LONDON METROPOLITAN UNIVERSITY

islington college
(इस्लिङ्टन कलेज)

CC5067NI - Smart Data and Discovery

60 % Individual Coursework

2022-23 Spring

Student Name: Nahush MS Karki

London Met ID: 22015811

College ID: np01cp4s220117

Assignment Due Date: Thursday, May 4, 2023

Assignment Submission Date: Thursday, May 4, 2023

Word Count: 2321

I confirm that I understand my coursework needs to be submitted online via MySecondTeacher 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 marks of zero will be awarded.

Acknowledgment

I am grateful for all the support I have received from my friends and family. Every encouraging, productive help resulted in the completion of this overall project, and without it, this report would not have come to be. I would like to thank our professors who happily guided me through this report's phases. Online instructors were also a participant who helped me walk through many problems. With guidance from everyone involved, this project has been a complete whole.

Abstract

Business growth is obtained through the means of a growing client population. As the customers pour in, the sales increase. Recording these sales and interactions between the customer and the business is another beneficial step to obtaining profit and this is done through collecting, understanding, preparing, analyzing, and exploring the recorded data which is what's done throughoutut this project. A company called ABC has a set of data recorded to view and use. These data go through all the beneficial steps required to see any idea or prediction for growth. Using Python as a programming language and Jupyter as an environment to manipulate the language, the data will go through many phases and steps getting manipulated through the means of Python and Jupyter hoping to see the beneficial side of data discovery. The idea of statistics such as mean, skewness, standard deviation, etc, when used will extract valuable information that contributes to the idea of growth which is done in this Project.

Table of Contents

Table of Figure

Table of table

Introduction

This report is an assessment handed down by the '*Smart Data and Discovery*' module that deals with *analyzing*, *demonstrating*, and, *evaluating*, a set of data with the knowledge of programming. The data is collected from the *sale analysis* of **ABC company** in 2019. The goal of this project is to manipulate data from the sale analysis dataset using programming skills, specifically **Python**, which is an object-oriented, high-level programming language used mostly in applications that deal with *data science*, *software*, *web development*, *automation*, etc. All the Python codes are inserted and manipulated in a Platform called **Jupyter Notebook**, which is an interactive environment for data, notebooks, and codes. Since this project deals with a massive amount of data, jupyter is used and can manipulate those sales data.

Data Understanding

When dealing with data science, an individual must *understand* and *recognize* the data and the type of data being used. This step is an essential task of data understanding because it gives insight into the data that will be beneficial during data analysis.

Data can be collected from many sources and since ABC company is a **sales-related** company, the dataset this project manipulates is primarily focused on **sales-related data**. As mentioned before, the data collected is brought in from the sales of ABC company during the year 2019. These data include *Product*, *Quantity Ordered*, *Price Each*, *Order Date*, and *Purchase Address*. Additional data sets may be added such as *Quantity Price*, *Month*, and *City* to assist the data analysis process. The attributes of these data sets are explained in the table below.

(**str** = String values, **int** = integer values)

Dataset	Column Name	Description	Data Type
Product	*Product*	The name of the product that was bought.	str
Quantity	*Quantity Ordered*	The amount of product that was bought.	int
Price	*Price Each*	The price of each product.	int
Date	*Order Date*	The date of purchase.	DateTime
Address	*Purchase Address*	The address of the individual that made a successful transaction.	str
Total Quantity Price	*Quantity Price*	The total amount paid for n amount of product.	int
Month	*Month*	The month in which the Product was bought.	int
City	*City*	The name of the city where the product was bought.	str

Table 1. Data Description

Data Preparation

After understanding the type of data that's being dealt with, the data must be organized in a matter to be manipulated and stored without issues. This process of organizing is called **Data preparation.** For these raw data to be analyzed and processed, they must be collected, cleaned, and structured appropriately for the programming language to use its algorithms, explore, and visualize the dataset. The organization is done in *Jupyter Notebook*. Following the guidelines, some questions need to be answered in this process,

1. *Write a Python program to merge data from each month into one CSV and read in an updated data frame.*

ANS: First and foremost, the data that is in CSV format should be pulled and displayed in a Table. In the code below, the data is pulled and inserted into a data frame and sorted based on order date. Order date must have Date Time as its data type.

```
In [1]: import pandas as p
        import numpy as n
        import glob
        import matplotlib.pyplot as plt


        # Get data file
        path = r'files/*.csv'
        filenames = glob.glob(path)

        dfs = []
        for filename in filenames:
            dfs.append(p.read_csv(filename))

        # Concatenate all data into one DataFrame
        data = p.concat(dfs, ignore_index=True)




        # Change the data type of Order date to Datetime
        data['Order Date'] = p.to_datetime(data['Order Date'])

        # Short Data based on Order Date
        data.sort_values(by='Order Date', inplace = True)
```

Figure 1: Merge data

Result: Displays the data from CSV in a data frame.

	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
1	NaN	NaN	NaN	NaN	NaN	NaN
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
...	...	...	...	...	...	...
186845	259353.0	AAA Batteries (4-pack)	3.0	2.99	9/17/2019 20:56	840 Highland St, Los Angeles, CA 90001
186846	259354.0	iPhone	1.0	700.00	9/1/2019 16:00	216 Dogwood St, San Francisco, CA 94016
186847	259355.0	iPhone	1.0	700.00	9/23/2019 7:39	220 12th St, San Francisco, CA 94016
186848	259356.0	34in Ultrawide Monitor	1.0	379.99	9/19/2019 17:30	511 Forest St, San Francisco, CA 94016
186849	259357.0	USB-C Charging Cable	1.0	11.95	9/30/2019 0:18	250 Meadow St, San Francisco, CA 94016

186850 rows × 6 columns

Figure 2: 1st Result

2. *Write a Python program to remove the NaN missing values from the updated data frame.*

ANS:

```
#remove NaN Missing Value
data.dropna(inplace=True)
```

Figure 3: dropna

Using dataframe.**dropna()** with parameters, missing values or NaN values may be removed if needed. **'inplace'** decides to weather manipulate or create a new data frame.

Result: Displays the data frame with **no** *NaN* value

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address
73891	147268.0	Wired Headphones	1.0	11.99	2019-01-01 03:07:00	9 Lake St, New York City, NY 10001
74701	148041.0	USB-C Charging Cable	1.0	11.95	2019-01-01 03:40:00	760 Church St, San Francisco, CA 94016
76054	149343.0	Apple Airpods Headphones	1.0	150.00	2019-01-01 04:56:00	735 5th St, New York City, NY 10001
76708	149964.0	AAA Batteries (4-pack)	1.0	2.99	2019-01-01 05:53:00	75 Jackson St, Dallas, TX 75001
76061	149350.0	USB-C Charging Cable	2.0	11.95	2019-01-01 06:03:00	943 2nd St, Atlanta, GA 30301
...	...	...	...	...	...	...
39308	304165.0	AAA Batteries (4-pack)	1.0	2.99	2020-01-01 04:13:00	825 Adams St, Portland, OR 97035
34027	299125.0	USB-C Charging Cable	1.0	11.95	2020-01-01 04:21:00	754 Hickory St, New York City, NY 10001
41061	305840.0	Bose SoundSport Headphones	1.0	99.99	2020-01-01 04:54:00	784 River St, San Francisco, CA 94016
35497	300519.0	Bose SoundSport Headphones	1.0	99.99	2020-01-01 05:13:00	657 Spruce St, New York City, NY 10001
35498	300519.0	Lightning Charging Cable	1.0	14.95	2020-01-01 05:13:00	657 Spruce St, New York City, NY 10001

185950 rows × 6 columns

Figure 4: 2nd Result

3. *Write a Python program to convert Quantity Ordered and Price Each to numeric.*

ANS:

```
p.to_numeric(data['Quantity Ordered'], downcast='integer')

p.to_numeric(data['Price Each'], downcast='float')
```

Figure 5: to_numeric

Downcast may be ‘integer’, ‘signed’, ‘unsigned’, or ‘float’.

Result: Displays the whole data frame with int as a data type for the column Quantity Ordered and Price Each.

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address
73891	147268.0	Wired Headphones	1.0	11.99	2019-01-01 03:07:00	9 Lake St, New York City, NY 10001
74701	148041.0	USB-C Charging Cable	1.0	11.95	2019-01-01 03:40:00	760 Church St, San Francisco, CA 94016
76054	149343.0	Apple Airpods Headphones	1.0	150.00	2019-01-01 04:56:00	735 5th St, New York City, NY 10001
76708	149964.0	AAA Batteries (4-pack)	1.0	2.99	2019-01-01 05:53:00	75 Jackson St, Dallas, TX 75001
76061	149350.0	USB-C Charging Cable	2.0	11.95	2019-01-01 06:03:00	943 2nd St, Atlanta, GA 30301
...	...	...	...	...	...	...
39308	304165.0	AAA Batteries (4-pack)	1.0	2.99	2020-01-01 04:13:00	825 Adams St, Portland, OR 97035
34027	299125.0	USB-C Charging Cable	1.0	11.95	2020-01-01 04:21:00	754 Hickory St, New York City, NY 10001
41061	305840.0	Bose SoundSport Headphones	1.0	99.99	2020-01-01 04:54:00	784 River St, San Francisco, CA 94016
35497	300519.0	Bose SoundSport Headphones	1.0	99.99	2020-01-01 05:13:00	657 Spruce St, New York City, NY 10001
35498	300519.0	Lightning Charging Cable	1.0	14.95	2020-01-01 05:13:00	657 Spruce St, New York City, NY 10001

185950 rows × 6 columns

Figure 6: 3rd result

4. *Create a new column named Month from Ordered Date of the updated data frame and convert it to integer as data type.*

ANS:

```
data['Month'] = data['Order Date'].dt.strftime('%m')

data['Month'] = data['Month'].astype(int)
```

Figure 7: Month

strftime() with parameters is used to pull the month from a date.

astype() with an int parameter is used to change the data type of the month column.

Result: Displays the data frame with a new column, Month.

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address	Month
73891	147268.0	Wired Headphones	1.0	11.99	2019-01-01 03:07:00	9 Lake St, New York City, NY 10001	1
74701	148041.0	USB-C Charging Cable	1.0	11.95	2019-01-01 03:40:00	760 Church St, San Francisco, CA 94016	1
76054	149343.0	Apple Airpods Headphones	1.0	150.00	2019-01-01 04:56:00	735 5th St, New York City, NY 10001	1
76708	149964.0	AAA Batteries (4-pack)	1.0	2.99	2019-01-01 05:53:00	75 Jackson St, Dallas, TX 75001	1
76061	149350.0	USB-C Charging Cable	2.0	11.95	2019-01-01 06:03:00	943 2nd St, Atlanta, GA 30301	1
...	...	...	...	...	...	...	...
39308	304165.0	AAA Batteries (4-pack)	1.0	2.99	2020-01-01 04:13:00	825 Adams St, Portland, OR 97035	1
34027	299125.0	USB-C Charging Cable	1.0	11.95	2020-01-01 04:21:00	754 Hickory St, New York City, NY 10001	1
41061	305840.0	Bose SoundSport Headphones	1.0	99.99	2020-01-01 04:54:00	784 River St, San Francisco, CA 94016	1
35497	300519.0	Bose SoundSport Headphones	1.0	99.99	2020-01-01 05:13:00	657 Spruce St, New York City, NY 10001	1
35498	300519.0	Lightning Charging Cable	1.0	14.95	2020-01-01 05:13:00	657 Spruce St, New York City, NY 10001	1

185950 rows × 7 columns

Figure 8: 4th result

5. *Create a new column named City from Purchase Address based on the value in the updated data frame.*

ANS:

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

Figure 9: City

The figure above shows how the Purchase address has been split using **str.split()** to return the first string set, i.e., city.

Result: Displays the city where the item was purchased.

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address	Month	City
73891	147268.0	Wired Headphones	1.0	11.99	2019-01-01 03:07:00	9 Lake St, New York City, NY 10001	1	New York City
74701	148041.0	USB-C Charging Cable	1.0	11.95	2019-01-01 03:40:00	760 Church St, San Francisco, CA 94016	1	San Francisco
76054	149343.0	Apple Airpods Headphones	1.0	150.00	2019-01-01 04:56:00	735 5th St, New York City, NY 10001	1	New York City
76708	149964.0	AAA Batteries (4-pack)	1.0	2.99	2019-01-01 05:53:00	75 Jackson St, Dallas, TX 75001	1	Dallas
76061	149350.0	USB-C Charging Cable	2.0	11.95	2019-01-01 06:03:00	943 2nd St, Atlanta, GA 30301	1	Atlanta
...	...	...	...	...	...	...	...	...
39308	304165.0	AAA Batteries (4-pack)	1.0	2.99	2020-01-01 04:13:00	825 Adams St, Portland, OR 97035	1	Portland
34027	299125.0	USB-C Charging Cable	1.0	11.95	2020-01-01 04:21:00	754 Hickory St, New York City, NY 10001	1	New York City
41061	305840.0	Bose SoundSport Headphones	1.0	99.99	2020-01-01 04:54:00	784 River St, San Francisco, CA 94016	1	San Francisco
35497	300519.0	Bose SoundSport Headphones	1.0	99.99	2020-01-01 05:13:00	657 Spruce St, New York City, NY 10001	1	New York City
35498	300519.0	Lightning Charging Cable	1.0	14.95	2020-01-01 05:13:00	657 Spruce St, New York City, NY 10001	1	New York City

185950 rows × 8 columns

Figure 10: 5th Result

At this point, the data has been collected and organized. Now the process jumps into the next phase.

Data Analysis

The process of *cleaning*, *transforming*, and *modeling* data to extract useful information for business decision-making is called **Data Analysis**. This phase discovers any important statistics of data. The past collection of data once analyzed will be used to predict future outcomes. Python is a very well-known programming language and also a data analysis tool. To forward the analyzing process, the following questions must be answered.

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

Sum: The **quantity ordered** has been summed and the told amount of all products bought so far is **209079**.

```
In [4]: sum1 = data.loc[:,'Quantity Ordered'].sum()

        sum1

Out[4]: 209079
```

Figure 11: SUM

Mean: Based on the average value of data from the **quantity ordered**, when people buy products, they usually only buy 1 of any product. The mean of the quantity ordered is *1.1244.*

```
In [6]:
        mean = data.loc[:,'Quantity Ordered'].mean()

        mean

Out[6]: 1.1243828986286637
```

Figure 12: MEAN

Standard Deviation: This is the value that tells how, in relation to mean, scattered the data is. Taking the standard deviation of the **Quantity Ordered,** we get.

```
In [2]: Standard_Deviation = data.loc[:,'Quantity Ordered'].std()

        Standard_Deviation

        # Low standard deviation means data are clustered around the mean.

        # High standard deviation indicates data are more spread out.


        # Here the average people only buys one or more Product of similar item since the value of standard deviation is low.

Out[2]: 0.44279262402849046
```

Figure 13: Standard Deviation

Skewness: This is the concept that measures the asymmetry of distribution in a data set. If the value is 0 then the data set is symmetrical. Taking the skewness of **Month**, we get. (Value of Range is between +1 to -1)

```
In [10]: skew = data.loc[:,'Month'].skew(numeric_only=False)

         skew


         #Here the number is negative so data peak right and the tail points left

         #the mean of negatively skewed data will be less than the median

         # Average purchase happens before june

Out[10]: -0.08858776558911187
```

Figure 14: Skewness

Kurtosis: This measure describes the characteristics of the dataset. When plotted takes the shape of an upside-down hill. When the data is furthest from the mean, the graph forms tails on each side. The value of kurtosis indicated the amount of data that is in each tail. Taking the kurtosis of **Month**, we get. (Value of kurtosis may go from 1 to infinity but the standard value is 2, and vice versa with a negative number.)

```
In [4]: kurtosis = data.loc[:,'Month'].kurtosis()

        kurtosis

        # The higher the kurtosis the more amount of data there is

        # The tails of kurtosis will be short since the number is in negative

Out[4]: -1.293554842668311
```

Figure 15: Kurtosis

2. *Write a Python program to calculate and show the correlation of all variables.*

Correlation is when two variables form any kind of relationship with each other. The variables might be linearly related and might fluctuate in relation to each other but no matter the outcome, the relationship stays. The function to show the correlation is, dataframe.**corr()** as shown below.

```
In [7]: data.corr()


        # A correlation of -1.0 indicates a perfect negative correlation,
        # A correlation of 1.0 indicates a perfect positive correlation.
        # A value of zero indicates that there is no relationship

        # when two variables move in the same direction, the correlation coefficient is positive.
        # when two variables move in opposite directions, the correlation coefficient is negative.
```

Out[7]:

	Order ID	Quantity Ordered	Price Each	Month
Order ID	1.000000	0.000702	-0.002861	0.993063
Quantity Ordered	0.000702	1.000000	-0.148335	0.000791
Price Each	-0.002861	-0.148335	1.000000	-0.003379
Month	0.993063	0.000791	-0.003379	1.000000

Figure 16: Correlation

```
In [9]: sns.heatmap(data.corr());
```

Figure 17: HeatMap

We can see in the figures above that variables with less to no relationship have a darker color in the heat map while those with a relationship have a lighter color. The variable with the most correlation is the **Price Each** and the **Quantity Ordered**.

Data Exploration

This process is like analyzing data with the help of visualizing sets of data to reveal important insights or local patterns. Plotting sets of data is an important task during this step to visualize data and uncover samples of designs that may be useful. The most used attributes during this process are **size**, **quantity**, and **accuracy.** To further the process, questions must be answered.

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

In [10]:
```
per_month = data.groupby('Month').sum()

per_month
# month 12 has the highest sales since quantity price, i.e., total price of ordered items, is 3719295. Highest.
```

Out[10]:

Month	Order ID	Quantity Ordered	Price Each
1	1.421631e+09	10903	1804577
2	1.871053e+09	13449	2179934
3	2.564811e+09	17005	2779903
4	3.387347e+09	20558	3354065
5	3.345872e+09	18667	3122775
6	2.932976e+09	15253	2551984
7	3.284140e+09	16072	2621867
8	2.899374e+09	13448	2221451
9	2.948727e+09	13109	2076254
10	5.457110e+09	22703	3700299
11	5.047203e+09	19798	3167467
12	7.685905e+09	28114	4569702

Figure 18. Monthly Sales

The figure above groups the data frame by months while taking the sum of all other columns based on the month. The results as shown tells *December* has the best sales since the months produced $ **4569702**. Representing this on a graph, we get.

```
In [14]: months = range(1,13)

         plt.bar(months, per_month['Quantity Price'])
         plt.xlabel('Months')
         plt.ylabel('Sales in $Million')
         plt.show()
```

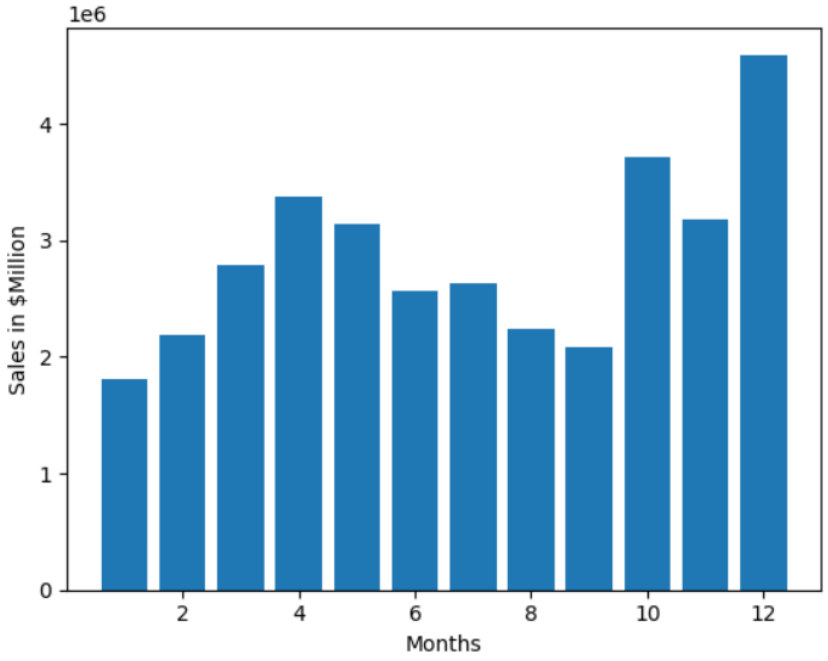

Figure 19: Month with Best Sales

2. *Which city has sold the highest product?*

In [15]:
```
per_city = data.groupby('City').sum()

per_city

#San Francisco has the highest with 8223669 sales
# 50239 were ordered
```

Out[15]:

City	Order ID	Quantity Ordered	Price Each	Month	Quantity Price
Atlanta	3.423838e+09	16602	2768857	104794	2782841
Austin	2.280982e+09	11153	1802516	69829	1811054
Boston	4.598265e+09	22528	3622510	141112	3644327
Dallas	3.415644e+09	16730	2741555	104620	2755117
Los Angeles	6.811085e+09	33289	5399261	208325	5426973
New York City	5.736334e+09	27932	4616764	175741	4642872
Portland	2.868861e+09	14053	2298450	87765	2309717
San Francisco	1.030444e+10	50239	8178055	315520	8223669
Seattle	3.406694e+09	16553	2722310	104941	2735070

Figure 20: City Sales

Grouping the data frame by city will generate a table shown above. Based on the table *San Francisco* has made the most sales with $ **8223669** worth of products sold.

3. *Which product was sold the most overall? Illustrate it through a bar graph.*

```
In [30]: per_product = data.groupby('Product')

         per_product.sum()

         #AAA Batteries (4-pack) has been sold mostly
```

Out[30]:

Product	Order ID	Quantity Ordered	Price Each	Month	Quantity Price
20in Monitor	9.508897e+08	4129	447009	29336	450061
27in 4K Gaming Monitor	1.442589e+09	6244	2423470	44440	2428916
27in FHD Monitor	1.724224e+09	7550	1118543	52558	1124950
34in Ultrawide Monitor	1.418986e+09	6199	2342599	43304	2349421
AA Batteries (4-pack)	4.744174e+09	27635	61731	145558	82905
AAA Batteries (4-pack)	4.764959e+09	31017	41282	146370	62034
Apple Airpods Headphones	3.579120e+09	15661	2332350	109477	2349150
Bose SoundSport Headphones	3.071496e+09	13457	1319175	94113	1332243
Flatscreen TV	1.110943e+09	4819	1440000	34224	1445700
Google Phone	1.262237e+09	5532	3315000	38305	3319200
LG Dryer	1.465563e+08	646	387600	4383	387600
LG Washing Machine	1.507187e+08	666	399600	4523	399600
Lightning Charging Cable	4.994091e+09	23217	303212	153092	325038
Macbook Pro Laptop	1.091958e+09	4728	8030800	33548	8037600
ThinkPad Laptop	9.487932e+08	4130	4123872	28950	4125870
USB-C Charging Cable	5.049538e+09	23975	240933	154819	263725
Vareebadd Phone	4.725325e+08	2068	826000	14309	827200
Wired Headphones	4.350952e+09	20557	207702	133397	226127
iPhone	1.571390e+09	6849	4789400	47941	4794300

Figure 21: Most Sold Product

Grouping the data frame based on products and summing all the outcomes generated a table shown above. According to the table, *AAA Batteries* were sold the most overall. Presenting this in a graph, we get,

```
quantity_ordered = per_product.sum()['Quantity Ordered']
quantity_ordered

product = [product for product, df in per_product]

plt.bar(product,quantity_ordered)
plt.xticks(product,rotation = 'vertical')
plt.xlabel('Product')
plt.ylabel('Quantity Ordered')
plt.show()
```

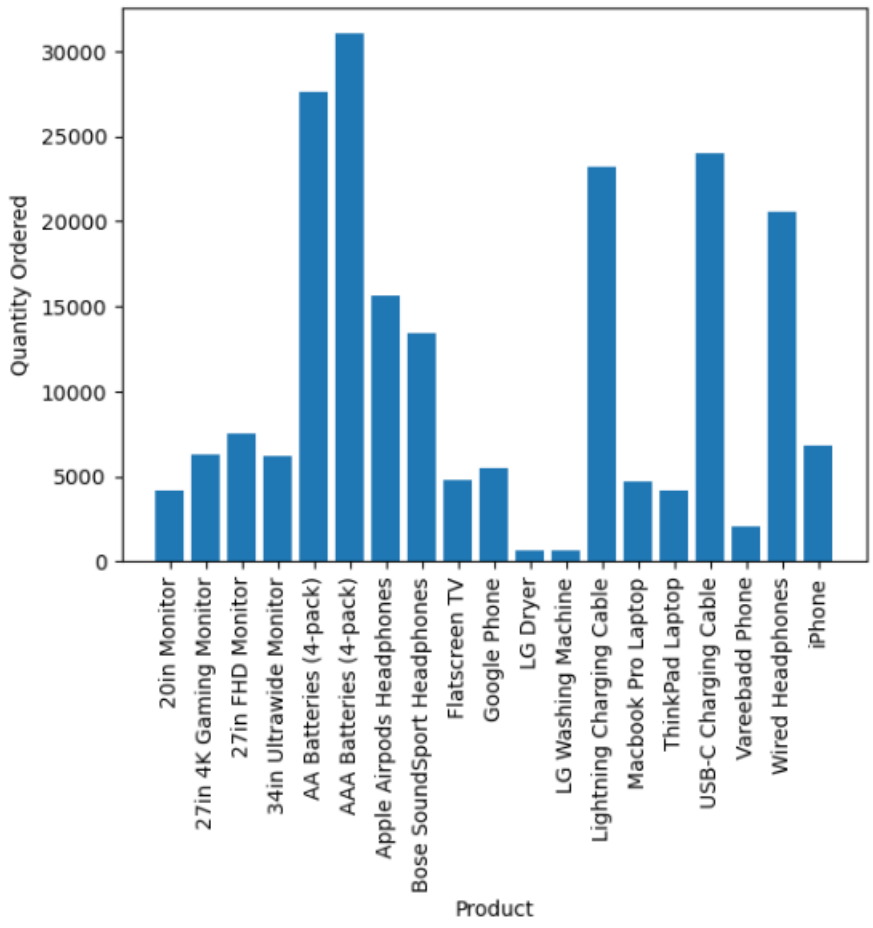

Figure 22: Most sold product Bar Graph

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

The distribution of values is shown in a diagram called a **histogram**. Many data points are taken and grouped in *logical ranges* or *bins*. Y-axis represents the count of occurrences in the data for each column while the x-axis can be used to visualize patterns of data distributions. The diagram below is a histogram of products and based on it the bars with the most length can be said is sold more than others.

```
In [60]: plt.hist(product, bins=15)
         plt.xticks(product,rotation = 'vertical')
         plt.show()

         #Frequently occurqance of product shown in diagram (Histogram)
```

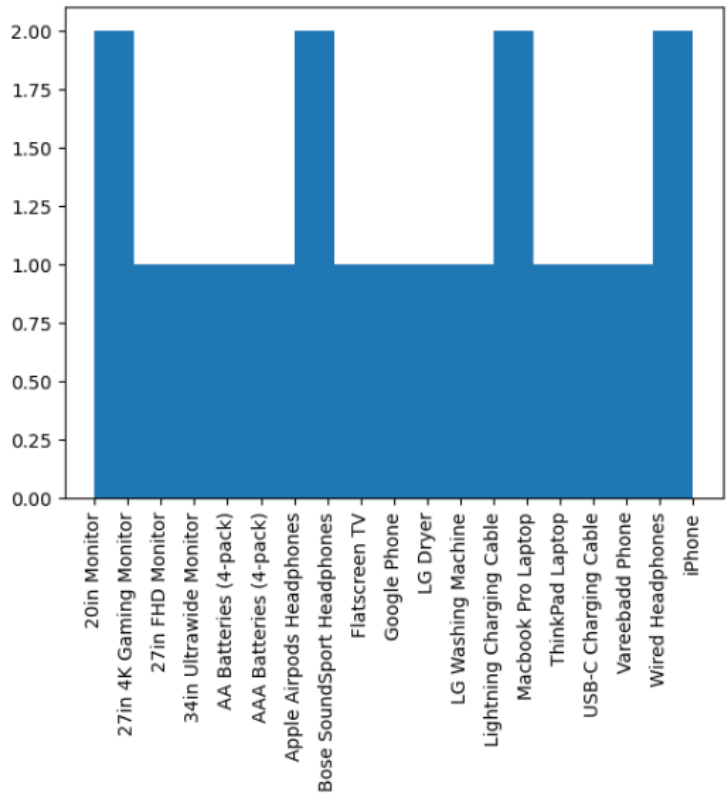

Figure 23: Histogram

Conclusion

This research aims to educate students on the topic of data discovery and manipulation. It had many steps that required extracting, manipulating, and visualizing data to detect a pattern that would be beneficial. The ABC Company had recorded its sales data set which was extracted and manipulated within this research project. Using Python, jupyter Notebook, was a great learning experience that was used to understand a bunch of data manipulation and analysis techniques to present the modified data in a desired form. Understanding the data was a time taking process, the amount of data that was involved was in a huge quantity, and bringing all those data into an editable language did take some time but was possible. Business and statistics are two main sectors that heavily rely on data discovery and manipulation allowing users with knowledge of data analysis to participate and be a part of. Understanding the concepts of business and statistics and relating them to the project was a difficult task but after some research and help from the professor, it was handled and understood. A hint of business and statistics concepts was explored, enough to handle a few basic evaluations such as mean, sum, skewness, etc. Visualizing the data and the relationship among them helped create an understanding of the data that were in correlation to each other. The aim was successfully reached since all the concepts were grabbed and understood after hard work and extensive research.

Reference

Accessed 3 May 2023. “What Is Python Used for? A Beginner’s Guide.” *Coursera*, www.coursera.org/articles/what-is-python-used-for-a-beginners-guide-to-using-python.

https://www.coursera.org/articles/what-is-python-used-for-a-beginners-guide-to-using-python

“What Is ...” *Amazon*, 1978, aws.amazon.com/what-is/data-preparation/#:~:text=Data%20preparation%20is%20the%20process,exploring%20and%20visualizing%20the%20data.

https://aws.amazon.com/what-is/data-preparation/#:~:text=Data%20preparation%20is%20the%20process,exploring%20and%20visualizing%20the%20data.

Guru99, 25 Mar. 2023, Johnson, Daniel. “What Is Data Analysis? Research, Types & Example.” www.guru99.com/what-is-data-analysis.html#:~:text=Data%20analysis%20is%20defined%20as,based%20upon%20the%20data%20analysis.

https://www.guru99.com/what-is-data-analysis.html#:~:text=Data%20analysis%20is%20defined%20as,based%20upon%20the%20data%20analysis.

Investopedia, 18 Jan. 2023, Chen, James. “How a Histogram Works to Display Data.” www.investopedia.com/terms/h/histogram.asp.

https://www.investopedia.com/terms/h/histogram.asp