Assignment 16

kartik thakur

In [121]:
```
# 1. Merge the files in one dataframe.
# 2. Clean the data.
# 3. Change the object type column into integer type or float type.
# 4. Get the month value from the order date?
# 5. Which was the most productive month in terms of sales?
# 6.  Which city had the highest number of sales?
# 7. At what time people mostly purchase the product?
# 8. What is the average purchase by city?
# 9. Which product has the highest sales?
# 10. In Month of September, which product has the lowest sales?
```

In [122]:
```
# firstly we imported all files into on code and save in one dataframe
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

In [123]:
```
files=['Sales_January_2019.csv','Sales_February_2019.csv','Sales_March_2019.cs
dataframe=[]
for file in files:
    df=pd.read_csv(file)
    dataframe.append(df)
```

In [124]:
```
merge=pd.concat(dataframe,ignore_index=True)
```

In [125]:
```
merge
```

	ID	Product	Ordered	Each	Date	Purchase Address
0	141234	iPhone	1	700	01/22/19 21:25	944 Walnut St, Boston, MA 02215
1	141235	Lightning Charging Cable	1	14.95	01/28/19 14:15	185 Maple St, Portland, OR 97035
2	141236	Wired Headphones	2	11.99	01/17/19 13:33	538 Adams St, San Francisco, CA 94016
3	141237	27in FHD Monitor	1	149.99	01/05/19 20:33	738 10th St, Los Angeles, CA 90001
4	141238	Wired Headphones	1	11.99	01/25/19 11:59	387 10th St, Austin, TX 73301
...	...	...	...	...	...	...
186845	319666	Lightning Charging Cable	1	14.95	12/11/19 20:58	14 Madison St, San Francisco, CA 94016
186846	319667	AA Batteries (4-pack)	2	3.84	12/01/19 12:01	549 Willow St, Los Angeles, CA 90001

clean the data

In [126]:
```
data=merge.copy()
data
```

Out[126]:

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address
0	141234	iPhone	1	700	01/22/19 21:25	944 Walnut St, Boston, MA 02215
1	141235	Lightning Charging Cable	1	14.95	01/28/19 14:15	185 Maple St, Portland, OR 97035
2	141236	Wired Headphones	2	11.99	01/17/19 13:33	538 Adams St, San Francisco, CA 94016
3	141237	27in FHD Monitor	1	149.99	01/05/19 20:33	738 10th St, Los Angeles, CA 90001
4	141238	Wired Headphones	1	11.99	01/25/19 11:59	387 10th St, Austin, TX 73301
...	...	...	...	...	...	...
186845	319666	Lightning Charging Cable	1	14.95	12/11/19 20:58	14 Madison St, San Francisco, CA 94016
186846	319667	AA Batteries (4-pack)	2	3.84	12/01/19 12:01	549 Willow St, Los Angeles, CA 90001
186847	319668	Vareebadd Phone	1	400	12/09/19 06:43	273 Wilson St, Seattle, WA 98101
186848	319669	Wired Headphones	1	11.99	12/03/19 10:39	778 River St, Dallas, TX 75001
186849	319670	Bose SoundSport Headphones	1	99.99	12/21/19 21:45	747 Chestnut St, Los Angeles, CA 90001

186850 rows × 6 columns

In [127]:
```
data.isnull().sum()
```

Out[127]:
```
Order ID            545
Product             545
Quantity Ordered    545
Price Each          545
Order Date          545
Purchase Address    545
dtype: int64
```

In [128]:
```
data.dropna(inplace=True)
```

In [129]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 186305 entries, 0 to 186849
Data columns (total 6 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
dtypes: object(6)
memory usage: 9.9+ MB
```

In [130]:
```
data['Order ID']=pd.to_numeric(data['Order ID'],errors='coerce')
data.dropna(inplace=True)
```

In [131]:
```
# cleaning the order time column from duplicates and null values

data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 185950 entries, 0 to 186849
Data columns (total 6 columns):
 #   Column            Non-Null Count   Dtype  
---  ------            --------------   -----  
 0   Order ID          185950 non-null  float64
 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 
dtypes: float64(1), object(5)
memory usage: 9.9+ MB
```

3. Change the object type column into integer type or float type.

In [132]:
```
data[data["Price Each"]=="Price Each"].index
```

Out[132]:
```
Int64Index([], dtype='int64')
```

In [133]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 185950 entries, 0 to 186849
Data columns (total 6 columns):
 #   Column            Non-Null Count   Dtype  
---  ------            --------------   -----  
 0   Order ID          185950 non-null  float64
 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 
dtypes: float64(1), object(5)
memory usage: 9.9+ MB
```

In [134]:
```
data.drop(data[data["Price Each"]=="Price Each"].index,inplace=True)
```

In [135]:
```
data['Price Each']=data['Price Each'].astype('float')
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 185950 entries, 0 to 186849
Data columns (total 6 columns):
 #   Column            Non-Null Count   Dtype  
---  ------            --------------   -----  
 0   Order ID          185950 non-null  float64
 1   Product           185950 non-null  object 
 2   Quantity Ordered  185950 non-null  object 
 3   Price Each        185950 non-null  float64
 4   Order Date        185950 non-null  object 
 5   Purchase Address  185950 non-null  object 
dtypes: float64(2), object(4)
memory usage: 9.9+ MB
```

In [136]:
```
data[data["Price Each"]=="Price Each"].index
```

Out[136]:
```
Int64Index([], dtype='int64')
```

In [137]:
```
data["Price Each"]=data["Price Each"].astype('float64')
```

In [138]:
```
data["Order ID"]=data["Order ID"].astype('int64')
```

In [139]:
```
data["Quantity Ordered"]=data["Quantity Ordered"].astype('int32')
```

In [140]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 185950 entries, 0 to 186849
Data columns (total 6 columns):
 #   Column            Non-Null Count   Dtype  
---  ------            --------------   -----  
 0   Order ID          185950 non-null  int64  
 1   Product           185950 non-null  object 
 2   Quantity Ordered  185950 non-null  int32  
 3   Price Each        185950 non-null  float64
 4   Order Date        185950 non-null  object 
 5   Purchase Address  185950 non-null  object 
dtypes: float64(1), int32(1), int64(1), object(3)
memory usage: 9.2+ MB
```

4. Get the month value from the order date?

In [*]:
```
import datetime
month=pd.to_datetime(data['Order Date']).dt.month
month
```

In [*]:
```
data['month']=month
data
```

5. Which was the most productive month in terms of sales?

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

In [*]:
```
data.head()
```

In [*]:
```
a=data.groupby('month')["Sales"].sum().sort_values(ascending=False)
a
```

In [*]:
```
c_data = data.groupby(by = 'month', as_index = False )['Sales'].sum()
c_data = c_data.sort_values(by = 'Sales', ascending = False)
c_data.head()
```

In [*]:
```
sns.barplot(x='month',y='Sales',data=c_data,ci=None)
```

6. Which city had the highest number of sales?

In [*]:
```
data['city']=data['Purchase Address'].apply(lambda x: x.split(',')[1])
data.head()
```

In [*]:
```
c_data = data.groupby(by = 'city', as_index = False )['Sales'].sum()
c_data = c_data.sort_values(by = 'Sales', ascending = False)
c_data.head()
```

In [*]:
```
sns.barplot(x='city',y='Sales',data=c_data,ci=None)
plt.xticks(rotation=45);
```

7. At what time people mostly purchase the product?

In [*]:
```
hour= pd.to_datetime(data['Order Date']).dt.hour
hour
```

In [*]:
```
data['hour']=hour
data.head()
```

In [*]:
```
k=data.groupby(by='hour' ,as_index=False)['Quantity Ordered'].sum()
k.sort_values(by='Quantity Ordered' ,ascending=False)
```

In [*]:
```
data['hour'].value_counts().head()
```

In [*]:
```
sns.barplot(x='hour',y='Quantity Ordered',data=k,palette='magma')
plt.xticks(rotation=45);
```

8. What is the average purchase by city?

In [*]:
```
g=data.groupby(by='city' ,as_index=False)['Sales'].mean()
g.sort_values(by='Sales' ,ascending=False)
```

In [*]:
```
sns.barplot(x='city',y='Sales',data=g,palette='magma')
plt.xticks(rotation=45);
```

9. Which product has the highest sales?

In [*]:
```
g=data.groupby(by='Product' ,as_index=False)['Sales'].sum()
g.sort_values(by='Sales' ,ascending=False)
```

In [*]:
```
sns.barplot(x='Product',y='Sales',data=g,palette='magma');
plt.xticks(rotation=90);
```

In [*]:
```
sept=data[data['month']== 9]
sept
```

In [*]:
```
g=sept.groupby(by='Product' ,as_index=False)['Sales'].sum()
t=g.sort_values(by='Sales' ,ascending=False)
t
```

In [*]:
```
sns.barplot(x='Product',y='Sales',data=t,palette='magma')
plt.xticks(rotation=90);
```

In month of sept,AAA Batteries (4-Pack) has lowest sale

5. Which was the most productive month in terms of sales?

In [*]:
```
### December was the most productive month in terms of sales
```

6. Which city had the highest number of sales?

In [*]:
```
# San Francisco has the highest number of sales
```

7. At what time people mostly purchase the product?

In [*]:
```
# At 19:00 people mostly purchase the product
```

8. What is the average purchase by city?

In [*]:
```
# Atlanta has the maximum average purchase by the cities
```

9. Which product has the highest sales?

In [*]:
```
# Macbook Pro Laptop has the highest sales.
```

10. In Month of September, which product has the lowest sales?

In [*]:
```
# In month of september AAA Batteries (4-pack) has the lowest sale
```

In [ ]:

In [ ]: