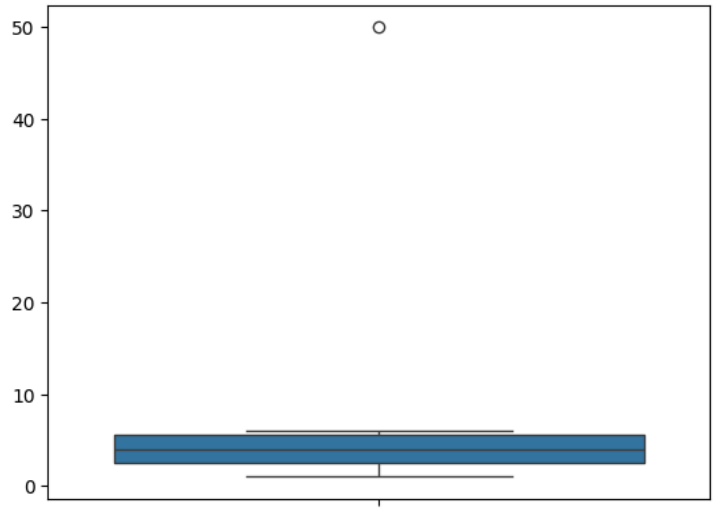
```
import numpy as np
data=[1,2,2,1,1,2,3,2,3,1,1,15,3]
mean=np.mean(data)
std=np.std(data)
print('mean is',mean)
print('std is ',std)
threshold=3
outlier=[]
for i in data:
    z=(i-mean)/std
    if z>threshold:
        outlier.append(i)
print('outlier in dataset is',outlier)
```

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

interquartile range to detect outliers in datA. REPRESENT the 25th quartile of the data. represents the 50 th percentile of the data. represents the 75 th percentile of the data. if a datset has 2n/2n+1 data points,then Q1=median of the dataset Q2=median of n smallest data points. Q3=median of n highest data points. IQR is the range between the first and the third quartiles namely Q1 and Q3: IQR=Q3-Q1

```
#step1:Import necessary librarians.
import numpy as np
import seaborn as sns
#step 2:take the data and sort it in ascending order.
data=[6,2,3,4,5,1,50]
sort_data=np.sort(data)
sort_data
#step3: calculate Q1,Q2,Q3, and IQR.
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 range is ',IQR)
#step 4:find the lower and upper limits as Q1-1.5 IQR and Q3+1.5 IQR ,respectively
low_lim=Q1-1.5*IQR
up_lim=Q3+1.5*IQR
print('low_limit is',low_lim)
print('up_limit is',up_lim)
#step 5: data points greater than the upper limit or less than the lowerlimit are
outlier=[]
for x in data:
  if((x>up_lim) or (x<low_lim)):
    outlier.append(x)
  print('outlier in the dataset is',outlier)
#step6: plot the box plot to the highlight outliers.
sns.boxplot(data)
```

```
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 range is  3.0
low_limit is -2.0
up_limit is 10.0
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]
<Axes: >
```




```
import pandas as pd
def load_data():
  df_all=pd.read_csv('/content/train.csv')
  df
  #take the subset
  return df_all.loc[:300,['Survived','Pclass','Sex','Cabin','Embarked']]
  #load the subset
df=load_data()
#fpr single column
df.Cabin.duplicated()
```

```
0      False
1      False
2      False
3       True
4       True
       ...
296     True
297     True
298     True
299    False
300    False
Name: Cabin, Length: 301, dtype: bool
```

```
#to consu=ider cetrian columns for identifying duplicates
df.duplicated(subset=['Survived','Pclass','Sex'])
```

```
0      False
1      False
2      False
3      False
4      False
       ...
296     True
297     True
298     True
299     True
300     True
Length: 301, dtype: bool
```

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

```
230
```

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

```
199
```

```
#keep defaults to 'first;
df.loc[df.duplicated(keep='first'),:]
```

	Survived	Pclass	Sex	Cabin	Embarked
5	1	2	female	NaN	S
6	0	3	male	NaN	S
7	0	2	male	NaN	S
11	0	3	male	NaN	S
12	0	3	male	NaN	S
...	...	...	...	...	...
294	0	2	female	NaN	S
295	1	3	female	NaN	C
296	1	3	female	NaN	S
297	0	3	male	NaN	S
298	1	3	male	NaN	S