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

```
dataset=pd.read_excel("/content/sample_data/QVI_transaction_data.xlsx")
```

```
dataset.head()
```

	DATE	STORE_NBR	LYLTY_CARD_NBR	TXN_ID	PROD_NBR	PROD_NAME	PROD_QTY	TOT_SALES
0	43390	1	1000	1	5	Natural Chip Compny SeaSalt175g	2	6.0
1	43599	1	1307	348	66	CCs Nacho Cheese 175g	3	6.3
2	43605	1	1343	383	61	Smiths Crinkle Cut Chips Chicken 170g	2	2.9
3	43329	2	2373	974	69	Smiths Chip Thinly S/Cream&Onion 175g	5	15.0
4	43330	2	2426	1038	108	Kettle Tortilla ChpsHny&Jlpno Chili 150g	3	13.8

description of data (summaries of the data)

```
dataset.describe()
```

	DATE	STORE_NBR	LYLTY_CARD_NBR	TXN_ID	PROD_NBR	PROD_QTY	TOT_SALES
count	264836.000000	264836.00000	2.648360e+05	2.648360e+05	264836.000000	264836.000000	264836.000000
mean	43464.036260	135.08011	1.355495e+05	1.351583e+05	56.583157	1.907309	7.304200
std	105.389282	76.78418	8.057998e+04	7.813303e+04	32.826638	0.643654	3.083226
min	43282.000000	1.00000	1.000000e+03	1.000000e+00	1.000000	1.000000	1.500000
25%	43373.000000	70.00000	7.002100e+04	6.760150e+04	28.000000	2.000000	5.400000
50%	43464.000000	130.00000	1.303575e+05	1.351375e+05	56.000000	2.000000	7.400000
75%	43555.000000	203.00000	2.030942e+05	2.027012e+05	85.000000	2.000000	9.200000
max	43646.000000	272.00000	2.373711e+06	2.415841e+06	114.000000	200.000000	650.000000

```
dataset.dtypes
```

	0
DATE	int64
STORE_NBR	int64
LYLTY_CARD_NBR	int64
TXN_ID	int64
PROD_NBR	int64
PROD_NAME	object
PROD_QTY	int64
TOT_SALES	float64

```
dtype: object
```

```
dataset.shape
```

```
(264836, 8)
```

```
dataset.ndim
```

```
2
```

```
#check the null vallue
```

```
dataset.isnull()
```

	DATE	STORE_NBR	LYLTY_CARD_NBR	TXN_ID	PROD_NBR	PROD_NAME	PROD_QTY	TOT_SALES
0	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False
...	...	...	...	...	...	...	...	...
264831	False	False	False	False	False	False	False	False
264832	False	False	False	False	False	False	False	False
264833	False	False	False	False	False	False	False	False
264834	False	False	False	False	False	False	False	False
264835	False	False	False	False	False	False	False	False

264836 rows × 8 columns

```
# total null values

dataset.isnull().sum()
```

	0
DATE	0
STORE_NBR	0
LYLTY_CARD_NBR	0
TXN_ID	0
PROD_NBR	0
PROD_NAME	0
PROD_QTY	0
TOT_SALES	0

dtype: int64

```
#check the duplicates

dataset.duplicated()
```

	0
0	False
1	False
2	False
3	False
4	False
...	...
264831	False
264832	False
264833	False
264834	False
264835	False

264836 rows × 1 columns

dtype: bool

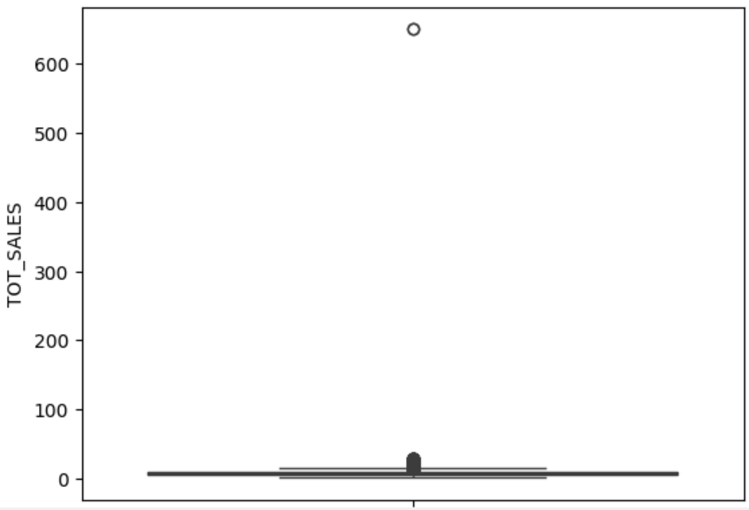
```
# total duplicates

dataset.duplicated().sum()
```

```
1
```

```
# Check the outlier
```

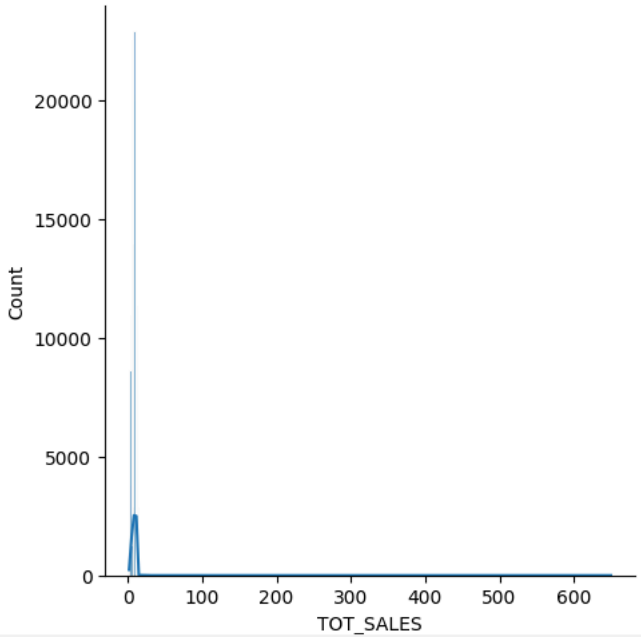
```
sns.boxplot(dataset.TOT_SALES)
```

```
<Axes: ylabel='TOT_SALES'>
```



```
sns.displot(dataset.TOT_SALES, kde=True)
```

```
<seaborn.axisgrid.FacetGrid at 0x7df648de6980>
```



```
#remove all the unwanted data from the main data
```

```
numericdata=dataset.select_dtypes(["float","int"])
```

```
numericdata.head() #got all float and numerica data
```

	DATE	STORE_NBR	LYLTY_CARD_NBR	TXN_ID	PROD_NBR	PROD_QTY	TOT_SALES
0	43390	1	1000	1	5	2	6.0
1	43599	1	1307	348	66	3	6.3
2	43605	1	1343	383	61	2	2.9
3	43329	2	2373	974	69	5	15.0
4	43330	2	2426	1038	108	3	13.8

```
x=numericdata[numericdata["TOT_SALES"]<8.000]
```

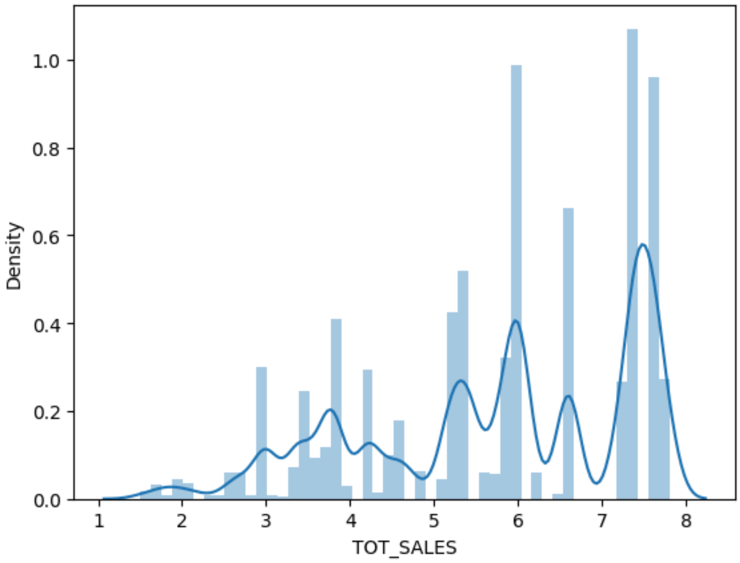
```
x.head()
```

	DATE	STORE_NBR	LYLTY_CARD_NBR	TXN_ID	PROD_NBR	PROD_QTY	TOT_SALES
0	43390	1	1000	1	5	2	6.0
1	43599	1	1307	348	66	3	6.3
2	43605	1	1343	383	61	2	2.9
5	43604	4	4074	2982	57	1	5.1
6	43601	4	4149	3333	16	1	5.7

```
# Now plot the x to check weather we have outlier or not
```

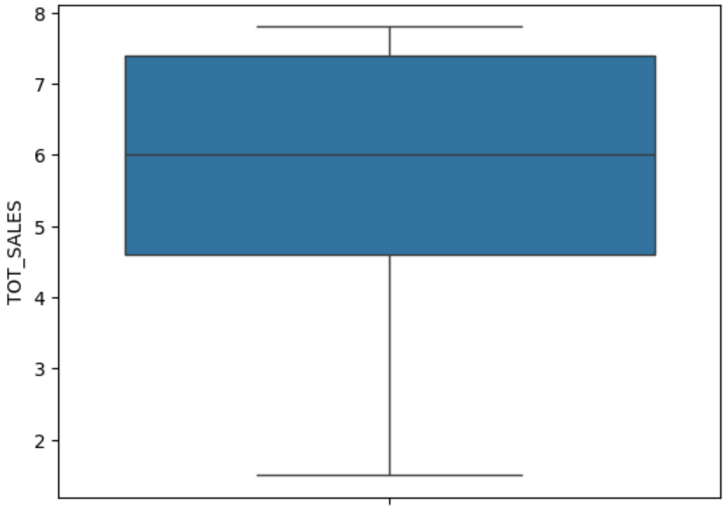
```
sns.distplot(x.TOT_SALES,kde=True)
```

```
<ipython-input-26-91e4357a2d5d>:3: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(x.TOT_SALES,kde=True)
<Axes: xlabel='TOT_SALES', ylabel='Density'>
```



```
sns.boxplot(x.TOT_SALES)
```

```
<Axes: ylabel='TOT_SALES'>
```



```
#now check the data formates
```

```
dataset.dtypes
```

```
                      0
DATE              int64
STORE_NBR         int64
LYLTY_CARD_NBR    int64
TXN_ID            int64
PROD_NBR          int64
PROD_NAME        object
PROD_QTY          int64
TOT_SALES       float64

dtype: object
```

```
dataset.head()
```

```
    DATE  STORE_NBR  LYLTY_CARD_NBR  TXN_ID  PROD_NBR                                 PROD_NAME  PROD_QTY  TOT_SALES
0  43390          1            1000       1         5      Natural Chip Compny SeaSalt175g         2        6.0
1  43599          1            1307     348        66                  CCs Nacho Cheese 175g         3        6.3
2  43605          1            1343     383        61    Smiths Crinkle Cut Chips Chicken 170g         2        2.9
3  43329          2            2373     974        69  Smiths Chip Thinly S/Cream&Onion 175g         5       15.0
4  43330          2            2426    1038       108  Kettle Tortilla ChpsHny&Jlpno Chili 150g         3       13.8
```