Importing Necessary Libraries

```
import pandas as pd
import numpy as np
```

```
C:\Users\aakas\AppData\Local\Temp\ipykernel_3720\2162656668.py:1:
DeprecationWarning:
Pyarrow will become a required dependency of pandas in the next major
release of pandas (pandas 3.0),
(to allow more performant data types, such as the Arrow string type,
and better interoperability with other libraries)
but was not found to be installed on your system.
If this would cause problems for you,
please provide us feedback at
https://github.com/pandas-dev/pandas/issues/54466

  import pandas as pd
```

```
edu = pd.read_csv("assignment2.csv")
```

```
pd.set_option('display.max_rows', None)
pd.set_option('display.max_columns', None)
```

```
edu.head()
```

```
  gender NationalITy PlaceofBirth     StageID GradeID SectionID Topic
\
0      M          KW       KuwaIT  lowerlevel    G-04         A    IT

1      M          KW       KuwaIT  lowerlevel    G-04         A    IT

2      M          KW       KuwaIT  lowerlevel    G-04         A    IT

3      M          KW       KuwaIT  lowerlevel    G-04         A    IT

4      M          KW       KuwaIT  lowerlevel    G-04         A    IT


  Semester Relation raisedhands  VisITedResources
AnnouncementsView  \
0        F   Father          15                16                  2

1        F   Father          20                20                  3

2        F   Father          10                 7                  0

3        F   Father          30                25                  5

4        F   Father          40                50                 12


   Discussion ParentAnsweringSurvey ParentschoolSatisfaction  \
```

```
0          20                   Yes                       Good
1          25                   Yes                       Good
2          30                    No                        Bad
3          35                    No                        Bad
4          50                    No                        Bad

  StudentAbsenceDays Class
0            Under-7     M
1            Under-7     M
2            Above-7     L
3            Above-7     L
4            Above-7     M
```

```
edu.replace("?", np.nan, inplace=True)
```

```
edu.isna().sum()
```

```
gender                       0
NationalITy                  0
PlaceofBirth                 0
StageID                      0
GradeID                      2
SectionID                    0
Topic                        2
Semester                     0
Relation                     8
raisedhands                  2
VisITedResources             0
AnnouncementsView            0
Discussion                   0
ParentAnsweringSurvey        0
ParentschoolSatisfaction     0
StudentAbsenceDays           0
Class                        0
dtype: int64
```

Removing Missing Values

```
edu.dropna(subset=["Relation"], inplace=True)
edu.dropna(subset=["GradeID"], inplace=True)

edu["raisedhands"] = edu["raisedhands"].astype(float)

mean_raised_hand = edu["raisedhands"].mean()

edu["raisedhands"].replace(np.nan, value=mean_raised_hand,
inplace=True)
```

```
C:\Users\aakas\AppData\Local\Temp\ipykernel_3720\810218330.py:3:
FutureWarning: A value is trying to be set on a copy of a DataFrame or
Series through chained assignment using an inplace method.
```

```
The behavior will change in pandas 3.0. This inplace method will never
work because the intermediate object on which we are setting values
always behaves as a copy.

For example, when doing 'df[col].method(value, inplace=True)', try
using 'df.method({col: value}, inplace=True)' or df[col] =
df[col].method(value) instead, to perform the operation inplace on the
original object.


  edu["raisedhands"].replace(np.nan, value=mean_raised_hand,
inplace=True)
```

```
freq_topic = edu["Topic"].value_counts()
print(freq_topic)
```

```
Topic
IT           93
French       62
Arabic       57
Science      51
English      42
Biology      30
Spanish      25
Chemistry    24
Geology      24
Quran        21
Math         20
History      19
Name: count, dtype: int64
```

```
edu["Topic"].replace(np.nan, value="IT", inplace=True)
```

```
C:\Users\aakas\AppData\Local\Temp\ipykernel_3720\2982627878.py:1:
FutureWarning: A value is trying to be set on a copy of a DataFrame or
Series through chained assignment using an inplace method.
The behavior will change in pandas 3.0. This inplace method will never
work because the intermediate object on which we are setting values
always behaves as a copy.

For example, when doing 'df[col].method(value, inplace=True)', try
using 'df.method({col: value}, inplace=True)' or df[col] =
df[col].method(value) instead, to perform the operation inplace on the
original object.


  edu["Topic"].replace(np.nan, value="IT", inplace=True)
```

```
edu.isna().sum()
```

```
gender                      0
NationalITy                 0
PlaceofBirth                0
StageID                     0
GradeID                     0
SectionID                   0
Topic                       0
Semester                    0
Relation                    0
raisedhands                 0
VisITedResources            0
AnnouncementsView           0
Discussion                  0
ParentAnsweringSurvey       0
ParentschoolSatisfaction    0
StudentAbsenceDays          0
Class                       0
dtype: int64
```

Removing Outliers

```
freq_semester = edu["Semester"].value_counts()
print(freq_semester)
```

```
Semester
F    240
S    228
T      2
Name: count, dtype: int64
```

```
edu["Semester"].replace(freq_semester.index[-1],
value=freq_semester.index[0], inplace=True)
```

```
C:\Users\aakas\AppData\Local\Temp\ipykernel_3720\3023071285.py:1:
FutureWarning: A value is trying to be set on a copy of a DataFrame or
Series through chained assignment using an inplace method.
The behavior will change in pandas 3.0. This inplace method will never
work because the intermediate object on which we are setting values
always behaves as a copy.

For example, when doing 'df[col].method(value, inplace=True)', try
using 'df.method({col: value}, inplace=True)' or df[col] =
df[col].method(value) instead, to perform the operation inplace on the
original object.


  edu["Semester"].replace(freq_semester.index[-1],
value=freq_semester.index[0], inplace=True)
```

```
freq_semester = edu["Semester"].value_counts()
print(freq_semester)
```

```
Semester
F    242
S    228
Name: count, dtype: int64
```

```
freq_parent_answer = edu["ParentAnsweringSurvey"].value_counts()
print(freq_parent_answer)
```

```
ParentAnsweringSurvey
Yes    262
No     208
Name: count, dtype: int64
```

```
edu.drop(columns=["ParentAnsweringSurvey"], inplace=True)
```

```
edu.head()
```

```
  gender NationalITy PlaceofBirth     StageID GradeID SectionID Topic
\
0      M          KW       KuwaIT  lowerlevel    G-04         A    IT

1      M          KW       KuwaIT  lowerlevel    G-04         A    IT

2      M          KW       KuwaIT  lowerlevel    G-04         A    IT

3      M          KW       KuwaIT  lowerlevel    G-04         A    IT

4      M          KW       KuwaIT  lowerlevel    G-04         A    IT


  Semester Relation  raisedhands  VisITedResources  AnnouncementsView
\
0        F   Father         15.0                16                  2

1        F   Father         20.0                20                  3

2        F   Father         10.0                 7                  0

3        F   Father         30.0                25                  5

4        F   Father         40.0                50                 12


   Discussion ParentschoolSatisfaction StudentAbsenceDays Class
0          20                     Good            Under-7     M
1          25                     Good            Under-7     M
2          30                      Bad            Above-7     L
3          35                      Bad            Above-7     L
4          50                      Bad            Above-7     M
```

Normalization

```
max_raisedhands = edu["raisedhands"].max()
max_visitedresources = edu["VisITedResources"].max()

edu["raisedhands"] = edu["raisedhands"]/max_raisedhands
edu["VisITedResources"] = edu["VisITedResources"]/max_visitedresources

freq_gradeid = edu["GradeID"].value_counts()
print(freq_gradeid.index.sort_values())
```

```
Index(['G-02', 'G-04', 'G-05', 'G-06', 'G-07', 'G-08', 'G-09', 'G-10',
'G-11',
       'G-12'],
      dtype='object', name='GradeID')
```

Transformation

```
# values = edu["GradeID"].unique().tolist()
# values.sort()
# i = 1
# for value in values:
#     edu["GradeID"].replace(value, i, inplace=True)
#     i+=1

gradeid_to_num = {
    "G-02":1,
    "G-04":2,
    "G-05":3,
    "G-06":4,
    "G-07":5,
    "G-08":6,
    "G-09":7,
    "G-10":8,
    "G-11":9,
    "G-12":10
}

edu["GradeID"] = edu["GradeID"].map(gradeid_to_num)

freq_stud_abs = edu["StudentAbsenceDays"].value_counts()
print(freq_stud_abs)
```

```
StudentAbsenceDays
Under-7    283
Above-7    187
Name: count, dtype: int64
```

```
student_abs = {
    "Under-7":0,
    "Above-7":1
}
```

```
edu["StudentAbsenceDays"] = edu["StudentAbsenceDays"].map(student_abs)

edu.head()
```

```
  gender NationalITy PlaceofBirth     StageID  GradeID SectionID Topic
\
0      M          KW       KuwaIT  lowerlevel        2          A    IT

1      M          KW       KuwaIT  lowerlevel        2          A    IT

2      M          KW       KuwaIT  lowerlevel        2          A    IT

3      M          KW       KuwaIT  lowerlevel        2          A    IT

4      M          KW       KuwaIT  lowerlevel        2          A    IT


  Semester Relation  raisedhands  VisITedResources  AnnouncementsView
\
0        F   Father          0.15          0.161616                  2

1        F   Father          0.20          0.202020                  3

2        F   Father          0.10          0.070707                  0

3        F   Father          0.30          0.252525                  5

4        F   Father          0.40          0.505051                 12


   Discussion ParentschoolSatisfaction  StudentAbsenceDays Class
0          20                     Good                   0     M
1          25                     Good                   0     M
2          30                      Bad                   1     L
3          35                      Bad                   1     L
4          50                      Bad                   1     M
```

Binning

```
num_bins = 5
```

Equal Width Binning

```
bin_label_ewb = ["very low", "low","medium","high","very high"]

edu["Discussion-Equal-Width-Bin"] = pd.cut(edu["Discussion"], bins =
num_bins, labels=bin_label_ewb)
```

Equal Frequency Binning

```
bin_label_efb = ["A", "B", "C", "D", "E"]

edu["Discussion-Equal-Frequency-Bin"] = pd.qcut(edu["Discussion"], q =
num_bins, labels=bin_label_efb)
```

Custom Binning

```
bin_edges = [18, 36, 57, 69, 75, 93]
bin_label_cb = ["P","Q","R","S","T"]

edu["Discussion-Custom-Binning"] = pd.cut(edu["Discussion"],
bins=bin_edges, labels=bin_label_cb)

edu.head()
```

```
  gender NationalITy PlaceofBirth     StageID  GradeID SectionID Topic
\
0      M          KW       KuwaIT  lowerlevel        2         A    IT

1      M          KW       KuwaIT  lowerlevel        2         A    IT

2      M          KW       KuwaIT  lowerlevel        2         A    IT

3      M          KW       KuwaIT  lowerlevel        2         A    IT

4      M          KW       KuwaIT  lowerlevel        2         A    IT


  Semester Relation  raisedhands  VisITedResources  AnnouncementsView
\
0        F   Father         0.15          0.161616                  2

1        F   Father         0.20          0.202020                  3

2        F   Father         0.10          0.070707                  0

3        F   Father         0.30          0.252525                  5

4        F   Father         0.40          0.505051                 12


   Discussion ParentschoolSatisfaction  StudentAbsenceDays Class  \
0          20                    Good                   0     M
1          25                    Good                   0     M
2          30                     Bad                   1     L
3          35                     Bad                   1     L
4          50                     Bad                   1     M

  Discussion-Equal-Width-Bin Discussion-Equal-Frequency-Bin  \
0                   very low                            low
```

```
1                          low                                  low
2                          low                                  low
3                          low                               medium
4                       medium                                 high

  Discussion-Custom-Binning
0                  very low
1                  very low
2                  very low
3                  very low
4                       low
```

Saving as a CSV File

```
edu.to_csv("updated-assignment2.csv")
```