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

In [2]:
```
import os
```

In [3]:
```
df = pd.read_csv("./Sales_Data/Sales_April_2019.csv")
```

In [4]:
```
df.head()
```

Out[4]:

	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

In [6]:
```
files = [file for file in os.listdir("./Sales_Data")]
all_months_data = pd.DataFrame()
for file in files:
    df = pd.read_csv("./Sales_data/"+file)
    all_months_data = pd.concat([all_months_data,df])
```

In [7]:
```
all_months_data.to_csv("all_data.csv", index=False)
```

In [9]:
```
all_data = pd.read_csv("all_data.csv")
```

In [10]:
```
all_data.head()
```

Out[10]:

	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

Drop rows of NaN Values

In [11]:
```
nan_df = all_data[all_data.isna().any(axis=1)]
```

In [12]:
```
nan_df.head()
```

Out[12]:

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address
1	NaN	NaN	NaN	NaN	NaN	NaN
356	NaN	NaN	NaN	NaN	NaN	NaN
735	NaN	NaN	NaN	NaN	NaN	NaN
1433	NaN	NaN	NaN	NaN	NaN	NaN
1553	NaN	NaN	NaN	NaN	NaN	NaN

In [13]:
```
all_data = all_data.dropna(how ="all")
```

In [17]:
```
## all_data = all_data[all_data['Order Date'].str[0:2] != 'Or']
```

In [18]:
```
all_data['Quantity Ordered'] = pd.to_numeric(all_data['Quantity Ordered'])
all_data['Price Each'] = pd.to_numeric(all_data['Price Each'])
```

```
<ipython-input-18-81603091079b>: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-docs/stable/u
ser_guide/indexing.html#returning-a-view-versus-a-copy
  all_data['Quantity Ordered'] = pd.to_numeric(all_data['Quantity Ordered'])
<ipython-input-18-81603091079b>:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/u
ser_guide/indexing.html#returning-a-view-versus-a-copy
  all_data['Price Each'] = pd.to_numeric(all_data['Price Each'])
```

In [19]:
```
all_data['Month'] = all_data['Order Date'].str[0:2]
```

```
<ipython-input-19-d69b0e923aac>: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-docs/stable/u
ser_guide/indexing.html#returning-a-view-versus-a-copy
  all_data['Month'] = all_data['Order Date'].str[0:2]
```

In [20]:
```
all_data.head()
```

Out[20]:

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

Question 1: Which was the highest month of sale? How much was earned in that month?

In [21]:
```
all_data['Sales'] = all_data['Quantity Ordered'] * all_data['Price Each']
```

```
<ipython-input-21-4b4247d7272e>: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-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  all_data['Sales'] = all_data['Quantity Ordered'] * all_data['Price Each']
```

In [22]:
```
all_data.head()
```

Out[22]:

	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	04	23.90
2	176559	Bose SoundSport Headphones	1	99.99	04/07/19 22:30	682 Chestnut St, Boston, MA 02215	04	99.99
3	176560	Google Phone	1	600.00	04/12/19 14:38	669 Spruce St, Los Angeles, CA 90001	04	600.00
4	176560	Wired Headphones	1	11.99	04/12/19 14:38	669 Spruce St, Los Angeles, CA 90001	04	11.99
5	176561	Wired Headphones	1	11.99	04/30/19 09:27	333 8th St, Los Angeles, CA 90001	04	11.99

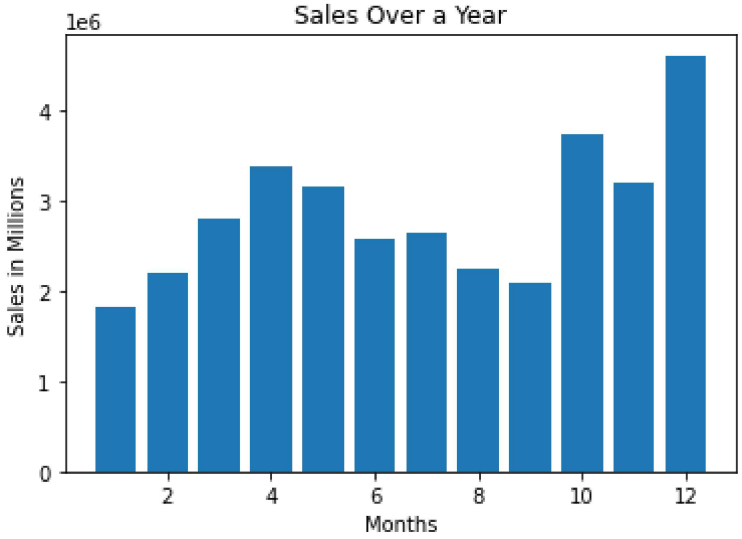
In [28]:
```
results = all_data.groupby('Month').sum()
```

Ans: Month of December was the highest selling sales month with over $4Million

In [29]:
```
import matplotlib.pyplot as plt
```

```
Matplotlib is building the font cache; this may take a moment.
```

In [34]:
```
months = range(1,13)
plt.bar(months,results['Sales'])
plt.xlabel('Months')
plt.ylabel('Sales in Millions')
plt.title('Sales Over a Year')
plt.show()
```



Question 2: Which City had the highest sales?

In [36]:
```
all_data['City'] = all_data['Purchase Address'].apply(lambda x: x.split(',')[1])
```

In [37]:
```
all_data.head()
```

Out[37]:

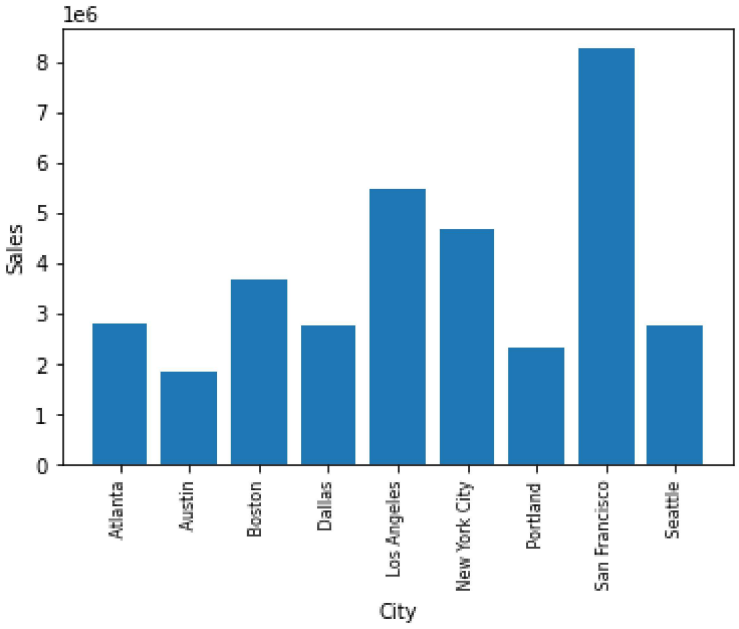
	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address	Month	Sales	City
0	176558	USB-C Charging Cable	2	11.95	04/19/19 08:46	917 1st St, Dallas, TX 75001	04	23.90	Dallas
2	176559	Bose SoundSport Headphones	1	99.99	04/07/19 22:30	682 Chestnut St, Boston, MA 02215	04	99.99	Boston
3	176560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

In [40]:
```
results = all_data.groupby('City').sum()
```

In [39]:
```
import matplotlib.pyplot as plt
```

In [42]:
```
cities = [city for city, df in all_data.groupby('City')]

plt.bar(cities,results['Sales'])
plt.xticks(cities,rotation = 'vertical',size=8)
plt.xlabel('City')
plt.ylabel('Sales')
plt.show()
```



In [ ]:
```
```

Question 3: What products are most often sold together?

In [77]:
```
df = all_data[all_data['Order ID'].duplicated(keep=False)]
```

In [78]:
```
df.head(20)
```

Out[78]:

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address	Month	Sales	City
3	176560	Google Phone	1	600.00	176560	669 Spruce St, Los Angeles, CA 90001	04	600.00	Los Angeles
4	176560	Wired Headphones	1	11.99	176560	669 Spruce St, Los Angeles, CA 90001	04	11.99	Los Angeles
18	176574	Google Phone	1	600.00	176574	20 Hill St, Los Angeles, CA 90001	04	600.00	Los Angeles
19	176574	USB-C Charging Cable	1	11.95	176574	20 Hill St, Los Angeles, CA 90001	04	11.95	Los Angeles
30	176585	Bose SoundSport Headphones	1	99.99	176585	823 Highland St, Boston, MA 02215	04	99.99	Boston
31	176585	Bose SoundSport Headphones	1	99.99	176585	823 Highland St, Boston, MA 02215	04	99.99	Boston
32	176586	AAA Batteries (4-pack)	2	2.99	176586	365 Center St, San Francisco, CA 94016	04	5.98	San Francisco
33	176586	Google Phone	1	600.00	176586	365 Center St, San Francisco, CA 94016	04	600.00	San Francisco
119	176672	Lightning Charging Cable	1	14.95	176672	778 Maple St, New York City, NY 10001	04	14.95	New York City
120	176672	USB-C Charging Cable	1	11.95	176672	778 Maple St, New York City, NY 10001	04	11.95	New York City
129	176681	Apple Airpods Headphones	1	150.00	176681	331 Cherry St, Seattle, WA 98101	04	150.00	Seattle
130	176681	ThinkPad Laptop	1	999.99	176681	331 Cherry St, Seattle, WA 98101	04	999.99	Seattle
138	176689	Bose SoundSport Headphones	1	99.99	176689	659 Lincoln St, New York City, NY 10001	04	99.99	New York City
139	176689	AAA Batteries (4-pack)	2	2.99	176689	659 Lincoln St, New York City, NY 10001	04	5.98	New York City
189	176739	34in Ultrawide Monitor	1	379.99	176739	730 6th St, Austin, TX 73301	04	379.99	Austin

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address	Month	Sales	City
190	176739	Google Phone	1	600.00	176739	730 6th St, Austin, TX 73301	04	600.00	Austin
225	176774	Lightning Charging Cable	1	14.95	176774	372 Church St, Los Angeles, CA 90001	04	14.95	Los Angeles
226	176774	USB-C Charging Cable	1	11.95	176774	372 Church St, Los Angeles, CA 90001	04	11.95	Los Angeles
233	176781	iPhone	1	700.00	176781	976 Hickory St, Dallas, TX 75001	04	700.00	Dallas
234	176781	Lightning Charging Cable	1	14.95	176781	976 Hickory St, Dallas, TX 75001	04	14.95	Dallas

In [79]:
```
df['Grouped'] = df.groupby('Order ID')['Product'].transform(lambda x: ','.join(x))
```

```
<ipython-input-79-9d66bca9fec4>: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-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df['Grouped'] = df.groupby('Order ID')['Product'].transform(lambda x: ','.join(x))
```

In [80]:
```
df = df[['Order ID','Grouped']].drop_duplicates()
```

In [81]:
```
df.head(20)
```

Out[81]:

	Order ID	Grouped
3	176560	Google Phone,Wired Headphones
18	176574	Google Phone,USB-C Charging Cable
30	176585	Bose SoundSport Headphones,Bose SoundSport Hea...
32	176586	AAA Batteries (4-pack),Google Phone
119	176672	Lightning Charging Cable,USB-C Charging Cable
129	176681	Apple Airpods Headphones,ThinkPad Laptop
138	176689	Bose SoundSport Headphones,AAA Batteries (4-pack)
189	176739	34in Ultrawide Monitor,Google Phone
225	176774	Lightning Charging Cable,USB-C Charging Cable
233	176781	iPhone,Lightning Charging Cable
250	176797	Google Phone,Bose SoundSport Headphones,Wired ...
260	176805	Google Phone,USB-C Charging Cable
264	176808	Google Phone,Wired Headphones
270	176813	Google Phone,Wired Headphones
394	176935	AAA Batteries (4-pack),27in FHD Monitor

	Order ID	Grouped
435	176975	USB-C Charging Cable,AAA Batteries (4-pack)
450	176989	Google Phone,USB-C Charging Cable
455	176993	iPhone,Wired Headphones
485	177022	iPhone,Wired Headphones
567	177102	iPhone,27in 4K Gaming Monitor

In [82]:
```
from itertools import combinations
from collections import Counter
```

In [83]:
```
count = Counter()
for row in df['Grouped']:
    row_list = row.split(',')
    count.update(Counter(combinations(row_list,2)))
for key,value in count.most_common(20):
    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
('Lightning Charging Cable', 'Apple Airpods Headphones') 81
('Vareebadd Phone', 'Bose SoundSport Headphones') 80
('USB-C Charging Cable', 'Bose SoundSport Headphones') 77
('Apple Airpods Headphones', 'Wired Headphones') 69
('Lightning Charging Cable', 'USB-C Charging Cable') 58
('Lightning Charging Cable', 'AA Batteries (4-pack)') 55
('Lightning Charging Cable', 'Lightning Charging Cable') 54
('Bose SoundSport Headphones', 'Wired Headphones') 53
('AA Batteries (4-pack)', 'Lightning Charging Cable') 51
('AAA Batteries (4-pack)', 'USB-C Charging Cable') 50
```

In [ ]: