MA 541-A: Statistical Methods

Spring 2022

Student: Gitanjali **Email:**g1@stevens.edu

Title: Students Performance in Exams

TABLE OF CONTENTS

Abstract:

Examination is a part of the education system; it is a process for testing the abilities or achievement of the student in any area of academic program and measures students progress. There are some factors, which helps to measure the performance of the student. The aim of this project is to understand the influence of "Gender, race/ethnicity, parental level of education, lunch and test preparation course on the student's performance. This project consists of several parts that includes data analysis, data exploration in Statistics. The project consists of a dataset that consist of 1000 observations and 8 columns which we can check using data.shape. In this project I am using python programming language to analyze the data also checking the data set using bar plots, box plots, count plots etc., correlation analysis, relationship between data. Performing various statistical techniques like comparing two groups, hypothesis testing, confidence intervals, Analysis of variance called ANOVA, categorical data analysis, logistic regression, linear regression, lasso regression and model accuracy.

Chapter 1: Introduction

In this project we will find the interesting inferences from Students Performance dataset. The objective is to understand how the students' performance/test scores is affected by the other variables/columns (Gender, Ethnicity, Parental level of education, Lunch, Test preparation course). With these factors or columns, we can identify what factor must be consider when enhancing the performance of our students in the future. The dataset contains 1000 observations of student grades in math, reading and writing. We examine a number of factors that relate to student performance in exams. In particular, we wish to understand the association between Student gender and scores obtained in various courses, as well as the test preparation course and lunch on the exam scores.

Goal: choose appropriate statistical techniques and provide evidence that the suggested methods are suitable for the chosen data. Used hypothesis test to check whether the scores of all three scores are same

In this project I will address some questions that interested me using statistical methodology. After looking at the columns of our dataset I wanted to test how effective the test preparation course is, which major factors contribute to test outcomes, what is the effect of lunch type in the student performance and is there any relation between the data or groups.

we can learn relationships and structure from the dataset. compare different groups, perform one-way ANOVA to test the null hypothesis that two or more groups have the same population mean and testing the correlation between categorical variables like gender, parental level of education, test preparation course. Test whether the categorical dataset is correlated to math score, reading and writing scores and then find the model accuracy using various regression models where I used 3 predictors (math, reading and writing scores) and "gender" as the response.

Chapter 2: Methodology

The methods, practices, processes, techniques, procedures, and rules used in this project are as follows:

2.1 Hypothesis Testing

Hypothesis testing is a key concept in statistics because it gives statistical evidence to show the validity of the study.

- H0(null hypothesis): the variables are independent, there is no relationship between the two categorical variables. Knowing the value of one variable does not help to predict the value of the other variable.
- H1(alternative hypothesis): the variables are dependent, there is a relationship between the two categorical variables. Knowing the value of one variable helps to predict the value of the other variable

As we have three different subjects, I will be testing some hypothesis of each course based on the gender. (More details can be found in chapter 4 analysis and results)

2.2 ANOVA (Analysis of Variance)

Anova is used to check if there is a difference between means of three or more groups, unlike t-test which is only capable of examining two groups. Because of that, we need a categorical variable which distinct values are more than two.

In this project we are using One Way ANOVA The one-way ANOVA tests the null hypothesis that two or more groups have the same population mean. The test is applied to samples from two or more groups, possibly with differing sizes. Here we are testing the correlation between categorical variables and test scores using one-Way ANOVA

H0: There is no difference between groups and equality between means H1: There is a difference between the means and groups.

data = Data frame and variable = Categorical columns like gender, parental level of education, test preparation course are used for one-way ANOVA test

2.3 Categorical Data Analysis

Independence tests are used to determine if there is a significant relationship between two categorical variables. There exist two different types of independence test:

- the Chi-square test (the most common)
- the Fisher's exact test

The Chi-square test is used when the sample is large enough. On the other hand, the Fisher's exact test is used when the sample is small.

The Chi-square test of independence tests whether there is a relationship between two categorical variables/columns in the dataset. In this project, we are going to test in Python if there is a relationship between the two categorical variables.

The formula for the Chi-square test is as follows: $\chi^2 = \sum \frac{(O_i - E_i)^2}{E_i}$

Were, χ^2 = chi-square, O_i = observed value, E_i = expected value

The Contingency Table: A Contingency table is used in statistics to summarize the relationship between several categorical variables. Here, we take a table that shows the number of male and female taking test preparation course. We can verify the hypothesis by these methods:

Using **p-value**:

The **significance factor** is used to determine whether the relation between the variables is of considerable significance. Significance factor/alpha value of **0.05** is chosen. This *alpha value* denotes the probability of erroneously rejecting **H0** when it is true. A lower *alpha value* is chosen in cases where we expect more precision. If the **p-value** for the test comes out to be strictly greater than the alpha value, then H0 holds true.

Using **chi-square** value:

If our calculated value of chi-square is less or equal to the critical value of chi-square, then **H0** holds true.

degrees of freedom: (rows - 1) * (cols - 1)

2.4 Logistic Regression

Logistic regression models a relationship between predictor variables and a categorical response variable. In our project the gender will be the response and writing reading and math score will be the 3 predictors.

2.5 Linear Regression

Linear regression is the standard algorithm for regression that assumes a linear relationship between inputs and the target variable. A linear regression is where the relationships between the variables can be described with a straight line. A scatter plot is used in this project to plot the actual and predicted values to test for linearity. Linear regression has two primary purposes—understanding the relationships between variables and forecasting.

- The coefficients represent the estimated magnitude and direction (positive/negative) of the relationship between each independent variable and the dependent variable.
- A linear regression equation allows you to predict the mean value of the dependent variable given values of the independent variables that you specify.

2.6 Multicollinearity, Lasso Regression and Ridge regression

Multicollinearity, or collinearity, is the existence of near-linear relationships among the independent variables.

Lasso regression is like linear regression, the lasso shrinks the coefficient estimates towards zero. This type of regression is used when the dataset shows high multicollinearity or when you want to automate variable elimination and feature selection**.**

Ridge Regression is a technique for analyzing multiple regression data that suffer from multicollinearity. When the issue of multicollinearity occurs, least-squares are unbiased, and variances are large, this results in predicted values being far away from the actual value.

Chapter 3: Data Description

The data is about the students' performance in exam. The source of the dataset used is "Kaggle". The dataset has information about 1000 students' performance. This dataset has 1000 rows and 8 columns which include the following fields.

- Gender: male, female
- race/ethnicity: Group A, B, C, D
- Parental level of education: bachelor's degree, master's degree, some college, high school
- Lunch: Standard, free/reduced
- Test preparation: none, completed
- Math score
- Reading score
- Writing score

The purpose of this dataset is to analyze the test scores of the students based on their test preparation course, parental level of education, race/ethnicity, lunch, gender. Also comparing the relationships between the data and groups. Checking if each group is independent of each other or not.

#Chapter 4: Analysis and Results

Importing Libraries

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

Loading the Dataset

In [2]:
```
data =pd.read_csv('StudentsPerformance.csv')
```

Exploring dataset

In [3]:
```
print(data)
```

```
     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
..      ...            ...                         ...           ...
995  female        group E             master's degree      standard
996    male        group C                 high school  free/reduced
997  female        group C                 high school  free/reduced
998  female        group D                some college      standard
999  female        group D                some college  free/reduced

    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
..                      ...         ...            ...            ...
995               completed          88             99             95
996                    none          62             55             55
997               completed          59             71             65
998               completed          68             78             77
999                    none          77             86             86

[1000 rows x 8 columns]
```

In [4]:
```
#Checking missing values
data.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
```

there are no null values in our dataset

In [5]:
```
#Size of the data frame. Tells us the number of rows and columns
data.shape
```

Out[5]:
```
(1000, 8)
```

There are 1000 observations in our dataset and have 8 columns

In [6]:
```
data.head()
```

Out[6]:

	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 [7]:
```
data.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]:
```
#Gives the information about the data like Count, minimum,maximum, mean and stand
data.describe()
```

Out[8]:

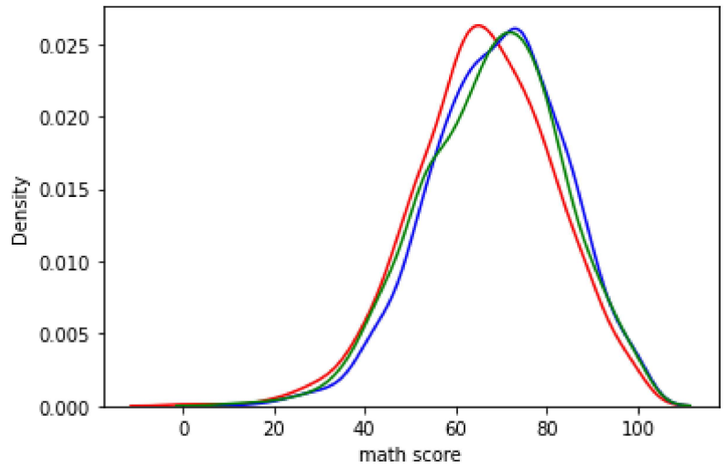
	math score	reading score	writing score
count	1000.00000	1000.000000	1000.000000
mean	66.08900	69.169000	68.054000
std	15.16308	14.600192	15.195657
min	0.00000	17.000000	10.000000
25%	57.00000	59.000000	57.750000
50%	66.00000	70.000000	69.000000
75%	77.00000	79.000000	79.000000
max	100.00000	100.000000	100.000000

#Data Visualization

Math, Reading and Writing score kdeplot

In [9]:
```
sns.kdeplot(data['math score'],color='red')
sns.kdeplot(data['reading score'],color='blue')
sns.kdeplot(data['writing score'],color='green')
```

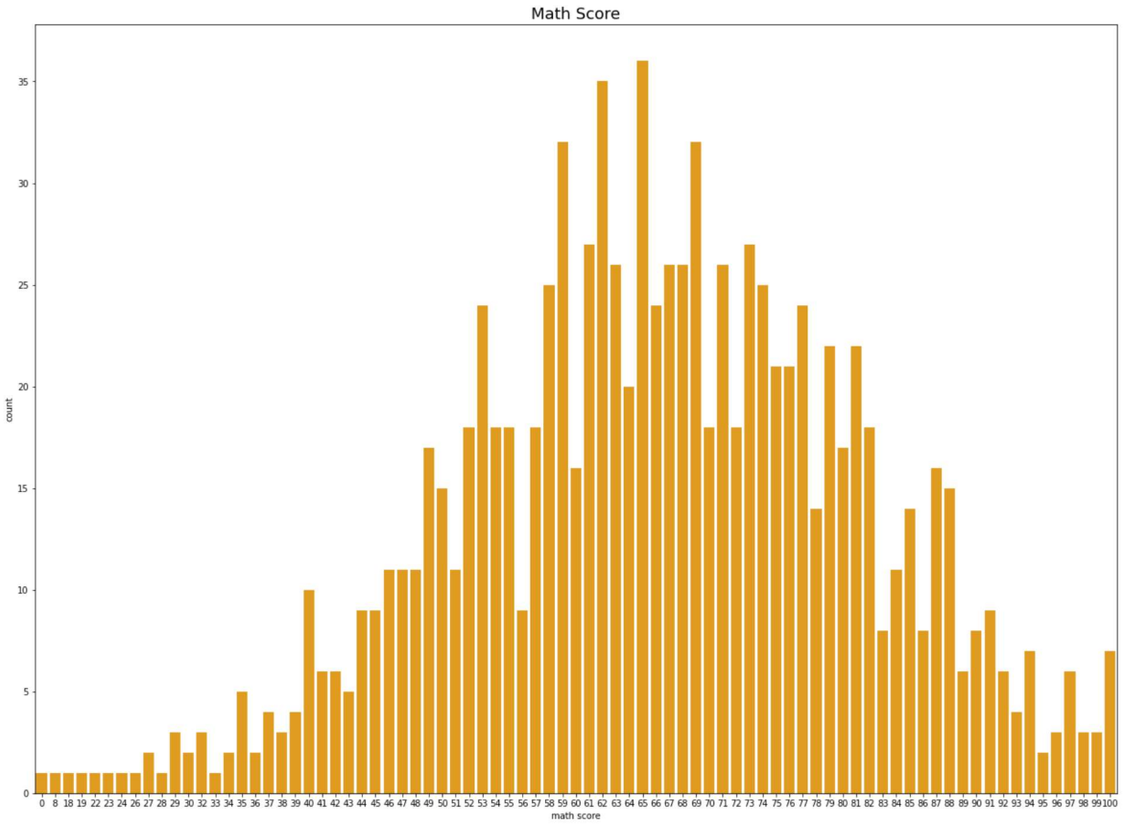
Out[9]: <matplotlib.axes._subplots.AxesSubplot at 0x7fc27bf33450>



Count plots of all the 3 courses overall scores

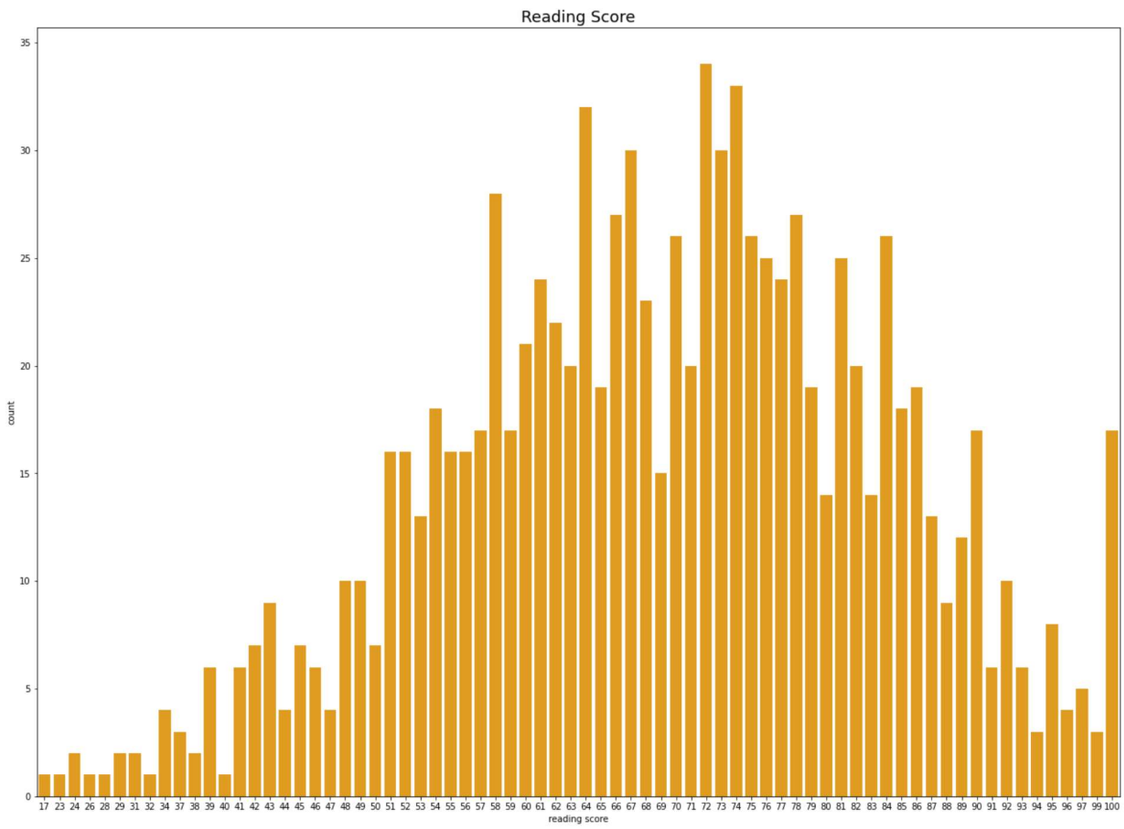
In [21]:
```
plt.figure(figsize=(22,16))
sns.countplot(data['math score'],color='orange')
plt.title('Math Score',fontsize = 18)
plt.show()
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarnin
g: Pass the following variable as a keyword arg: x. From version 0.12, the only
valid positional argument will be `data`, and passing other arguments without a
n explicit keyword will result in an error or misinterpretation.
  FutureWarning
```



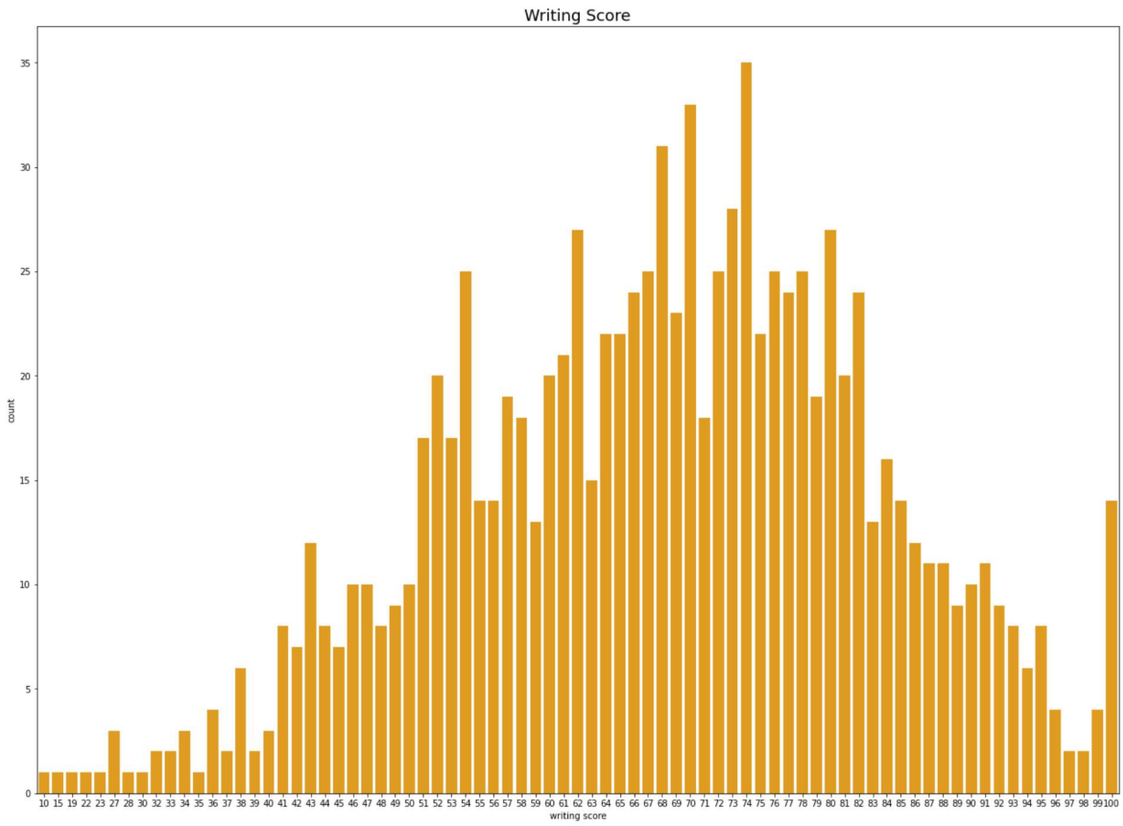
In [20]:
```
plt.figure(figsize=(22,16))
sns.countplot(data['reading score'],color='orange')
plt.title('Reading Score',fontsize = 18)
plt.show()
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarnin
g: Pass the following variable as a keyword arg: x. From version 0.12, the only
valid positional argument will be `data`, and passing other arguments without a
n explicit keyword will result in an error or misinterpretation.
  FutureWarning
```



In [23]:
```
plt.figure(figsize=(22,16))
sns.countplot(data['writing score'],color='orange')
plt.title('Writing Score',fontsize = 18)
plt.show()
```

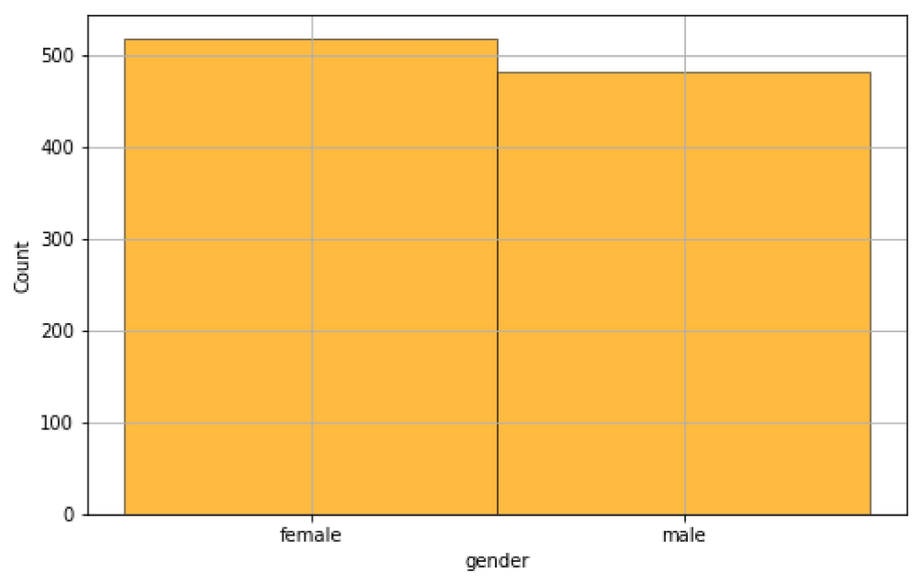
```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarnin
g: Pass the following variable as a keyword arg: x. From version 0.12, the only
valid positional argument will be `data`, and passing other arguments without a
n explicit keyword will result in an error or misinterpretation.
  FutureWarning
```



Total number of male and female students

In [24]:
```
plt.figure(figsize=(8,5))
sns.histplot(data,color='orange',x = 'gender',linestyle='-',linewidth='0.5')
plt.grid()
data['gender'].value_counts()
```

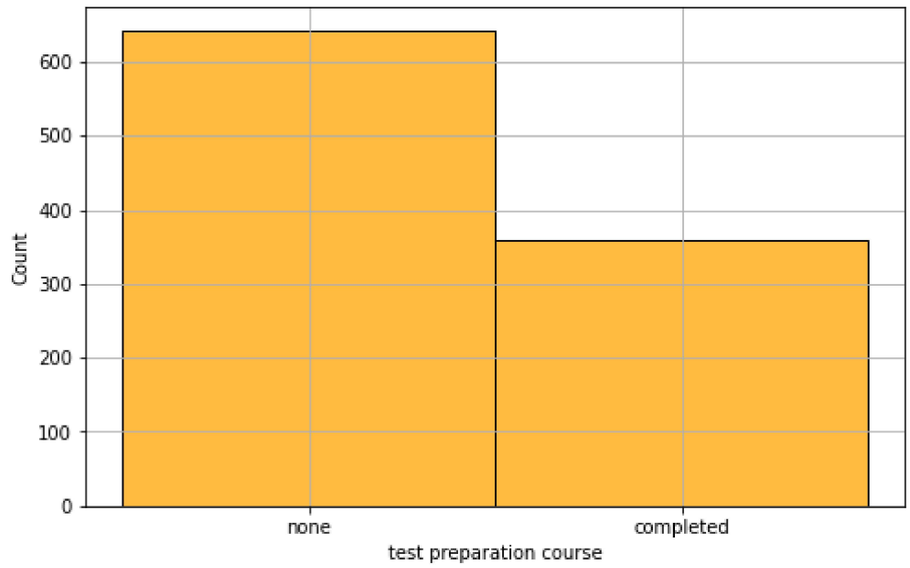
Out[24]:
```
female    518
male      482
Name: gender, dtype: int64
```



Test preparation course data: total number of test course completed and not taken/none

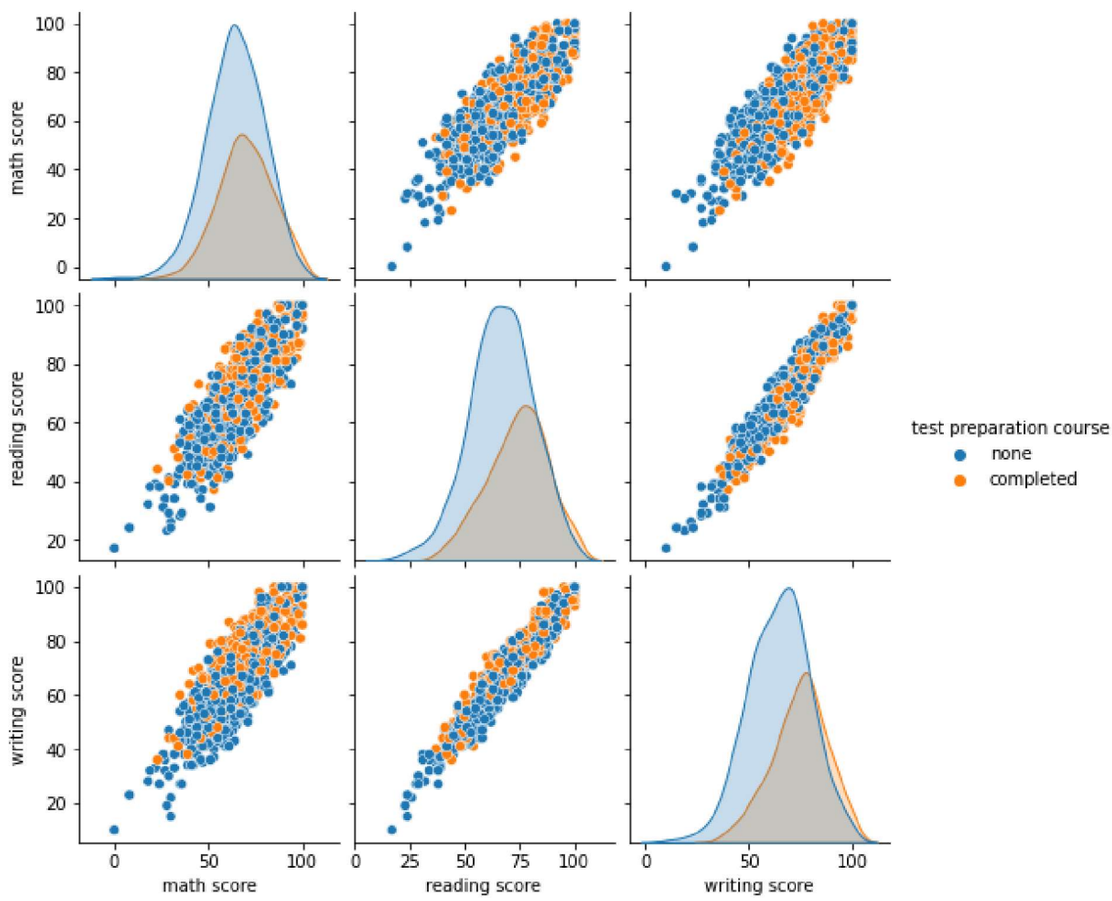
In [25]:
```
plt.figure(figsize=(8,5))
sns.histplot(data,color='orange',x = 'test preparation course')
plt.grid()
data['test preparation course'].value_counts()
```

Out[25]:
```
none         642
completed    358
Name: test preparation course, dtype: int64
```



In [26]:
```
sns.pairplot(data,hue='test preparation course', vars=['math score','reading scor
```

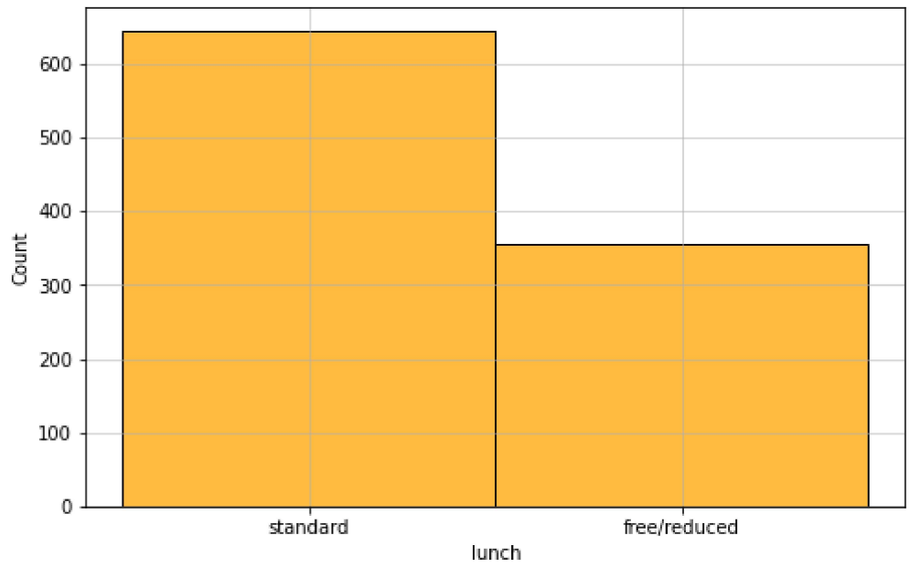
Out[26]:
```
<seaborn.axisgrid.PairGrid at 0x7fc2791b6250>
```



Total number of lunch: Standard and Free/reduced

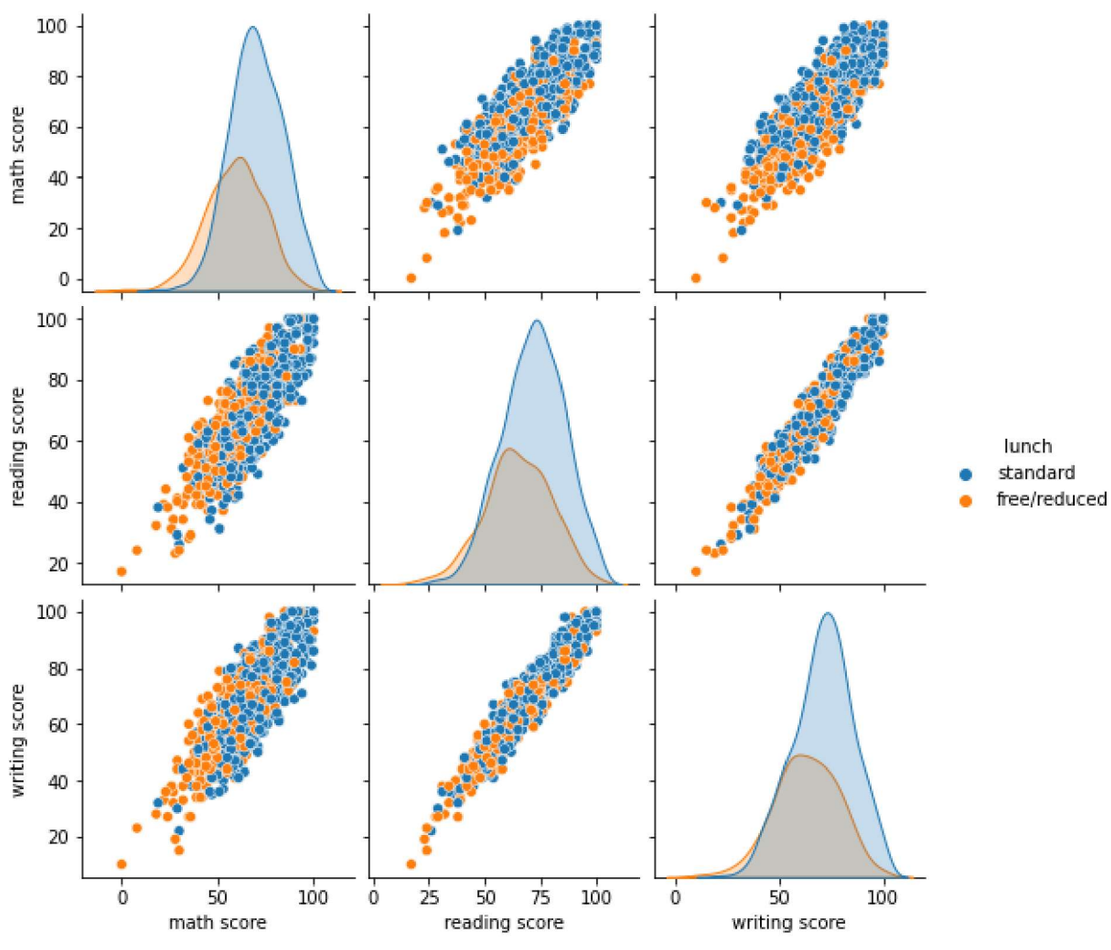
In [27]:
```
plt.figure(figsize=(8,5))
sns.histplot(data,color='orange',x = 'lunch')
plt.grid(linestyle='-',linewidth='0.5')
data['lunch'].value_counts()
```

Out[27]:
```
standard        645
free/reduced    355
Name: lunch, dtype: int64
```



In [28]:
```
sns.pairplot(data, hue='lunch', vars=['math score','reading score','writing score
```

Out[28]:
```
<seaborn.axisgrid.PairGrid at 0x7fc278898350>
```

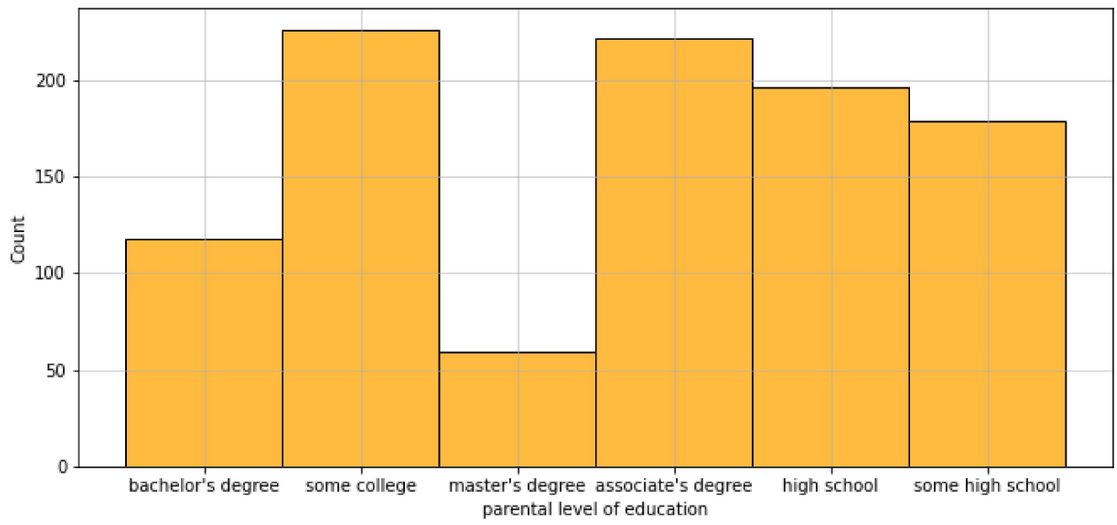


From the above graph we can see that student who take standard lunch score higher than the student who take free/reduced lunch.

Parental level of education

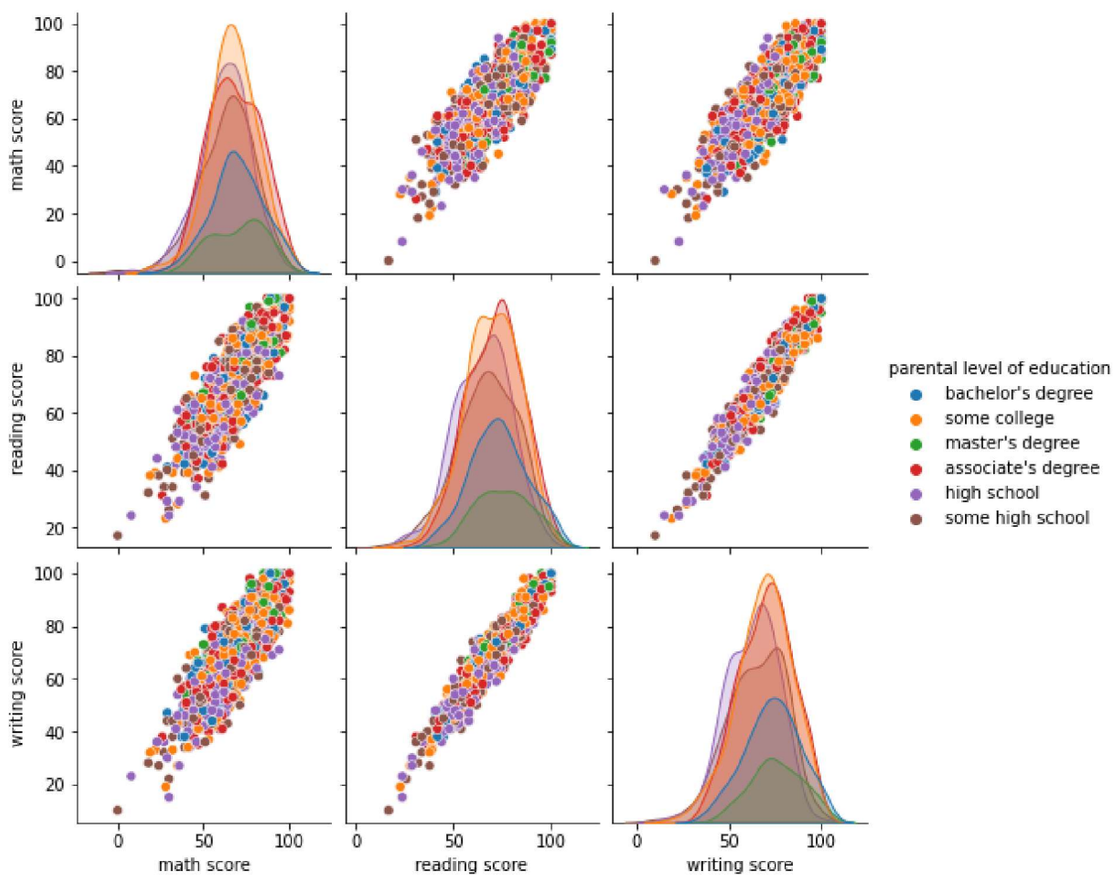
In [29]:
```
plt.figure(figsize=(11,5))
sns.histplot(data,color='orange',x = 'parental level of education')
plt.grid(linestyle='-',linewidth='0.5')
data['parental level of education'].value_counts()
```

Out[29]:
```
some college          226
associate's degree    222
high school           196
some high school      179
bachelor's degree     118
master's degree        59
Name: parental level of education, dtype: int64
```



In [30]:
```
sns.pairplot(data, hue='parental level of education', vars=['math score','reading
```

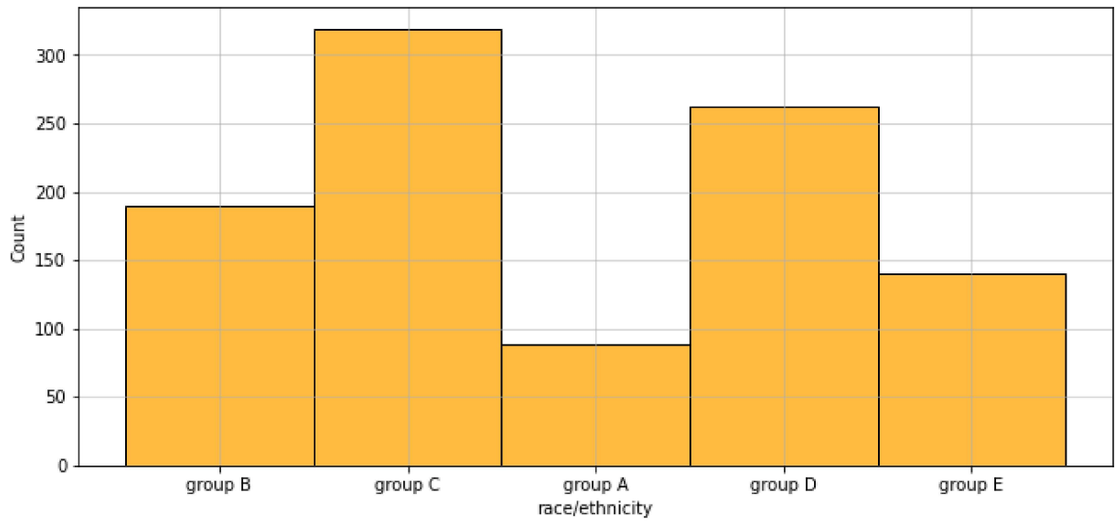
Out[30]:
```
<seaborn.axisgrid.PairGrid at 0x7fc278a4c950>
```



Scores by race/ethnicity

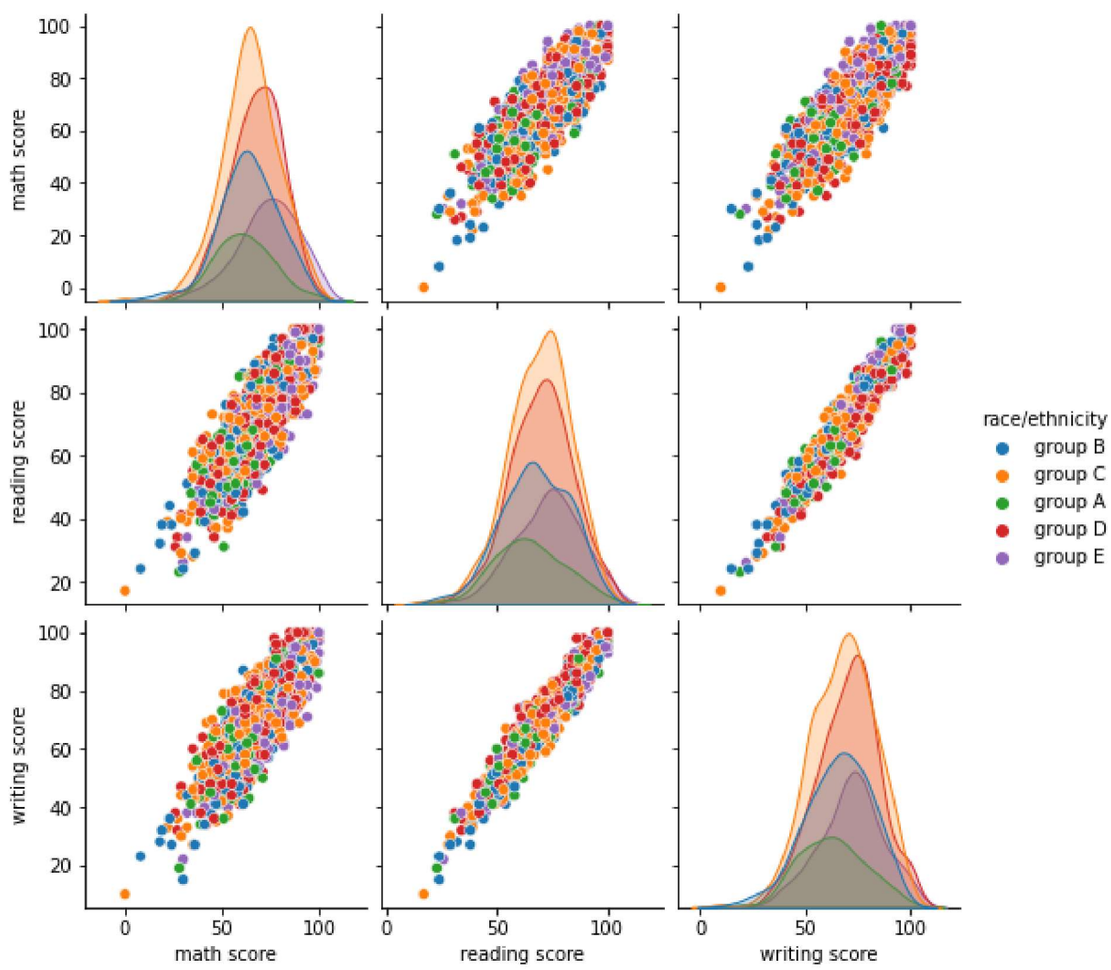
In [31]:
```
plt.figure(figsize=(11,5))
sns.histplot(data,color='orange',x = 'race/ethnicity')
plt.grid(linestyle='-',linewidth='0.5')
data['race/ethnicity'].value_counts()
```

Out[31]:
```
group C    319
group D    262
group B    190
group E    140
group A     89
Name: race/ethnicity, dtype: int64
```



In [32]:
```
sns.pairplot(data, hue='race/ethnicity')
```

Out[32]:
```
<seaborn.axisgrid.PairGrid at 0x7fc274397d10>
```



In [33]:
```
data.groupby(['race/ethnicity','parental level of education']).mean()
```

Out[33]:

race/ethnicity	parental level of education	math score	reading score	writing score
group A	associate's degree	61.000000	67.071429	63.571429
	bachelor's degree	67.166667	68.083333	68.333333
	high school	60.444444	62.888889	60.500000
	master's degree	57.666667	64.666667	67.666667
	some college	63.888889	65.777778	65.000000
	some high school	58.916667	62.083333	58.583333
group B	associate's degree	66.097561	69.585366	68.243902
	bachelor's degree	69.300000	72.950000	71.650000
	high school	59.791667	63.458333	61.250000
	master's degree	67.166667	80.166667	77.166667
	some college	63.189189	65.756757	64.189189
	some high school	61.815789	66.447368	64.605263
group C	associate's degree	66.730769	71.128205	70.269231
	bachelor's degree	68.150000	75.675000	75.900000
	high school	60.906250	64.421875	61.656250
	master's degree	67.052632	70.526316	69.526316
	some college	65.130435	69.420290	68.869565
	some high school	60.551020	65.632653	63.285714
group D	associate's degree	67.600000	70.540000	69.860000
	bachelor's degree	67.571429	70.142857	71.892857
	high school	62.863636	64.409091	63.159091
	master's degree	72.521739	77.173913	79.739130
	some college	68.731343	70.880597	71.701493
	some high school	66.760000	69.980000	69.100000
group E	associate's degree	74.897436	73.820513	73.205128
	bachelor's degree	76.555556	74.833333	75.388889
	high school	70.772727	70.318182	67.545455
	master's degree	74.625000	82.125000	80.500000
	some college	73.828571	72.628571	70.200000
	some high school	72.111111	69.555556	66.555556

In [34]:
```
data.groupby(['race/ethnicity','parental level of education','test preparation co
```

Out[34]:

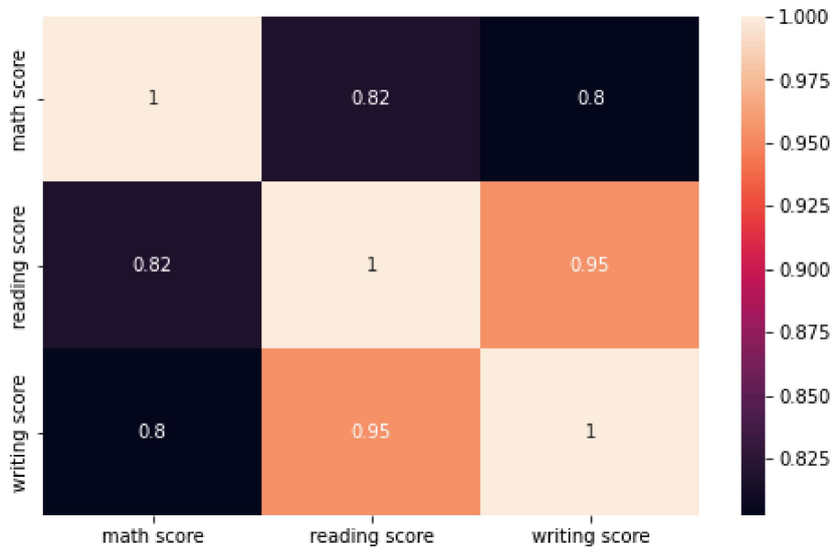
race/ethnicity	parental level of education	test preparation course	lunch	gender	math score	reading score	writing score
group A	associate's degree	completed	free/reduced	male	59.500000	68.500000	67.50
			standard	female	60.000000	67.500000	68.00
				male	97.000000	92.000000	86.00
		none	free/reduced	female	47.666667	64.333333	60.00
				male	54.500000	59.000000	49.50
...	...	...	...	...	...	...	...
group E	some high school	completed	standard	female	80.000000	85.000000	85.00
				male	79.333333	72.333333	70.50
			free/reduced	female	54.000000	59.000000	58.00
		none	standard	female	77.000000	79.000000	80.00
				male	74.500000	67.000000	59.75

211 rows × 3 columns

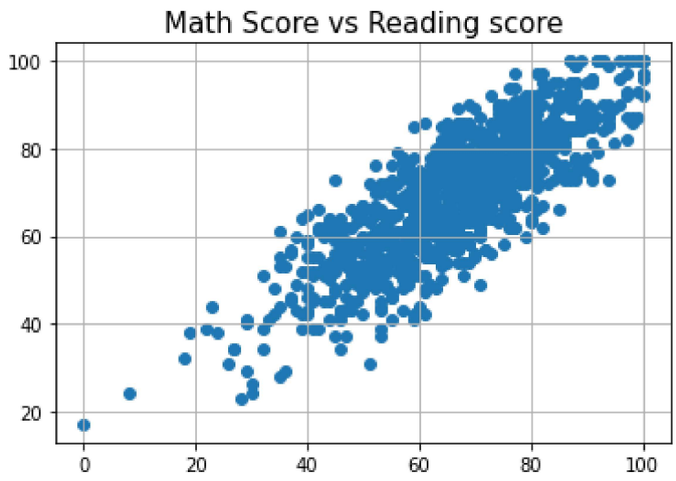
#Correlation Analysis

In [42]:
```
#Correlation between data
data.corr()
plt.figure(figsize=(8,5))
sns.heatmap(data = data.corr(), annot=True)
```

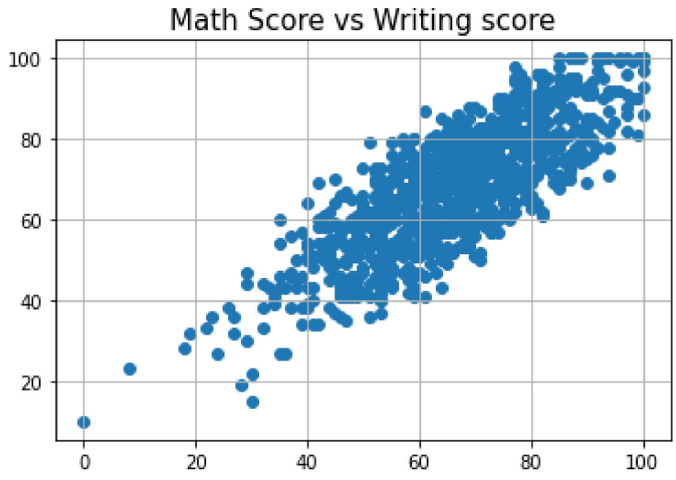
Out[42]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7fc273c346d0>
```



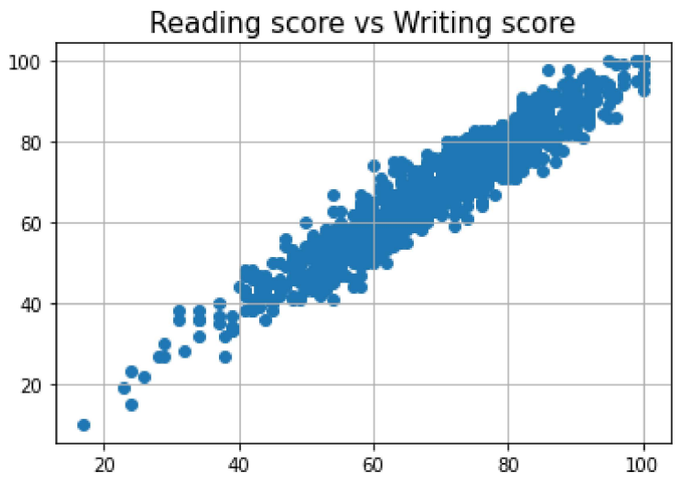
In [45]:
```
fig, ax= plt.subplots()
scatter= ax.scatter(x=data['math score'], y=data['reading score'])
plt.title('Math Score vs Reading score',fontsize = 15)
plt.grid()
```



In [46]:
```
fig, ax= plt.subplots()
scatter= ax.scatter(x=data['math score'], y=data['writing score'])
plt.title('Math Score vs Writing score',fontsize = 15)
plt.grid()
```



In [47]:
```
fig, ax= plt.subplots()
scatter= ax.scatter(x=data['reading score'], y=data['writing score'])
plt.title('Reading score vs Writing score',fontsize = 15)
plt.grid()
```



#Comparing different groups

Here we are checking various factors that affect student performance

Finding the average score of the three courses

In [48]:
```
d= data["math score"]+data["reading score"]+data["writing score"]
total_score=d
avg_score=round(d)/3
avg_score
data['avg_score']=avg_score
```

In [49]:
```
data.head()
```

Out[49]:

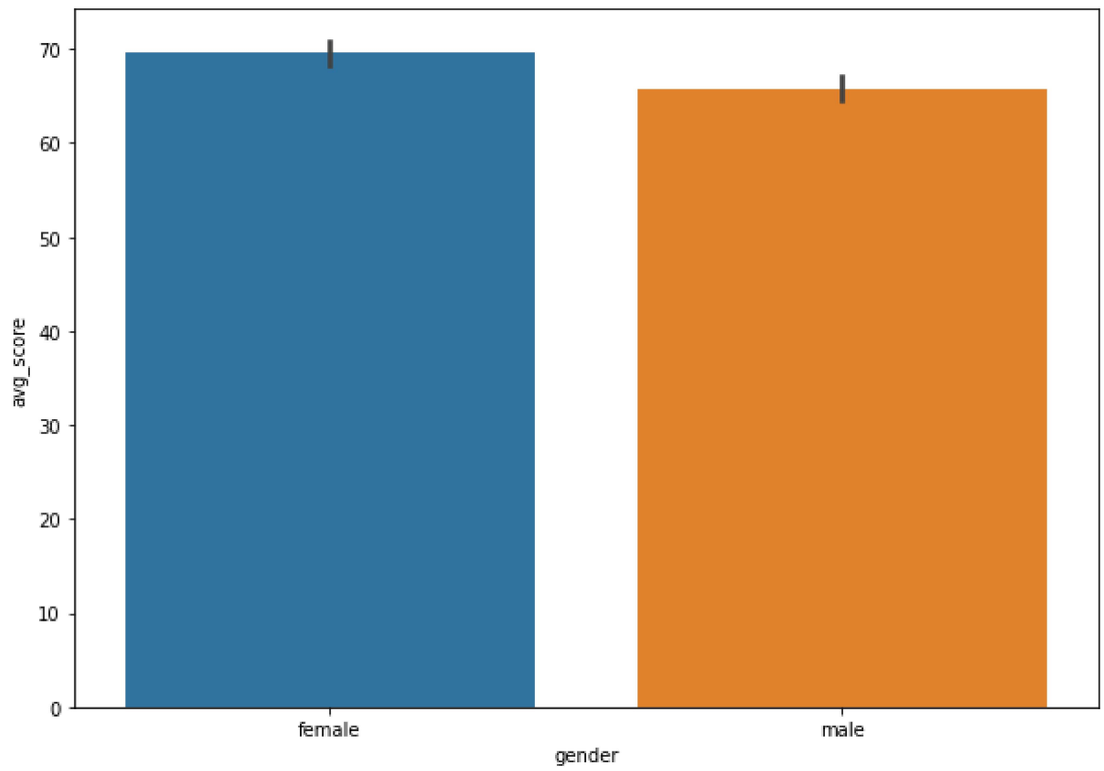
	gender	race/ethnicity	parental level of education	lunch	test preparation course	math score	reading score	writing score	avg_score
0	female	group B	bachelor's degree	standard	none	72	72	74	72.666667
1	female	group C	some college	standard	completed	69	90	88	82.333333
2	female	group B	master's degree	standard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

Relation between the student's score and gender

In [50]:
```
plt.figure(figsize=(10,7))
sns.barplot(x='gender',y='avg_score',data=data)
data.groupby('gender').mean()
```

Out[50]:

gender	math score	reading score	writing score	avg_score
female	63.633205	72.608108	72.467181	69.569498
male	68.728216	65.473029	63.311203	65.837483



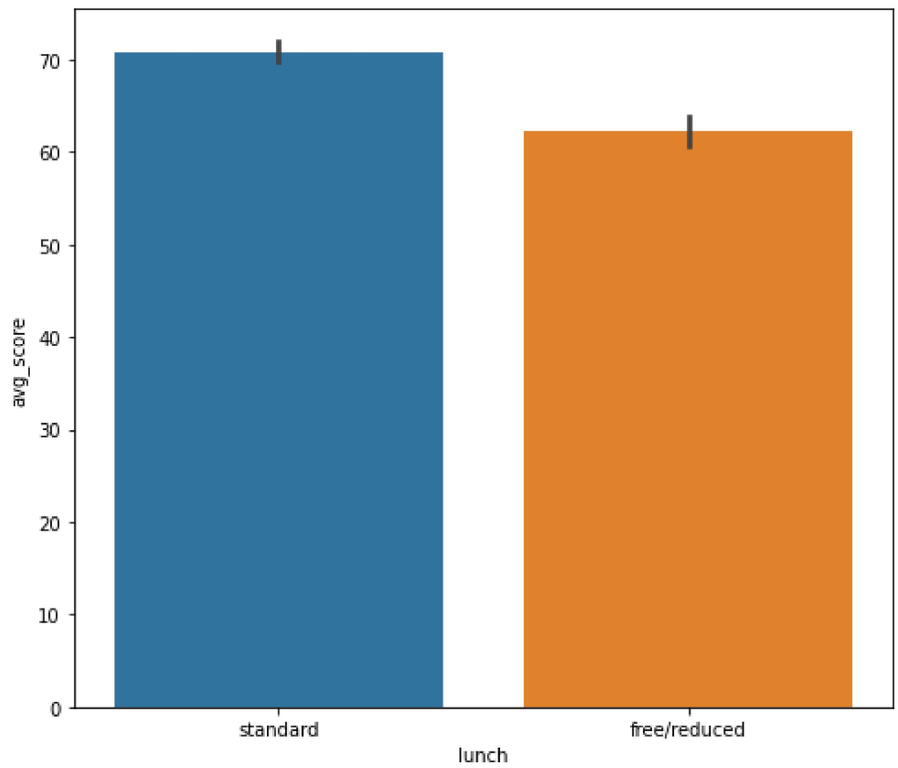
From above we can see that female average score is greater than male student. Male student performed better in math whereas female students scored less in math but have greater score in reading and writing.

Relation between the student's score and their lunch type

In [51]:
```
plt.figure(figsize=(8,7))
sns.barplot(x='lunch',y='avg_score',data=data)
data.groupby('lunch').mean()
```

Out[51]:

lunch	math score	reading score	writing score	avg_score
free/reduced	58.921127	64.653521	63.022535	62.199061
standard	70.034109	71.654264	70.823256	70.837209



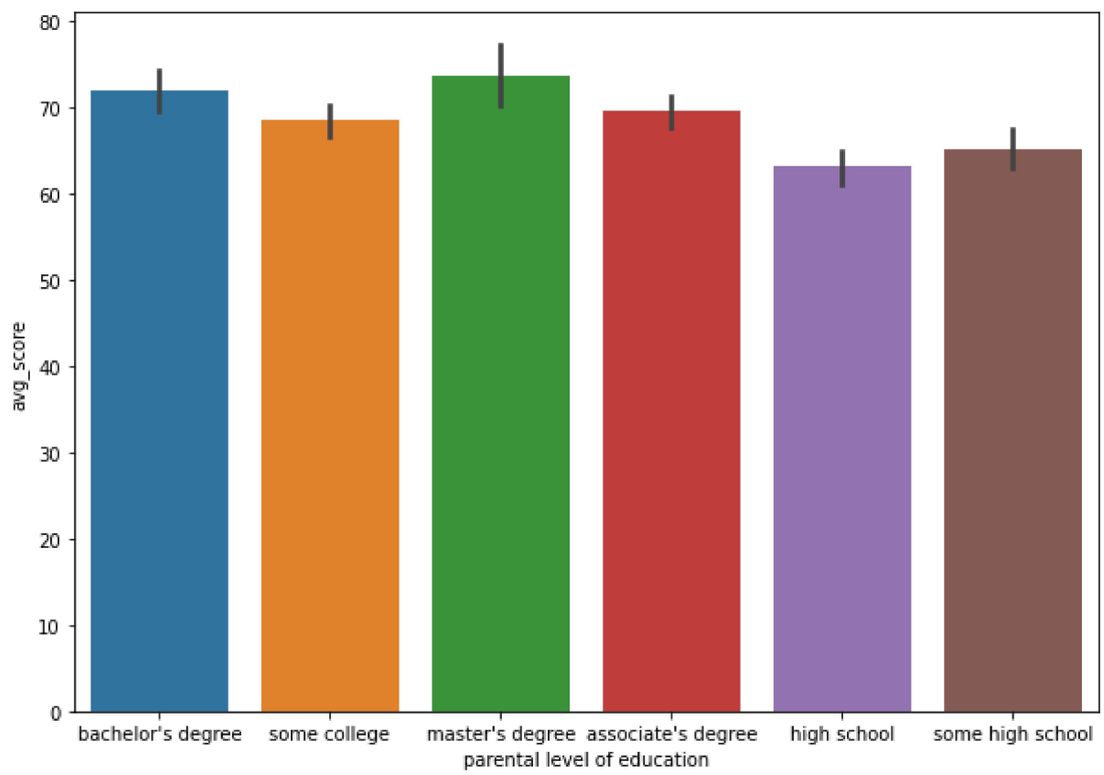
From above we can observe that student who have standard lunch score higher than the free/reduced lunch type. Free lunch affected the most in math score of the sudents

Relation between the student's score and Parental level of education

In [52]:
```
plt.figure(figsize=(10,7))
sns.barplot(x='parental level of education',y='avg_score',data=data)
data.groupby('parental level of education').mean()
```

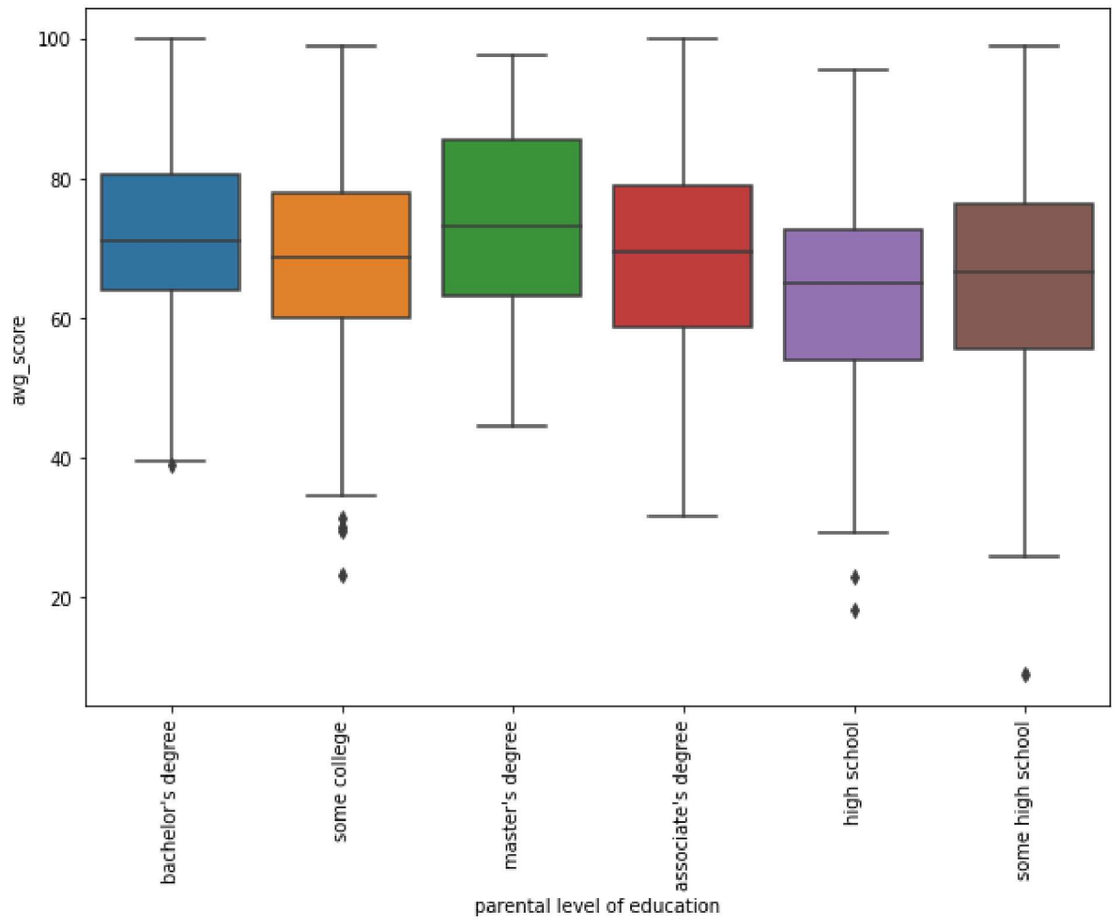
Out[52]:

parental level of education	math score	reading score	writing score	avg_score
associate's degree	67.882883	70.927928	69.896396	69.569069
bachelor's degree	69.389831	73.000000	73.381356	71.923729
high school	62.137755	64.704082	62.448980	63.096939
master's degree	69.745763	75.372881	75.677966	73.598870
some college	67.128319	69.460177	68.840708	68.476401
some high school	63.497207	66.938547	64.888268	65.108007



In [53]:
```
plt.figure(figsize=(10,7))
plt.xticks(rotation=90)
sns.boxplot(data=data, x="parental level of education", y='avg_score')
```

Out[53]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7fc2737fffd0>
```



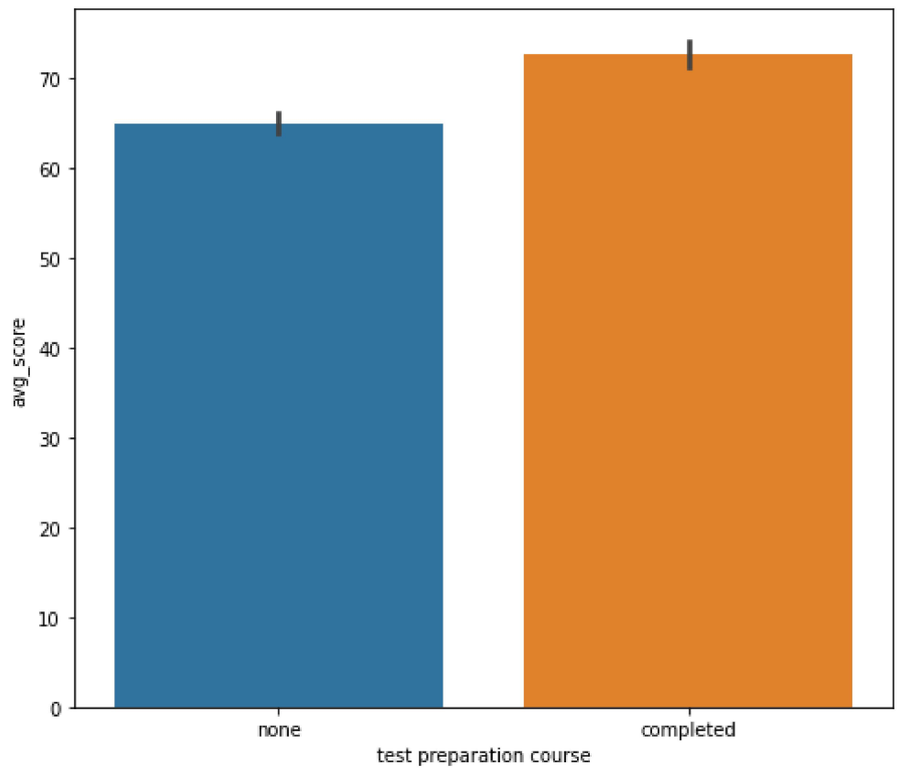
From above barplot and boxplot we can see that the students who have Master degree parental level of education scores the highest.However, students whose parents have 'high school' education level scored the lowest average.

Relation between the student's score and the test preparation course

In [54]:
```
plt.figure(figsize=(8,7))
sns.barplot(x='test preparation course',y='avg_score',data=data)
data.groupby('test preparation course').mean()
```

Out[54]:

test preparation course	math score	reading score	writing score	avg_score
completed	69.695531	73.893855	74.418994	72.669460
none	64.077882	66.534268	64.504673	65.038941



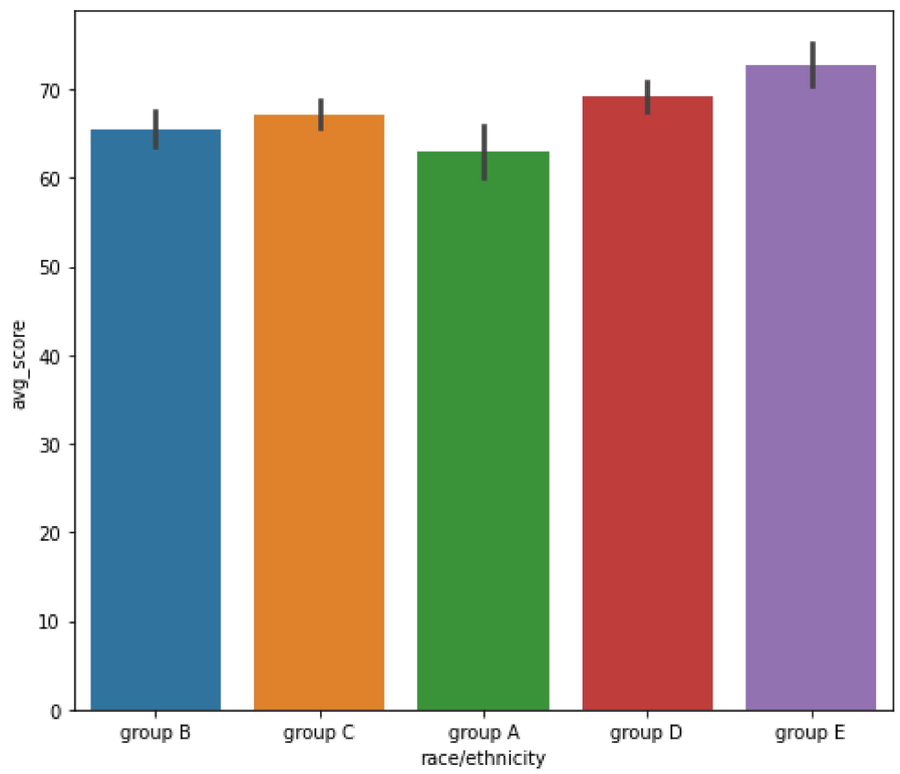
It is clear from the above graph that student who complete the test preparation course perform better than who didn't took the course or completed. If student will complete the course they can perform better in maths and other courses in future.

Relation between the student's score and their race/ethnicity

In [55]:
```
plt.figure(figsize=(8,7))
sns.barplot(x='race/ethnicity',y='avg_score',data=data)
data.groupby('race/ethnicity').mean()
```

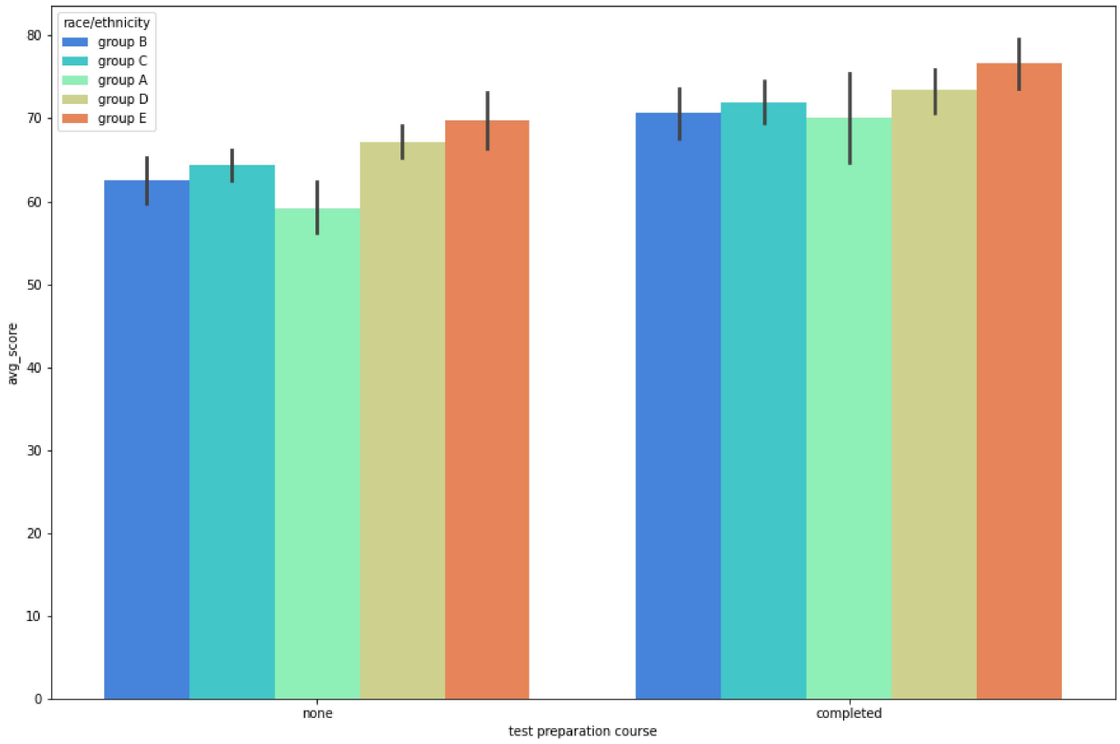
Out[55]:

race/ethnicity	math score	reading score	writing score	avg_score
group A	61.629213	64.674157	62.674157	62.992509
group B	63.452632	67.352632	65.600000	65.468421
group C	64.463950	69.103448	67.827586	67.131661
group D	67.362595	70.030534	70.145038	69.179389
group E	73.821429	73.028571	71.407143	72.752381



In [60]:
```
plt.figure(figsize=(15,10))
sns.barplot(data = data, x = 'test preparation course',
            y = 'avg_score',hue = 'race/ethnicity',palette = 'rainbow' )
```

Out[60]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7fc273254cd0>
```



#Hypothesis Testing

• H0(null hypothesis): the variables are independent, there is no relationship between the two categorical variables. Knowing the value of one variable does not help to predict the value of the other variable.

• H1(alternative hypothesis): the variables are dependent, there is a relationship between the two categorical variables. Knowing the value of one variable helps to predict the value of the other variable

In [61]:
```
#Total number of male and female students
data['gender'].value_counts()
```

Out[61]:
```
female    518
male      482
Name: gender, dtype: int64
```

In [62]:
```
#checking the mean score according to gender
data.groupby('gender').mean()
```

Out[62]:

gender	math score	reading score	writing score	avg_score
female	63.633205	72.608108	72.467181	69.569498
male	68.728216	65.473029	63.311203	65.837483

Hypothesis test of Math score by gender

From the above we can analyse below hypothesis:

H0: Male and Female math scores are same

H1: Male and Female math score are not same

In [63]:
```
Male_mathscore = data[data['gender']=='male']
Female_mathscore = data[data['gender']=='female']
```

In [64]:
```
import scipy.stats
scipy.stats.ttest_ind(Male_mathscore['math score'], Female_mathscore['math score'
```

Out[64]:
```
Ttest_indResult(statistic=5.398000564160736, pvalue=8.420838109090415e-08)
```

Result: The pvalue=8.420838109090415e-08(p<0.05), Hence there is a difference between math score between genders. Male performance in math is better as they score higher than female. We reject the null hypothesis.

Hypothesis test of Reading score by gender

H0: Male and Female reading scores are same

H1: Male and Female reading score are not same

In [65]:
```
Male_readingscore = data[data['gender']=='male']
Female_readingscore = data[data['gender']=='female']
```

In [66]:
```
scipy.stats.ttest_ind(Male_readingscore['reading score'], Female_readingscore['re
```

Out[66]:
```
Ttest_indResult(statistic=-7.9683565184844, pvalue=4.3762967534976715e-15)
```

Result: The pvalue=4.3762967534976715e-15(p<0.05), Hence there is a difference between reading score between genders. Female performance in reading is better as they score higher than male. We reject the null hypothesis.

Hypothesis test of Writing score by gender

H0: Male and Female reading scores are same

H1: Male and Female reading score are not same

In [67]:
```
Male_writingscore = data[data['gender']=='male']
Female_writingscore = data[data['gender']=='female']
scipy.stats.ttest_ind(Male_writingscore['writing score'], Female_writingscore['wr
```

Out[67]:
```
Ttest_indResult(statistic=-9.997718973491885, pvalue=1.7118093718497237e-22)
```

Result: Because the p value is 1.711809371e-22(p<0.05), there is a difference in writing

performance between genders. Therefore, female are better at writing than male. Reject null hypothesis

Hypothesis testing for numerical variable

H0 = Scores have correlation

H1 = Scores do not have correlation

In [74]:
```
#Math and Reading scores
MR = scipy.stats.pearsonr(data['math score'],data['reading score'])
MR
```

Out[74]:
```
(0.8175796636720539, 1.7877531099062402e-241)
```

In [72]:
```
#Writing and Reading scores
WR = scipy.stats.pearsonr(data['writing score'],data['reading score'])
WR
```

Out[72]:
```
(0.9545980771462479, 0.0)
```

In [73]:
```
#Math and Writing scores
MW = scipy.stats.pearsonr(data['math score'],data['writing score'])
MW
```

Out[73]:
```
(0.8026420459498078, 3.3760270425694173e-226)
```

The p-values of MR,WR and MW are less than 0.05. We Reject null hypothesis. The scores have correlation between them.

#Analysis of Variance (ANOVA)

Anova is used to check if there is a difference between means of three or more groups, unlike t-test which is only capable of examining two groups. Because of that, we need a categorical variable which distinct values are more than two.

One Way ANOVA The one-way ANOVA tests the null hypothesis that two or more groups have the same population mean. The test is applied to samples from two or more groups, possibly with differing sizes. Here we are testing the correlation between categorical variables and test scores using one-Way ANOVA

H0: There is no difference between groups and means

H1: There is a difference between the means and groups.

data = Data frame and variable = Categorical columns like gender,parental level of education,test preparation course are used for one-way ANOVA test

In [75]:
```
f, p = scipy.stats.f_oneway(data['math score'], data['reading score'], data['writ
print("FStatistic = ",f, "and P-value = ",p)
```

```
FStatistic =  10.824191628378626 and P-value =  2.0701893192229333e-05
```

Result: From the p-value we can state that all three subjects have different population mean.

In [107]:
```
import statsmodels.api as sm
data.columns = ['gender', 'race/ethnicity', 'parental level of education', 'lunch
from statsmodels.formula.api import ols
def anova_test(data, variable):
    x = ['math_score', 'reading_score', 'writing_score']
    for i,j in enumerate(x):
        lm = ols('{} ~ {}'.format(x[i],variable), data=data).fit()
        table = sm.stats.anova_lm(lm)
        print("P-value for ANOVA test between {} and {} is ".format(x[i],variable
```

In [80]:
```
anova_test(data, 'gender')
```

```
P-value for ANOVA test between math_score and gender is  9.120185549333453e-08
P-value for ANOVA test between reading_score and gender is  4.680538743934009e-
15
P-value for ANOVA test between writing_score and gender is  2.0198777068682407e
-22
```

In [81]:
```
anova_test(data, 'test_prep_course')
```

```
P-value for ANOVA test between math_score and test_prep_course is  1.5359134607
155386e-08
P-value for ANOVA test between reading_score and test_prep_course is  9.0817833
36895556e-15
P-value for ANOVA test between writing_score and test_prep_course is  3.6852917
352476696e-24
```

In [82]:
```
anova_test(data, 'parental_education')
```

```
P-value for ANOVA test between math_score and parental_education is  5.59227238
4108375e-06
P-value for ANOVA test between reading_score and parental_education is  1.16824
57045709003e-08
P-value for ANOVA test between writing_score and parental_education is  1.12027
99969774331e-13
```

In [83]:
```
anova_test(data, 'race')
```

```
P-value for ANOVA test between math_score and race is  1.3732194030370688e-11
P-value for ANOVA test between reading_score and race is  0.0001780089103235852
P-value for ANOVA test between writing_score and race is  1.0979189070066777e-0
5
```

Result: From the above we can see that p-value is less than 0.05. we **Reject** null hypothesis.

All categorical data was statistically tested against the exam scores using a one-Way ANOVA test. This test allows us to accurately confirm whether a category of data is correlated to the numerical outcome. The catergorical data in this dataset is correlated to the reading, writing, and math scores.

#Categorical Data Analysis

Student Performance in exam dataset have two types of variables that includes both numerical values and categorical values:

Numerical variables: Math score, Reading score and Writing score

Categorical Variables: Gender, Test preparation Course, Race/Ethnicity, Parental level of education, Lunch

The **Chi-square test** of independence tests whether there is a relationship between two categorical variables. Foe chi-square test we need to find observed value, expected value, degree of freedom, chi-square statistic, p-value and critical value. The null and alternative hypotheses are:

H0:Gender and Test preparation are independent

H1:Gender and Test preparation are not independent

In [86]:
```
contingency_table  = pd.crosstab(data['test preparation course'], data['gender'])
contingency_table
```

Out[86]:

test preparation course	gender	female	male
completed		184	174
none		334	308

Observed Values

In [87]:
```
Observed_Values = contingency_table.values
print("Observed Values :\n",Observed_Values)
```

```
Observed Values :
 [[184 174]
 [334 308]]
```

Expected Values

In [89]:
```
import scipy.stats
b=scipy.stats.chi2_contingency(contingency_table)
Expected_Values = b[3]
print("Expected Values :\n",Expected_Values)
```

```
Expected Values :
 [[185.444 172.556]
 [332.556 309.444]]
```

Degree of Freedom

In [93]:
```
no_of_rows=len(contingency_table.iloc[0:2,0])
no_of_columns=len(contingency_table.iloc[0,0:2])
df=(no_of_rows-1)*(no_of_columns-1)
df
```

Out[93]: 1

In [94]:
```
#Significance level : We will test our hypothesis at a 5% significance level
alpha=0.05
```

chi-square statistic - $\chi 2$

In [95]:
```
from scipy.stats import chi2
chi_square=sum([(o-e)**2./e for o,e in zip(Observed_Values,Expected_Values)])
chi_square_statistic=chi_square[0]+chi_square[1]
```

In [96]:
```
#critical_value
critical_value=chi2.ppf(q=1-alpha,df=df)

#p-value
p_value=1-chi2.cdf(x=chi_square_statistic,df=df)
```

In [97]:
```
print('Significance level: ',alpha)
print('Degree of Freedom: ',df)
print('chi-square statistic:',chi_square_statistic)
print('critical_value:',critical_value)
print('p-value:',p_value)
```

```
Significance level:  0.05
Degree of Freedom:  1
chi-square statistic: 0.036336202214249484
critical_value: 3.841458820694124
p-value: 0.8488228721237676
```

In [98]:
```
if chi_square_statistic>=critical_value:
    print("Reject H0,There is a relationship between 2 categorical variables")
else:
    print("Fail to reject H0,There is no relationship between 2 categorical varia
```

```
Fail to reject H0,There is no relationship between 2 categorical variables
```

As per our test results, χ2 is 0.03 and the p-value is 0.848822.The chi-square-statistic is less than the critical value, we fail to reject the null hypothesis. There is no relationship between 2 categorical variables. We can conclude that the gender is independent from the test preparation course at a 95% confidence level.

#Logistic Regression

Here ae are taking gender as the response and writing reading and math scores will be the three predictors.

In [99]:
```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import confusion_matrix,precision_score,recall_score,accurac
```

As the gender data type is int64 (male/female) we will use LabelEncoder to transform non-numerical labels to numerical labels. This will convert male/female to binary 0 or 1.

In [105]:
```
from sklearn.preprocessing import LabelEncoder
lc = LabelEncoder()
data['gender'] = lc.fit_transform(data['gender'])
data['race/ethnicity'] = lc.fit_transform(data['race/ethnicity'])
data['parental level of education'] = lc.fit_transform(data['parental level of ed
data['lunch'] = lc.fit_transform(data['lunch'])
data['test preparation course'] = lc.fit_transform(data['test preparation course'
data.head()
```

Out[105]:

	gender	race/ethnicity	parental level of education	lunch	test preparation course	math_score	reading_score	writing_score
0	0	1	1	1	1	72	72	74
1	0	2	4	1	0	69	90	88
2	0	1	3	1	1	90	95	93
3	1	0	0	0	1	47	57	44
4	1	2	4	1	1	76	78	75

In [108]:
```
df = data.drop(['math score', 'writing score', 'reading score'],axis = 1)
df.head()
```

Out[108]:

	gender	race/ethnicity	parental level of education	lunch	test preparation course	avg_score
0	0	1	1	1	1	72.666667
1	0	2	4	1	0	82.333333
2	0	1	3	1	1	92.666667
3	1	0	0	0	1	49.333333
4	1	2	4	1	1	76.333333

In [109]:
```
X = data.drop(columns=['math score','reading score','writing score'])
y = data['gender']
```

In [116]:
```
#Splitting the dataset for 80% of the training set and 20% test set
from sklearn.metrics import confusion_matrix,precision_score,recall_score,accurac
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.20, random_
```

In [117]:
```
model = LogisticRegression()
model.fit(X_train,y_train)
```

Out[117]:
```
LogisticRegression()
```

In [118]:
```
y_pred = model.predict(X_test)
print("Model Accuracy:", accuracy_score(y_test,y_pred)*100,"%")
```

```
Model Accuracy: 100.0 %
```

In [119]:
```
print(classification_report(y_test, y_pred))
```

```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        97
           1       1.00      1.00      1.00       103

    accuracy                           1.00       200
   macro avg       1.00      1.00      1.00       200
weighted avg       1.00      1.00      1.00       200
```

#Linear Regression

Linear regression is the standard algorithm for regression that assumes a linear relationship between inputs and the target variable. A linear regression is where the relationships between the variables can be described with a straight line

A scatter plot is used to plot the actual and predicted values to test for linearity.

In [ ]:
```
X=data[['gender','race/ethnicity','parental level of education','lunch','test pre
Y=data['math score']
```

In [ ]:
```
X = pd.get_dummies(data=X, drop_first=True)
X.head()
```

Out[499]:

	gender	race/ethnicity	parental level of education	lunch	test preparation course	reading score	writing score
0	0	1	1	1	1	72	74
1	0	2	4	1	0	90	88
2	0	1	3	1	1	95	93
3	1	0	0	0	1	57	44
4	1	2	4	1	1	78	75

In [ ]:
```
X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size=0.2, random_s
```

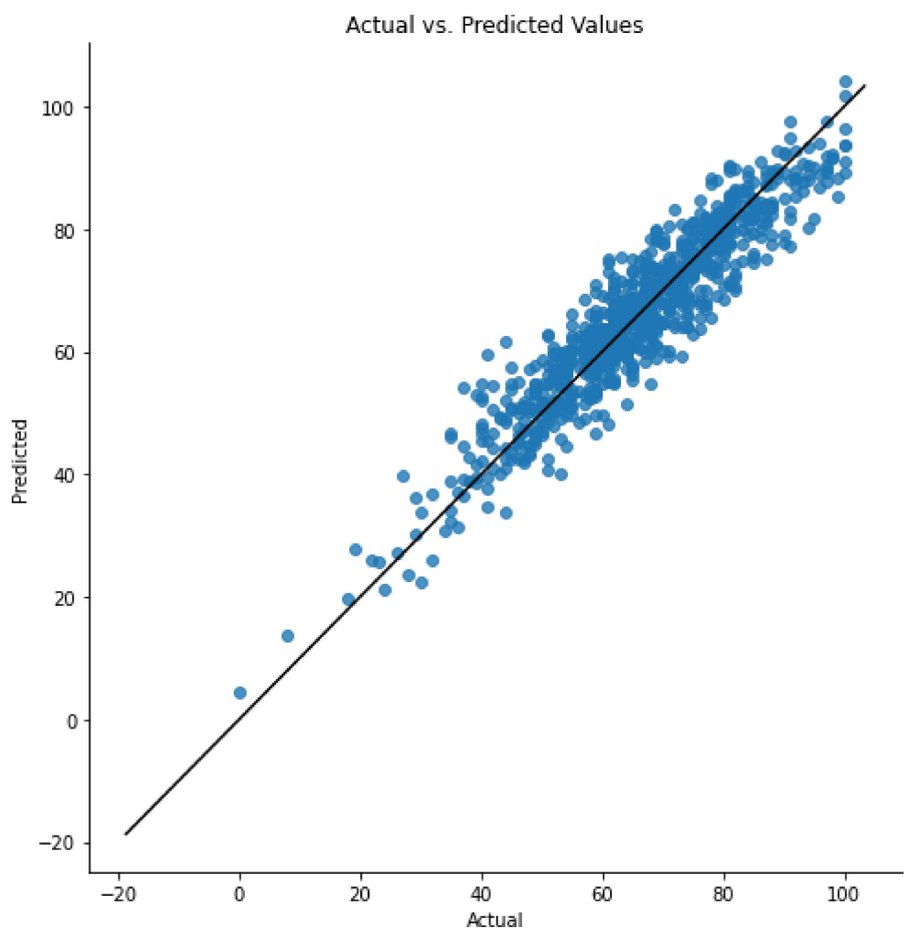
In [ ]:
```
from sklearn.linear_model import LinearRegression
model = LinearRegression()
model.fit(X_train,y_train)
```

Out[501]:
```
LinearRegression()
```

In [ ]:
```
def residuals(model, features, label):
    predictions = model.predict(features)
    results = pd.DataFrame({'Actual': label, 'Predicted': predictions})
    results['Residuals'] = abs(results['Actual']) - abs(results['Predicted'])
    return results
```

In [ ]:
```
def linearity(model, features, label):
    results = residuals(model, features, label)
    sns.lmplot(x='Actual', y='Predicted', data=results, fit_reg=False, size=7)
    line_coords = np.arange(results.min().min(), results.max().max())
    plt.title('Actual vs. Predicted Values')
    plt.plot(line_coords, line_coords, color='black', linestyle='-')

linearity(model,X_train,y_train)
```



In the above graph scatter plot is used to plot the the actual and predicted values. the above graph shows that the variables have a linear relationship.

In [ ]:
```
import statsmodels.api as sm
X_train_Sm= sm.add_constant(X_train)
X_train_Sm= sm.add_constant(X_train)
ls=sm.OLS(y_train,X_train_Sm).fit()
print(ls.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:             math score   R-squared:                       0.872
Model:                            OLS   Adj. R-squared:                  0.871
Method:                 Least Squares   F-statistic:                     771.6
Date:                Mon, 16 May 2022   Prob (F-statistic):               0.00
Time:                        03:27:42   Log-Likelihood:                -2501.3
No. Observations:                 800   AIC:                             5019.
Df Residuals:                     792   BIC:                             5056.
Df Model:                           7
Covariance Type:            nonrobust
================================================================================
================
                                   coef    std err          t      P>|t|
[0.025      0.975]
--------------------------------------------------------------------------------
----------------
const                          -12.6991      1.279     -9.932      0.000     -1
5.209     -10.189
gender                          13.5603      0.425     31.892      0.000      1
2.726      14.395
race/ethnicity                   0.8781      0.174      5.048      0.000
0.537       1.220
parental level of education      0.0383      0.108      0.355      0.722      -
0.173       0.250
lunch                            3.6344      0.437      8.310      0.000
2.776       4.493
test preparation course          3.0781      0.453      6.792      0.000
2.188       3.968
reading score                    0.3524      0.048      7.384      0.000
0.259       0.446
writing score                    0.6092      0.048     12.579      0.000
0.514       0.704
==============================================================================
Omnibus:                        0.521   Durbin-Watson:                   1.940
Prob(Omnibus):                  0.771   Jarque-Bera (JB):                0.487
Skew:                          -0.060   Prob(JB):                        0.784
Kurtosis:                       3.007   Cond. No.                         664.
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctl
y specified.
```

Reject the null hypothesis for gender, race,lunch as p-value is small. Fail to reject null hypothesis for parental level of education. This model has a higher R-squared (0.872). This model provides a better fit to the data.

In [ ]:
```
print("Linear Regression score : ",model.score(X_train,y_train))
```

```
Linear Regression score :  0.8721232515061602
```

The table gives a summary of the regression results. The regression coefficients give the relationship between the dependent variable, math score, with each of the independent variables. Using the coefficient values, the regression line can be estimated. From the adjusted R-squared, the obtained regression model is a 87% fit for the data, indicating the regression model is a good fit.

#Lasso Regression and Ridge Regression

In [ ]:
```
#Lasso Regression model
from sklearn.linear_model import Ridge
from sklearn.linear_model import Lasso

ls = Lasso()
ls.fit(X_train, y_train)
y_pred = ls.predict(X_test)
print("Lasso regression score : ", ls.score(X_test, y_test))
```

```
Lasso regression score :  0.8218504858911362
```

In [ ]:
```
#Ridge Regression model

rd = Ridge()
rd.fit(X_train,y_train)
y_pred = rd.predict(X_test)
print("Ridge regression score : ", rd.score(X_test, y_test))
```

```
Ridge regression score :  0.9999683421049511
```

Regression accuracies:

The logistic regression gives us 100% model accuracy.

The linear regression gives us 87% model accuracy.

The Lasso regression model gives us 82% model accuracy.

The ridge regression gives us 99% model accuracy. Hence model is good fit.

Conclusion:

- The major factors contribute to test outcomes are Lunch, test preparation and gender.
- Comparing different groups tells us that taking a test preparation course will increase the overall performance of the students and having standard lunch will help perform better. But the parental level of education impacted the student performance.
- from hypothesis testing we get that the male and female performance are not same. Male scored more in math and Female scored more in reading and writing. Concluding that female students have better overall scores than male students.
- We concluded that using the chi-square test, the P-value is greater than the significance level of 5%, hence we fail to reject the null hypothesis. The chi-square statistic is inferior to the critical value, we fail to reject the null hypothesis. That's why we conclude that the gender is independent from the test preparation course at a 95% confidence level
- The model accuracy is 100% for the logistic regression and in other regression models it more than 82%. hence this model provides better fit to our data.

References:

[1]John A. Rice - Mathematical Statistics and Data Analysis 3ed (Duxbury Advanced) (2006, Duxbury Press)

[2]https://www.scribbr.com/statistics/anova-in-r/

[3]Lecture: Stat Lab with R: The Analysis of Categorical Data

[4]Data Source: https://www.kaggle.com/datasets/spscientist/students-performance-in-exams

[5] https://statsandr.com/blog/chi-square-test-of-independence-in-r/

[6]An Introduction to Statistical Learning with Applications in R, by G. James, D. Witten, T. Hastie, &