Documentation

PYTHON PROJECT-MARVEL MART

Veng Leap Chan

March 13th, 2022

Part 1. Cleaning the Data

First, I imported the following libraries: numpy, pandas, seaborn, matplotlib, and warnings.

I read the MM_Sales.csv using pandas as a data frame and saved it as MM. I then checked for how many rows and columns it has. We found out that it had 50,000 rows and 14 columns.

I start the data cleaning process by first checking which columns are missing any data. For instance, I used MM.isna().sum() to find the total number of missing data in each column. I then created a clean copy of MM_Sales.csv called 'MMClean' to alter because we cannot reverse the changes to the actual data, but if we make a copy, we can always create a new copy if something goes wrong.

We found that Item Type was missing 6 data points and Order Priority was missing 15 data points. I began to fill them with 'NULL' as they are both text-type columns.

Then, using the information given, I knew that the Country, Item Type, Order Priority, and Order ID might have missing/incorrect data to start my investigation. As we have found the missing data and filled it already, I will be finding the incorrect data.

First, I looked at the Country column. We saw that the Country column is stated as an object type, which is a string. Numbers can also be strings, as such, I had to loop through the data points in the Country column to try to convert them into a number. If it had been converted successfully, then I would change it to 'NULL'. Through this, we found and successfully converted 3 numbers into 'NULL'. I also checked to see if there is any data that has not been converted, but the number has been successfully converted.

Then, I checked the Item Type column for invalid Item Types. Invalid Item Types are considered as Item Types that only have one data or only have one instance in our data. As such, I used the group by to group all the different Item Types and the count function to see if there are any Item Types that only appeared once. Through my observation, no such Item Type appeared, meaning that there are no invalid item types.

Moving on, I checked the Order Priority column for invalid priority codes to change them into 'NULL'. Valid priority codes are defined as 'C', 'H', 'M', 'L', or 'NULL'. As such, I iterated through all the data in the Order Priority column and checked if there are any items not considered valid priority codes and would change them to null if it is identified. After running this, we did not identify any data points that are considered invalid priority codes.

Finally, I checked for invalid Order IDs, meaning I tried to convert all data points in the Order ID column, looping through each one and trying to convert them into integers. Any data point that does not pass this will be printed and changed to a 0. We found that 5 data points in Order ID were erroneous, and we changed them to 0. To confirm that all the erroneous data points were

changed, we changed the Order ID column and its data points to integers, and it was successful, meaning, Order ID only contained numbers.

To end, we deleted the rows with 0 and 'NULL', and we saved MMClean as a CSV file called 'MM_Sales_clean.csv'.

Part 2. Exploratory Data Analysis with Reports and Visualizations

1) **Country Rankings:**

   We had to use Seaborn or Matplotlib to make a chart to show the top 10 countries based on Sales Transactions. I decided to use a Seaborn bar chart as I felt as though it would show the difference between each country most clearly.

   To start, I found the top 10 countries by sales transactions and saved it to the Top10 variable. I did this by using the group by function, the count function, and the nlargest function. I grouped by country and counted the number that each country would appear in the Country column and got the top 10 countries that had the highest count.

   I created the bar plot using seaborn with the name of the top 10 countries as the x-label, the count of sales transactions of the top 10 countries as the y-variables.
   I also created a NTop10 variable to see the country rankings that only included countries that did not have any shipping center.

   We had to write the results into a text file named 'MM_Rankings.txt'. I used 'w+' to create the new text file. I wrote in the top 10 countries and their number of sales transactions, in order from most to least in the text file. I also wrote my reasons for why we should build a shipping Center in Cape Verde.

2) **Count of Sales Channels & Order Priorities:**

   First, I wanted to find the count of online and offline orders we take. So, I took a group by of the Sales Channel. I essentially grouped by Online or Offline and then I counted how many times each of them appeared in the Sales Channel column. I saved the results to the variable called 'OnOff'. We saw that Online sales are more prominent than Offline sales.

   Then, I wanted to find the count of each order priority. Similar to the task above, I grouped each order priority code, and then found their count of how many each order priority appeared in the Order Priority column. This shows how many counts each order priority has. I saved it to a variable called 'PrioType'.

I created a pie chart using matplotlib to show the differences in values for Online and Offline sales in a pie chart called ‘Sales Channel’. I, then, created a pie chart using matplotlib to show the difference in values for the different types of Order Priority in a pie chart called ‘Order Priorities’. I also added legends to both pie charts. The pie chart was able to show

I appended the results of the count for each method of Sales Channel, and I pointed out which method of sales is more frequent. I also appended the results of the count for each order priority, as well as exclaiming which order priority has the most count.

3) **Profits by Item Type:**

First, we had to make a box plot and I chose to use Seaborn to make the box plot. I used it to show Total Profits by Item Type. As such, we see that a box plot for each Item Type was created to show the profit distribution for that item type. Surprisingly, we saw that fruits profit distribution is very small and it looks like we need to investigate it more.

Then, we had to use Python to determine the sum of Total Profit for each Item Type. As such, I created a variable called summation which holds the total profit column grouped by each item type and I summed the total profit for each item type together using the sum function from pandas.

I also created a bar chart called ‘Sum of Profit for each Item Type’. I thought bar charts would be the best option as they can really illustrate the difference in sum of total profit for each item type. Next, I created that bar chart using seaborn, and I had the different item types on the x-axis and the sum of profits as the y-axis. Through this, we can really see the difference in profit each item type makes.

We needed to find the top 3 item types based on the sum of the total profit. I remembered that I discovered the nlargest(n) function, which shows only the top n number of items. So, I used that function to find the top 3 item types based on the sum of the total profits. I found Cosmetics, Household, and Office Supplies were in the top 3 item types, Cosmetics being the item type that brings in the most profit.

Finally, I appended the top 3 item types and the sum of its total profit to ‘MM_Rankings.txt’.

4) **Descriptive Statistics:**

We were asked to determine the sum, average, and maximums of Units Sold, Unit Cost, Total Revenue, Total Cost, and Total Profit. I know that it would be a lot of repetitive

code if I were to determine the sum for each column, then average for each column, and maximums for each column. I know I can determine the sum, average, and maximums for every column, but I do not know how to limit it to only a few columns, so I had to find another way. I plan to find the sum, average, and maximums for every column and using a function, sift through all the information and only keep the statistics from the 5 columns above. I did that by creating a function called seriesMaker, which iterates through the series generated from finding the sum, average, and maximums, and creates a dictionary, with only the 5 selected columns. Then, I turned it back into a series. I save the results into 3 different variables called sums, means, and maxes.

Then, I had to create a line plot but without Units Sold and Unit Cost. As such, I had to create another function to drop these two columns from the 3 variables and I saved them to new variables: Nsums, Nmaxes, and Nmeans.

Using the Nsums variable, I create a line plot using Seaborn, called 'Sums', which shows the sums of Total Revenue, Total Cost, and Total Profit. In another line plot called 'Maximums and Averages', I had two lines, one representing the Average, and the other representing the maximums, of Total Revenue, Total Cost, and Total Profit.

Finally, I created a new text file called 'MM_Calc.txt', writing in sums, averages, and maxes for Units Sold, Unit Cost, Total Revenue, Total Cost, and Total Profit.

Part 3. Cross-Reference Statistics:

For part 3, I grouped countries by region and used the unique function. I saved the results to the variable regions. I tried converting regions from panda series directly into a data frame, but, unfortunately, I was not able to get the results I wanted as the data frame had all the regions in one column and another with a list with all the countries.

After trying to find a solution, I tried solving it another way. I thought of converting the regions into a dictionary first, having the regions being a key and the values being the list corresponding to the region. After doing so, I saved the dictionary called dictionaryRegions. Finally, I converted the dictionary into a data frame, I oriented the index as the columns and the regions as rows and then transposed it so that the region was the columns again.

Finally, I converted the data frame as a CSV file.