Assignment 2 - Data Wrangling II (Data Science)

Atharva Taras (TE A - 73)

Create an “Academic performance” dataset of students and perform the following operations using Python.

Dataset - https://www.kaggle.com/code/bhartiprasad17/student-academic-performance-analysis/input (https://www.kaggle.com/code/bhartiprasad17/student-academic-performance-analysis/input)

In [1]:

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

In [2]:

```
data = pd.read_csv(r'A:\Python Projects\College Practicals\DS/StudentsPerformance
```

In [3]:

```
data.head()
```

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

In [4]:

```
data.shape
```

Out[4]:

```
(1000, 8)
```

1. Scan all variables for missing values and inconsistencies. If there are missing values and/or inconsistencies, use any of the suitable techniques to deal with them.

In [5]:

```
data.isnull().sum()
```

Out[5]:

```
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 [6]:

```
data.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   int64 
 6   reading score                1000 non-null   int64 
 7   writing score                1000 non-null   int64 
dtypes: int64(3), object(5)
memory usage: 62.6+ KB
```

No missing values/inconsistencies found

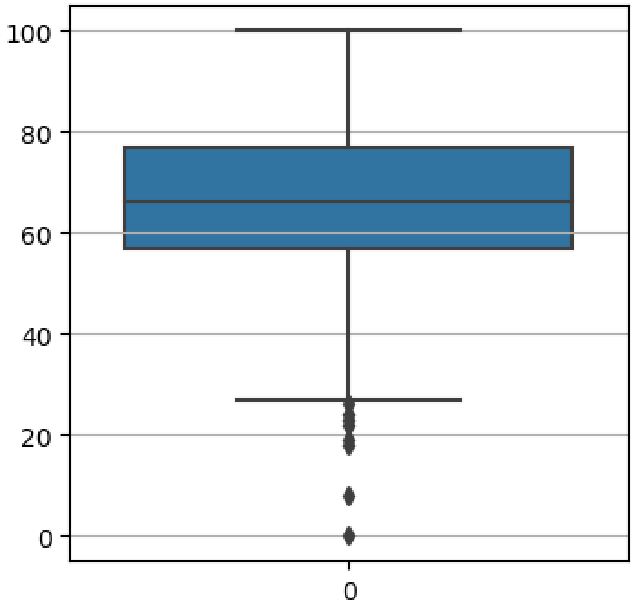
2. Scan all numeric variables for outliers. If there are outliers, use any of the suitable techniques to deal with them

In [7]:

```
plt.figure(figsize=(4, 4))
plt.grid()
sns.boxplot(data=data['math score'])
```

Out[7]:

```
<AxesSubplot: >
```

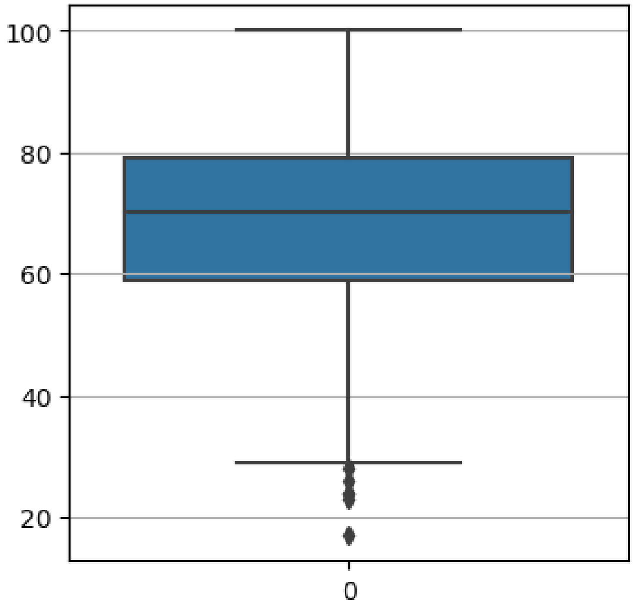


In [8]:

```
plt.figure(figsize=(4, 4))
plt.grid()
sns.boxplot(data=data['reading score'])
```

Out[8]:

```
<AxesSubplot: >
```

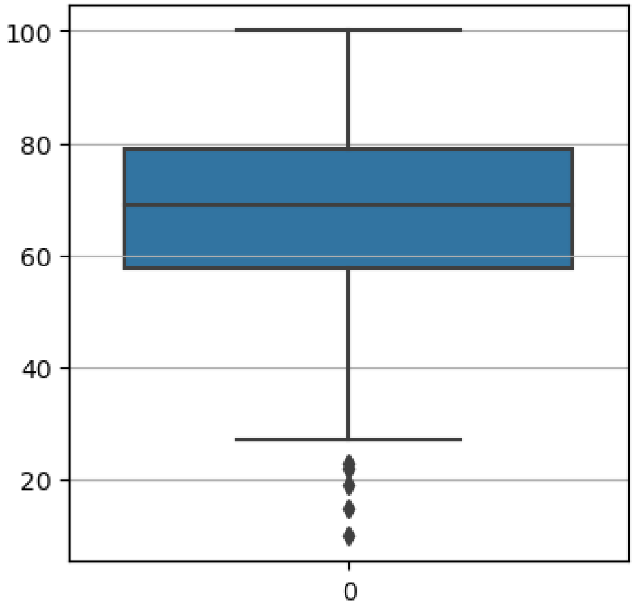


In [9]:

```
plt.figure(figsize=(4, 4))
plt.grid()
sns.boxplot(data=data['writing score'])
```

Out[9]:

```
<AxesSubplot: >
```



Since student marks can range from 0-100, outlier values cannot be dropped

3. Apply data transformations on at least one of the variables. The purpose of this transformation should be one of the following reasons: to change the scale for better understanding of the variable, to convert a non-linear relation into a linear one, or to decrease the skewness and convert the distribution into a normal distribution.

Adding a separate column for pass/fail based on aggregate score of three subjects - math, reading and writing

In [10]:

```python
marks = []

for i in range(0, data.shape[0]):
    tmp = []

    tmp.append(data['math score'][i])
    tmp.append(data['reading score'][i])
    tmp.append(data['writing score'][i])

    aggregate = (sum(tmp)/len(tmp)).round(2)

    marks.append(aggregate)
```

In [11]:

```python
data['Aggregate'] = marks
```

In [12]:

```python
data.head(2)
```

Out[12]:

	gender	race/ethnicity	parental level of education	lunch	test preparation course	math score	reading score	writing score	Aggregate
0	female	group B	bachelor's degree	standard	none	72	72	74	72.67
1	female	group C	some college	standard	completed	69	90	88	82.33

Setting aggregate pass marks = 35

In [13]:

```python
tmplist = []

for score in data['Aggregate']:

    if score > 34:
        tmplist.append(True)

    else:
        tmplist.append(False)

data['Passed'] = tmplist
```

In [14]:

```python
data.head(2)
```

Out[14]:

	gender	race/ethnicity	parental level of education	lunch	test preparation course	math score	reading score	writing score	Aggregate	Passed
0	female	group B	bachelor's degree	standard	none	72	72	74	72.67	True
1	female	group C	some college	standard	completed	69	90	88	82.33	True

In [15]:

```python
data.skew(numeric_only=True)
```

Out[15]:

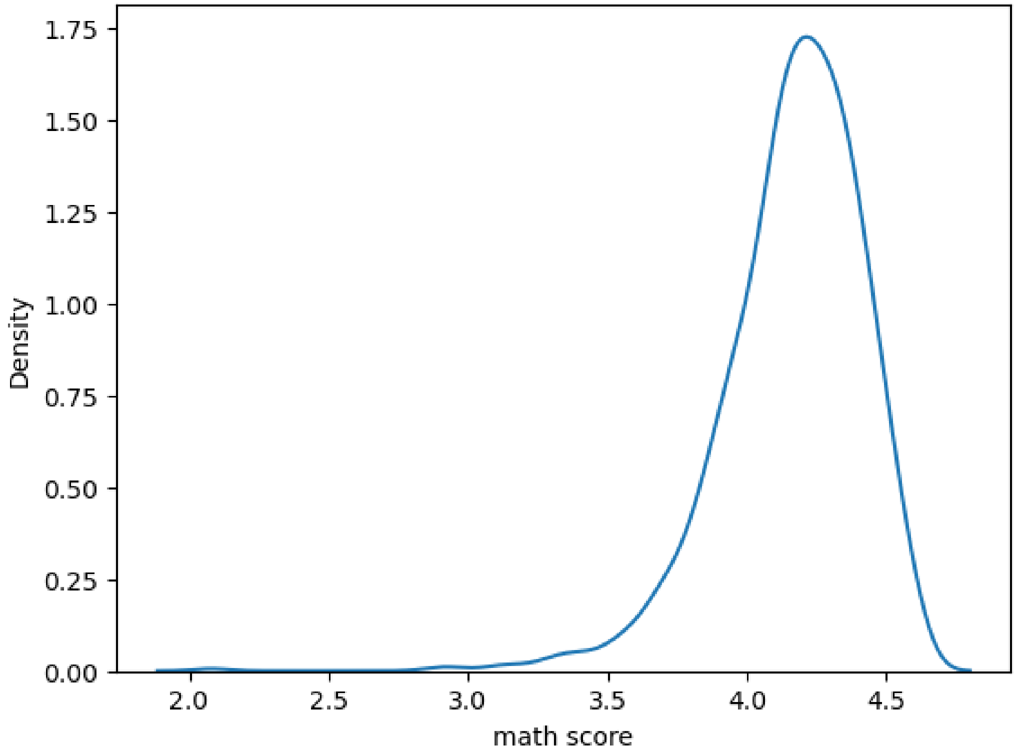
```
math score       -0.278935
reading score    -0.259105
writing score    -0.289444
Aggregate        -0.299042
Passed           -7.992087
dtype: float64
```

Converting to a normal distribution

```
In [16]:
t = np.log(data['math score'])
sns.kdeplot(t)
```

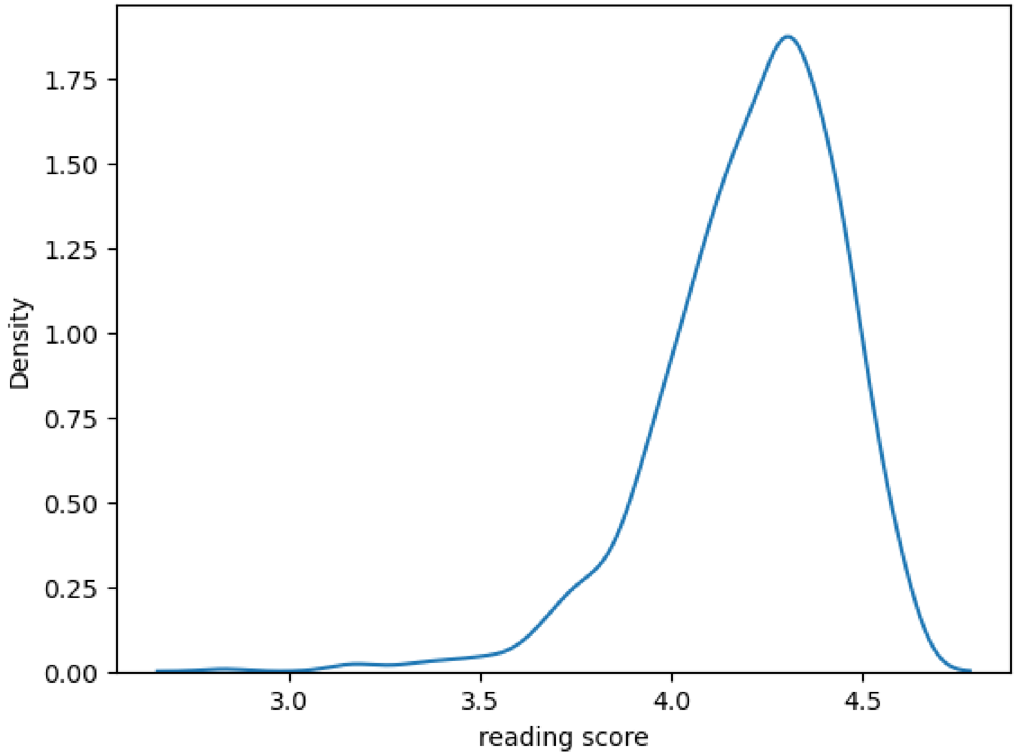
```
A:\Applications\Anaconda\lib\site-packages\pandas\core\arraylike.py:402: RuntimeWarn
ing: divide by zero encountered in log
  result = getattr(ufunc, method)(*inputs, **kwargs)
```

```
Out[16]: <AxesSubplot: xlabel='math score', ylabel='Density'>
```



```
In [17]:
t = np.log(data['reading score'])
sns.kdeplot(t)
```

```
Out[17]: <AxesSubplot: xlabel='reading score', ylabel='Density'>
```

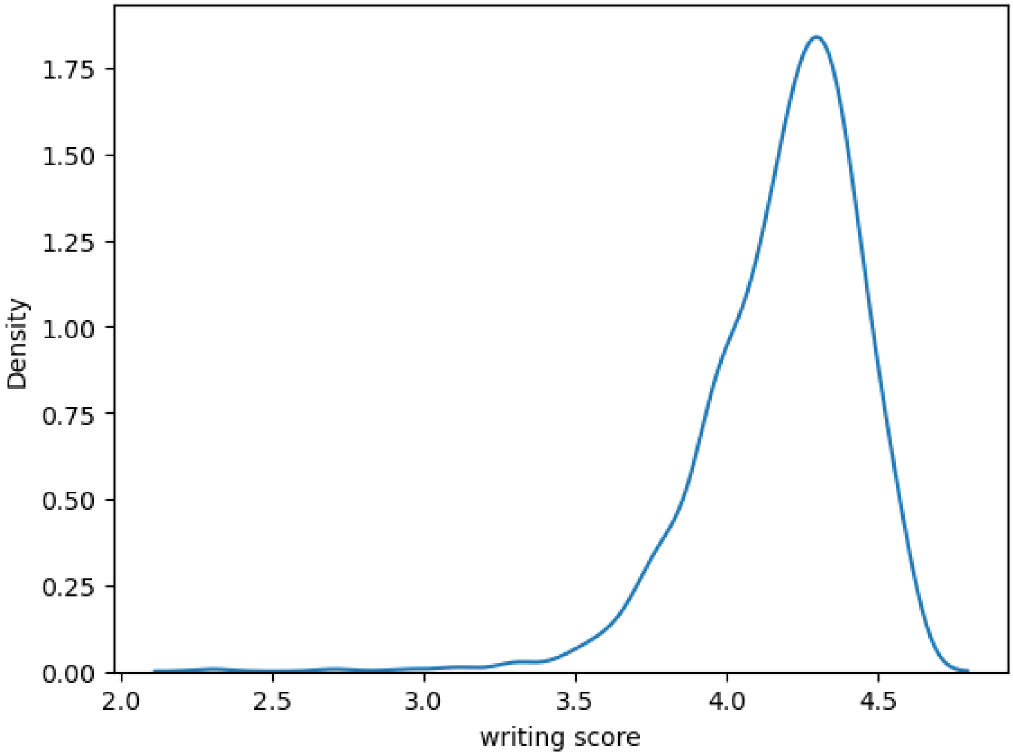


In [18]:

```
t = np.log(data['writing score'])
sns.kdeplot(t)
```

Out[18]:

```
<AxesSubplot: xlabel='writing score', ylabel='Density'>
```



In [ ]:

```

```