

```
In [34]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import os
```

Performing Data Analysis on the Sales Data

PREPARING THE DATA

```
In [35]: path = r'D:\*****\Sales\Sales_Data'
```

```
In [36]: files = [file for file in os.listdir(path)]
files
```

```
Out[36]: ['all_data.csv',
'Sales_April_2019.csv',
'Sales_August_2019.csv',
'Sales_December_2019.csv',
'Sales_February_2019.csv',
'Sales_January_2019.csv',
'Sales_July_2019.csv',
'Sales_June_2019.csv',
'Sales_March_2019.csv',
'Sales_May_2019.csv',
'Sales_November_2019.csv',
'Sales_October_2019.csv',
'Sales_September_2019.csv']
```

```
In [37]: #Concatinating all datasets to one dataframe

all_data = pd.DataFrame()

for file in files[1:]:
    current_data = pd.read_csv(path + '/' + file)
    all_data = pd.concat([all_data,current_data])

all_data.shape
```

Out[37]: (186850, 6)

ANALYSING MONTHLY SALES

```
In [38]: #check the dataframe
all_data.head()
```

Out[38]:

	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 [39]: #check for null values
all_data.isnull().sum()
```

Out[39]:

Order ID	545
Product	545
Quantity Ordered	545
Price Each	545
Order Date	545
Purchase Address	545
dtype:	int64

```
In [40]: all_data.dropna(how='all',inplace=True)
all_data.shape
```

```
Out[40]: (186305, 6)
```

```
In [ ]:
```

Finding the best month for sales

```
In [41]: #Extract month from Order Date
all_data['month'] = all_data['Order Date'].apply(lambda date : date.split('/')[0])
#Check the updatated dataframe
all_data
```

```
Out[41]:
```

	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

11681	259353	AAA Batteries (4-pack)	3	2.99	09/17/19 20:56	840 Highland St, Los Angeles, CA 90001	09	
11682	259354	iPhone	1	700	09/01/19 16:00	216 Dogwood St, San Francisco, CA 94016	09	
11683	259355	iPhone	1	700	09/23/19 07:39	220 12th St, San Francisco, CA 94016	09	
11684	259356	34in Ultrawide Monitor	1	379.99	09/19/19 17:30	511 Forest St, San Francisco, CA 94016	09	
11685	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 [42]: #Check the datatype of feature  
all_data.dtypes
```

```
Out[42]: Order ID          object  
Product          object  
Quantity Ordered  object  
Price Each       object  
Order Date       object  
Purchase Address  object  
month            object  
dtype: object
```

```
In [43]: #check unique values of month feature  
all_data['month'].unique()
```

```
Out[43]: array(['04', '05', 'Order Date', '08', '09', '12', '01', '02', '03', '07',  
               '06', '11', '10'], dtype=object)
```

```
In [44]: #Applying filter to remove invalid entry  
filter=all_data['month']=='Order Date'  
all_data = all_data[~filter]
```

```
In [45]: #Convert to month feature to integer  
all_data['month'] = all_data['month'].astype('int')
```

C:\Users\DELL PC\AppData\Local\Temp\ipykernel_13320\2262781322.py:2: 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 (https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)

```
all_data['month'] = all_data['month'].astype('int')
```

```
In [46]: #Check the datatype of feature  
all_data.dtypes
```

```
Out[46]: Order ID          object  
Product          object  
Quantity Ordered  object  
Price Each       object  
Order Date       object  
Purchase Address  object  
month            int32  
dtype: object
```

```
In [47]: #converting 'Quantity Ordered' feature to int  
all_data['Quantity Ordered']=all_data['Quantity Ordered'].astype('int')  
all_data['Quantity Ordered'].dtype
```

```
C:\Users\DELL PC\AppData\Local\Temp\ipykernel_13320\1384291444.py:2: 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 (https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)

```
all_data['Quantity Ordered']=all_data['Quantity Ordered'].astype('int')
```

```
Out[47]: dtype('int32')
```

```
In [48]: #converting 'Price Each' feature to float  
all_data['Price Each']=all_data['Price Each'].astype('float')
```

```
C:\Users\DELL PC\AppData\Local\Temp\ipykernel_13320\3981700378.py:2: 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 (https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)

```
all_data['Price Each']=all_data['Price Each'].astype('float')
```

```
In [49]: all_data.dtypes
```

```
Out[49]: Order ID          object
Product          object
Quantity Ordered  int32
Price Each       float64
Order Date       object
Purchase Address  object
month            int32
dtype: object
```

```
In [50]: #Add sales feature to dataframe
```

```
all_data['sales'] = all_data['Quantity Ordered'] * all_data['Price Each']
all_data.head()
```

C:\Users\DELL PC\AppData\Local\Temp\ipykernel_13320\357589813.py:2: 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 (https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)

```
all_data['sales'] = all_data['Quantity Ordered'] * all_data['Price Each']
```

```
Out[50]:
```

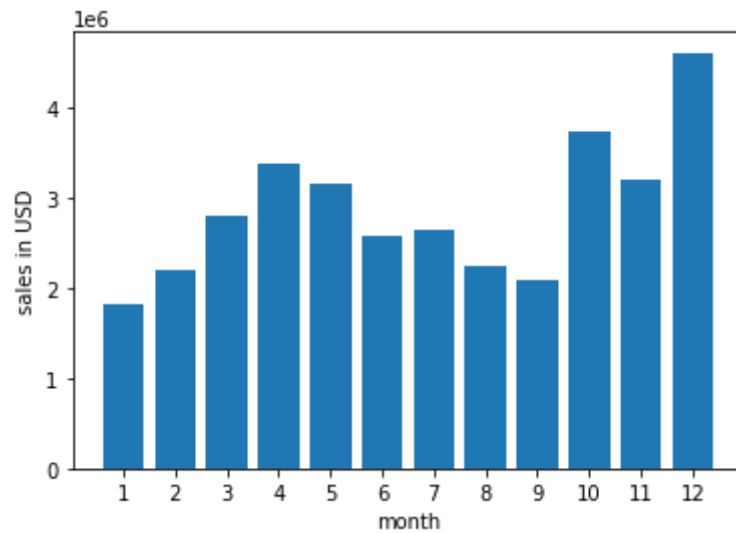
	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address	month	sales
0	176558	USB-C Charging Cable	2	11.95	04/19/19 08:46	917 1st St, Dallas, TX 75001	4	23.90
2	176559	Bose SoundSport Headphones	1	99.99	04/07/19 22:30	682 Chestnut St, Boston, MA 02215	4	99.99
3	176560	Google Phone	1	600.00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

```
In [51]: #Group by data on month feature  
data=all_data.groupby('month')['sales'].sum()  
data
```

```
Out[51]: 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
```

```
In [52]: #Creating a bar chart
plt.bar(data.index, data)
plt.xticks(data.index)
plt.xlabel('month')
plt.ylabel('sales in USD')
#Conclusion:- December month has the best sales
```

```
Out[52]: Text(0, 0.5, 'sales in USD')
```



```
In [ ]:
```

Analysing Maximum Order & Hour Analysis

Which city has maximum sales

```
In [53]: # Extract city form the purchase address
all_data['city'] = all_data['Purchase Address'].apply(lambda city : city.split(',')[ -2])
all_data.head()
```

C:\Users\DELL PC\AppData\Local\Temp\ipykernel_13320\959901239.py:2: 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 (https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)

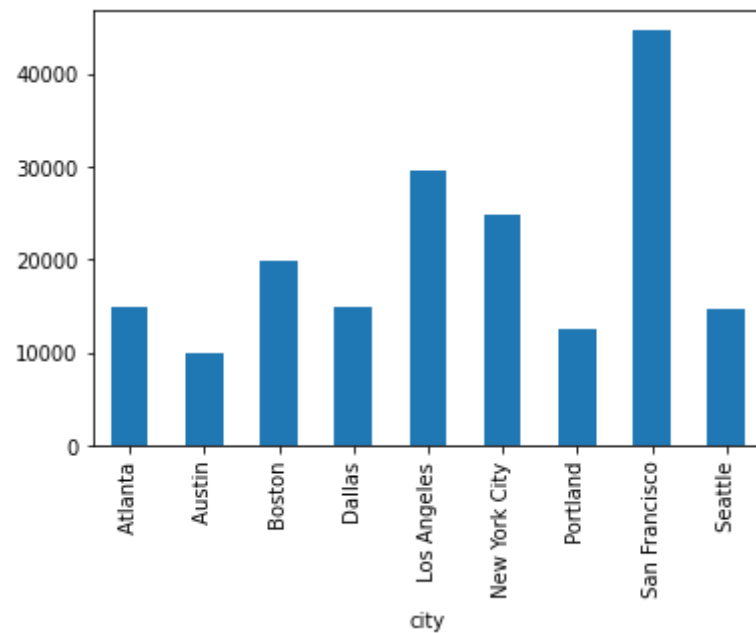
```
all_data['city'] = all_data['Purchase Address'].apply(lambda city : city.split(',')[ -2])
```

Out[53]:

	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

```
In [54]: #Count and plot by city
all_data.groupby('city')['city'].count().plot.bar()
all_data.groupby('city')['city'].count()
#Conclusion:- San Francisco has maximum orders
```

```
Out[54]: city
Atlanta      14881
Austin        9905
Boston       19934
Dallas       14820
Los Angeles  29605
New York City 24876
Portland     12465
San Francisco 44732
Seattle      14732
Name: city, dtype: int64
```



```
In [ ]:
```

At what times sales of a product is maximum

```
In [55]: #check the datatyoep of Order Date feature
all_data['Order Date'].dtypes
#string datatype
```

```
Out[55]: dtype('O')
```

```
In [56]: #Convert to datetime and extract hour
all_data['Hour']=pd.to_datetime(all_data['Order Date']).dt.hour
all_data.head()
```

C:\Users\DELL PC\AppData\Local\Temp\ipykernel_13320\205359938.py:2: 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 (https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)

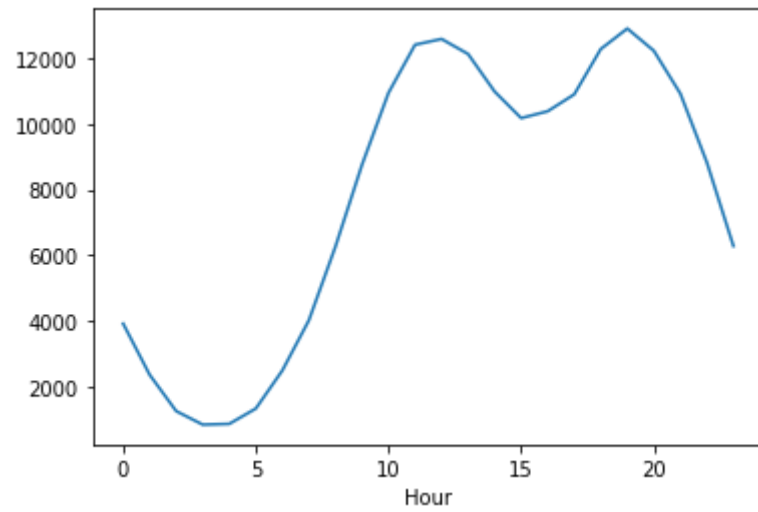
```
all_data['Hour']=pd.to_datetime(all_data['Order Date']).dt.hour
```

```
Out[56]:
```

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address	month	sales	city	Hour
0	176558	USB-C Charging Cable	2	11.95	04/19/19 08:46	917 1st St, Dallas, TX 75001	4	23.90	Dallas	8
2	176559	Bose SoundSport Headphones	1	99.99	04/07/19 22:30	682 Chestnut St, Boston, MA 02215	4	99.99	Boston	22
3	176560	Google Phone	1	600.00	04/12/19 14:38	669 Spruce St, Los Angeles, CA 90001	4	600.00	Los Angeles	14
4	176560	Wired Headphones	1	11.99	04/12/19 14:38	669 Spruce St, Los Angeles, CA 90001	4	11.99	Los Angeles	14
5	176561	Wired Headphones	1	11.99	04/30/19 09:27	333 8th St, Los Angeles, CA 90001	4	11.99	Los Angeles	9

```
In [57]: # Plot the orders by hours
hour=all_data.groupby('Hour')['Quantity Ordered'].count()
hour.plot.line()
hour.sort_values(ascending=False)
#Conclusion:- 19th hour has maximum orders
```

```
Out[57]: Hour
19      12905
12      12587
11      12411
18      12280
20      12228
13      12129
14      10984
10      10944
21      10921
17      10899
16      10384
15      10175
22       8822
9        8748
23       6275
8        6256
7        4011
0        3910
6        2482
1        2350
5        1321
2        1243
4         854
3         831
Name: Quantity Ordered, dtype: int64
```



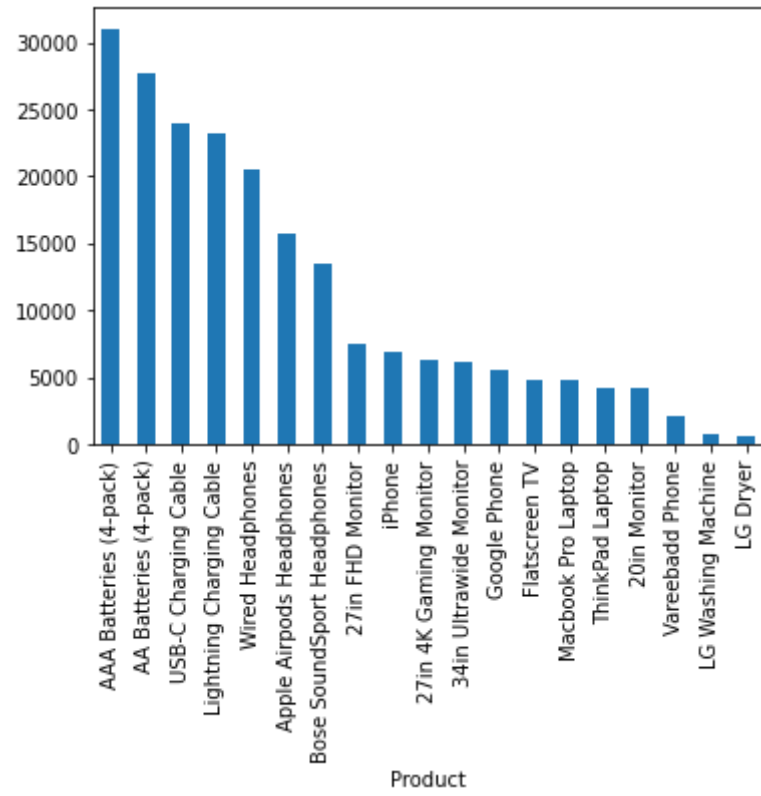
In []:

Analysing Most Sold Products

Which product is most sold and why?

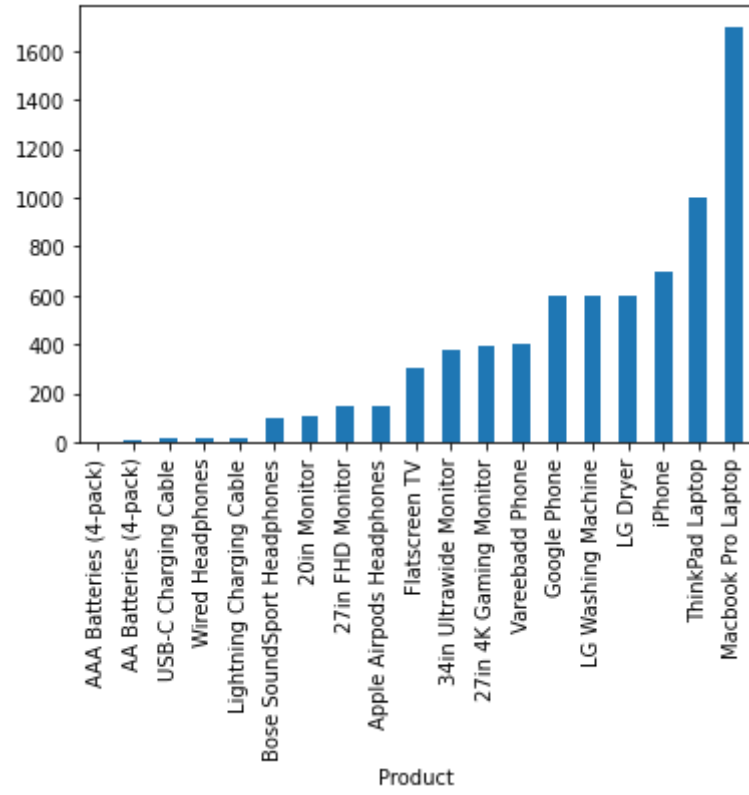
```
In [58]: #Group by on Product
all_data.groupby('Product')['Quantity Ordered'].sum().sort_values(ascending=False).plot.bar()
all_data.groupby('Product')['Quantity Ordered'].sum().sort_values(ascending=False)
#conclusion:- AAA Batteries (4-pack) is most sold
```

```
Out[58]: Product
AAA Batteries (4-pack)      31017
AA Batteries (4-pack)      27635
USB-C Charging Cable       23975
Lightning Charging Cable   23217
Wired Headphones           20557
Apple Airpods Headphones   15661
Bose SoundSport Headphones 13457
27in FHD Monitor           7550
iPhone                     6849
27in 4K Gaming Monitor     6244
34in Ultrawide Monitor     6199
Google Phone               5532
Flatscreen TV              4819
Macbook Pro Laptop         4728
ThinkPad Laptop            4130
20in Monitor               4129
Vareebadd Phone            2068
LG Washing Machine         666
LG Dryer                    646
Name: Quantity Ordered, dtype: int32
```



```
In [59]: #check mean price of every product
all_data.groupby('Product')['Price Each'].mean().sort_values().plot(kind='bar')
all_data.groupby('Product')['Price Each'].mean().sort_values()
#conclusion:- AAA Batteries (4-pack) has lowest mean price, so it is the most sold
```

```
Out[59]: 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
Vareebadd 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 [60]: #Creating Twin plots
products = all_data.groupby('Product')['Quantity Ordered'].sum().index
quantity = all_data.groupby('Product')['Quantity Ordered'].sum()
prices = all_data.groupby('Product')['Price Each'].mean()

fig,ax1 = plt.subplots()
ax2=ax1.twinx()
ax1.bar(products,quantity,color='r')
ax2.plot(products,prices, color='g')
ax1.set_xticklabels(products,rotation='vertical')

#conclusion:- cheaper the product, more it will be sold
```

```
Out[60]: [Text(0, 0, '20in Monitor'),
Text(1, 0, '27in 4K Gaming Monitor'),
Text(2, 0, '27in FHD Monitor'),
Text(3, 0, '34in Ultrawide Monitor'),
Text(4, 0, 'AA Batteries (4-pack)'),
Text(5, 0, 'AAA Batteries (4-pack)'),
Text(6, 0, 'Apple AirPods Headphones'),
Text(7, 0, 'Bose SoundSport Headphones'),
Text(8, 0, 'Flatscreen TV'),
Text(9, 0, 'Google Phone'),
Text(10, 0, 'LG Dryer'),
Text(11, 0, 'LG Washing Machine'),
Text(12, 0, 'Lightning Charging Cable'),
Text(13, 0, 'Macbook Pro Laptop'),
Text(14, 0, 'ThinkPad Laptop'),
Text(15, 0, 'USB-C Charging Cable'),
Text(16, 0, 'Vareebadd Phone'),
Text(17, 0, 'Wired Headphones'),
Text(18, 0, 'iPhone')]
```

In []:

Which Products are most often sold together

```
In [61]: df=all_data['Order ID'].duplicated(keep=False)
df2=all_data[df]
df2.head()
```

Out[61]:

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address	month	sales	city	Hour
3	176560	Google Phone	1	600.00	04/12/19 14:38	669 Spruce St, Los Angeles, CA 90001	4	600.00	Los Angeles	14
4	176560	Wired Headphones	1	11.99	04/12/19 14:38	669 Spruce St, Los Angeles, CA 90001	4	11.99	Los Angeles	14
18	176574	Google Phone	1	600.00	04/03/19 19:42	20 Hill St, Los Angeles, CA 90001	4	600.00	Los Angeles	19
19	176574	USB-C Charging Cable	1	11.95	04/03/19 19:42	20 Hill St, Los Angeles, CA 90001	4	11.95	Los Angeles	19
30	176585	Bose SoundSport Headphones	1	99.99	04/07/19 11:31	823 Highland St, Boston, MA 02215	4	99.99	Boston	11

```
In [62]: # creating a new feature in df2 dataframe
df2['Grouped'] = df2.groupby('Order ID')['Product'].transform(lambda x: ','.join(x))
df2.head()
```

C:\Users\DELL PC\AppData\Local\Temp\ipykernel_13320\2828232519.py:2: 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 (https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)

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

Out[62]:

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address	month	sales	city	Hour	Grouped
3	176560	Google Phone	1	600.00	04/12/19 14:38	669 Spruce St, Los Angeles, CA 90001	4	600.00	Los Angeles	14	Google Phone,Wired Headphones
4	176560	Wired Headphones	1	11.99	04/12/19 14:38	669 Spruce St, Los Angeles, CA 90001	4	11.99	Los Angeles	14	Google Phone,Wired Headphones
18	176574	Google Phone	1	600.00	04/03/19 19:42	20 Hill St, Los Angeles, CA 90001	4	600.00	Los Angeles	19	Google Phone,USB-C Charging Cable
19	176574	USB-C Charging Cable	1	11.95	04/03/19 19:42	20 Hill St, Los Angeles, CA 90001	4	11.95	Los Angeles	19	Google Phone,USB-C Charging Cable
30	176585	Bose SoundSport Headphones	1	99.99	04/07/19 11:31	823 Highland St, Boston, MA 02215	4	99.99	Boston	11	Bose SoundSport Headphones,Bose SoundSport Hea...

```
In [63]: #Drop duplicate order ID rows
df2 = df2.drop_duplicates(subset=['Order ID'])
df2.head()
```

Out[63]:

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address	month	sales	city	Hour	Grouped
3	176560	Google Phone	1	600.00	04/12/19 14:38	669 Spruce St, Los Angeles, CA 90001	4	600.00	Los Angeles	14	Google Phone,Wired Headphones
18	176574	Google Phone	1	600.00	04/03/19 19:42	20 Hill St, Los Angeles, CA 90001	4	600.00	Los Angeles	19	Google Phone,USB-C Charging Cable
30	176585	Bose SoundSport Headphones	1	99.99	04/07/19 11:31	823 Highland St, Boston, MA 02215	4	99.99	Boston	11	Bose SoundSport Headphones,Bose SoundSport Hea...
32	176586	AAA Batteries (4-pack)	2	2.99	04/10/19 17:00	365 Center St, San Francisco, CA 94016	4	5.98	San Francisco	17	AAA Batteries (4-pack),Google Phone
119	176672	Lightning Charging Cable	1	14.95	04/12/19 11:07	778 Maple St, New York City, NY 10001	4	14.95	New York City	11	Lightning Charging Cable,USB-C Charging Cable

```
In [64]: #Top 5 Products which are most often sold together
df2['Grouped'].value_counts()[0:5].plot.pie()
df2['Grouped'].value_counts()[0:5]
```

```
Out[64]: iPhone,Lightning Charging Cable      882
Google Phone,USB-C Charging Cable      856
iPhone,Wired Headphones      361
Vareebadd Phone,USB-C Charging Cable      312
Google Phone,Wired Headphones      303
Name: Grouped, dtype: int64
```

