Lab Assignment 3: Exploratory Data Analysis (EDA)

Aim: To perform statistical analysis, correlation studies, and data visualization using the Student Performance dataset, helping to understand patterns and relationships within the data.

Task 1: Load and Explore the Dataset

1. Load the Student Performance dataset using pandas.

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
import warnings
warnings.filterwarnings('ignore')

df = pd.read_csv('student-por.csv')

df.head()
```

```
  school sex  age address famsize Pstatus  Medu  Fedu     Mjob
Fjob  ...  \
0     GP   F   18       U     GT3       A     4     4  at_home
teacher  ...
1     GP   F   17       U     GT3       T     1     1  at_home
other  ...
2     GP   F   15       U     LE3       T     1     1  at_home
other  ...
3     GP   F   15       U     GT3       T     4     2   health
services  ...
4     GP   F   16       U     GT3       T     3     3    other
other  ...

  famrel freetime  goout  Dalc  Walc health absences  G1  G2  G3
0      4        3      4     1     1      3        4   0  11  11
1      5        3      3     1     1      3        2   9  11  11
2      4        3      2     2     3      3        6  12  13  12
3      3        2      2     1     1      5        0  14  14  14
4      4        3      2     1     2      5        0  11  13  13

[5 rows x 33 columns]
```

2. Display dataset characteristics:

– Number of records and features

```
df.shape
```

```
(649, 33)
```

– Data types of columns

```
df.dtypes
```

```
school        object
sex           object
age            int64
address       object
famsize       object
Pstatus       object
Medu           int64
Fedu           int64
Mjob          object
Fjob          object
reason        object
guardian      object
traveltime     int64
studytime      int64
failures       int64
schoolsup     object
famsup        object
paid          object
activities    object
nursery       object
higher        object
internet      object
romantic      object
famrel         int64
freetime       int64
goout          int64
Dalc           int64
Walc           int64
health         int64
absences       int64
G1             int64
G2             int64
G3             int64
dtype: object
```

– Summary statistics (mean, median, mode, standard deviation, etc.).

```
df.describe()
```

```
              age        Medu        Fedu  traveltime   studytime
failures  \
count  649.000000  649.000000  649.000000  649.000000  649.000000
649.000000
mean    16.744222    2.514638    2.306626    1.568567    1.930663
0.221880
std      1.218138    1.134552    1.099931    0.748660    0.829510
0.593235
```

```
min     15.000000    0.000000    0.000000    1.000000    1.000000
0.000000
25%     16.000000    2.000000    1.000000    1.000000    1.000000
0.000000
50%     17.000000    2.000000    2.000000    1.000000    2.000000
0.000000
75%     18.000000    4.000000    3.000000    2.000000    2.000000
0.000000
max     22.000000    4.000000    4.000000    4.000000    4.000000
3.000000

           famrel    freetime       goout        Dalc        Walc
health  \
count  649.000000  649.000000  649.000000  649.000000  649.000000
649.000000
mean     3.930663    3.180277    3.184900    1.502311    2.280431
3.536210
std      0.955717    1.051093    1.175766    0.924834    1.284380
1.446259
min      1.000000    1.000000    1.000000    1.000000    1.000000
1.000000
25%      4.000000    3.000000    2.000000    1.000000    1.000000
2.000000
50%      4.000000    3.000000    3.000000    1.000000    2.000000
4.000000
75%      5.000000    4.000000    4.000000    2.000000    3.000000
5.000000
max      5.000000    5.000000    5.000000    5.000000    5.000000
5.000000

         absences          G1          G2          G3
count  649.000000  649.000000  649.000000  649.000000
mean     3.659476   11.399076   11.570108   11.906009
std      4.640759    2.745265    2.913639    3.230656
min      0.000000    0.000000    0.000000    0.000000
25%      0.000000   10.000000   10.000000   10.000000
50%      2.000000   11.000000   11.000000   12.000000
75%      6.000000   13.000000   13.000000   14.000000
max     32.000000   19.000000   19.000000   19.000000
```

3. Identify missing values, duplicates, and outliers in the dataset.

```
df.isnull()

     school    sex    age  address  famsize  Pstatus   Medu   Fedu
Mjob  \
0     False  False  False    False    False    False  False  False
False
1     False  False  False    False    False    False  False  False
False
```

```
2     False  False  False    False    False    False  False  False
False
3     False  False  False    False    False    False  False  False
False
4     False  False  False    False    False    False  False  False
False
..      ...    ...    ...      ...      ...      ...    ...    ...
...
644   False  False  False    False    False    False  False  False
False
645   False  False  False    False    False    False  False  False
False
646   False  False  False    False    False    False  False  False
False
647   False  False  False    False    False    False  False  False
False
648   False  False  False    False    False    False  False  False
False

      Fjob  ...  famrel  freetime  goout   Dalc   Walc  health
absences  \
0    False  ...   False     False  False  False  False   False
False
1    False  ...   False     False  False  False  False   False
False
2    False  ...   False     False  False  False  False   False
False
3    False  ...   False     False  False  False  False   False
False
4    False  ...   False     False  False  False  False   False
False
..     ...  ...     ...       ...    ...    ...    ...     ...       .
..
644  False  ...   False     False  False  False  False   False
False
645  False  ...   False     False  False  False  False   False
False
646  False  ...   False     False  False  False  False   False
False
647  False  ...   False     False  False  False  False   False
False
648  False  ...   False     False  False  False  False   False
False

        G1     G2     G3
0    False  False  False
1    False  False  False
2    False  False  False
3    False  False  False
```

```
4    False  False  False
..     ...    ...    ...
644  False  False  False
645  False  False  False
646  False  False  False
647  False  False  False
648  False  False  False

[649 rows x 33 columns]
```

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

```
school        0
sex           0
age           0
address       0
famsize       0
Pstatus       0
Medu          0
Fedu          0
Mjob          0
Fjob          0
reason        0
guardian      0
traveltime    0
studytime     0
failures      0
schoolsup     0
famsup        0
paid          0
activities    0
nursery       0
higher        0
internet      0
romantic      0
famrel        0
freetime      0
goout         0
Dalc          0
Walc          0
health        0
absences      0
G1            0
G2            0
G3            0
dtype: int64
```

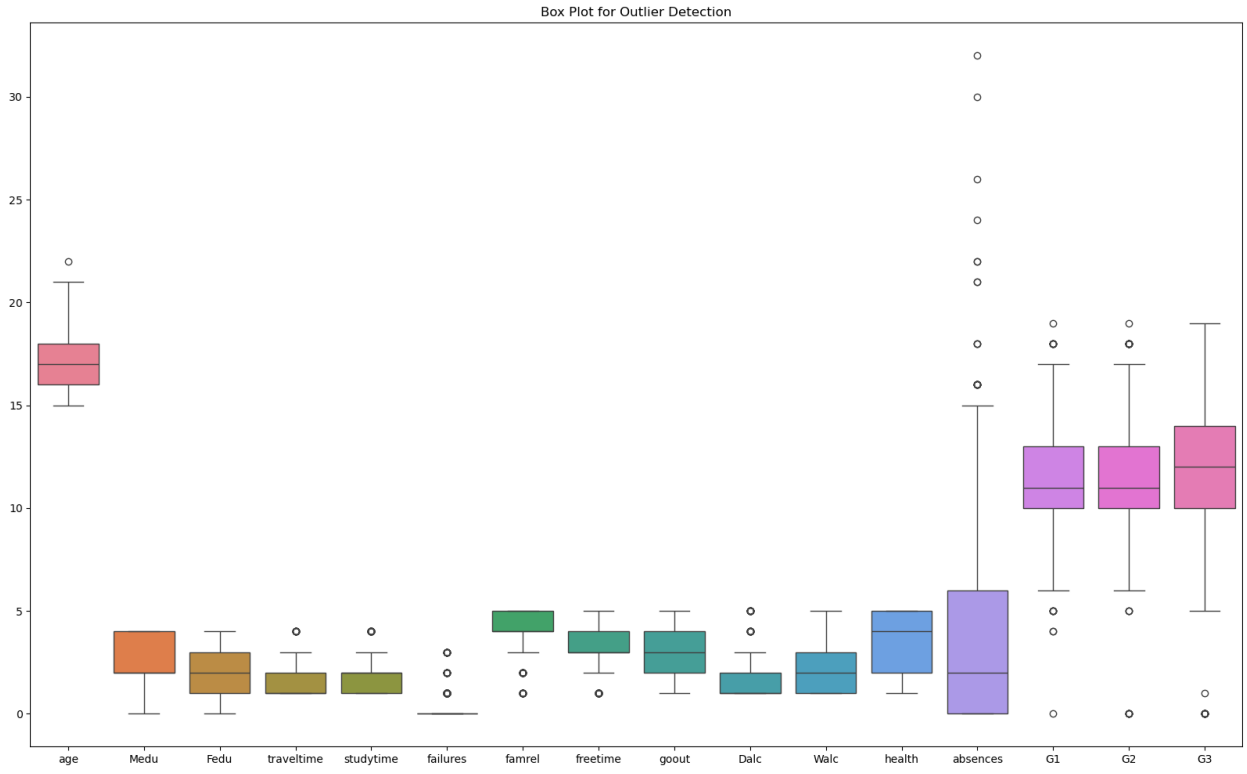
```
df.duplicated()
```

```
0      False
1      False
2      False
3      False
4      False
       ...
644    False
645    False
646    False
647    False
648    False
Length: 649, dtype: bool
```

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

```
0
```

```
plt.figure(figsize=(20, 12))
sns.boxplot(data=df)
plt.title('Box Plot for Outlier Detection')
plt.show()
```




```
df.columns
```

```
Index(['school', 'sex', 'age', 'address', 'famsize', 'Pstatus',
'Medu', 'Fedu',
```

```
       'Mjob', 'Fjob', 'reason', 'guardian', 'traveltime',
'studytime',
       'failures', 'schoolsup', 'famsup', 'paid', 'activities',
'nursery',
       'higher', 'internet', 'romantic', 'famrel', 'freetime',
'goout', 'Dalc',
       'Walc', 'health', 'absences', 'G1', 'G2', 'G3'],
      dtype='object')
```

Task 2: Statistical Analysis

1. Compute basic statistics (mean, median, variance, skewness, kurtosis).

```
mean_age = df['age'].mean()
print('Mean of age:',mean_age)
```

```
Mean of age: 16.7442218798151
```

```
mean_traveltime = df['traveltime'].mean()
print('Mean of TravelTime:',mean_traveltime)
```

```
Mean of TravelTime: 1.568567026194145
```

```
median_age = df['age'].median()
print('Median of ages:',median_age)
```

```
Median of ages: 17.0
```

```
mode_age = df['age'].mode()
print('Mode of age:',mode_age)
```

```
Mode of age: 0    17
Name: age, dtype: int64
```

```
from scipy.stats import skew
from scipy.stats import kurtosis
print("Mean:\n", df.mean(numeric_only=True))
print("Median:\n", df.median(numeric_only=True))
print("Variance:\n", df.var(numeric_only=True))
print("Skewness:\n", df.skew(numeric_only=True))
print("Kurtosis:\n", df.kurtosis(numeric_only=True))
```

```
Mean:
 age           16.744222
Medu           2.514638
Fedu           2.306626
traveltime     1.568567
studytime      1.930663
failures       0.221880
famrel         3.930663
freetime       3.180277
```

```
goout           3.184900
Dalc            1.502311
Walc            2.280431
health          3.536210
absences        3.659476
G1             11.399076
G2             11.570108
G3             11.906009
dtype: float64
Median:
 age            17.0
Medu            2.0
Fedu            2.0
traveltime      1.0
studytime       2.0
failures        0.0
famrel          4.0
freetime        3.0
goout           3.0
Dalc            1.0
Walc            2.0
health          4.0
absences        2.0
G1             11.0
G2             11.0
G3             12.0
dtype: float64
Variance:
 age             1.483859
Medu            1.287208
Fedu            1.209848
traveltime      0.560492
studytime       0.688086
failures        0.351928
famrel          0.913395
freetime        1.104796
goout           1.382426
Dalc            0.855319
Walc            1.649632
health          2.091665
absences       21.536642
G1              7.536481
G2              8.489290
G3             10.437140
dtype: float64
Skewness:
 age            0.416795
Medu           -0.029950
Fedu            0.215343
```

```
traveltime    1.247648
studytime     0.699619
failures      3.092699
famrel       -1.105934
freetime     -0.181277
goout        -0.008580
Dalc          2.141913
Walc          0.635904
health       -0.500656
absences      2.020694
G1           -0.002774
G2           -0.360283
G3           -0.912909
dtype: float64
Kurtosis:
 age           0.071509
Medu         -1.260619
Fedu         -1.109241
traveltime    1.108865
studytime     0.037846
failures      9.824409
famrel        1.348973
freetime     -0.396959
goout        -0.865454
Dalc          4.349297
Walc         -0.770689
health       -1.121175
absences      5.781078
G1            0.036638
G2            1.662465
G3            2.712204
dtype: float64
```

```
if df['age'].isin(['0']).any():
    print("Value found!")
else:
    print("Value not found.")
```

```
Value not found.
```

```
skewness = skew(df['age'])
print("Skewness:", skewness)
```

```
Skewness: 0.41583144316169546
```

```
kurtosis = kurtosis(df['age'])
print("Kurtosis:", kurtosis)
```

```
Kurtosis: 0.06172808922743078
```

2. Perform a correlation analysis to study relationships between features.

```
corr = df.corr(numeric_only=True)
corr
```

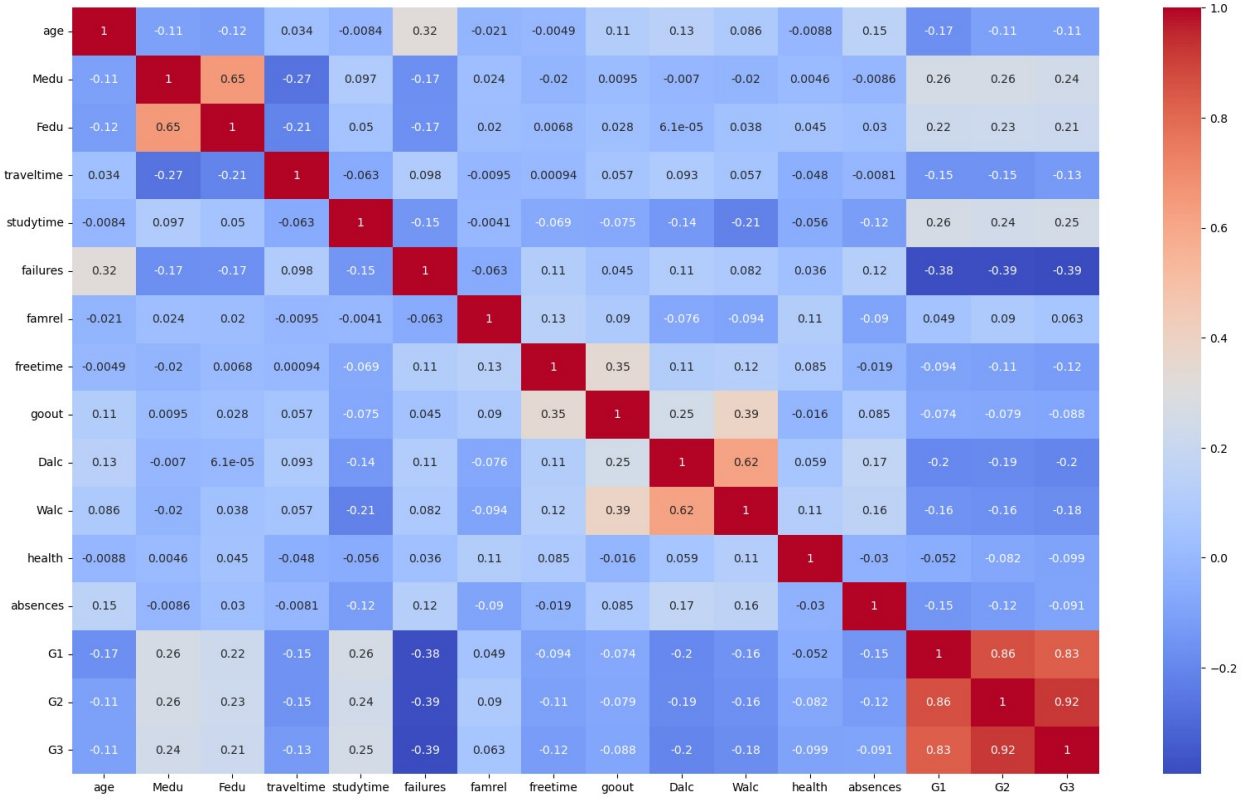
```
                 age      Medu      Fedu  traveltime  studytime
failures  \
age         1.000000 -0.107832 -0.121050    0.034490  -0.008415
0.319968
Medu       -0.107832  1.000000  0.647477   -0.265079   0.097006 -
0.172210
Fedu       -0.121050  0.647477  1.000000   -0.208288   0.050400 -
0.165915
traveltime  0.034490 -0.265079 -0.208288    1.000000  -0.063154
0.097730
studytime  -0.008415  0.097006  0.050400   -0.063154   1.000000 -
0.147441
failures    0.319968 -0.172210 -0.165915    0.097730  -0.147441
1.000000
famrel     -0.020559  0.024421  0.020256   -0.009521  -0.004127 -
0.062645
freetime   -0.004910 -0.019686  0.006841    0.000937  -0.068829
0.108995
goout       0.112805  0.009536  0.027690    0.057454  -0.075442
0.045078
Dalc        0.134768 -0.007018  0.000061    0.092824  -0.137585
0.105949
Walc        0.086357 -0.019766  0.038445    0.057007  -0.214925
0.082266
health     -0.008750  0.004614  0.044910   -0.048261  -0.056433
0.035588
absences    0.149998 -0.008577  0.029859   -0.008149  -0.118389
0.122779
G1         -0.174322  0.260472  0.217501   -0.154120   0.260875 -
0.384210
G2         -0.107119  0.264035  0.225139   -0.154489   0.240498 -
0.385782
G3         -0.106505  0.240151  0.211800   -0.127173   0.249789 -
0.393316

              famrel  freetime     goout      Dalc      Walc    health
\
age        -0.020559 -0.004910  0.112805  0.134768  0.086357 -0.008750

Medu        0.024421 -0.019686  0.009536 -0.007018 -0.019766  0.004614

Fedu        0.020256  0.006841  0.027690  0.000061  0.038445  0.044910

traveltime -0.009521  0.000937  0.057454  0.092824  0.057007 -0.048261

studytime  -0.004127 -0.068829 -0.075442 -0.137585 -0.214925 -0.056433
```

```
failures   -0.062645  0.108995  0.045078  0.105949  0.082266  0.035588

famrel      1.000000  0.129216  0.089707 -0.075767 -0.093511  0.109559

freetime    0.129216  1.000000  0.346352  0.109904  0.120244  0.084526

goout       0.089707  0.346352  1.000000  0.245126  0.388680 -0.015741

Dalc       -0.075767  0.109904  0.245126  1.000000  0.616561  0.059067

Walc       -0.093511  0.120244  0.388680  0.616561  1.000000  0.114988

health      0.109559  0.084526 -0.015741  0.059067  0.114988  1.000000

absences   -0.089534 -0.018716  0.085374  0.172952  0.156373 -0.030235

G1          0.048795 -0.094497 -0.074053 -0.195171 -0.155649 -0.051647

G2          0.089588 -0.106678 -0.079469 -0.189480 -0.164852 -0.082179

G3          0.063361 -0.122705 -0.087641 -0.204719 -0.176619 -0.098851
```

```
             absences        G1        G2        G3
age          0.149998 -0.174322 -0.107119 -0.106505
Medu        -0.008577  0.260472  0.264035  0.240151
Fedu         0.029859  0.217501  0.225139  0.211800
traveltime -0.008149 -0.154120 -0.154489 -0.127173
studytime  -0.118389  0.260875  0.240498  0.249789
failures    0.122779 -0.384210 -0.385782 -0.393316
famrel     -0.089534  0.048795  0.089588  0.063361
freetime   -0.018716 -0.094497 -0.106678 -0.122705
goout       0.085374 -0.074053 -0.079469 -0.087641
Dalc        0.172952 -0.195171 -0.189480 -0.204719
Walc        0.156373 -0.155649 -0.164852 -0.176619
health     -0.030235 -0.051647 -0.082179 -0.098851
absences    1.000000 -0.147149 -0.124745 -0.091379
G1         -0.147149  1.000000  0.864982  0.826387
G2         -0.124745  0.864982  1.000000  0.918548
G3         -0.091379  0.826387  0.918548  1.000000
```

3.Generate a correlation matrix and visualize it using a heatmap.

```
plt.figure(figsize=(20, 12))
sns.heatmap(corr,annot= True,cmap = 'coolwarm', fmt='.2g',)
plt.show()
```

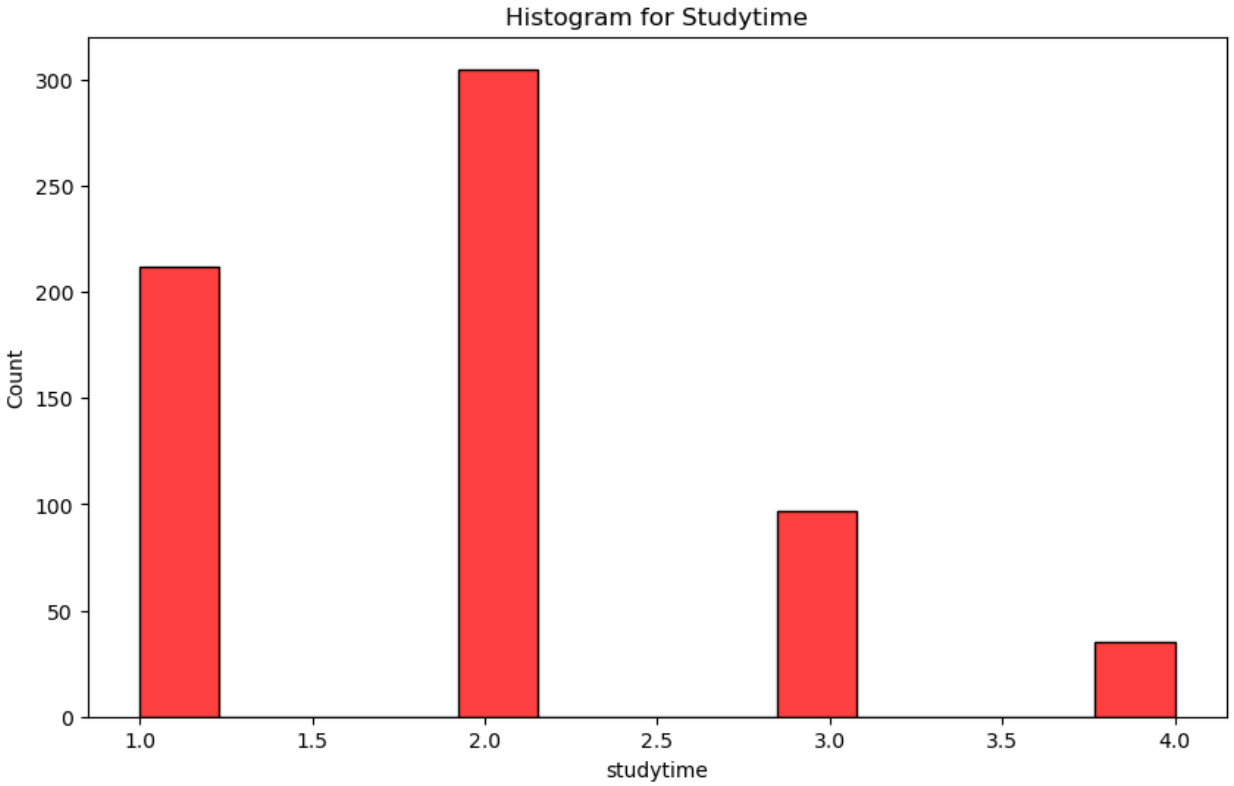


Task 3: Data Visualization

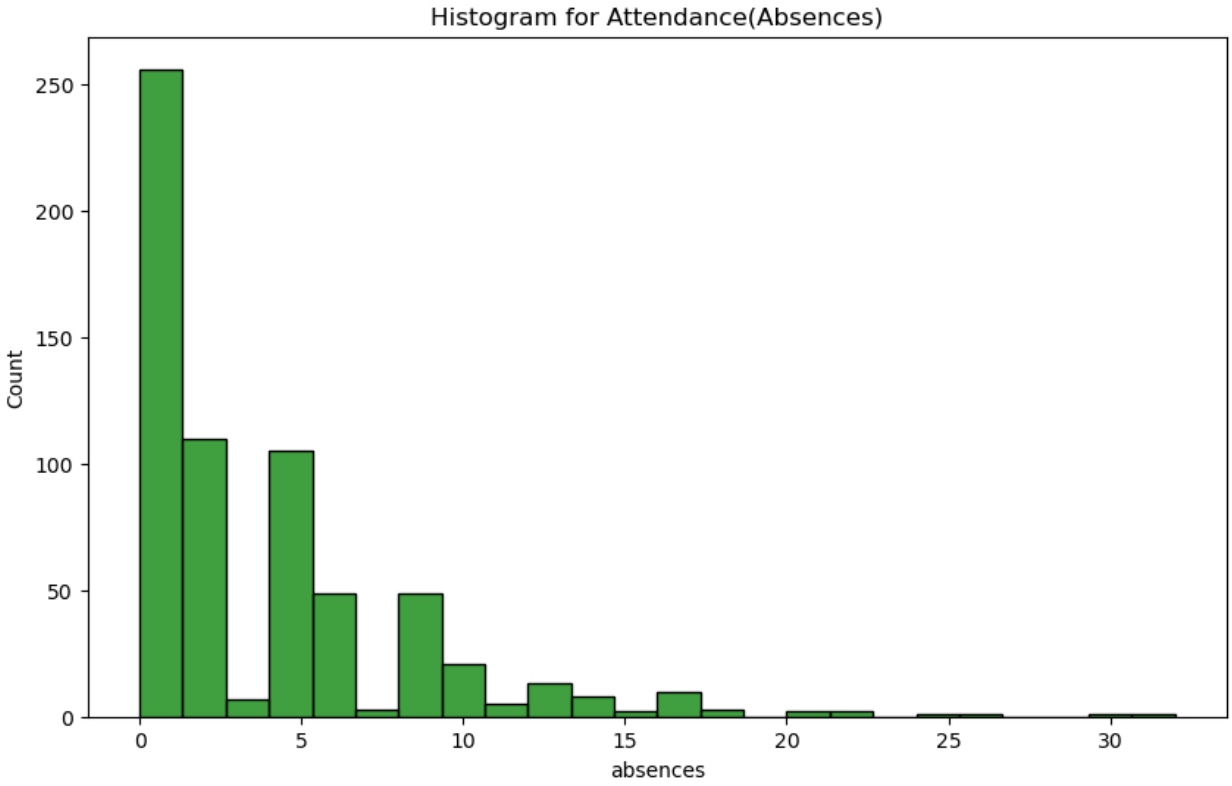
1. Univariate Analysis (analyzing individual features):

– Histograms for exam scores, study time, and attendance.

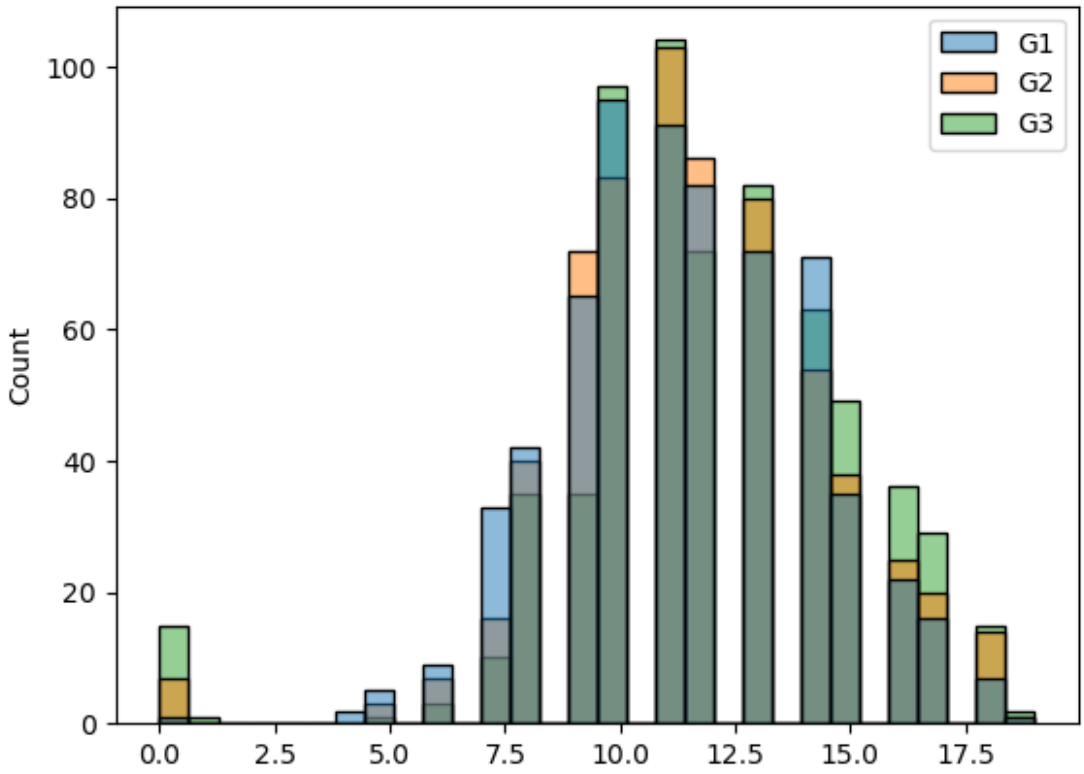
```
plt.figure(figsize=(10,6))
sns.histplot(data = df['studytime'],color = 'red')
plt.title('Histogram for Studytime')
plt.show()
```




```
plt.figure(figsize=(10,6))
sns.histplot(data = df['absences'],color = 'green')
plt.title('Histogram for Attendance(Absences)')
plt.show()
```

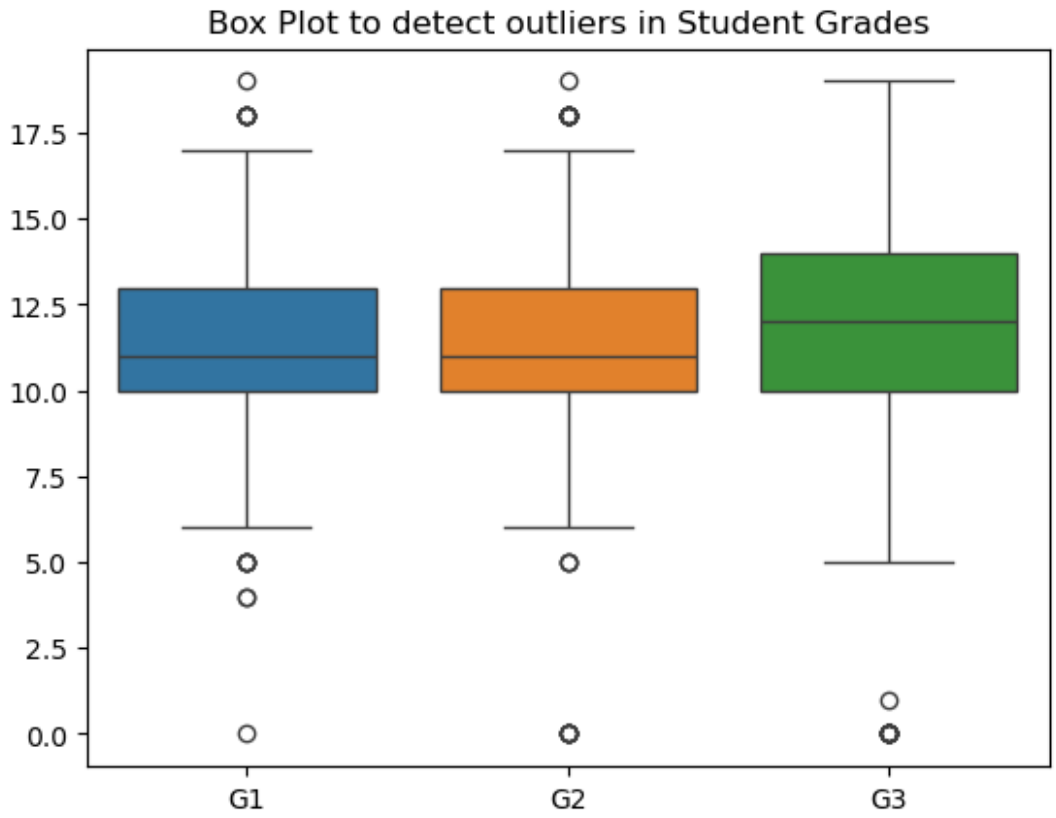



```
sns.histplot(data = df[['G1','G2','G3']])
plt.show()
```



– Box plots to detect outliers in student grades.

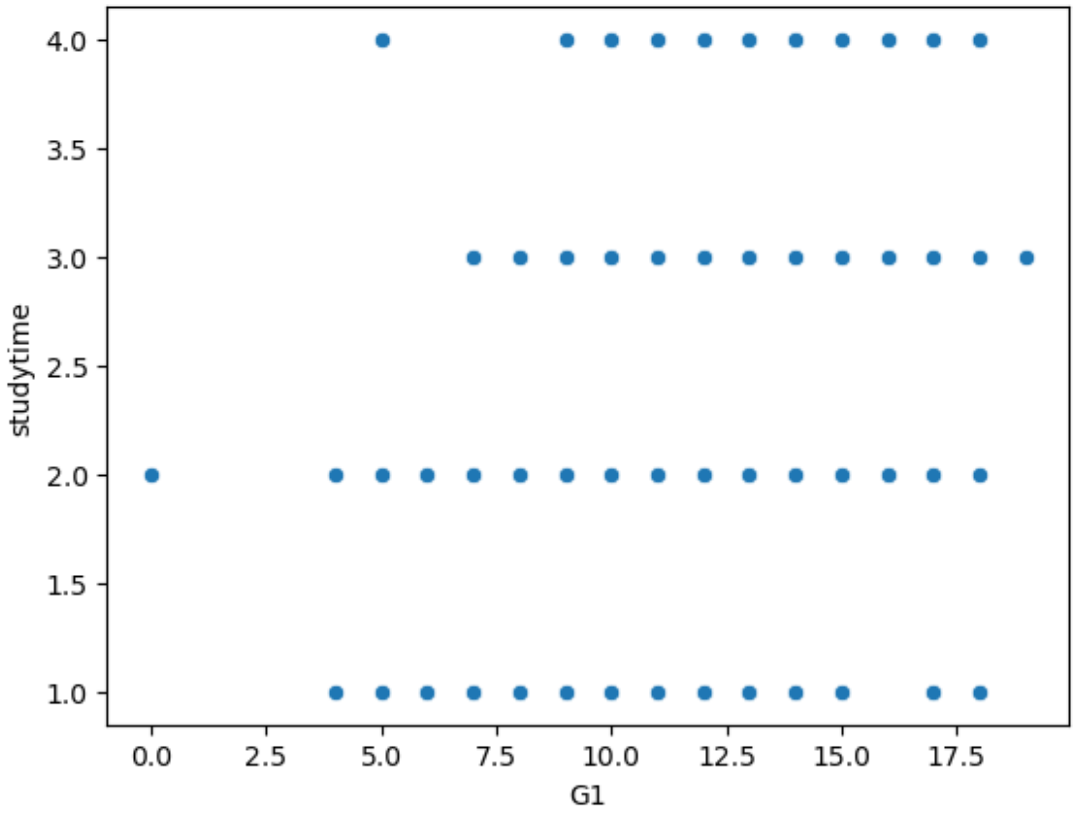
```
sns.boxplot(data = df[['G1','G2','G3']])
plt.title('Box Plot to detect outliers in Student Grades')
plt.show()
```



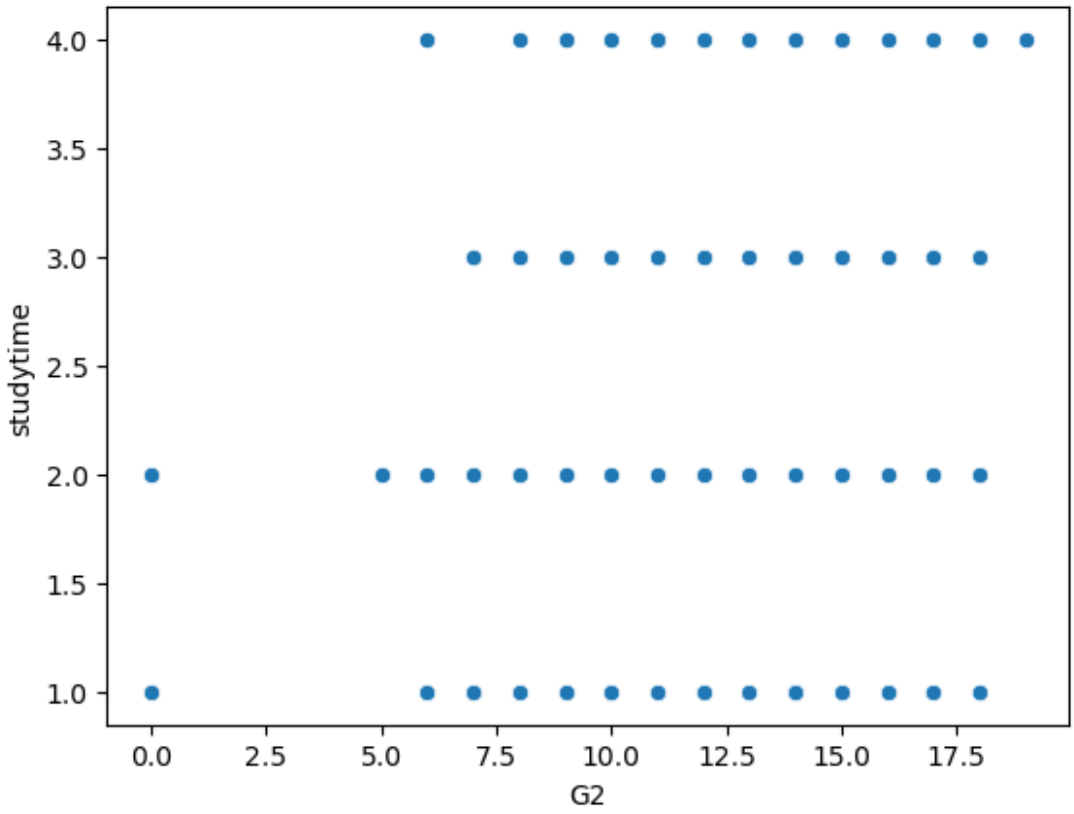
2. Bivariate Analysis (analyzing relationships between features):

– Scatter plots to visualize the relationship between study time and final grades.

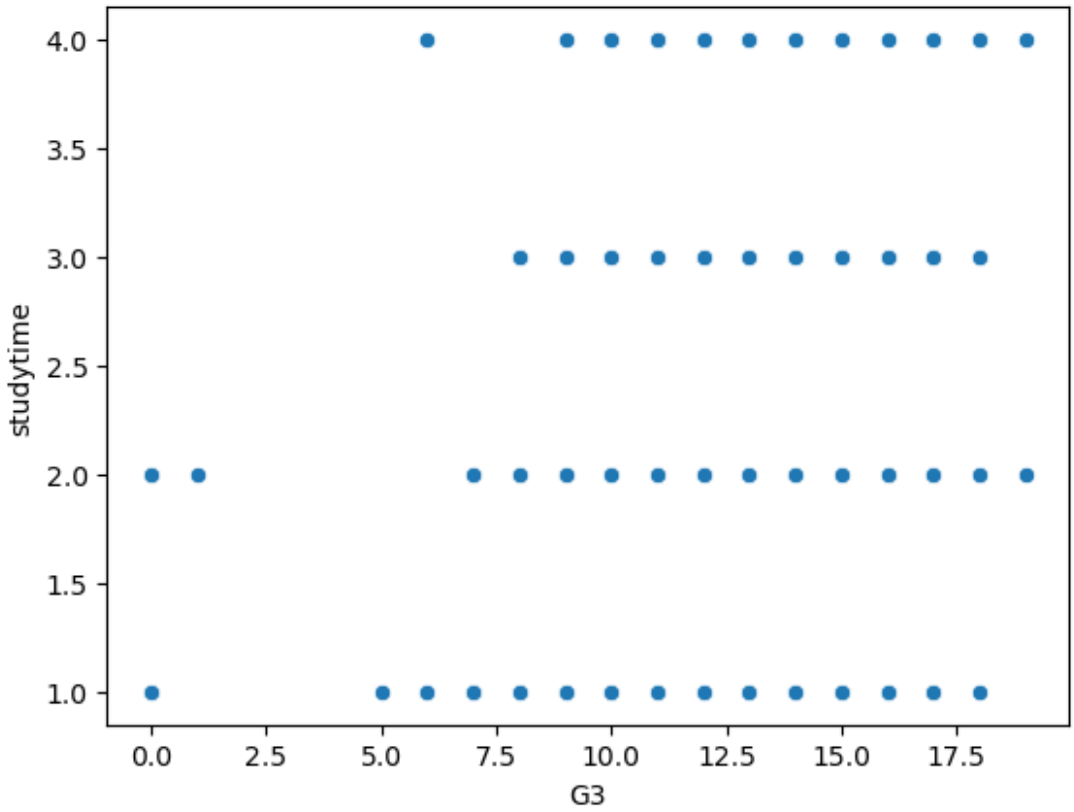
```
sns.scatterplot(y='studytime', x='G1', data=df)
plt.show()
```




```
sns.scatterplot(y='studytime', x='G2', data=df)
plt.show()
```

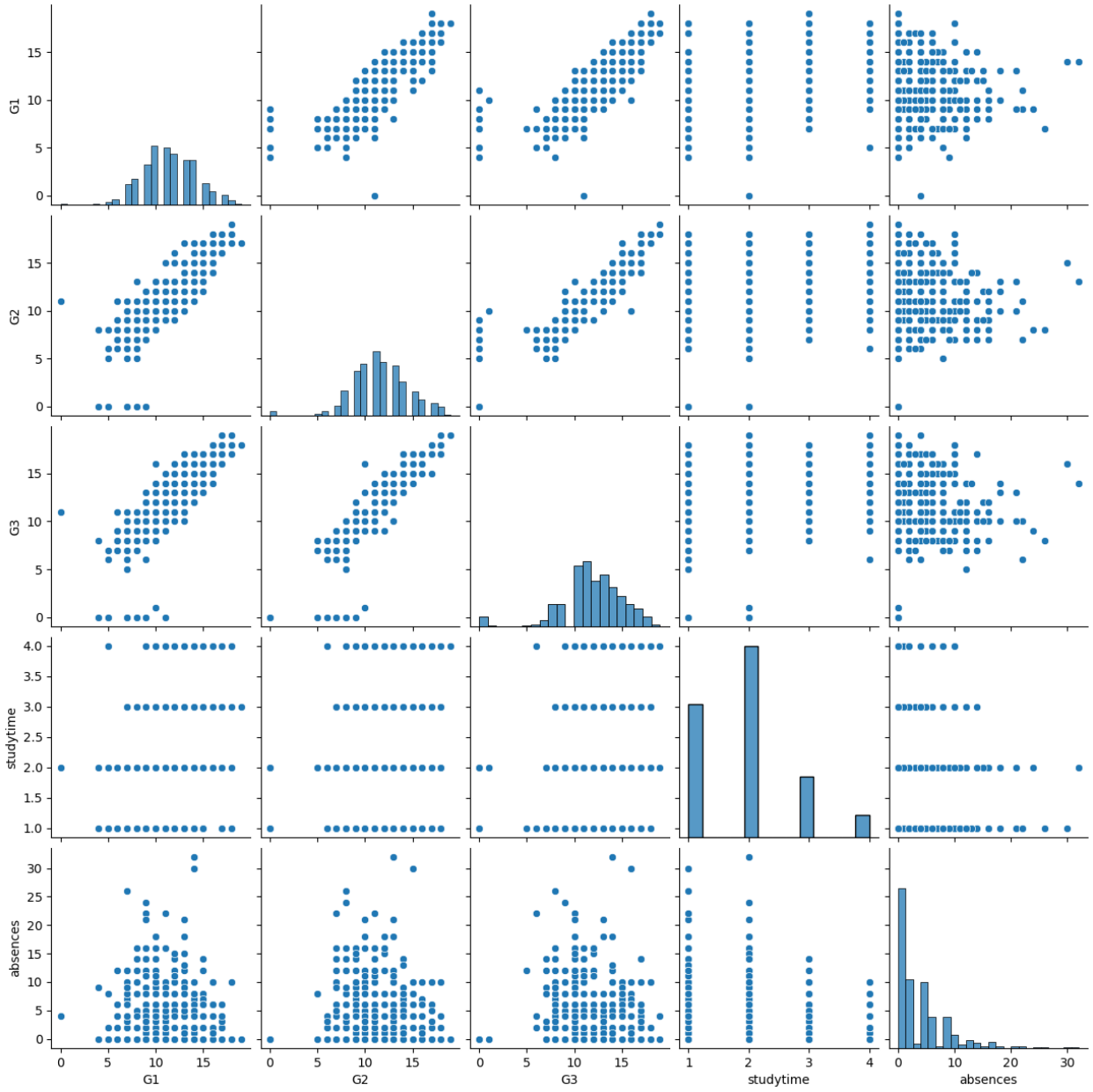



```
sns.scatterplot(y='studytime', x='G3', data=df)
plt.show()
```



– Pair plots using Seaborn to explore multiple relationships.

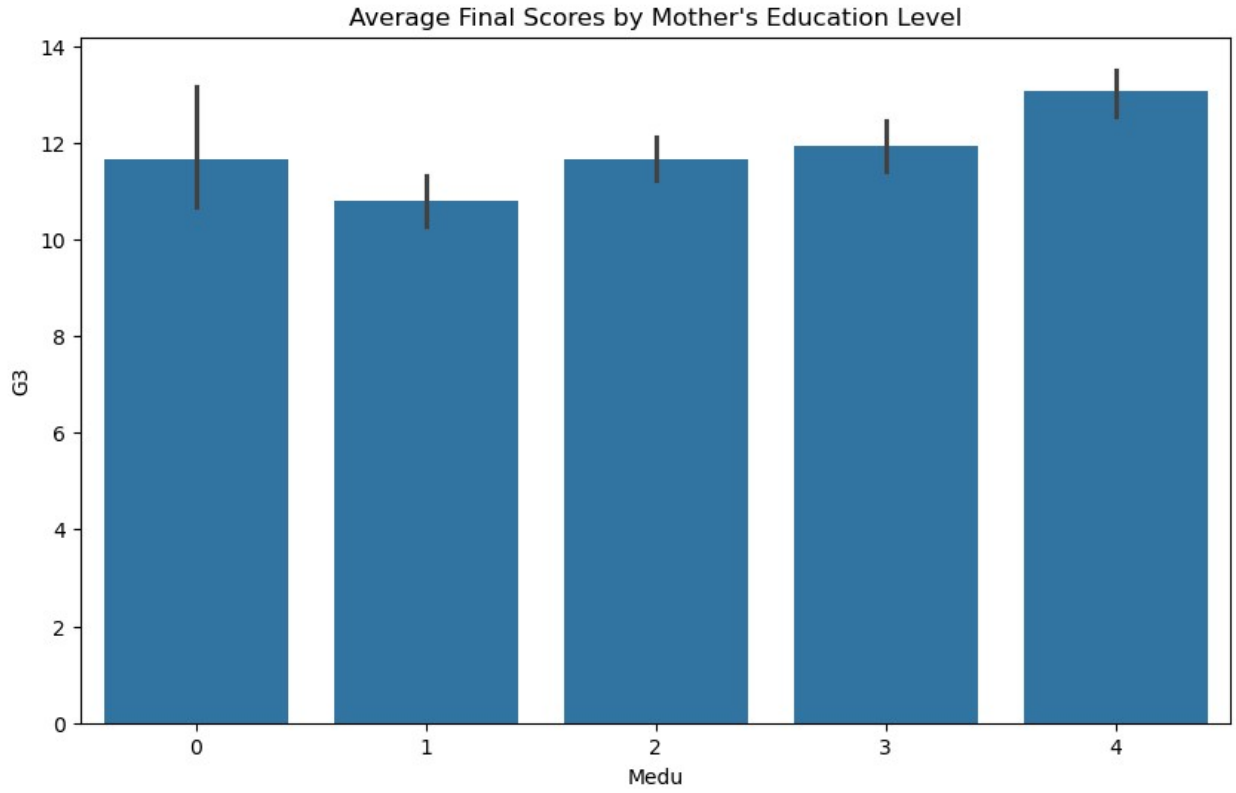
```
sns.pairplot(df[['G1','G2','G3','studytime','absences']])
plt.show()
```



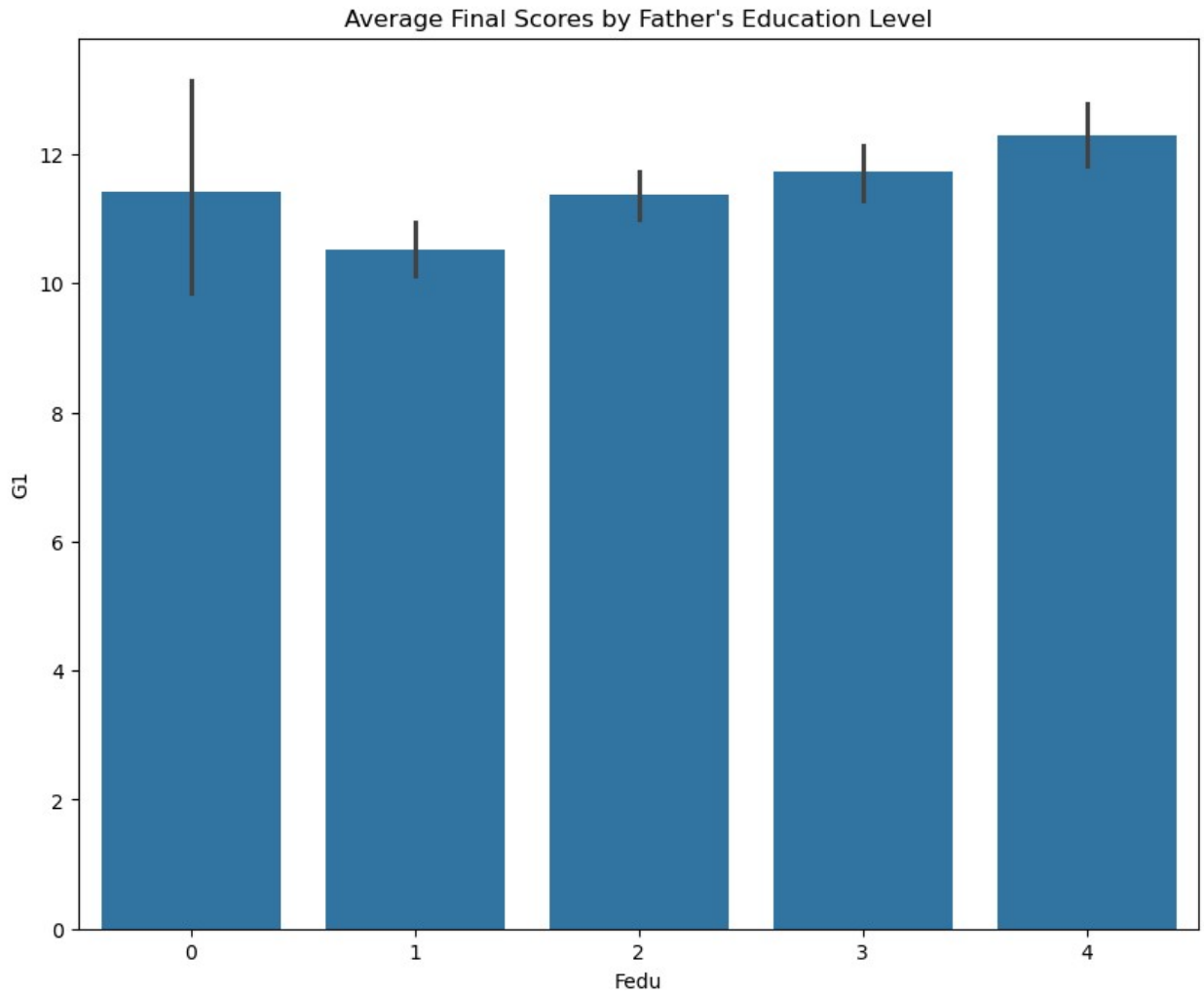
3. Categorical Data Visualization:

– Bar charts to analyze the impact of parental education level on student performance.

```
plt.figure(figsize=(10,6))
sns.barplot(x='Medu', y='G3', data=df)
plt.title("Average Final Scores by Mother's Education Level")
plt.show()
```

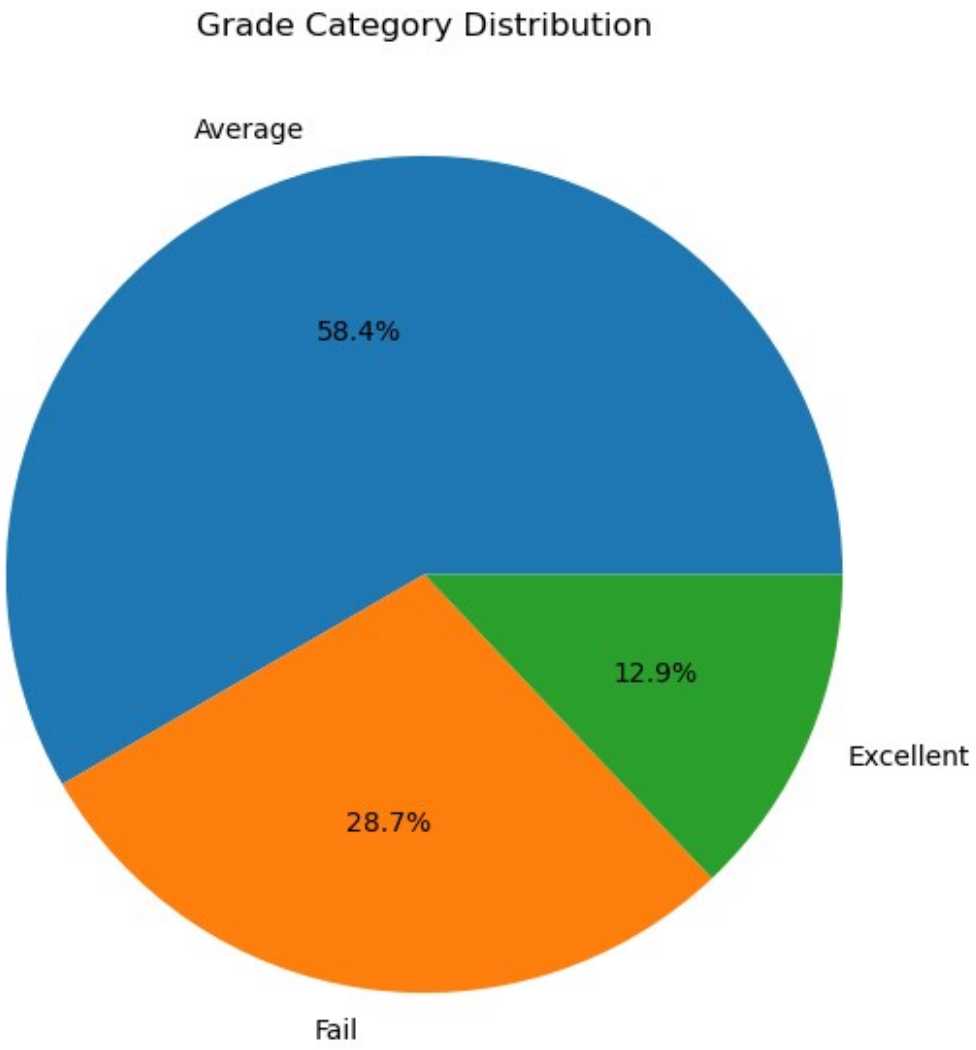



```
plt.figure(figsize=(10,8))
sns.barplot(x='Fedu', y='G1', data=df)
plt.title("Average Final Scores by Father's Education Level")
plt.show()
```



– Pie charts to show percentage distribution of grade categories.

```
grade_categories = pd.cut(df["G3"], bins=[0, 10, 15, 20],
labels=["Fail", "Average", "Excellent"])
grade_distribution = grade_categories.value_counts()
plt.figure(figsize=(7, 7))
plt.pie(grade_distribution, labels=grade_distribution.index,
autopct='%1.1f%%')
plt.title("Grade Category Distribution")
plt.show()
```



Task 4: Insights and Report Generation

1. Performance Trends: Higher parental education often correlates with better student grades.
2. Feature Relationships: Study time positively impacts final grades but shows diminishing returns. Attendance has a strong negative correlation with performance.
3. Outliers: Outliers are present in Grades G1,G2,G3
4. Feature Selection: Parental education (Medu, Fedu) Study time (studytime) Prior grades (G1, G2,G3) Attendance (absences)