DATA ANALYSIS PYTHON PROJECT - BLINKIT ANALYSIS

Import Libraries

In [1]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

Import Raw Data

In [10]:

```
df = pd.read_csv("C:/Users/oluwa/Desktop/Data Analysis/BlinkIT Grocery Data.csv")
```

Sample Data

In [15]:

```
df.head(20)
```

Out[15]:

	Item Fat Content	Item Identifier	Item Type	Outlet Establishment Year	Outlet Identifier	Outlet Location Type	Outlet Size	Outlet Type	Item Visibility
0	Regular	FDX32	Fruits and Vegetables	2012	OUT049	Tier 1	Medium	Supermarket Type1	0.100014
1	Low Fat	NCB42	Health and Hygiene	2022	OUT018	Tier 3	Medium	Supermarket Type2	0.008596
2	Regular	FDR28	Frozen Foods	2016	OUT046	Tier 1	Small	Supermarket Type1	0.025896
3	Regular	FDL50	Canned	2014	OUT013	Tier 3	High	Supermarket Type1	0.042278
4	Low Fat	DRI25	Soft Drinks	2015	OUT045	Tier 2	Small	Supermarket Type1	0.033970
5	low fat	FDS52	Frozen Foods	2020	OUT017	Tier 2	Small	Supermarket Type1	0.005505
6	Low Fat	NCU05	Health and Hygiene	2011	OUT010	Tier 3	Small	Grocery Store	0.098312
7	Low Fat	NCD30	Household	2015	OUT045	Tier 2	Small	Supermarket Type1	0.026904
8	Low Fat	FDW20	Fruits and Vegetables	2014	OUT013	Tier 3	High	Supermarket Type1	0.024129
9	Low Fat	FDX25	Canned	2018	OUT027	Tier 3	Medium	Supermarket Type3	0.101562

	Item Fat Content	Item Identifier	Item Type	Outlet Establishment Year	Outlet Identifier	Outlet Location Type	Outlet Size	Outlet Type	Item Visibility
10	LF	FDX21	Snack Foods	2018	OUT027	Tier 3	Medium	Supermarket Type3	0.084555
11	Low Fat	NCU41	Health and Hygiene	2017	OUT035	Tier 2	Small	Supermarket Type1	0.052045
12	Low Fat	FDL20	Fruits and Vegetables	2022	OUT018	Tier 3	Medium	Supermarket Type2	0.128938
13	Low Fat	NCR54	Household	2014	OUT013	Tier 3	High	Supermarket Type1	0.090487
14	Low Fat	FDH19	Meat	2018	OUT027	Tier 3	Medium	Supermarket Type3	0.032928
15	Regular	FDB57	Fruits and Vegetables	2017	OUT035	Tier 2	Small	Supermarket Type1	0.018802
16	Low Fat	FDO23	Breads	2022	OUT018	Tier 3	Medium	Supermarket Type2	0.147024
17	Low Fat	NCB07	Household	2012	OUT049	Tier 1	Medium	Supermarket Type1	0.077628
18	Low Fat	FDJ56	Fruits and Vegetables	2018	OUT027	Tier 3	Medium	Supermarket Type3	0.182515
19	Low Fat	DRN47	Hard Drinks	2022	OUT018	Tier 3	Medium	Supermarket Type2	0.016895

Size of Data

In [20]:

```
print("Size of Data:", df.shape)
```

```
Size of Data: (8523, 12)
```

Field Info

In [22]:

```
df.columns
```

Out[22]:

```
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')
```

Data Types

In [25]:

```
df.dtypes
```

Out[25]:

```
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
```

Data Cleaning

In [30]:

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

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

In [36]:

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

In [38]:

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

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

BUSINESS REQUIREMENTS

KPI REQUIREMENTS

In [48]:

```
#Blinkit Total Sales

total_sales = df['Sales'].sum()

#Blinkit Average Sales

avg_sales = df['Sales'].mean()

#Number of items Sold

No_of_item_sold = df['Sales'].count()

#Average Ratings

avg_ratings = df['Rating'].mean()

#Display

print(f"Total Sales: ${total_sales:,.0f}")
print(f"Average Sales: ${avg_sales:,.0f}")
```

```
print(f"Number of Items Sold: {No_of_item_sold:.0f}")
print(f"Average Ratings: {avg_ratings:,.1f}")
```

```
Total Sales: $1,201,681
Average Sales: $141
Number of Items Sold: 8523
Average Ratings: 4.0
```

