sales-data-project

August 4, 2023

1 Sales Analysis

Libraries

[1]:
```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import seaborn as sns
import os
```

Merge 12 months of sales data into a single file

[2]:
```
month = pd.read_csv("/Datasets/Sales_Data/Sales_April_2019.csv")
month.head()
```

[2]:

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address
0	176558	USB-C Charging Cable	2	11.95	04/19/19 08:46	917 1st St, Dallas, TX 75001
1	NaN	NaN	NaN	NaN	NaN	NaN
2	176559	Bose SoundSport Headphones	1	99.99	04/07/19 22:30	682 Chestnut St, Boston, MA 02215
3	176560	Google Phone	1	600	04/12/19 14:38	669 Spruce St, Los Angeles, CA 90001
4	176560	Wired Headphones	1	11.99	04/12/19 14:38	669 Spruce St, Los Angeles, CA 90001

[3]:
```
files = [file for file in os.listdir('/Datasets/Sales_Data/')]

df = pd.DataFrame()
for file in files:
    month = pd.read_csv("/Datasets/Sales_Data/"+file)
    df = pd.concat([df,month])

df.to_csv("all_data.csv",index=False)
```

Read updated dataframe

[4]:
```
df = pd.read_csv("all_data.csv")
df.head(3)
```

[4]:
```
  Order ID                     Product Quantity Ordered Price Each  \
0   176558        USB-C Charging Cable                2      11.95
1      NaN                         NaN              NaN        NaN
2   176559  Bose SoundSport Headphones                1      99.99

       Order Date                   Purchase Address
0  04/19/19 08:46       917 1st St, Dallas, TX 75001
1             NaN                                NaN
2  04/07/19 22:30  682 Chestnut St, Boston, MA 02215
```

1.1 Clean up the data

[5]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 186850 entries, 0 to 186849
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
```

1.1.1 Drop rows of NAN

[6]:
```
nan_df = df[df.isna().any(axis=1)]

df= df.dropna(how='all')
```

1.1.2 find 'or" and delete it

[7]:
```
df["Order Date"].str[0:2].value_counts()
```

[7]:
```
Order Date
12    24984
10    20282
04    18279
11    17573
05    16566
```

```
03    15153
07    14293
06    13554
02    11975
08    11961
09    11621
01     9709
Or      355
Name: count, dtype: int64
```

[8]:
```
temp = df[df["Order Date"].str[0:2]=="Or"]
temp.head()
```

[8]:
```
      Order ID  Product  Quantity Ordered  Price Each  Order Date  \
519   Order ID  Product  Quantity Ordered  Price Each  Order Date
1149  Order ID  Product  Quantity Ordered  Price Each  Order Date
1155  Order ID  Product  Quantity Ordered  Price Each  Order Date
2878  Order ID  Product  Quantity Ordered  Price Each  Order Date
2893  Order ID  Product  Quantity Ordered  Price Each  Order Date

      Purchase Address
519   Purchase Address
1149  Purchase Address
1155  Purchase Address
2878  Purchase Address
2893  Purchase Address
```

[9]:
```
df = temp = df[df["Order Date"].str[0:2]!="Or"]
df.head()
```

[9]:
```
  Order ID                    Product Quantity Ordered Price Each  \
0   176558        USB-C Charging Cable                2      11.95
2   176559  Bose SoundSport Headphones                1      99.99
3   176560                Google Phone                1        600
4   176560            Wired Headphones                1      11.99
5   176561            Wired Headphones                1      11.99

       Order Date                      Purchase Address
0  04/19/19 08:46          917 1st St, Dallas, TX 75001
2  04/07/19 22:30     682 Chestnut St, Boston, MA 02215
3  04/12/19 14:38  669 Spruce St, Los Angeles, CA 90001
4  04/12/19 14:38  669 Spruce St, Los Angeles, CA 90001
5  04/30/19 09:27     333 8th St, Los Angeles, CA 90001
```

Convert columns to correct type

[11]:
```
df.dtypes
```

[11]:
```
Order ID            object
Product             object
Quantity Ordered    object
Price Each          object
Order Date          object
Purchase Address    object
dtype: object
```

[12]:
```
df["Quantity Ordered"] = pd.to_numeric(df["Quantity Ordered"])
df["Price Each"] = pd.to_numeric(df["Price Each"])
df.head(3)
```

[12]:

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address
0	176558	USB-C Charging Cable	2	11.95	04/19/19 08:46	917 1st St, Dallas, TX 75001
2	176559	Bose SoundSport Headphones	1	99.99	04/07/19 22:30	682 Chestnut St, Boston, MA 02215
3	176560	Google Phone	1	600.00	04/12/19 14:38	669 Spruce St, Los Angeles, CA 90001

1.2 Augment data with additional columns

1.2.1 Task 2: Add Month Column

[13]:
```
df["Month"] = df["Order Date"].str[0:2]
df["Month"]=df["Month"].astype('int32')
df.head()
```

[13]:

	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
2	176559	Bose SoundSport Headphones	1	99.99	04/07/19 22:30	682 Chestnut St, Boston, MA 02215	4
3	176560	Google Phone	1	600.00	04/12/19 14:38	669 Spruce St, Los Angeles, CA 90001	4
4	176560	Wired Headphones	1	11.99	04/12/19 14:38	669 Spruce St, Los Angeles, CA 90001	4
5	176561	Wired Headphones	1	11.99	04/30/19 09:27	333 8th St, Los Angeles, CA 90001	4

[24]:
```
df["Month"].value_counts()
```

[24]:
```
Month
12    24984
10    20282
4     18279
11    17573
5     16566
3     15153
7     14293
6     13554
2     11975
8     11961
9     11621
1      9709
Name: count, dtype: int64
```

1.2.2 Task 3: Add a sales column

[14]:
```
df["Sales"] = df["Quantity Ordered"]* df["Price Each"]
df.head(3)
```

[14]:

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address	Month	Sales
0	176558	USB-C Charging Cable	2	11.95	04/19/19 08:46	917 1st St, Dallas, TX 75001	4	23.90
2	176559	Bose SoundSport Headphones	1	99.99	04/07/19 22:30	682 Chestnut St, Boston, MA 02215	4	99.99
3	176560	Google Phone	1	600.00	04/12/19 14:38	669 Spruce St, Los Angeles, CA 90001	4	600.00

Task 4: Add a city column

[15]:
```
def get_city(address):
    return address.split(",")[1]

def get_state(address):
    return address.split(',')[2].split(" ")[1]

df["City"]= df["Purchase Address"].apply(lambda x: f"{get_city(x)}␣
 ↪({get_state(x)})")

df.head(3)
```

[15]:

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address	Month	Sales	City
0	176558	USB-C Charging Cable	2	11.95	04/19/19 08:46	917 1st St, Dallas, TX 75001	4	23.90	Dallas (TX)
2	176559	Bose SoundSport Headphones	1	99.99	04/07/19 22:30	682 Chestnut St, Boston, MA 02215	4	99.99	Boston (MA)
3	176560	Google Phone	1	600.00	04/12/19 14:38	669 Spruce St, Los Angeles, CA 90001	4	600.00	Los Angeles (CA)

```
      Order Date                        Purchase Address  Month   Sales  \
0  04/19/19 08:46          917 1st St, Dallas, TX 75001      4   23.90
2  04/07/19 22:30     682 Chestnut St, Boston, MA 02215      4   99.99
3  04/12/19 14:38  669 Spruce St, Los Angeles, CA 90001      4  600.00

                City
0        Dallas (TX)
2        Boston (MA)
3   Los Angeles (CA)
```

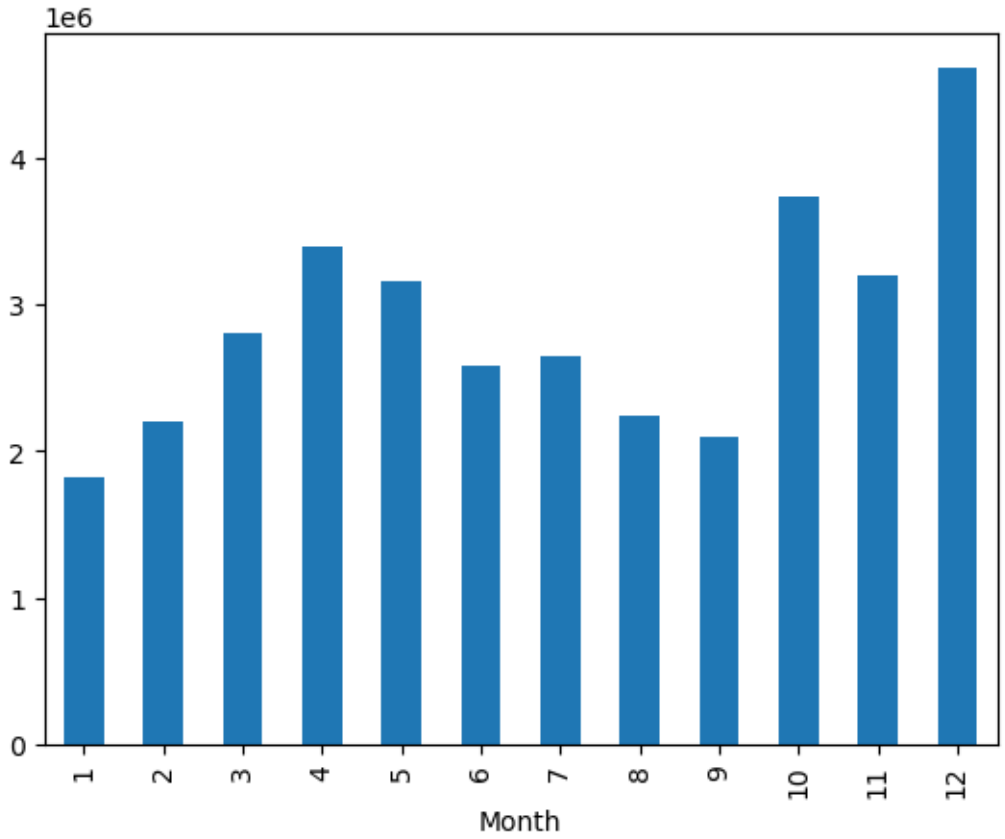
Question 1: Best month for sales? how much was earnes that month?

[16]:
```
df.groupby("Month").sum()["Sales"]
```

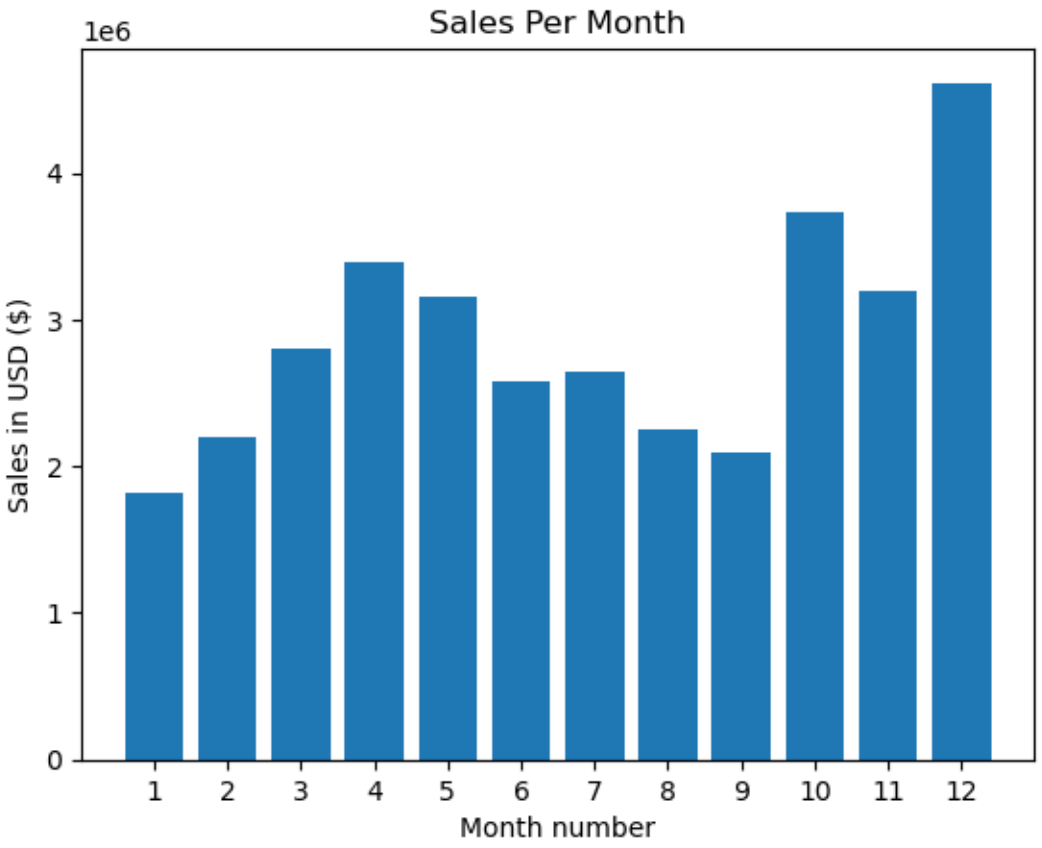
[16]:
```
Month
1     1822256.73
2     2202022.42
3     2807100.38
4     3390670.24
5     3152606.75
6     2577802.26
7     2647775.76
8     2244467.88
9     2097560.13
10    3736726.88
11    3199603.20
12    4613443.34
Name: Sales, dtype: float64
```

[17]:
```
df.groupby("Month").sum()["Sales"].plot(kind="bar")
```

[17]:
```
<Axes: xlabel='Month'>
```



[18]:
```
result = df.groupby("Month").sum()["Sales"]
months = range(1,13)
plt.bar(months,result)
plt.xticks(months)
plt.ylabel("Sales in USD ($)")
plt.xlabel("Month number")
plt.title("Sales Per Month")
plt.show()
```



Question 2: What city had the highest sales

[19]:
```
df.head(3)
```

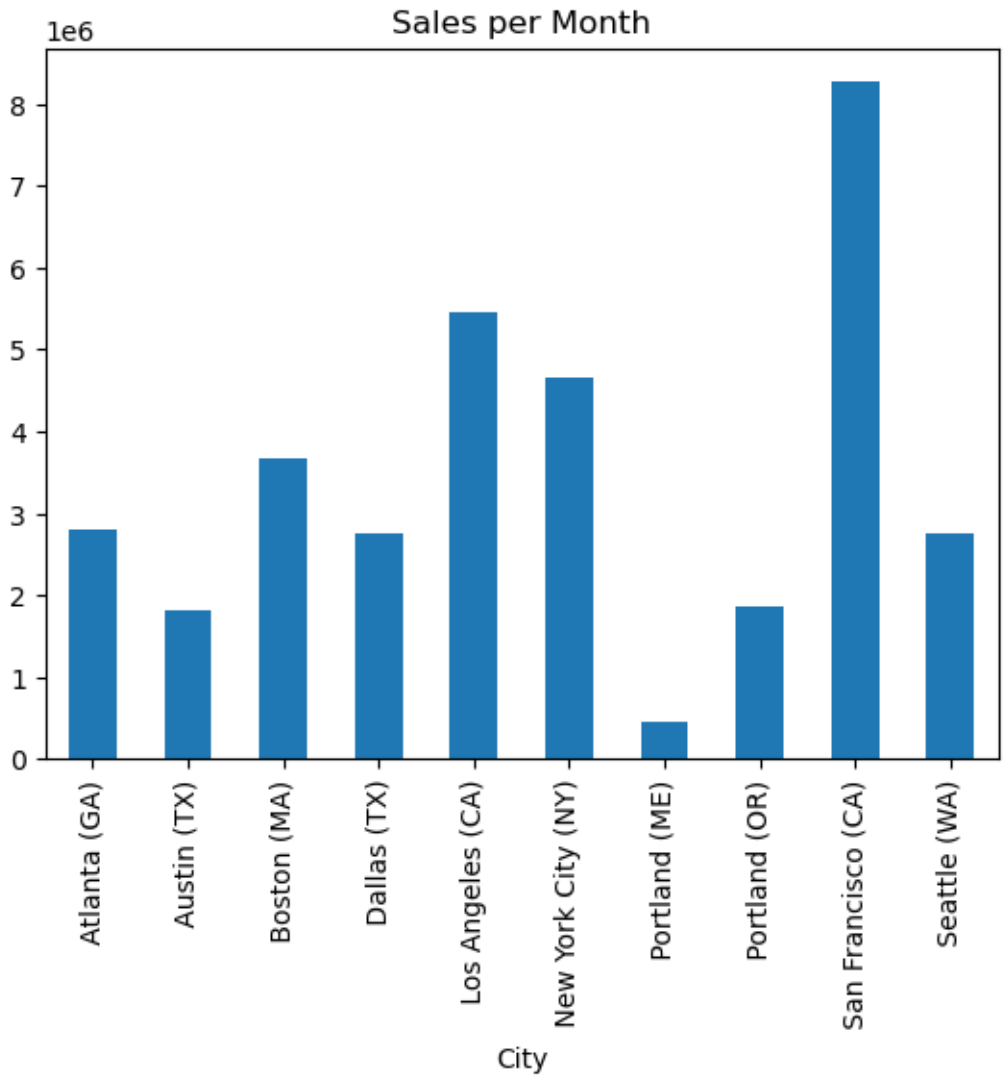
[19]:
```
   Order ID                     Product  Quantity Ordered  Price Each  \
0    176558        USB-C Charging Cable                 2       11.95
2    176559  Bose SoundSport Headphones                 1       99.99
3    176560                Google Phone                 1      600.00

        Order Date                      Purchase Address  Month   Sales  \
0  04/19/19 08:46          917 1st St, Dallas, TX 75001      4   23.90
2  04/07/19 22:30     682 Chestnut St, Boston, MA 02215      4   99.99
3  04/12/19 14:38  669 Spruce St, Los Angeles, CA 90001      4  600.00

               City
0        Dallas (TX)
2        Boston (MA)
3   Los Angeles (CA)
```

[20]:
```
results = df.groupby("City").sum()["Sales"]
results
```

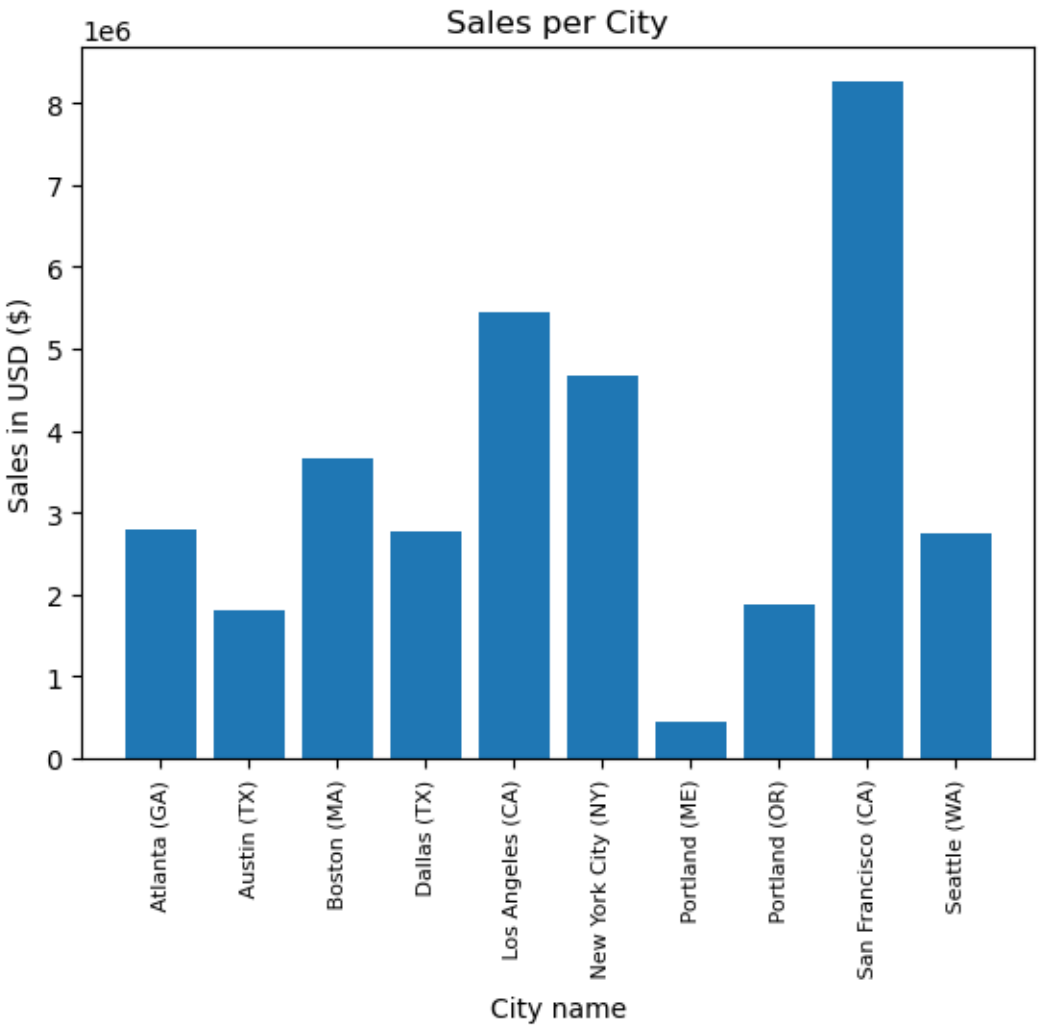
[20]:
```
City
 Atlanta (GA)          2795498.58
 Austin (TX)           1819581.75
 Boston (MA)           3661642.01
 Dallas (TX)           2767975.40
 Los Angeles (CA)      5452570.80
 New York City (NY)    4664317.43
 Portland (ME)          449758.27
 Portland (OR)         1870732.34
 San Francisco (CA)    8262203.91
 Seattle (WA)          2747755.48
Name: Sales, dtype: float64
```

[21]:
```
results.plot(kind='bar',title = "Sales per Month")
```

[21]:
```
<Axes: title={'center': 'Sales per Month'}, xlabel='City'>
```



[23]:
```
cities = [city for city, df in df.groupby('City')]

plt.bar(cities,results)
plt.xticks(cities,rotation = 'vertical',size=8)
plt.ylabel("Sales in USD ($)")
plt.xlabel("City name")
plt.title("Sales per City")
plt.show()
```



Question 3: What time should we display advertisements to maximize likelihood of customer's buying product?

[27]:
```
df["Order Date"] = pd.to_datetime(df["Order Date"])
df.head(3)
```

```
C:\Users\Ahmed\AppData\Local\Temp\ipykernel_11240\1917125877.py:1: UserWarning:
Could not infer format, so each element will be parsed individually, falling
back to `dateutil`. To ensure parsing is consistent and as-expected, please
specify a format.
  df["Order Date"] = pd.to_datetime(df["Order Date"])
```

[27]:

	Order ID	Product	Quantity Ordered	Price Each	\
0	176558	USB-C Charging Cable	2	11.95	
2	176559	Bose SoundSport Headphones	1	99.99	

```
3   176560                Google Phone                 1      600.00

           Order Date                       Purchase Address  Month   Sales  \
0 2019-04-19 08:46:00          917 1st St, Dallas, TX 75001      4   23.90
2 2019-04-07 22:30:00     682 Chestnut St, Boston, MA 02215      4   99.99
3 2019-04-12 14:38:00  669 Spruce St, Los Angeles, CA 90001      4  600.00

                City
0        Dallas (TX)
2        Boston (MA)
3   Los Angeles (CA)
```

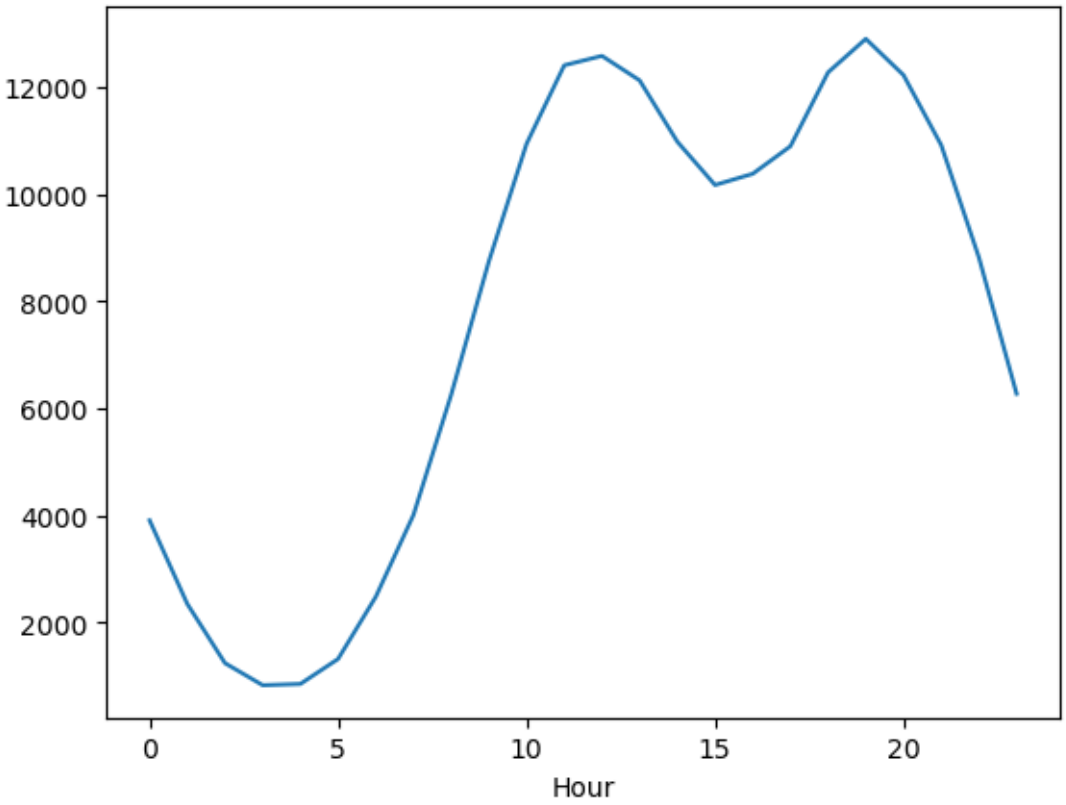
[28]:
```
df['Hour'] = df['Order Date'].dt.hour
df["Minute"] = df["Order Date"].dt.minute

df.head()
```

[28]:
```
  Order ID                    Product  Quantity Ordered  Price Each  \
0   176558        USB-C Charging Cable                 2       11.95
2   176559  Bose SoundSport Headphones                 1       99.99
3   176560                Google Phone                 1      600.00
4   176560            Wired Headphones                 1       11.99
5   176561            Wired Headphones                 1       11.99

           Order Date                       Purchase Address  Month   Sales  \
0 2019-04-19 08:46:00          917 1st St, Dallas, TX 75001      4   23.90
2 2019-04-07 22:30:00     682 Chestnut St, Boston, MA 02215      4   99.99
3 2019-04-12 14:38:00  669 Spruce St, Los Angeles, CA 90001      4  600.00
4 2019-04-12 14:38:00  669 Spruce St, Los Angeles, CA 90001      4   11.99
5 2019-04-30 09:27:00     333 8th St, Los Angeles, CA 90001      4   11.99

                City  Hour  Minute
0        Dallas (TX)     8      46
2        Boston (MA)    22      30
3   Los Angeles (CA)    14      38
4   Los Angeles (CA)    14      38
5   Los Angeles (CA)     9      27
```

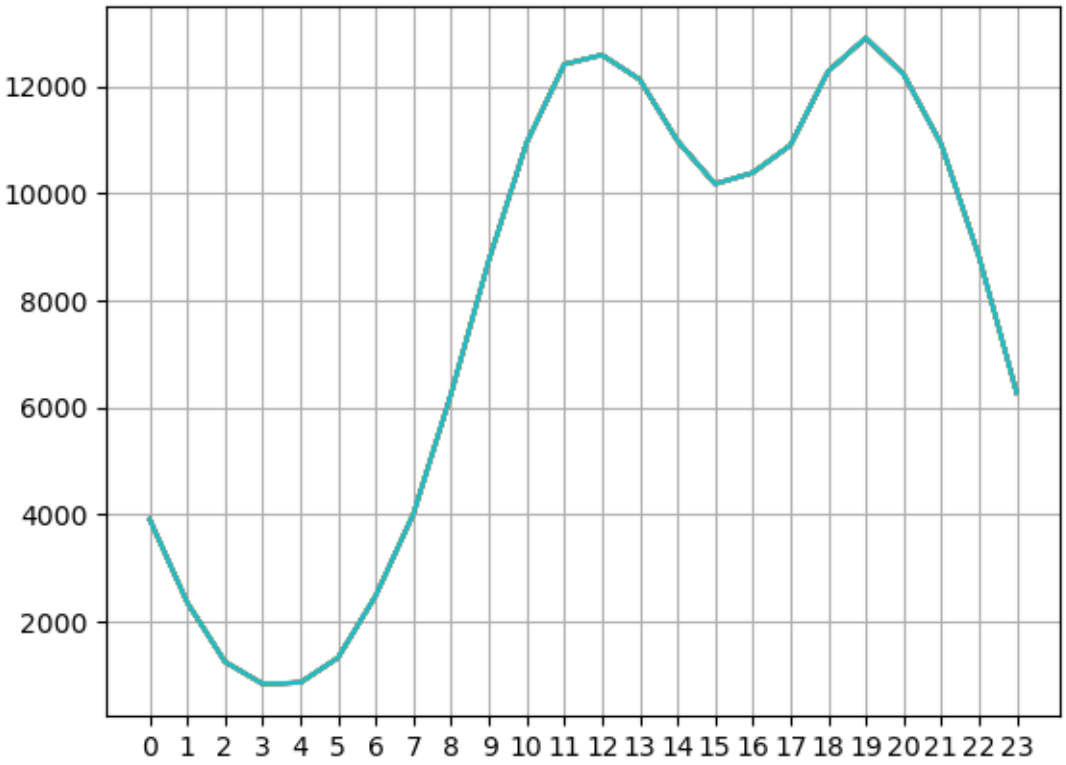
[40]:
```
df["Hour"] = pd.to_numeric(df["Hour"])
```

[42]:
```
df.groupby(["Hour"]).count()["Sales"].plot()
```

[42]:
```
<Axes: xlabel='Hour'>
```



[43]:
```
hours = [hour for hour, data in df.groupby("Hour")]
plt.plot(hours,df.groupby(["Hour"]).count())
plt.xticks(hours)
plt.grid()
plt.show()

# 11 am or 7pm
```



Question 4: What products are most often sold together

[45]:
```
new = df[df["Order ID"].duplicated(keep=False)]
new.head()
```

[45]:

	Order ID	Product	Quantity Ordered	Price Each	\
3	176560	Google Phone	1	600.00	
4	176560	Wired Headphones	1	11.99	
18	176574	Google Phone	1	600.00	
19	176574	USB-C Charging Cable	1	11.95	
30	176585	Bose SoundSport Headphones	1	99.99	

	Order Date	Purchase Address	Month	Sales	\
3	2019-04-12 14:38:00	669 Spruce St, Los Angeles, CA 90001	4	600.00	
4	2019-04-12 14:38:00	669 Spruce St, Los Angeles, CA 90001	4	11.99	
18	2019-04-03 19:42:00	20 Hill St, Los Angeles, CA 90001	4	600.00	
19	2019-04-03 19:42:00	20 Hill St, Los Angeles, CA 90001	4	11.95	
30	2019-04-07 11:31:00	823 Highland St, Boston, MA 02215	4	99.99	

	City	Hour	Minute
3	Los Angeles (CA)	14	38
4	Los Angeles (CA)	14	38

```
18   Los Angeles (CA)    19      42
19   Los Angeles (CA)    19      42
30        Boston (MA)    11      31
```

[49]:
```
new["Grouped"] = new.groupby("Order ID")["Product"].transform(lambda x: ",".
 ↪join(x))

new = new[["Order ID","Grouped"]].drop_duplicates()
new.head()
```

```
C:\Users\Ahmed\AppData\Local\Temp\ipykernel_11240\2338753758.py:1:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  new["Grouped"] = new.groupby("Order ID")["Product"].transform(lambda x:
",".join(x))
```

[49]:

	Order ID	Grouped
3	176560	Google Phone,Wired Headphones
18	176574	Google Phone,USB-C Charging Cable
30	176585	Bose SoundSport Headphones,Bose SoundSport Hea…
32	176586	AAA Batteries (4-pack),Google Phone
119	176672	Lightning Charging Cable,USB-C Charging Cable

[50]:
```
from itertools import combinations
from collections import Counter
```

[52]:
```
count = Counter()

for row in new["Grouped"]:
    row_list = row.split(",")
    count.update(Counter(combinations(row_list,2)))

for key,value in count.most_common(10):
    print(key,value)
```

```
('iPhone', 'Lightning Charging Cable') 1005
('Google Phone', 'USB-C Charging Cable') 987
('iPhone', 'Wired Headphones') 447
('Google Phone', 'Wired Headphones') 414
('Vareebadd Phone', 'USB-C Charging Cable') 361
('iPhone', 'Apple Airpods Headphones') 360
('Google Phone', 'Bose SoundSport Headphones') 220
('USB-C Charging Cable', 'Wired Headphones') 160
```

```
('Vareebadd Phone', 'Wired Headphones') 143
('Lightning Charging Cable', 'Wired Headphones') 92
```

Question 5: What product sold the most? why

[54]:
```
df.dtypes
```

[54]:
```
Order ID                    object
Product                     object
Quantity Ordered             int64
Price Each                 float64
Order Date          datetime64[ns]
Purchase Address            object
Month                        int32
Sales                      float64
City                        object
Hour                         int32
Minute                       int32
dtype: object
```

[63]:
```
product_group = df.groupby("Product")
product_group.head(2)
```

[63]:

	Order ID	Product	Quantity Ordered	Price Each	\
0	176558	USB-C Charging Cable	2	11.95	
2	176559	Bose SoundSport Headphones	1	99.99	
3	176560	Google Phone	1	600.00	
4	176560	Wired Headphones	1	11.99	
5	176561	Wired Headphones	1	11.99	
6	176562	USB-C Charging Cable	1	11.95	
7	176563	Bose SoundSport Headphones	1	99.99	
9	176565	Macbook Pro Laptop	1	1700.00	
11	176567	Google Phone	1	600.00	
12	176568	Lightning Charging Cable	1	14.95	
13	176569	27in 4K Gaming Monitor	1	389.99	
14	176570	AA Batteries (4-pack)	1	3.84	
15	176571	Lightning Charging Cable	1	14.95	
16	176572	Apple Airpods Headphones	1	150.00	
20	176575	AAA Batteries (4-pack)	1	2.99	
21	176576	Apple Airpods Headphones	1	150.00	
24	176579	AA Batteries (4-pack)	1	3.84	
26	176581	iPhone	1	700.00	
28	176583	AAA Batteries (4-pack)	2	2.99	
29	176584	Flatscreen TV	1	300.00	
34	176587	27in FHD Monitor	1	149.99	
35	176588	20in Monitor	1	109.99	
47	176600	27in 4K Gaming Monitor	1	389.99	
53	176606	LG Dryer	1	600.00	

55	176608	iPhone	1	700.00
66	176619	Flatscreen TV	1	300.00
70	176623	27in FHD Monitor	1	149.99
80	176633	ThinkPad Laptop	1	999.99
82	176635	Vareebadd Phone	1	400.00
86	176639	Macbook Pro Laptop	1	1700.00
99	176652	LG Washing Machine	1	600.00
104	176657	ThinkPad Laptop	1	999.99
106	176659	20in Monitor	1	109.99
109	176662	34in Ultrawide Monitor	1	379.99
124	176676	LG Dryer	1	600.00
125	176677	34in Ultrawide Monitor	1	379.99
256	176801	Vareebadd Phone	1	400.00
283	176825	LG Washing Machine	1	600.00

	Order Date	Purchase Address	Month	\
0	2019-04-19 08:46:00	917 1st St, Dallas, TX 75001	4	
2	2019-04-07 22:30:00	682 Chestnut St, Boston, MA 02215	4	
3	2019-04-12 14:38:00	669 Spruce St, Los Angeles, CA 90001	4	
4	2019-04-12 14:38:00	669 Spruce St, Los Angeles, CA 90001	4	
5	2019-04-30 09:27:00	333 8th St, Los Angeles, CA 90001	4	
6	2019-04-29 13:03:00	381 Wilson St, San Francisco, CA 94016	4	
7	2019-04-02 07:46:00	668 Center St, Seattle, WA 98101	4	
9	2019-04-24 10:38:00	915 Willow St, San Francisco, CA 94016	4	
11	2019-04-18 17:18:00	444 7th St, Los Angeles, CA 90001	4	
12	2019-04-15 12:18:00	438 Elm St, Seattle, WA 98101	4	
13	2019-04-16 19:23:00	657 Hill St, Dallas, TX 75001	4	
14	2019-04-22 15:09:00	186 12th St, Dallas, TX 75001	4	
15	2019-04-19 14:29:00	253 Johnson St, Atlanta, GA 30301	4	
16	2019-04-04 20:30:00	149 Dogwood St, New York City, NY 10001	4	
20	2019-04-27 00:30:00	433 Hill St, New York City, NY 10001	4	
21	2019-04-28 11:42:00	771 Ridge St, Los Angeles, CA 90001	4	
24	2019-04-11 10:23:00	886 Jefferson St, New York City, NY 10001	4	
26	2019-04-09 21:38:00	84 Jackson St, Boston, MA 02215	4	
28	2019-04-20 12:00:00	146 Jackson St, Portland, OR 97035	4	
29	2019-04-24 20:39:00	936 Church St, San Francisco, CA 94016	4	
34	2019-04-29 19:38:00	557 5th St, Los Angeles, CA 90001	4	
35	2019-04-02 04:00:00	765 Cherry St, Seattle, WA 98101	4	
47	2019-04-30 15:54:00	87 West St, Boston, MA 02215	4	
53	2019-04-21 14:16:00	487 Maple St, San Francisco, CA 94016	4	
55	2019-04-11 12:01:00	15 Cherry St, San Francisco, CA 94016	4	
66	2019-04-16 18:37:00	116 North St, Los Angeles, CA 90001	4	
70	2019-04-20 23:51:00	807 12th St, Atlanta, GA 30301	4	
80	2019-04-23 14:03:00	863 Hickory St, Los Angeles, CA 90001	4	
82	2019-04-26 09:55:00	85 North St, San Francisco, CA 94016	4	
86	2019-04-28 16:14:00	853 Cedar St, San Francisco, CA 94016	4	
99	2019-04-09 20:04:00	502 14th St, New York City, NY 10001	4	

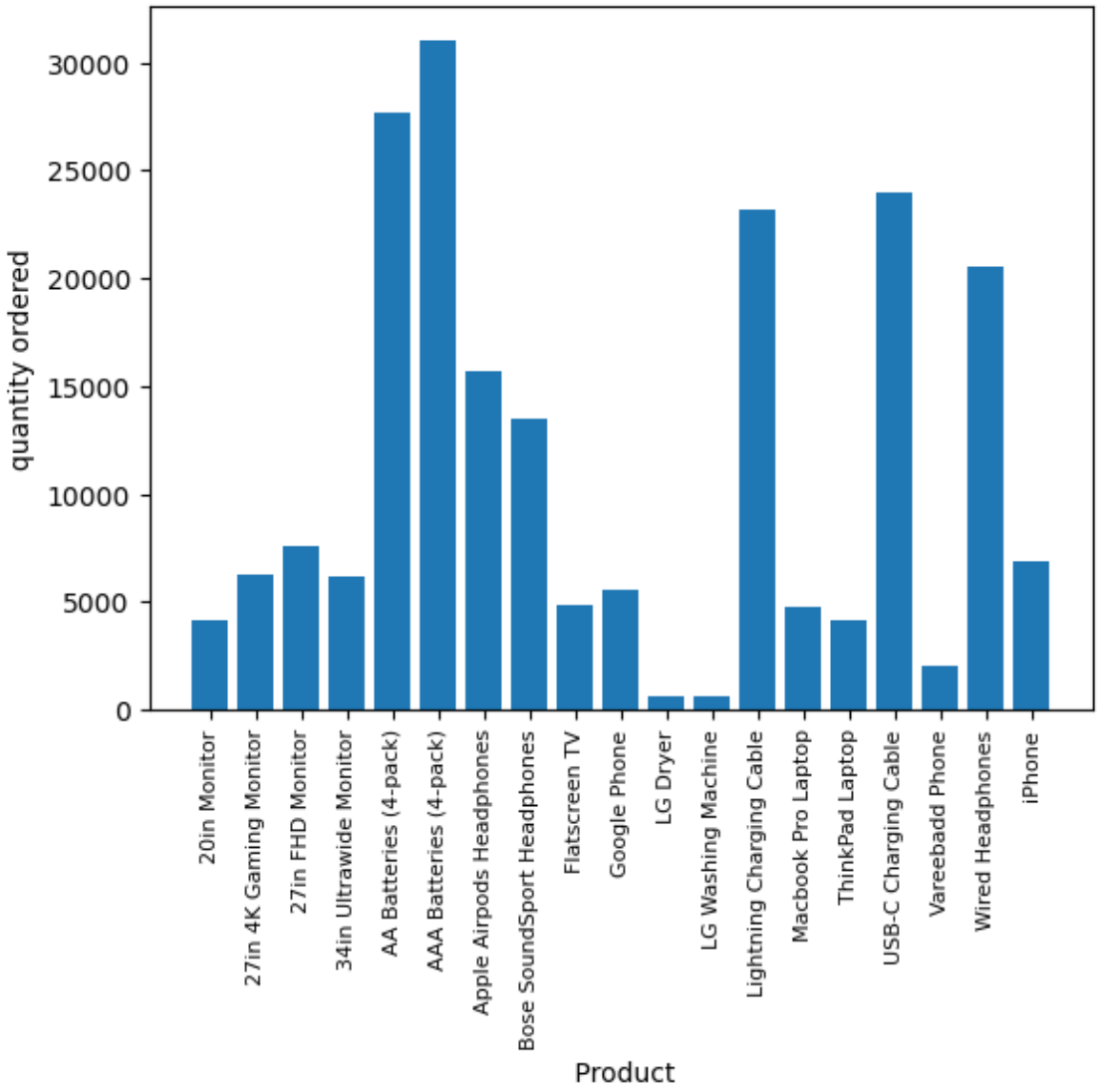
```
104 2019-04-23 10:16:00       774 Forest St, Los Angeles, CA 90001      4
106 2019-04-29 14:40:00            804 Church St, Dallas, TX 75001      4
109 2019-04-05 12:46:00          678 Hickory St, Portland, OR 97035     4
124 2019-04-09 00:35:00      788 Lincoln St, Los Angeles, CA 90001      4
125 2019-04-01 11:50:00        661 Washington St, Austin, TX 73301      4
256 2019-04-28 18:52:00      125 North St, San Francisco, CA 94016      4
283 2019-04-13 22:31:00        338 6th St, San Francisco, CA 94016      4
```

	Sales	City	Hour	Minute
0	23.90	Dallas (TX)	8	46
2	99.99	Boston (MA)	22	30
3	600.00	Los Angeles (CA)	14	38
4	11.99	Los Angeles (CA)	14	38
5	11.99	Los Angeles (CA)	9	27
6	11.95	San Francisco (CA)	13	3
7	99.99	Seattle (WA)	7	46
9	1700.00	San Francisco (CA)	10	38
11	600.00	Los Angeles (CA)	17	18
12	14.95	Seattle (WA)	12	18
13	389.99	Dallas (TX)	19	23
14	3.84	Dallas (TX)	15	9
15	14.95	Atlanta (GA)	14	29
16	150.00	New York City (NY)	20	30
20	2.99	New York City (NY)	0	30
21	150.00	Los Angeles (CA)	11	42
24	3.84	New York City (NY)	10	23
26	700.00	Boston (MA)	21	38
28	5.98	Portland (OR)	12	0
29	300.00	San Francisco (CA)	20	39
34	149.99	Los Angeles (CA)	19	38
35	109.99	Seattle (WA)	4	0
47	389.99	Boston (MA)	15	54
53	600.00	San Francisco (CA)	14	16
55	700.00	San Francisco (CA)	12	1
66	300.00	Los Angeles (CA)	18	37
70	149.99	Atlanta (GA)	23	51
80	999.99	Los Angeles (CA)	14	3
82	400.00	San Francisco (CA)	9	55
86	1700.00	San Francisco (CA)	16	14
99	600.00	New York City (NY)	20	4
104	999.99	Los Angeles (CA)	10	16
106	109.99	Dallas (TX)	14	40
109	379.99	Portland (OR)	12	46
124	600.00	Los Angeles (CA)	0	35
125	379.99	Austin (TX)	11	50
256	400.00	San Francisco (CA)	18	52
283	600.00	San Francisco (CA)	22	31

[64]:
```
quantity_ordered = product_group["Quantity Ordered"].sum()
```

[65]:
```
products = [product for product, df in product_group]
```

[68]:
```
plt.bar(products,quantity_ordered)
plt.ylabel("quantity ordered")
plt.xlabel(" Product")
plt.xticks(products,rotation = "vertical",size=8)
plt.show()
```

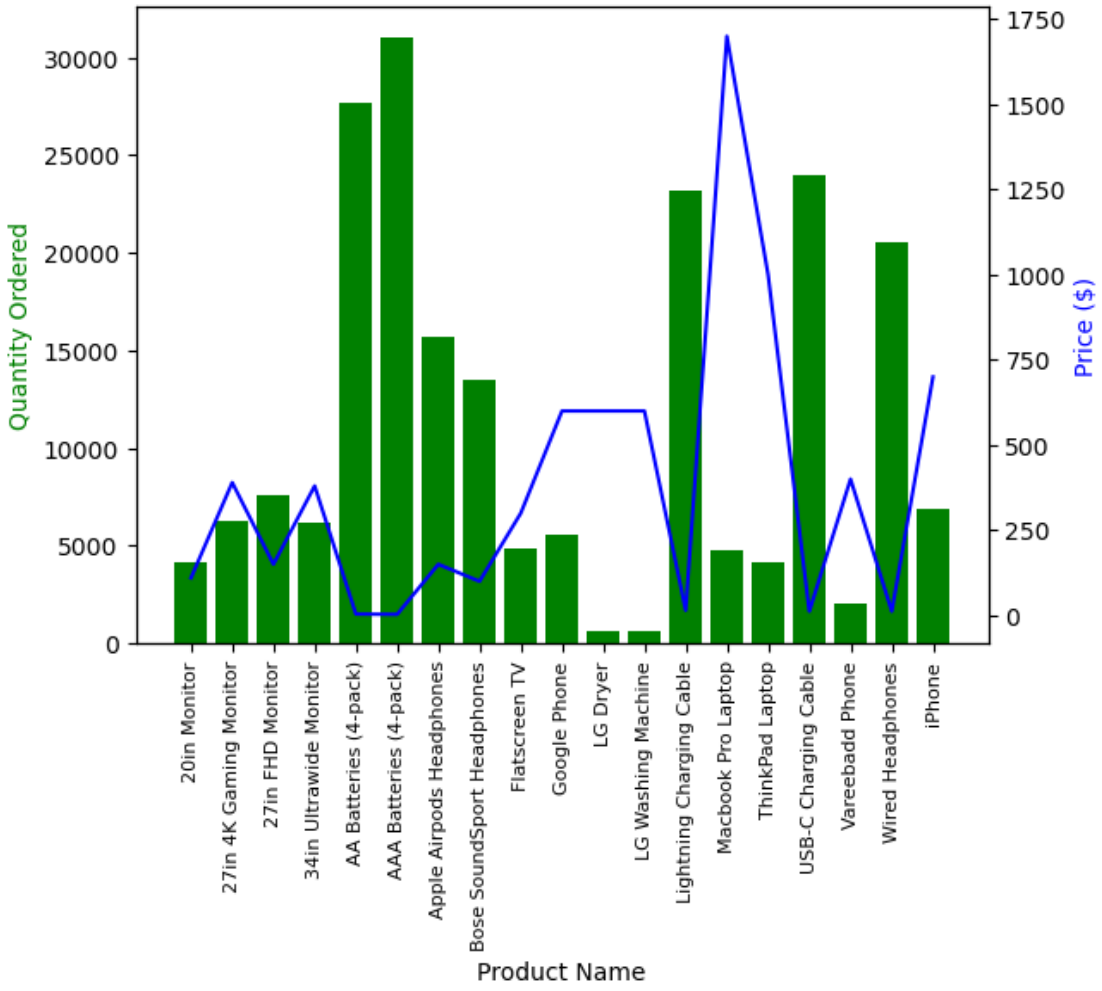


[70]:
```
prices = df.groupby("Product")["Price Each"].mean()
print(prices)
```

```
Product
20in Monitor                   109.99
27in 4K Gaming Monitor         389.99
27in FHD Monitor               149.99
34in Ultrawide Monitor         379.99
AA Batteries (4-pack)            3.84
AAA Batteries (4-pack)           2.99
Apple Airpods Headphones       150.00
Bose SoundSport Headphones      99.99
Flatscreen TV                  300.00
Google Phone                   600.00
LG Dryer                       600.00
LG Washing Machine             600.00
Lightning Charging Cable        14.95
Macbook Pro Laptop            1700.00
ThinkPad Laptop                999.99
USB-C Charging Cable            11.95
Vareebadd Phone                400.00
Wired Headphones                11.99
iPhone                         700.00
Name: Price Each, dtype: float64
```

[76]:
```
fig, ax1 = plt.subplots()
ax2 = ax1.twinx()
ax1.bar(products, quantity_ordered, color='g')
ax2.plot(products, prices, 'b-')

ax1.set_xlabel("Product Name")
ax1.set_ylabel("Quantity Ordered", color = 'g')
ax2.set_ylabel("Price ($)", color='b')
ax1.set_xticklabels(products,rotation="vertical",size=8)
plt.show()
```

```
C:\Users\Ahmed\AppData\Local\Temp\ipykernel_11240\2633492285.py:9: UserWarning:
FixedFormatter should only be used together with FixedLocator
  ax1.set_xticklabels(products,rotation="vertical",size=8)
```


Quantity Ordered
Price ($)
Product Name
0
5000
10000
15000
20000
25000
30000
0
250
500
750
1000
1250
1500
1750
20in Monitor
27in 4K Gaming Monitor
27in FHD Monitor
34in Ultrawide Monitor
AA Batteries (4-pack)
AAA Batteries (4-pack)
Apple Airpods Headphones
Bose SoundSport Headphones
Flatscreen TV
Google Phone
LG Dryer
LG Washing Machine
Lightning Charging Cable
Macbook Pro Laptop
ThinkPad Laptop
USB-C Charging Cable
Vareebadd Phone
Wired Headphones
iPhone

[ ]: