SALES ANALYSIS

Import required libraries

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

In [2]:
```
import glob
```

In [3]:
```
import os
```

In [4]:
```
import numpy as np
```

Merge csv files into single file

In [5]:
```
files = os.path.join("/Users/ilker/Pandas-Data-Science-Tasks-master/Sales
```

In [6]:
```
files = glob.glob(files)
```

In [7]:
```
salesData = pd.concat(map(pd.read_csv, files), ignore_index=True)
```

In [8]:
```
newData = salesData.to_csv("Sales.csv")
```

In [9]:
```
salesData.head(200)
```

Out[9]:

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address
0	295665	Macbook Pro Laptop	1	1700	12/30/19 00:01	136 Church St, New York City, NY 10001
1	295666	LG Washing Machine	1	600.0	12/29/19 07:03	562 2nd St, New York City, NY 10001
2	295667	USB-C Charging Cable	1	11.95	12/12/19 18:21	277 Main St, New York City, NY 10001
3	295668	27in FHD Monitor	1	149.99	12/22/19 15:13	410 6th St, San Francisco, CA 94016
4	295669	USB-C Charging Cable	1	11.95	12/18/19 12:38	43 Hill St, Atlanta, GA 30301
...	...	...	...	...	...	...
195	295843	Bose SoundSport Headphones	1	99.99	12/19/19 14:02	700 West St, San Francisco, CA 94016
196	295844	AA Batteries (4-pack)	1	3.84	12/31/19 17:51	663 12th St, Atlanta, GA 30301
197	295845	Lightning Charging Cable	1	14.95	12/20/19 09:53	955 Sunset St, San Francisco, CA 94016
198	295846	Flatscreen TV	1	300	12/27/19 19:22	453 Johnson St, Atlanta, GA 30301
199	295847	34in Ultrawide Monitor	1	379.99	12/13/19 15:56	40 Madison St, Dallas, TX 75001

200 rows × 6 columns

In [10]:

```
salesData["Month"] = salesData["Order Date"].str[0:2]
```

Clean up Data

In [11]:

```
salesData.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 186850 entries, 0 to 186849
Data columns (total 7 columns):
 #   Column            Non-Null Count   Dtype 
---  ------            --------------   ----- 
 0   Order ID          186305 non-null  object
 1   Product           186305 non-null  object
 2   Quantity Ordered  186305 non-null  object
 3   Price Each        186305 non-null  object
 4   Order Date        186305 non-null  object
 5   Purchase Address  186305 non-null  object
 6   Month             186305 non-null  object
dtypes: object(7)
memory usage: 10.0+ MB
```

In [12]:

```
newSalesData = salesData.dropna(how = "any")
```

In [13]:
```
newSdata = newSalesData[newSalesData["Order Date"].str[0:2] != "Or"]
```

In [14]:
```
newSdata
```

Out[14]:

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address	Month
0	295665	Macbook Pro Laptop	1	1700	12/30/19 00:01	136 Church St, New York City, NY 10001	12
1	295666	LG Washing Machine	1	600.0	12/29/19 07:03	562 2nd St, New York City, NY 10001	12
2	295667	USB-C Charging Cable	1	11.95	12/12/19 18:21	277 Main St, New York City, NY 10001	12
3	295668	27in FHD Monitor	1	149.99	12/22/19 15:13	410 6th St, San Francisco, CA 94016	12
4	295669	USB-C Charging Cable	1	11.95	12/18/19 12:38	43 Hill St, Atlanta, GA 30301	12
...	...	...	...	...	...	...	...
186845	222905	AAA Batteries (4-pack)	1	2.99	06/07/19 19:02	795 Pine St, Boston, MA 02215	06
186846	222906	27in FHD Monitor	1	149.99	06/01/19 19:29	495 North St, New York City, NY 10001	06
186847	222907	USB-C Charging Cable	1	11.95	06/22/19 18:57	319 Ridge St, San Francisco, CA 94016	06
186848	222908	USB-C Charging Cable	1	11.95	06/26/19 18:35	916 Main St, San Francisco, CA 94016	06
186849	222909	AAA Batteries (4-pack)	1	2.99	06/25/19 14:33	209 11th St, Atlanta, GA 30301	06

185950 rows × 7 columns

In [15]:
```
newSdata["Month"].astype("int")
```

```
Out[15]: 0         12
         1         12
         2         12
         3         12
         4         12
                   ..
         186845     6
         186846     6
         186847     6
         186848     6
         186849     6
         Name: Month, Length: 185950, dtype: int64
```

Q1 : What was the best month for sales ? How much was earned that month ?

```
In [16]: month_group = newSdata.groupby(["Month"])
```

```
In [17]: month_group.get_group("12")
```

Out[17]:

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address	Month
0	295665	Macbook Pro Laptop	1	1700	12/30/19 00:01	136 Church St, New York City, NY 10001	12
1	295666	LG Washing Machine	1	600.0	12/29/19 07:03	562 2nd St, New York City, NY 10001	12
2	295667	USB-C Charging Cable	1	11.95	12/12/19 18:21	277 Main St, New York City, NY 10001	12
3	295668	27in FHD Monitor	1	149.99	12/22/19 15:13	410 6th St, San Francisco, CA 94016	12
4	295669	USB-C Charging Cable	1	11.95	12/18/19 12:38	43 Hill St, Atlanta, GA 30301	12
...	...	...	...	...	...	...	...
115313	293993	AAA Batteries (4-pack)	1	2.99	12/01/19 00:41	999 10th St, San Francisco, CA 94016	12
115418	294093	USB-C Charging Cable	1	11.95	12/01/19 00:37	51 Cherry St, New York City, NY 10001	12
115783	294441	Wired Headphones	1	11.99	12/01/19 00:52	469 Sunset St, San Francisco, CA 94016	12
115980	294629	Wired Headphones	1	11.99	12/01/19 01:54	668 Ridge St, San Francisco, CA 94016	12
117040	295636	AA Batteries (4-pack)	2	3.84	12/01/19 00:20	204 Walnut St, Dallas, TX 75001	12

24984 rows × 7 columns

Convert columns the correct type

In [18]:

```
newSdata['Quantity Ordered'] = pd.to_numeric(newSdata['Quantity Ordered']
newSdata['Price Each'] = pd.to_numeric(newSdata['Price Each'])
```

```
/var/folders/1m/24t_f0_j3bb46tzcw1s7w04w0000gp/T/ipykernel_8729/29649400
57.py:1: 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-d
ocs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  newSdata['Quantity Ordered'] = pd.to_numeric(newSdata['Quantity Ordere
d'])
/var/folders/1m/24t_f0_j3bb46tzcw1s7w04w0000gp/T/ipykernel_8729/29649400
57.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-d
ocs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  newSdata['Price Each'] = pd.to_numeric(newSdata['Price Each'])
```

```
In [19]: newSdata['Price Each'].dtype
```

```
Out[19]: dtype('float64')
```

Add sales column

```
In [20]: newSdata["Sales"] = newSdata['Quantity Ordered'] * newSdata["Price Each"]
```

```
/var/folders/1m/24t_f0_j3bb46tzcw1s7w04w0000gp/T/ipykernel_8729/15885602
07.py:1: 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-d
ocs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  newSdata["Sales"] = newSdata['Quantity Ordered'] * newSdata["Price Eac
h"]
```

```
In [21]: newSdata
```

Out[21]:

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address	Month	Sales
0	295665	Macbook Pro Laptop	1	1700.00	12/30/19 00:01	136 Church St, New York City, NY 10001	12	1700.00
1	295666	LG Washing Machine	1	600.00	12/29/19 07:03	562 2nd St, New York City, NY 10001	12	600.00
2	295667	USB-C Charging Cable	1	11.95	12/12/19 18:21	277 Main St, New York City, NY 10001	12	11.95
3	295668	27in FHD Monitor	1	149.99	12/22/19 15:13	410 6th St, San Francisco, CA 94016	12	149.99
4	295669	USB-C Charging Cable	1	11.95	12/18/19 12:38	43 Hill St, Atlanta, GA 30301	12	11.95
...	...	...	...	...	...	...	...	...
186845	222905	AAA Batteries (4-pack)	1	2.99	06/07/19 19:02	795 Pine St, Boston, MA 02215	06	2.99
186846	222906	27in FHD Monitor	1	149.99	06/01/19 19:29	495 North St, New York City, NY 10001	06	149.99
186847	222907	USB-C Charging Cable	1	11.95	06/22/19 18:57	319 Ridge St, San Francisco, CA 94016	06	11.95
186848	222908	USB-C Charging Cable	1	11.95	06/26/19 18:35	916 Main St, San Francisco, CA 94016	06	11.95
186849	222909	AAA Batteries (4-pack)	1	2.99	06/25/19 14:33	209 11th St, Atlanta, GA 30301	06	2.99

185950 rows × 8 columns

In [22]:

```
newSdata
```

Out[22]:

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address	Month	Sales
0	295665	Macbook Pro Laptop	1	1700.00	12/30/19 00:01	136 Church St, New York City, NY 10001	12	1700.00
1	295666	LG Washing Machine	1	600.00	12/29/19 07:03	562 2nd St, New York City, NY 10001	12	600.00
2	295667	USB-C Charging Cable	1	11.95	12/12/19 18:21	277 Main St, New York City, NY 10001	12	11.95
3	295668	27in FHD Monitor	1	149.99	12/22/19 15:13	410 6th St, San Francisco, CA 94016	12	149.99
4	295669	USB-C Charging Cable	1	11.95	12/18/19 12:38	43 Hill St, Atlanta, GA 30301	12	11.95
...	...	...	...	...	...	...	...	...
186845	222905	AAA Batteries (4-pack)	1	2.99	06/07/19 19:02	795 Pine St, Boston, MA 02215	06	2.99
186846	222906	27in FHD Monitor	1	149.99	06/01/19 19:29	495 North St, New York City, NY 10001	06	149.99
186847	222907	USB-C Charging Cable	1	11.95	06/22/19 18:57	319 Ridge St, San Francisco, CA 94016	06	11.95
186848	222908	USB-C Charging Cable	1	11.95	06/26/19 18:35	916 Main St, San Francisco, CA 94016	06	11.95
186849	222909	AAA Batteries (4-pack)	1	2.99	06/25/19 14:33	209 11th St, Atlanta, GA 30301	06	2.99

185950 rows × 8 columns

In [23]:

```
SalesGr = newSdata.groupby(["Month"]).sum()
```

```
/var/folders/1m/24t_f0_j3bb46tzcw1s7w04w0000gp/T/ipykernel_8729/16422665
90.py:1: FutureWarning: The default value of numeric_only in DataFrameGr
oupBy.sum is deprecated. In a future version, numeric_only will default
to False. Either specify numeric_only or select only columns which shoul
d be valid for the function.
  SalesGr = newSdata.groupby(["Month"]).sum()
```

In [24]:

```
SalesGr
```

Out[24]:

Month	Quantity Ordered	Price Each	Sales
01	10903	1811768.38	1822256.73
02	13449	2188884.72	2202022.42
03	17005	2791207.83	2807100.38
04	20558	3367671.02	3390670.24
05	18667	3135125.13	3152606.75
06	15253	2562025.61	2577802.26
07	16072	2632539.56	2647775.76
08	13448	2230345.42	2244467.88
09	13109	2084992.09	2097560.13
10	22703	3715554.83	3736726.88
11	19798	3180600.68	3199603.20
12	28114	4588415.41	4613443.34

In [25]:

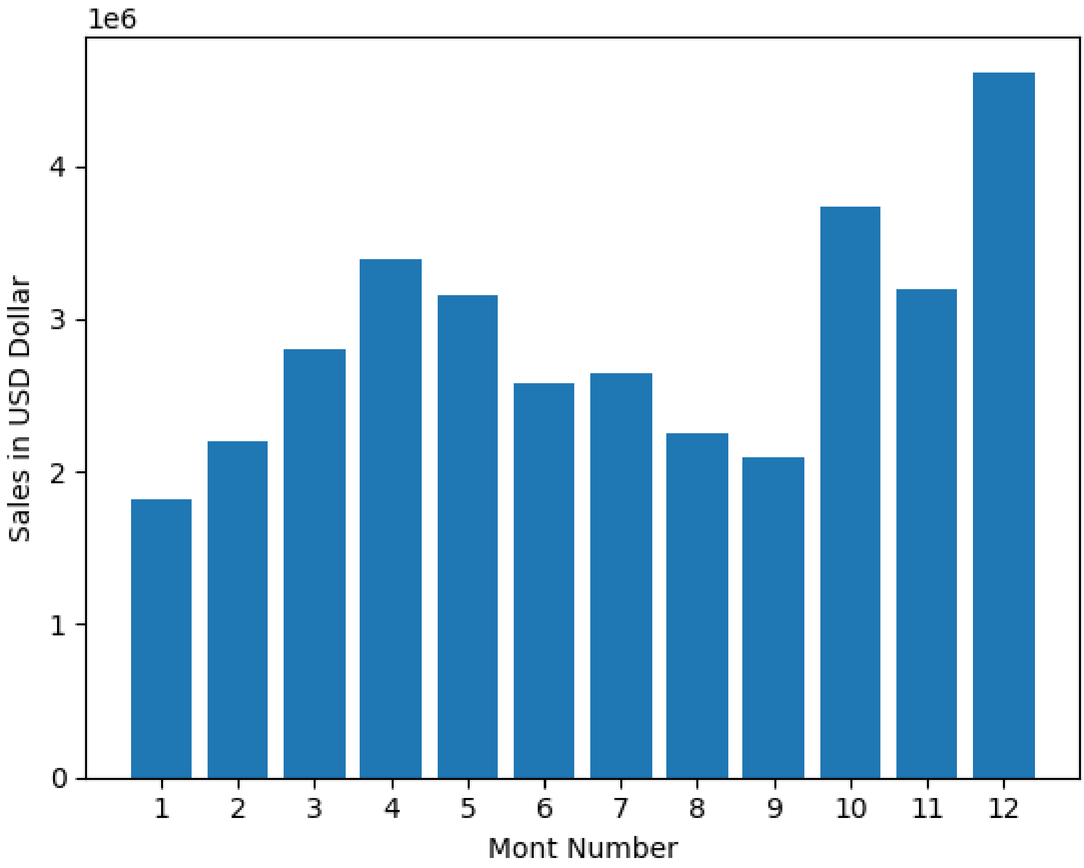
```
import matplotlib.pyplot as plt
```

In [26]:

```
month =range(1,13)
```

In [27]:

```
plt.bar(month,SalesGr["Sales"])
plt.xticks(month)
plt.ylabel("Sales in USD Dollar")
plt.xlabel("Mont Number")
plt.show()
```



Q2 What city had the highest number of sales ?

task: add city column

In [28]:

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

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



newSdata["city"] = newSdata["Purchase Address"].apply(lambda x : get_city
```

```
/var/folders/1m/24t_f0_j3bb46tzcw1s7w04w0000gp/T/ipykernel_8729/30707175
4.py:9: 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-d
ocs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  newSdata["city"] = newSdata["Purchase Address"].apply(lambda x : get_c
ity(x)+" "+ get_state(x))
```

In [29]:

```
newSdata
```

Out[29]:

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address	Month	Sales	
0	295665	Macbook Pro Laptop	1	1700.00	12/30/19 00:01	136 Church St, New York City, NY 10001	12	1700.00	New Cit
1	295666	LG Washing Machine	1	600.00	12/29/19 07:03	562 2nd St, New York City, NY 10001	12	600.00	New Cit
2	295667	USB-C Charging Cable	1	11.95	12/12/19 18:21	277 Main St, New York City, NY 10001	12	11.95	New Cit
3	295668	27in FHD Monitor	1	149.99	12/22/19 15:13	410 6th St, San Francisco, CA 94016	12	149.99	Franc
4	295669	USB-C Charging Cable	1	11.95	12/18/19 12:38	43 Hill St, Atlanta, GA 30301	12	11.95	At
...	...	...	...	...	...	...	...	...	
186845	222905	AAA Batteries (4-pack)	1	2.99	06/07/19 19:02	795 Pine St, Boston, MA 02215	06	2.99	Bc
186846	222906	27in FHD Monitor	1	149.99	06/01/19 19:29	495 North St, New York City, NY 10001	06	149.99	New Cit
186847	222907	USB-C Charging Cable	1	11.95	06/22/19 18:57	319 Ridge St, San Francisco, CA 94016	06	11.95	Franc
186848	222908	USB-C Charging Cable	1	11.95	06/26/19 18:35	916 Main St, San Francisco, CA 94016	06	11.95	Franc
186849	222909	AAA Batteries (4-pack)	1	2.99	06/25/19 14:33	209 11th St, Atlanta, GA 30301	06	2.99	At

185950 rows × 9 columns

In [30]:

```
results = newSdata.groupby(["city"]).sum()
```

```
/var/folders/1m/24t_f0_j3bb46tzcw1s7w04w0000gp/T/ipykernel_8729/41448368
96.py:1: FutureWarning: The default value of numeric_only in DataFrameGr
oupBy.sum is deprecated. In a future version, numeric_only will default
to False. Either specify numeric_only or select only columns which shoul
d be valid for the function.
  results = newSdata.groupby(["city"]).sum()
```

In [31]:
```
results
```

Out[31]:

	Quantity Ordered	Price Each	Sales
city			
Atlanta GA	16602	2779908.20	2795498.58
Austin TX	11153	1809873.61	1819581.75
Boston MA	22528	3637409.77	3661642.01
Dallas TX	16730	2752627.82	2767975.40
Los Angeles CA	33289	5421435.23	5452570.80
New York City NY	27932	4635370.83	4664317.43
Portland ME	2750	447189.25	449758.27
Portland OR	11303	1860558.22	1870732.34
San Francisco CA	50239	8211461.74	8262203.91
Seattle WA	16553	2733296.01	2747755.48

In [32]:
```
results.index
```

Out[32]:
```
Index([' Atlanta GA', ' Austin TX', ' Boston MA', ' Dallas TX',
       ' Los Angeles CA', ' New York City NY', ' Portland ME', ' Portlan
d OR',
       ' San Francisco CA', ' Seattle WA'],
      dtype='object', name='city')
```

In [33]:
```
newSdata
```

Out[33]:

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address	Month	Sales	
0	295665	Macbook Pro Laptop	1	1700.00	12/30/19 00:01	136 Church St, New York City, NY 10001	12	1700.00	New Cit
1	295666	LG Washing Machine	1	600.00	12/29/19 07:03	562 2nd St, New York City, NY 10001	12	600.00	New Cit
2	295667	USB-C Charging Cable	1	11.95	12/12/19 18:21	277 Main St, New York City, NY 10001	12	11.95	New Cit
3	295668	27in FHD Monitor	1	149.99	12/22/19 15:13	410 6th St, San Francisco, CA 94016	12	149.99	Fran
4	295669	USB-C Charging Cable	1	11.95	12/18/19 12:38	43 Hill St, Atlanta, GA 30301	12	11.95	At
...	...	...	...	...	...	...	...	...	
186845	222905	AAA Batteries (4-pack)	1	2.99	06/07/19 19:02	795 Pine St, Boston, MA 02215	06	2.99	Bo
186846	222906	27in FHD Monitor	1	149.99	06/01/19 19:29	495 North St, New York City, NY 10001	06	149.99	New Cit
186847	222907	USB-C Charging Cable	1	11.95	06/22/19 18:57	319 Ridge St, San Francisco, CA 94016	06	11.95	Fran
186848	222908	USB-C Charging Cable	1	11.95	06/26/19 18:35	916 Main St, San Francisco, CA 94016	06	11.95	Fran
186849	222909	AAA Batteries (4-pack)	1	2.99	06/25/19 14:33	209 11th St, Atlanta, GA 30301	06	2.99	At

185950 rows × 9 columns

In [34]:

```
cities = [city for city, df in newSdata.groupby("city")]
```

In [35]:

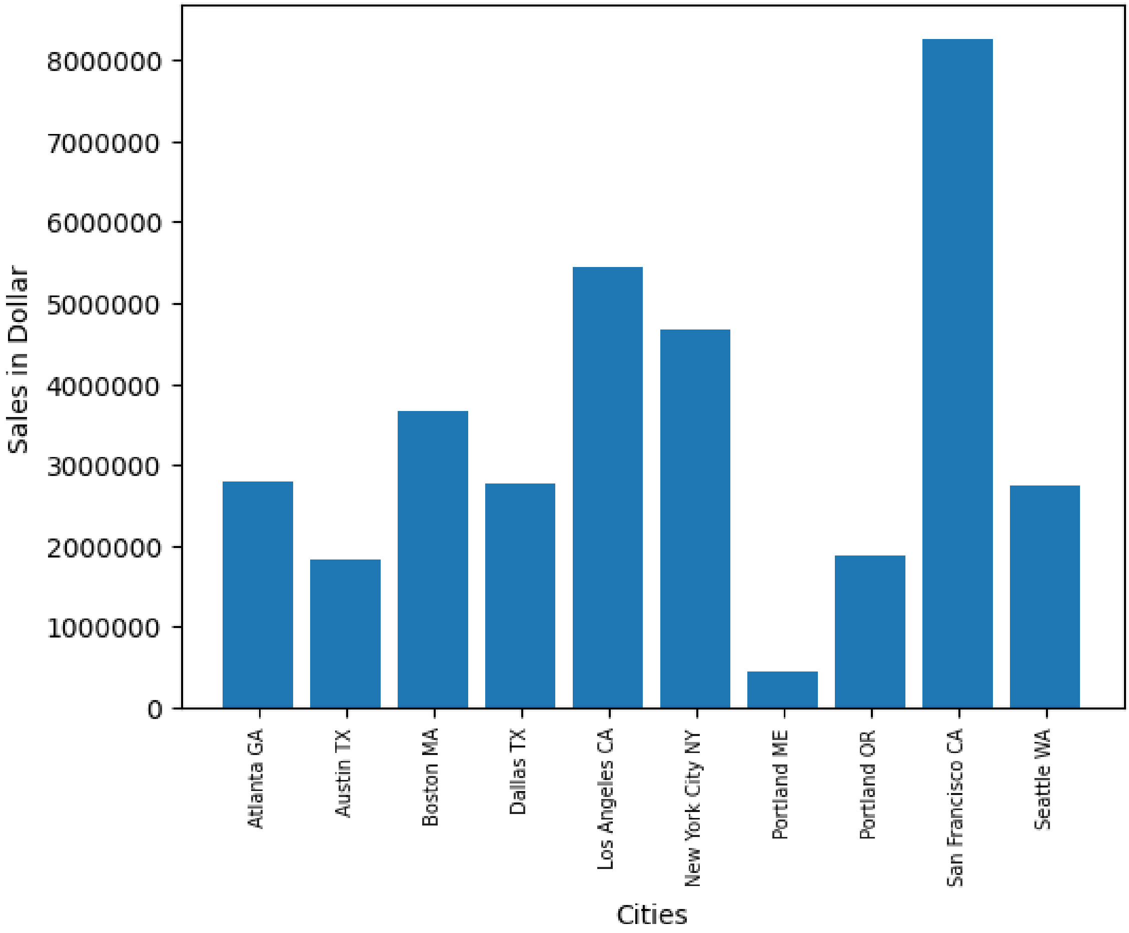
```
cities
```

```
Out[35]: [' Atlanta GA',
          ' Austin TX',
          ' Boston MA',
          ' Dallas TX',
          ' Los Angeles CA',
          ' New York City NY',
          ' Portland ME',
          ' Portland OR',
          ' San Francisco CA',
          ' Seattle WA']
```

```
In [36]: plt.ticklabel_format(style='plain')
         plt.bar(cities, results["Sales"])
         plt.xticks(cities,rotation = "vertical", size = 7)

         plt.xlabel("Cities")
         plt.ylabel("Sales in Dollar")
```

```
Out[36]: Text(0, 0.5, 'Sales in Dollar')
```



Q3:What time should we display advertisement to maximize likelihood of customers buying product ?

```
In [37]: newSdata
```

Out[37]:

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address	Month	Sales	
0	295665	Macbook Pro Laptop	1	1700.00	12/30/19 00:01	136 Church St, New York City, NY 10001	12	1700.00	New Cit
1	295666	LG Washing Machine	1	600.00	12/29/19 07:03	562 2nd St, New York City, NY 10001	12	600.00	New Cit
2	295667	USB-C Charging Cable	1	11.95	12/12/19 18:21	277 Main St, New York City, NY 10001	12	11.95	New Cit
3	295668	27in FHD Monitor	1	149.99	12/22/19 15:13	410 6th St, San Francisco, CA 94016	12	149.99	Franc
4	295669	USB-C Charging Cable	1	11.95	12/18/19 12:38	43 Hill St, Atlanta, GA 30301	12	11.95	At
...	...	...	...	...	...	...	...	...	
186845	222905	AAA Batteries (4-pack)	1	2.99	06/07/19 19:02	795 Pine St, Boston, MA 02215	06	2.99	Bo
186846	222906	27in FHD Monitor	1	149.99	06/01/19 19:29	495 North St, New York City, NY 10001	06	149.99	New Cit
186847	222907	USB-C Charging Cable	1	11.95	06/22/19 18:57	319 Ridge St, San Francisco, CA 94016	06	11.95	Franc
186848	222908	USB-C Charging Cable	1	11.95	06/26/19 18:35	916 Main St, San Francisco, CA 94016	06	11.95	Franc
186849	222909	AAA Batteries (4-pack)	1	2.99	06/25/19 14:33	209 11th St, Atlanta, GA 30301	06	2.99	At

185950 rows × 9 columns

In [38]:

```
newSdata["Order Date"] = pd.to_datetime(newSdata["Order Date"])
```

```
/var/folders/1m/24t_f0_j3bb46tzcw1s7w04w0000gp/T/ipykernel_8729/33258907
02.py:1: 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-d
ocs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  newSdata["Order Date"] = pd.to_datetime(newSdata["Order Date"])
```

In [39]:

```
newSdata
```

Out[39]:

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address	Month	Sales	
0	295665	Macbook Pro Laptop	1	1700.00	2019-12-30 00:01:00	136 Church St, New York City, NY 10001	12	1700.00	New Cit
1	295666	LG Washing Machine	1	600.00	2019-12-29 07:03:00	562 2nd St, New York City, NY 10001	12	600.00	New Cit
2	295667	USB-C Charging Cable	1	11.95	2019-12-12 18:21:00	277 Main St, New York City, NY 10001	12	11.95	New Cit
3	295668	27in FHD Monitor	1	149.99	2019-12-22 15:13:00	410 6th St, San Francisco, CA 94016	12	149.99	Fran
4	295669	USB-C Charging Cable	1	11.95	2019-12-18 12:38:00	43 Hill St, Atlanta, GA 30301	12	11.95	At
...	...	...	...	...	...	...	...	...	
186845	222905	AAA Batteries (4-pack)	1	2.99	2019-06-07 19:02:00	795 Pine St, Boston, MA 02215	06	2.99	Bo
186846	222906	27in FHD Monitor	1	149.99	2019-06-01 19:29:00	495 North St, New York City, NY 10001	06	149.99	New Cit
186847	222907	USB-C Charging Cable	1	11.95	2019-06-22 18:57:00	319 Ridge St, San Francisco, CA 94016	06	11.95	Fran
186848	222908	USB-C Charging Cable	1	11.95	2019-06-26 18:35:00	916 Main St, San Francisco, CA 94016	06	11.95	Fran
186849	222909	AAA Batteries (4-pack)	1	2.99	2019-06-25 14:33:00	209 11th St, Atlanta, GA 30301	06	2.99	At

185950 rows × 9 columns

In [40]:

```
newSdata["Hour"] = newSdata["Order Date"].dt.minute
```

```
/var/folders/1m/24t_f0_j3bb46tzcw1s7w04w0000gp/T/ipykernel_8729/18762708
93.py:1: 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-d
ocs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  newSdata["Hour"] = newSdata["Order Date"].dt.hour
```

In [44]:

```
newSdata["Minute"] = newSdata["Order Date"].dt.hour
```

```
/var/folders/1m/24t_f0_j3bb46tzcw1s7w04w0000gp/T/ipykernel_8729/80444347
3.py:1: 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-d
ocs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  newSdata["Minute"] = newSdata["Order Date"].dt.hour
```

In [45]:

```
newSdata
```

Out[45]:

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address	Month	Sales	
0	295665	Macbook Pro Laptop	1	1700.00	2019-12-30 00:01:00	136 Church St, New York City, NY 10001	12	1700.00	New Cit
1	295666	LG Washing Machine	1	600.00	2019-12-29 07:03:00	562 2nd St, New York City, NY 10001	12	600.00	New Cit
2	295667	USB-C Charging Cable	1	11.95	2019-12-12 18:21:00	277 Main St, New York City, NY 10001	12	11.95	New Cit
3	295668	27in FHD Monitor	1	149.99	2019-12-22 15:13:00	410 6th St, San Francisco, CA 94016	12	149.99	Fran
4	295669	USB-C Charging Cable	1	11.95	2019-12-18 12:38:00	43 Hill St, Atlanta, GA 30301	12	11.95	At
...	...	...	...	...	...	...	...	...	
186845	222905	AAA Batteries (4-pack)	1	2.99	2019-06-07 19:02:00	795 Pine St, Boston, MA 02215	06	2.99	Bo
186846	222906	27in FHD Monitor	1	149.99	2019-06-01 19:29:00	495 North St, New York City, NY 10001	06	149.99	New Cit
186847	222907	USB-C Charging Cable	1	11.95	2019-06-22 18:57:00	319 Ridge St, San Francisco, CA 94016	06	11.95	Fran
186848	222908	USB-C Charging Cable	1	11.95	2019-06-26 18:35:00	916 Main St, San Francisco, CA 94016	06	11.95	Fran
186849	222909	AAA Batteries (4-pack)	1	2.99	2019-06-25 14:33:00	209 11th St, Atlanta, GA 30301	06	2.99	At

185950 rows × 11 columns

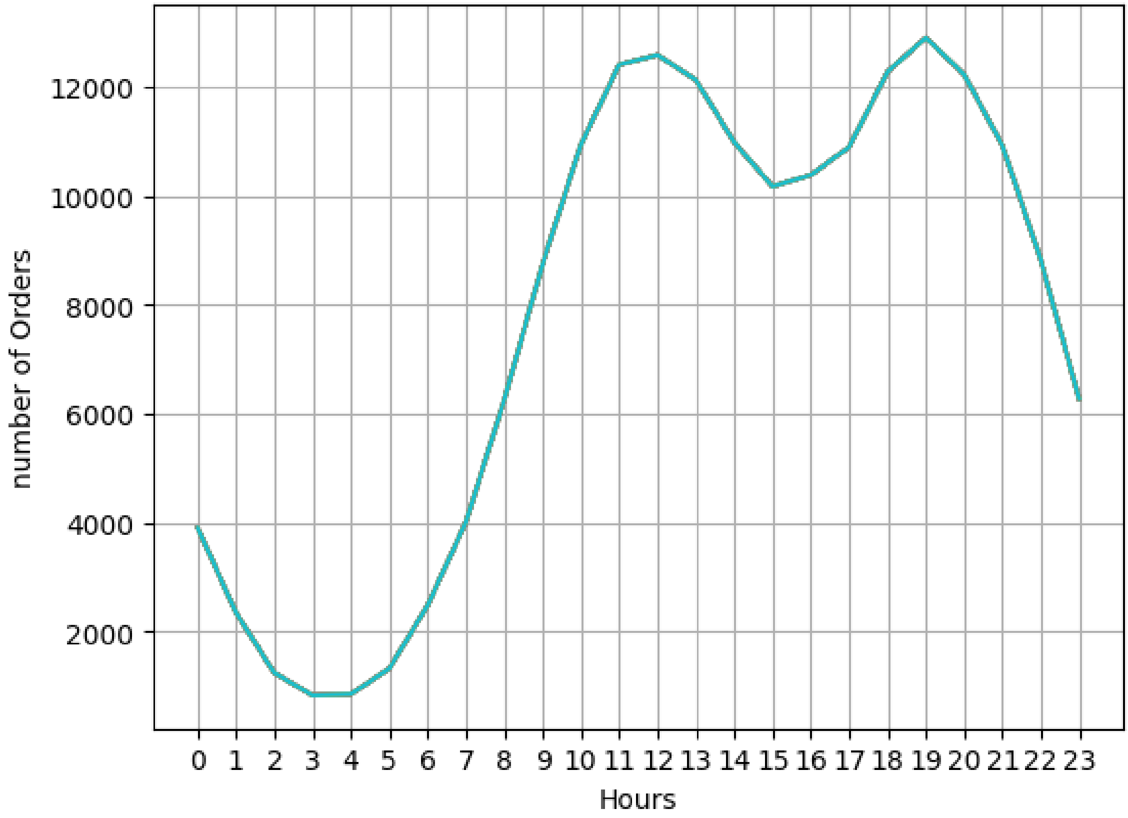
In [50]:

```
hours = [hour for hour, df in newSdata.groupby("Hour")]


filter = newSdata.groupby(["Hour"]).count()

plt.xticks(hours)
plt.xlabel("Hours")
plt.ylabel("number of Orders")
plt.grid()
plt.plot(hours, filter)
plt.show()

# the answer we can say 11 am and 7 pm.
```



What products are most often sold together ? (Order ID si aynı olan birlikte satılmıştır.)

In [59]:

```
df = newSdata[newSdata["Order ID"].duplicated(keep = False)]
```

In [61]:

```
df["Grouped"] = df.groupby("Order ID")["Product"].transform(lambda x: ','

df = df[["Order ID", "Grouped"]].drop_duplicates()
```

```
/var/folders/1m/24t_f0_j3bb46tzcw1s7w04w0000gp/T/ipykernel_8729/10974591
25.py:1: 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-d
ocs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df["Grouped"] = df.groupby("Order ID")["Product"].transform(lambda x:
','.join(x))
```

In [62]:

```
df
```

Out[62]:

	Order ID	Grouped
16	295681	Google Phone,USB-C Charging Cable,Bose SoundSp...
36	295698	Vareebadd Phone,USB-C Charging Cable
42	295703	AA Batteries (4-pack),Bose SoundSport Headphones
66	295726	iPhone,Lightning Charging Cable
76	295735	iPhone,Apple Airpods Headphones,Wired Headphones
...	...	...
186798	222863	27in FHD Monitor,Bose SoundSport Headphones
186804	222868	iPhone,Apple Airpods Headphones
186819	222882	Apple Airpods Headphones,AA Batteries (4-pack)
186822	222884	Google Phone,USB-C Charging Cable
186842	222903	iPhone,Lightning Charging Cable

7136 rows × 2 columns

In [70]:

```
from collections import Counter
from itertools import combinations

count = Counter()

for row in df["Grouped"]:
    row_list = row.split(",")
    count.update(Counter(combinations(row_list,2)))

#count.most_common(10)
for key,value in count.most_common(10):
    print(key,value)
```

```
('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 ?

In [71]:

```
newSdata
```

Out[71]:

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address	Month	Sales	
0	295665	Macbook Pro Laptop	1	1700.00	2019-12-30 00:01:00	136 Church St, New York City, NY 10001	12	1700.00	New Cit
1	295666	LG Washing Machine	1	600.00	2019-12-29 07:03:00	562 2nd St, New York City, NY 10001	12	600.00	New Cit
2	295667	USB-C Charging Cable	1	11.95	2019-12-12 18:21:00	277 Main St, New York City, NY 10001	12	11.95	New Cit
3	295668	27in FHD Monitor	1	149.99	2019-12-22 15:13:00	410 6th St, San Francisco, CA 94016	12	149.99	Fran
4	295669	USB-C Charging Cable	1	11.95	2019-12-18 12:38:00	43 Hill St, Atlanta, GA 30301	12	11.95	At
...	...	...	...	...	...	...	...	...	
186845	222905	AAA Batteries (4-pack)	1	2.99	2019-06-07 19:02:00	795 Pine St, Boston, MA 02215	06	2.99	Bo
186846	222906	27in FHD Monitor	1	149.99	2019-06-01 19:29:00	495 North St, New York City, NY 10001	06	149.99	New Cit
186847	222907	USB-C Charging Cable	1	11.95	2019-06-22 18:57:00	319 Ridge St, San Francisco, CA 94016	06	11.95	Fran
186848	222908	USB-C Charging Cable	1	11.95	2019-06-26 18:35:00	916 Main St, San Francisco, CA 94016	06	11.95	Fran
186849	222909	AAA Batteries (4-pack)	1	2.99	2019-06-25 14:33:00	209 11th St, Atlanta, GA 30301	06	2.99	At

185950 rows × 11 columns

In [73]:

```
product_group = newSdata.groupby("Product")
```

In [77]:

```
quantity_ordered = product_group["Quantity Ordered"].sum()

products = [product for product, df in product_group]
```

In [1]:

```
prices = newSdata["Product"].mean()["Price Each"]
```

```
fig, ax1 = plt.subplots()
ax2 = ax1.twinx()
ax1.bar(products,quantity_ordered)
ax2.plot(products,prices,"b-")

ax1.set_xlabel("Product Name")
ax1.set_ylabel("Quantity ordered")
ax2.set_xlabel("price", color = "b")
ax1.set_xticklabels(products, rotation = "vertical", size = 8)

plt.bar(products, quantity_ordered)

plt.show()
```

```
-------------------------------------------------------------------------
---
NameError                                 Traceback (most recent call la
st)
Cell In[1], line 1
----> 1 prices = newSdata["Product"].mean()["Price Each"]
      3 fig, ax1 = plt.subplots()
      4 ax2 = ax1.twinx()

NameError: name 'newSdata' is not defined
```

In [ ]:

In [ ]: