In [1]:

```
import pandas as pd
import numpy as np
import seaborn as sns
```

Uploading the dataset

In [3]:

```
data = pd.read_csv("student_data.csv")
data
```

Out[3]:

	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	72	74
1	female	group C	some college	standard	completed	69	90	88
2	female	group B	master's degree	standard	none	90	95	93
3	male	group A	associate's degree	free/reduced	none	47	57	44
4	male	group C	some college	standard	none	76	78	75
...	...	...	...	...	...	...	...	...
995	female	group E	master's degree	standard	completed	88	99	95
996	male	group C	high school	free/reduced	none	62	55	55
997	female	group C	high school	free/reduced	completed	59	71	65
998	female	group D	some college	standard	completed	68	78	77
999	female	group D	some college	free/reduced	none	77	86	86

1000 rows × 8 columns

1. Understanding the data

In [4]:

```
data.head()
```

Out[4]:

	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	72	74
1	female	group C	some college	standard	completed	69	90	88
2	female	group B	master's degree	standard	none	90	95	93
3	male	group A	associate's degree	free/reduced	none	47	57	44
4	male	group C	some college	standard	none	76	78	75

In [5]:
```
data.tail()
```

Out[5]:

	gender	race/ethnicity	parental level of education	lunch	test preparation course	math score	reading score	writing score
995	female	group E	master's degree	standard	completed	88	99	95
996	male	group C	high school	free/reduced	none	62	55	55
997	female	group C	high school	free/reduced	completed	59	71	65
998	female	group D	some college	standard	completed	68	78	77
999	female	group D	some college	free/reduced	none	77	86	86

In [6]:
```
data.shape
```

Out[6]:
```
(1000, 8)
```

In [7]:
```
data.describe()
```

Out[7]:

	math score	reading score	writing score
count	1000.00000	1000.000000	1000.000000
mean	66.08900	69.169000	68.054000
std	15.16308	14.600192	15.195657
min	0.00000	17.000000	10.000000
25%	57.00000	59.000000	57.750000
50%	66.00000	70.000000	69.000000
75%	77.00000	79.000000	79.000000
max	100.00000	100.000000	100.000000

In [8]:
```
data.columns
```

Out[8]:
```
Index(['gender', 'race/ethnicity', 'parental level of education', 'lunch',
       'test preparation course', 'math score', 'reading score',
       'writing score'],
      dtype='object')
```

In [9]:
```
data.nunique()
```

Out[9]:
```
gender                          2
race/ethnicity                  5
parental level of education     6
lunch                           2
test preparation course         2
math score                     81
reading score                  72
writing score                  77
dtype: int64
```

In [10]:
```
data['gender'].unique()
```

Out[10]:
```
array(['female', 'male'], dtype=object)
```

In [11]:
```
data['lunch'].unique()
```

Out[11]:
```
array(['standard', 'free/reduced'], dtype=object)
```

In [12]:
```
data['parental level of education'].unique()
```

Out[12]:
```
array(["bachelor's degree", 'some college', "master's degree",
       "associate's degree", 'high school', 'some high school'],
      dtype=object)
```

2. Cleaning the data

In [13]:
```
data.isnull()
```

Out[13]:

	gender	race/ethnicity	parental level of education	lunch	test preparation course	math score	reading score	writing score
0	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False
...	...	...	...	...	...	...	...	...
995	False	False	False	False	False	False	False	False
996	False	False	False	False	False	False	False	False
997	False	False	False	False	False	False	False	False
998	False	False	False	False	False	False	False	False
999	False	False	False	False	False	False	False	False

1000 rows × 8 columns

In [14]:
```
data.isnull().sum()
```

Out[14]:
```
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
```

In [15]:
```
student = data.drop(['race/ethnicity', 'parental level of education'], axis = 1)
```

In [16]:
```
student.head()
```

Out[16]:

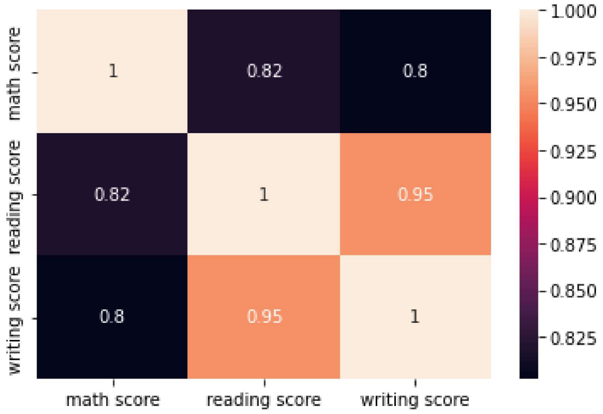
	gender	lunch	test preparation course	math score	reading score	writing score
0	female	standard	none	72	72	74
1	female	standard	completed	69	90	88
2	female	standard	none	90	95	93
3	male	free/reduced	none	47	57	44
4	male	standard	none	76	78	75

3. Relationship analysis

```
In [18]: corelation = student.corr()
```

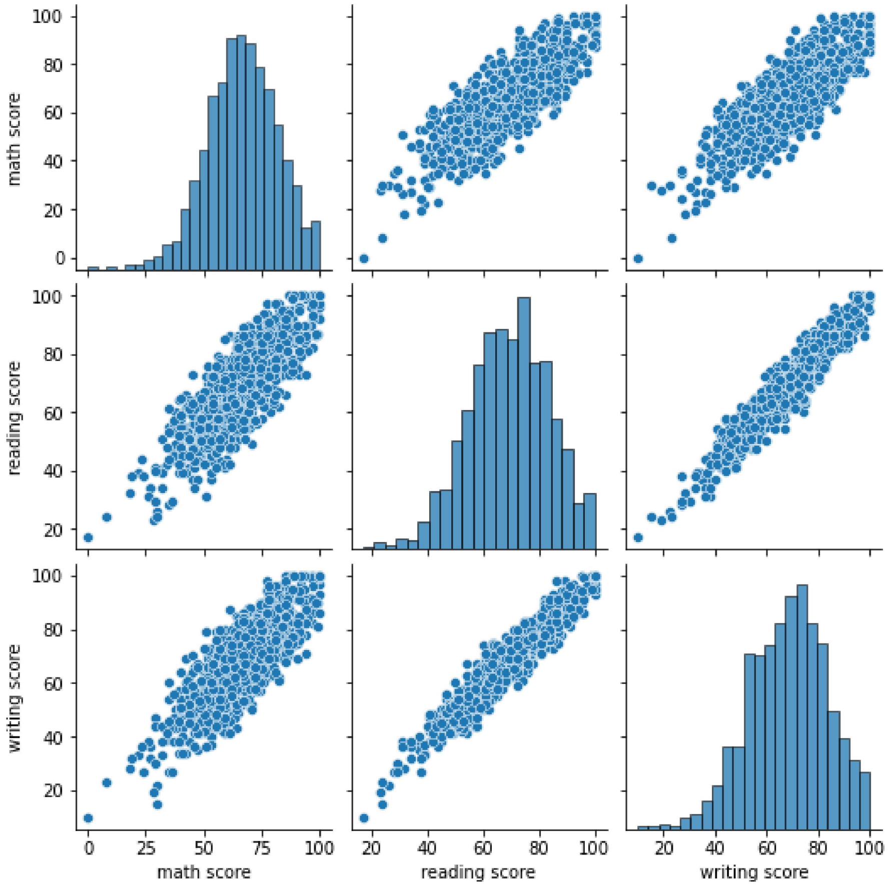
```
In [19]: sns.heatmap(corelation, xticklabels=corelation.columns, yticklabels=corelation.columns, annot=True
```

```
Out[19]: <AxesSubplot:>
```



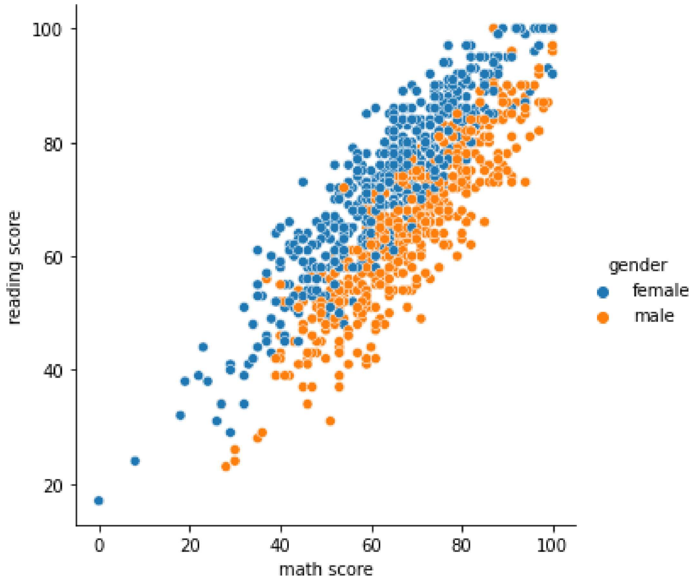
```
In [20]: sns.pairplot(student)
```

```
Out[20]: <seaborn.axisgrid.PairGrid at 0x15ea98e7130>
```



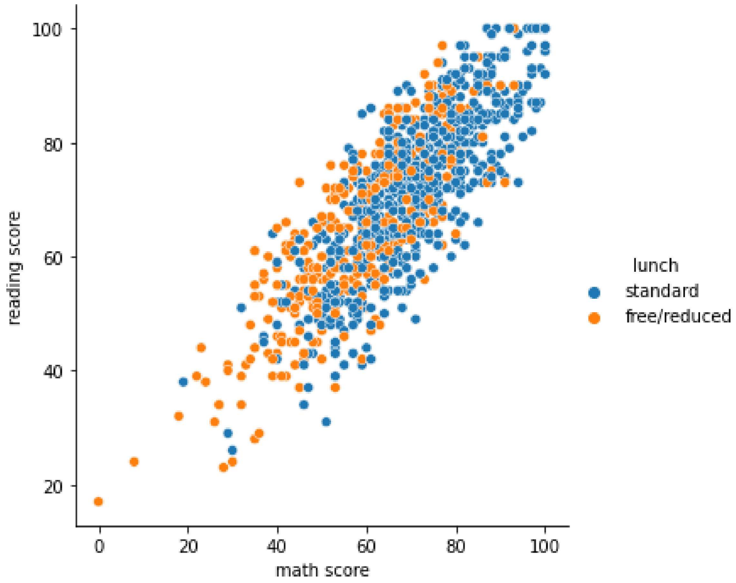
```
In [21]: sns.relplot(x = 'math score', y = 'reading score', hue = 'gender', data = student)
```

```
Out[21]: <seaborn.axisgrid.FacetGrid at 0x15eaa893850>
```



```
In [22]: sns.relplot(x = 'math score', y = 'reading score', hue = 'lunch', data = student)
```

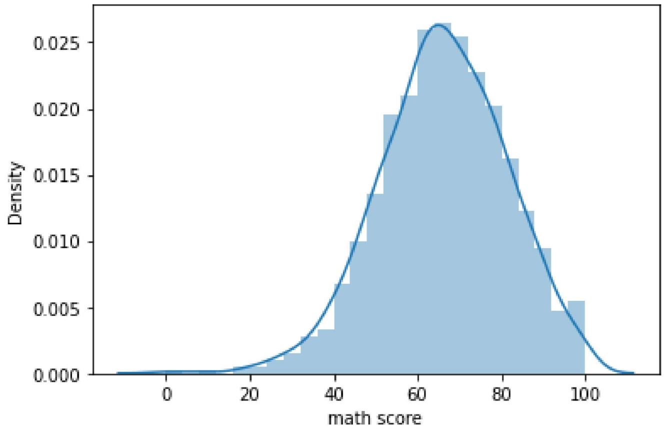
```
Out[22]: <seaborn.axisgrid.FacetGrid at 0x15eaa90d790>
```



```
In [23]: sns.distplot(student['math score'])
```

```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplo
t` is a deprecated function and will be removed in a future version. Please adapt your code to us
e either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-leve
l function for histograms).
  warnings.warn(msg, FutureWarning)
```

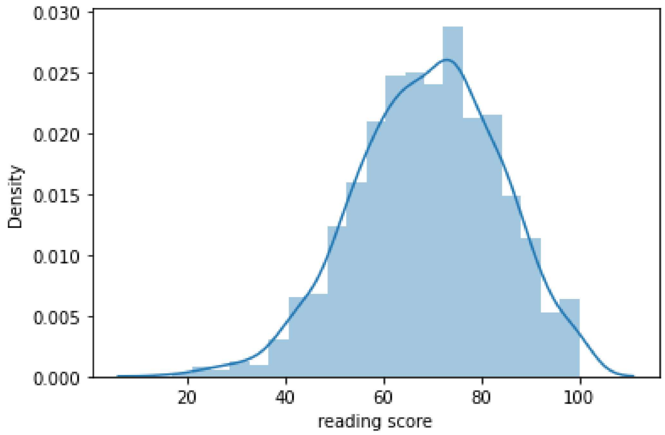
```
Out[23]: <AxesSubplot:xlabel='math score', ylabel='Density'>
```



```
In [24]: sns.distplot(student['reading score'])
```

```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplo
t` is a deprecated function and will be removed in a future version. Please adapt your code to us
e either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-leve
l function for histograms).
  warnings.warn(msg, FutureWarning)
```

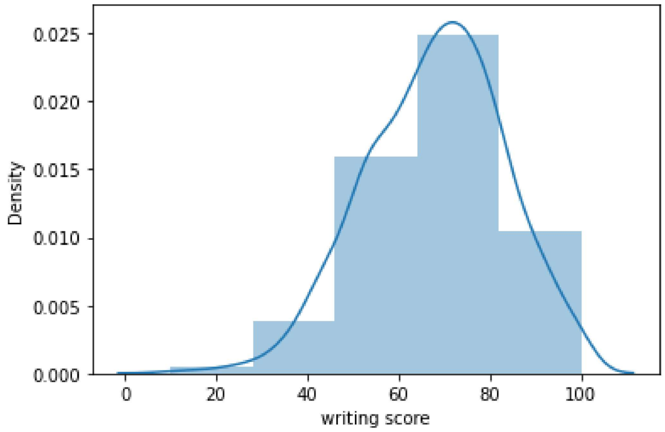
```
Out[24]: <AxesSubplot:xlabel='reading score', ylabel='Density'>
```



```
In [25]: sns.distplot(student['writing score'], bins=5)
```

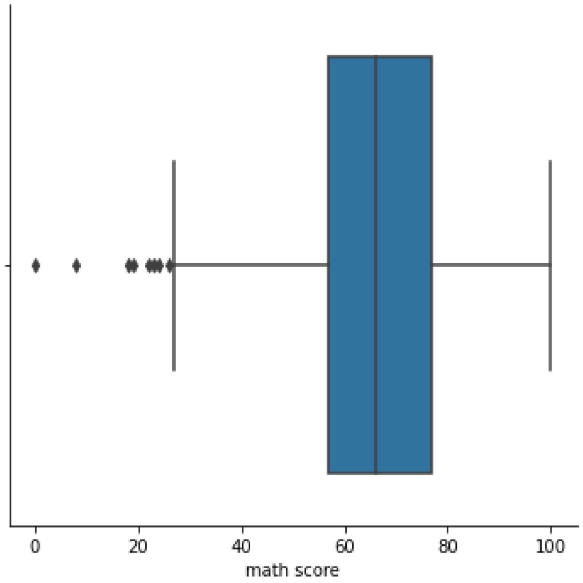
```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplo
t` is a deprecated function and will be removed in a future version. Please adapt your code to us
e either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-leve
l function for histograms).
  warnings.warn(msg, FutureWarning)
```

```
Out[25]: <AxesSubplot:xlabel='writing score', ylabel='Density'>
```



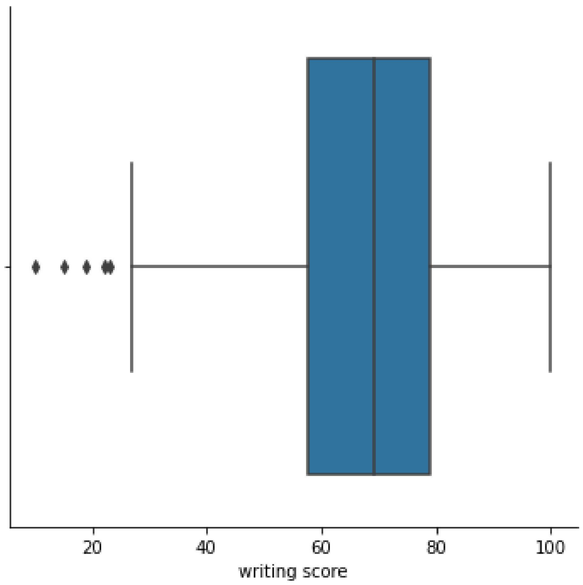
```
In [26]: sns.catplot(x = 'math score', kind= 'box', data = student)
```

```
Out[26]: <seaborn.axisgrid.FacetGrid at 0x15eaa876a30>
```



```
In [27]: sns.catplot(x = 'writing score', kind= 'box', data = student)

Out[27]: <seaborn.axisgrid.FacetGrid at 0x15eaa8769d0>
```



```
In [ ]:
```