

```
In [84]: import pandas as pd
```

```
In [85]: import numpy as np
```

```
In [86]: data= pd.DataFrame()
```

```
In [87]: data['Rollno']=[1,2,3,4,5,6,7,8,9,10]
data['Maths']=[66,85,78,60,45,56,70,np.nan,80,110]
data['Science']=[90,83,46,78,84,57,68,43,67,58]
data['English']=[79,86,57,66,49,87,73,69,52,68]
data['Attendance']=[90,80,74,86,'93%',88,69,77,95,96]
```

```
In [88]: data
```

Out[88]:

	Rollno	Maths	Science	English	Attendance
0	1	66.0	90	79	90
1	2	85.0	83	86	80
2	3	78.0	46	57	74
3	4	60.0	78	66	86
4	5	45.0	84	49	93%
5	6	56.0	57	87	88
6	7	70.0	68	73	69
7	8	NaN	43	69	77
8	9	80.0	67	52	95
9	10	110.0	58	68	96

```
In [89]: data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10 entries, 0 to 9
Data columns (total 5 columns):
#   Column      Non-Null Count  Dtype
---  ---
0   Rollno      10 non-null    int64
1   Maths       9 non-null     float64
2   Science     10 non-null    int64
3   English     10 non-null    int64
4   Attendance  10 non-null    object
dtypes: float64(1), int64(3), object(1)
memory usage: 528.0+ bytes
```

In [90]: `data.describe()`

Out[90]:

	Rollno	Maths	Science	English
<b>count</b>	10.000000	9.000000	10.000000	10.000000
<b>mean</b>	5.500000	72.222222	67.400000	68.600000
<b>std</b>	3.02765	18.978789	16.304055	13.209424
<b>min</b>	1.000000	45.000000	43.000000	49.000000
<b>25%</b>	3.250000	60.000000	57.250000	59.250000
<b>50%</b>	5.500000	70.000000	67.500000	68.500000
<b>75%</b>	7.750000	80.000000	81.750000	77.500000
<b>max</b>	10.000000	110.000000	90.000000	87.000000

In [91]: `data.isnull().sum()`

Out[91]:

Rollno	0
Maths	1
Science	0
English	0
Attendance	0
dtype:	int64

In [92]: `data['Maths'].fillna(data['Maths'].mean(),inplace=True)`

In [93]: `data`

Out[93]:

	Rollno	Maths	Science	English	Attendance
<b>0</b>	1	66.000000	90	79	90
<b>1</b>	2	85.000000	83	86	80
<b>2</b>	3	78.000000	46	57	74
<b>3</b>	4	60.000000	78	66	86
<b>4</b>	5	45.000000	84	49	93%
<b>5</b>	6	56.000000	57	87	88
<b>6</b>	7	70.000000	68	73	69
<b>7</b>	8	72.222222	43	69	77
<b>8</b>	9	80.000000	67	52	95
<b>9</b>	10	110.000000	58	68	96

In [94]: `data.isnull().sum()`

Out[94]:

Rollno	0
Maths	0
Science	0
English	0
Attendance	0
dtype:	int64

In [95]: `data['Attendance']=pd.to_numeric(data['Attendance'],errors='coerce')`

In [96]: `data`

Out[96]:

	Rollno	Maths	Science	English	Attendance
0	1	66.000000	90	79	90.0
1	2	85.000000	83	86	80.0
2	3	78.000000	46	57	74.0
3	4	60.000000	78	66	86.0
4	5	45.000000	84	49	NaN
5	6	56.000000	57	87	88.0
6	7	70.000000	68	73	69.0
7	8	72.222222	43	69	77.0
8	9	80.000000	67	52	95.0
9	10	110.000000	58	68	96.0

In [97]: `data['Attendance'].fillna(data['Attendance'].mean(),inplace=True)`

In [98]: `data`

Out[98]:

	Rollno	Maths	Science	English	Attendance
0	1	66.000000	90	79	90.000000
1	2	85.000000	83	86	80.000000
2	3	78.000000	46	57	74.000000
3	4	60.000000	78	66	86.000000
4	5	45.000000	84	49	83.888889
5	6	56.000000	57	87	88.000000
6	7	70.000000	68	73	69.000000
7	8	72.222222	43	69	77.000000
8	9	80.000000	67	52	95.000000
9	10	110.000000	58	68	96.000000

```
In [99]: data.describe()
```

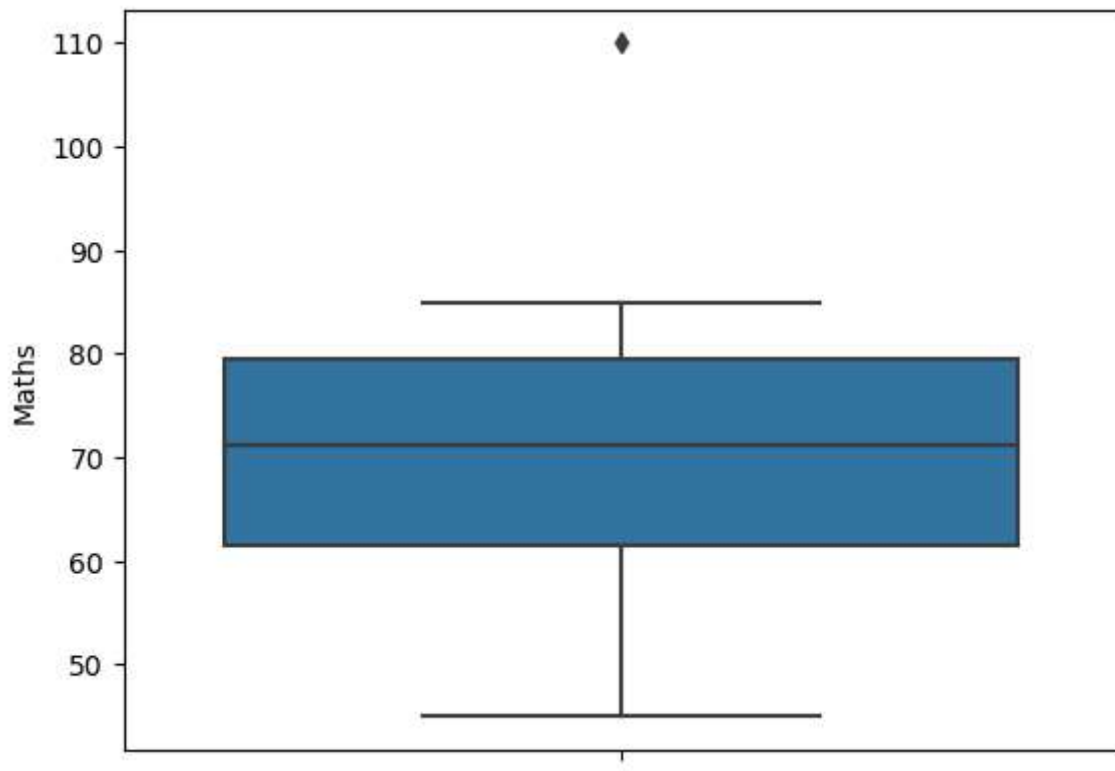
```
Out[99]:
```

	Rollno	Maths	Science	English	Attendance
count	10.00000	10.000000	10.000000	10.000000	10.000000
mean	5.50000	72.222222	67.400000	68.600000	83.888889
std	3.02765	17.893374	16.304055	13.209424	8.887500
min	1.00000	45.000000	43.000000	49.000000	69.000000
25%	3.25000	61.500000	57.250000	59.250000	77.750000
50%	5.50000	71.111111	67.500000	68.500000	84.944444
75%	7.75000	79.500000	81.750000	77.500000	89.500000
max	10.00000	110.000000	90.000000	87.000000	96.000000

```
In [100]: import seaborn as sns
```

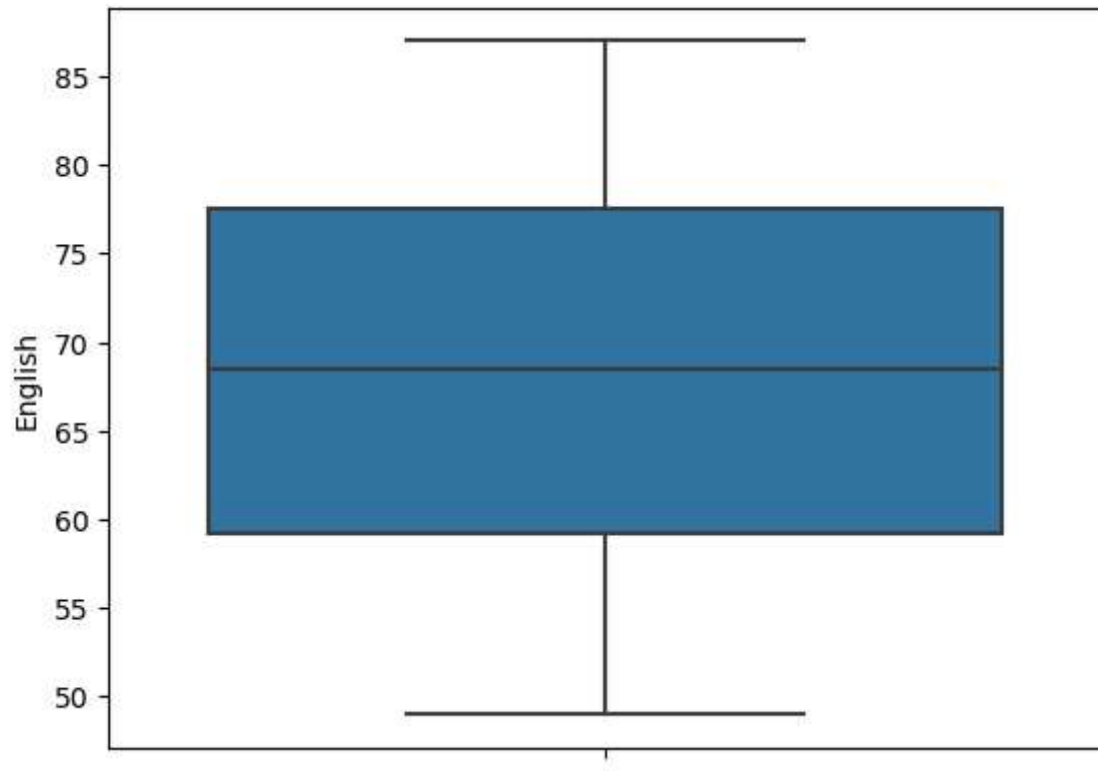
```
In [101]: sns.boxplot(y=data['Maths'])
```

```
Out[101]: <Axes: ylabel='Maths'>
```



```
In [102]: sns.boxplot(y=data['English'])
```

```
Out[102]: <Axes: ylabel='English'>
```



```
In [103]: Q1=data['Maths'].quantile(0.25)
Q3=data['Maths'].quantile(0.75)
IQR = Q3-Q1
lower_bound =Q1-1.1*IQR
upper_bound =Q3+1.1*IQR
```

```
In [104]: lower_bound
```

```
Out[104]: 41.7
```

```
In [105]: upper_bound
```

```
Out[105]: 99.3
```

```
In [111]: data['Maths'] = np.where(data['Maths'] > upper_bound, upper_bound,
np.where(data['Maths'] < lower_bound, lower_bound, data['Maths']))
```

```
In [112]: data
```

```
Out[112]:
```

	Rollno	Maths	Science	English	Attendance
0	1	66.000000	90	79	90.000000
1	2	85.000000	83	86	80.000000
2	3	78.000000	46	57	74.000000
3	4	60.000000	78	66	86.000000
4	5	45.000000	84	49	83.888889
5	6	56.000000	57	87	88.000000
6	7	70.000000	68	73	69.000000
7	8	72.222222	43	69	77.000000
8	9	80.000000	67	52	95.000000
9	10	99.300000	58	68	96.000000

```
In [113]: from sklearn.preprocessing import MinMaxScaler
```

```
In [114]: scaler =MinMaxScaler()  
data[['Attendance']] =scaler.fit_transform(data[['Attendance']])
```

```
In [115]: data
```

```
Out[115]:
```

	Rollno	Maths	Science	English	Attendance
0	1	66.000000	90	79	0.777778
1	2	85.000000	83	86	0.407407
2	3	78.000000	46	57	0.185185
3	4	60.000000	78	66	0.629630
4	5	45.000000	84	49	0.551440
5	6	56.000000	57	87	0.703704
6	7	70.000000	68	73	0.000000
7	8	72.222222	43	69	0.296296
8	9	80.000000	67	52	0.962963
9	10	99.300000	58	68	1.000000

```
In [ ]:
```