

# Blinkit Analysis

December 28, 2025

## 0.1 DATA ANALYSIS PYTHON PROJECT - BLINKIT ANALYSIS

### 0.1.1 Import Libraries

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

### 0.1.2 Import Raw Data

```
[2]: df = pd.read_csv("C:/Users/vijay laxmi/Downloads/BlinkIT Grocery Data.csv")
```

### 0.1.3 Sample Data

```
[3]: df.head(20)
```

```
[3]:   Item Fat Content Item Identifier Item Type \
0      Regular      FDX32 Fruits and Vegetables
1    Low Fat      NCB42  Health and Hygiene
2      Regular      FDR28    Frozen Foods
3      Regular      FDL50      Canned
4    Low Fat      DRI25    Soft Drinks
5    low fat      FDS52    Frozen Foods
6    Low Fat      NCU05  Health and Hygiene
7    Low Fat      NCD30    Household
8    Low Fat      FDW20 Fruits and Vegetables
9    Low Fat      FDX25      Canned
10         LF      FDX21    Snack Foods
11    Low Fat      NCU41  Health and Hygiene
12    Low Fat      FDL20 Fruits and Vegetables
13    Low Fat      NCR54    Household
14    Low Fat      FDH19      Meat
15    Regular      FDB57 Fruits and Vegetables
16    Low Fat      FD023      Breads
17    Low Fat      NCB07    Household
18    Low Fat      FDJ56 Fruits and Vegetables
19    Low Fat      DRN47    Hard Drinks
```

	Outlet Establishment Year	Outlet Identifier	Outlet Location Type	\
0	2012	OUT049	Tier 1	
1	2022	OUT018	Tier 3	
2	2010	OUT046	Tier 1	
3	2000	OUT013	Tier 3	
4	2015	OUT045	Tier 2	
5	2020	OUT017	Tier 2	
6	2011	OUT010	Tier 3	
7	2015	OUT045	Tier 2	
8	2000	OUT013	Tier 3	
9	1998	OUT027	Tier 3	
10	1998	OUT027	Tier 3	
11	2017	OUT035	Tier 2	
12	2022	OUT018	Tier 3	
13	2000	OUT013	Tier 3	
14	1998	OUT027	Tier 3	
15	2017	OUT035	Tier 2	
16	2022	OUT018	Tier 3	
17	2012	OUT049	Tier 1	
18	1998	OUT027	Tier 3	
19	2022	OUT018	Tier 3	

	Outlet Size	Outlet Type	Item Visibility	Item Weight	Sales	\
0	Medium	Supermarket Type1	0.100014	15.10	145.4786	
1	Medium	Supermarket Type2	0.008596	11.80	115.3492	
2	Small	Supermarket Type1	0.025896	13.85	165.0210	
3	High	Supermarket Type1	0.042278	12.15	126.5046	
4	Small	Supermarket Type1	0.033970	19.60	55.1614	
5	Small	Supermarket Type1	0.005505	8.89	102.4016	
6	Small	Grocery Store	0.098312	11.80	81.4618	
7	Small	Supermarket Type1	0.026904	19.70	96.0726	
8	High	Supermarket Type1	0.024129	20.75	124.1730	
9	Medium	Supermarket Type3	0.101562	NaN	181.9292	
10	Medium	Supermarket Type3	0.084555	NaN	109.8912	
11	Small	Supermarket Type1	0.052045	18.85	192.1846	
12	Medium	Supermarket Type2	0.128938	17.10	112.3886	
13	High	Supermarket Type1	0.090487	16.35	195.2110	
14	Medium	Supermarket Type3	0.032928	NaN	173.1738	
15	Small	Supermarket Type1	0.018802	20.25	222.1772	
16	Medium	Supermarket Type2	0.147024	17.85	93.7436	
17	Medium	Supermarket Type1	0.077628	19.20	197.6110	
18	Medium	Supermarket Type3	0.182515	NaN	98.7700	
19	Medium	Supermarket Type2	0.016895	12.10	178.5660	

	Rating
0	5.0

```

1      5.0
2      5.0
3      5.0
4      5.0
5      5.0
6      5.0
7      5.0
8      5.0
9      5.0
10     5.0
11     5.0
12     5.0
13     5.0
14     5.0
15     5.0
16     5.0
17     5.0
18     5.0
19     5.0

```

```
[4]: df.tail(20)
```

```

[4]:      Item Fat Content Item Identifier      Item Type \
8503      Regular      FDR22      Snack Foods
8504      Regular      FDS09      Snack Foods
8505      Regular      FDS34      Snack Foods
8506      Regular      FDU09      Snack Foods
8507      Regular      FDU33      Snack Foods
8508      Regular      FDU57      Snack Foods
8509      Regular      FDU58      Snack Foods
8510      Regular      FDX46      Snack Foods
8511      Regular      FDX57      Snack Foods
8512      Regular      FDY33      Snack Foods
8513      Regular      DRY23      Soft Drinks
8514      low fat      FDA11      Baking Goods
8515      low fat      FDK38      Canned
8516      low fat      FD038      Canned
8517      low fat      FDG32      Fruits and Vegetables
8518      low fat      NCT53      Health and Hygiene
8519      low fat      FDN09      Snack Foods
8520      low fat      DRE13      Soft Drinks
8521      reg      FDT50      Dairy
8522      reg      FDM58      Snack Foods

      Outlet Establishment Year Outlet Identifier Outlet Location Type \
8503      1998      OUT027      Tier 3
8504      1998      OUT027      Tier 3

```

8505	1998	OUT027	Tier 3
8506	1998	OUT027	Tier 3
8507	1998	OUT027	Tier 3
8508	1998	OUT027	Tier 3
8509	1998	OUT027	Tier 3
8510	1998	OUT027	Tier 3
8511	1998	OUT027	Tier 3
8512	1998	OUT027	Tier 3
8513	1998	OUT027	Tier 3
8514	1998	OUT027	Tier 3
8515	1998	OUT027	Tier 3
8516	1998	OUT027	Tier 3
8517	1998	OUT027	Tier 3
8518	1998	OUT027	Tier 3
8519	1998	OUT027	Tier 3
8520	1998	OUT027	Tier 3
8521	1998	OUT027	Tier 3
8522	1998	OUT027	Tier 3

	Outlet Size	Outlet Type	Item Visibility	Item Weight	Sales \
8503	Medium	Supermarket Type3	0.018473	NaN	109.9544
8504	Medium	Supermarket Type3	0.080696	NaN	51.3008
8505	Medium	Supermarket Type3	0.076387	NaN	112.1518
8506	Medium	Supermarket Type3	0.066275	NaN	54.1956
8507	Medium	Supermarket Type3	0.134057	NaN	45.3402
8508	Medium	Supermarket Type3	0.089121	NaN	149.8708
8509	Medium	Supermarket Type3	0.028871	NaN	188.7898
8510	Medium	Supermarket Type3	0.057835	NaN	57.5562
8511	Medium	Supermarket Type3	0.047037	NaN	96.4068
8512	Medium	Supermarket Type3	0.096730	NaN	159.0262
8513	Medium	Supermarket Type3	0.108568	NaN	42.9112
8514	Medium	Supermarket Type3	0.043029	NaN	94.7436
8515	Medium	Supermarket Type3	0.053032	NaN	149.1734
8516	Medium	Supermarket Type3	0.072486	NaN	78.9986
8517	Medium	Supermarket Type3	0.175143	NaN	222.3772
8518	Medium	Supermarket Type3	0.000000	NaN	164.5526
8519	Medium	Supermarket Type3	0.034706	NaN	241.6828
8520	Medium	Supermarket Type3	0.027571	NaN	86.6198
8521	Medium	Supermarket Type3	0.107715	NaN	97.8752
8522	Medium	Supermarket Type3	0.000000	NaN	112.2544

	Rating
8503	4.0
8504	4.0
8505	4.0
8506	4.0
8507	4.0

8508	4.0
8509	4.0
8510	4.0
8511	4.0
8512	4.0
8513	4.0
8514	4.0
8515	4.0
8516	4.0
8517	4.0
8518	4.0
8519	4.0
8520	4.0
8521	4.0
8522	4.0

#### 0.1.4 Size of the Data

```
[5]: print("size of data:",df.shape)
```

```
size of data: (8523, 12)
```

#### 0.1.5 Field Info

```
[6]: df.columns
```

```
[6]: Index(['Item Fat Content', 'Item Identifier', 'Item Type',
         'Outlet Establishment Year', 'Outlet Identifier',
         'Outlet Location Type', 'Outlet Size', 'Outlet Type', 'Item Visibility',
         'Item Weight', 'Sales', 'Rating'],
         dtype='object')
```

#### 0.1.6 Data Types

```
[7]: df.dtypes
```

```
[7]: Item Fat Content      object
     Item Identifier      object
     Item Type            object
     Outlet Establishment Year  int64
     Outlet Identifier      object
     Outlet Location Type    object
     Outlet Size           object
     Outlet Type           object
     Item Visibility        float64
     Item Weight           float64
     Sales                 float64
     Rating                float64
```

dtype: object

### 0.1.7 Data Cleaning

```
[8]: print(df['Item Fat Content'].unique())
```

```
['Regular' 'Low Fat' 'low fat' 'LF' 'reg']
```

```
[9]: df['Item Fat Content'] = df['Item Fat Content'].replace({'LF': 'Low Fat', 'low_fat': 'Low Fat', 'reg': 'Regular'})
```

```
[10]: print(df['Item Fat Content'].unique())
```

```
['Regular' 'Low Fat']
```

### 0.1.8 BUSINESS REQUIREMENTS

#### 0.1.9 KPI's REQUIREMENTS

```
[11]: #Total Sales
total_sales = df['Sales'].sum()

#Avg Sales
Avg_sales = df['Sales'].mean()

#no of Items sold
no_of_Items_sold = df['Sales'].count()

#Avg Ratings
avg_rating = df['Rating'].mean()

#Display

print(f"Total Sales: ${total_sales:,.0f}")
print(f"Avg Sales: ${Avg_sales:,.1f}")
print(f"No of Items Sold: {no_of_Items_sold:,.0f}")
print(f"Avg Ratings: {avg_rating:,.1f}")
```

Total Sales: \$1,201,681

Avg Sales: \$141.0

No of Items Sold: 8,523

Avg Ratings: 4.0

#### 0.1.10 CHART REQUIREMENTS

##### Total Sales By Fat Content

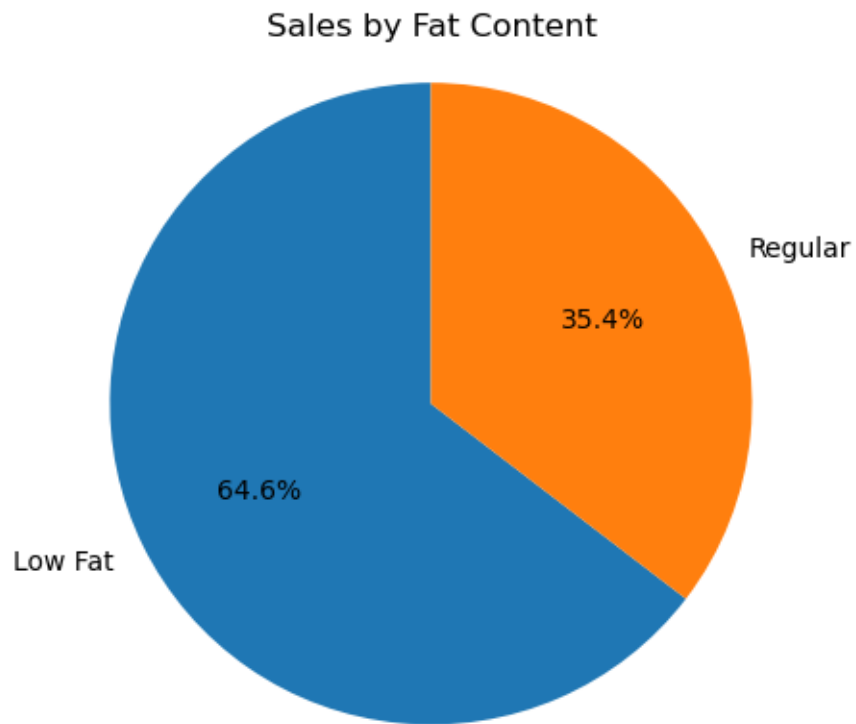
```
[12]: sales_by_fat = df.groupby('Item Fat Content')['Sales'].sum()

plt.pie(sales_by_fat, labels = sales_by_fat.index,
```

```

        autopct = '%.1f%%',
        startangle = 90)
plt.title('Sales by Fat Content')
plt.axis('equal')
plt.show()

```



### Total Sales By Item Type

```

[13]: sales_by_type = df.groupby('Item Type')['Sales'].sum().
      ↪ sort_values(ascending=False)

plt.figure(figsize=(10, 6))
bars = plt.bar(sales_by_type.index, sales_by_type.values)

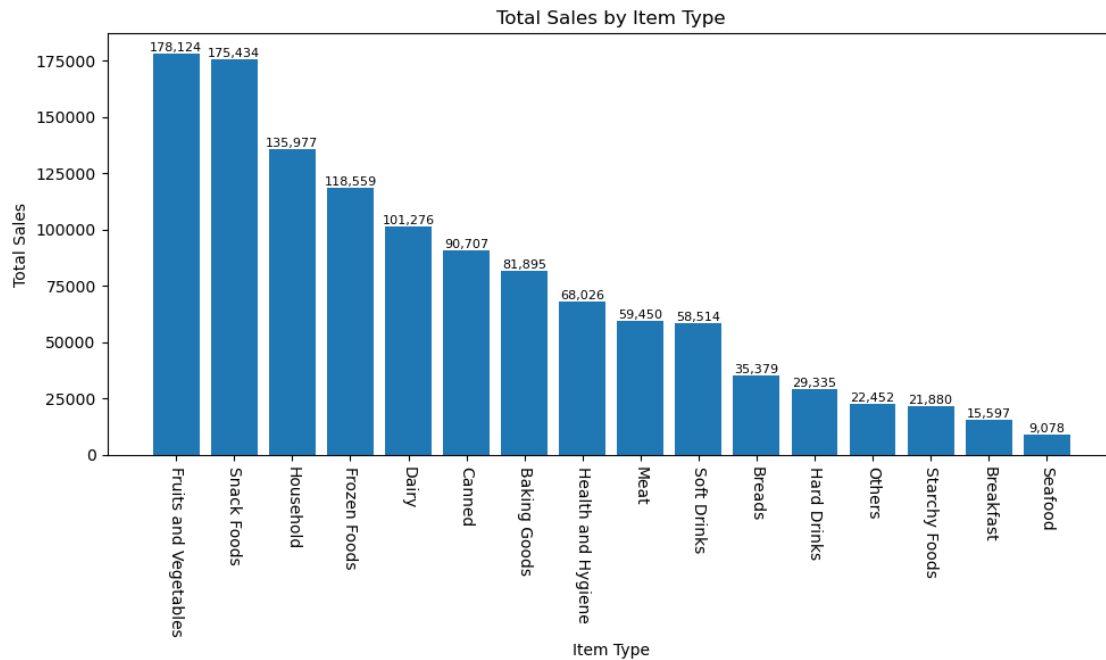
plt.xticks(rotation=-90)
plt.xlabel('Item Type')
plt.ylabel('Total Sales')
plt.title('Total Sales by Item Type')

for bar in bars:
    plt.text(bar.get_x() + bar.get_width() / 2, bar.get_height(),

```

```
f'{bar.get_height():,.0f}', ha = 'center', va = 'bottom', fontsize=
↵= 8)
```

```
plt.tight_layout()
plt.show()
```

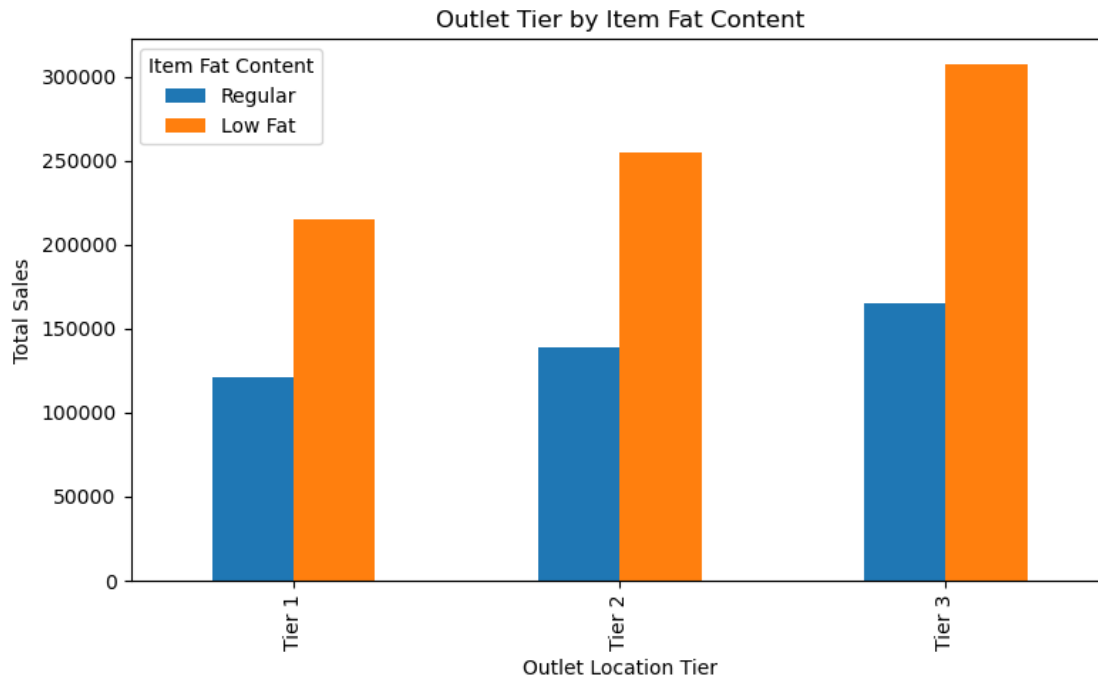


### Fat Content by Outlet for Total Sales

```
[14]: grouped = df.groupby(['Outlet Location Type', 'Item Fat Content'])['Sales'].
↵sum().unstack()
grouped = grouped[['Regular', 'Low Fat']]

ax = grouped.plot(kind='bar', figsize=(8, 5), title='Outlet Tier by Item Fat_
↵Content')
plt.xlabel('Outlet Location Tier')
plt.ylabel('Total Sales')
plt.legend(title='Item Fat Content')
plt.tight_layout()
plt.show()
```





### Total Sales by Outlet Establishment

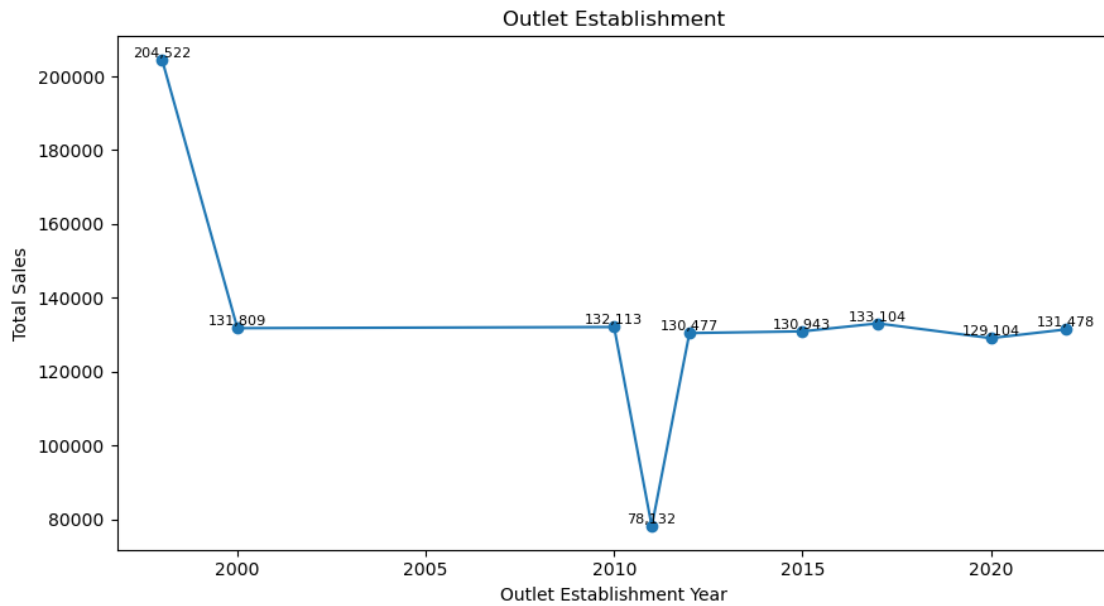
```
[15]: sales_by_year = df.groupby('Outlet Establishment Year')['Sales'].sum().
      ↪sort_index()

plt.figure(figsize=(9,5))
plt.plot(sales_by_year.index, sales_by_year.values, marker = 'o', linestyle = '—',
      ↪'-')

plt.xlabel('Outlet Establishment Year')
plt.ylabel('Total Sales')
plt.title('Outlet Establishment')

for x,y in zip(sales_by_year.index, sales_by_year.values):
    plt.text(x,y,f'{y:,.0f}',ha= 'center', va = 'bottom', fontsize = 8)

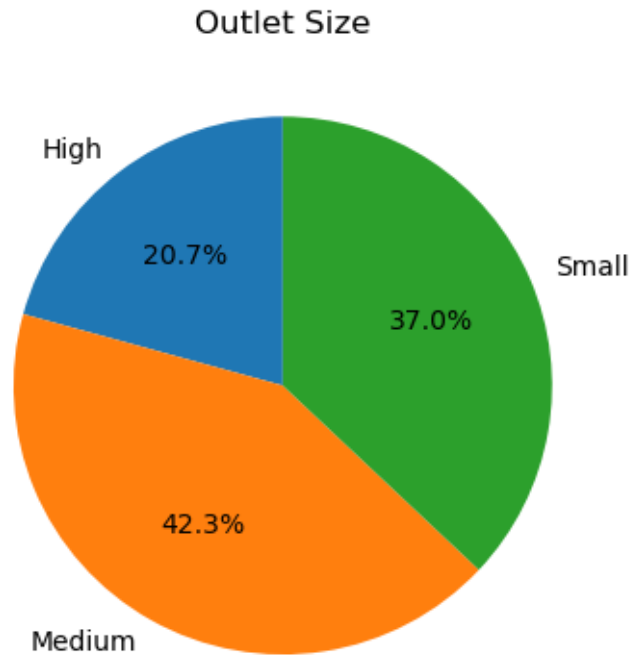
plt.tight_layout()
plt.show()
```



### Sales by Outlet Size

```
[16]: sales_by_size = df.groupby('Outlet Size')['Sales'].sum()

plt.figure(figsize=(4,4))
plt.pie(sales_by_size ,labels=sales_by_size.index, autopct = '%1.1f%%',
        ↪startangle =90)
plt.title('Outlet Size')
plt.tight_layout()
plt.show()
```



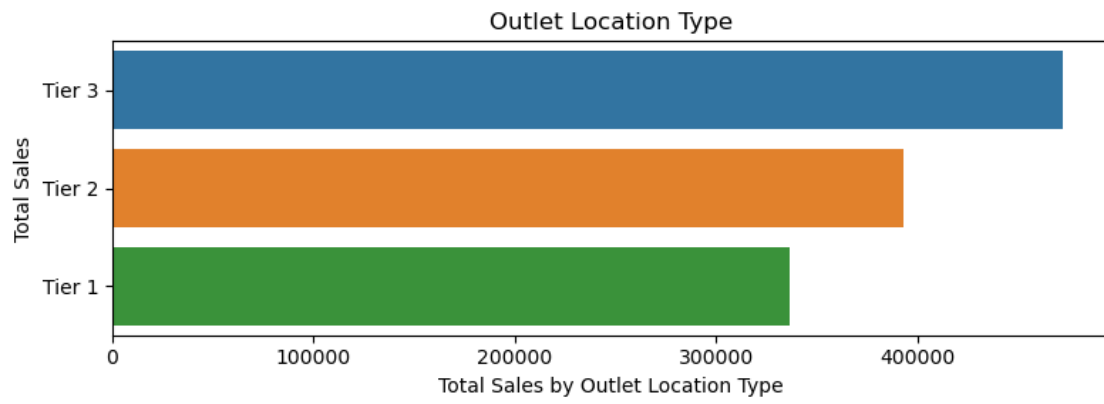
#### Sales by Outlet Location

```
[17]: sales_by_location = df.groupby('Outlet Location Type')['Sales'].sum().
      ↪reset_index()
sales_by_location = sales_by_location.sort_values('Sales', ascending = False)

plt.figure(figsize =(8,3))
ax = sns.barplot(x='Sales' ,y='Outlet Location Type', data= sales_by_location)

plt.xlabel('Total Sales by Outlet Location Type')
plt.ylabel('Total Sales')
plt.title('Outlet Location Type')

plt.tight_layout()
plt.show()
```



[ ]: