

```
In [15]: ## import our packages
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import os
```

```
In [30]: ## load our data
df = pd.read_csv('all_data.csv')
d = pd.read_csv('Sales_April_2019.csv')
df.head()
d.head()
```

Out[30]:

	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

```
In [23]: df.head
```

Out[23]:

		Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address
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		
...			
186845	259353		AAA Batteries (4-pack)	3	2.99		
186846	259354		iPhone	1	700		
186847	259355		iPhone	1	700		
186848	259356		34in Ultrawide Monitor	1	379.99		
186849	259357		USB-C Charging Cable	1	11.95		
...			
186845	09/17/19 20:56		840 Highland St, Los Angeles, CA 90001				
186846	09/01/19 16:00		216 Dogwood St, San Francisco, CA 94016				
186847	09/23/19 07:39		220 12th St, San Francisco, CA 94016				
186848	09/19/19 17:30		511 Forest St, San Francisco, CA 94016				
186849	09/30/19 00:18		250 Meadow St, San Francisco, CA 94016				

[186850 rows x 6 columns]>

```
In [40]: ## now we are creating a new column for months
df = df.dropna(how='any')
df['Month'] = df['Order Date'].str[0:2]
df.head()
```

Out[40]:

	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	04
2	176559	Bose SoundSport Headphones	1	99.99	04/07/19 22:30	682 Chestnut St, Boston, MA 02215	04
3	176560	Google Phone	1	600	04/12/19 14:38	669 Spruce St, Los Angeles, CA 90001	04
4	176560	Wired Headphones	1	11.99	04/12/19 14:38	669 Spruce St, Los Angeles, CA 90001	04
5	176561	Wired Headphones	1	11.99	04/30/19 09:27	333 8th St, Los Angeles, CA 90001	04

```
In [29]: df
```

Out[29]:

	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	04
1	NaN	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	04
3	176560	Google Phone	1	600	04/12/19 14:38	669 Spruce St, Los Angeles, CA 90001	04
4	176560	Wired Headphones	1	11.99	04/12/19 14:38	669 Spruce St, Los Angeles, CA 90001	04
...
186845	259353	AAA Batteries (4-pack)	3	2.99	09/17/19 20:56	840 Highland St, Los Angeles, CA 90001	09
186846	259354	iPhone	1	700	09/01/19 16:00	216 Dogwood St, San Francisco, CA 94016	09
186847	259355	iPhone	1	700	09/23/19 07:39	220 12th St, San Francisco, CA 94016	09
186848	259356	34in Ultrawide Monitor	1	379.99	09/19/19 17:30	511 Forest St, San Francisco, CA 94016	09
186849	259357	USB-C Charging Cable	1	11.95	09/30/19 00:18	250 Meadow St, San Francisco, CA 94016	09

186850 rows × 7 columns

In [37]:

```
## now we want to find our null values and drop them to clean the data
nana_df = df[df.isna().any(axis=1)]
nana_df
df = df.dropna(how='any')
df
```

Out[37]:

	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	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
5	176561	Wired Headphones	1	11.99	04/30/19 09:27	333 8th St, Los Angeles, CA 90001
...
186845	259353	AAA Batteries (4-pack)	3	2.99	09/17/19 20:56	840 Highland St, Los Angeles, CA 90001
186846	259354	iPhone	1	700	09/01/19 16:00	216 Dogwood St, San Francisco, CA 94016
186847	259355	iPhone	1	700	09/23/19 07:39	220 12th St, San Francisco, CA 94016
186848	259356	34in Ultrawide Monitor	1	379.99	09/19/19 17:30	511 Forest St, San Francisco, CA 94016
186849	259357	USB-C Charging Cable	1	11.95	09/30/19 00:18	250 Meadow St, San Francisco, CA 94016

186305 rows × 6 columns

In [48]:

```
all_data=df[df['Order Date'].str[0,2]=='0r']
all_data
```

Out[48]:

	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	04
2	176559	Bose SoundSport Headphones	1	99.99	04/07/19 22:30	682 Chestnut St, Boston, MA 02215	04
3	176560	Google Phone	1	600	04/12/19 14:38	669 Spruce St, Los Angeles, CA 90001	04
4	176560	Wired Headphones	1	11.99	04/12/19 14:38	669 Spruce St, Los Angeles, CA 90001	04
5	176561	Wired Headphones	1	11.99	04/30/19 09:27	333 8th St, Los Angeles, CA 90001	04
...
186845	259353	AAA Batteries (4-pack)	3	2.99	09/17/19 20:56	840 Highland St, Los Angeles, CA 90001	09
186846	259354	iPhone	1	700	09/01/19 16:00	216 Dogwood St, San Francisco, CA 94016	09
186847	259355	iPhone	1	700	09/23/19 07:39	220 12th St, San Francisco, CA 94016	09
186848	259356	34in Ultrawide Monitor	1	379.99	09/19/19 17:30	511 Forest St, San Francisco, CA 94016	09
186849	259357	USB-C Charging Cable	1	11.95	09/30/19 00:18	250 Meadow St, San Francisco, CA 94016	09

186305 rows × 7 columns

In [58]: *## now we are going to convert the month column to an integar*

```
all_data = df[df['Order Date'].str[0:2]!='0r']
all_data

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

Out[58]:

	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	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

In [60]: *## Covert columns to the correct type and add a sales column*

```
all_data['Quantity Ordered'] = pd.to_numeric(all_data['Quantity Ordered'])
all_data['Price Each'] = pd.to_numeric(all_data['Price Each'])
all_data['Sales'] = all_data['Quantity Ordered'] * all_data['Price Each']
all_data
```

Out[60]:

	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
4	176560	Wired Headphones	1	11.99	04/12/19 14:38	669 Spruce St, Los Angeles, CA 90001	4	11.99
5	176561	Wired Headphones	1	11.99	04/30/19 09:27	333 8th St, Los Angeles, CA 90001	4	11.99
...
186845	259353	AAA Batteries (4-pack)	3	2.99	09/17/19 20:56	840 Highland St, Los Angeles, CA 90001	9	8.97
186846	259354	iPhone	1	700.00	09/01/19 16:00	216 Dogwood St, San Francisco, CA 94016	9	700.00
186847	259355	iPhone	1	700.00	09/23/19 07:39	220 12th St, San Francisco, CA 94016	9	700.00
186848	259356	34in Ultrawide Monitor	1	379.99	09/19/19 17:30	511 Forest St, San Francisco, CA 94016	9	379.99
186849	259357	USB-C Charging Cable	1	11.95	09/30/19 00:18	250 Meadow St, San Francisco, CA 94016	9	11.95

185950 rows × 8 columns

Question 1: what was the best month for sales? how much was earned that month

In [71]:

```
papa = all_data.groupby('Month').sum().sort_values("Sales",ascending=False)
## grouping the data by month, the sum actualizes the code sort by sorts it by sales high to low
```

Out[71]:

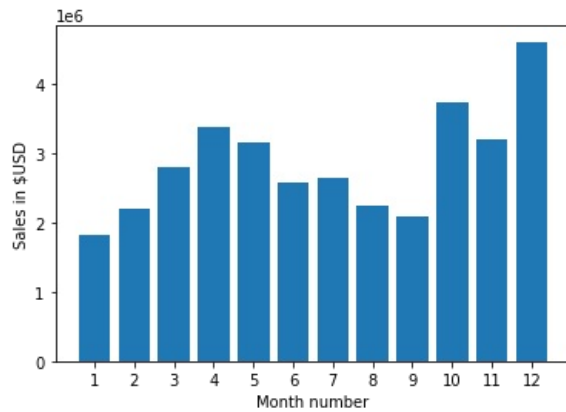
	Quantity Ordered	Price Each	Sales
Month			
12	28114	4588415.41	4613443.34
10	22703	3715554.83	3736726.88
4	20558	3367671.02	3390670.24
11	19798	3180600.68	3199603.20
5	18667	3135125.13	3152606.75
3	17005	2791207.83	2807100.38
7	16072	2632539.56	2647775.76
6	15253	2562025.61	2577802.26
8	13448	2230345.42	2244467.88
2	13449	2188884.72	2202022.42
9	13109	2084992.09	2097560.13
1	10903	1811768.38	1822256.73

In [99]:

```
## lets graph it
results = all_data.groupby('Month').sum()
months = range(1,13)
plt.bar(months,results['Sales'])
plt.xticks(months)
plt.ylabel("Sales in $USD")
plt.xlabel('Month number')
plt.show
```

Out[99]:

<function matplotlib.pyplot.show(close=None, block=None)>



Question 2: what city had the highest sales

```
In [107]: ## Add city column first
## we used .apply() to create a function to pull out the city from the address
all_data['City'] = all_data['Purchase Address'].apply(lambda x: x.split(',')[1])
all_data
```

Out[107]:

	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
2	176559	Bose SoundSport Headphones	1	99.99	04/07/19 22:30	682 Chestnut St, Boston, MA 02215	4	99.99	Boston
3	176560	Google Phone	1	600.00	04/12/19 14:38	669 Spruce St, Los Angeles, CA 90001	4	600.00	Los Angeles
4	176560	Wired Headphones	1	11.99	04/12/19 14:38	669 Spruce St, Los Angeles, CA 90001	4	11.99	Los Angeles
5	176561	Wired Headphones	1	11.99	04/30/19 09:27	333 8th St, Los Angeles, CA 90001	4	11.99	Los Angeles
...
186845	259353	AAA Batteries (4-pack)	3	2.99	09/17/19 20:56	840 Highland St, Los Angeles, CA 90001	9	8.97	Los Angeles
186846	259354	iPhone	1	700.00	09/01/19 16:00	216 Dogwood St, San Francisco, CA 94016	9	700.00	San Francisco
186847	259355	iPhone	1	700.00	09/23/19 07:39	220 12th St, San Francisco, CA 94016	9	700.00	San Francisco
186848	259356	34in Ultrawide Monitor	1	379.99	09/19/19 17:30	511 Forest St, San Francisco, CA 94016	9	379.99	San Francisco
186849	259357	USB-C Charging Cable	1	11.95	09/30/19 00:18	250 Meadow St, San Francisco, CA 94016	9	11.95	San Francisco

185950 rows × 9 columns

```
In [110]: def get_city(address):
            return address.split(",")[1].strip(" ")

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

all_data['City'] = all_data['Purchase Address'].apply(lambda x: f"{get_city(x)} ({get_state(x)})")
all_data.head()
```

Out[110]:	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)
4	176560	Wired Headphones	1	11.99	04/12/19 14:38	669 Spruce St, Los Angeles, CA 90001	4	11.99	Los Angeles (CA)
5	176561	Wired Headphones	1	11.99	04/30/19 09:27	333 8th St, Los Angeles, CA 90001	4	11.99	Los Angeles (CA)

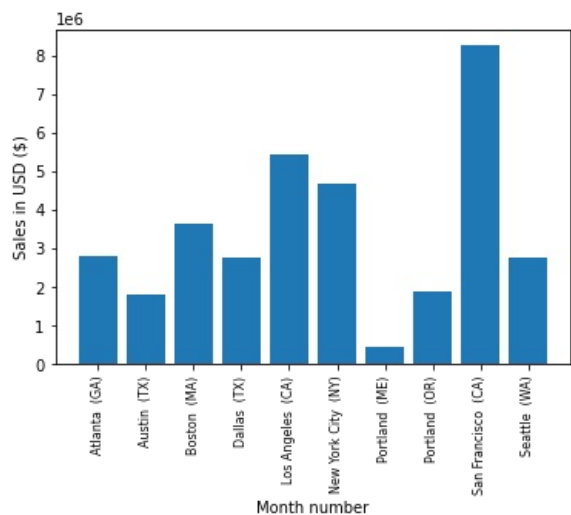
```
In [111]: all_data.groupby('City').sum().sort_values("Sales",ascending=False)
```

Out[111]:	Quantity Ordered	Price Each	Month	Sales
City				
San Francisco (CA)	50239	8211461.74	315520	8262203.91
Los Angeles (CA)	33289	5421435.23	208325	5452570.80
New York City (NY)	27932	4635370.83	175741	4664317.43
Boston (MA)	22528	3637409.77	141112	3661642.01
Atlanta (GA)	16602	2779908.20	104794	2795498.58
Dallas (TX)	16730	2752627.82	104620	2767975.40
Seattle (WA)	16553	2733296.01	104941	2747755.48
Portland (OR)	11303	1860558.22	70621	1870732.34
Austin (TX)	11153	1809873.61	69829	1819581.75
Portland (ME)	2750	447189.25	17144	449758.27

