

# Learning the Kolmogorov-Smirnov Test: A Practical Guide in Python

Authored by  
**Mohammed loot**

November 7, 2025

## RECOMMENDED CITATION

Mohammed loot (2025). *Learning the Kolmogorov-Smirnov Test: A Practical Guide in Python*. PSYCHOLOGICAL STATISTICS. Retrieved from <https://statistics.arabpsychology.com/?p=12380>

The [Kolmogorov-Smirnov test](#) (commonly abbreviated as the KS test) is a highly versatile and powerful [non-parametric statistical tool](#) used extensively in data analysis. Its primary function is twofold: first, to assess whether a given sample dataset is plausibly drawn from a theoretical probability distribution (the one-sample test), and second, to determine if two independent datasets originate from the same underlying distribution (the two-sample test). This methodology is absolutely foundational, especially when validating the critical assumptions required by parametric procedures--for instance, verifying that data adheres to a [normal distribution](#) before applying t-tests or ANOVA.

At a mechanical level, the KS test quantifies the maximum discrepancy between the observed [Cumulative Distribution Function](#) (CDF) of the sample and the reference CDF (which can be theoretical or empirical). This maximum distance is summarized by the test statistic, typically denoted as  $D$ . A larger  $D$ -statistic implies a greater difference between the compared distributions. A significant advantage of the KS test lies in its generality; it operates without making prior assumptions about the specific shape or parameters of the population distribution, ensuring its broad applicability across various types of data and research questions.

In the Python ecosystem, executing the KS test is simplified through the widely utilized **SciPy** library, specifically contained within the `scipy.stats` module. Data professionals conducting a one-sample test--comparing data against a known distribution--will rely on [scipy.stats.kstest\(\)](#). Conversely, when the goal is to compare two distinct independent samples, determining if they share a common distribution, the function [scipy.stats.ks\\_2samp\(\)](#) is employed. This detailed guide will walk through practical Python implementations and provide clear interpretation strategies for both powerful applications.

## Framing the Statistical Hypotheses

Before any statistical testing commences, it is imperative to formally define the statistical query using the established methodology of [hypothesis testing](#). The Kolmogorov-Smirnov test, whether applied in its one-sample or two-sample form, relies on clearly stated null and alternative hypotheses concerning the distributional equivalence of the data being analyzed.

The formal statements governing the null hypothesis ( $H_0$ ) and the alternative hypothesis ( $H_1$ ) structure the interpretation of the test results:

$H_0$  (Null Hypothesis): This hypothesis asserts that there is no statistically significant difference between the distributions. Specifically, in the one-sample context, the sample data conforms to the specified theoretical distribution. In the two-sample context, the assertion is that both independent samples are drawn from identical underlying distributions.

$H_1$  (Alternative Hypothesis): This hypothesis contradicts the null, asserting that a statistically significant difference exists. The sample data does not follow the specified distribution, or the two

samples originate from fundamentally different distributions.

The conclusive decision--whether to reject or fail to reject the null hypothesis--is fundamentally dependent on the calculated **p-value**. If the resulting p-value falls below the predefined significance level (alpha, typically set at \$0.05\$), we must reject  $H_0$ . This rejection signifies that the observed differences are too large to be attributed to random chance, thereby confirming that the distributions are statistically distinct. Conversely, if the p-value is greater than alpha, we lack sufficient statistical evidence to claim a difference, leading us to fail to reject the null hypothesis, suggesting the distributions are plausibly the same.

## Practical Application: One-Sample KS Test in Python

The one-sample KS test is the perfect tool for validating if a collected dataset aligns with a specific, known theoretical model, such as the widely used normal, uniform, or exponential distributions. In Python, this goodness-of-fit assessment is handled seamlessly by the `scipy.stats.kstest()` function. This function minimally requires two key parameters: the dataset itself and the string identifier corresponding to the theoretical distribution against which the data is being compared.

To provide a robust demonstration, consider a scenario where we intentionally generate data using a process known to be non-normal. We will then test this data against the common assumption of normality. For this case study, we generate 100 data points governed by a [Poisson distribution](#), setting the mean parameter (lambda) to 5. Crucially, before generating any random variables in statistical programming, best practice dictates setting a random seed to ensure the reproducibility of the results across different executions.

The following Python code snippet utilizes the **NumPy** library to generate our sample data:

```
from numpy.random import seed
from numpy.random import poisson
```

```
#set seed for reproducible results
seed(0)
```

```
#generate dataset of 100 values that follow a Poisson distribution with mean=5
data = poisson(5, 100)
```

Now, we proceed to execute the one-sample KS test, specifically challenging the null hypothesis that this `data` could have originated from a standard normal distribution. We import `kstest` from `scipy.stats` and invoke the function, passing our sample data array and the distribution identifier string `'norm'`:

```
from scipy.stats import kstest
```

```
#perform Kolmogorov-Smirnov test against the normal distribution
```

```
kstest(data, 'norm')
```

```
KstestResult(statistic=0.9072498680518208, pvalue=1.0908062873170218e-103)
```

The results clearly show a test statistic (D) of approximately **0.9072** and a corresponding **p-value** that is virtually zero (**1.0908e-103**). Since this p-value is monumentally smaller than our predetermined significance level ( $\alpha = 0.05$ ), we must decisively reject the null hypothesis. The statistical conclusion is unambiguous: there is overwhelming evidence that the sample data is not drawn from a normal distribution. This outcome perfectly validates our experimental setup, confirming that the KS test successfully identified the difference between the Poisson-generated data and the theoretical normal model.

## Practical Application: Two-Sample KS Test for Distribution Comparison

The two-sample KS test, accessible through `scipy.stats.ks_2samp()`, addresses the fundamental question of homogeneity: are two independent samples statistically similar enough to be considered drawn from the same underlying probability distribution? This test holds immense value when comparing outcomes across different experimental groups, analyzing demographic subsets, or checking for consistency between two different manufacturing batches.

To illustrate the power of `ks_2samp()`, we will construct two datasets with known differences. The first dataset (`data1`) will be drawn from a standard normal distribution, characterized by its symmetric, bell-shaped curve. The second dataset (`data2`) will be sampled from a **lognormal distribution**, which is inherently positively skewed. By ensuring these two samples possess dramatically distinct distributional shapes, we create a challenging yet clear test scenario for the KS methodology to resolve.

We begin the preparation process by generating both samples using NumPy's random module, ensuring that the seed is consistently set for complete replication of results:

```
from numpy.random import seed
from numpy.random import randn
from numpy.random import lognormal
```

```
#set seed for reproducible results
```

```
seed(0)
```

```
#generate data1 (Standard Normal) and data2 (Lognormal)
```

```
data1 = randn(100)
data2 = lognormal(3, 1, 100)
```

With both heterogeneous datasets now ready, we invoke the `ks_2samp()` function. This function analyzes the two data arrays provided, testing the null hypothesis that the cumulative distribution functions of `data1` and `data2` are statistically indistinguishable:

```
from scipy.stats import ks_2samp
```

```
#perform Two-Sample Kolmogorov-Smirnov test
ks_2samp(data1, data2)
```

```
KstestResult(statistic=0.99, pvalue=4.417521386399011e-57)
```

The resulting output is highly informative, showing a KS test statistic (D) of **0.99**. This exceptionally high value signals that the maximum vertical separation between the two empirical CDFs is nearly the maximum possible, indicating profound dissimilarity. Consequently, the calculated **p-value** is extremely small, registering at **4.4175e-57**. Since this value is far below the conventional 0.05 threshold, we confidently reject the null hypothesis. The inescapable conclusion is that the two samples are drawn from statistically different distributions, reinforcing the effectiveness of the KS test in detecting major differences in underlying probability structures.

## Interpreting the D-Statistic and P-Value: A Closer Look

Effective utilization of the Kolmogorov-Smirnov test requires a precise understanding of its two core outputs: the D-statistic and the p-value. These metrics are the quantitative foundation upon which we make decisions regarding the distributional fit or equivalence.

The **D-statistic** serves as the measure of magnitude of difference. It is formally defined as the greatest absolute vertical distance observed between the two cumulative distribution functions being compared (either one empirical CDF versus a theoretical CDF, or two empirical CDFs). In essence, it captures the point of maximum deviation. A D-statistic close to zero suggests an excellent fit or similarity between the distributions, whereas a value approaching one indicates severe divergence. For instance, in our first example, the D-statistic of 0.9072 immediately signaled a massive difference between the Poisson sample and the hypothesized Normal distribution.

The **p-value** is the probability of observing a D-statistic as extreme as, or more extreme than, the calculated value, assuming that the null hypothesis ( $H_0$ ) is true. It is the crucial metric for decision-making in statistical hypothesis testing. If this p-value is small (typically less than 0.05), it implies

the observed discrepancy is highly unlikely to have occurred merely by random chance, leading to the rejection of  $H_0$ . Conversely, a large p-value suggests the data is consistent with the null hypothesis. Both of our preceding examples yielded p-values so small they provided undeniable statistical proof that the underlying distributions were distinct.

It is vital to acknowledge a critical caveat regarding the one-sample KS test: the standard procedure assumes that the parameters of the theoretical distribution (e.g., the mean and standard deviation for the normal distribution) are known prior to testing. If these parameters are estimated directly from the sample data itself, the test statistic becomes inherently biased, potentially making the test overly conservative. For situations involving parameters estimated from the sample, specialized modifications, such as the [Lilliefors test](#), are technically more rigorous, though SciPy often provides robust and accurate approximations for common distribution comparisons.

## Considering Alternatives and Best Practices

While the Kolmogorov-Smirnov test is celebrated for its non-parametric robustness and general applicability, practitioners must be aware of its limitations and consider alternatives, particularly when the focus is goodness-of-fit testing for [normality](#)--a common requirement in many statistical models. One known drawback of the KS test is that it can be relatively less sensitive to discrepancies occurring in the extreme tails of the distribution compared to other dedicated tests.

When the need arises to assess distribution fit, especially for smaller samples or when tail sensitivity is critical, the following goodness-of-fit tests are often preferred:

**The Shapiro-Wilk Test:** This method is frequently cited as the most powerful test specifically designed for assessing normality, particularly effective when dealing with smaller sample sizes (N typically less than 50).

**The Anderson-Darling Test:** This test is an adaptation and improvement over the KS test because it places greater weight on deviations observed in the tails of the distribution. It is highly recommended when accurately detecting differences in the extreme values of the data is paramount.

**The Cramér-von Mises Criterion:** Similar to the Anderson-Darling test, this criterion provides a comprehensive measure of the overall difference across the entire Cumulative Distribution Function (CDF), offering another reliable assessment of distributional fit.

Regardless of the specific test chosen within the **SciPy** library, adherence to best practices remains crucial. Always ensure that the data input into the statistical function is a clean, one-dimensional array of numerical values. Furthermore, for comprehensive and transparent reporting, always present both the test statistic (D or its equivalent) and the corresponding p-value. These two figures offer distinct yet complementary insights into the magnitude and significance of the observed difference between the distributions under comparison.

## **Additional Resources for Goodness-of-Fit Testing**

To further advance your proficiency in goodness-of-fit testing and to delve into alternative methods used for assessing critical distributional assumptions in Python, we recommend consulting the following related resources:

[How to Perform a Shapiro-Wilk Test in Python](#)

[How to Perform an Anderson-Darling Test in Python](#)