Data Analysis Project -Blinkit Analysis

Import Libraries

In [7]:

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

Import Raw Data

In [8]:

```
df = pd.read_csv(r"E:\PYTHON\Blinkit.csv")
```

Sample Data

In [11]:

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

Out[11]:

	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
0	Regular	FDX32	Fruits and Vegetables	2012	OUT049	Tier 1	Medium	Supermarket Type1	0.100014	15.10	145.4786	5.0
1	Low Fat	NCB42	Health and Hygiene	2022	OUT018	Tier 3	Medium	Supermarket Type2	0.008596	11.80	115.3492	5.0
2	Regular	FDR28	Frozen Foods	2010	OUT046	Tier 1	Small	Supermarket Type1	0.025896	13.85	165.0210	5.0
3	Regular	FDL50	Canned	2000	OUT013	Tier 3	High	Supermarket Type1	0.042278	12.15	126.5046	5.0
4	Low Fat	DRI25	Soft Drinks	2015	OUT045	Tier 2	Small	Supermarket Type1	0.033970	19.60	55.1614	5.0
5	low fat	FDS52	Frozen Foods	2020	OUT017	Tier 2	Small	Supermarket Type1	0.005505	8.89	102.4016	5.0
6	Low Fat	NCU05	Health and Hygiene	2011	OUT010	Tier 3	Small	Grocery Store	0.098312	11.80	81.4618	5.0
7	Low Fat	NCD30	Household	2015	OUT045	Tier 2	Small	Supermarket Type1	0.026904	19.70	96.0726	5.0
8	Low Fat	FDW20	Fruits and Vegetables	2000	OUT013	Tier 3	High	Supermarket Type1	0.024129	20.75	124.1730	5.0
9	Low Fat	FDX25	Canned	1998	OUT027	Tier 3	Medium	Supermarket Type3	0.101562	NaN	181.9292	5.0
10	LF	FDX21	Snack Foods	1998	OUT027	Tier 3	Medium	Supermarket Type3	0.084555	NaN	109.8912	5.0
11	Low Fat	NCU41	Health and Hygiene	2017	OUT035	Tier 2	Small	Supermarket Type1	0.052045	18.85	192.1846	5.0
12	Low Fat	FDL20	Fruits and Vegetables	2022	OUT018	Tier 3	Medium	Supermarket Type2	0.128938	17.10	112.3886	5.0

	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
13	Low Fat	NCR54	Household	2000	OUT013	Tier 3	High	Supermarket Type1	0.090487	16.35	195.2110	5.0
14	Low Fat	FDH19	Meat	1998	OUT027	Tier 3	Medium	Supermarket Type3	0.032928	NaN	173.1738	5.0
15	Regular	FDB57	Fruits and Vegetables	2017	OUT035	Tier 2	Small	Supermarket Type1	0.018802	20.25	222.1772	5.0
16	Low Fat	FDO23	Breads	2022	OUT018	Tier 3	Medium	Supermarket Type2	0.147024	17.85	93.7436	5.0
17	Low Fat	NCB07	Household	2012	OUT049	Tier 1	Medium	Supermarket Type1	0.077628	19.20	197.6110	5.0
18	Low Fat	FDJ56	Fruits and Vegetables	1998	OUT027	Tier 3	Medium	Supermarket Type3	0.182515	NaN	98.7700	5.0
19	Low Fat	DRN47	Hard Drinks	2022	OUT018	Tier 3	Medium	Supermarket Type2	0.016895	12.10	178.5660	5.0

Size of Data

In [89]:

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

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

Field info

In [13]:

```
df.columns
```

Out[13]:

```
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')
```

In [15]:

```
df.dtypes
```

Out[15]:

```
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 [17]:

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

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

In [19]:

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

In [132…

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

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

Business Requirements

KPI's Requirements

In [21]:

```
# 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_ratings = df['Rating'].mean()

