Blinkit :

India’s Last Mintue app

Project Overview:

Objective:

To analyze Blinkit's item sales performance across different outlet types, sizes, and geographic tiers in order to uncover insights that can help optimize product placement, outlet operations, and customer targeting strategies.

Business Context:

Blinkit is a quick commerce platform serving diverse customer segments across India. With rapid expansion into Tier 2 and Tier 3 cities, understanding sales trends at a granular level is critical for supply chain, merchandising, and marketing decisions.

Key Goals:

•Evaluate sales trends by item type, fat content, and outlet characteristics.
•Identify high-performing outlet types, sizes, and locations.
•Understand how visibility, item count, and ratings affect performance.
•Leverage BI tools and programming to automate reporting and enhance data storytelling.

Outcome:

A dynamic dashboard (built using Power BI) with filterable views and performance KPIs, supported by Python, SQL, Excel, and Tableau for data prep, analysis, and visualization.

🛠 Tools & Technologies Used :

Tool	Usage Description
SQL	Data extraction, transformation, filtering outlet/item information
Excel	Initial data cleaning, pivot tables for quick summaries
Power BI	Interactive dashboard with filters, charts, and KPIs
Tableau	(Optional) Exploratory visualizations to compare insights
Python	Report automation and PDF generation, data manipulation (Pandas

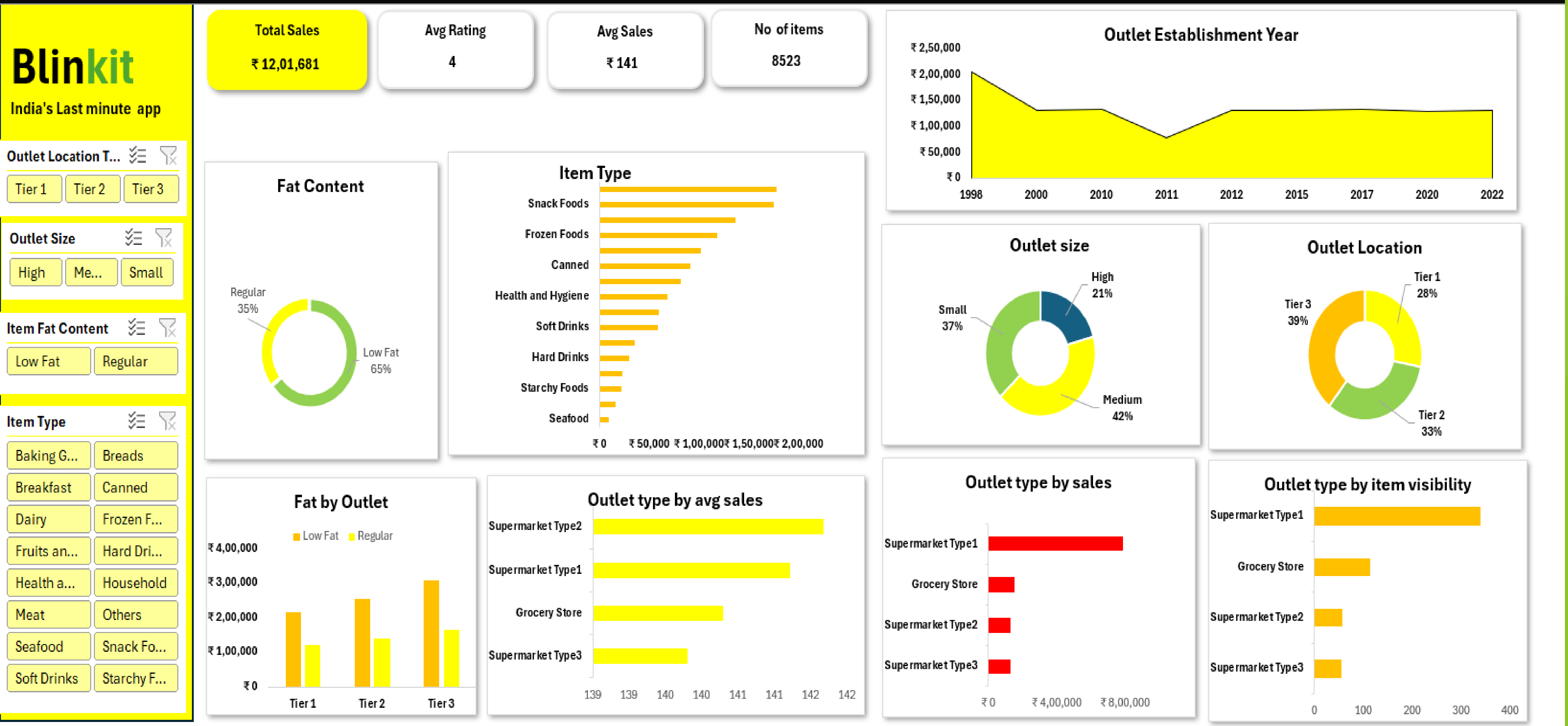
📈 Insights & Visualizations :

This section summarizes key findings derived from the Power BI dashboard, supported by data exploration in Tableau and Python. The dashboard allowed dynamic filtering by outlet type, size, location tier, and item characteristics.

✅ Key Insights:

1. **Tier 3 outlets generate the highest sales** ($472K), showing strong demand in semi-urban and rural areas.
2. **Fruits and Snacks** are the top-performing categories, each contributing close to $200K in revenue.
3. **High-sized outlets account for 42.3% of total sales**, suggesting that larger formats attract more customers and have more inventory.
4. **Regular fat items lead with 64.6% of total item sales**, while low-fat items represent a smaller but notable share.
5. **Supermarket Type 1 and Grocery Stores outperform others**, despite differences in item count and visibility.
6. **Sales in Tier 1 cities are the lowest** ($336K), indicating possible market saturation or competition.
7. **Outlet visibility has a direct impact on item sales**, especially in Supermarket Type 1 outlets.
8. **Outlet establishment dipped in 2010**, but has remained consistent post-2015, showing stable growth.
9. **Categories like Seafood and Breakfast have lower sales**, highlighting potential areas for marketing or inventory review.
10. **Item sales follow the 80/20 rule**, where a few categories drive most of the revenue.

Excel dashboard :



Blinkit

India's Last minute app

Outlet Location T...

Tier 1 | Tier 2 | Tier 3

Outlet Size

High | Me... | Small

Item Fat Content

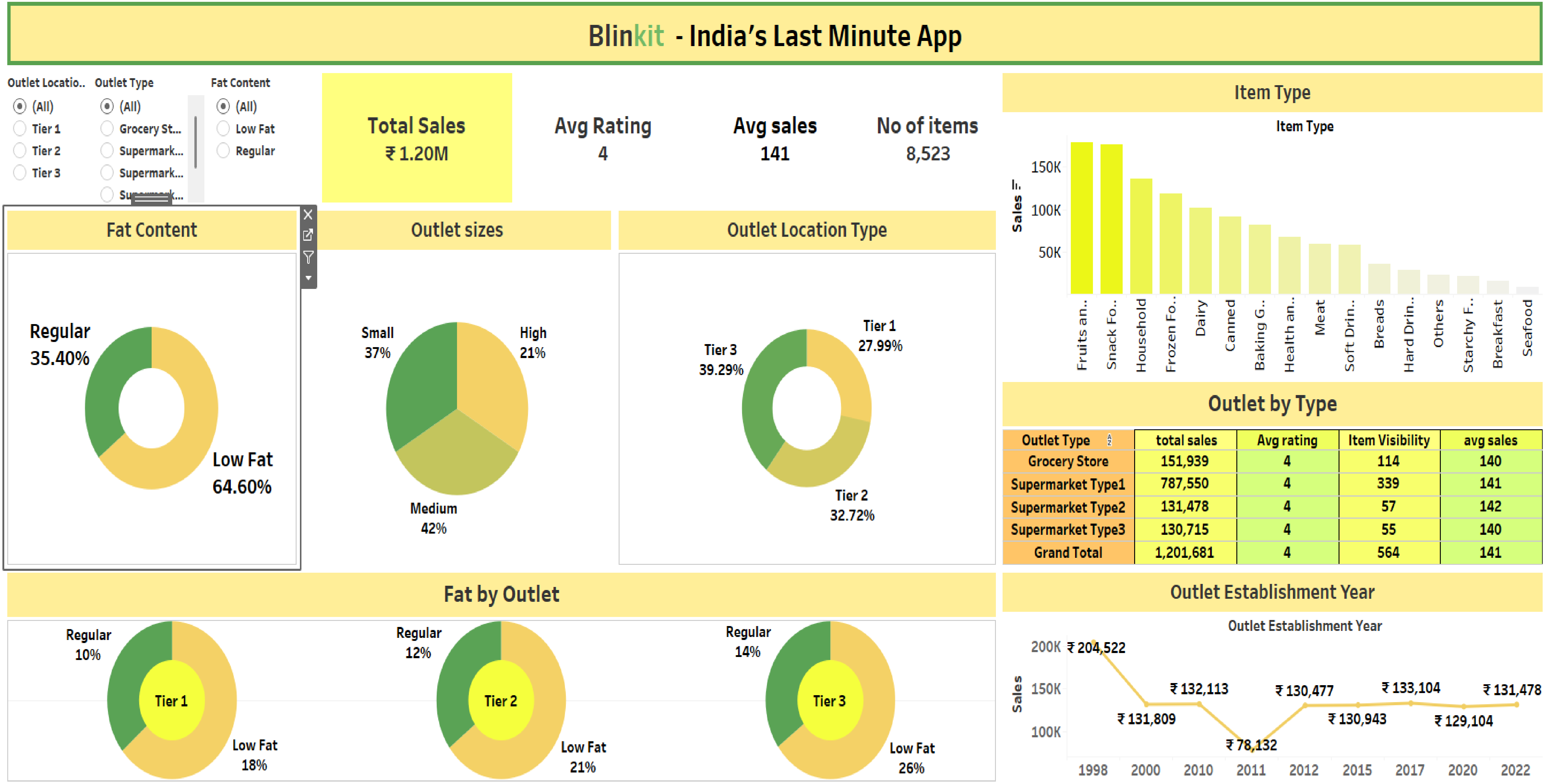
Low Fat | Regular

Item Type

Baking G... | Breads | Breakfast | Canned | Dairy | Frozen F... | Fruits an... | Hard Dri... | Health a... | Household | Meat | Others | Seafood | Snack Fo... | Soft Drinks | Starchy F...

Total Sales	Avg Rating	Avg Sales	No of items
₹ 12,01,681	4	₹ 141	8523

Tableua Dashboard:



Blinkit - India’s Last Minute App

Outlet Locatio..
- ● (All)
- ○ Tier 1
- ○ Tier 2
- ○ Tier 3

Outlet Type
- ● (All)
- ○ Grocery St...
- ○ Supermark...
- ○ Supermark...
- ○ Su...

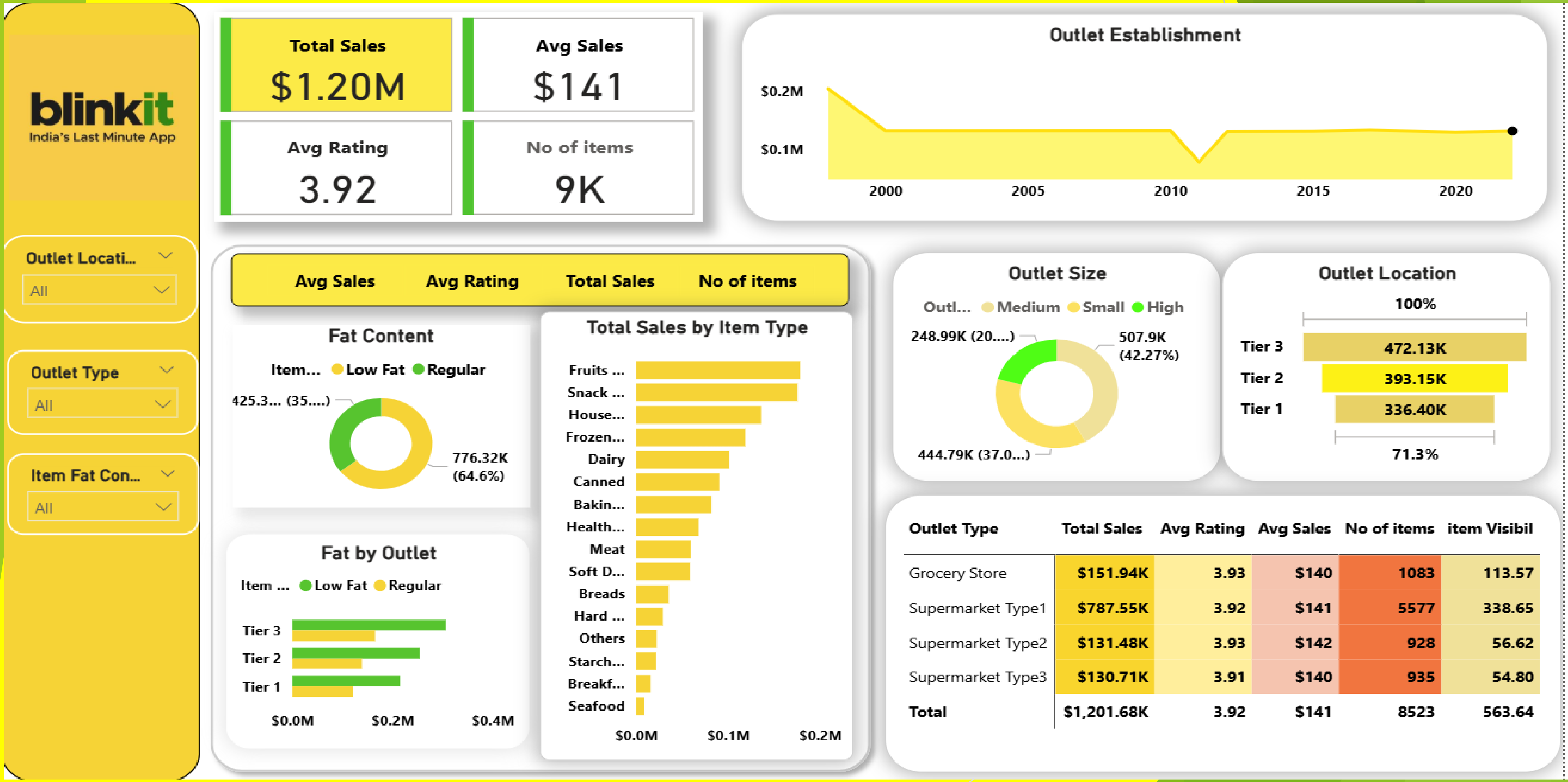
Fat Content
- ● (All)
- ○ Low Fat
- ○ Regular

Total Sales	Avg Rating	Avg sales	No of items
₹ 1.20M	4	141	8,523

Outlet by Type

Outlet Type	total sales	Avg rating	Item Visibility	avg sales
Grocery Store	151,939	4	114	140
Supermarket Type1	787,550	4	339	141
Supermarket Type2	131,478	4	57	142
Supermarket Type3	130,715	4	55	140
Grand Total	1,201,681	4	564	141

Powerbi dashboard:



Total Sales
$1.20M

Avg Sales
$141

Avg Rating
3.92

No of items
9K

Outlet Locati... All

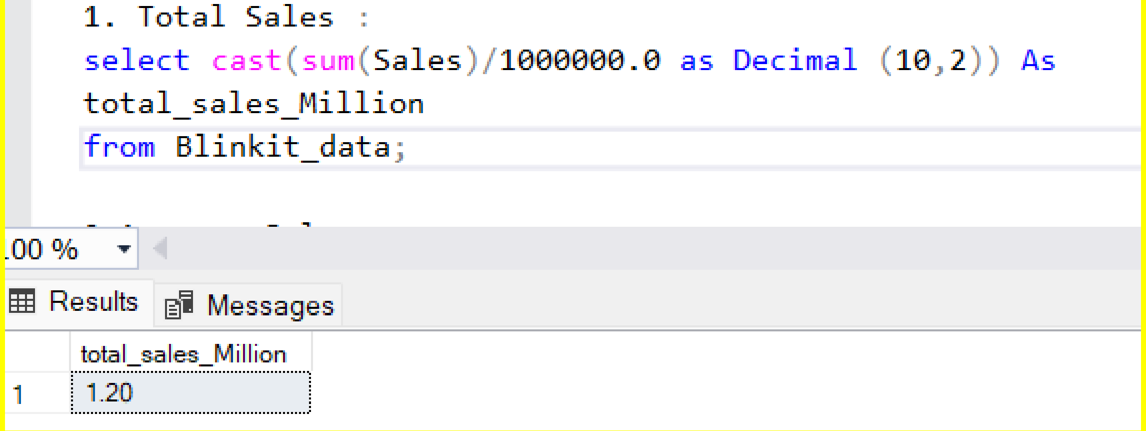
Outlet Type All

Item Fat Con... All

Avg Sales | Avg Rating | Total Sales | No of items

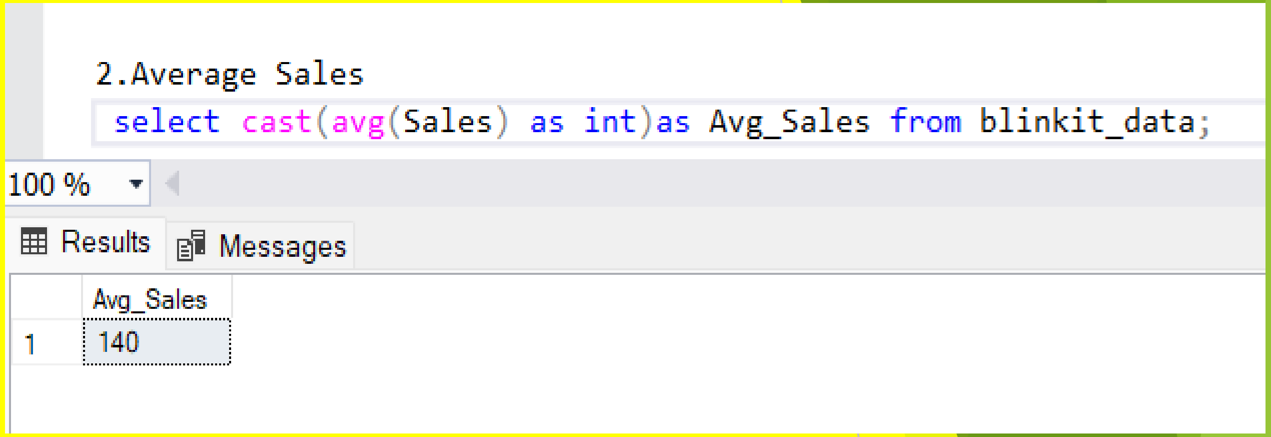
Outlet Type	Total Sales	Avg Rating	Avg Sales	No of items	item Visibil
Grocery Store	$151.94K	3.93	$140	1083	113.57
Supermarket Type1	$787.55K	3.92	$141	5577	338.65
Supermarket Type2	$131.48K	3.93	$142	928	56.62
Supermarket Type3	$130.71K	3.91	$140	935	54.80
Total	$1,201.68K	3.92	$141	8523	563.64

SQl Queries :



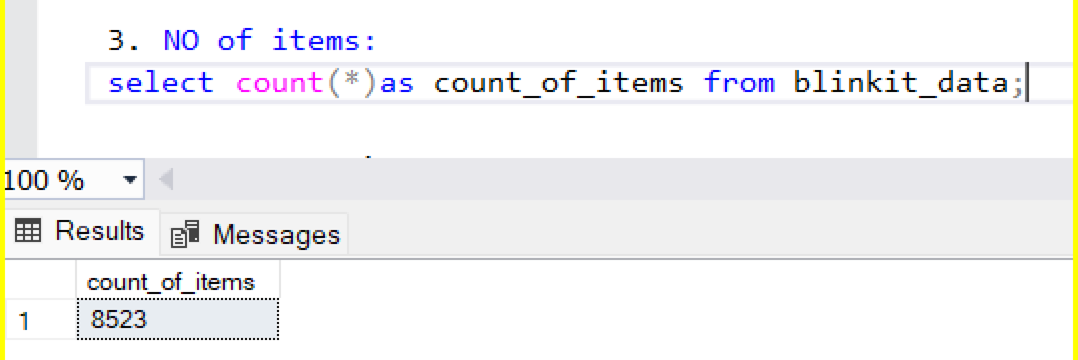
```
1. Total Sales :
select cast(sum(Sales)/1000000.0 as Decimal (10,2)) As
total_sales_Million
from Blinkit_data;
```

	total_sales_Million
1	1.20



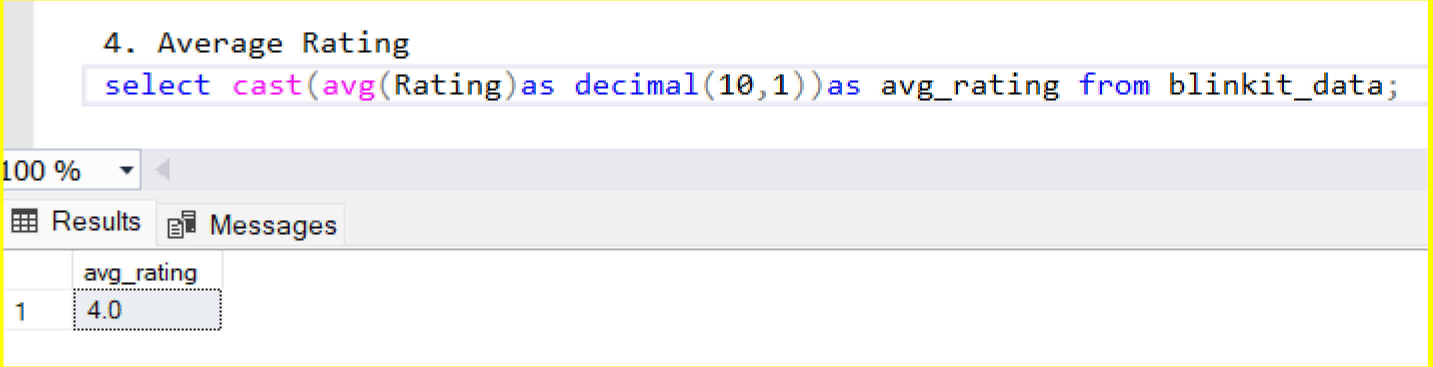
```
2.Average Sales
select cast(avg(Sales) as int)as Avg_Sales from blinkit_data;
```

	Avg_Sales
1	140



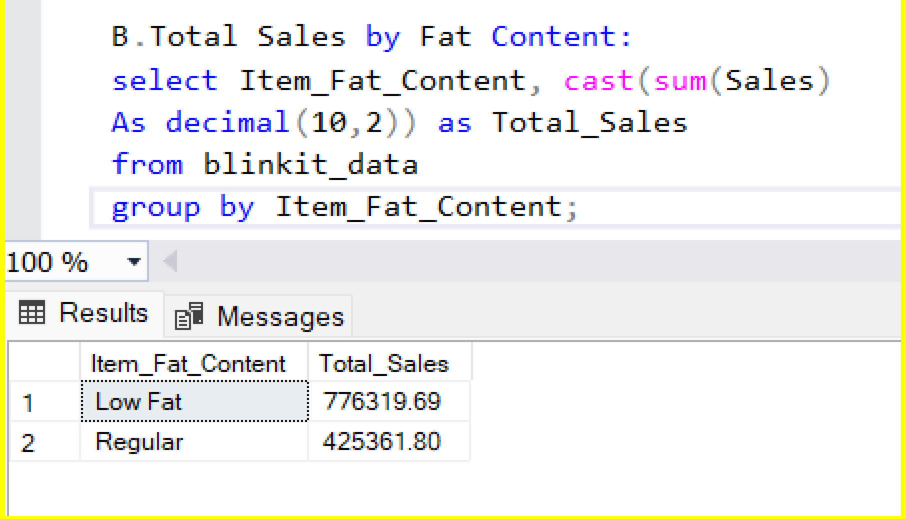
```
3. NO of items:
select count(*)as count_of_items from blinkit_data;
```

	count_of_items
1	8523



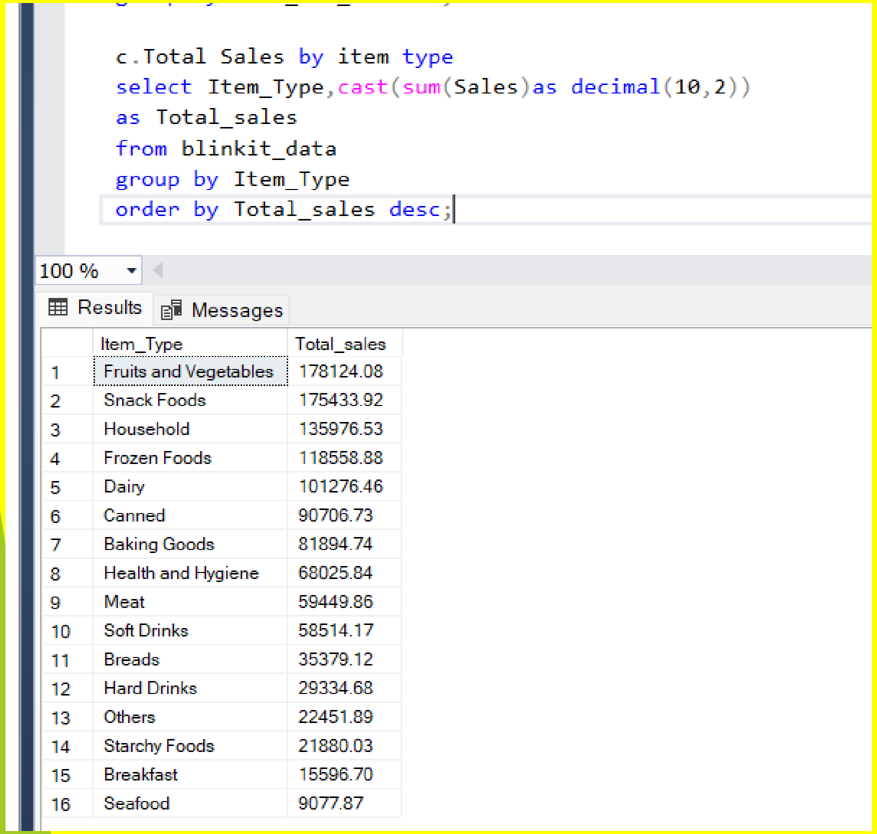
```
4. Average Rating
select cast(avg(Rating)as decimal(10,1))as avg_rating from blinkit_data;
```

	avg_rating
1	4.0



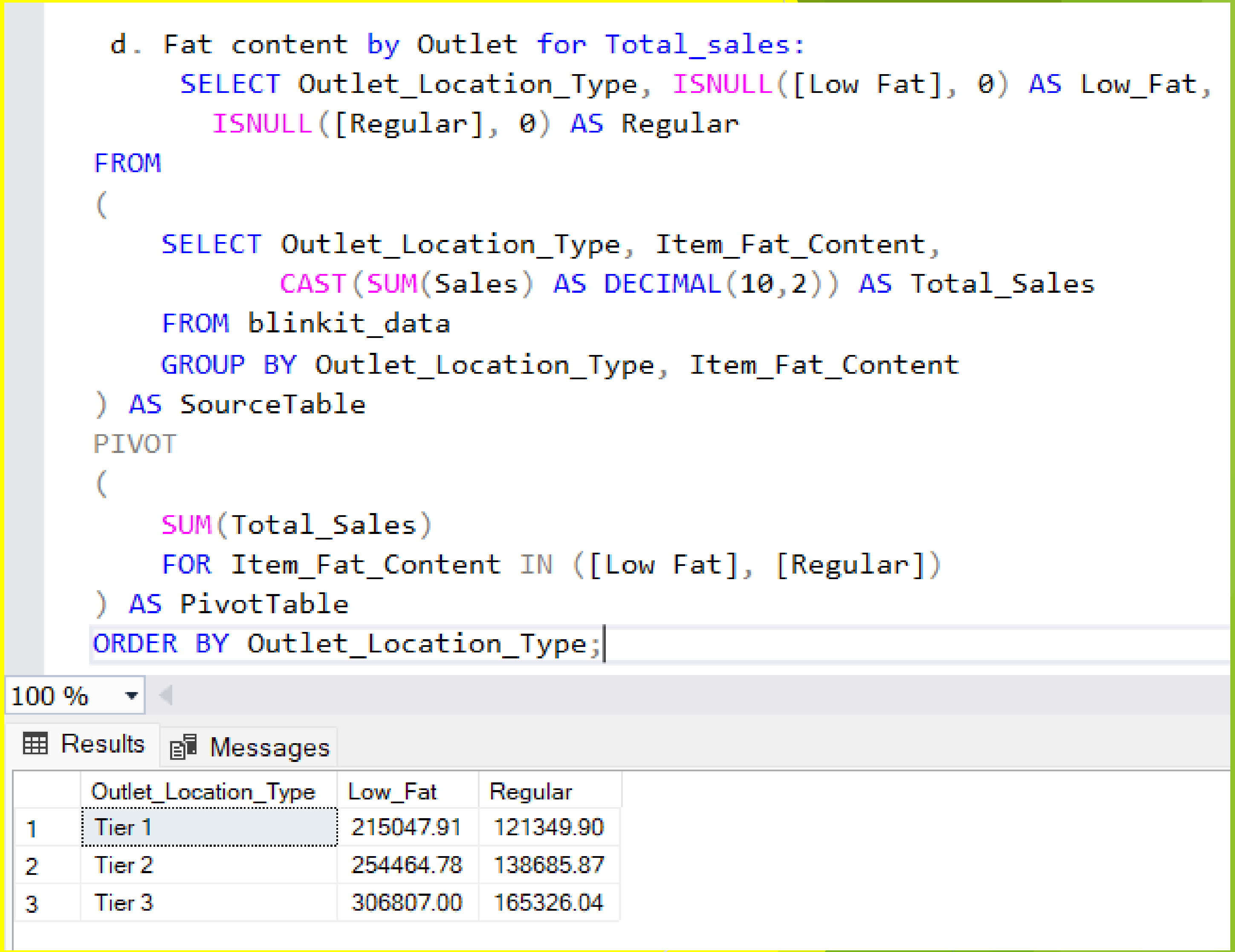
```
B.Total Sales by Fat Content:
select Item_Fat_Content, cast(sum(Sales)
As decimal(10,2)) as Total_Sales
from blinkit_data
group by Item_Fat_Content;
```

	Item_Fat_Content	Total_Sales
1	Low Fat	776319.69
2	Regular	425361.80



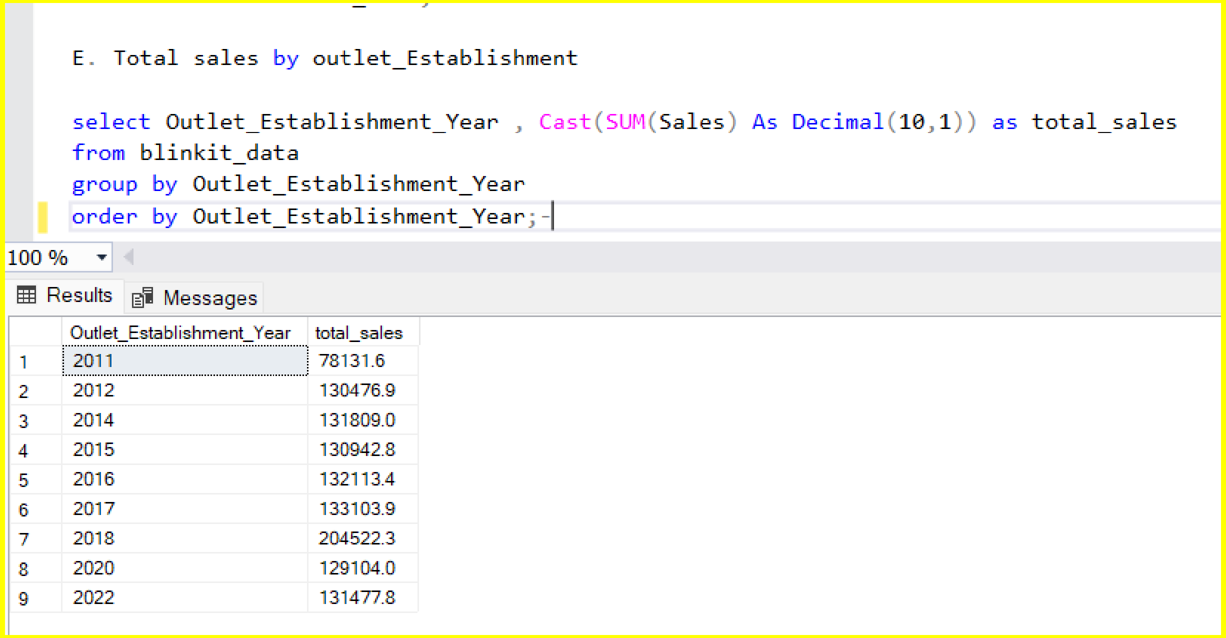
```
c.Total Sales by item type
select Item_Type,cast(sum(Sales)as decimal(10,2))
as Total_sales
from blinkit_data
group by Item_Type
order by Total_sales desc;
```

	Item_Type	Total_sales
1	Fruits and Vegetables	178124.08
2	Snack Foods	175433.92
3	Household	135976.53
4	Frozen Foods	118558.88
5	Dairy	101276.46
6	Canned	90706.73
7	Baking Goods	81894.74
8	Health and Hygiene	68025.84
9	Meat	59449.86
10	Soft Drinks	58514.17
11	Breads	35379.12
12	Hard Drinks	29334.68
13	Others	22451.89
14	Starchy Foods	21880.03
15	Breakfast	15596.70
16	Seafood	9077.87



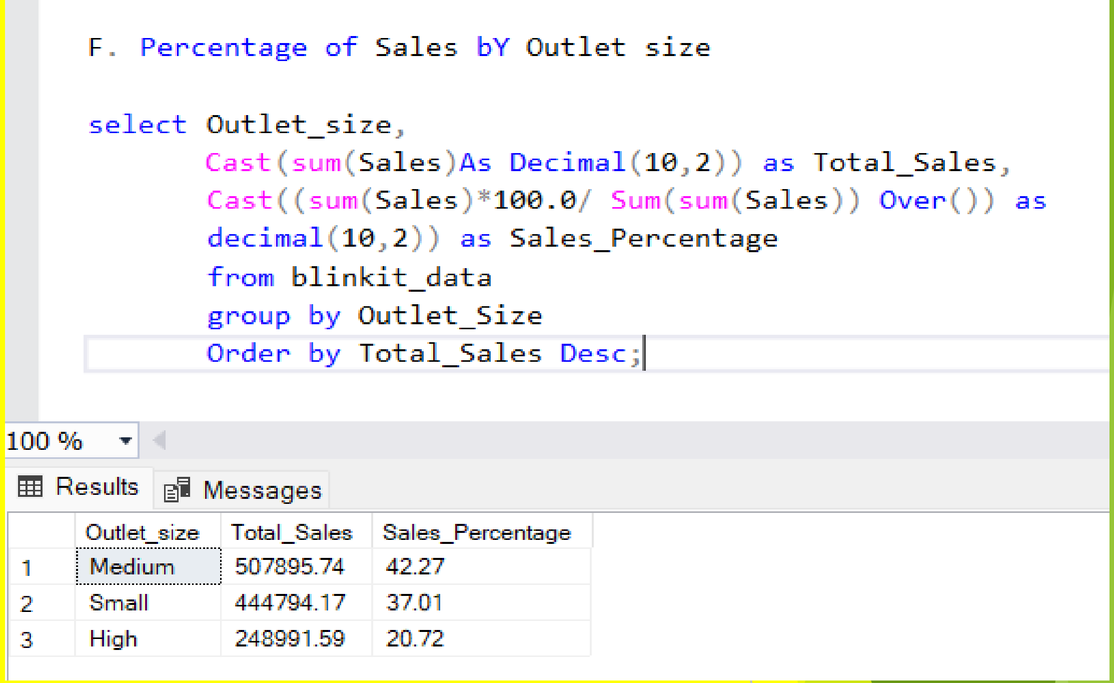
```
 d. Fat content by Outlet for Total_sales:
     SELECT Outlet_Location_Type, ISNULL([Low Fat], 0) AS Low_Fat,
       ISNULL([Regular], 0) AS Regular
FROM
(
    SELECT Outlet_Location_Type, Item_Fat_Content,
           CAST(SUM(Sales) AS DECIMAL(10,2)) AS Total_Sales
    FROM blinkit_data
    GROUP BY Outlet_Location_Type, Item_Fat_Content
) AS SourceTable
PIVOT
(
    SUM(Total_Sales)
    FOR Item_Fat_Content IN ([Low Fat], [Regular])
) AS PivotTable
ORDER BY Outlet_Location_Type;
```

	Outlet_Location_Type	Low_Fat	Regular
1	Tier 1	215047.91	121349.90
2	Tier 2	254464.78	138685.87
3	Tier 3	306807.00	165326.04



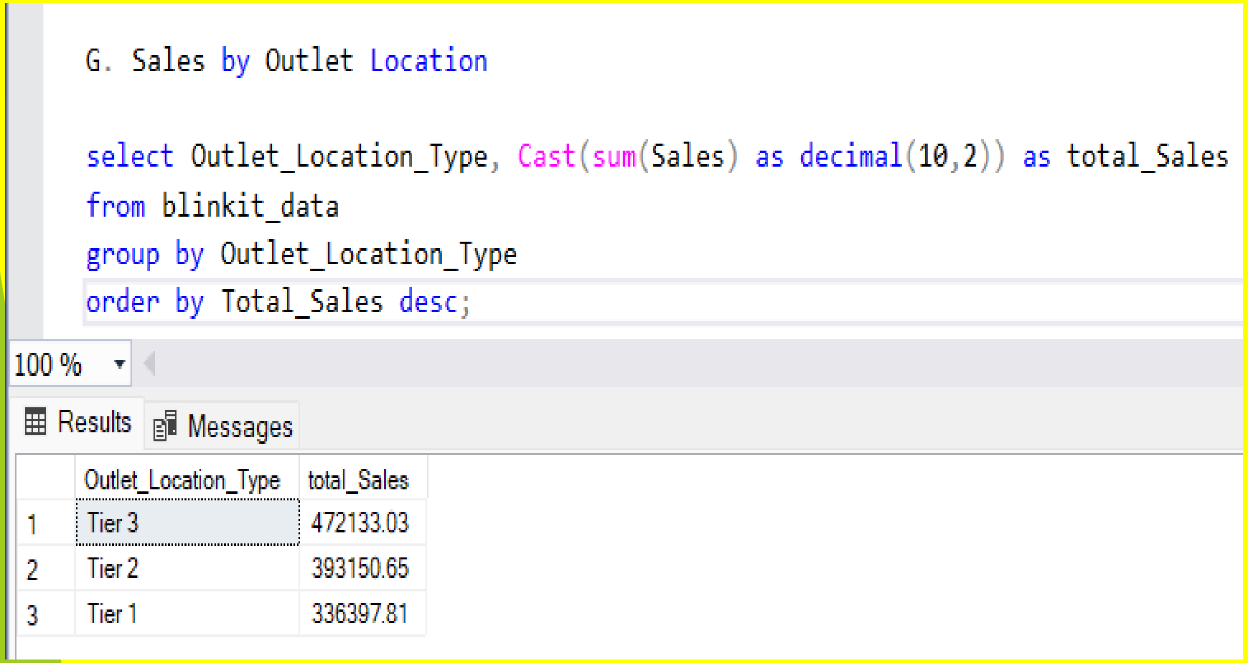
```
E. Total sales by outlet_Establishment

select Outlet_Establishment_Year , Cast(SUM(Sales) As Decimal(10,1)) as total_sales
from blinkit_data
group by Outlet_Establishment_Year
order by Outlet_Establishment_Year;
```

	Outlet_Establishment_Year	total_sales
1	2011	78131.6
2	2012	130476.9
3	2014	131809.0
4	2015	130942.8
5	2016	132113.4
6	2017	133103.9
7	2018	204522.3
8	2020	129104.0
9	2022	131477.8



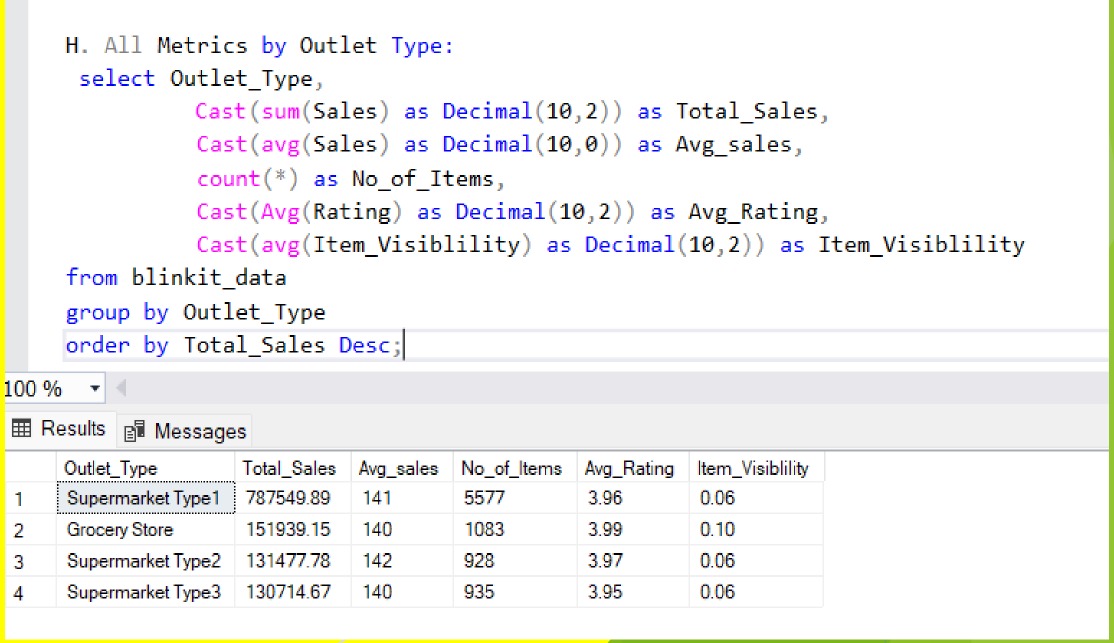
```
F. Percentage of Sales bY Outlet size

select Outlet_size,
       Cast(sum(Sales)As Decimal(10,2)) as Total_Sales,
       Cast((sum(Sales)*100.0/ Sum(sum(Sales)) Over()) as
       decimal(10,2)) as Sales_Percentage
       from blinkit_data
       group by Outlet_Size
       Order by Total_Sales Desc;
```

	Outlet_size	Total_Sales	Sales_Percentage
1	Medium	507895.74	42.27
2	Small	444794.17	37.01
3	High	248991.59	20.72



```
G. Sales by Outlet Location

select Outlet_Location_Type, Cast(sum(Sales) as decimal(10,2)) as total_Sales
from blinkit_data
group by Outlet_Location_Type
order by Total_Sales desc;
```

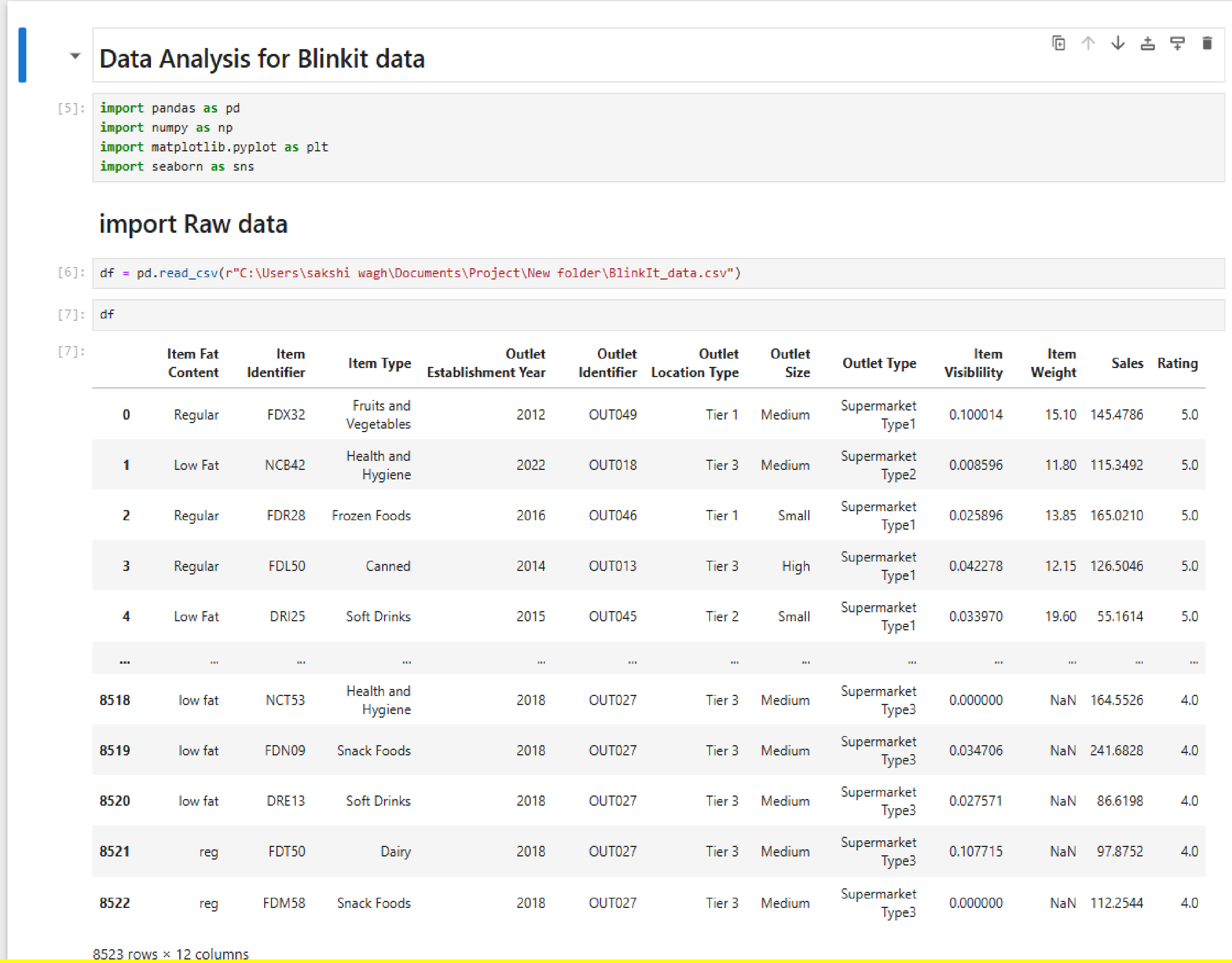
	Outlet_Location_Type	total_Sales
1	Tier 3	472133.03
2	Tier 2	393150.65
3	Tier 1	336397.81



```
H. All Metrics by Outlet Type:
 select Outlet_Type,
        Cast(sum(Sales) as Decimal(10,2)) as Total_Sales,
        Cast(avg(Sales) as Decimal(10,0)) as Avg_sales,
        count(*) as No_of_Items,
        Cast(Avg(Rating) as Decimal(10,2)) as Avg_Rating,
        Cast(avg(Item_Visiblility) as Decimal(10,2)) as Item_Visiblility
from blinkit_data
group by Outlet_Type
order by Total_Sales Desc;
```

	Outlet_Type	Total_Sales	Avg_sales	No_of_Items	Avg_Rating	Item_Visiblility
1	Supermarket Type1	787549.89	141	5577	3.96	0.06
2	Grocery Store	151939.15	140	1083	3.99	0.10
3	Supermarket Type2	131477.78	142	928	3.97	0.06
4	Supermarket Type3	130714.67	140	935	3.95	0.06

Python :



Data Analysis for Blinkit data

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

import Raw data

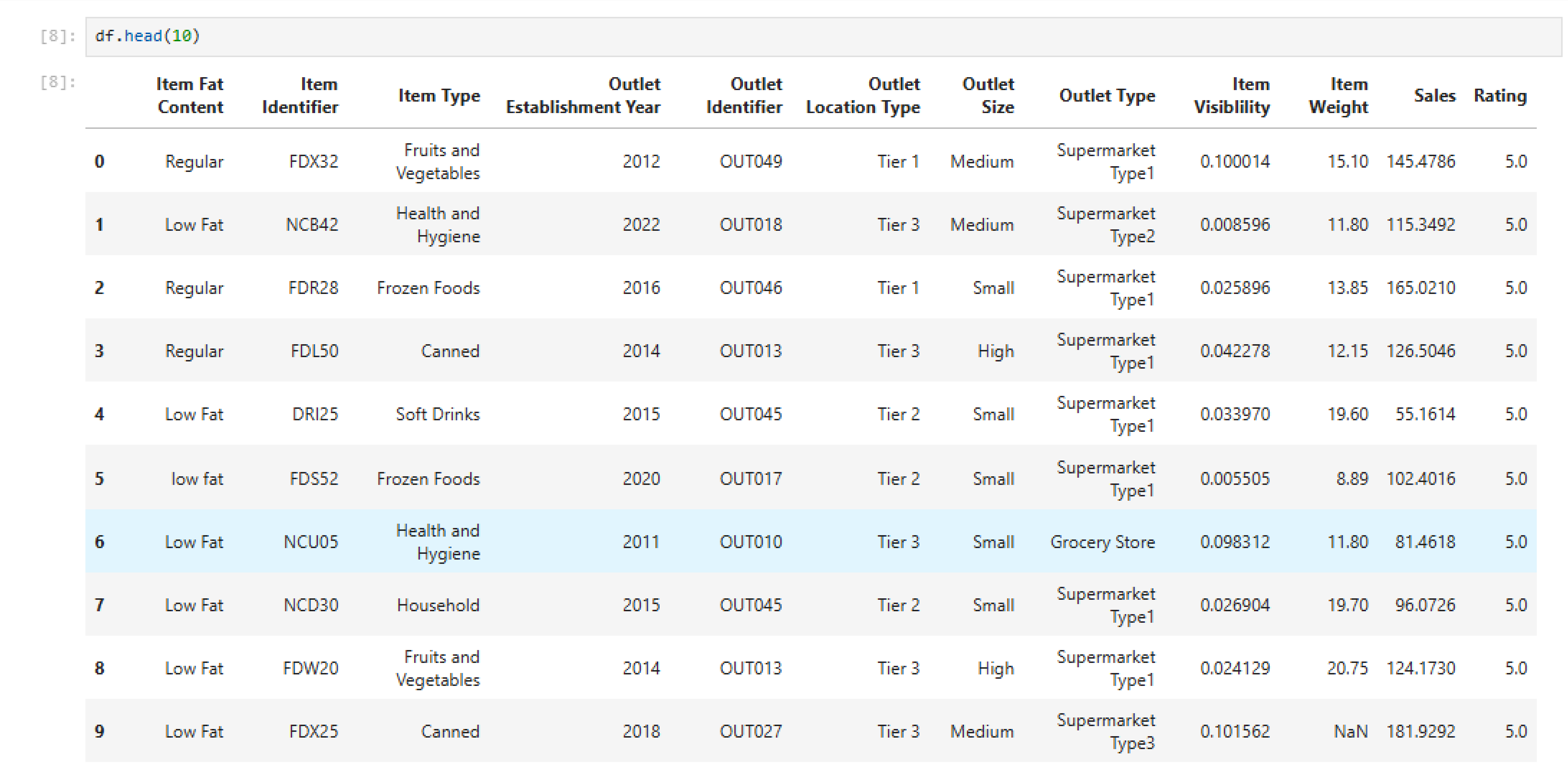
```
[6]: df = pd.read_csv(r"C:\Users\sakshi wagh\Documents\Project\New folder\BlinkIt_data.csv")
```

```
[7]: df
```

[7]:

	Item Fat Content	Item Identifier	Item Type	Outlet Establishment Year	Outlet Identifier	Outlet Location Type	Outlet Size	Outlet Type	Item Visiblility	Item Weight	Sales	Rating
0	Regular	FDX32	Fruits and Vegetables	2012	OUT049	Tier 1	Medium	Supermarket Type1	0.100014	15.10	145.4786	5.0
1	Low Fat	NCB42	Health and Hygiene	2022	OUT018	Tier 3	Medium	Supermarket Type2	0.008596	11.80	115.3492	5.0
2	Regular	FDR28	Frozen Foods	2016	OUT046	Tier 1	Small	Supermarket Type1	0.025896	13.85	165.0210	5.0
3	Regular	FDL50	Canned	2014	OUT013	Tier 3	High	Supermarket Type1	0.042278	12.15	126.5046	5.0
4	Low Fat	DRI25	Soft Drinks	2015	OUT045	Tier 2	Small	Supermarket Type1	0.033970	19.60	55.1614	5.0
...	...	...	...	...	...	...	...	...	...	...	...	...
8518	low fat	NCT53	Health and Hygiene	2018	OUT027	Tier 3	Medium	Supermarket Type3	0.000000	NaN	164.5526	4.0
8519	low fat	FDN09	Snack Foods	2018	OUT027	Tier 3	Medium	Supermarket Type3	0.034706	NaN	241.6828	4.0
8520	low fat	DRE13	Soft Drinks	2018	OUT027	Tier 3	Medium	Supermarket Type3	0.027571	NaN	86.6198	4.0
8521	reg	FDT50	Dairy	2018	OUT027	Tier 3	Medium	Supermarket Type3	0.107715	NaN	97.8752	4.0
8522	reg	FDM58	Snack Foods	2018	OUT027	Tier 3	Medium	Supermarket Type3	0.000000	NaN	112.2544	4.0

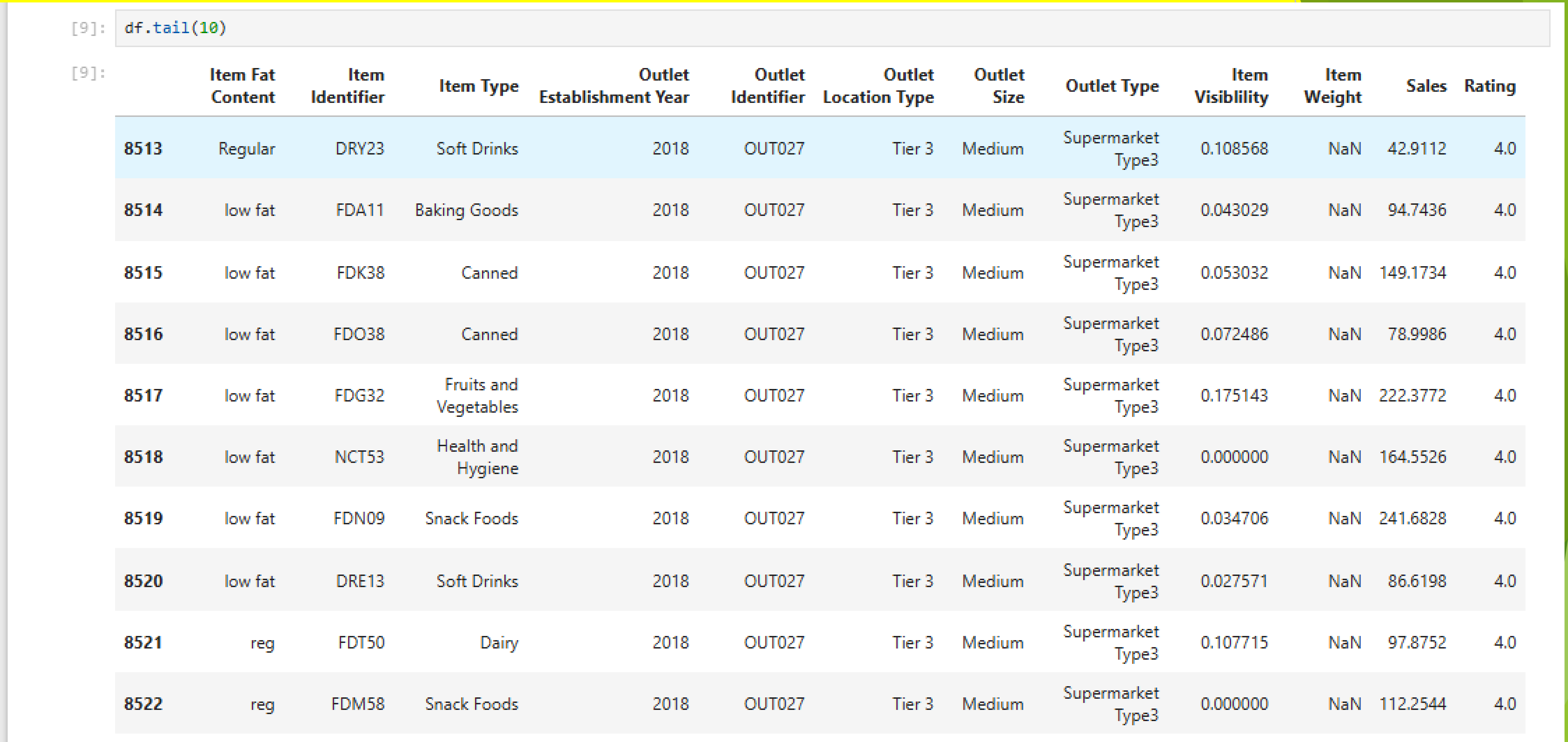
8523 rows × 12 columns



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

[8]:

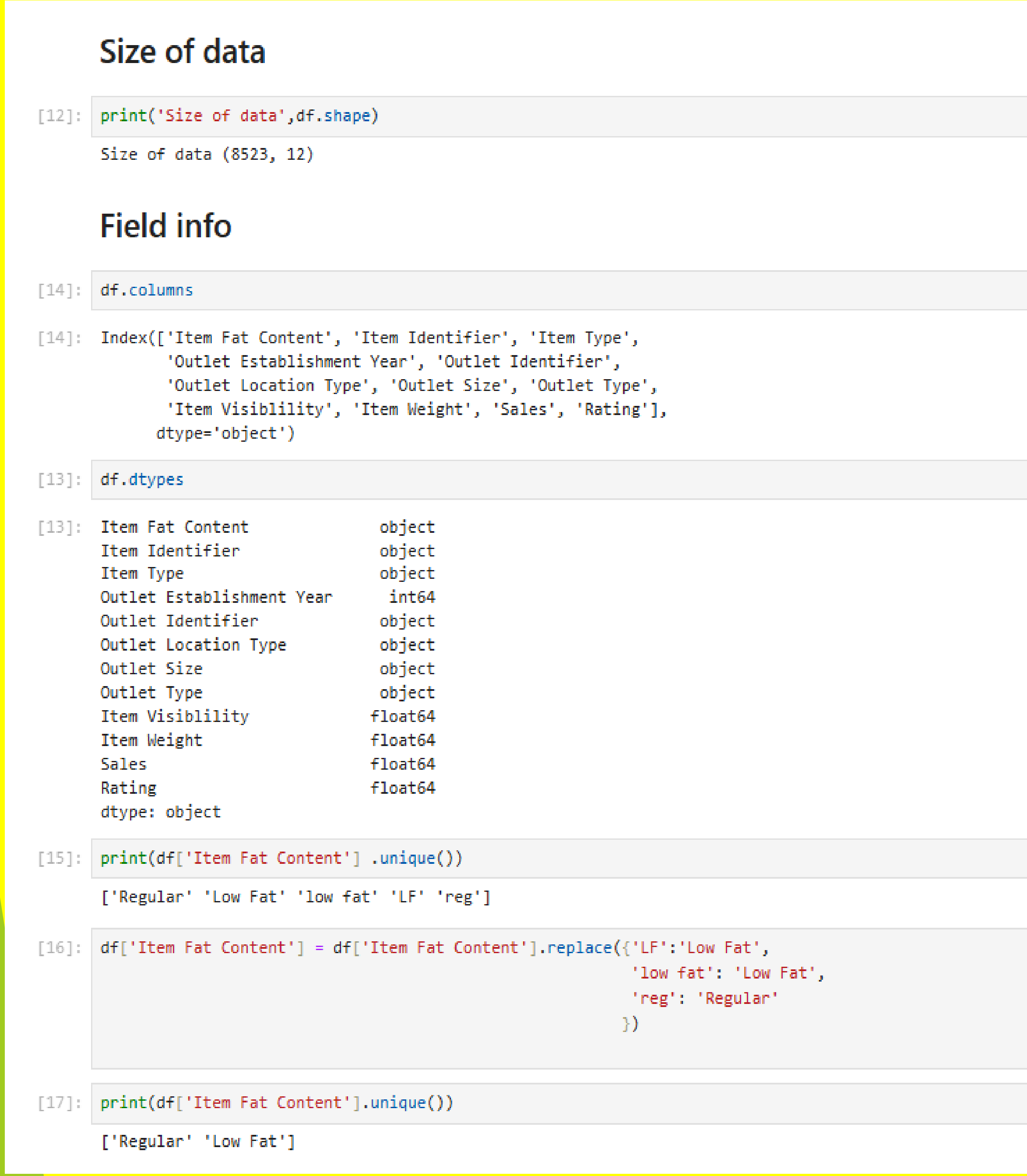
	Item Fat Content	Item Identifier	Item Type	Outlet Establishment Year	Outlet Identifier	Outlet Location Type	Outlet Size	Outlet Type	Item Visiblility	Item Weight	Sales	Rating
0	Regular	FDX32	Fruits and Vegetables	2012	OUT049	Tier 1	Medium	Supermarket Type1	0.100014	15.10	145.4786	5.0
1	Low Fat	NCB42	Health and Hygiene	2022	OUT018	Tier 3	Medium	Supermarket Type2	0.008596	11.80	115.3492	5.0
2	Regular	FDR28	Frozen Foods	2016	OUT046	Tier 1	Small	Supermarket Type1	0.025896	13.85	165.0210	5.0
3	Regular	FDL50	Canned	2014	OUT013	Tier 3	High	Supermarket Type1	0.042278	12.15	126.5046	5.0
4	Low Fat	DRI25	Soft Drinks	2015	OUT045	Tier 2	Small	Supermarket Type1	0.033970	19.60	55.1614	5.0
5	low fat	FDS52	Frozen Foods	2020	OUT017	Tier 2	Small	Supermarket Type1	0.005505	8.89	102.4016	5.0
6	Low Fat	NCU05	Health and Hygiene	2011	OUT010	Tier 3	Small	Grocery Store	0.098312	11.80	81.4618	5.0
7	Low Fat	NCD30	Household	2015	OUT045	Tier 2	Small	Supermarket Type1	0.026904	19.70	96.0726	5.0
8	Low Fat	FDW20	Fruits and Vegetables	2014	OUT013	Tier 3	High	Supermarket Type1	0.024129	20.75	124.1730	5.0
9	Low Fat	FDX25	Canned	2018	OUT027	Tier 3	Medium	Supermarket Type3	0.101562	NaN	181.9292	5.0



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

[9]:

	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
8513	Regular	DRY23	Soft Drinks	2018	OUT027	Tier 3	Medium	Supermarket Type3	0.108568	NaN	42.9112	4.0
8514	low fat	FDA11	Baking Goods	2018	OUT027	Tier 3	Medium	Supermarket Type3	0.043029	NaN	94.7436	4.0
8515	low fat	FDK38	Canned	2018	OUT027	Tier 3	Medium	Supermarket Type3	0.053032	NaN	149.1734	4.0
8516	low fat	FDO38	Canned	2018	OUT027	Tier 3	Medium	Supermarket Type3	0.072486	NaN	78.9986	4.0
8517	low fat	FDG32	Fruits and Vegetables	2018	OUT027	Tier 3	Medium	Supermarket Type3	0.175143	NaN	222.3772	4.0
8518	low fat	NCT53	Health and Hygiene	2018	OUT027	Tier 3	Medium	Supermarket Type3	0.000000	NaN	164.5526	4.0
8519	low fat	FDN09	Snack Foods	2018	OUT027	Tier 3	Medium	Supermarket Type3	0.034706	NaN	241.6828	4.0
8520	low fat	DRE13	Soft Drinks	2018	OUT027	Tier 3	Medium	Supermarket Type3	0.027571	NaN	86.6198	4.0
8521	reg	FDT50	Dairy	2018	OUT027	Tier 3	Medium	Supermarket Type3	0.107715	NaN	97.8752	4.0
8522	reg	FDM58	Snack Foods	2018	OUT027	Tier 3	Medium	Supermarket Type3	0.000000	NaN	112.2544	4.0



Size of data

```
[12]: print('Size of data',df.shape)
```

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

Field info

```
[14]: df.columns
```

```
[14]: Index(['Item Fat Content', 'Item Identifier', 'Item Type',
              'Outlet Establishment Year', 'Outlet Identifier',
              'Outlet Location Type', 'Outlet Size', 'Outlet Type',
              'Item Visiblility', 'Item Weight', 'Sales', 'Rating'],
             dtype='object')
```

```
[13]: df.dtypes
```

```
[13]: 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 Visiblility             float64
      Item Weight                  float64
      Sales                        float64
      Rating                       float64
      dtype: object
```

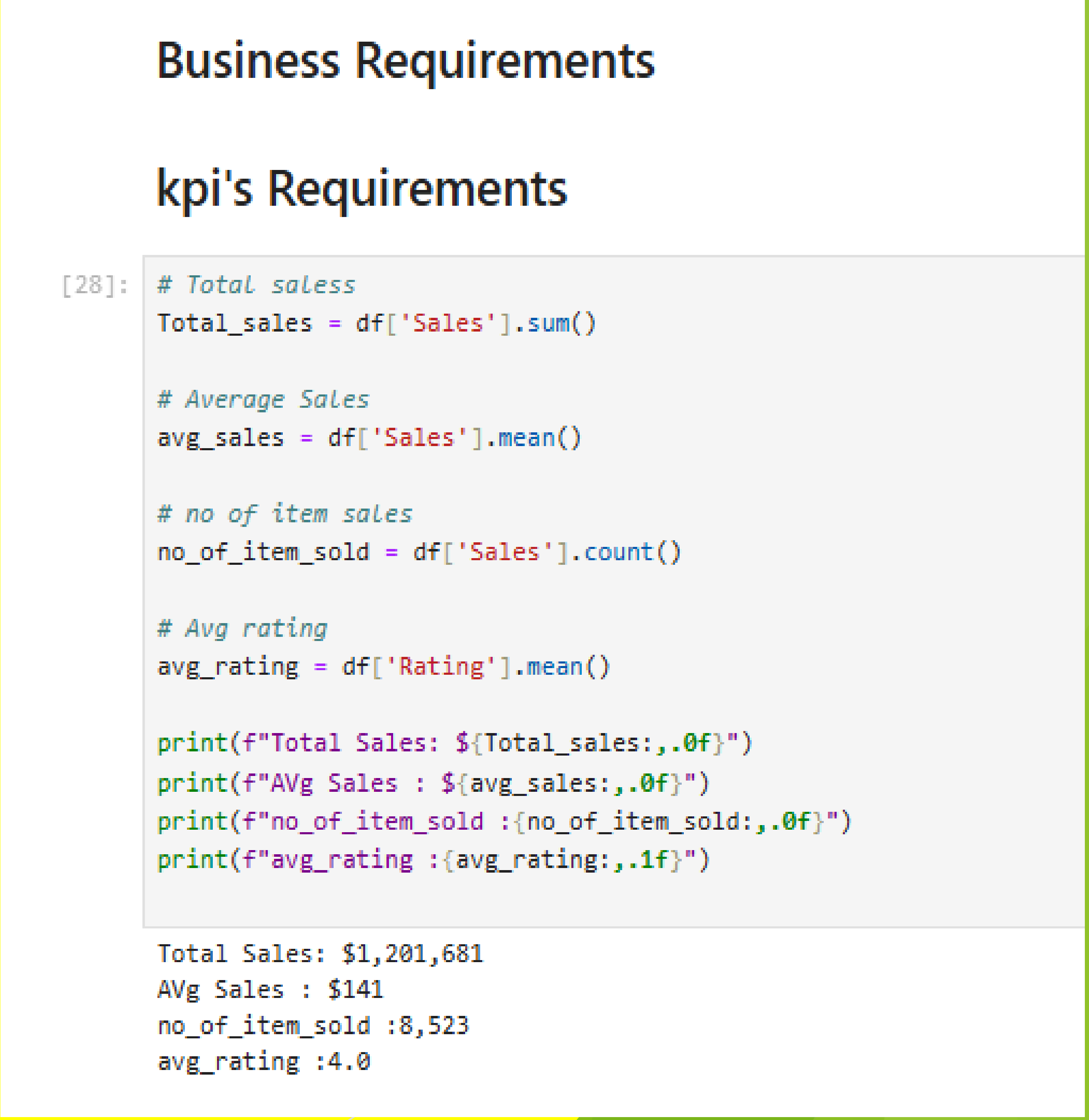
```
[15]: print(df['Item Fat Content'] .unique())
```

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

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

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

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



Business Requirements

kpi's Requirements

```
[28]: # Total saless
      Total_sales = df['Sales'].sum()

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

      # no of item sales
      no_of_item_sold = df['Sales'].count()

      # Avg rating
      avg_rating = df['Rating'].mean()

      print(f"Total Sales: ${Total_sales:,.0f}")
      print(f"AVg Sales : ${avg_sales:,.0f}")
      print(f"no_of_item_sold :{no_of_item_sold:,.0f}")
      print(f"avg_rating :{avg_rating:,.1f}")
```

```
Total Sales: $1,201,681
AVg Sales : $141
no_of_item_sold :8,523
avg_rating :4.0
```

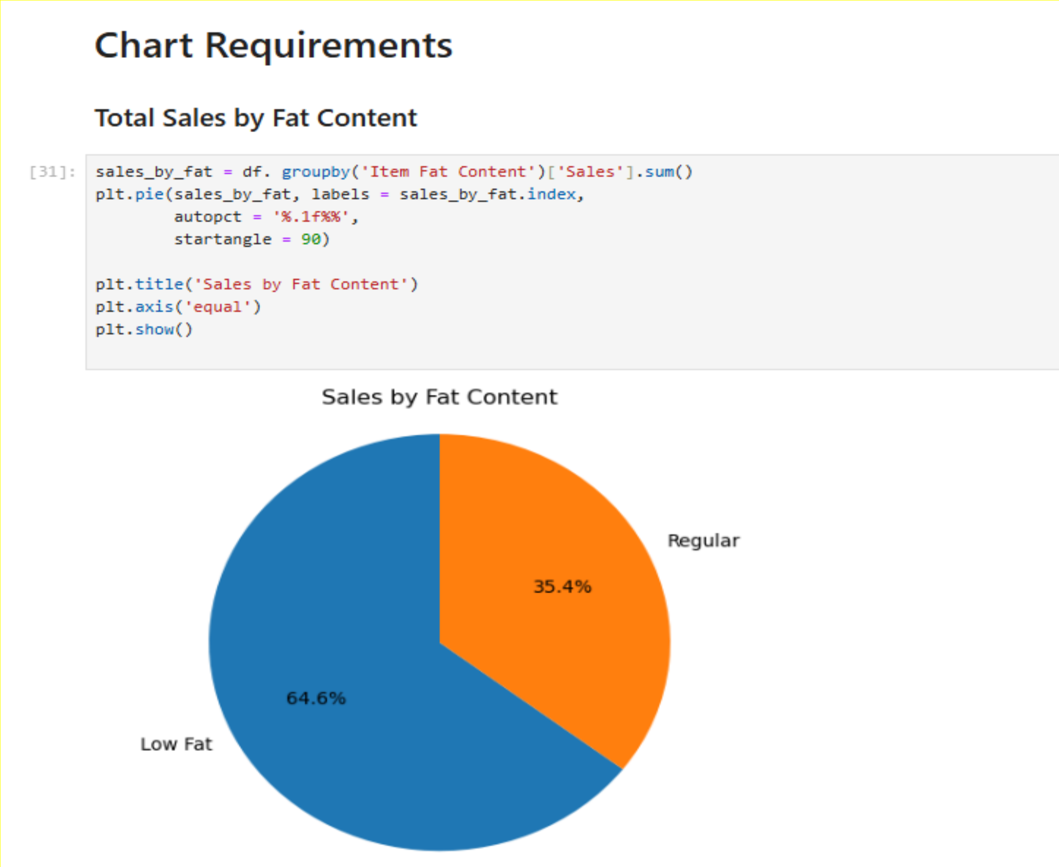
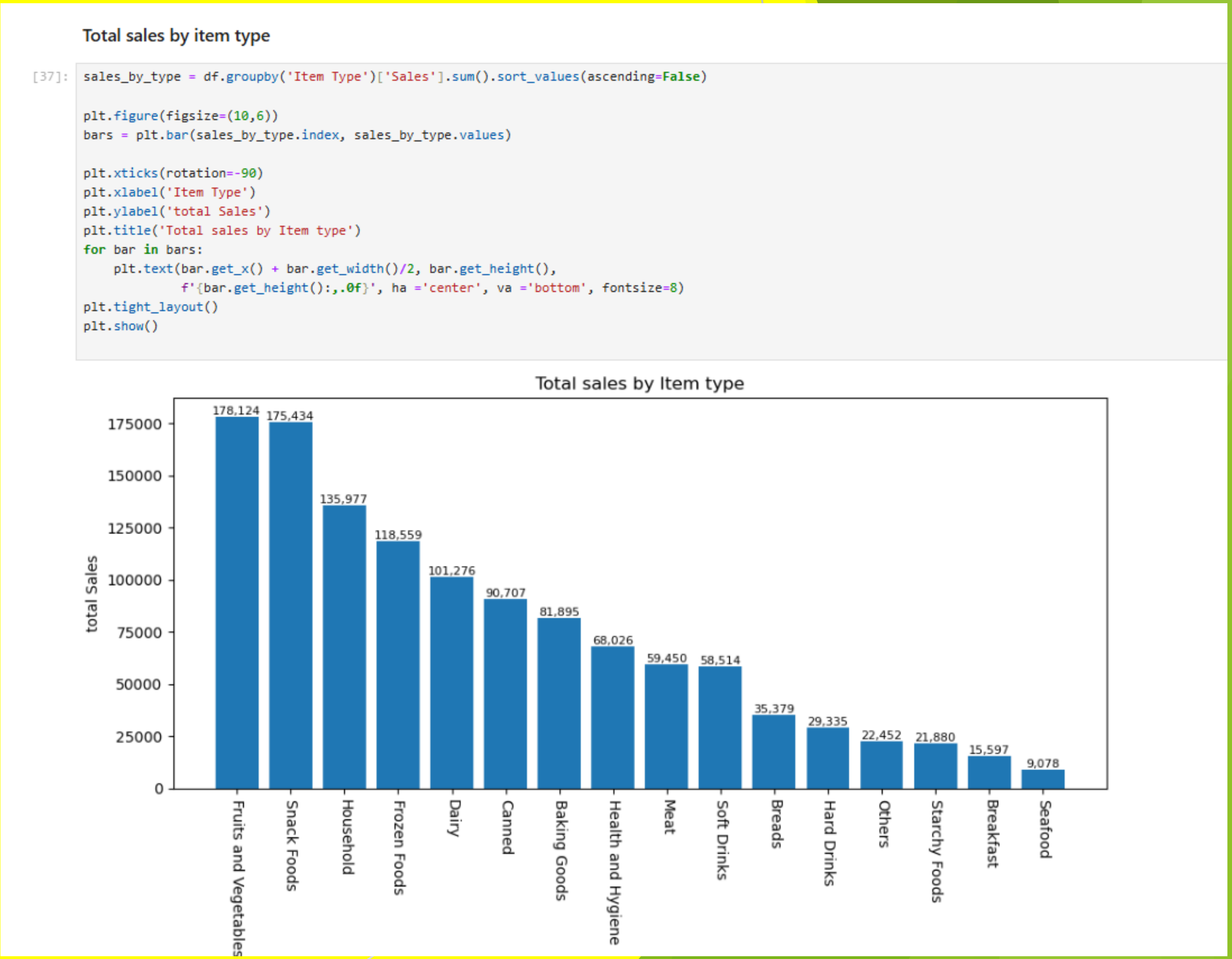


Chart Requirements

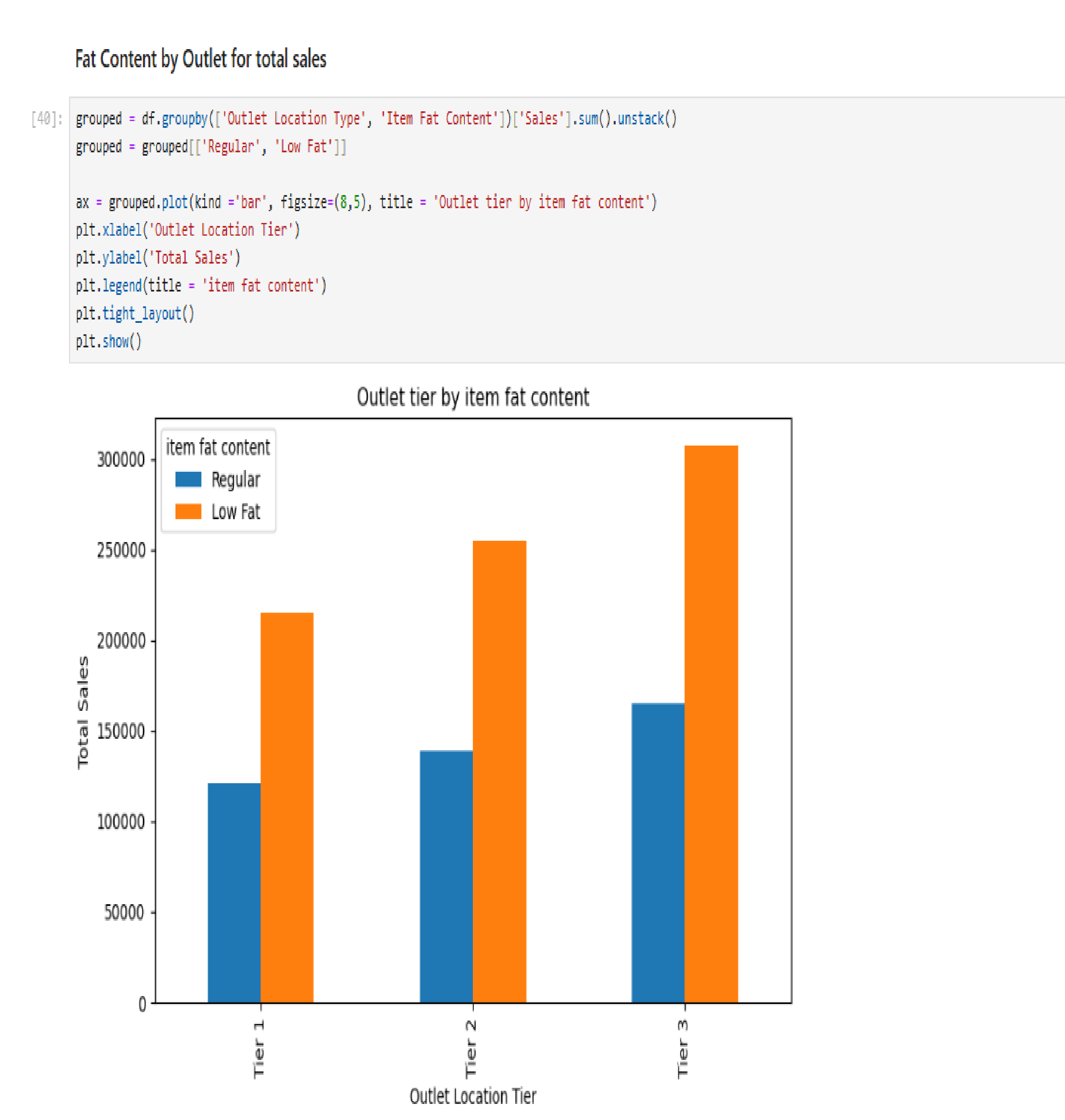
Total Sales by Fat Content

```
[31]: 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()
```



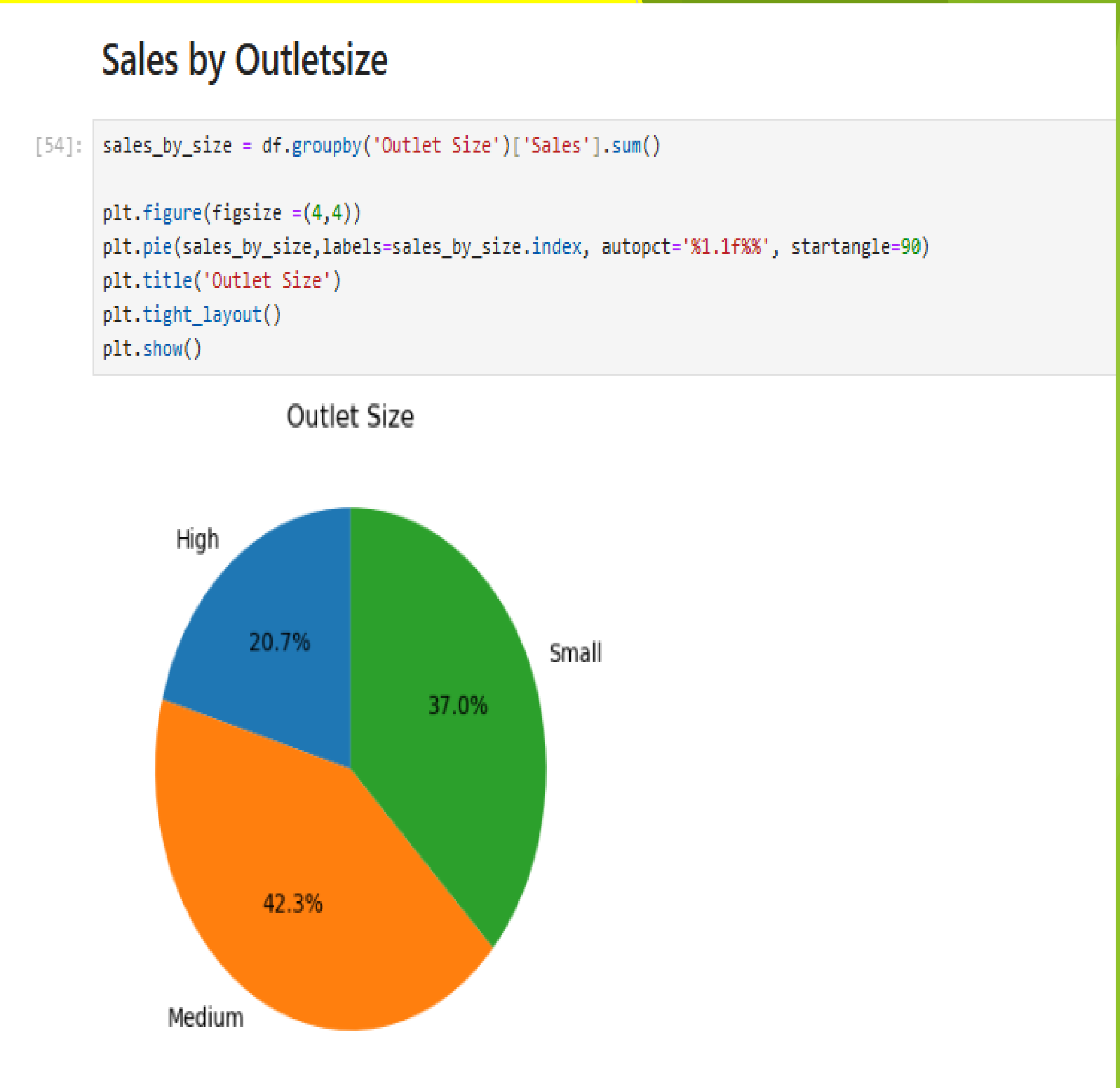
Total sales by item type

```
[37]: 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()
```



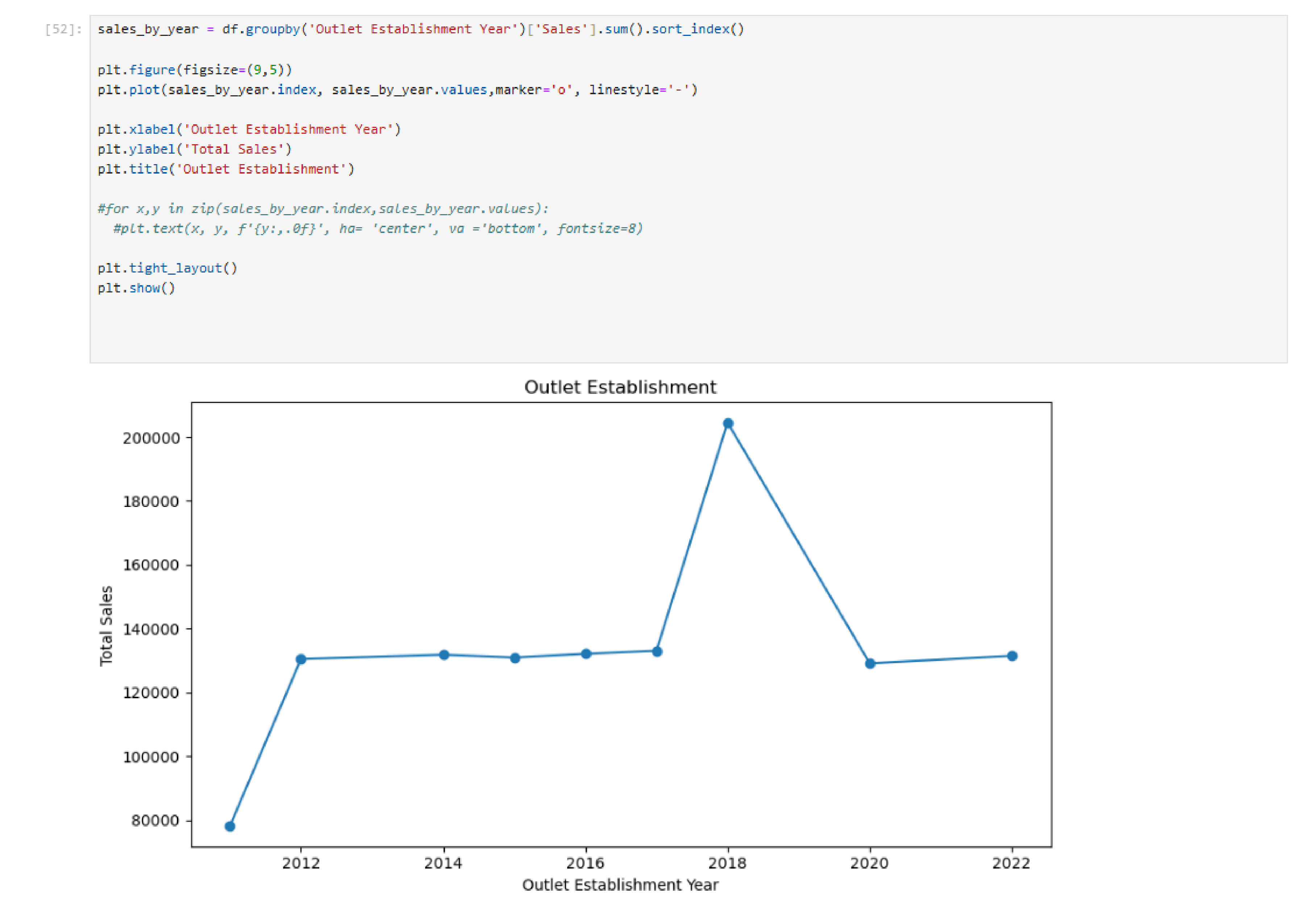
Fat Content by Outlet for total sales

```
[40]: 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 fat content')
      plt.xlabel('Outlet Location Tier')
      plt.ylabel('Total Sales')
      plt.legend(title = 'item fat content')
      plt.tight_layout()
      plt.show()
```

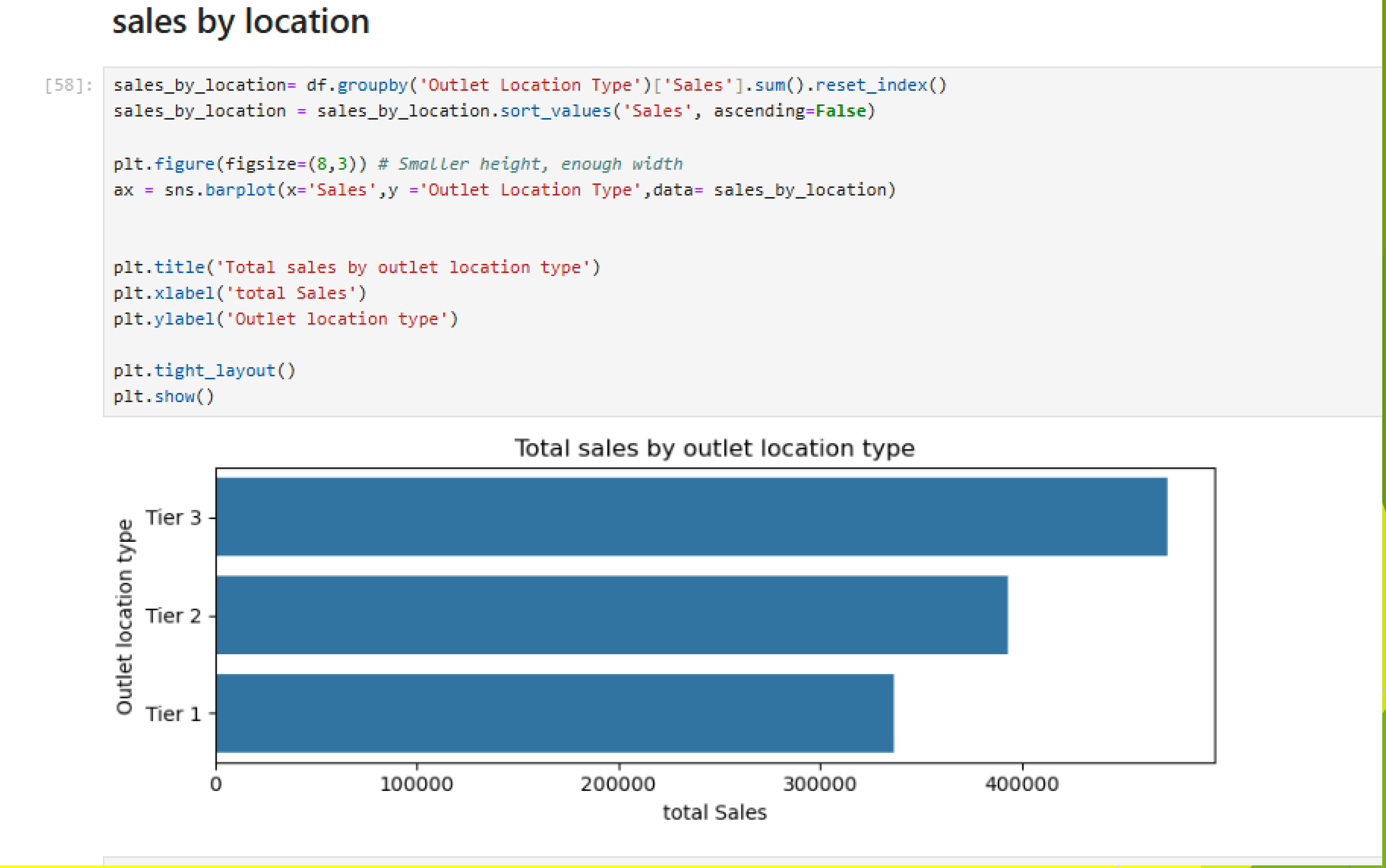


Sales by Outletsize

```
[54]: 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()
```



```
[52]: 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 location

[58]:

```
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)) # 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()
plt.show()
```