

Blinkit Analysis using python(EDA) by Guruprasad

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

df = pd.read_csv("blinkit_data.csv")

df.head(10)
```

	Item Fat	Content	Item Identifier	Item Type
0		Regular	FDX32	Fruits and Vegetables
1		Low Fat	NCB42	Health and Hygiene
2		Regular	FDR28	Frozen Foods
3		Regular	FDL50	Canned
4		Low Fat	DRI25	Soft Drinks
5		low fat	FDS52	Frozen Foods
6		Low Fat	NCU05	Health and Hygiene
7		Low Fat	NCD30	Household
8		Low Fat	FDW20	Fruits and Vegetables
9		Low Fat	FDX25	Canned

	Outlet Establishment	Year	Outlet Identifier	Outlet Location	Type
0		2012	OUT049		Tier 1
1		2022	OUT018		Tier 3
2		2010	OUT046		Tier 1
3		2000	OUT013		Tier 3
4		2015	OUT045		Tier 2
5		2020	OUT017		Tier 2
6		2011	OUT010		Tier 3
7		2015	OUT045		Tier 2
8		2000	OUT013		Tier 3
9		1998	OUT027		Tier 3

	Outlet Size	Outlet Type	Item Visibility	Item Weight
Sales \ 145.4786				
0	Medium	Supermarket	Type1	0.100014 15.10
1	Medium	Supermarket	Type2	0.008596 11.80
115.3492				
2	Small	Supermarket	Type1	0.025896 13.85
165.0210				
3	High	Supermarket	Type1	0.042278 12.15
126.5046				
4	Small	Supermarket	Type1	0.033970 19.60
55.1614				
5	Small	Supermarket	Type1	0.005505 8.89
102.4016				
6	Small	Grocery Store		0.098312 11.80

```

81.4618
7      Small  Supermarket  Type1      0.026904      19.70
96.0726
8      High   Supermarket  Type1      0.024129      20.75
124.1730
9      Medium Supermarket  Type3      0.101562      NaN
181.9292

```

```

      Rating
0      5.0
1      5.0
2      5.0
3      5.0
4      5.0
5      5.0
6      5.0
7      5.0
8      5.0
9      5.0

```

```
df.shape
```

```
(8523, 12)
```

```
df.dtypes
```

```

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

```

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

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

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

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

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

Business Requierments

KPI's Requirements

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

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

# No. of items Sold
no_item_sold = df['Sales'].count()

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

print(f"Total Sales: ${total_sales:,.0f}")
print(f"Average Sales: ${avg_sales:,.0f}")
print(f"Number Of items sold: {no_item_sold:,.0f}")
print(f"Average Rating: {avg_ratings:,.0f}")

Total Sales: $1,201,681
Average Sales: $141
Number Of items sold: 8,523
Average Rating: 4
```

