DATA ANALYSIS PYTHON PROJECT - BLINKIT ANALYSIS

BUSINESS REQUIREMENT

The aim of this project is to uncover key sales and performance metrics from Blinkit's item sales dataset. The focus lies on understanding how product fat content, item types, outlet size, and outlet location influence sales and customer behavior.

Tech Stack Used

- **Python**: Data handling and visualization
- **Pandas**: Grouping, aggregation
- **Matplotlib**: Pie, Bar, Donut, and Funnel plots

A) KPI's Requirements –

1) Total Sales: The overall revenue generated from all items sold.
2) Average Sales: The average revenue per sale.
3) Number of Items: The total count of different items sold.
4) Average Rating: The average customer rating for items sold.

KPI's REQUIREMENT

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

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

# NUMBER OF ITEMS
number_of_items = df['Sales'].count()

# AVERAGE RATING
avg_rating = df['Rating'].mean()

#display

print(f"Total Sales: ${total_sales:,.2f}")
print(f"Average Sales: ${avg_sales:,.2f}")
print(f"Number of Items: {number_of_items}")
print(f"Avg Rating: {avg_rating:,.1f}")
```

```
Total Sales: $1,201,681.48
Average Sales: $140.99
Number of Items: 8523
Avg Rating: 4.0
```

1. Total Sales by Fat Content:

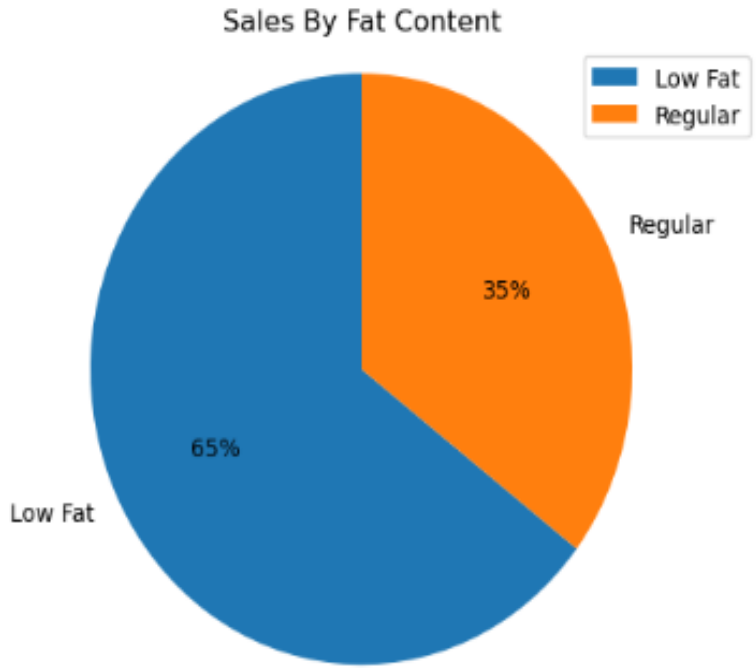
Objective: Analyze the impact of fat content on total sales.

Additional KPI Metrics: Assess how other KPIs (Average Sales, Number of Items, Average Rating) vary with fat content.

Chart Type: pie Chart.

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

plt.pie(sales_by_fat,labels = sales_by_fat.index,
                                autopct = '%.0f%%',
                                    startangle = 90)
plt.title('Sales By Fat Content')
plt.axis('equal')
plt.legend()
plt.show()
```



2. Total Sales by Item Type:

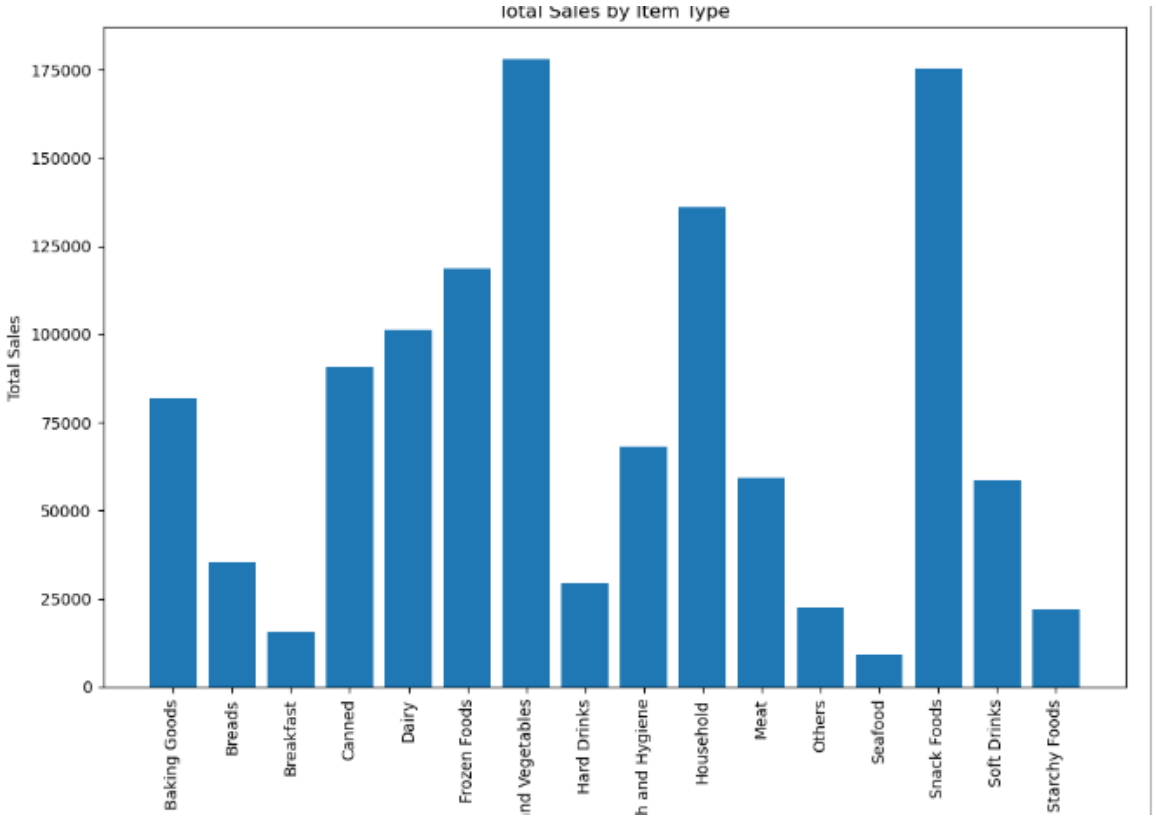
Objective: Identify the performance of different item types in terms of total sales.

Additional KPI Metrics: Assess how other KPIs (Average Sales, Number of Items, Average Rating) vary with fat content.

Chart Type: Bar Chart.

```
In [116]: sales_by_type = df.groupby('Item Type')['Sales'].sum()
          #print(sales_by_type)

          plt.figure(figsize=(10,8))
          plt.bar(sales_by_type.index,sales_by_type.values)
          plt.xlabel('Item Type')
          plt.xticks(rotation = 90)
          plt.ylabel('Total Sales')
          plt.title('Total Sales by Item Type')
          plt.tight_layout()
```



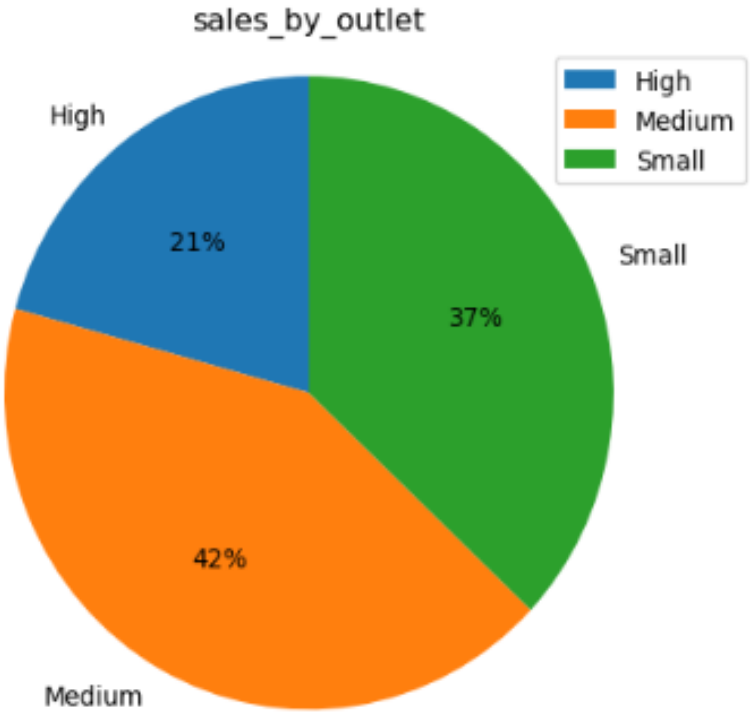
3. Sales by Outlet Size:

Objective: Analyze the correlation between outlet size and total sales.

Chart Type: Donut/ Pie Chart.

```
sales_by_outlet = df.groupby('Outlet Size')['Sales'].sum()
#print(sales_by_outlet)

plt.pie(sales_by_outlet,labels = sales_by_outlet.index,
                                autopct = '%.0f%%',
                                    startangle = 90)
plt.title('sales_by_outlet')
plt.axis('equal')
plt.legend()
plt.show()
```



4. Sales by Outlet Location:

Objective: Assess the geographic distribution of sales across different locations.

Chart Type: Funnel Map.

```
sales_by_location = df.groupby('Outlet Location Type')['Sales'].sum()
print(sales_by_location)

plt.figure(figsize=(8,6))
plt.barh(sales_by_location.index,sales_by_location.values,color = 'Green')
plt.xlabel('Total Sales')
plt.ylabel('Outlet Location')
plt.title('Funnel View of Sales by Outlet Location Type')
plt.show()
```