FIT1043 A1 KAN ROU YI 34661093

In [713…
```python
import pandas as pd
from matplotlib import pyplot as plt
import matplotlib.pyplot as plt
```

In [714…
```python
df = pd.read_csv('Student_List.csv')
df
```

Out[714…

	StudentID	Age	ParentalEducation	StudyTimeWeekly	Absences	Tutoring	ParentalSupport	Extracurricular	Sports	Music	Volunteering	GPA	GradeClass
0	1640	18	Some College	10.318918	5	No	2	No	Yes	No	No	2.655994	C
1	2939	16	Bachelor's	6.517803	2	Yes	2	No	Yes	No	No	3.474562	B
2	2877	15	Some College	0.815700	1	No	1	No	No	Yes	Yes	2.806878	C
3	1628	16	High School	6.304335	8	No	3	No	No	No	No	2.150546	D
4	2052	15	Some College	2.516047	14	Yes	3	Yes	No	No	No	2.253871	D
...	...	...	...	...	...	...	...	...	...	...	...	...	...
1495	1944	15	No Education	10.596678	12	No	1	No	No	No	Yes	1.621012	F
1496	1566	16	Some College	3.278634	4	Yes	4	Yes	No	No	No	3.244882	B
1497	3268	18	Some College	7.598010	0	Yes	3	No	No	No	No	3.040730	F
1498	3237	17	Some College	15.078754	24	No	4	No	No	Yes	No	1.245091	F
1499	1149	15	Higher	1.360205	22	No	1	Yes	Yes	No	Yes	1.007226	F

1500 rows × 13 columns

A1. Exploring the dataset

1. How many students are there in this dataset?

In [717…
```python
num_students = df.shape[0]
print(f'Total number of students: {num_students}')
```

```
Total number of students: 1500
```

2. What is the age range of students in this dataset?

In [719…
```python
df.describe()
```

Out[719…

	StudentID	Age	StudyTimeWeekly	Absences	ParentalSupport	GPA
count	1500.000000	1500.000000	1500.000000	1500.000000	1500.000000	1500.000000
mean	2209.142667	16.464000	9.625516	14.540000	2.131333	1.901045
std	693.068389	1.124372	5.718608	8.459557	1.116654	0.923638
min	1002.000000	15.000000	0.001057	0.000000	0.000000	0.000000
25%	1599.750000	15.000000	4.717155	7.000000	1.000000	1.171847
50%	2211.500000	16.000000	9.362636	15.000000	2.000000	1.881249
75%	2804.750000	17.000000	14.374170	22.000000	3.000000	2.623266
max	3389.000000	18.000000	19.978094	29.000000	4.000000	4.000000

In [809…
```python
max = df['Age'].max()
min = df['Age'].min()
range = max - min
print('Range:',range)
```

```
Range: 3
```

3. Identify and discuss the data types of each column in the dataset.

In [722…
```python
df.dtypes
```

Out[722…
```
StudentID              int64
Age                    int64
ParentalEducation     object
StudyTimeWeekly      float64
Absences               int64
Tutoring              object
ParentalSupport        int64
Extracurricular       object
Sports                object
Music                 object
Volunteering          object
GPA                  float64
GradeClass            object
dtype: object
```

4. GradeClass corresponds to different categories of grades, where, for example, 'A' corresponds to the highest grade (GPA ≥ 3.5) and 'F' corresponds to the lowest grade (GPA < 2.0). Calculate the percentage of students in each grade category.

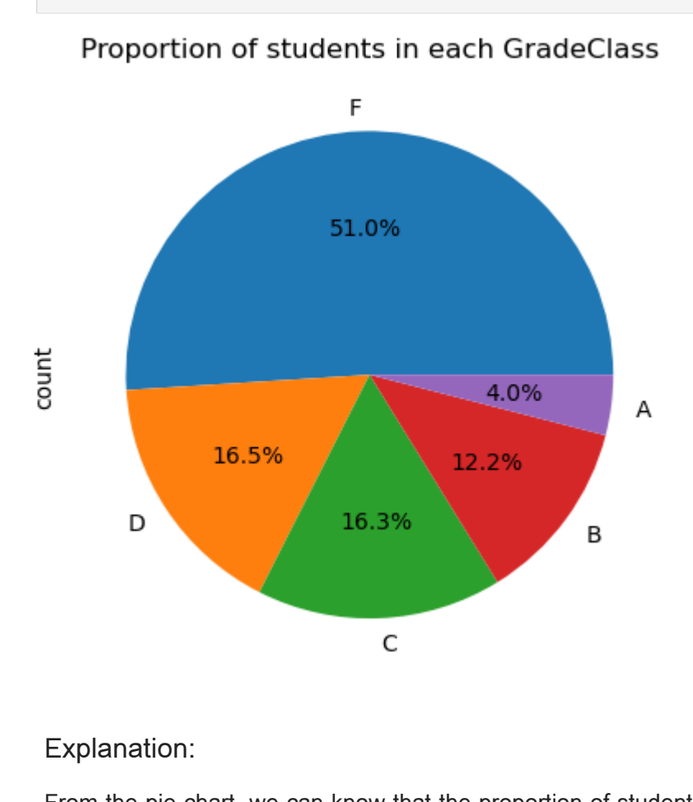
In [724…
```python
grade_counts = df['GradeClass'].value_counts()
total_students = len(df)
grade_percentages = (grade_counts / total_students) * 100
grade_percentages
```

Out[724…
```
GradeClass
F    51.000000
D    16.533333
C    16.266667
B    12.200000
A     4.000000
Name: count, dtype: float64
```

5. Create a pie chart to show the proportion of students in each GradeClass and discuss your observations.

In [726…
```python
import matplotlib.pylab as plt
%matplotlib inline
```

In [727…
```python
df['GradeClass'].value_counts().plot.pie(autopct = '%1.1f%%')
plt.title('Proportion of students in each GradeClass')
plt.show()
```



Explanation:

From the pie chart, we can know that the proportion of students in each grade class.

F - 51.0% So, the majority of studets are in grade class F.

D - 16.5% in grade class D

C - 16.3% in grade class C

B - 12.2% in grade class B

A - 4.0% in grade class A

In conclusion ,most of the studetns are failing and only few achieving high grades of A and B

A2. Exploring Parental Education

1. Determine the frequency of different levels of parental education.

- How many parents have the highest education level (denoted by Higher) in this dataset?
- How many have no education?
- What is the most common level of parental education?

In [731…
```python
#read data from csv file
df = pd.read_csv('Student_List.csv')
df.ParentalEducation
df
```

Out[731…

	StudentID	Age	ParentalEducation	StudyTimeWeekly	Absences	Tutoring	ParentalSupport	Extracurricular	Sports	Music	Volunteering	GPA	GradeClass
0	1640	18	Some College	10.318918	5	No	2	No	Yes	No	No	2.655994	C
1	2939	16	Bachelor's	6.517803	2	Yes	2	No	Yes	No	No	3.474562	B
2	2877	15	Some College	0.815700	1	No	1	No	No	Yes	Yes	2.806878	C
3	1628	16	High School	6.304335	8	No	3	No	No	No	No	2.150546	D
4	2052	15	Some College	2.516047	14	Yes	3	Yes	No	No	No	2.253871	D
...	...	...	...	...	...	...	...	...	...	...	...	...	...
1495	1944	15	No Education	10.596678	12	No	1	No	No	No	Yes	1.621012	F
1496	1566	16	Some College	3.278634	4	Yes	4	Yes	No	No	No	3.244882	B
1497	3268	18	Some College	7.598010	0	Yes	3	No	No	No	No	3.040730	F
1498	3237	17	Some College	15.078754	24	No	4	No	No	Yes	No	1.245091	F
1499	1149	15	Higher	1.360205	22	No	1	Yes	Yes	No	Yes	1.007226	F

1500 rows × 13 columns

In [732…
```python
#value counts functions to count
highest_education_level = df['ParentalEducation'].value_counts().get('Higher')
print('Highest education level:',highest_education_level)
```

```
Highest education level: 77
```

In [815…
```python
no_education = df['ParentalEducation'].value_counts().get('No Education')
print('No Education:',no_education)
```

```
No Education: None
```

In [734…
```python
#use mode function
most_common_education_level = df['ParentalEducation'].mode()
print('Most common education level:',most_common_education_level)
```

```
Most common education level: 0    Some College
Name: ParentalEducation, dtype: object
```

2.Replace the values in the 'ParentalEducation' column according to following table:

In [804…
```python
#create a dictionary
column = {
    'No Education': 0,
    'High School' : 1,
    'Some College': 2,
    "Bachelor's": 3,
    'Higher': 4
}

#use replace function
df['ParentalEducation'] = df['ParentalEducation'].replace(column)
df
```

Out[804…

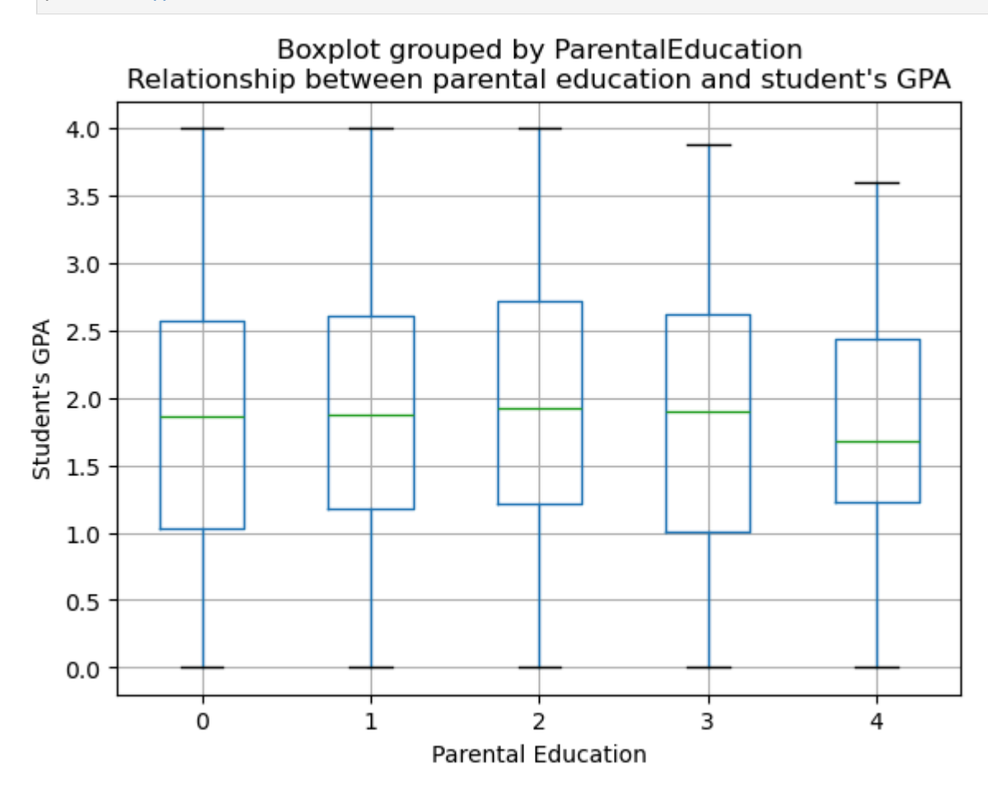
	StudentID	Age	ParentalEducation	StudyTimeWeekly	Absences	Tutoring	ParentalSupport	Extracurricular	Sports	Music	Volunteering	GPA	GradeClass
0	1640	18	2	10.318918	5	No	2	No	Yes	No	No	2.655994	C
1	2939	16	3	6.517803	2	Yes	2	No	Yes	No	No	3.474562	B
2	2877	15	2	0.815700	1	No	1	No	No	Yes	Yes	2.806878	C
3	1628	16	1	6.304335	8	No	3	No	No	No	No	2.150546	D
4	2052	15	2	2.516047	14	Yes	3	Yes	No	No	No	2.253871	D
...	...	...	...	...	...	...	...	...	...	...	...	...	...
1495	1944	15	0	10.596678	12	No	1	No	No	No	Yes	1.621012	F
1496	1566	16	2	3.278634	4	Yes	4	Yes	No	No	No	3.244882	B
1497	3268	18	2	7.598010	0	Yes	3	No	No	No	No	3.040730	F
1498	3237	17	2	15.078754	24	No	4	No	No	Yes	No	1.245091	F
1499	1149	15	4	1.360205	22	No	1	Yes	Yes	No	Yes	1.007226	F

1500 rows × 13 columns

3. Use a boxplot to visualise the distribution of GPA for each level of ParentalEducation.

Looking at the boxplot, what can you say about the relationship between parental education and the students' GPA?

In [738…
```python
df.boxplot(column ='GPA', by = 'ParentalEducation')
plt.xlabel('Parental Education')
plt.ylabel("Student's GPA")
plt.title("Relationship between parental education and student's GPA")
plt.show()
```



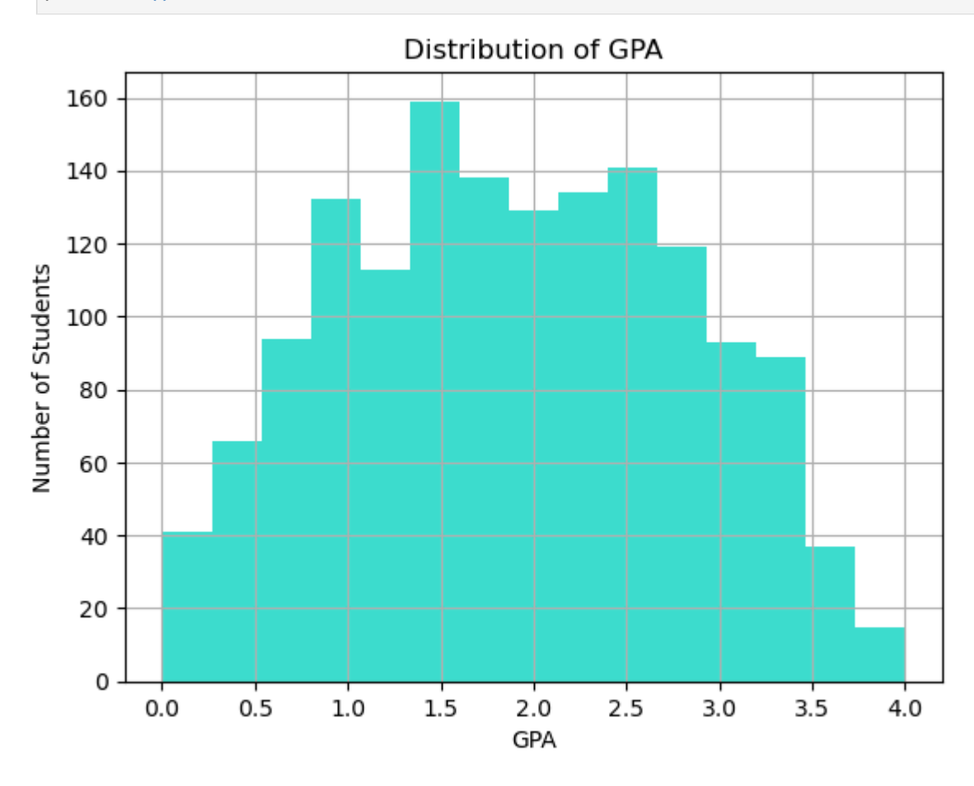
Explanation:

The boxplot shows that there isn't a clear or significant correlation between students' GPA and parental education. From the interquartile range in boxplot , we can know that 50% of students' GPAs are spread out similarly, regardless of their parents' education level.The students' GPA distributions are rather comparable regardless of whether their parents have no education,high school, or some college , bachelor or higher . This suggests that the level of education of a parent may not be a really important factor in influencing a student's GPA.

A3. GPA distribution and Correlation Analysis

1.Create a histogram to visualise the distribution of GPA. Summarise your observations.

In [743…
```python
df.GPA.hist(bins=15 , color = "turquoise")
plt.xlabel('GPA')
plt.ylabel('Number of Students')
plt.title('Distribution of GPA')
plt.show()
```

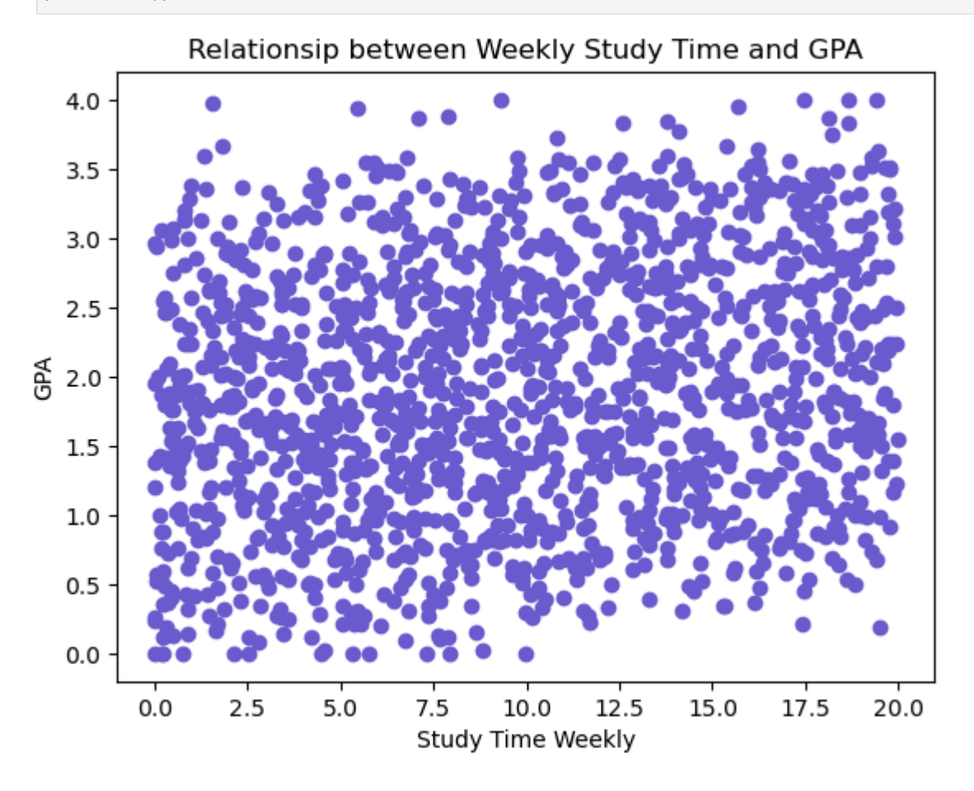


Explanation:

This histogram shows that majority of students have GPA with lower to middle range which is 1.0 - 3.0 ,relatively students have high performance of 3.5 - 4.0 , and a small number are underperforming which is 0.0 - 1.5.

2.Does a higher weekly study time correlate with better GPA? Use a scatter plot to visualise this relationship and calculate the correlation coefficient.

In [746…
```python
plt.scatter(df['StudyTimeWeekly'], df['GPA'] , color = "slateblue")
plt.xlabel('Study Time Weekly')
plt.ylabel('GPA')
plt.title('Relationsip between Weekly Study Time and GPA')
plt.show()
```

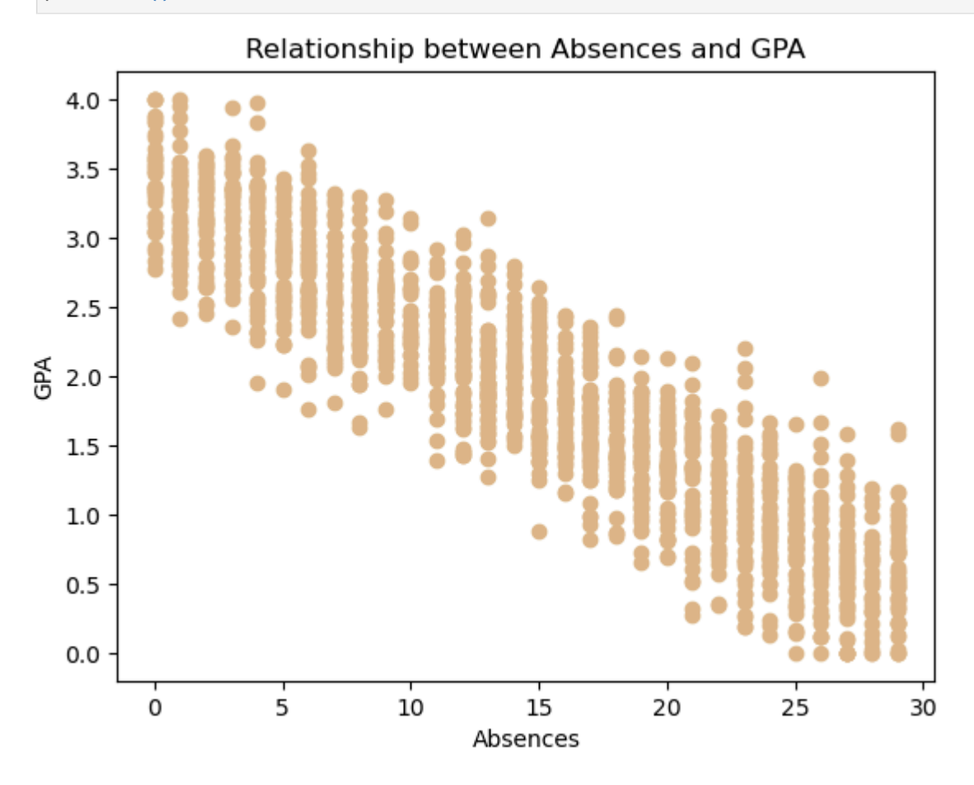


In [823…
```python
#use .corr() function
correct_correlation = df['StudyTimeWeekly'].corr(df['GPA'])
print('Correlation Coefficient:',correct_correlation)
```

```
Correlation Coefficient: 0.19049313037112514
```

3. How about any correlation between absences and GPA? Use a scatter plot to visualise this relationship and calculate the correlation coefficient.

In [749…
```python
plt.scatter(df['Absences'], df['GPA'] , color = "burlywood")
plt.xlabel('Absences')
plt.ylabel('GPA')
plt.title('Relationship between Absences and GPA')
plt.show()
```



In [826…
```python
correct_correlation = df['Absences'].corr(df['GPA'])
print('Correlation Coefficient:',correct_correlation)
```

```
Correlation Coefficient: -0.9194876943290949
```

A4.Extracurricular Activities

1. Select a list of students who are involved in Sports, Music, Volunteering and Extracurricular activities. How many students are there? Assume this is called group A

In [857…
```python
#use '&' to make sure all the criteria is selected
# 'Yes' means that the student involved in the activity
group_A = df[(df['Sports'] == 'Yes') & (df['Music'] == 'Yes') & (df['Volunteering'] == 'Yes') & (df['Extracurricular'] == 'Yes')]
#determine the length of group A
num_students = len(group_A)
print('Number of students in Group A:', num_students)
```

```
Number of students in Group A: 5
```

2. Now Select students who did not involve in any of the above activities. How many students are there? Assume this is called group B.

In [860…
```python
# 'No' means that the student does not involved in the activity
group_B = df[(df['Sports'] == 'No') & (df['Music'] == 'No') & (df['Volunteering'] == 'No') & (df['Extracurricular'] == 'No')]
num_students = len(group_B)
print('Number of students in Group B:', num_students)
```

```
Number of students in Group B: 432
```

3. Compare the mean GPA of students in group A versus group B. What can you say?

In [863…
```python
#use mean() function
mean_GPA_A = group_A['GPA'].mean()
mean_GPA_B = group_B['GPA'].mean()
print('GPA of students in Group A:',mean_GPA_A)
print('GPA of students in Group B:',mean_GPA_B)
```

```
GPA of students in Group A: 2.4475262217999996
GPA of students in Group B: 1.7312226005532407
```

Explanation:

The mean of GPA of students in group A is 2.45,while mean of GPA of students in group B is 1.73. This means that students of group A are performing better than students of group B.

A5. Exploring Parental Support and Tutoring

1. Aggregate the data by ParentalSupport and find the mean and median GPA for each group. Also, find the number of students who are 18 years old in each group..

In [868…
```python
#groupby the data frame
#select 'GPA' for aggregation
# use agg() function
new_data = df.groupby('ParentalSupport')['GPA'].agg(['mean', 'median'])
print(new_data)
```

```
                     mean    median
ParentalSupport
0                1.521602  1.471672
1                1.735855  1.740455
2                1.845914  1.817007
3                2.068174  2.070669
4                2.227639  2.215516
```

In [872…
```python
#use rename() function
#use index and dictionary to map the old value to new labels
new_data.rename(
    index ={
        0: 'None',
        1: 'Low',
        2: 'Moderate',
        3: 'High',
        4: 'Very High'
    },
    inplace=True
)
new_data
```

Out[872…

ParentalSupport	mean	median
None	1.521602	1.471672
Low	1.735855	1.740455
Moderate	1.845914	1.817007
High	2.068174	2.070669
Very High	2.227639	2.215516

In [874…
```python
#choose the data frame of age that includes onl 18
students_18 = df[df['Age'] == 18]
#use size() function counts the number
students_18_count = students_18.groupby('ParentalSupport').size()
print(students_18_count)
```

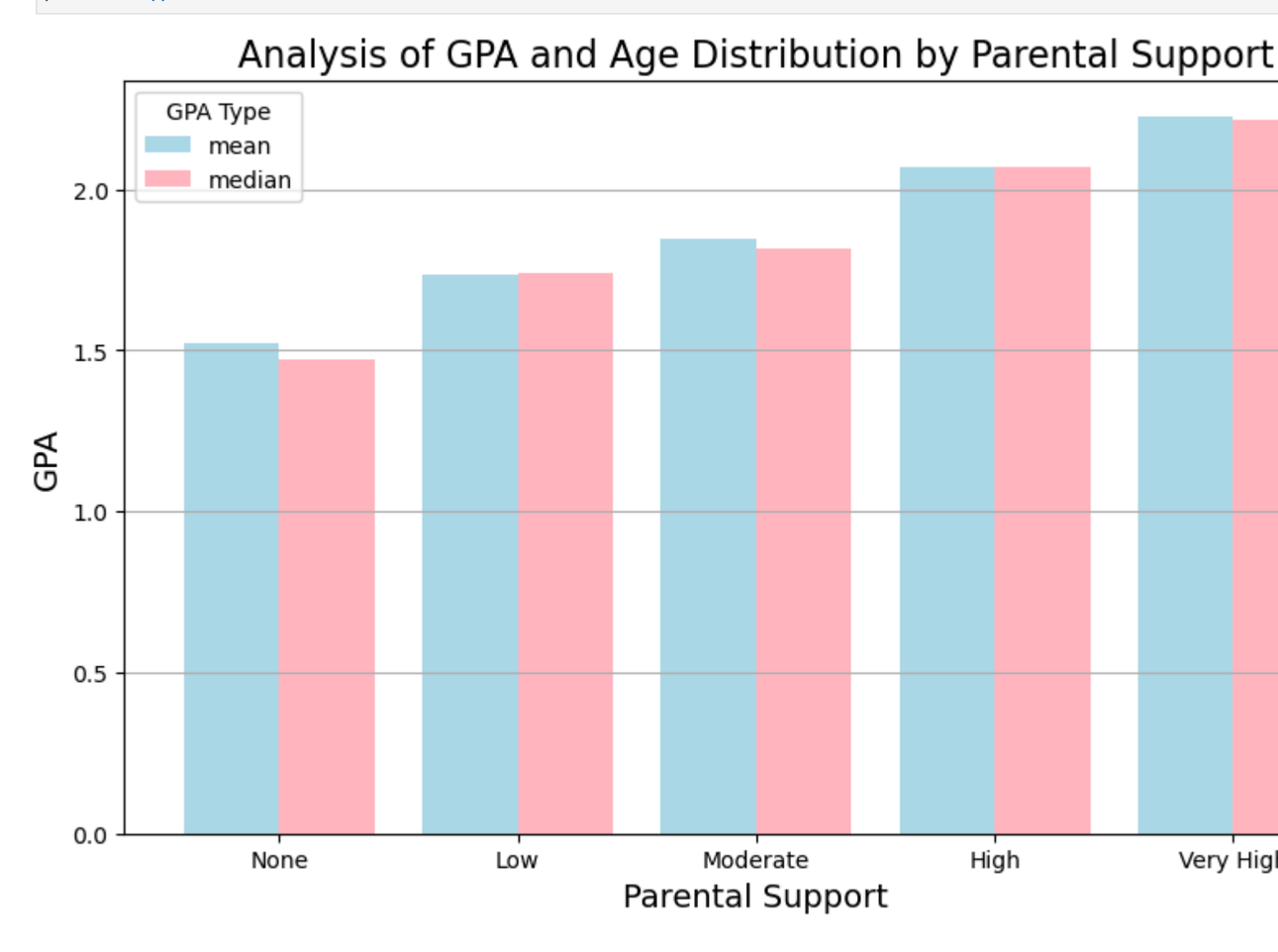
```
ParentalSupport
0     33
1     80
2     98
3    116
4     35
dtype: int64
```

In [876…
```python
#use lambda x:(x == 18).sum() to count number of students who are 18 in each group
grouped_data = df.groupby('ParentalSupport').agg({
    'GPA': ['mean', 'median'],
    'Age': lambda x: (x == 18).sum()
})

grouped_data.columns = ['Mean GPA', 'Median GPA', 'Number of 18-Year-Old Students']

print(grouped_data)
```

```
                 Mean GPA  Median GPA  Number of 18-Year-Old Students
ParentalSupport
0                1.521602    1.471672                              33
1                1.735855    1.740455                              80
2                1.845914    1.817007                              98
3                2.068174    2.070669                             116
4                2.227639    2.215516                              35
```

2. Plot a side by side bar chart to visualise the mean and median GPA in each level of parental support and discuss your observations. Your bar chart should illustrate each level of parental support as below.

In [879…
```python
new_data.plot(kind='bar', figsize=(10, 6), width=0.8 ,color = {"mean" :"lightblue" , "median" : "lightpink"})
plt.title('Analysis of GPA and Age Distribution by Parental Support', fontsize=16)
plt.xlabel('Parental Support', fontsize=14)
plt.ylabel('GPA', fontsize=14)
plt.xticks(rotation=0)
plt.legend(title='GPA Type', loc='upper left')
plt.grid(True, axis='y')
plt.show()
```



Explanation:

This barchart provides a visual summary of how parental supports affect GPA. When the mean and median of GPA is high, the parental support is high. While the mean and median of GPA is low, the parental support is low.This shows that parental support is very important for students GPA.

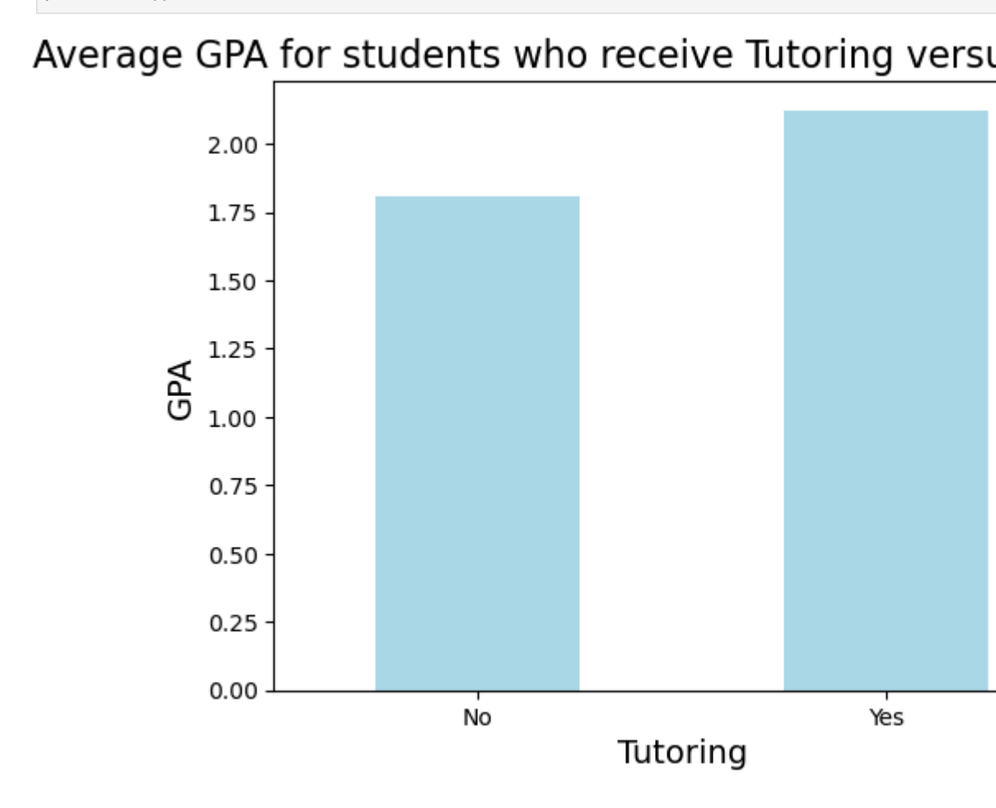
3. Calculate the average GPA for students who receive Tutoring versus those who don't.

Visualise the results using a barchart and explain what you observe.

In [881…
```python
average_gpa = df.groupby('Tutoring')['GPA'].mean()
average_gpa
```

Out[881…
```
Tutoring
No     1.804967
Yes    2.123106
Name: GPA, dtype: float64
```

In [883…
```python
average_gpa.plot.bar(x = 'Tutoring' , y = 'GPA' ,color ="lightblue",rot=0)
plt.title("Average GPA for students who receive Tutoring versus those who don't", fontsize=16)
plt.xlabel("Tutoring", fontsize=14)
plt.ylabel('GPA', fontsize=14)
plt.show()
```



Explanation:

This barchart shows the average GPA for students who receiving Tutoring versus those who dont. The barchart prove that the students receive tutoring will leads to a higher average GPA. Which means tutoring will help students to get better academic performance.