Sales Analysis - EL AIRAJ ANAS

June 7, 2023

1 ELECTRONIC SALES ANALYSIS PROJECT MADE BY EL AIRAJ ANAS

[73]:
```
import pandas as pd
import os
```

[45]:
```
import os

current_directory = os.getcwd()
print("Current Directory:", current_directory)
```

```
Current Directory: C:\Users\ANAS
```

Merging 12 months of sales data into a single file

[74]:
```
df = pd.read_csv ("./Sales_Data/Sales_April_2019.csv")

files = [file for file in os.listdir ('Sales_Data')]

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


all_month_data.to_csv("all_Data.csv", index = False )
```

[75]:
```
all_Data = pd.read_csv("all_Data.csv")
all_Data.head()
```

[75]:

	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	
3	176560	Google Phone	1	600	
4	176560	Wired Headphones	1	11.99	

	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
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
```

1.1 Data cleaning

Drop Nan

[76]:
```
nan_df = all_Data[all_Data.isna().any(axis=1)]
nan_df.head()

all_Data = all_Data.dropna(how='all')
all_Data.head()
```

[76]:
```
  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
```

Drop 'Or' form Order Date

[77]:
```
all_Data = all_Data [all_Data['Order Date'].str[0:2] != 'Or']
all_Data.head ()
```

[77]:
```
  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
```

Converte Data to the correct type

[90]:
```
#all_Data['Quatity Ordered'] = pd.to_numeric(['Quantity Ordered'])
all_Data['Price Each'] = pd.to_numeric(all_Data['Price Each'], errors='coerce')
all_Data['Quantity Ordered'] = pd.to_numeric(all_Data['Quantity Ordered'],␣
↪errors='coerce')
all_Data.head()

#data = data.dropna(subset=['my_column'])

#print(all_Data['Quantity Ordered'].unique())
```

[90]:

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

Adding additional columns

adding the month columns

[91]:
```
all_Data['Month'] = all_Data['Order Date'].str[0:2]
all_Data['Month'] = all_Data['Month'].astype('int32')

all_Data.head()
```

[91]:

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

adding sale columns

[92]:
```
all_Data ['Sales'] =  all_Data['Quantity Ordered'] * all_Data ['Price Each']
all_Data.head()
```

[92]:

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

add a city column

[115]:
```
# let's use .apply function

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

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


all_Data['City'] = all_Data['Purchase Address'].apply(lambda x : get_city(x) +␣
↪' (' + get_State(x) + ' )')
all_Data.head()
```

[115]:

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

	City
0	Dallas (TX)

```
2         Boston (MA )
3    Los Angeles (CA )
4    Los Angeles (CA )
5    Los Angeles (CA )
```

The best month for sales , and how mush was earned that month

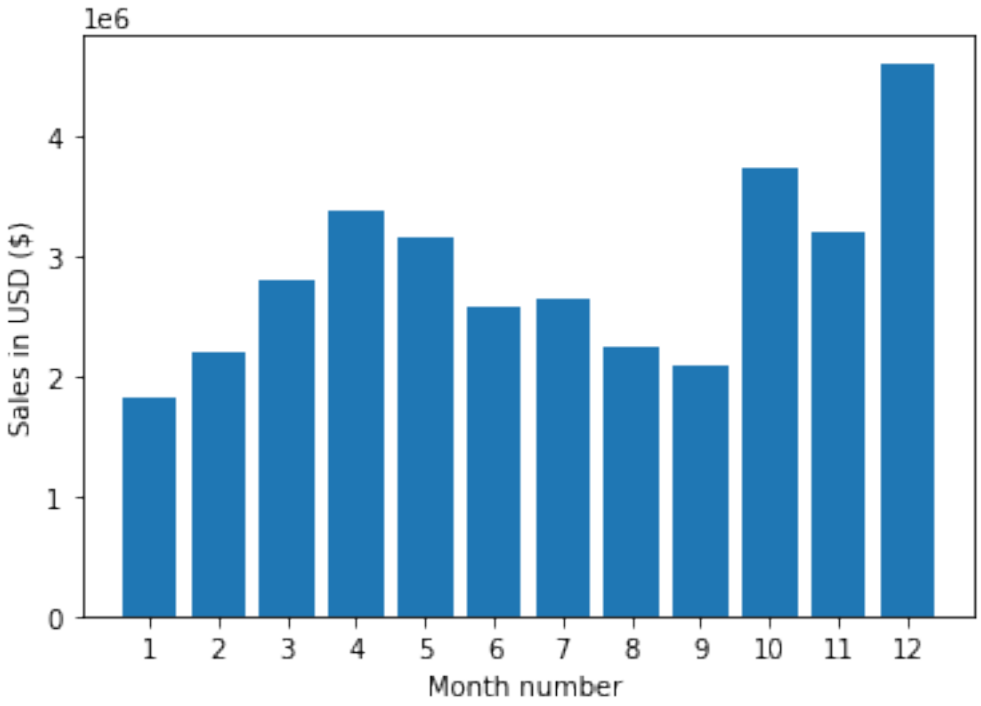
[98]:
```
all_Data.groupby('Month').sum()
```

[98]:

Month	Quantity Ordered	Price Each	Sales
1	10903	1.811768e+06	1.822257e+06
2	13449	2.188885e+06	2.202022e+06
3	17005	2.791208e+06	2.807100e+06
4	20558	3.367671e+06	3.390670e+06
5	18667	3.135125e+06	3.152607e+06
6	15253	2.562026e+06	2.577802e+06
7	16072	2.632540e+06	2.647776e+06
8	13448	2.230345e+06	2.244468e+06
9	13109	2.084992e+06	2.097560e+06
10	22703	3.715555e+06	3.736727e+06
11	19798	3.180601e+06	3.199603e+06
12	28114	4.588415e+06	4.613443e+06

[95]:
```
result = all_Data.groupby('Month').sum()
```

[99]:
```
import matplotlib.pyplot as plt

Months = range(1,13)

plt.bar(Months , result['Sales'])
plt.xticks(Months)
plt.ylabel('Sales in USD ($)')
plt.xlabel('Month number ')
plt.show()
```



City had the highest number of sales

[127]:
```
all_Data.groupby('City').sum()
```

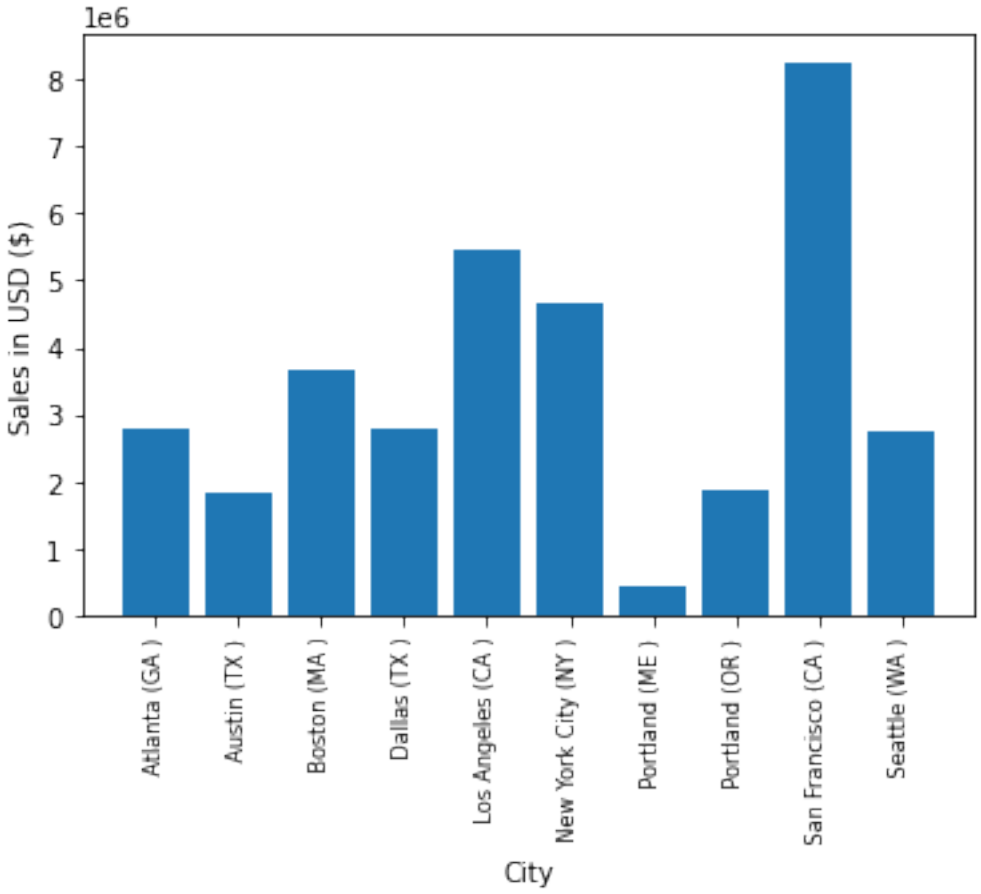
[127]:

City	Quantity Ordered	Price Each	Sales	Month
Atlanta (GA)	16602	2.779908e+06	2.795499e+06	104794
Austin (TX)	11153	1.809874e+06	1.819582e+06	69829
Boston (MA)	22528	3.637410e+06	3.661642e+06	141112
Dallas (TX)	16730	2.752628e+06	2.767975e+06	104620
Los Angeles (CA)	33289	5.421435e+06	5.452571e+06	208325
New York City (NY)	27932	4.635371e+06	4.664317e+06	175741
Portland (ME)	2750	4.471893e+05	4.497583e+05	17144
Portland (OR)	11303	1.860558e+06	1.870732e+06	70621
San Francisco (CA)	50239	8.211462e+06	8.262204e+06	315520
Seattle (WA)	16553	2.733296e+06	2.747755e+06	104941

[120]:
```
results = all_Data.groupby('City').sum()
```

[132]:
```
import matplotlib.pyplot as plt

# there is a problem of order here if we use the simple following script:␣
↪cities = all_Data[('City')].unique()
```

```
cities = [ city for city , df in all_Data.groupby('City') ] # prob fixed
plt.bar(cities , results['Sales'])
plt.xticks(cities , rotation = 'vertical' , size = 8)
plt.ylabel ('Sales in USD ($)')
plt.xlabel ('City')
plt.show()
```



the time that we should display advertissements to maximize likehood of customer's buying product

[136]:
```
all_Data['Order Date'] = pd.to_datetime(all_Data['Order Date'])
all_Data.head()
```

[136]:

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

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

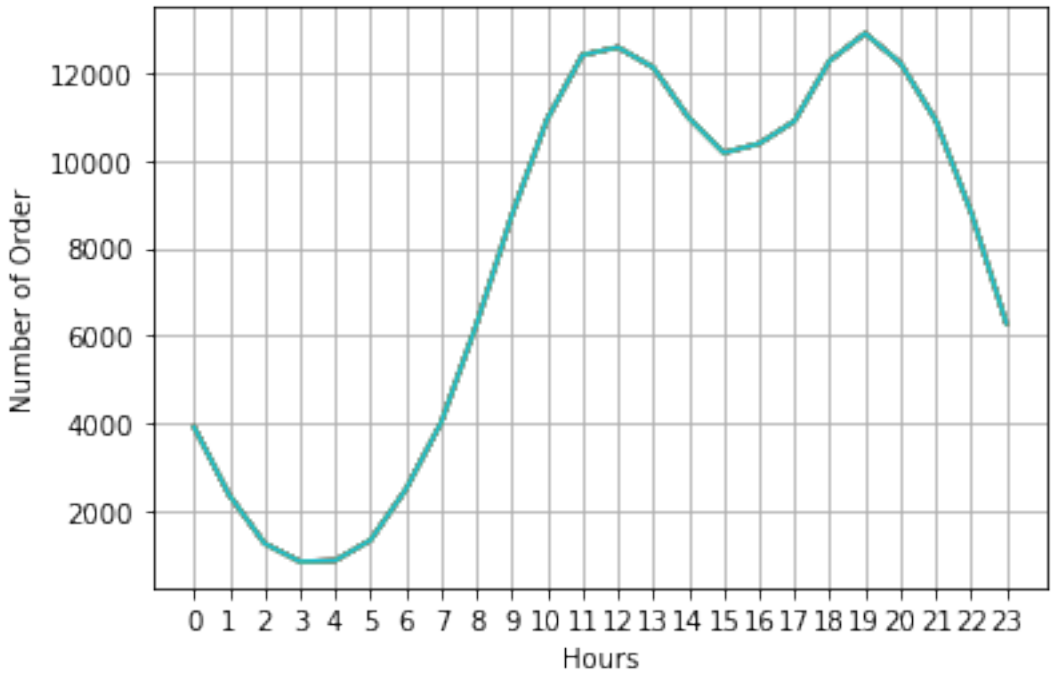
[138]:
```
all_Data['Hour'] = all_Data['Order Date'].dt.hour
all_Data['Minute'] = all_Data['Order Date'].dt.minute

all_Data.head()
```

[138]:
```
  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   Sales  Month  \
0 2019-04-19 08:46:00          917 1st St, Dallas, TX 75001   23.90      4
2 2019-04-07 22:30:00     682 Chestnut St, Boston, MA 02215   99.99      4
3 2019-04-12 14:38:00  669 Spruce St, Los Angeles, CA 90001  600.00      4
4 2019-04-12 14:38:00  669 Spruce St, Los Angeles, CA 90001   11.99      4
5 2019-04-30 09:27:00     333 8th St, Los Angeles, CA 90001   11.99      4

                 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
```

[152]:
```
hours = [ hour for hour , df in all_Data.groupby('Hour') ]

plt.xticks(hours)
plt.xlabel ('Hours')
plt.ylabel ('Number of Order')
plt.grid ()
```

```
plt.plot(Hours , all_Data.groupby('Hour').count())
```

[152]:
```
[<matplotlib.lines.Line2D at 0x1b21460c190>,
 <matplotlib.lines.Line2D at 0x1b21460c280>,
 <matplotlib.lines.Line2D at 0x1b21460cd90>,
 <matplotlib.lines.Line2D at 0x1b21460cbe0>,
 <matplotlib.lines.Line2D at 0x1b2087f1c40>,
 <matplotlib.lines.Line2D at 0x1b2087f1d00>,
 <matplotlib.lines.Line2D at 0x1b2087f1460>,
 <matplotlib.lines.Line2D at 0x1b2087f1130>,
 <matplotlib.lines.Line2D at 0x1b2087f1040>,
 <matplotlib.lines.Line2D at 0x1b2087f12e0>]
```



The product are most often solde together

[163]:
```
df = all_Data[all_Data['Order ID'].duplicated(keep =False)]

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

df = df[['Order ID', 'Grouped']].drop_duplicates()
df.head ()
```

```
<ipython-input-163-0f6a281990ac>:3: 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
  df['Grouped'] = df.groupby('Order ID')['Product'].transform(lambda x:
','.join(x))
```

[163]:

	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

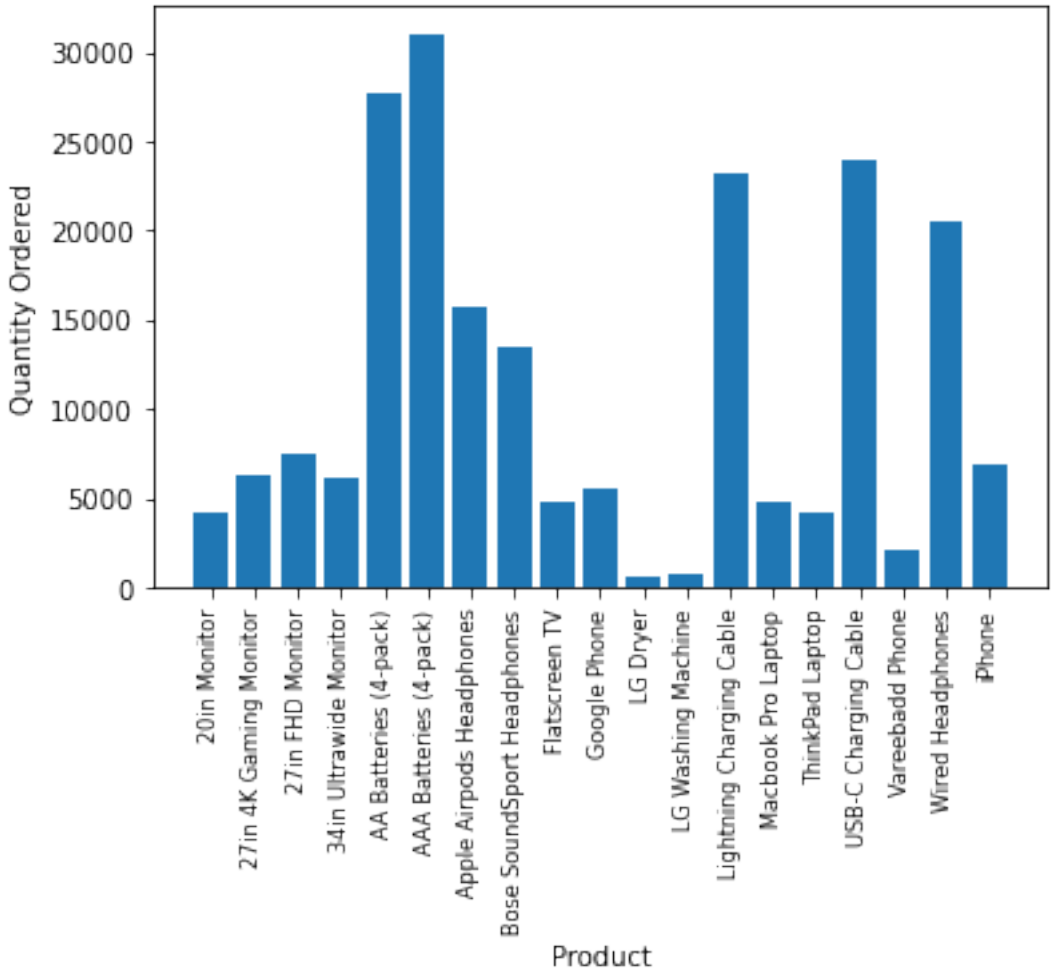
[171]:
```
from itertools import combinations
from collections import Counter


count = Counter()
for row in df['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
```

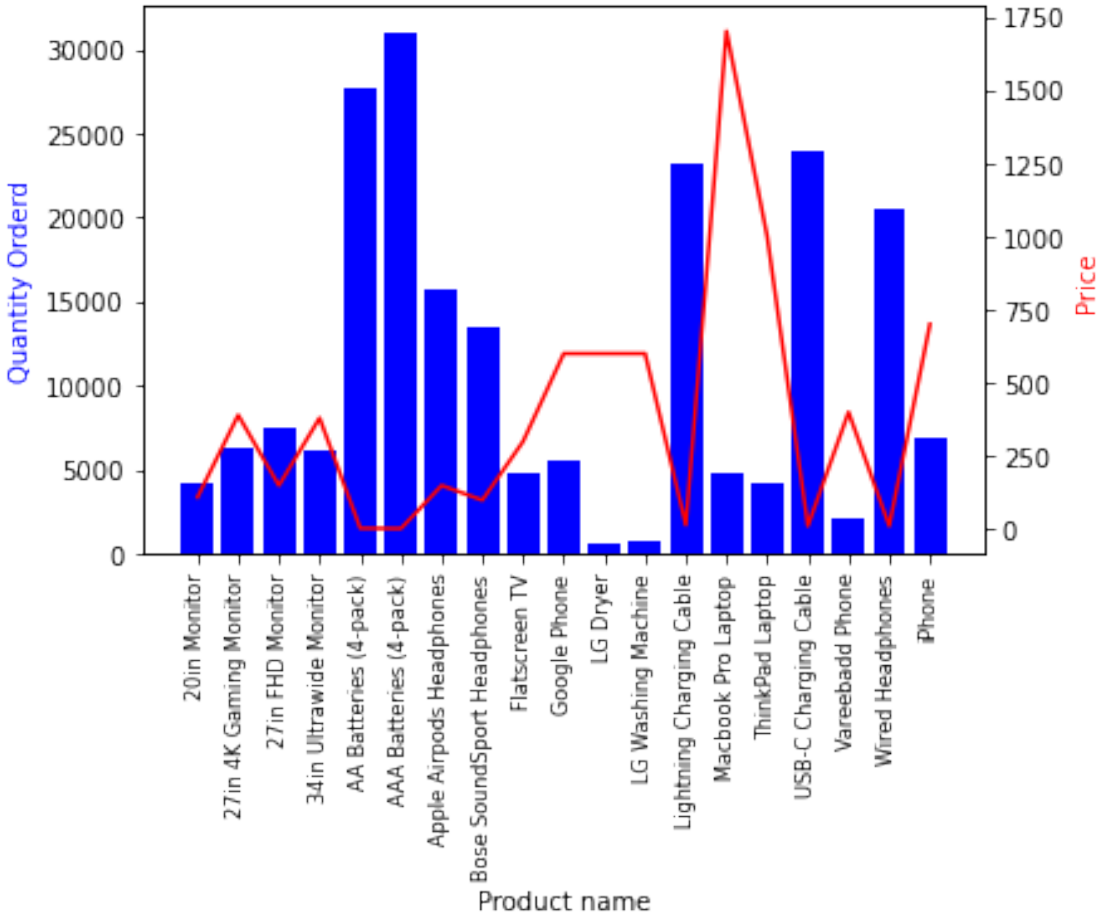
The most sold product

[174]:
```
product_group = all_Data.groupby('Product')
quantity_Orderd = product_group.sum()['Quantity Ordered']

products = [ product for product, df in product_group]



plt.bar(products , quantity_Orderd)
plt.xticks(products , rotation = 'vertical' , size = 8)
plt.ylabel('Quantity Ordered')
plt.xlabel('Product')
```

```
plt.show ()
```



[188]:
```
price = all_Data.groupby('Product').mean()['Price Each']


fig, ax1 = plt.subplots()
ax2 = ax1.twinx()
ax1.bar(products , quantity_Orderd, color='b')
ax2.plot(products , price , 'r-')

ax1.set_xlabel('Product name')
ax1.set_ylabel('Quantity Orderd', color ='b')
ax2.set_ylabel('Price', color ='r')
ax1.set_xticklabels(products, rotation = 'vertical' , size = 8)
plt.show()
```


Quantity Orderd
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

[ ]: