Group Name : Data_Wizards

Group Members

Pushkar Jain

Tanuj Pancholi

In [123]:

```
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
sns.set() #setting seaborn default for plots
```

Note - Any comment related to a cell is mentioned exactly below it

In [124]:

```
df = pd.read_csv('C://Users//Pushkar Rajesh Jain//Desktop//Pushkar Jain//Data
 Sci//Practice.csv')
```

In [125]:

```
df.head() #Preveiw of data
```

Out[125]:

	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

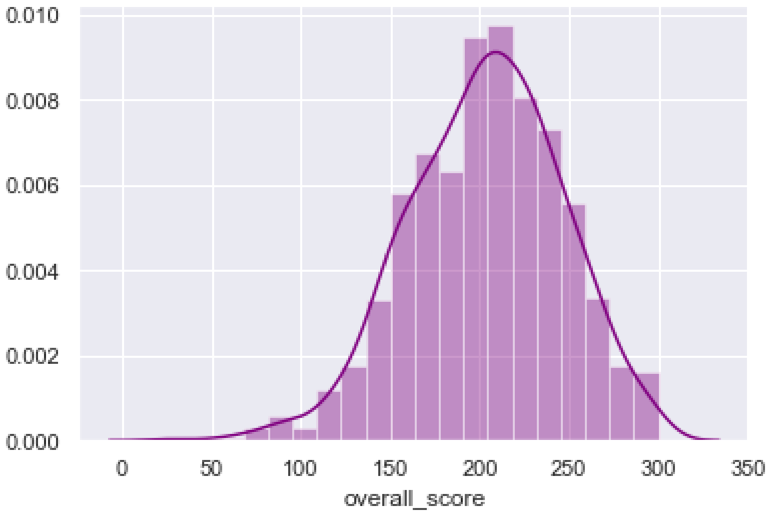
In [126]:
```
df.info()
```

```
<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               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
```

In [127]:
```
df['overall_score'] = df["math score"]+df["reading score"]+df["writing score"]
```

In [128]:
```
sns.distplot(df["overall_score"], bins=20, color="purple")
```

Out[128]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1bc07f8f948>
```



In [129]:

```
df.describe() #statistical values
```

Out[129]:

	math score	reading score	writing score	overall_score
count	1000.00000	1000.000000	1000.000000	1000.000000
mean	66.08900	69.169000	68.054000	203.312000
std	15.16308	14.600192	15.195657	42.771978
min	0.00000	17.000000	10.000000	27.000000
25%	57.00000	59.000000	57.750000	175.000000
50%	66.00000	70.000000	69.000000	205.000000
75%	77.00000	79.000000	79.000000	233.000000
max	100.00000	100.000000	100.000000	300.000000

Deduction:

As in all cases CV(std/mean)<1, the standard deviation is low. That's why we an use mean and median to predict center of a numerical data set.

Math Score data is centered aroud 66 marks

Reading Score data is centered aroud 69 marks

Writing Score data is centered aroud 68 marks

Overall Score data is centered aroud 203 marks

In [130]:
```
df.describe(include="all") # including all non-int values also
```

Out[130]:

	gender	race/ethnicity	parental level of education	lunch	test preparation course	math score	reading score	w
count	1000	1000	1000	1000	1000	1000.00000	1000.000000	1000.0
unique	2	5	6	2	2	NaN	NaN	
top	female	group C	some college	standard	none	NaN	NaN	
freq	518	319	226	645	642	NaN	NaN	
mean	NaN	NaN	NaN	NaN	NaN	66.08900	69.169000	68.0
std	NaN	NaN	NaN	NaN	NaN	15.16308	14.600192	15.1
min	NaN	NaN	NaN	NaN	NaN	0.00000	17.000000	10.0
25%	NaN	NaN	NaN	NaN	NaN	57.00000	59.000000	57.7
50%	NaN	NaN	NaN	NaN	NaN	66.00000	70.000000	69.0
75%	NaN	NaN	NaN	NaN	NaN	77.00000	79.000000	79.0
max	NaN	NaN	NaN	NaN	NaN	100.00000	100.000000	100.0

In [131]:
```
df.shape #shape of data
```

Out[131]:
```
(1000, 9)
```

In [132]:
```
df.isnull().sum() #Rechecking for NaN
```

Out[132]:
```
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
overall_score                  0
dtype: int64
```

Deduction:

No null values, so no data cleaning required

In [133]:
```
df.columns #column names
```

Out[133]:
```
Index(['gender', 'race/ethnicity', 'parental level of education', 'lunch',
       'test preparation course', 'math score', 'reading score',
       'writing score', 'overall_score'],
      dtype='object')
```

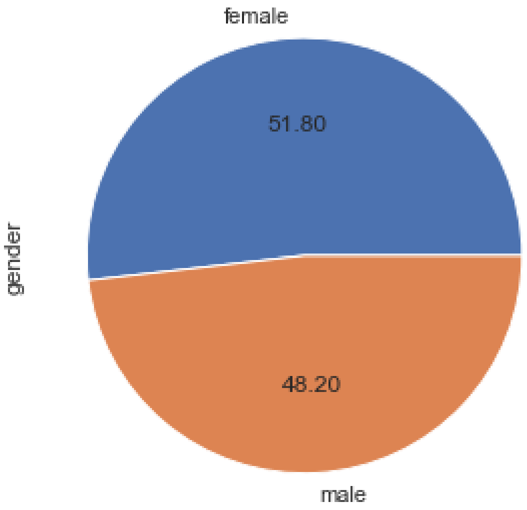
Finding Trends in each Column

```
In [135]: df['gender'].value_counts()
```

```
Out[135]: female    518
          male      482
          Name: gender, dtype: int64
```

```
In [136]: df.groupby('gender')['gender'].count().plot.pie(autopct='%.2f',figsize=(5,5))
```

```
Out[136]: <matplotlib.axes._subplots.AxesSubplot at 0x1bc08012a88>
```

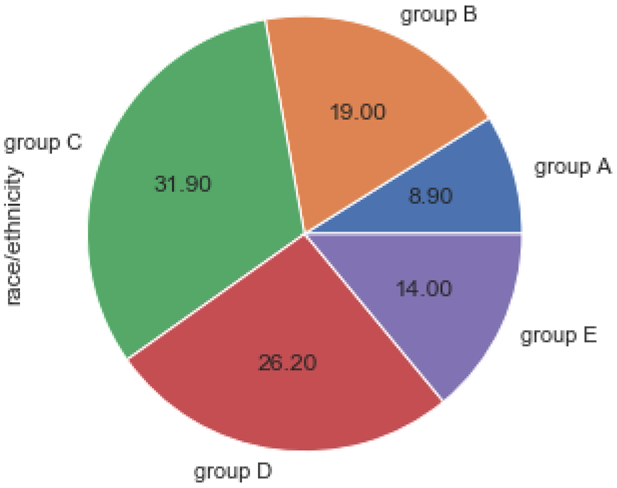


```
In [137]: df['race/ethnicity'].value_counts()
```

```
Out[137]: group C    319
          group D    262
          group B    190
          group E    140
          group A     89
          Name: race/ethnicity, dtype: int64
```

```
In [138]: df.groupby('race/ethnicity')['race/ethnicity'].count().plot.pie(autopct='%.2f'
          ,figsize=(5,5))
```

```
Out[138]: <matplotlib.axes._subplots.AxesSubplot at 0x1bc0807ae08>
```



```
In [139]: df['parental level of education'].value_counts()
```

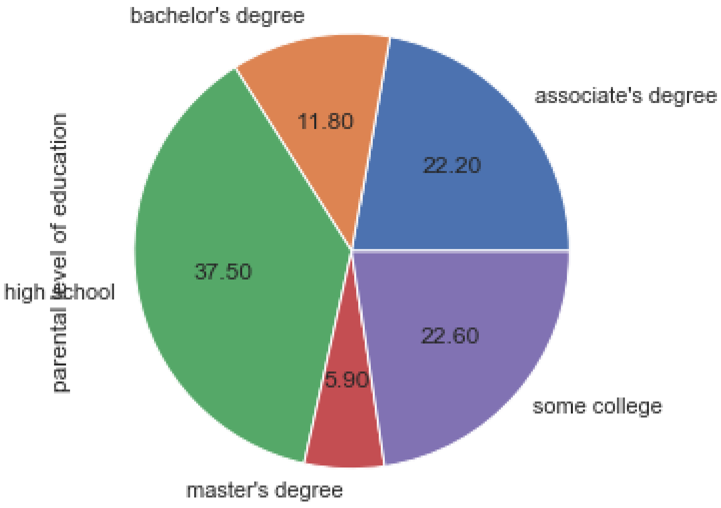
```
Out[139]: some college          226
          associate's degree    222
          high school           196
          some high school      179
          bachelor's degree     118
          master's degree        59
          Name: parental level of education, dtype: int64
```

