Blinkit_data_analysis

May 27, 2025

1 DATA ANALYSIS PYTHON PROJECT -BLINKIT ANALYSIS

1.1 Import Libraries

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

1.2 Import Raw Data

[2]:
```
df = pd.read_csv("BlinkIT Grocery Data.csv")
df
```

[2]:

	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 3
2	Regular	FDR28	Frozen Foods	2016	OUT046	Tier 1
3	Regular	FDL50	Canned	2014	OUT013	Tier 3
4	Low Fat	DRI25	Soft Drinks	2015	OUT045	Tier 2
…	…	…	…	…	…	…
8518	low fat	NCT53	Health and Hygiene	2018	OUT027	Tier 3
8519	low fat	FDN09	Snack Foods	2018	OUT027	Tier 3
8520	low fat	DRE13	Soft Drinks			
8521	reg	FDT50	Dairy			
8522	reg	FDM58	Snack Foods			

```
8520                    2018           OUT027               Tier 3
8521                    2018           OUT027               Tier 3
8522                    2018           OUT027               Tier 3

     Outlet Size        Outlet Type  Item Visibility  Item Weight     Sales  \
0         Medium  Supermarket Type1         0.100014        15.10  145.4786
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
…              …                  …                …            …        …
8518      Medium  Supermarket Type3         0.000000          NaN  164.5526
8519      Medium  Supermarket Type3         0.034706          NaN  241.6828
8520      Medium  Supermarket Type3         0.027571          NaN   86.6198
8521      Medium  Supermarket Type3         0.107715          NaN   97.8752
8522      Medium  Supermarket Type3         0.000000          NaN  112.2544

      Rating
0        5.0
1        5.0
2        5.0
3        5.0
4        5.0
…        …
8518     4.0
8519     4.0
8520     4.0
8521     4.0
8522     4.0

[8523 rows x 12 columns]
```

1.3 Sample Data

[3]:
```
df.head(10)
```

[3]:
```
  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                       2016            OUT046               Tier 1
3                       2014            OUT013               Tier 3
4                       2015            OUT045               Tier 2
5                       2020            OUT017               Tier 2
6                       2011            OUT010               Tier 3
7                       2015            OUT045               Tier 2
8                       2014            OUT013               Tier 3
9                       2018            OUT027               Tier 3

  Outlet Size        Outlet Type  Item Visibility  Item Weight     Sales  \
0      Medium  Supermarket Type1         0.100014        15.10  145.4786
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
```

[4]:
```
df.tail(10)
```

[4]:
```
     Item Fat Content Item Identifier              Item Type  \
8513          Regular           DRY23            Soft Drinks
8514          low fat           FDA11           Baking Goods
8515          low fat           FDK38                 Canned
8516          low fat           FDO38                 Canned
8517          low fat           FDG32  Fruits and Vegetables
8518          low fat           NCT53     Health and Hygiene
8519          low fat           FDN09             Snack Foods
```

```
8520          low fat           DRE13            Soft Drinks
8521              reg           FDT50                  Dairy
8522              reg           FDM58            Snack Foods

      Outlet Establishment Year Outlet Identifier Outlet Location Type  \
8513                       2018            OUT027                Tier 3
8514                       2018            OUT027                Tier 3
8515                       2018            OUT027                Tier 3
8516                       2018            OUT027                Tier 3
8517                       2018            OUT027                Tier 3
8518                       2018            OUT027                Tier 3
8519                       2018            OUT027                Tier 3
8520                       2018            OUT027                Tier 3
8521                       2018            OUT027                Tier 3
8522                       2018            OUT027                Tier 3

     Outlet Size        Outlet Type  Item Visibility  Item Weight     Sales  \
8513      Medium  Supermarket Type3         0.108568          NaN   42.9112
8514      Medium  Supermarket Type3         0.043029          NaN   94.7436
8515      Medium  Supermarket Type3         0.053032          NaN  149.1734
8516      Medium  Supermarket Type3         0.072486          NaN   78.9986
8517      Medium  Supermarket Type3         0.175143          NaN  222.3772
8518      Medium  Supermarket Type3         0.000000          NaN  164.5526
8519      Medium  Supermarket Type3         0.034706          NaN  241.6828
8520      Medium  Supermarket Type3         0.027571          NaN   86.6198
8521      Medium  Supermarket Type3         0.107715          NaN   97.8752
8522      Medium  Supermarket Type3         0.000000          NaN  112.2544

      Rating
8513     4.0
8514     4.0
8515     4.0
8516     4.0
8517     4.0
8518     4.0
8519     4.0
8520     4.0
8521     4.0
8522     4.0
```

1.4 Size of Data

[5]:
```
print("Size of data; ",df.shape)
```

```
Size of data;  (8523, 12)
```

1.5 Field Information

[6]:
```
df.columns
```

[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')
```

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

[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
```

1.6 Data Cleaning

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

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

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

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

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

2 BUSINESS REQUIREMENTS

2.1 KPI's REQUIREMENTS

[11]:
```
# Total Sales
Total_Sales = df['Sales'].sum()
# Average Sales
Average_Sales = df['Sales'].mean()
# No of items
```

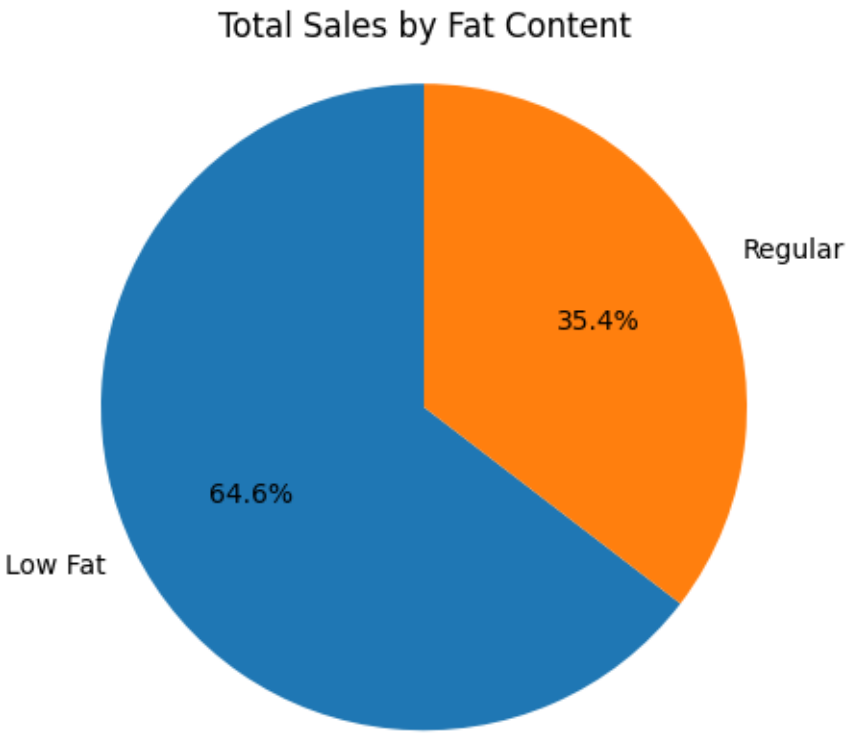
```
No_of_solds =df['Sales'].count()
#Average Ratings
avg_ratings = df['Rating'].mean()



print(f"Total Sales: $ {Total_Sales:,.0f}")
print(f"Average Sales: ${Average_Sales:,.1f}")
print(f"No of Items solds: ${No_of_solds:,.0f}")
print(f"Average rating: ${avg_ratings:,.1f}")
```

```
Total Sales: $ 1,201,681
Average Sales: $141.0
No of Items solds: $8,523
Average rating: $4.0
```

3 CHARTS REQUIREMENTS

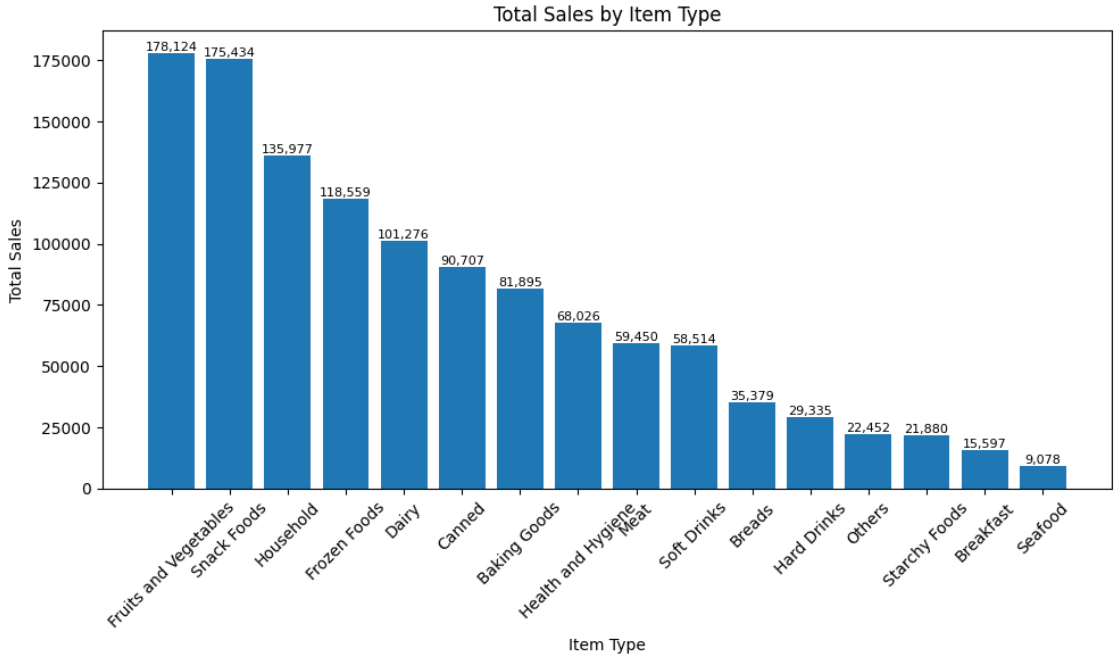
[12]:
```
### Total Sales by Fat Content
```

[13]:
```
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("Total Sales by Fat Content")
plt.axis('equal')
plt.show()
```



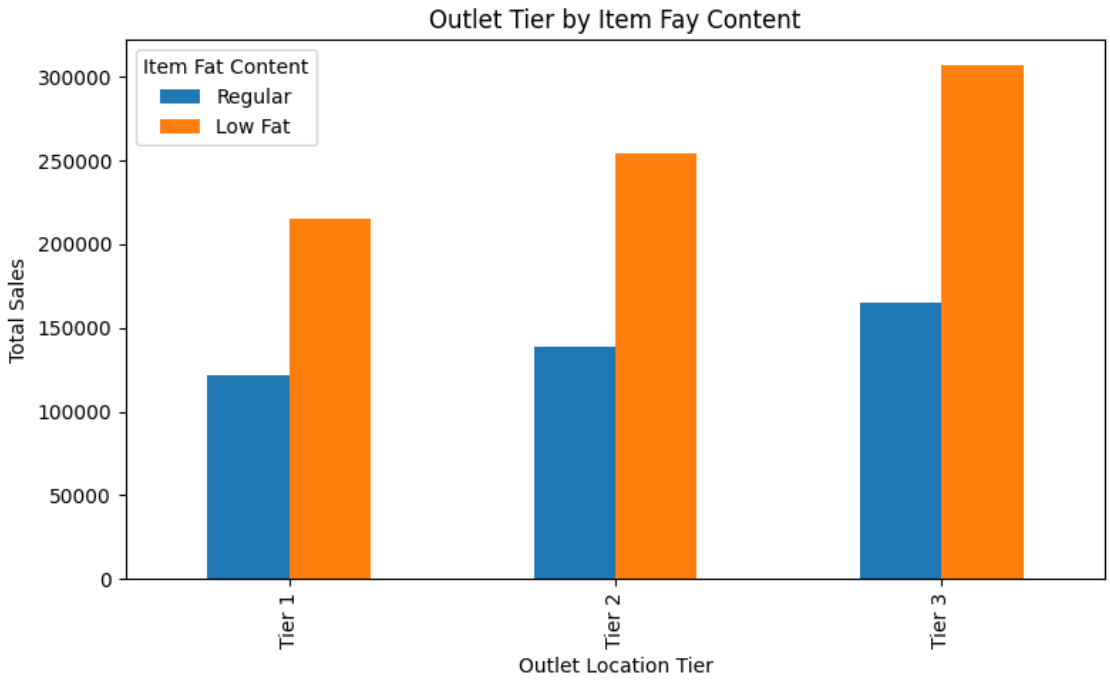
3.1 Total Sales by Item Type

[14]:
```
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=45)
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()
```



3.2 Fat Content by Outlet for Total Sales

[15]:
```
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 Fay␣
 ↪Content')
plt.xlabel('Outlet Location Tier')
plt.ylabel('Total Sales')
plt.legend(title='Item Fat Content')
plt.tight_layout()
plt.show()
```



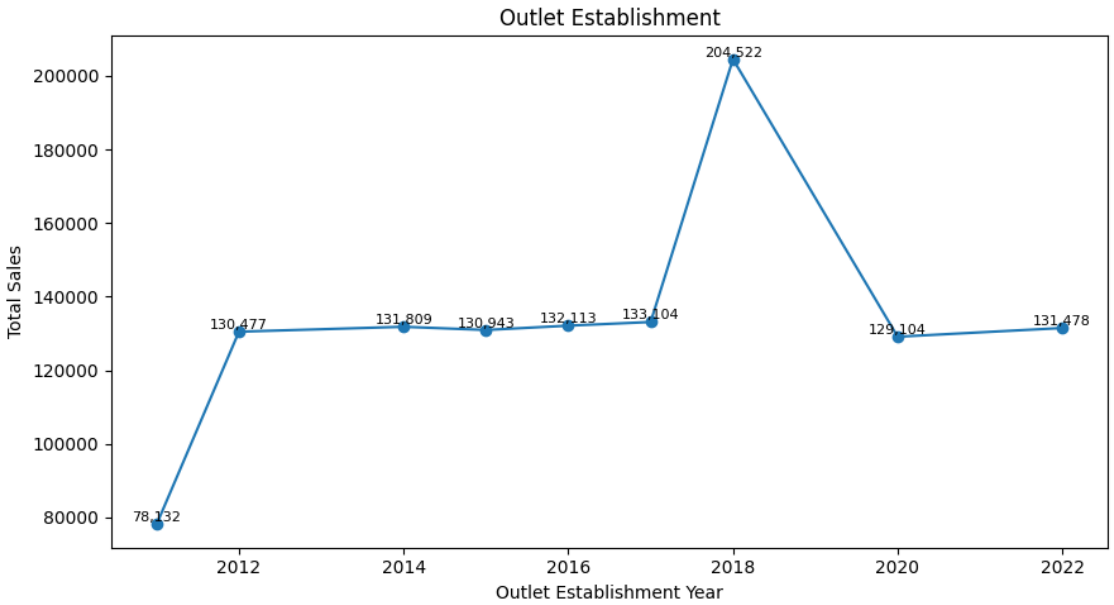
3.3 Total Sales by Outlet Establishment

[16]:
```
df.dtypes
```

[16]:
```
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
```

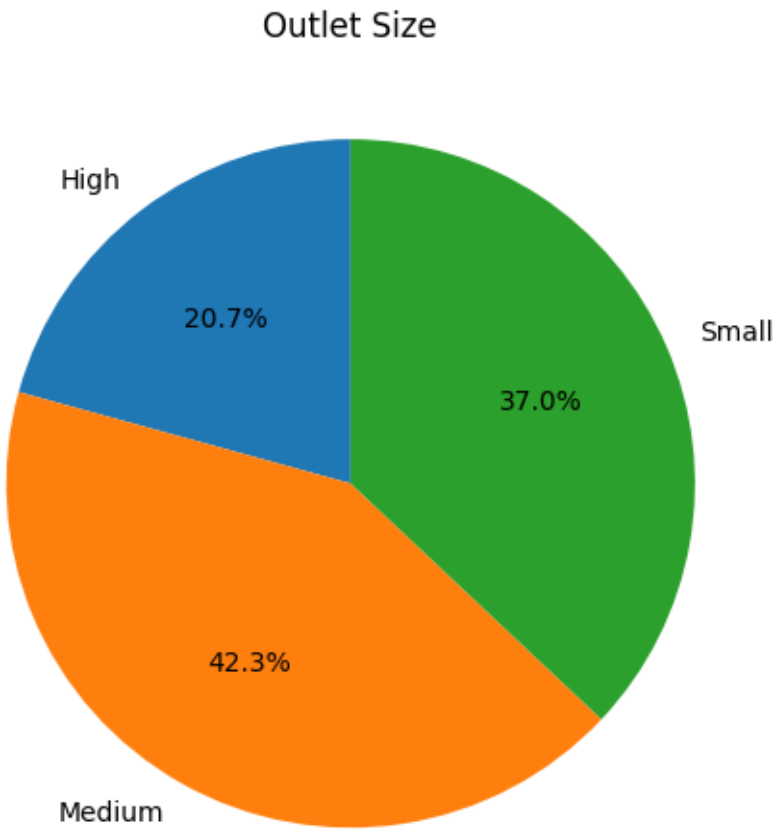
[17]:
```
sales_by_year = df.groupby('Outlet Establishment Year')['Sales'].sum().
 ↪sort_index()

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()
```



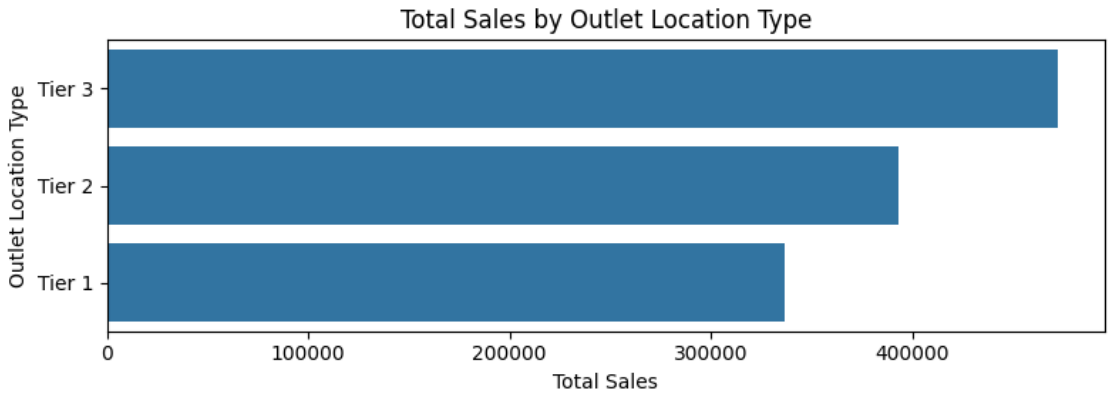
3.4 Sales by Outlet Size

[22]:
```
sales_by_size = df.groupby('Outlet Size')['Sales'].sum()

plt.figure(figsize=(5,5))
plt.pie(sales_by_size, labels=sales_by_size.index, autopct='%1.
  ↪1f%%',startangle=90)
plt.title('Outlet Size')
plt.tight_layout()
plt.show()
```



3.5 Sales by Outlet Location

[26]:
```
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('Total Sales by Outlet Location Type')
plt.xlabel('Total Sales')
plt.ylabel('Outlet Location Type')
plt.tight_layout()
plt.show()
```


Total Sales by Outlet Location Type
Outlet Location Type
Tier 3
Tier 2
Tier 1
0
100000
200000
300000
400000
Total Sales

[ ]: