```
import pandas as pd
import numpy as np
df = pd.DataFrame(np.random.randn(5,3),index =['a','b','e','f','h'],columns = ['one','two','three'])
df = df.reindex(['a','b','c','d','e','f','g','h'])
print(df['one'].isnull())
```

```
a    False
b    False
c     True
d     True
e    False
f    False
g     True
h    False
Name: one, dtype: bool
```

```
df = pd.DataFrame(np.random.randn(5,3),index =['a','b','e','f','h'],columns = ['one','two','three'])
print(df)
df = df.reindex(['a','b','c','d','e','f','g','h'])
print(df)
```

```
        one       two     three
a  0.196811  0.026280  0.073417
b -1.941607 -1.849357 -1.540346
e  0.292737 -0.196345  0.485063
f -0.110042 -0.029726 -1.085315
h -0.859521  0.262806 -0.428103
        one       two     three
a  0.196811  0.026280  0.073417
b -1.941607 -1.849357 -1.540346
c       NaN       NaN       NaN
d       NaN       NaN       NaN
e  0.292737 -0.196345  0.485063
f -0.110042 -0.029726 -1.085315
g       NaN       NaN       NaN
h -0.859521  0.262806 -0.428103
```

```
df = pd.DataFrame(np.random.randn(5,3),index =['a','b','e','f','h'],columns = ['one','two','three'])
df = df.reindex(['a','b','c','d','e','f','g','h'])
print(df)
print('---------')
print(df.fillna(method='pad'))
```

```
        one       two     three
a -0.526320  1.078405  1.373390
b  1.558215  0.040224  1.794562
c       NaN       NaN       NaN
d       NaN       NaN       NaN
e  0.944129  0.711445 -0.453592
f  0.690681 -1.194808  0.188242
g       NaN       NaN       NaN
h  0.014021  0.041830  1.482654
---------
        one       two     three
a -0.526320  1.078405  1.373390
b  1.558215  0.040224  1.794562
c  1.558215  0.040224  1.794562
d  1.558215  0.040224  1.794562
e  0.944129  0.711445 -0.453592
f  0.690681 -1.194808  0.188242
g  0.690681 -1.194808  0.188242
h  0.014021  0.041830  1.482654
```

```
df = pd.DataFrame(np.random.randn(5,3),index =['a','b','e','f','h'],columns = ['one','two','three'])
df = df.reindex(['a','b','c','d','e','f','g','h'])
print(df.fillna(method='bfill'))
```

```
        one       two     three
a  0.394044 -0.933484 -0.508881
b -0.170647 -1.147017 -0.027974
c -0.637137  0.317871  0.372641
d -0.637137  0.317871  0.372641
e -0.637137  0.317871  0.372641
f  2.497916  0.210504  0.125328
g -0.988982 -1.251751 -2.434725
h -0.988982 -1.251751 -2.434725
```

```
df = pd.DataFrame(np.random.randn(5,3),index =['a','b','e','f','h'],columns = ['one','two','three'])
df = df.reindex(['a','b','c','d','e','f','g','h'])
print(df)
print(df.dropna())
```

```
        one       two     three
a  0.080449 -0.463041 -1.015137
```

```
b -1.585621 -1.078766 -0.377185
c       NaN       NaN       NaN
d       NaN       NaN       NaN
e -0.474628  0.972166 -0.031023
f  1.945806  0.843586 -2.202388
g       NaN       NaN       NaN
h  1.784514  1.212007 -0.264692
        one       two     three
a  0.080449 -0.463041 -1.015137
b -1.585621 -1.078766 -0.377185
e -0.474628  0.972166 -0.031023
f  1.945806  0.843586 -2.202388
h  1.784514  1.212007 -0.264692
```

```
import pandas as pd
import numpy as np
df =pd.read_csv('/content/2,1 dataset titanic.csv')
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 12 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   PassengerId  891 non-null    int64  
 1   Survived     891 non-null    int64  
 2   Pclass       891 non-null    int64  
 3   Name         891 non-null    object 
 4   Sex          891 non-null    object 
 5   Age          714 non-null    float64
 6   SibSp        891 non-null    int64  
 7   Parch        891 non-null    int64  
 8   Ticket       891 non-null    object 
 9   Fare         891 non-null    float64
 10  Cabin        204 non-null    object 
 11  Embarked     889 non-null    object 
dtypes: float64(2), int64(5), object(5)
memory usage: 83.7+ KB
```

```
cols =  ['Name','Ticket','Cabin']
df = df.drop(cols,axis=1)
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 9 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   PassengerId  891 non-null    int64  
 1   Survived     891 non-null    int64  
 2   Pclass       891 non-null    int64  
 3   Sex          891 non-null    object 
 4   Age          714 non-null    float64
 5   SibSp        891 non-null    int64  
 6   Parch        891 non-null    int64  
 7   Fare         891 non-null    float64
 8   Embarked     889 non-null    object 
dtypes: float64(2), int64(5), object(2)
memory usage: 62.8+ KB
```

```
df = df.dropna()
df.info(0)
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 712 entries, 0 to 890
Data columns (total 9 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   PassengerId  712 non-null    int64  
 1   Survived     712 non-null    int64  
 2   Pclass       712 non-null    int64  
 3   Sex          712 non-null    object 
 4   Age          712 non-null    float64
 5   SibSp        712 non-null    int64  
 6   Parch        712 non-null    int64  
 7   Fare         712 non-null    float64
 8   Embarked     712 non-null    object 
dtypes: float64(2), int64(5), object(2)
memory usage: 55.6+ KB
```

```
dummies = []
cols = ['Pclass','Sex','Embarked']
for col in cols:
  dummies.append(pd.get_dummies(df[col]))
```

```
#transfor the eight columns
titanic_dummies = pd.concat(dummies,axis=1)
```

```
#concatenate the values with data frame
df = pd.concat((df,titanic_dummies),axis =1)
df
```

	PassengerId	Survived	Pclass	Sex	Age	SibSp	Parch	Fare	Embarked	1	2
0	1	0	3	male	22.0	1	0	7.2500	S	0	0
1	2	1	1	female	38.0	1	0	71.2833	C	1	0
2	3	1	3	female	26.0	0	0	7.9250	S	0	0
3	4	1	1	female	35.0	1	0	53.1000	S	1	0
4	5	0	3	male	35.0	0	0	8.0500	S	0	0
...	...	...	...	...	...	...	...	...	...	...	...
885	886	0	3	female	39.0	0	5	29.1250	Q	0	0
886	887	0	2	male	27.0	0	0	13.0000	S	0	1
887	888	1	1	female	19.0	0	0	30.0000	S	1	0
889	890	1	1	male	26.0	0	0	30.0000	C	1	0
890	891	0	3	male	32.0	0	0	7.7500	Q	0	0

712 rows × 17 columns

```
#remove the unwanted cols
df = df.drop(['Pclass','Sex','Embarked'],axis=1)
```

```
df.head()
```

	PassengerId	Survived	Age	SibSp	Parch	Fare	1	2	3	female	male	C	Q	S
0	1	0	22.0	1	0	7.2500	0	0	1	0	1	0	0	1
1	2	1	38.0	1	0	71.2833	1	0	0	1	0	1	0	0
2	3	1	26.0	0	0	7.9250	0	0	1	1	0	0	0	1
3	4	1	35.0	1	0	53.1000	1	0	0	1	0	0	0	1
4	5	0	35.0	0	0	8.0500	0	0	1	0	1	0	0	1

```
df ['Age']=df['Age'].interpolate()
```

```
df.Age.isnull().sum()
```

```
0
```

```
from sklearn.preprocessing  import MinMaxScaler
df = [[-1,2],[-0.5,6],[0,10],[1,18]]
scaler = MinMaxScaler()
print(scaler.fit(df))
print("---------")
MinMaxScaler()
print(scaler.data_max_)
print("----------")
```

```
MinMaxScaler()
---------
[ 1. 18.]
----------
```

```
from numpy import asarray
from sklearn.preprocessing import StandardScaler
#define df
df = asarray([[100,0.001],[8,0.05],[50,0.005],[88,0.07],[4,0.1]])
print(df)
#define standard scaler
scaler = StandardScaler()
#transform data
scaled = scaler.fit_transform(df)
print(scaled)
```

```
[[1.0e+02 1.0e-03]
 [8.0e+00 5.0e-02]
 [5.0e+01 5.0e-03]
 [8.8e+01 7.0e-02]
 [4.0e+00 1.0e-01]]
[[ 1.26398112 -1.16389967]
 [-1.06174414  0.12639634]
 [ 0.         -1.05856939]
 [ 0.96062565  0.65304778]
 [-1.16286263  1.44302493]]
```

Start coding or generate with AI.

```
#23-02-2023
import numpy as np
data = [1,2,2,2,2,3,1,1,15,2,2,2,3,1,1,2]
mean = np.mean(data)
std = np.std(data)
print('mean of the dataset is ',mean)
print('std. deviation 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 of the dataset is  2.625
std. deviation is  3.2572035551988456
outlier in dataset is [15]
```

interquaritle range to detect outliers in data

Q1 represents the 25th percentile of the data Q2 represents the 50th percentile of the data Q3 represents the 75th percentile of the data

---

if a dataset has 2n/2n+1 datapoints,then Q1 = median of the dataset Q2 = median of n smallest datapoints Q3 = median of n highest datapoints

---

IQR is the range between the first and the third quartiles namely Q1 and Q3 IQR = Q3-Q1

```
# step 1: import necessary libraries.
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
```

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

```
#step 3: 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('Q1 25 percentile of the given data is , ',Q2)
print('Q1 25 percentile of the given data is , ',Q3)

IQR = Q3-Q1
print('interquartile Range is',IQR)
```

```
Q1 25 percentile of the given data is ,  2.5
Q1 25 percentile of the given data is ,  4.0
Q1 25 percentile of the given data is ,  5.5
interquartile Range is 3.0
```

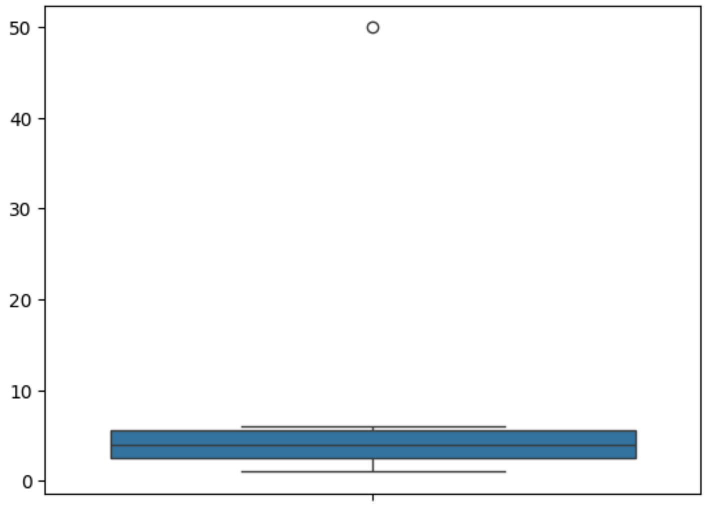
```
#step 4: find the lower and upper limits as Q1-1.5 IQR and Q3+1.5 IQR respectivey

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
```

```
#step 5: data points greater than the upper limit
```

```
#step 6: plot the box plot to highlight outliers
sns.boxplot(data)
```

```
<Axes: >
```



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