In [4]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
import os
import warnings
warnings.filterwarnings('ignore')
```

In [5]:
```
df = pd.read_csv('data/stud.csv')
df.head()
```

Out[5]:

	gender	race_ethnicity	parental_level_of_education	lunch	test_preparation_course	math_score	reading_sc
0	female	group B	bachelor's degree	standard	none	72	
1	female	group C	some college	standard	completed	69	
2	female	group B	master's degree	standard	none	90	
3	male	group A	associate's degree	free/reduced	none	47	
4	male	group C	some college	standard	none	76	

In [6]:
```
df.tail()
```

Out[6]:

	gender	race_ethnicity	parental_level_of_education	lunch	test_preparation_course	math_score	reading
995	female	group E	master's degree	standard	completed	88	
996	male	group C	high school	free/reduced	none	62	
997	female	group C	high school	free/reduced	completed	59	
998	female	group D	some college	standard	completed	68	
999	female	group D	some college	free/reduced	none	77	

In [7]:
```
df.shape
```

Out[7]:
```
(1000, 8)
```

Data Checks

1. Check Missing values
2. Check Duplicates
3. Check data type
4. Check the number of unique values of each column
5. Check statistics of data set
6. Check various categories present in the different categorical column

In [8]:
```
#Check Missing values
df.isnull().sum()
```

Out[8]:
```
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 [12]:
```
#Check Duplicates
df.duplicated().sum()
```

Out[12]:
```
0
```

In [13]:
```
#Check data type
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 [16]:
```
#Check the number of unique values of each column
df.nunique()
```

Out[16]:
```
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 [17]:
```
#Check statistics of data set
df.describe()
```

Out[17]:

	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 [23]:
```
#categorical columns
[col for col in df.columns if df[col].dtypes=='O']
```

Out[23]:
```
['gender',
 'race_ethnicity',
 'parental_level_of_education',
 'lunch',
 'test_preparation_course']
```

In [22]:
```
#Check various categories present in the different categorical column
[df[col].unique() for col in df.columns if df[col].dtypes=='O']
```

Out[22]:
```
[array(['female', 'male'], dtype=object),
 array(['group B', 'group C', 'group A', 'group D', 'group E'],
       dtype=object),
 array(["bachelor's degree", 'some college', "master's degree",
        "associate's degree", 'high school', 'some high school'],
       dtype=object),
 array(['standard', 'free/reduced'], dtype=object),
 array(['none', 'completed'], dtype=object)]
```

In [24]:
```
# define numerical & categorical columns
numeric_features = [feature for feature in df.columns if df[feature].dtype != 'O']
categorical_features = [feature for feature in df.columns if df[feature].dtype == 'O']
```

In [25]:
```
#Adding 2 new columns total and avg score
df['total score'] = df['math_score'] + df['reading_score'] + df['writing_score']
df['average'] = df['total score']/3
df.head()
```

Out[25]:

	gender	race_ethnicity	parental_level_of_education	lunch	test_preparation_course	math_score	reading_sc
0	female	group B	bachelor's degree	standard	none	72	
1	female	group C	some college	standard	completed	69	
2	female	group B	master's degree	standard	none	90	
3	male	group A	associate's degree	free/reduced	none	47	
4	male	group C	some college	standard	none	76	

In [26]:
```
reading_full = df[df['reading_score'] == 100]['average'].count()
writing_full = df[df['writing_score'] == 100]['average'].count()
math_full = df[df['math_score'] == 100]['average'].count()

print(f'Number of students with full marks in Maths: {math_full}')
```

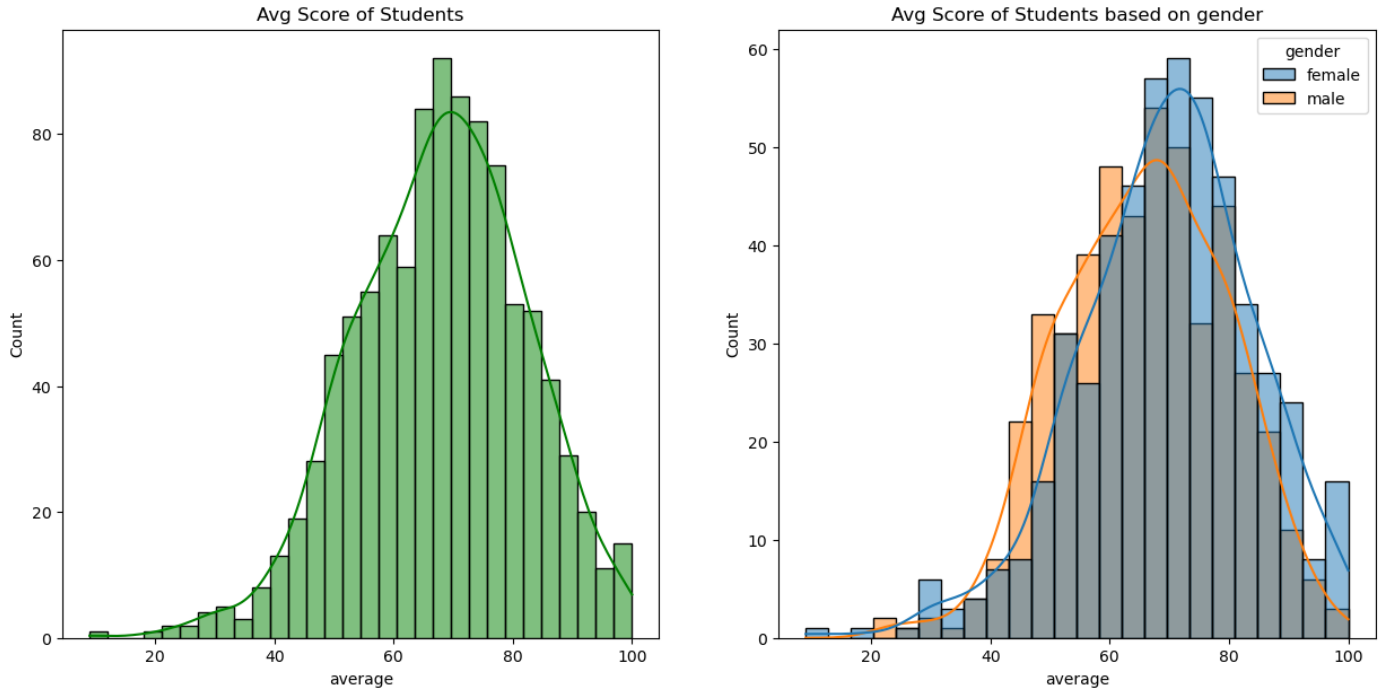
```
print(f'Number of students with full marks in Writing: {writing_full}')
print(f'Number of students with full marks in Reading: {reading_full}')
```

```
Number of students with full marks in Maths: 7
Number of students with full marks in Writing: 14
Number of students with full marks in Reading: 17
```

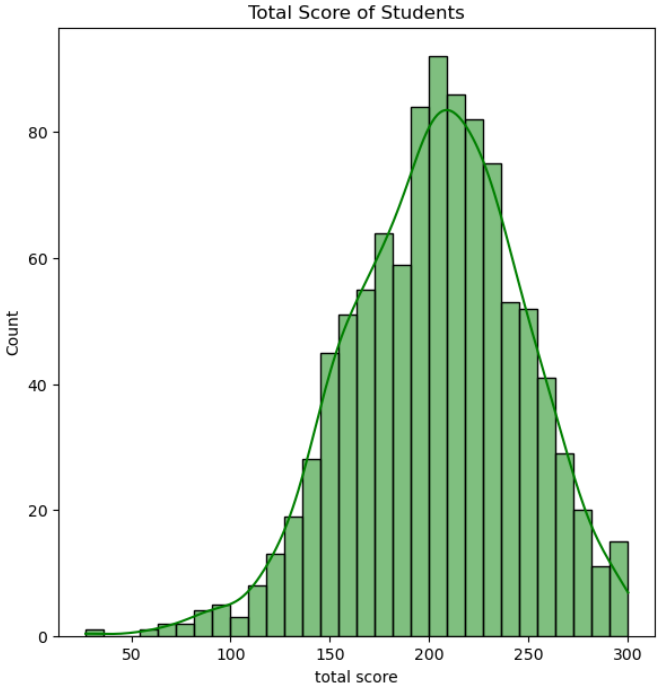
In [27]:
```
reading_less_20 = df[df['reading_score'] <= 20]['average'].count()
writing_less_20 = df[df['writing_score'] <= 20]['average'].count()
math_less_20 = df[df['math_score'] <= 20]['average'].count()

print(f'Number of students with less than 20 marks in Maths: {math_less_20}')
print(f'Number of students with less than 20 marks in Writing: {writing_less_20}')
print(f'Number of students with less than 20 marks in Reading: {reading_less_20}')
```

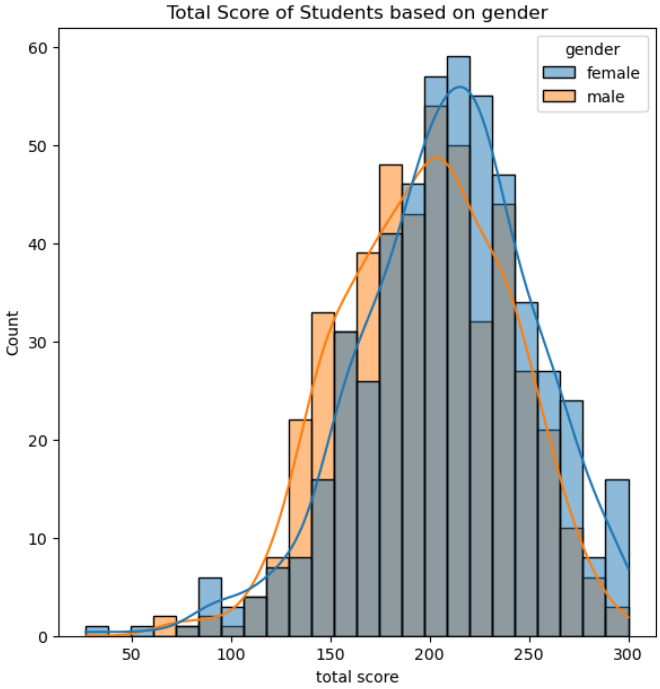
```
Number of students with less than 20 marks in Maths: 4
Number of students with less than 20 marks in Writing: 3
Number of students with less than 20 marks in Reading: 1
```

In [33]:
```
fig, axs = plt.subplots(1, 2, figsize=(15, 7))
plt.subplot(121)
plt.title('Avg Score of Students')
sns.histplot(data=df,x='average',bins=30,kde=True,color='g')
plt.subplot(122)
plt.title('Avg Score of Students based on gender')
sns.histplot(data=df,x='average',kde=True,hue='gender')
plt.show()
```

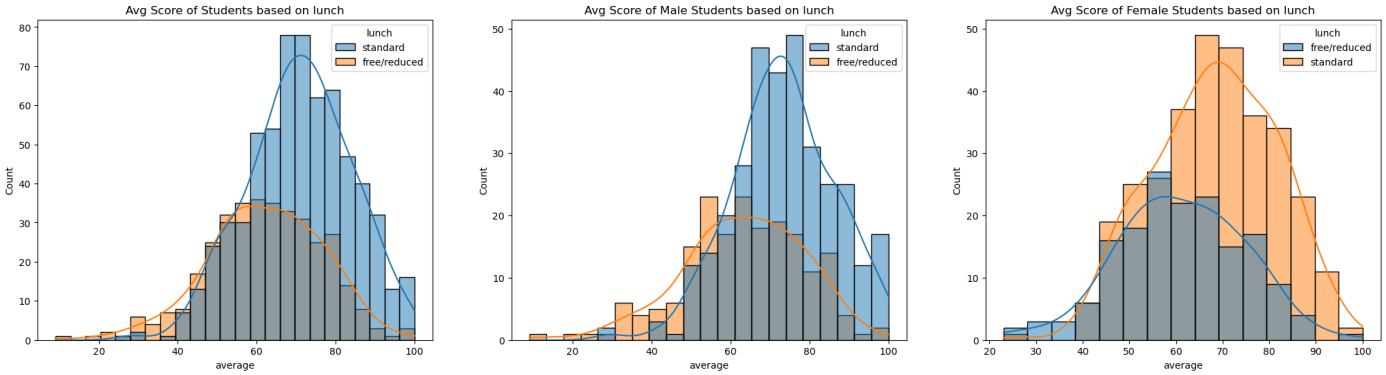


In [34]:
```
fig, axs = plt.subplots(1, 2, figsize=(15, 7))
plt.subplot(121)
plt.title('Total Score of Students')
sns.histplot(data=df,x='total score',bins=30,kde=True,color='g')
plt.subplot(122)
plt.title('Total Score of Students based on gender')
sns.histplot(data=df,x='total score',kde=True,hue='gender')
plt.show()
```

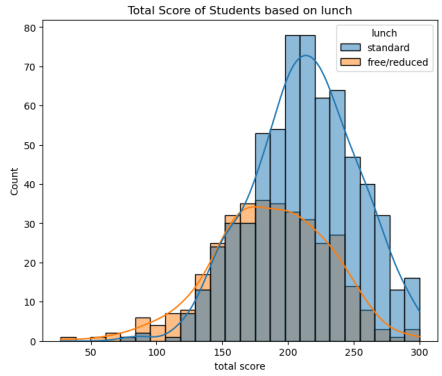


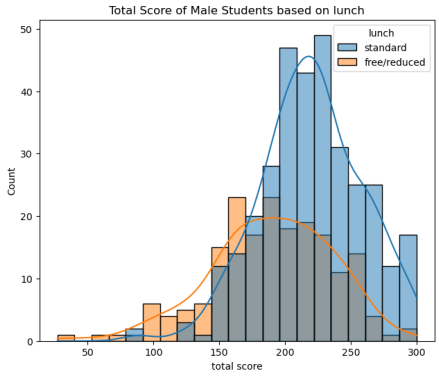


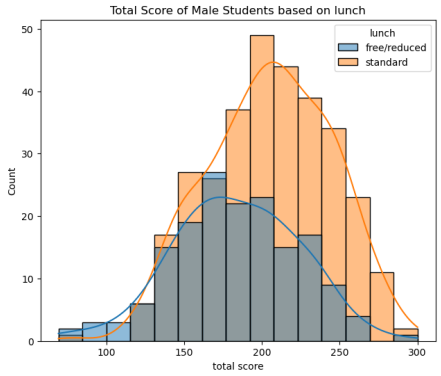
In [35]:
```
plt.subplots(1,3,figsize=(25,6))
plt.subplot(131)
plt.title('Avg Score of Students based on lunch')
sns.histplot(data=df,x='average',kde=True,hue='lunch')
plt.subplot(132)
plt.title('Avg Score of Female Students based on lunch')
sns.histplot(data=df[df.gender=='female'],x='average',kde=True,hue='lunch')
plt.subplot(133)
plt.title('Avg Score of Male Students based on lunch')
sns.histplot(data=df[df.gender=='male'],x='average',kde=True,hue='lunch')
plt.show()
```



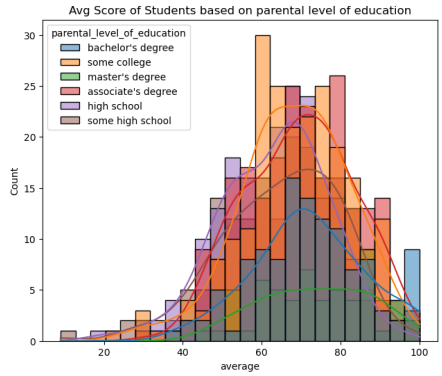
In [36]:
```
plt.subplots(1,3,figsize=(25,6))
plt.subplot(131)
plt.title('Total Score of Students based on lunch')
sns.histplot(data=df,x='total score',kde=True,hue='lunch')
plt.subplot(132)
plt.title('Total Score of Female Students based on lunch')
sns.histplot(data=df[df.gender=='female'],x='total score',kde=True,hue='lunch')
plt.subplot(133)
plt.title('Total Score of Male Students based on lunch')
sns.histplot(data=df[df.gender=='male'],x='total score',kde=True,hue='lunch')
plt.show()
```

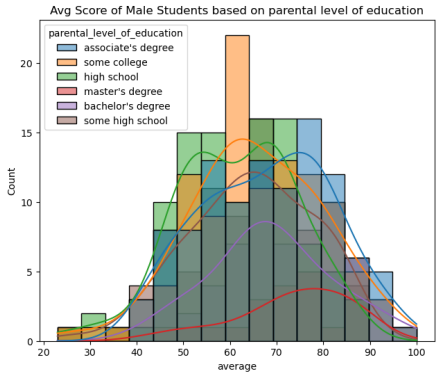


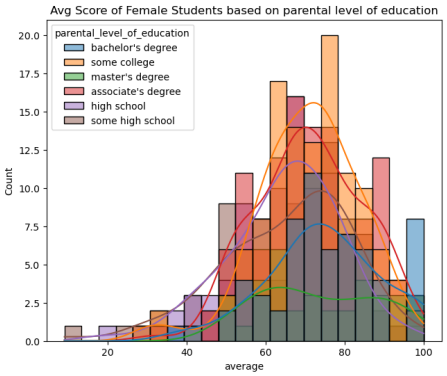




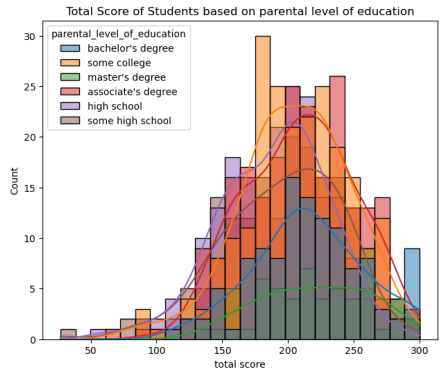
In [48]:
```
plt.subplots(1,3,figsize=(25,6))
plt.subplot(131)
plt.title('Avg Score of Students based on parental level of education')
ax =sns.histplot(data=df,x='average',kde=True,hue='parental_level_of_education')
plt.subplot(132)
plt.title('Avg Score of Male Students based on parental level of education')
ax =sns.histplot(data=df[df.gender=='male'],x='average',kde=True,hue='parental_level_of_educatio
plt.subplot(133)
plt.title('Avg Score of Female Students based on parental level of education')
ax =sns.histplot(data=df[df.gender=='female'],x='average',kde=True,hue='parental_level_of_educat
plt.show()
```

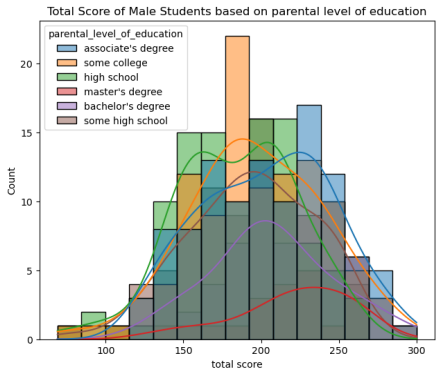


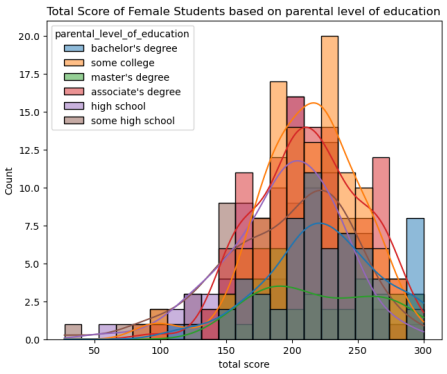




In [49]:
```
plt.subplots(1,3,figsize=(25,6))
plt.subplot(131)
plt.title('Total Score of Students based on parental level of education')
ax =sns.histplot(data=df,x='total score',kde=True,hue='parental_level_of_education')
plt.subplot(132)
plt.title('Total Score of Male Students based on parental level of education')
ax =sns.histplot(data=df[df.gender=='male'],x='total score',kde=True,hue='parental_level_of_educ
plt.subplot(133)
plt.title('Total Score of Female Students based on parental level of education')
ax =sns.histplot(data=df[df.gender=='female'],x='total score',kde=True,hue='parental_level_of_ed
plt.show()
```

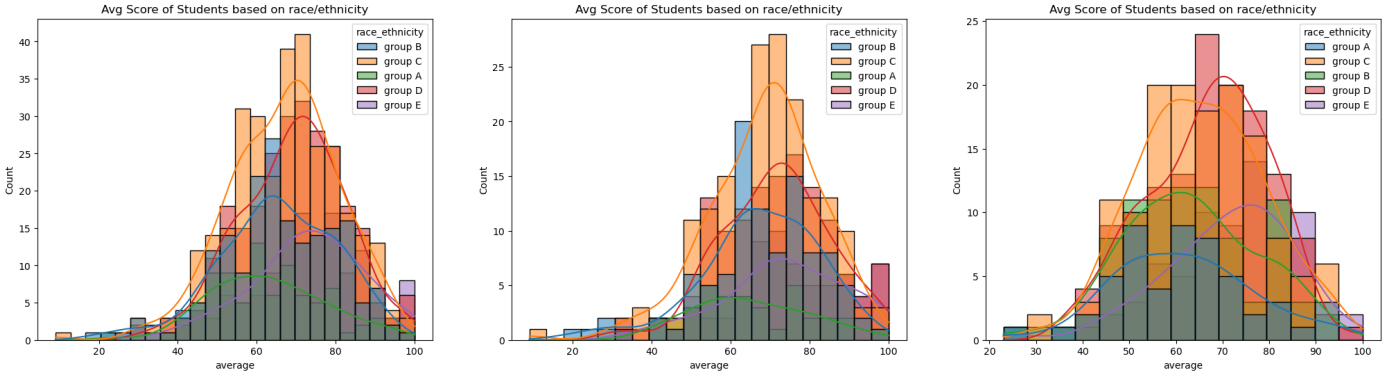






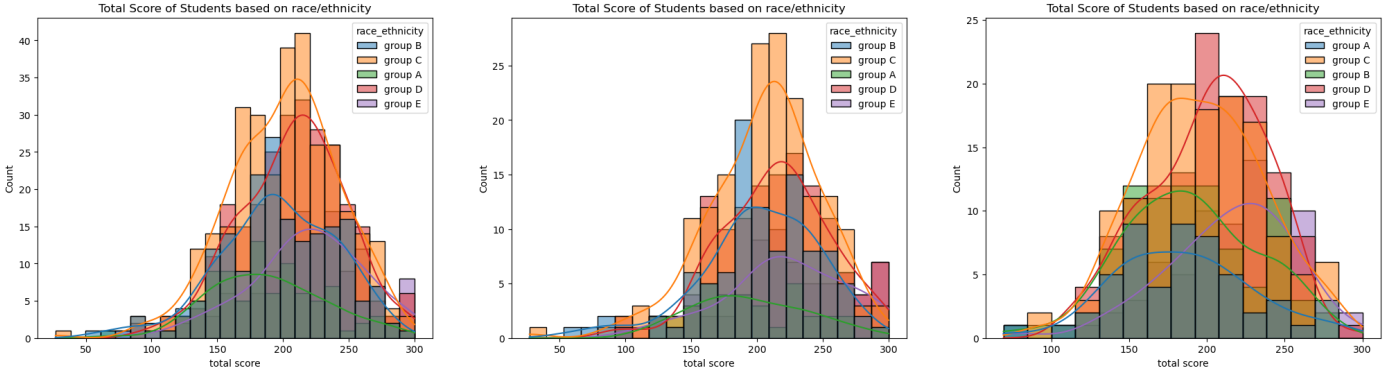
In [51]:
```
plt.subplots(1,3,figsize=(25,6))
plt.subplot(131)
plt.title('Avg Score of Students based on race/ethnicity')
```

```
ax =sns.histplot(data=df,x='average',kde=True,hue='race_ethnicity')
plt.subplot(132)
plt.title('Avg Score of Students based on race/ethnicity')
ax =sns.histplot(data=df[df.gender=='female'],x='average',kde=True,hue='race_ethnicity')
plt.subplot(133)
plt.title('Avg Score of Students based on race/ethnicity')
ax =sns.histplot(data=df[df.gender=='male'],x='average',kde=True,hue='race_ethnicity')
plt.show()
```



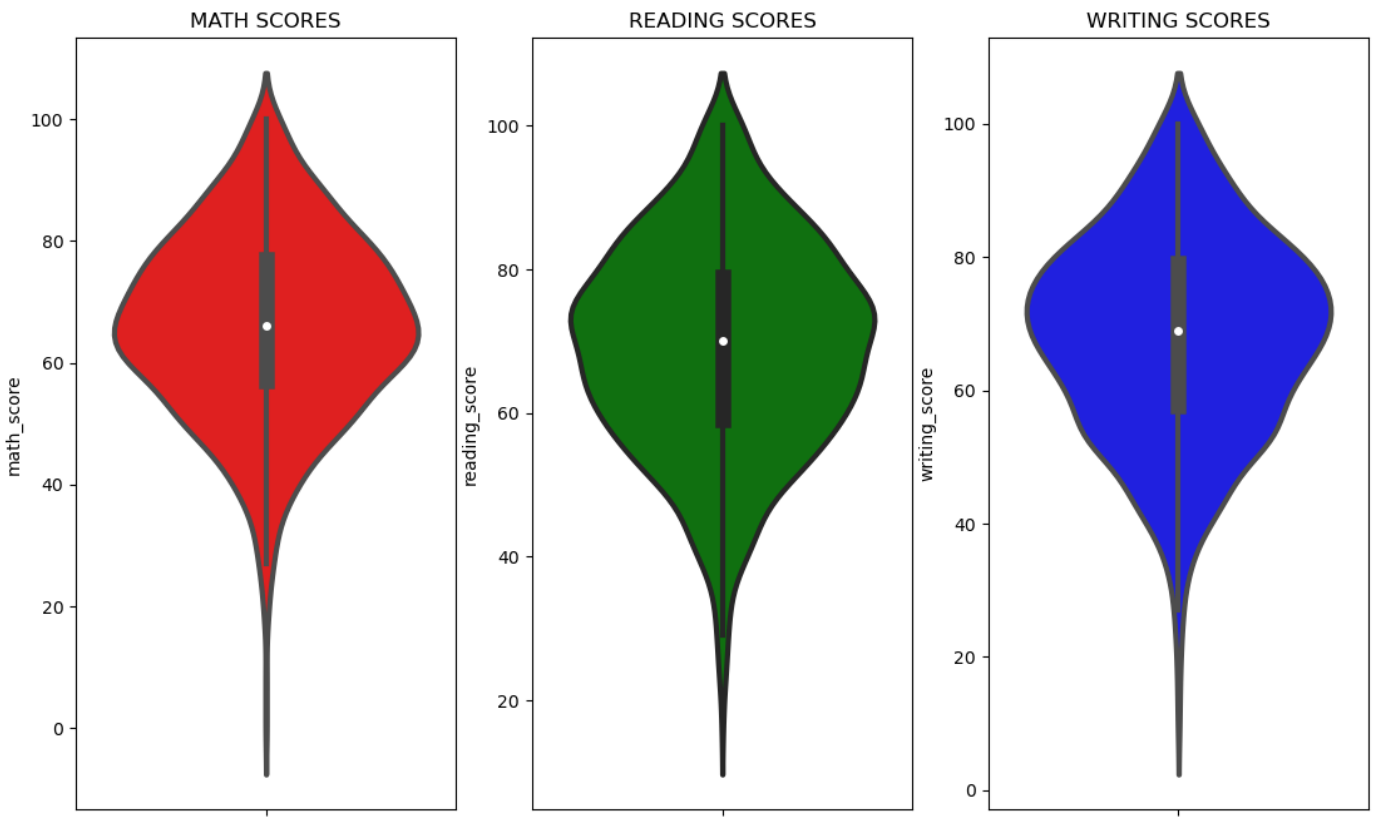
In [52]:

```
plt.subplots(1,3,figsize=(25,6))
plt.subplot(131)
plt.title('Total Score of Students based on race/ethnicity')
ax =sns.histplot(data=df,x='total score',kde=True,hue='race_ethnicity')
plt.subplot(132)
plt.title('Total Score of Students based on race/ethnicity')
ax =sns.histplot(data=df[df.gender=='female'],x='total score',kde=True,hue='race_ethnicity')
plt.subplot(133)
plt.title('Total Score of Students based on race/ethnicity')
ax =sns.histplot(data=df[df.gender=='male'],x='total score',kde=True,hue='race_ethnicity')
plt.show()
```



In [54]:

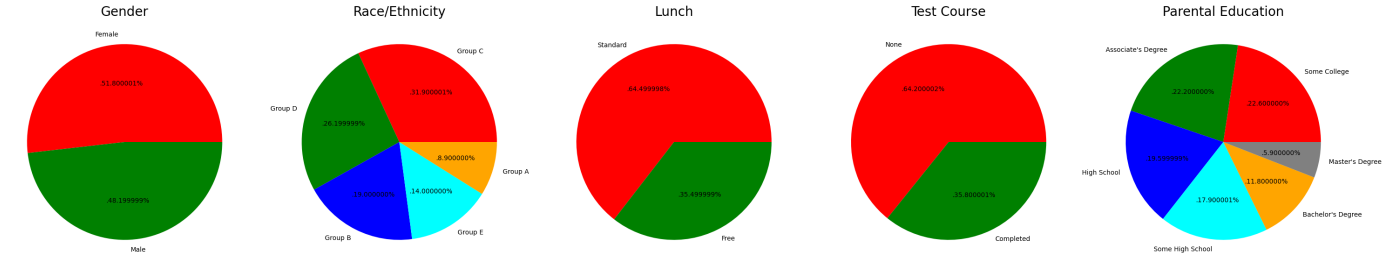
```
plt.figure(figsize=(18,8))
plt.subplot(1, 4, 1)
plt.title('MATH SCORES')
sns.violinplot(y='math_score',data=df,color='red',linewidth=3)
plt.subplot(1, 4, 2)
plt.title('READING SCORES')
sns.violinplot(y='reading_score',data=df,color='green',linewidth=3)
plt.subplot(1, 4, 3)
plt.title('WRITING SCORES')
sns.violinplot(y='writing_score',data=df,color='blue',linewidth=3)
plt.show()
```



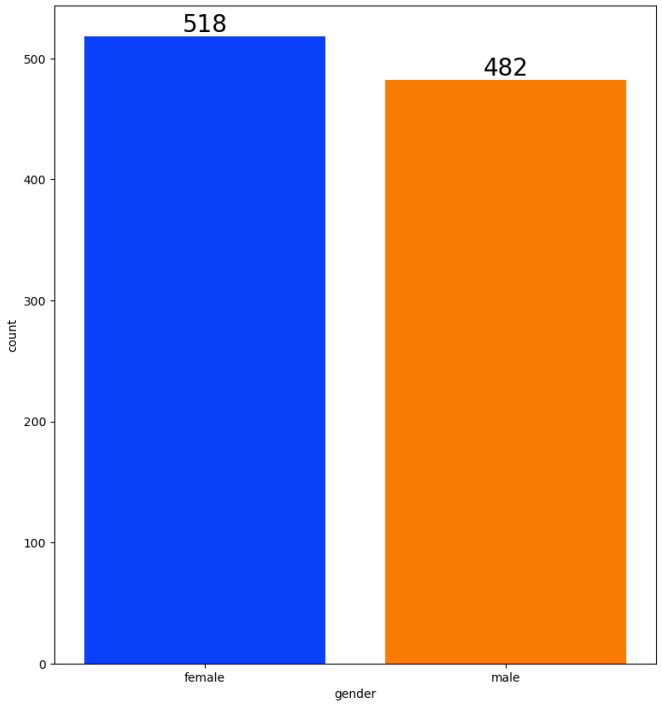
In [55]:

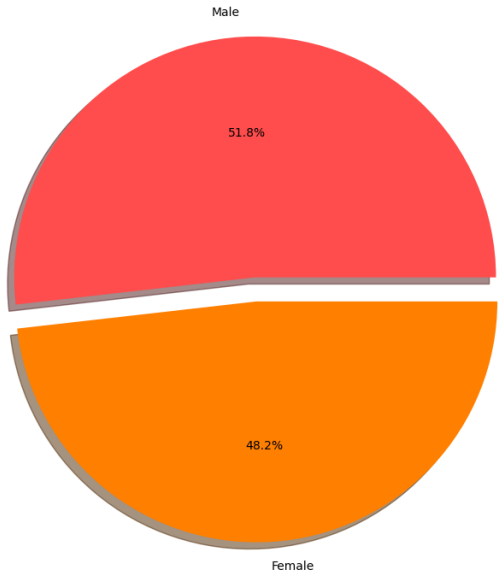
```
plt.rcParams['figure.figsize'] = (30, 12)

plt.subplot(1, 5, 1)
size = df['gender'].value_counts()
labels = 'Female', 'Male'
color = ['red','green']


plt.pie(size, colors = color, labels = labels,autopct = '.%2f%%')
plt.title('Gender', fontsize = 20)
plt.axis('off')



plt.subplot(1, 5, 2)
size = df['race_ethnicity'].value_counts()
labels = 'Group C', 'Group D','Group B','Group E','Group A'
color = ['red', 'green', 'blue', 'cyan','orange']

plt.pie(size, colors = color,labels = labels,autopct = '.%2f%%')
plt.title('Race/Ethnicity', fontsize = 20)
plt.axis('off')



plt.subplot(1, 5, 3)
size = df['lunch'].value_counts()
labels = 'Standard', 'Free'
color = ['red','green']

plt.pie(size, colors = color,labels = labels,autopct = '.%2f%%')
plt.title('Lunch', fontsize = 20)
plt.axis('off')


plt.subplot(1, 5, 4)
size = df['test_preparation_course'].value_counts()
labels = 'None', 'Completed'
```

```
color = ['red','green']

plt.pie(size, colors = color,labels = labels,autopct = '.%2f%%')
plt.title('Test Course', fontsize = 20)
plt.axis('off')


plt.subplot(1, 5, 5)
size = df['parental_level_of_education'].value_counts()
labels = 'Some College', "Associate's Degree",'High School','Some High School',"Bachelor's Degre
color = ['red', 'green', 'blue', 'cyan','orange','grey']

plt.pie(size, colors = color,labels = labels,autopct = '.%2f%%')
plt.title('Parental Education', fontsize = 20)
plt.axis('off')


plt.tight_layout()
plt.grid()

plt.show()
```



In [56]:
```
f,ax=plt.subplots(1,2,figsize=(20,10))
sns.countplot(x=df['gender'],data=df,palette ='bright',ax=ax[0],saturation=0.95)
for container in ax[0].containers:
    ax[0].bar_label(container,color='black',size=20)

plt.pie(x=df['gender'].value_counts(),labels=['Male','Female'],explode=[0,0.1],autopct='%1.1f%%'
plt.show()
```





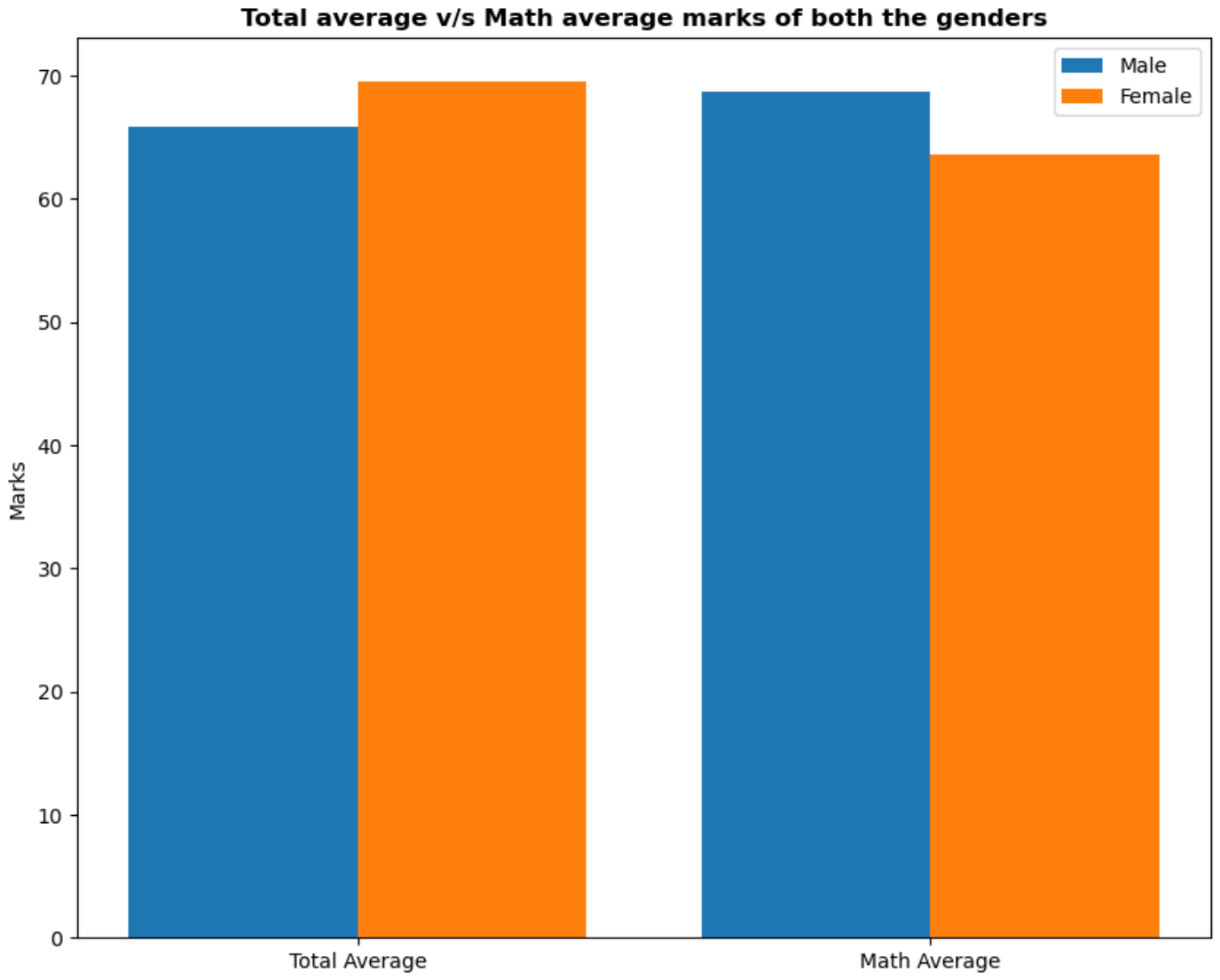
In [80]:
```
gender_group = df.groupby('gender')[['math_score', 'reading_score', 'writing_score','average','t
gender_group
```

Out[80]:

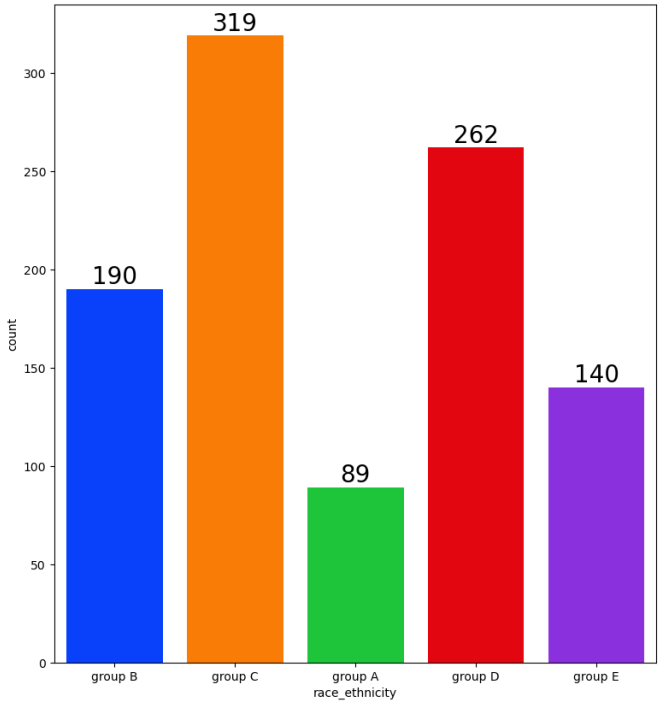
gender	math_score	reading_score	writing_score	average	total score
female	63.633205	72.608108	72.467181	69.569498	208.708494
male	68.728216	65.473029	63.311203	65.837483	197.512448

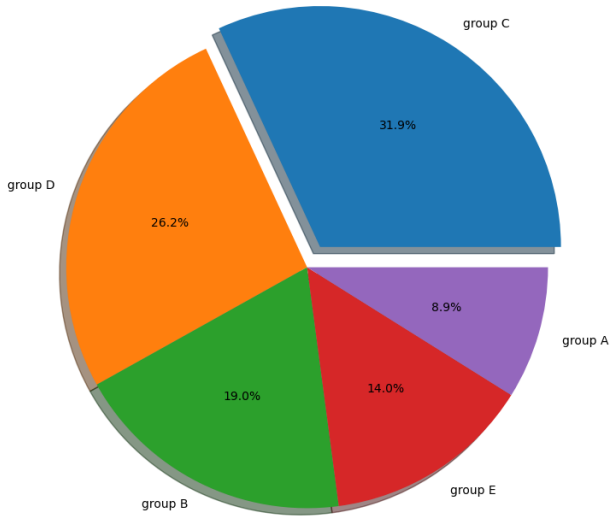
In [82]:

```
plt.figure(figsize=(10, 8))

X = ['Total Average','Math Average']


female_scores = [gender_group['average'][0], gender_group['math_score'][0]]
male_scores = [gender_group['average'][1], gender_group['math_score'][1]]

X_axis = np.arange(len(X))

plt.bar(X_axis - 0.2, male_scores, 0.4, label = 'Male')
plt.bar(X_axis + 0.2, female_scores, 0.4, label = 'Female')

plt.xticks(X_axis, X)
plt.ylabel("Marks")
plt.title("Total average v/s Math average marks of both the genders", fontweight='bold')
plt.legend()
plt.show()
```



In [83]:
```
f,ax=plt.subplots(1,2,figsize=(20,10))
sns.countplot(x=df['race_ethnicity'],data=df,palette = 'bright',ax=ax[0],saturation=0.95)
for container in ax[0].containers:
    ax[0].bar_label(container,color='black',size=20)

plt.pie(x = df['race_ethnicity'].value_counts(),labels=df['race_ethnicity'].value_counts().index
plt.show()
```

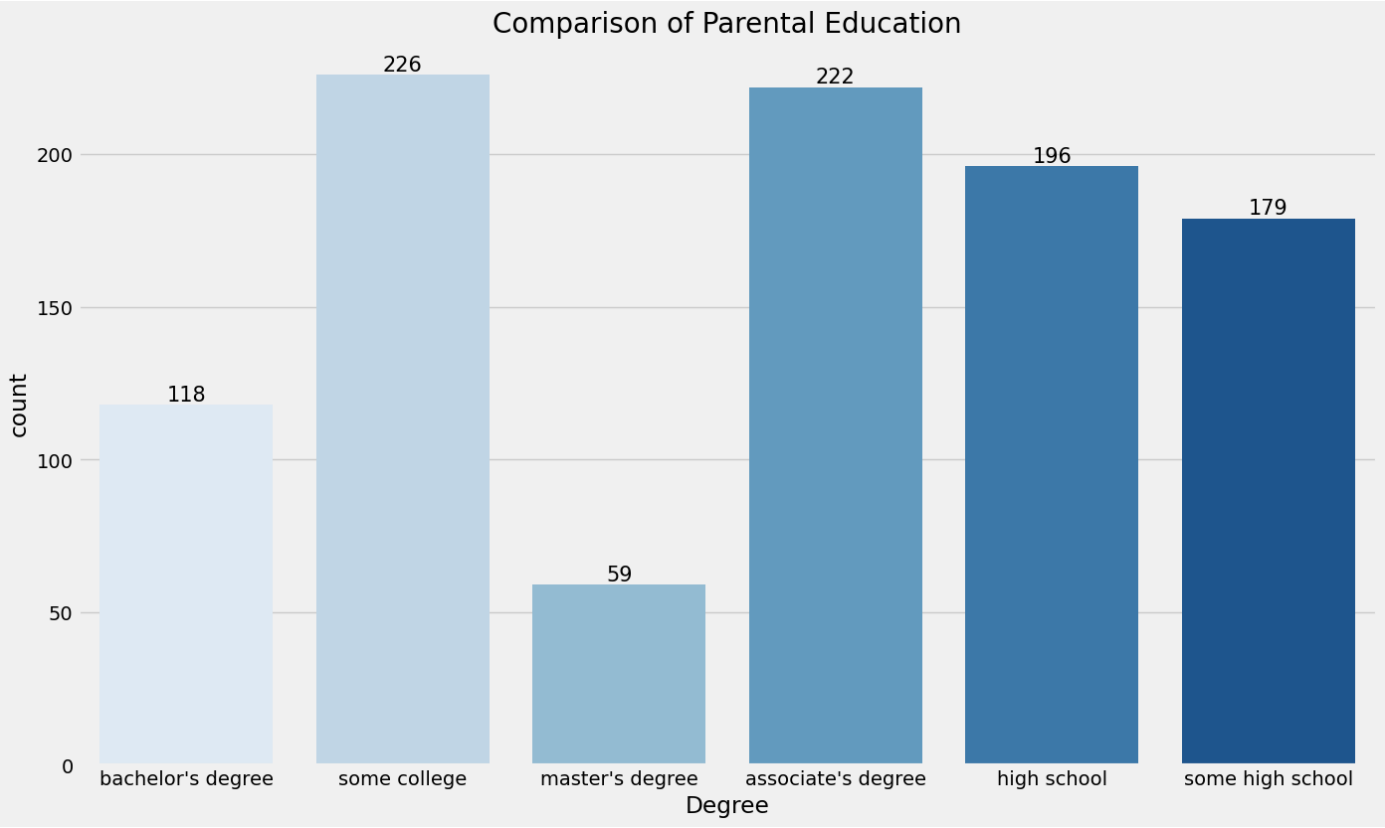




In [86]:
```
group_data2=df.groupby('race_ethnicity')
f,ax=plt.subplots(1,3,figsize=(20,8))
sns.barplot(x=group_data2['math_score'].mean().index,y=group_data2['math_score'].mean().values,p
ax[0].set_title('Math score',color='#005ce6',size=20)

for container in ax[0].containers:
    ax[0].bar_label(container,color='black',size=15)

sns.barplot(x=group_data2['reading_score'].mean().index,y=group_data2['reading_score'].mean().va
ax[1].set_title('Reading score',color='#005ce6',size=20)

for container in ax[1].containers:
    ax[1].bar_label(container,color='black',size=15)

sns.barplot(x=group_data2['writing_score'].mean().index,y=group_data2['writing_score'].mean().va
ax[2].set_title('Writing score',color='#005ce6',size=20)

for container in ax[2].containers:
    ax[2].bar_label(container,color='black',size=15)
```

In [107…

```
plt.rcParams['figure.figsize'] = (15, 9)
plt.style.use('fivethirtyeight')
ax=sns.countplot(x='parental_level_of_education',data =df , palette = 'Blues')
for container in ax.containers:
    ax.bar_label(container,color='black',size=15)
plt.title('Comparison of Parental Education', fontweight = 30, fontsize = 20)
plt.xlabel('Degree')
plt.ylabel('count')
plt.show()
```

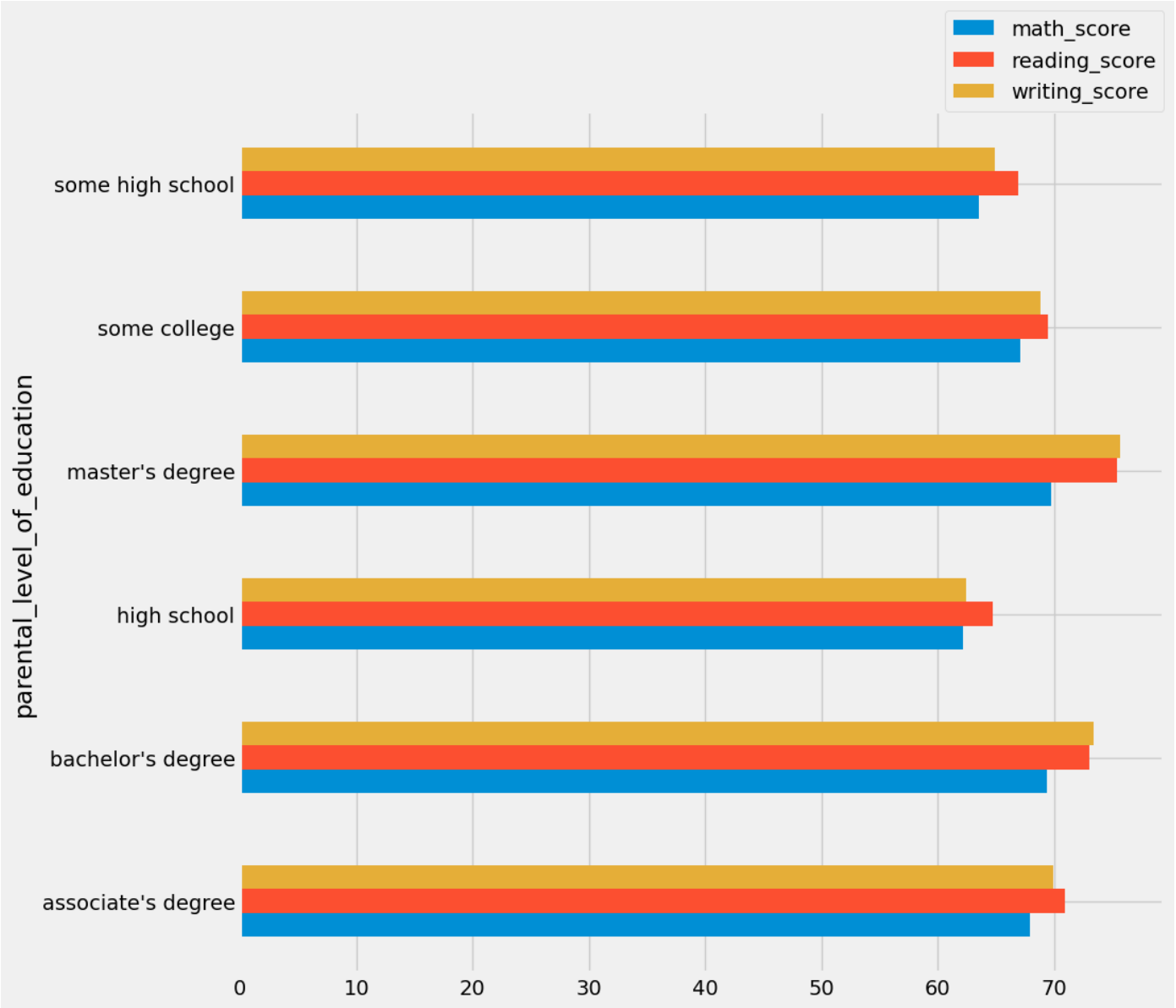


In [91]:

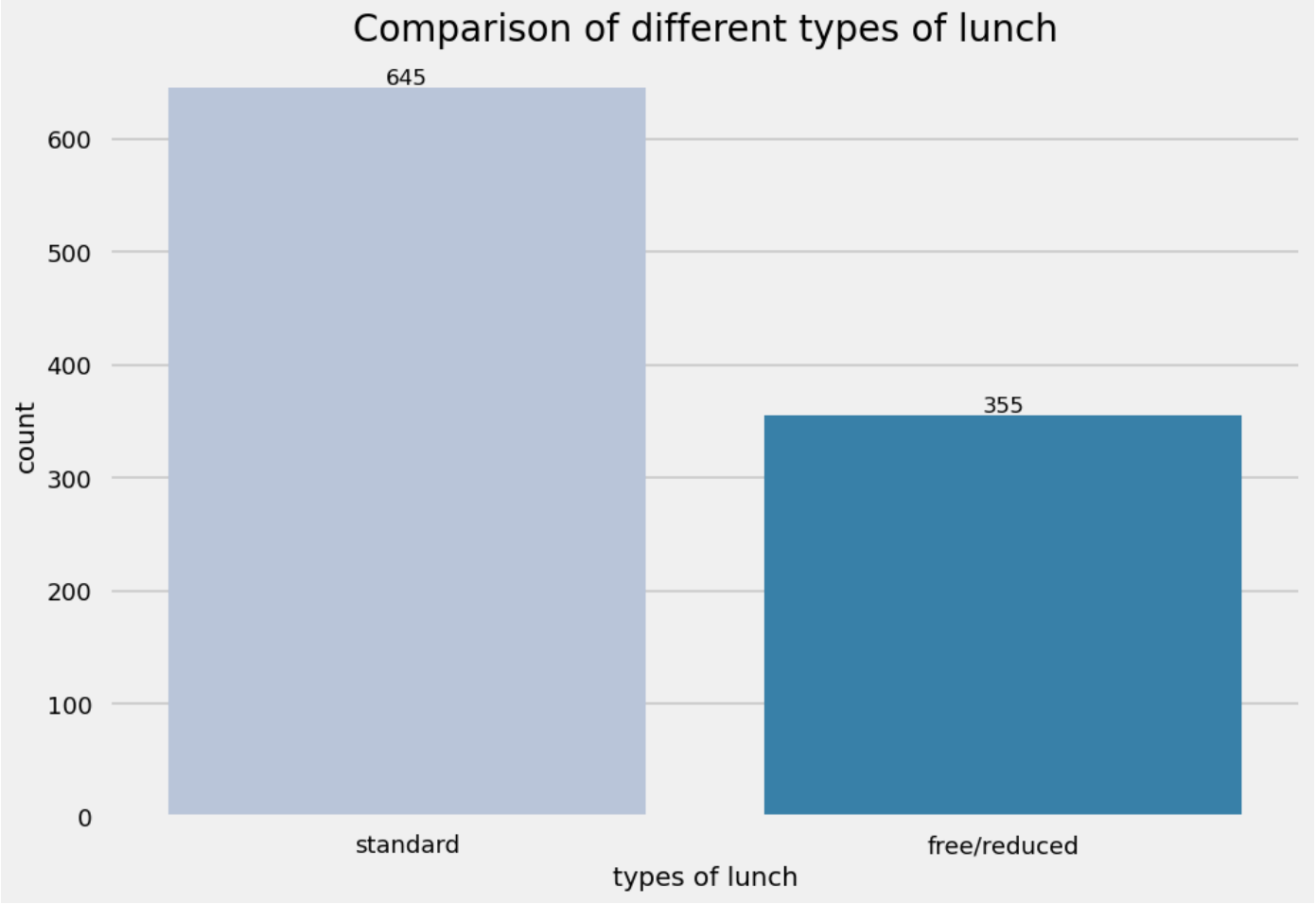
```
df['parental_level_of_education']
```

Out[91]:
```
parental_level_of_education
some college          226
associate's degree    222
high school           196
some high school      179
bachelor's degree     118
master's degree        59
Name: count, dtype: int64
```

In [120…
```
df.groupby('parental_level_of_education')[['math_score', 'reading_score', 'writing_score']].mean
plt.legend(bbox_to_anchor=(1, 1), loc=4, borderaxespad=0.)
plt.show()
```

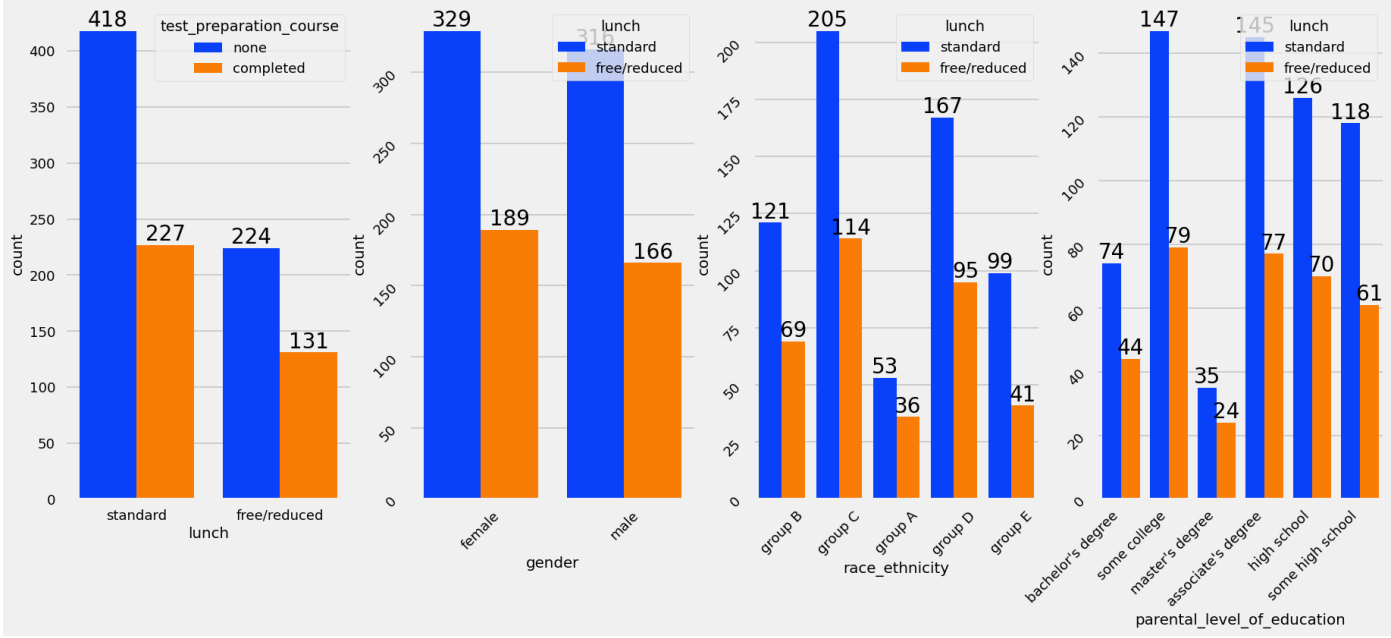


In [127…
```
plt.rcParams['figure.figsize'] = (15, 9)
plt.style.use('seaborn-v0_8-talk')
ax=sns.countplot(x='lunch',data=df, palette = 'PuBu')
for container in ax.containers:
    ax.bar_label(container,color='black',size=12)
plt.title('Comparison of different types of lunch', fontweight = 30, fontsize = 20)
plt.xlabel('types of lunch')
plt.ylabel('count')
plt.show()
```



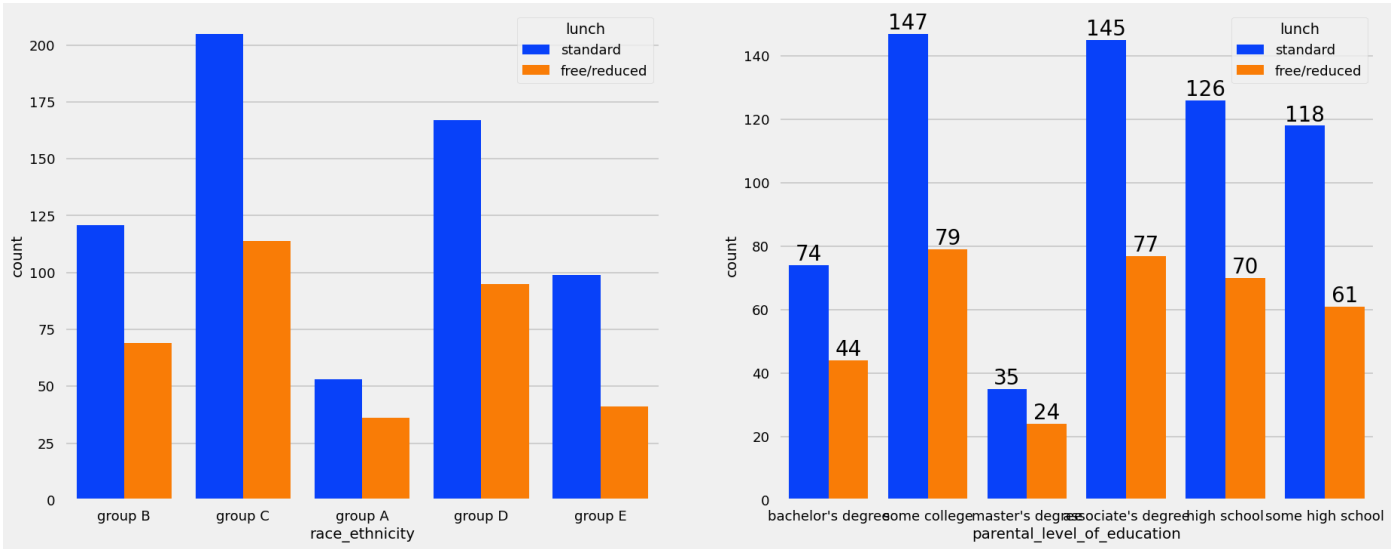
In [178…

```
f,ax=plt.subplots(1,4,figsize=(20,8))

ax0=sns.countplot(x=df['lunch'],data=df,palette = 'bright',hue='test_preparation_course',saturat
plt.xticks(rotation=45,ha='right')
for container in ax[0].containers:
    ax[0].bar_label(container,color='black',size=20)

ax1=sns.countplot(x=df['gender'],data=df,palette = 'bright',hue='lunch',saturation=0.95,ax=ax[1]
ax1.tick_params(labelrotation=45)
for container in ax[1].containers:
    ax[1].bar_label(container,color='black',size=20)

ax2= sns.countplot(x=df['race_ethnicity'],data=df,palette = 'bright',hue='lunch',saturation=0.95
ax2.tick_params(labelrotation=45)

for container in ax[2].containers:
    ax[2].bar_label(container,color='black',size=20)

ax3=sns.countplot(x=df['parental_level_of_education'],data=df,palette = 'bright',hue='lunch',sat
ax3.tick_params(labelrotation=45)
for container in ax[3].containers:
    ax[3].bar_label(container,color='black',size=20)
```



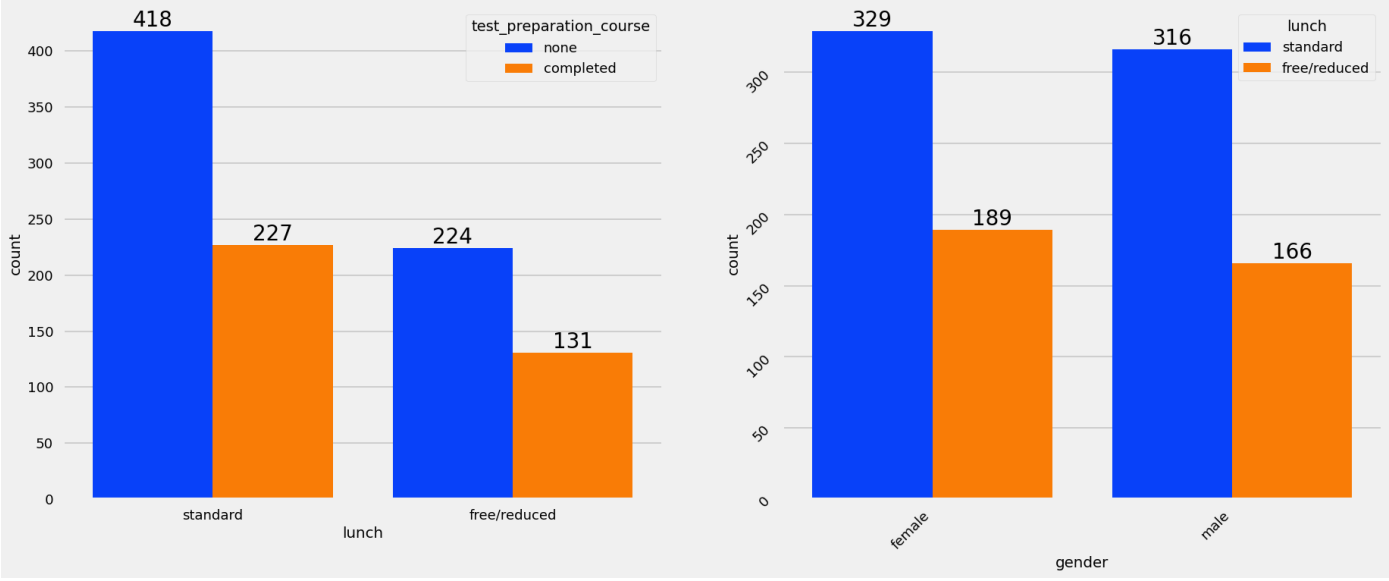
In [164…

```
f,ax=plt.subplots(1,2,figsize=(20,8))

sns.countplot(x=df['race_ethnicity'],data=df,palette = 'bright',hue='lunch',saturation=0.95,ax=a
for container in ax[1].containers:
    ax[0].bar_label(container,color='black',size=20)

sns.countplot(x=df['parental_level_of_education'],data=df,palette = 'bright',hue='lunch',saturat
for container in ax[1].containers:
    ax[1].bar_label(container,color='black',size=20)
```



In [179…

```
f,ax=plt.subplots(1,2,figsize=(20,8))
ax0=sns.countplot(x=df['lunch'],data=df,palette = 'bright',hue='test_preparation_course',saturat
plt.xticks(rotation=45,ha='right')
for container in ax[0].containers:
    ax[0].bar_label(container,color='black',size=20)

ax1=sns.countplot(x=df['gender'],data=df,palette = 'bright',hue='lunch',saturation=0.95,ax=ax[1]
ax1.tick_params(labelrotation=45)
for container in ax[1].containers:
    ax[1].bar_label(container,color='black',size=20)
```

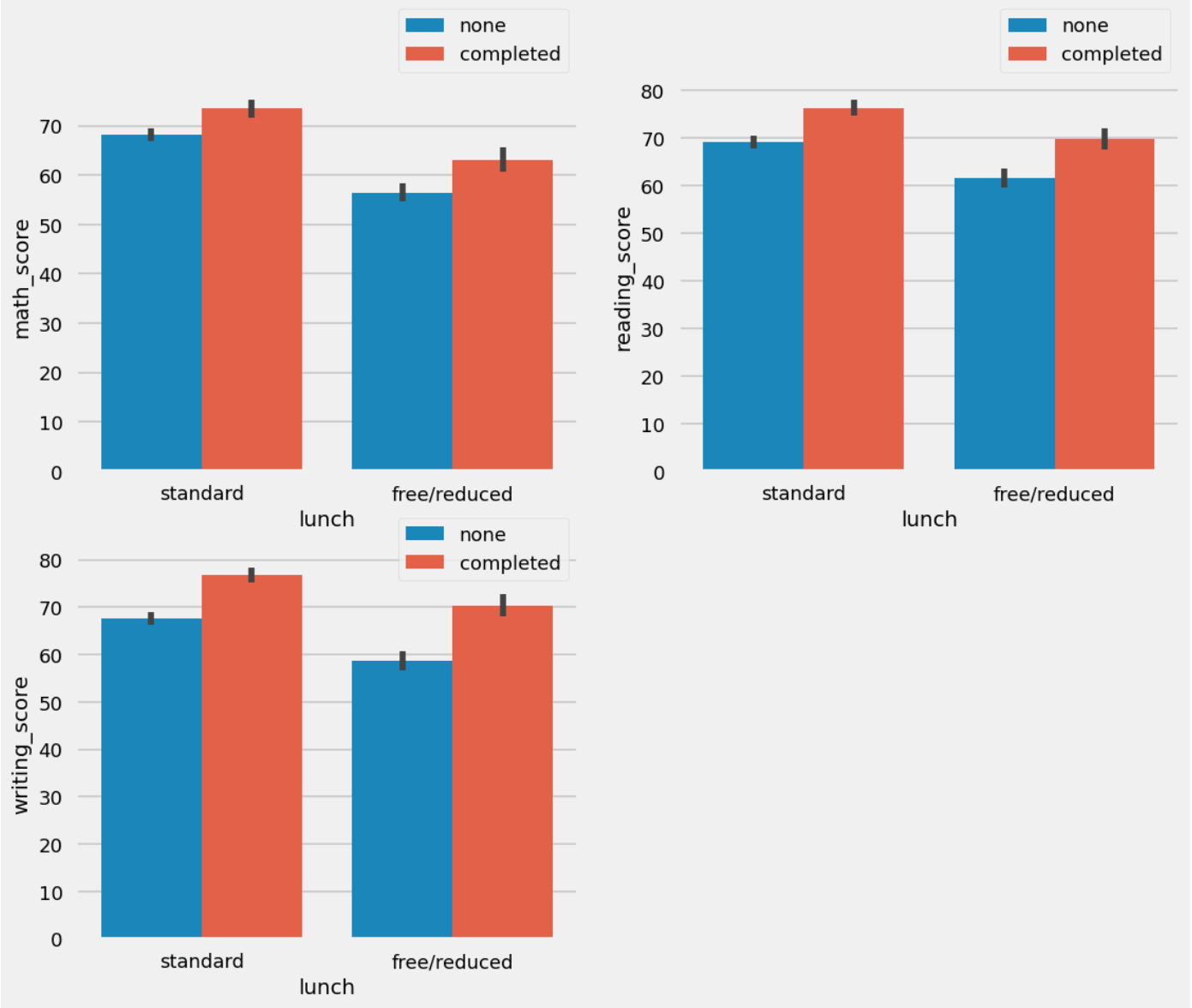


In [195…

```
plt.figure(figsize=(12,10))
plt.subplot(2,2,1)
sns.barplot (x=df['lunch'], y=df['math_score'], hue=df['test_preparation_course'])
plt.legend(bbox_to_anchor=(1,1),loc=4)
#plt.legend(bbox_to_anchor=(1, 1), loc=4, borderaxespad=0.)
plt.subplot(2,2,2)
sns.barplot (x=df['lunch'], y=df['reading_score'], hue=df['test_preparation_course'])
plt.legend(bbox_to_anchor=(1,1),loc=4)
plt.subplot(2,2,3)
sns.barplot (x=df['lunch'], y=df['writing_score'], hue=df['test_preparation_course'])
plt.legend(bbox_to_anchor=(1,1),loc=7)
```

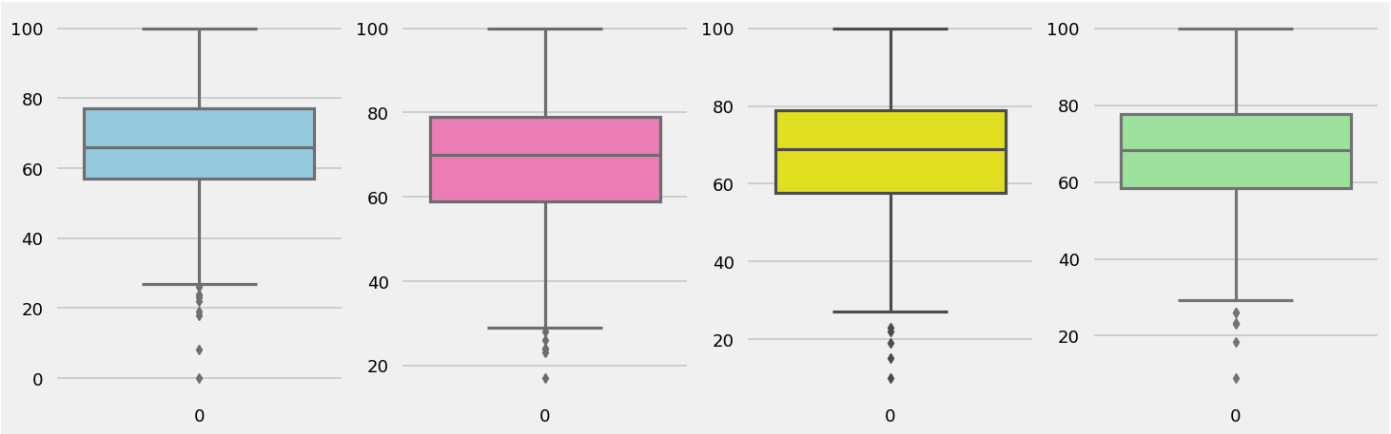
Out[195]:

```
<matplotlib.legend.Legend at 0x27ee33d2050>
```



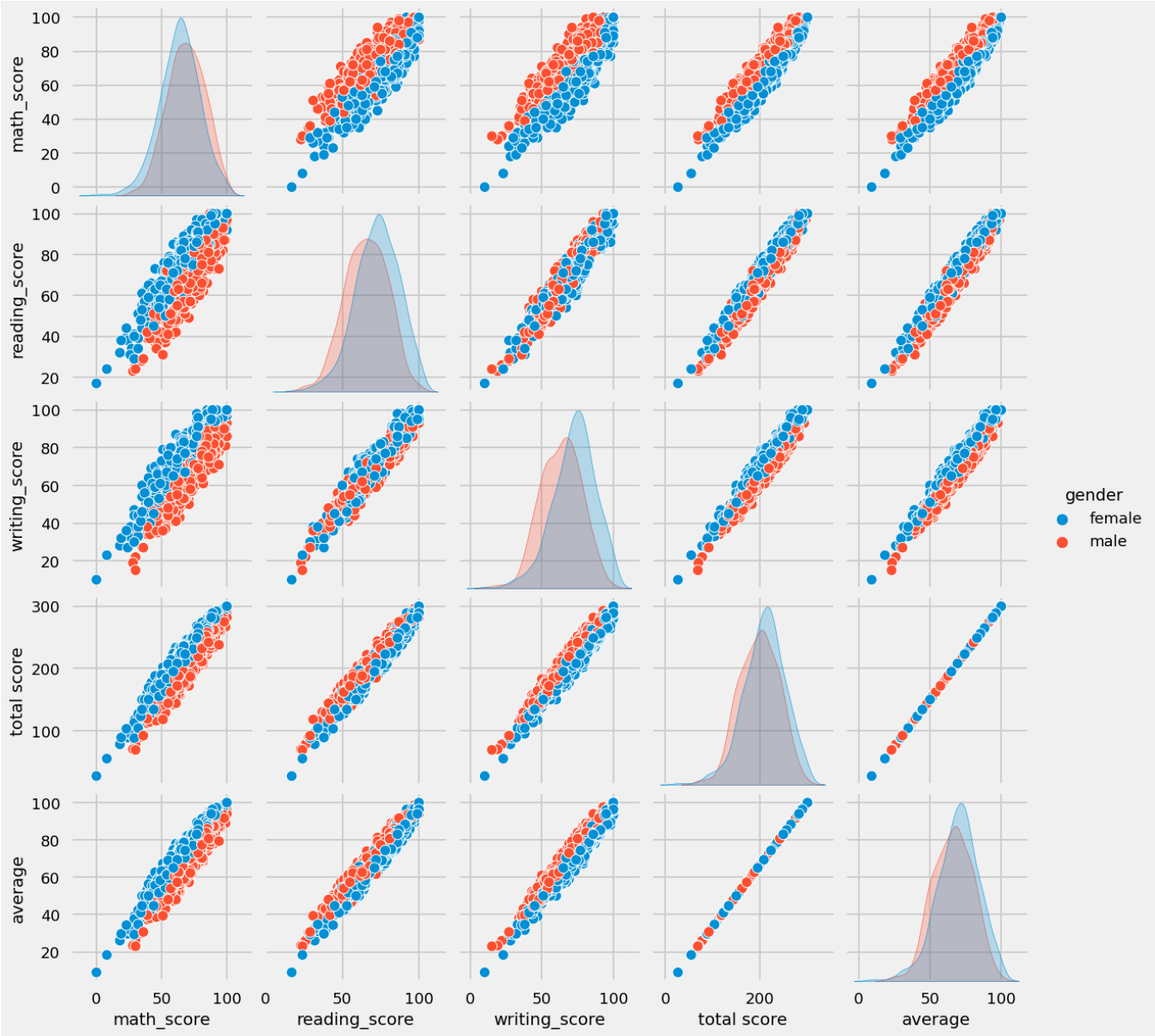
In [196…

```
plt.subplots(1,4,figsize=(16,5))
plt.subplot(141)
sns.boxplot(df['math_score'],color='skyblue')
plt.subplot(142)
sns.boxplot(df['reading_score'],color='hotpink')
plt.subplot(143)
sns.boxplot(df['writing_score'],color='yellow')
plt.subplot(144)
sns.boxplot(df['average'],color='lightgreen')
plt.show()
```



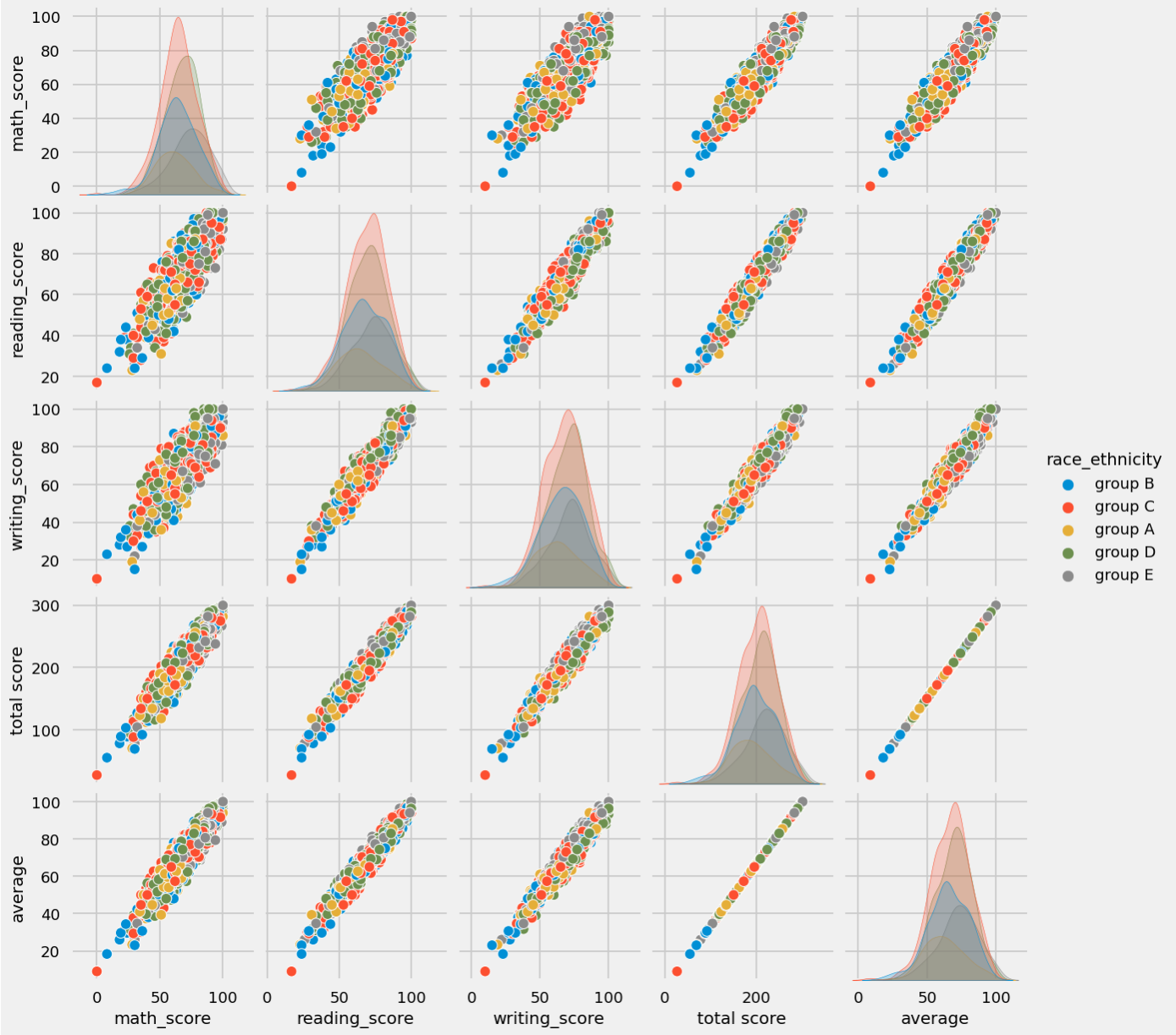
In [197…

```
sns.pairplot(df,hue = 'gender')
plt.show()
```



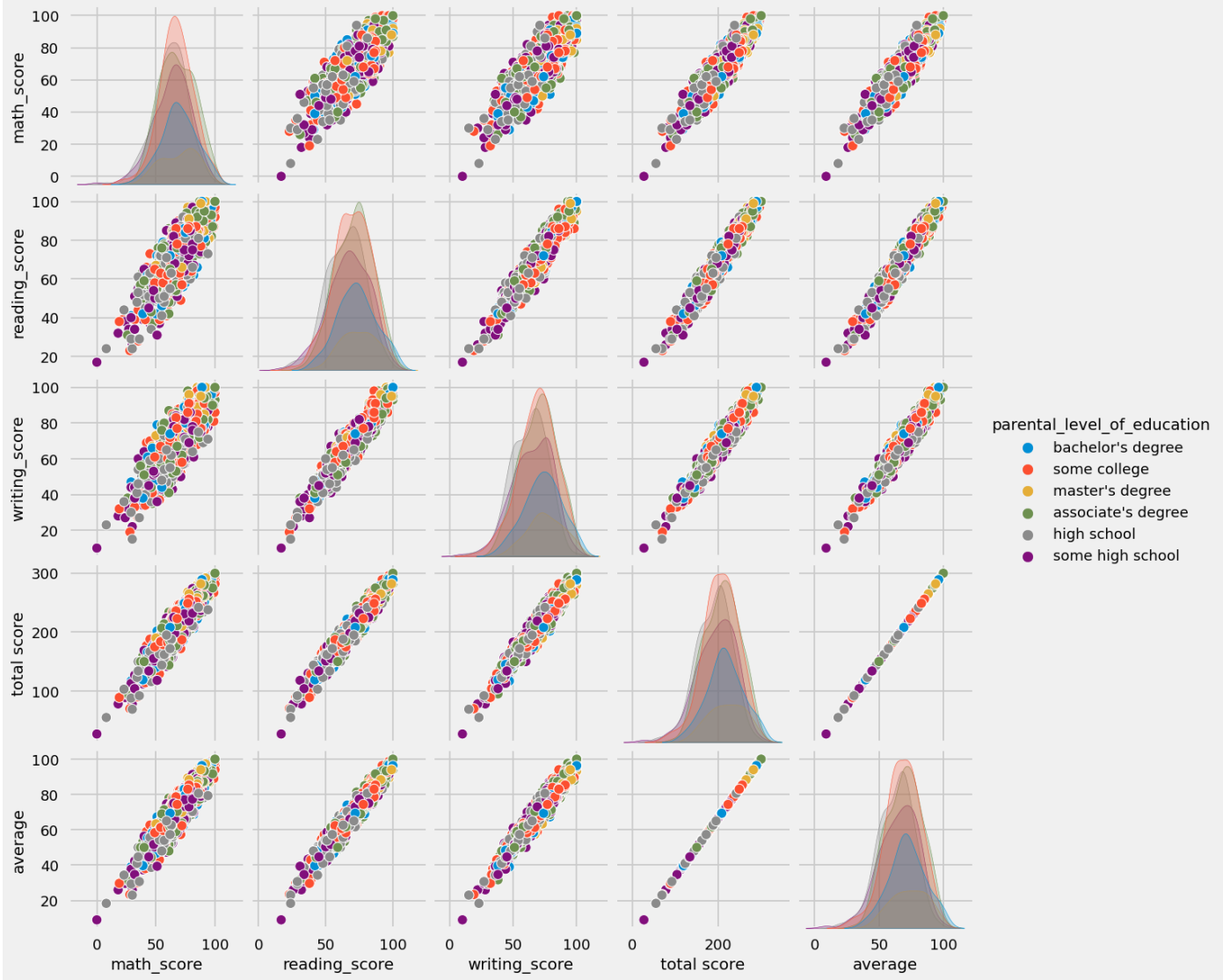
In [199…

```
sns.pairplot(df,hue = 'race_ethnicity')
plt.show()
```



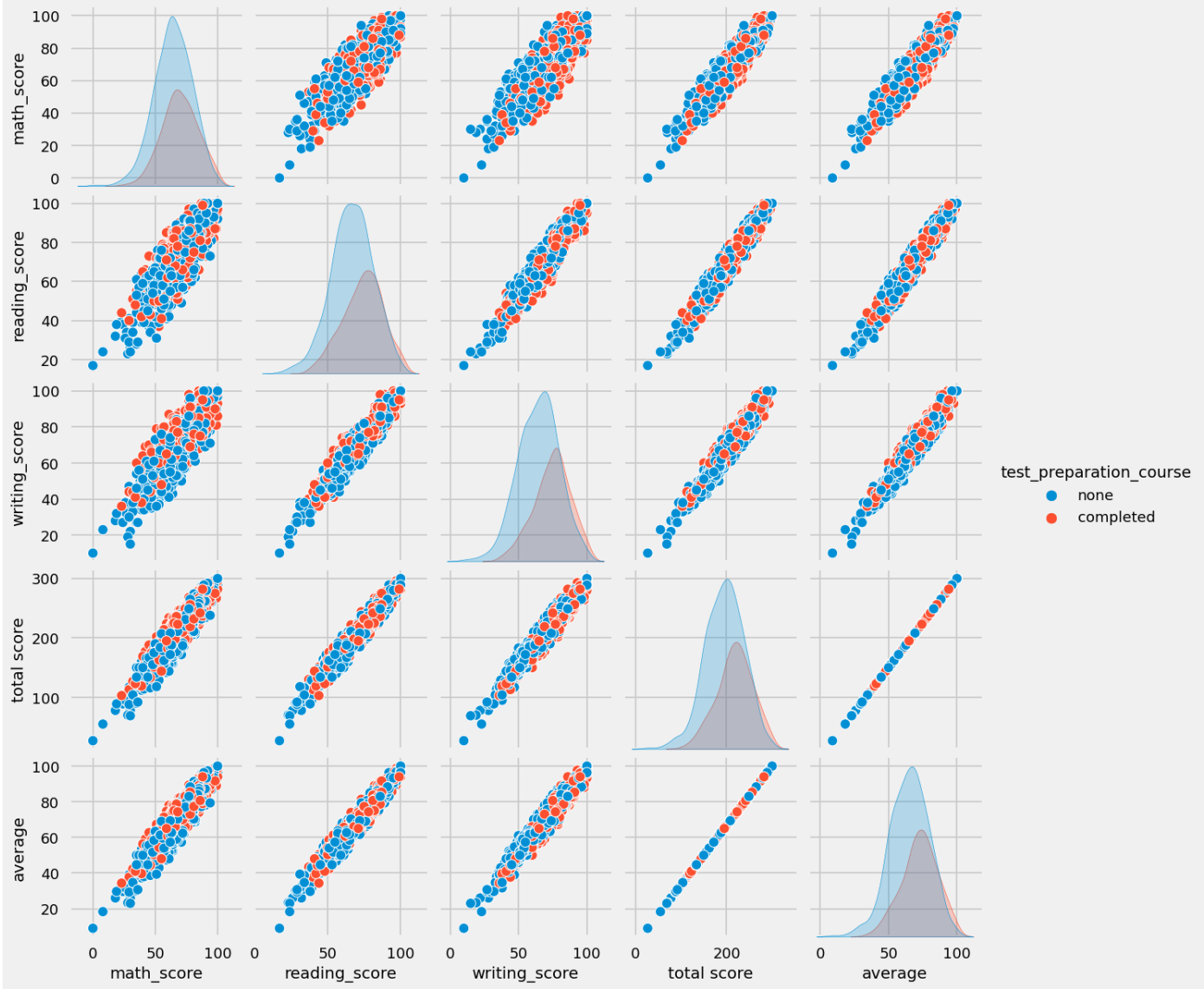
In [200…

```
sns.pairplot(df,hue = 'parental_level_of_education')
plt.show()
```



In [201…

```
sns.pairplot(df,hue = 'test_preparation_course')
plt.show()
```


math_score
reading_score
writing_score
total score
average
test_preparation_course
none
completed

In [ ]: