```
## Name : Sakshi Ravindra Aher
PRN no : 202201040089
Roll no : 602
Assignment : 4
```

Name : Sakshi Ravindra Aher

PRN no : 202201040089 Roll no : 602 Assignment : 4

```
import numpy as np
import pandas as pd
all_data=pd.read_csv("/content/sample_data/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

```
#clean up the 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 = all_data.dropna(how='all')
all_data.head()
```

	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

	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

```
#Get rid 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 column correct type
all_data['Quantity Ordered'] = pd.to_numeric(all_data['Quantity Ordered'])
all_data['Price Each'] = pd.to_numeric(all_data['Price Each'])


#add month column (Alternative method)
all_data['Month'] = pd.to_datetime(all_data['Order Date']).dt.month
all_data.head()
```

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

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

```
#Data Exploration!
#Question 1 : 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')
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  Month  \
0   04-07-2019 22:30       682 Chestnut St, Boston, MA 02215      4
1   04-12-2019 14:38    669 Spruce St, Los Angeles, CA 90001      4
2   04-12-2019 14:38    669 Spruce St, Los Angeles, CA 90001      4
3      05/30/19 9:27       333 8th St, Los Angeles, CA 90001      5
4     04/29/19 13:03  381 Wilson St, San Francisco, CA 94016      4
..               ...                                   ...    ...
64  09-05-2019 19:00       480 Lincoln St, Atlanta, GA 30301      9
65    09/25/19 22:01    763 Washington St, Seattle, WA 98101      9
66     09/29/19 7:00     770 4th St, New York City, NY 10001      9
67    09/16/19 19:21          782 Lake St, Atlanta, GA 30301      9
68    09/19/19 18:03   347 Ridge St, San Francisco, CA 94016      9

                  City   sales
0          Boston (MA)   99.99
1     Los Angeles (CA)  600.00
2     Los Angeles (CA)   11.99
```

```
3     Los Angeles (CA)   11.99
4   San Francisco (CA)   11.95
..                 ...     ...
64        Atlanta (GA)   14.95
65        Seattle (WA)    7.68
66  New York City (NY)  150.00
67        Atlanta (GA)  150.00
68  San Francisco (CA)   99.99

[67 rows x 9 columns]
```

```
all_data.groupby(['Month']).sum()
#print(all_data)
```

```
<ipython-input-68-45e4799cd1bc>:1: FutureWarning: The default value of numeric_only in DataFrameGroupBy
  all_data.groupby(['Month']).sum()
```

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

```
#Question 2 : What city sold the most product ?
Dummy = all_data.groupby(['City'])
print(Dummy)
city_max=all_data.groupby(['City']).sum
print(city_max)
```

```
<pandas.core.groupby.generic.DataFrameGroupBy object at 0x7fa40a8dc250>
<bound method GroupBy.sum of <pandas.core.groupby.generic.DataFrameGroupBy object at 0x7fa40a8dfe20>>
```

```
#Question 4 : What product are 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-75-97677b64caf4>:4: 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))
```

```
#Question 3 : 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']
```

```
<ipython-input-82-0a2b4f321535>:3: 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)
```

```
Product
AA Batteries (4-pack)          64.0
AAA Batteries (4-pack)        109.0
Apple Airpods Headphones        3.0
Bose SoundSport Headphones      3.0
Google Phone                    1.0
Lightning Charging Cable        4.0
USB-C Charging Cable            8.0
Wired Headphones                7.0
Name: Quantity Ordered, dtype: float64
```

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

```
<ipython-input-84-225049d1ed32>:1: FutureWarning: The default value of numeric_only in DataFrameGroupBy.mean is deprecated. In a fu
  prices = all_data.groupby('Product').mean()['Price Each']
```

```
print(prices)
```

```
Product
AA Batteries (4-pack)           3.84
AAA Batteries (4-pack)          2.99
Apple Airpods Headphones      150.00
Bose SoundSport Headphones     99.99
Google Phone                  600.00
Lightning Charging Cable       14.95
USB-C Charging Cable           11.95
Wired Headphones               11.99
Name: Price Each, dtype: float64
```