


```
import pandas as pd
df = pd.read_csv("/content/student.csv")
df.head()
```




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



Next steps:

[Generate code with df](#)


 [View recommended plots](#)

```
df.describe()
```



| | Unnamed: 0 | NrSiblings | MathScore | ReadingScore | WritingScore |  |
|-------|--------------|--------------|--------------|--------------|--------------|---|
| count | 11040.000000 | 10472.000000 | 11040.000000 | 11040.000000 | 11039.000000 |  |
| mean | 491.028804 | 2.125000 | 66.689493 | 69.441757 | 68.449316 | |
| std | 286.041837 | 1.471642 | 15.442964 | 14.747144 | 15.435190 | |
| min | 0.000000 | 0.000000 | 0.000000 | 8.000000 | 6.000000 | |
| 25% | 245.000000 | 1.000000 | 56.000000 | 59.000000 | 58.000000 | |
| 50% | 488.000000 | 2.000000 | 67.000000 | 70.000000 | 69.000000 | |
| 75% | 733.000000 | 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: 11040 entries, 0 to 11039
```

Data columns (total 15 columns):

| # | Column | Non-Null Count | Dtype |
|----|---------------------|----------------|---------|
| 0 | Unnamed: 0 | 11040 non-null | int64 |
| 1 | Gender | 11040 non-null | object |
| 2 | EthnicGroup | 10375 non-null | object |
| 3 | ParentEduc | 10382 non-null | object |
| 4 | LunchType | 11040 non-null | object |
| 5 | TestPrep | 10381 non-null | object |
| 6 | ParentMaritalStatus | 10605 non-null | object |
| 7 | PracticeSport | 10816 non-null | object |
| 8 | IsFirstChild | 10678 non-null | object |
| 9 | NrSiblings | 10472 non-null | float64 |
| 10 | TransportMeans | 9900 non-null | object |
| 11 | WklyStudyHours | 10701 non-null | object |
| 12 | MathScore | 11040 non-null | int64 |
| 13 | ReadingScore | 11040 non-null | int64 |
| 14 | WritingScore | 11039 non-null | float64 |

dtypes: float64(2), int64(3), object(10)
memory usage: 1.3+ MB

df.isnull().sum()

```

→ Unnamed: 0      0
  Gender          0
  EthnicGroup     665
  ParentEduc      658
  LunchType       0
  TestPrep        659
  ParentMaritalStatus 435
  PracticeSport   224
  IsFirstChild    362
  NrSiblings      568
  TransportMeans  1140
  WklyStudyHours  339
  MathScore       0
  ReadingScore    0
  WritingScore    1
dtype: int64

```

2nd step> Transformation

✓ *DROP Unnamed Column*

```

df = df.drop("Unnamed: 0", axis=1) # Remove the extra space after "Unnamed: 0"

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 | 74.0 |
| 1 | 5 - 10 | 69 | 90 | 88.0 |
| 2 | < 5 | 87 | 93 | 91.0 |
| 3 | 5 - 10 | 45 | 56 | 42.0 |
| 4 | 5 - 10 | 76 | 78 | 75.0 |

#change Weekly StudyHours col.

```
import pandas as pd
```

```
# Load your data into a DataFrame
```

```
df = pd.read_csv('/content/student.csv')
```

```
df["WklyStudyHours"] = df["WklyStudyHours"].str.replace("05-Oct", "5-10")
df.head()
```

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

Next steps:

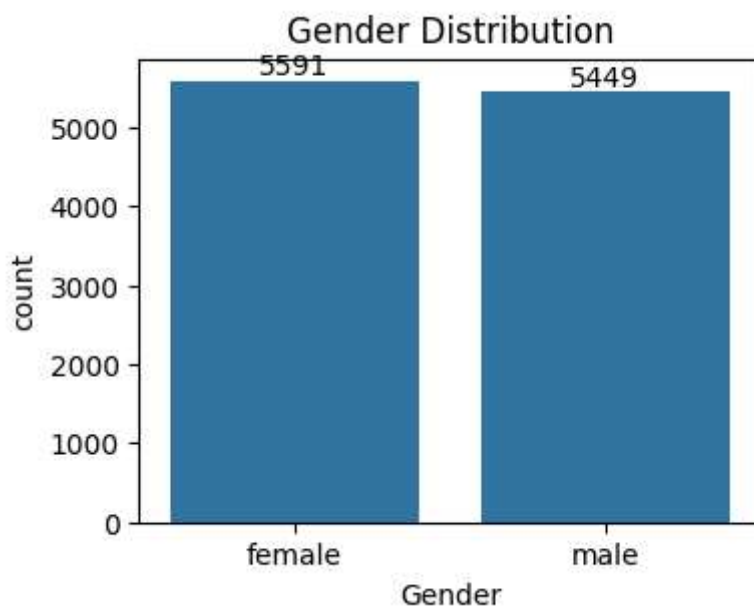
[Generate code with df](#)
[View recommended plots](#)

step 3 > Analysis

✓ Gender Distribution

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

df = pd.read_csv('/content/student.csv')
plt.figure(figsize=(4,3))
Ax=sns.countplot(data = df, x="Gender")
plt.title("Gender Distribution")
Ax.bar_label(Ax.containers[0])
plt.show()
```



#interpretation

- ✓ from the above chart we have analysed that the no of females in the data is more then the males

Start coding or [generate](#) with AI.

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

```

ParentEduc
associate's degree 68.664493    71.435306    70.552156
bachelor's degree  70.554304    73.164119    73.448470
high school        64.259701    67.036318    65.079602
master's degree    72.488712    76.173971    76.857902
some college       66.506947    68.912421    68.194526
some high school   62.806178    65.812656    63.932735

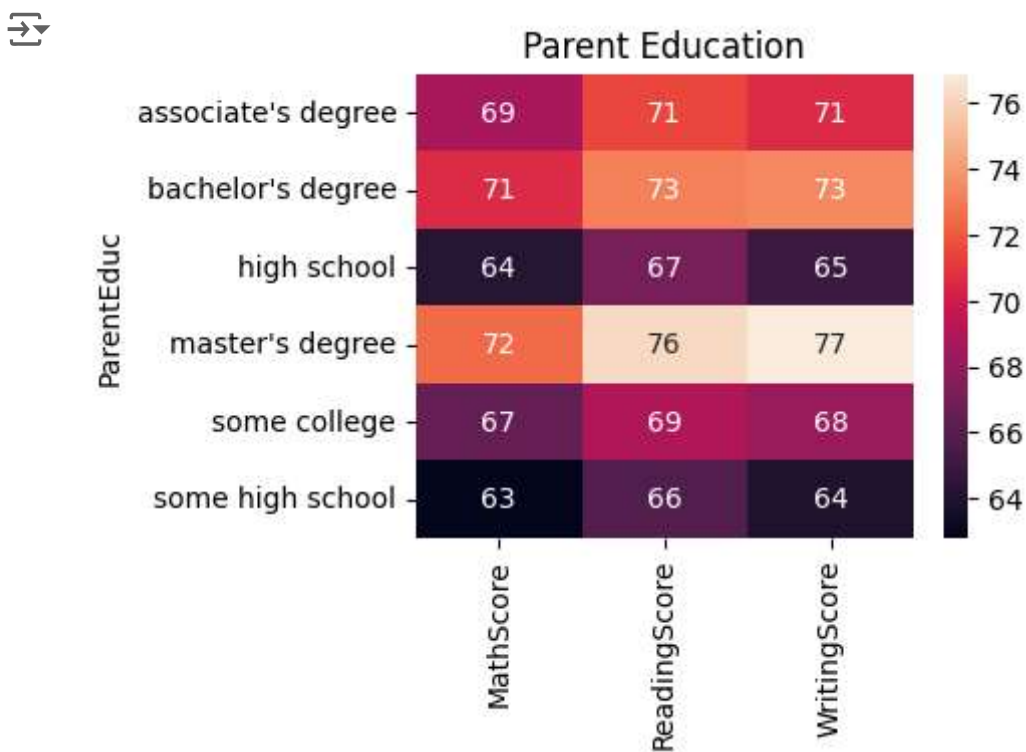
```

```

import seaborn as sns
import matplotlib.pyplot as plt # Import matplotlib.pyplot before using it

plt.figure(figsize=(4,3))
plt.title("Parent Education")
sns.heatmap(gb, annot=True) #annot is used to show value in portion
plt.show()

```



#From the above chart we have conclude that there good impact on the student score's, due to their parents education

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

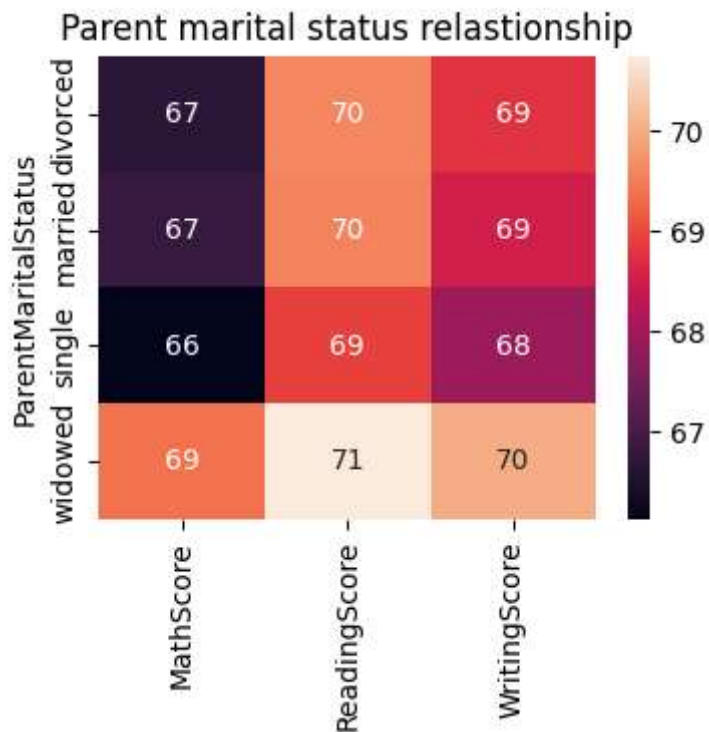
```

ParentMaritalStatus
MathScore  ReadingScore  WritingScore

```

| | | | |
|----------|-----------|-----------|-----------|
| divorced | 66.650255 | 69.607244 | 68.776331 |
| married | 66.821329 | 69.562633 | 68.517720 |
| single | 66.131811 | 68.921875 | 67.920673 |
| widowed | 69.392694 | 70.744292 | 69.995434 |


```
plt.figure(figsize=(4,3))
plt.title("Parent marital status relationship")
sns.heatmap(gb1, annot=True) #annot is used to show value in portion
plt.show()
```

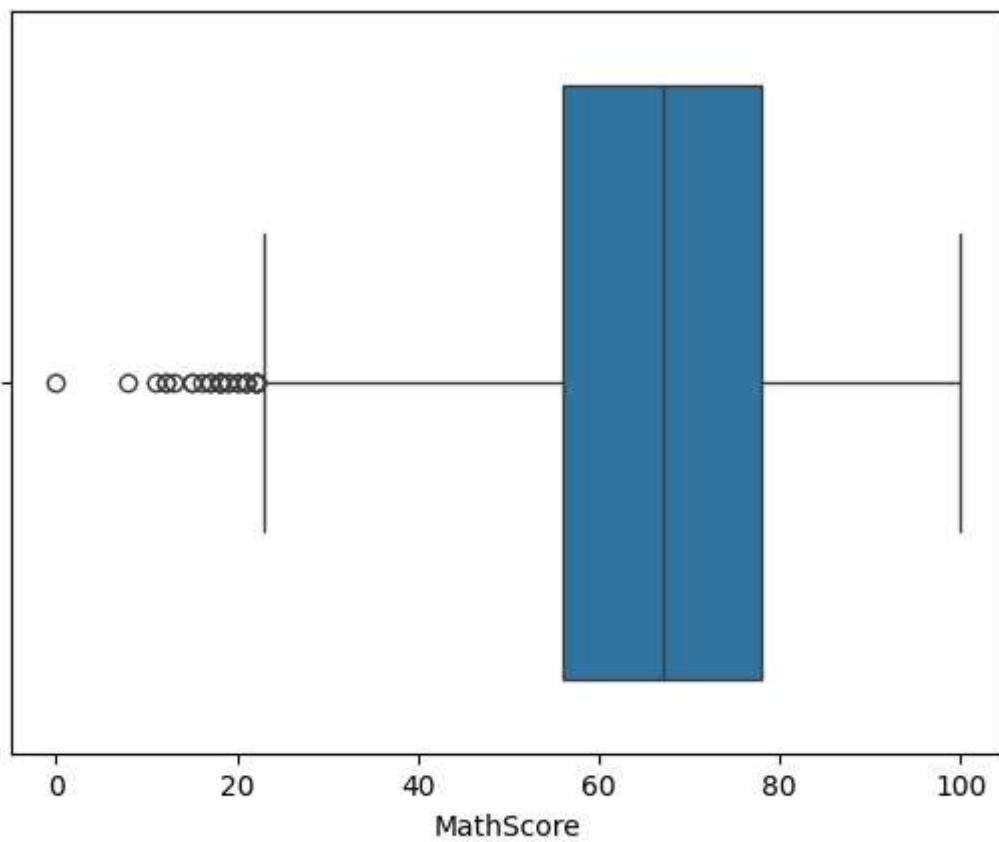


#From the above chart we have conclude that there is

- ✓ no/negligible impact on the student score's, due to their parents marital parents

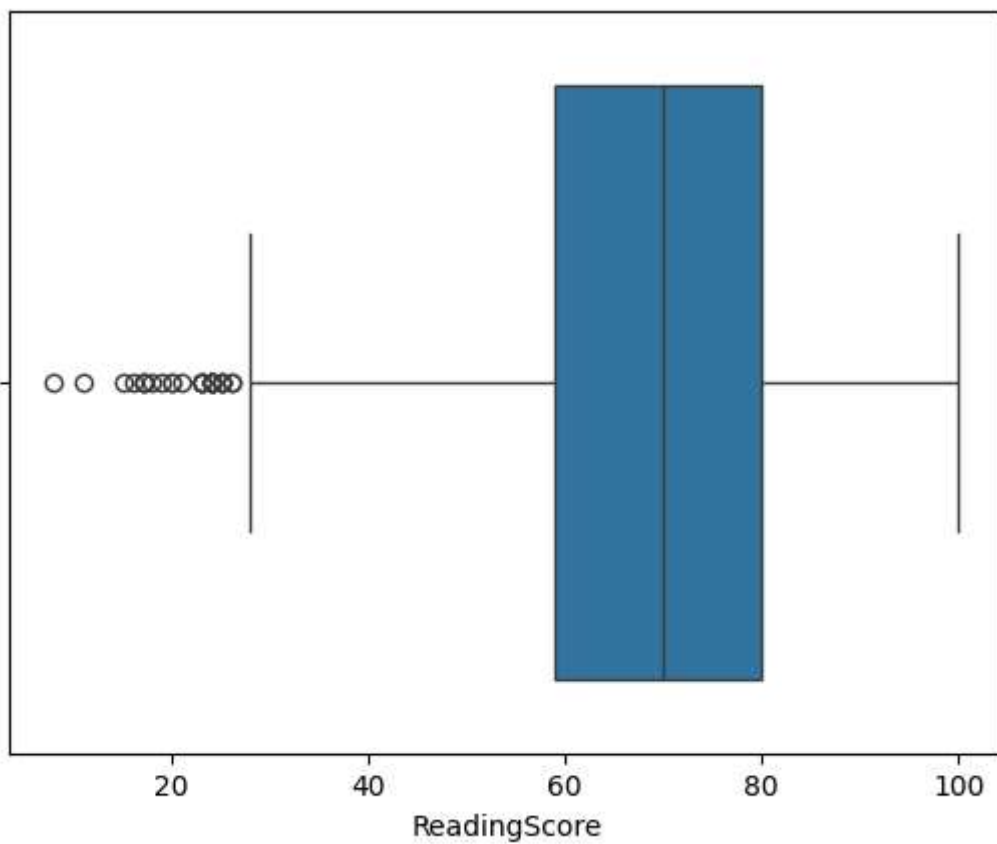
```
#to detect outliers in MathScor
sns.boxplot(data=df,x="MathScore")
```

 <Axes: xlabel='MathScore'>



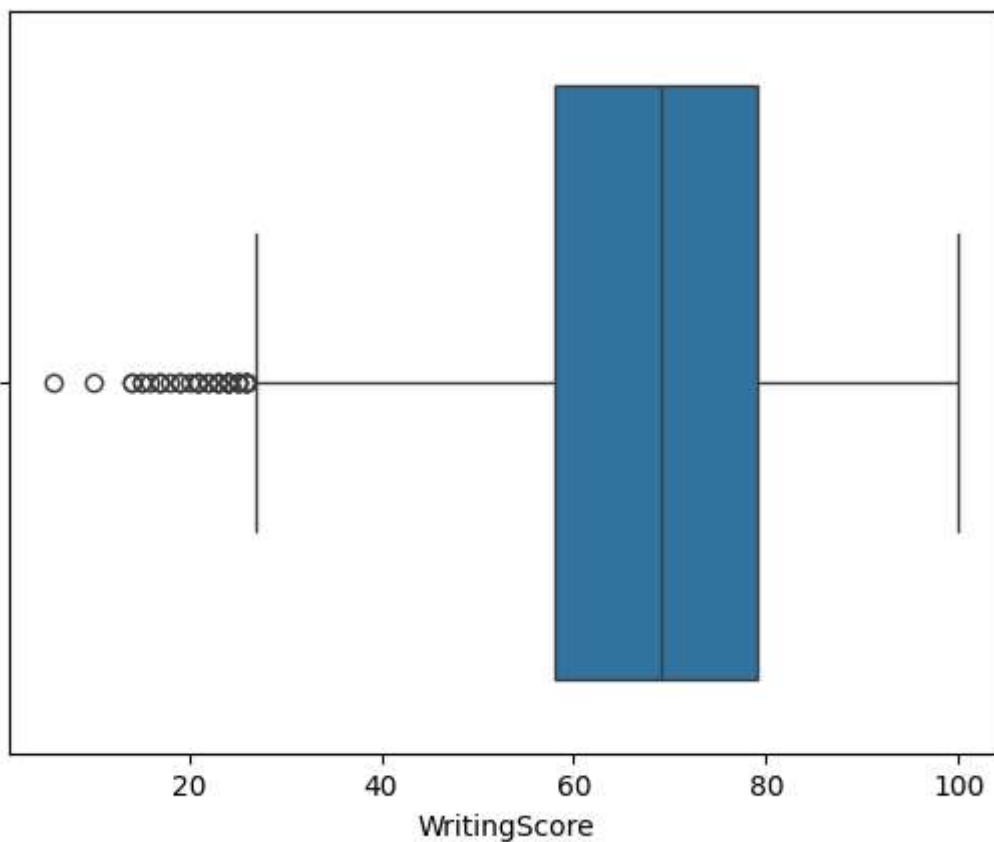
```
#to detect outliers in ReadingScore  
sns.boxplot(data=df,x="ReadingScore")
```

↔ <Axes: xlabel='ReadingScore'>



```
#to detect outliers in WritingScore  
sns.boxplot(data=df,x="WritingScore")
```


↔ <Axes: xlabel='WritingScore'>



✓ Distribution of EthnicGroup

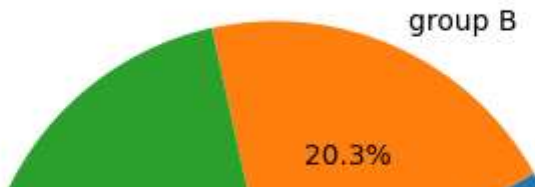
```
groupA= df.loc[df['EthnicGroup'] == 'group A'].count()
groupb= df.loc[df['EthnicGroup'] == 'group B'].count()
groupc= df.loc[df['EthnicGroup'] == 'group C'].count()
groupd= df.loc[df['EthnicGroup'] == 'group D'].count()
groupe= df.loc[df['EthnicGroup'] == 'group E'].count()
plt.title("Distribution of EthnicGroup")
plt.pie([groupA['EthnicGroup'],groupb['EthnicGroup'],groupc['EthnicGroup'],groupd['EthnicGr
```

```

([<matplotlib.patches.Wedge at 0x7e8243e5b790>,
 <matplotlib.patches.Wedge at 0x7e8243e5b6a0>,
 <matplotlib.patches.Wedge at 0x7e8243eb84c0>,
 <matplotlib.patches.Wedge at 0x7e8243eb8b50>,
 <matplotlib.patches.Wedge at 0x7e8243eb91e0>],
 [Text(1.064692009417757, 0.27646143507183574, 'group A'),
 Text(0.4520206311327371, 1.0028346568753805, 'group B'),
 Text(-1.0284133166321539, 0.3903409409396527, 'group C'),
 Text(-0.1492597443788949, -1.089826375487374, 'group D'),
 Text(0.9900725755261317, -0.4793290051635229, 'group E')],
 [Text(0.5807410960460492, 0.1507971464028195, '8.1%'),
 Text(0.24655670789058384, 0.5470007219320258, '20.3%'),
 Text(-0.560952718162993, 0.21291324051253782, '31.6%'),
 Text(-0.08141440602485174, -0.5944507502658404, '25.6%'),
 Text(0.5400395866506172, -0.2614521846346488, '14.4%')])

```

Distribution of EthnicGroup



```

ax=sns.countplot(data=df,x="EthnicGroup")
ax.bar_label(ax.containers[0])
plt.title("Distribution of EthnicGroup")
plt.show()

```



Distribution of EthnicGroup

