Sales Data Project

In [126…
```python
import pandas as pd
import os
```

In [127…
```python
files = [file for file in os.listdir(r"C:\Users\946851\OneDrive - Cognizant\Desktop\Pandas-Data-Science-Tasks-master\SalesAnalysis\Sales_Data")]
all_months_data = pd.DataFrame()
for file in files:
    df=pd.read_csv(r"C:\Users\946851\OneDrive - Cognizant\Desktop\Pandas-Data-Science-Tasks-master\SalesAnalysis\Sales_Data/" + file)
    all_months_data = pd.concat([all_months_data,df])

all_months_data.to_csv('salesfulldata.csv', index=False)
```

In [128…
```python
all_data = pd.read_csv('salesfulldata.csv')
all_data
```

Out[128]:

	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
...	...	...	...	...	...	...
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

186850 rows × 6 columns

CLEANING

In [129…
```python
#dropping Nan value rows
temp_df = all_data[all_data.isna().any(axis=1)]
temp_df.head()
all_data = all_data.dropna(how='all')
temp_df.head()
```

Out[129]:

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address
1	NaN	NaN	NaN	NaN	NaN	NaN
356	NaN	NaN	NaN	NaN	NaN	NaN
735	NaN	NaN	NaN	NaN	NaN	NaN
1433	NaN	NaN	NaN	NaN	NaN	NaN
1553	NaN	NaN	NaN	NaN	NaN	NaN

In [130…
```python
#Removing OR character in month of orderdate column
all_data[all_data['Order Date'].str[0:2] == 'Or']
all_data = all_data[all_data['Order Date'].str[0:2] != 'Or']
all_data[all_data['Order Date'].str[0:2] == 'Or']
```

Out[130]:

Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address

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

In [131…
```python
all_data.head()
```

Out[131]:

	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

In [132…
```python
#Creating a month column and converting the data type from string to int
all_data['Month'] = all_data['Order Date'].str[0:2]
all_data['Month'] = all_data['Month'].astype('int32')
```

In [133…
```python
#coverting data types
all_data['Quantity Ordered'] = pd.to_numeric(all_data['Quantity Ordered'])
all_data['Price Each'] = pd.to_numeric(all_data['Price Each'])
```

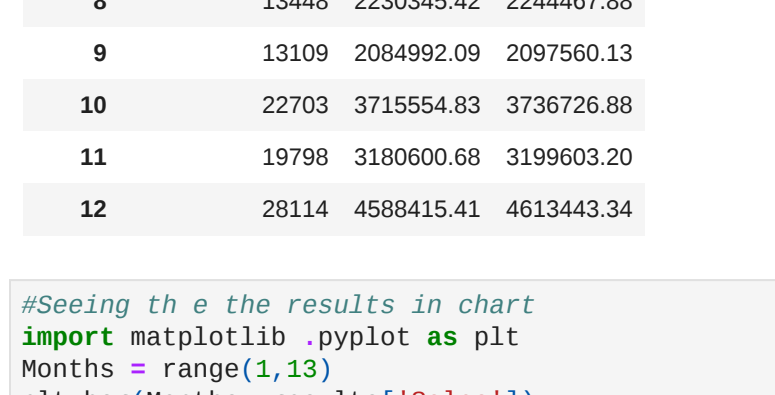
In [134…
```python
#ADD SALES COLUMN
all_data['Sales'] = all_data['Price Each'] * all_data['Quantity Ordered']
```

In [135…
```python
results = all_data.groupby('Month').sum()
results
```

Out[135]:

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 [136…
```python
#Seeing th e the results in chart
import matplotlib .pyplot as plt
Months = range(1,13)
plt.bar(Months, results['Sales'])
plt.xticks(Months)
plt.xlabel('Month')
plt.ylabel('Sales')
plt.show()
#December
```



What city sold the most product?

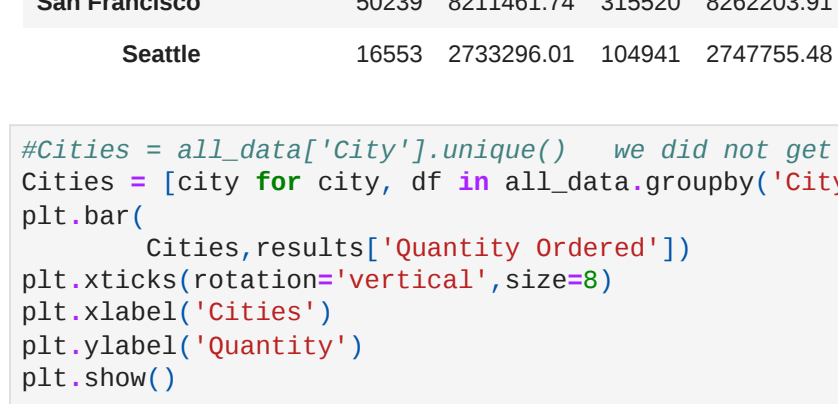
In [137…
```python
#Taking city from address
def get_city(address):
    return address.split(',')[1]
all_data['City'] = all_data['Purchase Address'].apply(lambda x: get_city (x))
```

In [138…
```python
results = all_data.groupby('City').sum()
results
#San Francisco
```

Out[138]:

City	Quantity Ordered	Price Each	Month	Sales
Atlanta	16602	2779908.20	104794	2795498.58
Austin	11153	1809873.61	69829	1819581.75
Boston	22528	3637409.77	141112	3661642.01
Dallas	16730	2752627.82	104620	2767975.40
Los Angeles	33289	5421435.23	208325	5452570.80
New York City	27932	4635370.83	175741	4664317.43
Portland	14053	2307747.47	87765	2320490.61
San Francisco	50239	8211461.74	315520	8262203.91
Seattle	16553	2733296.01	104941	2747755.48

In [139…
```python
#Cities = all_data['City'].unique()   we did not get data in order
Cities = [city for city, df in all_data.groupby('City')]
plt.bar(
        Cities,results['Quantity Ordered'])
plt.xticks(rotation='vertical',size=8)
plt.xlabel('Cities')
plt.ylabel('Quantity')
plt.show()
```



What time should we display advertisements to maximize the likelihood of purchases?

In [140…
```python
all_data.head()
```

Out[140]:

	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 [141…
```python
#Take time from order date
all_data['Time'] = all_data['Order Date'].apply(lambda x: x.split(' ')[1].split(':')[0])
all_data
```

Out[141]:

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address	Month	Sales	City	Time
0	176558	USB-C Charging Cable	2	11.95	04/19/19 08:46	917 1st St, Dallas, TX 75001	4	23.90	Dallas	08
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	09
...	...	...	...	...	...	...	...	...	...	...
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	20
186846	259354	iPhone	1	700.00	09/01/19 16:00	216 Dogwood St, San Francisco, CA 94016	9	700.00	San Francisco	16
186847	259355	iPhone	1	700.00	09/23/19 07:39	220 12th St, San Francisco, CA 94016	9	700.00	San Francisco	07
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	17
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	00

185950 rows × 10 columns

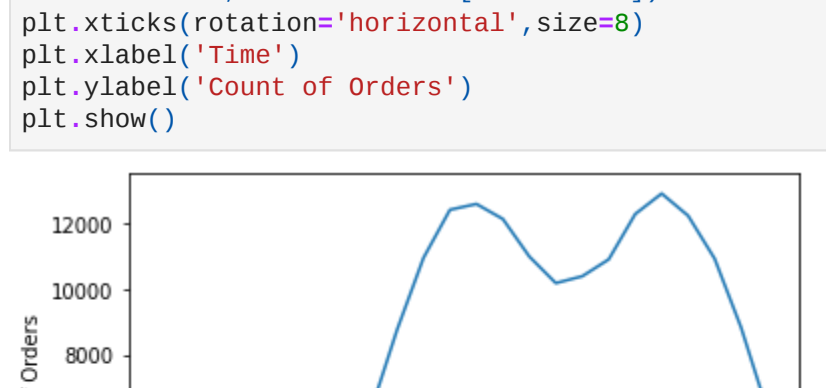
In [142…
```python
#result_ad_time    assing to a dataframe or just straighlty can give in y axis
result_ad_time = all_data.groupby('Time').count()
```

In [143…
```python
result_ad_time
```

Out[143]:

Time	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address	Month	Sales	City
00	3910	3910	3910	3910	3910	3910	3910	3910	3910
01	2350	2350	2350	2350	2350	2350	2350	2350	2350
02	1243	1243	1243	1243	1243	1243	1243	1243	1243
03	831	831	831	831	831	831	831	831	831
04	854	854	854	854	854	854	854	854	854
05	1321	1321	1321	1321	1321	1321	1321	1321	1321
06	2482	2482	2482	2482	2482	2482	2482	2482	2482
07	4011	4011	4011	4011	4011	4011	4011	4011	4011
08	6256	6256	6256	6256	6256	6256	6256	6256	6256
09	8748	8748	8748	8748	8748	8748	8748	8748	8748
10	10944	10944	10944	10944	10944	10944	10944	10944	10944
11	12411	12411	12411	12411	12411	12411	12411	12411	12411
12	12587	12587	12587	12587	12587	12587	12587	12587	12587
13	12129	12129	12129	12129	12129	12129	12129	12129	12129
14	10984	10984	10984	10984	10984	10984	10984	10984	10984
15	10175	10175	10175	10175	10175	10175	10175	10175	10175
16	10384	10384	10384	10384	10384	10384	10384	10384	10384
17	10899	10899	10899	10899	10899	10899	10899	10899	10899
18	12280	12280	12280	12280	12280	12280	12280	12280	12280
19	12905	12905	12905	12905	12905	12905	12905	12905	12905
20	12228	12228	12228	12228	12228	12228	12228	12228	12228
21	10921	10921	10921	10921	10921	10921	10921	10921	10921
22	8822	8822	8822	8822	8822	8822	8822	8822	8822
23	6275	6275	6275	6275	6275	6275	6275	6275	6275

In [144…
```python
Clock = [Time for Time, df in all_data.groupby('Time')]
plt.plot(
        Clock,result_ad_time['Order ID'])
plt.xticks(rotation='horizontal',size=8)
plt.xlabel('Time')
plt.ylabel('Count of Orders')
plt.show()
```



In [145…
```python
#2nd method by converting data type of order data to date time

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

In [146…
```python
all_data.head()
```

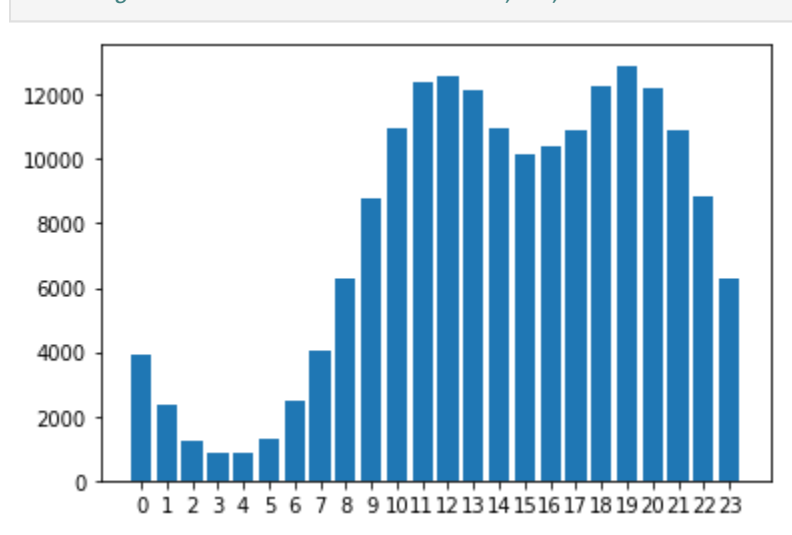
Out[146]:

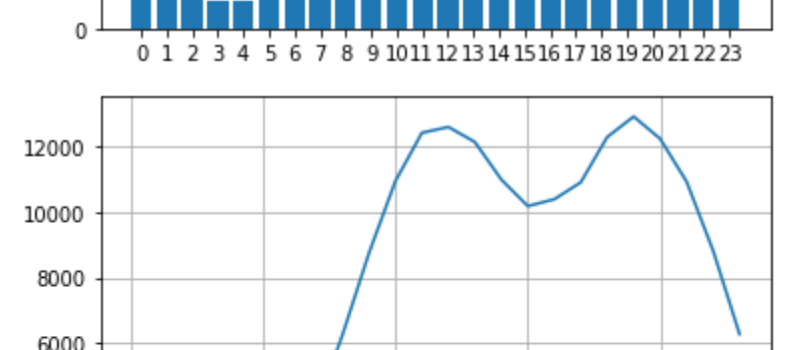
	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address	Month	Sales	City	Time
0	176558	USB-C Charging Cable	2	11.95	2019-04-19 08:46:00	917 1st St, Dallas, TX 75001	4	23.90	Dallas	08
2	176559	Bose SoundSport Headphones	1	99.99	2019-04-07 22:30:00	682 Chestnut St, Boston, MA 02215	4	99.99	Boston	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	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	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	09

In [147…
```python
#Create a hours column
all_data['Hour'] = all_data['Order Date'].dt.hour


Result = all_data.groupby('Hour').count()
```

In [148…
```python
Clock = [time for time, df in all_data.groupby('Hour')]
plt.bar(Clock,Result['Order ID'])
plt.xticks(Clock)
plt.show()
plt.plot(Clock,Result['Order ID'])
plt.grid()
plt.show()
#timings with maximuum order 11,12,19
```





What products are most often sold together?

In [149…
```python
all_data.head()
```

Out[149]:

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address	Month	Sales	City	Time	Hour
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	08	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	22	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	14	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	14	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	09	9

In [150…
```python
#Taking dublicaed row alosne based on order id to a new data frame
new_df =  all_data[all_data['Order ID'].duplicated(keep=False)]
new_df
```

Out[150]:

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address	Month	Sales	City	Time	Hour
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	14	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	14	14
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	19	19
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	19	19
30	176585	Bose SoundSport Headphones	1	99.99	2019-04-07 11:31:00	823 Highland St, Boston, MA 02215	4	99.99	Boston	11	11
...	...	...	...	...	...	...	...	...	...	...	...
186792	259303	AA Batteries (4-pack)	1	3.84	2019-09-20 20:18:00	106 7th St, Atlanta, GA 30301	9	3.84	Atlanta	20	20
186803	259314	Wired Headphones	1	11.99	2019-09-16 00:25:00	241 Highland St, Atlanta, GA 30301	9	11.99	Atlanta	00	0
186804	259314	AAA Batteries (4-pack)	2	2.99	2019-09-16 00:25:00	241 Highland St, Atlanta, GA 30301	9	5.98	Atlanta	00	0
186841	259350	Google Phone	1	600.00	2019-09-30 13:49:00	519 Maple St, San Francisco, CA 94016	9	600.00	San Francisco	13	13
186842	259350	USB-C Charging Cable	1	11.95	2019-09-30 13:49:00	519 Maple St, San Francisco, CA 94016	9	11.95	San Francisco	13	13

14649 rows × 11 columns

In [151…
```python
#Group by based on similiar orderid and concatination of products bought
new_df['all Products'] = new_df.groupby('Order ID') ['Product'].transform(lambda x: ','.join(x))
new_df
```

```
C:\Users\946851\AppData\Local\Temp\7\ipykernel_11084\422982604.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
  new_df['all Products'] = new_df.groupby('Order ID') ['Product'].transform(lambda x: ','.join(x))
```

Out[151]:

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address	Month	Sales	City	Time	Hour	all Products
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	14	14	Google Phone,Wired Headphones
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	14	14	Google Phone,Wired Headphones
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	19	19	Google Phone,USB-C Charging Cable
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	19	19	Google Phone,USB-C Charging Cable
30	176585	Bose SoundSport Headphones	1	99.99	2019-04-07 11:31:00	823 Highland St, Boston, MA 02215	4	99.99	Boston	11	11	Bose SoundSport Headphones,Bose SoundSport Hea...
...	...	...	...	...	...	...	...	...	...	...	...	...
186792	259303	AA Batteries (4-pack)	1	3.84	2019-09-20 20:18:00	106 7th St, Atlanta, GA 30301	9	3.84	Atlanta	20	20	34in Ultrawide Monitor,AA Batteries (4-pack)
186803	259314	Wired Headphones	1	11.99	2019-09-16 00:25:00	241 Highland St, Atlanta, GA 30301	9	11.99	Atlanta	00	0	Wired Headphones,AAA Batteries (4-pack)
186804	259314	AAA Batteries (4-pack)	2	2.99	2019-09-16 00:25:00	241 Highland St, Atlanta, GA 30301	9	5.98	Atlanta	00	0	Wired Headphones,AAA Batteries (4-pack)
186841	259350	Google Phone	1	600.00	2019-09-30 13:49:00	519 Maple St, San Francisco, CA 94016	9	600.00	San Francisco	13	13	Google Phone,USB-C Charging Cable
186842	259350	USB-C Charging Cable	1	11.95	2019-09-30 13:49:00	519 Maple St, San Francisco, CA 94016	9	11.95	San Francisco	13	13	Google Phone,USB-C Charging Cable

14649 rows × 12 columns

In [152…
```python
#Dropping dublicates
new_df  = new_df[['Order ID','all Products']].drop_duplicates()
new_df
```

Out[152]:

	Order ID	all Products
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 [153…
```python
new_df.groupby('all Products').count().sort_values('Order ID',ascending= False)
#As here for one order id 2 products or 3 etc so this might not give the exact result so use below 2 libraries
```

Out[153]:

all Products	Order ID
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
...	...
LG Dryer,27in FHD Monitor	1
LG Dryer,27in 4K Gaming Monitor	1
Google Phone,Wired Headphones,USB-C Charging Cable	1
Google Phone,Wired Headphones,Macbook Pro Laptop	1
iPhone,iPhone	1

366 rows × 1 columns

In [154…
```python
from itertools import combinations
from collections import Counter

count=Counter()

for products in new_df['all Products']:
    product_list = products.split(',')
    count.update(Counter(combinations(product_list,2)))

count.most_common(10)
#'iPhone' and 'Lightning Charging Cable' in  1005 order bought together
```

Out[154]:
```
[(('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)]
```

What product sold the most? Why do you think it did?

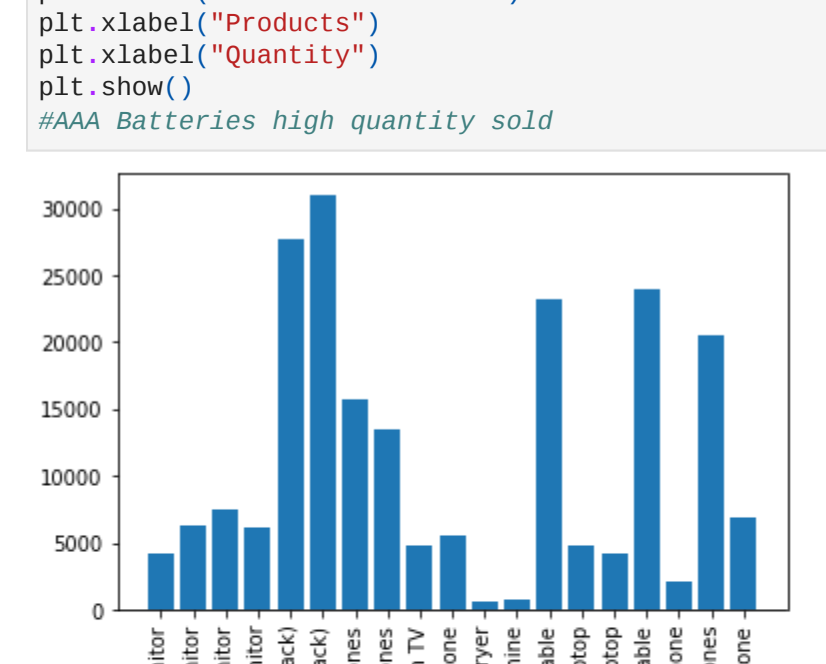
In [157…
```python
all_data.head()
```

Out[157]:

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address	Month	Sales	City	Time	Hour
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	08	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	22	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	14	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	14	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	09	9

In [162…
```python
Products_unique = all_data.groupby('Product')
Purchased_quantity = Products_unique.sum()['Quantity Ordered']
Purchased_quantity

Xaxis = [product for product,df in Products_unique]
plt.bar(Xaxis,Purchased_quantity)
plt.xticks(rotation='vertical')
plt.xlabel("Products")
plt.xlabel("Quantity")
plt.show()
#AAA Batteries high quantity sold
```

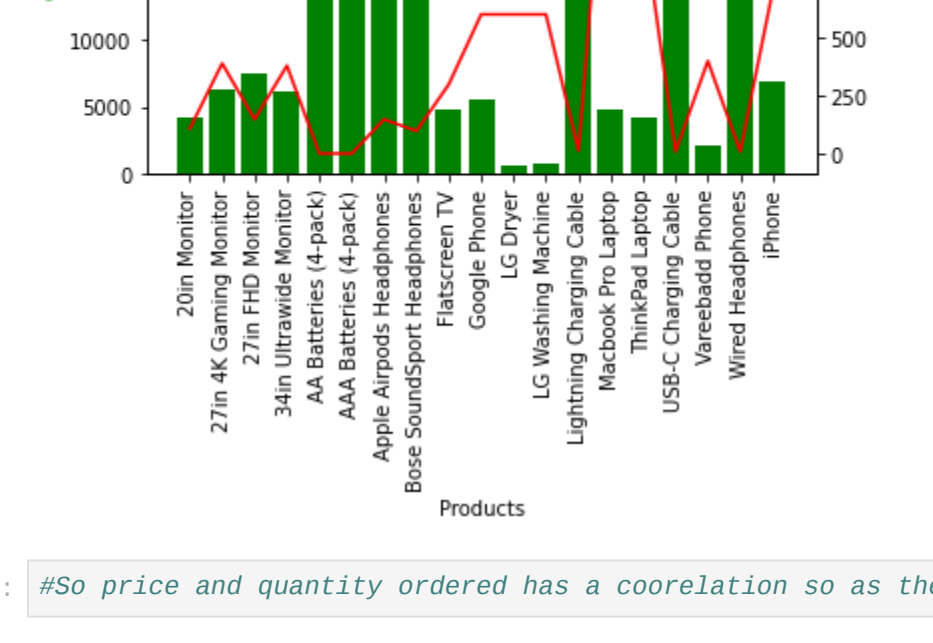


In [184…
```python
Prices = all_data.groupby('Product').mean()['Price Each']
Prices
```

Out[184]:
```
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
```

In [192…
```python
fig, ax1 = plt.subplots()
ax2=ax1.twinx()
ax1.bar(Xaxis,Purchased_quantity,color='g')
ax2.plot(Xaxis,Prices,color ='r')
ax1.set_xticklabels(Xaxis,rotation ='vertical')
ax1.set_xlabel('Products')
ax1.set_ylabel('Quantity',color='g')
ax2.set_ylabel('Price', color='r')
plt.show()
```

```
C:\Users\946851\AppData\Local\Temp\7\ipykernel_11084\2770928800.py:5: UserWarning: FixedFormatter should only be used together with FixedLocator
  ax1.set_xticklabels(Xaxis,rotation ='vertical')
```



In [ ]:
```python
#So price and quantity ordered has a coorelation so as the price is low quantity sold is high.
```