In [186…

```python
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from scipy.stats import ttest_ind
```

In [187…

```python
data = pd.read_csv('student-por.csv',sep=';')
data
```

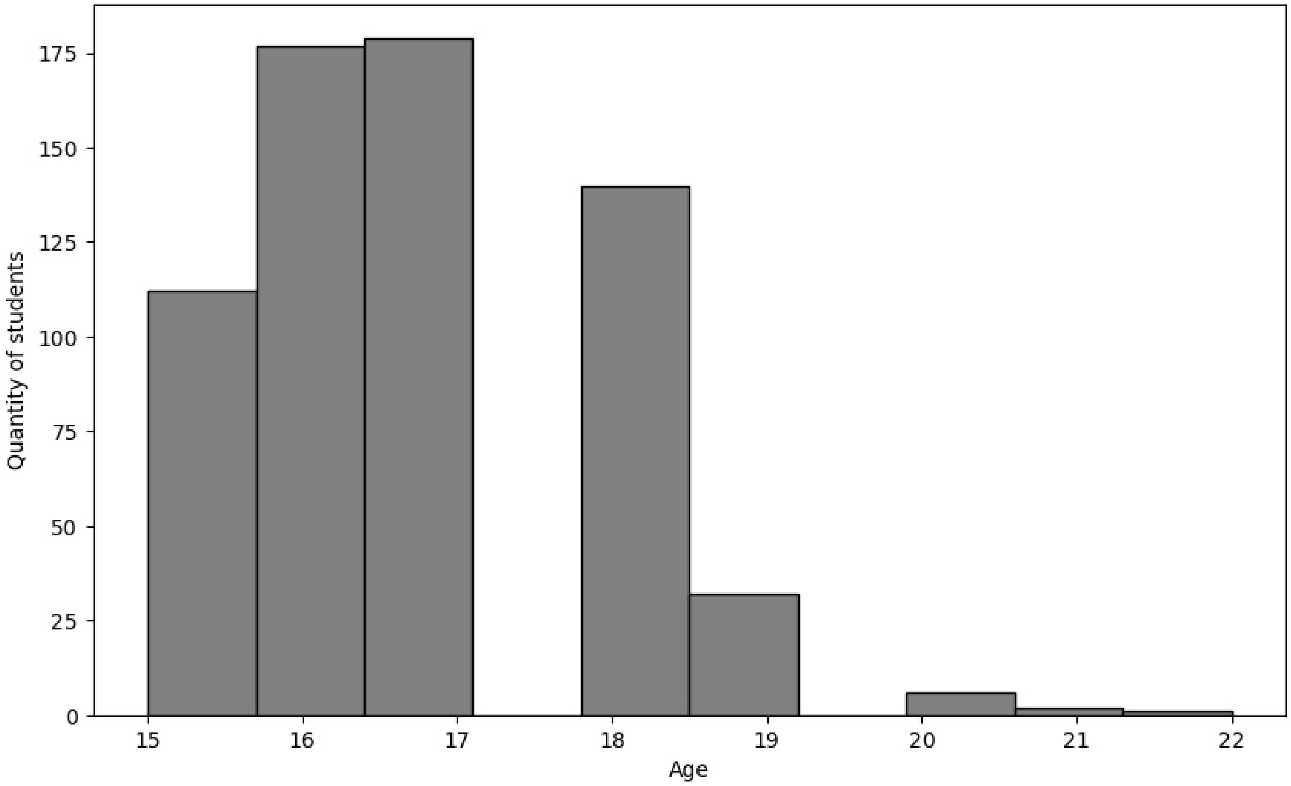
Out[187]:

	school	sex	age	address	famsize	Pstatus	Medu	Fedu	Mjob	Fjob	...	famrel	freetime	goout	Dalc	Walc	health	absence
0	GP	F	18	U	GT3	A	4	4	at_home	teacher	...	4	3	4	1	1	3	
1	GP	F	17	U	GT3	T	1	1	at_home	other	...	5	3	3	1	1	3	
2	GP	F	15	U	LE3	T	1	1	at_home	other	...	4	3	2	2	3	3	
3	GP	F	15	U	GT3	T	4	2	health	services	...	3	2	2	1	1	5	
4	GP	F	16	U	GT3	T	3	3	other	other	...	4	3	2	1	2	5	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
644	MS	F	19	R	GT3	T	2	3	services	other	...	5	4	2	1	2	5	
645	MS	F	18	U	LE3	T	3	1	teacher	services	...	4	3	4	1	1	1	
646	MS	F	18	U	GT3	T	1	1	other	other	...	1	1	1	1	1	5	
647	MS	M	17	U	LE3	T	3	1	services	services	...	2	4	5	3	4	2	
648	MS	M	18	R	LE3	T	3	2	services	other	...	4	4	1	3	4	5	

649 rows × 33 columns

In [188…

```python
#1
plt.figure(figsize = (10,6))
data['age'].hist(color = 'grey',edgecolor = 'black',bins = 10)
plt.xlabel('Age')
plt.ylabel('Quantity of students')
plt.grid(False)
plt.show()
```



In [189…

```python
#2
GP = data[data['school']=='GP']
MS = data[data['school']=='MS']
print(f"Number of GP students:{len(GP)}")
print(f"Number of MS students:{len(MS)}")
```

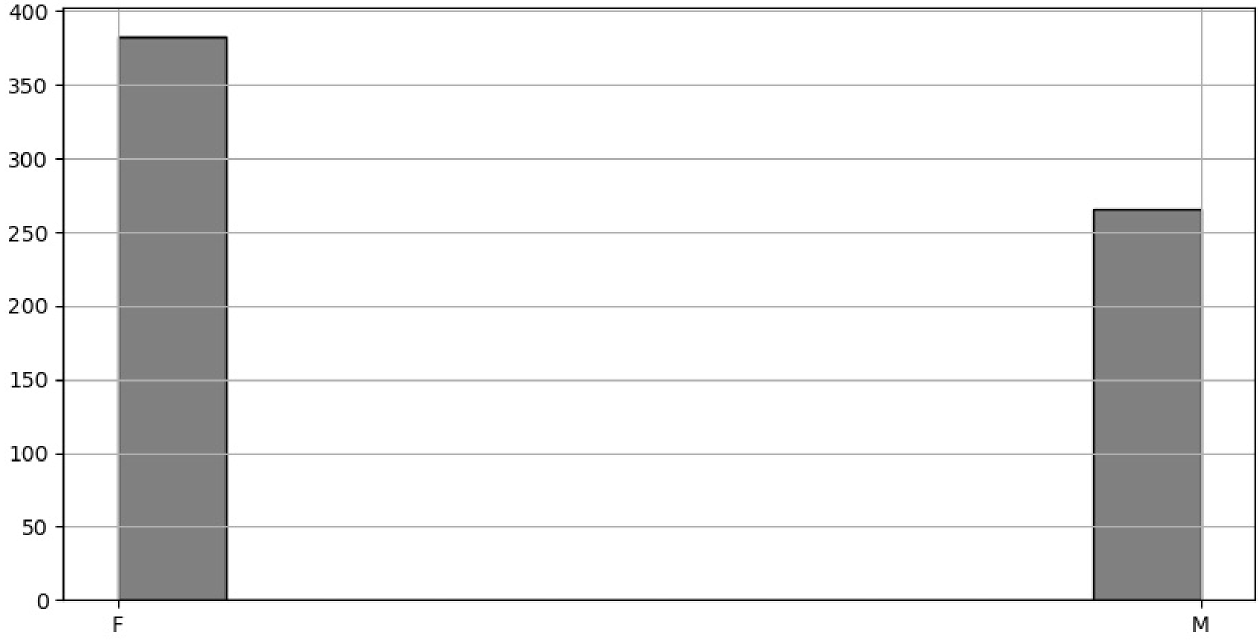
```
Number of GP students:423
Number of MS students:226
```

In [190…

```python
#3
print(data['sex'].value_counts())
plt.figure(figsize = (10,5))
```

```
data['sex'].hist(color = 'grey',edgecolor = 'black',bins = 10)
plt.show()
```

```
F    383
M    266
Name: sex, dtype: int64
```



In [191…

```
data.iloc[:,8:24]
```

Out[191]:

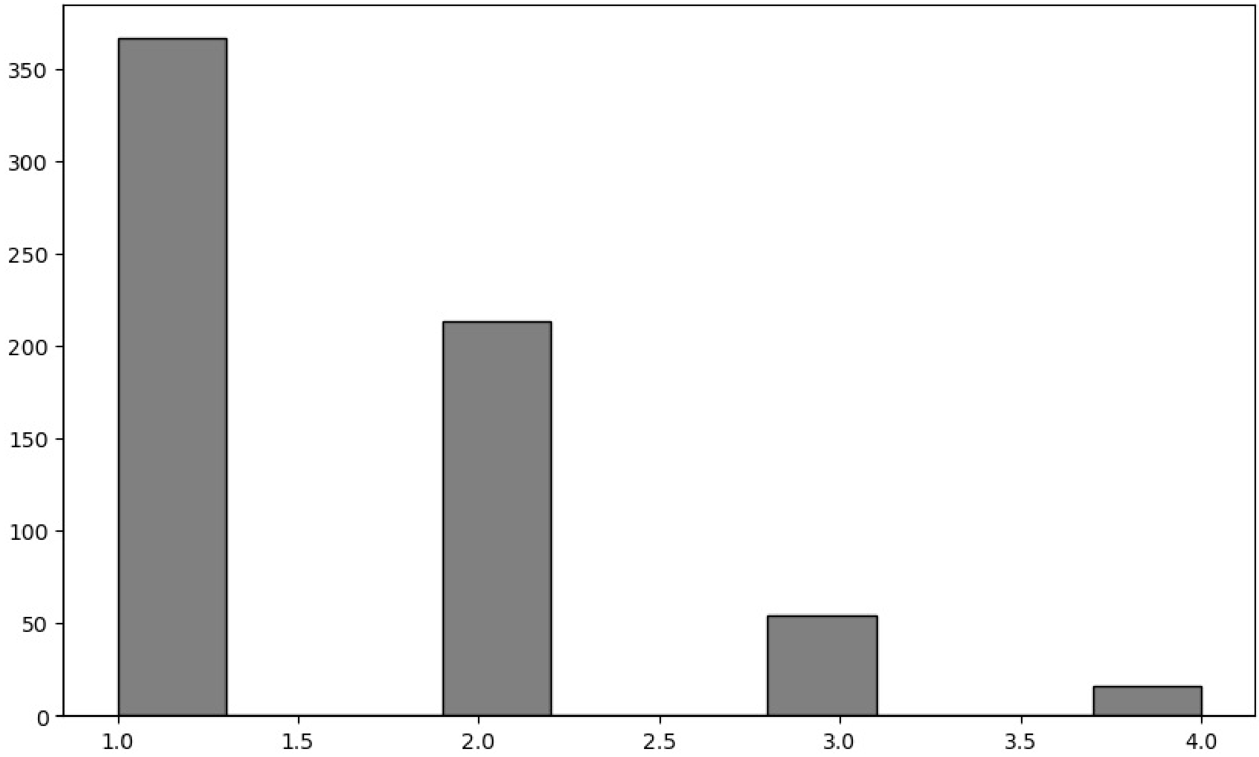
	Mjob	Fjob	reason	guardian	traveltime	studytime	failures	schoolsup	famsup	paid	activities	nursery	higher	internet	rom
0	at_home	teacher	course	mother	2	2	0	yes	no	no	no	yes	yes	no	
1	at_home	other	course	father	1	2	0	no	yes	no	no	no	yes	yes	
2	at_home	other	other	mother	1	2	0	yes	no	no	no	yes	yes	yes	
3	health	services	home	mother	1	3	0	no	yes	no	yes	yes	yes	yes	
4	other	other	home	father	1	2	0	no	yes	no	no	yes	yes	no	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
644	services	other	course	mother	1	3	1	no	no	no	yes	no	yes	yes	
645	teacher	services	course	mother	1	2	0	no	yes	no	no	yes	yes	yes	
646	other	other	course	mother	2	2	0	no	no	no	yes	yes	yes	no	
647	services	services	course	mother	2	1	0	no	no	no	no	no	yes	yes	
648	services	other	course	mother	3	1	0	no	no	no	no	no	yes	yes	

649 rows × 16 columns

In [192…

```
#4
print(data['traveltime'].value_counts())
plt.figure(figsize = (10,6))
data['traveltime'].hist(bins = 10,color ='grey',edgecolor = 'black')
plt.grid(False)
plt.show()
```

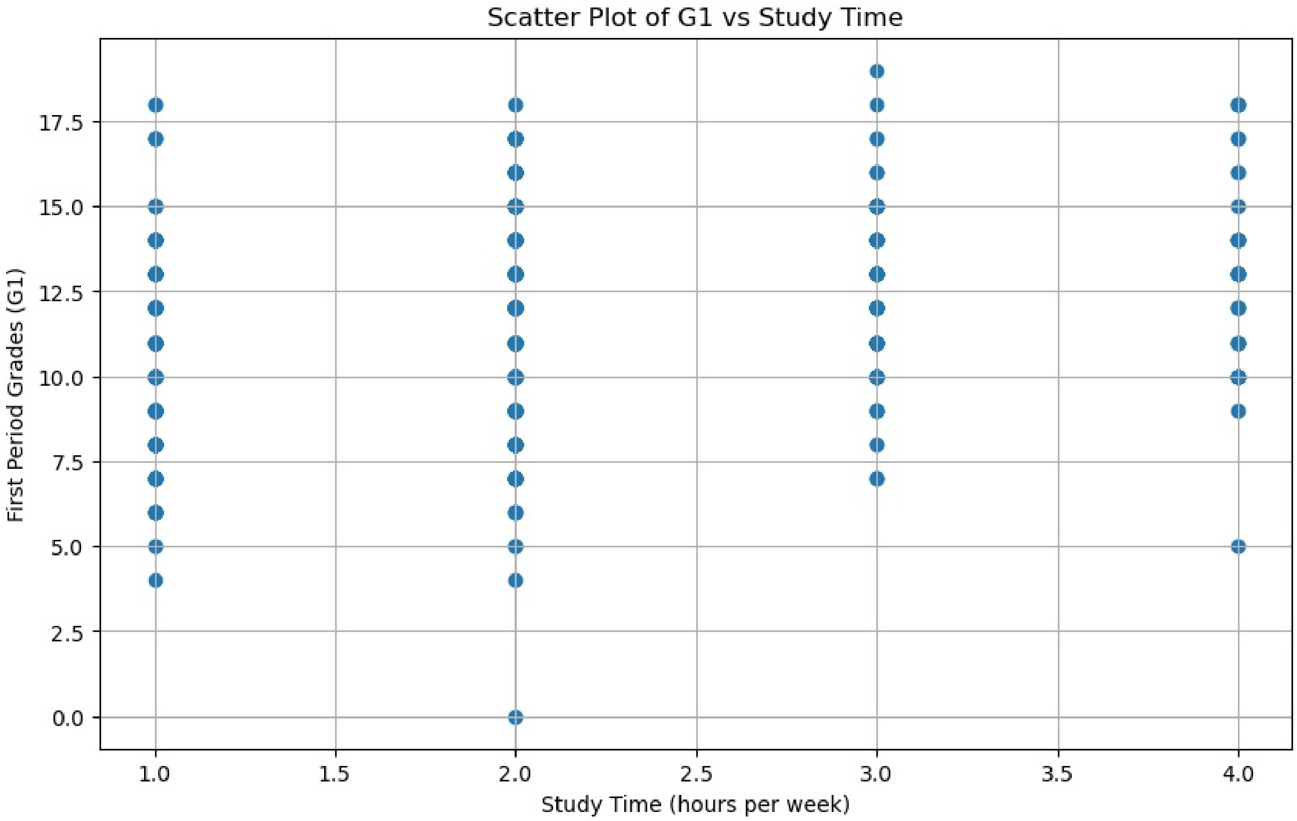
```
1    366
2    213
3     54
4     16
Name: traveltime, dtype: int64
```



In [193…

```
#5
print(data['G1'].corr(data['studytime']))
plt.figure(figsize=(10, 6))
plt.scatter(data['studytime'], data['G1'])
plt.title('Scatter Plot of G1 vs Study Time')
plt.xlabel('Study Time (hours per week)')
plt.ylabel('First Period Grades (G1)')
plt.grid(True)
plt.show()
```

```
0.2608753803131906
```



In [194…

```
#6
data['G3'].corr(data['studytime'])
```

Out[194]:

```
0.24978868999886356
```

In [195…

```
#7
data['absences'].corr(data['G3'])
```

Out[195]:

```
-0.09137905643875621
```

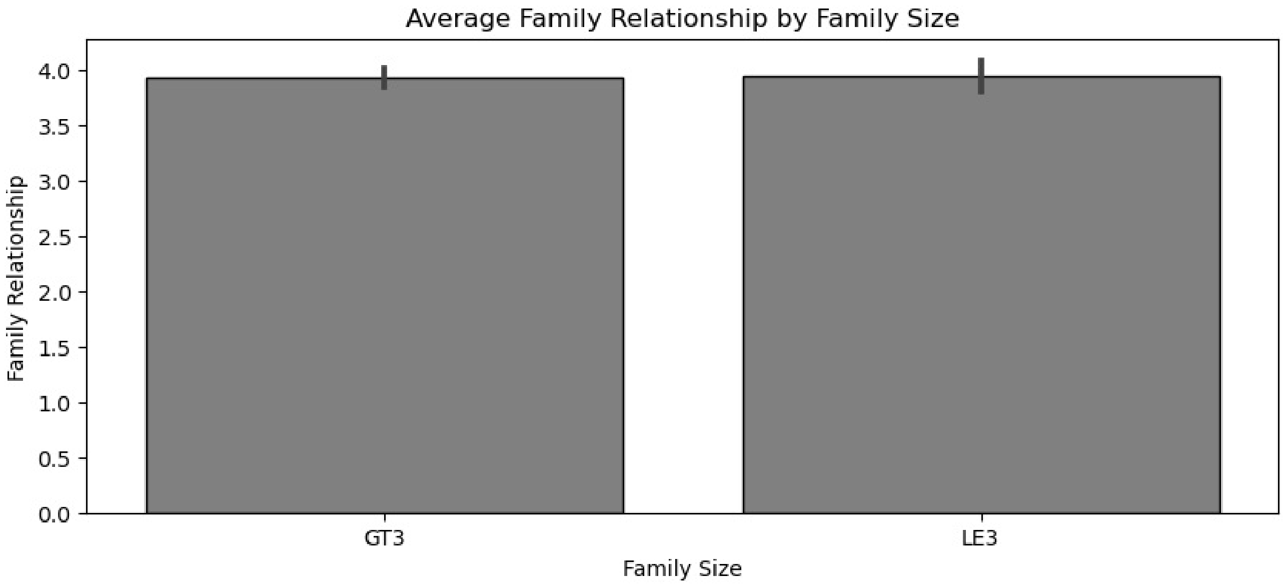
In [196…

```
#8
urban = data[data['address'] == 'U']['G3']
rural = data[data['address'] == 'R']['G3']

t_stat, p_value = ttest_ind(urban, rural)

print(f"P-value: {p_value}")

alpha = 0.05
if p_value < alpha:
    print(" There is a significant difference in G3 between students living in urban (U) and rural (R) areas")
else:
    print("There is no significant difference in final grades (G3) between students living in urban (U) and rur
P-value: 1.764153460922413e-05
 There is a significant difference in G3 between students living in urban (U) and rural (R) areas
```

In [197…

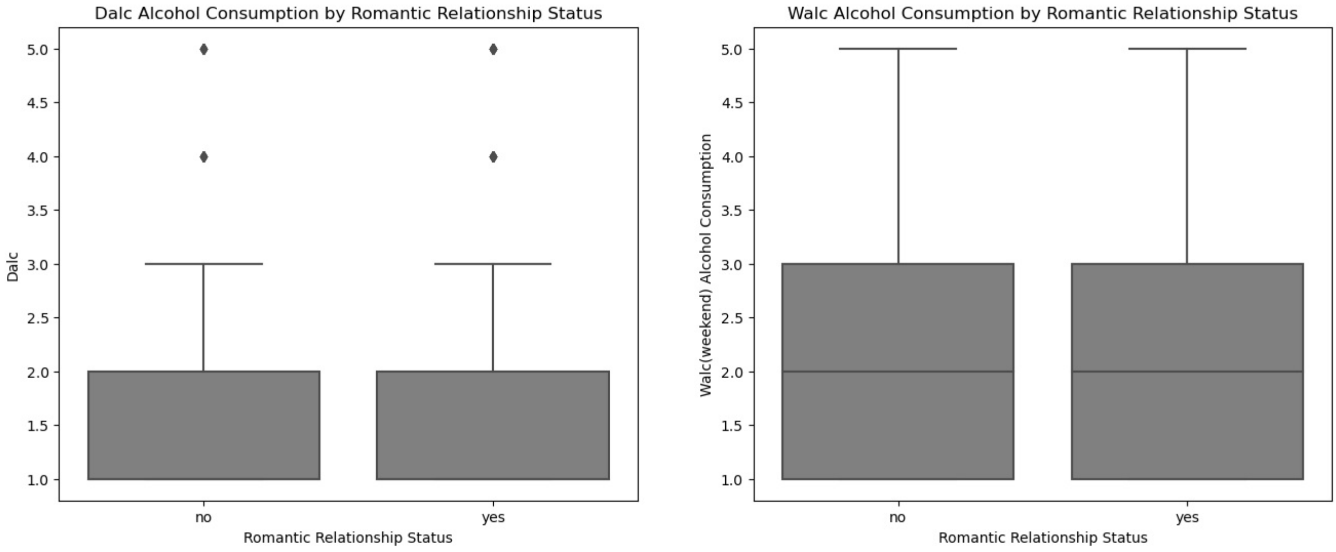
```
#9
print(data['famsize'].unique())
print(data['famrel'].unique())
data['famsize_numeric'] = data['famsize'].map({'GT3': 0, 'LE3': 1})
correlation_coefficient = data['famsize_numeric'].corr(data['famrel'])
print(f"Correlation Coefficient: {correlation_coefficient}")
plt.figure(figsize=(10, 4))
sns.barplot(x='famsize', y='famrel', color = 'grey',edgecolor = 'black',data=data)
plt.title('Average Family Relationship by Family Size')
plt.xlabel('Family Size')
plt.ylabel('Family Relationship')
plt.show()
```

```
['GT3' 'LE3']
[4 5 3 1 2]
Correlation Coefficient: 0.004640788403623516
```



In [164…

```
#10
fig, axes = plt.subplots(nrows=1, ncols=2, figsize=(16, 6))
sns.boxplot(x='romantic', y='Dalc', color='grey', data=data, ax=axes[0])
axes[0].set_title('Dalc Alcohol Consumption by Romantic Relationship Status')
axes[0].set_xlabel('Romantic Relationship Status')
axes[0].set_ylabel('Dalc')

sns.boxplot(x='romantic', y='Walc', color='grey', data=data, ax=axes[1])
axes[1].set_title('Walc Alcohol Consumption by Romantic Relationship Status')
axes[1].set_xlabel('Romantic Relationship Status')
axes[1].set_ylabel('Walc(weekend) Alcohol Consumption')
plt.show()
```



In [165…

```
#11

correlation_coefficient = data['Medu'].corr(data['Fedu'])
print(f"Correlation Coefficient between Medu and Fedu: {correlation_coefficient}")
```

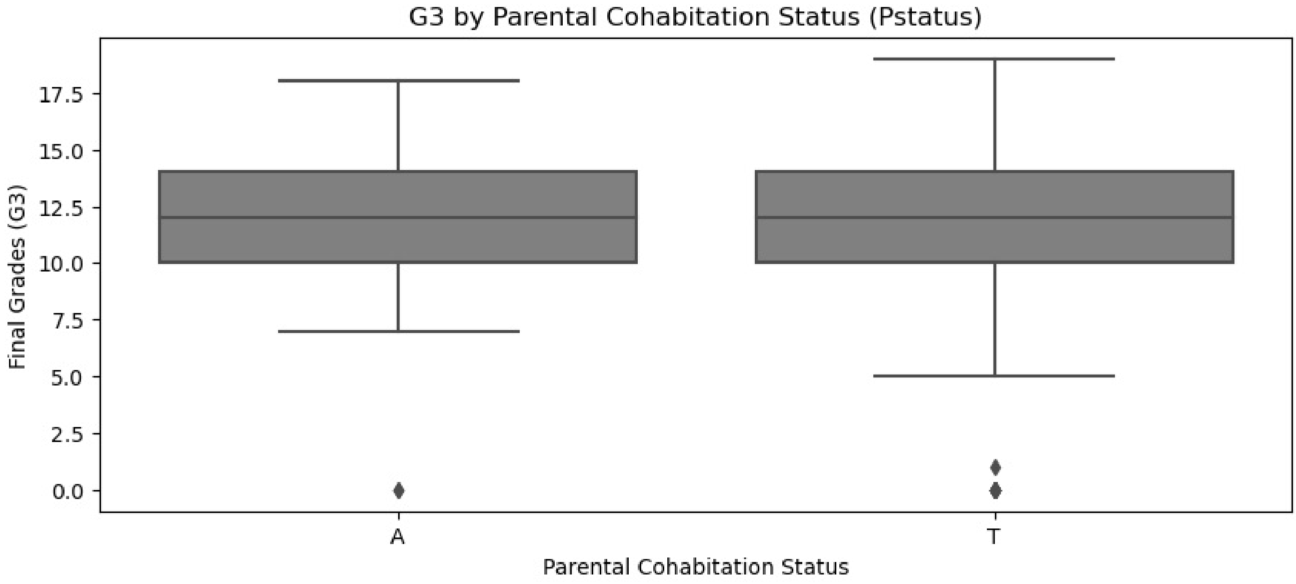
```
Correlation Coefficient between Medu and Fedu: 0.6474766091364946
```

In [166…

```
#12
```

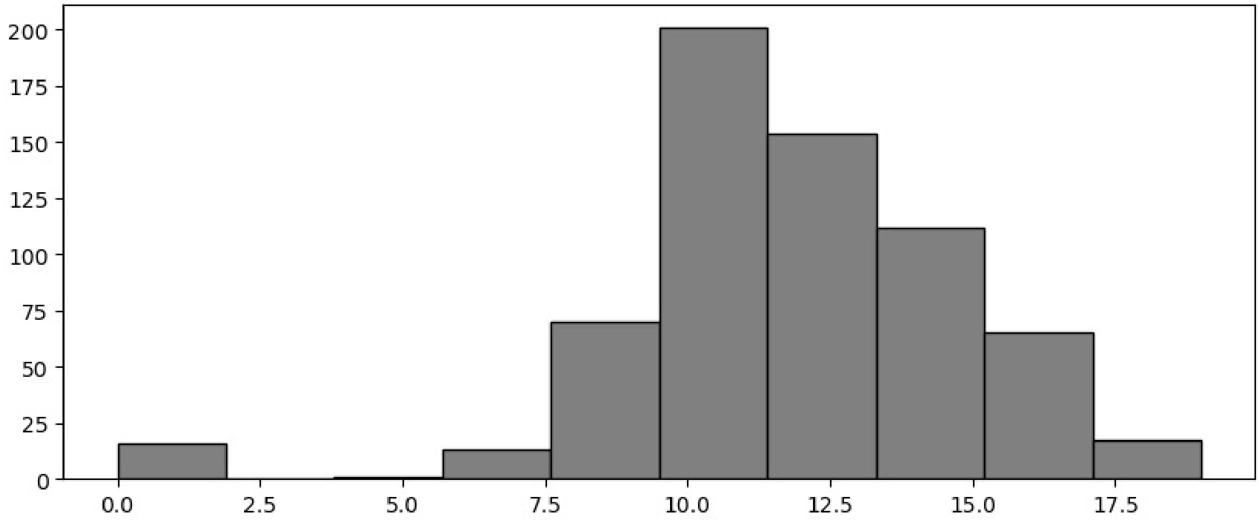
```
print(data['Pstatus'].unique())  #A:living apart;T:living together
plt.figure(figsize = (10,4))
sns.boxplot(x='Pstatus', y='G3', color = 'grey',data=data)
plt.title('G3 by Parental Cohabitation Status (Pstatus)')
plt.xlabel('Parental Cohabitation Status')
plt.ylabel('Final Grades (G3)')
plt.show()
```

```
['A' 'T']
```



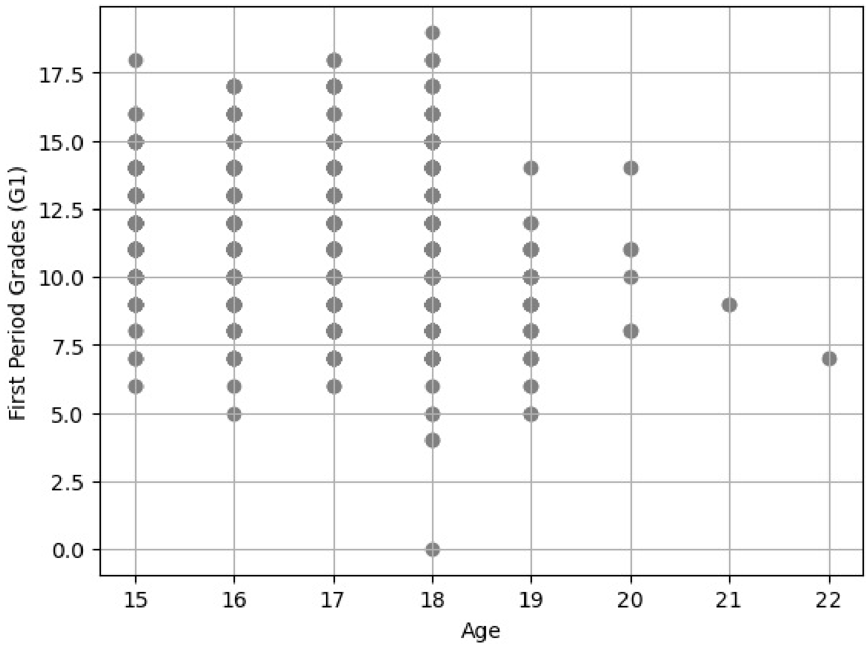
In [167…

```
#13
plt.figure(figsize=(10,4))
data['G3'].hist(color = 'grey', edgecolor='black')
plt.grid(False)
plt.show()
```



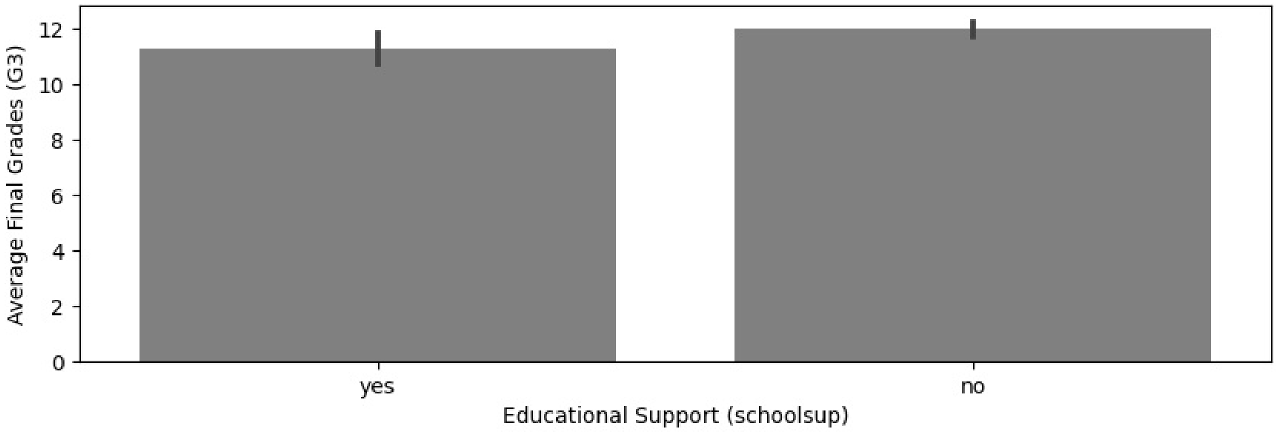
In [168…

```
#14
plt.scatter(data['age'], data['G1'],color = 'grey')
plt.xlabel('Age')
plt.ylabel('First Period Grades (G1)')
plt.grid(True)
plt.show()
```



In [169…

```
#15

plt.figure(figsize=(10, 3))
sns.barplot(x='schoolsup', y='G3',color = 'grey', data=data)
plt.xlabel('Educational Support (schoolsup)')
plt.ylabel('Average Final Grades (G3)')
plt.show()
```



In [170…

```
dataMat = pd.read_csv("student-mat.csv",sep=";")
dataMat
```

Out[170]:

	school	sex	age	address	famsize	Pstatus	Medu	Fedu	Mjob	Fjob	...	famrel	freetime	goout	Dalc	Walc	health	absence
0	GP	F	18	U	GT3	A	4	4	at_home	teacher	...	4	3	4	1	1	3	
1	GP	F	17	U	GT3	T	1	1	at_home	other	...	5	3	3	1	1	3	
2	GP	F	15	U	LE3	T	1	1	at_home	other	...	4	3	2	2	3	3	1
3	GP	F	15	U	GT3	T	4	2	health	services	...	3	2	2	1	1	5	
4	GP	F	16	U	GT3	T	3	3	other	other	...	4	3	2	1	2	5	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	.
390	MS	M	20	U	LE3	A	2	2	services	services	...	5	5	4	4	5	4	1
391	MS	M	17	U	LE3	T	3	1	services	services	...	2	4	5	3	4	2	
392	MS	M	21	R	GT3	T	1	1	other	other	...	5	5	3	3	3	3	
393	MS	M	18	R	LE3	T	3	2	services	other	...	4	4	1	3	4	5	
394	MS	M	19	U	LE3	T	1	1	other	at_home	...	3	2	3	3	3	5	

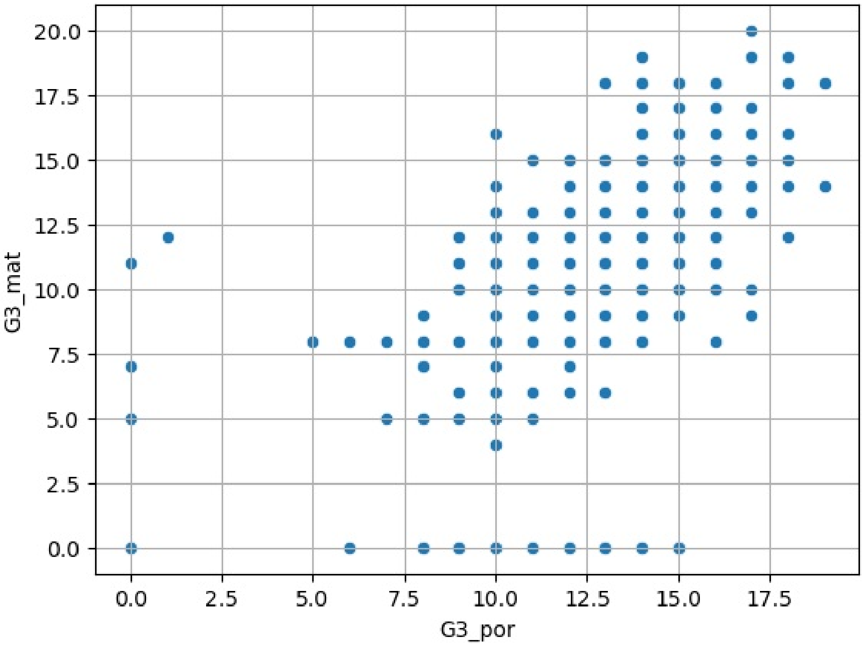
395 rows × 33 columns

In [171…

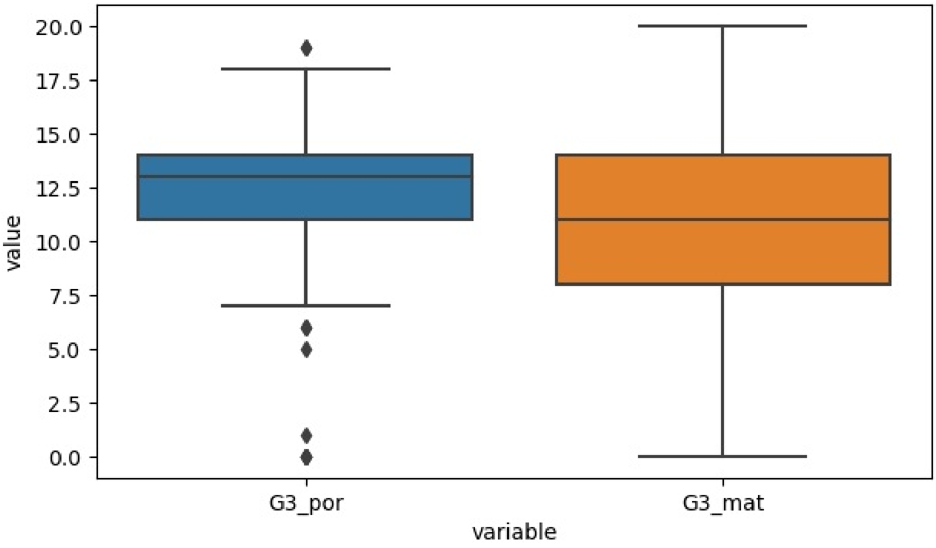
```
data_merged = pd.merge(data,dataMat,on=['school', 'sex', 'age','address','famsize','Pstatus','Medu','Fedu','Mjo
data_merged
```

```
Out[171]:
     school sex age address famsize Pstatus Medu Fedu     Mjob     Fjob ... famsup_mat paid_mat activities_mat nursery_mat
  0      GP   F  18       U     GT3       A    4    4  at_home  teacher ...         no       no             no         yes
  1      GP   F  17       U     GT3       T    1    1  at_home    other ...        yes       no             no          no
  2      GP   F  15       U     LE3       T    1    1  at_home    other ...         no      yes             no         yes
  3      GP   F  15       U     GT3       T    4    2   health services ...        yes      yes            yes         yes
  4      GP   F  16       U     GT3       T    3    3    other    other ...        yes      yes             no         yes
...     ...  ... ...     ...     ...     ...  ...  ...      ...      ... ...        ...      ...            ...         ...
365      MS   F  19       R     GT3       T    2    3 services    other ...         no       no            yes          no
366      MS   F  18       U     LE3       T    3    1  teacher services ...        yes      yes             no         yes
367      MS   F  18       U     GT3       T    1    1    other    other ...         no       no            yes         yes
368      MS   M  17       U     LE3       T    3    1 services services ...         no       no             no          no
369      MS   M  18       R     LE3       T    3    2 services    other ...         no       no             no          no

370 rows × 49 columns
```

```
In [172…
#16

sns.scatterplot(x='G3_por', y='G3_mat', data=data_merged)
plt.grid(True)
plt.show()
```

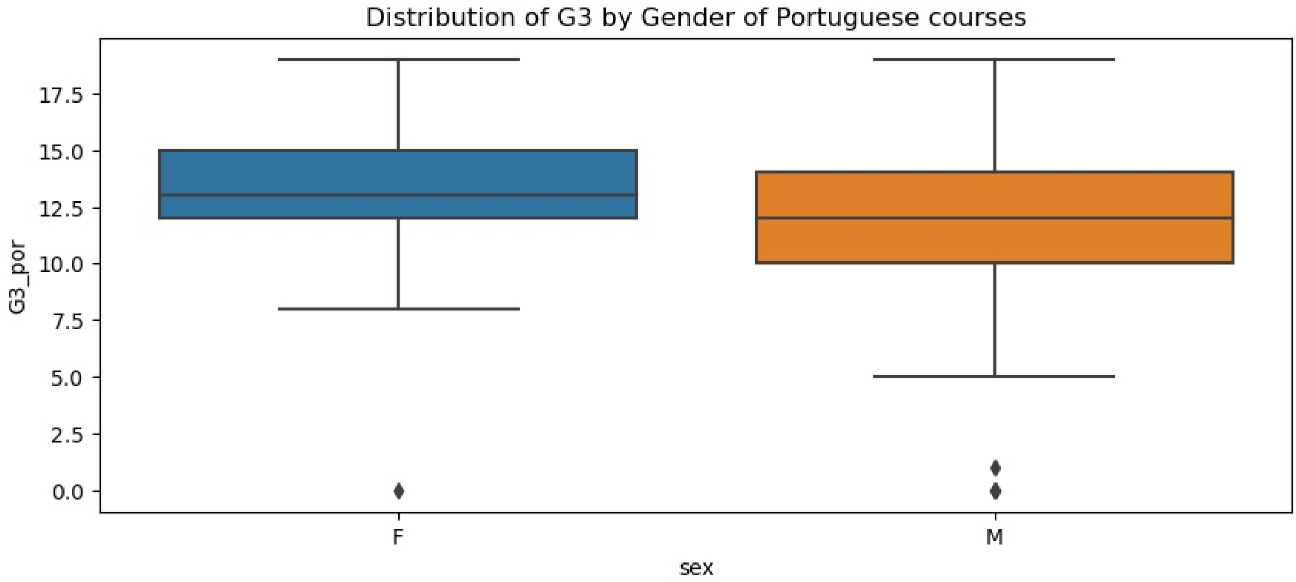


```
In [173…
#17
plt.figure(figsize = (7,4))
sns.boxplot(x='variable', y='value', data=pd.melt(data_merged[['G3_por', 'G3_mat']]))
plt.show()
```



```
In [174…
#18
```

```
plt.figure(figsize=(10, 4))
sns.boxplot(x='sex', y='G3_por', data=data_merged)
plt.title('Distribution of G3 by Gender of Portuguese courses')
plt.show()
male = data_merged[data_merged['sex'] == 'M']['G3_por']
female = data_merged[data_merged['sex'] == 'F']['G3_por']
t_statistic, p_value = ttest_ind(male, female)
print(f'P-value: {p_value}')
alpha = 0.05
if p_value < alpha:print('significant difference in average final grades between male and female')
else: print('no significant difference')
```

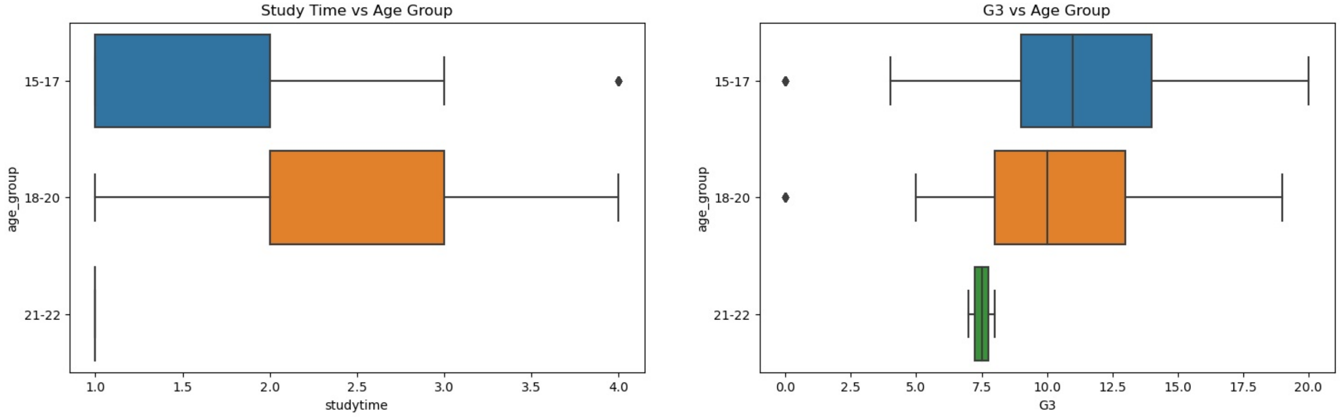


```
P-value: 8.765488960209672e-05
significant difference in average final grades between male and female
```

In [175…

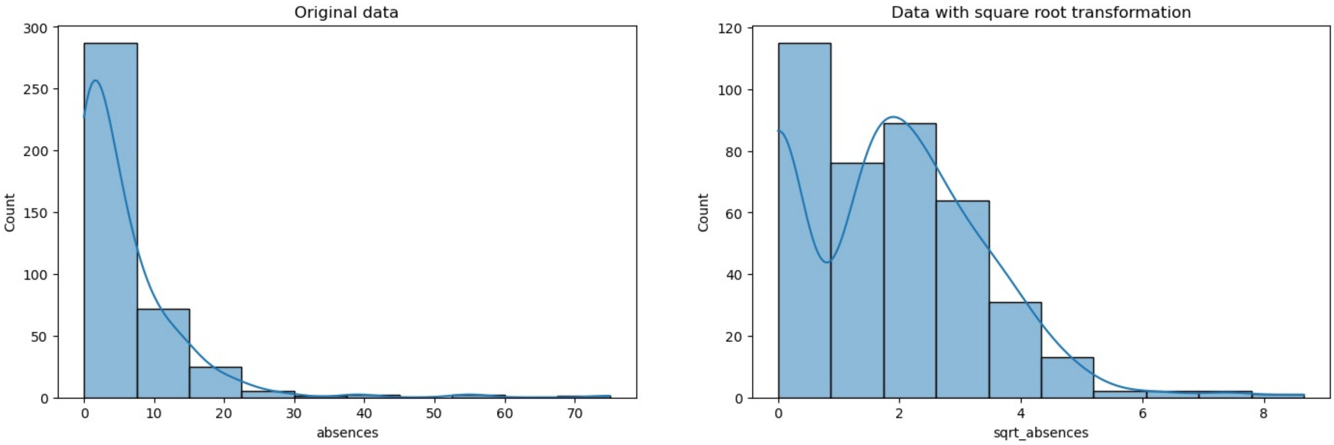
```
#19

ages = [15,18,21,23]
label = ['15-17','18-20','21-22']
dataMat['age_group']=pd.cut(dataMat['age'],bins = ages,labels = label,right = False)
print(dataMat['age_group'].value_counts())
plt.figure(figsize=(18, 5))
plt.subplot(1, 2, 1)
sns.boxplot(x='studytime', y='age_group', data=dataMat)
plt.title('Study Time vs Age Group')
plt.subplot(1, 2, 2)
sns.boxplot(x='G3', y='age_group', data=dataMat)
plt.title('G3 vs Age Group')
plt.show()
```

```
15-17    284
18-20    109
21-22      2
Name: age_group, dtype: int64
```



In [231…

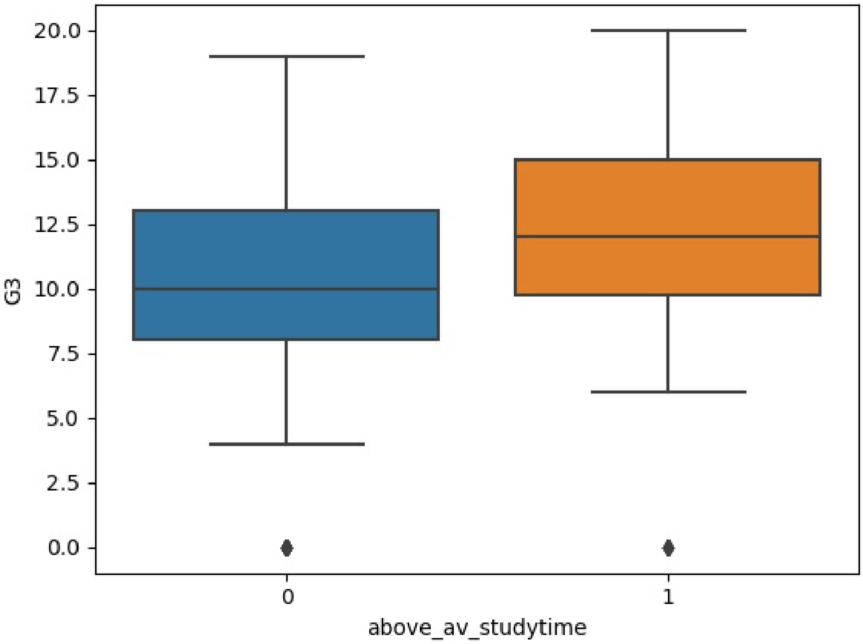
```
#20
dataMat['sqrt_absences'] = np.sqrt(dataMat['absences'])
plt.figure(figsize= (17,5))
plt.subplot(1, 2, 1)
sns.histplot(dataMat['absences'], bins=10, kde=True)
plt.title("Original data")
plt.subplot(1, 2, 2)
sns.histplot(dataMat['sqrt_absences'], bins=10, kde=True)
plt.title("Data with square root transformation")
plt.figure(figsize= (17,8))
plt.show()
```



```
<Figure size 1700x800 with 0 Axes>
```

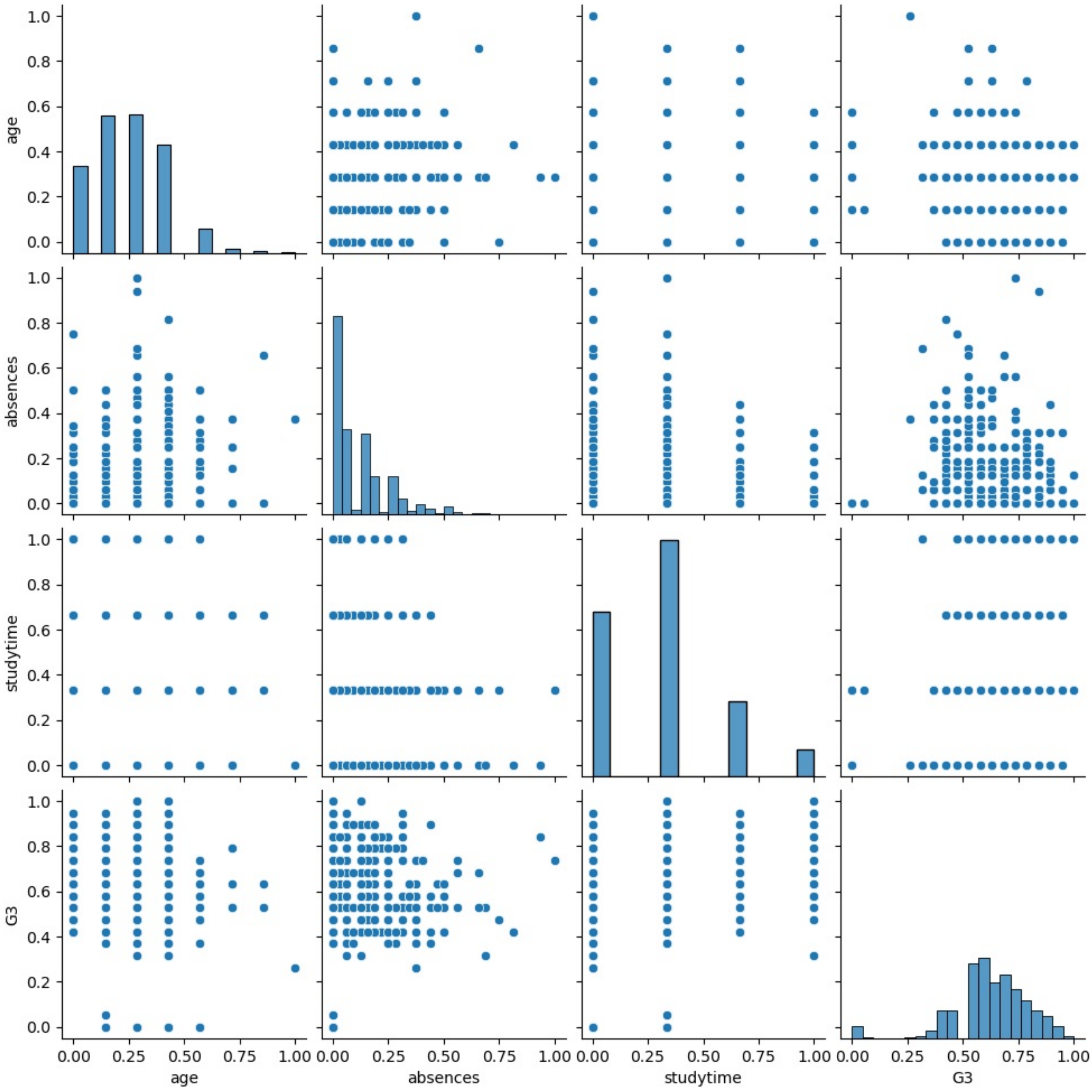
In [177…

```
#21

dataMat['above_av_studytime'] = (dataMat['studytime']>dataMat['studytime'].mean()).astype(int)
sns.boxplot(x = 'above_av_studytime',y = 'G3',data = dataMat)
plt.show()
```



In [178…

```
#22
from sklearn.preprocessing import MinMaxScaler
numeric = ['age', 'absences', 'studytime', 'G3']
numeric_data = data[numeric]
scaler = MinMaxScaler()
scaled_data = scaler.fit_transform(numeric_data)
scaled_df = pd.DataFrame(scaled_data, columns=numeric)
scatter_matrix = sns.pairplot(scaled_df)
plt.show()
```

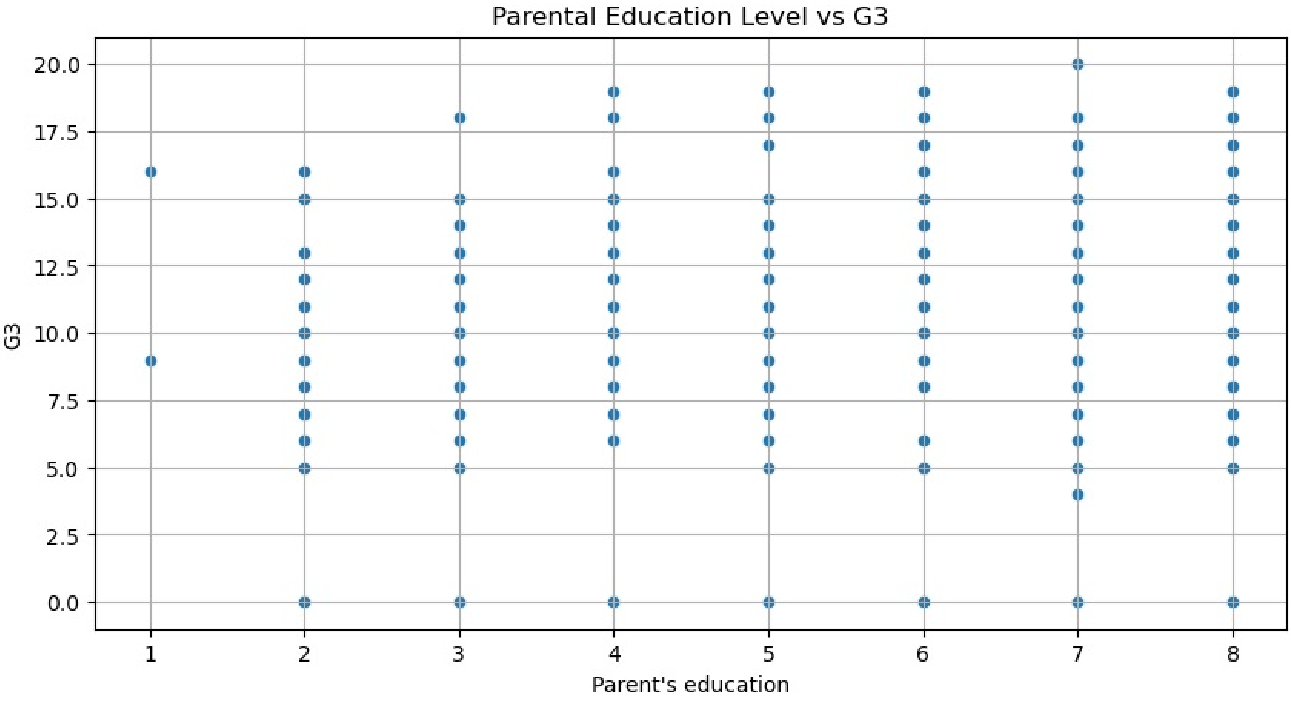


```
In [98]: #23

         data_into_num = pd.get_dummies(dataMat, columns=['school','sex','reason', 'Mjob'], prefix=['school','sex','reas
         data_into_num
```

```
Out[98]:
       age  address  famsize  Pstatus  Medu  Fedu      Fjob  guardian  traveltime  studytime  ...  sex_M  reason_course  reason_home  reasoı
  0     18        U      GT3        A     4     4   teacher    mother           2          2  ...      0              1            0
  1     17        U      GT3        T     1     1     other    father           1          2  ...      0              1            0
  2     15        U      LE3        T     1     1     other    mother           1          2  ...      0              0            0
  3     15        U      GT3        T     4     2  services    mother           1          3  ...      0              0            1
  4     16        U      GT3        T     3     3     other    father           1          2  ...      0              0            1
 ...   ...      ...      ...      ...   ...   ...       ...       ...         ...        ...  ...    ...            ...          ...
 390    20        U      LE3        A     2     2  services     other           1          2  ...      1              1            0
 391    17        U      LE3        T     3     1  services    mother           2          1  ...      1              1            0
 392    21        R      GT3        T     1     1     other     other           1          1  ...      1              1            0
 393    18        R      LE3        T     3     2     other    mother           3          1  ...      1              1            0
 394    19        U      LE3        T     1     1   at_home    father           1          1  ...      1              1            0

395 rows × 45 columns
```

```
In [199…: #24

         dataMat['parent_edu'] = dataMat['Medu'] + dataMat['Fedu']
         plt.figure(figsize=(10, 5))
```

```
sns.scatterplot(x='parent_edu', y='G3', data=dataMat)
plt.title('Parental Education Level vs G3')
plt.xlabel("Parent's education")
plt.ylabel('G3')
plt.grid()
plt.show()
correlation = dataMat['parent_edu'].corr(dataMat['G3'])
print(correlation)
```



```
0.20522443411453914
```

In [200…

```
#25

print(data[data['address'] == 'U']['studytime'].mean())
print(data[data['address'] == 'R']['studytime'].mean())
```

```
1.9646017699115044
1.8527918781725887
```

In [201…

```
#26

print(data['famrel'].unique())
data['famrel'] = data['famrel'].replace(1, 'very bad')
data['famrel'] = data['famrel'].replace(2, 'bad')
data['famrel'] = data['famrel'].replace(3, 'neutral')
data['famrel'] = data['famrel'].replace(4, 'good')
data['famrel'] = data['famrel'].replace(5, 'excellent')
data.iloc[:,20:24]
```

```
[4 5 3 1 2]
```

Out[201]:

	higher	internet	romantic	famrel
0	yes	no	no	good
1	yes	yes	no	excellent
2	yes	yes	no	good
3	yes	yes	yes	neutral
4	yes	no	no	good
...	...	...	...	...
644	yes	yes	no	excellent
645	yes	yes	no	good
646	yes	no	no	very bad
647	yes	yes	no	bad
648	yes	yes	no	good

649 rows × 4 columns

In [202…

```
#27
def age_range(series):
    return series.max() - series.min()
result1 = data.groupby('school')['age'].agg(age_range)
print(result1)
print()
```

```
def internet(series):
    return (series == 'yes').mean() * 100
result2 = data.groupby('sex')['internet'].agg(internet)
print(result2)
```

```
school
GP    7
MS    5
Name: age, dtype: int64

sex
F    74.412533
M    80.075188
Name: internet, dtype: float64
```

In [103…

```
# 28 done
```

In [203…

```
#29
group1 = data[data['schoolsup'] == 'yes']
print(group1['absences'].median())
group2 = data[data['schoolsup'] == 'no']
print(group2['absences'].median())
```

```
2.0
2.0
```

In [204…

```
#30

def higher(series):
    return (series == 'yes').mean() * 100
data.groupby('Fedu')['higher'].agg(higher)
```

Out[204]:

```
Fedu
0    100.000000
1     81.034483
2     87.559809
3     93.893130
4     98.437500
Name: higher, dtype: float64
```

In [206…

```
#31

data['G3'].corr(data['traveltime'])
```

Out[206]:

```
-0.12717296675842077
```

In [207…

```
#32
av_weight = np.average(data['G3'],weights = data['studytime'])
print(f"Weigted average of G3 by studytime as weights:{av_weight}")
```

```
Weigted average of G3 by studytime as weights:12.25219473264166
```

In [208…

```
#33
student = data.loc[data['Walc'].idxmax()]
print("Student information with highest Walc")
print(student)
```

```
Student information with highest Walc
school                    GP
sex                        M
age                       16
address                    U
famsize                  GT3
Pstatus                    T
Medu                       4
Fedu                       4
Mjob                 teacher
Fjob                 teacher
reason                  home
guardian              mother
traveltime                 1
studytime                  2
failures                   0
schoolsup                 no
famsup                   yes
paid                     yes
activities               yes
nursery                  yes
higher                   yes
internet                 yes
romantic                 yes
famrel                  good
freetime                   4
goout                      5
Dalc                       5
Walc                       5
health                     5
absences                   4
G1                        12
G2                        11
G3                        12
famsize_numeric            0
Name: 29, dtype: object
```

In [109…

```
#34
print(data['guardian'].isnull().sum())
data['guardian'].fillna('unknown', inplace=True)
print(data['guardian'].value_counts())
```

```
0
mother    455
father    153
other      41
Name: guardian, dtype: int64
```

In [110…

```
#35
data['romantic'].isnull().sum()
most_common = data['romantic'].mode()
print(most_common)
data['romantic'].fillna(most_common, inplace=True)
print(data['romantic'].value_counts())
```

```
0    no
Name: romantic, dtype: object
no     410
yes    239
Name: romantic, dtype: int64
```

In [209…

```
#36
data.pivot_table(values='studytime',index='reason',aggfunc={'studytime': ['max', 'min']})
```

Out[209]:

	max	min
reason		
course	4	1
home	4	1
other	4	1
reputation	4	1

In [112…

```
#37

has = data[(data['Mjob'] == 'teacher') & (data['Fjob'] == 'teacher')]
print("Students with parents with both works as teacher")
has
```

```
Students with parents with both works as teacher
```

Out[112]:

	school	sex	age	address	famsize	Pstatus	Medu	Fedu	Mjob	Fjob	...	freetime	goout	Dalc	Walc	health	absences	G1
29	GP	M	0.142857	U	GT3	T	4	4	teacher	teacher	...	4	5	5	5	5	0.1250	12
110	GP	M	0.000000	U	LE3	A	4	4	teacher	teacher	...	5	3	1	1	4	0.1250	13
115	GP	M	0.142857	U	GT3	T	4	4	teacher	teacher	...	4	4	1	2	5	0.1875	16
128	GP	M	0.142857	R	GT3	T	4	4	teacher	teacher	...	5	5	2	5	4	0.2500	14
147	GP	F	0.000000	U	GT3	T	4	4	teacher	teacher	...	3	2	1	1	5	0.1875	13
161	GP	M	0.142857	U	GT3	T	4	4	teacher	teacher	...	3	2	2	1	5	0.5000	9
213	GP	F	0.142857	U	LE3	T	4	4	teacher	teacher	...	5	2	1	2	3	0.0000	11
246	GP	M	0.285714	U	GT3	T	4	4	teacher	teacher	...	5	5	1	3	2	0.0000	13
257	GP	M	0.285714	U	GT3	T	4	4	teacher	teacher	...	2	1	1	2	5	0.1875	10
335	GP	M	0.428571	U	LE3	A	4	4	teacher	teacher	...	4	3	1	1	2	0.0000	17
344	GP	M	0.428571	U	LE3	T	4	4	teacher	teacher	...	4	2	2	2	1	0.0000	18
356	GP	F	0.285714	R	GT3	T	4	4	teacher	teacher	...	4	4	1	1	5	0.0625	15
381	GP	F	0.285714	U	GT3	T	4	4	teacher	teacher	...	3	3	1	2	4	0.1250	15
448	MS	F	0.142857	R	GT3	T	4	4	teacher	teacher	...	2	2	1	1	4	0.1875	16
594	MS	F	0.428571	U	GT3	T	4	4	teacher	teacher	...	3	5	1	2	1	0.0000	18
636	MS	M	0.428571	U	GT3	T	4	4	teacher	teacher	...	2	4	1	4	2	0.1250	17

16 rows × 34 columns

In [113…

```
#38
data['Mjob'].replace('at_home','homemaker',inplace = True)
data['Fjob'].replace('at_home','homemaker',inplace = True)
data
```

Out[113]:

	school	sex	age	address	famsize	Pstatus	Medu	Fedu	Mjob	Fjob	...	freetime	goout	Dalc	Walc	health	absences
0	GP	F	0.428571	U	GT3	A	4	4	homemaker	teacher	...	3	4	1	1	3	0.1250
1	GP	F	0.285714	U	GT3	T	1	1	homemaker	other	...	3	3	1	1	3	0.0625
2	GP	F	0.000000	U	LE3	T	1	1	homemaker	other	...	3	2	2	3	3	0.1875
3	GP	F	0.000000	U	GT3	T	4	2	health	services	...	2	2	1	1	5	0.0000
4	GP	F	0.142857	U	GT3	T	3	3	other	other	...	3	2	1	2	5	0.0000
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
644	MS	F	0.571429	R	GT3	T	2	3	services	other	...	4	2	1	2	5	0.1250
645	MS	F	0.428571	U	LE3	T	3	1	teacher	services	...	3	4	1	1	1	0.1250
646	MS	F	0.428571	U	GT3	T	1	1	other	other	...	1	1	1	1	5	0.1875
647	MS	M	0.285714	U	LE3	T	3	1	services	services	...	4	5	3	4	2	0.1875
648	MS	M	0.428571	R	LE3	T	3	2	services	other	...	4	1	3	4	5	0.1250

649 rows × 34 columns

In [210…

```
#39
melted_data = pd.melt(data, id_vars=data.columns.difference(['Mjob', 'Fjob']).tolist(), value_vars=['Mjob', 'Fj
                      value_name='ParentJob')
melted_data
```

Out[210]:

	Dalc	Fedu	G1	G2	G3	Medu	Pstatus	Walc	absences	activities	...	paid	reason	romantic	school	schoolsup	sex	studytime
0	1	4	0	11	11	4	A	1	4	no	...	no	course	no	GP	yes	F	2
1	1	1	9	11	11	1	T	1	2	no	...	no	course	no	GP	no	F	2
2	2	1	12	13	12	1	T	3	6	no	...	no	other	no	GP	yes	F	2
3	1	2	14	14	14	4	T	1	0	yes	...	no	home	yes	GP	no	F	3
4	1	3	11	13	13	3	T	2	0	no	...	no	home	no	GP	no	F	2
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1293	1	3	10	11	10	2	T	2	4	yes	...	no	course	no	MS	no	F	3
1294	1	1	15	15	16	3	T	1	4	no	...	no	course	no	MS	no	F	2
1295	1	1	11	12	9	1	T	1	6	yes	...	no	course	no	MS	no	F	2
1296	3	1	10	10	10	3	T	4	6	no	...	no	course	no	MS	no	M	1
1297	3	2	10	11	11	3	T	4	4	no	...	no	course	no	MS	no	M	1

1298 rows × 34 columns

In [211…

```python
#40
def letter_grade(score):
    if score >= 16:
        return 'A'
    elif score >= 14:
        return 'B'
    elif score >= 12:
        return 'C'
    elif score >= 10:
        return 'D'
    else:
        return 'F'
data['letter_grade'] = data['G3'].apply(letter_grade)
data
```

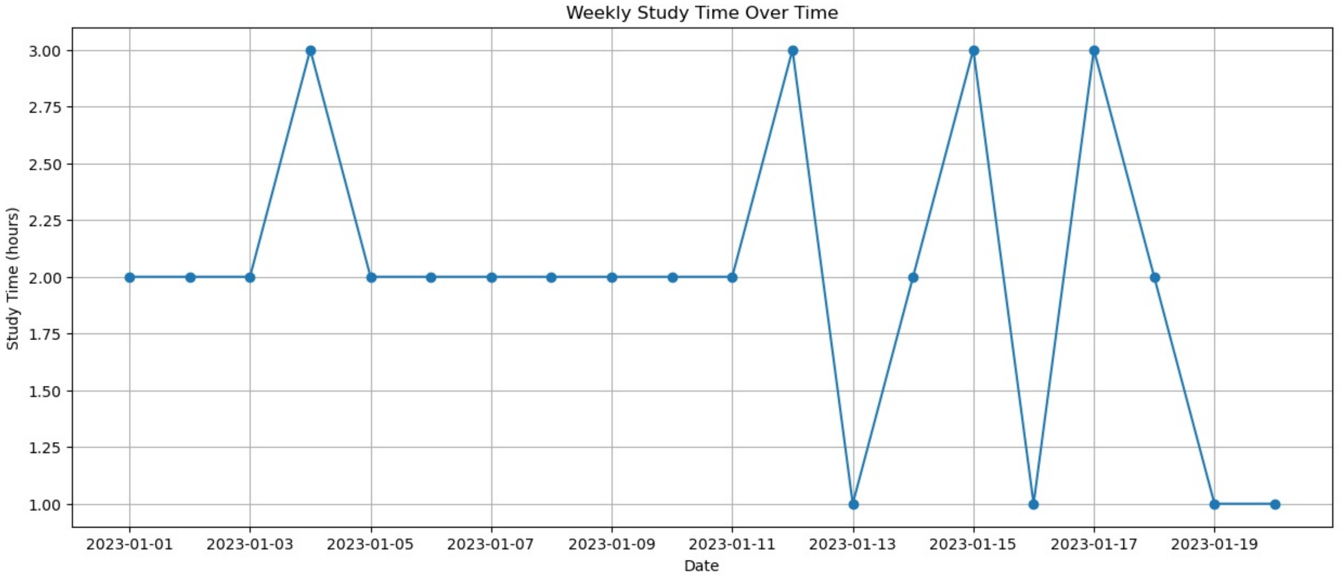
Out[211]:

	school	sex	age	address	famsize	Pstatus	Medu	Fedu	Mjob	Fjob	...	goout	Dalc	Walc	health	absences	G1	G2	G3	f
0	GP	F	18	U	GT3	A	4	4	at_home	teacher	...	4	1	1	3	4	0	11	11	
1	GP	F	17	U	GT3	T	1	1	at_home	other	...	3	1	1	3	2	9	11	11	
2	GP	F	15	U	LE3	T	1	1	at_home	other	...	2	2	3	3	6	12	13	12	
3	GP	F	15	U	GT3	T	4	2	health	services	...	2	1	1	5	0	14	14	14	
4	GP	F	16	U	GT3	T	3	3	other	other	...	2	1	2	5	0	11	13	13	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
644	MS	F	19	R	GT3	T	2	3	services	other	...	2	1	2	5	4	10	11	10	
645	MS	F	18	U	LE3	T	3	1	teacher	services	...	4	1	1	1	4	15	15	16	
646	MS	F	18	U	GT3	T	1	1	other	other	...	1	1	1	5	6	11	12	9	
647	MS	M	17	U	LE3	T	3	1	services	services	...	5	3	4	2	6	10	10	10	
648	MS	M	18	R	LE3	T	3	2	services	other	...	1	3	4	5	4	10	11	11	

649 rows × 35 columns

In [217…

```python
#41

needed_data = data.head(20).copy()
needed_data['Date'] = pd.date_range(start='2023-01-01', periods=len(needed_data))
needed_data.set_index('Date', inplace=True)
plt.figure(figsize=(15, 6))
plt.plot(needed_data.index, needed_data['studytime'], marker='o', linestyle='-')
plt.title('Weekly Study Time Over Time')
plt.xlabel('Date')
plt.ylabel('Study Time (hours)')
plt.grid(True)
plt.show()
```



In [218…

```python
#42
data_merged
```

Out[218]:

	school	sex	age	address	famsize	Pstatus	Medu	Fedu	Mjob	Fjob	...	famsup_mat	paid_mat	activities_mat	nursery_mat
0	GP	F	18	U	GT3	A	4	4	at_home	teacher	...	no	no	no	yes
1	GP	F	17	U	GT3	T	1	1	at_home	other	...	yes	no	no	no
2	GP	F	15	U	LE3	T	1	1	at_home	other	...	no	yes	no	yes
3	GP	F	15	U	GT3	T	4	2	health	services	...	yes	yes	yes	yes
4	GP	F	16	U	GT3	T	3	3	other	other	...	yes	yes	no	yes
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
365	MS	F	19	R	GT3	T	2	3	services	other	...	no	no	yes	no
366	MS	F	18	U	LE3	T	3	1	teacher	services	...	yes	yes	no	yes
367	MS	F	18	U	GT3	T	1	1	other	other	...	no	no	yes	yes
368	MS	M	17	U	LE3	T	3	1	services	services	...	no	no	no	no
369	MS	M	18	R	LE3	T	3	2	services	other	...	no	no	no	no

370 rows × 49 columns

In [219…

```
#43
sorted = GP.sort_values(by='G3', ascending=False)
sorted.head()
```

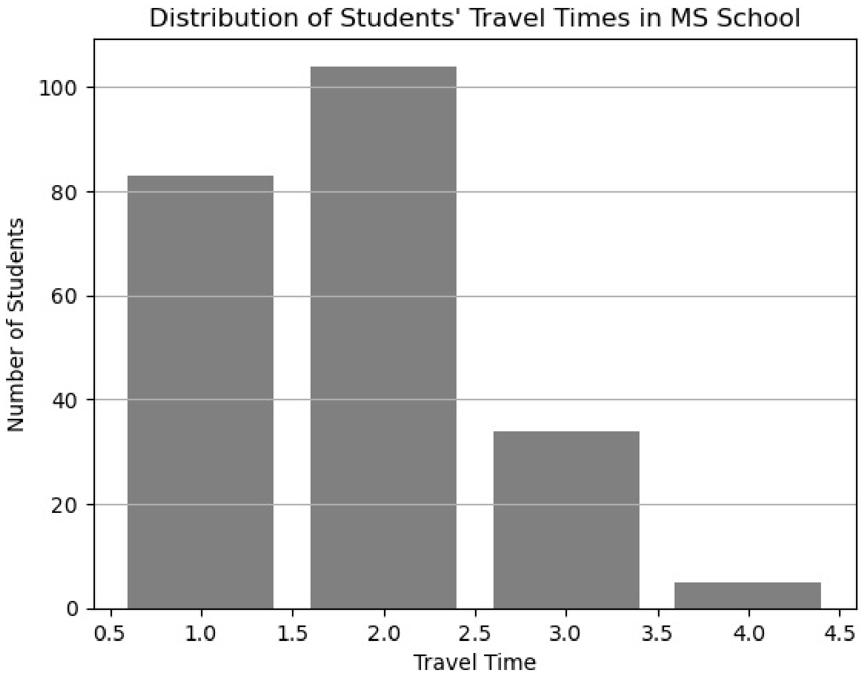
Out[219]:

	school	sex	age	address	famsize	Pstatus	Medu	Fedu	Mjob	Fjob	...	famrel	freetime	goout	Dalc	Walc	health	absence
338	GP	F	17	R	LE3	T	3	1	services	other	...	3	1	2	1	1	3	
416	GP	M	17	U	LE3	A	3	2	other	other	...	4	4	4	1	2	5	1
185	GP	M	16	U	GT3	T	1	0	other	other	...	4	3	2	1	1	3	
332	GP	F	18	U	GT3	T	2	2	at_home	at_home	...	4	3	3	1	2	2	
314	GP	M	17	R	GT3	T	1	2	at_home	at_home	...	3	5	2	2	2	1	

5 rows × 33 columns

In [221…

```
#44
plt.bar(MS['traveltime'].value_counts().index, MS['traveltime'].value_counts(), color='grey')
plt.title('Distribution of Students\' Travel Times in MS School')
plt.xlabel('Travel Time')
plt.ylabel('Number of Students')
plt.grid(axis='y')
plt.show()
```



In [222…

```
#45
with_activities = data[data['activities'] == 'yes']
print(f"Mean age of students with activities:{with_activities['age'].mean()}")
no_activities = data[data['activities'] =='no']
print(f"Mean age of students without activities:{no_activities['age'].mean()}")
```

```
Mean age of students with activities:16.676190476190477
Mean age of students without activities:16.808383233532933
```

In [223…

```
#46
median_absences_by_group = data.groupby(['sex', 'address'])['absences'].median().reset_index()

print(median_absences_by_group)
```

```
  sex address  absences
0   F       R       2.0
1   F       U       2.0
2   M       R       2.0
3   M       U       2.0
```

In [224…

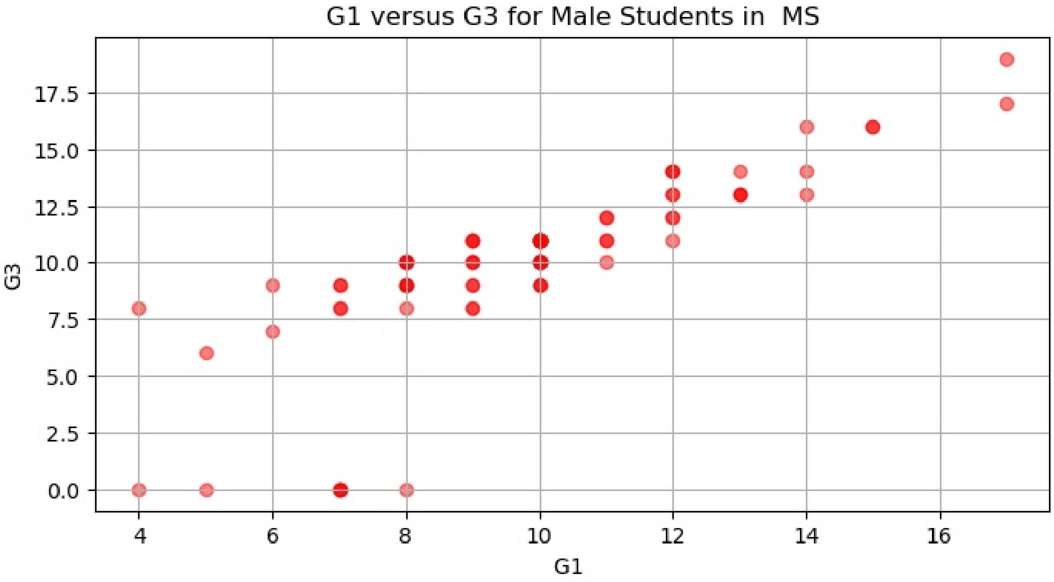
```
#47
new = GP[GP['schoolsup'] == 'yes']
res= (len(new)/len(GP))*100
res
```

Out[224]:

```
13.238770685579196
```

In [225…

```
#48

male_MS = MS[MS['sex'] == 'M']
plt.figure(figsize=(8, 4))
plt.scatter(male_MS['G1'], male_MS['G3'], color='red', alpha=0.5)
plt.title('G1 versus G3 for Male Students in  MS')
plt.xlabel('G1 ')
plt.ylabel('G3')
plt.grid(True)
plt.show()
```



In [228…

```
#49
unique = data.groupby(['Mjob', 'Fjob']).filter(lambda x: len(x) == 1)
print(unique[['Mjob', 'Fjob']])
```

```
       Mjob     Fjob
588  health  at_home
```

In [230…

```
#50
print(f"To the GP students: {GP.groupby('studytime')['G3'].mean()}")
print(f"To the MS students : {MS.groupby('studytime')['G3'].mean()}")
```

```
To the GP students: studytime
1    11.529412
2    12.733010
3    13.563380
4    13.407407
Name: G3, dtype: float64
To the MS students : studytime
1     9.967742
2    10.757576
3    12.307692
4    11.875000
Name: G3, dtype: float64
```

In [ ]:

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