```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
import os
```

In[2]: data = pd.read_csv("all_data.csv")
data.head()

Out[2]:

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address	Month
0	176558	USB Charging Cable	2	11.95	04/19/19 08:46	917 1st St, Dallas, TX 75001	4
1	NaN	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	4
3	176560	Google Phone	1	600	04/12/19 14:38	669 Spruce St, Los Angeles, CA 90001	4
4	176560	Wired Headphones	1	11.99	04/12/19 14:38	669 Spruce St, Los Angeles, CA 90001	4

Clean up the data!

The first step in this is figuring out what we need to clean. I have found in practice, that you find things you need to clean as you perform operations and get errors. Based on the error, you decide how you should go about cleaning the data.

In[3]: data = all_data.dropna()

In[4]: data = all_data[all_data['Order Date'] .str[0:2]!='Or']
data.head(5)

In[5]: data['Quantity Ordered'] = pd.to_numeric(data['Quantity Ordered'])
data['Price Each'] = pd.to_numeric(data['Price Each'])

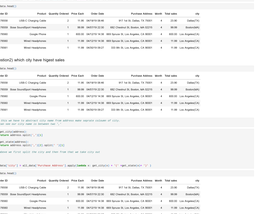
In[6]: data['Month'] = data['Order Date'].str[0:2]
data['Month'] = data['Month'].astype('int32')
data.head()

Out[6]:

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address	Month
0	176558	USB Charging Cable	2	11.95	04/19/19 08:46	917 1st St, Dallas, TX 75001	4
2	176559	Bose SoundSport Headphones	1	99.99	04/07/19 22:30	682 Chestnut St, Boston, MA 02215	4
3	176560	Google Phone	1	600	04/12/19 14:38	669 Spruce St, Los Angeles, CA 90001	4
4	176560	Wired Headphones	1	11.99	04/12/19 14:38	669 Spruce St, Los Angeles, CA 90001	4
5	176561	Wired Headphones	1	11.99	04/30/19 09:27	333 8th St, Los Angeles, CA 90001	4

In[7]: data['Total Sales'] = data['Quantity Ordered']*data['Price Each']

In[8]: a = sns.barplot(data=data[:7], x='Product', y='Total Sales')
for bar in a.patches:
bar_value = bar.get_height()
text = f'{bar_value:,}'
text_x = bar.get_x()
text_y = bar.get_height()
a.text(text_x, text_y, text)



we can see that which month have highest amount of sales and which one is lowest

this is answer to question 3 on at which month sale is highest

Question2) which city have highest sales

In[9]: def get_city(address):
return address.split(",")[1]

In[10]: def get_state(address):
return address.split(",")[2]

this above or this split both the city and state from that we take city and state

In[11]: data['City'] = all_data['Purchase Address'] .apply(lambda x: get_city(x) + ' ' +get_state(x))

In[12]: data.head()

Out[12]:

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address	Month	Total sales	City
0	176558	USB Charging Cable	2	11.95	04/19/19 08:46	917 1st St, Dallas, TX 75001	4	23.90	Dallas TX
2	176559	Bose SoundSport Headphones	1	99.99	04/07/19 22:30	682 Chestnut St, Boston, MA 02215	4	99.99	Boston MA
3	176560	Google Phone	1	600.00	04/12/19 14:38	669 Spruce St, Los Angeles, CA 90001	4	600.00	Los Angeles CA
4	176560	Wired Headphones	1	11.99	04/12/19 14:38	669 Spruce St, Los Angeles, CA 90001	4	11.99	Los Angeles CA
5	176561	Wired Headphones	1	11.99	04/30/19 09:27	333 8th St, Los Angeles, CA 90001	4	11.99	Los Angeles CA

Question2) which city have highest sales

In[13]: def get_city(address):
return address.split(",")[1]

In[14]: def get_state(address):
return address.split(",")[2]

this above or this split both the city and state from that we take city and state

In[15]: data['City'] = all_data['Purchase Address'] .apply(lambda x: get_city(x) + ' ' +get_state(x))

In[16]: data.head()

Out[16]:

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address	Month	Total sales	City
0	176558	USB Charging Cable	2	11.95	04/19/19 08:46	917 1st St, Dallas, TX 75001	4	23.90	Dallas TX
2	176559	Bose SoundSport Headphones	1	99.99	04/07/19 22:30	682 Chestnut St, Boston, MA 02215	4	99.99	Boston MA
3	176560	Google Phone	1	600.00	04/12/19 14:38	669 Spruce St, Los Angeles, CA 90001	4	600.00	Los Angeles CA
4	176560	Wired Headphones	1	11.99	04/12/19 14:38	669 Spruce St, Los Angeles, CA 90001	4	11.99	Los Angeles CA
5	176561	Wired Headphones	1	11.99	04/30/19 09:27	333 8th St, Los Angeles, CA 90001	4	11.99	Los Angeles CA

In[17]: sns.lineplot(data=
group_city_sales.index, y=group_city_sales.values)
plt.xlabel('City')
plt.ylabel('Total Sales')

Question 3) when we should show advertise

In[18]: data.head()

Out[18]:

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address	Month	Total sales	City	Hour
0	176558	USB Charging Cable	2	11.95	04/19/19 08:46	917 1st St, Dallas, TX 75001	4	23.90	Dallas TX	8
2	176559	Bose SoundSport Headphones	1	99.99	04/07/19 22:30	682 Chestnut St, Boston, MA 02215	4	99.99	Boston MA	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 CA	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 CA	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 CA	9

In[19]: a = all_data.groupby('Hour')

In[20]: def get_time(x):
return str(x)+':00'

In[21]: a['Hour'].count().sort_values(ascending=False)[:5].index.map(lambda x: get_time(x))

Out[21]:

```
Index(['19:00', '12:00', '11:00', '18:00', '20:00'], dtype='object')
```

In[22]: a = a['Hour'].count()
sns.lineplot(data=a)
plt.xlabel('Hour')
plt.ylabel('Total Sales')

there we can see that we have peak coming of orders 11 am and 7 pm .

q4) what product often sold together

In[23]: data.head()

Out[23]:

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address	Month	Total sales	City	Hour
0	176558	USB Charging Cable	2	11.95	04/19/19 08:46	917 1st St, Dallas, TX 75001	4	23.90	Dallas TX	8
2	176559	Bose SoundSport Headphones	1	99.99	04/07/19 22:30	682 Chestnut St, Boston, MA 02215	4	99.99	Boston MA	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 CA	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 CA	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 CA	9

In[24]: df = all_data[all_data['Order ID'] .duplicated(keep = False)]
df['Grouped'] = df.groupby('Order ID')['Product'].transform(lambda x : ', '.join(x))
df = df[['Order ID', 'Grouped']].drop_duplicates()
df.head(20)

C:\Users\HP\AppData\Local\Temp\ipykernel_24872\3\4\0\0\0\0.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
df['Grouped'] = df.groupby('Order ID')['Product'].transform(lambda x : ', '.join(x))

Out[24]:

	Order ID	Grouped
8	176574	Google Phone,Wired Headphones
18	176674	Google Phone,USB-C Charging Cable
29	176685	Bose SoundSport Headphones,Bose Soundsport Headphones
35	176690	AAA Batteries (4-pack),Google Phone
62	176718	Lightning Charging Cable,USB-C Charging Cable
73	176861	Apple Airpods Headphones,Lightning Charging Cable
149	176960	Bose SoundSport Headphones,AA Batteries (4-pack)
186	177018	34in Ultrawide Monitor,Google Phone
205	177074	Lightning Charging Cable,USB-C Charging Cable
239	177111	Wired Headphones,Lightning Charging Cable
299	177172	iPhone,Lightning Charging Cable
306	177182	Google Phone,Bose SoundSport Headphones,Wired Headphones
308	177184	Google Phone,USB-C Charging Cable
309	177185	Google Phone,Wired Headphones
318	177193	Google Phone,Wired Headphones
384	176683	Wired Headphones,iPhone,Lightning Charging Cable

In[25]: *#now we have to count the number of products in list grouped and arrange them in decending order.*
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)))
```

In[26]: count.most_common(10)

Out[26]:

```
[(('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),
 (('Lightning Charging Cable', 'Apple Airpods Headphones'), 81),
 (('Google Phone', 'USB-C Charging Cable'), 987),
 (('Apple Airpods Headphones', 'Lightning Charging Cable'), 81),
 (('Bose SoundSport Headphones', 'Google Phone'), 220),
 (('USB-C Charging Cable', 'Google Phone'), 987),
 (('Wired Headphones', 'Google Phone'), 414),
 (('Wired Headphones', 'USB-C Charging Cable'), 160),
 (('Wired Headphones', 'Vareebadd Phone'), 143),
 (('Apple Airpods Headphones', 'iPhone'), 360),
 (('Lightning Charging Cable', 'iPhone'), 1005),
 (('Wired Headphones', 'Lightning Charging Cable'), 92),
 (('Wired Headphones', 'Apple Airpods Headphones'), 81),
 (('Bose SoundSport Headphones', 'Bose Soundsport Headphones'), 90),
 (('USB-C Charging Cable', 'Vareebadd Phone'), 361),
 (('Vareebadd Phone', 'Google Phone'), 143),
 (('Vareebadd Phone', 'Wired Headphones'), 143),
 (('Google Phone', 'Bose Soundsport Headphones'), 90),
 (('Bose Soundsport Headphones', 'Bose SoundSport Headphones'), 90),
 (('Bose Soundsport Headphones', 'Google Phone'), 220),
 (('Bose Soundsport Headphones', 'Google Phone'), 220),
 (('Google Phone', 'Bose SoundSport Headphones'), 220),
 (('Google Phone', 'Bose Soundsport Headphones'), 220),
 (('Google Phone', 'Wired Headphones'), 414),
 (('Google Phone', 'USB-C Charging Cable'), 987),
 (('Google Phone', 'Bose Soundsport Headphones'), 220),
 (('Google Phone', 'Bose SoundS
```