```
import pandas as pd
import numpy as np
```

```
df=pd.read_csv('/content/StudentsPerformance.csv')
df
```

	gender	race/ethnicity	parental level of education	lunch	test preparation course	math score	reading score	writing score
0	female	group B	bachelor's degree	standard	none	72.0	72	74
1	female	group C	some college	standard	completed	69.0	90	88
2	female	group B	master's degree	standard	none	NaN	95	93
3	male	group A	associate's degree	free/reduced	none	NaN	57	44
4	male	group C	some college	standard	none	NaN	78	75
...	...	...	...	...	...	...	...	...
995	female	group E	master's degree	standard	completed	88.0	99	95

```
df.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               989 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                   967 non-null    float64
 6   reading score                1000 non-null   int64  
 7   writing score                1000 non-null   int64  
dtypes: float64(1), int64(2), object(5)
memory usage: 62.6+ KB
```

```
df.describe()
```

	math score	reading score	writing score
count	967.000000	1000.000000	1000.000000
mean	66.209928	69.169000	68.054000
std	15.082876	14.600192	15.195657
min	0.000000	17.000000	10.000000
25%	57.000000	59.000000	57.750000
50%	66.000000	70.000000	69.000000
75%	77.000000	79.000000	79.000000
max	100.000000	100.000000	100.000000

```
df.dtypes
```

```
gender                         object
race/ethnicity                 object
parental level of education    object
lunch                          object
test preparation course        object
math score                    float64
reading score                   int64
```

```
writing score                   int64
dtype: object
```

```
df.shape
```

```
(1000, 8)
```

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

```
gender                          0
race/ethnicity                 11
parental level of education     0
lunch                           0
test preparation course         0
math score                     33
reading score                   0
writing score                   0
dtype: int64
```

```
df.replace(np.nan,df['math score'].mean(),inplace=True)
```

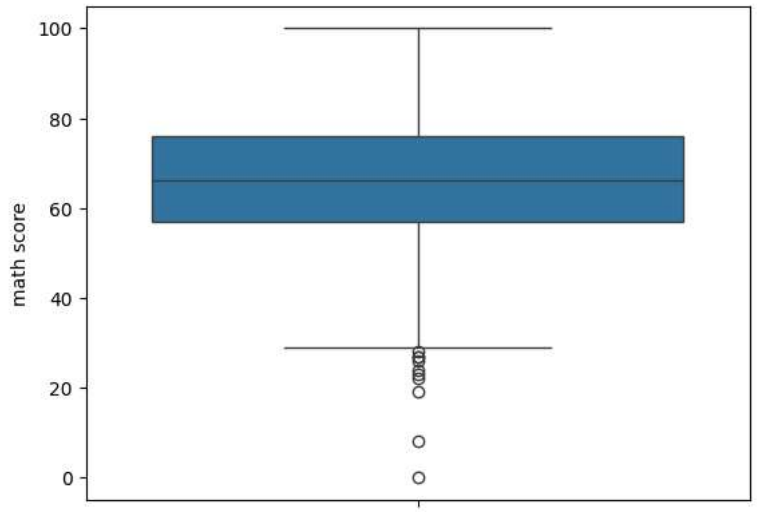
```
df.isnull().sum()
```

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

```
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

```
sns.boxplot(y=df['math score'])
```

```
<Axes: ylabel='math score'>
```



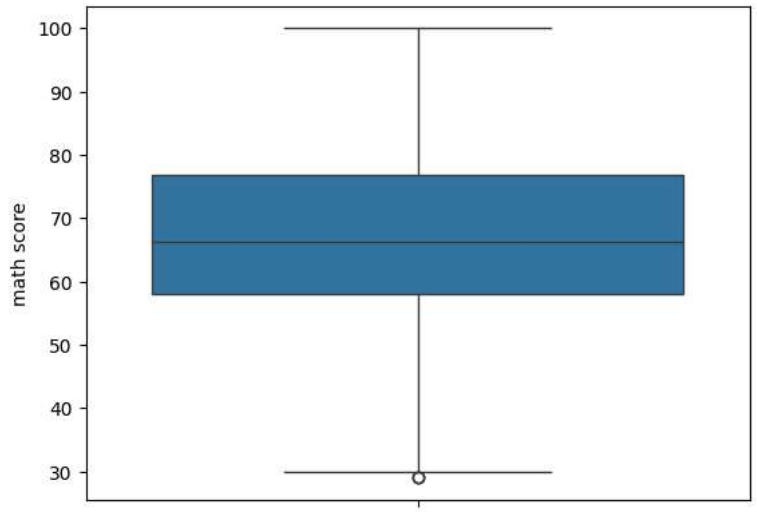

```
Q1 = df['math score'].quantile(0.25)
Q3 = df['math score'].quantile(0.75)
IQR=Q3-Q1
df=df[(df['math score']>=Q1-1.5 *IQR)& (df['math score']<=Q3+1.5*IQR)]
```

```
print(df['math score'].head())
```

```
0    72.000000
1    69.000000
2    66.209928
3    66.209928
4    66.209928
Name: math score, dtype: float64
```

```
sns.boxplot(y=df['math score'])
```

```
<Axes: ylabel='math score'>
```




```
# prompt: Apply data transformations on at least one of the variables to change the scale for better
# understanding of the variable

import numpy as np
#Apply a log transformation to the 'math-score' variable

df['math score_log']=np.log(df['math score'])

print(df['math score_log'].head())
```

```
0    4.276666
1    4.234107
2    4.192830
3    4.192830
4    4.192830
Name: math score_log, dtype: float64
<ipython-input-32-9080dd1c07f1>:7: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-cc
  df['math score_log']=np.log(df['math score'])
```

```
df
```

	gender	race/ethnicity	parental level of education	lunch	test preparation course	math score	reading score	wri s
0	female	group B	bachelor's degree	standard	none	72.000000	72	
1	female	group C	some college	standard	completed	69.000000	90	
2	female	group B	master's degree	standard	none	66.209928	95	
3	male	group A	associate's degree	free/reduced	none	66.209928	57	
4	male	group C	some college	standard	none	66.209928	78	
...	...	...	...	...	...	...	...	
995	female	group E	master's degree	standard	completed	88.000000	99	

```
df['Total Marks'] = df[['math score', 'reading score', 'writing score']].sum(axis=1)
df
```

```
<ipython-input-35-a90b576437b3>:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_
  df['Total Marks'] = df[['math score', 'reading score', 'writing score']].sum(axis=1)
```

	gender	race/ethnicity	parental level of education	lunch	test preparation course	math score	reading score	wri s
0	female	group B	bachelor's degree	standard	none	72.000000	72	
1	female	group C	some college	standard	completed	69.000000	90	
2	female	group B	master's degree	standard	none	66.209928	95	
3	male	group A	associate's degree	free/reduced	none	66.209928	57	
4	male	group C	some college	standard	none	66.209928	78	
...	...	...	...	...	...	...	...	
995	female	group E	master's degree	standard	completed	88.000000	99	

```
df['Percentage'] = (df['Total Marks'] / 3)
df
```

```
<ipython-input-40-c5a2423280d5>:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_
  df['Percentage'] = (df['Total Marks'] / 3)
```

	gender	race/ethnicity	parental level of	lunch	test preparation	math score	reading score	wri s

```
def get_grade(percentage):
    if percentage < 60:
        return 'F'
    elif 60 <= percentage < 70:
        return 'B'
    elif 70<= percentage < 80:
        return 'C'
    elif 80<= percentage < 90:
        return 'A'
    else:
        return 'O'
```

4	male	group C		standard	none	66.209928	78

```
df['Grade'] = df['Percentage'].apply(get_grade)
```

```
<ipython-input-47-ec3e4ab53c07>:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-cc
  df['Grade'] = df['Percentage'].apply(get_grade)
```

```
df
```

	gender	race/ethnicity	parental level of education	lunch	test preparation course	math score	reading score	writing score	math score_log	Total Marks	Percentage	Grade
0	female	group B	bachelor's degree	standard	none	72.000000	72	74	4.276666	218.000000	72.666667	C
1	female	group C	some college	standard	completed	69.000000	90	88	4.234107	247.000000	82.333333	A
2	female	group B	master's degree	standard	none	66.209928	95	93	4.192830	254.209928	84.736643	A
3	male	group A	associate's degree	free/reduced	none	66.209928	57	44	4.192830	167.209928	55.736643	F
4	male	group C	some college	standard	none	66.209928	78	75	4.192830	219.209928	73.069976	C
...	...	...	...	...	...	...	...	...	...	...	...	...
995	female	group E	master's degree	standard	completed	88.000000	99	95	4.477337	282.000000	94.000000	O
996	male	group C	high school	free/reduced	none	62.000000	55	55	4.127134	172.000000	57.333333	F

Start coding or generate with AI.