LONDON METROPOLITAN UNIVERSITY

CC5067NI

Smart Data Discovery

60% Individual Coursework

2023 Spring Semester

Student Name: BISHAL MAHAT CHHETRI

London Met ID: 21049501

College ID: **np01cp4a210155**

Assignment Submission Date: May 4, 2023

Assignment Submission Date: May 4, 2023

Word Count: 2057

I confirm that I understand my coursework needs to be submitted online via Google Classroom under the relevant module page before the deadline in order for my assignment to be accepted and marked. I am fully aware that late submissions will be treated as non-submission and a mark of zero will be awarded.

Table of Contents

List of Figures

List of Table

1 Data Understanding

Data understanding phase of CRISP-DM involves taking a closer look at the data available for mining. It is the process of comprehending the data that you have collected or acquired, in order to develop an understanding of its quality, structure, format, and content. This involves exploring the data in various ways, such as visualizing it, summarizing it, and identifying patterns or trends in it (Corporation, 2021).

Any data analysis or machine learning project must start with the process of understanding the data because it lays the groundwork for later steps like data cleaning, feature engineering, and model development. Making wise decisions and producing reliable results require a deep grasp of the data.

Following are the four tasks include in this phase.

- **Data collection:**

  All necessary information is gathered, and if necessary, it is loaded into the analysis tool.

- **Data Descriptions:**

  After analysis, the data's surface features, such as data type, record count, and so forth, are documented.

- **Data exploration:**

  Most deals study the data The relationships between the data can be discovered by querying, visualization, and data analysis.

- **Data Validation:**

  To determine if the data is pure or impure, it is examined. Any issues with the quality are noted (EMILY STEVENS, 2022).

1.2 Table of Description

S.NO	Column Name	Description	Data Type
1	Order ID	This Column holds unique Id of the order.	Float64
2	Product	This Column holds name of the product.	Object
3	Quantity Ordered	This Column holds the data of Quantity of the product Ordered.	Float64
4	Price Each	This Column holds the price of each product.	Float64
5	Order Date	This Column holds the data of date in which the product was ordered	Object
6	Purchased Address	This Column holds address of the customer who parched the products	Object

Table 1 Table of Description of DataFrame.

```
DataFrame.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 186850 entries, 0 to 11685
Data columns (total 6 columns):
 #   Column            Non-Null Count   Dtype
---  ------            --------------   -----
 0   Order ID          185950 non-null  float64
 1   Product           185950 non-null  object
 2   Quantity Ordered  185950 non-null  float64
 3   Price Each        185950 non-null  float64
 4   Order Date        185950 non-null  object
 5   Purchase Address  185950 non-null  object
dtypes: float64(3), object(3)
memory usage: 10.0+ MB
```

Figure 1:Figure of table Information.

2.Data Preparation

The process of comparing two or more data sets using visual tools, like graphs, is known as data presentation. You can depict how the information links to other data using a graph (team, 2022).

2.1 Write a python program to merge data from each month into one CSV and read in updated dataframe.

Firstly, we have to import all the necessary libraries.

```
import pandas as pd #Importing Pandas library
import os #Importing Pandas library
from matplotlib import pyplot as plt
```

Figure 2: Figure of importing the library.

Here, pandas are a popular data manipulation library, matplotlib pyplot is a library for creating visualizations in python and os the module in python standard library.

After importing libraries, we have to load file to merge the data.

```
path = './Sales_Analysis/CSV/'
for files in os.listdir(path):
    print(files)
DataFrame = pd.DataFrame()
for files in os.listdir(path):
    df = pd.read_csv(path+files)
    DataFrame = pd.concat([DataFrame,df])
```

```
Sales_April_2019.csv
Sales_August_2019.csv
Sales_December_2019.csv
Sales_February_2019.csv
Sales_January_2019.csv
Sales_July_2019.csv
Sales_June_2019.csv
Sales_March_2019.csv
Sales_May_2019.csv
Sales_November_2019.csv
Sales_October_2019.csv
Sales_September_2019.csv
```

Figure 3:Figure of Updating the dataframe

Here we give path to the csv file, the an empty data frame if created with in the loop “DataFarme” , the df variable read the csv file with help pf pd.read.csv(). Finally the dataframe is updated to the empty dataframe created.

2.2 Write a python program to remove the NaN missing values from updated dataframe.

Detecting the missing values from the updated dataframe.

```
## Removing the NaN Missing value From udated dataframe.
```

```
DataFrame = DataFrame.dropna()
```

```
DataFrame
```

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address
0	176558.0	USB-C Charging Cable	2.0	11.95	4/19/2019 8:46	917 1st St, Dallas, TX 75001
2	176559.0	Bose SoundSport Headphones	1.0	99.99	4/7/2019 22:30	682 Chestnut St, Boston, MA 02215
3	176560.0	Google Phone	1.0	600.00	4/12/2019 14:38	669 Spruce St, Los Angeles, CA 90001
4	176560.0	Wired Headphones	1.0	11.99	4/12/2019 14:38	669 Spruce St, Los Angeles, CA 90001
5	176561.0	Wired Headphones	1.0	11.99	4/30/2019 9:27	333 8th St, Los Angeles, CA 90001
...	...	...	...	...	...	...
11681	259353.0	AAA Batteries (4-pack)	3.0	2.99	9/17/2019 20:56	840 Highland St, Los Angeles, CA 90001
11682	259354.0	iPhone	1.0	700.00	9/1/2019 16:00	216 Dogwood St, San Francisco, CA 94016
11683	259355.0	iPhone	1.0	700.00	9/23/2019 7:39	220 12th St, San Francisco, CA 94016
11684	259356.0	34in Ultrawide Monitor	1.0	379.99	9/19/2019 17:30	511 Forest St, San Francisco, CA 94016
11685	259357.0	USB-C Charging Cable	1.0	11.95	9/30/2019 0:18	250 Meadow St, San Francisco, CA 94016

185950 rows × 6 columns

Figure 4: Figure of Detecting the missing values from the updated datafram.

Here, we can see the output where NaN missing data are removed.

2.3 Write a python program to convert Quantity Ordered and Price Each to numeric.

Converting Quantity Ordered and Price Each to numeric values.

```
## Write a program to convert Quantity Ordered and Price Each to numeric.
```

```
DataFrame['Quantity Ordered']=pd.to_numeric(DataFrame['Quantity Ordered'])
DataFrame['Price Each']=pd.to_numeric(DataFrame['Price Each'])
DataFrame
```

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address
0	176558.0	USB-C Charging Cable	2.0	11.95	4/19/2019 8:46	917 1st St, Dallas, TX 75001
2	176559.0	Bose SoundSport Headphones	1.0	99.99	4/7/2019 22:30	682 Chestnut St, Boston, MA 02215
3	176560.0	Google Phone	1.0	600.00	4/12/2019 14:38	669 Spruce St, Los Angeles, CA 90001
4	176560.0	Wired Headphones	1.0	11.99	4/12/2019 14:38	669 Spruce St, Los Angeles, CA 90001
5	176561.0	Wired Headphones	1.0	11.99	4/30/2019 9:27	333 8th St, Los Angeles, CA 90001
...	...	...	...	...	...	...
11681	259353.0	AAA Batteries (4-pack)	3.0	2.99	9/17/2019 20:56	840 Highland St, Los Angeles, CA 90001
11682	259354.0	iPhone	1.0	700.00	9/1/2019 16:00	216 Dogwood St, San Francisco, CA 94016
11683	259355.0	iPhone	1.0	700.00	9/23/2019 7:39	220 12th St, San Francisco, CA 94016
11684	259356.0	34in Ultrawide Monitor	1.0	379.99	9/19/2019 17:30	511 Forest St, San Francisco, CA 94016
11685	259357.0	USB-C Charging Cable	1.0	11.95	9/30/2019 0:18	250 Meadow St, San Francisco, CA 94016

185950 rows × 6 columns

Figure 5: Figure of Converting Quantity Ordered and Price Each to numeric values.

Here, we have used to_numeric method to convert Quantity Ordered and Price to numeric values which is one of the functions in Pandas which is used to convert argument to a numeric type.

