Dalton Murray

Python for Data Analysis and Visualization

Professor Shams

February 19, 2024

**Assignment #2**

**Overall**

A screen shot of a computer program

Description automatically generated

A screen shot of a computer code

Description automatically generated

**Task 1 – Data loading and inspection**

Load the datasets sales\_q1.csv and sales\_q2.csv into separate DataFrames.

Inspect the first 5 rows of each DataFrame to understand the structure of the data.

Check for any missing values in both datasets.

**A computer screen shot of text

Description automatically generated**

In task one I prepare the program by importing sys, os, pandas, numpy, and pyplot as these will be the required libraries to use in this program. I then also prepare it with my basic template putting everything into a def main and calling and exiting the def main.

For the first part of the task I easily read the supplied .csv files and make them a dataframe and set them to salesQ1 and salesQ2.

For the second part I use the .head(5) function to list out the first five rows of each of the dataframes with a little formatting.

A screenshot of a computer screen

Description automatically generated

For the third part for each of the dataframes I check what is null in the dataframe and then I print out a summary of what is null for neatness of the print statement.

A screen shot of a computer

Description automatically generated

**Task 2 – Basic data manipulation**

Add a new column to each DataFrame, Total\_Sales, calculated as Price \* Quantity.

Create a Series from the Product\_ID column of the first DataFrame and explore it using methods like head(), tail(), and describe().

A screen shot of a computer code

Description automatically generated

For the first part of task 2 I add a new column for both of the dataframes called Total\_Sales I then set the column equal to the price multiplied by the quantity, after this for the second part I create a series for salesQ1 of the column Product\_ID. I then print out information of the series such as its head, tail, and the description.

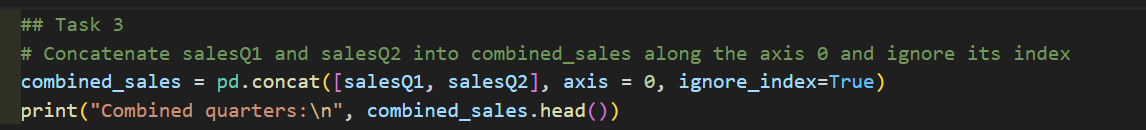
A screenshot of a computer

Description automatically generated

**Task 3 – Concatenation task**

Concatenate the two DataFrames along rows (axis=0) into a single DataFrame named combined\_sales.

Ensure that the indexes are reset appropriately in the combined DataFrame.



For the first part of task 3 I concatenate salesQ1 with salesQ2 along the 0 axis and I set it to ignore index to make sure that the indexes are getting set/reset properly. I then print out a head of this new concatenated list called combined\_sales.

A screenshot of a computer screen

Description automatically generated

**Task 4 – Advanced data manipulation**

Group the combined DataFrame by Product\_ID and calculate the total sales for each product across both quarters.

Sort the products based on the total sales in descending order.

A screen shot of a computer code

Description automatically generated

For the first part of task 4 I make a dataframe called productGroupBy which I set to the grouped by Product ID’s and I add a summary to each of the product id’s of the total sales column. I then sort this in descending order and print it out.

**A screenshot of a computer

Description automatically generated**

**\*\*Please note that this is where I get confused of what I should be and shouldn’t be setting to the overall combined\_list dataframe so if the task doesn’t explicitly say to set it to that then I don’t and treat it as if it were a branch of the dataframe rather than setting it to the dataframe. I asked about this in class and I believe this is what you said to do.**

**Task 5 – Data filtering**

Filter out the rows in the combined\_sales DataFrame where Quantity sold is greater than 50.

**A screen shot of a computer code

Description automatically generated**

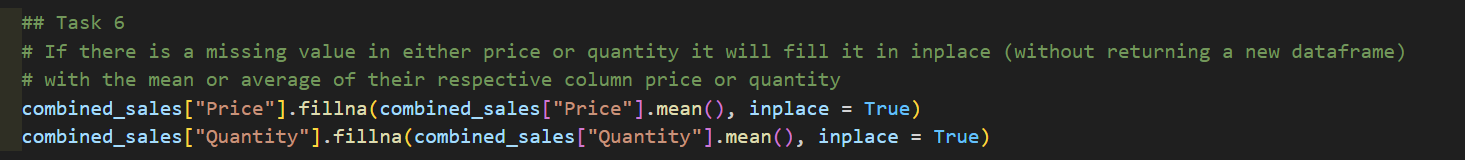
For task 5 I filter out ever row of combined sales where the quantity is greater than 50, note that this says “Quantity” < 50 which would mean less than 50, however, that is what it is keeping not getting rid of, by doing this it is getting rid of anything that is greater than 50.

A screenshot of a computer screen

Description automatically generated

**Task 6 – Handling missing data**

If there are any missing values in Price or Quantity, fill them with the average of their respective columns.



For this task, for each of the dataframes combined\_sales I fill any missing or NaN valued items with the mean of their respective values I then set this to do it inplace because I want to do it directly on the combined\_sales dataframe and not set it to something new.

**Task 7 – File operations**

Save the final combined\_sales DataFrame to a new CSV file, combined\_sales\_analysis.csv.

A black background with green and orange text

Description automatically generated

In this task I tell the dataframe to be saved as a csv with no index.

A screen shot of a computer

Description automatically generated

**Task 8 – Bonus visualize**

Visualize the sales data using any library of your choice (like Matplotlib or Seaborn). You could create a bar chart showing total sales for the top 5 products.

**A screen shot of a computer code

Description automatically generated**

For this task I did a very simple visualization of the example given. I took the sorted product’s top 5 rows and set it to a variable top5Products I then made a bar with its index and values as the x and y. I then put x and y labels and set a title and told it to show the bar chart.

**A screenshot of a computer

Description automatically generated**

**Code**

#####################

## Dalton Murray #

## 02/12/2024 #

## Assingment 2 #

#####################

import sys

import os

import pandas as pd # Import the pandas library and sets an alias to pa

import numpy as np

import matplotlib.pyplot as plt # Imports matplotlib pyplot library and sets an alias to plt

# Defines the \_\_main\_\_ function

def \_\_main\_\_():

## Task 1

# Sets the database location to the relative folder

database = "./"

# Sets the salesQ1 and salesQ2 variables to dataframes of the read csv's

salesQ1 = pd.read\_csv(os.path.join(database, "./sales\_q1-1.csv"))

salesQ2 = pd.read\_csv(os.path.join(database, "./sales\_q2-1.csv"))

# Prints the first 5 rows of the dataframes

print("Sales Q1:\n", salesQ1.head(5))

print("Sales Q2:\n", salesQ2.head(5))

# Checks for missing values in the dataframes by checking if a value is null

# Then summarizing/printing out the list of null values

print("Sales Q1 missing summary:\n", salesQ1.isnull().sum())

print("Sales Q2 missing summary:\n", salesQ2.isnull().sum())

## Task 2

# Adds a total sales column by multiplying the existing price by quantity for each dataframe

salesQ1["Total\_Sales"] = salesQ1["Price"] \* salesQ1["Quantity"]

salesQ2["Total\_Sales"] = salesQ2["Price"] \* salesQ2["Quantity"]

# Creates a series from the first data frame's product id column

salesQ1Series = salesQ1["Product\_ID"]

# Print out information about the series

print("Sales Q1 product ID head:\n",salesQ1Series.head())

print("Sales Q1 product ID tail:\n", salesQ1Series.tail())

print("Sales Q1 product ID describe:\n", salesQ1Series.describe())

## Task 3

# Concatenate salesQ1 and salesQ2 into combined\_sales along the axis 0 and ignore its index

combined\_sales = pd.concat([salesQ1, salesQ2], axis = 0, ignore\_index=True)

print("Combined quarters:\n", combined\_sales.head())

## Task 4

# Groups the Product\_ID's and then sorts it by the total sales in descending order

productGroupBy = combined\_sales.groupby("Product\_ID")["Total\_Sales"].sum()

productSort = productGroupBy.sort\_values(ascending = False)

print("Grouped & sorted products:\n", productSort)

## Task 5

# Filters out/removes the rows where quantity is greater than 50

quantityFilter = combined\_sales[combined\_sales["Quantity"] < 50]

print("Filtered quantity:\n", quantityFilter)

## Task 6

# If there is a missing value in either price or quantity it will fill it in inplace (without returning a new dataframe)

# with the mean or average of their respective column price or quantity

combined\_sales["Price"].fillna(combined\_sales["Price"].mean(), inplace = True)

combined\_sales["Quantity"].fillna(combined\_sales["Quantity"].mean(), inplace = True)

## Task 7

# Saves a csv file of the analysis performed

combined\_sales.to\_csv("combined\_sales\_analysis.csv", index = False)

## Task 8

top5Products = productSort.head(5)

plt.bar(top5Products.index, top5Products.values)

plt.xlabel("Product ID")

plt.ylabel("Total Sales in USD")

plt.title("Top 5 Products")

plt.show()

# Checks if the "\_\_name\_\_" variable equals "\_\_main\_\_"

if \_\_name\_\_ == "\_\_main\_\_":

sys.exit(\_\_main\_\_()) # Calls the "\_\_main\_\_" function and then after running exits smoothly

I have neither given nor received unauthorized aid in completing this work, nor have I presented someone else's work as my own.

*Dalton Murray*