

## Assignment 1.2:

### 1. Average Test Scores by Race.

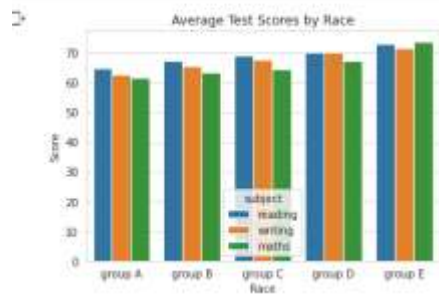
```
# Group the data by race and calculate the mean scores for each subject
mean_scores = stud.groupby('race')[['reading', 'writing', 'maths']].mean()

# Reset the index to create a 'race' column
mean_scores = mean_scores.reset_index()

# Melt the data to create a long-form dataframe for plotting
mean_scores_melt = pd.melt(mean_scores, id_vars='race', var_name='subject', value_name='score')

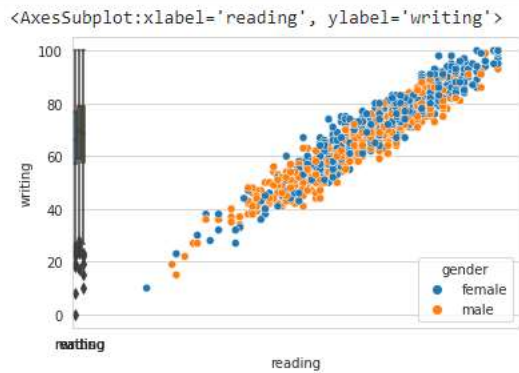
# Plot the data as a grouped bar chart
sns.barplot(x='race', y='score', hue='subject', data=mean_scores_melt)

plt.title('Average Test Scores by Race')
plt.xlabel('Race')
plt.ylabel('Score')
plt.show()
```



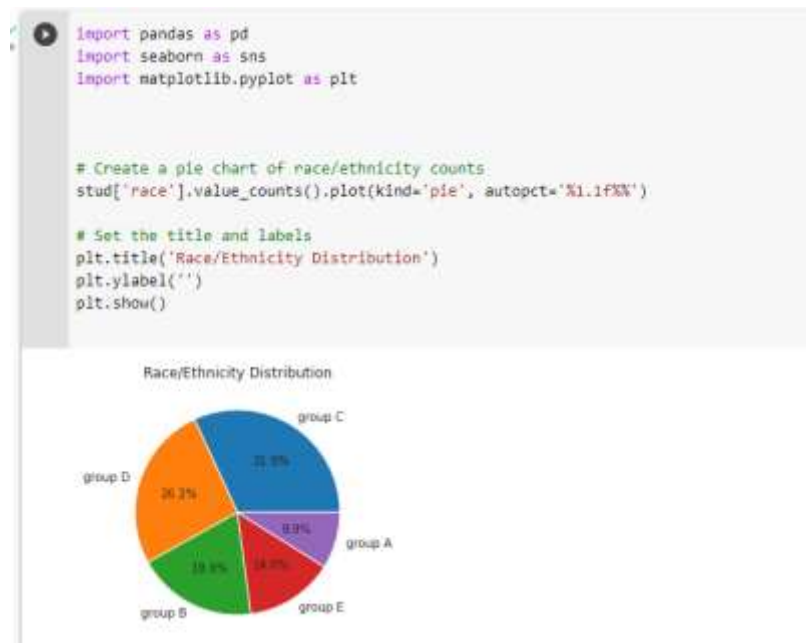
### 2. Reading and writing using gender. The female scored more in writing and reading than male.

```
[21]
import seaborn as sns
sns.boxplot(data=stud)
sns.scatterplot(x='reading', y='writing', hue='gender', data=stud)
```



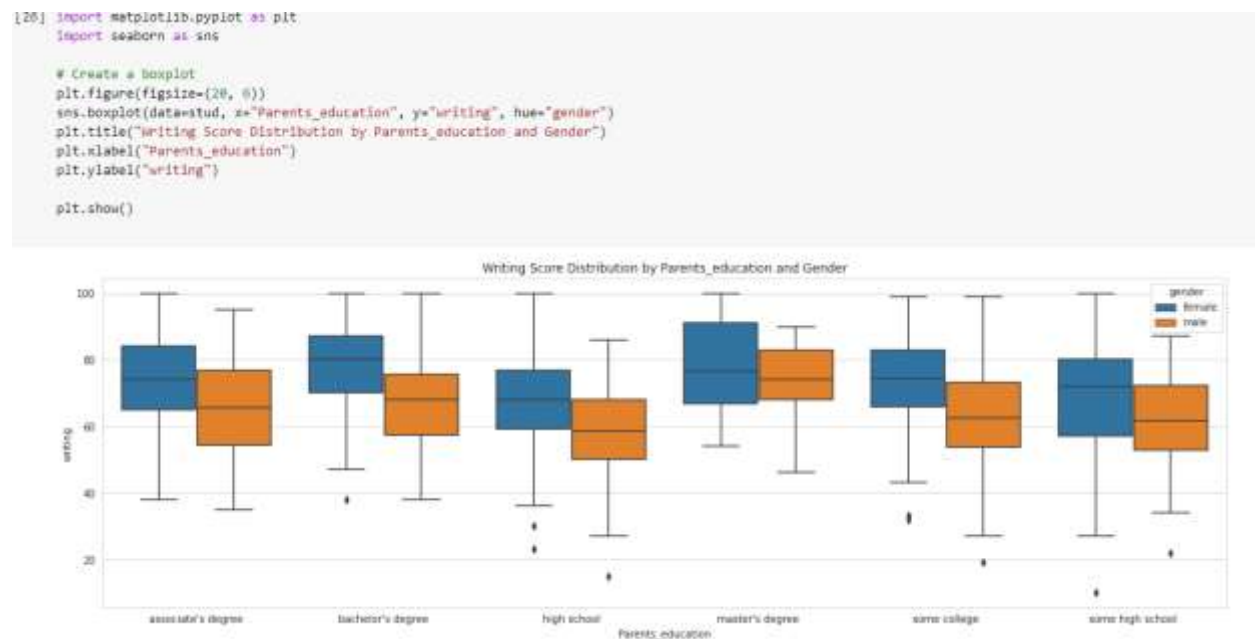
### 3.Race/ethnicity Distribution.

The majority of the race/ethnicity distribution is group C.



### 4.Writing Score Distribution by parents' education and gender

The female writing score is always more than male writing score,



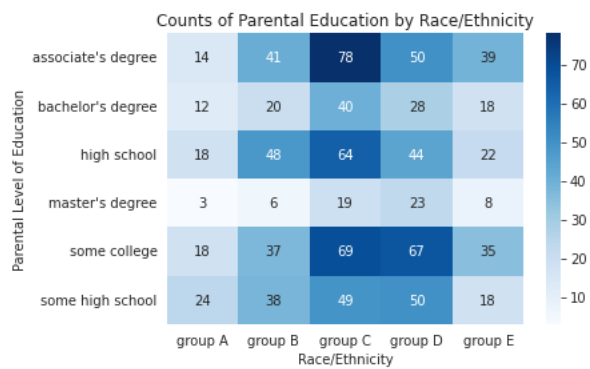
## 5. Counts of Parental education by Race/ethnicity.

```
[29] pivot = stud.pivot_table(index='Parents_education', columns='race', values='maths', aggfunc='count')

# Create a heatmap
sns.heatmap(pivot, cmap='Blues', annot=True, fmt='.0f')

# Set the plot title and axis labels
plt.title('Counts of Parental Education by Race/Ethnicity')
plt.xlabel('Race/Ethnicity')
plt.ylabel('Parental Level of Education')

# Show the plot
plt.show()
```



## 6.

```
[30] import pandas as pd
import matplotlib.pyplot as plt

# compute the total time spent on each subject
math_total = stud['maths'].sum()
reading_total = stud['reading'].sum()
writing_total = stud['writing'].sum()

# compute the percentage of time spent on each subject
math_percentage = math_total / (math_total + reading_total + writing_total) * 100
reading_percentage = reading_total / (math_total + reading_total + writing_total) * 100
writing_percentage = writing_total / (math_total + reading_total + writing_total) * 100

# create a pie chart with the computed percentages
labels = ['math', 'reading', 'writing']
sizes = [math_percentage, reading_percentage, writing_percentage]
colors = ['yellow', 'green', 'blue']
explode = (0.05, 0.05, 0.05) # explode the slices a bit to highlight them

fig, ax = plt.subplots()
ax.pie(sizes, colors=colors, labels=labels, autopct='%1.1f%%', startangle=90, pctdistance=0.85, explode=explode)

# draw a circle at the center of the chart to make it look like a donut
centre_circle = plt.Circle((0,0),0.75,fc='white')
fig = plt.gcf()
fig.gca().add_artist(centre_circle)

# set the aspect ratio to be equal so that the chart looks circular
ax.axis('equal')

plt.tight_layout()
plt.show()
```

