In [15]:

```
import pandas as pd
```

In [18]:

```
df = pd.read_csv('Retail_Data_Transactions.csv')
```

In [19]:

```
df
```

Out[19]:

	customer_id	trans_date	tran_amount
0	CS5295	11-Feb-13	35
1	CS4768	15-Mar-15	39
2	CS2122	26-Feb-13	52
3	CS1217	16-Nov-11	99
4	CS1850	20-Nov-13	78
...	...	...	...
124995	CS8433	26-Jun-11	64
124996	CS7232	19-Aug-14	38
124997	CS8731	28-Nov-14	42
124998	CS8133	14-Dec-13	13
124999	CS7996	13-Dec-14	36

125000 rows × 3 columns

In [20]:

```
response= pd.read_csv('Retail_Data_Response.csv')
```

In [21]:

```
response
```

Out[21]:

	customer_id	response
0	CS1112	0
1	CS1113	0
2	CS1114	1
3	CS1115	1
4	CS1116	1
...	...	...
6879	CS8996	0
6880	CS8997	0
6881	CS8998	0
6882	CS8999	0
6883	CS9000	0

6884 rows × 2 columns

In [22]:

```
a=df.merge(response, on='customer_id', how='left')
```

In [23]:

```
a
```

Out[23]:

	customer_id	trans_date	tran_amount	response
0	CS5295	11-Feb-13	35	1.0
1	CS4768	15-Mar-15	39	1.0
2	CS2122	26-Feb-13	52	0.0
3	CS1217	16-Nov-11	99	0.0
4	CS1850	20-Nov-13	78	0.0
...	...	...	...	...
124995	CS8433	26-Jun-11	64	0.0
124996	CS7232	19-Aug-14	38	0.0
124997	CS8731	28-Nov-14	42	0.0
124998	CS8133	14-Dec-13	13	0.0
124999	CS7996	13-Dec-14	36	0.0

125000 rows × 4 columns

In [26]:

```
a.dtypes
```

Out[26]:

```
customer_id     object
trans_date      object
tran_amount      int64
response       float64
dtype: object
```

In [27]:

```
a.shape
```

Out[27]:

```
(125000, 4)
```

In [28]:

```
a.head()
```

Out[28]:

	customer_id	trans_date	tran_amount	response
0	CS5295	11-Feb-13	35	1.0
1	CS4768	15-Mar-15	39	1.0
2	CS2122	26-Feb-13	52	0.0
3	CS1217	16-Nov-11	99	0.0
4	CS1850	20-Nov-13	78	0.0

In [29]:

```
a.tail()
```

Out[29]:

	customer_id	trans_date	tran_amount	response
124995	CS8433	26-Jun-11	64	0.0
124996	CS7232	19-Aug-14	38	0.0
124997	CS8731	28-Nov-14	42	0.0
124998	CS8133	14-Dec-13	13	0.0
124999	CS7996	13-Dec-14	36	0.0

In [30]:

```
a.describe()
```

Out[30]:

	tran_amount	response
count	125000.000000	124969.000000
mean	64.991912	0.110763
std	22.860006	0.313840
min	10.000000	0.000000
25%	47.000000	0.000000
50%	65.000000	0.000000
75%	83.000000	0.000000
max	105.000000	1.000000

In [31]:

```
a.isnull().sum()
```

Out[31]:

```
customer_id     0
trans_date      0
tran_amount     0
response       31
dtype: int64
```

In [33]:

```
 a=a.dropna()
a
```

Out[33]:

	customer_id	trans_date	tran_amount	response
0	CS5295	11-Feb-13	35	1.0
1	CS4768	15-Mar-15	39	1.0
2	CS2122	26-Feb-13	52	0.0
3	CS1217	16-Nov-11	99	0.0
4	CS1850	20-Nov-13	78	0.0
...	...	...	...	...
124995	CS8433	26-Jun-11	64	0.0
124996	CS7232	19-Aug-14	38	0.0
124997	CS8731	28-Nov-14	42	0.0
124998	CS8133	14-Dec-13	13	0.0
124999	CS7996	13-Dec-14	36	0.0

124969 rows × 4 columns

In [36]:

```
a['trans_date']=pd.to_datetime(a['trans_date'])
a['response']=a['response'].astype('int64')
a
```

```
C:\Users\roxst\AppData\Local\Temp\ipykernel_25684\2217250390.py:1: Settin
gWithCopyWarning:
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-do
cs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (http
s://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returni
ng-a-view-versus-a-copy)
  a['trans_date']=pd.to_datetime(a['trans_date'])
C:\Users\roxst\AppData\Local\Temp\ipykernel_25684\2217250390.py:2: Settin
gWithCopyWarning:
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-do
cs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (http
s://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returni
ng-a-view-versus-a-copy)
  a['response']=a['response'].astype('int64')
```

Out[36]:

	customer_id	trans_date	tran_amount	response
0	CS5295	2013-02-11	35	1
1	CS4768	2015-03-15	39	1
2	CS2122	2013-02-26	52	0
3	CS1217	2011-11-16	99	0
4	CS1850	2013-11-20	78	0
...	...	...	...	...
124995	CS8433	2011-06-26	64	0
124996	CS7232	2014-08-19	38	0
124997	CS8731	2014-11-28	42	0
124998	CS8133	2013-12-14	13	0
124999	CS7996	2014-12-13	36	0

124969 rows × 4 columns

In [37]:

```
set(a['response'])
```

Out[37]:

```
{0, 1}
```

In [38]:

```
a.dtypes
```

Out[38]:

```
customer_id            object
trans_date     datetime64[ns]
tran_amount             int64
response                int64
dtype: object
```

In [40]:

```
from scipy import stats
import numpy as np

z_scores=np.abs(stats.zscore(a['tran_amount']))

threshold=3
outliers=z_scores>threshold
print(a[outliers])
```

```
Empty DataFrame
Columns: [customer_id, trans_date, tran_amount, response]
Index: []
```

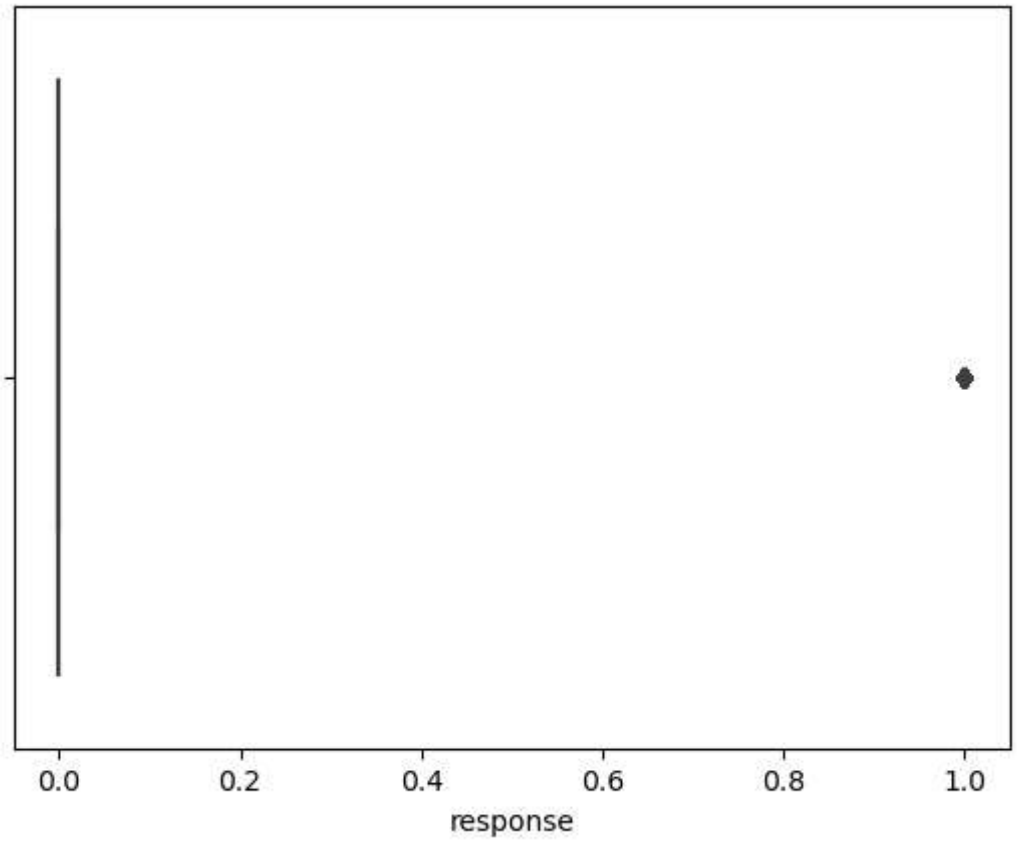
In [41]:

```
from scipy import stats
import numpy as np

z_scores=np.abs(stats.zscore(a['response']))

threshold=3
outliers=z_scores>threshold
print(a[outliers])
```

```
Empty DataFrame
Columns: [customer_id, trans_date, tran_amount, response]
Index: []
```

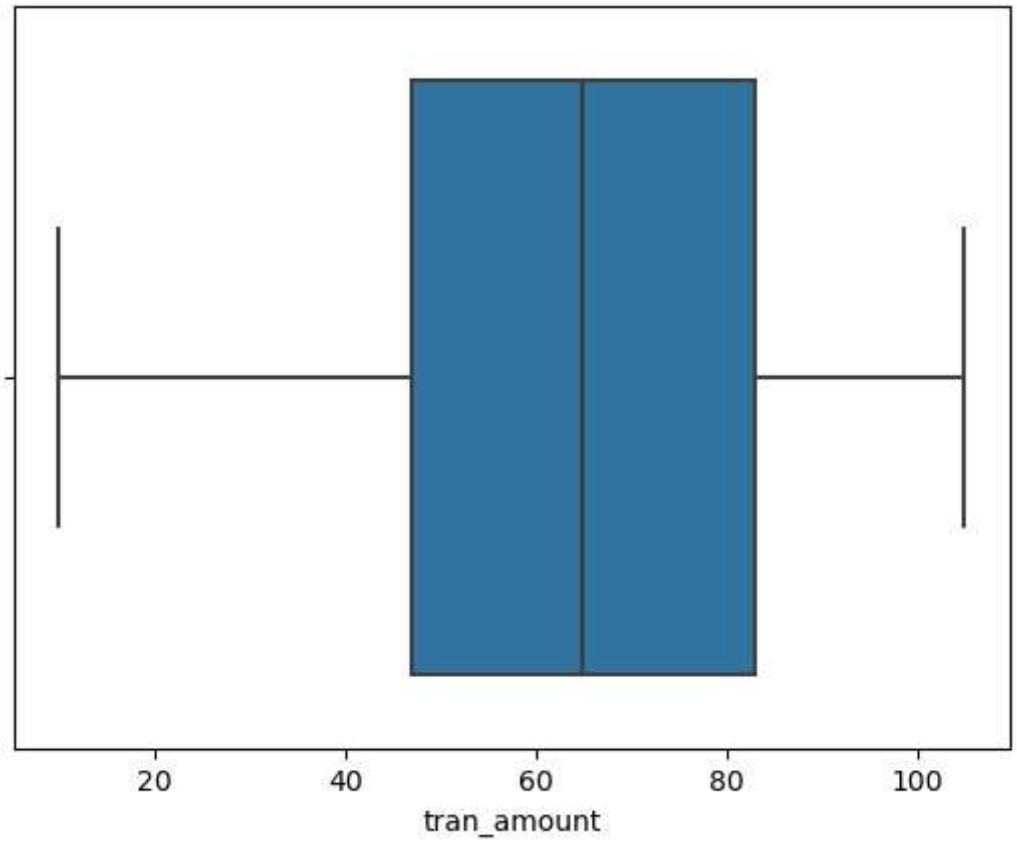
In [42]:

```
import seaborn as sns
import matplotlib.pyplot as plt

sns.boxplot(x=a['response'])
plt.show()
```



In [43]:

```
import seaborn as sns
import matplotlib.pyplot as plt

sns.boxplot(x=a['tran_amount'])
plt.show()
```



In [44]:

```
a['month']=a['trans_date'].dt.month
a
```

```
C:\Users\roxst\AppData\Local\Temp\ipykernel_25684\836628134.py:1: Setting
WithCopyWarning:
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-do
cs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (http
s://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returni
ng-a-view-versus-a-copy)
  a['month']=a['trans_date'].dt.month
```

Out[44]:

	customer_id	trans_date	tran_amount	response	month
0	CS5295	2013-02-11	35	1	2
1	CS4768	2015-03-15	39	1	3
2	CS2122	2013-02-26	52	0	2
3	CS1217	2011-11-16	99	0	11
4	CS1850	2013-11-20	78	0	11
...	...	...	...	...	...
124995	CS8433	2011-06-26	64	0	6
124996	CS7232	2014-08-19	38	0	8
124997	CS8731	2014-11-28	42	0	11
124998	CS8133	2013-12-14	13	0	12
124999	CS7996	2014-12-13	36	0	12

124969 rows × 5 columns

In [46]:

```
monthly_Sales=a.groupby('month')['tran_amount'].sum()
monthly_Sales= monthly_Sales.sort_values(ascending=False).reset_index()
monthly_Sales
```

Out[46]:

	month	tran_amount
0	8	726775
1	10	725058
2	1	724089
3	7	717011
4	12	709795
5	11	698024
6	6	697014
7	9	694201
8	2	645028
9	3	636475
10	5	633162
11	4	515746

In [52]:

```
customer_counts=a['customer_id'].value_counts().reset_index()
customer_counts.columns=['customer_id','count']
```

In [53]:

```
top_5_cus=customer_counts.sort_values(by='count',ascending=False).head(5)
top_5_cus
```

Out[53]:

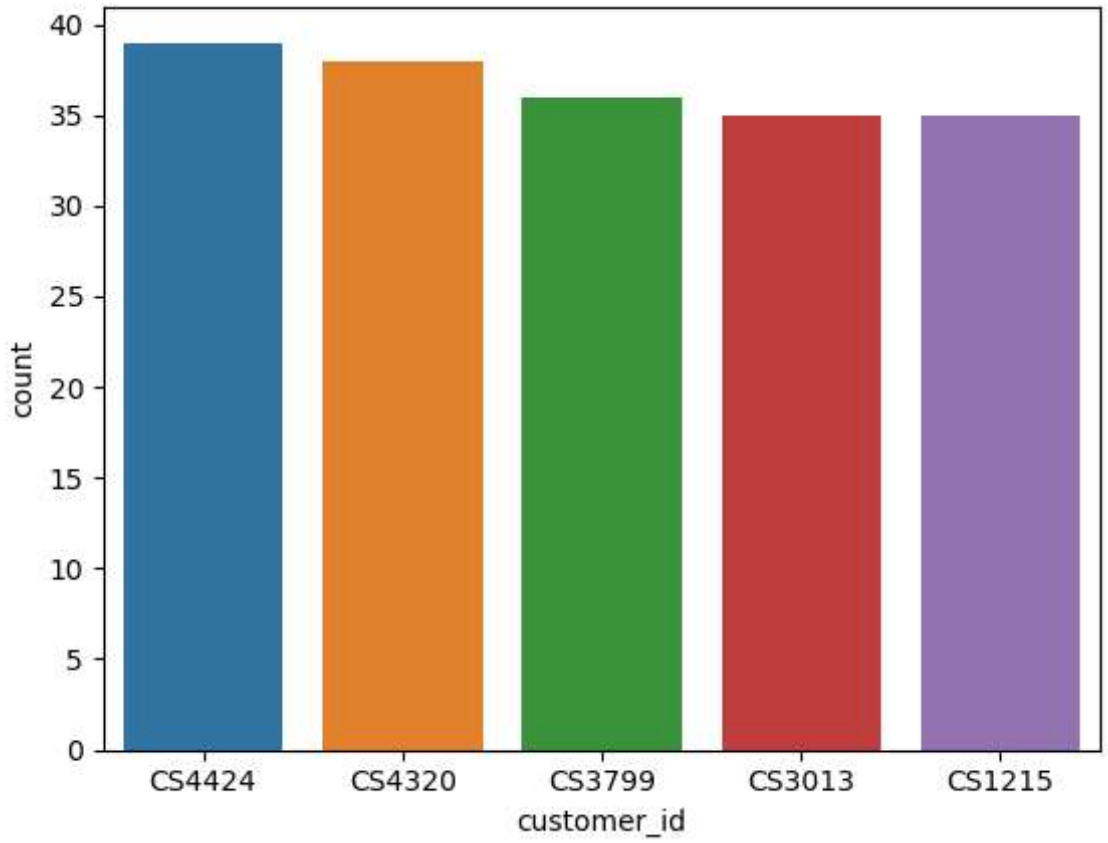
	customer_id	count
0	CS4424	39
1	CS4320	38
2	CS3799	36
3	CS3013	35
4	CS1215	35

In [54]:

```
sns.barplot(x='customer_id',y='count',data=top_5_cus)
```

Out[54]:

```
<Axes: xlabel='customer_id', ylabel='count'>
```



In [57]:

```
customer_sales=a.groupby('customer_id')['tran_amount'].sum().reset_index()
customer_sales
top_5_sal=customer_sales.sort_values(by='tran_amount',ascending=False).head(5)
top_5_sal
```

Out[57]:

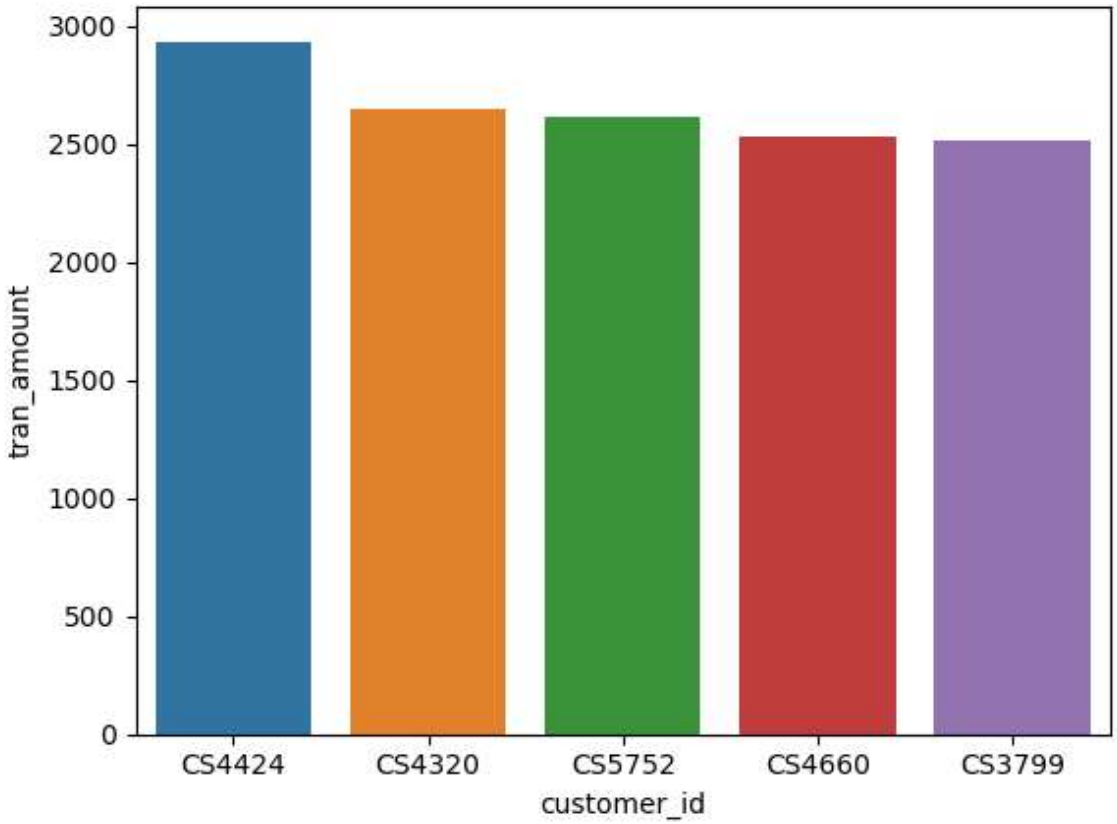
	customer_id	tran_amount
3312	CS4424	2933
3208	CS4320	2647
4640	CS5752	2612
3548	CS4660	2527
2687	CS3799	2513

In [58]:

```
sns.barplot(x='customer_id',y='tran_amount',data=top_5_sal)
```

Out[58]:

```
<Axes: xlabel='customer_id', ylabel='tran_amount'>
```



In [ ]: