

Test for Normality in Python (4 Methods)?

Authored by
stats writer

November 28, 2025

RECOMMENDED CITATION

stats writer (2025). *Test for Normality in Python (4 Methods)?*. PSYCHOLOGICAL SCALES.
Retrieved from <https://scales.arabpsychology.com/?p=101237>

Testing for normality is a critical preliminary step in many statistical analyses. The normal distribution, often referred to as the Gaussian distribution or the bell curve, serves as the foundation for numerous powerful parametric statistical tests, such as ANOVA and t-tests.

In the realm of Python and scientific computing, several robust techniques are available to formally assess whether a dataset deviates significantly from a normal distribution. These methods range from simple visual inspections to rigorous hypothesis testing procedures.

We will explore four primary techniques for assessing normality in Python, including both visual and formal statistical approaches: the Shapiro-Wilk test, the Kolmogorov-Smirnov test, and the visual inspection methods using histograms and Q-Q plots. We will also briefly mention other established tests like the Anderson-Darling test and D'Agostino's K-squared test.

Most established statistical tests rely on the assumption that the underlying data distribution is Gaussian. Failure to validate this assumption can lead to inaccurate conclusions regarding hypothesis testing. In Python, we rely on a combination of visual tools and formal hypothesis tests.

The four most common categories of methods employed in Python are:

Visual Method: Creating a Histogram. This provides a rapid, initial assessment of the distribution's shape and symmetry. If the resulting shape approximates the characteristic 'bell-curve', the assumption of normality is tentatively supported.

Visual Method: Constructing a Q-Q plot (Quantile-Quantile Plot). This specialized plot compares the quantiles of the dataset against the theoretical quantiles of a standard normal distribution. If the data is normal, the points should align closely along a straight diagonal line.

Formal Statistical Test: Implementing the Shapiro-Wilk Test. Generally considered one of the most powerful normality tests, especially for smaller to medium sample sizes ($n < 50$), it assesses the goodness-of-fit to a normal distribution.

Formal Statistical Test: Utilizing the Kolmogorov-Smirnov Test (K-S Test). This non-parametric test compares the empirical cumulative distribution function of the sample data with the theoretical cumulative distribution function of the normal distribution.

In the subsequent sections, we will demonstrate the practical implementation and interpretation of these methods using real-world Python code examples, focusing on a dataset known to follow a log-normal distribution, thereby allowing us to clearly identify non-normality.

Understanding the Assumptions of Normality Testing

The requirement for data normality stems from the underlying mathematical assumptions of many parametric procedures. Tests like the Student's t-test, ANOVA, and linear regression rely on the Central Limit Theorem and the assumption that residuals (or the population from which the sample

was drawn) are normally distributed.

If these distributional assumptions are violated, the reported significance levels (α) and confidence intervals may be inaccurate, leading to incorrect statistical inference. While some tests are robust to minor violations, particularly with large sample sizes, severe non-normality necessitates either data transformation or the use of non-parametric equivalents.

A key aspect of formal normality tests is the formulation of the null hypothesis (H_0) and the alternative hypothesis (H_a):

H_0 : The sample data is drawn from a normal distribution.

H_a : The sample data is not drawn from a normal distribution.

We interpret the test results based on the calculated p-value relative to a predetermined significance level (α), conventionally set at 0.05.

Method 1: Visual Assessment via Histogram

The histogram is perhaps the most straightforward way to gain initial insight into the distribution of a dataset. When data is truly normally distributed, the histogram should exhibit a classic, symmetrical 'bell shape', with the majority of observations clustered around the mean and tapering off evenly at the tails.

However, visual assessment is inherently subjective and is best used as a preliminary check rather than definitive proof of normality. Features like skewness (asymmetry) or kurtosis (the heaviness of the tails) immediately signal deviations from the Gaussian ideal. If a strong skew is evident, as often happens with financial or environmental data, formal tests are essential.

The following Python code uses the **NumPy** and **Matplotlib** libraries to generate and plot a dataset derived from a log-normal distribution--a distribution type commonly known to be positively skewed and non-normal. This intentionally non-normal data serves as an excellent example for demonstrating non-compliance with the normality assumption.

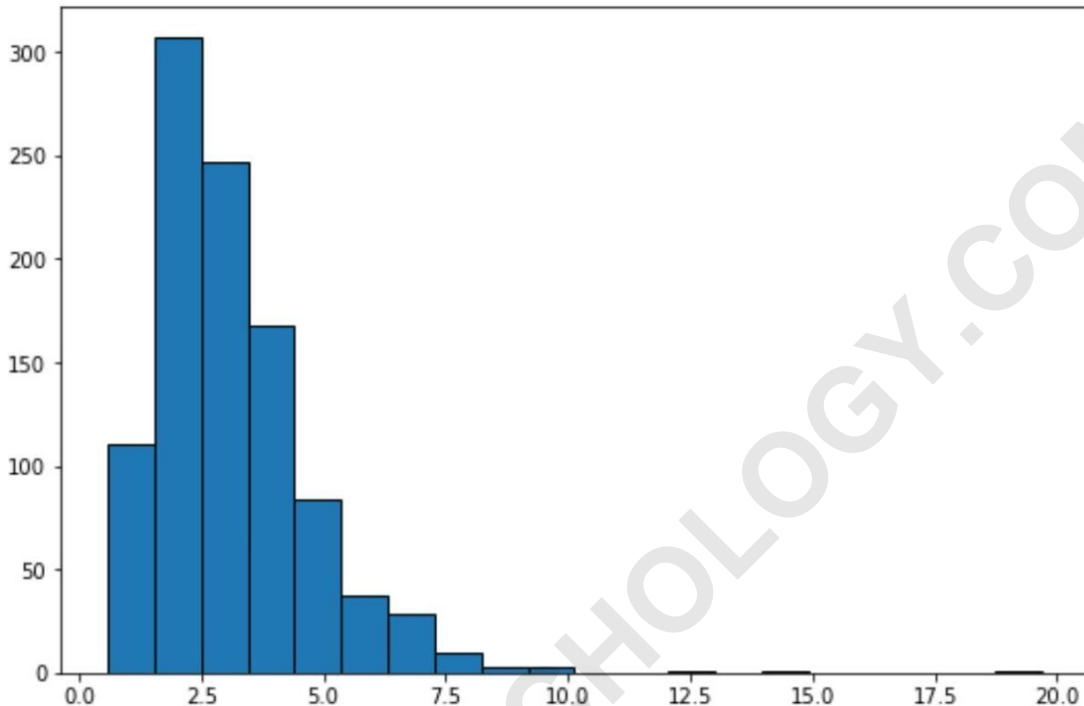
```
import math
import numpy as np
from scipy.stats import lognorm
import matplotlib.pyplot as plt
```

```
#make this example reproducible
np.random.seed(1)
```

```
#generate dataset that contains 1000 log-normal distributed values
```

```
lognorm_dataset = lognorm.rvs(s=.5, scale=math.exp(1), size=1000)
```

```
#create histogram to visualize values in dataset  
plt.hist(lognorm_dataset, edgecolor='black', bins=20)
```



Upon visual examination of this histogram, the severe right-skewness is immediately apparent. The distribution clearly lacks the required symmetry and bell-shape, confirming that the dataset, which was generated using a log-normal distribution function, is definitively not normally distributed.

Method 2: Detailed Analysis Using a Quantile-Quantile (Q-Q) Plot

While the histogram offers a general shape overview, the Q-Q plot provides a much more precise diagnostic tool for assessing normality. A Q-Q plot compares the ordered data points (quantiles) from the actual sample against the theoretical quantiles expected from a standard normal distribution. If the two distributions match, the points should fall perfectly on the straight diagonal reference line (often specified as a 45-degree line).

Deviations from this straight line are highly informative. Curvature at the ends indicates heavy tails (kurtosis issues), while an S-shape or consistent curve suggests skewness. The Q-Q plot is generally considered superior to the histogram for identifying subtle departures from normality, making it an indispensable tool alongside formal statistical tests.

We utilize the powerful statistical modeling library, **Statsmodels**, in conjunction with Matplotlib, to generate the Q-Q plot for our previously defined, non-normal log-normal distribution dataset. The `line='45'` argument specifies the desired reference line.

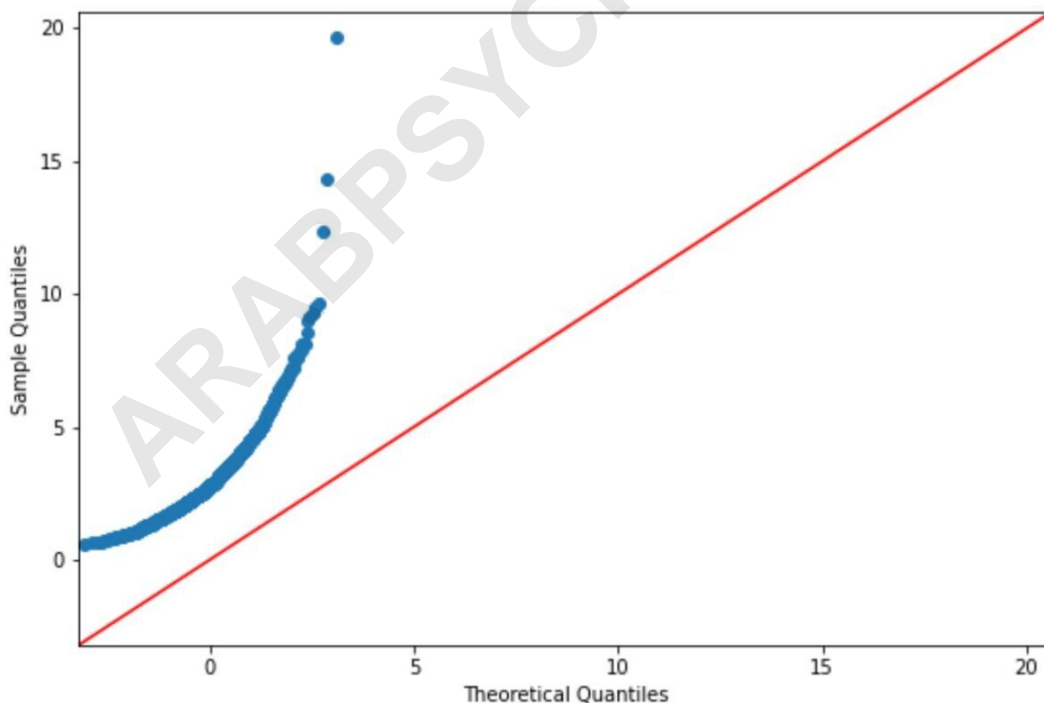
```
import math
import numpy as np
from scipy.stats import lognorm
import statsmodels.api as sm
import matplotlib.pyplot as plt

#make this example reproducible
np.random.seed(1)

#generate dataset that contains 1000 log-normal distributed values
lognorm_dataset = lognorm.rvs(s=.5, scale=math.exp(1), size=1000)

#create Q-Q plot with 45-degree line added to plot
fig = sm.qqplot(lognorm_dataset, line='45')

plt.show()
```



For a dataset to be considered normally distributed, the plotted points must hug the diagonal red line closely. As observed above, the points deviate significantly from the reference line, particularly

at the higher quantiles, exhibiting a strong upward curve characteristic of a positively skewed, non-normal distribution. This visual evidence strongly corroborates the non-normality suggested by the histogram and confirms that the sample is not drawn from a Gaussian population.

Method 3: Formal Testing using the Shapiro-Wilk Test

The Shapiro-Wilk test is one of the most widely recommended tests for assessing normality, particularly for sample sizes ranging from small to medium (up to $n=5,000$). It measures how well the sample data correlates with the corresponding quantiles of a normal distribution. Unlike visual methods, the Shapiro-Wilk test provides an objective, quantifiable result based on a p-value.

The core procedure involves setting up the null hypothesis (H_0), which states that the data is normally distributed. If the resulting p-value is less than the significance level ($\alpha = 0.05$), we reject H_0 , concluding that the data is statistically non-normal. Conversely, if the p-value is greater than 0.05, we fail to reject H_0 , suggesting insufficient evidence to claim non-normality.

We apply the `shapiro` function from the `scipy.stats` module to our non-normal dataset, which was intentionally generated using a log-normal distribution function. This demonstrates how a formal statistical test handles data that visually appears skewed.

```
import math
import numpy as np
from scipy.stats import shapiro
from scipy.stats import lognorm

#make this example reproducible
np.random.seed(1)

#generate dataset that contains 1000 log-normal distributed values
lognorm_dataset = lognorm.rvs(s=.5, scale=math.exp(1), size=1000)

#perform Shapiro-Wilk test for normality
shapiro(lognorm_dataset)

ShapiroResult(statistic=0.8573324680328369, pvalue=3.880663073872444e-29)
```

The output provides two key metrics: the test statistic (W) and the p-value. Here, the W statistic is **0.857**, and the corresponding p-value is **3.88e-29**. This p-value is extremely small--effectively zero.

Since the resulting p-value (3.88×10^{-29}) is far less than our chosen significance level ($\alpha = 0.05$), we decisively reject the null hypothesis. This formal statistical confirmation aligns perfectly with our visual inspection, providing strong evidence that the sample data does not

originate from a normal distribution.

Method 4: Formal Testing using the Kolmogorov-Smirnov Test (K-S Test)

The Kolmogorov-Smirnov test (K-S test) is a non-parametric goodness-of-fit test. It determines if a sample distribution differs significantly from a specified theoretical distribution, in this case, the normal distribution. While the K-S test is versatile and can be used to compare two empirical distributions, when testing for normality, it specifically calculates the maximum distance between the empirical cumulative distribution function (CDF) of the sample and the theoretical CDF of a standard normal distribution.

It is important to note that when testing for normality, the K-S test often performs poorly compared to specialized tests like the Shapiro-Wilk test, especially when the mean and standard deviation of the population are estimated from the sample data. For this reason, many statisticians recommend the Lilliefors test (a modification of K-S) or Shapiro-Wilk for normality checks.

Using the `kstest` function from `scipy.stats`, we pass our log-normal distribution dataset and specify `'norm'` as the theoretical distribution against which to test. The K-S test follows the same hypothesis framework as Shapiro-Wilk: if the p-value is low, we reject the null hypothesis of normality.

```
import math
import numpy as np
from scipy.stats import kstest
from scipy.stats import lognorm

#make this example reproducible
np.random.seed(1)

#generate dataset that contains 1000 log-normal distributed values
lognorm_dataset = lognorm.rvs(s=.5, scale=math.exp(1), size=1000)

#perform Kolmogorov-Smirnov test for normality
kstest(lognorm_dataset, 'norm')

KstestResult(statistic=0.84125708308077, pvalue=0.0)
```

The test returns a K-S statistic of **0.841** and a corresponding p-value of **0.0**. Since $0.0 < 0.05$, we clearly reject the null hypothesis. This conclusion indicates that there is overwhelming statistical evidence to assert that the sample data is not derived from a normal distribution.

Alternative Formal Tests for Normality

While the Shapiro-Wilk and Kolmogorov-Smirnov tests are highly popular in introductory statistics, experts often rely on other dedicated tests that offer better sensitivity to specific deviations from normality, particularly concerning the tails of the distribution (kurtosis).

The **Anderson-Darling test** is an extension of the K-S test but gives greater weight to the tails of the distribution. This increased sensitivity makes it excellent for detecting heavy-tailed or light-tailed distributions, which often violate the assumption of normality. In Python, this test is available in `scipy.stats` using the `anderson` function, although its interpretation requires comparing the test statistic against specific critical values rather than a direct p-value output.

The **D'Agostino's K-squared test**, sometimes called the Omnibus test, specifically combines measures of skewness and kurtosis into a single test statistic. This approach is highly effective because non-normality often manifests as a combination of extreme skewness (asymmetry) and kurtosis (tail thickness). For large sample sizes ($n > 20$), this test is often considered a reliable alternative to the Shapiro-Wilk test, offering strong power against various non-normal shapes.

When selecting a test, the general recommendation is to use the Shapiro-Wilk test for small samples ($n < 50$) due to its high power, and the Anderson-Darling test or D'Agostino's K-squared test for larger datasets where sensitivity to tails or computational speed are concerns. Regardless of the test chosen, the interpretation remains consistent: a small p-value leads to the rejection of the null hypothesis.

Strategies for Handling Non-Normal Data

Discovering that a dataset violates the assumption of normality does not necessitate abandoning the analysis. Statisticians have two main paths forward: applying data transformations or utilizing non-parametric statistical tests that do not rely on distributional assumptions.

Data transformation techniques aim to mathematically adjust the data points such that the distribution becomes more symmetrical and bell-shaped, allowing the use of powerful parametric tests. These methods are particularly effective when dealing with positive skewness, such as that found in our log-normal distribution example. Common transformations include the following:

Log Transformation: Replacing each value x with $\log(x)$. This is highly effective for reducing severe positive skewness and stabilizing variance. It is the natural choice for data known to follow a log-normal distribution.

Square Root Transformation: Replacing each value x with \sqrt{x} . This is a milder transformation than the logarithm and is often used when dealing with count data or moderately skewed distributions.

Cube Root Transformation: Replacing each value x with $x^{1/3}$. This provides an intermediate level of skew correction, useful when the log transformation is too aggressive but the square root is insufficient.

If transformation fails to achieve normality or if the resulting transformed data is difficult to interpret, the alternative is to employ non-parametric methods. These methods (e.g., Mann-Whitney U test, Kruskal-Wallis H test) analyze ranks instead of raw data values, bypassing the need for a Gaussian distribution assumption entirely. The choice between transformation and non-parametric testing depends on the severity of the non-normality and the desired interpretability of the results.

For practical instruction on implementing these mathematical adjustments, please refer to [this detailed tutorial on transforming data in Python](#).