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 [2]:

```
df = pd.read_csv("C:/Users/KHUSI SINGH/Downloads/blinkit Python project/blinkit_data.csv
```

Sample Data

In [3]:

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

Out[3]:

	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	2010	OUT046	Tier 1	Small	Supermarket Type1	0.025896
3	Regular	FDL50	Canned	2000	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	2000	OUT013	Tier 3	High	Supermarket Type1	0.024129
9	Low Fat	FDX25	Canned	1998	OUT027	Tier 3	Medium	Supermarket Type3	0.101562
10	LF	FDX21	Snack Foods	1998	OUT027	Tier 3	Medium	Supermarket Type3	0.084555

	Item Fat Content	Item Identifier	Item Type	Outlet Establishment Year	Outlet Identifier	Outlet Location Type	Outlet Size	Outlet Type	Item Visibility
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	2000	OUT013	Tier 3	High	Supermarket Type1	0.090487
14	Low Fat	FDH19	Meat	1998	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	1998	OUT027	Tier 3	Medium	Supermarket Type3	0.182515
19	Low Fat	DRN47	Hard Drinks	2022	OUT018	Tier 3	Medium	Supermarket Type2	0.016895

Sample Data From Bottom

In [4]:

```
df.tail(20)
```

Out[4]:

	Item Fat Content	Item Identifier	Item Type	Outlet Establishment Year	Outlet Identifier	Outlet Location Type	Outlet Size	Outlet Type	Item Visibility
8503	Regular	FDR22	Snack Foods	1998	OUT027	Tier 3	Medium	Supermarket Type3	0.018473
8504	Regular	FDS09	Snack Foods	1998	OUT027	Tier 3	Medium	Supermarket Type3	0.080696
8505	Regular	FDS34	Snack Foods	1998	OUT027	Tier 3	Medium	Supermarket Type3	0.076387
8506	Regular	FDU09	Snack Foods	1998	OUT027	Tier 3	Medium	Supermarket Type3	0.066275
8507	Regular	FDU33	Snack Foods	1998	OUT027	Tier 3	Medium	Supermarket Type3	0.134057
8508	Regular	FDU57	Snack Foods	1998	OUT027	Tier 3	Medium	Supermarket Type3	0.089121
8509	Regular	FDU58	Snack Foods	1998	OUT027	Tier 3	Medium	Supermarket Type3	0.028871
8510	Regular	FDX46	Snack Foods	1998	OUT027	Tier 3	Medium	Supermarket Type3	0.057835

	Item Fat Content	Item Identifier	Item Type	Outlet Establishment Year	Outlet Identifier	Outlet Location Type	Outlet Size	Outlet Type	Item Visibility
8511	Regular	FDX57	Snack Foods	1998	OUT027	Tier 3	Medium	Supermarket Type3	0.047037
8512	Regular	FDY33	Snack Foods	1998	OUT027	Tier 3	Medium	Supermarket Type3	0.096730
8513	Regular	DRY23	Soft Drinks	1998	OUT027	Tier 3	Medium	Supermarket Type3	0.108568
8514	low fat	FDA11	Baking Goods	1998	OUT027	Tier 3	Medium	Supermarket Type3	0.043029
8515	low fat	FDK38	Canned	1998	OUT027	Tier 3	Medium	Supermarket Type3	0.053032
8516	low fat	FDO38	Canned	1998	OUT027	Tier 3	Medium	Supermarket Type3	0.072486
8517	low fat	FDG32	Fruits and Vegetables	1998	OUT027	Tier 3	Medium	Supermarket Type3	0.175143
8518	low fat	NCT53	Health and Hygiene	1998	OUT027	Tier 3	Medium	Supermarket Type3	0.000000
8519	low fat	FDN09	Snack Foods	1998	OUT027	Tier 3	Medium	Supermarket Type3	0.034706
8520	low fat	DRE13	Soft Drinks	1998	OUT027	Tier 3	Medium	Supermarket Type3	0.027571
8521	reg	FDT50	Dairy	1998	OUT027	Tier 3	Medium	Supermarket Type3	0.107715
8522	reg	FDM58	Snack Foods	1998	OUT027	Tier 3	Medium	Supermarket Type3	0.000000

Size of Data

In [5]:

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

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

Field info

In [6]:

```
df.columns
```

Out[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')
```

Data Types

In [7]:

```
df.dtypes
```

```
Out[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
```

Data Cleaning

In [8]:

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

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

In [9]:

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

In [10]:

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

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

BUSINESS REQUIREMENTS

Kpi's REQUIREMENTS

In [11]:

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

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


#No of Item Sold
no_of_items_sold = df['Sales'].count()


#Average Ratings
avg_ratings = df['Rating'].mean()


#Display
print(f"Total Sales: ${total_sales:,.1f}")
print(f"Average Sales: ${avg_sales:,.1f}")
print(f"No of Items Sold: {no_of_items_sold:,.1f}")
print(f"Average Rating: {avg_ratings:,.1f}")
```

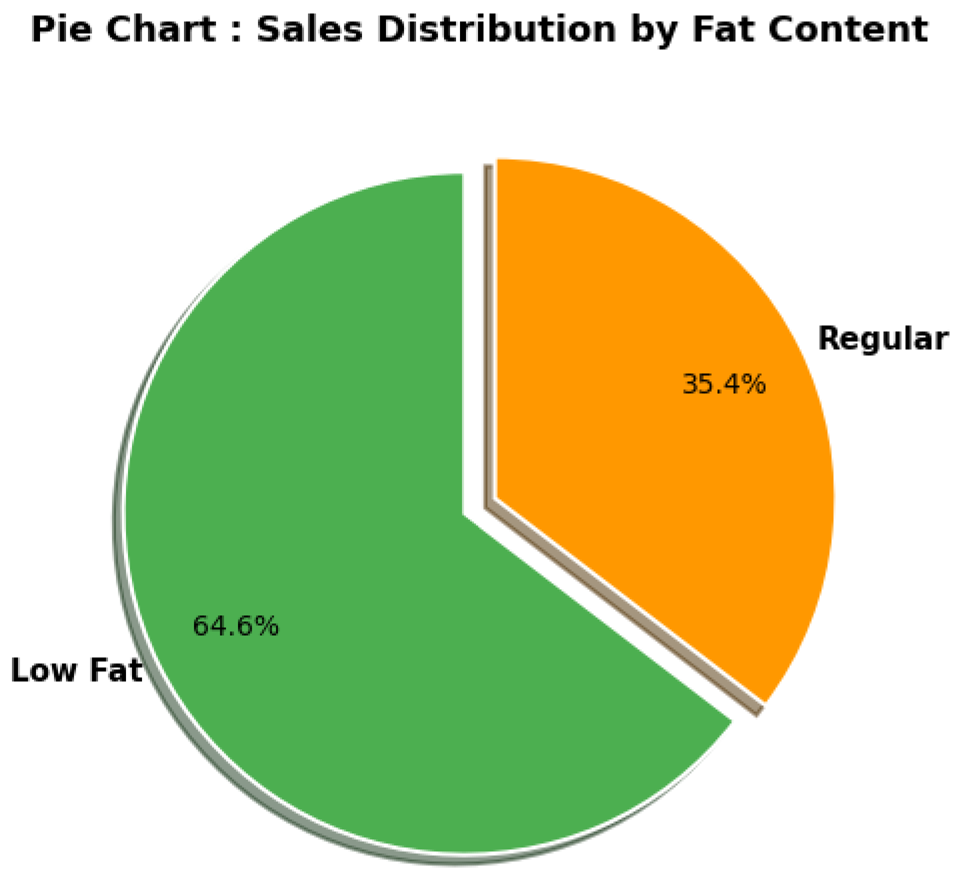
```
Total Sales: $1,201,681.5
Average Sales: $141.0
No of Items Sold: 8,523.0
Average Rating: 4.0
```

CHARTS REQUIREMENTS

Total Sales by Fat Content

In [41]:

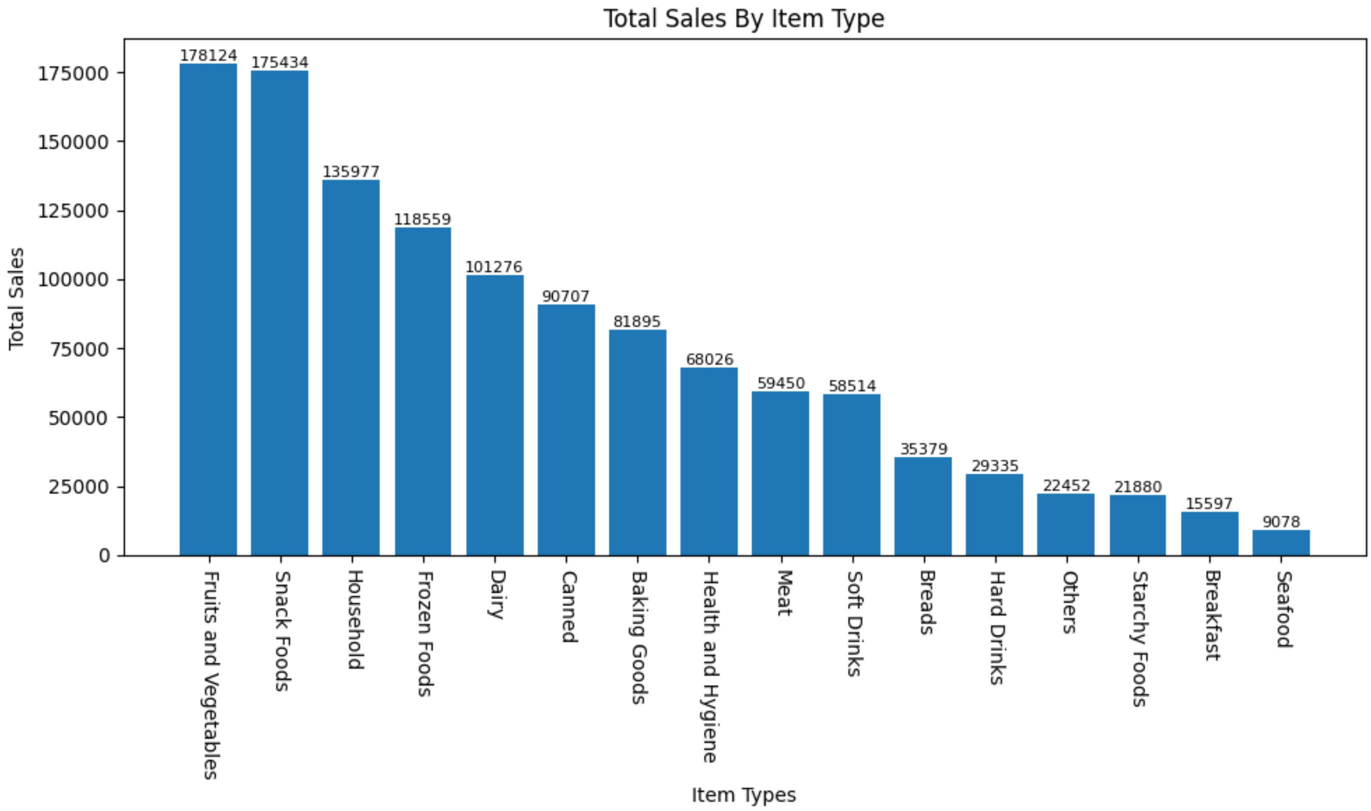
```
# Group sales by Item Fat Content
sales_by_fat = df.groupby('Item Fat Content')['Sales'].sum()

plt.figure(figsize=(5, 5))

colors = ['#4CAF50', '#FF9800']  # green = Low Fat, orange = Regular

wedges, texts, autotexts = plt.pie(
    sales_by_fat.values,
    labels=sales_by_fat.index,
    autopct='%1.1f%%',
    startangle=90,
    colors=colors,
    explode=[0.05] * len(sales_by_fat),  # slight separation for style
    shadow=True,
    pctdistance=0.75,
    labeldistance=1.05,
    wedgeprops={'edgecolor': 'white', 'linewidth': 1.5}
)

# Improve text style
for text in texts:
    text.set_fontsize(11)
    text.set_fontweight('bold')

for autotext in autotexts:
    autotext.set_fontsize(10)
    autotext.set_color('black')

plt.title('Pie Chart : Sales Distribution by Fat Content', fontsize=13, fontweight='bold
plt.axis('equal')  # perfect circle
plt.tight_layout()
plt.show()
```



Total Sales by Item Type

In [13]:

