Student Performance Dataset Exploratory Data Analysis

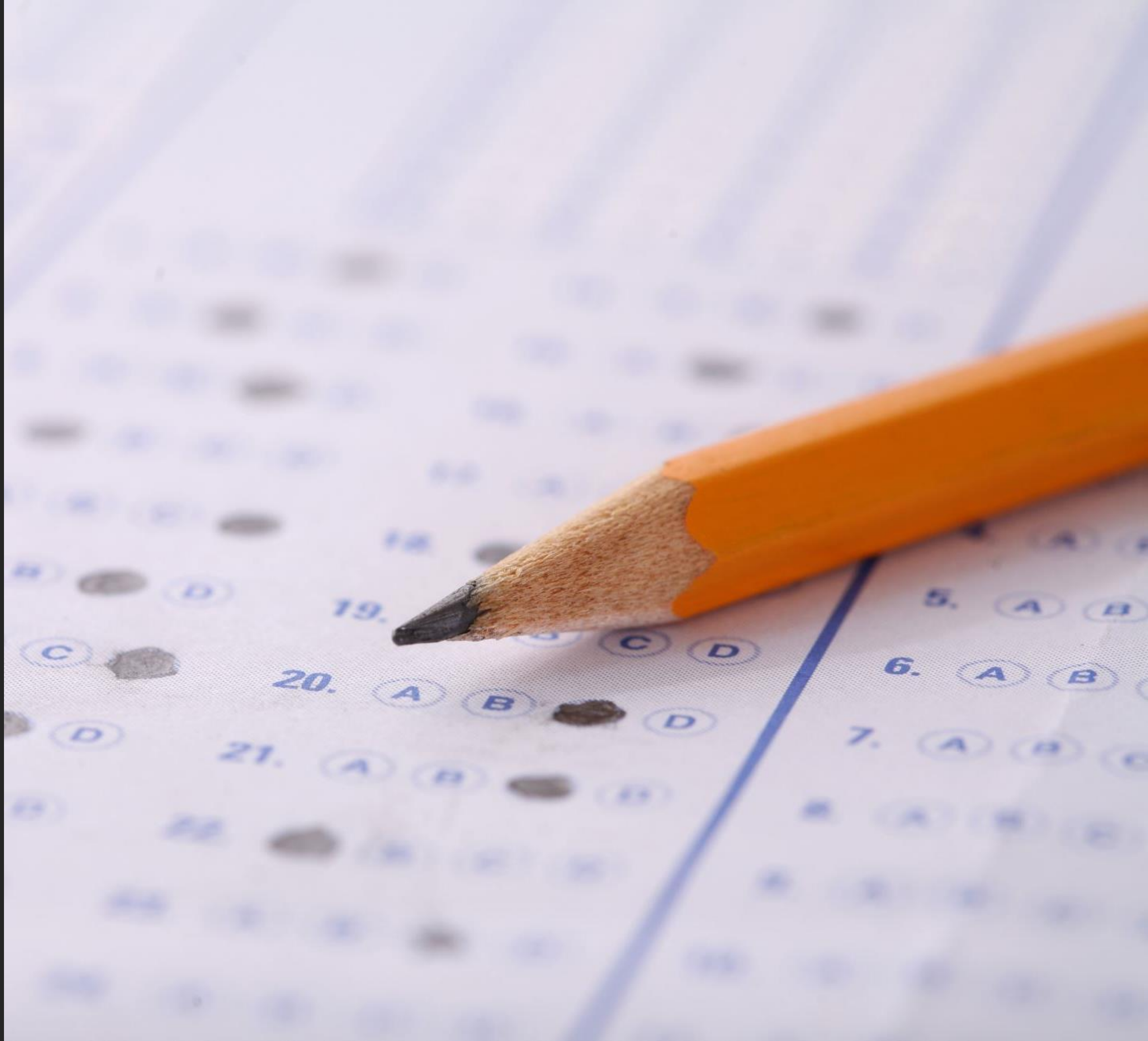
Neha Roy (Data Analyst)

https://github.com/Neha2001roy

www.linkedin.com/in/neha-roy-6469311b2

About the dataset

This dataset includes scores from three test scores of students at a (fictional) public school and a variety of personal and socio-economic factors that may have interaction effects upon them.



Data Features:

Gender: Gender of the student (male/female)

EthnicGroup: Ethnic group of the student (group A to E)

ParentEduc: Parent(s) education background (from some_highschool to master's degree)

LunchType: School lunch type (standard or free/reduced)

TestPrep: Test preparation course followed (completed or none)

ParentMaritalStatus: Parent(s) marital status (married/single/widowed/divorced)

PracticeSport: How often the student parctice sport (never/sometimes/regularly))

IsFirstChild: If the child is first child in the family or not (yes/no)

NrSiblings: Number of siblings the student has (0 to 7)

TransportMeans: Means of transport to school (schoolbus/private)

WklyStudyHours: Weekly self-study hours(less that 5hrs; between 5 and 10hrs; more than 10hrs)

MathScore: math test score(0-100)

ReadingScore: reading test score(0-100)

WritingScore: writing test score(0-100)

Objective

The objective of analyzing the Student Performance dataset is multifaceted, aimed at understanding, exploring, and utilizing the data to gain insights into factors affecting student academic achievement.

Essential libraries used:

1. Pandas : It is a powerful library for data manipulation and analysis. It provides data structure like DataFrame and Series, which are essential for working with structured data.
2. Numpy : it is a fundamental package for numerical computing in python.
3. Matplotlib.pyplot: it is a comprehensive library for creating static, animated and interactive visualizations in python.
4. Seaborn: it is a python visualization library based on matplotlib.
5. Import warnings: This line imports the warnings module which provides functions to issue warnings.
6. Warnings.filterwarnings("ignore"): This line sets up a filter to ignore all warnings.

LOADING THE DATASET

```
df=pd.read_csv("student performance.csv")
```

VISUALIZING FIRST 5 ROWS

```
df.head()
```

IMPORTING NECESSARY LIBRARIES

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings("ignore")
```

STEP 1: Exploratory Data Analysis - Data Collection

	Unnamed: 0	Gender	EthnicGroup	ParentEduc	LunchType	TestPrep	ParentMaritalStatus	PracticeSport	IsFirstChild	NrSiblings	TransportMeans	WklyStudyHours	MathScore	ReadingScore	WritingScore
0	0	female	NaN	bachelor's degree	standard	none	married	regularly	yes	3.0	school_bus	< 5	71	71	74
1	1	female	group C	some college	standard	NaN	married	sometimes	yes	0.0	NaN	5 - 10	69	90	88
2	2	female	group B	master's degree	standard	none	single	sometimes	yes	4.0	school_bus	< 5	87	93	91
3	3	male	group A	associate's degree	free/reduced	none	married	never	no	1.0	NaN	5 - 10	45	56	42
4	4	male	group C	some college	standard	none	married	sometimes	yes	0.0	school_bus	5 - 10	76	78	75

There are total 30,641 rows and 15 columns in which we have 1 float, 4 integer, and 10 object type column containing categorical data.

```
<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)
```

DROPPING THE UNNAMED UNNECESSARY COLUMN

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

```
df.head()
```

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

STEP 2: Data Cleaning

SINCE THERE WAS AN UNNECESSARY COLUMN NAMED "UNNAMED: 0", WE HAD TO DROP IT.

VISUALIZING THE NULL VALUES

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

```
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
```

CHECKING IF THERE IS ANY DUPLICATED VALUES

```
df.duplicated().sum()
```

```
0
```

FILLING THE NULL VALUES WITH THEIR MODE VALUES

```
for column in df.columns:
    mode_value = df[column].mode()[0]
    df[column].fillna(mode_value, inplace=True)
```

VISUALISING THE CLASS IN GENDER COLUMN

```
df["Gender"].value_counts()
```

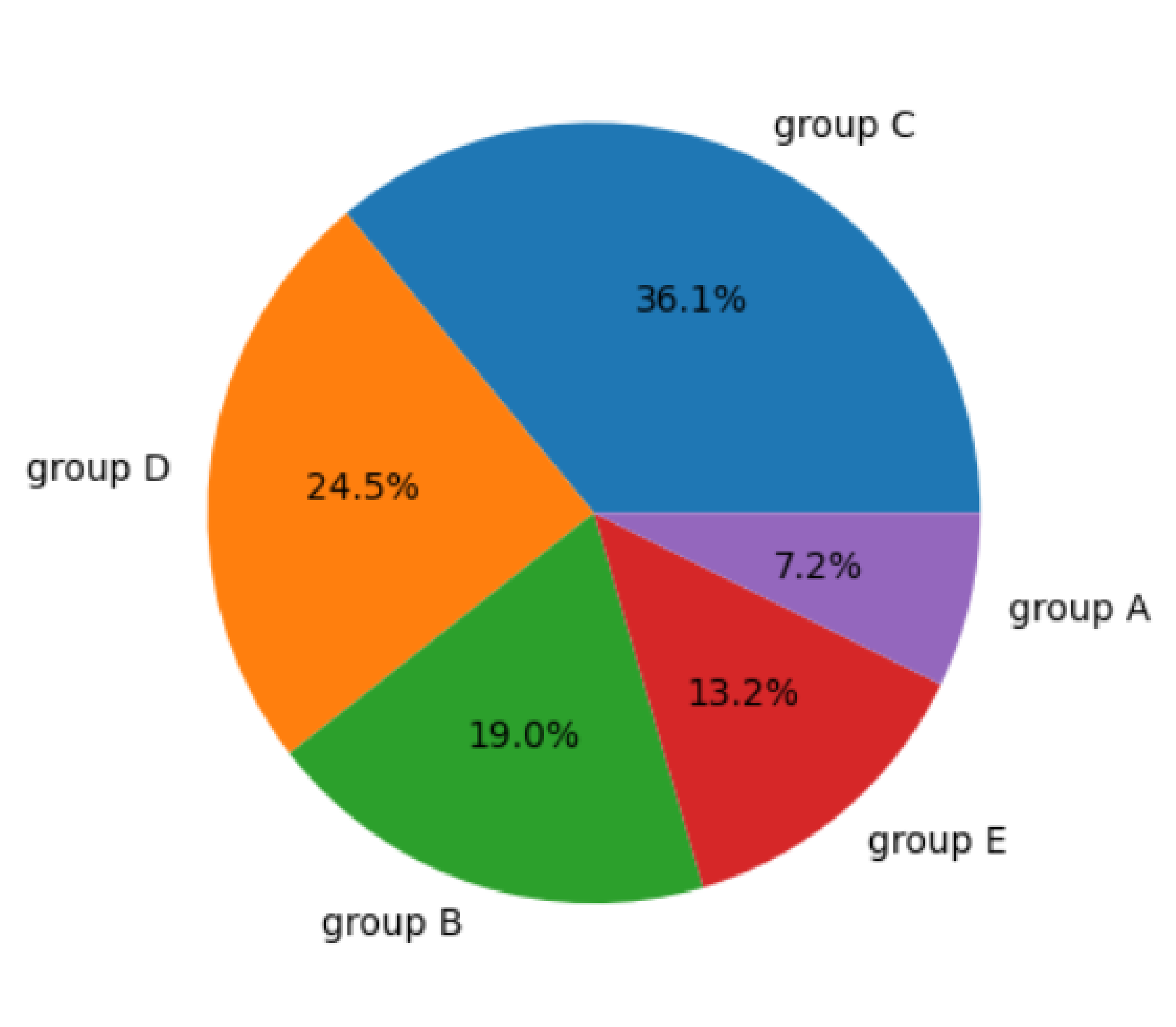
```
Gender
female     15424
male       15217
Name: count, dtype: int64
```

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

STEP 3: Data Exploration & Visualization

VISUALISING THE DISTRIBUTION OF ETHNIC GROUP

```
plt.pie(df["EthnicGroup"].value_counts(), labels=df["EthnicGroup"].value_counts().index, autopct='%1.1f%%')
plt.show()
```



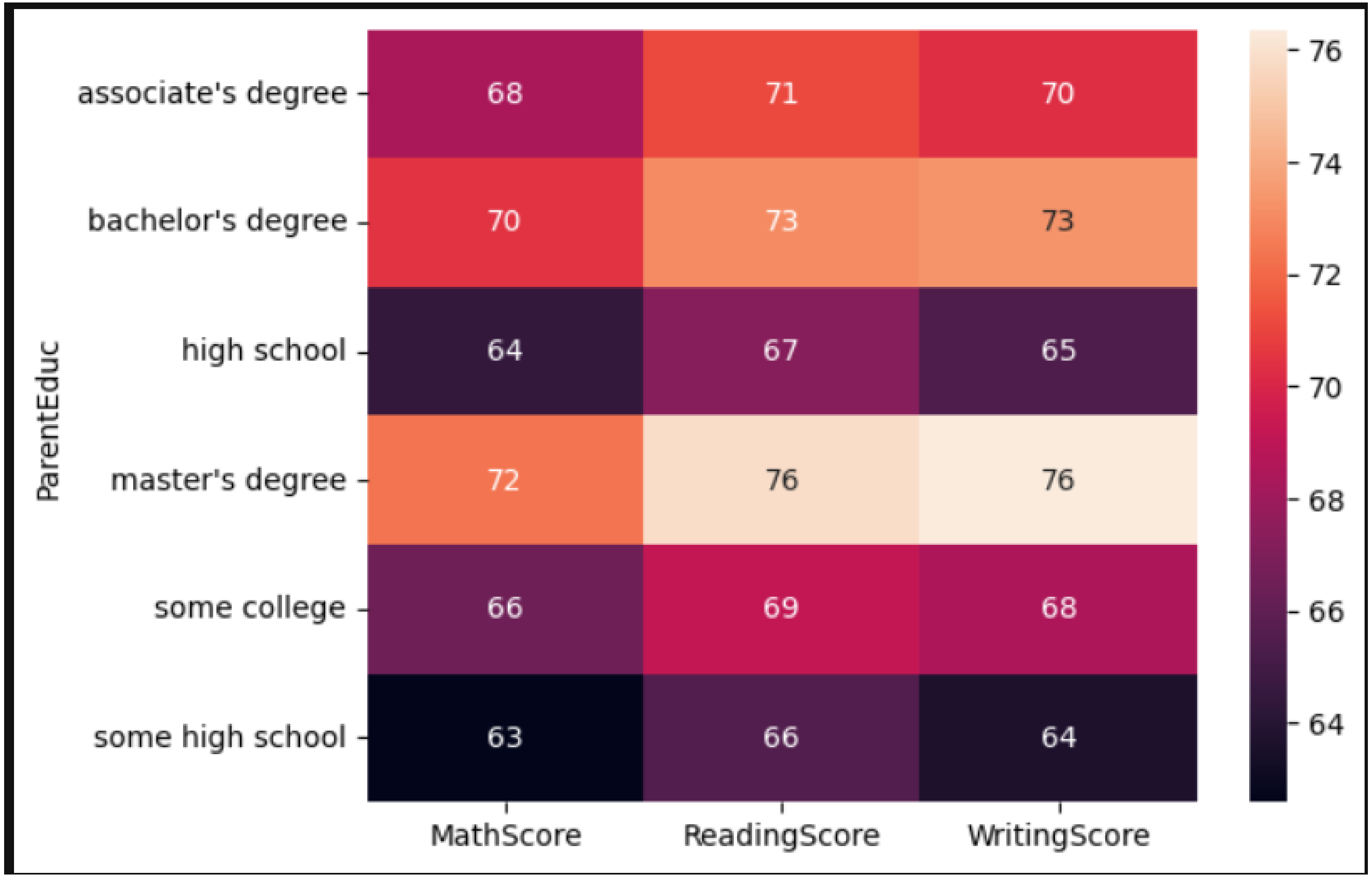
CHECKING THE RELATION OF MEAN OF MATH SCORE, READING SCORE, WRITING SCORE WITH THE DIFFERENT CLASSES OF PARENTS EDUCATION DEGREE.

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

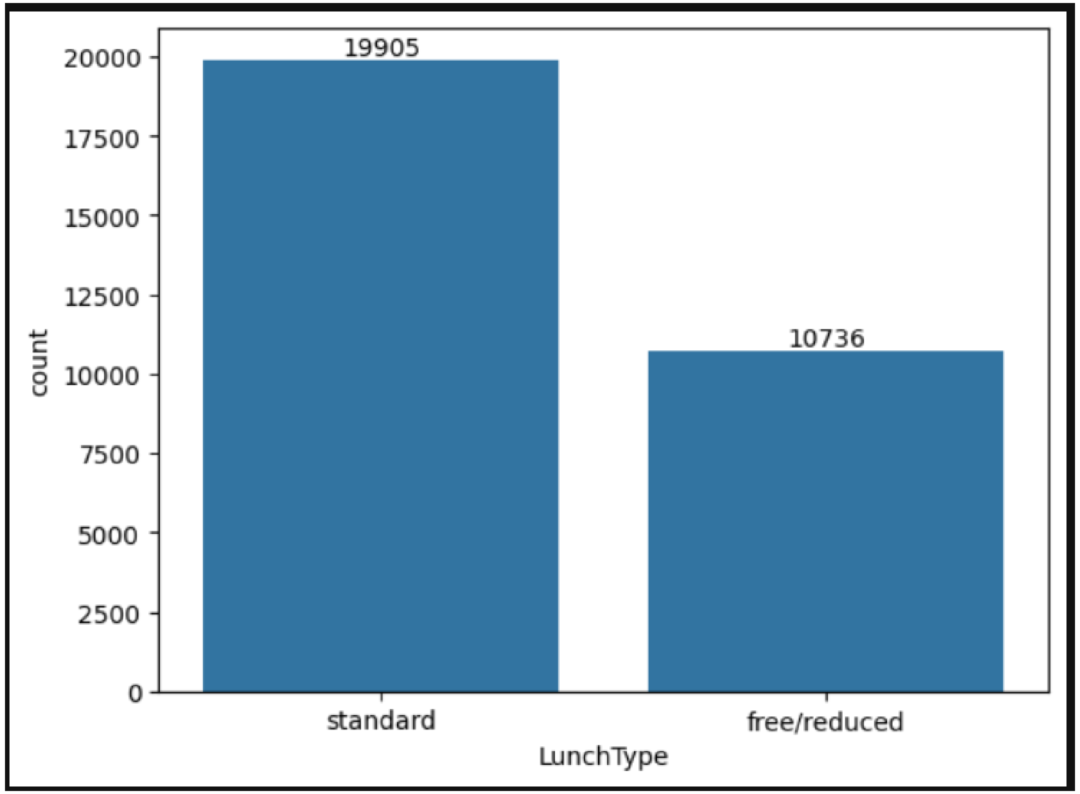
```
parents_edu
```

ParentEduc	MathScore	ReadingScore	WritingScore
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	66.445978	69.189667	68.456711
some high school	62.584013	65.510785	63.632409

```
sns.heatmap(parents_edu, annot=True)
plt.show()
```



associate's degree
bachelor's degree
high school
master's degree
some college
some high school
ParentEduc
68
71
70
70
73
73
64
67
65
72
76
76
66
69
68
63
66
64
MathScore
ReadingScore
WritingScore
76
74
72
70
68
66
64

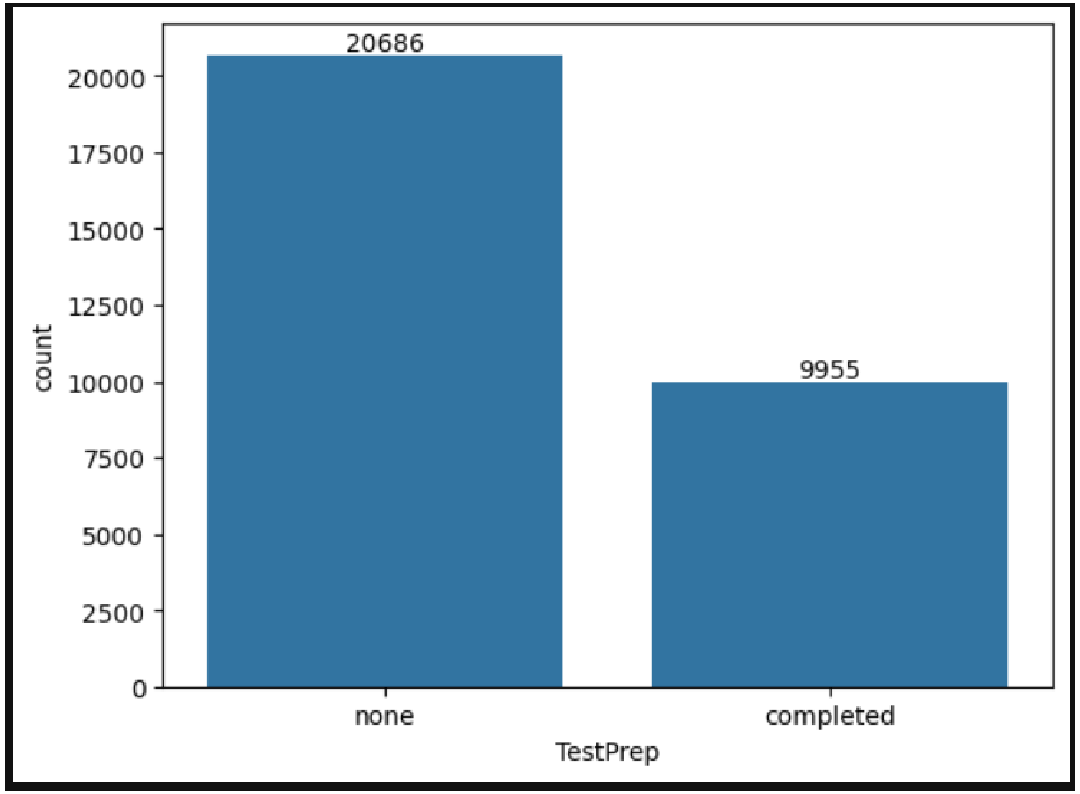


VISUALISING THE CLASS NUMBERS OF LUNCHTYPE

```
df["LunchType"].value_counts()
```

```
LunchType
standard        19905
free/reduced    10736
Name: count, dtype: int64
```

```
lunch_type=sns.countplot(data=df, x="LunchType")
lunch_type.bar_label(lunch_type.containers[0])
plt.show()
```



VISUALISING THE CLASS NUMBERS OF TEST PREPARATION

```
df["TestPrep"].value_counts()
```

```
TestPrep
none         20686
completed     9955
Name: count, dtype: int64
```

```
testprep=sns.countplot(data=df, x="TestPrep")
testprep.bar_label(testprep.containers[0])
plt.show()
```

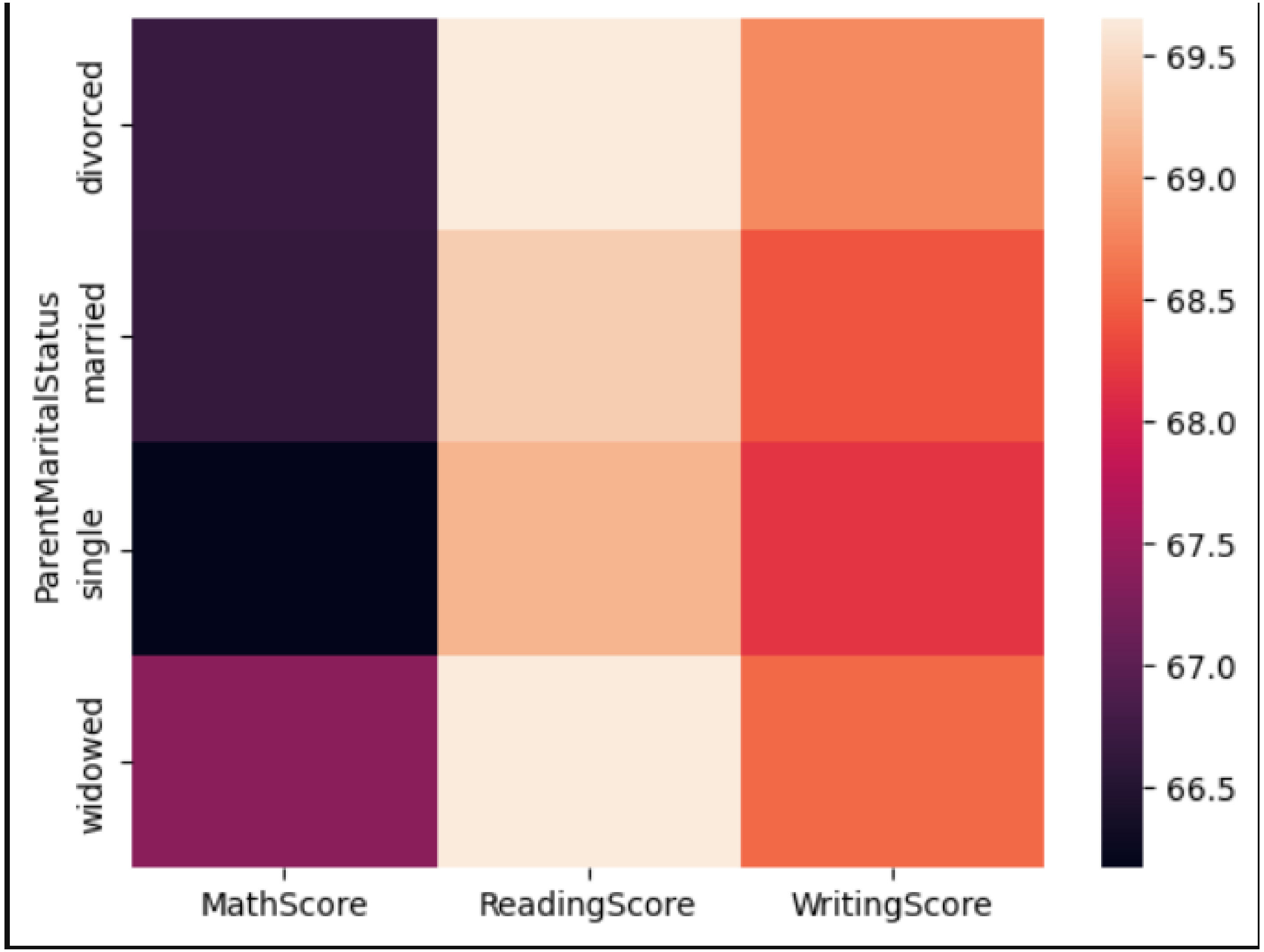
CHECKING THE RELATION OF MEAN OF MATH SCORE, READING SCORE, WRITING SCORE WITH THE DIFFERENT CLASSES OF PARENTS MARITAL STATUS.

```
parents_marital_status=df.groupby("ParentMaritalStatus").agg({"MathScore": 'mean', "ReadingScore": 'mean', "WritingScore": 'mean'})
```

```
parents_marital_status
```

ParentMaritalStatus	MathScore	ReadingScore	WritingScore
divorced	66.691197	69.655011	68.799146
married	66.650161	69.379561	68.406177
single	66.165704	69.157250	68.174440
widowed	67.368866	69.651438	68.563452

```
sns.heatmap(parents_marital_status)
plt.show()
```



divorced
married
single
widowed
ParentMaritalStatus
MathScore
ReadingScore
WritingScore
69.5
69.0
68.5
68.0
67.5
67.0
66.5

CHECKING THE RELATION OF MEAN OF MATH SCORE, READING SCORE, WRITING SCORE WITH THE STUDENTS WHO PRACTICE SPORTS.

```
sports=df.groupby("PracticeSport").agg({"MathScore": 'mean', "ReadingScore": 'mean', "WritingScore": 'mean'})
```

```
sports
```

PracticeSport	MathScore	ReadingScore	WritingScore
never	64.171079	68.337662	66.522727
regularly	67.839155	69.943019	69.604003
sometimes	66.289258	69.255112	68.090255

CHECKING THE RELATION OF MEAN OF MATH SCORE, READING SCORE, WRITING SCORE WITH THE FIRST CHILD STATUS.

```
first_child=df.groupby("IsFirstChild").agg({"MathScore": 'mean', "ReadingScore": 'mean', "WritingScore": 'mean'})
```

```
first_child
```

IsFirstChild	MathScore	ReadingScore	WritingScore
no	66.246832	69.132614	68.210887
yes	66.724507	69.508106	68.529371

CHANGING THE DATATYPE OF NO. OF SIBLINGS TO INTEGER TYPE INSTEAD OF FLOAT.

```
df["NrSiblings"]=df["NrSiblings"].astype("int")
```

CHECKING THE STATISTICAL DATA OF NUMERICAL COLUMN

```
df.describe()
```

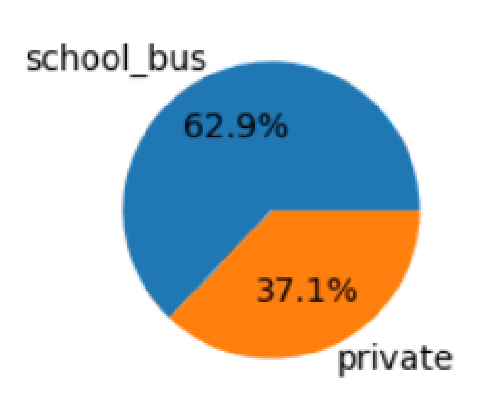
	NrSiblings	MathScore	ReadingScore	WritingScore
count	30641.000000	30641.000000	30641.000000	30641.000000
mean	2.087106	66.558402	69.377533	68.418622
std	1.442665	15.361616	14.758952	15.443525
min	0.000000	0.000000	10.000000	4.000000
25%	1.000000	56.000000	59.000000	58.000000
50%	2.000000	67.000000	70.000000	69.000000
75%	3.000000	78.000000	80.000000	79.000000
max	7.000000	100.000000	100.000000	100.000000

VISUALISING THE CLASS NUMBERS OF TRANSPORTATION MEANS

```
df["TransportMeans"].value_counts()
```

```
TransportMeans
school_bus    19279
private       11362
Name: count, dtype: int64
```

```
plt.figure(figsize=(4,2))
plt.pie(df["TransportMeans"].value_counts(), labels=df["TransportMeans"].value_counts().index, autopct="%1.1f%%")
plt.show()
```

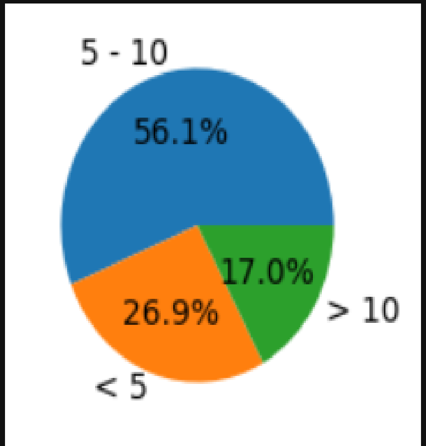


VISUALISING THE DISTRIBUTION OF WEEKLY STUDY HOURS OF A STUDENT

```
df["WklyStudyHours"].value_counts()
```

```
WklyStudyHours
5 - 10    17201
< 5        8238
> 10       5202
Name: count, dtype: int64
```

```
plt.figure(figsize=(4,2))
plt.pie(df["WklyStudyHours"].value_counts(), labels=df["WklyStudyHours"].value_counts().index, autopct="%1.1f%%")
plt.show()
```



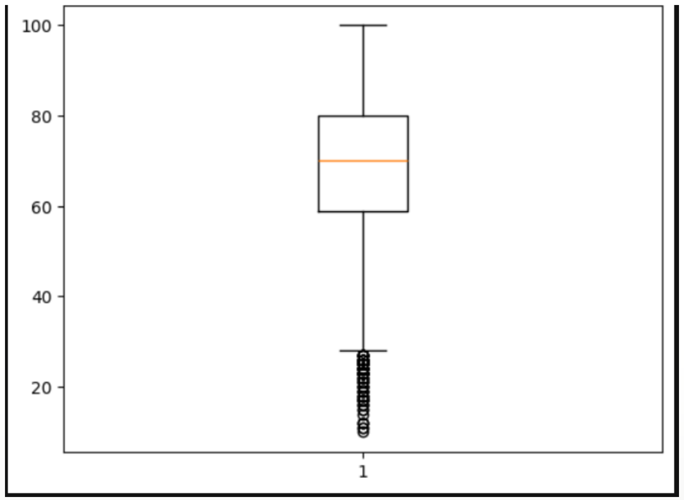
VISUALIZING THE OUTIERS OF MATHS SCORE, READING SCORE, AND WRITING SCORE

```
plt.boxplot(df["MathScore"])
plt.show()
```

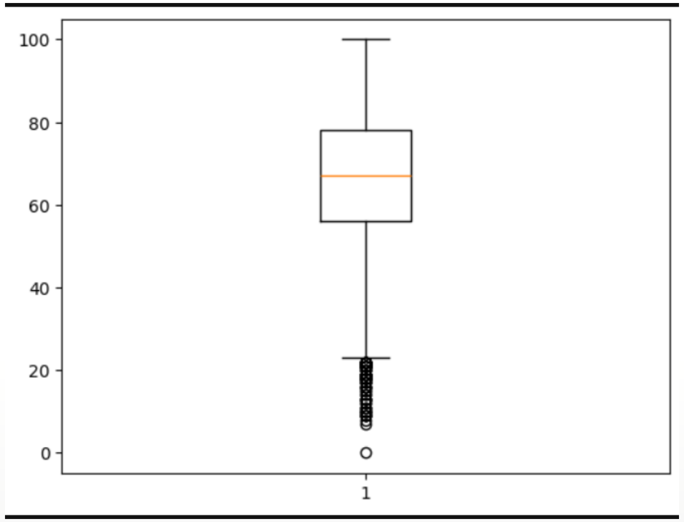
```
plt.boxplot(df["ReadingScore"])
plt.show()
```

```
plt.boxplot(df["WritingScore"])
plt.show()
```

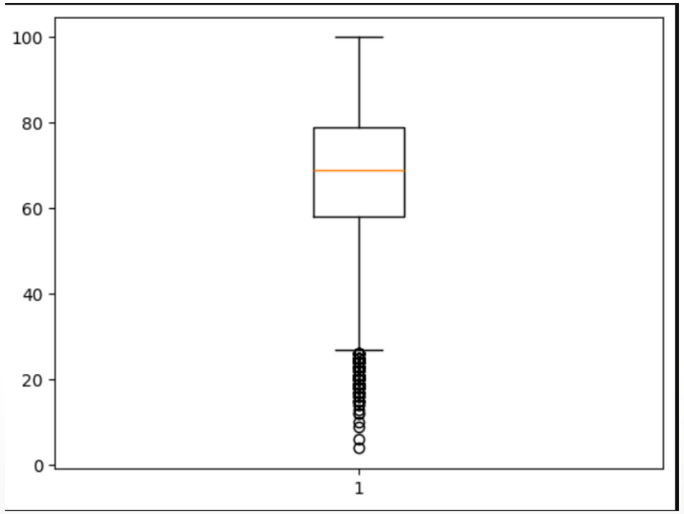
MATH SCORE



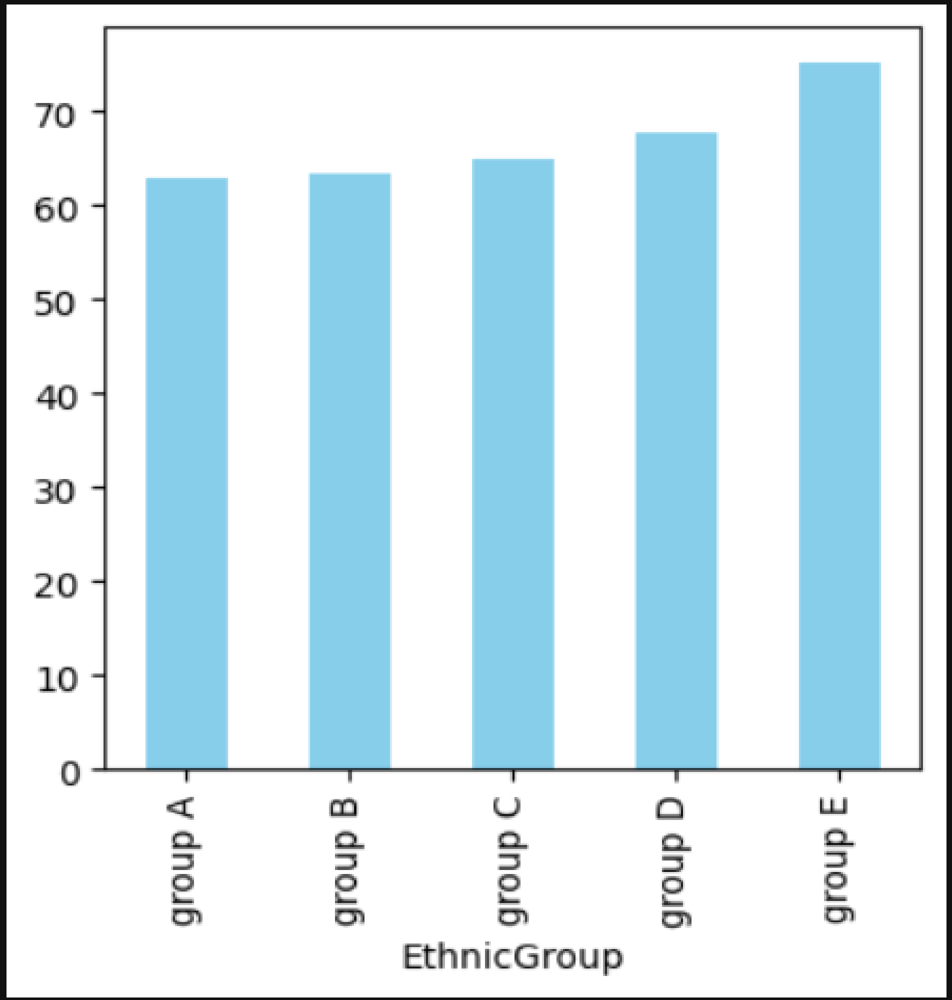
READING SCORE



WRITING SCORE

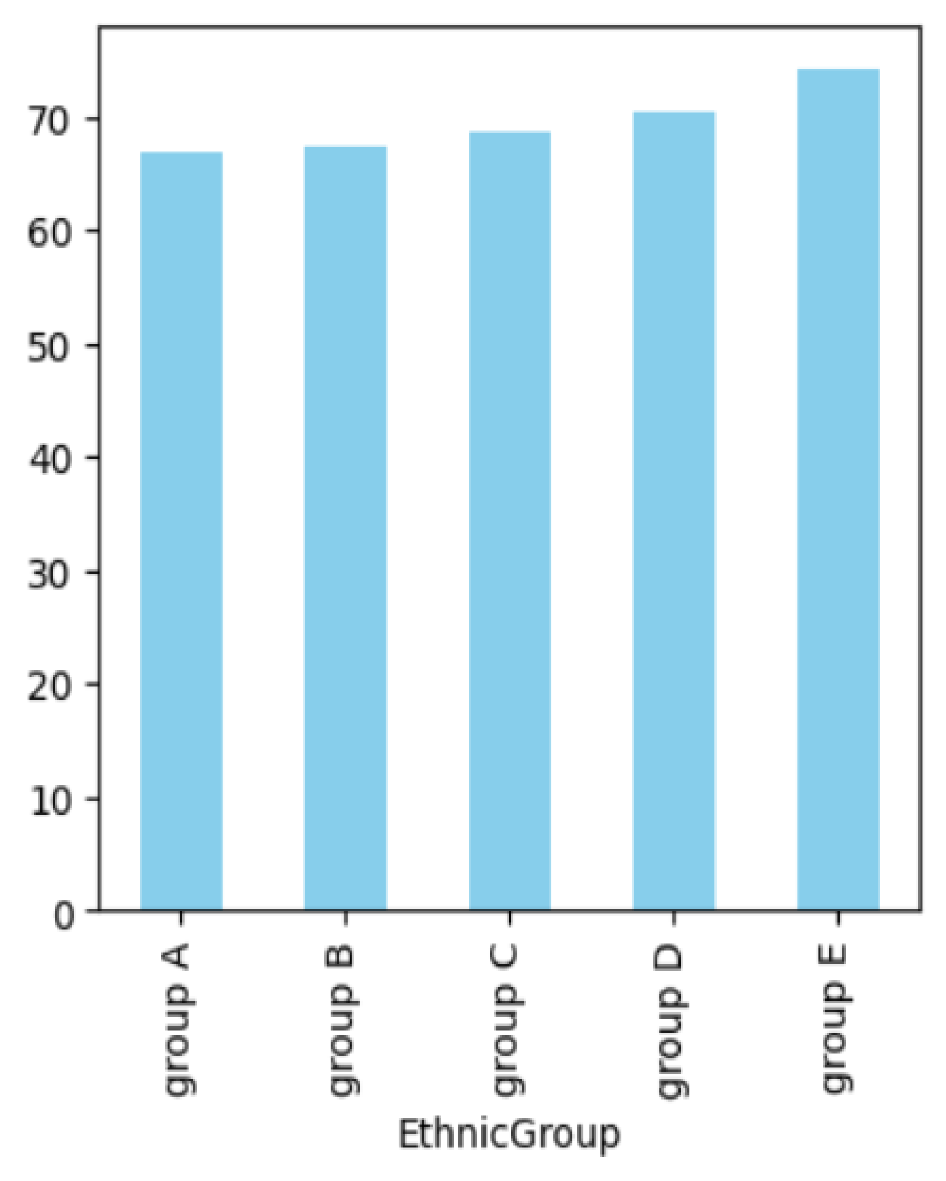


VISUALISING THE RELATION BETWEEN MATHS_SCORE, READING_SCORE, WRITING_SCORE WITH ETHNIC GROUP



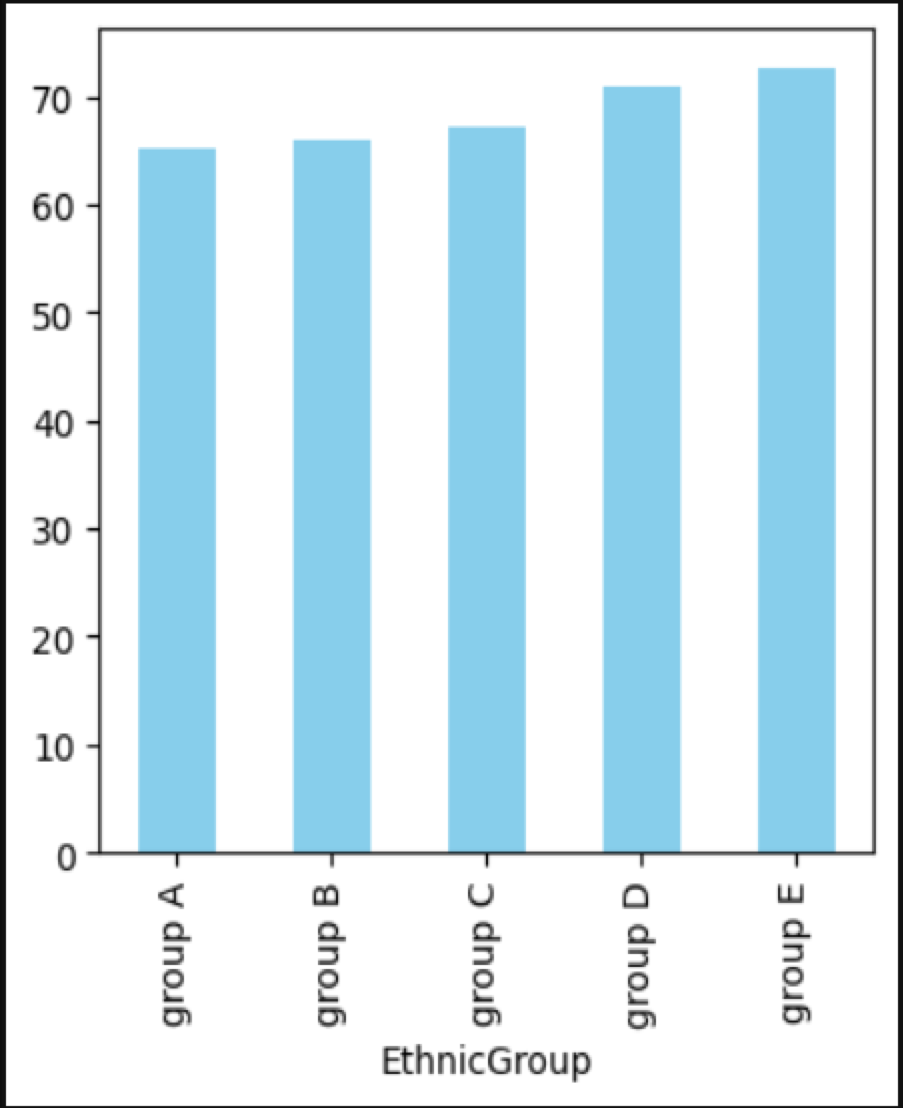

```
x=df.groupby('EthnicGroup')['MathScore'].mean().sort_values()
```

```
plt.figure(figsize=(4, 4))
x.plot(kind='bar', color='skyblue')
plt.show()
```




```
y=df.groupby('EthnicGroup')['ReadingScore'].mean().sort_values()
```

```
plt.figure(figsize=(4, 4))
y.plot(kind='bar', color='skyblue')
plt.show()
```




```
z=df.groupby('EthnicGroup')['WritingScore'].mean().sort_values()
```

```
plt.figure(figsize=(4, 4))
z.plot(kind='bar', color='skyblue')
plt.show()
```

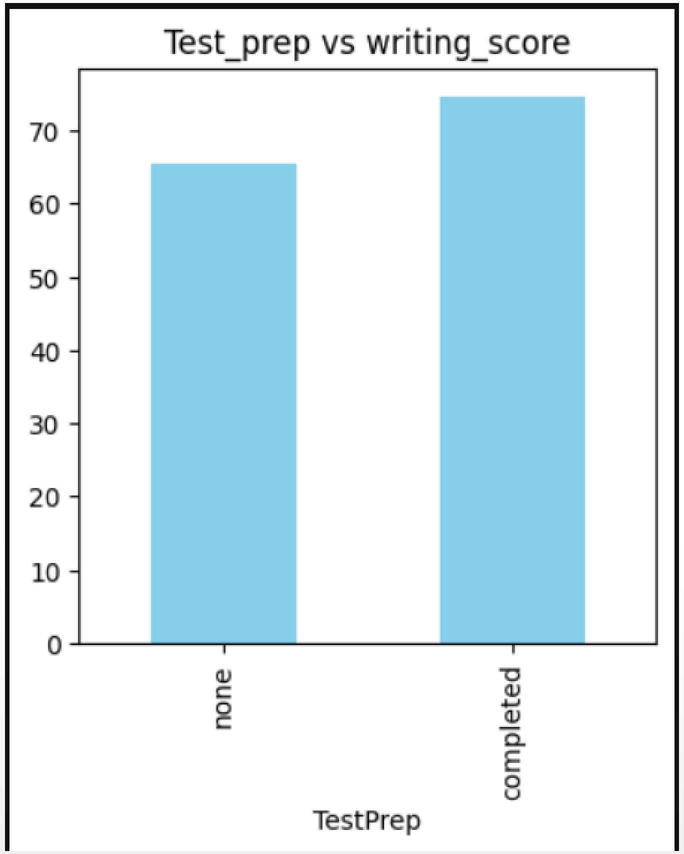
VISUALISING THE RELATION BETWEEN MATHS_SCORE, READING_SCORE, WRITING_SCORE WITH TEST PREPARATION

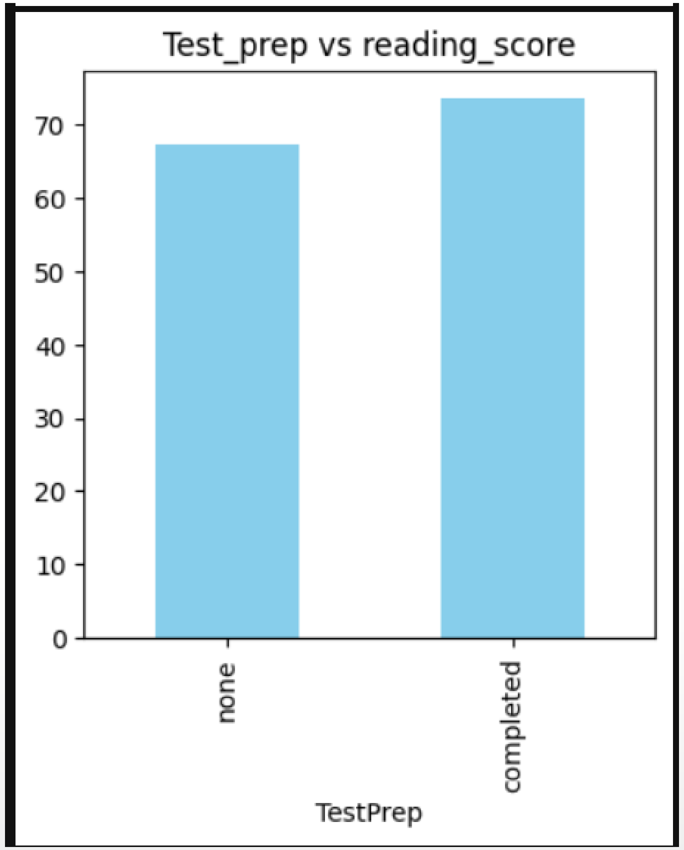
```
a=df.groupby('TestPrep')['MathScore'].mean().sort_values()
b=df.groupby('TestPrep')['ReadingScore'].mean().sort_values()
c=df.groupby('TestPrep')['WritingScore'].mean().sort_values()
```

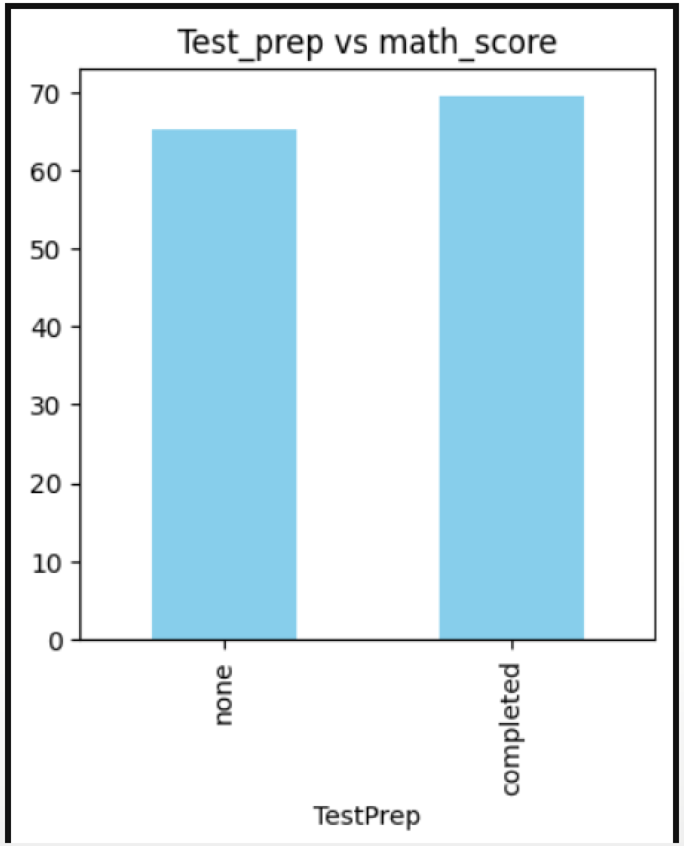
```
plt.figure(figsize=(4, 4))
a.plot(kind='bar', color='skyblue')
plt.title("Test_prep vs math_score")
plt.show()

plt.figure(figsize=(4, 4))
b.plot(kind='bar', color='skyblue')
plt.title("Test_prep vs reading_score")
plt.show()

plt.figure(figsize=(4, 4))
c.plot(kind='bar', color='skyblue')
plt.title("Test_prep vs writing_score")
plt.show()
```


Test_prep vs writing_score
70
60
50
40
30
20
10
0
none
completed
TestPrep


Test_prep vs reading_score
70
60
50
40
30
20
10
0
none
completed
TestPrep


Test_prep vs math_score
70
60
50
40
30
20
10
0
none
completed
TestPrep

SUMMARY:

1. There were many null values, which was been replaced by their mode values.
2. The count of males were lesser than the females.
3. The distribution of group C in ethnic group was the highest i.e. 36.1% and group A was the least.
4. It was visualized that the student's score were highest, if their parents had a master degree and the score were less when the parent had only the high school degree.
5. The count of standard type of lunch were more than the free/reduced type.
6. It was observed that less number of students were prepared for the test.
7. It was seen that there was no significant effect of parent's marital status with the math, reading and writing score.
8. There is no significant relation between the scores obtained by students who played sports regularly, play sometimes or never play sports.
9. It was seen that students score less marks in maths as compared to reading and writing.
10. The ethnic group E scores highest and group A scores the least.

THANK YOU!