

DATA ANALYSIS PYTHON PROJECT

Import Libraries

In [21]:

```
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/ABHISHEK/Documents/jupyter/food_delivery_data.csv")
```

Sample Data

In [11]:

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

Out [11]:

	Item Fat Content	Item Identifier	Item Type	Outlet Establishment Year	Outlet Identifier	Outlet Location Type
0	Regular	FDX32	Fruits and Vegetables	2012	OUT049	Tier 1
1	Low Fat	NCB42	Health and Hygiene	2022	OUT018	Tier 1
2	Regular	FDR28	Frozen Foods	2016	OUT046	Tier 1
3	Regular	FDL50	Canned	2014	OUT013	Tier 1
4	Low Fat	DRI25	Soft Drinks	2015	OUT045	Tier 1
5	low fat	FDS52	Frozen Foods	2020	OUT017	Tier 1
6	Low Fat	NCU05	Health and Hygiene	2011	OUT010	Tier 1
7	Low Fat	NCD30	Household	2015	OUT045	Tier 1
8	Low Fat	FDW20	Fruits and Vegetables	2014	OUT013	Tier 1
9	Low Fat	FDX25	Canned	2018	OUT027	Tier 1

In [12]:

```
df.tail(10)
```

Out [12]:

	Item Fat Content	Item Identifier	Item Type	Outlet Establishment Year	Outlet Identifier	Outlet Location Type
8513	Regular	DRY23	Soft Drinks	2018	OUT027	Tier 1
8514	low fat	FDA11	Baking Goods	2018	OUT027	Tier 1
8515	low fat	FDK38	Canned	2018	OUT027	Tier 1
8516	low fat	FDO38	Canned	2018	OUT027	Tier 1
8517	low fat	FDG32	Fruits and Vegetables	2018	OUT027	Tier 1
8518	low fat	NCT53	Health and Hygiene	2018	OUT027	Tier 1
8519	low fat	FDN09	Snack Foods	2018	OUT027	Tier 1
8520	low fat	DRE13	Soft Drinks	2018	OUT027	Tier 1
8521	reg	FDT50	Dairy	2018	OUT027	Tier 1
8522	reg	FDM58	Snack Foods	2018	OUT027	Tier 1

Size of Data

In [13]:

```
print("Size of Data:",df.shape)
```

Size of Data: (8523, 12)

Field Info

In [14]:

```
df.columns
```

Out [14]:

```
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 [15]:

```
df.dtypes
```

Out [15]:

```
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 [18]:

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

In [19]:

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

['Regular' 'Low Fat']

Business Requirments

KPI's Requirments

In [30]:

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

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

#Number 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"Average sales: ${avg_sales:,.1f}")
print(f"No of Items sold: {no_of_items_sold:,.0f}")
print(f"Average Ratings: {avg_rating:,.1f}")
```

Total sales: \$1,201,681

Average sales: \$141.0

No of Items sold: 8,523

Average Ratings: 4.0

CHARTS REQUIRMENTS

Total Sale by Fat Content

In [22]:

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

plt.pie(Sales_by_fat, labels = Sales_by_fat.index,
        autopct = '%1.0f%%',
        startangle = 90)
plt.title('Sales by fat Content')
plt.axis('equal')
plt.show()
```



Total Sale by Fat Item Type

In [23]:

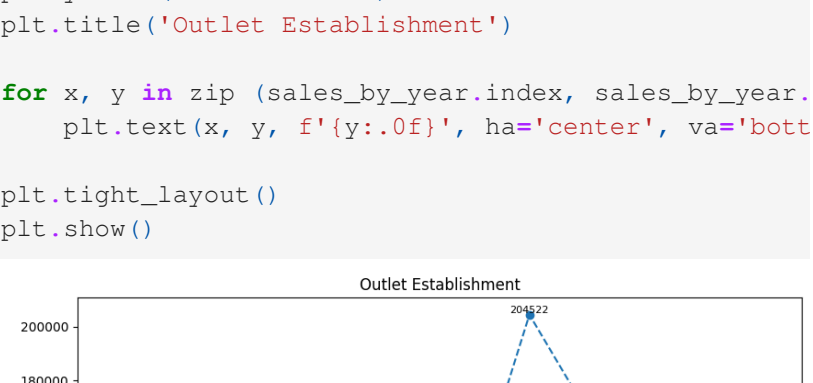
```
sales_by_type = df.groupby('Item Type')['Sales'].sum()

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()
```

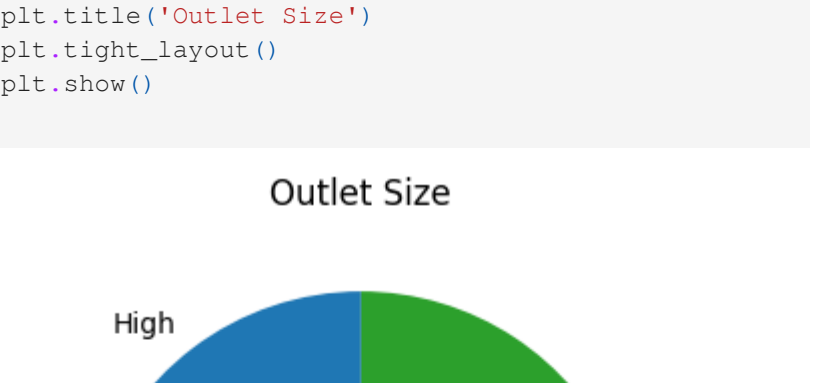


Fat Content by Outlet for Total Sales

In [24]:

```
grouped=df.groupby(['Outlet Location Type', 'Item Fat Content'])
grouped=grouped[['Regular', 'Low Fat']]

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



Total Sales by Outlet Establishment

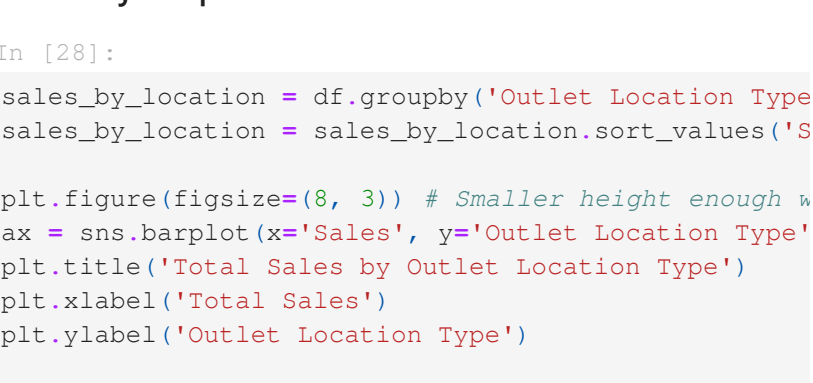
In [25]:

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

plt.figure(figsize=(9,5))
plt.plot(sales_by_year.index,sales_by_year.values, marker='o')
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')

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



Sales by Outlet Size

In [27]:

```
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='%1.0f%%')
plt.title('Outlet Size')
plt.tight_layout()
plt.show()
```


Sales by Output Location

In [28]:

```
sales_by_location = df.groupby('Outlet Location Type')['Sales'].sum()
sales_by_location = sales_by_location.sort_values('Sales', ascending=False)
```

```
plt.figure(figsize=(8, 3)) # Smaller height enough w
ax = sns.barplot(x='Sales', y='Outlet Location Type')
plt.title('Total Sales by Outlet Location Type')
plt.xlabel('Total Sales')
plt.ylabel('Outlet Location Type')
```

```
plt.tight_layout() # Ensure layout fits without scrollbars
plt.show()
```

