In [1]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
import warnings
warnings.filterwarnings('ignore')
```

In [2]:
```
df=pd.read_csv('stud.csv')
df.head()
```

Out[2]:

	gender	race_ethnicity	parental_level_of_education	lunch	test_preparation_course	math_s
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	

In [3]:
```
#check missing values

df.isnull().sum()
```

Out[3]:
```
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 [4]:
```
df.isna().sum()
```

Out[4]:
```
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 [5]:
```
#check duplicates
df.duplicated().sum()
```

Out[5]:
```
0
```

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

Out[6]:
```
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 [7]:
```
#check Datatype
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 [8]:
```
#checking the no of each column
df.nunique()
```

Out[8]:
```
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 [9]:
```
#check the statistics of dataset
df.describe()
```

Out[9]:

	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

Insight or observation

from the above description all means are very close to ech other between 66 and 69 all the standard deviation are also close between 14.6-15.9 While there is a minimum of 0 for maths others are having 17 and 10 value

In [10]:
```
#explore more info about the data
df.head()
```

Out[10]:

	gender	race_ethnicity	parental_level_of_education	lunch	test_preparation_course	math_s
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	

In [11]:
```
df.tail()
#last 5 record of data
```

Out[11]:

	gender	race_ethnicity	parental_level_of_education	lunch	test_preparation_course	matl
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	

In [12]:
```
[feature for feature in df.columns]
#segregate numerical and categorical feature
```

Out[12]:
```
['gender',
 'race_ethnicity',
 'parental_level_of_education',
 'lunch',
 'test_preparation_course',
 'math_score',
 'reading_score',
 'writing_score']
```

In [13]:
```
[feature for feature in df.columns if df[feature].dtype!='O']
#numerical feature
```

Out[13]:
```
['math_score', 'reading_score', 'writing_score']
```

In [14]:
```
[feature for feature in df.columns if df[feature].dtype=='O']
#categorical feature
```

Out[14]:
```
['gender',
 'race_ethnicity',
 'parental_level_of_education',
 'lunch',
 'test_preparation_course']
```

In [15]:
```
df['gender'].value_counts()
```

Out[15]:
```
female    518
male      482
Name: gender, dtype: int64
```

In [16]:
```
#Aggregate the total score with mean
df['total_score']=(df['math_score']+df['reading_score']+df['writing_score'])
df['average']=df['total_score']/3
df.head()
```

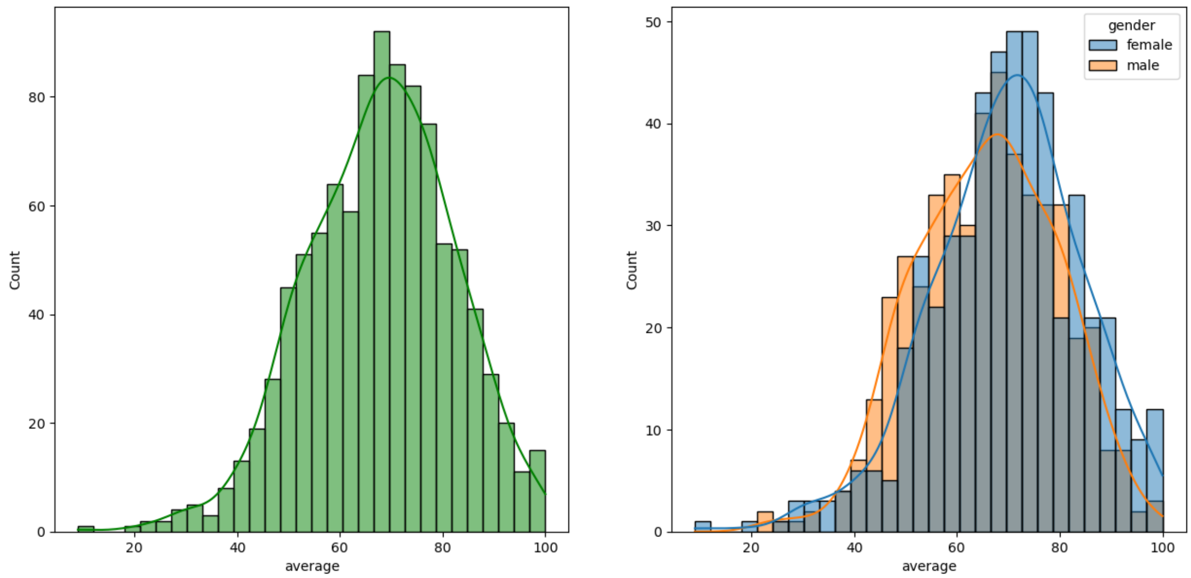
Out[16]:

	gender	race_ethnicity	parental_level_of_education	lunch	test_preparation_course	math_s
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	

In [17]:
```
#explore more visualization
fig,axis=plt.subplots(1,2,figsize=(15,7))
#one row two column in first box
plt.subplot(121)
sns.histplot(data=df,x='average',bins=30,kde=True,color='g')
plt.subplot(122)
```

```
#one row two column in second box
sns.histplot(data=df,x='average',bins=30,kde=True,hue='gender')
```

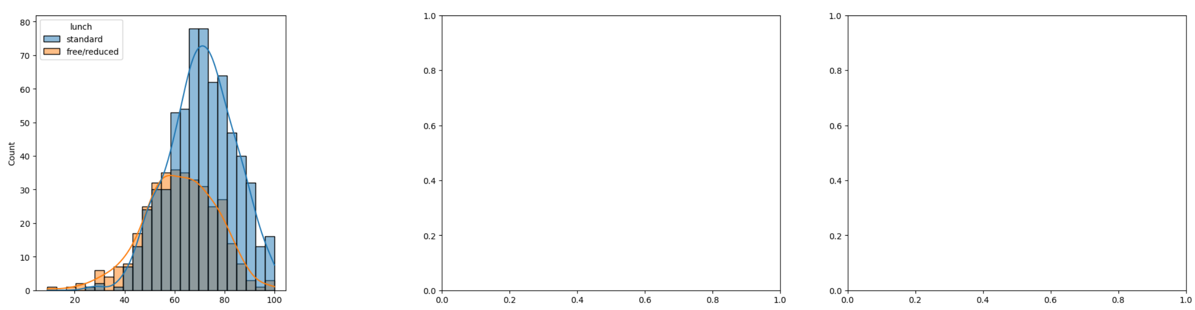
Out[17]: <AxesSubplot: xlabel='average', ylabel='Count'>



Insights:-Female student tend to perform well than male students

In [18]:
```
plt.subplots(1,3,figsize=(25,6))
plt.subplot(141)
sns.histplot(data=df,x='average',kde=True,hue='lunch')
```

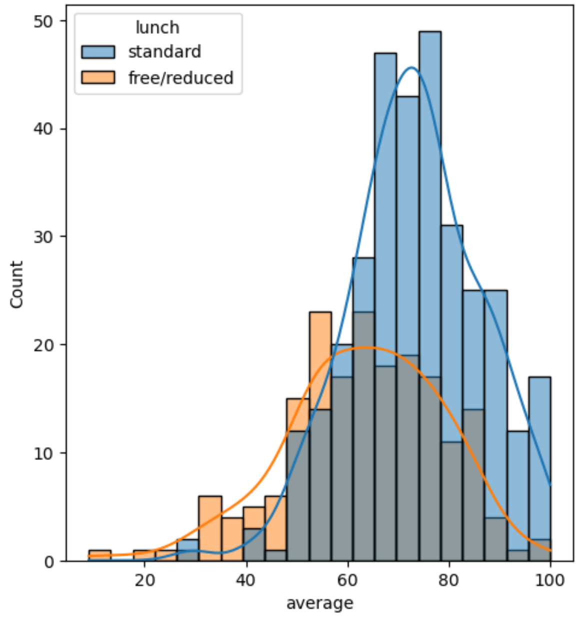
Out[18]: <AxesSubplot: xlabel='average', ylabel='Count'>

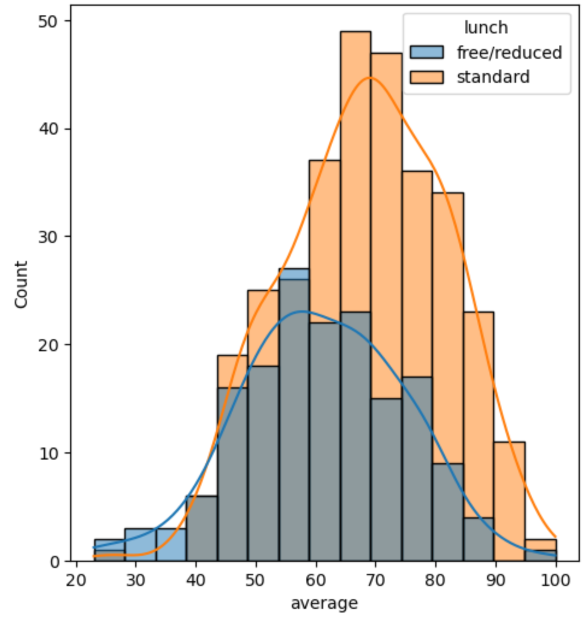


Insight:- #Average of standard lunch student is more than the free or reduced lunch student

In [19]:
```
plt.subplots(1,3,figsize=(25,6))
plt.subplot(142)
sns.histplot(data=df[df.gender=='female'],x='average',kde=True,hue='lunch')
plt.subplot(143)
sns.histplot(data=df[df.gender=='male'],x='average',kde=True,hue='lunch')
```

Out[19]: <AxesSubplot: xlabel='average', ylabel='Count'>





Insight Standard lunch help students perform well in exams standard lunch helps perform well in exams be it a male or female

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

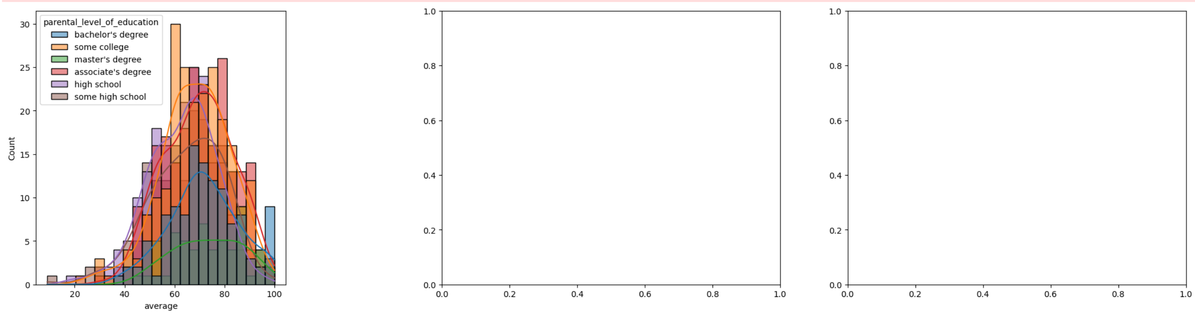
Out[20]:

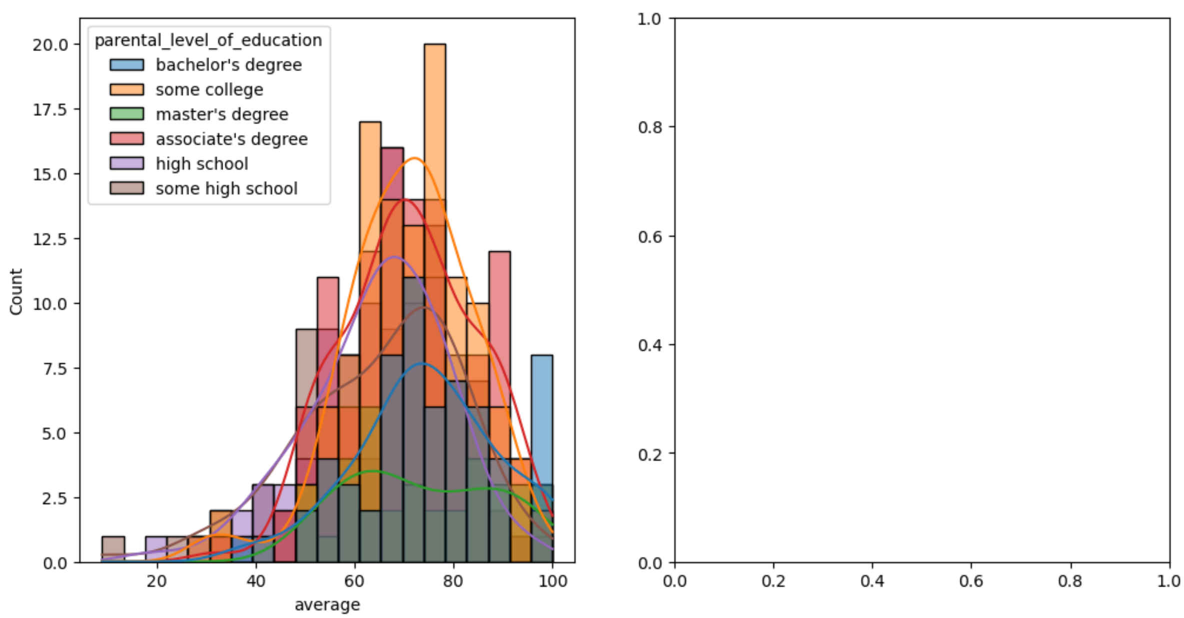
	gender	race_ethnicity	parental_level_of_education	lunch	test_preparation_course	math_s
0	female	group B	bachelor's degree	standard	none	
1	female	group C	some college	standard	completed	
2	female	group B	master's degree	standard	none	
3	male	group A	associate's degree	free/reduced	none	
4	male	group C	some college	standard	none	

In [21]:
```
plt.subplots(1,3,figsize=(25,6))
plt.subplot(141)
sns.histplot(data=df,x='average',kde=True,hue='parental_level_of_education')


plt.subplots(1,3,figsize=(25,6))
plt.subplot(142)
sns.histplot(data=df[df.gender=='female'],x='average',kde=True,hue='parental_level_
plt.subplot(143)
sns.histplot(data=df[df.gender=='male'],x='average',kde=True,hue='parental_level_of
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Cell In[21], line 10
      8 sns.histplot(data=df[df.gender=='female'],x='average',kde=True,hue='parent
al_level_of_education')
      9 plt.subplot(143)
---> 10 sns.histplot(data=df[df.gender=='male'],x='average',kde=True,hue='parental
_level_of_educaton')

File /opt/conda/lib/python3.10/site-packages/seaborn/distributions.py:1395, in his
tplot(data, x, y, hue, weights, stat, bins, binwidth, binrange, discrete, cumulati
ve, common_bins, common_norm, multiple, element, fill, shrink, kde, kde_kws, line_
kws, thresh, pthresh, pmax, cbar, cbar_ax, cbar_kws, palette, hue_order, hue_norm,
color, log_scale, legend, ax, **kwargs)
   1374 def histplot(
   1375     data=None, *,
   1376     # Vector variables
   (...)
   1392     **kwargs,
   1393 ):
-> 1395     p = _DistributionPlotter(
   1396         data=data,
   1397         variables=_DistributionPlotter.get_semantics(locals())
   1398     )
   1400     p.map_hue(palette=palette, order=hue_order, norm=hue_norm)
   1402     if ax is None:

File /opt/conda/lib/python3.10/site-packages/seaborn/distributions.py:113, in _Dis
tributionPlotter.__init__(self, data, variables)
    107 def __init__(
    108     self,
    109     data=None,
    110     variables={},
    111 ):
--> 113     super().__init__(data=data, variables=variables)

File /opt/conda/lib/python3.10/site-packages/seaborn/_oldcore.py:640, in VectorPlo
tter.__init__(self, data, variables)
    635 # var_ordered is relevant only for categorical axis variables, and may
    636 # be better handled by an internal axis information object that tracks
    637 # such information and is set up by the scale_* methods. The analogous
    638 # information for numeric axes would be information about log scales.
    639 self._var_ordered = {"x": False, "y": False}  # alt., used DefaultDict
--> 640 self.assign_variables(data, variables)
    642 for var, cls in self._semantic_mappings.items():
    643 
    644     # Create the mapping function
    645     map_func = partial(cls.map, plotter=self)

File /opt/conda/lib/python3.10/site-packages/seaborn/_oldcore.py:701, in VectorPlo
tter.assign_variables(self, data, variables)
    699 else:
    700     self.input_format = "long"
--> 701     plot_data, variables = self._assign_variables_longform(
    702         data, **variables,
    703     )
```

```
    705 self.plot_data = plot_data
    706 self.variables = variables

File /opt/conda/lib/python3.10/site-packages/seaborn/_oldcore.py:938, in VectorPlo
tter._assign_variables_longform(self, data, **kwargs)
    933 elif isinstance(val, (str, bytes)):
    934
    935     # This looks like a column name but we don't know what it means!
    937     err = f"Could not interpret value `{val}` for parameter `{key}`"
--> 938     raise ValueError(err)
    940 else:
    941
    942     # Otherwise, assume the value is itself data
    943
    944     # Raise when data object is present and a vector can't matched
    945     if isinstance(data, pd.DataFrame) and not isinstance(val, pd.Series):

ValueError: Could not interpret value `parental_level_of_educaton` for parameter `
hue`
```

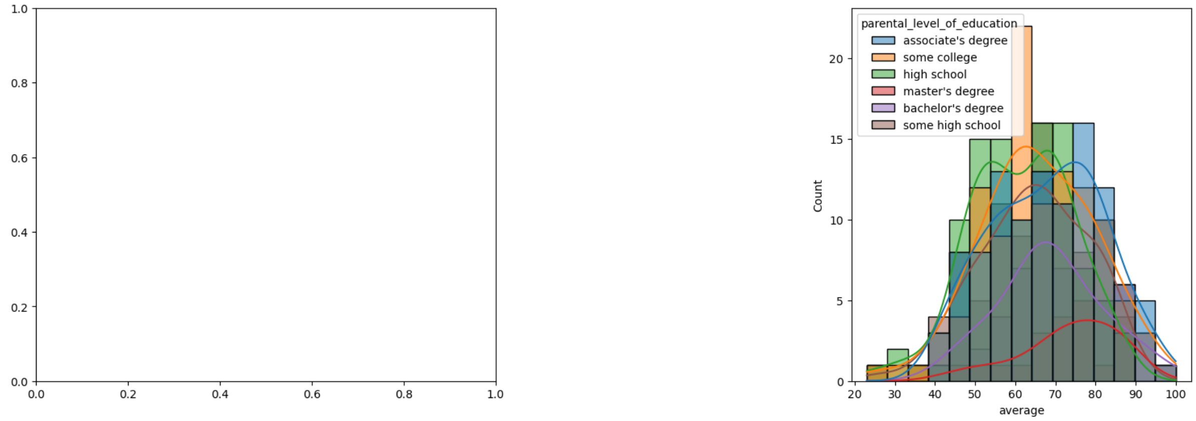




Insight In general parents education dont help student perform well in exams 2nd plot we can see there is no effect of parents education on females students.

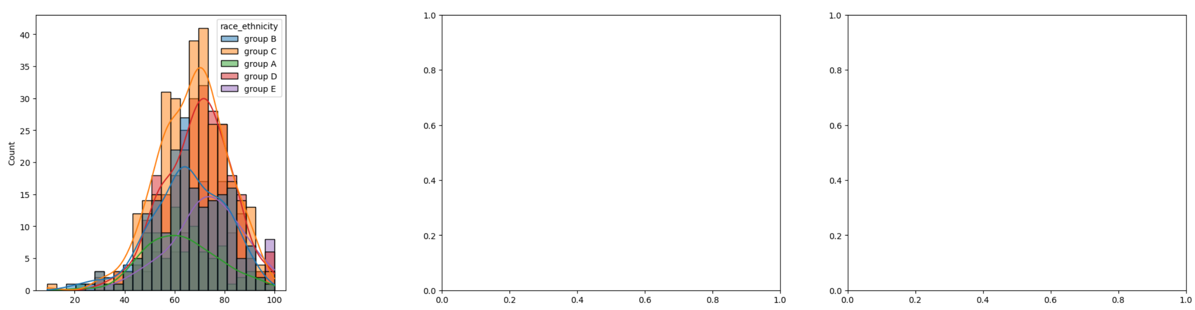
```
In [22]: plt.subplots(1,3,figsize=(25,6))
plt.subplot(143)
sns.histplot(data=df[df.gender=='male'],x='average',kde=True,hue='parental_level_of
```

Out[22]: <AxesSubplot: xlabel='average', ylabel='Count'>



Insight 3rd plot we can see that parents education is of associate's degree or masters degree their male child tend to perform well in exam

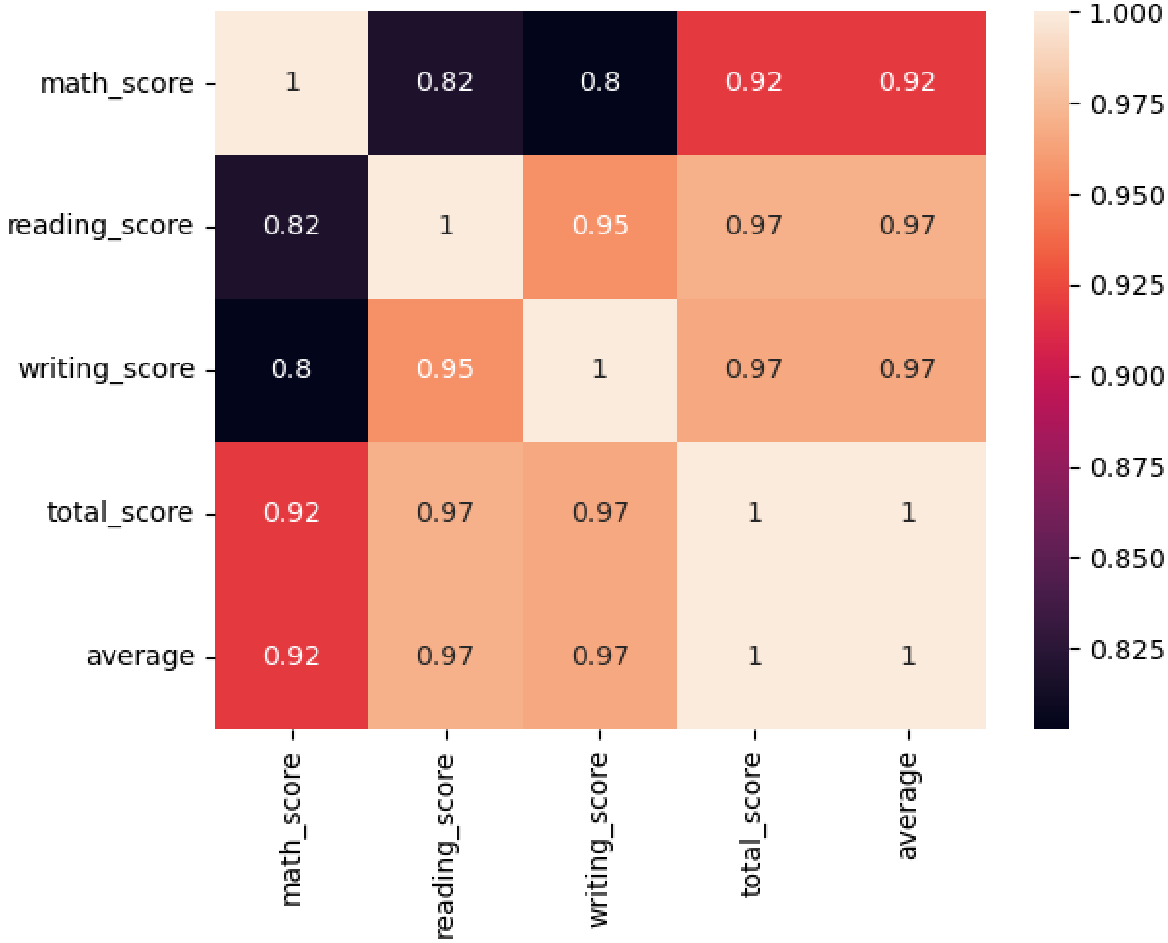
In [23]:
```
#race_ethnicity
plt.subplots(1,3,figsize=(25,6))
plt.subplot(141)
ax=sns.histplot(data=df,x='average',kde=True,hue='race_ethnicity')
```



Insights 1.Students of group A and group B tends to perform poorly in exam 2.Students of group A and group B to perform poorly in exam irrespective of whether they are male or female

In [24]:
```
sns.heatmap(df.corr(),annot=True)
```

Out[24]: <AxesSubplot: >



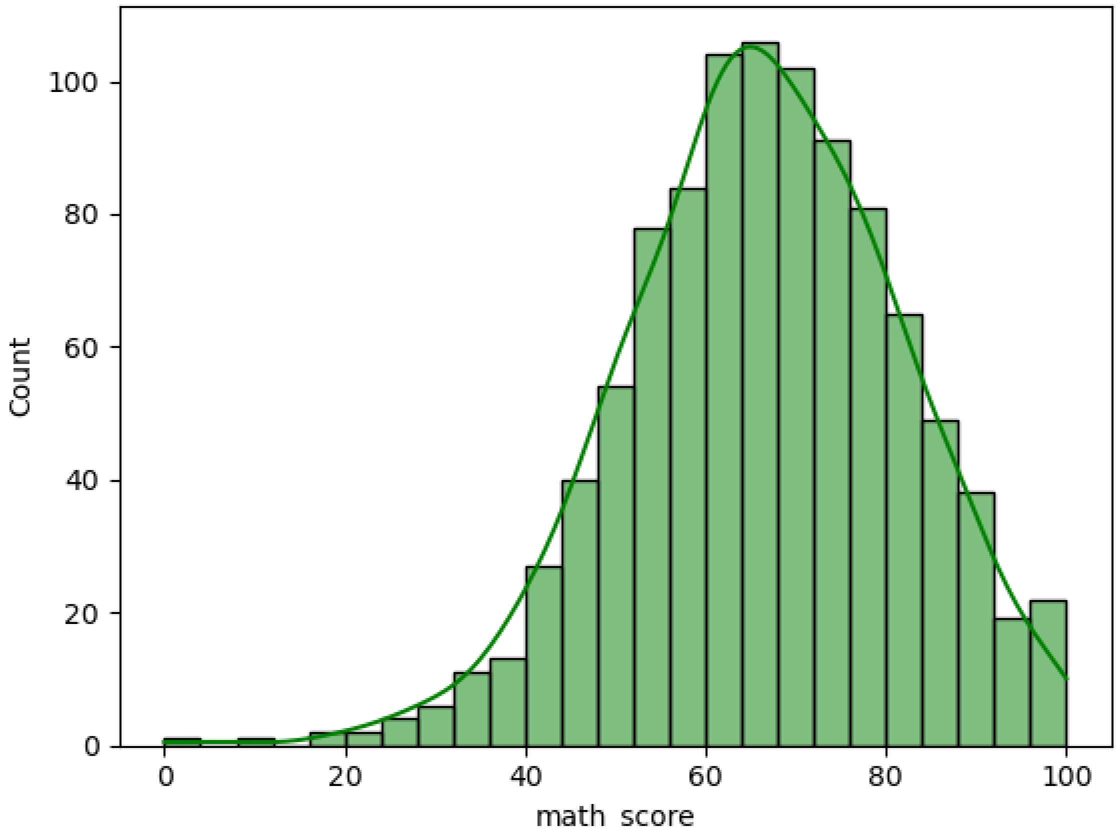
In [25]:
```
df.head(2)
```

Out[25]:

	gender	race_ethnicity	parental_level_of_education	lunch	test_preparation_course	math_scor
0	female	group B	bachelor's degree	standard	none	7
1	female	group C	some college	standard	completed	6

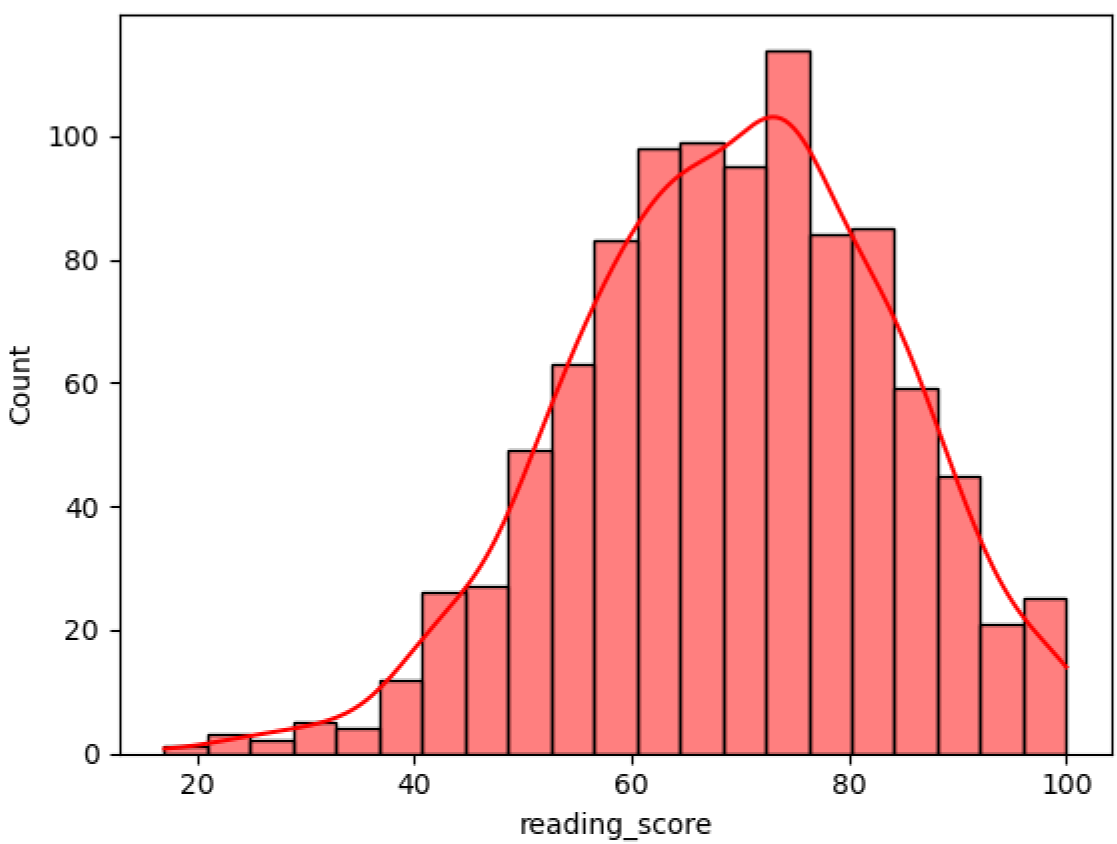
In [26]:
```
sns.histplot(df['math_score'],kde=True,color='g')
```

Out[26]: <AxesSubplot: xlabel='math_score', ylabel='Count'>



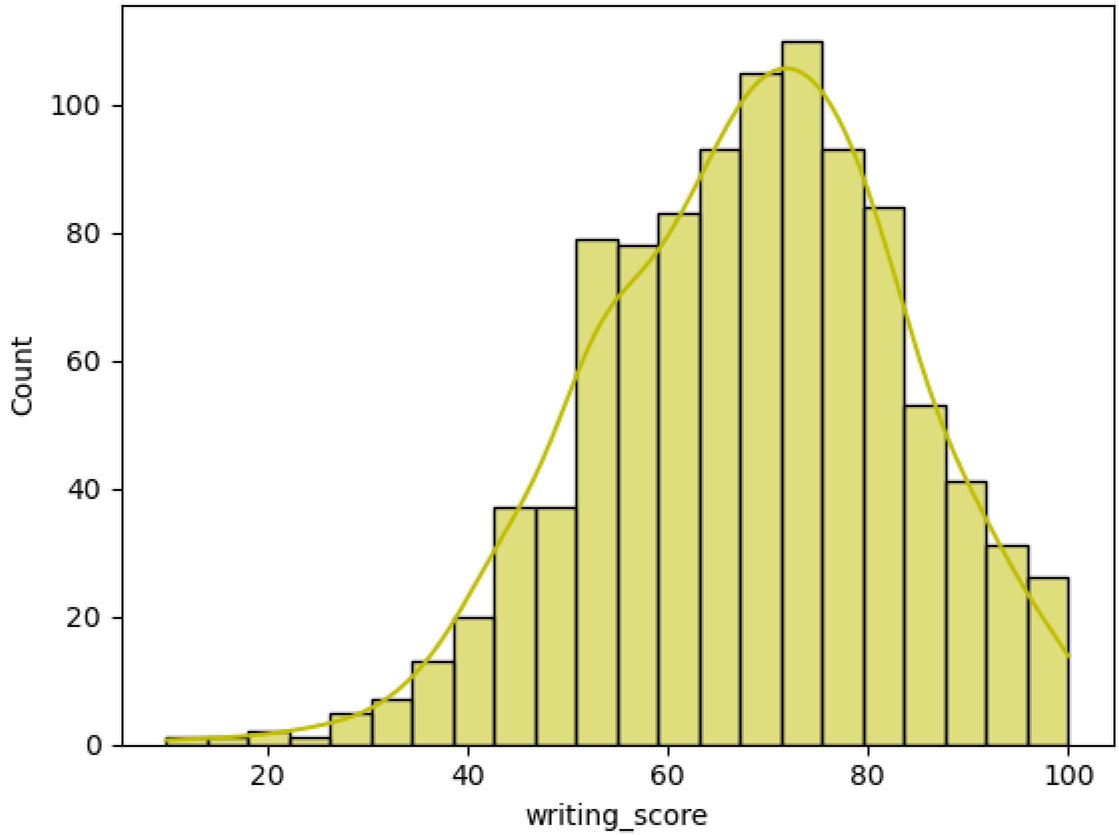
In [27]:
```
sns.histplot(df['reading_score'],kde=True,color='r')
```

Out[27]: <AxesSubplot: xlabel='reading_score', ylabel='Count'>



In [28]:
```
sns.histplot(df['writing_score'],kde=True,color='y')
```

Out[28]: <AxesSubplot: xlabel='writing_score', ylabel='Count'>



Insight: 1.In writing_score more number of students marks between 60 to 80 but no of students marks between 80 to 100 is more compare to reading score. 2.In Both reading and math score more number of students marks between 60 to 80.

In [29]:
```
df.head(5)
```

Out[29]:

	gender	race_ethnicity	parental_level_of_education	lunch	test_preparation_course	math_s
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	

In [30]:
```
df['test_preparation_course'].str.replace('none','Incompleted')
```

Out[30]:
```
0      Incompleted
1        completed
2      Incompleted
3      Incompleted
4      Incompleted
          ...
995      completed
996    Incompleted
997      completed
998      completed
999    Incompleted
Name: test_preparation_course, Length: 1000, dtype: object
```

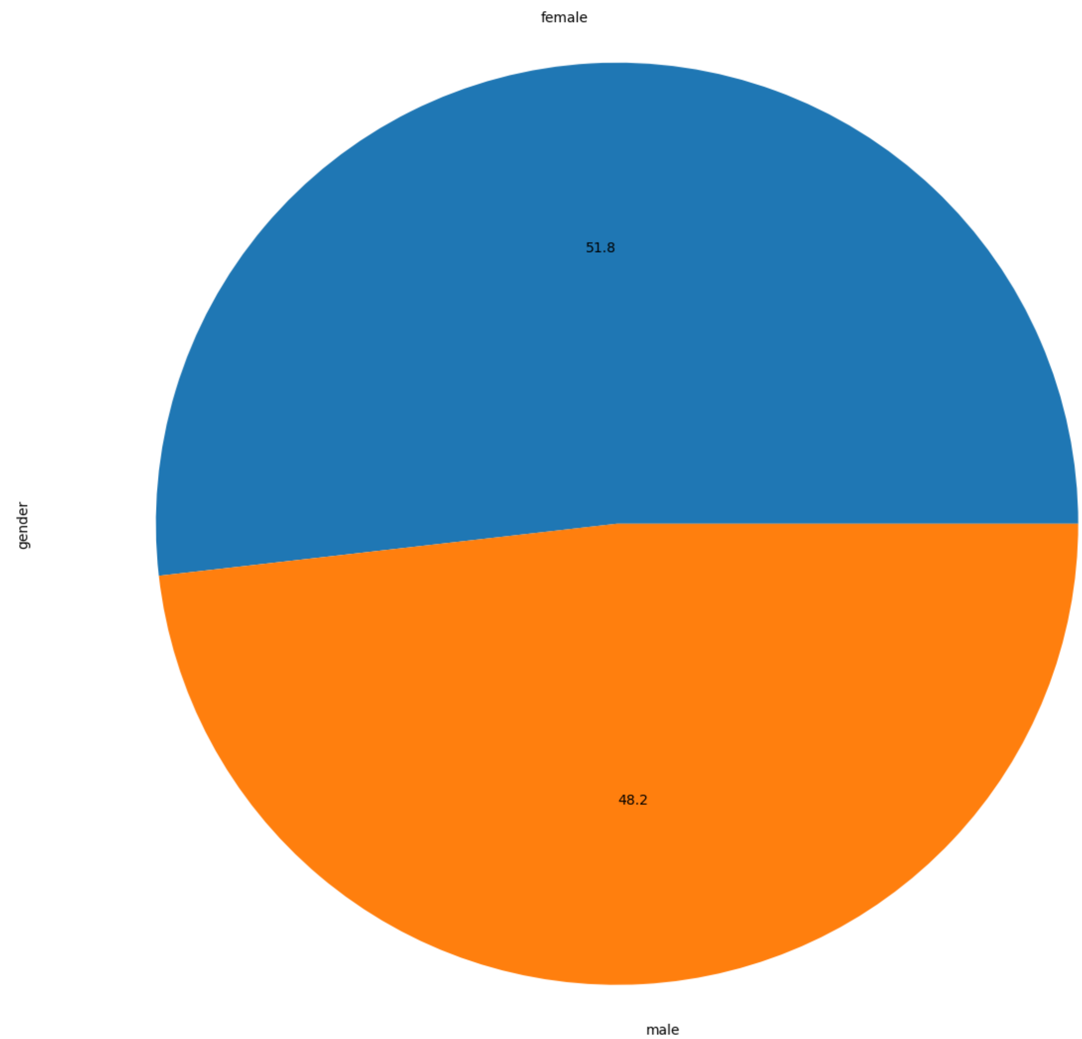
In [31]:
```
df.tail(5)
```

Out[31]:

	gender	race_ethnicity	parental_level_of_education	lunch	test_preparation_course	matl
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	

In [32]:
```
df['gender'].value_counts().plot.pie(x=df['gender'],figsize=(15,16),autopct='%1.1f'
```

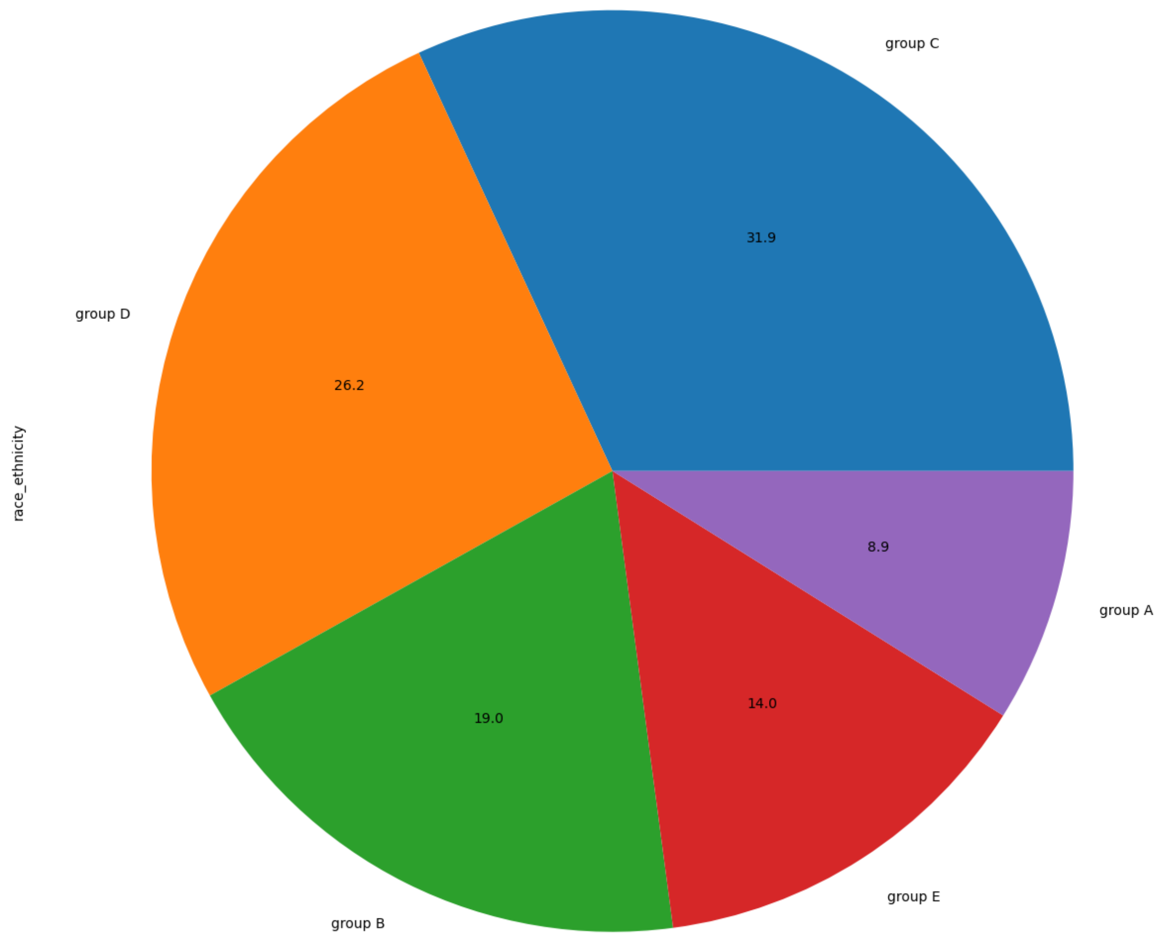
Out[32]:
```
<AxesSubplot: ylabel='gender'>
```



Insight:-In the student performance more number of female compare to male. female:-51.8% MAle:-48.2%

In [33]:
```
df['race_ethnicity'].value_counts().plot.pie(x=df['race_ethnicity'],figsize=(15,16)
```

Out[33]: <AxesSubplot: ylabel='race_ethnicity'>



In this observation more number of Students(female n male)belongs to Group C,and less number of students(female n male) belongs to Group A.

In [34]:
```
df.head(2)
```

Out[34]:

	gender	race_ethnicity	parental_level_of_education	lunch	test_preparation_course	math_scor
0	female	group B	bachelor's degree	standard	none	7
1	female	group C	some college	standard	completed	6

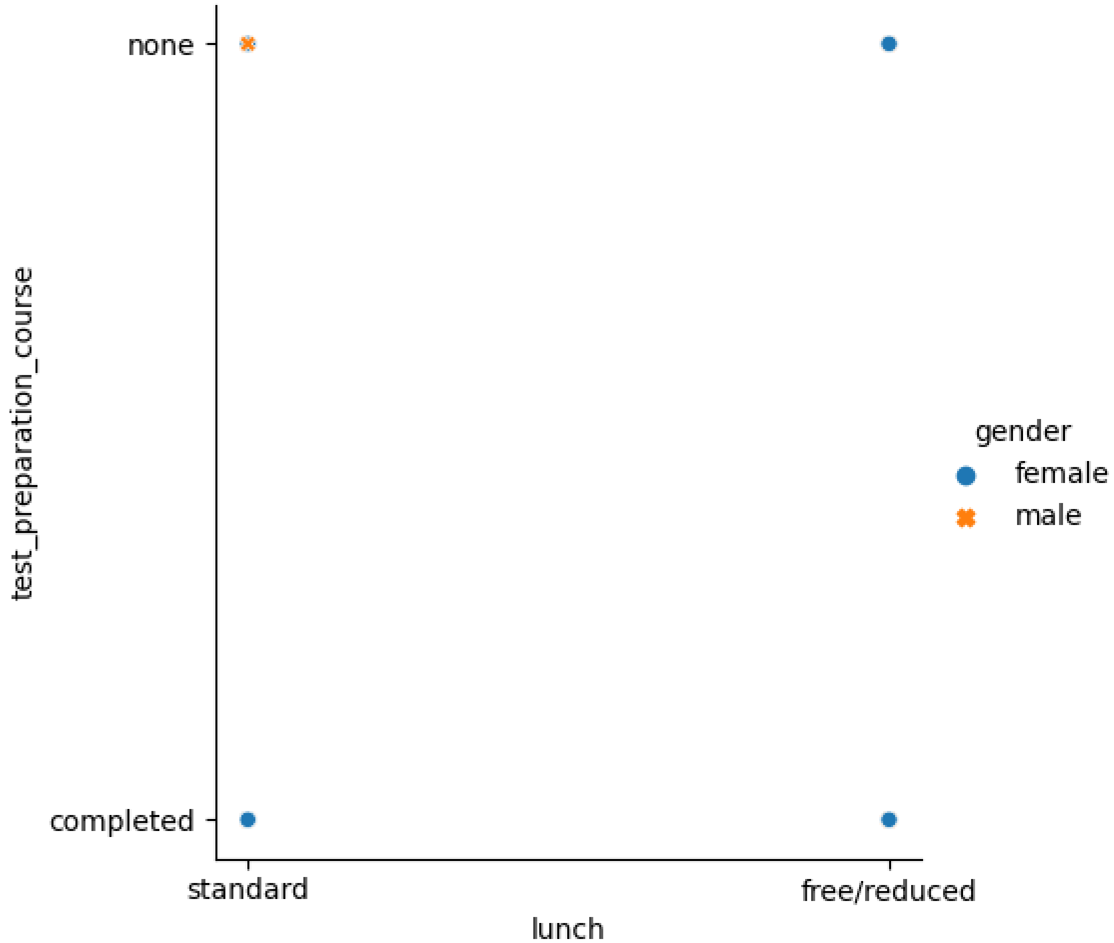
In [35]:
```
df['parental_level_of_education'].max()
```

Out[35]: 'some high school'

Above observation,more number of parents education in 'some high school'.

In [36]:
```
sns.relplot(x='lunch',y='test_preparation_course',data=df,hue='gender',style='gende
```

Out[36]: <seaborn.axisgrid.FacetGrid at 0x7fb0a75bfcd0>



In this above observation females belong to standard lunch they completed their test preparation but no male belongs to standard or free lunch completed their test preparation course

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: