

Data Analysis Python Project - Blinkit Analysis

Import Libraries

```
In [1]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

Import Raw Data


```
In [2]: df = pd.read_csv("C:/Users/ankit/Downloads/blinkit_data.csv")
```

Sample of data top 5 Rows

```
In [3]: df.head(5)
```

Out[3]:

	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




Sample of data bottom 5 Rows

```
In [4]: df.tail(5)
```

Out[4]:

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



Size of Data

```
In [5]: print("Size of the Data =", df.shape)
```

Size of the Data = (8523, 12)

Field Info

```
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')
```

Data Types

```
In [7]: df.dtypes
```

```
Out[7]: 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 [8]: print(df['Item Fat Content'].unique())
```

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

```
In [9]: df['Item Fat Content'] = df['Item Fat Content'].replace({'low fat' : 'Low Fat',
                                                                'LF' : 'Low Fat',
                                                                'reg' : 'Regular'})
```

```
In [10]: print(df['Item Fat Content'].unique())
```

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

BUSINESS REQUIREMENTS

KPI's REQUIREMENTS

```
In [11]: #Total sales
Total_sales = df['Sales'].sum()

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

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

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

#Display

print(f"Total sales: ${Total_sales:,.0f}")
print(f"Avg Sales: ${Avg_sales:,.0f}")
print(f"No of Items Sold: {No_of_items_sold:,.0f}")
print(f"Avg Rating: {Avg_rating:,.1f}")
```

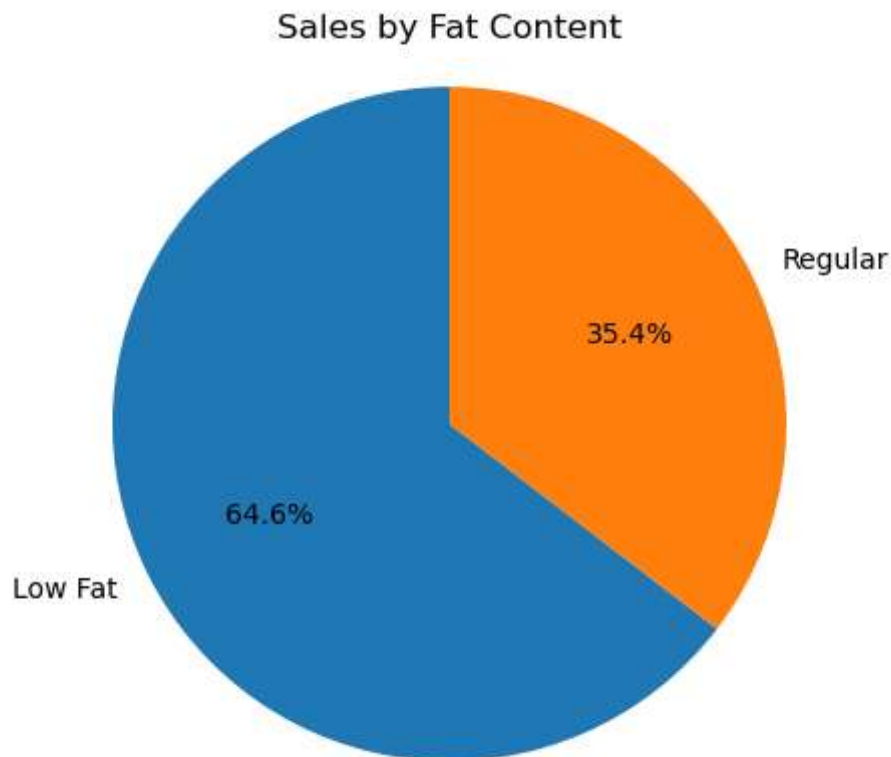
Total sales: \$1,201,681
Avg Sales: \$141
No of Items Sold: 8,523
Avg Rating: 4.0

CHARTS REQUIREMENTS

Total Sales By Fat Content

```
In [12]: 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 [13]: sales_by_type = df.groupby('Item Type')['Sales'].sum().sort_values(ascending=False)

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

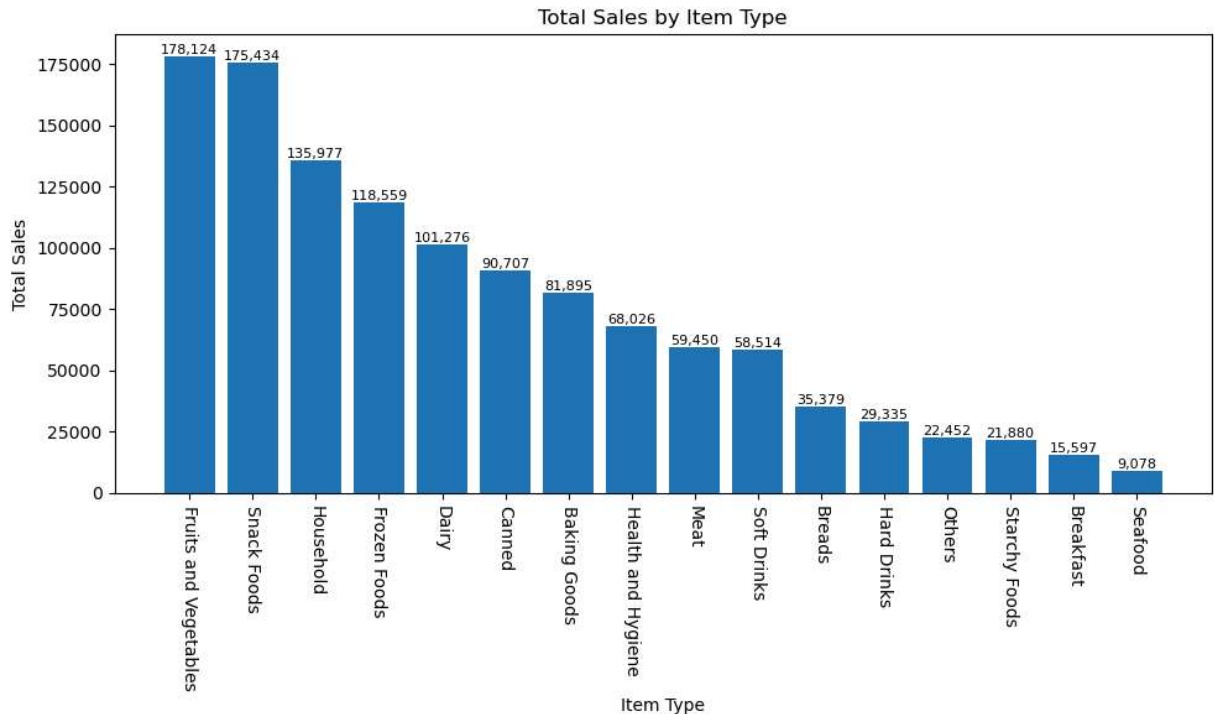
plt.xticks(rotation=-90)
plt.xlabel('Item Type')
plt.ylabel('Total Sales')
plt.title('Total Sales by Item Type')
```

```

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()

```



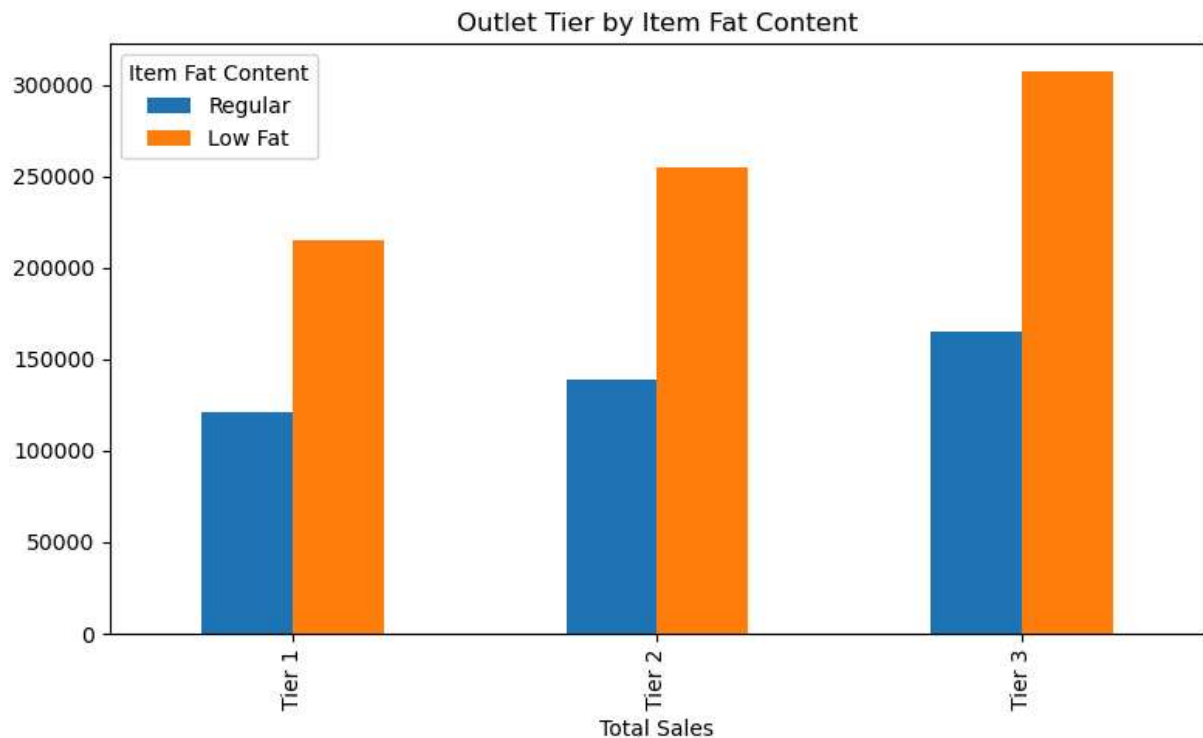
Fat Content by Outlet For Total Sales

```

In [14]: grouped = df.groupby(['Outlet Location Type', 'Item Fat Content'])['Sales'].sum().u
grouped = grouped[['Regular', 'Low Fat']]

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

```



Total Sales By Outlet Establishment

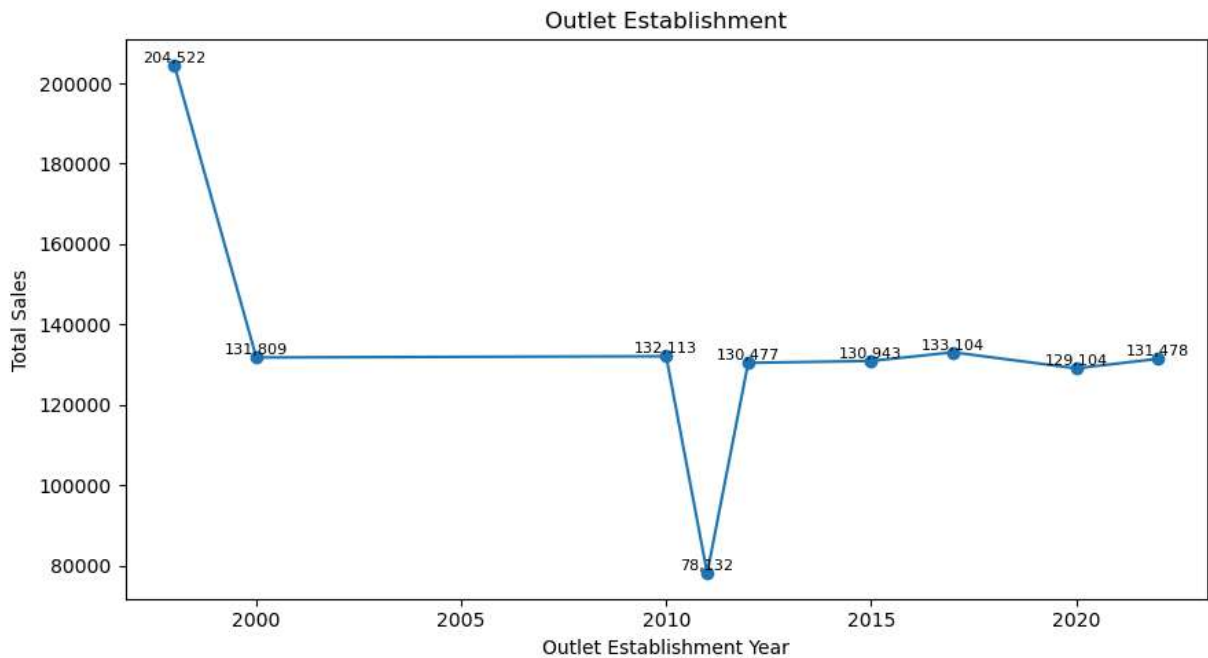
```
In [19]: sales_by_year = df.groupby('Outlet Establishment Year')['Sales'].sum().sort_index()

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')

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)

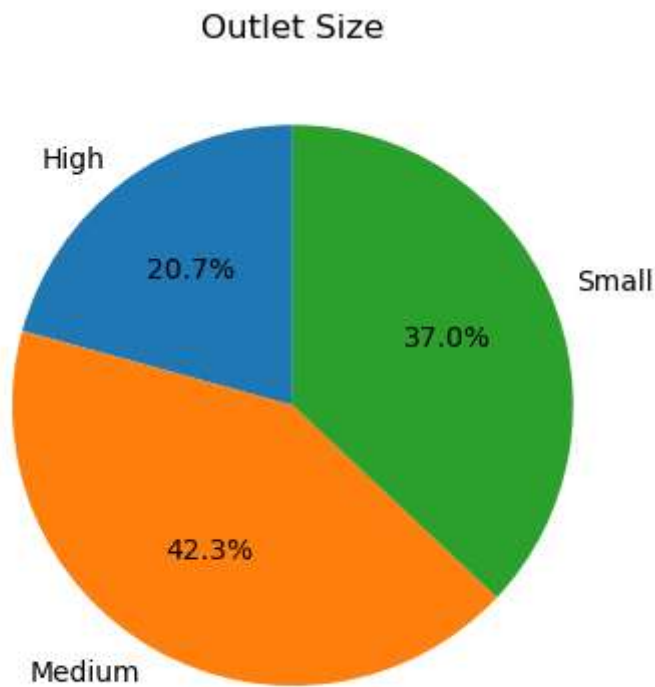
plt.tight_layout()
plt.show()
```



Sales By Outlet Size

```
In [21]: 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.tight_layout()
plt.show()
```



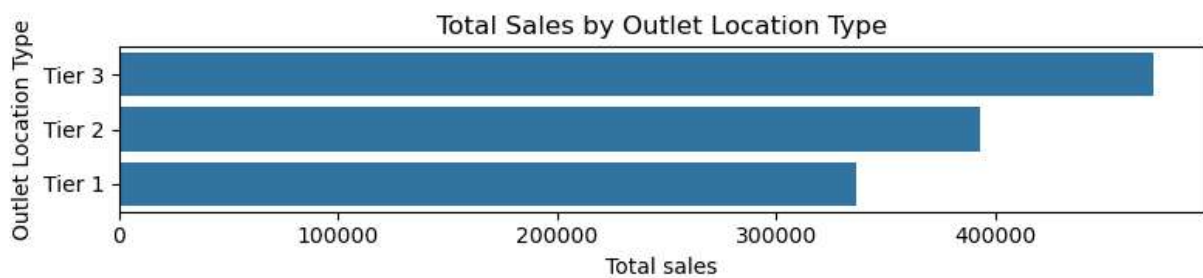
Sales By Outlet Location

```
In [23]: 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, 2)) #Smaller height, enough width
ax = 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.tight_layout() #Ensures Layput fits without scroll
plt.show()
```



In []: