Documentation

Python Project – Marvel Mart

[Part 1]:

For part one, I used a variety of methods to clean each column. For 'Country', I used a try-except block, to convert values to floats. Then I pulled the indexes of all countries that could be converted to floats and replaced the value at that index with 'NULL'.

For 'Item Type' and 'Order Priority', I used the fillna method. On 'Order Priority' I created a frequency distribution in order to see whether there were any outliers indicating an invalid priority code. Since the only problem was empty data, I commented it out.

For the 'Order ID' column, I applied the same umbrella strategy as I did for the countries. I pulled the indexes of all values that were not numbers and replaced the values at those indexes with 'NULL'.

[Part 2]:

First step is solving part two was to follow the suggestion on Canvas. To reiterate, I created a dictionary with the data frame headers as keys and the column data (as a list) as the values.

For 2.1.A and 2.1.B, I kept using the original data frame because it was the easiest method I could find. For each part (A and B), I made a pandas series with "df['COLUMN_NAME'].value_counts()". This function returned the number of occurrences for each unique row value. And from there I simply renamed the columns.

2.1.C required that I use the dictionary created in the first step. I made two empty lists, one for years and one for profits. I then sliced the values in 'Order Date' to return only the year part. And finally, appended the years to the list years. It was the same process for 'Total Profit', except I did not have to slice. After, I created a new data frame with the two lists, renamed the columns, grouped by 'year' and summed, sorted the profits in descending order, and the reset the index to get proper columns/column names.

After all the rankings were completed, I appended and formatted each to a new text file called, 'Marvel_Mart_Rankings.txt'. I found using 'w+' to be easiest when testing the entire program repeatedly. Also, 'a+' would have caused redundancies in the code.

[Part 3]:

Boy was part three a drag! That being said, I enjoyed the challenge, and nothing can beat the feeling of completing an entire script and having it work flawlessly (especially after days of trials and errors). I tried so many different strategies for this part. I ended up sticking with pandas data frames for most of it, and then once I found the right piece, putting the puzzle together was easy as pie! After our email conversation (Dr. Lloyd), I realized that while my lists of countries were one long string each, it still meant that the region and country series were the same length (7 elements each). All I had to do was convert each series to a separate list, and then zip the two lists into a dictionary. The final step was to convert the dictionary back to a data frame, and thanks to Dr. Lloyd I was able to do that rather quickly, along with transposing it for the proper format before exporting it to a CSV file.