Sales_data

December 15, 2023

[67]:

```python
import pandas as pd
import warnings
warnings.filterwarnings("ignore")
```

0.0.1 Reading data set!

[68]:

```python
df = pd.read_csv('Sales_April_2019_updated.csv')
df.head()
```

[68]:

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

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

0.0.2 Checking Null values

[69]:

```python
df.isnull().sum()
```

[69]:

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

0.0.3 Data set size of rows and columns

```
[70]: df.shape
```

```
[70]: (18289, 6)
```

0.0.4 Data set Columns

```
[71]: df.columns
```

```
[71]: Index(['Order ID', 'Product', 'Quantity Ordered', 'Price Each', 'Order Date',
             'Purchase Address'],
            dtype='object')
```

0.0.5 Columns and it's Data types

```
[72]: df.dtypes
```

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

0.0.6 Statastical Data Description

```
[73]: df.describe()
```

```
[73]:             Order ID  Quantity Ordered    Price Each
      count   18289.000000       18289.00000  18289.000000
      mean   185328.816720           1.12461    184.431026
      std      5061.520829           0.43641    330.913377
      min    176558.000000           1.00000      2.990000
      25%    180952.000000           1.00000     11.950000
      50%    185328.000000           1.00000     14.950000
      75%    189706.000000           1.00000    150.000000
      max    194094.000000           7.00000   1700.000000
```

0.0.7 Adding new column Month Based on Order date column Extract Month only

```
[74]: df['Month'] = df['Order Date'].str[0:2]
      df.head()
```

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

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

0.0.8 Creating two methods which takes address as parameter and give City,state from 'Purchase Address' column

```
[75]: def get_state(address):
          return address.split(",")[2].strip(" ")

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

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

      df.head()
```

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

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

                          City
      0       Dallas(TX 75001)
      1       Boston(MA 02215)
      2  Los Angeles(CA 90001)
      3  Los Angeles(CA 90001)
```

```
4  Los Angeles(CA 90001)
```

0.0.9 Checking Data types

[76]:
```
df.dtypes
```

[76]:
```
Order ID              int64
Product              object
Quantity Ordered      int64
Price Each          float64
Order Date           object
Purchase Address     object
Month                object
City                 object
dtype: object
```

0.1 Data Exploration!

0.1.1 Question 1: What was the best month for sales? How much was earned that month?

0.1.2 Adding new column which is Sales as calculated with 'Quantity Ordered' multipling with 'Price Each'

[77]:
```
df['Sales'] = df['Quantity Ordered']*df['Price Each']
df.head()
```

[77]:
```
   Order ID                     Product  Quantity Ordered  Price Each  \
0    176558        USB-C Charging Cable                 2       11.95
1    176559  Bose SoundSport Headphones                 1       99.99
2    176560                Google Phone                 1      600.00
3    176560            Wired Headphones                 1       11.99
4    176561            Wired Headphones                 1       11.99

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

                    City   Sales
0        Dallas(TX 75001)   23.90
1        Boston(MA 02215)   99.99
2  Los Angeles(CA 90001)  600.00
3  Los Angeles(CA 90001)   11.99
4  Los Angeles(CA 90001)   11.99
```

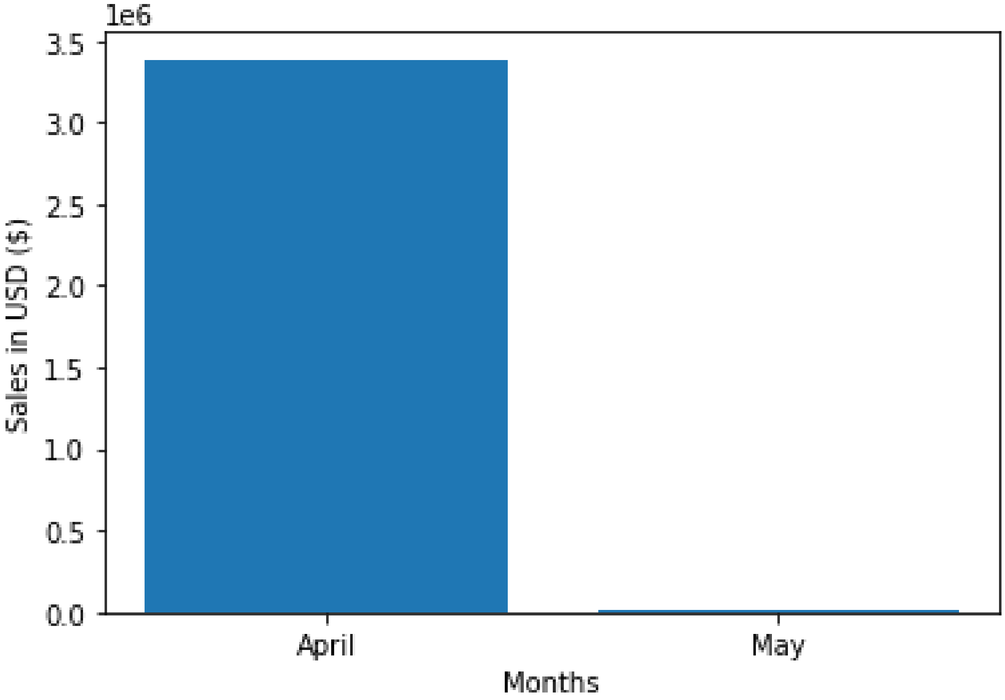
0.1.3 Monthly vise Sales

[78]:
```
df.groupby(['Month']).sum()
```

[78]:
```
          Order ID  Quantity Ordered  Price Each       Sales
Month
04      3384310980             20539  3362503.59  3385499.82
05         5167749                29    10555.45    10559.29
```

0.1.4 Bar chart for monthly vise total sales

[79]:
```
import matplotlib.pyplot as plt

months=['April','May']
#print(months)

plt.bar(months,df.groupby(['Month']).sum()['Sales'])
plt.xlabel('Months')
plt.ylabel('Sales in USD ($)')
plt.show()
```



0.1.5 Question2: City Vise Sales

[80]:

```
df.groupby(['City']).sum()
```

[80]:

```
                          Order ID  Quantity Ordered  Price Each      Sales
City
Atlanta(GA 30301)        273087674              1633   282879.88  284454.92
Austin(TX 73301)         180656573              1092   171487.65  172683.59
Boston(MA 02215)         355468629              2190   351742.75  353880.16
Dallas(TX 75001)         250139729              1519   251689.04  252840.47
Los Angeles(CA 90001)    560637494              3399   547991.02  551399.07
New York City(NY 10001)  450696802              2741   442392.68  446587.78
Portland(ME 04101)        45098002               265    42370.29   42536.49
Portland(OR 97035)       184454339              1134   197722.63  198591.62
San Francisco(CA 94016)  822325761              4987   810338.31  817074.77
Seattle(WA 98101)        266913726              1608   274444.79  276010.24
```

[81]:

```
keys = [city for city, df1 in df.groupby(['City'])]
keys
```

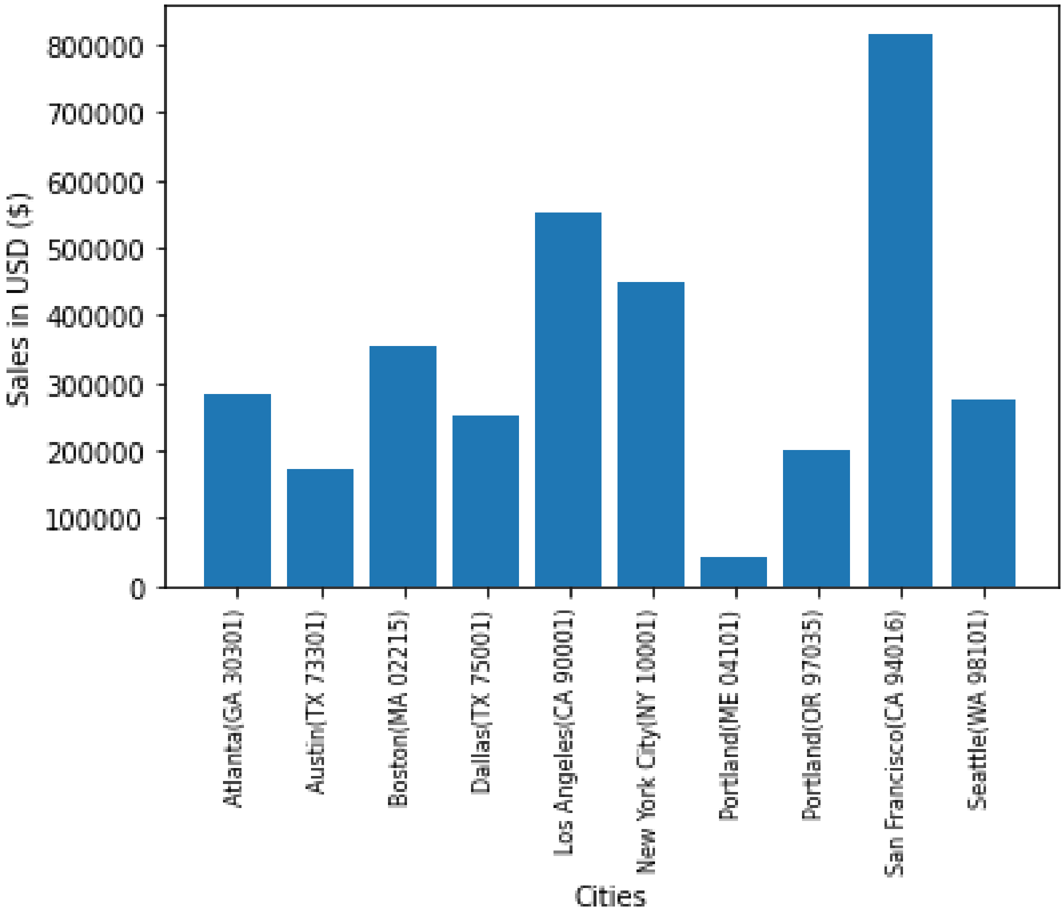
[81]:

```
['Atlanta(GA 30301)',
 'Austin(TX 73301)',
 'Boston(MA 02215)',
 'Dallas(TX 75001)',
 'Los Angeles(CA 90001)',
 'New York City(NY 10001)',
 'Portland(ME 04101)',
 'Portland(OR 97035)',
 'San Francisco(CA 94016)',
 'Seattle(WA 98101)']
```

0.1.6 Bar chart for city vise sales

[82]:

```
plt.bar(keys,df.groupby(['City']).sum()['Sales'])
plt.xlabel('Cities')
plt.ylabel('Sales in USD ($)')
plt.xticks(keys, rotation='vertical', size=8)#plt.
 ↪xticks(keys,rotation='vertical' size=8)
plt.show()
```



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

[83]:

```
df['Hour']=pd.to_datetime(df['Order Date']).dt.hour
df['Minute']=pd.to_datetime(df['Order Date']).dt.minute
df['Count']=1
df.head()
```

[83]:

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

        Order Date                      Purchase Address Month  \
0   04/19/19 08:46          917 1st St, Dallas, TX 75001    04
1 04-07-2019 22:30     682 Chestnut St, Boston, MA 02215    04
```

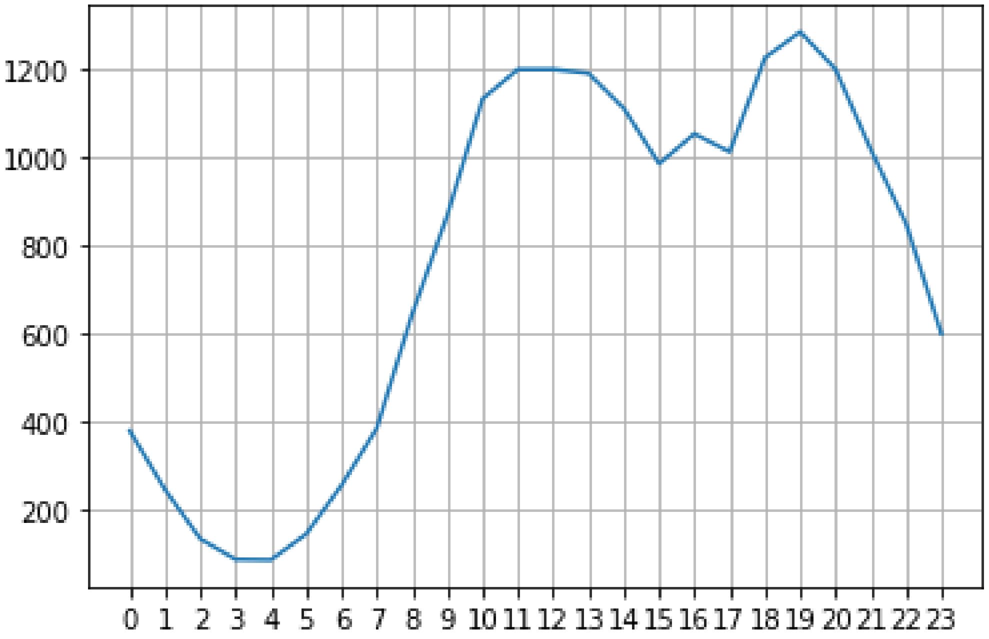
```
2  04-12-2019 14:38  669 Spruce St, Los Angeles, CA 90001    04
3  04-12-2019 14:38  669 Spruce St, Los Angeles, CA 90001    04
4    04/30/19 09:27      333 8th St, Los Angeles, CA 90001    04

                    City   Sales  Hour  Minute  Count
0        Dallas(TX 75001)   23.90     8      46      1
1        Boston(MA 02215)   99.99    22      30      1
2  Los Angeles(CA 90001)  600.00    14      38      1
3  Los Angeles(CA 90001)   11.99    14      38      1
4  Los Angeles(CA 90001)   11.99     9      27      1
```

```
[84]: keys = [pair for pair, df1 in df.groupby(['Hour'])]
      keys
```

```
[84]: [0,
       1,
       2,
       3,
       4,
       5,
       6,
       7,
       8,
       9,
       10,
       11,
       12,
       13,
       14,
       15,
       16,
       17,
       18,
       19,
       20,
       21,
       22,
       23]
```

0.1.8 Hour vise sale data

```
[85]: plt.plot(keys,df.groupby(['Hour']).count()['Count'])
      plt.xticks(keys)
      plt.grid()
      plt.show()
```



0.1.9 My recommendation is slightly before 11am or 7pm

```
[86]: df=df[df['Order ID'].duplicated(keep=False)]
      df
```

```
[86]:        Order ID                    Product  Quantity Ordered  Price Each  \
      2        176560               Google Phone                 1      600.00
      3        176560          Wired Headphones                 1       11.99
      17       176574               Google Phone                 1      600.00
      18       176574       USB-C Charging Cable                 1       11.95
      29       176585   Bose SoundSport Headphones                1       99.99
      …          …                          …                  …          …
      18242    194050       USB-C Charging Cable                 1       11.95
      18248    194056                     iPhone                 1      700.00
      18249    194056   Lightning Charging Cable                 1       14.95
      18254    194061                     iPhone                 1      700.00
      18255    194061   Lightning Charging Cable                 3       14.95

                  Order Date                      Purchase Address Month  \
      2      04-12-2019 14:38  669 Spruce St, Los Angeles, CA 90001    04
      3      04-12-2019 14:38  669 Spruce St, Los Angeles, CA 90001    04
      17     04-03-2019 19:42     20 Hill St, Los Angeles, CA 90001    04
      18     04-03-2019 19:42     20 Hill St, Los Angeles, CA 90001    04
      29     04-07-2019 11:31     823 Highland St, Boston, MA 02215    04
      …                  …                                  …   …
```

```
18242    04/27/19 00:27   997 9th St, San Francisco, CA 94016    04
18248  04-10-2019 10:05   280 7th St, San Francisco, CA 94016    04
18249  04-10-2019 10:05   280 7th St, San Francisco, CA 94016    04
18254    04/14/19 20:22         209 6th St, Atlanta, GA 30301    04
18255    04/14/19 20:22         209 6th St, Atlanta, GA 30301    04

                          City   Sales  Hour  Minute  Count
2        Los Angeles(CA 90001)  600.00    14      38      1
3        Los Angeles(CA 90001)   11.99    14      38      1
17       Los Angeles(CA 90001)  600.00    19      42      1
18       Los Angeles(CA 90001)   11.95    19      42      1
29            Boston(MA 02215)   99.99    11      31      1
…                          …     …   …      …    …
18242  San Francisco(CA 94016)   11.95     0      27      1
18248  San Francisco(CA 94016)  700.00    10       5      1
18249  San Francisco(CA 94016)   14.95    10       5      1
18254        Atlanta(GA 30301)  700.00    20      22      1
18255        Atlanta(GA 30301)   44.85    20      22      1

[1469 rows x 12 columns]
```

0.1.10 Question 4: What products are most often sold together?

[87]:
```
df['Grouped'] = df.groupby('Order ID')['Product'].transform(lambda x: ','.
  ↪join(x))
df2= df[['Order ID','Grouped']].drop_duplicates()
```

[89]:
```
df2
```

[89]:
```
       Order ID                                            Grouped
2        176560                       Google Phone,Wired Headphones
17       176574                   Google Phone,USB-C Charging Cable
29       176585  Bose SoundSport Headphones,Bose SoundSport Hea…
31       176586                 AAA Batteries (4-pack),Google Phone
118      176672     Lightning Charging Cable,USB-C Charging Cable
…           …                                                  …
18197    194008             AA Batteries (4-pack),Wired Headphones
18211    194021                       Google Phone,Wired Headphones
18241    194050         AA Batteries (4-pack),USB-C Charging Cable
18248    194056                   iPhone,Lightning Charging Cable
18254    194061                   iPhone,Lightning Charging Cable

[717 rows x 2 columns]
```

[90]:
```
from itertools import combinations
from collections import Counter
```

```
count = Counter()

for row in df2['Grouped']:
    row_list = row.split(',')
    count.update(Counter(combinations(row_list, 2)))

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

```
('Google Phone', 'USB-C Charging Cable') 106
('iPhone', 'Lightning Charging Cable') 106
('iPhone', 'Wired Headphones') 43
('Google Phone', 'Wired Headphones') 41
('iPhone', 'Apple Airpods Headphones') 37
('Vareebadd Phone', 'USB-C Charging Cable') 36
('Google Phone', 'Bose SoundSport Headphones') 24
('Vareebadd Phone', 'Wired Headphones') 15
('USB-C Charging Cable', 'Wired Headphones') 14
('Bose SoundSport Headphones', 'Wired Headphones') 8
```

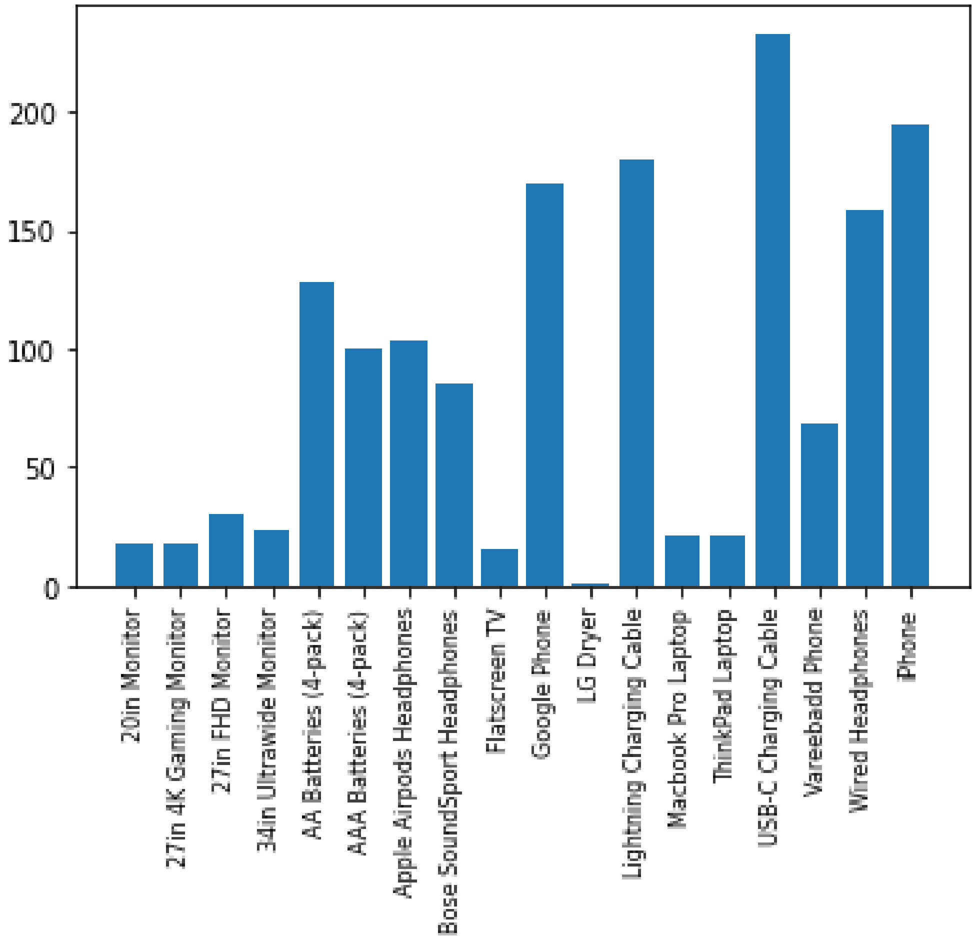
0.1.11 What product sold the most? Why do you think it sold the most?

[91]:

```
product_group = df.groupby('Product')
quantity_ordered = product_group.sum()['Quantity Ordered']
```

[92]:

```
keys = [pair for pair, df in product_group]
plt.bar(keys, quantity_ordered)
plt.xticks(keys, rotation='vertical', size=8)
plt.show()
```

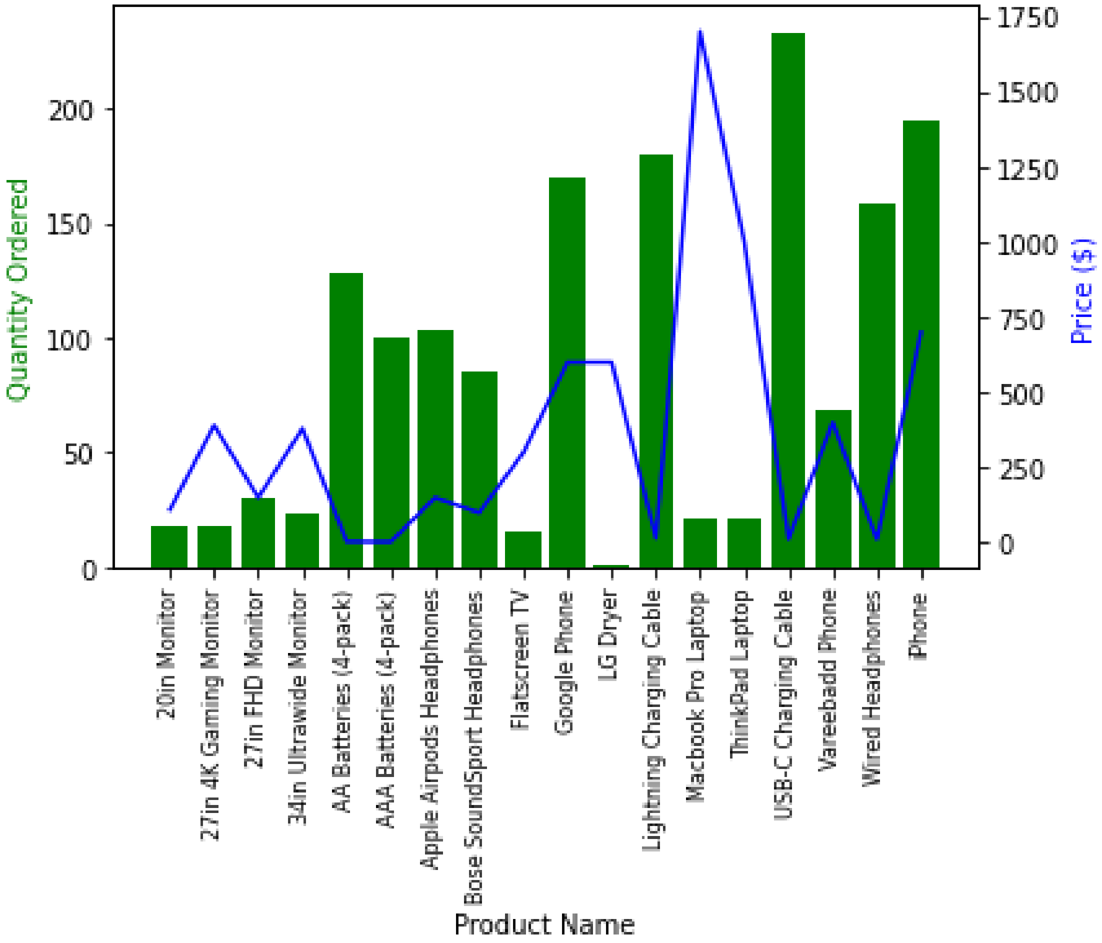


[93]:

```
prices = df.groupby('Product').mean()['Price Each']
```

[94]:

```
fig, ax1 = plt.subplots()

ax2 = ax1.twinx()
ax1.bar(keys, quantity_ordered, color='g')
ax2.plot(keys, prices, color='b')

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

fig.show()
```


Quantity Ordered
0
50
100
150
200
Price ($)
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
Lightning Charging Cable
Macbook Pro Laptop
ThinkPad Laptop
USB-C Charging Cable
Vareebadd Phone
Wired Headphones
iPhone
Product Name