

### Problem Statement:

gender : Gender of the student..... race/ethnicity : Race of the Student As Group A/B/C..... parental level of education : What is the education Qualification of Students Parent..... lunch : Whether the lunch is Standard type/Free lunch or Some discounted lunch..... test preparation course : Whether Student has Taken or not and Completed..... math score : Scores in Maths.... reading score : Scores in Reading..... writing score : Scores in Writing.....

Objective of this Analysis: To understand the how the student's performance (test scores) is affected by the other variables (Gender, Ethnicity, Parental level of education, Lunch, Test preparation course). What to do in Exploratory Data Analysis: To Analyse insights in the dataset. To understand the connection between the variables and to uncover the underlying structure To extract the important Variables. To test the underlying assumptions. Provide Insights with Suitable Graphs and Visualizations. Write all your inferences with supporting Analysis and Visualizations.

In [47]:

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

import os
for dirname, _, filenames in os.walk('//input'):
    for filename in filenames:
        print(os.path.join(dirname, filename))
```

In [48]:

```
df=pd.read_csv('StudentsPerformance (1).csv')
```

In [49]:

```
df.describe()
```

Out[49]:

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

In [50]:

```
df.shape
```

Out[50]:

```
(1000, 8)
```

In [51]:

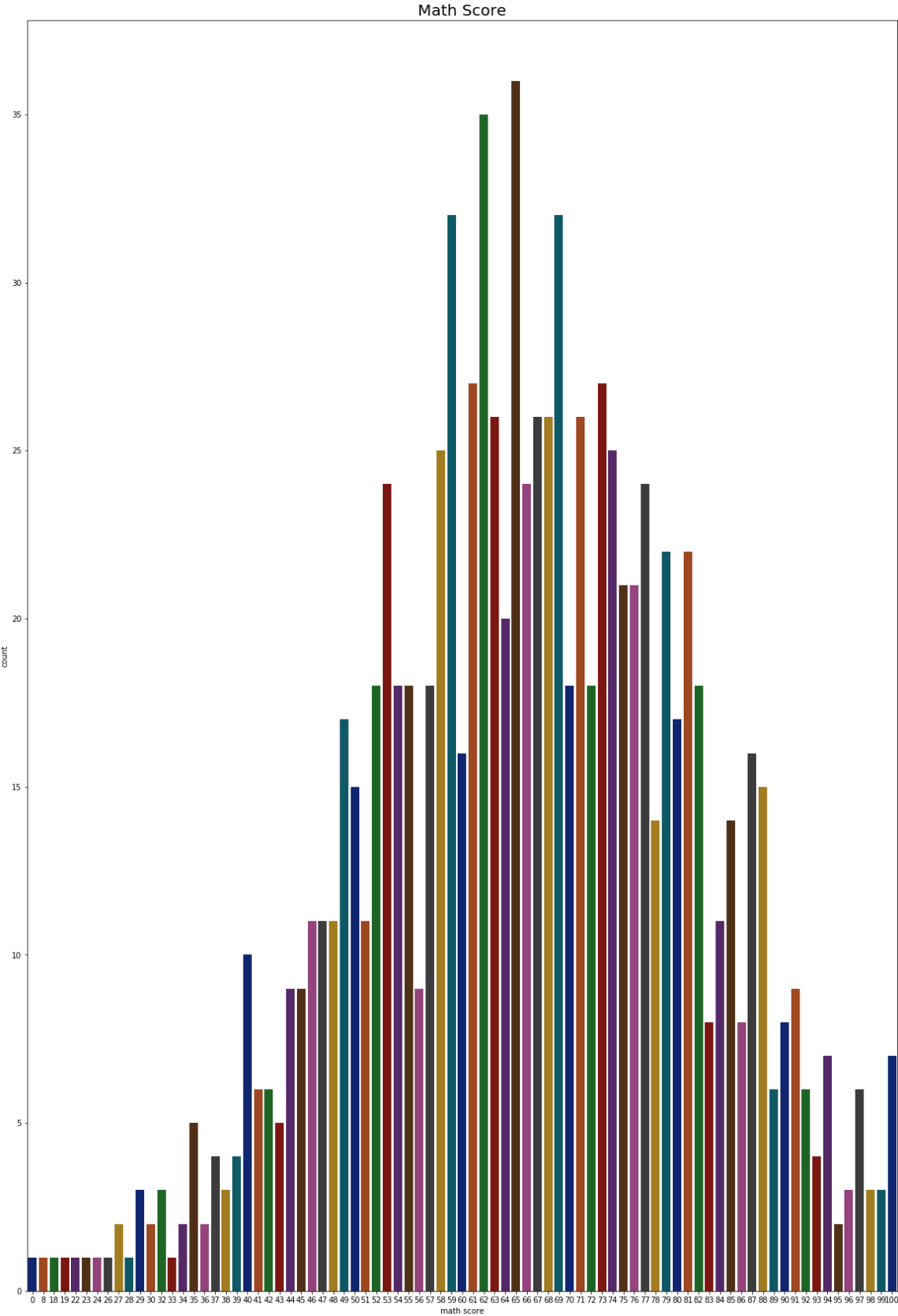
```
df.isnull().sum()
```

Out[51]:

```
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
```

In [52]:

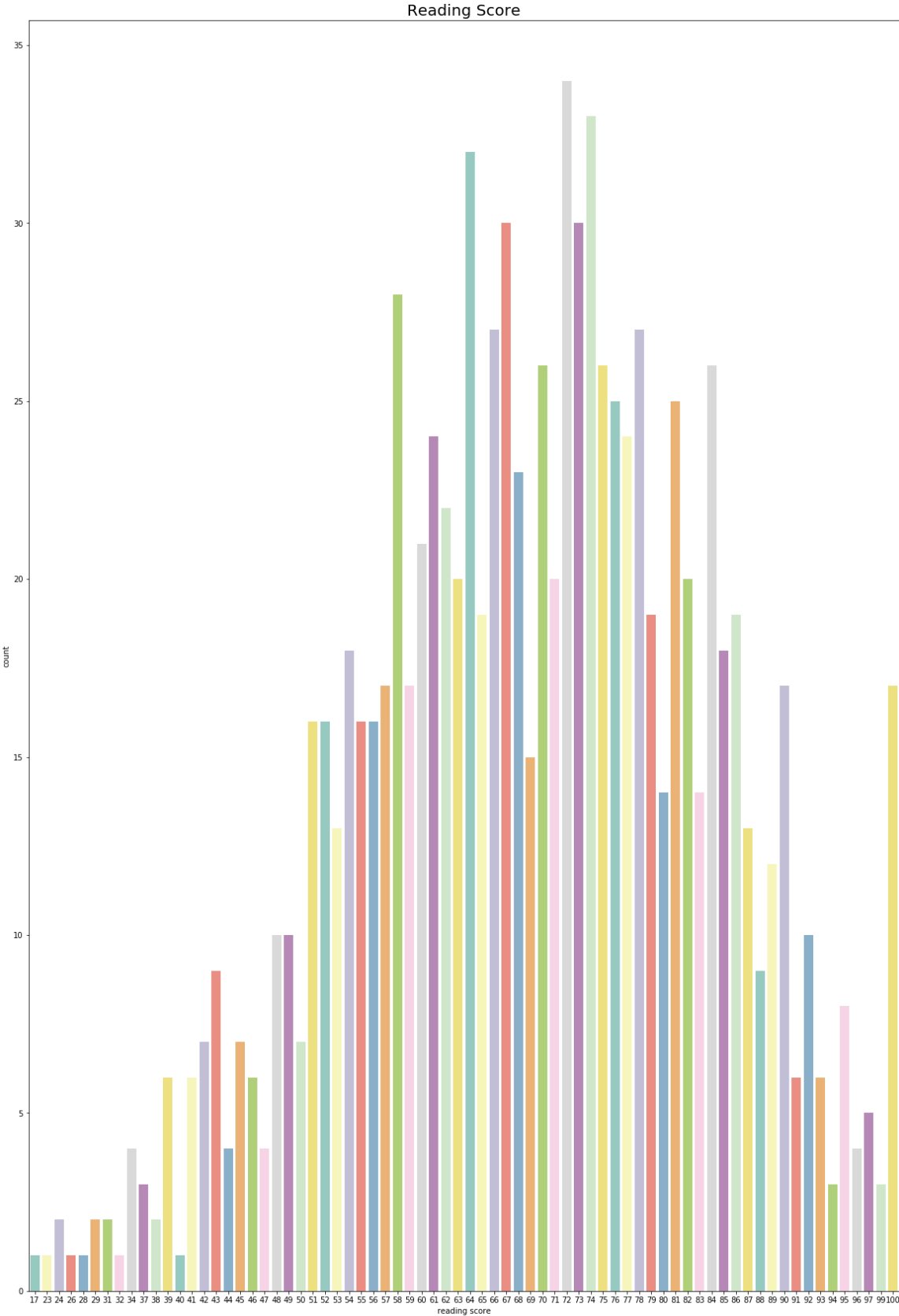
```
plt.rcParams['figure.figsize'] = {20, 30}  
sns.countplot(df['math score'], palette = 'dark')  
plt.title('Math Score',fontsize = 20)  
plt.show()
```





In [39]:

```
plt.rcParams['figure.figsize'] = {20, 30}  
sns.countplot(df['reading score'], palette = 'Set3')  
plt.title('Reading Score',fontsize = 20)  
plt.show()
```

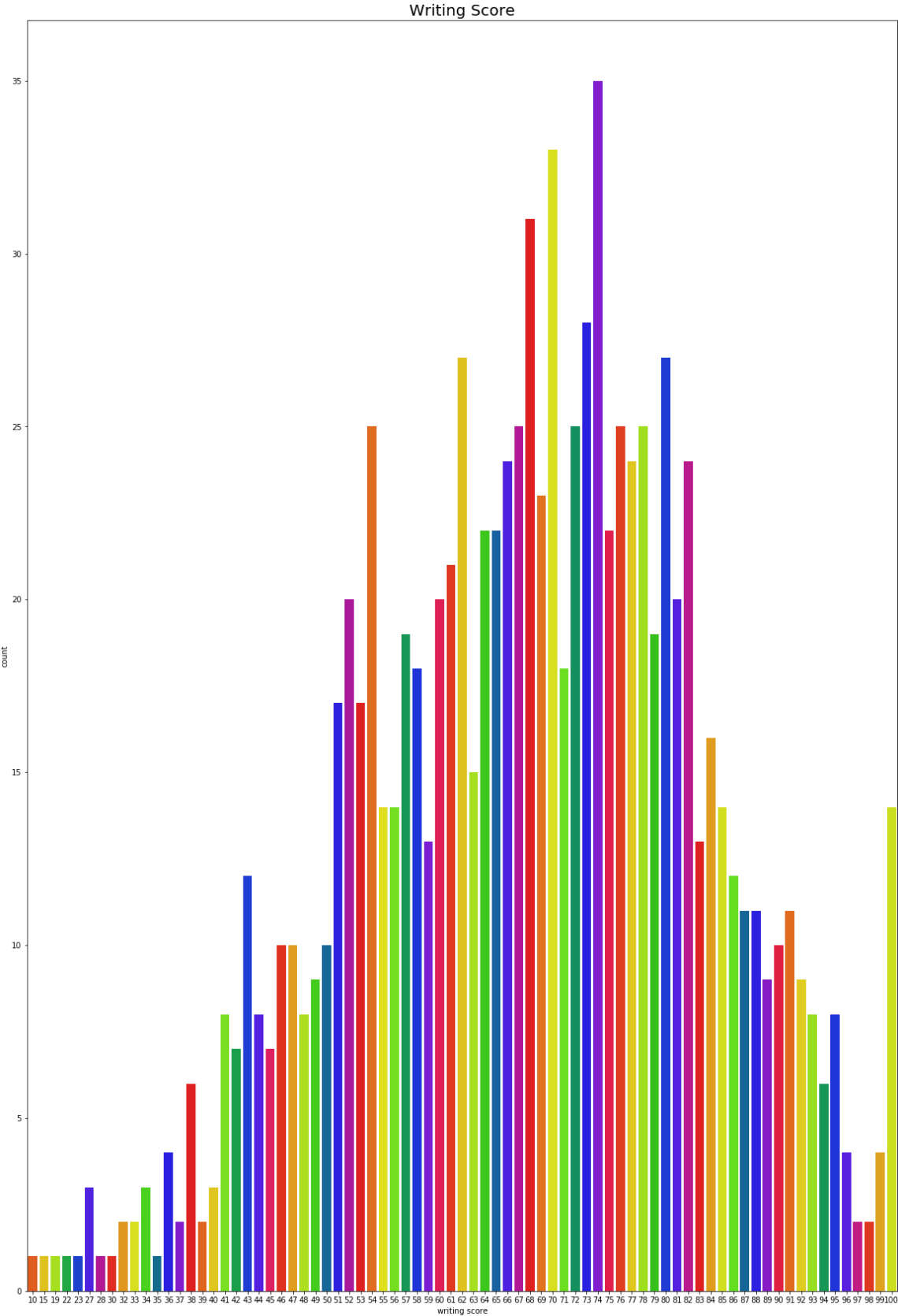






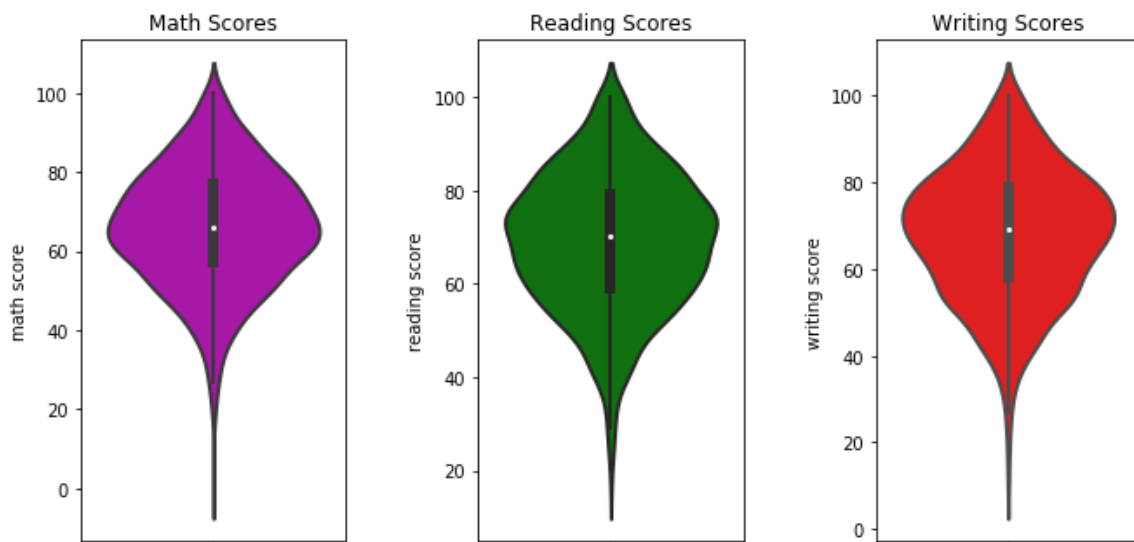
In [40]:

```
plt.rcParams['figure.figsize'] = {20, 30}
sns.countplot(df['writing score'], palette = 'prism')
plt.title('Writing Score',fontsize = 20)
plt.show()
```



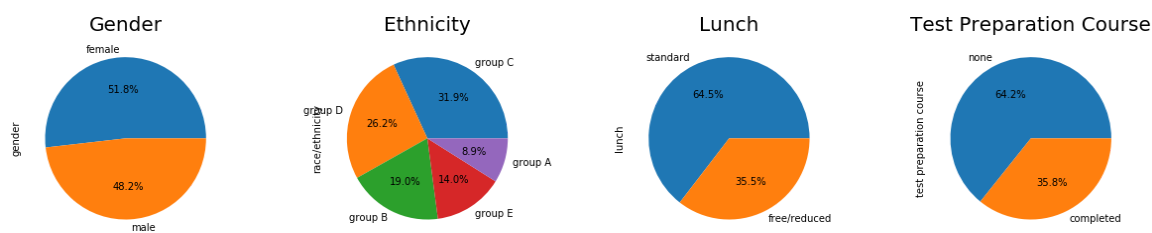
In [9]:

```
plt.figure(figsize=(15,5))
plt.subplots_adjust(left=0.125, bottom=0.1, right=0.9, top=0.9, wspace=0.5, hspace=0.2)
plt.subplot(141)
plt.title('Math Scores')
sns.violinplot(y='math score',data=df,color='m',linewidth=2)
plt.subplot(142)
plt.title('Reading Scores')
sns.violinplot(y='reading score',data=df,color='g',linewidth=2)
plt.subplot(143)
plt.title('Writing Scores')
sns.violinplot(y='writing score',data=df,color='r',linewidth=2)
plt.show()
```



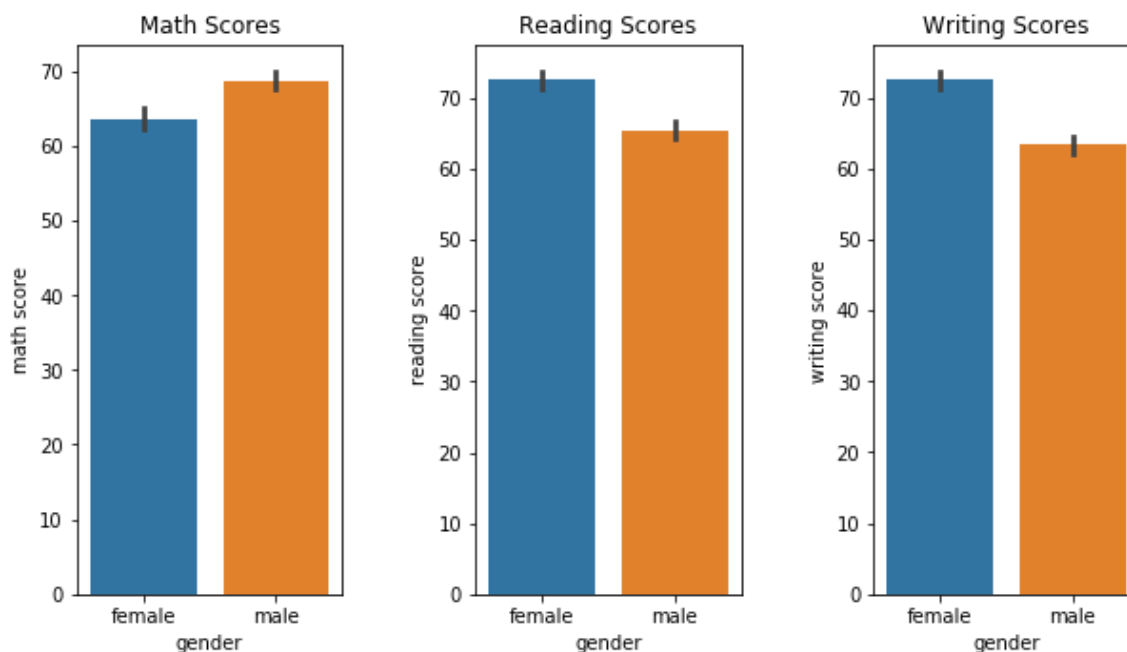
In [10]:

```
plt.figure(figsize=(20,10))
plt.subplots_adjust(left=0.125, bottom=0.1, right=0.9, top=0.9, wspace=0.5, hspace=0.2)
plt.subplot(141)
plt.title('Gender',fontsize =20)
df['gender'].value_counts().plot.pie(autopct='%1.1f%%')
plt.subplot(142)
plt.title('Ethnicity',fontsize =20)
df['race/ethnicity'].value_counts().plot.pie(autopct='%1.1f%%')
plt.subplot(143)
plt.title('Lunch',fontsize =20)
df['lunch'].value_counts().plot.pie(autopct='%1.1f%%')
plt.subplot(144)
plt.title('Test Preparation Course',fontsize =20)
df['test preparation course'].value_counts().plot.pie(autopct='%1.1f%%')
plt.show()
```



In [11]:

```
plt.figure(figsize=(10,5))
plt.subplots_adjust(left=0.125, bottom=0.1, right=0.9, top=0.9, wspace=0.5, hspace=0.2)
plt.subplot(131)
plt.title('Math Scores')
sns.barplot(x='gender', y='math score', data=df)
plt.subplot(132)
plt.title('Reading Scores')
sns.barplot(x='gender', y='reading score', data=df)
plt.subplot(133)
plt.title('Writing Scores')
sns.barplot(x='gender', y='writing score', data=df)
plt.show()
```



In [12]:

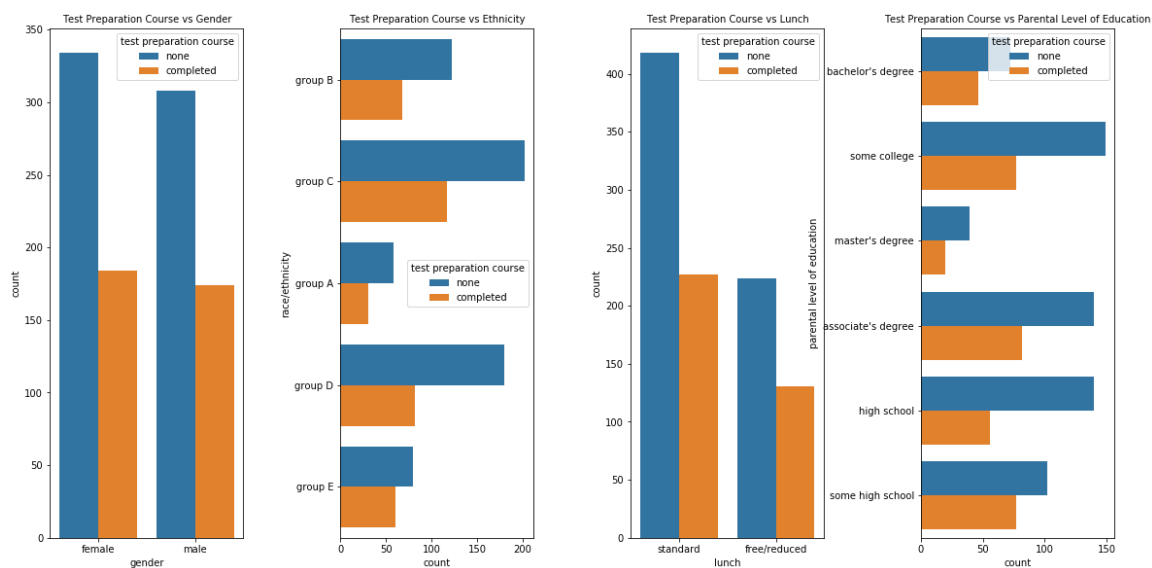
```
plt.figure(figsize=(25,20))
plt.subplots_adjust(left=0.125, bottom=0.1, right=0.9, top=0.9, wspace=0.5, hspace=0.2)
plt.subplot(251)
plt.title('Test Preparation Course vs Gender', fontsize = 10)
sns.countplot(hue='test preparation course', x='gender', data=df)

plt.subplot(252)
plt.title('Test Preparation Course vs Ethnicity', fontsize = 10)
sns.countplot(hue='test preparation course', y='race/ethnicity', data=df)

plt.subplot(253)
plt.title('Test Preparation Course vs Lunch', fontsize = 10)
sns.countplot(hue='test preparation course', x='lunch', data=df)

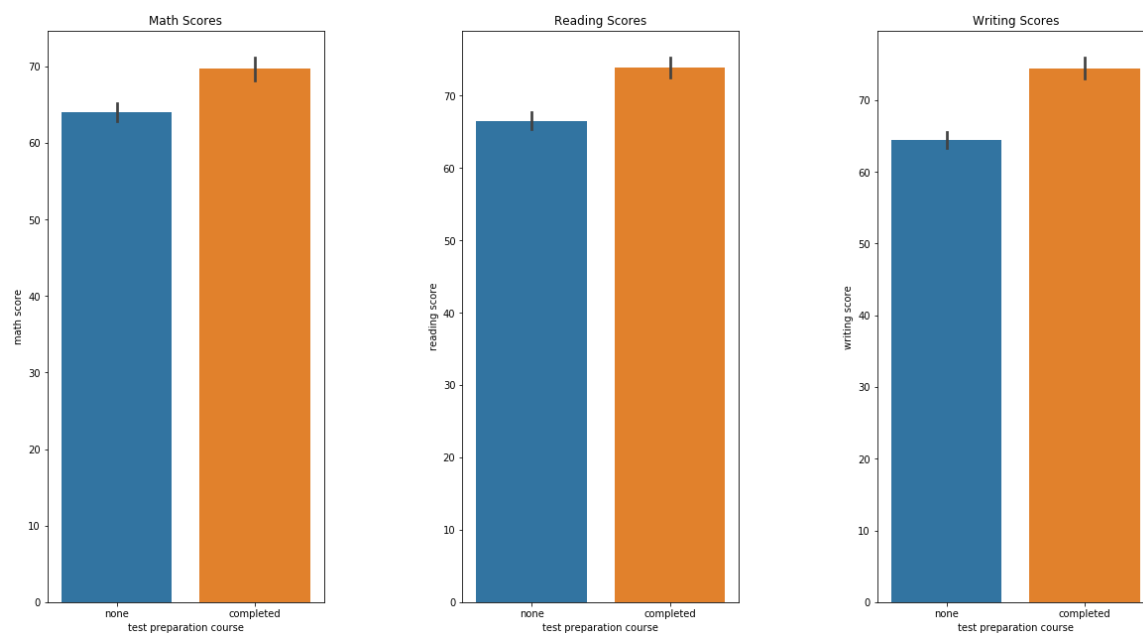
plt.subplot(254)
plt.title('Test Preparation Course vs Parental Level of Education', fontsize = 10)
sns.countplot(hue='test preparation course', y='parental level of education', data=df)

plt.show()
```



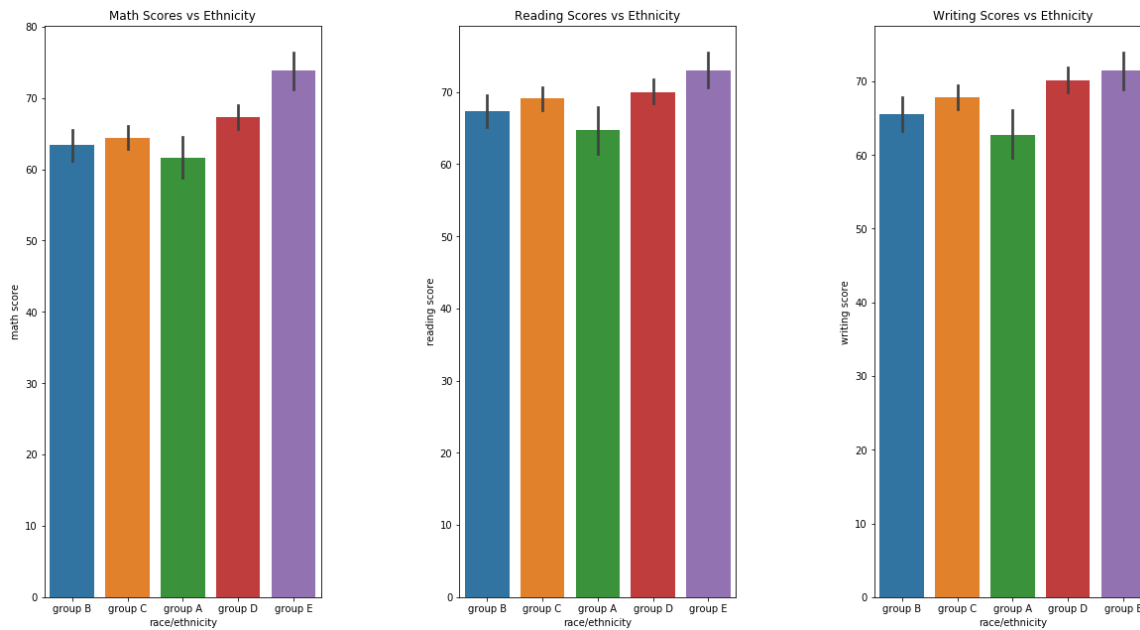
In [13]:

```
plt.figure(figsize=(20,10))
plt.subplots_adjust(left=0.125, bottom=0.1, right=0.9, top=0.9, wspace=0.5, hspace=0.2)
plt.subplot(131)
plt.title('Math Scores')
sns.barplot(x='test preparation course', y='math score', data=df)
plt.subplot(132)
plt.title('Reading Scores')
sns.barplot(x='test preparation course', y='reading score', data=df)
plt.subplot(133)
plt.title('Writing Scores')
sns.barplot(x='test preparation course', y='writing score', data=df)
plt.show()
```



In [14]:

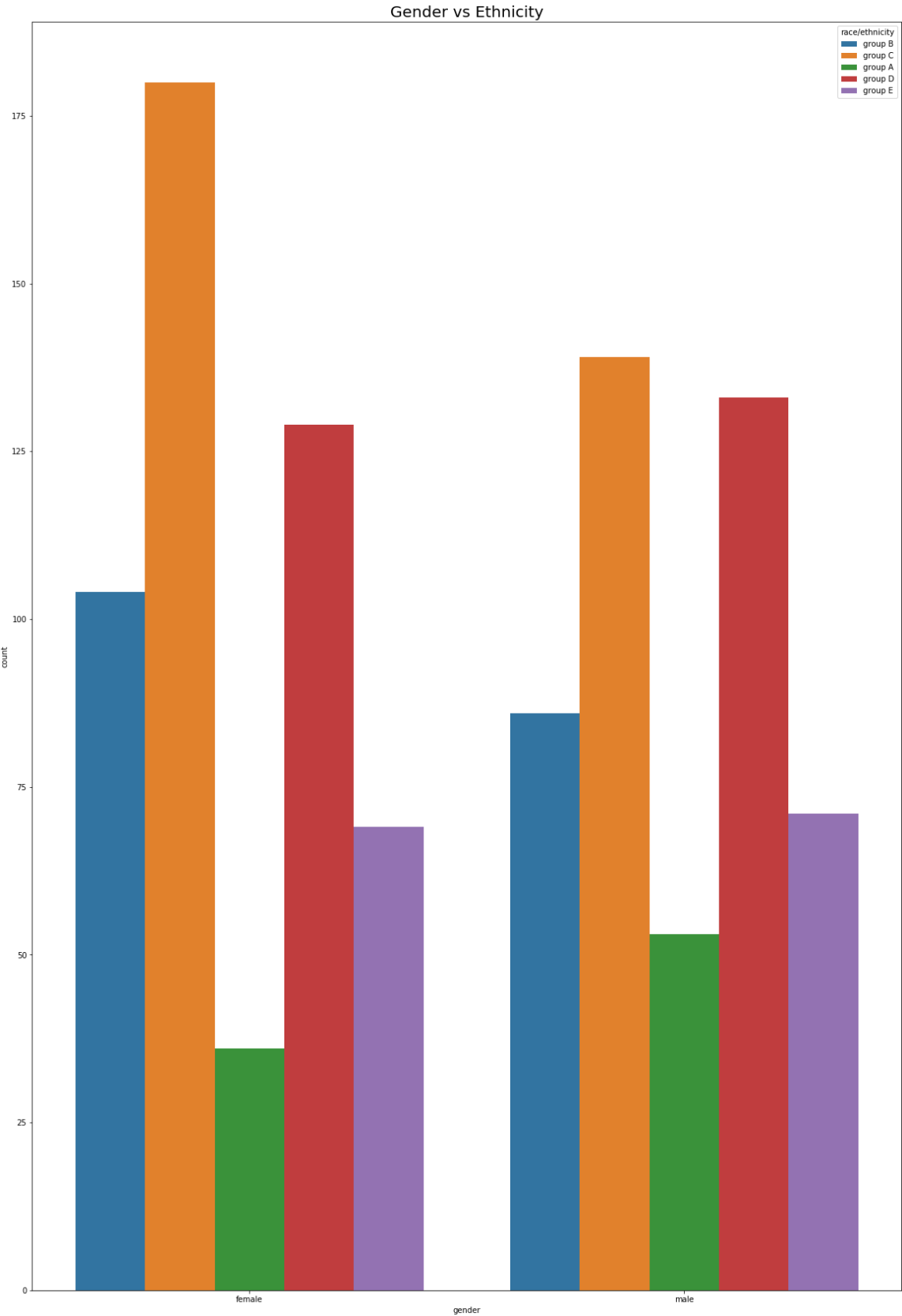
```
plt.figure(figsize=(20,10))
plt.subplots_adjust(left=0.125, bottom=0.1, right=0.9, top=0.9, wspace=0.5, hspace=0.2)
plt.subplot(131)
plt.title('Math Scores vs Ethnicity')
sns.barplot(x='race/ethnicity', y='math score', data=df)
plt.subplot(132)
plt.title('Reading Scores vs Ethnicity')
sns.barplot(x='race/ethnicity', y='reading score', data=df)
plt.subplot(133)
plt.title('Writing Scores vs Ethnicity')
sns.barplot(x='race/ethnicity', y='writing score', data=df)
plt.show()
```





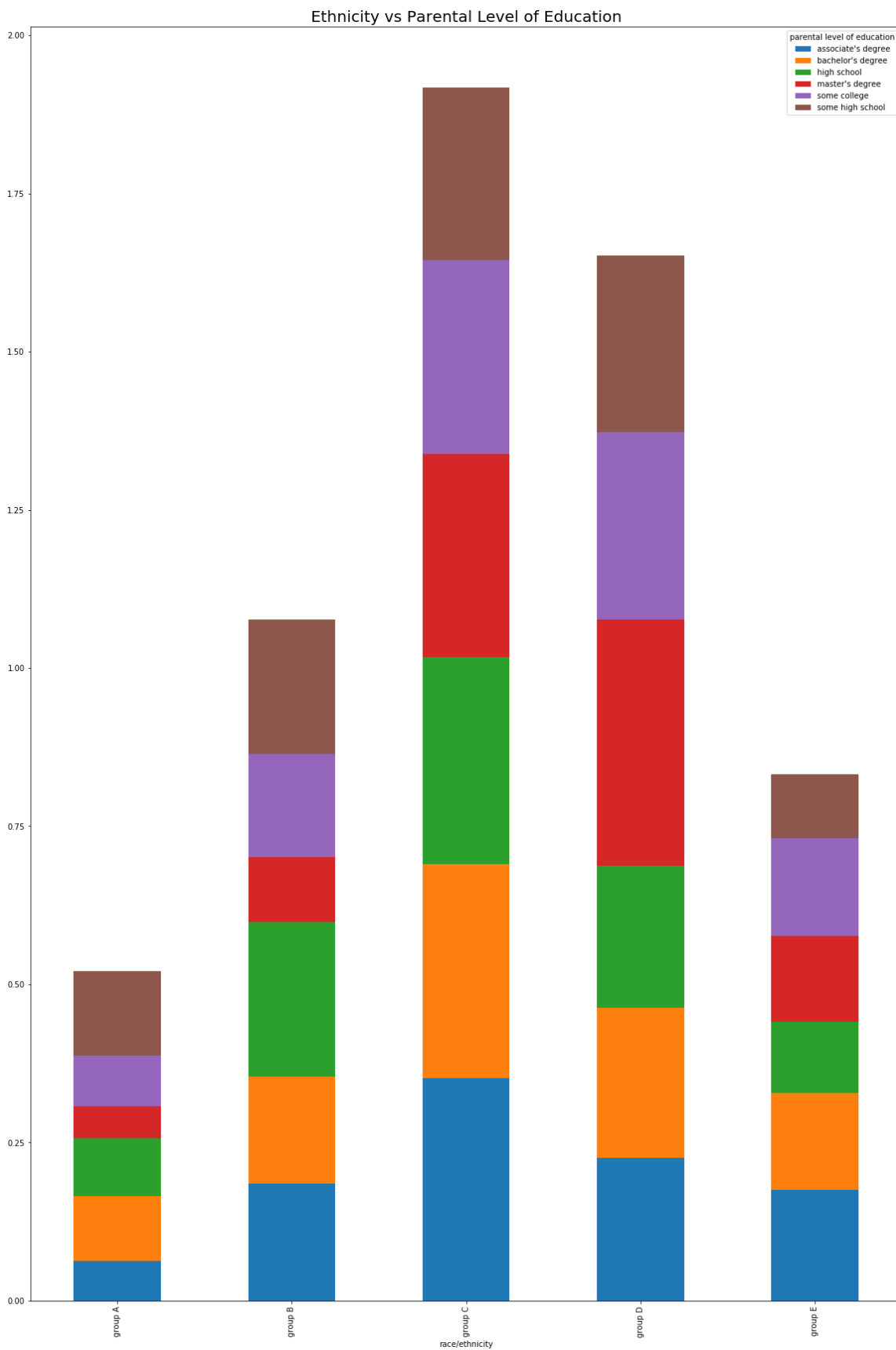
In [15]:

```
plt.title('Gender vs Ethnicity',fontsize = 20)
sns.countplot(x='gender', hue='race/ethnicity', data=df)
plt.show()
```



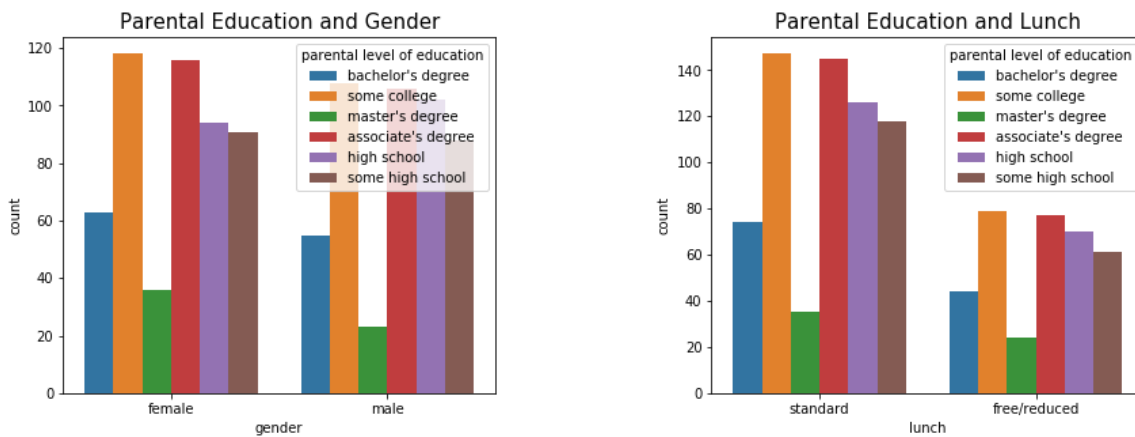
In [16]:

```
pr=pd.crosstab(df['race/ethnicity'],df['parental level of education'],normalize=1)
pr.plot.bar(stacked=True)
plt.title('Ethnicity vs Parental Level of Education',fontsize = 20)
plt.show()
```



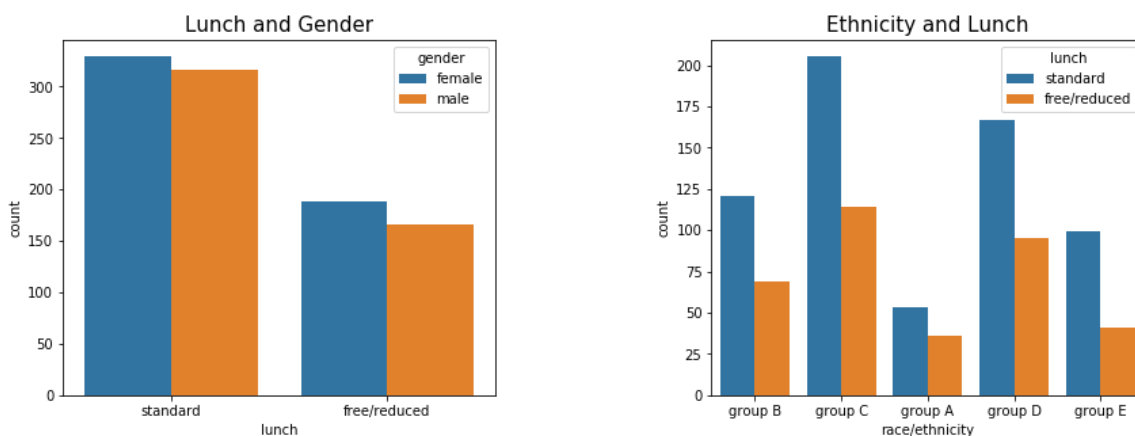
In [17]:

```
plt.figure(figsize=(40,10))
plt.subplots_adjust(left=0.125, bottom=0.1, right=0.9, top=0.9, wspace=0.5, hspace=0.2)
plt.subplot(251)
plt.title('Parental Education and Gender', fontsize=15)
sns.countplot(x='gender', hue='parental level of education', data=df)
plt.subplot(252)
plt.title('Parental Education and Lunch', fontsize=15)
sns.countplot(x='lunch', hue='parental level of education', data=df)
plt.show()
```



In [18]:

```
plt.figure(figsize=(40,10))
plt.subplots_adjust(left=0.125, bottom=0.1, right=0.9, top=0.9, wspace=0.5, hspace=0.2)
plt.subplot(251)
plt.title('Lunch and Gender', fontsize=15)
sns.countplot(x='lunch', hue='gender', data=df)
plt.subplot(252)
plt.title('Ethnicity and Lunch', fontsize=15)
sns.countplot(x='race/ethnicity', hue='lunch', data=df)
plt.show()
```



In [19]:

```
df['total_score'] = df['math score'] + df['reading score'] + df['writing score']
```

In [20]:

```
df.append(['total_score'], ignore_index=True, verify_integrity=False, sort=None)
```

Out[20]:

	0	gender	lunch	math score	parental level of education	race/ethnicity	reading score	test preparation course
0	NaN	female	standard	72.0	bachelor's degree	group B	72.0	none
1	NaN	female	standard	69.0	some college	group C	90.0	completed
2	NaN	female	standard	90.0	master's degree	group B	95.0	none
3	NaN	male	free/reduced	47.0	associate's degree	group A	57.0	none
4	NaN	male	standard	76.0	some college	group C	78.0	none
...	...	...	...	...	...	...	...	...
996	NaN	male	free/reduced	62.0	high school	group C	55.0	none
997	NaN	female	free/reduced	59.0	high school	group C	71.0	completed
998	NaN	female	standard	68.0	some college	group D	78.0	completed
999	NaN	female	free/reduced	77.0	some college	group D	86.0	none
1000	total_score	NaN	NaN	NaN	NaN	NaN	NaN	NaN

1001 rows × 10 columns

In [21]:

```
df['percentage']=df['total_score']/300*100
```

In [22]:

```
df
```

Out[22]:

	gender	race/ethnicity	parental level of education	lunch	test preparation course	math score	reading score	writing score	tot
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 × 10 columns

In [23]:

```
per = df['percentage']
```

In [24]:

```

g=[]
def determine_grade():
    for i in per:
        if int(i) >= 85 and int(i) <= 100:
            g.append('A')
        elif int(i) >= 70 and int(i) < 85:
            g.append('B')
        elif int(i) >= 55 and int(i) < 70:
            g.append('C')
        elif int(i) >= 36 and int(i) < 55:
            g.append('D')
        elif int(i) >= 0 and int(i) < 35:
            g.append('E')
determine_grade()

```

In [26]:

```
df.insert(10, 'grade', g, allow_duplicates=False)
```

In [27]:

df

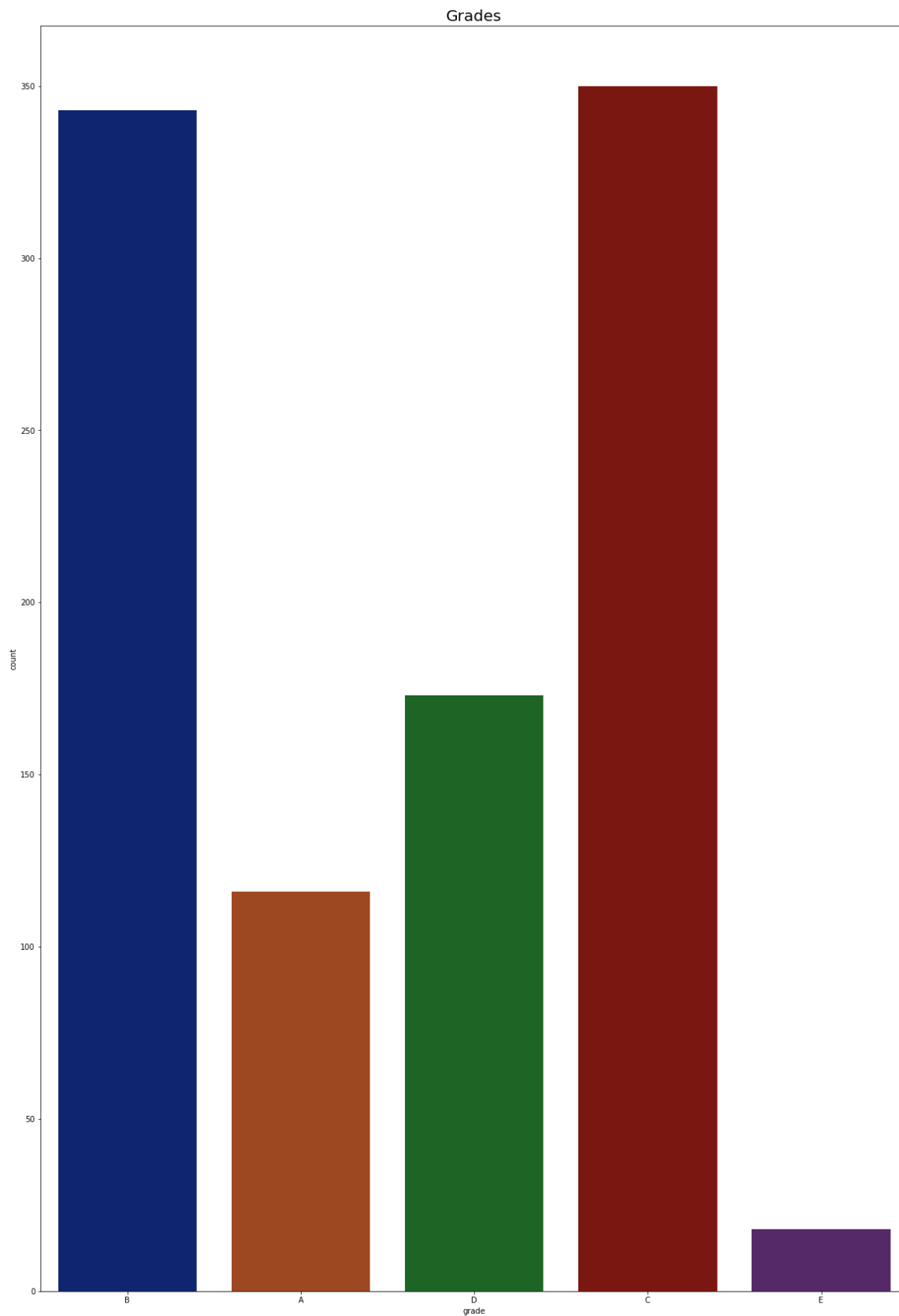
Out[27]:

	gender	race/ethnicity	parental level of education	lunch	test preparation course	math score	reading score	writing score	tot
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 × 11 columns

In [29]:

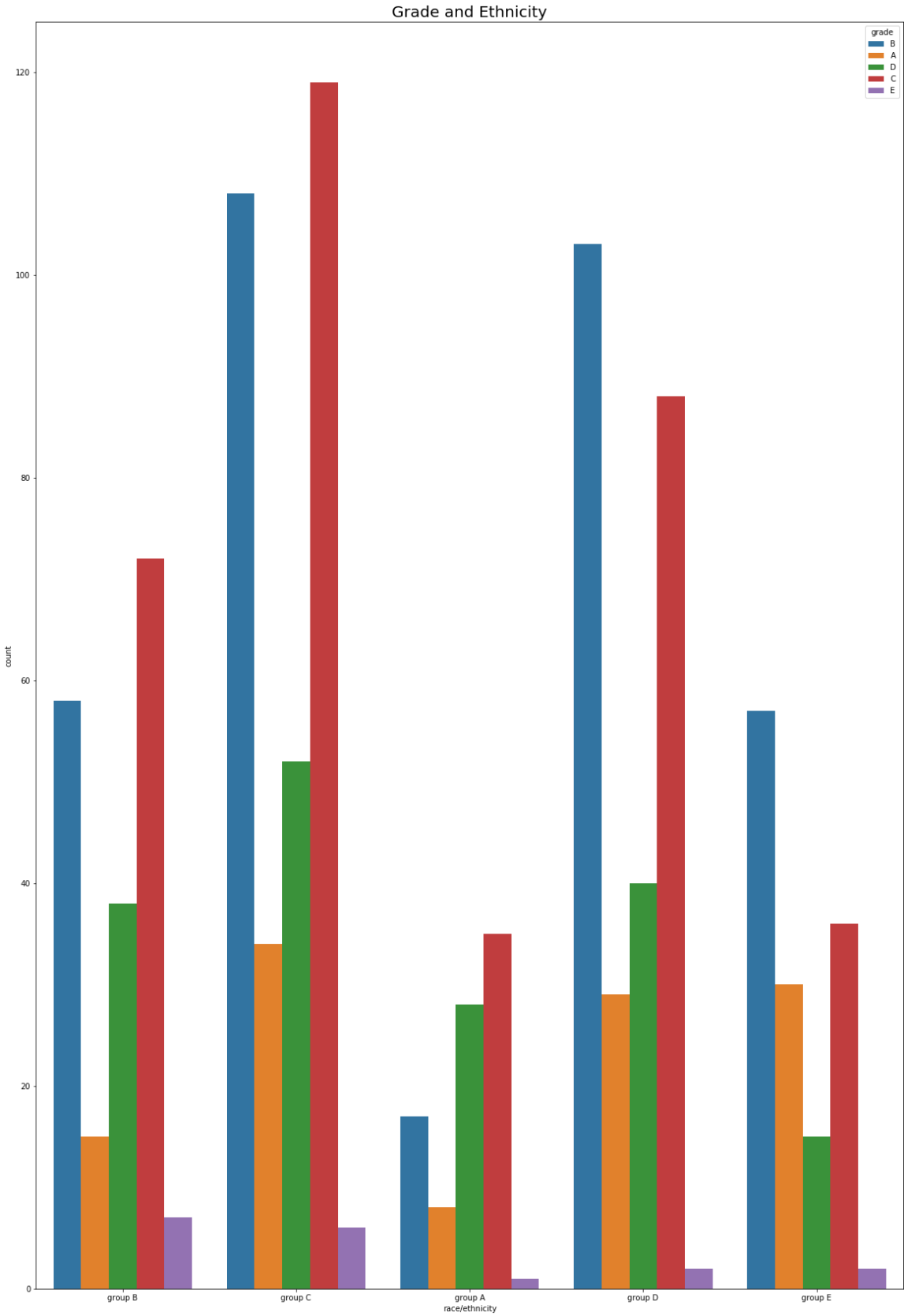
```
plt.rcParams['figure.figsize'] = {20, 30}  
sns.countplot(df['grade'], palette = 'dark')  
plt.title('Grades', fontsize = 20)  
plt.show()
```

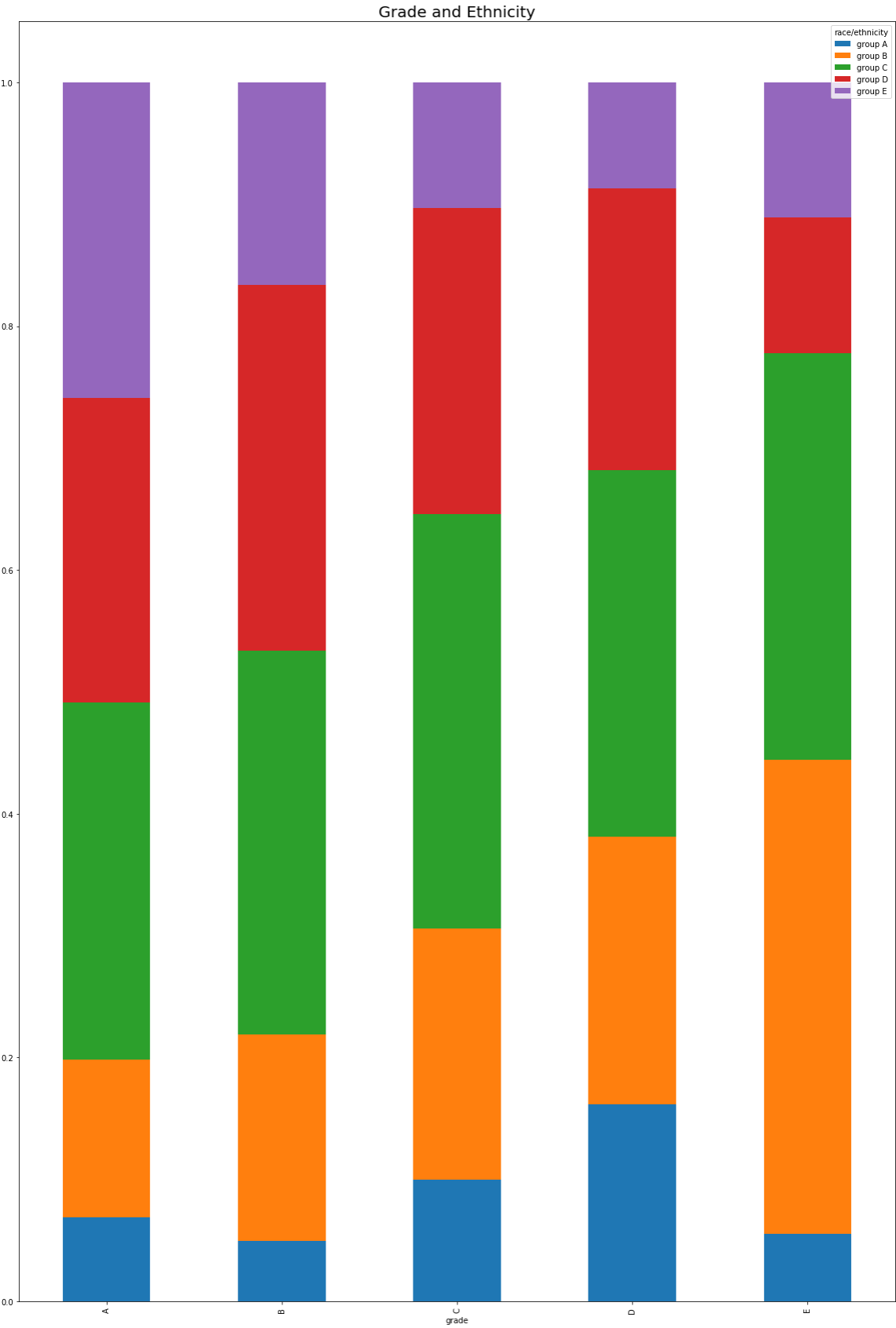




In [31]:

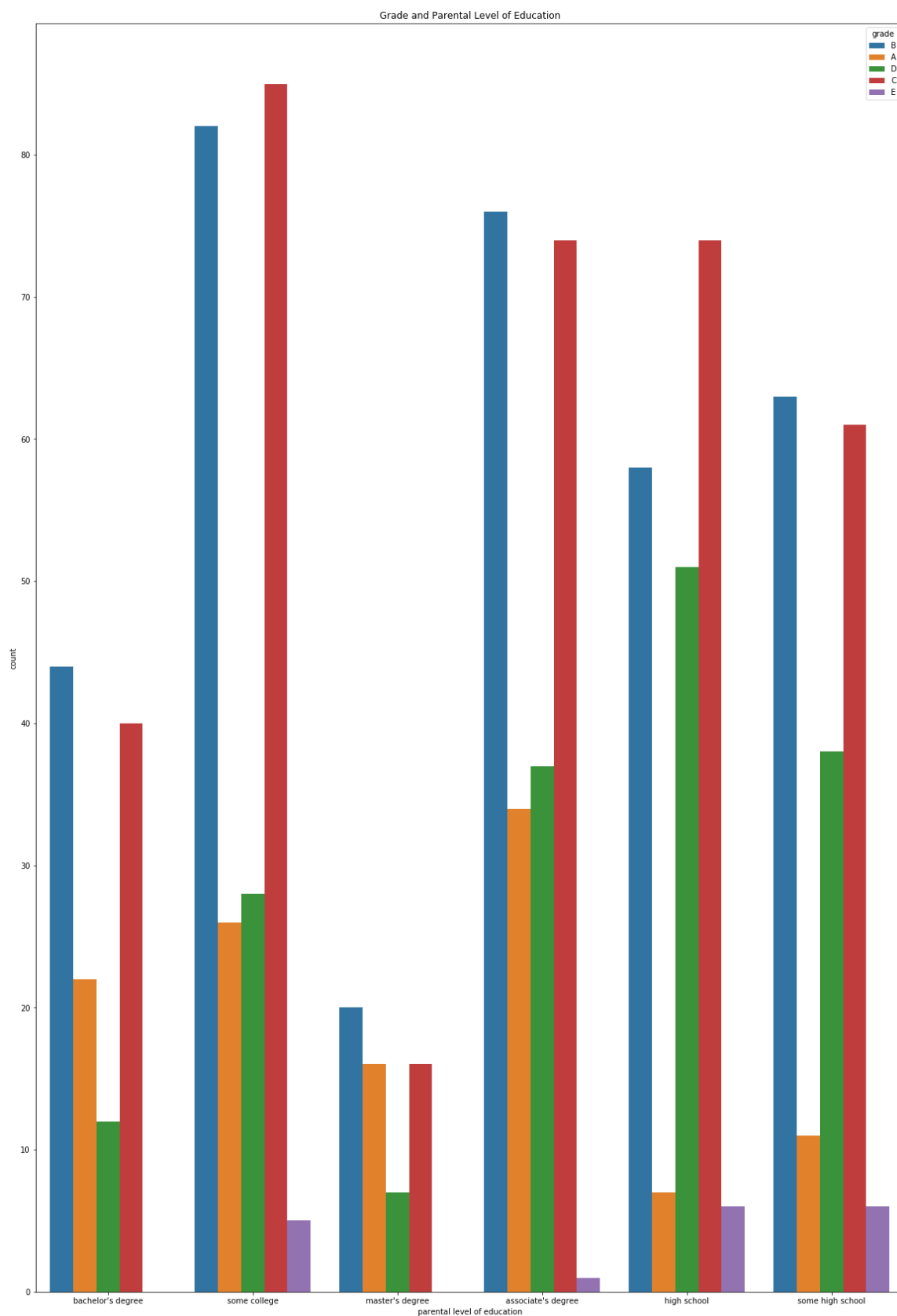
```
plt.title('Grade and Ethnicity',fontsize=20)
sns.countplot(x='race/ethnicity', hue='grade', data=df)
gr=pd.crosstab(df['grade'],df['race/ethnicity'],normalize=0)
gr.plot.bar(stacked=True)
plt.title('Grade and Ethnicity',fontsize=20)
plt.show()
```





In [32]:

```
plt.title('Grade and Parental Level of Education')  
sns.countplot(x='parental level of education', hue='grade', data=df)  
plt.show()
```



In [35]:

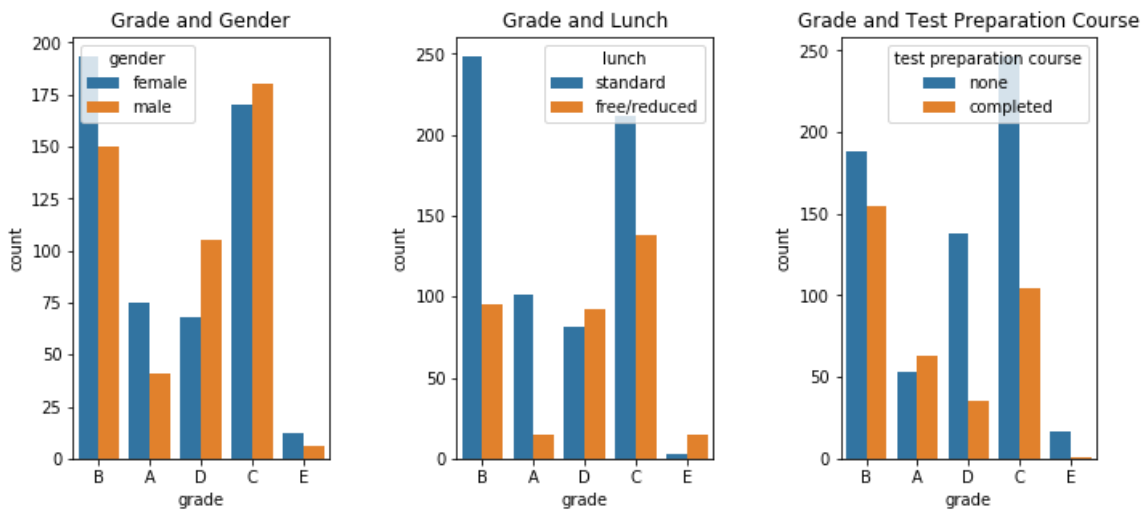
```
plt.figure(figsize=(20,10))
plt.subplots_adjust(left=0.125, bottom=0.1, right=0.9, top=0.9, wspace=0.5, hspace=0.2)

plt.subplot(251)
plt.title('Grade and Gender')
sns.countplot(hue='gender', x='grade', data=df)

plt.subplot(252)
plt.title('Grade and Lunch')
sns.countplot(hue='lunch', x='grade', data=df)

plt.subplot(253)
plt.title('Grade and Test Preparation Course')
sns.countplot(hue='test preparation course', x='grade', data=df)

plt.show()
```



In [42]:

```
plt.figure(figsize=(60,50))  
  
plt.subplot(141)  
plt.title('Grade',fontsize =20)  
df['grade'].value_counts().plot.pie(autopct='%1.1f%%')
```

Out[42]:

<matplotlib.axes.\_subplots.AxesSubplot at 0x2119ef16f08>

