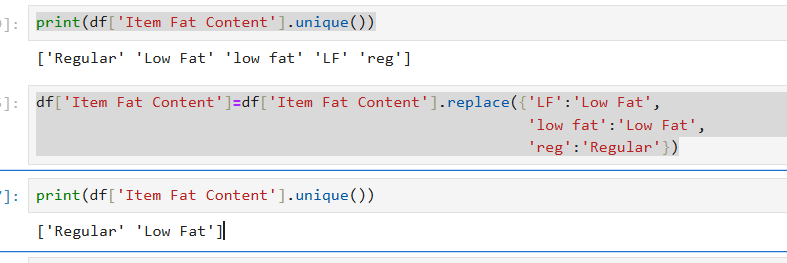
**DATA CLEANING :**

df['Item Fat Content']=df['Item Fat Content'].replace({'LF':'Low Fat',

'low fat':'Low Fat',

'reg':'Regular'})

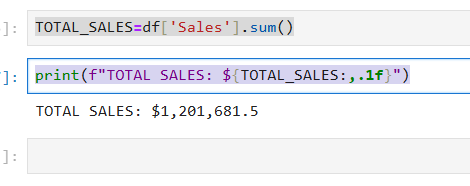


**REQUIREMENTS**

**1)TOTAL SALES:**

TOTAL\_SALES=df['Sales'].sum()

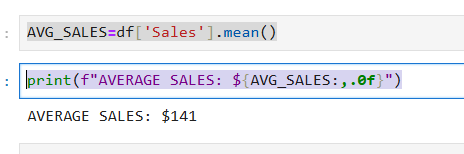
print(f"TOTAL SALES: ${TOTAL\_SALES:,.1f}")



**2)AVERAGE SALES:**

AVG\_SALES=df['Sales'].mean()

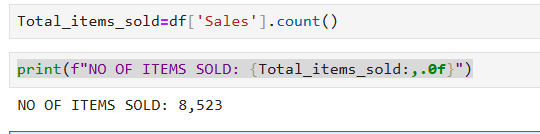
print(f"AVERAGE SALES: ${AVG\_SALES:,.0f}")



**3)NO OF ITEMS SOLD:**

Total\_items\_sold=df['Sales'].count()

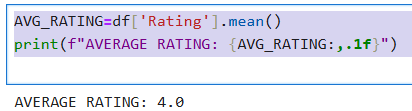
print(f"NO OF ITEMS SOLD: {Total\_items\_sold:,.0f}")



**4)AVERAGE RATING:**

AVG\_RATING=df['Rating'].mean()

print(f"AVERAGE RATING: {AVG\_RATING:,.1f}")



**5)TOTAL SALES BASED ON FAT CONTENT:**

def sum\_value(val):

total=sum(total\_sales\_based\_on\_fatcontent)

sumvalue=int(round(val\*total/100))

return f"${sumvalue:,}"

plt.pie(total\_sales\_based\_on\_fatcontent,labels=total\_sales\_based\_on\_fatcontent.index,

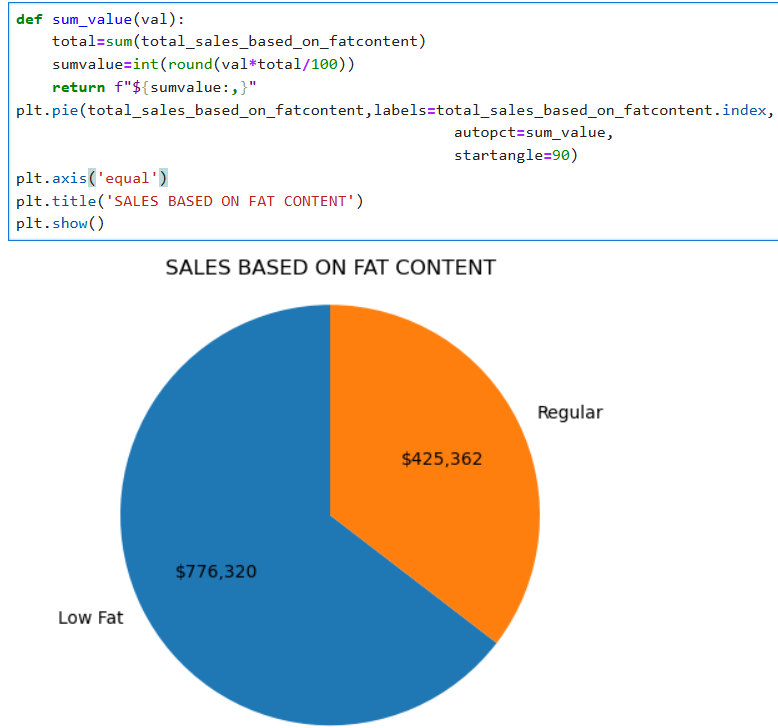
autopct=sum\_value,

startangle=90)

plt.axis('equal')

plt.title('SALES BASED ON FAT CONTENT')

plt.show()



**6)TOTAL SALES BASED ON ITEM TYPE:**

TOTAL\_SALES\_BASED\_ON\_ITEMTYPE=df.groupby('Item Type')['Sales'].sum().sort\_values(ascending=False)

plt.figure(figsize=(10,6))

bars=plt.bar(TOTAL\_SALES\_BASED\_ON\_ITEMTYPE.index,TOTAL\_SALES\_BASED\_ON\_ITEMTYPE.values)

plt.xticks(rotation=-90)

plt.xlabel('Item Type')

plt.ylabel('Sales')

plt.title('TOTAL SALES BASED ON ITEM TYPE')

for bar in bars:

plt.text(

bar.get\_x() + bar.get\_width() / 2,

bar.get\_height(),

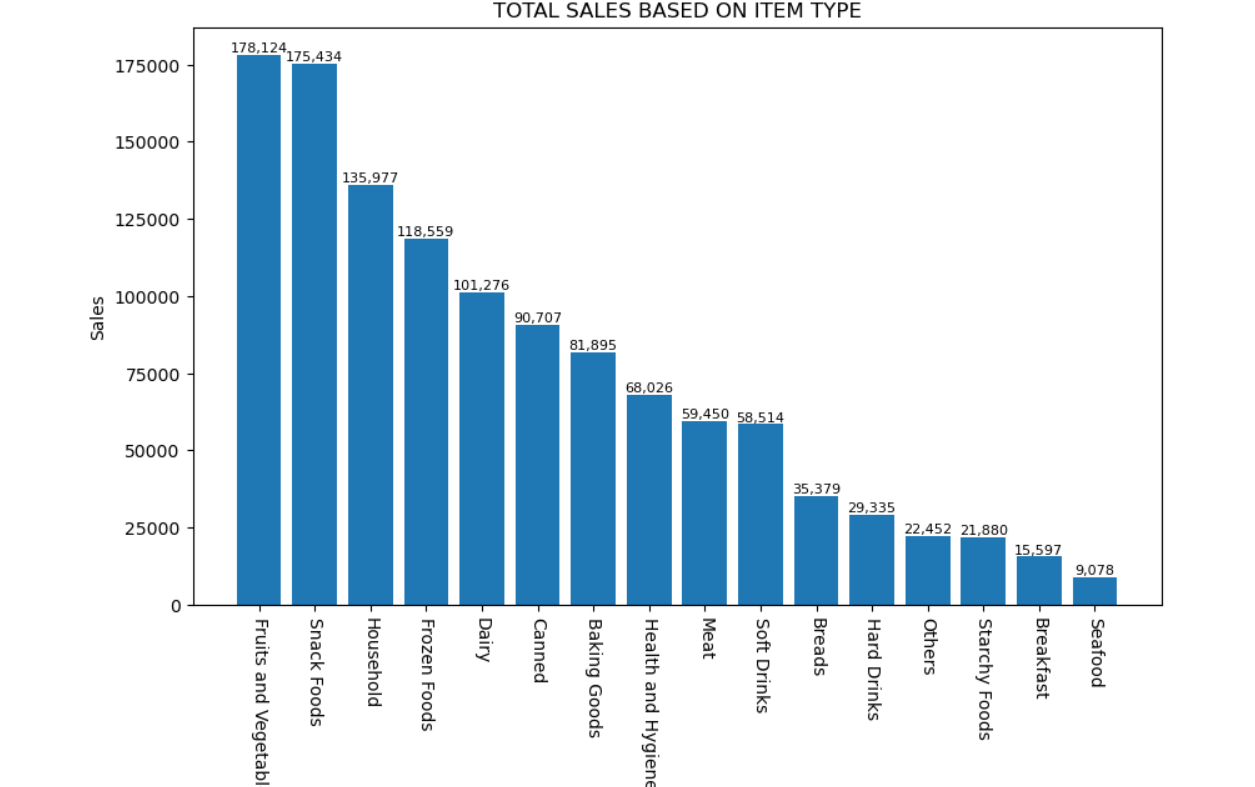
f'{bar.get\_height():,.0f}',

ha='center',

va='bottom',

fontsize=8

)



**7)TOTAL SALES BASED ON ITEM TYPE:**

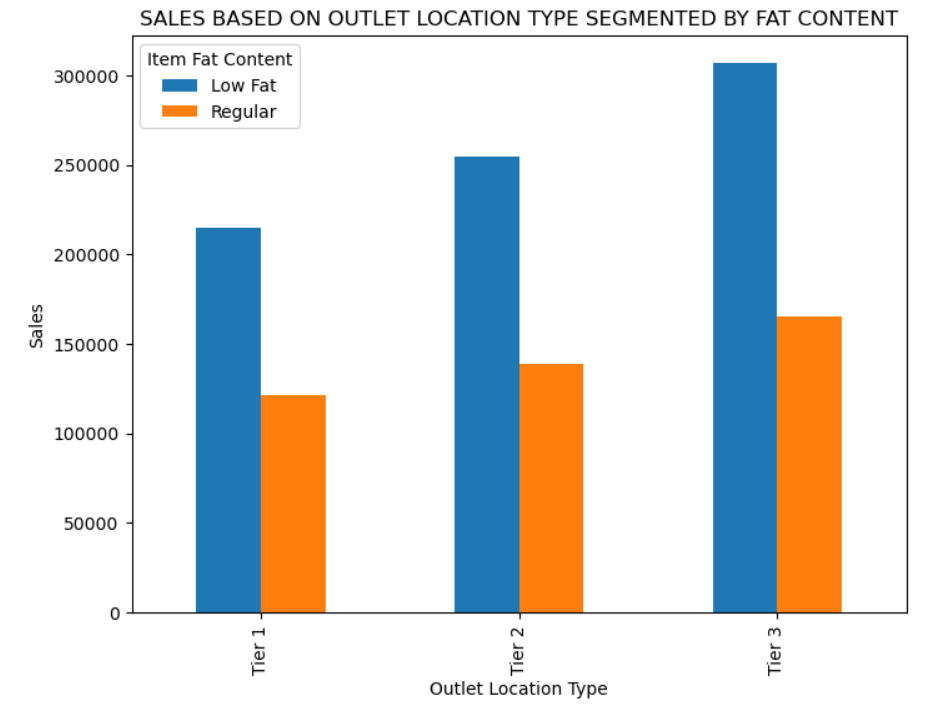
grouped=df.groupby(['Outlet Location Type','Item Fat Content'])['Sales'].sum().unstack()

grouped=grouped[['Low Fat','Regular']]

plot=grouped.plot(kind='bar',figsize=(8,6),title='SALES BASED ON OUTLET LOCATION TYPE SEGMENTED BY FAT CONTENT')

plt.ylabel('Sales')

plt.show()



**8)TOTAL SALES BASED ON OUTLET ESTABLISHMENT YEAR:**

SALES\_BASED\_ON\_ESTABLISHMENT\_YEAR=df.groupby('Outlet Establishment Year')['Sales'].sum().sort\_index()

plt.figure(figsize=(8,5))

line\_plot=plt.plot(SALES\_BASED\_ON\_ESTABLISHMENT\_YEAR.index,SALES\_BASED\_ON\_ESTABLISHMENT\_YEAR.values,marker='o',linestyle='-')

plt.xlabel('Outlet Establishment Year')

plt.ylabel('Sales')

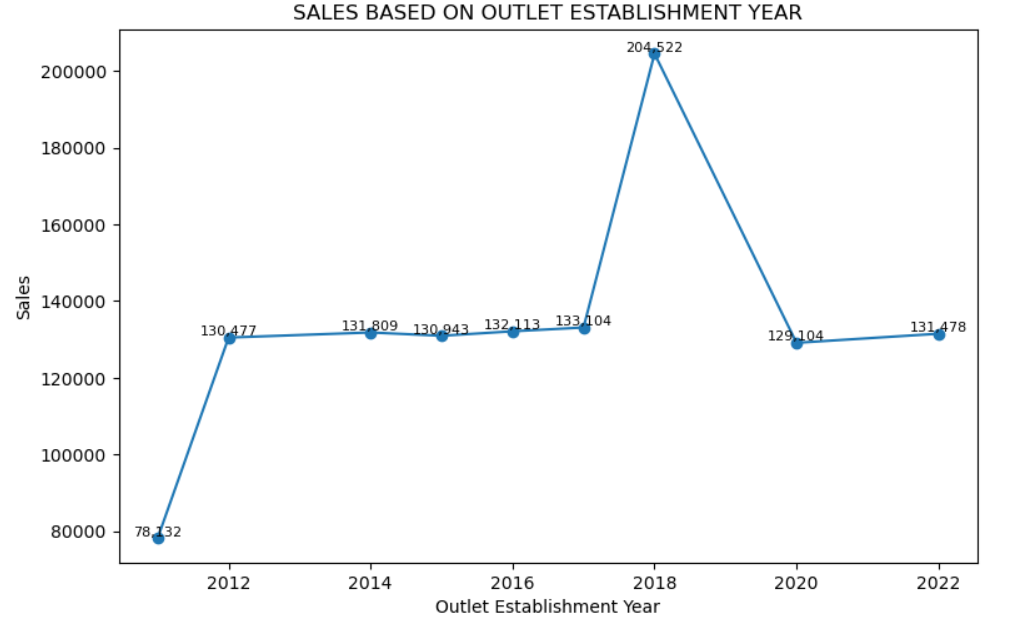
for x,y in zip(SALES\_BASED\_ON\_ESTABLISHMENT\_YEAR.index,SALES\_BASED\_ON\_ESTABLISHMENT\_YEAR.values):

plt.text(x,y, f'{y:,.0f}',fontsize=8,ha='center',va='bottom')

plt.tight\_layout()

plt.title('SALES BASED ON OUTLET ESTABLISHMENT YEAR')

plt.show()



**9)TOTAL SALES BASED ON ITEM TYPE:**

total\_sales\_based\_on\_outletsize=df.groupby('Outlet Size')['Sales'].sum()

def total\_value(val):

total\_sum=sum(total\_sales\_based\_on\_outletsize)

sum\_value=int(round(val\*total\_sum/100))

return f"${sum\_value:,}"

plt.pie(total\_sales\_based\_on\_outletsize,labels=total\_sales\_based\_on\_outletsize.index,

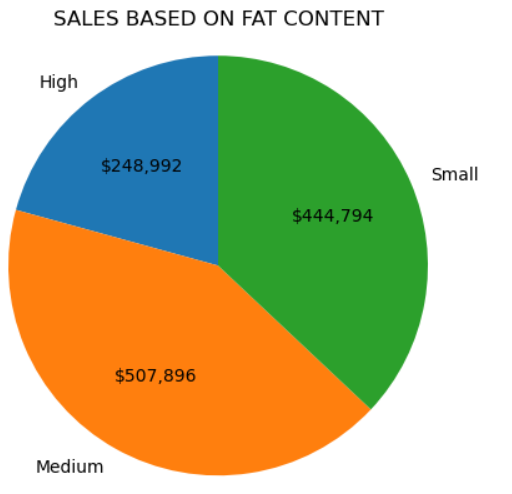
autopct=total\_value,

startangle=90)

plt.axis('equal')

plt.title('SALES BASED ON OUTLET SIZE')

plt.show()

****

**10)TOTAL SALES BASED ON OUTLET LOCATION:**

**USING MYPLOTLIB:**

TOTAL\_SALES\_BASED\_ON\_OUTLETTYPE=df.groupby('Outlet Type')['Sales'].sum().sort\_values(ascending=False)

plt.figure(figsize=(8,6))

bars=plt.bar(TOTAL\_SALES\_BASED\_ON\_OUTLETTYPE.index,TOTAL\_SALES\_BASED\_ON\_OUTLETTYPE.values)

plt.xlabel('Outlet Type')

plt.ylabel('Sales')

plt.title('SALES BASED ON OUTLET TYPE')

for bar in bars:

plt.text(bar.get\_x()+bar.get\_width()/2,

bar.get\_height(),

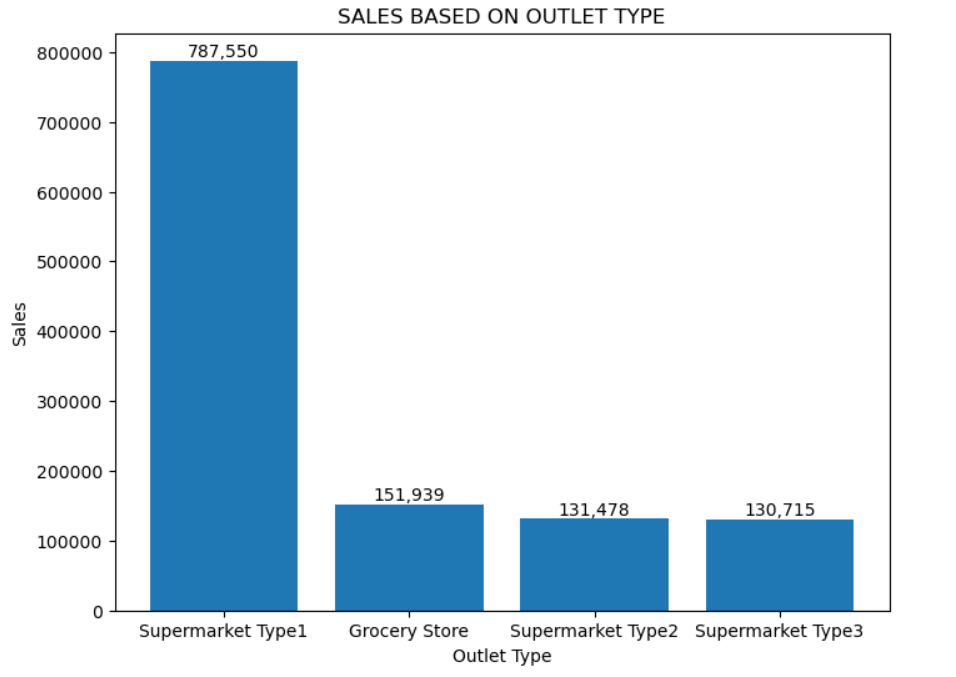
f'{bar.get\_height():,.0f}',

va='bottom',

ha='center',

fontsize=10

)



**USING SEABORN:**

TOTAL\_SALES\_BASED\_ON\_OUTLETTYPE=df.groupby('Outlet Type')['Sales'].sum().sort\_values(ascending=False)

df\_outlet = TOTAL\_SALES\_BASED\_ON\_OUTLETTYPE.reset\_index()

df\_outlet.columns = ['Outlet Type', 'Sales']

plt.figure(figsize=(8,6))

bars=sns.barplot(data=df\_outlet, x='Outlet Type', y='Sales')

plt.xlabel('Outlet Type')

plt.ylabel('Sales')

plt.title('SALES BASED ON OUTLET TYPE')

for bar in bars.patches:

plt.text(bar.get\_x()+bar.get\_width()/2,

bar.get\_height(),

f'{bar.get\_height():,.0f}',

va='bottom',

ha='center',

fontsize=10

)

