Project 2: Marvel Mart
3/16/2023

96/100 Points

Attempt 1	∨	◯	**Review Feedback**

Review Feedback
3/15/2023

Attempt 1 Score:
96/100

🗨 View Feedback

Anonymous Grading: **no**

Unlimited Attempts Allowed
1/8/2023 to 3/16/2023

∨ **Details**

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 y
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

Please take a few moments to familiarize yourself with the CSV file called **MM_Sales.csv (https://seattleu.instructure.com/courses/1606759/files/68821274?**
(https://seattleu.instructure.com/courses/1606759/files/68821274/download?download_frd=1) Notice the columns, the headings, the format of the data in each

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 ar
countries. Marvel Mart divides their order up by an 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 k
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 file

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. MM_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. Documentation will just be an explanation in your own words
   create the Python script. Start with how to figured out to do the data cleaning and write about each part.

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)...
- 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

Outline

Part 1: Cleaning the Data

Part 2: Exploratory Data Analysis with Reports & Visualizations

1. Country Rankings

2. Count of Sales Channels & Order Priorities

3. Profits by Item Type

4. Descriptive Statistics

Part 3: Cross-referencing the Data

Part 1: Cleaning the Data

"It came to our attention that some of the data were either incorrectly entered or missing entirely! This is going to throw off our calculations if it is left unchec 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 do know to help your investigation to find the missing/incorrect data. Of the columns that we have, we know the missing/incorrect data comes from these co

- Country   (either missing AND/OR will be a number as a string)
- Item Type   (either missing AND/OR won't be a valid Item Type from the other ones listed)
- Order Priority   (either missing AND/OR won't be a valid priority code of 'C', 'H', 'M', 'L', or 'NULL')
- Order ID   (either missing AND/OR won't be a number)

Note: invalid Item Type in this case will have only one instance of that invalid Item Type. Usually, a list of valid item types would be supplied.

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).

Once you find all the incorrect and missing values and replace them with "NULL" or 0, then you need to remove all rows that have been altered. You need to change the values, then rewrite to a new CSV file called MM_Sales_clean.csv so it can be used later with the correct values.

Hints from Instructor:

1. Test for missing values FIRST then if you find the ones that are missing, you don't have to test those for incorrect values.
2. 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 li statements at the start of the script: `pd.set_option('display.float_format', lambda x: '%.3f' % x`
3. 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 difficul to how to do that is here and some other resources:

- *https://www.geeksforgeeks.org/working-with-missing-data-in-pandas/* ➦ *(https://www.geeksforgeeks.org/working-with-missing-data-in-pandas/)*
- *https://www.geeksforgeeks.org/python-read-csv-using-pandas-read_csv/* ➦ *(https://www.geeksforgeeks.org/python-read-csv-using-pandas-read_cs*
- *https://www.geeksforgeeks.org/saving-a-pandas-dataframe-as-a-csv/* ➦ *(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: Exploratory Data Analysis with Reports & Visualizations

"First, we would like you to get us some general statistics from the data."

1. "We want to know which countries we sell the most so we can pick a new location to build a shipping center. Rank the Top 10 countries we sell to the m
   number of sales we've had with that country." *(note you are getting a count of the number of sale transactions here not the sum of the total sales)*
   1. **Use Seaborn or Matplotlib to create a chart of your choice showing these top 10 values by country.**
      1. We have shipping centers in Trinidad and Tobago, Guinea, and Maldives right now. Which country should we build a shipping center in based on
         shipping center? When I say most sales, I'm talking about the most sales transactions, not the most sales amount. A sales transaction is represe
         your reasoning.
   2. **Write the results to a text file called MM_Rankings.txt.**
      1. Be sure to use append so that you can append data rather than writing over top of the previous data.
      2. Include a newline between each append to the file.
      3. When writing to the file, please output in a text form such as:

         Countries Most Sale Transactions:
         (Country Name): (number of sales transactions)
         (Country Name): (number of sales transactions)
         ...
         (Answer question) "The country we should build our shipping center is _____ because _____..."

2. "Now we will need you to determine how many online and offline orders that our company takes. Also, if you could let us know the count of the different
   would be great. Please show us this in a pie chart format."
   1. **Determine the count for how many online and offline orders we take.**
   2. **Determine the count of the different Order Priority types.**
   3. **Create a pie chart for each showing the differences in values (use Seaborn or Matplotlib).**
   4. **Add the results of the sales channel types and the order priorities to the file MM_Rankings.txt.**
      1. Be sure to use append so that you can append data rather than writing over top of the previous data.
      2. Include a newline between each append to the file.
      3. When writing to the file, please output in a text form such as:

         Sales Channels:
         Online: ####
         Offline: ####
         We do more online/offline sales.

         Order Priorities:
         L: ###
         M: ###
         H: ###
         C: ###
         We do more L/M/H/C order priorities.

3. "For our next section, we will need you to give us an idea of how well our Item Types are producing profits for us. At the end, report to us which 3 item ty
   profit."
   1. **Create a Boxplot using Seaborn showing the Total Profits DISTRIBUTION by Item Type.**
   2. **Use Python to determine the sum of Total Profit by Item Type.**
   3. **Now create a chart type of your choice (Seaborn or Matplotlib) showing the sums of the different Item Types.**
   4. **Now, using Python, rank the top 3 item types we did the most sales (brought in most profit) in to the least sales. (Use 'Total Profit' to deter
      item types and the amount of profit made from sales.**
   5. **Add the results of the top 3 item types to the file MM_Rankings.txt.**
      1. Be sure to use append so that you can append data rather than writing over top of the previous data.
      2. Include a newline between each append to the file.
      3. When writing to the file, please output in a text form such as:

Highest Selling Items:
Item 1: ####
Item 2: ####
Item 3: ####
We profited from _____ the most.

6. **Provide a markdown section discussing the results of the boxplots. Discuss what is being shown in the boxplots amd do some business of use this sort of chart might help in making decisions. Are there any unexpected results? Discuss them.**

4. "Finally, we need you to determine some descriptive statistics for us. Please determine the sum, average and maximum values for the Units Sold, Unit Cost and Total Profit. Please put this in a report."
   1. **Produce the data above for the sum of the requested columns.**
   2. **Produce the data above for the average of the requested columns. (Average Units Sold, Average Cost, etc)**
   3. **Produce the data above for the maximum of the requested columns. (Max Units Sold, Max Cost, etc.)**
   4. **Create two line plots using Seaborn or Matplotlib, one for the sums and one for both the averages and the maximums. DO NOT INCLUDE COST.**
   5. **Now you will save these calculations below to a text file called MM_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)

Averages:
Units Sold: (Number)
...

Maximums:
Units Sold:
...

Part 3: Cross-Reference Statistics

"We are in desperate need of a concise list of the Regions we sell to with the Countries that are located in each one."

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. **Please 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**.
   1. (Does not have to be done this way)
      1. Non-pandas Solution advice: Please return this as a dictionary of lists with the keys of the dictionary being the name of each Region and the list countries we sell to for that region. You may also return it as a Series of Lists (although I found that to be harder).
      2. 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 yo
2. **Write this out to a CSV file called Countries_By_Region.csv.**
   1. (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 remo 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 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. Here are the counts for t

North America: 4, Europe: 48, Asia: 27, Australia & Oceania: 15, Central America & The Carribean: 20, Sub-Saharan Africa: 48, Middle East and N

North America	Europe	Asia	Australia and Oceania	Central America and the Caribbean	Sub-Saharan Africa
Mexico	Bosnia and Herzegovina	Sri Lanka	Vanuatu	Grenada	Ghana
Canada	Armenia	Mongolia	Palau	Antigua and Barbuda	Swaziland
United States of America	Germany	Malaysia	East Timor	Dominica	Burkina Faso
Greenland	Sweden	Singapore	Federated States of Micronesia	Nicaragua	Kenya
	Slovenia	Brunei	Marshall Islands	Belize	Uganda
	Belarus	Kyrgyzstan	Kiribati	Saint Kitts and Nevis	Zimbabwe
	Ireland	Taiwan	New Zealand	El Salvador	Cote d'Ivoire
	Monaco	North Korea	Nauru	Costa Rica	Republic of the Congo
	Estonia	Cambodia	Solomon Islands	Barbados	Djibouti
	Czech Republic	Tajikistan	Australia	Cuba	Cameroon
	Slovakia	Turkmenistan	Fiji	Honduras	Guinea
	Moldova	Maldives	Papua New Guinea	Jamaica	Nigeria
	Montenegro	Laos	Tonga	Saint Lucia	Eritrea
	Malta	Myanmar	Tuvalu	Dominican Republic	Equatorial Guinea
	Macedonia	Bhutan	Samoa	Haiti	Mali
	Vatican City	Vietnam		Guatemala	Togo
	Netherlands	Japan		The Bahamas	Chad
	Ukraine	Philippines		Trinidad and Tobago	South Sudan
	Greece	China		Saint Vincent and the Grenadines	Mozambique
	France	Bangladesh		Panama	Sierra Leone
	Liechtenstein	Uzbekistan			South Africa
	Cyprus	Thailand			Guinea-Bissau
	Spain	Kazakhstan			Cape Verde
	Russia	South Korea			Sao Tome and Principe
	Andorra	Nepal			Tanzania
	San Marino	India			Sudan
	Kosovo	Indonesia			Botswana
	Hungary				Niger

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. It's better to submit *wrong* code that runs than any code that does not run.

The task items to be graded are broken down as: (total of 23 tasks)

Part 1

4 Data Cleaning: Country, Item Type, Order Priority, Order ID

Part 2

1.1, 1.2, 2.1, 2.2, 2.3, 2.4, 3.1, 3.2, 3.3, 3.4, 3.5, 3.6, 4.1, 4.2, 4.3, 4.4, 4.5,

Part 3

1.1, 1.2

⌄ **View Rubric**

Select Grader

Geetha Sree Kariyavula Narasimha   ⌄

Marvel Mart Project

Criteria	Ratings									Pts
Accuracy of Code [view longer description]	**50 to >45 pts** **Superior** All tasks is accurate.	**45 to >40 pts** **Awesome** At least 21 of the 23 tasks are accurate.	**40 to >35 pts** **Great** At least 19 of the 23 tasks are accurate.	**35 to >30 pts** **Good** At least 17 of the 23 tasks are accurate.	**30 to >25 pts** **Over Half** **Accurate** At least 15 of the 23 tasks are accurate.	**25 to >20 pts** **Half** **Accurate** At least 12 of the 23 tasks are accurate.	**20 to >15 pts** **Poor** At least 10 of the 23 tasks are accurate.	**15 to >10 pts** **Very Poor** Less than 7 of 23 tasks are accurate.	**10 to >0 pts** **Failing** Less than 4 of 23 tasks are accurate.	50 / 50
Code Elegance [view longer description]	**20 to >18 pts** **Superior** Minimal code repetition. Use of functions in code		**18 to >16 pts** **Good** Minimal code repetition. Use of functions in code		**16 to >14 pts** **Adequate** Minimal code repetition. Code is adequately		**14 to >12 pts** **Poor** Redundant code; repetition. Code has		**12 to >0 pts** **Failing** Massive redundant code repetition. Lack of code	18 / 20

Marvel Mart Project

Criteria	Ratings					Pts
	AND use of Numpy and/or Pandas.	OR use of Numpy and/or Pandas.	elegant. Numpy/Pandas either not used, used very infrequently or incorrect. Same for Functions. Redundant code exists.	little elegance. Loops not used well.	elegance completely. (Avoiding using loops completely)	
Organization and Readability of Script view longer description	**20 to >18 pts** **Superior** Script follows all organization guidelines. Variables have meaningful names with camel case method. White space used to make script vey easy to read.	**18 to >16 pts** **Good** Script follows all organization guidelines. Variables have meaningful names. White space used to make script easy to read.	**16 to >14 pts** **Adequate** Script follows all organization guidelines in assignment. Script is readable.	**14 to >12 pts** **Poor** Script follows some of the organization guidelines in assignment. Script is difficult to read.	**12 to >0 pts** **Failing** Script follows little to none of the organization guidelines in assignment. Script is almost impossible to read.	18 / 20
Documentation view longer description	**10 to >9 pts** **Superior** Documentation details each part of the program. It is well-organized and easy to read. It provides thoughtful reflections on the development of the code and its usefulness.	**9 to >8 pts** **Good** Documentation details each part of the program. It isn't difficult to read and provides some thoughtful reflection on the purpose of the code.	**8 to >7 pts** **Adequate** Documentation details each part of the program.	**7 to >0 pts** **Poor** Documentation details some of the program.	**0 pts** **Failing** No documentation present.	10 / 10

File Name	Size	
‹ › Nguyen_Project2.html	1.1 MB	✓
Nguyen_Pr...tion.docx	769 KB	✓
Nguyen_Project2.ipynb	456 KB	✓
Countries...egion.csv	2.11 KB	✓
MM_Ranking.txt	656 Bytes	✓
MM_Sales_clean.csv	5.99 MB	✓
MM_Calc.txt	538 Bytes	✓
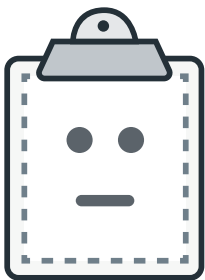
Preview Unavailable

Nguyen_Project2.html

⤓ Download
(https://seattleu.instructure.com/files/69202445/download?download_frd=1&verifier=gsRXKyiK65fg9S6MYGuZsMUsb8eUOrWjQM93rpzT)

‹
(https://seattleu.instructure.com/courses/1606759/modules/items/17700464)

›
(https://seattleu.instructure.com/courses/1606759/modules/items/17700467)

You are unable to submit to this assignment as your enrollment in this course has been concluded.