.2) x (2 / 6) + 13.2

                         = 12.8
```

The 5-month EMA for **Apr 2021** can be calculated by the EMA formula, as shown below:

```
5-month EMA for Apr 2021 = (Monthly average price – previous month's EMA) x
                           smoothing constant + previous month's EMA

                         = (11 – 12.8) x (2 / 6) + 12.8

                         = 12.2
```

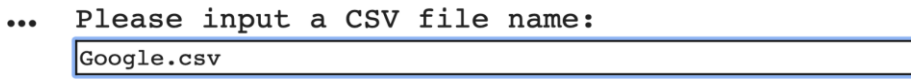
The resulting 5-month EMA stock prices are shown below:

Month	***Average Price***	***5-month EMA Price***
Oct 2020	14	-
Nov 2020	13	-
Dec 2020	14	-
Jan 2021	12	-
Feb 2021	13	13.2
Mar 2021	12	12.8
Apr 2021	11	12.2

A tuple with 2 items, including the date (year and month only) and the 5-month EMA price for that month, will be generated for each month except the first 4 months. Each tuple will be appended to a main list, namely `moving_averages_list`. The `moving_averages_list` will be returned at the end of the function.

Program Input and Output

At the outset, your program needs to ask the user for a CSV file name:



Based on the entered CSV file name, a corresponding output text file (e.g. "`Google_output.txt`" for this case) will be generated. In the output file, you are eventually required to print the best month (with the highest EMA price) and the worst month (with the lowest EMA price) for the stock. You need to first print a header line for the stock, and then print a date (MM-YYYY), a comma followed by a moving average price (in 2 decimal places) on another line. You must follow the output format as shown below (please note the values are not true, which are for reference only):

```
The best month for Google:
01-2007, 123.45

The worst month for Google:
12-2004, 678.90
```

III. Deliverable

You have to include your student name and ID in your source code and name your project solution as "`XXXXXXXX_project.py`" (where `XXXXXXXX` is your 8-digit student ID). Please remember to upload your source code solution to Moodle by the submission deadline.

IV. Evaluation Criteria (40% of Overall Course Assessment)

The project will be graded using the following criteria:

- 15% - Correctness of program execution and output data
- 10% - Modularization (e.g. dividing the program functionality into different functions)
- 5% - Error handling
- 5% - Consistent style (e.g., capitalization, indenting, etc.)
- 5% - Appropriate comments

V. Plagiarism Policy

Your project should represent your own work. **Do not include any code not written by you in your project**. **You are <u>NOT</u> allowed to import any Python libraries in your solution except the modules namely `os`** (https://docs.python.org/3/library/os.html), **`sys`** (https://docs.python.org/3/library/sys.html) and **`csv`** (https://docs.python.org/3/library/csv.html). If cheating is found or the import requirement is violated, you will receive a zero mark.