Sales Analysis

Import necessary libraries

In [1]:

```
import os
import pandas as pd
```

Reading single month csv file from data

In [20]:

```
pd=pd.read_csv('Sales_April_2019.csv')
df.head()
```

Out[20]:

	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
1	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
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

Merge data from each month into one CSV

In [96]:

```
# importing pandas
import pandas as pd

# merging two csv files
all_data = pd.concat(
    map(pd.read_csv, ['Sales_April_2019.csv', 'Sales_June_2019.csv', 'Sales_May_2019.csv', 'Sales_July_2019.csv',
                      'Sales_August_2019.csv', 'Sales_September_2019.csv', 'Sales_October_2019.csv',
                      'Sales_November_2019.csv', 'Sales_December_2019.csv', 'Sales_January_2019.csv',
                      'Sales_February_2019.csv', 'Sales_March_2019.csv']), ignore_index=True)
all_data.head()
```

Out[96]:

	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
1	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
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

CLEANING AND SOARTING THE DATA

while executing i found out that there are many NAN values in the data,so i used "dropna" command to delete or drop all the null or NAN values from the data table.

FINDING "NAN" VALUES

In [97]:

```
all_data=all_data.dropna()
all_data.head()
```

Out[97]:

	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

Get rid of text in order date column

In [99]:

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

Make columns correct type

Augment data with additional columns

In [128…

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

Out[128]:

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

CONVERTING COLOUMNS INTO THERE CORRECT TYPE

In [104…

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

ADDING SALES COLUMN

In [107…

```
all_data['sales']=all_data['Quantity Ordered'] * all_data['Price Each']
all_data.head()
```

Out[107]:

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

Q.1 which is the best month for sales? How much was earn that month?

In [123…

```
all_data.groupby('Month').sum()
```

```
/var/folders/_5/qjr_hl8n4613j6v36hg_q96m0000gn/T/ipykernel_5651/2834125621.py:1: FutureWarning: The default value of numeric_only in DataFrameGroupBy.sum is deprecated. In a fut
ure version, numeric_only will default to False. Either specify numeric_only or select only columns which should be valid for the function.
  all_data.groupby('Month').sum()
```

Out[123]:

	Quantity Ordered	Price Each	sales
Month			
1	10903	1811768.38	1822256.73
2	13449	2188884.72	2202022.42
3	17005	2791207.83	2807100.38
4	20558	3367671.02	3390670.24
5	18667	3135125.13	3152606.75
6	15253	2562025.61	2577802.26
7	16072	2632539.56	2647775.76
8	13448	2230345.42	2244467.88
9	13109	2084992.09	2097560.13
10	22703	3715554.83	3736726.88
11	19798	3180600.68	3199603.20
12	28114	4588415.41	4613443.34

In [126…

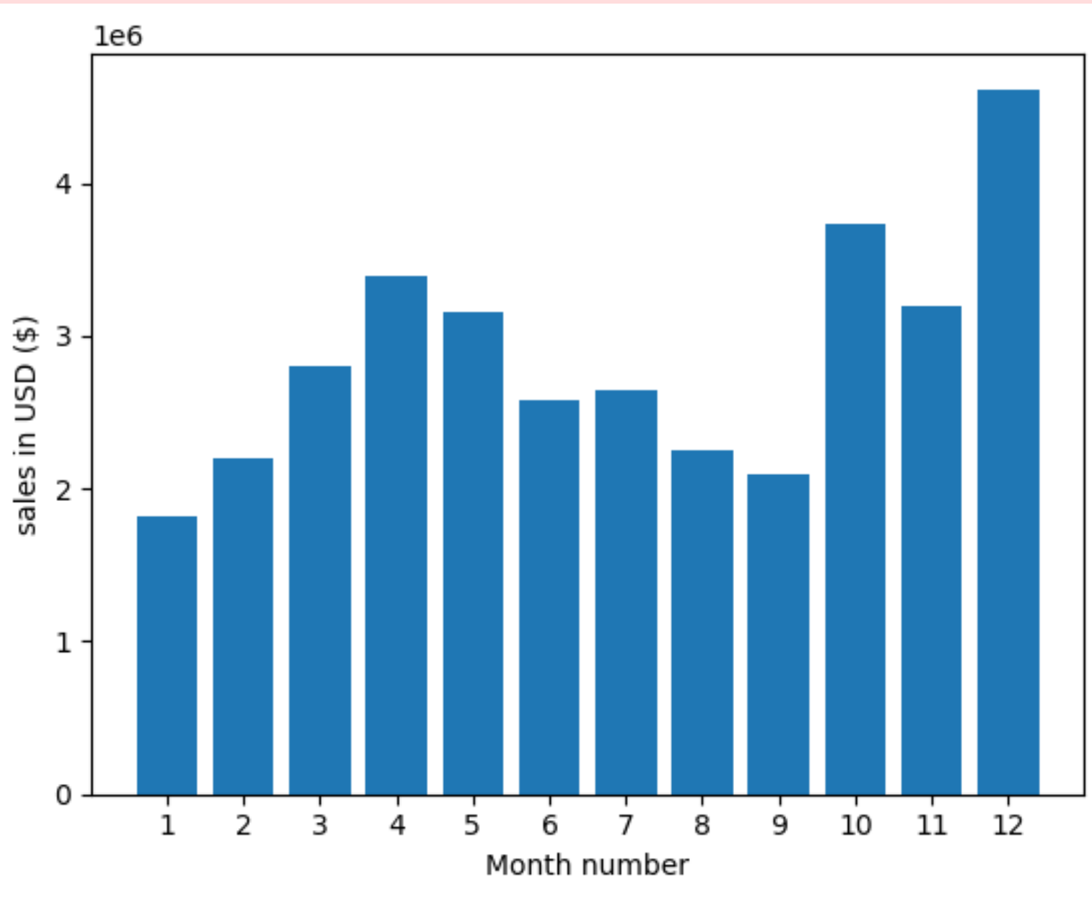
```
import matplotlib.pyplot as plt

months = range(1,13)
print(months)

plt.bar(months,all_data.groupby(['Month']).sum()['sales'])
plt.xticks(months)
plt.ylabel('sales in USD ($)')
plt.xlabel('Month number')
plt.show()
```

```
range(1, 13)
```

```
/var/folders/_5/qjr_hl8n4613j6v36hg_q96m0000gn/T/ipykernel_5651/541806747.py:6: FutureWarning: The default value of numeric_only in DataFrameGroupBy.sum is deprecated. In a futu
re version, numeric_only will default to False. Either specify numeric_only or select only columns which should be valid for the function.
  plt.bar(months,all_data.groupby(['Month']).sum()['sales'])
```



TASK 2 : Which city has the most higest number of sales?

SEPRATING CITY FROM PURCHASE ADDRESS AND CREATING NEW COLOUMN NAME AS "CITY"

In [141…

```
all_data['city']=all_data['Purchase Address'].apply(lambda x: x.split(',')[1])
all_data.head(10)
```

Out[141]:

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

In [169…

```
all_data.groupby('city').sum()
```

```
/var/folders/_5/qjr_hl8n4613j6v36hg_q96m0000gn/T/ipykernel_5651/2284805805.py:1: FutureWarning: The default value of numeric_only in DataFrameGroupBy.sum is deprecated. In a fut
ure version, numeric_only will default to False. Either specify numeric_only or select only columns which should be valid for the function.
  all_data.groupby('city').sum()
```

Out[169]:

	Quantity Ordered	Price Each	Month	sales
city				
Atlanta	16602	2779908.20	104794	2795498.58
Austin	11153	1809873.61	69829	1819581.75
Boston	22528	3637409.77	141112	3661642.01
Dallas	16730	2752627.82	104620	2767975.40
Los Angeles	33289	5421435.23	208325	5452570.80
New York City	27932	4635370.83	175741	4664317.43
Portland	14053	2307747.47	87765	2320490.61
San Francisco	50239	8211461.74	315520	8262203.91
Seattle	16553	2733296.01	104941	2747755.48

In [183…

```
all_data = all_data.sort_values(by='sales',ascending=False)
all_data.head()
#we can clearly see that san francisco has the most number of sales in the year
```

Out[183]:

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address	Month	sales	city	time
5219	181544	Macbook Pro Laptop	2	1700.00	04/22/19 12:48	731 11th St, New York City, NY 10001	4	3400.00	New York City	12:48
38781	200528	Macbook Pro Laptop	2	1700.00	05/13/19 13:40	643 4th St, Boston, MA 02215	5	3400.00	Boston	13:40
4717	181069	Macbook Pro Laptop	2	1700.00	04/27/19 21:01	668 Park St, San Francisco, CA 94016	4	3400.00	San Francisco	21:01
18768	210292	Macbook Pro Laptop	2	1700.00	06/08/19 09:00	953 Ridge St, San Francisco, CA 94016	6	3400.00	San Francisco	09:00
106921	278637	ThinkPad Laptop	2	999.99	10/02/19 16:06	643 Cedar St, Boston, MA 02215	10	1999.98	Boston	16:06

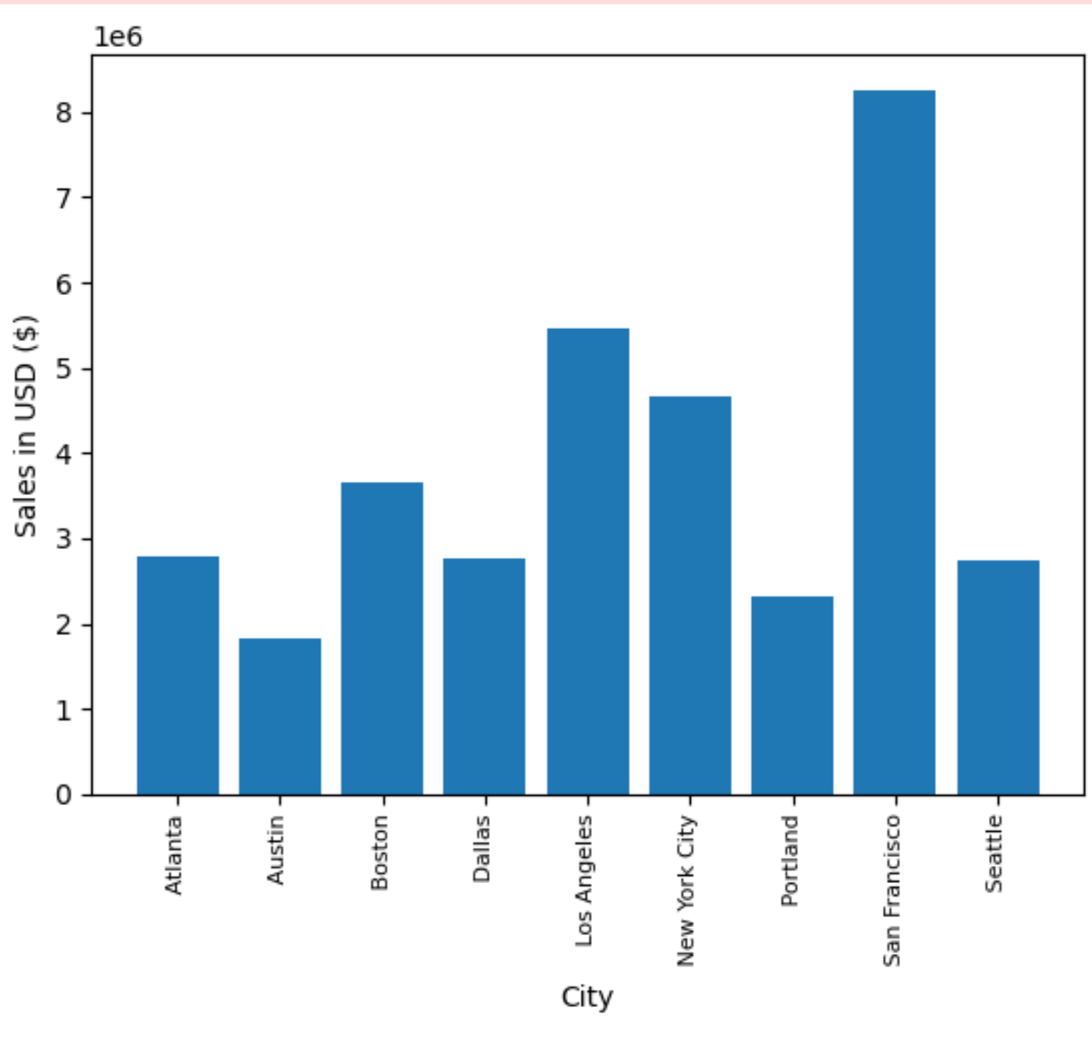
In [189…

```
import matplotlib.pyplot as plt

keys = [city for city, df in all_data.groupby(['city'])]

plt.bar(keys,all_data.groupby(['city']).sum()['sales'])
plt.ylabel('Sales in USD ($)')
plt.xlabel('City')
plt.xticks(keys, rotation='vertical', size=8)
plt.show()
```

```
/var/folders/_5/qjr_hl8n4613j6v36hg_q96m0000gn/T/ipykernel_5651/3387295046.py:3: FutureWarning: In a future version of pandas, a length 1 tuple will be returned when iterating o
ver a groupby with a grouper equal to a list of length 1. Don't supply a list with a single grouper to avoid this warning.
  keys = [city for city, df in all_data.groupby(['city'])]
/var/folders/_5/qjr_hl8n4613j6v36hg_q96m0000gn/T/ipykernel_5651/3387295046.py:5: FutureWarning: The default value of numeric_only in DataFrameGroupBy.sum is deprecated. In a fut
ure version, numeric_only will default to False. Either specify numeric_only or select only columns which should be valid for the function.
  plt.bar(keys,all_data.groupby(['city']).sum()['sales'])
```



TASK 3 : what time should we display advertisement to maximize likehood of customer buying things?

In [179…

```
all_data['time']=all_data['Order Date'].apply(lambda x: x.split(' ')[1])
all_data.head()
```

Out[179]:

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address	Month	sales	city	time
5219	181544	Macbook Pro Laptop	2	1700.00	04/22/19 12:48	731 11th St, New York City, NY 10001	4	3400.00	New York City	12:48
4717	181069	Macbook Pro Laptop	2	1700.00	04/27/19 21:01	668 Park St, San Francisco, CA 94016	4	3400.00	San Francisco	21:01
18768	210292	Macbook Pro Laptop	2	1700.00	06/08/19 09:00	953 Ridge St, San Francisco, CA 94016	6	3400.00	San Francisco	09:00
38781	200528	Macbook Pro Laptop	2	1700.00	05/13/19 13:40	643 4th St, Boston, MA 02215	5	3400.00	Boston	13:40
106921	278637	ThinkPad Laptop	2	999.99	10/02/19 16:06	643 Cedar St, Boston, MA 02215	10	1999.98	Boston	16:06

In [187…

```
all_data.groupby('time').sum()
```

```
/var/folders/_5/qjr_hl8n4613j6v36hg_q96m0000gn/T/ipykernel_5651/536136296.py:1: FutureWarning: The default value of numeric_only in DataFrameGroupBy.sum is deprecated. In a futu
re version, numeric_only will default to False. Either specify numeric_only or select only columns which should be valid for the function.
  all_data.groupby('time').sum()
```

Out[187]:

	Quantity Ordered	Price Each	Month	sales
time				
00:00	91	22450.44	584	22488.86
00:01	73	12772.26	459	12814.95
00:02	81	12464.85	514	12951.64
00:03	101	18683.49	630	18759.14
00:04	96	16914.39	549	16960.08
...	...	...	...	...
23:55	113	17215.84	656	17294.89
23:56	79	19424.50	506	19463.40
23:57	96	16190.19	605	16237.62
23:58	104	19544.17	641	19606.93
23:59	108	18041.50	626	18079.08

1440 rows × 4 columns

In [190…

```
# Add hour column
all_data['Hour'] = pd.to_datetime(all_data['Order Date']).dt.hour
all_data['Minute'] = pd.to_datetime(all_data['Order Date']).dt.minute
all_data['Count'] = 1
all_data.head()
```

Out[190]:

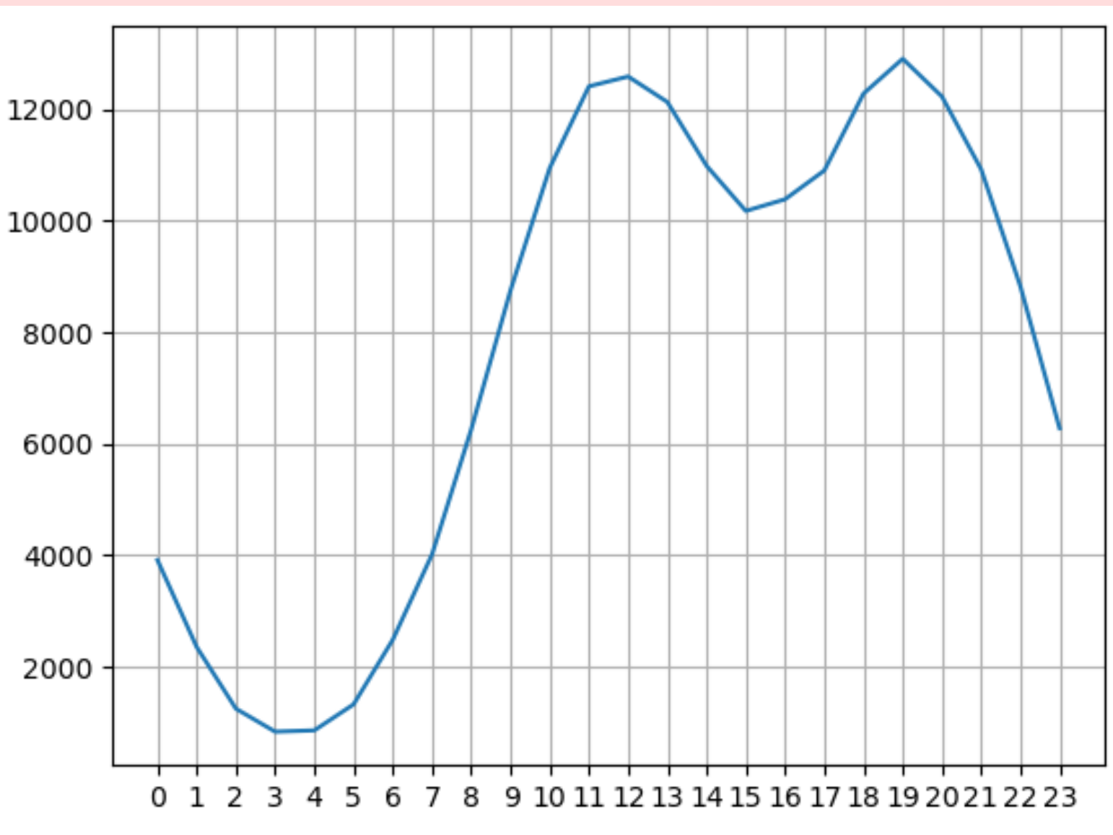
	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address	Month	sales	city	time	Hour	Minute	Count
5219	181544	Macbook Pro Laptop	2	1700.00	04/22/19 12:48	731 11th St, New York City, NY 10001	4	3400.00	New York City	12:48	12	48	1
4717	181069	Macbook Pro Laptop	2	1700.00	04/27/19 21:01	668 Park St, San Francisco, CA 94016	4	3400.00	San Francisco	21:01	21	1	1
18768	210292	Macbook Pro Laptop	2	1700.00	06/08/19 09:00	953 Ridge St, San Francisco, CA 94016	6	3400.00	San Francisco	09:00	9	0	1
38781	200528	Macbook Pro Laptop	2	1700.00	05/13/19 13:40	643 4th St, Boston, MA 02215	5	3400.00	Boston	13:40	13	40	1
106921	278637	ThinkPad Laptop	2	999.99	10/02/19 16:06	643 Cedar St, Boston, MA 02215	10	1999.98	Boston	16:06	16	6	1

In [191…

```
keys = [pair for pair, df in all_data.groupby(['Hour'])]

plt.plot(keys, all_data.groupby(['Hour']).count()['Count'])
plt.xticks(keys)
plt.grid()
plt.show()
```

```
/var/folders/_5/qjr_hl8n4613j6v36hg_q96m0000gn/T/ipykernel_5651/2891002684.py:1: FutureWarning: In a future version of pandas, a length 1 tuple will be returned when iterating o
ver a groupby with a grouper equal to a list of length 1. Don't supply a list with a single grouper to avoid this warning.
  keys = [pair for pair, df in all_data.groupby(['Hour'])]
```



In [ ]: