Student Performance in Exam Analysis Using Python

The Student Performance Analysis project investigates the correlation between various demographic and educational factors and student performance in math, reading, and writing. By examining attributes such as gender, race/ethnicity, parental level of education, lunch type, and test preparation course completion, the project aims to uncover insights that can inform educational strategies and interventions aimed at improving student outcomes. Through descriptive statistics and data visualization techniques, the analysis seeks to identify disparities in performance among different student groups and highlight potential areas for targeted interventions to address these disparities and promote academic success.

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

```
df = pd.read_csv('14_Student Performance in Exam Analysis.csv')
```

```
columns = df.columns
print(columns)
```

```
Index(['gender', 'race/ethnicity', 'parental level of education', 'lunch',
       'test preparation course', 'math score', 'reading score',
       'writing score'],
      dtype='object')
```

```
print(df.head())
print(df.info())
print(df.describe())
```

```
   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

  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
<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
None
       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
```

```
print("\nDescriptive Statistics:")
print(df.describe(include='all'))
```

```
Descriptive Statistics:
```

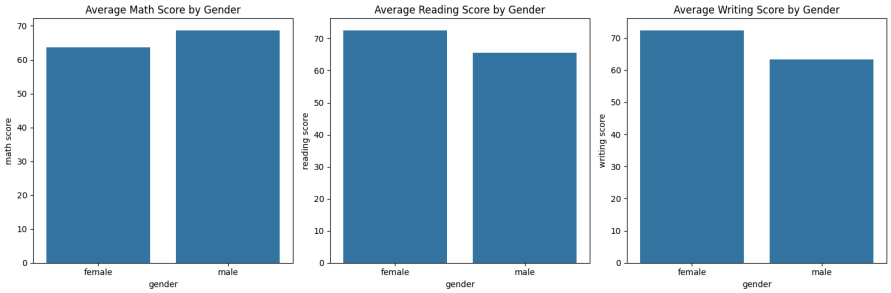
	gender	race/ethnicity	parental level of education	lunch	\
count	1000	1000	1000	1000	
unique	2	5	6	2	
top	female	group C	some college	standard	
freq	518	319	226	645	
mean	NaN	NaN	NaN	NaN	
std	NaN	NaN	NaN	NaN	
min	NaN	NaN	NaN	NaN	
25%	NaN	NaN	NaN	NaN	
50%	NaN	NaN	NaN	NaN	
75%	NaN	NaN	NaN	NaN	
max	NaN	NaN	NaN	NaN	

	test preparation course	math score	reading score	writing score
count	1000	1000.00000	1000.000000	1000.000000
unique	2	NaN	NaN	NaN
top	none	NaN	NaN	NaN
freq	642	NaN	NaN	NaN
mean	NaN	66.08900	69.169000	68.054000
std	NaN	15.16308	14.600192	15.195657
min	NaN	0.00000	17.000000	10.000000
25%	NaN	57.00000	59.000000	57.750000
50%	NaN	66.00000	70.000000	69.000000
75%	NaN	77.00000	79.000000	79.000000
max	NaN	100.00000	100.000000	100.000000

Gender distribution

```
plt.figure(figsize=(15, 5))
plt.subplot(1, 3, 1)
sns.barplot(x='gender', y='math score', data=df, ci=None)
plt.title('Average Math Score by Gender')

plt.subplot(1, 3, 2)
sns.barplot(x='gender', y='reading score', data=df, ci=None)
plt.title('Average Reading Score by Gender')

plt.subplot(1, 3, 3)
sns.barplot(x='gender', y='writing score', data=df, ci=None)
plt.title('Average Writing Score by Gender')

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

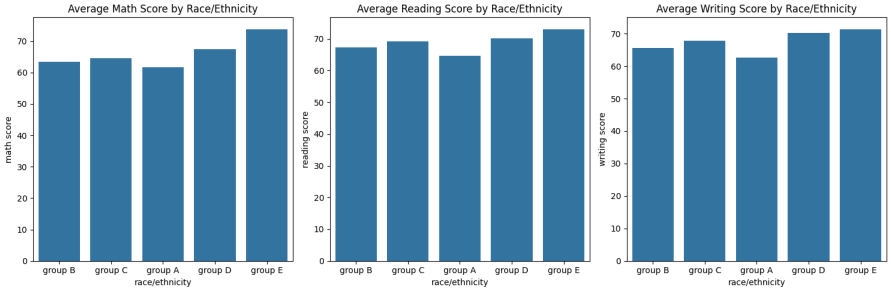
```
<ipython-input-39-32f6ea9f8528>:3: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar=None` for the same effect.

  sns.barplot(x='gender', y='math score', data=df, ci=None)
<ipython-input-39-32f6ea9f8528>:7: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar=None` for the same effect.

  sns.barplot(x='gender', y='reading score', data=df, ci=None)
<ipython-input-39-32f6ea9f8528>:11: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar=None` for the same effect.

  sns.barplot(x='gender', y='writing score', data=df, ci=None)
```



Race/ethnicity distribution

```
plt.figure(figsize=(15, 5))
plt.subplot(1, 3, 1)
sns.barplot(x='race/ethnicity', y='math score', data=df, ci=None)
plt.title('Average Math Score by Race/Ethnicity')

plt.subplot(1, 3, 2)
sns.barplot(x='race/ethnicity', y='reading score', data=df, ci=None)
plt.title('Average Reading Score by Race/Ethnicity')

plt.subplot(1, 3, 3)
sns.barplot(x='race/ethnicity', y='writing score', data=df, ci=None)
plt.title('Average Writing Score by Race/Ethnicity')

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

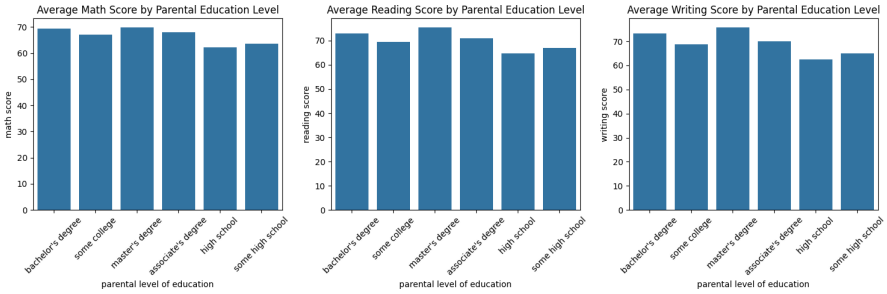
```
<ipython-input-40-4aacd387ea3b>:3: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar=None` for the same effect.

  sns.barplot(x='race/ethnicity', y='math score', data=df, ci=None)
<ipython-input-40-4aacd387ea3b>:7: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar=None` for the same effect.

  sns.barplot(x='race/ethnicity', y='reading score', data=df, ci=None)
<ipython-input-40-4aacd387ea3b>:11: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar=None` for the same effect.

  sns.barplot(x='race/ethnicity', y='writing score', data=df, ci=None)
```



Parental education impact

```
plt.figure(figsize=(15, 5))
plt.subplot(1, 3, 1)
sns.barplot(x='parental level of education', y='math score', data=df, ci=None)
plt.title('Average Math Score by Parental Education Level')
plt.xticks(rotation=45)

plt.subplot(1, 3, 2)
sns.barplot(x='parental level of education', y='reading score', data=df, ci=None)
plt.title('Average Reading Score by Parental Education Level')
plt.xticks(rotation=45)

plt.subplot(1, 3, 3)
sns.barplot(x='parental level of education', y='writing score', data=df, ci=None)
plt.title('Average Writing Score by Parental Education Level')
plt.xticks(rotation=45)

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

```
<ipython-input-41-8884b9b10a4b>:3: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar=None` for the same effect.

  sns.barplot(x='parental level of education', y='math score', data=df, ci=None)
<ipython-input-41-8884b9b10a4b>:8: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar=None` for the same effect.

  sns.barplot(x='parental level of education', y='reading score', data=df, ci=None)
<ipython-input-41-8884b9b10a4b>:13: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar=None` for the same effect.

  sns.barplot(x='parental level of education', y='writing score', data=df, ci=None)
```



Lunch type impact

```
plt.figure(figsize=(15, 5))
plt.subplot(1, 3, 1)
sns.barplot(x='lunch', y='math score', data=df, ci=None)
plt.title('Average Math Score by Lunch Type')

plt.subplot(1, 3, 2)
sns.barplot(x='lunch', y='reading score', data=df, ci=None)
plt.title('Average Reading Score by Lunch Type')

plt.subplot(1, 3, 3)
sns.barplot(x='lunch', y='writing score', data=df, ci=None)
plt.title('Average Writing Score by Lunch Type')

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

```
<ipython-input-42-7bc40d24f736>:3: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar=None` for the same effect.

  sns.barplot(x='lunch', y='math score', data=df, ci=None)
<ipython-input-42-7bc40d24f736>:7: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar=None` for the same effect.

  sns.barplot(x='lunch', y='reading score', data=df, ci=None)
<ipython-input-42-7bc40d24f736>:11: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar=None` for the same effect.

  sns.barplot(x='lunch', y='writing score', data=df, ci=None)
```

Test preparation course impact

```
plt.figure(figsize=(15, 5))
plt.subplot(1, 3, 1)
sns.barplot(x='test preparation course', y='math score', data=df, ci=None)
plt.title('Average Math Score by Test Preparation Course')

plt.subplot(1, 3, 2)
sns.barplot(x='test preparation course', y='reading score', data=df, ci=None)
plt.title('Average Reading Score by Test Preparation Course')

plt.subplot(1, 3, 3)
sns.barplot(x='test preparation course', y='writing score', data=df, ci=None)
plt.title('Average Writing Score by Test Preparation Course')

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