ineuron-internship-project

September 30, 2023

0.0.1 Importing Libraries

[28]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import os
```

[10]:
```
df = pd.read_excel(r"./Data/SALESDATA.xlsx")
```

[29]:
```
df.head()
```

[29]:

	CustKey	DateKey	Discount Amount	Invoice Date	Invoice Number	\
0	10000481.0	2017-04-30	-237.91	2017-04-30	100012.0	
1	10002220.0	2017-07-14	368.79	2017-07-14	100233.0	
2	10002220.0	2017-10-17	109.73	2017-10-17	116165.0	
3	10002489.0	2017-06-03	-211.75	2017-06-03	100096.0	
4	10004516.0	2017-05-27	96627.94	2017-05-27	103341.0	

	Item Class	Item Number	Item	Line Number	\
0	NaN	NaN	Urban Large Eggs	2000.0	
1	P01	20910.0	Moms Sliced Turkey	1000.0	
2	P01	38076.0	Cutting Edge Foot-Long Hot Dogs	1000.0	
3	NaN	NaN	Kiwi Lox	1000.0	
4	P01	60776.0	High Top Sweet Onion	1000.0	

	List Price	…	Sales Amount	Sales Amount Based on List Price	\
0	0.00	…	237.91	0.00	
1	824.96	…	456.17	824.96	
2	548.66	…	438.93	548.66	
3	0.00	…	211.75	0.00	
4	408.52	…	89248.66	185876.60	

	Sales Cost Amount	Sales Margin Amount	Sales Price	Sales Quantity	\
0	0.0	237.91	237.910000	1.0	
1	0.0	456.17	456.170000	1.0	
2	0.0	438.93	438.930000	1.0	
3	0.0	211.75	211.750000	1.0	

```
4                0.0             89248.66   196.150901           455.0

   Sales Rep  U/M  @dropdown                              Unnamed: 21
0      184.0   EA        NaN                    U/M = unit of measure
1      127.0   EA        NaN                                     NaN
2      127.0   EA        NaN                                EA = each
3      160.0   EA        NaN                                     NaN
4      124.0   SE        NaN  SE = some SI unit like kgs or gallons

[5 rows x 22 columns]
```

[30]:
```
df.columns
```

[30]:
```
Index(['CustKey', 'DateKey', 'Discount Amount', 'Invoice Date',
       'Invoice Number', 'Item Class', 'Item Number', 'Item', 'Line Number',
       'List Price', 'Order Number', 'Promised Delivery Date', 'Sales Amount',
       'Sales Amount Based on List Price', 'Sales Cost Amount',
       'Sales Margin Amount', 'Sales Price', 'Sales Quantity', 'Sales Rep',
       'U/M', '@dropdown', 'Unnamed: 21'],
      dtype='object')
```

[31]:
```
df['Unnamed: 21'].value_counts()
```

[31]:
```
U/M = unit of measure                   1
EA = each                               1
SE = some SI unit like kgs or gallons   1
PR = pair                               1
Name: Unnamed: 21, dtype: int64
```

0.0.2 Removing Unnecessary Columns

[32]:
```
df.drop(["@dropdown"], axis = 1, inplace=True)
```

[33]:
```
df.drop(["Unnamed: 21"], axis = 1, inplace=True)
```

[34]:
```
df.drop(["DateKey"], axis = 1, inplace=True)
```

[35]:
```
df.head()
```

[35]:
```
       CustKey  Discount Amount Invoice Date  Invoice Number Item Class  \
0  10000481.0          -237.91   2017-04-30        100012.0        NaN
1  10002220.0           368.79   2017-07-14        100233.0        P01
2  10002220.0           109.73   2017-10-17        116165.0        P01
3  10002489.0          -211.75   2017-06-03        100096.0        NaN
4  10004516.0         96627.94   2017-05-27        103341.0        P01

  Item Number                           Item  Line Number  List Price  \
```

```
0         NaN                 Urban Large Eggs       2000.0        0.00
1     20910.0               Moms Sliced Turkey       1000.0      824.96
2     38076.0  Cutting Edge Foot-Long Hot Dogs       1000.0      548.66
3         NaN                         Kiwi Lox       1000.0        0.00
4     60776.0            High Top Sweet Onion       1000.0      408.52

   Order Number Promised Delivery Date  Sales Amount  \
0      200015.0              2017-04-30        237.91
1      200245.0              2017-07-14        456.17
2      213157.0              2017-10-16        438.93
3      200107.0              2017-06-03        211.75
4      203785.0              2017-05-28      89248.66

   Sales Amount Based on List Price  Sales Cost Amount  Sales Margin Amount  \
0                              0.00                0.0                237.91
1                            824.96                0.0                456.17
2                            548.66                0.0                438.93
3                              0.00                0.0                211.75
4                         185876.60                0.0              89248.66

   Sales Price  Sales Quantity  Sales Rep U/M
0   237.910000             1.0      184.0  EA
1   456.170000             1.0      127.0  EA
2   438.930000             1.0      127.0  EA
3   211.750000             1.0      160.0  EA
4   196.150901           455.0      124.0  SE
```

0.0.3 Adding Columns

[36]:
```
df['Year'] = df['Invoice Date'].dt.year
```

[37]:
```
df['Month'] = df['Invoice Date'].dt.month
```

[38]:
```
def quarter(x):
    if x in range(1, 4):
        return 'Q1'
    elif x in range(4, 7):
        return 'Q2'
    elif x in range(7, 10):
        return 'Q3'
    else:
        return 'Q4'



df['Quarter'] = df['Month'].apply(lambda x : quarter(x))
```

0.0.4 Saving the modified excel file

[ ]:
```
df.to_excel('Amazon Sales Data.xlsx')
```

0.0.5 Basic Exploration

[39]:
```
df.head(15)[['List Price', 'Sales Quantity', 'Sales Amount Based on List␣
↪Price', 'Discount Amount', 'Sales Amount', 'Sales Price']]
```

[39]:

	List Price	Sales Quantity	Sales Amount Based on List Price	Discount Amount	Sales Amount	Sales Price
0	0.0000	1.0	0.0000	-237.9100	237.91	237.910000
1	824.9600	1.0	824.9600	368.7900	456.17	456.170000
2	548.6600	1.0	548.6600	109.7300	438.93	438.930000
3	0.0000	1.0	0.0000	-211.7500	211.75	211.750000
4	408.5200	455.0	185876.6000	96627.9400	89248.66	196.150901
5	0.0000	1.0	0.0000	-1950.0000	1950.00	1950.000000
6	795.3140	1.0	795.3140	371.0140	424.30	424.300000
7	575.0000	2.0	1150.0000	608.0800	541.92	270.960000
8	51.8800	15.0	778.2000	424.8000	353.40	23.560000
9	412.0300	60.0	24721.8000	13492.8000	11229.00	187.150000
10	548.6600	35.0	19203.1000	10481.1000	8722.00	249.200000
11	50.5051	15.0	757.5765	404.1465	353.43	23.562000
12	1379.7938	2.0	2759.5876	1287.3476	1472.24	736.120000
13	1134.7700	9.0	10212.9300	4764.3300	5448.60	605.400000
14	0.0000	1.0	0.0000	-526.6400	526.64	526.640000

0.1 The relationships between the attributes are as follows:-

Sales Amount Based on List Price = List Price * Sales Quantity

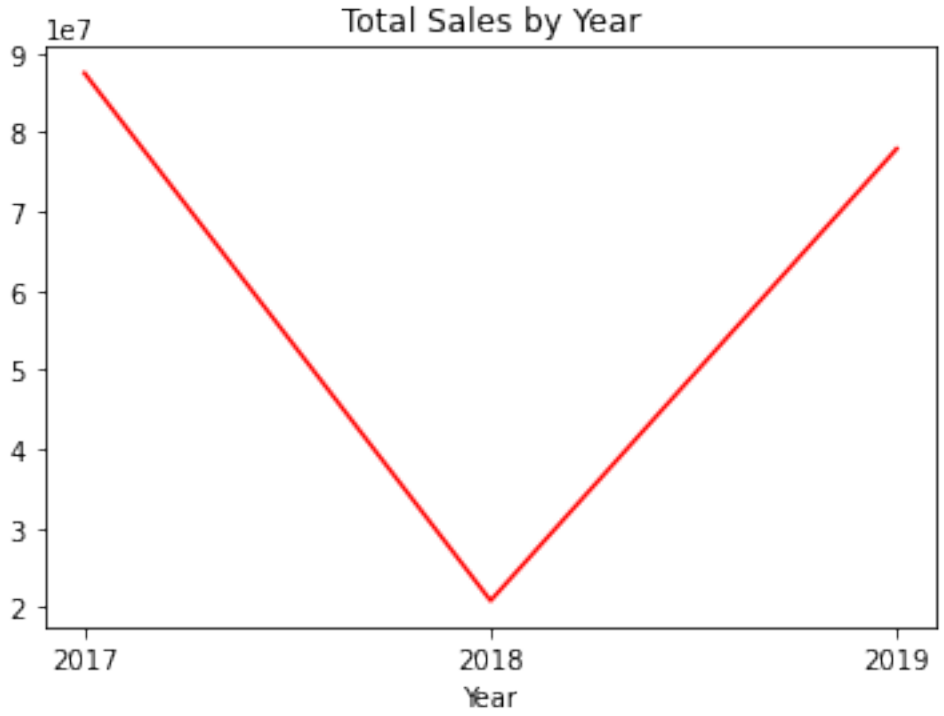
Sales Amount = Sales Amount Based on List Price - Discount Amount

Sales Price = Sales Amount/Sales Quantity

Sales Margin Amount = Sales Amount - Sales Cost Amount

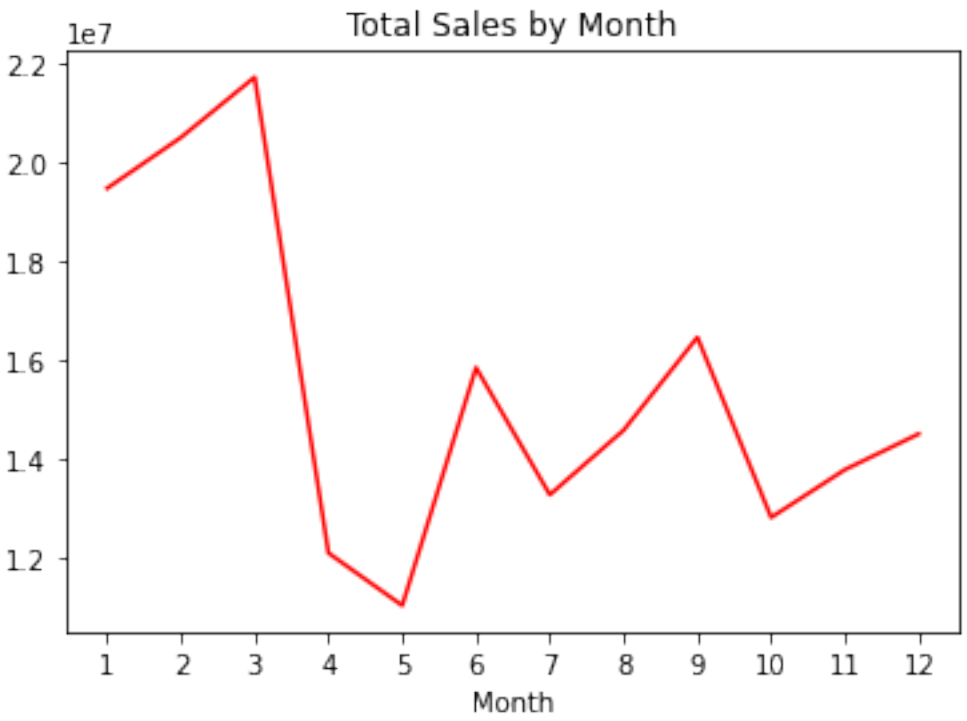
[40]:
```
df.groupby(by = 'Year')['Sales Amount'].sum().plot(kind = 'line', color = 'Red')
plt.xticks([2017, 2018, 2019])
plt.title("Total Sales by Year")
```

[40]: Text(0.5, 1.0, 'Total Sales by Year')



[42]:
```
df.groupby(by = 'Month')['Sales Amount'].sum().plot(kind = 'line', color =␣
↪'Red')
plt.xticks(np.arange(1,13))
plt.title("Total Sales by Month")
```

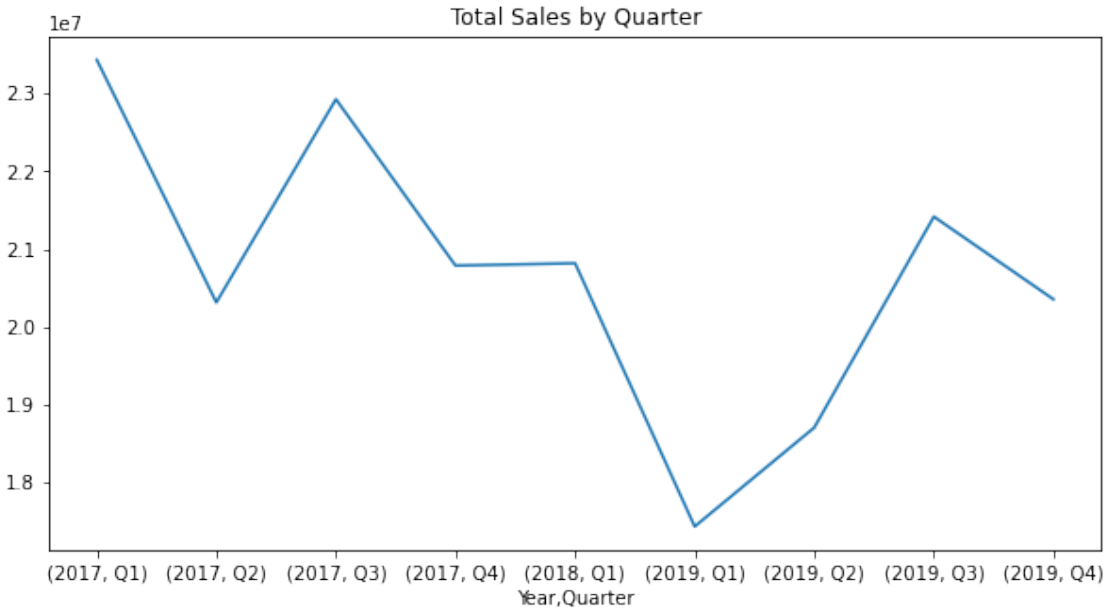
[42]: Text(0.5, 1.0, 'Total Sales by Month')



[43]:
```
#Total Sales by Quarter

plt.figure(figsize = (10,5))
plt.title("Total Sales by Quarter")
df.groupby(by = ['Year', 'Quarter'])['Sales Amount'].sum().plot()
```

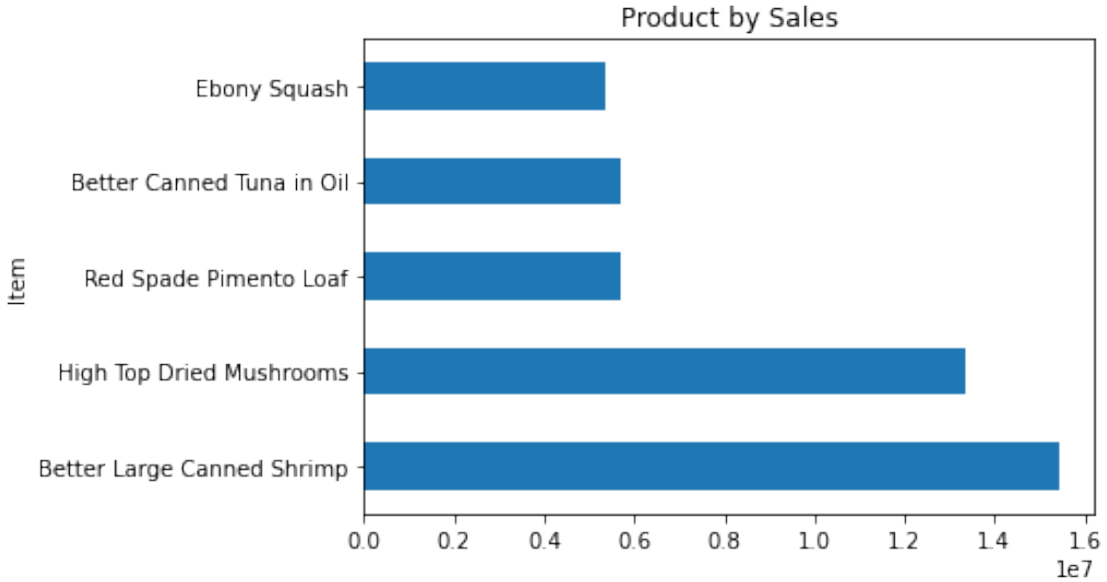
[43]: <AxesSubplot:title={'center':'Total Sales by Quarter'}, xlabel='Year,Quarter'>



[20]:
```
#Top 5 products with the highest Sales

df.groupby(by = 'Item')['Sales Amount'].sum().round().sort_values(ascending =␣
↪False)[:5].plot(kind = 'barh')
plt.title('Product by Sales')
```

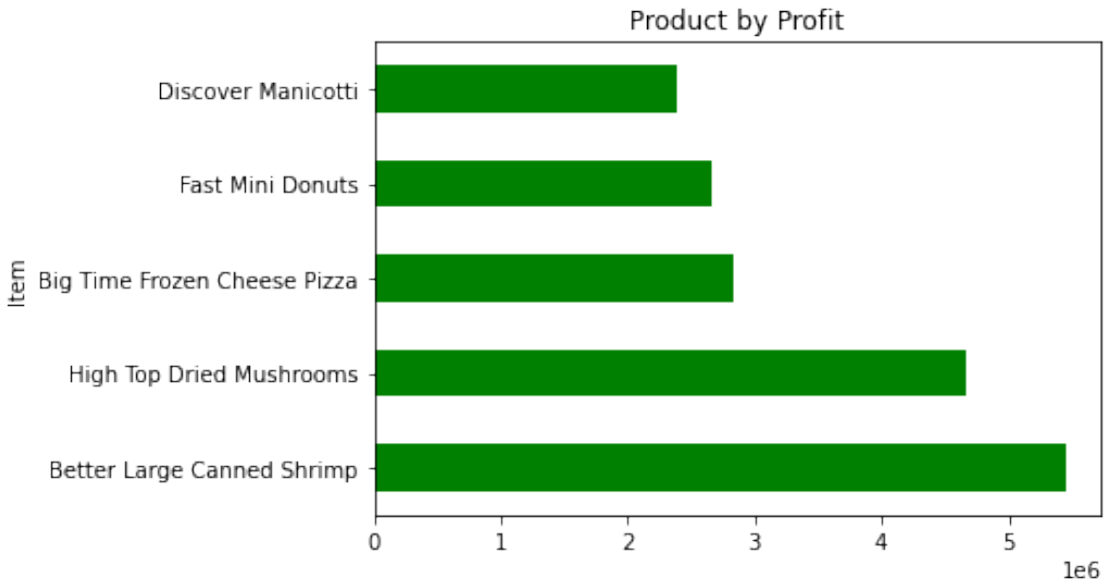
[20]:
```
Text(0.5, 1.0, 'Product by Sales')
```



[23]:
```
#Top 5 profit generating products

df.groupby(by = 'Item')['Sales Margin Amount'].sum().round().
 ↪sort_values(ascending = False)[:5].plot(kind = 'barh', color = 'green')
plt.title('Product by Profit')
```

[23]:
```
Text(0.5, 1.0, 'Product by Profit')
```



[ ]:

[ ]:

[ ]:

[ ]:

[ ]: