"Data Preprocessing on Student Data Set"

```
# Steps of preprocessing of data

1.Import necessasary library
2.Read Datset
3.sanity check of data
4.Exploratory Data Analysis(EDA)
5.Missing Value treatments
6.Outliers treatments
7.Duplicates & garbage value treatments
8.NormaliZation
9.Encoding of Data
```

Step-1: Import the Libraries

In [303…

```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
```

Step-2: Read or Load the Dataset

In [505…

```python
df=pd.read_csv('test.csv')
```

In [148…

```python
df.head()
```

Out[148…

	StudentID	Age	Gender	Ethnicity	ParentalEducation	StudyTimeWeekly	Absences	Tutoring	ParentalSupport	Extracurricular
0	2340	16	1	Other	Higher	5.044048	25	1	Moderate	1
1	2923	18	0	Other	Bachelor	18.731312	12	0	Moderate	1
2	2077	16	0	Asian	Some College	0.213403	23	1	Moderate	0
3	2735	15	1	African American	Higher	14.645811	28	0	Moderate	0
4	2245	17	0	Other	Some College	11.436575	1	0	High	1

In [150…

```python
df.tail()
```

Out[150…

	StudentID	Age	Gender	Ethnicity	ParentalEducation	StudyTimeWeekly	Absences	Tutoring	ParentalSupport	Extracurricula
378	1380	15	0	Caucasian	Some College	8.991978	10	1	High	
379	1929	16	1	African American	Some College	16.023430	4	1	Moderate	
380	2280	18	1	Caucasian	Some College	2.832227	18	1	Very High	
381	2353	17	0	Caucasian	Some College	13.600921	22	0	Low	
382	1592	18	0	Asian	Some College	7.560499	1	0	Low	

Step-3: Sanity check of Data

In [153…

```python
#shape()
df.shape
```

Out[153…

```
(383, 14)
```

In [155…

```python
#info()
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 383 entries, 0 to 382
Data columns (total 14 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   StudentID          383 non-null    int64  
 1   Age                383 non-null    int64  
 2   Gender             383 non-null    int64  
 3   Ethnicity          383 non-null    object 
 4   ParentalEducation  343 non-null    object 
 5   StudyTimeWeekly    383 non-null    float64
 6   Absences           383 non-null    int64  
 7   Tutoring           383 non-null    int64  
 8   ParentalSupport    353 non-null    object 
 9   Extracurricular    383 non-null    int64  
 10  Sports             383 non-null    int64  
 11  Music              383 non-null    int64  
 12  Volunteering       383 non-null    int64  
 13  GPA                383 non-null    float64
dtypes: float64(2), int64(9), object(3)
memory usage: 42.0+ KB
```

In [157…

```
df.describe()
```

Out[157…

	StudentID	Age	Gender	StudyTimeWeekly	Absences	Tutoring	Extracurricular	Sports	Music	Vo
count	383.000000	383.000000	383.000000	383.000000	383.000000	383.000000	383.000000	383.000000	383.000000	
mean	2191.046997	16.493473	0.516971	9.851567	14.629243	0.308094	0.360313	0.326371	0.214099	
std	687.144172	1.094649	0.500366	5.706828	8.478083	0.462310	0.480719	0.469498	0.410733	
min	1004.000000	15.000000	0.000000	0.025689	0.000000	0.000000	0.000000	0.000000	0.000000	
25%	1598.500000	16.000000	0.000000	5.148142	8.000000	0.000000	0.000000	0.000000	0.000000	
50%	2172.000000	17.000000	1.000000	9.727833	14.000000	0.000000	0.000000	0.000000	0.000000	
75%	2815.000000	17.000000	1.000000	14.558504	22.000000	1.000000	1.000000	1.000000	0.000000	
max	3373.000000	18.000000	1.000000	19.916047	29.000000	1.000000	1.000000	1.000000	1.000000	

finding the missing Values

In [160…

```
df.isnull().sum()
```

Out[160…

```
StudentID             0
Age                   0
Gender                0
Ethnicity             0
ParentalEducation    40
StudyTimeWeekly       0
Absences              0
Tutoring              0
ParentalSupport      30
Extracurricular       0
Sports                0
Music                 0
Volunteering          0
GPA                   0
dtype: int64
```

In [170…

```
(df.isnull().sum()/len(df))*100
```

Out[170…

```
StudentID             0.000000
Age                   0.000000
Gender                0.000000
Ethnicity             0.000000
ParentalEducation    10.443864
StudyTimeWeekly       0.000000
Absences              0.000000
Tutoring              0.000000
ParentalSupport       7.832898
Extracurricular       0.000000
Sports                0.000000
Music                 0.000000
Volunteering          0.000000
GPA                   0.000000
dtype: float64
```

Finding the duplicates

In [173…

```
df.duplicated().sum()
```

Out[173…

```
0
```

Identifying garbage values

```
which are non-related to object data types or in the another format of data.
```

In [179…

```
for i in df.select_dtypes(include='object').columns:
    print(df[i].value_counts())
    print('****'*10)
```

```
Ethnicity
Caucasian           197
African American     74
Asian                68
Other                44
Name: count, dtype: int64
****************************************
ParentalEducation
Some College    149
High School     120
Bachelor         52
Higher           22
Name: count, dtype: int64
****************************************
ParentalSupport
Moderate     127
High         103
Low           88
Very High     35
Name: count, dtype: int64
****************************************
```

Step-4: EDA (Exploratory Data Analysis)

descriptive statistics

In [183…

```
df.describe().T
```

Out[183…

	count	mean	std	min	25%	50%	75%	max
StudentID	383.0	2191.046997	687.144172	1004.000000	1598.500000	2172.000000	2815.000000	3373.000000
Age	383.0	16.493473	1.094649	15.000000	16.000000	17.000000	17.000000	18.000000
Gender	383.0	0.516971	0.500366	0.000000	0.000000	1.000000	1.000000	1.000000
StudyTimeWeekly	383.0	9.851567	5.706828	0.025689	5.148142	9.727833	14.558504	19.916047
Absences	383.0	14.629243	8.478083	0.000000	8.000000	14.000000	22.000000	29.000000
Tutoring	383.0	0.308094	0.462310	0.000000	0.000000	0.000000	1.000000	1.000000
Extracurricular	383.0	0.360313	0.480719	0.000000	0.000000	0.000000	1.000000	1.000000
Sports	383.0	0.326371	0.469498	0.000000	0.000000	0.000000	1.000000	1.000000
Music	383.0	0.214099	0.410733	0.000000	0.000000	0.000000	0.000000	1.000000
Volunteering	383.0	0.161880	0.368822	0.000000	0.000000	0.000000	0.000000	1.000000
GPA	383.0	1.881080	0.912499	0.000000	1.141208	1.899198	2.633472	4.000000

In [185…

```
df.describe(include='object')
```

Out[185…

	Ethnicity	ParentalEducation	ParentalSupport
count	383	343	353
unique	4	4	4
top	Caucasian	Some College	Moderate
freq	197	149	127

In [221…

```
df['Ethnicity'].unique()
```

Out[221…

```
array(['Other', 'Asian', 'African American', 'Caucasian'], dtype=object)
```

In [227…

```
df['Ethnicity'].value_counts()
```

Out[227…

```
Ethnicity
Caucasian           197
African American     74
Asian                68
Other                44
Name: count, dtype: int64
```

In [223…

```
df['ParentalEducation'].unique()
```

Out[223…

```
array(['Higher', 'Bachelor', 'Some College', 'High School', nan],
      dtype=object)
```

In [229…

```
df['ParentalSupport'].unique()
```

Out[229…

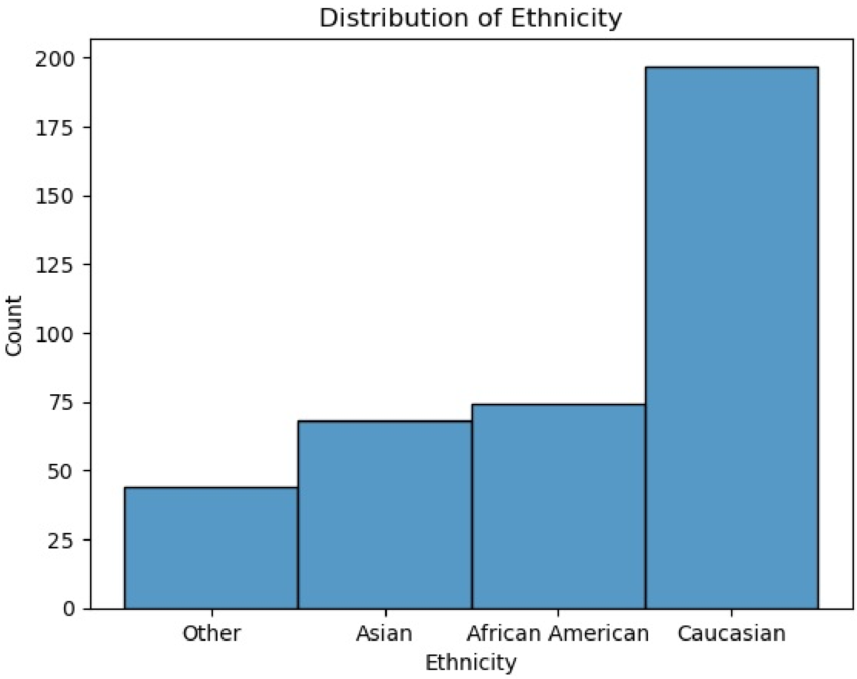
```
array(['Moderate', 'High', 'Very High', 'Low', nan], dtype=object)
```

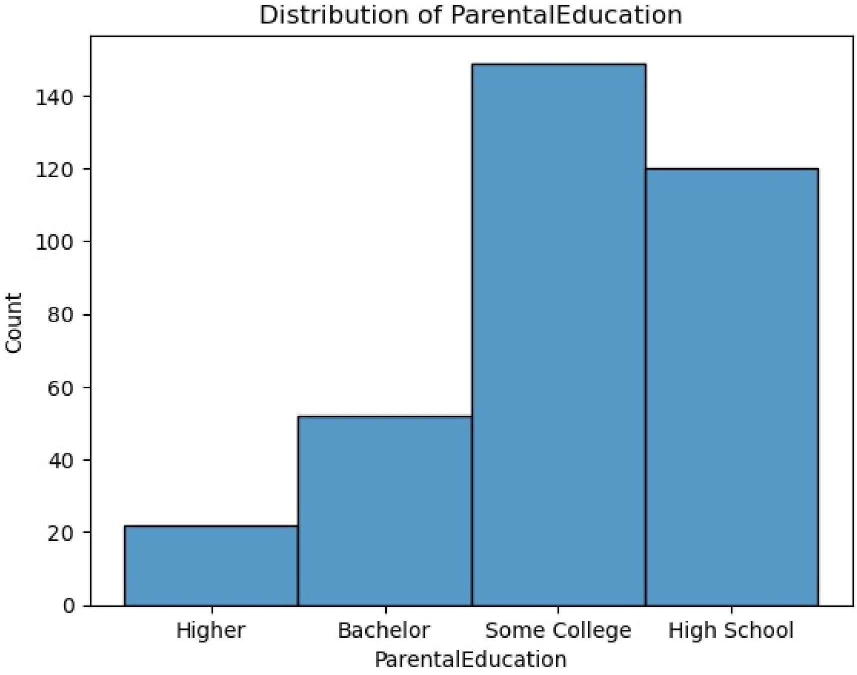
histogram to understand the distribution

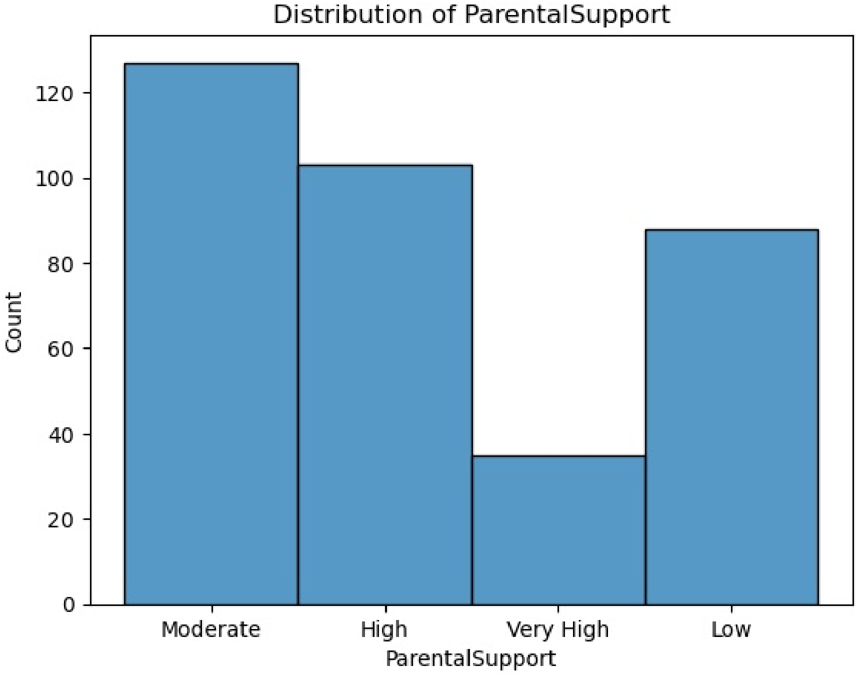
In [ ]:

In [206…

```
import warnings
warnings.filterwarnings('ignore')
for i in df.select_dtypes(include='object').columns:
    sns.histplot(data=df,x=i)
    plt.title(f"Distribution of {i}")
    plt.show()
```



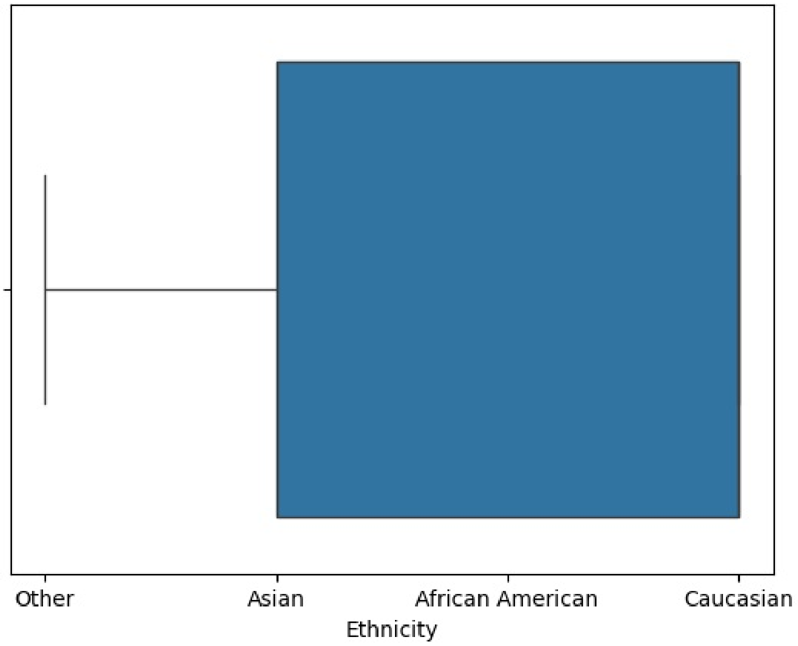


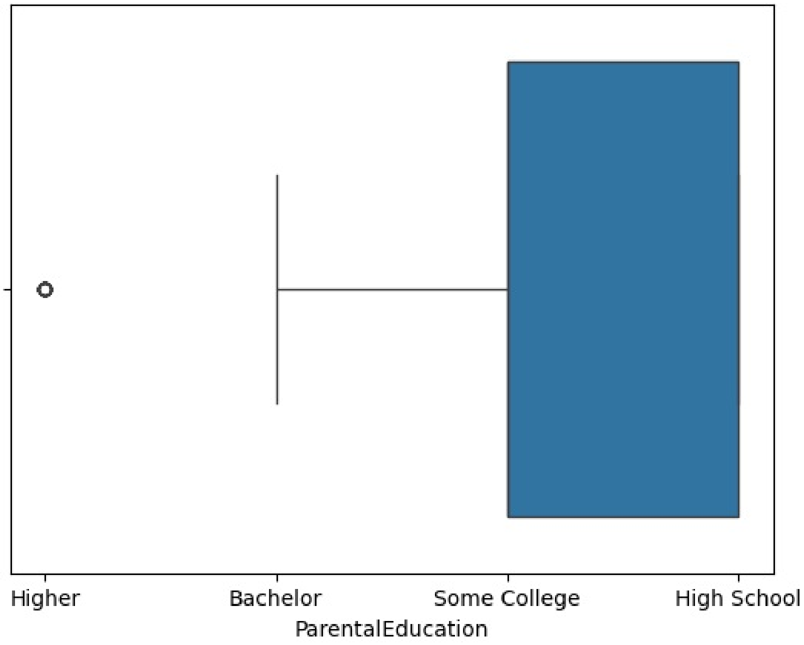


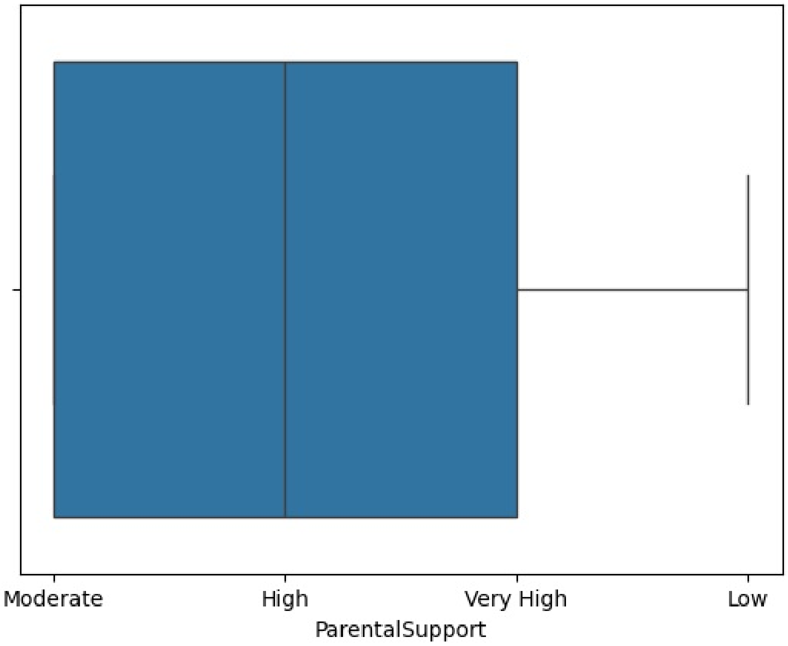
Box-plot-to identify the outliers

In [212…

```
for i in df.select_dtypes(include='object').columns:
    sns.boxplot(data=df,x=i)
    plt.show()
```







```
So,Here we don't have any outliers to detect and remove ,so we can proceed with next step
```

scatter plot to undesrand the relationships

```
Here it is used to show the relationship between the target variable and independent varaible
```

In [217…

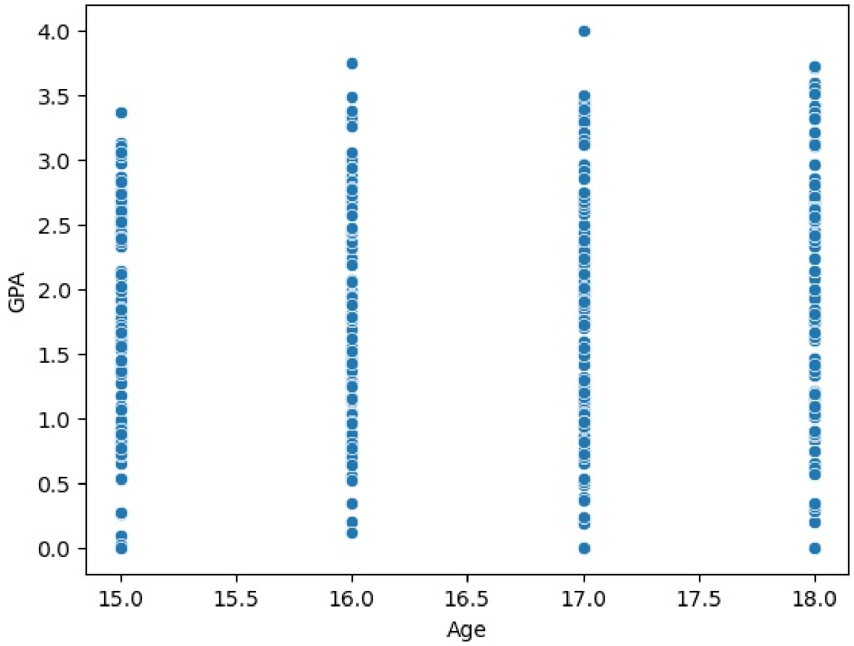
```
df.head()
```

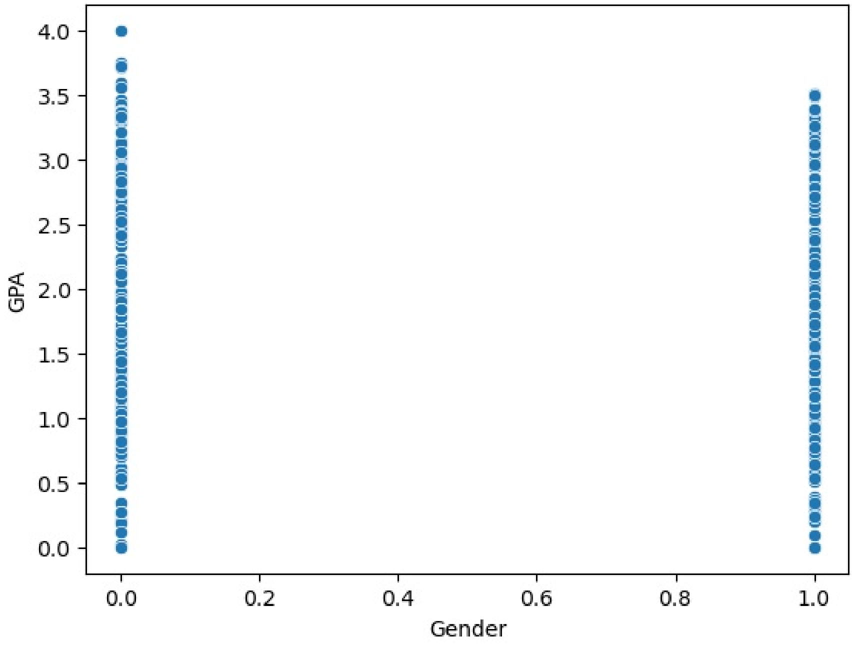
Out[217…

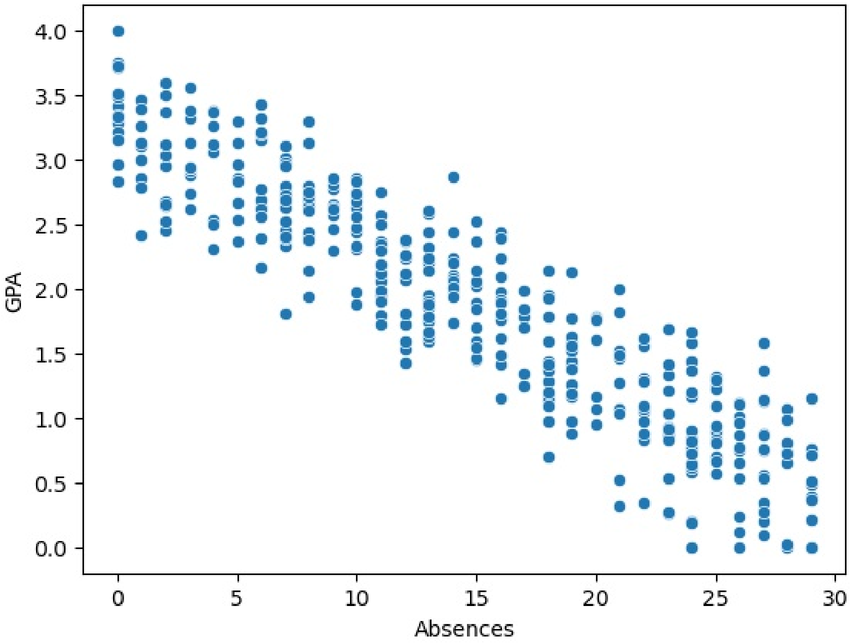
	StudentID	Age	Gender	Ethnicity	ParentalEducation	StudyTimeWeekly	Absences	Tutoring	ParentalSupport	Extracurricular
0	2340	16	1	Other	Higher	5.044048	25	1	Moderate	1
1	2923	18	0	Other	Bachelor	18.731312	12	0	Moderate	1
2	2077	16	0	Asian	Some College	0.213403	23	1	Moderate	0
3	2735	15	1	African American	Higher	14.645811	28	0	Moderate	0
4	2245	17	0	Other	Some College	11.436575	1	0	High	1

In [231…

```
for i in ['Age','Gender','Absences']:
    sns.scatterplot(data=df,x=i,y='GPA')
    plt.show()
```







correlation with heatmaps to interpret the relation and multicolliniarity

In [236…

```
s=df.select_dtypes(include='number').corr()
s
```

Out[236…

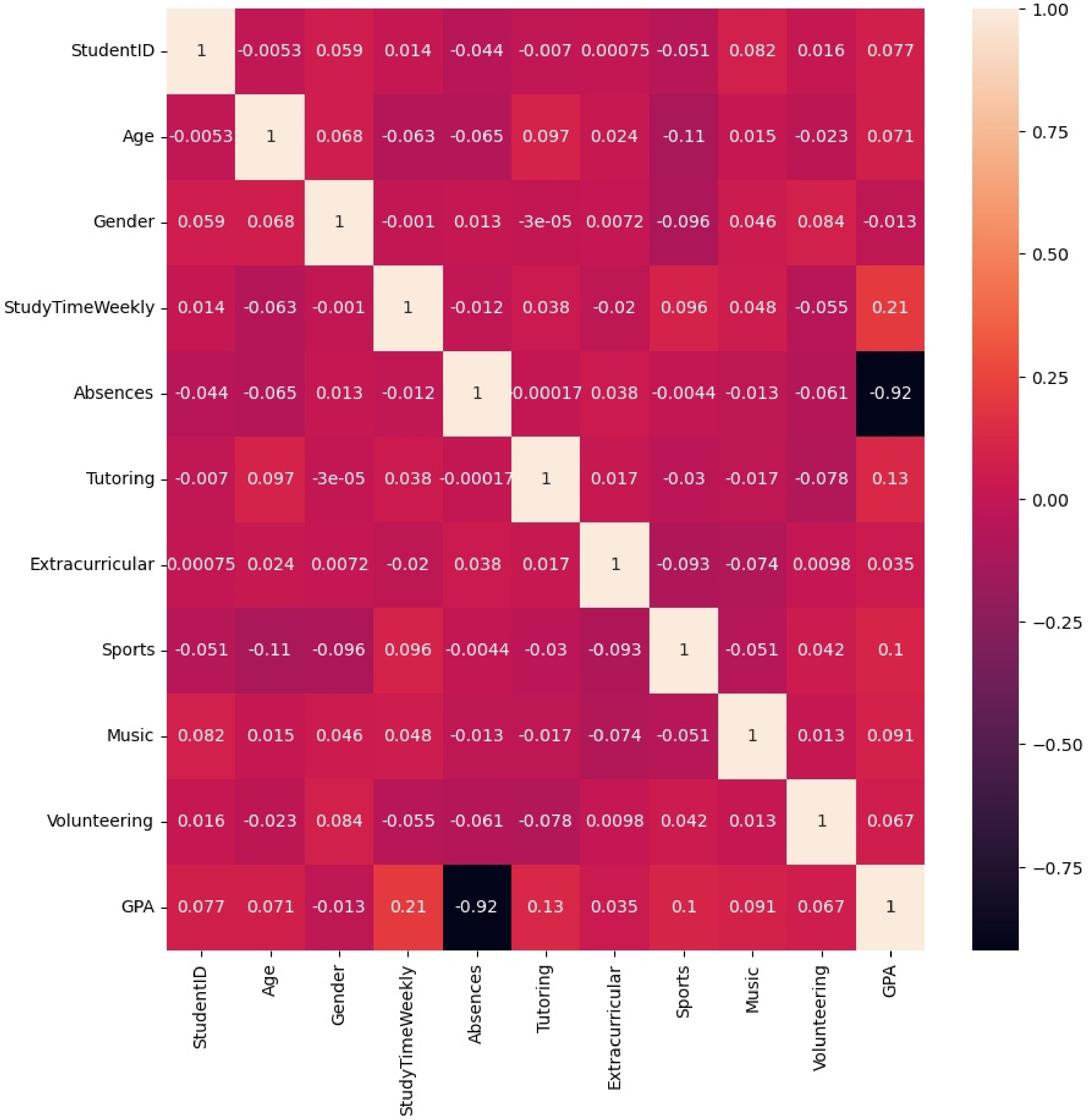
	StudentID	Age	Gender	StudyTimeWeekly	Absences	Tutoring	Extracurricular	Sports	Music	V
StudentID	1.000000	-0.005349	0.058974	0.014485	-0.044156	-0.007009	0.000749	-0.051176	0.081541	
Age	-0.005349	1.000000	0.068309	-0.062533	-0.064574	0.097095	0.024380	-0.110451	0.014761	
Gender	0.058974	0.068309	1.000000	-0.001002	0.012595	-0.000030	0.007161	-0.096071	0.045962	
StudyTimeWeekly	0.014485	-0.062533	-0.001002	1.000000	-0.011664	0.037733	-0.019670	0.095653	0.047501	
Absences	-0.044156	-0.064574	0.012595	-0.011664	1.000000	-0.000167	0.038002	-0.004377	-0.013229	
Tutoring	-0.007009	0.097095	-0.000030	0.037733	-0.000167	1.000000	0.017469	-0.030293	-0.017422	
Extracurricular	0.000749	0.024380	0.007161	-0.019670	0.038002	0.017469	1.000000	-0.093244	-0.073526	
Sports	-0.051176	-0.110451	-0.096071	0.095653	-0.004377	-0.030293	-0.093244	1.000000	-0.051075	
Music	0.081541	0.014761	0.045962	0.047501	-0.013229	-0.017422	-0.073526	-0.051075	1.000000	
Volunteering	0.015526	-0.023312	0.084370	-0.054592	-0.061125	-0.078327	0.009753	0.041801	0.012543	
GPA	0.077020	0.071287	-0.013022	0.208879	-0.920437	0.127987	0.034650	0.103424	0.091144	

In [238…

```
plt.figure(figsize=(10,10))
sns.heatmap(s,annot=True)  #annot is for showing the values on the boxes
```

Out[238…

```
<Axes: >
```



5.Missing Value treatments

In [241…

```
df.isnull().sum()
```

Out[241…

```
StudentID            0
Age                  0
Gender               0
Ethnicity            0
ParentalEducation   40
StudyTimeWeekly      0
Absences             0
Tutoring             0
ParentalSupport     30
Extracurricular      0
Sports               0
Music                0
Volunteering         0
GPA                  0
dtype: int64
```

In [251…

```
for i in ['ParentalEducation','ParentalSupport']:
    df[i].fillna(df[i].mode()[0],inplace=True)
```

In [245…

```
df.
```

Out[245…

	StudentID	Age	Gender	Ethnicity	ParentalEducation	StudyTimeWeekly	Absences	Tutoring	ParentalSupport	Extracurricular
0	2340	16	1	Other	Higher	5.044048	25	1	Moderate	1
1	2923	18	0	Other	Bachelor	18.731312	12	0	Moderate	1
2	2077	16	0	Asian	Some College	0.213403	23	1	Moderate	0
3	2735	15	1	African American	Higher	14.645811	28	0	Moderate	0
4	2245	17	0	Other	Some College	11.436575	1	0	High	1

In [253…

```
df.isnull().sum()
```

Out[253…

```
StudentID            0
Age                  0
Gender               0
Ethnicity            0
ParentalEducation    0
StudyTimeWeekly      0
Absences             0
Tutoring             0
ParentalSupport      0
Extracurricular      0
Sports               0
Music                0
Volunteering         0
GPA                  0
dtype: int64
```

In [257…

```
df['ParentalEducation'].value_counts().unique()
```

Out[257…

```
array([189, 120,  52,  22], dtype=int64)
```

In [259…

```
df['ParentalSupport'].value_counts().unique()
```

Out[259…

```
array([157, 103,  88,  35], dtype=int64)
```

Dropping Unnecessary columns

In [262…

```
df.head()
```

Out[262…

	StudentID	Age	Gender	Ethnicity	ParentalEducation	StudyTimeWeekly	Absences	Tutoring	ParentalSupport	Extracurricular
0	2340	16	1	Other	Higher	5.044048	25	1	Moderate	1
1	2923	18	0	Other	Bachelor	18.731312	12	0	Moderate	1
2	2077	16	0	Asian	Some College	0.213403	23	1	Moderate	0
3	2735	15	1	African American	Higher	14.645811	28	0	Moderate	0
4	2245	17	0	Other	Some College	11.436575	1	0	High	1

In [507…

```
df.drop(columns=['Extracurricular','Music','Volunteering'],inplace=True)
```

In [268…

```
df.head()
```

Out[268…

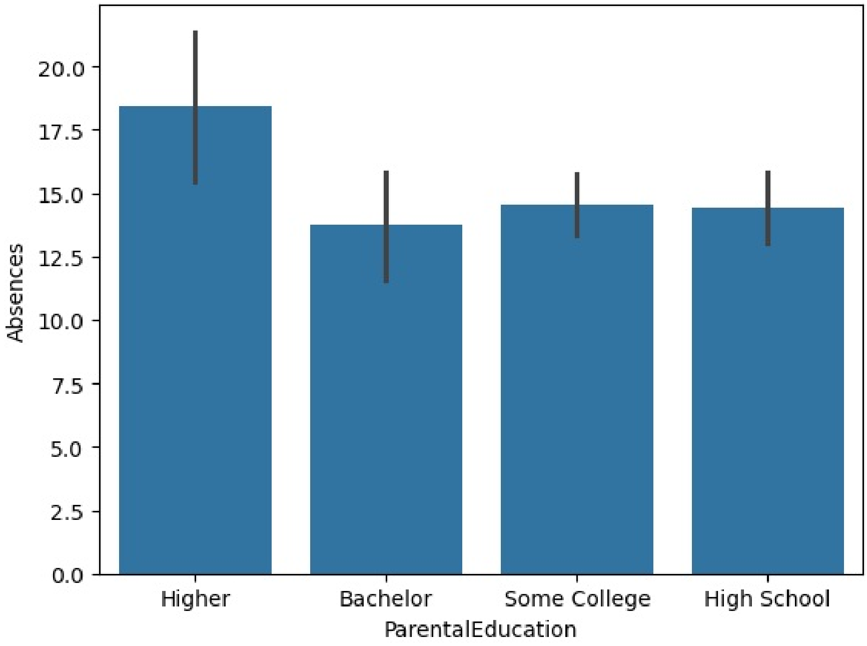
	StudentID	Age	Gender	Ethnicity	ParentalEducation	StudyTimeWeekly	Absences	Tutoring	ParentalSupport	Sports	GPA
0	2340	16	1	Other	Higher	5.044048	25	1	Moderate	0	0.886889
1	2923	18	0	Other	Bachelor	18.731312	12	0	Moderate	0	2.234696
2	2077	16	0	Asian	Some College	0.213403	23	1	Moderate	1	0.875367
3	2735	15	1	African American	Higher	14.645811	28	0	Moderate	0	0.648705
4	2245	17	0	Other	Some College	11.436575	1	0	High	0	3.463688

In [328…

```python
sns.barplot(data=df,x='ParentalEducation',y='Absences')
```

Out[328…

```
<Axes: xlabel='ParentalEducation', ylabel='Absences'>
```



In [407…

```python
# Group by 'City' and aggregate
grouped_ParentalE = df.groupby('ParentalEducation')['StudentID'].count()


grouped_ParentalE
```

Out[407…

```
ParentalEducation
Bachelor         52
High School     120
Higher           22
Some College    189
Name: StudentID, dtype: int64
```

In [403…

```python
# Group by 'City' and aggregate
grouped_Parental = df.groupby('ParentalSupport')['StudentID'].count()


grouped_Parental
```

Out[403…

```
ParentalSupport
High         103
Low           88
Moderate     157
Very High     35
Name: StudentID, dtype: int64
```

In [405…

```python
# Group by 'City' and aggregate
grouped_Ethnicity = df.groupby('Ethnicity')['StudentID'].count()


grouped_Ethnicity
```

Out[405…

```
Ethnicity
African American     74
Asian                68
Caucasian           197
Other                44
Name: StudentID, dtype: int64
```

In [409…

```python
df1=pd.DataFrame(grouped_Ethnicity)
```

```
df1
```

Out[409…

	StudentID
Ethnicity	
African American	74
Asian	68
Caucasian	197
Other	44

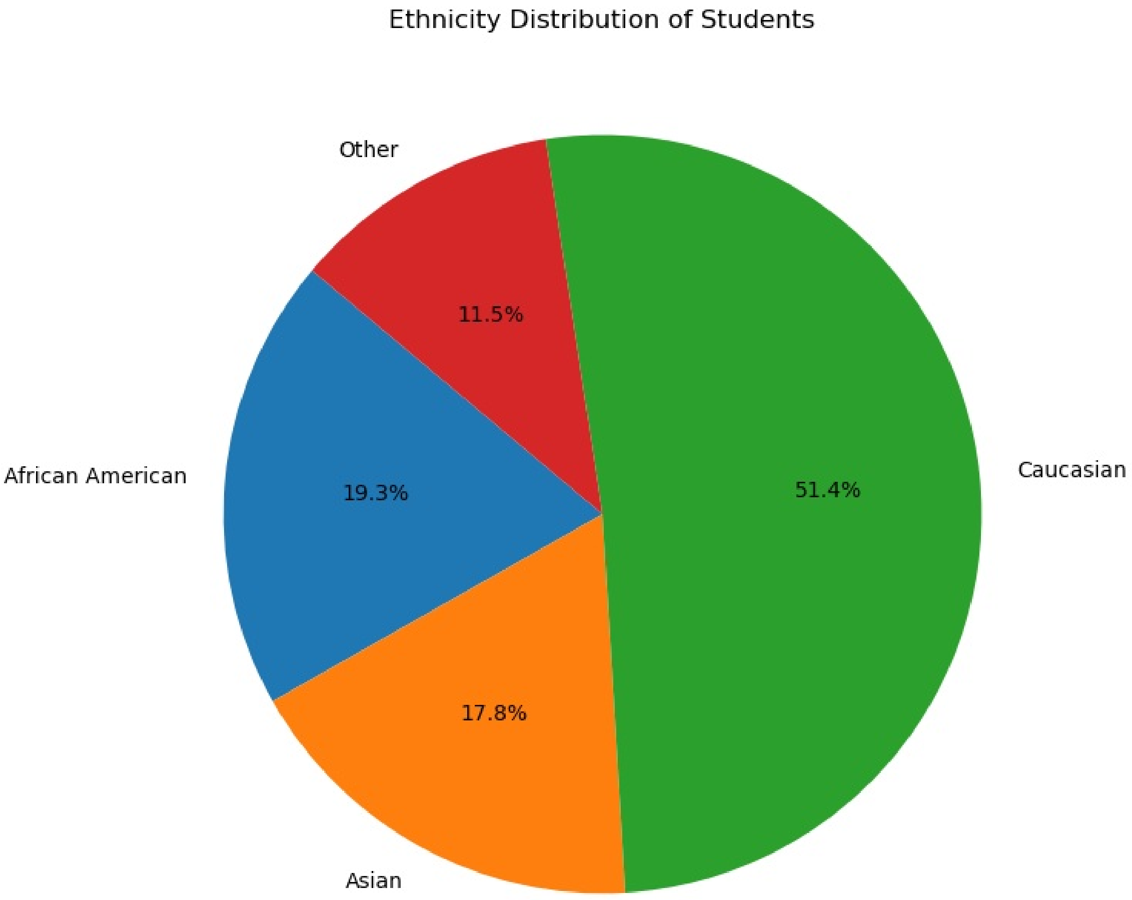
In [473…

```
# Data
ethnicities = ["African American", "Asian", "Caucasian", "Other"]
counts = [74, 68, 197, 44]

# Creating the pie chart
plt.figure(figsize=(8, 8))
plt.pie(counts, labels=ethnicities, autopct='%1.1f%%', startangle=140)

# Adding a title
plt.title("Ethnicity Distribution of Students")

# Displaying the chart
plt.show()
```



In [413…

```
df2=pd.DataFrame(grouped_ParentalE)
df2
```

Out[413…

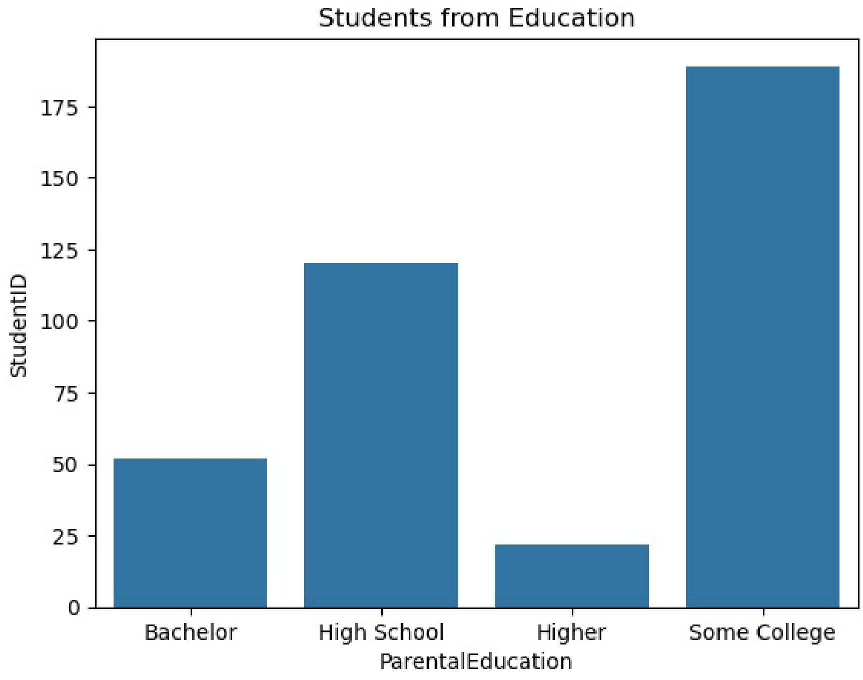
	StudentID
ParentalEducation	
Bachelor	52
High School	120
Higher	22
Some College	189

In [430…

```
sns.barplot(data=df2,x='ParentalEducation',y='StudentID')
plt.title('Students from Education')
```

Out[430…

```
Text(0.5, 1.0, 'Students from Education')
```



In [415…

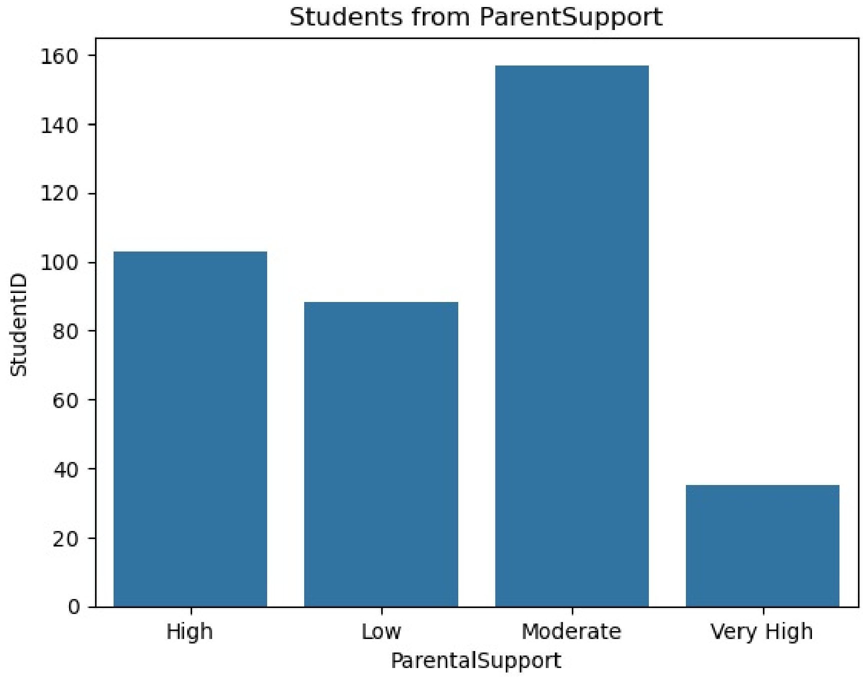
```
df3=pd.DataFrame(grouped_Parental)
df3
```

Out[415…

	StudentID
ParentalSupport	
High	103
Low	88
Moderate	157
Very High	35

In [440…

```
sns.barplot(data=df3,x='ParentalSupport',y='StudentID')
plt.title('Students from ParentSupport')
plt.show()
```



In [442…

```
def Gender(Gender):
    if Gender==0:
```

Out[442…

	StudentID	Age	Gender	Ethnicity	ParentalEducation	StudyTimeWeekly	Absences	Tutoring	ParentalSupport	Sports	GPA
0	2340	16	1	Other	Higher	5.044048	25	1	Moderate	0	0.886889
1	2923	18	0	Other	Bachelor	18.731312	12	0	Moderate	0	2.234696
2	2077	16	0	Asian	Some College	0.213403	23	1	Moderate	1	0.875367
3	2735	15	1	African American	Higher	14.645811	28	0	Moderate	0	0.648705
4	2245	17	0	Other	Some College	11.436575	1	0	High	0	3.463688

In [501…

```
df.info()
```

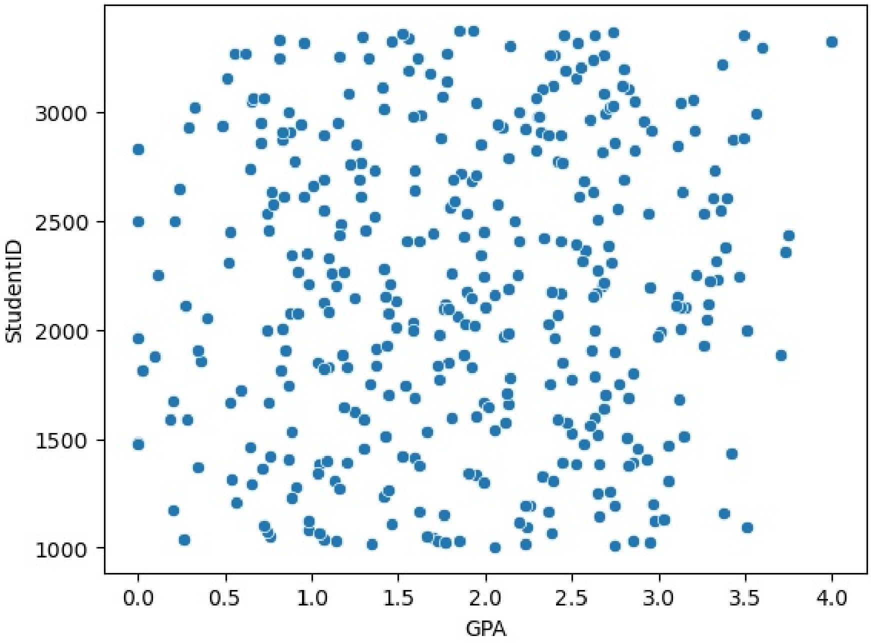
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1704 entries, 0 to 1703
Data columns (total 8 columns):
 #   Column     Non-Null Count  Dtype  
---  ------     --------------  -----  
 0   country    1704 non-null   object 
 1   continent  1704 non-null   object 
 2   year       1704 non-null   int64  
 3   lifeExp    1704 non-null   float64
 4   pop        1704 non-null   int64  
 5   gdpPercap  1704 non-null   float64
 6   iso_alpha  1704 non-null   object 
 7   iso_num    1704 non-null   int64  
dtypes: float64(2), int64(3), object(3)
memory usage: 106.6+ KB
```

In [527…

```
sns.scatterplot(data=df,x='GPA',y='StudentID')
```

Out[527…

```
<Axes: xlabel='GPA', ylabel='StudentID'>
```



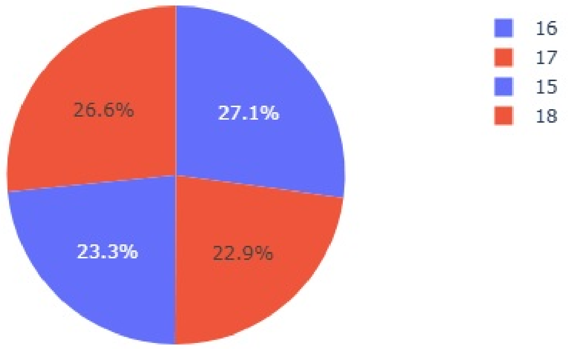
In [539…

```
fig=px.pie(df,names='Age',values='StudentID',color='Gender')
fig.show()
```

In [537…

```
from PIL import Image
Image.open('newplot.png')
```

Out[537…



In [ ]:

In [ ]:

In [ ]:

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js