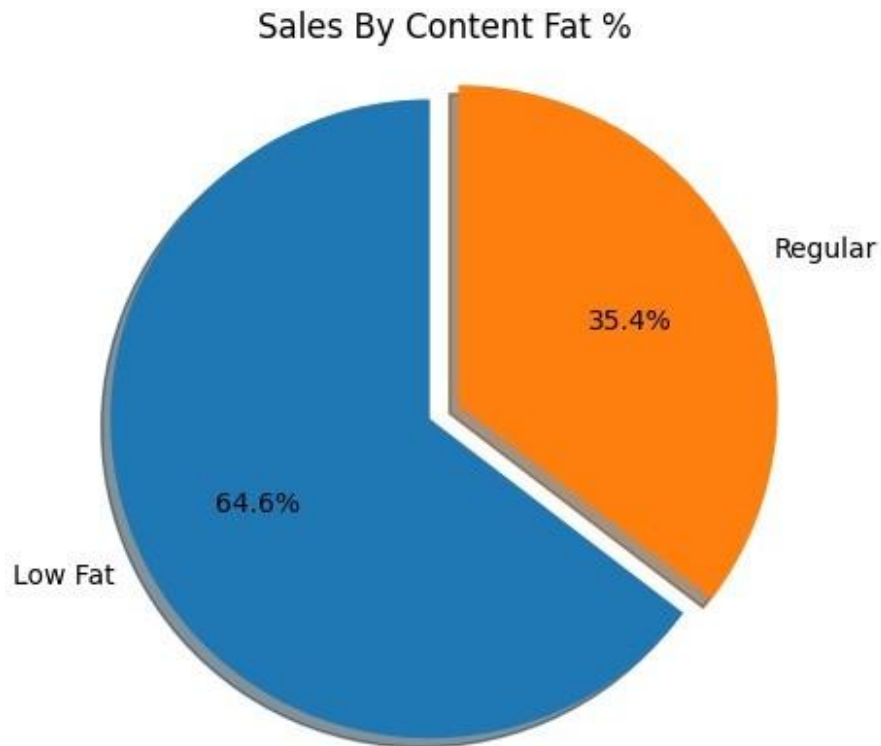
Chart Requirements

Total Sales by Fat Content

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

plt.pie(sales_by_fat, labels = sales_by_fat.index,
        autopct = '%.1f%%',
        shadow = True,
        explode=[0.05,0.05],
        startangle = 90)

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



Total Sales by Item Type

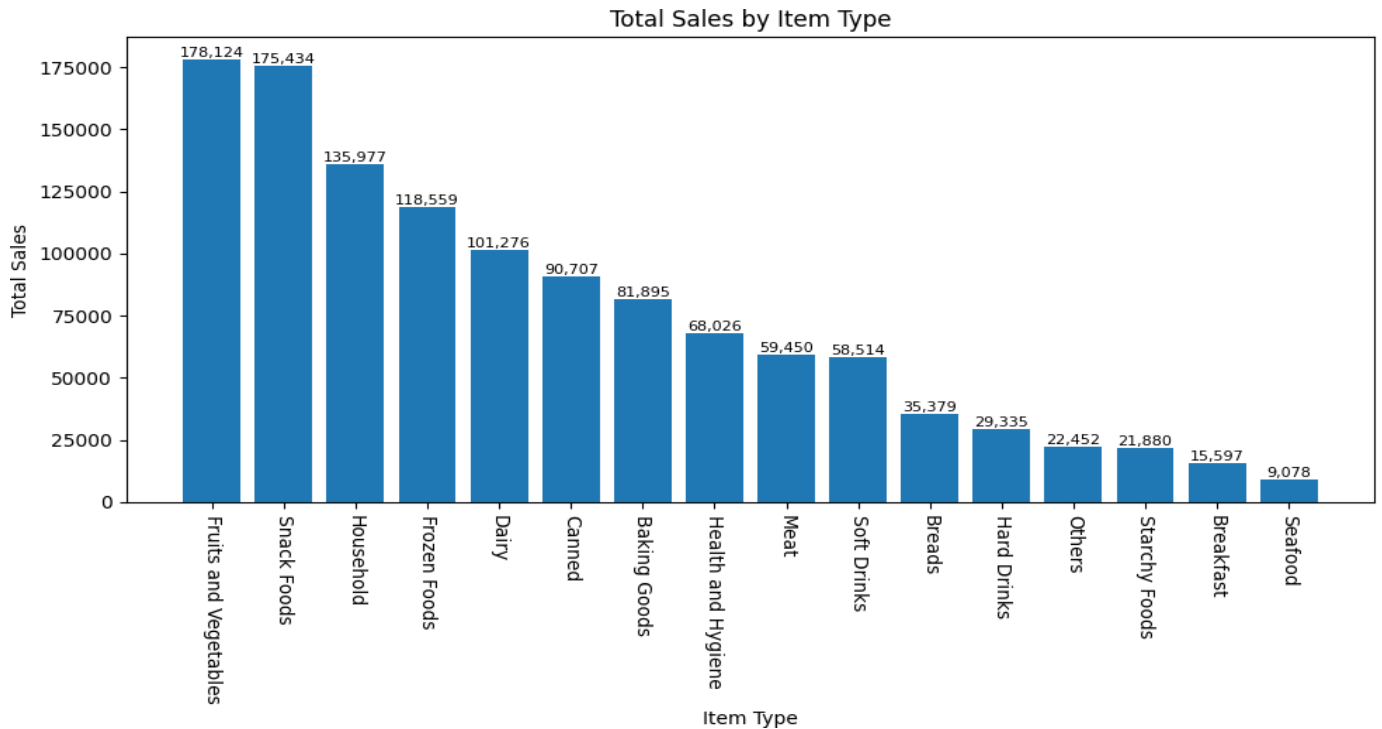
```
sales_by_type = df.groupby('Item Type')
['Sales'].sum().sort_values(ascending=False)

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()
```

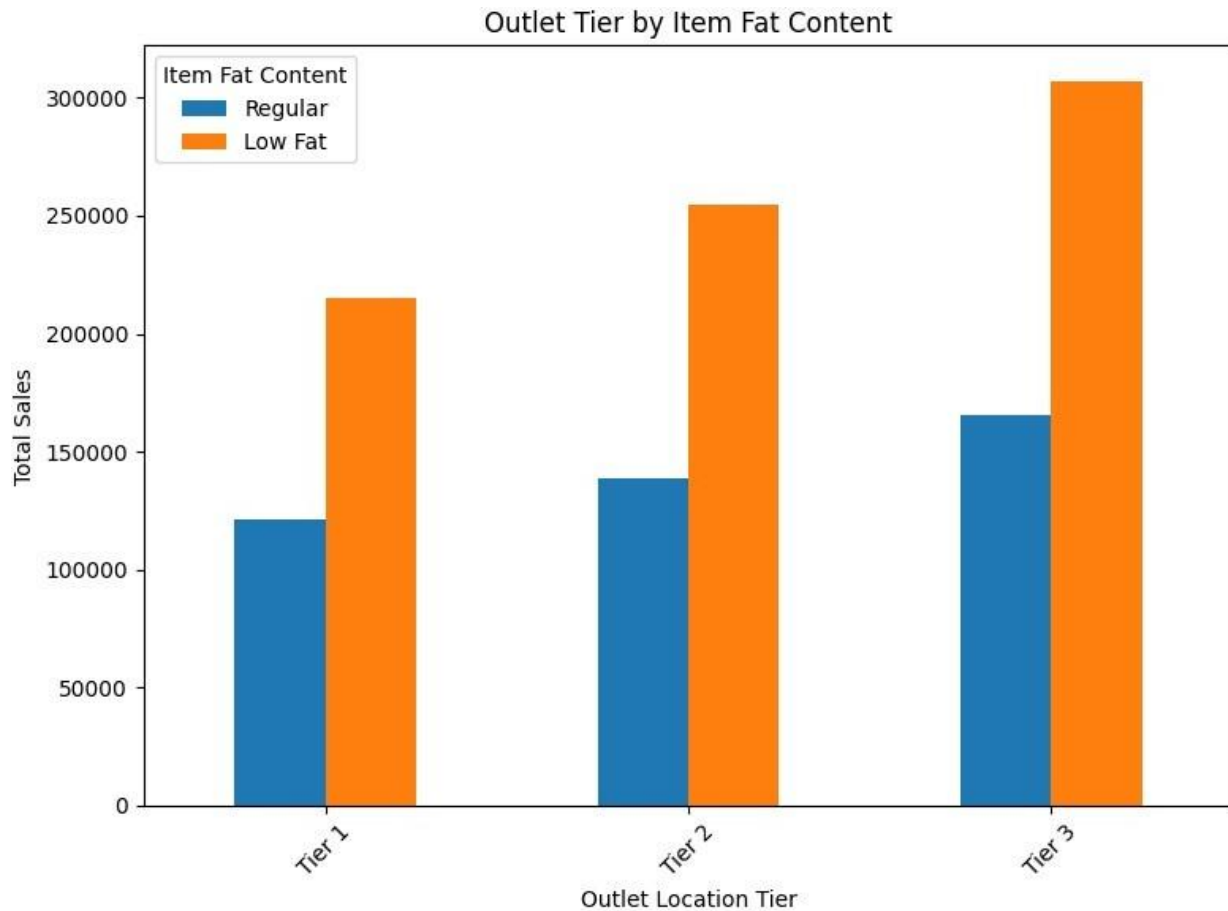


Fat Content by Outlet for Sales

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

ax = grouped.plot(kind='bar', figsize=(8, 6), title='Outlet Tier by Item
Fat Content')
plt.xticks(rotation=45)
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

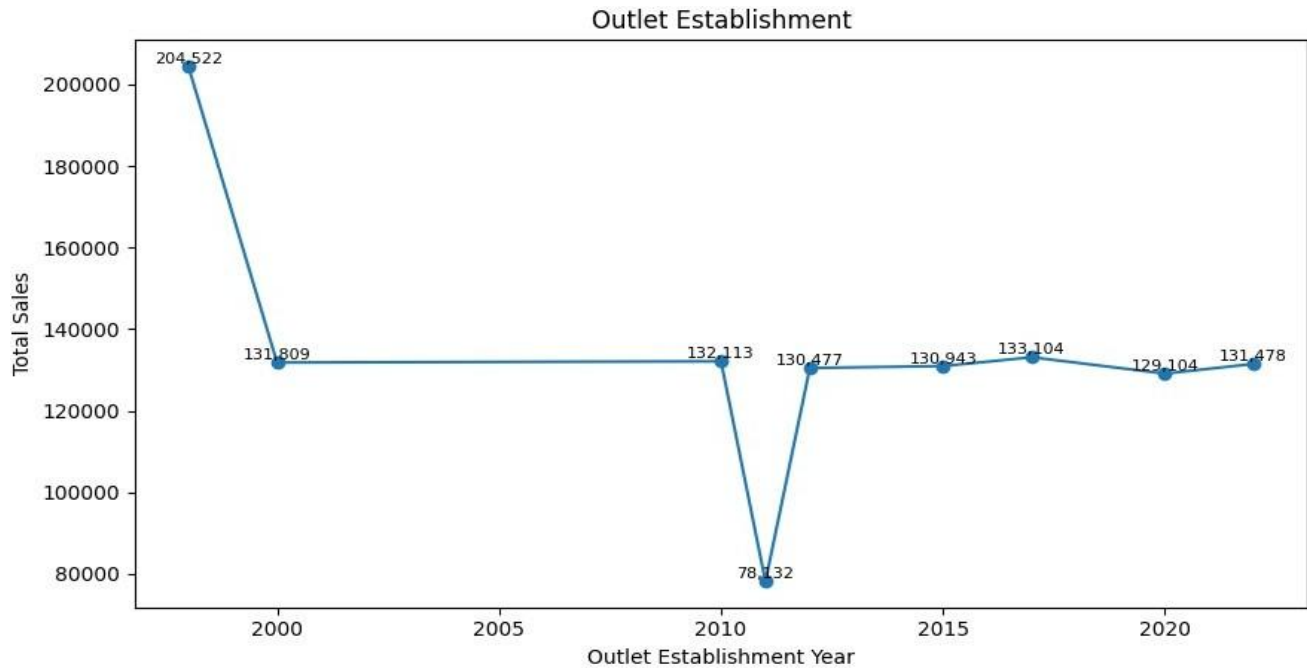
```
sales_by_year = df.groupby('Outlet Establishment Year')
['Sales'].sum().sort_index()

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)

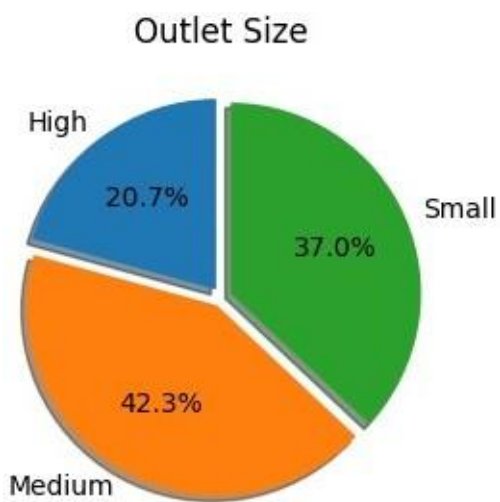
plt.tight_layout()
plt.show()
```



Sales by Outlet Size

```
sales_by_size = df.groupby('Outlet Size')['Sales'].sum()

plt.figure(figsize=(3,3))
plt.pie(sales_by_size, labels=sales_by_size.index, autopct='%1.1f%%', shadow=True, startangle=90, explode=[0.05, 0.05, 0.05])
plt.title('Outlet Size')
plt.tight_layout()
plt.show()
```



Sales by Outlet Location

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

plt.figure(figsize=(8,3))
ax = sns.barplot(x='Sales', y='Outlet Location Type', data = location)

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

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

