In [40]:
```
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 [41]:
```
# read the dataset
df= pd.read_csv('D:\datasets\StudentsPerformance.csv')
```

In [42]:
```
df.head()
```

Out[42]:

	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 [43]:
```
## check missing values
df.isnull().sum()
```

Out[43]:
```
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
```

Insights or Observation

There are no missing values

In [44]:
```
df.isna().sum()
```

Out[44]:
```
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 [45]:
```
## check duplicates
df.duplicated().sum()
```

Out[45]:
```
0
```

There are no duplicates values in the dataset

In [46]:
```
## check datatypes
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 [47]:
```
## 3.1 Checking the number of uniques values of each columns
df.nunique()
```

Out[47]:
```
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 [48]:
```
## Check the statistics of the datasets
df.describe()
```

Out[48]:

	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

Insights or oservation

- From the above description of numerical data, all means are very close to each other between 66 and 69.
- All the standard deviation are also close between 14.6-15.19.
- While ther is a minimum of 0 for maths, others are having 17 and 10 value.

In [49]:
```
[feature for feature in df.columns if df[feature].dtype=='O']
```

Out[49]:
```
['gender',
 'race/ethnicity',
 'parental level of education',
 'lunch',
 'test preparation course']
```

In [50]:
```
# segrregate numerical and categorical features
numerical_features=[feature for feature in df.columns if df[feature].dtype!='O']
categorical_feature=[feature for feature in df.columns if df[feature].dtype=='O']
```

In [51]:
```
numerical_features
```

Out[51]:
```
['math score', 'reading score', 'writing score']
```

In [52]:
```
categorical_feature
```

Out[52]:
```
['gender',
 'race/ethnicity',
 'parental level of education',
 'lunch',
 'test preparation course']
```

In [53]:
```
df['gender'].value_counts()
```

Out[53]:
```
gender
female    518
male      482
Name: count, dtype: int64
```

In [54]:
```
df['race/ethnicity'].value_counts()
```

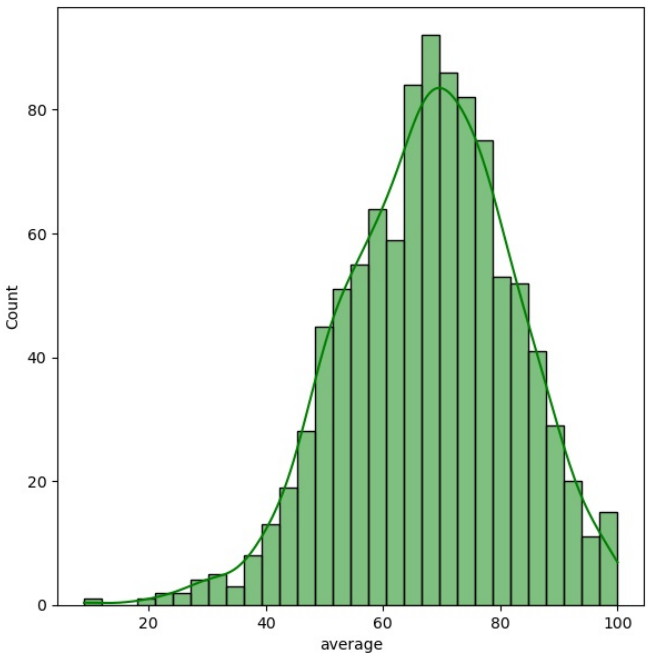
Out[54]:
```
race/ethnicity
group C    319
group D    262
group B    190
group E    140
group A     89
Name: count, dtype: int64
```

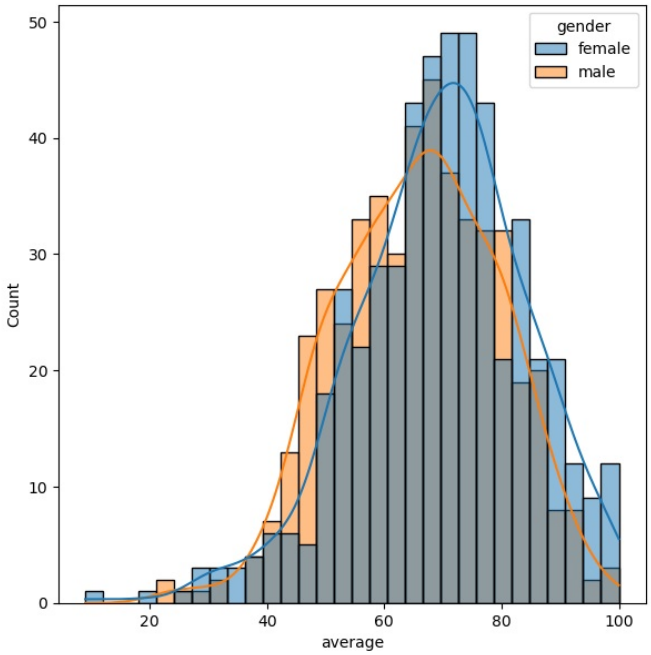
In [55]:
```
## Aggregate the total score with the mean

df['total_score']=(df['math score']+df['reading score']+df['writing score'])
df['average']=df['total_score']/3
df.head()
```

Out[55]:

	gender	race/ethnicity	parental level of education	lunch	test preparation course	math score	reading score	writing score	total_score	average
0	female	group B	bachelor's degree	standard	none	72	72	74	218	72.666667
1	female	group C	some college	standard	completed	69	90	88	247	82.333333
2	female	group B	master's degree	standard	none	90	95	93	278	92.666667
3	male	group A	associate's degree	free/reduced	none	47	57	44	148	49.333333
4	male	group C	some college	standard	none	76	78	75	229	76.333333

In [59]:
```
### Explore More Visualization
fig,axis=plt.subplots(1,2,figsize=(15,7))
plt.subplot(121)
sns.histplot(data=df,x='average',bins=30,kde=True,color='g')
plt.subplot(122)
sns.histplot(data=df,x='average',bins=30,kde=True,hue='gender')
plt.show()
```

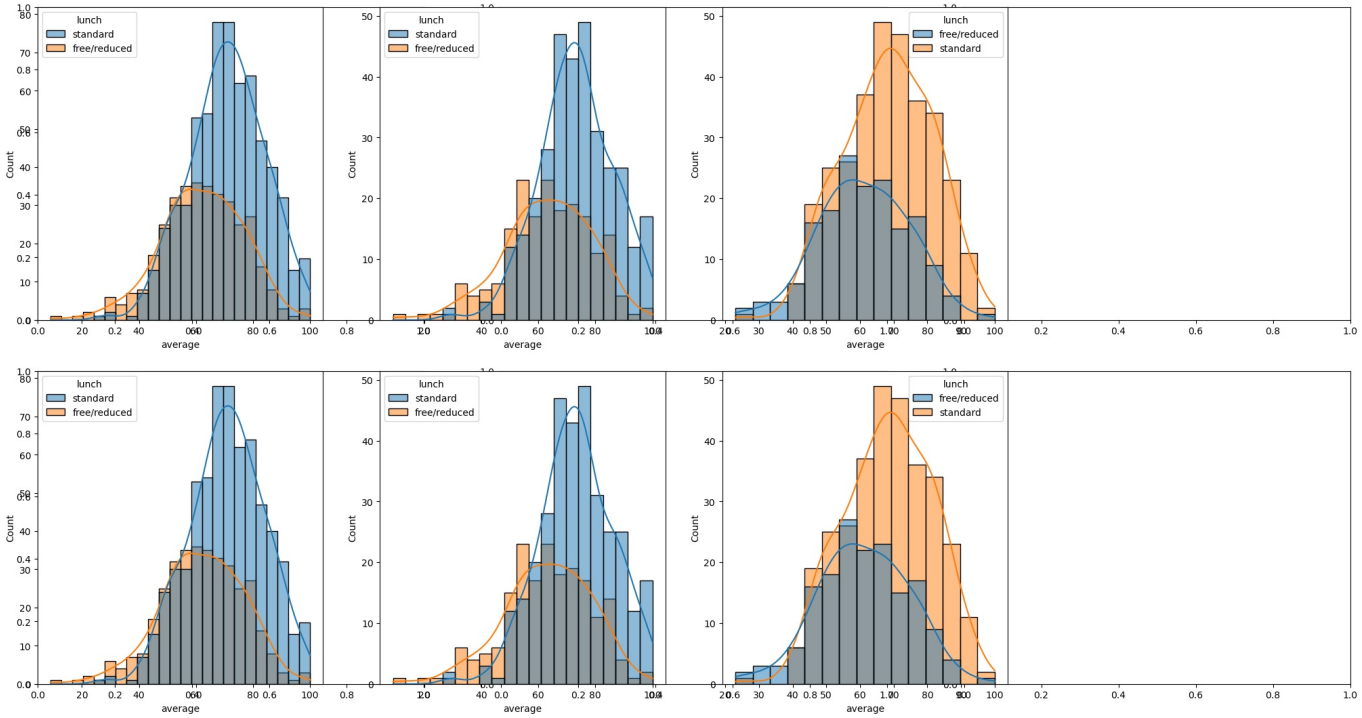




Insights

- Female student tend to perform well than male students

In [61]:
```
plt.subplots(1,3,figsize=(25,6))
plt.subplot(141)
sns.histplot(data=df,x='average',kde=True,hue='lunch')
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')
plt.show()
```



Insights

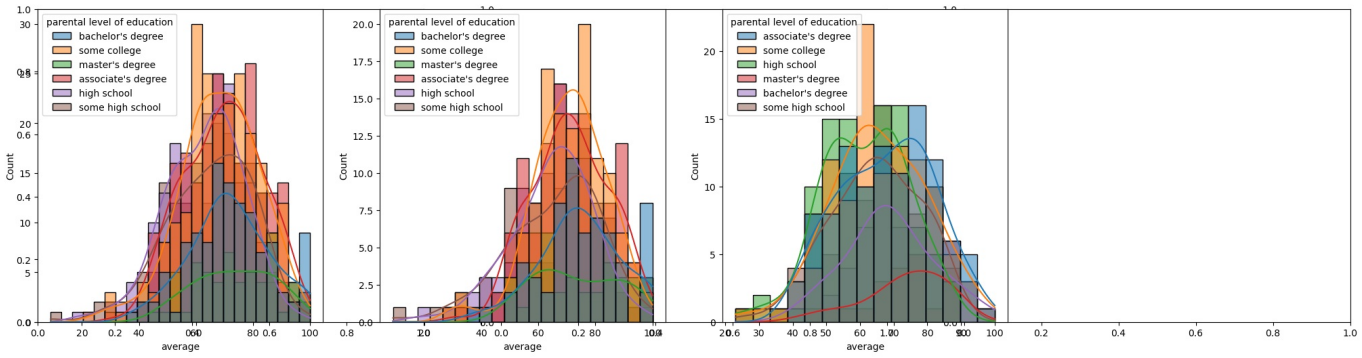
- Standard lunch help students perform well in exams
- Standard lunch helps perform well in exams be it a male of female

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

Out[63]:

	gender	race/ethnicity	parental level of education	lunch	test preparation course	math score	reading score	writing score	total_score	average
0	female	group B	bachelor's degree	standard	none	72	72	74	218	72.666667
1	female	group C	some college	standard	completed	69	90	88	247	82.333333
2	female	group B	master's degree	standard	none	90	95	93	278	92.666667
3	male	group A	associate's degree	free/reduced	none	47	57	44	148	49.333333
4	male	group C	some college	standard	none	76	78	75	229	76.333333

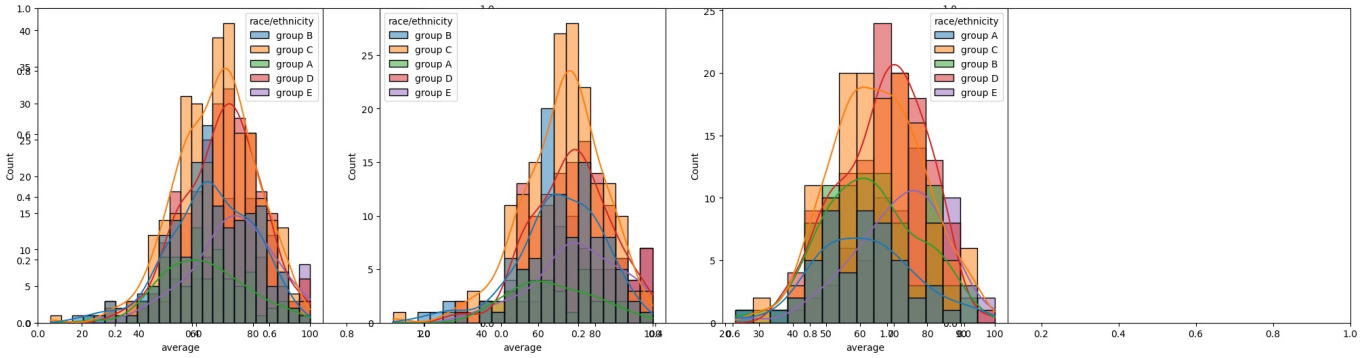
In [66]:
```
plt.subplots(1,3,figsize=(25,6))
plt.subplot(141)
sns.histplot(data=df,x='average',kde=True,hue='parental level of education')
plt.subplot(142)
sns.histplot(data=df[df.gender=='female'],x='average',kde=True,hue='parental level of education')
plt.subplot(143)
sns.histplot(data=df[df.gender=='male'],x='average',kde=True,hue='parental level of education')
plt.show()
```



Insights

- In general parent's education don't help student perform well in exam
- 3rd plot shows that parent's whose education is of associate's degree or master's degree their male child tend to perform well in exam
- 2nd plot we can see there is no effect of parent's education on female students.

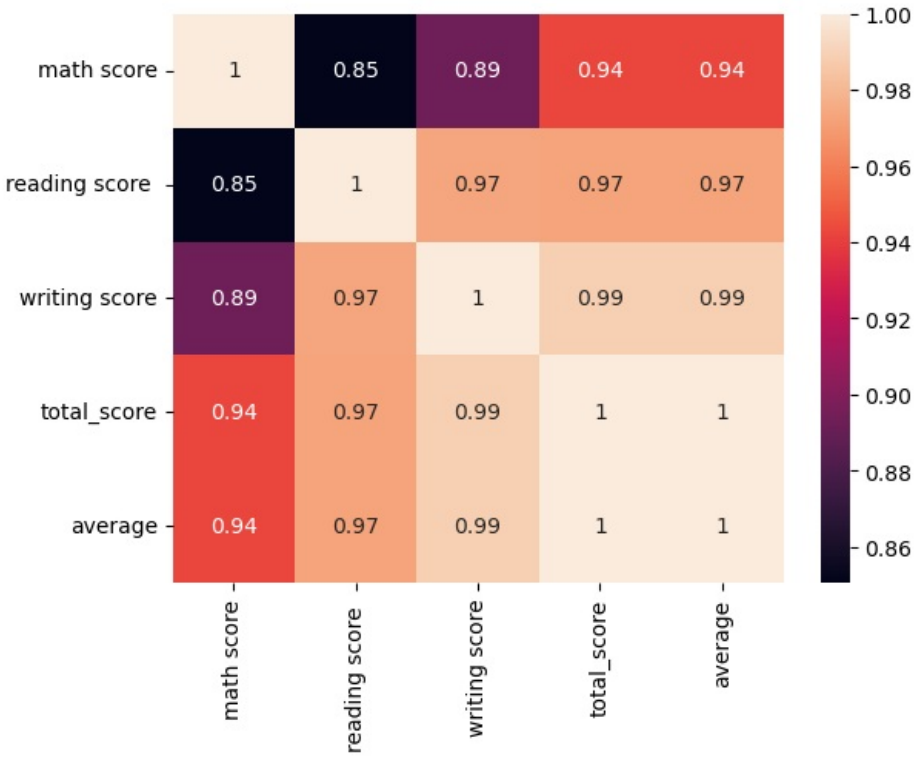
In [70]:
```
plt.subplots(1,3,figsize=(25,6))
plt.subplot(141)
sns.histplot(data=df,x='average',kde=True,hue='race/ethnicity')
plt.subplot(142)
sns.histplot(data=df[df.gender=='female'],x='average',kde=True,hue='race/ethnicity')
plt.subplot(143)
sns.histplot(data=df[df.gender=='male'],x='average',kde=True,hue='race/ethnicity')
plt.show()
```



Insights

- Students of group A and group B tends to perform poorly in exam
- Students of group A and group B tends to perform poorly in exam irrespective of whether they are male or female

In [97]:
```
[]
sns.heatmap(df.corr(), annot=True)
plt.show()
```



In [ ]:

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js