Practical 2: Data Wrangling II

Create an "Academic performance" dataset of students and perform the following operations using Python.

1. Scan all variables for missing values and inconsistencies. If there are missing values and/or inconsistencies, use any of the suitable techniques to deal with them.
2. Scan all numeric variables for outliers. If there are outliers, use any of the suitable techniques to deal with them.
3. Apply data transformations on at least one of the variables. The purpose of this transformation should be one of the following reasons: to change the scale for better understanding of the variable, to convert a non-linear relation into a linear one, or to decrease the skewness and convert the distribution into a normal distribution.

Reason and document your approach properly.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

# Step 1: Create the "Academic Performance" dataset
data = {
  "Name": ["Priya", "Malik", "Pawar", "Aarti", "Sameer"],
  "Gender": ["F", "M", "M", "F", "M"],
  "Science_Score": [88, 91, 78, None, 77],  # Missing value
  "Math_Score": [85, 92, 76, 80, 400],  # An outlier
  "English_Score": [82, 78, 74, 90, 87],
  "Attendance_Percentage": [90, 85, 72, 78, -5],  # Negative value is an inconsistency
}

# Create DataFrame
df = pd.DataFrame(data)

# Step 2: Scan for missing values and inconsistencies
print("\nMissing Values:")
print(df.isnull().sum())

# Fill missing Science_Score values with the median
df['Science_Score'] = df['Science_Score'].fillna(df['Science_Score'].median())
```

```
# Fix negative values in Attendance_Percentage by setting them to the minimum valid
percentage (0)
df['Attendance_Percentage'] = df['Attendance_Percentage'].apply(lambda x: max(x, 0))

# Step 3: Scan numeric variables for outliers
numeric_columns = ['Science_Score', 'Math_Score', 'English_Score',
'Attendance_Percentage']

for col in numeric_columns:
    plt.figure()  #Ensures that the plot starts fresh, without overlapping with previous plots.
    sns.boxplot(x=df[col])
    plt.title(f"Boxplot for {col}")
    plt.show()

# Remove outliers in Math_Score using the IQR (Interquartile Range) method
Q1 = df['Math_Score'].quantile(0.25)
Q3 = df['Math_Score'].quantile(0.75)
IQR = Q3 - Q1 #It gives the range where the middle 50% of the data lies.
#Define the range of acceptable values using the following formulas
lower_bound = Q1 - 1.5 * IQR
upper_bound = Q3 + 1.5 * IQR
#Values outside this range are considered outliers.

df['Math_Score'] = np.where(df['Math_Score'] > upper_bound, df['Math_Score'].median(),
df['Math_Score'])
df['Math_Score'] = np.where(df['Math_Score'] < lower_bound, df['Math_Score'].median(),
df['Math_Score'])

# Step 4: Apply data transformations
# Transform Attendance_Percentage to reduce skewness using square root transformation

#  Handle negative attendance values
df["Attendance_Percentage"] = df["Attendance_Percentage"].apply(lambda x: max(0, x))

# Transform "Attendance_Percentage" to scale it between 0 and 1 (normalisation)
df['Attendance_Rate'] = df['Attendance_Percentage'] / 100



# Document reasoning:
```

```
# - Missing values in Science_Score were filled with the median to avoid bias.
# - Outliers in Math_Score were capped to the median to maintain data integrity.
# - Negative values in Attendance_Percentage were corrected to ensure logical consistency.
# - The Attendance_Percentage variable was normalised to a scale of 0-1 for easier
interpretation.

# Final data summary
print("\nFinal Dataset Summary:")
print(df.describe())
```