

# How can I perform a runs test in Python?

Authored by  
**stats writer**

April 20, 2024

## RECOMMENDED CITATION

stats writer (2024). *How can I perform a runs test in Python?*. PSYCHOLOGICAL SCALES.  
Retrieved from <https://scales.arabpsychology.com/?p=137320>

A runs test is a statistical analysis method used to determine if a sequence of data values is random or exhibits a pattern. To perform a runs test in Python, one can use the `scipy.stats` module which provides functions for calculating runs tests. The first step is to import the module and load the data into an array. Then, the appropriate function can be called to calculate the test statistic and p-value. The result can be interpreted to determine if the data is random or exhibits a trend. Additionally, visualizations can be created using Python libraries such as `matplotlib` to further analyze the data. Overall, performing a runs test in Python involves utilizing built-in functions and libraries to analyze the data and draw conclusions about its randomness.

## Perform Runs Test in Python

**Runs test is a statistical test that is used to determine whether or not a dataset comes from a random process.**

**The null and alternative hypotheses of the test are as follows:**

**H0 (null): The data was produced in a random manner.**

**Ha (alternative): The data was *not* produced in a random manner.**

**This tutorial explains two methods you can use to perform Runs test in Python.**

**Example: Runs Test in Python**

**We can perform Runs test on a given dataset in Python by using the `runstest_1samp()` function from the `statsmodels` library, which uses the following syntax:**

```
runtest_1samp(x, cutoff='mean', correction=True)
```

where:

**x:** Array of data values  
**cutoff:** The cutoff to use to split the data into large and small values. Default is 'mean' but you can also specify 'median' as an alternative.  
**correction:** For a sample size below 50, this function subtracts 0.5 as a correction. You can specify False to turn this correction off.

This function produces a z-test statistic and a corresponding p-value as the output.

The following code shows how to perform Run's test using this function in Python:

```
from statsmodels.sandbox.stats.runs import  
runtest_1samp  
  
#create dataset  
data =  
  
#Perform Runs test  
runtest_1samp(data, correction=False)
```

**(-0.6708203932499369, 0.5023349543605021)**

The z-test statistic turns out to be -0.67082 and the corresponding p-value is 0.50233. Since this p-value is not less than  $\alpha = .05$ , we fail to reject the null hypothesis. We have sufficient evidence to say that the data was produced in a random manner.

**Note:** For this example we turned off the correction when calculating the test statistic. This matches the formula that is used to perform a Runs test in R, which does not use a correction when performing the test.