Assignment 2

Name : Ajay Aravindh R

Roll No : 73771921103

Question 1: What was the best month for sales? How much was earned that month?

```
results = all_data.groupby('Month').sum()
results
# It only sums up the columns with numeric datatype!!
```

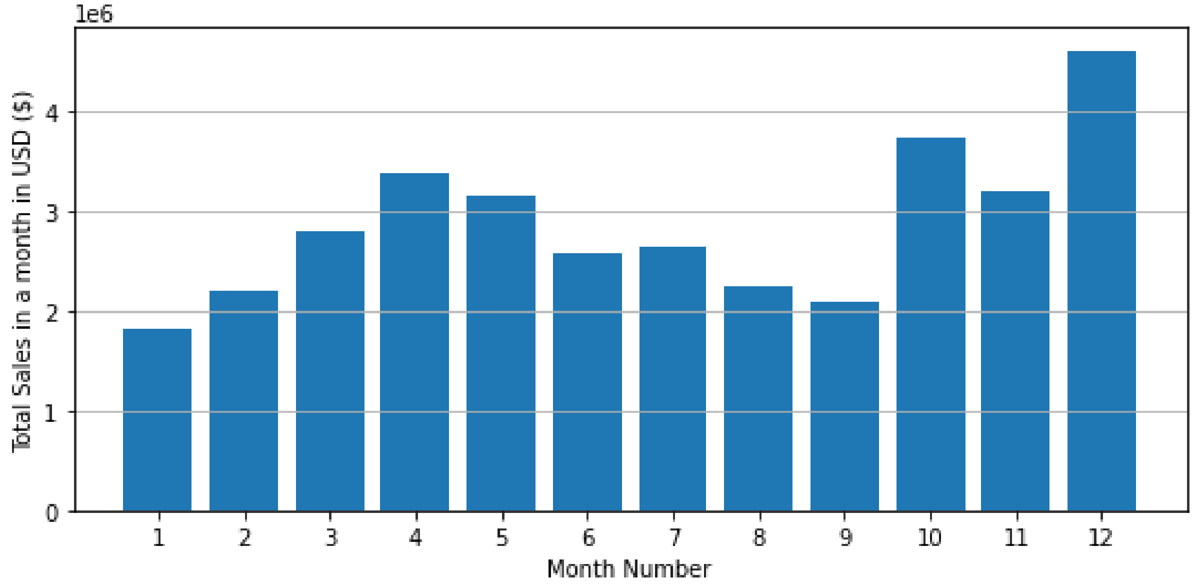
	Quantity Ordered	Price Each	Sales
Month			
1	10903	1.811768e+06	1.822257e+06
2	13449	2.188885e+06	2.202022e+06
3	17005	2.791208e+06	2.807100e+06
4	20558	3.367671e+06	3.390670e+06
5	18667	3.135125e+06	3.152607e+06
6	15253	2.562026e+06	2.577802e+06
7	16072	2.632540e+06	2.647776e+06
8	13448	2.230345e+06	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

	Quantity Ordered	Price Each	Sales
Month			
12	28114	4.588415e+06	4.613443e+06

Let's plot this!

```
import  matplotlib.pyplot as plt

months = range(1,13)
plt.figure(figsize=(9,4))
plt.bar(months,results['Sales'])
#plt.bar(results.index,results['Sales'])  <- Can use this too
plt.xticks(months)
plt.ylabel('Total Sales in a month in USD ($)')
plt.xlabel('Month Number')
plt.grid(axis='y')
plt.show()
```



```
print(f"Thus, the best month in sales was Month {results.idxmax()['Sales']}
with a sale of " + "${:,.2f}".format(results.max()['Sales']))
```

```
Thus, the best month in sales was Month 12 with a sale of $4,613,443.34
```

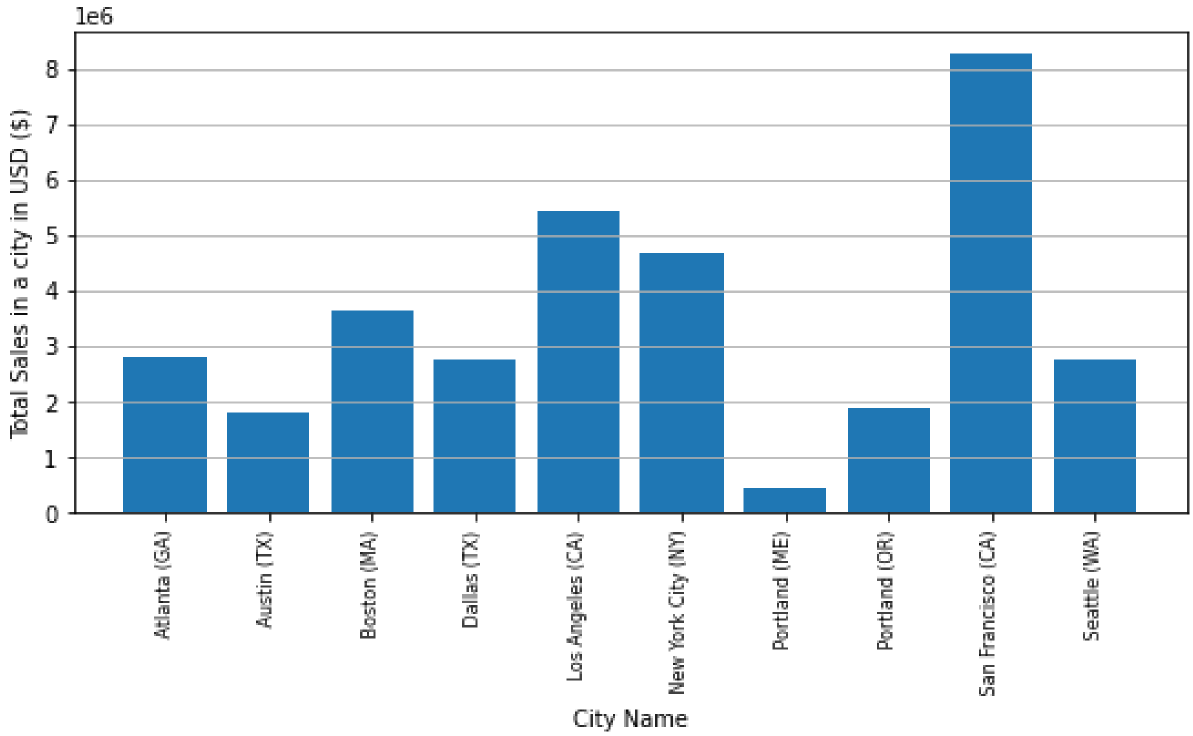
So, December being the best month in sales followed by October, April and November can be explained by the festivities that are in these months. Generaly in December, shopping spends peak around Christmas and New Year's and is closely followed by the festivities in the other months respectively.

Question 2: What city has the highest sales?

```
results = all_data.groupby('City').sum()
results
```

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

```
cities = results.index
plt.figure(figsize=(9,4))
plt.bar(cities,results['Sales'])
#plt.bar(results.index,results['Sales'])  <- Can use this too
plt.xticks(cities, rotation='vertical', size=8)
plt.ylabel('Total Sales in a city in USD ($)')
plt.xlabel('City Name')
plt.grid(axis='y')
plt.show()
```



Question 3: What time should we display advertisements to maximize likehood of customers buying products?

First converting Order Date column from str to datetime object.

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

```
all_data.head()
```

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

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

```
all_data['Hour'] = all_data['Order Date'].dt.hour
all_data['Minute'] = all_data['Order Date'].dt.minute
all_data.head()
```

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address	Month	Sales	City	Hour	Minute
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	46
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	30
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	38
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	38
5	176561	Wired Headphones	1	11.99	2019-04-30	333 8th St, Los Angeles,	4	11.99	Los Angeles (CA)	9	27

