

How can multivariate normality tests be performed in Python?

Authored by
stats writer

April 20, 2024

RECOMMENDED CITATION

stats writer (2024). *How can multivariate normality tests be performed in Python?*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=137327>

Multivariate normality tests in Python can be performed through various statistical packages such as SciPy and Statsmodels. These tests aim to determine whether a set of data follows a multivariate normal distribution, which is a key assumption in many statistical analyses. To perform these tests, the first step is to import the necessary packages and load the dataset into Python. Then, using functions such as "normaltest" or "shapiro" from the SciPy package, the data can be tested for normality based on its skewness and kurtosis values. Additionally, the Statsmodels package offers the "multivariate_normality" function which can test for normality using the Chi-square test or the Jarque-Bera test. These tests provide statistical measures and p-values to determine the level of normality in the data. By performing these tests, researchers can assess the validity of their results and make appropriate adjustments if necessary.

Perform Multivariate Normality Tests in Python

When we'd like to test whether or not a single variable is normally distributed, we can create a Q-Q plot to visualize the distribution or we can perform a formal statistical test like an Anderson Darling Test or a Jarque-Bera Test.

However, when we'd like to test whether or not *several* variables are normally distributed as a group we must perform a multivariate normality test.

This tutorial explains how to perform the Henze-Zirkler multivariate normality test for a given dataset in Python.

If we'd like to identify outliers in a multivariate setting, we can use the Mahalanobis distance.

Example: Henze-Zirkler Multivariate Normality Test in Python

The Henze-Zirkler Multivariate Normality Test determines whether or not a group of variables follows a multivariate normal distribution. The null and alternative hypotheses for the test are as follows:

H0 (null): The variables follow a multivariate normal distribution.

Ha (alternative): The variables *do not* follow a multivariate normal distribution.

To perform this test in Python we can use the multivariate_normality() function from the pingouin library.

First, we need to install pingouin:

```
pip install pingouin
```

Next, we can import the `multivariate_normality()` function and use it to perform a Multivariate Test for Normality for a given dataset:

```
#import necessary packagesfrom pingouin import
```

```
multivariate_normality  
import pandas as pd  
import numpy as np  
  
#create a dataset with three variables x1, x2, and x3  
df = pd.DataFrame({'x1':np.random.normal(size=50),  
'x2': np.random.normal(size=50),  
'x3': np.random.normal(size=50)})  
  
#perform the Henze-Zirkler Multivariate Normality Test  
multivariate_normality(df, alpha=.05)  
  
HZResults(hz=0.5956866563391165,  
pval=0.6461804077893423, normal=True)
```

The results of the test are as follows:

H-Z Test Statistic: 0.59569p-value: 0.64618

Since the p-value of the test is not less than our specified alpha value of .05, we fail to reject the null hypothesis. The dataset can be assumed to follow a multivariate normal distribution.

Learn how the Henze-Zirkler test is used in real-life medical applications in [this research paper](#).