IMPORT LIBRARIES

In [2]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
sns.set(color_codes=True)
import datetime as dt
import os
import warnings
warnings.filterwarnings('ignore')
from mlxtend.frequent_patterns import apriori
!pip install mlxtend
from mlxtend.frequent_patterns import fpgrowth
from mlxtend.frequent_patterns import association_rules
```

```
Requirement already satisfied: mlxtend in c:\users\pc\anaconda3\lib\site-packages
(0.19.0)
Requirement already satisfied: matplotlib>=3.0.0 in c:\users\pc\anaconda3\lib\site-p
ackages (from mlxtend) (3.4.3)
Requirement already satisfied: pandas>=0.24.2 in c:\users\pc\anaconda3\lib\site-pack
ages (from mlxtend) (1.3.4)
Requirement already satisfied: scikit-learn>=0.20.3 in c:\users\pc\anaconda3\lib\sit
e-packages (from mlxtend) (0.24.2)
Requirement already satisfied: setuptools in c:\users\pc\anaconda3\lib\site-packages
(from mlxtend) (58.0.4)
Requirement already satisfied: scipy>=1.2.1 in c:\users\pc\anaconda3\lib\site-packag
es (from mlxtend) (1.7.1)
Requirement already satisfied: joblib>=0.13.2 in c:\users\pc\anaconda3\lib\site-pack
ages (from mlxtend) (1.1.0)
Requirement already satisfied: numpy>=1.16.2 in c:\users\pc\anaconda3\lib\site-packa
ges (from mlxtend) (1.20.3)
Requirement already satisfied: pyparsing>=2.2.1 in c:\users\pc\anaconda3\lib\site-pa
ckages (from matplotlib>=3.0.0->mlxtend) (3.0.4)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\pc\anaconda3\lib\sit
e-packages (from matplotlib>=3.0.0->mlxtend) (2.8.2)
Requirement already satisfied: pillow>=6.2.0 in c:\users\pc\anaconda3\lib\site-packa
ges (from matplotlib>=3.0.0->mlxtend) (8.4.0)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\pc\anaconda3\lib\site-p
ackages (from matplotlib>=3.0.0->mlxtend) (1.3.1)
Requirement already satisfied: cycler>=0.10 in c:\users\pc\anaconda3\lib\site-packag
es (from matplotlib>=3.0.0->mlxtend) (0.10.0)
Requirement already satisfied: six in c:\users\pc\anaconda3\lib\site-packages (from
cycler>=0.10->matplotlib>=3.0.0->mlxtend) (1.16.0)
Requirement already satisfied: pytz>=2017.3 in c:\users\pc\anaconda3\lib\site-packag
es (from pandas>=0.24.2->mlxtend) (2021.3)
Requirement already satisfied: threadpoolctl>=2.0.0 in c:\users\pc\anaconda3\lib\sit
e-packages (from scikit-learn>=0.20.3->mlxtend) (2.2.0)
```

Reviewing the Dataset

In [3]:
```
df = pd.read_csv(r'C:\Users\pc\Documents\shahad\python project\archive\all_data.csv'
df.head()
```

Out[3]:

Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address
0	236670	Wired Headphones	2	11.99	08/31/19 22:21	359 Spruce St, Seattle, WA 98101
1	236671	Bose SoundSport Headphones	1	99.99	08/15/19 15:11	492 Ridge St, Dallas, TX 75001
2	236672	iPhone	1	700.0	08/06/19 14:40	149 7th St, Portland, OR 97035
3	236673	AA Batteries (4-pack)	2	3.84	08/29/19 20:59	631 2nd St, Los Angeles, CA 90001
4	236674	AA Batteries (4-pack)	2	3.84	08/15/19 19:53	736 14th St, New York City, NY 10001

In [4]:
```
df.tail()
```

Out[4]:

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address
186845	319666	Lightning Charging Cable	1	14.95	12/11/19 20:58	14 Madison St, San Francisco, CA 94016
186846	319667	AA Batteries (4-pack)	2	3.84	12/01/19 12:01	549 Willow St, Los Angeles, CA 90001
186847	319668	Vareebadd Phone	1	400	12/09/19 06:43	273 Wilson St, Seattle, WA 98101
186848	319669	Wired Headphones	1	11.99	12/03/19 10:39	778 River St, Dallas, TX 75001
186849	319670	Bose SoundSport Headphones	1	99.99	12/21/19 21:45	747 Chestnut St, Los Angeles, CA 90001

Data Understanding

In [5]:
```
# data type each column
df.dtypes
```

Out[5]:
```
Order ID            object
Product             object
Quantity Ordered    object
Price Each          object
Order Date          object
Purchase Address    object
dtype: object
```

In [6]:
```
# Number of rows and columns
print(df.shape)
print(df.columns)
```

```
(186850, 6)
Index(['Order ID', 'Product', 'Quantity Ordered', 'Price Each', 'Order Date',
       'Purchase Address'],
      dtype='object')
```

In [7]:
```
# Number of unique values in each column
for col in df:
    print(col)
    print(df[col].unique())
    print('\n')
```

```
Order ID
['236670' '236671' '236672' ... '319668' '319669' '319670']


Product
['Wired Headphones' 'Bose SoundSport Headphones' 'iPhone'
 'AA Batteries (4-pack)' '34in Ultrawide Monitor' '20in Monitor'
 'Macbook Pro Laptop' 'LG Washing Machine' '27in FHD Monitor'
 'Lightning Charging Cable' 'Apple Airpods Headphones'
 'AAA Batteries (4-pack)' 'USB-C Charging Cable' '27in 4K Gaming Monitor'
 'ThinkPad Laptop' 'Flatscreen TV' 'Google Phone' 'Vareebadd Phone'
 'Product' nan 'LG Dryer']


Quantity Ordered
['2' '1' '3' '4' 'Quantity Ordered' nan '6' '7' '5' '8' '9']


Price Each
['11.99' '99.99' '700.0' '3.84' '379.99' '109.99' '1700.0' '600.0'
 '149.99' '14.95' '150.0' '2.99' '11.95' '389.99' '999.99' '300.0' '400.0'
 'Price Each' '600' '150' '1700' '300' '400' '700' nan]


Order Date
['08/31/19 22:21' '08/15/19 15:11' '08/06/19 14:40' ... '12/01/19 12:01'
 '12/09/19 06:43' '12/03/19 10:39']


Purchase Address
['359 Spruce St, Seattle, WA 98101' '492 Ridge St, Dallas, TX 75001'
 '149 7th St, Portland, OR 97035' ... '273 Wilson St, Seattle, WA 98101'
 '778 River St, Dallas, TX 75001' '747 Chestnut St, Los Angeles, CA 90001']
```

In [8]:
```
df.nunique()
```

Out[8]:
```
Order ID            178438
Product                 20
Quantity Ordered        10
Price Each              24
Order Date          142396
Purchase Address    140788
dtype: int64
```

In [9]:
```
# Number of missing values in each column
df.isnull().sum()
```

Out[9]:
```
Order ID            545
Product             545
Quantity Ordered    545
Price Each          545
Order Date          545
Purchase Address    545
dtype: int64
```

In [10]:
```
df.isnull()
```

Out[10]:

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

186850 rows × 6 columns

In [11]:
```
df.duplicated().sum()
```

Out[11]:
```
1162
```

Data Preparation

In [12]:
```
#convert columns to correct data types
df["Order ID"] = pd.to_numeric(df["Order ID"], errors='coerce').fillna(0, downcast='
df["Price Each"] = pd.to_numeric(df["Price Each"], errors='coerce').fillna(0, downca
df["Quantity Ordered"] = pd.to_numeric(df["Quantity Ordered"], errors='coerce').fill
df['Product']= df['Product'].convert_dtypes()
df['Purchase Address']= df['Purchase Address'].convert_dtypes()

df.dtypes
```

Out[12]:
```
Order ID             int64
Product             string
Quantity Ordered     int64
Price Each         float64
Order Date          object
Purchase Address    string
dtype: object
```

In [13]:
```
#drop identical duplicates rows
df.drop_duplicates(keep='first', inplace=True)
df
```

Out[13]:

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address
0	236670	Wired Headphones	2	11.99	08/31/19 22:21	359 Spruce St, Seattle, WA 98101

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address
1	236671	Bose SoundSport Headphones	1	99.99	08/15/19 15:11	492 Ridge St, Dallas, TX 75001
2	236672	iPhone	1	700.00	08/06/19 14:40	149 7th St, Portland, OR 97035
3	236673	AA Batteries (4-pack)	2	3.84	08/29/19 20:59	631 2nd St, Los Angeles, CA 90001
4	236674	AA Batteries (4-pack)	2	3.84	08/15/19 19:53	736 14th St, New York City, NY 10001
...	...	...	...	...	...	...
186845	319666	Lightning Charging Cable	1	14.95	12/11/19 20:58	14 Madison St, San Francisco, CA 94016
186846	319667	AA Batteries (4-pack)	2	3.84	12/01/19 12:01	549 Willow St, Los Angeles, CA 90001
186847	319668	Vareebadd Phone	1	400.00	12/09/19 06:43	273 Wilson St, Seattle, WA 98101
186848	319669	Wired Headphones	1	11.99	12/03/19 10:39	778 River St, Dallas, TX 75001
186849	319670	Bose SoundSport Headphones	1	99.99	12/21/19 21:45	747 Chestnut St, Los Angeles, CA 90001

185688 rows × 6 columns

In [14]:
```
df.duplicated().sum()
```

Out[14]:
```
0
```

In [15]:
```
df.duplicated().values.any()
```

Out[15]:
```
False
```

In [16]:
```
df.describe()
```

Out[16]:

	Order ID	Quantity Ordered	Price Each
count	185688.000000	185688.000000	185688.000000
mean	230408.894522	1.124531	184.517268
std	51516.989791	0.443082	332.842597
min	0.000000	0.000000	0.000000
25%	185831.750000	1.000000	11.950000
50%	230354.000000	1.000000	14.950000
75%	275028.250000	1.000000	150.000000
max	319670.000000	9.000000	1700.000000

In [17]:
```
# Impute Missing Values
df.fillna(df.mean(),inplace=True)
print(df.isnull().sum())
```

```
Order ID            0
Product             1
Quantity Ordered    0
Price Each          0
Order Date          1
Purchase Address    1
dtype: int64
```

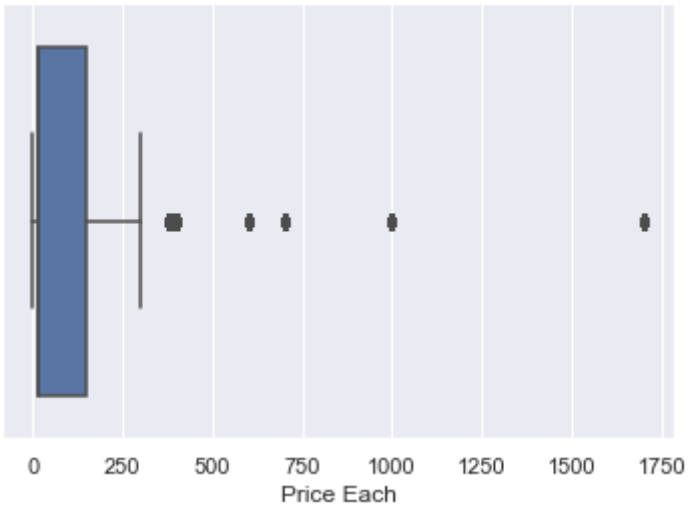
In [18]:
```
df['Product'].fillna('Nan ', inplace=True)
df['Order Date'].fillna('Nan', inplace=True)
df['Purchase Address'].fillna('Nan', inplace=True)
df.isnull().sum()
```

Out[18]:
```
Order ID            0
Product             0
Quantity Ordered    0
Price Each          0
Order Date          0
Purchase Address    0
dtype: int64
```

Data Exploratory Analysis

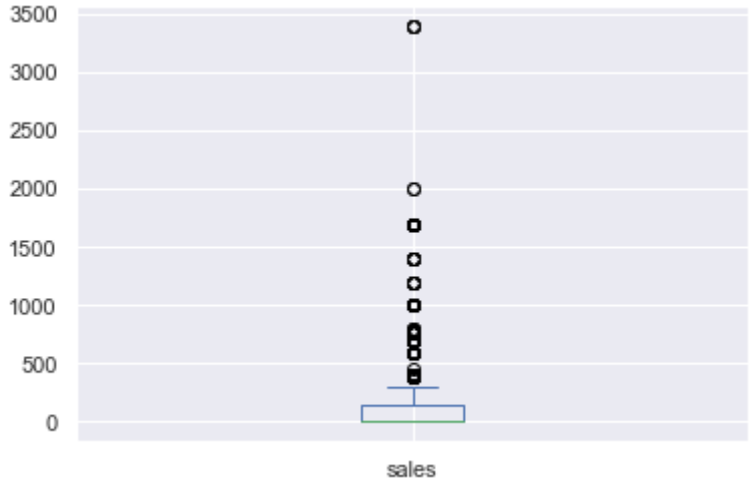
In [19]:
```
#df[['Price Each']].plot.box()
sns.boxplot(x=df['Price Each'])
```

Out[19]:
```
<AxesSubplot:xlabel='Price Each'>
```



In [22]:
```
df[['sales']].plot.box()
```

Out[22]:
```
<AxesSubplot:>
```



In [23]:

```
df['month']= df['Order Date'].str[0:2]
#df['month'] = df['month'].astype('int32')
df["month"] = pd.to_numeric(df["month"], errors='coerce').fillna(0, downcast='infer'
df.head(5)
# sales column
df['sales'] = df['Quantity Ordered'] * df['Price Each']
df.head(5)
## add city column

df['city']=df['Purchase Address'].apply(lambda x:x.split(',')[-1])

df.head()
```

Out[23]:

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address	month	sales	city
0	236670	Wired Headphones	2	11.99	08/31/19 22:21	359 Spruce St, Seattle, WA 98101	8	23.98	WA 98101
1	236671	Bose SoundSport Headphones	1	99.99	08/15/19 15:11	492 Ridge St, Dallas, TX 75001	8	99.99	TX 75001
2	236672	iPhone	1	700.00	08/06/19 14:40	149 7th St, Portland, OR 97035	8	700.00	OR 97035
3	236673	AA Batteries (4-pack)	2	3.84	08/29/19 20:59	631 2nd St, Los Angeles, CA 90001	8	7.68	CA 90001
4	236674	AA Batteries (4-pack)	2	3.84	08/15/19 19:53	736 14th St, New York City, NY 10001	8	7.68	NY 10001

In [24]:

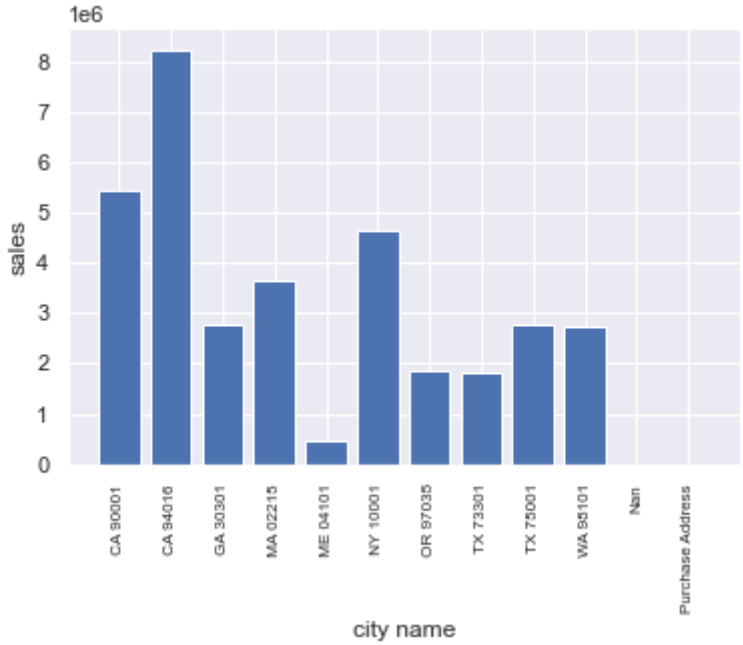
```
city_result = df.groupby('city').sum()
city_result
```

Out[24]:

city	Order ID	Quantity Ordered	Price Each	month	sales
CA 90001	6801454376	33247	5417171.70	208020	5448304.28
CA 94016	10287205330	50169	8204001.38	314949	8254743.55

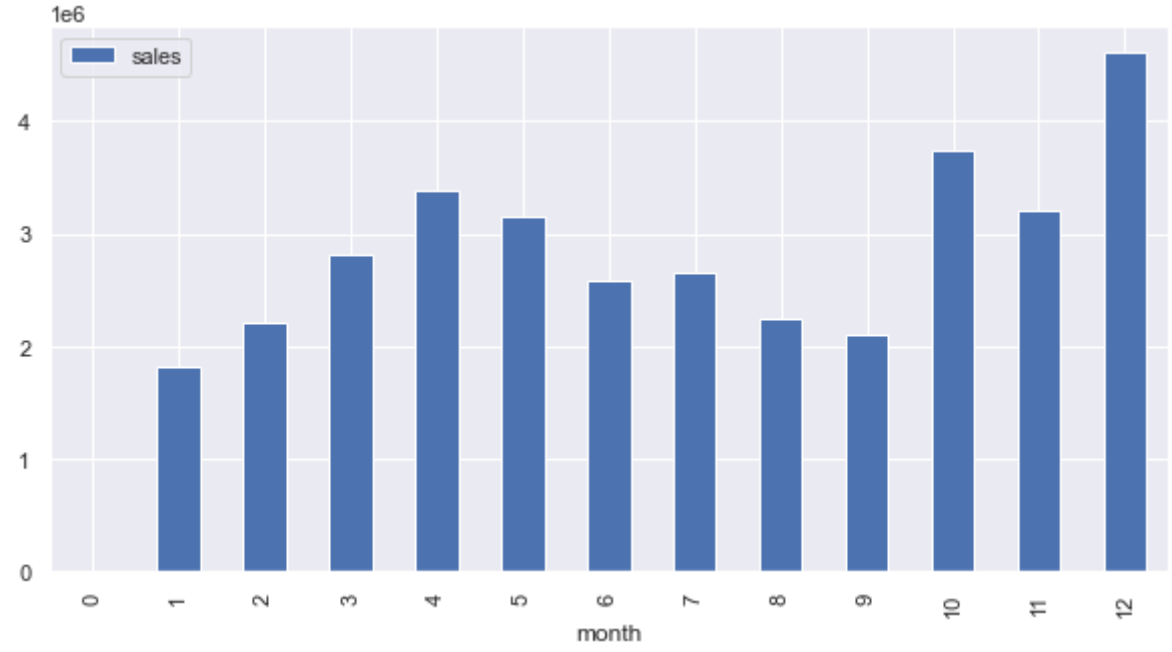
city	Order ID	Quantity Ordered	Price Each	month	sales
GA 30301	3419484721	16584	2778608.69	104649	2794199.07
MA 02215	4590961337	22494	3634398.40	140898	3658627.65
ME 04101	562382774	2746	446752.36	17119	449321.38
NY 10001	5729940452	27903	4632920.54	175557	4661867.14
OR 97035	2302544477	11291	1859836.44	70517	1870010.56
TX 73301	2277506240	11137	1808340.03	69720	1818044.33
TX 75001	3410206282	16707	2750026.38	104447	2765373.96
WA 98101	3402480817	16534	2730586.55	104817	2745046.02
Nan	0	0	0.00	0	0.00
Purchase Address	0	0	0.00	0	0.00

In [25]:
```
import matplotlib.pyplot as plt
cities = [city for city , df in df.groupby('city')]
plt.bar(cities,city_result['sales'])
plt.xticks(cities, rotation = 'vertical',size = '8')
plt.ylabel('sales')
plt.xlabel('city name')
plt.show()
```



In [26]:
```
monthlySales = df.groupby('month').agg({'sales' : 'sum'})
monthlySales.plot.bar(figsize=(10,5))
```

Out[26]: <AxesSubplot:xlabel='month'>



In [27]:
```
itemsMonthly = df.groupby(['Product','month']).agg({'Quantity Ordered' : 'sum', 'sal
itemsMonthly
```

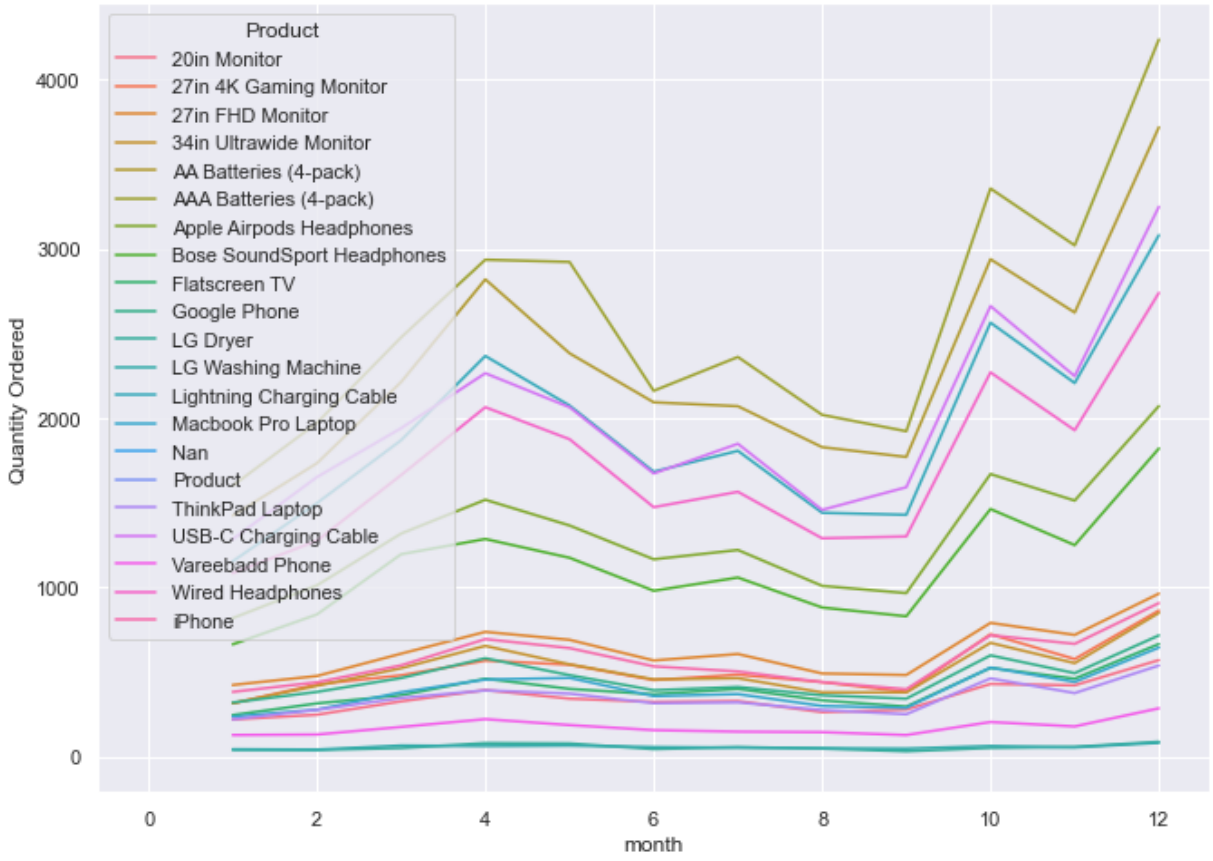
Out[27]:

	Product	month	Quantity Ordered	sales
0	20in Monitor	1	218	23977.82
1	20in Monitor	2	246	27057.54
2	20in Monitor	3	325	35746.75
3	20in Monitor	4	392	43116.08
4	20in Monitor	5	341	37506.59
...	...	...	...	...
225	iPhone	8	439	307300.00
226	iPhone	9	397	277900.00
227	iPhone	10	716	501200.00
228	iPhone	11	665	465500.00
229	iPhone	12	906	634200.00

230 rows × 4 columns

In [28]:
```
fig, ax = plt.subplots()
fig.set_size_inches(11,8)
sns.lineplot(data = itemsMonthly, x='month',y='Quantity Ordered', hue='Product',ax=a
```

Out[28]:
```
<AxesSubplot:xlabel='month', ylabel='Quantity Ordered'>
```



In [ ]:

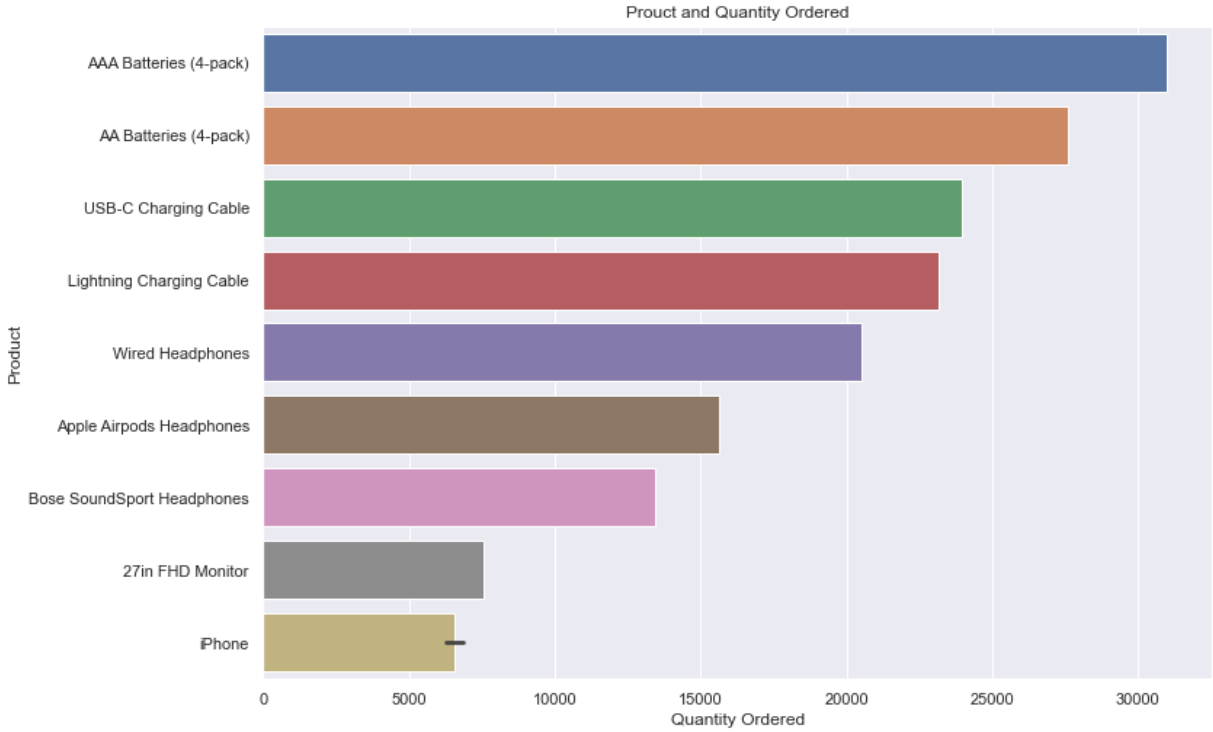
In [29]:
```
# The Most Quantity of prodec
top10=df.groupby('Product').agg({
'Quantity Ordered':'sum'
}).sort_values('Quantity Ordered',ascending=False)[:10]
top10
```

Out[29]:

Product	Quantity Ordered
AAA Batteries (4-pack)	30986
AA Batteries (4-pack)	27615
USB-C Charging Cable	23931
Lightning Charging Cable	23169
Wired Headphones	20524
Apple Airpods Headphones	15637
Bose SoundSport Headphones	13430
27in FHD Monitor	7541
iPhone	6847
27in 4K Gaming Monitor	6239

In [30]:
```
product=["AAA Batteries (4-pack)","AA Batteries (4-pack)","USB-C Charging Cable","Li
top10["Product"]=product
```

In [31]:
```
from matplotlib import pyplot
a4_dims = (11.7, 8.27)
fig, ax = pyplot.subplots(figsize=a4_dims)
sns.barplot(x = "Quantity Ordered",
 y = "Product",
 data = top10)
plt.title("Prouct and Quantity Ordered")
plt.show()
```



In [32]:
```
#what product are most often sold togethe
df3 = df[df['Order ID'].duplicated(keep=False)] ## check duplicates orderId
```

In [33]:
```
df3['grouped product'] = df.groupby('Order ID')['Product'].transform(lambda x: ','.j
df3 = df3[['Order ID','grouped product']].drop_duplicates()
df3.head(20)
```

Out[33]:

	Order ID	grouped product
46	236716	AA Batteries (4-pack),USB-C Charging Cable
60	236729	iPhone,Apple Airpods Headphones
62	236730	Google Phone,Wired Headphones
132	236799	Google Phone,Wired Headphones
163	236829	Vareebadd Phone,USB-C Charging Cable
211	0	Product,Nan
221	236885	27in 4K Gaming Monitor,USB-C Charging Cable
331	236992	Google Phone,USB-C Charging Cable
353	237013	iPhone,Lightning Charging Cable
386	237045	Bose SoundSport Headphones,Wired Headphones
405	237063	LG Dryer,AA Batteries (4-pack)

	Order ID	grouped product
410	237067	iPhone,Lightning Charging Cable
492	237148	iPhone,Lightning Charging Cable
563	237218	iPhone,Lightning Charging Cable
578	237232	Bose SoundSport Headphones,AAA Batteries (4-pack)
592	237245	AAA Batteries (4-pack),AA Batteries (4-pack)
599	237251	iPhone,Wired Headphones
610	237261	USB-C Charging Cable,Apple Airpods Headphones
617	237267	iPhone,Wired Headphones
640	237288	Google Phone,Wired Headphones

Apriori Algorithms

In [34]:
```
transactions_str = df.groupby(['Order ID', 'Product'])['Product'].count().reset_inde
transactions_str
```

Out[34]:

	Order ID	Product	Count
0	0	Nan	1
1	0	Product	1
2	141234	iPhone	1
3	141235	Lightning Charging Cable	1
4	141236	Wired Headphones	1
...	...	...	...
185636	319666	Lightning Charging Cable	1
185637	319667	AA Batteries (4-pack)	1
185638	319668	Vareebadd Phone	1
185639	319669	Wired Headphones	1
185640	319670	Bose SoundSport Headphones	1

185641 rows × 3 columns

In [35]:
```
TranEncod= transactions_str.pivot_table(index='Order ID', columns='Product', values=

TranEncod.head()
```

Out[35]:

Product	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	Fla
Order ID									

Product Order ID	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	Fla
0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
141234	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
141235	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
141236	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
141237	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	

5 rows × 21 columns

In [36]:
```
def encode(x):
    if x<=0:
        return 0
    if x>=1:
        return 1

# applying the function to the dataset

Tran_sets = TranEncod.applymap(encode)
Tran_sets.head()
```

Out[36]:

Product Order ID	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	Fla
0	0	0	0	0	0	0	0	0	
141234	0	0	0	0	0	0	0	0	
141235	0	0	0	0	0	0	0	0	
141236	0	0	0	0	0	0	0	0	
141237	0	0	1	0	0	0	0	0	

5 rows × 21 columns

In [37]:
```
frequent_items = apriori(Tran_sets, min_support = 0.001,use_colnames = True)
frequent_items
```

Out[37]:

	support	itemsets
0	0.022966	(20in Monitor)
1	0.034886	(27in 4K Gaming Monitor)
2	0.042020	(27in FHD Monitor)

	support	itemsets
3	0.034600	(34in Ultrawide Monitor)
4	0.115121	(AA Batteries (4-pack))
5	0.115407	(AAA Batteries (4-pack))
6	0.087005	(Apple Airpods Headphones)
7	0.074524	(Bose SoundSport Headphones)
8	0.026866	(Flatscreen TV)
9	0.030946	(Google Phone)
10	0.003620	(LG Dryer)
11	0.003732	(LG Washing Machine)
12	0.121073	(Lightning Charging Cable)
13	0.026457	(Macbook Pro Laptop)
14	0.023123	(ThinkPad Laptop)
15	0.122480	(USB-C Charging Cable)
16	0.011573	(Vareebadd Phone)
17	0.105622	(Wired Headphones)
18	0.038333	(iPhone)
19	0.002090	(iPhone, Apple Airpods Headphones)
20	0.001278	(Google Phone, Bose SoundSport Headphones)
21	0.005587	(Google Phone, USB-C Charging Cable)
22	0.002365	(Google Phone, Wired Headphones)
23	0.005666	(Lightning Charging Cable, iPhone)
24	0.002062	(Vareebadd Phone, USB-C Charging Cable)
25	0.001138	(Wired Headphones, USB-C Charging Cable)
26	0.002589	(Wired Headphones, iPhone)

In [38]:

```
frequent_itemsets=apriori(Tran_sets, min_support=0.001 , use_colnames=True )
frequent_itemsets['length']=frequent_itemsets['itemsets'].apply(lambda x:len(x))
frequent_itemsets
```

Out[38]:

	support	itemsets	length
0	0.022966	(20in Monitor)	1
1	0.034886	(27in 4K Gaming Monitor)	1
2	0.042020	(27in FHD Monitor)	1
3	0.034600	(34in Ultrawide Monitor)	1
4	0.115121	(AA Batteries (4-pack))	1
5	0.115407	(AAA Batteries (4-pack))	1
6	0.087005	(Apple Airpods Headphones)	1

	support	itemsets	length
7	0.074524	(Bose SoundSport Headphones)	1
8	0.026866	(Flatscreen TV)	1
9	0.030946	(Google Phone)	1
10	0.003620	(LG Dryer)	1
11	0.003732	(LG Washing Machine)	1
12	0.121073	(Lightning Charging Cable)	1
13	0.026457	(Macbook Pro Laptop)	1
14	0.023123	(ThinkPad Laptop)	1
15	0.122480	(USB-C Charging Cable)	1
16	0.011573	(Vareebadd Phone)	1
17	0.105622	(Wired Headphones)	1
18	0.038333	(iPhone)	1
19	0.002090	(iPhone, Apple Airpods Headphones)	2
20	0.001278	(Google Phone, Bose SoundSport Headphones)	2
21	0.005587	(Google Phone, USB-C Charging Cable)	2
22	0.002365	(Google Phone, Wired Headphones)	2
23	0.005666	(Lightning Charging Cable, iPhone)	2
24	0.002062	(Vareebadd Phone, USB-C Charging Cable)	2
25	0.001138	(Wired Headphones, USB-C Charging Cable)	2
26	0.002589	(Wired Headphones, iPhone)	2

In [39]:
```
frequent_itemsets[(frequent_itemsets['length']==2)&
(frequent_itemsets['support']>=0.001)]
```

Out[39]:

	support	itemsets	length
19	0.002090	(iPhone, Apple Airpods Headphones)	2
20	0.001278	(Google Phone, Bose SoundSport Headphones)	2
21	0.005587	(Google Phone, USB-C Charging Cable)	2
22	0.002365	(Google Phone, Wired Headphones)	2
23	0.005666	(Lightning Charging Cable, iPhone)	2
24	0.002062	(Vareebadd Phone, USB-C Charging Cable)	2
25	0.001138	(Wired Headphones, USB-C Charging Cable)	2
26	0.002589	(Wired Headphones, iPhone)	2

evaluation

In [40]:
```
from mlxtend.frequent_patterns import association_rules, apriori
```

```
rules = association_rules(frequent_items, metric = "lift", min_threshold = 0.001)
rules.sort_values('confidence', ascending = False, inplace = True)
rules
```

Out[40]:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage
4	(Google Phone)	(USB-C Charging Cable)	0.030946	0.122480	0.005587	0.180551	1.474128	0.001797
10	(Vareebadd Phone)	(USB-C Charging Cable)	0.011573	0.122480	0.002062	0.178208	1.455004	0.000645
9	(iPhone)	(Lightning Charging Cable)	0.038333	0.121073	0.005666	0.147807	1.220810	0.001025
6	(Google Phone)	(Wired Headphones)	0.030946	0.105622	0.002365	0.076422	0.723538	-0.000904
15	(iPhone)	(Wired Headphones)	0.038333	0.105622	0.002589	0.067544	0.639486	-0.001460
0	(iPhone)	(Apple Airpods Headphones)	0.038333	0.087005	0.002090	0.054532	0.626770	-0.001245
8	(Lightning Charging Cable)	(iPhone)	0.121073	0.038333	0.005666	0.046797	1.220810	0.001025
5	(USB-C Charging Cable)	(Google Phone)	0.122480	0.030946	0.005587	0.045619	1.474128	0.001797
2	(Google Phone)	(Bose SoundSport Headphones)	0.030946	0.074524	0.001278	0.041289	0.554038	-0.001029
14	(Wired Headphones)	(iPhone)	0.105622	0.038333	0.002589	0.024513	0.639486	-0.001460
1	(Apple Airpods Headphones)	(iPhone)	0.087005	0.038333	0.002090	0.024026	0.626770	-0.001245
7	(Wired Headphones)	(Google Phone)	0.105622	0.030946	0.002365	0.022391	0.723538	-0.000904
3	(Bose SoundSport Headphones)	(Google Phone)	0.074524	0.030946	0.001278	0.017145	0.554038	-0.001029
11	(USB-C Charging Cable)	(Vareebadd Phone)	0.122480	0.011573	0.002062	0.016838	1.455004	0.000645
12	(Wired Headphones)	(USB-C Charging Cable)	0.105622	0.122480	0.001138	0.010771	0.087941	-0.011799
13	(USB-C Charging Cable)	(Wired Headphones)	0.122480	0.105622	0.001138	0.009288	0.087941	-0.011799

In [41]:
```
#Number of rule
print(len(rules))
```

16

FPgrowth

In [42]:
```
frequent_itemsets_fp=fpgrowth(Tran_sets,min_support=0.001,use_colnames=True)
frequent_itemsets_fp
```

Out[42]:

	support	itemsets
0	0.038333	(iPhone)
1	0.121073	(Lightning Charging Cable)
2	0.105622	(Wired Headphones)
3	0.042020	(27in FHD Monitor)
4	0.115407	(AAA Batteries (4-pack))
5	0.034886	(27in 4K Gaming Monitor)
6	0.122480	(USB-C Charging Cable)
7	0.074524	(Bose SoundSport Headphones)
8	0.087005	(Apple Airpods Headphones)
9	0.026457	(Macbook Pro Laptop)
10	0.026866	(Flatscreen TV)
11	0.011573	(Vareebadd Phone)
12	0.115121	(AA Batteries (4-pack))
13	0.030946	(Google Phone)
14	0.022966	(20in Monitor)
15	0.034600	(34in Ultrawide Monitor)
16	0.023123	(ThinkPad Laptop)
17	0.003620	(LG Dryer)
18	0.003732	(LG Washing Machine)
19	0.002090	(iPhone, Apple Airpods Headphones)
20	0.002589	(Wired Headphones, iPhone)
21	0.005666	(Lightning Charging Cable, iPhone)
22	0.001138	(Wired Headphones, USB-C Charging Cable)
23	0.002062	(Vareebadd Phone, USB-C Charging Cable)
24	0.005587	(Google Phone, USB-C Charging Cable)
25	0.001278	(Google Phone, Bose SoundSport Headphones)
26	0.002365	(Google Phone, Wired Headphones)

evaluation

In [43]:
```
rules_fb = association_rules(frequent_itemsets_fp, metric = "confidence", min_thresh
rules_fb
```

Out[43]:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage
0	(iPhone)	(Apple Airpods Headphones)	0.038333	0.087005	0.002090	0.054532	0.626770	-0.001245
1	(Apple Airpods Headphones)	(iPhone)	0.087005	0.038333	0.002090	0.024026	0.626770	-0.001245
2	(Wired Headphones)	(iPhone)	0.105622	0.038333	0.002589	0.024513	0.639486	-0.001460
3	(iPhone)	(Wired Headphones)	0.038333	0.105622	0.002589	0.067544	0.639486	-0.001460
4	(Lightning Charging Cable)	(iPhone)	0.121073	0.038333	0.005666	0.046797	1.220810	0.001025
5	(iPhone)	(Lightning Charging Cable)	0.038333	0.121073	0.005666	0.147807	1.220810	0.001025
6	(Wired Headphones)	(USB-C Charging Cable)	0.105622	0.122480	0.001138	0.010771	0.087941	-0.011799
7	(USB-C Charging Cable)	(Wired Headphones)	0.122480	0.105622	0.001138	0.009288	0.087941	-0.011799
8	(Vareebadd Phone)	(USB-C Charging Cable)	0.011573	0.122480	0.002062	0.178208	1.455004	0.000645
9	(USB-C Charging Cable)	(Vareebadd Phone)	0.122480	0.011573	0.002062	0.016838	1.455004	0.000645
10	(Google Phone)	(USB-C Charging Cable)	0.030946	0.122480	0.005587	0.180551	1.474128	0.001797
11	(USB-C Charging Cable)	(Google Phone)	0.122480	0.030946	0.005587	0.045619	1.474128	0.001797
12	(Google Phone)	(Bose SoundSport Headphones)	0.030946	0.074524	0.001278	0.041289	0.554038	-0.001029
13	(Bose SoundSport Headphones)	(Google Phone)	0.074524	0.030946	0.001278	0.017145	0.554038	-0.001029
14	(Google Phone)	(Wired Headphones)	0.030946	0.105622	0.002365	0.076422	0.723538	-0.000904

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage
15	(Wired Headphones)	(Google Phone)	0.105622	0.030946	0.002365	0.022391	0.723538	-0.000904