Bigmart Sales Data

Load the CSV files into pandas DataFrames

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

test_file_path = '/content/Test.csv'
train_file_path = '/content/Train.csv'

test_df = pd.read_csv(test_file_path)
train_df = pd.read_csv(train_file_path)

merged_df = pd.concat([test_df, train_df], ignore_index=True)

merged_df.to_csv('/content/merged_data.csv', index=False)

#print(merged_df)
merged_df.head(5)
```

	Item_Identifier	Item_Weight	Item_Fat_Content	Item_Visibility	Item_Type	Item_MRP	Outlet_Identifier	Outlet_Establishment_Year	Ou
0	FDW58	20.750	Low Fat	0.007565	Snack Foods	107.8622	OUT049	1999	
1	FDW14	8.300	reg	0.038428	Dairy	87.3198	OUT017	2007	
2	NCN55	14.600	Low Fat	0.099575	Others	241.7538	OUT010	1998	
3	FDQ58	7.315	Low Fat	0.015388	Snack Foods	155.0340	OUT017	2007	
4	FDY38	NaN	Regular	0.118599	Dairy	234.2300	OUT027	1985	

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

Analyze BigMart Sales Data

Descriptive Statistics:

```
print(merged_df.describe())
```

```
        Item_Weight  Item_Visibility      Item_MRP  Outlet_Establishment_Year  \
count  11765.000000     14204.000000  14204.000000               14204.000000
mean      12.792854         0.065953    141.004977                1997.830681
std        4.652502         0.051459     62.086938                   8.371664
min        4.555000         0.000000     31.290000                1985.000000
25%        8.710000         0.027036     94.012000                1987.000000
50%       12.600000         0.054021    142.247000                1999.000000
75%       16.750000         0.094037    185.855600                2004.000000
max       21.350000         0.328391    266.888400                2009.000000

       Item_Outlet_Sales
count        8523.000000
mean         2181.288914
std          1706.499616
min            33.290000
25%           834.247400
50%          1794.331000
75%          3101.296400
max         13086.964800
```

Data Information:

```
print(merged_df.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 14204 entries, 0 to 14203
Data columns (total 12 columns):
 #   Column                     Non-Null Count  Dtype  
---  ------                     --------------  -----  
 0   Item_Identifier            14204 non-null  object 
 1   Item_Weight                11765 non-null  float64
 2   Item_Fat_Content           14204 non-null  object 
 3   Item_Visibility            14204 non-null  float64
 4   Item_Type                  14204 non-null  object 
 5   Item_MRP                   14204 non-null  float64
 6   Outlet_Identifier          14204 non-null  object 
 7   Outlet_Establishment_Year  14204 non-null  int64  
 8   Outlet_Size                10188 non-null  object 
 9   Outlet_Location_Type       14204 non-null  object 
 10  Outlet_Type                14204 non-null  object 
 11  Item_Outlet_Sales          8523 non-null   float64
dtypes: float64(4), int64(1), object(7)
memory usage: 1.3+ MB
None
```

Explore Outlet Information:

```
print(merged_df['Outlet_Identifier'].value_counts())
print(merged_df.groupby('Outlet_Type')['Item_Outlet_Sales'].mean())
```

```
Outlet_Identifier
OUT027    1559
OUT013    1553
OUT049    1550
OUT046    1550
OUT035    1550
OUT045    1548
OUT018    1546
OUT017    1543
OUT010     925
OUT019     880
Name: count, dtype: int64
Outlet_Type
Grocery Store         339.828500
Supermarket Type1    2316.181148
Supermarket Type2    1995.498739
Supermarket Type3    3694.038558
Name: Item_Outlet_Sales, dtype: float64
```

Analyze Item Information:

```
print(merged_df['Item_Type'].value_counts())
print(merged_df.groupby('Item_Type')['Item_Outlet_Sales'].mean())
```

```
Item_Type
Fruits and Vegetables    2013
Snack Foods              1989
Household                1548
Frozen Foods             1426
Dairy                    1136
Baking Goods             1086
Canned                   1084
Health and Hygiene        858
Meat                      736
Soft Drinks               726
Breads                    416
Hard Drinks               362
Others                    280
Starchy Foods             269
Breakfast                 186
Seafood                    89
Name: count, dtype: int64
Item_Type
Baking Goods             1952.971207
```

```
Breads                   2204.132226
Breakfast                2111.808651
Canned                   2225.194904
Dairy                    2232.542597
Frozen Foods             2132.867744
Fruits and Vegetables    2289.009592
Hard Drinks              2139.221622
Health and Hygiene       2010.000265
Household                2258.784300
Meat                     2158.977911
Others                   1926.139702
Seafood                  2326.065928
Snack Foods              2277.321739
Soft Drinks              2006.511735
Starchy Foods            2374.332773
Name: Item_Outlet_Sales, dtype: float64
```

Relationship between Item Visibility and Sales:

```
print(merged_df[['Item_Visibility', 'Item_Outlet_Sales']].corr())
```

```
                   Item_Visibility  Item_Outlet_Sales
Item_Visibility           1.000000          -0.128625
Item_Outlet_Sales        -0.128625           1.000000
```

Double-click (or enter) to edit

Sales Trends Analysis

```
sales_by_year = merged_df.groupby('Outlet_Establishment_Year')['Item_Outlet_Sales'].sum()
print(sales_by_year)
```

```
Outlet_Establishment_Year
1985    3.633620e+06
1987    2.142664e+06
1997    2.118395e+06
1998    1.883402e+05
1999    2.183970e+06
2002    2.036725e+06
2004    2.268123e+06
2007    2.167465e+06
2009    1.851823e+06
Name: Item_Outlet_Sales, dtype: float64
```

Sales Trends by Outlet Type

```
sales_by_outlet_type = merged_df.groupby('Outlet_Type')['Item_Outlet_Sales'].sum()
print(sales_by_outlet_type)
```

```
Outlet_Type
Grocery Store        3.680343e+05
Supermarket Type1    1.291734e+07
Supermarket Type2    1.851823e+06
Supermarket Type3    3.453926e+06
Name: Item_Outlet_Sales, dtype: float64
```

Double-click (or enter) to edit

Sales Trends by Item Type

```
sales_by_item_type = merged_df.groupby('Item_Type')['Item_Outlet_Sales'].sum()
print(sales_by_item_type)
```

```
Item_Type
Baking Goods             1.265525e+06
Breads                   5.532372e+05
```

```
Breakfast                2.322990e+05
Canned                   1.444151e+06
Dairy                    1.522594e+06
Frozen Foods             1.825735e+06
Fruits and Vegetables    2.820060e+06
Hard Drinks              4.577934e+05
Health and Hygiene       1.045200e+06
Household                2.055494e+06
Meat                     9.175656e+05
Others                   3.255176e+05
Seafood                  1.488682e+05
Snack Foods              2.732786e+06
Soft Drinks              8.928977e+05
Starchy Foods            3.514013e+05
Name: Item_Outlet_Sales, dtype: float64
```

Explore Item Fat Content and Sales:

```
print(merged_df.groupby('Item_Fat_Content')['Item_Outlet_Sales'].mean())
```

```
Item_Fat_Content
LF         2073.551928
Low Fat    2164.477336
Regular    2235.186702
low fat    2087.740737
reg        1962.192268
Name: Item_Outlet_Sales, dtype: float64
```

Analyze Outlet Location Type and Sales:

```
print(merged_df.groupby('Outlet_Location_Type')['Item_Outlet_Sales'].mean())
```

```
Outlet_Location_Type
Tier 1    1876.909159
Tier 2    2323.990559
Tier 3    2279.627651
Name: Item_Outlet_Sales, dtype: float64
```

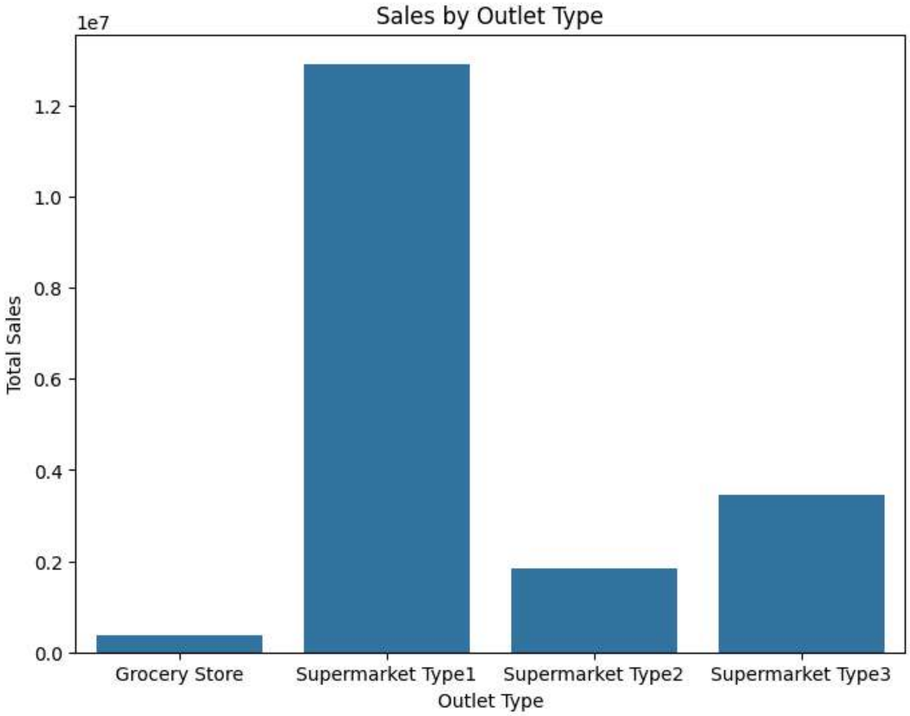
Identify Missing Values:

```
print(merged_df.isnull().sum())
```

```
Item_Identifier                 0
Item_Weight                  2439
Item_Fat_Content                0
Item_Visibility                 0
Item_Type                       0
Item_MRP                        0
Outlet_Identifier               0
Outlet_Establishment_Year       0
Outlet_Size                  4016
Outlet_Location_Type            0
Outlet_Type                     0
Item_Outlet_Sales            5681
dtype: int64
```

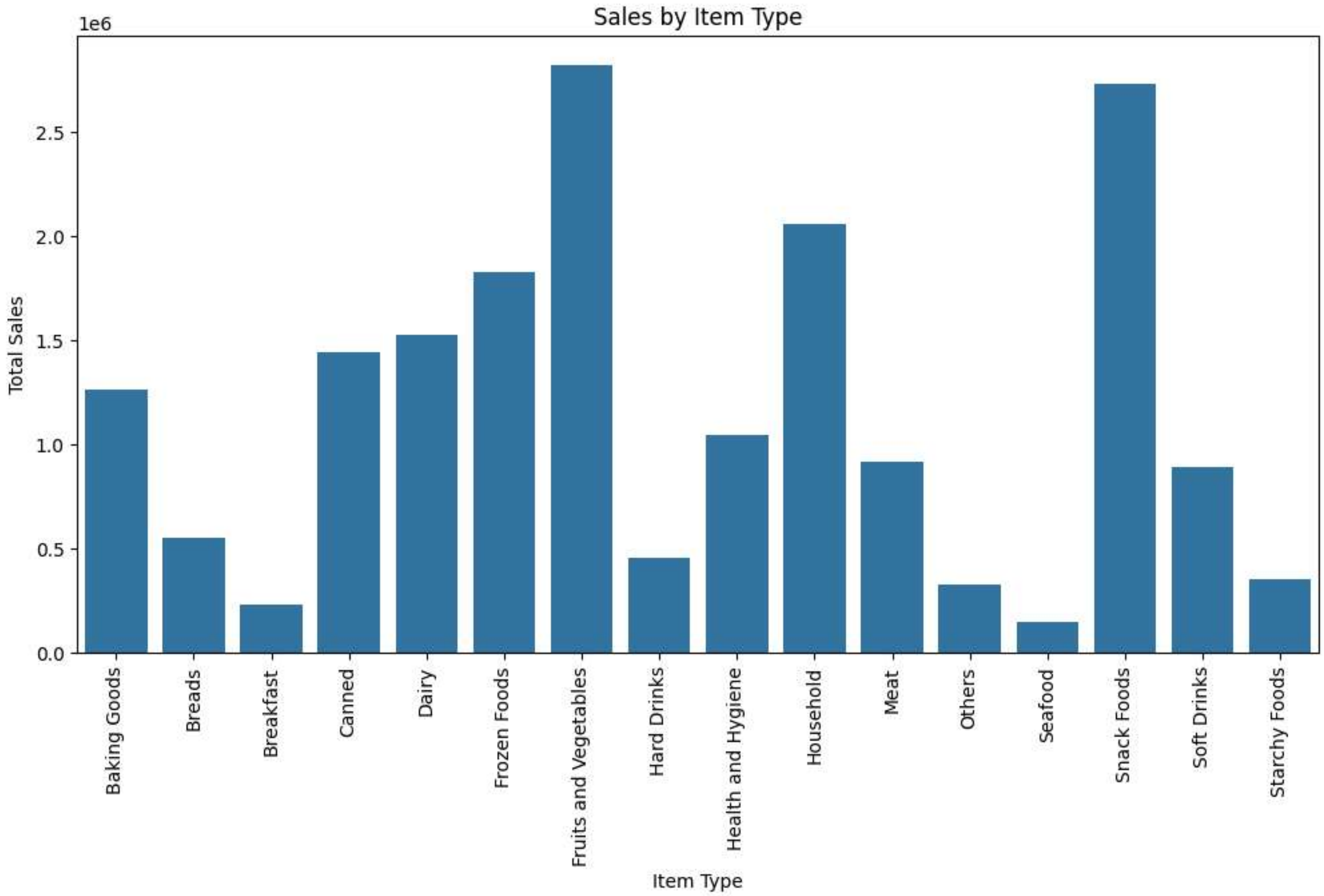
Analyze Sales by Outlet Type:

```
outlet_sales = merged_df.groupby('Outlet_Type')['Item_Outlet_Sales'].sum()
plt.figure(figsize=(8, 6))
sns.barplot(x=outlet_sales.index, y=outlet_sales.values)
plt.xlabel('Outlet Type')
plt.ylabel('Total Sales')
plt.title('Sales by Outlet Type')
plt.show()
```



Analyze Sales by Item Type:

```
item_sales = merged_df.groupby('Item_Type')['Item_Outlet_Sales'].sum()
plt.figure(figsize=(12, 6))
sns.barplot(x=item_sales.index, y=item_sales.values)
plt.xlabel('Item Type')
plt.ylabel('Total Sales')
plt.title('Sales by Item Type')
plt.xticks(rotation=90)
plt.show()
```

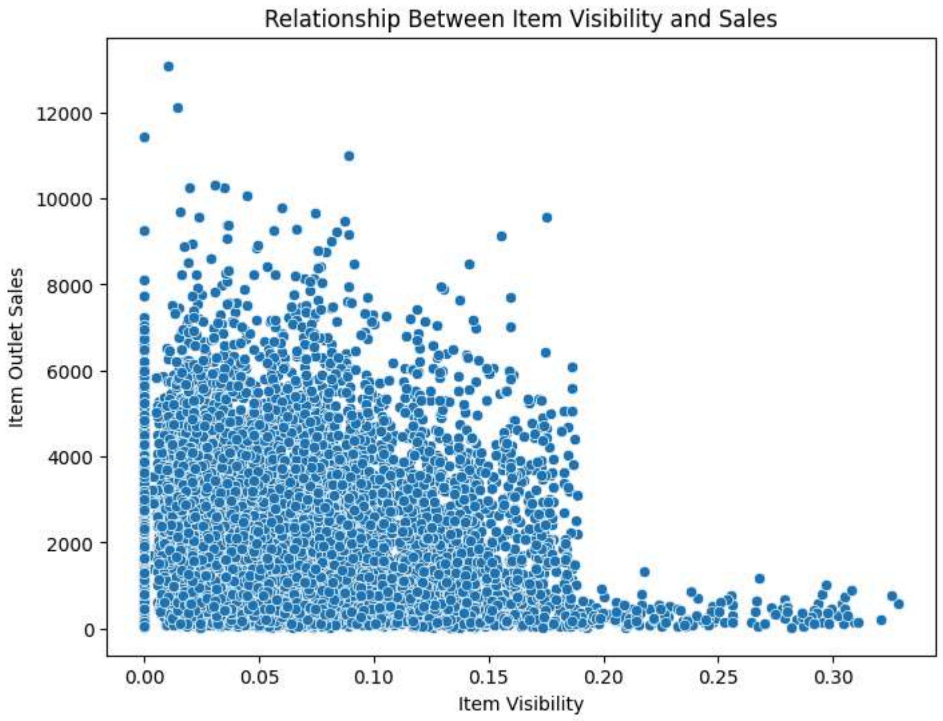


Analyze Sales by Outlet Location Type:

```
location_sales = merged_df.groupby('Outlet_Location_Type')['Item_Outlet_Sales'].sum()
plt.figure(figsize=(8, 6))
sns.barplot(x=location_sales.index, y=location_sales.values)
plt.xlabel('Outlet Location Type')
plt.ylabel('Total Sales')
plt.title('Sales by Outlet Location Type')
plt.show()
```

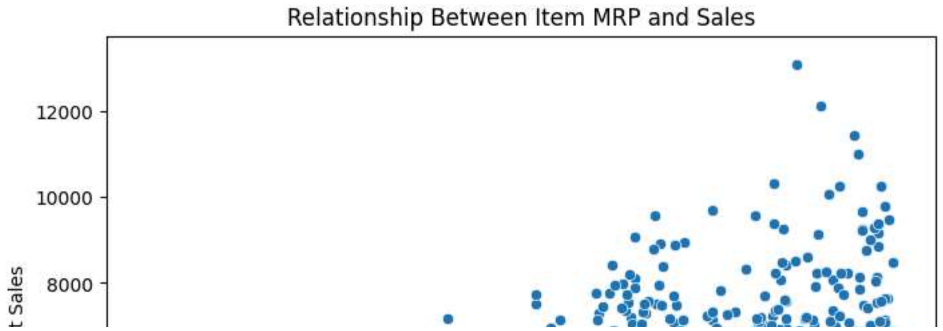
Analyze Relationship Between Item Visibility and Sales:

```
plt.figure(figsize=(8, 6))
sns.scatterplot(x='Item_Visibility', y='Item_Outlet_Sales', data=merged_df)
plt.xlabel('Item Visibility')
plt.ylabel('Item Outlet Sales')
plt.title('Relationship Between Item Visibility and Sales')
plt.show()
```



Analyze Relationship Between Item MRP and Sales:

```
plt.figure(figsize=(8, 6))
sns.scatterplot(x='Item_MRP', y='Item_Outlet_Sales', data=merged_df)
plt.xlabel('Item MRP')
plt.ylabel('Item Outlet Sales')
plt.title('Relationship Between Item MRP and Sales')
plt.show()
```


Relationship Between Item MRP and Sales
t Sales