```
import numpy as np
data=[1,2,2,1,1,2,3,2,3,1,1,15,3]
mean=np.mean(data)
std=np.std(data)
print('mean is',mean)
print('std is ',std)
threshold=3
outlier=[]
for i in data:
    z=(i-mean)/std
    if z>threshold:
        outlier.append(i)
print('outlier in dataset is',outlier)
```

```
mean is 2.8461538461538463
std is  3.591574624593462
outlier in dataset is [15]
```

Interquartile range to detect outliers in data QR is used to measure variability by dividing data set into quartiles

Q1 represemts the 25th percentile of the data

Q2 represemts the 50th percentile of the data

Q3 represemts the 75th percentile of the data

IQR =Q3-Q1

```
import numpy as np
import seaborn as sns
data=[6,2,3,4,5,1,50]
sort_data=np.sort(data)
sort_data
```

```
array([ 1,  2,  3,  4,  5,  6, 50])
```

```
Q1=np.percentile(data, 25, interpolation ='midpoint')
Q2=np.percentile(data, 50, interpolation ='midpoint')
Q3=np.percentile(data, 75, interpolation ='midpoint')

print('Q1 25 percentile of the given data is, ',Q1)
print('Q2 50 percentile of the given data is, ',Q2)
print('Q3 75 percentile of the given data is, ',Q3)
IQR =Q3-Q1
print('Interquartile rande is ',IQR)
```

```
Q1 25 percentile of the given data is,  2.5
Q2 50 percentile of the given data is,  4.0
Q3 75 percentile of the given data is,  5.5
Interquartile rande is  3.0
```

```
# find the lower and upper limits
low_lim=Q1 -1.5*IQR
up_lim=Q3+1.5*IQR
print('low limit is',low_lim)
print('up limit is',up_lim)
```

```
low limit is -2.0
up limit is 10.0
```

```
Start coding or generate with AI.
```

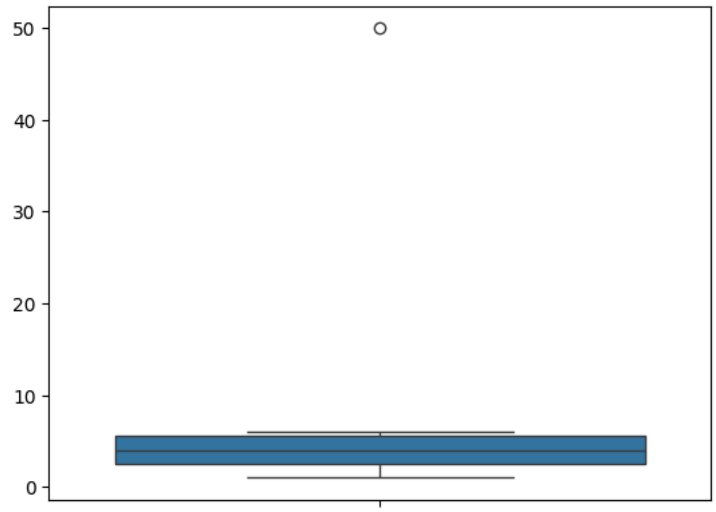
```
#Data points greatewr than lower the upper limit or less than the lower limit

outlier =[]
for x in data:
  if((x >up_lim)or (x<low_lim)):
    outlier.append(x)
  print('outlier in the dataset is',outlier)
```

```
outlier in the dataset is []
outlier in the dataset is []
outlier in the dataset is []
outlier in the dataset is []
outlier in the dataset is []
outlier in the dataset is []
outlier in the dataset is [50]
```

```
#plot the box plot to highlight outliers
sns.boxplot(data)
```

```
<Axes: >
```




```
import pandas as pd
df=pd.read_csv('/content/train.csv')
df
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	493	0	1	Molson, Mr. Harry Markland	male	55.0	0	0	113787	30.5000	C30	S
1	53	1	1	Harper, Mrs. Henry Sleeper (Myna Haxtun)	female	49.0	1	0	PC 17572	76.7292	D33	C
2	388	1	2	Buss, Miss. Kate	female	36.0	0	0	27849	13.0000	NaN	S
3	192	0	2	Carbines, Mr. William	male	19.0	0	0	28424	13.0000	NaN	S
4	687	0	3	Panula, Mr. Jaako Arnold	male	14.0	4	1	3101295	39.6875	NaN	S
...	...	...	...	...	...	...	...	...	...	...	...	...
707	859	1	3	Baclini, Mrs. Solomon (Latifa Qurban)	female	24.0	0	3	2666	19.2583	NaN	C
708	65	0	1	Stewart, Mr. Albert A	male	NaN	0	0	PC 17605	27.7208	NaN	C
709	130	0	3	Ekstrom, Mr. Johan	male	45.0	0	0	347061	6.9750	NaN	S
710	21	0	2	Fynney, Mr. Joseph J	male	35.0	0	0	239865	26.0000	NaN	S
711	476	0	1	Clifford, Mr. George Quincy	male	NaN	0	0	110465	52.0000	A14	S

712 rows × 12 columns

```
#For single column
df.Cabin.duplicated()
```

```
0      False
1      False
2      False
3       True
4       True
       ...
707     True
708     True
709     True
710     True
```

```
711    False
Name: Cabin, Length: 712, dtype: bool
```

```
# For a dataframe as a whole
df.duplicated()
```

```
0      False
1      False
2      False
3      False
4      False
       ...
707    False
708    False
709    False
710    False
711    False
Length: 712, dtype: bool
```

```
df.duplicated(subset=['Survived','Pclass','Sex'])
```

```
0      False
1      False
2      False
3      False
4      False
       ...
707     True
708     True
709     True
710     True
711     True
Length: 712, dtype: bool
```

```
df.Cabin.duplicated().sum()
```

```
583
```

```
df.duplicated().sum()
```

```
0
```

```
df.loc[df.duplicated(keep='first'),:]
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked

```
df.loc[df.duplicated(keep=False),:]
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked