Sales Anlaysis

In [1]:
```
import pandas as pd
import os
import matplotlib.pyplot as plt
```

Merging 12 months of sales data into a single file ¶

In [2]:
```
files = os.listdir("C:\\Users\\Admin\\Downloads\\all_months_data")
```

In [3]:
```
all_data = [i for i in files]
```

In [4]:
```
data = pd.DataFrame()
# for i in all_data:
#     df = pd.read_csv("C:\\Users\\Admin\\Downloads\\all_months_data\\"+i)
#     all_months_data =pd.concat([all_months_data, df])


for i in all_data:
    df = pd.read_csv("C:\\Users\\Admin\\Downloads\\all_months_data\\"+i)
    data = pd.concat([data,df])
```

In [5]:
```
data
```

Out[5]:

	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
...	...	...	...	...	...	...
11681	259353	AAA Batteries (4-pack)	3	2.99	09/17/19 20:56	840 Highland St, Los Angeles, CA 90001
11682	259354	iPhone	1	700	09/01/19 16:00	216 Dogwood St, San Francisco, CA 94016
11683	259355	iPhone	1	700	09/23/19 07:39	220 12th St, San Francisco, CA 94016
11684	259356	34in Ultrawide Monitor	1	379.99	09/19/19 17:30	511 Forest St, San Francisco, CA 94016
11685	259357	USB-C Charging Cable	1	11.95	09/30/19 00:18	250 Meadow St, San Francisco, CA 94016

186850 rows × 6 columns

In [6]:
```
all_data_csv = data.to_csv("all_data.csv", index = False)
```

In [7]:
```
all_data = pd.read_csv("all_data.csv")
```

In [8]:
```
all_data.head()
```

Out[8]:

	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 [9]:
```
all_data.tail()
```

Out[9]:

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address
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

clean up the data

In [10]:
```
all_data_nan = all_data[all_data.isnull()]
```

In [11]:
```
all_data_nan
```

Out[11]:

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address
0	NaN	NaN	NaN	NaN	NaN	NaN
1	NaN	NaN	NaN	NaN	NaN	NaN
2	NaN	NaN	NaN	NaN	NaN	NaN
3	NaN	NaN	NaN	NaN	NaN	NaN
4	NaN	NaN	NaN	NaN	NaN	NaN
...	...	...	...	...	...	...
186845	NaN	NaN	NaN	NaN	NaN	NaN
186846	NaN	NaN	NaN	NaN	NaN	NaN
186847	NaN	NaN	NaN	NaN	NaN	NaN
186848	NaN	NaN	NaN	NaN	NaN	NaN
186849	NaN	NaN	NaN	NaN	NaN	NaN

186850 rows × 6 columns

In [12]:
```
all_data = all_data.dropna(how = "all")
```

In [ ]:
```

```

In [13]:
```
temp_df = all_data[all_data["Order Date"].str[0:2] == "Or"]
```

In [ ]:
```

```

In [14]:
```
all_data = all_data[all_data["Order Date"].str[0:2] != "Or"]
```

In [ ]:
```

```

Convert columns to correct datatype

In [15]:
```
all_data["Quantity Ordered"] = pd.to_numeric(all_data["Quantity Ordered"])
all_data["Price Each"] = pd.to_numeric(all_data["Price Each"])  # convert
```

In [16]:
```
all_data.head()
```

Out[16]:

	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
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

In [ ]:
```

```

Augment the data with additional columns

Add month column

In [17]:
```
all_data["Month"] = all_data["Order Date"].str[0:2]
all_data["Month"] = all_data["Month"].astype("int32")
```

In [18]:
```
all_data.head()
```

Out[18]:

	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

In [ ]:
```

```

Add a Sales Column

In [19]:
```
all_data["Sales"] = all_data["Quantity Ordered"] * all_data["Price Each"]
```

In [20]:
```
all_data
```

Out[20]:

	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

In [ ]:
```
```

add a city column

In [21]:
```
all_data["City"] = all_data["Purchase Address"].apply(lambda x: x.split(",
```

In [22]:
```
all_data.head()
```

Out[22]:

	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 [ ]:

In [ ]:

In [ ]:

In [ ]:

which month was the best month for sales ? How much earned in that month

In [23]:
```
results = all_data.groupby("Month").sum()
```

```
C:\Users\Admin\AppData\Local\Temp\ipykernel_13268\3404397834.py:1: FutureWarning: The default value of numeric_only in DataFrameGroupBy.sum is deprecated. In a future version, numeric_only will default to False. Either specify numeric_only or select only columns which should be valid for the function.
  results = all_data.groupby("Month").sum()
```

In [24]:
```
results
```

Out[24]:

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

In [25]:
```
results.columns
```

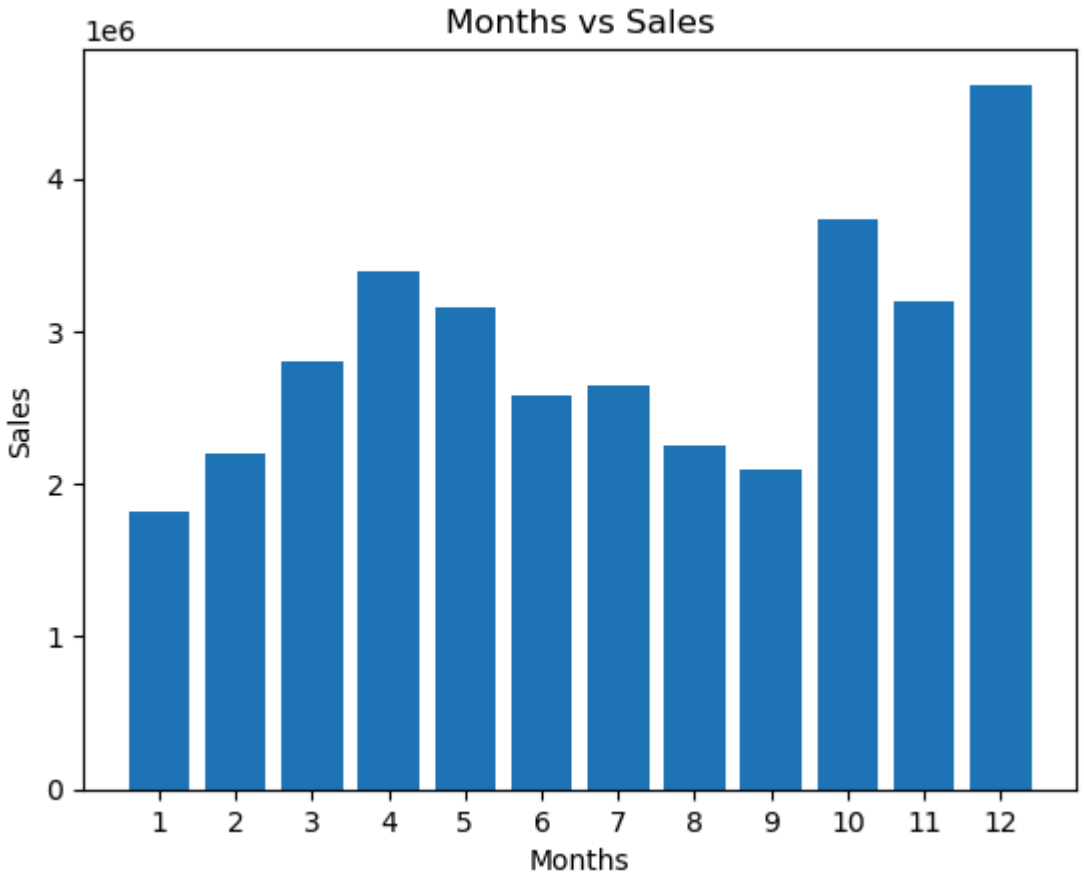
Out[25]:
```
Index(['Quantity Ordered', 'Price Each', 'Sales'], dtype='object')
```

In [26]:
```
list1 = []
for i in results["Sales"]:
    list1.append(i)
y = list1
```

In [27]:
```
months = range(1, 13)

plt.bar(months,y)
plt.xticks(months)

plt.xlabel("Months")
plt.ylabel("Sales")
plt.title("Months vs Sales")
plt.show()
```



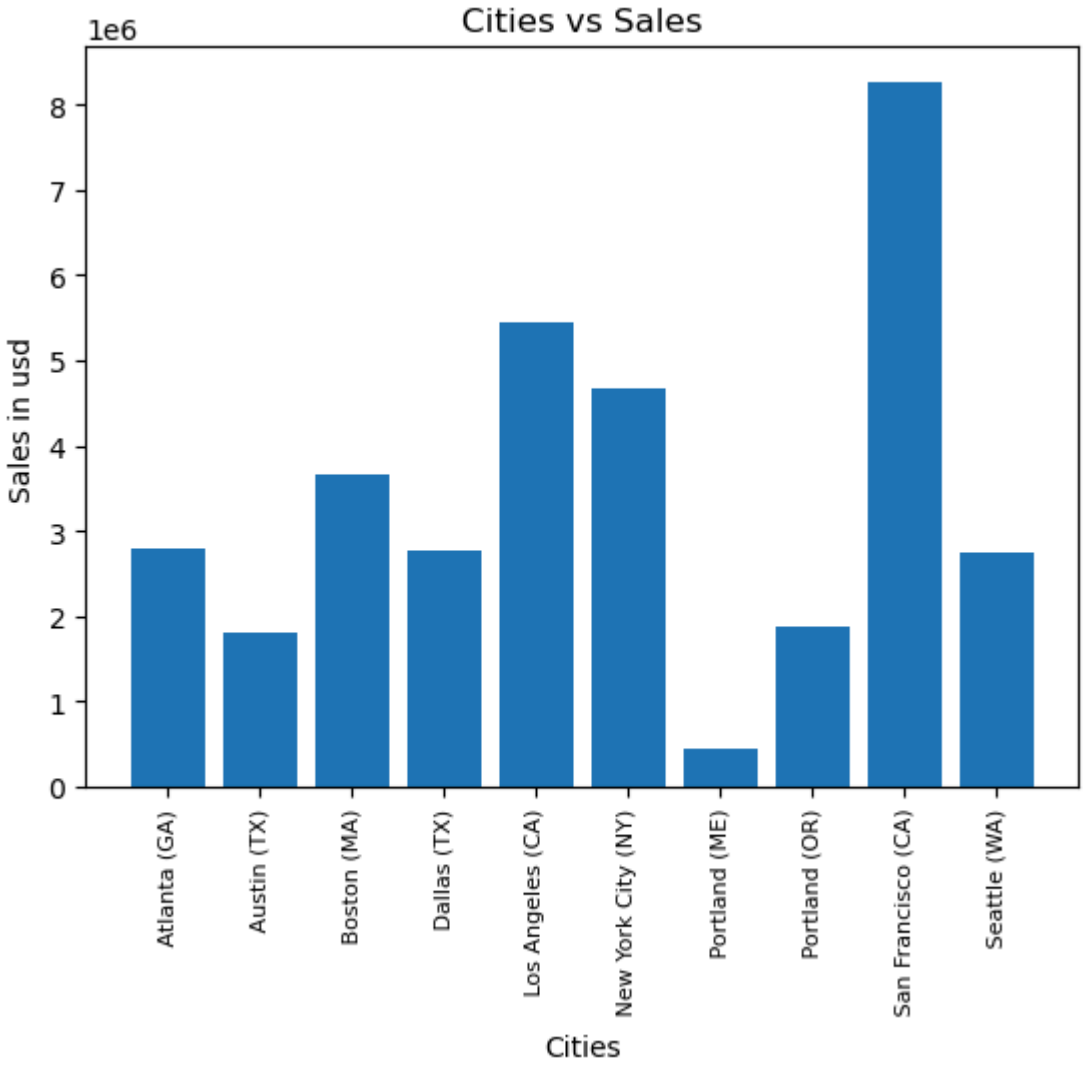
In [ ]:
```
```

which city had the highest no of sales

In [28]:
```
city_sales = all_data.groupby(["City"]).aggregate({"Sales": "sum"})
```

In [50]:
```
cities = [city for city, df in all_data.groupby("City")]
plt.bar(cities, all_data.groupby("City").sum()["Sales"])
plt.xticks(cities, rotation = "vertical", size = 8)
plt.xlabel("Cities")
plt.ylabel("Sales in usd")
plt.title("Cities vs Sales")
plt.show()
```

```
C:\Users\Admin\AppData\Local\Temp\ipykernel_13268\3625463538.py:2: FutureWarning: The default value of numeric_only in DataFrameGroupBy.sum is deprecated. In a future version, numeric_only will default to False. Either specify numeric_only or select only columns which should be valid for the function.
  plt.bar(cities, all_data.groupby("City").sum()["Sales"])
```



In [ ]:

What time we should display advertisements to maximize the likelihood the of customers buying product

In [30]:
```
all_data["Order Date"] = pd.to_datetime(all_data["Order Date"])
```

In [31]:
```
all_data["minute"] = all_data["Order Date"].dt.minute
```

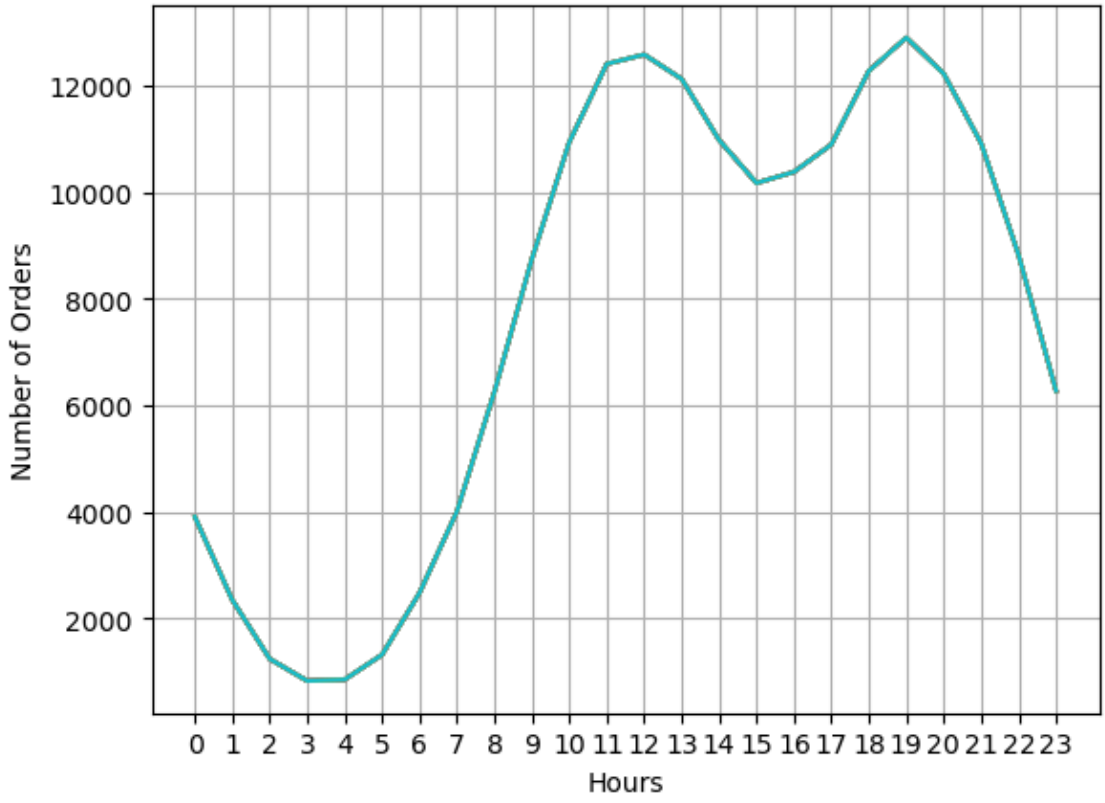
In [32]:
```
all_data["hour"] = all_data["Order Date"].dt.hour
```

In [33]:
```
all_data.head()
```

Out[33]:

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address	Month	Sales	City	minu
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)	
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)	
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)	
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)	
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)	

In [34]:
```
hours = [hour for hour,df in all_data.groupby("hour")]
plt.plot(hours, all_data.groupby(["hour"]).count())
plt.xticks(hours)
plt.grid()
plt.xlabel("Hours")
plt.ylabel("Number of Orders")
plt.show()
```



What products are most often sold together

In [35]:
```
df = all_data[all_data["Order ID"].duplicated(keep = False)]
```

In [36]:
```
df.head()
```

Out[36]:

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address	Month	Sales	City	mir
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)	
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)	
18	176574	Google Phone	1	600.00	2019-04-03 19:42:00	20 Hill St, Los Angeles, CA 90001	4	600.00	Los Angeles (CA)	
19	176574	USB-C Charging Cable	1	11.95	2019-04-03 19:42:00	20 Hill St, Los Angeles, CA 90001	4	11.95	Los Angeles (CA)	
30	176585	Bose SoundSport Headphones	1	99.99	2019-04-07 11:31:00	823 Highland St, Boston, MA 02215	4	99.99	Boston (MA)	

In [ ]:
```
```

In [37]:
```
df = all_data[all_data["Order ID"].duplicated(keep = False)]
df["Grouped"] = df.groupby("Order ID")["Product"].transform(lambda x: ",".
```

```
C:\Users\Admin\AppData\Local\Temp\ipykernel_13268\880847127.py:2: Setting
WithCopyWarning:
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-do
cs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (http
s://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returni
ng-a-view-versus-a-copy)
  df["Grouped"] = df.groupby("Order ID")["Product"].transform(lambda x:
",".join(x))
```

In [38]:
```
df.head()
```

Out[38]:

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address	Month	Sales	City	mir
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)	
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)	
18	176574	Google Phone	1	600.00	2019-04-03 19:42:00	20 Hill St, Los Angeles, CA 90001	4	600.00	Los Angeles (CA)	
19	176574	USB-C Charging Cable	1	11.95	2019-04-03 19:42:00	20 Hill St, Los Angeles, CA 90001	4	11.95	Los Angeles (CA)	
30	176585	Bose SoundSport Headphones	1	99.99	2019-04-07 11:31:00	823 Highland St, Boston, MA 02215	4	99.99	Boston (MA)	

In [39]:
```
df = df[["Order ID", "Grouped"]].drop_duplicates()
df
```

Out[39]:

	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 [40]:
```
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)))


print(count)
```

```
Counter({('iPhone', 'Lightning Charging Cable'): 1005, ('Google Phon
e', '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 Headphone
s'): 143, ('Lightning Charging Cable', 'Wired Headphones'): 92, ('Ligh
tning Charging Cable', 'Apple Airpods Headphones'): 81, ('Vareebadd Ph
one', 'Bose SoundSport Headphones'): 80, ('USB-C Charging Cable', 'Bos
e SoundSport Headphones'): 77, ('Apple Airpods Headphones', 'Wired Hea
dphones'): 69, ('Lightning Charging Cable', 'USB-C Charging Cable'): 5
8, ('Lightning Charging Cable', 'AA Batteries (4-pack)'): 55, ('Lightn
ing Charging Cable', 'Lightning Charging Cable'): 54, ('Bose SoundSpor
t Headphones', 'Wired Headphones'): 53, ('AA Batteries (4-pack)', 'Lig
htning Charging Cable'): 51, ('AAA Batteries (4-pack)', 'USB-C Chargin
g Cable'): 50, ('Apple Airpods Headphones', 'AAA Batteries (4-pack)'):
48, ('AA Batteries (4-pack)', 'AAA Batteries (4-pack)'): 48, ('USB-C C
harging Cable', 'USB-C Charging Cable'): 48, ('AAA Batteries (4-pac
k)', 'AAA Batteries (4-pack)'): 48, ('USB-C Charging Cable', 'AAA Batt
```

In [41]:
```
count.most_common(10)
```

Out[41]:
```
[(('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)]
```

In [42]:
```
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
```

In [ ]:
```
```

what product sold the most and why do you think it sold the most

In [43]:
```
all_data.head()
```

Out[43]:

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address	Month	Sales	City	minu
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)	
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)	
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)	
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)	
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)	

In [44]:
```
product_group = all_data.groupby("Product")
product_group.sum()
```

```
C:\Users\Admin\AppData\Local\Temp\ipykernel_13268\467729845.py:2: FutureW
arning: The default value of numeric_only in DataFrameGroupBy.sum is depr
ecated. In a future version, numeric_only will default to False. Either s
pecify numeric_only or select only columns which should be valid for the
function.
  product_group.sum()
```

Out[44]:

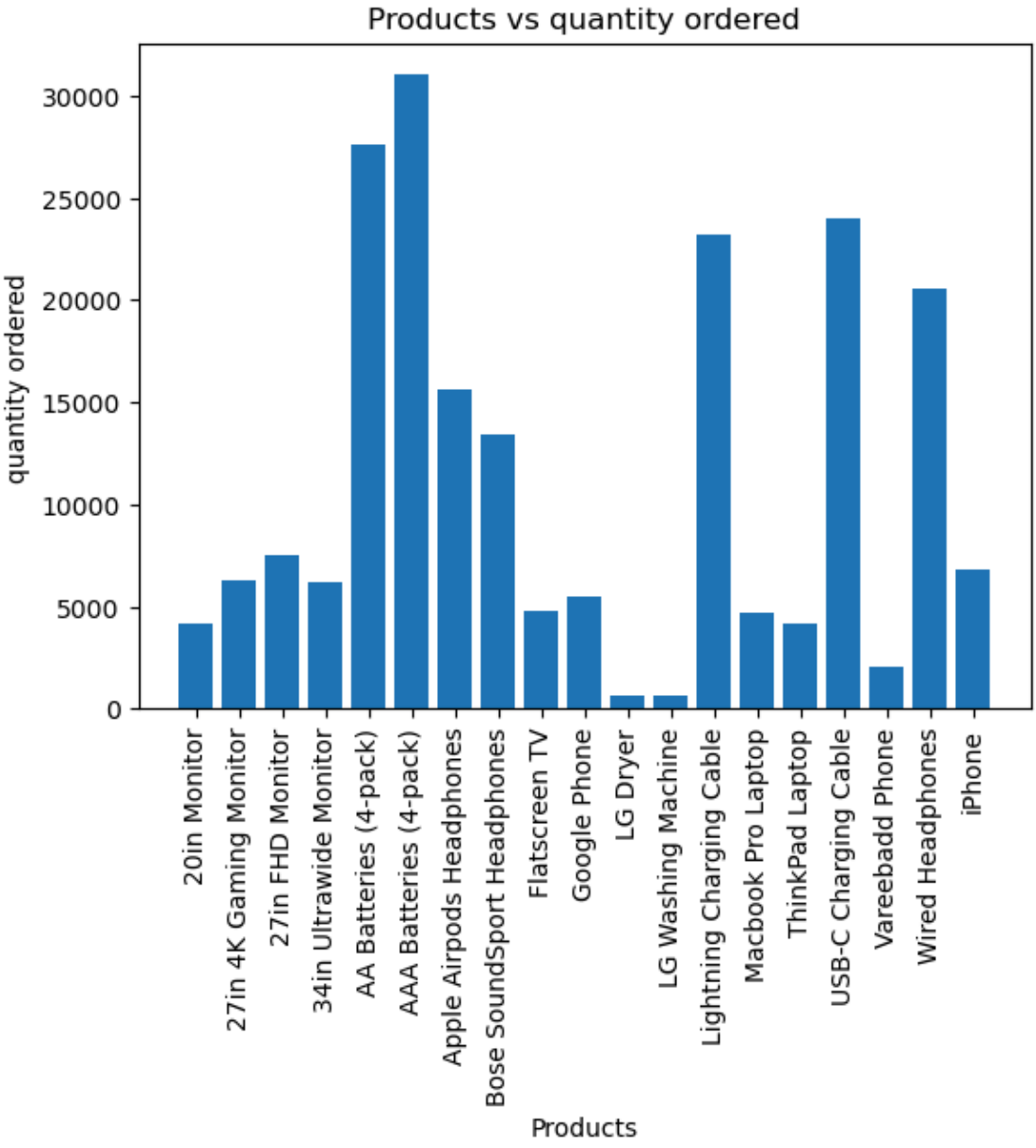
Product	Quantity Ordered	Price Each	Month	Sales	minute	hour
20in Monitor	4129	451068.99	29336	454148.71	122252	58764
27in 4K Gaming Monitor	6244	2429637.70	44440	2435097.56	184331	90916
27in FHD Monitor	7550	1125974.93	52558	1132424.50	219948	107540
34in Ultrawide Monitor	6199	2348718.19	43304	2355558.01	183480	89076
AA Batteries (4-pack)	27635	79015.68	145558	106118.40	609039	298342
AAA Batteries (4-pack)	31017	61716.59	146370	92740.83	612113	297332
Apple Airpods Headphones	15661	2332350.00	109477	2349150.00	455570	223304
Bose SoundSport Headphones	13457	1332366.75	94113	1345565.43	392603	192445
Flatscreen TV	4819	1440000.00	34224	1445700.00	142789	68815
Google Phone	5532	3315000.00	38305	3319200.00	162773	79479
LG Dryer	646	387600.00	4383	387600.00	19043	9326
LG Washing Machine	666	399600.00	4523	399600.00	19462	9785
Lightning Charging Cable	23217	323787.10	153092	347094.15	634442	312529
Macbook Pro Laptop	4728	8030800.00	33548	8037600.00	137574	68261
ThinkPad Laptop	4130	4127958.72	28950	4129958.70	121508	59746
USB-C Charging Cable	23975	261740.85	154819	286501.25	647586	314645
Vareebadd Phone	2068	826000.00	14309	827200.00	61835	29472
Wired Headphones	20557	226395.18	133397	246478.43	554023	271720
iPhone	6849	4789400.00	47941	4794300.00	201688	98657

In [45]:
```
quantity_ordered = product_group.sum()["Quantity Ordered"]
```

```
C:\Users\Admin\AppData\Local\Temp\ipykernel_13268\247680598.py:1: FutureW
arning: The default value of numeric_only in DataFrameGroupBy.sum is depr
ecated. In a future version, numeric_only will default to False. Either s
pecify numeric_only or select only columns which should be valid for the
function.
  quantity_ordered = product_group.sum()["Quantity Ordered"]
```

In [46]:
```
products = [product for product, df in product_group]
```

In [47]:
```
plt.bar(products, quantity_ordered)
plt.xticks(products, rotation = "vertical")
plt.xlabel("Products")
plt.ylabel("quantity ordered")
plt.title("Products vs quantity ordered")
plt.show()
```

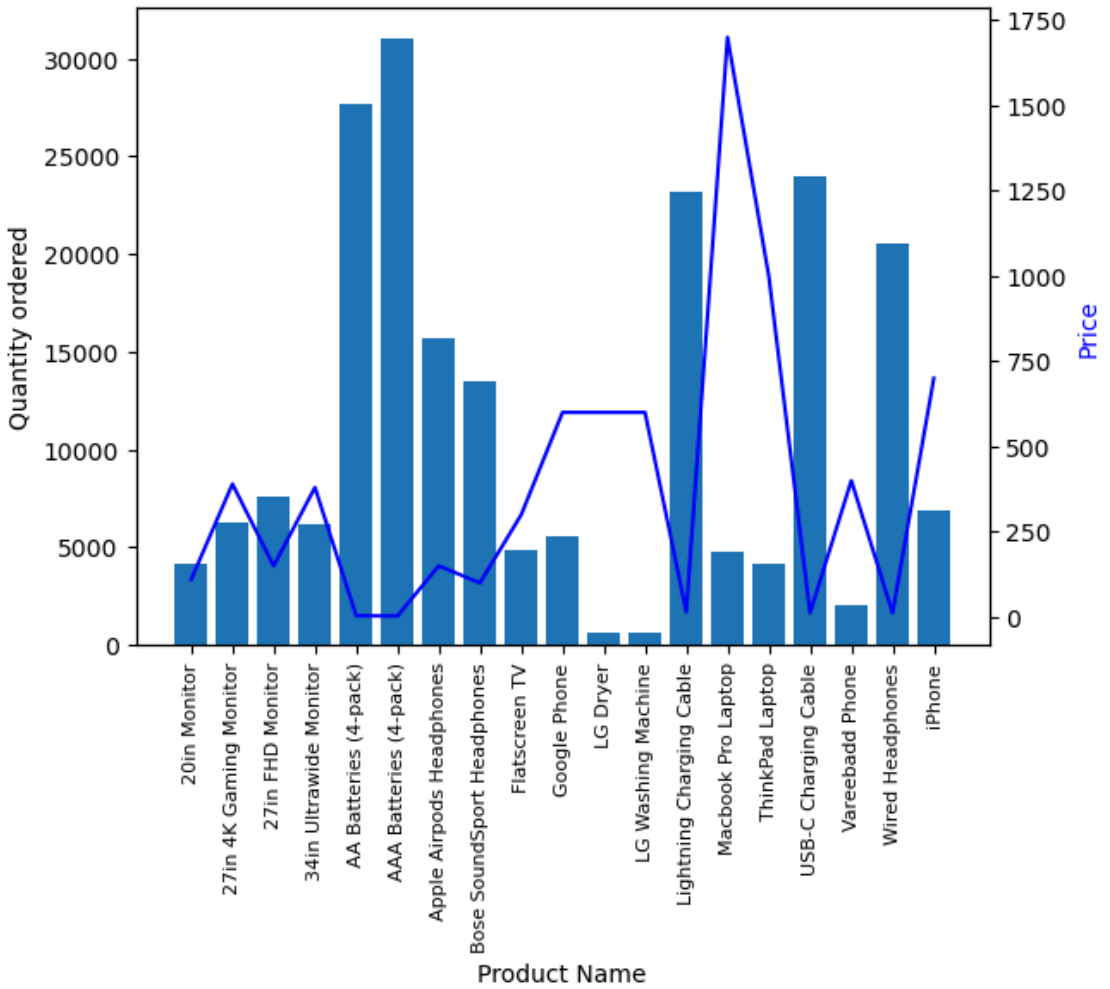


In [57]:

```
prices = all_data.groupby("Product").mean()["Price Each"]
fig, ax1 = plt.subplots()

ax2 = ax1.twinx()
ax1.bar(products, quantity_ordered)
ax2.plot(products, prices, "b-")
ax1.set_xlabel("Product Name")
ax1.set_ylabel("Quantity ordered")
ax2.set_ylabel("Price", color = "b")
ax1.set_xticklabels(products, rotation = "vertical", size = 8)
plt.show()
```

```
C:\Users\Admin\AppData\Local\Temp\ipykernel_13268\2513193907.py:1: Future
Warning: The default value of numeric_only in DataFrameGroupBy.mean is de
precated. In a future version, numeric_only will default to False. Either
specify numeric_only or select only columns which should be valid for the
function.
  prices = all_data.groupby("Product").mean()["Price Each"]
C:\Users\Admin\AppData\Local\Temp\ipykernel_13268\2513193907.py:10: UserW
arning: FixedFormatter should only be used together with FixedLocator
  ax1.set_xticklabels(products, rotation = "vertical", size = 8)
```



In [ ]:

In [ ]: