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

df = pd.read_csv(r"C:\Users\Salman Ramzan\Downloads\archive (3)\
Expanded_data_with_more_features.csv")

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
```

```
df.shape
```

```
(30641, 15)
```

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

```
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
```

```
df.duplicated()
```

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

```
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
```

```
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

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
```

Data Cleaning

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

df.head()
```

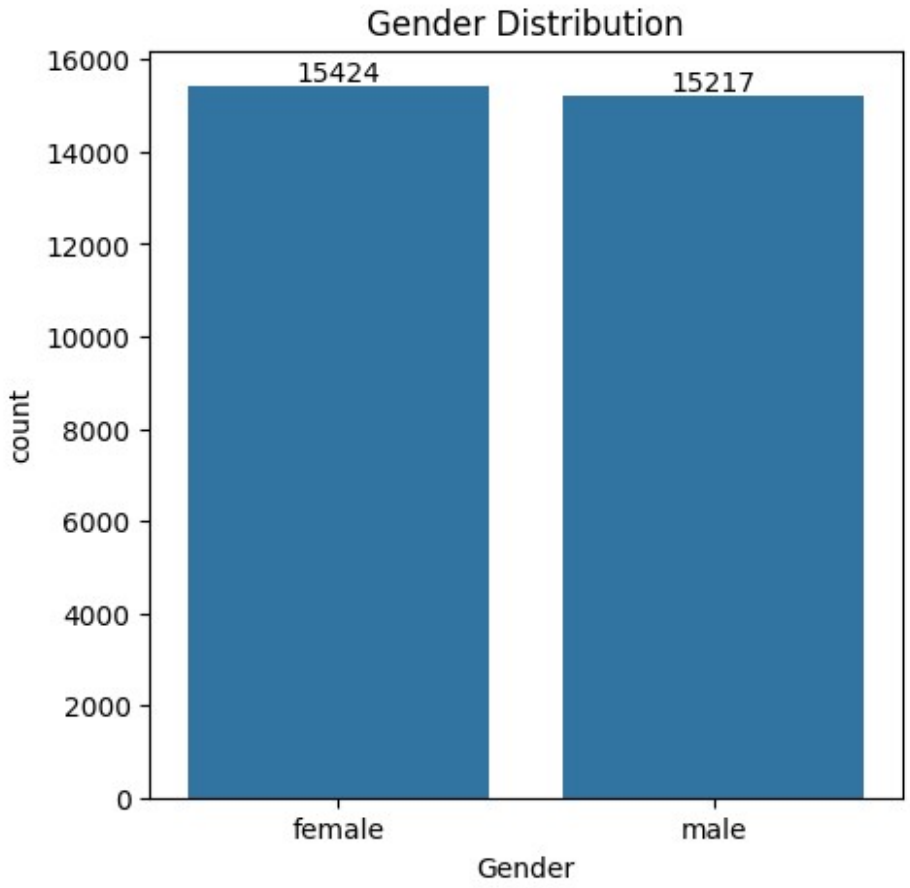
```
   Gender EthnicGroup          ParentEduc     LunchType TestPrep  \
0  female         NaN   bachelor's degree      standard     none
1  female     group C        some college      standard      NaN
2  female     group B     master's degree      standard     none
3    male     group A  associate's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

```
# For replacing values
# df["WklyStudyHours"]=df["WklyStudyHours"].str.replace("5 - 10,
89"),,waise hi
```

EDA

