One-Sample t-Test: Step-by-Step Implementation

Step 1: Define the Dataset

We are working with the following sample data:

$$\text{Sample Data: } [50, 52, 47, 49, 51, 53, 48, 50, 49, 52]$$

The known population mean is:

$$\mu_0 = 50$$

Step 2: Check Normality of the Data

Before performing the t-test, we check if the data is normally distributed using the **Shapiro-Wilk test**.

- **Null Hypothesis (H_0)**: The data is normally distributed.
- **Alternative Hypothesis (H_a)**: The data is not normally distributed.

The Shapiro-Wilk test is computed in Python as:

```
stat, p_normality = shapiro(data)
```

If:

$$p_{\text{normality}} > \alpha \quad (\alpha = 0.05),$$

then the data is normally distributed.

Step 3: Calculate the Sample Mean ($\bar{x}$)

The formula for the sample mean is:

$$\bar{x} = \frac{\sum_{i=1}^{n} x_i}{n}$$

Substitute the values:

$$\bar{x} = \frac{50 + 52 + 47 + 49 + 51 + 53 + 48 + 50 + 49 + 52}{10}$$

Simplify:

$$\bar{x} = \frac{501}{10} = 50.1$$

Step 4: Calculate the Sample Standard Deviation (s)

The formula for the sample standard deviation is:

$$s = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{n-1}}$$

1. Compute squared deviations for each data point:

$$(50 - 50.1)^2 = 0.01, \quad (52 - 50.1)^2 = 3.61, \quad (47 - 50.1)^2 = 9.61, \ldots$$

2. Sum the squared deviations:

$$\sum_{i=1}^{n}(x_i - \bar{x})^2 = 32.9$$

3. Divide by $n - 1 = 10 - 1 = 9$:

$$s^2 = \frac{32.9}{9} = 3.66$$

4. Take the square root:

$$s = \sqrt{3.66} \approx 1.91$$

Step 5: Calculate the Sample Size (n)

The number of data points is:

$$n = 10$$

Step 6: Calculate the t-Statistic (t)

The formula for the t-statistic is:

$$t = \frac{\bar{x} - \mu_0}{s/\sqrt{n}}$$

Substitute the values:

$$t = \frac{50.1 - 50}{1.91/\sqrt{10}}$$

Simplify the denominator:

$$\frac{1.91}{\sqrt{10}} = \frac{1.91}{3.16} \approx 0.60$$

Compute the t-statistic:

$$t = \frac{0.1}{0.60} \approx 0.17$$

Step 7: Perform the t-Test Using SciPy

The t-test can also be computed using the `scipy.stats.ttest_1samp` function:

```
t_statistic, p_value = ttest_1samp(data, population_mean)
```

The results are:

$$t_{\text{scipy}} \approx 0.17, \quad p_{\text{value}} \approx 0.87$$

Step 8: Interpret the Results

- **Null Hypothesis** (H_0): The sample mean is equal to the population mean.
- **Alternative Hypothesis** (H_a): The sample mean is not equal to the population mean.

Compare the p-value ($p = 0.87$) with $\alpha = 0.05$:

$$p > \alpha \implies \text{Fail to reject } H_0$$

Conclusion:

The sample mean is not significantly different from the population mean.

Step 9: Visualize the Data

The data can be visualized with a histogram showing the sample distribution. Vertical lines represent:

- The population mean ($\mu_0 = 50$): **red dashed line**.
- The sample mean ($\bar{x} = 50.1$): **blue solid line**.

The Python code for visualization is:

```
plt.figure(figsize=(8, 5))
plt.hist(data, bins=5, alpha=0.7, label='Sample Data')
plt.axvline(x=population_mean, color='red', linestyle='--', label='Population Mean')
plt.axvline(x=mean, color='blue', linestyle='-', label='Sample Mean')
plt.xlabel('Value')
plt.ylabel('Frequency')
plt.title('Sample Data vs. Population Mean')
plt.legend()
plt.show()
```

Compile Solution

```
The data appears to be normally distributed.

Manually Calculated Statistics:
Sample Mean: 50.10
Sample Standard Deviation: 1.91
Sample Size: 10
T-Statistic (Manual Calculation): 0.17

One-Sample t-Test Results (Using SciPy)
T-Statistic (SciPy): 0.17
P-Value: 0.8723
Result: Fail to reject the null hypothesis.
The sample mean is not significantly different from the population mean.
```