EDA Implementation with the help of Automated EDA Tool:-

Pandas Profiling

Sweetviz

Autoviz

In [1]:
```
# importing CSV dataset with a hepl of Pandas Library
```

In [2]:
```
import pandas as pd
```

In [3]:
```
df=pd.read_csv(r"C:\Users\Mukul\Data Kaggle\Downloads\Datasets\student.csv")
```

In [4]:
```
df.head()  # top 5 data fron dataset
```

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]:
```
df.tail()  # Last 5 data from dataset
```

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]:
```
#  Performing EDA with the help of Pandas Profiling Automated Tools:-
```

In [7]:
```
import pandas as pd
from pandas_profiling import ProfileReport
Profile = ProfileReport(df, title='Report')
Profile
```

Summarize dataset: 100% 30/30 [00:05<00:00, 5.64it/s, Completed]

Generate report structure: 100% 1/1 [00:02<00:00, 2.53s/it]

Render HTML: 100% 1/1 [00:00<00:00, 1.05it/s]

Overview

Dataset statistics

Number of variables	8
Number of observations	1000
Missing cells	0
Missing cells (%)	0.0%
Duplicate rows	0
Duplicate rows (%)	0.0%
Total size in memory	62.6 KiB
Average record size in memory	64.1 B

Variable types

Categorical	5
Numeric	3

Alerts

`math score` is highly correlated with `reading score` and 1 other fields (reading score, writing score)	High correlation
`reading score` is highly correlated with `math score` and 1 other fields (math score, writing score)	High correlation

Out[7]:

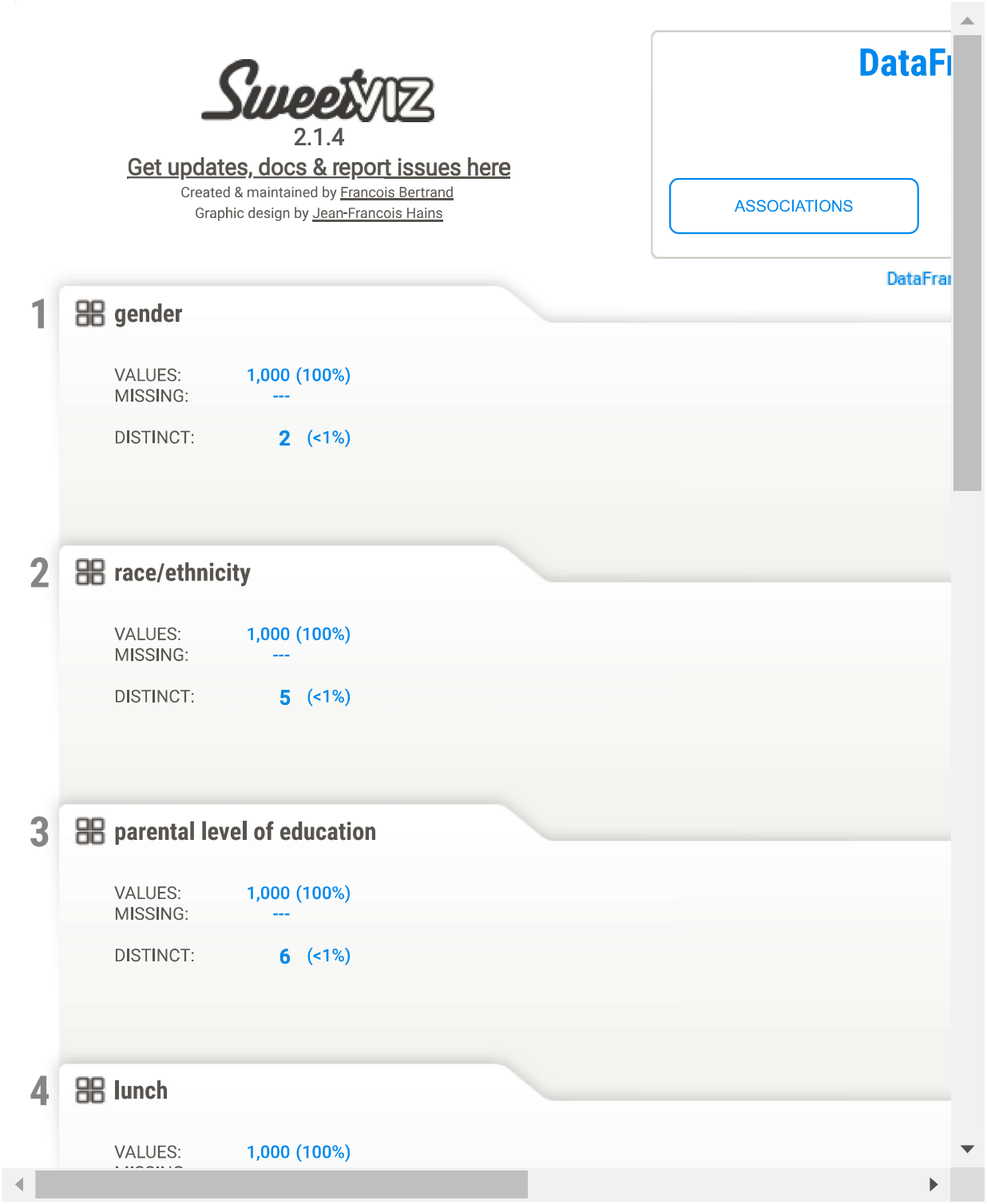
In [8]:
```
# Performing EDA with the help of SWeetviz Automated Tools:-
```

In [9]:
```
pip install sweetviz
```

```
Requirement already satisfied: sweetviz in c:\users\mukul\anaconda3\lib\site-pa
ckages (2.1.4)
Requirement already satisfied: jinja2>=2.11.1 in c:\users\mukul\anaconda3\lib\s
ite-packages (from sweetviz) (2.11.3)
Requirement already satisfied: scipy>=1.3.2 in c:\users\mukul\anaconda3\lib\sit
e-packages (from sweetviz) (1.7.3)
Requirement already satisfied: numpy>=1.16.0 in c:\users\mukul\anaconda3\lib\si
te-packages (from sweetviz) (1.21.5)
Requirement already satisfied: importlib-resources>=1.2.0 in c:\users\mukul\ana
conda3\lib\site-packages (from sweetviz) (5.9.0)
Requirement already satisfied: tqdm>=4.43.0 in c:\users\mukul\anaconda3\lib\sit
e-packages (from sweetviz) (4.64.0)
Requirement already satisfied: matplotlib>=3.1.3 in c:\users\mukul\anaconda3\li
b\site-packages (from sweetviz) (3.5.1)
Requirement already satisfied: pandas!=1.0.0,!=1.0.1,!=1.0.2,>=0.25.3 in c:\use
rs\mukul\anaconda3\lib\site-packages (from sweetviz) (1.4.2)
Requirement already satisfied: zipp>=3.1.0 in c:\users\mukul\anaconda3\lib\site
-packages (from importlib-resources>=1.2.0->sweetviz) (3.7.0)
Requirement already satisfied: MarkupSafe>=0.23 in c:\users\mukul\anaconda3\lib
\site-packages (from jinja2>=2.11.1->sweetviz) (2.0.1)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\mukul\anaconda3\li
b\site-packages (from matplotlib>=3.1.3->sweetviz) (1.3.2)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\mukul\anaconda3
\lib\site-packages (from matplotlib>=3.1.3->sweetviz) (2.8.2)
Requirement already satisfied: cycler>=0.10 in c:\users\mukul\anaconda3\lib\sit
e-packages (from matplotlib>=3.1.3->sweetviz) (0.11.0)
Requirement already satisfied: pillow>=6.2.0 in c:\users\mukul\anaconda3\lib\si
te-packages (from matplotlib>=3.1.3->sweetviz) (9.0.1)
Requirement already satisfied: packaging>=20.0 in c:\users\mukul\anaconda3\lib
\site-packages (from matplotlib>=3.1.3->sweetviz) (21.3)
Requirement already satisfied: pyparsing>=2.2.1 in c:\users\mukul\anaconda3\lib
\site-packages (from matplotlib>=3.1.3->sweetviz) (3.0.4)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\mukul\anaconda3\li
b\site-packages (from matplotlib>=3.1.3->sweetviz) (4.25.0)
Requirement already satisfied: pytz>=2020.1 in c:\users\mukul\anaconda3\lib\sit
e-packages (from pandas!=1.0.0,!=1.0.1,!=1.0.2,>=0.25.3->sweetviz) (2021.3)
Requirement already satisfied: six>=1.5 in c:\users\mukul\anaconda3\lib\site-pa
ckages (from python-dateutil>=2.7->matplotlib>=3.1.3->sweetviz) (1.16.0)
Requirement already satisfied: colorama in c:\users\mukul\anaconda3\lib\site-pa
ckages (from tqdm>=4.43.0->sweetviz) (0.4.4)
Note: you may need to restart the kernel to use updated packages.
```

In [10]:
```
import sweetviz as sv
analyze_report = sv.analyze(df)
analyze_report.show_notebook()
```

Done! Use 'show' commands to display/save. [100%] 00:00 -> (00:00 left)

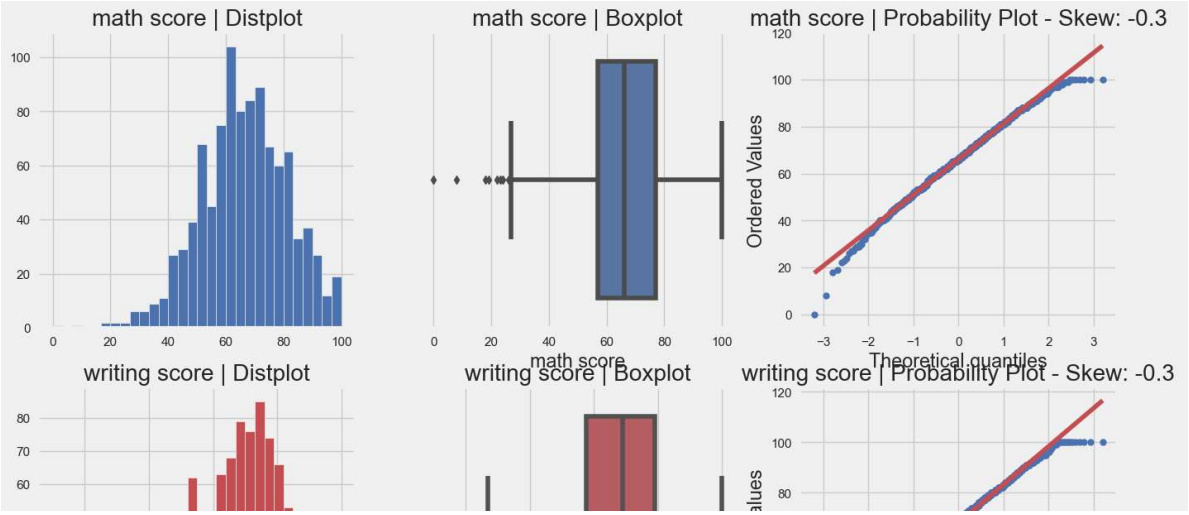


In [11]:
```
# Performing EDA with the help of AUTOVIZ EDA Tool:
```

In [12]:
```
pip install autoviz
```

```
Requirement already satisfied: autoviz in c:\users\mukul\anaconda3\lib\site-p
ackages (0.1.55)
Requirement already satisfied: pandas in c:\users\mukul\anaconda3\lib\site-pa
ckages (from autoviz) (1.4.2)
Requirement already satisfied: pyamg in c:\users\mukul\anaconda3\lib\site-pac
kages (from autoviz) (4.2.3)
Requirement already satisfied: xgboost>=0.82 in c:\users\mukul\anaconda3\lib
\site-packages (from autoviz) (1.6.2)
Requirement already satisfied: emoji in c:\users\mukul\anaconda3\lib\site-pac
kages (from autoviz) (2.1.0)
Requirement already satisfied: panel~=0.12.6 in c:\users\mukul\anaconda3\lib
\site-packages (from autoviz) (0.12.7)
Requirement already satisfied: ipython in c:\users\mukul\anaconda3\lib\site-p
ackages (from autoviz) (8.2.0)
Requirement already satisfied: holoviews>=1.14.6 in c:\users\mukul\anaconda3
\lib\site-packages (from autoviz) (1.14.8)
Requirement already satisfied: xlrd in c:\users\mukul\anaconda3\lib\site-pack
ages (from autoviz) (2.0.1)
Requirement already satisfied: fsspec>=0.8.3 in c:\users\mukul\anaconda3\lib
```

In [13]:
```
from autoviz.AutoViz_Class import AutoViz_Class
AV = AutoViz_Class()
df_av = AV.AutoViz(r'C:\Users\Mukul\Data Kaggle\Downloads\Datasets\student.csv')
```



In [ ]:

In [14]:
```
df
```

Out[14]:

	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

In [15]:
```
df.head()
```

Out[15]:

	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 [16]:
```
df.tail()
```

Out[16]:

	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 [17]:
```
df.info
```

Out[17]:
```
<bound method DataFrame.info of      gender race/ethnicity parental level of ed
ucation         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             72             74
1                 completed          69             90             88
2                      none          90             95             93
3                      none          47             57             44
4                      none          76             78             75
..                      ...         ...            ...            ...
995               completed          88             99             95
996                    none          62             55             55
997               completed          59             71             65
998               completed          68             78             77
999                    none          77             86             86

[1000 rows x 8 columns]>
```

In [18]:
```
df.shape
```

Out[18]:
```
(1000, 8)
```

In [19]:
```
df.isnull
```

Out[19]:
```
<bound method DataFrame.isnull of      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             72             74
1                 completed          69             90             88
2                      none          90             95             93
3                      none          47             57             44
4                      none          76             78             75
..                      ...         ...            ...            ...
995               completed          88             99             95
996                    none          62             55             55
997               completed          59             71             65
998               completed          68             78             77
999                    none          77             86             86

[1000 rows x 8 columns]>
```

In [20]:
```
df.isnull()
```

Out[20]:

	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 [21]:
```
df.columns
```

Out[21]:
```
Index(['gender', 'race/ethnicity', 'parental level of education', 'lunch',
       'test preparation course', 'math score', 'reading score',
       'writing score'],
      dtype='object')
```

In [22]:
```
df.dtypes
```

Out[22]:
```
gender                       object
race/ethnicity               object
parental level of education  object
lunch                        object
test preparation course      object
math score                    int64
reading score                 int64
writing score                 int64
dtype: object
```

In [23]:
```
df.dtypes=='object'
```

Out[23]:
```
gender                          True
race/ethnicity                  True
parental level of education     True
lunch                           True
test preparation course         True
math score                     False
reading score                  False
writing score                  False
dtype: bool
```

In [24]:
```
df.dtypes[df.dtypes=='object']
```

Out[24]:
```
gender                         object
race/ethnicity                 object
parental level of education    object
lunch                          object
test preparation course        object
dtype: object
```

In [25]:
```
df.dtypes=='int64'
```

Out[25]:
```
gender                         False
race/ethnicity                 False
parental level of education    False
lunch                          False
test preparation course        False
math score                      True
reading score                   True
writing score                   True
dtype: bool
```

In [26]:
```
df.dtypes[df.dtypes=='int64']
```

Out[26]:
```
math score       int64
reading score    int64
writing score    int64
dtype: object
```

In [27]:
```
df.columns
```

Out[27]:
```
Index(['gender', 'race/ethnicity', 'parental level of education', 'lunch',
       'test preparation course', 'math score', 'reading score',
       'writing score'],
      dtype='object')
```

In [28]:
```
df['gender']
```

Out[28]:
```
0      female
1      female
2      female
3        male
4        male
        ...
995    female
996      male
997    female
998    female
999    female
Name: gender, Length: 1000, dtype: object
```

In [29]:
```
df['race/ethnicity']
```

Out[29]:
```
0      group B
1      group C
2      group B
3      group A
4      group C
        ...
995    group E
996    group C
997    group C
998    group D
999    group D
Name: race/ethnicity, Length: 1000, dtype: object
```

In [30]:
```
df['parental level of education']
```

Out[30]:
```
0       bachelor's degree
1            some college
2          master's degree
3       associate's degree
4            some college
              ...
995        master's degree
996            high school
997            high school
998           some college
999           some college
Name: parental level of education, Length: 1000, dtype: object
```

In [31]:
```
df['math score']
```

Out[31]:
```
0      72
1      69
2      90
3      47
4      76
       ..
995    88
996    62
997    59
998    68
999    77
Name: math score, Length: 1000, dtype: int64
```

In [32]:
```
df.dtypes=='object'
```

Out[32]:
```
gender                         True
race/ethnicity                 True
parental level of education    True
lunch                          True
test preparation course        True
math score                    False
reading score                 False
writing score                 False
dtype: bool
```

In [33]:
```
df['gender'].dtypes
```

Out[33]:
```
dtype('O')
```

In [34]:
```
df['reading score'].dtypes=='int64'
```

Out[34]:
```
True
```

In [35]:
```
[feature for feature in df.columns]
```

Out[35]:
```
['gender',
 'race/ethnicity',
 'parental level of education',
 'lunch',
 'test preparation course',
 'math score',
 'reading score',
 'writing score']
```

In [36]:
```
cat_column=[feature for feature in df.columns if df[feature].dtype=='object']
```

In [37]:
```
cat_column
```

Out[37]:
```
['gender',
 'race/ethnicity',
 'parental level of education',
 'lunch',
 'test preparation course']
```

In [38]:
```
num_column=[feature for feature in df.columns if df[feature].dtype!='object']
```

In [39]:
```
num_column
```

Out[39]:
```
['math score', 'reading score', 'writing score']
```

In [40]:
```
df[cat_column]
```

Out[40]:

	gender	race/ethnicity	parental level of education	lunch	test preparation course
0	female	group B	bachelor's degree	standard	none
1	female	group C	some college	standard	completed
2	female	group B	master's degree	standard	none
3	male	group A	associate's degree	free/reduced	none
4	male	group C	some college	standard	none
...	...	...	...	...	...
995	female	group E	master's degree	standard	completed
996	male	group C	high school	free/reduced	none
997	female	group C	high school	free/reduced	completed
998	female	group D	some college	standard	completed
999	female	group D	some college	free/reduced	none

1000 rows × 5 columns

In [41]:
```
df[num_column]
```

Out[41]:

	math score	reading score	writing score
0	72	72	74
1	69	90	88
2	90	95	93
3	47	57	44
4	76	78	75
...	...	...	...
995	88	99	95
996	62	55	55
997	59	71	65
998	68	78	77
999	77	86	86

1000 rows × 3 columns

In [42]:
```
df.memory_usage
```

Out[42]:
```
<bound method DataFrame.memory_usage of      gender race/ethnicity parental lev
el 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             72             74
1                 completed          69             90             88
2                      none          90             95             93
3                      none          47             57             44
4                      none          76             78             75
..                      ...         ...            ...            ...
995               completed          88             99             95
996                    none          62             55             55
997               completed          59             71             65
998               completed          68             78             77
999                    none          77             86             86

[1000 rows x 8 columns]>
```

In [43]:
```
df.memory_usage()
```

Out[43]:
```
Index                          128
gender                        8000
race/ethnicity                8000
parental level of education    8000
lunch                         8000
test preparation course       8000
math score                    8000
reading score                 8000
writing score                 8000
dtype: int64
```

In [44]:
```
# check missing value
```

In [ ]:
```

```

In [45]:
```
df.isnull()   # no missing values
```

Out[45]:

	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 [46]:
```
df.isnull().sum()  # indicate no missing values
```

Out[46]:
```
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 [47]:
```
df.isnull().sum().sum()
```

Out[47]:
```
0
```

In [48]:
```
df.duplicated()
```

Out[48]:
```
0      False
1      False
2      False
3      False
4      False
       ...
995    False
996    False
997    False
998    False
999    False
Length: 1000, dtype: bool
```

In [49]:
```
df.duplicated().sum()
```

Out[49]:
```
0
```

In [50]:
```
df.nunique()
```

Out[50]:
```
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 [51]:
```
df.nunique().sum()
```

Out[51]:
```
247
```

In [52]:
```
df.max()
```

Out[52]:
```
gender                                male
race/ethnicity                     group E
parental level of education    some high school
lunch                              standard
test preparation course                none
math score                              100
reading score                           100
writing score                           100
dtype: object
```

In [53]:
```
df.max()
```

Out[53]:
```
gender                              male
race/ethnicity                   group E
parental level of education    some high school
lunch                           standard
test preparation course             none
math score                           100
reading score                        100
writing score                        100
dtype: object
```

In [54]:
```
df.min()
```

Out[54]:
```
gender                              female
race/ethnicity                    group A
parental level of education    associate's degree
lunch                            free/reduced
test preparation course             completed
math score                                  0
reading score                              17
writing score                              10
dtype: object
```

In [55]:
```
df[['gender','math score']].max()
```

Out[55]:
```
gender        male
math score     100
dtype: object
```

In [56]:
```
df[['gender','math score']].min()
```

Out[56]:
```
gender        female
math score         0
dtype: object
```

In [57]:
```
# Statistical
```

In [58]:
```
df.describe()
```

Out[58]:

	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 [59]:
```
df.describe().T
```

Out[59]:

	count	mean	std	min	25%	50%	75%	max
math score	1000.0	66.089	15.163080	0.0	57.00	66.0	77.0	100.0
reading score	1000.0	69.169	14.600192	17.0	59.00	70.0	79.0	100.0
writing score	1000.0	68.054	15.195657	10.0	57.75	69.0	79.0	100.0

In [60]:
```
df.mean()
```

Out[60]:
```
math score       66.089
reading score    69.169
writing score    68.054
dtype: float64
```

In [61]:
```
df.median()
```

Out[61]:
```
math score       66.0
reading score    70.0
writing score    69.0
dtype: float64
```

In [62]:
```
df.mode()
```

Out[62]:

	gender	race/ethnicity	parental level of education	lunch	test preparation course	math score	reading score	writing score
0	female	group C	some college	standard	none	65	72	74

In [63]:
```
df.corr()
```

Out[63]:

	math score	reading score	writing score
math score	1.000000	0.817580	0.802642
reading score	0.817580	1.000000	0.954598
writing score	0.802642	0.954598	1.000000

In [64]:
```
df.cov()
```

Out[64]:

	math score	reading score	writing score
math score	229.918998	180.998958	184.939133
reading score	180.998958	213.165605	211.786661
writing score	184.939133	211.786661	230.907992

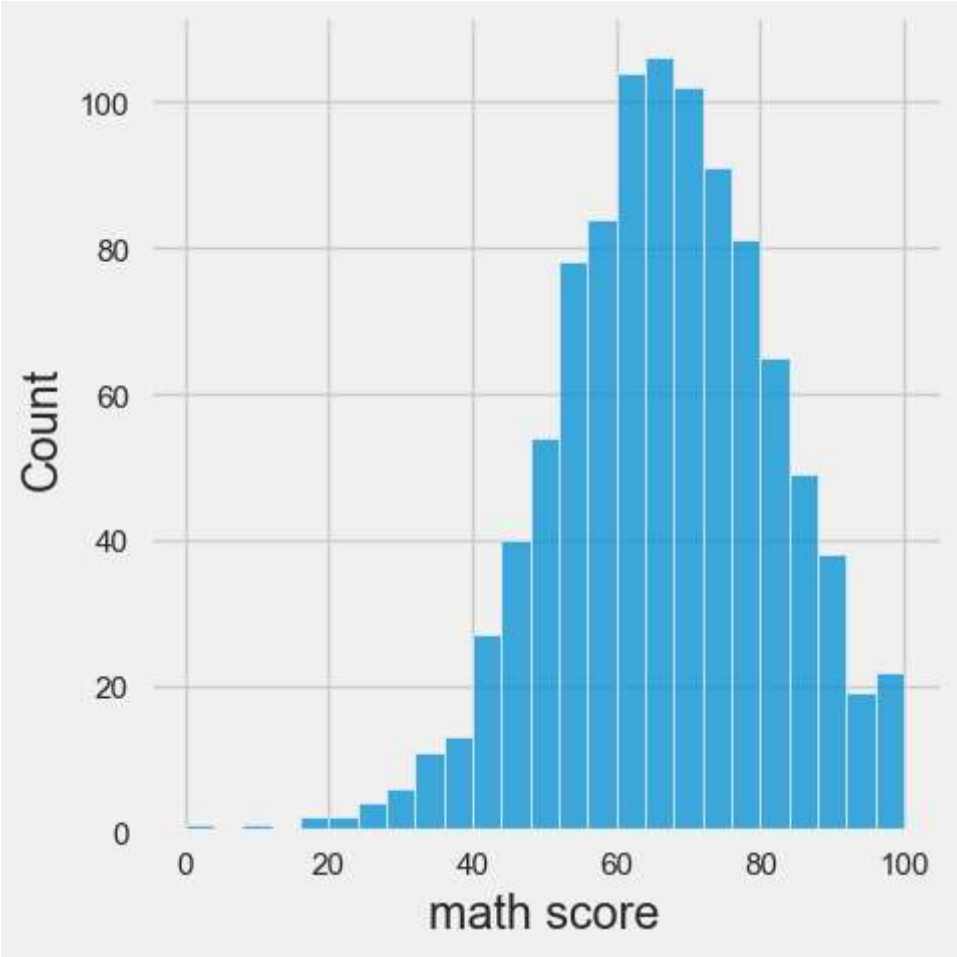
In [65]:
```
df.skew()
```

Out[65]:
```
math score      -0.278935
reading score   -0.259105
writing score   -0.289444
dtype: float64
```

In [66]:
```
import seaborn as sns
```

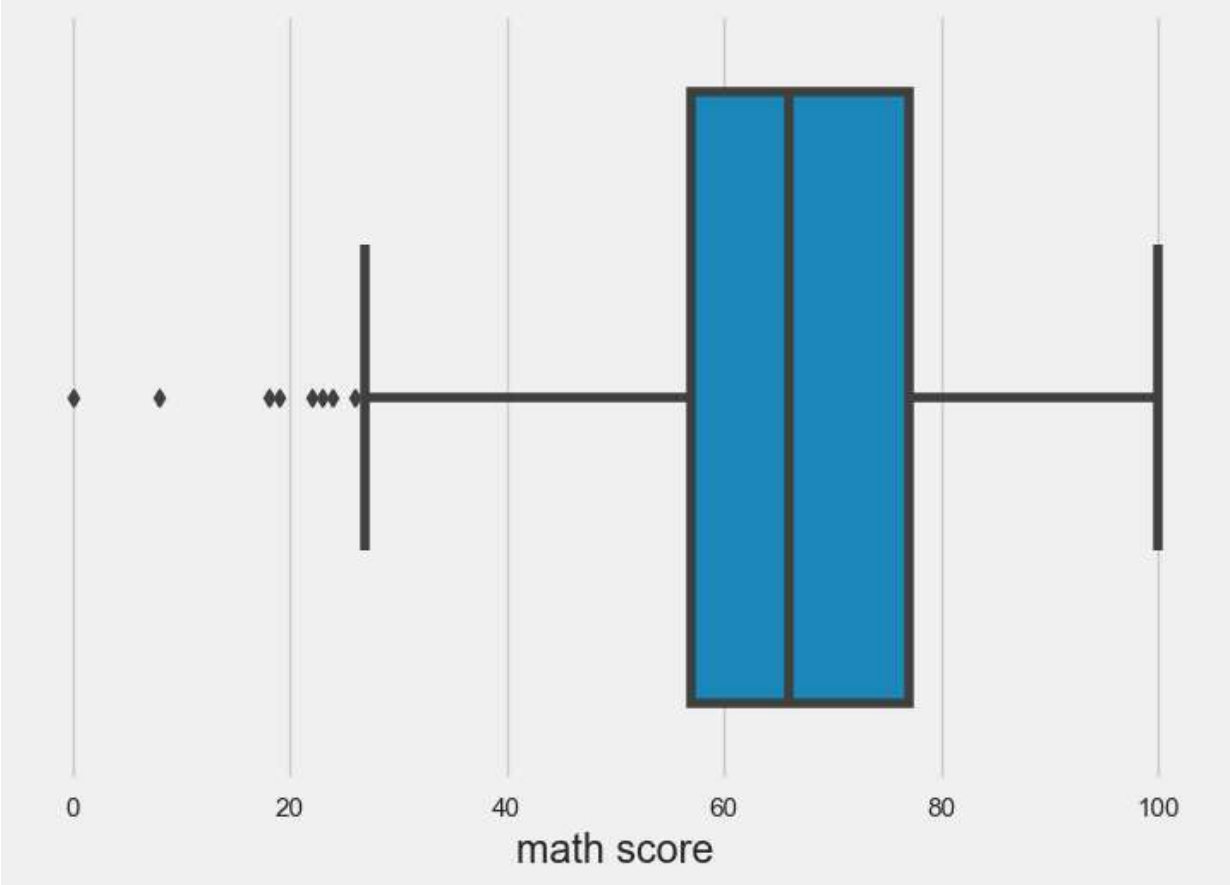
In [67]:
```
sns.displot(df['math score'])  # histogram
```

Out[67]:
```
<seaborn.axisgrid.FacetGrid at 0x23ae1061730>
```



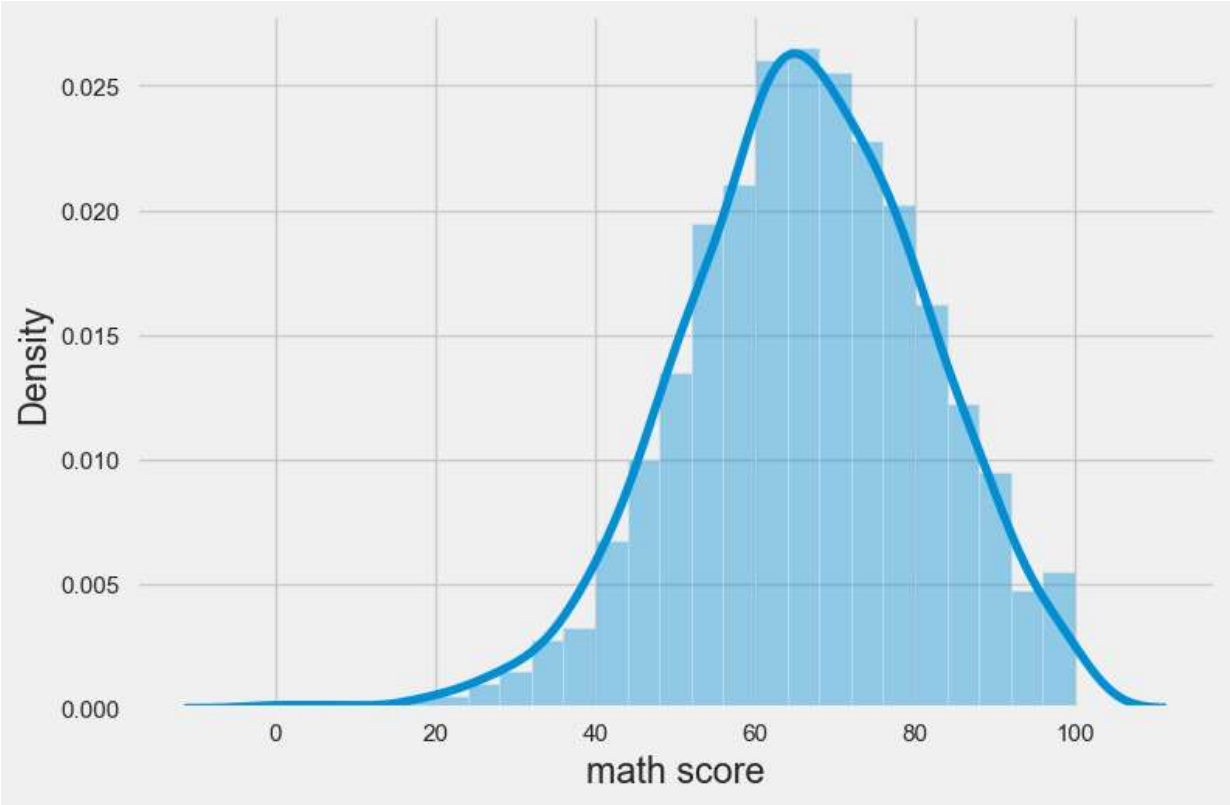
In [68]:
```
sns.boxplot(df['math score'])  # show outliear
```

Out[68]: <AxesSubplot:xlabel='math score'>



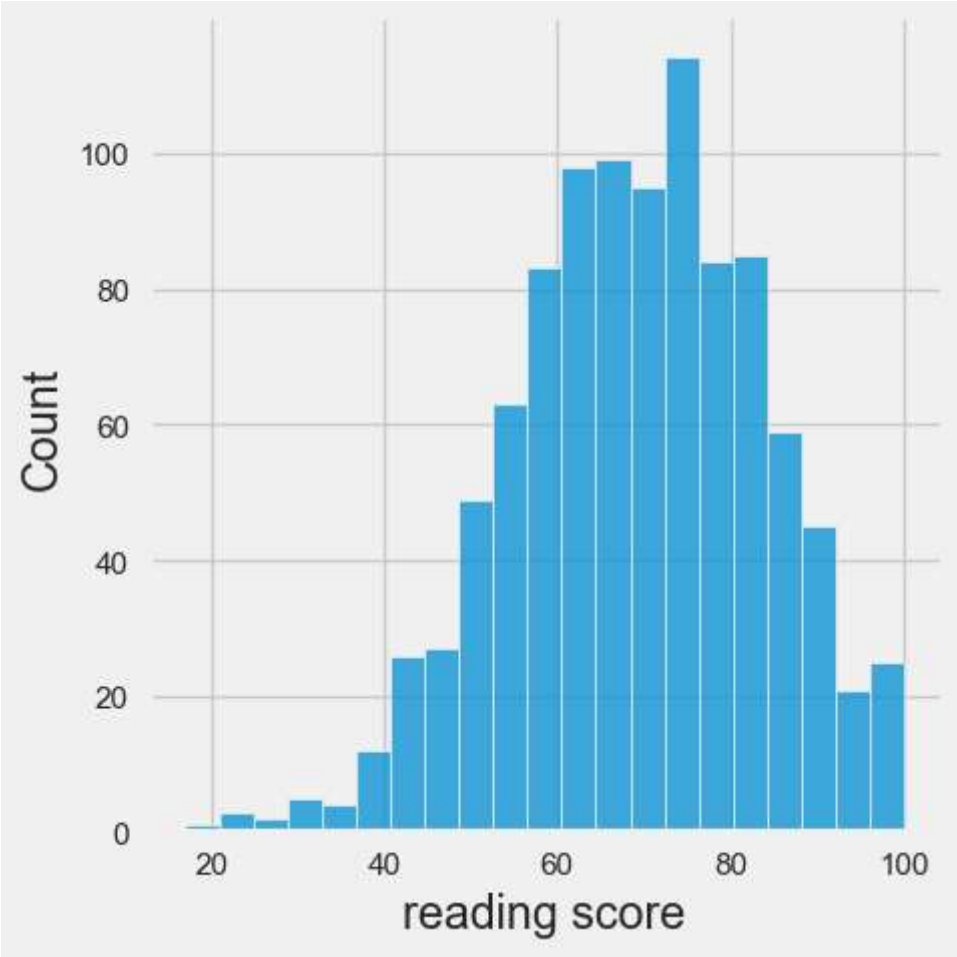
In [69]:
```
sns.distplot(df['math score'])
```

Out[69]:
```
<AxesSubplot:xlabel='math score', ylabel='Density'>
```



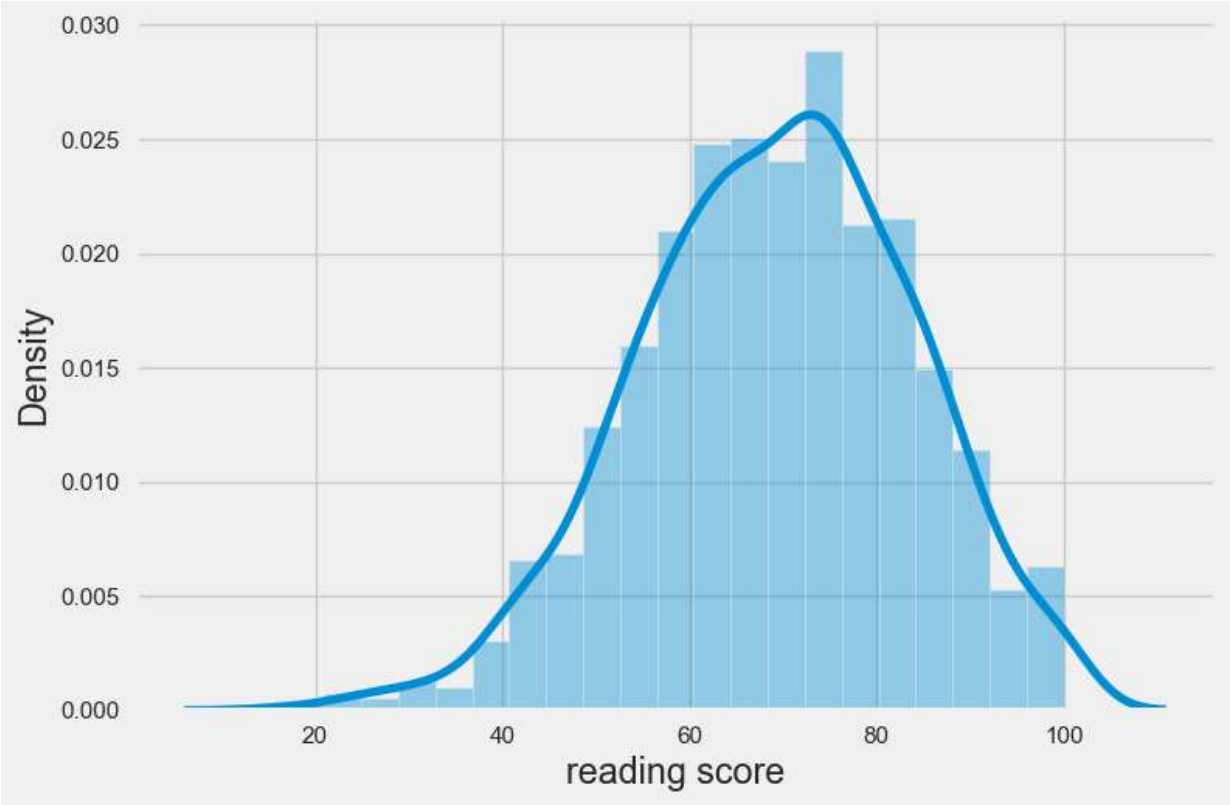
In [70]:
```
sns.displot(df['reading score'])
```

Out[70]:
```
<seaborn.axisgrid.FacetGrid at 0x23ae158ad90>
```



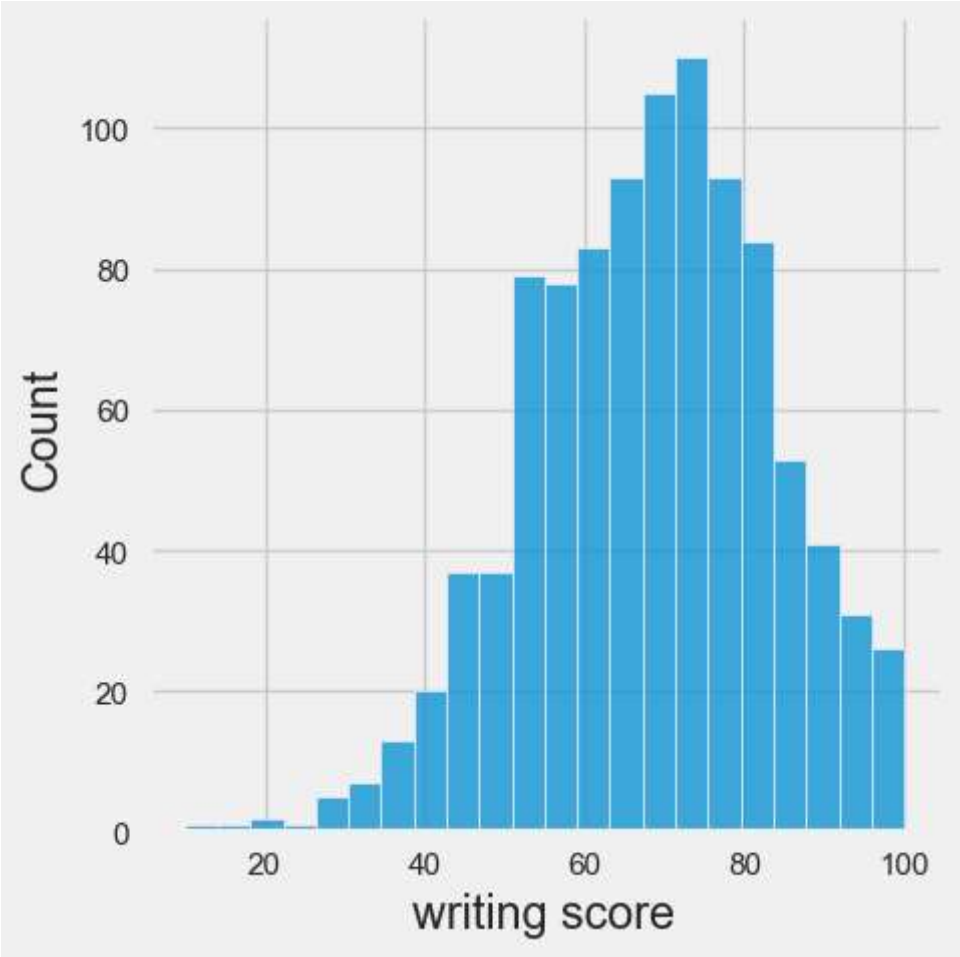
In [71]:
```
sns.distplot(df['reading score'])
```

Out[71]:
```
<AxesSubplot:xlabel='reading score', ylabel='Density'>
```



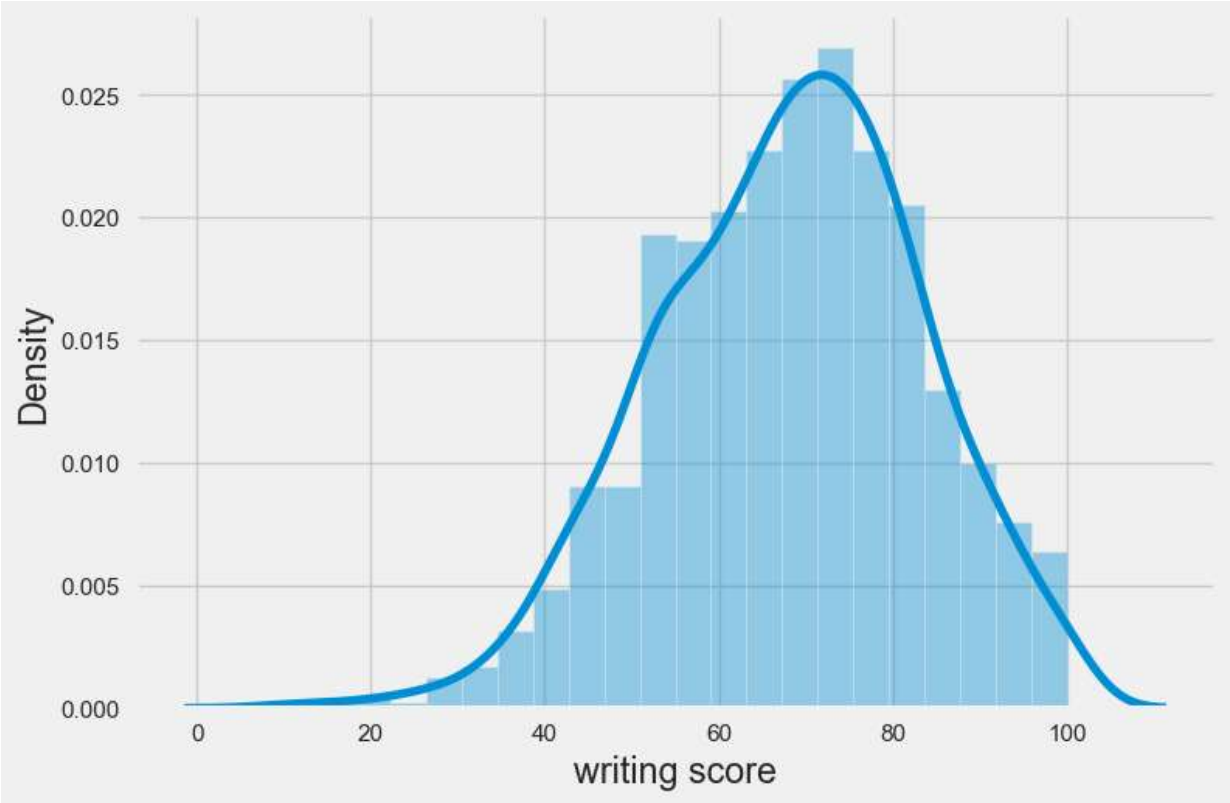
In [72]:
```
sns.displot(df['writing score'])
```

Out[72]:
```
<seaborn.axisgrid.FacetGrid at 0x23ae262b6d0>
```



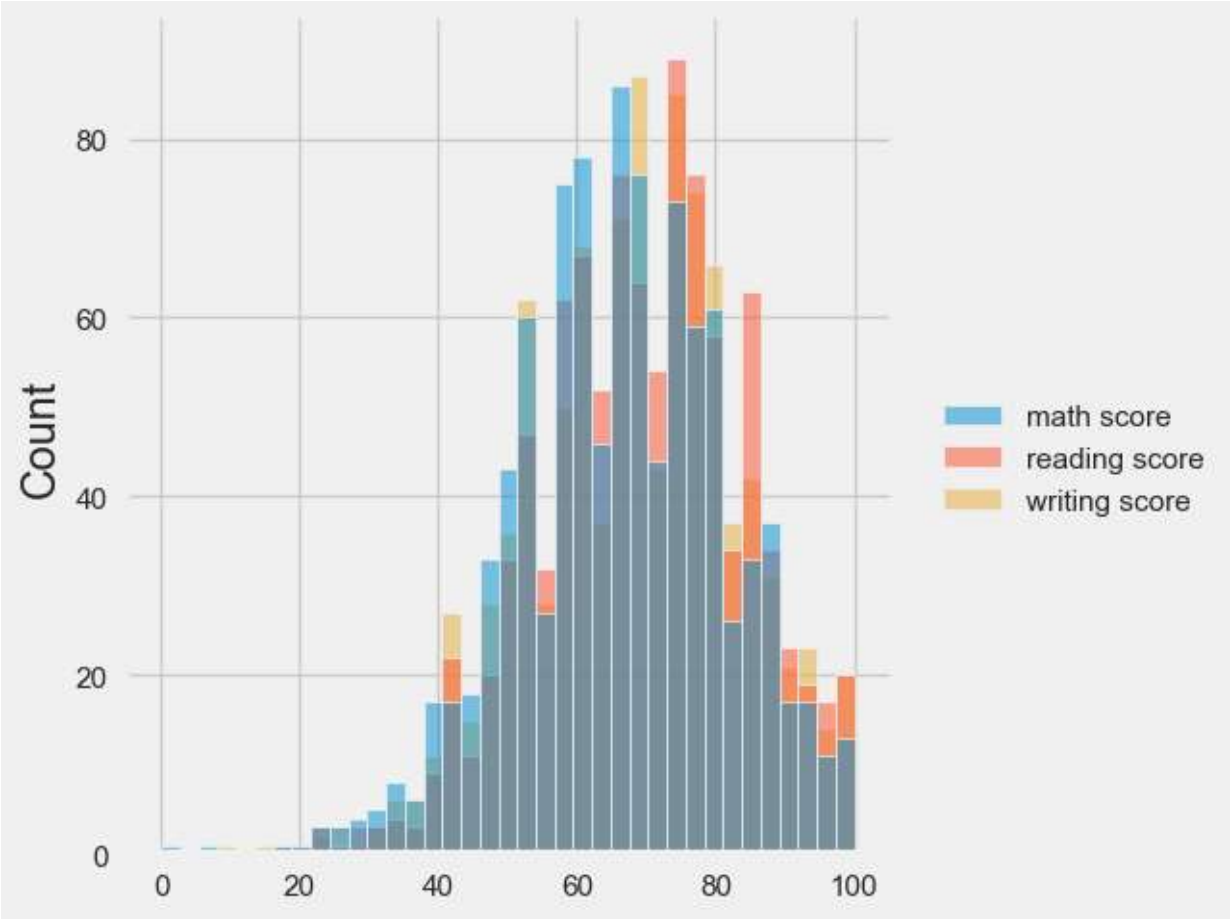
In [73]:
```
sns.distplot(df['writing score'])
```

Out[73]:
```
<AxesSubplot:xlabel='writing score', ylabel='Density'>
```



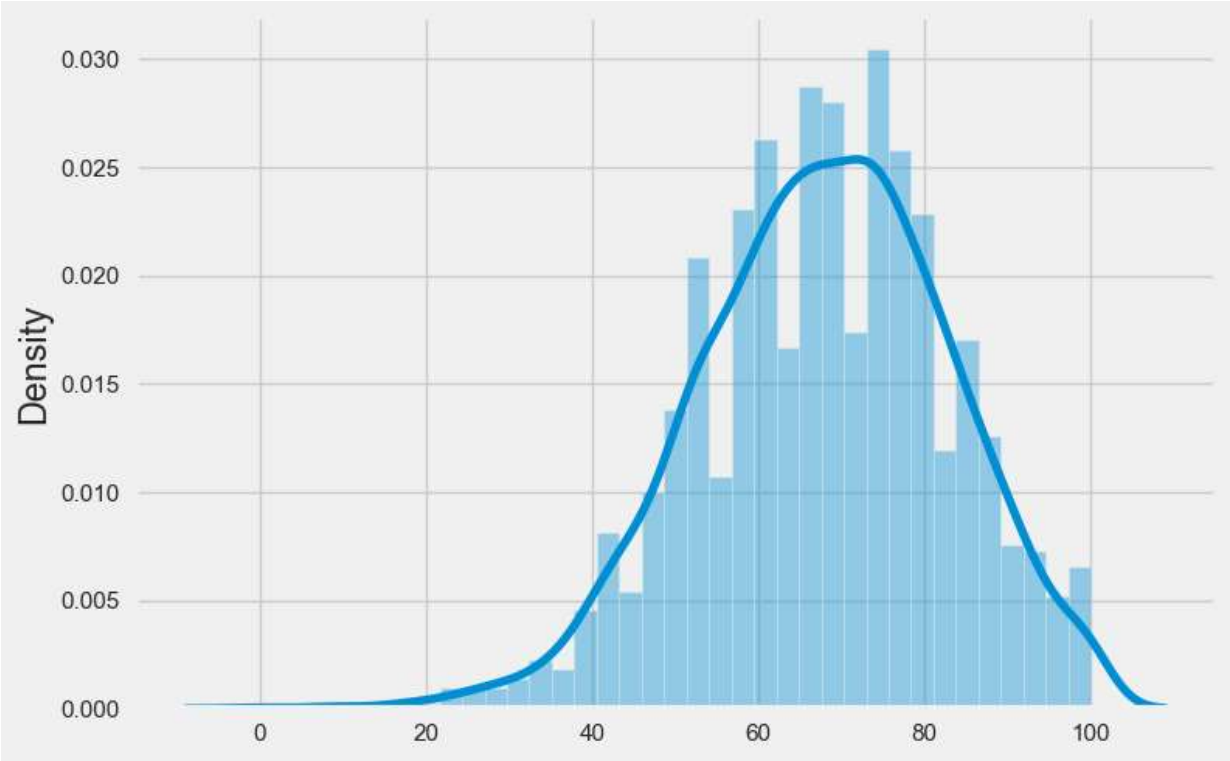
In [74]:
```
sns.displot(df[['math score' ,'reading score' , 'writing score']])
```

Out[74]:
```
<seaborn.axisgrid.FacetGrid at 0x23ae15d4340>
```



In [75]:
```
sns.distplot(df[['math score' , 'reading score' , 'writing score']])
```

Out[75]: <AxesSubplot:ylabel='Density'>



In [76]:
```
df['Average']=(df['math score'] + df['reading score']+df['writing score'])/3
```

In [77]:
```
df['Average']
```

Out[77]:
```
0      72.666667
1      82.333333
2      92.666667
3      49.333333
4      76.333333
         ...
995    94.000000
996    57.333333
997    65.000000
998    74.333333
999    83.000000
Name: Average, Length: 1000, dtype: float64
```

In [78]:
```
df.head()
```

Out[78]:

	gender	race/ethnicity	parental level of education	lunch	test preparation course	math score	reading score	writing score	Average
0	female	group B	bachelor's degree	standard	none	72	72	74	72.666667
1	female	group C	some college	standard	completed	69	90	88	82.333333
2	female	group B	master's degree	standard	none	90	95	93	92.666667
3	male	group A	associate's degree	free/reduced	none	47	57	44	49.333333
4	male	group C	some college	standard	none	76	78	75	76.333333

In [79]:
```
df.shape
```

Out[79]:
```
(1000, 9)
```

In [80]:
```
df.groupby('gender')
```

Out[80]:
```
<pandas.core.groupby.generic.DataFrameGroupBy object at 0x0000023AE2A7EDF0>
```

In [81]:
```
df.groupby('gender').count()
```

Out[81]:

gender	race/ethnicity	parental level of education	lunch	test preparation course	math score	reading score	writing score	Average
female	518	518	518	518	518	518	518	518
male	482	482	482	482	482	482	482	482

In [82]:
```
df.groupby('math score').count()
```

Out[82]:

math score	gender	race/ethnicity	parental level of education	lunch	test preparation course	reading score	writing score	Average
0	1	1	1	1	1	1	1	1
8	1	1	1	1	1	1	1	1
18	1	1	1	1	1	1	1	1
19	1	1	1	1	1	1	1	1
22	1	1	1	1	1	1	1	1
...	...	...	...	...	...	...	...	...
96	3	3	3	3	3	3	3	3
97	6	6	6	6	6	6	6	6
98	3	3	3	3	3	3	3	3
99	3	3	3	3	3	3	3	3
100	7	7	7	7	7	7	7	7

81 rows × 8 columns

In [83]:
```
df.groupby('reading score').count()
```

Out[83]:

reading score	gender	race/ethnicity	parental level of education	lunch	test preparation course	math score	writing score	Average
17	1	1	1	1	1	1	1	1
23	1	1	1	1	1	1	1	1
24	2	2	2	2	2	2	2	2
26	1	1	1	1	1	1	1	1
28	1	1	1	1	1	1	1	1
...	...	...	...	...	...	...	...	...
95	8	8	8	8	8	8	8	8
96	4	4	4	4	4	4	4	4
97	5	5	5	5	5	5	5	5
99	3	3	3	3	3	3	3	3
100	17	17	17	17	17	17	17	17

72 rows × 8 columns

In [84]:
```
df.groupby('writing score').sum()
```

Out[84]:

writing score	math score	reading score	Average
10	0	17	9.000000
15	30	24	23.000000
19	28	23	23.333333
22	30	26	26.000000
23	8	24	18.333333
...	...	...	...
96	338	382	368.000000
97	192	192	192.666667
98	162	175	177.666667
99	386	393	391.666667
100	1312	1393	1368.333333

77 rows × 3 columns

In [85]:
```
df[df['math score']<30]
```

Out[85]:

	gender	race/ethnicity	parental level of education	lunch	test preparation course	math score	reading score	writing score	Average
17	female	group B	some high school	free/reduced	none	18	32	28	26.000000
59	female	group C	some high school	free/reduced	none	0	17	10	9.000000
91	male	group C	high school	free/reduced	none	27	34	36	32.333333
145	female	group C	some college	free/reduced	none	22	39	33	31.333333
327	male	group A	some college	free/reduced	none	28	23	19	23.333333
338	female	group B	some high school	free/reduced	none	24	38	27	29.666667
363	female	group D	some high school	free/reduced	none	27	34	32	31.000000
466	female	group D	associate's degree	free/reduced	none	26	31	38	31.666667
528	female	group D	bachelor's degree	free/reduced	none	29	41	47	39.000000
601	female	group C	high school	standard	none	29	29	30	29.333333
683	female	group C	some high school	free/reduced	completed	29	40	44	37.666667
787	female	group B	some college	standard	none	19	38	32	29.666667
842	female	group B	high school	free/reduced	completed	23	44	36	34.333333
980	female	group B	high school	free/reduced	none	8	24	23	18.333333

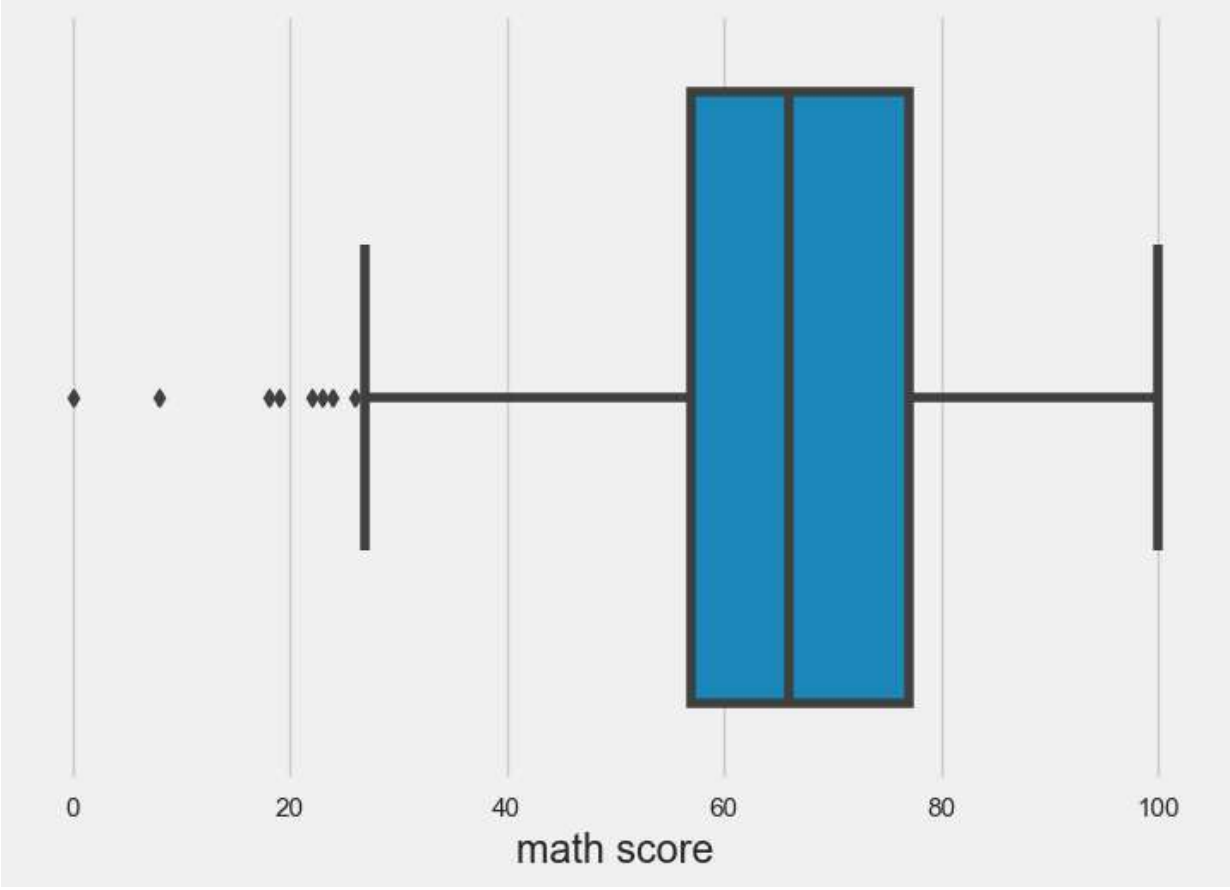
In [86]:
```
df[df['math score']<30].count()
```

Out[86]:
```
gender                         14
race/ethnicity                 14
parental level of education    14
lunch                          14
test preparation course        14
math score                     14
reading score                  14
writing score                  14
Average                        14
dtype: int64
```

In [87]:
```
#  OUTLIEAR
```

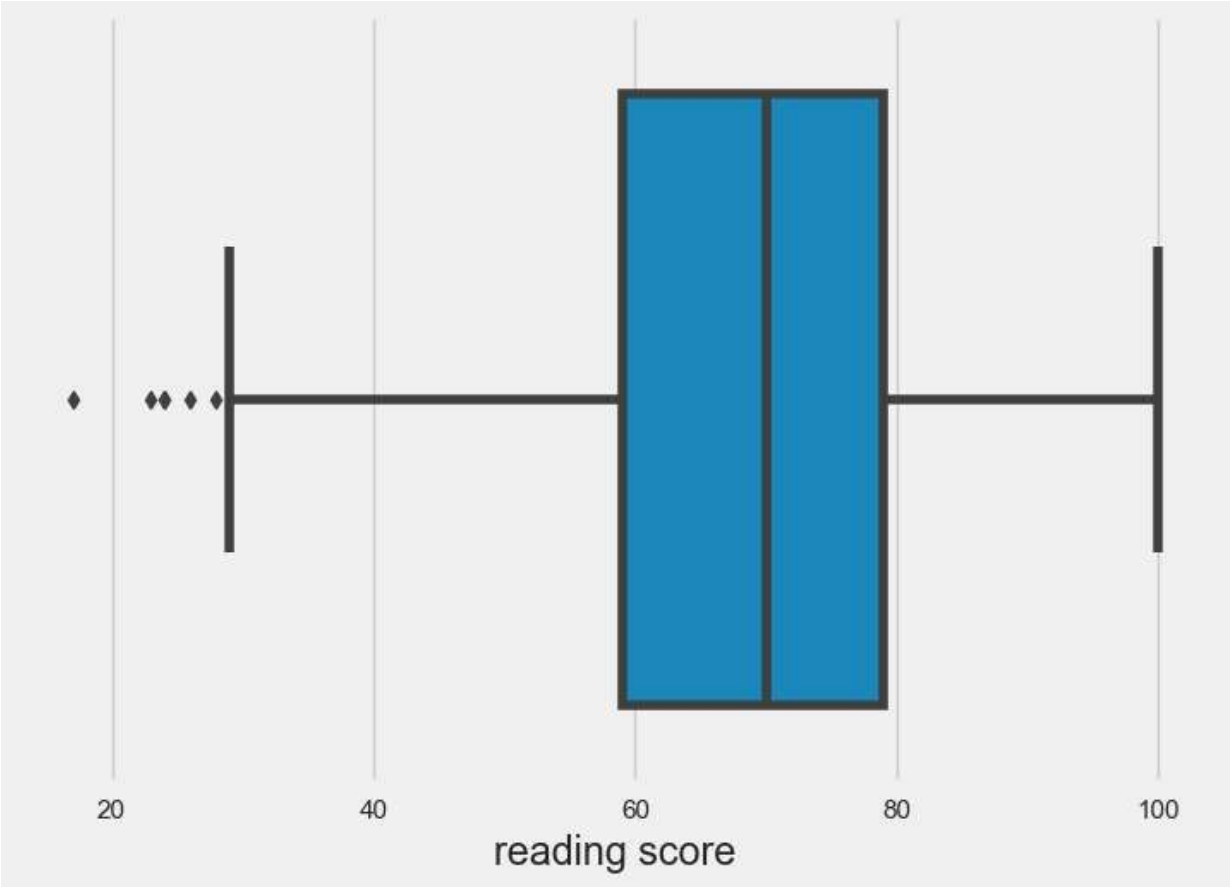
In [88]:
```
sns.boxplot(df['math score'])
```

Out[88]:
```
<AxesSubplot:xlabel='math score'>
```



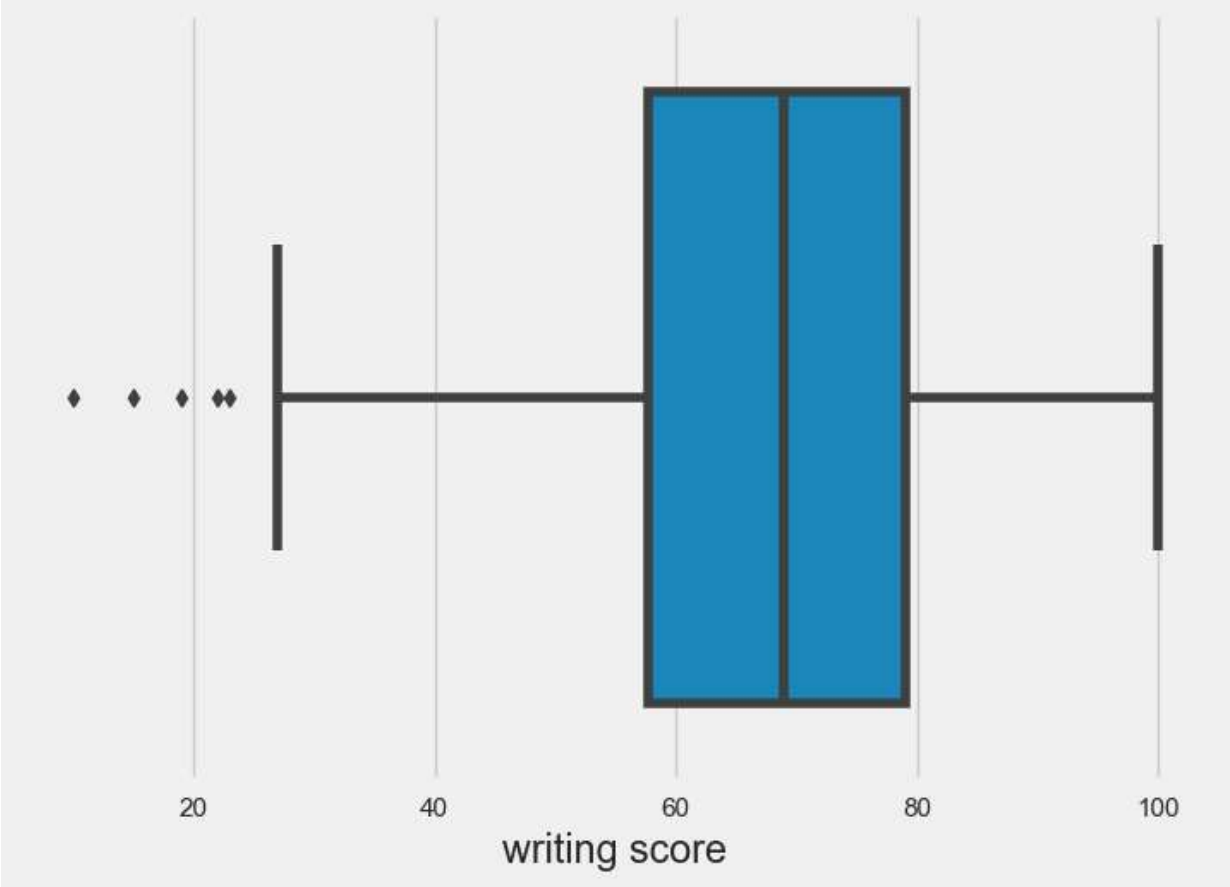
In [89]:
```
sns.boxplot(df['reading score'])
```

Out[89]:
```
<AxesSubplot:xlabel='reading score'>
```



In [90]:
```
sns.boxplot(df['writing score'])
```

Out[90]: <AxesSubplot:xlabel='writing score'>



In [91]:
```
q1=df['math score'].quantile(0.25)
q1
```

Out[91]: 57.0

In [92]:
```
q3=df['math score'].quantile(0.75)
q3
```

Out[92]: 77.0

In [93]:
```
IQR = q3-q1
```

In [94]:
```
upper_limit = q3+(1.5*IQR)
upper_limit
```

Out[94]: 107.0

In [95]:
```
lower_limit = q1-(1.5*IQR)
lower_limit
```

Out[95]: 27.0

In [96]:
```
df[df['math score']<lower_limit]  # indicate total outliear
```

Out[96]:

	gender	race/ethnicity	parental level of education	lunch	test preparation course	math score	reading score	writing score	Average
17	female	group B	some high school	free/reduced	none	18	32	28	26.000000
59	female	group C	some high school	free/reduced	none	0	17	10	9.000000
145	female	group C	some college	free/reduced	none	22	39	33	31.333333
338	female	group B	some high school	free/reduced	none	24	38	27	29.666667
466	female	group D	associate's degree	free/reduced	none	26	31	38	31.666667
787	female	group B	some college	standard	none	19	38	32	29.666667
842	female	group B	high school	free/reduced	completed	23	44	36	34.333333
980	female	group B	high school	free/reduced	none	8	24	23	18.333333

In [97]:
```
df[df['math score']>upper_limit]
```

Out[97]:

	gender	race/ethnicity	parental level of education	lunch	test preparation course	math score	reading score	writing score	Average

In [98]:
```
df[df['reading score']<lower_limit]
```

Out[98]:

	gender	race/ethnicity	parental level of education	lunch	test preparation course	math score	reading score	writing score	Average
59	female	group C	some high school	free/reduced	none	0	17	10	9.000000
76	male	group E	some high school	standard	none	30	26	22	26.000000
327	male	group A	some college	free/reduced	none	28	23	19	23.333333
596	male	group B	high school	free/reduced	none	30	24	15	23.000000
980	female	group B	high school	free/reduced	none	8	24	23	18.333333

In [99]:
```
df[df['writing score']<lower_limit]
```

Out[99]:

	gender	race/ethnicity	parental level of education	lunch	test preparation course	math score	reading score	writing score	Average
59	female	group C	some high school	free/reduced	none	0	17	10	9.000000
76	male	group E	some high school	standard	none	30	26	22	26.000000
327	male	group A	some college	free/reduced	none	28	23	19	23.333333
596	male	group B	high school	free/reduced	none	30	24	15	23.000000
980	female	group B	high school	free/reduced	none	8	24	23	18.333333

In [100]:
```
num_column  # p values
```

Out[100]:
```
['math score', 'reading score', 'writing score']
```

In [101]:
```
df[num_column]
```

Out[101]:

	math score	reading score	writing score
0	72	72	74
1	69	90	88
2	90	95	93
3	47	57	44
4	76	78	75
...	...	...	...
995	88	99	95
996	62	55	55
997	59	71	65
998	68	78	77
999	77	86	86

1000 rows × 3 columns

In [102]:
```
df_num=df[num_column]
```

In [103]:
```
df_num.head()
```

Out[103]:

	math score	reading score	writing score
0	72	72	74
1	69	90	88
2	90	95	93
3	47	57	44
4	76	78	75

In [104]:
```
from scipy.stats import normaltest
```

In [105]:
```
normaltest(df_num['math score'])
```

Out[105]:
```
NormaltestResult(statistic=15.408960513931822, pvalue=0.00045080293869937836)
```

In [106]:
```
normaltest(df_num['math score'])[1]
```

Out[106]:
```
0.00045080293869937836
```

In [107]:
```
normaltest(df_num['math score'])[0]
```

Out[107]:
```
15.408960513931822
```

In [108]:
```
def outlier_threshold(df,variable):
    q1 = df[variable].quantile(0.25)
    q3 = df[variable].quantile(0.75)
    IQR = q3-q1
    upper_limit = q3+(1.5*IQR)
    lower_limit = q1-(1.5*IQR)
    return lower_limit, upper_limit
```

In [109]:
```
def replace_with_threshold(df,numeric_col):
    for variable in numeric_col:
        lower_limit,upper_limit=outlier_threshold(df_num,variable)
        df.loc[df[variable]<lower_limit,variable]=lower_limit
        df.loc[df[variable]<upper_limit,variable]=upper_limit
```

In [110]:
```
replace_with_threshold(df_num,df_num.columns)
```

In [111]:
```
df_num.columns
```

Out[111]:
```
Index(['math score', 'reading score', 'writing score'], dtype='object')
```

In [112]:
```
df_num.loc[df_num['math score']<lower_limit,'math score']=lower_limit
```

In [113]:
```
df_num
```

Out[113]:

	math score	reading score	writing score
0	107	109	110.875
1	107	109	110.875
2	107	109	110.875
3	107	109	110.875
4	107	109	110.875
...	...	...	...
995	107	109	110.875
996	107	109	110.875
997	107	109	110.875
998	107	109	110.875
999	107	109	110.875

1000 rows × 3 columns

In [ ]: