```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt


data = pd.read_csv('/content/Bigbazaar_sales_data.csv')
data
```

:em_Visibility	Year	Outlet_Identifier	Item_Type	Item_MRP	MRP_Level	Outlet_Size	C
0.016047	17	OUT049	Dairy	249.8092	Very_High	2	
0.019278	7	OUT018	Soft Drinks	48.2692	Low	2	
0.016760	17	OUT049	Meat	141.6180	High	2	
0.022930	18	OUT010	Fruits and Vegetables	182.0950	High	2	
0.014670	29	OUT013	Household	53.8614	Low	1	
...	...	...	...	...	...	...	
0.013496	19	OUT046	Snack Foods	141.3154	High	3	
0.142991	7	OUT018	Starchy Foods	169.1448	High	2	
0.073529	14	OUT045	Health and Hygiene	118.7440	Medium	2	
0.098200	9	OUT017	Canned	214.6218	Very_High	2	
0.104720	14	OUT045	Canned	79.7960	Medium	2	

Next steps: Generate code with `data` View recommended plots

```
print(data.head())
```

```
  Item_Identifier  Item_Weight  Item_Visibility  Year Outlet_Identifier  \
0           FDA15         9.30         0.016047    17            OUT049
1           DRC01         5.92         0.019278     7            OUT018
2           FDN15        17.50         0.016760    17            OUT049
3           FDX07        19.20         0.022930    18            OUT010
4           NCD19         8.93         0.014670    29            OUT013

              Item_Type  Item_MRP  MRP_Level  Outlet_Size  \
```

```
0                  Dairy  249.8092  Very_High            2
1            Soft Drinks   48.2692        Low            2
2                   Meat  141.6180       High            2
3  Fruits and Vegetables  182.0950       High            2
4              Household   53.8614        Low            1

  Outlet_Location_Type        Outlet_Type  Item_Outlet_Sales  Converted Sales
0               Tier 1  Supermarket Type1          3735.1380            2.428
1               Tier 3  Supermarket Type2           443.4228            0.288
2               Tier 1  Supermarket Type1          2097.2700            1.363
3               Tier 3      Grocery Store           732.3800            0.476
4               Tier 3  Supermarket Type1           994.7052            0.647
```

```
column_count = data.count()
print(column_count)
```

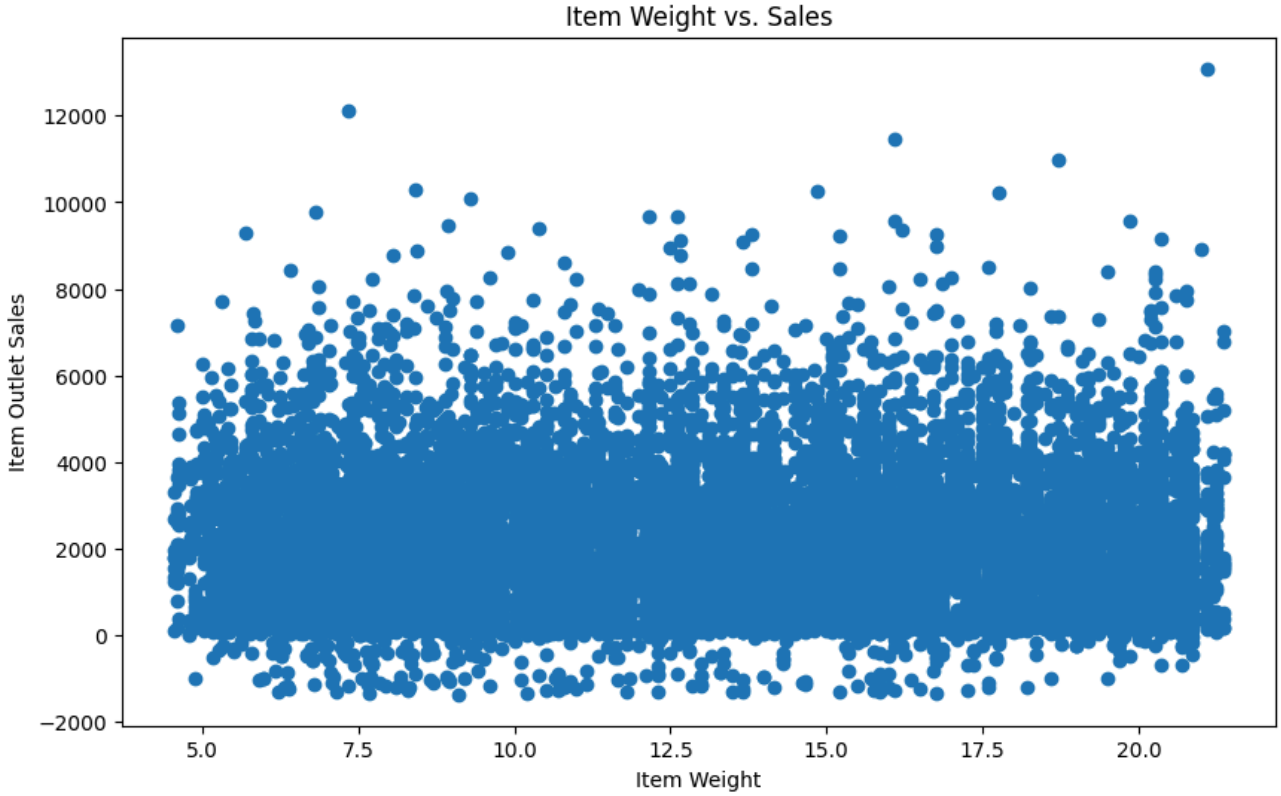
```
Item_Identifier         14204
Item_Weight             14204
Item_Visibility         14204
Year                    14204
Outlet_Identifier       14204
Item_Type               14204
Item_MRP                14204
MRP_Level               14204
Outlet_Size             14204
Outlet_Location_Type    14204
Outlet_Type             14204
Item_Outlet_Sales       14204
Converted Sales         14204
dtype: int64
```

```
group_data = data.groupby('Item_Type').size() #how many item sells
print(group_data)
```

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

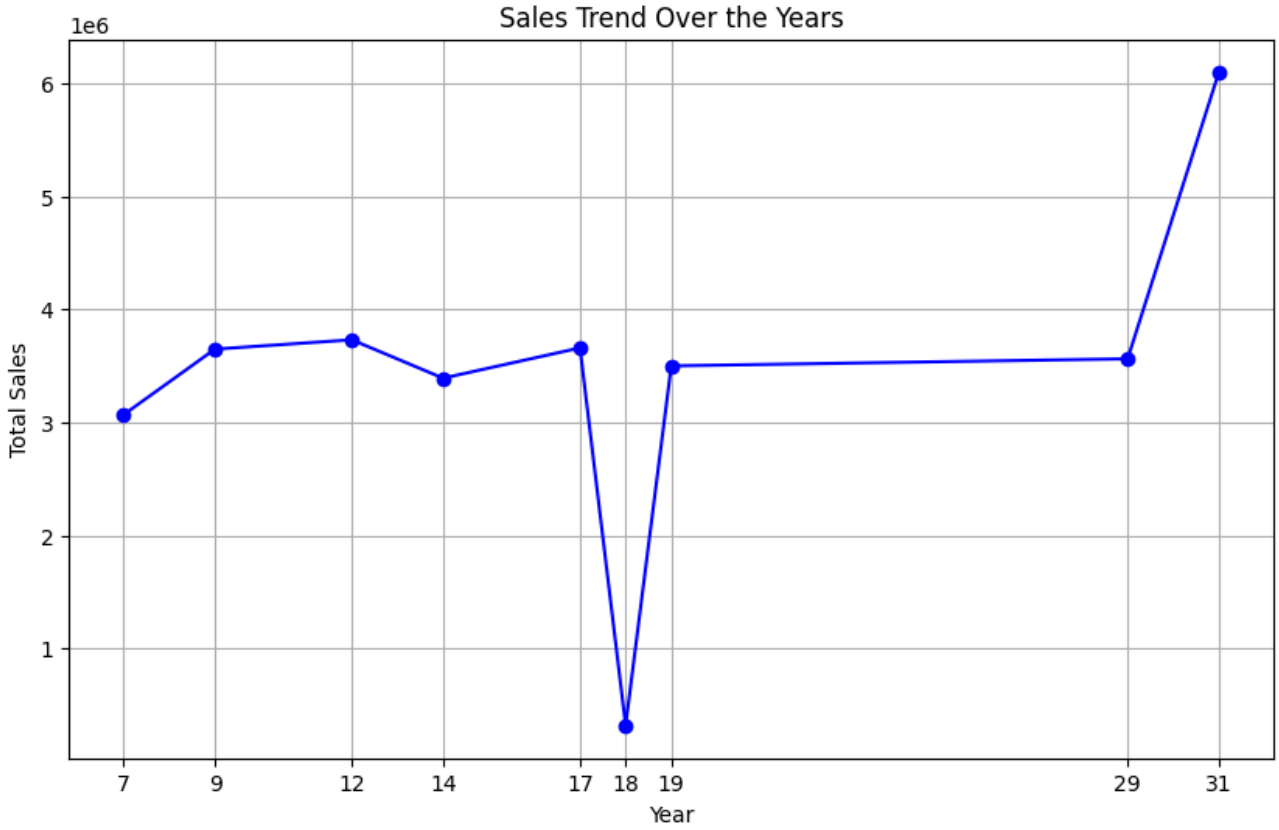
```
#Is there any correlation between the weight of the items and their sales? Does heavier i
correlation = data['Item_Weight'].corr(data['Item_Outlet_Sales'])
print("correlation between the weight of the items and their sales",correlation)
plt.figure(figsize=(10,6))
plt.scatter(data['Item_Weight'], data['Item_Outlet_Sales'])
plt.title('Item Weight vs. Sales')
plt.xlabel('Item Weight')
plt.ylabel('Item Outlet Sales')
plt.show()

if correlation > 0:
    print("There is positive correlation between weight and sales.")
elif correlation < 0:
    print("There is Negative correlation between weight and sales.")
else:
  print("There is No correlation between weight and sales.")
```

```
correlation between the weight of the items and their sales 0.023162408761173252
```



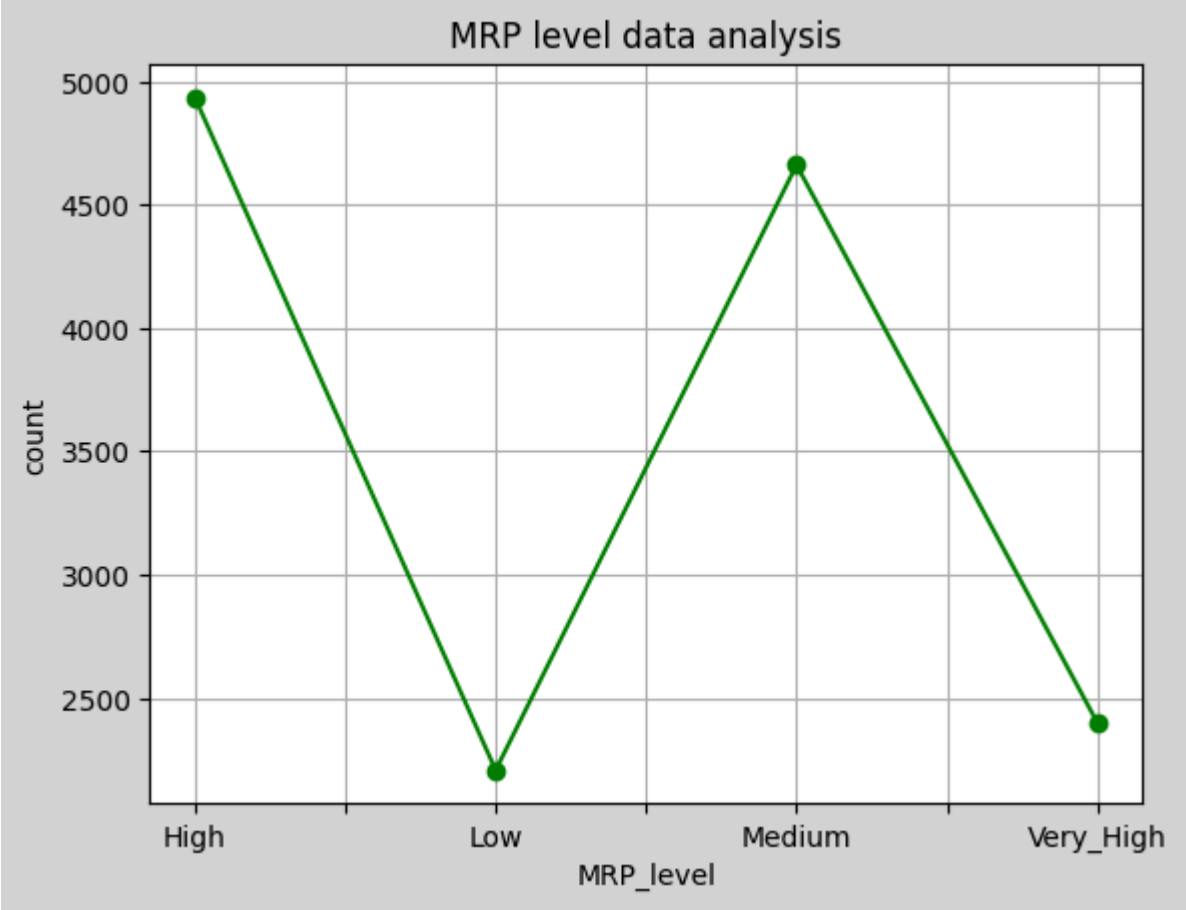

```
There is positive correlation between weight and sales.
```

```
yearly_sales = data.groupby('Year')['Item_Outlet_Sales'].sum()
# Step 2: Plot the sales trend over the years
plt.figure(figsize=(10, 6))
yearly_sales.plot(kind='line', marker='o', color='b', linestyle='-')
plt.title('Sales Trend Over the Years')
plt.xlabel('Year')
plt.ylabel('Total Sales')
plt.grid(True)
plt.xticks(yearly_sales.index)  # Set x-axis ticks to the years in the data
plt.show()
```

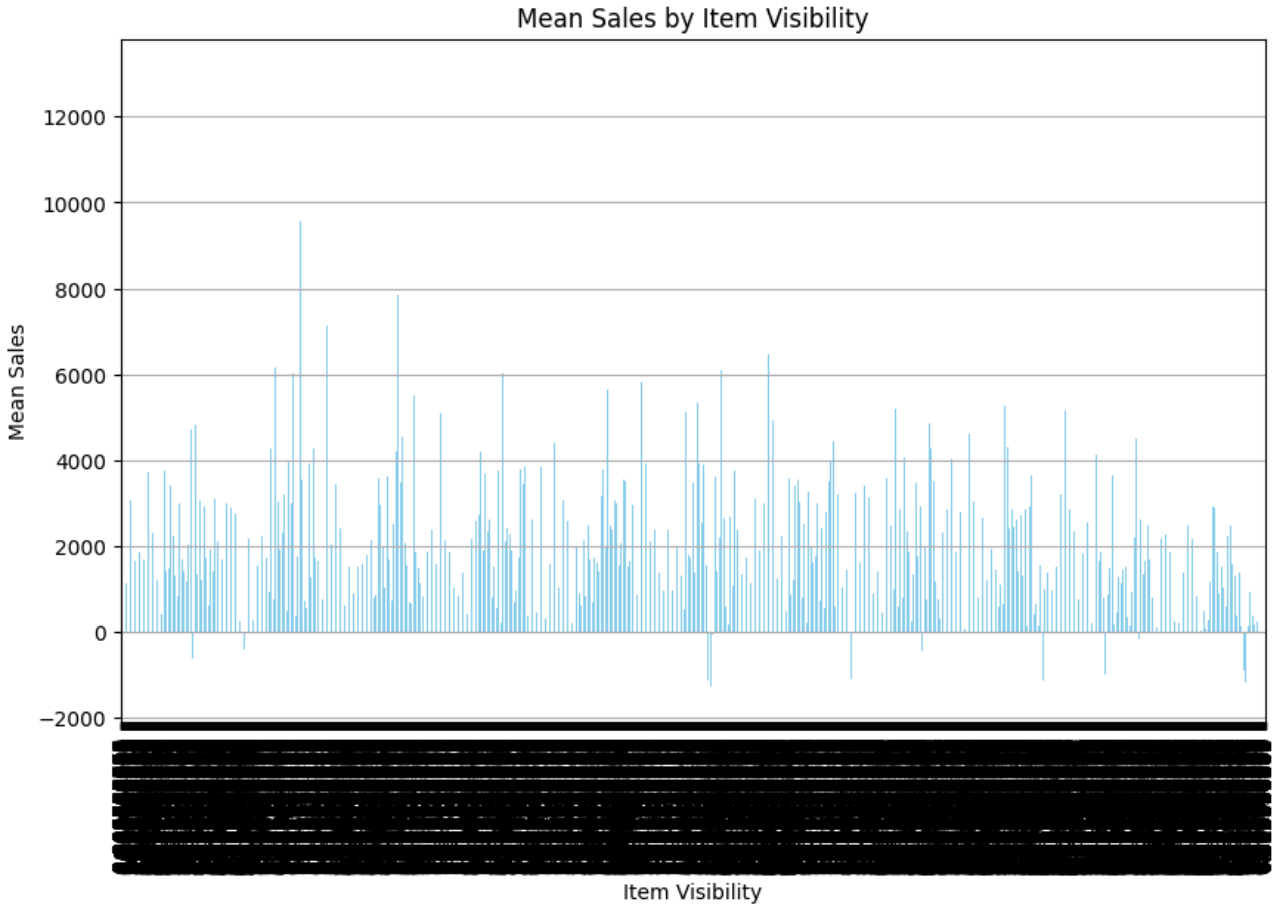



```
group_data1 = data.groupby('MRP_Level').size() #which type of item sell most
print(group_data1)
plt.figure()
plt.gcf().set_facecolor('lightgrey')#background color sentax
group_data1.plot(marker ='o', color = 'green' )
plt.title('MRP level data analysis')
plt.xlabel('MRP_level')
plt.ylabel('count')
plt.grid()
```

```
MRP_Level
High         4935
Low          2208
Medium       4661
Very_High    2400
dtype: int64
```




```
#How does the visibility of items in stores affect their sales? Are items with higher vi
visibility_groups = data.groupby('Item_Visibility')

# Step 2: Calculate mean sales for each visibility group
mean_sales_by_visibility = visibility_groups['Item_Outlet_Sales'].mean()

# Step 3: Visualize the relationship
plt.figure(figsize=(10, 6))
mean_sales_by_visibility.plot(kind='bar', color='skyblue')
plt.title('Mean Sales by Item Visibility')
plt.xlabel('Item Visibility')
plt.ylabel('Mean Sales')
plt.grid(axis='y')
plt.show()
```



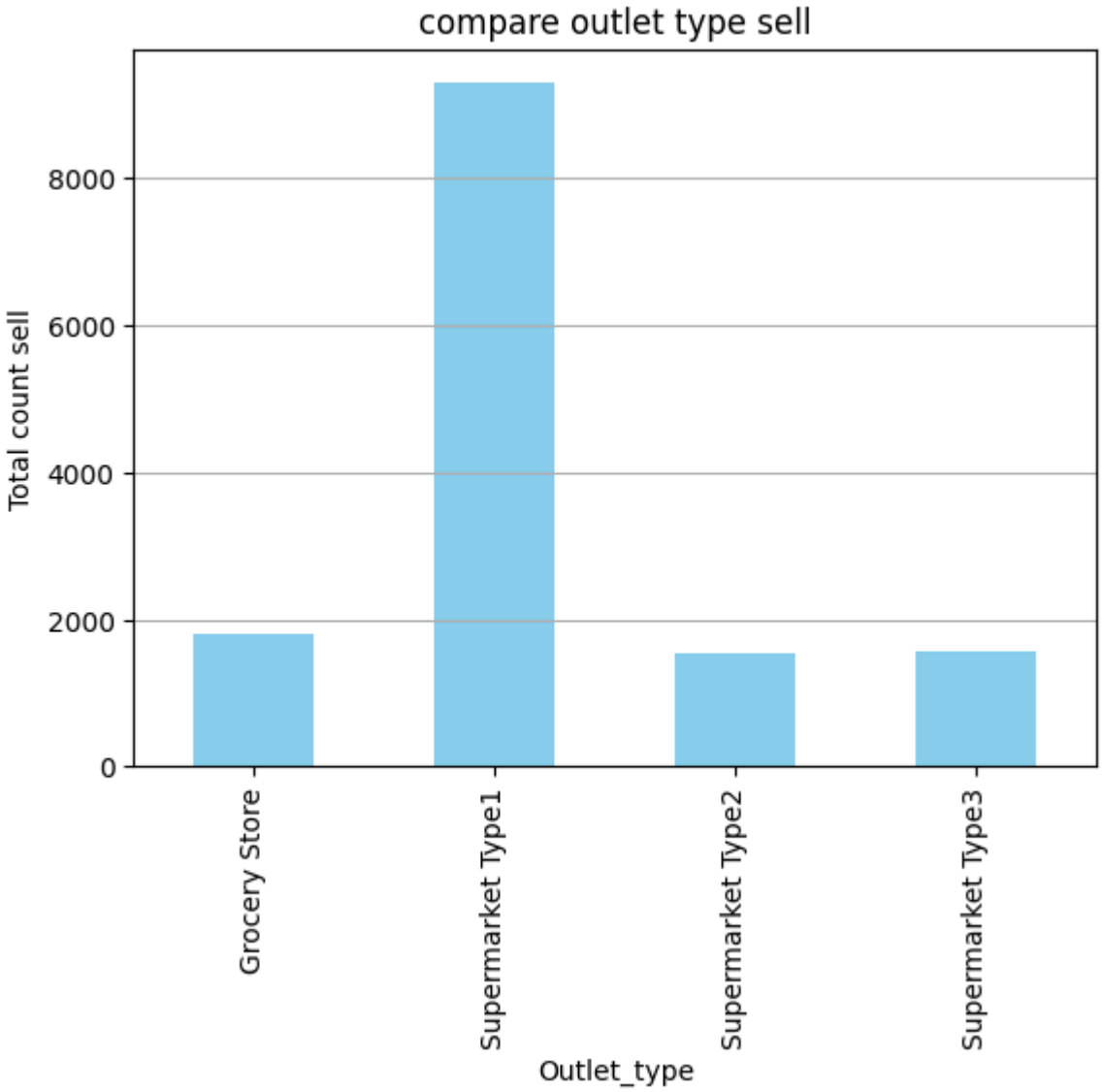

```
visibility_groups1 = data.groupby('Item_Visibility').mean()
print(visibility_groups1)
```

```
                  Item_Weight  Year  Item_MRP  Outlet_Size  Item_Outlet_Sales  \
Item_Visibility
0.003575                5.880  31.0  154.6998          2.0         3229.795800
0.003589                5.880  29.0  155.5998          1.0         1691.797800
0.003591                5.880  12.0  153.3998          3.0         2597.995487
0.003592                5.880  19.0  154.1998          3.0         2466.073202
0.003598                5.880  17.0  153.8998          2.0         2922.196200
...                       ...   ...       ...          ...                 ...
0.313935                8.355  18.0  146.5418          2.0          382.273706
0.321115                8.985  31.0  100.7700          3.0          199.740000
0.323637                7.670  31.0   35.5216          3.0        -1354.072946
0.325781                7.825  31.0  252.7698          3.0          761.009400
0.328391                8.355  31.0  146.0418          3.0          588.567200

                  Converted Sales
Item_Visibility
0.003575                    2.099
0.003589                    1.100
```

```
0.003591                  1.689
0.003592                  1.603
0.003598                  1.899
...                         ...
0.313935                  0.248
0.321115                  0.130
0.323637                 -0.880
0.325781                  0.495
0.328391                  0.383

[13688 rows x 6 columns]
<ipython-input-42-4c0566e070be>:1: FutureWarning: The default value of numeric_only
  visibility_groups1 = data.groupby('Item_Visibility').mean()
```

```
outlet_type_group= data.groupby('Outlet_Type').size()
print(outlet_type_group)
outlet_type_group.plot(kind='bar', color='skyblue')
plt.title('compare outlet type sell')
plt.xlabel('Outlet_type')
plt.ylabel('Total count sell')
plt.grid(axis='y')
plt.show()
```

```
Outlet_Type
Grocery Store        1805
Supermarket Type1    9294
Supermarket Type2    1546
Supermarket Type3    1559
dtype: int64
```




```
sort1 = data[data['Outlet_Type']=='Supermarket Type1']
print(sort1)
```

```
      Item_Identifier  Item_Weight  Item_Visibility  Year Outlet_Identifier  \
0               FDA15         9.30         0.016047    17            OUT049
2               FDN15        17.50         0.016760    17            OUT049
4               NCD19         8.93         0.014670    29            OUT013
6               FDO10        13.65         0.012741    29            OUT013
8               FDH17        16.20         0.016687    14            OUT045
...               ...          ...              ...   ...               ...
14198           FDW46        13.00         0.070411    17            OUT049
14199           FDB58        10.50         0.013496    19            OUT046
14201           NCO17        10.00         0.073529    14            OUT045
14202           FDJ26        15.30         0.098200     9            OUT017
14203           FDU37         9.50         0.104720    14            OUT045

               Item_Type  Item_MRP  MRP_Level  Outlet_Size  \
0                  Dairy  249.8092  Very_High            2
```

```
2                   Meat  141.6180       High            2
4              Household   53.8614        Low            1
6            Snack Foods   57.6588        Low            1
8           Frozen Foods   96.9726     Medium            2
...                  ...       ...        ...          ...
14198        Snack Foods   63.4484        Low            2
14199        Snack Foods  141.3154       High            3
14201  Health and Hygiene  118.7440     Medium            2
14202              Canned  214.6218  Very_High            2
14203              Canned   79.7960     Medium            2

      Outlet_Location_Type        Outlet_Type  Item_Outlet_Sales  \
0                   Tier 1  Supermarket Type1        3735.138000
2                   Tier 1  Supermarket Type1        2097.270000
4                   Tier 3  Supermarket Type1         994.705200
6                   Tier 3  Supermarket Type1         343.552800
8                   Tier 2  Supermarket Type1        1076.598600
...                    ...                ...                ...
14198               Tier 1  Supermarket Type1        1184.601202
14199               Tier 1  Supermarket Type1        2308.372585
14201               Tier 2  Supermarket Type1        1809.869722
14202               Tier 2  Supermarket Type1        3538.210920
14203               Tier 2  Supermarket Type1        1267.453484

       Converted Sales
0                2.428
2                1.363
4                0.647
6                0.223
8                0.700
...                ...
14198            0.770
14199            1.500
14201            1.176
14202            2.300
14203            0.824

[9294 rows x 13 columns]
```

```
MRP_type = sort1.groupby('MRP_Level').size() #which type of item sell most
print(MRP_type)
plt.figure()
plt.gcf().set_facecolor('lightgrey')#background color sentax
MRP_type.plot(marker ='o', color = 'green' )
plt.title('MRP level data analysis')
plt.xlabel('MRP_level')
plt.ylabel('count')
plt.grid()
```

```
MRP_Level
High         3219
```