Students result analysis.

January 19, 2024

[18]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

[2]:
```
df= pd.read_csv("Expanded_data_with_more_features.csv")
print(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
```

[3]:
```
df.describe()
```

[3]:

	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
```

[4]:
```
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
```

[5]:
```
df.isnull().sum()
```

[5]:
```
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
```

0.1 Drop Unnamed: 0

[6]:
```
df= df.drop("Unnamed: 0",axis=1)
```

[7]:
```
df.head()
```

[7]:
```
   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
```

1 Change WklyStudyHours column

[8]:
```
df["WklyStudyHours"].unique()
```

[8]:
```
array(['< 5', '5 - 10', '> 10', nan], dtype=object)
```

[9]:
```
df["WklyStudyHours"] = df["WklyStudyHours"].str.replace("> 10","5 - 10")
```

[10]:
```
df["WklyStudyHours"].unique()
```

[10]:
```
array(['< 5', '5 - 10', nan], dtype=object)
```

[11]:
```
df["WklyStudyHours"].value_counts()
```

[11]:
```
5 - 10    21448
< 5        8238
Name: WklyStudyHours, dtype: int64
```

[12]:
```
df.dropna(subset=['WklyStudyHours'], inplace=True)
```

[13]:
```
df["WklyStudyHours"].unique()
```

[13]:
```
array(['< 5', '5 - 10'], dtype=object)
```

[14]:
```
df.isnull().sum()
```

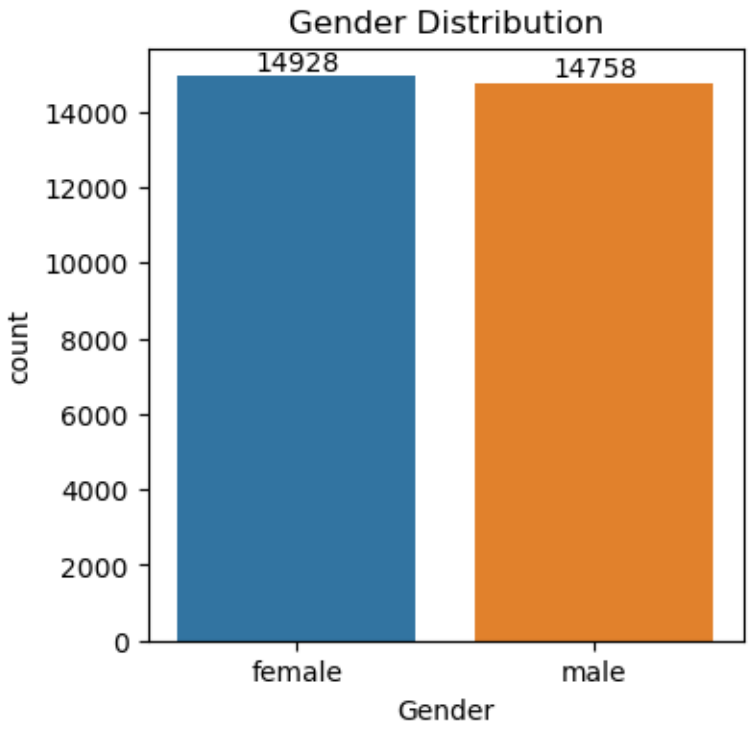
[14]:
```
Gender                   0
EthnicGroup           1771
ParentEduc            1783
LunchType                0
TestPrep              1779
ParentMaritalStatus   1156
PracticeSport          611
IsFirstChild           881
NrSiblings            1527
TransportMeans        3044
WklyStudyHours           0
MathScore                0
ReadingScore             0
WritingScore             0
dtype: int64
```

[15]:
```
dataset_length = len(df)
print("Length of the dataset:", dataset_length)
```

```
Length of the dataset: 29686
```

2 Gender Distribution

[40]:
```
plt.figure(figsize=(4,4))
ax=sns.countplot(data=df,x="Gender")
plt.title("Gender Distribution")
ax.bar_label(ax.containers[0])
plt.show()
```



2.0.1 From the above analysis we found that in our data number of female is more than number of male.

3 Parent Education

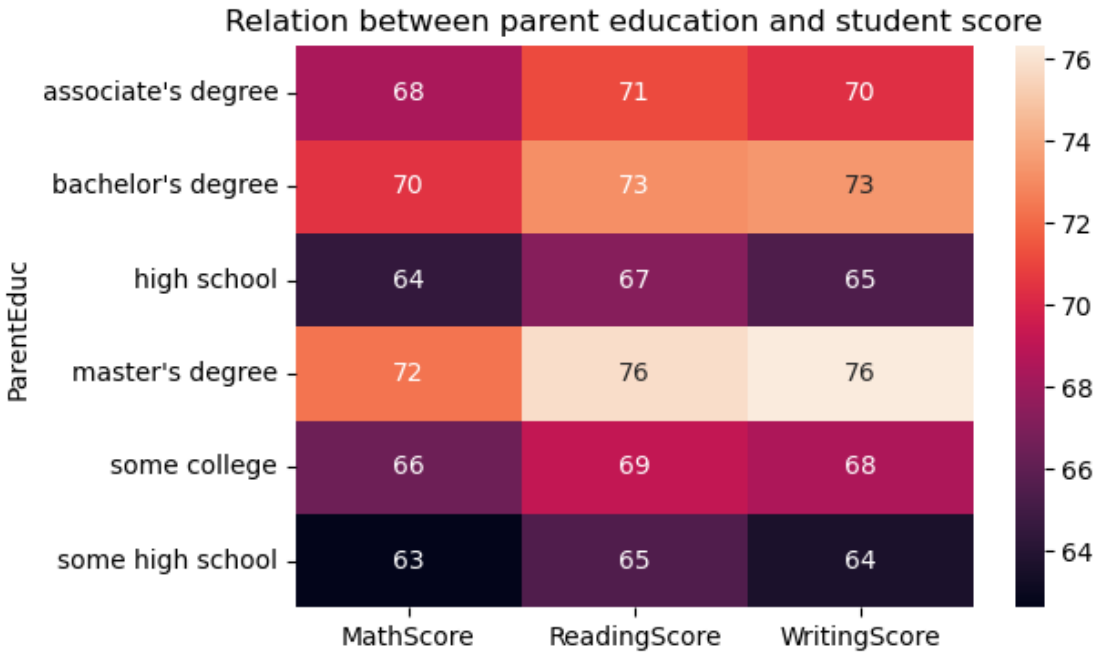
[29]:
```
df['ParentEduc'].unique()
```

[29]:
```
array(["bachelor's degree", 'some college', "master's degree",
       "associate's degree", 'high school', 'some high school', nan],
      dtype=object)
```

[31]:
```
gb=df.groupby("ParentEduc").agg({"MathScore":"mean","ReadingScore":
  ↪"mean","WritingScore":"mean"})
print(gb)
```

```
                    MathScore  ReadingScore  WritingScore
ParentEduc
associate's degree  68.365124     71.168061     70.344302
bachelor's degree   70.444478     73.119608     73.392781
high school         64.453752     67.214792     65.402689
master's degree     72.307067     75.813930     76.320793
some college        66.413536     69.175256     68.497672
some high school    62.605278     65.483249     63.591615
```

[41]:
```
plt.figure(figsize=(6,4))
sns.heatmap(gb,annot=True)
plt.title("Relation between parent education and student score")
plt.show()
```



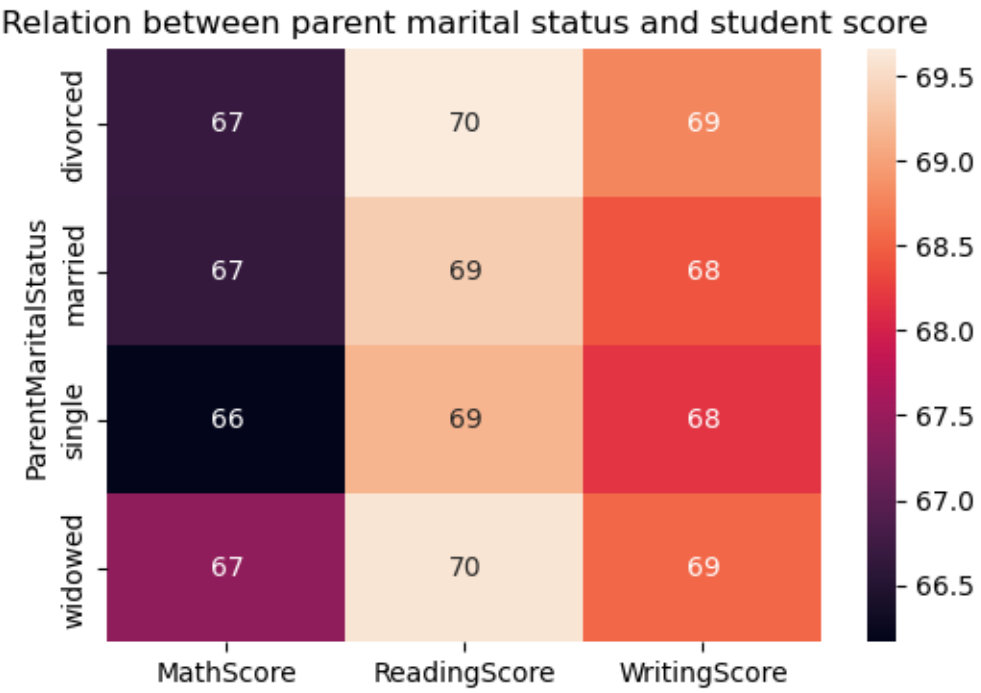
3.0.1 from the above Chart we concude that education of parents have good impact on their kids score.

4 Parent maratial status

[42]:
```
gb_marital_status=df.groupby('ParentMaritalStatus').agg({"MathScore":
  ↪"mean","ReadingScore":"mean","WritingScore":"mean"})
print(gb_marital_status)
```

```
                    MathScore  ReadingScore  WritingScore
ParentMaritalStatus
divorced            66.677948     69.657137     68.782846
married             66.646072     69.377013     68.403650
single              66.163028     69.162154     68.177001
widowed             67.419580     69.601399     68.541958
```

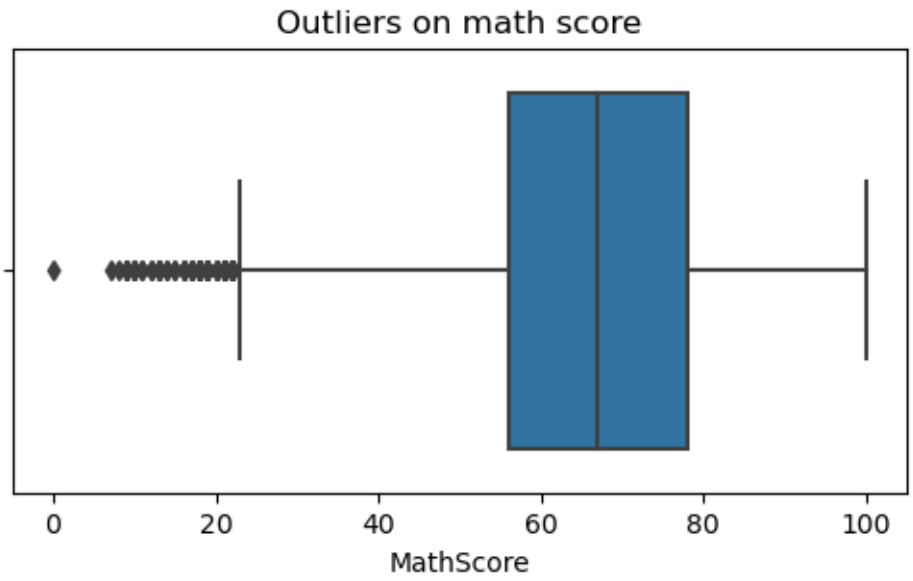
[43]:
```
plt.figure(figsize=(6,4))
sns.heatmap(gb_marital_status,annot=True)
plt.title("Relation between parent marital status and student score")
plt.show()
```



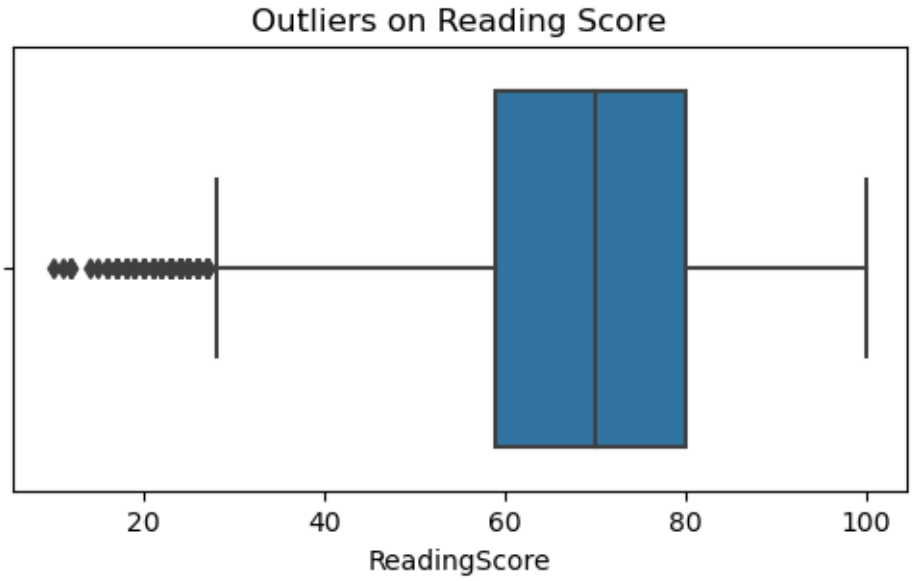
4.0.1 from the above Chart we concude that marital status of parents have no impact on their kids score.

5 Finding outliers

[48]:
```
plt.figure(figsize=(6,3))
sns.boxplot(data=df,x=('MathScore'))
plt.title("Outliers on math score")
plt.show()
```

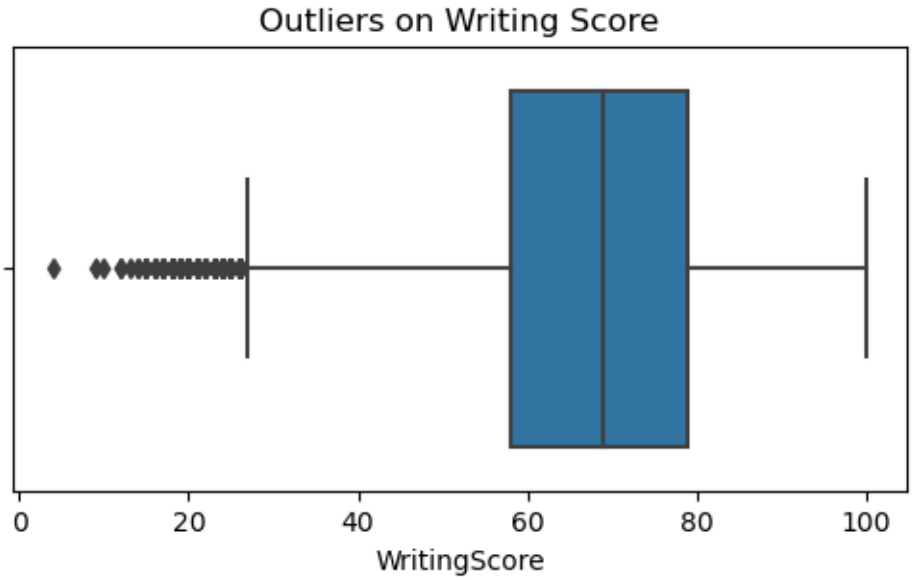


[49]:
```
plt.figure(figsize=(6,3))
sns.boxplot(data=df,x=('ReadingScore'))
plt.title("Outliers on Reading Score")
plt.show()
```



[50]:
```
plt.figure(figsize=(6,3))
sns.boxplot(data=df,x=('WritingScore'))
```

```
plt.title("Outliers on Writing Score")
plt.show()
```



5.0.1 From the above three boxplot we conclute that students face more difficulty in math as compair to other two.

6 Distribution of Ethnic Group

[51]:
```
df['EthnicGroup'].unique()
```

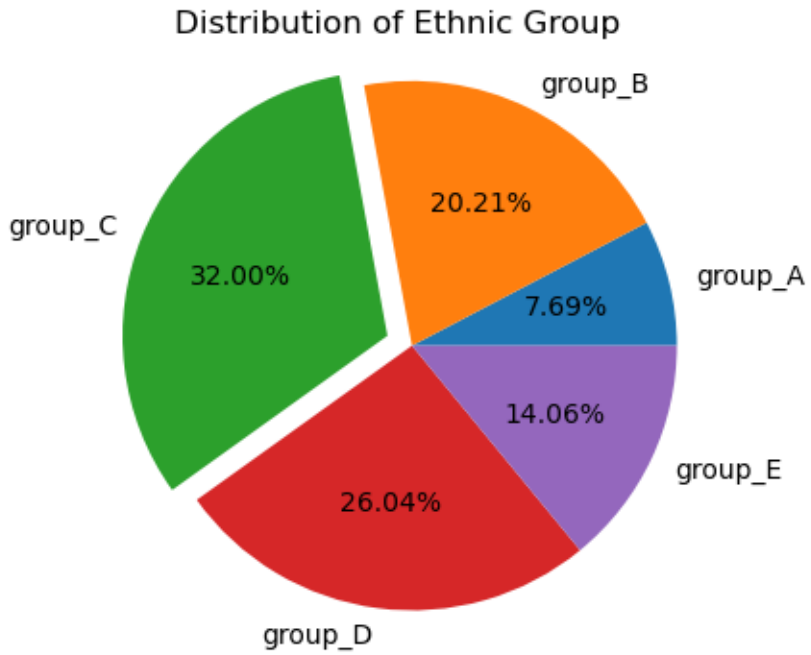
[51]:
```
array([nan, 'group C', 'group B', 'group A', 'group D', 'group E'],
      dtype=object)
```

[61]:
```
group_A =df.loc[(df['EthnicGroup']=="group A")].count()
group_B=df.loc[(df['EthnicGroup']=="group B")].count()
group_C=df.loc[(df['EthnicGroup']=="group C")].count()
group_D=df.loc[(df['EthnicGroup']=="group D")].count()
group_E=df.loc[(df['EthnicGroup']=="group E")].count()
```

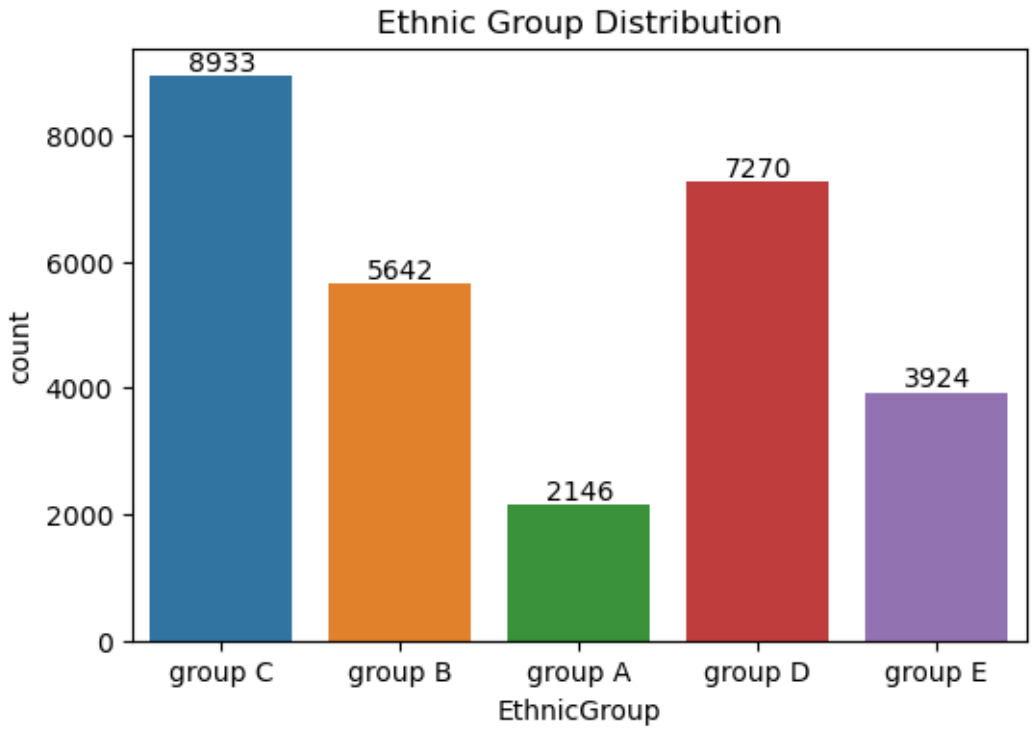
[65]:
```
l=['group_A','group_B','group_C','group_D','group_E']
myList=[group_A['EthnicGroup'],group_B['EthnicGroup'],group_C['EthnicGroup'],group_D['EthnicGr
```

[73]:
```
plt.figure(figsize=(4,4))
explode = (0, 0, 0.1, 0, 0)
plt.pie(myList,labels=l,autopct="%1.2f%%",explode=explode)
plt.title("Distribution of Ethnic Group")
plt.axis('equal')
```

```
plt.show()
```



[75]:
```
plt.figure(figsize=(6,4))
ax=sns.countplot(data=df,x="EthnicGroup")
plt.title("Ethnic Group Distribution")
ax.bar_label(ax.containers[0])
plt.show()
```



[76]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 29686 entries, 0 to 30640
Data columns (total 14 columns):
 #   Column               Non-Null Count  Dtype  
---  ------               --------------  -----  
 0   Gender               29686 non-null  object 
 1   EthnicGroup          27915 non-null  object 
 2   ParentEduc           27903 non-null  object 
 3   LunchType            29686 non-null  object 
 4   TestPrep             27907 non-null  object 
 5   ParentMaritalStatus  28530 non-null  object 
 6   PracticeSport        29075 non-null  object 
 7   IsFirstChild         28805 non-null  object 
 8   NrSiblings           28159 non-null  float64
 9   TransportMeans       26642 non-null  object 
 10  WklyStudyHours       29686 non-null  object 
 11  MathScore            29686 non-null  int64  
 12  ReadingScore         29686 non-null  int64  
 13  WritingScore         29686 non-null  int64  
dtypes: float64(1), int64(3), object(10)
memory usage: 3.4+ MB
```

7 On the basic of PracticeSport

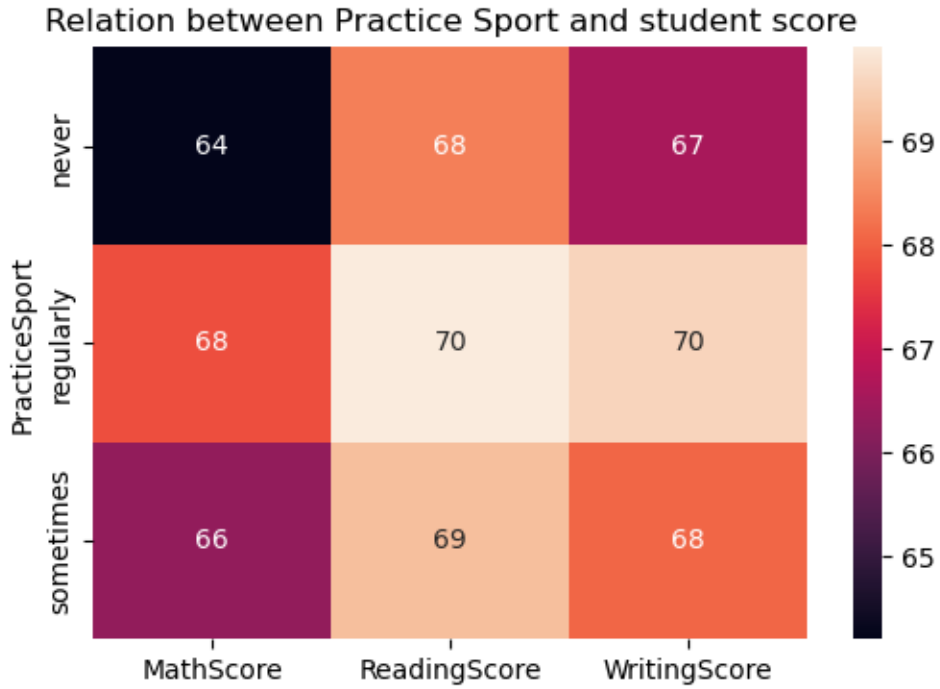
[77]:
```
df['PracticeSport'].unique()
```

[77]:
```
array(['regularly', 'sometimes', 'never', nan], dtype=object)
```

[78]:
```
gb_PracticeSport=df.groupby('PracticeSport').agg({"MathScore":
  ↪"mean","ReadingScore":"mean","WritingScore":"mean"})
print(gb_PracticeSport)
```

```
               MathScore  ReadingScore  WritingScore
PracticeSport
never          64.205959     68.387565     66.566839
regularly      67.799255     69.900869     69.557253
sometimes      66.282488     69.249457     68.069995
```

[79]:
```
plt.figure(figsize=(6,4))
sns.heatmap(gb_PracticeSport,annot=True)
plt.title("Relation between Practice Sport and student score")
plt.show()
```



7.0.1 from the above Chart we concude that student who involve in sport and have not involve in sport are almost score same. only few marks difference in math who never active in sports.

8 On the basic of test prepration

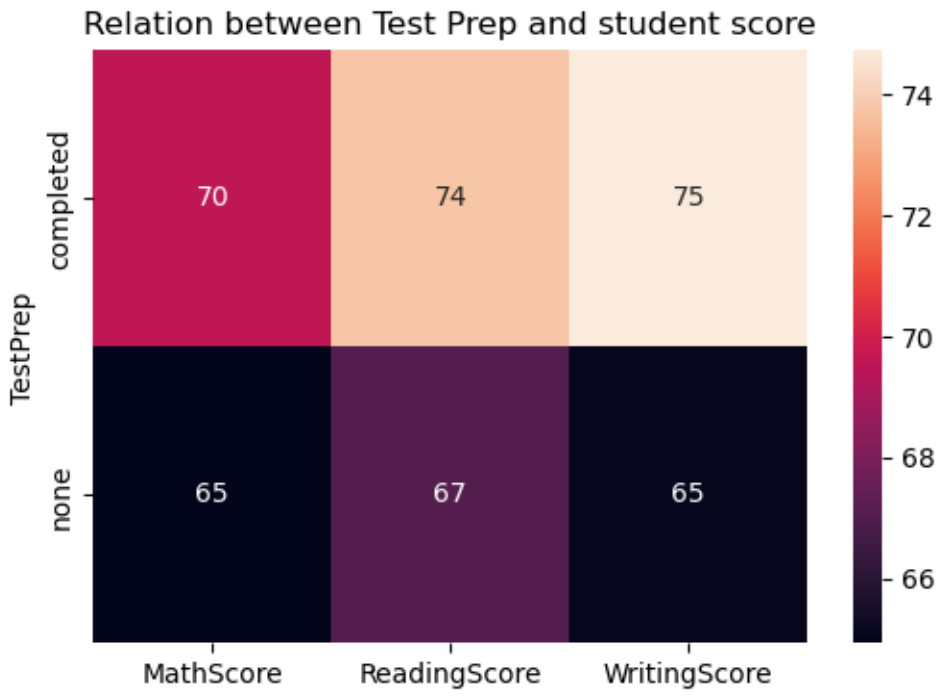
[81]:
```
df['TestPrep'].unique()
```

[81]:
```
array(['none', nan, 'completed'], dtype=object)
```

[82]:
```
gb_TestPrep=df.groupby('TestPrep').agg({"MathScore":"mean","ReadingScore":
  ↪"mean","WritingScore":"mean"})
print(gb_TestPrep)
```

```
           MathScore  ReadingScore  WritingScore
TestPrep
completed  69.551419     73.740668     74.720287
none       64.941443     67.042865     65.074959
```

[86]:
```
plt.figure(figsize=(6,4))
sns.heatmap(gb_TestPrep,annot=True)
plt.title("Relation between Test Prep and student score")
plt.show()
```



8.0.1 From the above analysis we say that students who prepare for test they got high score in the test

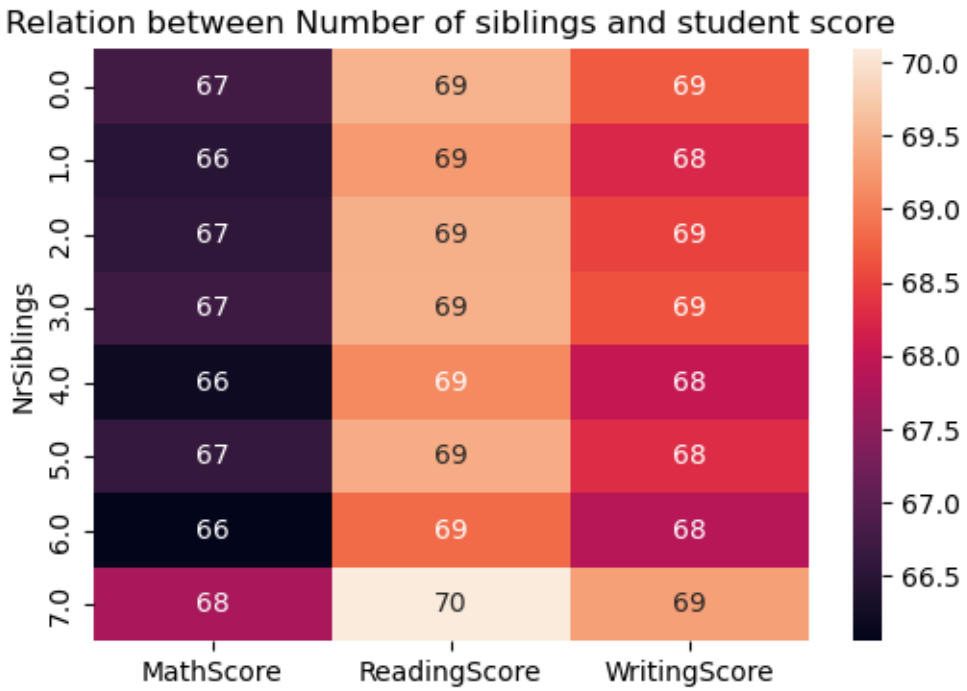
[84]:
```
df['NrSiblings'].unique()
```

[84]:
```
array([ 3.,  0.,  4.,  1., nan,  2.,  5.,  7.,  6.])
```

[85]:
```
gb_NrSib=df.groupby(df['NrSiblings']).agg({"MathScore":"mean","ReadingScore":
  ↪"mean","WritingScore":"mean"})
print(gb_NrSib)
```

```
             MathScore  ReadingScore  WritingScore
NrSiblings
0.0          66.762317     69.495672     68.696405
1.0          66.484680     69.257723     68.237927
2.0          66.549106     69.463171     68.504395
3.0          66.709775     69.484383     68.627216
4.0          66.196209     69.106002     68.022464
5.0          66.611371     69.438474     68.297508
6.0          66.055749     68.832753     67.881533
7.0          67.738516     70.088339     69.303887
```

[91]:
```
plt.figure(figsize=(6,4))
sns.heatmap(gb_NrSib,annot=True)
plt.title("Relation between Number of siblings and student score")
plt.show()
```



8.0.2 As we can see that there are ngligible/no impact of numbers of sibling on student score.

9 Analysis on Transport system

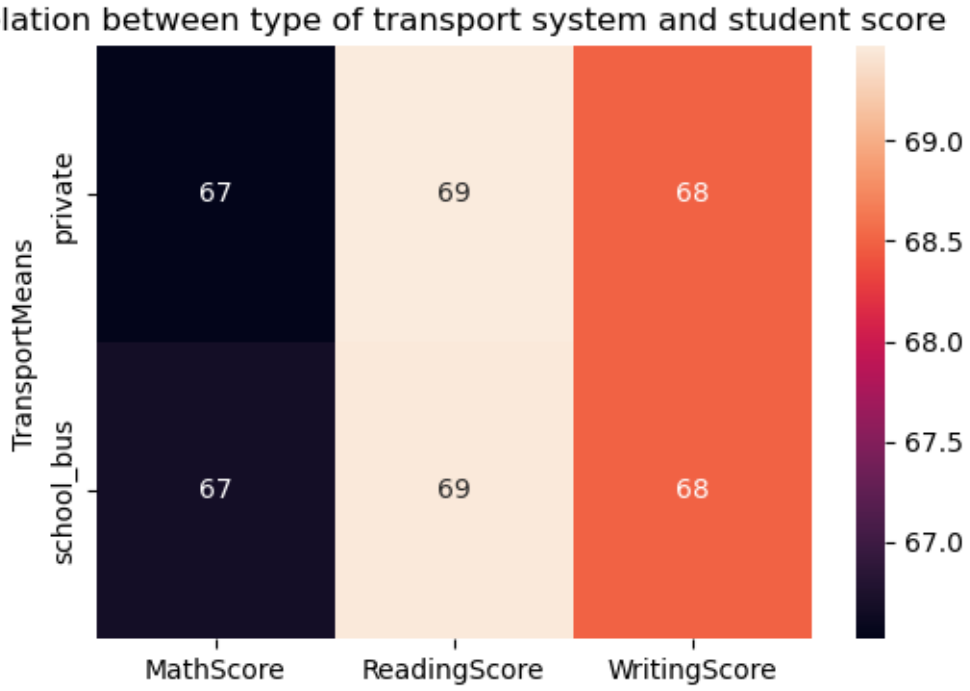
[88]:
```
df['TransportMeans'].unique()
```

[88]:
```
array(['school_bus', nan, 'private'], dtype=object)
```

[89]:
```
gb_trans=df.groupby(df['TransportMeans']).agg({"MathScore":
  ↪"mean","ReadingScore":"mean","WritingScore":"mean"})
print(gb_trans)
```

```
                MathScore  ReadingScore  WritingScore
TransportMeans
private         66.516931     69.468232     68.498726
school_bus      66.673097     69.450115     68.494124
```

[90]:
```
plt.figure(figsize=(6,4))
sns.heatmap(gb_trans,annot=True)
plt.title("Relation between type of transport system and student score")
plt.show()
```



9.0.1 As we can see that there are ngligible/no impact of transport systm on student score.

10 ————————————————————————————

 # From the whole analysis final conclusion is in our data we have more female than male.

 # Education of partent are import for their child good score.

 # Parents marital status are not import for good score of students.

 # Math score has more outliers. ie it is more diffcult than others.

 # Largest Ethnic group is Group_C whis had 8933 students.

 # Students who play sport and have not play sport are almost score same. only few marks difference in math who never active in sports.

 # Other factors impact are not importent for good scores

[ ]: