In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:
```
import warnings
warnings.filterwarnings("ignore")
```

In [3]:
```
pb=pd.read_csv("QVI_purchase_behaviour.csv")
```

In [4]:
```
pb.head()
```

Out[4]:

	LYLTY_CARD_NBR	LIFESTAGE	PREMIUM_CUSTOMER
0	1000	YOUNG SINGLES/COUPLES	Premium
1	1002	YOUNG SINGLES/COUPLES	Mainstream
2	1003	YOUNG FAMILIES	Budget
3	1004	OLDER SINGLES/COUPLES	Mainstream
4	1005	MIDAGE SINGLES/COUPLES	Mainstream

In [27]:
```
td=pd.read_excel("QVI_transaction_data.xlsx")
```

In [6]:
```
td.head()
```

Out[6]:

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

In [7]:
```
pb.columns
```

Out[7]:
```
Index(['LYLTY_CARD_NBR', 'LIFESTAGE', 'PREMIUM_CUSTOMER'], dtype='object')
```

In [8]:
```
td.columns
```

Out[8]:
```
Index(['DATE', 'STORE_NBR', 'LYLTY_CARD_NBR', 'TXN_ID', 'PROD_NBR',
       'PROD_NAME', 'PROD_QTY', 'TOT_SALES'],
      dtype='object')
```

CHECKING THE DATA TYPES

In [9]:
```
pb.dtypes
```

Out[9]:
```
LYLTY_CARD_NBR      int64
LIFESTAGE          object
PREMIUM_CUSTOMER   object
dtype: object
```

In [10]:
```
td.dtypes
```

Out[10]:
```
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
```

FINDING COUNT OF PRODUCTS

In [43]:
```
pd.DataFrame(td.PROD_NAME.value_counts())
```

Out[43]:

	PROD_NAME
Kettle Mozzarella Basil & Pesto 175g	3304
Kettle Tortilla ChpsHny&Jlpno Chili 150g	3296
Cobs Popd Swt/Chlli &Sr/Cream Chips 110g	3269
Tyrrells Crisps Ched & Chives 165g	3268
Cobs Popd Sea Salt Chips 110g	3265
...	...
RRD Pc Sea Salt 165g	1431
Woolworths Medium Salsa 300g	1430
NCC Sour Cream & Garden Chives 175g	1419
French Fries Potato Chips 175g	1418
WW Crinkle Cut Original 175g	1410

CONVERTING DATATYPE

In [32]:
```
td.DATE=pd.to_datetime(td.DATE,errors='ignore')
```

In [33]:
```
td.dtypes
```

Out[33]:
```
DATE              datetime64[ns]
STORE_NBR                  int64
LYLTY_CARD_NBR             int64
TXN_ID                     int64
PROD_NBR                   int64
PROD_NAME                 object
PROD_QTY                   int64
TOT_SALES                float64
dtype: object
```

SUMMARIZATION

In [11]:
```
pb.shape
```

Out[11]:
```
(72637, 3)
```

In [12]:
```
td.shape
```

Out[12]:
```
(264836, 8)
```

In [13]:
```
pb.describe()
```

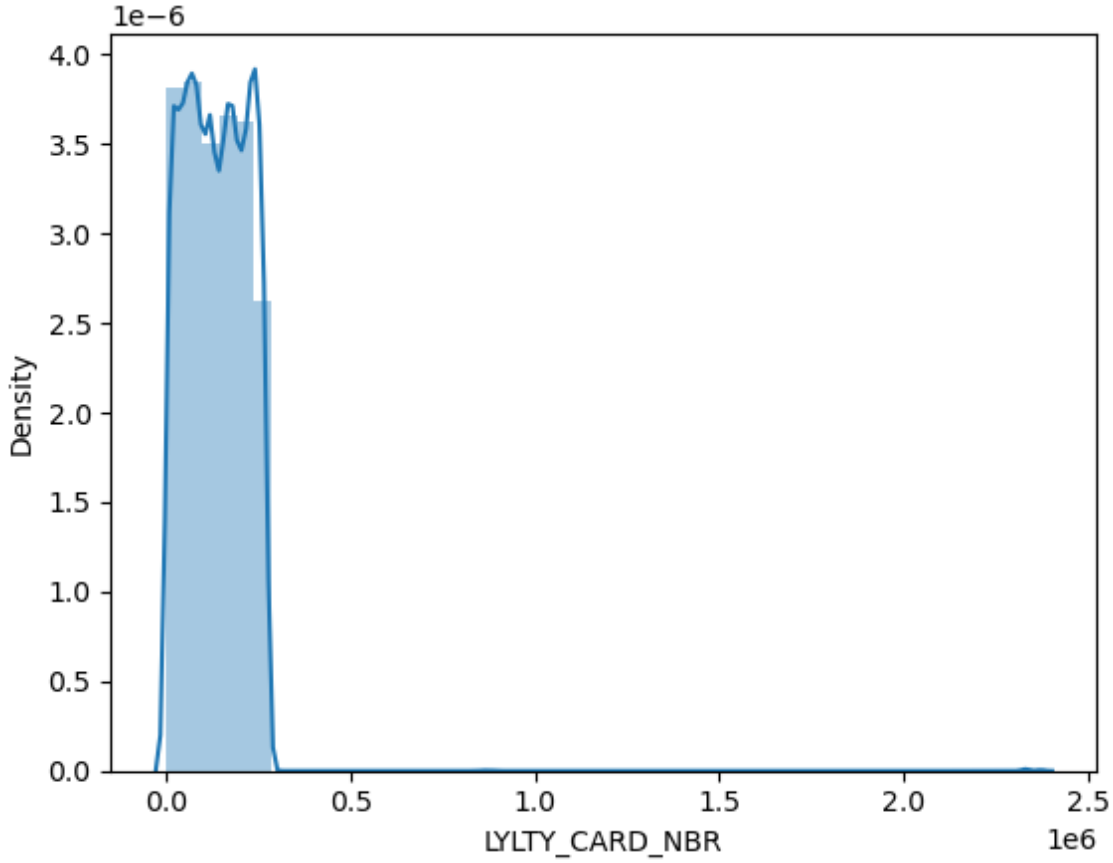
Out[13]:

	LYLTY_CARD_NBR
count	7.263700e+04
mean	1.361859e+05
std	8.989293e+04
min	1.000000e+03
25%	6.620200e+04
50%	1.340400e+05
75%	2.033750e+05
max	2.373711e+06

AS WE CAN SEE MEAN AND MEDIAN ARE ALMOST SIMILAR SO WE CAN CONCLUDE THAT THE DATA IS ALMOST NORMALLLY DISTRIBUTED

In [14]:
```
sns.distplot(pb['LYLTY_CARD_NBR'])
```

Out[14]:
```
<Axes: xlabel='LYLTY_CARD_NBR', ylabel='Density'>
```



In [15]:
```
td.describe()
```

Out[15]:

	DATE	STORE_NBR	LYLTY_CARD_NBR	TXN_ID	PROD_NBR	PR
count	264836.000000	264836.00000	2.648360e+05	2.648360e+05	264836.000000	26483
mean	43464.036260	135.08011	1.355495e+05	1.351583e+05	56.583157	
std	105.389282	76.78418	8.057998e+04	7.813303e+04	32.826638	
min	43282.000000	1.00000	1.000000e+03	1.000000e+00	1.000000	
25%	43373.000000	70.00000	7.002100e+04	6.760150e+04	28.000000	
50%	43464.000000	130.00000	1.303575e+05	1.351375e+05	56.000000	
75%	43555.000000	203.00000	2.030942e+05	2.027012e+05	85.000000	
max	43646.000000	272.00000	2.373711e+06	2.415841e+06	114.000000	20

AS WE CAN SEE THAT TOTAL SALES COLUMN HAS MAX VALUE OF 650 WHICH IS GREATER THAN DESIRED RANGE SO THERE ARE OUTLIERS PRESENT

CHECKING FOR NULL VALUES

In [16]:
```
td.isnull().sum()
```

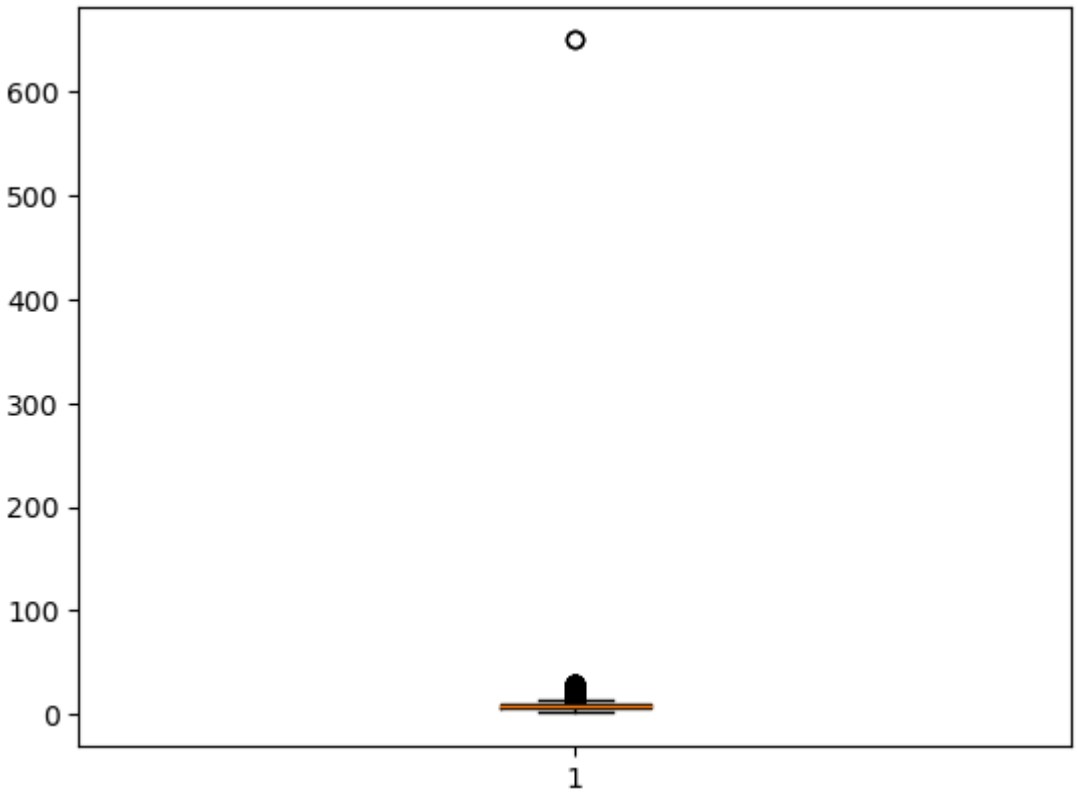
Out[16]:
```
DATE              0
STORE_NBR         0
LYLTY_CARD_NBR    0
TXN_ID            0
PROD_NBR          0
PROD_NAME         0
PROD_QTY          0
TOT_SALES         0
dtype: int64
```

In [17]:
```
pb.isnull().sum()
```

Out[17]:
```
LYLTY_CARD_NBR      0
LIFESTAGE           0
PREMIUM_CUSTOMER    0
dtype: int64
```

CLEANING THE OUTLIER

In [18]:
```
bp=plt.boxplot(td.TOT_SALES)
```

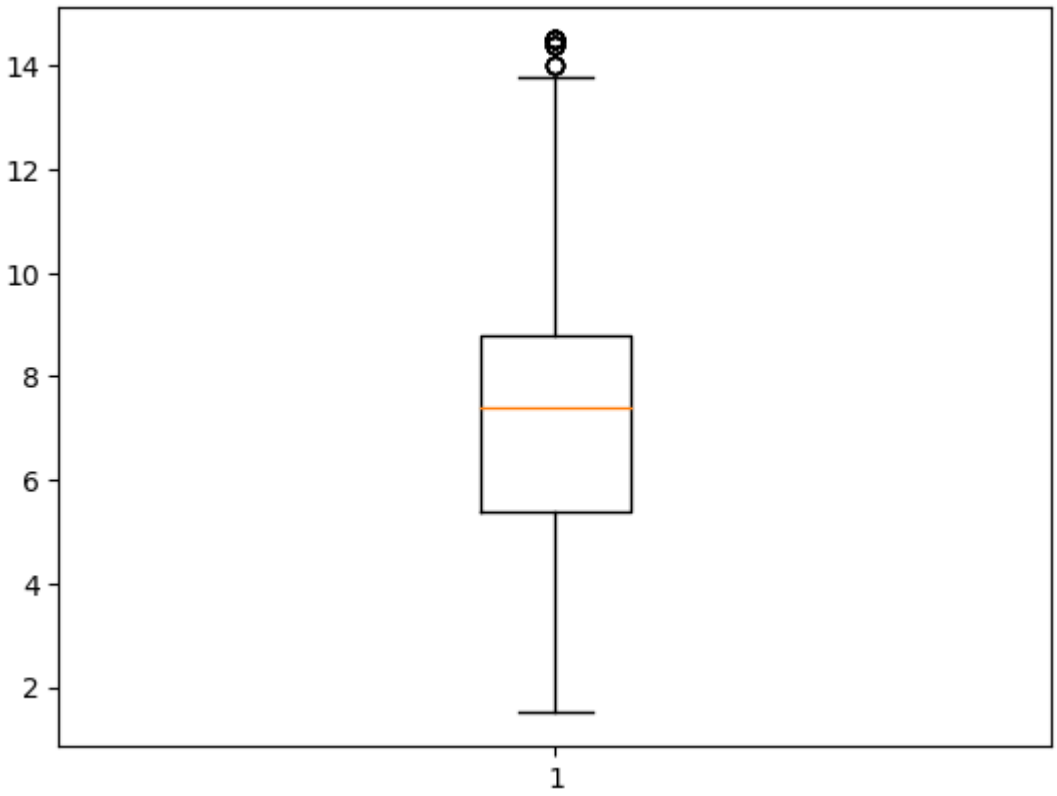


In [19]:
```
[x.get_ydata() for x in bp['whiskers']]
```

Out[19]:
```
[array([5.4, 1.5]), array([ 9.2, 14.8])]
```

In [20]:
```
td_clean=td[td.TOT_SALES<14.8]
```

In [21]:
```
pb1=plt.boxplot(td_clean.TOT_SALES)
```



In [22]:
```
[x.get_ydata() for x in pb1['fliers']]
```

Out[22]:
```
[array([14.5, 14.5, 14.5, 14.4, 14.5, 14.5, 14.5, 14.5, 14.5, 14.5, 14. ,
        14.5, 14. , 14.5, 14.4, 14.4, 14.4, 14.5, 14.4, 14.4, 14. , 14.5,
        14. , 14.5, 14.5, 14.5, 14.5, 14.5, 14. , 14.5, 14.4])]
```

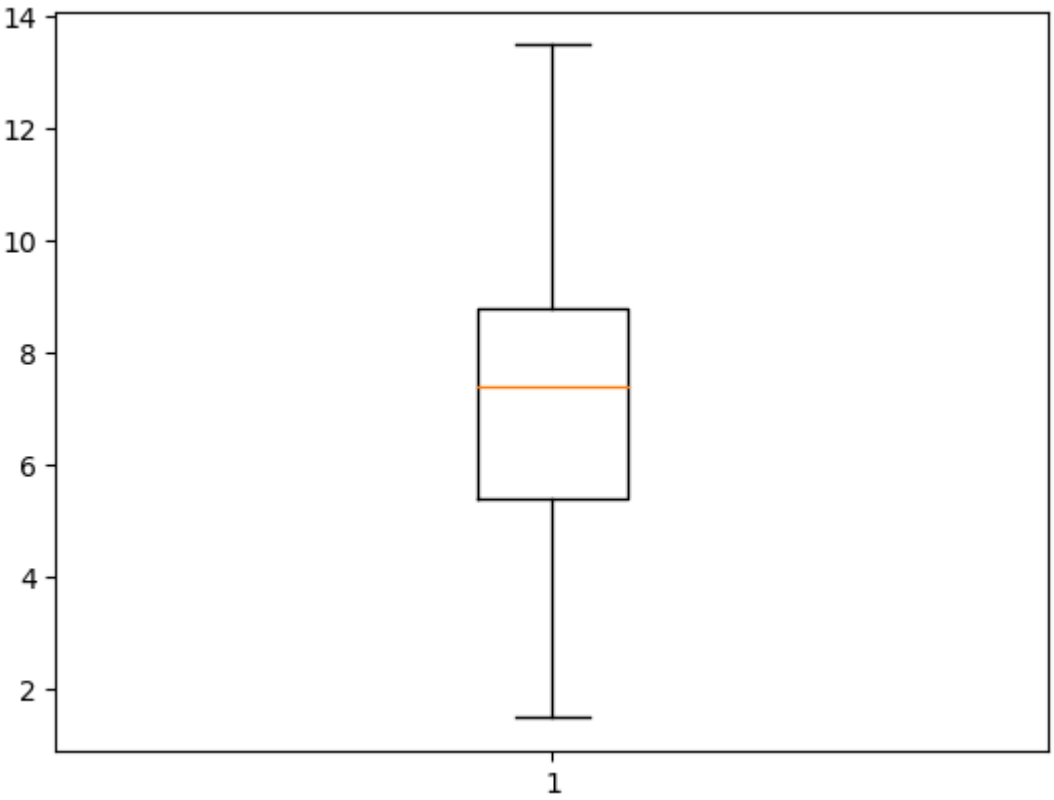
In [23]:
```
td_clean[td_clean.TOT_SALES>13.8]
```

55270	43604	43	43047	38903	100	Smiths Crinkle Cut Chips Chs&Onion170g
55489	43327	156	156195	157931	61	Smiths Crinkle Cut Chips Chicken 170g
55561	43331	191	191126	192500	86	Cheetos Puffs 165g
55635	43332	229	229227	231732	61	Smiths Crinkle Cut Chips Chicken 170g
69771	43605	227	227038	228513	86	Cheetos Puffs 165g
80742	43603	56	56164	50893	1	Smiths Crinkle Cut Chips Barbecue 170g
						Grain Waves

In [24]:
```
td_clean=td_clean[td_clean.TOT_SALES<13.8]
```

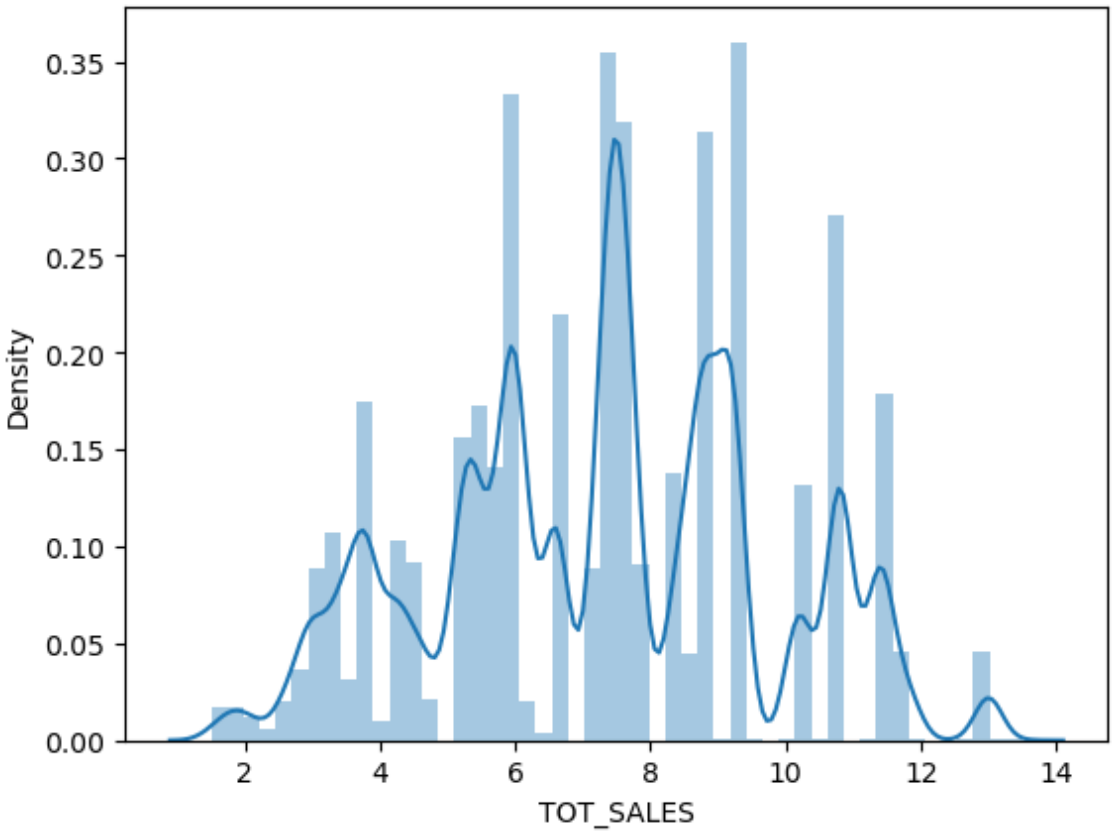
In [25]:
```
plt.boxplot(td_clean.TOT_SALES)
```

Out[25]:
```
{'whiskers': [<matplotlib.lines.Line2D at 0x295774ff050>,
  <matplotlib.lines.Line2D at 0x295774fd850>],
 'caps': [<matplotlib.lines.Line2D at 0x295774fdfd0>,
  <matplotlib.lines.Line2D at 0x295774fc350>],
 'boxes': [<matplotlib.lines.Line2D at 0x29576db3050>],
 'medians': [<matplotlib.lines.Line2D at 0x295774ffd50>],
 'fliers': [<matplotlib.lines.Line2D at 0x295774fe250>],
 'means': []}
```



In [26]:
```
sns.distplot(td_clean.TOT_SALES)
```

Out[26]:
```
<Axes: xlabel='TOT_SALES', ylabel='Density'>
```



THE ABOVE DISTPLOT SHOWS THAT AFTER THE REMOVAL OF OUTLIERS OUR DATA IS NORMALLY DISTRIBUTED