Python Project - Marvel Mart Project

Tracy Tran

Documentation

After reading the original file into the data frame via csv library and pandas library, I made a copy of the original data frame to work.

My initial step to filter out the errored data was to use the “.ISNULL()” method, after sorting out the missing data, I tried to replace them by using “.FILLNA()” . However,it worked only for the missing value while the data set has more errors than that. After learning the method in class, I tried and succeeded. This method is more convenient than the previous method because it utilizes TRY...EXCEPT in the FOR...LOOP to both clean and replace missing values at the same time. When I finished cleaning, I wrote the cleaned data frame to a csv file called MM\_Sales\_Clean.csv. The result file still had the column of number index, so I tried to set the index=False to avoid the automatically created number index in the result file.

For Part 2, as Python has the GROUPBY function which works quite similarly to the one in SQL, I used .GROUPBY() together with .COUNT() to group the counted Order ID by Country, and then use the NLARGEST (n,column) to filter only the top 10 results. I decided to count Order ID to find the countries with most sale transactions because each Order ID is a unique transaction, I don’t count the Total Profit or Total Revenue are the total earning of each transaction.

For the preparation step before visualization, I created a Pandas data frame from the above result. To visualize the top 10 countries with most sales, I used the barplot using Seaborn and Matplotlib with the horizontal axis as the Country name and the vertical axis as its Number of Transactions. However I realized the graph would be easier to read if I switched the horizontal and vertical axis order. Then, I made the vertical axis as the Country name and the horizontal axis as its Number of Transactions.

To write the result to a text file, I created a file called MM\_Ranking, and then append lines in order not to overwrite the previous lines.

In the process of determining how many online and offline orders as well as the count of Order Priority types, I decided to use the same method when I worked on question 1, using .GROUPBY() and .COUNT(). I grouped the counted Order IDs by Sales Channel to determine how many online and offline orders we had and grouped the counted Order IDs by Order Priority types to see how many orders we have for each type of Order Priority.

I created two data frames for the two above results to plot pie charts. I used Matplotlib to plot pie charts. I set autopct with %1.1f%% while plotting to display the percent value with 1 decimal places.

To add the information about the Sales Channels take-up and the Order Priority take-up to MM\_Ranking, I used the same appending method as the previous step.

In order to visualize the Profit Distribution by Item Types, I used Seaborn to create a boxplot. Due to the huge numeric values, I decided to add .YSCALE(‘LOG’) to put the y axis in logarithmic scale. Because the number we are working with is huge,not adjusting the scale could lead to the fact that the plot would be distorted.

In the process of determining the sum of Total Profit by Item Types, I used .GROUPBY() together with .SUM(): I calculated the sum of Total Profit by Item types.

For the visualization of the sums of Profit of different Item Types. I decided to graph a barchart using Matplotlib. Before graphing, my preparation step is to create a dictionary in which the key is Item Types and the value is Sum of Total Profit. After that I created a pandas data frame out of the dictionary. When plotting, I also add the YSALCE('LOG’) so the bar chart wouldn’t be distorted due to huge data values of the y axis.

To rank the top 3 item types we did the most sales (brought in most profit) using Total Profit, I group the sum of Total Profit by Item Types and added the .N.LARGEST(3,column).

To write to MM\_Ranking, I used the same mentioned method in the previous steps.

In the process of determining the sum, average and maximum values for the Units Sold, Unit Cost, Total Revenue, Total Cost and Total Profit, I simply use .(SUM), .(MEAN) and .(MAX) function.

When printing each result, I added .ROUND() to round it to 3 decimal places.

To create and write the result to a text file called MM\_Calc, I simply used the previous method.

I used Matplotlib to plot the line plots for the sum and for the averages and maximums. Before plotting the two line plots , I created a pandas data frame for the Sum of each requested column (Units Sold, Unit Cost, Total Revenue, Total Cost and Total Profit) and a data frame containing the Average and Maximum for those columns. In order to prevent distorted plots, I added .YSCALE(‘LOG’).

In Part 3, I grouped the Country by Region using .GROUPBY(). However, I added .UNIQUE() to make sure there are no duplicates Regions or Countries exist. I also created a dictionary from the result for later use. For the last step, I created a pandas data frame from the previous dictionary and then wrote the data frame to a csv file called Countries\_By\_Region.csv. I set the index=False to avoid the automatically created number index in the result file.