```
In [115]: import matplotlib.pyplot as plt

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

plt.bar(keys,all_data.groupby(['City']).sum()['Sales'])
plt.ylabel('Sales in USD ($)')
plt.xlabel('Month number')
plt.xticks(keys, rotation='vertical', size=8)
plt.show()
```



```
In [165]:
```

Question 3: what time should we advertise to maximize likelihood of customer's buying Product

```
In [120]: ## Now we will have to change to orderdate data to a datetime data type and create new columns

all_data['Order Date']= pd.to_datetime(all_data['Order Date'])
all_data['Hours'] = all_data['Order Date'].dt.hour
all_data
```

Out[120]:

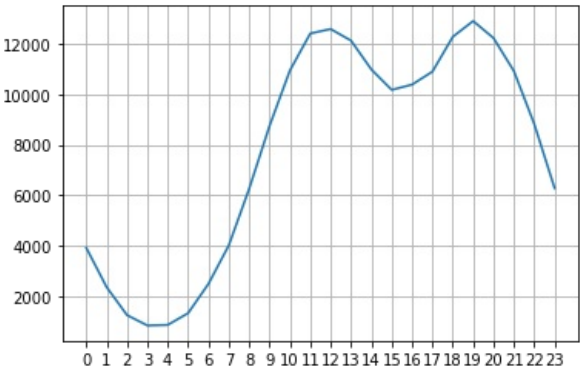
	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address	Month	Sales	City	Hours
0	176558	USB-C Charging Cable	2	11.95	2019-04-19 08:46:00	917 1st St, Dallas, TX 75001	4	23.90	Dallas (TX)	8
2	176559	Bose SoundSport Headphones	1	99.99	2019-04-07 22:30:00	682 Chestnut St, Boston, MA 02215	4	99.99	Boston (MA)	22
3	176560	Google Phone	1	600.00	2019-04-12 14:38:00	669 Spruce St, Los Angeles, CA 90001	4	600.00	Los Angeles (CA)	14
4	176560	Wired Headphones	1	11.99	2019-04-12 14:38:00	669 Spruce St, Los Angeles, CA 90001	4	11.99	Los Angeles (CA)	14
5	176561	Wired Headphones	1	11.99	2019-04-30 09:27:00	333 8th St, Los Angeles, CA 90001	4	11.99	Los Angeles (CA)	9
...
186845	259353	AAA Batteries (4-pack)	3	2.99	2019-09-17 20:56:00	840 Highland St, Los Angeles, CA 90001	9	8.97	Los Angeles (CA)	20
186846	259354	iPhone	1	700.00	2019-09-01 16:00:00	216 Dogwood St, San Francisco, CA 94016	9	700.00	San Francisco (CA)	16
186847	259355	iPhone	1	700.00	2019-09-23 07:39:00	220 12th St, San Francisco, CA 94016	9	700.00	San Francisco (CA)	7
186848	259356	34in Ultrawide Monitor	1	379.99	2019-09-19 17:30:00	511 Forest St, San Francisco, CA 94016	9	379.99	San Francisco (CA)	17
186849	259357	USB-C Charging Cable	1	11.95	2019-09-30 00:18:00	250 Meadow St, San Francisco, CA 94016	9	11.95	San Francisco (CA)	0

185950 rows × 10 columns

In [134..

```
keys = [pair for pair, df in all_data.groupby(['Hours'])]
all_data['Count'] = 1
plt.plot(keys, all_data.groupby(['Hours']).count()['Count'])
plt.xticks(keys)
plt.grid()
plt.show()
```

it is best for us to advertise around 11am or 7pm



Question 4: what products are most often sold together

In [148..

```
## First we will have to find the duplicate order ids to show that these products were bought together
df= all_data[all_data['Order ID'].duplicated(keep=False)]
df.head(20)
## Next we will combine and transform the datat so each product that has the same order id is group together
df["Grouped"] =df.groupby('Order ID')['Product'].transform(lambda x: ','.join(x))
## now that they are combined we can remove any duplicates
df =df[['Order ID', 'Grouped']].drop_duplicates()
df
```

C:\Users\coold\AppData\Local\Temp\ipykernel_768\3059447652.py:5: 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))
```

Out[148]:

	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
...
186781	259296	Apple Airpods Headphones,Apple Airpods Headphones
186783	259297	iPhone,Lightning Charging Cable,Lightning Char...
186791	259303	34in Ultrawide Monitor,AA Batteries (4-pack)
186803	259314	Wired Headphones,AAA Batteries (4-pack)
186841	259350	Google Phone,USB-C Charging Cable

7136 rows × 2 columns

In [160]..

now we are trying to find out which pairs were the most sold together

```
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)))
count.most_common(10)
```

Out[160]:

[(('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 do you think it sold the most ?

In [163]..

all_data.groupby("Product").sum().sort_values("Quantity Ordered", ascending=False)

Out[163]:

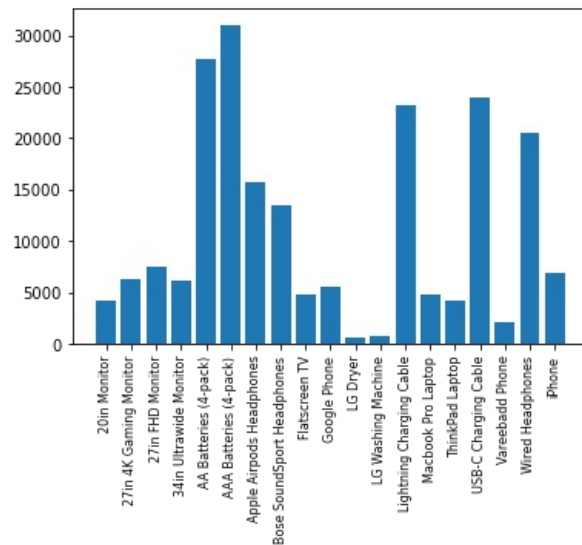
	Quantity Ordered	Price Each	Month	Sales	Hours	Count
Product						
AAA Batteries (4-pack)	31017	61716.59	146370	92740.83	297332	20641
AA Batteries (4-pack)	27635	79015.68	145558	106118.40	298342	20577
USB-C Charging Cable	23975	261740.85	154819	286501.25	314645	21903
Lightning Charging Cable	23217	323787.10	153092	347094.15	312529	21658
Wired Headphones	20557	226395.18	133397	246478.43	271720	18882
Apple Airpods Headphones	15661	2332350.00	109477	2349150.00	223304	15549
Bose SoundSport Headphones	13457	1332366.75	94113	1345565.43	192445	13325
27in FHD Monitor	7550	1125974.93	52558	1132424.50	107540	7507
iPhone	6849	4789400.00	47941	4794300.00	98657	6842
27in 4K Gaming Monitor	6244	2429637.70	44440	2435097.56	90916	6230
34in Ultrawide Monitor	6199	2348718.19	43304	2355558.01	89076	6181
Google Phone	5532	3315000.00	38305	3319200.00	79479	5525
Flatscreen TV	4819	1440000.00	34224	1445700.00	68815	4800
Macbook Pro Laptop	4728	8030800.00	33548	8037600.00	68261	4724
ThinkPad Laptop	4130	4127958.72	28950	4129958.70	59746	4128
20in Monitor	4129	451068.99	29336	454148.71	58764	4101
Vareebadd Phone	2068	826000.00	14309	827200.00	29472	2065
LG Washing Machine	666	399600.00	4523	399600.00	9785	666
LG Dryer	646	387600.00	4383	387600.00	9326	646

In [172]..

Product_group = all_data.groupby('Product')


```
quantity_ordered = Product_group.sum()['Quantity Ordered']
products = [product for product, df in Product_group]
```

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



```
In [178]: ## AAA batteries are purchased the most due to them being a item that has to constantly be replaced and a low p
price = all_data.groupby('Product').mean()['Price Each'].sort_values()
prices = all_data.groupby('Product').mean()['Price Each']
price
```

```
Out[178]: Product
AAA Batteries (4-pack)          2.99
AA Batteries (4-pack)          3.84
USB-C Charging Cable          11.95
Wired Headphones              11.99
Lightning Charging Cable      14.95
Bose SoundSport Headphones    99.99
20in Monitor                  109.99
27in FHD Monitor              149.99
Apple AirPods Headphones      150.00
Flatscreen TV                 300.00
34in Ultrawide Monitor        379.99
27in 4K Gaming Monitor        389.99
Vairebadd Phone               400.00
Google Phone                  600.00
LG Washing Machine            600.00
LG Dryer                      600.00
iPhone                        700.00
ThinkPad Laptop               999.99
Macbook Pro Laptop            1700.00
Name: Price Each, dtype: float64
```

```
In [179]: fig, ax1 = plt.subplots()

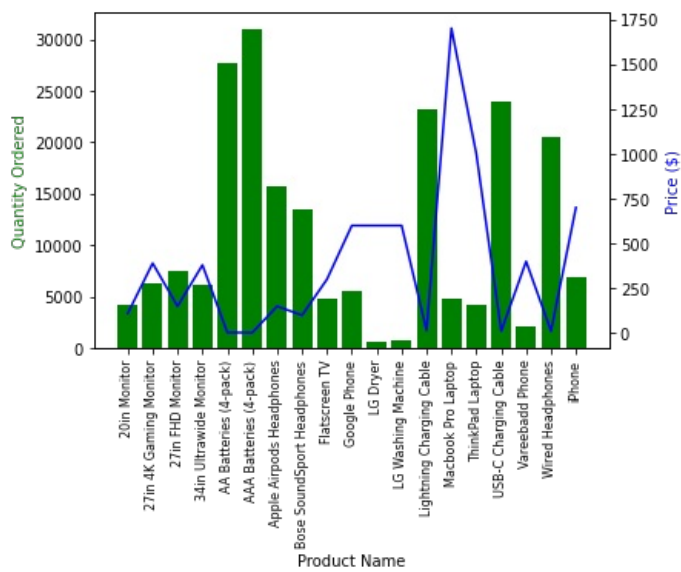
ax2 = ax1.twinx()
ax1.bar(keys, quantity_ordered, color='g')
ax2.plot(keys, prices, color='b')

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()
## you use a combination chart when you want to see a correlations between two variables
## example of this would be the quantity of an item sold relative to their price
```

C:\Users\coold\AppData\Local\Temp\ipykernel_768\1776558852.py:10: UserWarning: FixedFormatter should only be used together with FixedLocator

ax1.set_xticklabels(keys, rotation='vertical', size=8)
C:\Users\coold\AppData\Local\Temp\ipykernel_768\1776558852.py:12: UserWarning: Matplotlib is currently using module://matplotlib_inline.backend_inline, which is a non-GUI backend, so cannot show the figure.
fig.show()



In [180] keys

```
Out[180]: ['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']
```