```
import pandas as pd
data  = pd.read_csv('/content/supermarket_sales - Sheet1.csv')
```

```
data.head()
```

	Invoice ID	Branch	City	Customer type	Gender	Product line	Unit price	Quantity	Tax 5%	
0	750-67-8428	A	Yangon	Member	Female	Health and beauty	74.69	7	26.1415	548
1	226-31-3081	C	Naypyitaw	Normal	Female	Electronic accessories	15.28	5	3.8200	80
2	631-41-3108	A	Yangon	Normal	Male	Home and lifestyle	46.33	7	16.2155	340
3	123-19-1176	A	Yangon	Member	Male	Health and beauty	58.22	8	23.2880	489
4	373-73-7910	A	Yangon	Normal	Male	Sports and travel	86.31	7	30.2085	634

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

```
data.isnull().sum()
```

	0
Invoice ID	0
Branch	0
City	0
Customer type	0
Gender	0
Product line	0
Unit price	0
Quantity	0
Tax 5%	0
Total	0
Date	0
Time	0
Payment	0
cogs	0
gross margin percentage	0
gross income	0
Rating	0

dtype: int64

```
data_excel = pd.read_excel('/content/Superstore.xlsx')
```

```
data_excel.head()
```

	Row ID	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	Country	City
0	1	CA-2013-152156	2013-11-09	2013-11-12	Second Class	CG-12520	Claire Gute	Consumer	United States	Henderson
1	2	CA-2013-152156	2013-11-09	2013-11-12	Second Class	CG-12520	Claire Gute	Consumer	United States	Henderson
2	3	CA-2013-138688	2013-06-13	2013-06-17	Second Class	DV-13045	Darrin Van Huff	Corporate	United States	Los Angeles
3	4	US-2012-108966	2012-10-11	2012-10-18	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States	Fort Lauderdale
4	5	US-2012-108966	2012-10-11	2012-10-18	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States	Fort Lauderdale

5 rows × 21 columns

```
data_excel.isnull().sum()
```

```
                  0
Row ID            0
Order ID          0
Order Date        0
Ship Date         0
Ship Mode         0
Customer ID       0
Customer Name     0
Segment           0
Country           0
City              0
State             0
Postal Code       0
Region            0
Product ID        0
Category          0
Sub-Category      0
Product Name      0
Sales             0
Quantity          0
Discount          0
Profit            0

dtype: int64
```

```
data_json = pd.read_json("/content/StoreSales.json")
```

```
data_json.head()
```

	Row ID	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	City	S
0	32298	CA-2012-124891	31-07-2012	31-07-2012	Same Day	RH-19495	Rick Hansen	Consumer	New York City	New
1	26341	IN-2013-77878	05-02-2013	07-02-2013	Second Class	JR-16210	Justin Ritter	Corporate	Wollongong	New S W
2	25330	IN-2013-71249	17-10-2013	18-10-2013	First Class	CR-12730	Craig Reiter	Consumer	Brisbane	Queens
3	13524	ES-2013-1579342	28-01-2013	30-01-2013	First Class	KM-16375	Katherine Murray	Home Office	Berlin	E
4	47221	SG-2013-4320	05-11-2013	06-11-2013	Same Day	RH-9495	Rick Hansen	Consumer	Dakar	D

5 rows × 24 columns

```
new_data = data_json.iloc[1:3]
```

```
new_data
```

	Row ID	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	City	Sta
1	26341	IN-2013-77878	05-02-2013	07-02-2013	Second Class	JR-16210	Justin Ritter	Corporate	Wollongong	New Sou Wal
2	25330	IN-2013-71249	17-10-2013	18-10-2013	First Class	CR-12730	Craig Reiter	Consumer	Brisbane	Queensla

2 rows × 24 columns

```
merged_data = merged_data = data_json.merge(data_json,how="outer",on="Row ID")
```

```
merged_data
```

	Row ID	Order ID_x	Order Date_x	Ship Date_x	Ship Mode_x	Customer ID_x	Customer Name_x	Segment_x	City_x	S
0		NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
1	1	MX-2014-143658	02-10-2014	06-10-2014	Standard Class	SC-20575	Sonia Cooley	Consumer	Mexico City	I
2	10	MX-2013-134096	27-09-2013	01-10-2013	Standard Class	DP-13000	Darren Powers	Consumer	S�o Paulo	
3	100	US-2013-125892	08-08-2013	10-08-2013	First Class	NW-18400	Natalie Webber	Consumer	Santo Domingo	Do
4	1000	MX-2013-126361	17-12-2013	19-12-2013	Second Class	AH-10690	Anna H�berlin	Corporate	Granada	G
...	...	...	...	...	...	...	...	...	...	
51286	9995	US-2014-110667	02-10-2014	06-10-2014	Standard Class	SC-20575	Sonia Cooley	Consumer	Pirapora	
51287	9996	US-2012-142734	15-10-2012	20-10-2012	Standard Class	KW-16570	Kelly Williams	Consumer	Indaial	C
51288	9997	US-2012-142734	15-10-2012	20-10-2012	Standard Class	KW-16570	Kelly Williams	Consumer	Indaial	C
51289	9998	US-2012-142734	15-10-2012	20-10-2012	Standard Class	KW-16570	Kelly Williams	Consumer	Indaial	C
51290	9999	US-2012-142734	15-10-2012	20-10-2012	Standard Class	KW-16570	Kelly Williams	Consumer	Indaial	C

51291 rows × 47 columns

```
data_json=data_json.iloc[:,:-3]
```

```
data_json.shape
```

```
(51291, 21)
```

```
data_excel.shape
```

```
(9994, 21)
```

```
data_json.describe()
```

	Sales	Quantity	Discount
count	51290.000000	51290.000000	51290.000000
mean	246.490581	3.476545	0.142908
std	487.565361	2.278766	0.212280
min	0.444000	1.000000	0.000000
25%	30.758625	2.000000	0.000000
50%	85.053000	3.000000	0.000000
75%	251.053200	5.000000	0.200000
max	22638.480000	14.000000	0.850000

```
data.describe()
```

	Unit price	Quantity	Tax 5%	Total	cogs	gross margin percentage	i
count	1000.000000	1000.000000	1000.000000	1000.000000	1000.00000	1.000000e+03	1000.0
mean	55.672130	5.510000	15.379369	322.966749	307.58738	4.761905e+00	15.3
std	26.494628	2.923431	11.708825	245.885335	234.17651	6.131498e-14	11.7
min	10.080000	1.000000	0.508500	10.678500	10.17000	4.761905e+00	0.5
25%	32.875000	3.000000	5.924875	124.422375	118.49750	4.761905e+00	5.9
50%	55.230000	5.000000	12.088000	253.848000	241.76000	4.761905e+00	12.0
75%	77.935000	8.000000	22.445250	471.350250	448.90500	4.761905e+00	22.4
max	99.960000	10.000000	49.650000	1042.650000	993.00000	4.761905e+00	49.6

```
print("Total no of sales : ")
round(data_json['Sales'].sum())
```

```
Total no of sales :
12642502
```

```
data_json.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 51291 entries, 0 to 51290
Data columns (total 21 columns):
 #   Column         Non-Null Count  Dtype  
---  ------         --------------  -----  
 0   Row ID         51291 non-null  object 
 1   Order ID       51290 non-null  object 
 2   Order Date     51290 non-null  object 
 3   Ship Date      51290 non-null  object 
 4   Ship Mode      51290 non-null  object 
 5   Customer ID    51290 non-null  object 
 6   Customer Name  51290 non-null  object 
 7   Segment        51290 non-null  object 
 8   City           51290 non-null  object 
 9   State          51290 non-null  object 
 10  Country        51290 non-null  object 
 11  Postal Code    51290 non-null  object 
 12  Market         51290 non-null  object 
 13  Region         51290 non-null  object 
 14  Product ID     51290 non-null  object 
 15  Category       51290 non-null  object 
 16  Sub-Category   51290 non-null  object 
 17  Product Name   51290 non-null  object 
 18  Sales          51290 non-null  float64
 19  Quantity       51290 non-null  float64
 20  Discount       51290 non-null  float64
dtypes: float64(3), object(18)
memory usage: 8.2+ MB
```

```
print("Average Order Value : ")
(data['Unit price'] * data['Quantity']).mean()
```

```
Average Order Value :
307.58738
```

```
data_json.isnull().sum()
```

```
                  0
       Row ID     0
     Order ID     1
   Order Date     1
    Ship Date     1
    Ship Mode     1
  Customer ID     1
Customer Name     1
      Segment     1
         City     1
        State     1
      Country     1
  Postal Code     1
       Market     1
       Region     1
   Product ID     1
     Category     1
 Sub-Category     1
 Product Name     1
        Sales     1
     Quantity     1
     Discount     1

dtype: int64
```

```
data_json.dropna(inplace=True)
```

```
import matplotlib.pyplot as plt
import numpy as np
import seaborn as sns
new_data = data_json.groupby('Category')
x = new_data['Quantity'].count().index
y = new_data['Quantity'].count().values
```

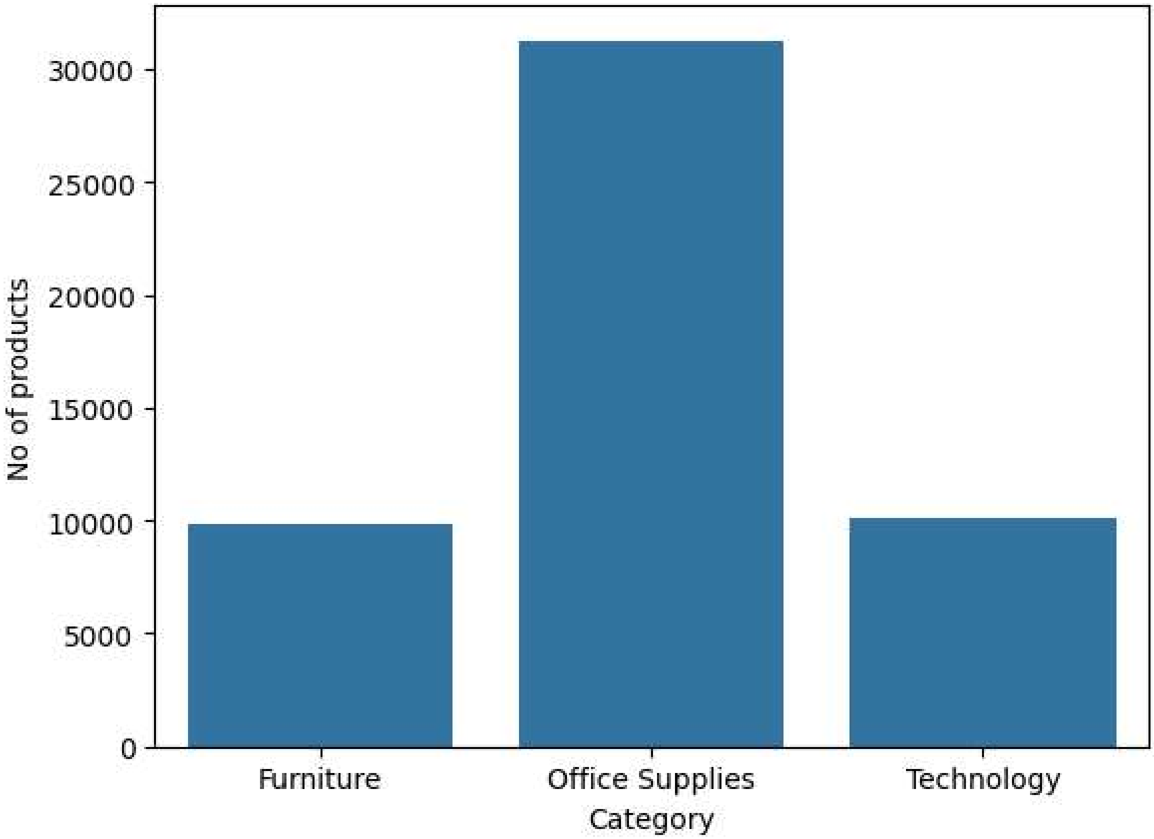
```
x
```

```
Index(['Furniture', 'Office Supplies', 'Technology'], dtype='object', name='Category')
```

```
y
```

```
array([ 9876, 31273, 10141])
```

```
sns.barplot(x=x, y=y)
plt.xlabel("Category")
plt.ylabel("No of products")
plt.show()
```



```
grouped_data = data_json.groupby('Category')
```

```
grouped_data.head()
```

ity	
ork City	Ne
ong	New
ane	Quee
erlin	
kar	
ney	New
rua	We
ton	W
nto	Ca
ord	C

```
dria

bul

ang   Heilo

aris

ato       T

◀
```

```
x = grouped_data['Sales'].sum()
```

```
data_json.columns
```

```
Index(['Row ID', 'Order ID', 'Order Date', 'Ship Date', 'Ship Mode',
       'Customer ID', 'Customer Name', 'Segment', 'City', 'State', 'Country',
       'Postal Code', 'Market', 'Region', 'Product ID', 'Category',
       'Sub-Category', 'Product Name', 'Sales', 'Quantity', 'Discount'],
      dtype='object')
```

```
x
```

```
                        Sales
      Category
     Furniture   4.110874e+06
Office Supplies  3.787070e+06
    Technology   4.744557e+06

dtype: float64
```

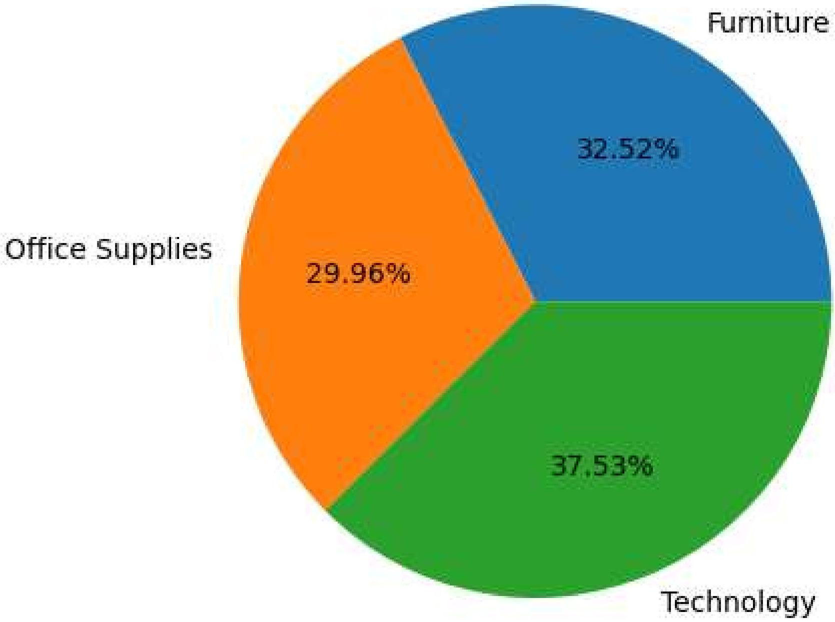
```
x.index
```

```
Index(['Furniture', 'Office Supplies', 'Technology'], dtype='object', name='Category')
```

```
import numpy as np
labels = np.array(x.index)
type(labels)
```

```
numpy.ndarray
```

```
import matplotlib.pyplot as plt
plt.pie(abs(x.values),labels=labels,autopct="%.2f%%")
plt.show()
```



```
state = data_json.groupby(['State','City'])
```

```
y = abs(state['Sales'].sum())
```

```
y = y.reset_index()
```

```
y = y.sort_values(by="Sales",ascending=False)
```

```
import seaborn as sns
sns.barplot(x = y['City'][:5],y = y['Sales'][:5])
plt.show()
```