

DATA ANALYSIS PYTHON PROJECT - BLINKIT ANALYSIS

Import Libraries

```
In [2]: import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```

Import Raw Data

```
In [10]: df = pd.read_csv(r"C:\Users\kisha\Downloads\blinkit_data.csv")
```

Sample Data

```
In [14]: df.head(5)
```

```
Out[14]:
```

	Item Fat Content	Item Identifier	Item Type	Outlet Establishment Year	Outlet Identifier	Outlet Location Type	Outlet Size	Outlet Type	It Visibi
0	Regular	FDX32	Fruits and Vegetables	2012	OUT049	Tier 1	Medium	Supermarket Type1	0.100
1	Low Fat	NCB42	Health and Hygiene	2022	OUT018	Tier 3	Medium	Supermarket Type2	0.008
2	Regular	FDR28	Frozen Foods	2010	OUT046	Tier 1	Small	Supermarket Type1	0.025
3	Regular	FDL50	Canned	2000	OUT013	Tier 3	High	Supermarket Type1	0.042
4	Low Fat	DRI25	Soft Drinks	2015	OUT045	Tier 2	Small	Supermarket Type1	0.033

```
In [15]: df.tail(5)
```

Out[15]:

	Item Fat Content	Item Identifier	Item Type	Outlet Establishment Year	Outlet Identifier	Outlet Location Type	Outlet Size	Outlet Type	Visil
8518	low fat	NCT53	Health and Hygiene	1998	OUT027	Tier 3	Medium	Supermarket Type3	0.00
8519	low fat	FDN09	Snack Foods	1998	OUT027	Tier 3	Medium	Supermarket Type3	0.03
8520	low fat	DRE13	Soft Drinks	1998	OUT027	Tier 3	Medium	Supermarket Type3	0.02
8521	reg	FDT50	Dairy	1998	OUT027	Tier 3	Medium	Supermarket Type3	0.10
8522	reg	FDM58	Snack Foods	1998	OUT027	Tier 3	Medium	Supermarket Type3	0.00

Number of rows and columns

In [17]: `print("Size of data:",df.shape)`

Size of data: (8523, 12)

Column Names

In [18]: `df.columns`

Out[18]: 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')

Datatypes

In [19]: `df.dtypes`

Out[19]: 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

Data Cleaning the Item Fat Content Column

```
In [21]: print(df['Item Fat Content'].unique())
['Regular' 'Low Fat' 'low fat' 'LF' 'reg']

In [22]: df['Item Fat Content'] = df['Item Fat Content'].replace({'LF':'Low Fat','low fat':'Low Fat'})

In [24]: print(df['Item Fat Content'].unique())
['Regular' 'Low Fat']
```

Business Requirements

KPIs Requirements

```
In [48]: # Total Sales
total_sales = df['Sales'].sum()

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

# Number of items sold
no_of_item_sold = df['Item Identifier'].count()

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

# Display
print(f'Total Sales: ${total_sales:,.1f}')
print(f'Average Sales: ${avg_sales:,.1f}')
print(f'Number of items sold: {no_of_item_sold}')
print(f'Average Rating: {avg_rating:.2f}')
```

Total Sales: \$1,201,681.5
 Average Sales: \$141.0
 Number of items sold: 8523
 Average Rating: 3.97

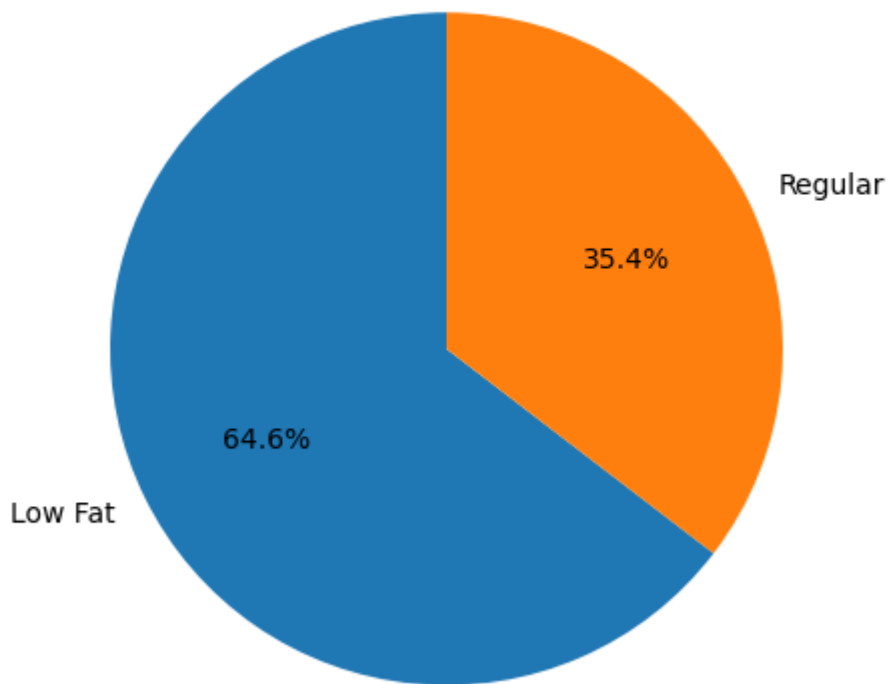
Charts Requirements

```
In [67]: sales_by_fat = df.groupby('Item Fat Content')['Sales'].sum()

plt.pie(sales_by_fat,
        labels=sales_by_fat.index,
        autopct='%0.1f%%',
        startangle=90)

plt.title('Sales by Fat Content')
plt.axis('equal') # Ensures a circular pie
plt.show()
```

Sales by Fat Content



Total Sales By Item Type

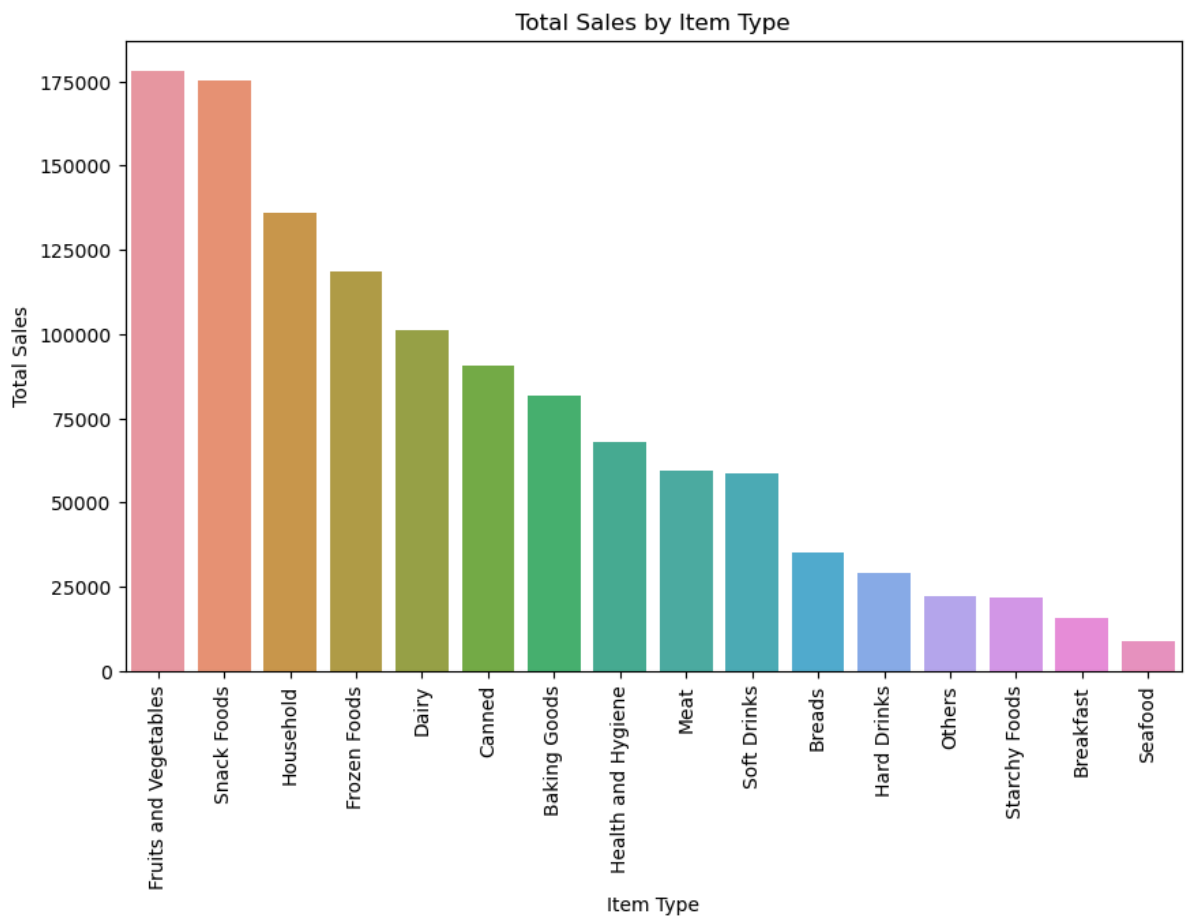
```
In [74]: sales_by_item_type = df.groupby('Item Type')['Sales'].sum().reset_index().sort_valu
```

```
In [75]: sales_by_item_type.head()
```

```
Out[75]:
```

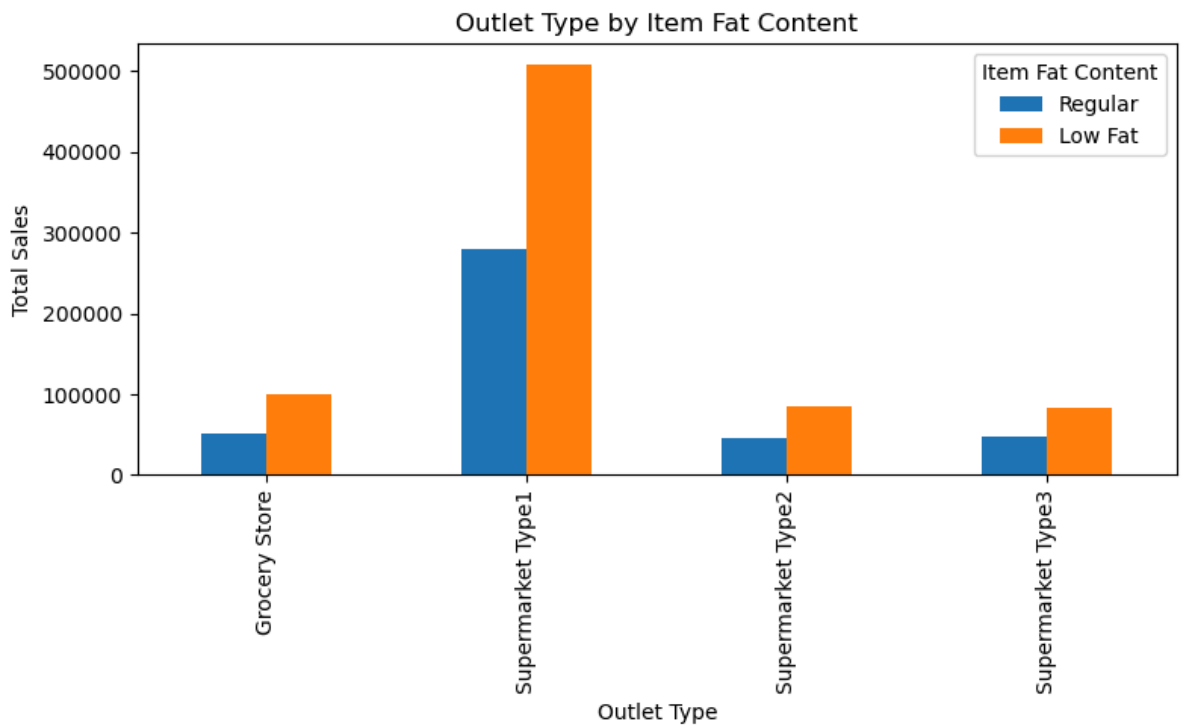
	Item Type	Sales
6	Fruits and Vegetables	178124.0810
13	Snack Foods	175433.9204
9	Household	135976.5254
5	Frozen Foods	118558.8814
4	Dairy	101276.4596

```
In [80]: plt.figure(figsize = (10,6))
sns.barplot(data = sales_by_item_type,x = 'Item Type',y = 'Sales')
plt.xticks(rotation = 90)
plt.xlabel('Item Type')
plt.ylabel('Total Sales')
plt.title('Total Sales by Item Type')
plt.show()
```



Fat Content by Outlet for Total Sales

```
In [87]: grouped = df.groupby(['Outlet Type', 'Item Fat Content'])['Sales'].sum().unstack()
grouped = grouped[['Regular', 'Low Fat']] # Only if these exist as Item Fat Content
ax = grouped.plot(kind='bar', figsize=(8,5), title='Outlet Type by Item Fat Content')
plt.xlabel('Outlet Type')
plt.ylabel('Total Sales')
plt.legend(title='Item Fat Content')
plt.tight_layout()
plt.show()
```

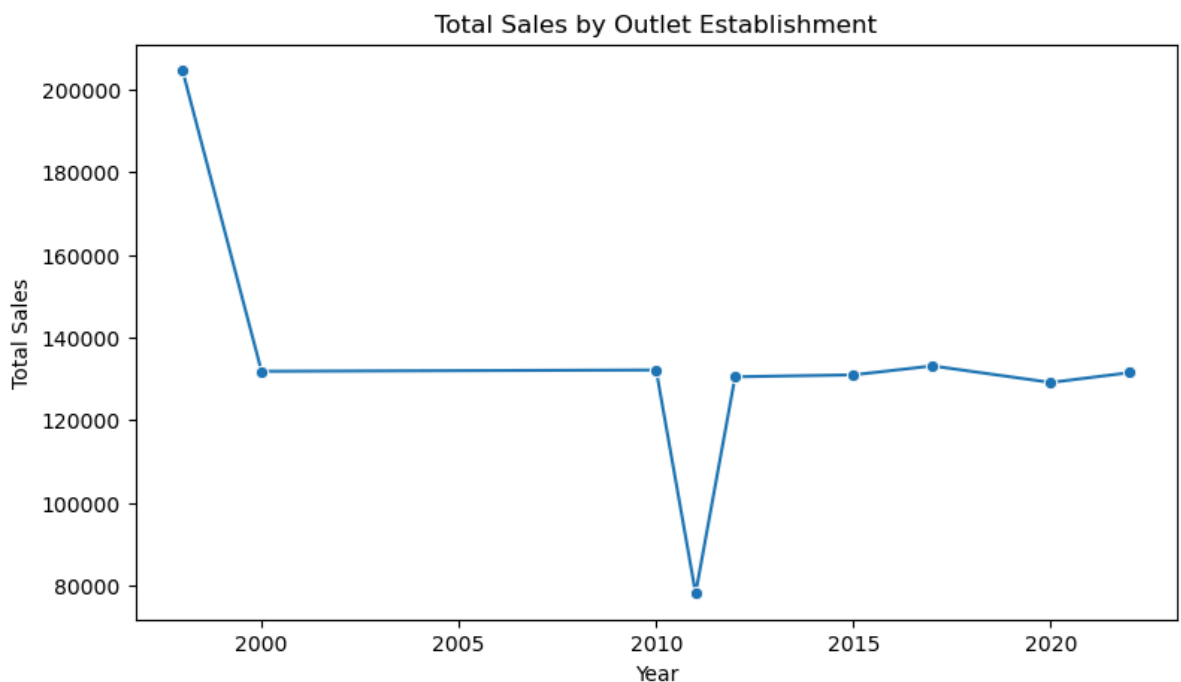


Total Sales by Outlet Establishment

```
In [93]: df_sales_by_establishment = df.groupby('Outlet Establishment Year')['Sales'].sum()
```

```
In [100... plt.figure(figsize=(9,5))
sns.lineplot(data = df_sales_by_establishment,x = 'Outlet Establishment Year',y = 'Sales')

plt.title('Total Sales by Outlet Establishment')
plt.xlabel('Year')
plt.ylabel('Total Sales')
plt.show()
```

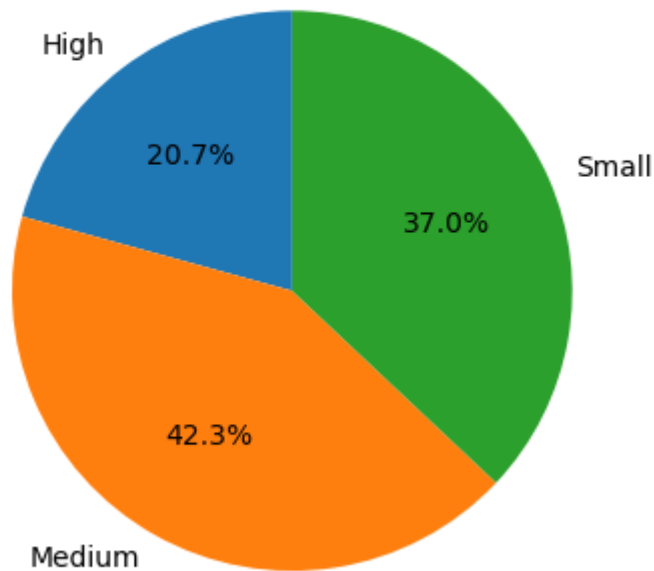


Sales by Outlet Size

```
In [105... df_sales_by_outlet = df.groupby('Outlet Size')['Sales'].sum().reset_index()
```

```
In [110... plt.figure(figsize = (4,4))  
plt.pie(df_sales_by_outlet['Sales'],labels = df_sales_by_outlet['Outlet Size'],star  
plt.title('Sales by Outlet Size')  
plt.tight_layout()  
plt.show()
```

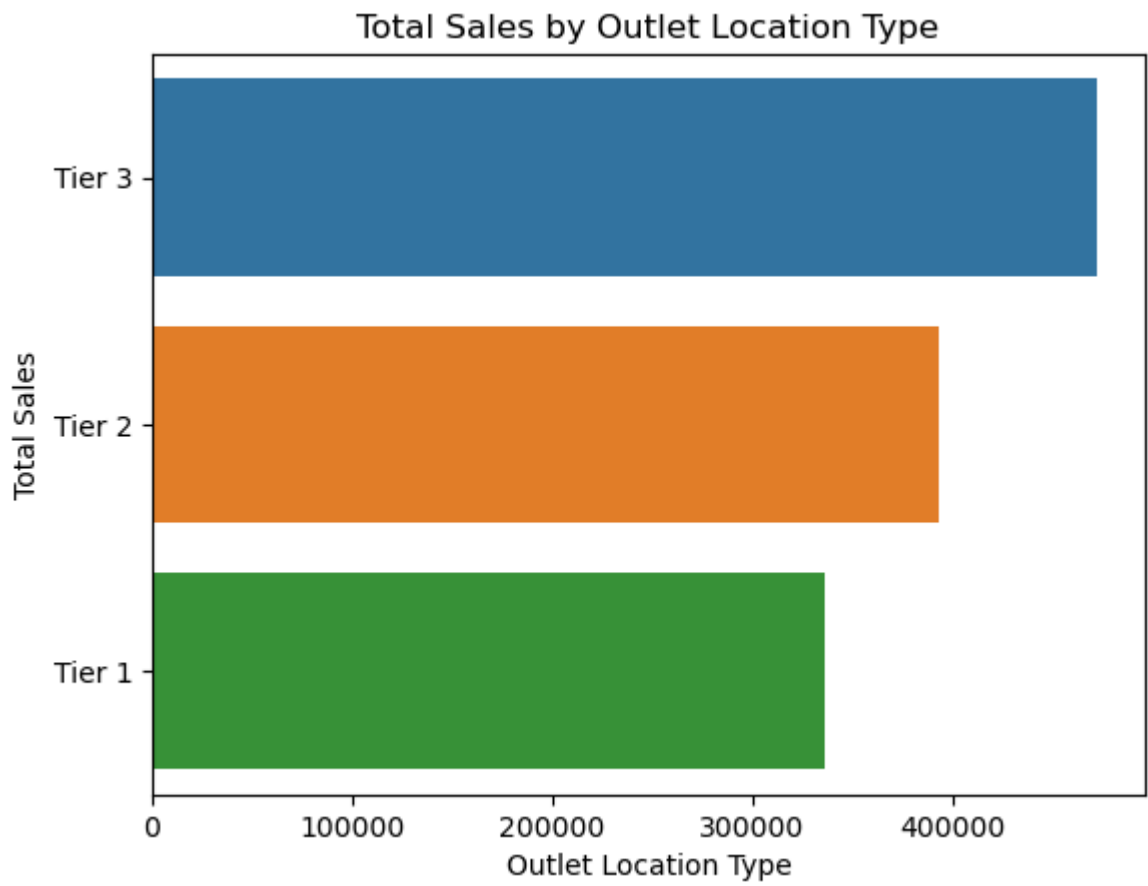
Sales by Outlet Size



Sales by Outlet Location

```
In [120... df_sales_by_location = df.groupby('Outlet Location Type')['Sales'].sum().reset_index()
```

```
In [122... sns.barplot(data = df_sales_by_location,x = 'Sales',y = 'Outlet Location Type')  
plt.title('Total Sales by Outlet Location Type')  
plt.xlabel('Outlet Location Type')  
plt.ylabel('Total Sales')  
plt.show()
```



In []: