

How to Calculate Skewness & Kurtosis in Python

Authored by
stats writer

December 21, 2025

RECOMMENDED CITATION

stats writer (2025). *How to Calculate Skewness & Kurtosis in Python*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=108202>

Analyzing the structure and characteristics of data is fundamental to statistical inference and machine learning. Beyond central tendencies (like the mean or median) and dispersion (like variance or standard deviation), statisticians rely on measures that describe the overall shape of the probability distribution. The two most critical statistics for describing distribution shape are Skewness and Kurtosis.

In the realm of data science, Python provides robust tools for these computations, primarily through the scipy.stats module. This module offers dedicated functions--specifically `skew()` and `kurtosis()`--designed to efficiently calculate these shape metrics from a given dataset. These functions accept data points provided as a Python list or NumPy array, returning a precise floating-point value.

This comprehensive guide delves into the theoretical basis of Skewness and Kurtosis and provides a step-by-step tutorial on calculating and interpreting these crucial measures using the powerful computational capabilities of Scipy in Python.

Understanding the Geometry of Data Distributions

In statistics, understanding the graphical representation of a dataset--its distribution--is paramount. When visualizing data, such as with a histogram, we often look for ideal characteristics, particularly symmetry. However, real-world data rarely conforms perfectly to these ideals. Skewness and Kurtosis provide quantifiable metrics that move beyond simple visual inspection, offering precise numerical descriptions of deviations from standard shapes.

These two measures serve distinct but complementary roles. Skewness focuses on the horizontal symmetry of the distribution, evaluating whether the mass of the data is concentrated equally on both sides of the center. If the distribution is perfectly symmetrical, its skewness will be zero. Conversely, Kurtosis examines the vertical characteristics, specifically concentrating on the height of the peak (peakedness) and the heaviness of the tails relative to a benchmark distribution, usually the Normal Distribution.

By combining these two metrics, analysts gain a much richer understanding of the underlying data generation process. For instance, high positive skewness might indicate external factors creating extreme high values, while high Kurtosis often signals a greater risk of extreme outliers compared to a standard model. Calculating these measures is essential for validating statistical assumptions, especially when modeling data using regression or time series analysis, where normality is frequently a prerequisite.

Deconstructing Skewness: Measuring Asymmetry

Skewness is formally defined as the third standardized moment of a probability distribution. It quantifies the degree and direction of asymmetry. When a distribution is perfectly symmetrical--such as the Normal Distribution--the mean, median, and mode coincide, resulting in a skewness value of exactly zero. However, when these central measures diverge, the distribution is skewed, meaning one tail is longer or fatter than the other.

The direction of the skew is determined by whether the extended tail stretches toward the positive or negative end of the number line. There are three primary classifications of Skewness: **Negative Skew** (Left Skewed), **Positive Skew** (Right Skewed), and Zero Skew. Understanding the practical implication is crucial: the skew points toward the direction of the outliers. For instance, data like income distribution often exhibits strong positive skewness because most people earn moderate salaries, but a few individuals earn extremely high amounts, pulling the average higher than the median.

We can summarize the interpretations based on the calculated value:

Negative Skew (Left-Skewed): The left tail is longer or heavier. The majority of the data falls on the right side of the mean. This suggests the presence of lower extreme values or outliers. Mathematically, the Mean is typically less than the Median.

Positive Skew (Right-Skewed): The right tail is longer or heavier. The mass of the distribution is concentrated on the left side. This indicates the existence of higher extreme values or outliers. Mathematically, the Mean is typically greater than the Median.

Zero Skew: The distribution is perfectly symmetrical around its mean. The Mean, Median, and Mode are equal, indicating a balanced distribution of values.

When analyzing Skewness, it is important to consider the magnitude. A skewness value close to zero (e.g., between -0.5 and 0.5) is generally considered acceptable for models assuming normality, while values far outside this range suggest significant asymmetry that may require data transformation or the use of non-parametric methods.

The Role of Kurtosis in Tail Analysis

While Skewness assesses horizontal symmetry, **Kurtosis** measures the shape's "tailedness" and peakedness, specifically in relation to the Normal Distribution. Formally, Kurtosis is the fourth standardized moment. A common misconception is that it only measures the peak height; in reality, it is a measure of how much variance arises from the tails versus the shoulders of the distribution. High Kurtosis generally means heavy, long tails and a sharp central peak.

The comparison benchmark is crucial. The standard Normal Distribution, which is classified as **mesokurtic**, has a Kurtosis value of 3 (when using the standard definition, often called Pearson's definition). Distributions are classified into three types based on how their tails compare to this

standard: **Leptokurtic**, **Mesokurtic**, and **Platykurtic**. In financial risk analysis, high Kurtosis is particularly significant as it implies a higher probability of observing extreme events (outliers) than a normal model would predict.

It is vital to distinguish between standard Kurtosis (Pearson) and **Excess Kurtosis** (Fisher). Because the mesokurtic value (Normal Distribution) is 3, many statistical packages, including Scipy, automatically calculate the Excess Kurtosis by subtracting 3 from the raw value. This allows for a more direct interpretation relative to the Normal Distribution (where Excess Kurtosis is 0).

Leptokurtic (Kurtosis > 3; Excess Kurtosis > 0): These distributions have heavier tails and a sharper, taller peak than the normal distribution. They are characterized by a greater chance of generating extreme outliers.

Mesokurtic (Kurtosis = 3; Excess Kurtosis = 0): This is the characteristic of the Normal Distribution. The distribution of data in the tails is standard.

Platykurtic (Kurtosis < 3; Excess Kurtosis < 0): These distributions have lighter tails and a flatter peak than the normal distribution. They tend to produce fewer and less extreme outliers, meaning the data is more uniformly spread out around the mean.

When using Python's statistical libraries, researchers must always check the default definition being used. If the output is near zero, it typically signifies the use of Fisher's Excess Kurtosis definition, making the interpretation relative to zero straightforward.

Utilizing the Scipy Library for Shape Metrics

For efficient and accurate statistical computation in Python, the Scipy ecosystem is indispensable. Specifically, the `scipy.stats` module provides comprehensive statistical functions, including `skew()` and `kurtosis()` (or `kurt()`). Before running any calculation, ensure that the library is installed and imported, typically alongside NumPy for handling array structures, as these functions are optimized for numerical arrays.

The primary complexity when calculating these moments is deciding whether to compute the population statistic or the sample statistic. When working with a subset of a larger population (a sample), certain adjustments--or biases--must be corrected to ensure the resulting estimate is unbiased. This correction involves using specific divisors in the formulas for Skewness and Kurtosis.

In the `scipy.stats` functions, this critical distinction is managed by the `bias` parameter:

Setting `bias=True` (the default) calculates the raw population Skewness or Kurtosis, assuming the input array represents the entire population.

Setting `bias=False` implements the necessary corrections to calculate the **sample** skewness or

kurtosis, which is standard practice when analyzing experimental or survey data. This is crucial for obtaining statistically robust estimates for the population parameters.

Furthermore, the `kurtosis()` function in `Scipy` defaults to calculating **Excess Kurtosis** (Fisher's definition), meaning it subtracts 3 from the raw result. If the standard Pearson definition is required, an additional parameter must be specified, though for most comparative analyses against the Normal Distribution, the Excess Kurtosis is preferred.

Practical Implementation: Calculating Skewness in Python

To illustrate the practical calculation of these shape metrics, we will use a sample dataset representing scores from a hypothetical test. First, we must import the necessary modules, namely `scipy.stats` for the functions and `numpy`, which is often used in conjunction with `Scipy`, even though Python lists are acceptable input. Defining the data as a NumPy array is usually recommended for computational efficiency.

Consider the following raw data points representing the scores:

```
import numpy as np
from scipy.stats import skew, kurtosis

data = np.array()
print(f"Dataset size: {len(data)}")
```

We are interested in the **sample skewness**, as this dataset of 15 scores is likely a sample drawn from a much larger student population. Therefore, we must explicitly set the `bias` parameter to `False` within the `skew()` function. This tells `Scipy` to use the unbiased calculation formula, adjusting the divisor to ensure the result accurately estimates the population parameter.

Executing the calculation yields the following result:

```
data =

# calculate sample skewness
skew(data, bias=False)

0.032697
```

The resulting value, approximately **0.0327**, immediately suggests a distribution that is very close to

symmetrical. Had the value been, for example, 1.5 or -2.0, this would necessitate deeper investigation into the cause of the extreme asymmetry, perhaps through visualization using a histogram or a box plot to identify potential influential outliers.

Practical Implementation: Calculating Kurtosis in Python

Following the calculation of Skewness, the next step is to calculate the **Kurtosis** for the same dataset. As noted previously, the `kurtosis()` function in `scipy.stats`, when used without specifying the `fisher` parameter, defaults to calculating Fisher's Excess Kurtosis, meaning the expected value for a Normal Distribution is zero.

Just as with Skewness, we must apply the sample correction by setting `bias=False`. This ensures that the Kurtosis estimate is unbiased. The function used is simply `kurtosis()`, which is an alias for the older `kurt()` function and is generally preferred for clarity.

Executing the calculation for the sample kurtosis involves passing the data array and the appropriate parameters:

```
# calculate sample kurtosis
kurtosis(data, bias=False)
```

```
0.118157
```

If an analyst needed the raw Pearson Kurtosis (where a Normal Distribution equals 3), they would need to specify `fisher=False` in the function call, in addition to managing the bias correction. However, for most modern statistical testing, the Excess Kurtosis value is the standard required input. The computed value of approximately **0.1182** for the Excess Kurtosis is now ready for interpretation alongside the Skewness value.

Interpreting the Results of Shape Measures

Our analysis yielded two critical values for the dataset of test scores:

Sample Skewness: **0.032697**

Sample Excess Kurtosis: **0.118157**

These numerical outputs provide a robust quantification of the data's shape, allowing us to move beyond subjective visual assessment. The interpretation process requires comparing these values against the standard benchmarks of zero for Skewness and zero for Excess Kurtosis (the Normal Distribution baseline).

Starting with Skewness, the value of 0.032697 is positive but extremely close to zero. This indicates that the distribution is **slightly positively skewed** (right-skewed). In practical terms, while the distribution is near-symmetrical, there is a marginal tendency for the mean to be pulled slightly higher than the median due to a few higher scores. Because the value is so small (well within the typical acceptable range of -0.5 to 0.5), we can confidently treat this distribution as symmetrical for most practical statistical modeling purposes.

Next, we examine the Excess Kurtosis value of 0.118157. Since this value is positive (greater than zero), the distribution is classified as **Leptokurtic**. A Leptokurtic distribution signifies that the data has slightly heavier tails and a sharper peak than a standard Normal Distribution. This suggests that while the overall skew is negligible, the data generates slightly more extreme outliers than would be expected under a purely normal model. However, similar to Skewness, the magnitude is small, indicating that the deviation from mesokurtic behavior is minimal.

Advanced Considerations: Sample vs. Population and Alternative Libraries

While the `scipy.stats` library offers the most direct and academically rigorous method for calculating skewness and kurtosis in Python, analysts often encounter other powerful libraries that provide similar functionality, notably **Pandas** and **Statsmodels**. Pandas, the primary tool for data manipulation, includes built-in methods like `.skew()` and `.kurt()` on its Series and DataFrame objects. These Pandas methods also default to calculating **sample** statistics, applying the necessary bias correction, making them highly convenient for exploratory data analysis (EDA).

A crucial advanced consideration revolves around the use of the `bias=False` argument we applied. When dealing with large datasets, the difference between the population calculation (biased) and the sample calculation (unbiased) becomes negligible. However, for smaller samples ($N < 30$), failing to set `bias=False` can lead to significant systematic errors in the estimation of the true population Skewness and Kurtosis. Therefore, maintaining the discipline of explicitly setting `bias=False` is a hallmark of rigorous statistical practice, particularly in academic research or critical regulatory environments.

Furthermore, while Scipy handles the standard moments well, for highly non-normal data or for specific modeling requirements, other metrics may be needed. For instance, sometimes the **L-moments** (Linear moments) are preferred over the traditional product moments (which Skewness and Kurtosis are) because L-moments are less sensitive to outliers. While these advanced techniques are beyond the scope of a standard calculation tutorial, recognizing the limitations of the traditional moments is key to becoming an expert statistical analyst.

Final Thoughts and Next Steps

The ability to accurately compute and interpret Skewness and Kurtosis is an essential skill for any

data analyst or quantitative researcher. These metrics provide the necessary quantitative foundation to evaluate the distributional assumptions that underpin most parametric statistical models. When the data deviates significantly from normality (i.e., high skew or high excess kurtosis), analysts must consider transformations (like logarithmic or square root transformations) or switch to robust statistical models that do not rely on the assumption of normality.

The `scipy.stats` library makes these complex calculations straightforward in [Python](#), ensuring quick, reliable results. Always remember to prioritize the sample calculation (`bias=False`) unless the entire population is available. Furthermore, be aware of the default definitions: [Scipy](#) provides Excess Kurtosis by default, aligning with modern comparative statistics.

For those looking for immediate results or educational practice, online calculators can be an excellent resource for verification. For example, various statistical resources offer dedicated calculators that process raw data inputs and automatically yield both the skewness and kurtosis values, confirming the calculations performed in [Python](#).

You can also calculate the skewness for a given dataset using the [arabpsychology Skewness and Kurtosis Calculator](#), which automatically calculates both the skewness and kurtosis for a given dataset.