2.4 Create a new column named Month from Ordered Date of updated dataframe and convert it to integer as data type

Creating a new column named Month then updating dataframe from ordered data in it.

```
## Create a new column named Month from Ordered Date of updated dataframe and convert it to integer as data type.
```

```
DataFrame['Month'] = DataFrame['Order Date'].str.split('/').str[0].astype(int)
DataFrame
```

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address	Month	City
0	176558.0	USB-C Charging Cable	2.0	11.95	4/19/2019 8:46	917 1st St, Dallas, TX 75001	4	Dallas
2	176559.0	Bose SoundSport Headphones	1.0	99.99	4/7/2019 22:30	682 Chestnut St, Boston, MA 02215	4	Boston
3	176560.0	Google Phone	1.0	600.00	4/12/2019 14:38	669 Spruce St, Los Angeles, CA 90001	4	Los Angeles
4	176560.0	Wired Headphones	1.0	11.99	4/12/2019 14:38	669 Spruce St, Los Angeles, CA 90001	4	Los Angeles
5	176561.0	Wired Headphones	1.0	11.99	4/30/2019 9:27	333 8th St, Los Angeles, CA 90001	4	Los Angeles
...	...	...	...	...	...	...	...	...
11681	259353.0	AAA Batteries (4-pack)	3.0	2.99	9/17/2019 20:56	840 Highland St, Los Angeles, CA 90001	9	Los Angeles
11682	259354.0	iPhone	1.0	700.00	9/1/2019 16:00	216 Dogwood St, San Francisco, CA 94016	9	San Francisco
11683	259355.0	iPhone	1.0	700.00	9/23/2019 7:39	220 12th St, San Francisco, CA 94016	9	San Francisco
11684	259356.0	34in Ultrawide Monitor	1.0	379.99	9/19/2019 17:30	511 Forest St, San Francisco, CA 94016	9	San Francisco
11685	259357.0	USB-C Charging Cable	1.0	11.95	9/30/2019 0:18	250 Meadow St, San Francisco, CA 94016	9	San Francisco

185950 rows × 8 columns

```
DataFrame['Month'].dtype
```

```
dtype('int32')
```

Figure 6: Figure of Creating a new column named Month then updating dataframe from ordered data.

Here, we've created a new column called "Month" for this. The following code uses str.split (‘ ‘) to separate the city from Oder Date data frame. The we use dataframe[].dtype to see the data type of the dataframe created.

2.5. Create a new column named City from Purchase Address based on the value in updated dataframe.

Creating a new column named City then updating dataframe from Purchase Address in it.

```
##Create a new column named City from Purchase Address based on the value in updated dataframe.
```

```
DataFrame['City'] = DataFrame['Purchase Address'].str.split(',').str[1]
DataFrame
```

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address	Month	City
0	176558.0	USB-C Charging Cable	2.0	11.95	4/19/2019 8:46	917 1st St, Dallas, TX 75001	4	Dallas
2	176559.0	Bose SoundSport Headphones	1.0	99.99	4/7/2019 22:30	682 Chestnut St, Boston, MA 02215	4	Boston
3	176560.0	Google Phone	1.0	600.00	4/12/2019 14:38	669 Spruce St, Los Angeles, CA 90001	4	Los Angeles
4	176560.0	Wired Headphones	1.0	11.99	4/12/2019 14:38	669 Spruce St, Los Angeles, CA 90001	4	Los Angeles
5	176561.0	Wired Headphones	1.0	11.99	4/30/2019 9:27	333 8th St, Los Angeles, CA 90001	4	Los Angeles
...	...	...	...	...	...	...	...	...
11681	259353.0	AAA Batteries (4-pack)	3.0	2.99	9/17/2019 20:56	840 Highland St, Los Angeles, CA 90001	9	Los Angeles
11682	259354.0	iPhone	1.0	700.00	9/1/2019 16:00	216 Dogwood St, San Francisco, CA 94016	9	San Francisco
11683	259355.0	iPhone	1.0	700.00	9/23/2019 7:39	220 12th St, San Francisco, CA 94016	9	San Francisco
11684	259356.0	34in Ultrawide Monitor	1.0	379.99	9/19/2019 17:30	511 Forest St, San Francisco, CA 94016	9	San Francisco
11685	259357.0	USB-C Charging Cable	1.0	11.95	9/30/2019 0:18	250 Meadow St, San Francisco, CA 94016	9	San Francisco

185950 rows × 8 columns

Figure 7: Figure of Creating a new column named City then updating dataframe from Purchase Address.

Here, we've created a new column called "City" for this. The following code uses str.split (‘ ‘) to separate the city from Purchase Address data frame.

3 Data Analysis

The process of cleansing, converting, and modelling data in order to find relevant information for business decision-making is known as data analysis. Extracting usable information from data and making decisions based on that analysis are the goals of data analysis (Johnson, 2023).

3.1 Write a Python program to show summary statistics of sum, mean, standard deviation, skewness, and kurtosis of any chosen variable.

```
##Python program to show summary statistics of sum, mean, standard deviation, skewness, and kurtosis of any chosen variable.

Quantity_sum =DataFrame['Quantity Ordered'].sum()##Using .sum() function
Quantity_mean =DataFrame['Quantity Ordered'].mean()##Using .mean() function
Quantity_std =DataFrame['Quantity Ordered'].std()##Using .std() function
Quantity_skew =DataFrame['Quantity Ordered'].skew()##Using .skew() function
Quantity_kurt =DataFrame['Quantity Ordered'].kurt()##Using .kurt() function

print("The summary statistics of sum, mean, standard deviation, skewness, and kurtosis of Quantity Ordered are:")
print('----------------------------------------------------------------------------------------------------------')
print('Sum:',Quantity_sum)
print("Mean:",Quantity_mean,)
print("Standard:",Quantity_std)
print("SKewness:",Quantity_skew)
print("Kurtosis",Quantity_kurt)
```

```
The summary statistics of sum, mean, standard deviation, skewness, and kurtosis of Quantity Ordered are:
----------------------------------------------------------------------------------------------------------
Sum: 209079.0
Mean: 1.1243828986286637
Standard: 0.44279262402849096
SKewness: 4.833164172577953
Kurtosis 31.82048892027536
```

Figure 8:Figure of show summary statistics of sum, mean, standard deviation, skewness, and kurtosis.

The code Quality_sum=DataFrame['Quantity Ordered'].sum() returns a summary statistics of the 'Quantity Ordered' column in the DataFrame filte, including count, mean, standard deviation, minimum, maximum, and quartile values.

Then the respected values are been displayed as below by using print method.

3.2 Write a Python program to calculate and show correlation of all variables.

A statistical indicator of the strength of a linear link between two variables is the correlation coefficient. Its values may be between -1 and 1. Values in one series rise as those in the other drop, and vice versa, according to a correlation coefficient of 1, which denotes a complete negative or inverse connection. A value of 1 indicates a direct and flawlessly positive link. No linear relationship exists when the correlation coefficient is 0.

Displaying the correlation of all variables (FERNANDO, 2021)

.

$$r = \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum (x_i - \bar{x})^2 (y_i - \bar{y})^2}}$$

Figure 9: Figure to calculate Personal correlation

```
##Write a Python program to calculate and show correlation of all variables.
```

```
# calculate correlation matrix
DataFrame.corr()##Using .corr() function
```

	Order ID	Quantity Ordered	Price Each	Month
Order ID	1.000000	0.000702	-0.002857	0.993063
Quantity Ordered	0.000702	1.000000	-0.148272	0.000791
Price Each	-0.002857	-0.148272	1.000000	-0.003375
Month	0.993063	0.000791	-0.003375	1.000000

Figure 10: Figure of Displaying the correlation of all variables.

Here, we utilize the corr() function to determine the correlation of the variable using the "Pearson" approach to determine the relationship between the columns in the Dataframe.When a variable's correlation with itself is determined to be 1, the correlation value is 1.

4 Data Exploration

Data exploration is the first stage of data analysis, used to examine and display data in order to gain first insights or to spot potential regions or patterns for further investigation. Users may better see the big picture and discover insights more quickly by using interactive dashboards and point-and-click data exploration (TIBC, 2023).

4.1 Which Month has the best sales? and how much was the earning in that month? Make a bar graph of sales as well.

```
best_sales = monthly.max()
best_sales
```

```
4613443.34
```

```
# sort the monthly sales by month in ascending order
monthly = monthly.sort_index()

# create a bar graph of the monthly sales

plt.bar(monthly.index, monthly.values, color='Red', alpha=0.8)

# add a title to the plot
plt.title("Total Sales by Month")

# show the plot
plt.show()
```

Figure 11:Figure of Finding the best sakes of the month.

Here, we utilized the group-by approach to determine how much money was sold in each month. The "groupby()" method is used in this code to group the DataFrame "DataFrame" by the values in the "Month" column, and the bar graph shows that the latest month had the largest sales in comparison to other months.

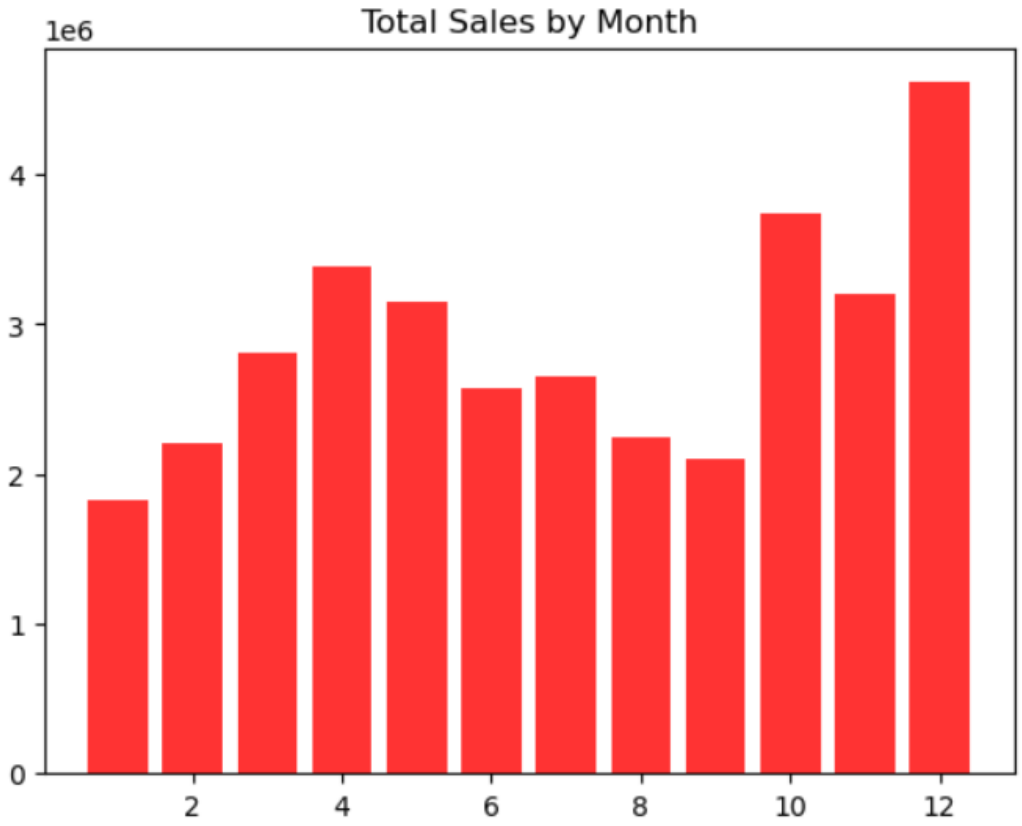

Figure 12: Figure the Bar char the shows the best sales of the month.

4.2 Which city has sold the highest product?

Displaying the product which was sold the most.

```
## Which city has sold the highest product?
```

```
city_sales = DataFrame.groupby('City')['Sales'].sum().sort_values(ascending=False)
max_sales_city = city_sales.idxmax()
max_sales_city
```

```
' San Francisco'
```

Figure 13: Figure of Highest product sold in the city.

We used groupby() to group ‘city’ and ‘Sales’ the adding it with the hep of sum() function

The we ued .idmax() to find the most sold month.

```
# Create a bar plot of the total sales by city
plt.bar(city_sales.index, city_sales.values, color='orange' ,alpha=0.8)
plt.title('Total Sales by City')
plt.xlabel('City')
plt.ylabel('Total Sales ($)')
plt.xticks(rotation=90)##
plt.show()
```

Figure 14:Figure of plotting total sales by the city in the bar graph.

Now we plot the total sale in the bar graph

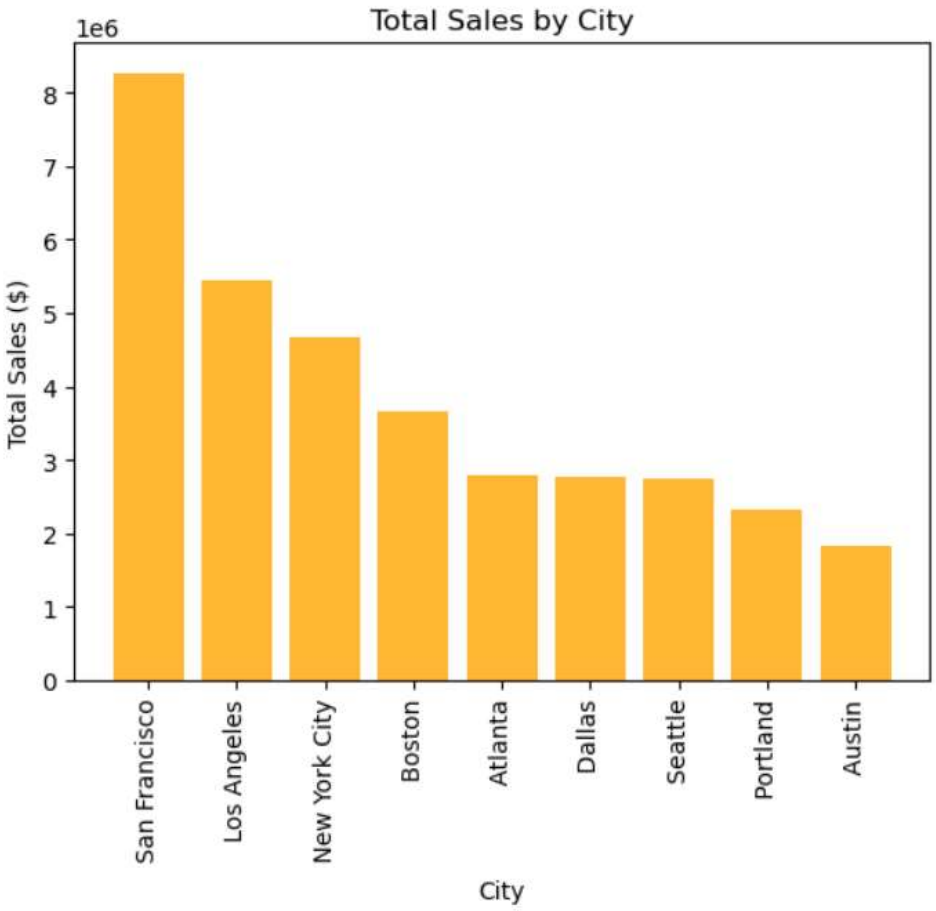

Figure 15 :Figure of Total sales of city.

4.3 Which product was sold the most in overall? Illustrate it through bar graph

Displaying the product which was sold the most.

```
## Which product was sold the most in overall?
```

```
product = DataFrame.groupby('Product')['Sales'].sum().sort_values(ascending=False)
most_sold_product = product.idxmax()
most_sold_product
```

```
'Macbook Pro Laptop'
```

Figure 16: Figure of most sold overall product.

We used groupby() to group 'Product' and 'Sales' the adding it with the hep of sum() function

The we ued .idmax() to find the most sold product.

```
##Illustrate it through bar graph.
```

```
plt.bar(product.index, product.values, color='red' ,alpha=0.8)
plt.title('Graph Of Total Product Sold')
plt.xlabel('Name of product')
plt.ylabel('Total Sales ($)')
plt.xticks(rotation=90)
plt.show()
```

Figure 17: Figure of plotting the data of most sold product in bar graph.

Now we plot the total sale in the bar graph.

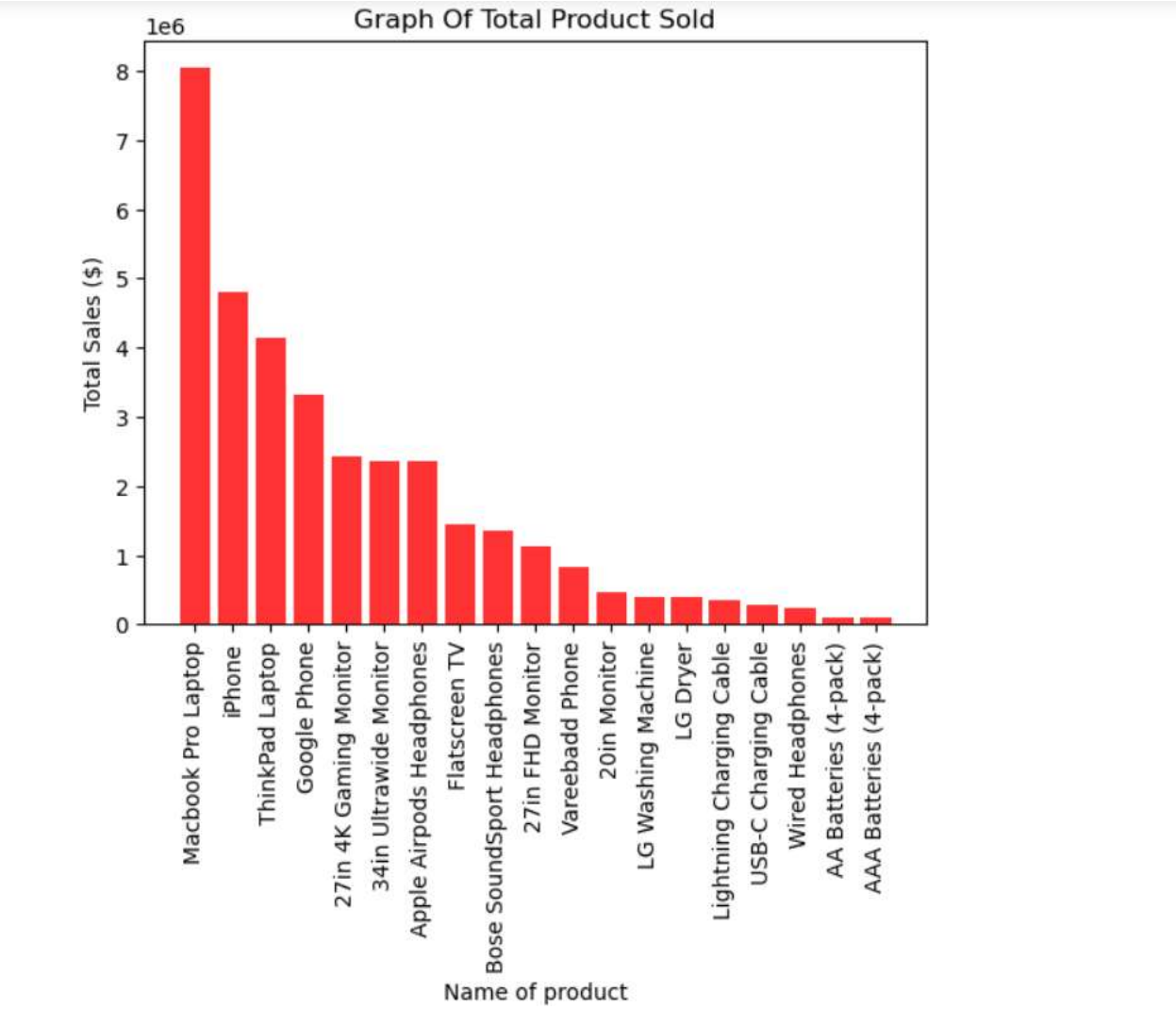

Figure 18: Figure of Bar graph of total product sold,

4.4 Write a Python program to show histogram plot of any chosen variables. Use proper labels in the graph.

Displaying the histogram plot of 'Price Each'

```
## Write a Python program to show histogram plot of any chosen variables. Use proper labels in the graph.
```

```
plt.hist(DataFrame['Price Each'], bins=27, color='orange' ,alpha=0.8)
plt.title('Histogram of Price Each')
plt.xlabel("price Each")
plt.ylabel('Count')
plt.show()
```

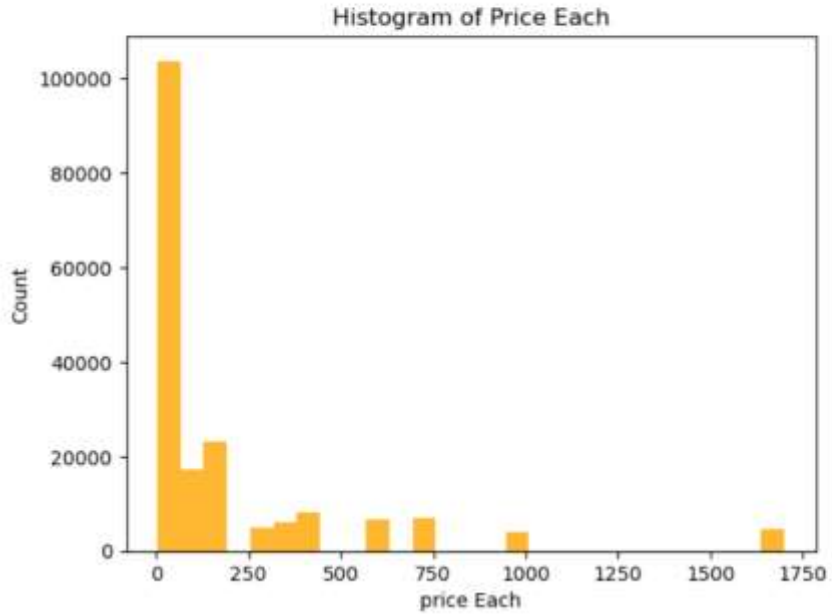

Figure 19: Figure of Histogram of the Price Each.

The above code makes a histogram of the "Price Each" column in the DataFrame using the Python Matplotlib package. The x-axis label, y-axis label, and title are set using the "xlabel()", "ylabel()", and "title()" functions, respectively, to construct the histogram using Matplotlib.

5 Conclusion

The coursework is designed to assess how well learners can apply their technical expertise and programming skills to evaluate actual data and prepare it for further study. The objective is to demonstrate your capacity for critical thought, problem-solving, and effective communication of your findings in an academic article. In order to demonstrate both analytical and problem-solving skills, this class required students to apply their programming knowledge and abilities to data analysis. This project will evaluate 2019 sales data for ABC Company using Python programming. Numerous searches on different websites, including Google, YouTube, and other social media platforms, were conducted to finish the coursework.

In this coursework we used Anaconda where we used jupyter Notebook, we learned different function, method, library of the python we even can to implement some of them like: .sum(), .max(), info(), dropna(), pd.to_numeric() etc.

In conclusion, I want to thank my respected instructors and mentors for always being there to guide us and help us work through any problems that came up during the homework. It was much simpler to understand when there was some direction. When I initially started programming, I acted like a novice, but regular study and effort have brought me a little bit closer to the level of a Python coder.

8 References

Corporation, I. (2021, JAn 01). *IBM*. Retrieved from IBM: https://www.ibm.com/docs/en/spss-modeler/saas?topic=guide-data-understanding

EMILY STEVENS. (2022, November 30). *careerFondary*. Retrieved from careerFondary: https://careerfoundry.com/en/blog/data-analytics/different-types-of-data-analysis/

FERNANDO, J. (2021, October 05). *Investopedia*. Retrieved from Investopedia: https://www.investopedia.com/terms/c/correlationcoefficient.asp

Johnson, D. (2023, march 25). *GURU99*. Retrieved from GURU99: https://www.guru99.com/what-is-data-analysis.html

team, I. e. (2022, October 1). *indeed* . Retrieved from indeed : https://in.indeed.com/career-advice/career-development/data-presentation

TIBC. (2023, May 4). *TIBC*. Retrieved from TIBC: https://www.tibco.com/reference-center/what-is-data-exploration