

```

1 import numpy as np
2 import pandas as pd
3 import matplotlib.pyplot as plt
4 import seaborn as sns
5 df = pd.read_csv("student scores.csv")
6 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

```

```
1 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

```
1 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

```

```
1 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

```

```

1 #Drop 'Unnamed:0' column
2 df = df.drop("Unnamed: 0", axis = 1)
3 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 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 df.head()
```

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

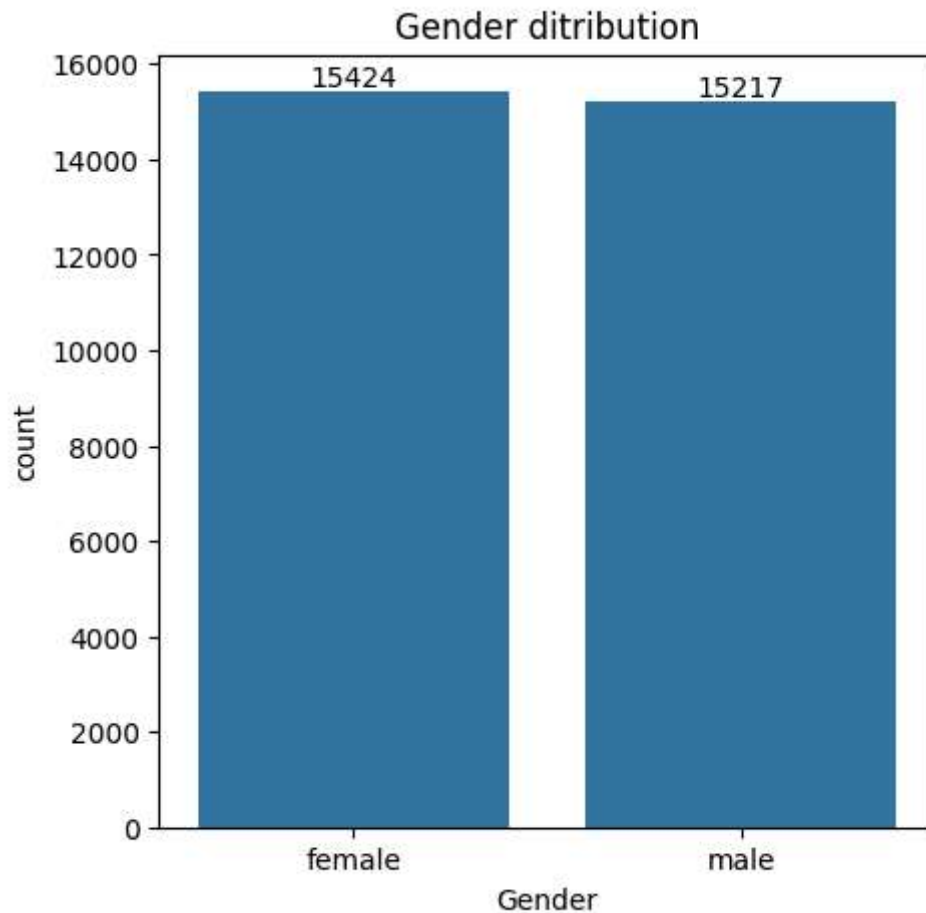
Next steps:

[Generate code with df](#)[View recommended plots](#)

```

1 #Gender Distribution
2 plt.figure(figsize= (5,5))
3 ax = sns.countplot(data = df, x = "Gender")
4 ax.bar_label(ax.containers[0])
5 plt.title("Gender ditribution")
6 plt.show()

```

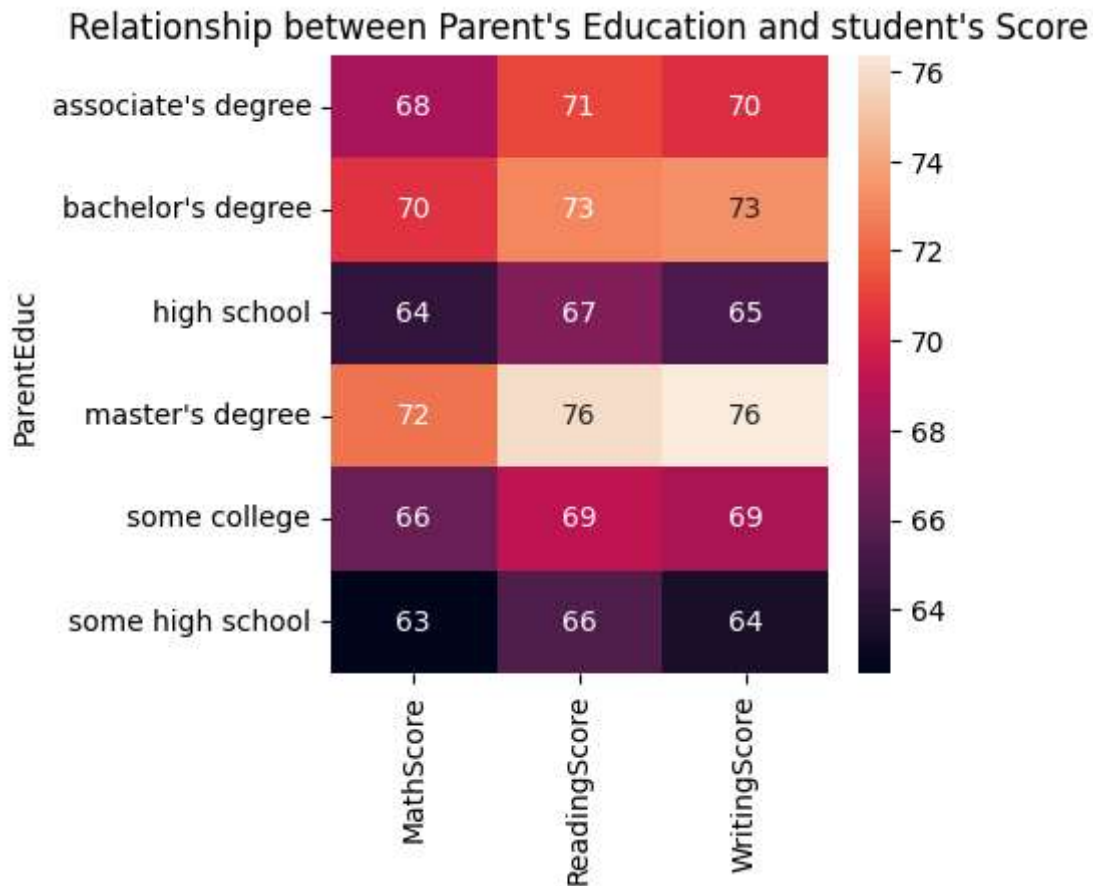


```
1 #from the above chart we have analyze thar:
2 #the numbers of females in the data is more than the number of males
```

```
1 gb = df.groupby("ParentEduc").agg({"MathScore":"mean","ReadingScore":"mean","WritingScore":"mean"})
2 print(gb)
```

ParentEduc	MathScore	ReadingScore	WritingScore
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

```
1 #Relationship between Parent's Education and student's Score
2 plt.figure(figsize= (4,4))
3 sns.heatmap(gb , annot = True)
4 plt.title("Relationship between Parent's Education and student's Score")
5 plt.show()
```



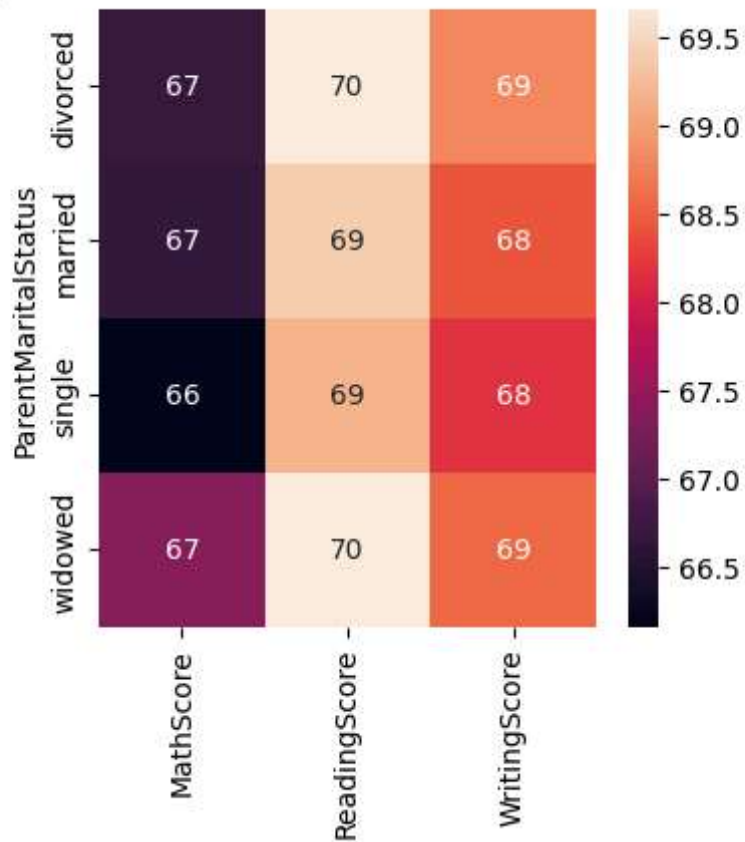
1 #from the above chart we have concluded that the education:
 2 #of the parents have a good effect of students

```
1 gb1 = df.groupby("ParentMaritalStatus").agg({"MathScore":"mean", "ReadingScore":"mean", "Wr
2 print(gb1)
```

ParentMaritalStatus	MathScore	ReadingScore	WritingScore
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

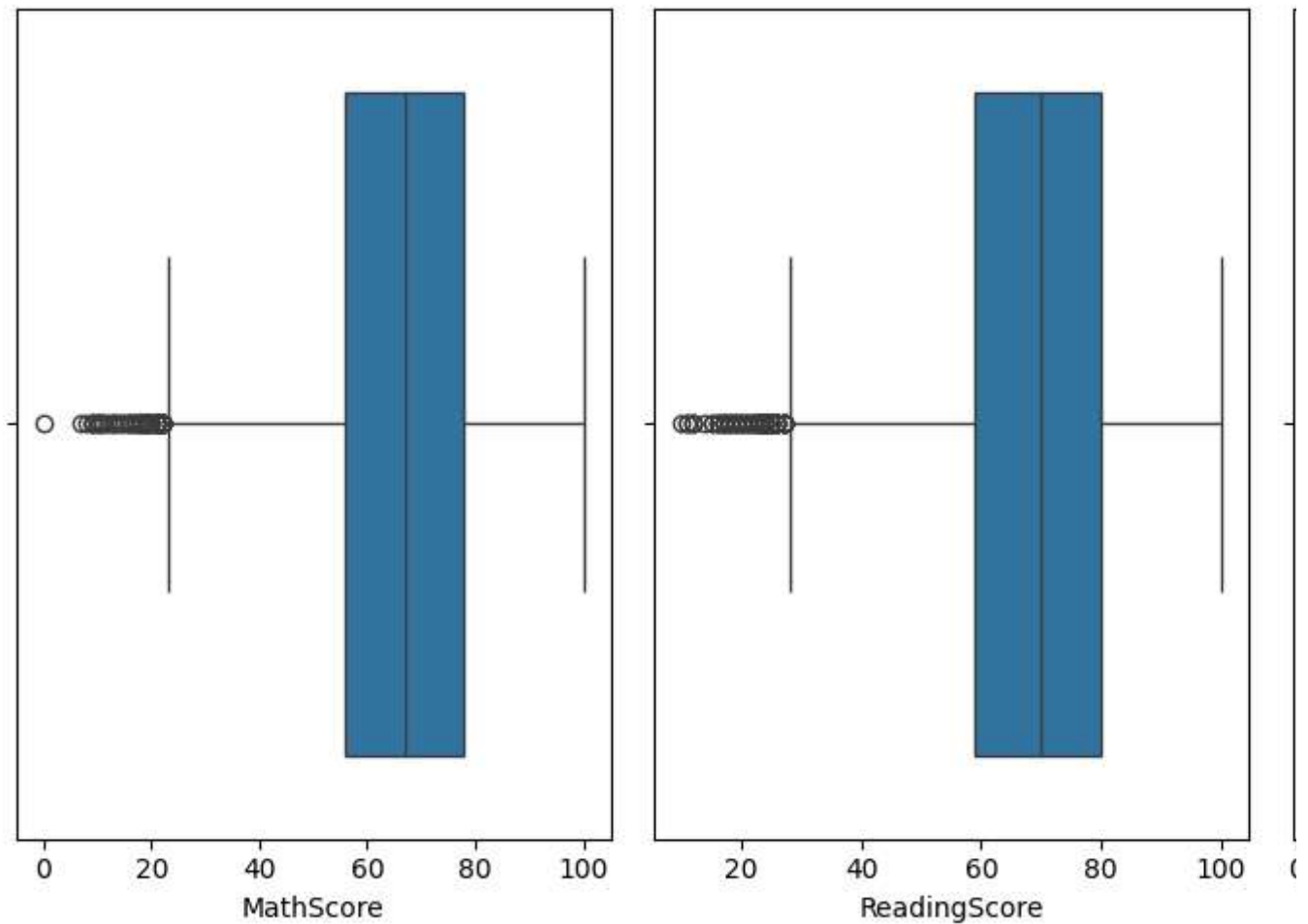
```
1 #Relationship between Parent's Marital status and student's Score
2 plt.figure(figsize= (4,4))
3 sns.heatmap(gb1 , annot = True)
4 plt.title("Relationship between Parent's Marital status and student's Score")
5 plt.show()
```

Relationship between Parent's Marital status and student's Score



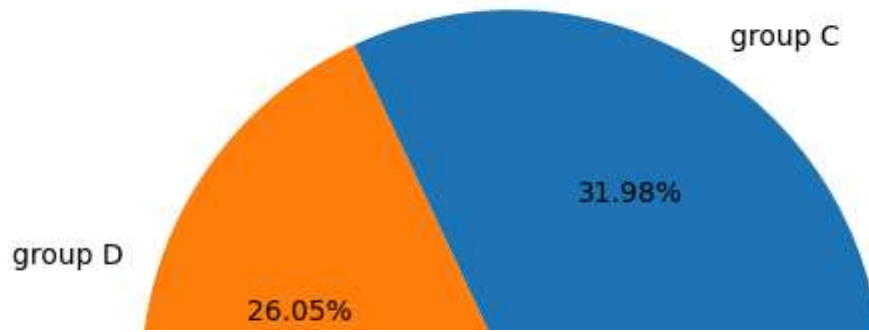
1 #from the above chart we have concluded that their is no\negligible:
 2 # impact on the student score due to their parent's marital status

```
1 plt.figure(figsize=(10, 5))
2 plt.subplot(1, 3, 1)
3 sns.boxplot(data=df, x="MathScore")
4 plt.subplot(1, 3, 2)
5 sns.boxplot(data=df, x="ReadingScore")
6 plt.subplot(1, 3, 3)
7 sns.boxplot(data=df, x="WritingScore")
8 plt.tight_layout()
9 plt.show()
```



```
1 # Distribution of Ethnic Groups
2 group_counts = df['EthnicGroup'].value_counts()
3 plt.figure(figsize=(6, 6))
4 plt.pie(group_counts, labels=group_counts.index, autopct="%1.2f%%")
5 plt.title("Distribution of Ethnic Groups")
6 plt.show()
7
```

Distribution of Ethnic Groups



Double-click (or enter) to edit



```
1 # Distribution of Ethnic Groups
2 plt.figure(figsize=(8, 6))
3 ax = sns.countplot(data=df, x='EthnicGroup', palette="Set2")
4 ax.bar_label(ax.containers[0])
5 plt.title("Distribution of Ethnic Groups")
6 plt.show()
7
```

<ipython-input-12-a99935c5eeb3>:3: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0.

```
ax = sns.countplot(data=df, x='EthnicGroup', palette="Set2")
```

