

DATA ANALYSIS PYTHON PROJECT - BLINKIT ANALYSIS

Import Libraries

```
In [3]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

Import Raw Data

```
In [5]: df = pd.read_csv("C:/Users/HP/OneDrive/Desktop/blinkit_data.csv")
```

Sample Data

```
In [7]: df.head(10)
```

Out[7]:

	Item Fat Content	Item Identifier	Item Type	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
5	low fat	FDS52	Frozen Foods	2020	OUT017	Tier 2	Small	Supermarke Type
6	Low Fat	NCU05	Health and Hygiene	2011	OUT010	Tier 3	Small	Grocery Store
7	Low Fat	NCD30	Household	2015	OUT045	Tier 2	Small	Supermarke Type
8	Low Fat	FDW20	Fruits and Vegetables	2000	OUT013	Tier 3	High	Supermarke Type
9	Low Fat	FDX25	Canned	1998	OUT027	Tier 3	Medium	Supermarke Type3

In [8]: `df.tail(10)`

Out[8]:

	Item Fat Content	Item Identifier	Item Type	Outlet Establishment Year	Outlet Identifier	Outlet Location Type	Outlet Size	Outlet Type
8513	Regular	DRY23	Soft Drinks	1998	OUT027	Tier 3	Medium	Supermarket
8514	low fat	FDA11	Baking Goods	1998	OUT027	Tier 3	Medium	Supermarket
8515	low fat	FDK38	Canned	1998	OUT027	Tier 3	Medium	Supermarket
8516	low fat	FDO38	Canned	1998	OUT027	Tier 3	Medium	Supermarket
8517	low fat	FDG32	Fruits and Vegetables	1998	OUT027	Tier 3	Medium	Supermarket
8518	low fat	NCT53	Health and Hygiene	1998	OUT027	Tier 3	Medium	Supermarket
8519	low fat	FDN09	Snack Foods	1998	OUT027	Tier 3	Medium	Supermarket
8520	low fat	DRE13	Soft Drinks	1998	OUT027	Tier 3	Medium	Supermarket
8521	reg	FDT50	Dairy	1998	OUT027	Tier 3	Medium	Supermarket
8522	reg	FDM58	Snack Foods	1998	OUT027	Tier 3	Medium	Supermarket



Size of Data

```
In [10]: print("Size of Data: ", df.shape)
```

Size of Data: (8523, 12)

Field Information

```
In [12]: df.columns
```

```
Out[12]: 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')
```

Data Types

```
In [14]: df.dtypes
```

```
Out[14]: 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

```
In [16]: print(df['Item Fat Content'].unique())
```

```
['Regular' 'Low Fat' 'low fat' 'LF' 'reg']
```

```
In [17]: df['Item Fat Content'] = df['Item Fat Content'].replace ({ 'low fat': 'Low Fat',
                                                                    'LF': 'Low Fat',
                                                                    'reg': 'Regular'})
```

```
In [18]: print(df['Item Fat Content'].unique())
```

```
['Regular' 'Low Fat']
```

BUSINESS REQUIREMENTS

KPI's Requirements

```
In [21]: #Total sales
total_sales = df['Sales'].sum()

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

#Number of items
num_of_sales= df['Sales'].count()

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

#Display
print(f"Total Sales: ${total_sales: ,.1f}")
print(f"Average of Sales: ${avg_sales: ,.1f}")
print(f"Number of items: {num_of_sales: ,.0f}")
print(f"Average rating: {avg_rating: ,.1f}")
```

Total Sales: \$ 1,201,681.5
Average of Sales: \$ 141.0
Number of items: 8,523
Average rating: 4.0

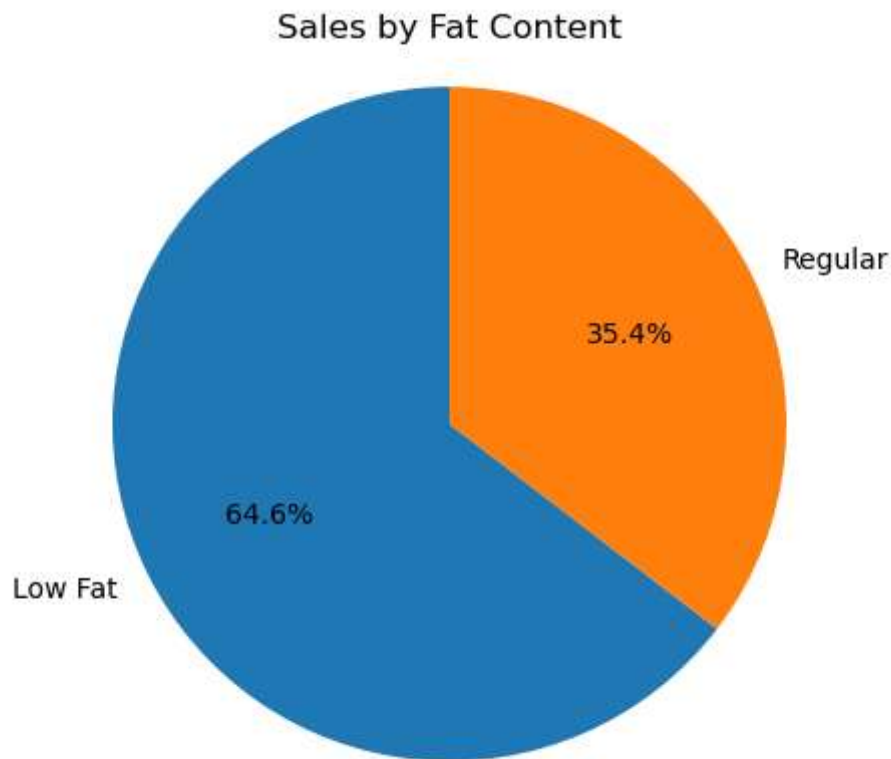
CHART'S REQUIREMENTS

Total Sales by Fat Content

```
In [39]: Sales_by_fat = df.groupby('Item Fat Content')['Sales'].sum()

plt.pie(Sales_by_fat, labels = Sales_by_fat.index,
        autopct = '%.1f%',
        startangle = 90)

plt.title('Sales by Fat Content')
plt.axis('equal')
plt.show()
```



Total Sales by Item Type

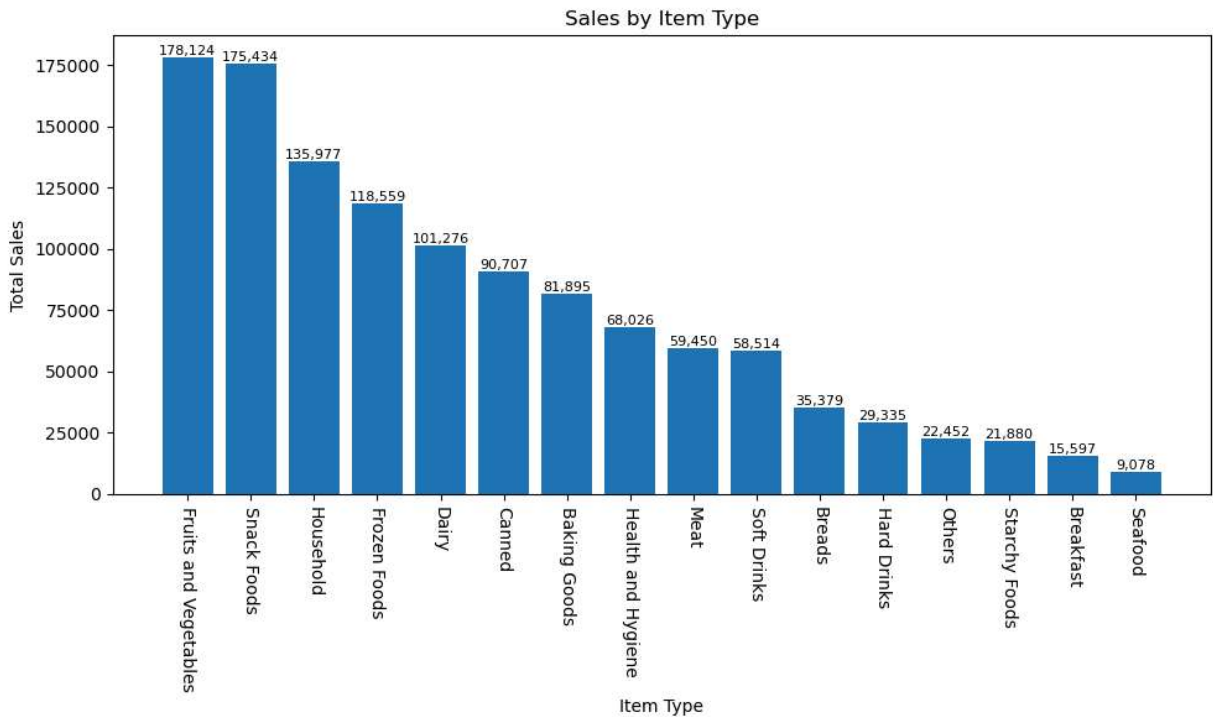
```
In [62]: Sales_by_Item_Type = df.groupby('Item Type')['Sales'].sum().sort_values(ascending=F
plt.figure(figsize=(10,6))
bars= plt.bar(Sales_by_Item_Type.index, Sales_by_Item_Type.values)

plt.xticks(rotation = -90)
plt.xlabel('Item Type')
plt.ylabel('Total Sales')
```

```
plt.title('Sales by Item Type')

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', fontsize = 8)

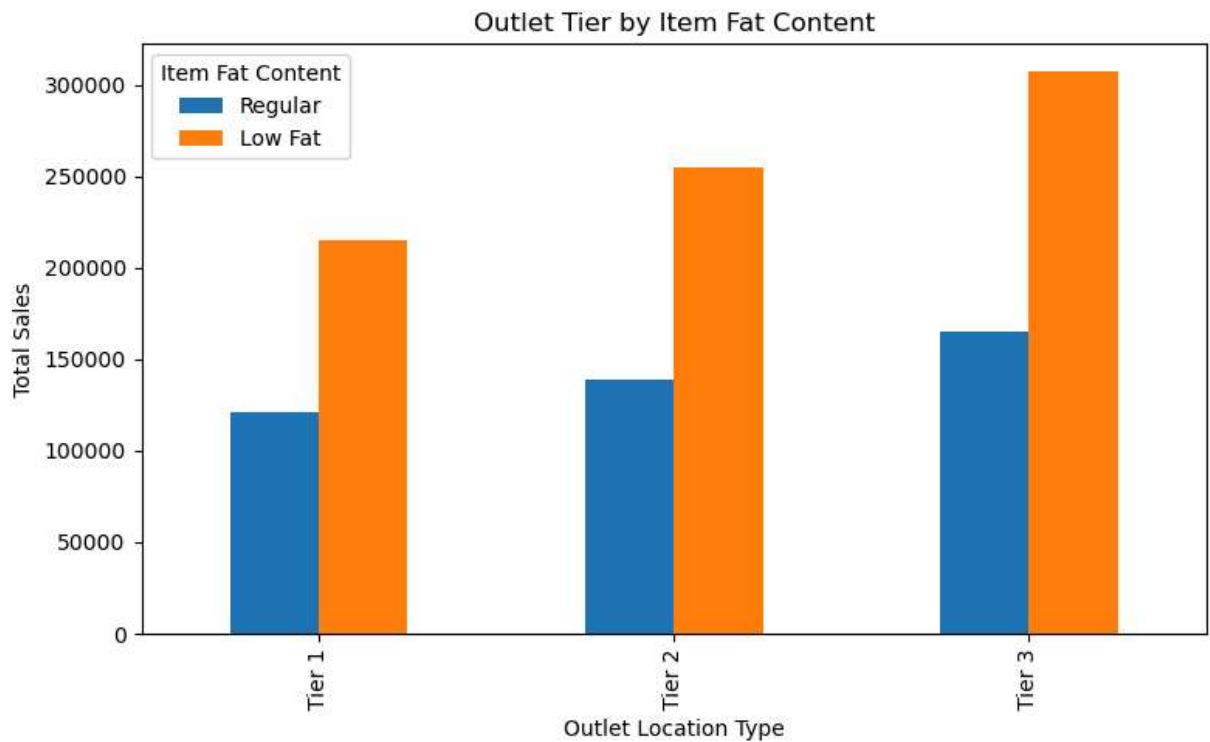
plt.tight_layout()
plt.show()
```



Fat Content by Outlet for Total Sales

```
In [71]: grouped = df.groupby(['Outlet Location Type', 'Item Fat Content'])['Sales'].sum().unstack()
grouped = grouped[['Regular', 'Low Fat']]

ax = grouped.plot( kind= 'bar', figsize = (8,5), title = 'Outlet Tier by Item Fat Content')
plt.xlabel('Outlet Location Type')
plt.ylabel('Total Sales')
plt.legend(title='Item Fat Content')
plt.tight_layout()
plt.show()
```



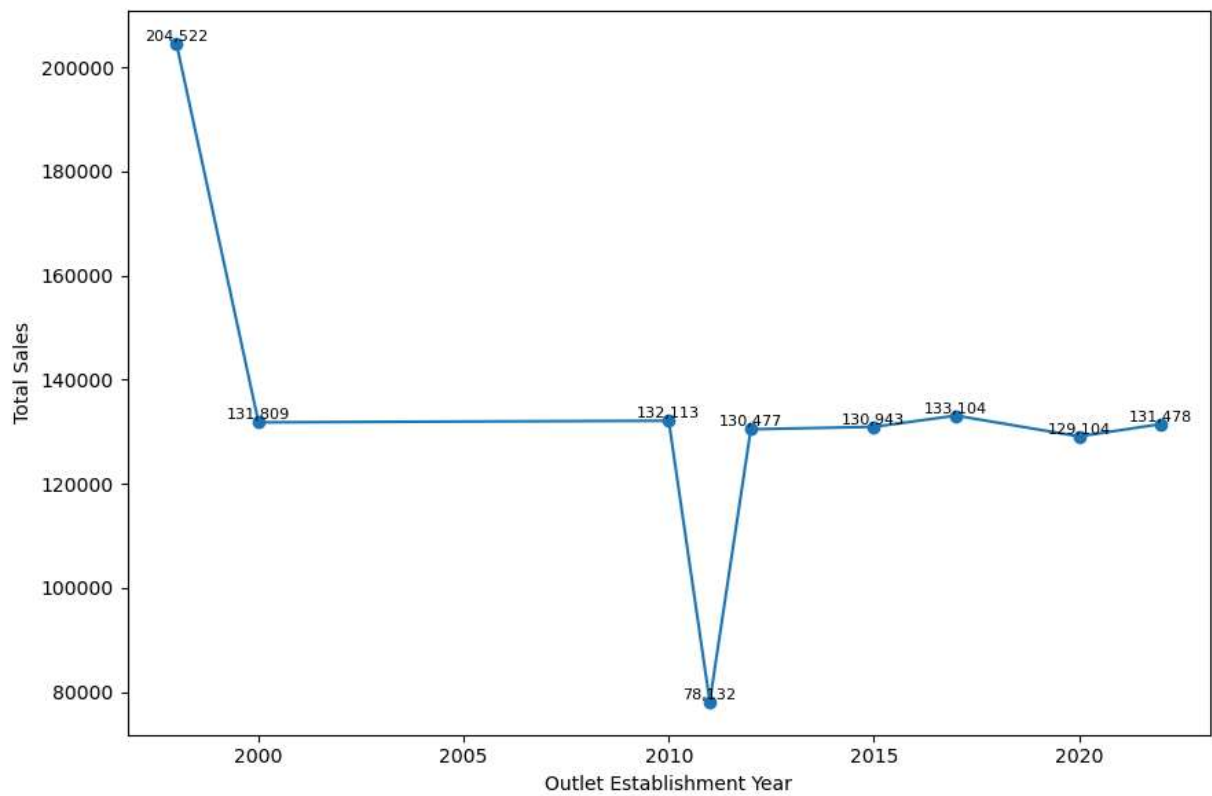
Total Sales by Outlet Establishment

```
In [78]: sales_by_year= df.groupby(['Outlet Establishment Year'])['Sales'].sum().sort_index()

plt.figure(figsize=(9, 6))
plt.plot(sales_by_year.index, sales_by_year.values, marker = 'o', linestyle = '-')
plt.xlabel('Outlet Establishment Year')
plt.ylabel('Total Sales')

for x,y in zip(sales_by_year.index, sales_by_year.values):
    plt.text(x, y, f'{y:,.0f}', ha='center', va= 'bottom', fontsize = 8)

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



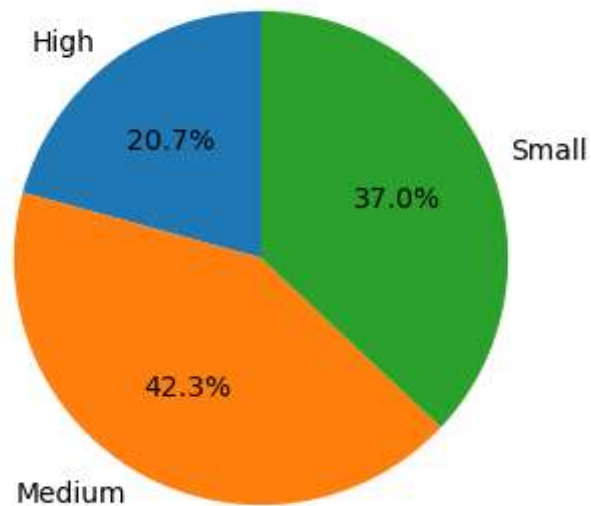
Total Sales by Outlet Size

```
In [86]: 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 = '%.1f%',
        startangle = 90)

plt.title('Sales by Outlet Size')
plt.tight_layout
plt.show()
```


Sales by Outlet Size

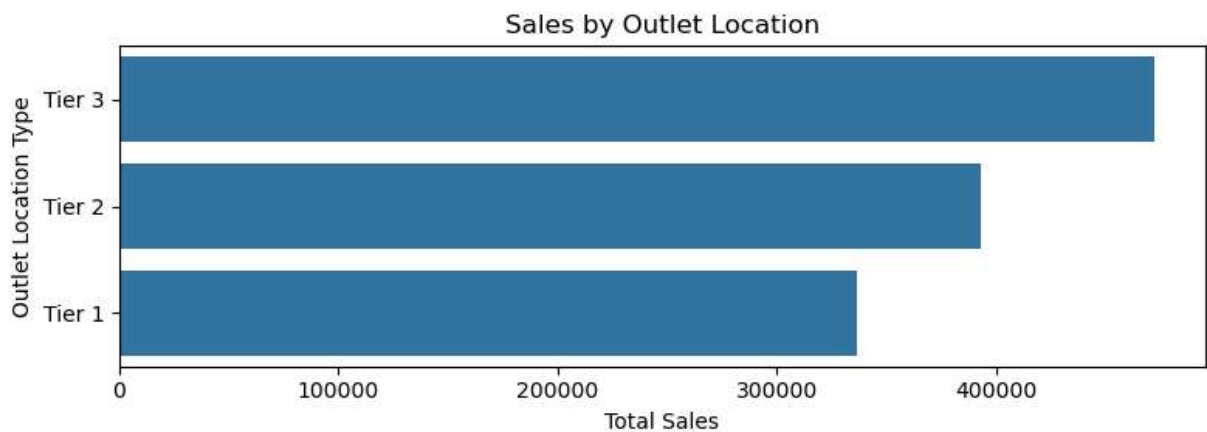


Total Sales by Outlet Location

```
In [90]: 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(x='Sales', y='Outlet Location Type', data=sales_by_location)

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



In []: