Blinkit Grocery store End to End Data Analysis Project

Using

SQL + Python +

Power BI + Microsoft Excel

Report

Tools Used :

1. MS SQL Server Management Studio 21
2. PowerBI Version: 2.148.878.0 64-bit (October, 2025)
3. Python version :3.11
4. MS Excel Office 2019

STEPS IN PROJECT :

1. Requirement Gathering/ Business Requirements
2. Data Walkthrough
3. Data Connection
4. Data Cleaning / Quality Check
5. Data Modeling
6. Data Processing
7. DAX Calculations
8. Dashboard Layouting
9. Charts Development and Formatting
10. Dashboard / Report Development
11. Insights Generation

Analysis of Blinkit data using SQL

==========Business Requirements==========

Query :

1. **Number of Items :** The total count of different items sold.

```
select count(*) as no_of_items from blinkit_data;
```

output :

	no_of_items
1	8523

2. Total Sales : The overall revenue generatedfrom all items sold.

```
select sum(Total_Sales) from blinkit_data ;
```

	Total_sales
1	1201681.47996712

select cast(sum(Total_Sales)/1000000 as decimal(10,2)) as Total_Sales_Millions from blinkit_data;

	Total_Sales_Millions
1	1.20

3. **Average Sales** : The average revenue per sale.

select avg(Total_Sales) as average_total_sales from blinkit_data ;

	average_total_sales
1	140.992781880455

SELECT CAST(SUM(Total_Sales)/1000000 AS DECIMAL(10,2)) AS Total_Sales_Millions

FROM blinkit_data

WHERE Item_Fat_Content = 'Low Fat'

```
Total_Sales_Millions
---------------------------------------
0.78

(1 row affected)
```

4. Average Rating : The customer rating for items sold.

SELECT CAST(AVG(Rating) AS DECIMAL(10,2)) AS Avg_Rating FROM blinkit_data

```
Avg_Rating
---------------------------------------
3.97

(1 row affected)
```

========== GRANULAR REQUIREMENTS ==========

1. **Total Sales by Fat Content :**

select Item_Fat_Content,

cast(sum(Total_Sales) as decimal(10,2)) as Total_Sales

from blinkit_data

group by Item_Fat_Content

order by Total_Sales DESC

Item_Fat_Content	Total_Sales
Low Fat	776319.68
Regular	425361.80

2. Total Sales by Item Type :

```
select Item_Type,
        cast(sum(Total_Sales)/1000 as decimal(10,2)) as Total_Sales_Thousands,
        cast(avg(Total_Sales) as decimal (10,1)) as Avg_Sales,
        count(*) as N0_Of_Items,
        cast(avg(Rating) as decimal (10,2)) as Avg_Rating
from blinkit_data
group by Item_Type
order by Total_Sales_Thousands DESC
```

Item_Type	Total_Sales_Thousands	Avg_Sales
Fruits and Vegetables	178.12	144.6
Snack Foods	175.43	146.2
Household	135.98	149.4
Frozen Foods	118.56	138.5

3. 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(Total_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	254464.77	138685.87
3	Tier 3	306806.99	165326.03

4. **Total Sales by Outlet Establishment** :

```
select Outlet_Establishment_Year,
        cast(sum(Total_Sales) as decimal(10,2)) as Total_Sales
from blinkit_data
group by Outlet_Establishment_Year
order by Outlet_Establishment_Year asc
```

	Outlet_Establishment_Year	Total_Sales
1	1998	204522.26
2	2000	131809.02
3	2010	132113.37
4	2011	78131.56
5	2012	130476.86
6	2015	130942.78
7	2017	133103.91
8	2020	129103.96
9	2022	131477.77

5. **Percentage of Sales by Outlet size :**

```
select
        outlet_size,
        cast(sum(Total_Sales) as decimal(10,2)) as Total_Sales,
        cast((sum(Total_Sales) * 100.0 / sum(sum(Total_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	507895.73	42.27
2	Small	444794.17	37.01
3	High	248991.58	20.72

6. **Sales by Outlet Location :**

```
select Outlet_Location_Type,
        cast(sum(Total_Sales) as decimal(10,2)) as Total_Sales,
```

```
        cast(avg(Total_Sales) as decimal(10,1)) as Avg_Sales,
        count(*) as No_of_Items,
        cast(avg(Rating) as decimal(10,2) as Avg_Rating
from blinkit_data
group by Outlet_Location_Type
order by Total_Sales DESC
```

	Outlet_Location_Type	Total_Sales	Avg_Sales	No_of_Items	Avg_Rating
1	Tier 3	472133.03	140.9	3350	3.96
2	Tier 2	393150.64	141.2	2785	3.96
3	Tier 1	336397.81	140.9	2388	3.98

7. All Metrics by Outlet type :

```
select Outlet_Type,
        cast(sum(Total_Sales) as decimal(10,2)) as Total_Sales,
        cast((sum(Total_Sales) * 100.0 /  sum(sum(Total_Sales)) over())  as decimal(10,2)) as
Sales_Percentage,
        cast(avg(Total_Sales) as decimal (10,1)) as Avg_Sales,
        count(*) as No_of_Items,
        cast(avg(Rating) as decimal(10,2)) as Avg_Rating
from blinkit_data
group by Outlet_Type
order by Total_Sales DESC
```

	Outlet_Type	Total_Sales	Sales_Percentage	Avg_Sales	No_of_Items	Avg_Rating
1	Supermarket Type1	787549.89	65.54	141.2	5577	3.96
2	Grocery Store	151939.15	12.64	140.3	1083	3.99
3	Supermarket Type2	131477.77	10.94	141.7	928	3.97
4	Supermarket Type3	130714.67	10.88	139.8	935	3.95

Analysis of Blinkit data using Python :

Import libraries

In [15]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

Load data ¶

In [3]:
```
df=pd.read_excel(r"D:\Yess Infotech\End to End  Portfolio Projects\Blinkit D
```

In [4]:
```
df
```

Out[4]:

	Item Fat Content	Item Identifier	Item Type	Outlet Establishment Year	Outlet Identifier	Outlet Location Type	Outlet Size	Outlet Type	I Visib
0	Regular	FDX32	Fruits and Vegetables	2012	OUT049	Tier 1	Medium	Supermarket Type1	0.100
1	Low Fat	NCB42	Health and Hygiene	2022	OUT018	Tier 3	Medium	Supermarket Type2	0.008
2	Regular	FDR28	Frozen Foods	2016	OUT046	Tier 1	Small	Supermarket Type1	0.025
3	Regular	FDL50	Canned	2014	OUT013	Tier 3	High	Supermarket Type1	0.042
4	Low Fat	DRI25	Soft Drinks	2015	OUT045	Tier 2	Small	Supermarket Type1	0.033
...	...	...	...	...	...	...	...	...	...
8518	low fat	NCT53	Health and Hygiene	2018	OUT027	Tier 3	Medium	Supermarket Type3	0.000
8519	low fat	FDN09	Snack Foods	2018	OUT027	Tier 3	Medium	Supermarket Type3	0.034
8520	low fat	DRE13	Soft Drinks	2018	OUT027	Tier 3	Medium	Supermarket Type3	0.027
8521	reg	FDT50	Dairy	2018	OUT027	Tier 3	Medium	Supermarket Type3	0.107
8522	reg	FDM58	Snack Foods	2018	OUT027	Tier 3	Medium	Supermarket Type3	0.000

8523 rows × 12 columns

Data Size

In [5]:
```
df.shape
```

Out[5]:
```
(8523, 12)
```

In [6]:
```
df.tail()
```

Out[6]:

	Item Fat Content	Item Identifier	Item Type	Outlet Establishment Year	Outlet Identifier	Outlet Location Type	Outlet Size	Outlet Type	Ite Visibilit
8518	low fat	NCT53	Health and Hygiene	2018	OUT027	Tier 3	Medium	Supermarket Type3	0.00000
8519	low fat	FDN09	Snack Foods	2018	OUT027	Tier 3	Medium	Supermarket Type3	0.03470
8520	low fat	DRE13	Soft Drinks	2018	OUT027	Tier 3	Medium	Supermarket Type3	0.02757
8521	reg	FDT50	Dairy	2018	OUT027	Tier 3	Medium	Supermarket Type3	0.10771
8522	reg	FDM58	Snack Foods	2018	OUT027	Tier 3	Medium	Supermarket Type3	0.00000

In [7]:
```
df.columns
```

Out[7]:
```
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 [8]:
```
df.dtypes
```

Out[8]:
```
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 [11]:
```
print(df['Item Fat Content'].unique())
```
```
['Regular' 'Low Fat' 'low fat' 'LF' 'reg']
```

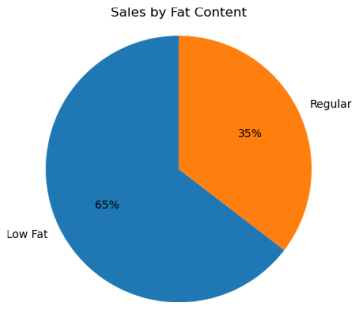
In [12]:
```
df['Item Fat Content'] = df['Item Fat Content'].replace({'LF': 'Low Fat',
                                                         'low fat' : 'Low Fat'
                                                         'reg' :'Regular'})
```

In [13]:
```
print (df['Item Fat Content'].unique())
```
```
['Regular' 'Low Fat']
```

CHARTS REQUIREMENTS

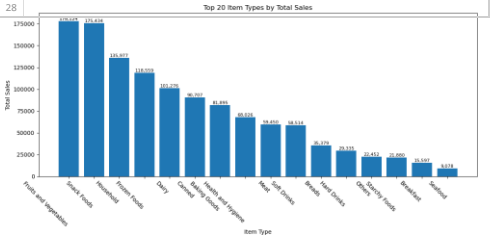
Total Sales by Fat Content

In [17]:
```
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()
```



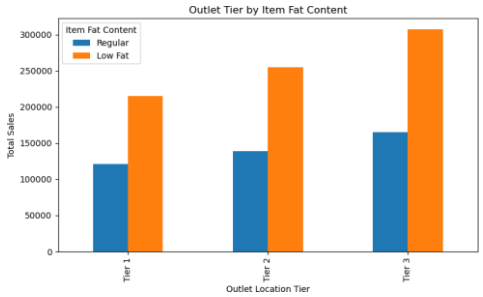
Total Sales by Item Type

In [22]:
```
sales_by_type = (
    df.groupby('Item Type')['Sales']
      .sum()
      .sort_values(ascending=False)
      .head(20)  # show only top 20
)

plt.figure(figsize=(12, 6))
bars = plt.bar(sales_by_type.index, sales_by_type.values)

plt.xticks(rotation=-45, ha='right')
plt.xlabel('Item Type')
plt.ylabel('Total Sales')
plt.title('Top 20 Item Types by Total Sales')

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()
```



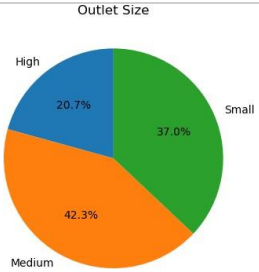
Fat Content by Outlet for Total Sales

In [23]:
```
grouped = df.groupby(['Outlet Location Type','Item Fat Content'])['Sales'].s
grouped = grouped[['Regular', 'Low Fat']]

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



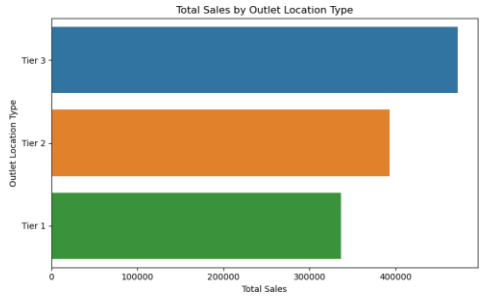
Sales by Outlet Size

In [24]:
```
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%%', starta

plt.title('Outlet Size')

plt.tight_layout()

plt.show()
```



Sales by Outlet Location

In [31]:
```
sales_by_location = df.groupby('Outlet Location Type') ['Sales'].sum().reset

sales_by_location =  sales_by_location.sort_values('Sales', ascending=False)

plt.figure(figsize=(8, 5)) # 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() # Ensures Layout fits without scroll

plt.show()
```



Total Sales by Outlet Establishment:

In [33]:
```
sales_by_year = df.groupby('Outlet Establishment Year') ['Sales'].sum().sort

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()
```

Analysis of Blinkit data using PowerBI :

Chart's Requirements

1. Total Sales by Fat Content:

Objective: Analyze the impact of fat content on total sales

Additional KPI Metrics: Assess how other KPIs (Average Sales, Number of Items, Average Rating) vary with fat content.

Chart Type: Donut Chart.

2. Total Sales by Item Type:

Objective: Identify the performance of different item types in terms of total sales.

Additional KPI Metrics: Assess how other KPIs (Average Sales, Number of Items, Average Rating) vary with fat content.

Chart Type: Bar Chart.

3. Fat Content by Outlet for Total Sales:

Objective: Compare total sales across different outlets segmented by fat content.

Additional KPI Metrics: Assess how other KPIs (Average Sales, Number of Items, Average Rating) vary with fat content.

Chart Type: Stacked Column Chart.

4. Total Sales by Outlet Establishment:

Objective: Evaluate how the age or type of outlet establishment influences total sales.

Chart Type: Line Chart.

5. Sales by Outlet Size:

Objective: Analyze the correlation between outlet size and total sales.

Chart Type:Donut/ Pie Chart.

6. Sales by Outlet Location:

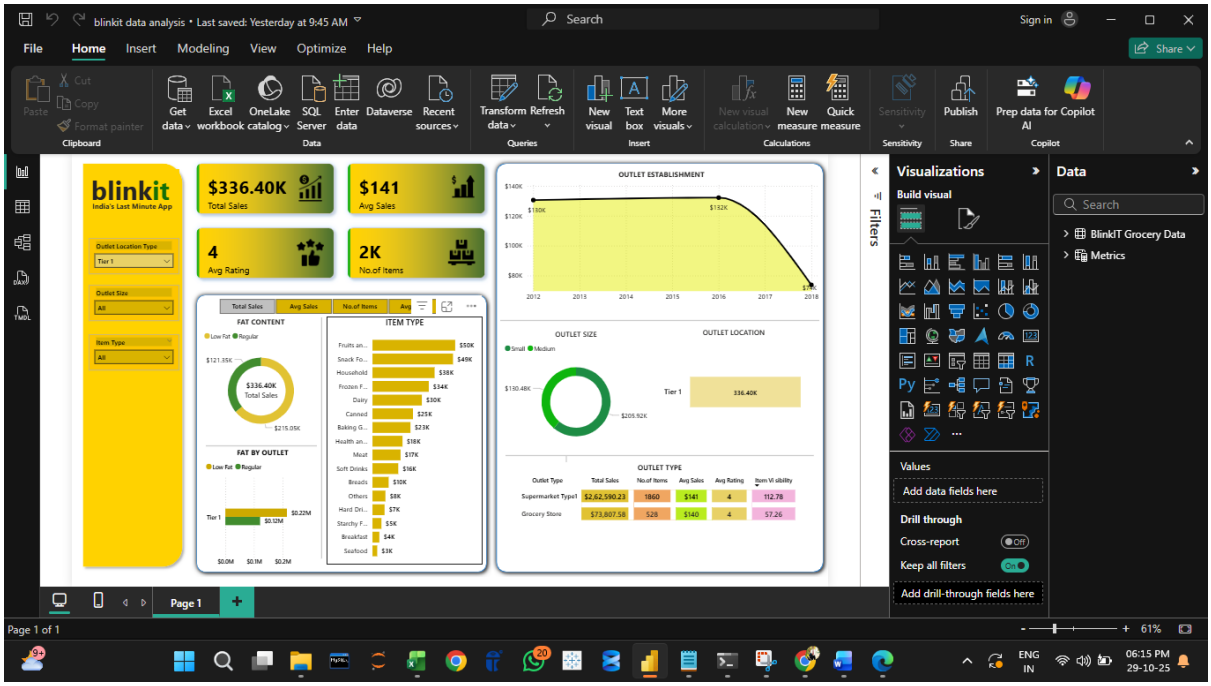
Objective: Assess the geographic distribution of sales across different locations.

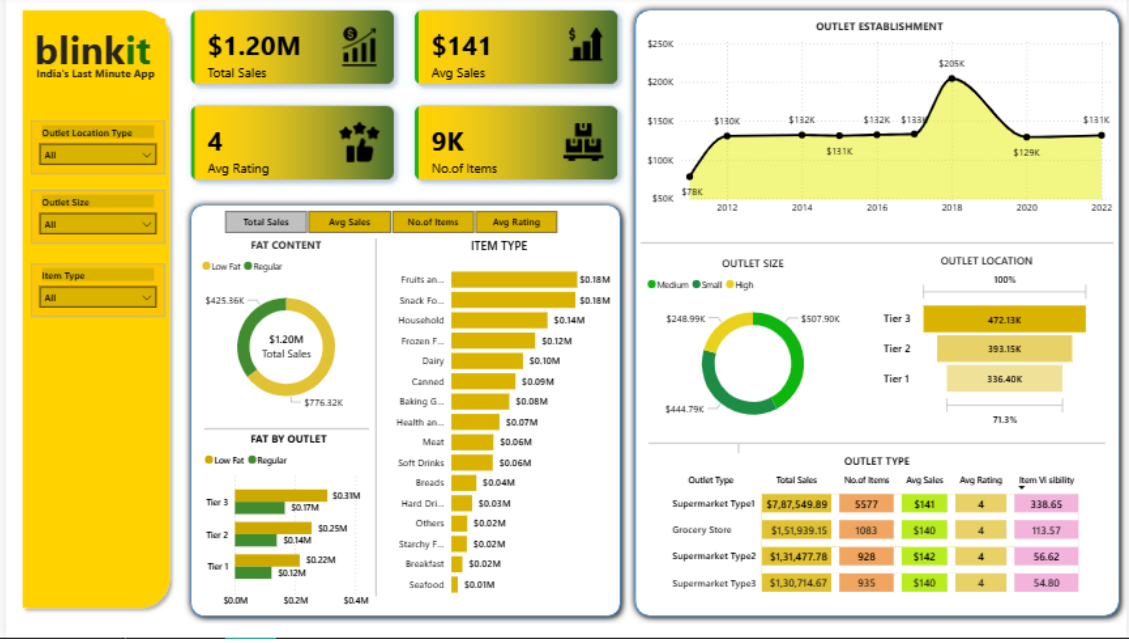
Chart Type: Funnel Map.

7. All Metrics by Outlet Type:

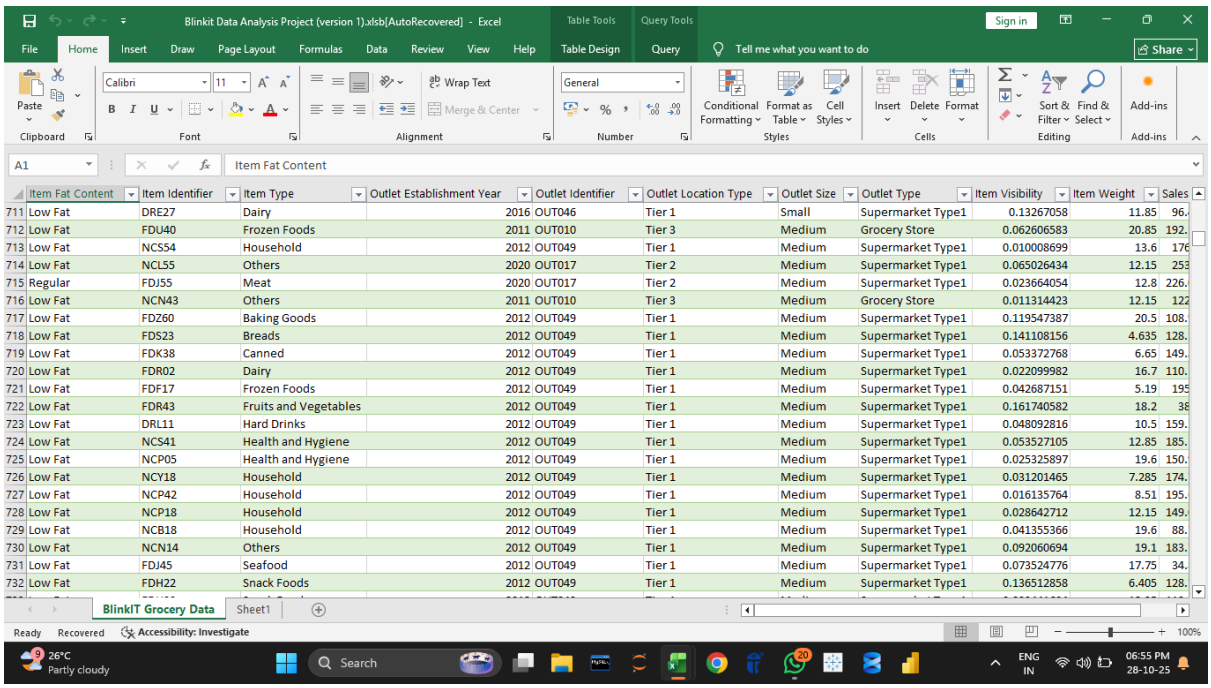
Objective: Provide a comprehensive view of all key metrics (Total Sales, Average Sales, Number of Items, Average Rating) broken down by different outlet types.

Chart Type: Matrix Card.


blinkit data analysis • Last saved: Yesterday at 9:45 AM
Search
Sign in
File
Home
Insert
Modeling
View
Optimize
Help
Share
Get data
Excel workbook
OneLake catalog
SQL Server
Enter data
Dataverse
Recent sources
Transform data
Refresh
New visual
Text box
More visuals
New measure
Quick measure
Sensitivity
Publish
Prep data for AI
Copilot
blinkit
India's Last Minute App
$336.40K
Total Sales
$141
Avg Sales
4
Avg Rating
2K
No.of Items
OUTLET ESTABLISHMENT
FAT CONTENT
ITEM TYPE
FAT BY OUTLET
OUTLET SIZE
OUTLET LOCATION
OUTLET TYPE
Visualizations
Build visual
Data
Search
BlinkIT Grocery Data
Metrics
Values
Add data fields here
Drill through
Cross-report
Keep all filters
Add drill-through fields here
Filters
Page 1
Page 1 of 1
61%
ENG
IN
06:15 PM
29-10-25


blinkit
India's Last Minute App
$1.20M
Total Sales
$141
Avg Sales
4
Avg Rating
9K
No.of Items
Outlet Location Type
Outlet Size
Item Type
All
Total Sales
Avg Sales
No.of Items
Avg Rating
FAT CONTENT
ITEM TYPE
FAT BY OUTLET
OUTLET ESTABLISHMENT
OUTLET SIZE
OUTLET LOCATION
OUTLET TYPE
Outlet Type
Total Sales
No.of Items
Avg Sales
Avg Rating
Item Visibility
Supermarket Type1
$7,87,549.89
5577
$141
4
338.65
Grocery Store
$1,51,939.15
1083
$140
4
113.57
Supermarket Type2
$1,31,477.78
928
$142
4
56.62
Supermarket Type3
$1,30,714.67
935
$140
4
54.80

Analysis of Blinkit data using Excel :



	Item Fat Content	Item Identifier	Item Type	Outlet Establishment Year	Outlet Identifier	Outlet Location Type	Outlet Size	Outlet Type	Item Visibility	Item Weight	Sales
711	Low Fat	DRE27	Dairy	2016	OUT046	Tier 1	Small	Supermarket Type1	0.13267058	11.85	96.
712	Low Fat	FDU40	Frozen Foods	2011	OUT010	Tier 3	Medium	Grocery Store	0.062606583	20.85	192.
713	Low Fat	NCS54	Household	2012	OUT049	Tier 1	Medium	Supermarket Type1	0.010008699	13.6	176
714	Low Fat	NCL55	Others	2020	OUT017	Tier 2	Medium	Supermarket Type1	0.065026434	12.15	253
715	Regular	FDJ55	Meat	2020	OUT017	Tier 2	Medium	Supermarket Type1	0.023664054	12.8	226.
716	Low Fat	NCN43	Others	2011	OUT010	Tier 3	Medium	Grocery Store	0.011314423	12.15	122
717	Low Fat	FDZ60	Baking Goods	2012	OUT049	Tier 1	Medium	Supermarket Type1	0.119547387	20.5	108.
718	Low Fat	FDS23	Breads	2012	OUT049	Tier 1	Medium	Supermarket Type1	0.141108156	4.635	128.
719	Low Fat	FDK38	Canned	2012	OUT049	Tier 1	Medium	Supermarket Type1	0.053372768	6.65	149.
720	Low Fat	FDR02	Dairy	2012	OUT049	Tier 1	Medium	Supermarket Type1	0.022099982	16.7	110.
721	Low Fat	FDF17	Frozen Foods	2012	OUT049	Tier 1	Medium	Supermarket Type1	0.042687151	5.19	195
722	Low Fat	FDR43	Fruits and Vegetables	2012	OUT049	Tier 1	Medium	Supermarket Type1	0.161740582	18.2	38
723	Low Fat	DRL11	Hard Drinks	2012	OUT049	Tier 1	Medium	Supermarket Type1	0.048092816	10.5	159.
724	Low Fat	NCS41	Health and Hygiene	2012	OUT049	Tier 1	Medium	Supermarket Type1	0.053527105	12.85	185.
725	Low Fat	NCP05	Health and Hygiene	2012	OUT049	Tier 1	Medium	Supermarket Type1	0.025325897	19.6	150.
726	Low Fat	NCY18	Household	2012	OUT049	Tier 1	Medium	Supermarket Type1	0.031201465	7.285	174.
727	Low Fat	NCP42	Household	2012	OUT049	Tier 1	Medium	Supermarket Type1	0.016135764	8.51	195.
728	Low Fat	NCP18	Household	2012	OUT049	Tier 1	Medium	Supermarket Type1	0.028642712	12.15	149.
729	Low Fat	NCB18	Household	2012	OUT049	Tier 1	Medium	Supermarket Type1	0.041355366	19.6	88.
730	Low Fat	NCN14	Others	2012	OUT049	Tier 1	Medium	Supermarket Type1	0.092060694	19.1	183.
731	Low Fat	FDJ45	Seafood	2012	OUT049	Tier 1	Medium	Supermarket Type1	0.073524776	17.75	34.
732	Low Fat	FDH22	Snack Foods	2012	OUT049	Tier 1	Medium	Supermarket Type1	0.136512858	6.405	128.