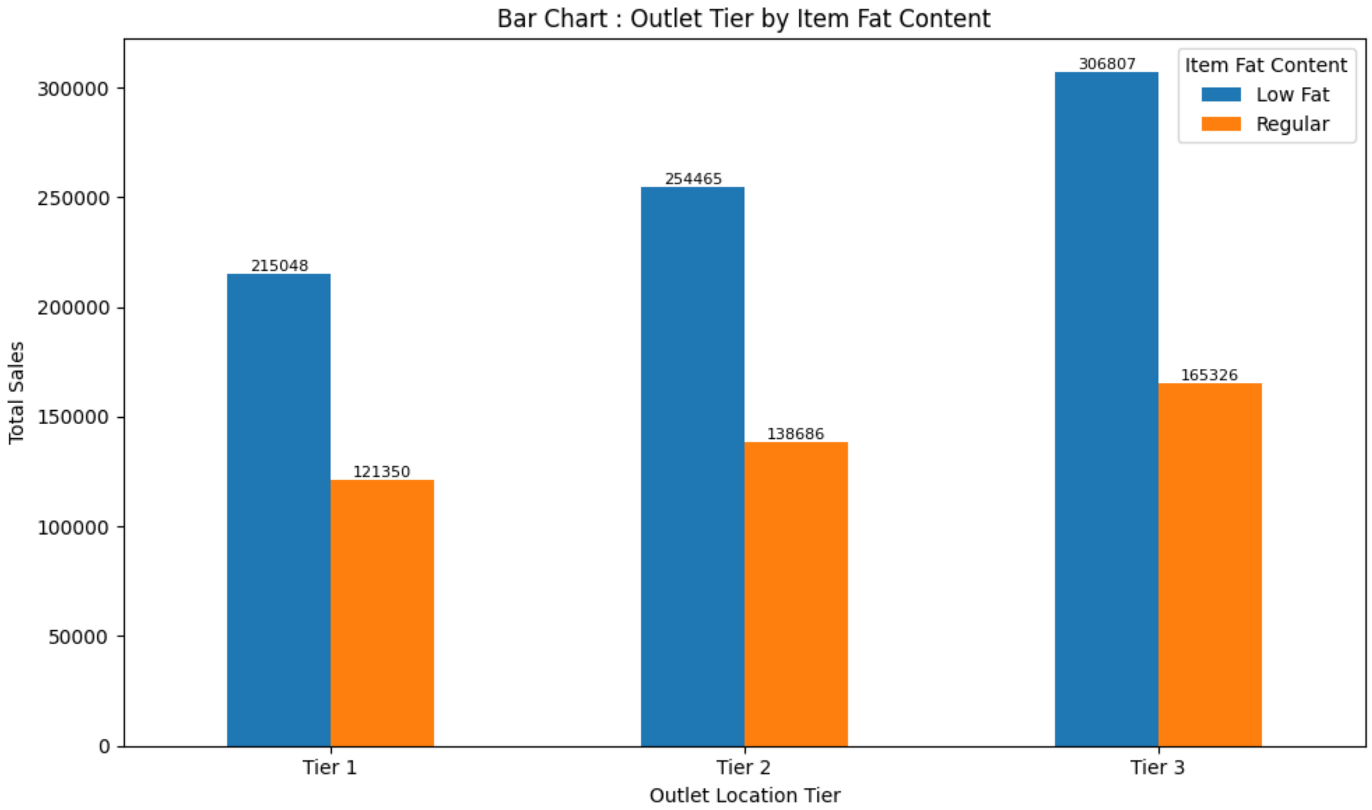
```
sales_by_type = df.groupby('Item Type')['Sales'].sum().sort_values(ascending=False)

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

plt.xticks(rotation=-90)
plt.xlabel('Item Types')
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 Contentby Outlet For Total Sales

In [40]:

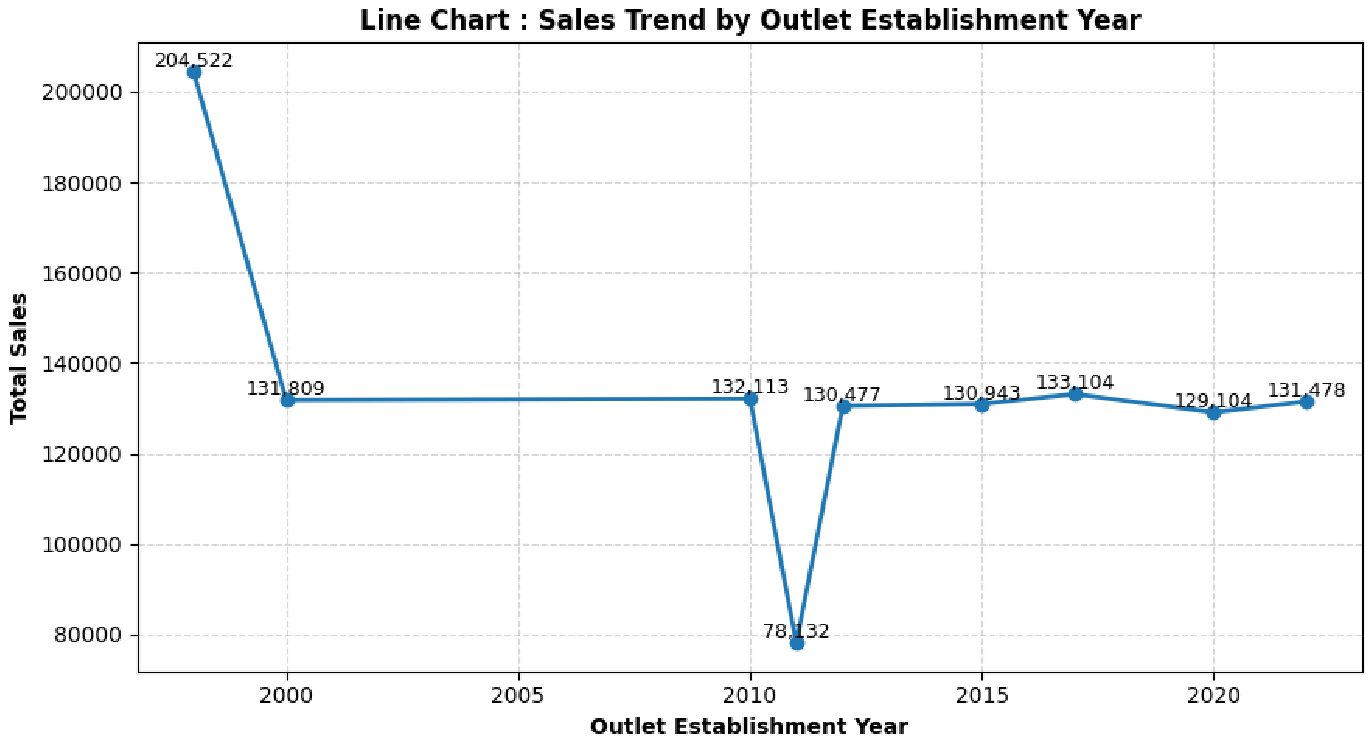
```
# Group data by Outlet Location Type and Item Fat Content
grouped = df.groupby(['Outlet Location Type', 'Item Fat Content'])['Sales'].sum().unstac

# Plot
grouped.plot(kind='bar', figsize=(10, 6))

plt.xlabel('Outlet Location Tier')
plt.ylabel('Total Sales')
plt.title('Bar Chart : Outlet Tier by Item Fat Content')
plt.xticks(rotation=0)

# Add value labels on bars
for container in plt.gca().containers:
    plt.bar_label(container, fmt='%.0f', fontsize=8)

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



Total Sales by Outlet Establishment

In [39]:

```
# Group sales by Outlet Establishment Year
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',
    linewidth=2
)

# Add grid for better readability
plt.grid(True, linestyle='--', alpha=0.5)

# Labels & title
plt.xlabel('Outlet Establishment Year', fontsize=10, fontweight='bold')
plt.ylabel('Total Sales', fontsize=10, fontweight='bold')
plt.title('Line Chart : Sales Trend by Outlet Establishment Year', fontsize=12, fontweig

# Add value labels with proper spacing
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=9
    )

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

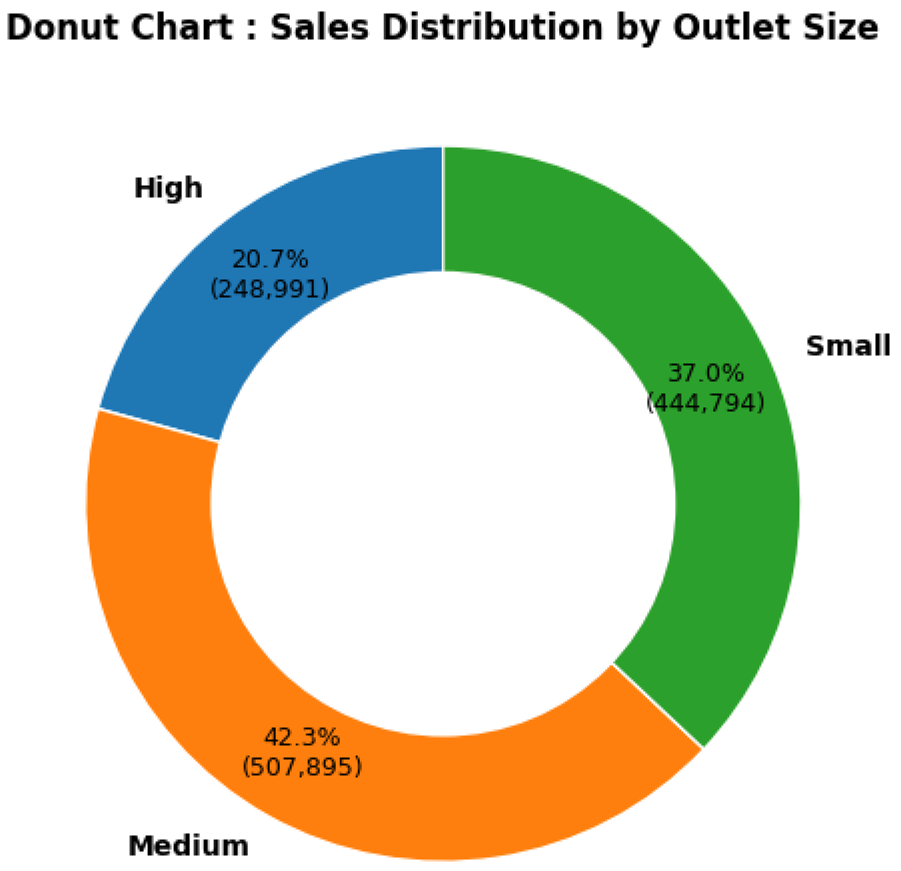


Sales by Outlet Size

In [38]:

```
# Group sales by Outlet Size
sales_by_size = df.groupby('Outlet Size')['Sales'].sum()

# Plot donut chart
plt.figure(figsize=(5, 5))

wedges, texts, autotexts = plt.pie(
    sales_by_size.values,
    labels=sales_by_size.index,
    autopct=lambda p: f'{p:.1f}%\n({int(p*sum(sales_by_size)/100):,})',
    startangle=90,
    pctdistance=0.80,
    labeldistance=1.10,
    wedgeprops={'edgecolor': 'white'}
)

# Donut hole
centre_circle = plt.Circle((0, 0), 0.65, fc='white')
plt.gca().add_artist(centre_circle)

# Improve text appearance
for text in texts:
    text.set_fontsize(10)
    text.set_fontweight('bold')
```

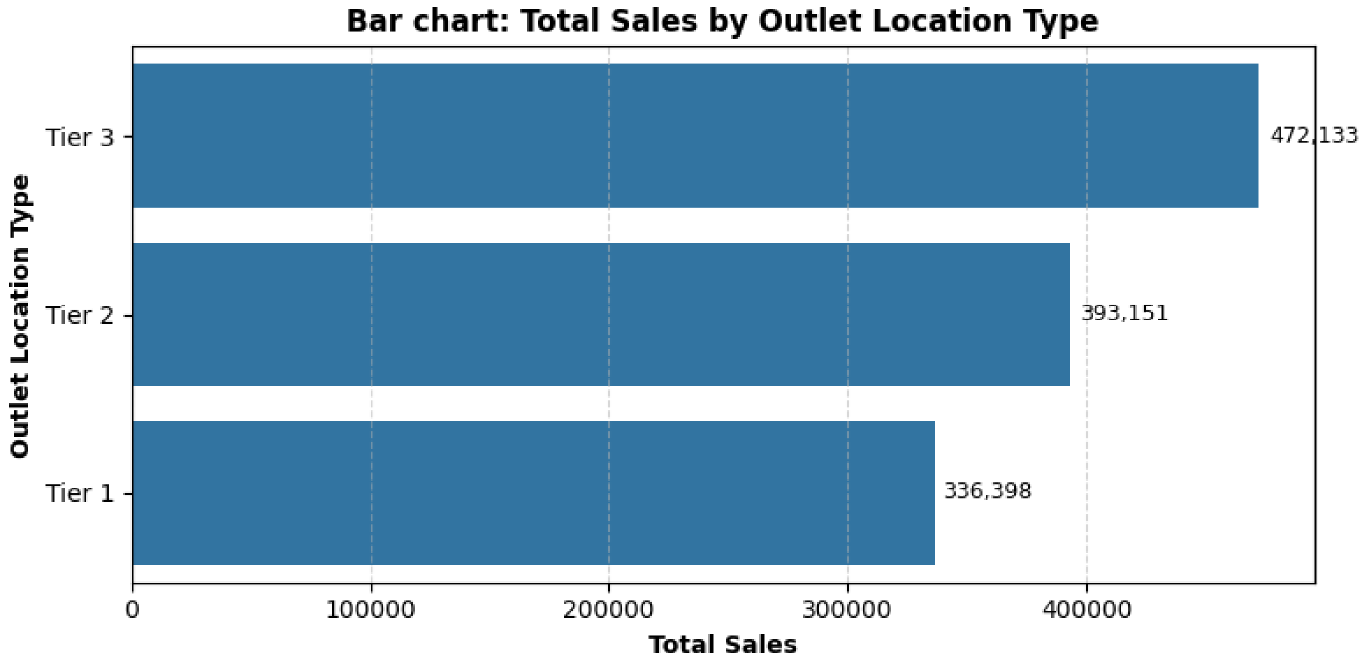
```
for autotext in autotexts:
    autotext.set_fontsize(9)
    autotext.set_color('black')

# Title
plt.title('Donut Chart : Sales Distribution by Outlet Size', fontsize=12, fontweight='bo

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



Sales by Outlet Location

In [37]:

```
import matplotlib.pyplot as plt
import seaborn as sns

# Group and sort data
sales_by_location = (
    df.groupby('Outlet Location Type')['Sales']
    .sum()
    .reset_index()
    .sort_values(by='Sales', ascending=False)
)

plt.figure(figsize=(8, 4))

ax = sns.barplot(
    x='Sales',
    y='Outlet Location Type',
    data=sales_by_location
)

# Grid for better readability
ax.grid(axis='x', linestyle='--', alpha=0.5)

# Labels & title
plt.xlabel('Total Sales', fontsize=10, fontweight='bold')
plt.ylabel('Outlet Location Type', fontsize=10, fontweight='bold')
plt.title('Bar chart: Total Sales by Outlet Location Type', fontsize=12, fontweight='bol

# Add value labels with clean spacing
for bar in ax.patches:
    ax.text(
        bar.get_width() + (bar.get_width() * 0.01),  # slight gap after bar
        bar.get_y() + bar.get_height() / 2,
        f'{bar.get_width():,.0f}',
        va='center',
        ha='left',
        fontsize=9
    )

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

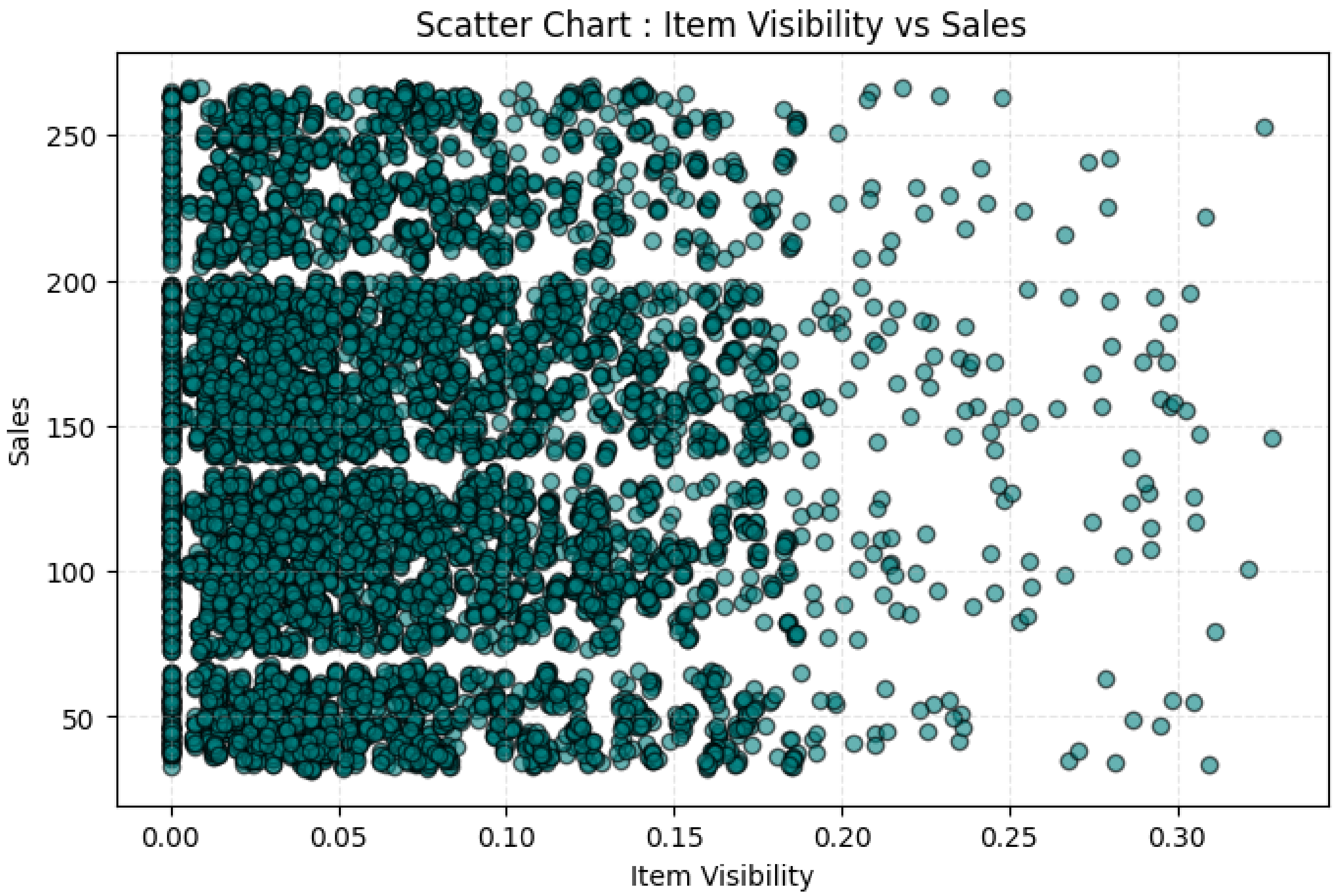


Item Visibility vs Sales

In [36]:

```
plt.figure(figsize=(8,5))
plt.scatter(
    df['Item Visibility'],
    df['Sales'],
    color='teal',
    alpha=0.6,
    edgecolors='black'
)
```

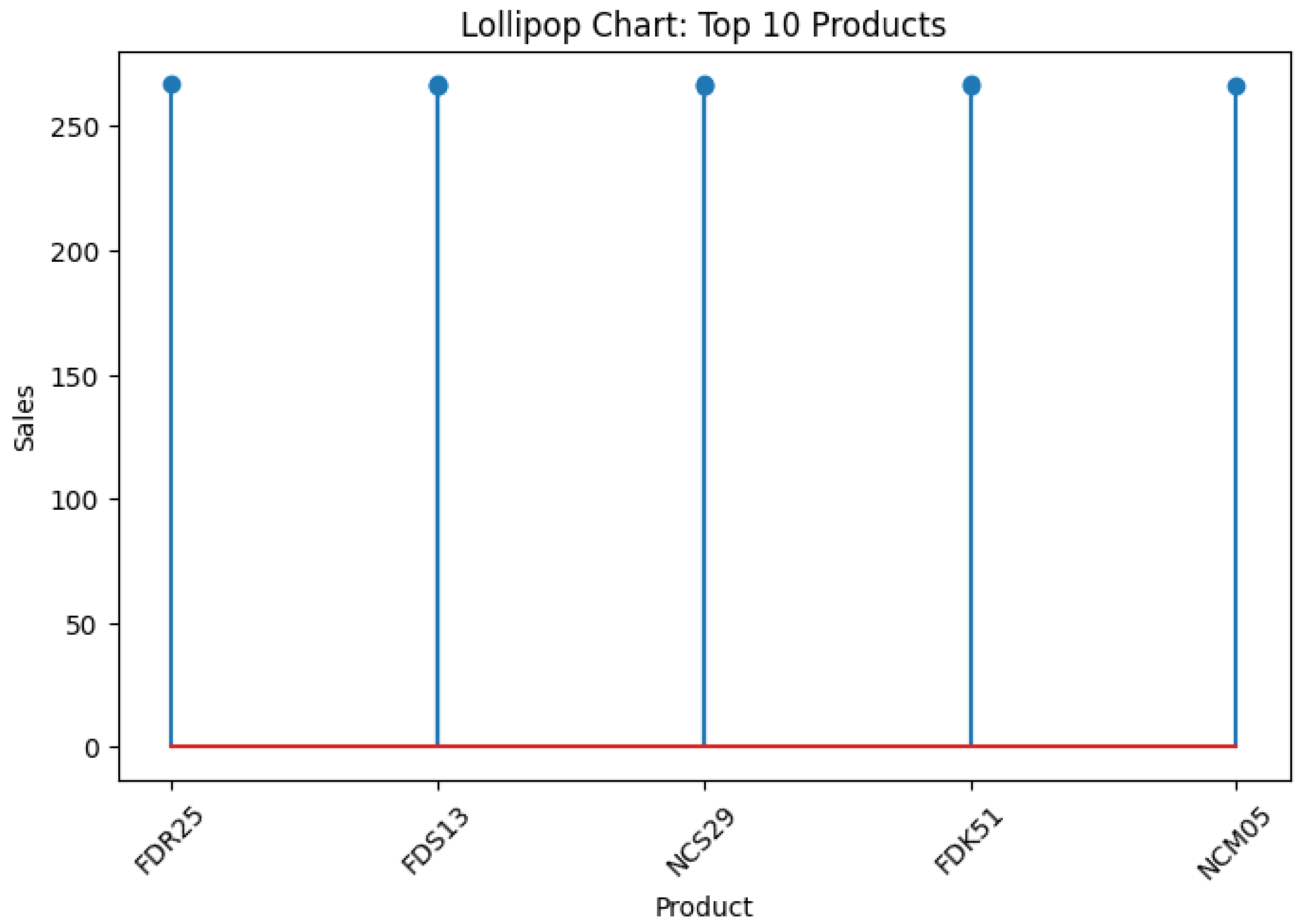
```
plt.title("Scatter Chart : Item Visibility vs Sales")
plt.xlabel("Item Visibility")
plt.ylabel("Sales")
plt.grid(True, linestyle='--', alpha=0.3)
plt.show()
```



Top 10 Highest Selling Products

In [33]:

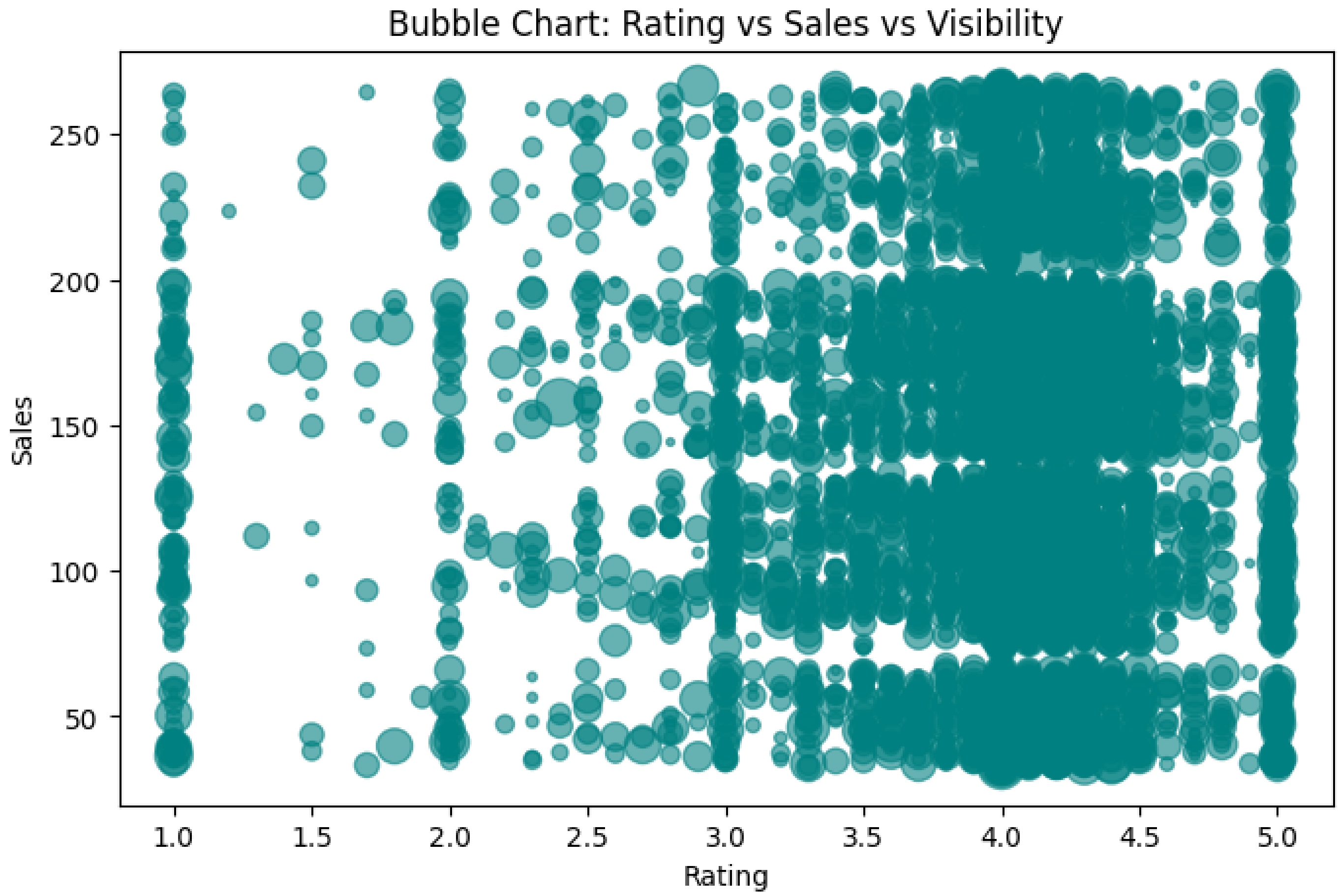
```
top10 = df.sort_values('Sales', ascending=False).head(10)

plt.figure(figsize=(8,5))
plt.stem(top10['Item Identifier'], top10['Sales'])

plt.title("Lollipop Chart: Top 10 Products")
plt.xlabel("Product")
plt.ylabel("Sales")
plt.xticks(rotation=45)
plt.show()
```



Rating vs Sales vs Visibility

In [31]:

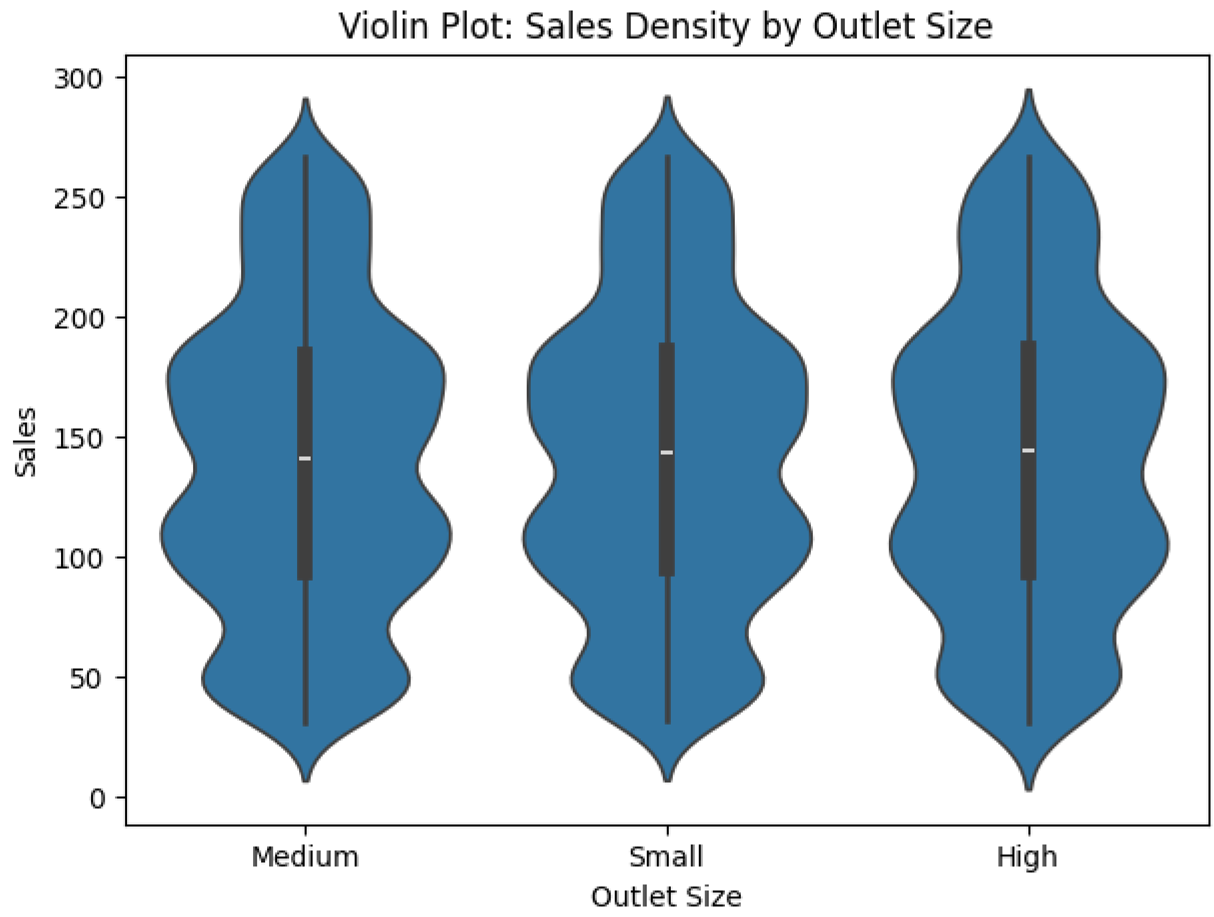
```
plt.figure(figsize=(8,5))
plt.scatter(df['Rating'], df['Sales'],
            s=df['Item Visibility']*1000,
            alpha=0.6, color='teal')

plt.title("Bubble Chart: Rating vs Sales vs Visibility")
plt.xlabel("Rating")
plt.ylabel("Sales")
plt.show()
```



Sales Density by Outlet Size

In [32]:

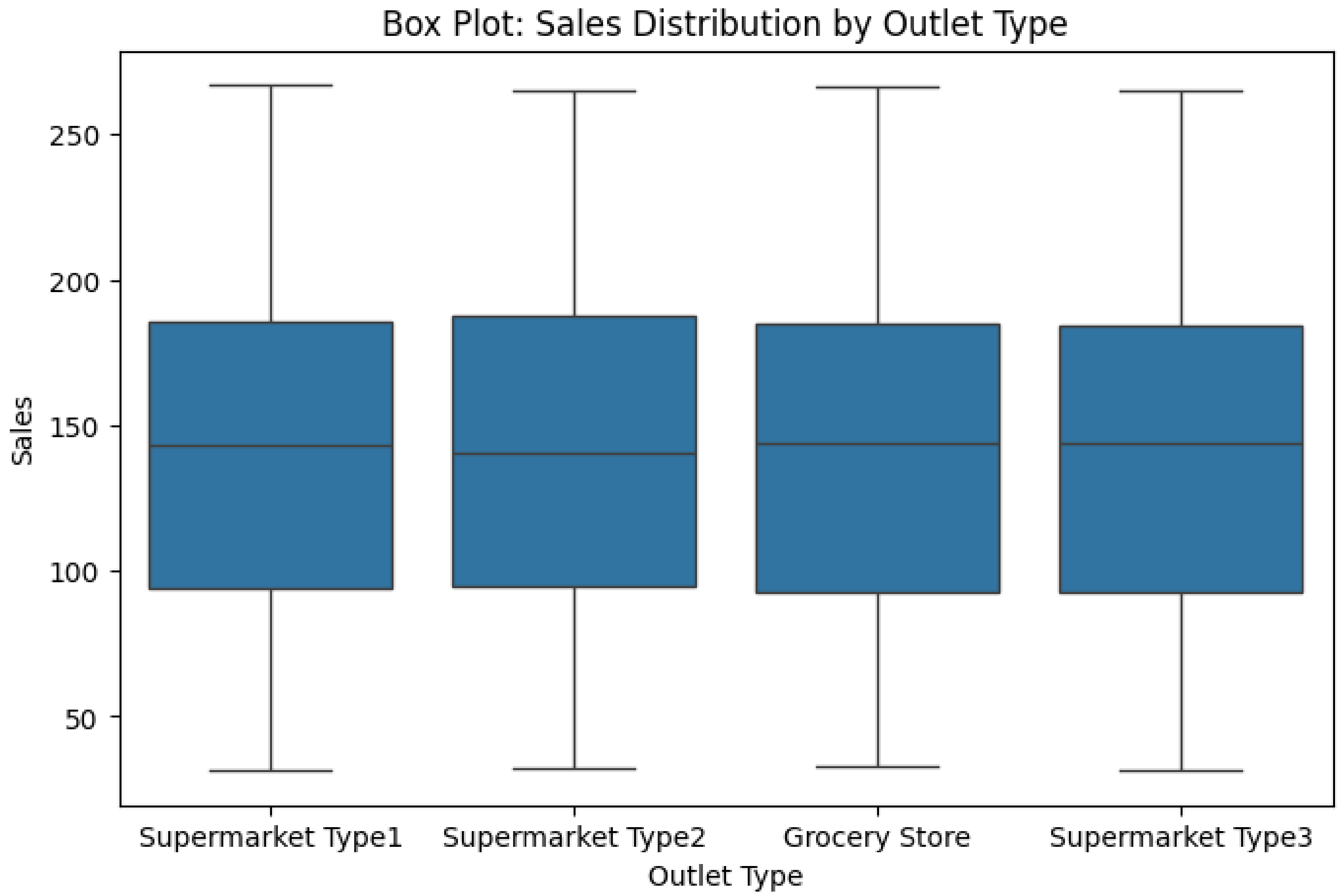
```
plt.figure(figsize=(7,5))
sns.violinplot(x='Outlet Size', y='Sales', data=df)

plt.title("Violin Plot: Sales Density by Outlet Size")
plt.xlabel("Outlet Size")
plt.ylabel("Sales")
plt.show()
```



Sales Distribution by Outlet Type

In [34]:

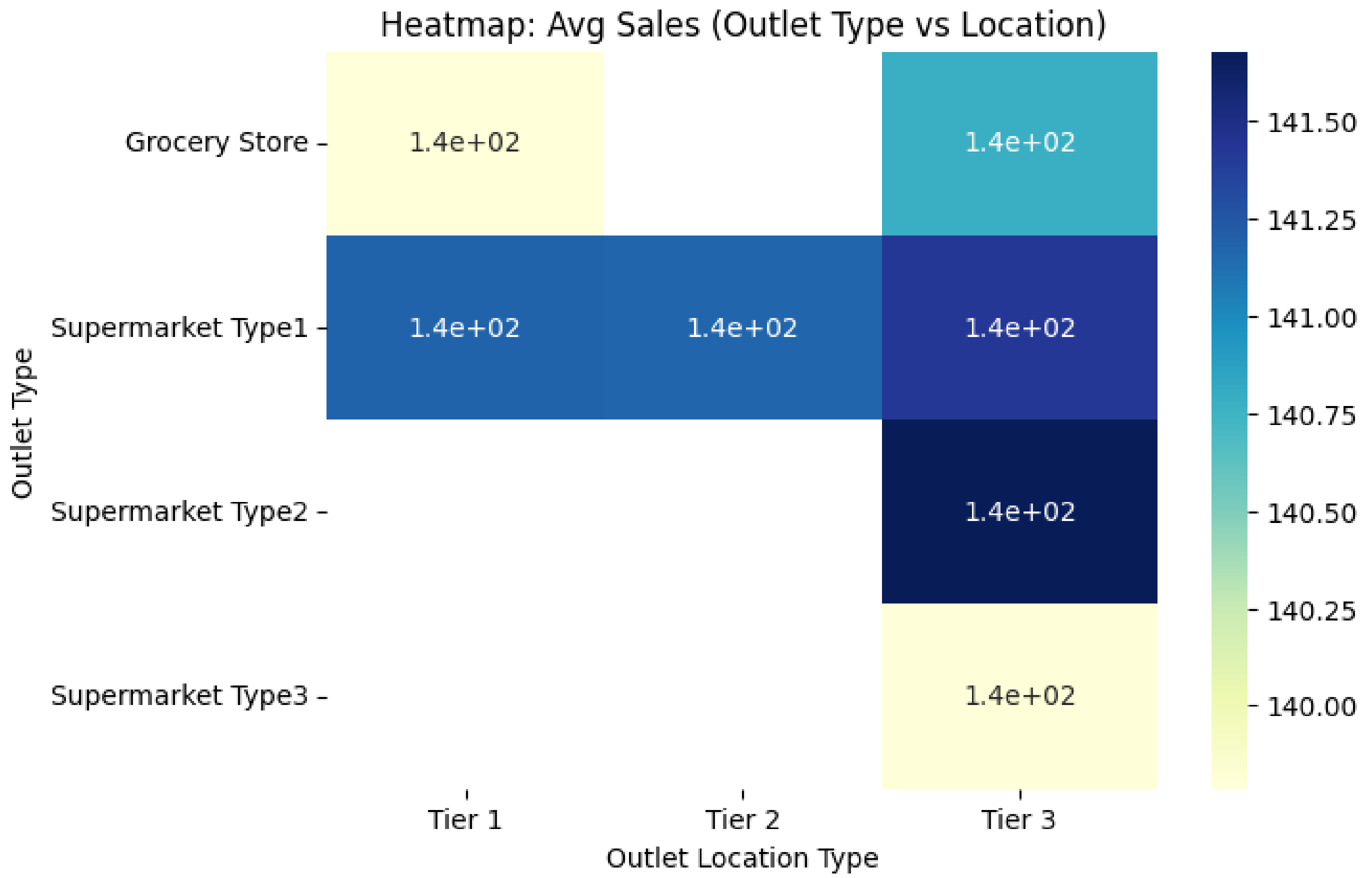
```
plt.figure(figsize=(8,5))
sns.boxplot(x='Outlet Type', y='Sales', data=df)

plt.title("Box Plot: Sales Distribution by Outlet Type")
plt.xlabel("Outlet Type")
plt.ylabel("Sales")
plt.show()
```



Outlet Type vs Location Type

In [35]:

```
heat = pd.crosstab(
    df['Outlet Type'],
    df['Outlet Location Type'],
    values=df['Sales'],
    aggfunc='mean'
)

plt.figure(figsize=(7,5))
sns.heatmap(heat, annot=True, cmap='YlGnBu')

plt.title("Heatmap: Avg Sales (Outlet Type vs Location)")
plt.show()
```


Heatmap: Avg Sales (Outlet Type vs Location)
Outlet Type
Grocery Store
Supermarket Type1
Supermarket Type2
Supermarket Type3
1.4e+02
1.4e+02
1.4e+02
1.4e+02
1.4e+02
1.4e+02
1.4e+02
Tier 1
Tier 2
Tier 3
Outlet Location Type
141.50
141.25
141.00
140.75
140.50
140.25
140.00

```
In [ ]:
```