Importing Libraries

In [1]:
```
import numpy as np
import pandas as pd
```

Loading the Dataset

In [5]:
```
student = pd.read_csv('Desktop/DataMentor Hub/Python Data Cleaning/students_data.csv')
```

Previewing the First Five Records

In [6]:
```
student.head()
```

Out[6]:

	student_id	name	age	gender	grade	math_score	english_score	science_score	enrolled_date	remarks
0	100	jane smith	16.0	female	11	75.0	NaN	66	2022-06-10	excellent
1	101	John Doe	16.0	Male	10th	74.0	95	94	10-06-2022	GOOD
2	102	Chris P.	NaN	MALE	10	NaN	missing	69	06/12/2022	needs improvement
3	103	jane smith	16.0	FEMALE	10	NaN	missing	62	10-06-2022	average
4	104	Sara O'Neil	16.0	male	11	NaN	96	64	2022-06-10	GOOD

Previewing the Last Five Records

In [8]:
```
student.tail()
```

Out[8]:

	student_id	name	age	gender	grade	math_score	english_score	science_score	enrolled_date	remarks
26	126	John Doe	NaN	Male	10th	64.0	missing	67	06/12/2022	good student
27	127	Simran Singh	16.0	FEMALE	12	NaN	76	80	06/12/2022	poor
28	128	Sara O'Neil	17.0	female	11	NaN	64	89	2022-06-10	average
29	129	Patel R.	17.0	male	11	NaN	64	83	2022/06/11	Average
30	129	Patel R.	17.0	male	11	NaN	64	83	2022/06/11	Average

Checking Dataset Dimensions

In [7]:
```
student.shape
```

Out[7]:
```
(31, 10)
```

Checking Column Names

In [9]:
```
student.columns
```

Out[9]:
```
Index(['student_id', 'name', 'age', 'gender', 'grade', 'math_score',
       'english_score', 'science_score', 'enrolled_date', 'remarks'],
      dtype='object')
```

Checking for Duplicate Records

In [14]:
```
student.duplicated().sum()
```

Out[14]:
```
1
```

Removing Duplicates

In [25]:
```
student.drop_duplicates(inplace = True)
```

In [27]:
```
student
```

Out[27]:

	Student_Id	Name	Age	Gender	Grade	Math Score	English Score	Science Score	Enrolled Date	Remarks
0	100	jane smith	16.0	female	11	75.0	NaN	66	2022-06-10	excellent
1	101	John Doe	16.0	Male	10th	74.0	95	94	10-06-2022	GOOD
2	102	Chris P.	NaN	MALE	10	NaN	missing	69	06/12/2022	needs improvement
3	103	jane smith	16.0	FEMALE	10	NaN	missing	62	10-06-2022	average
4	104	Sara O'Neil	16.0	male	11	NaN	96	64	2022-06-10	GOOD
5	105	Mike O'Reilly	16.0	Female	10	NaN	NaN	83	06/12/2022	needs improvement
6	106	ali Khan	17.0	female	11	64.0	NaN	75	06/12/2022	Good
7	107	Sara O'Neil	17.0	female	12	NaN	63	62	2022/06/11	excellent
8	108	Mike O'Reilly	16.0	Female	12	80.0	missing	89	06/12/2022	poor
9	109	Robert Brown	17.0	female	12	NaN	missing	97	10-06-2022	needs improvement
10	110	lucy gray	17.0	male	11th	65.0	67	100	06/12/2022	excellent
11	111	Simran Singh	16.0	FEMALE	11th	NaN	missing	95	2022-06-10	average
12	112	Patel R.	17.0	female	11	NaN	87	89	2022-06-10	poor
13	113	Patel R.	17.0	male	10	NaN	NaN	98	06/12/2022	Average
14	114	Ali Khan	17.0	male	12	NaN	91	67	2022/06/11	poor
15	115	Lucy gray	16.0	Female	12	65.0	91	94	06/12/2022	Average
16	116	Chris P.	NaN	Female	11	NaN	NaN	72	2022/06/11	excellent
17	117	Ali Khan	18.0	male	11th	67.0	74	81	2022-06-10	GOOD
18	118	Simran Singh	17.0	Male	10	100.0	74	62	2022/06/11	average

	Student_Id	Name	Age	Gender	Grade	Math Score	English Score	Science Score	Enrolled Date	Remarks
19	119	Patel R.	17.0	MALE	11th	73.0	NaN	90	06/12/2022	Good
20	120	Sara O'Neil	17.0	Male	10	NaN	missing	89	2022-06-10	average
21	121	John Doe	18.0	female	11	66.0	72	94	10-06-2022	Average
22	122	Sara O'Neil	17.0	MALE	11th	75.0	NaN	66	2022-06-10	good student
23	123	jane smith	17.0	Female	11th	NaN	missing	63	06/12/2022	excellent
24	124	John Doe	18.0	MALE	11th	NaN	NaN	91	06/12/2022	GOOD
25	125	Mike O'Reilly	17.0	Male	12	94.0	80	63	10-06-2022	Average
26	126	John Doe	NaN	Male	10th	64.0	missing	67	06/12/2022	good student
27	127	Simran Singh	16.0	FEMALE	12	NaN	76	80	06/12/2022	poor
28	128	Sara O'Neil	17.0	female	11	NaN	64	89	2022-06-10	average
29	129	Patel R.	17.0	male	11	NaN	64	83	2022/06/11	Average

Dataset Information Summary

In [28]:
```
student.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 30 entries, 0 to 29
Data columns (total 10 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   Student_Id     30 non-null     int64
 1   Name           30 non-null     object
 2   Age            27 non-null     float64
 3   Gender         30 non-null     object
 4   Grade          30 non-null     object
 5   Math Score     13 non-null     float64
 6   English Score  22 non-null     object
 7   Science Score  30 non-null     int64
 8   Enrolled Date  30 non-null     object
 9   Remarks        30 non-null     object
dtypes: float64(2), int64(2), object(6)
memory usage: 2.6+ KB
```

Renaming Columns

In [29]:
```
cols_dict = {'student_id':'Student_Id', 'name':'Name', 'age':'Age', 'gender':'Gender', 'grade':'Grade', 'math_score'
```

In [30]:
```
student.rename(columns = cols_dict, inplace = True)
```

In [48]:
```
student.head()
```

Out[48]:

	Student_Id	Name	Age	Gender	Grade	Math Score	English Score	Science Score	Enrolled Date	Remarks
0	100	Jane Smith	16.0	female	11	75.0	NaN	66	2022-06-10	excellent
1	101	John Doe	16.0	Male	10th	74.0	95	94	10-06-2022	GOOD
2	102	Chris P.	NaN	MALE	10	NaN	missing	69	06/12/2022	needs improvement
3	103	Jane Smith	16.0	FEMALE	10	NaN	missing	62	10-06-2022	average
4	104	Sara O'Neil	16.0	male	11	NaN	96	64	2022-06-10	GOOD

Rechecking Column Names

In [32]:
```
student.columns
```

Out[32]:
```
Index(['Student_Id', 'Name', 'Age', 'Gender', 'Grade', 'Math Score',
       'English Score', 'Science Score', 'Enrolled Date', 'Remarks'],
      dtype='object')
```

Inspecting Unique Names

In [46]:
```
student['Name'].unique()
```

Out[46]:
```
array(['Jane Smith', 'John Doe', 'Chris P.', "Sara O'Neil",
       'Mike O’Reilly', 'Ali Khan', 'Robert Brown', 'Lucy Gray',
       'Simran Singh', 'Patel R.'], dtype=object)
```

Standardizing Names

In [47]:
```
student.loc[:, ['Name']] =  student['Name'].str.lower().str.title()
```

Inspecting Unique Ages

In [49]:
```
student['Age'].unique()
```

Out[49]:
```
array([16., nan, 17., 18.])
```

Calculating the Average Age

In [60]:
```
avg_age = student['Age'].mean().round()
```

In [61]:
```
avg_age
```

Out[61]:
```
17.0
```

Filling Missing Ages with the Average Age

In [63]:
```
student.loc[:, ['Age']] = student['Age'].fillna(avg_age)
```

Inspecting Unique Gender Values

In [64]:
```
student['Gender'].unique()
```

Out[64]:
```
array(['female', 'Male', 'MALE', 'FEMALE', 'male', 'Female'], dtype=object)
```

Standardizing Gender Values

In [66]:
```
student.loc[:, ['Gender']] = student['Gender'].str.lower().str.title()
```

In [67]:
```
student['Gender'].unique()
```

Out[67]:
```
array(['Female', 'Male'], dtype=object)
```

Inspecting Grade Values

In [69]:
```
student['Grade'].unique()
```

Out[69]:
```
array(['11', '10th', '10', '12', '11th'], dtype=object)
```

Standardizing Grade Format

In [70]:
```
grade_dict = {'10th': '10', '11th': '11'}
```

In [85]:
```
student.loc[:, ['Grade']] = student['Grade'].replace(grade_dict).astype('int')
```

In [86]:
```
student['Grade'].unique()
```

Out[86]:
```
array([11, 10, 12], dtype=object)
```

Inspecting Math Scores

In [74]:
```
student['Math Score'].unique()
```

Out[74]:
```
array([ 75.,  74.,  nan,  64.,  80.,  65.,  67., 100.,  73.,  66.,  94.])
```

Calculating Average Math Score

In [81]:
```
avg_math_score = student['Math Score'].mean().astype('int')
```

In [87]:
```
avg_math_score
```

Out[87]: 74

Filling Missing Math Scores with the Average Score

In [83]:
```
student.loc[:, ['Math Score']] = student['Math Score'].fillna(avg_math_score)
```

In [84]:
```
student['Math Score'].unique()
```

Out[84]:
```
array([ 75.,  74.,  64.,  80.,  65.,  67., 100.,  73.,  66.,  94.])
```

Inspecting English Scores

In [80]:
```
student['English Score'].unique()
```

Out[80]:
```
array([nan, '95', 'missing', '96', '63', '67', '87', '91', '74', '72',
       '80', '76', '64'], dtype=object)
```

Converting English Scores to Numeric

In [91]:
```
student.loc[:, ['English Score']] = pd.to_numeric(student['English Score'], errors = 'coerce')
```

Inspecting Enrollment Dates

In [111…
```
student['Enrolled Date'].unique()
```

Out[111…
```
array(['2022-06-10', '10-06-2022', '06/12/2022', '2022/06/11'],
      dtype=object)
```

Standardizing Enrollment Dates

In [116…
```
date_dict = {'10-06-2022':'2022-06-10', '06/12/2022':'2022-06-12', '2022/06/11':'2022-06-11'}
```

In [117…
```
student.loc[:, ['Enrolled Date']] = student['Enrolled Date'].replace(date_dict)
```

Converting to Datetime Format

In [133…
```
student['Enrolled Date'] = pd.to_datetime(student['Enrolled Date'], errors = 'coerce')
```

Inspecting Remarks

In [119…
```
student['Remarks'].unique()
```

Out[119…
```
array(['excellent', 'GOOD', 'needs improvement', 'average', 'Good',
       'poor', 'Average', 'good student'], dtype=object)
```

Standardizing Remarks

In [120…
```
student.loc[:, ['Remarks']] = student['Remarks'].str.lower().str.title()
```

Final Dataset Summary

In [134…
```
student.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 30 entries, 0 to 29
Data columns (total 10 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   Student_Id     30 non-null     int64
 1   Name           30 non-null     object
 2   Age            30 non-null     float64
 3   Gender         30 non-null     object
 4   Grade          30 non-null     object
 5   Math Score     30 non-null     float64
 6   English Score  30 non-null     float64
 7   Science Score  30 non-null     int64
 8   Enrolled Date  30 non-null     datetime64[ns]
 9   Remarks        30 non-null     object
dtypes: datetime64[ns](1), float64(3), int64(2), object(4)
memory usage: 2.6+ KB
```

Confirmed that:

- All columns have the correct data type.

- No missing values remain.
- Column names are clear and professional.

Dataset is now fully cleaned and ready for analysis or visualization.

In [ ]: