EDA - Exploratory Data Analysis :

Importing Libraries :

In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
```

Reading Data :

In [4]:
```
df = pd.read_csv("C:/users/amade/OneDrive/Área de Trabalho/Portfolio Projects/Used data/StudentsPerformance.csv")
df
```

Out[4]:

	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
...	...	...	...	...	...	...	...	...
995	female	group E	master's degree	standard	completed	88	99	95
996	male	group C	high school	free/reduced	none	62	55	55
997	female	group C	high school	free/reduced	completed	59	71	65
998	female	group D	some college	standard	completed	68	78	77
999	female	group D	some college	free/reduced	none	77	86	86

1000 rows × 8 columns

In [5]:
```
df.describe(include = "all")
```

Out[5]:

	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

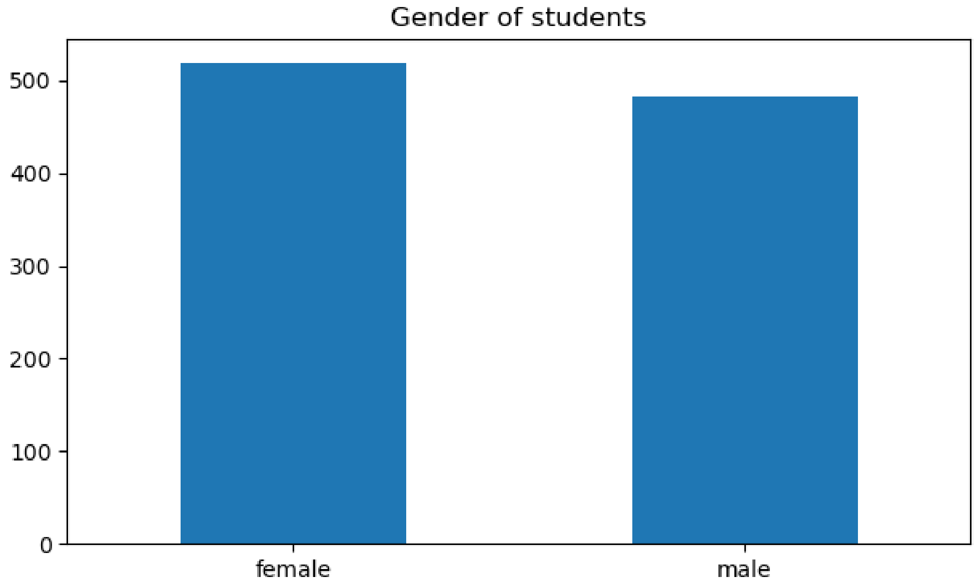
In [6]:
```
# Checking if theres any blank values :

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

Out[6]:
```
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
```

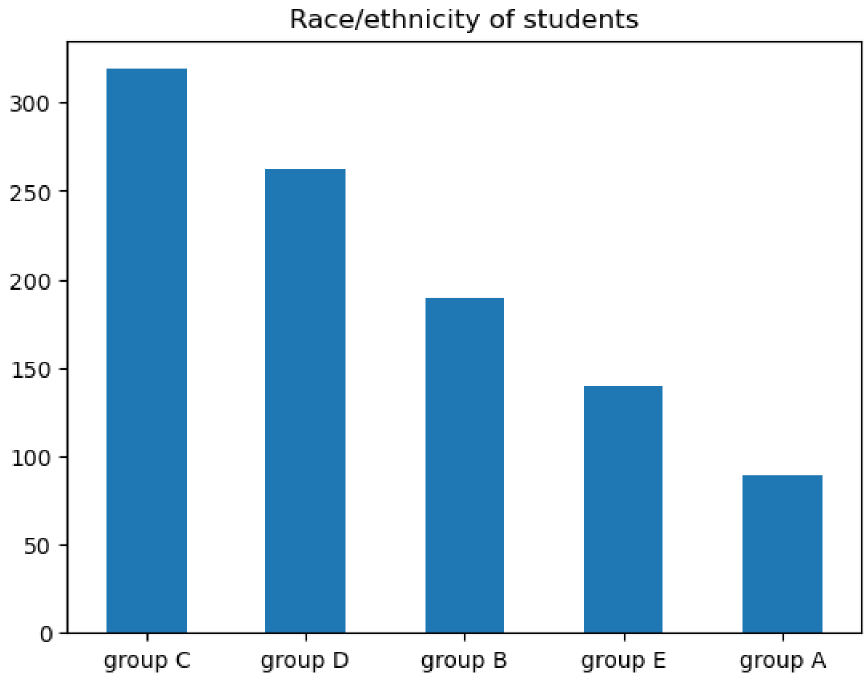
Graphical representation :

In [7]:
```
# Bar graphs :
```

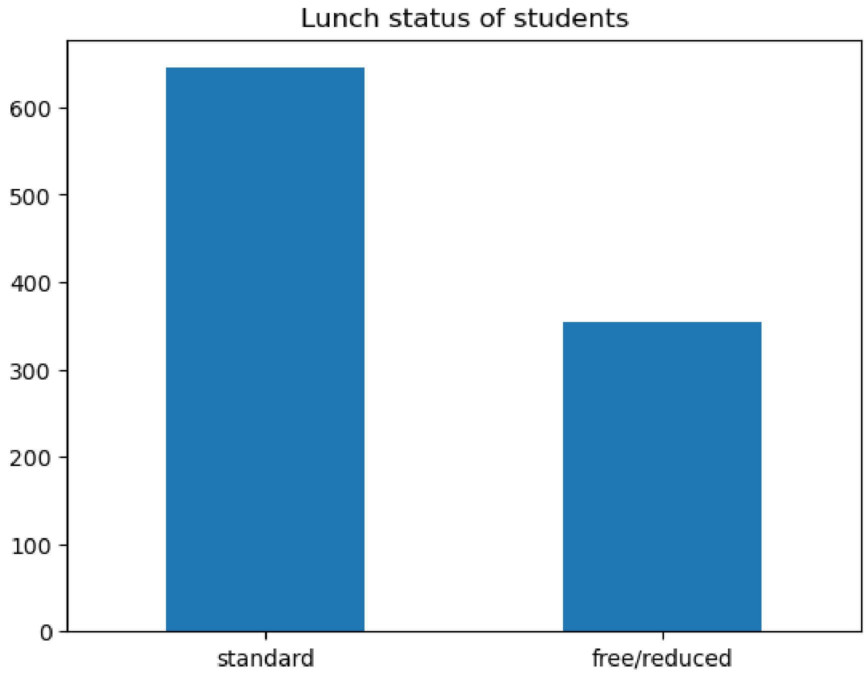
In [8]:
```
plt.subplot(221)

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

plt.xticks(rotation=0)

plt.show()
```



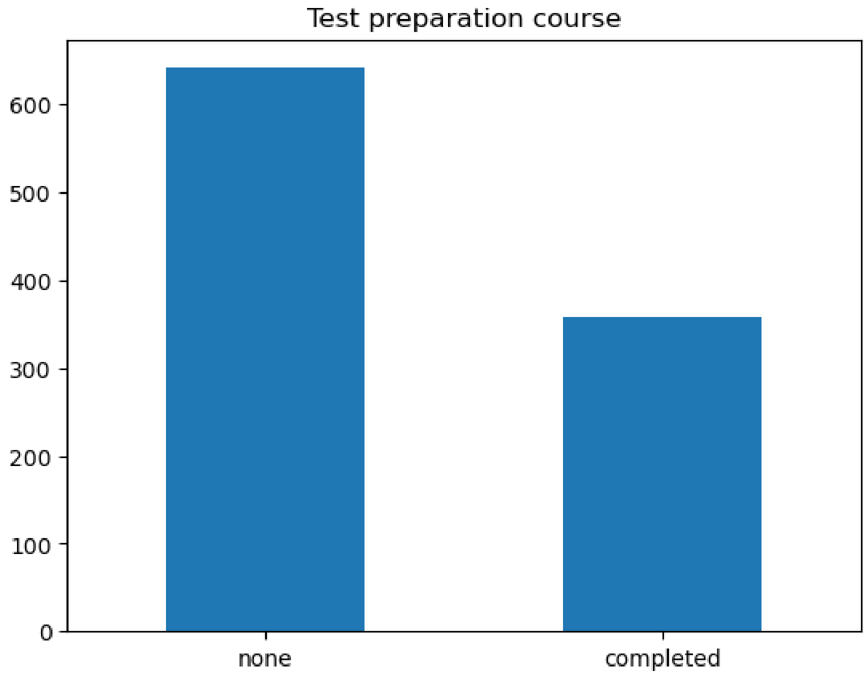
In [9]:
```
df["race/ethnicity"].value_counts().plot(kind = "bar", title = "Race/ethnicity of students")

plt.xticks(rotation=0)

plt.show()
```



In [10]:
```
df["lunch"].value_counts().plot(kind = "bar", title = "Lunch status of students")

plt.xticks(rotation=0)

plt.show()
```



In [11]:
```
df["test preparation course"].value_counts().plot(kind = "bar", title = "Test preparation course")

plt.xticks(rotation=0)

plt.show()
```



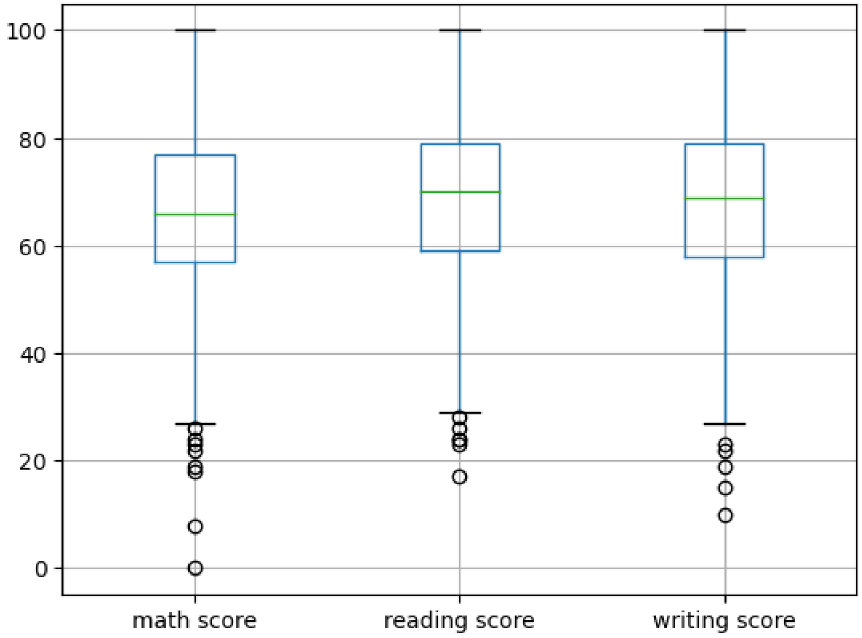
We can infer many things from the graph. There are more girls in the school than boys. The majority of the students belong to groups C and D. More than 60% of the students have a standard lunch at school. Also, more than 60% of students have not taken any test preparation course.

In [ ]:

In [12]:
```
# Boxplot :
```

In [13]:
```
df.boxplot()
```

Out[13]: <Axes: >



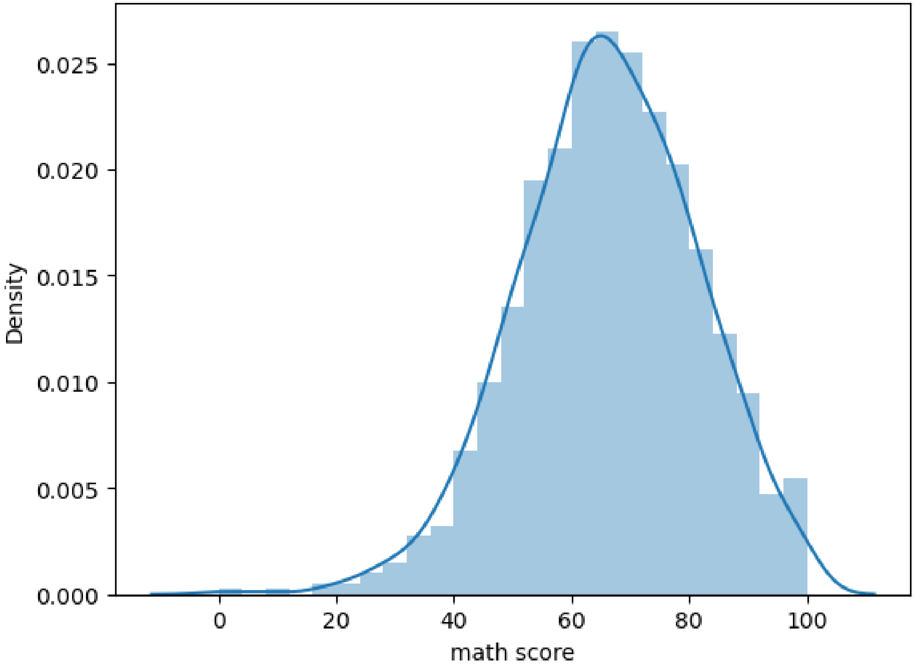
The middle portion represents the inter-quartile range (IQR). The horizontal green line in the middle represents the median of the data. The hollow circles near the tails represent outliers in the dataset. However, since it is very much possible for a student to score extremely low marks in a test, we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

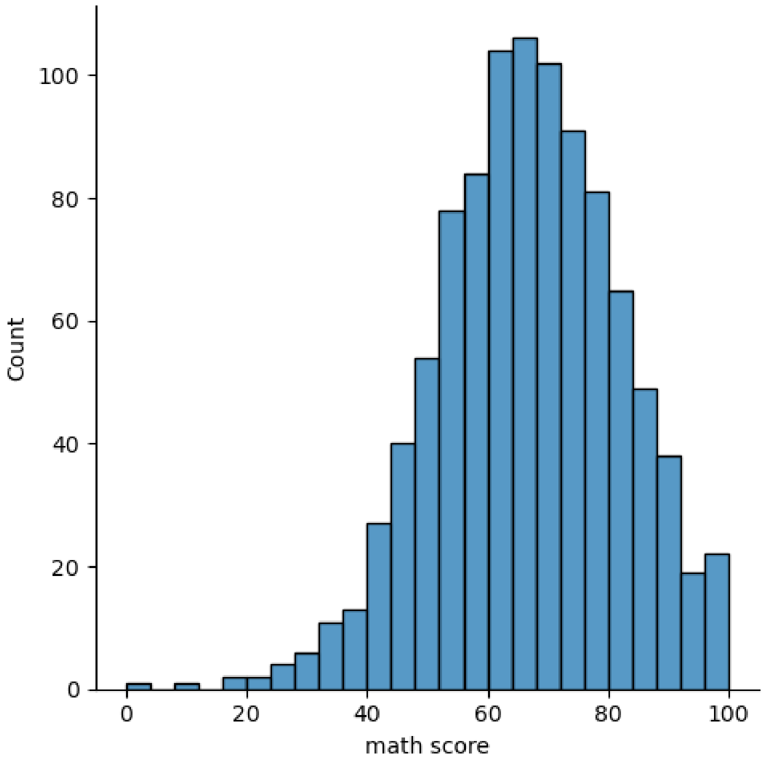
In [ ]:

In [17]:
```
# Distribution plot:

sns.distplot(df["math score"])

sns.displot(df["math score"])
```

```
C:\Users\amade\AppData\Local\Temp\ipykernel_11428\91820285.py:3: 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 (https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751)

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

Out[17]: <seaborn.axisgrid.FacetGrid at 0x1c582dd6620>





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.

In [ ]:

In [18]:
```
# Correlation map :

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

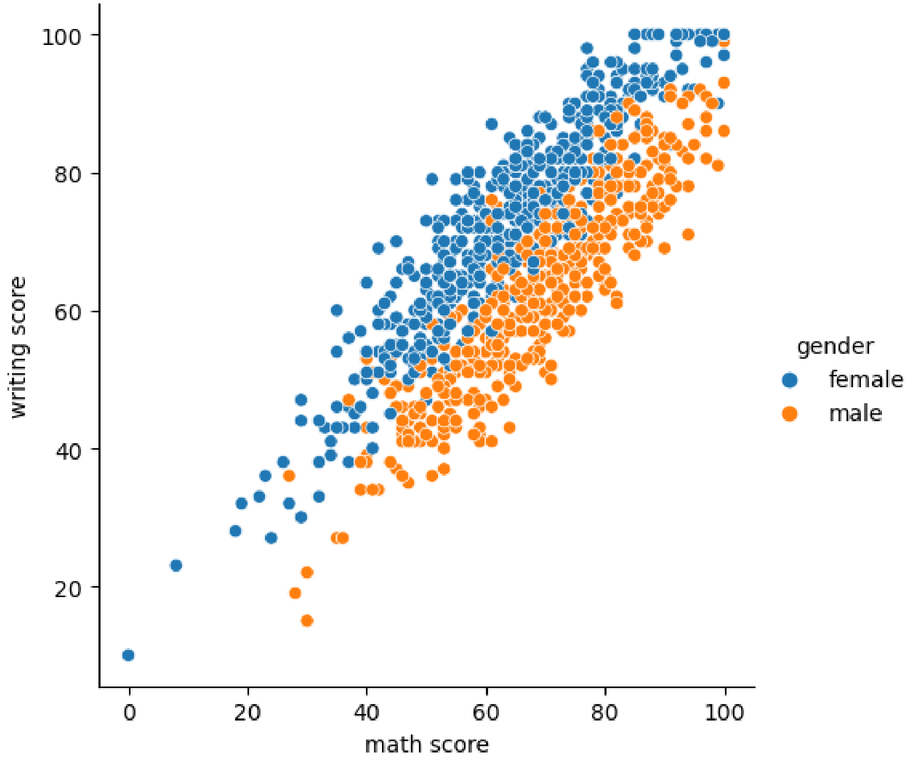
```
C:\Users\amade\AppData\Local\Temp\ipykernel_11428\2745810099.py:3: FutureWarning: The default value of numeric_only in DataFrame.corr is deprecated. In a future version, it will default to False. Select only valid columns or specify the value of numeric_only to silence this warning.
  corr = df.corr()
```

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.

In [ ]:

In [19]:
```
# Bivariate analysis :

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

Out[19]: `<seaborn.axisgrid.FacetGrid at 0x1c582df6470>`



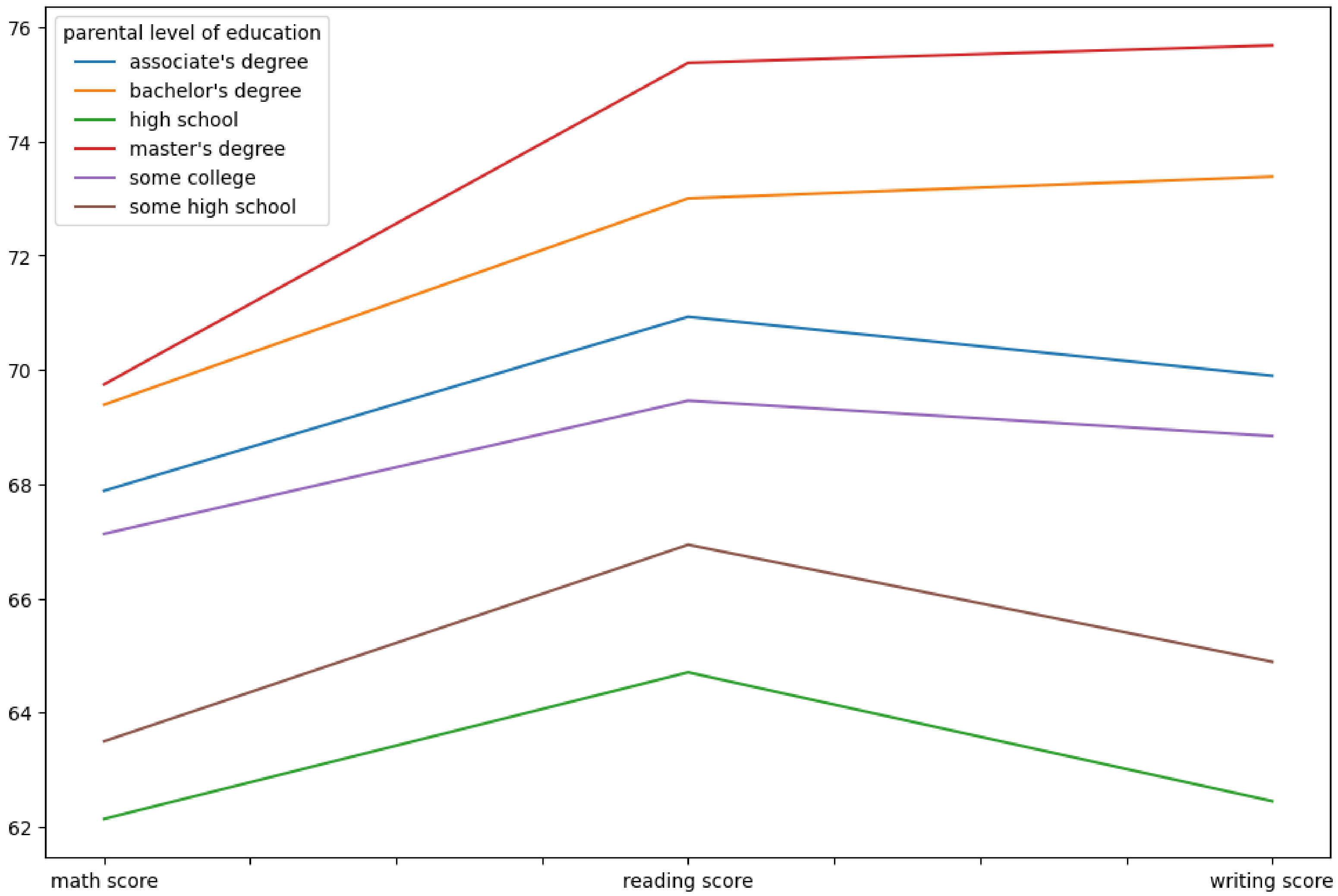
The graph shows a clear difference in scores between the male and female students. For the same math score, female students are more likely to have a higher writing score than male students. However, for the same writing score, male students are expected to have a higher math score than female students.

In [ ]:

In [20]:
```
# Line plot :
```

In [21]:
```
df.groupby('parental level of education')[['math score', 'reading score', 'writing score']].mean().T.plot(figsize=(12,8))
```

Out[21]: <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.

In [ ]:

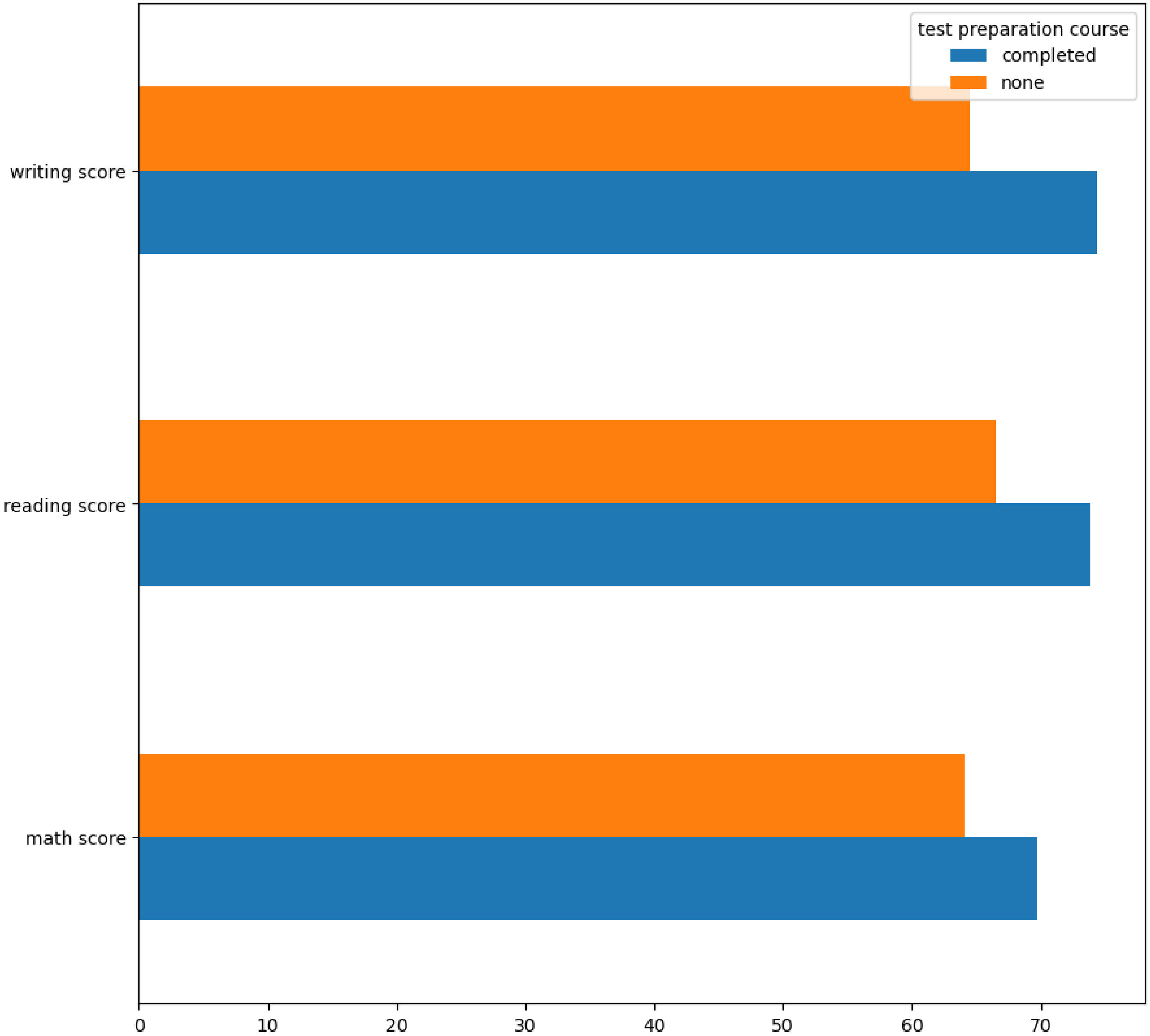
```

```

In [22]:

```
# Horizontal bar graph :
```

In [23]:
```
df.groupby('test preparation course')[['math score', 'reading score', 'writing score']].mean().T.plot(kind='barh',figsize=(10,10))
```

Out[23]:
```
<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.