

# How can a binomial test be performed in Python?

Authored by  
**stats writer**

April 17, 2024

## RECOMMENDED CITATION

stats writer (2024). *How can a binomial test be performed in Python?*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=136251>

A binomial test is a statistical test used to determine whether a proportion or percentage is significantly different from a predetermined value. In order to perform a binomial test in Python, the following steps can be followed:

1. Import the necessary libraries such as "scipy.stats" and "numpy" to perform statistical calculations.
2. Define the sample size, number of successes, and the expected proportion or percentage.
3. Use the "binom\_test" function from the "scipy.stats" library to calculate the p-value for the test.
4. Compare the obtained p-value with the desired significance level to determine whether the null hypothesis can be rejected or not.
5. Optionally, the "binom" function from the "numpy" library can be used to generate a binomial distribution and plot a histogram of the results.

By following these steps, a binomial test can be easily performed in Python to analyze and draw conclusions about the proportion or percentage of a given sample.

## Perform a Binomial Test in Python

**A binomial test compares a sample proportion to a hypothesized proportion.**

**For example, suppose we have a 6-sided die. If we roll it 12 times, we would expect the number "3" to show up 1/6 of the time, which would be  $12 * (1/6) = 2$  times.**

**If the number "3" actually shows up 4 times, is that evidence that the die is biased towards the number "3"? We could perform a binomial test to answer that question.**

In Python, we can perform a binomial test using the `binom_test` function from the `scipy.stats` library, which uses the following syntax:

```
binom_test(x, n=None, p=0.5, alternative='two-sided')
```

where:

**x:** number of "successes"  
**n:** total number of trials  
**p:** the probability of success on each trial  
**alternative:** the alternative hypothesis. Default is 'two-sided' but you can also specify 'greater' or 'less.'

This function returns the p-value of the test. We can load this function by using the following syntax:

```
from scipy.stats import binom_test
```

The following examples illustrate how to perform binomial tests in Python.

**Example 1:** We roll a 6-sided die 24 times and it lands on the number "3" exactly 6 times. Perform a binomial test to determine if the die is biased towards the number "3."

The null and alternative hypotheses for our test are as follows:

**H0:**  $\pi \leq 1/6$  (the die is not biased towards the number "3")

**HA:**  $\pi > 1/6$

*\* $\pi$  is the symbol for population proportion.*

We will enter the following formula into Python:

```
binom_test(x=6, n=24, p=1/6, alternative='greater')
```

```
0.1995295129479586
```

**Example 2:** We flip a coin 30 times and it lands on heads exactly 19 times. Perform a binomial test to determine if the coin is biased towards heads.

The null and alternative hypotheses for our test are as follows:

**H0:**  $\pi \leq 1/2$  (the coin is not biased towards heads)

**HA:**  $\pi > 1/2$

We will enter the following formula into Python:

```
binom_test(x=19, n=30, p=1/2, alternative='greater')
```

```
0.10024421103298661
```

Because this p-value (0.10024) is not less than 0.05, we fail to reject the null hypothesis. We do not have sufficient evidence to say the coin is biased towards heads.

**Example 3:** A shop makes widgets with 80% effectiveness. They implement a new system that they hope will improve the rate of effectiveness. They randomly select 50 widgets from a recent production run and find that 47 of them are effective. Perform a binomial test to determine if the new system leads to higher effectiveness.

The null and alternative hypotheses for our test are as follows:

**H<sub>0</sub>:  $\pi \leq 0.80$**  (the new system does not lead to an increase in effectiveness)

**H<sub>A</sub>:  $\pi > 0.80$**

**We will enter the following formula into Python:**

```
binom_test(x=47, n=50, p=0.8, alternative='greater')
```

```
0.005656361012155314
```

**Because this p-value (0.00565) is less than 0.05, we reject the null hypothesis. We have sufficient evidence to say the new system leads to an increase in effectiveness.**

ARABPSYCHOLOGY.COM