```
# Gender Distribution
```

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



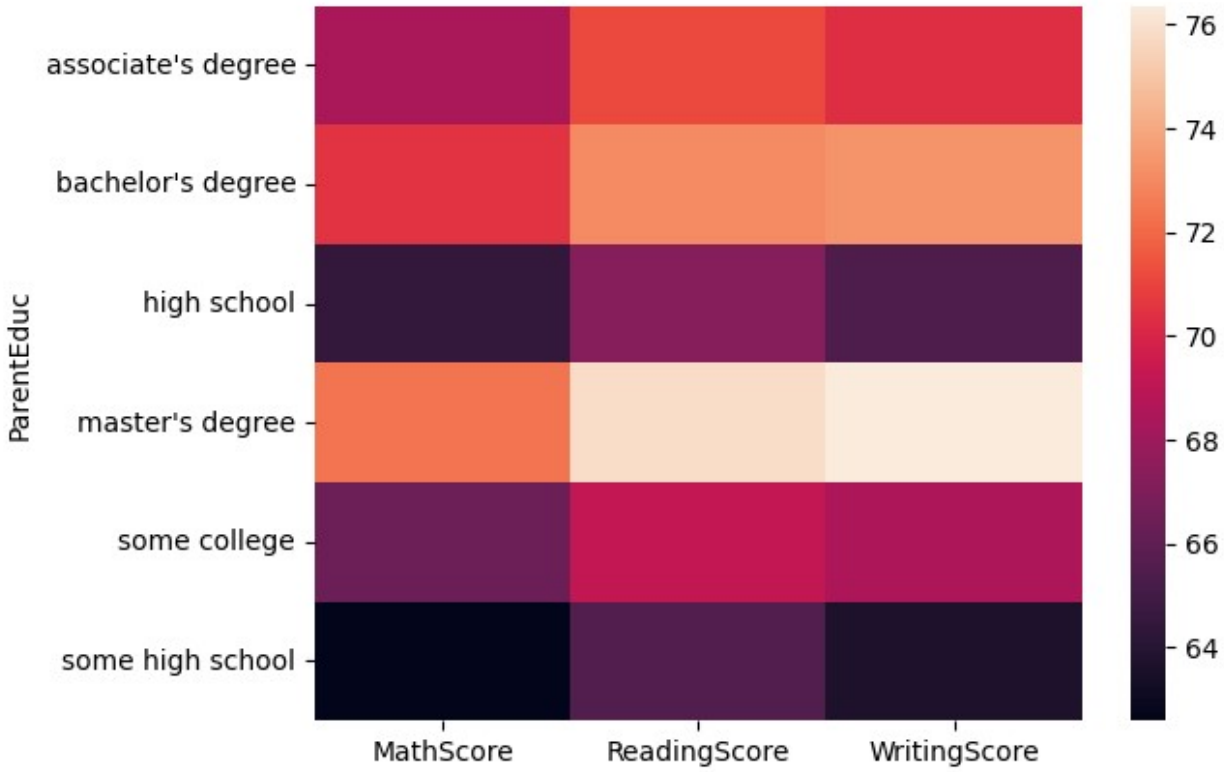
From the above analysis we analysed that the number of female in the data is more than male

Score on the base of Parent's Education

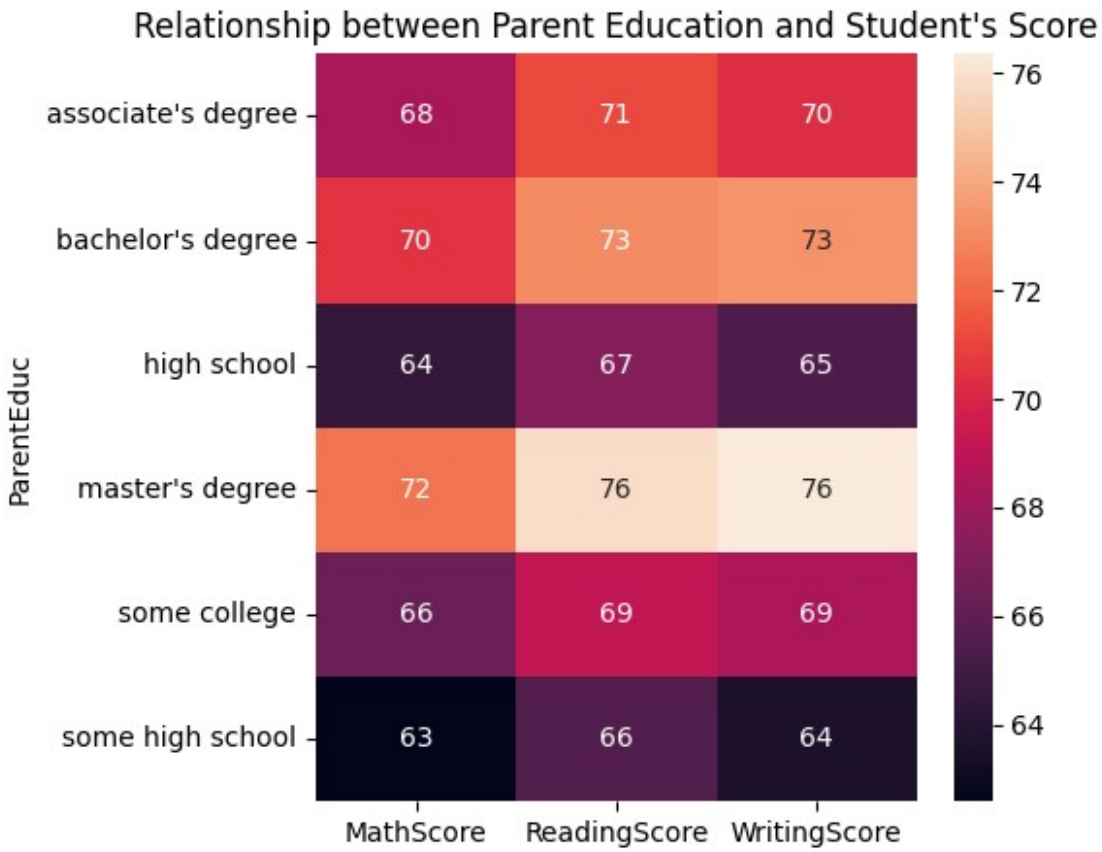
```
gb =
df.groupby("ParentEduc").agg({"MathScore":"mean","ReadingScore":"mean"
,"WritingScore":"mean"})
print(gb)
```

```
                    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

sns.heatmap(gb)

<Axes: ylabel='ParentEduc'>
```




```
#for showing values in heatmap
plt.figure(figsize=(5,5))
sns.heatmap(gb, annot = True)
plt.title("Relationship between Parent Education and Student's Score")

Text(0.5, 1.0, "Relationship between Parent Education and Student's
Score")
```



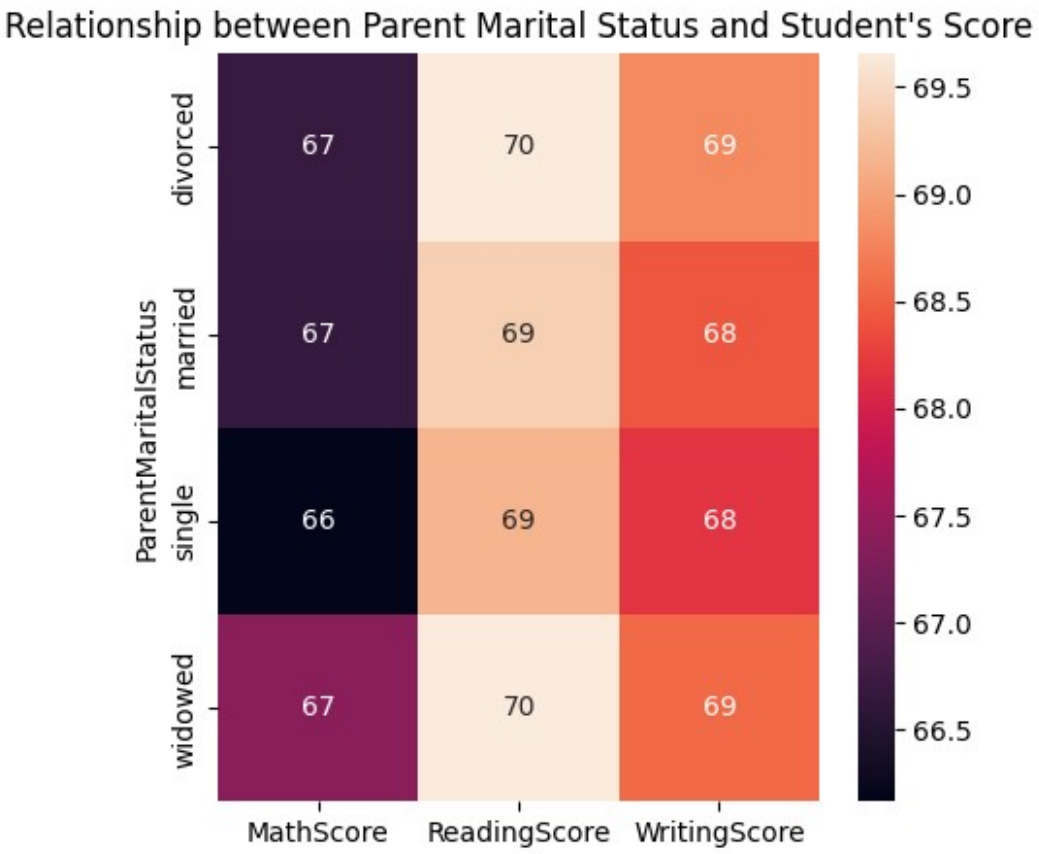
From the above chart we analysed the education of parent has a good impact on the student's score

Score on the base of Parent Marital Status

```
gb1=
df.groupby("ParentMaritalStatus").agg({"MathScore":"mean","ReadingScor
e":"mean","WritingScore":"mean"})
print(gb1)
```

```
                     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(gb1, annot=True)
plt.title("Relationship between Parent Marital Status and Student's
Score")
plt.show()
```



From above chart we can analysis that there is no/negligible impact of parent marital status on Student's score

```
df.head()
```

	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
```

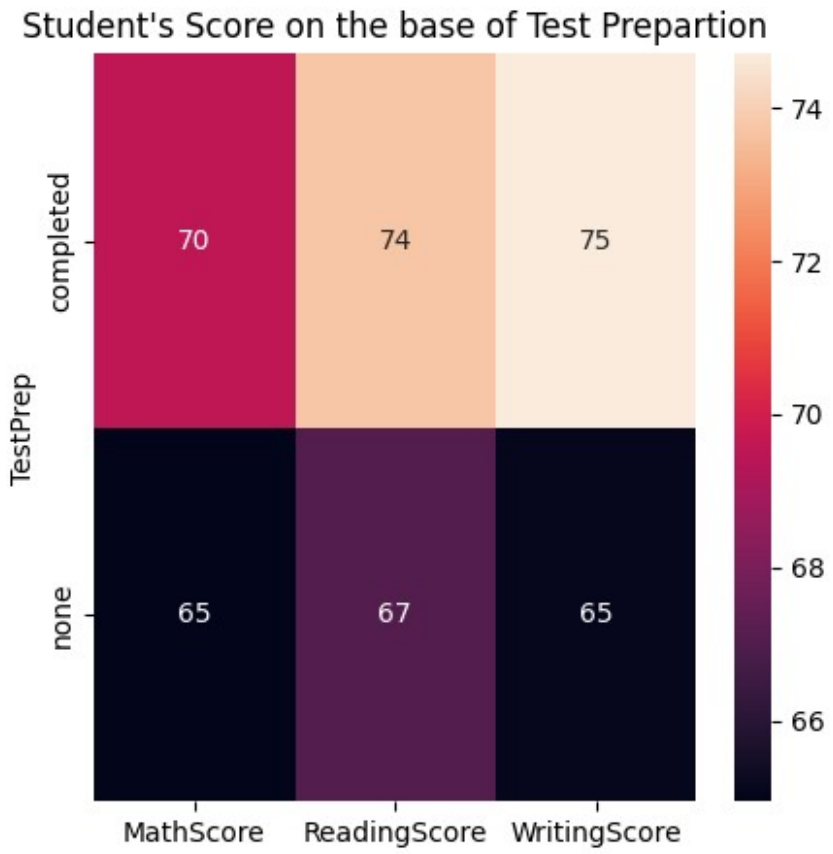
```
  WklyStudyHours  MathScore  ReadingScore  WritingScore
0            < 5         71            71            74
1         5 - 10         69            90            88
2            < 5         87            93            91
3         5 - 10         45            56            42
4         5 - 10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

Score on the base of Test Preparation

```
gb2 =
df.groupby("TestPrep").agg({"MathScore":"mean","ReadingScore":"mean","
WritingScore":"mean"})
print(gb2)
```

```
           MathScore  ReadingScore  WritingScore
TestPrep
completed   69.54666     73.732998     74.703265
none        64.94877     67.051071     65.092756
```

```
plt.figure(figsize=(5,5))
sns.heatmap(gb2, annot=True)
plt.title("Student's Score on the base of Test Prepartion")
plt.show()
```



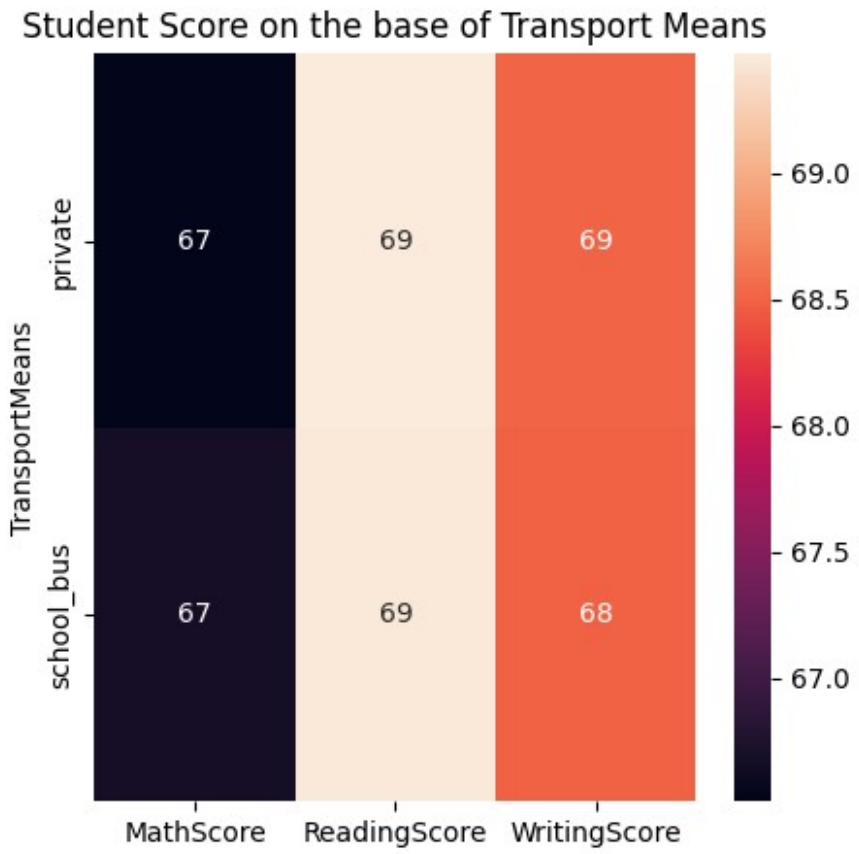
From above analysis we can see there is large impact of "Test Preparation" on the student's score

Score on the base of Transport Means

```
gb3=
df.groupby("TransportMeans").agg({"MathScore":"mean","ReadingScore":"m
ean","WritingScore":"mean"})
print(gb3)
```

```
                MathScore  ReadingScore  WritingScore
TransportMeans
private         66.511354     69.472364     68.509593
school_bus      66.674636     69.446206     68.492351
```

```
plt.figure(figsize=(5,5))
sns.heatmap(gb3, annot=True)
plt.title("Student Score on the base of Transport Means")
plt.show()
```



From above chart we can analysis that there is no/negligible impact of "Transport Mean" status on Student's score

```
df.head()
```

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

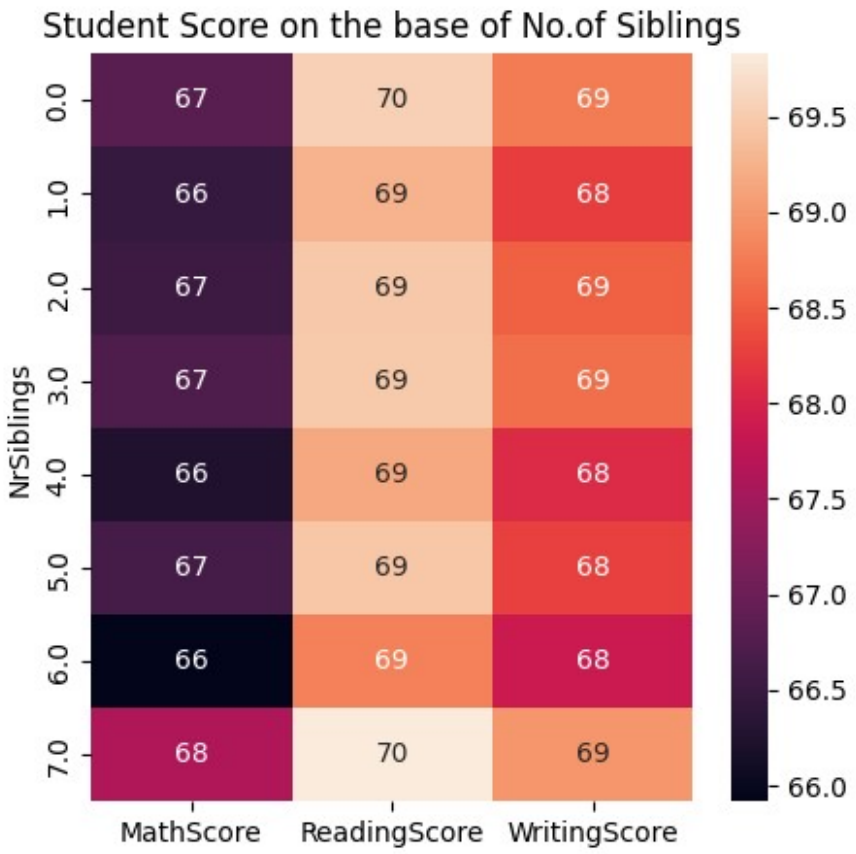
	WklyStudyHours	MathScore	ReadingScore	WritingScore
0	< 5	71	71	74
1	5 - 10	69	90	88
2	< 5	87	93	91
3	5 - 10	45	56	42
4	5 - 10	76	78	75

Score on the base on No.of Siblings

```
gb4 =
df.groupby("NrSiblings").agg({"MathScore":"mean","ReadingScore":"mean"
,"WritingScore":"mean"})
print(gb4)
```

NrSiblings	MathScore	ReadingScore	WritingScore
0.0	66.819449	69.547812	68.746515
1.0	66.473896	69.259097	68.245345
2.0	66.554934	69.472018	68.522533
3.0	66.719092	69.488159	68.650498
4.0	66.245495	69.144169	68.073444
5.0	66.630303	69.453788	68.282576
6.0	65.917219	68.801325	67.860927
7.0	67.615120	69.828179	68.986254

```
plt.figure(figsize=(5,5))
sns.heatmap(gb4, annot=True)
plt.title("Student Score on the base of No.of Siblings")
plt.show()
```

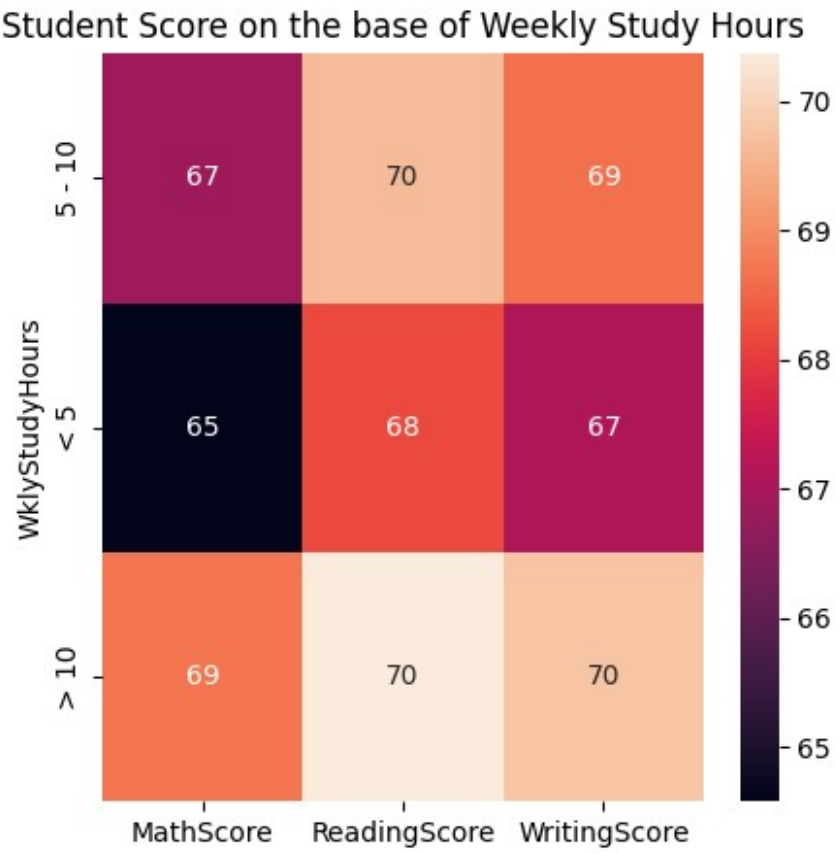


From above chart we can analysis that there is no/negligible impact of "No.of Siblings" on Student's score

```
gb5 =
df.groupby("WklyStudyHours").agg({"MathScore":"mean","ReadingScore":"m
ean","WritingScore":"mean"})
print(gb5)
```

```
                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
```

```
plt.figure(figsize=(5,5))
sns.heatmap(gb5, annot=True)
plt.title("Student Score on the base of Weekly Study Hours")
plt.show()
```



From above chart we can analysis that there is no/negligible impact of "Weekly Study Hours" on Student's score

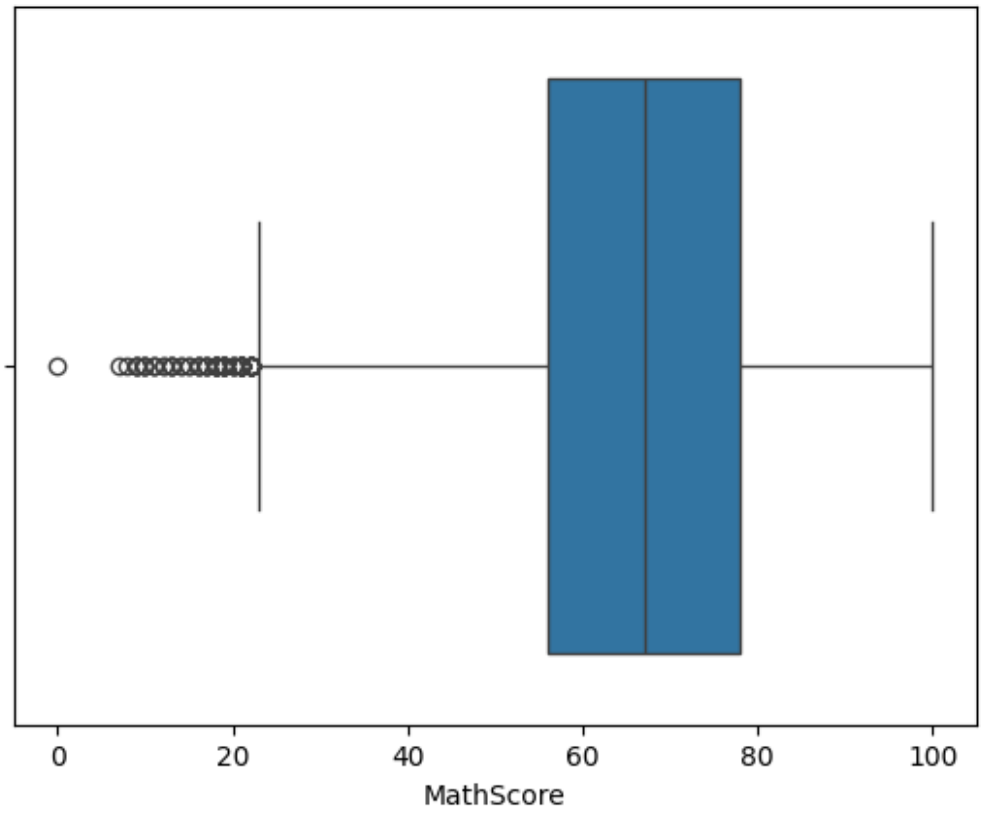
Checking Outliers

```
df.head()
```

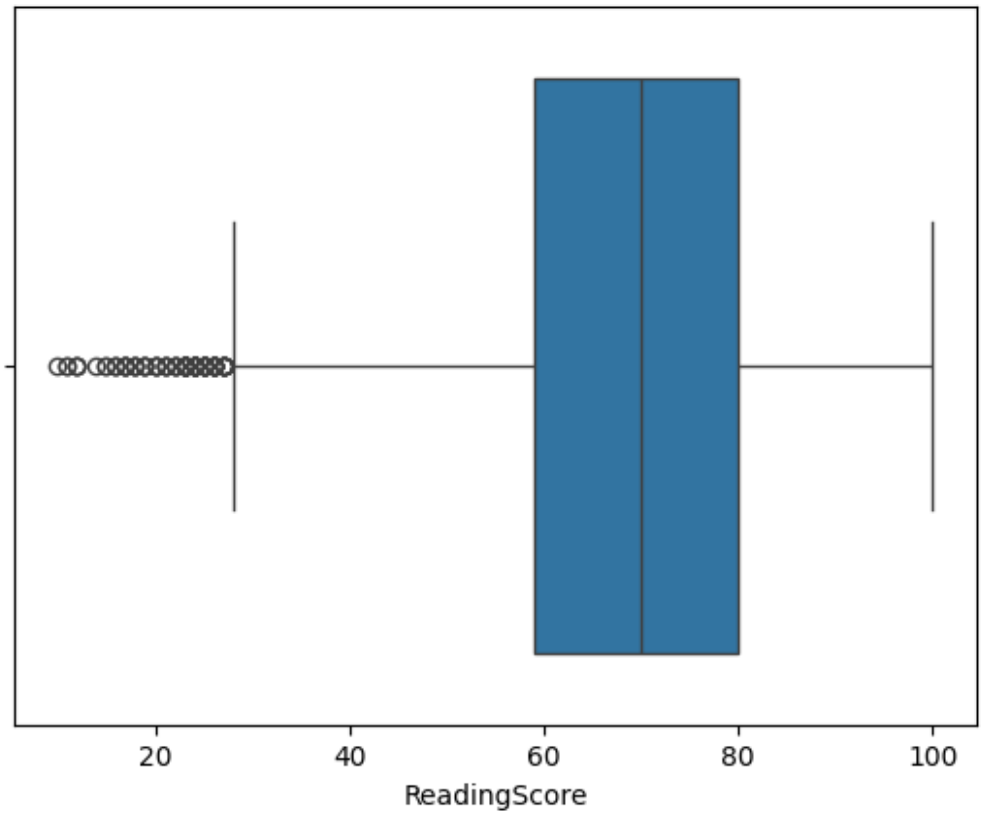
```
   Gender EthnicGroup          ParentEduc     LunchType TestPrep  \
0  female         NaN   bachelor's degree      standard     none
1  female     group C        some college      standard      NaN
2  female     group B     master's degree      standard     none
3    male     group A  associate's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

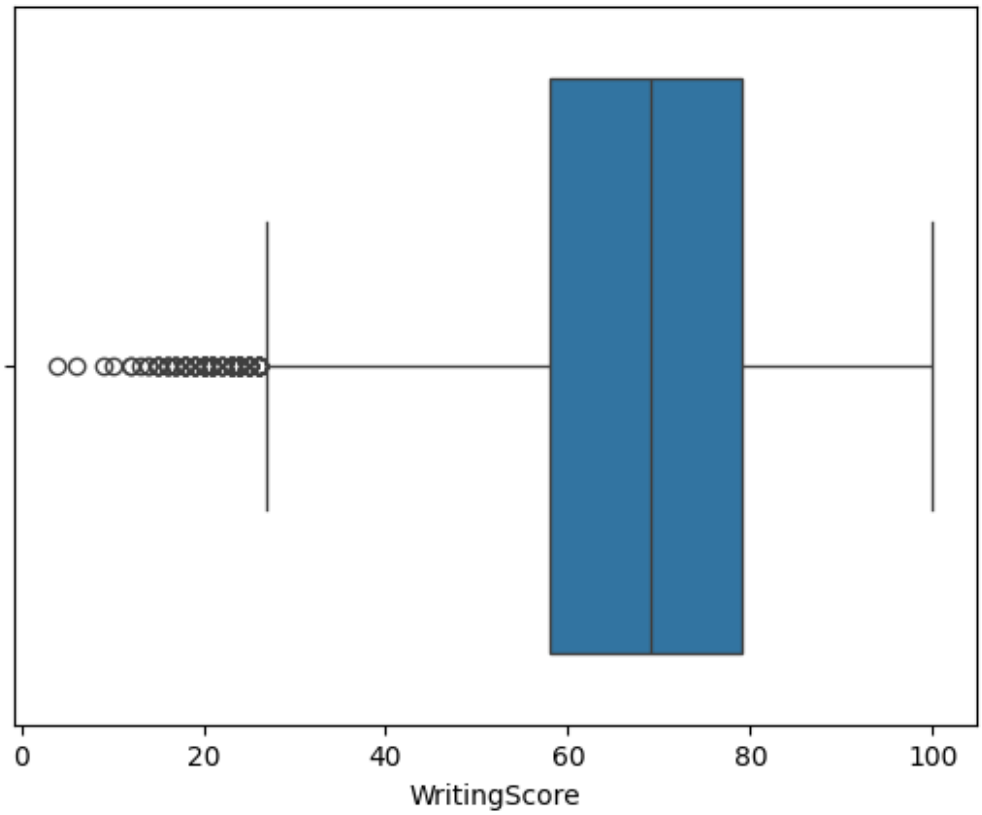
```
sns.boxplot(x="MathScore", data=df)
plt.show()
```




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

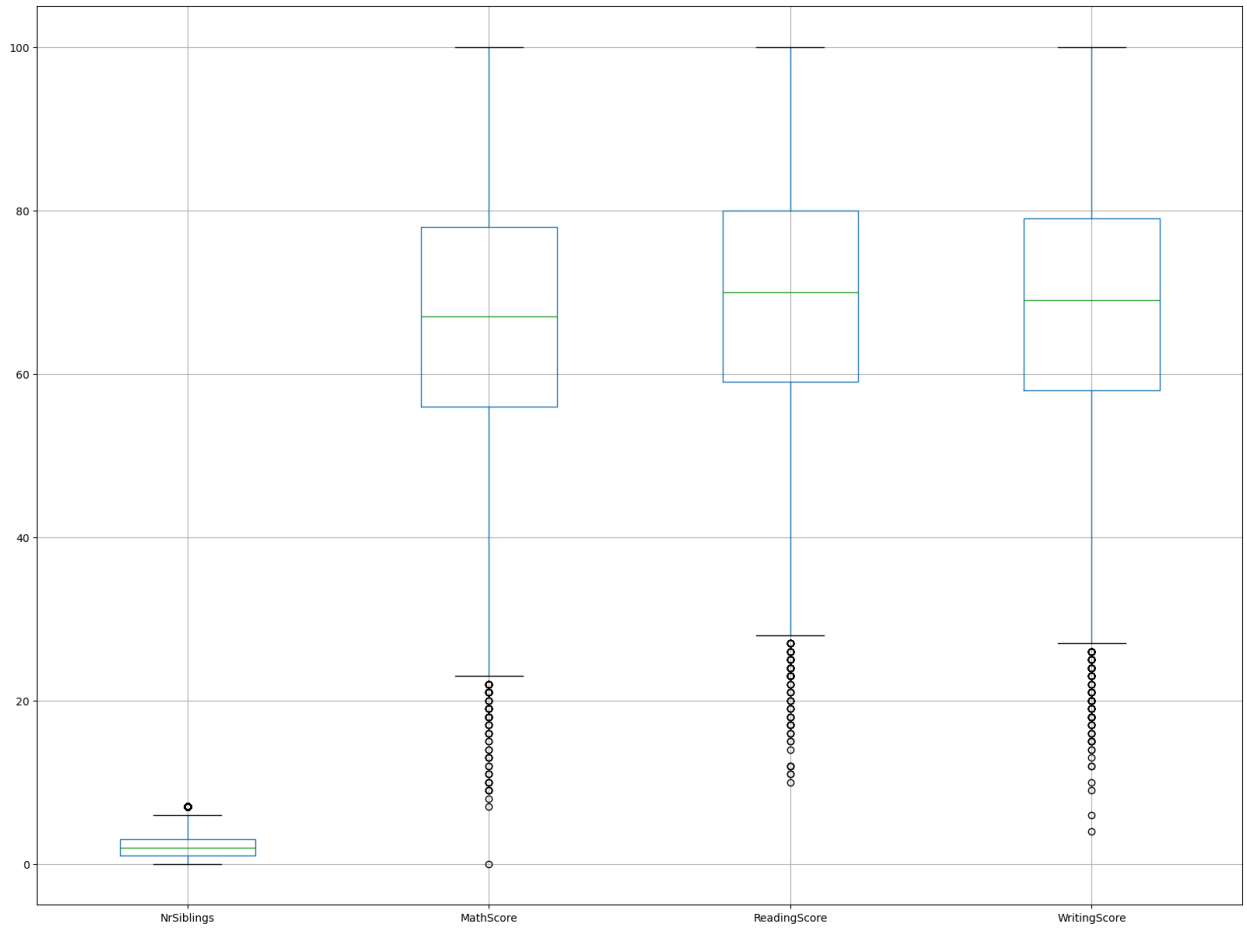



```
sns.boxplot(x="WritingScore",data=df)
plt.show()
```



From above graphs we analysed that students are comparitively weak in "Math" instead of reading and writing score

```
df.boxplot(figsize=(20,15))
plt.show()
```




```
df.head()
```

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

```
df["EthnicGroup"].unique()
```

```
array([nan, 'group C', 'group B', 'group A', 'group D', 'group E'],
      dtype=object)
```

```
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()

l = ["GroupA","GroupB","GroupC","GroupD","GroupE"]
mlist
=(GroupA["EthnicGroup"],GroupB["EthnicGroup"],GroupC["EthnicGroup"],Gr
oupD["EthnicGroup"],GroupE["EthnicGroup"])
plt.pie(mlist, labels = l,autopct="%1.2f%%")
plt.title("distribution of Ethnic Group")
plt.show()
```