```
In [140]: df['parental level of education'].replace(["high school","some high school"],[
          "high school","high school"],inplace=True)
```

Combining "High school" and "some High school"

```
In [141]: df.groupby('parental level of education')['parental level of education'].count
().plot.pie(autopct='%.2f',figsize=(5,5))
```

```
Out[141]: <matplotlib.axes._subplots.AxesSubplot at 0x1bc080dce48>
```

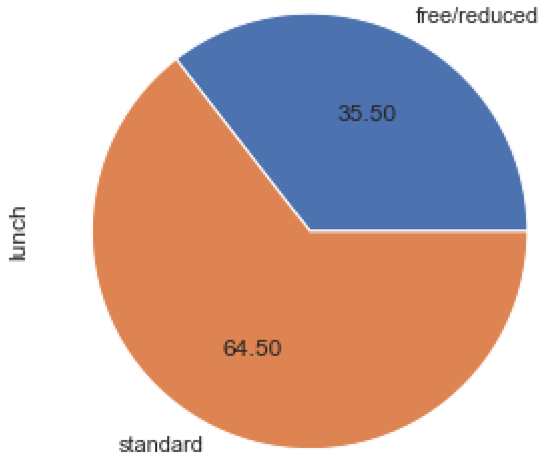


```
In [142]: df['lunch'].value_counts()
```

```
Out[142]: standard        645
          free/reduced    355
          Name: lunch, dtype: int64
```

```
In [143]: df.groupby('lunch')['lunch'].count().plot.pie(autopct='%.2f',figsize=(5,5))
```

```
Out[143]: <matplotlib.axes._subplots.AxesSubplot at 0x1bc0813b408>
```

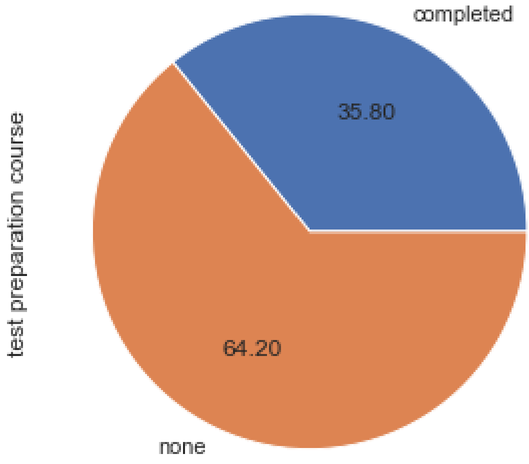


```
In [144]: df['test preparation course'].value_counts()
```

```
Out[144]: none         642
          completed    358
          Name: test preparation course, dtype: int64
```

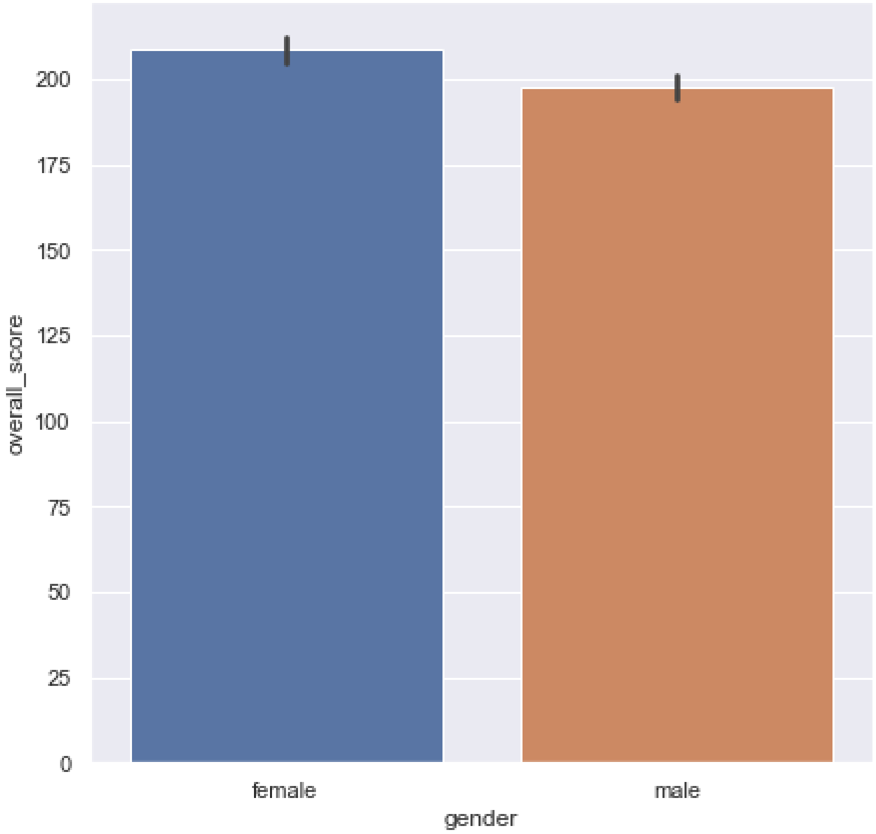
In [145]:
```
df.groupby('test preparation course')['test preparation course'].count().plot.
pie(autopct='%.2f',figsize=(5,5))
```

Out[145]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1bc06a43d08>
```



Basic Analysis from the data

In [146]:
```
g = sns.barplot(x="gender", y="overall_score", data=df)
g.figure.set_size_inches(7,7)
```

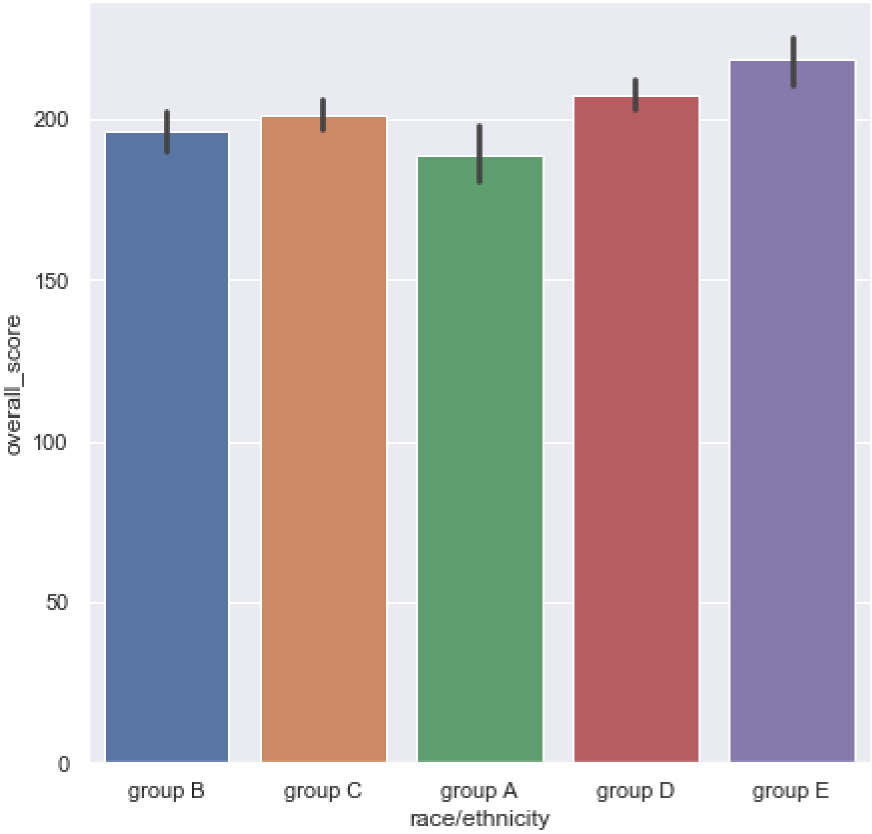


Deduction

Females score better than males

In [147]:

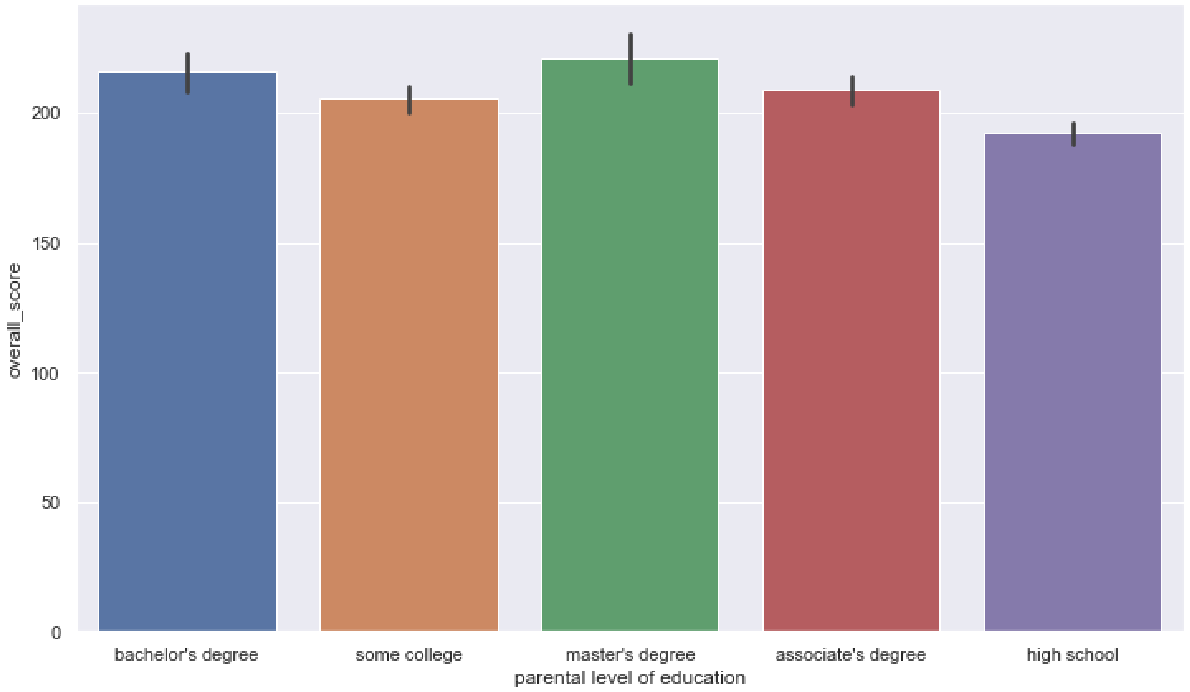
```
g = sns.barplot(x="race/ethnicity", y="overall_score", data=df)
g.figure.set_size_inches(7,7)
```



Deduction

Group E students are the best performers and Group A students are the worst.

```
In [148]: g = sns.barplot(x="parental level of education", y="overall_score", data=df)
          g.figure.set_size_inches(12,7)
```

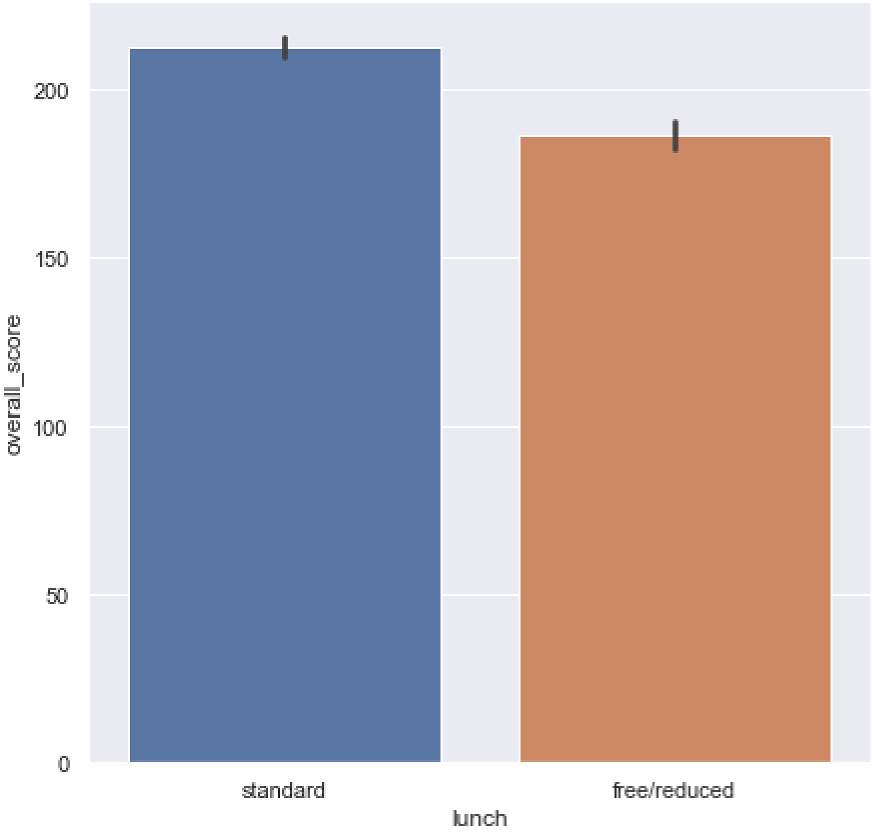


Deduction

As per parental education level increases, the scores of children increases

In [149]:

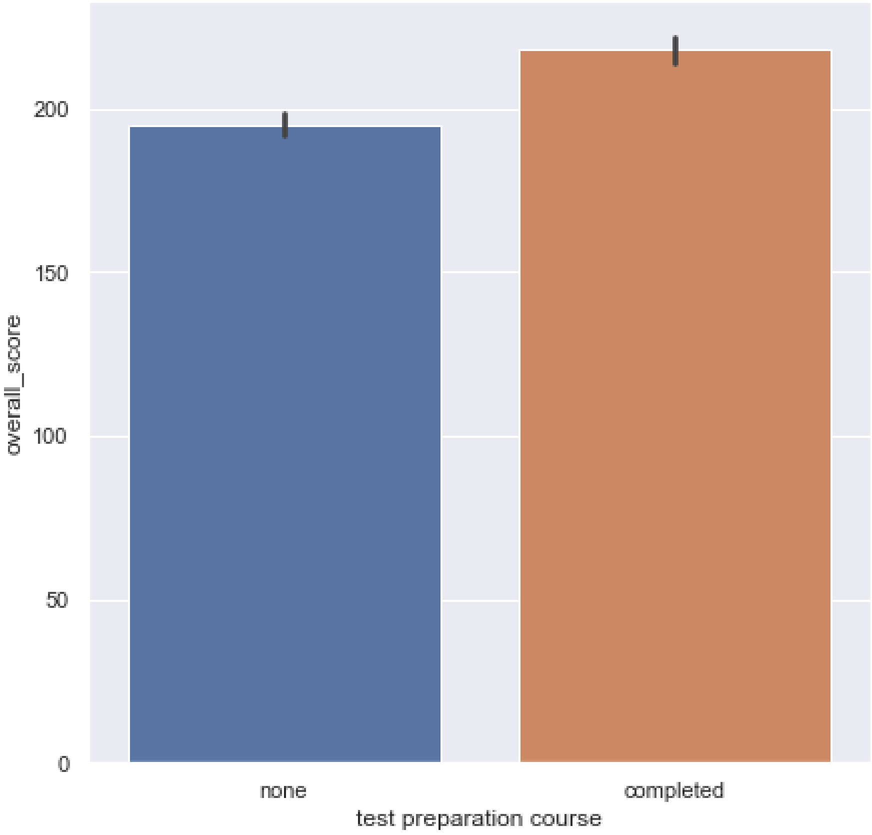
```
g = sns.barplot(x="lunch", y="overall_score", data=df)
g.figure.set_size_inches(7,7)
```



Deduction

Standard lunch students are better performers than free lunch students

In [150]:
```
g = sns.barplot(x="test preparation course", y="overall_score", data=df)
g.figure.set_size_inches(7,7)
```



Deduction

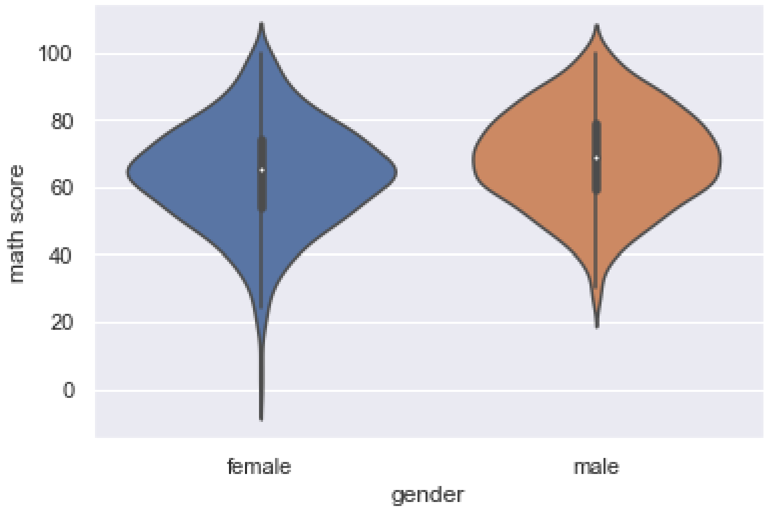
Students who have completed the Test Preparation Course have outperformed

Analysing traits for the student having highest Maths Score

As stated earlier as the standard deviation is low, so median will give us a better idea of the data. It will point out the central value and by using violin plots we will be able to see the population density around the same

```
In [151]: sns.violinplot(x = "gender", y = "math score", data = df)
```

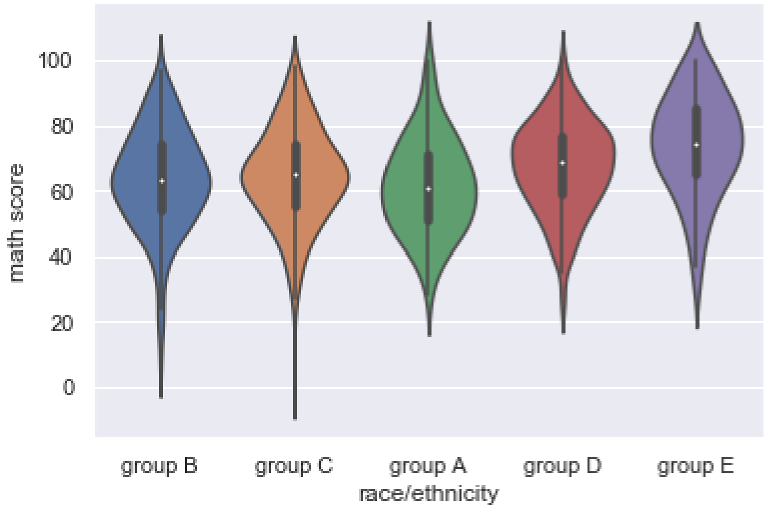
```
Out[151]: <matplotlib.axes._subplots.AxesSubplot at 0x1bc081ca5c8>
```



The median score of males is higher, also the population density of males at higher score is greater than males

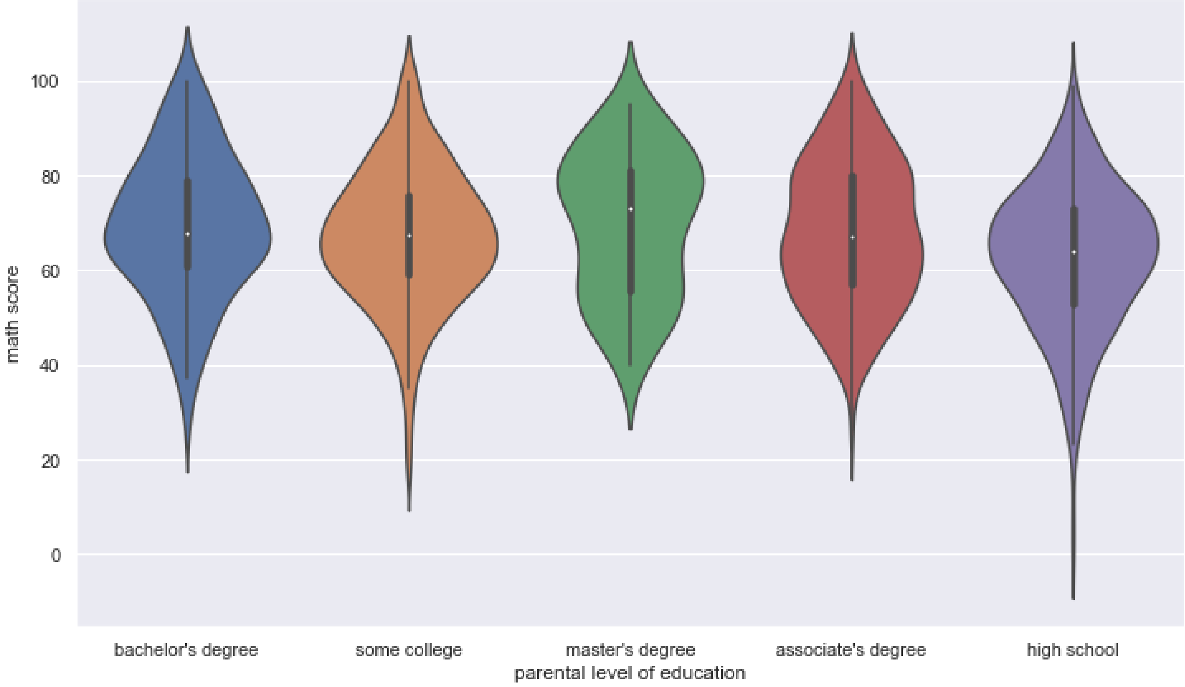
```
In [152]: sns.violinplot(x = "race/ethnicity", y = "math score", data = df)
```

```
Out[152]: <matplotlib.axes._subplots.AxesSubplot at 0x1bc081d9a08>
```



The median score of Group E is higher, also the population density of Group E at higher score is greater than other groups.

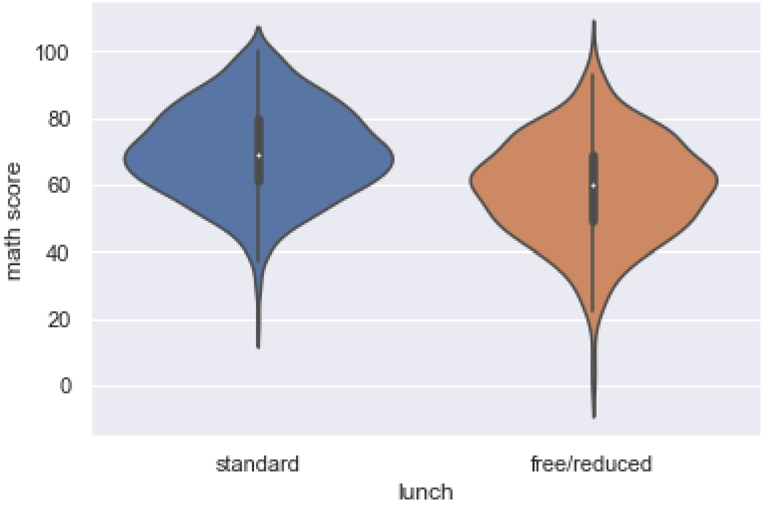
In [153]:
```
g=sns.violinplot(x = "parental level of education", y = "math score", data = d
f)
g.figure.set_size_inches(12,7)
```



The median score of children who's parents have Masters is higher, also their population density at higher score is greater than other groups.

In [154]:
```
sns.violinplot(x = "lunch", y = "math score", data = df)
```

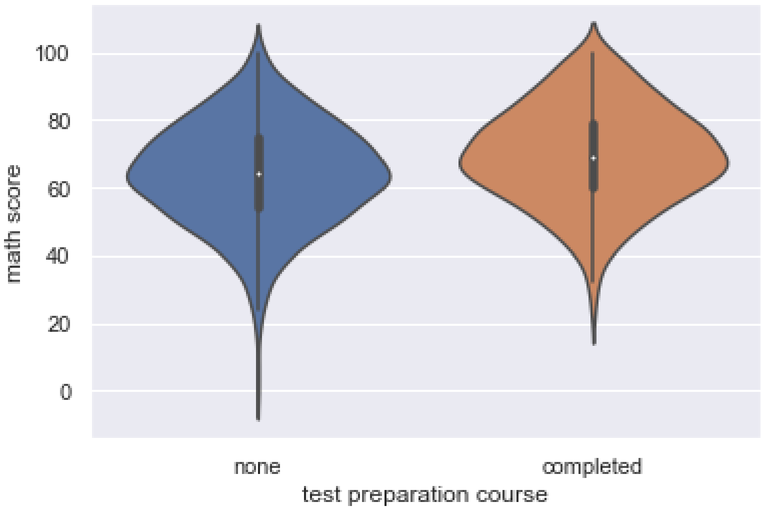
Out[154]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1bc02df24c8>
```



The median score of standard lunch students is higher, also their population density at higher score is greater than other groups.

```
In [155]: sns.violinplot(x = "test preparation course", y = "math score", data = df)
```

```
Out[155]: <matplotlib.axes._subplots.AxesSubplot at 0x1bc094e62c8>
```



The median score of students who completed the Test Preparation Course is higher, also their population density at higher score is greater than other groups

Deduction:

Following are the traits for high Math Score students -

Gender - Male

Race/Ethinicity - Group E

Parental level of education - Masters Degree

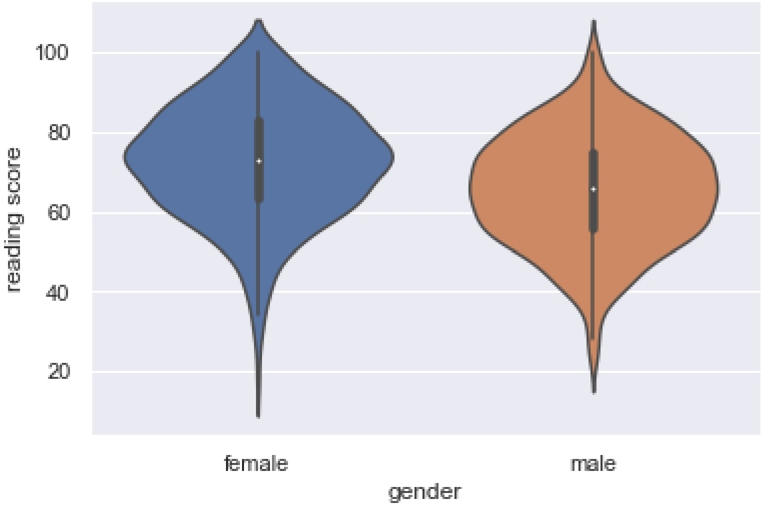
Lunch - Standard

Test Preparation Course - Completed

Analysing traits for the student having highest Reading Score

```
In [156]: sns.violinplot(x = "gender", y = "reading score", data = df)
```

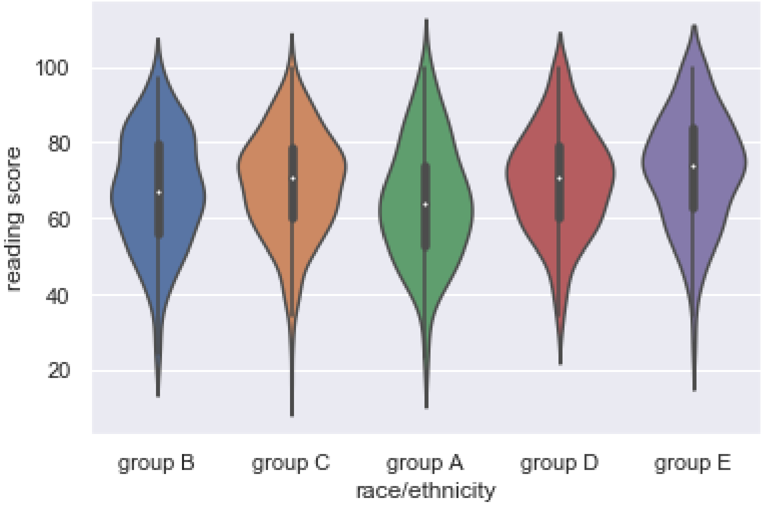
```
Out[156]: <matplotlib.axes._subplots.AxesSubplot at 0x1bc0953ffc8>
```



The median score of females is higher, also the population density of females at higher score is greater than males

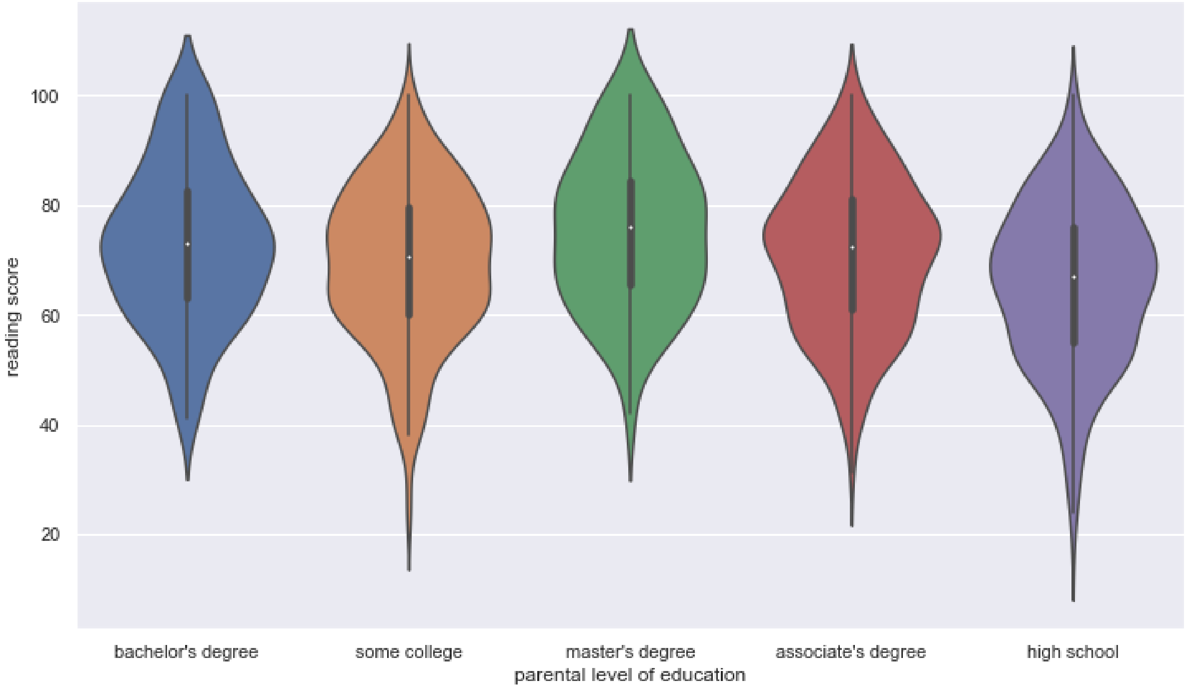
```
In [157]: sns.violinplot(x = "race/ethnicity", y = "reading score", data = df)
```

```
Out[157]: <matplotlib.axes._subplots.AxesSubplot at 0x1bc095a9288>
```



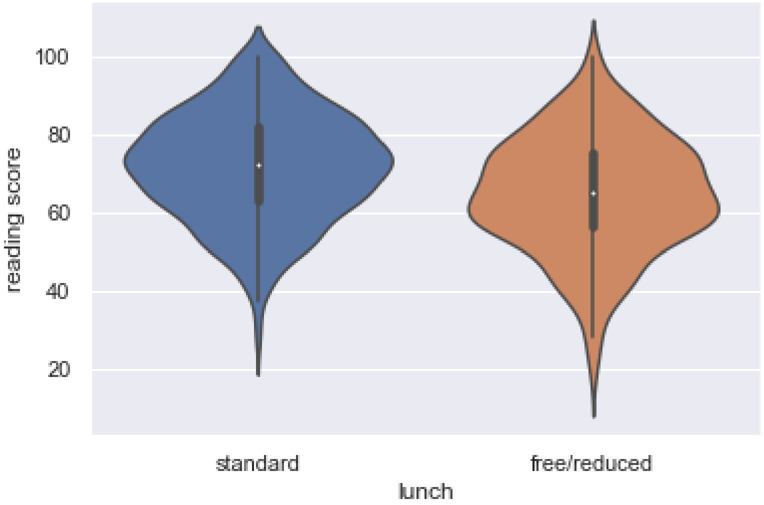
The median score of Group E is higher, also the population density of Group E at higher score is greater than other groups.

```
In [158]: g=sns.violinplot(x = "parental level of education", y = "reading score", data
          = df)
          g.figure.set_size_inches(12,7)
```



The median score of children who's parents have Masters is higher, also their population density at higher score is greater than other groups.

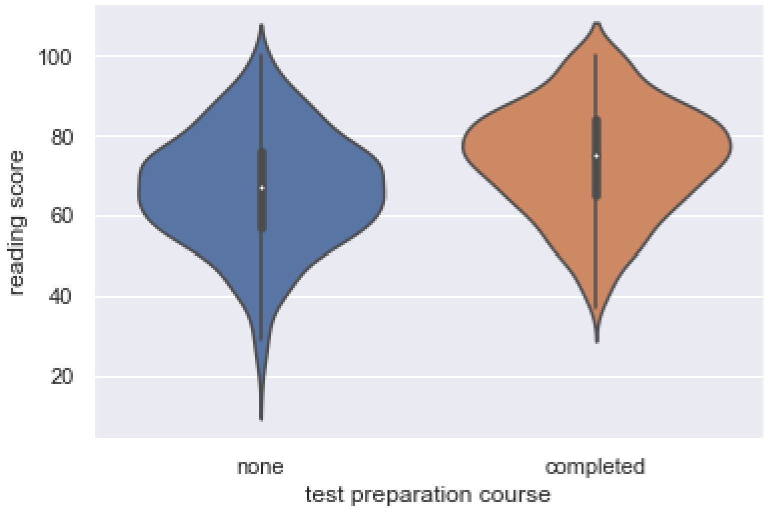
```
In [159]: sns.violinplot(x = "lunch", y = "reading score", data = df)

Out[159]: <matplotlib.axes._subplots.AxesSubplot at 0x1bc099d0548>
```



The median score of standard lunch students is higher, also their population density at higher score is greater than other groups.

```
In [160]: sns.violinplot(x = "test preparation course", y = "reading score", data = df)
```

```
Out[160]: <matplotlib.axes._subplots.AxesSubplot at 0x1bc09835748>
```



The median score of students who completed the Test Preparation Course is higher, also their population density at higher score is greater than other groups

Deduction:

Following are the traits for high Reading Score students -

Gender - Female

Race/Ethinicity - Group E

Parental level of education - Masters Degree

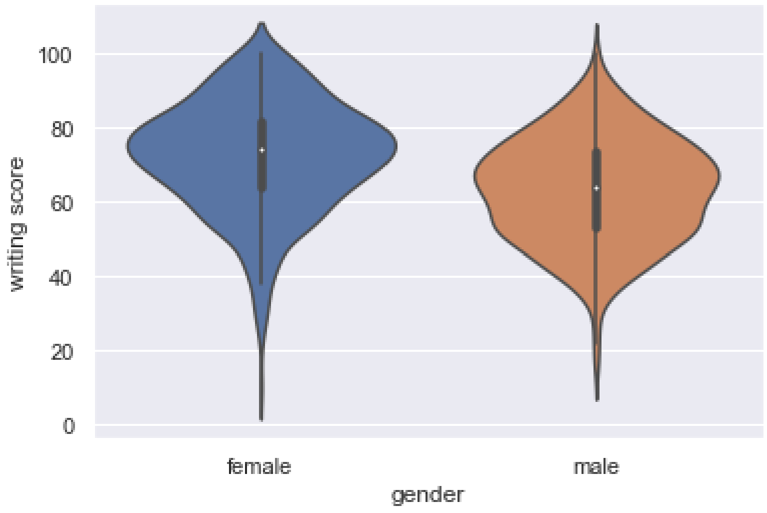
Lunch - Standard

Test Preparation Course - Completed

Analysing traits for the student having highest Writing Score

```
In [161]: sns.violinplot(x = "gender", y = "writing score", data = df)
```

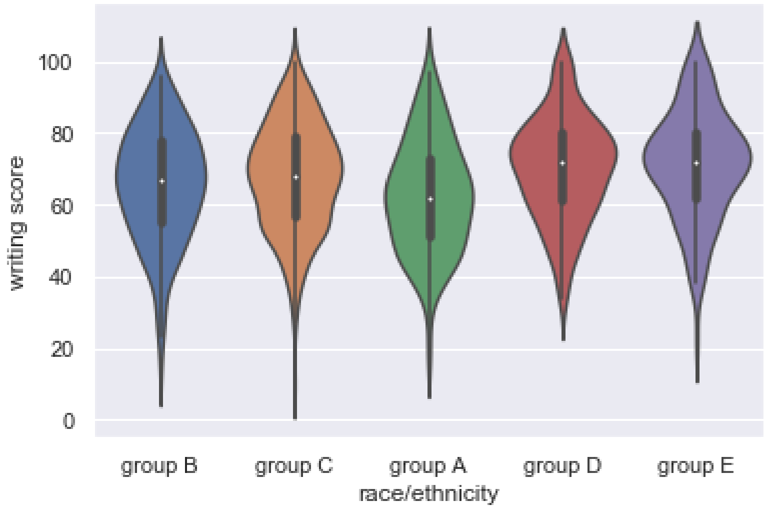
```
Out[161]: <matplotlib.axes._subplots.AxesSubplot at 0x1bc09895188>
```



The median score of females is higher, also the population density of females at higher score is greater than males

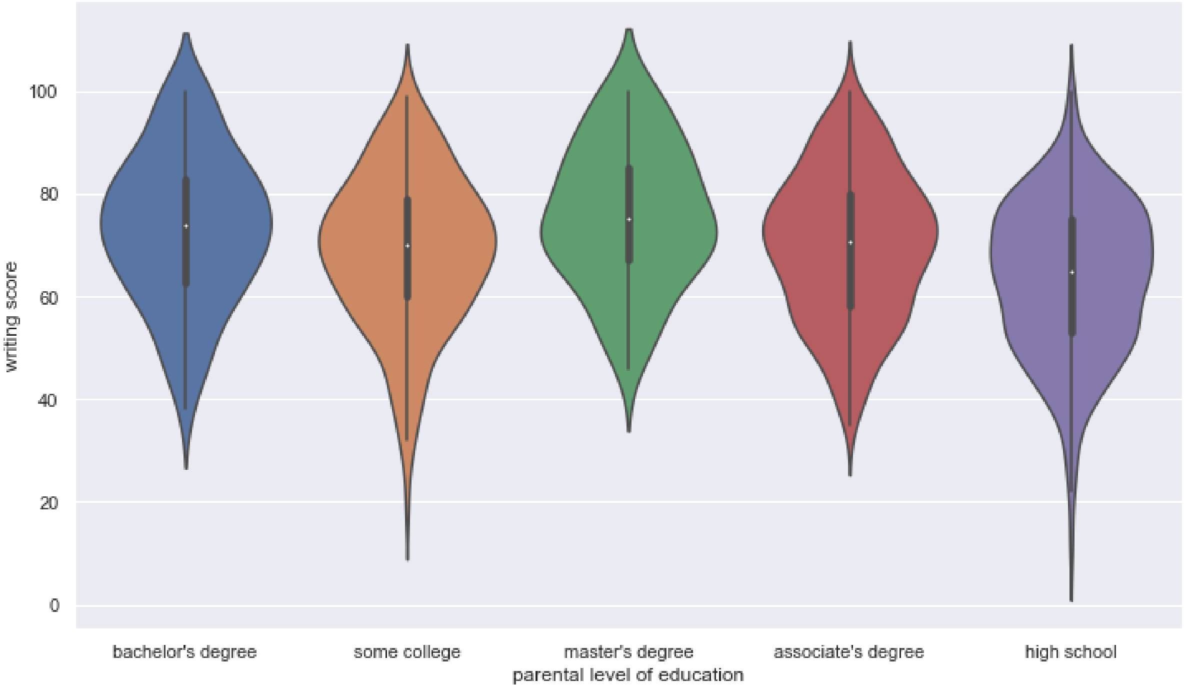
```
In [162]: sns.violinplot(x = "race/ethnicity", y = "writing score", data = df)
```

```
Out[162]: <matplotlib.axes._subplots.AxesSubplot at 0x1bc099025c8>
```



Though the median score of Group E and Group D is same, the population density of Group D at higher score is greater than other groups.

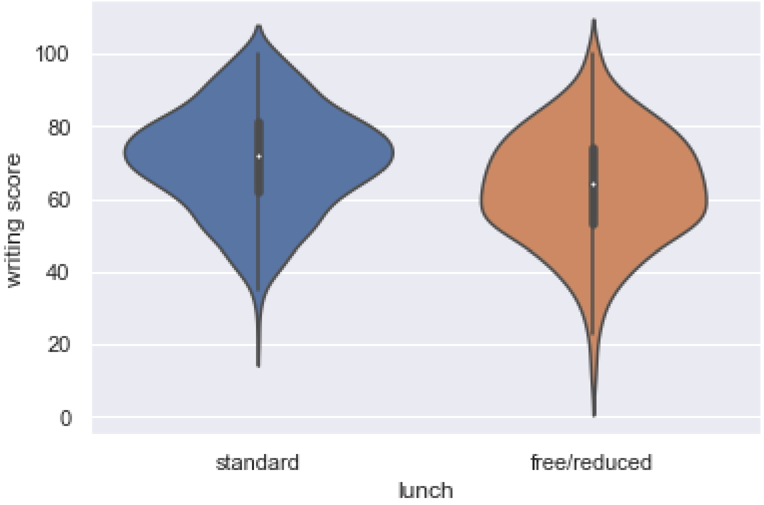
In [163]:
```
g=sns.violinplot(x = "parental level of education", y = "writing score", data
= df)
g.figure.set_size_inches(12,7)
```



Though the median score of children who's parents have Masters and Bachelors degree is same, the population density of children who's parents have Bachelors degree at higher score is greater than other groups.

In [164]:
```
sns.violinplot(x = "lunch", y = "writing score", data = df)
```

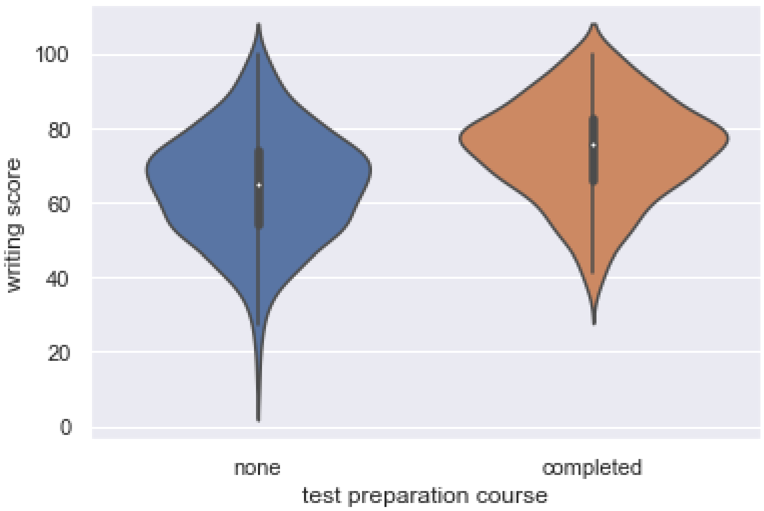
Out[164]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1bc081c6ec8>
```



The median score of standard lunch students is higher, also their population density at higher score is greater than other groups.

```
In [165]: sns.violinplot(x = "test preparation course", y = "writing score", data = df)
```

```
Out[165]: <matplotlib.axes._subplots.AxesSubplot at 0x1bc07ec7f08>
```



The median score of students who completed the Test Preparation Course is higher, also their population density at higher score is greater than other groups

Deduction:

Following are the traits for high Writing Score students -

Gender - Female

Race/Ethinicity - Group D

Parental level of education - Bachelors Degree

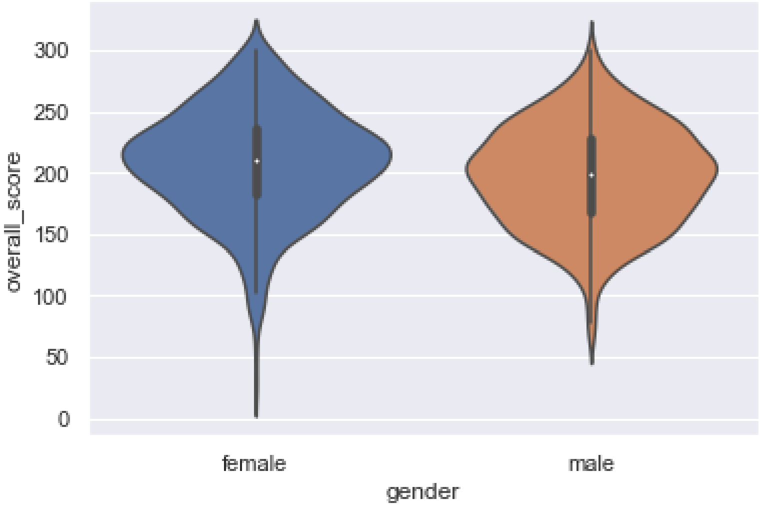
Lunch - Standard

Test Preparation Course - Completed

Analysing traits for the student having highest Overall Score

```
In [166]: sns.violinplot(x = "gender", y = "overall_score", data = df)
```

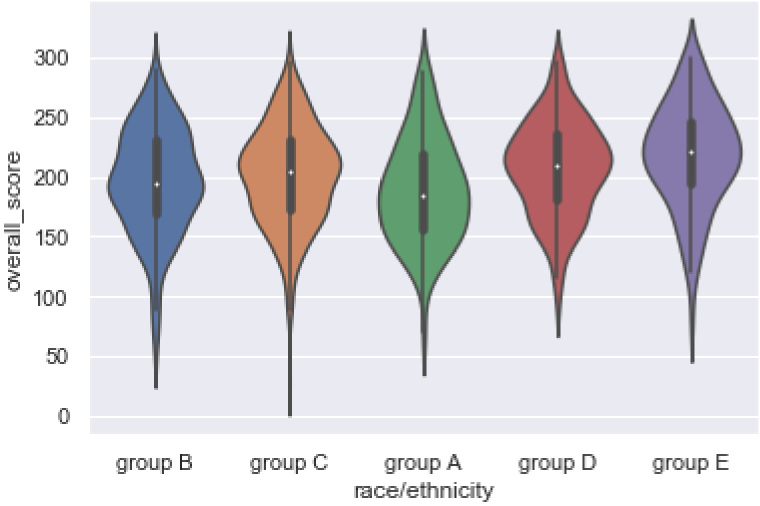
```
Out[166]: <matplotlib.axes._subplots.AxesSubplot at 0x1bc07e70e88>
```



The median score of females is higher, also the population density of females at higher score is greater than males

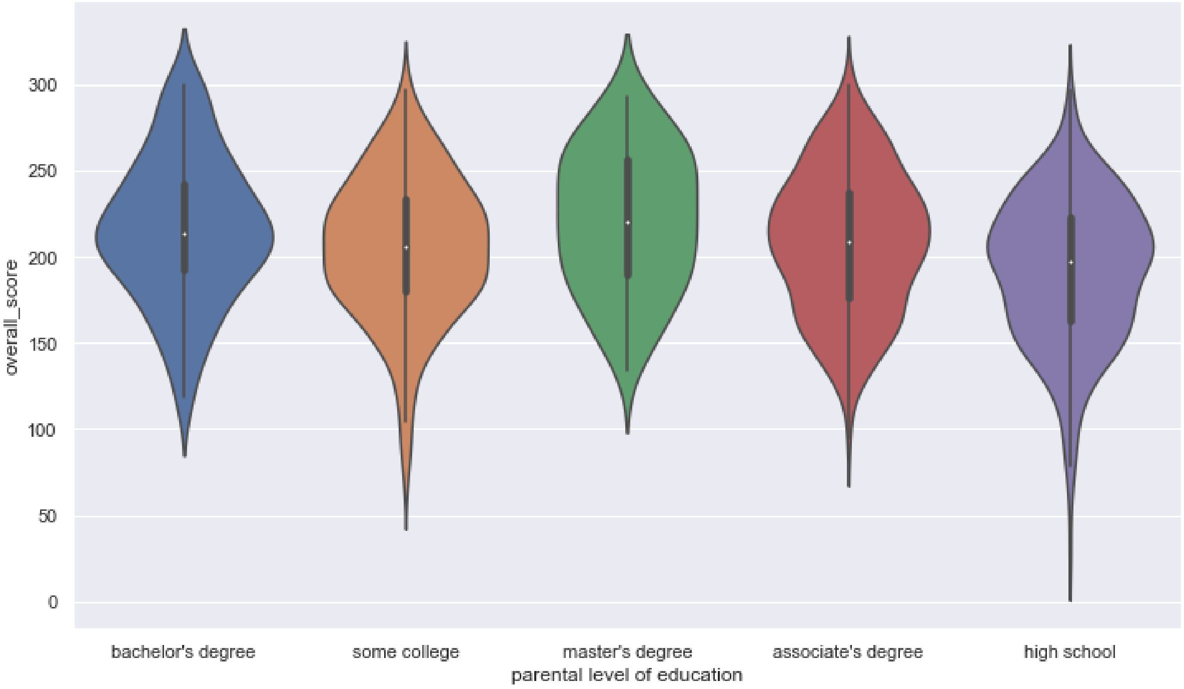
```
In [167]: sns.violinplot(x = "race/ethnicity", y = "overall_score", data = df)
```

```
Out[167]: <matplotlib.axes._subplots.AxesSubplot at 0x1bc06511708>
```



The median score of Group E is higher, also the population density of Group E at higher score is greater than other groups.

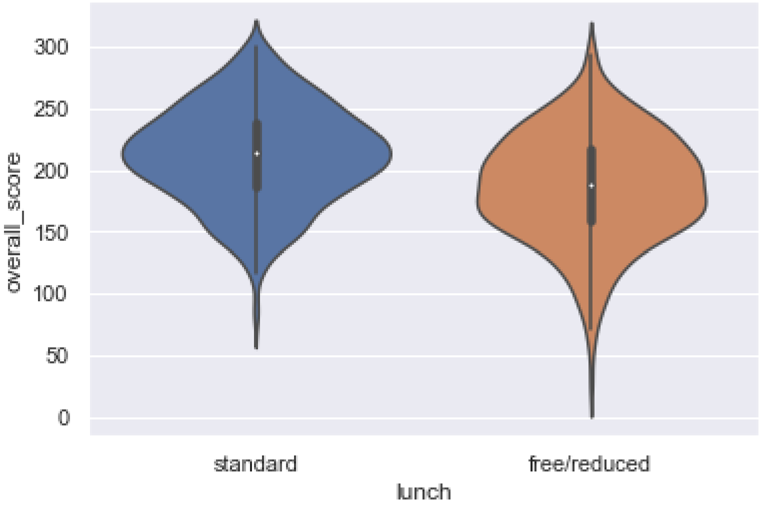
In [168]:
```
g=sns.violinplot(x = "parental level of education", y = "overall_score", data
= df)
g.figure.set_size_inches(12,7)
```



The median score of children who's parents have Masters is higher, also their population density at higher score is greater than other groups.

In [169]:
```
sns.violinplot(x = "lunch", y = "overall_score", data = df)
```

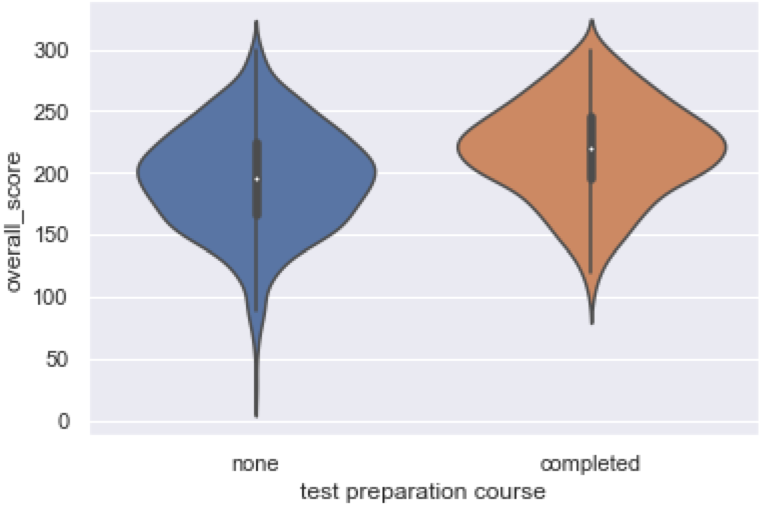
Out[169]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1bc07e24d48>
```



The median score of standard lunch students is higher, also their population density at higher score is greater than other groups.

```
In [170]: sns.violinplot(x = "test preparation course", y = "overall_score", data = df)
```

```
Out[170]: <matplotlib.axes._subplots.AxesSubplot at 0x1bc09665dc8>
```



The median score of students who completed the Test Preparation Course is higher, also their population density at higher score is greater than other groups

Deduction:

Following are the traits for high Overall Score students -

Gender - Female

Race/Ethinicity - Group E

Parental level of education - Masters Degree

Lunch - Standard

Test Preparation Course - Completed

Summary:

1.Females score better than males

2.Group E students are the best performers and Group A students are the worst.

3.As per parental education level increases, the scores of children increases

4.Standard lunch students are better performers than free lunch students

5.Students who have completed the Test Preparation Course have outperformed

6.Following are the traits for high Score students -

a)Gender - Female

b)Race/Ethinicity - Group E

c)Parental level of education - Masters Degree

d)Lunch - Standard

Hot encoding

In [171]:

```
df['gender'].replace(["male","female"],[0,1],inplace=True)
```

In [172]:

```
df['race/ethnicity'].replace(["group A","group B","group C","group D","group E"],[0,1,2,3,4],inplace=True)
```

In [173]:

```
df['parental level of education'].replace(["high school","some college","associate's degree","bachelor's degree","master's degree"],[0,1,2,3,4],inplace=True)
```

In [174]:

```
df['lunch'].replace(["standard","free/reduced"],[0,1],inplace=True)
```

In [175]:

```
df['test preparation course'].replace(["none","completed"],[0,1],inplace=True)
```

In [176]:
```
df.head()
```

Out[176]:

	gender	race/ethnicity	parental level of education	lunch	test preparation course	math score	reading score	writing score	overall_score
0	1	1	3	0	0	72	72	74	218
1	1	2	1	0	1	69	90	88	247
2	1	1	4	0	0	90	95	93	278
3	0	0	2	1	0	47	57	44	148
4	0	2	1	0	0	76	78	75	229

Data Dictionary:

Gender : 0 = Male, 1= Female

Race/Ethnicity : 0 = group A, 1 = group B, 2 = group C, 3 = group D, 4 = group E

Parental level of education : 0 = high school/some high school, 1 = some college, 2 = associates degree, 3 = bachelors degree, 4 = masters degree

Lunch : 0 = standard, 1 = free/reduced

Test preparation course : 0 = none, 1 = completed