

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

```
In [2]: data = pd.read_csv(r"C:\Users\ASUS\Downloads\blinkit_data.csv")
```

```
In [3]: data.head()
```

Out[3]:

	Item Fat Content	Item Identifier	Item Type	Outlet Establishment Year	Outlet Identifier	Outlet Location Type	Outlet Size	Outlet Type
0	Regular	FDX32	Fruits and Vegetables	2012	OUT049	Tier 1	Medium	Supermarket
1	Low Fat	NCB42	Health and Hygiene	2022	OUT018	Tier 3	Medium	Supermarket
2	Regular	FDR28	Frozen Foods	2010	OUT046	Tier 1	Small	Supermarket
3	Regular	FDL50	Canned	2000	OUT013	Tier 3	High	Supermarket
4	Low Fat	DRI25	Soft Drinks	2015	OUT045	Tier 2	Small	Supermarket

```
In [4]: data.tail()
```

Out[4]:

	Item Fat Content	Item Identifier	Item Type	Outlet Establishment Year	Outlet Identifier	Outlet Location Type	Outlet Size	Outlet Type
8518	low fat	NCT53	Health and Hygiene	1998	OUT027	Tier 3	Medium	Supermarket
8519	low fat	FDN09	Snack Foods	1998	OUT027	Tier 3	Medium	Supermarket
8520	low fat	DRE13	Soft Drinks	1998	OUT027	Tier 3	Medium	Supermarket
8521	reg	FDT50	Dairy	1998	OUT027	Tier 3	Medium	Supermarket
8522	reg	FDM58	Snack Foods	1998	OUT027	Tier 3	Medium	Supermarket

```
In [5]: data.shape
```

```
Out[5]: (8523, 12)
```

```
In [6]: data.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')
```

```
In [7]: data.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
```

```
In [8]: data['Item Fat Content'].unique()
```

```
Out[8]: array(['Regular', 'Low Fat', 'low fat', 'LF', 'reg'], dtype=object)
```

```
In [9]: data['Item Fat Content'] = data['Item Fat Content'].replace({'LF' : 'Low Fat',  
                                                                    'low fat' : 'Low Fat',  
                                                                    'reg' : 'Regular'})
```

```
In [ ]:
```

Business Requirements

```
In [12]: # TOTAL SALES  
total_sales = data['Sales'].sum()
```

```
In [13]: total_sales
```

```
Out[13]: 1201681.4808
```

```
In [14]: # AVERAGE SALES  
Avg_Sales = data['Sales'].mean()
```

```
In [15]: Avg_Sales
```

Out[15]: 140.9927819781767

```
In [16]: # No. of Items Sold
no_of_items_sold = data['Sales'].count()
```

```
In [17]: no_of_items_sold
```

Out[17]: 8523

```
In [18]: # Average Ratings
avg_ratings = data['Rating'].mean()
```

```
In [19]: avg_ratings
```

Out[19]: 3.965857092573038

```
In [41]: print(f"Total Sales: ${total_sales:,.0f}")
print(f"Avg Sales: ${Avg_Sales:,.0f}")
print(f"Number of items Sold: ${no_of_items_sold:,.0f}")
print(f"Average Rating: ${avg_ratings:,.1f}")
```

Total Sales: \$1,201,681

Avg Sales: \$141

Number of items Sold: \$8,523

Average Rating: \$4.0

```
In [ ]:
```

Charts Requirements

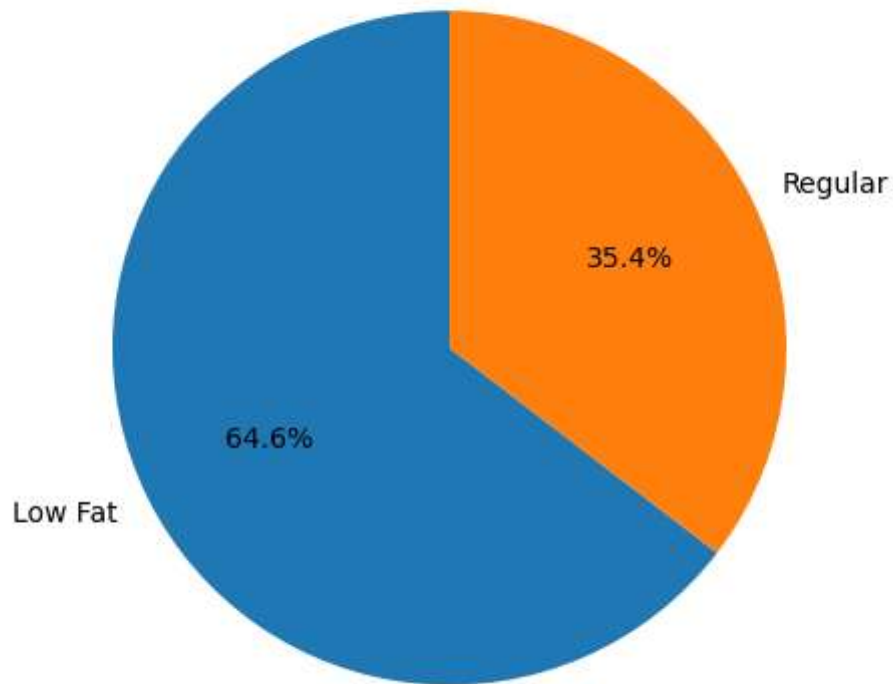
Total Sales by Fat Content

```
In [58]: sales_by_fat = data.groupby('Item Fat Content')['Sales'].sum()

plt.pie(sales_by_fat, labels = sales_by_fat.index, autopct = '%.1f%%', startangle =

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

Sales by Fat Content



Total Sales by Item Type

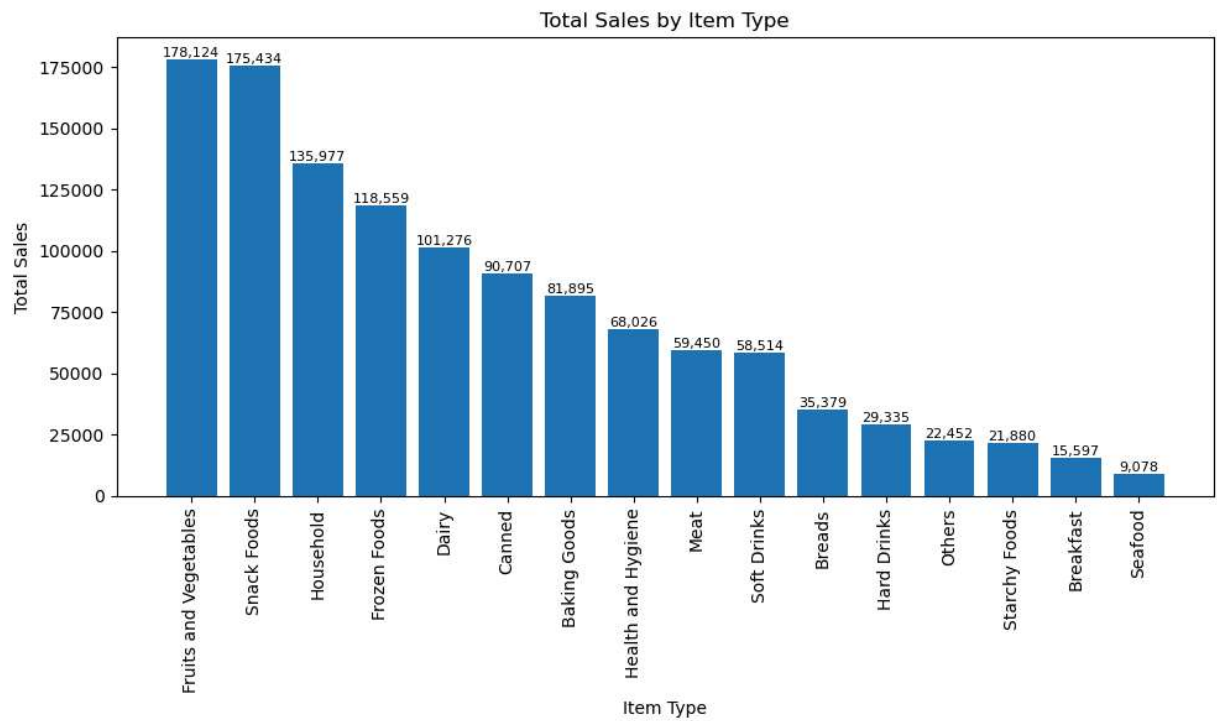
```
In [73]: sales_by_type = data.groupby('Item Type')['Sales'].sum().sort_values(ascending = Fa

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)

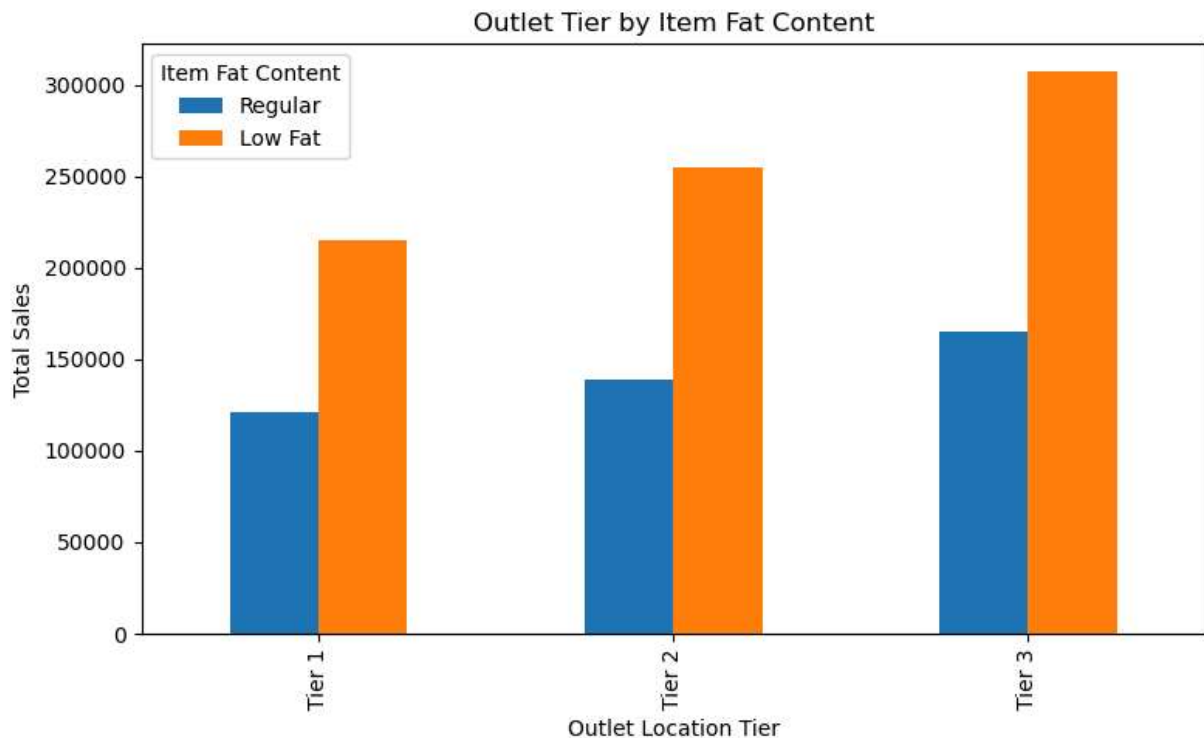
plt.tight_layout()
plt.show()
```



Fat Content by Outlet for Total Sales

```
In [80]: grouped = data.groupby(['Outlet Location Type', 'Item Fat Content'])['Sales'].sum()
grouped = grouped[['Regular', 'Low Fat']]

ax = grouped.plot(kind = 'bar', figsize = (8,5), title = 'Outlet Tier by Item Fat C
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

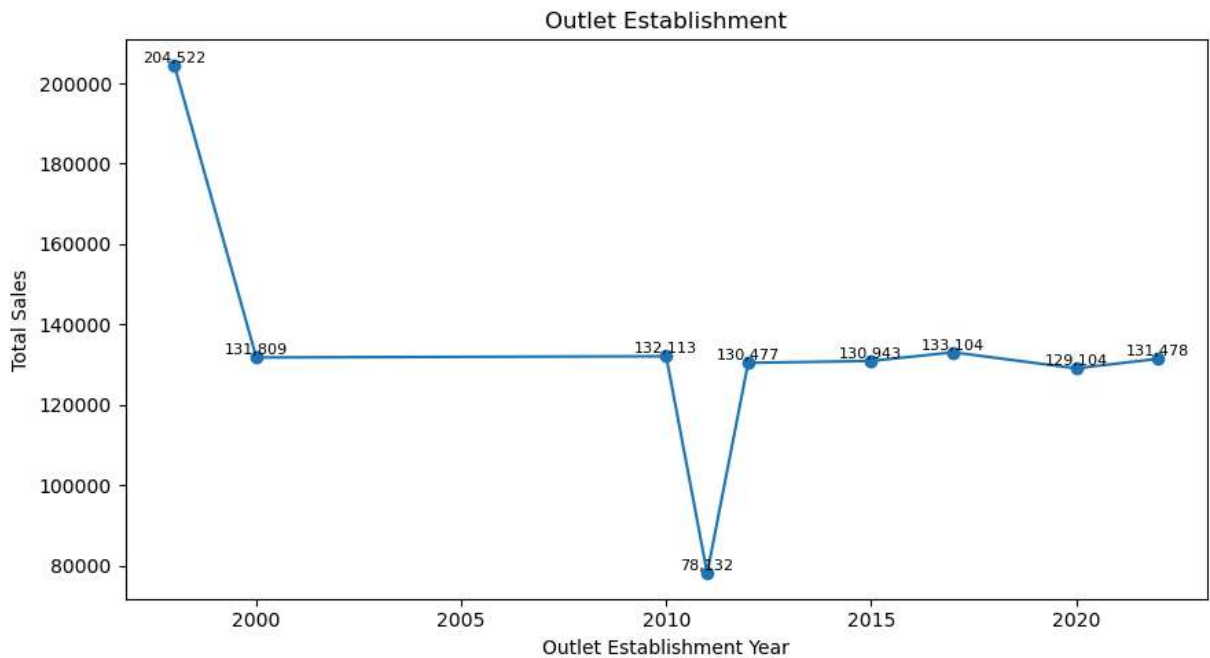
```
In [85]: sales_by_year = data.groupby('Outlet Establishment Year')['Sales'].sum().sort_index

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

```
sales_by_size = data.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()
```

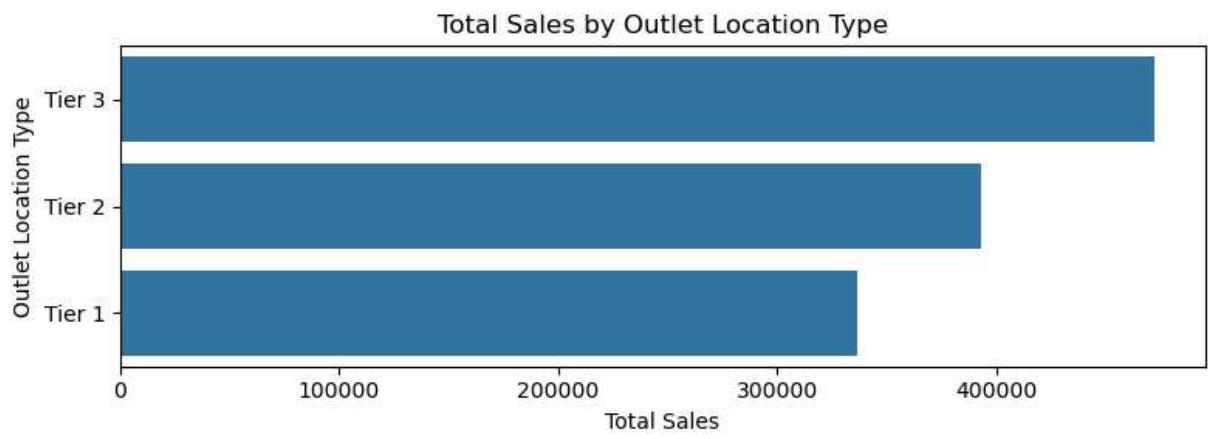
Sales by Outlet Location

```
In [93]: sales_by_location = data.groupby('Outlet Location Type')['Sales'].sum().reset_index
sales_by_location = sales_by_location.sort_values('Sales', ascending = False)

plt.figure(figsize=(8,3)) # Smaller height, enough width
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()
```



In []: