Valentina Bernal

March 13, 2022

BUAN 4210

Marvel Mart Project Documentation

Part I: Cleaning the Data

This section of the project was a bit of a challenge as learning how to clean data was still a topic I was just beginning to understand. To create the correct script to successfully clean the data so I could properly use, manipulate, and analyze it throughout the rest of the project I first began with using "shape" and "count" functions to get an initial overview of what data values were missing from each column. I also used the "dtypes" function to obtain the data type of each column to see if there were any erroneous types explicitly shown in that way. Once my clean copy of the CSV file was created, I used the FOR loops demonstrated in class to efficiently iterate through every value of each column and tested for any erroneous values. I used the logic behind the float conversion test, as some data types that appeared as objects and were meant to be strings, could have also been a float data type. Once I performed the loop on all four of the columns that I knew had to be tested for missing/erroneous data, I removed the rows by referencing the NULL data values and the 0.0 float data values and making sure that none of them were associated to the mmSalesClean dataframe copy. All of this clean data was then transferred to a completely new CSV file to use throughout the rest of the project.

Part 2: Exploratory Data Analysis with Reports & Visualizations

In this section of the project, I relied heavily on my labs and in class notes. It was also an area that gave me an opportunity to explore other sources online for functions that could help me produce the code with code functions I hadn't previously used in class. I was able to use pandas library to easily create visualizations that best fit the data that I was obtaining. For the first few questions I acquired the data I needed with a "value_counts" function for a grand majority of them, and then alternated between matplotlib and seaborn to display the data and exploring different ways to manipulate the graphical depictions. While most of these data rankings I found required me to append them onto a text file, I found it more challenging to choose the most effective way to transfer my data without having a redundant code block. I used various internet sources to research different forms of code that were able to append to the text files and I switched between using a loop and a function throughout the project when appending to the text files. For my calculations, I used the specific function that automatically performs the calculations on the columns referenced and after several tries was able to shorten the code to a shorter, more efficient block. To then put these calculations onto a visual, I researched thebest way to transfer data with one set of values and the other with strings as the keys and concluded that creating a dataframe for both (the grouped one too) was the best approach. Once these two visuals were completed, I added the calculations I initially found to a new text file and tried using an actual function to avoid repeating "writer" statements.

Part 3: Cross-Reference Statistics

This section was the most challenging section of the project for me, and I referred mostly to different online sources, so I dedicated most of my time to researching and trying to find ways to successfully answer what was being asked. After much research and video watching, I found the best format to obtain the answer to be using the suggestions provided in the question prompt and building from that with the research I found. I first created an empty dictionary to have lists that would hold all the unique countries belonging to each of the regions needed. I then associated those regions with the keys of the empty dictionary to the countries which would then be the values of the dictionary. Since I established a dataframe name for this particular problem at the beginning of the code block, I referred to it when calling to each region that is linked to the respective country column. The “unique” function is what allows the country column to call each distinct country so that there are no duplicates within the dataframe I was creating. I utilized a for loop with a concatenation to easily read each region list being printed first, and in the final code block I used “keys” function to iterate each line, using “concat” to join the region columns for best readability in the file, and finally transferring that data to the csv file I created.