# Display

print(f"Total Sales: ${total_sales:,.0f}")
print(f"AVG_Sales: ${avg_sales:,.0f}")
print(f"No_of_items_sold: {no_of_items_sold:,.0f}")
print(f"AVG_ratings: {avg_ratings:,.1f}")
```

```
Total Sales: $1,201,681
AVG_Sales: $141
No_of_items_sold: 8,523
AVG_ratings: 4.0
```

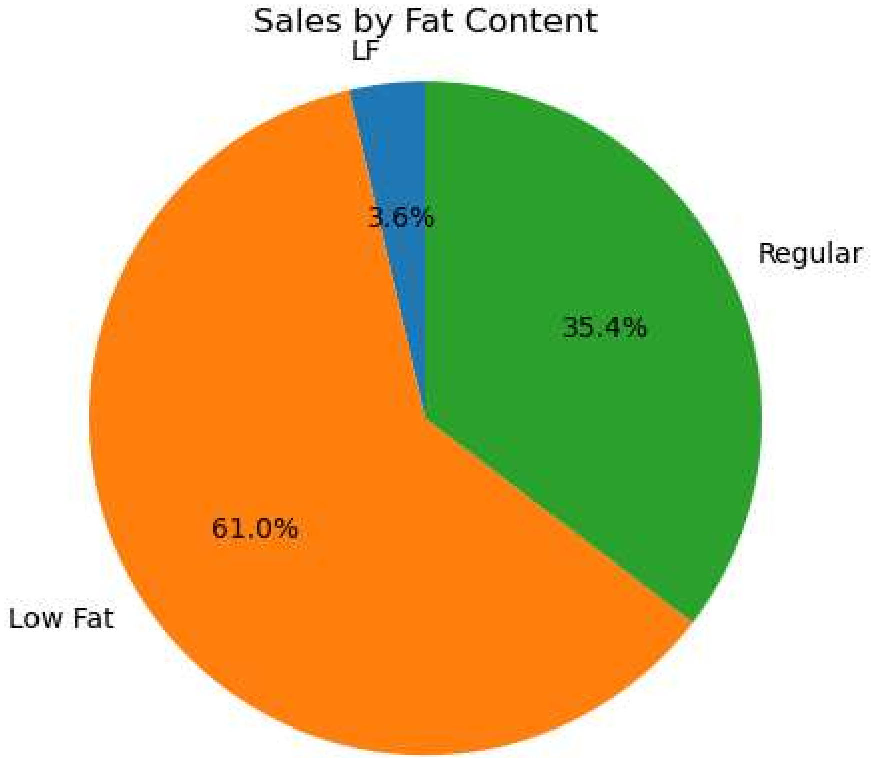
Chart's Requirements

Total Sales by Fat Content

In [23]:

```
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

In [25]:

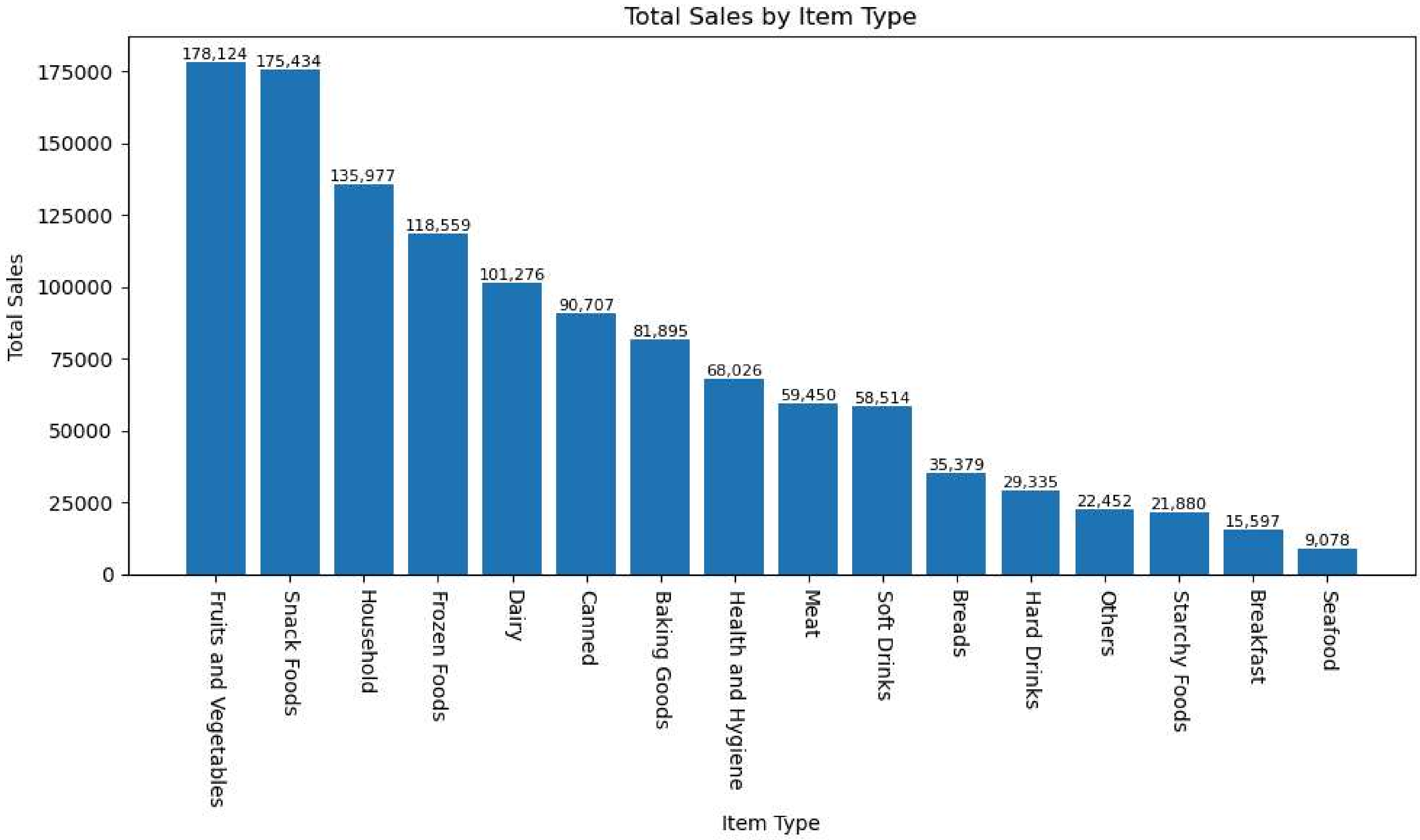
```python
sales_by_type = df.groupby('Item Type')['Sales'].sum().sort_values(ascending=False)

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()
```

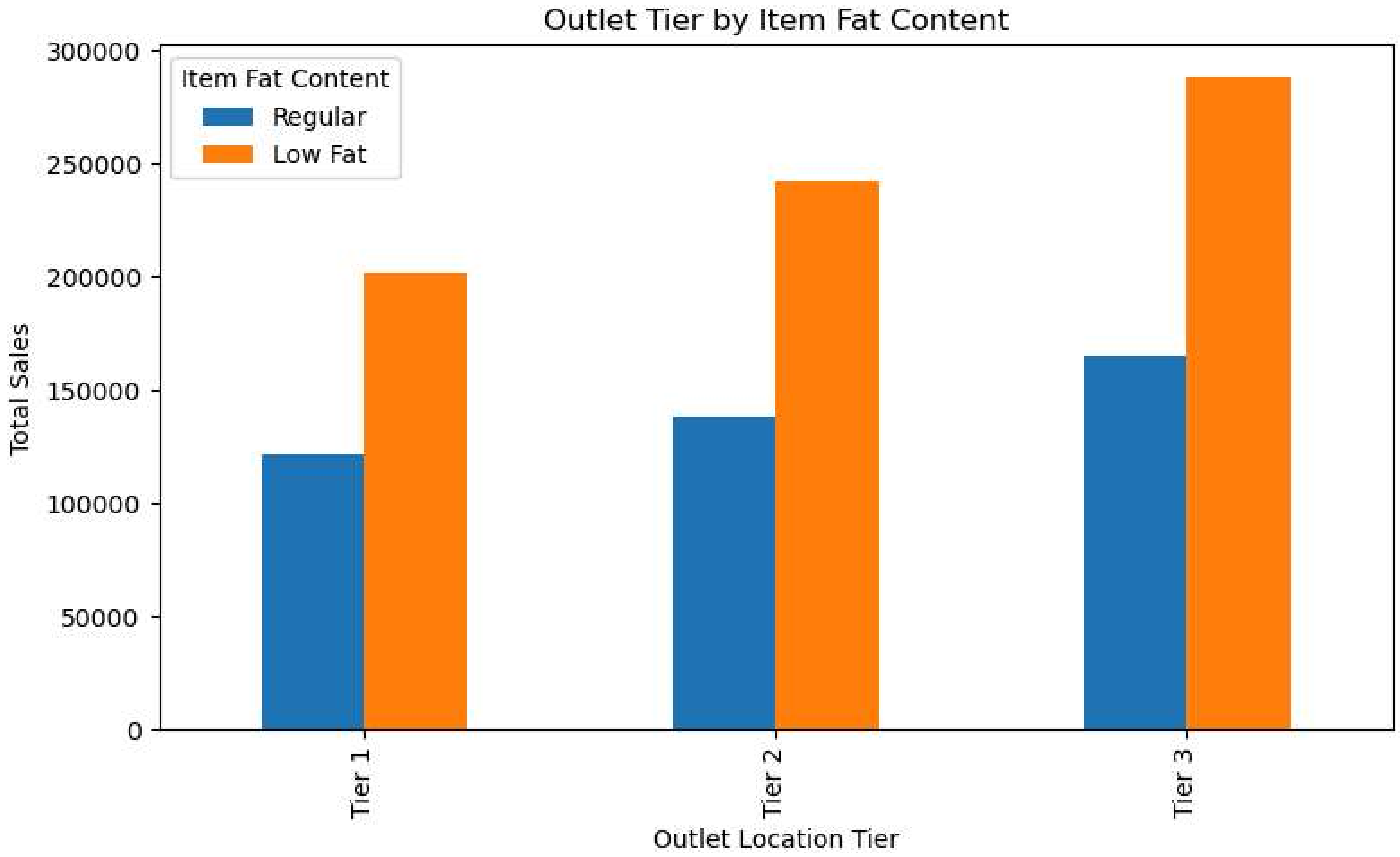


Fat Content by Outlet for Total Sales

In [27]:

```
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 [40]:

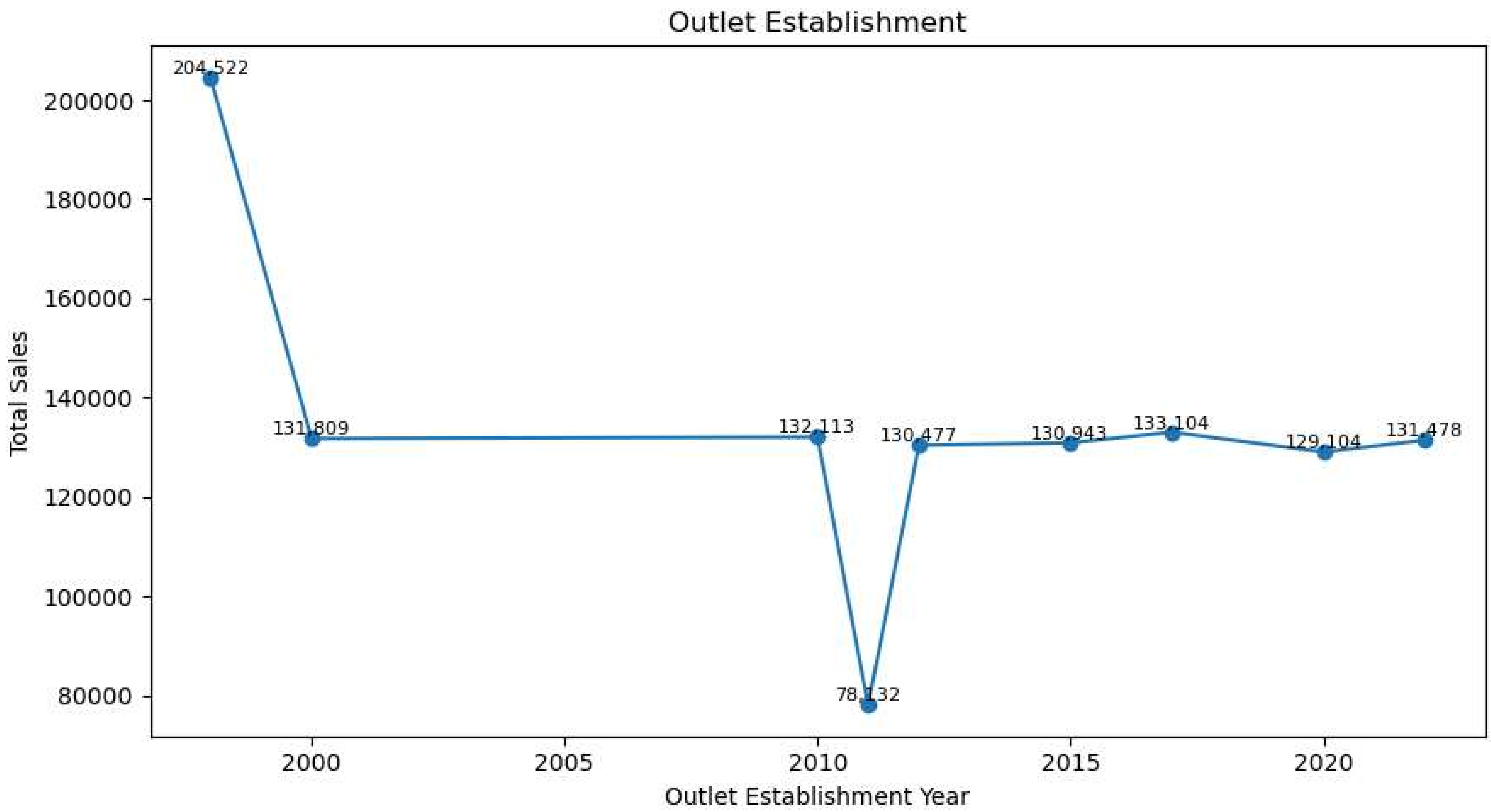
```
sales_by_year = df.groupby('Outlet Establishment Year')['Sales'].sum().sort_index()

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()
```

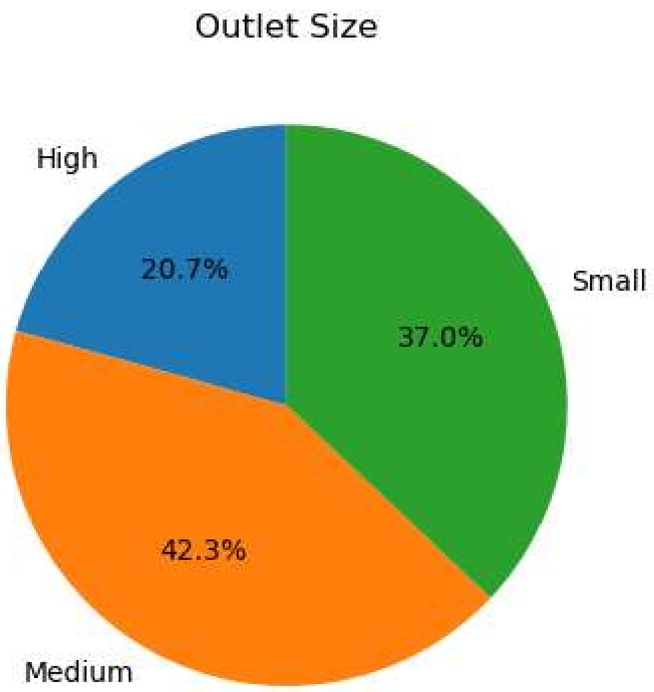


Sales by Outlet Size

In [57]:

```
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

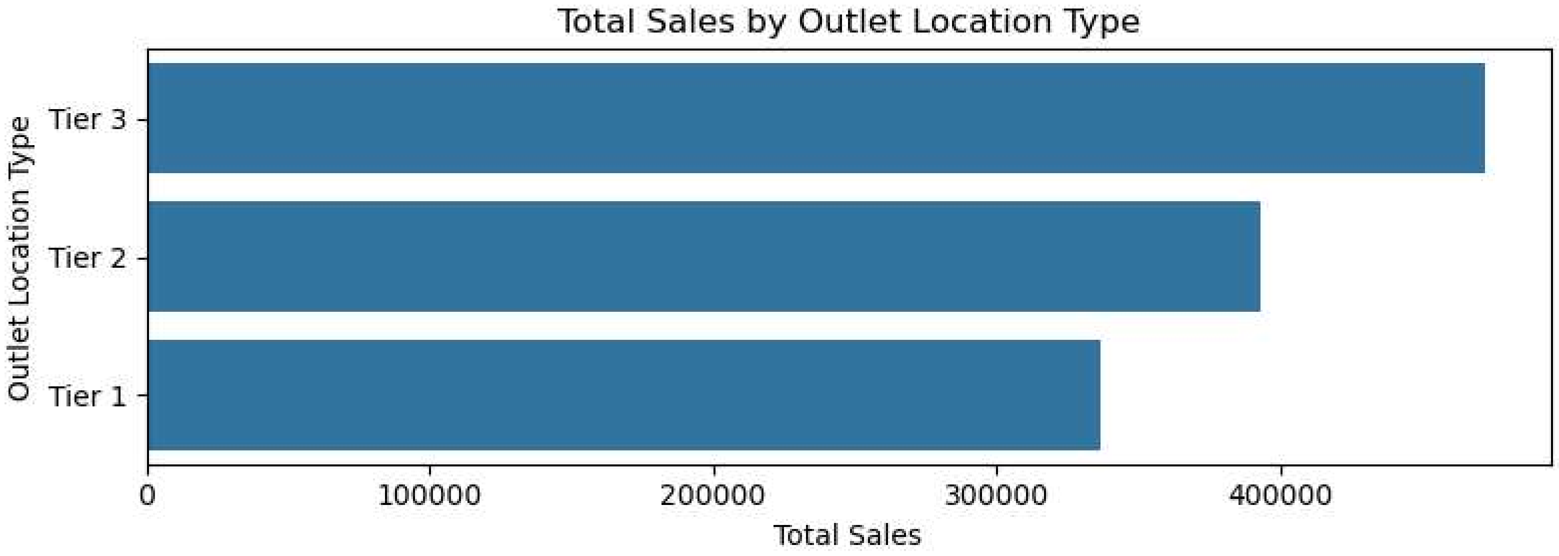
In [68]:

```
sales_by_location = df.groupby('Outlet Location Type').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.title('Total Sales by Outlet Location Type')
plt.xlabel('Total Sales')
plt.ylabel('Outlet Location Type')

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


Total Sales by Outlet Location Type
Outlet Location Type
Tier 3
Tier 2
Tier 1
0
100000
200000
300000
400000
Total Sales

```
In [ ]:
```