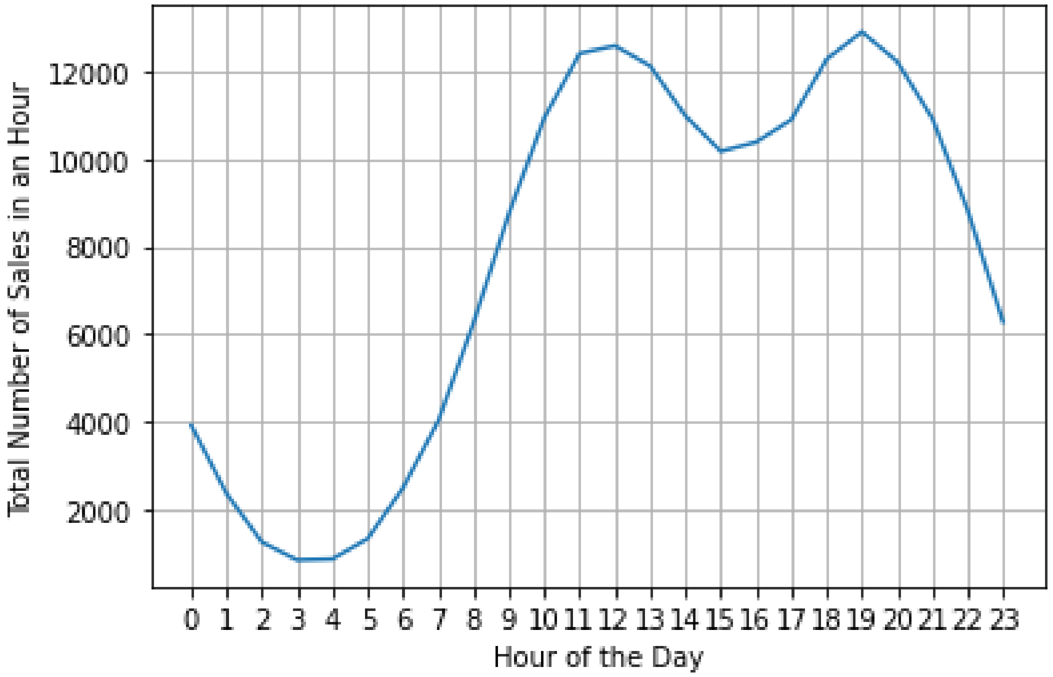
Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address	Month	Sales	City	Hour	Minute
				09:27:00	CA 90001					

```
results = all_data.groupby(['Hour']).count()
results
```

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address	Month	Sales	City	Minute
Hour										
0	3910	3910	3910	3910	3910	3910	3910	3910	3910	3910
1	2350	2350	2350	2350	2350	2350	2350	2350	2350	2350
2	1243	1243	1243	1243	1243	1243	1243	1243	1243	1243
3	831	831	831	831	831	831	831	831	831	831
4	854	854	854	854	854	854	854	854	854	854
5	1321	1321	1321	1321	1321	1321	1321	1321	1321	1321
6	2482	2482	2482	2482	2482	2482	2482	2482	2482	2482
7	4011	4011	4011	4011	4011	4011	4011	4011	4011	4011
8	6256	6256	6256	6256	6256	6256	6256	6256	6256	6256
9	8748	8748	8748	8748	8748	8748	8748	8748	8748	8748
10	10944	10944	10944	10944	10944	10944	10944	10944	10944	10944
11	12411	12411	12411	12411	12411	12411	12411	12411	12411	12411
12	12587	12587	12587	12587	12587	12587	12587	12587	12587	12587

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address	Month	Sales	City	Minute
Hour										
13	12129	12129	12129	12129	12129	12129	12129	12129	12129	12129
14	10984	10984	10984	10984	10984	10984	10984	10984	10984	10984
15	10175	10175	10175	10175	10175	10175	10175	10175	10175	10175
16	10384	10384	10384	10384	10384	10384	10384	10384	10384	10384
17	10899	10899	10899	10899	10899	10899	10899	10899	10899	10899
18	12280	12280	12280	12280	12280	12280	12280	12280	12280	12280
19	12905	12905	12905	12905	12905	12905	12905	12905	12905	12905
20	12228	12228	12228	12228	12228	12228	12228	12228	12228	12228
21	10921	10921	10921	10921	10921	10921	10921	10921	10921	10921
22	8822	8822	8822	8822	8822	8822	8822	8822	8822	8822
23	6275	6275	6275	6275	6275	6275	6275	6275	6275	6275

```
hours = results.index
plt.plot(hours, results['Order ID'])
plt.xticks(hours)
plt.xlabel('Hour of the Day')
plt.ylabel('Total Number of Sales in an Hour')
plt.grid()
plt.show()
```



From the chart above, it is clearly evident that the peaks in shopping occur around 12pm (1200 hrs) and 7pm (1900 hrs) across the entire 10 US cities.

Question 4: What products are most often sold together?

```
all_data.head(10)
```

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address	Month	Sales	City	Hour	Minute
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	46
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	30
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	38

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address	Month	Sales	City	Hour	Minute
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	38
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	27
6	176562	USB-C Charging Cable	1	11.95	2019-04-29 13:03:00	381 Wilson St, San Francisco, CA 94016	4	11.95	San Francisco (CA)	13	3
7	176563	Bose SoundSport Headphones	1	99.99	2019-04-02 07:46:00	668 Center St, Seattle, WA 98101	4	99.99	Seattle (WA)	7	46
8	176564	USB-C Charging Cable	1	11.95	2019-04-12 10:58:00	790 Ridge St, Atlanta, GA 30301	4	11.95	Atlanta (GA)	10	58
9	176565	Macbook Pro Laptop	1	1700.00	2019-04-24 10:38:00	915 Willow St, San Francisco, CA 94016	4	1700.00	San Francisco (CA)	10	38
10	176566	Wired Headphones	1	11.99	2019-04-08 14:05:00	83 7th St, Boston, MA 02215	4	11.99	Boston (MA)	14	5

Here, by carefully observing the data, we can say that if the Order ID of two or more rows match, the corresponding Products were sold together.

```
# Keeping only the ones which have duplicated Order ID
df = all_data[all_data['Order ID'].duplicated(keep=False)]

# Joining all the Products with same Order ID by ',' and storing in
'Grouped' column
df['Grouped'] = df.groupby('Order ID')['Product'].transform(lambda x:
','.join(x))

# Take only the Order ID and Grouped columns and drop duplicates
df = df[['Order ID', 'Grouped']].drop_duplicates()
df.head(100)
```

```
/home/sinjoy/.local/lib/python3.6/site-packages/ipykernel_launcher.py:5: Se
ttingWithCopyWarning:
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: http://pandas.pydata.org/pandas-docs/
stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  """
```

	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
129	176681	Apple Airpods Headphones,ThinkPad Laptop
138	176689	Bose SoundSport Headphones,AAA Batteries (4-pack)
189	176739	34in Ultrawide Monitor,Google Phone
225	176774	Lightning Charging Cable,USB-C Charging Cable
233	176781	iPhone,Lightning Charging Cable
250	176797	Google Phone,Bose SoundSport Headphones,Wired ...
260	176805	Google Phone,USB-C Charging Cable

	Order ID	Grouped
264	176808	Google Phone,Wired Headphones
270	176813	Google Phone,Wired Headphones
394	176935	AAA Batteries (4-pack),27in FHD Monitor
435	176975	USB-C Charging Cable,AAA Batteries (4-pack)
450	176989	Google Phone,USB-C Charging Cable
455	176993	iPhone,Wired Headphones
485	177022	iPhone,Wired Headphones
567	177102	iPhone,27in 4K Gaming Monitor
581	177115	iPhone,Lightning Charging Cable
584	177117	ThinkPad Laptop,AAA Batteries (4-pack)
635	177167	iPhone,Apple Airpods Headphones,AAA Batteries ...
648	177178	iPhone,Lightning Charging Cable
652	177181	Wired Headphones,Apple Airpods Headphones
654	177182	Macbook Pro Laptop,ThinkPad Laptop
657	177184	AA Batteries (4-pack),Flatscreen TV
672	177198	Vareebadd Phone,USB-C Charging Cable
676	177201	USB-C Charging Cable,Lightning Charging Cable
689	177213	iPhone,Lightning Charging Cable
...	...	...

	Order ID	Grouped
1695	178168	AA Batteries (4-pack),Macbook Pro Laptop
1718	178190	Vareebadd Phone,Wired Headphones
1750	178221	Flatscreen TV,Lightning Charging Cable
1772	178242	iPhone,Lightning Charging Cable
1789	178258	iPhone,Apple Airpods Headphones
1812	178280	iPhone,Lightning Charging Cable
1823	178290	Wired Headphones,ThinkPad Laptop
1938	178404	Bose SoundSport Headphones,USB-C Charging Cable
1947	178412	Vareebadd Phone,Wired Headphones
1968	178432	Bose SoundSport Headphones,USB-C Charging Cable
2061	178523	USB-C Charging Cable,20in Monitor
2064	178525	AAA Batteries (4-pack),AA Batteries (4-pack)
2067	178527	AA Batteries (4-pack),Bose SoundSport Headphones
2113	178572	USB-C Charging Cable,27in 4K Gaming Monitor
2292	178749	iPhone,Lightning Charging Cable
2325	178781	Google Phone,Wired Headphones
2370	178825	iPhone,Apple Airpods Headphones
2432	178886	Google Phone,Wired Headphones
2501	178954	Google Phone,USB-C Charging Cable

	Order ID	Grouped
2537	178989	Vareebadd Phone,USB-C Charging Cable
2539	178990	Google Phone,USB-C Charging Cable
2550	179000	iPhone,Lightning Charging Cable
2579	179028	27in 4K Gaming Monitor,Bose SoundSport Headphones
2609	179057	Google Phone,USB-C Charging Cable
2629	179076	Google Phone,USB-C Charging Cable
2662	179108	Lightning Charging Cable,AAA Batteries (4-pack)
2683	179128	iPhone,Apple Airpods Headphones
2718	179162	Google Phone,USB-C Charging Cable
2783	179226	34in Ultrawide Monitor,Macbook Pro Laptop
2829	179270	iPhone,Lightning Charging Cable

100 rows × 2 columns

Now we can count number of occurences of the combinations.

```
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
```

Voila! iPhone and Lightning Charging Cable are sold together the most!! 1005 times!!

For more than 2 items taken at a time

```
all_data.groupby('Order ID').count().sort_values(['Product'], axis=0,
ascending=False)
```

	Product	Quantity Ordered	Price Each	Order Date	Purchase Address	Month	Sales	City	Hour	Minute
Order ID										
160873	5	5	5	5	5	5	5	5	5	5
312462	4	4	4	4	4	4	4	4	4	4
242936	4	4	4	4	4	4	4	4	4	4
235798	4	4	4	4	4	4	4	4	4	4
165665	4	4	4	4	4	4	4	4	4	4
304802	4	4	4	4	4	4	4	4	4	4
196615	4	4	4	4	4	4	4	4	4	4
263918	4	4	4	4	4	4	4	4	4	4
226625	4	4	4	4	4	4	4	4	4	4
277875	4	4	4	4	4	4	4	4	4	4
289117	4	4	4	4	4	4	4	4	4	4
193511	4	4	4	4	4	4	4	4	4	4
296353	4	4	4	4	4	4	4	4	4	4

	Product	Quantity Ordered	Price Each	Order Date	Purchase Address	Month	Sales	City	Hour	Minute
Order ID										
178158	4	4	4	4	4	4	4	4	4	4
212334	4	4	4	4	4	4	4	4	4	4
194253	4	4	4	4	4	4	4	4	4	4
312407	4	4	4	4	4	4	4	4	4	4
295681	4	4	4	4	4	4	4	4	4	4
309098	3	3	3	3	3	3	3	3	3	3
195475	3	3	3	3	3	3	3	3	3	3
212774	3	3	3	3	3	3	3	3	3	3
181177	3	3	3	3	3	3	3	3	3	3
162189	3	3	3	3	3	3	3	3	3	3
290151	3	3	3	3	3	3	3	3	3	3
143243	3	3	3	3	3	3	3	3	3	3
186904	3	3	3	3	3	3	3	3	3	3
212753	3	3	3	3	3	3	3	3	3	3
183436	3	3	3	3	3	3	3	3	3	3
188097	3	3	3	3	3	3	3	3	3	3
205271	3	3	3	3	3	3	3	3	3	3

	Product	Quantity Ordered	Price Each	Order Date	Purchase Address	Month	Sales	City	Hour	Minute
Order ID										
...	...	...	...	...	...	...	...	...	...	...
201991	1	1	1	1	1	1	1	1	1	1
201992	1	1	1	1	1	1	1	1	1	1
201993	1	1	1	1	1	1	1	1	1	1
201994	1	1	1	1	1	1	1	1	1	1
201995	1	1	1	1	1	1	1	1	1	1
201996	1	1	1	1	1	1	1	1	1	1
201997	1	1	1	1	1	1	1	1	1	1
201998	1	1	1	1	1	1	1	1	1	1
201981	1	1	1	1	1	1	1	1	1	1
201978	1	1	1	1	1	1	1	1	1	1
201958	1	1	1	1	1	1	1	1	1	1
201977	1	1	1	1	1	1	1	1	1	1
201959	1	1	1	1	1	1	1	1	1	1
201960	1	1	1	1	1	1	1	1	1	1
201961	1	1	1	1	1	1	1	1	1	1
201963	1	1	1	1	1	1	1	1	1	1

	Product	Quantity Ordered	Price Each	Order Date	Purchase Address	Month	Sales	City	Hour	Minute
Order ID										
201964	1	1	1	1	1	1	1	1	1	1
201965	1	1	1	1	1	1	1	1	1	1
201966	1	1	1	1	1	1	1	1	1	1
201967	1	1	1	1	1	1	1	1	1	1
201968	1	1	1	1	1	1	1	1	1	1
201969	1	1	1	1	1	1	1	1	1	1
201970	1	1	1	1	1	1	1	1	1	1
201971	1	1	1	1	1	1	1	1	1	1
201972	1	1	1	1	1	1	1	1	1	1
201973	1	1	1	1	1	1	1	1	1	1
201974	1	1	1	1	1	1	1	1	1	1
201975	1	1	1	1	1	1	1	1	1	1
201976	1	1	1	1	1	1	1	1	1	1
319670	1	1	1	1	1	1	1	1	1	1

178437 rows × 10 columns

From the above table we can see that a particular Order ID occurs at most 5 times.

```
from itertools import combinations
from collections import Counter

for comb in range(2, 6):
    count = Counter()
    print(f"\nTaking {comb} items at a time:")
    for row in df['Grouped']:
        row_list = row.split(',')
        count.update(Counter(combinations(row_list, comb)))

    for key, value in count.most_common(10):
        print(key, value)
```

```
Taking 2 items at a time:
('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

Taking 3 items at a time:
('Google Phone', 'USB-C Charging Cable', 'Wired Headphones') 87
('iPhone', 'Lightning Charging Cable', 'Wired Headphones') 62
('iPhone', 'Lightning Charging Cable', 'Apple Airpods Headphones') 47
('Google Phone', 'USB-C Charging Cable', 'Bose SoundSport Headphones') 35
('Vareebadd Phone', 'USB-C Charging Cable', 'Wired Headphones') 33
('iPhone', 'Apple Airpods Headphones', 'Wired Headphones') 27
('Google Phone', 'Bose SoundSport Headphones', 'Wired Headphones') 24
('Vareebadd Phone', 'USB-C Charging Cable', 'Bose SoundSport Headphones') 1
6
('USB-C Charging Cable', 'Bose SoundSport Headphones', 'Wired Headphones')
5
('Vareebadd Phone', 'Bose SoundSport Headphones', 'Wired Headphones') 5

Taking 4 items at a time:
('iPhone', 'Lightning Charging Cable', 'Apple Airpods Headphones', 'Wired H
eadphones') 4
('Google Phone', 'USB-C Charging Cable', 'Bose SoundSport Headphones', 'Wir
ed Headphones') 3
('Vareebadd Phone', 'USB-C Charging Cable', 'Bose SoundSport Headphones', '
Wired Headphones') 2
('Google Phone', 'USB-C Charging Cable', 'Wired Headphones', 'USB-C Chargin
g Cable') 1
('iPhone', 'Lightning Charging Cable', 'Wired Headphones', 'AA Batteries (4
-pack)') 1
('Google Phone', 'USB-C Charging Cable', 'Bose SoundSport Headphones', '34i
n Ultrawide Monitor') 1
('Google Phone', 'USB-C Charging Cable', 'Wired Headphones', 'Apple Airpods
Headphones') 1
('iPhone', 'Lightning Charging Cable', 'Apple Airpods Headphones', 'Google
Phone') 1
('iPhone', 'Lightning Charging Cable', 'Wired Headphones', 'Google Phone')
1
('iPhone', 'Apple Airpods Headphones', 'Wired Headphones', 'Google Phone')
1

Taking 5 items at a time:
```

```
('iPhone', 'Lightning Charging Cable', 'Apple Airpods Headphones', 'Wired H
eadphones', 'Google Phone') 1
```

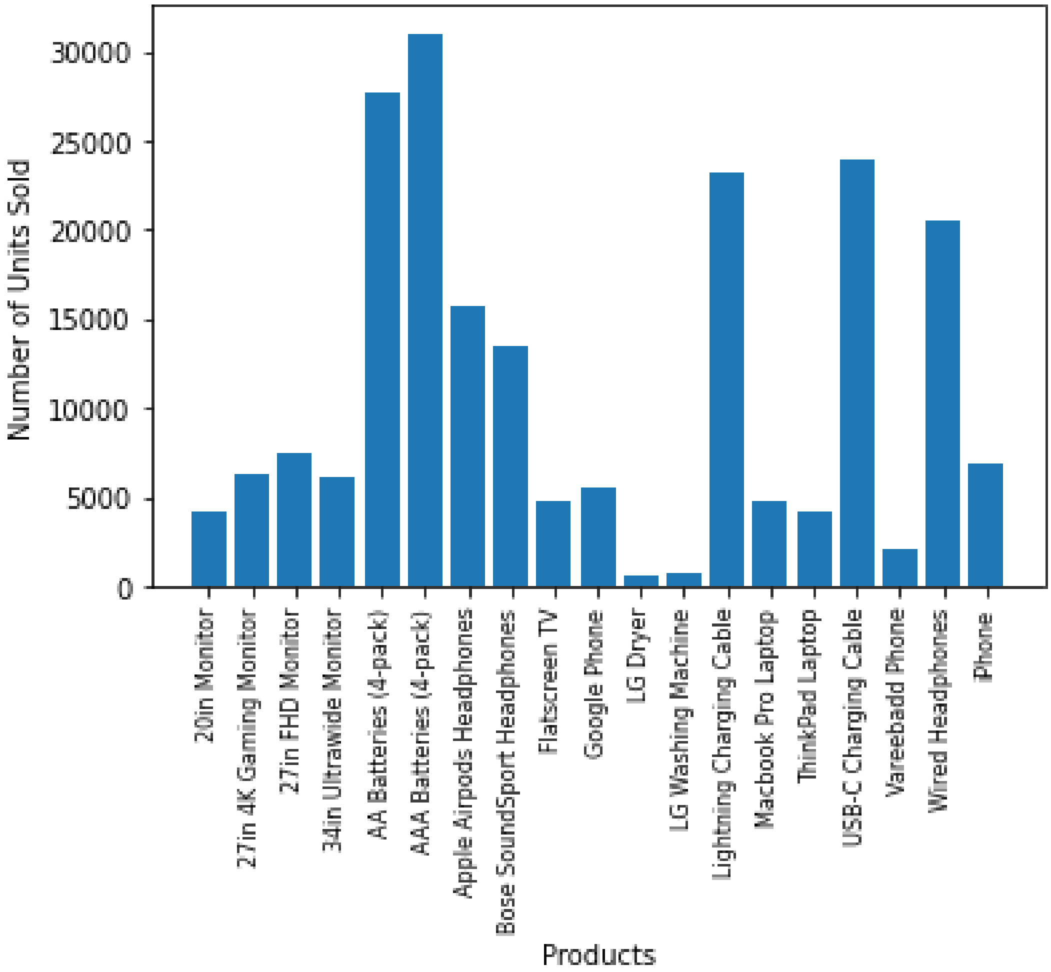
Question 5: What product sold the most? Why do you think it sold the most?

```
product_group = all_data.groupby('Product')
quantity_ordered = product_group.sum()['Quantity Ordered']
quantity_ordered
```

```
Product
20in Monitor                   4129
27in 4K Gaming Monitor         6244
27in FHD Monitor               7550
34in Ultrawide Monitor         6199
AA Batteries (4-pack)         27635
AAA Batteries (4-pack)        31017
Apple Airpods Headphones      15661
Bose SoundSport Headphones    13457
Flatscreen TV                  4819
Google Phone                   5532
LG Dryer                        646
LG Washing Machine              666
Lightning Charging Cable      23217
Macbook Pro Laptop             4728
ThinkPad Laptop                4130
USB-C Charging Cable          23975
Vareebadd Phone                2068
Wired Headphones              20557
iPhone                         6849
Name: Quantity Ordered, dtype: int64
```

```
products = quantity_ordered.index

plt.bar(products, quantity_ordered)
plt.xticks(products, rotation='vertical', size=8)
plt.ylabel('Number of Units Sold')
plt.xlabel('Products')
plt.show()
```



We can say that AAA Batteries (4-pack) were the most. This may be because the per unit price of this item is lowest. Let's see if we are correct!

```
prices = all_data.groupby('Product').mean()['Price Each']
prices
```

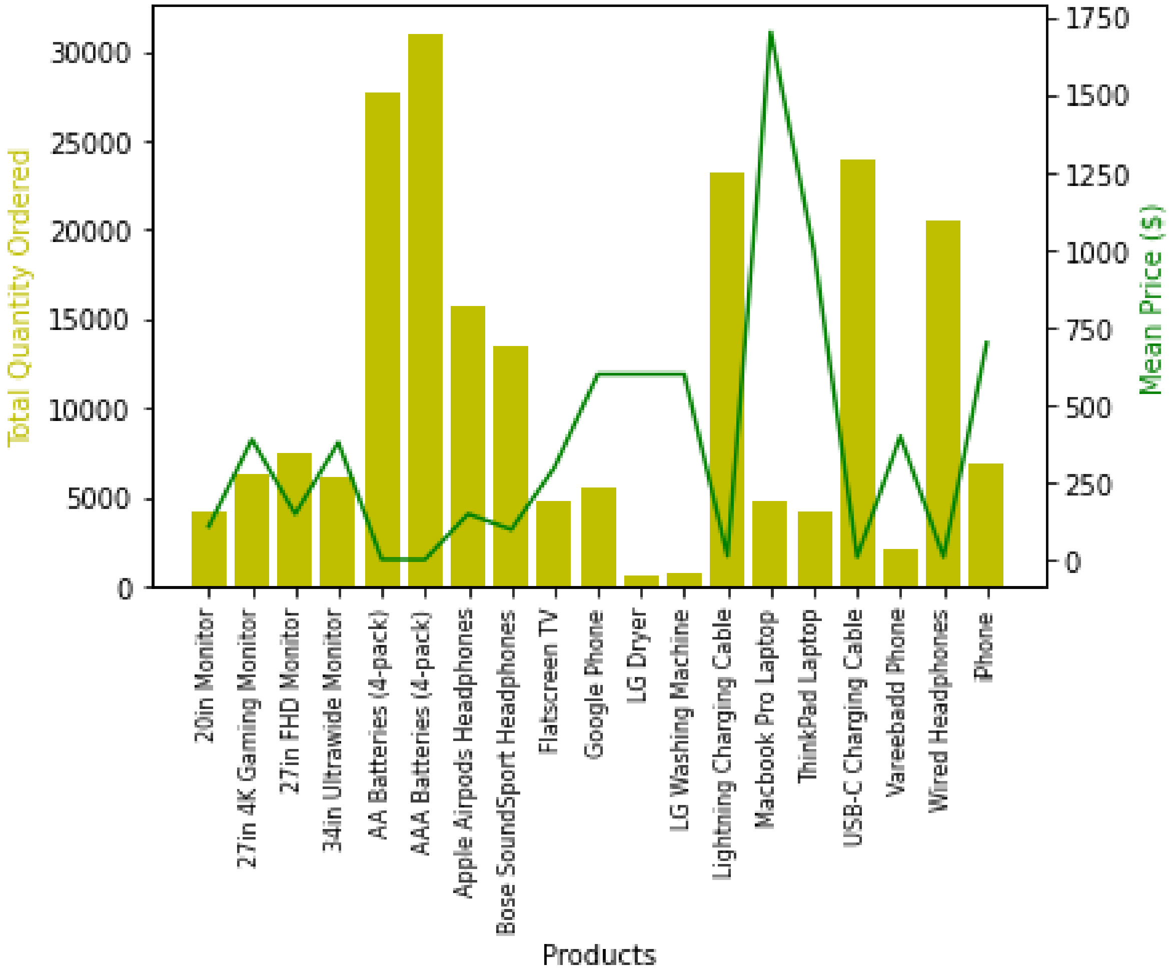
```
Product
20in Monitor                   109.99
27in 4K Gaming Monitor         389.99
27in FHD Monitor               149.99
34in Ultrawide Monitor         379.99
AA Batteries (4-pack)            3.84
AAA Batteries (4-pack)           2.99
Apple Airpods Headphones       150.00
Bose SoundSport Headphones      99.99
Flatscreen TV                  300.00
Google Phone                   600.00
LG Dryer                       600.00
LG Washing Machine             600.00
Lightning Charging Cable        14.95
Macbook Pro Laptop            1700.00
ThinkPad Laptop                999.99
USB-C Charging Cable            11.95
Vareebadd Phone                400.00
Wired Headphones                11.99
```

```
iPhone                         700.00
Name: Price Each, dtype: float64
```

```
fig, ax1 = plt.subplots()
ax2 = ax1.twinx()
ax1.bar(products, quantity_ordered, color='y')
ax1.set_xticklabels(products, rotation='vertical', size=8)
ax1.set_ylabel('Total Quantity Ordered', color='y')
ax1.set_xlabel('Products')

ax2.plot(products, prices, color='g')
ax2.set_ylabel('Mean Price ($)', color='g')
plt.show()
```

```
/home/sinjoy/.local/lib/python3.6/site-packages/ipykernel_launcher.py:4: Us
erWarning: FixedFormatter should only be used together with FixedLocator
  after removing the cwd from sys.path.
```



So, we see an inverse correlation in the Quantity Ordered and Mean Prices. There are some inconsistencies such as Macbook Pro Laptop as greater price than LG Dryer but still Quantity Ordered is more for Mackbook Pro Laptop than for LG Dryer. This may be because demand is more for Macbook Pro Laptop than for LG Dryer.