

# Students marks insights

## Problem statement:

This project analyzes the factors influencing student performance by examining a variety of attributes. The dataset contains the following columns:

- Gender: Male or Female.
- Ethnic Group: The ethnic background of the student.
- Parents' Education: The highest education level attained by the student's parents.
- Parents' Marital Status: Whether the parents are married, divorced, or separated.
- Weekly Study Hours: The average number of hours the student dedicates to studying each week.
- Transport Means: The primary mode of transport used by the student.
- Lunch Type: Whether the student receives free or paid lunch.
- Is First Child: Whether the student is the first child in the family.
- NRI Siblings: Whether the student has siblings living abroad.
- Math Score: The student's score in math.
- Reading Score: The student's score in reading.
- Writing Score: The student's score in writing.

## Objectives

- Analyze how factors like gender, ethnicity, and parental education impact student performance in math, reading, and writing.
- Identify key influences such as weekly study hours, lunch type, and transport means on student marks.
- Explore correlations between math, reading, and writing scores.
- Investigate the effect of family background factors marital status on academic outcomes.
- Provide insights to help improve student performance through data-driven interventions.

In [168...

# Importing libraries  
import numpy as np  
import pandas as pd  
import matplotlib.pyplot as plt  
import seaborn as sns

In [132...

df=pd.read\_csv("students\_data.csv")

In [133...

df.head()

Out[133...

	Unnamed: 0	Gender	EthnicGroup	ParentEduc	LunchType	TestPrep	ParentMaritalStatus	PracticeSport	IsFirstChild	NrSiblings
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

In [134...

df.columns

Out[134...

Index(['Unnamed: 0', 'Gender', 'EthnicGroup', 'ParentEduc', 'LunchType', 'TestPrep', 'ParentMaritalStatus', 'PracticeSport', 'IsFirstChild', 'NrSiblings', 'TransportMeans', 'WklyStudyHours', 'MathScore', 'ReadingScore', 'WritingScore'], dtype='object')

In [135...

df.shape

Out[135...

(30641, 15)

In [136...

df.dtypes

```
Out[136... Unnamed: 0          int64
Gender            object
EthnicGroup       object
ParentEduc        object
LunchType         object
TestPrep          object
ParentMaritalStatus object
PracticeSport     object
IsFirstChild      object
NrSiblings        float64
TransportMeans     object
WklyStudyHours    object
MathScore         int64
ReadingScore      int64
WritingScore      int64
dtype: object
```

```
In [137... df.duplicated(keep="first")
```

```
Out[137... 0      False
1      False
2      False
3      False
4      False
...
30636  False
30637  False
30638  False
30639  False
30640  False
Length: 30641, dtype: bool
```

```
In [138... df.describe()
```

	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 [139... 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 [140... df.isnull().sum()
```

```
Out[140... 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
```

Drop unnamed column

```
In [141... df=df.drop("Unnamed: 0",axis=1)
```

```
In [142... df.head()
```

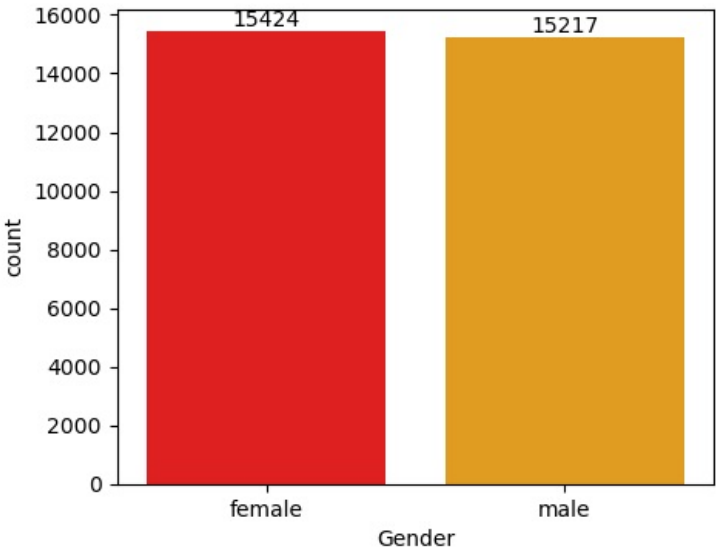
Out[142...]	Gender	EthnicGroup	ParentEduc	LunchType	TestPrep	ParentMaritalStatus	PracticeSport	IsFirstChild	NrSiblings	TransportM
0	female	NaN	bachelor's degree	standard	none	married	regularly	yes	3.0	school bus
1	female	group C	some college	standard	NaN	married	sometimes	yes	0.0	parent drop-off
2	female	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	parent drop-off
4	male	group C	some college	standard	none	married	sometimes	yes	0.0	school bus

Gender distribution

```
In [143... df["Gender"].value_counts()
```

```
Out[143... Gender
female    15424
male      15217
Name: count, dtype: int64
```

```
In [144... plt.figure(figsize=(5,4))
ax=sns.countplot(data=df,x="Gender", hue="Gender",palette=['red', 'orange'],legend=False)
for container in ax.containers:
    ax.bar_label(container)
plt.show()
```

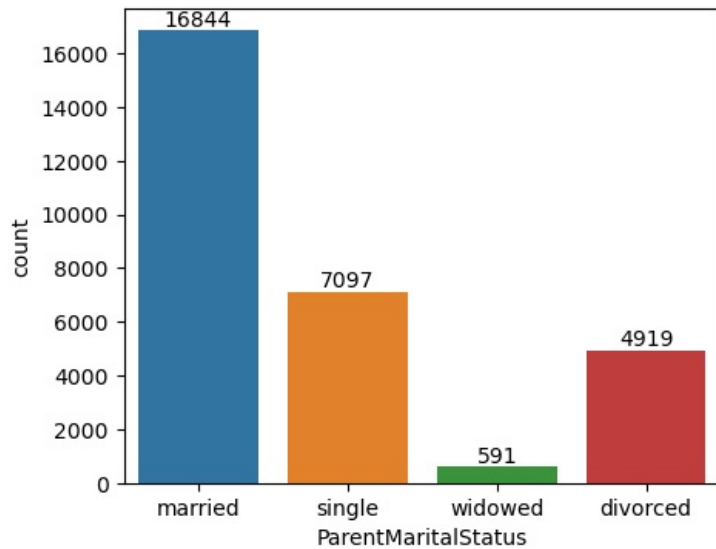


Parents martial status breakdown

```
In [145... df["ParentMaritalStatus"].value_counts()
```

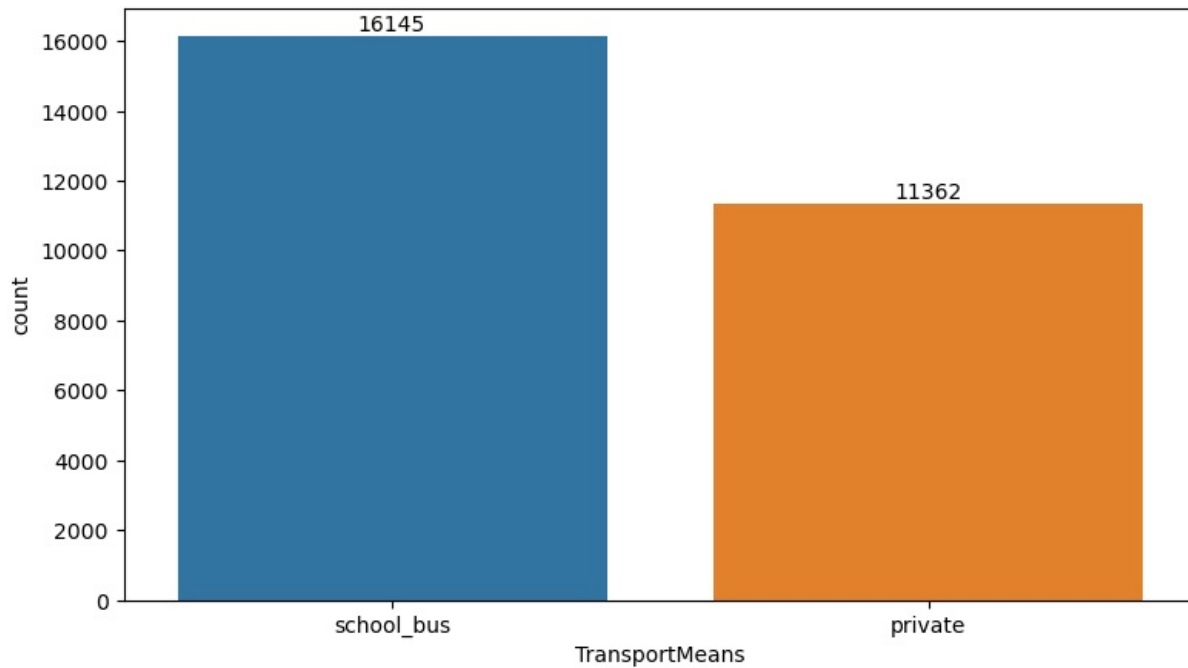
```
Out[145... ParentMaritalStatus
married      16844
single       7097
divorced     4919
widowed      591
Name: count, dtype: int64
```

```
In [146... plt.figure(figsize=(5,4))
ax=sns.countplot(data=df,x="ParentMaritalStatus", hue="ParentMaritalStatus", legend=False)
for container in ax.containers:
    ax.bar_label(container)
plt.show()
```



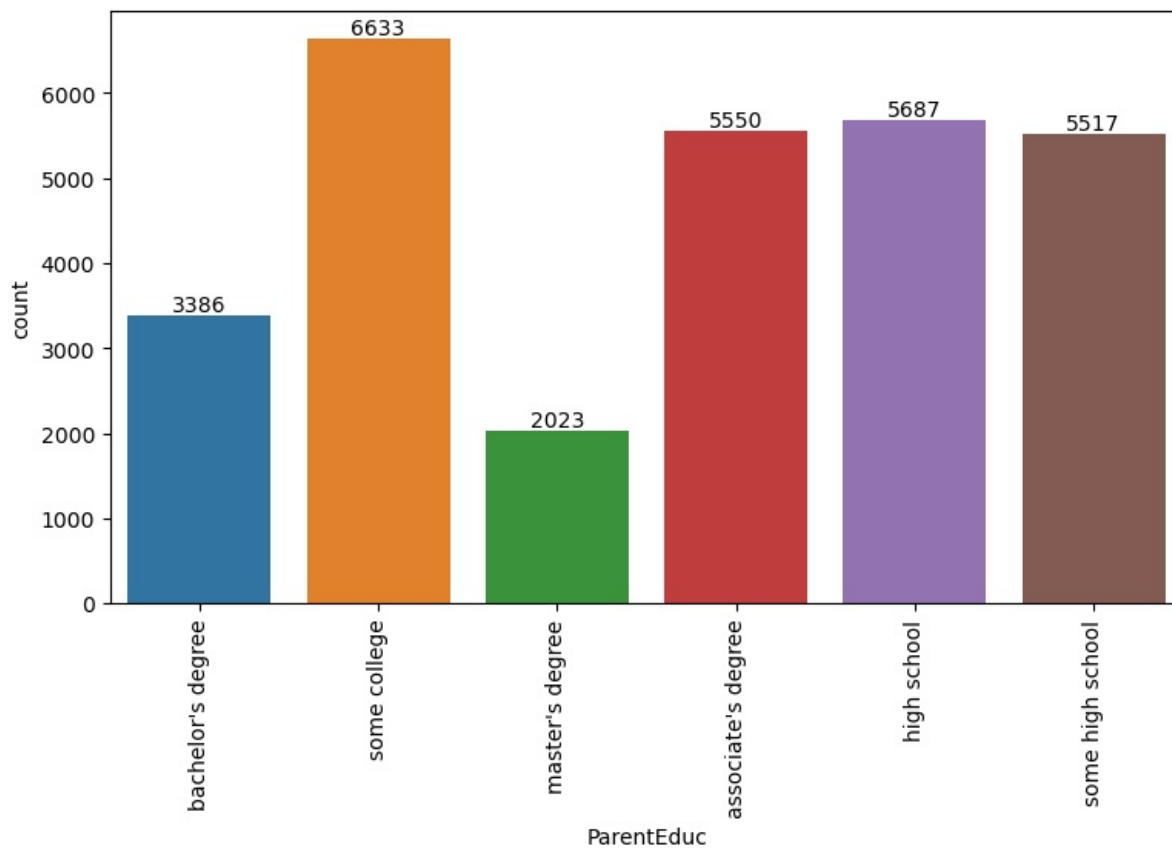
## Transport means type

```
In [147... plt.figure(figsize=(9,5))
ax=sns.countplot(data=df,x="TransportMeans", hue="TransportMeans", legend=False)
#ax.bar_label(ax.containers[0])
for container in ax.containers:
    ax.bar_label(container)
plt.show()
```



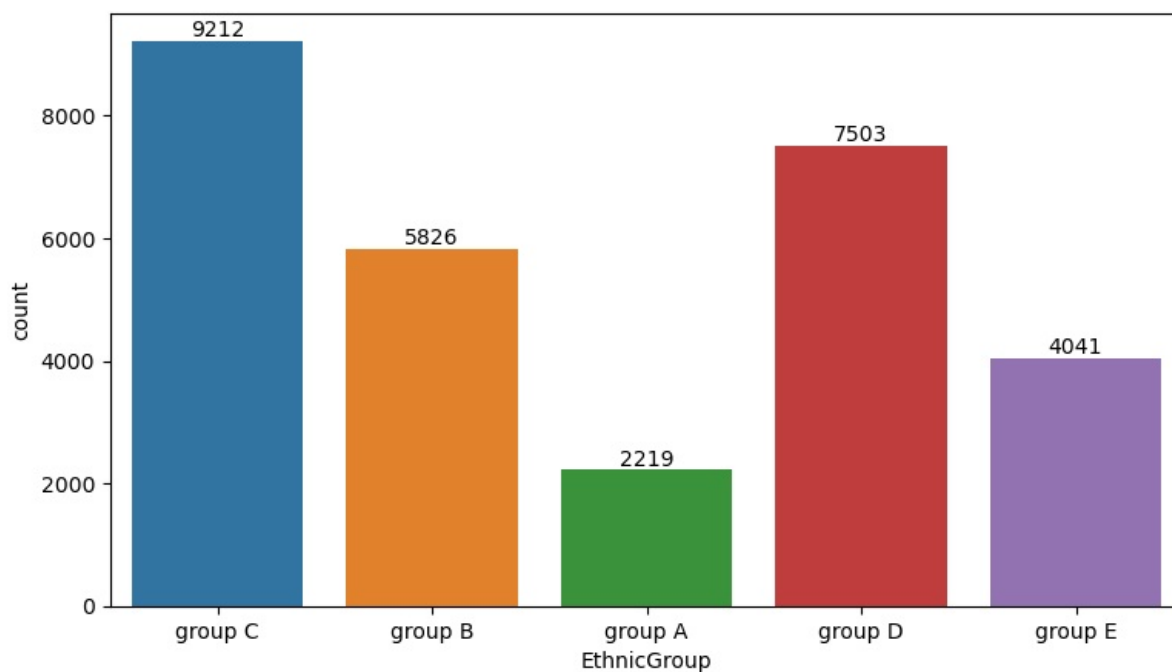
## Parents college degree

```
In [148... plt.figure(figsize=(9,5))
ax=sns.countplot(data=df,x="ParentEduc", hue="ParentEduc", legend=False)
#ax.bar_label(ax.containers[0])
for container in ax.containers:
    ax.bar_label(container)
plt.xticks(rotation=90)
plt.show()
```



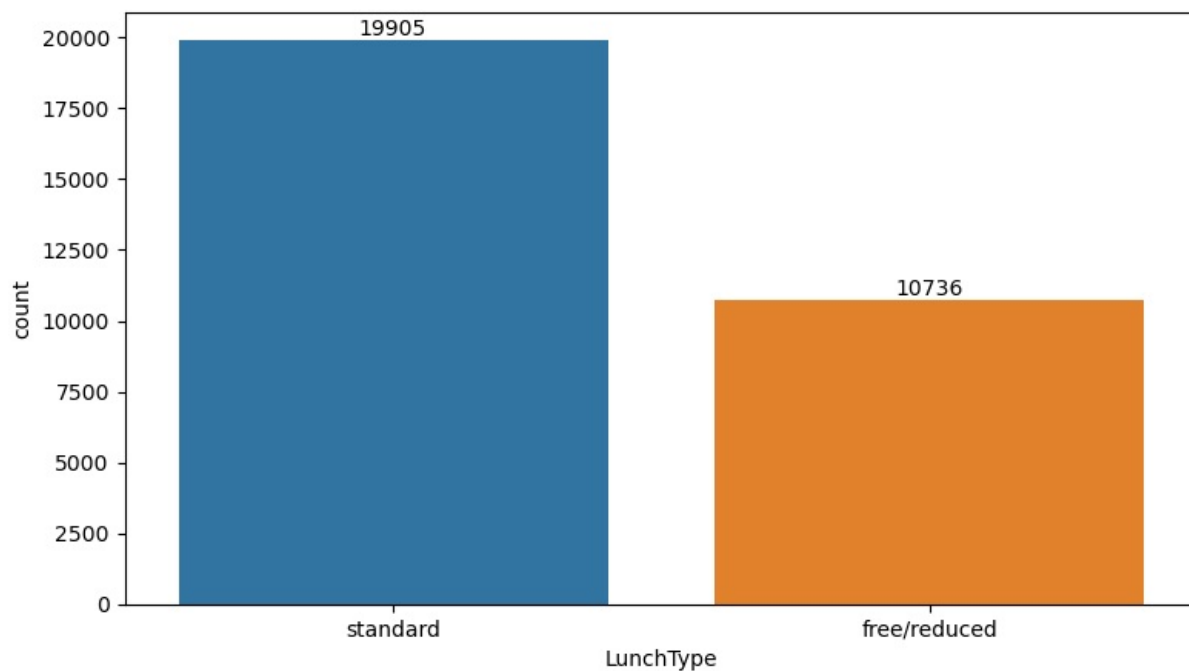
Distribution of ethic group

```
In [149.. plt.figure(figsize=(9,5))
ax=sns.countplot(data=df,x="EthnicGroup",hue="EthnicGroup",legend=False)
#ax.bar_label(ax.containers[0])
for container in ax.containers:
    ax.bar_label(container)
plt.show()
```



Lunch type breakdown

```
In [151.. plt.figure(figsize=(9,5))
ax=sns.countplot(data=df,x="LunchType",hue="LunchType",legend=False)
#ax.bar_label(ax.containers[0])
for container in ax.containers:
    ax.bar_label(container)
plt.show()
```



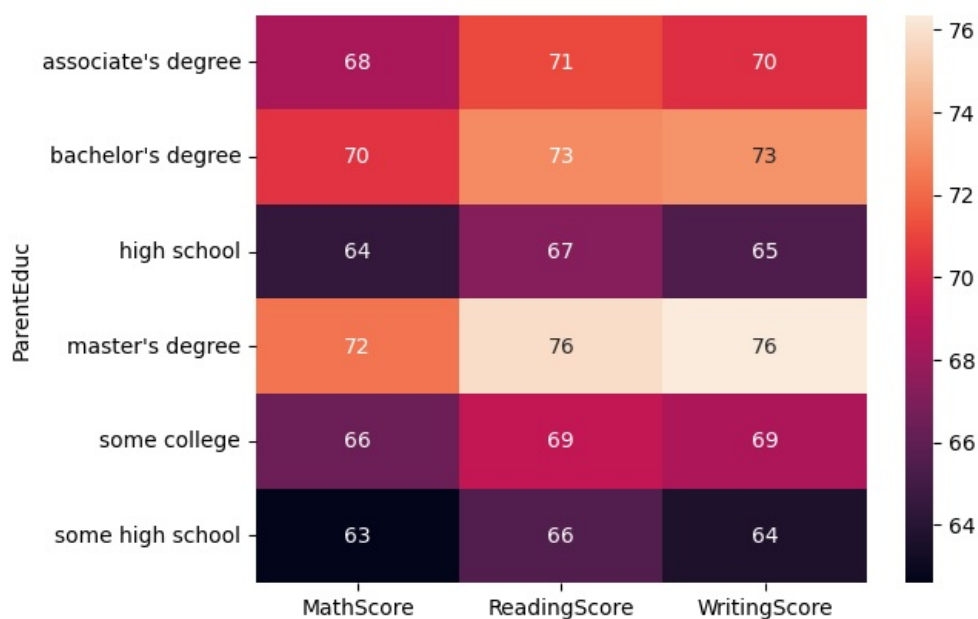
Does parents education affect their children score?

```
In [153.. gb=df.groupby("ParentEduc").agg({"MathScore":"mean","ReadingScore":"mean","WritingScore":"mean"})
gb
```

```
Out[153..
```

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	66.390472	69.179708	68.501432
some high school	62.584013	65.510785	63.632409

```
In [154.. sns.heatmap(gb, annot=True)
plt.show()
```



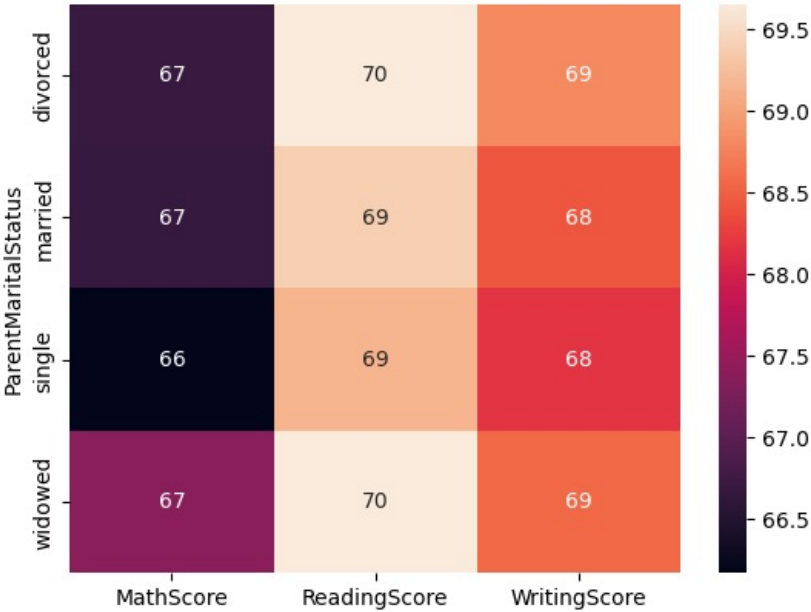
```
In [156.. ga=df.groupby("ParentMaritalStatus").agg({"MathScore":"mean","ReadingScore":"mean","WritingScore":"mean"})
ga
```

Out [156..

	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

In [157..

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



In [158..

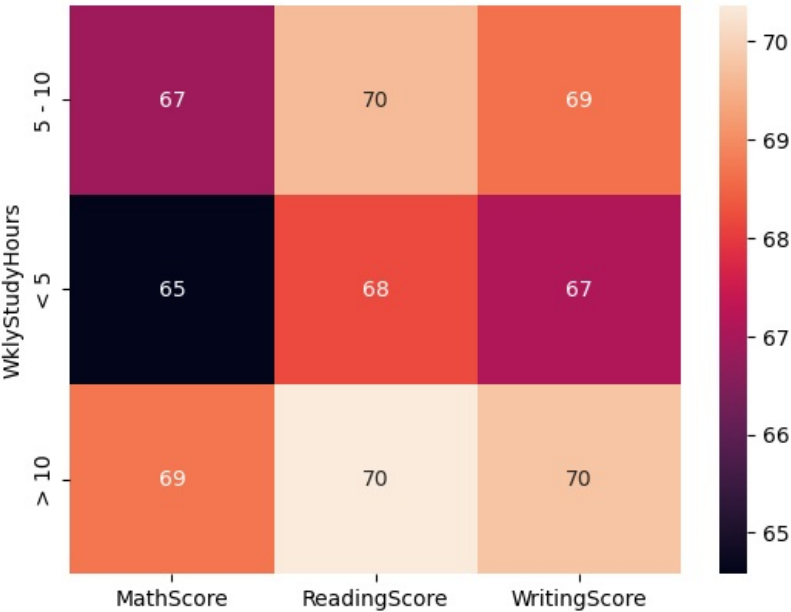
```
gc=df.groupby("WklyStudyHours").agg({"MathScore":"mean","ReadingScore":"mean","WritingScore":"mean"})
gc
```

Out [158..

	MathScore	ReadingScore	WritingScore
WklyStudyHours			
5 - 10	66.870491	69.660532	68.636280
< 5	64.580359	68.176135	67.090192
> 10	68.696655	70.365436	69.777778

In [159..

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



In [160..

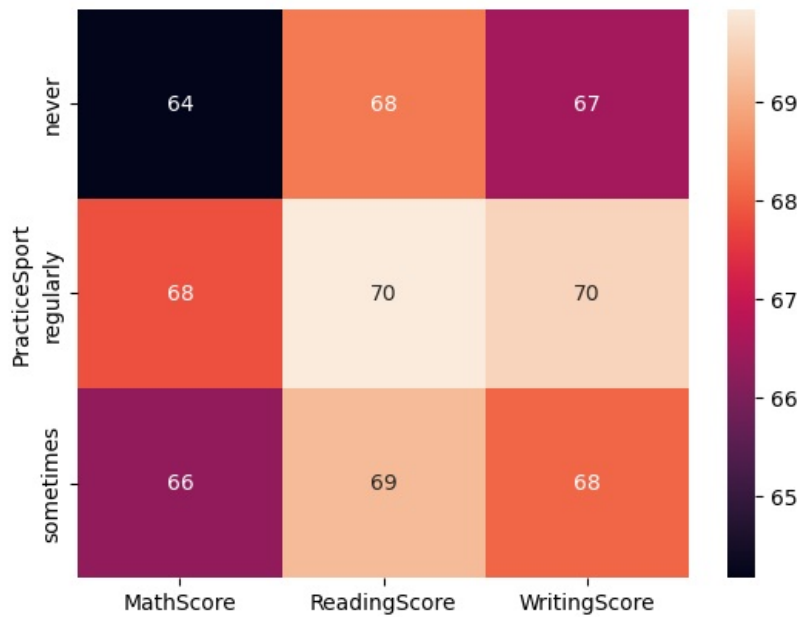
```
gaa=df.groupby("PracticeSport").agg({"MathScore":"mean","ReadingScore":"mean","WritingScore":"mean"})
gaa
```

Out[160..

	MathScore	ReadingScore	WritingScore
PracticeSport			
never	64.171079	68.337662	66.522727
regularly	67.839155	69.943019	69.604003
sometimes	66.274831	69.241307	68.072438

In [161..

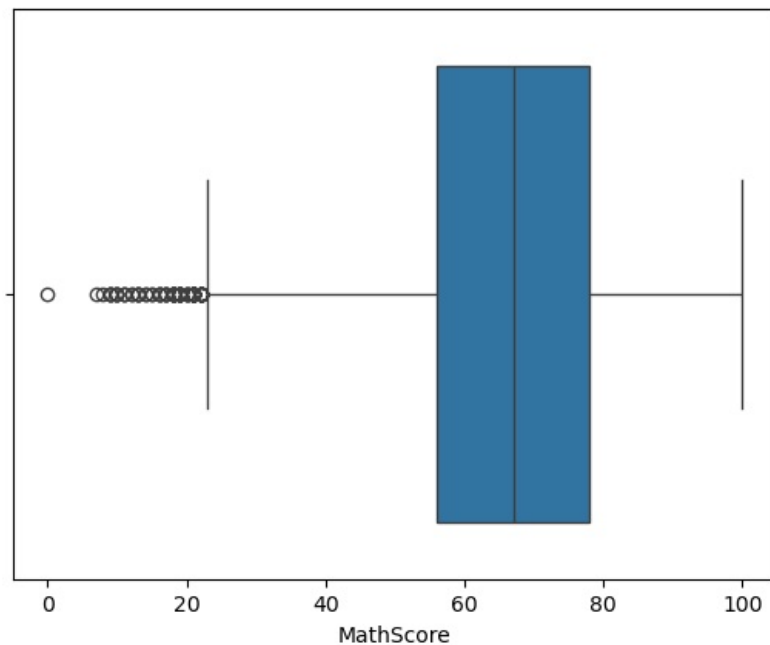
```
sns.heatmap(gaa,annot=True)  
plt.show()
```



### Detecting outliers

In [167..

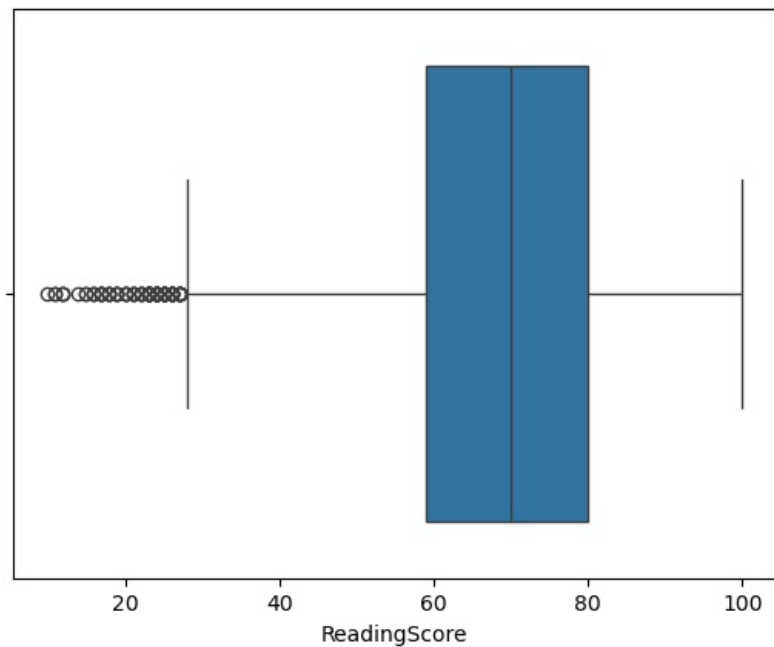
```
sns.boxplot(data=df,x="MathScore")  
plt.show()
```



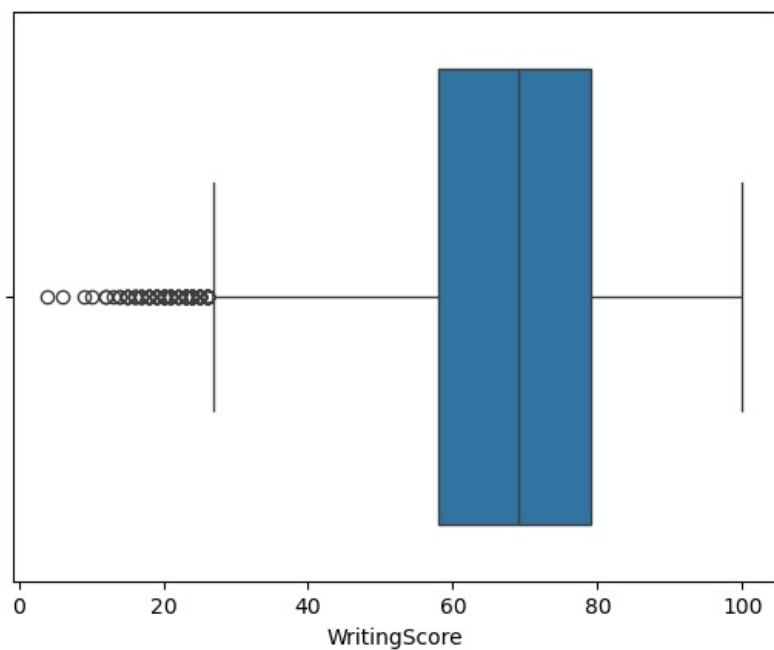
In [164..

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





```
In [165]: sns.boxplot(data=df, x="WritingScore")
plt.show()
```



## key takeaways

Female are slightly more than males.

Most parents martial status are married and very few of them are widowed.

Most parents education qualification are somewhat educated from some college and very few of them have master degree.

Majority of them are from group D ethnic group and minority of them are from Group A.

Parents education affects their children marks if their parents are highly educated their kids likely to get more marks and vice versa.

Parents Marital status have not much affects their children marks.

Kids who study more hours get slightly more marks in all subjects.

Kids who practice sports are likely to get good marks.

