sales

March 31, 2024

[1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings('ignore')
```

[2]:
```
data=pd.read_csv(r"E:\excel\all_data_copy.csv")
```

[3]:
```
data.columns
```

[3]:
```
Index(['Order ID', 'Product', 'Quantity Ordered', 'Price Each', 'Order Date',
       'Purchase Address'],
      dtype='object')
```

[4]:
```
data.iloc[1679,:]
```

[4]:
```
Order ID                    Order ID
Product                      Product
Quantity Ordered    Quantity Ordered
Price Each                Price Each
Order Date                Order Date
Purchase Address    Purchase Address
Name: 1679, dtype: object
```

[5]:
```
data['Month']=data['Order Date'].str[:2]
```

[6]:
```
data['Month'].unique()
```

[6]:
```
array(['09', nan, '10', 'Or', '11', '12', '01', '02', '03', '04', '05',
       '07', '08', '06'], dtype=object)
```

[7]:
```
df=data[~(data['Month']=="Or")]
```

[8]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 186495 entries, 0 to 186849
Data columns (total 7 columns):
 #   Column            Non-Null Count   Dtype
```

```
---  ------            --------------   -----
 0   Order ID          185950 non-null  object
 1   Product           185950 non-null  object
 2   Quantity Ordered  185950 non-null  object
 3   Price Each        185950 non-null  object
 4   Order Date        185950 non-null  object
 5   Purchase Address  185950 non-null  object
 6   Month             185950 non-null  object
dtypes: object(7)
memory usage: 11.4+ MB
```

[9]:
```
def convert_dtype(data_frame):
    for i in data_frame.columns:
        try:
            data_frame[i]=data_frame[i].astype('float32')
        except:
            continue
    return data_frame
```

[10]:
```
df=convert_dtype(df)
```

[11]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 186495 entries, 0 to 186849
Data columns (total 7 columns):
 #   Column            Non-Null Count   Dtype
---  ------            --------------   -----
 0   Order ID          185950 non-null  float32
 1   Product           185950 non-null  object
 2   Quantity Ordered  185950 non-null  float32
 3   Price Each        185950 non-null  float32
 4   Order Date        185950 non-null  object
 5   Purchase Address  185950 non-null  object
 6   Month             185950 non-null  float32
dtypes: float32(4), object(3)
memory usage: 8.5+ MB
```

[12]:
```
df.dropna(inplace=True)
```

[13]:
```
df.isnull().sum()
```

[13]:
```
Order ID            0
Product             0
Quantity Ordered    0
Price Each          0
Order Date          0
```

```
Purchase Address    0
Month               0
dtype: int64
```

[14]:
```
df.dtypes
```

[14]:
```
Order ID            float32
Product              object
Quantity Ordered    float32
Price Each          float32
Order Date           object
Purchase Address     object
Month               float32
dtype: object
```

[15]:
```
def text_split(x):
    return x.split(",")[1]+ "("+ x.split(",")[2].split(" ")[1] + ")"
```

[16]:
```
df['City']=df['Purchase Address'].apply(lambda x: x.split(",")[1]+ "("+ x.
 ↪split(",")[2].split(" ")[1] + ")")
```

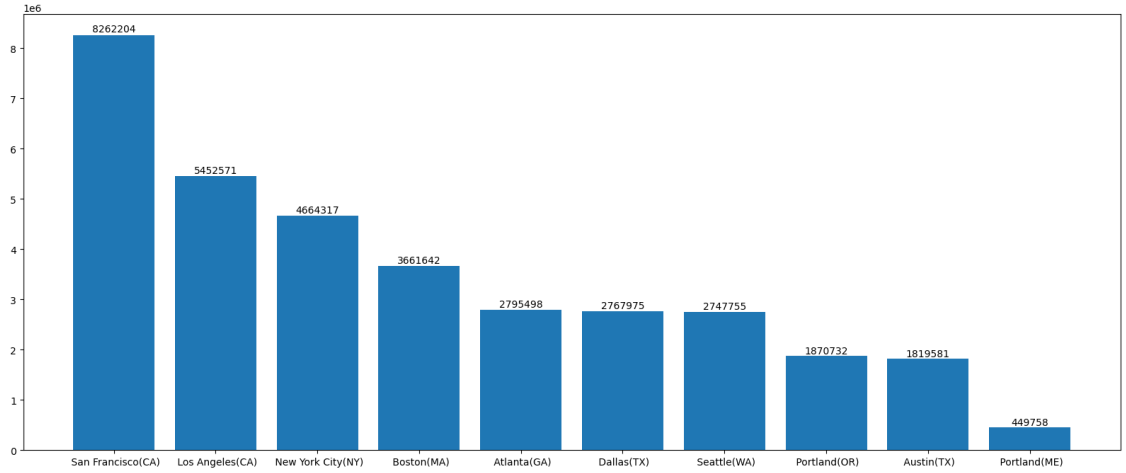
[17]:
```
df=df.assign(Total_Price=lambda x : x['Price Each']*x['Quantity Ordered'])
```

[18]:
```
df.groupby('City')['Total_Price'].agg('sum').sort_values(ascending=False).
 ↪astype('int32')
```

[18]:
```
City
San Francisco(CA)    8262204
Los Angeles(CA)      5452571
New York City(NY)    4664317
Boston(MA)           3661642
Atlanta(GA)          2795498
Dallas(TX)           2767975
Seattle(WA)          2747755
Portland(OR)         1870732
Austin(TX)           1819581
Portland(ME)          449758
Name: Total_Price, dtype: int32
```

[19]:
```
plt.figure(figsize=(20,8))
plot=plt.bar(df.groupby('City')['Total_Price'].agg('sum').
 ↪sort_values(ascending=False).astype('int32').index,df.
 ↪groupby('City')['Total_Price'].agg('sum').sort_values(ascending=False).
 ↪astype('int32'))
for ractangle in plot:
    height=ractangle.get_height()
    width=ractangle.get_x()+ractangle.get_width()/2
```

```
    plt.text(width,height*1.003,'%d' %int(height),va='bottom',ha='center')
```



[20]:
```
df.groupby('Product')['Total_Price'].agg('sum').sort_values(ascending=False).
↪astype('int32').to_frame()
```

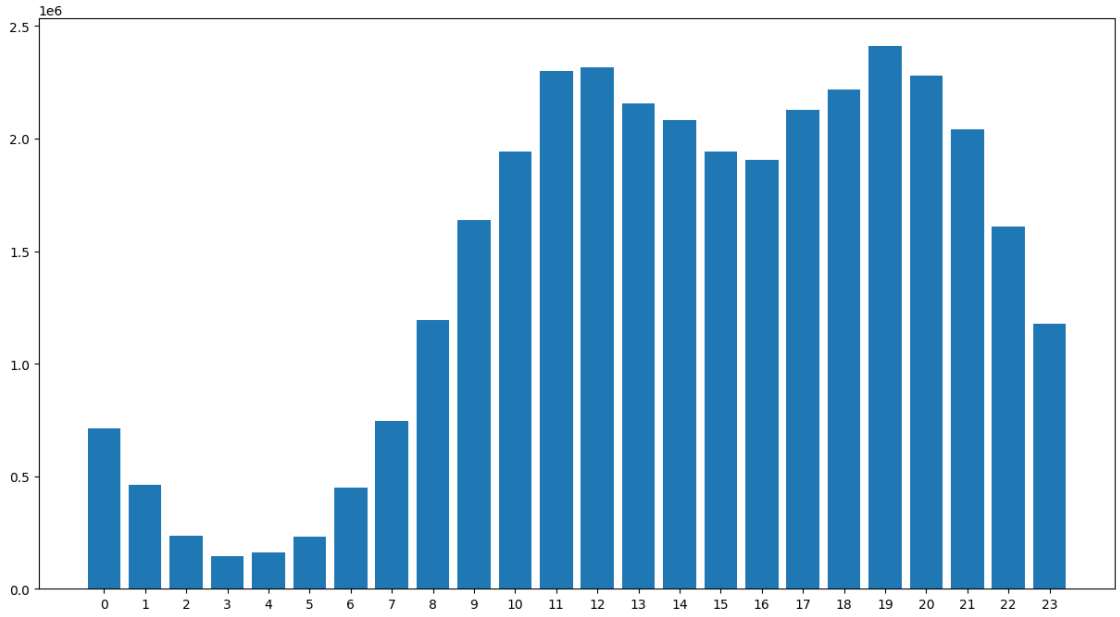
[20]:

Product	Total_Price
Macbook Pro Laptop	8037600
iPhone	4794300
ThinkPad Laptop	4129958
Google Phone	3319200
27in 4K Gaming Monitor	2435097
34in Ultrawide Monitor	2355558
Apple Airpods Headphones	2349150
Flatscreen TV	1445700
Bose SoundSport Headphones	1345565
27in FHD Monitor	1132424
Vareebadd Phone	827200
20in Monitor	454148
LG Washing Machine	399600
LG Dryer	387600
Lightning Charging Cable	347094
USB-C Charging Cable	286501
Wired Headphones	246478
AA Batteries (4-pack)	106118
AAA Batteries (4-pack)	92740

[21]:
```
df['Order Date']=pd.to_datetime(df['Order Date'])
```

[22]:
```
df['Hour']=df['Order Date'].dt.hour
```

[23]:
```
plt.figure(figsize=(15,8))
plt.bar(df.groupby('Hour')['Total_Price'].agg('sum').index,df.
  ↪groupby('Hour')['Total_Price'].agg('sum'))
plt.xticks(df.groupby('Hour')['Total_Price'].agg('sum').index);
```



[30]:
```
df['Hour']=df['Order Date'].dt.hour
df['Year']=df['Order Date'].dt.year
```

[33]:
```
df['Month_name']=df['Order Date'].dt.month_name()
```

[32]:
```
df['Year'].unique()
```

[32]:
```
array([2019, 2020])
```

[43]:
```
order=['January','February','March','April','May','June','July','August','September','October'
df['Month_name']=pd.Categorical(df['Month_name'],categories=order,ordered=True)
```

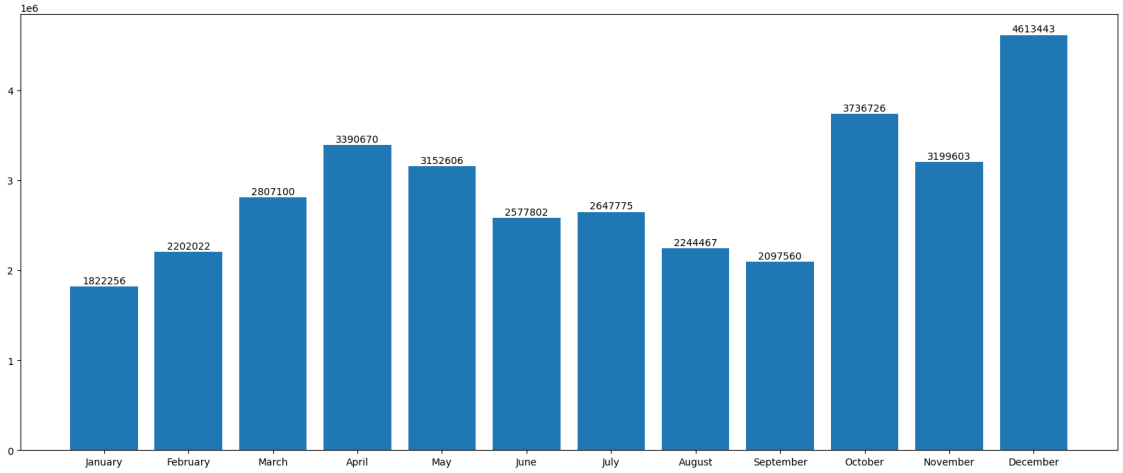
[44]:
```
df.groupby('Month_name')['Total_Price'].agg('sum').to_frame()
```

[44]:

Month_name	Total_Price
January	1822256.75
February	2202022.50
March	2807100.25
April	3390670.25
May	3152606.75
June	2577802.25

```
July          2647775.75
August        2244467.75
September     2097560.00
October       3736726.75
November      3199603.25
December      4613443.50
```

[46]:
```
plt.figure(figsize=(20,8))
plot=plt.bar(df.groupby('Month_name')['Total_Price'].agg('sum').index,df.
 ↪groupby('Month_name')['Total_Price'].agg('sum'))
for ractangle in plot:
    height=ractangle.get_height()
    width=ractangle.get_x()+ractangle.get_width()/2
    plt.text(width,height*1.003,'%d' %int(height),va='bottom',ha='center')
```



[194]:
```
grouped_data=df.groupby(['Year','Month_name'])['Total_Price'].agg('sum').
 ↪to_frame().reset_index()
grouped_data
```

[194]:

	Year	Month_name	Total_Price
0	2019	January	1.813586e+06
1	2019	February	2.202022e+06
2	2019	March	2.807100e+06
3	2019	April	3.390670e+06
4	2019	May	3.152607e+06
5	2019	June	2.577802e+06
6	2019	July	2.647776e+06
7	2019	August	2.244468e+06
8	2019	September	2.097560e+06
9	2019	October	3.736727e+06

```
10  2019   November  3.199603e+06
11  2019   December  4.613444e+06
12  2020    January  8.670290e+03
13  2020   February  0.000000e+00
14  2020      March  0.000000e+00
15  2020      April  0.000000e+00
16  2020        May  0.000000e+00
17  2020       June  0.000000e+00
18  2020       July  0.000000e+00
19  2020     August  0.000000e+00
20  2020  September  0.000000e+00
21  2020    October  0.000000e+00
22  2020   November  0.000000e+00
23  2020   December  0.000000e+00
```

[109]:
```
data=pd.DataFrame()
```

[185]:
```
a=df[(df['Year']==2019) &␣
 ↪(df['Month_name']=='February')][['Product','Total_Price']].
 ↪nlargest(5,'Total_Price').reset_index(drop=True)
a.insert(0,'Year',2019)
a.insert(0,'MOnth',"Feb")
```

[192]:
```
df[(df['Year']==2019) &␣
 ↪(df['Month_name']=='February')][['Product','Total_Price']].
 ↪reset_index(drop=True).groupby('Product')['Total_Price'].agg('sum').
 ↪nlargest(5).to_frame().reset_index()
```

[192]:
```
                  Product   Total_Price
0      Macbook Pro Laptop  469200.00000
1                  iPhone  307300.00000
2         ThinkPad Laptop  274997.25000
3            Google Phone  228600.00000
4  27in 4K Gaming Monitor  166525.71875
```

[204]:
```
year=[2019,2020]
mmm=['January','February','March','April','May','June','July','August','September','October','
def top_product(data):
    d1=pd.DataFrame()
    for i in year:
        for j in mmm:
            zz=data[(data['Year']==i) & (data['Month_name']==j)].
 ↪reset_index(drop=True).groupby('Product')['Total_Price'].agg('sum').
 ↪nsmallest(3).to_frame().reset_index()
            zz['Total_Price']=zz['Total_Price'].astype('int32')
            zz.insert(0,'Year',i)
            zz.insert(0,'Month',j)
```

```
            d1=pd.concat([d1,zz])

    return d1.reset_index(drop=True)
```

[205]:
```
df1=top_product(df)
```

[206]:
```
df1
```

[206]:

	Month	Year	Product	Total_Price
0	January	2019	AAA Batteries (4-pack)	4769
1	January	2019	AA Batteries (4-pack)	5468
2	January	2019	Wired Headphones	12961
3	February	2019	AAA Batteries (4-pack)	5896
4	February	2019	AA Batteries (4-pack)	6662
5	February	2019	Wired Headphones	15335
6	March	2019	AAA Batteries (4-pack)	7412
7	March	2019	AA Batteries (4-pack)	8509
8	March	2019	Wired Headphones	19951
9	April	2019	AAA Batteries (4-pack)	8787
10	April	2019	AA Batteries (4-pack)	10832
11	April	2019	Wired Headphones	24759
12	May	2019	AAA Batteries (4-pack)	8751
13	May	2019	AA Batteries (4-pack)	9154
14	May	2019	Wired Headphones	22493
15	June	2019	AAA Batteries (4-pack)	6464
16	June	2019	AA Batteries (4-pack)	8048
17	June	2019	Wired Headphones	17673
18	July	2019	AAA Batteries (4-pack)	7071
19	July	2019	AA Batteries (4-pack)	7952
20	July	2019	Wired Headphones	18764
21	August	2019	AAA Batteries (4-pack)	6042
22	August	2019	AA Batteries (4-pack)	7019
23	August	2019	Wired Headphones	15479
24	September	2019	AAA Batteries (4-pack)	5749
25	September	2019	AA Batteries (4-pack)	6800
26	September	2019	Wired Headphones	15598
27	October	2019	AAA Batteries (4-pack)	10055
28	October	2019	AA Batteries (4-pack)	11289
29	October	2019	Wired Headphones	27265
30	November	2019	AAA Batteries (4-pack)	9044
31	November	2019	AA Batteries (4-pack)	10080
32	November	2019	Wired Headphones	23188
33	December	2019	AAA Batteries (4-pack)	12680
34	December	2019	AA Batteries (4-pack)	14300
35	December	2019	Wired Headphones	32960
36	January	2020	AAA Batteries (4-pack)	14
37	January	2020	USB-C Charging Cable	47

```
38    January  2020        Wired Headphones          47
```

[ ]: