TASK: Attendance Rates Before and After Policy Change

Objective: To determine whether a new school policy has affected student attendance rates using statistical analysis in Python with NumPy and pandas.

Scenario:

A school in Uganda has implemented a new policy aimed at improving student attendance. Attendance rates for 20 students were recorded both before and after the policy change. Your task is to analyze this data using Python, specifically utilizing the NumPy and pandas libraries, to test whether the policy has had a significant effect on attendance rates.

Instructions:

Begin by setting up your Python environment. Ensure that Python is installed on your computer, along with the NumPy and pandas libraries, which are essential for numerical computations and data manipulation. If these libraries are not already installed, you can install them using a package manager like pip.

Next, generate two sets of attendance data using Python:

1. **Before the Policy Change**: Create a dataset representing the attendance rates before the policy change. This dataset should have a mean attendance rate of 85%, a standard deviation of 5%, and include 20 students.
2. **After the Policy Change**: Create a second dataset representing the attendance rates after the policy change. This dataset should have a mean attendance rate of 90%, a standard deviation of 4%, and also include 20 students.

Use NumPy to generate random attendance rates that follow a normal distribution with the specified means and standard deviations. Ensure that all attendance rates are realistic by adjusting any values that fall below 0% or above 100%, since attendance rates cannot exceed these bounds.

After generating the data, organize it into a pandas DataFrame for ease of manipulation and analysis. The DataFrame should include the following columns:

- **Student ID**: Assign a unique identifier to each student, ranging from 1 to 20.
- **Attendance Rate Before Policy**: Include the generated attendance rates before the policy change.

- **Attendance Rate After Policy**: Include the generated attendance rates after the policy change.

For each student, calculate the difference in attendance rates before and after the policy change. Add a new column to your DataFrame called "Difference" to store these values. The difference is calculated by subtracting the attendance rate before the policy from the attendance rate after the policy for each student.

Formulate the hypotheses for your statistical test:

- **Null Hypothesis (H_0)**: The new policy has no effect on attendance rates, meaning the mean difference in attendance rates is zero.
- **Alternative Hypothesis (H_1)**: The new policy has increased attendance rates, meaning the mean difference in attendance rates is greater than zero.

Proceed to calculate the paired t-test statistic manually. Begin by computing the mean of the differences (the average of the "Difference" column). Then, calculate the standard deviation of the differences using the sample standard deviation formula, which divides by (n - 1), where n is the sample size. Next, calculate the standard error of the mean difference by dividing the standard deviation of the differences by the square root of the sample size. Finally, compute the t-statistic by dividing the mean difference by the standard error of the mean difference. Since the null hypothesis assumes the mean difference is zero, this formula applies directly.

Determine the critical value and p-value for your test. Set the significance level (alpha) to 0.05. Calculate the degrees of freedom, which is the sample size minus one (n - 1). Refer to a t-distribution table or use statistical software to find the critical t-value for a one-tailed test at the 0.05 significance level with the calculated degrees of freedom. Compare your calculated t-statistic with the critical t-value to decide whether to reject the null hypothesis. Additionally, calculate the p-value associated with your t-statistic and degrees of freedom to assess the statistical significance of your results.

Interpret the results of your analysis. Based on your comparisons, decide whether to reject or fail to reject the null hypothesis. If the t-statistic is greater than the critical t-value and the p-value is less than the significance level, you can reject the null hypothesis. Summarize your findings in the context of the school's policy change, discussing whether there is significant evidence to suggest that the new policy has increased attendance rates.

Throughout your analysis, document your work carefully. Ensure that your code is well-organized and includes comments explaining each step of your analysis. Present your results clearly, possibly including visualizations such as graphs or charts to illustrate the attendance rates before and after the policy change.

Additional Guidance:

- **Understanding Statistical Concepts**: Make sure you grasp the meanings of mean, standard deviation, standard error, t-statistic, degrees of freedom, critical value, and p-value. These concepts are crucial for interpreting your results correctly.
- **Using Python Libraries**: Familiarize yourself with the functions in NumPy and pandas that are relevant for generating random numbers, performing statistical calculations, and manipulating DataFrames.
- **Data Integrity**: Pay careful attention to the accuracy of your data. Ensure that all calculations are performed correctly and that attendance rates remain within the realistic bounds of 0% to 100%.
- **Interpretation**: Focus on interpreting the results in a meaningful way. Explain what the statistical outcomes imply about the effectiveness of the new policy in practical terms.
- **Seeking Assistance**: If you encounter difficulties or have questions about the concepts or programming aspects, do not hesitate to seek help from your instructor or refer to educational resources.

Objective Recap:

By completing this task, you will gain practical experience in statistical analysis using Python. You will learn how to perform hypothesis testing, specifically the paired t-test, and interpret the results in a real-world context. This exercise will enhance your skills in data analysis, programming with Python, and understanding statistical methodologies.