STUDENT RESULT ANALYSIS

In [1]:

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

In [2]:

```
df = pd.read_csv(r'C:\Users\ADMIN\Desktop\Python Projects - Data Sci Lovers\student_scores.csv')
```

In [3]:

```
df.head()
```

Out[3]:

	Unnamed: 0	Gender	EthnicGroup	ParentEduc	LunchType	TestPrep	ParentMaritalStatus	PracticeSport	IsFirstChild	NrSiblings	Tra
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 [4]:

```
df.describe()
```

Out[4]:

	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 [5]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 30641 entries, 0 to 30640
Data columns (total 15 columns):
Unnamed: 0             30641 non-null int64
Gender                 30641 non-null object
EthnicGroup            28801 non-null object
ParentEduc             28796 non-null object
LunchType              30641 non-null object
TestPrep               28811 non-null object
ParentMaritalStatus    29451 non-null object
PracticeSport          30010 non-null object
IsFirstChild           29737 non-null object
NrSiblings             29069 non-null float64
TransportMeans         27507 non-null object
WklyStudyHours         29686 non-null object
MathScore              30641 non-null int64
ReadingScore           30641 non-null int64
WritingScore           30641 non-null int64
dtypes: float64(1), int64(4), object(10)
memory usage: 3.5+ MB
```

In [6]:

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

Out[6]:

```
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 [7]:

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

In [8]:

```
df.head()
```

Out[8]:

	Gender	EthnicGroup	ParentEduc	LunchType	TestPrep	ParentMaritalStatus	PracticeSport	IsFirstChild	NrSiblings	TransportMeans
0	female	NaN	bachelor's degree	standard	none	married	regularly	yes	3.0	school_bu
1	female	group C	some college	standard	NaN	married	sometimes	yes	0.0	Na
2	female	group B	master's degree	standard	none	single	sometimes	yes	4.0	school_bu
3	male	group A	associate's degree	free/reduced	none	married	never	no	1.0	Na
4	male	group C	some college	standard	none	married	sometimes	yes	0.0	school_bu

Gender Wise Distribution

In [9]:

```
df.groupby('Gender').sum()
```

Out[9]:

	NrSiblings	MathScore	ReadingScore	WritingScore
Gender				
female	31383.0	988380	1123688	1123738
male	30996.0	1051036	1002109	972677

In [10]:

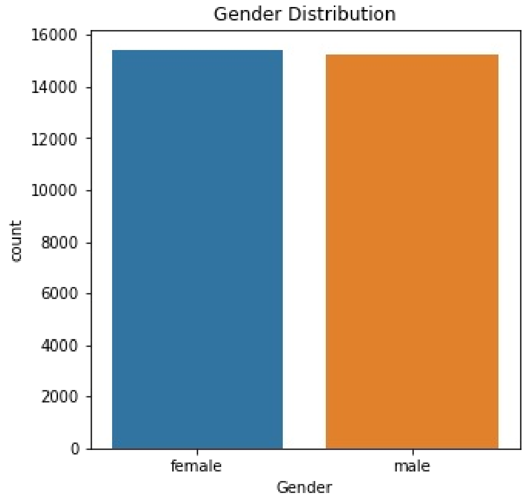
```
df['Gender'].value_counts()
```

Out[10]:

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

In [11]:

```
plt.figure(figsize=(5,5))
sns.countplot(data=df,x='Gender')
plt.title('Gender Distribution')
plt.show()
```



From the above, we can say that number of females in the data are more than the number of males

In [12]:

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

In [13]:

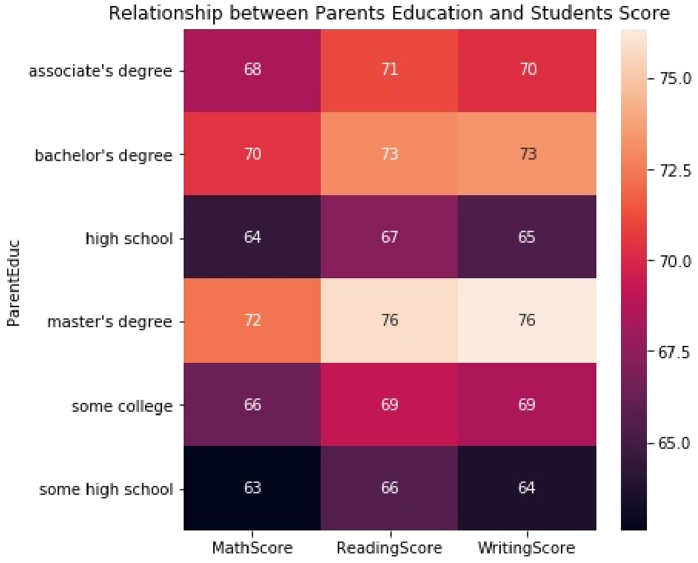
```
gb.head(2)
```

Out[13]:

	MathScore	ReadingScore	WritingScore
ParentEduc			
associate's degree	68.365586	71.124324	70.299099
bachelor's degree	70.466627	73.062020	73.331069

In [14]:

```
plt.figure(figsize=(6,6))
sns.heatmap(gb,annot=True)
plt.title('Relationship between Parents Education and Students Score')
plt.show()
```



From the above chart, we can conclude that education of the parents make good impact on their marks

In [15]:

```
df.head()
```

Out[15]:

	Gender	EthnicGroup	ParentEduc	LunchType	TestPrep	ParentMaritalStatus	PracticeSport	IsFirstChild	NrSiblings	TransportMeans
0	female	NaN	bachelor's degree	standard	none	married	regularly	yes	3.0	school_bu
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_bu
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_bu

In [16]:

```
gb1=df.groupby('ParentMaritalStatus').agg({'MathScore':'mean','ReadingScore':'mean','WritingScore':'mean'})
```

In [17]:

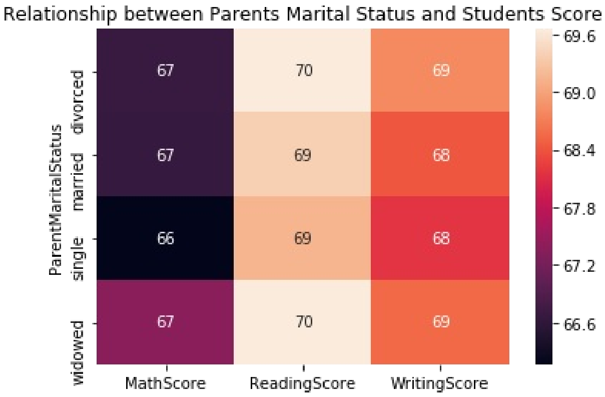
```
gb1
```

Out[17]:

	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 [18]:

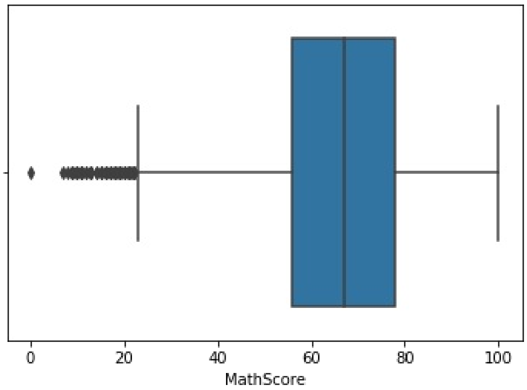
```
sns.heatmap(gb1,annot=True)
plt.title('Relationship between Parents Marital Status and Students Score')
plt.show()
```



From the above chart we have concluded that there is no / negligible impact on the students score due to their parents marital status

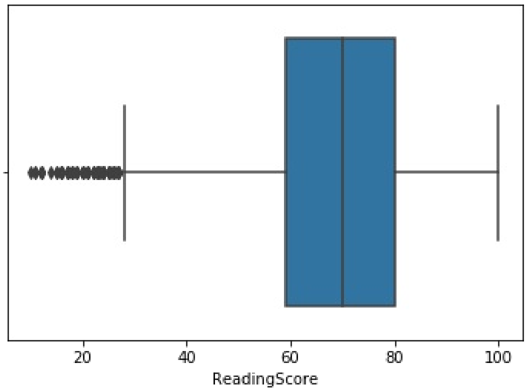
In [19]:

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



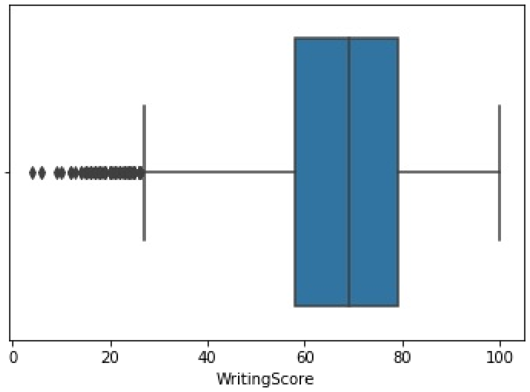
In [20]:

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



In [21]:

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



In [22]:

```
print(df['EthnicGroup'].unique())
```

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

Distribution of Ethnic Group

In [23]:

```
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()
```

In [24]:

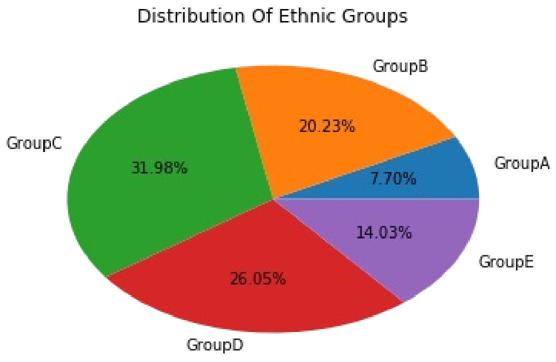
```
print(groupA['EthnicGroup'])
```

```
2219
```

In [25]:

```
labels = ['GroupA','GroupB','GroupC','GroupD','GroupE']
mlist = [groupA['EthnicGroup'],groupB['EthnicGroup'],groupC['EthnicGroup'],groupD['EthnicGroup'],groupE['EthnicGr
oup']]

plt.pie(mlist,labels=labels,autopct="%1.2f%%")
plt.title('Distribution Of Ethnic Groups')
plt.show()
```



In [ ]:

```

```