


```
In [52]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings("ignore")
df=pd.read_csv("blinkit.csv")
df.head()
```

Out[52]:

	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	Supermarke Type'
1	Low Fat	NCB42	Health and Hygiene	2022	OUT018	Tier 3	Medium	Supermarke Type2
2	Regular	FDR28	Frozen Foods	2010	OUT046	Tier 1	Small	Supermarke Type'
3	Regular	FDL50	Canned	2000	OUT013	Tier 3	High	Supermarke Type'
4	Low Fat	DRI25	Soft Drinks	2015	OUT045	Tier 2	Small	Supermarke Type'



```
In [54]: df.shape
```

Out[54]: (8523, 12)

```
In [56]: df.columns
```

Out[56]: 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 [58]: 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 [60]: df.describe()
```

Out[60]:

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

```
In [43]: df.dtypes
```

```
Out[43]: 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 []:

DATA CLEANING

In [62]: `df.isnull().sum()`

```
Out[62]: 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 [64]: `df.fillna(0,inplace=True)`

In [66]: `df.isnull().sum()`

```
Out[66]: 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             0
         Sales                   0
         Rating                  0
         dtype: int64
```

In [68]: `df['Item Fat Content'].unique()`

Out[68]: `array(['Regular', 'Low Fat', 'low fat', 'LF', 'reg'], dtype=object)`

In [70]: `df['Item Fat Content']=df['Item Fat Content'].replace({"LF":"Low Fat",
 "low fat":"Low Fat",
 "reg":"Regular"})`
`df`

Out[70]:

	Item Fat Content	Item Identifier	Item Type	Outlet Establishment Year	Outlet Identifier	Outlet Location Type	Outlet Size	Outlet T
0	Regular	FDX32	Fruits and Vegetables	2012	OUT049	Tier 1	Medium	Superma T
1	Low Fat	NCB42	Health and Hygiene	2022	OUT018	Tier 3	Medium	Superma T
2	Regular	FDR28	Frozen Foods	2010	OUT046	Tier 1	Small	Superma T
3	Regular	FDL50	Canned	2000	OUT013	Tier 3	High	Superma T
4	Low Fat	DRI25	Soft Drinks	2015	OUT045	Tier 2	Small	Superma T
...
8518	Low Fat	NCT53	Health and Hygiene	1998	OUT027	Tier 3	Medium	Superma T
8519	Low Fat	FDN09	Snack Foods	1998	OUT027	Tier 3	Medium	Superma T
8520	Low Fat	DRE13	Soft Drinks	1998	OUT027	Tier 3	Medium	Superma T
8521	Regular	FDT50	Dairy	1998	OUT027	Tier 3	Medium	Superma T
8522	Regular	FDM58	Snack Foods	1998	OUT027	Tier 3	Medium	Superma T

8523 rows × 12 columns



```
In [72]: df['Item Fat Content'].unique()
```

```
Out[72]: array(['Regular', 'Low Fat'], dtype=object)
```

```
In [ ]: BUSINESS REQUIREMENT
```

```
In [51]: total_sales=df['Sales'].sum()  
total_sales
```

```
Out[51]: 1201681.4808
```

```
In [53]: avg_sales=df['Sales'].mean()  
avg_sales
```

Out[53]: 140.9927819781767

```
In [55]: no_of_item_sold=df['Sales'].count()  
no_of_item_sold
```

Out[55]: 8523

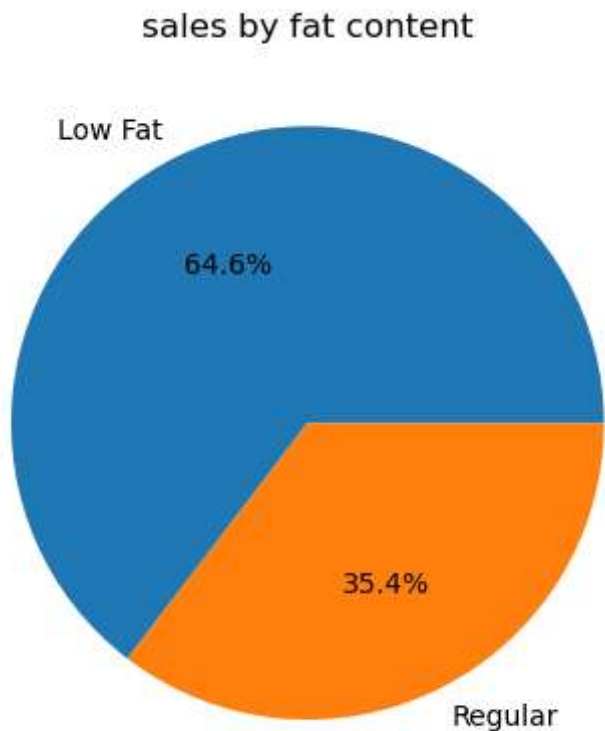
```
In [57]: rating=df['Rating'].mean()  
rating
```

Out[57]: 3.965857092573038

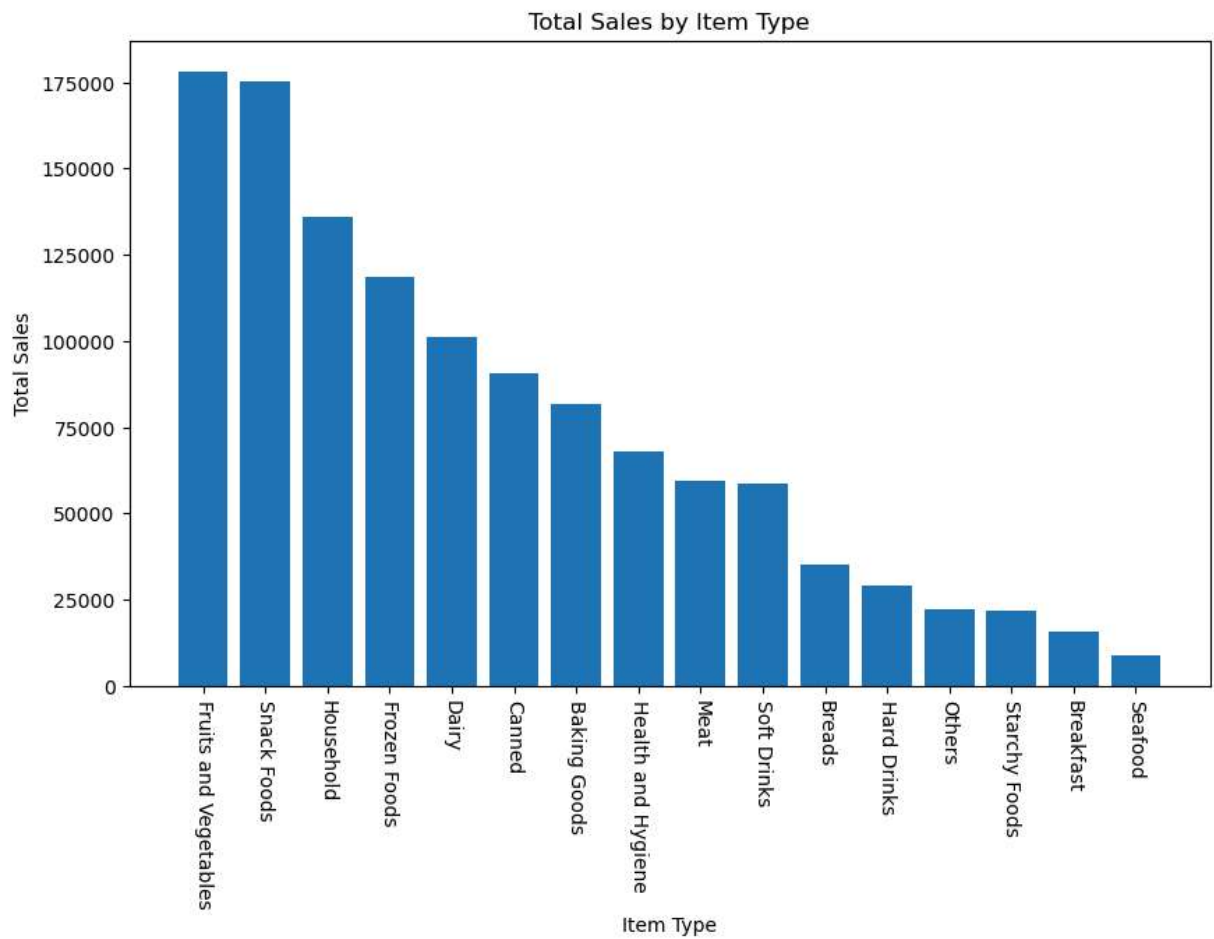
In []:

CHARTS

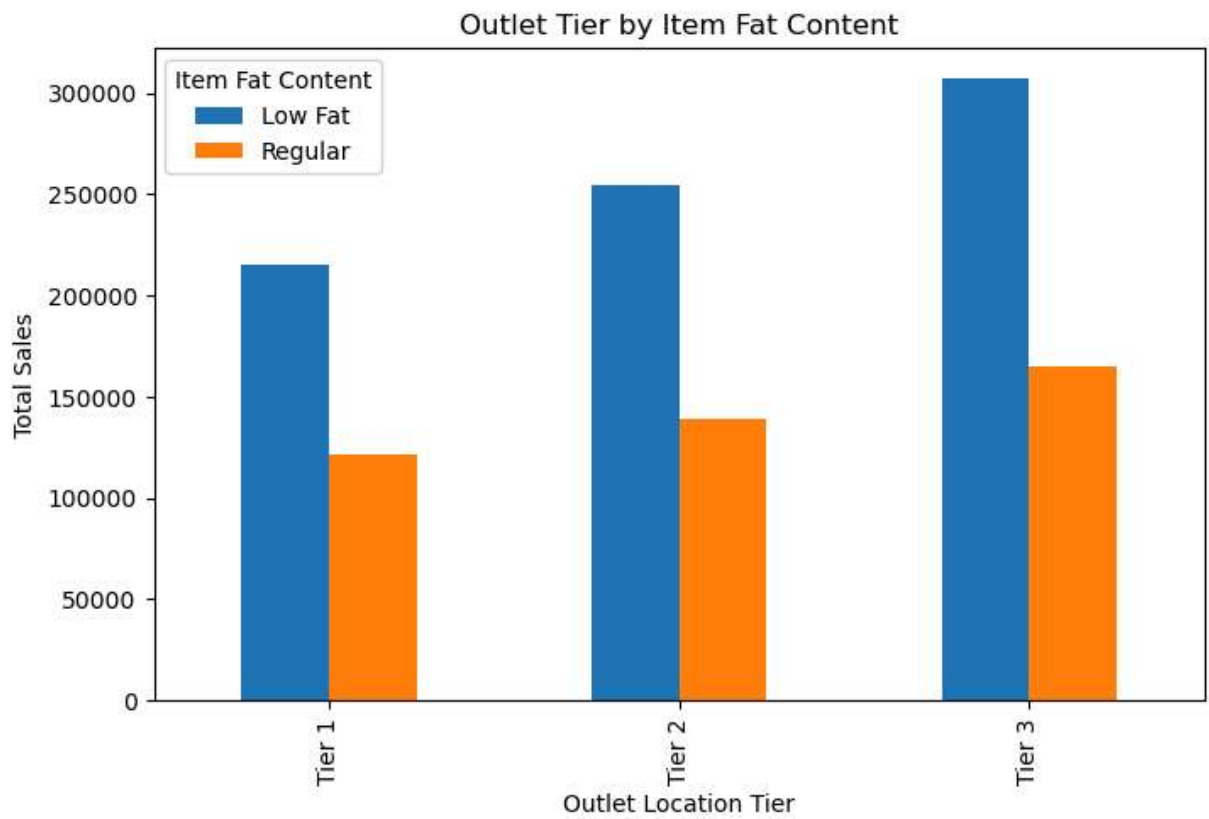
```
In [59]: sales_by_fat=df.groupby('Item Fat Content')['Sales'].sum()  
plt.pie(sales_by_fat,labels=sales_by_fat.index,  
        autopct='%.1f%%')  
plt.title('sales by fat content')  
plt.show()
```



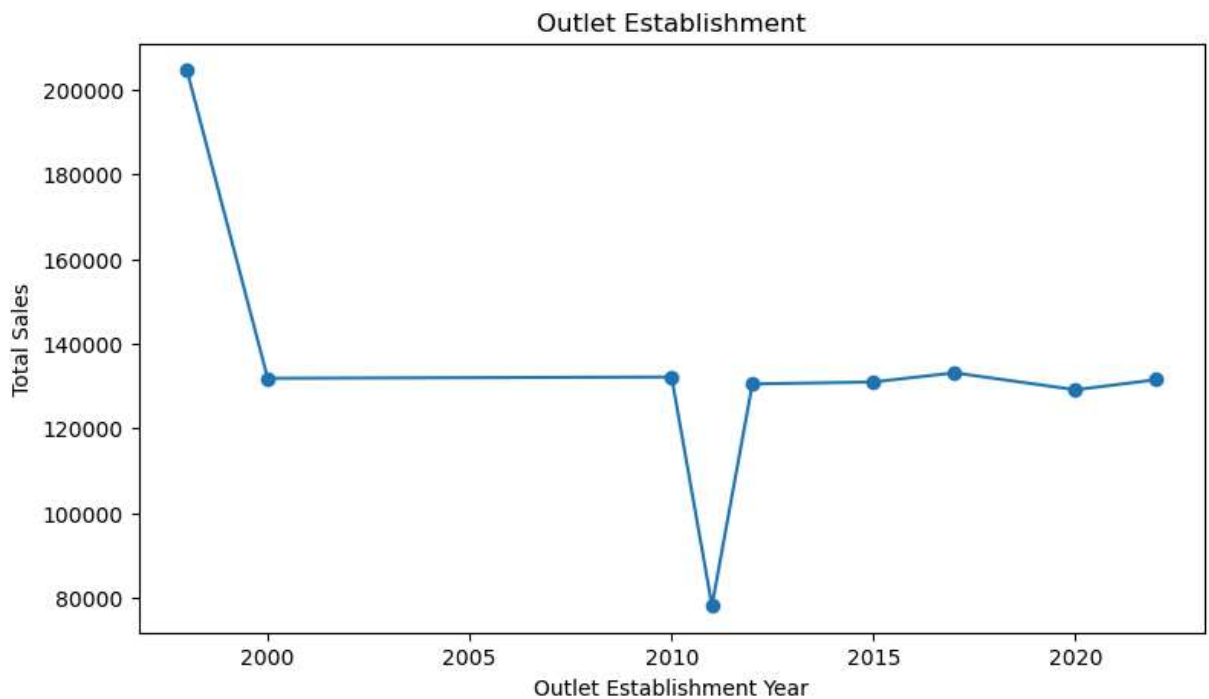
```
In [169... sales_by_type = df.groupby('Item Type')['Sales'].sum().sort_values(ascending=False)  
plt.figure(figsize=(10, 6))  
plt.bar(sales_by_type.index, sales_by_type)  
plt.xticks (rotation=-90)  
plt.xlabel('Item Type')  
plt.ylabel('Total Sales')  
plt.title('Total Sales by Item Type')  
plt.show()
```



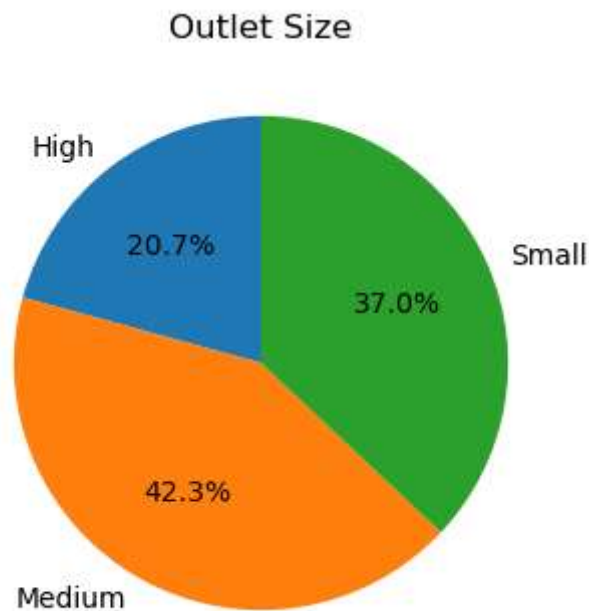
```
In [48]: grouped= df.groupby(['Outlet Location Type', 'Item Fat Content']) ['Sales'].sum().u
ax=grouped.plot(kind='bar', figsize=(8, 5), title='Outlet Tier by Item Fat Content'
plt.xlabel('Outlet Location Tier')
plt.ylabel('Total Sales')
plt.legend(title='Item Fat Content')
plt.show()
```



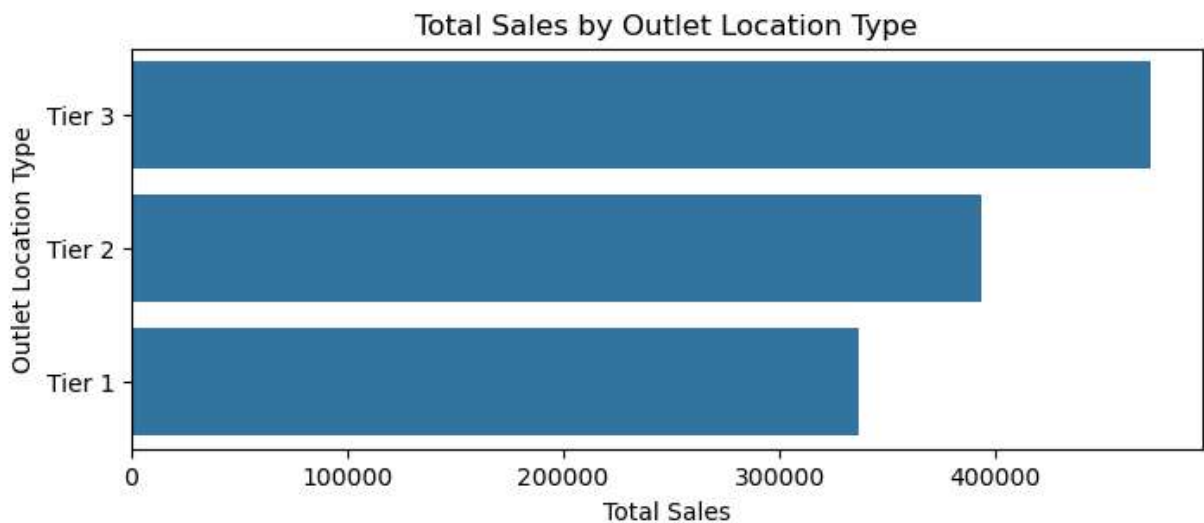
```
In [163... sales_by_year = df.groupby ('Outlet Establishment Year') ['Sales'].sum()
plt.figure(figsize=(9,5))
plt.plot(sales_by_year.index, sales_by_year, marker='o', linestyle='-')
plt.xlabel('Outlet Establishment Year')
plt.ylabel('Total Sales')
plt.title('Outlet Establishment')
plt.show()
```



```
In [167]: 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.show()
```



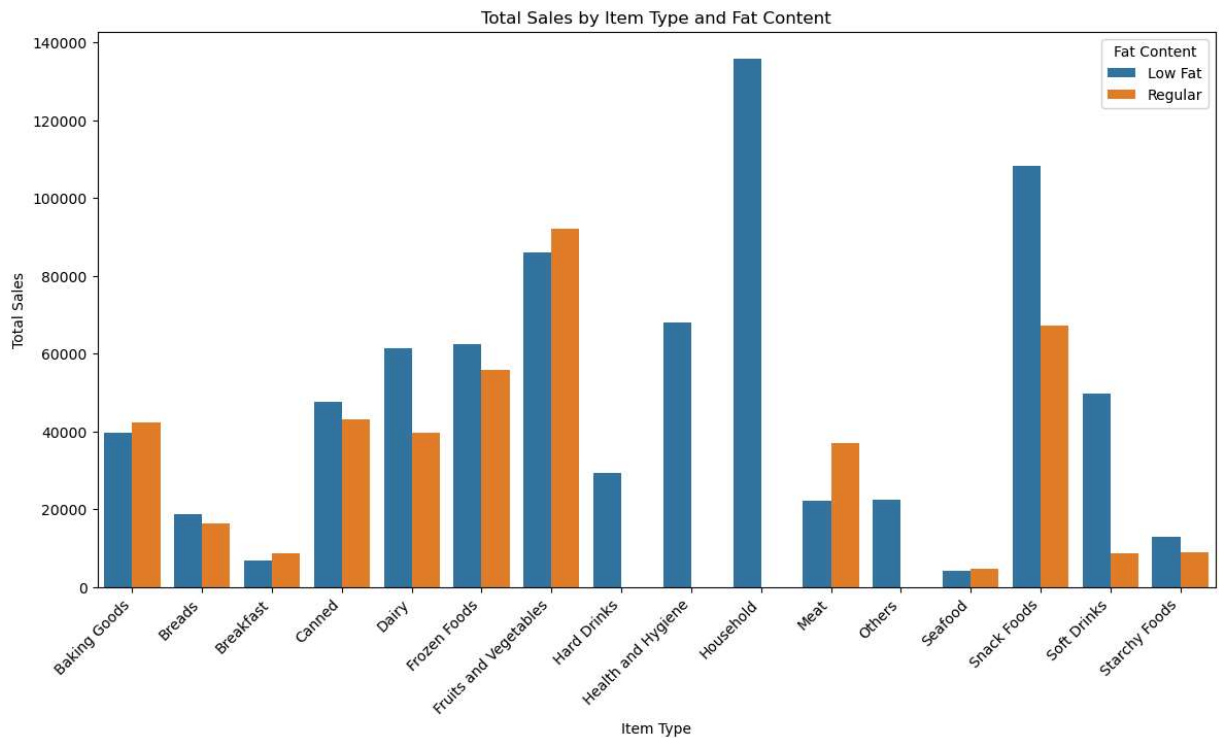
```
In [81]: sales_by_location= df.groupby('Outlet Location Type') ['Sales'].sum().reset_index()
sales_by_location=sales_by_location.sort_values('Sales', ascending=False)
plt.figure(figsize=(8,3))
sns.barplot(x='Sales', y='Outlet Location Type', data=sales_by_location)
plt.title('Total Sales by Outlet Location Type')
plt.xlabel('Total Sales')
plt.ylabel('Outlet Location Type')
plt.show()
```



```
In [50]: grouped = df.groupby(['Item Type', 'Item Fat Content']).agg(
    Total_Sales=('Sales', 'sum'))
```

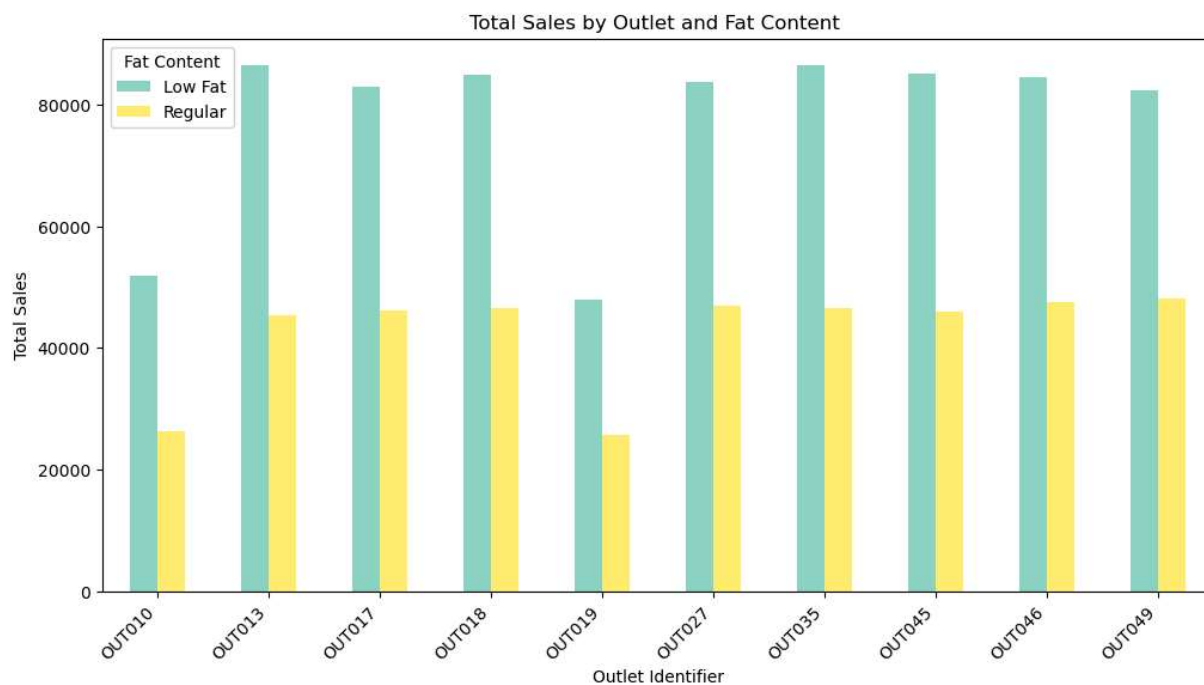


```
# Step 3: Plotting Total Sales by Item Type & Fat Content as a Bar Chart
plt.figure(figsize=(14, 7))
sns.barplot(data=grouped, x='Item Type', y='Total_Sales', hue='Item Fat Content')
plt.title('Total Sales by Item Type and Fat Content')
plt.xticks(rotation=45, ha='right')
plt.xlabel('Item Type')
plt.ylabel('Total Sales')
plt.legend(title='Fat Content')
plt.show()
```



In [151...

```
group = df.groupby(['Outlet Identifier', 'Item Fat Content']).agg(
    Total_Sales=('Sales', 'sum')
).reset_index()
data = group.pivot(index='Outlet Identifier', columns='Item Fat Content', values='Total_Sales')
data.plot(kind='bar', figsize=(12, 6), colormap='Set3')
plt.title('Total Sales by Outlet and Fat Content ')
plt.xlabel('Outlet Identifier')
plt.ylabel('Total Sales')
plt.xticks(rotation=45, ha='right')
plt.legend(title='Fat Content')
plt.show()
```



In []: