EDA Student Performance Indicator

1) Problem statement

- This project understands how the student's performance (test scores) is affected by other variables such as Gender, Ethnicity, Parental level of education, Lunch and Test preparation course.

2) Data Collection

- Dataset Source - https://www.kaggle.com/datasets/spscientist/students-performance-in-exams?datasetId=74977 (https://www.kaggle.com/datasets/spscientist/students-performance-in-exams?datasetId=74977)
- The data consists of 8 column and 1000 rows.

3) Dataset Information

- gender : sex of students -> (Male/female)
- race/ethnicity : ethnicity of students -> (Group A, B,C, D,E)
- parental level of education : parents' final education ->(bachelor's degree,some college,master's degree,associate's degree,high school)
- lunch : having lunch before test (standard or free/reduced)

- test preparation course : complete or not complete before test
- math score
- reading score
- writing score

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]:

```
# Read the dataset
df=pd.read_csv('Student Performance Dataset.csv')
df.head()
```

Out[2]:

	gender	race_ethnicity	parental_level_of_education	lunch	test_preparation_course	matl
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]:

```
df.shape
```

Out[3]:

```
(1000, 8)
```

3. Data Checks to perform

- Check Missing values
- Check Duplicates
- Check data type
- Check the number of unique values of each column
- Check statistics of data set
- Check various categories present in the different categorical column

In [4]:

```
## check missing Values
df.isnull().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
```

Insights or Observation

- There Are No Missing Values

In [5]:

```
df.isna().sum()
```

Out[5]:

```
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 Null Values

In [6]:

```
## Check Duplicates
df.duplicated().sum()
```

Out[6]:

```
0
```

Insights or Observation

- There are no duplicates values in the dataset

In [7]:

```
## 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 [8]:

```
## 3.1 Checking the number of uniques values of each columns
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 the 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

Insights or Observations

- 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 there is a minimum of 0 for maths ,other are having 17 score for reading and 10 score for writing.

In [10]:

```
## Explore more info about the data
df.head()
```

Out[10]:

	gender	race_ethnicity	parental_level_of_education	lunch	test_preparation_course	matl
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()
```

Out[11]:

	gender	race_ethnicity	parental_level_of_education	lunch	test_preparation_course	m
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 if df[feature].dtype=='O']
```

Out[12]:

```
['gender',
 'race_ethnicity',
 'parental_level_of_education',
 'lunch',
 'test_preparation_course']
```

In [13]:

```
#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 [14]:

```
numerical_features
```

Out[14]:

```
['math_score', 'reading_score', 'writing_score']
```

In [15]:

```
categorical_feature
```

Out[15]:

```
['gender',
 'race_ethnicity',
 'parental_level_of_education',
 'lunch',
 'test_preparation_course']
```

In [16]:

```
df['gender'].value_counts()
```

Out[16]:

```
female    518
male      482
Name: gender, dtype: int64
```

In [17]:

```
df['race_ethnicity'].value_counts()
```

Out[17]:

```
group C    319
group D    262
group B    190
group E    140
group A     89
Name: race_ethnicity, dtype: int64
```

In [18]:

```
## 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[18]:

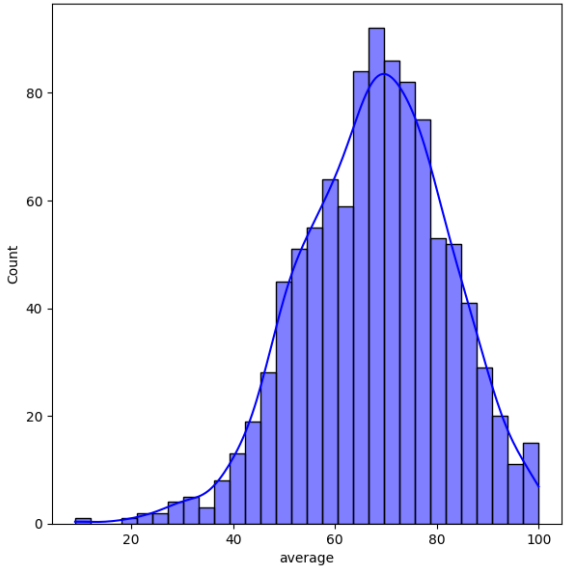
	gender	race_ethnicity	parental_level_of_education	lunch	test_preparation_course	matl
0	female	group B	bachelor's degree	standard	none	
1	female	group C	some college	standard	completed	
2	female	group B	master's degree	standard	none	
3	male	group A	associate's degree	free/reduced	none	
4	male	group C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

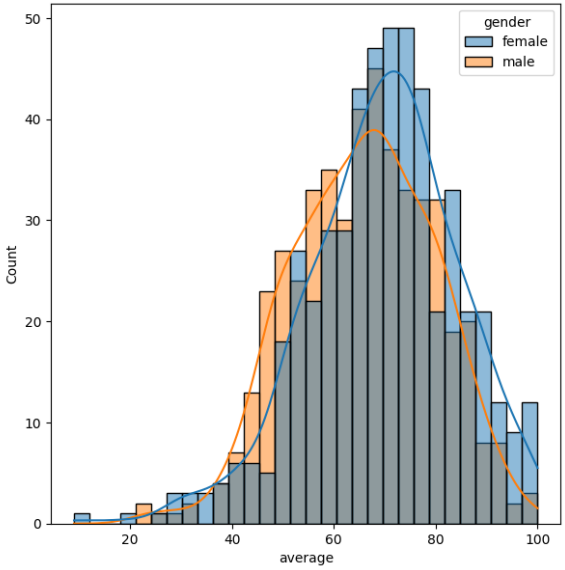
In [19]:

```
### 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='b')
plt.subplot(122)
sns.histplot(data=df,x='average',bins=30,kde=True,hue='gender')
```

Out[19]:

```
<Axes: xlabel='average', ylabel='Count'>
```


Count
average
0
20
40
60
80
100
20
40
60
80

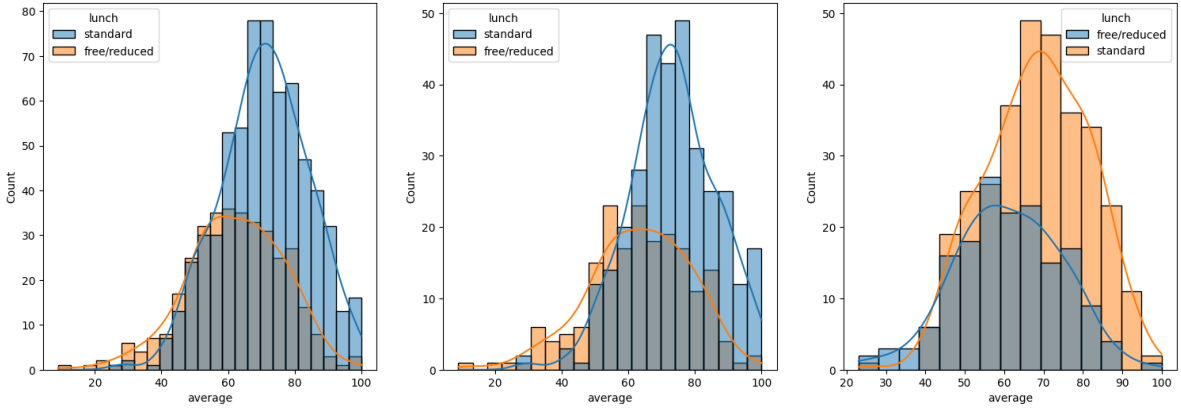

gender
female
male
Count
average
0
10
20
30
40
50
20
40
60
80
100

In [20]:

```
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')
```

Out[20]:

```
<Axes: xlabel='average', ylabel='Count'>
```



Insights or Observations

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

In [21]:

```
df.head()
```

Out[21]:

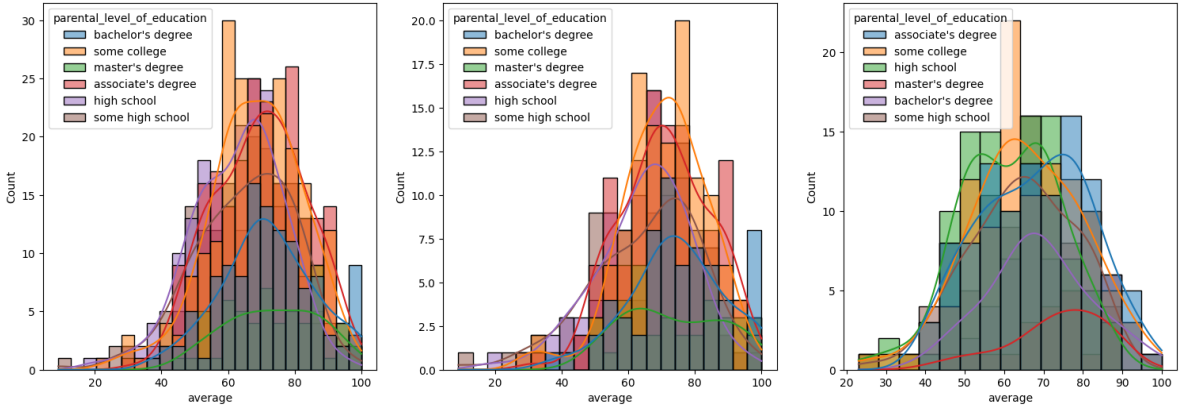
	gender	race_ethnicity	parental_level_of_education	lunch	test_preparation_course	matl
0	female	group B	bachelor's degree	standard	none	
1	female	group C	some college	standard	completed	
2	female	group B	master's degree	standard	none	
3	male	group A	associate's degree	free/reduced	none	
4	male	group C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

In [22]:

```
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_educa
plt.subplot(143)
sns.histplot(data=df[df.gender=='male'],x='average',kde=True,hue='parental_level_of_educati
```

Out[22]:

```
<Axes: xlabel='average', ylabel='Count'>
```

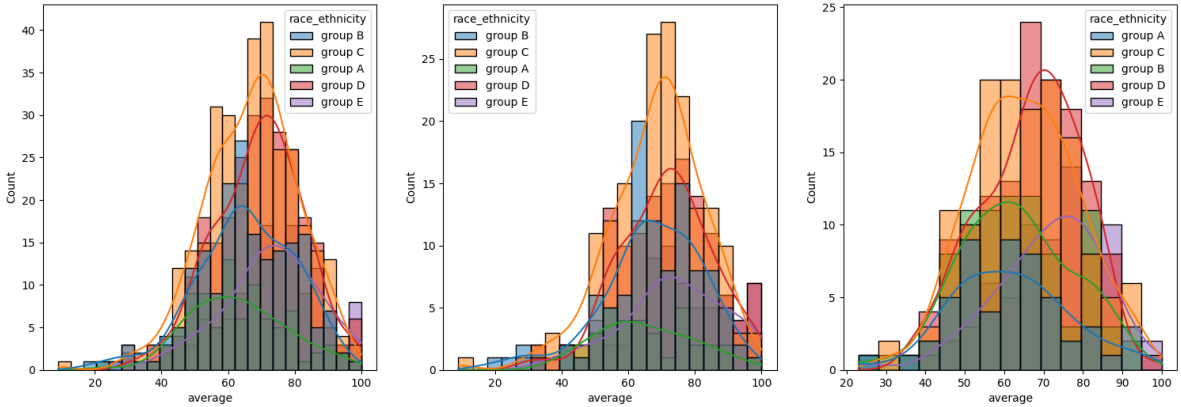


Insights or Observations

- In general parent's education don't help student perform well in exam.
- 2nd plot we can see there is no effect of parent's education on female students.
- 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

In [23]:

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



Insights or Observations

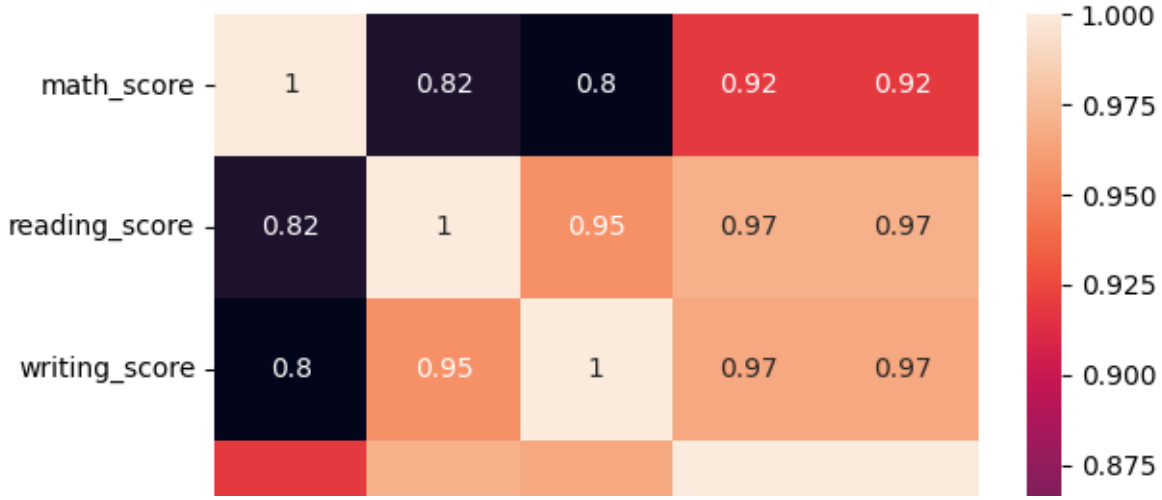
- Students of group A and group B tends to perform poorly in exam irrespective of whether they are male or female
- Students Of Group C performs well in general and in female graph.

In [24]:

```
sns.heatmap(df.corr(),annot=True)
```

Out[24]:

<Axes: >


math_score
reading_score
writing_score
1
0.82
0.8
0.92
0.92
0.82
1
0.95
0.97
0.97
0.8
0.95
1
0.97
0.97
1.000
0.975
0.950
0.925
0.900
0.875