Order ID	Product	Quantity O	Price Each	Order Date	Purchase A
176559	Bose Sound	1	99.99	########	682 Chestn
176560	Google Pho	1	600	########	669 Spruce
176560	Wired Head	1	11.99	########	669 Spruce
176561	Wired Head	1	11.99	05/30/19	9333 8th St,
176562	USB-C Char	1	11.95	04/29/19	1381 Wilson
176563	Bose Sound	1	99.99	########	668 Center
176564	USB-C Char	1	11.95	########	790 Ridge S
184074	AAA Batter	1	2.99	########	544 1st St,
184075	AAA Batter	1	2.99	########	241 11th St
184076	Lightning C	1	14.95	06/17/19	1752 Chestn
179776	Wired Head	2	11.99	04/21/19	2978 Wilson
179790	AAA Batter	2	2.99	########	475 Adams
179815	Wired Head	2	11.99	04/28/19	191 9th St, S
179821	AAA Batter	2	2.99	04/22/19	1975 Main S
179845	AAA Batter	2	2.99	########	541 Adams
179871	AAA Batter	2	2.99	04/24/19	9362 Lake St
179917	AA Batterie	2	3.84	04/24/19	2152 Spruce
179918	USB-C Char	2	11.95	04/14/19	1216 7th St,
179920	AA Batterie	2	3.84	########	598 14th St
179943	AAA Batter	2	2.99	08/14/19	1344 River S
179970	AAA Batter	2	2.99	04/18/19	1668 Lake St
180004	Lightning C	2	14.95	########	307 Lincoln
187091	AA Batterie	3	3.84	########	303 Ridge S
187147	AAA Batter	3	2.99	04/23/19	1929 Willow
187170	AAA Batter	3	2.99	########	810 Ridge S
187467	AA Batterie	3	3.84	04/13/19	1747 Center
187475	AAA Batter	3	2.99	########	859 Wilson
187505	AA Batterie	3	3.84	04/14/19	1752 Ridge S
187567	AAA Batter	3	2.99	04/16/19	2855 Highlan
187588	AA Batterie	3	3.84	04/16/19	280 Main St,
187656	AA Batterie	3	3.84	04/23/19	1376 6th St,
187728	AAA Batter	3	2.99	########	430 Park St
187740	AA Batterie	3	3.84	09/30/19	177 South St
187806	AAA Batter	3	2.99	04/19/19	1159 1st St,
187857	AAA Batter	3	2.99	09/23/19	2384 Hickory
187895	AA Batterie	3	3.84	########	647 14th St
188069	AAA Batter	3	2.99	10/21/19	1472 Chestn
180790	AA Batterie	4	3.84	########	626 7th St,
180941	AAA Batter	4	2.99	04/21/19	1725 2nd St,
181213	AA Batterie	4	3.84	10/28/19	1919 5th St,
181642	AA Batterie	4	3.84	########	695 12th St
181976	USB-C Char	4	11.95	########	561 2nd St,
182448	AAA Batter	4	2.99	########	456 Dogwo
182538	AAA Batter	4	2.99	04/23/19	2516 6th St,
182660	AAA Batter	4	2.99	########	291 Chestn
185349	AAA Batter	5	2.99	04/26/19	7675 Center
185711	AA Batterie	5	3.84	########	481 South S
185915	AA Batterie	5	3.84	11/14/19	1749 Sunset
186028	AAA Batter	5	2.99	########	414 Chestn
186331	AAA Batter	5	2.99	########	553 2nd St,

187182 AA Batterie 5 3.84 ######## 34 South St

190462 AA Batterie 5 3.84 ######## 273 Jackson

Name:Omkar Karlekar

Roll no.:644

PRN:202201090088 BATCH-F3

```
import numpy as np import pandas as pd
all_data=pd.read_csv("/content/1686715083343_all_data.csv"
)
```

```
all_data.head()
```

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address
0	176559.0	Bose SoundSport Headphones	1.0	99.99	04-07-2019 22:30	682 Chestnut St, Boston, MA 02215
1	176560.0	Google Phone	1.0	600.00	04-12-2019 14:38	669 Spruce St, Los Angeles, CA 90001
2	176560.0	Wired Headphones	1.0	11.99	04-12-2019 14:38	669 Spruce St, Los Angeles, CA 90001
3	176561.0	Wired Headphones	1.0	11.99	05/30/19 9:27	333 8th St, Los Angeles, CA 90001
4	176562.0	USB-C Charging Cable	1.0	11.95	04/29/19 13:03	381 Wilson St, San Francisco, CA 94016

clean up data

```
all_data.shape
```

```
(69, 6)
```

Drop rows of nan

```
#find nan
nan_df=all_data[all_data.isna().any(axis=1)]
display(nan_df.head()) all_data.shape
all_data= all_data.dropna(how='all')
all_data.head() all_data.shape
```

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address
36	NaN	NaN	NaN	NaN	NaN	NaN
51	NaN	NaN	NaN	NaN	NaN	NaN

```
(67, 6)
```

Get rid of text in order date column

```
all_data=all_data[all_data['Order Date'].str[0:2]!='Or']
print(all_data)
```

```
    Order ID                    Product  Quantity Ordered  Price Each  \
0   176559.0  Bose SoundSport Headphones               1.0       99.99
1   176560.0                Google Phone               1.0      600.00
2   176560.0            Wired Headphones               1.0       11.99
3   176561.0            Wired Headphones               1.0       11.99
4   176562.0        USB-C Charging Cable               1.0       11.95   ..       ...                      ...              ...
    ...
64  259329.0    Lightning Charging Cable               1.0       14.95
65  259330.0       AA Batteries (4-pack)               2.0        3.84
66  259331.0    Apple Airpods Headphones               1.0      150.00
```

```
67  259332.0    Apple Airpods Headphones              1.0      150.00
68  259333.0  Bose SoundSport Headphones              1.0       99.99

          Order Date                      Purchase Address
0      04-07-2019 22:30       682 Chestnut St, Boston, MA 02215
1      04-12-2019 14:38    669 Spruce St, Los Angeles, CA 90001
2      04-12-2019 14:38    669 Spruce St, Los Angeles, CA 90001
3      05/30/19 9:27       333 8th St, Los Angeles, CA 90001
4      04/29/19 13:03  381 Wilson St, San Francisco, CA 94016  ..              ...                                  ...
64     09-05-2019 19:00       480 Lincoln St, Atlanta, GA 30301
65     09/25/19 22:01    763 Washington St, Seattle, WA 98101
66     09/29/19 7:00     770 4th St, New York City, NY 10001
67     09/16/19 19:21          782 Lake St, Atlanta, GA 30301
68     09/19/19 18:03   347 Ridge St, San Francisco, CA 94016

[67 rows x 6 columns]
```

Make columns correct type

```
all_data['Quality ordered']=pd.to_numeric(all_data['Quantity Ordered'])
all_data['Price Each']= pd.to_numeric(all_data['Price Each'])
```

Augment data with additional columns

add month column

```
all_data['Month'] = all_data['Order Date'].str[0:2]
all_data['Month'] = all_data['Month'].astype('int32')
all_data.head()
```

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address	Quality ordered	Month
0	176559.0	Bose SoundSport Headphones	1.0	99.99	1.0	04-07-2019 22:30	682 Chestnut St, Boston, MA 02215	4
1	176560.0	Google Phone	1.0	600.00	1.0	04-12-2019 14:38	669 Spruce St, Los Angeles, CA 90001	4
2	176560.0	Wired Headphones	1.0	11.99	1.0	04-12-2019 14:38	669 Spruce St, Los Angeles, CA 90001	4

Add month column(alternative method) `all_data['Month 2'] = pd.to_datetime(all_data['Order Date']).dt.month all_data.head()`

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address	Quality ordered	Month	Month 2
0	176559.0	Bose SoundSport Headphones	1.0	99.99	04-072019 22:30	682 Chestnut St, Boston, MA 02215	1.0	04	4.0
1	176560.0	Google Phone	1.0	600.00	04-122019 14:38	669 Spruce St, Los Angeles, CA 90001	1.0	04	4.0
					04-12-	669 Spruce St			

Add city column

```
def get_city(address):
  return address.split(",")[1].strip(" ")
def get_state(address):   return
address.split(",")[2].split(" ")[1]

all_data['City'] = all_data['Purchase Address'].apply(lambda x:f"{get_city(x)} ({get_state(x)})")
all_data.head()
```

	Order ID	Quantity Ordered	Price Each	Order Date	Purchase Address	Quality ordered	Product	Month	city	City
0	176559.0	Bose SoundSport Headphones	1.0	99.99	04-072019 22:30	682 Chestnut St, Boston, MA 02215	1.0	4	Boston (A))	Boston (MA)
1	176560.0	Google Phone	1.0	600.00	04-14:38	669 Spruce St, Los Angeles, CA 90001	1.0	4	Los Angeles (A))	Los Angeles (CA)

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

```
all_data['Sales']=all_data['Quantity Ordered'].astype('int')*all_data['Price Each'].astype('float')
all_data.groupby(['Month']).sum()
```

```
<ipython-input-21-8ba29a3e5d2a>:2: FutureWarning: The default value of numeric_only in DataFrameGroupBy
all_data.groupby(['Month']).sum()
```

Month	Order ID	Quantity Ordered	Price Each	Quality ordered	Sales
4	7335546.0	123.0	885.80	123.0	1210.76
5	353124.0	2.0	111.98	2.0	111.98
6	184076.0	1.0	14.95	1.0	14.95
8	726962.0	9.0	23.92	9.0	50.83
9	2378802.0	17.0	591.44	17.0	616.62
10	550924.0	11.0	10.67	11.0	39.69
11	740314.0	19.0	13.66	19.0	65.31
12	550635.0	17.0	8.97	17.0	50.83

WHICH CITY SOLD THE MOST PRODUCT?

```
Dummycity=all_data.groupby(['city'])
print(Dummycity)
city_max=all_data.groupby(['city']).sum()
print(max(city_max))
```

```
<pandas.core.groupby.generic.DataFrameGroupBy object at 0x7fad4fa7b580>
Sales
<ipython-input-23-b183391abaf5>:3: FutureWarning: The default value of numeric_only in DataFrameGroupBy.sum is deprecated. In a fut
city_max=all_data.groupby(['city']).sum()
```

what products are most often sold together

```
df=all_data[all_data['Order ID'].duplicated(keep=False)]
df['Grouped']=df.groupby('Order ID')['Product'].transform(lambda x:','.join(x))
df2=df[['Order ID','Grouped']].drop_duplicates() print(df['Grouped'])
```

```
1    Google Phone,Wired Headphones
2    Google Phone,Wired Headphones
Name: Grouped, dtype: object <ipython-input-24be4b8fe819be>: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
df['Grouped']=df.groupby('Order ID')['Product'].transform(lambda x:','.join(x))
```

```
from itertools import combinations
from collections import Counter
count=Counter() for row in
df2['Grouped']:
row_list=row.split(',')
count.update(Counter(combinations(row_list,2)))
for key,value in count.most_common(10):
print(key,value)
```

```
('Google Phone', 'Wired Headphones') 1
```

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

```
<ipython-input-28-11142b314e0e>:2: FutureWarning: The default value of numeric_only in DataFrameGroupBy.sum is deprecated. In a fut
quantity_ordered=product_group.sum()['Quantity Ordered']
```

```
print(quantity_ordered)
```

```
Pr                                                alue of numeric_only in DataFrameGroupBy.mean is deprecated. In a fu
oduct
 Batteries (4-pack)          64.0
A Batteries (4-pack)        109.0
ple Airpods Headphones        3.0
se SoundSport Headphones      3.0
ogle Phone                    1.0
ghtning Charging Cable        4.0
B-C Charging Cable            8.0
red Headphones                7.0
me: Quantity Ordered, dtype:
float64
```

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

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

```
oduct
 Batteries (4-pack)           3.84
A Batteries (4-pack)          2.99
ple Airpods Headphones      150.00
se SoundSport Headphones     99.99
ogle Phone                  600.00
ghtning Charging Cable       14.95
B-C Charging Cable           11.95
red Headphones               11.99
me: Price Each, dtype: float64
```