Assessment 2 - 43031 Python Programming for Data Processing - Autumn 2025

In [2]:
```
#Step 0: Mount Google Drive to give Collab access to the dataset

from google.colab import drive

try:
    drive.mount('/content/drive')
    print("Google Drive mounted successfully!")

except Exception as e:
    print(f"An error occurred during mounting: {e}") #Easier to understand
the error
```

```
Mounted at /content/drive
Google Drive mounted successfully!
```

In [3]:
```
# Step 1: Read the Kaggle dataset (csv) using read_csv function in Pandas

import pandas as pd

dataset_path = '/content/drive/MyDrive/Python
Class/Students_Grading_Dataset.csv' #Path from drive set

try:
       df = pd.read_csv(dataset_path, header=0)
       print("CSV file loaded successfully!")

except Exception as e:
       print(f"An unexpected error occurred: {e}")
```

```
CSV file loaded successfully!
```

In [6]:
```
# Step 2: Prove it works by displaying top 5 and bottom 5 rows

try:
    print("Top 5 rows:\n")
    print(df.head())

    print("\nBottom 5 rows:\n")
    print(df.tail())

except Exception as e:
    print(f"An error occurred while displaying the data: {e}")
```

```
Top 5 rows:

  Student_ID First_Name Last_Name                    Email  Gender
Age  \
0      S1000       Omar  Williams  student0@university.com  Female
22
1      S1001      Maria     Brown  student1@university.com    Male
18
2      S1002      Ahmed     Jones  student2@university.com    Male
24
3      S1003       Omar  Williams  student3@university.com  Female
24
4      S1004       John     Smith  student4@university.com  Female
23

    Department  Attendance (%)  Midterm_Score  Final_Score  ...  \
0  Engineering           52.29          55.03        57.82  ...
1  Engineering           97.27          97.23        45.80  ...
2     Business           57.19          67.05        93.68  ...
3  Mathematics           95.15          47.79        80.63  ...
4           CS           54.18          46.59        78.89  ...

   Projects_Score  Total_Score  Grade  Study_Hours_per_Week  \
0           85.90        56.09      F                   6.2
1           55.65        50.64      A                  19.0
2           73.79        70.30      D                  20.7
3           92.12        61.63      A                  24.8
4           68.42        66.13      F                  15.4

   Extracurricular_Activities Internet_Access_at_Home
Parent_Education_Level  \
0                          No                   Yes
High School
1                          No                   Yes
NaN
2                          No                   Yes
Master's
3                         Yes                   Yes
High School
4                         Yes                   Yes
High School
```

```
  Family_Income_Level Stress_Level (1-10) Sleep_Hours_per_Night
0              Medium                   5                    4.7
1              Medium                   4                    9.0
2                 Low                   6                    6.2
3                High                   3                    6.7
4                High                   2                    7.1

[5 rows x 23 columns]

Bottom 5 rows:

     Student_ID First_Name Last_Name                       Email
Gender  Age  \
4995      S5995      Ahmed     Jones  student4995@university.com
Male   19
4996      S5996       Emma     Brown  student4996@university.com
Male   19
4997      S5997       John     Brown  student4997@university.com
Female   24
4998      S5998       Sara     Davis  student4998@university.com
Male   23
4999      S5999      Maria     Brown  student4999@university.com
Female   21

       Department  Attendance (%)  Midterm_Score  Final_Score  ...  \
4995     Business             NaN          82.15        60.33  ...
4996     Business           65.11          86.31        49.80  ...
4997           CS           87.54          63.55        64.21  ...
4998           CS           92.56          79.79        94.28  ...
4999  Engineering           83.92          83.24        53.47  ...

      Projects_Score  Total_Score  Grade  Study_Hours_per_Week  \
4995           58.42        85.21      D                  25.5
4996           60.87        95.96      C                   5.0
4997           82.65        54.25      A                  24.8
4998           94.29        55.84      A                  16.1
4999           69.25        77.86      F                  29.2

      Extracurricular_Activities Internet_Access_at_Home  \
4995                         No                    Yes
4996                         No                    Yes
4997                        Yes                     No
4998                        Yes                    Yes
4999                         No                    Yes

      Parent_Education_Level Family_Income_Level Stress_Level (1-10)
\
4995            High School                Low                  10
4996                    NaN             Medium                   4
4997            High School             Medium                   4
4998             Bachelor's                Low                   1
4999                    PhD                Low                   2
```

```
     Sleep_Hours_per_Night
4995                    8.3
4996                    4.0
4997                    6.3
4998                    8.4
4999                    6.1

[5 rows x 23 columns]
```

In [7]:
```
# Step 3: Provide insights into the dataset by displaying its size and the
data types of its columns

try:
    print("Dataset size:", df.shape)
    print("\nData types of columns:")
    print(df.dtypes)

except Exception as e:
    print(f"An error occurred while displaying dataset insights: {e}")
```

```
Dataset size: (5000, 23)

Data types of columns:
Student_ID                    object
First_Name                    object
Last_Name                     object
Email                         object
Gender                        object
Age                            int64
Department                    object
Attendance (%)               float64
Midterm_Score                float64
Final_Score                  float64
Assignments_Avg              float64
Quizzes_Avg                  float64
Participation_Score          float64
Projects_Score               float64
Total_Score                  float64
Grade                         object
Study_Hours_per_Week         float64
Extracurricular_Activities    object
Internet_Access_at_Home       object
Parent_Education_Level        object
Family_Income_Level           object
Stress_Level (1-10)            int64
Sleep_Hours_per_Night        float64
dtype: object
```

In [8]:
```
#Step 4: Select the required columns and filter out the data you don't
need.

try:
    Sub_df = df[(df['Department'] == 'Engineering') & (df['Attendance (%)']
< 85) & (df['Study_Hours_per_Week'] <= 18)]
[['Student_ID','Age','Department','Attendance
(%)','Study_Hours_per_Week','Final_Score','Stress_Level (1-10)']]
    print("Filtered DataFrame:\n")
    print(Sub_df)

    print("\nFiltered DataFrame size:", Sub_df.shape)

except Exception as e:
    print(f"An unexpected error occurred: {e}")
```

```
Filtered DataFrame:

     Student_ID  Age   Department  Attendance (%)
Study_Hours_per_Week  \
0         S1000   22  Engineering           52.29
6.2
9         S1009   22  Engineering           64.01
9.6
14        S1014   19  Engineering           72.62
5.8
26        S1026   18  Engineering           66.94
14.4
53        S1053   23  Engineering           76.60
13.9
...         ...  ...          ...             ...
...
4954      S5954   19  Engineering           70.61
10.2
4958      S5958   21  Engineering           53.19
17.6
4964      S5964   19  Engineering           65.05
6.3
4969      S5969   20  Engineering           78.08
6.5
4984      S5984   20  Engineering           58.63
14.9

      Final_Score  Stress_Level (1-10)
0           57.82                    5
9           98.47                   10
14          44.50                    4
26          42.52                   10
53          97.18                    5
...           ...                  ...
4954        86.06                    3
4958        41.24                    5
4964        76.43                    5
```

```
4969        89.29                    9
4984        51.04                    3

[464 rows x 7 columns]

Filtered DataFrame size: (464, 7)
```

In [9]:

```
# Step 5: explore the dataset comprehensively by generating summary
statistics, including frequency, mean, median, standard deviation, minimum,
maximum, and quartiles
try:
    print("Summary statistics for Sub DataFrame:\n")
    print(Sub_df.describe())


except Exception as e:
    print(f"An error occurred while generating summary statistics: {e}")
```

```
Summary statistics for Sub DataFrame:

              Age  Attendance (%)  Study_Hours_per_Week  Final_Score
\
count  464.000000      464.000000            464.000000   464.000000
mean    20.956897       67.660280             11.467241    70.669978
std      1.993043       10.097719              3.721307    17.089147
min     18.000000       50.010000              5.000000    40.620000
25%     19.000000       58.990000              8.375000    56.605000
50%     21.000000       68.095000             11.500000    71.540000
75%     23.000000       76.102500             14.400000    85.247500
max     24.000000       84.990000             18.000000    99.950000

       Stress_Level (1-10)
count           464.000000
mean              5.426724
std               2.791343
min               1.000000
25%               3.000000
50%               5.000000
75%               8.000000
max              10.000000
```

In [10]:
```
# Step 6: Identify the quality issues in the dataset to provide a
comprehensive overview of its integrity and completeness.
try:
    print("Columns with Data missing:")
    print(df.isnull().sum())
    print("\nNumber of duplicates in 'Student_ID':",
df['Student_ID'].duplicated().sum())
    print("\nNumber of duplicates in 'E-mail':",
df['Email'].duplicated().sum())
    print("\nCheck for Columns with incorrect data types:")
    print(df.dtypes)


except Exception as e:
    print(f"An unexpected error occurred: {e}")

#https://www.aporia.com/resources/how-to/count-nan-values-
dataframe/#:~:text=We%20can%20use%20the%20isna,together%20with%20isna%20or%
20isnull. Helped me realise I can add .sum() to isnull() function
#https://www.w3schools.com/python/pandas/ref_df_duplicated.asp helped me
understand the use of duplicated()
```

```
Columns with Data missing:
Student_ID                      0
First_Name                      0
Last_Name                       0
Email                           0
Gender                          0
Age                             0
Department                      0
Attendance (%)                516
Midterm_Score                   0
Final_Score                     0
Assignments_Avg               517
Quizzes_Avg                     0
Participation_Score             0
Projects_Score                  0
Total_Score                     0
Grade                           0
Study_Hours_per_Week            0
Extracurricular_Activities      0
Internet_Access_at_Home         0
Parent_Education_Level       1794
Family_Income_Level             0
Stress_Level (1-10)             0
Sleep_Hours_per_Night           0
dtype: int64

Number of duplicates in 'Student_ID': 0

Number of duplicates in 'E-mail': 0

Check for Columns with incorrect data types:
```

```
Student_ID                    object
First_Name                    object
Last_Name                     object
Email                         object
Gender                        object
Age                            int64
Department                    object
Attendance (%)               float64
Midterm_Score                float64
Final_Score                  float64
Assignments_Avg              float64
Quizzes_Avg                  float64
Participation_Score          float64
Projects_Score               float64
Total_Score                  float64
Grade                         object
Study_Hours_per_Week         float64
Extracurricular_Activities    object
Internet_Access_at_Home       object
Parent_Education_Level        object
Family_Income_Level           object
Stress_Level (1-10)            int64
Sleep_Hours_per_Night        float64
dtype: object
```

In [11]:
```
# Step 7: Generate a correlation table for the numerical columns
try:
    print("Data types of columns for Sub DataFrame:\n")
    print(Sub_df.dtypes) #Used to get the columns names that are then added
for correlation analysis

    cor_df = Sub_df[['Attendance
(%)','Study_Hours_per_Week','Final_Score','Stress_Level (1-10)']]


    print("\nCorrelation table:\n")
    print(cor_df.corr())


except Exception as e:
    print(f"An error occurred while generating the correlation table: {e}")
```

```
Data types of columns for Sub DataFrame:

Student_ID              object
Age                      int64
Department              object
Attendance (%)         float64
Study_Hours_per_Week   float64
Final_Score            float64
Stress_Level (1-10)      int64
dtype: object

Correlation table:

                      Attendance (%)  Study_Hours_per_Week
Final_Score  \
Attendance (%)              1.000000            -0.057134
0.064881
Study_Hours_per_Week       -0.057134             1.000000
-0.030125
Final_Score                 0.064881            -0.030125
1.000000
Stress_Level (1-10)         0.007832            -0.016741
-0.033999

                      Stress_Level (1-10)
Attendance (%)                   0.007832
Study_Hours_per_Week            -0.016741
Final_Score                     -0.033999
Stress_Level (1-10)              1.000000
An error occurred while generating the correlation table: could not
convert string to float: 'S1000'
```