Sales Analysis

In [ ]:

```
##Import Necessary Libraries
```

In [ ]:

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

Task #1 Merge the 12 months of sales data into a single CSV file

In [ ]:

```
df = pd.read_csv("C:/Users/User/Desktop/Sales_Data/Sales_April_2019.csv")

df.shape
```

In [ ]:

```
files = [file for file in os.listdir('C:/Users/User/Desktop/Sales_Data')]

all_months_data = pd.DataFrame()

for file in files:
    df = pd.read_csv("C:/Users/User/Desktop/Sales_Data/"+file)
    all_months_data = pd.concat([all_months_data,df])

all_months_data.to_csv("all_data.csv",index = False)
```

In [ ]:

```
all_data = pd.read_csv("all_data.csv")
```

In [ ]:

```
all_data.head()
```

In [ ]:

```
all_data.isnull().sum()
```

In [ ]:

```
all_data.info()
```

In [ ]:

```
all_data.dropna(inplace = True)
```

In [ ]:

```
all_data.isnull().sum()
```

In [ ]:

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

Augment data with additional columns

Task #2 : Add Month Column

In [ ]:

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

In [ ]:

```
all_data.info()
```

Columns to the right type

In [ ]:

```
all_data['Quantity Ordered'] =all_data['Quantity Ordered'].astype('int32')
all_data['Price Each'] = all_data['Price Each'].astype('float')
```

Task #3 Add a sales column

In [ ]:

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

In [ ]:

```
all_data.head()
```

Task #4 Add a city column

In [ ]:

```
all_data['Purchase Address'].unique()
```

Too much of data here........... But we can still figure out the pattern.

In [ ]:

```
#Let's use .apply()

all_data['City'] = all_data['Purchase Address'].str.split(",").str[1]
```

In [ ]:

```
all_data.head()
```

Question #1: What was the best month for sales? How much was earned that month?

In [ ]:

```
results = all_data.groupby('Month').sum()# .sort_values(by = 'Sales',ascending = False)
```

In [ ]:

```
## Let's plot to see the sales
```

In [ ]:

```
## Importing the library

import matplotlib.pyplot as plt

months = range(1,13)

plt.plot(months,results['Sales'])
plt.xlabel('Months')
plt.ylabel('Sales')
plt.show()
```

Answer : December was the month with the maximum sales. And 4.613443e+06 was earned during december.

Question #2: What city sold the most product?

In [ ]:

```
all_data.groupby('City')['Quantity Ordered'].sum().sort_values(ascending = False)
```

In [ ]:

```
result = all_data.groupby('City')['Quantity Ordered'].sum().sort_values(ascending = False)
```

Sanfrancisco was the city selling the highest quantity of product.

In [ ]:

```
### Let's plot it down

result.plot(kind='barh')
plt.plot()
```

Question #3: What time should we display advertisements to maximize the likelihood of purchases?

In [ ]:

```
all_data['Order Date'] = pd.to_datetime(all_data['Order Date'])
```

In [ ]:

```
all_data.head()
```

In [ ]:

```
all_data.info()
```

In [ ]:

```
all_data['Hour'] = all_data['Order Date'].dt.hour
all_data['Minute']= all_data['Order Date'].dt.minute
```

In [ ]:

```
all_data.head()
```

In [ ]:

```
hours = [hour for hour,df in all_data.groupby('Hour')]

plt.plot(hours,all_data.groupby(['Hour']).count())
plt.xlabel('Hours')
plt.ylabel('The sales')
plt.xticks(hours)
plt.grid()
plt.show()
```

It's pretty clear to perform the sales from 8 -12am and then from 6 to 8 pm.

Question #4: What products are most often sold together?

In [37]:

In [37]:

```
all_data['Product'].unique()
```

Out[37]:

```
array(['USB-C Charging Cable', 'Bose SoundSport Headphones',
       'Google Phone', 'Wired Headphones', 'Macbook Pro Laptop',
       'Lightning Charging Cable', '27in 4K Gaming Monitor',
       'AA Batteries (4-pack)', 'Apple Airpods Headphones',
       'AAA Batteries (4-pack)', 'iPhone', 'Flatscreen TV',
       '27in FHD Monitor', '20in Monitor', 'LG Dryer', 'ThinkPad Laptop',
       'Vareebadd Phone', 'LG Washing Machine', '34in Ultrawide Monitor'],
      dtype=object)
```

In [38]:

```
all_data.groupby('Product')['Product'].count().sort_values(ascending = False)
```

Out[38]:

```
Product
USB-C Charging Cable          21903
Lightning Charging Cable      21658
AAA Batteries (4-pack)        20641
AA Batteries (4-pack)         20577
Wired Headphones              18882
Apple Airpods Headphones      15549
Bose SoundSport Headphones    13325
27in FHD Monitor               7507
iPhone                         6842
27in 4K Gaming Monitor         6230
34in Ultrawide Monitor         6181
Google Phone                   5525
Flatscreen TV                  4800
Macbook Pro Laptop             4724
ThinkPad Laptop                4128
20in Monitor                   4101
Vareebadd Phone                2065
LG Washing Machine              666
LG Dryer                        646
Name: Product, dtype: int64
```

Lets check for the orders with the same order id. It probably tells us that the 4

Product with the same order id suggests that they were brough by the same person. So looking for the duplicate OrderID

In [41]:

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

```
<ipython-input-41-ace56740cd6e>:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_g
uide/indexing.html#returning-a-view-versus-a-copy
  df['Grouped'] = df.groupby('Order ID')['Product'].transform(lambda x: ','.join(x))
```

Out[41]:

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address	Month	Sales	City	Hour	Minute	Grouped
3	176560	Google Phone	1	600.00	2019-04-12 14:38:00	669 Spruce St, Los Angeles, CA 90001	4	600.00	Los Angeles	14	38	Google Phone,Wired Headphones
						669 Spruce						

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address	Month	Sales	City	Hour	Minute	Grouped
4	176560	Wired Headphones	1	11.99	2019-04-12 14:38:00	Spruce St, Los Angeles, CA 90001	4	11.99	Los Angeles	14	38	Google Phone,Wired Headphones
18	176574	Google Phone	1	600.00	2019-04-03 19:42:00	20 Hill St, Los Angeles, CA 90001	4	600.00	Los Angeles	19	42	Google Phone,USB-C Charging Cable
19	176574	USB-C Charging Cable	1	11.95	2019-04-03 19:42:00	20 Hill St, Los Angeles, CA 90001	4	11.95	Los Angeles	19	42	Google Phone,USB-C Charging Cable
30	176585	Bose SoundSport Headphones	1	99.99	2019-04-07 11:31:00	823 Highland St, Boston, MA 02215	4	99.99	Boston	11	31	Bose SoundSport Headphones,Bose SoundSport Hea...

In [49]:

```
##Lets dro out the duplicate occerance

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

Out[49]:

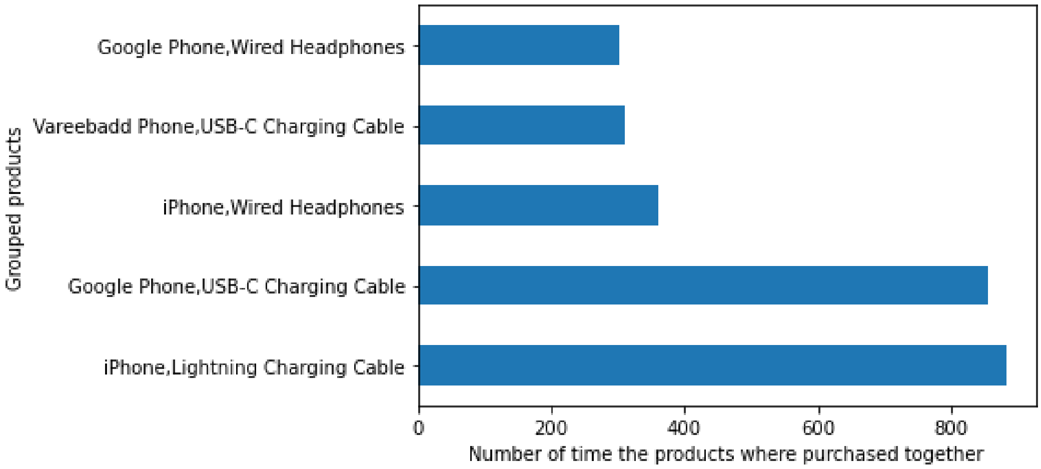
	Order ID	Grouped
3	176560	Google Phone,Wired Headphones
18	176574	Google Phone,USB-C Charging Cable
30	176585	Bose SoundSport Headphones,Bose SoundSport Hea...
32	176586	AAA Batteries (4-pack),Google Phone
119	176672	Lightning Charging Cable,USB-C Charging Cable

In [55]:

```
res = df.groupby('Grouped')['Grouped'].count().sort_values(ascending= False)
```

In [70]:

```
res.head().plot(kind = 'barh')
plt.xlabel('Number of time the products where purchased together')
plt.ylabel('Grouped products')
plt.show()
```



Question #5: What product sold the most? Why do you think it did?

In [60]:

```
all_data.head()
```

Out[60]:

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address	Month	Sales	City	Hour	Minute
0	176558	USB-C Charging Cable	2	11.95	2019-04-19 08:46:00	917 1st St, Dallas, TX 75001	4	23.90	Dallas	8	46
2	176559	Bose SoundSport Headphones	1	99.99	2019-04-07 22:30:00	682 Chestnut St, Boston, MA 02215	4	99.99	Boston	22	30
3	176560	Google Phone	1	600.00	2019-04-12 14:38:00	669 Spruce St, Los Angeles, CA 90001	4	600.00	Los Angeles	14	38
4	176560	Wired Headphones	1	11.99	2019-04-12 14:38:00	669 Spruce St, Los Angeles, CA 90001	4	11.99	Los Angeles	14	38
5	176561	Wired Headphones	1	11.99	2019-04-30 09:27:00	333 8th St, Los Angeles, CA 90001	4	11.99	Los Angeles	9	27

In [69]:

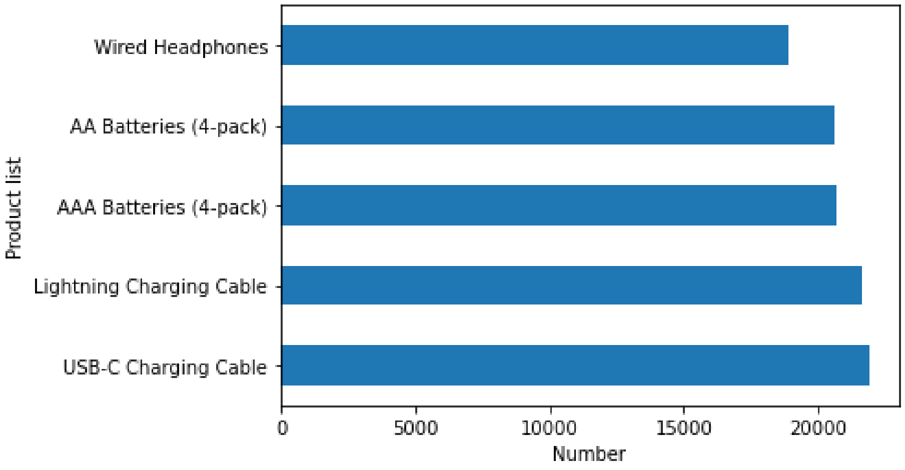
```
new_product = all_data.groupby('Product')['Product'].count().sort_values(ascending = False)
new_product.head()
```

Out[69]:

```
Product
USB-C Charging Cable          21903
Lightning Charging Cable      21658
AAA Batteries (4-pack)        20641
AA Batteries (4-pack)         20577
Wired Headphones              18882
Name: Product, dtype: int64
```

In [77]:

```
new_product.head().plot(kind ='barh')
plt.xlabel('Number')
plt.ylabel('Product list')
plt.show()
```


Wired Headphones
AA Batteries (4-pack)
AAA Batteries (4-pack)
Lightning Charging Cable
USB-C Charging Cable
Product list
Number