Exploratory Data Analysis(EDA) using Students Performance Dataset

```
        -Pranav Chaudhari
```

Exploratory Data Analysis, or EDA, is an important step in any Data Analysis or Data Science project. EDA is the process of investigating the dataset to discover patterns, and anomalies (outliers), and form hypotheses based on our understanding of the dataset.

Importing the libraries we will require for performing EDA

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
```

Reading data, You can download the dataset for your reference.

```
df = pd.read_csv('StudentsPerformance.csv')
print(df.head())
```

```
   gender race/ethnicity parental level of education         lunch  \
0  female        group B           bachelor's degree      standard   
1  female        group C                some college      standard   
2  female        group B             master's degree      standard   
3    male        group A          associate's degree  free/reduced   
4    male        group C                some college      standard   

  test preparation course  math score  reading score  writing score  
0                    none          72             72             74  
1               completed          69             90             88  
2                    none          90             95             93  
3                    none          47             57             44  
4                    none          76             78             75  
```

Descriptive Statistics

Perfect! The data looks just like we wanted it to. You can easily tell just by looking at the dataset that it contains data about different students at a school/college, and their scores in 3 subjects. Let us start by looking at descriptive statistic parameters for the dataset. We will use describe() for this.

By assigning include attribute a value of 'all', we make sure that categorical features are also included in the result.

```
df.describe(include='all')
```

	gender	race/ethnicity	parental level of education	lunch	test preparation course	math score	reading score	writing score
count	1000	1000	1000	1000	1000	1000.00000	1000.000000	1000.000000
unique	2	5	6	2	2	NaN	NaN	NaN
top	female	group C	some college	standard	none	NaN	NaN	NaN
freq	518	319	226	645	642	NaN	NaN	NaN
mean	NaN	NaN	NaN	NaN	NaN	66.08900	69.169000	68.054000
std	NaN	NaN	NaN	NaN	NaN	15.16308	14.600192	15.195657
min	NaN	NaN	NaN	NaN	NaN	0.00000	17.000000	10.000000
25%	NaN	NaN	NaN	NaN	NaN	57.00000	59.000000	57.750000
50%	NaN	NaN	NaN	NaN	NaN	66.00000	70.000000	69.000000
75%	NaN	NaN	NaN	NaN	NaN	77.00000	79.000000	79.000000
max	NaN	NaN	NaN	NaN	NaN	100.00000	100.000000	100.000000

Missing value imputation

We will now check for missing values in our dataset. In case there are any missing entries, we will impute them with appropriate values (mode in case of categorical feature, and median or mean in case of numerical feature). We will use the isnull() function for this purpose.

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

	0
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

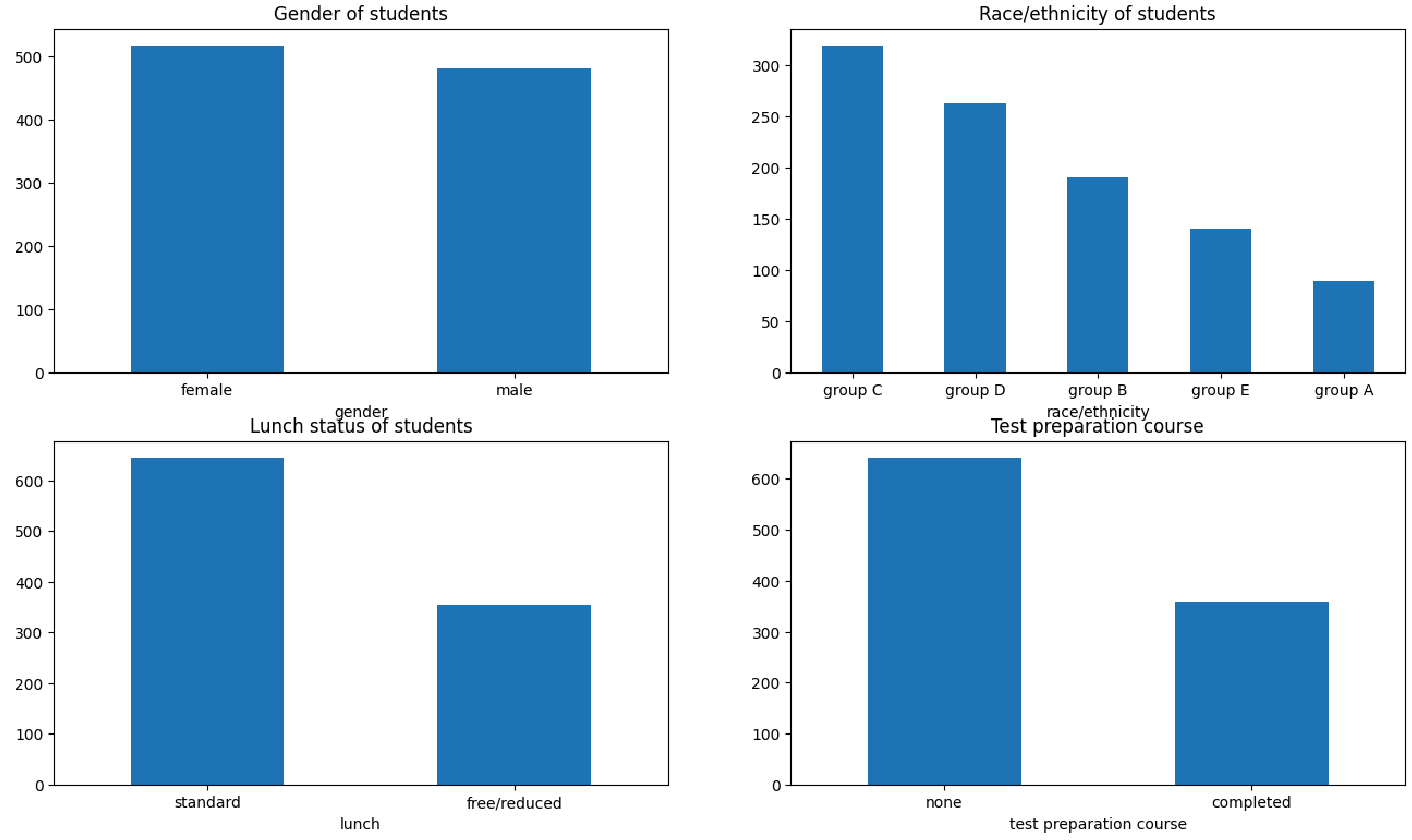
Fortunately for us, there are no missing values in this dataset. We will now proceed to analyze this dataset, observe patterns, and identify outliers with the help of graphs and figures.

Graphical representation

We will start with Univariate Analysis. We will be using a bar graph for this purpose. We will look at the distribution of students across gender, race/ethnicity, their lunch status, and whether they have a test preparation course or not.

```
plt.subplot(221)

df['gender'].value_counts().plot(kind ='bar', title = 'Gender of students', figsize=(16,9))
```

```
plt.xticks(rotation=0)

plt.subplot(222)

df['race/ethnicity'].value_counts().plot(kind='bar', title='Race/ethnicity of students')

plt.xticks(rotation=0)

plt.subplot(223)

df['lunch'].value_counts().plot(kind='bar', title='Lunch status of students')

plt.xticks(rotation=0)

plt.subplot(224)

df['test preparation course'].value_counts().plot(kind='bar', title='Test preparation course')

plt.xticks(rotation=0)

plt.show()
```

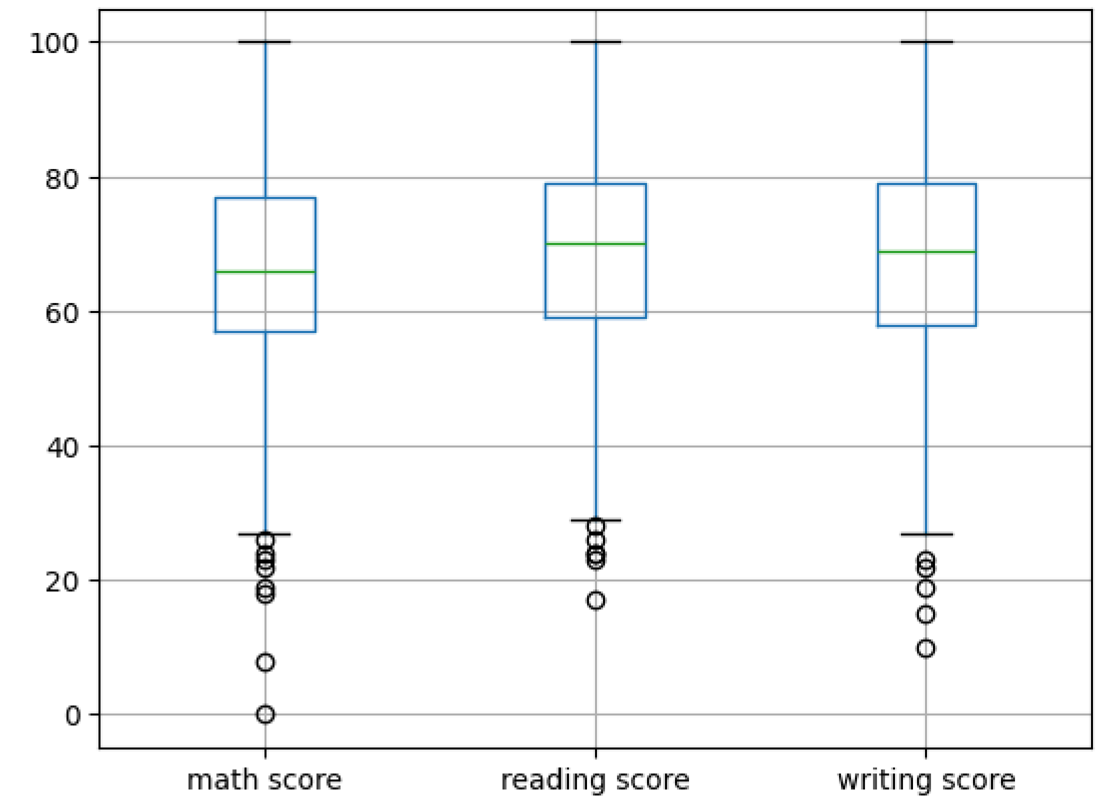


We can infer many things from the graph. There are more girls in the school than boys. The majority of the students belong to groups C and D. More than 60% of the students have a standard lunch at school. Also, more than 60% of students have not taken any test preparation course.

Continuing with Univariate Analysis, next, we will be making a boxplot of the numerical columns (math score, reading score, and writing score) in the dataset. A boxplot helps us in visualizing the data in terms of quartiles. It also identifies outliers in the dataset, if any. We will use the boxplot() function for this.

```
df.boxplot()
```

```
<Axes: >
```

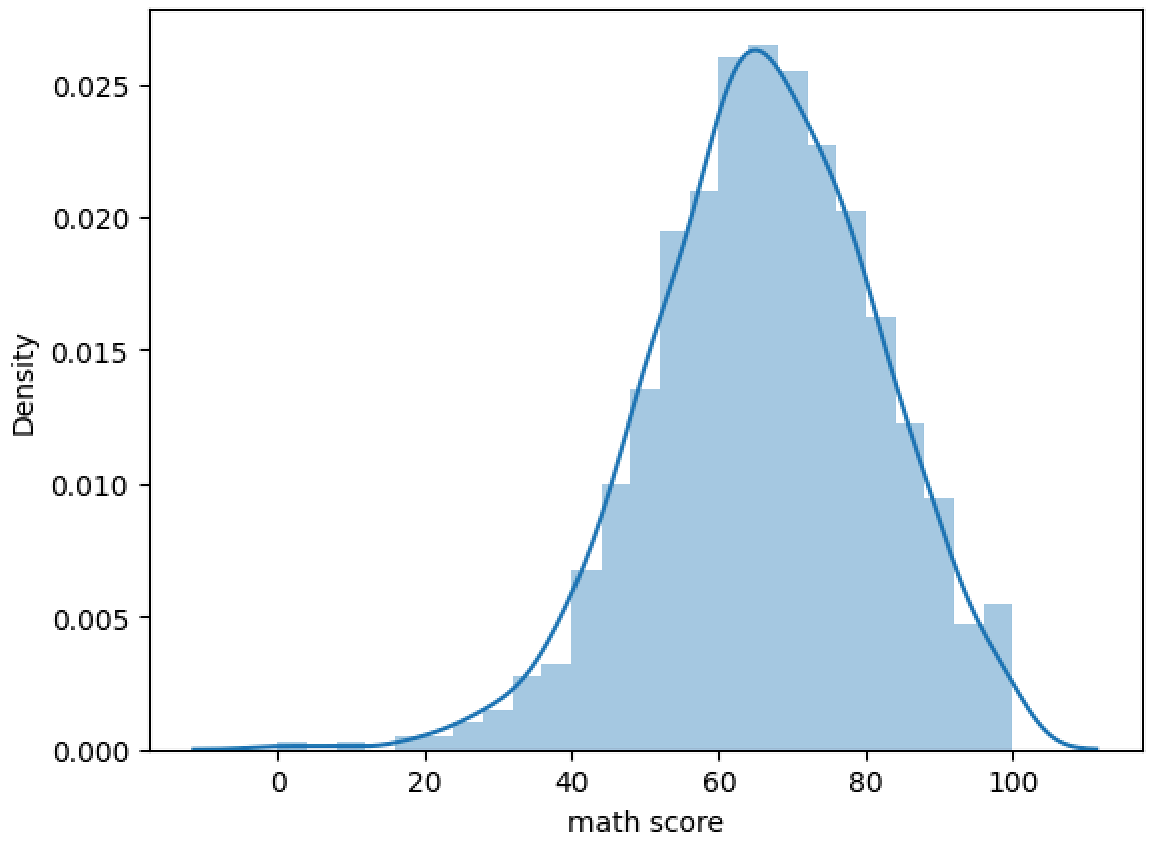


The middle portion represents the inter-quartile range (IQR). The horizontal green line in the middle represents the median of the data. The hollow circles near the tails represent outliers in the dataset. However, since it is very much possible for a student to score extremely low marks in a test, we will not remove these outliers.

We will now make a distribution plot of the math score of the students. A distribution plot tells us how the data is distributed. We will use the distplot function.

```
sns.distplot(df['math score'])
```

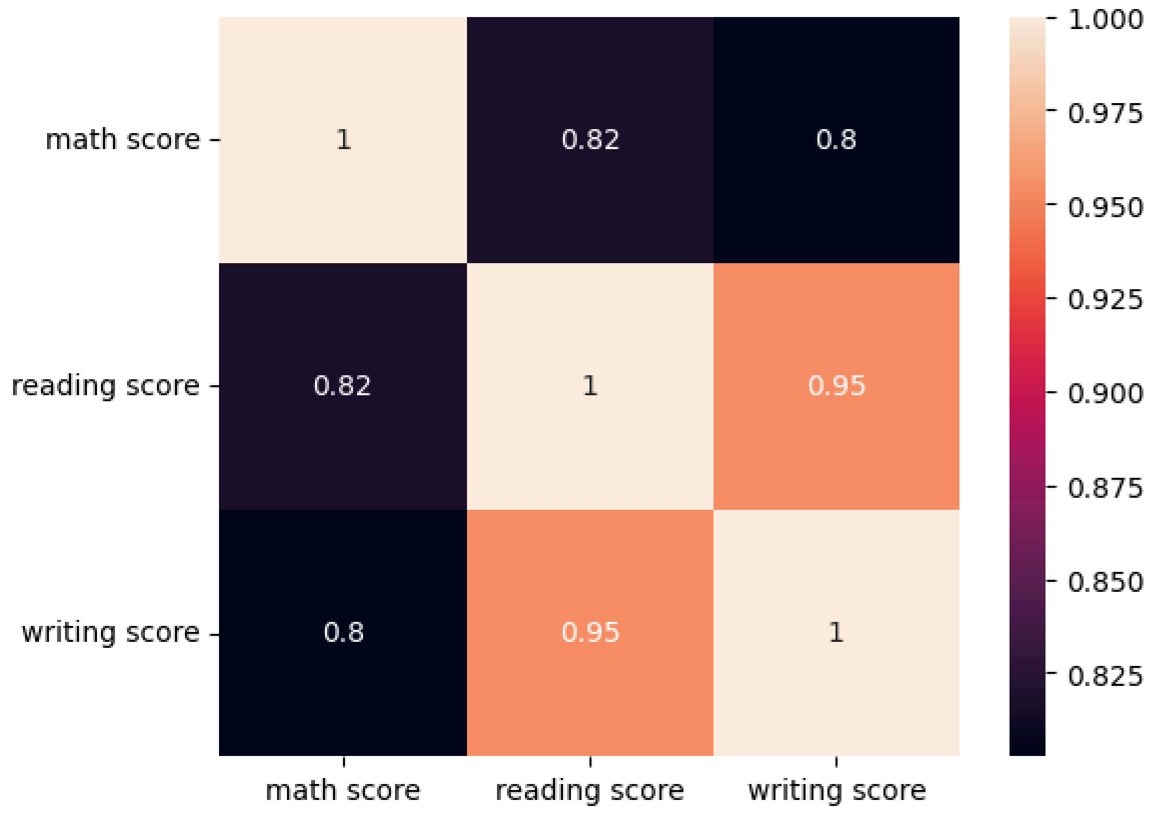
```
<ipython-input-11-913cdd0f89a7>:1: UserWarning: 

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(df['math score'])
<Axes: xlabel='math score', ylabel='Density'>
```



The graph represents a perfect bell curve closely. The peak is at around 65 marks, the mean of the math score of the students in the dataset. A similar distribution plot can also be made for reading scores and writing scores.

We will now look at the correlation between the 3 scores with the help of a heatmap. For this, we will use corr() and heatmap() function for this exercise.

```
corr = df.select_dtypes(include=np.number).corr()
sns.heatmap(corr, annot=True, square=True)
plt.yticks(rotation=0)
plt.show()
```

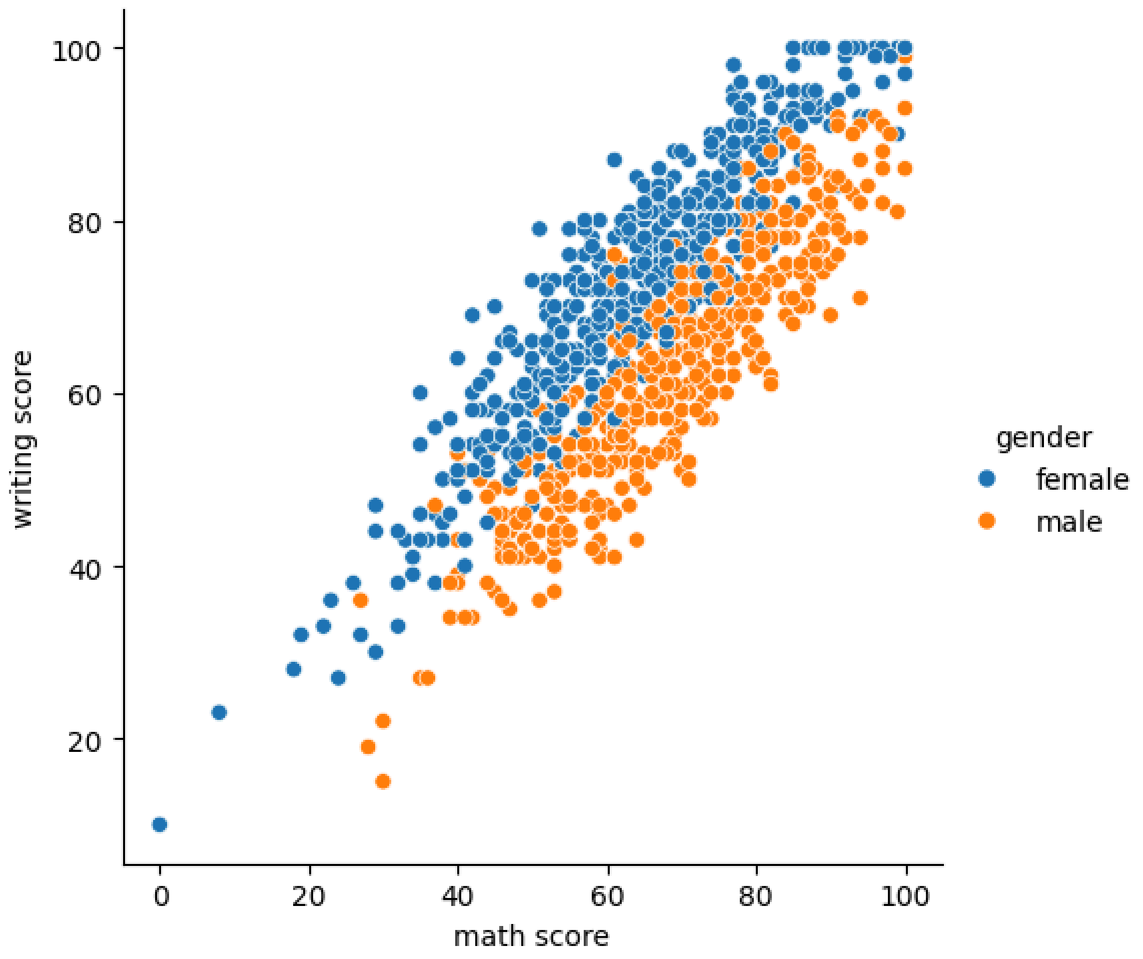


The heatmap shows that the 3 scores are highly correlated. Reading score has a correlation coefficient of 0.95 with the writing score. Math score has a correlation coefficient of 0.82 with the reading score, and 0.80 with the writing score.

We will now move on to Bivariate Analysis. We will look at a relational plot in Seaborn. It helps us to understand the relationship between 2 variables on different subsets of the dataset. We will try to understand the relationship between the math score and the writing score of students of different genders.

```
sns.relplot(x='math score', y='writing score', hue='gender', data=df)
```

```
<seaborn.axisgrid.FacetGrid at 0x7fddaf648c10>
```



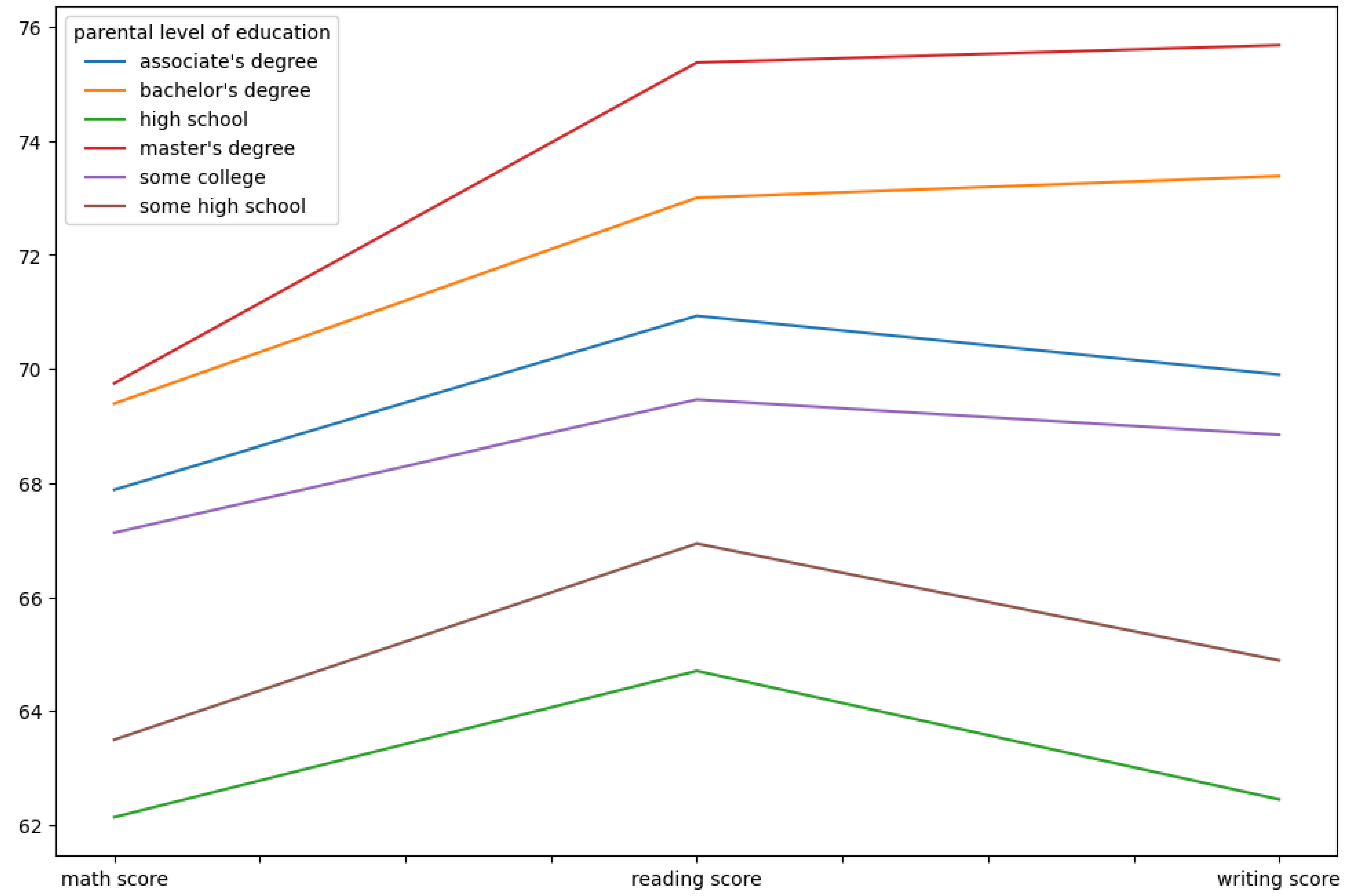
The graph shows a clear difference in scores between the male and female students. For the same math score, female students are more likely to have a higher writing score than male students. However, for the same writing score, male students are expected to have a higher math score than female students.

Relational plots help us in conducting bivariate analysis.

Finally, we will analyze students' performance in math, reading, and writing based on the level of education of their parents and test preparation course. First, let us have a look at the impact of parents' level of education on their child's performance in school using a **line plot**.

```
df.groupby('parental level of education')[['math score', 'reading score', 'writing score']].mean().T.plot(figsize=(12,8))
```

```
<Axes: >
```

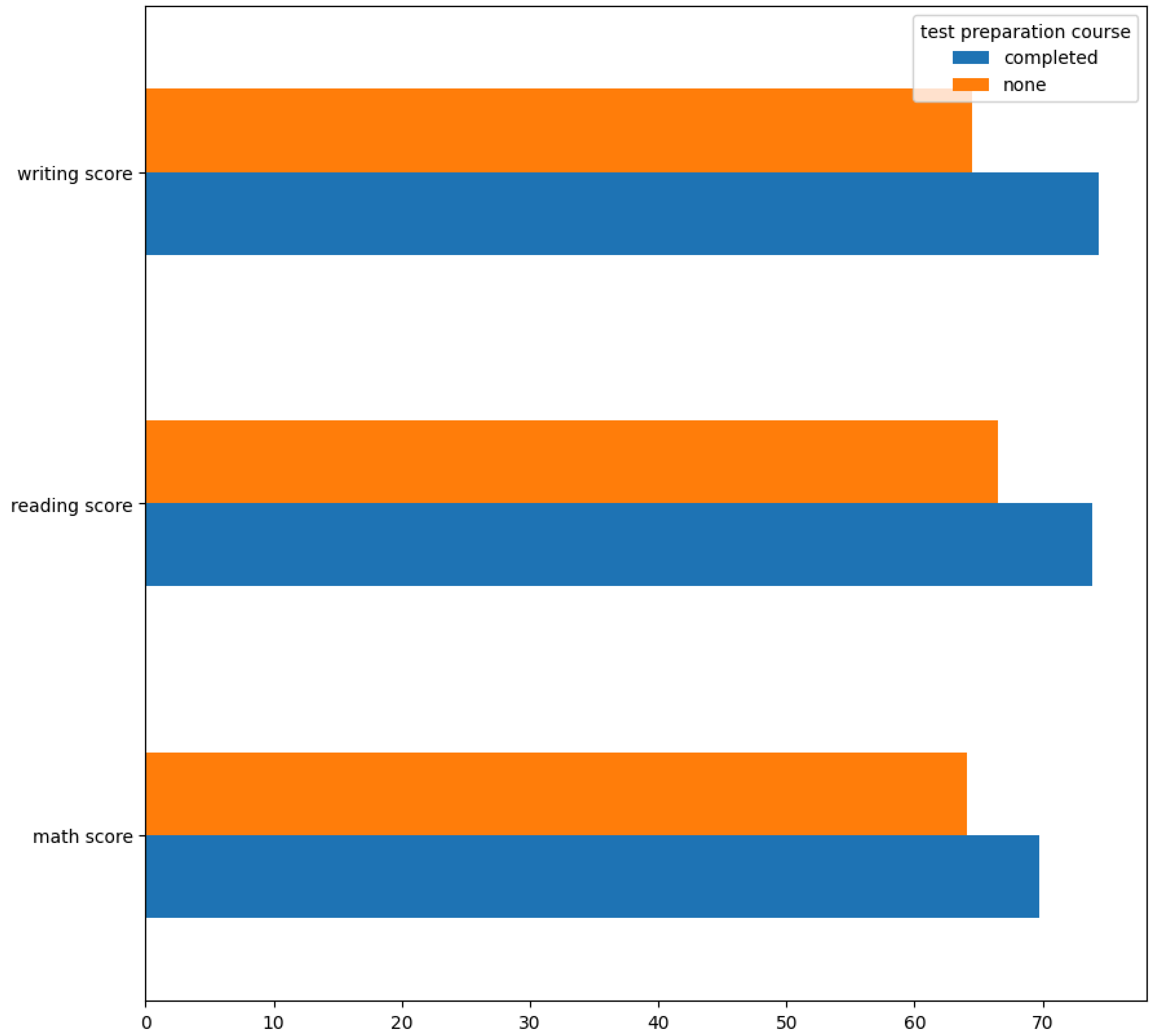


It is very clear from this graph that students whose parents are more educated than others (master's degree, bachelor's degree, and associate's degree) are performing better on average than students whose parents are less educated (high school). This can be a genetic difference, or simply a difference in the students' environment at home. More educated parents are more likely to push their students towards studies.

Secondly, let's look at the impact of the test preparation course on students' performance using a **horizontal bar graph**.

```
df.groupby('test preparation course')[['math score', 'reading score', 'writing score']].mean().T.plot(kind='barh', figsize=(10,10))
```

```
<Axes: >
```



Again, it is very clear that students who have completed the test preparation course have performed better, on average, as compared to students who have not opted for the course.

End Notes

In this article, we understood the meaning of Exploratory Data Analysis (EDA) with the help of an example dataset. We looked at how we can analyze the dataset, draw conclusions from the same, and form a hypothesis based on that.

Hope you like the article! Exploratory data analysis (EDA) is crucial in data science, enabling researchers to uncover insights through exploratory statistics and visualizing exploratory data effectively.