In [78]:
```python
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib
from matplotlib import pyplot as plt
import warnings
warnings.filterwarnings("ignore")
```

In [3]:
```python
sp=pd.read_csv("C:\\Users\\gajja\\Downloads\\StudentsPerformance.csv")
sp.head(5)
```

Out[3]:

	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 [4]:
```python
sp.describe()
```

Out[4]:

	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

In [7]:
```python
sp['gender'].value_counts().plot.bar(title='Frequency Distribution of Gender',color=
```
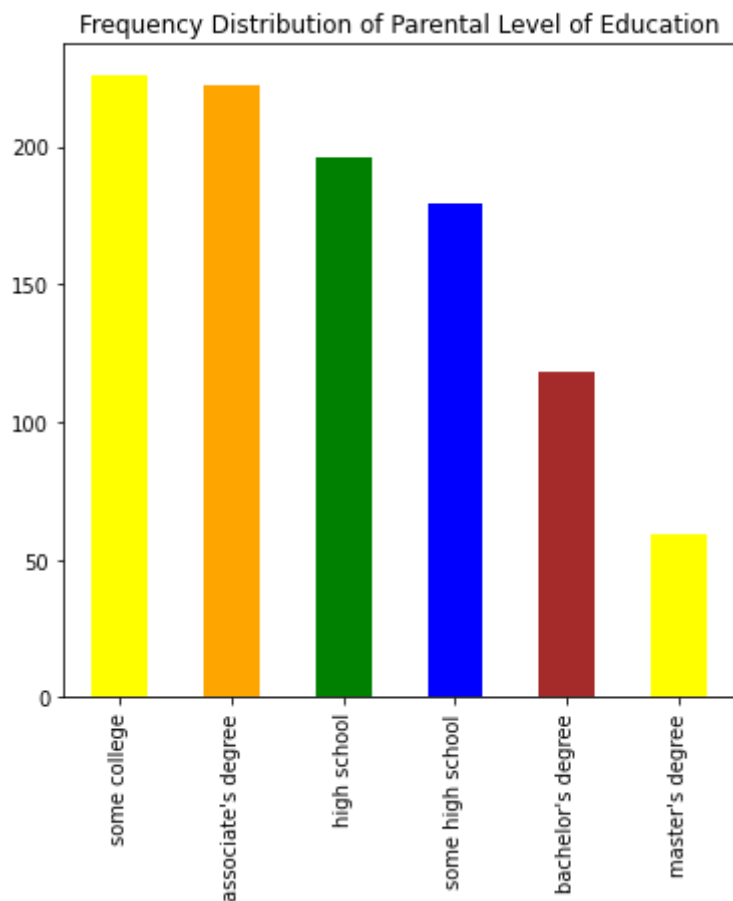
Out[7]: <Axes: title={'center': 'Frequency Distribution of Gender'}>

Frequency Distribution of Gender
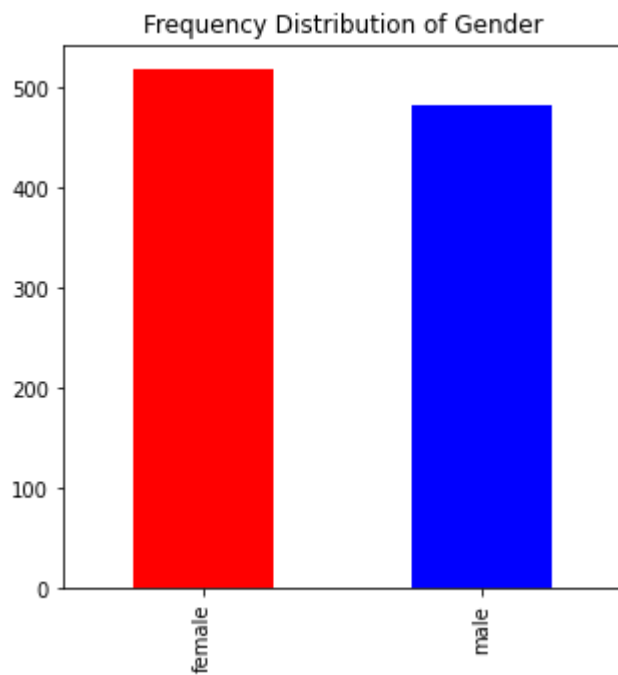
In [8]:
```python
sp['race/ethnicity'].value_counts().plot.bar(title='Frequency Distribution of Race/E
```

Out[8]: `<Axes: title={'center': 'Frequency Distribution of Race/Ethnicity'}>`

Frequency Distribution of Race/Ethnicity
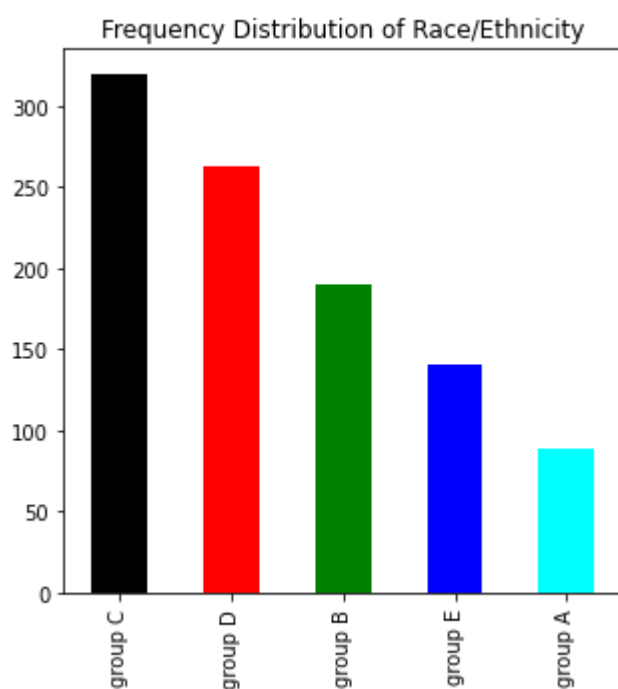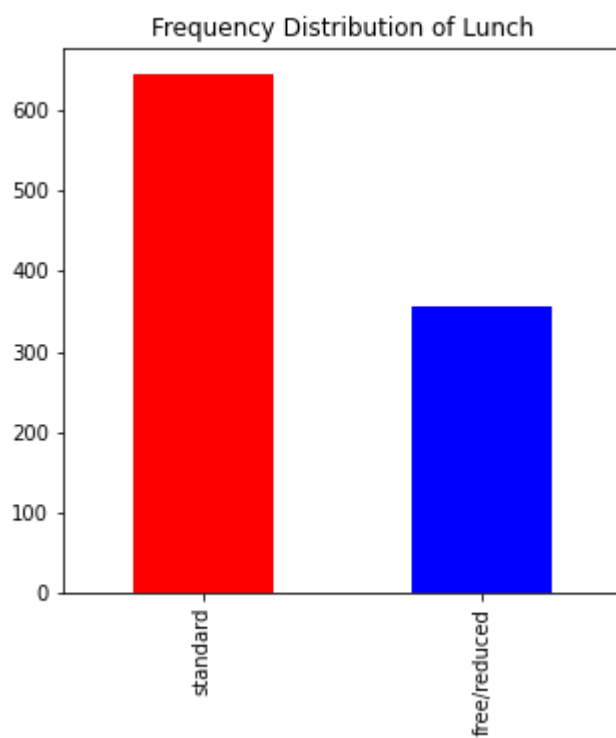

In [13]:
```python
sp['parental level of education'].value_counts().plot.bar(title='Frequency Distribut
```

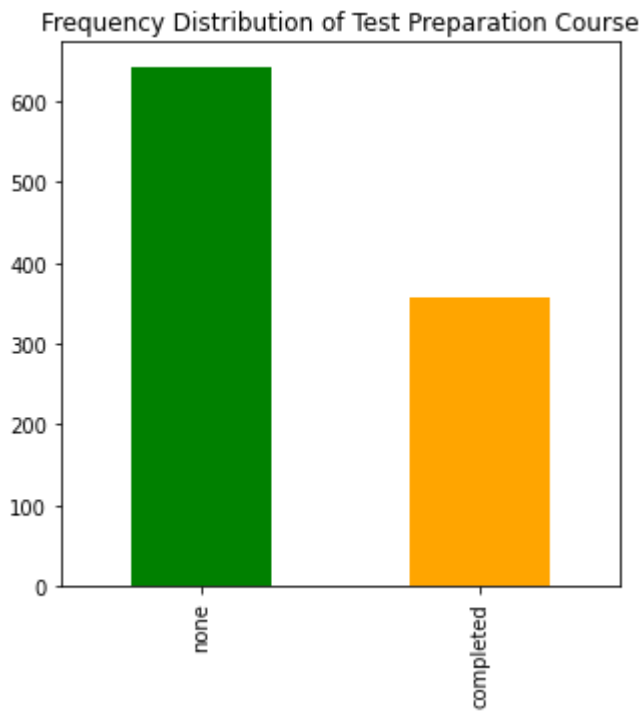Out[13]: `<Axes: title={'center': 'Frequency Distribution of Parental Level of Education'}>`

Frequency Distribution of Parental Level of Education

In [15]:
```python
sp['lunch'].value_counts().plot.bar(title='Frequency Distribution of Lunch',color=['
```

Out[15]: `<Axes: title={'center': 'Frequency Distribution of Lunch'}>`

Frequency Distribution of Lunch

In [18]:
```python
sp['test preparation course'].value_counts().plot.bar(title='Frequency Distribution
```

Out[18]: `<Axes: title={'center': 'Frequency Distribution of Test Preparation Course'}>`

Frequency Distribution of Test Preparation Course

In [19]:

```python
plt.rcParams['figure.figsize'] = (30, 12)

plt.subplot(1, 5, 1)
size = sp['gender'].value_counts()
labels = 'Female', 'Male'
color = ['red','green']


plt.pie(size, colors = color, labels = labels,autopct = '.%2f%%')
plt.title('Gender', fontsize = 20)
plt.axis('off')



plt.subplot(1, 5, 2)
size = sp['race/ethnicity'].value_counts()
labels = 'Group C', 'Group D','Group B','Group E','Group A'
color = ['red', 'green', 'blue', 'cyan','orange']

plt.pie(size, colors = color,labels = labels,autopct = '.%2f%%')
plt.title('Race/Ethnicity', fontsize = 20)
plt.axis('off')



plt.subplot(1, 5, 3)
size = sp['lunch'].value_counts()
labels = 'Standard', 'Free'
color = ['red','green']

plt.pie(size, colors = color,labels = labels,autopct = '.%2f%%')
plt.title('Lunch', fontsize = 20)
plt.axis('off')


plt.subplot(1, 5, 4)
size = sp['test preparation course'].value_counts()
labels = 'None', 'Completed'
color = ['red','green']
```

```python
plt.pie(size, colors = color,labels = labels,autopct = '.%2f%%')
plt.title('Test Course', fontsize = 20)
plt.axis('off')


plt.subplot(1, 5, 5)
size = sp['parental level of education'].value_counts()
labels = 'Some College', "Associate's Degree",'High School','Some High School',"Bach
color = ['red', 'green', 'blue', 'cyan','orange','grey']

plt.pie(size, colors = color,labels = labels,autopct = '.%2f%%')
plt.title('Parental Education', fontsize = 20)
plt.axis('off')


plt.tight_layout()
plt.grid()

plt.show()
```
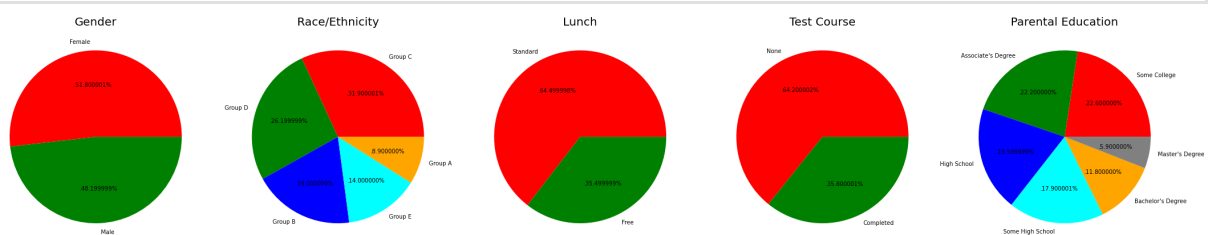


Number of Male and Female students is almost equal

Number students are greatest in Group C

Number of students who have standard lunch are greater

Number of students who have not enrolled in any test preparation course is greater

# Number of students whose parental education is "Some College" is greater followed closely by "Associate's Degree"*

In [30]:
```python
plt.rcParams['figure.figsize'] = (20, 10)
sns.countplot(sp['math score'], palette = 'bright')
plt.title('Math Score Comparison',fontsize = 25)
plt.show()
```

Math Score Comparison
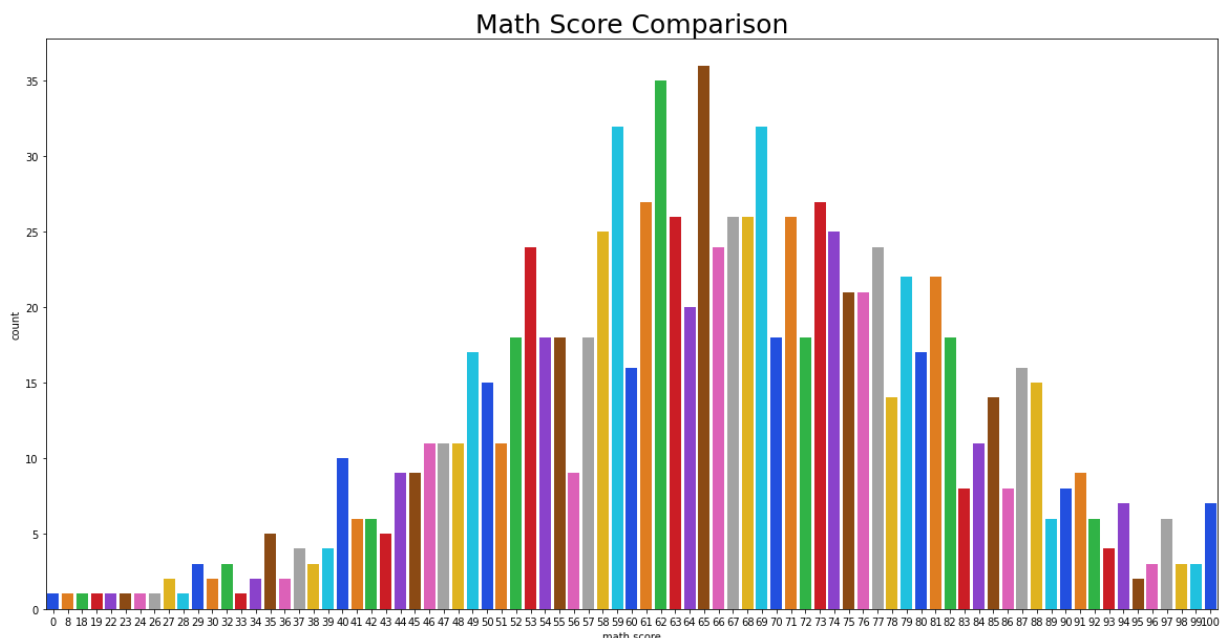
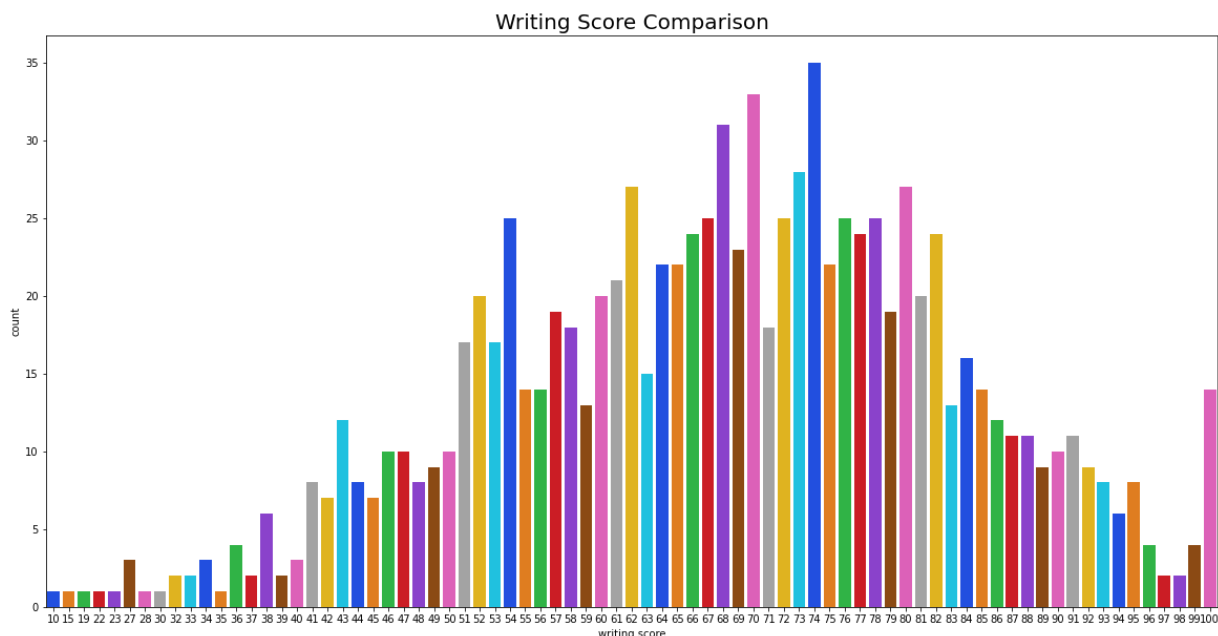# Maximum number of students had a Maths Score of 65

In [31]:
```python
plt.rcParams['figure.figsize'] = (20, 10)
sns.countplot(sp['reading score'], palette = 'bright')
plt.title('Reading Score Comparison',fontsize = 20)
plt.show()
```


Reading Score Comparison
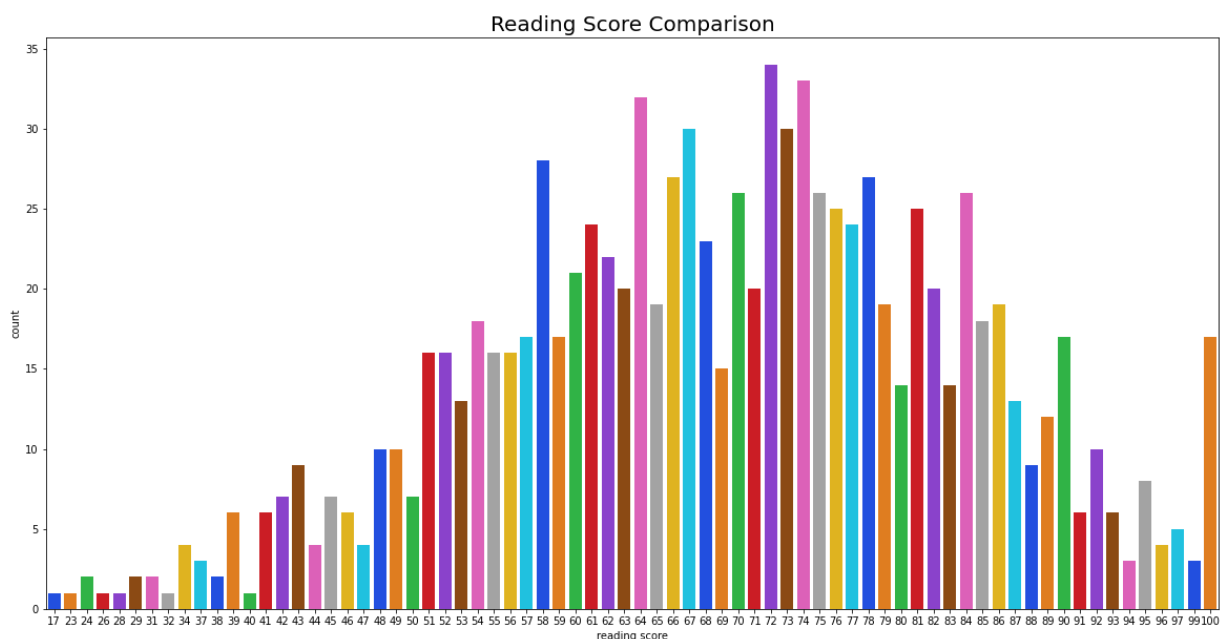
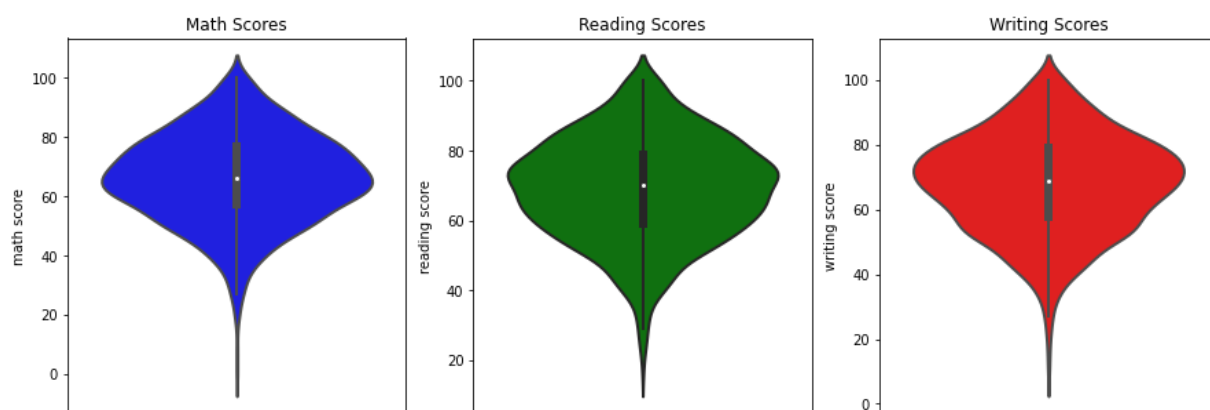# Maximum number of students had a Reading Score of 72

In [32]:
```python
plt.rcParams['figure.figsize'] = (20, 10)
sns.countplot(sp['writing score'], palette = 'bright')
plt.title('Writing Score Comparison',fontsize = 20)
plt.show()
```

Maximum number of students had a Writing Score of 74

```
In [33]:  plt.figure(figsize=(15,5))
          plt.subplot(131)
          plt.title('Math Scores')
          sns.violinplot(y='math score',data=sp,color='blue',linewidth=2)
          plt.subplot(132)
          plt.title('Reading Scores')
          sns.violinplot(y='reading score',data=sp,color='green',linewidth=2)
          plt.subplot(133)
          plt.title('Writing Scores')
          sns.violinplot(y='writing score',data=sp,color='red',linewidth=2)
          plt.show()
```
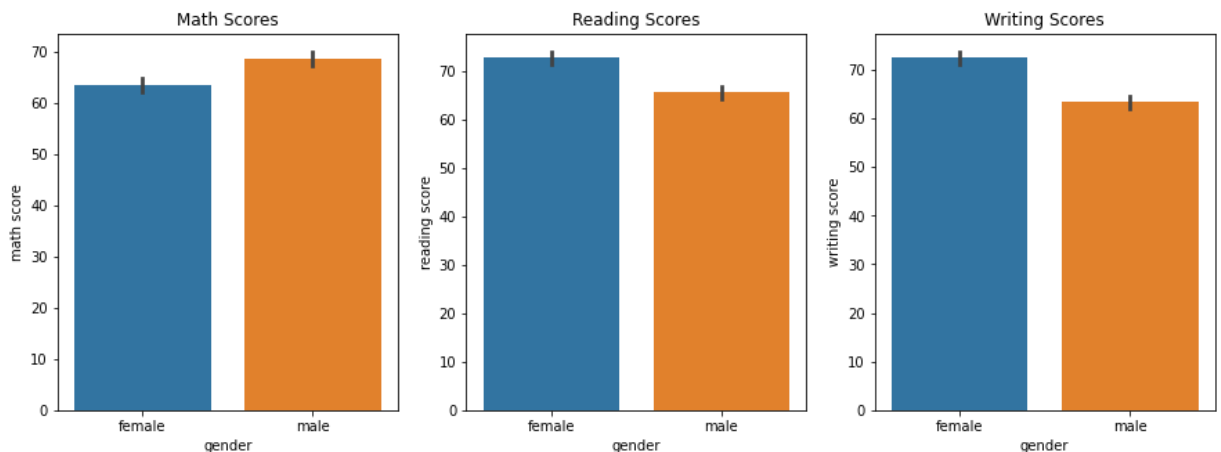

Maximum number of students have scored between 60-80 marks in all 3 subjects

```
In [34]:  plt.figure(figsize=(15,5))
          plt.subplot(131)
          plt.title('Math Scores')
          sns.barplot(x="gender", y="math score", data=sp)
          plt.subplot(132)
```
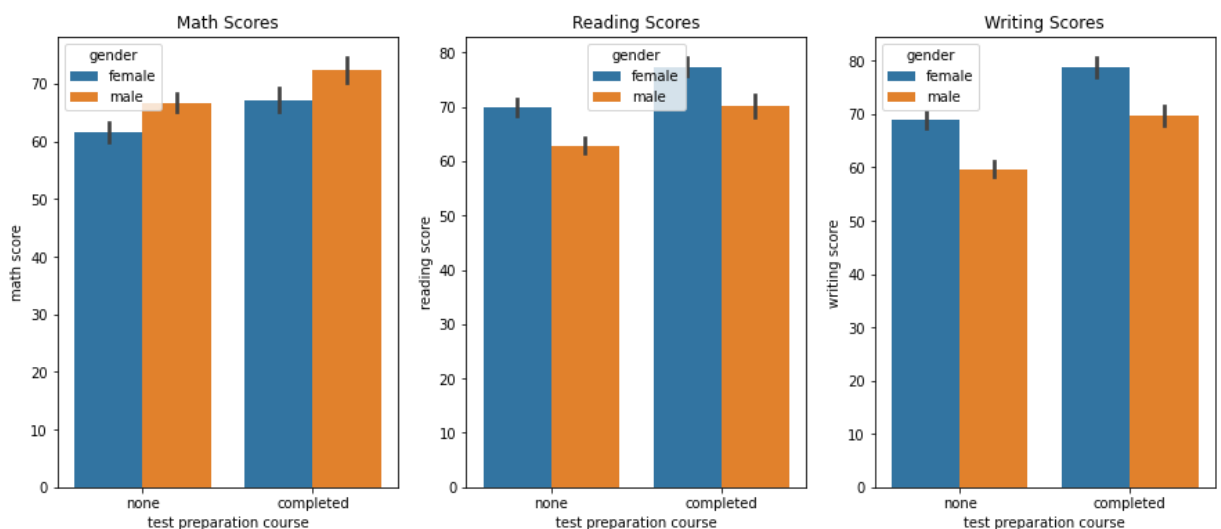
```
plt.title('Reading Scores')
sns.barplot(x="gender", y="reading score", data=sp)
plt.subplot(133)
plt.title('Writing Scores')
sns.barplot(x="gender", y="writing score", data=sp)
plt.show()
```


Male students scored higher in Maths whereas Female students scored higher in Reading and Writing

In [41]:
```
plt.figure(figsize=(15,6))
plt.subplot(131)
plt.title('Math Scores')
sns.barplot(hue="gender", y="math score", x="test preparation course", data=sp)
plt.subplot(132)
plt.title('Reading Scores')
sns.barplot(hue="gender", y="reading score", x="test preparation course", data=sp)
plt.subplot(133)
plt.title('Writing Scores')
sns.barplot(hue="gender", y="writing score", x="test preparation course", data=sp)
plt.show()
```


Students who completed the test preparation course acheived higher score in all 3 subject

In [42]:
```python
plt.figure(figsize=(15,5))
plt.subplot(131)
plt.title('Math Scores')
sns.barplot(hue="gender", y="math score", x="lunch", data=sp)
plt.subplot(132)
plt.title('Reading Scores')
sns.barplot(hue="gender", y="reading score", x="lunch", data=sp)
plt.subplot(133)
plt.title('Writing Scores')
sns.barplot(hue="gender", y="writing score", x="lunch", data=sp)
plt.show()
```
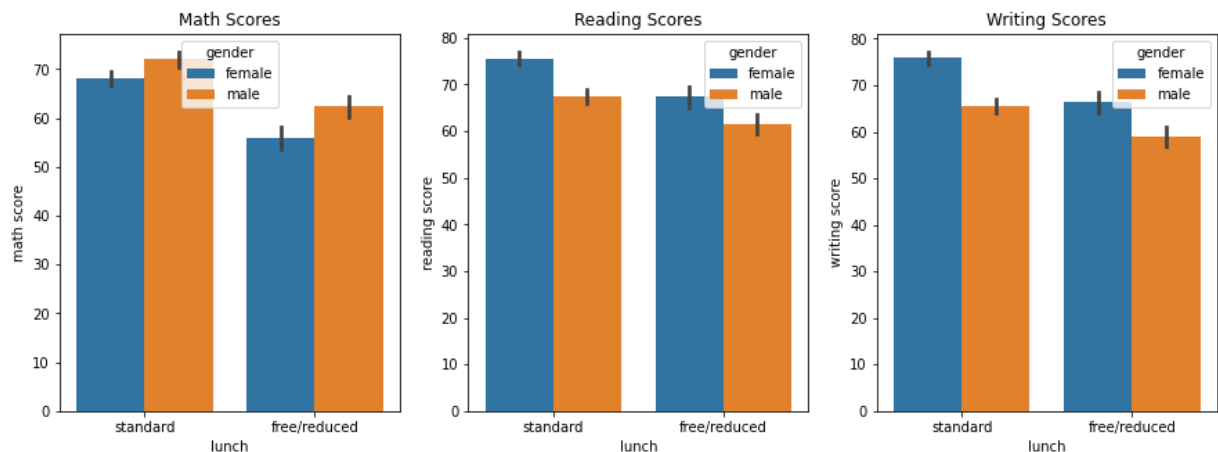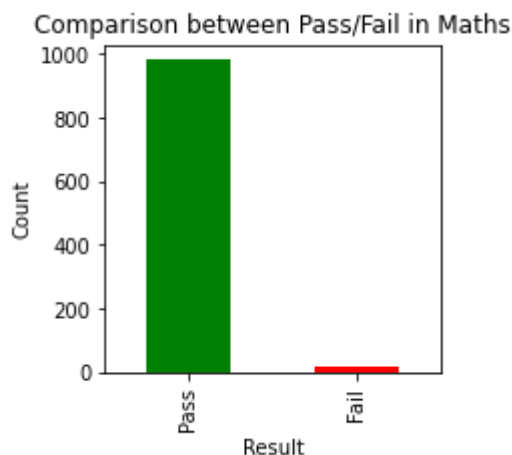
# Students who chose standard lunch acheived higher score in all 3 subject

In [43]:
```python
passingmark=33
sp['pass_math'] = np.where(sp['math score']>= passingmark, 'Pass', 'Fail')
sp['pass_math'].value_counts(dropna = False).plot.bar(color=['green','red'], figsize

plt.title('Comparison between Pass/Fail in Maths')
plt.xlabel('Result')
plt.ylabel('Count')
plt.show()
sp['pass_math'].value_counts()
```

Out[43]:
```
Pass   981
Fail   19
Name: pass_math, dtype: int64
```

In [44]:
```python
sp['pass_read'] = np.where(sp['reading score']>= passingmark, 'Pass', 'Fail')
sp['pass_read'].value_counts(dropna = False).plot.bar(color = ['green','red'], figsi
plt.title('Comparison between Pass/Fail in Reading')
plt.xlabel('Result')
plt.ylabel('Count')
plt.show()
sp['pass_read'].value_counts()
```

Comparison between Pass/Fail in Reading
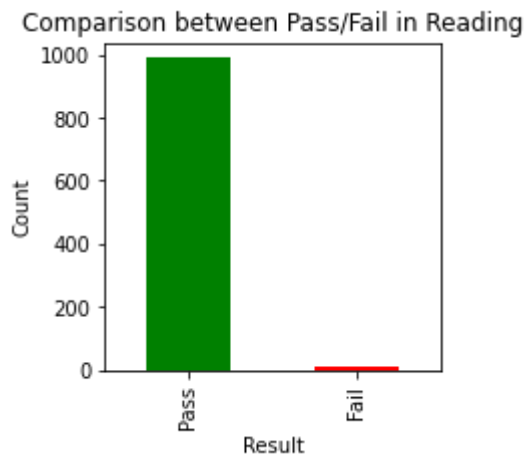
Out[44]:
```
Pass    989
Fail     11
Name: pass_read, dtype: int64
```

In [45]:
```python
sp['pass_write'] = np.where(sp['writing score']>= passingmark, 'Pass', 'Fail')
sp['pass_write'].value_counts(dropna = False).plot.bar(color = ['green','red'], figs
plt.title('Comparison between Pass/Fail in Writing')
plt.xlabel('Result')
plt.ylabel('Count')
plt.show()
sp['pass_write'].value_counts()
```

Comparison between Pass/Fail in Writing
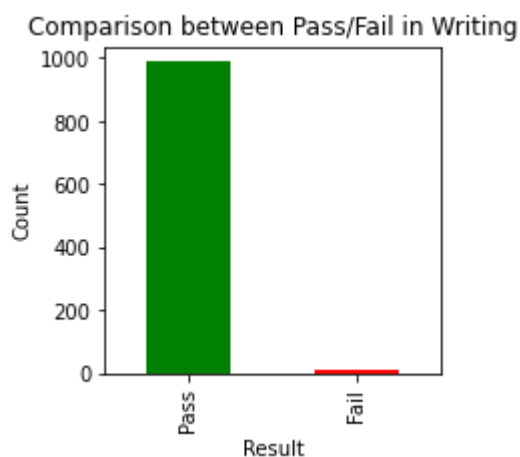
Out[45]:
```
Pass    988
Fail     12
Name: pass_write, dtype: int64
```

In [55]:
```python
sp['total_score'] = sp['math score'] + sp['reading score'] + sp['writing score']

sp['total_score'].value_counts(normalize = True)
sp['total_score'].value_counts(dropna = True).plot.bar(color = 'green', figsize = (4

plt.title('Comparison Of Total Score Of All Students',size=30,c="red")
plt.xlabel('Total Score',size=30)
```

```
plt.ylabel('Count',size=30)
plt.show()
```


Comparison Of Total Score Of All Students
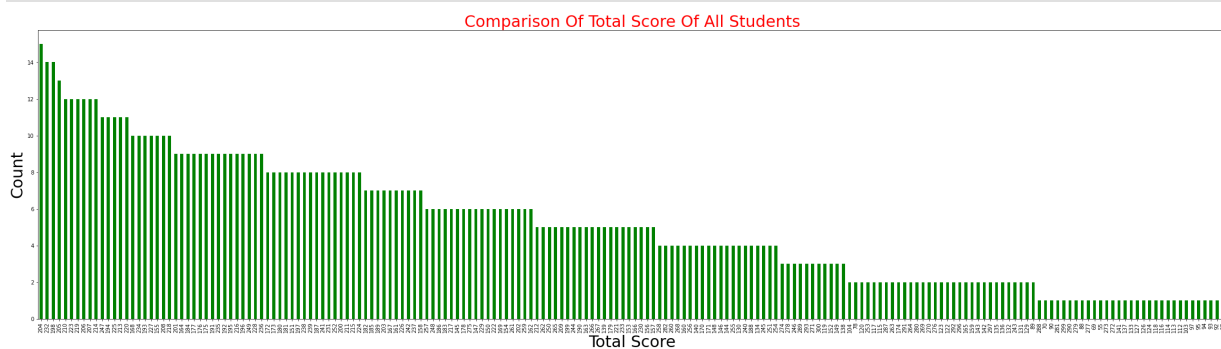
Maximum number of students had a Total Score of 204

In [56]:

```
#For calculating percentage scored by each student
from math import *
sp['percentage'] = sp['total_score']/3
for i in range(0, 1000):
  sp['percentage'][i] = ceil(sp['percentage'][i])

sp['percentage'].value_counts(normalize = True)
sp['percentage'].value_counts(dropna = False).plot.bar(figsize = (20, 8), color = 'g

plt.title('Comparison Of Percentage Of All The Students')
plt.xlabel('Percentage')
plt.ylabel('Count')
plt.show()
```
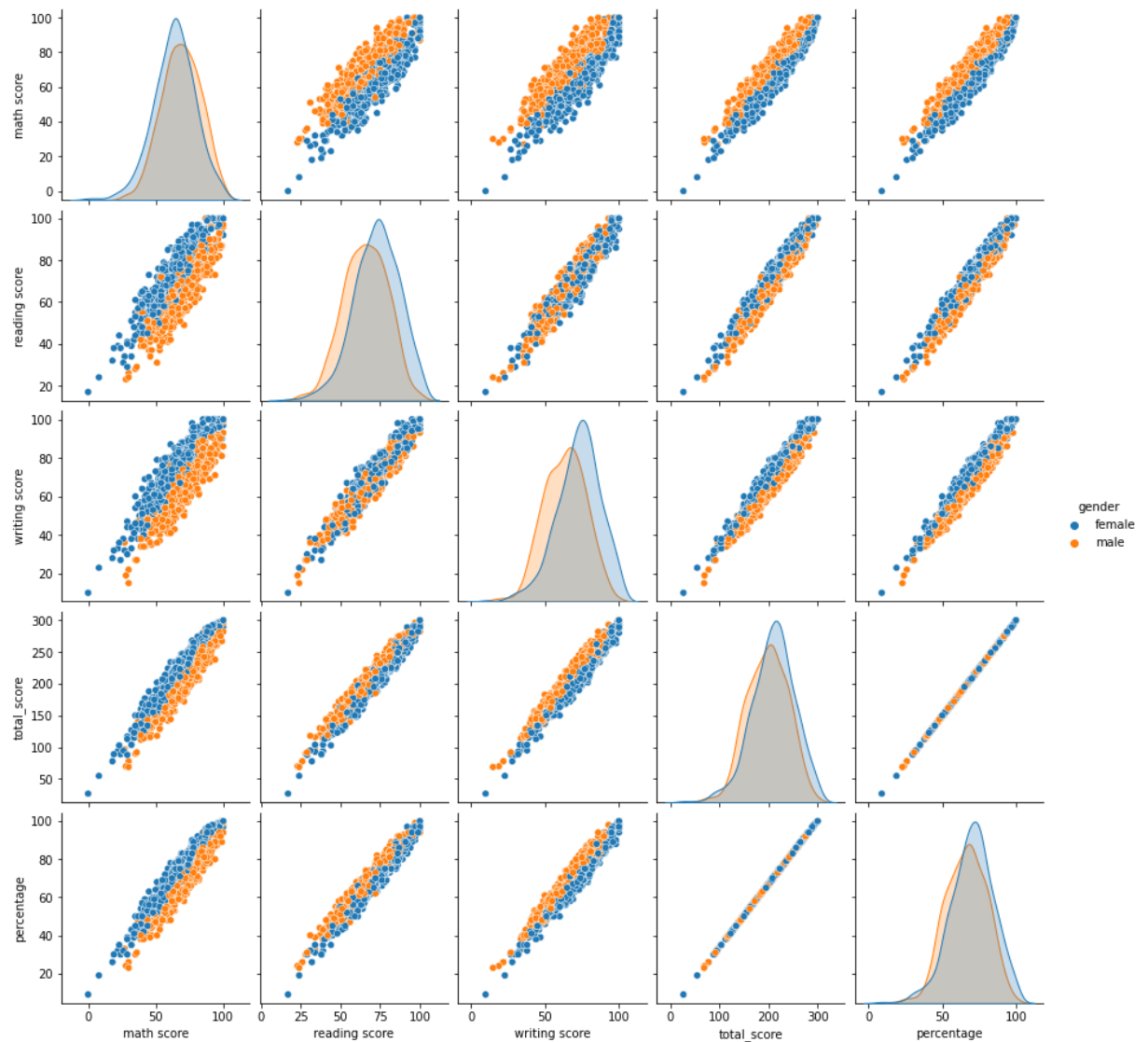

Comparison Of Percentage Of All The Students
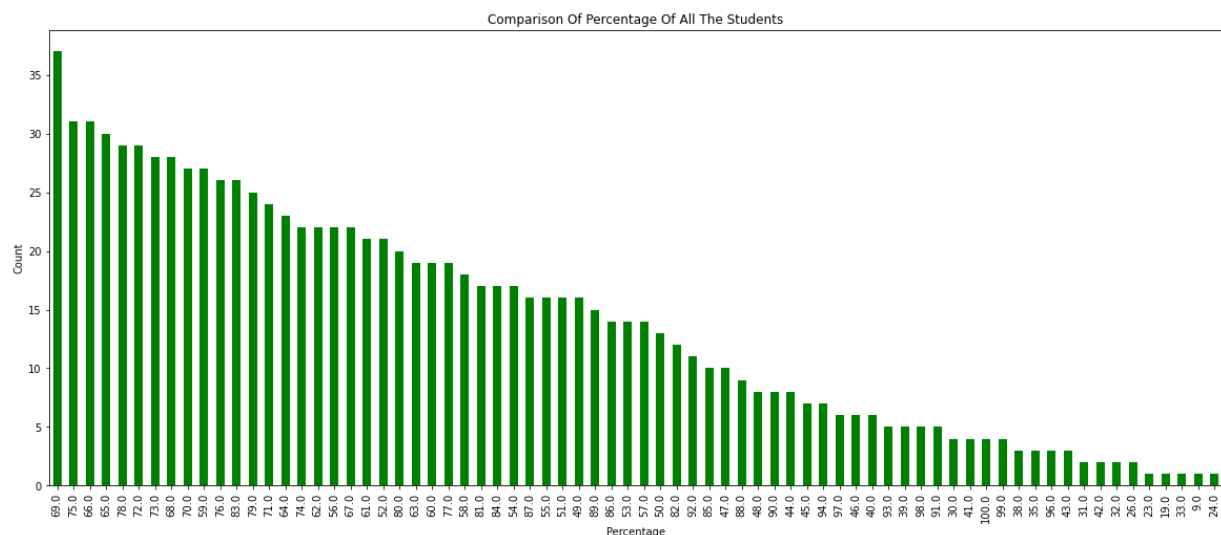
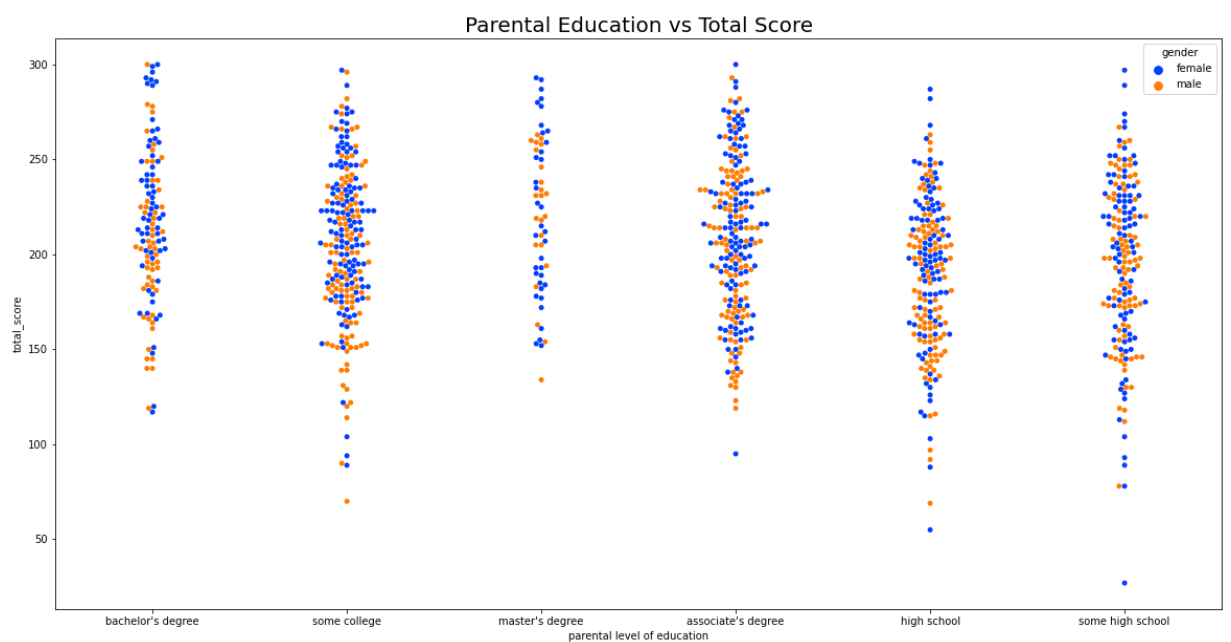# Maximum number of students scored 69%

In [64]:

```
sns.pairplot(sp,hue = 'gender')
plt.show()
```

In [68]:
```python
plt.rcParams['figure.figsize'] = (20,10)
plt.subplot(1, 1, 1)
sns.swarmplot(sp['parental level of education'], sp['total_score'], hue = sp['gender
plt.title('Parental Education vs Total Score', fontsize = 20)
plt.show()
```
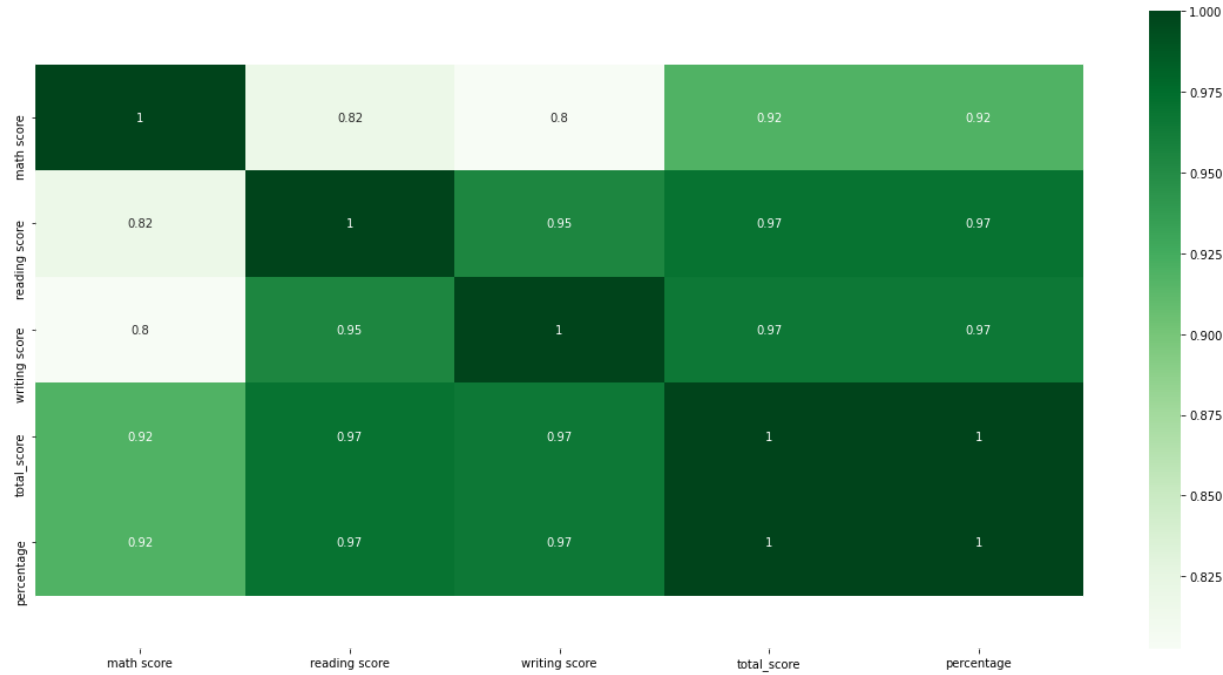
In [69]:
```python
plt.rcParams['figure.figsize'] = (20,10)
plt.subplot(1, 1, 1)
sns.swarmplot(sp['race/ethnicity'], sp['total_score'], hue = sp['gender'], palette =
plt.title('Race/Ethnicity vs Total Score', fontsize = 20)
plt.show()
```
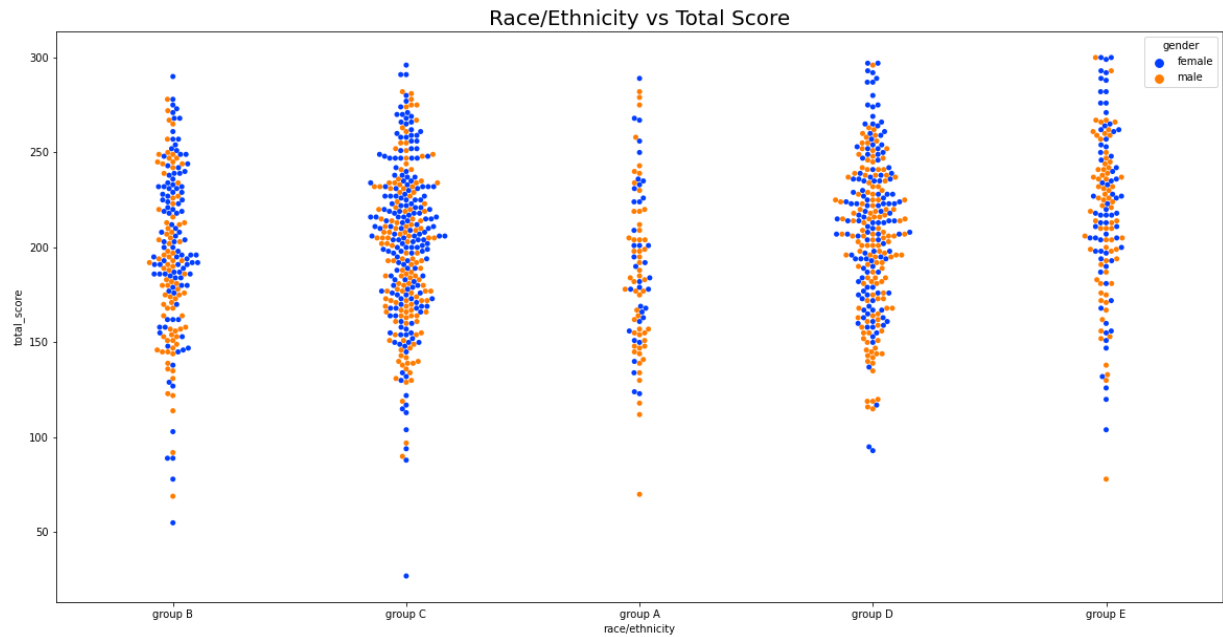

In [70]:
```python
#Visualizing realtions between various attributes using heatmap
plt.figure(figsize=(20,10))
plt.rcParams['figure.figsize'] = (18, 16)
sp_corr = sp.corr()
ax = sns.heatmap(sp_corr, annot=True,cmap="Greens")
bottom, top = ax.get_ylim()
ax.set_ylim(bottom + 0.5, top - 0.5)
sp_corr
```

Out[70]:

	math score	reading score	writing score	total_score	percentage
math score	1.000000	0.817580	0.802642	0.918746	0.918521
reading score	0.817580	1.000000	0.954598	0.970331	0.970271
writing score	0.802642	0.954598	1.000000	0.965667	0.965422
total_score	0.918746	0.970331	0.965667	1.000000	0.999813
percentage	0.918521	0.970271	0.965422	0.999813	1.000000

In [ ]: