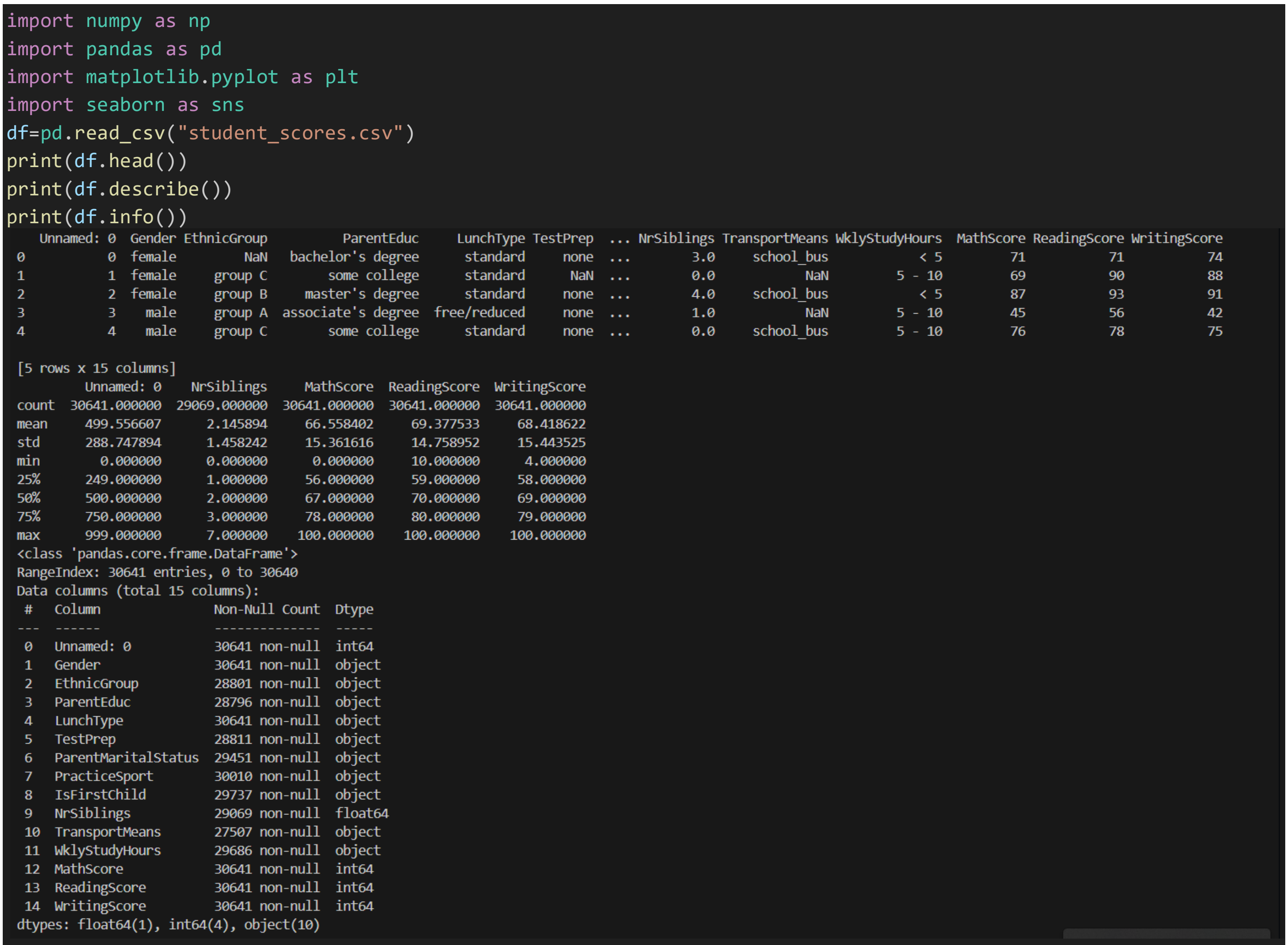
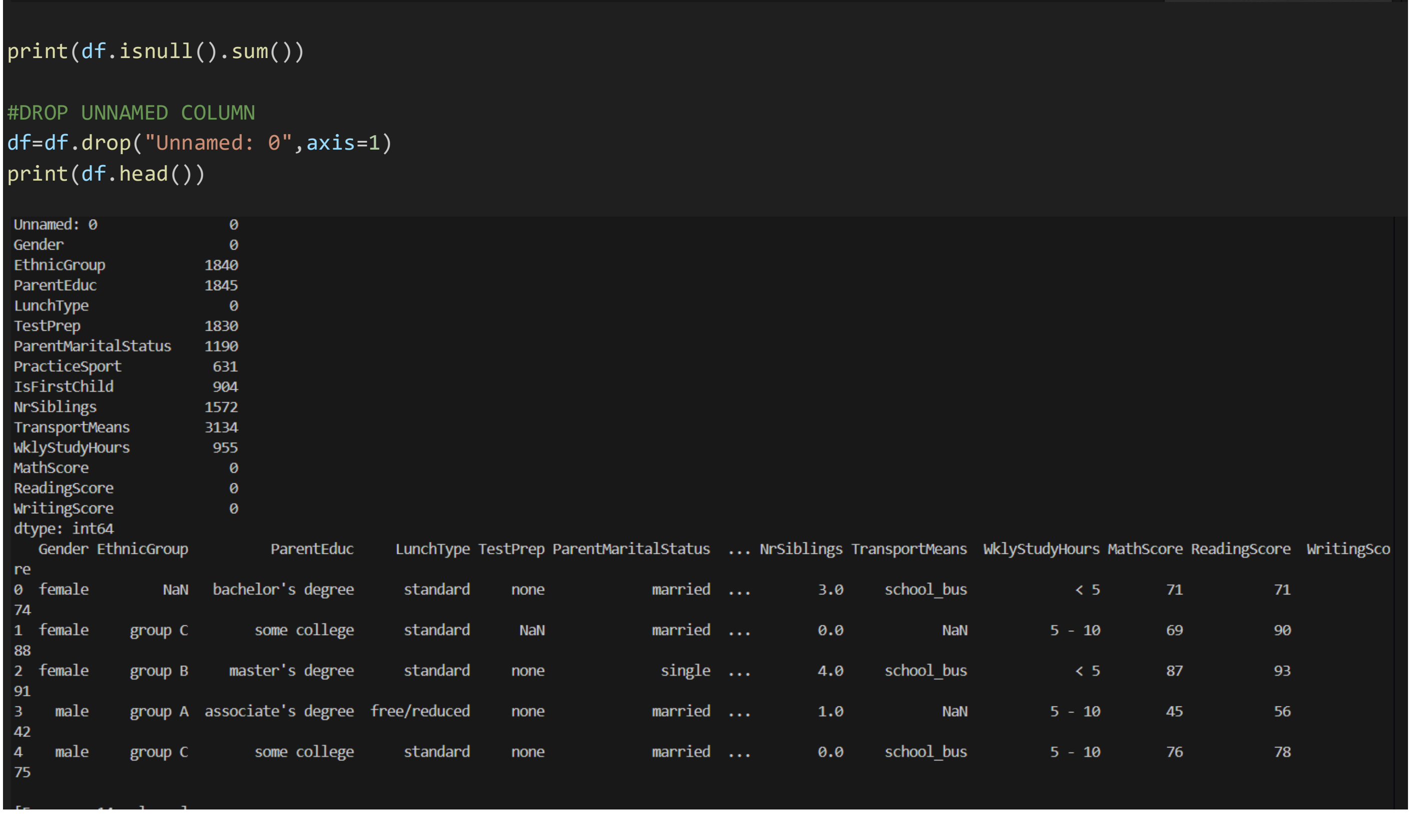
STUDENT RESULT DATA ANALYSIS



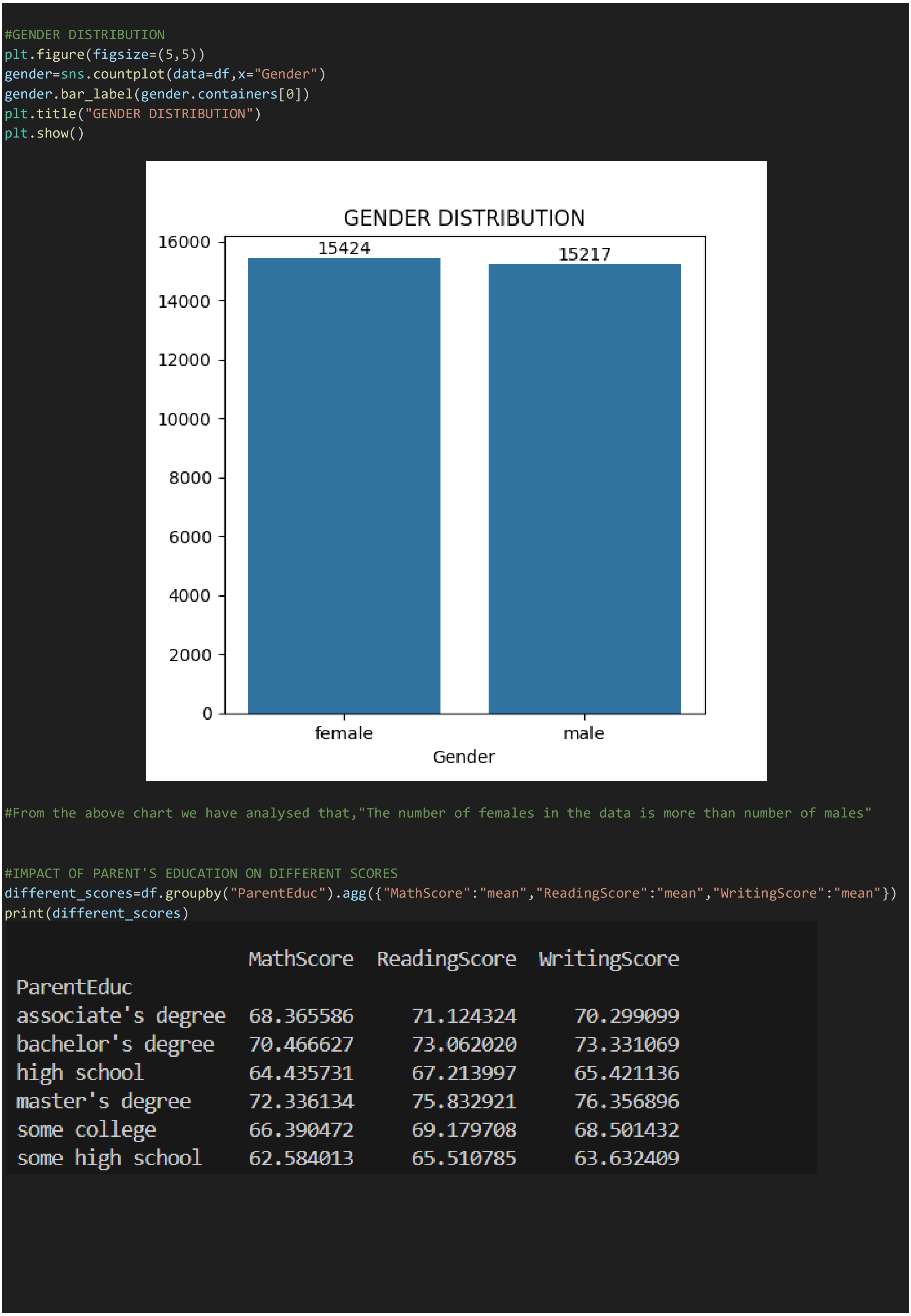
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
df=pd.read_csv("student_scores.csv")
print(df.head())
print(df.describe())
print(df.info())
```

```
   Unnamed: 0  Gender EthnicGroup          ParentEduc     LunchType TestPrep  ... NrSiblings TransportMeans WklyStudyHours  MathScore ReadingScore WritingScore
0           0  female         NaN   bachelor's degree      standard     none  ...        3.0     school_bus            < 5         71           71           74
1           1  female     group C        some college      standard      NaN  ...        0.0            NaN         5 - 10         69           90           88
2           2  female     group B     master's degree      standard     none  ...        4.0     school_bus            < 5         87           93           91
3           3    male     group A  associate's degree  free/reduced     none  ...        1.0            NaN         5 - 10         45           56           42
4           4    male     group C        some college      standard     none  ...        0.0     school_bus         5 - 10         76           78           75

[5 rows x 15 columns]
         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
<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)
```



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

#DROP UNNAMED COLUMN
df=df.drop("Unnamed: 0",axis=1)
print(df.head())
```

```
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
   Gender EthnicGroup          ParentEduc     LunchType TestPrep ParentMaritalStatus  ... NrSiblings TransportMeans  WklyStudyHours MathScore ReadingScore  WritingSco
re
0  female         NaN   bachelor's degree      standard     none             married  ...        3.0     school_bus             < 5        71           71
74
1  female     group C        some college      standard      NaN             married  ...        0.0            NaN          5 - 10        69           90
88
2  female     group B     master's degree      standard     none              single  ...        4.0     school_bus             < 5        87           93
91
3    male     group A  associate's degree  free/reduced     none             married  ...        1.0            NaN          5 - 10        45           56
42
4    male     group C        some college      standard     none             married  ...        0.0     school_bus          5 - 10        76           78
75
```

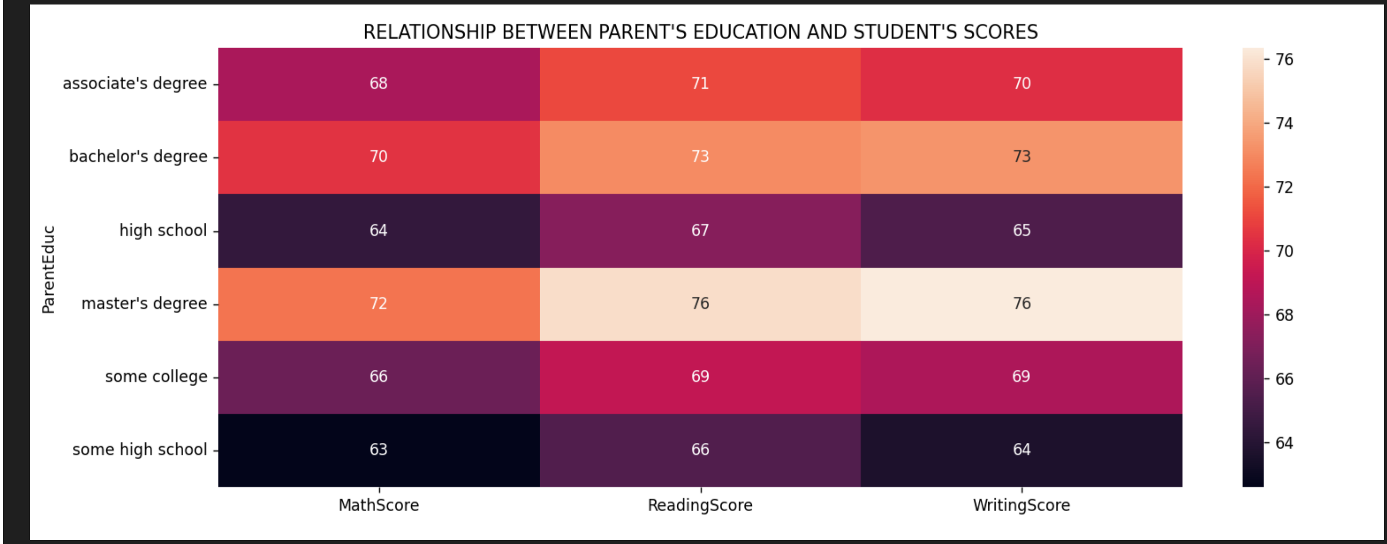


```
#GENDER DISTRIBUTION
plt.figure(figsize=(5,5))
gender=sns.countplot(data=df,x="Gender")
gender.bar_label(gender.containers[0])
plt.title("GENDER DISTRIBUTION")
plt.show()
```

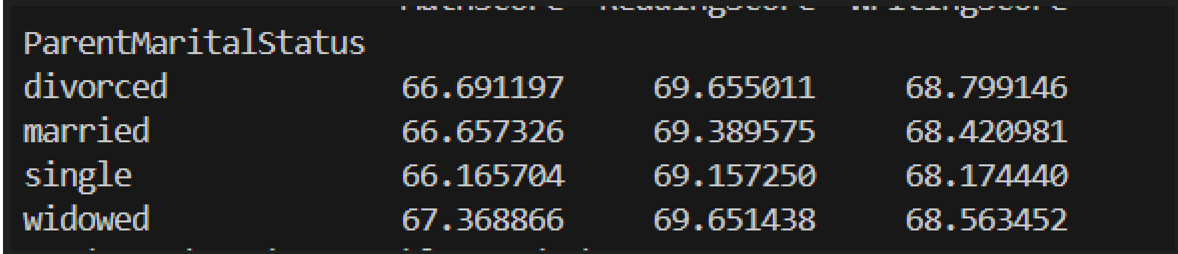
```
#From the above chart we have analysed that,"The number of females in the data is more than number of males"


#IMPACT OF PARENT'S EDUCATION ON DIFFERENT SCORES
different_scores=df.groupby("ParentEduc").agg({"MathScore":"mean","ReadingScore":"mean","WritingScore":"mean"})
print(different_scores)
```

```
                    MathScore  ReadingScore  WritingScore
ParentEduc
associate's degree  68.365586     71.124324     70.299099
bachelor's degree   70.466627     73.062020     73.331069
high school         64.435731     67.213997     65.421136
master's degree     72.336134     75.832921     76.356896
some college        66.390472     69.179708     68.501432
some high school    62.584013     65.510785     63.632409
```

```
plt.figure(figsize=(5,5))
sns.heatmap(different_scores,annot=True)
plt.title("RELATIONSHIP BETWEEN PARENT'S EDUCATION AND STUDENT'S SCORES")
plt.show()
```

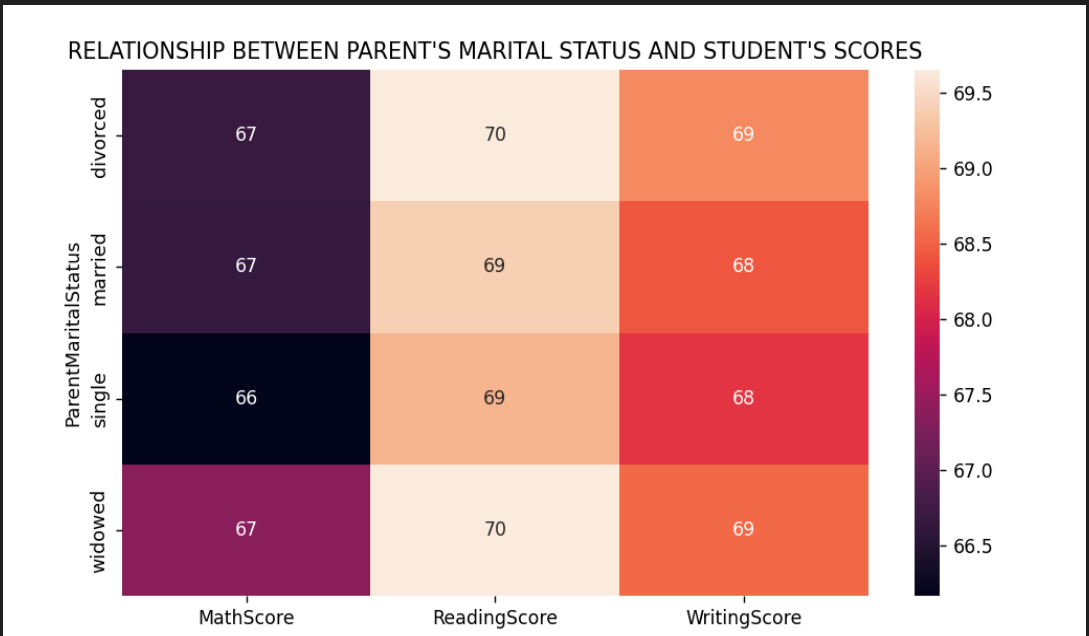


```
#From the above chart we have concluded that the education of the parents have a good impact on the student's scores

#IMPACT OF PARENT'S MARITAL STATUS ON STUDENT'S SCORES
marital_status=df.groupby("ParentMaritalStatus").agg({"MathScore":"mean","ReadingScore":"mean","WritingScore":"mean"})
print(marital_status)
```

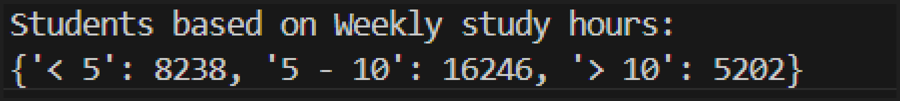


```
                      MathScore  ReadingScore  WritingScore
ParentMaritalStatus
divorced              66.691197     69.655011     68.799146
married               66.657326     69.389575     68.420981
single                66.165704     69.157250     68.174440
widowed               67.368866     69.651438     68.563452
```

```
plt.figure(figsize=(5,5))
sns.heatmap(marital_status,annot=True)
plt.title("RELATIONSHIP BETWEEN PARENT'S MARITAL STATUS AND STUDENT'S SCORES")
plt.show()
```

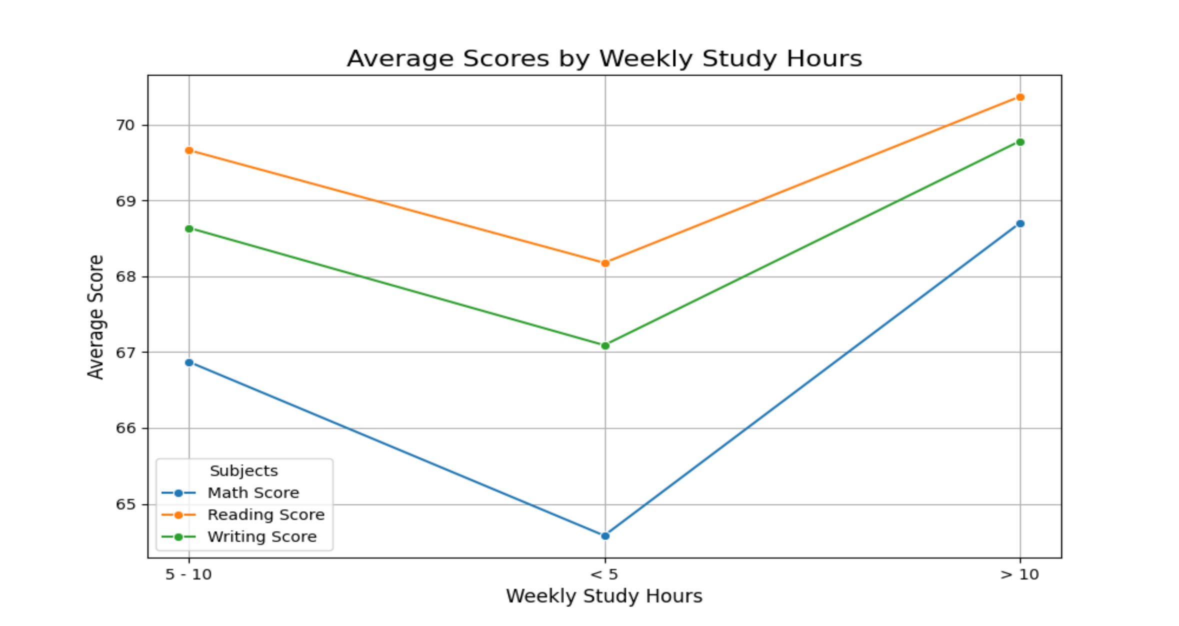


```
#From the above chart we have concluded that there is no/negligible impact on the student's scores due to their
parent's marital status

#COUNT OF STUDENTS BASED ON WEEKLY STUDY HOURS

study_hours_categories={"< 5":0,"5 - 10":0,"> 10":0}
for hours in df["WklyStudyHours"]:
    if hours=="< 5":
        study_hours_categories["< 5"]+=1
    elif hours=="5 - 10":
        study_hours_categories["5 - 10"]+=1
    elif hours=="> 10":
        study_hours_categories["> 10"]+=1
print("Students based on Weekly study hours:")
print(study_hours_categories)
```



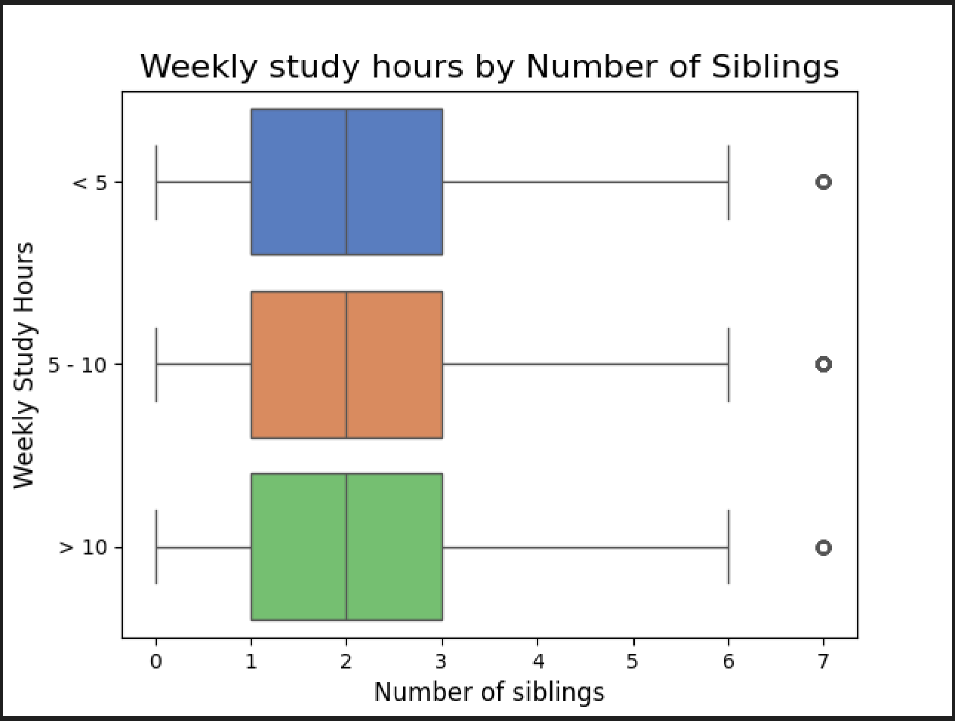
```
Students based on Weekly study hours:
{'< 5': 8238, '5 - 10': 16246, '> 10': 5202}
```

```
#Understand how students distribute their study hours and access if more hours result in better performance

# Group the dataset by study hours and calculate average scores

study_hours_groups = df.groupby('WklyStudyHours')[['MathScore', 'ReadingScore',
'WritingScore']].mean().reset_index()

# Plot using Seaborn
plt.figure(figsize=(10, 6))
sns.lineplot(data=study_hours_groups, x='WklyStudyHours', y='MathScore', label='Math Score',marker='o')
sns.lineplot(data=study_hours_groups, x='WklyStudyHours', y='ReadingScore', label='Reading Score', marker='o')
sns.lineplot(data=study_hours_groups, x='WklyStudyHours', y='WritingScore', label='Writing Score', marker='o')

# Adding titles and labels
plt.title('Average Scores by Weekly Study Hours', fontsize=16)
plt.xlabel('Weekly Study Hours', fontsize=12)
plt.ylabel('Average Score', fontsize=12)
plt.legend(title='Subjects')
plt.grid(True)

# Show plot
plt.show()
```

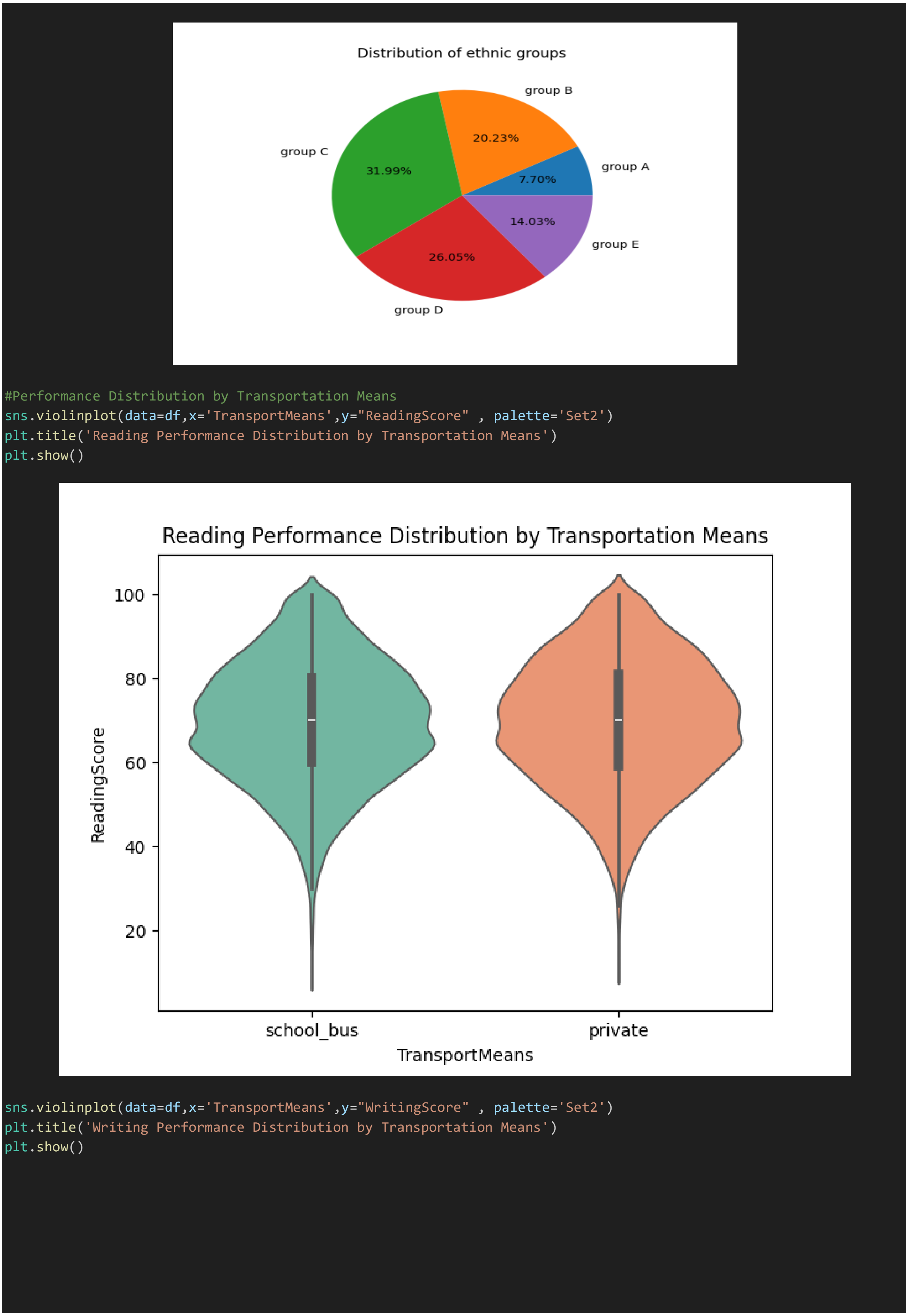


```
#Clear trends showing how scores increase as study hours increase
```

```
#IMPACT OF SIBLINGS ON STUDY HOURS
sns.boxplot(data=df,x="NrSiblings",y="WklyStudyHours", hue="WklyStudyHours",legend=False,palette="muted")
plt.title("Weekly study hours by Number of Siblings",fontsize=16)
plt.xlabel('Number of siblings', fontsize=12)
plt.ylabel('Weekly Study Hours', fontsize=12)
plt.show()
```

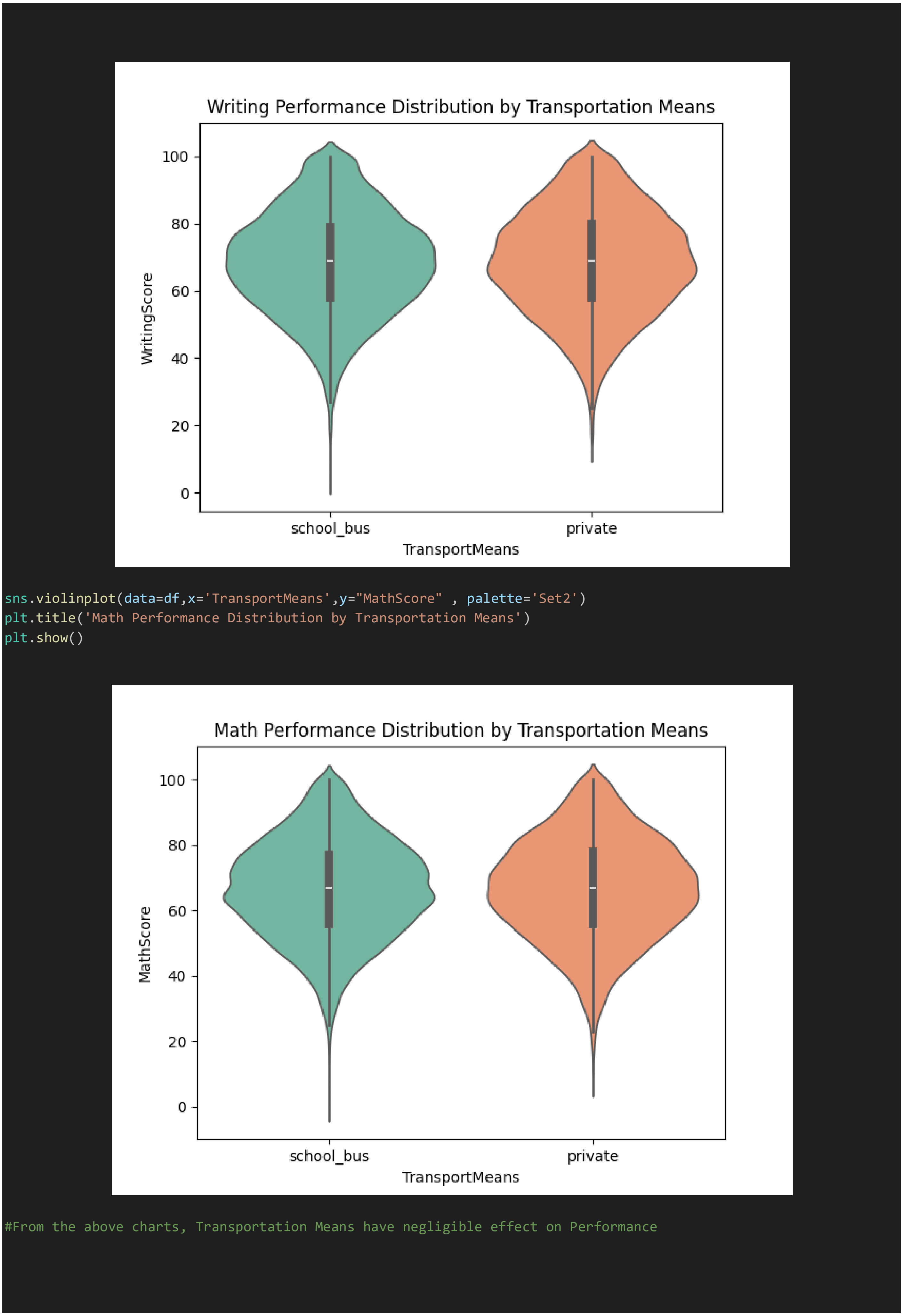


```
#DISTRIBUTION OF ETHNIC GROUPS
print(df["EthnicGroup"].unique())
groupA=df.loc[(df["EthnicGroup"]=="group A")].count()
groupB=df.loc[(df["EthnicGroup"]=="group B")].count()
groupC=df.loc[(df["EthnicGroup"]=="group C")].count()
groupD=df.loc[(df["EthnicGroup"]=="group D")].count()
groupE=df.loc[(df["EthnicGroup"]=="group E")].count()
label_name=["group A","group B","group C","group D","group E"]
ethnic_list=[groupA["EthnicGroup"],groupB["EthnicGroup"],groupC["EthnicGroup"],groupD["EthnicGroup"],groupE["Et
hnicGroup"]]
plt.pie(ethnic_list,labels=label_name,autopct="%1.2f%%")
plt.title("Distribution of ethnic groups")
plt.show()
```



```
#Performance Distribution by Transportation Means
sns.violinplot(data=df,x='TransportMeans',y="ReadingScore" , palette='Set2')
plt.title('Reading Performance Distribution by Transportation Means')
plt.show()
```

```
sns.violinplot(data=df,x='TransportMeans',y="WritingScore" , palette='Set2')
plt.title('Writing Performance Distribution by Transportation Means')
plt.show()
```

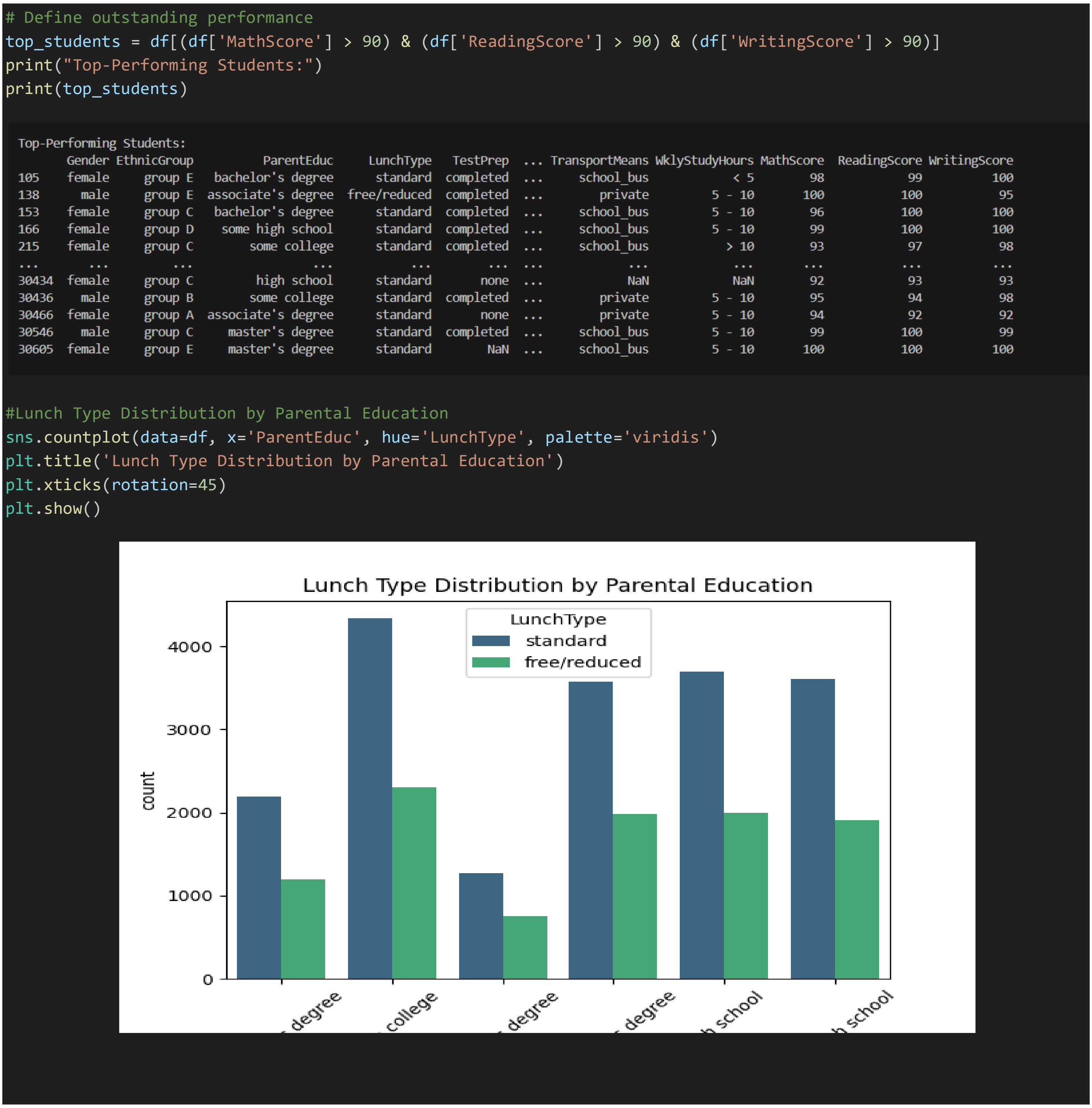


Writing Performance Distribution by Transportation Means
WritingScore
100
80
60
40
20
0
school_bus
private
TransportMeans

```
sns.violinplot(data=df,x='TransportMeans',y="MathScore" , palette='Set2')
plt.title('Math Performance Distribution by Transportation Means')
plt.show()
```

Math Performance Distribution by Transportation Means
MathScore
100
80
60
40
20
0
school_bus
private
TransportMeans

```
#From the above charts, Transportation Means have negligible effect on Performance
```



```python
# Define outstanding performance
top_students = df[(df['MathScore'] > 90) & (df['ReadingScore'] > 90) & (df['WritingScore'] > 90)]
print("Top-Performing Students:")
print(top_students)
```

```
Top-Performing Students:
      Gender EthnicGroup          ParentEduc     LunchType   TestPrep  ... TransportMeans WklyStudyHours MathScore  ReadingScore WritingScore
105   female     group E   bachelor's degree      standard  completed  ...     school_bus            < 5        98            99          100
138     male     group E  associate's degree  free/reduced  completed  ...        private         5 - 10       100           100           95
153   female     group C   bachelor's degree      standard  completed  ...     school_bus         5 - 10        96           100          100
166   female     group D    some high school      standard  completed  ...     school_bus         5 - 10        99           100          100
215   female     group C        some college      standard  completed  ...     school_bus           > 10        93            97           98
...      ...         ...                 ...           ...        ...  ...            ...            ...       ...           ...          ...
30434 female     group C         high school      standard       none  ...            NaN            NaN        92            93           93
30436   male     group B        some college      standard  completed  ...        private         5 - 10        95            94           98
30466 female     group A  associate's degree      standard       none  ...        private         5 - 10        94            92           92
30546   male     group C     master's degree      standard  completed  ...     school_bus         5 - 10        99           100           99
30605 female     group E     master's degree      standard        NaN  ...     school_bus         5 - 10       100           100          100
```

```python
#Lunch Type Distribution by Parental Education
sns.countplot(data=df, x='ParentEduc', hue='LunchType', palette='viridis')
plt.title('Lunch Type Distribution by Parental Education')
plt.xticks(rotation=45)
plt.show()
```