Python Project - Marvel Mart Project

**Summary**

You have been recently hired by Marvel Mart, one of the world's leading department store chains, to be their new data analyst. You were hired because of your technical skills with Python. Immediately, they offer you a CSV file and ask you to provide specific business analytics based on the data.

You are allowed required to use Python and the numPy and pandas libraries for this project. You may use any other Python libraries that you would like as well.

 Please take a few moments to familiarize yourself with the CSV file called  Marvel\_Mart\_Sales\_Project.csv. Notice the columns, the headings, the format of the data in each column.

Marvel Mart has been providing both online and offline sales of a variety of  products for many years. The provide services to countries all over the world and have stores in many countries. Marvel Mart divides their order up by a

alphabetical priority labeling system:

    C: Critical (most essential to be delivered quickly and accurately)

    H: High

    M: Medium

    L: Low

There are several columns of data for sales:

    Unit Price: money collected for sale of 1 unit

    Unit Cost: money spent for purchase of 1 unit

    Total Revenue: money collected for sale of the collection of units

    Total Cost: money spent for purchase of the collection of units

    Total Profit: Total Cost - Total Revenue (profit)

The rest of the columns should be self-explanatory.

**Deliverables**

When I run your script, I expect the output to make sense. You may have print statements which print things - I hope you do. But I want when I look at it to know what I am looking at so you should always print a line explaining what the output is. You are not REQUIRED to have output when I run the script other than the CSV and TEXT files I am requiring.

When you submit your project, it should contain these below. **Do not zip these files and submit.**

1. Python script of complete code. Submit as .ipynb format.
2. PDF or HTML version of the .ipynb file
3. Marvel\_Mart\_Sales\_clean.csv
4. Marvel\_Mart\_Rankings.txt
5. Marvel\_Mart\_Calc.txt
6. Countries\_By\_Region.csv
7. Documentation surrounding your design process. Please submit this as a Word document.

**Organizational Guidelines**

Please organize your script as follows:

* Introductory comment section
* Import statements
* Comments indicating the start of each Part
* Comments indicating what code the question number goes with 1(A), 2(A)... for Part 2 #2, you are free to just have all the code under a comment that says the number and not the letters (since each letter can be done in one line if you're savvy!)
* Comment where you think its appropriate to explain your code. Including Markdown sections is recommended.

Example of Intro Markdown section heading and markdown to divide sections:

Python Project - Marvel Mart Project  
Eric Lloyd  
(Date)  
  
(import statements in code block)  
  
Part 1: Cleaning the data  
  
...  
  
Part 2: General Statistics  
  
1 (A)  
  
...

**The Report Questions**

Part 1: Cleaning the Data

It came to our attention that some of the data was either incorrectly entered or missing entirely! This is going to throw off our calculations if it is left unchecked. We were grateful to discover none of it happened in our accounting columns of the data for that would be very detrimental but we aren't sure where the missing/incorrect data is elsewhere. Here's what we do know to help your investigation into finding the missing/incorrect data. Of the columns that we have, we know the missing/incorrect data comes from these columns:

* Country   (either missing OR will be a number as a string)
* Item Type   (either missing OR won't be a valid Item Type from the other ones listed)
* Order Priority   (either missing OR won't be a valid priority code of 'C', 'H', 'M', 'L', or 'NULL')
* Order ID   (either missing OR won't be a number)  
    
  If you find incorrect/missing data and its text type for that column, change it to the string "NULL". If you find incorrect/missing data and its a number type for that column, change it to 0. (or 0.0 if its a float). Test for missing values FIRST then if you find the ones that are missing, you don't have to test those for incorrect values. **You need to change the values, then rewrite to a new CSV file called Marvel\_Mart\_Sales\_clean.csv so it can be used later with the correct values.**

*(Note from Instructor: While it is possible and acceptable to produce a non-pandas using solution, I would suggest that you use pandas Dataframes for this. It makes a difficult job  much easier. The guide to how to do that is here and some other resources:*

* [*https://www.geeksforgeeks.org/working-with-missing-data-in-pandas/ (Links to an external site.)*](https://www.geeksforgeeks.org/working-with-missing-data-in-pandas/)
* [*https://www.geeksforgeeks.org/python-read-csv-using-pandas-read\_csv/ (Links to an external site.)*](https://www.geeksforgeeks.org/python-read-csv-using-pandas-read_csv/)
* [*https://www.geeksforgeeks.org/saving-a-pandas-dataframe-as-a-csv/ (Links to an external site.)*](https://www.geeksforgeeks.org/saving-a-pandas-dataframe-as-a-csv/)

***You will need to google things to get this done. Part of being a programmer is knowing how to find solutions on the internet and adapting them.****)*

Part 2: General Statistics

First, we would like you to get us some general statistics from the data. I suggest you create a dictionary of lists with the keys being the heads of the CSV file (or columns of clean DataFrame) and the list attached to it being all of the values for that heading. Duplicates should be included here. This is only a suggestion. If you want to just keep using the DataFrame you are welcome to.

1. Produce the following and write it to a text file called **Marvel\_Mart\_Rankings.txt**. Be sure to use append so that you can append data rather than writing over top of the previous data. Be sure to include a newline between each append to the file. When writing to the file, please output in a text form such as: (note you are getting a count of the number of sale transactions here not the sum of the total sales! To be clear: you want a count of the number of sale transactions. Do not sum up the Units Sold. Do not use sum at all. You are getting the count of the number of transactions done per country)  
     
   Countries Most Sale Transactions:  
   Country 1: (number of sales transactions)  
   Country 2: (number of sales transactions)  
   ...  
   (Answer question) "The country we should build our shipping center is \_\_\_\_\_\_ because \_\_\_\_..."   
     
   (A) We want to know which countries we sell the most to so we can pick a new location to build a shipping center. Rank the Top 10 countries we sell to the most to least along with the number of sales we've had with that country. We have shipping centers in Trinidad and Tobago, Guinea, and Maldives right now. Which country should we build a shipping center in based on most sales and lack of shipping center? Please justify your reasoning.  
     
   (B) Provide a count for how many online and offline orders we take. Which do we take more of? You do not need to justify your answer.  
     
   (C) Rank the top 3 years we did the most sales (brought in most profit) in to the least sales. (Use 'Total Profit' to determine this) (Just the years, not the whole dates).  Use the Order Date, not the Ship Date. Please list the years and the amount sold. Answer the question "Which year did we sell the most in?"  
     
   *(Note from instructor: doing large number sums with floats in Python usually produces scientific notation but we don't want that. You can turn that off by putting the following line under the import statements at the start of the script:* pd.set\_option('display.float\_format', lambda x: '%.3f' % x)
2. Now you will save these calculations below to a text file called **Marvel\_Mart\_Calc.txt**. When writing to the file, please output format such as:  
     
   Sums:  
   Units Sold: (Number)  
   Unit Cost: (Number)  
   Total Revenue: (Number)  
   Total Cost: (Number)  
   Total Profit: (Number)  
   (Newline)  
   Averages:  
   Units Sold: (Number)  
   ...  
   (Newline)  
   Maximums:  
   Units Sold:  
   ...  
     
   (A) Produce the data above for the sum of each one.  
   (B) Produce the data above for the average of each one. (Average Units Sold, Average Cost, etc)  
   (C) Produce the data above for the maximum of each one. (Max Units Sold, Max Cost, etc.)

Part 3: Cross-Reference Statistics

For this part you will be cross-referencing the data in the CSV file and the getting an output and writing it to a new CSV file.

1. We need you to get a list of the Regions and then the countries we sell to in that region. Please be sure no duplicates Regions or countries exist. Please return this as a dictionary of lists with the keys of the dictionary being the name of each Region and the list attached to that being all the countries we sell to for that region. You may also return it as a Series of Lists (although I found that to be harder). Finally, if you want to use an alternate method with pandas dataframes, that will be accepted as well. Be sure your output is easy to read and your code makes sense.  
   Write this out to a CSV file called Countries\_By\_Region.csv.  
     
   **(Be careful here as there is a header row when you convert the csv to a dictionary. If you end up getting the header row in your final result, just remove it. You are free to do it however you want as long as in the end its a dictionary of lists (or Series of Lists), the keys being the Regions and then the list for that key being the countries that is sold to, with no duplicates. And then print it to the csv file.)**

**Your CSV file should look like below. Order of Regions and Countries is unimportant but structure should be the same:**

**Graphical user interface, application, table

Description automatically generated**

**Grading Rubric**

The Process:

* Does it run?
  + Before anything is graded in this project, the first test is ***DOES IT RUN?***
    - Yes, it runs without an exception stopping it: *Continue grading*
    - No, it produces an exception and does not finish running: *Stop grading. Deliver grade of 0. Contact student for follow up.*

**Please ensure your code runs without an exception that stops it. Its better to submit *wrong* code that runs than any code that does not run.**

Okay, now that is out of the way...

(The grading rubric itself is at the bottom of this document. Please use it when referring to this explanation. )

* Now, we determine Max points. This is the number of points that you can receive as a maximum after it is calculated how many of the task items in the project were completed.
* Max points is determined by a basic percentage analysis of the number of task items you have written. So if you have all the tasks completed, that is a Max Points balance of 100 pts. Note, if any of them produce an exception, see the first few sentences of this section. Please be aware, the first task is cleaning the data. If you do not do that properly, ALL of the rest of the tasks will end up wrong.

Example:

If you have 7 of the 8 tasks done, you have 88% and so you have a Max Points balance of 88 points. This means the maximum grade you can get on the assignment is 88%. I take that amount and then use the rubric below to calculate your grade.

Your Max Points for each part is:

* Accuracy of CSV Files Created: 40% of 88 points = 35.2 possible points for this part.
* Code Elegance: 35% of 88 points = 30.8 possible points for this part.
* Organization and Readability of Script: 25% of 88 points = 22 possible points for this part.

Let's say you got these results:

* Accuracy of CSV Files Created: 82%.  82% of the 35.2 possible points for this part is 28.9 points.
* Code Elegance: 90%. 90% of the 30.8 possible points for this part is 27.7 points.
* Organization and Readability of the Script: 75%. 75% of the 22 possible points for this part is 16.5 points.

So, 28.9 + 27.7 + 16.5 = 73% for the final grade.