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

```
df = pd.read_csv("Expanded_data_with_more_features.csv")
```

```
df.head()
```

	Unnamed: 0	Gender	EthnicGroup	ParentEduc	LunchType	TestPrep	ParentMaritalStatus	PracticeSport	IsFirstChild	NrSiblings	Tr
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	

```
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.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
```

Dropping Unnamed Column

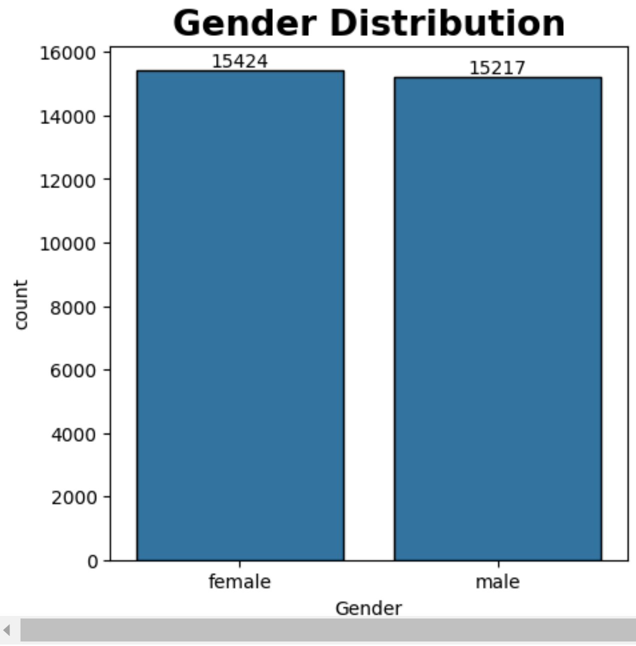
```
df = df.drop("Unnamed: 0", axis = 1)
```

```
df.head()
```

	Gender	EthnicGroup	ParentEduc	LunchType	TestPrep	ParentMaritalStatus	PracticeSport	IsFirstChild	NrSiblings	TransportMea
0	female	NaN	bachelor's degree	standard	none	married	regularly	yes	3.0	school_b
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_b
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_b

Gender Distribution

```
plt.figure(figsize = (5,5))
plt.title("Gender Distribution", fontdict={'fontweight':'bold','fontsize':'19'})
ax = sns.countplot(data = df, x = "Gender", edgecolor = "black")
ax.bar_label(ax.containers[0])
plt.show()
```



From the above chart we have analysed that the number of females is greater than the number of males

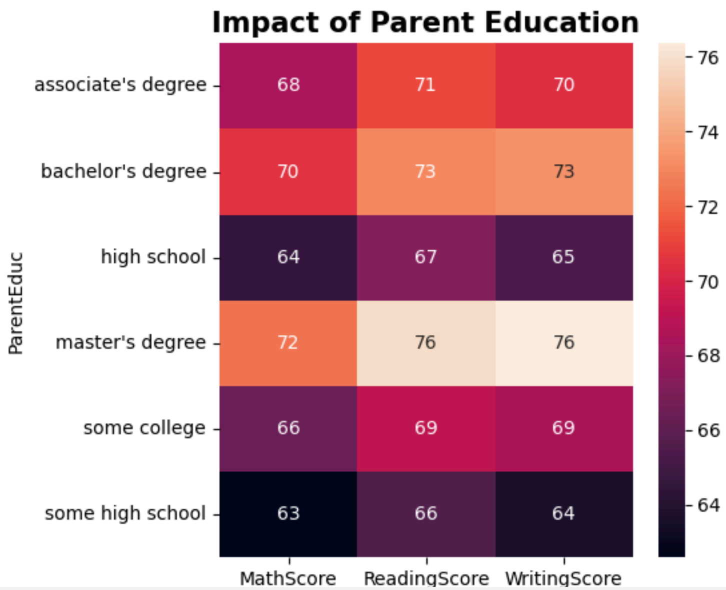
Analyzing impact of ParentEducation on Scores

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

```
groupby
```

	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

```
plt.figure(figsize = (5,5))
plt.title("Impact of Parent Education", fontdict={'fontweight':'bold','fontsize':'15'})
sns.heatmap(groupby, annot = True)
plt.show()
```



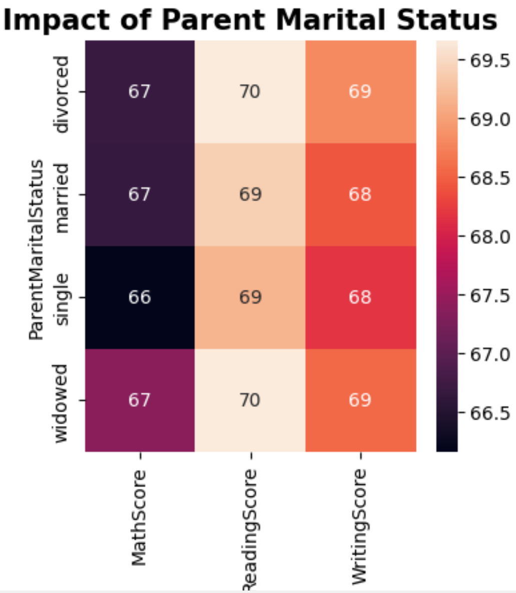
From the above chart we have concluded that education of parents have a good impact on student's scores

Parent Marital Status Analysis

```
groupby1 = df.groupby("ParentMaritalStatus").agg({"MathScore":'mean',"ReadingScore":'mean',"WritingScore":'mean'})
groupby1
```

	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

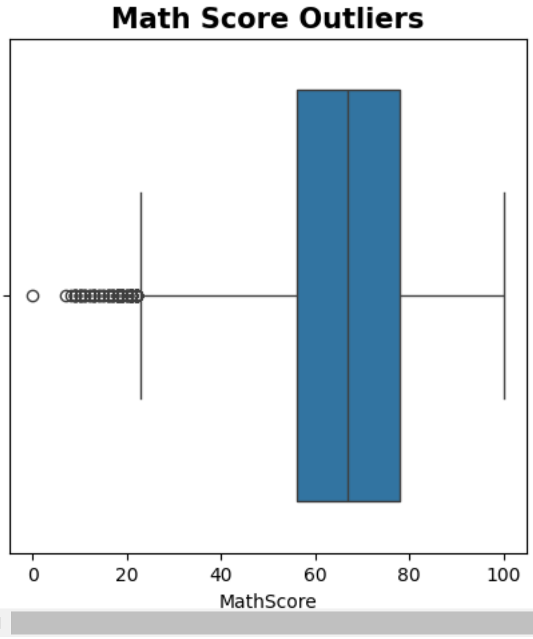
```
plt.figure(figsize = (4,4))
plt.title("Impact of Parent Marital Status", fontdict={'fontweight':'bold','fontsize':'15'})
sns.heatmap(groupby1, annot = True)
plt.show()
```



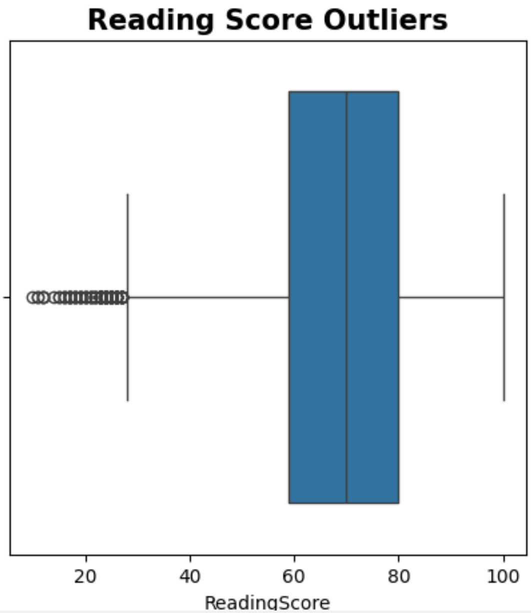
So from the above chart we have analysed that the Marital status of the parents have negligible impact on student's score

Analyzing Outliers

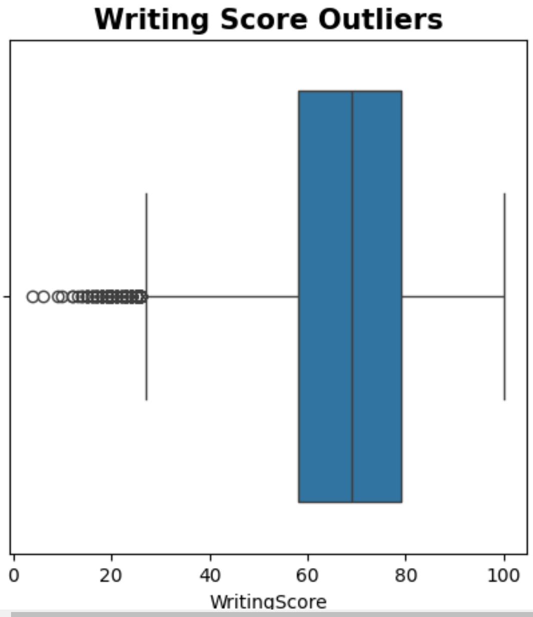
```
plt.figure(figsize = (5,5))
plt.title("Math Score Outliers", fontdict={'fontweight':'bold','fontsize':'15'})
sns.boxplot(data = df, x = "MathScore")
plt.show()
```



```
plt.figure(figsize = (5,5))
plt.title("Reading Score Outliers", fontdict={'fontweight':'bold','fontsize':'15'})
sns.boxplot(data = df, x = "ReadingScore")
plt.show()
```



```
plt.figure(figsize = (5,5))
plt.title("Writing Score Outliers", fontdict={'fontweight':'bold','fontsize':'15'})
sns.boxplot(data = df, x = "WritingScore")
plt.show()
```



Here we conclude that Math subject seems difficult for the students to score good marks

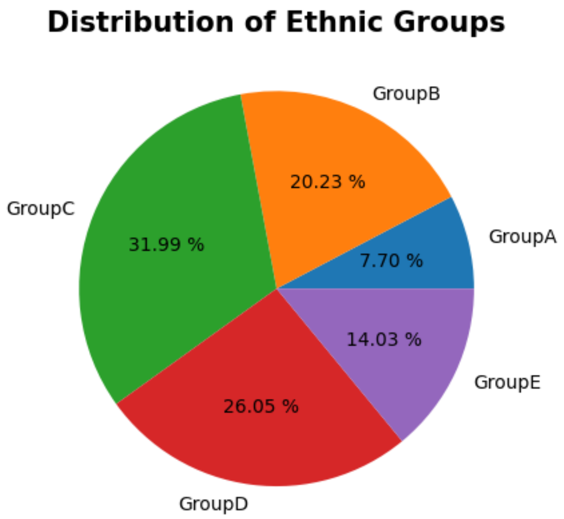
Ethnic Group Distribution

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

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

```
grpA = df.loc[(df["EthnicGroup"] == "group A")].count()
grpB = df.loc[(df["EthnicGroup"] == "group B")].count()
grpC = df.loc[(df["EthnicGroup"] == "group C")].count()
grpD = df.loc[(df["EthnicGroup"] == "group D")].count()
grpE = df.loc[(df["EthnicGroup"] == "group E")].count()

labels = ["GroupA", "GroupB", "GroupC", "GroupD", "GroupE"]
groups = [grpA["EthnicGroup"], grpB["EthnicGroup"], grpC["EthnicGroup"], grpD["EthnicGroup"], grpE["EthnicGroup"]]
plt.pie(groups, labels = labels, autopct = "%1.2f %%")
plt.title("Distribution of Ethnic Groups", fontdict={'fontweight':'bold','fontsize':'15'})
plt.show()
```



From the above pie chart we conclude that Group C holds major portion in the distribution