

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

```
In [2]: df=pd.read_csv("blinkit_data.csv")
```

```
In [3]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8523 entries, 0 to 8522
Data columns (total 12 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   Item Fat Content                      8523 non-null  object
1   Item Identifier                      8523 non-null  object
2   Item Type                           8523 non-null  object
3   Outlet Establishment Year            8523 non-null  int64
4   Outlet Identifier                   8523 non-null  object
5   Outlet Location Type                8523 non-null  object
6   Outlet Size                         8523 non-null  object
7   Outlet Type                         8523 non-null  object
8   Item Visibility                     8523 non-null  float64
9   Item Weight                         7060 non-null  float64
10  Sales                              8523 non-null  float64
11  Rating                             8523 non-null  float64
dtypes: float64(4), int64(1), object(7)
memory usage: 799.2+ KB
```

```
In [21]: df.head()
```

Out[21]:

	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

In [15]: `df.isnull().sum()`

Out[15]:

Item Fat Content	0
Item Identifier	0
Item Type	0
Outlet Establishment Year	0
Outlet Identifier	0
Outlet Location Type	0
Outlet Size	0
Outlet Type	0
Item Visibility	0
Item Weight	1463
Sales	0
Rating	0

dtype: int64

In [ ]:

In [29]: `df.shape`

Out[29]: (8523, 12)

```
In [31]: df.describe()
```

Out[31]:

	Outlet Establishment Year	Item Visibility	Item Weight	Sales	Rating
<b>count</b>	8523.000000	8523.000000	7060.000000	8523.000000	8523.000000
<b>mean</b>	2010.831867	0.066132	12.857645	140.992782	3.965857
<b>std</b>	8.371760	0.051598	4.643456	62.275067	0.605651
<b>min</b>	1998.000000	0.000000	4.555000	31.290000	1.000000
<b>25%</b>	2000.000000	0.026989	8.773750	93.826500	4.000000
<b>50%</b>	2012.000000	0.053931	12.600000	143.012800	4.000000
<b>75%</b>	2017.000000	0.094585	16.850000	185.643700	4.200000
<b>max</b>	2022.000000	0.328391	21.350000	266.888400	5.000000

```
In [33]: df.dtypes
```

Out[33]:

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

```
In [11]: print(df['Item Fat Content'].unique())
```

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

```
In [13]: df['Item Fat Content']= df['Item Fat Content'].replace({'LF' : 'Low Fat', 'low fat' : 'Low Fat', 'reg' : 'Regular'})
```

```
In [15]: print(df['Item Fat Content'].unique())
```

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

```
In [85]: # Total Sales
Total_Sales = df['Sales'].sum()
# Avg Sales
Avg_Sales= df['Sales'].mean()
# Num of items Sold
Items_Sold=df['Sales'].count()
# Avg Rating
Average_Rating=df['Rating'].mean()
print(f"Total Sales : ${ Total_Sales:,.0f}")
print(f"Average Sales : ${ Avg_Sales:,.1f}")
print(f"Number_of_Items_Sold : { Items_Sold:,.0f}")
print(f"Average_Rating : { Average_Rating:.1f}")
```

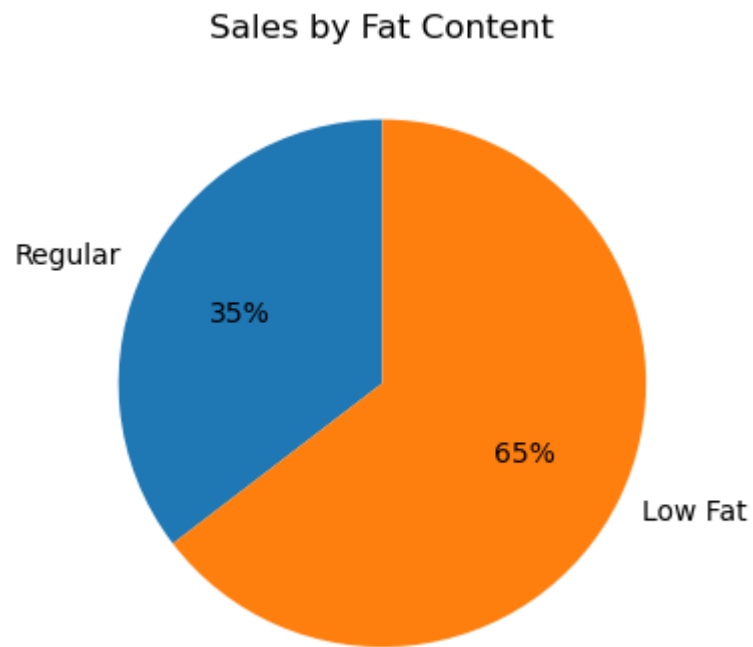
```
Total Sales : $1,201,681
```

```
Average Sales : $141.0
```

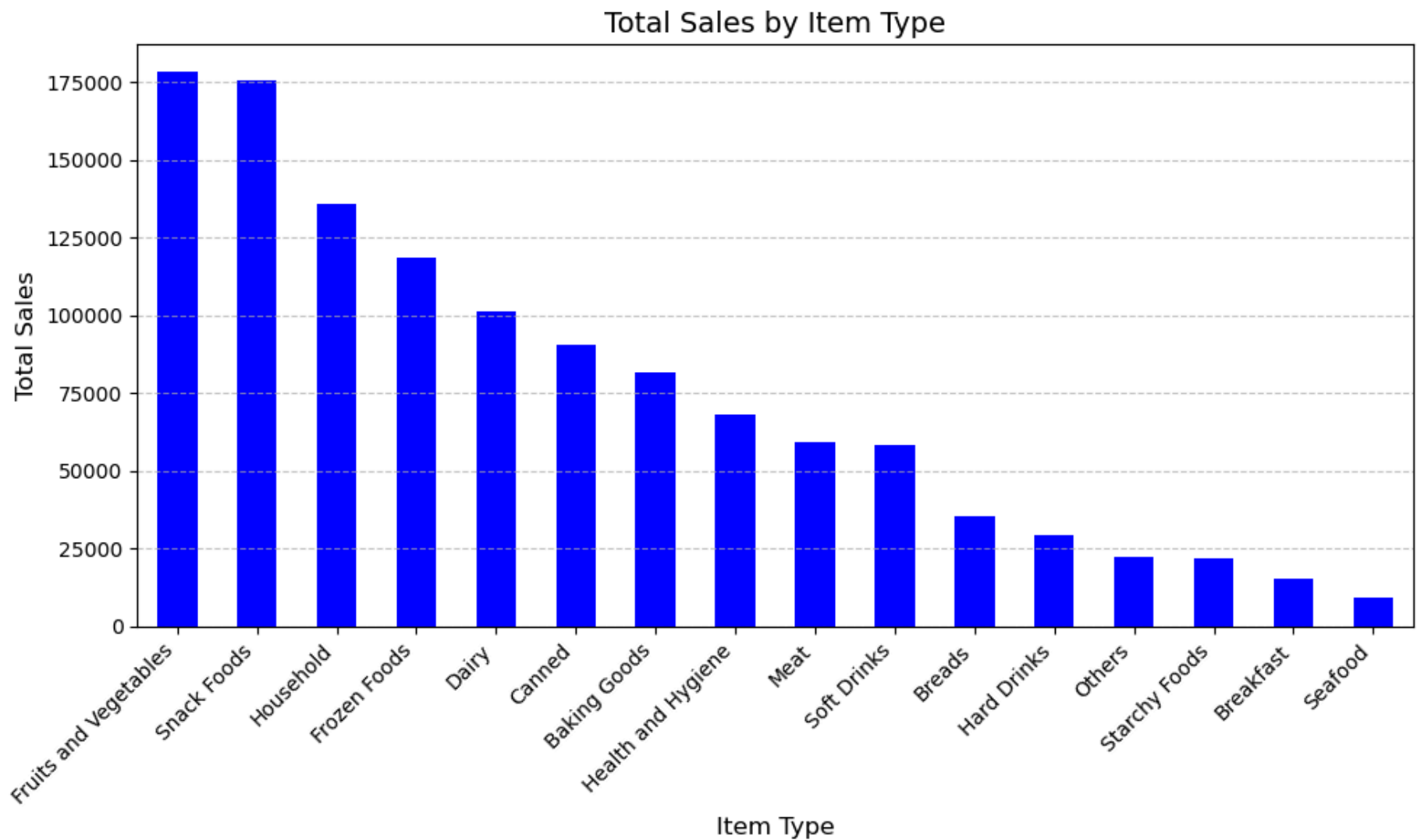
```
Number_of_Items_Sold : 8,523
```

```
Average_Rating : 4.0
```

```
In [111... sales_by_fat=df.groupby('Item Fat Content')['Sales'].sum()
sales_by_fat.sort_values().plot(kind='pie',autopct='%%.0f%%',      # Show percentage
    startangle=90,          # Rotate start angle
    figsize=(4, 5))        # Size of the plot ) # horizontal bar chart
plt.title("Sales by Fat Content")
plt.ylabel('') # Removes default y-label
plt.tight_layout()
plt.show()
```

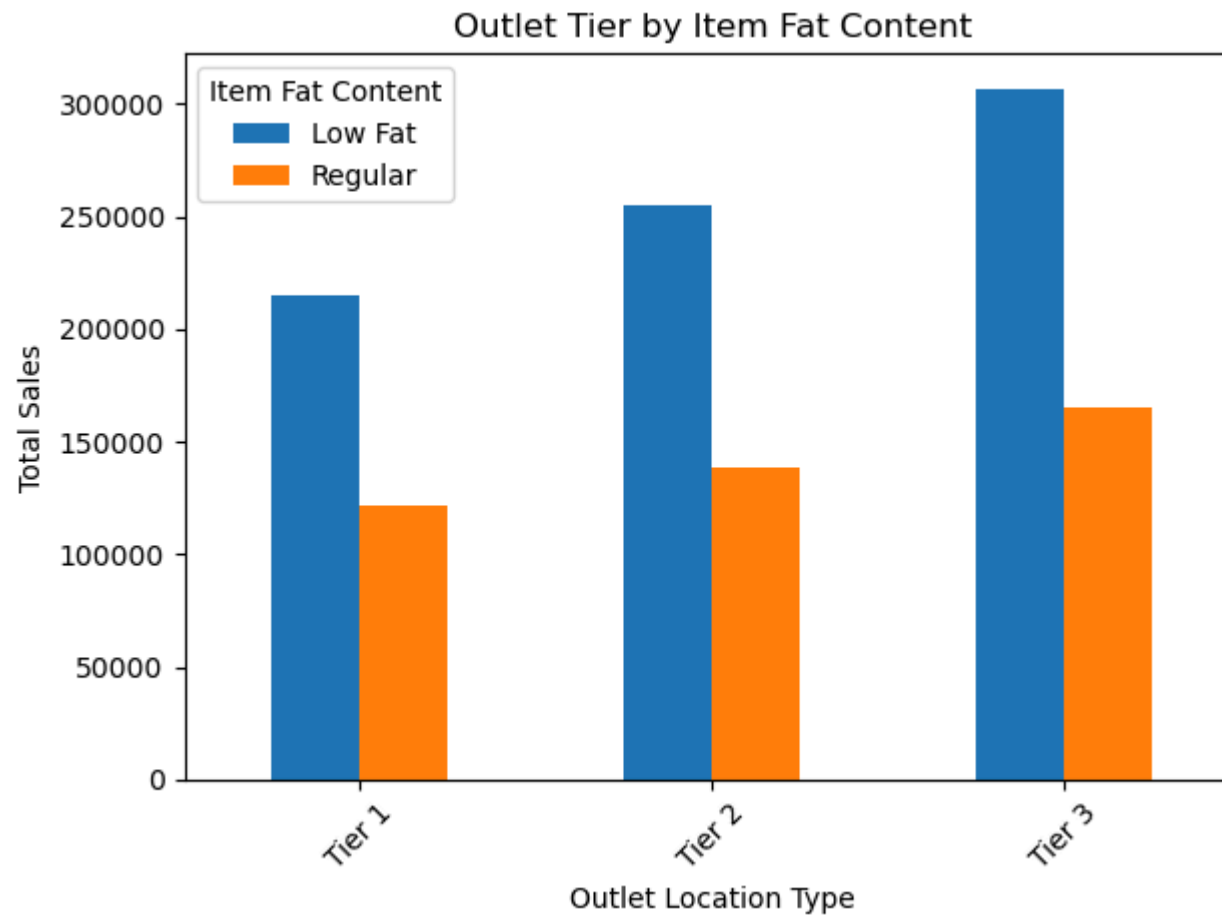


```
In [121... sales_by_type = df.groupby('Item Type')['Sales'].sum().sort_values(ascending=False)
ax=sales_by_type.plot(kind='bar', color='blue', figsize=(10, 6))
plt.title("Total Sales by Item Type", fontsize=14)
plt.xlabel("Item Type", fontsize=12)
plt.ylabel("Total Sales", fontsize=12)
plt.xticks(rotation=45, ha='right')
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.tight_layout()
plt.show()
```

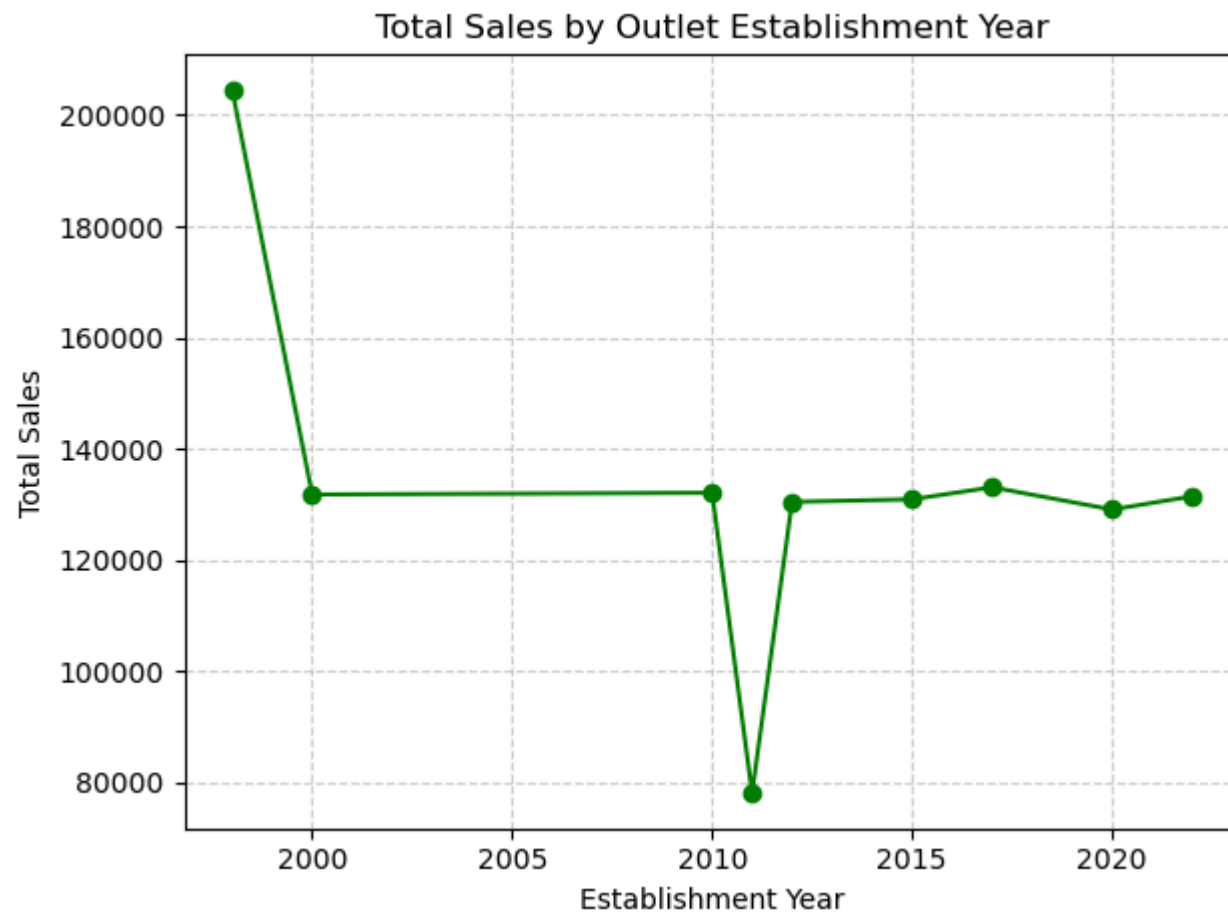


```
In [33]: sales_by_outlet_fat = df.groupby(['Outlet Location Type', 'Item Fat Content'])['Sales'].sum()
sales_pivot = sales_by_outlet_fat.unstack().fillna(0)
sales_pivot.plot(kind='bar')
plt.title("Outlet Tier by Item Fat Content")
plt.xlabel("Outlet Location Type")
plt.ylabel("Total Sales")
plt.xticks(rotation=45)
```

```
plt.legend(title="Item Fat Content")
plt.tight_layout()
plt.show()
```



```
In [31]: sales_by_year = df.groupby('Outlet Establishment Year')['Sales'].sum().sort_index()
plt.plot(sales_by_year.index, sales_by_year.values, marker='o', color='green')
plt.title("Total Sales by Outlet Establishment Year")
plt.xlabel("Establishment Year")
plt.ylabel("Total Sales")
plt.grid(True, linestyle='--', alpha=0.6)
plt.tight_layout()
plt.show()
```



In [ ]:

```
In [63]: sales_by_size = df.groupby('Outlet Size')['Sales'].sum()

# Step 2: Plot donut chart
plt.figure(figsize=(4,4))
wedges, texts, autotexts = plt.pie(
    sales_by_size,
    labels=sales_by_size.index,
    autopct='%1.1f%%',
    startangle=90,
```

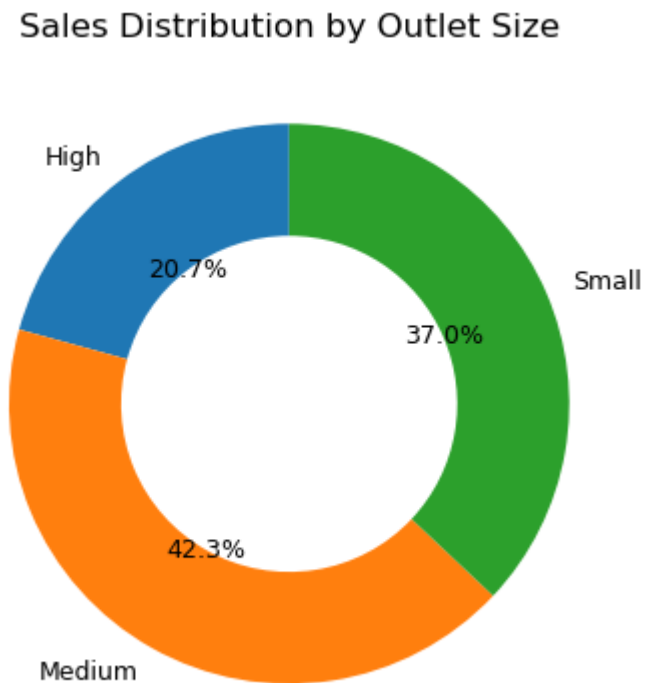


```

    wedgeprops={'width': 0.4}, # This creates the "donut" hole
    textprops={'fontsize': 9}
)

# Step 3: Add title
plt.title("Sales Distribution by Outlet Size")
plt.tight_layout()
plt.show()

```



```

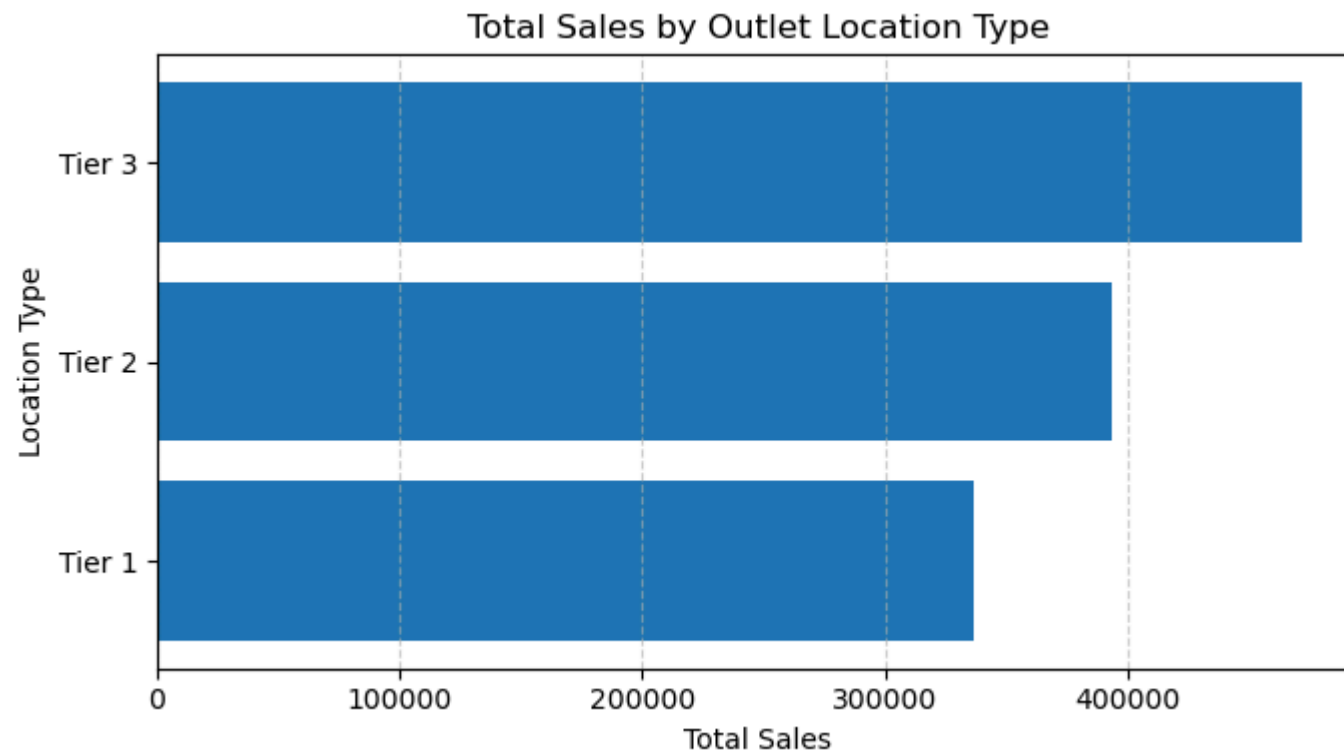
In [11]: sales_by_location = df.groupby('Outlet Location Type')['Sales'].sum().sort_values()

# Step 2: Plot horizontal bar chart
plt.figure(figsize=(7, 4))
plt.barh(sales_by_location.index, sales_by_location.values)

# Step 3: Customize chart
plt.title("Total Sales by Outlet Location Type")

```

```
plt.xlabel("Total Sales")
plt.ylabel("Location Type")
plt.grid(axis='x', linestyle='--', alpha=0.6)
plt.tight_layout()
plt.show()
```



In [ ]: