Introduction

The "Student Result Analysis" project aims to provide insights into student performance based on various factors. The analysis utilizes Python programming language along with popular data manipulation and visualization libraries such as NumPy, Pandas, Matplotlib, and Seaborn.

Dataset Overview

This dataset contains information about students, including demographic details, test scores, and other relevant factors.

Import libraries

In [1]:

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

In [2]:

```
df=pd.read_csv("student_scores.csv")
```

In [3]:

```
df.head()
```

Out[3]:

	Unnamed: 0	Gender	EthnicGroup	ParentEduc	LunchType	TestPrep	ParentMaritalStatus	PracticeSport	IsFirstChild	NrSiblings	Transp
0	0	female	NaN	bachelor's degree	standard	none	married	regularly	yes	3.0	
1	1	female	group C	some college	standard	NaN	married	sometimes	yes	0.0	
2	2	female	group B	master's degree	standard	none	single	sometimes	yes	4.0	
3	3	male	group A	associate's degree	free/reduced	none	married	never	no	1.0	
4	4	male	group C	some college	standard	none	married	sometimes	yes	0.0	

Display basic information about the dataset

In [4]:

```
print(df.head())
```

```
   Unnamed: 0  Gender EthnicGroup          ParentEduc     LunchType TestPrep  \
0           0  female         NaN   bachelor's degree      standard     none   
1           1  female     group C        some college      standard      NaN   
2           2  female     group B     master's degree      standard     none   
3           3    male     group A  associate's degree  free/reduced     none   
4           4    male     group C        some college      standard     none   

  ParentMaritalStatus PracticeSport IsFirstChild  NrSiblings TransportMeans  \
0             married     regularly          yes         3.0     school_bus   
1             married     sometimes          yes         0.0            NaN   
2              single     sometimes          yes         4.0     school_bus   
3             married         never           no         1.0            NaN   
4             married     sometimes          yes         0.0     school_bus   

  WklyStudyHours  MathScore  ReadingScore  WritingScore  
0            < 5         71            71            74  
1         5 - 10         69            90            88  
2            < 5         87            93            91  
3         5 - 10         45            56            42  
4         5 - 10         76            78            75  
```

In [5]:

```
df.describe()
```

Out[5]:

	Unnamed: 0	NrSiblings	MathScore	ReadingScore	WritingScore
count	30641.000000	29069.000000	30641.000000	30641.000000	30641.000000
mean	499.556607	2.145894	66.558402	69.377533	68.418622
std	288.747894	1.458242	15.361616	14.758952	15.443525
min	0.000000	0.000000	0.000000	10.000000	4.000000
25%	249.000000	1.000000	56.000000	59.000000	58.000000
50%	500.000000	2.000000	67.000000	70.000000	69.000000
75%	750.000000	3.000000	78.000000	80.000000	79.000000
max	999.000000	7.000000	100.000000	100.000000	100.000000

In [6]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 30641 entries, 0 to 30640
Data columns (total 15 columns):
 #   Column               Non-Null Count  Dtype  
---  ------               --------------  -----  
 0   Unnamed: 0           30641 non-null  int64  
 1   Gender               30641 non-null  object 
 2   EthnicGroup          28801 non-null  object 
 3   ParentEduc           28796 non-null  object 
 4   LunchType            30641 non-null  object 
 5   TestPrep             28811 non-null  object 
 6   ParentMaritalStatus  29451 non-null  object 
 7   PracticeSport        30010 non-null  object 
 8   IsFirstChild         29737 non-null  object 
 9   NrSiblings           29069 non-null  float64
 10  TransportMeans       27507 non-null  object 
 11  WklyStudyHours       29686 non-null  object 
 12  MathScore            30641 non-null  int64  
 13  ReadingScore         30641 non-null  int64  
 14  WritingScore         30641 non-null  int64  
dtypes: float64(1), int64(4), object(10)
memory usage: 3.5+ MB
```

In [7]:
```
df.isnull()
```

Out[7]:

	Unnamed: 0	Gender	EthnicGroup	ParentEduc	LunchType	TestPrep	ParentMaritalStatus	PracticeSport	IsFirstChild	NrSiblings	Tr
0	False	False	True	False	False	False	False	False	False	False	
1	False	False	False	False	False	True	False	False	False	False	
2	False	False	False	False	False	False	False	False	False	False	
3	False	False	False	False	False	False	False	False	False	False	
4	False	False	False	False	False	False	False	False	False	False	
...	...	...	...	...	...	...	...	...	...	...	
30636	False	False	False	False	False	False	False	False	False	False	
30637	False	False	False	False	False	False	False	False	False	False	
30638	False	False	True	False	False	False	False	False	False	False	
30639	False	False	False	False	False	False	False	False	False	False	
30640	False	False	False	False	False	False	False	False	False	False	

30641 rows × 15 columns

In [8]:

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

Out[8]:

```
Unnamed: 0                0
Gender                    0
EthnicGroup            1840
ParentEduc             1845
LunchType                 0
TestPrep               1830
ParentMaritalStatus    1190
PracticeSport           631
IsFirstChild            904
NrSiblings             1572
TransportMeans         3134
WklyStudyHours          955
MathScore                 0
ReadingScore              0
WritingScore              0
dtype: int64
```

Data Cleaning

The initial exploration includes a summary of the dataset, highlighting the count of non-null values, data types, and a brief statistical overview. Missing values are identified and addressed by dropping the "Unnamed: 0" column.

In [9]:

```python
# Drop unnecessary column

df=df.drop("Unnamed: 0", axis=1)
```

In [10]:

```python
df.head()
```

Out[10]:

	Gender	EthnicGroup	ParentEduc	LunchType	TestPrep	ParentMaritalStatus	PracticeSport	IsFirstChild	NrSiblings	TransportMeans
0	female	NaN	bachelor's degree	standard	none	married	regularly	yes	3.0	school_bus
1	female	group C	some college	standard	NaN	married	sometimes	yes	0.0	NaN
2	female	group B	master's degree	standard	none	single	sometimes	yes	4.0	school_bus
3	male	group A	associate's degree	free/reduced	none	married	never	no	1.0	NaN
4	male	group C	some college	standard	none	married	sometimes	yes	0.0	school_bus

Data Transformation

The project involves transforming the data, specifically in the "WklyStudyHours" column. The code snippet below replaces a specific string, enhancing the clarity of the data

In [11]:

```
df["WklyStudyHours"] = df["WklyStudyHours"].str.replace("05-Oct", "5-10")
```

In [12]:

```
df.head()
```

Out[12]:

	Gender	EthnicGroup	ParentEduc	LunchType	TestPrep	ParentMaritalStatus	PracticeSport	IsFirstChild	NrSiblings	TransportMeans
0	female	NaN	bachelor's degree	standard	none	married	regularly	yes	3.0	school_bus
1	female	group C	some college	standard	NaN	married	sometimes	yes	0.0	NaN
2	female	group B	master's degree	standard	none	single	sometimes	yes	4.0	school_bus
3	male	group A	associate's degree	free/reduced	none	married	never	no	1.0	NaN
4	male	group C	some college	standard	none	married	sometimes	yes	0.0	school_bus

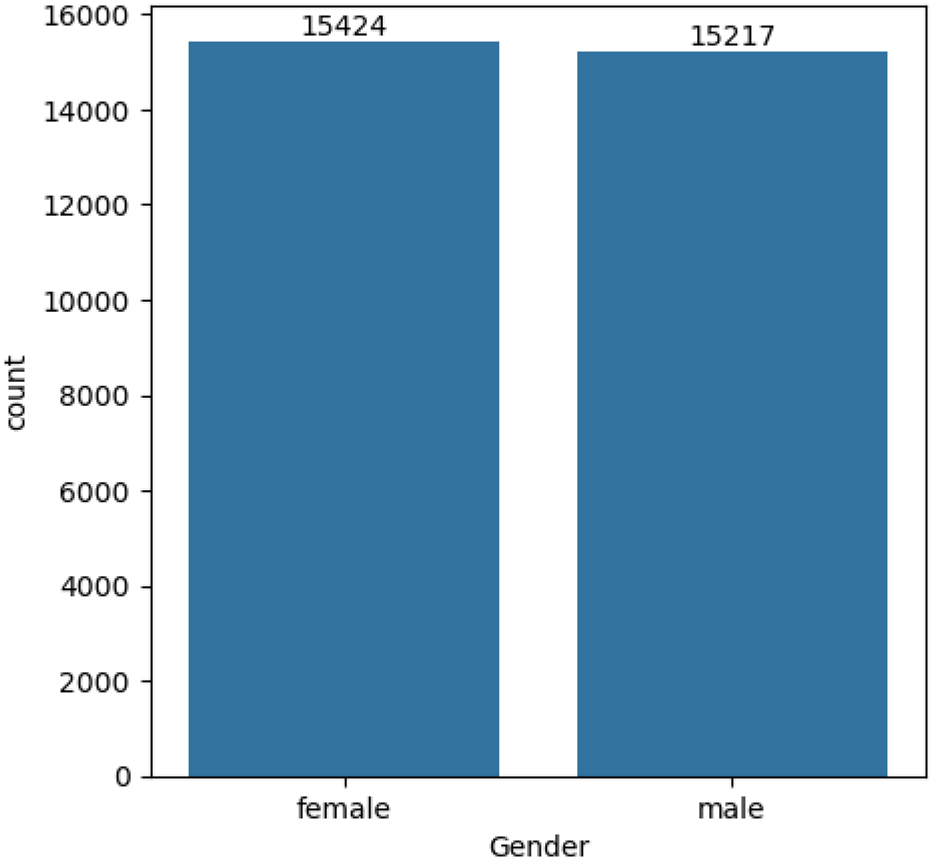
Exploratory Data Analysis

Gender Distribution

Explore the dataset for the gender distribution using a count plot, revealing that the number of females is slightly higher than males.

In [14]:

```
plt.figure(figsize = (5,5))
ax=sns.countplot(data = df, x="Gender")
ax.bar_label(ax.containers[0])
plt.show()
```



Analysis:: Gender Distribution represents, number of females are greater then male.

Parent's Education and Student's score Relationship

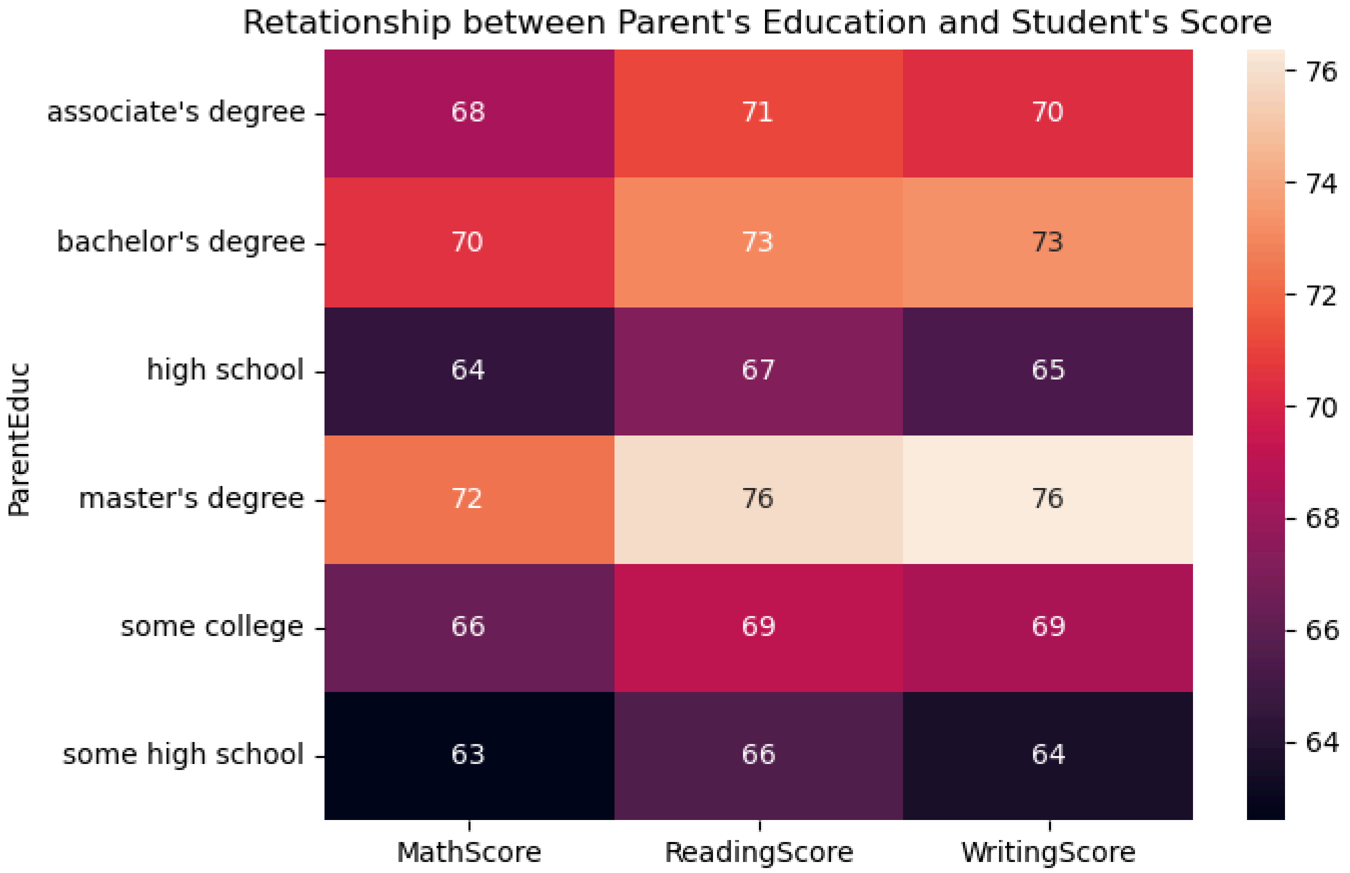
This analysis continues by examining the relationship between parent education levels and student scores.

A grouped bar plot with a heatmap provides a visual representation of this relationship.

Group by Parent's Education and Calculate mean scores

gb=df.groupby("ParentEduc").agg({"MathScore":"mean", "ReadingScore":"mean", "WritingScore":"mean"})

In [19]:
```
## Display the heatmap

plt.figure(figsize=(7,5))
sns.heatmap(gb,annot=True)
plt.title("Retatishsip between Parent's Education and Student's Score")
plt.show()
```



Parent Marital Status and Student's Score Relationship

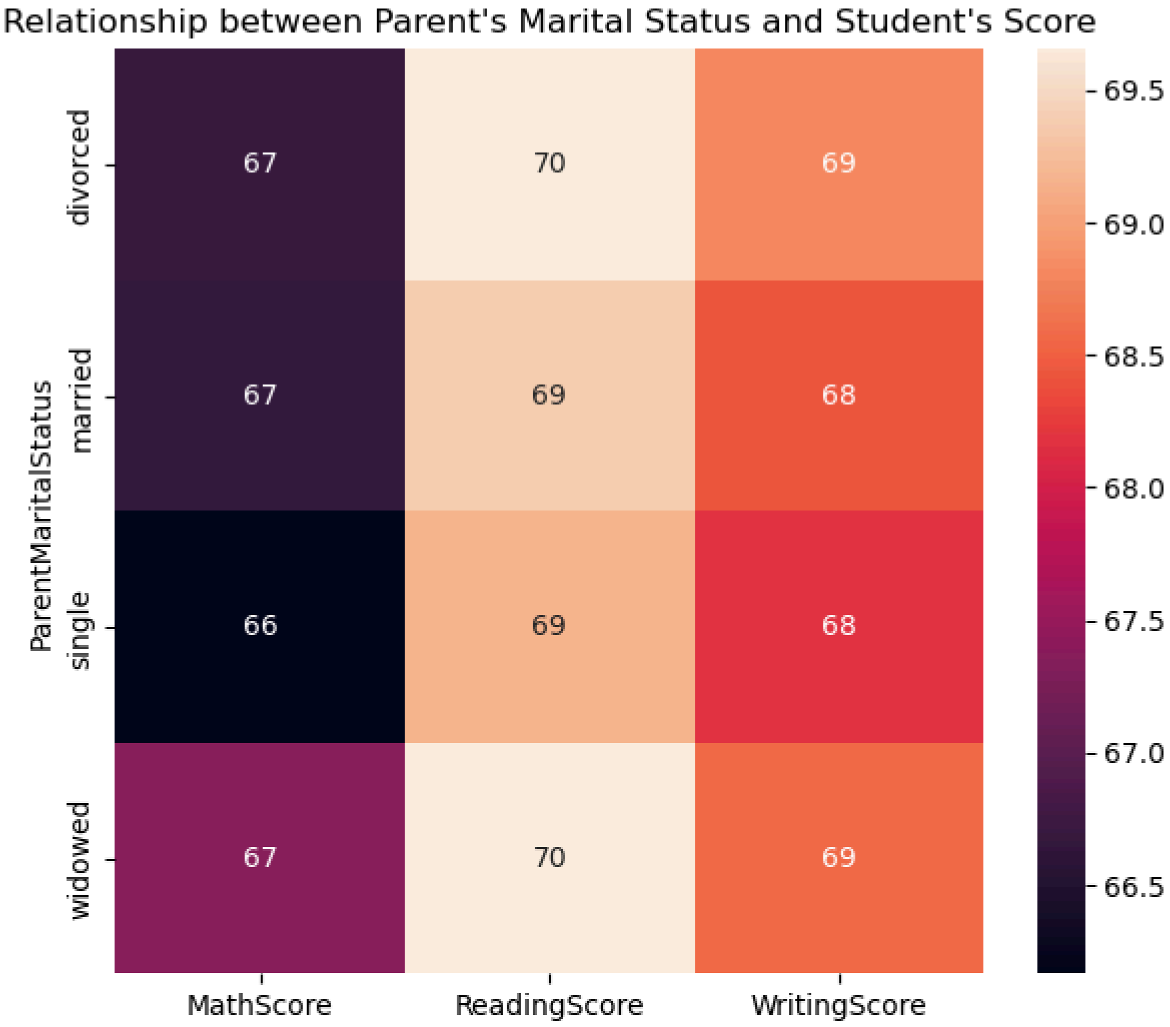
Now investigates the impact of parent marital status on student scores, presenting the finding through a heatmap

In [20]:

```
# Group by Parent's Marital Status and calculate mean scores
gb1 = df.groupby("ParentMaritalStatus").agg({"MathScore": "mean", "ReadingScore": "mean", "WritingScore": "mean"})
```

In [21]:

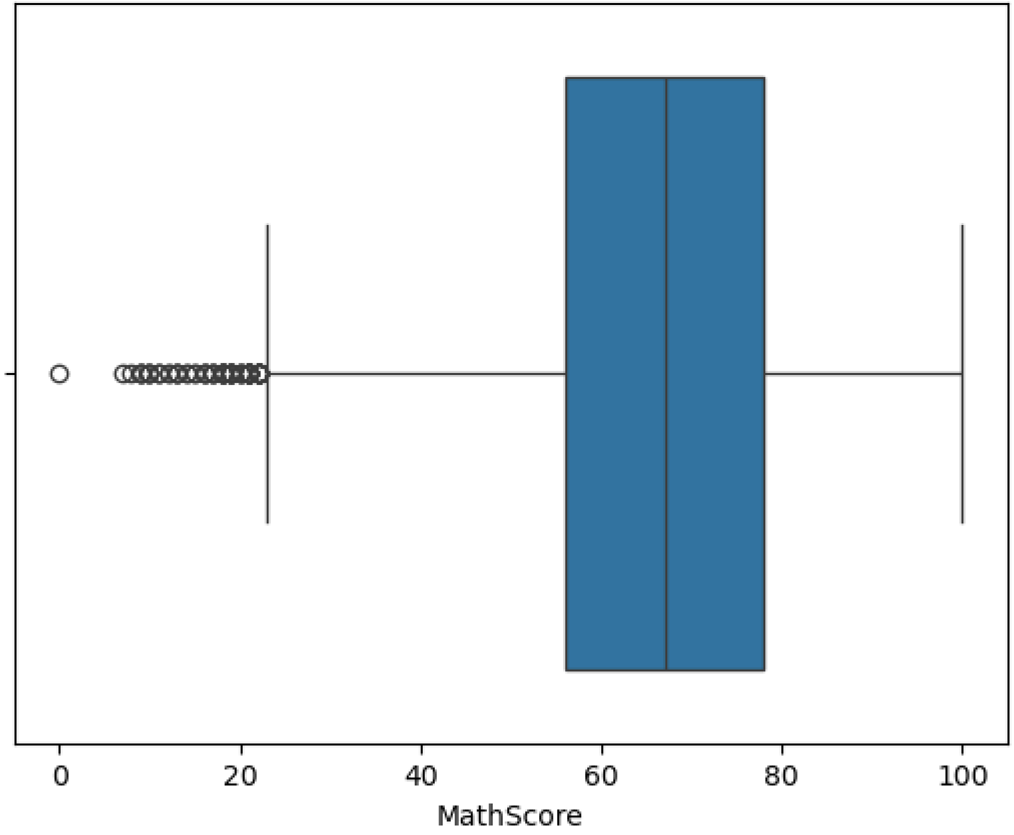
```
# Display the heatmap

plt.figure(figsize=(7,6))
sns.heatmap(gb1, annot=True)
plt.title("Relationship between Parent's Marital Status and Student's Score")
plt.show()
```



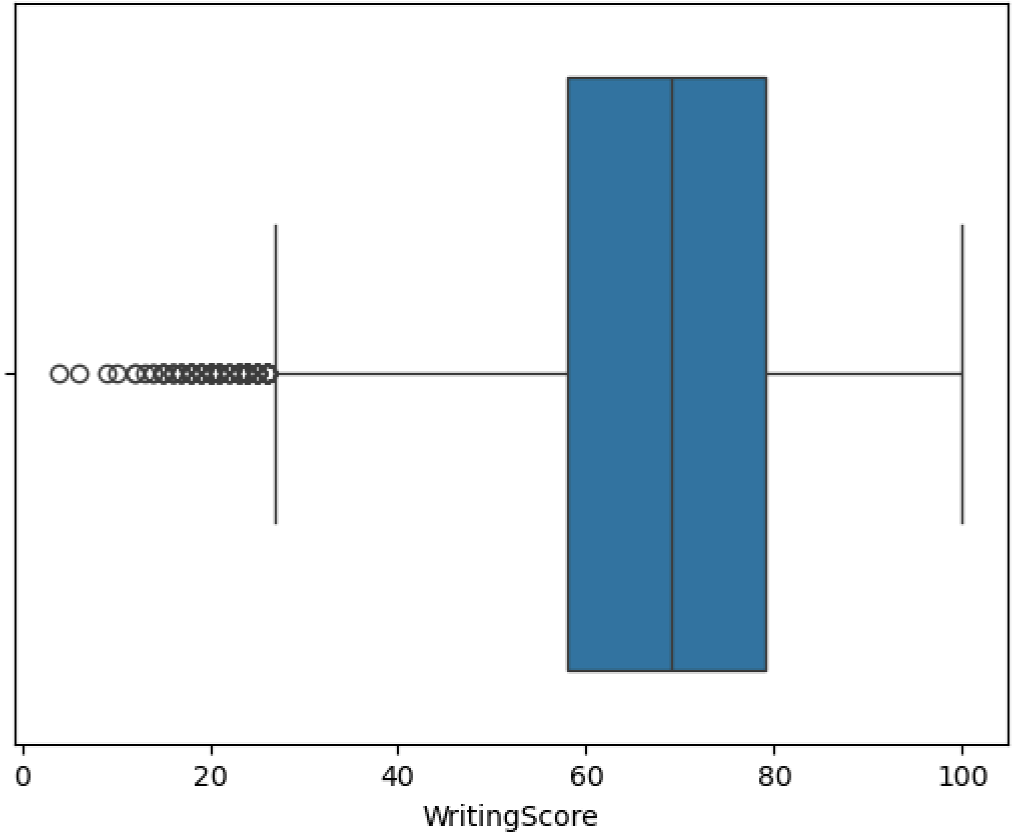
Get the Score Analysis using Box Plot

The distribution of scores is visualized using boxplots, providing insights into the spread and central tendency of Math, Writing, and Reading scores.

In [22]:
```
sns.boxplot(data=df, x="MathScore")
plt.show()
```

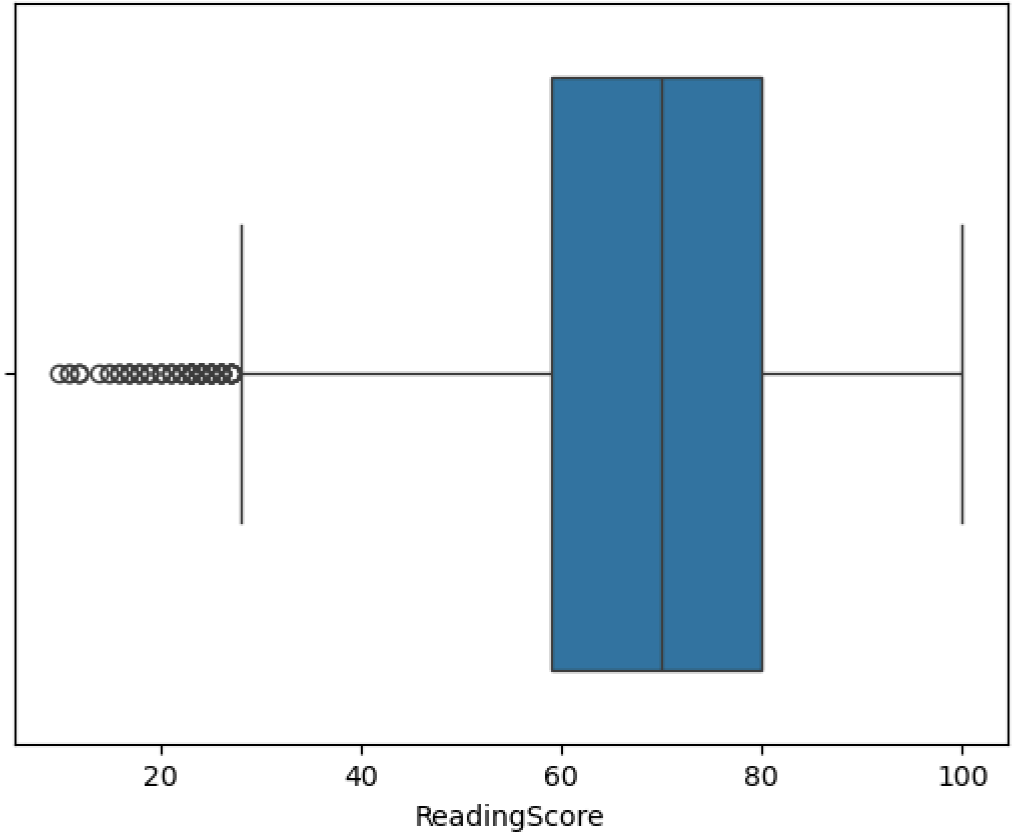


In [23]:
```
sns.boxplot(data=df, x="WritingScore")
plt.show()
```



In [24]:

```
sns.boxplot(data=df, x="ReadingScore")
plt.show()
```



Distribution of Ethnic Groups

The analysis concludes by exploring the distribution of ethnic groups within the dataset.

A pie chart and a count plot are used to present the findings.

In [25]:
```
# Distribution of Ethnic Groups
ethnic_counts = df["EthnicGroup"].value_counts()
```

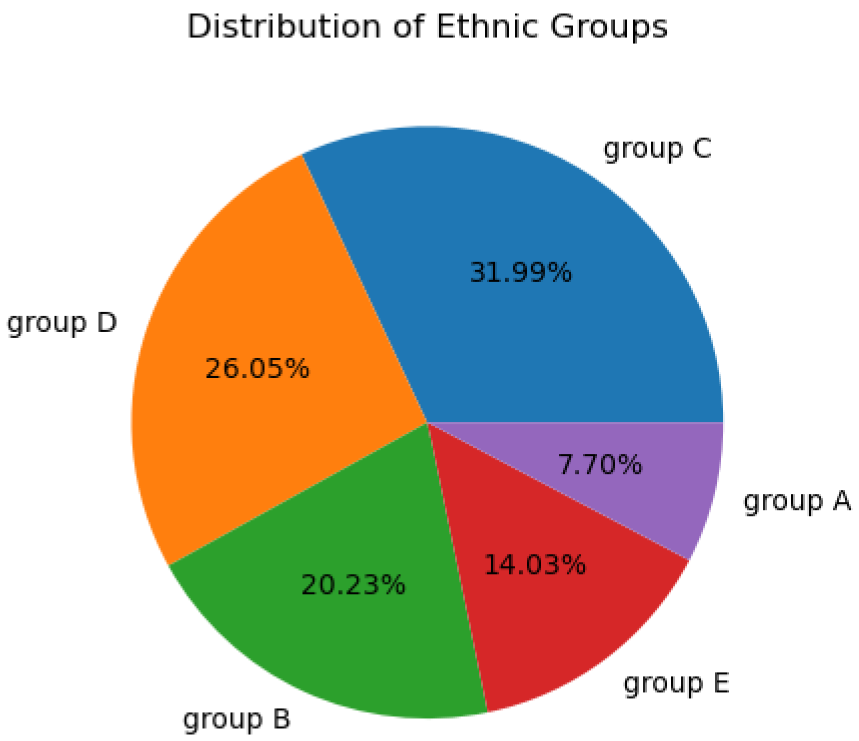
In [26]:
```
ethnic_counts
```

Out[26]:

```
EthnicGroup
group C    9212
group D    7503
group B    5826
group E    4041
group A    2219
Name: count, dtype: int64
```

In [27]:

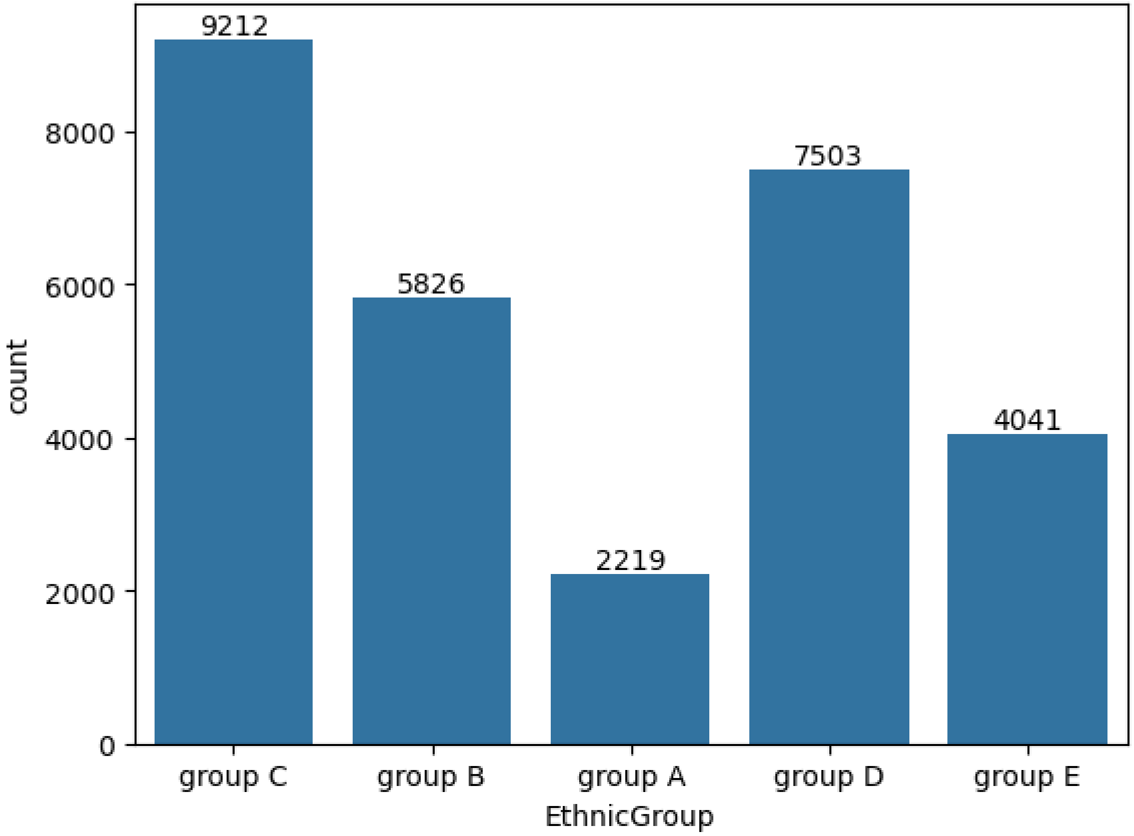
```python
# Pie chart
plt.pie(ethnic_counts, labels=ethnic_counts.index, autopct="%1.2f%%")
plt.title("Distribution of Ethnic Groups")
plt.show()
```



In [28]:

```python
# Count plot
ax = sns.countplot(data=df, x="EthnicGroup")
```

```
ax.bar_label(ax.containers[0])
plt.show()
```



Summary

The "Student Result Analysis" project provides a comprehensive exploration of student data, uncovering insights into various factors influencing academic performance.

The use of Python and powerful libraries facilitates effective data manipulation and visualization for a meaningful analysis.