Project:-Student result analysis

Presented by :- Sumit Sharma ¶

First of all i will import some library to analysis my data

Import Library

In [1]:

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

Import the data from system

In [2]:

```
df=pd.read_csv("D:\Extract data set\Expanded_data_with_more_features.csv")
df.head()
```

Out[2]:

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

after importing the my file path here i analysis that there are 15 column.

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

Out[6]:

	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 [7]:
```
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 [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
```

Drop unnamed column

In [10]:
```
df.columns
```

Out[10]:
```
Index(['Unnamed: 0', 'Gender', 'EthnicGroup', 'ParentEduc', 'LunchType',
       'TestPrep', 'ParentMaritalStatus', 'PracticeSport', 'IsFirstChild',
       'NrSiblings', 'TransportMeans', 'WklyStudyHours', 'MathScore',
       'ReadingScore', 'WritingScore'],
      dtype='object')
```

In [11]:
```
df=df.drop("Unnamed: 0",axis=1)
print(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 TransportMean
s  \
0             married     regularly          yes         3.0     school_bu
s
1             married     sometimes          yes         0.0            Na
N
2              single     sometimes          yes         4.0     school_bu
s
3             married         never           no         1.0            Na
N
4             married     sometimes          yes         0.0     school_bu
s

  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 [12]:
```
df.head()
```

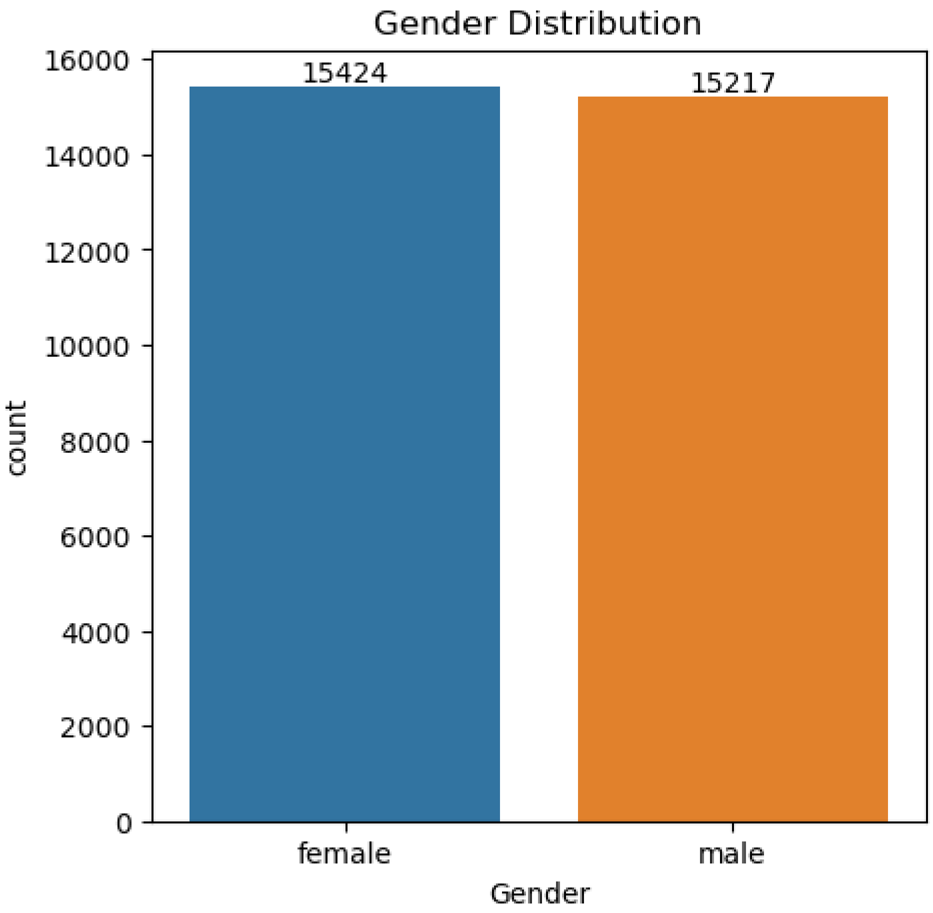
Out[12]:

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

Gender Distribution

```
In [31]: plt.figure(figsize=(5,5))
         ax=sns.countplot(data=df,x="Gender")
         ax.bar_label(ax.containers[0])
         plt.title("Gender Distribution")
         plt.show()
```

```
C:\Users\ASUS\anaconda3\lib\site-packages\seaborn\_core.py:1225: FutureWar
ning: is_categorical_dtype is deprecated and will be removed in a future v
ersion. Use isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
C:\Users\ASUS\anaconda3\lib\site-packages\seaborn\_core.py:1225: FutureWar
ning: is_categorical_dtype is deprecated and will be removed in a future v
ersion. Use isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
C:\Users\ASUS\anaconda3\lib\site-packages\seaborn\_core.py:1225: FutureWar
ning: is_categorical_dtype is deprecated and will be removed in a future v
ersion. Use isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
```



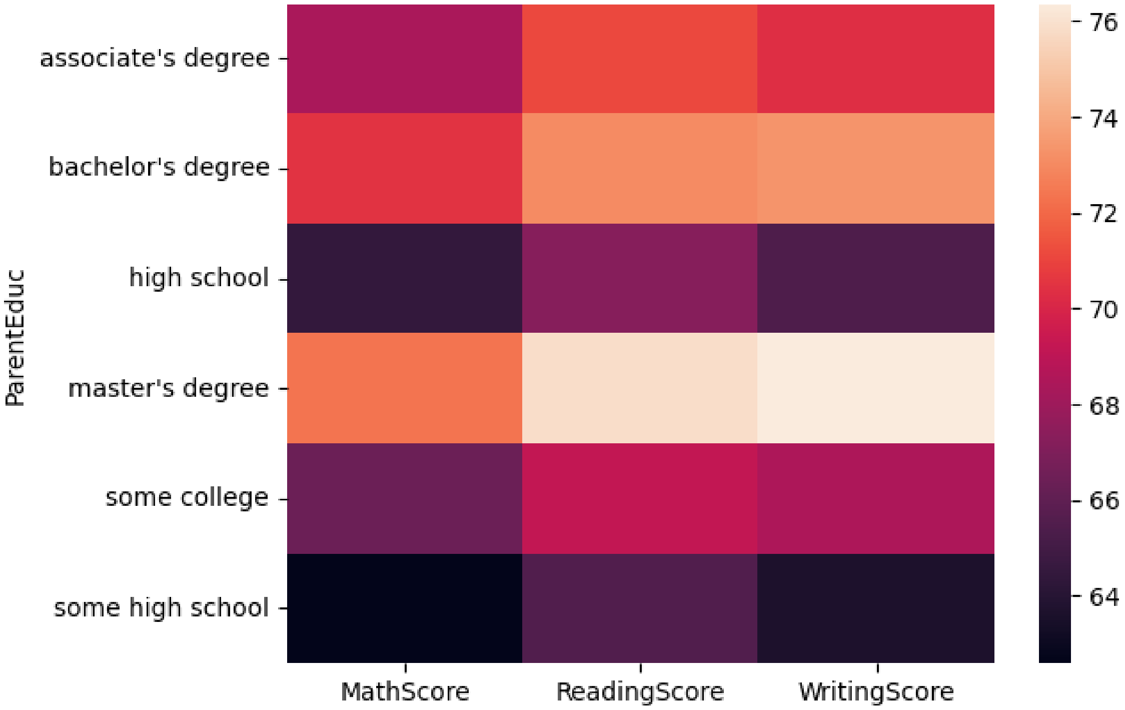
from above data we have to analysed that the number of females in the above data more then the number of males.

Now I wants to analysis that whats impact Parents Education on students education.

```
In [18]: gb=df.groupby("ParentEduc").agg({"MathScore":"mean","ReadingScore":"mean","W
         print(gb)
```

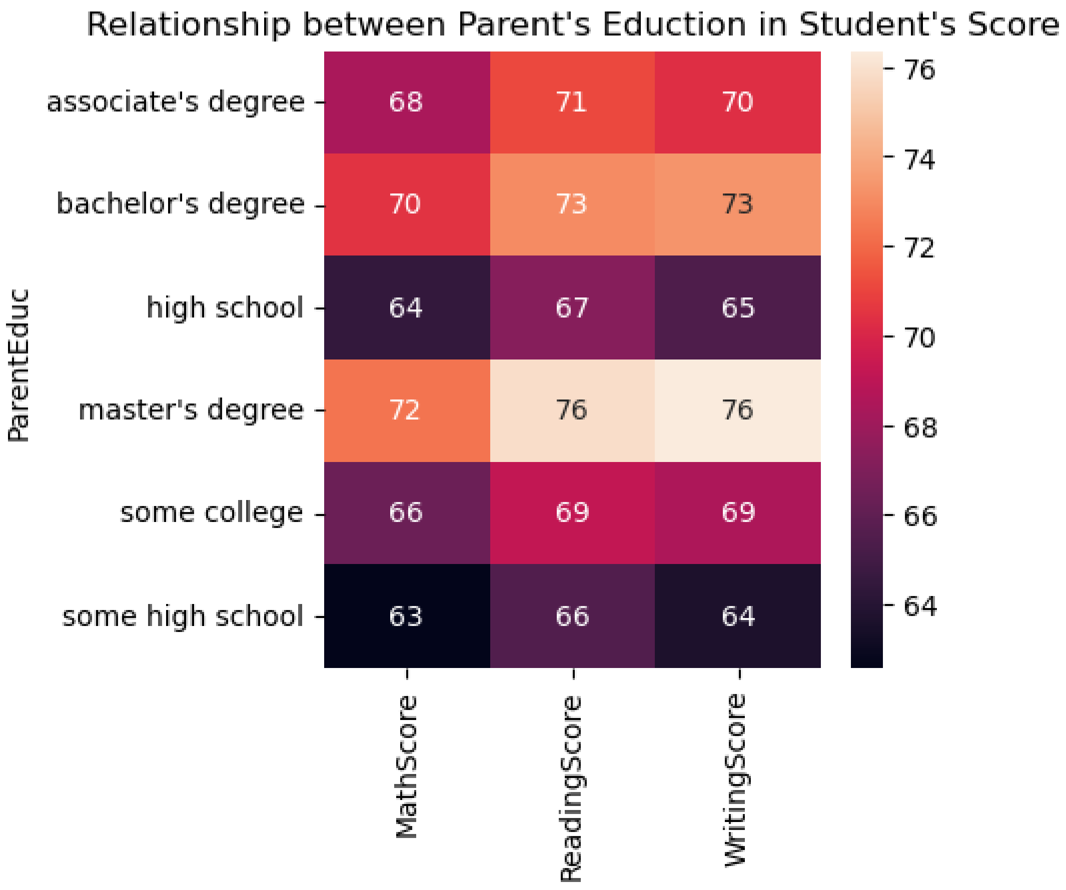
```
                    MathScore  ReadingScore  WritingScore
ParentEduc
associate's degree  68.365586     71.124324     70.299099
bachelor's degree   70.466627     73.062020     73.331069
high school         64.435731     67.213997     65.421136
master's degree     72.336134     75.832921     76.356896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

```
In [20]: sns.heatmap(gb)
         plt.show()
```



show the value in heat map

In [32]:
```
plt.figure(figsize=(4,4))
sns.heatmap(gb,annot=True)
plt.title("Relationship between Parent's Eduction in Student's Score")
plt.show()
```



from the above chart we have to concluded that Parents have a good impact on child edcation.

In [25]:
```
df.columns
```

Out[25]:
```
Index(['Gender', 'EthnicGroup', 'ParentEduc', 'LunchType', 'TestPrep',
       'ParentMaritalStatus', 'PracticeSport', 'IsFirstChild', 'NrSibling
s',
       'TransportMeans', 'WklyStudyHours', 'MathScore', 'ReadingScore',
       'WritingScore'],
      dtype='object')
```

Now check the Any impact on child of ParentMaritalStatus?

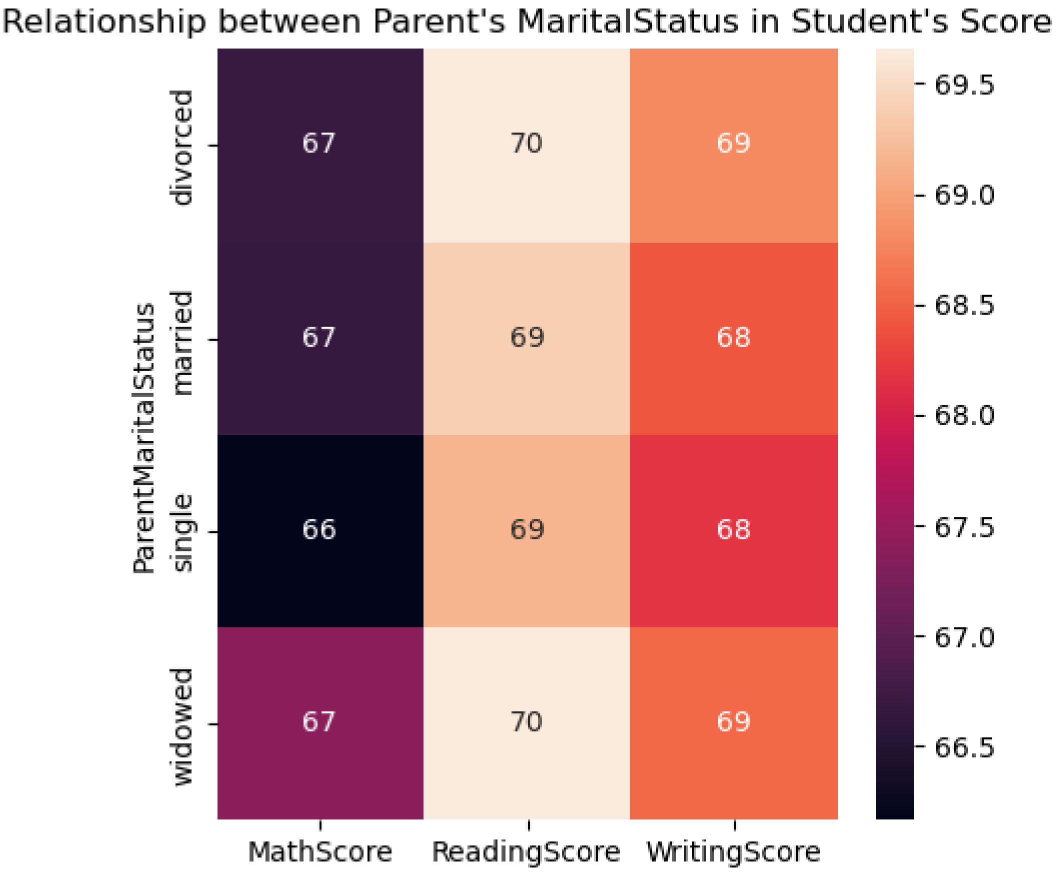
In [26]:

```
gp=df.groupby("ParentMaritalStatus").agg({"MathScore":"mean","ReadingScore":
print(gp)
```

```
                     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 [33]:

```
plt.figure(figsize=(5,5))
sns.heatmap(gp,annot=True)
plt.title("Relationship between Parent's MaritalStatus in Student's Score")
plt.show()
```

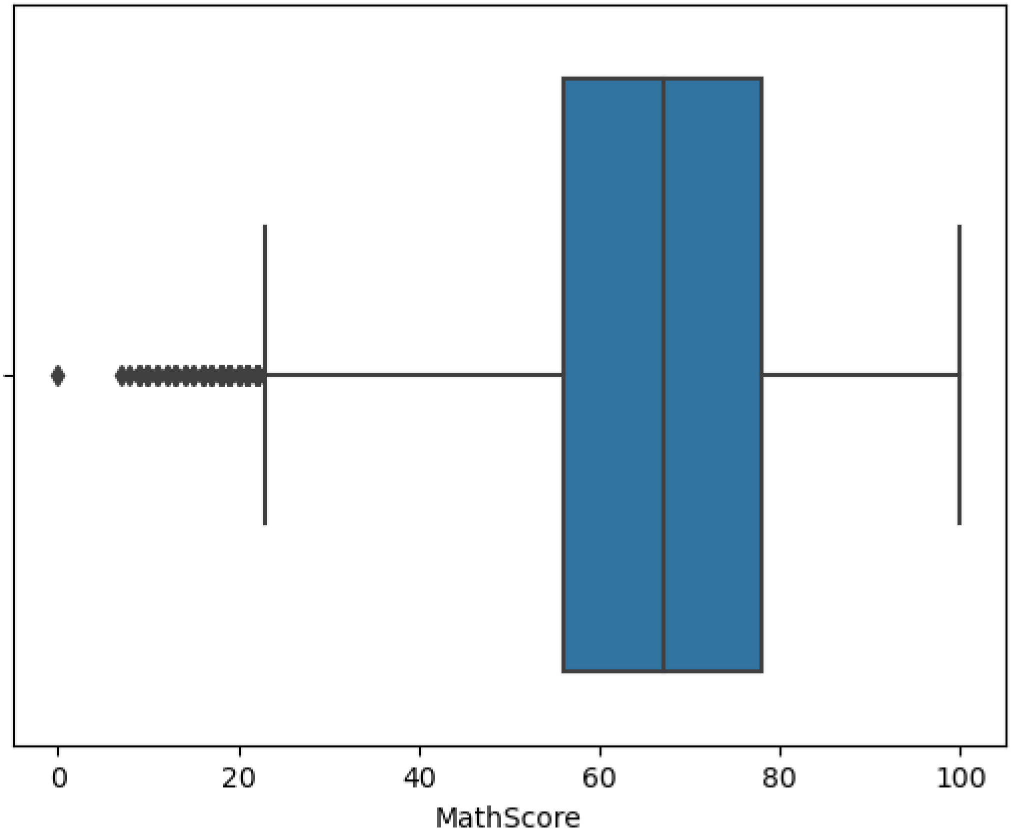


from the above chart we have to concluded that Parents have no more impact or neglagibel impact on child edcation

In [ ]:

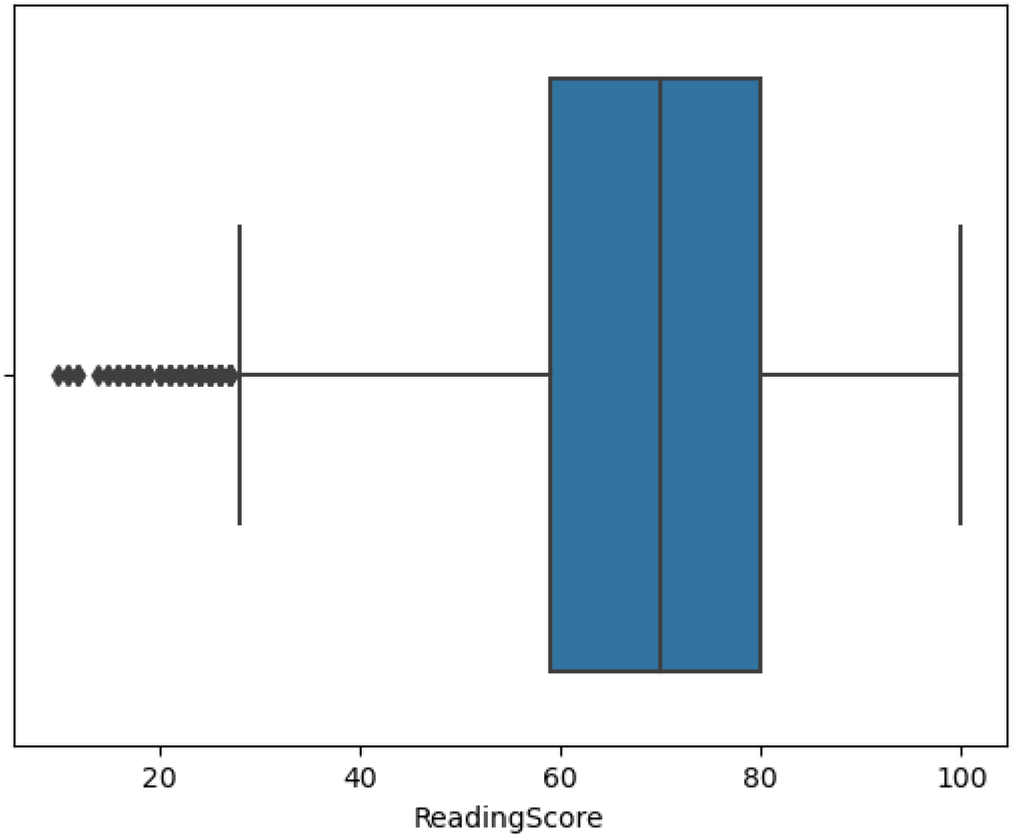
In [37]:
```
sns.boxplot(data=df,x="MathScore")
plt.show()
```

```
C:\Users\ASUS\anaconda3\lib\site-packages\seaborn\_core.py:1225: FutureWar
ning: is_categorical_dtype is deprecated and will be removed in a future v
ersion. Use isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
```



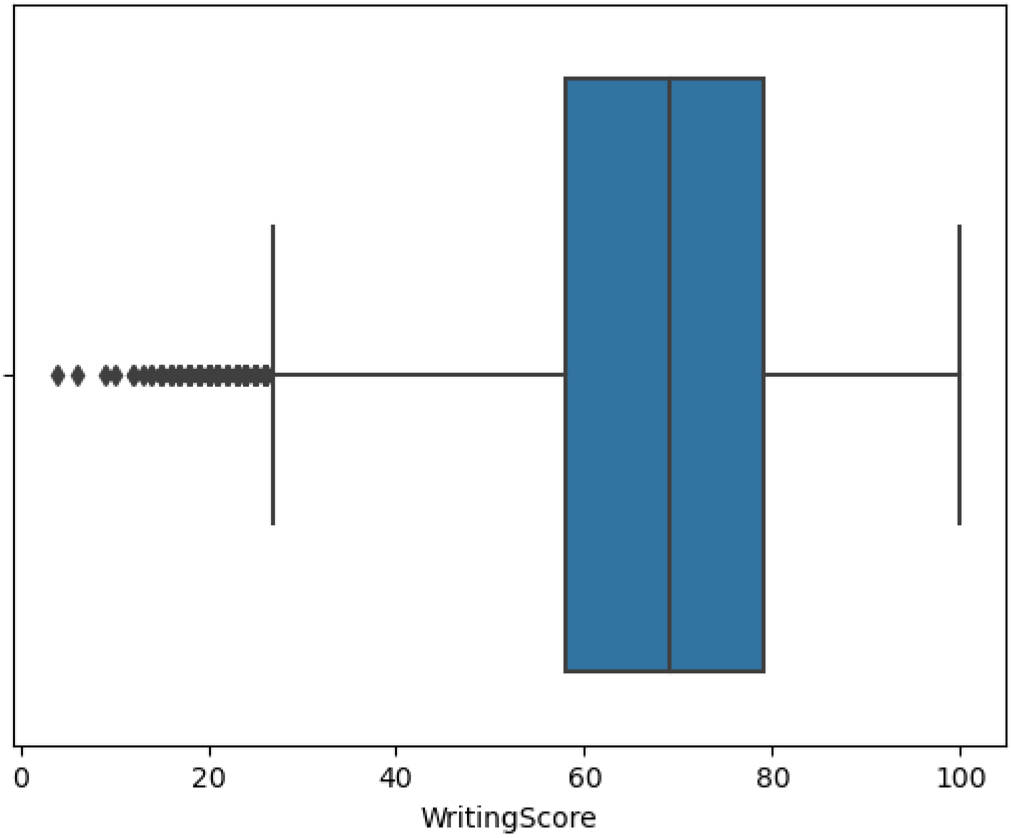
In [40]:
```
sns.boxplot(data=df,x="ReadingScore")
plt.show()
```

```
C:\Users\ASUS\anaconda3\lib\site-packages\seaborn\_core.py:1225: FutureWar
ning: is_categorical_dtype is deprecated and will be removed in a future v
ersion. Use isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
```



In [42]:
```
sns.boxplot(data=df,x="WritingScore")
plt.show()
```

```
C:\Users\ASUS\anaconda3\lib\site-packages\seaborn\_core.py:1225: FutureWar
ning: is_categorical_dtype is deprecated and will be removed in a future v
ersion. Use isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
```



In [43]:
```
df.columns
```

Out[43]:
```
Index(['Gender', 'EthnicGroup', 'ParentEduc', 'LunchType', 'TestPrep',
       'ParentMaritalStatus', 'PracticeSport', 'IsFirstChild', 'NrSibling
s',
       'TransportMeans', 'WklyStudyHours', 'MathScore', 'ReadingScore',
       'WritingScore'],
      dtype='object')
```

In [45]:
```
print(df["EthnicGroup"].unique())
```

```
[nan 'group C' 'group B' 'group A' 'group D' 'group E']
```

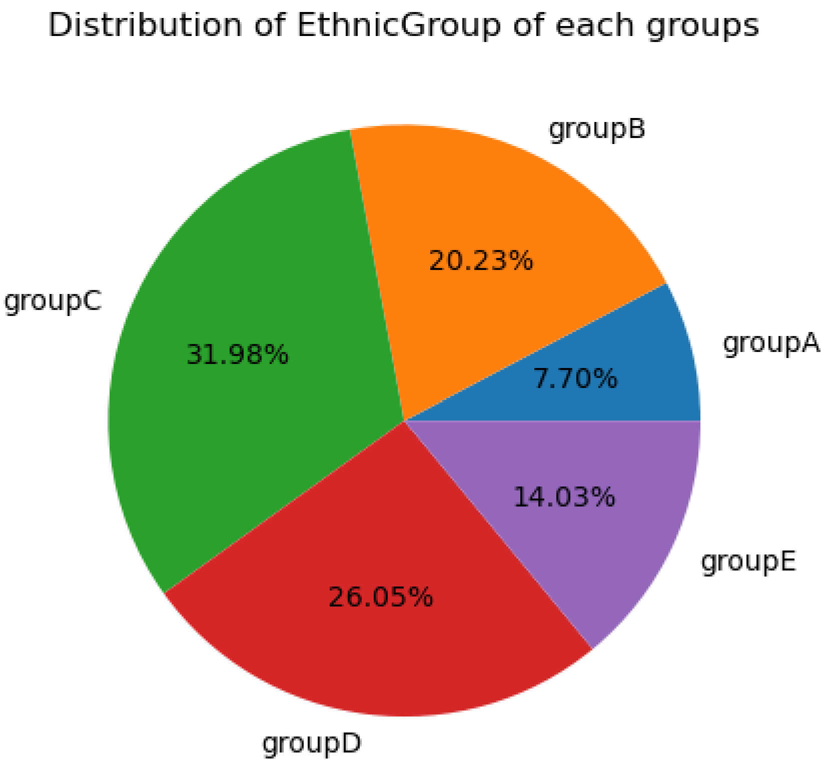
Distribuution of Ethinic groups

In [46]:

```
groupA=df.loc[(df["EthnicGroup"]== "group A")].count()
print(groupA)
```

```
Gender                 2219
EthnicGroup            2219
ParentEduc             2078
LunchType              2219
TestPrep               2081
ParentMaritalStatus    2121
PracticeSport          2167
IsFirstChild           2168
NrSiblings             2096
TransportMeans         1999
WklyStudyHours         2146
MathScore              2219
ReadingScore           2219
WritingScore           2219
dtype: int64
```

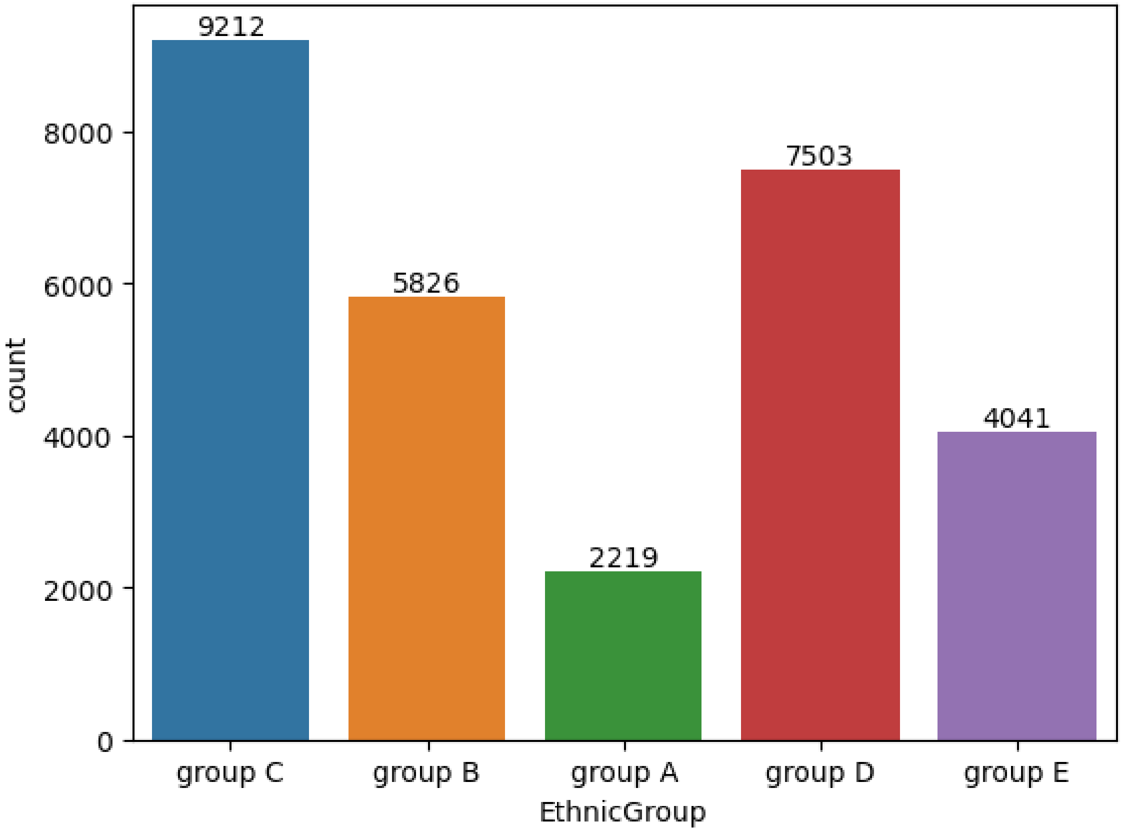
```
In [59]: #  count all value in all groups in dataframe

         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"],grc
         print(mlist)
         plt.pie(mlist,labels=l,autopct="%1.2f%%")
         plt.title("Distribution of EthnicGroup of each groups")

         [2219, 5826, 9212, 7503, 4041]

Out[59]: Text(0.5, 1.0, 'Distribution of EthnicGroup of each groups')
```



In [57]:
```
ax=sns.countplot(data=df,x="EthnicGroup")
ax.bar_label(ax.containers[0])
```

```
C:\Users\ASUS\anaconda3\lib\site-packages\seaborn\_core.py:1225: FutureWar
ning: is_categorical_dtype is deprecated and will be removed in a future v
ersion. Use isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
C:\Users\ASUS\anaconda3\lib\site-packages\seaborn\_core.py:1225: FutureWar
ning: is_categorical_dtype is deprecated and will be removed in a future v
ersion. Use isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
C:\Users\ASUS\anaconda3\lib\site-packages\seaborn\_core.py:1225: FutureWar
ning: is_categorical_dtype is deprecated and will be removed in a future v
ersion. Use isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
```

Out[57]:
```
[Text(0, 0, '9212'),
 Text(0, 0, '5826'),
 Text(0, 0, '2219'),
 Text(0, 0, '7503'),
 Text(0, 0, '4041')]
```



In [ ]: