Yearly Sales Report Analysis using Python

In [1]:
```
import pandas as pd
```

In [2]:
```
#file path having all the month sales file

file_path = "E:\\Sales_Data\\Sales_Data\\"
```

In [3]:
```
#importing os for reading all the directories in the given path

import os
```

In [4]:
```
#List of all the file present in path name 'file_path

os.listdir(file_path)
```

Out[4]:
```
['Sales_April_2019.csv',
 'Sales_August_2019.csv',
 'Sales_December_2019.csv',
 'Sales_February_2019.csv',
 'Sales_January_2019.csv',
 'Sales_July_2019.csv',
 'Sales_June_2019.csv',
 'Sales_March_2019.csv',
 'Sales_May_2019.csv',
 'Sales_November_2019.csv',
 'Sales_October_2019.csv',
 'Sales_September_2019.csv']
```

In [5]:
```
#for look for checking if the combined file path and file name is creating proper file

for file in os.listdir(file_path):
    complete_file_path = file_path + file #here file_path is the folder path and file
    print(complete_file_path)
    df=pd.read_csv(complete_file_path)
```

```
E:\Sales_Data\Sales_Data\Sales_April_2019.csv
E:\Sales_Data\Sales_Data\Sales_August_2019.csv
E:\Sales_Data\Sales_Data\Sales_December_2019.csv
E:\Sales_Data\Sales_Data\Sales_February_2019.csv
E:\Sales_Data\Sales_Data\Sales_January_2019.csv
E:\Sales_Data\Sales_Data\Sales_July_2019.csv
E:\Sales_Data\Sales_Data\Sales_June_2019.csv
E:\Sales_Data\Sales_Data\Sales_March_2019.csv
E:\Sales_Data\Sales_Data\Sales_May_2019.csv
E:\Sales_Data\Sales_Data\Sales_November_2019.csv
E:\Sales_Data\Sales_Data\Sales_October_2019.csv
E:\Sales_Data\Sales_Data\Sales_September_2019.csv
```

In [6]:
```
#for loop for checking if the combined file path and file name is creating proper file

for file in os.listdir(file_path):
    complete_file_path = file_path + file #here file_path is the folder path and file
    print(complete_file_path)
```

```
    df=pd.read_csv(complete_file_path)
```

```
E:\Sales_Data\Sales_Data\Sales_April_2019.csv
E:\Sales_Data\Sales_Data\Sales_August_2019.csv
E:\Sales_Data\Sales_Data\Sales_December_2019.csv
E:\Sales_Data\Sales_Data\Sales_February_2019.csv
E:\Sales_Data\Sales_Data\Sales_January_2019.csv
E:\Sales_Data\Sales_Data\Sales_July_2019.csv
E:\Sales_Data\Sales_Data\Sales_June_2019.csv
E:\Sales_Data\Sales_Data\Sales_March_2019.csv
E:\Sales_Data\Sales_Data\Sales_May_2019.csv
E:\Sales_Data\Sales_Data\Sales_November_2019.csv
E:\Sales_Data\Sales_Data\Sales_October_2019.csv
E:\Sales_Data\Sales_Data\Sales_September_2019.csv
```

In [7]:
```
#step of creating an empty data frame and step by step concatinating the monthly sales

final_df=pd.DataFrame()

for file in os.listdir(file_path):
    complete_file_path=file_path+file
    df=pd.read_csv(complete_file_path) #complete path have the exact location for each
    final_df=pd.concat([final_df,df],ignore_index=True)
```

In [607…
```
#after concatinating all the files here is the final dataframe named as final_df

final_df
```

Out[607]:

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address	Total Bill	city	state	city_st
0	176558	USB-C Charging Cable	2	11.95	04/19/19 08:46	917 1st St, Dallas, TX 75001	23.90	Dallas	TX	Dallas
2	176559	Bose SoundSport Headphones	1	99.99	04/07/19 22:30	682 Chestnut St, Boston, MA 02215	99.99	Boston	MA	Boston,
3	176560	Google Phone	1	600.00	04/12/19 14:38	669 Spruce St, Los Angeles, CA 90001	600.00	Los Angeles	CA	Angeles
4	176560	Wired Headphones	1	11.99	04/12/19 14:38	669 Spruce St, Los Angeles, CA 90001	11.99	Los Angeles	CA	Angeles
5	176561	Wired Headphones	1	11.99	04/30/19 09:27	333 8th St, Los Angeles, CA 90001	11.99	Los Angeles	CA	Angeles
...	...	...	...	...	...	...	...	...	...	
186845	259353	AAA Batteries (4-pack)	3	2.99	09/17/19 20:56	840 Highland St, Los Angeles, CA 90001	8.97	Los Angeles	CA	Angeles
186846	259354	iPhone	1	700.00	09/01/19 16:00	216 Dogwood St, San Francisco, CA 94016	700.00	San Francisco	CA	Francisco
186847	259355	iPhone	1	700.00	09/23/19 07:39	220 12th St, San Francisco, CA 94016	700.00	San Francisco	CA	Francisco
186848	259356	34in Ultrawide Monitor	1	379.99	09/19/19 17:30	511 Forest St, San Francisco, CA 94016	379.99	San Francisco	CA	Francisco
186849	259357	USB-C Charging Cable	1	11.95	09/30/19 00:18	250 Meadow St, San Francisco, CA 94016	11.95	San Francisco	CA	Francisco

185950 rows × 11 columns

In [9]:
```
final_df.shape
```

Out[9]:
```
(186850, 6)
```

In [10]:
```
final_df.dtypes
```

Out[10]:
```
Order ID            object
Product             object
Quantity Ordered    object
Price Each          object
Order Date          object
Purchase Address    object
dtype: object
```

currently all the the column have data type of type object also for the column having int values

In [11]:
```
#dropping entries having NaN value for one or more than one column values

final_df.dropna(how = "any", inplace=True)
```

In [12]:
```
#again checking the shape of the finaldf after dropping the NaN values

final_df.shape
```

Out[12]:
```
(186305, 6)
```

Initial entries was 186850 and after dropping NaN values the entries are 186305, so not much significant loss of entries, so we can continue with these dataframe for further analysis

In [13]:
```
#now checking if all the values in int column are integer or all having some other dat

final_df['Quantity Ordered'].value_counts()
```

Out[13]:
```
1                   168552
2                    13324
3                     2920
4                      806
Quantity Ordered       355
5                      236
6                       80
7                       24
8                        5
9                        3
Name: Quantity Ordered, dtype: int64
```

We can see from the above output that the column 'Quantity Ordered' also have a string value having 355 entries, now we have to drop those value as they are a type of error

In [14]:
```
#First creating a boelian mask for entries having Quantity Ordered with will show True
final_df['Quantity Ordered'].str.contains('Quantity Ordered')
```

Out[14]:
```
0         False
2         False
3         False
4         False
5         False
          ...
186845    False
186846    False
186847    False
186848    False
186849    False
Name: Quantity Ordered, Length: 186305, dtype: bool
```

In [15]:
```
#using loc to find those entries

final_df.loc[final_df['Quantity Ordered'].str.contains('Quantity Ordered'),:]
```

Out[15]:

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address
519	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address
1149	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address
1155	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address
2878	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address
2893	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address
...	...	...	...	...	...	...
185164	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address
185551	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address
186563	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address
186632	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address
186738	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address

355 rows × 6 columns

So above are the 355 entries which has to filtered from the actual dataframe

In [16]:
```
#creating a function vice_versa which interchange the boelian values so that we get er

def vice_versa(x):
    if x==True:
        return False
    else:
        return True
```

In [17]:
```
#Now creating the boolean series for

final_df['Quantity Ordered'].str.contains('Quantity Ordered').apply(vice_versa)
```

Out[17]:
```
0         True
2         True
3         True
4         True
5         True
          ...
186845    True
186846    True
186847    True
186848    True
186849    True
Name: Quantity Ordered, Length: 186305, dtype: bool
```

In [18]:
```
#Now using the above boolean mask we will eliminated entries containing string values

final_df.loc[final_df['Quantity Ordered'].str.contains('Quantity Ordered').apply(vice_
```

Out[18]:

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

185950 rows × 6 columns

In [19]:
```
final_df = final_df.loc[final_df['Quantity Ordered'].str.contains('Quantity Ordered').
```

In [20]:
```
#Now checking the value count for the final dataframe

final_df['Quantity Ordered'].value_counts()
```

Out[20]:
```
1    168552
2     13324
3      2920
4       806
5       236
6        80
7        24
8         5
9         3
Name: Quantity Ordered, dtype: int64
```

We can see that now the column 'Quantity Ordered' have no entries as string datatype

In [21]:
```
#Also checking for column 'Price Each'

final_df['Price Each'].value_counts()
```

Out[21]:
```
11.95     21903
14.95     21658
2.99      20641
3.84      20577
11.99     18882
150       15450
99.99     13325
149.99     7507
700        6804
389.99     6230
379.99     6181
600        5490
300        4780
1700       4702
999.99     4128
109.99     4101
400        2056
600.0      1347
150.0        99
700.0        38
1700.0       22
300.0        20
400.0         9
Name: Price Each, dtype: int64
```

In [22]:
```
final_df['Quantity Ordered'].astype(int)
```

Out[22]:
```
0         2
2         1
3         1
4         1
5         1
        ..
186845    3
186846    1
186847    1
186848    1
186849    1
Name: Quantity Ordered, Length: 185950, dtype: int32
```

In [23]:
```
#Changing the data type of column 'Quantity Ordered' and 'Price Each'

final_df=final_df.astype({'Quantity Ordered':'int','Price Each':'float'})
```

In [24]:
```
final_df.dtypes
```

Out[24]:
```
Order ID            object
Product             object
Quantity Ordered     int32
Price Each         float64
Order Date          object
Purchase Address    object
dtype: object
```

In [25]:
```
#Now let us add a column of Total Bill by creating the below operation

final_df['Total Bill']=final_df['Quantity Ordered']*final_df['Price Each']
```

In [26]:
```
final_df
```

Out[26]:

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address	Total Bill
0	176558	USB-C Charging Cable	2	11.95	04/19/19 08:46	917 1st St, Dallas, TX 75001	23.90
2	176559	Bose SoundSport Headphones	1	99.99	04/07/19 22:30	682 Chestnut St, Boston, MA 02215	99.99
3	176560	Google Phone	1	600.00	04/12/19 14:38	669 Spruce St, Los Angeles, CA 90001	600.00
4	176560	Wired Headphones	1	11.99	04/12/19 14:38	669 Spruce St, Los Angeles, CA 90001	11.99
5	176561	Wired Headphones	1	11.99	04/30/19 09:27	333 8th St, Los Angeles, CA 90001	11.99
...	...	...	...	...	...	...	...
186845	259353	AAA Batteries (4-pack)	3	2.99	09/17/19 20:56	840 Highland St, Los Angeles, CA 90001	8.97
186846	259354	iPhone	1	700.00	09/01/19 16:00	216 Dogwood St, San Francisco, CA 94016	700.00
186847	259355	iPhone	1	700.00	09/23/19 07:39	220 12th St, San Francisco, CA 94016	700.00
186848	259356	34in Ultrawide Monitor	1	379.99	09/19/19 17:30	511 Forest St, San Francisco, CA 94016	379.99
186849	259357	USB-C Charging Cable	1	11.95	09/30/19 00:18	250 Meadow St, San Francisco, CA 94016	11.95

185950 rows × 7 columns

Now anlysing various factor affecting sales

Seperating citie and state from column Purchase address

In [27]:
```
final_df['Purchase Address'].str.split(',')
```

Out[27]:
```
0                   [917 1st St,  Dallas,  TX 75001]
2                [682 Chestnut St,  Boston,  MA 02215]
3             [669 Spruce St,  Los Angeles,  CA 90001]
4             [669 Spruce St,  Los Angeles,  CA 90001]
5                [333 8th St,  Los Angeles,  CA 90001]
                              ...
186845     [840 Highland St,  Los Angeles,  CA 90001]
186846    [216 Dogwood St,  San Francisco,  CA 94016]
186847       [220 12th St,  San Francisco,  CA 94016]
186848     [511 Forest St,  San Francisco,  CA 94016]
186849     [250 Meadow St,  San Francisco,  CA 94016]
Name: Purchase Address, Length: 185950, dtype: object
```

In [28]:
```
#Assigning city column to the final_df dataframe
final_df['city']=final_df['Purchase Address'].str.split(',').str[1].str.strip()
```

In [29]:
```
final_df
```

Out[29]:

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address	Total Bill	city
0	176558	USB-C Charging Cable	2	11.95	04/19/19 08:46	917 1st St, Dallas, TX 75001	23.90	Dallas
2	176559	Bose SoundSport Headphones	1	99.99	04/07/19 22:30	682 Chestnut St, Boston, MA 02215	99.99	Boston
3	176560	Google Phone	1	600.00	04/12/19 14:38	669 Spruce St, Los Angeles, CA 90001	600.00	Los Angeles
4	176560	Wired Headphones	1	11.99	04/12/19 14:38	669 Spruce St, Los Angeles, CA 90001	11.99	Los Angeles
5	176561	Wired Headphones	1	11.99	04/30/19 09:27	333 8th St, Los Angeles, CA 90001	11.99	Los Angeles
...	...	...	...	...	...	...	...	...
186845	259353	AAA Batteries (4-pack)	3	2.99	09/17/19 20:56	840 Highland St, Los Angeles, CA 90001	8.97	Los Angeles
186846	259354	iPhone	1	700.00	09/01/19 16:00	216 Dogwood St, San Francisco, CA 94016	700.00	San Francisco
186847	259355	iPhone	1	700.00	09/23/19 07:39	220 12th St, San Francisco, CA 94016	700.00	San Francisco
186848	259356	34in Ultrawide Monitor	1	379.99	09/19/19 17:30	511 Forest St, San Francisco, CA 94016	379.99	San Francisco
186849	259357	USB-C Charging Cable	1	11.95	09/30/19 00:18	250 Meadow St, San Francisco, CA 94016	11.95	San Francisco

185950 rows × 8 columns

In [30]:
```
#Unique value of country with no of entries

final_df['city'].value_counts()
```

Out[30]:
```
San Francisco    44732
Los Angeles      29605
New York City    24876
Boston           19934
Atlanta          14881
Dallas           14820
Seattle          14732
Portland         12465
Austin            9905
Name: city, dtype: int64
```

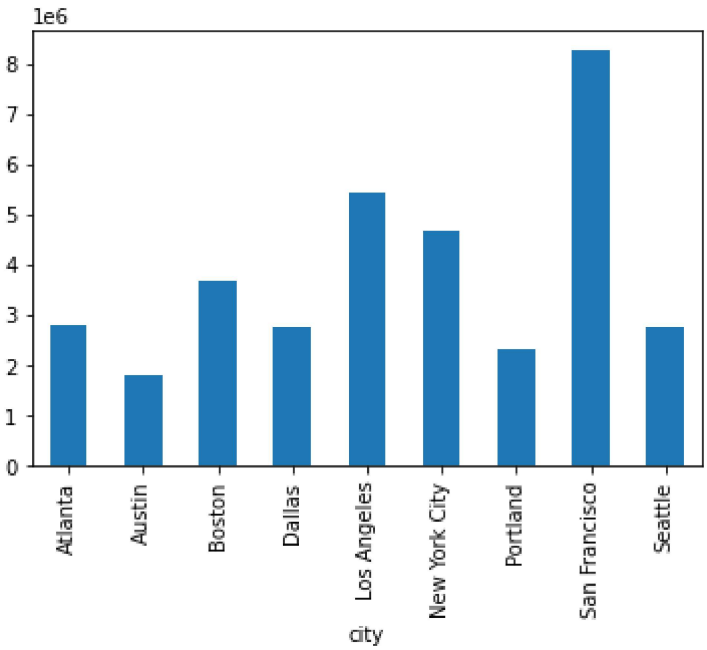
In [31]:
```
#calculating total bill for each country

final_df.groupby(['city'])['Total Bill'].sum()
```

Out[31]:
```
city
Atlanta          2795498.58
Austin           1819581.75
Boston           3661642.01
Dallas           2767975.40
Los Angeles      5452570.80
New York City    4664317.43
Portland         2320490.61
San Francisco    8262203.91
Seattle          2747755.48
Name: Total Bill, dtype: float64
```

In [32]:
```
import matplotlib
```

In [33]:
```
final_df.groupby(['city'])['Total Bill'].sum().plot(kind='bar')
```

Out[33]:
```
<AxesSubplot:xlabel='city'>
```



In [34]:
```
#Seperation State from the Purchase order column and adding to the dataframe as a new

final_df['Purchase Address'].str.split(',').str[2].str.strip().str.split(' ').str[0]
```

Out[34]:
```
0         TX
2         MA
3         CA
4         CA
5         CA
          ..
186845    CA
186846    CA
186847    CA
186848    CA
186849    CA
Name: Purchase Address, Length: 185950, dtype: object
```

In [35]:
```
final_df['state']=final_df['Purchase Address'].str.split(',').str[2].str.strip().str.s
```

In [36]:
```
(final_df['city']+','+final_df['state']).unique()
```

Out[36]:
```
array(['Dallas,TX', 'Boston,MA', 'Los Angeles,CA', 'San Francisco,CA',
       'Seattle,WA', 'Atlanta,GA', 'New York City,NY', 'Portland,OR',
       'Austin,TX', 'Portland,ME'], dtype=object)
```

We can see that CA city is present in two states so grouping city on will give us wrong data vaue we have to concat city and state and then we have to group is to get Sales per city

In [38]:
```
final_df['city_state']=final_df['city']+','+final_df['state']
```

In [39]:
```
final_df
```

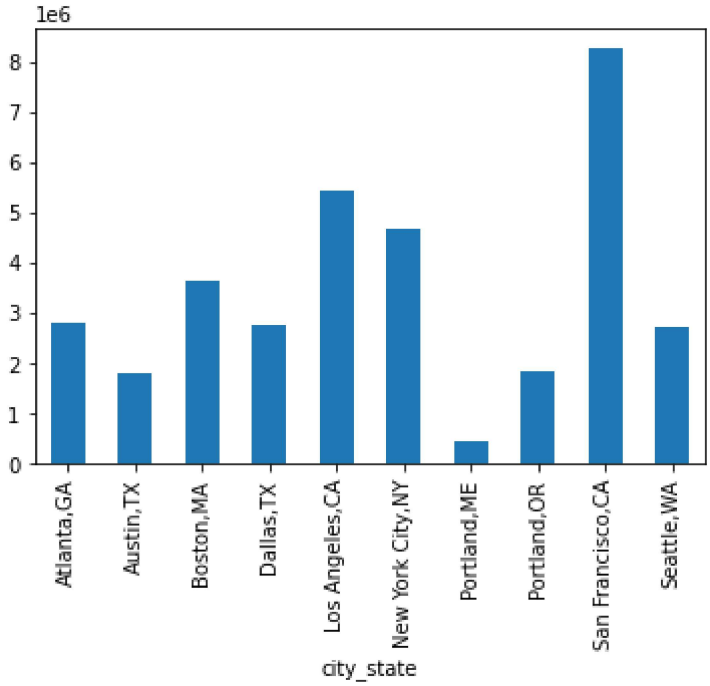
Out[39]:

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address	Total Bill	city	state	city_st
0	176558	USB-C Charging Cable	2	11.95	04/19/19 08:46	917 1st St, Dallas, TX 75001	23.90	Dallas	TX	Dallas
2	176559	Bose SoundSport Headphones	1	99.99	04/07/19 22:30	682 Chestnut St, Boston, MA 02215	99.99	Boston	MA	Boston,
3	176560	Google Phone	1	600.00	04/12/19 14:38	669 Spruce St, Los Angeles, CA 90001	600.00	Los Angeles	CA	Angeles
4	176560	Wired Headphones	1	11.99	04/12/19 14:38	669 Spruce St, Los Angeles, CA 90001	11.99	Los Angeles	CA	Angeles
5	176561	Wired Headphones	1	11.99	04/30/19 09:27	333 8th St, Los Angeles, CA 90001	11.99	Los Angeles	CA	Angeles
...	...	...	...	...	...	...	...	...	...	
186845	259353	AAA Batteries (4-pack)	3	2.99	09/17/19 20:56	840 Highland St, Los Angeles, CA 90001	8.97	Los Angeles	CA	Angeles
186846	259354	iPhone	1	700.00	09/01/19 16:00	216 Dogwood St, San Francisco, CA 94016	700.00	San Francisco	CA	Francisco
186847	259355	iPhone	1	700.00	09/23/19 07:39	220 12th St, San Francisco, CA 94016	700.00	San Francisco	CA	Francisco
186848	259356	34in Ultrawide Monitor	1	379.99	09/19/19 17:30	511 Forest St, San Francisco, CA 94016	379.99	San Francisco	CA	Francisco
186849	259357	USB-C Charging Cable	1	11.95	09/30/19 00:18	250 Meadow St, San Francisco, CA 94016	11.95	San Francisco	CA	Francisco

185950 rows × 10 columns

In [41]:
```
final_df.groupby(['city_state'])['Total Bill'].sum().plot(kind='bar')
```

Out[41]:
```
<AxesSubplot:xlabel='city_state'>
```



In [43]:
```
final_df

#Now we want to see which hour of the day maximum order is being made so that the adve
```

Out[43]:

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address	Total Bill	city	state	city_st
0	176558	USB-C Charging Cable	2	11.95	04/19/19 08:46	917 1st St, Dallas, TX 75001	23.90	Dallas	TX	Dallas
2	176559	Bose SoundSport Headphones	1	99.99	04/07/19 22:30	682 Chestnut St, Boston, MA 02215	99.99	Boston	MA	Boston,
3	176560	Google Phone	1	600.00	04/12/19 14:38	669 Spruce St, Los Angeles, CA 90001	600.00	Los Angeles	CA	Angeles
4	176560	Wired Headphones	1	11.99	04/12/19 14:38	669 Spruce St, Los Angeles, CA 90001	11.99	Los Angeles	CA	Angeles
5	176561	Wired Headphones	1	11.99	04/30/19 09:27	333 8th St, Los Angeles, CA 90001	11.99	Los Angeles	CA	Angeles
...	...	...	...	...	...	...	...	...	...	
186845	259353	AAA Batteries (4-pack)	3	2.99	09/17/19 20:56	840 Highland St, Los Angeles, CA 90001	8.97	Los Angeles	CA	Angeles
186846	259354	iPhone	1	700.00	09/01/19 16:00	216 Dogwood St, San Francisco, CA 94016	700.00	San Francisco	CA	Francisco
186847	259355	iPhone	1	700.00	09/23/19 07:39	220 12th St, San Francisco, CA 94016	700.00	San Francisco	CA	Francisco
186848	259356	34in Ultrawide Monitor	1	379.99	09/19/19 17:30	511 Forest St, San Francisco, CA 94016	379.99	San Francisco	CA	Francisco
186849	259357	USB-C Charging Cable	1	11.95	09/30/19 00:18	250 Meadow St, San Francisco, CA 94016	11.95	San Francisco	CA	Francisco

185950 rows × 10 columns

In [53]:
```
final_df['Order Date'].str.split(' ').str[1]
```

Out[53]:
```
0         08:46
2         22:30
3         14:38
4         14:38
5         09:27
          ...
186845    20:56
186846    16:00
186847    07:39
186848    17:30
186849    00:18
Name: Order Date, Length: 185950, dtype: object
```

In [56]:
```
#Now seperating the hour from the about time series

final_df['Order Date'].str.split(' ').str[1].str.split(':').str[0].str.strip()
```

Out[56]:
```
0         08
2         22
3         14
4         14
5         09
          ..
186845    20
186846    16
186847    07
186848    17
186849    00
Name: Order Date, Length: 185950, dtype: object
```

In [57]:
```
#Assisgning the above hour series into a new column in the dataframe

final_df['Hour_of_Purchase']=final_df['Order Date'].str.split(' ').str[1].str.split('
```

In [58]:
```
final_df
```

Out[58]:

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address	Total Bill	city	state	city_st
0	176558	USB-C Charging Cable	2	11.95	04/19/19 08:46	917 1st St, Dallas, TX 75001	23.90	Dallas	TX	Dallas
2	176559	Bose SoundSport Headphones	1	99.99	04/07/19 22:30	682 Chestnut St, Boston, MA 02215	99.99	Boston	MA	Boston,
3	176560	Google Phone	1	600.00	04/12/19 14:38	669 Spruce St, Los Angeles, CA 90001	600.00	Los Angeles	CA	Angeles
4	176560	Wired Headphones	1	11.99	04/12/19 14:38	669 Spruce St, Los Angeles, CA 90001	11.99	Los Angeles	CA	Angeles
5	176561	Wired Headphones	1	11.99	04/30/19 09:27	333 8th St, Los Angeles, CA 90001	11.99	Los Angeles	CA	Angeles
...	...	...	...	...	...	...	...	...	...	
186845	259353	AAA Batteries (4-pack)	3	2.99	09/17/19 20:56	840 Highland St, Los Angeles, CA 90001	8.97	Los Angeles	CA	Angeles
186846	259354	iPhone	1	700.00	09/01/19 16:00	216 Dogwood St, San Francisco, CA 94016	700.00	San Francisco	CA	Francisco
186847	259355	iPhone	1	700.00	09/23/19 07:39	220 12th St, San Francisco, CA 94016	700.00	San Francisco	CA	Francisco
186848	259356	34in Ultrawide Monitor	1	379.99	09/19/19 17:30	511 Forest St, San Francisco, CA 94016	379.99	San Francisco	CA	Francisco
186849	259357	USB-C Charging Cable	1	11.95	09/30/19 00:18	250 Meadow St, San Francisco, CA 94016	11.95	San Francisco	CA	Francisco

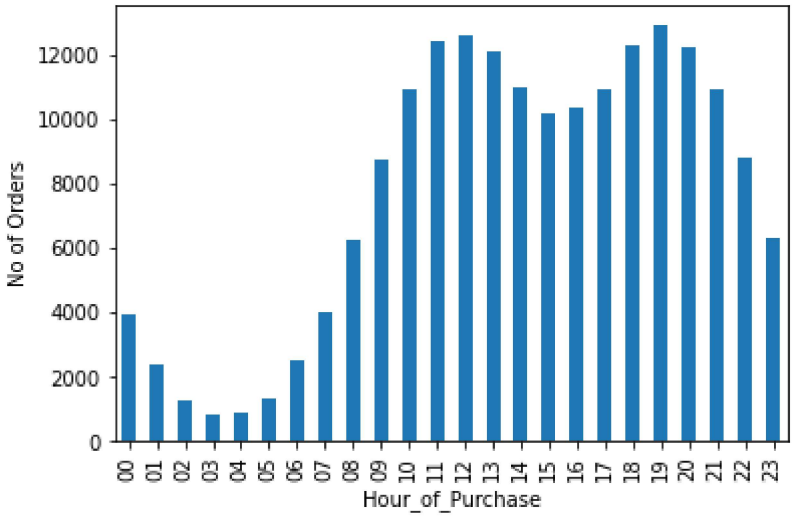
185950 rows × 11 columns

In [542…
```
final_df.groupby(['Hour_of_Purchase'])['Order ID'].count()
```

Out[542]:
```
Hour_of_Purchase
00     3910
01     2350
02     1243
03      831
04      854
05     1321
06     2482
07     4011
08     6256
09     8748
10    10944
11    12411
12    12587
13    12129
14    10984
15    10175
16    10384
17    10899
18    12280
19    12905
20    12228
21    10921
22     8822
23     6275
Name: Order ID, dtype: int64
```

In [544…
```
%matplotlib inline
```

In [609…
```
final_df.groupby(['Hour_of_Purchase'])['Order ID'].count().plot(kind='bar')
plt.ylabel('No of Orders')
plt.show()
```



The above graph shows count of order at every hour and we can see that between 11 and 12am in the morning and 5 to 6pm in the afternoon maximum order is being placed, so the advertisement can be best run at this time more sales

Now let us analyse the data and find out which two product is sold together most of the time If any product will is sold together will surely have same product id so using DUPLICATE function we can find product will same order no

In [71]:
```
final_df
```

Out[71]:

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address	Total Bill	city	state	city_st
0	176558	USB-C Charging Cable	2	11.95	04/19/19 08:46	917 1st St, Dallas, TX 75001	23.90	Dallas	TX	Dallas
2	176559	Bose SoundSport Headphones	1	99.99	04/07/19 22:30	682 Chestnut St, Boston, MA 02215	99.99	Boston	MA	Boston,
3	176560	Google Phone	1	600.00	04/12/19 14:38	669 Spruce St, Los Angeles, CA 90001	600.00	Los Angeles	CA	Angeles
4	176560	Wired Headphones	1	11.99	04/12/19 14:38	669 Spruce St, Los Angeles, CA 90001	11.99	Los Angeles	CA	Angeles
5	176561	Wired Headphones	1	11.99	04/30/19 09:27	333 8th St, Los Angeles, CA 90001	11.99	Los Angeles	CA	Angeles
...	...	...	...	...	...	...	...	...	...	
186845	259353	AAA Batteries (4-pack)	3	2.99	09/17/19 20:56	840 Highland St, Los Angeles, CA 90001	8.97	Los Angeles	CA	Angeles
186846	259354	iPhone	1	700.00	09/01/19 16:00	216 Dogwood St, San Francisco, CA 94016	700.00	San Francisco	CA	Francisco
186847	259355	iPhone	1	700.00	09/23/19 07:39	220 12th St, San Francisco, CA 94016	700.00	San Francisco	CA	Francisco
186848	259356	34in Ultrawide Monitor	1	379.99	09/19/19 17:30	511 Forest St, San Francisco, CA 94016	379.99	San Francisco	CA	Francisco
186849	259357	USB-C Charging Cable	1	11.95	09/30/19 00:18	250 Meadow St, San Francisco, CA 94016	11.95	San Francisco	CA	Francisco

185950 rows × 11 columns

In [74]:
```
#This dataframe shows the product with same order id

final_df.loc[final_df['Order ID'].duplicated(keep=False),['Order ID','Product']]
```

Out[74]:

	Order ID	Product
3	176560	Google Phone
4	176560	Wired Headphones
18	176574	Google Phone
19	176574	USB-C Charging Cable
30	176585	Bose SoundSport Headphones
...	...	...
186792	259303	AA Batteries (4-pack)
186803	259314	Wired Headphones
186804	259314	AAA Batteries (4-pack)
186841	259350	Google Phone
186842	259350	USB-C Charging Cable

14649 rows × 2 columns

In [75]:
```
dup_df= final_df.loc[final_df['Order ID'].duplicated(keep=False),['Order ID','Product'
```

In [77]:
```
dup_df
```

Out[77]:

	Order ID	Product
3	176560	Google Phone
4	176560	Wired Headphones
18	176574	Google Phone
19	176574	USB-C Charging Cable
30	176585	Bose SoundSport Headphones
...	...	...
186792	259303	AA Batteries (4-pack)
186803	259314	Wired Headphones
186804	259314	AAA Batteries (4-pack)
186841	259350	Google Phone
186842	259350	USB-C Charging Cable

14649 rows × 2 columns

In [82]:
```
#self joining the dataframe 'dup_if' on column Order ID

merge_df=pd.merge(dup_df,dup_df,on='Order ID')
```

In [83]:
```
merge_df
```

Out[83]:

	Order ID	Product_x	Product_y
0	176560	Google Phone	Google Phone
1	176560	Google Phone	Wired Headphones
2	176560	Wired Headphones	Google Phone
3	176560	Wired Headphones	Wired Headphones
4	176574	Google Phone	Google Phone
...	...	...	...
30464	259314	AAA Batteries (4-pack)	AAA Batteries (4-pack)
30465	259350	Google Phone	Google Phone
30466	259350	Google Phone	USB-C Charging Cable
30467	259350	USB-C Charging Cable	Google Phone
30468	259350	USB-C Charging Cable	USB-C Charging Cable

30469 rows × 3 columns

In [86]:
```
#Merging the column Product_x and Product_y

merge_df['All Products']=merge_df['Product_x']+','+merge_df['Product_y']
```

In [94]:
```
#Final merged dataframe

merge_df
```

Out[94]:

	Order ID	Product_x	Product_y	All Products
0	176560	Google Phone	Google Phone	Google Phone,Google Phone
1	176560	Google Phone	Wired Headphones	Google Phone,Wired Headphones
2	176560	Wired Headphones	Google Phone	Wired Headphones,Google Phone
3	176560	Wired Headphones	Wired Headphones	Wired Headphones,Wired Headphones
4	176574	Google Phone	Google Phone	Google Phone,Google Phone
...	...	...	...	...
30464	259314	AAA Batteries (4-pack)	AAA Batteries (4-pack)	AAA Batteries (4-pack),AAA Batteries (4-pack)
30465	259350	Google Phone	Google Phone	Google Phone,Google Phone
30466	259350	Google Phone	USB-C Charging Cable	Google Phone,USB-C Charging Cable
30467	259350	USB-C Charging Cable	Google Phone	USB-C Charging Cable,Google Phone
30468	259350	USB-C Charging Cable	USB-C Charging Cable	USB-C Charging Cable,USB-C Charging Cable

30469 rows × 4 columns

In [93]:
```
import numpy as np
```

In [95]:
```
#Creating a boolean mask to eliminated rows having same Product_x and Product_y value

x=np.where(merge_df['Product_x']==merge_df['Product_y'],False,True)
```

In [96]:
```
x
```

Out[96]:
```
array([False,  True,  True, ...,  True,  True, False])
```

In [110…
```
# Now finally filtering the merge_df dataframe

merge_df=merge_df.loc[x,:]
```

In [111…
```
merge_df
```

Out[111]:

	Order ID	Product_x	Product_y	All Products
1	176560	Google Phone	Wired Headphones	Google Phone,Wired Headphones
2	176560	Wired Headphones	Google Phone	Wired Headphones,Google Phone
5	176574	Google Phone	USB-C Charging Cable	Google Phone,USB-C Charging Cable
6	176574	USB-C Charging Cable	Google Phone	USB-C Charging Cable,Google Phone
13	176586	AAA Batteries (4-pack)	Google Phone	AAA Batteries (4-pack),Google Phone
...	...	...	...	...
30459	259303	AA Batteries (4-pack)	34in Ultrawide Monitor	AA Batteries (4-pack),34in Ultrawide Monitor
30462	259314	Wired Headphones	AAA Batteries (4-pack)	Wired Headphones,AAA Batteries (4-pack)
30463	259314	AAA Batteries (4-pack)	Wired Headphones	AAA Batteries (4-pack),Wired Headphones
30466	259350	Google Phone	USB-C Charging Cable	Google Phone,USB-C Charging Cable
30467	259350	USB-C Charging Cable	Google Phone	USB-C Charging Cable,Google Phone

15198 rows × 4 columns

In [112…

```
#Keeping both the items of a particular Order ID in a list

merge_df['All Products'].str.split(',')
```

Out[112]:

```
1                       [Google Phone, Wired Headphones]
2                       [Wired Headphones, Google Phone]
5                    [Google Phone, USB-C Charging Cable]
6                    [USB-C Charging Cable, Google Phone]
13                  [AAA Batteries (4-pack), Google Phone]
                                ...
30459    [AA Batteries (4-pack), 34in Ultrawide Monitor]
30462         [Wired Headphones, AAA Batteries (4-pack)]
30463         [AAA Batteries (4-pack), Wired Headphones]
30466               [Google Phone, USB-C Charging Cable]
30467               [USB-C Charging Cable, Google Phone]
Name: All Products, Length: 15198, dtype: object
```

In [122…

```
merge_df['All_Product_list']=merge_df['All Products'].str.split(',')
```

```
C:\Users\dell\AppData\Local\Temp\ipykernel_7756\3384082564.py:1: SettingWithCopyWarni
ng:
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/us
er_guide/indexing.html#returning-a-view-versus-a-copy
  merge_df['All_Product_list']=merge_df['All Products'].str.split(',')
```

In [127…

```
#The All_Product_list contain a list of items bought together

merge_df
```

Out[127]:

	Order ID	Product_x	Product_y	All Products	All_Product_list
1	176560	Google Phone	Wired Headphones	Google Phone,Wired Headphones	[Google Phone, Wired Headphones]
2	176560	Wired Headphones	Google Phone	Wired Headphones,Google Phone	[Wired Headphones, Google Phone]
5	176574	Google Phone	USB-C Charging Cable	Google Phone,USB-C Charging Cable	[Google Phone, USB-C Charging Cable]
6	176574	USB-C Charging Cable	Google Phone	USB-C Charging Cable,Google Phone	[USB-C Charging Cable, Google Phone]
13	176586	AAA Batteries (4-pack)	Google Phone	AAA Batteries (4-pack),Google Phone	[AAA Batteries (4-pack), Google Phone]
…	…	…	…	…	…
30459	259303	AA Batteries (4-pack)	34in Ultrawide Monitor	AA Batteries (4-pack),34in Ultrawide Monitor	[AA Batteries (4-pack), 34in Ultrawide Monitor]
30462	259314	Wired Headphones	AAA Batteries (4-pack)	Wired Headphones,AAA Batteries (4-pack)	[Wired Headphones, AAA Batteries (4-pack)]
30463	259314	AAA Batteries (4-pack)	Wired Headphones	AAA Batteries (4-pack),Wired Headphones	[AAA Batteries (4-pack), Wired Headphones]
30466	259350	Google Phone	USB-C Charging Cable	Google Phone,USB-C Charging Cable	[Google Phone, USB-C Charging Cable]
30467	259350	USB-C Charging Cable	Google Phone	USB-C Charging Cable,Google Phone	[USB-C Charging Cable, Google Phone]

15198 rows × 5 columns

In [131…

```
#Apply the function 'lambda x:x.sort()'for sorting the values of all entries of columr

merge_df['All_Product_list'].apply(lambda x:x.sort())
```

Out[131]:
```
1         None
2         None
5         None
6         None
13        None
          ...
30459     None
30462     None
30463     None
30466     None
30467     None
Name: All_Product_list, Length: 15198, dtype: object
```

In [132…
```
#So, now we have to eliminate the duplicate for column 'Order ID' and 'All_Product_lis

merge_df
```

Out[132]:

	Order ID	Product_x	Product_y	All Products	All_Product_list
1	176560	Google Phone	Wired Headphones	Google Phone,Wired Headphones	[Google Phone, Wired Headphones]
2	176560	Wired Headphones	Google Phone	Wired Headphones,Google Phone	[Google Phone, Wired Headphones]
5	176574	Google Phone	USB-C Charging Cable	Google Phone,USB-C Charging Cable	[Google Phone, USB-C Charging Cable]
6	176574	USB-C Charging Cable	Google Phone	USB-C Charging Cable,Google Phone	[Google Phone, USB-C Charging Cable]
13	176586	AAA Batteries (4-pack)	Google Phone	AAA Batteries (4-pack),Google Phone	[AAA Batteries (4-pack), Google Phone]
…	…	…	…	…	…
30459	259303	AA Batteries (4-pack)	34in Ultrawide Monitor	AA Batteries (4-pack),34in Ultrawide Monitor	[34in Ultrawide Monitor, AA Batteries (4-pack)]
30462	259314	Wired Headphones	AAA Batteries (4-pack)	Wired Headphones,AAA Batteries (4-pack)	[AAA Batteries (4-pack), Wired Headphones]
30463	259314	AAA Batteries (4-pack)	Wired Headphones	AAA Batteries (4-pack),Wired Headphones	[AAA Batteries (4-pack), Wired Headphones]
30466	259350	Google Phone	USB-C Charging Cable	Google Phone,USB-C Charging Cable	[Google Phone, USB-C Charging Cable]
30467	259350	USB-C Charging Cable	Google Phone	USB-C Charging Cable,Google Phone	[Google Phone, USB-C Charging Cable]

15198 rows × 5 columns

In [140…
```
merge_df
```

Out[140]:

	Order ID	Product_x	Product_y	All Products	All_Product_list
1	176560	Google Phone	Wired Headphones	Google Phone,Wired Headphones	[Google Phone, Wired Headphones]
2	176560	Wired Headphones	Google Phone	Wired Headphones,Google Phone	[Google Phone, Wired Headphones]
5	176574	Google Phone	USB-C Charging Cable	Google Phone,USB-C Charging Cable	[Google Phone, USB-C Charging Cable]
6	176574	USB-C Charging Cable	Google Phone	USB-C Charging Cable,Google Phone	[Google Phone, USB-C Charging Cable]
13	176586	AAA Batteries (4-pack)	Google Phone	AAA Batteries (4-pack),Google Phone	[AAA Batteries (4-pack), Google Phone]
...	...	...	...	...	...
30459	259303	AA Batteries (4-pack)	34in Ultrawide Monitor	AA Batteries (4-pack),34in Ultrawide Monitor	[34in Ultrawide Monitor, AA Batteries (4-pack)]
30462	259314	Wired Headphones	AAA Batteries (4-pack)	Wired Headphones,AAA Batteries (4-pack)	[AAA Batteries (4-pack), Wired Headphones]
30463	259314	AAA Batteries (4-pack)	Wired Headphones	AAA Batteries (4-pack),Wired Headphones	[AAA Batteries (4-pack), Wired Headphones]
30466	259350	Google Phone	USB-C Charging Cable	Google Phone,USB-C Charging Cable	[Google Phone, USB-C Charging Cable]
30467	259350	USB-C Charging Cable	Google Phone	USB-C Charging Cable,Google Phone	[Google Phone, USB-C Charging Cable]

15198 rows × 5 columns

In [159…

```
#the duplicate values gets eliminated

merge_df.drop_duplicates(subset='Order ID',keep='first',inplace=True)
```

```
C:\Users\dell\AppData\Local\Temp\ipykernel_7756\1271223200.py:1: SettingWithCopyWarni
ng:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/us
er_guide/indexing.html#returning-a-view-versus-a-copy
  merge_df.drop_duplicates(subset='Order ID',keep='first',inplace=True)
```

In [167…

```
merge_df
```

Out[167]:

	Order ID	Product_x	Product_y	All Products	All_Product_list
1	176560	Google Phone	Wired Headphones	Google Phone,Wired Headphones	[Google Phone, Wired Headphones]
5	176574	Google Phone	USB-C Charging Cable	Google Phone,USB-C Charging Cable	[Google Phone, USB-C Charging Cable]
13	176586	AAA Batteries (4-pack)	Google Phone	AAA Batteries (4-pack),Google Phone	[AAA Batteries (4-pack), Google Phone]
17	176672	Lightning Charging Cable	USB-C Charging Cable	Lightning Charging Cable,USB-C Charging Cable	[Lightning Charging Cable, USB-C Charging Cable]
21	176681	Apple Airpods Headphones	ThinkPad Laptop	Apple Airpods Headphones,ThinkPad Laptop	[Apple Airpods Headphones, ThinkPad Laptop]
...	...	...	...	...	...
30441	259277	iPhone	Wired Headphones	iPhone,Wired Headphones	[Wired Headphones, iPhone]
30449	259297	iPhone	Lightning Charging Cable	iPhone,Lightning Charging Cable	[Lightning Charging Cable, iPhone]
30458	259303	34in Ultrawide Monitor	AA Batteries (4-pack)	34in Ultrawide Monitor,AA Batteries (4-pack)	[34in Ultrawide Monitor, AA Batteries (4-pack)]
30462	259314	Wired Headphones	AAA Batteries (4-pack)	Wired Headphones,AAA Batteries (4-pack)	[AAA Batteries (4-pack), Wired Headphones]
30466	259350	Google Phone	USB-C Charging Cable	Google Phone,USB-C Charging Cable	[Google Phone, USB-C Charging Cable]

6832 rows × 5 columns

In [178…

```
#Now finding no of times various groups of product bought together.

merge_df['All_Product_list'].value_counts().head(50)
```

Out[178]:
```
[Lightning Charging Cable, iPhone]                    1010
[Google Phone, USB-C Charging Cable]                   996
[Wired Headphones, iPhone]                             375
[USB-C Charging Cable, Vareebadd Phone]                367
[Apple Airpods Headphones, iPhone]                     326
[Google Phone, Wired Headphones]                       314
[Bose SoundSport Headphones, Google Phone]             193
[Vareebadd Phone, Wired Headphones]                    113
[AA Batteries (4-pack), Lightning Charging Cable]      103
[Lightning Charging Cable, USB-C Charging Cable]        96
[AAA Batteries (4-pack), USB-C Charging Cable]          92
[AA Batteries (4-pack), AAA Batteries (4-pack)]         87
[AAA Batteries (4-pack), Wired Headphones]              86
[USB-C Charging Cable, Wired Headphones]                82
[AA Batteries (4-pack), Wired Headphones]               80
[AAA Batteries (4-pack), Lightning Charging Cable]      79
[AAA Batteries (4-pack), Apple Airpods Headphones]      78
[AA Batteries (4-pack), USB-C Charging Cable]           72
[Apple Airpods Headphones, Wired Headphones]            71
[Bose SoundSport Headphones, Lightning Charging Cable]  71
[AA Batteries (4-pack), Apple Airpods Headphones]       70
[Apple Airpods Headphones, Lightning Charging Cable]    69
[Lightning Charging Cable, Wired Headphones]            66
[Bose SoundSport Headphones, Vareebadd Phone]           66
[Apple Airpods Headphones, USB-C Charging Cable]        60
[AAA Batteries (4-pack), Bose SoundSport Headphones]    57
[AA Batteries (4-pack), Bose SoundSport Headphones]     55
[Bose SoundSport Headphones, USB-C Charging Cable]      51
[Apple Airpods Headphones, Bose SoundSport Headphones]  48
[Bose SoundSport Headphones, Wired Headphones]          45
[27in FHD Monitor, AAA Batteries (4-pack)]              43
[27in FHD Monitor, USB-C Charging Cable]                41
[27in FHD Monitor, Lightning Charging Cable]            36
[27in 4K Gaming Monitor, Lightning Charging Cable]      33
[34in Ultrawide Monitor, AA Batteries (4-pack)]         32
[34in Ultrawide Monitor, Lightning Charging Cable]      32
[27in 4K Gaming Monitor, AAA Batteries (4-pack)]        30
[AA Batteries (4-pack), iPhone]                         29
[27in 4K Gaming Monitor, Wired Headphones]              28
[34in Ultrawide Monitor, Wired Headphones]              28
[AAA Batteries (4-pack), ThinkPad Laptop]               27
[AAA Batteries (4-pack), iPhone]                        27
[27in 4K Gaming Monitor, AA Batteries (4-pack)]         26
[27in FHD Monitor, AA Batteries (4-pack)]               26
[20in Monitor, Lightning Charging Cable]                26
[20in Monitor, USB-C Charging Cable]                    25
[34in Ultrawide Monitor, AAA Batteries (4-pack)]        25
[27in FHD Monitor, Bose SoundSport Headphones]          25
[34in Ultrawide Monitor, USB-C Charging Cable]          25
[USB-C Charging Cable, iPhone]                          24
Name: All_Product_list, dtype: int64
```

This will suggest the business to always keep these items together in any store to maximize sales

In [180…
```
import matplotlib.pyplot as plt
```

In [182…
```
final_df
```

Out[182]:

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address	Total Bill	city	state	city_st
0	176558	USB-C Charging Cable	2	11.95	04/19/19 08:46	917 1st St, Dallas, TX 75001	23.90	Dallas	TX	Dallas
2	176559	Bose SoundSport Headphones	1	99.99	04/07/19 22:30	682 Chestnut St, Boston, MA 02215	99.99	Boston	MA	Boston,
3	176560	Google Phone	1	600.00	04/12/19 14:38	669 Spruce St, Los Angeles, CA 90001	600.00	Los Angeles	CA	Angeles
4	176560	Wired Headphones	1	11.99	04/12/19 14:38	669 Spruce St, Los Angeles, CA 90001	11.99	Los Angeles	CA	Angeles
5	176561	Wired Headphones	1	11.99	04/30/19 09:27	333 8th St, Los Angeles, CA 90001	11.99	Los Angeles	CA	Angeles
...	...	...	...	...	...	...	...	...	...	
186845	259353	AAA Batteries (4-pack)	3	2.99	09/17/19 20:56	840 Highland St, Los Angeles, CA 90001	8.97	Los Angeles	CA	Angeles
186846	259354	iPhone	1	700.00	09/01/19 16:00	216 Dogwood St, San Francisco, CA 94016	700.00	San Francisco	CA	Francisco
186847	259355	iPhone	1	700.00	09/23/19 07:39	220 12th St, San Francisco, CA 94016	700.00	San Francisco	CA	Francisco
186848	259356	34in Ultrawide Monitor	1	379.99	09/19/19 17:30	511 Forest St, San Francisco, CA 94016	379.99	San Francisco	CA	Francisco
186849	259357	USB-C Charging Cable	1	11.95	09/30/19 00:18	250 Meadow St, San Francisco, CA 94016	11.95	San Francisco	CA	Francisco

185950 rows × 11 columns

In [184…
```
final_df["Hour_of_Purchase"].unique()
```

Out[184]:
```
array(['08', '22', '14', '09', '13', '07', '10', '17', '12', '19', '15',
       '20', '18', '00', '11', '23', '21', '04', '16', '05', '02', '01',
       '06', '03'], dtype=object)
```

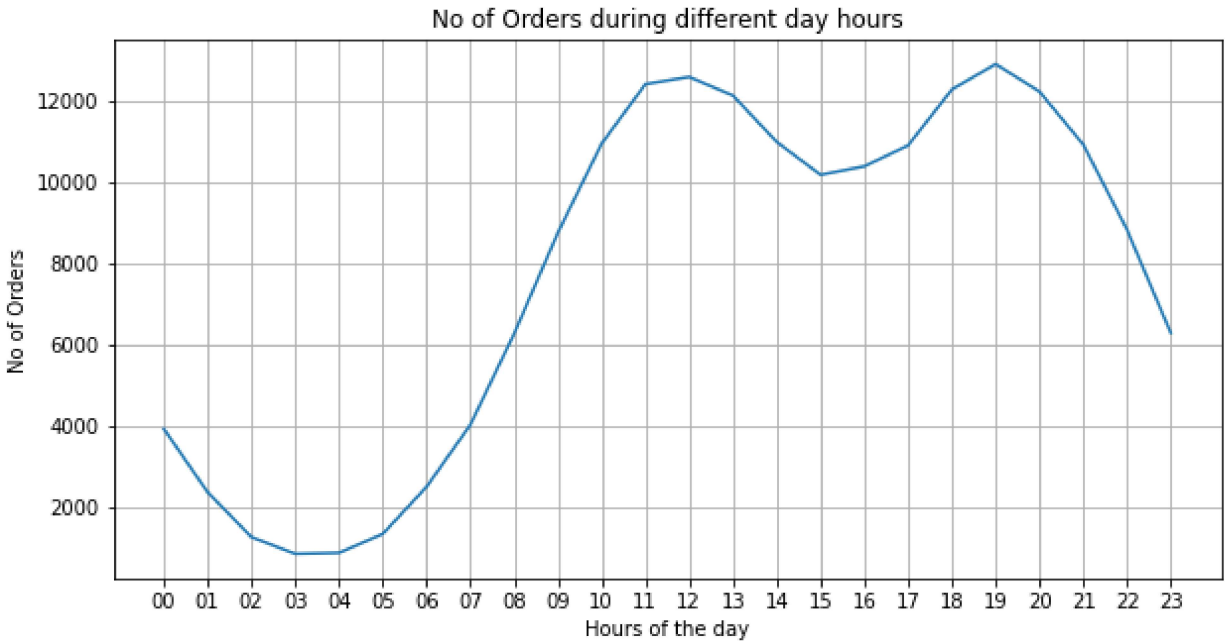
In [186…
```
final_df.groupby(['Hour_of_Purchase'])['Total Bill'].sum()
```

Out[186]:
```
Hour_of_Purchase
00     713721.27
01     460866.88
02     234851.44
03     145757.89
04     162661.01
05     230679.82
06     448113.00
07     744854.12
08    1192348.97
09    1639030.58
10    1944286.77
11    2300610.24
12    2316821.34
13    2155389.80
14    2083672.73
15    1941549.60
16    1904601.31
17    2129361.61
18    2219348.30
19    2412938.54
20    2281716.24
21    2042000.86
22    1607549.21
23    1179304.44
Name: Total Bill, dtype: float64
```

In [188…
```
final_df.groupby(['Hour_of_Purchase'])['Order ID'].count()
```

Out[188]:
```
Hour_of_Purchase
00     3910
01     2350
02     1243
03      831
04      854
05     1321
06     2482
07     4011
08     6256
09     8748
10    10944
11    12411
12    12587
13    12129
14    10984
15    10175
16    10384
17    10899
18    12280
19    12905
20    12228
21    10921
22     8822
23     6275
Name: Order ID, dtype: int64
```

In [615…
```
plt.figure(figsize=(10,5))
plt.plot(final_df.groupby(['Hour_of_Purchase'])['Order ID'].count())
plt.title('No of Orders during different day hours')
plt.xlabel('Hours of the day')
plt.ylabel('No of Orders')
plt.grid()
plt.show()
```



In [224…
```
final_df.head(20)
```

Out[224]:

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address	Total Bill	city	state	city_state
0	176558	USB-C Charging Cable	2	11.95	04/19/19 08:46	917 1st St, Dallas, TX 75001	23.90	Dallas	TX	Dallas,TX
2	176559	Bose SoundSport Headphones	1	99.99	04/07/19 22:30	682 Chestnut St, Boston, MA 02215	99.99	Boston	MA	Boston,MA
3	176560	Google Phone	1	600.00	04/12/19 14:38	669 Spruce St, Los Angeles, CA 90001	600.00	Los Angeles	CA	Los Angeles,CA
4	176560	Wired Headphones	1	11.99	04/12/19 14:38	669 Spruce St, Los Angeles, CA 90001	11.99	Los Angeles	CA	Los Angeles,CA
5	176561	Wired Headphones	1	11.99	04/30/19 09:27	333 8th St, Los Angeles, CA 90001	11.99	Los Angeles	CA	Los Angeles,CA
6	176562	USB-C Charging Cable	1	11.95	04/29/19 13:03	381 Wilson St, San Francisco, CA 94016	11.95	San Francisco	CA	San Francisco,CA
7	176563	Bose SoundSport Headphones	1	99.99	04/02/19 07:46	668 Center St, Seattle, WA 98101	99.99	Seattle	WA	Seattle,WA
8	176564	USB-C Charging Cable	1	11.95	04/12/19 10:58	790 Ridge St, Atlanta, GA 30301	11.95	Atlanta	GA	Atlanta,GA
9	176565	Macbook Pro Laptop	1	1700.00	04/24/19 10:38	915 Willow St, San Francisco, CA 94016	1700.00	San Francisco	CA	San Francisco,CA
10	176566	Wired Headphones	1	11.99	04/08/19 14:05	83 7th St, Boston, MA 02215	11.99	Boston	MA	Boston,MA
11	176567	Google Phone	1	600.00	04/18/19 17:18	444 7th St, Los Angeles, CA 90001	600.00	Los Angeles	CA	Los Angeles,CA

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address	Total Bill	city	state	city_state
12	176568	Lightning Charging Cable	1	14.95	04/15/19 12:18	438 Elm St, Seattle, WA 98101	14.95	Seattle	WA	Seattle,WA
13	176569	27in 4K Gaming Monitor	1	389.99	04/16/19 19:23	657 Hill St, Dallas, TX 75001	389.99	Dallas	TX	Dallas,TX
14	176570	AA Batteries (4-pack)	1	3.84	04/22/19 15:09	186 12th St, Dallas, TX 75001	3.84	Dallas	TX	Dallas,TX
15	176571	Lightning Charging Cable	1	14.95	04/19/19 14:29	253 Johnson St, Atlanta, GA 30301	14.95	Atlanta	GA	Atlanta,GA
16	176572	Apple Airpods Headphones	1	150.00	04/04/19 20:30	149 Dogwood St, New York City, NY 10001	150.00	New York City	NY	New York City,NY
17	176573	USB-C Charging Cable	1	11.95	04/27/19 18:41	214 Chestnut St, San Francisco, CA 94016	11.95	San Francisco	CA	San Francisco,CA
18	176574	Google Phone	1	600.00	04/03/19 19:42	20 Hill St, Los Angeles, CA 90001	600.00	Los Angeles	CA	Los Angeles,CA
19	176574	USB-C Charging Cable	1	11.95	04/03/19 19:42	20 Hill St, Los Angeles, CA 90001	11.95	Los Angeles	CA	Los Angeles,CA
20	176575	AAA Batteries (4-pack)	1	2.99	04/27/19 00:30	433 Hill St, New York City, NY 10001	2.99	New York City	NY	New York City,NY

In [388…
```
#creating a series P1 which gives sum of different product sold in different city

P=final_df.groupby(['city','Product'])['Total Bill'].sum()
```

In [479…
```
P
```

Out[479]:
```
city     Product
Atlanta  20in Monitor               37616.58
         27in 4K Gaming Monitor    192265.07
         27in FHD Monitor           88194.12
         34in Ultrawide Monitor    183155.18
         AA Batteries (4-pack)       8421.12
                                       ...
Seattle  ThinkPad Laptop           332996.67
         USB-C Charging Cable       22334.55
         Vareebadd Phone            71600.00
         Wired Headphones           19807.48
         iPhone                    382200.00
Name: Total Bill, Length: 171, dtype: float64
```

In [480…
```
#so checking the which product made the maximum sale in Atlanta

P['Atlanta'][P1['Atlanta']==P1['Atlanta'].max()]
```

Out[480]:
```
Product
Macbook Pro Laptop    644300.0
Name: Total Bill, dtype: float64
```

In [410…
```
#Similarly for city Boston

P['Boston'][P1['Boston']==P1['Boston'].max()]
```

Out[410]:
```
Product
Macbook Pro Laptop    814300.0
Name: Total Bill, dtype: float64
```

In [483…
```
#Converting the Series P to a dataframe Q

Q=P.reset_index()
```

In [484…
```
Q
```

Out[484]:

	city	Product	Total Bill
0	Atlanta	20in Monitor	37616.58
1	Atlanta	27in 4K Gaming Monitor	192265.07
2	Atlanta	27in FHD Monitor	88194.12
3	Atlanta	34in Ultrawide Monitor	183155.18
4	Atlanta	AA Batteries (4-pack)	8421.12
...	...	...	...
166	Seattle	ThinkPad Laptop	332996.67
167	Seattle	USB-C Charging Cable	22334.55
168	Seattle	Vareebadd Phone	71600.00
169	Seattle	Wired Headphones	19807.48
170	Seattle	iPhone	382200.00

171 rows × 3 columns

In [485…
```
#adding a new column 'max_sales' which is partition by city and show maximum value for

Q['max_sales']=Q.groupby(['city'])['Total Bill'].transform('max')
```

In [492…
```
Q
```

Out[492]:

	city	Product	Total Bill	max_sales
0	Atlanta	20in Monitor	37616.58	644300.0
1	Atlanta	27in 4K Gaming Monitor	192265.07	644300.0
2	Atlanta	27in FHD Monitor	88194.12	644300.0
3	Atlanta	34in Ultrawide Monitor	183155.18	644300.0
4	Atlanta	AA Batteries (4-pack)	8421.12	644300.0
...	...	...	...	...
166	Seattle	ThinkPad Laptop	332996.67	605200.0
167	Seattle	USB-C Charging Cable	22334.55	605200.0
168	Seattle	Vareebadd Phone	71600.00	605200.0
169	Seattle	Wired Headphones	19807.48	605200.0
170	Seattle	iPhone	382200.00	605200.0

171 rows × 4 columns

In [254…
```
#Comparing column Total Bill and max_sales of dataframe Q which given true for each co

Q['Total Bill']==Q['max_sales']
```

In [589…
```
# Masking the above boolean series to the dataframe Q

Q[Q['Total Bill']==Q['max_sales']]
```

Out[589]:

	index	city	Product	Total Bill	max_sales
0	13	Atlanta	Macbook Pro Laptop	644300.0	644300.0
1	32	Austin	Macbook Pro Laptop	426700.0	426700.0
2	51	Boston	Macbook Pro Laptop	814300.0	814300.0
3	70	Dallas	Macbook Pro Laptop	649400.0	649400.0
4	89	Los Angeles	Macbook Pro Laptop	1276700.0	1276700.0
5	108	New York City	Macbook Pro Laptop	1116900.0	1116900.0
6	127	Portland	Macbook Pro Laptop	572900.0	572900.0
7	146	San Francisco	Macbook Pro Laptop	1931200.0	1931200.0
8	165	Seattle	Macbook Pro Laptop	605200.0	605200.0

In [591…
```
# Masking the above boolean series to the dataframe Q

Q[Q['Total Bill']==Q['max_sales']].reset_index().drop(['index'],axis=1)
```

Out[591]:

	city	Product	Total Bill	max_sales
0	Atlanta	Macbook Pro Laptop	644300.0	644300.0
1	Austin	Macbook Pro Laptop	426700.0	426700.0
2	Boston	Macbook Pro Laptop	814300.0	814300.0
3	Dallas	Macbook Pro Laptop	649400.0	649400.0
4	Los Angeles	Macbook Pro Laptop	1276700.0	1276700.0
5	New York City	Macbook Pro Laptop	1116900.0	1116900.0
6	Portland	Macbook Pro Laptop	572900.0	572900.0
7	San Francisco	Macbook Pro Laptop	1931200.0	1931200.0
8	Seattle	Macbook Pro Laptop	605200.0	605200.0

The above Dataframe shows every cities max selling product on basis of total sales

In [592…
```
Q1=Q[Q['Total Bill']==Q['max_sales']].reset_index().drop(['index'],axis=1)
```

In [508…
```
Q[Q['Total Bill']==Q['max_sales']]['city']
```

Out[508]:
```
13           Atlanta
32            Austin
51            Boston
70            Dallas
89       Los Angeles
108    New York City
127         Portland
146    San Francisco
165          Seattle
Name: city, dtype: object
```

In [510…
```
Q[Q['Total Bill']==Q['max_sales']]['max_sales']

#assigning the above series to y
```

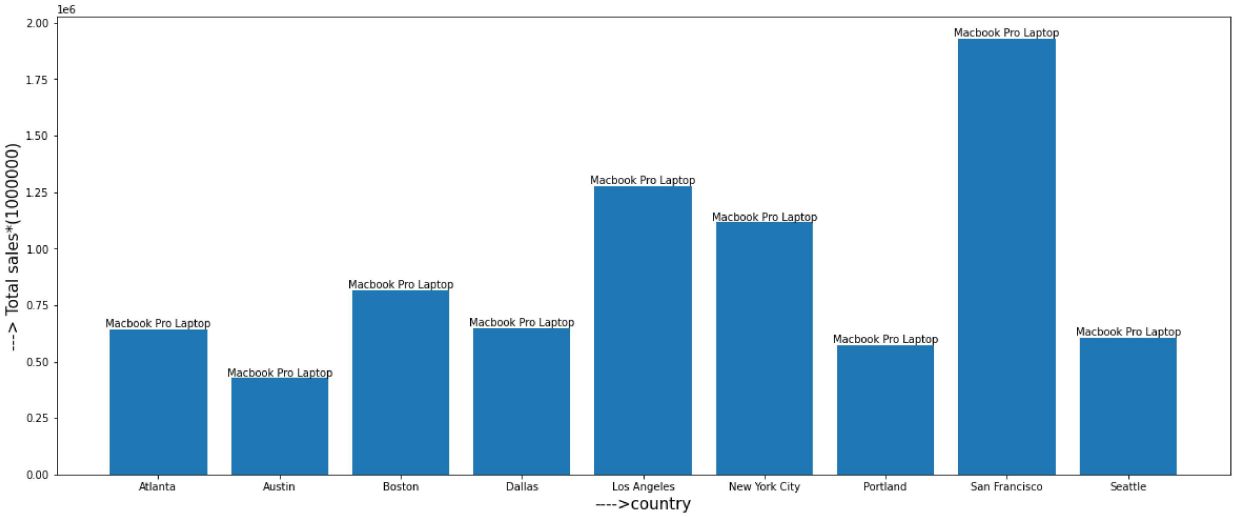
Out[510]:
```
13      644300.0
32      426700.0
51      814300.0
70      649400.0
89     1276700.0
108    1116900.0
127     572900.0
146    1931200.0
165     605200.0
Name: max_sales, dtype: float64
```

In [595…
```
Q1['Product'][0]
```

Out[595]:
```
'Macbook Pro Laptop'
```

In [596…

```
f = plt.figure(figsize=(20,8))
x=Q[Q['Total Bill']==Q['max_sales']]['city'].reset_index().drop(['index'],axis=1)['cit
y=Q[Q['Total Bill']==Q['max_sales']]['max_sales'].reset_index().drop(['index'],axis=1)
plt.bar(x,y)
plt.xlabel('---->country', fontsize=15)
plt.ylabel('---> Total sales*(1000000)',fontsize=15)
for i in range(len(x)):
    plt.text(i,y[i],Q1['Product'][i],ha='center',va='bottom')

plt.show()
```



So, in every city 'Macbook Pro Laptop' was the product we contributed in maximum sales

In [604…

```
final_df
```

Out[604]:

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address	Total Bill	city	state	city_st
0	176558	USB-C Charging Cable	2	11.95	04/19/19 08:46	917 1st St, Dallas, TX 75001	23.90	Dallas	TX	Dallas
2	176559	Bose SoundSport Headphones	1	99.99	04/07/19 22:30	682 Chestnut St, Boston, MA 02215	99.99	Boston	MA	Boston,
3	176560	Google Phone	1	600.00	04/12/19 14:38	669 Spruce St, Los Angeles, CA 90001	600.00	Los Angeles	CA	Angeles
4	176560	Wired Headphones	1	11.99	04/12/19 14:38	669 Spruce St, Los Angeles, CA 90001	11.99	Los Angeles	CA	Angeles
5	176561	Wired Headphones	1	11.99	04/30/19 09:27	333 8th St, Los Angeles, CA 90001	11.99	Los Angeles	CA	Angeles
...	...	...	...	...	...	...	...	...	...	
186845	259353	AAA Batteries (4-pack)	3	2.99	09/17/19 20:56	840 Highland St, Los Angeles, CA 90001	8.97	Los Angeles	CA	Angeles
186846	259354	iPhone	1	700.00	09/01/19 16:00	216 Dogwood St, San Francisco, CA 94016	700.00	San Francisco	CA	Francisco
186847	259355	iPhone	1	700.00	09/23/19 07:39	220 12th St, San Francisco, CA 94016	700.00	San Francisco	CA	Francisco
186848	259356	34in Ultrawide Monitor	1	379.99	09/19/19 17:30	511 Forest St, San Francisco, CA 94016	379.99	San Francisco	CA	Francisco
186849	259357	USB-C Charging Cable	1	11.95	09/30/19 00:18	250 Meadow St, San Francisco, CA 94016	11.95	San Francisco	CA	Francisco

185950 rows × 11 columns

In [ ]:

In [ ]:

In [ ]:

In [ ]: