

DATA ANALYSIS PYTHON PROJECT - CITYKART

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 [7]: df=pd.read_csv("C:\\Users\\lokin\\OneDrive\\Desktop\\data analyst\\sq
```

Sample Data

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

	Item Fat Content	Item Identifier	Item Type	Outlet Establishment Year	Outlet Identifier	Outlet Location Type	Outlet Size	Outlet 'Type'
0	Regular	FDX32	Fruits and Vegetables	2012	OUT049	Tier 1	Medium	Superma T
1	Low Fat	NCB42	Health and Hygiene	2022	OUT018	Tier 3	Medium	Superma T
2	Regular	FDR28	Frozen Foods	2016	OUT046	Tier 1	Small	Superma T
3	Regular	FDL50	Canned	2014	OUT013	Tier 3	High	Superma T
4	Low Fat	DRI25	Soft Drinks	2015	OUT045	Tier 2	Small	Superma T
5	low fat	FDS52	Frozen Foods	2020	OUT017	Tier 2	Small	Superma T
6	Low Fat	NCU05	Health and Hygiene	2011	OUT010	Tier 3	Small	Grocery S
7	Low Fat	NCD30	Household	2015	OUT045	Tier 2	Small	Superma T
8	Low Fat	FDW20	Fruits and Vegetables	2014	OUT013	Tier 3	High	Superma T
9	Low Fat	FDX25	Canned	2018	OUT027	Tier 3	Medium	Superma T

```
In [9]: df.tail(10)
```

	Item Fat Content	Item Identifier	Item Type	Outlet Establishment Year	Outlet Identifier	Outlet Location Type	Outlet Size	Outl
8513	Regular	DRY23	Soft Drinks	2018	OUT027	Tier 3	Medium	Super
8514	low fat	FDA11	Baking Goods	2018	OUT027	Tier 3	Medium	Super
8515	low fat	FDK38	Canned	2018	OUT027	Tier 3	Medium	Super
8516	low fat	FDO38	Canned	2018	OUT027	Tier 3	Medium	Super
8517	low fat	FDG32	Fruits and Vegetables	2018	OUT027	Tier 3	Medium	Super
8518	low fat	NCT53	Health and Hygiene	2018	OUT027	Tier 3	Medium	Super
8519	low fat	FDN09	Snack Foods	2018	OUT027	Tier 3	Medium	Super
8520	low fat	DRE13	Soft Drinks	2018	OUT027	Tier 3	Medium	Super
8521	reg	FDT50	Dairy	2018	OUT027	Tier 3	Medium	Super
8522	reg	FDM58	Snack Foods	2018	OUT027	Tier 3	Medium	Super

Size of Data

```
In [10]: print('size of data:',df.shape)
```

size of data: (8523, 12)

Field info

```
In [11]: df.columns
```

```
Out[11]: Index(['Item Fat Content', 'Item Identifier', 'Item Type',
              'Outlet Establishment Year', 'Outlet Identifier',
              'Outlet Location Type', 'Outlet Size', 'Outlet Type', 'Item Vi-
              sibility',
              'Item Weight', 'Sales', 'Rating'],
              dtype='object')
```

data type

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

```
Out[12]: 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 [13]: print(df['Item Fat Content'].unique())
```

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

```
In [14]: df['Item Fat Content'] = df['Item Fat Content'].replace({'LF':'low fat',
                        'low fat':'Lo
                        'reg':'Regula
```

```
In [15]: print(df['Item Fat Content'].unique())
```

['Regular' 'Low Fat' 'low Fat']

Bussiness Requirements

KPI's Requirements

```
In [22]: #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_ratings = df['Rating'].mean()
```

```
#display
print(f'total sales:${total_sales:,0f}')
print(f'average sales:${avg_sales:,.1f}')
print(f'no of items sales:(no_of_items_sold:,0f)')
print(f'average ratings: {avg_ratings:,.1f}')
```

total sales:\$1,201,681  
average sales:\$141.0  
no of items sales:8,523  
average ratings: 4.0

CHARTS REQUIREMENTS

Total Sales by Fat content

```
In [30]: sales_by_fat = df.groupby('Item Fat Content')['Sales'].sum()

plt.pie(sales_by_fat, labels = sales_by_fat.index,
        autopct = '%.0f%%',
        startangle = 90)
```

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



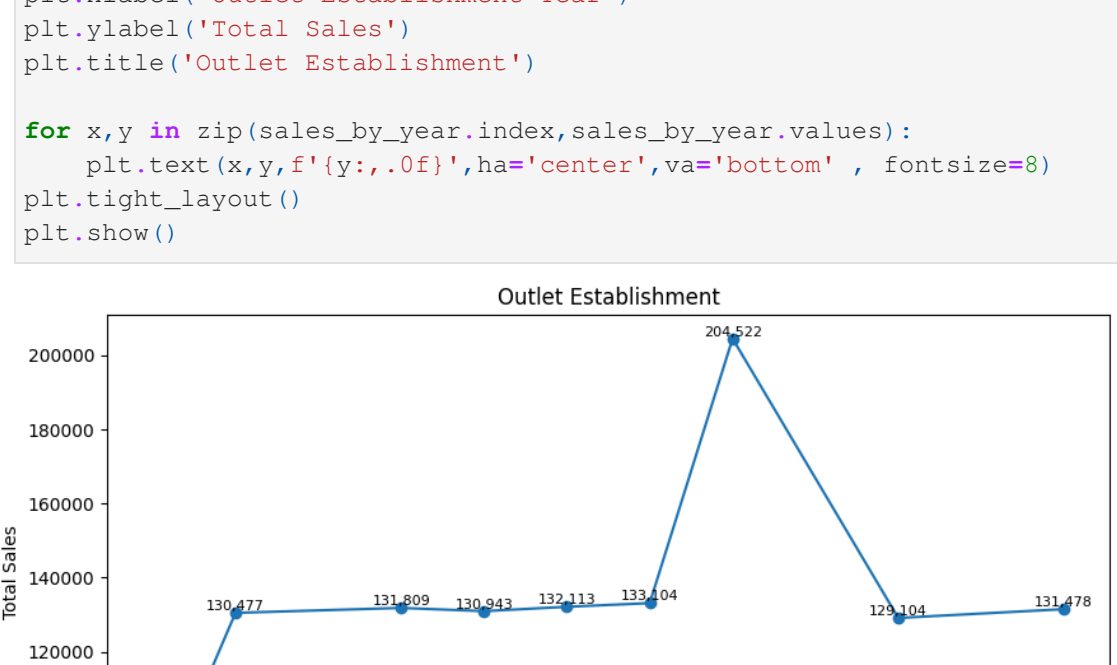
Total Sales by item type

```
In [32]: sales_by_type = df.groupby('Item Type')['Sales'].sum().sort_values(asc
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 Items 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', font
```

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

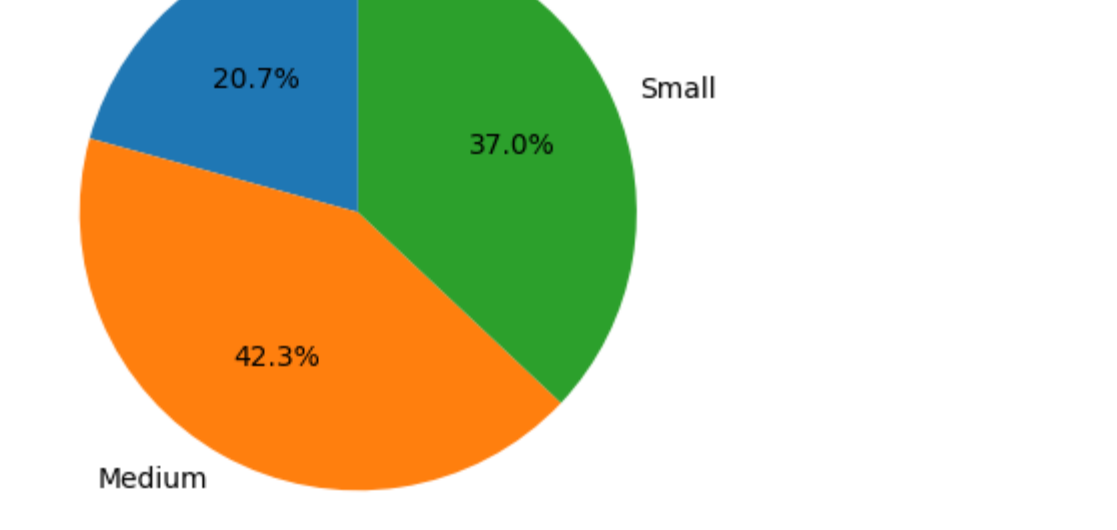


Fat Content by Outlet for Total sales

```
In [36]: grouped = df.groupby(['Outlet Location Type','Item Fat Content'])['Sal
grouped = grouped[['Regular','Low Fat']]
```

```
ax = grouped.plot(kind='bar',figsize=(8,5),title='Outlet Tier by Item
plt.xlabel('Outlet Location Type')
```

```
plt.ylabel('Total Sales')
plt.legend(title = 'Item Fat Content')
plt.tight_layout()
plt.show()
```

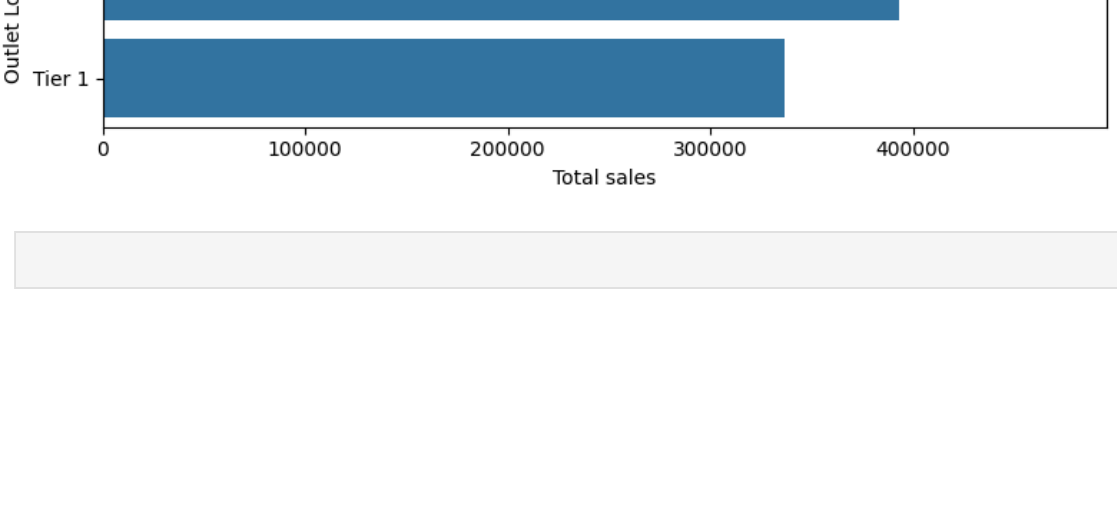


Total sales by outlet establishment

```
In [42]: 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',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 [47]: 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%%', s
```

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



Sales by outlet location

```
In [51]: 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))
ax=sns.barplot(x='Sales',y='Outlet Location Type', data=sales_by_loca
```

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

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



```
In [ ]:
```