**PRODUCT SALES ANALYSIS**

**Team memebers**

* **511921104008 - Arshvanth .V**
* **511921104040 - Musharaf Sharif .M**
* **511921104037 - Moha Rishi .S**
* **511921104017 - Dharneshwaran .H**

**Phase 3 submission document**

**Project Name : product sales analysis**

****

**What was the best month for sales? How much was earned that month?**

First of all, we need to see what kind of data we are trying to analyze. we can simply do this

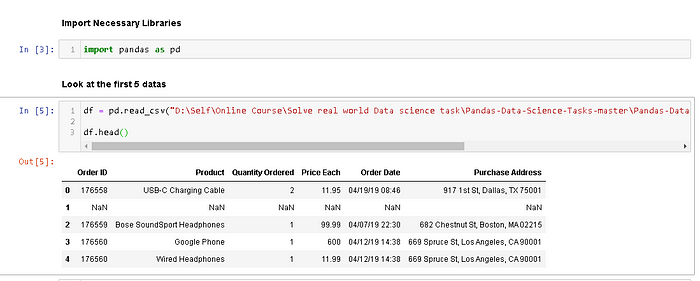


Figure 1. Code to show the top 5 data in Sales\_April\_2019.csv

We use panda to read the csv file and create a dataframe from it. The file’s directory can be put anywhere. I personally put it in D. You can copy the directory and paste it in the syntax. At Figure 1, we can see that we have 6 columns. Now the first task is to merge all 12 months worth of sales data (12 csv files) into a single csv file. To do that, we need to import new library called os.



Figure 2. Import new library os.

We need os library to read all csv files’ title and call it using *for loop*. As you can see from Figure 2, we successfully read all the csv files’ title and we’re ready to merge it. To do that, we can simply do

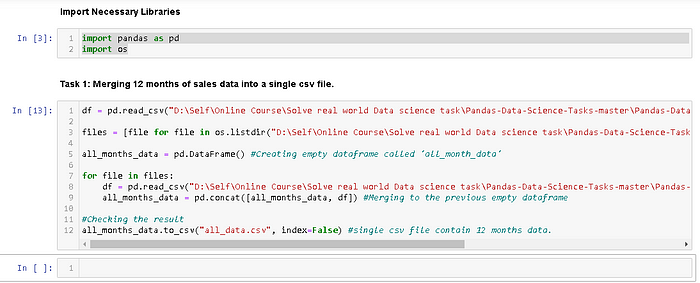


Figure 3. Creating a new file contains all 12 months data.

**— — — — — — — — DETAIL CODE — — — — — — — — — —**

import pandas as pd  
import os

df = pd.read\_csv(“D:\Self\Online Course\Solve real world Data science task\Pandas-Data-Science-Tasks-master\Pandas-Data-Science-Tasks-master\SalesAnalysis\Sales\_Data\Sales\_April\_2019.csv”)  
  
files = [file for file in os.listdir(“D:\Self\Online Course\Solve real world Data science task\Pandas-Data-Science-Tasks-master\Pandas-Data-Science-Tasks-master\SalesAnalysis\Sales\_Data”)]

all\_months\_data = pd.DataFrame() #Creating empty dataframe called ‘all\_month\_data’

for file in files:  
df = pd.read\_csv(“D:\Self\Online Course\Solve real world Data science task\Pandas-Data-Science-Tasks-master\Pandas-Data-Science-Tasks-master\SalesAnalysis\Sales\_Data/”+file)  
all\_months\_data = pd.concat([all\_months\_data, df]) #Merging to the previous empty dataframe

#Checking the result  
all\_months\_data.to\_csv(“all\_data.csv”, index=False) #single csv file contain 12 months data.

It will take a little longer time because of heavy computation. But once it’s done, you can open the same directory folder and check the folder called “Output”. You will see a new csv file contains all 12 months data.

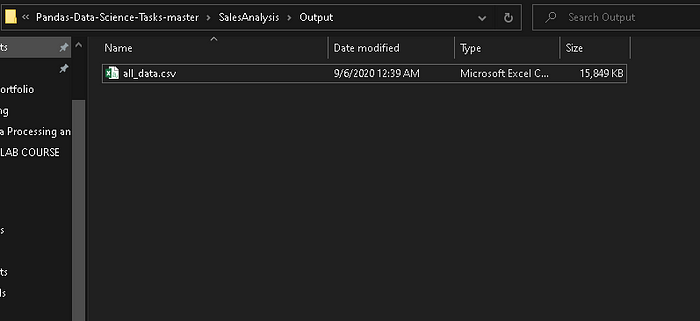


Figure 4. A new csv file contains all 12 months data.

After we create this new csv file, you can delete the previous code (if you want) and we will use this file to answer all of the problems.

Now we only use this code to read all of 12 months data.

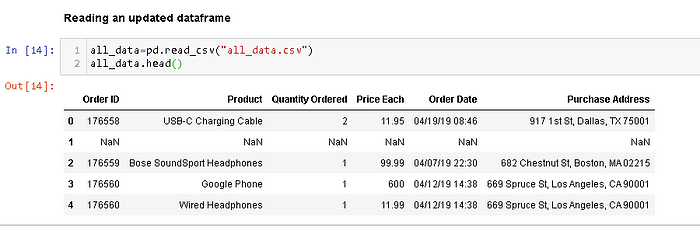


Figure 5. Reading an updated dataframe

Now we’re ready to answer the problem number 1. To remind you, the question is: **What was the best month for sales? How much was earned that month?**

To answer this problem, obviously we need an additional column called “Month”. If you look carefully at Figure 5, you will see the first 2 characters in “Order Date” values represent months. So the next task we will do is to add “Month” Column.



https://miro.medium.com/v2/resize:fit:700/1*winqW-qJ3a6IjFaC1ePnSw.png

Figure 6. Adding “Month” Column

Now, we get an issue here. There are NaN values in our data. You could spot on of NaN value at Figure 1 or Figure 5 in index 1. Now we need to clean up the data by dropping rows of NaN. Let’s spot more NaN value here. You don’t have to do this, I am just curious.

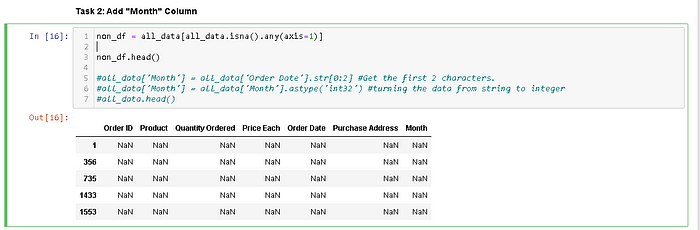
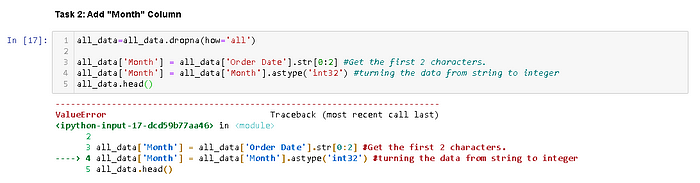


Figure 7. Spotting the NaN Values in our data.

We use *.isna().any(axis=1)* to spot rows containing the NaN values (axis = 0 to spot column containing NaN values). Now, we’re gonna remove it from our dataframe using *.dropna()* method.



https://miro.medium.com/v2/resize:fit:700/1*J8zN6VfV-_O0ImzOjwQP3w.png

Figure 8. Dropping the NaN values from our dataframe.

*.dropna()* method is successful, but we get a new issue here. There are values “Or” in our data. Let’s find it first.

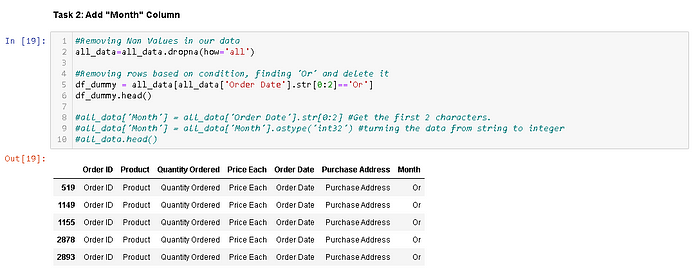


Figure 9. Finding “Or” values in our dataframe.

We can see clearly from Figure 9 that the issue is the rows contain the same words as the title rows. So clearly ‘Or’ is coming from ’Order Date’. We need to drop this “Or” rows just simply change the equal sign (“==”) to not equal sign (“!=”).

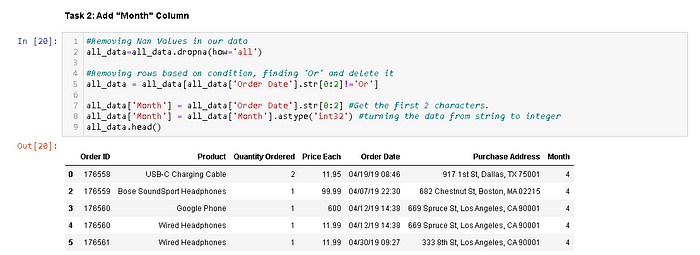


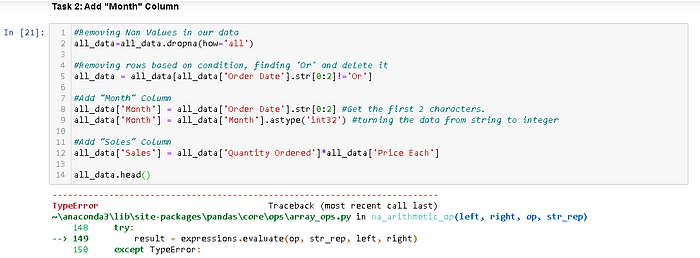
Figure 10. Dropping all unnecesary values in our dataframe.

We can see clearly from Figure 10 that we successfully created “Month” column and make its data type to integer.

Now, are we ready to answer the question? Not yet, we need obviously one more column called “Sales” Column.

How can we get that? We get “Sales” by multiplying “Quantity Ordered” and “Price Each” values.

Let’s create it.



https://miro.medium.com/v2/resize:fit:700/1*qvm7TJJJemAPuBlYweHg-A.png

Figure 11. Adding “Sales” Column in our dataframe.

Now we encounter a new issue. The values of the column “Quantity Ordered” and “Price Each” are strings.

So the next task is to convert these columns to the correct type (“Quantity Ordered” is integer and “Price Each” is float).

We’re gonna use *pd.to\_numeric()* method to convert them to numeric.

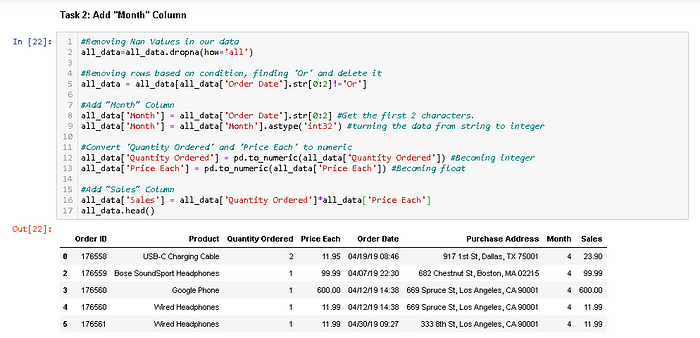


Figure 12. Converting “Quantity Ordered” and “Price Each” Columns to Numeric

Now the “Sales” Column is successfully created, we can answer the first question.

What was the best month for sales? How much was earned that month? We can easily answer it by using*groupby(‘Month’).sum()* method.

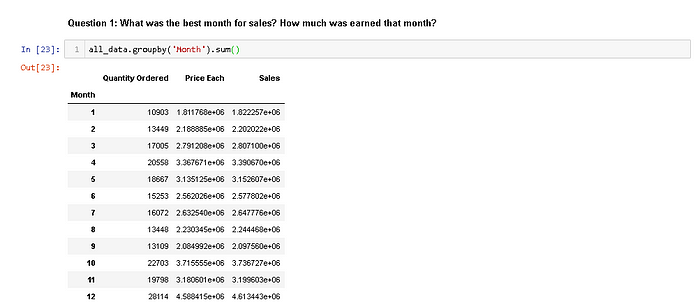


Figure 13. Grouping by month and summing the Sales.

Look carefully at Figure 13. We can clearly see that month 12 (December) is the highest sales in 2019 with approximately $4,810,000.

But we need to visualize it to make our bussiness partner easier to understand.

So we’re gonna import matplotlib and visualizing our results with bar chart.

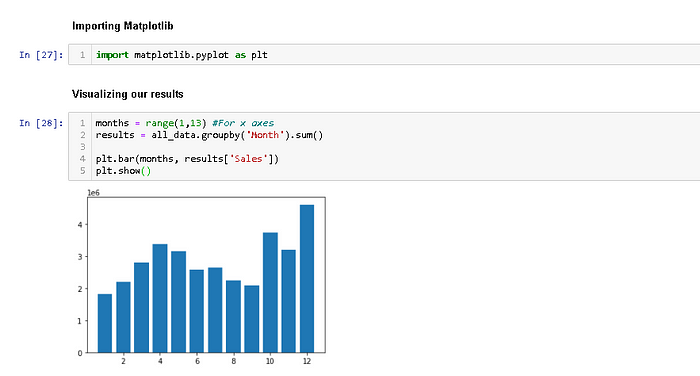


Figure 14. Visualizing our results using matplotlib library.

It’s good. But we need to make it looks neater. So we’re just gonna add a little code.

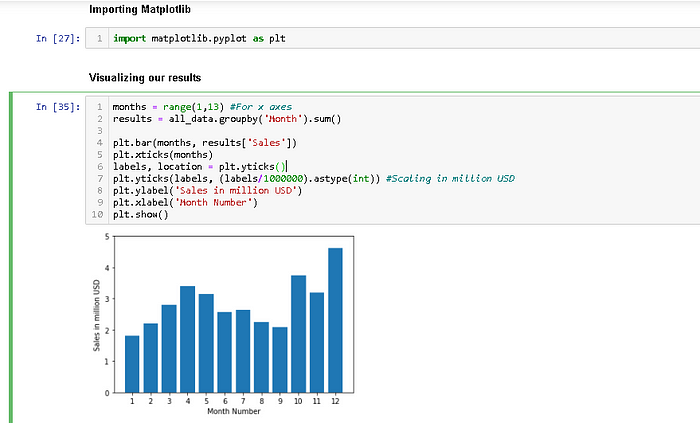


Figure 15. Improving the visualization.

Now, not only we can get the highest sales, but we can also get the lowest sales just by looking it for a few seconds. As a data scientist, we have to figure out why a certain month is better than others.

Maybe the company spend more money on April so the product sales are increasing. Maybe the best product sales are on December because it’s holiday and Christmas.

Those are just my hyphothesis, right now we don’t have enough data to prove that hypothesis.But we can take these as a consideration if you want to decide something that relates to product sales.

**Conclusion:**

We use panda to read the csv file and create a dataframe from it. The file’s directory can be put anywhere. In this phase we show that the how to create csv file and read the csv in panda , and to create the dataframe form the function. In which the files are directory ,such directory can put anywhere in the process for the function. using the panda to read the csv file’s and create a dataframe.