In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

In [7]:
```
df=pd.read_csv(r"C:\Users\Arpit jain\Downloads\student  exam score python project\student_exam_score.csv" )
```

In [9]:
```
print(df.head(5))
```

```
   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
```

In [10]:
```
df.describe()
```

Out[10]:

	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 [11]:
```
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 [13]:
```
df.isnull().sum()
```

Out[13]:
```
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 [15]:
```
df=df.drop("Unnamed: 0",axis=1)
```

In [17]:
```
print(df.head(2))
```

```
   Gender EthnicGroup         ParentEduc LunchType TestPrep  \
0  female         NaN  bachelor's degree  standard     none
1  female     group C       some college  standard      NaN

  ParentMaritalStatus PracticeSport IsFirstChild  NrSiblings TransportMeans  \
0             married     regularly          yes         3.0     school_bus
1             married     sometimes          yes         0.0            NaN

  WklyStudyHours  MathScore  ReadingScore  WritingScore
0            < 5         71            71            74
1        5 - 10         69            90            88
```

change weekly study hours column

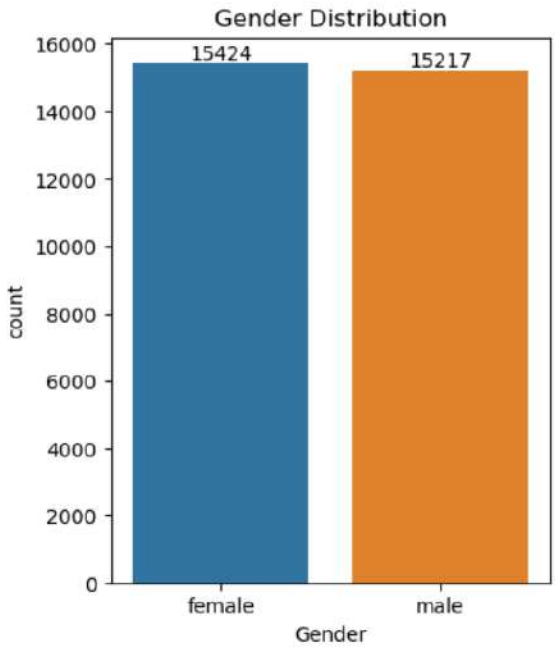
In [24]:
```
df["WklyStudyHours"]=df["WklyStudyHours"].str.replace("5-oct","5-10")
df.head()
```

Out[24]:

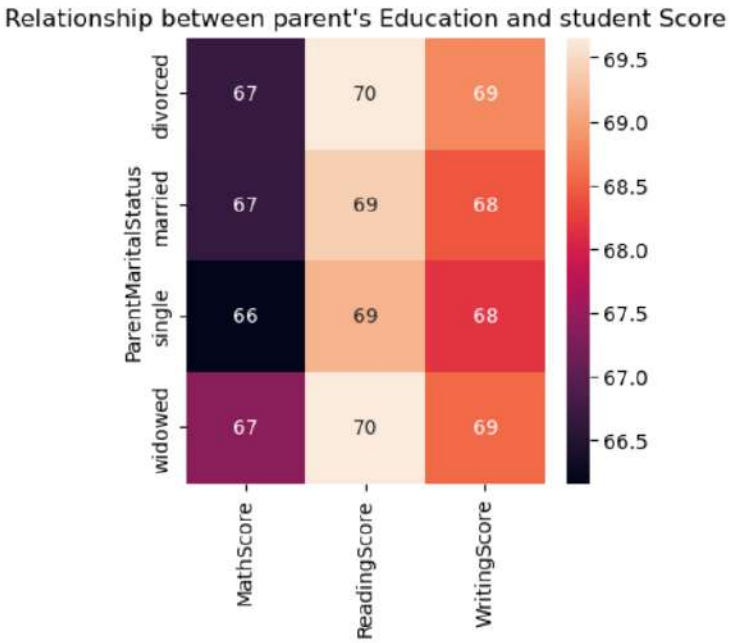
	Gender	EthnicGroup	ParentEduc	LunchType	TestPrep	ParentMaritalStatus	PracticeSport	IsFirstChild	NrSiblings	TransportMeans	WklyStudyHours
0	female	NaN	bachelor's degree	standard	none	married	regularly	yes	3.0	school_bus	< 5
1	female	group C	some college	standard	NaN	married	sometimes	yes	0.0	NaN	5-10
2	female	group B	master's degree	standard	none	single	sometimes	yes	4.0	school_bus	< 5
3	male	group A	associate's degree	free/reduced	none	married	never	no	1.0	NaN	5-10
4	male	group C	some college	standard	none	married	sometimes	yes	0.0	school_bus	5-10

Gender Distribution

In [8]:
```
plt.figure(figsize=(4,5))
ax=sns.countplot(data=df,x="Gender")
ax.bar_label(ax.containers[0])
plt.title("Gender Distribution")
plt.show()
```

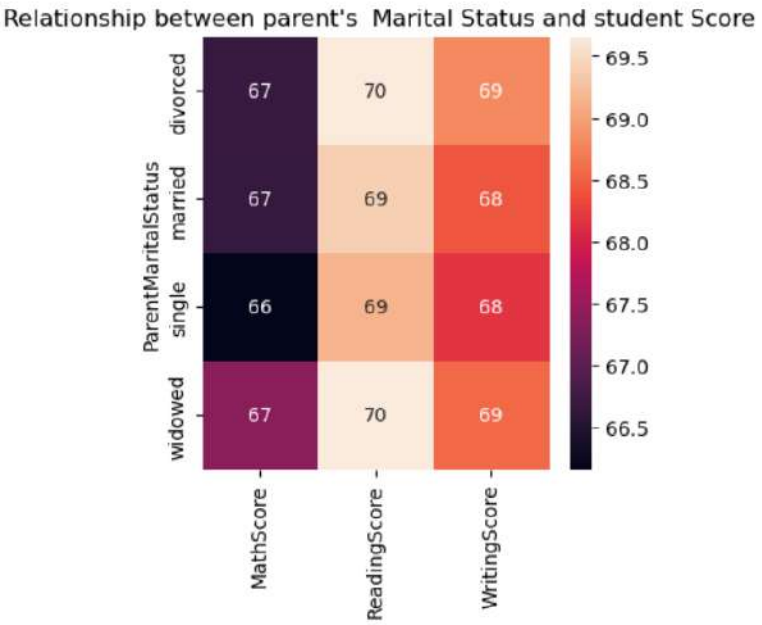


In [25]:
```
plt.figure(figsize=(4,4))
sns.heatmap(gp,annot=True)
plt.title(" Relationship between parent's Education and student Score ")
plt.show()
```



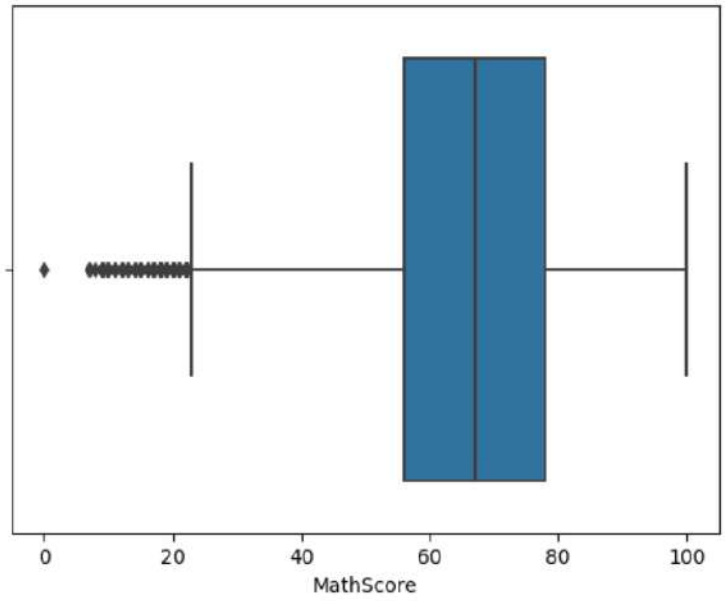
In [17]:
```
# from the above chart we have conculed that the education of the parents
# have a good  impact on the student
```

In [26]:
```
plt.figure(figsize=(4,4))
sns.heatmap(gp,annot=True)
plt.title(" Relationship between parent's  Marital Status and student Score")
plt.show()
```

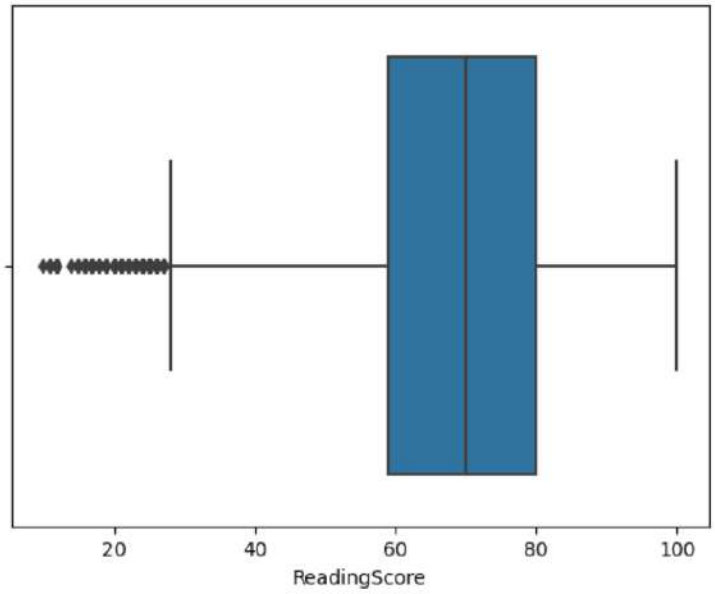


In [ ]:
```
# Above charts thier is no/negligible  impact on the
# student education based on parents marital status
```

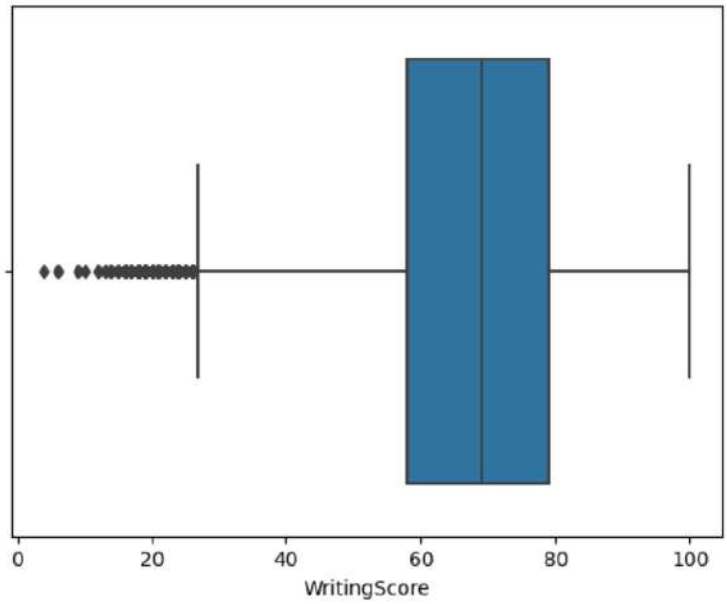
In [27]:
```
sns.boxplot(data=df,x='MathScore')
plt.show()
```



In [28]:
```
sns.boxplot(data=df,x='ReadingScore')
plt.show()
```



In [29]:
```
sns.boxplot(data=df,x='WritingScore')
plt.show()
```



In [31]:
```
print(df["EthnicGroup"].unique())
```

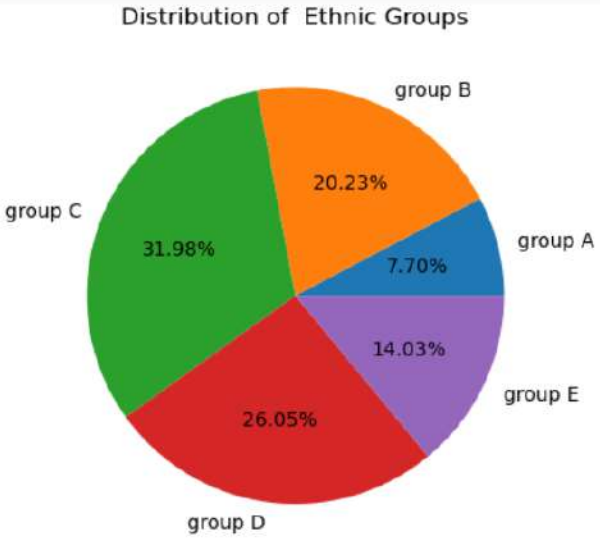
```
[nan 'group C' 'group B' 'group A' 'group D' 'group E']
```

Distribution of Ethnic Groups

In [41]:
```
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()
mylist=[groupA["EthnicGroup"],groupB["EthnicGroup"],groupC["EthnicGroup"],
        groupD["EthnicGroup"],groupE["EthnicGroup"]]
print(mylist)
l=["group A","group B","group C","group D","group E"]

plt.pie(mylist,labels=l,autopct="%1.2f%%")
plt.title("Distribution of  Ethnic Groups")
plt.show()
```

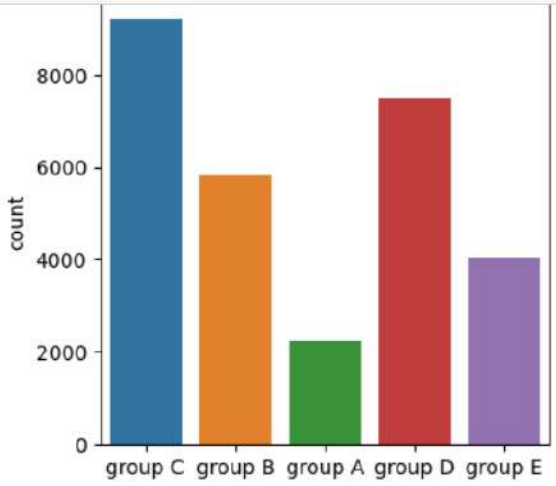
```
[2219, 5826, 9212, 7503, 4041]
```


Distribution of Ethnic Groups
group B
20.23%
group C
31.98%
group A
7.70%
14.03%
group E
26.05%
group D

In [10]:

```
plt.figure(figsize=(4,4))

sns.countplot(data=df,x='EthnicGroup')
ax.bar_label(ax.containers[0])
```



conclusion

Overall student performance in each exam is good, especially in reading and writing exams.

Student who gets high scores in math is likely to get high score in reading and writing score, and vice versa.

Gender and ethnic group affect student performance in each exam.

Female student seems to have a good performance in writing and reading, while male student has a good performance in math.

Parent education level also affects student performance, the higher the level of education, the higher the exam score.

Student who comes from ethnic group E seems likely to have a good performance in all exam regardless of their parent's education level and even though their group is the minority.

Student who completed the test preparation course seems to have better performance in all of the exams compared to those who don’t.

Student who spends more than 10 hours weekly studying seems to have a good performance in the math exam.