

student-result-analysis

June 10, 2025

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

```
[48]: df = pd.read_csv(r"/content/Student_Score.csv")
```

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

```
[49]: 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
```

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

```
[50]: 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

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

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

```
[52]:
```

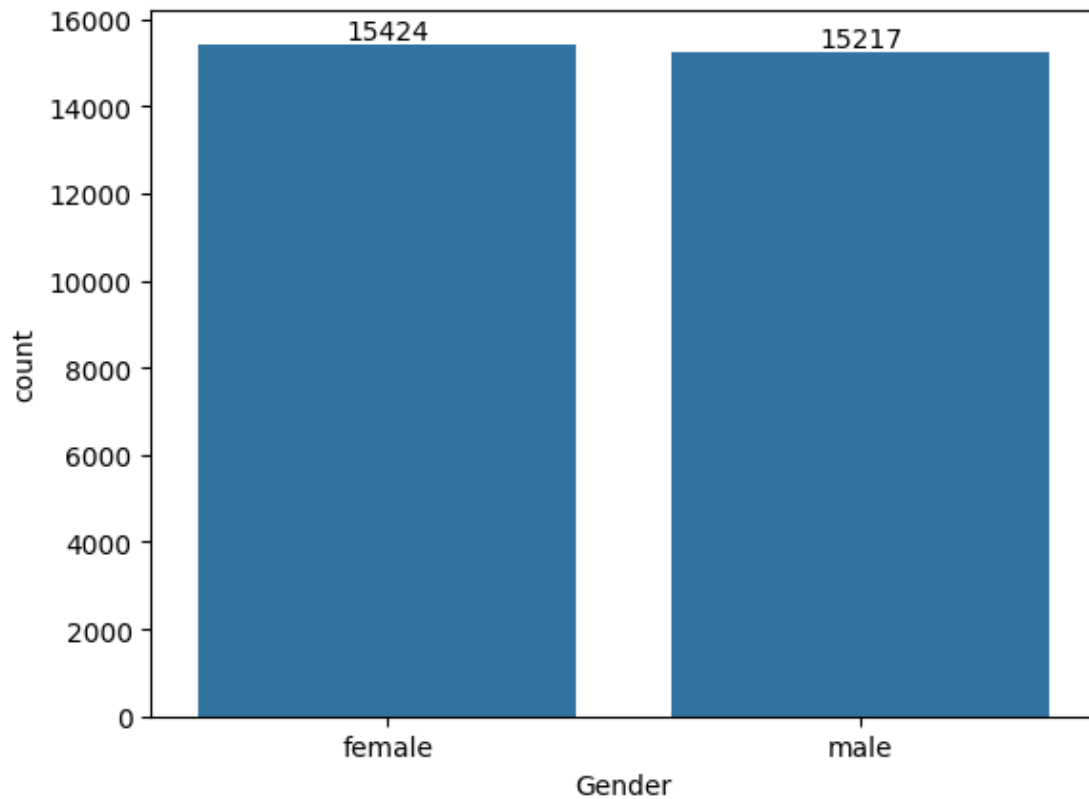
	Gender	EthnicGroup	ParentEduc	LunchType	TestPrep	\
0	female	NaN	bachelor's degree	standard	none	
1	female	group C	some college	standard	NaN	
2	female	group B	master's degree	standard	none	
3	male	group A	associate's degree	free/reduced	none	
4	male	group C	some college	standard	none	

	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

Gender Distribution

```
[53]: ax = sns.countplot(data = df, x = "Gender")
plt.figure(figsize = (10,5))
for bars in ax.containers:
    ax.bar_label(bars)
plt.show()
```



<Figure size 1000x500 with 0 Axes>

#from the above chart we have analysed that:

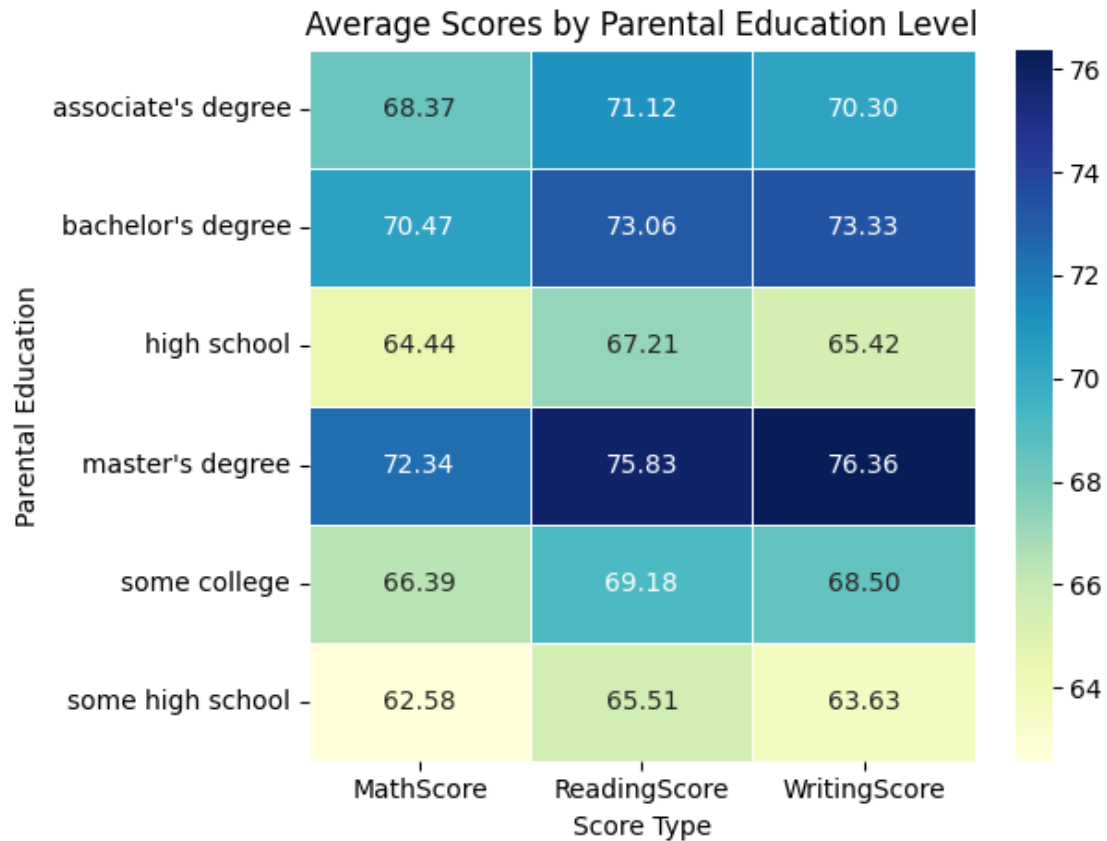
The number of **females** in the data is more than the number of **males**

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

	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

```
[55]: sns.heatmap(data = gb, annot=True, cmap='YlGnBu', linewidths=0.5, fmt=".2f")
      plt.title("Average Scores by Parental Education Level")
```

```
plt.xlabel("Score Type")
plt.ylabel("Parental Education")
plt.tight_layout()
plt.show()
```

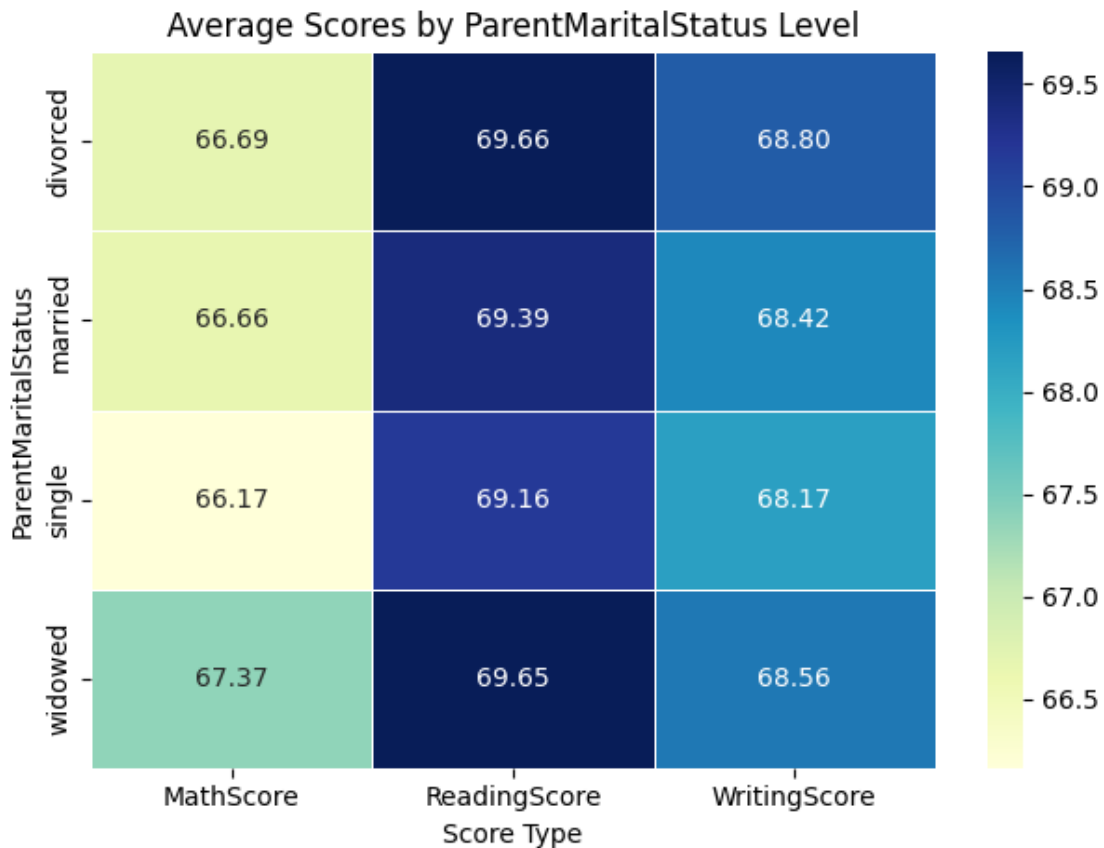


From the above chart we have concluded that the education of the parent have a good impact on the students score

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

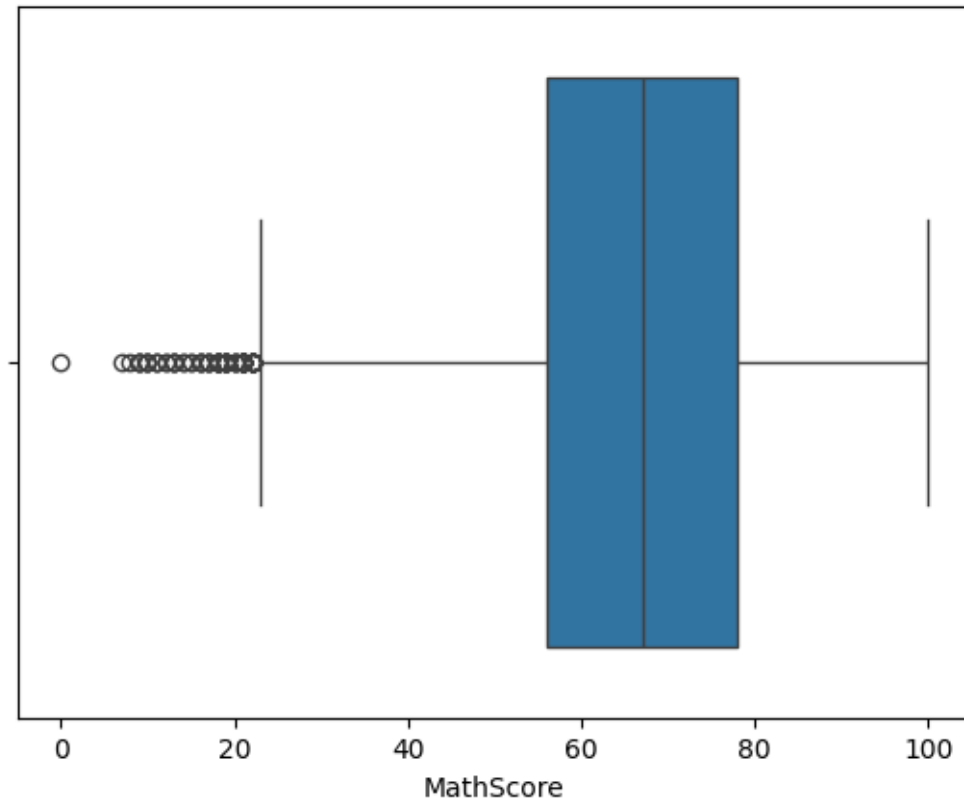
ParentMaritalStatus	MathScore	ReadingScore	WritingScore
divorced	66.691197	69.655011	68.799146
married	66.657326	69.389575	68.420981
single	66.165704	69.157250	68.174440
widowed	67.368866	69.651438	68.563452

```
[59]: sns.heatmap(data = gb1, annot=True, cmap='YlGnBu', linewidths=0.5, fmt=".2f")
plt.title("Average Scores by ParentMaritalStatus Level")
plt.xlabel("Score Type")
plt.ylabel("ParentMaritalStatus")
plt.tight_layout()
plt.show()
```

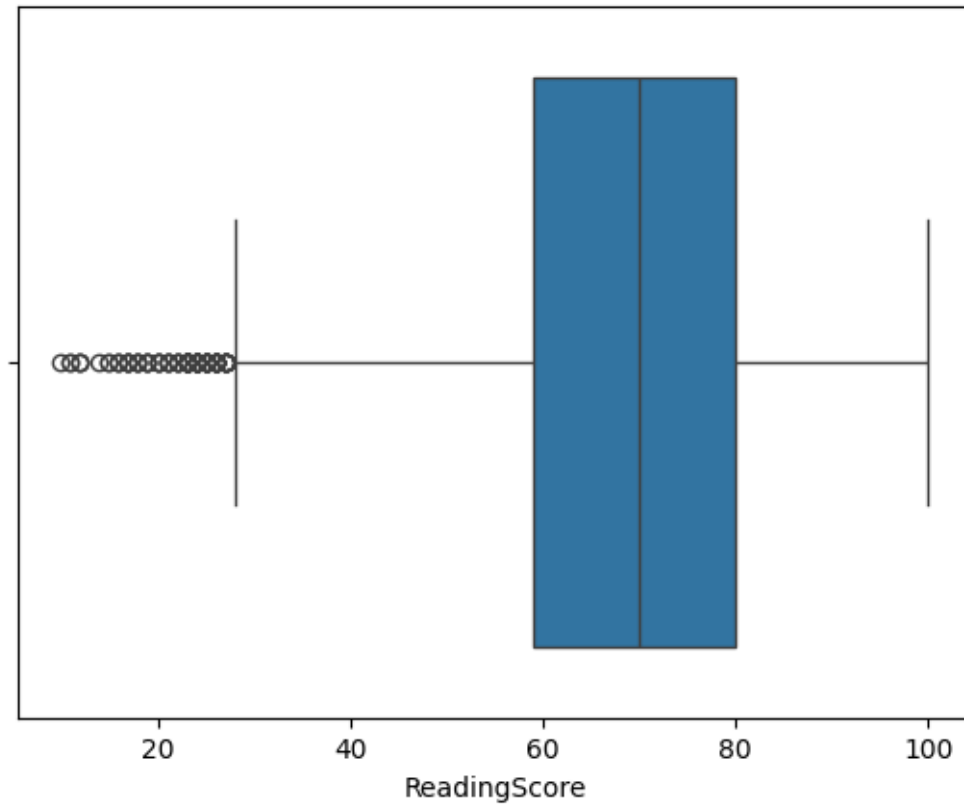


From the above chart we have concluded that the marital status of the parent have no/negligible impact on the students score due to there Marital Status

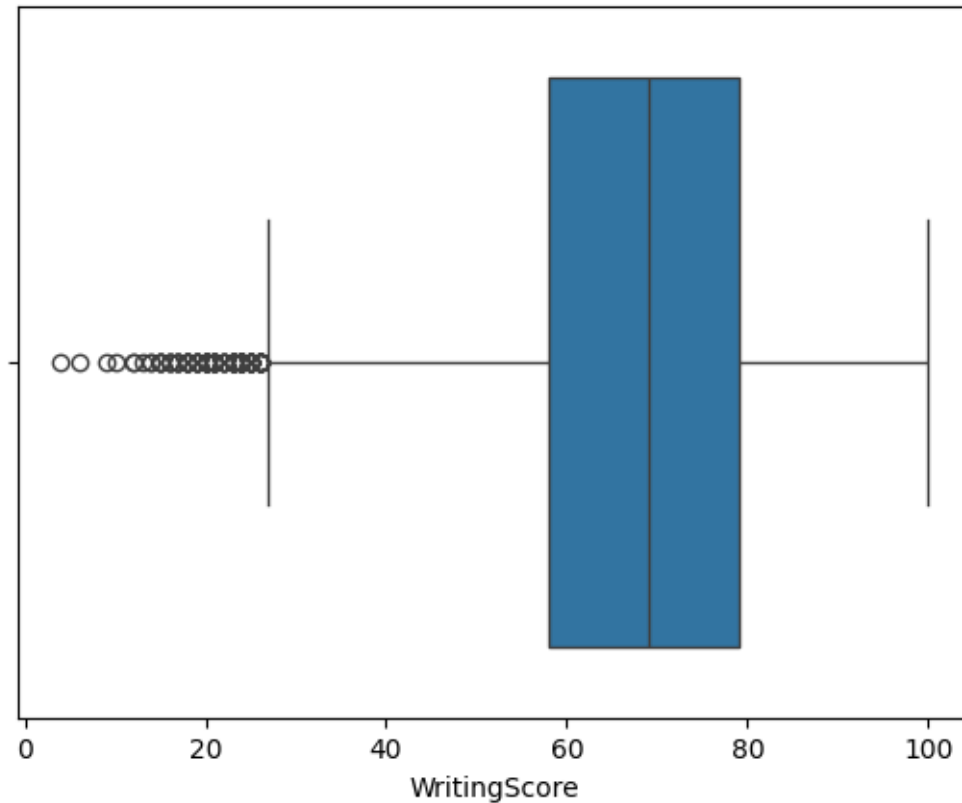
```
[61]: sns.boxplot(x = "MathScore", data = df)
plt.show()
```



```
[62]: sns.boxplot(x = "ReadingScore", data = df)  
plt.show()
```



```
[63]: sns.boxplot(x = "WritingScore", data = df)  
plt.show()
```



The box plots for Math, Reading, and Writing scores show the distribution and potential outliers for each subject, indicating the range and spread of student performance.

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

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

Distribution Of EthnicGroup

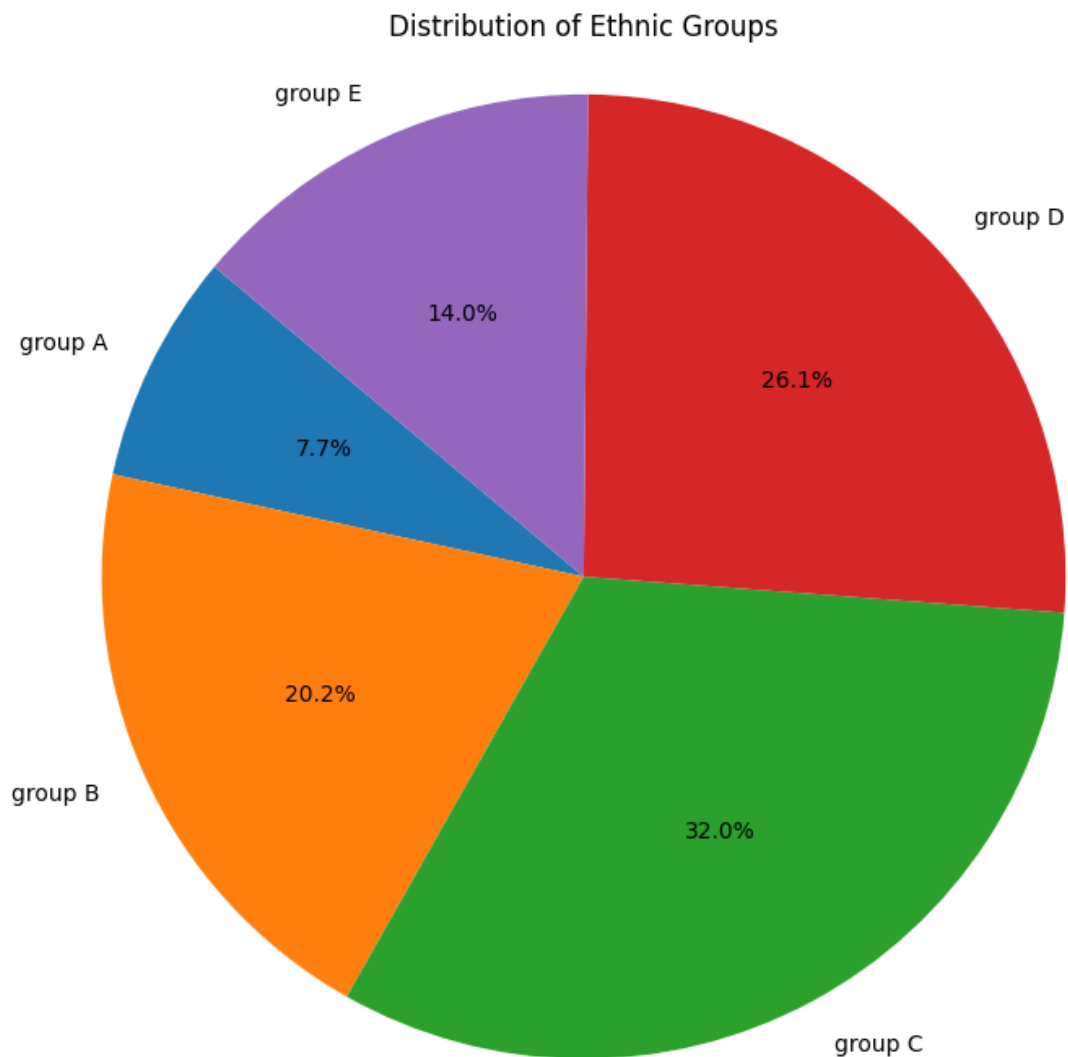
```
[ ]:
```

```
[79]: # Count the number of students in each ethnic group
groupA = df[df["EthnicGroup"] == "group A"].shape[0]
groupB = df[df["EthnicGroup"] == "group B"].shape[0]
groupC = df[df["EthnicGroup"] == "group C"].shape[0]
groupD = df[df["EthnicGroup"] == "group D"].shape[0]
groupE = df[df["EthnicGroup"] == "group E"].shape[0]

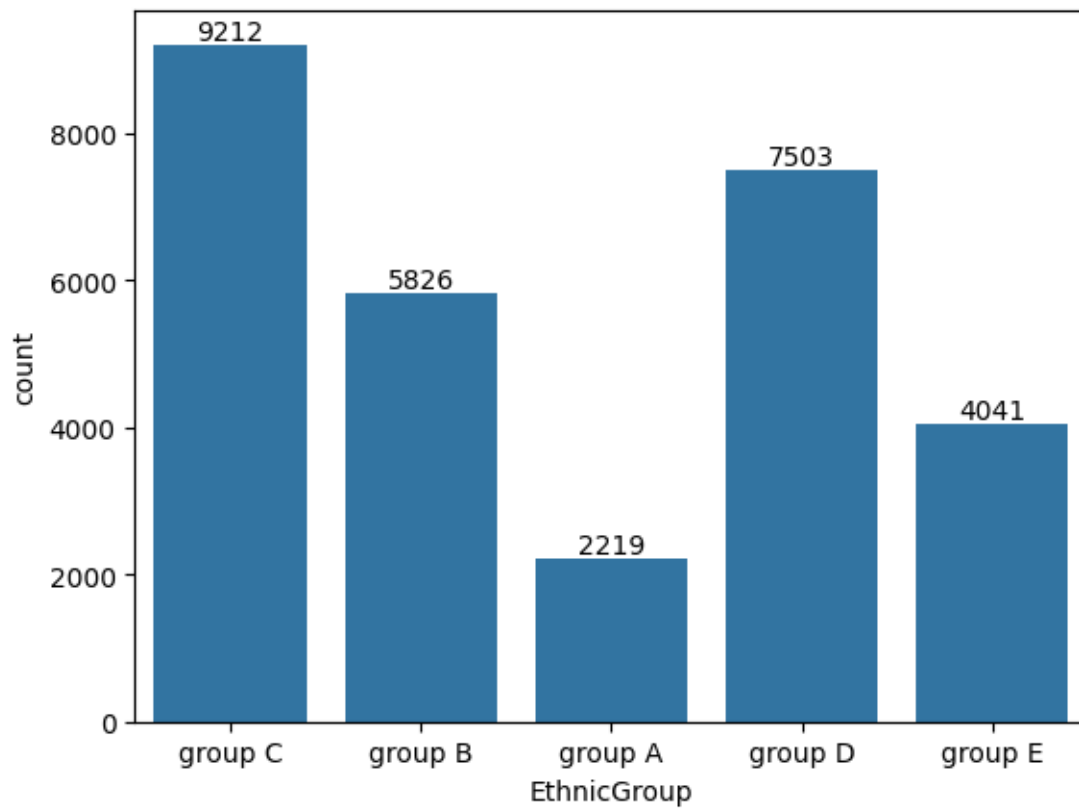
# Labels and values
labels = ["group A", "group B", "group C", "group D", "group E"]
sizes = [groupA, groupB, groupC, groupD, groupE]
```



```
# Plot pie chart
plt.figure(figsize=(7, 7))
plt.pie(sizes, labels=labels, autopct='%1.1f%%', startangle=140)
plt.title("Distribution of Ethnic Groups")
plt.axis('equal') # Equal aspect ratio ensures pie is a circle
plt.tight_layout()
plt.show()
```



```
[80]: ax = sns.countplot(data = df, x= "EthnicGroup")
ax.bar_label(ax.containers[0])
plt.show()
```



The distribution of ethnic groups in the dataset is also shown, with group C being the largest.