BLINKIT EDA PROJECT

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(r"C:\blinkit_project\blinkit_data.csv")
```

Data Types and info

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
```

Raw Data

In [4]:
```
df.head()
```

Out[4]:

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

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

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

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')
```

In [7]:
```
df["Item Fat Content"].value_counts()
```

Out[7]:
```
Item Fat Content
Low Fat    5089
Regular    2889
LF          316
reg         117
low fat     112
Name: count, dtype: int64
```

Data Cleaning

In [8]:
```
df["Item Fat Content"]=df["Item Fat Content"].replace({'LF':"Low Fat",
                                                       'low fat': 'Low Fat',
                                                       'reg':'Regular'})
```

In [9]:
```
df["Item Fat Content"].value_counts()
```

Out[9]:
```
Item Fat Content
Low Fat    5517
Regular    3006
Name: count, dtype: int64
```

Business Requirements

KPI's Requirements

In [10]:
```
#Total_sales
Total_sales=df["Sales"].sum()
```

In [11]:
```
#Average_sales
Avg_sales=df["Sales"].mean()
```

In [12]:
```
#Number of items sold
no_of_item_sold=df["Sales"].count()
```

In [13]:
```
#Average Ratings
Avg_ratings=df["Rating"].mean()
```

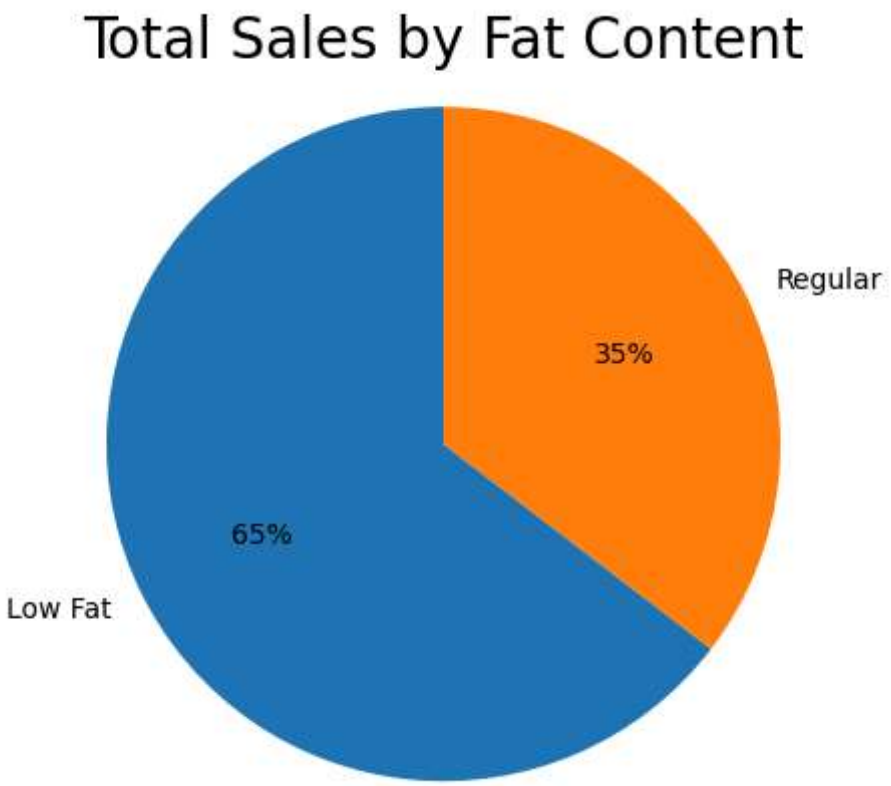
In [14]:
```
print(f"Total Sales: ${Total_sales:,.0f}")
print(f"Average_sales: ${Avg_sales:,.0f}")
print(f"No of Items sold: ${no_of_item_sold:,.0f}")
print(f"Average_Rating: ${Avg_ratings:,.1f}")
```

```
Total Sales: $1,201,681
Average_sales: $141
No of Items sold: $8,523
Average_Rating: $4.0
```

Charts Requirements

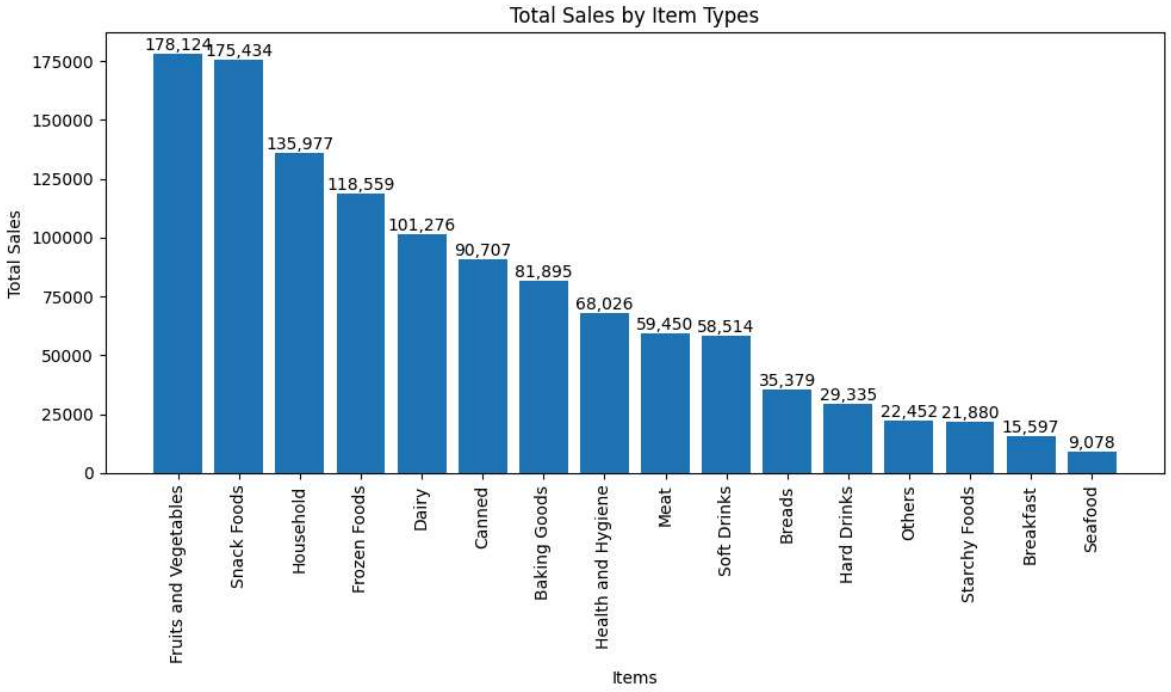
Total Sales by Fat content

In [15]:
```
Sales_by_fat=df.groupby("Item Fat Content")["Sales"].sum()

plt.pie(Sales_by_fat,labels=Sales_by_fat.index,startangle=90,autopct='%.0f%%')
plt.axis('equal')
plt.title("Total Sales by Fat Content",size=20)
plt.show()
```



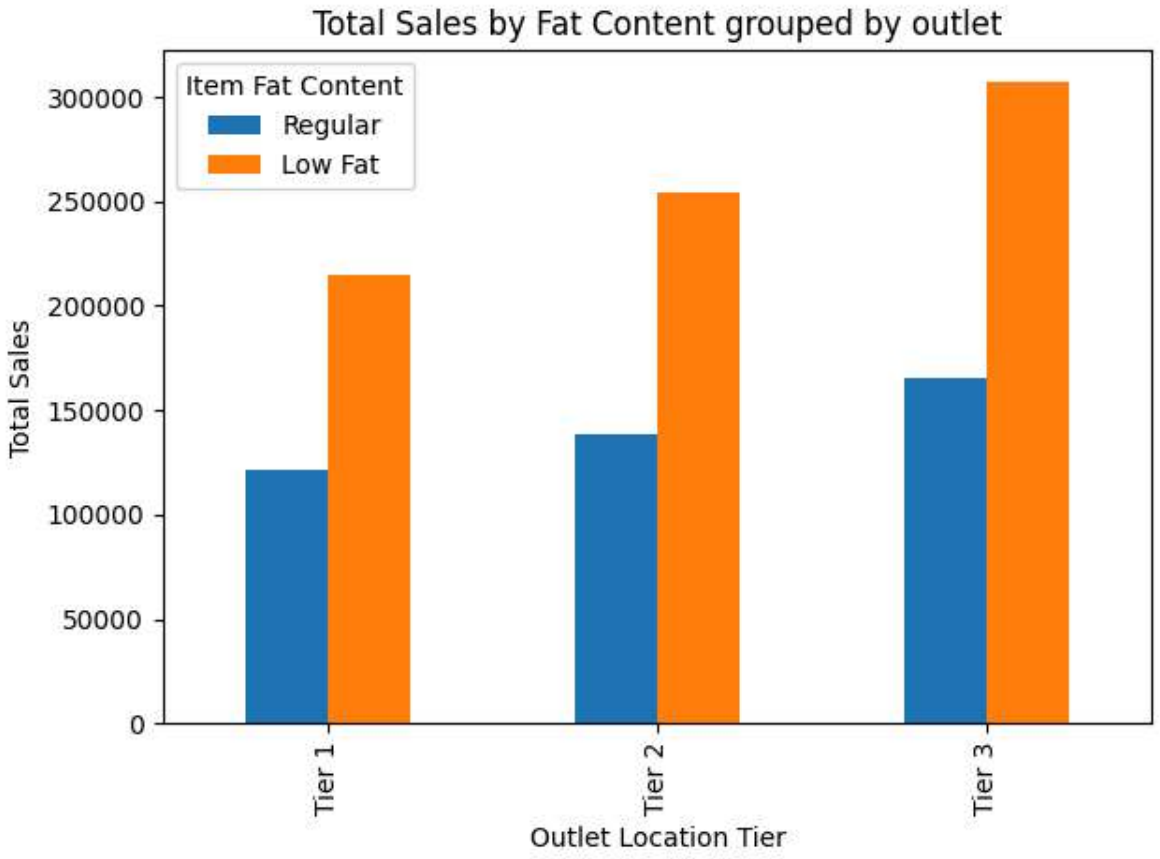
Total Sales by Item Types

In [16]:
```
sales_by_item_type=df.groupby("Item Type")["Sales"].sum().sort_values(ascending=

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

plt.xticks(rotation=90)
plt.xlabel("Items")
plt.ylabel("Total Sales")
plt.title("Total Sales by Item Types")

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", )
plt.tight_layout()
plt.show()
```



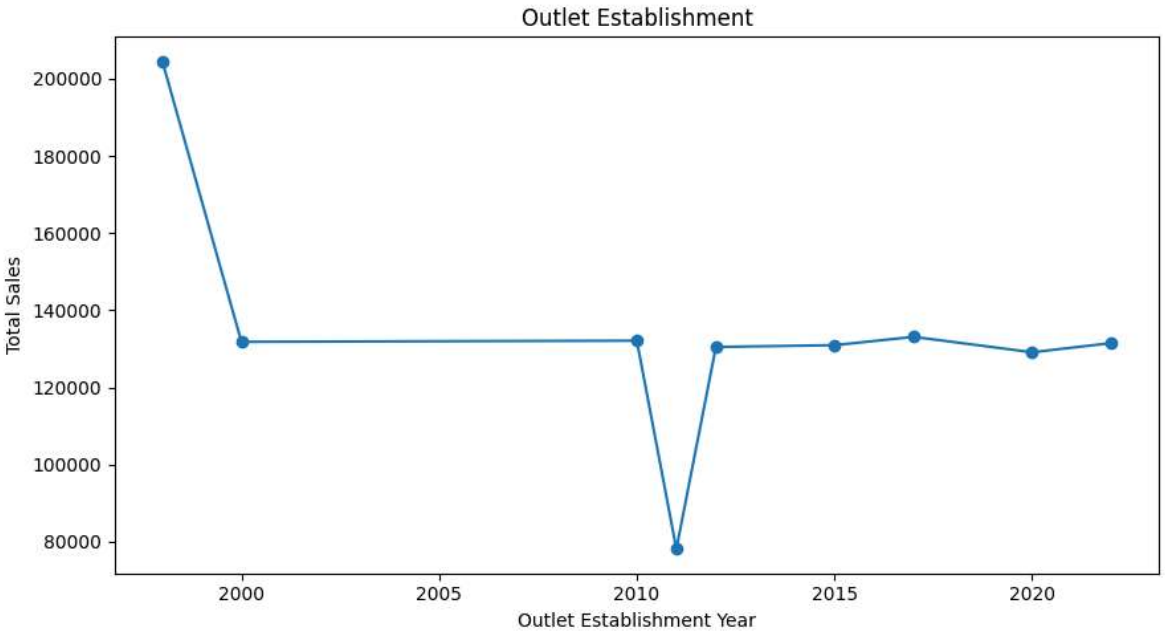
Total Sales by Fat Content grouped by outlet

In [17]:
```
grouped=df.groupby(["Outlet Location Type","Item Fat Content"])["Sales"].sum().u
grouped=grouped[["Regular","Low Fat"]]

ax=grouped.plot(kind='bar')
plt.xlabel("Outlet Location Tier")
plt.ylabel("Total Sales")
plt.title("Total Sales by Fat Content grouped by outlet")
plt.legend(title="Item Fat Content")
plt.tight_layout()

plt.show()
```



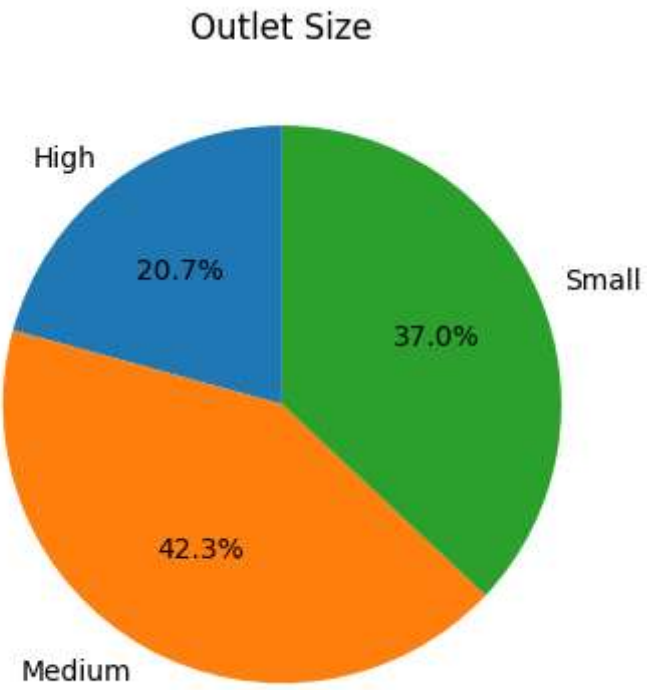
Total Sales by Outlet Establishment

In [18]:
```
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",linestyle="-")

plt.xlabel("Outlet Establishment Year")
plt.ylabel("Total Sales")
plt.title("Outlet Establishment")

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



Sales by Outlet Size

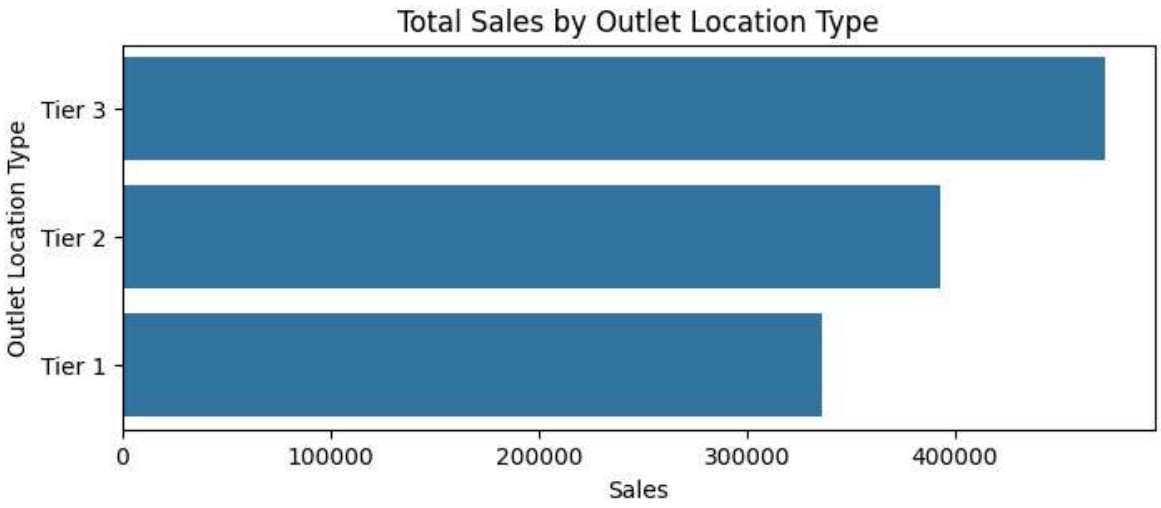
In [19]:
```
sales_by_outlet_size=df.groupby("Outlet Size")["Sales"].sum()

plt.figure(figsize=(4,4))
plt.pie(sales_by_outlet_size,labels=sales_by_outlet_size.index,autopct="%1.1f%%"
plt.title("Outlet Size")
plt.tight_layout()
plt.show()
```



Total Sales by Outlet Location type

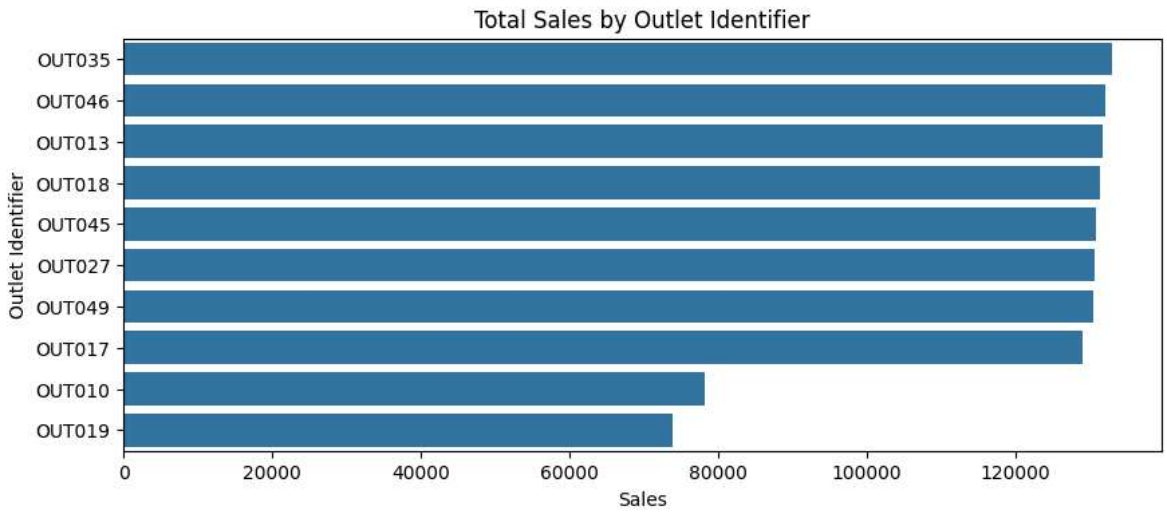
In [20]:
```
sales_by_location=df.groupby("Outlet Location Type")["Sales"].sum().reset_index(
sales_by_location=sales_by_location.sort_values("Sales",ascending=False)
```

```
plt.figure(figsize=(8,3))
ax=sns.barplot(data=sales_by_location,x="Sales",y="Outlet Location Type")
plt.title("Total Sales by Outlet Location Type")
plt.show()
```



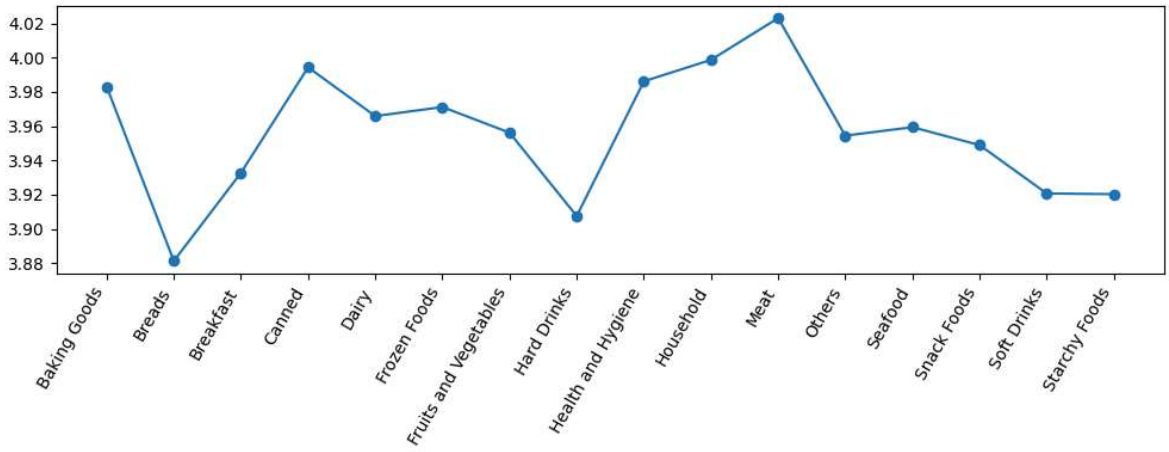
Total Sales by Outlet Identifier

In [21]:
```
Total_Sales_by_outlet_identifier=df.groupby("Outlet Identifier")["Sales"].sum().
Total_Sales_by_outlet_identifier=Total_Sales_by_outlet_identifier.sort_values("S
plt.figure(figsize=(10,4))
bar=sns.barplot(data=Total_Sales_by_outlet_identifier,x="Sales",y="Outlet Identi

plt.title("Total Sales by Outlet Identifier")
plt.show()
```



Average Ratings by Item Types

In [23]:
```
Avg_ratings_by_Itemtype=df.groupby("Item Type")["Rating"].mean()
plt.figure(figsize=(10,4))
plt.plot(Avg_ratings_by_Itemtype.index,Avg_ratings_by_Itemtype.values,marker="o"

plt.xticks(rotation=60,ha="right")
plt.tight_layout()
plt.show()
```


4.02
4.00
3.98
3.96
3.94
3.92
3.90
3.88
Baking Goods
Breads
Breakfast
Canned
Dairy
Frozen Foods
Fruits and Vegetables
Hard Drinks
Health and Hygiene
Household
Meat
Others
Seafood
Snack Foods
Soft Drinks
Starchy Foods