```
# NAME: SAKSHI SONBA PINGALE
# ROLL NO. : 49
# . Data Wrangling, I
# Perform the following operations using Python on any open source dataset (e.g., data
# 1. Import all the required Python Libraries.
# 2. Locate an open source data from the web (e.g., https://www.kaggle.com). Provide a
#  description of the data and its source (i.e., URL of the web site).
# 3. Load the Dataset into pandas dataframe.
# 4. Data Preprocessing: check for missing values in the data using pandas isnull(), d
# function to get some initial statistics. Provide variable descriptions. Types of var
# Check the dimensions of the data frame.
# 5. Data Formatting and Data Normalization: Summarize the types of variables by check
# the data types (i.e., character, numeric, integer, factor, and logical) of the varia
# data set. If variables are not in the correct data type, apply proper type conversio
# 6. Turn categorical variables into quantitative variables in Python.
```

```
# 1. Import all the required Python Libraries.
import pandas as pd
import matplotlib as plt
```

```
# 2. Dataset and its source: https://www.kaggle.com/datasets/devansodariya/student-p
# 3. Load the dataset
data = pd.read_csv("/content/study_performance.csv")
data.head()
```

	gender	race_ethnicity	parental_level_of_education	
0	female	group B	bachelor's degree	
1	female	group C	some college	
2	female	group B	master's degree	
3	male	group A	associate's degree	fre
4	male	group C	some college	

4. Data Preprocessing

```
data.isnull().sum()
```

```
gender                         0
race_ethnicity                 0
parental_level_of_education    0
lunch                          0
test_preparation_course        0
math_score                     0
reading_score                  0
writing_score                  0
dtype: int64
```

```
# describe() function to get some initial statistics
data.describe()
```

	math_score	reading_score	writing_score
count	1000.00000	1000.000000	1000.000000
mean	66.08900	69.169000	68.054000
std	15.16308	14.600192	15.195657
min	0.00000	17.000000	10.000000
25%	57.00000	59.000000	57.750000
50%	66.00000	70.000000	69.000000
75%	77.00000	79.000000	79.000000
max	100.00000	100.000000	100.000000

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 8 columns):
 #   Column                       Non-Null Count  Dtype
---  ------                       --------------  -----
 0   gender                       1000 non-null   object
 1   race_ethnicity               1000 non-null   object
 2   parental_level_of_education  1000 non-null   object
 3   lunch                        1000 non-null   object
 4   test_preparation_course      1000 non-null   object
 5   math_score                   1000 non-null   int64
 6   reading_score                1000 non-null   int64
 7   writing_score                1000 non-null   int64
dtypes: int64(3), object(5)
memory usage: 62.6+ KB
```

```
data.shape
```

```
(1000, 8)
```

5.Data Formatting and Data Normalization

```
data.dtypes
```

```
gender                         object
race_ethnicity                 object
parental_level_of_education    object
lunch                          object
test_preparation_course        object
math_score                      int64
reading_score                   int64
writing_score                   int64
dtype: object
```

```
# Converting Data Types
data['math_score'] = data['math_score'].astype(float)
data.dtypes
```

```
gender                         object
race_ethnicity                 object
parental_level_of_education    object
```

```
lunch                          object
test_preparation_course        object
math_score                    float64
reading_score                   int64
writing_score                   int64
dtype: object
```

```
# Data Normalization
# min-max feature scaling

min = data['writing_score'].min()
print("MIN=" ,min)

max = data['writing_score'].max()
print("MAX=", max)

data['writing_score'] = (data['writing_score'] - min) / (max- min)
print(data["writing_score"])
```

```
MIN= 10
MAX= 100
0      0.711111
1      0.866667
2      0.922222
3      0.377778
4      0.722222
         ...
995    0.944444
996    0.500000
997    0.611111
998    0.744444
999    0.844444
Name: writing_score, Length: 1000, dtype: float64
```

6. Turn categorical variables into quantitative variables in Python.

```
data=data.replace({'female': 0, 'male': 1})
data.head()
```

	gender	race_ethnicity	parental_level_of_education	
0	0	group B	bachelor's degree	
1	0	group C	some college	
2	0	group B	master's degree	
3	1	group A	associate's degree	fre
4	1	group C	some college	

Start coding or generate with AI.