EDA CASE STUDY ANALYSIS

Load the Dataset

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
```

```
df = pd.read_csv("BigMart.csv")
```

```
df.head()
```

	Item_Identifier	Item_Weight	Item_Fat_Content	Item_Visibility	Item_Type	Item_MRP	Outlet_Identifier	Outlet_Establishment_Year	Outlet_Size	Outlet_Location_Type
0	FDA15	9.30	Low Fat	0.016047	Dairy	249.8092	OUT049	1999.0	Medium	Tier 1
1	DRC01	5.92	Regular	0.019278	Soft Drinks	48.2692	OUT018	2009.0	Medium	Tier 3
2	FDN15	17.50	Low Fat	0.016760	Meat	141.6180	OUT049	1999.0	Medium	Tier 1
3	FDX07	19.20	Regular	0.000000	Fruits and Vegetables	182.0950	OUT010	1998.0	Medium	Tier 3
4	NCD19	8.93	Low Fat	0.000000	Household	53.8614	OUT013	1987.0	High	Tier 3

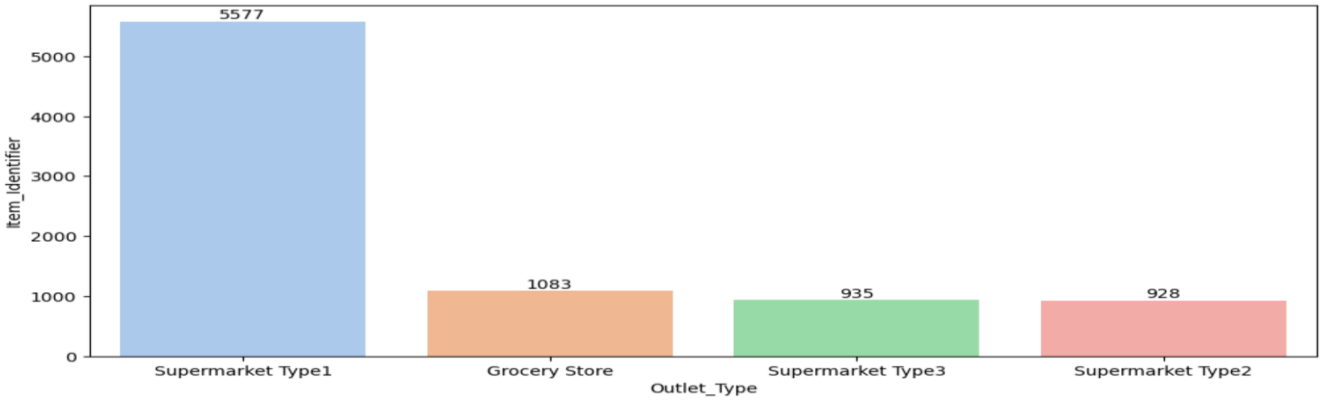
Tasks:

EDA(Exploratory Data Analysis)

```
#### How many rows and columns are there in the dataset?
rows, columns = df.shape
print("number of Rows:", rows)
print("number of columns:", columns)
```

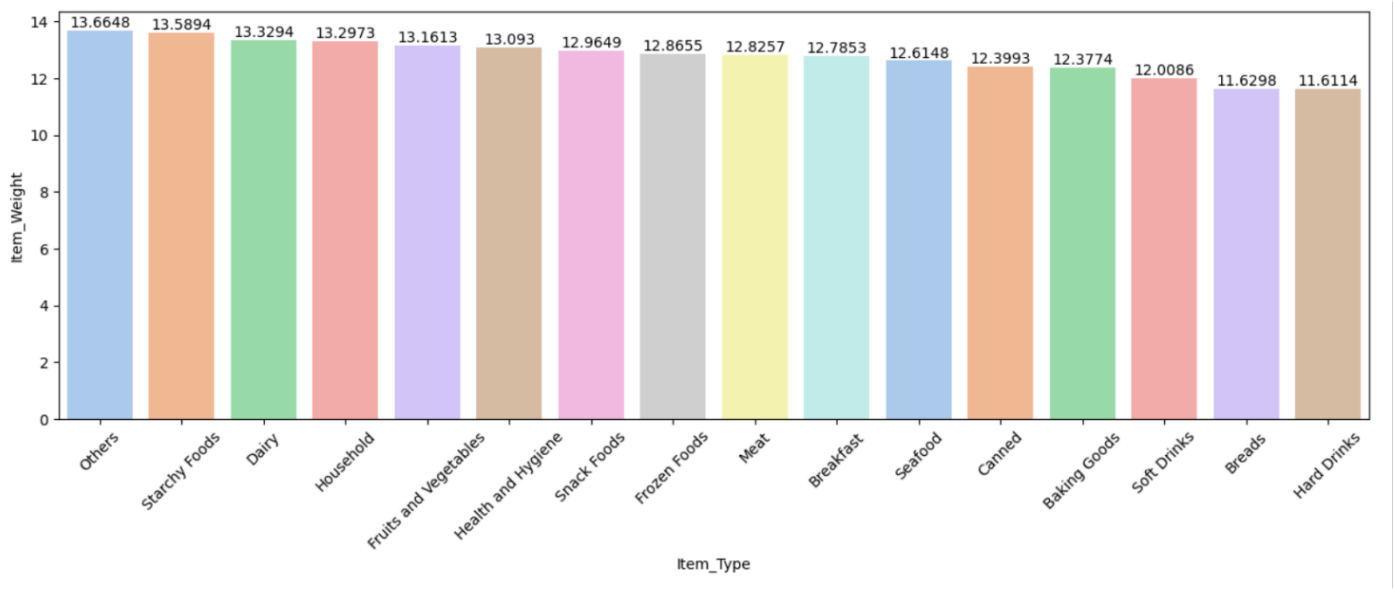
```
number of Rows: 8523
number of columns: 12
```

```
#### What are the different types of Outlet Types in the dataset, and how many records belong to each type?
outlet_record = df.groupby(['Outlet_Type'], as_index= False)['Item_Identifier'].count().sort_values(by='Item_Identifier', ascending= False)

plt.figure(figsize= (12,6))
ax= sns.barplot(x='Outlet_Type', y='Item_Identifier',data= outlet_record, hue= 'Outlet_Type', palette= 'pastel')

### this loop will help you to get the bar labels
for bar in ax.containers:
    ax.bar_label(bar)
plt.show()
```

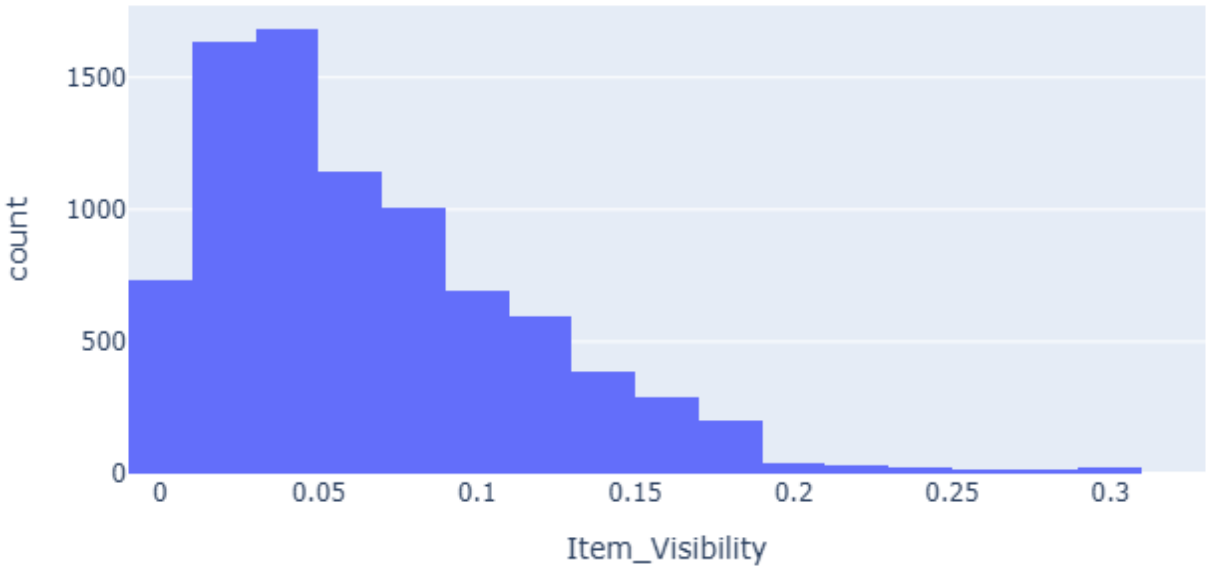


- It means Supermarket type_1 has maximum Records

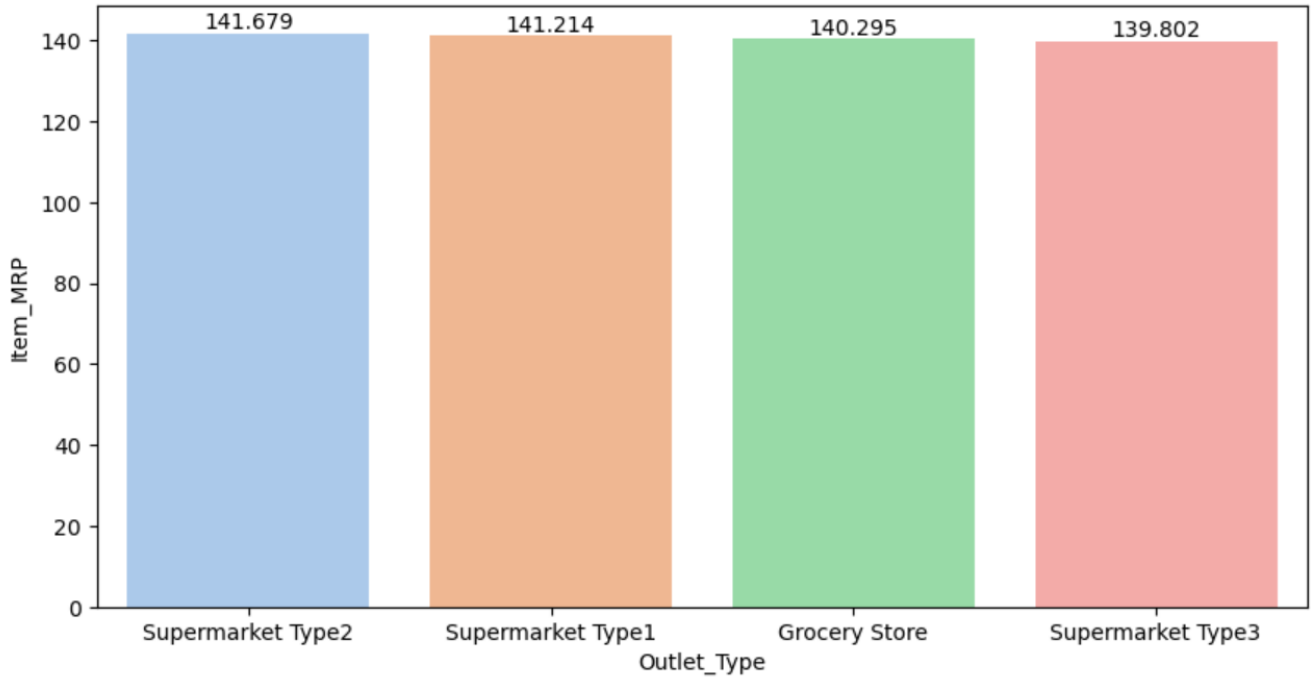
```
#### What is the average Item Weight across all products?

avg_weight = df.groupby(['Item_Type'], as_index= False)['Item_Weight'].mean().sort_values(by='Item_Weight', ascending= False)

plt.figure(figsize= (16,5))
ax= sns.barplot(x='Item_Type', y='Item_Weight',data= avg_weight, hue= 'Item_Type', palette= 'pastel')

### this loop will help you to get the bar labels
for bar in ax.containers:
    ax.bar_label(bar)
plt.xticks(rotation= 45)
plt.show()
```



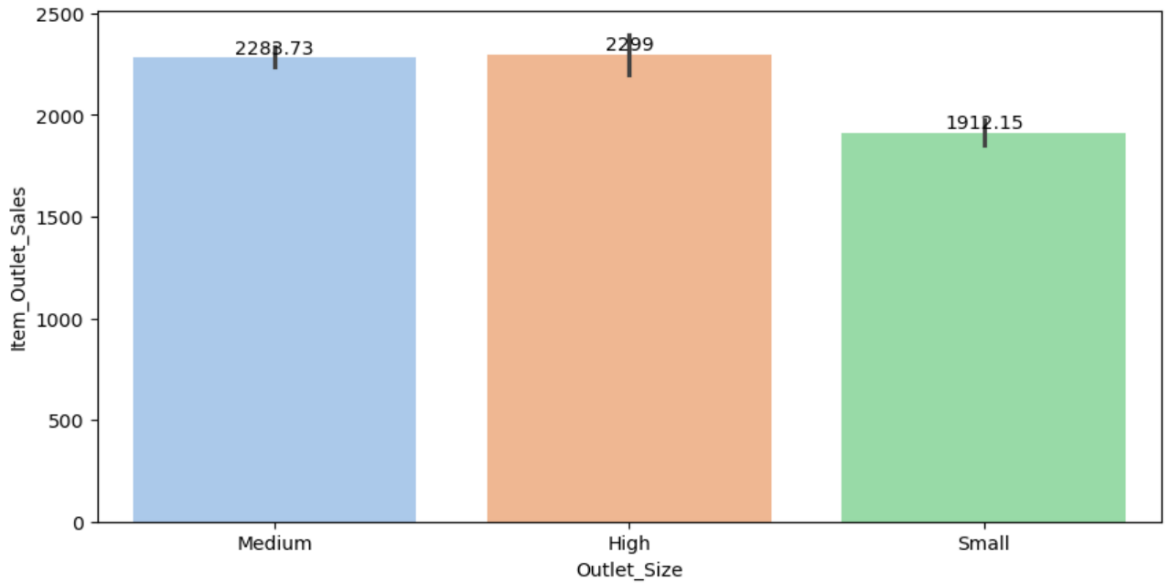

```
#### Plot a histogram of Item Visibility to visualize its distribution..

plt.figure(figsize=(10,8))
fig= px.histogram(df, x= 'Item_Visibility', nbins= 30)
fig.show()
```



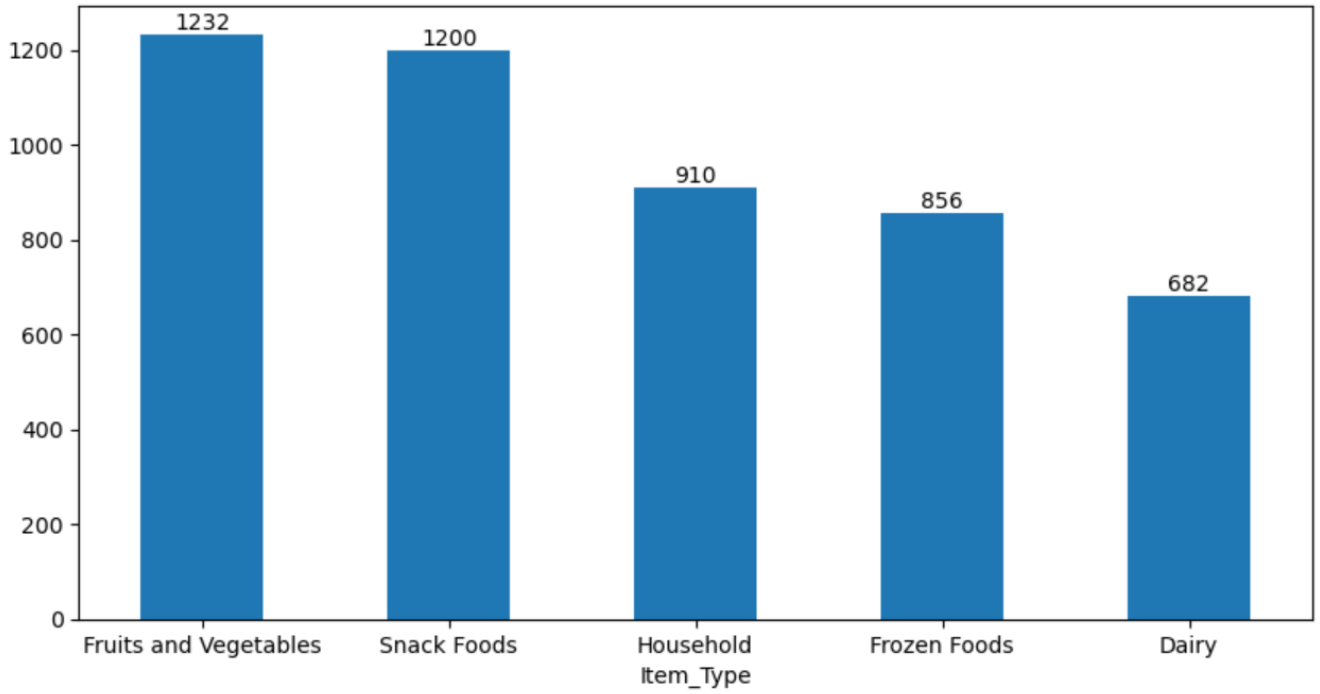
```
#### What is the average Item MRP (Maximum Retail Price) for each Outlet Type?

avg_MRP= df.groupby(['Outlet_Type'], as_index= False)['Item_MRP'].mean().sort_values(by='Item_MRP', ascending= False)

plt.figure(figsize= (10,6))
ax= sns.barplot(x='Outlet_Type', y='Item_MRP',data= avg_MRP, hue= 'Outlet_Type', palette= 'pastel')

### this loop will help you to get the bar labels
for bar in ax.containers:
    ax.bar_label(bar)
plt.show()
```



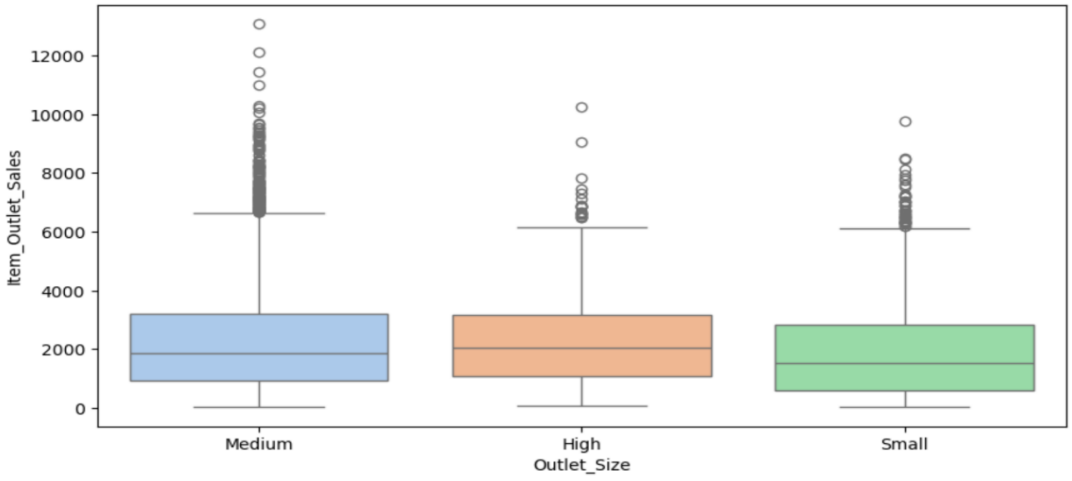

```
#### Which Outlet Size has the highest total Item Outlet Sales.
size_sales = df.groupby(['Outlet_Size'], as_index= False)['Item_Outlet_Sales'].max().sort_values(by='Item_Outlet_Sales', ascending= False)

plt.figure(figsize=(10,5))
ax= sns.barplot(x='Outlet_Size', y= 'Item_Outlet_Sales', data= df, hue= 'Outlet_Size' , palette= 'pastel')

for bar in ax.containers:
    ax.bar_label(bar)

plt.show()
```



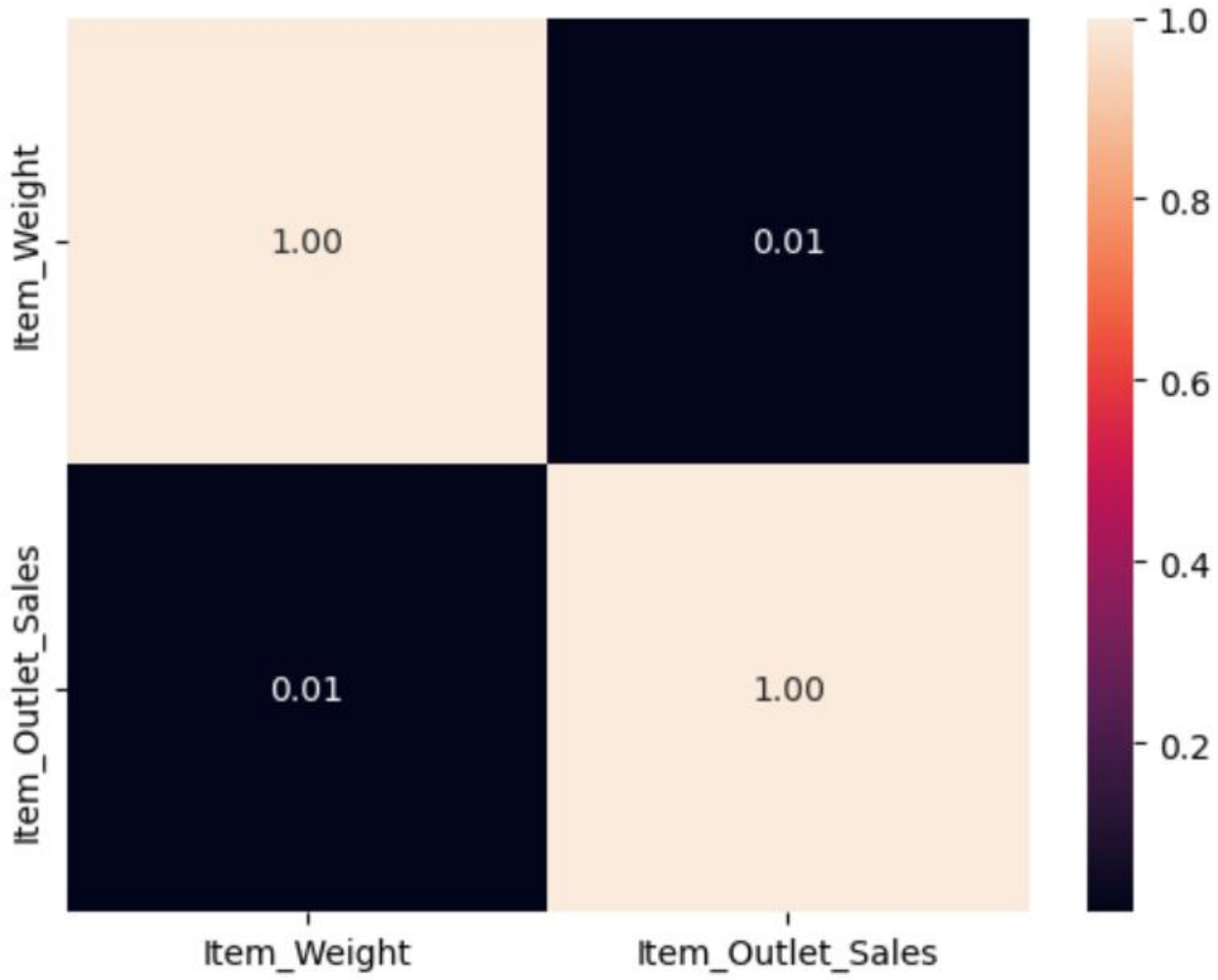
```
####  Identify and list the top 5 most common Item Types.

common_5=df['Item_Type'].value_counts().head(5)

plt.figure(figsize=(10,5))
ax= common_5.plot(kind= 'bar')

for bar in ax.containers:
    ax.bar_label(bar)

plt.xticks(rotation= 360)
plt.show()
```




```
#### Plot a box plot of Item Outlet Sales for each Outlet Size to identify potential outliers.
df['Outlet_Size'].unique()

plt.figure(figsize=(10,6))
sns.boxplot(x= 'Outlet_Size', y= 'Item_Outlet_Sales', data= df ,hue ='Outlet_Size' , palette= 'pastel')
plt.show()
```



```
#### Calculate the correlation between Item Weight and Item Outlet Sales.
cor = df[['Item_Weight','Item_Outlet_Sales']].corr(method= 'spearman')
cor
sns.heatmap(cor,annot= True, fmt= '.2f')
plt.show()
```




```
#### Create a pivot table showing the average Item Outlet Sales for each Outlet Type and Outlet Location.
pivot = df.pivot_table(
    values= 'Item_Outlet_Sales',
    index= 'Outlet_Type',
    columns= 'Outlet_Location_Type',
    aggfunc= 'mean'
)
pivot
```

Outlet_Location_Type	Tier 1	Tier 2	Tier 3
Outlet_Type			
Grocery Store	340.329723	NaN	339.351662
Supermarket Type1	2313.099451	2323.990559	2298.995256
Supermarket Type2	NaN	NaN	1995.498739
Supermarket Type3	NaN	NaN	3694.038558

```
### Visualize the pivot table


pivot.plot(kind='bar',figsize=(10,6))
plt.title('Average Item Outlet Sales by Outlet Type and Location')
plt.xticks(rotation = 360)
plt.show()
```