In [1]:
```
%matplotlib inline
```

In [2]:
```
import pandas as pd
import numpy as np
import seaborn as sb
```

In [3]:
```
# import purchase dataset
purchase = pd.read_csv('QVI_purchase_behaviour.csv')
purchase.head()
```

Out[3]:

	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 [4]:
```
# import transaction dataset
transaction = pd.read_excel('QVI_transaction_data.xlsx')
transaction.head()
```

Out[4]:

	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

In [5]:
```
# summary of transaction dataset
transaction.describe()
```

Out[5]:

	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

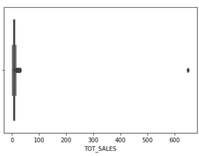
In [6]:
```
# check null values
transaction.isnull().sum()
```

Out[6]:
```
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
```

Remove Outliers

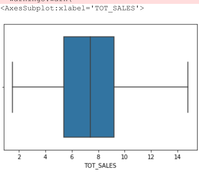
In [7]:
```
# find outliers for transaction dataset
sb.boxplot(transaction.TOT_SALES)
```

```
C:\Users\86189\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```
Out[7]:
```
<AxesSubplot:xlabel='TOT_SALES'>
```



In [8]:
```
# show boxplot without outliers
sb.boxplot(transaction.TOT_SALES, showfliers = False)
```

```
C:\Users\86189\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```
Out[8]:
```
<AxesSubplot:xlabel='TOT_SALES'>
```



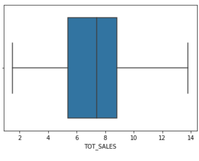
In [11]:
```
# remove outliers from dataset
transaction_clean = transaction[transaction.TOT_SALES < 14]
transaction_clean.head()
```

Out[11]:

	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
4	43330	2	2426	1038	108	Kettle Tortilla ChpsHny&Jlpno Chili 150g	3	13.8
5	43604	4	4074	2982	57	Old El Paso Salsa Dip Tomato Mild 300g	1	5.1

In [12]:
```
# check clean dataset's boxplot
sb.boxplot(transaction_clean.TOT_SALES)
```

```
C:\Users\86189\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```
Out[12]:
```
<AxesSubplot:xlabel='TOT_SALES'>
```



Check data format

In [14]:
```
transaction_clean.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 264187 entries, 0 to 264835
Data columns (total 8 columns):
 #   Column          Non-Null Count   Dtype  
---  ------          --------------   -----  
 0   DATE            264187 non-null  int64  
 1   STORE_NBR       264187 non-null  int64  
 2   LYLTY_CARD_NBR  264187 non-null  int64  
 3   TXN_ID          264187 non-null  int64  
 4   PROD_NBR        264187 non-null  int64  
 5   PROD_NAME       264187 non-null  object 
 6   PROD_QTY        264187 non-null  int64  
 7   TOT_SALES       264187 non-null  float64
dtypes: float64(1), int64(6), object(1)
memory usage: 18.1+ MB
```

In [15]:
```
# change DATE's format
transaction_clean.DATE = pd.to_datetime(transaction_clean.DATE, unit='d', origin='1899-12-30')
transaction_clean.head()
```

```
C:\Users\86189\anaconda3\lib\site-packages\pandas\core\generic.py:5507: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  self[name] = value
```
Out[15]:

	DATE	STORE_NBR	LYLTY_CARD_NBR	TXN_ID	PROD_NBR	PROD_NAME	PROD_QTY	TOT_SALES
0	2018-10-17	1	1000	1	5	Natural Chip Compny SeaSalt175g	2	6.0
1	2019-05-14	1	1307	348	66	CCs Nacho Cheese 175g	3	6.3
2	2019-05-20	1	1343	383	61	Smiths Crinkle Cut Chips Chicken 170g	2	2.9
4	2018-08-18	2	2426	1038	108	Kettle Tortilla ChpsHny&Jlpno Chili 150g	3	13.8
5	2019-05-19	4	4074	2982	57	Old El Paso Salsa Dip Tomato Mild 300g	1	5.1

In [16]:
```
transaction_clean.dtypes
```

Out[16]:
```
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
```

Combine two dataframes

In [19]:
```
df = transaction_clean.join(purchase.set_index('LYLTY_CARD_NBR'), on='LYLTY_CARD_NBR')
df.head()
```

Out[19]:

	DATE	STORE_NBR	LYLTY_CARD_NBR	TXN_ID	PROD_NBR	PROD_NAME	PROD_QTY	TOT_SALES	LIFESTAGE	PREMIUM_CUSTOMER
0	2018-10-17	1	1000	1	5	Natural Chip Compny SeaSalt175g	2	6.0	YOUNG SINGLES/COUPLES	Premium
1	2019-05-14	1	1307	348	66	CCs Nacho Cheese 175g	3	6.3	MIDAGE SINGLES/COUPLES	Budget
2	2019-05-20	1	1343	383	61	Smiths Crinkle Cut Chips Chicken 170g	2	2.9	MIDAGE SINGLES/COUPLES	Budget
4	2018-08-18	2	2426	1038	108	Kettle Tortilla ChpsHny&Jlpno Chili 150g	3	13.8	MIDAGE SINGLES/COUPLES	Budget
5	2019-05-19	4	4074	2982	57	Old El Paso Salsa Dip Tomato Mild 300g	1	5.1	MIDAGE SINGLES/COUPLES	Budget

Export to csv

In [20]:
```
df.to_csv('clean_data.csv')
```

In [ ]:
```

```