```
import pandas as pd
from plotly import graph_objects as go
import plotly.express as px
```

```
blinkit_df = pd.read_excel('/content/BlinkIT Grocery Data.xlsx')
blinkit_df.head()
```

	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
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

Next steps: Generate code with blinkit_df | View recommended plots | New interactive sheet

```
blinkit_df.tail()
```

	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
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	2018	OUT027	Tier 3	Medium	Supermarket	0.000000	NaN	112.2544	4.0

```
blinkit_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8523 entries, 0 to 8522
Data columns (total 12 columns):
 #   Column                     Non-Null Count  Dtype  
---  ------                     --------------  -----  
 0   Item Fat Content           8523 non-null   object 
 1   Item Identifier            8523 non-null   object 
 2   Item Type                  8523 non-null   object 
 3   Outlet Establishment Year  8523 non-null   int64  
 4   Outlet Identifier          8523 non-null   object 
 5   Outlet Location Type       8523 non-null   object 
 6   Outlet Size                8523 non-null   object 
 7   Outlet Type                8523 non-null   object 
 8   Item Visibility            8523 non-null   float64
 9   Item Weight                7060 non-null   float64
 10  Sales                      8523 non-null   float64
 11  Rating                     8523 non-null   float64
dtypes: float64(4), int64(1), object(7)
memory usage: 799.2+ KB
```

```
blinkit_df.describe()
```

	Outlet Establishment Year	Item Visibility	Item Weight	Sales	Rating
count	8523.000000	8523.000000	7060.000000	8523.000000	8523.000000
mean	2016.450546	0.066132	12.857645	140.992783	3.965857
std	3.189396	0.051598	4.643456	62.275067	0.605651
min	2011.000000	0.000000	4.555000	31.290000	1.000000
25%	2014.000000	0.026989	8.773750	93.826500	4.000000
50%	2016.000000	0.053931	12.600000	143.012800	4.000000
75%	2018.000000	0.094585	16.850000	185.643700	4.200000
max	2022.000000	0.328391	21.350000	266.888400	5.000000

```
blinkit_df.describe(include='object')
```

	Item Fat Content	Item Identifier	Item Type	Outlet Identifier	Outlet Location Type	Outlet Size	Outlet Type
count	8523	8523	8523	8523	8523	8523	8523
unique	5	1559	16	10	3	3	4
top	Low Fat	FDW13	Fruits and Vegetables	OUT027	Tier 3	Medium	Supermarket Type1

```
blinkit_df['Item Fat Content'] = blinkit_df['Item Fat Content'].replace(['LF','low fat','reg'],['Low Fat','Low Fat','Regular'])
```

```
blinkit_df['Item Fat Content'].value_counts()
```

	count
Item Fat Content	
Low Fat	5517
Regular	3006

dtype: int64

```
Total_Sales = blinkit_df['Sales'].sum()
print(f'Rs.{Total_Sales.round()}')
```

```
Rs.1201681.0
```

```
Average_Sales = blinkit_df['Sales'].mean()
print(f'Rs.{Average_Sales.round()}')
```

```
Rs.141.0
```

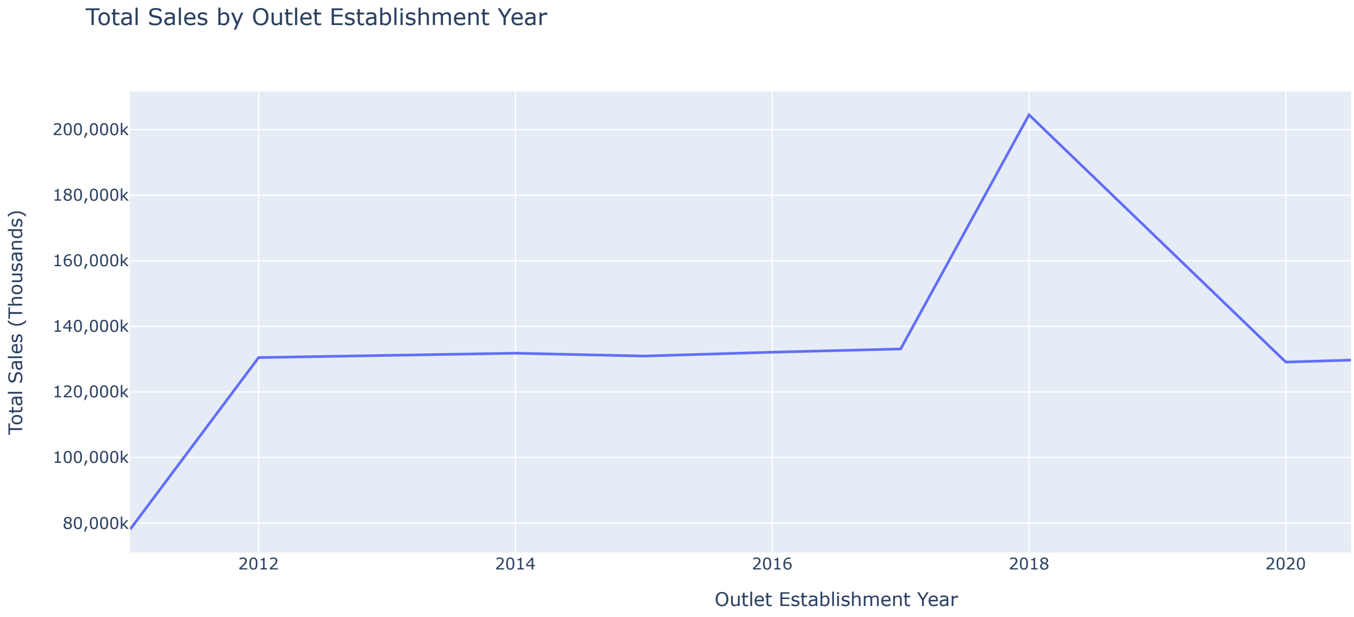
```
No_of_Items = blinkit_df['Item Identifier'].count()
print( 'Number of items ',No_of_Items)
```

```
Number of items  8523
```

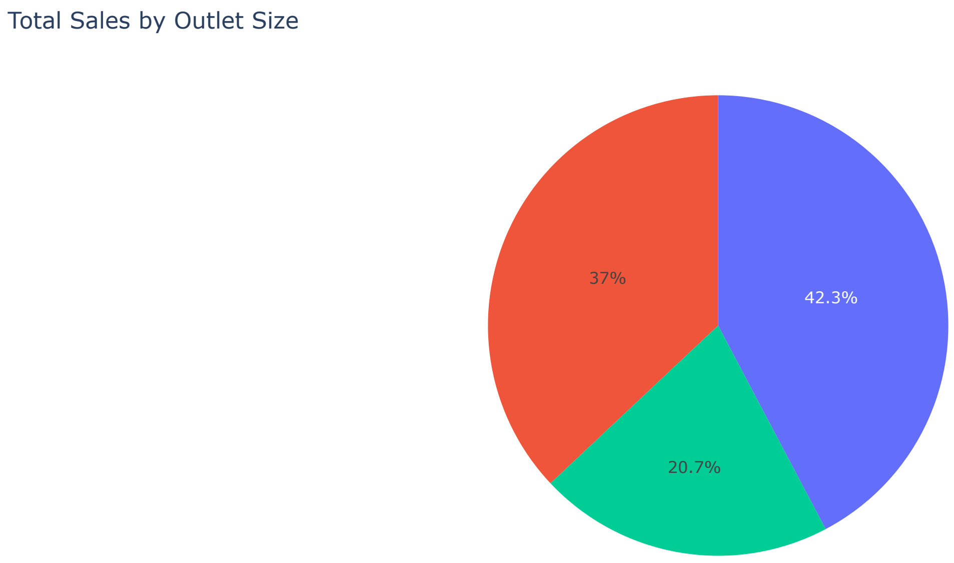
```
Average_ratings = blinkit_df['Rating'].mean()
print(f'{Average_ratings.round()}%')
```

```
4.0%
```

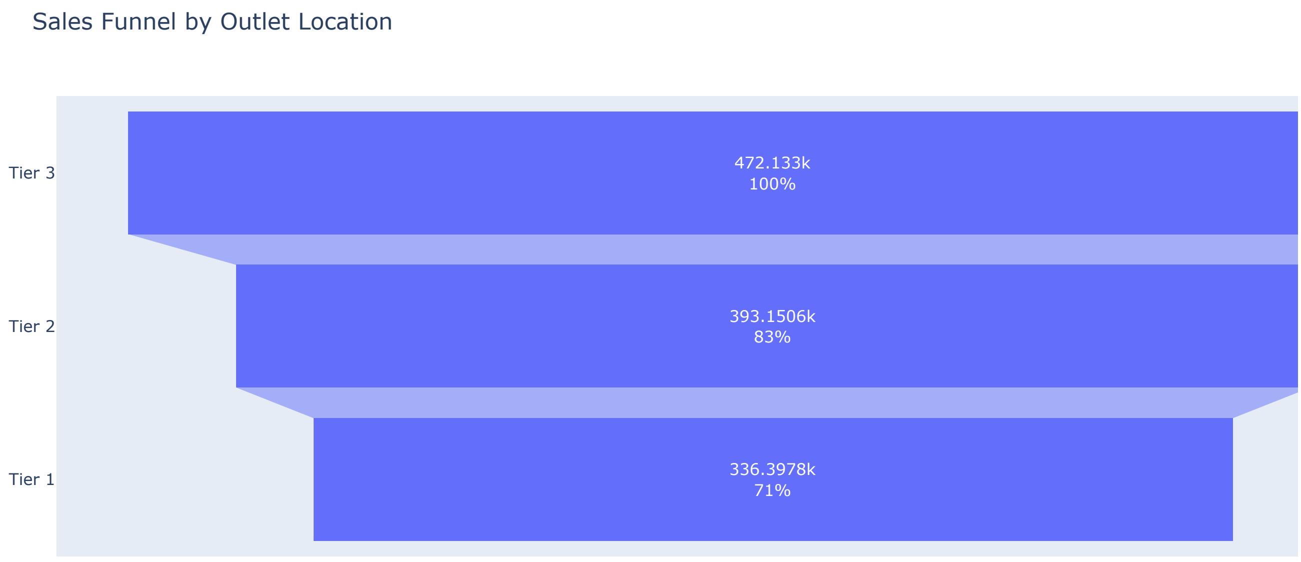
```
sales_by_outlet_year = blinkit_df.groupby('Outlet Establishment Year')['Sales'].sum().reset_index()
fig = px.line(sales_by_outlet_year, x='Outlet Establishment Year', y='Sales', title='Total Sales by Outlet Establishment Year')
fig.update_yaxes(title_text='Total Sales (Thousands)', tickformat=',.0f', ticksuffix='k')
fig.show()
```



```
sales_by_outlet_size = blinkit_df.groupby('Outlet Size')['Sales'].sum().reset_index()
fig = px.pie(sales_by_outlet_size, values='Sales', names='Outlet Size',
             title='Total Sales by Outlet Size')
fig.show()
```



```
sales_by_outlet_location = blinkit_df.groupby('Outlet Location Type')['Sales'].sum().sort_values(ascending=False)
stages = sales_by_outlet_location.index.tolist()
values = sales_by_outlet_location.values.tolist()
fig = go.Figure(go.Funnel(
    y = stages,
    x = values,
    textinfo = "value+percent initial"))
fig.update_layout(title='Sales Funnel by Outlet Location')
fig.show()
```


Sales Funnel by Outlet Location

```
sales_by_outlet_type = blinkit_df.groupby('Outlet Type').agg(
    Total_Sales=('Sales', 'sum'),
    Number_of_Items=('Item Identifier', 'count'),
    Average_Sales=('Sales', 'mean'),
    Average_Rating=('Rating', 'mean')
).reset_index()
sales_by_outlet_type.sort_values(by='Total_Sales', ascending=False, inplace=True)
sales_by_outlet_type
```

	Outlet Type	Total_Sales	Number_of_Items	Average_Sales	Average_Rating
1	Supermarket Type1	787549.8928	5577	141.213895	3.963242
0	Grocery Store	151939.1490	1083	140.294690	3.985873
2	Supermarket Type2	131477.7764	928	141.678638	3.971228
3	Supermarket Type3	130714.6746	935	139.801791	3.952941

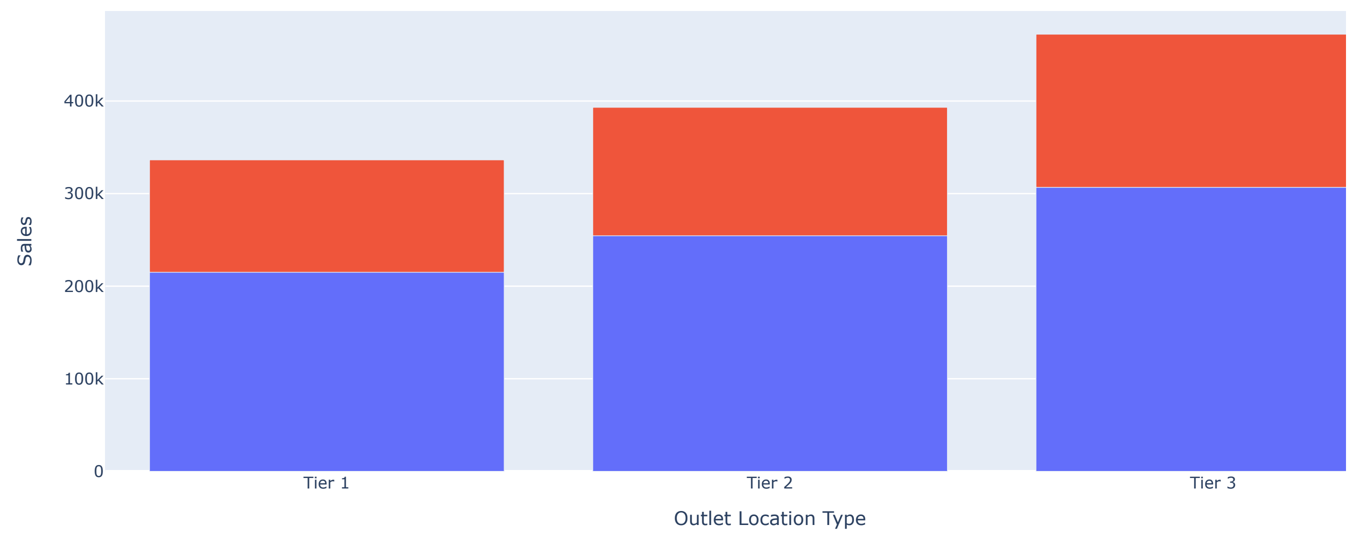
Next steps: Generate code with `sales_by_outlet_type` View recommended plots New interactive sheet

```
total_sales_by_fat_content = blinkit_df.groupby('Item Fat Content')['Sales'].sum().reset_index()
fig = px.pie(total_sales_by_fat_content, values='Sales', names='Item Fat Content',
             title='Total Sales by Item Fat Content')
fig.show()
```

Total Sales by Item Fat Content

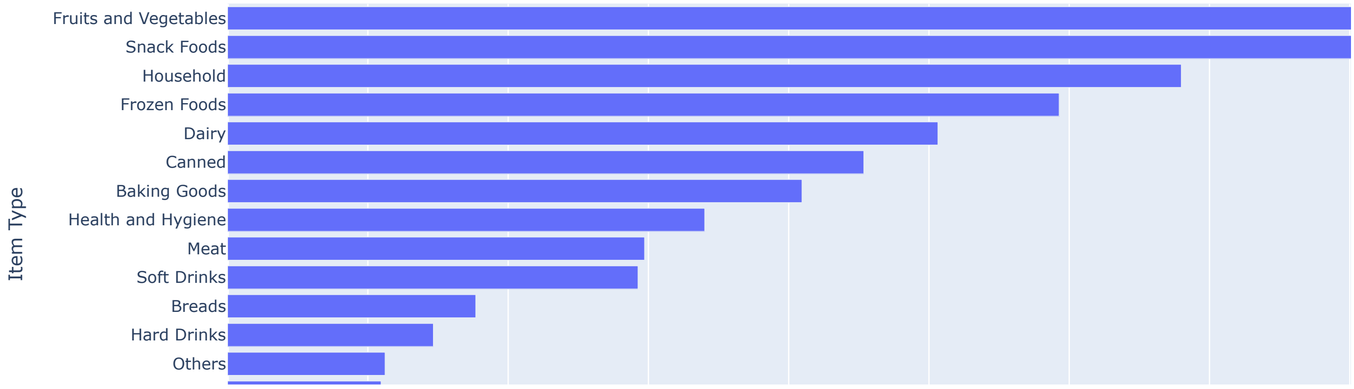
```
outlet_location_type_fat_content  = blinkit_df.groupby(['Outlet Location Type', 'Item Fat Content'])['Sales'].sum().reset_index()
fig = px.bar(outlet_location_type_fat_content, x='Outlet Location Type', y='Sales', color='Item Fat Content',
             title='Total Sales by Outlet Location Type and Item Fat Content')
fig.show()
```

Total Sales by Outlet Location Type and Item Fat Content



```
total_sales_item_type = blinkit_df.groupby('Item Type')['Sales'].sum().reset_index()
fig = px.bar(total_sales_item_type, y='Item Type', x='Sales', title='Total Sales by Item Type',category_orders= {'Item Type': sorted(tot
fig.update_layout(yaxis={'categoryorder':'total ascending'})
fig.show()
```

Total Sales by Item Type


Fruits and Vegetables
Snack Foods
Household
Frozen Foods
Dairy
Canned
Baking Goods
Health and Hygiene
Meat
Soft Drinks
Breads
Hard Drinks
Others
Item Type