

## ▼ \*\*Data Analysis - Blinkit Analysis\*\*

```
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('/content/blinkit_data.csv')
df.head()
```

	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
0	Regular	FDX32	Fruits and Vegetables	2012	OUT049	Tier 1	Medium	Supermarket Type1	0.100014	15.10	145.4786	5.0
1	Low Fat	NCB42	Health and Hygiene	2022	OUT018	Tier 3	Medium	Supermarket Type2	0.008596	11.80	115.3492	5.0
2	Regular	FDR28	Frozen Foods	2010	OUT046	Tier 1	Small	Supermarket Type1	0.025896	13.85	165.0210	5.0
3	Regular	FDL50	Canned	2000	OUT013	Tier 3	High	Supermarket Type1	0.042278	12.15	126.5046	5.0
4	Low Fat	DRI25	Soft Drinks	2015	OUT045	Tier 2	Small	Supermarket Type1	0.033970	19.60	55.1614	5.0

```
df.head()
```

	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
0	Regular	FDX32	Fruits and Vegetables	2012	OUT049	Tier 1	Medium	Supermarket Type1	0.100014	15.10	145.4786	5.0
1	Low Fat	NCB42	Health and Hygiene	2022	OUT018	Tier 3	Medium	Supermarket Type2	0.008596	11.80	115.3492	5.0
2	Regular	FDR28	Frozen Foods	2010	OUT046	Tier 1	Small	Supermarket Type1	0.025896	13.85	165.0210	5.0
3	Regular	FDL50	Canned	2000	OUT013	Tier 3	High	Supermarket Type1	0.042278	12.15	126.5046	5.0
4	Low Fat	DRI25	Soft Drinks	2015	OUT045	Tier 2	Small	Supermarket Type1	0.033970	19.60	55.1614	5.0

```
df['Item Fat Content'] = df['Item Fat Content'].str.replace('LF', 'Low Fat').str.replace('low fat', 'Low Fat').str.replace('reg',
```

```
df.head(10)
```

	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
0	Regular	FDX32	Fruits and Vegetables	2012	OUT049	Tier 1	Medium	Supermarket Type1	0.100014	15.10	145.4786	5.0
1	Low Fat	NCB42	Health and Hygiene	2022	OUT018	Tier 3	Medium	Supermarket Type2	0.008596	11.80	115.3492	5.0
2	Regular	FDR28	Frozen Foods	2010	OUT046	Tier 1	Small	Supermarket Type1	0.025896	13.85	165.0210	5.0
3	Regular	FDL50	Canned	2000	OUT013	Tier 3	High	Supermarket Type1	0.042278	12.15	126.5046	5.0
4	Low Fat	DRI25	Soft Drinks	2015	OUT045	Tier 2	Small	Supermarket Type1	0.033970	19.60	55.1614	5.0
5	Low Fat	FDS52	Frozen Foods	2020	OUT017	Tier 2	Small	Supermarket Type1	0.005505	8.89	102.4016	5.0
6	Low Fat	NCU05	Health and Hygiene	2011	OUT010	Tier 3	Small	Grocery Store	0.098312	11.80	81.4618	5.0
7	Low Fat	NCD30	Household	2015	OUT045	Tier 2	Small	Supermarket Type1	0.026904	19.70	96.0726	5.0
8	Low Fat	FDW20	Fruits and Vegetables	2000	OUT013	Tier 3	High	Supermarket Type1	0.024129	20.75	124.1730	5.0
9	Low Fat	FDX25	Canned	1998	OUT027	Tier 3	Medium	Supermarket Type1	0.101562	NaN	181.9292	5.0

df.columns

```
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')
```

Start coding or [generate](#) with AI.

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
```

Business Requirments

KPI's Requiremnts

```
# Total sales
Total_sales = df['Sales'].sum()
print(Total_sales)
```

1201681.4808

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

```
print(Average_sales)
```

```
140.9927819781767
```

```
# No. of Items Sold  
No_of_items_sold = df['Sales'].count()  
print(No_of_items_sold)
```

```
8523
```

```
# Average Rating  
avg_rating = df['Rating'].mean()  
print(avg_rating)
```

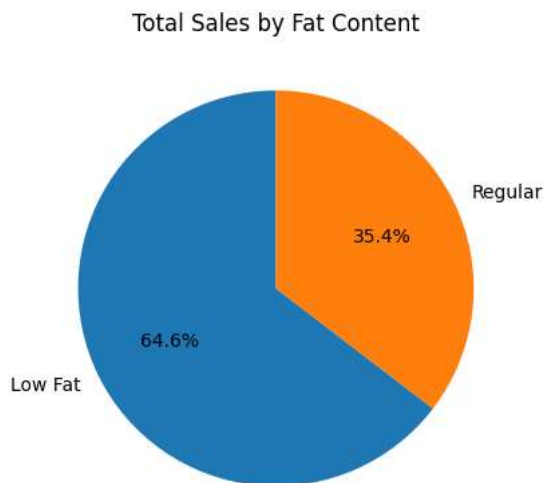
```
3.965857092573038
```

```
print(f"Total Sales: {Total_sales:.1f}")  
print(f"Average Sales: {Average_sales:.1f}")  
print(f"No. of Items Sold: {No_of_items_sold:.0f}")  
print(f"Average Rating: {avg_rating:.1f}")
```

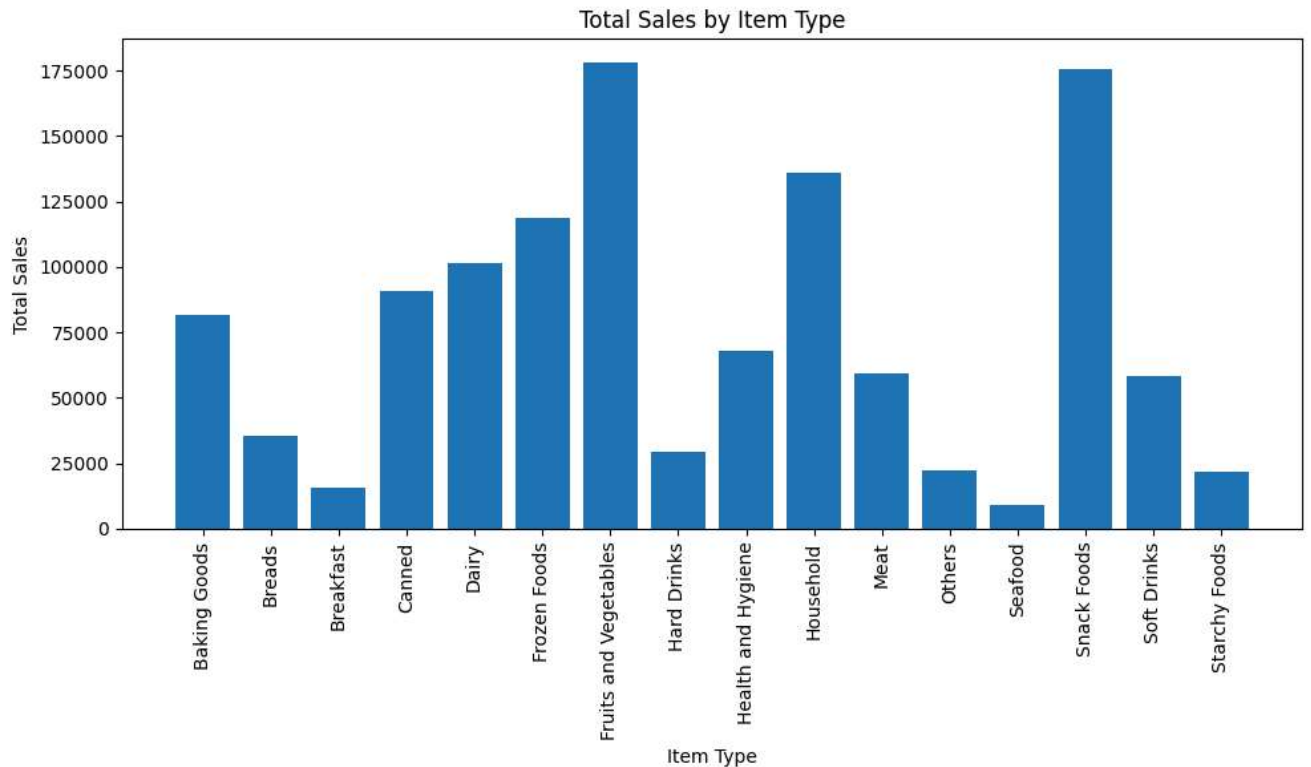
```
Total Sales: 1201681.5  
Average Sales: 141.0  
No. of Items Sold: 8523  
Average Rating: 4.0
```

### Charts Requirements

```
# Total Sales by Fat Content  
total_sales_by_fat_content = df.groupby('Item Fat Content')['Sales'].sum()  
plt.pie(total_sales_by_fat_content, labels=total_sales_by_fat_content.index,  
        autopct='%1.1f%%', startangle=90)  
plt.title('Total Sales by Fat Content')  
plt.show()
```

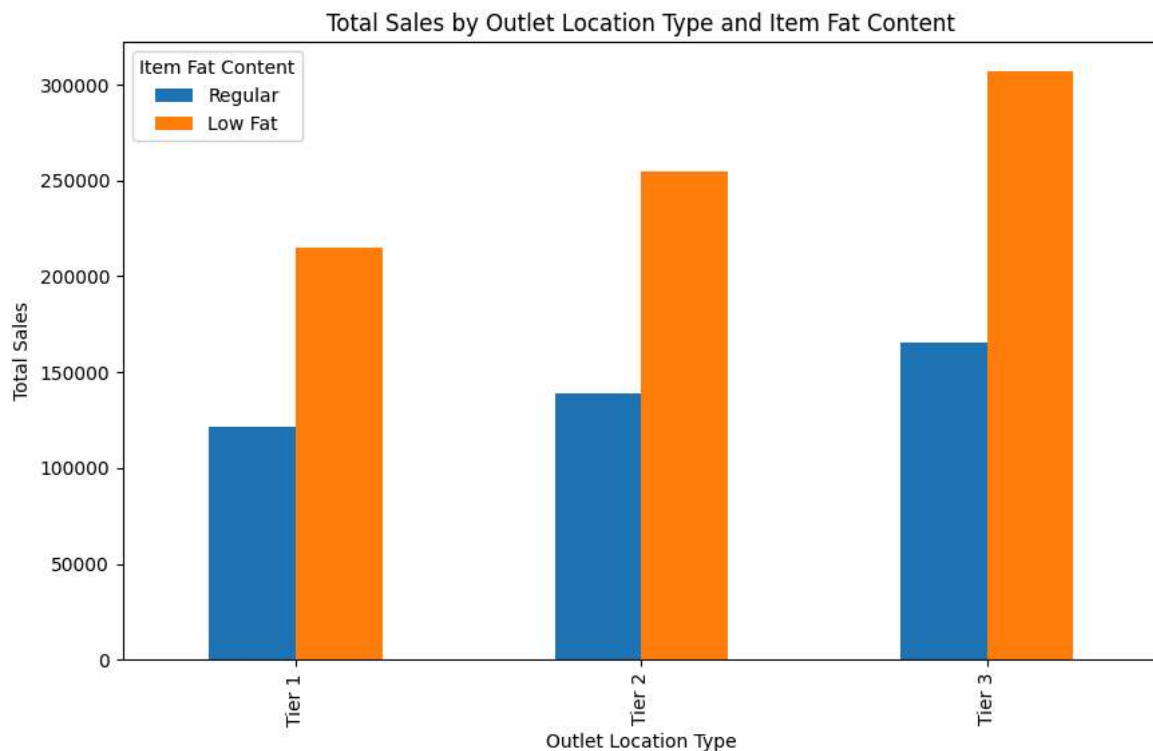


```
# Total sales by item type  
plt.figure(figsize=(10,6))  
total_sales_by_item_type = df.groupby('Item Type')['Sales'].sum()  
plt.bar(total_sales_by_item_type.index, total_sales_by_item_type.values)  
plt.xticks(rotation=90)  
plt.xlabel('Item Type')  
plt.ylabel('Total Sales')  
plt.title('Total Sales by Item Type')  
plt.tight_layout()  
plt.show()
```



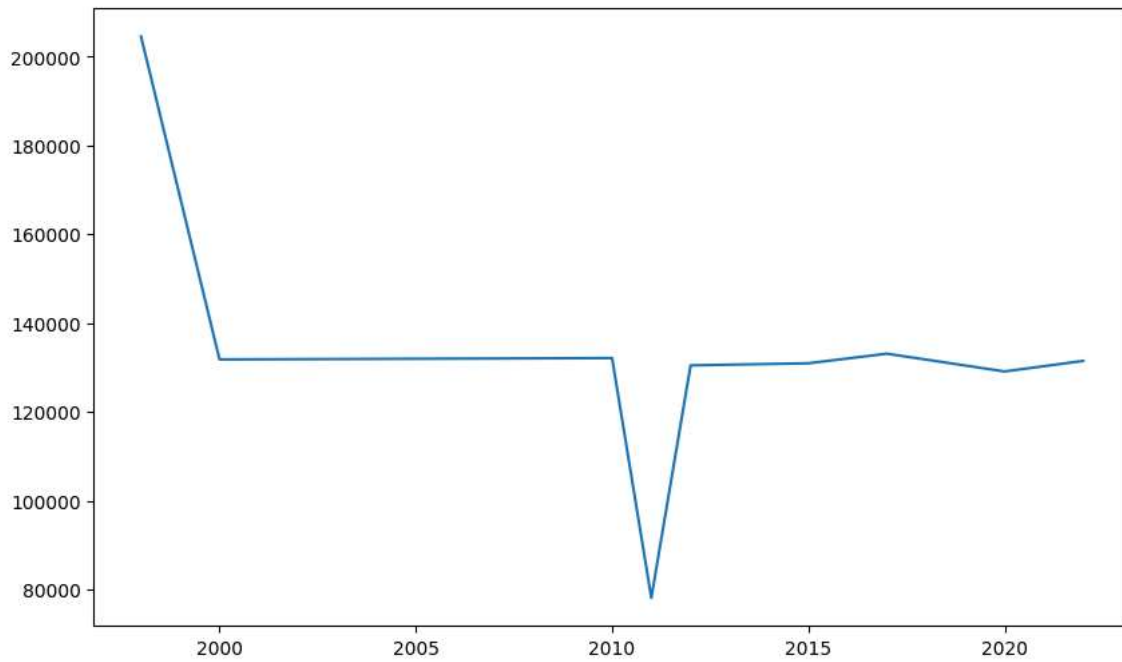
```
grouped = df.groupby(['Outlet Location Type', 'Item Fat Content'])['Sales'].sum().unstack()
grouped = grouped[['Regular', 'Low Fat']]
ax = grouped.plot(kind='bar', figsize=(10,6))
plt.xlabel('Outlet Location Type')
plt.ylabel('Total Sales')
plt.title('Total Sales by Outlet Location Type and Item Fat Content')
```

Text(0.5, 1.0, 'Total Sales by Outlet Location Type and Item Fat Content')



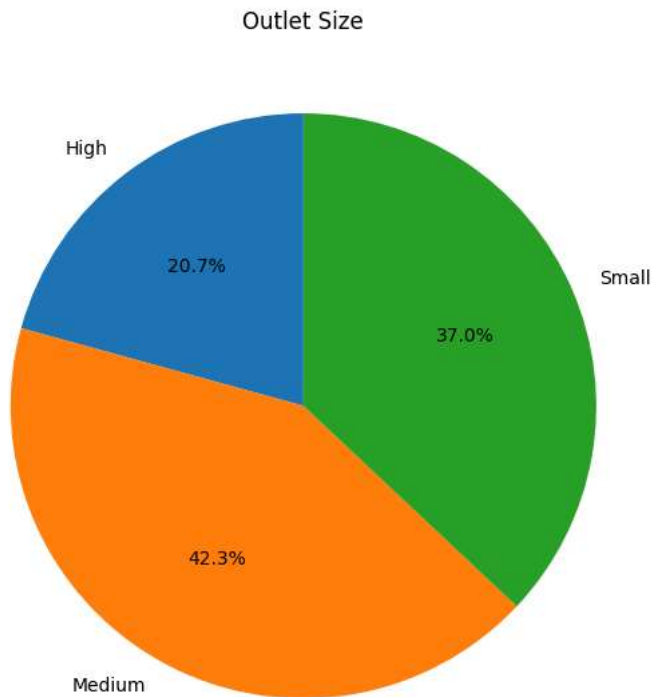
```
# Total sales by outlet Establishment
plt.figure(figsize=(10,6))
total_sales_by_outlet_establishment = df.groupby('Outlet Establishment Year')['Sales'].sum()
plt.plot(total_sales_by_outlet_establishment.index, total_sales_by_outlet_establishment.values)
```

```
[<matplotlib.lines.Line2D at 0x7c91edf437d0>]
```



```
# Sales by outlet_size
```

```
sales_by_size = df.groupby('Outlet Size')['Sales'].sum()
plt.figure(figsize=(10,6))
plt.pie(sales_by_size, labels=sales_by_size.index, autopct='%1f%%', startangle=90)
plt.title('Outlet Size')
plt.tight_layout()
plt.show()
```



```
# Sales by outlier location
sales_by_location = df.groupby('Outlet Location Type')['Sales'].sum()
plt.figure(figsize=(8,6))
ax = sns.barplot(y=sales_by_location.index, x=sales_by_location.values)
plt.xlabel('Outlet Location Type')
plt.ylabel('Total Sales')
```

```
plt.title('Total Sales by Outlet Location Type')  
plt.tight_layout()  
plt.show()
```

