Module Code & Module Title

CC5067NP Smart Data Discovery

Assessment Weightage & Type

60% Individual Coursework

Year and Semester

2nd year, 2nd semester

Student Name: Vamsha Palja Tamu

Group: L2C3

London Met ID: 21050019

College ID: NP04CP4A210106

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 marks of zero will be awarded.

Table of Contents

Table of figures

Table of Tables

1. Introduction

Smart data refers to the process of extracting meaningful insights from vast amounts of an existing raw, passive and unstructured form of data through the use of sophisticated algorithms and techniques to turn it into useful information that we can understand and improve and drive forward (Hosseinpour, et al., 2016).

Data discovery is the process of prying meaningful patterns from data. It does this by collecting data from a variety of sources and then applying advanced analytics to it to identify specific patterns or themes (TIBCO, n.d.). Data is being generated at an unprecedented rate, Data Discovery brings data sources together from different sources and analyses them to uncover deeper patterns, trends and insights, giving organizations a comprehensive view of their data. Such a view provides better insights into the available data, which in turn leads to better-informed decisions that help leaders become more agile in their day-to-day operations, helping them make better decisions that impact the business and help the business thrive toward sustained success and gain a competitive advantage in their industry by adapting to the dynamic market and making better decisions and gain a competitive edge in their industry.

Data could be broadly defined as a collection of raw facts and figures that have no inherent meaning or context. But in a business context, data is the collective information related to a company and its operations, which can be generated from a variety of sources including internal business systems, external databases, social media platforms, statistical information, raw analytical data, customer feedback data, sales figures and other information.

For example, a business may conduct surveys or focus groups to gather information about customer preferences or opinions. Another way to get data is through secondary research, which collects data from existing sources such as industry reports or government statistics.

Data could be broadly defined as a collection of raw facts and figures that have no inherent meaning or context. But in a business context, data is the collective information related to a company and its operations, which can be generated from a variety of sources including internal business systems, external databases, social media platforms, statistical information, raw analytical data, customer feedback data, sales figures and other information (Indeed Editorial Team, 2023). For example, a business may conduct surveys or focus groups to gather information about customer preferences or opinions. Another way to get data is through secondary research, which collects data from existing sources such as industry reports or government statistics.

Data processing refers to the conversion of raw data into meaningful information (Lahore College for Women University, n.d.). Data processing involves many steps: data gathering, data cleansing, data entry, data processing, and output. Data processing involves removing errors and inconsistencies from the datasets and combining data from multiple sources into a single dataset. Output involves using statistical methods and algorithms to identify patterns and trends in the data, and presenting the results of the analysis in a clear and concise manner.

Data processing is important for several reasons. First, it can help companies make informed decisions based on accurate and up-to-date information. By processing and analyzing data, companies can gain insights into customer behavior, market trends, and operational efficiencies that can inform organizations to develop better business strategies through strategic planning and decision-making to increase their competitive advantage over others. Finally, data processing can help companies improve their bottom line by identifying areas for cost savings or revenue increases.

Data processing is a crucial component of any successful business strategy. By using data to generate insights and make informed decisions, companies can gain a competitive advantage in their industry and achieve sustainable success. Therefore, investing in data processing tools and techniques is essential for companies looking to stay on top in today's data-driven economy.

The data processing cycle is a series of steps that are used to convert raw data into useful information (Mathur, 2022). These steps include data gathering, data preparing, entering data, data processing, outputting data, and storing data.

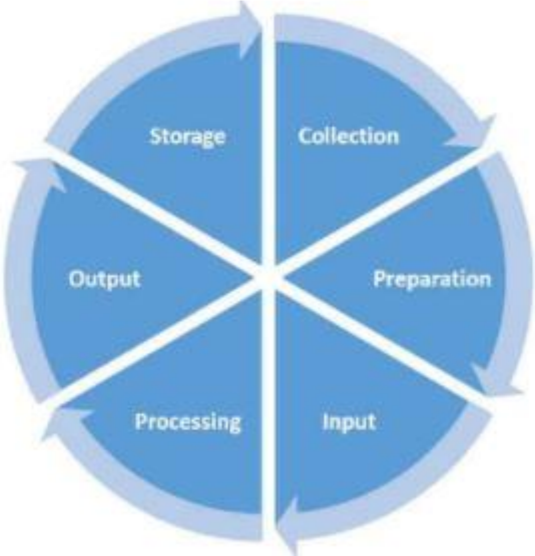


Figure 1: image of data processing cycle

In the data processing cycle, the first step is to collect data from various sources for processing. The second step is to clean the dataset to remove errors and inconsistencies from the dataset. The third step is to merge different datasets into one large dataset for easier processing. The fourth step is to enter the raw collection of the data set ready for further processing. The fifth step is to put the huge collection of raw data into a meaningful form. The sixth step is to get the output of valuable information from processing the data set. Finally, the output data set is stored for later future use, for easy access and retrieval of information when needed, as well as for immediate use as input in the next data processing cycle.

The data processing cycle is a critical component of any successful business strategy. By using data to generate insights and create better business strategies through strategic planning and decision making, companies can gain a competitive advantage in their industry and achieve sustained success. Therefore, investing in data processing tools and techniques is essential for companies looking to stay on top in today's data-driven economy.

2. Objectives

- to learn how to prepare a dataset for Businesses Analytics and machine learners
- to develop critical thinking skills, to better solve problems that may arise during dataset preparation
- to work as a data scientist for companies, to analyze data for companies to increase their competitive advantage Others

3. Development

❖ Data understanding

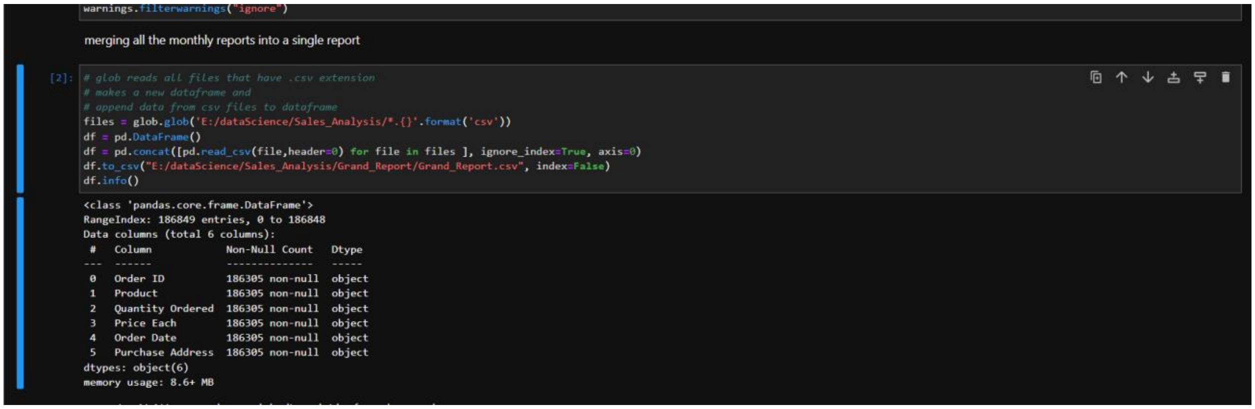
Using data from company ABC, it contained the sales data for each month of 2019, which products were sold to the customer, in what quantity, on what date and time, and where to ship them. It contained attributes such as order ID, product, order quantity, price per unit, order date and purchase address. There were some mixed null values in the data set.

s.no	Column name	Description	Data type
1	Order ID	This column contains the id number of orders that was made by the customer	Object
2	Product	This column contains the names of product that were sold	Object
3	Quantity Ordered	This column contains the quantity of products that was ordered by the customer	Object
4	Price Each	This column contains the price for that product	Object
5	Order Date	This column contains the date and time of when the order was made.	Object
6	Purchase Address	This column contains customer’s personal address that the product was to be delivered	Object

Table 1: describing the columns present in csv files

❖ Data preparation

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



```
warnings.filterwarnings("ignore")
```

merging all the monthly reports into a single report

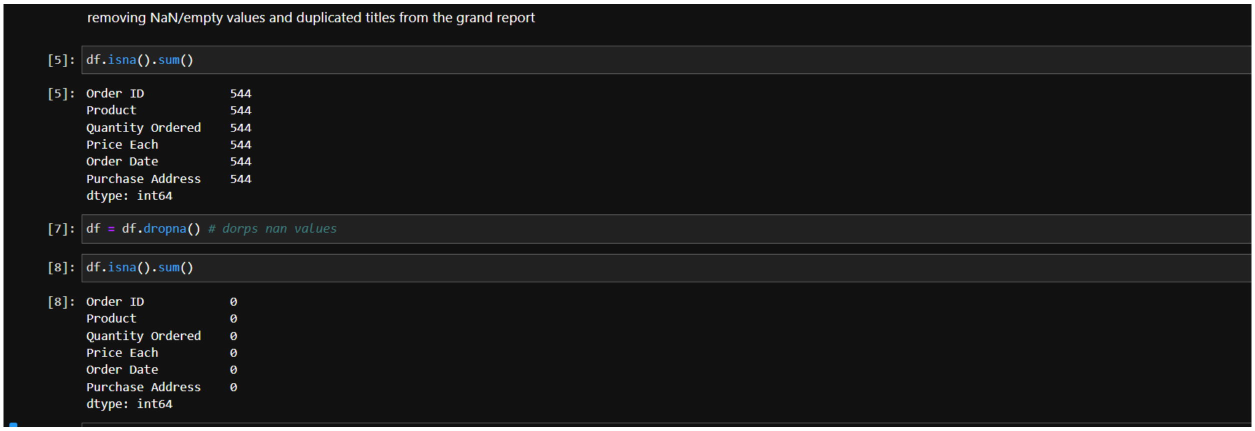
```
[2]: # glob reads all files that have .csv extension
     # makes a new dataframe and
     # append data from csv files to dataframe
     files = glob.glob('E:/dataScience/Sales_Analysis/*.{}'.format('csv'))
     df = pd.DataFrame()
     df = pd.concat([pd.read_csv(file,header=0) for file in files ], ignore_index=True, axis=0)
     df.to_csv("E:/dataScience/Sales_Analysis/Grand_Report/Grand_Report.csv", index=False)
     df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 186849 entries, 0 to 186848
Data columns (total 6 columns):
 #   Column            Non-Null Count   Dtype 
---  ------            --------------   ----- 
 0   Order ID          186305 non-null  object
 1   Product           186305 non-null  object
 2   Quantity Ordered  186305 non-null  object
 3   Price Each        186305 non-null  object
 4   Order Date        186305 non-null  object
 5   Purchase Address  186305 non-null  object
dtypes: object(6)
memory usage: 8.6+ MB
```

Figure 2: merging all monthly reports into one final report

As shown in the above picture, monthly reports are being converted into one single grand file named “Grand_Report” and its format is CSV.

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



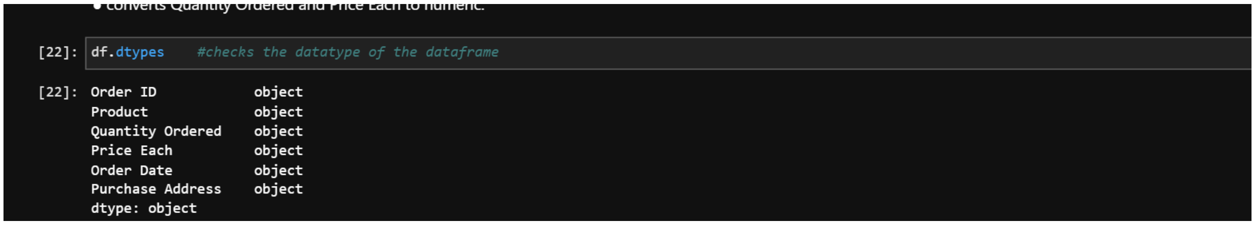
removing NaN/empty values and duplicated titles from the grand report

```
[5]: df.isna().sum()

[5]: Order ID            544
     Product             544
     Quantity Ordered    544
     Price Each          544
     Order Date          544
     Purchase Address    544
     dtype: int64

[7]: df = df.dropna() # dorps nan values

[8]: df.isna().sum()

[8]: Order ID            0
     Product             0
     Quantity Ordered    0
     Price Each          0
     Order Date          0
     Purchase Address    0
     dtype: int64
```

Figure 3: checked if there was null values and removed them

In above block of code, null values are checked and it there are any then it is removed from the file.

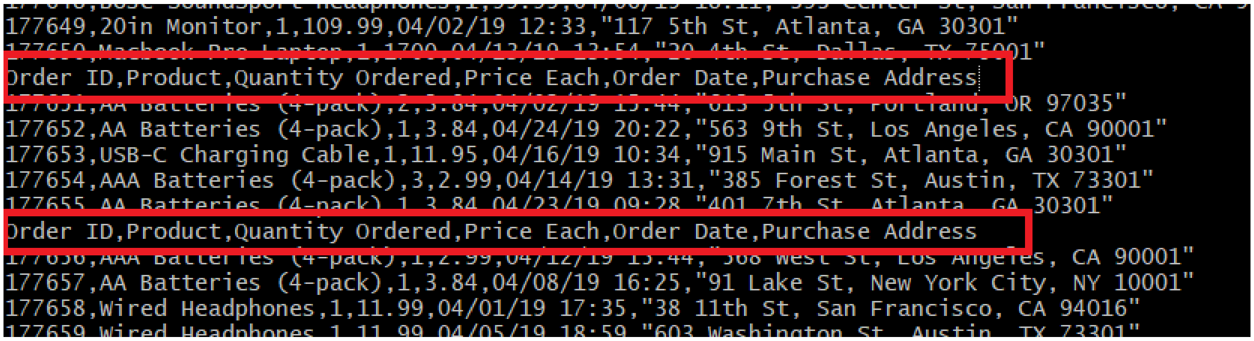
- Write a python program to convert **Quantity Ordered** and **Price Each** to numeric.



• converts Quantity Ordered and Price Each to numeric.

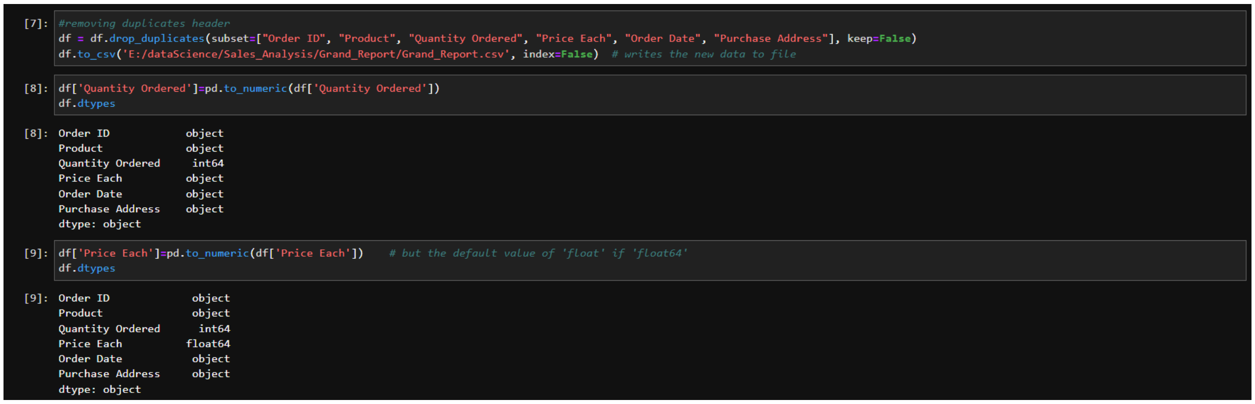
```
[22]: df.dtypes    #checks the datatype of the dataframe

[22]: Order ID            object
      Product             object
      Quantity Ordered    object
      Price Each          object
      Order Date          object
      Purchase Address    object
      dtype: object
```

Figure 4: checking the data type of columns



```
177648,Bose SoundSport Headphones,1,99.99,04/06/19 16:11,"555 Center St, San Francisco, CA 9
177649,20in Monitor,1,109.99,04/02/19 12:33,"117 5th St, Atlanta, GA 30301"
177650,Macbook Pro Laptop,1,1700,04/13/19 13:54,"20 4th St, Dallas, TX 75001"
Order ID,Product,Quantity Ordered,Price Each,Order Date,Purchase Address
177651,AA Batteries (4-pack),2,3.84,04/02/19 15:44,"613 5th St, Portland, OR 97035"
177652,AA Batteries (4-pack),1,3.84,04/24/19 20:22,"563 9th St, Los Angeles, CA 90001"
177653,USB-C Charging Cable,1,11.95,04/16/19 10:34,"915 Main St, Atlanta, GA 30301"
177654,AAA Batteries (4-pack),3,2.99,04/14/19 13:31,"385 Forest St, Austin, TX 73301"
177655,AA Batteries (4-pack),1,3.84,04/23/19 09:28,"401 7th St, Atlanta, GA 30301"
Order ID,Product,Quantity Ordered,Price Each,Order Date,Purchase Address
177656,AAA Batteries (4-pack),1,2.99,04/12/19 15:44,"368 West St, Los Angeles, CA 90001"
177657,AA Batteries (4-pack),1,3.84,04/08/19 16:25,"91 Lake St, New York City, NY 10001"
177658,Wired Headphones,1,11.99,04/01/19 17:35,"38 11th St, San Francisco, CA 94016"
177659,Wired Headphones,1,11.99,04/05/19 18:59,"603 Washington St, Austin, TX 73301"
```

Figure 5: checked if there were any repeating headings

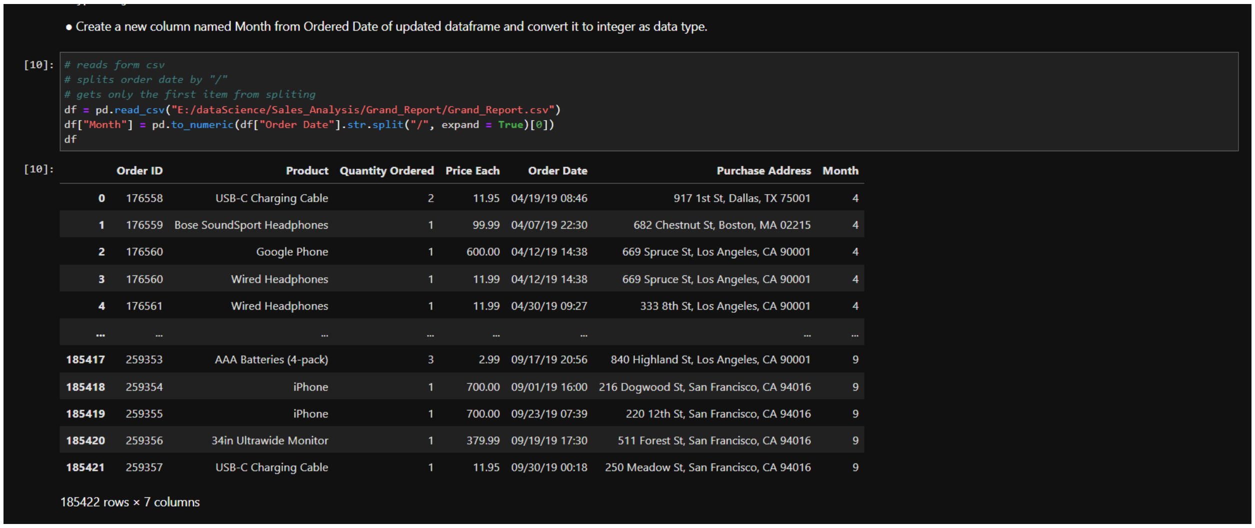


```
[7]: #removing duplicates header
     df = df.drop_duplicates(subset=["Order ID", "Product", "Quantity Ordered", "Price Each", "Order Date", "Purchase Address"], keep=False)
     df.to_csv('E:/dataScience/Sales_Analysis/Grand_Report/Grand_Report.csv', index=False)  # writes the new data to file

[8]: df['Quantity Ordered']=pd.to_numeric(df['Quantity Ordered'])
     df.dtypes

[8]: Order ID            object
     Product             object
     Quantity Ordered     int64
     Price Each          object
     Order Date          object
     Purchase Address    object
     dtype: object

[9]: df['Price Each']=pd.to_numeric(df['Price Each'])    # but the default value of 'float' if 'float64'
     df.dtypes

[9]: Order ID             object
     Product              object
     Quantity Ordered      int64
     Price Each          float64
     Order Date           object
     Purchase Address     object
     dtype: object
```

Figure 6: removing the duplicated headers and converting the respective columns to integer

Because there were duplicated heading kept as string, we needed to remove the duplicated headings first before converting the columns to integer as they were numbers.

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



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

[10]:

```python
# reads form csv
# splits order date by "/"
# gets only the first item from spliting
df = pd.read_csv("E:/dataScience/Sales_Analysis/Grand_Report/Grand_Report.csv")
df["Month"] = pd.to_numeric(df["Order Date"].str.split("/", expand = True)[0])
df
```

[10]:

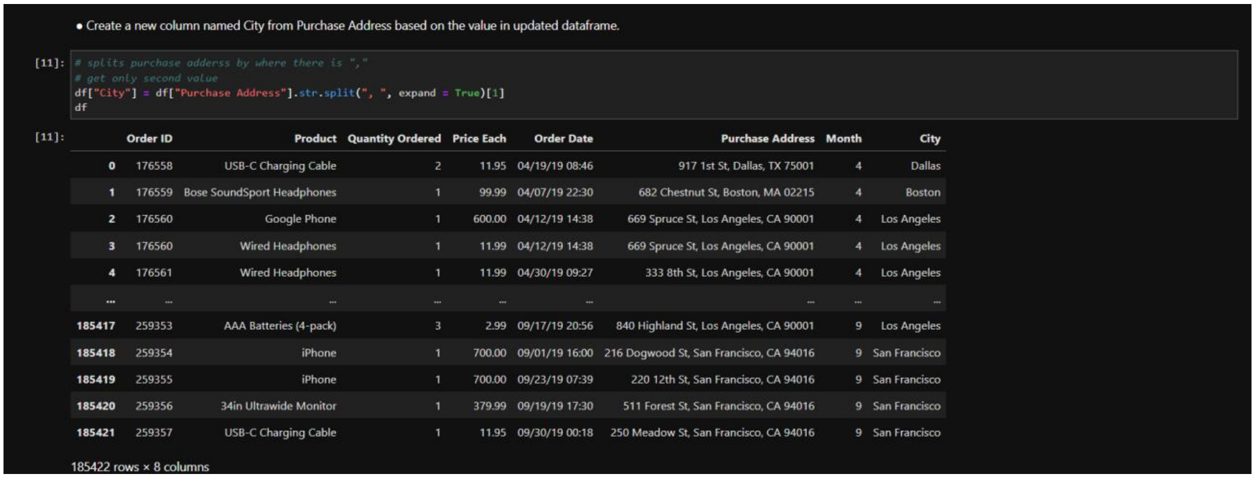
	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address	Month
0	176558	USB-C Charging Cable	2	11.95	04/19/19 08:46	917 1st St, Dallas, TX 75001	4
1	176559	Bose SoundSport Headphones	1	99.99	04/07/19 22:30	682 Chestnut St, Boston, MA 02215	4
2	176560	Google Phone	1	600.00	04/12/19 14:38	669 Spruce St, Los Angeles, CA 90001	4
3	176560	Wired Headphones	1	11.99	04/12/19 14:38	669 Spruce St, Los Angeles, CA 90001	4
4	176561	Wired Headphones	1	11.99	04/30/19 09:27	333 8th St, Los Angeles, CA 90001	4
...	...	...	...	...	...	...	...
185417	259353	AAA Batteries (4-pack)	3	2.99	09/17/19 20:56	840 Highland St, Los Angeles, CA 90001	9
185418	259354	iPhone	1	700.00	09/01/19 16:00	216 Dogwood St, San Francisco, CA 94016	9
185419	259355	iPhone	1	700.00	09/23/19 07:39	220 12th St, San Francisco, CA 94016	9
185420	259356	34in Ultrawide Monitor	1	379.99	09/19/19 17:30	511 Forest St, San Francisco, CA 94016	9
185421	259357	USB-C Charging Cable	1	11.95	09/30/19 00:18	250 Meadow St, San Francisco, CA 94016	9

185422 rows × 7 columns

Figure 7: splitting the month from ordered date and converted to integer

In the above picture, month is being extracted from “Ordered Date” and is being changed into integer as a number that indicates month for easier unerstanding.

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



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

[11]:
```python
# splits purchase adderss by where there is ","
# get only second value
df["City"] = df["Purchase Address"].str.split(", ", expand = True)[1]
df
```

[11]:

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address	Month	City
0	176558	USB-C Charging Cable	2	11.95	04/19/19 08:46	917 1st St, Dallas, TX 75001	4	Dallas
1	176559	Bose SoundSport Headphones	1	99.99	04/07/19 22:30	682 Chestnut St, Boston, MA 02215	4	Boston
2	176560	Google Phone	1	600.00	04/12/19 14:38	669 Spruce St, Los Angeles, CA 90001	4	Los Angeles
3	176560	Wired Headphones	1	11.99	04/12/19 14:38	669 Spruce St, Los Angeles, CA 90001	4	Los Angeles
4	176561	Wired Headphones	1	11.99	04/30/19 09:27	333 8th St, Los Angeles, CA 90001	4	Los Angeles
...	...	...	...	...	...	...	...	...
185417	259353	AAA Batteries (4-pack)	3	2.99	09/17/19 20:56	840 Highland St, Los Angeles, CA 90001	9	Los Angeles
185418	259354	iPhone	1	700.00	09/01/19 16:00	216 Dogwood St, San Francisco, CA 94016	9	San Francisco
185419	259355	iPhone	1	700.00	09/23/19 07:39	220 12th St, San Francisco, CA 94016	9	San Francisco
185420	259356	34in Ultrawide Monitor	1	379.99	09/19/19 17:30	511 Forest St, San Francisco, CA 94016	9	San Francisco
185421	259357	USB-C Charging Cable	1	11.95	09/30/19 00:18	250 Meadow St, San Francisco, CA 94016	9	San Francisco

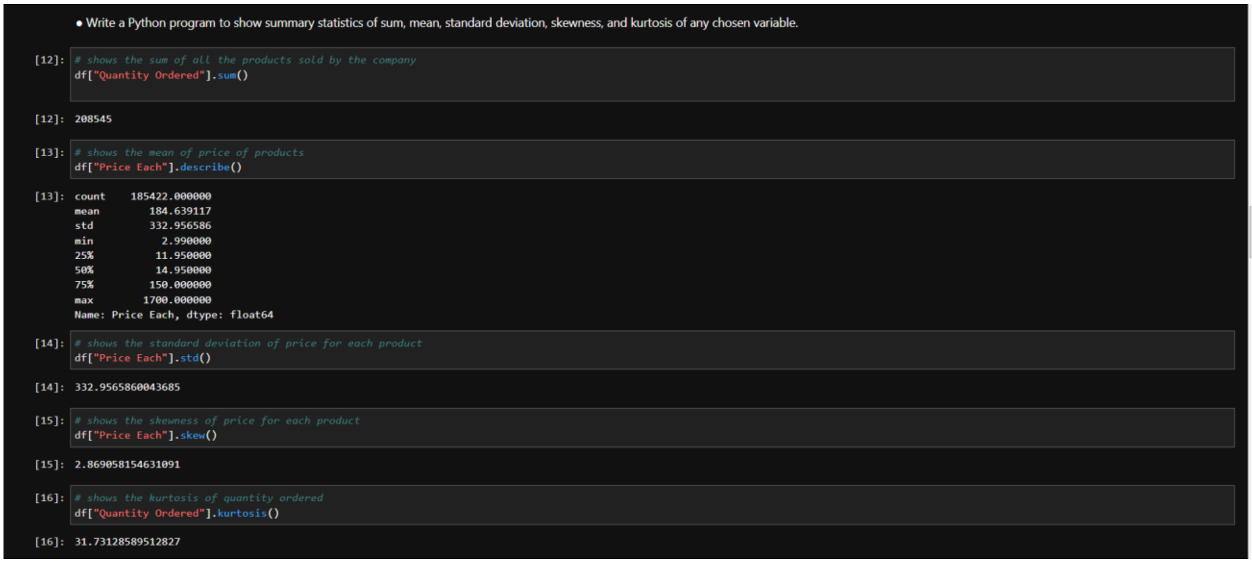
185422 rows × 8 columns

Figure 8: extracting city name from purchased address

In the above piece of code, city name is being extracted from purchased address and kept at another column.

❖ Data Analysis

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

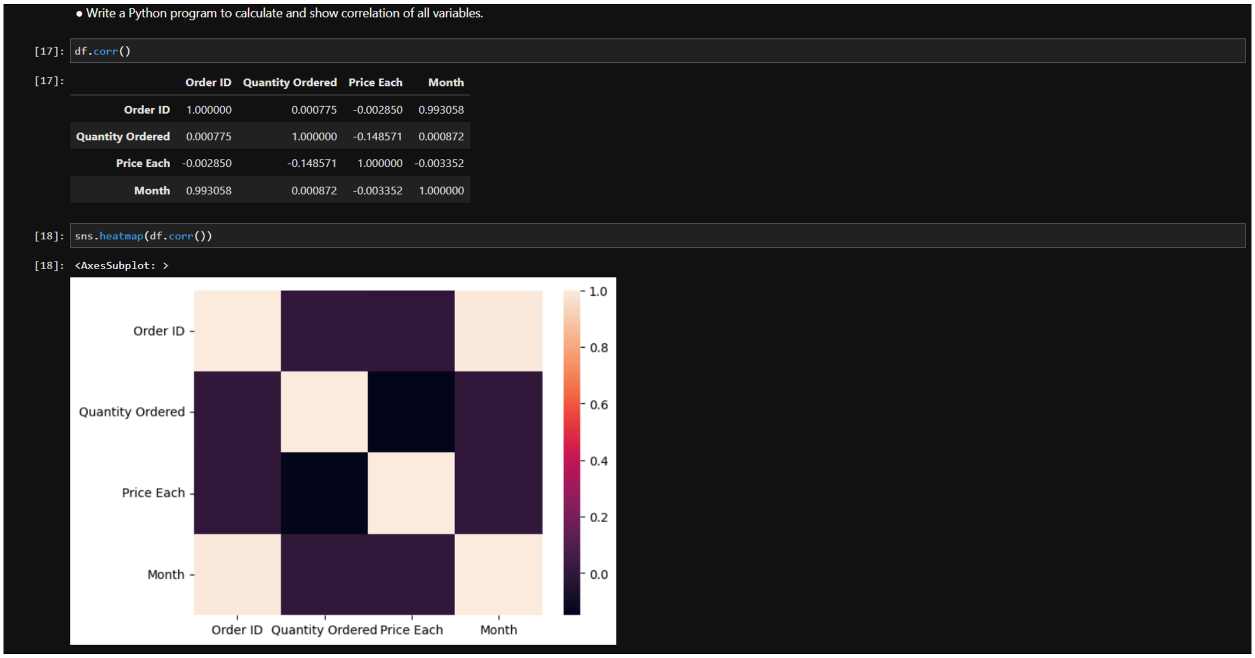


• Write a Python program to show summary statistics of sum, mean, standard deviation, skewness, and kurtosis of any chosen variable.

```
[12]: # shows the sum of all the products sold by the company
      df["Quantity Ordered"].sum()

[12]: 208545

[13]: # shows the mean of price of products
      df["Price Each"].describe()

[13]: count    185422.000000
      mean        184.639117
      std         332.956586
      min           2.990000
      25%          11.950000
      50%          14.950000
      75%         150.000000
      max        1700.000000
      Name: Price Each, dtype: float64

[14]: # shows the standard deviation of price for each product
      df["Price Each"].std()

[14]: 332.9565860043685

[15]: # shows the skewness of price for each product
      df["Price Each"].skew()

[15]: 2.869058154631091

[16]: # shows the kurtosis of quantity ordered
      df["Quantity Ordered"].kurtosis()

[16]: 31.73128589512827
```

Figure 9: statistics of sum, mean, standard deviation, skewness, and kurtosis

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



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

```
[17]: df.corr()
```

```
[17]:
                   Order ID  Quantity Ordered  Price Each     Month
Order ID           1.000000          0.000775   -0.002850  0.993058
Quantity Ordered   0.000775          1.000000   -0.148571  0.000872
Price Each        -0.002850         -0.148571    1.000000 -0.003352
Month              0.993058          0.000872   -0.003352  1.000000
```

```
[18]: sns.heatmap(df.corr())
```

```
[18]: <AxesSubplot: >
```

Figure 10: calculating correlation of all variables

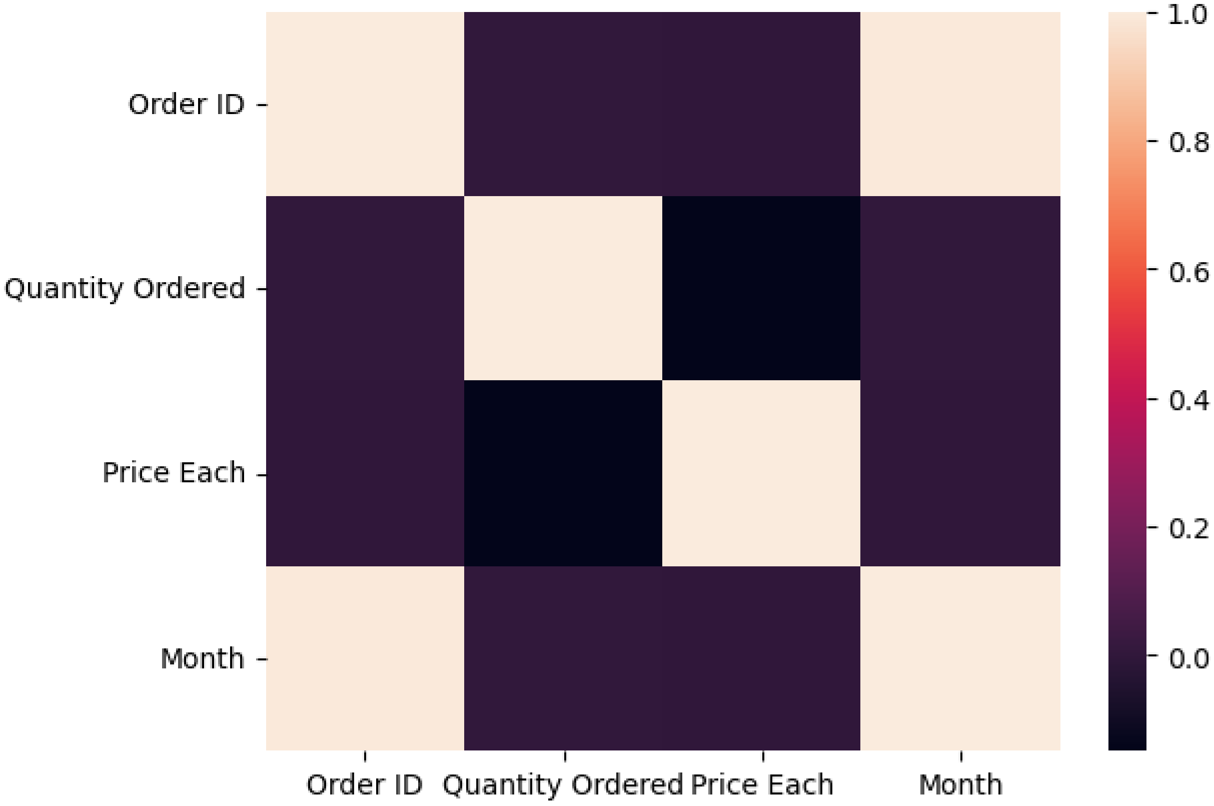
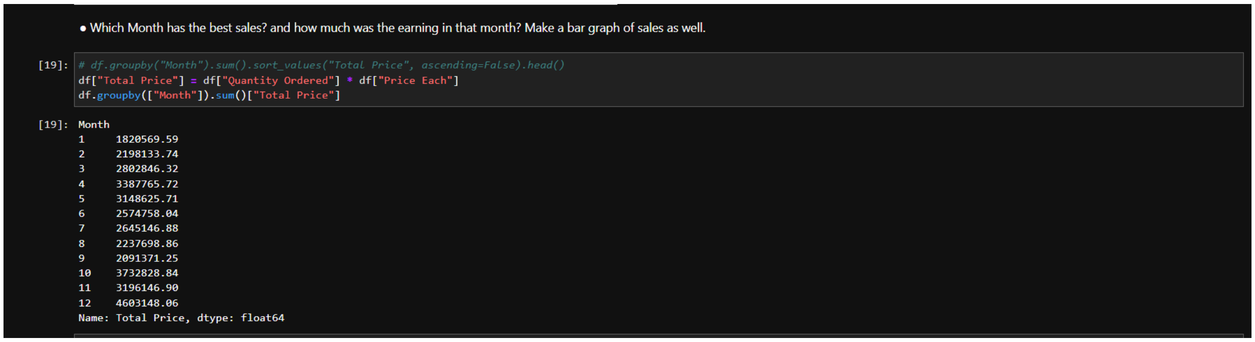


Figure 11: showing correlation of all variables

In the above picture presented, correlation of all variables is being calculated and is being represented on a graph.

❖ Data Exploration

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

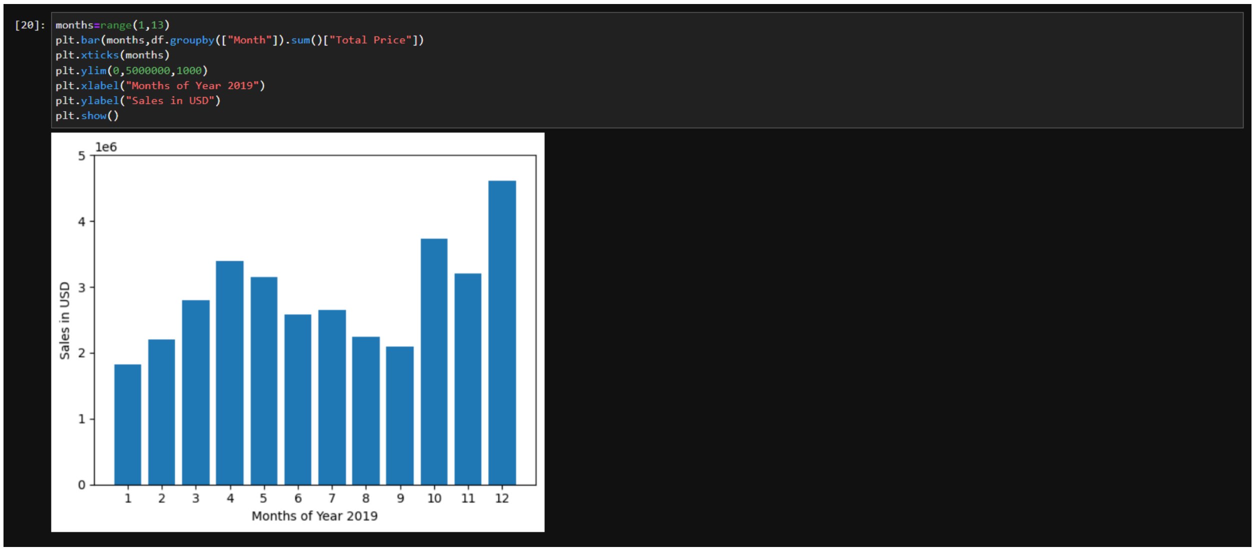


- Which Month has the best sales? and how much was the earning in that month? Make a bar graph of sales as well.

```
[19]: # df.groupby("Month").sum().sort_values("Total Price", ascending=False).head()
      df["Total Price"] = df["Quantity Ordered"] * df["Price Each"]
      df.groupby(["Month"]).sum()["Total Price"]
```

```
[19]: Month
      1     1820569.59
      2     2198133.74
      3     2802846.32
      4     3387765.72
      5     3148625.71
      6     2574758.04
      7     2645146.88
      8     2237698.86
      9     2091371.25
      10    3732828.84
      11    3196146.90
      12    4603148.06
      Name: Total Price, dtype: float64
```

Figure 12: which month had the best sale and what was the earning



```
[20]: months=range(1,13)
      plt.bar(months,df.groupby(["Month"]).sum()["Total Price"])
      plt.xticks(months)
      plt.ylim(0,5000000,1000)
      plt.xlabel("Months of Year 2019")
      plt.ylabel("Sales in USD")
      plt.show()
```

Figure 13: best sale being represented on a graph

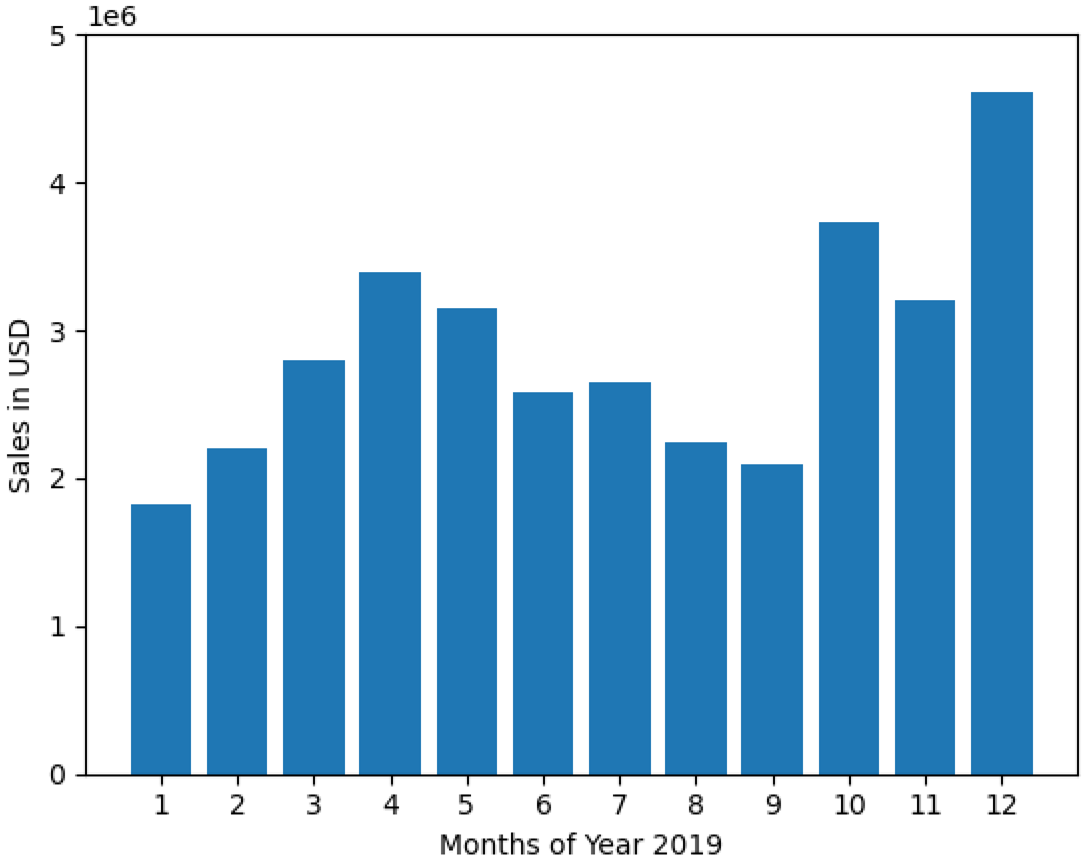
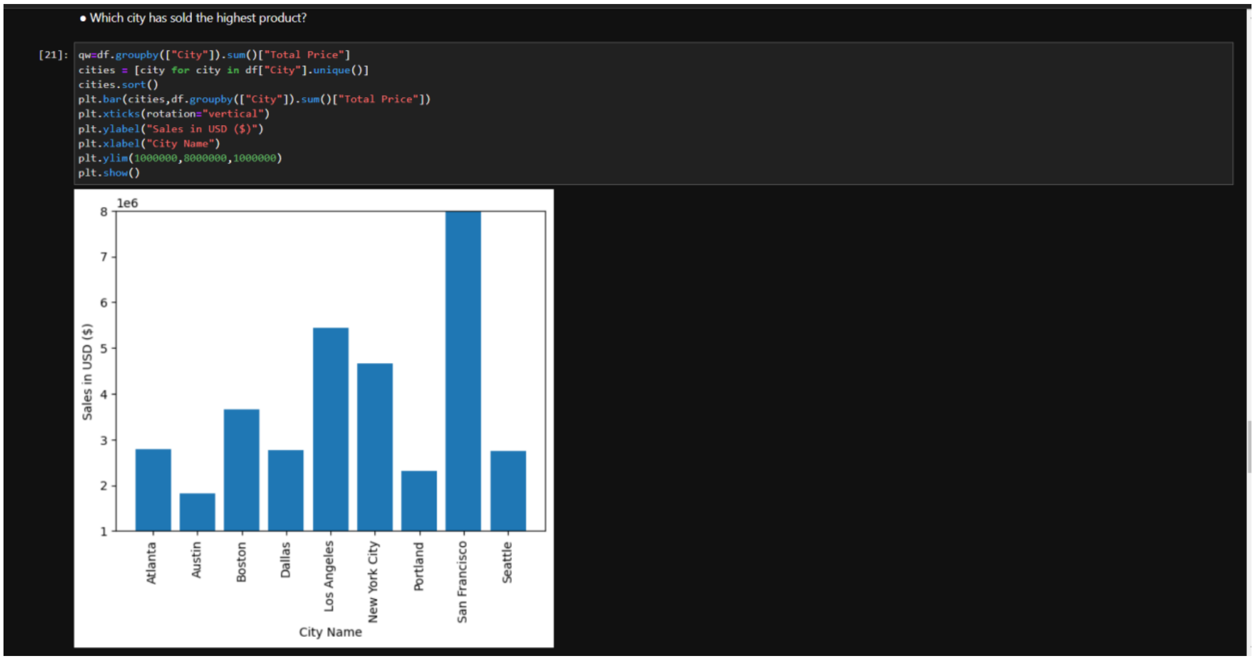


Figure 14: graph on which month had the highest sale

In the above picture, which month had the best sale is being calculated and represented on a graph.

- Which city has sold the highest product?



- Which city has sold the highest product?

```
[21]: qw=df.groupby(["City"]).sum()["Total Price"]
      cities = [city for city in df["City"].unique()]
      cities.sort()
      plt.bar(cities,df.groupby(["City"]).sum()["Total Price"])
      plt.xticks(rotation="vertical")
      plt.ylabel("Sales in USD ($)")
      plt.xlabel("City Name")
      plt.ylim(1000000,8000000,1000000)
      plt.show()
```

Figure 15: which city has highest product sold

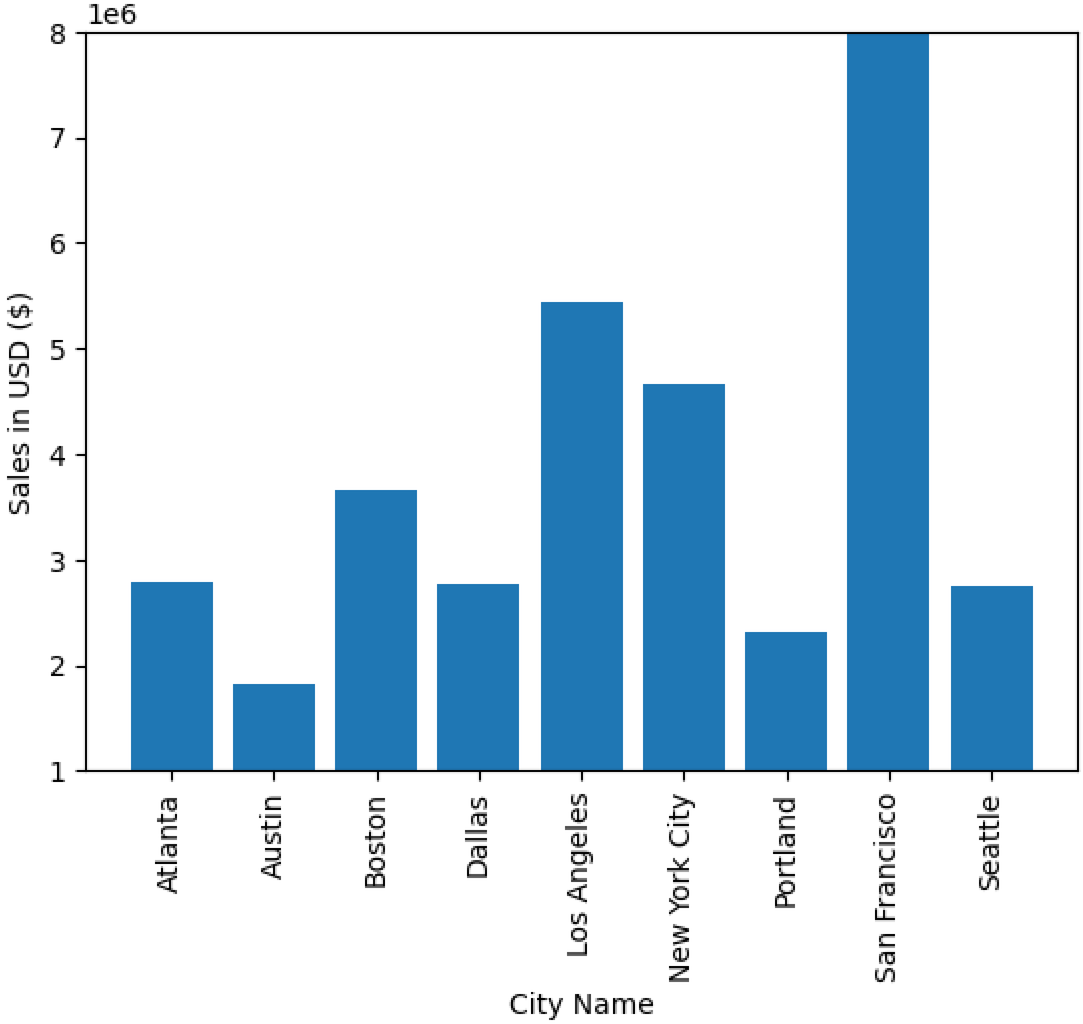
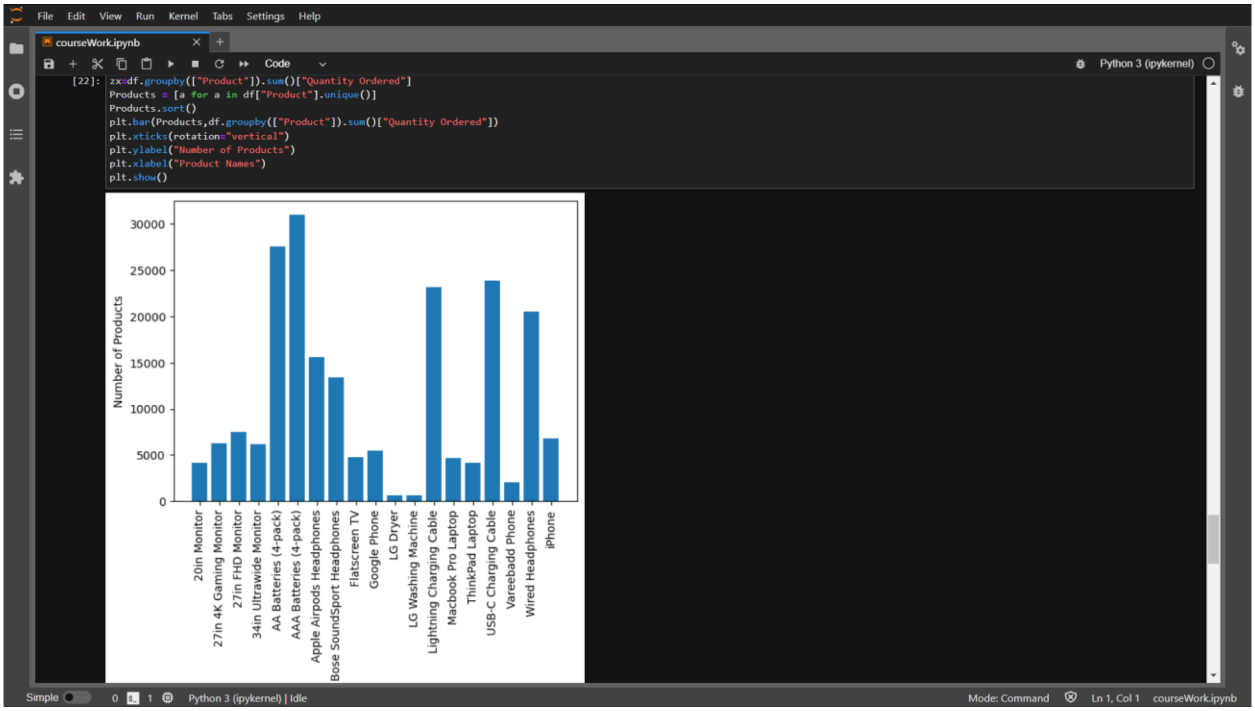


Figure 16: graph on which city had sold the most product

In the above pictures, which city had the highest product sale was calculated and represented on a graph.

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



```
[22]: zx=df.groupby(["Product"]).sum()["Quantity Ordered"]
      Products = [a for a in df["Product"].unique()]
      Products.sort()
      plt.bar(Products,df.groupby(["Product"]).sum()["Quantity Ordered"])
      plt.xticks(rotation="vertical")
      plt.ylabel("Number of Products")
      plt.xlabel("Product Names")
      plt.show()
```

Figure 17: which product was sold most being calculated

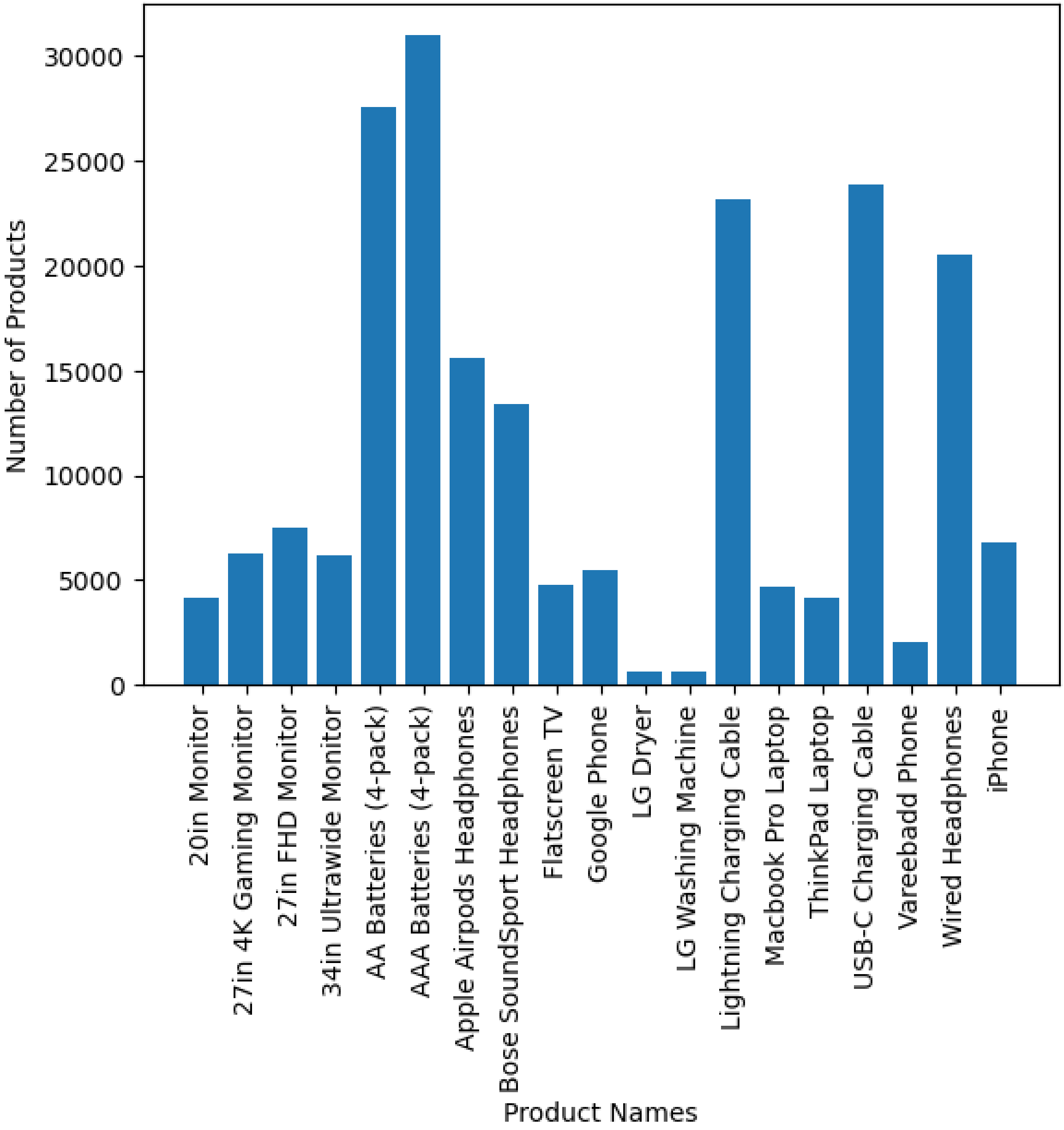
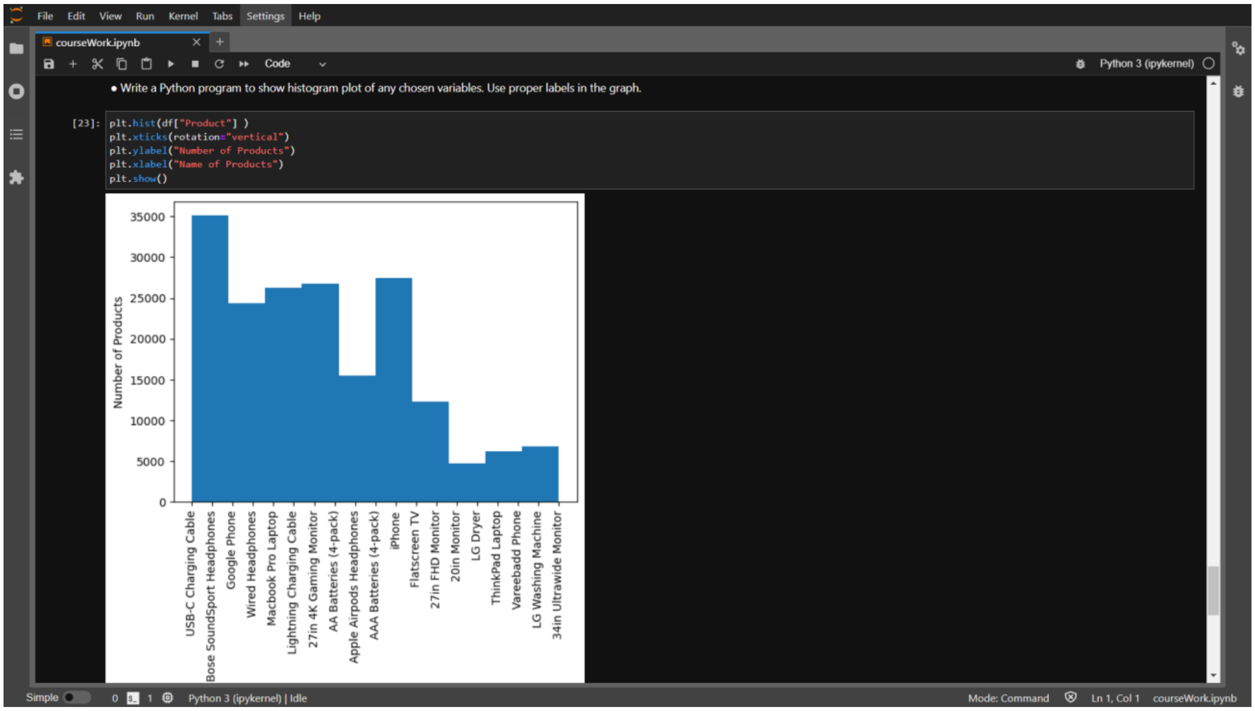


Figure 18: graph on which product was sold the most

In the above pictures, which product was sold most is calculated and represented on graph.

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



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

```
[23]: plt.hist(df["Product"] )
      plt.xticks(rotation="vertical")
      plt.ylabel("Number of Products")
      plt.xlabel("Name of Products")
      plt.show()
```

Figure 19: calculating the numbers of products that were sold

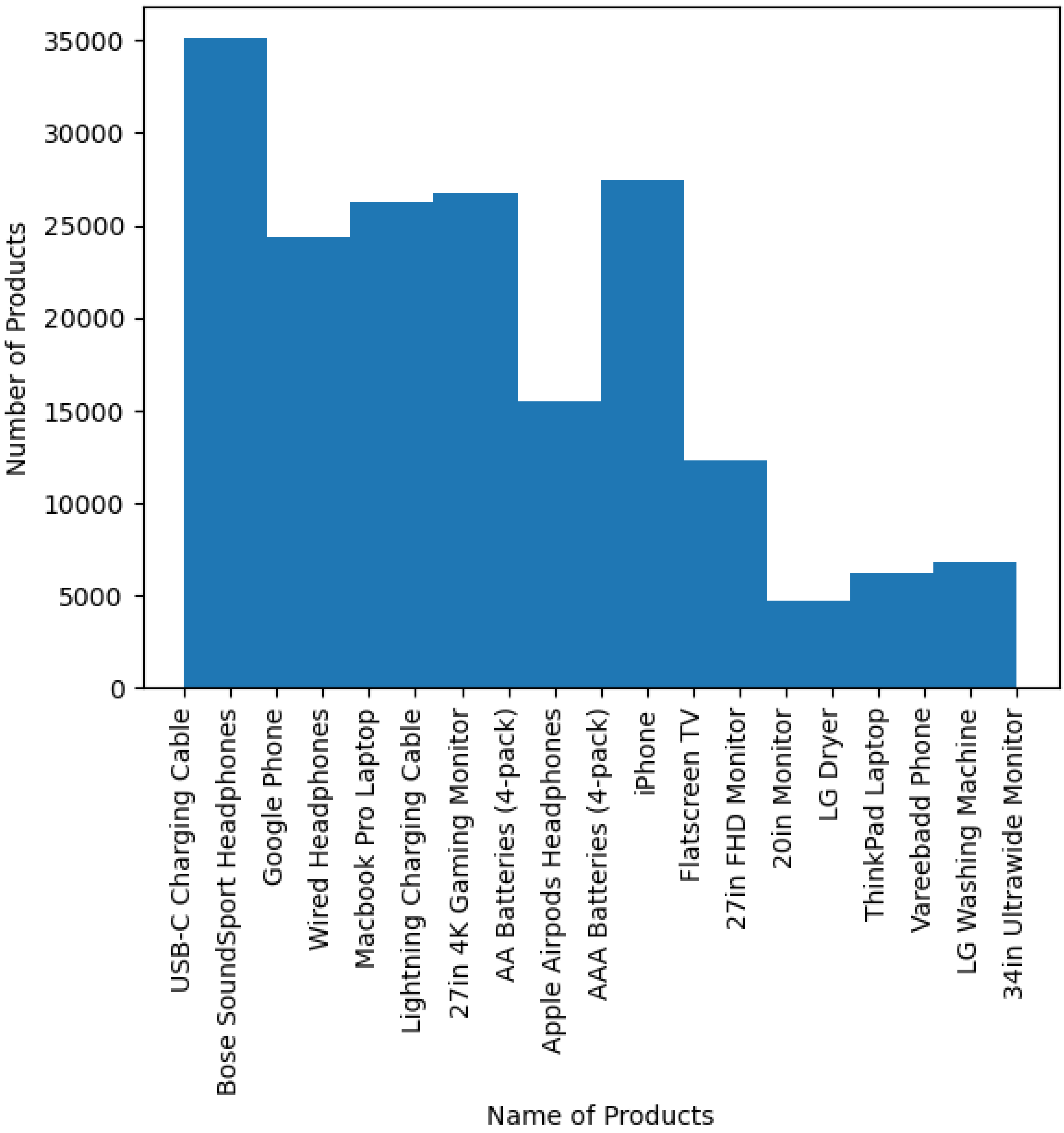


Figure 20: histogram graph on which product was sold the most

In the above pictures, which product was sold most is being calculated and represented on histogram graph.

4. Conclusion

Through much effort, determination and research, the coursework was completed. This was an individual job that required us to process raw business data into information that was insightful for the company to achieve sustained success through an advantage over other competitors.

A lot of research was needed to get the right result. When I tried to convert the columns to integer and float respectively, I encountered a problem which meant that the column's data type could not be converted. I had searched the internet why I was getting this error when running a script on a Jupyter notebook and was also trying to figure out why. I had set up a few others who were also having the same problem and they were able to solve it online with the help of a stranger. I tried to implement the same script and still couldn't solve it. Unable to find the cause I dug into the CSV file and found that many headings kept repeating and since there was a string in the middle of the integer data and after removing the repeating heading I was able to convert the columns to integer and float.

While doing the scripting for the automation, I was able to get a new idea of how I might use Python or other data processing programs. It also expands my knowledge of business analysis and how to use it to gain an advantage over other competitors. I've been able to learn a lot about Python, and this is what they are: I've been able to learn about various libraries that make computation easier, like Pandas and NumPy, learn about list comprehension, and implement it in scripting to see the imminent result it produces, and others.

In summary, the Data Processing and Business Analysis coursework was a valuable learning experience that required significant effort, determination, and research. It was an individual task that required processing raw business data into meaningful information that could be used to achieve competitive advantage and sustained success.

There were challenges to overcome throughout the project, such as converting columns to integers and floats, but these have been overcome through research and creative thinking. The project also provided an opportunity to learn more about Python and its libraries like Pandas and Numpy, which made data processing more efficient. It expanded my knowledge of business analytics and how it can be used to make informed decisions that drive success in a competitive business landscape. Overall, this coursework demonstrated the importance of data processing and business analysis in achieving and sustaining success in the business world.

5. References

Hosseinpour, F., Plosila, J. & Tenhunen, H., 2016. Smart Data: A New Perspective of Tackling the Big Data Phenomena Leveraging a Fog Computing System. *Smart Data: A New Perspective of Tackling the Big Data Phenomena Leveraging a Fog Computing System,* Volume 10, p. 124.

Indeed Editorial Team, 2023. *Indeed.* [Online] Available at: https://www.indeed.com/career-advice/career-development/data-in-business#:~:text=Business%20data%20is%20the%20collective,and%20other%20sets%20of%20information.
[Accessed 3 April 2023].

Lahore College for Women University, n.d. *4DATAPROCESSING-converted.pdf.* [Online] Available at: https://ocd.lcwu.edu.pk/cfiles/Pharmacy%20Practice/Paper%20VIII/4DATAPROCESSING-converted.pdf

Mathur, V., 2022. *What is Data Processing and Why is it Important?.* [Online] Available at: https://www.analyticssteps.com/blogs/what-data-processing-and-why-it-important

TIBCO, n.d. *What is Data Discovery?.* [Online] Available at: https://www.tibco.com/reference-center/what-is-data-discovery
[Accessed 12 04 2023].