

```

import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt

a = pd.read_csv("/home/mmcoe/Downloads/StudentsPerformance_m.csv")

a.isnull().sum()

gender                0
race/ethnicity         0
parental level of education  0
lunch                 0
test preparation course  0
math score            2
reading score         4
writing score         2
dtype: int64

a.isnull().values.sum()

8

a['math score'].fillna(value=a['math score'].mean(),inplace=True)

a.isnull().values.sum()

6

a['reading score'].fillna(value=a['reading
score'].mean(),inplace=True)

a.isnull().values.sum()

2

a['writing score'].fillna(value=a['writing
score'].mean(),inplace=True)

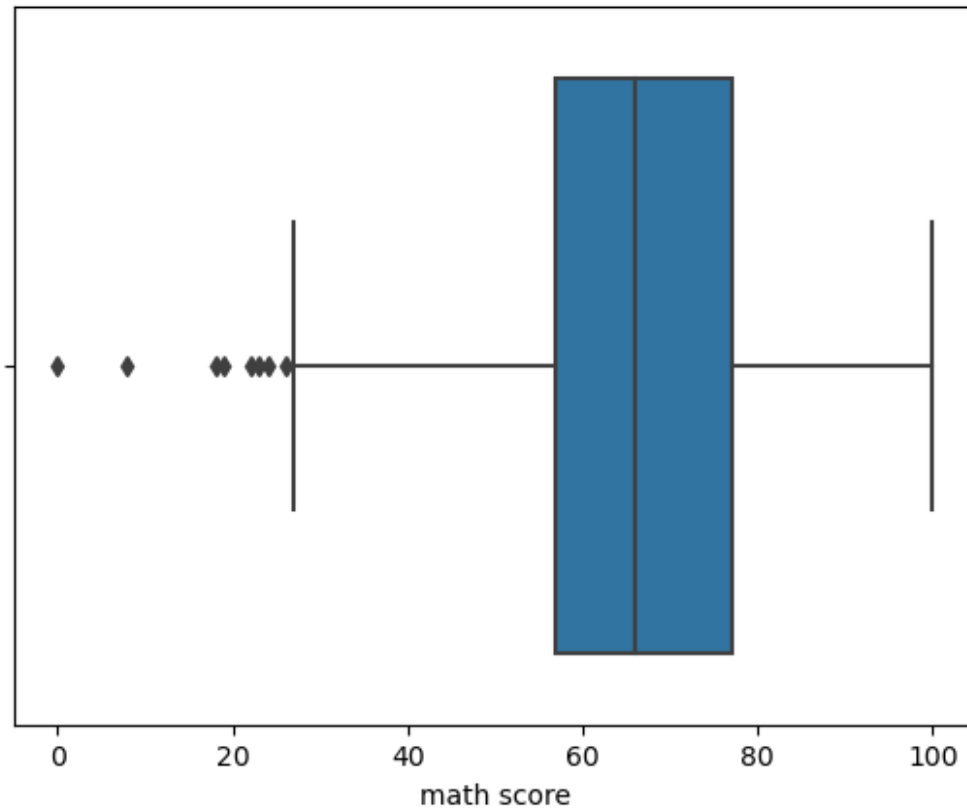
a.isnull().values.sum()

0

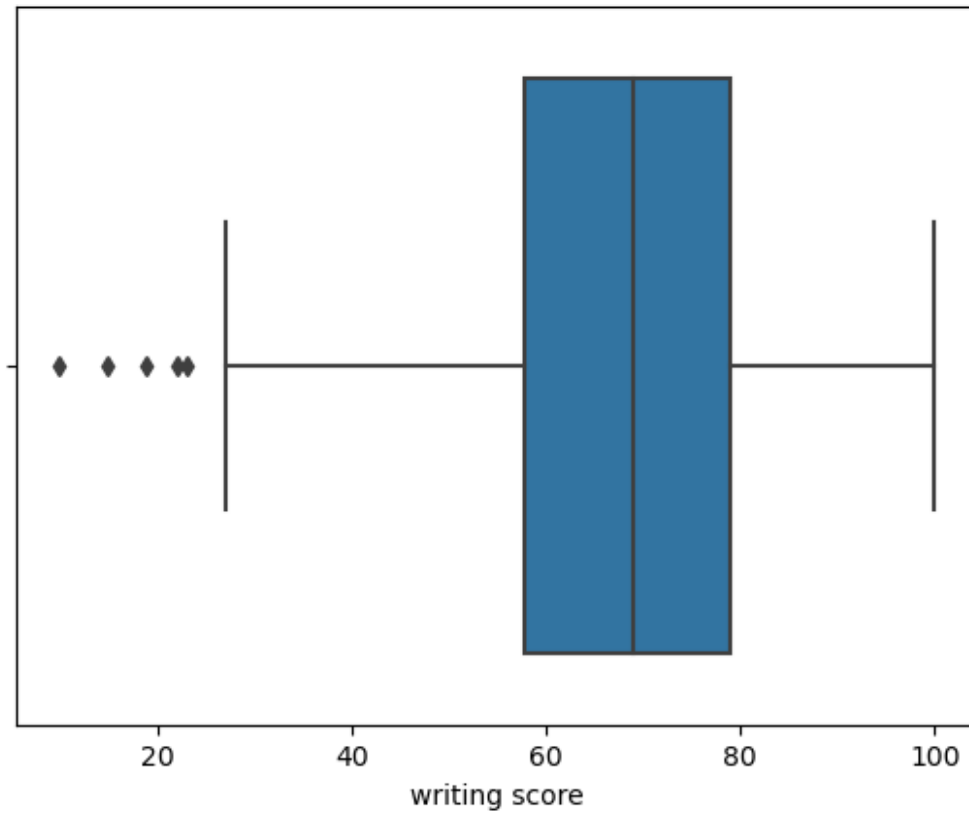
sns.boxplot(x=a['math score'])

<AxesSubplot:xlabel='math score'>

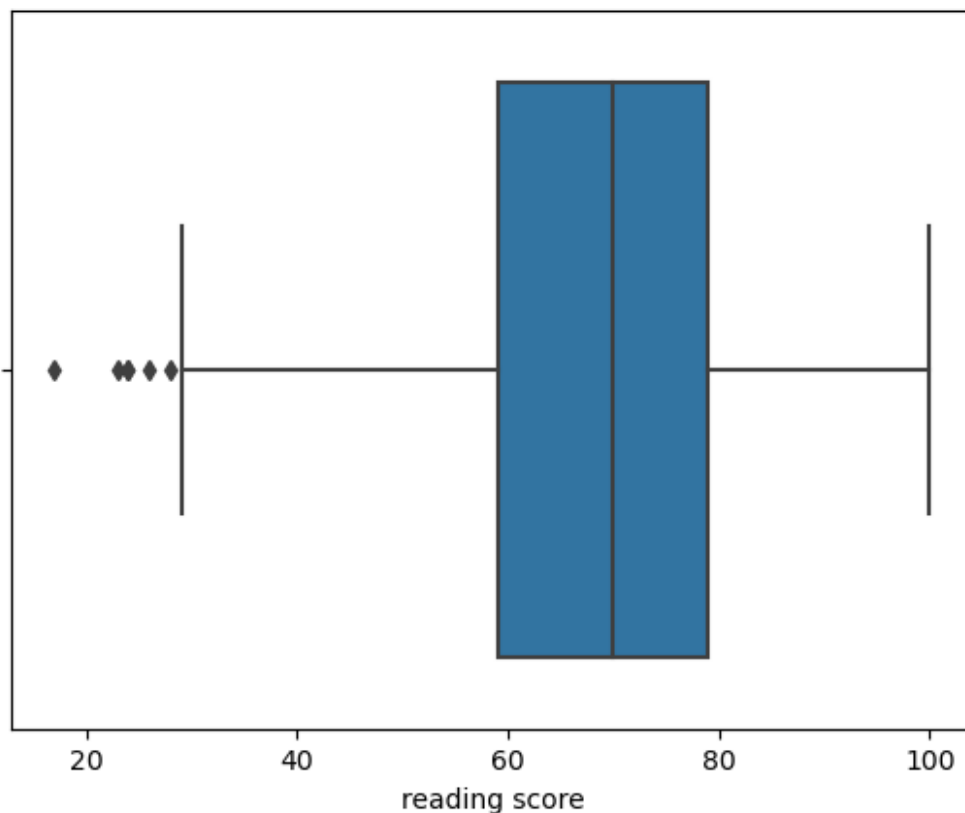
```



```
sns.boxplot(x=a['writing score'])  
<AxesSubplot:xlabel='writing score'>
```



```
sns.boxplot(x=a['reading score'])  
<AxesSubplot:xlabel='reading score'>
```



```
a.info()
```

```
<class 'pandas.core.frame.DataFrame'>
```

```
RangeIndex: 1000 entries, 0 to 999
```

```
Data columns (total 8 columns):
```

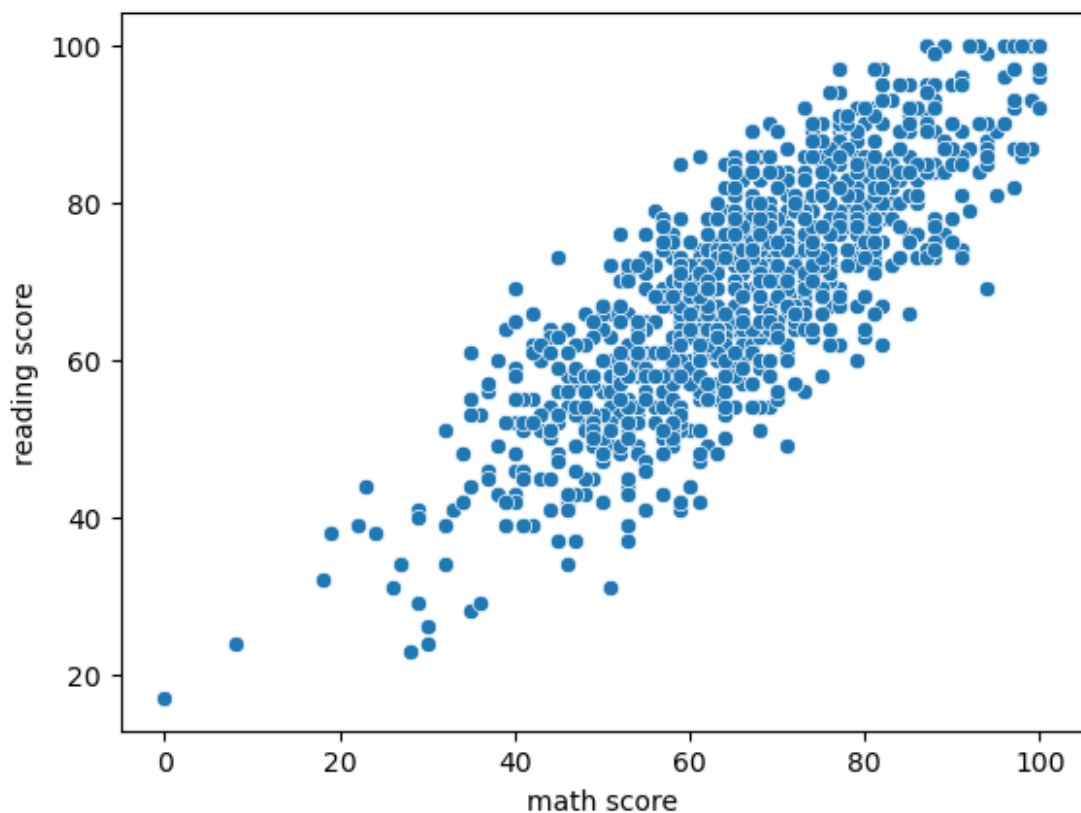
#	Column	Non-Null Count	Dtype
0	gender	1000 non-null	object
1	race/ethnicity	1000 non-null	object
2	parental level of education	1000 non-null	object
3	lunch	1000 non-null	object
4	test preparation course	1000 non-null	object
5	math score	1000 non-null	float64
6	reading score	1000 non-null	float64
7	writing score	1000 non-null	float64

```
dtypes: float64(3), object(5)
```

```
memory usage: 62.6+ KB
```

```
sns.scatterplot(data=a, x="math score", y = "reading score")
```

```
<AxesSubplot:xlabel='math score', ylabel='reading score'>
```



a

	gender	race/ethnicity	parental level of education	
lunch \				
0	female	group B	bachelor's degree	standard
1	female	group C	some college	standard
2	female	group B	master's degree	standard
3	male	group A	associate's degree	free/reduced
4	male	group C	some college	standard
..
995	female	group E	master's degree	standard
996	male	group C	high school	free/reduced
997	female	group C	high school	free/reduced
998	female	group D	some college	standard

999	female	group D		some college	free/reduced
	test preparation course	math score	reading score	writing score	
0	none	72.0	72.0	74.0	
1	completed	69.0	90.0	88.0	
2	none	90.0	95.0	93.0	
3	none	47.0	57.0	44.0	
4	none	76.0	78.0	75.0	
..	
995	completed	88.0	99.0	95.0	
996	none	62.0	55.0	55.0	
997	completed	59.0	71.0	65.0	
998	completed	68.0	78.0	77.0	
999	none	77.0	86.0	86.0	

[1000 rows x 8 columns]

```

# it will calculate the Q1
q1 = a['math score'].quantile(0.25)

# It will calculate the Q3
q3 = a['math score'].quantile(0.75)

# IQR is Inter Quatile range.
IQR = q3-q1

print(IQR)

20.0

# It will return the values less than the Q1
a['math score']<(q1-1.5*IQR)

0      False
1      False
2      False
3      False

```

```

4      False
...
995    False
996    False
997    False
998    False
999    False
Name: math score, Length: 1000, dtype: bool

```

```

# It will return the values less than the Q3
a['math score']>(q3-1.5*IQR)

```

```

0      True
1      True
2      True
3     False
4      True
...
995    True
996    True
997    True
998    True
999    True
Name: math score, Length: 1000, dtype: bool

```

```

# It will store all the values without outliers in a2 of all columns.

```

```

a2 = a[(a['math score']< (q1-1.5*IQR)) | (a['math score']> (q3-
1.5*IQR))]

```

```

a2

```

	gender	race/ethnicity	parental level of education	lunch \
0	female	group B	bachelor's degree	standard
1	female	group C	some college	standard
2	female	group B	master's degree	standard
4	male	group C	some college	standard
5	female	group B	associate's degree	standard
..
995	female	group E	master's degree	standard
996	male	group C	high school	free/reduced

997	female	group C	high school	free/reduced
998	female	group D	some college	standard
999	female	group D	some college	free/reduced

	test preparation course	math score	reading score	writing score
0	none	72.0	72.0	74.0
1	completed	69.0	90.0	88.0
2	none	90.0	95.0	93.0
4	none	76.0	78.0	75.0
5	none	71.0	83.0	78.0
..
995	completed	88.0	99.0	95.0
996	none	62.0	55.0	55.0
997	completed	59.0	71.0	65.0
998	completed	68.0	78.0	77.0
999	none	77.0	86.0	86.0

[901 rows x 8 columns]

a2.shape

(901, 8)

```
import scipy.stats as stats
```

calculating value of z using stats function of scipy library. Here, abs is a absolute function which will return the positive value. abs is a function presesnt in numpt library.

```
z = np.abs(stats.zscore(a['writing score']))
```

```
con = np.where(z<3)
```

```
a3 = a.iloc[con]
```

a3

	gender	race/ethnicity	parental level of education	
lunch \				
0	female	group B	bachelor's degree	standard
1	female	group C	some college	standard
2	female	group B	master's degree	standard
3	male	group A	associate's degree	free/reduced
4	male	group C	some college	standard
..
995	female	group E	master's degree	standard
996	male	group C	high school	free/reduced
997	female	group C	high school	free/reduced
998	female	group D	some college	standard
999	female	group D	some college	free/reduced

	test preparation course	math score	reading score	writing score
0	none	72.0	72.0	74.0
1	completed	69.0	90.0	88.0
2	none	90.0	95.0	93.0
3	none	47.0	57.0	44.0
4	none	76.0	78.0	75.0
..
995	completed	88.0	99.0	95.0
996	none	62.0	55.0	55.0
997	completed	59.0	71.0	65.0
998	completed	68.0	78.0	77.0
999	none	77.0	86.0	86.0

```
[1000 rows x 8 columns]
```

```
z
```

```
0      0.392081
1      1.314021
2      1.643285
3      1.583502
4      0.457934
```

```
      ...
995    1.774990
996    0.859122
997    0.200594
998    0.589640
999    1.182315
```

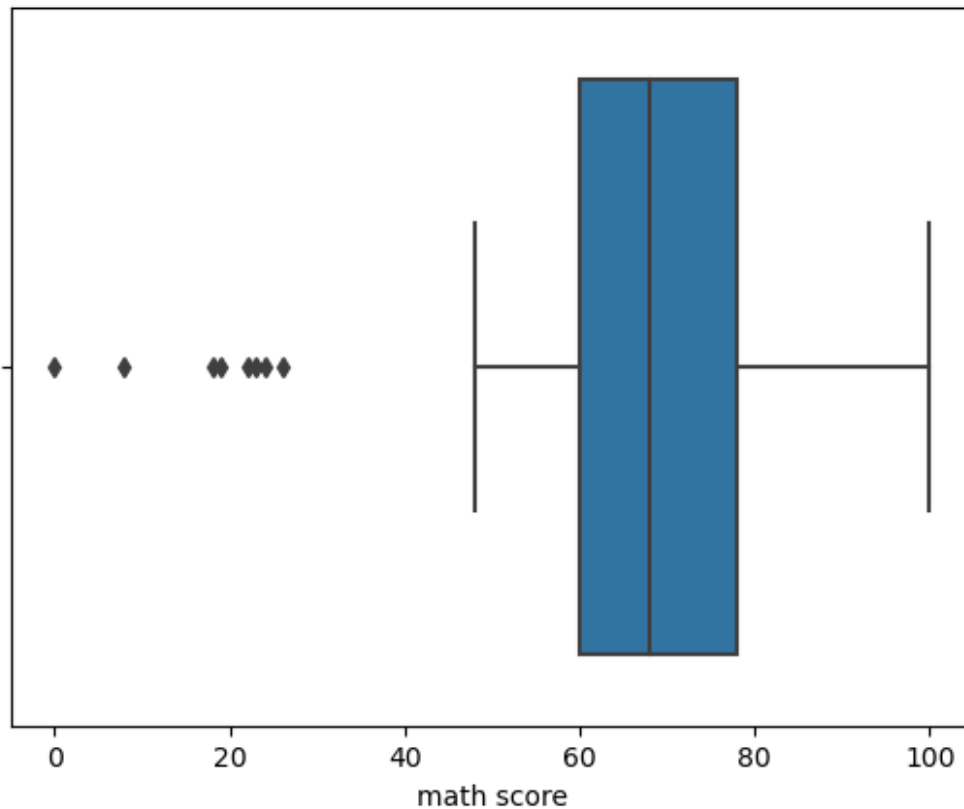
```
Name: writing score, Length: 1000, dtype: float64
```

```
a3.shape
```

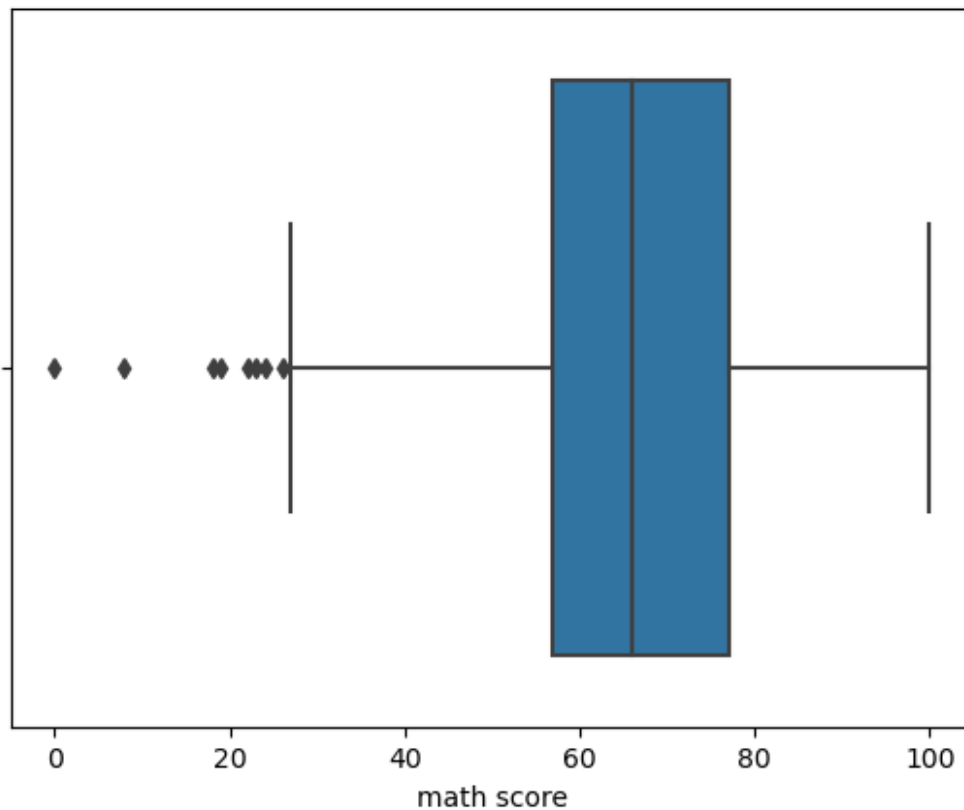
```
(996, 8)
```

```
sns.boxplot(x=a2['math score'])
```

```
<AxesSubplot:xlabel='math score'>
```



```
sns.boxplot(x=a['math score'])
<AxesSubplot:xlabel='math score'>
```



```
from sklearn.preprocessing import MinMaxScaler
```

```
scaler = MinMaxScaler()
a[['reading score']] = scaler.fit_transform(a[['reading score']])
a['reading score'] = a['reading score'].round(1)
```

```
a.head()
```

	gender	race/ethnicity	parental level of education	lunch	\
0	female	group B	bachelor's degree	standard	
1	female	group C	some college	standard	
2	female	group B	master's degree	standard	
3	male	group A	associate's degree	free/reduced	
4	male	group C	some college	standard	

	test preparation course	math score	reading score	writing score
0	none	72.0	0.7	74.0
1	completed	69.0	0.9	88.0
2	none	90.0	0.9	93.0

3	none	47.0	0.5	44.0
4	none	76.0	0.7	75.0

```
from sklearn.preprocessing import MinMaxScaler
```

```
scaler = MinMaxScaler(feature_range=(1,3))
a[['reading score']] = scaler.fit_transform(a[['reading score']])
a['reading score'] = a['reading score'].round(1)
```

```
a['reading score']
```

```
0      2.4
1      2.8
2      2.8
3      2.0
4      2.4
```

```
...
995    3.0
996    2.0
997    2.4
998    2.4
999    2.6
```

```
Name: reading score, Length: 1000, dtype: float64
```