CHART REQUIREMENTS

Total Sales By Fat Content

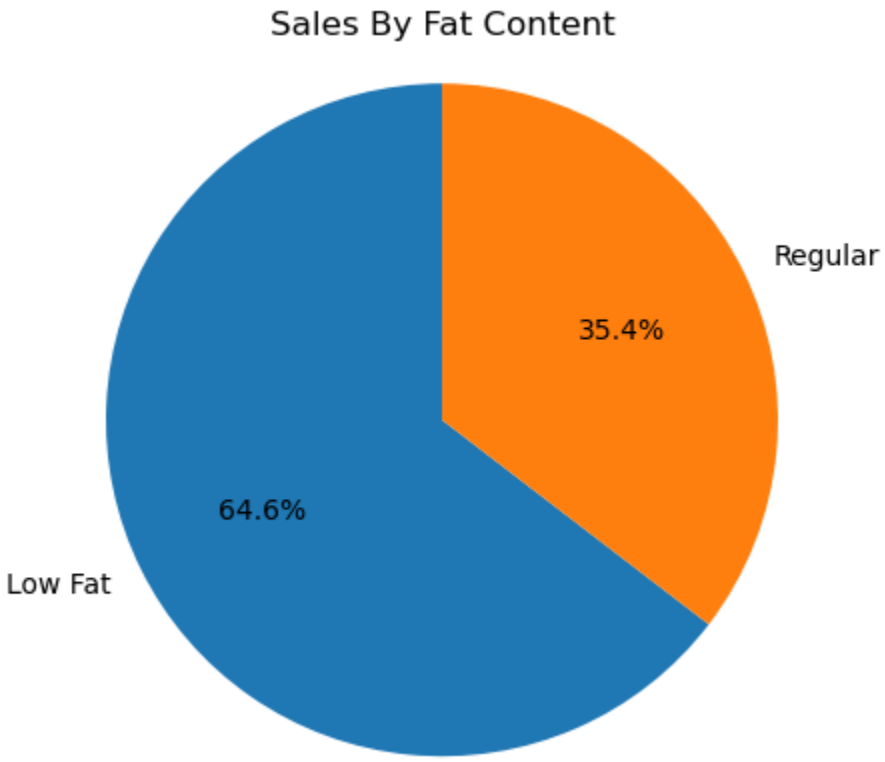
In [57]:

```
sales_by_fat_content = df.groupby('Item Fat Content')['Sales'].sum()

plt.pie(sales_by_fat_content, labels= sales_by_fat_content.index,
                                autopct = '%.1f%%',
                                startangle = 90)

plt.title('Sales By Fat Content')
plt.axis('equal')

plt.show()
```



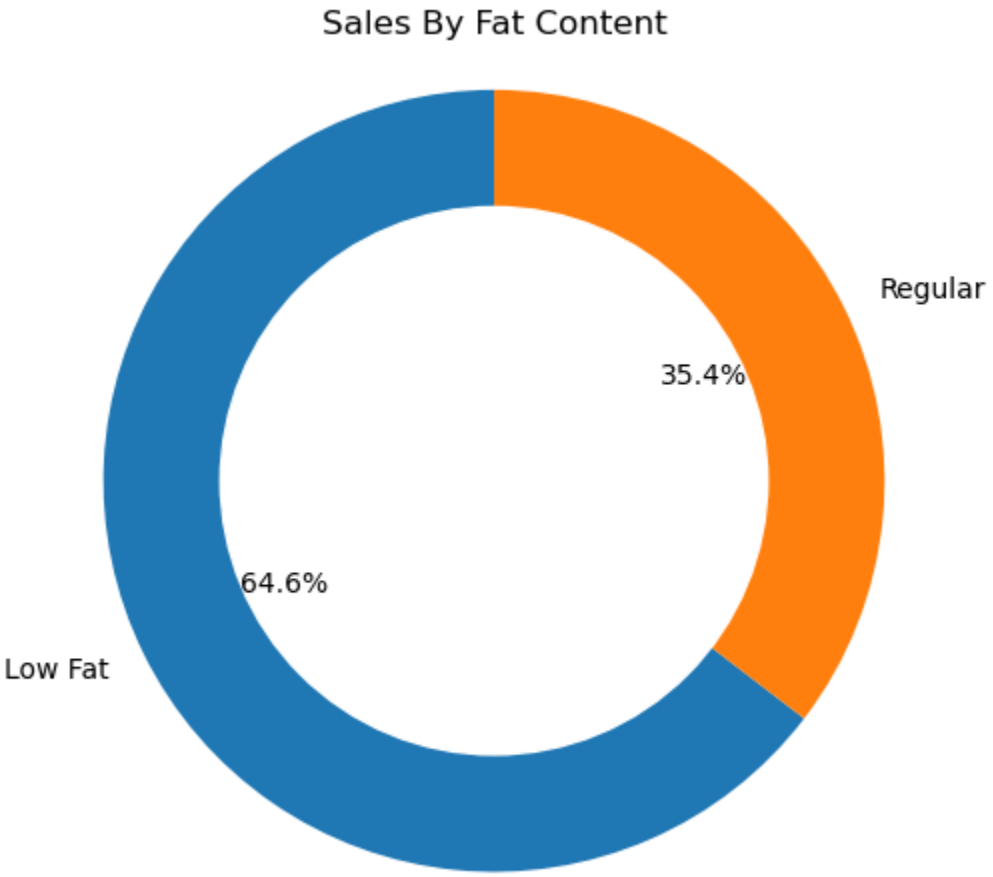
In [75]:

```
sales_by_fat_content = df.groupby('Item Fat Content')['Sales'].sum()

plt.pie(sales_by_fat_content, labels= sales_by_fat_content.index,
                                autopct = '%.1f%%',
                                startangle = 90)

plt.title('Sales By Fat Content')
plt.axis('equal')
```

```
my_circle = plt.Circle((0,0), 0.7, color ='white')
p=plt.gcf()
p.gca().add_artist(my_circle)
plt.tight_layout()
plt.show()
```



Total Sales By Item Type

In [80]:

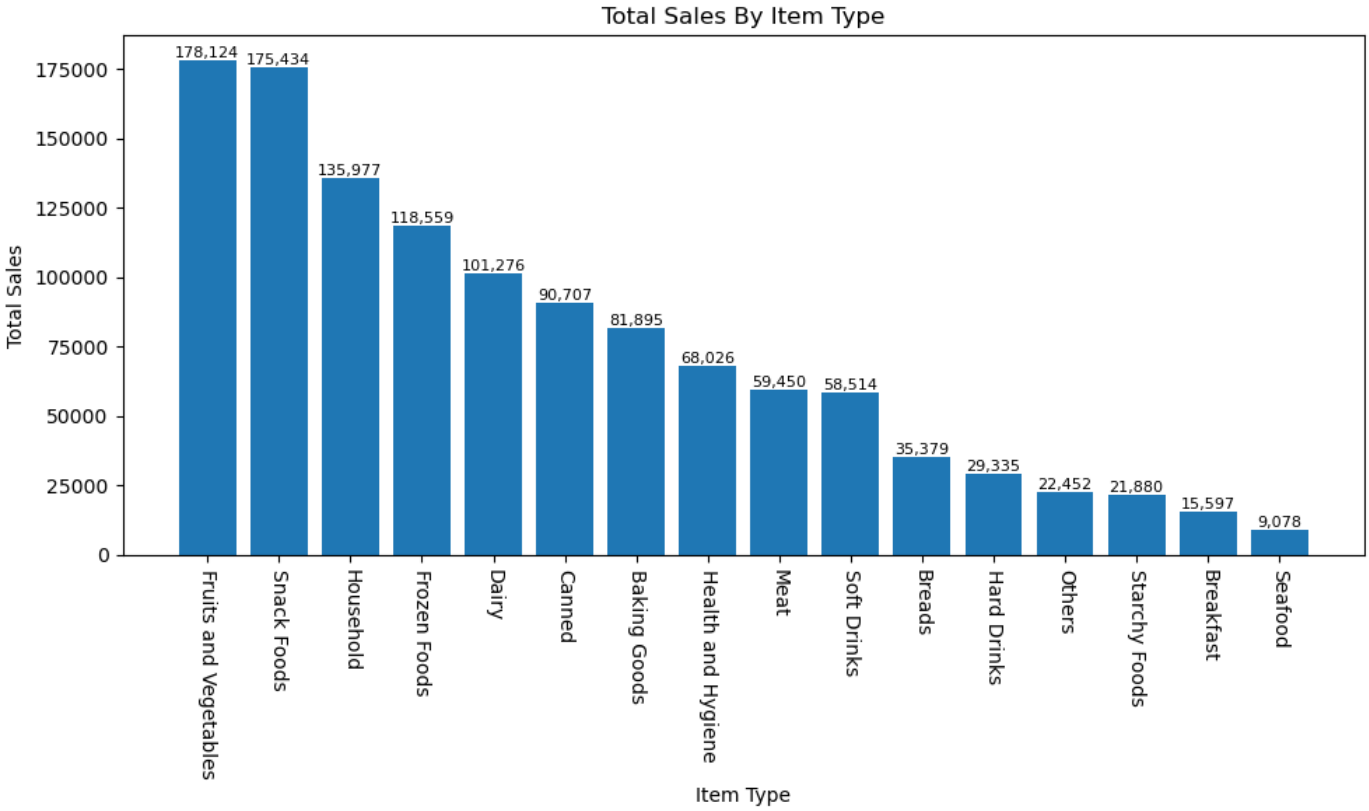
```
sales_by_type = df.groupby('Item Type')['Sales'].sum().sort_values(ascending=False)

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()
```

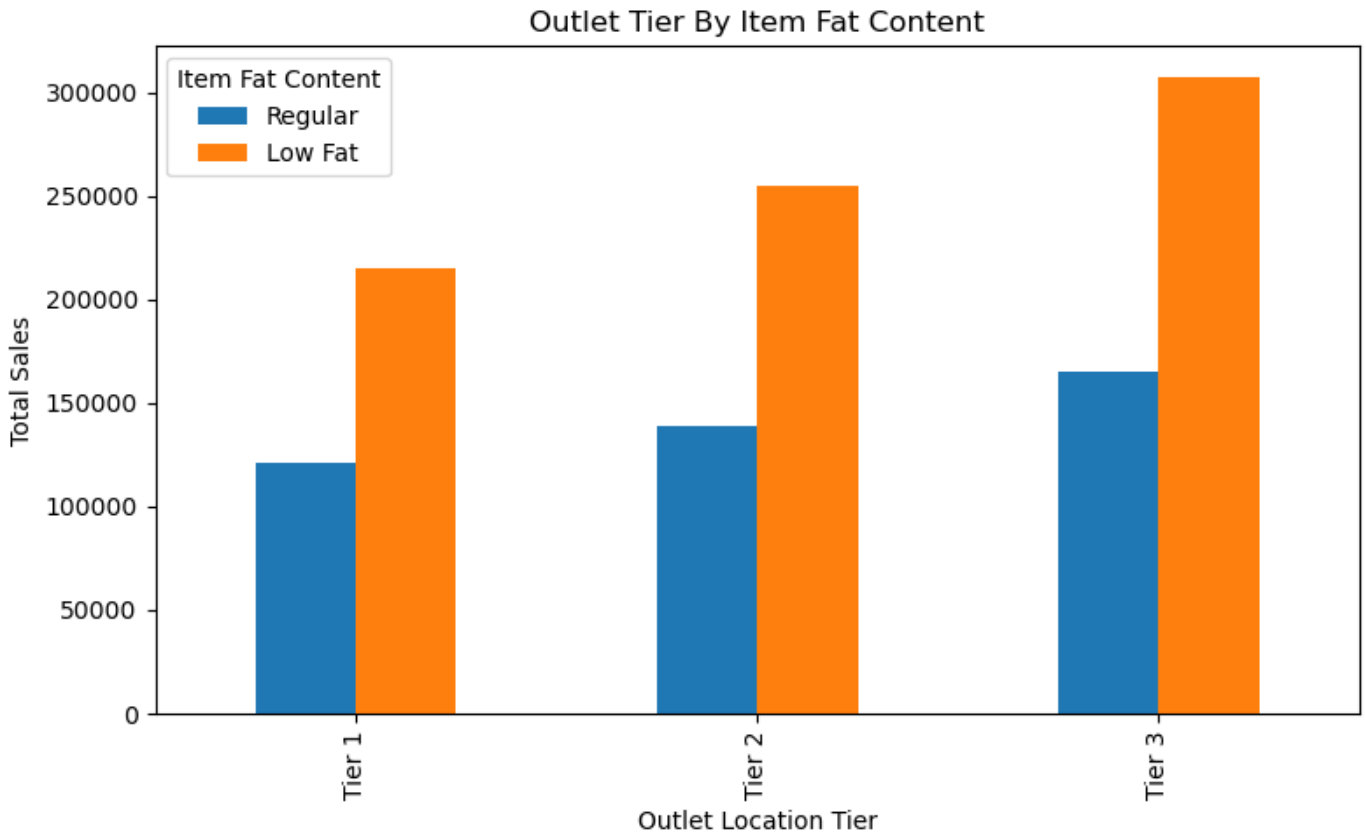


Fat Content By Outlet For Total Sales

In [91]:

```
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

In [94]:

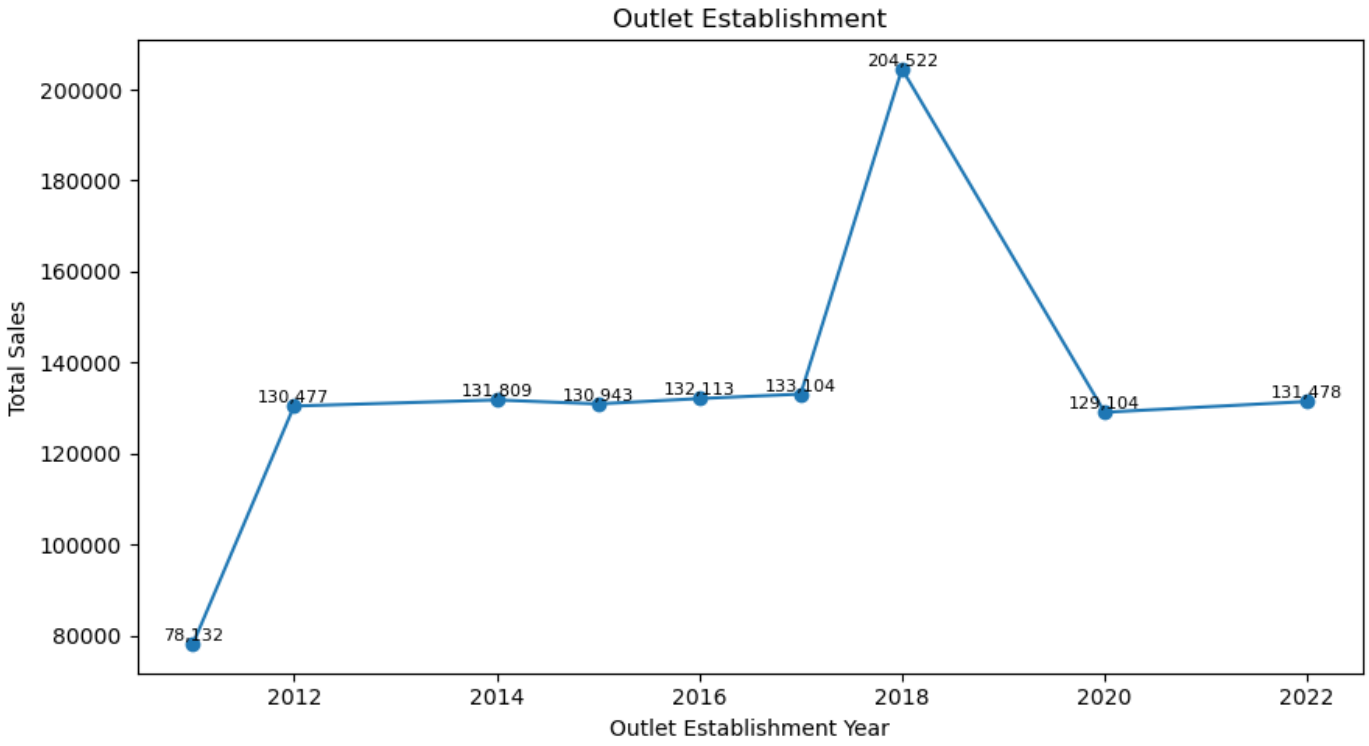
```
sales_by_year = df.groupby('Outlet Establishment Year')['Sales'].sum().sort_index()

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()
```

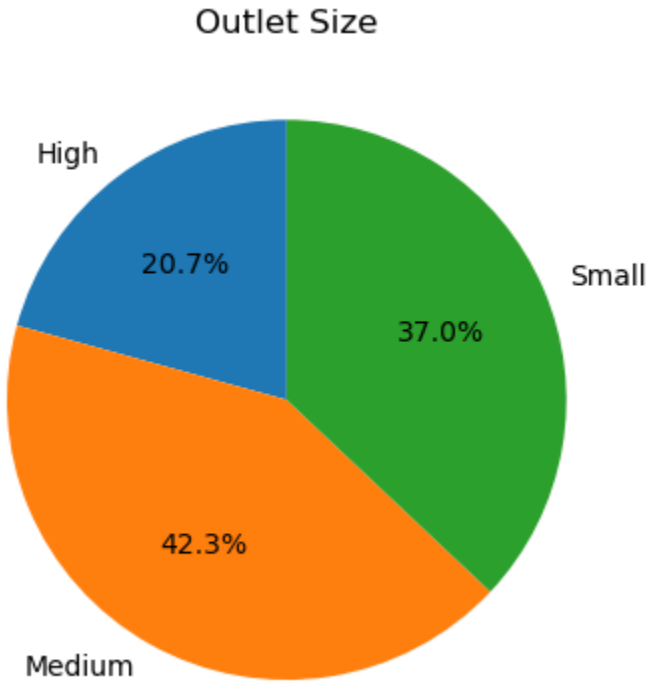


Sales By Outlet Size

In [102]:

```
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

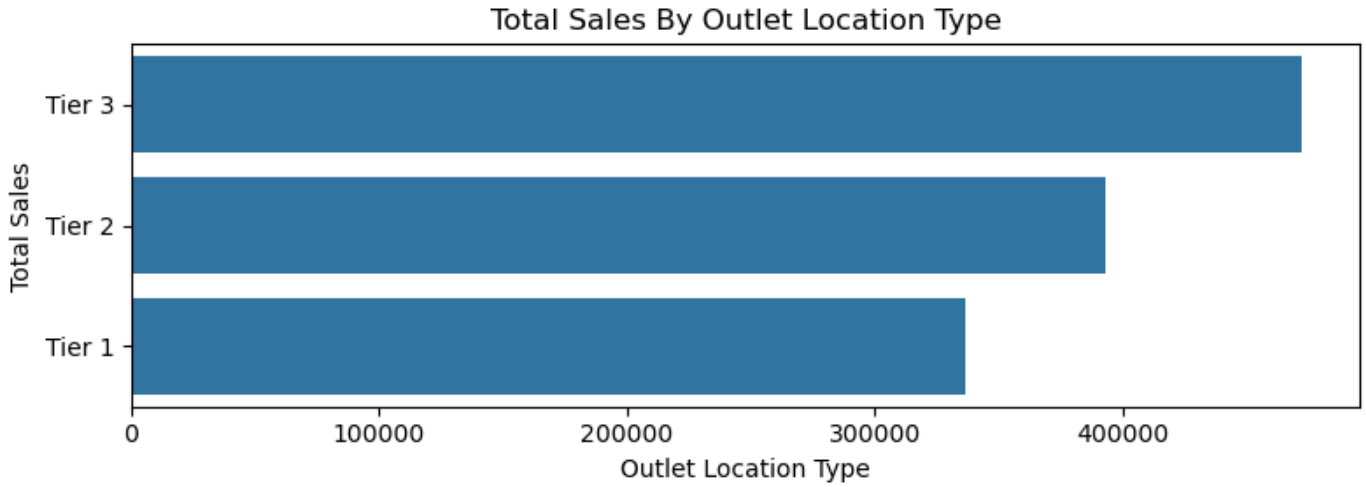
In [107]:

```
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('Outlet Location Type')
plt.ylabel('Total Sales')
plt.title('Total Sales By Outlet Location Type')


plt.tight_layout()   #Ensures layout fits without scroll
plt.show()
```



In [ ]:

```

```